

Universidad de la República
Facultad de Ingeniería

Proyecto de fin de carrera

Recarga Fácil por Radio Frecuencia

RF²

Daniel Aicardi, Melina Rabinovich, Edgardo Vaz

Tutores: Ing. Juan Pablo Oliver, Ing. Andrés Aguirre

Montevideo, Uruguay

Agosto 2011

Recarga Fácil por Radio Frecuencia

Resumen

El presente documento describe el prototipo Recarga Fácil por Radio Frecuencia, RF², realizado como proyecto de fin de carrera de Ingeniería Eléctrica en la Universidad de la República entre marzo de 2010 y julio de 2011. El mismo consiste en un sistema embebido para recarga y consulta de tarjetas RFID, como las que se utilizan hoy día en el Sistema de Transporte Metropolitano, y fue diseñado para operar de forma autónoma interactuando directamente con el usuario.

El hardware fue enteramente diseñado por el grupo de trabajo a excepción de la single board computer. Las herramientas de software utilizadas son open source, así como también las bibliotecas usadas para desarrollar la aplicación final. El diseño, la fabricación, y el armado del prototipo, fueron realizados en su totalidad en Uruguay.

Agradecimientos

En primer lugar queremos agradecer a nuestras familias y amigos. Agradecemos al grupo mina del INCO, a Leonardo Steinfeld, Nicolás Barabino, Francisco Lanzari, María Eugenia Corti, Santiago Reyes, Viterbo Rodríguez, Silvana Castro, Christian Gutierrez, Andrés Bergeret, Gonzalo Tabares, Klaus Rotzinger, Marcelo Fiori, Pablo Cancela, Ana y Claudia Rabino. Y a todos los que de alguna forma u otra colaboraron con nosotros.

A nuestras familias.

Prefacio

“El ciudadano Línea saca su billetera, extrae su tarjeta y la introduce en la máquina registradora; una serie de gestos automáticos. Unas mandíbulas de aluminio se cierran sobre ella, unos dientes de cobre buscan la clave magnética, y una lengua electrónica saborea la vida del ciudadano Línea. Lugar y fecha de nacimiento. Padres. Raza. Religión. Historial educativo, militar y de servicios civiles. Estado. Hijos. Ocupaciones, desde el comienzo hasta el presente. Asociaciones. Medidas físicas, huellas digitales, retiniales, grupo sanguíneo. Grupo psíquico básico. Porcentaje de lealtad, índice de lealtad en función del tiempo hasta el momento del último análisis... ... El ciudadano Línea se encuentra en la ciudad donde, la noche anterior, dijo que estaría, así que no ha tenido que hacer una corrección. Los nuevos informes se añaden al historial del ciudadano Línea. Toda su vida regresa al banco de datos. Desaparece de la unidad exploradora y la unidad comparativa, para que éstas atiendan la próxima llegada. La máquina ha tragado y digerido otro día. Está satisfecha.”

Sam Hall (1953), Poul Anderson

La narración anterior es parte de un cuento de ciencia ficción llamado “Sam Hall”, escrito por Poul Anderson en 1953. En esta historia el autor describe un mundo donde cada persona tiene asignada una tarjeta conteniendo datos que la caracterizan, y puede ser controlado su accionar a través de una super computadora que almacena y procesa los datos de toda la humanidad. En nuestros días este cuento de ciencia ficción no está tan alejado de la realidad, las tarjetas inteligentes (smart cards) son cada vez más usadas en múltiples aplicaciones como ser, pasaporte electrónico, pago electrónico, sistemas de transporte, controles de acceso y sistemas de seguridad, entre otros. El siguiente proyecto se desarrolla con la intención de aprender las bases del mundo de las tarjetas inteligentes y que sirva como punto de partida para que otros entiendan su funcionamiento. Los autores no desean que se use el contenido de este documento con fines como los que se indicaban en la narrativa de ciencia ficción, muy por el contrario, el empleo de esta tecnología debe estar en favor de las personas y no en su contra.

Índice general

Título	I
Resumen	II
Agradecimientos	III
Dedicatoria	IV
Prefacio	V
Tabla de contenidos	VI
Índice de figuras	X
Índice de cuadros	XII
Glosario	XIII

I Introducción 1

1. Introducción	2
1.1. Objetivo general	4
1.2. Antecedentes	4
1.3. Alcance	5
1.4. Especificación funcional	5
1.5. Criterios de éxito	5

II Diseño 6

2. Funcionamiento del prototipo	7
2.1. Requerimientos	7
2.2. Descripción del prototipo	7
2.3. Funcionamiento general del prototipo	9

3. Hardware	11
3.1. Arquitecturas estudiadas	11
3.2. Arquitectura seleccionada	14
3.3. Elección de hardware	16
3.3.1. SBC	16
3.3.2. VLT - Conversor de Voltajes	17

3.3.3. SCUI - Lector de tarjetas de contacto e Interfaz de Usuario	18
3.3.4. Lector/Escritor RFID	20
3.4. Funcionamiento de módulos	21
3.4.1. SBC	21
3.4.2. VLT - Conversor de Voltajes	21
3.4.3. SCUI - Lector de tarjetas de contacto e Interfaz de Usuario	22
3.4.4. Lector/Escritor RFID	23
4. Software	28
4.1. Introducción	28
4.2. Arquitectura de Software	28
4.2.1. Descripción	28
4.2.2. Sistema Operativo	30
4.3. Herramientas utilizadas en el desarrollo del sistema	31
4.3.1. Introducción	31
4.3.2. MLO, u-boot.bin y uImage	31
4.3.3. FileSystem	32
4.3.4. Croscompilación	32
4.3.5. Depuración de código	32
4.3.6. Bibliotecas	33
4.3.7. Otras herramientas	34
4.4. Desarrollo	34
4.4.1. MLO	34
4.4.2. Multiplexado de pines	34
4.4.3. u-boot	35
4.4.4. uImage	39
4.4.5. FileSystem	43
4.4.6. Beagleboard	44
Preparación de la memoria SD	44
Configuración de los parámetros de arranque	44
Copia de MLO y u-boot.bin en NAND	46
Beagleboard en la red	48
4.4.7. Bibliotecas	49
4.4.8. Aplicación final	56
III Ensayos	63
5. Ensayos	64
5.1. SBC	64
5.2. VLT - Conversor de Voltajes	65
5.3. SCUI - Lector de tarjetas de contacto e Interfaz de usuario	66
5.4. Lector/Escritor RFID	70

IV Gestión de proyecto	77
6. Compras y costos	78
6.1. SBC	79
6.2. PCBs	79
6.3. VLT	80
6.4. SCUI	81
6.5. Lector/Escritor RFID	83
7. Tiempos	86
V Conclusiones	89
8. Conclusiones	90
8.1. Conclusión final	90
8.2. Ventajas y desventajas	91
8.3. Aprendizaje	92
8.4. Mejoras y trabajos a futuro	93
VI Apéndices	95
A. Tarjetas inteligentes (Smart Cards)	96
A.1. Clasificaciones	96
A.1.1. Tipos de tarjetas según su capacidad	96
A.1.2. Tipos de tarjetas según la estructura de su sistema operativo	97
A.1.3. Tipos de tarjetas según el formato (tamaño)	98
A.1.4. Tipos de tarjetas según la interfaz	98
A.2. ISO 14443	101
A.3. Mifare	102
A.3.1. Operación	103
B. Lector/Escritor RFID	106
B.1. Reglas y Parámetros de Diseño de una Antena RF	106
B.1.1. Diseño del inductor	107
B.1.2. Capacitores del circuito resonante	108
B.1.3. Sintonizar el circuito resonante	108
B.1.4. Valor de ITVDD	110
B.1.5. Factor de calidad Q	110
B.1.6. Circuito receptor	112

C. Documentos y esquemáticos del hardware diseñado	115
C.1. Herramientas de diseño	115
C.2. Esquemáticos y componentes	115
C.2.1. SBC	117
C.2.2. VLT - Conversor de Voltajes	118
C.2.3. SCUI - Lector de tarjetas de contacto e Interfaz de Usuario	120
C.2.4. Lector/Escritor RFID	125
D. Software	129
D.1. Gestor de paquetes opkg	129
D.2. Instalación, configuración y uso de SDK	131
D.3. OpenEmbedded-Bitbake	132
D.4. uImage	137
D.5. Instalación y configuración de librfid-tool	141
D.6. Depuración de código	144
D.7. Depuración remota	145
D.8. Preparación de la memoria SD	145
D.8.1. Formateo de la memoria SD	145
D.8.2. Copia de archivos a la SD	150
D.9. Configuración en el PC para conexión serial con la Beagleboard	151
D.10.printenv	151
D.11.Ejemplo de arranque del sistema	153
D.12.Conexión de Beagleboard a internet	154
D.13.Instalación de PCSC-Lite, CCID y pcsc-tools, y agregado de lector serial	154
D.14.Pruebas sobre las interfaces	157
D.14.1. led.c	157
D.14.2. uart.c	160
E. Hojas de datos	163
VII Bibliografía	165
Bibliografía	166

Índice de figuras

1.1.	Bloques que conforman el sistema a diseñar	3
2.1.	Vista general del prototipo	8
2.2.	Vista anversa y reversa del prototipo	8
2.3.	Diagrama de flujo	10
3.1.	Solución considerada 1	11
3.2.	Solución considerada 2	12
3.3.	Solución considerada 3	12
3.4.	Solución considerada 4	13
3.5.	Solución considerada 5	13
3.6.	Diagrama de bloques de la arquitectura seleccionada	15
3.7.	Vista anversa y reversa de la SBC	17
3.8.	Vista anversa y reversa de VLT	18
3.9.	Vista anversa y reversa de SCUI	19
3.10.	Vista anversa y reversa del lector/escritor RFID	20
3.11.	Arquitectura de una celda I/O del TXB0108	22
4.1.	Sistema Linux	29
4.2.	Memoria SD para funcionar en Beagleboard	30
4.3.	Editor de configuración del kernel	41
4.4.	Configuración SPI	41
4.5.	Configuración USB Gadget	42
4.6.	Capas de software de trabajo	50
4.7.	Capas de software en una primera etapa	51
4.8.	Capa de software RFID	54
4.9.	Capas de software del sistema RF ²	56
4.10.	Estructura de árbol de aplicación RF ²	58
5.1.	Señal en canal Rx para un valor de resistencia R9 de 500Ω	67
5.2.	Señal en canal Rx para un valor de resistencia R9 de 90Ω	68
5.3.	Detector de campo magnético casero	71
5.4.	Lector/escritor RFID con bobina	72
5.5.	Portadora modulada (canal 1) y señal moduladora (canal 2)	73

5.6. Módulo de impedancia z vs. frecuencia ($C_1 = 10\text{pF}$, $C_2 = 47\text{pF}$)	74
5.7. Módulo de impedancia z vs. frecuencia ($C_1 = 10\text{pF}$, $C_2 = 94\text{pF}$)	75
A.1. Tarjeta de contacto	98
A.2. Acoplamiento entre lector y tarjeta	102
A.3. Mifare Classic de 4K	103
B.1. Forma de pulso acorde a la norma ISO 14443	111
B.2. Esquema de una antena, identificando sus principales secciones	113
B.3. Blindaje de una antena en un diseño de 4 capas	114
C.1. Esquemático de la placa VLT - Voltage Level Translator	119
C.2. Esquemático de la placa SCUI	122
C.3. Esquemático del lector de tarjetas de contacto, incluido en la placa SCUI	123
C.4. Esquemático de la interfaz de usuario, incluido en la placa SCUI	124
C.5. Esquemático del módulo digital del lector/escritor RFID	127
C.6. Esquemático de la antena RFID, Inductor + Adaptación de impedancia .	128

Índice de cuadros

4.1.	Modo de pines en bloque de expansión	37
6.1.	Single board computer y lista de accesorios.	79
6.2.	Lista de placas de circuito impreso.	79
6.3.	Lista de componentes de la placa de circuito impreso VLT.	80
6.4.	Lista de componentes del lector de tarjetas de contacto, SC.	81
6.5.	Lista de componentes para la interfaz de usuario, LCD.	82
6.6.	Lista de componentes del lector/escritor RFID. Antena.	83
6.7.	Lista de componentes del lector/escritor RFID.	85
6.8.	Comparación del costo del prototipo	85
7.1.	Descripción de tareas más importantes	88
B.1.	Duración de los pulsos en μs - ISO 14443	111
C.1.	Conector 14x2 Beagleboard – VLT	116
C.2.	Conector 20x2 VLT - SCUI	117
C.3.	SBC y lista de accesorios	117
C.4.	Lista de componentes de la placa de circuito impreso VLT	118
C.5.	Lista de componentes del lector de tarjetas de contacto, SC	120
C.6.	Lista de componentes para la interfaz de usuario, LCD	121
C.7.	Lista de componentes de la antena RF, Inductor + Adaptación de impedancia	125
C.8.	Lista de componentes del lector/escritor RFID, sin la antena RF	126

Glosario

- AFE** – Artefacto Feo de Exhibir, dispositivo para recargar tarjetas RFID.
- APDU** – Application Protocol Data Unit, comando enviado desde la capa de aplicación a través de un lector a una tarjeta inteligente.
- ASCII** – American Standard Code for Information Interchange, Código Estadounidense Estándar para el Intercambio de Información.
- ASIC** – Application-Specific Integrated Circuit, circuito integrado para aplicaciones específicas
- ASK** – Amplitude Shift Key, modulación por amplitud de pulsos.
- ATR** – Answer To Reset, respuesta de una tarjeta inteligente luego de un reset.
- baud rate** – Usado de manera análoga a la tasa de bits por segundo.
- CCID** – Chip/Smart Card Interface Devices controlador genérico de dispositivos lectores de smart card con interfaz USB.
- CL RC632** – Circuito integrado para lectores de tarjetas RFID de protocolo múltiple.
- EMC** – Electromagnetic Compatibility, conjunto de reglas dadas por la FCC para la compatibilidad electrónica de sistemas electrónicos.
- ext3** – Sistema de archivos extendido, usado en sistemas operativos Linux.
- FAT32** – Sistema de archivos FAT (File Allocation Table) de 32 bits, desarrollado para MS-DOS.
- FCC** – Federal Communications Commission, Comisión Federal de Comunicaciones de EEUU.
- FIFO** – First In First Out, concepto usado para indicar que el primer dato en ser leído es el primero en ser procesado y liberado.
- fileSystem** – Sistema de archivos manejado por el sistema operativo.
- GDB** – GNU Debugger, herramienta para depurar código fuente.
- GPIO** – General Purpose Input/Output puerto de entrada/salida de propósito general.
- I/O** – Input/Output puerto de entrada/salida.
- I2C** – Inter-Integrated Circuit, bus de comunicaciones serie para interconectar microcontroladores y/o circuitos integrados entre sí.
- IDE** – Integrated Development Environment, herramienta para el desarrollo de software.

IIE – Instituto de Ingeniería Eléctrica.

ISO – International Organization for Standardization, organización internacional para la estandarización.

ISR – Interrupt Service Routine, rutina de atención a una interrupción.

JTAG – Joint Test Action Group, puerto que permite entre otras cosas depurar la aplicación que se ejecuta en el sistema embebido.

LCD16x2 – Liquid Crystal Display, pantalla de dos líneas de 16 caracteres cada una.

LDO – Low Drop Output, salida de baja caída en reguladores de tensión.

lector mudo – Lector de tarjetas, que no cuenta con un ASIC entre la tarjeta y el microcontrolador.

Mifare – Tecnología de tarjetas inteligentes sin contacto (TISC) que cumplen con el estándar ISO14443.

Mifare Classic – Tarjetas inteligentes sin contacto, con memoria no volátil de 1K o 4K.

open-firmware – Software que permite el manejo directo del hardware donde se encuentra almacenado, cuyo código es abierto, permitiendo ser modificado y distribuído en forma libre.

open-hardware – Dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público.

open source – Software cuyo código es abierto, permitiendo ser modificado y distribuído en forma libre.

OpenPCD – Dispositivo lector/escritor de tarjetas RFID.

OTG – On The Go, clase de puerto USB que puede ser usado como Host o como Device.

PC – Personal Computer, computador personal.

PCB – Printed Circuited Board, placa de circuito impresa.

pcscelite – Biblioteca para el uso con smart cards.

PIC – Familia de microcontroladores tipo RISC (reduced instruction set computer) fabricados por Microchip Technology Inc.

Protocolo T=0 – Protocolo de datos orientado a byte.

Protocolo T=1 – Protocolo de datos orientado a bloques.

RF – Radiofrecuencia.

RF² – Recarga Fácil por Radio Frecuencia.

RS232 – Interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (Data Communication Equipment, Equipo de Comunicación de datos).

RSTPD – Reset and Power Down, pin de reset y apagado del integrado CL RC632.

SBC – Single Board Computer.

SC – Smart Card, tarjeta inteligente.

SCUI – PCB que contiene un lector/escritor de tarjetas de contacto e interfaz de usuario.

SD – Secure Digital, es un formato de tarjeta de memoria.

SDK – Software Development Kit, herramienta para desarrollo de software.

SIMO – Slave In Master Out, entrada esclavo salida maestro en un puerto SPI.

SOC – System On a Chip.

SOMI – Slave Out Master In, salida esclavo entrada maestro en un puerto SPI.

SPI – Serial Peripheral Interface, bus de comunicación serie.

UART – Universal Asynchronous Receiver-Transmitter, puerto serie con transferencia de datos de forma asíncrona.

UID – Código único de identificación de tarjeta RFID.

USB – Universal Serial Bus.

VLT – Voltage Level Translator, nombre del PCB que contiene circuitos integrados para el traslado de niveles de tensión.

Parte I

Introducción

Capítulo 1

Introducción

En el año 2010, se pone en marcha en Montevideo el Sistema de Transporte Metropolitano. El mismo propone el pago de viajes a través de una tarjeta que simplemente se acerca a un dispositivo para efectuarlo. Esa tarjeta logra comunicarse con el dispositivo en forma inalámbrica utilizando radiofrecuencia y se denomina tarjeta RFID.

El usuario del Sistema de Transporte Metropolitano puede recargar su tarjeta en los puntos de venta. Allí debe presentar su tarjeta e indicar el monto que desea acreditar, un operador recibe la tarjeta y la apoya sobre un dispositivo que se encuentra conectado a un PC de escritorio. Luego que la tarjeta es leída aparecen en pantalla los datos asociados a la misma, nombre del usuario, saldo disponible, etc., que están almacenados en un servidor centralizado en la Intendencia de Montevideo (IM). El operador utiliza un programa que por un lado registra la transacción en el servidor central de la IM y por otro actualiza el saldo en la tarjeta RFID.

Por razones de seguridad cada dispositivo de consulta/recarga posee una tarjeta de contactos denominada SAM (Security Access Module) que se utiliza para encriptar las transacciones entre el terminal y el servidor central, y por otro lado se encarga de generar las claves de acceso para la comunicación entre la tarjeta RFID y el terminal. Esta tarjeta es la encargada de dar seguridad en la transacción de modo que el sistema no sea vulnerable.

Si el usuario desea consultar el saldo disponible en la tarjeta, podrá observarlo en el boleto cuando haga uso de su tarjeta en el ómnibus, o a través de un operador en los

puntos de venta.

Se puede desprender de lo anterior, que para poder consultar y/o recargar una tarjeta, es necesario un sistema (ver figura 1.1) que pueda comunicarse con el servidor central de la IM, con las tarjetas RFID de los usuarios a efectos de leerlas y escribirlas, con la tarjeta de seguridad, SAM, y que cuente con algún tipo de interfaz de usuario.

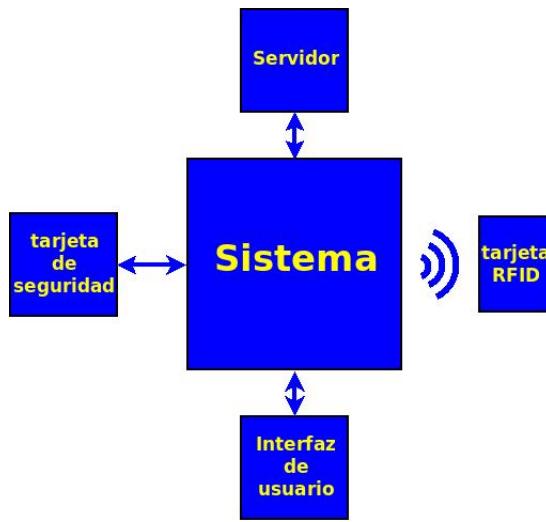


Figura 1.1: Bloques que conforman el sistema a diseñar

Este proyecto busca desarrollar un dispositivo autónomo, de bajo costo y mantenimiento, que permita realizar estas operaciones funcionando en línea con un servidor, de forma rápida, segura y autogestionada por parte del usuario, sin necesidad de personal, en diversos puntos de Montevideo.

La propuesta entonces es realizar un dispositivo, a partir del cual se puedan consultar y recargar tarjetas como las utilizadas en el Sistema de Transporte Metropolitano. En este sistema, el usuario que desee consultar el saldo actual no tiene más que acercar su tarjeta y esperar a que el sistema le indique el saldo disponible en la misma. Para poder acreditar saldo en su tarjeta debería antes efectuar el pago de dinero que desea acreditarle, a través de una red de pagos, mensajes de texto, web, o un mecanismo similar que se encuentra desacoplado del sistema implementado. La transacción anterior actualiza el servidor central con el saldo pendiente para que al acercar la tarjeta al dispositivo simplemente se acredite el saldo por el cual se pagó.

1.1. Objetivo general

El objetivo del proyecto es la fabricación de un prototipo de sistema embebido capaz de consultar y recargar tarjetas. Para ésto, como se mencionó en el punto 1.1, deberá lograr establecer comunicación con tarjetas como las utilizadas en el Sistema de Transporte Metropolitano (comunicación RFID a 13,56MHz), con tarjetas de contacto (módulo de seguridad SAM), con el usuario a través de una interfaz simple y con el servidor central.

Esto implica entonces la fabricación de dos lectores/escritores de tarjetas, uno para tarjetas RFID (sin contacto) y otro para tarjetas con contacto (SAM), una interfaz para el usuario capaz de informar el estado de la transacción mediante mensajes adecuados, y la utilización de un sistema basado en un microprocesador para controlar los periféricos y realizar las operaciones. Esto último implica además el desarrollo del software para que todo funcione adecuadamente.

1.2. Antecedentes

Existen antecedentes de todas las partes a diseñar.

El lector/escritor de tarjetas empleado en el Sistema de Transporte Metropolitano (IntegriSys iMFR) cumple la función de lectura y escritura de las tarjetas sin contacto y además contiene dentro un lector de tarjetas de contacto usado para la seguridad en las transacciones. Otro ejemplo de lector/escritor de ambos tipos de tarjetas, es el SCM SDI010.

Hay lectores/escritores exclusivamente de tarjetas de contacto como puede ser el Omnikey 3121, y lectores/escritores de tarjetas sin contacto como el ACR120.

Algo más completo (a nivel hardware), es un dispositivo lector/escritor RFID llamado OpenPCD [29], de hardware abierto, fabricado en Alemania.

El único prototipo completo que se conoce como antecedente es el realizado por el grupo de electrónica de la IM llamado AFE (Artefacto Feo de Exhibir), que realiza lo

mismo que se propone pero con un enfoque de diseño diferente, ya que utiliza lectores/escritores de escritorio como los mencionados antes y no de diseño propio.

1.3. Alcance

Respecto al hardware, se fabricará un lector/escritor RFID (basado en el antecedente OpenPCD), un lector de tarjetas de contacto donde se insertará el módulo de seguridad (SAM), una interfaz de usuario donde se incluirá un display, leds y un indicador sonoro. Se estudiará la forma de conectar dichos periféricos a un sistema central capaz de controlar todas las funcionalidades.

Respecto al software, se desarrollará todo el software necesario para que el sistema funcione, haciendo lo posible para reutilizar código ya implementado para fines similares y lograr la mayor compatibilidad con lo ya existente.

1.4. Especificación funcional

El prototipo final deberá ser capaz de interactuar con tarjetas RFID a través del lector/escritor RFID, con una tarjeta de contacto SAM (seguridad) y con un servidor. Luego de los controles correspondientes en la tarjeta RFID, comenzará la interacción con el usuario mediante un display, leds e indicador sonoro, que será la interfaz de comunicación con el mismo. El display informará al usuario de las tareas que se estén realizando con mensajes cortos y descriptivos. Los tiempos de recarga y consulta deberán ser menores a un minuto.

1.5. Criterios de éxito

El proyecto será considerado exitoso si se logra construir un dispositivo de sistema embebido capaz de consultar y recargar tarjetas RFID, con tiempos de operación razonablemente cortos.

Parte II

Diseño

Capítulo 2

Funcionamiento del prototipo

2.1. Requerimientos

El principal requerimiento a cumplir es la interacción con tarjetas RFID (ver apéndice A), tanto para su lectura como escritura. La comunicación con tarjetas de contacto (ver apéndice A) es necesaria para la interacción con un módulo de seguridad que permita, la generación de las claves utilizadas para autenticarse con las tarjetas RFID, y una transacción segura con un servidor. En ambos casos es necesario cumplir con las normas y estándares adecuados (tarjetas RFID - ISO 14443 y tarjetas de contacto - ISO7816). Por último mantener informado al usuario de lo que sucede durante una transacción a través de una interfaz visual y sonora.

2.2. Descripción del prototipo

Este prototipo integra la lista de dispositivos que hoy en día se denominan sistemas embebidos. Su hardware está integrado por un sistema basado en un microprocesador que recibe el nombre de Single Board Computer (SBC), a la que se agrega un conversor de niveles (VLT) que permite interconectarla con, un lector/escritor de tarjetas RFID a través de un puerto SPI, un lector de tarjetas de contacto a través de un puerto serial (UART), y la interfaz de usuario compuesta por un buzzer, tres leds (rojo, amarillo, verde) y un display conectado a través de puertos de entrada/salida de propósito general (GPIO).

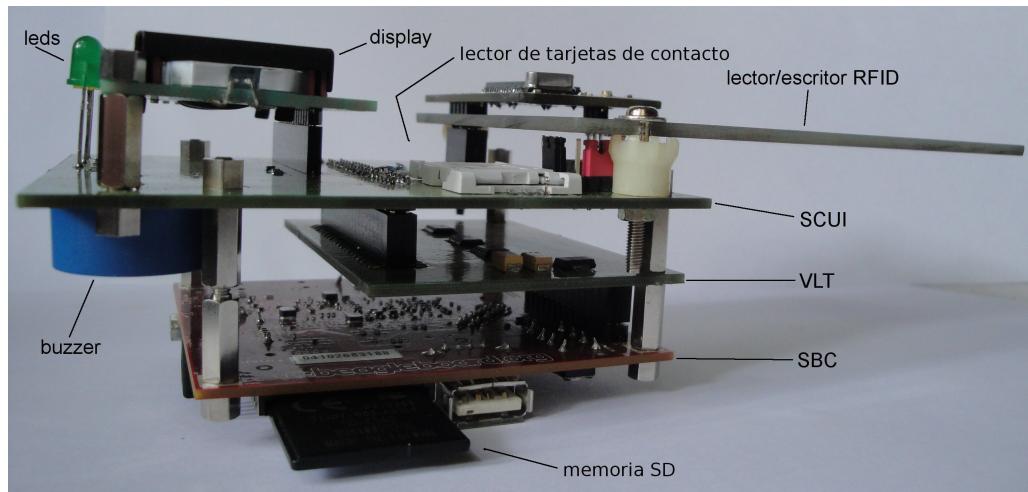
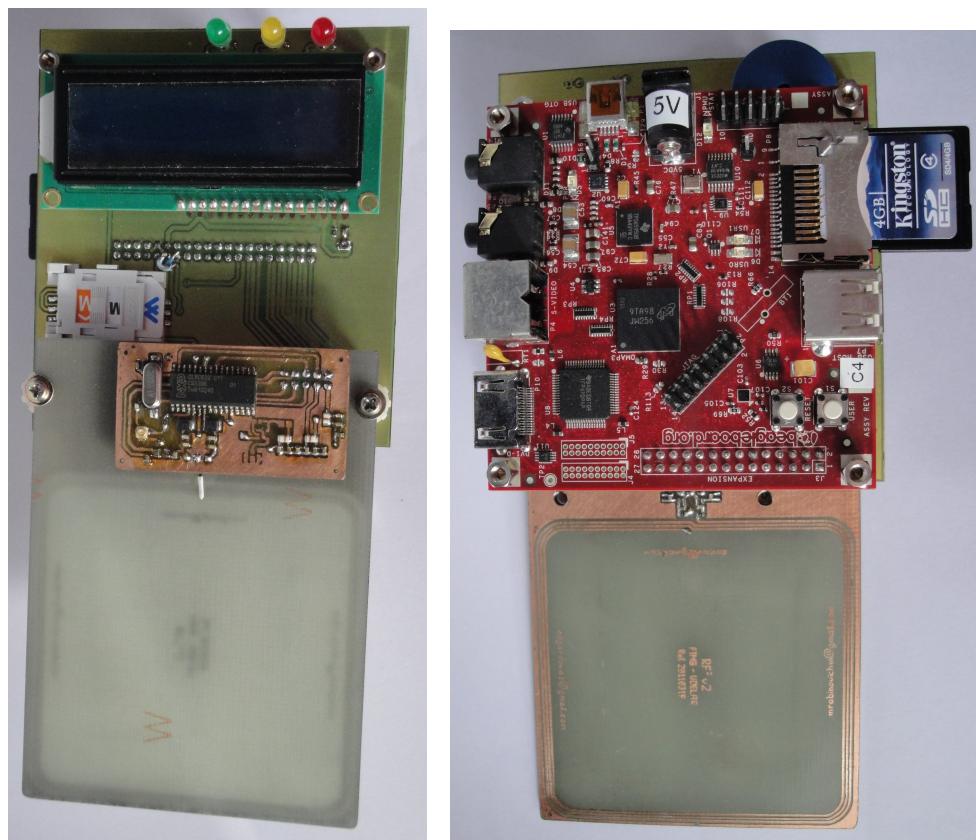


Figura 2.1: Vista general del prototipo



(a) Vista anversa del prototipo

(b) Vista reversa del prototipo

Figura 2.2: Vista anversa y reversa del prototipo

En cuanto al software, está basado en bibliotecas de código abierto que permiten desarrollar la aplicación que asegura el manejo del hardware y el funcionamiento de todo el sistema en conjunto.

2.3. Funcionamiento general del prototipo

Una vez que el prototipo RF² se encuentra operativo, el dispositivo despliega en el display el mensaje “Aproxime su tarjeta”, permaneciendo en dicho estado hasta que algún usuario acerque una tarjeta al lector/escritor RFID. En la primera transacción entre lector y tarjeta se obtiene el identificador único (UID) de ésta última, que será enviado al módulo de seguridad SAM (previa autenticación exitosa), para que a partir de éste, se generen las claves de acceso que permitan la lectura y escritura de la tarjeta RFID. Mientras se lleva a cabo la operación, se despliega en el display el mensaje, “No retire su tarjeta” a la vez que el led amarillo es encendido para indicar precaución ya que se están procesando datos. La siguiente acción a llevar a cabo es verificar que la tarjeta del usuario tenga saldo pendiente de acreditar, en caso afirmativo se indica al usuario el saldo a acreditar a través del display con el mensaje “Saldo a acreditar \$...”. Si todo fue exitoso, se borra el saldo transferido de la lista de saldos pendientes a acreditar para que no se transfiera saldo indefinidas veces. A continuación se despliega en el display el nuevo monto almacenado en la tarjeta, “Su saldo es de \$...”, se enciende el led verde y se emite un pitido mediante el buzzer en señal que la operación fue satisfactoria. Por último se muestran en el display los mensajes “Transacción finalizada”, “Gracias” y vuelve al inicio para comenzar un nuevo ciclo.

En caso que la tarjeta no tuviera saldo pendiente de acreditar, el prototipo RF² funciona en modo consulta y despliega en el display el saldo disponible en la tarjeta, “Su saldo es de \$...”, encendiendo el led verde y emitiendo un pitido, seguido de los mensajes “Transacción finalizada”, “Gracias” y vuelve al inicio para comenzar un nuevo ciclo.

En caso de ocurrir un error durante alguno de los pasos anteriores, ya sea porque el usuario retiró la tarjeta en un momento inadecuado, o simplemente porque el prototipo RF² no logró leer o escribir la tarjeta en forma correcta, se enciende el led rojo, se emite un doble pitido mediante el buzzer, y el display muestra el mensaje “Error, vuelva a intentarlo”, acto seguido el ciclo vuelve a comenzar.

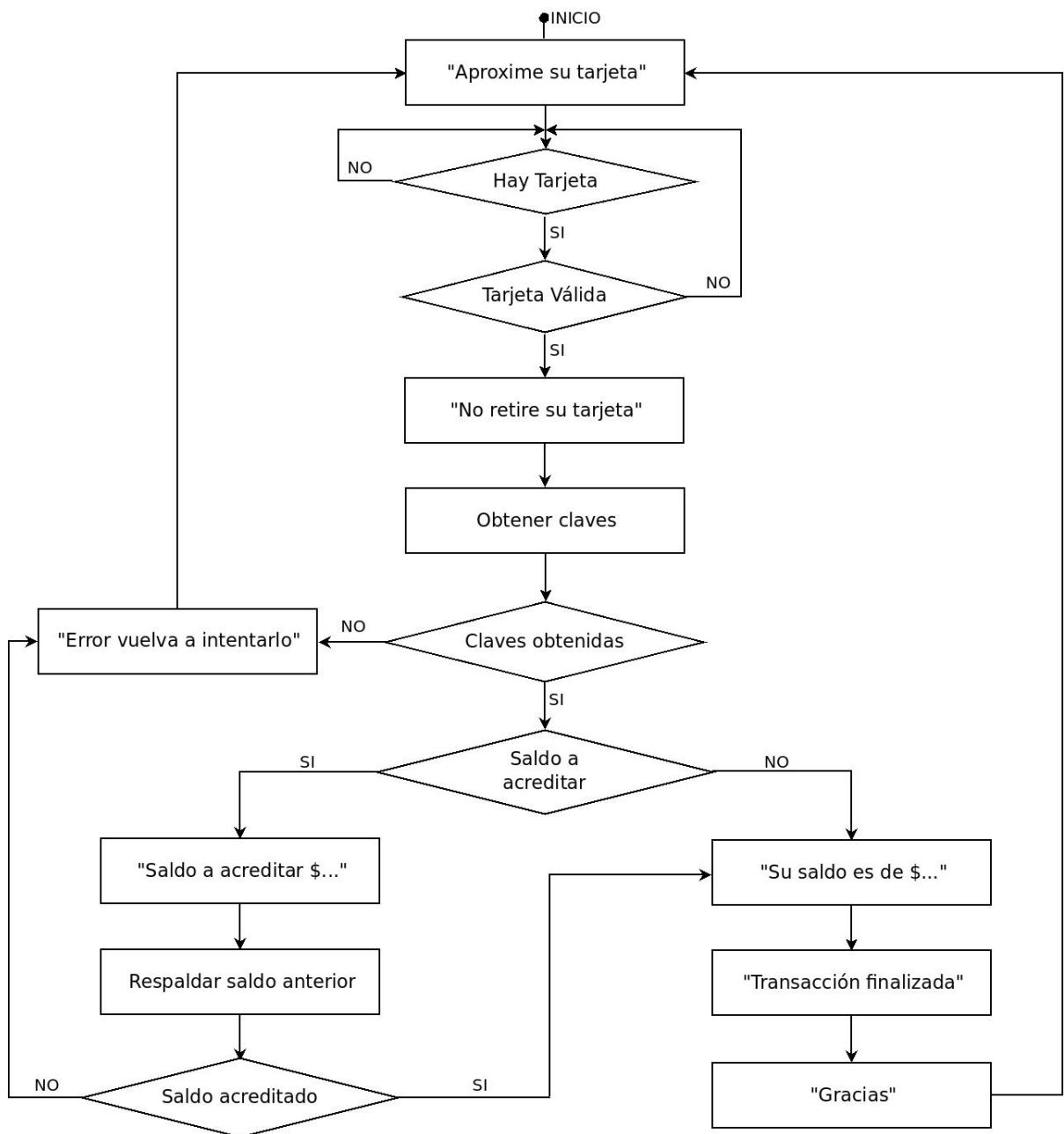


Figura 2.3: Diagrama de flujo

Capítulo 3

Hardware

3.1. Arquitecturas estudiadas

Se plantearon varias alternativas como posible solución. A medida que se encontraron limitantes o que no se cumplían los requerimientos exigidos, se fueron descartando dichas opciones.

A continuación se describen algunas de las arquitecturas consideradas:

- 1 - OpenPCD + lector de tarjetas de contacto + display + buzzer + leds

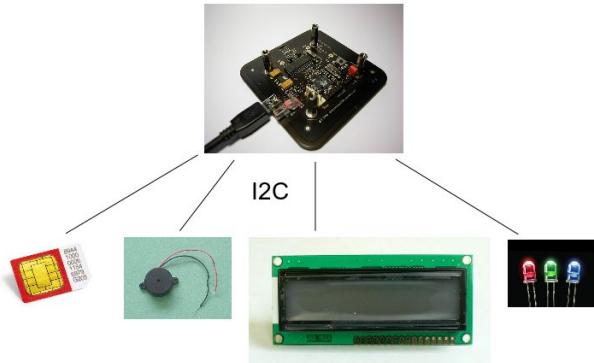


Figura 3.1: Solución considerada 1

Al dispositivo OpenPCD, se conecta el resto del hardware a través de su único puerto de entrada/salida disponible que es de tipo I2C.

- 2 - SBC + OpenPCD + microcontrolador + lector de tarjetas de contacto + display + buzzer + leds

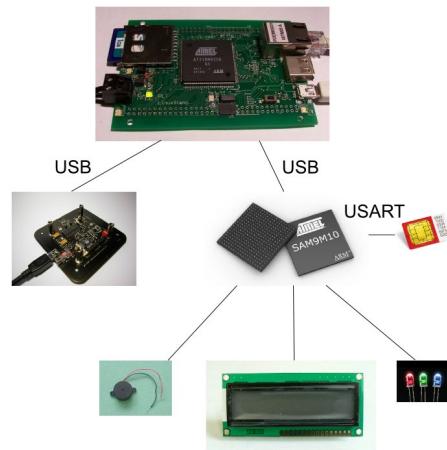


Figura 3.2: Solución considerada 2

Tanto el dispositivo OpenPCD como el microcontrolador se conectan directamente por USB a la SBC. El microcontrolador maneja el resto de los dispositivos (lector de tarjetas de contacto, display, buzzer y leds).

- 3 - SBC + OpenPCD + lector de tarjetas de contacto + display + buzzer + leds

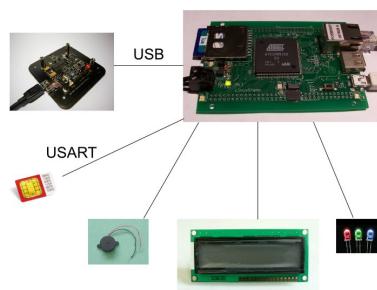


Figura 3.3: Solución considerada 3

El dispositivo OpenPCD se conecta por USB a la SBC. La SBC maneja los dispositivos (lector de tarjetas de contacto, display, buzzer y leds) a través de sus interfaces nativas.

- 4 - SBC + lector de tarjetas RFID + lector de tarjetas de contacto + display + buzzer + leds

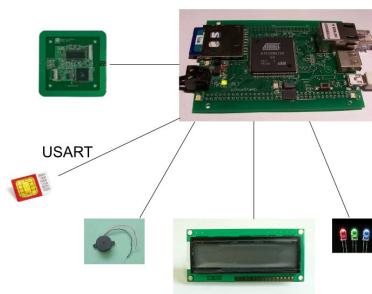


Figura 3.4: Solución considerada 4

Todos los periféricos se conectan a la SBC a través de sus interfaces nativas, esto incluye también el integrado CL RC632 de Philips [12] (ver hoja de datos en el apéndice E). Se debe diseñar la antena para propagar la señal RF hacia las tarjetas.

- 5 - microcontrolador + lector de tarjetas RFID + lector de tarjetas de contacto + display + buzzer + leds

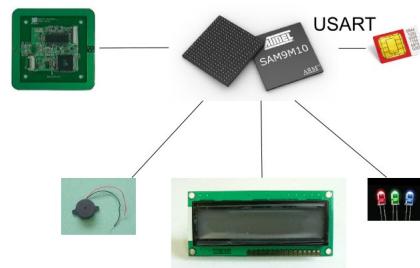


Figura 3.5: Solución considerada 5

Consta de un único PCB, que posee un microcontrolador como sistema central al cual se conectan el resto de los dispositivos. Dicho PCB tiene incorporada la antena para la propagación de RF.

3.2. Arquitectura seleccionada

En una primera instancia se pretendía utilizar únicamente el dispositivo OpenPCD (ver figura 3.1), ya que el mismo cuenta con un microcontrolador de la familia ARM, el AT91SAM7S128. Una vez estudiado se llegó a la conclusión de que no permitía la instalación de un sistema operativo Linux, ya que el mismo precisa más de 4MB de RAM para poder hacer algo útil. Otra desventaja encontrada fue que sólo tiene un puerto I2C como forma de conectar periféricos.

Surgió entonces la necesidad de usar una SBC como dispositivo capaz de ejecutar un sistema operativo y las aplicaciones necesarias para que el dispositivo cumpla con los requerimientos exigidos. El dispositivo OpenPCD pasaría entonces a cumplir la función de lector/escritor de tarjetas RFID (ver figuras 3.2 y 3.3), conectado a la SBC a través de su puerto USB, mientras que para el resto de los periféricos se diseñaría un PCB que fuera capaz de ser conectado a la SBC a través de sus interfaces nativas. Esta arquitectura fue descartada por el incremento en el costo del proyecto.

Fue necesario entonces descartar el uso del dispositivo OpenPCD y dar lugar a un diseño propio del lector/escritor de tarjetas RFID (ver figura 3.4), utilizando para esto el integrado CL RC632 de Philips.

La última opción y la más ambiciosa, plantea el diseño completo de un PCB (ver figura 3.5) contenido un microcontrolador y memoria capaz de ejecutar un sistema operativo, los lectores de tarjetas, tanto de contacto como RFID, y el resto de los periféricos (display, leds, buzzer). Esta opción fue dejada de lado por entender que excedería los plazos de tiempo del proyecto.

Se pensó entonces en diseñar la arquitectura 4 indicada en la figura 3.4, SBC + lector de tarjetas RFID + lector de tarjeta de contacto + display + buzzer + leds, y dado que se cuenta con un OpenPCD, la opción 3 mostrada en la figura 3.3, SBC + OpenPCD + lector de tarjetas de contacto + display + buzzer + leds, se dejaría como arquitectura alternativa si no se alcanzaran buenos resultados con el lector/escritor de tarjetas RFID.

Luego de estudiar ventajas y desventajas de las arquitecturas planteadas, se eligió la indicada en el ítem 4 en la sección 3.1, que es la que más se adaptó a los requerimientos necesarios:

- SBC + lector de tarjetas RFID + lector de tarjeta de contacto + display + buzzer + leds

En la figura 3.6 se muestra un diagrama de bloques correspondiente a la arquitectura seleccionada:

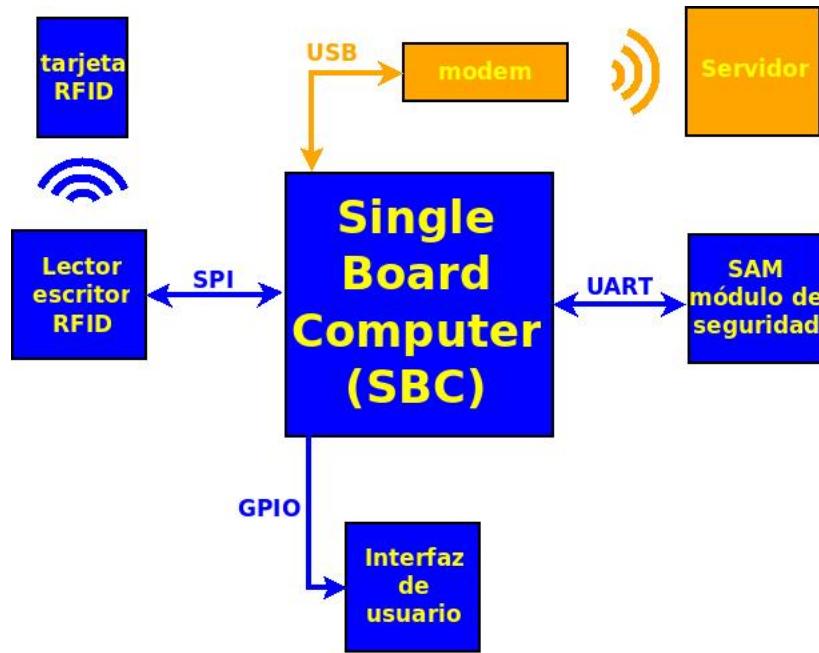


Figura 3.6: Diagrama de bloques de la arquitectura seleccionada

Los bloques oscuros serán implementados, no así los claros.

3.3. Elección de hardware

3.3.1. SBC

En primera instancia se confeccionó una lista con posibles candidatas de SBC disponibles en el mercado internacional, teniendo en cuenta factores como: precio, puertos de I/O, memoria RAM, memoria Flash, puertos USB, soporte para GNU/Linux, entre otros. Se definieron una serie de requisitos mínimos necesarios para seleccionar de la lista la SBC que más se adecuara a la arquitectura definida. Para la comunicación con el resto de los módulos será necesario: una interfaz UART para el módulo de seguridad (SAM); una interfaz SPI para el módulo lector/escritor RFID (CL RC632 de Philips); 20 GPIO para display, leds, buzzer, otros; 1 USB host para una posible conexión de un modem 3G (intercambio de datos con un servidor). En cuanto a la memoria disponible, tomando como referencia el AFE, debe ser de 32Mb de RAM y 8Mb de flash para un funcionamiento aceptable. Es conveniente, pensando a futuro, que el procesador trabaje a una frecuencia no menor a 200MHz. Dado el presupuesto estimado para el proyecto, el precio no debe superar los 150 dólares en origen. Como requisito adicional se exigió que existiera un foro actualizado y soporte técnico que permitiera evacuar dudas.

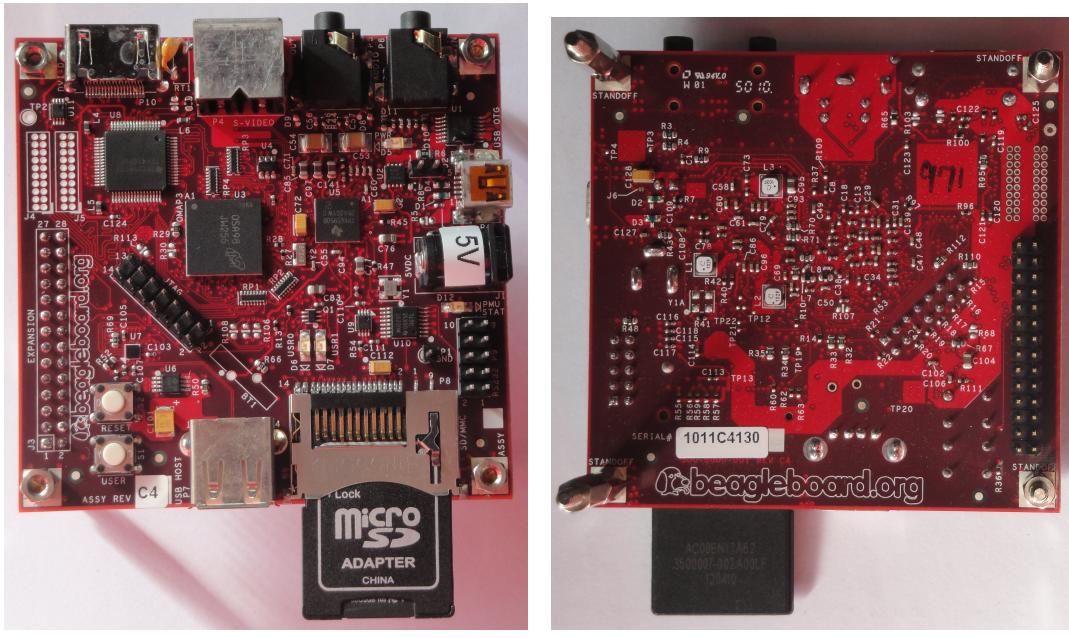
Aplicados los requisitos mínimos a la lista previamente confeccionada de SBC candidatas, se optó por dos: GESBC-9G20 [31] y Hawkboard [32]. En cuanto a la primera opción, GESBC-9G20, los fabricantes no respondieron consultas, por tanto se descartó. Se optó entonces por la segunda opción, Hawkboard, puesto que respondieron a las consultas en tiempos razonables y se logró evacuar dudas desde el foro.

Luego de comprar dos Hawkboard, ambas resultaron defectuosas a nivel de hardware, después de varios meses de pruebas sin resultados y sin respuestas concretas por parte del proveedor y fabricante y con la intención de cumplir con los plazos del proyecto, se optó por utilizar una SBC (Beagleboard [33]) que se consiguió en préstamo por medio del INCO. Esta SBC cumplió con los requisitos mínimos, aunque en ese momento tenía un costo del doble de la Hawkboard, teniéndose que diseñar un módulo hardware adicional. Finalmente, la SBC seleccionada para trabajar fue la Beagleboard rev.C4.

Las características generales de la BeagleBoard son: cuenta con un procesador OMAP3530 de 720MHz con arquitectura ARM. Posee memoria NAND-flash de 256Mb

y memoria ROM de igual tamaño. Tiene una ranura adicional para extender la memoria a través de una memoria SD. Entre otras cosas cuenta con un puerto USB OTG, un puerto USB host, un bloque de expansión de 28 pines (con señales a 1,8 Volts), puerto JTAG, conector RS232, etc.

En lo que respecta a la potencia disipada, la Beagleboard tiene un consumo de pico de 2W, y un consumo promedio de 560mW [35] [36].



(a) Vista anversa de la SBC

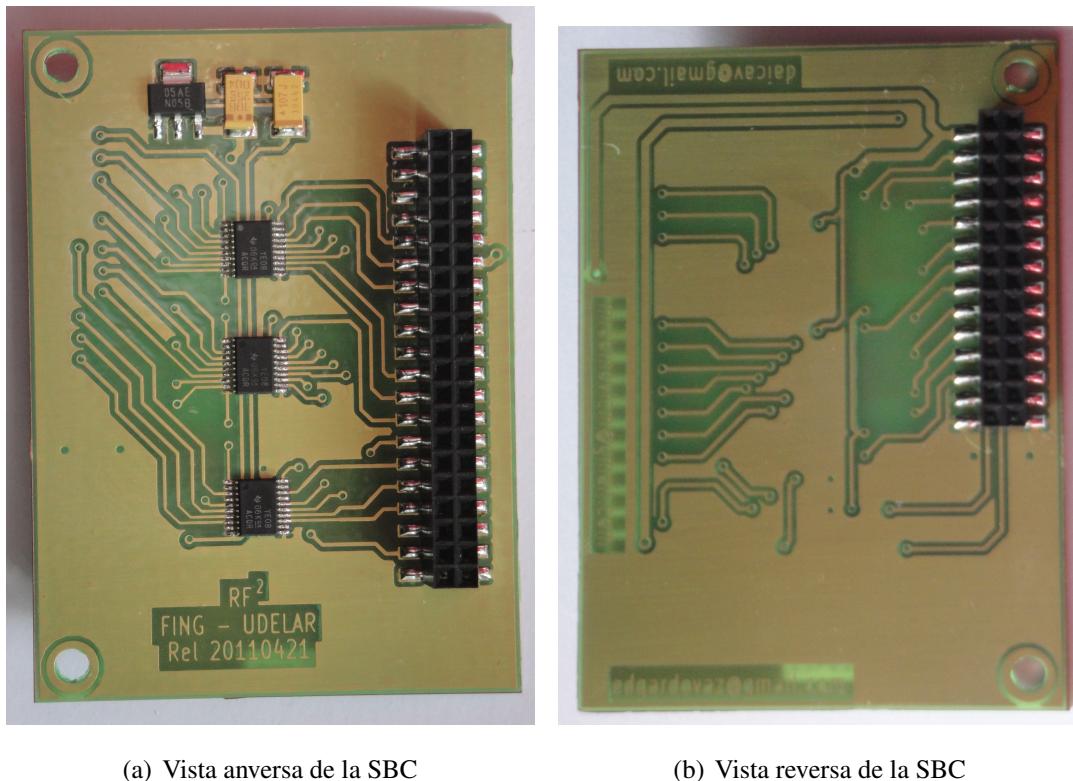
(b) Vista reversa de la SBC

Figura 3.7: Vista anversa y reversa de la SBC

3.3.2. VLT - Conversor de Voltajes

Este módulo no fue tenido en cuenta en la primera etapa del diseño de la arquitectura hardware, sino que surge como necesidad debido al cambio de SBC. Como consecuencia de lo anterior se vio la ventaja de incorporar una placa que permite la conexión entre la SBC y el resto del hardware, el cual puede permanecer inalterado por más que no ocurra lo mismo con la SBC, ya que ésta puede cambiar de versión o dejar de fabricarse en un breve lapso de tiempo. El único elemento a cambiar sería entonces la placa VLT, que es más simple y barata de fabricar que las restantes partes. La placa de circuito impreso VLT consta básicamente de dos conectores, uno de ellos permite la conexión

con la Beagleboard y el otro la conexión con el restante hardware el cual se encuentra intergrado en un PCB llamdo SCUI. Ambos conectores no se encuentran directamente interconectados entre sí a través de pistas, pues para el caso particular de Beagleboard fue necesario incorporar conversores de tensión que permitieran el traslado del nivel de tensión desde 1,8 Volts que usa esta SBC, a las tensiones con las que operan los periféricos, ya sea 3,3 o 5 Volts. El último elemento, no menos importante, es un regulador de tensión LDO que permite generar 3,3 Volts a partir de la fuente de tensión de 5 Volts de la propia Beagleboard.



(a) Vista anversa de la SBC

(b) Vista reversa de la SBC

Figura 3.8: Vista anversa y reversa de VLT

3.3.3. SCUI - Lector de tarjetas de contacto e Interfaz de Usuario

El módulo SCUI puede dividirse en dos partes, una de ellas es un lector de tarjetas de contacto basadas en la norma ISO7816, y la otra es una simple interfaz para el usuario. El lector de tarjetas de contacto (smart cards), está compuesto por un conversor full duplex a half duplex el cual se encuentra conectado a uno de los puertos UART de la SBC a

través del módulo VLT, que se describió en el punto anterior. Este conversor permite la transmisión de datos directamente entre la tarjeta y la SBC, sin necesidad de intercalar un ASIC para el manejo de tarjetas del tipo ISO7816. Cuenta también con un oscilador para alimentar la entrada de reloj de las tarjetas. La entrada de control (OE) del oscilador operada desde la SBC permite poner la salida de reloj en tercer estado, cosa muy útil a la hora de cumplir con la secuencia de inicialización de las tarjetas descritas en el estándar. El lector permite operar con tarjetas clase A (alimentadas a 5 Volts) y clase B (alimentadas a 3,3 Volts) haciendo uso de un jumper que permite intercambiar la tensión de alimentación suministrada a la tarjeta. Se cuenta con un zócalo para insertar la tarjeta de contacto. Por otra parte, la interfaz de usuario está compuesta por tres leds (verde, amarillo y rojo), buzzer y un display LCD16x2 donde son desplegados los mensajes que indican al usuario la operación que se efectúa sobre su tarjeta Mifare. El último elemento a describir aquí es un conector receptáculo 5x2 (100mils) en el que se conecta el módulo lector/escritor RFID que opera con las tarjetas RFID Mifare.

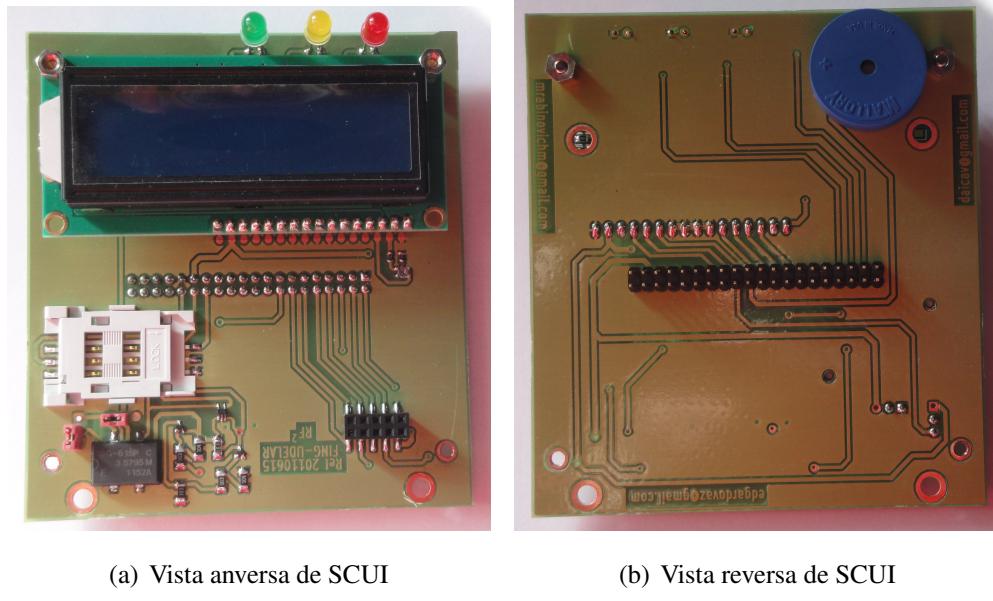


Figura 3.9: Vista anversa y reversa de SCUI

3.3.4. Lector/Escritor RFID

Este módulo es el encargado de la comunicación con las tarjetas RFID que cumplen con la norma ISO14443. Consta básicamente de cuatro secciones entre las que se encuentran: el integrado CL RC632; el filtro EMC, el circuito de adaptación de impedancia (matching); y el inductor de la antena. El ASIC CL RC632 permite, por un lado la comunicación digital con un microprocesador a través de su puerto de datos y por el otro lado la transmisión de datos hacia la antena que emitirá la señal RF para la comunicación con las tarjetas ISO14443. Lo que se llama propiamente antena RF está conformada por el circuito de adaptación de impedancia (matching) y por el inductor, que propaga el campo magnético para lograr el acoplamiento necesario entre lector y tarjeta, de aquí la sigla PCD (Proximity Coupling Device).

Los principios básicos de funcionamiento de la antena se detallan en el apéndice B.



(a) Vista anversa del lector/escritor RFID

(b) Vista reversa del lector/escritor RFID

Figura 3.10: Vista anversa y reversa del lector/escritor RFID

Por detalles de esquemáticos y listas de componentes referirse al apéndice C.

3.4. Funcionamiento de módulos

3.4.1. SBC

La SBC está formada por un SOC y memoria suficiente para ejecutar un sistema operativo Linux orientado a desarrollar sistemas embebidos. Sobre el sistema operativo se instalan los módulos y bibliotecas necesarias para hacer uso del hardware que contiene la SBC. En la aplicación se utilizará uno de sus puertos SPI para la comunicación con el lector/escritor de tarjetas RF, un puerto UART para la comunicación de datos con el lector de tarjetas de contacto y varias salidas GPIO para el control de la interfaz de usuario.

3.4.2. VLT - Conversor de Voltajes

El corazón de esta placa son los integrados TXB0108 [13] (ver hoja de datos en el apéndice E) que permiten la interconexión de dispositivos que operan en distintos niveles de tensión. Básicamente el integrado está constituido por dos puertos, puerto A y puerto B cada uno de 8 bits. El puerto A opera con la tensión de 1,8 Volts que permite ser conectado a la Beagleboard, el puerto B opera con la tensión de 3,3 Volts cuando se encuentra conectado al CL RC632, y de 5 Volts para los restantes periféricos. Cada I/O de un puerto es sensible a los flancos de subida o bajada, trasladando estos cambios a la I/O correspondiente del puerto opuesto. Este integrado posee también una entrada de control para poner los puertos en estado de alta impedancia. Una ventaja es que no poseen entrada de control de dirección de flujo de datos, de modo que se ahorran pines de control que no se tienen disponibles en la Beagleboard. En la figura 3.11 se puede observar como están constituidas cada una de las entradas/salidas del integrado. Otra pieza que compone esta placa es el regulador de tensión LDO implementado a partir del integrado LM1117 [15] (ver hoja de datos en el apéndice E), éste se utiliza para convertir la entrada de tensión de 5 Volts en una salida de tensión de 3,3 Volts y así poder alimentar el periférico correspondiente.

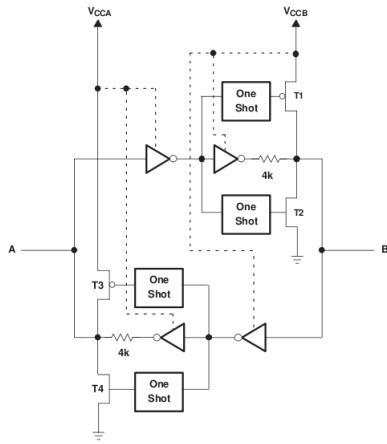


Figura 3.11: Arquitectura de una celda I/O del TXB0108

3.4.3. SCUI - Lector de tarjetas de contacto e Interfaz de Usuario

Lector de tarjetas de contacto ISO7816

Es un lector muy simple de implementar, su construcción se basa en un conversor full a half duplex construido a partir de un circuito transistorizado trabajando en zona de corte y saturación. Los transistores empleados son el NPN 2N3904 (ver hoja de datos en el apéndice E) y el PNP 2N3906 (ver hoja de datos en el apéndice E) los cuales fueron seleccionados en base a su rápida característica de commutación que es del orden de algunas decenas de nanosegundos. Dada la característica del circuito, es posible recibir el eco de la transmisión de datos generados por la SBC. Un elemento fundamental que compone el circuito del lector es el oscilador de frecuencia 3,579545 Mhz, este valor no es antojadizo sino que permite generar la base de tiempo adecuada para la transmisión de datos entre la tarjeta y la SBC. Otras frecuencias de reloj fueron empleadas, como ser 4 Mhz y 5 Mhz, con resultados inciertos en la recepción de los datos, aún cuando sería posible usar estos valores según la referencia [18] para los parámetros obtenidos desde el ATR de la tarjeta. El circuito cuenta también con protección de descarga ESDA6V1W5 (ver hoja de datos en el apéndice E) para los contactos de la smart card.

Interfaz de usuario

El elemento a destacar es un display LCD16x2 que basa su funcionamiento en el controlador Hitachi HD44780 [14] (ver hoja de datos en el apéndice E). La transferencia de datos hacia el display se hace a través de un puerto con 4/8 bits de datos y 3 bits de control. Debido a que no se cuenta con la cantidad de pines disponibles en la Beagleboard para operar en el modo de 8 bits, se empleó en su lugar el modo 4 bits del display. El bit de control RS indica si el byte a enviar por el puerto de datos es una palabra de control o un carácter ASCII a ser almacenado en la memoria interna del display. El bit R/W por su parte indica si se efectuará una lectura o una escritura de la memoria interna del display. Por último en el bit E se indica mediante flanco de bajada que se ejecute la operación indicada con los anteriores dos bits de control, previo a este flanco las señales en el puerto de datos deben permanecer fijas. El display cuenta también con una entrada para calibrar el contraste del LCD, la calibración se realiza a partir de un divisor resistivo implementado con resistencias y un preset. El backlight del display es accionado desde uno de los pines de la SBC a partir de un circuito transistorizado que opera en zona de corte/saturación. Los restantes elementos que componen la interfaz de usuario son leds y buzzer que son accionados directamente desde los pines del puerto de expansión de la SBC.

3.4.4. Lector/Escritor RFID

En el corazón del lector/escritor de tarjetas RFID, se encuentra el chip CL RC632 que forma parte de una familia de integrados empleados para la comunicación con tarjetas sin contacto, pertenecientes a la norma ISO14443 las cuales operan a la frecuencia 13,56 Mhz. El CL RC632 soporta todas las capas del esquema de comunicación que se establecen en la mencionada norma, incluyendo el algoritmo de seguridad (CRYPTO1) para autenticar las tarjetas Mifare Classic. En lo que sigue se describen algunas de las características principales del integrado.

Interfaz

Los comandos, bits de configuración y las banderas se acceden a través de la interfaz con un microprocesador. El puerto elegido para la comunicación desde la SBC es el SPI, aunque es posible la comunicación a través de su puerto paralelo.

Registros

La configuración del chip se lleva a cabo a partir de un mapa de registros de control que se encuentra dividido en 8 páginas con 8 registros cada una. La manera de alcanzar estos registros es mediante el intercambio de página, mecanismo que puede ser deshabilitado mediante escritura de un “1” en el bit 7 del registro 0 en la página 0, logrando direccionamiento plano. La función de cada uno de sus registros puede ser observada en la hoja de datos del integrado [12] (ver apéndice E).

Memoria EEPROM

La memoria está dividida en 32 bloques con 16 bytes cada uno. El contenido de memoria EEPROM en los bloques 1 y 2 (dirección 10hex a 2Fhex) se utilizan para configurar los registros del CL RC632 durante la fase de inicialización, de forma automática. La configuración por defecto soporta la comunicación Mifare ISO 14443 A, aunque los usuarios pueden especificar la inicialización para I-Code1, ISO 15693 o ISO 14443 B, mediante los bloques de memoria 3 al 7. Se reservan 384 bytes para almacenar las claves CRYPTO1 que son usadas para la autenticación con las tarjetas. El formato de una de estas claves puede verse en [12] (ver apéndice E) y tiene una longitud de 12 bytes, por tanto es posible almacenar en memoria las 32 claves que posee una tarjeta.

Buffer FIFO

El integrado contiene un buffer FIFO de 64 bytes para flujo de datos con un microprocesador. La entrada y salida del buffer de datos está conectado con el registro FIFOData. Escribir en este registro almacena un byte en el buffer e incrementa el puntero de escritura del buffer. La lectura de este registro muestra el contenido del buffer e incrementa el puntero de lectura. La distancia entre el puntero de escritura y lectura se puede obtener mediante la lectura del registro FIFOLength, indicando así la cantidad de bytes que se llevan almacenados. Es posible observar y controlar el estado del buffer mediante varios registros, para evitar que se produzcan errores de comunicación con el microprocesador.

Interrupciones

El CL RC632 indica ciertos eventos estableciendo el bit IRQ en el registro PrimaryStatus, y además, por la activación del pin IRQ. La señal en el pin IRQ se puede utilizar para interrumpir un microprocesador. Las posibles fuentes de interrupción son:

- Timer, a través de su bandera TimerIRq
- Transmisor, coprocesador CRC y memoria E2PROM, a través de su bandera TxIRq
- Receptor, a través de su bandera RxIRq
- Registro de comando, a través de su bandera IdleIRq
- Buffer FIFO, a través de sus banderas HiAlertIRq y LoAlertIRq

El CL RC632 informa al microprocesador sobre el origen de una interrupción mediante el establecimiento del bit adecuado en el registro InterruptRq. La relevancia de cada bit de petición de interrupción como fuente de una interrupción puede ser enmascarada con el bit de habilitación de interrupciones en el registro InterruptEn. Si alguna bandera de solicitud de interrupción se establece en 1 (una solicitud de interrupción está pendiente) y la correspondiente bandera de habilitación de interrupción está en "1", la bandera de estado IRq en el registro PrimaryStatus se establece en 1. Por otra parte diferentes fuentes de interrupción pueden estar activas al mismo tiempo. Por lo tanto, se hace un OR con todos los bits de solicitud de interrupción, el resultado se envía a la bandera IRq y se conecta al pin IRQ. Los bits de petición de interrupciones están seteados de forma automática por las máquinas de estado internas del CL RC632. Adicionalmente, el microprocesador tiene acceso para setearlos o borrarlos. Una implementación especial de los registros InterruptRQ y InterruptEn permiten el cambio de un único bit de estado sin tocar el resto.

Configuración del Pin IRQ: El nivel lógico de la bandera de estado IRq es visible por el pin IRQ. Además, la señal en el pin puede ser controlada por los siguientes bits del registro IRQPinConfig

- IRQInv: Si este bit es 0, la señal en el pin IRQ es igual al nivel lógico del bit IRq. Si es 1, la señal en el pin IRQ está invertida con respecto al bit IRq.
- IRQPushPull: Si este bit es 1, el pin IRQ tiene características de una salida estándar CMOS, de otra manera la salida es open drain y un resistor externo es necesario para alcanzar un nivel alto en este pin.

Para poder hacer uso de lo descrito anteriormente se previó y reservó una entrada en el conector de expansión de la Beagleboard (ver figura C.1), sin embargo el software empleado no hace uso del mecanismo de interrupciones sino que opera mediante polling.

Transmisor, pines Tx1 y Tx2

La señal en Tx1 y Tx2 es la portadora, centrada en 13,56 Mhz, modulada ASK 100 % con los datos a transmitir. Estos pines son conectados directamente a la antena para propagar la señal RF hacia las tarjetas RFID. La distancia de operación alcanzada es de hasta 10cm de longitud, dependiendo de la geometría de la antena, así como también adaptación de impedancia lograda, entre otros (ver [4] y [5] incluídas en el apéndice E). Algunos registros del integrado permiten la configuración del transmisor, posibilitando entre otras cosas apagar la señal portadora en caso de ser necesario.

Conjunto de comandos

El CL RC632 opera como una máquina de estado capaz de interpretar y ejecutar un conjunto de comandos pre establecidos. La ejecución de uno de ellos es posible escribiendo su código correspondiente en el “Registro de Comandos”, si fuera necesario el pasaje de parámetros, éstos se colocarán en el buffer FIFO mencionado antes. Una lista detallada de comandos junto con los parámetros necesarios es mostrada en la hoja de datos (ver apéndice E), entre ellos se pueden destacar los siguientes: Authent, Transceive, LoadKey.

Antena RF

En lo que sigue se describen algunas de las partes que integran la antena RF que se conecta directamente a los pines Tx1 y Tx2 del integrado descrito antes.

Filtro EMC

La frecuencia de la portadora de la señal transmitida se centra en 13,56 Mhz, sin embargo se generan también armónicos de mayor frecuencia. Para cumplir con la regulación internacional EMC es que se agrega este filtro pasa bajos, cuya frecuencia de corte debe ubicarse en 14,4 Mhz, o sea 13,56 Mhz más 847,5 KHz para permitir el ancho de banda necesario que logre el baud rate requerido en la transmisión de los bits. En síntesis el filtro ayuda a mejorar la relación señal a ruido para la señal recibida y decrementa el sobretiro en los pulsos transmitidos mejorando la calidad de la señal

transmitida. Los valores propuestos para los componentes de este filtro se encuentran en las notas de aplicación [4].

Matching

Por su parte, el circuito de adaptación de impedancia permite que la antena resuene a la frecuencia deseada, en este caso 13,56 Mhz. Los valores de los elementos que conforman este circuito deben ser estimados y sintonizados a partir del diseño del inductor de la antena. El factor de calidad total de la antena debe ser tenido en cuenta para cumplir con los requerimientos establecidos en la norma ISO14443. El mecanismo para el cálculo de los elementos que foman este circuito se detallan en las notas de aplicación [4].

Inductor

El inductor de la antena es quien propaga el campo magnético para la transmisión de datos hacia las tarjetas. El diseño de la antena comienza a partir de este elemento. El cálculo detallado del valor del inductor se encuentra en las notas de aplicación [1], aunque su costo y tiempo en la práctica son considerables; una estimación del valor de la inductancia puede verse en el apéndice B, en el que se deben tener en cuenta los siguientes elementos: geometría de la antena, ancho y espesor del conductor del PCB, longitud de una espira, número de vueltas, etc.

Receptor

El circuito receptor de la antena se encuentra bien detallado en las notas de aplicación [4] y no fue necesario efectuar ningún cambio para lograr buenos resultados en este diseño particular.

Capítulo 4

Software

4.1. Introducción

Se debe destacar que todo el desarrollo de software se basó exclusivamente en herramientas de software libre. La distribución Linux elegida para el sistema embebido se llama Angström [41]. Esta distribución es muy usada en aplicaciones que usan una Beagleboard y cuenta con una gran cantidad de bibliotecas implementadas en lenguaje C, que permiten gran escalabilidad a la hora de incorporar nuevos periféricos en la aplicación.

4.2. Arquitectura de Software

4.2.1. Descripción

Un sistema Linux se compone de diferentes partes que interactúan entre sí, formando capas ordenadas con distintos grados de abstracción respecto al hardware. Ésto se puede apreciar en la figura 4.1 donde se muestra a grandes rasgos el sistema.

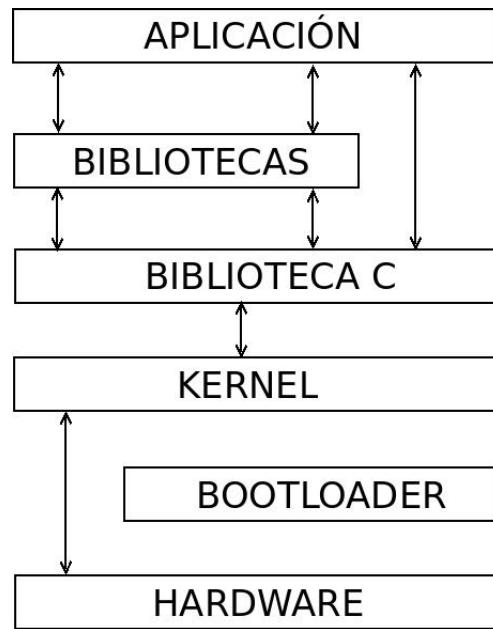


Figura 4.1: Sistema Linux

El bootloader es la parte del sistema más primitiva y su función es la de cargar el kernel en memoria RAM para su ejecución. En general el bootloader es una aplicación que se divide en dos etapas, la primera etapa es fuertemente dependiente del CPU con que cuenta la placa, y su función es buscar en particiones activas para luego cargar en memoria RAM la segunda etapa del bootloader. Esta segunda etapa se encarga de descomprimir en memoria RAM la imagen comprimida del kernel, para luego ser ejecutado y que éste tome el control del sistema. El kernel se encarga a grandes rasgos de habilitar interrupciones, configurar la memoria, y montar un sistema de archivos primitivo que permite a su vez cargar los módulos necesarios para la interfaz con periféricos. Luego se monta el verdadero sistema de archivos (fileSystem). En este nuevo sistema de archivos es donde se instalarán diferentes programas y bibliotecas para la correcta ejecución de aplicaciones. En funcionamiento, toda la comunicación con periféricos se realiza a través del kernel que es la parte más cercana al hardware. Cada vez que se ejecuta una aplicación, ésta hace uso de las bibliotecas para poder comunicarse con el kernel, y éste se encarga de la comunicación con los periféricos. Las bibliotecas pueden ser nativas como es el caso de la biblioteca de lenguaje C, o desarrolladas para que determinada aplicación funcione correctamente.

Una referencia interesante para entender el proceso de arranque es [45].

4.2.2. Sistema Operativo

En el arranque, la Beagleboard tiene la posibilidad de buscar el bootloader y el kernel en NAND, o en dispositivos extraíbles tales como memorias USB o memorias SD. Para el sistema RF², se eligió un arranque a través de una memoria SD ya que es fácil de manipular.

En la figura 4.2 se puede ver como queda distribuida la memoria SD con las distintas partes que conforman el sistema operativo.

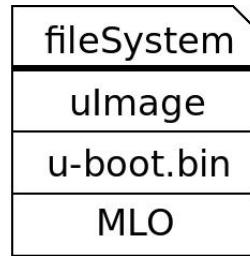


Figura 4.2: Memoria SD para funcionar en Beagleboard

En la memoria SD se pueden distinguir dos particiones, una en formato FAT32 y otra en formato ext3. La partición en FAT32 es llamada “de arranque”, y es donde se encuentra el bootloader (MLO, u-boot.bin) y la imagen comprimida del kernel (uImage). La partición en ext3 es donde se encuentra el sistema de archivos (fileSystem).

El MLO es el equivalente al bootloader de la primera etapa, en general ya viene pre-cargado en la memoria NAND de la Beagleboard. Es posible generararlo, o incluso bajar una versión ya compilada desde la web de Angström [41]. Como característica principal tiene la capacidad de buscar el u-boot.bin en dispositivos extraíbles como memorias SD o USB.

El u-boot.bin es equivalente al bootloader de la segunda etapa. En el sistema RF² fue necesario generararlo, ya que tiene la posibilidad de configurar el bloque de expansión de la Beagleboard.

El uImage es el kernel del sistema. Fue necesario generararlo, ya que se debieron modificar sus fuentes para que queden habilitadas las interfaces de comunicación con

los dispositivos periféricos.

El fileSystem (también conocido como RootFileSystem) es el correspondiente a una distribución Linux llamada Angström. Se pueden llegar a precargar distintos programas y bibliotecas dependiendo de la forma en que se genere. Angström es una distribución Linux diseñada específicamente para sistemas embebidos desarrollados para SBCs como la usada para este prototipo. Ésto lo hace más eficiente que otros sistemas operativos para la aplicación desarrollada. La elección de esta distribución se debió a que es de los más recomendados y utilizados en la documentación y foros de Beagleboard [34].

4.3. Herramientas utilizadas en el desarrollo del sistema

4.3.1. Introducción

Para el desarrollo de sistemas existe una gran variedad de herramientas útiles, algunas de software libre y otras privativas. El hecho de tener tantas opciones disponibles, a pesar de ser una ventaja, a veces dificulta la elección de las herramientas correctas.

Para la elección de las herramientas se tomó como primer criterio de decisión el hecho que sean libres, así como las experiencias de otras personas que ya han transitado caminos comunes, consultando y participando en foros activos.

Cabe señalar que en el PC de desarrollo se utilizó Ubuntu 10.10.

A continuación se detallan las herramientas utilizadas para el desarrollo del sistema.

4.3.2. MLO, u-boot.bin y uImage

No fue necesario generar el MLO debido a su simpleza, puesto que el binario pre-compilado realiza bien su función.

El u-boot.bin y el uImage fueron generados con la herramienta de desarrollo y compilación OpenEmbedded-Bitbake [37] que es una fusión de dos herramientas: OpenEmbedded, herramienta para construcción y mantenimiento de distribuciones, y Bitbake, herramienta de compilación similar al Make [46] que automatiza la construcción de ejecutables entre otros. Esto es, OpenEmbedded utiliza Bitbake para su objetivo.

OpenEmbedded-Bitbake es una herramienta muy potente y difícil de aprender al principio. Luego de entendido su principio de funcionamiento se hace muy simple su uso, para lo que es necesario tener acceso a una buena conexión a internet debido a que realiza descargas de paquetes en forma habitual. Con esta herramienta también se pueden generar el MLO y el fileSystem, aunque se prefirió utilizar otras herramientas por sobre ésta. Su instalación, configuración, estructura y uso se pueden ver en el apéndice D.3.

4.3.3. FileSystem

Para la generación del fileSystem de Angström, se utilizó la herramienta web Narcissus [43] disponible en la página de Angström. Esta herramienta permite seleccionar entre diferentes dispositivos (entre los cuales está Beagleboard), los programas que se quieran instalar, el formato de la imagen seleccionada, e incluso se puede generar un kit de desarrollo (SDK) para el PC de desarrollo. Debido a la facilidad de uso y a los buenos resultados obtenidos, se decidió utilizar esta opción por sobre la del fileSystem generado por la herramienta OpenEmbedded-Bitbake.

4.3.4. Croscompilación

Se llama croscompilar al proceso de compilación de una arquitectura a otra. En el caso del proyecto RF², se compilan paquetes en una arquitectura x86 (PC de desarrollo) para ser utilizados en una arquitectura ARM.

Para la croscompilación se utilizó el SDK generado por Narcissus y la herramienta Make para generar los archivos necesarios. La instalación, configuración y uso del SDK se encuentran en el apéndice D.2.

4.3.5. Depuración de código

Para la depuración, se utilizó la herramienta GDB del proyecto GNU. Al momento de compilar, es necesario agregar la opción -g para que la aplicación pueda ser depurada. Esta opción agrega información en el código de la aplicación. La interfaz del GDB es por consola, aunque existen algunos programas que utilizan GDB y además ofrecen una interfaz gráfica (DDD [47]).

Algunos de los comandos útiles y sus usos más comunes pueden encontrarse en el apéndice D.6.

Sin olvidar el hecho que la aplicación RF² está diseñada para una arquitectura distinta a la del PC de desarrollo, para el depurado de la aplicación existen dos alternativas.

La primera opción es lo que se podría llamar depuración local, esto es, instalar GDB en la Beagleboard y depurar la aplicación en ésta. Para saber lo que sucede es necesario acceder de forma remota a la Beagleboard desde el PC de desarrollo.

La segunda opción es la depuración remota. La depuración remota consiste en realizar la depuración de la aplicación desde el PC de desarrollo. Para esto, es necesario instalar GDBServer en la Beagleboard y tener instalado el GDB específico de la Beagleboard en el PC de desarrollo. Luego se establece una conexión, serial o ethernet, entre la Beagleboard y el PC de desarrollo. Por más detalles sobre la configuración referirse al apéndice D.7.

La primera opción no es posible para sistemas embebidos chicos en los cuales no se puede instalar GDB, aunque éste no es el caso de la Beagleboard. Se tienen más y mejores herramientas en el PC de desarrollo, por ejemplo programas con interfaz gráfica que ayudan a entender mejor lo que está pasando. Por algunas de estas razones, se prefiere el uso de la depuración remota.

Se utilizaron indistintamente tanto la primera opción como la segunda.

4.3.6. Bibliotecas

- pscsc-scan es una herramienta de uso por línea de comandos, que una vez ejecutado imprime el nombre del lector PC/SC conectado y en caso que exista una tarjeta de contacto insertada imprime también su ATR. Luego escanea el lector constantemente e indica si se retira o inserta una tarjeta. Esta aplicación fue utilizada para testear el lector de tarjetas de contacto.
- librfid-tool es una herramienta de uso por línea de comandos que da acceso de bajo nivel RFID utilizando los lectores soportados por la biblioteca para manejo de lectores/escritores RFID, librfid. Esta herramienta fue utilizada para testear lectores/escritores de tarjetas RFID.

A continuación se detallan la forma de llamar a la aplicación y algunas de las opciones soportadas. Para la instalación y uso referirse al apéndice D.5

4.3.7. Otras herramientas

subversion y Git, para control de versiones.

Geany para edición y compilación de código.

minicom para comunicación serial.

strace es un comando Linux que lista las llamadas al sistema, útil para la depuración.

4.4. Desarrollo

4.4.1. MLO

Como se mencionó anteriormente, no fue necesario generar el MLO debido a que la Beagleboard viene con uno pre-instalado, y en caso de que no funcione en forma correcta se puede descargar desde un repositorio de Angström [42]. No fue necesario realizar ningún cambio en el archivo ya que realiza su función correctamente.

4.4.2. Multiplexado de pines

El microprocesador OMAP3530 tiene muchos pines con distintas interfaces entre las que se cuentan puertos UART, SPI, GPIO, etc., pero no todos son accesibles desde la Beagleboard. Para poder acceder a algunos de estos puertos del microprocesador, existe en la placa de la Beagleboard un bloque de expansión de 28 pines.

Por defecto, en el bloque de expansión no se encuentran las señales que se quieren. Esto lleva a que se tenga que modificar el estado inicial de los pines. Existen dos formas de modificar los pines de modo de tener las señales que se precisan. Una de ellas es modificar el bootloader y la otra es modificar el kernel. Esto implica cambios en los archivos fuentes y posterior compilación que genere los nuevos binarios u-boot o uImage.

Para la modificación de las señales disponibles en el bloque de expansión se decidió modificar el u-boot, ya que la modificación por u-boot es más intuitiva y por experiencia se sabe que lo que más se actualiza y/o modifica es el kernel.

4.4.3. u-boot

Como se mencionó anteriormente en el u-boot se realizó la configuración de los pines del bloque de expansión de la Beagleboard. Cada pin del bloque de expansión tiene varias funcionalidades asociadas, y la configuración de la funcionalidad depende de un multiplexado modificable a nivel de software. Esto es, dependiendo del “modo de pin” elegido, la función que se obtiene en dicho pin. Para que los cambios hechos en el u-boot tengan el efecto esperado al arrancar el sistema, es necesario que en la configuración del kernel esté la opción CONFIG_OMAP_MUX=no, lo que imposibilita al kernel a realizar este mismo cambio. Esta opción no está activada por defecto para versiones de kernel 2.6.32 y posteriores.

Antes que pueda ser modificado el estado de los pines del bloque de expansión, es necesario obtener los archivos fuente con los cuales se genera el archivo binario u-boot.

Nota: Puede que cuando se realiza la instalación de OpenEmbedded-Bitbake, se genere un directorio relacionado con u-boot en /stuff/build/tmp/work/beagleboard-angstrom-linux-gnueabi/, si esto es así, no es necesario volver a obtener los fuentes.

Se configura el bitbake para poder utilizarlo:

```
$ export BBPATH=/stuff/build:/stuff/openembedded  
$ export PATH=/stuff/bitbake/bin:$PATH
```

Se obtienen los fuentes:

```
$ cd /stuff/build  
$ bitbake -f -c clean -b ../openembedded/recipes/u-boot/u-boot_git.bb  
$ bitbake -f -c compile -b ../openembedded/recipes/u-boot/u-boot_git.bb
```

Los fuentes se encuentran en

/stuff/build/tmp/work/beagleboard-angstrom-linux-gnueabi/u-boot.../git/.

En los fuentes del u-boot dentro de board/ti/beagle/ se encuentra el archivo beagle.h que es donde se establece la configuración de los pines del bloque de expansión de la Beagleboard.

Si se abre este archivo se ven líneas del estilo:

```
MUX_VAL(CP(MCBSP3_DX), (IEN | PTD | DIS | M4)) /*GPIO_140*/\
```

MUX_VAL indica que se va a modificar el valor de multiplexado de lo que está entre paréntesis.

CP(MCBSP3_DX) es el Control_PadConf, esto es el registro del microprocesador asociado con el pin a modificar.

(IEN | PTD | DIS | M4) es la configuración del pin en cuestión:

La opción IEN (input enable) hace que el pin sea bidireccional.

La opción PTD y PTU, indica si el pin tiene un pull down o pull up respectivamente.

La opción DIS y EN, indica si se deshabilitan o no las opciones PTD y PTU respectivamente.

La opción M4 es el modo seleccionado para el pin. Para la Beagleboard existen cuatro modos: M1, M2, M3 y M4.

GPIO_140 es el nombre del pin (solo es un comentario).

El Control_PadConf es un registro de 32bits que controla el estado de dos pines, esto es, la parte baja del registro controla un pin y la parte alta controla otro.

Analizando el “manual de referencia Beagleboard del usuario” (“Expansion connector signals” – tabla 20), y “manual técnico de referencia OMAP35x” (“SCM functional description” – capítulo 7.4.4); ambos adjuntos en el apéndice E; y agregando las opciones que interesan para los pines, se confeccionó la siguiente tabla:

CONTROL_PADCONF	EQUIV. BEAGLE.H	PIN EXPANSIÓN	SEÑAL EN EL PIN	MODO
MMC2_DAT6[31:16]	MMC2_DAT7	3	GPIO_139	4
UART2_CTS[15:0]	UART2_CTS	4	GPIO_144	4
MMC2_DAT6[15:0]	MMC2_DAT6	5	GPIO_138	4
UART2_TX[15:0]	UART2_TX	6	UART2_TX	0
MMC2_DAT4[31:16]	MMC2_DAT5	7	GPIO_137	4
McBSP3_CLKX[31:16]	McBSP3_FSX	8	UART2_RX	1
MMC2_DAT4[15:0]	MMC2_DAT4	9	GPIO_136	4
UART2_CTS[31:16]	UART2_RTS	10	GPIO_145	4
MMC2_DAT2[31:16]	MMC2_DAT3	11	McSPI3_CS0	1
McBSP1_DX[15:0]	McBSP1_DX	12	GPIO_158	4
MMC2_DAT2[15:0]	MMC2_DAT2	13	GPIO_134	4
McBSP1_CLKX[15:0]	McBSP1_CLKX	14	GPIO_162	4
MMC2_DAT0[31:16]	MMC2_DAT1	15	GPIO_133	4
McBSP_CLKS[31:16]	McBSP1_FSX	16	GPIO_161	4
MMC2_DAT0[15:0]	MMC2_DAT0	17	McSPI3_SOMI	1
McBSP1_DX[31:16]	McBSP1_DR	18	GPIO_159	4
MMC2_CLK[31:16]	MMC2_CMD	19	McSPI3_SIMO	1
McBSP1_CLKR[15:0]	McBSP1_CLKR	20	GPIO_156	4
MMC2_CLK[15:0]	MMC2_CLK	21	McSPI3_CLK	1
McBSP1_CLKR[31:16]	McBSP1_FSR	22	GPIO_157	4
I2C2_SDA[15:0]	I2C2_SDA	23	GPIO_183	4
I2C1_SDA[31:16]	I2C2_SCL	24	GPIO_168	4

Cuadro 4.1: Modo de pines en bloque de expansión

Al modificar el archivo beagle.h hay que tener mucho cuidado, ya que al sustituir los valores no se deben repetir pines ni registros, no deben haber incoherencias, un registro por cada pin y un pin por cada registro. Dentro del archivo hay un macro definido MUX_BEAGLE_C(), donde se deben realizar las modificaciones ya que este macro

está asociado con el modelo de Beagleboard utilizado (C4).

En una primera instancia se sustituyeron los valores del cuadro 4.1 buscando los equivalentes del PadConf en el archivo beagle.h.

```
\#define MUX_BEAGLE_C() \
MUX_VAL(CP(MCBSP3_DX), (IEN | PTD | DIS | M4)) /*GPIO_140*/ \
MUX_VAL(CP(MCBSP3_DR), (IEN | PTD | DIS | M4)) /*GPIO_142*/ \
MUX_VAL(CP(MCBSP3_CLKX), (IEN | PTD | DIS | M4)) /*GPIO_141*/ \
MUX_VAL(CP(MCBSP3_FSX), (IEN | PTD | DIS | M1)) /*UART2_RX*/ \
MUX_VAL(CP(UART2_TX), (IDIS | PTD | DIS | M0)) /*UART2_TX*/ \
MUX_VAL(CP(MMC2_DAT7), (IEN | PTD | EN | M4)) /*GPIO_139*/ \
MUX_VAL(CP(UART2_CTS), (IEN | PTD | DIS | M4)) /*GPIO_144*/ \
MUX_VAL(CP(MMC2_DAT6), (IEN | PTD | EN | M4)) /*GPIO_138*/ \
MUX_VAL(CP(MMC2_DAT5), (IEN | PTD | EN | M4)) /*GPIO_137*/ \
MUX_VAL(CP(MMC2_DAT4), (IEN | PTD | EN | M4)) /*GPIO_136*/ \
MUX_VAL(CP(UART2_RTS), (IEN | PTD | EN | M4)) /*GPIO_145*/ \
MUX_VAL(CP(MCBSP1_DX), (IEN | PTD | EN | M4)) /*GPIO_158*/ \
MUX_VAL(CP(MMC2_DAT2), (IEN | PTD | EN | M4)) /*GPIO_134*/ \
MUX_VAL(CP(MCBSP1_CLKX), (IEN | PTD | EN | M4)) /*GPIO_162*/ \
MUX_VAL(CP(MMC2_DAT1), (IEN | PTU | EN | M4)) /*GPIO_133*/ \
MUX_VAL(CP(MCBSP1_FSX), (IEN | PTD | EN | M4)) /*GPIO_161*/ \
MUX_VAL(CP(MCBSP1_DR), (IEN | PTD | EN | M4)) /*GPIO_159*/ \
MUX_VAL(CP(MCBSP1_CLKR), (IEN | PTD | EN | M4)) /*GPIO_156*/ \
MUX_VAL(CP(MCBSP1_FSR), (IEN | PTD | EN | M4)) /*GPIO_157*/ \
MUX_VAL(CP(I2C2_SDA), (IEN | PTD | EN | M4)) /*GPIO_183*/ \
MUX_VAL(CP(I2C2_SCL), (IEN | PTU | EN | M4)) /*GPIO_168*/ \
MUX_VAL(CP(MMC2_DAT3), (IEN | PTD | EN | M1)) /*SPI3_CS0*/ \
MUX_VAL(CP(MMC2_DAT0), (IEN | PTU | EN | M1)) /*SPI3_SOMI*/ \
MUX_VAL(CP(MMC2_CMD), (IEN | PTU | DIS | M1)) /*SPI3_SIMO*/ \
MUX_VAL(CP(MMC2_CLK), (IEN | PTU | DIS | M1)) /*SPI3_CLK*/
```

Luego es necesario compilar para obtener el u-boot.bin.

```
$ cd /stuff/build
$ bitbake -f -c compile -b ../openembedded/recipes/u-boot/u-boot_git.bb
$ bitbake -f -c deploy -b ../openembedded/recipes/u-boot/u-boot_git.bb
```

Nota: Cada vez que se introduzca un nuevo cambio, no es necesario ejecutar el comando con la opción clean (lo que implica volver a bajar los fuentes), solo basta con volver a compilar.

El archivo generado (u-boot.bin) se encuentra en /stuff/build/tmp/deploy/glibc/images/beagleboard/ aunque con su nombre seguido de un número identificatorio, el cual debe ser borrado para poder mantener el nombre necesario (u-boot.bin).

Pese a que en la literatura y foros, se plantea lo contrario, no fue posible establecer los atributos “valor” y “dirección” de los pines GPIO mediante la modificación planteada (una posible solución puede verse en el apéndice D.4). Lo que sí cambia efectivamente es el modo del pin, permitiendo obtener las interfaces adecuadas en el bloque de expansión.

4.4.4. uImage

La versión del kernel elegida fue la 2.6.32 ya que en el momento que se comenzó el desarrollo era la versión más estable. Aunque también se hicieron pruebas con las versiones 2.6.29 y 2.6.37.

Durante el arranque del sistema, el kernel carga los módulos y controladores necesarios para el funcionamiento del hardware que forma parte del sistema embebido. También se montan las interfaces para poder interactuar con los distintos dispositivos a ser conectados a la Beagleboard como lo son: SPI, GPIO, UART, etc.. Las interfaces SPI y UART se encuentran bajo el directorio /dev en el sistema de archivos, y GPIO bajo el directorio /sys/class/gpio. En algunos casos, en /dev no aparecen algunas de las interfaces configuradas, lo que lleva a modificar los fuentes del kernel para que esto así suceda. Este fue el caso de la interfaz SPI que no quedó mapeada en /dev pese a que había sido configurada en los fuentes del u-boot. También hubo problemas con los atributos “valor” y “dirección” de los GPIO, como se mencionó anteriormente. Adicionalmente hacía falta un módulo para simular una conexión ethernet sobre una interfaz USB para establecer una conexión entre la Beagleboard y un PC como si fuera un enlace de red. Todo esto llevó a que se tuvieran que modificar los archivos fuente del kernel.

A continuación se detallan los pasos a seguir para la modificación de los fuentes del kernel:

Comando necesario para el desarrollo del uImage:

```
$ bitbake virtual/kernel -c [comando]
```

virtual/kernel: refiere a que se quiere generar un kernel.

Entre los comandos:

clean: borra el contenido del directorio work. Borra todos los cambios hechos en la configuración del uImage fuente y parches agregados.

patch: genera los archivos de configuración y el fuente del uImage. Además le aplica los parches.

menuconfig: abre el editor de la configuración del kernel.

compile: compila todo.

deploy: genera los archivos referidos en este caso al uImage (.config, módulos, uImage) y los guarda en el directorio /stuff/build/tmp/deploy/glibc/images/beagleboard/.

Nota: Es necesario que todos estos comandos sean ejecutados en el orden adecuado para que todo funcione correctamente.

Primero se configura bitbake para poder utilizarlo:

```
$ export BBPATH=/stuff/build:/stuff/openembedded  
$ export PATH=/stuff/bitbake/bin:$PATH
```

Se comienza el desarrollo:

```
$ cd /stuff/build  
$ bitbake virtual/kernel -c clean  
$ bitbake virtual/kernel -c patch  
$ bitbake virtual/kernel -c menuconfig
```

Luego de ejecutar este comando se abre el editor de la configuración del kernel (ver figura 4.3). Este editor indica qué módulos cargar y cuáles no. En este caso un cambio

de configuración es necesario para el buen funcionamiento de la interfaz SPI y de la conexión USB-ethernet con la Beagleboard.

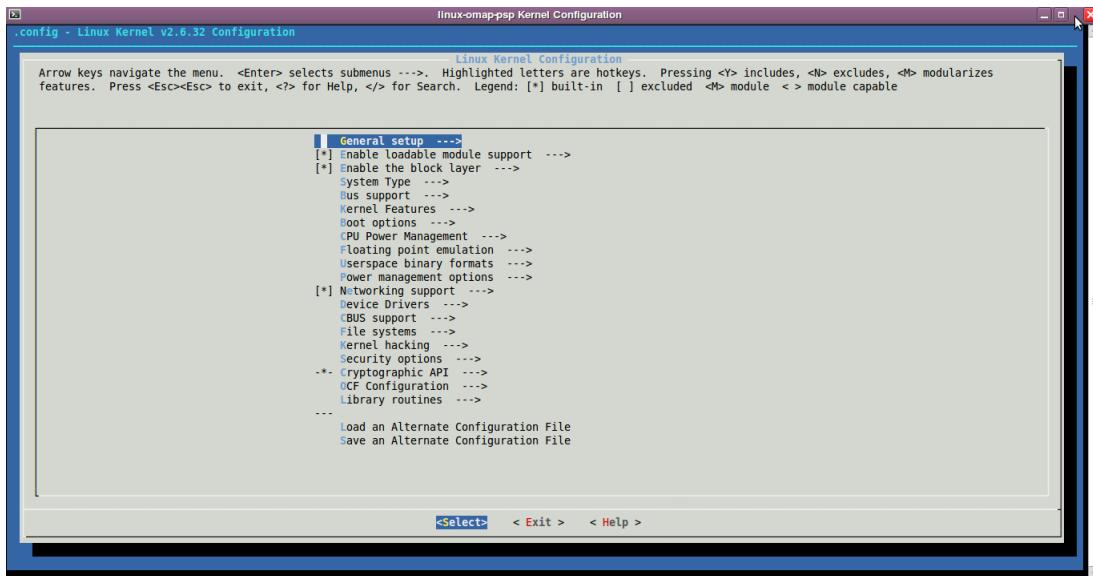


Figura 4.3: Editor de configuración del kernel

Para configurar la interfaz SPI, se debe configurar como sigue:

Device Drivers – SPI Support=y y luego como en la figura 4.4.

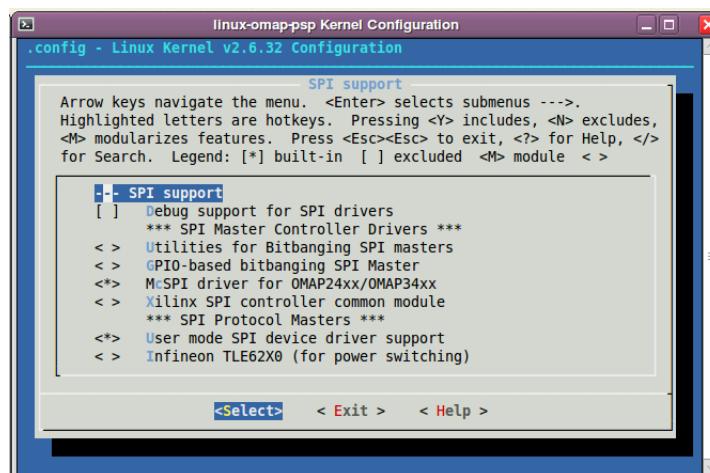


Figura 4.4: Configuración SPI

Para poder establecer la conexión por USB con la Beagleboard, se debe configurar como sigue:

Device Drivers – USB Support=y – USB Gadget Support=y y luego como en la figura 4.5.

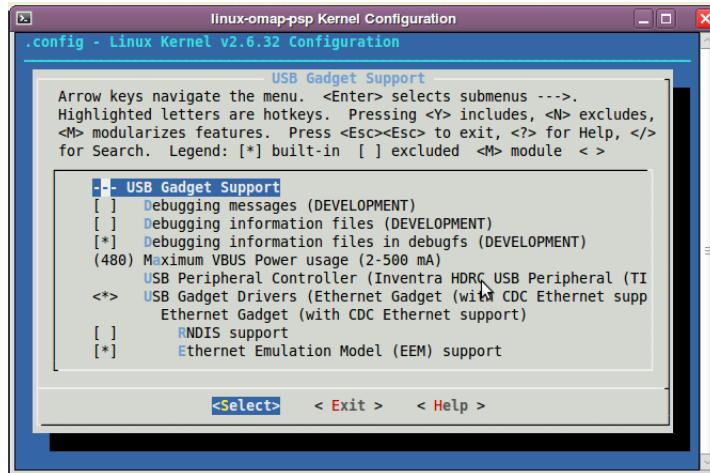


Figura 4.5: Configuración USB Gadget

Luego, es necesario modificar el archivo `board_omap3beagle.c` que se encuentra en `/stuff/build/tmp/work/beagleboard-angstrom-linux-gnueabi/linux-omap-.../git/arch/arm/mach-omap2/`. En este archivo está toda la inicialización de las interfaces. Los detalles de los cambios introducidos en este archivo se pueden observar en el apéndice D.4.

Se compila y genera el archivo uImage:

```
$ bitbake virtual/kernel -c compile
$ bitbake virtual/kernel -c deploy
```

Dentro de `/stuff/build/tmp/deploy/glibc/images/beagleboard/` se encuentra el archivo uImage generado.

Nota: Respecto al nombre del archivo, al igual que con el caso del `u-boot.bin` el nombre que aparece es un nombre más largo y necesita ser renombrado a `uImage` para que se pueda ejecutar correctamente.

4.4.5. FileSystem

Como se mencionó anteriormente, el fileSystem se generó a partir de la herramienta web Narcissus.

En el fileSystem es donde se encuentran los paquetes y programas ya instalados. Cuanto más programas se instalen más grande será en tamaño el fileSystem.

Para utilizar la herramienta Narcissus se debe acceder a la web de la herramienta [43].

A continuación se detallan las diferentes características y opciones que se eligieron para crear un fileSystem a medida para la Beagleboard y un SDK para el PC de desarrollo:

Select Machine: Beagleboard.

Image Name: el nombre que se le quiera dar.

Complexity: complejidad, se eligió “advanced” ya que la opción “simple” no brinda libertad de configuración.

Release: versión, aquí hay varias opciones disponibles, se eligió “unstable” ya que es la más estable de las disponibles. La primera opción disponible es la más estable.

Base System: aquí se elige el soporte de drivers y paquetes que se pretenden. “bare bones” es la opción con menos soporte y “extended” es la de mayor soporte. Cuanto más soporte, más pesado se hace el fileSystem. Una opción interesante es la opción “regular”, y es la que se eligió.

/dev manager: esto es el manejador de /dev, se recomienda “udev”.

Type of Image: formato en el que se quiere descargar el fileSystem. Se eligió “tar.gz” ya que es la opción más versátil.

Software manifest: se genera un archivo en la web con todos los paquetes que se instalaron en detalle.

SDK type: esta opción permite generar un SDK para el PC de desarrollo compatible con el fileSystem generado. Esto es sumamente útil por ejemplo para croscompilar. Aquí se eligió la opción “Full SDK”.

User environment section: aquí se indica que tipo de sistema operativo se quiere, básicamente se tienen dos opciones; una es un fileSystem sin interfaz gráfica y las otra con entorno gráfico. Se eligió la opción “console” (sin entorno gráfico) ya que la aplicación no exige entorno gráfico.

Luego se permite seleccionar programas que se desean instalar. Las aplicaciones elegidas son: nano editor (editor de texto) ya que hace las cosas más fáciles que el programa vi, GDB y GDBServer necesarios para la depuración de la aplicación, y toolchain para tener herramientas de compilación nativas en la Beagleboard.

Cuando todo fue seleccionado, se da un click en “build me” (demora un poco). Cuando el proceso termina, se generan dos archivos comprimidos, un archivo con el nombre elegido para la imagen en un formato .tar.gz y el SDK para el PC de desarrollo en formato .tar.bz2.

4.4.6. Beagleboard

Preparación de la memoria SD

Con los archivos MLO, u-boot.bin, uImage y fileSystem generados, se procede a la preparación de la memoria SD para correr por primera vez en la Beagleboard. En el apéndice D.8 se describen los pasos necesarios para el correcto funcionamiento del sistema.

Configuración de los parámetros de arranque

Existe una forma de saber lo que sucede en el arranque del sistema conectando la Beagleboard a un PC a través de un cable con conversor USB-SERIAL y utilizando la interfaz serial de la Beagleboard (/dev/ttyS2) para la comunicación con el PC.

Para saber sobre la configuración en el PC para conexión serial con la Beagleboard ver apéndice D.9.

Primer arranque de la Beagleboard

Para evitar problemas, se recomienda alimentar la Beagleboard con el transformador correspondiente.

Antes de arrancar se abre minicom por consola:

```
$ minicom
```

Se coloca la SD en la Beagleboard, se enchufa, y comienza el arranque del sistema.

Para que el sistema arranque correctamente con los archivos generados previamente, se debe mantener presionado el botón “user” de la Beagleboard ya que en caso contrario en el arranque se busca el u-boot.bin primero en NAND, y si ya existe una versión preinstalada es ésta la que se va a cargar. Al apretar el botón “user” (no más de tres segundos) se logra un cambio en el orden de arranque, lo que hace que se cargue primero el u-boot.bin de la memoria SD.

Luego se debe ver una cuenta regresiva en pantalla. Si se apreta alguna tecla del teclado esta cuenta regresiva se detiene y se accede a la configuración de los parámetros de arranque del sistema (similar a una BIOS de un PC de escritorio). Para ver una lista de todos los parámetros del arranque, se debe ejecutar el comando printenv. En el apéndice D.10 puede verse el listado que devuelve la ejecución de dicho comando.

Entre los parámetros a modificar están los que indican como acceder a la SD (bootargs) y los comandos de arranque (bootcmd) donde se indica en qué lugar encontrar el uImage. Para un arranque correcto a través de la SD, los valores de los parámetros anteriores deben ser los siguientes:

```
bootargs= 'console=ttyS2,115200n8 root=/dev/mmcblk0p2  
rw rootwait'  
bootcmd 'mmc init;fatload mmc 0 80300000 uImage;  
bootm 80300000'
```

En general esto ya viene seteado correctamente. Si no es así, van a haber errores al arrancar el sistema. Una solución es setear los valores de estos atributos mediante el comando `setenv`:

```
OMAP3 beagleboard.org # setenv bootargs 'console=ttyS2,115200n8  
root=/dev/mmcblk0p2 rw rootwait'
```

```
OMAP3 beagleboard.org # setenv bootcmd 'mmc init;fatload mmc  
0 80300000 uImage;bootm 80300000'
```

```
OMAP3 beagleboard.org # saveenv
```

Nota: `saveenv` guarda los valores seteados en NAND.

También se puede modificar el tiempo de la cuenta regresiva (`bootdelay`) mencionada anteriormente para lograr un mejor tiempo de arranque.

```
OMAP3 beagleboard.org # setenv bootdelay 1
```

```
OMAP3 beagleboard.org # saveenv
```

Luego de aplicados los cambios en los parámetros de arranque, se reinicia el sistema.

```
OMAP3 beagleboard.org # reset
```

Un ejemplo de arranque del sistema puede verse en el apéndice D.11.

Copia de MLO y u-boot.bin en NAND

Recordando que se debe dejar apretado el botón “user” de la Beagleboard para lograr que se cargue el `u-boot.bin` de la memoria SD y no el de la NAND (por defecto), y dado que ésto no es conveniente para un sistema que pretende ser autónomo, se decidió copiar los archivos `MLO` y `u-boot.bin` a la memoria NAND. Esta operación es muy delicada y se deben seguir los pasos que se mencionan a continuación al pie de la letra, ya que cualquier error puede dejar inservible a la Beagleboard. Antes de hacerlo es conveniente tener la versión definitiva de los archivos a copiar ya que la NAND es una memoria delicada.

Se accede a la ventana de cambios en los parámetros de arranque y se realiza lo siguiente:

```
OMAP3 beagleboard.org # mmc init  
mmc1 is available  
OMAP3 beagleboard.org # fatload mmc 0:1 80000000 MLO  
reading MLO  
19908 bytes read
```

Primero se inicializa el subsistema mmc para la memoria SD y luego se copia de la partición 1 de la mmc 0 el archivo MLO en la dirección 80000000.

Para escribir en la NAND se debe habilitar la siguiente funcionalidad.

```
OMAP3 beagleboard.org # nandecc hw  
HW ECC selected
```

Se borran las direcciones de memoria en donde se quiere escribir.

```
OMAP3 beagleboard.org # nand erase 0 80000  
NAND erase: device 0 offset 0x0, size 0x80000  
Erasing at 0x60000 -- 100% complete.  
OK
```

Se copia en NAND el MLO.

```
OMAP3 beagleboard.org # nand write 80000000 0 80000  
NAND write: device 0 offset 0x0, size 0x80000  
524288 bytes written: OK
```

Para copiar el u-boot se procede de la misma manera.

```
OMAP3 beagleboard.org # mmc init  
mmc1 is available  
OMAP3 beagleboard.org # fatload mmc 0:1 80000000 u-boot.bin  
reading u-boot.bin  
188600 bytes read  
OMAP3 beagleboard.org # nandecc sw  
SW ECC selected  
OMAP3 beagleboard.org # nand erase 80000 160000  
NAND erase: device 0 offset 0x80000, size 0x160000  
Erasing at 0x1c0000 -- 100% complete.  
OK  
OMAP3 beagleboard.org # nand write 80000000 80000 160000  
NAND write: device 0 offset 0x80000, size 0x160000  
1441792 bytes written: OK
```

Beagleboard en la red

Es de interés conectar la Beagleboard a través de una interfaz ethernet con el PC de desarrollo para poder intercambiar archivos e incluso conectar la Beagleboard a internet. Para lograr lo anterior se agregaron los módulos necesarios durante el desarrollo del uImage. Para la emulación de la conexión por red se utiliza el USB-OTG de la Beagleboard.

Conección con beagle a través del USB-OTG

Tener una conexión por red con la BeagleBoard permite el uso de comandos como ssh para acceso remoto y el envío de archivos mediante el comando scp. Si se conecta la Beagleboard con el PC de desarrollo, se puede ver en el PC la aparición de una conexión de red mediante la interfaz usb0. Se debe configurar esta interfaz como sigue:

```
$ ifconfig usb0 172.16.1.17 netmask 255.255.255.0
```

La única condición que debe tener la ip es que no sea parte de la subred a la que pertenece el PC desarrollo (en general 192.168.x.x) ya que esto lleva a errores de comunicación y por ejemplo, no permite la salida de la Beagleboard a internet.

Para que se establezca la conexión es necesaria la ejecución de algunos comandos en la Beagleboard, por ejemplo interesa una ip fija para identificarla en la red. Para que la conexión se realice cada vez que se inicia el sistema, se creó un script (ethusb) que se corre al inicio y ejecuta los comandos antes mencionados. A continuación se muestra el script ethusb:

```
#!/bin/sh
echo Ya se puede conectar...
ifconfig usb0 172.16.1.14 netmask 255.255.255.0
route add default gw 172.16.1.17
echo nameserver 192.168.1.2 > /etc/resolv.conf
```

Se configura la ip que va a tener la Beagleboard. La puerta de enlace debe ser la ip del PC de desarrollo y el nameserver debe ser la dirección de la ip de salida a internet que tiene la PC de desarrollo.

Para que el script se ejecute cada vez que arranca el sistema, se debe hacer lo siguiente:

```
$ cp ethusb /etc/init.d
```

```
$ cd /etc/rc5.d
```

```
$ ln -s ./init.d/ethusb S99ethusb
```

El S99 quiere decir que lo que se ejecuta es un script, y que es el último en ejecutarse.

Ya no va a ser necesaria una configuración cada vez que se inicie el sistema. Hay que tener el USB desconectado durante el inicio del sistema, en caso contrario al arrancar se presenta un kernel panic.

Para la conexión de la Beagleboard a internet, ver apéndice D.12.

Para el uso del gestor de paquetes (opkg), ver apéndice D.1.

4.4.7. Bibliotecas

Software para el manejo de GPIO

Este módulo de software fue realizado basándose en la interfaz genérica SYSFS para el control de GPIOs [48] [49], esto permite que el código pueda ser portado a cualquier otro sistema que use Linux y que disponga de este tipo de hardware.

El módulo de software para el uso de los puertos de propósito general, GPIO, en principio puede resultar poco importante a simple vista, pero esta porción de código es usada por el resto de los módulos que conforman la aplicación completa del prototipo RF². Este módulo cuenta básicamente con una estructura que permite almacenar el estado de cada puerto, una macro, y cuatro funciones que se datallan a continuación. La primera de las funciones se llama config_gpio_pin() y permite exportar desde el espacio kernel al espacio usuario las funcionalidades necesarias para hacer uso del puerto que se indica como argumento. Al momento en que se exporta, se indica la dirección, o sea si será un puerto de entrada o salida, a través de un parámetro que es pasado a la función. La función que permite leer el valor actual de un puerto se llama read_gpio_pin(), es necesario pasarle como argumento el indicador del puerto del cual se quiere conocer su valor. El valor del puerto es guardado en la estructura que almacena el estado de cada

puerto para posteriores consultas, sin tener que volver a llamar a dicha función. Las últimas dos funciones son contrapuestas, `set_gpio_pin()` y `clear_gpio_pin()`, éstas permiten poner el valor de un puerto específico en el valor lógico “1” o “0” respectivamente. Previo a establecer o borrar el valor del puerto ambas funciones verifican que la dirección del mismo sea de salida (como mecanismo de seguridad no es posible cambiar el valor de un puerto de entrada). Por su parte la macro `reset_status_gpio()` permite borrar el estado de un puerto que ya no esté en uso.

Software para comunicación SAM

Hoy en día la mayoría de los lectores de tarjetas de contacto tienen una interfaz USB para ser conectado en un PC en aplicaciones de escritorio. Para el uso de este tipo de lectores sobre Linux existe un controlador genérico llamado CCID. Sin embargo, las tarjetas de contacto no poseen un puerto USB sino un puerto serie para establecer la comunicación con algún dispositivo, es por esto que en la nueva generación de lectores siempre hay un ASIC para lograr la interacción, por un lado con la tarjeta de contacto y por el otro la comunicación con el PC. Como fue descrito en la sección de hardware, el lector de tarjetas de contacto tiene una interfaz serial pura para la transferencia de datos con las tarjetas. En base al diseño hardware elegido, las capas de software sobre las que se decidió trabajar son las que se detallan en la figura 4.6.

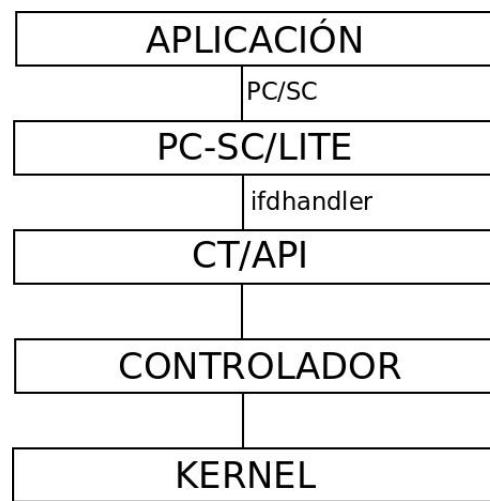


Figura 4.6: Capas de software de trabajo

En una primera etapa y para simplificar el desarrollo y la depuración del software, las capas empleadas fueron las que se muestran en la figura 4.7.

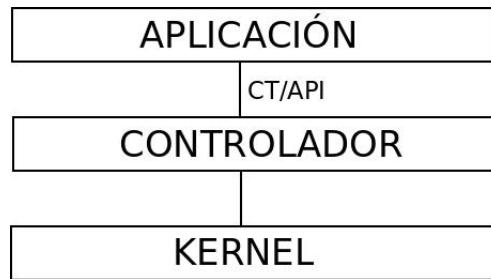


Figura 4.7: Capas de software en una primera etapa

A continuación se describen las capas asociadas a la figura 4.6.

Controlador:

El kernel es el encargado de manipular directamente los registros del puerto serial, las interrupciones que desde éste se generan y la ISR para atender las interrupciones. La implementación del controlador del lector de tarjetas se basó en el controlador serial de Linux a través de su estructura “termios” [44]. Esta estructura permite configurar todos los parámetros necesarios para la comunicación serial como ser, baud rate, cantidad de bits por byte, bit de paridad, bit de parada entre otros. Las funciones read y write permiten la lectura y escritura de los bytes de datos que son recibidos y transmitidos por el puerto serial.

Dentro del proyecto MUSCLE/PCSCLite existe código ejemplo de controladores de lectores de tarjetas de contacto que son compatibles con la biblioteca, si bien el desarrollo se basó en alguno de ellos, como el hecho de usar el mismo nombre de funciones, su contenido fue desarrollado específicamente para controlar el hardware específico del prototipo RF².

CT/API (Card Terminal / Application Programming Interface):

Por encima del controlador serial se encuentra CT/API [22], una interfaz definida por varias empresas (entre las que se incluye Telekom Alemania) en la década de los noventa, que permite encapsular el controlador específico de cada lector de tarjetas, de

manera que la aplicación final no se vea afectada al cambiar un lector por otro. Esta interfaz de programación está formada tan solo por tres funciones, CT_init, CT_data y CT_close, que permiten la inicialización del lector, la transferencia de datos entre host/lector o host/tarjeta (host se refiere a la SBC o PC donde se encuentra conectado el lector de tarjetas de contacto) directamente y el cierre de la comunicación.

CT_init se encarga del pasaje de parámetros a la capa del controlador, para la configuración del puerto de comunicación entre el host y el lector de tarjetas. Los parámetros en uso aquí son: la tasa de transferencia de datos, el número de bits por cada byte, el tipo de paridad empleado y el puerto serie a ser utilizado.

CT_data es la función encargada de transferir comandos y datos hacia y desde la tarjeta o hacia y desde el lector (en caso que el mismo esté formado por un ASIC o microprocesador). La manera de diferenciar desde donde es enviado el dato, es a través de un parámetro pasado a esta función, y de forma análoga se determina el destino del mensaje. El protocolo usado para la transferencia de datos es T=0, orientado a bytes y del cual pueden conocerse más detalles en [18].

CT_close es la contracara de CT_init, se encarga de cerrar la comunicación con el lector. Lo que hace básicamente es liberar el handle (puntero) asociado al puerto serial.

Las funciones de apertura y cierre de la comunicación no necesitan ser modificadas, son similares en todos los controladores de lectores seriales compatibles con la biblioteca. Por otro lado la función CT_data debió ser sustancialmente modificada para lograr el intercambio de datos a partir del protocolo T=0.

Para el caso en que los comandos y/o datos estén dirigidos hacia el lector, existe otra especificación, llamada CT/BCS [23] (Card Terminal / Basic Command Set), donde se encuentran definidos una serie de comandos básicos para el manejo del lector. Estos comandos se nombran a continuación y se da una breve descripción.

RESET CT permite reiniciar el lector o tarjeta (en el caso de ser un lector mudo); de manera opcional puede devolver el ATR.

REQUEST ICC tiene como objeto devolver el ATR de la tarjeta una vez que la misma se encuentra ubicada en el zócalo del lector.

GET STATUS es empleado para conocer información sobre el lector o si la tarjeta está insertada y eléctricamente conectada en el lector.

EJECT ICC genera la desactivación eléctrica de la tarjeta.

IFDHandler:

El siguiente componente en este stack de capas es ifdhandler [24], no es otra cosa que un conjunto de funciones formando una API, empleada por pcsclite para encapsular el manejo del hardware de lectores cuyos fabricantes quieran cumplir con las especificaciones PC/SC [102]. Una ventaja importante de esta API es que le permite a pcsclite operar tanto con lectores de puerto serial como con lectores de puerto USB. Esta capa de software podría usarse directamente sobre el controlador del lector, prescindiendo de CT/API, aunque se decidió mantenerla por motivos de simplicidad ya que sólo es necesario sustituir la capa de aplicación por las restantes capas superiores como se indica en las figuras 4.6 y 4.7.

Las funciones que contiene esta capa de software fueron modificadas para que contengan a su vez las funciones de CT/API, y de esta manera hacer más fácil de integrar el controlador del lector con la biblioteca PCSCLite.

En lo que sigue se enumeran algunas de las funciones de esta API y se describen brevemente. Por más detalles ver el manual ifdhandler [24].

IFDHCreateChannel establece el canal de comunicación con el lector. Para conseguirlo usa un parámetro llamado Channel, que indica cual es el puerto serial a usar, por ejemplo para el caso de Linux /dev/ttySx (x es el número que corresponda).

IFDHCloseChannel implementa la acción opuesta a la función anterior, cerrando el canal de comunicación con el lector de tarjetas.

IFDHGetCapabilities permite obtener las capacidades específicas del lector o de la tarjeta insertada en el mismo.

IFDHPowerICC se encarga del control de las señales de alimentación y reset que el lector suministra a la tarjeta. Desempeña tres acciones posibles, encendido, reset y apagado de la tarjeta.

IFDHTransmitToICC se encarga de la transferencia de datos con la tarjeta a través de alguno de los protocolos disponibles, como ser T=0 o T=1.

IFDHICCPresence retorna el estado de la tarjeta insertada en el zócalo del lector.

PCSCLite:

Por arriba de ifdhandler se encuentra la biblioteca pcsclite, ésta contiene todas las funciones necesarias para establecer la comunicación con un lector y la tarjeta conectada a éste último. Para el uso del controlador encapsulado por ifdhandler desde pcsclite es necesario seguir los pasos de configuración detallados en el apéndice D.13.

Aplicación final:

Por arriba de todas las capas descritas antes, se encuentra la aplicación del prototipo que hace uso de las funciones suministradas por pcsclite y donde se encuentran definidos los comandos APDU específicos con los que opera la tarjeta de contacto. Por razones de seguridad no se nos permite difundir la lista de comandos que son usados para la comunicación con el módulo SAM.

Software RFID

librfid es una biblioteca de software libre para manejo de lectores/escritores RFID. Implementa el stack de protocolos del lado del dispositivo lector/escritor ISO 14443A, ISO 14443B, ISO 15693, Mifare Ultralight y Mifare Classic.

Entre los lectores soportados están, OpenPCD y algunos modelos Omnikey, éstos con interfaz de conexión USB. Además tiene soporte para cualquier otro lector con comunicación directa con el CL RC632 mediante la interfaz SPI y es por esta razón que se tuvo en cuenta.

El manejo de librfid es de bajo nivel y se comunica directamente con el kernel utilizando el módulo spidev. Librfid-tool implementa funciones de más alto nivel que hacen uso de las funciones de bajo nivel de la librfid. La estructura de software comentada se muestra en la figura 4.8.

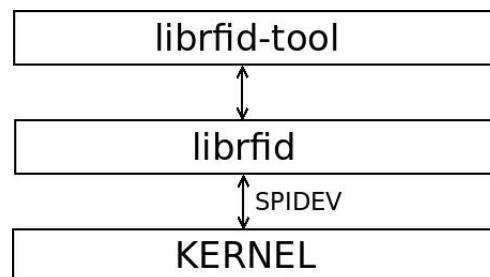


Figura 4.8: Capa de software RFID

A continuación se detallan los cambios introducidos en la biblioteca librfid para el correcto funcionamiento del lector/escritor RFID utilizando librfid-tool.

Se analizó el código principal de la aplicación librfid-tool y se vio que dentro del directorio utils en el archivo common.c se encuentra la función reader_init la cual busca un lector/escritor entre los soportados. Esta función no tenía implementada una búsqueda para dispositivos conectados por SPI. Por lo tanto se tuvieron que agregar las siguientes líneas a la función para que el funcionamiento fuera posible:

```
rh = rfid_reader_open ("/dev/spidev3.0", RFID_READER_SPIDEV);
if (!rh) {
    fprintf(stderr, "No SPIDEV found\n");
    return -1;
}
```

Este cambio permitió detectar la interfaz SPI (spidev3.0). Algo a tener en cuenta, es que como en Linux las interfaces están asociadas a un archivo, el hecho de abrir el archivo no implica que haya nada conectado en esa interfaz. Por esta razón, la búsqueda de un lector/escritor conectado por interfaz SPI se realiza en última instancia.

Otro cambio fundamental es en la frecuencia de reloj del puerto SPI. Se incrementa a 10MHz, ya que con la frecuencia establecida por defecto en la biblioteca librfid (1MHz) el lector/escritor RFID no funciona correctamente.

Toda la configuración de la comunicación por SPI se encuentra en rfid_reader_spidev.c que está en el directorio fuente src.

La función que se modificó es spidev_open y el cambio se muestra a continuación:

```
tmp = 10e6; /* 10 MHz */
if (ioctl(spidev_fd, SPI_IOC_WR_MAX_SPEED_HZ, &tmp) < 0)
    goto out_rath;
```

Se implementaron las funciones adecuadas para el manejo del pin RST_RF, el cual puede encender y apagar el lector/escritor RFID diseñado.

Después de estudiadas las funciones que provee la herramienta librfid-tool, se estudió la posibilidad de su uso para la aplicación RF².

Aunque sus funciones son muy útiles, la gran mayoría no sirven completamente para la aplicación RF² debido a que fueron definidas para otros propósitos.

Es de interés, que la herramienta librfid-tool siga manteniendo sus funcionalidades y pueda convivir con la aplicación RF², por lo que no se modificó el contenido de ninguna función de la herramienta. En el caso que alguna función fuera mayormente utilizable, se procedió a crear una nueva con los cambios necesarios.

Se implementaron la mayoría de las funciones, logrando compatibilidad con la biblioteca librfid. Entre las funciones creadas están las asociadas con la tarjeta RFID: autenticación según el tipo de clave, búsqueda de tarjetas próximas al lector, lectura y escritura de bloques de memoria, obtención del UID, etc.

4.4.8. Aplicación final

Para el desarrollo de la aplicación RF² se decidió trabajar sobre los fuentes de la herramienta librfid-tool, ya que maneja varias funciones de utilidad y es de ayuda a la hora de compilar para el armado de una aplicación completa. Se mantuvieron todas las opciones de la herramienta ya que son muy útiles, y pueden ayudar en un futuro para establecer orígenes de fallas. No se modificó ninguna función de la aplicación original y cuando fue necesaria alguna modificación, se procedió a implementar una nueva.

En la figura 4.9 se detallan las capas de software del sistema RF².

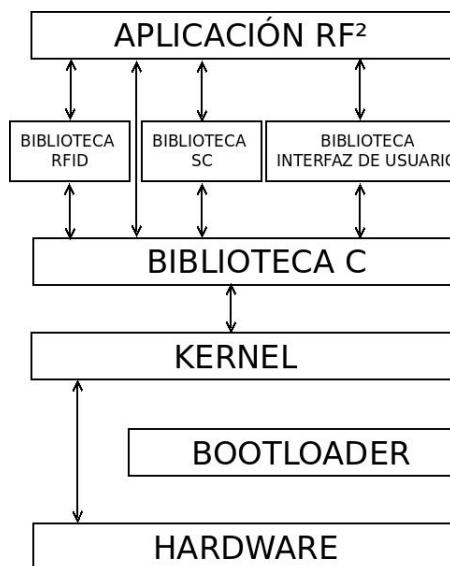


Figura 4.9: Capas de software del sistema RF²

Antes de seguir fue necesario entender el funcionamiento de las reglas de compilación creadas para la aplicación librfid-tool sobre librfid. En el directorio raíz, se encuentran los siguientes archivos importantes para el desarrollo del sistema: autogen.sh, configure.in, configure, Makefile.am, Makefile.flags.am, Makefile.in, Makefile.

A continuación se describe a grandes rasgos la utilidad de cada uno de estos archivos:

configure.in es el archivo de configuración con el cual se crea el archivo configure.

Makefile.am es el archivo que establece las reglas de compilación (orden, dependencias, etc) y es con el cual se crea el archivo Makefile.in. Diferentes Makefile.am se encuentran en los subdirectorios de la librfid, por lo que se crea un Makefile.in dentro de cada subdirectorio.

Makefile.flags.am es un archivo con banderas establecidas para el sistema. Este archivo está incluido en todos los Makefile.am de la aplicación.

autogen.sh es un script que se encarga de crear los archivos configure y Makefile.in, cada uno de ellos creados a partir de los archivos correspondientes antes mencionados.

Al ejecutar el archivo configure, éste establece la configuración que necesita el Makefile.in para conocer las reglas de compilación que se van a usar y algunos parámetros extra. Luego de este proceso se generan los Makefile.

El archivo Makefile es el que conoce las reglas de compilación, las dependencias y las opciones elegidas para el desarrollo. Sabe incluso donde se encuentran los demás Makefile para ejecutarlos cuando sea necesario.

Cada subdirectorio tiene reglas locales de construcción de sus objetos, y a su vez se ayudan mutuamente para lograr una aplicación más completa. Desde el Makefile principal se controla que todo funcione correctamente. Para saber más sobre reglas de compilación puede visitarse [46].

La aplicación RF² utiliza otras bibliotecas además de librfid, además estas bibliotecas interactúan entre sí, lo que lleva a que aparezcan nuevas dependencias. Por lo tanto, fue necesario modificar las reglas de compilación.

Se agregó en el raíz de la librfid el directorio rf2, en el cual se encuentran subdirectorios con los fuentes necesarios para la comunicación con el display, leds, buzzer, tarjeta de contacto y otros utilitarios, formando la estructura que se muestra en la figura 4.10.



Figura 4.10: Estructura de árbol de aplicación RF²

gpio incluye fuentes para el manejo de los GPIO.

lcd incluye fuentes para el manejo del display.

rf incluye fuentes con funciones extra de utilidad para el manejo del lector/escritor RFID diseñado.

sam incluye los fuentes para la comunicación con la tarjeta de contacto.

utiles incluye fuentes con funciones útiles de la aplicación.

Dentro de cada uno de estos directorios se encuentra un archivo Makefile con las reglas de compilación correspondientes a cada uno.

Las modificaciones a los archivos de construcción de la aplicación para que contemplan el agregado del directorio rf2, como la creación de nuevas reglas para la construcción dentro de éste, se detallan a continuación.

Antes que nada se decidió modificar los archivos Makefile.am para que sepan de la existencia del nuevo directorio. Se modifican estos archivos debido a que nunca son borrados. Por ejemplo, si los cambios se hacen sobre los Makefile.in, puede pasar que en algún momento se regeneren borrando los cambios que se hicieron. Además, luego de entender la estructura y funcionamiento de estos archivos, es más fácil incluir una modificación en Makefile.am que en un Makefile.in o Makefile directamente.

Dentro del directorio utiles, se creó el archivo Variables_Make que establece el valor de algunas variables específicas de la aplicación RF² como ser CC_arm que hace referencia al gcc asociado con la herramienta de croscompilación para ARM.

Makefile.flags.am: Aquí se indicó que se agregue rf2 a la ruta de búsqueda de archivos encabezados.

```
INCLUDES = \$(all_includes) -I$(top_srcdir)/include  
-I$(top_srcdir)/rf2
```

Makefile.am en el raíz:

```
include rf2/utiles/Variables_Make
```

Con esto se aseguró que el resto de los subdirectorios pertenecientes a la librfid, sepan los valores de las variables específicas de la aplicación RF².

Luego se incluye el directorio rf2 a la aplicación, teniendo en cuenta que solo se va a incluir cuando se use un lector/escritor RFID con interfaz SPI.

```
if ENABLE_SPIDEV  
SUBDIRS += rf2  
endif
```

Makefile.am en utils: Este es el cambio más difícil de entender y surge de la observación del Makefile.in generado cada vez (prueba y error). Como la aplicación RF² se basa en la modificación de los fuentes de la herramienta librfid-tool, se deben agregar todas las dependencias con archivos del nuevo subdirectorio rf2.

```
librfid_tool_SOURCES = librfid-tool.c librfid-tool.h  
common.c common.h ../rf2/gpio/gpio.c ../rf2/gpio/gpio.h  
../rf2/rf/rc632_utils.c ../rf2/rf/rc632_utils.h  
../rf2/lcd/lcd16x2.c ../rf2/lcd/lcd16x2.h ../rf2/sam/sam.c  
../rf2/sam/sam.h ../rf2/sam/sam_util.c ../rf2/sam/sam_util.h  
../rf2/utiles/utiles.c ../rf2/utiles/utiles.h
```

Cada .h y .c se traduce en un .o al crear el Makefile.in.

Otro cambio necesario, ya que si no se incluye provoca errores de compilación, es el siguiente.

```
mifare_tool_SOURCES = mifare-tool.c common.c  
..../rf2/gpio/gpio.c ..../rf2/rf/rc632_utils.c  
..../rf2/lcd/lcd16x2.c ..../rf2/sam/sam.c  
..../rf2/sam/sam_util.c ..../rf2/utiles/utiles.c
```

mifare-tool es otra herramienta de la librfid.

Luego se creó dentro del directorio rf2 un archivo Makefile que es el que ordena la construcción de todos los objetivos dentro de cada subdirectorío.

```
include utiles/Variables_Make

all: gpio/gpio.o rf/rc632_utils.o lcd/lcd16x2.o  
sam/sam.o utiles/utiles.o

gpio/gpio.o:  
$(MAKE) -C gpio

rf/rc632_utils.o:  
$(MAKE) -C rf

lcd/lcd16x2.o:  
$(MAKE) -C lcd

sam/sam.o:  
$(MAKE) -C sam

utiles/utiles.o:  
$(MAKE) -C utiles

install:

clean:  
rm -f *.o  
$(MAKE) -C gpio clean  
$(MAKE) -C rf clean  
$(MAKE) -C lcd clean  
$(MAKE) -C sam clean  
$(MAKE) -C utiles clean

distclean: clean
```

En este caso \$(MAKE) -C “directorio” indica que la regla de construcción del objetivo se encuentra en “directorio”.

Se implementó una función de inicialización, la cual inicializa todos los periféricos al arrancar la aplicación RF².

En librfid-tool.h se definieron unas constantes simbólicas para hacer configurable al sistema. Algunas de ellas son:

MONEDERO: lugar del arreglo “bloque” donde se encuentra el monedero.

SECTOR_MONEDERO: sector donde se encuentra el monedero.

BLOQUE_MONEDERO: bloque donde se encuentra el monedero.

BLOQUE_RESPALDO: bloque donde se encuentra el respaldo del saldo.

TIEMPO_BUZZER: tiempo de encendido del buzzer en μs .

PAUSA_BUZZER: tiempo entre dos encendidas del buzzer en μs .

TIEMPO_LED: tiempo de encendido del led en s.

INICIO: tiempo de espera al inicio en s.

ESPERA: tiempo de espera estándar en s.

FIN: tiempo de espera al final en s.

Se agregó una nueva opción (n) en el main de la aplicación librfid-tool que llama a la función principal(). Ésta, es la función principal de la aplicación RF², la cual se ejecuta en loop y se basa en el diagrama de flujo de la figura 2.3. De este modo no se modifica el main original de librfid-tool (solo se agrega la opción n) y se dejan las opciones por defecto. Se hizo uso de algunas funciones ya escritas y se crearon otras. En este punto no se modificó ninguna función de la aplicación original y cuando fue necesaria alguna modificación, se procedió a implementar una nueva función con las modificaciones previstas. Luego viene la etapa de croscompilación de la aplicación para ser probada en la SBC.

Croscompilación de la aplicación final: Para guardar el resultado de la croscompilación se creó un directorio work en el home del usuario.

```
$ ./autogen.sh
```

Este paso es necesario siempre que se modifiquen los Makefile.am, en otro caso no. Se recuerda que este script genera los archivos Makefile.in y configure.

```
$ ./configure --enable-spidev --host=arm-angstrom-linux-gnueabi --prefix=/home/proyecto/work
```

Se configura el sistema para lectores/escritores RFID con interfaz SPI, se indica la arquitectura de la SBC para la cual se compila y se indica el directorio donde se instala la aplicación. Luego de hecho esto ya no es necesario volver a ejecutarlo ya que los Makefile quedan creados con esta configuración.

```
$ make clean && make -j5 && make install
```

Se construye la aplicación.

En el directorio work se encuentran cuatro nuevos directorios creados:

bin: incluye los binarios construidos.

include: directorio con los .h necesarios para correr la aplicación.

lib: la biblioteca en sí.

share: nada de utilidad.

El paso siguiente es copiar estos directorios en el sistema de archivos de la SBC.

Copia de archivos a la SBC:

Para realizar la copia conviene comprimir el resultado de la croscompilación y luego enviarlo a la SBC.

```
$ tar -czf rf2.tar.gz bin include lib share
```

Luego de enviar el archivo comprimido, es necesaria la copia de estos directorios en el sistema de archivos de la SBC.

Instalación en la SBC:

```
$ tar -xf rf2.tar.gz -C /usr
```

Esto descomprime los directorios creados anteriormente, dentro del directorio /usr.

Se ejecuta la aplicación RF²:

```
$ librfid-tool -n
```

Parte III

Ensayos

Capítulo 5

Ensayos

5.1. SBC

Hawkboard

Las Hawkboard fabricadas entre el 1º de agosto y el 20 de octubre de 2010, fueron vendidas en el mercado con un error a nivel de hardware que no había sido constatado por el fabricante, y que no fue reconocido por éste hasta el mes de noviembre. La solución al problema fue liberada en la fecha 20 de diciembre de 2010, y constaba de sustituir en el circuito los ferrites FB12 y FB13 por un puente de soldadura de estaño (el uso de jumper 0R fue probado sin obtener buenos resultados). Mayores detalles de la solución pueden encontrarse en el documento Hawkboard Press Release Solution que se adjunta en el apéndice E. El inconveniente mencionado antes evitaba que el sistema operativo iniciara correctamente, en algunos casos se bloqueaba y en otros daba error en el arranque. Esto evitó que se pudieran probar las partes de hardware y software que se tenían desarrolladas hasta ese entonces, teniendo que recurrirse a mecanismos alternativos, como el uso de un microprocesador rabbit para efectuar pruebas sobre el lector/escritor RFID.

Beagleboard

Esta SBC no presentó problemas a nivel de hardware, por lo tanto las pruebas se basaron en verificar los cambios realizados a nivel de software.

Luego de la configuración de los pines del bloque de expansión a nivel de software, se realizaron las pruebas sobre las interfaces configuradas en dichos pines.

Pruebas sobre las interfaces

Testeo de GPIO: Para el testeo de los GPIO, se compiló y probó el archivo led.c (ver apéndice D.14.1) el cual cambia el valor del pin 13 del bloque de expansión cada un segundo. Si se coloca un led entre este pin y nivel de referencia 0V, se puede ver como el led prende y apaga.

Testeo de UART: Para el testeo de la interfaz serial UART, se compiló y probó el programa uart.c (ver apéndice D.14.2) que envía una serie de caracteres por el pin de transmisión de la interfaz UART, y luego lee por el pin de recepción de la misma el eco proveniente del pin de transmisión. Para verificar el correcto funcionamiento se deben cortocircuitar estos pines.

Testeo de SPI: Para el testeo del puerto SPI, se utilizó una aplicación (spidev_test) que verifica el eco recibido cuando los pines de transmisión y recepción son cortocircuitados.

En este caso se debe ejecutar el archivo con los parámetros correspondientes:

```
$ ./spidev_test -D /dev/spidev3.0
```

spidev3.0 porque se está utilizando el puerto spi3 con chip select 0.

Cuando se ejecuta, deben aparecer en pantalla varias filas con “FF”. Si se cortocircuita SIMO y SOMI algunas filas deberían cambiar, con lo que queda verificado el buen funcionamiento del puerto SPI.

También se hicieron pruebas con osciloscopio, con el fin de observar la forma de las señales generadas a partir del comando:

```
$ ./spidev_test -D /dev/spidev3.0 -s [Hz]
```

La opción s permite configurar la frecuencia en Hz de transmisión de datos.

5.2. VLT - Conversor de Voltajes

No existieron problemas en este módulo, y dadas las características del circuito no hay demasiados puntos de falla. Si fuera necesario verificar la diferencia de potencial en el regulador, la tensión de entrada puede ser medida desde el conector CONN_14x2 y la

de salida desde el conector CONN_20x2, ver figura C.1. Un detalle a tener en cuenta a la hora de medir los valores de tensión de las señales que pasan a través de los conversores de nivel, cuando las mismas se encuentren en estado ocioso (estáticas), es que no debe hacerse con multímetros de mala calidad, o se obtendrán valores incorrectos durante la medición. Se recomienda para una correcta medición el empleo de osciloscopio con puntas x10. Como se mencionó antes no se tuvieron inconvenientes con este módulo, pero generó conflictos en el circuito conversor full a half duplex del lector de tarjetas de contacto que serán detallados más adelante.

5.3. SCUI - Lector de tarjetas de contacto e Interfaz de usuario

Lector de tarjetas de contacto ISO7816

Las primeras pruebas realizadas sobre el lector de tarjetas de contacto se efectuaron sobre una placa de circuito impreso de fabricación propia, conectándose el lector directamente sobre el conector de expansión de la Beagleboard. La intención de esta prueba era más que nada la de probar el circuito conversor full a half duplex, transmitiendo una serie de bytes por el canal Tx y recibiendo el eco mediante el canal Rx, cotejando que los bytes recibidos coincidieran con los transmitidos. El primer problema encontrado estuvo asociado a una falla en uno de los transistores, el PNP 3906, que debió ser sustituido por encontrarse defectuoso. El software utilizado para efectuar las pruebas sobre el hardware se basa en un controlador serial desarrollado por el grupo mina del INCO [51], el cual fue modificado ya que uno de los parámetros, CSIZE, en la configuración del puerto afectaba el número de bits que conforman un byte recibido. La línea de código que hacía referencia a este parámetro fue comentada ya que modificaba el valor del parámetro csN, con N=5 en lugar de N=8 (donde N es el número de bits que forman el byte). El cambio anterior permitió que los bytes recibidos en el canal Rx coincidieran con los transmitidos en Tx, validando en una primera instancia el hardware conversor full a half duplex del lector de tarjetas. El siguiente paso fue intercalar entre la Beagleboard y el lector de tarjetas de contacto, el conversor de niveles (VLT) para realizar las mismas pruebas que se datallaron antes. En este caso los resultados no fueron alenta-

dores ya que los bytes recibidos no coincidían con los transmitidos. Todo indicaba que el conversor de nivel afectaba el conversor full a half duplex. Luego de algunas pruebas más sobre el circuito, sin cambios favorables, se decidió consultar al foro de Texas Instruments (fabricante del integrado TXB0108). Desde el soporte técnico solicitaron se les enviara una imagen capturada con osciloscopio de las señales en el puerto serial para observar la forma de los pulsos. En la figura 5.1, se puede ver la deformación de los pulsos en la señal Rx (canal 1 del osciloscopio) cuando el circuito contaba con un valor de 500 Ohms en la resistencia R9 (ver figura C.3); la solución encontrada fue disminuir el valor de R9 y no aumentarlo como se había intentado anteriormente sin resultados favorables.

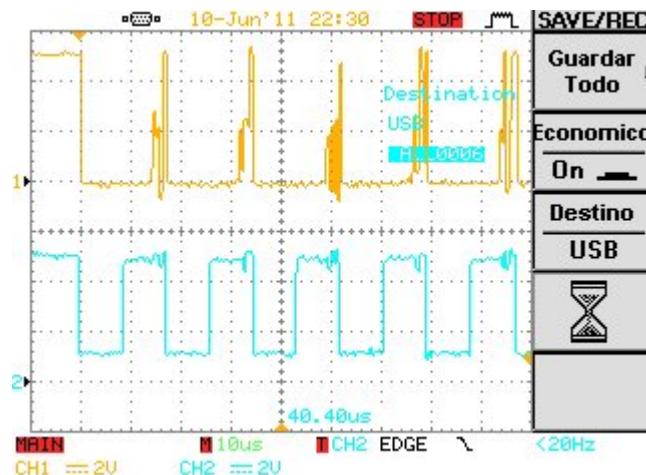


Figura 5.1: Señal en canal Rx para un valor de resistencia R9 de 500Ω

Al usar valores entre 90 Ohms y 180 Ohms para la resistencia R9, la forma de los pulsos recibidos en Rx (canal 1) fueron la copia de los pulsos transmitidos en Tx (canal 2), como puede verse en la figura 5.2 para un valor de 90 Ohms.

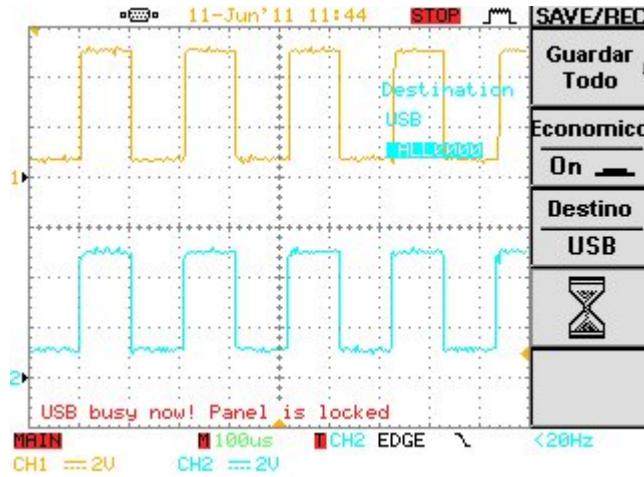


Figura 5.2: Señal en canal Rx para un valor de resistencia R9 de 90Ω

En la siguiente referencia puede verse el hilo de discusión en el foro [53].

Una vez superados los obstáculos anteriores, fue posible probar el circuito completo del lector, incluyendo la tarjeta de contacto en su zócalo correspondiente. El software utilizado en tal fin se basa en un controlador serial, implementado por David Corcoran (uno de los desarrolladores de pcsclite), el cual debió ser modificado para usarse en el lector de tarjetas de contacto del prototipo RF². Una de las mayores dificultades encontradas en esta etapa, fue hallar los parámetros adecuados de inicialización del puerto serial, que debe cumplir con las opciones 8E2 (8 bits por byte, bit de paridad par, y dos bits de parada) para operar con las tarjetas de contacto compatibles con la norma ISO7816. Sin embargo las opciones adecuadas elegidas en la configuración del puerto serial fueron 8E1. Adicionalmente al problema de encontrar las opciones correctas mencionadas antes, los bytes de datos recibidos como el ATR de la tarjeta no coincidían en su totalidad con los valores esperados (leídos con un lector Omnikey 3121 y la herramienta pcsc_scan de pcsclite), sólo algunos bytes y algunos nibbles bajos eran correctos. Esta diferencia estuvo asociada a la frecuencia utilizada para alimentar la señal de reloj de la tarjeta de contacto; se usaron frecuencias de 4 MHz y 5 MHz que si bien podrían usarse según se indica en [18] para los parámetros especificados en el ATR de las tarjetas empleadas, estos valores no fueron adecuados según lo ya comentado, teniendo que utilizar en su lugar un oscilador de frecuencia 3,579545 MHz, valor que no se consi-

guió cuando se realizó la primer compra de componentes. Una dificultad adicional tuvo que ser sorteada en este módulo de hardware, el diseño del PCB que se envió a fabricar tenía un error, las pistas de datos de Rx y Tx estaban intercambiadas. El diseño tuvo que ser corregido y se envió a fabricar un nuevo PCB.

Interfaz de usuario

En un principio, se intentó el uso de la biblioteca LCD4LINUX [52]. Éste, es un programa que se comunica directamente con el kernel para desplegar mensajes en displays con controladores como el del display utilizado en el proyecto. Es de fácil uso y configuración, el envío de los mensajes a desplegar en el display es simple. El único inconveniente encontrado, es que la única posibilidad de configuración para el display del proyecto en el programa era por puerto paralelo. La Beagleboard no cuenta con este tipo de interfaz. Se probaron varias soluciones tratando de emular un puerto paralelo a partir de cuatro pines GPIO, pero sin grandes resultados. Se decidió reutilizar una biblioteca que el grupo de trabajo desarrolló en el marco del proyecto Audio Fingerprint del curso Procesadores Digitales de Señal [54].

Luego, no existieron mayores inconvenientes con la interfaz para el usuario, sí fue necesaria la corrección en el valor de una resistencia en el circuito que calibra el contraste del LCD, ya que los caracteres se observaban muy tenues.

Al momento de probar el display imprimía caracteres extraños, salvo cuando se enviaban mensajes conteniendo una única palabra. Se probó cambiando los mensajes a desplegar en el mismo, y el problema persistía, pero se llegó a la conclusión de que era provocado por los espacios en blanco (“ ”) puesto que cuando se envió un mensaje omitiéndolos fue deplegado en forma correcta. Luego simplemente se modificó el código fuente, para que cada vez que recibiera un carácter espacio en blanco, enviara al display el carácter ASCII correspondiente solucionando el problema.

Cuando se comenzaron a imprimir los saldos de las tarjetas, volvió a imprimir caracteres extraños, esta vez el problema eran los caracteres “0” (cero). La solución encontrada fue imprimir “O” (letra o mayúscula) cada vez que llegara un carácter “0”, por lo que se modificó el código para que así sea. Estos errores se encuentran asociados al display y no al código, que ha funcionado correctamente con otros displays.

5.4. Lector/Escritor RFID

Al comenzar la fase de pruebas sobre el lector/escritor RFID, la SBC que había sido elegida no estaba operativa por razones que ya se explicaron, y la substituta no podía ser conectada directamente al lector por incompatibilidad en los niveles de tensión que manejan sus correspondientes interfaces. Como mecanismo alternativo se usó un módulo Rabbit, RCM4300, que permitió desarrollar una pequeña aplicación de software para hacer ciertas pruebas que validaran el diseño del hardware que se tenía hasta el momento. El software implementado se basó en el estudio de la hoja de datos del integrado CL RC632, y en la biblioteca librfid [30].

Una vez conectado el lector/escritor RFID al conector de expansión del microcontrolador, el paso siguiente fue configurar el puerto SPI, para esto se contó con una biblioteca provista por Dynamic C (IDE para los microcontroladores Rabbit) que permite la configuración del puerto y posee funciones para la transmisión y recepción de datos. Contar con 4 modos posibles para la configuración de este puerto del microcontrolador dificultó la tarea; fue necesario emplear un osciloscopio para observar cual de éstas se adecuaba a la forma de señal que se indica en la hoja de datos del integrado CL RC632. Una vez seleccionado el modo correcto se pudo obtener una lectura válida del valor por defecto de los registros de página 0 del CL RC632 luego de una inicialización, no así la de registros que se encuentran en otras páginas (ver 3.4.4 Registros), para alcanzar los demás registros fue necesario implementar una función que configurase el integrado para obtener direccionamiento plano de los registros y no por páginas. Validada la comunicación microcontrolador - CL RC632, lo siguiente fue implementar un conjunto de funciones que permitieran la lectura/escritura de registros de control, el buffer de datos, la memoria EEPROM, el establecimiento y borrado de bits de configuración, así como también se escribieron las funciones para generar el formato adecuado de las claves de autenticación y su posterior almacenamiento en memoria. Cuando se culminaron las pruebas sobre el módulo digital del CL RC632, lo próximo fue poner en funcionamiento el transmisor para establecer una comunicación con las tarjetas RFID, esta etapa fue la más compleja y que se prolongó por mayor tiempo, pues se cayó en la disyuntiva si la imposibilidad en la comunicación por RF se debía a un error cometido a nivel de hardware o de software. Por un lado se podía pensar que al seguir las pautas de diseño y los

cálculos indicados en [4] el error no debía ser de hardware, pero a medida que los cambios a nivel de software no generaban resultados favorables, la balanza se inclinó hacia el lado del hardware. Cuando no se tiene al alcance el instrumental adecuado, un elemento que fue de mucha utilidad a la hora de comprobar la existencia de campo magnético fue un detector de campo magnético fabricado a partir de un alambre de cobre al que se le dio forma de bobina y se le soldó un led en sus extremos (ver figura 5.3). Esta simple herramienta indica la presencia de campo magnético generado en las proximidades de la antena al encender su led; no olvidemos que debajo de la complejidad que puede llegar a tener este tipo de lector/escritor RFID se encuentran los principios básicos de la ley de Faraday.

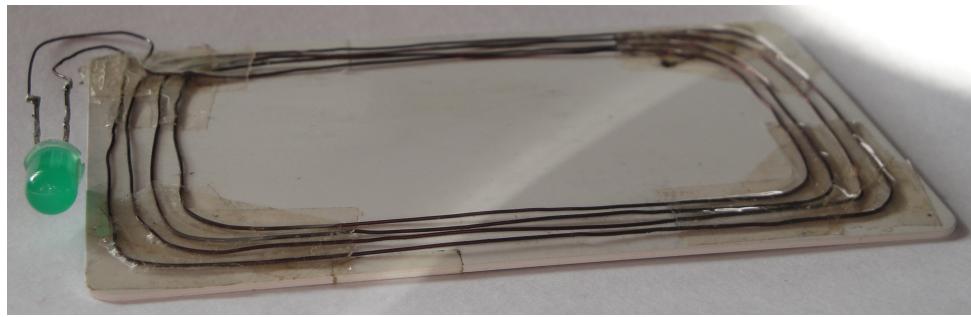


Figura 5.3: Detector de campo magnético casero

Posteriores pruebas usando un osciloscopio en el que una de sus puntas de prueba formaba una espira en lazo cerrado con su línea de tierra, permitió llegar a la misma conclusión que con el detector mencionado antes, no se producía campo magnético en el entorno próximo al inductor de la antena. Una manera poco elegante, pero ingeniosa, de resolver el problema anterior fue desconectar el inductor fabricado en el PCB y colocar en su lugar una bobina de 3 espiras y aproximadamente 10cm de diámetro (ver figura 5.4), hecha a partir de alambre de cobre con aislante incluido. La fabricación de esta bobina se basó en una similar que se puede encontrar en el lector/escritor Proxmark [40] (ver apéndice E).

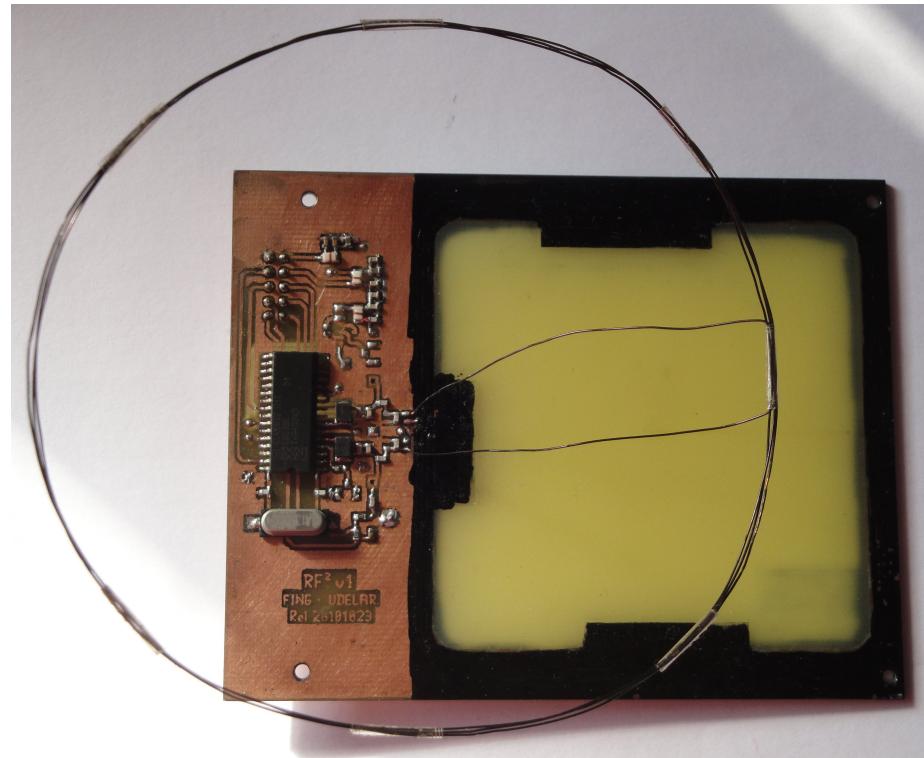


Figura 5.4: Lector/escritor RFID con bobina

Este cambio permitió que la antena resonara a la frecuencia adecuada propagando la señal portadora a 13,56 MHz, siendo posible observar ésta en el osciloscopio además de poder visualizar la forma de los pulsos que se generan en la transmisión de datos desde la antena hacia una tarjeta.

Si se observa la figura 5.5, puede verse en el canal 1 la señal portadora modulada, y en el canal 2 la señal moduladora.

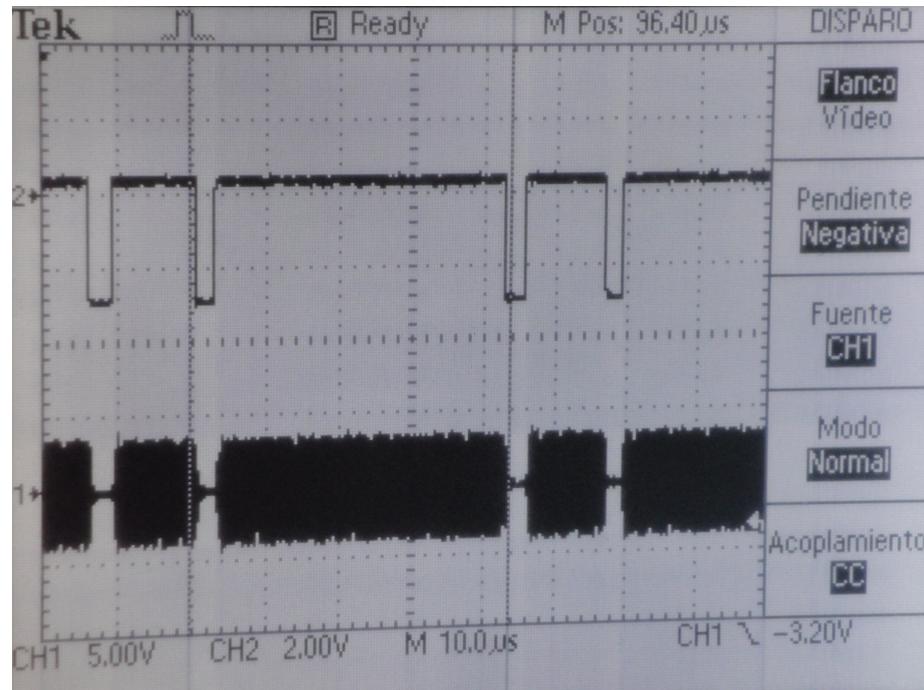


Figura 5.5: Portadora modulada (canal 1) y señal moduladora (canal 2)

Ya sin la incertidumbre a nivel de hardware, era necesario continuar con pruebas a nivel de software para lograr implementar el algoritmo de anticolisión, que permitiera obtener el UID de cada tarjeta que se aproxime al lector/escritor, así como también su autenticación y su posterior lectura y escritura. Mientras se desarrollaban las funciones necesarias, se avanzaba en paralelo en el diseño y la fabricación del conversor de niveles (VLT) que permitiera conectar el lector/escritor RFID directamente sobre la Beagleboard para hacer uso de la biblioteca librfid, que ya contaba con las funciones necesarias y simplificaría todo el desarrollo desde cero.

Para el testeo de la biblioteca librfid; esto es instalación, configuración y ejecución de librfid-tool; se utilizó en primera instancia un lector/escritor OpenPCD. En las primeras pruebas se conectó el OpenPCD directo en el PC de desarrollo para luego pasar a su utilización conectado a la SBC.

Una vez lista la placa VLT, se conectó el lector/escritor RFID directamente sobre la Beagleboard y comenzaron las pruebas de software con la biblioteca antes mencionada. Las primeras pruebas fueron infructíferas, decidiendo continuar en paralelo con las pruebas de software también sobre el microcontrolador Rabbit. Mientras se continua-

ba con las pruebas en software, el hardware también debía ser modificado ya que la solución alcanzada anteriormente no era definitiva sino transitoria; se resolvió entonces diseñar un nuevo lector/escritor con su módulo digital separado de la antena, esto permitiría realizar mediciones sobre ésta última haciendo uso de un analizador de red, este tipo de instrumental es sumamente útil y hasta imprescindible cuando se diseña en RF. Las mediciones se realizaron con el instrumento: "Vector Network Analyzers" de ROHDE&SCHWARZ (LXI Class C conformant) con que cuenta en préstamo el IIE. Por medio del mismo fue posible observar el comportamiento de la impedancia de la antena a medida que varía la frecuencia de trabajo. Para el circuito utilizado en el primer diseño de la antena, la frecuencia de resonancia se presentaba aproximadamente a 18 MHz, es posible observar este detalle en la figura 5.6. Esta frecuencia se encuentra lejos de la frecuencia de trabajo de 13,56 MHz y explica por qué no funcionaba el primer diseño.

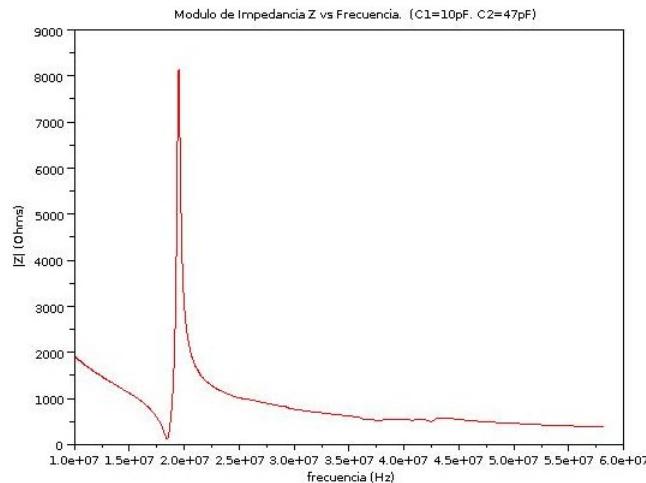


Figura 5.6: Módulo de impedancia z vs. frecuencia ($C_1 = 10\text{pF}$, $C_2 = 47\text{pF}$)

Posteriores modificaciones en el circuito de adaptación de impedancia, incrementando o disminuyendo el valor de los capacitores según el valor de impedancia obtenida, como se indica en [4], permitieron centrar la frecuencia de resonancia en 13,56 MHz como puede observarse en la figura 5.7. Se debe hacer notar que el uso de las ecuaciones para hallar los valores de los capacitores que se encuentran en las notas de aplicación [4] no son válidas, causando que se incurriera en error en la primer versión de la antena

fabricada.

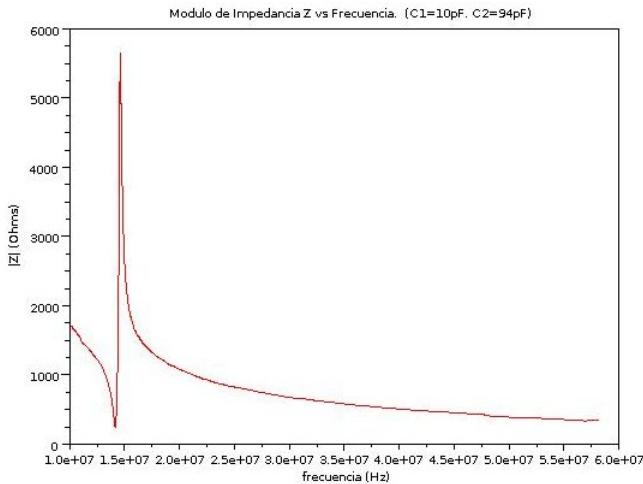


Figura 5.7: Módulo de impedancia z vs. frecuencia ($C_1 = 10\text{pF}$, $C_2 = 94\text{pF}$)

Las nuevas ecuaciones utilizadas se encuentran en el apéndice B, y los valores de los capacitores que se obtienen a partir de éstas, se encuentran próximos a los obtenidos por el método práctico haciendo uso del analizador de red. Restaba sólo incorporar el módulo digital a la antena para tener un lector/escritor RFID que permitiera continuar las pruebas de software. Las pruebas sobre el microcontrolador Rabbit continuaron mientras no se obtenían buenos resultados usando la biblioteca librfid sobre la Beagleboard, fue posible entonces la lectura del UID que posee cada tarjeta obtenido a partir del algoritmo de anticolisión, aunque no fue posible la lectura y/o escritura de las mismas por no lograr una adecuada autenticación. Las pruebas sobre la biblioteca librfid no arrojaron buenos resultados en una primera instancia, se tuvo que hacer algunas modificaciones, en la herramienta librfid-tool, para obtener el funcionamiento adecuado con el hardware fabricado. En primera instancia esta herramienta sólo inicializa el lector OpenPCD, sin tener en cuenta si la croscompilación fue efectuada con opciones para manejo del puerto SPI, incorporar la inicialización del nuevo lector/escritor en esta biblioteca fue el primer paso para hacer uso adecuado del hardware. La siguiente dificultad que se debió enfrentar fue el haber pasado por alto que el pin RSTPD del integrado CL RC632 se encontraba en el estado lógico “1” produciendo un reset permanente, esto impedía que el integrado respondiera a las instrucciones enviadas desde la aplicación. Resuelto lo an-

terior comenzaron las pruebas con la lectura de tarjetas Mifare; la lectura de un bloque de una tarjeta no ofrecía mayores inconvenientes, pero el intento de lectura completa de una tarjeta traía aparejado que no se pudieran leer sectores completos de la misma. Luego de varias pruebas en la configuración de tiempos para la transmisión de datos en el canal RF, sin buenos resultados, se probó otra opción relacionada con la tasa de transferencia de datos utilizada por librfid en el canal SPI, que se establece por defecto en el valor 1MHz. Este valor no era adecuado para lograr una lectura completa de los 64 bloques de una tarjeta Mifare 1K, teniéndose que incrementar a una frecuencia de 10MHz para lograr una lectura completa de una tarjeta en forma válida. Modificaciones adicionales que no revistieron mayores dificultades fueron realizadas sobre librfid-tool con el fin de hacer uso de las tarjetas Mifare que se usan en el sistema de transporte, las que cuentan con claves de autenticación que no son las que vienen por defecto cuando las tarjetas están vírgenes.

Parte IV

Gestión de proyecto

Capítulo 6

Compras y costos

La compra de componentes electrónicos se efectuó casi exclusivamente en empresas de Estados Unidos, entre las que figuran Newark [106], Digikey [107] y Special Computing [108]. Ésto se debió a la dificultad que existe en plaza para conseguir los insumos necesarios para la fabricación del prototipo RF².

Entre los pocos elementos comprados en plaza se encuentran las placas de circuito impreso que fueron fabricadas por la empresa Eneka.

En lo que sigue se muestran las listas de componentes separadas por el módulo de hardware al que pertenecen. Los precios que se detallan son en origen, en dólares americanos y no incluyen impuestos, excepto los PCB que incluyen IVA.

6.1. SBC

Componente	Descripción	Cantidad	Precio x1	Total
SBC	Beagleboard RevC4	1	125	125
Memoria SD	4GB SDHC Class 6 SD Card	1	15	15
Cable serial DB9 nulo	DB9F Null Modem (RS-232) (6-ft)	1	4	4
Cable conversor usb-serial	USB to DB9M RS-232 (PL-2302)	1	10	10
Cable USB	USB Mini-A to USB A Female, OTG	1	9	9
Cable USB	USB Mini-B Male to USB A Male	1	5	5
Fuente	5VDC/2,5A	1	10	10
				178

Cuadro 6.1: Single board computer y lista de accesorios.

6.2. PCBs

Componente	Descripción	Cantidad	Precio x1	Total
VLT	Interfaz entre SBC y SCUI	1	28	28
SCUI	Interfaz de usuario y lector de tarjetas ISO7816	1	53	53
RWD RFID	Lector/Escritor de tarjetas RFID ISO14443	1	64	64
				145

Cuadro 6.2: Lista de placas de circuito impreso.

6.3. VLT

Componente	Descripción	Footprint	Valor	Cantidad	Precio x1	Total
C1	Polarized Capacitor (Tantal)	6032[2312]	10uF, 25V	1	1,09	1,09
C2	Polarized Capacitor (Tantal)	6032[2312]	100uF, 6V3	1	1,16	1,16
U4	Regulador LM1117-3.3	SOT-223	3.3V, 800mA	1	1,1	1,1
U1, U2, U3	Voltage Level Translator	TSSOP20	-	3	2,24	6,72
P1	RECEPTACLE, 28WAY, 2ROW	SMD Pitch 2,54	28 pines	1	4,19	4,19
P2	RECEPTACLE, 40WAY, 2ROW	SMD Pitch 2,54	40 pines	1	4,36	4,36
P1b	HEADER, 28WAY, 2ROW	T H Pitch 2,54	28 pines	1	2	2
P2b	HEADER, 40WAY, 2ROW	T H Pitch 2,54	40 pines	1	1,94	1,94
						22,56

Cuadro 6.3: Lista de componentes de la placa de circuito impreso VLT.

6.4. SCUI

Componente	Descripción	Footprint	Valor	Cantidad	Precio x1	Total
R9	Resistor 100W 1/4W 1 %	3216[1206]	100W 1/4W 1 %	1	0,07	0,07
R10, R13	Resistor 100KW 1/4W 5 %	3216[1206]	100KW 1/4W 5 %	2	0,09	0,18
R11, R12, R14	Resistor 10KW 1/4W 5 %	3216[1206]	10KW 1/4W 5 %	3	0,08	0,24
Q2	TRANSISTOR, NPN, 300MHZ	SOT23	MMBT3904	1	0,125	0,125
Q3	TRANSISTOR, PNP, 250MHZ	SOT23	MMBT3906	1	0,18	0,18
J2	SIM socket (6 contacts)	SMD	-	1	1,25	1,25
JP3, JP4	HEADER, 1ROW, 3WAY	T H Pitch 2,54	3 pines	2	0,11	0,22
X1	Oscillator 3.579545MHz	SMD	3.579545 Mhz	1	5,25	5,25
ESD1	Anti ESD	SOT323	6V / 150W	1	0,45	0,45
						7,965

Cuadro 6.4: Lista de componentes del lector de tarjetas de contacto, SC.

Componente	Descripción	Footprint	Valor	Cantidad	Precio x1	Total
R1	Resistor 4K7 1/10W 1 %	1608[0603]	4,7KW 1/10W 1 %	1	0,05	0,05

R2, R8	Resistor 3R3 1/10W 1 %	1608[0603]	3,3W 1/10W 1 %	2	0,09	0,18
R3, R4, R5	Resistor 680R 1/10W 1 %	1609[0603]	680W 1/10W 1 %	3	0,05	0,15
R6, R7	Resistor 10K 1/10W 1 %	1608[0603]	10KW 1/10W 1 %	2	0,05	0,1
RV1	Preset 15K 1/10W 25 %	SMD	15KW 1/10W 25 %	1	0,71	0,71
Q1	TRANSISTOR, NPN, 300MHZ	SOT23	MMBT3904	1	0,125	0,125
S1	LCD MODULE 16X2 CHARAC- TER	Pitch 2,54	-	1	10,85	10,85
CONN1	HEADER FEMALE 16POS.1”TIN	Through Hole	16 pines	1	1,25	1,25
CONN2	HEADER, 1ROW, 16WAY	T H Pitch 2,54	16 pines	1	0,155	0,155
LED1	Led green 5mm	Through Hole	1,9V, 2mA	1	0,11	0,11
LED2	Led red 5mm	Through Hole	1,9V, 2mA	1	0,1	0,1
LED3	Led yellow 5mm	Through Hole	2,4V, 2mA	1	0,13	0,13
BUZZ1	Buzzer	Through Hole	3 20Vdc, 3 16mA	1	5,31	5,31
						19,17

Cuadro 6.5: Lista de componentes para la interfaz de usuario,
LCD.

6.5. Lector/Escritor RFID

Componente	Descripción	Footprint	Valor	Cantidad	Precio x1	Total
C1, C2	Capacitor	1608[0603]	10pF, Ceramic NPO, 2 %	2	0,135	0,27
C3, C4	Capacitor	1609[0603]	100pF, Ceramic NPO, 2 %	2	0,194	0,388
C5, C6, C7, C8	Capacitor	1608[0603]	NC	4	-	0
R1, R2	Resistor	1608[0603]	0W, 1/10W, 1 %	2	0,015	0,03
						0,688

Cuadro 6.6: Lista de componentes del lector/escritor RFID.

Antena.

Componente	Descripción	Footprint	Valor	Cantidad	Precio x1	Total
C10	Capacitor	1610[0603]	10pF, Ceramic NPO, 2 %	1	0,135	0,135
C1, C2	Capacitor	1608[0603]	15pF, Ceramic NPO, 5 %	2	0,03	0,06
C12, C13	Capacitor	1608[0603]	56pF, Ceramic NPO, 2 %	2	0,194	0,388
C14, C15	Capacitor	1608[0603]	68pF, Ceramic NPO, 1 %	2	0,197	0,394
C9	Capacitor	1609[0603]	100pF, Ceramic NPO, 2 %	1	0,194	0,194
C16	Capacitor	1608[0603]	1nF, Ceramic NPO, 10 %	1	0,08	0,08

C4, C5, C7, C8, C11, C17	Capacitor	1608[0603]	100nF, Ceramic X7R, 10 %	6	0,074	0,444
C3, C6, C18	Capacitor	1608[0603]	10uF, Ceramic X5R, 20 %	3	0,195	0,585
L1, L2, L3, L6	Inductor	2012[0805]	22nH, 700mA, 5 %	4	0,454	1,816
L4, L5	Inductor	3225[1210]	1uH, 400mA, 5 %	2	0,29	0,58
R3	Resistor	1608[0603]	50W, 1/10W 1 %	1	0,268	0,268
R2	Resistor	1608[0603]	820W, 1/10W 5 %	1	0,027	0,027
R1	Resistor	1608[0603]	2,2KW, 1/5W 1 %	1	0,08	0,08
U1	Reader ISO14443	SO32	CL RC632	1	14,22	14,22
U2	Crystal Oscillator, HC49 US SMD	49USMXL	13.56MHz, 10pF	1	0,98	0,98
U3	Operational Amplifier (up to 7.5V)	SOT23-5	OPA354	1	2,8	2,8

CONN1, CONN2	U.FL-R Con- nector	U.FL-R- SMT	-	2	1,76	3,52
J1	HEADER, 10WAY, 2ROW	T H Pitch 2,54	10 pines	1	0,389	0,389
J1b	RECEPTACLE, 10WAY, 2ROW	SMD Pitch 2,54	10 pines	1	2,27	2,27
						29,23

Cuadro 6.7: Lista de componentes del lector/escritor RFID.

En cuanto al costo de fabricación de los PCB, se puede decir que las diferencias en precio son sustanciales, dependiendo del origen del fabricante. Cuando se habla de enviar a fabricar las placas en el mercado local se está hablando de algunas decenas de dólares por PCB, mientras que enviarlas a fabricar en el exterior como ser en China o Malasia, las mismas placas cuestan entre tres y seis dólares para compras de 20 unidades. En las tablas siguientes se puede apreciar la diferencia de costo al fabricar un prototipo con placas hechas en Uruguay y con placas hechas en el exterior.

PCB Uruguay		PCB China	
Descripción	Precio (U\$)	Descripción	Precio (U\$)
SBC + Fuente + memoria SD	150	SBC + Fuente + memoria SD	150
Conversor de nivel VLT	50,56	Conversor de nivel VLT	25,16
Módulo SCUI	80,14	Módulo SCUI	32,54
Lector/escritor RFID	93,92	Lector/escritor RFID	33,88
Total	374,62	Total	241,58

Cuadro 6.8: Comparación del costo del prototipo

El costo total del proyecto fue solventado por los integrantes del grupo y asciende aproximadamente a 1385 dólares, habiendo estimado un gasto total de 1500 dólares.

Capítulo 7

Tiempos

Por lo general los proyectos no siguen estrictamente la planificación inicial, éste no fue la excepción a la regla. Varias son las causas que llevaron a no cumplirse con el cronograma establecido, algunas de ellas tenidas en cuenta durante el análisis de riesgos y otras impensadas desde todo punto de vista. Entre estas últimas se encuentran las placas Hawkboard defectuosas, este imprevisto absolutamente infortunio forzó una replanificación del cronograma, siendo además el factor preponderante al momento de solicitar prórroga. Lo anterior trajo aparejado el uso de una nueva SBC, que a su vez tuvo repercusión en el diseño hardware que se tenía hasta ese momento, nuevos componentes de hardware tuvieron que ser estudiados para compatibilizar los cambios necesarios.

El retraso alcanzado a causa de lo ya explicado rondó los tres meses.

Sumado a lo que se ha descrito antes, hubo un atraso de alrededor de un mes desde el momento que se seleccionó la SBC GESBC-9G20 hasta que se decidió utilizar la SBC Hawkboard, por falta de respuesta del soporte técnico de la primera. El otro camino crítico que tuvo que enfrentarse estuvo asociado al diseño de la antena RFID, las notas teóricas suministradas por el mismo fabricante del chip contenían errores en los cálculos de algunos componentes que conforman el circuito, provocando que la fecha indicada para la culminación de las pruebas de hardware de este componente se vieran pospuestas por tres meses aproximadamente.

Entre las causas que se previeron en el análisis de riesgos, se encuentra el financiamiento del proyecto por parte de los integrantes del grupo, esto debido a que la parte

interesada, IM, no encontró el mecanismo de compra de componentes e insumos que no pueden adquirirse en el mercado local, sus trabas burocráticas impiden que se manejen tiempos razonables para un proyecto de estas características.

Aún habiendo solicitado una prórroga de tres meses los plazos se vencieron sin haber culminado el total de las tareas previstas como ser la integración del lector de tarjetas de contacto a la lista de dispositivos soportados por la biblioteca pcsclite.

En lo que sigue se detalla una tabla con las tareas realizadas.

	Planificación Teórica		Planificación Real		Fechas Importantes
Tareas	inicio	fin	inicio	fin	
Definir arquitectura	13/05/10	29/05/10	13/05/10	21/05/10	
Definir SBC	31/05/10	05/06/10	31/05/10	20/07/10	
Primera compra					29/08/10
Hito 1					15/09/10
Cambio de SBC					12/11/10
Segunda compra					10/12/10
Hito 2					15/02/11
Estudio y diseño hardware lector/escritor RFID	26/06/10	28/12/10	05/06/10	31/03/11	
Tercera compra					05/04/11
Estudio y diseño hardware lector tarjetas de contacto	19/06/10	28/12/10	02/07/10	15/06/11	
Diseño hardware prototipo final					18/06/11
Adaptar código de lector/escritor RFID	05/04/11	12/05/11	08/03/11	13/06/11	
Cuarta compra					05/07/11

Comunicación con tarjeta de contacto (software)	25/04/11	16/05/11	16/05/11	16/07/11	
Interfaz de usuario (software)	13/05/11	27/05/11	13/05/11	04/06/11	
Integración de módulos (software)	27/05/11	20/06/11	13/06/11	16/07/2011 (*)	
Programa base para demo	01/07/11	10/07/11	13/06/11	30/06/11	
Pruebas del sistema	20/06/11	30/06/11	20/06/11	16/07/11	
Documentación final	11/07/11	31/07/11	18/07/11	15/08/11	

Cuadro 7.1: Descripción de tareas más importantes

*no terminado.

Parte V

Conclusiones

Capítulo 8

Conclusiones

8.1. Conclusión final

El mundo de la tecnología RFID está poco explorado en nuestro país, éste tal vez sea el primer proyecto que incluye el diseño y fabricación de un lector/escritor RFID capaz de operar con tarjetas sin contacto en la banda de frecuencia de 13,56 MHz.

Las posibilidades de aplicación son muy bastas, entre las que se pueden nombrar, sistemas de transporte, pasaporte electrónico, identificación civil, tarjetas de crédito y/o débito, salud, telefonía, control de acceso, entre otras.

El aporte realizado en este campo es tan solo una primera aproximación y aún queda mucho por hacer al respecto. Sobre este proyecto en particular es necesario mejorar varios aspectos antes de pasar de la fase de prototipo a la de producción.

Repasando en particular los criterios de éxito, el proyecto ha resultado satisfactorio porque se logró construir un dispositivo capaz de consultar y recargar tarjetas RFID, aunque la solución alcanzada no sea estrictamente igual a la propuesta en el comienzo.

En lo que tiene que ver con el módulo de seguridad, SAM, no fue posible su empleo por no lograr integrar el lector de tarjetas de contacto con la biblioteca pcselite. La función de derivar las claves, suministradas por la SAM, fue sustituída agregándolas directamente en el código de la aplicación final. En cuanto a la comunicación segura con un servidor establecida por la SAM, se descartó su uso en una de las replanificaciones de las tareas a cumplir.

El enfoque de este proyecto está dirigido al uso en un sistema de transporte, pero puede ser empleado en cualquiera de las aplicaciones ya mencionadas sin necesidad de hacer grandes modificaciones a nivel de software.

En suma, el equipo está conforme con el trabajo realizado.

Para finalizar, se sugiere que el Instituto de Ingeniería Eléctrica, y el Instituto de Computación, se integren en proyectos que involucren hardware que permite portar sistemas operativos como ser Linux (de los más usados en sistemas embebidos).

8.2. Ventajas y desventajas

La ventaja principal de este prototipo comparada con el sistema que se usa hoy en día mediante computadoras de escritorio, es su considerable menor consumo de energía, factor a tener en cuenta en un sistema 24/7 como es el transporte de pasajeros. La otra gran ventaja es que no incorpora una impresora de ticket, esto reduce el tamaño del prototipo así como disminuye su costo de mantenimiento en todo lo que tenga que ver con cambio de rollos de papel, atasco del mismo y cambio de piezas de la impresora. Los puntos mencionadas antes van de la mano con el tema ecología, tan en boga en estos días y por lo que se hace muy poco.

Entre las ventajas que se pueden hallar en este dispositivo es que el lector/escritor de tarjetas RFID es un diseño realizado en PCB de dos capas que lo hace más sencillo y económico que el diseño del lector/escritor OpenPCD, y al igual que este último es compatible con la biblioteca open source conocida como librfid.

Por su lado, en el lector de tarjetas de contacto se debe destacar su simplicidad, ya que no cuenta con ningún tipo de hardware específico (ASIC) que cumpla con el estandar ISO7816, sólo es necesario tener disponible un puerto serial (UART) y un par de puertos de entrada/salida de propósito general. Esto lo hace portable a cualquier SBC que cuente con los puertos detallados anteriormente.

Algo que no se tiene en cuenta muchas veces es qué tan simple puede resultar el armado de las partes del dispositivo, en este caso se pensó en un método simple y rápido donde no se emplearan cables para los conexionados. El ensamblado entre sí de las

placas de hardware que conforman el prototipo es una tarea simple que puede ser hecha por un niño, como si armara su mecano. Las distintas partes encajan una encima de otra y son aseguradas a través de un separador con tuerca, no existiendo posibilidad que sean conectadas al revés.

8.3. Aprendizaje

Los conceptos aprendidos sobre reglas de diseño de lectores/escritores de tarjetas sin contacto, compatibles con la norma ISO14443, son muy importantes para llevar a cabo mejoras y modificaciones futuras. Las dificultades afrontadas durante la fase inicial, generadas por errores en las ecuaciones suministradas por una de las notas de aplicación usada, y la falta de soporte técnico por parte del fabricante (solamente brindan soporte a empresas), permitió ver la necesidad del uso de instrumentos de medición adecuados para validar el diseño y comparar con los resultados teóricos. La moraleja en cuanto a los errores cometidos en la etapa de diseño es, el ser cautos a la hora de confiar 100 % en las publicaciones que son usadas como referencia, no debiendo ser tomadas como dogmas.

El manejo de herramientas de diseño CAD, fue de mucha utilidad para luego llevar adelante la fabricación de las placas de circuito impreso, se aprendió a utilizar no sólo una, sino dos aplicaciones ya que una de ellas (KiCad) no permite trazar pistas en forma de curva, cosa necesaria para el diseño del lector/escritor RFID.

A nivel de software se dejó de ser un simple usuario del sistema operativo Linux, adquiriendo conocimientos para croscompilar su Kernel, e incorporar modificaciones y parches que permitieran poner en funcionamiento el hardware que conforma el prototípo. Se debió entender el funcionamiento de buena parte de las bibliotecas usadas, para modificarlas y que fuera posible su uso en la aplicación final.

La elección de la herramienta de desarrollo es muy importante para obtener buenos resultados.

En cuanto al cliente, se entendió que maneja tiempos y prioridades distintas. Éstas últimas podrían cambiar durante el transcurso de un proyecto, por lo que no podía ser

tenido en cuenta como cliente. La IM pasó a ser un actor que suministró ambos tipos de tarjetas RFID y SAM, y el dispositivo OpenPCD.

Se aprendió que los mecanismos de compra en la IM son muy complejos y no tienen flexibilidad a la hora de hacer compras en el exterior.

Por último, pero no menos importante, lo que parece salir en dos días puede demorar dos meses por lo que es recomendable al planificar sobredimensionar tiempos previéndolo, además de calcular un período con el cual poder moverse con holgura si existen inconvenientes extra.

8.4. Mejoras y trabajos a futuro

Entre las cosas que se deben investigar, es que el blindaje incluido en la primer antena fabricada cumpla con las regulaciones de compatibilidad electromagnética, EMC, definidas por la EN de Europa o por la FCC de Estados Unidos. A tales efectos sería necesario contar con el equipamiento adecuado para efectuar las mediciones necesarias. Tal vez la disminución del tamaño del PCB de la antena, o la incorporación de ésta al resto del hardware, puedan ser vistas como mejoras desde el punto de vista de disminución de costos.

En cuanto al lector de tarjetas de contacto, su incorporación a la lista de lectores soportados por la biblioteca pcselite quedó inconclusa; si bien el hardware está operativo y es posible enviar comandos APDU al módulo SAM de seguridad a través de una pequeña aplicación, sería importante alcanzar el objetivo planteado al comenzar el proyecto. Un aspecto importante a destacar es que no se detallan los comandos APDU empleados, ya que por motivos de seguridad los mismos no pueden ser revelados.

Se podría integrar la comunicación del prototipo RF² con un servidor, que gestione todo lo relacionado con las transacciones entre los dispositivos y las tarjetas.

En cuanto a diseño industrial faltaría diseñar y fabricar una carcasa acorde, que permita el anclaje rápido y seguro de las distintas partes de hardware, sin que el material del cual esté hecha interfiera con la propagación de radio frecuencia, esto descarta la posibilidad de usar metal como elemento a emplear.

Algo no previsto en el prototipo, es el hecho de que ocurra un corte de energía. Puede solucionarse incluyendo un sistema con una batería, que se active al detectar la falta de energía externa, capaz de alimentar el dispositivo hasta terminar una posible transacción en curso y luego ingrese en modo fuera de línea.

Todos los elementos que forman parte del software del sistema se encuentran almacenados en la memoria SD, sin embargo este tipo de memoria es de menor calidad que la NAND Flash con la que cuenta la SBC utilizada, y por tanto aumenta la posibilidad de fallas, hecho que ocurrió con una de las SD al tener varios bloques de almacenamiento defectuosos. El inconveniente anterior, y otros asociados con temas mecánicos como vibraciones (si el dispositivo se instala en un vehículo) o suciedad en los contactos, etc., podrían evitarse si los archivos son almacenados directamente en memoria Flash.

Un detalle a mejorar en la inicialización del lector/escritor RFID es que verifique que realmente se trata de este dispositivo y no de otro que se encuentre conectado al puerto SPI. Esto puede hacerse efectuando una lectura del identificador de producto desde la memoria EEPROM del integrado CL RC632.

Actualmente existe la versión 3.0 de kernel, se podría probar migrar de la versión que se usó (2.6.32) a la más actual.

Parte VI

Apéndices

Apéndice A

Tarjetas inteligentes (Smart Cards)

Una tarjeta inteligente (smart card), o tarjeta con circuito integrado (ICC, de su sigla en inglés), es cualquier tarjeta del tamaño de un bolsillo con circuitos integrados que permiten la ejecución de cierta lógica programada. Aunque existe un diverso rango de aplicaciones, hay dos categorías principales de ICC. Las tarjetas de memoria contienen sólo componentes de memoria no volátil y posiblemente alguna lógica de seguridad. Las tarjetas microprocesadoras contienen memoria y microprocesador. La percepción estándar de una smart card es una tarjeta microprocesadora de las dimensiones de una tarjeta de crédito (o más pequeña, como por ejemplo, tarjetas SIM para GSM) con varias propiedades especiales (ej. un procesador criptográfico seguro, sistema de archivos seguro, características legibles por humanos) y es capaz de proveer servicios de seguridad (ej. confidencialidad de la información en la memoria). Las tarjetas no contienen baterías; la energía es suministrada por los lectores de tarjetas.

A.1. Clasificaciones

A.1.1. Tipos de tarjetas según su capacidad

Según las capacidades de su chip, las tarjetas más habituales son:

- Memoria: tarjetas que únicamente son un contenedor de ficheros pero que no albergan aplicaciones ejecutables. Por ejemplo, MIFARE. Éstas se usan general-

mente en aplicaciones de identificación y control de acceso sin altos requisitos de seguridad.

- Microprocesadas: tarjetas con una estructura análoga a la de una computadora (procesador, memoria volátil, memoria persistente). Éstas albergan ficheros y aplicaciones y suelen usarse para identificación y pago con monederos electrónicos.
- Criptográficas: tarjetas microprocesadas avanzadas en las que hay módulos hardware para la ejecución de algoritmos usados en cifrados y firmas digitales. En estas tarjetas se puede almacenar de forma segura un certificado digital (y su clave privada) y firmar documentos o autenticarse con la tarjeta sin que el certificado salga de la misma, ya que es el procesador de la propia tarjeta el que realiza la firma.

A.1.2. Tipos de tarjetas según la estructura de su sistema operativo

- Tarjetas de memoria. Tarjetas que únicamente son un contenedor de datos pero que no albergan aplicaciones ejecutables. Disponen de un sistema operativo limitado con una serie de comandos básicos de lectura y escritura de las distintas secciones de memoria y pueden tener capacidades de seguridad para proteger el acceso a determinadas zonas de memoria.
- Basadas en sistemas de ficheros, aplicaciones y comandos. Estas tarjetas disponen del equivalente a un sistema de ficheros compatible con el estándar ISO/IEC 7816 parte 4 y un sistema operativo en el que se incrustan una o más aplicaciones (durante el proceso de fabricación) que exponen una serie de comandos que se pueden invocar a través de API de programación.
- Java Card. Tarjeta capaz de ejecutar mini-aplicaciones Java. En este tipo de tarjetas el sistema operativo es una pequeña máquina virtual Java (JVM) y en ellas se pueden cargar dinámicamente aplicaciones desarrolladas específicamente para este entorno.

A.1.3. Tipos de tarjetas según el formato (tamaño)

En el estándar ISO/IEC 7816 parte 1 se definen los siguientes tamaños para tarjetas inteligentes:

- ID 000: el de las tarjetas SIM usadas para teléfonos móviles GSM. También acostumbran a tener este formato las tarjetas SAM (Security Access Module) utilizadas para la autenticación criptográfica mutua de tarjeta y terminal.
- ID 00: un tamaño intermedio poco utilizado comercialmente.
- ID 1: el más habitual, tamaño tarjeta de crédito.
- ID 1/000: permite remover la tarjeta ID 000 desde la tarjeta ID 1 sin herramientas de corte.

A.1.4. Tipos de tarjetas según la interfaz

Tarjeta inteligente de contacto

Estas tarjetas disponen de contactos metálicos visibles y debidamente estandarizados (parte 2 de la ISO/IEC 7816). Estas tarjetas, por tanto, deben ser insertadas en una ranura de un lector para poder operar con ellas. A través de estos contactos el lector alimenta eléctricamente a la tarjeta y transmite los datos oportunos para operar con ella conforme al estándar.



Figura A.1: Tarjeta de contacto

La serie de estándares ISO/IEC 7816 e ISO/IEC 7810 definen:

- La forma física (parte 1).
- La posición de las formas de los conectores eléctricos (parte 2).

- Las características eléctricas (parte 3).
- Los protocolos de comunicación (parte 3).
- El formato de los comandos (ADPU's) enviados a la tarjeta y las respuestas retornadas por la misma (parte 3).
- La dureza de la tarjeta.
- La funcionalidad.

Tarjetas Inteligentes sin Contacto

El segundo tipo es la tarjeta inteligente sin contacto, RFID, en el cual el chip se comunica con el lector de tarjetas mediante acoplamiento magnético a una tasa de transferencia de 106 a 848 Kbits/s. El estándar de comunicación de tarjetas inteligentes sin contacto es el ISO/IEC 14443. Define dos tipos de tarjetas sin contacto (A y B), permitidos para distancias de comunicación de hasta 10cm. Las más abundantes son las tarjetas de la familia MIFARE de Philips, las cuales representan a la ISO/IEC 14443-A. Las tarjetas inteligentes sin contacto son una evolución de la tecnología usada desde hace años por los RFID (identificación por radio frecuencia), añadiéndoles dispositivos que los chip RFID no suelen incluir, como memoria de escritura o microcontroladores.

Tarjetas híbridas y duales

Una tarjeta híbrida es una tarjeta sin contacto (contactless) a la cual se le agrega un segundo chip de contacto. Ambos chips pueden ser chips microprocesadores o simples chips de memoria. El chip sin contacto es generalmente usado en aplicaciones que requieren transacciones rápidas. Por ejemplo el transporte, mientras que el chip de contacto es generalmente utilizado en aplicaciones que requieren de alta seguridad como las bancarias.

Seguridad

La seguridad es una de las propiedades más importantes de las tarjetas inteligentes y se aplica a múltiples niveles y con distintos mecanismos. Cada fichero lleva asociadas unas condiciones de acceso y deben ser satisfechas antes de ejecutar un comando sobre ese fichero.

En el momento de personalización de la tarjeta (durante su fabricación) se puede indicar que mecanismos de seguridad se aplican a los ficheros. Normalmente se definirán:

- Ficheros de acceso libre.
- Ficheros protegidos por claves: Pueden definirse varias claves con distintos propósitos. Normalmente se definen claves para proteger la escritura de algunos ficheros y claves específicas para los comandos de consumo y carga de las aplicaciones de monedero electrónico. De ese modo la aplicación que intente ejecutar comandos sobre ficheros protegidos tendrá que negociar previamente con la tarjeta la clave oportuna.
- Ficheros protegidos por PIN: El PIN es un número secreto que va almacenado en un fichero protegido y que es solicitado al usuario para acceder a este tipo de ficheros protegidos. Cuando el usuario lo introduce y el programa se lo pasa a la operación que va a abrir el fichero en cuestión, el sistema valida que el PIN sea correcto para dar acceso al fichero.

La negociación de claves se realiza habitualmente apoyándose en un Módulo SAM, que no deja de ser otra tarjeta inteligente en formato ID-000 alojada en un lector interno propio dentro de la carcasa del lector principal o del TPV(Terminal de Punto de Venta) y que contiene aplicaciones criptográficas que permiten negociar las claves oportunas con la tarjeta inteligente del usuario. Operando de este modo se está autenticando el lector, la tarjeta y el módulo SAM involucrados en cada operación.

Programación de aplicaciones para los sistemas en los que se utiliza la tarjeta

Existen varias API de programación estandarizadas para comunicarse con los lectores de tarjetas inteligentes desde un computador. Las principales son:

- PC/SC (Personal Computer/Smart Card). El proyecto MUSCLE proporciona una implementación casi completa de esta especificación para los sistemas operativos GNU Linux-UNIX.

- OCF (OpenCard Framework), especificado por el grupo de empresas OpenCard. Este entorno intenta proporcionar un diseño orientado a objetos fácilmente extensible y modular. El consorcio OpenCard publica el API y proporciona una implementación de referencia en Java. Existe un adaptador para que OCF trabaje sobre PC/SC.

En ambos casos, el modelo de programación que utilizan las tarjetas inteligentes está basado en protocolos de petición-respuesta. La tarjeta (su software) expone una serie de comandos que pueden ser invocados. Estos comandos interactúan con los ficheros que subyacen a cada aplicación de la tarjeta y proporcionan un resultado. Desde el terminal se invocan estos comandos a través de cualquiera de las API antes descritas componiendo un APDU (Application Protocol Data Unit - comandos con parámetros) que son enviados a la tarjeta para que ésta responda.

A.2. ISO 14443

ISO 14443 es un estándar internacional relacionado con las tarjetas de identificación electrónicas, en especial las smart cards, gestionado conjuntamente por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC). Este estándar define una tarjeta de proximidad utilizada para identificación y pagos que por lo general utiliza el formato de tarjeta de crédito definida por ISO 7816 - ID 1 (aunque otros formatos son posibles). El sistema RFID utiliza un lector con un microcontrolador o ASIC y una antena que opera a 13,56MHz (frecuencia RFID). El lector mantiene a su alrededor un campo electromagnético de modo que al acercarse una tarjeta al campo, ésta se alimenta eléctricamente de esta energía inducida y puede establecerse la comunicación lector-tarjeta.

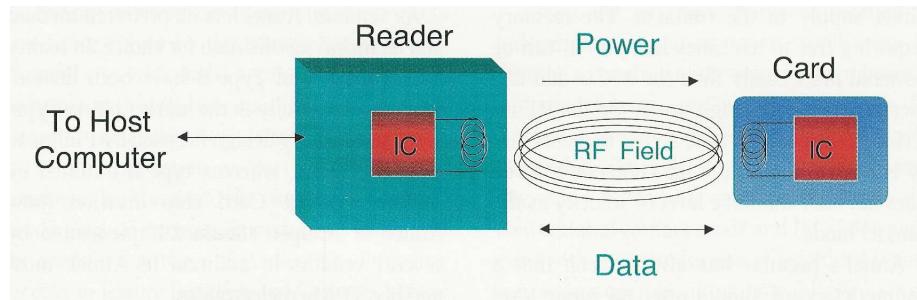


Figura A.2: Acoplamiento entre lector y tarjeta

El estándar ISO 14443 consta de cuatro partes y se describen dos tipos de tarjetas: tipo A y tipo B. Las principales diferencias entre estos tipos están en los métodos de modulación, codificación de los planes (parte 2) y el protocolo de inicialización de los procedimientos (parte 3). Las tarjetas de ambos tipos (A y B) utilizan el mismo protocolo de alto nivel (llamado T=CL) que se describe en la parte 4. El protocolo T=CL especifica los bloques de datos y los mecanismos de intercambio:

1. Bloque de datos de encadenamiento.
2. Tiempo de espera de extensión.
3. Múltiple activación.

Las tarjetas Mifare cumplen con las partes 1, 2 y 3 de tipo A de la especificación ISO/IEC 14443.

A.3. Mifare

Mifare es la tecnología de smart card sin contacto más ampliamente usada en el mundo. Es equivalente a las 3 primeras partes de la norma ISO 14443 Tipo A. La distancia típica de lectura es de hasta 10 cm, depende de la potencia del lector y factores del entorno, existiendo lectores de mayor y menor alcance. La tecnología Mifare es económica y rápida, razón por la cual es la más usada a nivel mundial hoy día.

A.3.1. Operación

Las tarjetas Mifare son tarjetas de memoria protegida. Están divididas en sectores que a su vez son subdivididos en bloques y poseen mecanismos de seguridad para el control de acceso. Su capacidad de cómputo no permite realizar operaciones criptográficas o de autenticación mutua de alto nivel, estando principalmente destinadas a monederos electrónicos simples, control de acceso, tarjetas de identidad corporativas, tarjetas de transporte urbano o para ticketing. Cada sector se divide en cuatro bloques, de los cuales tres pueden contener información del usuario, y el cuarto, llamado trailer, contiene elementos de seguridad. La información es almacenada sin un formato pre establecido, y se puede modificar con comandos simples de lectura y escritura. Mifare provee un formato especial llamado “bloque de valor”(value block); los bloques que tienen información guardada en este formato se comportan de una forma diferente, incluyendo operaciones de incremento y descuento. Los sectores utilizan dos claves de acceso llamadas “A” y “B”. Estas claves se almacenan en el cuarto bloque junto con los permisos de acceso a cada uno de los tres bloques que son parte del mismo sector. Estos permisos pueden ser: lectura, escritura, descuento o incremento (para bloques de valor).

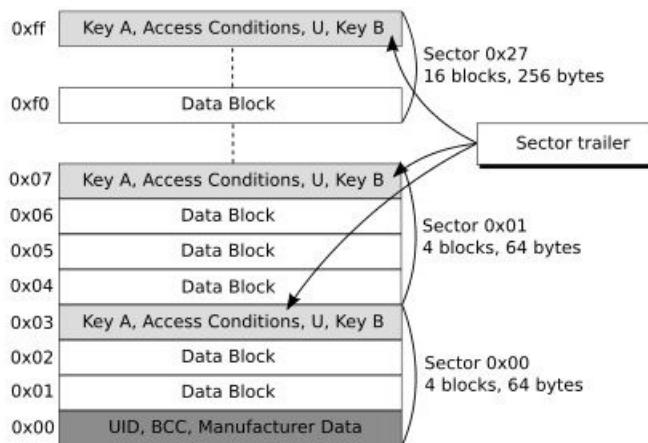


Figura A.3: Mifare Classic de 4K

Una vez que se acerca la tarjeta a un lector, ésta se activa e inicia un proceso de intercambio con el lector para establecer una comunicación cifrada. Este proceso es igual con todas las tarjetas y está diseñado para proveer protección del canal (evitando que

se espíe), y no para autenticar la tarjeta o el lector. Previo a establecer un canal cifrado la tarjeta envía un código de identificación, mediante el algoritmo de anticolisión, que usualmente es el número de serie de la tarjeta, aunque la norma ISO 14443 indica que este número puede ser aleatorio. Con este número de identificación el lector está en condiciones de realizar cualquier operación en la tarjeta, previa autenticación con las claves de acceso en los respectivos sectores. Se debe destacar que un sistema con claves diversificadas facilita el fortalecimiento de la seguridad, apoyada por una base de datos que pueda monitorear los aumentos de los saldos y demás estrategias operativas y finalmente la autenticación remota por SAM, no todos los sistemas poseen esto.

Variantes

- Mifare Classic. Son fundamentalmente de los dispositivos de almacenamiento de memoria. Existen tarjetas de 1Kb y de 4Kb. La Mifare Standard de 1KB ofrece unos 768 bytes de almacenamiento de datos, dividida en 16 sectores. La Mifare Standard de 4k ofrece 3 KB dividido en 64 sectores.
- Mifare Ultralight. Es semejante a la classic, pero sólo tiene 512 bits de memoria (es decir 64 octetos), sin seguridad. Esta tarjeta es muy barata así que se utiliza a menudo de forma desechable.
- Mifare T = CL. Bajo esta denominación se encuadran las tarjetas Mifare ProX y SmartMX. Son tarjetas con microprocesador que incorporan un sistema operativo de tarjeta (Card Operating System - COS) y aplicaciones desarrolladas específicamente para ser ejecutadas en la tarjeta. Estas tarjetas son capaces de ejecutar operaciones complejas de forma rápida y segura, igual que las tarjetas con contactos ISO 7816.
- Mifare DESFire. Esta tarjeta es una versión especial de Philips SmartMX. Se vende con un software de propósito general incorporado (el sistema operativo DESFire), que ofrece más o menos las mismas funciones que Mifare Standard (4kB de almacenamiento de datos dividido en 16 bloques), pero con una mayor flexibilidad, una mayor seguridad (triple DES), y con mayor rapidez (protocolo T=CL).

- Mifare DESFire EV1. Es la primera evolución de Mifare DESFire, compatible con la versión anterior, pero aún más segura, alcanzando la certificación EAL 4.

Información extraída de [20] [55].

Apéndice B

Lector/Escritor RFID

B.1. Reglas y Parámetros de Diseño de una Antena RF

Pasos para el diseño de la antena RF:

- A) Diseñar el inductor, medir su inductancia L y resistencia R (o factor de calidad Q).
- B) Calcular los capacitores para el circuito resonante, que forman junto con el inductor.
- C) Sintonizar el circuito resonante junto con el filtro pasa bajo a la impedancia requerida.
- D) Conectar el circuito resonante a la salida del integrado(TX1 y TX2), verificar la corriente ITVDD y si es necesario sintonizar los componentes para un desempeño óptimo.
- E) Verificar y ajustar el factor de calidad Q.
- F) Verificar y ajustar el circuito receptor.

B.1.1. Diseño del inductor

Se recomienda usar antenas cuyo inductor tenga forma circular o cuadrado. El valor exacto del inductor es difícil de calcular pero puede ser aproximado por la siguiente ecuación:

$$L_1[nH] = 2 \cdot l_1 \cdot (\ln(\frac{l_1}{D_1}) - K) \cdot N_1^{1,8}$$

donde:

l_1 Longitud de una vuelta del conductor (en cm).

D_1 Diámetro del conductor o ancho del conductor del PCB.

K Factor de forma ($K = 1,07$ antena circular y $K = 1,47$ antena cuadrada).

N_1 Número de vueltas.

La antena debe ser simétrica, donde el punto central puede estar conectado a GND. Si esto es así, se sugiere mantener este punto lo más cercano posible al conector de la antena. El radio de la antena deberá ser $r > 5cm$ para una sola vuelta de cada uno de los inductores simétricos L_a y L_b , o $r < 5cm$ para dos vueltas. El blindaje de campo eléctrico debe ser conectado a GND.

Los valores de inductancia y resistencia del inductor, si se siguen las reglas de diseño, se encuentran entre:

$$L = 300nH \dots 2\mu H$$

$$R_{coil} = 0,5\Omega \dots 5\Omega$$

Resistor externo

En serie con el inductor se agrega un resistor externo. El valor del mismo se encuentra mediante las ecuaciones:

$$Q = \frac{wL}{R_{coil}} \rightarrow R_{coil} = \frac{wL}{Q}$$

$$R = 2 \cdot R_s + R_{coil} = R_{Sa} + R_{Sb} + R_{coil}$$

donde R es la resistencia total, R_{Sa} y R_{Sb} son cada uno de los resistores externos simétricos.

Definiendo Q entre 20 y 30, se pueden hallar los resistores externos:

$$R_{Sa} = R_{Sb} = \frac{1}{2} \cdot (R - R_{coil}) = \frac{wL}{2Q} - \frac{R_{coil}}{2} \text{ con } w = 2\pi \cdot 13,56MHz$$

Dejando de lado la influencia de todos los otros componentes en el factor Q , este cálculo sólo da una estimación del valor de R_S , pero esta estimación es necesaria para el cálculo de los capacitores del circuito de resonancia.

En muchas aplicaciones prácticas es posible observar que se prescinde del valor de resistencia externa R_S , teniendo en cuenta sólo el valor de resistencia del inductor R_{coil} .

B.1.2. Capacitores del circuito resonante

Gracias a la simetría del circuito es posible simplificar los cálculos operando sólo con la mitad del circuito, por tanto los valores del inductor L , la resistencia R (incluyendo el resistor externo) y el valor de impedancia requerida de la antena Z_{ant} , empleados en los siguientes cálculos son la mitad del valor correspondiente a la totalidad del circuito. Aplicando entonces la suma de impedancias a la mitad del circuito y sabiendo que el resultado tiene que ser real e igual a Z_{ant} , es posible hallar los valores del capacitor paralelo C_2 y el capacitor serie C_1 mediante las siguientes igualdades:

$$C_{2a} = C_{2b} = \frac{L}{\omega^2 L^2 + R^2} - \frac{R}{(\omega^2 L^2 + R^2)\omega} \sqrt{\frac{Z_a}{\left(\frac{\omega^2 L^2 + R^2}{R} - Z_a\right)}}$$

$$C_{1a} = C_{1b} = \frac{1}{\omega \sqrt{Z_a \cdot \left(\frac{\omega^2 L^2 + R^2}{R} - Z_a\right)}}$$

$$Z_{ant} = 250\Omega$$

El valor de $Z = 2 \cdot Z_{ant} = 500\Omega$ podría ser incrementado hasta $Z = 2 \cdot Z_{ant} = 800\Omega$ para incrementar la potencia de salida, pero el límite de corriente de salida desde el integrado no debe ser excedido.

B.1.3. Sintonizar el circuito resonante

El circuito todo (incluyendo el filtro pasa bajos) tiene que ser adaptado a una impedancia de aproximadamente 40Ω entre TX1 y TX2 (500Ω si no tenemos en cuenta el filtro). Donde los valores propuestos para los componentes del filtro son:

$$L_0 = 1\mu H \text{ (e.g. TDK NL322522T-1R0J)}$$

$$C_{01} = 68pF \text{ (Ceramic NP0, tolerance } \leq \pm 2\% \text{)}$$

$$C_{02} = 56pF \text{ (Ceramic NP0, tolerance } \leq \pm 2\% \text{)}$$

Con estos valores, la frecuencia de resonancia del filtro se encuentra centrada en $14,4MHz(13,56MHz + 847,5KHz)$. Esto mejora la performance en dos formas:

- a) Incrementa la relación señal a ruido de la señal recibida.
- b) Decrementa el sobretiro de los pulsos transmitidos, mejorando la calidad de la señal transmitida.

El procedimiento para sintonizar el circuito, es el siguiente:

Los materiales necesarios son:

- Generador de señales ($13,56MHz$).
- Osciloscopio con puntas de prueba.
- Resistor de referencia (40Ω).

Se debe conectar las puntas del osciloscopio a la salida del generador y en paralelo un resistor de referencia de 40Ω (500Ω en caso de sintonizar el circuito sin conectar el filtro pasa bajos).

Calibración

Se genera una señal sinusoidal de frecuencia $13,56MHz$ y de amplitud entre $2V$ y $5V$. El osciloscopio se configura para observar las figuras de Lissajous, con la escala del eje X dos veces la del eje Y. Se calibra el capacitor, C_{cal} , de la punta de prueba del osciloscopio hasta que la figura de Lissajous sea un segmento de recta, inclinado 45° .

Sintonizado

Luego de la calibración se sustituye el resistor de referencia por la antena y se sintoniza la misma variando los capacitores C_1 y C_2 , hasta que se obtenga una figura como la obtenida en el caso anterior. En ese momento la antena se encuentra sintonizada.

En caso de contar con un analizador de redes, el método anterior puede ser evitado, ya que es posible sintonizar el circuito buscando que la impedancia en el diagrama de Smith se ubique sobre el eje real al alcanzar la frecuencia de trabajo, en este caso $13,56MHz$.

B.1.4. Valor de ITVDD

El integrado de la familia Micore, entrega a la salida una señal cuadrada, con valor de pico a pico $U_{TxAC} = 2,5V_{pp}$ centrada en el valor de continua $U_{TxD} = 2,5V$, con una frecuencia $f_0 = 13,56MHz$ y un máximo de salida de corriente:

$$I_{TVDD} \leq 150mA$$

Esto significa que la salida TX oscila entre 0V y 5V. TX1 y TX2 usualmente están desfasados 180°, dependiendo de la configuración del bit 3 (TX2Inv) del registro Tx-Control (ver hoja de datos del integrado RC632).

B.1.5. Factor de calidad Q

El factor de calidad Q está directamente asociado con la forma de los pulsos modulados, ésto puede ser usado para verificar el valor del factor.

Un osciloscopio con al menos 50MHz de ancho de banda puede ser usado para observar la forma de los pulsos; donde los canales son conectados de la siguiente forma:

CH1: La punta de prueba conectada en este canal forma un loop con su línea de tierra para generar el acoplamiento necesario al estar próxima a la antena.

CH2: Este canal es conectado a la salida del pin 4 (MFout) de integrado y es usado como canal de disparo.

El registro MFoutSelect (26h) es configurado con los valores:

“2” para que la señal sea modulada con código Miller.

“3” para flujo de datos serial (sin código Miller).

Ver hoja de datos del integrado RC632 por más detalles.

Es recomendado verificar que la forma de los pulsos cumpla con lo establecido en la norma ISO14443. La figura B.1 muestra como son estos pulsos.

Para garantizar que la antena se encuentre bien sintonizada y el factor Q sea el correcto, debe verificarse que:

- i. La señal caiga debajo del 5 % de su valor máximo (sin tener en cuenta el sobretiro).

ii. El tiempo t_2 debe estar limitado entre: $0,7\mu s < t_2 < 1,4\mu s$.

Si $t_2 < 0,7\mu s$, el factor Q es muy alto (mayor que 35), por lo tanto la resistencia externa R_{ext} debe ser incrementada.

Si $t_2 > 1,4\mu s$ el factor Q es muy bajo, la distancia de operación no será cumplida y por lo tanto R_{ext} debe ser decrementada.

La tabla siguiente muestra la duración de los pulsos en μs de acuerdo a la norma ISO 14443.

Pulses length	t1	t2 min	t3 max	t4 max
T1 MAX	3.0	0.7	1.0	0.4
T1 MIN	2.0	0.7	1.0	0.4

Cuadro B.1: Duración de los pulsos en μs - ISO 14443

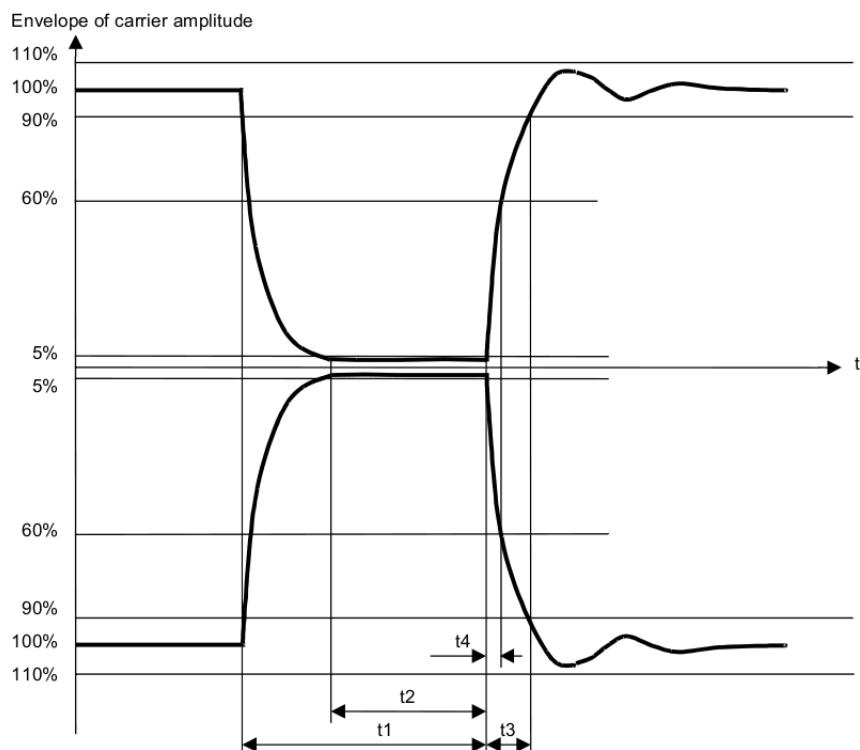


Figura B.1: Forma de pulso acorde a la norma ISO 14443

B.1.6. Circuito receptor

Cuando ya se han tomado en cuenta todos los cuidados en el diseño del transmisor, el circuito receptor debe ser conectado y ajustado. Los valores de los componentes sugeridos en el circuito receptor son los siguientes:

$$C_3 = 1nF \text{ (Ceramic NP0, tolerance } \leq \pm 10\%)$$

$$C_4 = 100nF \text{ (Ceramic X7R, tolerance } \leq \pm 10\%)$$

$$R_1 = 470\Omega \dots 4,7k\Omega$$

$$R_2 = 820\Omega$$

Dos reglas deben ser tenidas en cuenta para este circuito:

- i. El nivel de tensión de continua, DC, en la entrada Rx tiene que ser mantenido a V_{mid} (por eso es necesario R_2 y C_4 , ver figura B.2).
- ii. El nivel de tensión de alterna, AC, en la entrada Rx debe ser mantenido entre los siguientes límites: $1,5V_{pp} < V_{Rx} < 3,0V_{pp}$.

Si $V_{Rx} > 3,0V_{pp}$, R_1 debe ser incrementada.

Si $V_{Rx} < 1,5V_{pp}$, R_1 debe ser decrementada.

El voltaje a la entrada Rx debe ser verificado con y sin presencia de una tarjeta entre los límites máximo y mínimo de distancia de operación.

El valor límite $V_{Rx} = 3,0V_{pp}$ no debe ser excedido, un valor mayor puede causar fallos en la recepción.

Otros puntos a tener en cuenta

PCB

La parte más crítica de todo el circuito analógico es el directamente conectado al integrado, o sea el filtro pasa bajos y la conexión de TVDD a la fuente de alimentación. Entonces, por un lado un filtro puede ser usado para la conexión a la fuente de alimentación. Por otro lado el diseño del filtro a la salida del integrado, formado por L_0 y C_0 ,

debe ser considerado con mucho cuidado. *El área y la distancia del filtro al integrado deben ser mantenidas lo más pequeñas posibles. Es recomendado además un plano de tierra.*

La figura B.2 muestra un esquemático del diseño de una antena:

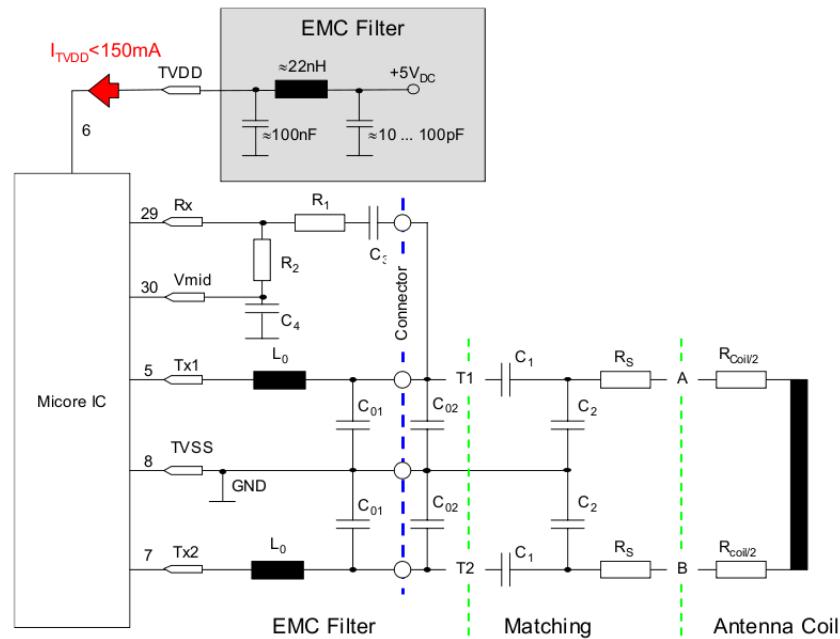


Figura B.2: Esquema de una antena, identificando sus principales secciones

Filtro de entrada de alimentación

Aunque no sería necesario, un filtro puede ser conectado a la entrada TVDD para mejorar los siguientes puntos:

- suprimir ruido llegado desde la fuente de alimentación.
- suprimir armónicos provenientes desde el transmisor.

Filtros idénticos pueden ser ubicados en las entradas AVDD y DVDD.

Blindaje

El blindaje eléctrico absorbe el campo eléctrico generado por la antena. Para construir un blindaje, es recomendable usar un PCB de al menos 4 capas, donde el loop del blindaje se encuentra en las 2 capas externas. Este loop no debe ser cerrado y debe estar conectado en su punto central al sistema de tierra mediante una vía. Los extremos de la bobina deben ser ruteados próximos entre sí para evitar inductancias adicionales. La figura B.3 da una idea de como debe ser un blindaje:

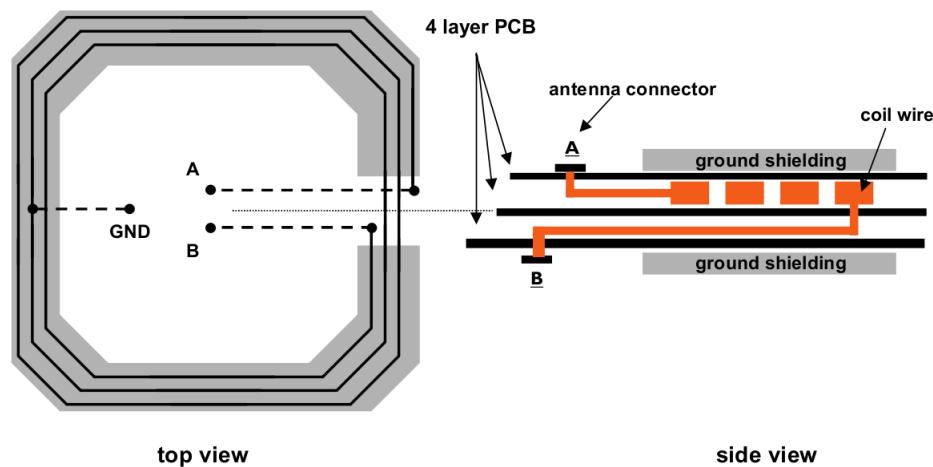


Figura B.3: Blindaje de una antena en un diseño de 4 capas

Apéndice C

Documentos y esquemáticos del hardware diseñado

C.1. Herramientas de diseño

Las herramientas CAD utilizadas para el diseño del hardware son Kicad [38] y gEDA [39], ambas son de uso libre y open source. Para el caso del lector/escritor RFID se empleó la aplicación gEDA, ya que permite generar pistas con forma de arco de circunferencia, elemento necesario para formar el inductor de la antena. El resto del hardware se diseñó en Kicad, dado que tiene una interfaz gráfica más amigable que gEDA lo que la hace más sencilla de usar. Ambas aplicaciones tienen la ventaja de permitir agregar y editar componentes o módulos diseñados por el mismo usuario y permiten generar archivos en formato gerber necesarios para enviar al fabricante de circuitos impresos.

C.2. Esquemáticos y componentes

En lo que sigue se muestran una serie de cuadros y esquemáticos para lograr un mayor detalle del diseño de las distintas partes que conforman el hardware del prototipo RF². En el cuadro C.1 se observa la distribución de pines asociada con el conector que une la SBC (Beagleboard), con el conversor de niveles de tensión, VLT. Por su parte el cuadro C.2 muestra el orden de los pines en el conector que interconecta la placa de circuito impreso, VLT, con la placa de circuito impreso, SCUI, que contiene el resto del

hardware.

Función	Nombre	Nº de pin	Nº de pin	Nombre	Función
Fuente 1,8 Volts	1V8	1	2	5V	Fuente 5 Volts
Led Verde	GPIO	3	4	GPIO	RST_SC
Led Rojo	GPIO	5	6	UART_TX	UART_TX
XOE	GPIO	7	8	UART_RX	UART_RX
D7	GPIO	9	10	GPIO	Led Amarillo
SPI_CS	SPI_CS	11	12	GPIO	Buzzer
D5	GPIO	13	14	GPIO	BacK Light
E	GPIO	15	16	GPIO	D6
SPI_SOMI	SPI_SOMI	17	18	GPIO	D4
SPI_SIMO	SPI_SIMO	19	20	GPIO	RW
SPI_CLK	SPI_CLK	21	22	GPIO	RS
IRQ_RF	GPIO	23	24	GPIO	RST_RF
OE	REGEN	25	26	nRESET	N.C.
Referencia 0 Volts	GND	27	28	GND	Referencia 0 Volts

Cuadro C.1: Conector 14x2 Beagleboard – VLT

Función	Nombre	Nº de pin	Nº de pin	Nombre	Función
Referencia 0 Volts	GND	1	2	N/C	RFU
Fuente 3,3 Volts	3V3	3	4	RST_RF	Reset RC632
RFU	N/C	5	6	SPI_SOMI	SPI para RC632
RFU	N/C	7	8	SPI_CLK	Reloj SPI
RFU	N/C	9	10	SPI_SIMO	SPI para RC632
RFU	N/C	11	12	SPI_CS	Chip Select RC632
Control LCD16x2	RS	13	14	IRQ_RF	Interrup RC632
Control LCD16x2	RW	15	16	BLK	BacK Light
Control LCD16x2	E	17	18	XOE	Reset oscilador
Dato LCD16x2	D4	19	20	BUZZ	Buzzer

Dato LCD16x2	D5	21	22	N/C	RFU
Dato LCD16x2	D6	23	24	N/C	RFU
Dato LCD16x2	D7	25	26	N/C	RFU
RFU	N/C	27	28	UART_RX	Smart card
Fuente 5 Volts	5V	29	30	UART_TX	Smart card
Referencia 0 Volts	GND	31	32	RST_SC	Reset Smart card
Led Rojo	LED_R	33	34	N/C	RFU
Led Amarillo	LED_A	35	36	N/C	RFU
Led Verde	LED_V	37	38	N/C	RFU
RFU	N/C	39	40	N/C	RFU

Cuadro C.2: Conector 20x2 VLT - SCUI

C.2.1. SBC

Componente	Descripción
SBC	Beagleboard RevC4
Memoria SD	4GB SDHC Class 6 SD Card
Cable serial DB9 nulo	DB9F Null Modem (RS-232) (6-ft)
Cable conversor usb–serial	USB to DB9M RS-232 (PL-2302)
Cable USB	USB Mini-A to USB A Female, OTG
Cable USB	USB Mini-B Male to USB A Male
Fuente	5VDC/2,5A

Cuadro C.3: SBC y lista de accesorios

C.2.2. VLT - Conversor de Voltajes

Componente	Descripción	Footprint	Valor
C1	Polarized Capacitor (Tantal)	6032[2312]	10uF, 25V
C2	Polarized Capacitor (Tantal)	6032[2312]	100uF, 6V3
U4	Regulador LM1117-3.3	SOT-223	3.3V, 800mA
U1, U2, U3	Voltage Level Translator	TSSOP20	-
P1	RECEPTACLE, 28WAY, 2ROW	SMD Pitch 2,54	28 pines
P2	RECEPTACLE, 40WAY, 2ROW	SMD Pitch 2,54	40 pines
P1b	HEADER, 28WAY, 2ROW	T H Pitch 2,54	28 pines
P2b	HEADER, 40WAY, 2ROW	T H Pitch 2,54	40 pines

Cuadro C.4: Lista de componentes de la placa de circuito impreso VLT

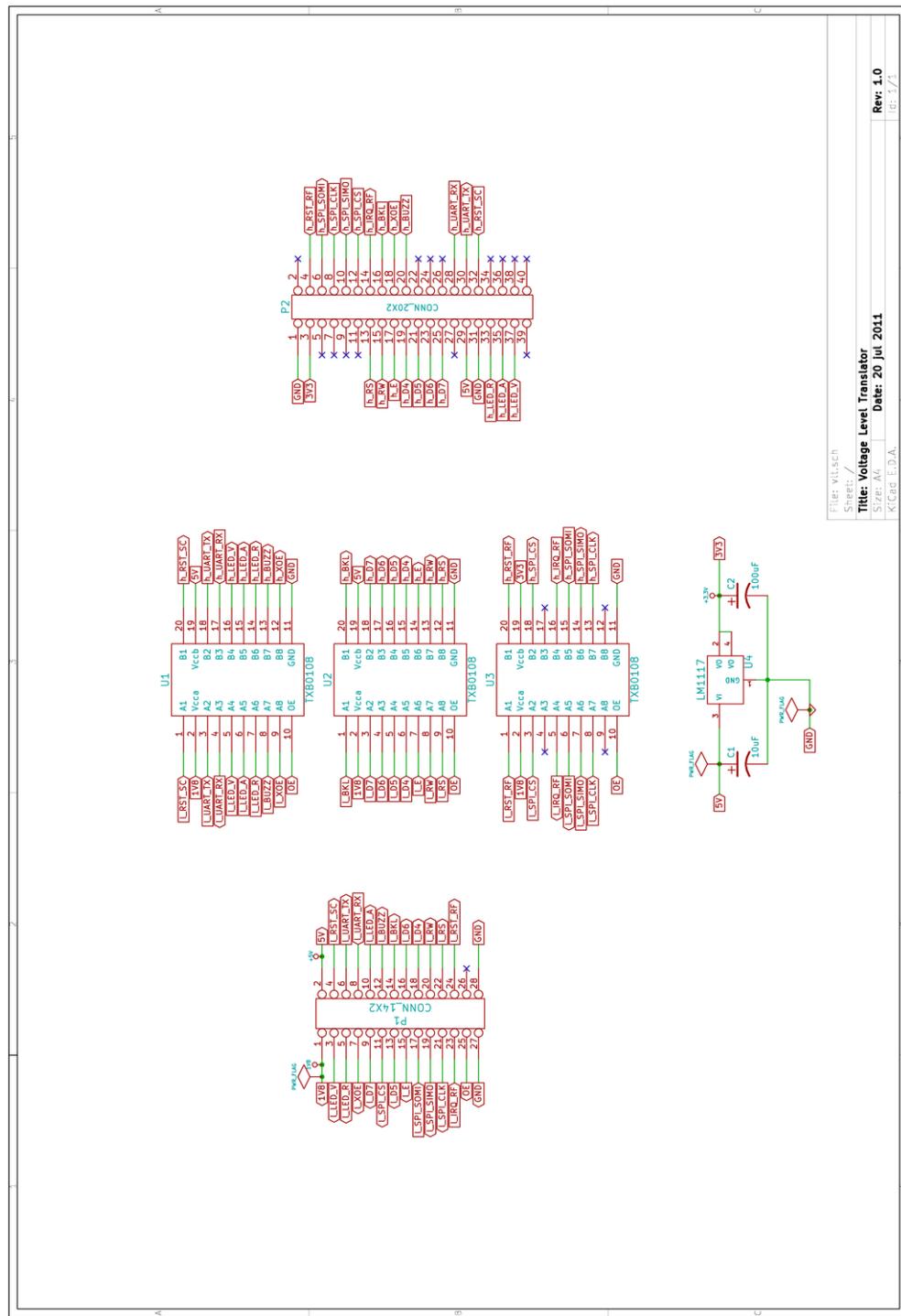


Figura C.1: Esquemático de la placa VLT - Voltage Level Translator

C.2.3. SCUI - Lector de tarjetas de contacto e Interfaz de Usuario

Componente	Descripción	Footprint	Valor
R9	Resistor 100W 1/4W 1 %	3216[1206]	100W 1/4W 1 %
R10, R13	Resistor 100KW 1/4W 5 %	3216[1206]	100KW 1/4W 5 %
R11, R12, R14	Resistor 10KW 1/4W 5 %	3216[1206]	10KW 1/4W 5 %
Q2	TRANSISTOR, NPN, 300MHZ	SOT23	MMBT3904
Q3	TRANSISTOR, PNP, 250MHZ	SOT23	MMBT3906
J2	SIM socket (6 contacts)	SMD	-
JP3, JP4	HEADER, 1ROW, 3WAY	T H Pitch 2,54	3 pines
X1	Oscillator 3.579545MHz	SMD	3.579545 Mhz
ESD1	Anti ESD	SOT323	6V / 150W

Cuadro C.5: Lista de componentes del lector de tarjetas de contacto, SC

Componente	Descripción	Footprint	Valor
R1	Resistor 4K7 1/10W 1 %	1608[0603]	4,7KW 1/10W 1 %
R2, R8	Resistor 3R3 1/10W 1 %	1608[0603]	3,3W 1/10W 1 %
R3, R4, R5	Resistor 680R 1/10W 1 %	1609[0603]	680W 1/10W 1 %
R6, R7	Resistor 10K 1/10W 1 %	1608[0603]	10KW 1/10W 1 %
RV1	Preset 15K 1/10W 25 %	SMD	15KW 1/10W 25 %
Q1	TRANSISTOR, NPN, 300MHZ	SOT23	MMBT3904
S1	LCD MODULE 16X2 CHARACTER	Pitch 2,54	-
CONN1	HEADER FEMALE 16POS.1”TIN	Through Hole	16 pines
CONN2	HEADER, 1ROW, 16WAY	T H Pitch 2,54	16 pines
LED1	Led green 5mm	Through Hole	1,9V, 2mA
LED2	Led red 5mm	Through Hole	1,9V, 2mA
LED3	Led yellow 5mm	Through Hole	2,4V, 2mA
BUZZ1	Buzzer	Through Hole	3 20Vdc, 3 16mA

Cuadro C.6: Lista de componentes para la interfaz de usuario, LCD

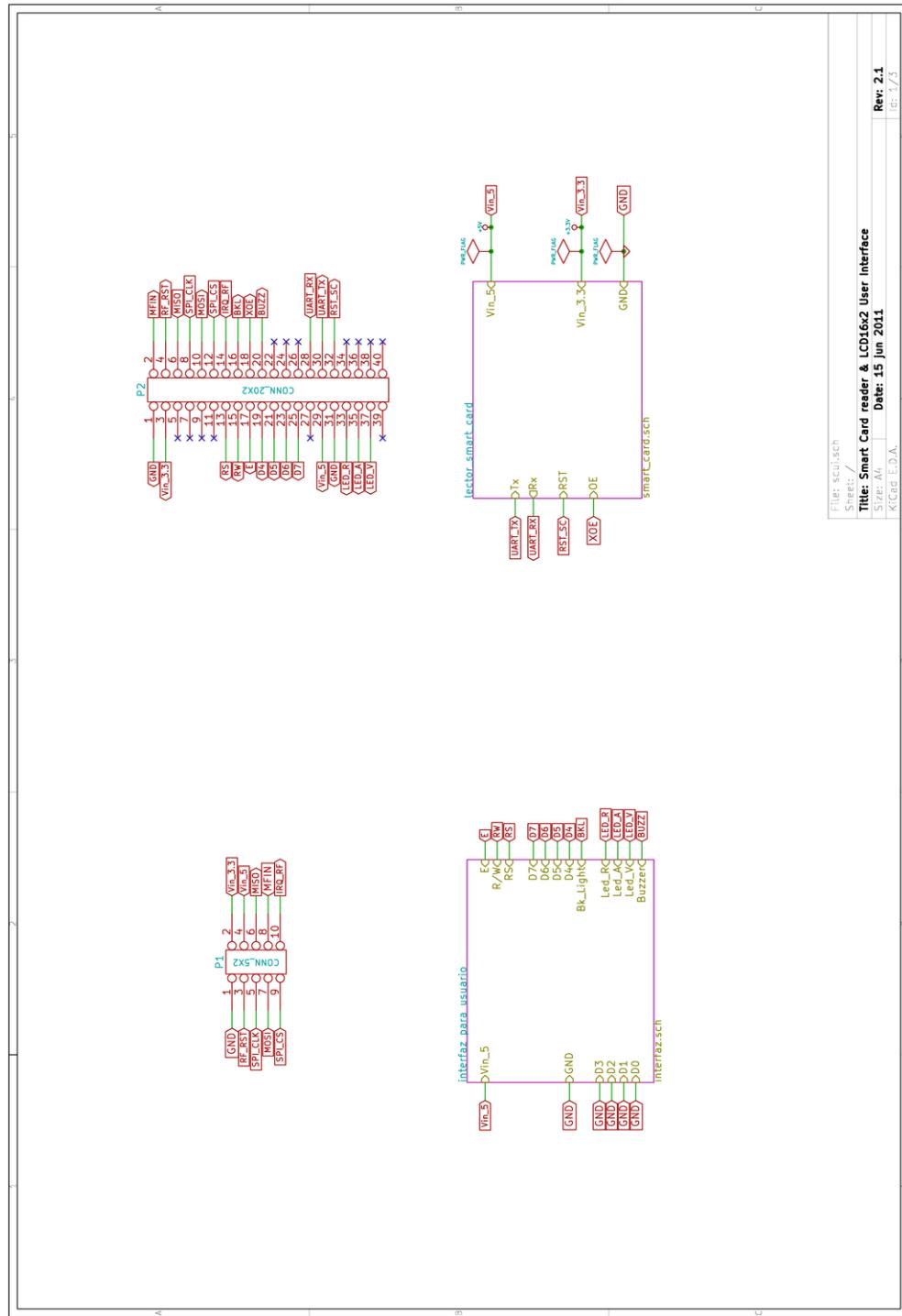


Figura C.2: Esquemático de la placa SCUI

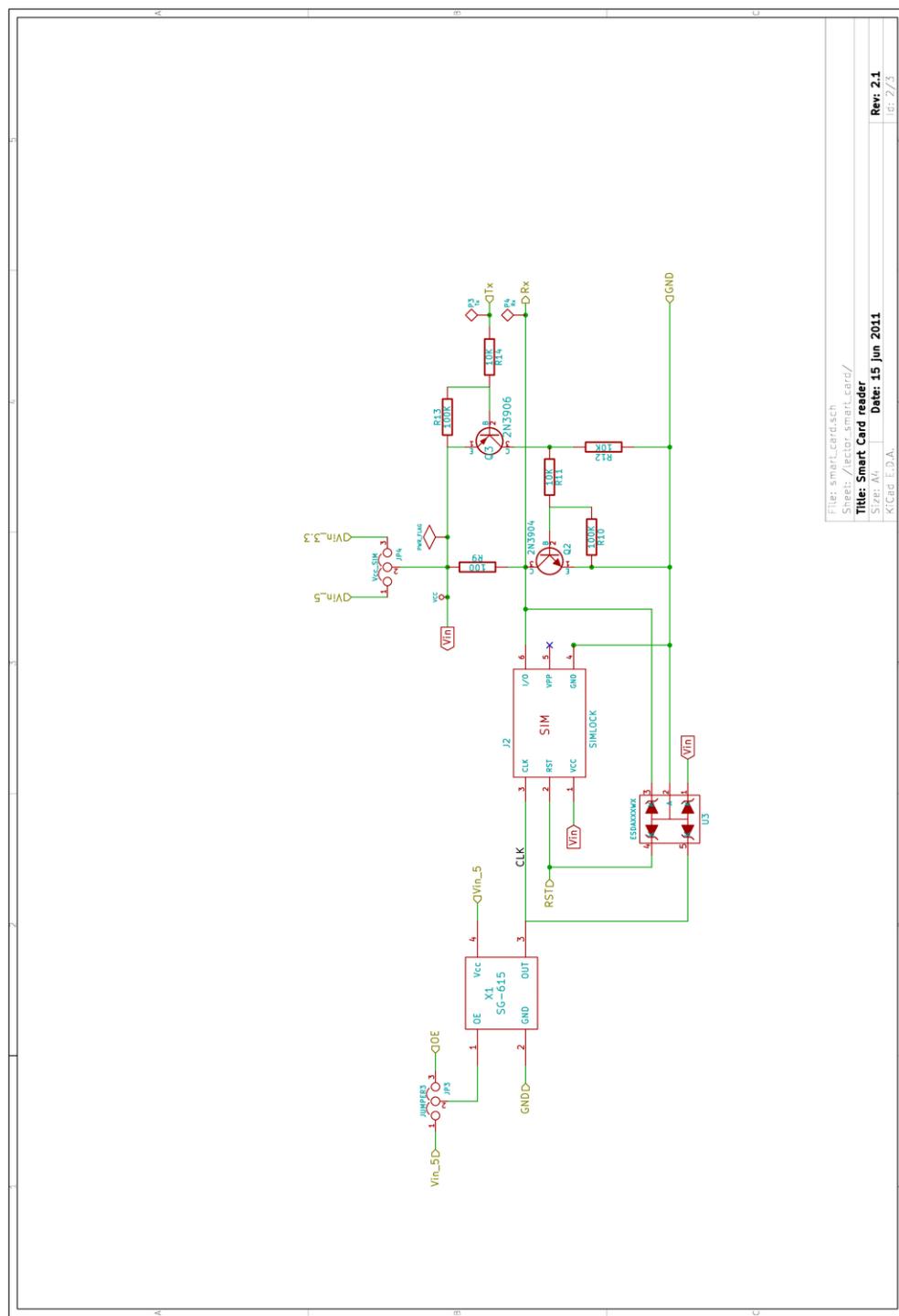


Figura C.3: Esquemático del lector de tarjetas de contacto, incluido en la placa SCUI

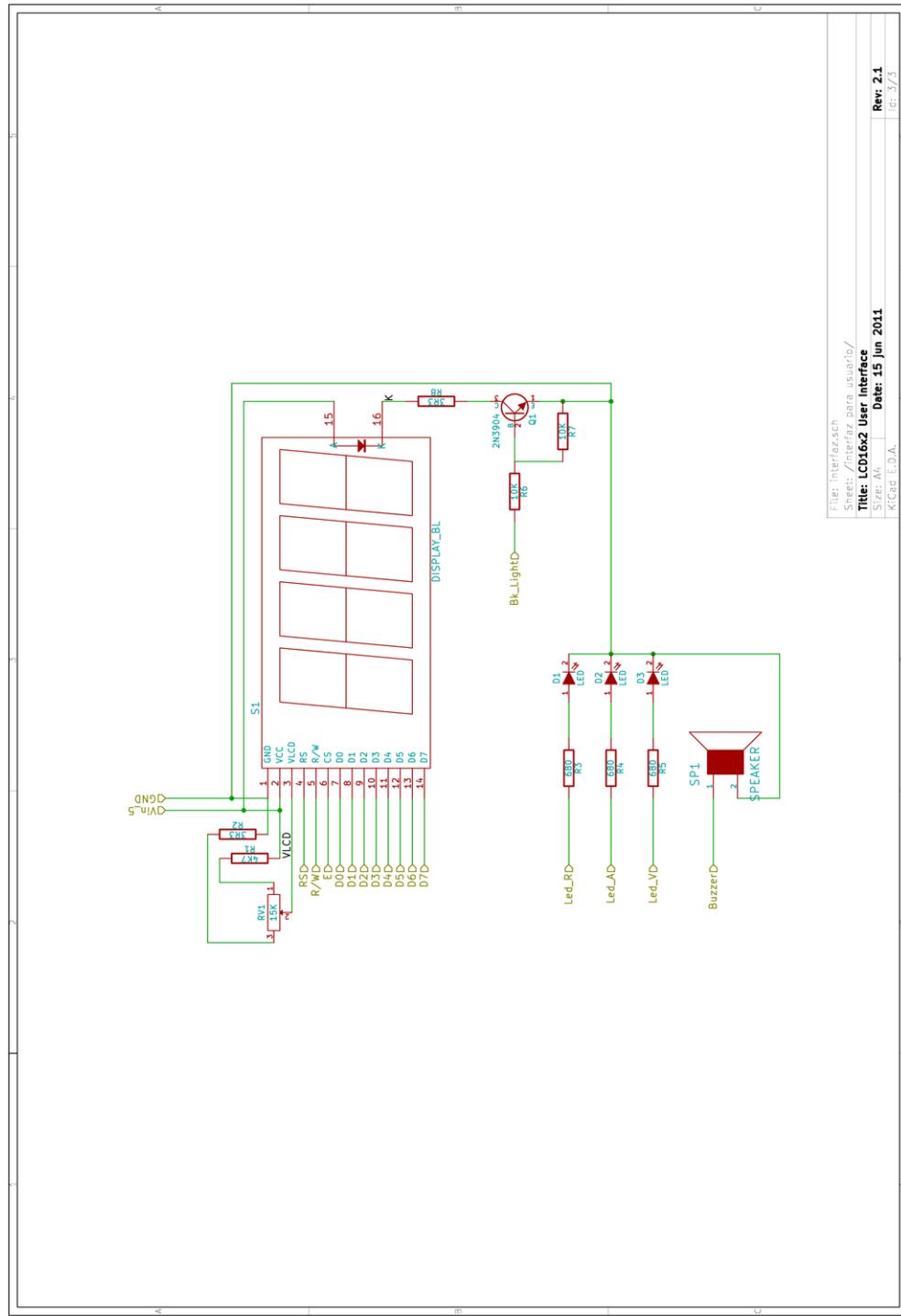


Figura C.4: Esquemático de la interfaz de usuario, incluido en la placa SCUI

C.2.4. Lector/Escritor RFID

Componente	Descripción	Footprint	Valor
C1, C2	Capacitor	1608[0603]	10pF, Ceramic NPO, 2 %
C3, C4	Capacitor	1609[0603]	100pF, Ceramic NPO, 2 %
C5, C6, C7, C8	Capacitor	1608[0603]	NC
R1, R2	Resistor	1608[0603]	0W, 1/10W, 1 %

Cuadro C.7: Lista de componentes de la antena RF, Inductor

+ Adaptación de impedancia

Componente	Descripción	Footprint	Valor
C10	Capacitor	1610[0603]	10pF, Ceramic NPO, 2 %
C1, C2	Capacitor	1608[0603]	15pF, Ceramic NPO, 5 %
C12, C13	Capacitor	1608[0603]	56pF, Ceramic NPO, 2 %
C14, C15	Capacitor	1608[0603]	68pF, Ceramic NPO, 1 %
C9	Capacitor	1609[0603]	100pF, Ceramic NPO, 2 %
C16	Capacitor	1608[0603]	1nF, Ceramic NPO, 10 %
C4, C5, C7, C8, C11, C17	Capacitor	1608[0603]	100nF, Ceramic X7R, 10 %
C3, C6, C18	Capacitor	1608[0603]	10uF, Ceramic X5R, 20 %
L1, L2, L3, L6	Inductor	2012[0805]	22nH, 700mA, 5 %
L4, L5	Inductor	3225[1210]	1uH, 400mA, 5 %
R3	Resistor	1608[0603]	50W, 1/10W 1 %
R2	Resistor	1608[0603]	820W, 1/10W 5 %
R1	Resistor	1608[0603]	2,2KW, 1/5W 1 %
U1	Reader ISO14443	SO32	CL RC632
U2	Crystal Oscilla- tor, HC49 US SMD	49USMXL	13.56MHz, 10pF

U3	Operational Amplifier (up to 7.5V)	SOT23-5	OPA354
CONN1, CONN2	U.FL-R Connector	U.FL-R-SMT	-
J1	HEADER, 10WAY, 2ROW	T H Pitch 2,54	10 pines
J1b	RECEPTACLE, 10WAY, 2ROW	SMD Pitch 2,54	10 pines

Cuadro C.8: Lista de componentes del lector/escritor RFID,
sin la antena RF

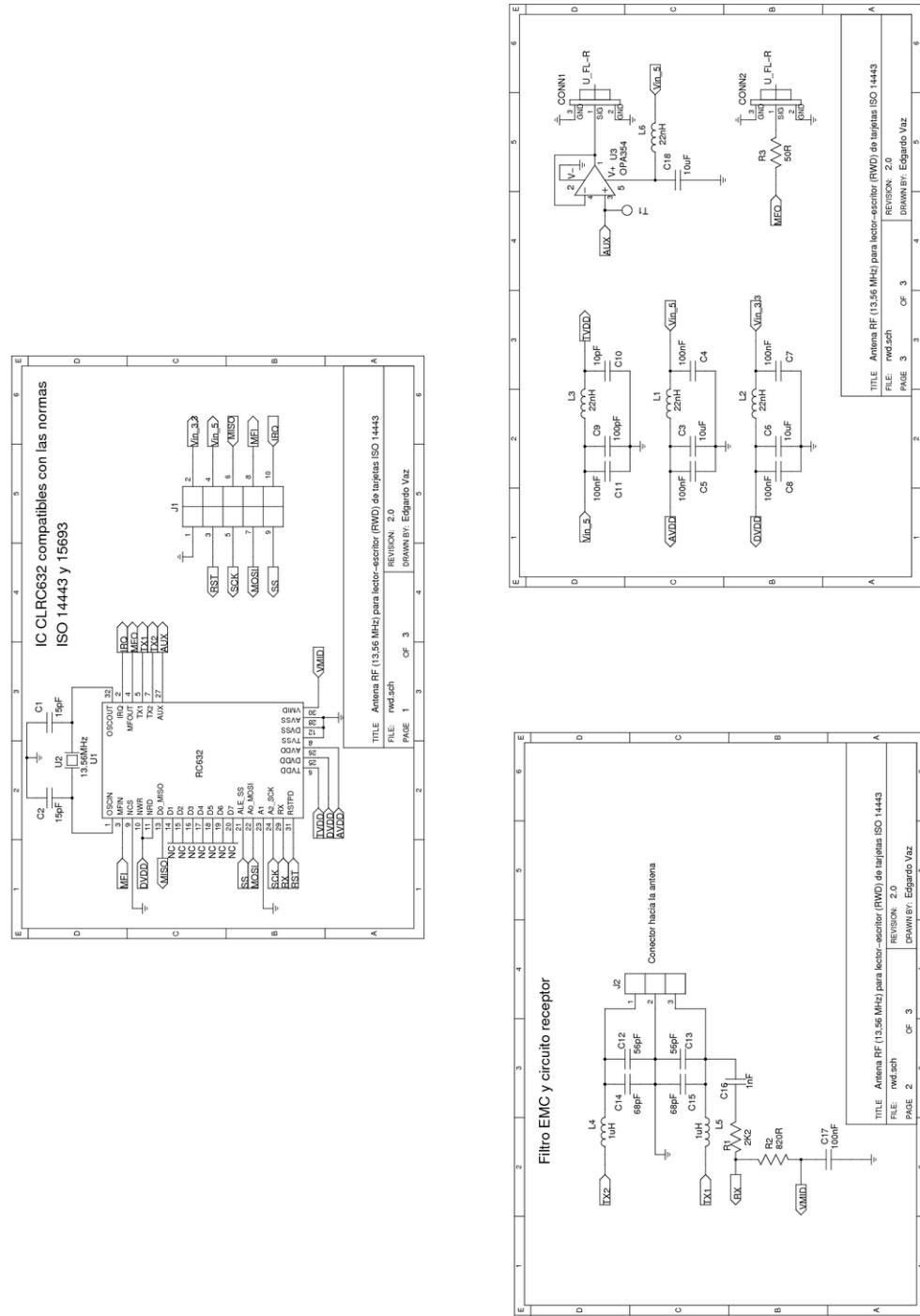


Figura C.5: Esquemático del módulo digital del lector/escritor RFID

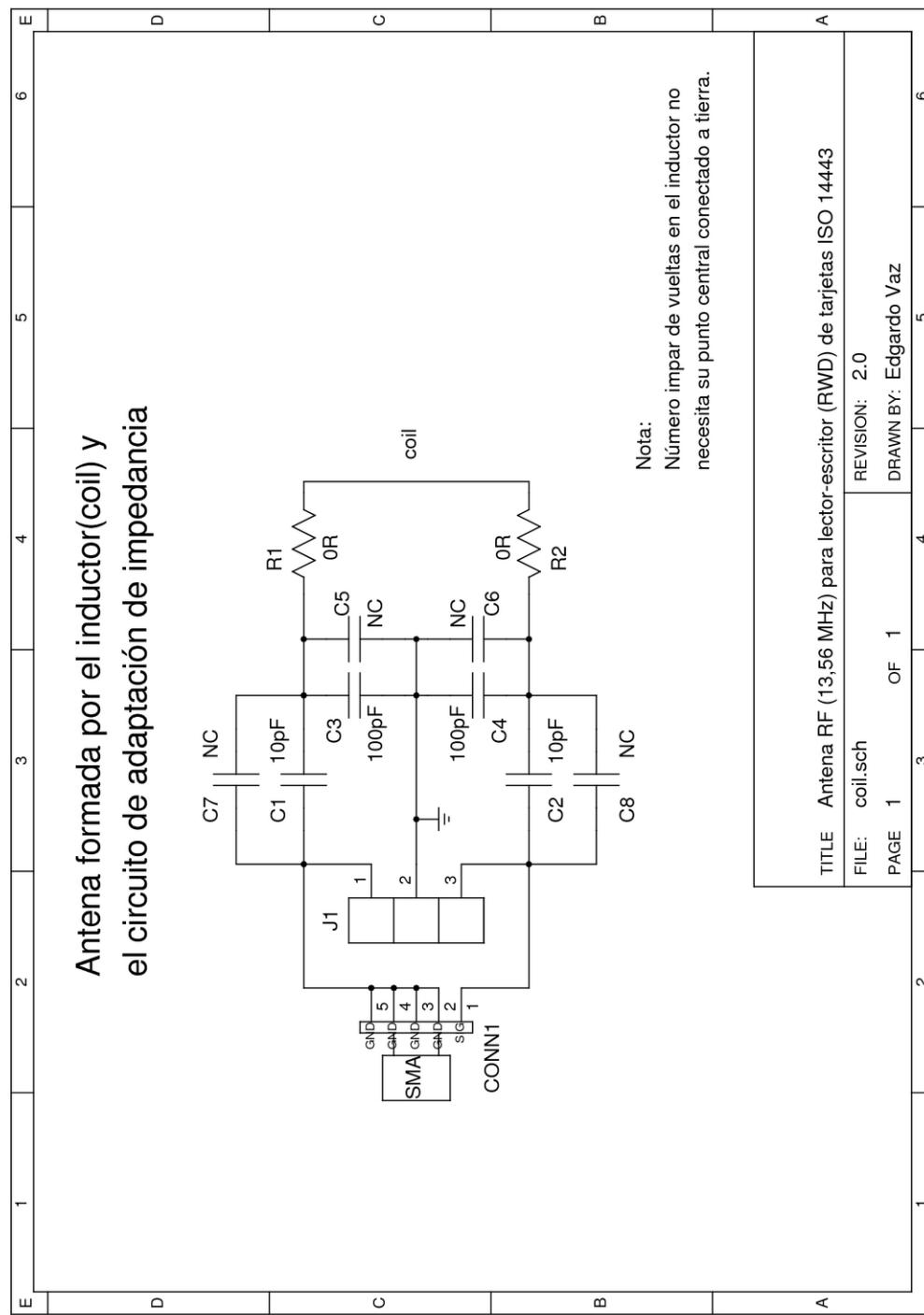


Figura C.6: Esquemático de la antena RFID, Inductor + Adaptación de impedancia

Apéndice D

Software

D.1. Gestor de paquetes opkg

Se puede configurar el gestor de paquetes de dos maneras para que encuentre los repositorios desde donde descargar los paquetes para su instalación.

La primera de ellas es la más prolífica y consiste en los siguientes pasos:

Se crean los archivos xxx-feed.conf (xxx hace referencia a los subdirectorios dentro del repositorio) que contienen la ruta a los repositorios:

```
$ echo src/gz angstrom-feed http://www.angstrom-distribution.org/feeds/unstable/  
ipk/glibc/armv7a/base > /etc/opkg/angstrom-feed.conf  
  
$ echo src/gz perl-feed http://www.angstrom-distribution.org/feeds/unstable/  
ipk/glibc/armv7a/perl/ > /etc/opkg/perl-feed.conf  
  
$ echo src/gz sdk-feed http://www.angstrom-distribution.org/feeds/unstable/  
ipk/glibc/sdk/ > /etc/opkg/sdk-feed.conf  
  
$ echo src/gz python-feed http://www.angstrom-distribution.org/feeds/unstable/  
ipk/glibc/armv7a/python/ > /etc/opkg/python-feed.conf  
  
$ echo src/gz debug-feed http://www.angstrom-distribution.org/feeds/unstable/  
ipk/glibc/armv7a/debug/ > /etc/opkg/debug-feed.conf  
  
$ echo src/gz beagleboard-feed http://www.angstrom-distribution.org/feeds/  
unstable/ipk/glibc/armv7a/machine/beagleboard/ > /etc/opkg/beagleboard-feed.conf
```

```
$ echo src/gz noarch-feed http://www.angstrom-distribution.org/feeds/unstable/  
ipk/glibc/all/ > /etc/opkg/noarch-feed.conf
```

Nota: /gz indica que los paquetes están comprimidos con extensión gz.

Se debe actualizar la lista de paquetes disponibles:

```
$ opkg update
```

El comando anterior crea los archivos /var/lib/opkg/xxx-feed que contienen la lista completa de paquetes en el repositorio.

Nota: Se puede usar el comando opkg list | wc -l para conocer la lista de paquetes.

La segunda opción es modificar directamente el archivo /etc/opkg/opkg.conf a continuación de la línea:

```
src <src-name> <source-url>
```

Se agrega lo siguiente:

```
src angstrom-feed http://www.angstrom-distribution.org/  
feeds/unstable/ipk/glibc/armv7a/base/  
  
src perl-feed http://www.angstrom-distribution.org/feeds/  
unstable/ipk/glibc/armv7a/perl/  
  
src sdk-feed http://www.angstrom-distribution.org/feeds/  
unstable/ipk/glibc/sdk/  
  
src python-feed http://www.angstrom-distribution.org/feeds/  
unstable/ipk/glibc/armv7a/python/  
  
src debug-feed http://www.angstrom-distribution.org/feeds/  
unstable/ipk/glibc/armv7a/debug/  
  
src beagleboard-feed http://www.angstrom-distribution.org/  
feeds/unstable/ipk/glibc/armv7a/machine/beagleboard/ src  
noarch-feed http://www.angstrom-distribution.org/feeds/  
unstable/ipk/glibc/all/
```

Luego se actualiza la lista de paquetes:

```
$ opkg update
```

Una vez que se agregan los repositorios es posible instalar paquetes mediante el comando:

```
$ opkg install <nombre del paquete>
```

El nombre del paquete puede buscarse en el sitio web que fue pasado como dirección del repositorio en la configuración anterior.

Por más información sobre le uso de este comando, referirse a [21] (ver apéndice E).

D.2. Instalación, configuración y uso de SDK

Como fue comentado antes, el SDK es generado a partir de la herramienta web Narcissus. Para la instalación se debe tener el archivo (SDK.tar.bz2, por ejemplo) generado.

En el PC de desarrollo se debe hacer lo que sigue:

Se realiza la instalación con el usuario root y luego se explica como proceder con cualquier otro usuario.

```
$ sudo su
```

Se descomprime el SDK en /

```
$ sudo tar -xf SDK.tar.bz2 -C /
```

Se debe haber copiado el directorio angstrom en /usr/local

Luego, se habilita el uso del gestor de paquetes (opkg) y se agraga en el croscompilador al PATH del sistema. Se debe establecer esta configuración con cada usuario que quiera utilizar la herramienta.

```
$ ./usr/local/angstrom/arm/environment-setup
```

Lo que se tiene es un sistema Angström configurado en el PC de desarrollo con el que se puede actualizar la lista de repositorios, bajar paquetes, compilarlos para la arquitectura deseada, etc.

Actualización de repositorios:

```
$ opkg-target update
```

Instalación de paquetes:

```
$ opkg-target install <paquete>
```

Actualización de paquetes:

```
$ opkg-target upgrade
```

El uso de opkg-target es solo permitido para el usuario root por un tema de permisos, aunque el uso del croscompilador es permitido para cualquier usuario siempre y cuando haga lo siguiente cada vez que vaya a utilizar la herramienta:

```
$ ./usr/local/angstrom/arm/environment-setup
```

Para referirse a las herramientas de la arquitectura para la cual se generan los binarios se utiliza el prefijo arm-angstrom-linux-gnueabi-.

Por más detalles referirse a [50].

D.3. OpenEmbedded-Bitbake

Para la instalación, configuración y ejecución de esta herramienta, se deben seguir los pasos en detalle. Se debe tener una buena conexión a internet para la descarga de fuentes, espacio libre en disco duro de no menos de 10GB, buen procesador y paciencia ya que los desarrollos completos demoran varias horas.

Primero se instalan los paquetes previos necesarios.

Paquetes importantes:

```
$ sudo apt-get install sed wget cvs subversion git-core \
    coreutils unzip texi2html texinfo docbook-utils \
    gawk python-pysqlite2 diffstat help2man make gcc build-essential g++ \
    desktop-file-utils chrpath
```

Paquetes secundarios (aceleran los procesos):

```
$ sudo apt-get install libxml2-utils xmlto python-psycodbc apr
```

Chequear que /bin/sh no tiene un enlace simbólico a “dash”:

```
$ ls -l /bin/sh
```

Debe estar enlazado a “bash”. Si no lo está:

```
$ sudo dpkg-reconfigure dash
```

Aquí seleccionamos NO instalar “dash” como /bin/sh

Compilador para aumentar la velocidad de bitbake:

```
$ sudo apt-get install python-psycodbc
```

Para versiones de Ubuntu mayores o iguales a la 10.04:

```
$ sudo su
$ echo 128 > /proc/sys/vm/mmap_min_addr
$ exit
$ sudo sysctl -w vm.mmap_min_addr=128
```

Instalación:

El directorio base seleccionado para el desarrollo con OpenEmbedded es stuff (puede ser otro), y se ubicó en /.

Se crea la estructura de directorios:

```
$ sudo mkdir -p /stuff/build/conf
```

```
$ cd /stuff/
```

Bitbake es la herramienta de construcción que utiliza OpenEmbedded. Está escrita en Python, por lo que no es necesario compilarlo para que funcione.

```
$ sudo wget http://download.berlios.de/bitbake/bitbake-1.10.2.tar.gz
```

Nota: esta es la última versión disponible al momento. Para saber cual es la última versión disponible, entrar a: <http://download.berlios.de/bitbake/>.

Para las nuevas versiones es necesario tener instalado Python 2.6 o posterior.

Luego se descomprime el tar.gz bajado:

```
$ sudo tar -xzf bitbake-1.10.2.tar.gz  
$ ls
```

Se debe ver un directorio llamado en este caso bitbake-1.10.2 (se puede renombrar a: “bitbake” por comodidad).

Para obtener OpenEmbedded es necesario tener instalado git.

Ahora se hace lo siguiente:

```
$ cd /stuff  
$ sudo git clone git://git.openembedded.org/openembedded
```

Nota: git clone es para hacer un checkout del repositorio.

Otro repositorio es <http://repo.or.cz/r/openembedded.git>

Demora mucho. Al finalizar se crea un directorio con el nombre openembedded.

Se recomienda actualizar OpenEmbedded una vez al día:

```
$ cd /stuff/openembedded  
$ sudo git pull
```

Configuración:

Se modifica la configuración local, donde se debe indicar todo lo que se quiera crear.

```
$ cd /stuff/  
$ sudo cp openembedded/conf/local.conf.sample build/conf/local.conf  
$ sudo vi build/conf/local.conf
```

Nota: en lugar de vi, se pueden usar nano o incluso gedit. Estos son editores de texto ordenados por complejidad.

En este punto hay que tener cuidado debido a que existen muchas variables a editar y se necesita mucho conocimiento para poder cambiarlas.

La mínima cantidad de variables a editar para un desarrollo correcto son las siguientes:

```
BBFILES = “/stuff/openembedded/recipes/*/*.bb”  
MACHINE = “beagleboard”  
DISTRO = “angstrom-2008.1”  
PARALLEL_MAKE = “-j 5”  
INHERIT += “rm_work”
```

Nota: Los parámetros detallados antes ya existen en el archivo de configuración, algunos deben ser descomentados y/o editados ya que lo único que cambia es el valor asignado.

BBFILES: indica que archivos son considerados durante el desarrollo.

MACHINE: el nombre asociado a la SBC que se esté usando. Para saber el nombre asociado a la SBC se puede ver el contenido del directorio
/stuff/openembedded/conf/machine.

DISTRO: qué versión de la distribución se quiere instalar. Aquí se eligió la última versión estable de Angström, aunque si se pone “angstrom-2010.x”, se obtiene una versión más nueva (no asegura estabilidad). La versión no solo afecta a la distribución sino que la versión del kernel generado va a depender de esta versión. Por ejemplo, con

2008.1 se obtiene un kernel 2.6.32 y con 2010.x se obtiene un kernel 2.6.37. Para saber cuales son las versiones disponibles al momento se puede ver el contenido del directorio /stuff/openembedded/conf/distro.

PARALLEL_MAKE: indica cuantas operaciones simultáneas puede realizar el procesador de la pc de desarrollo. En general el valor se puede calcular como sigue: (cantidad de procesadores)x2 +1. En este caso 5 equivale a una cpu del tipo core2duo.

INHERIT += "rm_work": esta opción elimina los fuentes después de haber construido los paquetes. Esto hace que el tamaño del desarrollo en disco, no supere los 10GB. Si esta opción no se elige, se deben tener por lo menos 40GB de espacio libre en disco duro. También se puede decir que si se elige esta opción algunos fuentes deberán ser bajados nuevamente en cada desarrollo, lo que lo puede hacer más lento.

Conviene leer todo el archivo para tener una idea básica de lo que hace. Si en algún lugar se quiere hacer referencia al “home” del usuario, la ruta se debe escribir completa (no se puede ~).

La última línea debe ser borrada (esto es para asegurarse de que se leyó todo).

Ejecución:

Siempre antes de empezar a desarrollar se debe ejecutar lo siguiente:

```
$ export BBPATH=/stuff/build:/stuff/openembedded  
$ export PATH=/stuff/bitbake/bin:$PATH
```

Comenzando el desarrollo:

```
$ cd /stuff/build  
$ bitbake console-image
```

Nota: Al ejecutarlo por primera vez, demora varias horas.

Si aparecen errores como “Please set persistent and cache” o “no se puede acceder al directorio tmp” esto se soluciona dando permisos al directorio stuff:

```
$ sudo su
```

```
$ chmod -R 777 /stuff
```

El comando bitbake console-image baja todos los fuentes y genera todos los binarios necesarios para la ejecución de la distribución Angström en modo consola. Otra opción, si solo queremos un kernel, es el comando bitbake virtual/kernel, pero para entender bien el funcionamiento del bitbake, la primera vez se recomienda bitbake console-image.

Luego de ejecutado el comando, se crea toda la estructura de directorios.

A continuación se detallan los más importantes:

/stuff/build/tmp/deploy/glibc/images/beagleboard/

Aquí se guardan los archivos generados.

/stuff/build/tmp/work/beagleboard-angstrom-linux-gnueabi/

Aquí se encuentran los directorios con los fuentes del kernel y el u-boot.

D.4. uImage

Se detallan las modificaciones en el archivo board_omap3beagle.c necesarios para la inicialización correcta de las interfaces SPI y GPIO. Como ya se comentó este archivo está ubicado en /stuff/build/tmp/work/beagleboard-angstrom-linux-gnueabi/linux-omap-.../git/arch/arm/mach-omap2/ y es el encargado de toda la inicialización de las interfaces del sistema.

En el caso de la interfaz SPI se vio que no se lograba un mapeo de la interfaz en /dev, lo que no permitía el acceso a ésta a nivel de usuario. En el caso de la interfaz GPIO se vio que para los pines GPIO no basta con los cambios realizados en el u-boot para establecer su dirección y valor al iniciar el sistema. A continuación se plantea una posible solución para ambos casos.

Cambios asociados al SPI:

Se creó una estructura (spi_board_info beagle_mcspi_board_info) que contempla todas las posibilidades de interfaz SPI en la Beagleboard y agrega información sobre éstas.

En la estructura se pueden ver tres formas de representar al SPI: spi3.0, spi3.1 y spi4.0. El microprocesador de la Beagleboard tiene 4 interfaces SPI disponibles de las cuales la 3 y la 4 son accesibles desde el bloque de expansión de la Beagleboard. Además la interfaz spi3 se puede encontrar en dos modalidades 3.0 o 3.1 dependiendo de si se utiliza el CS0 o el CS1 como chip select de la interfaz. La interfaz spi4 solo puede utilizar el CS0. Es por esto que en la estructura se definen spi3.0, spi3.1 y spi4.0. Dentro de cada interfaz definida se agrega información sobre la interfaz, como ser el nombre (modalias), la máxima velocidad de transferencia/recepción de datos (max_speed_hz), número de bus (bus_num), chip select (chip_select) y modo del spi (SPI_MODE_1).

```
static struct spi_board_info beagle_mcspi_board_info[] = {
    /* spi 3.0 */
    {
        .modalias = "spidev",
        .max_speed_hz = 48000000, /* 48 Mbps */
        .bus_num = 3,
        .chip_select = 0,
        .mode = SPI_MODE_1,
    },
    /* spi 3.1 */
    {
        .modalias = "spidev",
        .max_speed_hz = 48000000, /* 48 Mbps */
        .bus_num = 3,
        .chip_select = 1,
        .mode = SPI_MODE_1,
    },
    /* spi 4.0 */
    {
        .modalias = "spidev",
        .max_speed_hz = 48000000, /* 48 Mbps */
        .bus_num = 4,
        .chip_select = 0,
        .mode = SPI_MODE_1,
    },
};
```

Luego de definidas las interfaces fue creada una función (`omap3_beagle_init_spi_rf2`) que las inicializara.

```
static void __init omap3_beagle_init_spi_rf2(void)
{
    printk(KERN_INFO "Usando SPI\n");
    /* hook the spi ports to the spidev driver */
    spi_register_board_info(beagle_mcspi_board_info,
                           ARRAY_SIZE(beagle_mcspi_board_info));
}
```

Para que la función de inicialización de la interfaz SPI pueda ejecutarse, se debe hacer referencia a ésta en la función general de inicialización de interfaces asociada a la Beagleboard (`omap3_beagle_init`).

```
omap3_beagle_init_spi_rf2();
```

Con estos cambios se logró que las interfaces SPI sean accesibles en el espacio de usuario bajo `/dev` y mapeadas como `spidev3.0`, `spidev3.1`, `spidev4.0`.

Cambios asociados al GPIO:

Se creó una función (`gpio_config_rf2`) que dado un número asociado con el pin GPIO, establece su dirección y valor.

```
static void gpio_config_rf2(unsigned gpio, int direction,
int value) {
    /* Tell the kernel, we want to use the GPIO*/
    if (gpio_request(gpio, "gpio\n") != 0) {
        printk(KERN_ALERT "Unable to request GPIO %d\n",
               gpio);
    }
    else {
        /* Now tell the kernel that GPIO is an (in-out)put
        and should be set to value (only as output) */
        switch (direction) {
            case 0: if (gpio_direction_output(gpio, value) != 0) {
                      printk(KERN_ALERT "Unable to set GPIO
                     direction for GPIO %d\n", gpio);
                  }
            else {
                /* enable direction on userspace */

```

```

        if (gpio_export(gpio, 1) != 0) {
            printk(KERN_ALERT "Unable to set GPIO
                           export for GPIO %d\n", gpio);
        }
    }
break;
case 1: if (gpio_direction_input(gpio) != 0) {
            printk(KERN_ALERT "Unable to set GPIO
                           direction for GPIO %d\n", gpio);
        }
else {
    /* enable direction on userspace */
    if (gpio_export(gpio, 1) != 0) {
        printk(KERN_ALERT "Unable to set GPIO
                           export for GPIO %d\n", gpio);
    }
}
break;
default: break;
}
}
}

```

Se creó la función de inicialización (gpio_rf2) que establece la configuración de todos los pines GPIO del sistema RF².

```

static void __init gpio_rf2(void)
{
    printk(KERN_ALERT "Configurando GPIO RF2...\n");
    gpio_config_rf2(133, 0, 1); /*E*/
    gpio_config_rf2(134, 0, 0); /*D5*/
    gpio_config_rf2(136, 0, 0); /*D7*/
    gpio_config_rf2(137, 0, 0); /*XOE*/
    gpio_config_rf2(138, 0, 0); /*led rojo*/
    gpio_config_rf2(139, 0, 0); /*led_verde*/
    gpio_config_rf2(144, 0, 0); /*RST_SC*/
    gpio_config_rf2(145, 0, 0); /*led_amarillo*/
    gpio_config_rf2(156, 0, 0); /*RW*/
    gpio_config_rf2(157, 0, 0); /*RS*/
    gpio_config_rf2(158, 0, 0); /*Buzzer*/
    gpio_config_rf2(159, 0, 0); /*D4*/
    gpio_config_rf2(161, 0, 0); /*D6*/
    gpio_config_rf2(162, 0, 0); /*Backlight*/
    gpio_config_rf2(168, 0, 1); /*RST_RF*/
}

```

```
    gpio_config_rf2(183, 1, 0); /*IRQ_RF*/  
}
```

Luego, como en el caso de la interfaz SPI, se debe hacer referencia a la función anterior en la función de inicialización del sistema (omap3_beagle_init).

```
gpio_rf2();
```

Con estos cambios se logró una cambio en la configuración de los pines GPIO, según la dirección y el valor que les corresponde para el buen funcionamiento del sistema RF².

D.5. Instalación y configuración de librfid-tool

A continuación se detalla la instalación y configuración de librfid para su uso con OpenPCD y con el lector/escritor RFID diseñado.

Para comenzar se descarga la biblioteca:

```
$ svn checkout https://svn.gnumonks.org/trunk/librfid/
```

Dentro del directorio raíz, se encuentran una serie de directorios con los fuentes. Los más importantes se detallan a continuación:

utils: en este directorio de encuentran las funciones asociadas con la herramienta librfid-tool. Se destaca librfid-tool.c donde se encuentra la función main de la aplicación librfid-tool.

src: en este directorio se encuentran todos los fuentes de la biblioteca librfid.

include: en este directorio se encuentran todos los encabezados de las funciones de la biblioteca librfid.

OpenPCD

Para comenzar se intentó la comunicación desde el PC para luego pasar a la Beagle-board, ya que existe mucha documentación para el primer caso [REF]. En los dos casos fue necesario compilar la biblioteca para utilizar en la arquitectura elegida.

OpenPCD en PC:

Como primer paso, se conecta el OpenPCD al PC. Para saber si el dispositivo es detectado por la PC, es necesario lo siguiente:

```
$ lsusb
```

Aquí debe aparecer (entre otros dispositivos conectados) un dispositivo con un identificador ID 16c0:076b (vendor:product).

Nota: En caso de que este dispositivo no aparezca, se debe usar un HUB con alimentación externa.

Compilando librfid:

Para compilar la biblioteca son necesarios los siguientes paquetes: libtool, libusb-dev, libcurl-dev (libcurl4gnutls-dev instalado en este caso).

Se debe ir hasta el directorio donde se encuentra la biblioteca descargada (librfid por ejemplo) y escribir lo siguiente:

```
$ cd librfid  
$ ./configure  
$ make  
$ sudo make install
```

OpenPCD en la Beagleboard:

Antes que nada se debe conectar el OpenPCD a la Beagleboard y verificar que es detectado al igual que en la instalación en el PC:

```
$ lsusb
```

Se tuvo que utilizar un HUB con alimentación externa ya que la Beagleboard no logró ver al OpenPCD.

La compilación se realiza directamente en la Beagleboard.

Al igual que en el caso de la compilación para el uso en un PC, se debieron obtener los paquetes libtool, libusb-dev, libcurl-dev. Para instalar los paquetes se utilizó el siguiente comando:

```
$ opkg install "paquete"
```

Se copia la biblioteca (sin compilar) a la Beagleboard y luego:

```
$ cd librfid  
$ ./configure  
$ make  
$ make install
```

Lector-escritor RFID en la Beagleboard:

El lector/escritor RFID se conecta a la Beagleboard a través de una interfaz SPI. Para habilitar las funcionalidades de la librfid asociadas con la comunicación SPI, es necesario indicar al configurar, que se quiere utilizar esta interfaz a través de la opción --enable-spidev. La compilación se realiza en la Beagleboard:

```
$ cd librfid  
$ ./configure --enable-spidev  
$ make  
$ make install
```

Con esto se logró compilar la librfid con opciones para utilizar la interfaz SPI

A continuación se explica el uso de librfid-tool:

librfid-tool -[opción]

Dentro de las opciones:

s: realiza una búsqueda de tarjetas RFID hasta encontrar una.

S: loop infinito con la opción “s”, muestra información sobre la tarjeta RFID encontrada en cada paso.

p: especifica el protocolo RFID a utilizar, entre las opciones se encuentran “tcl”, “mifare-classic” y “mifare-ultralight”.

l: especifica el protocolo de capa 2 a utilizar, entre las opciones se encuentran “ISO14443a”, “ISO14443b” y “ISO15693”.

h: ayuda.

Ejemplos de uso recomendados:

```
$ librfid-tool -p mifare-classic
```

Si encuentra una tarjeta con protocolo Mifare-classic, devuelve la lectura completa de los bloques de memoria de la tarjeta.

```
$ librfid-tool -S
```

Devuelve el UID y el protocolo soportado por la tarjeta.

D.6. Depuración de código

A continuación se detallan algunos comandos útiles para el depurado de código con GDB:

breakpoint: para colocar un breakpoint. En general se lo llama seguido del nombre de una función de la aplicación.

print: seguido del nombre de una variable, muestra el contenido de la variable durante el proceso de depuración. Si la variable es local a alguna función, el valor de la variable se pierde al salir de la función.

next o “n”: sirve para ir línea a línea en modalidad step-over (sin entrar a las funciones).

step o “s”: sirve para ir línea a línea en modalidad step-into (entrando a las funciones).

backtrace o “bt”: despliega el stack de llamadas a funciones, sirve para saber por donde se pasó y donde se está.

D.7. Depuración remota

A continuación se detalla la configuración de depuración remota con conexión ethernet.

En la Beagleboard:

```
$ gdbserver localhost: <puerto> <ejecutable Aplicación> <argumentos>
```

En el pc de desarrollo:

```
$ gdb  
$ target remote <ip_Beagleboard>:<puerto>
```

<puerto>: número del puerto por el que se conectan. El puerto no debe estar ocupado por otra aplicación. Por ejemplo un número > 2000 funciona correctamente.

<ejecutable Aplicación> <argumentos>: es el nombre de la aplicación que se quiere depurar y si la aplicación necesita algún argumento para ejecutarse correctamente también se deben agregar.

<ip_Beagleboard>: es la ip de la Beagleboard.

D.8. Preparación de la memoria SD

Se van a necesitar dos particiones, una FAT32 de más de 32MB y una ext3 (el resto del espacio), la primer partición (FAT32) debe ser booteable. Hay dos maneras de hacerlo, una es usando el GParted (modo gráfico) y la otra es por consola.

D.8.1. Formateo de la memoria SD

Formateo utilizando GParted

Se instala el GParted:

```
$ sudo apt-get install gparted
```

Se crean las particiones y se les da formato. Se da click derecho sobre la partición FAT32, gestionar opciones y se marca boot. también se puede dar un nombre a cada partición.

Formateo manual de la memoria SD

El siguiente procedimiento se realiza por consola.

Antes que nada se debe saber dónde está la SD (memory card). Para saberlo se ejecuta en consola:

```
$ dmesg
```

Se obtiene en parte lo siguiente:

```
sd 3:0:0:0: Attached scsi generic sg1 type 0
sd 3:0:0:0: [sdb] 7729152 512-byte logical blocks:
(3.95 GB/3.68 GiB)
sd 3:0:0:0: [sdb] Write Protect is off
sd 3:0:0:0: [sdb] Mode Sense: 03 00 00 00
sd 3:0:0:0: [sdb] Assuming drive cache: write through
sd 3:0:0:0: [sdb] Assuming drive cache: write through
sdb: sdb1
sd 3:0:0:0: [sdb] Assuming drive cache: write through
sd 3:0:0:0: [sdb] Attached SCSI removable disk
```

En este caso la SD está en /dev/sdb

Se borran las particiones:

```
$ sudo fdisk /dev/sdb
```

```
Command (m for help): o
Building a new DOS disklabel. Changes will remain in
memory only, until you decide to write them. After
that, of course, the previous content won't be
recoverable.
Warning: invalid flag 0x0000 of partition table 4
will be corrected by w(rite)
```

Información de la memoria:

```
Command (m for help): p
Disk /dev/sdb: 3957 MB, 3957325824 bytes
....
```

Recordar la cantidad de bytes.

Se entra en Expert mode:

```
Command (m for help): x
```

Se quiere setear la geometría de la siguiente forma: 255 heads, 63 sectors, y se calcula el número de cilindros requeridos por la tarjeta:

$C = \text{trunk}(B/255/63/512)$, donde C:cilindros, B:número de bytes de la tarjeta (anotado previamente) En este caso: $C = \text{trunk}(481,117) = 481$

```
Expert command (m for help): h
Number of heads (1-256, default 4): 255
Expert command (m for help): s
Number of sectors (1-63, default 62): 63
Warning: setting sector offset for DOS compatibility
Expert command (m for help): c
Number of cylinders (1-1048576, default 1011): 481
```

Se va a crear la partición FAT32 y a marcarla como booteable:

```
Expert command (m for help): r
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-481, default 1): (Enter)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-481,
default 481): +50
//son como 400MB
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): c
Changed system type of partition 1 to c (W95 FAT32
(LBA))
Command (m for help): a
Partition number (1-4): 1
```

Se crea la segunda partición:

```
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (52-481, default 52): (Enter)
Using default value 52
Last cylinder or +size or +sizeM or +sizeK (52-481,
default 481):(Enter)
Using default value 481
```

Se imprime para ver como va todo:

```
Command (m for help): p
Disk /dev/sdb: 3957 MB, 3957325824 bytes
255 heads, 63 sectors/track, 481 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device Boot
/dev/sdb1 *
Start
1
End
Blocks Id System
51
409626 c W95 FAT32 (LBA)
/dev/sdb2
52
481
3453975 83 Linux
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with
error 16: Device or resource busy. The kernel still uses
the old table. The new table will be used at the next reboot.
WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for
additional information.
Syncing disks.
```

Se desmontan las particiones creadas:

```
$ umount /dev/sdb1  
$ umount /dev/sdb2
```

Puede que diga que ya no está montado, en ese caso se sigue igualmente.

Luego hay que formatear las particiones:

```
$ sudo mkfs.msdos -F 32 /dev/sdb1 -n nombre  
mkfs.msdos 3.0.7 (24 Dec 2009)  
  
$ sudo mkfs.ext3 /dev/sdb2 -L otroNombre  
  
mke2fs 1.41.11 (14-Mar-2010)  
Filesystem label=  
OS type: Linux  
Block size=4096 (log=2)  
Fragment size=4096 (log=2)  
216000 inodes, 863493 blocks  
43174 blocks (5.00%) reserved for the super user  
First data block=0  
Maximum filesystem blocks=402653184  
27 block groups  
32768 blocks per group, 32768 fragments per group  
8000 inodes per group  
Superblock backups stored on blocks:  
32768, 98304, 163840, 229376, 294912, 819200  
Writing inode tables: done  
Creating journal (16384 blocks): done  
Writing superblocks and filesystem accounting information: done
```

Si dice que no encuentra el dispositivo, se saca la SD y se vuelve a colocar.

D.8.2. Copia de archivos a la SD

A partir de aquí se supone que la partición FAT32 tiene el nombre boot y la partición ext3 tiene el nombre rootFS y que los archivos generados se encuentran en el directorio beagleboard ubicado en el home del usuario (~beagleboard es su ubicación).

Se realiza por consola. Los primeros tres archivos van a la partición FAT32 y el fileSystem (sin descomprimir) va a la partición ext3.

```
$ cd ~/beagleboard
```

```
$ sudo cp MLO /media/boot
```

Nota: El MLO debe ser el primer archivo a copiar.

```
$ sudo cp u-boot.bin uImage /media/boot
```

```
$ cp fileSystem.tar.gz /media/rootFS
```

Se descomprime el fileSystem en la partición ext3.

```
$ cd /media/rootFS
```

```
$ sudo tar -xvzf fileSystem.tar.gz
```

Esto demora un rato.

```
$ sudo rm -f fileSystem.tar.gz
```

Se borra el archivo original.

Se desmonta la memoria SD y queda lista para ser probada.

```
$ sync
```

```
$ umount /media/boot
```

```
$ umount /media/rootFS
```

D.9. Configuración en el PC para conexión serial con la Beagleboard

Para la comunicación con la Beagleboard se utilizó el programa minicom el cual se maneja desde consola (si se quiere uno con ambiente gráfico, se recomienda cutecom).

Para configurar minicom se accede a la consola y se escribe lo siguiente:

```
$ sudo minicom -s
```

Se accede a “Configuración de la puerta serial” y ahí se debe configurar como sigue:

A - Dispositivo Serial: /dev/ttyUSB0 (si no se encuentra nada, probar con ttyUSB1)

B - Localización del Archivo de Bloqueo: /var/lock

C - Programa de Acceso:

D - Programa de Salida:

E - Bps/Paridad/Bits: 115200 8N1

F - Control de Flujo por Hardware: No

G - Control de Flujo por Software: No

Luego, para guardar las opciones elegidas se marca la opción “Salvar configuración como dfl”.

D.10. printenv

```
OMAP3 beagleboard.org # printenv
bootdelay=3
baudrate=115200
loadaddr=0x80200000
rdaddr=0x81600000
usbtty=cdc_acm
console=ttyS2,115200n8
optargs=
bootscr=boot.scr
camera=1bcm3m1
vram=12M
dvimode=640x480MR-16@60
defaultdisplay=dvi
```

```
mmcdev=1
mmcroot=/dev/mmcblk0p2 rw
mmcrootfstype=ext3 rootwait
nandroot=/dev/mtdblock4 rw
nandrootfstype=jffs2
ramroot=/dev/ram0 rw
ramrootfstype=ext2
mmcargs=setenv bootargs console=${console} ${optargs}
mpurate=${mpurate} buddy=${buddy} camera=${camera}
vram=${vram} omapfb
nandargs=setenv bootargs console=${console} ${optargs}
mpurate=${mpurate} buddy=${buddy} camera=${camera}
vram=${vram} omapfb
loadbootscript=fatload mmc ${mmcdev} ${loadaddr} ${bootscr}
ramargs=setenv bootargs console=${console} ${optargs}
mpurate=${mpurate} buddy=${buddy} camera=${camera}
vram=${vram} omapfb
loadramdisk=fatload mmc ${mmcdev} ${rdaddr} ramdisk.gz
bootscript=echo Running bootscript from mmc ...;
source ${loadaddr}
loaduimage=fatload mmc ${mmcdev} ${loadaddr} uImage
mmcboot=echo Booting from mmc ...; run mmcargs;
bootm ${loadaddr}
nandboot=echo Booting from nand ...; run nandargs;
nand read ${loadaddr} 280000 400000; bootm ${loadaddr}
ramboot=echo Booting from ramdisk ...; run ramargs;
bootm ${loadaddr}
dieid#=0968000400000000040365fa1301901a
mtdid=nand0=nand
stdin=serial
stdout=serial
stderr=serial
bootcmd=mmc init;fatload mmc 0 80300000 uImage;bootm 80300000
bootargs=console=ttyS2,115200n8 root=/dev/mmcblk0p2 rw rootwait
buddy=unknown
beaglerev=Cx
mpurate=720
```

D.11. Ejemplo de arranque del sistema

```
U-Boot 2009.11-rc1-00601-g3aa4b51 (Jan 05 2010 - 20:56:38)
OMAP3530-GP ES3.1, CPU-OPP2 L3-165MHz
OMAP3 Beagle board + LPDDR/NAND
I2C: ready
DRAM: 256 MB
NAND: 256 MiB
In: serial
Out: serial
Err: serial
Board revision C4
Die ID #096800040000000040365fa1301901a

Hit any key to stop autoboot: 0
mmc1 is available
reading uImage
3194180 bytes read
## Booting kernel from Legacy Image at 80300000 ...
Image Name: Angström/2.6.32/beagleboard
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 3194116 Bytes = 3 MB
Load Address: 80008000
Entry Point: 80008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
OK
Starting kernel ...

...
...
...
```

The Angström Distribution beagleboard ttyS2
Angström 2009.X-test-20100104 beagleboard ttyS2
beagleboard login:

Aquí el usuario es root, sin contraseña.

D.12. Conexión de Beagleboard a internet

En el PC de desarrollo se debe configurar lo que sigue:

```
$ sudo gedit /etc/sysctl.conf
```

Se debe descomentar la línea que dice net.ipv4.ip_forward=1 para habilitar el “forwarding” de paquetes.

```
$ sudo sysctl -p  
$ sudo cat /proc/sys/net/ipv4/ip_forward
```

Lo anterior es para verificar que los cambios fueron hechos de manera correcta.

```
$ sudo iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE  
$ sudo iptables --append FORWARD --in-interface usb0 -j ACCEPT
```

Cuando se refiere a --out-interface, se refiere a la interfaz del PC por la que se accede a internet.

D.13. Instalación de PCSC-Lite, CCID y pcsc-tools, y agregado de lector serial

A continuación se describe el proceso de instalación en la Beagleboard, el procedimiento para un PC común es el mismo.

Primero se bajan los fuentes asociados a la PCSC-Lite [103], CCID [103] y pcsc-tools [104]. Luego se envían a la Beagleboard.

Paquetes necesarios en la Beagleboard: libusb-1.0-dev.

```
$ opkg install libusb-1.0-dev
```

PCSC-Lite:

Para la instalación de la PCSC-Lite se usó la biblioteca libusb, aunque se puede hacer con el uso de la biblioteca libudev. Se supone que los fuentes se encuentran en el directorio pcsclite.

```
$ cd pcselite  
$ ./configure --disable-libudev LIBUSB_CFLAGS=-I/usr/include/libusb-1.0 LIBUSB_LIBS=-  
L/usr/lib -lusb-1.0" --enable-ipcdir=/var/run/pcscd --enable-usbdropdir=/usr/drivers  
--enable-confdir=/etc/reader.conf.d --prefix=/usr
```

LIBUDEV_CFLAGS: Dirección donde se encuentran los .h asociados con libusb.

LIBUDEV_LIBS: Dirección donde está libusb.

--enable-ipcdir: Dirección donde se guardan archivos relacionados con la comunicación con el demonio de la pcselite (pcscd).

--enable-usbdropdir: Dirección donde se guardan drivers usb (asociado con la instalación del ccid).

--enable-confdir: Dirección donde se guardan archivos relacionados con la configuración serial.

--prefix: Dirección a partir de la cual se instalan los archivos necesarios para el correcto funcionamiento de la biblioteca.

```
$ make  
$ make install
```

CCID:

Se supone que los fuentes se encuentran en el directorio ccid.

```
$ cd ccid  
$ ./configure LIBUSB_CFLAGS=-I/usr/include/libusb-1.0 LIBUSB_LIBS=L/usr/lib  
-lusb-1.0"PCSC_CFLAGS=-I/usr/include/PCSC/ PCSC_LIBS=L/usr/lib -lpcselite" --enable-  
usbdropdir=/usr/drivers --prefix=/usr  
$ make  
$ make install
```

PCSC-TOOLS:

Se supone que los fuentes se encuentran en el directorio pcsc-tools.

```
$ make  
$ make install
```

Pasos para agregar el lector de tarjetas serial, en la biblioteca pcsclite:

Se crea el directorio para drivers de lectores (es posible que ya haya sido creado por CCID al ser instalado).

```
$ mkdir /usr/drivers/
```

Dentro de este directorio se ubica la biblioteca dinámica del driver para el lector de tarjetas.

```
$ scp rf2_sc.so root@172.16.1.14:/usr/drivers
```

Luego es necesario crear el archivo de configuración donde se encuentran los parámetros del lector serial. Es posible conocer el directorio donde debe ubicarse el archivo de configuración ejecutando el comando:

```
$ pcscd -v
```

Se crea el archivo:

```
$ mkdir /etc/reader.conf.d/
$ touch /etc/reader.conf.d/reader.conf
```

Se edita el archivo reader.conf agregando las siguientes líneas:

```
#####
# Configuration file for pcsc-lite
# Reader Projeto-RF2 FING-UDELAR

FRIENDLYNAME    rf2_sc
DEVICENAME       /dev/ttyS1
LIBPATH          /usr/drivers/rf2_sc.so
CHANNELID        1
#####
```

CHANNELID indica el número de puerto donde se encuentra el dispositivo:

- 1 ---> /dev/pcsc/1
- 2 ---> /dev/pcsc/2
- 3 ---> /dev/pcsc/3

El puerto serial se encuentra realmente en /dev/ttysi, por tanto se debe realizar un enlace simbólico /dev/pcsc/i —> /dev/ttysi . Para efectuar lo anterior se crea el directorio pcsc bajo dev:

```
$ mkdir /dev/pcsc  
$ cd /dev/pcsc  
$ ln -s /dev/ttysi i
```

donde i es el número de puerto serial que usado por el lector de tarjetas.

Se reinicia el demonio pcscd y se ejecuta la herramienta pcsc_scan para que muestre el dispositivo:

```
...  
Scanning present readers...  
0: rf2_sc 00 00  
  
...  
Reader 0: rf2_sc 00 00
```

D.14. Pruebas sobre las interfaces

D.14.1. led.c

```
#include <string.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <unistd.h>  
FILE *fp;  
  
int main(int argc, char** argv)  
{  
    printf("\n*****\n"  
    "* Welcome to PIN Blink program *\n"  
    "* ....blinking pin 13 on expansion port *\n"  
    "* ....rate of 1 Hz..... *\n"  
    "*****\n");  
    //create a variable to store whether we are  
    //sending a '1' or a '0'  
    char set_value[4];  
    //Integer to keep track of whether we want on or off
```

```
int toggle = 0;
//Using sysfs we need to write "134"
//to /sys/class/gpio/export
//This will create the folder /sys/class/gpio/gpio134
if ((fp = fopen("/sys/class/gpio/export", "ab")) == NULL)
{
    printf("Cannot open export file.\n");
    exit(1);
}
//Set pointer to begining of the file
rewind(fp);
//Write our value of "134" to the file
strcpy(set_value,"134");
fwrite(&set_value, sizeof(char), 3, fp);
fclose(fp);
printf("...export file accessed, new pin now accessible\n");
//SET DIRECTION
//Open the LED's sysfs file in binary for
//reading and writing, store file pointer in fp
if ((fp = fopen("/sys/class/gpio/gpio134/direction", "rb+"))
== NULL)
{
    printf("Cannot open direction file.\n");
    exit(1);
}
//Set pointer to begining of the file
rewind(fp);
//Write our value of "out" to the file
strcpy(set_value,"out");
fwrite(&set_value, sizeof(char), 3, fp);
fclose(fp);
printf("...direction set to output\n");
//SET VALUE
//Open the LED's sysfs file in binary for
//reading and writing, store file pointer in fp
if ((fp = fopen("/sys/class/gpio/gpio134/value", "rb+"))
== NULL)
{
    printf("Cannot open value file.\n");
    exit(1);
}
//Set pointer to begining of the file
rewind(fp);
//Write our value of "1" to the file
```

```
strcpy(set_value, "1");
fwrite(&set_value, sizeof(char), 1, fp);
fclose(fp);
printf("...value set to 1...\n");
//Run an infinite loop - will require
//Ctrl-C to exit this program
while(1)
{
    //Set it so we know the starting value
    //in case something above doesn't leave it as 1
    strcpy(set_value, "1");
    if ((fp = fopen("/sys/class/gpio/gpio134/value", "rb+")) == NULL)
    {
        printf("Cannot open value file.\n");
        exit(1);
    }
    toggle = !toggle;
    if(toggle)
    {
        //Set pointer to begining of the file
        rewind(fp);
        //Write our value of "1" to the file
        strcpy(set_value, "1");
        fwrite(&set_value, sizeof(char), 1, fp);
        fclose(fp);
        printf("...value set to 1...\n");
    }
    else
    {
        //Set pointer to begining of the file
        rewind(fp);
        //Write our value of "0" to the file
        strcpy(set_value, "0");
        fwrite(&set_value, sizeof(char), 1, fp);
        fclose(fp);
        printf("...value set to 0...\n");
    }
    //Pause for one second
    sleep(1);
}
return 0;
}
```

D.14.2. uart.c

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <stdio.h>

#define BAUDRATE B9600
#define SERIALPORT "/dev/ttyS1"
#define _POSIX_SOURCE 1 /* POSIX compliant source */
#define FALSE 0
#define TRUE 1
#define BUFFWRITE 10
#define BUFFREAD 255

int fd; //descriptor del archivo asociado con tty
struct termios oldtio,newtio;

//inicialización
int tty_init (void)
{
    //abrimos el puerto ttyS1
    fd = open(SERIALPORT, O_RDWR | O_NOCTTY );
    if(fd<0)
        return 0; //hubo errores
    else
    {
        printf("Puerto %s, abierto. fd: %d\n", SERIALPORT, fd);
        /* save current port settings */
        tcgetattr(fd,&oldtio);
        //bzero(&newtio, sizeof(newtio));
        newtio.c_cflag = BAUDRATE | CS8 | CLOCAL | CREAD;
        newtio.c_iflag = IGNPAR;
        newtio.c_oflag = 0;
        /* set input mode (non-canonical, no echo,...) */
        newtio.c_lflag = 0;
        /* inter-character timer unused */
        newtio.c_cc[VTIME] = 0;
        /* bloqueo la lectura hasta que llegue 1 caracter */
        newtio.c_cc[VMIN] = 1;
        tcflush(fd, TCIFLUSH);
        tcsetattr(fd,TCSANOW,&newtio);
        return 1; //no hubo errores
    }
}
```

```
        }
    }

//envio de un caracter por UART
void escribir (char s[10])
{
    write(fd, s, BUFFWRITE); //escribe el string en UART-Tx
    printf("dato enviado: %s\n", s);
}

//lectura de caracter por UART
char leer (void)
{
    char r[BUFFREAD];
    if(read(fd, r, BUFFREAD) == -1)
        return 0; //No hay datos o se incluyen ceros
    else
        return r[0]; //no hubo errores, se retorna el valor leido
}

int main (void)
{
    //cantidad de caracteres leidos por UART
    int UART_get_return;
    //indica si se pudo abrir el puerto o no
    int UART_init_return;
    char buff[BUFFREAD];
    //inicializacion de la UART
    UART_init_return = tty_init();

    //no se pudo abrir el puerto
    if (UART_init_return == 0)
        printf("no se pudo abrir el puerto ttyS1.\n");
    else //se pudo abrir el puerto
    {
        char str[10]={'a','s','d','f','g','h','j','k','l','t'};
        escribir(str);
        UART_get_return = read(fd,buff,BUFFREAD);

        // 0 = no se encontraron datos en Rx
        if (UART_get_return == 0)
            printf("no hay datos para leer\n");
        else //1 = dato encontrado
        {
```

```
    buff[UART_get_return]=0;
    printf("dato leido: %s:%d\n", buff, UART_get_return);
}
tcsetattr(fd, TCSANOW, &oldtio);
}
```

Apéndice E

Hojas de datos

A continuación se adjuntan algunas de las hojas de datos a las que se hace referencia en el texto. Se encuentran ordenadas de acuerdo al siguiente listado, respetando el orden en el cual se mencionan:

- MIFARE® and I Code CL RC632 Multiple protocol contactless reader IC.
- 8-BIT BIDIRECTIONAL VOLTAGE-LEVEL TRANSLATOR.
- LM1117/LM1117I 800mA Low-Dropout Linear Regulator.
- Transistor 2N3904.
- Transistor 2N3906.
- ESDA6V1W5.
- HD44780U (LCD-II), Dot Matrix Liquid Crystal Display Controller/Driver.
- LCM-S01602DSF-A - hoja de datos del LCD16x2.
- Micore Reader IC Family; Directly Matched Antenna Design.
- Mifare®(14443A) 13,56 MHz RFID Proximity Antennas.
- BeagleBoard System Reference Manual Rev C4, Revision 0.0.
- OMAP 35x Applications Processor Technical Reference Manual.

- Hawkboard Press Release.
- Hawkboard Press Release Solution.
- Build a cheap 13.56MHz MIFARE antenna for the Proxmark.
- Angström Manual - Embedded Power - .

DATA SHEET

mifare[®] & I•CODE

CL RC632

Multiple Protocol Contactless Reader IC

Product Specification

May 2003

Revision 3.0

Confidential

**Philips
Semiconductors**



PHILIPS

Multiple Protocol Contactless Reader IC**CL RC632**

CONTENTS

1	GENERAL INFORMATION	7
1.1	Scope	7
1.2	General Description.....	7
1.3	Features	8
1.4	Ordering Information	8
2	BLOCK DIAGRAM	9
3	PINNING INFORMATION	10
3.1	Pin Configuration.....	10
3.2	Pin Description	11
4	DIGITAL INTERFACE	13
4.1	Overview of Supported µ-Processor Interfaces	13
4.2	Automatic µ-Processor Interface Type Detection	13
4.3	Connection to Different µ-Processor Types	14
4.3.1	Separated Read/Write Strobe	14
4.3.2	Common Read/Write Strobe	15
4.3.3	Common Read/Write Strobe and Hand-Shake Mechanism: EPP	16
4.4	SPI compatible interface	17
5	CL RC632 REGISTER SET	20
5.1	CL RC632 Registers Overview	20
5.1.1	Register Bit Behaviour.....	22
5.2	Register Description	23
5.2.1	Page 0: Command and Status	23
5.2.2	Page 1: Control and Status	31
5.2.3	Page 2: Transmitter and Control	38
5.2.4	Page 3: Receiver and Decoder Control	44
5.2.5	Page 4: RF-Timing and Channel Redundancy	51
5.2.6	Page 5: FIFO, Timer and IRQ- Pin Configuration	58
5.2.7	Page 6: RFU.....	63
5.2.8	Page 7: Test Control	64
5.3	CL RC632 Register Flags Overview	68
5.4	Modes of Register Addressing	72
5.4.1	Paging Mechanism.....	72
5.4.2	Dedicated Address Bus.....	72
5.4.3	Multiplexed Address Bus.....	72
6	MEMORY ORGANISATION OF THE E²PROM	73
6.1	Diagram of the E ² PROM Memory Organisation.....	73
6.2	Product Information Field (Read Only).....	74

Multiple Protocol Contactless Reader IC**CL RC632**

6.3	Register Initialisation Files (Read/Write).....	75
6.3.1	Start Up Register Initialisation File (Read/Write).....	75
6.3.2	Shipment Content of Start Up Register Initialisation File	76
6.3.3	Register Initialisation File (Read/Write).....	77
6.3.4	Content of I•CODE1 and ISO15693 SStart Up Register Values.....	78
6.4	Crypto1 Keys (Write Only).....	78
6.4.1	Key Format.....	79
6.4.2	Storage of Keys in the E ² PROM	79
7	FIFO BUFFER.....	80
7.1	Overview.....	80
7.2	Accessing the FIFO Buffer	80
7.2.1	Access Rules.....	80
7.3	Controlling the FIFO-Buffer	81
7.4	Status Information about the FIFO-Buffer	81
7.5	Register Overview FIFO Buffer	82
8	INTERRUPT REQUEST SYSTEM	83
8.1	Overview.....	83
8.1.1	Interrupt Sources Overview.....	83
8.2	Implementation of Interrupt Request Handling.....	84
8.2.1	Controlling Interrupts and their Status.....	84
8.2.2	Accessing the Interrupt Registers	84
8.3	Configuration of Pin IRQ	84
8.4	Register Overview Interrupt Request System.....	85
9	TIMER UNIT	86
9.1	Overview.....	86
9.2	Implementation of the Timer Unit	87
9.2.1	Block Diagram	87
9.2.2	Controlling the Timer Unit.....	88
9.2.3	Timer Unit Clock and Period	88
9.2.4	Status of the Timer Unit.....	89
9.2.5	TimeSlotPeriod	89
9.3	Usage of the Timer Unit	90
9.3.1	Time-Out- and Watch-Dog-Counter	90
9.3.2	Stop Watch.....	91
9.3.3	Programmable One-Shot Timer	91
9.3.4	Periodical Trigger	91
9.4	Register Overview Timer Unit	92
10	POWER REDUCTION MODES	93
10.1	Hard Power Down	93
10.2	Soft Power Down.....	93

Multiple Protocol Contactless Reader IC**CL RC632**

10.3	Stand By Mode.....	94
10.4	Receiver Power Down.....	94
11	START UP PHASE.....	95
11.1	Hard Power Down Phase	95
11.2	Reset Phase.....	95
11.3	Initialising Phase	95
11.4	Initialising the Parallel Interface-Type	96
12	OSCILLATOR CIRCUITRY	97
13	TRANSMITTER PINS TX1 AND TX2	98
13.1	Configuration of TX1 and TX2.....	98
13.2	Operating Distance versus Power Consumption	99
13.3	Antenna Driver Output Source Resistance	99
13.3.1	Source Resistance Table	100
13.3.2	Formula for the Source Resistance.....	101
13.3.3	Calculating the Effective Source Resistance	101
13.4	Pulse Width	102
14	RECEIVER CIRCUITRY	103
14.1	General.....	103
14.2	Block Diagram	103
14.3	Putting the Receiver into Operation	104
14.3.1	Automatic Clock-Q Calibration	104
14.3.2	Amplifier.....	105
14.3.3	Correlation Circuitry.....	106
14.3.4	Evaluation and Digitizer Circuitry	106
15	SERIAL SIGNAL SWITCH	107
15.1	General.....	107
15.2	Block Diagram	107
15.3	Registers Relevant for the Serial Signal Switch.....	108
15.3.1	Active Antenna Concept.....	109
15.3.2	Driving Two RF-Parts	109
16	MIFARE® HIGHER BAUDRATES	110
17	ISO14443 B.....	111
18	CL RC632 COMMAND SET	112
18.1	General Description.....	112
18.2	General Behaviour	112
18.3	CL RC632 Commands Overview	112
18.3.1	Basic States	114
18.3.2	StartUp Command 3F _{hex}	114

Multiple Protocol Contactless Reader IC

CL RC632

18.3.3	Idle Command 00 _{hex}	114
18.4	Commands for ISO14443 A Card Communication	115
18.4.1	Transmit Command 1A _{hex}	115
18.4.2	Receive Command 16 _{hex}	119
18.4.3	Transceive Command 1E _{hex}	122
18.4.4	States of the Card Communication	122
18.4.5	State Diagram for the Card Communication	123
18.5	Commands for I•CODE1 and ISO15693 Label Communication	124
18.5.1	Transmit Command 1A _{hex}	124
18.5.2	Receive Command 16 _{hex}	126
18.5.3	Transceive Command 1E _{hex}	128
18.5.4	States of the Label Communication	128
18.5.5	State Diagram for the Label Communication	128
18.6	Commands to Access the E ² PROM	130
18.6.1	WriteE2 Command 01 _{hex}	130
18.6.2	ReadE2 Command 03 _{hex}	132
18.7	Diverse Commands.....	132
18.7.1	LoadConfig Command 07 _{hex}	132
18.7.2	CalcCRC Command 12 _{hex}	133
18.8	Error Handling during Command Execution.....	134
18.9	MIFARE® Classic Security Commands	135
18.9.1	LoadKeyE2 Command 0B _{hex}	135
18.9.2	LoadKey Command 19 _{hex}	135
18.9.3	Authent1 Command 0C _{hex}	136
18.9.4	Authent2 Command 14 _{hex}	136
19	MIFARE® CLASSIC AUTHENTICATION AND CRYPTO1.....	137
19.1	General.....	137
19.2	Crypto1 Key Handling	137
19.3	Performing MIFARE® Classic Authentication.....	138
20	TYPICAL APPLICATION.....	139
20.1	Circuit Diagram.....	139
20.2	Circuit Description	140
20.2.1	EMC Low Pass Filter.....	140
20.2.2	Antenna matching	140
20.2.3	Receiving Circuit	141
20.2.4	Antenna Coil.....	141
21	TEST SIGNALS	142
21.1	General.....	142
21.2	Measurements Using the Serial Signal Switch	142
21.2.1	Tx-Control.....	143
21.2.2	Rx-control	144

Multiple Protocol Contactless Reader IC**CL RC632**

21.3	Analog Test-Signals	145
21.4	Digital Test-Signals	146
21.5	Examples of ISO14443A Analog- and Digital Test Signals	147
21.6	Examples of I•CODE1 Analog- and Digital Test Signals	148
22	ELECTRICAL CHARACTERISTICS	149
22.1	Absolute Maximum Ratings.....	149
22.2	Operating Condition Range.....	149
22.3	Current Consumption	149
22.4	Pin Characteristics	150
22.4.1	Input Pin Characteristics	150
22.4.2	Digital Output Pin Characteristics	151
22.4.3	Antenna Driver Output Pin Characteristics	151
22.5	AC Electrical Characteristics	152
22.5.1	AC Symbols.....	152
22.5.2	AC Operating Specification	153
22.5.3	Clock Frequency	157
23	E²PROM CHARACTERISTICS	158
24	ESD SPECIFICATION	159
25	PACKAGE OUTLINES	160
25.1	SO32	160
26	DISCLAIMERS.....	161
26.1	Life support applications.....	161
26.2	Licence Policy	161
27	REVISION HISTORY	162
27.1	Update from Revision 2.0 to Revision 3.0.....	162
27.2	Versions Up to Revision 3.0.....	162
	Contact Information.....	163

Multiple Protocol Contactless Reader IC

CL RC632

1 GENERAL INFORMATION

1.1 Scope

This document describes the functionality of the CL RC632. It includes the functional and electrical specifications and gives details on how to design-in this device from system and hardware viewpoint.

1.2 General Description

The CL RC632 is member of a new family of highly integrated reader ICs for contactless communication at 13.56 MHz. This reader IC family utilises an outstanding modulation and demodulation concept completely integrated for all kinds of passive contactless communication methods and protocols at 13.56 MHz. The CL RC632 is pin- compatible to the MF RC500, the MF RC530, the MF RC531 and the SL RC 400.

The CL RC632 supports all layers of the ISO14443 including the type A and type B communication scheme. The CL RC632 supports contactless communication using MIFARE® Higher Baudrates. The receiver part provides a robust and efficient implementation of a demodulation and decoding circuitry for signals from ISO14443 compatible transponders.

The digital part handles the complete ISO14443 framing and error detection (Parity & CRC). Additionally it supports the fast MIFARE® Classic security algorithm to authenticate MIFARE® Classic (e.g. MIFARE® Standard, MIFARE® Light) products.

The CL RC632 supports all layers of I•CODE1 and ISO 15693. The receiver part provides a robust and efficient implementation of a demodulation and decoding circuitry for signals from I•CODE1 and ISO 15693 compatible transponders. The digital part handles I•CODE1 and ISO 15693 framing and error detection (CRC).

The internal transmitter part is able to drive an antenna designed for proximity operating distance (up to 100 mm) directly without additional active circuitry.

A comfortable parallel interface, which can be directly connected to any 8-bit µ-Processor gives high flexibility for the reader/terminal design.

Additionally a SPI compatible interface is supported.

Multiple Protocol Contactless Reader IC

CL RC632

1.3 Features

- Highly integrated analog circuitry to demodulate and decode card/label response
- Buffered output drivers to connect an antenna with minimum number of external components
- Proximity operating distance (up to 100 mm)
- Supports ISO 14443 A&B
- Supports MIFARE® Dual Interface Card ICs and supports MIFARE® Classic protocol
- Supports contactless communication with MIFARE® higher baudrates up to 424 kbaud
- Supports I•CODE1 and ISO 15693
- Crypto1 and secure non-volatile internal key memory
- Pin-compatible to the MF RC500, MF RC530, MF RC531 and the SL RC400
- Parallel µ-Processor interface with internal address latch and IRQ line
- SPI compatible interface
- Flexible interrupt handling
- Automatic detection of parallel µ-Processor interface type
- Comfortable 64 byte send and receive FIFO-buffer
- Hard reset with low power function
- Power down mode per software
- Programmable timer
- Unique serial number
- User programmable start-up configuration
- Bit- and byte-oriented framing
- Independent power supply pins for digital, analog and transmitter part
- Internal oscillator buffer to connect 13.56 MHz quartz, optimised for low phase jitter
- Clock frequency filtering
- 3.3 V to 5 V operation for transmitter (antenna driver) in short range and proximity applications
- 3.3 V or 5V operation for the digital part

1.4 Ordering Information

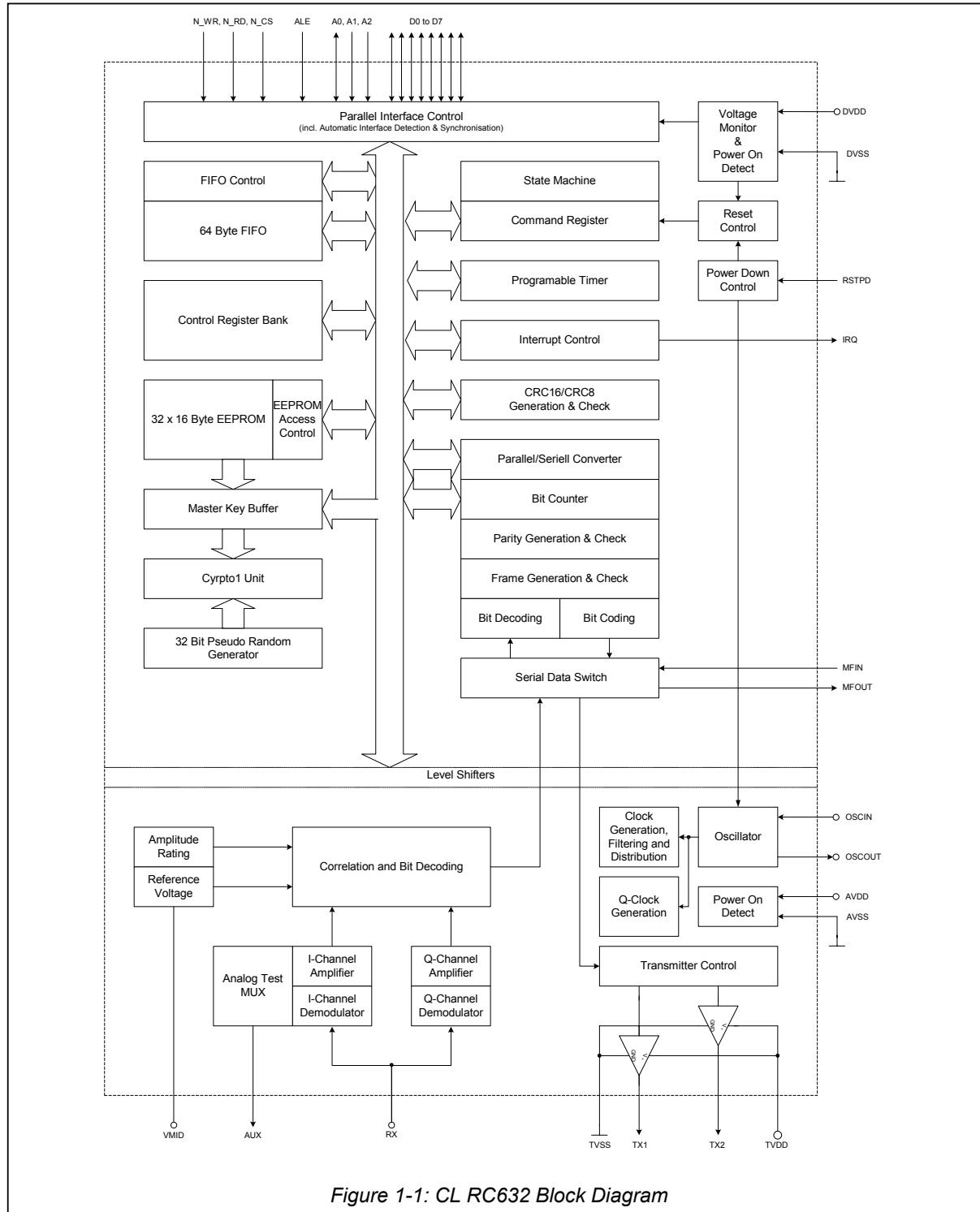
Type Number	Package	
	Name	Description
CL RC632 01T/0FE	SO32	Small Outline Package; 32 leads

Table 1-1: CL RC632 Ordering Information

Multiple Protocol Contactless Reader IC

CL RC632

2 BLOCK DIAGRAM



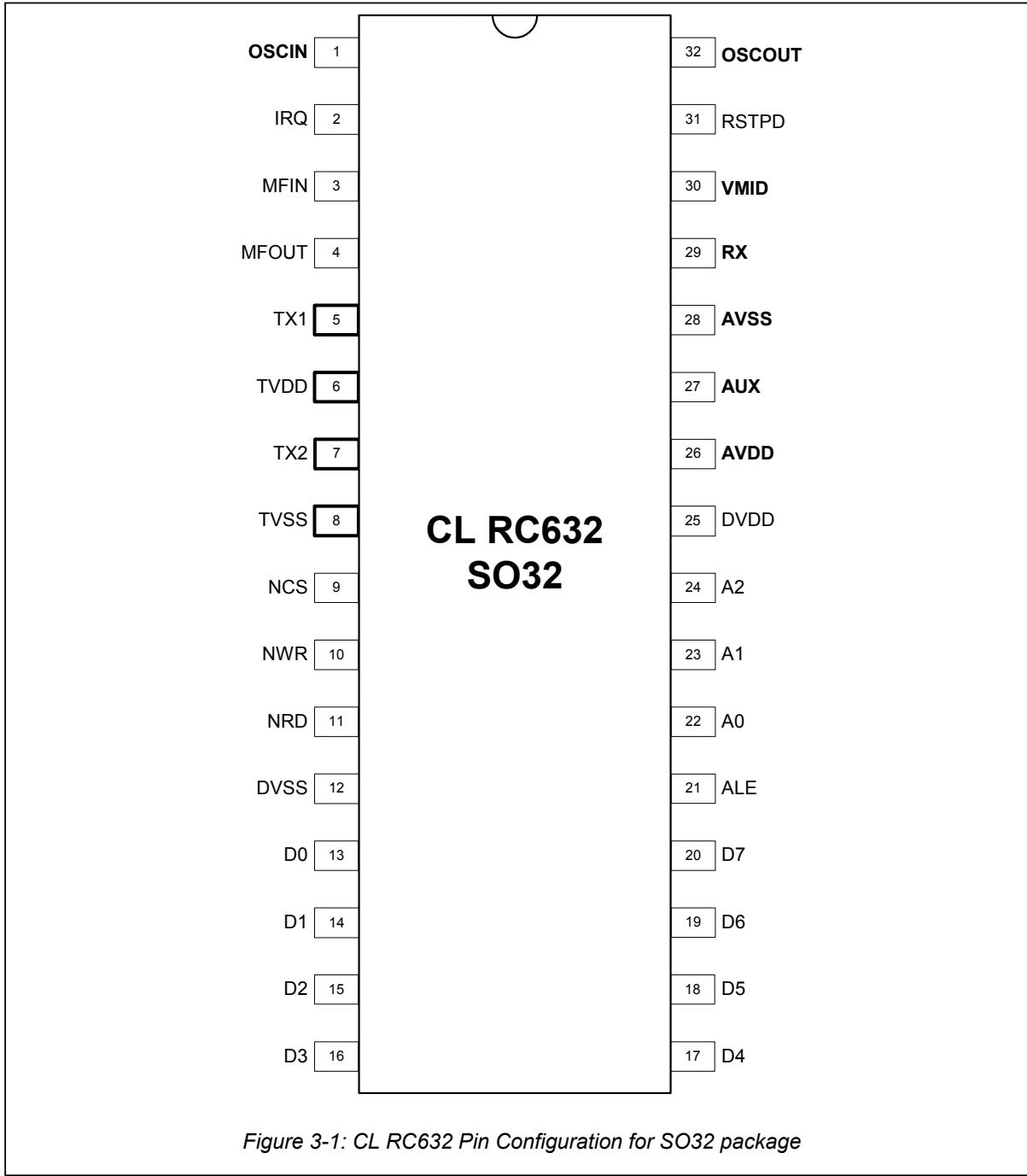
Multiple Protocol Contactless Reader IC

CL RC632

3 PINNING INFORMATION

3.1 Pin Configuration

Pins denoted by bold letters are supplied by AVDD and AVSS. Pins drawn with bold lines are supplied by TVSS and TVDD. All other pins are supplied by DVDD and DVSS.



Multiple Protocol Contactless Reader IC

CL RC632

3.2 Pin Description

Pin Types: I...Input; O...Output; PWR...Power

PIN	SYMBOL	TYPE	DESCRIPTION
1	OSCIN	I	Crystal Oscillator Input: input to the inverting amplifier of the oscillator. This pin is also the input for an externally generated clock ($f_{osc} = 13.56$ MHz).
2	IRQ	O	Interrupt Request: output to signal an interrupt event
3	MFIN	I	MIFARE® Interface Input: accepts a digital, serial data stream according to ISO14443A (MIFARE®)
4 ²	MFOUT	O	MIFARE® Interface Output: delivers a serial data stream according to ISO14443A (MIFARE®) I•CODE Interface Output: delivers a serial data stream according to I•CODE1 and ISO 15693
5	TX1	O	Transmitter 1: delivers the modulated 13.56 MHz energy carrier
6	TVDD	PWR	Transmitter Power Supply: supplies the output stage of TX1 and TX2
7	TX2	O	Transmitter 2: delivers the modulated 13.56 MHz energy carrier
8	TVSS	PWR	Transmitter Ground: supplies the output stage of TX1 and TX2
9	NCS	I	Not Chip Select: selects and activates the µ-Processor interface of the CL RC632
10 ¹	NWR	I	Not Write: strobe to write data (applied on D0 to D7) into the CL RC632 register
	R/NW	I	Read Not Write: selects if a read or write cycle shall be performed.
	nWrite	I	Not Write: selects if a read or write cycle shall be performed
11 ¹	NRD	I	Not Read: strobe to read data from the CL RC632 register (applied on D0 to D7)
	NDS	I	Not Data Strobe: strobe for the read and the write cycle
	nDStrb	I	Not Data Strobe: strobe for the read and the write cycle
12	DVSS	PWR	Digital Ground
13	D0	O	Master In Slave Out (MISO), SPI interface,
13	D0 to D7	I/O	8 Bit Bi-directional Data Bus
20 ¹	AD0 to AD7	I/O	8 Bit Bi-directional Address and Data Bus
21 ¹	ALE	I	Address Latch Enable: signal to latch AD0 to AD5 into the internal address latch when HIGH.
	AS	I	Address Strobe: strobe signal to latch AD0 to AD5 into the internal address latch when HIGH.
	nAStrb	I	Not Address Strobe: strobe signal to latch AD0 to AD5 into the internal address latch when LOW.
	NSS	I	Not Slave Select: strobe for the SPI communication
22 ¹	A0	I	Address Line 0: Bit 0 of register address
	nWait	O	Not Wait: signals with LOW that an access-cycle may started and with HIGH that it may be finished.
	MOSI	I	Master Out Slave In, SPI interface

PIN Description (continued)

¹ These pins offer different functionality according to the selected µ-Processor interface type. For detailed information, refer to chapter 4.² The SL RC400 uses the name SIGOUT for the MFOUT pin. The CLRC 632 functionality includes the test possibilities for the SL RC 400 using the pin MFOUT.

Multiple Protocol Contactless Reader IC

CL RC632

PIN	SYMBOL	TYPE	DESCRIPTION
23	A1	I	Address Line 1: Bit 1 of register address
24 ¹	A2	I	Address Line 2: Bit 2 of register address
	SCK	I	Serial Clock: Clock for the SPI interface
25	DVDD	PIWR	Digital Power Supply
26	AVDD	PWR	Analog Power Supply
27	AUX	O	Auxiliary Output: This pin delivers analog test signals. The signal delivered on this output may be selected by means of the <i>TestAnaOutSel Register</i> .
28	AVSS	PWR	Analog Ground
29	RX	I	Receiver Input: Input pin for the cards response, which is the load modulated 13.56 MHz energy carrier, that is coupled out from the antenna circuit.
30	VMID	PWR	Internal Reference Voltage: This pin delivers the internal reference voltage. <u>Note:</u> It has to be supported by means of a 100 nF block capacitor.
31	RSTPD	I	Reset and Power Down: When HIGH, internal current sinks are switched off, the oscillator is inhibited, and the input pads are disconnected from the outside world. With a negative edge on this pin the internal reset phase starts.
32	OSCOUT	O	Crystal Oscillator Output: Output of the inverting amplifier of the oscillator.

Table 3-1: CL RC632 Pin Description

Multiple Protocol Contactless Reader IC

CL RC632

4 DIGITAL INTERFACE

4.1 Overview of Supported µ-Processor Interfaces

The CL RC632 supports direct interfacing of various µ-Processors. Alternatively the Enhanced Parallel Port (EPP) of personal computers can be connected directly. The following table shows the parallel interface signals supported by the CL RC632:

Bus Control Signals	Bus	Separated Address and Data Bus	Multiplexed Address and Data Bus
Separated Read and Write Strobes	control	NRD, NWR, NCS	NRD, NWR, NCS, ALE
	address	A0, A1, A2	AD0, AD1, AD2, AD3, AD4, AD5
	data	D0 ... D7	AD0 ... AD7
Common Read and Write Strobe	control	R/NW, NDS, NCS	R/NW, NDS, NCS, AS
	address	A0, A1, A2	AD0, AD1, AD2, AD3, AD4, AD5
	data	D0 ... D7	AD0 ... AD7
Common Read and Write Strobe with Handshake (EPP)	control	-	nWrite, nDStrb, nAStrb, nWait
	address		AD0, AD1, AD2, AD3, AD4, AD5
	data		AD0 ... AD7

Table 4-1: Supported µ-Processor Interface Signals

4.2 Automatic µ-Processor Interface Type Detection

After every Power-On or Hard Reset, the CL RC632 also resets its parallel µ-Processor interface mode and checks the current µ-Processor interface type.

The CL RC632 identifies the µ-Processor interface by means of the logic levels on the control pins after the Reset Phase. This is done by a combination of fixed pin connections (see below) and a dedicated initialisation routine (see 11.4).

Multiple Protocol Contactless Reader IC

CL RC632

4.3 Connection to Different µ-Processor Types

The connection to different µ-Processor types is shown in the following table:

CL RC632	Parallel Interface Type				
	Separated Read/Write Strobe		Common Read/Write Strobe		
	Dedicated Address Bus	Multiplexed Address Bus	Dedicated Address Bus	Multiplexed Address Bus	Multiplexed Address Bus with Handshake
ALE	HIGH	ALE	HIGH	AS	nASrb
A2	A2	LOW	A2	LOW	HIGH
A1	A1	HIGH	A1	HIGH	HIGH
A0	A0	HIGH	A0	LOW	nWait
NRD	NRD	NRD	NDS	NDS	nDStrb
NWR	NWR	NWR	R/NW	R/NW	nWrite
NCS	NCS	NCS	NCS	NCS	LOW
D7 ... D0	D7 ... D0	AD7 ... AD0	D7 ... D0	AD7 ... AD0	AD7 ... AD0

Table 4-2: Connection Scheme for Detecting the Parallel Interface Type

4.3.1 SEPARATED READ/WRITE STROBE

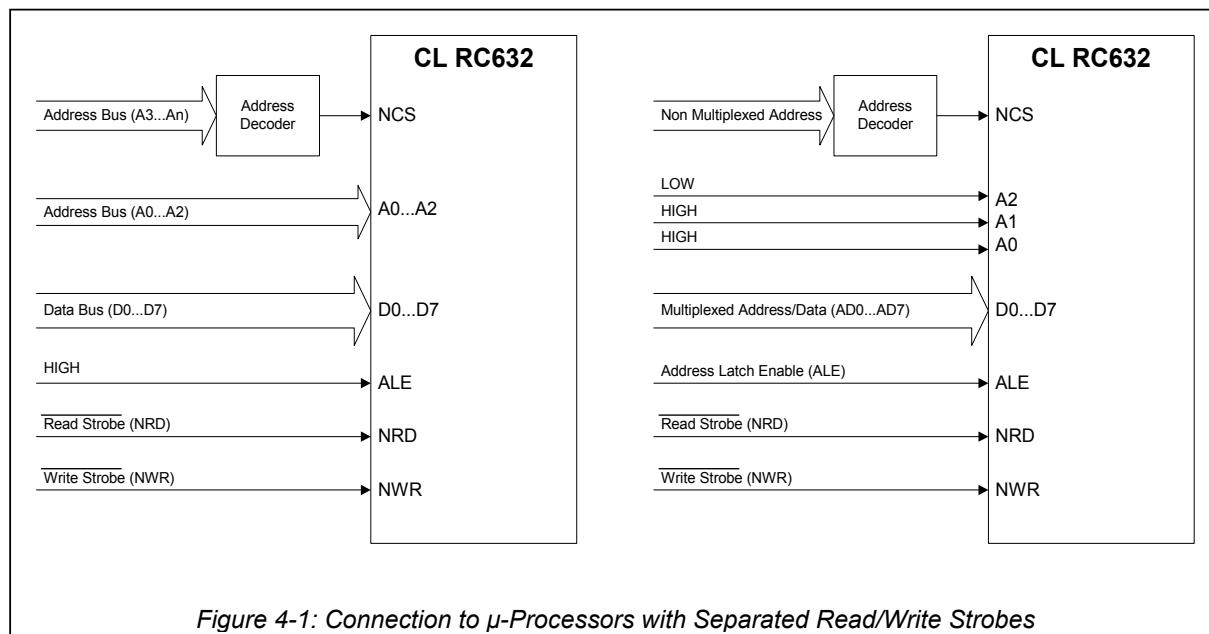


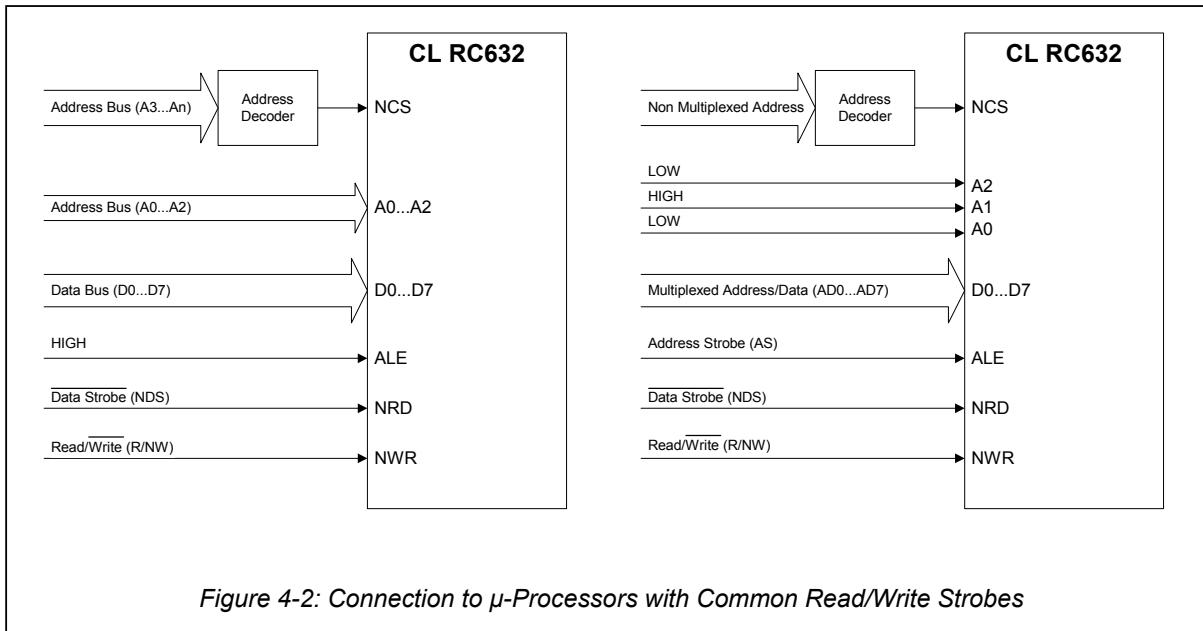
Figure 4-1: Connection to µ-Processors with Separated Read/Write Strobes

For timing specification refer to chapter 22.5.2.1.

Multiple Protocol Contactless Reader IC

CL RC632

4.3.2 COMMON READ/WRITE STROBE



For timing specification refer to chapter 22.5.2.2.

Multiple Protocol Contactless Reader IC

CL RC632

4.3.3 COMMON READ/WRITE STROBE AND HAND-SHAKE MECHANISM: EPP

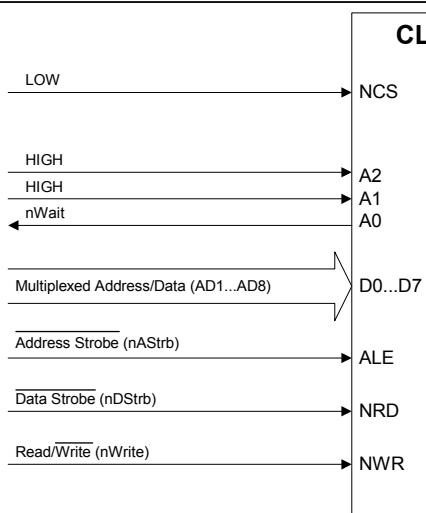


Figure 4-3: Connection to μ -Processors with Common Read/Write Strobes and Hand-Shake

For timing specification refer to chapter 22.5.2.3.

Remarks for EPP:

Although in the standard for the EPP no chip select signal is defined, the N_CS of the CL RC632 allows inhibiting the nDStrb signal. If not used, it shall be connected to DVSS.

After each Power-On or Hard Reset the nWait signal (delivered at pin A0) is high impedance. nWait will be defined at the first negative edge applied to nAStrb after the Reset Phase.

The CL RC632 does not support Read Address Cycle.

Multiple Protocol Contactless Reader IC

CL RC632

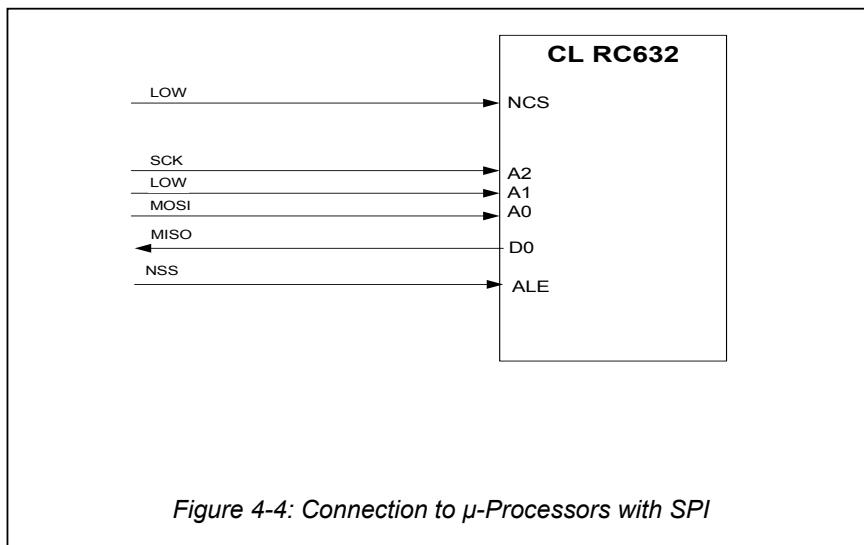
4.4 SPI compatible interface

Additionally the serial peripheral interface (SPI) will be supported. The CL RC632 acts as a slave during the SPI communication. The SPI clock SCK has to be generated by the master. Data communication from the master to the slave uses the line MOSI. Line MISO is used to send data back from the CL RC632 to the master.

CL RC632	SPI Interface
ALE	NSS
A2	SCK
A1	LOW
A0	MOSI
NRD	HIGH
NWR	HIGH
NCS	LOW
D7 ... D1	do not connect
D0	MISO

Table 4-3: SPI compatible interface

The following table shows the µ-Processor connection to the CL RC632 using the SPI interface.



Multiple Protocol Contactless Reader IC**CL RC632**Remarks for SPI:

The implemented SPI interface is according to a standard SPI interface.
The CL RC632 can only be addressed as a slave.

Read data:

To read out data using the SPI interface the following structure has to be used. It is possible to read out up to n-data bytes.
The first sent byte defines both, the mode itself and the address.

	byte 0	byte 1	byte 2	byte n	byte n+1
MOSI	adr 0	adr 1	adr. 2	adr n	00
MISO	XX	data 0	data 1	data n-1	data n

The address byte has to fulfil the following format. The MSB bit of the first byte sets the used mode. To read data from the CL RC632 the MSB bit is set to 1. The bits 6-1 define the address and the last bit should be set to 0.

According to scheme above, the last sent byte has been set to 0.

Address (MOSI)	bit 7, MSB	bit 6 - bit 1	bit 0
byte 0	1	address	RFU (0)
byte 1 to byte n	RFU (0)	address	RFU (0)
byte n+1	0	0	0

Write data:

To write data to the CL RC632 using the SPI interface the following structure has to be used. It is possible to write out up to n-data bytes.

The first send byte defines both, the mode itself and the address.

	byte 0	byte 1	byte 2	byte n	byte n+1
MOSI	adr	data 0	data 1	data n-1	data n
MISO	XX	XX	XX	XX	XX

The address byte has to fulfil the following format. The MSB bit of the first byte sets the used mode. To write data to the CL RC632 the MSB bit is set to 0. The bits 6-1 define the address and the last bit should be set to 0.

Multiple Protocol Contactless Reader IC**CL RC632**

The SPI write mode writes all data to the same address as defined in byte 0.
This allows an effective data writing to the CL RC632's FIFO buffer.

Address line (MOSI)	MSB	bit 6 - bit 1	bit 0
byte 0	0	address	RFU (0)
byte 1 to byte n+1	data		

Note:

The data bus pins D7...D1 have to be disconnected.

For timing specification refer to chapter 22.5.2.4

Multiple Protocol Contactless Reader IC

CL RC632

5 CL RC632 REGISTER SET

5.1 CL RC632 Registers Overview

Page	Address _{hex}	Register Name	Function
Page 0: Command and Status	0	Page	selects the register page
	1	Command	starts (and stops) the command execution
	2	FIFOData	in- and output of 64 byte FIFO buffer
	3	PrimaryStatus	status flags of the receiver and transmitter and of the FIFO buffer
	4	FIFOLength	number of bytes buffered in the FIFO
	5	SecondaryStatus	diverse status flags
	6	InterruptEn	control bits to enable and disable passing of interrupt requests
	7	InterruptRq	interrupt request flags
Page 1: Control and Status	8	Page	selects the register page
	9	Control	diverse control flags e.g.: timer, power saving
	A	ErrorFlag	error flags showing the error status of the last command executed
	B	CollPos	bit position of the first bit collision detected on the RF-interface
	C	TimerValue	actual value of the timer
	D	CRCResultLSB	LSB of the CRC-Coprocessor register
	E	CRCResultMSB	MSB of the CRC-Coprocessor register
	F	BitFraming	adjustments for bit oriented frames
Page 2: Transmitter and Coder Control	10	Page	selects the register page
	11	TxControl	controls the logical behaviour of the antenna driver pins TX1 and TX2
	12	CWConductance	selects the conductance of the antenna driver pins TX1 and TX2
	13	ModConductance	Defines the driver output conductance
	14	CoderControl	sets the clock rate and the coding mode
	15	ModWidth	selects the width of the modulation pulse
	16	ModWidthSOF	selects the width of the modulation pulse for SOF (I•CODE Fast-Mode)
	17	TypeBFrameing	Defines the framing for ISO14443-B communication
Page 3: Receiver and Decoder Control	18	Page	selects the register page
	19	RxControl1	controls receiver behaviour
	1A	DecoderControl	controls decoder behaviour
	1B	BitPhase	selects the bit-phase between transmitter and receiver clock
	1C	RxThreshold	selects thresholds for the bit decoder
	1D	BPSKDemControl	Control BPSK receiver behaviour
	1E	RxControl2	controls decoder behaviour and defines the input source for the receiver
	1F	ClockQControl	controls clock generation for the 90° phase shifted Q-channel clock

CL RC632 Register Set (continued)

Multiple Protocol Contactless Reader IC

CL RC632

Page	Address _{hex}	Register Name	Function
Page 4: RF-Timing and Channel Redundancy	20	Page	selects the register page
	21	RxWait	selects the time interval after transmission, before receiver starts
	22	ChannelRedundancy	selects the kind and mode of checking the data integrity on the RF-channel
	23	CRCPresetLSB	LSB of the pre-set value for the CRC register
	24	CRCPresetMSB	MSB of the pre-set value for the CRC register
	25	TimeSlotPeriod	selects the time between automatically mitted Frames
	26	MFOUTSelect	selects internal signal applied to pin MFOUT, includes the MSB of value TimeSlotPeriod see register 0x25
	27	PreSet27	these values shall not be changed
Page 5: FIFO, Timer and IRQ-Pin Configuration	28	Page	selects the register page
	29	FIFOLevel	defines level for FIFO over- and underflow warning
	2A	TimerClock	selects the divider for the timer clock
	2B	TimerControl	selects start and stop conditions for the timer
	2C	TimerReload	defines the pre-set value for the timer
	2D	IRQPinConfig	configures the output stage of pin IRQ
	2E	PreSet2E	these values shall not be changed
	2F	PreSet2F	these values shall not be changed
Page 6: RFU	30	Page	selects the register page
	31	RFU	reserved for future use
	32	RFU	reserved for future use
	33	RFU	reserved for future use
	34	RFU	reserved for future use
	35	RFU	reserved for future use
	36	RFU	reserved for future use
	37	RFU	reserved for future use
Page 7: Test Control	38	Page	selects the register page
	39	RFU	reserved for future use
	3A	TestAnaSelect	selects analog test mode
	3B	RFU	reserved for future use
	3C	RFU	reserved for future use
	3D	TestDigiSelect	selects digital test mode
	3E	RFU	reserved for future use
	3F	RFU	reserved for future use

Table 5-1: CL RC632 Register Overview

Multiple Protocol Contactless Reader IC**CL RC632****5.1.1 REGISTER BIT BEHAVIOUR**

Bits and flags for different registers behave differently, depending on their functions. In principle bits with same behaviour are grouped in common registers.

Abbreviation	Behaviour	Description
r/w	read and write	These bits can be written and read by the µ-Processor. Since they are used only for control means, their content is not influenced by internal state machines, e.g. the <i>TimerReload-Register</i> may be written and read by the µ-Processor. It will also be read by internal state machines, but never changed by them.
dy	dynamic	These bits can be written and read by the µ-Processor. Nevertheless, they may also be written automatically by internal state machines, e.g. the <i>Command-Register</i> changes its value automatically after the execution of the actual command.
r	read only	These registers hold flags, which value is determined by internal states only, e.g. the <i>ErrorFlag-Register</i> can not be written from external but shows internal states.
w	write only	These registers are used for control means only. They may be written by the µ-Processor but can not be read. Reading these registers returns an undefined value, e.g. the <i>TestAnaSelect-Register</i> is used to determine the signal on pin AUX, but it is not possible to read its content.

Table 5-2: Behaviour of Register Bits and its Designation

Multiple Protocol Contactless Reader IC

CL RC632

5.2 Register Description

5.2.1 PAGE 0: COMMAND AND STATUS

5.2.1.1 Page Register

Selects the register page.

Name: Page Address: 0x00, 0x08, 0x10, 0x18, Reset value: 10000000, 0x80
0x20, 0x28, 0x30, 0x38

	7	6	5	4	3	2	1	0
Access Rights	r/w	r/w						
UsePage Select	0	0	0	0	0		PageSelect	

Description of the bits

Bit	Symbol	Function
7	UsePageSelect	If set to 1, the value of <i>PageSelect</i> is used as register address A5, A4, and A3. The LSB bits of the register address are defined by the address pins or the internal address latch, respectively. If set to 0, the whole content of the internal address latch defines the register address. The address pins are used as described in Table 4-2.
6-3	0000	Reserved for future use.
2-0	PageSelect	The value of <i>PageSelect</i> is used only if <i>UsePageSelect</i> is set to 1. In this case, it specifies the register page (which is A5, A4, and A3 of the register address).

Multiple Protocol Contactless Reader IC**CL RC632****5.2.1.2 Command Register**

Starts and stops the command execution.

Name: Command Address: 0x01 Reset value:X0000000, 0X0

	7	6	5	4	3	2	1	0
Access Rights	r	r	dy	dy	dy	dy	dy	dy
IFDetect Busy	0				Command			

Description of the bits

Bit	Symbol	Function
7	IFDetectBusy	Shows the status of Interface Detection Logic: Set to 0 means 'Interface Detection finished successfully', Set to 1 signs 'Interface Detection Ongoing'.
6	0	Reserved for future use.
5-0	Command	Activates a command according the Command Code. Reading this register shows, which command is actually executed.

Multiple Protocol Contactless Reader IC**CL RC632**

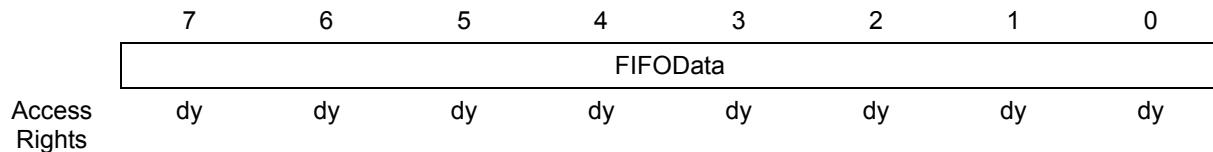
5.2.1.3 FIFOData Register

In- and output of the 64 byte FIFO buffer.

Name: FIFOData

Address: 0x02

Reset value: XXXXXXXX, 0xXX



Description of the bits

Bit	Symbol	Function
7-0	FIFOData	Data input and output port for the internal 64 byte FIFO buffer. The FIFO buffer acts as parallel in/parallel out converter for all data stream in- and outputs.

Multiple Protocol Contactless Reader IC

CL RC632

5.2.1.4 PrimaryStatus Register

Status flags of the receiver, transmitter and the FIFO buffer.

Name: PrimaryStatus Address: 0x03 Reset value: 00000101, 0x05

	7	6	5	4	3	2	1	0
Access Rights	0	ModemState			IRq	Err	HiAlert	LoAlert
r	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function		
7	0	Reserved for future use.		
6-4	ModemState	ModemState shows the state of the transmitter and receiver state machines.		
		State	Name of State	Description
		000	Idle	Neither the transmitter nor the receiver is in operation, since none of them is started or since none of them has input data.
		001	TxSOF	Transmitting the 'Start Of Frame' Pattern.
		010	TxData	Transmitting data from the FIFO buffer (or redundancy check bits).
		011	TxEOF	Transmitting the 'End Of Frame' Pattern.
		100	GoToRx1	Intermediate state, when receiver starts.
			GoToRx2	Intermediate state, when receiver finishes.
		101	PrepareRx	Waiting until the time period selected in the <i>RxWait Register</i> is expired.
		110	AwaitingRx	Receiver activated; Awaiting an input signal at pin Rx.
		111	Receiving	Receiving data.
3	IRQ	This bit shows, if any interrupt source requests attention (with respect to the setting of the interrupt enable flags in the <i>InterruptEn Register</i>).		
2	Err	This bit is set to 1, if any error flag in the <i>ErrorFlag Register</i> is set.		
1	HiAlert	Is set to 1, when the number of bytes stored in the FIFO buffer fulfil the following equation: $HiAlert = (64 - FIFOLength) \leq WaterLevel$		
		Example: FIFOLength=60, WaterLevel=4 $\Rightarrow HiAlert = 1$		
		FIFOLength=59, WaterLevel=4 $\Rightarrow HiAlert = 0$		
0	LoAlert	Is set to 1, when the number of bytes stored in the FIFO buffer fulfil the following equation: $LoAlert = FIFOLength \leq WaterLevel$		
		Example: FIFOLength=4, WaterLevel=4 $\Rightarrow LoAlert = 1$		
		FIFOLength=5, WaterLevel=4 $\Rightarrow LoAlert = 0$		

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.1.5 FIFOLength Register

Number of bytes buffered in the FIFO.

Name: FIFOLength

Address: 0x04

Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0	
	0	FIFOLength							
Access Rights	r	r	r	r	r	r	r	r	

Description of the bits

Bit	Symbol	Function
7	0	Reserved for future use.
6-0	FIFOLength	Indicates the number of bytes stored in the FIFO buffer. Writing to the <i>FIFOData</i> Register increments, reading decrements <i>FIFOlength</i> .

Multiple Protocol Contactless Reader IC

CL RC632**5.2.1.6 SecondaryStatus Register**

Diverse Status flags.

Name: SecondaryStatus

Address: 0x05

Reset value: 01100000, 0x60

	7	6	5	4	3	2	1	0
Access Rights	r	r	r	r	r	r	r	r
TRunning	E2Ready	CRCReady	0	0				RxLastBits

Description of the bits

Bit	Symbol	Function
7	TRunning	If set to 1, the CL RC632's timer unit is running, e.g. the counter will decrement the <i>Timer Value Register</i> with the next timer clock.
6	E2Ready	If set to 1, the CL RC632 has finished programming the E2PROM.
5	CRCReady	If set to 1, the CL RC632 has finished calculating the CRC.
4-3	00	Reserved for future use.
2-0	RxLastBits	Show the number of valid bits in the last received byte. If zero, the whole byte is valid.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.1.7 InterruptEn Register**

Control bits to enable and disable passing of interrupt requests.

Name: InterruptEn

Address: 0x06

Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
Access Rights	SetIEn	0	TimerIEn	TxIEn	RxIEn	IdleIEn	HiAlertIEn	LoAlertIEn
	w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	SetIEn	Set to 1 SetIEn defines that the marked bits in the <i>InterruptEn Register</i> are set, Set to 0 clears the marked bits.
6	0	Reserved for future use.
5	TimerIEn	Allows the timer interrupt request (indicated by bit <i>TimerIRq</i>) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit <i>SetIEn</i> .
4	TxIEn	Allows the transmitter interrupt request (indicated by bit <i>TxIRq</i>) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit <i>SetIEn</i> .
3	RxIEn	Allows the receiver interrupt request (indicated by bit <i>RxIRq</i>) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit <i>SetIEn</i> .
2	IdleIEn	Allows the idle interrupt request (indicated by bit <i>IdleIRq</i>) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit <i>SetIEn</i> .
1	HiAlertIEn	Allows the high alert interrupt request (indicated by bit <i>HiAlertIRq</i>) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit <i>SetIEn</i> .
0	LoAlertIEn	Allows the low alert interrupt request (indicated by bit <i>LoAlertIRq</i>) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit <i>SetIEn</i> .

Multiple Protocol Contactless Reader IC

CL RC632

5.2.1.8 *InterruptRq Register*

Interrupt request flags.

Name: InterruptRq Address: 0x07 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
Access Rights	SetIRq	0	TimerIRq	TxIRq	RxIRq	IdleIRq	HiAlertIRq	LoAlertIRq
w	r/w	dy	dy	dy	dy	dy	dy	dy

Description of the bits

Bit	Symbol	Function
7	SetIRq	Set to 1, <i>SetIRq</i> defines that the marked bits in the <i>InterruptRq Register</i> are set. Set to 0 <i>SetIRq</i> defines, that the marked bits in the <i>InterruptRq Register</i> are cleared.
6	0	Reserved for future use.
5	TimerIRq	Set to 1, when the timer decrements the <i>TimerValue Register</i> to zero.
4	TxIRq	Set to 1, when one of the following events occurs: <i>Transceive Command</i> : All data transmitted. <i>Auth1 and Auth2 Command</i> : All data transmitted. <i>WriteE2 Command</i> : All data is programmed. <i>CalCCRC Command</i> : All data is processed.
3	RxIRq	This bit is set to 1, when the receiver terminates.
2	IdleIRq	This bit is set to 1, when a command terminates by itself e.g. when the <i>Command Register</i> changes its value from any command to the <i>Idle Command</i> . If an unknown command is started bit <i>IdleIRq</i> is set. Starting the <i>Idle Command</i> by the µ-Processor does not set bit <i>IdleIRq</i> .
1	HiAlertIRq	This bit is set to 1, when bit <i>HiAlert</i> is set. In opposite to <i>HiAlert</i> , <i>HiAlertIRq</i> stores this event and can only be reset by means of bit <i>SetIRq</i> .
0	LoAlertIRq	This bit is set to 1, when bit <i>LoAlert</i> is set. In opposite to <i>LoAlert</i> , <i>LoAlertIRq</i> stores this event and can only be reset by means of bit <i>SetIRq</i> .

Multiple Protocol Contactless Reader IC**CL RC632****5.2.2 PAGE 1: CONTROL AND STATUS****5.2.2.1 Page Register**

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.2.2 Control Register

Diverse control flags, e.g.: timer, power saving.

Name: Control Address: 0x09 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
Access Rights	0	0	StandBy	PowerDown	Crypto1On	TStopNow	TStartNow	FlushFIFO
r/w	r/w	dy	dy	dy	w	w	w	w

Description of the bits

Bit	Symbol	Function
7-6	00	Reserved for future use
5	StandBy	Setting this bit to 1 enters the Soft PowerDown Mode. This means, internal current consuming blocks are switched off, the oscillator keeps running.
4	PowerDown	Setting this bit to 1 enters the Soft PowerDown Mode. This means, internal current consuming blocks are switched off including the oscillator.
3	Crypto1On	This bit indicates that the Crypto1 unit is switched on and therefore all data communication with the card is encrypted. This bit can only be set to 1 by a successful execution of the <i>Authent2 Command</i> .
2	TStopNow	Setting this bit to 1 stops the timer immediately. Reading this bit will always return 0.
1	TStartNow	Setting this bit to 1 starts the timer immediately. Reading this bit will always return 0.
0	FlushFIFO	Setting this bit to 1 clears the internal FIFO-buffer's read- and write-pointer (<i>FIFOLength</i> becomes 0) and the flag <i>FIFOovfl</i> immediately. Reading this bit will always return 0.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.2.3 ErrorFlag Register**

Error flags showing the error status of the last executed command.

Name: ErrorFlag

Address: 0x0A

Reset value: 01000000, 0x40

	7	6	5	4	3	2	1	0
Access Rights	0	KeyErr	AccessErr	FIFOovfl	CRCErr	FramingErr	ParityErr	CollErr
r	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7	0	Reserved for future use.
6	KeyErr	This bit is set to 1, if the LoadKeyE2 or the LoadKey Command recognises, that the input data is not coded according to the Key format definition. This bit is set to 0 starting the LoadkeyE2 or the LoadKey command.
5	AccessErr	This bit is set to 1, if the access rights to the E ² PROM are violated. This bit is set to 0 starting an E ² PROM related command.
4	FIFOovfl	This bit is set to 1, if the µ-Processor or a CL RC632's internal state machine (e.g. receiver) tries to write data into the FIFO buffer although the FIFO buffer is already full.
3	CRCErr	This bit is set to 1, if RxCRCEn is set and the CRC fails. It is cleared to 0 automatically at receiver start phase during the state PrepareRx.
2	FramingErr	This bit is set to 1, if the SOF is incorrect. It is cleared automatically at receiver start (that is during the state PrepareRx).
1	ParityErr	This bit is set to 1, if the parity check has failed. It is cleared automatically at receiver start (that is during the state PrepareRx).
0	CollErr	This bit is set to 1, if a bit-collision is detected. It is cleared automatically at receiver start (that is during the state PrepareRx). Note: only valid for communication according to ISO14443 A.

Multiple Protocol Contactless Reader IC

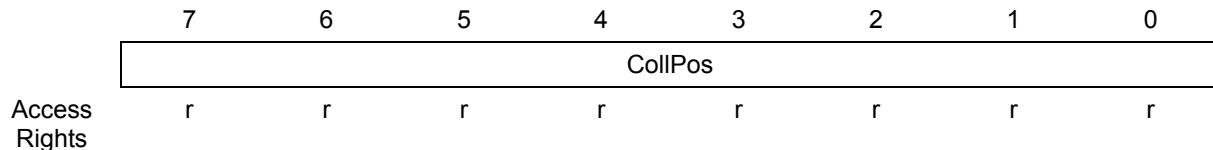
CL RC632**5.2.2.4 CollPos Register**

Bit position of the first bit collision detected on the RF- interface.

Name: CollPos

Address: 0x0B

Reset value: 00000000, 0x00



Description of the bits

Bit	Symbol	Function
7-0	CollPos	<p>This register shows the bit position of the first detected collision in a received frame.</p> <p>Example:</p> <ul style="list-style-type: none"> 0x00 indicates a bit collision in the start bit 0x01 indicates a bit collision in the 1st bit 0x08 indicates a bit collision in the 8th bit

Note: For ISO14443B a bit collision is not indicated in the CollPos register.

Multiple Protocol Contactless Reader IC**CL RC632**

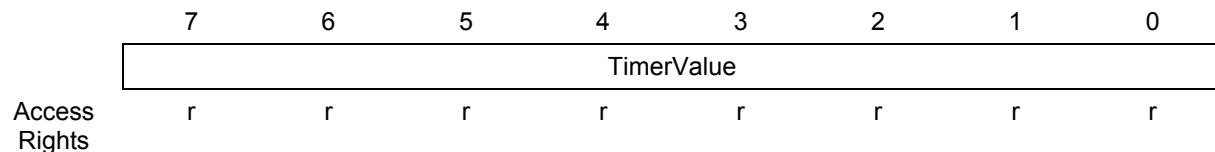
5.2.2.5 TimerValue Register

actual value of the timer.

Name: TimerValue

Address:0x0C

Reset value: XXXXXXXX, 0xXX



Description of the bits

Bit	Symbol	Function
7-0	TimerValue	This register shows the actual value of the timer counter.

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.2.6 CRCResultLSB Register

LSB of the CRC-Coprocessor register.

Name: CRCResultLSB

Address: 0x0D

Reset value: XXXXXXXX, 0xXX

	7	6	5	4	3	2	1	0
CRCResultLSB								
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7-0	CRCResultLSB	This register shows the actual value of the least significant byte of the CRC register. It is valid only if bit <i>CRCReady</i> is set to 1.

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.2.7 CRCResultMSB Register

MSB of the CRC-Coprocessor register.

Name: CRCResultMSB

Address: 0x0E

Reset value: XXXXXXXX, 0xXX

	7	6	5	4	3	2	1	0
CRCResultMSB								
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7-0	CRCResultMSB	This register shows the actual value of the most significant byte of the CRC register. It is valid only if bit <i>CRCReady</i> is set to 1. For 8-bit CRC calculation the registers value is undefined.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.2.8 BitFraming Register**

Adjustments for bit oriented frames.

Name: BitFraming Address: 0x0F Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
Access Rights	0	RxAlign			0	TxLastBits		
r/w	dy	dy	dy	dy	r/w	dy	dy	dy

Description of the bits

Bit	Symbol	Function
7	0	Reserved for future use
6-4	RxAlign	<p>Used for reception of bit oriented frames: <i>RxAlign</i> defines the bit position for the first bit received to be stored in the FIFO. Further received bits are stored in the following bit positions. After reception, <i>RxAlign</i> is cleared automatically.</p> <p>Example:</p> <ul style="list-style-type: none"> <i>RxAlign</i> = 0: the LSB of the received bit is stored at bit 0, second received bit is stored at bit position 1 <i>RxAlign</i> = 1: the LSB of the received bit is stored at bit 1, second received bit is stored at bit position 2 <i>RxAlign</i> = 7: the LSB of the received bit is stored at bit 7, second received bit is stored in the following byte at bit position 0
3	0	reserved for future use
2-0	TxLastBits	<p>Used for transmission of bit oriented frames: <i>TxLastBits</i> defines the number of bits of the last byte that shall be transmitted. A 000 indicates that all bits of the last byte shall be transmitted. After transmission, <i>TxLastBits</i> is cleared automatically.</p>

Multiple Protocol Contactless Reader IC

CL RC632

5.2.3 PAGE 2: TRANSMITTER AND CONTROL

5.2.3.1 *Page Register*

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.3.2 *TxControl Register*

Controls the logical behaviour of the antenna pin TX1 and TX2.

Name: TxControl Address: 0x11 Reset value: 01011000, 0x58

	7	6	5	4	3	2	1	0
Access Rights	0	ModulatorSource	Force 100ASK	TX2Inv	TX2Cw	TX2RFE _n	TX1RFE _n	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	0	This value shall not be changed
6-5	Modulator Source	Selects the source for the modulator input: 00: LOW 01: HIGH 10: Internal Coder 11: Pin MFIN
4	Force100ASK	Set to 1, <i>Force100ASK</i> forces a 100% ASK Modulation independent of the setting in the ModConductance Register.
3	TX2Inv	Set to 1, the output signal on pin TX2 will deliver an inverted 13.56 MHz energy carrier.
2	TX2Cw	Set to 1, the output signal on pin TX2 will deliver continuously the un-modulated 13.56 MHz energy carrier. Setting <i>TX2Cw</i> to 0 enables modulation of the 13.56 MHz energy carrier.
1	TX2RFE _n	Set to 1, the output signal on pin TX2 will deliver the 13.56 MHz energy carrier modulated by the transmission data. If <i>TX2RFE_n</i> is 0, TX2 drives a constant output level.
0	TX1RFE _n	Set to 1, the output signal on pin TX1 will deliver the 13.56 MHz energy carrier modulated by the transmission data. If <i>TX1RFE_n</i> is 0, TX1 drives a constant output level.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.3.3 CwConductance Register**

Selects the conductance of the antenna driver pins TX1 and TX2.

Name: CwConductance

Address: 0x12

Reset value: 00111111, 0x3F

	7	6	5	4	3	2	1	0
	0	0			GsCfgCW			
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	These values shall not be changed
5-0	GsCfgCW	The value of this register defines the conductance of the output driver. This may be used to regulate the output power and subsequently current consumption and operating distance.

Note: For detailed information about GsCfgCW see 13.3

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.3.4 ModConductance Register

defines the driver output conductance.

Name: ModConductance Address: 0x13 Reset value: 00111111, 0x03F

	7	6	5	4	3	2	1	0
	0	0				GsCfgMod		
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	These values shall not be changed
5-0	GsCfgMod	The value of this register defines the conductance of the output driver for the time of modulation. This may be used to regulate the modulation index.

Note: If Force100ASK is set to one, the value of GsCfgMod has no effect.

For detailed information about GsCfgMod see 13.3

Multiple Protocol Contactless Reader IC**CL RC632****5.2.3.5 CoderControl Register**

sets the clock rate and the coding mode

Name: CoderControl Address:0x14 Reset value: 00011001, 0x19

	7	6	5	4	3	2	1	0
Access Rights	SendOnePulse	0	CoderRate		TxCoding			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	SendOnePulse	Set to 1, forces to generate only one Modulation (for ISO 15693 only). This is used to switch to the next TimeSlot if the Inventory command is used. This bit is not cleared automatically, it has to be re-set to 0 by the user.
6	0	These values shall not be changed
5-3	CoderRate	<p>This register defines the clock rate for Coder Circuit</p> <p>000: MIFARE® 848 kBaud 001: MIFARE® 424 kBaud 010: MIFARE® 212 kBaud 011: MIFARE® 106 kBaud; ISO14443 A 100: ISO 14443-B 101: For I•CODE1 standard mode and ISO 15693 (~52.97kHz) 110: For I•CODE1 fast mode (~26.48kHz) 111: RFU</p>
2-0	TxCoding	<p>This register defines the bit coding Mode and Framing during Transmission</p> <p>000: NRZ according ISO14443-B 001: MIFARE®, ISO14443-A, (Miller coded) 010: RFU 011: RFU 100: For I•CODE1 standard mode (1 out of 256 coding) 101: For I•CODE1 fast mode (RZ coding) 110: For ISO 15693 standard mode (1 out of 256 coding) 111: For ISO 15693 fast mode (1 out of 4 coding)</p>

Multiple Protocol Contactless Reader IC

CL RC632**5.2.3.6 ModWidth Register**

selects the width of the modulation pulse.

Name: ModWidth Address: 0x15 Reset value: 00010011, 0x13

	7	6	5	4	3	2	1	0
Access Rights	r/w							
ModWidth								

Description of the bits

Bit	Symbol	Function
7-0	ModWidth	This register defines the width of the modulation pulse according to $T_{mod} = 2 \cdot (\text{ModWidth} + 1) / f_c$.

5.2.3.7 ModWidthSOF Register

Name: ModWidthSOF Address: 0x16 Reset value: 00111111, 0x3F

	7	6	5	4	3	2	1	0
Access Rights	r/w							
ModWidthSOF								

Description of the bits

Bit	Symbol	Function
7-0	ModWidthSOF	<p>This register defines the width of the modulation pulse for SOF $T_{mod} = 2 \cdot (\text{ModWidth} + 1) / f_c$.</p> <p>Register setting:</p> <ul style="list-style-type: none"> MIFARE® & ISO14443: 0x3F (Modulation width SOF: 9.44μs). I•CODE1 Standard Mode: 0x3F (Modulation width SOF: 9.44μs). I•CODE1 Fast Mode: 0x73 (Modulation width SOF: 18.88μs). ISO 15693: 0x3F (Modulation width SOF: 9.44μs).

Multiple Protocol Contactless Reader IC**CL RC632****5.2.3.8 TypeBFRaming**

defines the framing for ISO 14443 B communication

Name: TypeBFRaming Address: 0x17 Reset value: 00111011, 0x3B

	7	6	5	4	3	2	1	0
Access Rights	NoTx SOF	NoTx EOF	EOF Width	CharSpacing			SOFWidth	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	NoTxSOF	Set to 1 TxCoder suppresses the SOF
6	NoTXEOF	Set to 1 TxCoder suppresses the EOF
5	EOFWidth	0: Set the EOF to a length of 10 ETU 1: Set the EOF to a length of 11 ETU
4-2	CharSpacing	Set the length of the EGT length between 0 and 7 ETU.
1-0	SOFWidth	00: Set the SOF to a length of 10 ETU LOW and 2 ETU HIGH 01: Set the SOF to a length of 10 ETU LOW and 3 ETU HIGH 10: Set the SOF to a length of 11 ETU LOW and 2 ETU HIGH 11: Set the SOF to a length of 11 ETU LOW and 3 ETU HIGH

Multiple Protocol Contactless Reader IC

CL RC632

5.2.4 PAGE 3: RECEIVER AND DECODER CONTROL

5.2.4.1 *Page Register*

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.4.2 *RxControl1 Register*

controls receiver behaviour.

Name: RxControl1 Address: 0x19 Reset value: 01110011, 0x73

	7	6	5	4	3	2	1	0
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
	SubCPulses			ISO Selection		LPOff		Gain

Description of the bits

Bit	Symbol	Function
7-5	SubCPulses	Defines the number of subcarrier pulses per Bit 000: 1 Pulse 001: 2 Pulses 010: 4 Pulses 011: 8 Pulses ISO14443A&B 100: 16 Pulses I•CODE1 , ISO15693 101: RFU 110: RFU 111: RFU
4-3	ISO Selection	00: RFU 10: ISO 14443 A&B 01: I•CODE1 , ISO15693 11: RFU
2	LPOff	Switches off a LowPassFilter at the internal amplifier.
1-0	Gain	This register defines the receivers signal voltage gain factor: 00: 20 dB 01: 24 dB 10: 31 dB 11: 35 dB

Multiple Protocol Contactless Reader IC**CL RC632****5.2.4.3 DecoderControl Register**

Controls decoder behaviour.

Name: DecoderControl Address: 0x1A Reset value: 00001000, 0x08

	7	6	5	4	3	2	1	0
	0	RxMultiple	ZeroAfter Coll	RxFraming	RxInvert	0	RxCoding	
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	0	These values shall not be changed
6	RxMultiple	If set to 0, after receiving of the Frame the receiver is deactivated If set to 1, it is possible to receive more than one Frame
5	ZeroAfter Coll	If set to 1, any bits received after a bit-collision are masked to zero. This eases resolving the anti-collision procedure defined in ISO14443-A.
4-3	RxFraming	00: for I•CODE1 01: MIFARE®, ISO14443A 10: ISO 15693 11: ISO14443B
2	RxInvert	If set to 0, a modulation at the first half bit results a logic 1 (according I•CODE1) If set to 1, a modulation at the first half bit results a logic 0 (according ISO15693)
1	0	These values shall not be changed
0	RxCoding	0: Manchester Coding 1: BPSK Coding

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.4.4 BitPhase Register

selects the bit-phase between transmitter and receiver clock.

Name: BitPhase

Address: 0x1B

Reset value: 10101101, 0xAD

	7	6	5	4	3	2	1	0
BitPhase								
Access Rights	r/w							

Description of the bits

Bit	Symbol	Function
7-0	BitPhase	Defines the phase relation between transmitter and receiver clock. <u>Note:</u> The correct value of this register is essential for proper operation.

Multiple Protocol Contactless Reader IC**CL RC632****5.2.4.5 RxThreshold Register**

selects thresholds for the bit decoder.

Name: RxThreshold

Address: 0x1C

Reset value: 11111111, 0xFF

	7	6	5	4	3	2	1	0
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
MinLevel					CollLevel			

Description of the bits

Bit	Symbol	Function
7-4	MinLevel	Defines the minimum signal strength at the decoder input that shall be accepted. If the signal strength is below this level, it is not evaluated.
3-0	CollLevel	Defines the minimum signal strength at the decoder input that has to be reached by the weaker half-bit of the Manchester-coded signal to generate a bit-collision relatively to the amplitude of the stronger half-bit.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.4.6 BPSKDemControl**

controls BPSK demodulation

Name: BPSKDemControl Address: 0x1D Reset value: 00011110, 0x1E

	7	6	5	4	3	2	1	0
Access Rights	NoRx SOF r/w	NoRx EGT r/w	NoRx EOF r/w	Filter AmpDet r/w	TauD r/w		TauB r/w	

Description of the bits

Bit	Symbol	Function
7	NoRxSOF	If set to 1 a missing SOF in the receiving data stream will be ignored and no framing error indicated
6	NoRxEGT	If set to 1 a too short or too long EGT in the receiving data stream will be ignored and no framing error indicated
5	NoRxEOF	If set to 1 a missing EOF in the receiving data stream produces will be ignored and no framing error indicated
4	FilterAmpDet	Switches on a HighPassFilter for amplitude detection
3-2	TauD	Change time-constant of internal PLL during data receiving
1-0	TauB	Change time-constant of internal PLL during burst

Multiple Protocol Contactless Reader IC**CL RC632****5.2.4.7 RxControl2 Register**

controls decoder behaviour and defines the input source for the receiver.

Name:RxControl2

Address: 0x1E

Reset value: 01000001, 0x41

	7	6	5	4	3	2	1	0
Access Rights	RcvClkSell	RxAutoPD	0	0	0	0	DecoderSource	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	RcvClkSell	If set to 1, the I-clock is used for the receiver clock. Set to 0 indicates, that the Q-clock is used. I-clock and Q-clock are 90° phase shifted to each other
6	RxAutoPD	If set to 1, the receiver circuit is automatically switched on before receiving and switched off afterwards. This may be used to reduce current consumption. If set to 0, the receiver is always activated.
5-2	0000	These values shall not be changed
1-0	DecoderSource	Selects the source for the decoder input: 00: Low 01: Internal Demodulator 10: A subcarrier modulated Manchester coded signal at Pin MFIN 11: A baseband Manchester coded signal at Pin MFIN

Multiple Protocol Contactless Reader IC

CL RC632**5.2.4.8 ClockQControl Register**

controls clock generation for the 90° phase shifted Q-channel clock.

Name: ClockQControl

Address: 0x1F

Reset value: 000XXXXX, 0xXX

	7	6	5	4	3	2	1	0
Access Rights	r	r/w	r/w	dy	dy	dy	dy	dy
	ClkQ180Deg	ClkQCalib	0		ClkQDelay			

Description of the bits

Bit	Symbol	Function
7	ClkQ180Deg	If the Q-clock is phase shifted more than 180° compared to the I-clock, the bit <i>ClkQ180Deg</i> is set to 1, otherwise it is 0.
6	ClkQCalib	If this bit is 0, the Q-clock is calibrated automatically after the Reset Phase and after data reception from the card. If this bit is set to 1, no calibration is performed automatically.
5	0	This value shall not be changed
4-0	ClkQDelay	This register shows the number of delay elements actually used to generate a 90°phase shift of the I-clock to obtain the Q-clock. It can be written directly by the µ-Processor or by the automatic calibration cycle.

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.5 PAGE 4: RF-TIMING AND CHANNEL REDUNDANCY**5.2.5.1 *Page Register***

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.5.2 *RxWait Register*

Selects the time interval after transmission, before receiver starts.

Name: RxWait Address: 0x21 Reset value: 00000101, 0x06

	7	6	5	4	3	2	1	0
RxWait								
Access Rights	r/w							

Description of the bits

Bit	Symbol	Function
7-0	RxWait	After data transmission, the activation of the receiver is delayed for RxWait bitclocks. During this 'frame guard time' any signal at pin Rx is ignored.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.5.3 ChannelRedundancy Register**

Selects kind and mode of checking the data integrity on the RF-channel.

Name: ChannelRedundancy Address: 0x22 Reset value: 00000011, 0x03

	7	6	5	4	3	2	1	0
Access Rights	0	0	CRC 3309	CRC8	RxCRCEn	TxCRCEn	ParityOdd	ParityEn
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	This value shall not be changed
5	CRC3309	If set to 1, CRC-calculation is done according ISO/IEC3309 (ISO14443B) and ISO 15693. <u>Note:</u> For usage according to ISO14443A this bit has to be 0. For usage according to I•CODE1 this bit has to be 0.
4	CRC8	If set to 1, an 8-bit CRC is calculated. If set to 0, a 16-bit CRC is calculated.
3	RxCRCEn	If set to 1, the last byte(s) of a received frame is/are interpreted as CRC byte/s. If the CRC itself is correct the CRC byte(s) is/are not passed to the FIFO. In case of an error, the <i>CRCErr</i> flag is set. If set to 0, no CRC is expected.
2	TxCRCEn	If set to 1, a CRC is calculated over the transmitted data and the CRC byte(s) are appended to the data stream. If set to 0, no CRC is transmitted.
1	ParityOdd	If set to 1, an odd parity is generated or expected, respectively. If set to 0 an even parity is generated or expected, respectively. <u>Note:</u> For usage according to ISO14443-A this bit has to be 1.
0	ParityEn	If set to 1, a parity bit is inserted in the transmitted data stream after each byte and expected in the received data stream after each byte (MIFARE®, ISO14443A) If set to 0, no parity bit is inserted or expected (ISO14443B)

Multiple Protocol Contactless Reader IC**CL RC632****5.2.5.4 CRCPresetLSB Register**

LSB of the preset value for the CRC register.

Name: CRCPresetLSB Address: 0x23 Reset value: 01010011, 0x63

	7	6	5	4	3	2	1	0
CRCPresetLSB								
Access Rights	r/w							

Description of the bits

Bit	Symbol	Function
7-0	CRCPresetLSB	<p><i>CRCPresetLSB</i> defines the starting value for CRC-calculation. This value is loaded into the CRC at the beginning of transmission, reception and the CalcCRC Command, if the CRC calculation is enabled.</p> <p>To use the ISO 15693 functionality the <i>CRCPresetLSB</i> Register has to be set to 0xFF.</p>

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.5.5 CRCPresetMSB Register

MSB of the preset value for the CRC register.

Name: CRCPresetMSB Address: 0x24 Reset value: 01010011, 0x63

	7	6	5	4	3	2	1	0
CRCPresetMSB								
Access Rights	r/w							

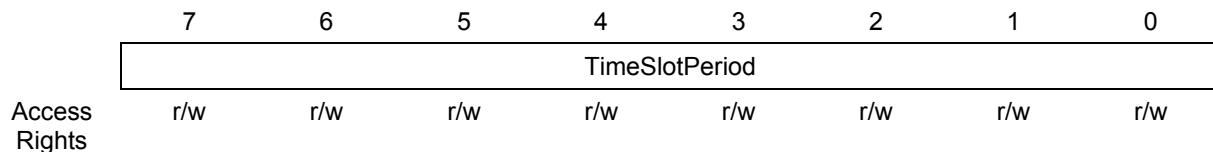
Description of the bits

Bit	Symbol	Function
7-0	CRCPresetMSB	<p><i>CRCPresetMSB</i> defines the starting value for CRC-calculation. This value is loaded into the CRC at the beginning of transmission, reception and the CalcCRC Command, if the CRC calculation is enabled.</p> <p><u>Note:</u> This register is not relevant, if CRC8 is 1.</p>

Multiple Protocol Contactless Reader IC**CL RC632****5.2.5.6 TimeSlotPeriod Register**

defines the time slot period for I•CODE1 protocol.

Name: TimeSlotPeriod Address: 0x25 Reset value: 00000000, 0x00



Description of the bits

Bit	Symbol	Function
7-0	TimeSlotPeriod	<i>TimeSlotPeriod</i> defines the time between automatically transmitted Frames. To send a Quit-Frame according to the I•CODE1 protocol, it is necessary to have a relation to the beginning of the Command-Frame. The TimeSlotPeriod will start at the End of the Command transmission. For detailed information see also chapter 9.2.5

Multiple Protocol Contactless Reader IC

CL RC632**5.2.5.7 MFOUTSelect Register**

Selects internal signal applied to pin MFOUT.

Name: MFOUTSelect

Address: 0x26

Reset value:00000000, 0x00

	7	6	5	4	3	2	1	0
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
	0	0	0	TimeSlot Period MSB	0		MFOUTSelect	

Description of the bits

Bit	Symbol	Function																				
7-5	00000	These values shall not be changed																				
4	TimeSlotPeriod MSB	MSB of value TimeSlotPeriod see register 0x25																				
3	0	These values shall not be changed																				
2-0	MFOUTSelect	<p><i>MFOUTSelect</i> defines which signal is routed to pin MFOUT.</p> <table> <tr> <td>000</td> <td>Constant Low</td> </tr> <tr> <td>001</td> <td>Constant High</td> </tr> <tr> <td>010</td> <td>Modulation Signal (envelope) from internal coder, Miller coded</td> </tr> <tr> <td>011</td> <td>Serial data stream, not Miller coded</td> </tr> <tr> <td>100</td> <td>Output signal of the energy carrier demodulator (card modulation signal)</td> </tr> <tr> <td></td> <td>Note: only valid MIFARE® and ISO14443 A at a baudrate of 106 kbaud.</td> </tr> <tr> <td>101</td> <td>Output signal of the subcarrier demodulator (Manchester coded card signal)</td> </tr> <tr> <td></td> <td>Note: only valid MIFARE® and ISO14443 A at a baudrate of 106 kbaud.</td> </tr> <tr> <td>110</td> <td>RFU</td> </tr> <tr> <td>111</td> <td>RFU</td> </tr> </table>	000	Constant Low	001	Constant High	010	Modulation Signal (envelope) from internal coder, Miller coded	011	Serial data stream, not Miller coded	100	Output signal of the energy carrier demodulator (card modulation signal)		Note: only valid MIFARE® and ISO14443 A at a baudrate of 106 kbaud.	101	Output signal of the subcarrier demodulator (Manchester coded card signal)		Note: only valid MIFARE® and ISO14443 A at a baudrate of 106 kbaud.	110	RFU	111	RFU
000	Constant Low																					
001	Constant High																					
010	Modulation Signal (envelope) from internal coder, Miller coded																					
011	Serial data stream, not Miller coded																					
100	Output signal of the energy carrier demodulator (card modulation signal)																					
	Note: only valid MIFARE® and ISO14443 A at a baudrate of 106 kbaud.																					
101	Output signal of the subcarrier demodulator (Manchester coded card signal)																					
	Note: only valid MIFARE® and ISO14443 A at a baudrate of 106 kbaud.																					
110	RFU																					
111	RFU																					

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.5.8 PreSet27 Register

Name: PreSet27

Address: 0x27

Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Multiple Protocol Contactless Reader IC

CL RC632

5.2.6 PAGE 5: FIFO, TIMER AND IRQ- PIN CONFIGURATION

5.2.6.1 *Page Register*

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.6.2 *FIFOLevel Register*

Defines the level for FIFO under- and overflow warning.

Name: FIFOLevel Address: 0x29 Reset value:00001000, 0x08

	7	6	5	4	3	2	1	0
Access Rights	0	0			WaterLevel			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	These values shall not be changed
5-0	WaterLevel	<p>This register defines, the warning level of the CL RC632 for the µ-Processor for a FIFO-buffer over- or underflow:</p> <p><i>HiAlert</i> is set to 1, if the remaining FIFO-buffer space is equal or less than <i>WaterLevel</i> bytes in the FIFO-buffer.</p> <p><i>LoAlert</i> is set to 1, if equal or less than <i>WaterLevel</i> bytes are in the FIFO-buffer.</p>

Multiple Protocol Contactless Reader IC

CL RC632**5.2.6.3 TimerClock Register**

Selects the divider for the timer clock.

Name: TimerClock Address: 0x2A Reset value: 00000111, 0x07

	7	6	5	4	3	2	1	0
Access Rights	0	0	TAutoRestart			TPreScaler		
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	These values shall not be changed
5	TAutoRestart	If set to 1, the timer automatically restart its count-down from <i>TReloadValue</i> , instead of counting down to zero. If set to 0 the timer decrements to zero and the bit <i>TimerRq</i> is set to 1.
4-0	TPreScaler	Defines the timer clock f_{Timer} . <i>TPreScaler</i> can be adjusted from 0 up to 21. The following formula is used to calculate f_{Timer} : $f_{\text{Timer}} = 13.56 \text{ MHz} / 2^{T\text{PreScaler}}$.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.6.4 TimerControl Register**

Selects start and stop conditions for the timer.

Name: TimerControl

Address: 0x2B

Reset value: 00000110, 0x06

	7	6	5	4	3	2	1	0
Access Rights	0	0	0	0	TStopRxEnd	TStopRxBegin	TStartTxEnd	TStartTxBegin
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-4	0000	These values shall not be changed
3	TStopRxEnd	If set to 1, the timer stops automatically when data reception ends. 0 indicates, that the timer is not influenced by this condition.
2	TStopRxBegin	If set to 1, the timer stops automatically, when the first valid bit is received. 0 indicates, that the timer is not influenced by this condition.
1	TStartTxEnd	If set to 1, the timer starts automatically when data transmission ends. If the timer is already running, the timer restarts by loading <i>TReloadValue</i> into the timer. 0 indicates, that the timer is not influenced by this condition.
0	TStartTxBegin	If set to 1, the timer is starts automatically when the first bit is transmitted. If the timer is already running, the timer restarts by loading <i>TReloadValue</i> into the timer. 0 indicates, that the timer is not influenced by this condition.

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.6.5 TimerReload Register

Defines the preset value for the timer.

Name: TimerReload Address: 0x2C Reset value: 00001010, 0x0A

	7	6	5	4	3	2	1	0
TReloadValue								
Access Rights	r/w							

Description of the bits

Bit	Symbol	Function
7-0	TreloadValue	With a start event the timer loads with the <i>TreloadValue</i> . Changing this register affects the timer only with the next start event. If <i>TReloadValue</i> is set to 0, the timer cannot start.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.6.6 IRQPinConfig Register**

Configures the output stage for pin IRQ.

Name: IRQPinConfig

Address: 0x2D

Reset value: 00000010, 0x02

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	IRQInv	IRQPushPull
Access Rights	r/w	r/w						

Description of the bits

Bit	Symbol	Function
7-2	000000	These values shall not be changed
1	IRQInv	If set to 1, the signal on pin IRQ is inverted with respect to bit /IRq. 0 indicates, that the signal on pin IRQ is equal to bit /IRQ.
0	IRQPushPull	If set to 1, pin IRQ works as standard CMOS output pad. 0 indicates, that pin IRQ works as open drain output pad.

5.2.6.7 PreSet2E

Name: PreSet2E

Address: 0x2E

Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

5.2.6.8 Preset2F

Name: PreSet2F

Address: 0x2F

Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Multiple Protocol Contactless Reader IC

CL RC632

5.2.7 PAGE 6: RFU

5.2.7.1 Page Register

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.7.2 RFU Registers

Name: RFU Address: 0x31, 0x32, 0x33, 0x34, Reset value:xxxxxxxx, 0xxx
0x35, 0x36, 037

Note: These registers are reserved for future use.

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.8 PAGE 7: TEST CONTROL**5.2.8.1 *Page Register***

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.8.2 *RFU Register*

Name: RFU Address: 0x39 Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Note: This register is reserved for future use.

Multiple Protocol Contactless Reader IC

CL RC632**5.2.8.3 TestAnaSelect Register**

Selects analog test signals.

Name: TestAnaSelect

Address: 0x3A

Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	0	0	0	0				
Access Rights	W	W	W	W	W	W	W	W

Description of the bits

Bit	Symbol	Function	
7-4	0000	These values shall not be changed	
3-0	TestAnaOutSel	This register selects the internal analog signal that is routed to pin AUX. For detailed information see 21.3	
Value	Signal Name		
0	V_{mid}		
1	$V_{bandgap}$		
2	$V_{RxFollI}$		
3	$V_{RxFollQ}$		
4	V_{RxAmpl}		
5	V_{RxAmpQ}		
6	V_{CorrNI}		
7	V_{CorrNQ}		
8	V_{CorrDI}		
9	V_{CorrDQ}		
A	V_{EvalL}		
B	V_{EvalR}		
C	V_{Temp}		
D	RFU		
E	RFU		
F	RFU		

Multiple Protocol Contactless Reader IC**CL RC632**

5.2.8.4 RFU Register

Name: RFU

Address: 0x3B

Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
Access Rights	x	x	x	x	x	x	x	x

Note: This register is reserved for future use.**5.2.8.5 RFU Register**

Name: RFU

Address: 0x3C

Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
Access Rights	x	x	x	x	x	x	x	x

Note: This register is reserved for future use.

Multiple Protocol Contactless Reader IC

CL RC632

5.2.8.6 *TestDigiSelect Register*

Selects digital test mode.

Name: TestDigiSelect

Address:0x3D

Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
Access Rights	W	W	W	W	W	W	W	W
SignalToMFOUT	TestDigiSignalSel							

Description of the bits

Bit	Symbol	Function	
7	SignalToMFOUT	Set to 1, overrules the setting in <i>MFOUTSelect</i> and the digital test signal defined in <i>TestDigiSignalSel</i> is routed to pin MFOUT instead. Set to 0, <i>MFOUTSelect</i> defines the signal delivered at pin MFOUT.	
6-0	TestDigiSignalSel	Selects the digital test signal to be routed to pin MFOUT. For detailed information refer to chapter 21.4	
		TestDigiSignalSel	Signal Name
		F4 _{hex}	s_data
		E4 _{hex}	s_valid
		D4 _{hex}	s_coll
		C4 _{hex}	s_clock
		B5 _{hex}	rd_sync
		A5 _{hex}	wr_sync
		96 _{hex}	int_clock
		83 _{hex}	BPSK_out
		E2 _{hex}	BPSK_sig

5.2.8.7 *RFU Registers*

Name: RFU

Address: 0x3E, 0x3F

Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
Access Rights	X	X	X	X	X	X	X	X

Note: These registers are reserved for future use.

Multiple Protocol Contactless Reader IC

CL RC632**5.3 CL RC632 Register Flags Overview**

Flag(s)	Register	Address Register, Bit Position
AccessErr	ErrorFlag	0x0A, bit 5
BitPhase	BitPhase	0x1B, bits 7:0
CharSpacing	TypeBFRaming	0x17, bits 4:2
ClkQ180Deg	ClockQControl	0x1F, bit 7
ClkQCalib	ClockQControl	0x1F, bit 6
ClkQDelay	ClockQControl	0x1F, bits 4:0
CoderRate	CoderControl	0x14, bits 5:3
CollErr	ErrorFlag	0x0A, bit 0
CollLevel	RxThreshold	0x1C, bits 3:0
CollPos	CollPos	0x0B, bits 7:0
Command	Command	0x01, bits 5:0
CRC3309	ChannelRedundancy	0x22, bit 5
CRC8	ChannelRedundancy	0x22, bit 4
CRCErr	ErrorFlag	0x0A, bit 3
CRCPresetLSB	CRCPresetLSB	0x23, bits 7:0
CRCPresetMSB	CRCPresetMSB	0x24, bits 7:0
CRCReady	SecondaryStatus	0x05 , bit 5
CRCResultMSB	CRCResultMSB	0x0E, bits 7:0
CRCResultLSB	CRCResultLSB	0x0D, bits 7:0
Crypto1On	Control	0x09, bit 3
DecoderSource	RxControl2	0x1E, bits 1:0
E2Ready	SecondaryStatus	0x05, bit 6
EOFWidth	TypeBFRaming	0x17, bit 5
Err	PrimaryStatus	0x03, bit 2
FIFOData	FIFOData	0x02, bits 7:0
FIFOLength	FIFOLength	0x04, bits 7:0
FIFOOverflow	ErrorFlag	0x0A, bit 4
FilterAmpDet	BPSKDemControl	0x1D, bit 4
FlushFIFO	Control	0x09, bit 0
Force100ASK	TxControl	0x11, bit 4
FramingErr	ErrorFlag	0x0A, bit 2
Gain	RxControl1	0x19, bits 1:0
GsCfgCW	CWConductance	0x12, bits 5:0

Multiple Protocol Contactless Reader IC

CL RC632

Flag(s)	Register	Address Register, Bit Position
GsCfgMod	ModConductance	0x13, bits 5:0
HiAlert	PrimaryStatus	0x03, bit 1
HiAlertIEn	InterruptEn	0x06, bit 1
HiAlertIRq	InterruptRq	0x07, bit 1
IdleIEn	InterruptEn	0x06, bit 2
IdleIRq	InterruptRq	0x07, bit 2
IFDetectBusy	Command	0x01, bit 7
Irq	PrimaryStatus	0x03, bit 3
IRQInv	IRQPinConfig	0x2D, bit 1
IRQPushPull	IRQPinConfig	0x2D, bit 0
ISO Selection	RxControl1	0x19, bits 4:3
KeyErr	ErrorFlag	0x0A, bit 6
LoAlert	PrimaryStatus	0x03, bit 0
LoAlertIEn	InterruptEn	0x06, bit 0
LoAlertIRq	InterruptRq	0x07, bit 0
LPOff	RxControl1	0x19, bit 2
MFOUTSelect	MFOUTSelect	0x26, bits 2:0
MinLevel	RxThreshold	0x1C, bits 7:4
ModemState	PrimaryStatus	0x03 , bit 6:4
ModulatorSource	TxControl	0x11, bits 6:5
ModWidth	ModWidth	0x15, bits 7:0
NoRxEGT	BPSKDemControl	0x1D, bit 6
NoRxEOF	BPSKDemControl	0x1D, bit 5
NoRxSOF	BPSKDemControl	0x1D, bit 7
NoTxEOF	TypeBFRaming	0x17, bit 6
NoTxSOF	TypeBFRaming	0x17, bit 7
PageSelect	Page	0x00, 0x08, 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, bits 2:0
ParityEn	ChannelRedundancy	0x22, bit 0
ParityErr	ErrorFlag	0x0A, bit 1
ParityOdd	ChannelRedundancy	0x22 , bit 1
PowerDown	Control	0x09, bit4
RcvClkSel	RxControl2	0x1E, bit 7
RxAlign	BitFraming	0x0F, bits 6:4

Multiple Protocol Contactless Reader IC

CL RC632

Flag(s)	Register	Address Register, Bit Position
RxAutoPD	RxControl2	0x1E, bit 6
RxCRCEn	ChannelRedundancy	0x22, bit 3
RxCoding	DecoderControl	0x1A, bit 0
RxFraming	DecoderControl	0x1A, bits 4:3
RxIEn	InterruptEn	0x06, bit 3
RxIRq	InterruptRq	0x07, bit 3
RxLastBits	SecondaryStatus	0x05, bits 2:0
RxMultiple	DecoderControl	0x1A, bit 6
RxWait	RxWait	0x21, bits 7:0
SetIEn	InterruptEn	0x06, bit 67
SetIRq	InterruptRq	0x07, bit 7
SignalToMFOUT	TestDigiSelect	0x3D, bit 7
SOFWidth	TypeBFRaming	0x17, bits 1:0
StandBy	Control	0x09, bit 5
SubCPulses	RxControl1	0x19, bits 7:5
TauB	BPSKDemControl	0x1D, bits 1:0
TauD	BPSKDemControl	0x1D, bits 3:2
TautoRestart	TimerClock	0x2A, bit 5
TestAnaOutSel	TestAnaSelect	0x3A, bits 6:4
TestDigiSignalSel	TestDigiSelect	0x3D, bit 6:0
TimerIEn	InterruptEn	0x06, bit 5
TimerIRq	InterruptRq	0x07, bit 5
TimerValue	TimerValue	0x0C, bits 7:0
TimeSlotPeriod	TimeSlotPeriod	0x25, bits 7:0
TimeSlotPeriodMSB	MFOUTSelect	0x26, bit 4
TpreScaler	TimerClock	0x2A, bits 4:0
TReloadValue	TimerReload	0x2C, bits 7:0
TRunning	SecondaryStatus	0x05, bit 7
TstartTxBegin	TimerControl	0x2B, bit 0
TstartTxEnd	TimerControl	0x2B, bit 1
TstartNow	Control	0x09, bit 1
TstopRxBegin	TimerControl	0x2B, bit 2
TstopRxEnd	TimerControl	0x2B, bit 3
TstopNow	Control	0x09, bit 2

Multiple Protocol Contactless Reader IC

CL RC632

Flag(s)	Register	Address Register, Bit Position
TX1RFEn	TxControl	0x11, bit 0
TX2Cw	TxControl	0x11, bit 3
TX2Inv	TxControl	0x11, bit 3
TX2RFEn	TxControl	0x11, bit 1
TxCoding	CoderControl	0x14, bits 2:0
TxCRCEn	ChannelRedundancy	0x22, bit 2
TxIEn	InterruptEn	0x06, bit 4
TxIRq	InterruptRq	0x07, bit 4
TxLastBits	BitFraming	0x0F, bits 2:0
UsePageSelect	Page	0x00, 0x08, 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, bit 7
WaterLevel	FIFOLevel	0x29, bits 5:0
ZeroAfterColl	DecoderControl	0x1A, bit 5

Multiple Protocol Contactless Reader IC

CL RC632

5.4 Modes of Register Addressing

Three mechanisms are valid to operate with the CL RC632:

- Initiating functions and controlling data manipulation by executing *commands*
- Configuring electrical and functional behaviour via a set of *configuration bits*
- Monitoring the state of the CL RC632 by reading *status flags*

The commands, configuration bits and flags are accessed via the µ-Processor interface. The CL RC632 can internally address 64 registers. This basically requires six address lines.

5.4.1 PAGING MECHANISM

The CL RC632 register set is segmented into 8 pages with 8 register each. The *Page-Register* can always be addressed, no matter which page is currently selected.

5.4.2 DEDICATED ADDRESS BUS

Using the CL RC632 with dedicated address bus, the µ-Processor defines three address lines via the address pins A0, A1, and A2. This allows addressing within a page. To switch between registers in different pages the paging mechanism needs then to be used.

The following table shows how the register address is assembled:

Register Bit: <i>UsePageSelect</i>		Register-Address					
1		<i>PageSelect2</i>	<i>PageSelect1</i>	<i>PageSelect0</i>	A2	A1	A0

Table 5-3: Dedicated Address Bus: Assembling the Register Address

5.4.3 MULTIPLEXED ADDRESS BUS

Using the CL RC632 with multiplexed address bus, the µ-Processor may define all 6 address lines at once. In this case either the paging mechanism or linear addressing may be used.

The following table shows how the register address is assembled:

Interface Bus Type	Register Bit: <i>UsePageSelect</i>	Register-Address					
Multiplexed Address Bus (paging mode)	1	<i>PageSelect2</i>	<i>PageSelect1</i>	<i>PageSelect0</i>	AD2	AD1	AD0
Multiplexed Address Bus (linear addressing)	0	AD5	AD4	AD3	AD2	AD1	AD0

Table 5-4: Multiplexed Address Bus: Assembling the Register Address

Multiple Protocol Contactless Reader IC

CL RC632

6 MEMORY ORGANISATION OF THE E²PROM6.1 Diagram of the E²PROM Memory Organisation

Block Number	Block Address	Byte Addresses	Access Rights	Memory Content	See Also
0	0	00 ... 0F	r	Product Information Field	6.2
1	1	10 ... 1F	r/w	Start Up Register Initialisation File	6.3.1
2	2	20 ... 2F	r/w		
3	3	30 ... 3F	r/w	Register Initialisation File For User data or second Initialisation	6.3.3
4	4	40 ... 4F	r/w		
5	5	50 ... 5F	r/w		
6	6	60 ... 6F	r/w		
7	7	70 ... 7F	r/w		
8	8	80 ... 8F	w		
9	9	90 ... 9F	w		
10	A	A0 ... AF	w		
11	B	B0 ... BF	w	Keys for Crypto1	6.4
12	C	C0 ... CF	w		
13	D	D0 ... DF	w		
14	E	E0 ... EF	w		
15	F	F0 ... FF	w		
16	10	100 ... 10F	w		
17	11	110 ... 11F	w		
18	12	120 ... 12F	w		
19	13	130 ... 13F	w		
20	14	140 ... 14F	w		
21	15	150 ... 15F	w		
22	16	160 ... 16F	w		
23	17	170 ... 17F	w		
24	18	180 ... 18F	w		
25	19	190 ... 19F	w		
26	1A	1A0 ... 1AF	w		
27	1B	1B0 ... 1BF	w		
28	1C	1C0 ... 1CF	w		
29	1D	1D0 ... 1DF	w		
30	1E	1E0 ... 1EF	w		
31	1F	1F0 ... 1FF	w		

Table 6-1: Diagram of E²PROM Memory Organisation

Multiple Protocol Contactless Reader IC**CL RC632****6.2 Product Information Field (Read Only)**

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Meaning	Product Type Identification				RFU			Product Serial Number				Internal		RsMaxP	CRC	

*Table 6-2: Product Information Field***PRODUCT TYPE IDENTIFICATION:**

The CL RC632 is a member of a new family for highly integrated reader IC's. Each member of the product family has its unique Product Type Identification. The value of the Product Type Identification is shown in the table below:

Byte	Product Type Identification				
	0	1	2	3	4
Value	30 _{hex}	FF _{hex}	FF _{hex}	0F _{hex}	XX _{hex}

Table 6-3: Product Type Identification Definition

Byte 4 indicates the current version number.

PRODUCT SERIAL NUMBER:

The CL RC632 holds a four byte serial number that is unique for each device.

INTERNAL:

These 2 bytes hold internal trimming parameters.

RsMaxP:

Maximum Source Resistance for the p-Channel Driver Transistor of pin TX1 and TX2

The source resistance of the p-channel driver transistors of pin TX1 and TX2 may be adjusted via the value *GsCfgCW* in the *CWConductance Register* (see chapter 13.3). The mean value of the maximum adjustable source resistance of the pins TX1 and TX2 is stored as an integer value in Ohms in byte RsMaxP.

This value is denoted as maximum adjustable source resistance $Rs_{ref,max,p}$ and is measured setting *GsCfgCW* in the *Register CWConductance* to 01_{hex}. It is in the range between about 80 to 120 Ohm.

CRC:

The content of the product information field is secured via a CRC-byte, which is checked during start up.

Multiple Protocol Contactless Reader IC

CL RC632

6.3 Register Initialisation Files (Read/Write)

Register initialisation in the register address range from 10_{hex} to $2F_{\text{hex}}$ is done automatically during the Initialising Phase (see 11.3), using the Start Up Register Initialisation File.

Furthermore, the user may initialise the CL RC632 registers with values from the Register Initialisation File executing the *LoadConfig-Command* (see 18.7.1).

Notes:

- The *Page-Register* (addressed with 10_{hex} , 18_{hex} , 20_{hex} , 28_{hex}) is skipped and not initialised.
- Make sure, that all *PreSet* registers are not changed.
- Make sure, that all register bits that are reserved for future use (RFU) are set to 0.

6.3.1 START UP REGISTER INITIALISATION FILE (READ/WRITE)

The content of the E²PROM memory block address 1 and 2 are used to initialise the CL RC632 registers 10_{hex} to $2F_{\text{hex}}$ during the Initialising Phase automatically. The default values written into the E²PROM during production are shown chapter 6.3.2.

The assignment is the following:

E ² PROM Byte Address	Register Address	Remark
10_{hex} (Block 1, Byte 0)	10_{hex}	Skipped
11_{hex}	11_{hex}	Copied
...
$2F_{\text{hex}}$ (Block 2, Byte 15)	$2F_{\text{hex}}$	Copied

Table 6-4: Byte Assignment for Register Initialisation at Start Up

Multiple Protocol Contactless Reader IC

CL RC632

6.3.2 SHIPMENT CONTENT OF START UP REGISTER INITIALISATION FILE

During production test, the Start Up Register Initialisation File is initialised with the values shown in the table below. With each power up these values are written into the CL RC632 register during the Initialising Phase.

E ² PROM Byte Address	Reg. Address	Value	Description
10	10	00	<i>Page</i> : free for user
11	11	58	<i>TxControl</i> : Transmitter pins TX1 and TX2 switched off, bridge driver configuration, modulator driven from internal digital circuitry
12	12	3F	<i>CwConductance</i> : Source resistance of TX1 and TX2 to minimum.
13	13	3F	<i>ModConductance</i> : defines the output conductance
14	14	19	<i>CoderControl</i> : ISO14443-A coding is set
15	15	13	<i>ModWidth</i> : Pulse width for Miller pulse coding is set to standard configuration.
16	16	3F	<i>ModWithSOF</i> : Pulse width of SOF
17	17	3B	<i>TypeFraming</i> : ISO 14443-A framing is set
18	18	00	<i>Page</i> : free for user
19	19	73	<i>RxControl1</i> : ISO 14443-A is set and internal amplifier gain is maximum.
1A	1A	08	<i>DecoderControl</i> : A bit-collision always evaluates to HIGH in the data bit stream.
1B	1B	AD	<i>BitPhase</i> : <i>BitPhase</i> is set to standard configuration.
1C	1C	FF	<i>RxThreshold</i> : <i>MinLevel</i> and <i>CollLevel</i> are set to maximum.
1D	1D	1E	<i>BPSKDemControl</i> : ISO14443-A is set
1E	1E	41	<i>RxControl2</i> : Use Q-clock for the receiver, 'Automatic Receiver Off' is switched on, decoder is driven from internal analog circuitry.
1F	1F	00	<i>ClockQControl</i> : 'Automatic Q-clock Calibration' is switched on.
20	20	00	<i>Page</i> : free for user
21	21	06	<i>RxWait</i> : Frame Guard Time is set to six bit clocks.
22	22	03	<i>ChannelRedundancy</i> : Channel Redundancy is set according to ISO14443-A.
23	23	63	<i>CRCResetLSB</i> : CRC-Preset value is set according to ISO14443-A.
24	24	63	<i>CRCResetMSB</i> : CRC-Preset value is set according to ISO14443-A.
25	25	00	<i>TimeSlotPeriod</i> : defines the time for the I \bullet CODE1 time slots
26	26	00	<i>MFOUTSelect</i> : Pin MFOUT is set to LOW.
27	27	00	<i>PreSet27</i>
28	28	00	<i>Page</i> : free for user
29	29	08	<i>FIFOLevel</i> : <i>WaterLevel</i> FIFO buffer warning level is set to standard configuration.
2A	2A	07	<i>TimerClock</i> : <i>TPreScaler</i> is set to standard configuration, timer unit restart function is switched off.
2B	2B	06	<i>TimerControl</i> : Timer is started at the end of transmission, stopped at the beginning of reception.
2C	2C	0A	<i>TimerReload</i> : <i>TReloadValue</i> : the timer unit preset value is set to standard configuration.
2D	2D	02	<i>IRQPinConfig</i> : Pin IRQ is set to high impedance.
2E	2E	00	<i>PreSet2E</i>
2F	2F	00	<i>PreSet2F</i>

Table 6-5: Shipment Content of Start Up Configuration File

Note: The default configuration of the CL RC632 supports the MIFARE® and ISO 14443 A communication scheme. The memory addresses 3 to 7 may be used for user specific initialisation files as I \bullet CODE1, ISO15693 or ISO14443 B.

Multiple Protocol Contactless Reader IC**CL RC632****6.3.3 REGISTER INITIALISATION FILE (READ/WRITE)**

The content of the E²PROM memory from block address 3 to 7 may be used to initialise the CL RC632 registers 10_{hex} to 2F_{hex} by execution of the *LoadConfig-Command* (see 18.7.1). It requires a two bytes argument, used as the two bytes long E²PROM starting byte address for the initialisation procedure.

The assignment is the following:

E ² PROM Byte Address	Register Address	Remark
Starting Byte address for the E ² PROM	10 _{hex}	Skipped
Starting Byte address for the E ² PROM +1	11 _{hex}	Copied
...
Starting Byte address for the E ² PROM + 31	2F _{hex}	Copied

Table 6-6: Byte Assignment for Register Initialisation at Start Up

The Register Initialisation File is big enough to hold the values for two initialisation sets and leaves one more block (16 bytes) for the user.

Note: The Register Initialisation File is read- and write-able for the user. Therefore, these bytes may also be used to store user specific data for other purposes.

The standard configuration for the CL RC632 enables the MIFARE® and ISO14443 setting after each power up.

To give the user the needed flexibility the startup configuration might be adapted and for example the ICODE1 start up configuration might be stored in the register block address 3 and 4.

Multiple Protocol Contactless Reader IC**CL RC632****6.3.4 CONTENT OF I•CODE1 AND ISO15693 START UP REGISTER VALUES**

To enable the I•CODE1 functionality the following table gives an overview on the start up values for communication according to the I•CODE1 and ISO15693 scheme

E2PROM Byte Address	Reg. Address	Value	Description
30	10	00	<i>Page</i> : free for user
31	11	58	<i>TxControl</i> : Transmitter pins TX1 and TX2 switched off, bridge driver configuration, modulator driven from internal digital circuitry
32	12	3F	<i>CwConductance</i> : Source resistance of TX1 and TX2 to minimum.
33	13	05	<i>ModGsCfg</i> : Source resistance of TX1 and TX2 at the time of Modulation, to determine the modulation index
34	14	2C	<i>CoderControl</i> : Selects the bit coding mode and the framing during transmission
35	15	3F	<i>ModWidth</i> : Pulse width for "used code (1 out of 256, RZ or 1 out of 4)" pulse coding is set to standard configuration.
36	16	3F	<i>ModWidthSOF</i> Pulse width of SOF
37	17	00	<i>TypeBFRaming</i>
38	18	00	<i>Page</i> : free for user
39	19	8B	<i>RxControl1</i> : Amplifier gain is maximum.
3A	1A	00	<i>DecoderControl</i> : A bit-collision always evaluates to HIGH in the data bit stream.
3B	1B	54	<i>BitPhase</i> : BitPhase is set to standard configuration.
3C	1C	68	<i>RxThreshold</i> : MinLevel and CollLevel are set to maximum.
3D	1D	00	<i>BPSKDemControl</i>
3E	1E	41	<i>RxControl2</i> : Use Q-clock for the receiver, 'Automatic Receiver Off' is switched on, decoder is driven from internal analog circuitry.
3F	1F	00	<i>ClockQControl</i> : Automatic Q-clock Calibration' is switched on.
40	20	00	<i>Page</i> : free for user
41	21	08	<i>RxWait</i> : Frame Guard Time is set to six bit clocks.
42	22	0C	<i>ChannelRedundancy</i> : Channel Redundancy is set according to I•CODE1.
43	23	FE	<i>CRCResetLSB</i> : CRC-Preset value is set according to I•CODE1.
44	24	FF	<i>CRCResetMSB</i> : CRC-Preset value is set according to I•CODE1.
45	25	00	<i>TimeSlot Period</i> : defines the time for the I•CODE1 time slots
46	26	00	<i>MFOUTSelect</i> : Pin MFOUT is set to LOW.
47	27	00	<i>PreSet27</i>
48	28	00	<i>Page</i> : free for user
49	29	3E	<i>FIFOLevel</i> : WaterLevel: FIFO buffer warning level is set to standard configuration.
4A	2A	0B	<i>TimerClock</i> : TPreScaler is set to standard configuration, timer unit restart function is switched off.
4B	2B	02	<i>TimerControl</i> : Timer is started at the end of transmission, stopped at the beginning of reception.
4C	2C	00	<i>TimerReload</i> : TReloadValue: the timer unit preset value is set to standard configuration
4D	2D	02	<i>IRQPinConfig</i> : Pin IRQ is set to high impedance.
4E	2E	00	<i>PreSet2E</i>
4F	2F	00	<i>PreSet2F</i>

Table 6-7: Content of I•CODE1 Start Up Configuration

6.4 Crypto1 Keys (Write Only)

The MIFARE® Classic security requires specific keys to encrypt the communication on the contactless interface. These keys are named as crypto1 keys.

Multiple Protocol Contactless Reader IC

CL RC632

6.4.1 KEY FORMAT

To store a key in the E²PROM, it has to be written in a specific format. Each key byte has to be split into the lower four bits k0 to k3 (lower nibble) and the higher four bits k4 to k7 (higher nibble). Each nibble is stored twice in one byte and one of the two nibbles is bit-wise inverted. This format is a precondition for successful execution of the *LoadKeyE2*- (see 18.9.1) and the *LoadKey-Command* (see 18.9.2).

With this format, 12 bytes of the E²PROM memory are needed to store a 6 byte long key.

This is shown in the following table:

Master Key Byte	0 (LSB)		1		5 (MSB)	
Master Key Bits	k7 k6 k5 k4 k7 k6 k5 k4	k3 k2 k1 k0	k3 k2 k1 k0	k7 k6 k5 k4 k7 k6 k5 k4	k3 k2 k1 k0	k3 k2 k1 k0
E ² PROM Byte Address	n	n+1	n+2	n+3	n+10	n+11
Example	5A _{hex}	F0 _{hex}	5A _{hex}	E1 _{hex}	5A _{hex}	A5 _{hex}

Table 6-8: Key Storage Format

Example: For the actual key A0 A1 A2 A3 A4 A5_{hex} the value 5A F0 5A E1 5A D2 5A C3 5A B4 5A A5_{hex} must be written into the E²PROM.

Note: Although it is possible to load data of any other format into the key storage location of the E²PROM, it is not possible to obtain a valid card authentication with such a key. The *LoadKeyE2-Command* (see 18.9.1) will fail.

6.4.2 STORAGE OF KEYS IN THE E²PROM

The CL RC632 reserves 384 bytes of memory area in the E²PROM to hold Crypto1 keys. It uses no memory segmentation to mirror the 12 bytes structure of key storage. Thus, every byte of the dedicated memory area may be the start of a key.

Example: If a key loading cycle starts at the last byte address of an E²PROM block, e.g. key byte 0 is stored at 12F_{hex}, the following bytes are stored in the next E²PROM block , e.g. key byte 1 is stored at 130_{hex}, byte 2 at 131_{hex}, up to byte 11 at 13A_{hex}.

With 384 bytes of memory and 12 bytes needed for one key, 32 different keys may be stored in the E²PROM.

Note: It is not possible to load a key exceeding the E²PROM byte location 1FF_{hex}.

Multiple Protocol Contactless Reader IC

CL RC632

7 FIFO BUFFER

7.1 Overview

An 8x64 bit FIFO buffer is implemented in the CL RC632 acting as a parallel-to-parallel converter. It buffers the input and output data stream between the µ-Processor and the internals of the CL RC632. Thus, it is possible to handle data streams with lengths of up to 64 bytes without taking timing constraints into account.

7.2 Accessing the FIFO Buffer

7.2.1 ACCESS RULES

The FIFO-buffer input and output data bus is connected to the *FIFOData Register*. Writing to this register stores one byte in the FIFO-buffer and increments the internal FIFO-buffer write-pointer. Reading from this register shows the FIFO-buffer contents stored at the FIFO-buffer read-pointer and increments the FIFO-buffer read-pointer. The distance between the write- and read-pointer can be obtained by reading the *FIFOLength Register*.

When the µ-Processor starts a command, the CL RC632 may, while the command is in progress, access the FIFO-buffer according to that command. Physically only one FIFO-buffer is implemented, which can be used in input- and output direction. Therefore the µ-Processor has to take care, not to access the FIFO-buffer in an unintended way.

The following table gives an overview on FIFO access during command processing:

Active Command	µ-Processor is allowed to		Remark
	Write to FIFO	Read from FIFO	
StartUp	-	-	
Idle	-	-	
Transmit	✓	-	
Receive	-	✓	
Transceive	✓	✓	µ-Processor has to know the actual state of the command (transmitting or receiving)
WriteE2	✓	-	
ReadE2	✓	✓	The µ-Processor has to prepare the arguments, afterwards only reading is allowed
LoadKeyE2	✓	-	
LoadKey	✓	-	
Authent1	✓	-	
Authent2	-	-	
LoadConfig	✓	-	
CalcCRC	✓	-	

Table 7-1: Allowed Access to the FIFO-Buffer

Multiple Protocol Contactless Reader IC

CL RC632

7.3 Controlling the FIFO-Buffer

Besides writing to and reading from the FIFO-buffer, the FIFO-buffer pointers might be reset by setting the bit *FlushFIFO*. Consequently, *FIFOLength* becomes zero, *FIFOovfl* is cleared, the actually stored bytes are not accessible anymore and the FIFO-buffer can be filled with another 64 bytes again.

7.4 Status Information about the FIFO-Buffer

The μ -Processor may obtain the following data about the FIFO-buffers status:

- Number of bytes already stored in the FIFO-buffer: *FIFOLength*
- Warning, that the FIFO-buffer is quite full: *HiAlert*
- Warning, that the FIFO-buffer is quite empty: *LoAlert*
- Indication, that bytes were written to the FIFO-buffer although it was already full: *FIFOovfl*
FIFOovfl can be cleared only by setting bit *FlushFIFO*.

The CL RC632 can generate an interrupt signal

- If *LoAlertIRq* is set to 1 it will activate Pin IRQ when *LoAlert* changes to 1.
- If *HiAlertIRq* is set to 1 it will activate Pin IRQ when *HiAlert* changes to 1.

The flag *HiAlert* is set to 1 if only *WaterLevel* bytes or less can be stored in the FIFO-buffer. It is generated by the following equation:

$$HiAlert = (64 - FIFOLength) \leq WaterLevel$$

The flag *LoAlert* is set to 1 if *WaterLevel* bytes or less are actually stored in the FIFO-buffer. It is generated by the following equation:

$$LoAlert = FIFOLength \leq WaterLevel$$

Multiple Protocol Contactless Reader IC**CL RC632**

7.5 Register Overview FIFO Buffer

The following table shows the related flags of the FIFO buffer in alphabetic order.

Flags	Register	Address Register, bit position
FIFOLength	FIFOLength	0x04, bits 6-0
FIFOOverflow	ErrorFlag	0x0A, bit 4
FlushFIFO	Control	0x09, bit 0
HiAlert	PrimaryStatus	0x03, bit 1
HiAlertIEn	InterruptIEn	0x06, bit 1
HiAlertIRq	InterruptIRq	0x07, bit 1
LoAlert	PrimaryStatus	0x03, bit 0
LoAlertIEn	InterruptIEn	0x06, bit 0
LoAlertIRq	InterruptIRq	0x07, bit 0
WaterLevel	FIFOLevel	0x29, bits 5-0

Table 7-2. Registers associated with the FIFO Buffer

Multiple Protocol Contactless Reader IC

CL RC632

8 INTERRUPT REQUEST SYSTEM

8.1 Overview

The CL RC632 indicates certain events by setting bit *IRq* in the *PrimaryStatus-Register* and, in addition, by activating pin IRQ. The signal on pin IRQ may be used to interrupt the µ-Processor using its interrupt handling capabilities. This allows the implementation of efficient µ-Processor software.

8.1.1 INTERRUPT SOURCES OVERVIEW

The following table shows the integrated interrupt flags, the related source and the condition for its setting. The interrupt flag *TimerIRq* indicates an interrupt set by the timer unit. The setting is done when the timer decrements from 1 either down to zero (*TAutoRestart flag disabled*) or to the TPreLoad value if *TAutoRestart* is enabled.

The *TxIRq* bit indicates interrupts from different sources. If the transmitter is active and the state changes from sending data to transmitting the end of frame pattern, the transmitter unit sets automatically the interrupt bit. The CRC coprocessor sets *TxIRq* after having processed all data from the FIFO buffer. This is indicated by the flag *CRCReady* = 1. If the E²Prom programming has finished the *TxIRq* bit is set, indicated by the bit *E2Ready* = 1.

The *RxIRq* flag indicates an interrupt when the end of the received data is detected.

The flag *IdleIRq* is set if a command finishes and the content of the command register changes to idle.

The flag *HiAlertIRq* is set to 1 if the *HiAlert* bit is set to one, that means the FIFO buffer has reached the level indicated by the bit *WaterLevel*, see chapter 7.4.

The flag *LoAlertIRq* is set to 1 if the *LoAlert* bit is set to one, that means the FIFO buffer has reached the level indicated by the bit *WaterLevel*, see chapter 7.4.

Interrupt Flag	Interrupt Source	Is set automatically, when
TimerIRq	Timer Unit	the timer counts from 1 to 0
TxIRq	Transmitter	a data stream, transmitted to the card, ends
	CRC-Coprocessor	all data from the FIFO buffer has been processed
	E ² PROM	all data from the FIFO buffer has been programmed
	Receiver	a data stream, received from the card, ends
IdleIRq	Command Register	a command execution finishes
HiAlertIRq	FIFO-buffer	the FIFO-buffer is getting full
LoAlertIRq	FIFO-buffer	the FIFO-buffer is getting empty

Table 8-1: Interrupt Sources

Multiple Protocol Contactless Reader IC

CL RC632

8.2 Implementation of Interrupt Request Handling

8.2.1 CONTROLLING INTERRUPTS AND THEIR STATUS

The CL RC632 informs the µ-Processor about the interrupt request source by setting the according bit in the *InterruptRq Register*. The relevance of each interrupt request bit as source for an interrupt may be masked with the interrupt enable bits of the *InterruptEn Register*.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
InterruptEn	SetlEn	rfu	TimerlEn	TxlEn	RxlEn	IdlelEn	HiAlertlEn	LoAlertlEn
InterruptRq	SetlRq	rfu	TimerlRq	TxlRq	RxlRq	IdlelRq	HiAlertlRq	LoAlertlRq

Table 8-2: Interrupt Control Registers

If any interrupt request flag is set to 1 (showing that an interrupt request is pending) and the corresponding interrupt enable flag is set the status flag *IRq* in the *PrimaryStatus Register* is set to 1. Furthermore different interrupt sources can be set active simultaneously. Therefore, all interrupt request bits are 'OR'ed and connected to the flag *IRq* and forwarded to pin IRQ.

8.2.2 ACCESSING THE INTERRUPT REGISTERS

The interrupt request bits are set automatically by the internal state machines of the CL RC632. Additionally the µ-Processor has access in order to set or to clear them.

A special implementation of the *InterruptRq* and the *InterruptEn* Register allows the change a single bit status without influencing the other ones. If a specific interrupt register shall be set to one, the bit *Setlxx* has to be set to 1 and simultaneously the specific bit has to be set to 1 too. Vice versa, if a specific interrupt flag shall be cleared, a zero has to be written to the *Setlxx* and simultaneously the specific address of the interrupt register has to be set to 1. If a bit content shall not be changed during the setting or clearing phase a zero has to be written to the specific bit location.

Example: writing 3F_{hex} to the *InterruptRq Register* clears all bits as *SetlRq* in this case is set to 0 and all other bits are set to 1. Writing 81_{hex} sets bit *LoAlertlRq* to 1 and leaves all other bits untouched.

8.3 Configuration of Pin IRQ

The logic level of the status flag *IRq* is visible at pin IRQ. In addition, the signal on pin IRQ may be controlled by the following bits of the *IRQPinConfig Register*:

- *IRQInv*: if set to 0, the signal on pin IRQ is equal to the logic level of bit *IRq*. If set to 1, the signal on pin IRQ is inverted with respect to bit *IRq*.
- *IRQPushPull*: if set to 1, pin IRQ has standard CMOS output characteristics otherwise it is an open drain output and an external resistor is necessary to achieve a HIGH level at this pin.

Note: During the Reset Phase (see 11.2) *IRQInv* is set to 1 and *IRQPushPull* to 0. This results in a high impedance at pin IRQ.

Multiple Protocol Contactless Reader IC**CL RC632****8.4 Register Overview Interrupt Request System**

The following table shows the related flags of the Interrupt Request System in alphabetic order.

Flags	Register	Address Register, bit position
HiAlertIEn	InterruptEn	0x06, bit 1
HiAlertIRq	InterruptRq	0x07, bit 1
IdleIEn	InterruptEn	0x06, bit 2
IdleIRq	InterruptRq	0x07, bit 2
IRq	PrimaryStatus	0x03, bit 3
IRQInv	IRQPinConfig	0x07, bit 1
IRQPushPull	IRQPinConfig	0x07, bit 0
LoAlertIEn	InterruptEn	0x06, bit 0
LoAlertIRq	InterruptRq	0x07, bit 0
RxIEn	InterruptEn	0x06, bit 3
RxIRq	InterruptRq	0x07, bit 3
SetIEn	InterruptEn	0x06, bit 7
SetIRq	InterruptRq	0x07, bit 7
TimerIEn	InterruptEn	0x06, bit 5
TimerIRq	InterruptRq	0x07, bit 5
TxIEn	InterruptEn	0x06, bit 4
TxIRq	InterruptRq	0x07, bit 4

Table 8-3 Registers associated with the Interrupt Request System

Multiple Protocol Contactless Reader IC

CL RC632

9 TIMER UNIT

9.1 Overview

A timer is implemented in the CL RC632. It derives its clock from the 13.56 MHz chip-clock. The µ-Processor may use this timer to manage timing relevant tasks.

The timer unit may be used in one of the following configurations:

- Timeout-Counter
- Watch-Dog Counter
- Stop Watch
- Programmable One-Shot
- Periodical Trigger

The timer unit can be used to measure the time interval between two events or to indicate that a specific event occurred after a specific time. The timer can be triggered by events which will be explained in the following, but the timer itself does not influence any internal event (e.g. A timeout during data receiving does not influence the receiving process automatically). Furthermore, several timer related flags are set and these flags can be used to generate an interrupt.

Multiple Protocol Contactless Reader IC

CL RC632

9.2 Implementation of the Timer Unit

9.2.1 BLOCK DIAGRAM

The following block diagram shows the timer module.

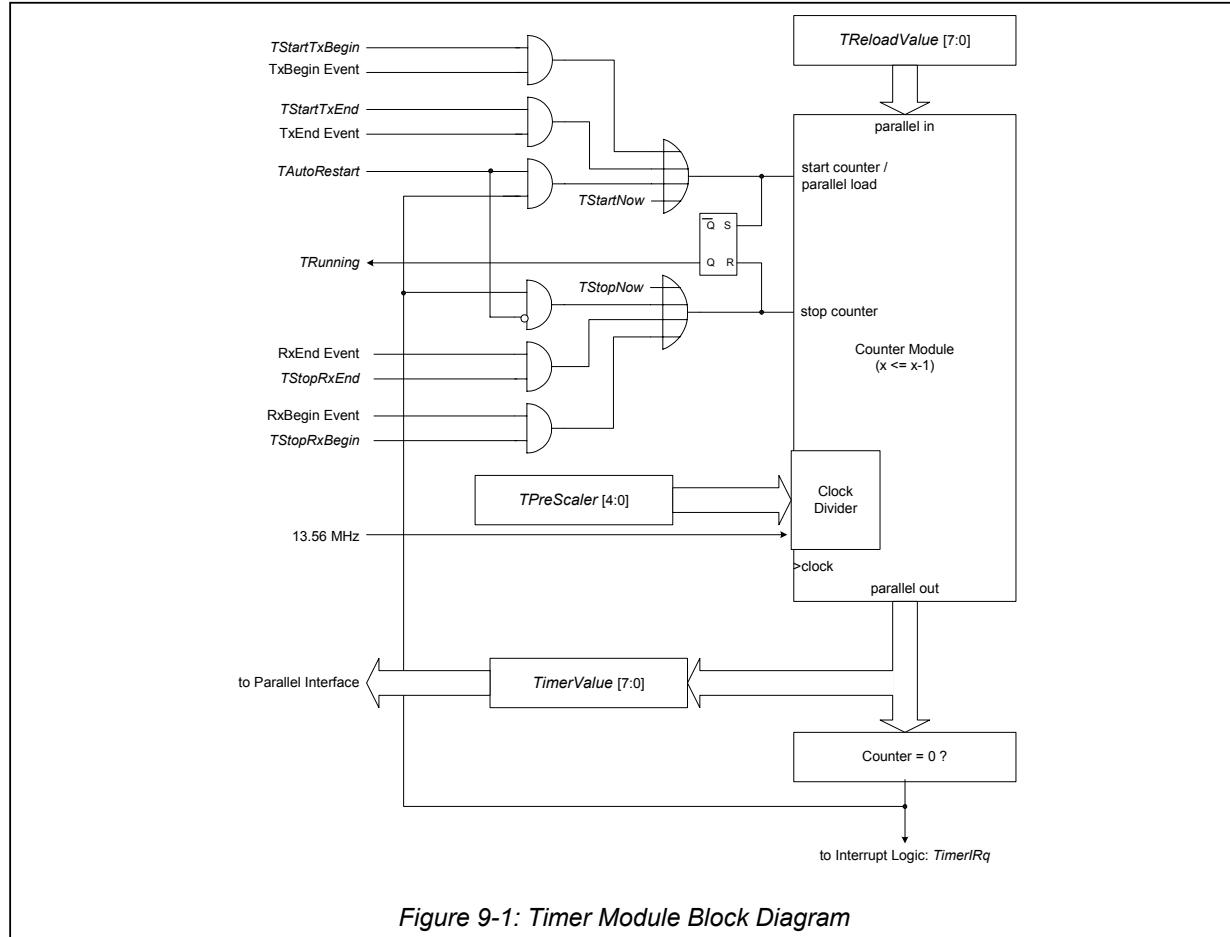


Figure 9-1: Timer Module Block Diagram

The timer unit is designed in a way, that several events in combination with enabling flags start or stop the counter. For example, setting the bit *TStartTxBegin* to 1 enables to control the receiving of data using the timer unit. In addition, the first received bit is indicated by *TxBeginEvent*. This combination starts the counter at the defined *TReloadValue*.

The timer stops either automatically if the counter value is equal to zero, or if a defined stop event happens.

Multiple Protocol Contactless Reader IC

CL RC632

9.2.2 CONTROLLING THE TIMER UNIT

The main part of the timer unit is a down-counter. As long as the down-counter value is unequal zero, it decrements its value with each timer clock.

If *TAutoRestart* is enabled the timer does not decrement down to zero. Having reached the value 1 the timer reloads with the next clock with the *TimerReload* value.

The timer is started immediately by loading a value from the *TimerReload Register* into the counter module. This may be triggered by one of the following events:

- Transmission of the first bit to the card (*TxBegin Event*) and bit *TStartTxBegin* is 1
- Transmission of the last bit to the card (*TxEnd Event*) and bit *TStartTxEnd* is 1
- Bit *TStartNow* is set to 1 (by the µ-Processor)

Note: Every start event reloads the timer from the *TimerReload Register*. Thus, the timer unit is re-triggered.

The timer can be configured to stop with one of the following events:

- Reception of the first valid bit from the card (*RxBegin Event*) and bit *TStopRxBegin* is set to 1
- Reception of the last bit from the card (*RxEnd event*) and bit *TStopRxEnd* is set to 1
- The counter module has decrement down to zero and bit *TAutoRestart* is set to 0
- Bit *TStopNow* is set to 1 (by the µ-Processor)

Loading a new value, e.g. zero, into the *TimerReload Register* does not immediately influence the counter, since the *TimerReload Register* affects the counter units content only with the next start event. Thus, the *TimerReload Register* may be changed even if the timer unit is already counting. The consequence of changing the *TimerReload Register* will be visible after the next start event.

If the counter is stopped by setting bit *TstopNow*, no *TimerIRq* is signalled.

9.2.3 TIMER UNIT CLOCK AND PERIOD

The clock of the timer unit is derived from the 13.56 MHz chip clock via a programmable divider. The clock selection is done with the *TPreScaler Register* that defines the timer unit clock frequency according to the following formula:

$$T_{\text{TimerClock}} = \frac{1}{f_{\text{TimerClock}}} = \frac{2^{T_{\text{PreScaler}}}}{13.56\text{MHz}}$$

The possible values for the *TPreScaler Register* range from 0 up to 21 resulting in minimum time $T_{\text{TimerClock}}$ of about 74 ns up to about 150 ms.

The time period elapsed since the last start event is calculated with

$$T_{\text{Timer}} = \frac{T_{\text{ReloadValue}} - \text{TimerValue}}{f_{\text{TimerClock}}}$$

resulting in a minimum time T_{Timer} of about 74 ns up to about 40 s.

Multiple Protocol Contactless Reader IC

CL RC632

9.2.4 STATUS OF THE TIMER UNIT

The *TRunning* bit in the *SecondaryStatus Register* shows the timer's current status. Any configured start event starts the timer at the *TReloadValue* and changes the status flag *TRunning* to 1, any configured stop event stops the timer and sets the status flag *TRunning* back to 0. As long as status flag *TRunning* is set to 1, the *TimerValue Register* changes with the next timer unit clock.

The actual timer unit content can be read directly via the *TimerValue Register*.

9.2.5 TIMESLOTPERIOD

For sending of I•CODE1-Quit-Frames it is necessary to generate a exact chronological relation to the begin of the command frame.

Is *TimeSlotPeriod* > 0, with the end of command transmission the TimeSlotPeriod starts.

If there are Data in the FIFO after reaching the end of TimeSlotPeriod, these data were sent at that moment.
If the FIFO is empty nothing happens.

As long as the contend of TimeSlotPeriod is > 0 the counter for the TimeSlotPeriod will start automatically after reaching the end.

This allows a exact time relation to the end (as well as to the beginning) of the command frame for the generation and sending of the I•CODE1-Quit-Frames

Is *TimeSlotPeriod* > 0 the next Frame starts exact with the interval

TimeSlotPeriod/CoderRate

delayed after each previous Send Frame. CoderRate defines the clock frequency of the coder.

If *TimeSlotPeriod* = 0, the send function will not be triggered automatically.

The content of the register *TimeSlotPeriod* can be changed during the active mode. The modification take effect at the next restart of the *TimeSlotPeriod*.

Multiple Protocol Contactless Reader IC**CL RC632**

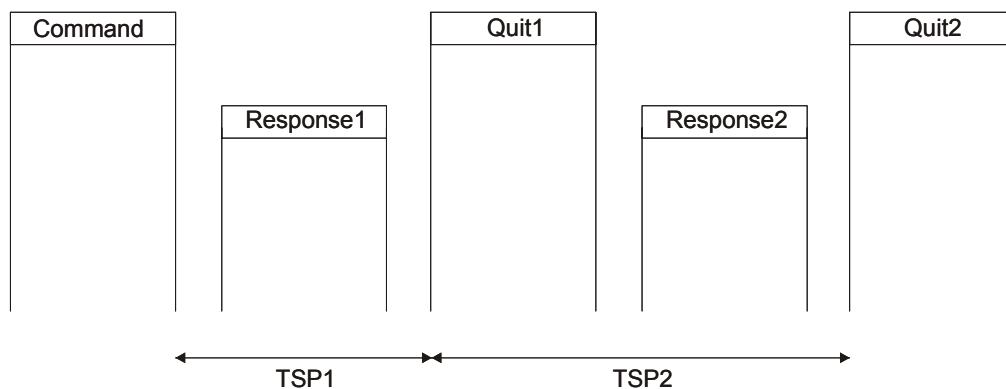
Example:

CoderRate = 0x05 (~52.97kHz)

For ICODE1 standard mode the interval should be 8.458ms

->TimeSlotPeriod = CoderRate * interval = 52.97kHz * 8.458ms -1 = 447 (447 = 0x1BF)

Note: The MSB of the TimeSlotPeriod is in the MFOUTSelect register.



	TimeSlotPeriod for TSP1	TimeSlotPeriod for TSP2
ICODE1 Standard Mode	0xBF	0x1BF
ICODE1 Fast Mode	0x5F	0x67

Note: It is strictly recommended that bit TxCRCEn is set to 0 before the Quit-Frame is sent. If the TxCRCEn is not set to 0 a CRC value is calculated and sent with the Quit-Frame.
To calculate the Quit value a CRC8 algorithm has to be used.

9.3 Usage of the Timer Unit**9.3.1 TIME-OUT- AND WATCH-DOG-COUNTER**

Having started the timer by setting *TReloadValue* the timer unit decrements the *TimerValue Register* beginning with a certain start event. If a certain stop event occurs e.g. a bit is received from the card, the timer unit stops (no interrupt is generated).

On the other hand, if no stop event occurs, e.g. the card does not answer in the expected time, the timer unit decrements down to zero and generates a timer interrupt request. This signals the µ-Processor that the expected event has not occurred in the given time T_{Timer} .

Multiple Protocol Contactless Reader IC

CL RC632

9.3.2 STOP WATCH

The time T_{Timer} between a certain start- and stop event may be measured by the μ -Processor by means of the CL RC632 timer unit. Setting $T_{Reload\ value}$ the timer starts to decrement. If the defined stop event occurs the timers stops. The time between start and stop can be calculated by

$$\Delta T = (T_{Reload\ value} - T_{Timer\ value}) * T_{Timer}$$

if the timer does not decrements down to zero.

9.3.3 PROGRAMMABLE ONE-SHOT TIMER

The μ -Processor starts the timer unit and waits for the timer interrupt. After the specified time T_{Timer} the interrupt will occur.

9.3.4 PERIODICAL TRIGGER

If the μ -Processor sets bit $T_{AutoRestart}$, it will generate an interrupt request periodically after every T_{Timer} .

Multiple Protocol Contactless Reader IC

CL RC632

9.4 Register Overview Timer Unit

The following table shows the related flags of the Timer Unit in alphabetic order.

Flags	Register	Address
TautoRestart	TimerClock	0x2A, bit 5
TimerValue	TimerValue	0x0C, bits 7-0
TimerReloadValue	TimerReload	0x2C, bits 7-0
TpreScaler	TimerClock	0x2A, bits 4-0
Trunning	SecondaryStatus	0x05, bit 7
TstartNow	Control	0x09, bit 1
TstartTxBegin	TimerControl	0x2B, bit 0
TstartTxEnd	TimerControl	0x2B, bit 1
TstopNow	Control	0x09, bit 2
TstopRxBegin	TimerControl	0x2B, bit 2
TstopRxEnd	TimerControl	0x2B, bit 3

Table 9-1 Registers associated with the Timer Unit

Multiple Protocol Contactless Reader IC

CL RC632

10 POWER REDUCTION MODES**10.1 Hard Power Down**

A Hard Power Down is enabled with HIGH on pin RSTPD. This turns off all internal current sinks including the oscillator. All digital input buffers are separated from the input pads and defined internally (except pin RSTPD itself). The output pins are frozen at a certain value.

This is shown in the following table.

SYMBOL	PIN	TYPE	DESCRIPTION
OSCIN	1	I	Not separated from input, pulled to AVSS
IRQ	2	O	High impedance
MFIN	3	I	Separated from Input
Mfout	4	O	LOW
TX1	5	O	HIGH, if TX1RFEn=1
			LOW, if TX1RFEn=0
TX2	7	O	HIGH, only if TX2RFEn=1 and TX2Inv=0
			LOW
NWR	9	I	Separated from Input
NRD	10	I	Separated from Input
NCS	11	I	Separated from Input
D0 to D7	13 to 20	I/O	Separated from Input
ALE	21	I	Separated from Input
A0	22	I/O	Separated from Input
A1	23	I	Separated from Input
A2	24	I	Separated from Input
AUX	27	O	High impedance
RX	29	I	Not changed
VMID	30	A	Pulled to AVDD
RSTPD	31	I	Not changed
OSCOUT	32	O	HIGH

Table 10-1: Signal on Pins during Hard Power Down

10.2 Soft Power Down

The Soft Power Down-mode is entered immediately by setting bit *PowerDown* in the *Control-Register*. All internal current sinks are switched off (including the oscillator buffer).

In difference to the Hard Power Down-mode, the digital input-buffers are not separated by the input pads and keep their functionality. The digital output pins do not change their state.

After resetting bit *PowerDown* in the *Control-Register* it needs 512 clocks until the Soft Power Down mode is left indicated by the *PowerDown* bit itself. Resetting it does not immediately clear it. It is cleared automatically by the CL RC632 when the Soft Power Down-Mode is left.

Note: If the internal oscillator is used, you have to take into account that it is supplied by AVDD and it will take a certain time t_{osc} until the oscillator is stable and the clock cycles can be detected by the internal logic.

Multiple Protocol Contactless Reader IC

CL RC632

10.3 Stand By Mode

The Stand By-mode is entered immediately by setting bit *StandBy* in the *Control-Register*. All internal current sinks are switched off (including the internal digital clock buffer but except the oscillator buffer).

Different from the Hard Power Down-Mode, the digital input-buffers are not separated by the input pads and keep their functionality. The digital output pins do not change their state.

Different from the Soft Power Down-Mode, the oscillator does not need time to wake up.

After resetting bit *StandBy* in the *Control-Register* it needs 4 clocks on pin OSCIN until the Stand By-Mode is left indicated by the *StandBy* bit itself. Resetting it does not immediately clear it. It is cleared automatically by the CL RC632 when the Stand By-Mode is left.

10.4 Receiver Power Down

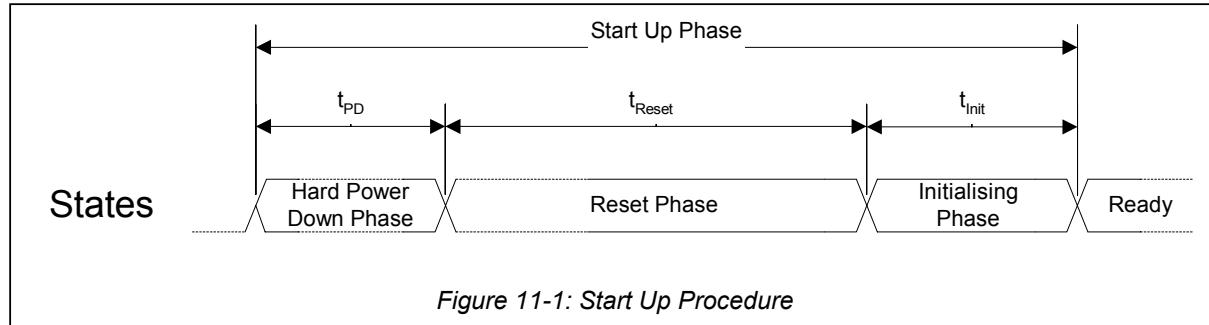
It is power saving to switch off the receiver circuit when it is not needed and switched it on again right before data is to be received from the card. This is done automatically by setting bit *RxAutoPD* to 1. If it is set to 0 the receiver is continuously switched on.

Multiple Protocol Contactless Reader IC

CL RC632

11 START UP PHASE

The phases executed during the start up are shown in the following figure.



11.1 Hard Power Down Phase

The Hard Power Down Phase is active during the following cases:

- Power On Reset caused by power up at pin DVDD
(active while DVDD is below the digital reset threshold)
- Power On Reset caused by power up at pin AVDD
(active while AVDD is below the analog reset threshold)
- A HIGH level on pin RSTPD
(active while pin RSTPD is HIGH)

11.2 Reset Phase

The Reset Phase follows the Hard Power Down Phase automatically. Once the oscillator is running stable, it takes 512 clocks. During the Reset Phase, some of the register bits are pre-set by hardware. The respective reset values are given in the description of each register (see 5.2.).

Note: If the internal oscillator is used, you have to take into account that it is supplied by AVDD and that it will take a certain time t_{osc} until the oscillator is stable.

11.3 Initialising Phase

The Initialising Phase follows the Reset Phase automatically. It takes 128 clocks. During the Initialising Phase the content of the E²PROM blocks 1 and 2 is copied into the registers 10_{hex} to 2F_{hex}. (see 6.3)

Note: At production test, the CL RC632 is initialised with default configuration values. This reduces the µ-Processors effort for configuring the device to a minimum.

Multiple Protocol Contactless Reader IC

CL RC632

11.4 Initialising the Parallel Interface-Type

For the different connections for the different µ-Processor interface types (see 4.3), a certain initialising sequence shall be applied to enable a proper µ-Processor interface type detection and to synchronise the µ-Processor's and the CL RC632's Start Up.

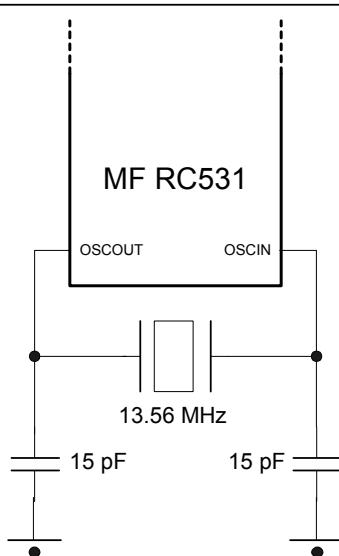
During the whole Start Up Phase, the *Command* value reads as 3F_{hex}. At the end of the Initialising Phase the CL RC632 enters the *Idle Command* automatically. Consequently the *Command* value changes to 00_{hex}.

To ensure proper detection of the µ-Processor interface, the following sequence shall be executed:

- Read from the *Command-Register* until the 6 bit register value for *Command* is 00_{hex}.
The internal initialisation phase is now completed and the CL RC632 is ready to be controlled.
- Write the value 80_{hex} to the *Page-Register* to initialise the µ-Processor interface.
- Read the *Command-Register*. If its value is 00_{hex} the µ-Processor interface initialisation was successful.

Having done the interface initialisation, the linear addressing mode can be activated by writing 0x00 to the page register(s).

Multiple Protocol Contactless Reader IC

CL RC632**12 OSCILLATOR CIRCUITRY***Figure 12-1: Quartz Connection*

The clock applied to the CL RC632 acts as time basis for the coder and decoder of the synchronous system. Therefore stability of the clock frequency is an important factor for proper performance. To obtain highest performance, clock jitter has to be as small as possible. This is best achieved by using the internal oscillator buffer with the recommended circuitry. If an external clock source is used, the clock signal has to be applied to pin OSCIN. In this case special care for clock duty cycle and clock jitter is needed and the clock quality has to be verified. It needs to be in accordance with the specifications in chapter 22.5.3.

Remark: We do not recommend to use an external clock source.

Multiple Protocol Contactless Reader IC

CL RC632

13 TRANSMITTER PINS TX1 AND TX2

The signal delivered on TX1 and TX2 is the 13.56 MHz energy carrier modulated by an envelope signal. It can be used to drive an antenna directly, using a few passive components for matching and filtering (see chapter 20). For that, the output circuitry is designed with a very low impedance source resistance. The signal of TX1 and TX2 can be controlled via the *TxControl Register*.

13.1 Configuration of TX1 and TX2

The configuration possibilities of TX1 are described in the table below:

Register Configuration in <i>TxControl</i>		Envelope	Signal on TX1
<i>TX1RFEn</i>	<i>FORCE100ASK</i>		
0	X	X	LOW (GND)
1	0	0	13.56 MHz carrier frequency modulated
		1	13.56 MHz carrier frequency
1	1	0	LOW
		1	13.56 MHz energy carrier

Table 13-1: Configurations of Pin TX1

The configuration possibilities of TX2 are described in the table below:

Register Configuration in <i>TxControl</i>				Envelope	Signal on TX2
<i>TX2RFEn</i>	<i>FORCE100 ASK</i>	<i>TX2CW</i>	<i>InvTX2</i>		
0	X	X	X	X	LOW
1	0	0	0	0	13.56 MHz carrier frequency modulated
				1	13.56 MHz carrier frequency
		1	1	0	13.56 MHz carrier frequency modulated, 180° phase shift relative to TX1
				1	13.56 MHz carrier frequency, 180° phase shift relative to TX1
	1	0	0	X	13.56 MHz carrier frequency
			1	X	13.56 MHz carrier frequency, 180° phase shift relative to TX1
		1	0	0	LOW
				1	13.56 MHz carrier frequency
	1	0	1	0	HIGH
				1	13.56 MHz carrier frequency, 180° phase shift relative to TX1
		1	0	X	13.56 MHz carrier frequency
			1	X	13.56 MHz carrier frequency, 180° phase shift relative to TX1

Table 13-2: Configurations of Pin TX2

Multiple Protocol Contactless Reader IC

CL RC632

13.2 Operating Distance versus Power Consumption

The user has the possibility to find a trade-off between maximum achievable operating distance and power consumption using different antenna matching circuits by varying the supply voltage at the antenna driver supply pin TVDD. Different antenna matching circuits are described in the Application Note, *MIFARE® Design of MF RC500 Matching Circuit and Antennas*.

13.3 Antenna Driver Output Source Resistance

The output source conductance of TX1 and TX2 for driving a HIGH level may be adjusted via the value *GsCfgCW* in the *CwConductance Register* in the range from about 1 up to 100 Ohm. The output source conductance of Tx1 and TX2 during the modulation phase may be adjusted via the value *GsCfgMod* in the *ModConductance Register* in the same range. The values given are relative to the reference resistance $R_{S_{rel}}$, that is measured during production test and stored in the CL RC632 E²PROM. It can be obtained from the Product Information Field (see chapter 6.2). The electrical specification can be found in chapter 22.4.3.

Multiple Protocol Contactless Reader IC

CL RC632

13.3.1 SOURCE RESISTANCE TABLE

GsCfgCW, GsCfgMod [decimal]	EXP_{GsCfgCW}, EXP_{GsCfgMod} [decimal]	MANT_{GsCfgCW}, MANT_{GsCfgMod} [decimal]	R_{s_{rel}} [Ohm]	GsCfgCW, GsCfgMod [decimal]	EXP_{GsCfgCW}, EXP_{GsCfgMod} [decimal]	MANT_{GsCfgCW}, MANT_{GsCfgMod} [decimal]	R_{s_{rel}} [Ohm]
0	0	0	∞	24	1	8	0,0652
16	1	0	∞	25	1	9	0,0580
32	2	0	∞	37	2	5	0,0541
48	3	0	∞	26	1	10	0,0522
1	0	1	1,0000	27	1	11	0,0474
17	1	1	0,5217	51	3	3	0,0467
2	0	2	0,5000	38	2	6	0,0450
3	0	3	0,3333	28	1	12	0,0435
33	2	1	0,2703	29	1	13	0,0401
18	1	2	0,2609	39	2	7	0,0386
4	0	4	0,2500	30	1	14	0,0373
5	0	5	0,2000	52	3	4	0,0350
19	1	3	0,1739	31	1	15	0,0348
6	0	6	0,1667	40	2	8	0,0338
7	0	7	0,1429	41	2	9	0,0300
49	3	1	0,1402	53	3	5	0,0280
34	2	2	0,1351	42	2	10	0,0270
20	1	4	0,1304	43	2	11	0,0246
8	0	8	0,1250	54	3	6	0,0234
9	0	9	0,1111	44	2	12	0,0225
21	1	5	0,1043	45	2	13	0,0208
10	0	10	0,1000	55	3	7	0,0200
11	0	11	0,0909	46	2	14	0,0193
35	2	3	0,0901	47	2	15	0,0180
22	1	6	0,0870	56	3	8	0,0175
12	0	12	0,0833	57	3	9	0,0156
13	0	13	0,0769	58	3	10	0,0140
23	1	7	0,0745	59	3	11	0,0127
14	0	14	0,0714	60	3	12	0,0117
50	3	2	0,0701	61	3	13	0,0108
36	2	4	0,0676	62	3	14	0,0100
15	0	15	0,0667	63	3	15	0,0093

Table 13-3: Source Resistance of n-Channel Driver Transistor of TX1 and TX2 vs. GsConfCW or GsCfgMod

Multiple Protocol Contactless Reader IC**CL RC632****13.3.2 FORMULA FOR THE SOURCE RESISTANCE**

The relative resistance Rs_{rel} can be calculated by

$$Rs_{rel} = \frac{1}{MANT_{GSCfgCW} \cdot \left(\frac{77}{40}\right)^{EXP_{GSCfgCW}}}$$

The relative resistance Rs_{rel} during the modulation phase can be calculated using $GSCfgMod$, respectively.

13.3.3 CALCULATING THE EFFECTIVE SOURCE RESISTANCE***13.3.3.1 Wiring Resistance***

Wiring and bonding add a constant offset to the driver resistance, that is relevant if TX1 and TX2 are switched to low impedance. The additional resistance for TX1 can be set approximately to

$$Rs_{wire,TX1} \approx 500m\Omega$$

13.3.3.2 Effective Resistance

The source resistances of the driver transistors R_{sMaxP} found in the Product Information Field (see chapter 6.2) are measured at production test with $GSCfgCW$ set to 01_{hex}. To get the driver resistance for a specific value set in $GSCfgMod$ the following formula may be used:

$$Rs_x = (Rs_{ref,max,p} - Rs_{wire,TX1}) \cdot Rs_{rel} + Rs_{wire,TX1} \cdot$$

Multiple Protocol Contactless Reader IC

CL RC632

13.4 Pulse Width

The envelope carries the information of the data signal that shall be transmitted to the card done by coding the data signal according to the Miller code. Furthermore, each pause of the Miller coded signal again is coded as a pulse of certain length. The width of this pulse can be adjusted by means of the *ModWidth Register*. The pulse length is calculated by

$$T_{Pulse} = 2 \frac{ModWidth + 1}{f_c}$$

where $f_c = 13.56\text{MHz}$.

Multiple Protocol Contactless Reader IC

CL RC632

14 RECEIVER CIRCUITRY

14.1 General

The CL RC632 employs an integrated quadrature-demodulation circuit giving the possibility to detect an ISO 14443 compliant subcarrier signal applied to pin RX. The ISO14443-A sub-carrier signal is defined as a Manchester coded ASK-modulated signal. The ISO14443-B sub-carrier signal is defined as an NRZ-L coded BPSK modulated ISO14443-B sub-carrier signal.

The quadrature-demodulator uses two different clocks, Q- and I-clock, with a phase shift of 90° between them. Both resulting sub-carrier signals are amplified, filtered and forwarded to a correlation circuitry. The correlation results are evaluated, digitised and passed to the digital circuitry.

For all processing units various adjustments can be made to obtain optimum performance.

14.2 Block Diagram

Figure 14-1 shows the block diagram of the receiver circuitry. The receiving process includes several steps. First the quadrature demodulation of the carrier signal of 13.56 MHz is done. To achieve an optimum in performance an automatic clock Q calibration is recommended (see 14.3.1). The demodulated signal is amplified by an adjustable amplifier. A correlation circuit calculates the degree of similarity between the expected and the received signal. The bit phase register allows aligning the position of the correlation intervals with the bit grid of the received signal. In the evaluation and digitizer circuitry the valid bits are detected and the digital results are send to the FIFO register. Several tuning steps in this circuit are possible.

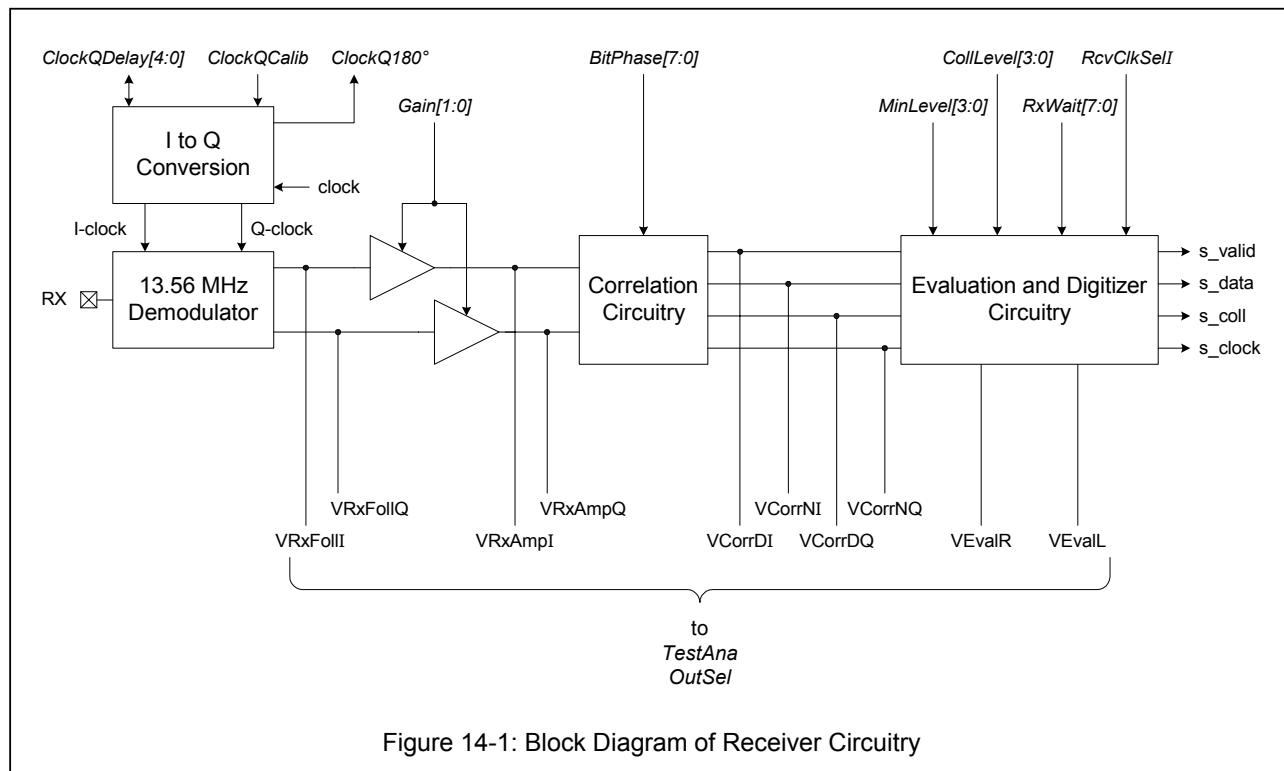


Figure 14-1: Block Diagram of Receiver Circuitry

The user may observe the signal on its way through the receiver as shown in the block diagram above. One signal at a time may be routed to pin AUX using the *TestAnaSelect-Register* as described in 21.3.

Multiple Protocol Contactless Reader IC

CL RC632

14.3 Putting the Receiver into Operation

In general, the default settings programmed in the Start Up Initialisation File are suitable to use the CL RC632 for data communication with MIFARE® cards. However, in some environments specific user settings may achieve better performance.

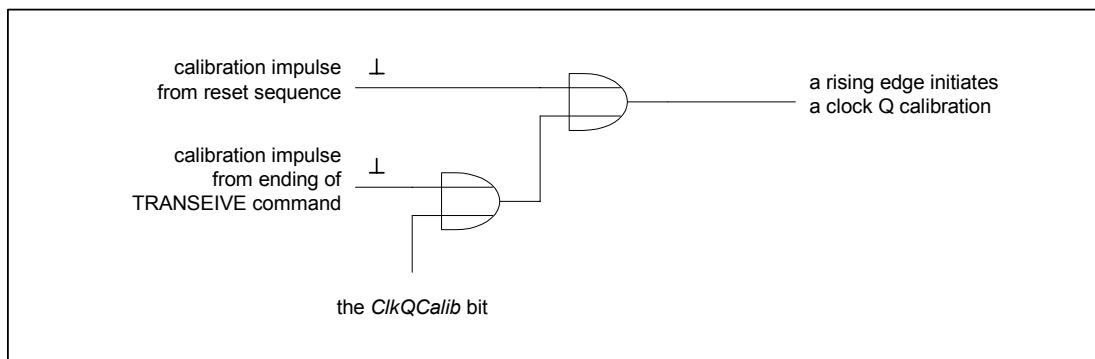
14.3.1 AUTOMATIC CLOCK-Q CALIBRATION

The quadrature demodulation concept of the receiver generates a phase signal I-clock and a 90°-shifted quadrature signal Q-clock. To achieve an optimum demodulator performance, the Q- and the I-clock have to have a difference in phase of 90°. After the reset phase of the CL RC632, a calibration procedure is done automatically. It is possible to have an automatic calibration done at the ending of each Transceive command. To do so, the *C/kQCalib* bit has to be configured to a value of 0.

Configuring this bit to a constant value of 1 disables all automatic calibrations except the one after the reset sequence.

It is also possible to initiate one automatic calibration by software. This is done with a 0 to 1 transition of bit *C/kQCalib*.

The details:



Note: The duration of the automatic clock Q calibration takes 65 oscillator periods which is approx. 4,8μs.

The value of *C/kQDelay* is proportional to the phase shift between the Q- and the I-clock. The status flag *C/kQ180Deg* shows, that the phase shift between the Q- and the I-clock is greater than 180°.

Notes:

- The start-up configuration file enables an automatically Q-clock calibration after the reset.
- While *C/kQCalib* is 1, no automatic calibration is done. Therefore leaving this bit 1 can be used to permanently disable the automatic calibration.
- It is possible to write data to *C/kQDelay* via the μ-Processor. The aim could be a disabling of the automatic calibration and to pre-set the delay by software. But notice, that configuring the delay value by software requires that bit *C/kQCalib* has already been set to 1 before and that a time interval of at least 4,8μs has elapsed since then. Each delay value must be written with the *C/kQCalib* bit set to 1. If *C/kQCalib* is 0 the configured delay value will be overwritten by the next interval automatic calibration.

Multiple Protocol Contactless Reader IC**CL RC632**

14.3.2 AMPLIFIER

The demodulated signal has to be amplified with the variable amplifier to achieve the best performance. The gain of the amplifiers can be adjusted by means of the register bits *Gain [1:0]*. The following gain factors are selectable:

Register Setting	Gain Factor [dB] (Simulation Results)
0	20
1	24
2	31
3	35

Table 14-1: Gain Factors for the Internal Amplifier

Multiple Protocol Contactless Reader IC

CL RC632

14.3.3 CORRELATION CIRCUITRY

The correlation circuitry calculates the degree of matching between the received and an expected signal. The output is a measure for the amplitude of the expected signal in the received signal. This is done for both, the Q- and the I-channel. The correlator delivers two outputs for each of the two input channels, resulting in four output signals in total.

For optimum performance, the correlation circuitry needs the phase information for the signal coming from the card. This information has to be defined by the μ -Processor by means of the register *BitPhase [7:0]*. This value defines the phase relation between the transmitter and receiver clock in multiples of $t_{BitPhase} = 1/13.56$ MHz.

14.3.4 EVALUATION AND DIGITIZER CIRCUITRY

For each bit-half of the Manchester coded signal the correlation results are evaluated. The evaluation and digitizer circuit decides from the signal strengths of both bit-halves, whether the current bit is valid, and, if it is valid, the value of the bit itself or whether the current bit-interval contains a collision.

To do this in an optimum way, the user may select the following levels:

- *MinLevel*: Defines the minimum signal strength of the stronger bit-half's signal for being considered valid.
- *CollLevel*: Defines the minimum signal strength that has to be exceeded by the weaker half-bit of the Manchester-coded signal to generate a bit-collision. If the signal's strength is below this value, a 1 and 0 can be determined unequivocally.

CollLevel defines the minimum signal strength relative to the amplitude of the stronger half-bit.

After transmission of data, the card is not allowed to send its response before a certain time period, called frame guard time in the standard ISO14443. The length of this time period after transmission shall be set in the *RxWait-Register*. The *RxWait-Register* defines when the receiver is switched on after data transmission to the card in multiples of one bit-duration.

If register bit *RcvClkSelI* is set to 1, the I-clock is used to clock the correlator and evaluation circuits. If set to 0, the Q-clock is used.

Note: It is recommended to use the Q-clock.

Multiple Protocol Contactless Reader IC

CL RC632

15 SERIAL SIGNAL SWITCH

15.1 General

Two main blocks are implemented in the CL RC632. A digital circuitry, comprising state machines, coder and decoder logic and so on and an analog circuitry with the modulator and antenna drivers, receiver and amplification circuitry. The interface between these two blocks can be configured in the way, that the interfacing signals may be routed to the pins MFIN and MFOUT.

This topology supports, that the analog part of the one CL RC632 may be connected to the digital part of another device.

The serial signal switch can be used to measure MIFARE® and ISO14443 as well ICODE1 and ISO15693 related signals.

Note: The MFIN pin can only be accessed by 106 kbaud according to ISO14443A . The Manchester with Subcarrier- and the Manchester signal can only be accessed at the MFOUT pin at 106 kbaud according to ISO14443A.

15.2 Block Diagram

Figure 15-1 describes the serial signal switches. Three different switches are implemented in the serial signal switch in order to use the CL RC632 in different configurations.

The serial signal switch may also be used during the design In phase or for test purposes to check the transmitted and received data. Chapter 21.2 describes analog test signals as well as measurements at the serial signal switch.

Note: The SL RC400 uses the name SIGOUT for the MFOUT pin. The CLRC 632 functionality includes the

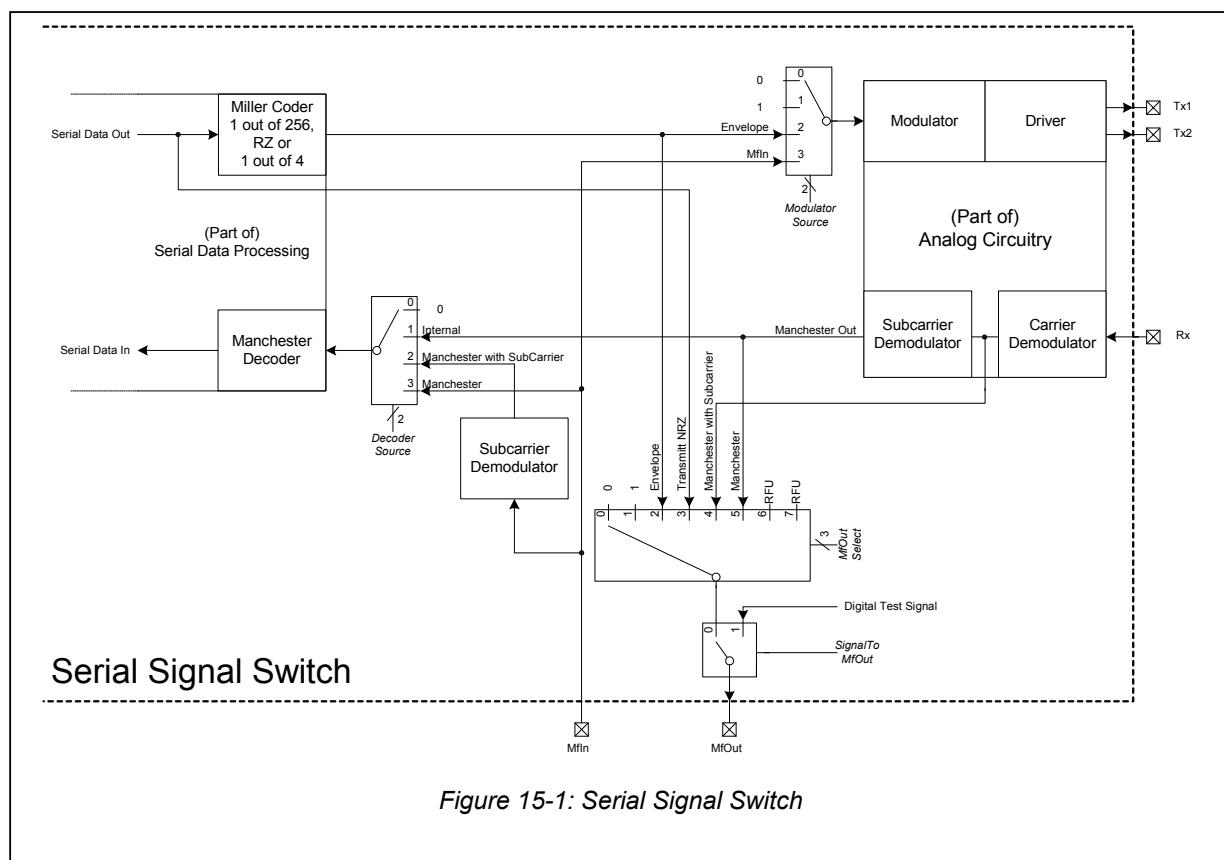


Figure 15-1: Serial Signal Switch

Multiple Protocol Contactless Reader IC

CL RC632

test possibilities for the SL RC 400 using the pin MFOUT.

The following chapters describe the relevant registers used to configure and control the serial signal switch.

15.3 Registers Relevant for the Serial Signal Switch

The flags *DecoderSource* define the input signal for the internal Manchester decoder in the following way:

DecoderSource	Input Signal for Decoder
0	Constant 0
1	Output of the analog part. This is the default configuration.
2	Direct connection to MFIN, expecting an 847.5 kHz sub-carrier signal modulated by a Manchester coded signal.
3	Direct connection to MFIN, expecting a Manchester coded signal.

Table 15-1: Values for DecoderSource

ModulatorSource defines the signal that modulates the transmitted 13.56 MHz energy carrier. The modulated signal drives the pins TX1 and TX2.

ModulatorSource	Input Signal for Modulator
0	Constant 0 (energy carrier off at pin TX1 and TX2).
1	Constant 1 (continuous energy carrier delivered at pin TX1 and TX2).
2	Modulation signal (envelope) from the internal coder. This is the default configuration.
3	Direct connection to MFIN, expecting a Miller pulse coded signal.

Table 15-2: Values for ModulatorSource

MFOUTSelect selects the output signal, which is routed to the pin MFOUT.

MFOUTSelect	Signal Routed to Pin MFOUT
0	Constant Low
1	Constant High
2	Modulation signal (envelope) from the internal coder.
3	Serial data stream that is to be transmitted (same as for <i>MFOUTSelect</i> = 2, but not coded by the selected pulse coder yet).
4	Output signal of the receiver circuit (card modulation signal regenerated and delayed)
5	Output signal of the subcarrier demodulator (Manchester-coded card signal)
6	RFU
7	RFU

Table 15-3: Values for MFOUTSelect

To use *MFOUTSelect*, the value of test signal control bit *SignalToMFOUT* has to be 0.

Multiple Protocol Contactless Reader IC

CL RC632

MIFARE® : Usage of the MFIN and MFOUT

15.3.1 ACTIVE ANTENNA CONCEPT

The CL RC632 analog circuitry may be used via the pins MFIN and MFOUT. To do so, the following register settings have to be made:

Register	Value	Signal	At CL RC632 Pin
ModulatorSource	3	Miller Pulse Coded	MFIN
MFOUTSelect	4	Manchester Coded with sub-carrier	MFOUT
DecoderSource	X	-	-

Table 15-4: Register setting to use the CL RC632 analog circuitry only

On the other hand, the CL RC632 digital circuitry may be used via the pins MFIN and MFOUT. To do so, the following register settings have to be made:

Register	Value	Signal	At CL RC632 Pin
ModulatorSource	X	-	-
MFOUTSelect	2	Miller Pulse Coded	MFOUT
DecoderSource	2	Manchester Coded with sub-carrier	MFIN

Table 15-5: Register setting to use the CL RC632 digital circuitry only

Two CL RC632 devices configured in the above described way may be connected to each other via the pins MFOUT and MFIN.

Note: The usage of the active antenna concept is only possible with a baudrate of 106kbaud according to ISO14443A.

15.3.2 DRIVING TWO RF-PARTS

It is possible, to connect a 'passive antenna' to pins TX1, TX2 and RX (via the appropriate filter and matching circuit) and at the same time an Active Antenna to the pins MFOUT and MFIN.

In this configuration, two RF-parts may be driven (one after another) by one µ-Processor.

Multiple Protocol Contactless Reader IC

CL RC632

16 MIFARE® HIGHER BAUDRATES

The MIFARE® Classic system is specified with a fix Baud-rate of 106 kBaud for the communication on the RF interface. ISO 14443 in the existing version also defines 106 kBaud at least for the initial phase of a communication between PICC and PCD.

To speed up the communication between a terminal and a card to cover requirements for large data transmission the CL RC632 supports the MIFARE® higher baudrates communication in combination with e.g. a µController IC like the MIFARE® ProX.

Communication direction	Baudrates [kbaud]
CL RC632 based PCD → µC PICC supporting higher baudrates	106, 212, 424
µC PICC supporting higher baudrates → CL RC632 based PCD	106, 212, 424

Table 16-1 MIFARE® Higher Baudrates

The MIFARE® Higher Baudrates' concept will be described in the Application Note: '*MIFARE® Implementation of Higher Baudrates*'. This Application Note will cover also the integration a MIFARE® Higher Baudrates communication concept in current applications.

Multiple Protocol Contactless Reader IC**CL RC632****17 ISO14443 B**

The international standard ISO14443 standard covers 2 communication schemes: the ISO14443-A and the ISO14443-B.

The CL RC632 reader IC fully supports the ISO14443.

The following registers and flags cover the ISO 14443B communication scheme:

Flags	Register	Address
CharSpacing	TypeBFraming	0x17, bits 4-3
CoderRate	CoderControl	0x14, bits 5-3
EOFWidth	TypeBFraming	0x17, bit 5
FilterAmpDet	BPSKDemControl	0x1D, bit 4
Force100ASK	TxControl	0x11, bit 4
GSCfgCW	CWConductance	0x12, bits 5-0
GSCfgMod	ModConductance	0x13, bits 5-0
MinLevel	RxTreshold	0x1C, bits 7-4
NoTxEOF	TypeBFraming	0x17, bit 6
NoTxSOF	TypeBFraming	0x17, bit 7
NoRxEGT	BPSKDemControl	0x1D, bit 6
NoRxEOF	BPSKDemControl	0x1D, bit 5
NoRxSOF	BPSKDemControl	0x1D, bit 7
RxCoding	DecoderControl	0x1A,bit 0
RxFraming	DecoderControl	0x1A,bits 4-3
SOFWidth	TypeBFraming	0x17,bits 1-0
SubCPulses	RxControl1	0x19, bits 7-5
TauB	BPSKDemControl	0x1D, bits 1-0
TauD	BPSKDemControl	0x1D, bits 3-2
TxCoding	CoderControl	0x14, bits 2-0

Table 17-1 Registers associated with ISO14443-B

As a reference documentation the international standard *ISO14443 'Identification cards- Contactless integrated circuit(s) cards- Proximity cards, part 1-4'* can be taken.

Note: Philips Semiconductors does not offer a basic function library to design in the ISO14443 B protocol.

Multiple Protocol Contactless Reader IC

CL RC632

18 CL RC632 COMMAND SET

18.1 General Description

The CL RC632 behaviour is determined by an internal state machine capable to perform a certain set of commands. The commands can be started by writing the according command-code to the *Command-Register*.

Arguments and/or data necessary to process a command are mainly exchanged via the FIFO buffer.

18.2 General Behaviour

- Each command, that needs a data stream (or data byte stream) as input will immediately process the data it finds in the FIFO buffer.
- Each command that needs a certain number of arguments will start processing only when it has received the correct number of arguments via the FIFO buffer.
- The FIFO buffer is not cleared automatically at command start. Therefore, it is also possible to write the command arguments and/or the data bytes into the FIFO buffer and start the command afterwards.
- Each command (except the *StartUp-Command*) may be interrupted by the µ-Processor by writing a new command code into the *Command-Register* e.g.: the *Idle-Command*.

18.3 CL RC632 Commands Overview

Command	Code	Action	Arguments and Data passed via FIFO	Returned Data via FIFO	see Chapter
StartUp	3F _{hex}	Runs the Reset- and Initialisation Phase. <u>Note:</u> This command can not be activated by software, but only by a Power-On or Hard Reset	-	-	18.3.2
Idle	00 _{hex}	No action; cancels current command execution.	-	-	18.3.3
Transmit	1A _{hex}	Transmits data from the FIFO buffer to the card.	Data Stream	-	18.4.1
Receive	16 _{hex}	Activates receiver circuitry. <u>Note:</u> Before the receiver actually starts, the state machine waits until the time configured in the register <i>RxWait</i> has passed. <u>Note:</u> This command may be used for test purposes only, since there is no timing relation to the <i>Transmit-Command</i> .	-	Data Stream	18.4.2

Multiple Protocol Contactless Reader IC

CL RC632

CL RC632 Commands Overview Continued

Command	Code	Action	Arguments and Data passed via FIFO	Returned Data via FIFO	see Chapter
Transceive	1E _{hex}	<p>Transmits data from FIFO buffer to the card and activates automatically the receiver after transmission.</p> <p><u>Note:</u> Before the receiver actually starts, the CL RC632 waits until the time configured in the register <i>RxWait</i> has passed.</p> <p><u>Note:</u> This command is the combination of Transmit and Receive</p>	Data Stream	Data Stream	18.4.3
WriteE2	01 _{hex}	Gets data from FIFO buffer and writes it to the internal E ² PROM.	Start Address LSB Start Address MSB Data Byte Stream	-	18.6.1
ReadE2	03 _{hex}	Reads data from the internal E ² PROM and puts it into the FIFO buffer. <u>Note:</u> Keys cannot be read back	Start Address LSB Start Address MSB Number of Data Bytes	Data Bytes	18.6.2
LoadKeyE2	0B _{hex}	Copies a key from the E ² PROM into the key buffer. <u>Note:</u> related to MIFARE® Classic Security	Start Address LSB Start Address MSB	-	18.9.1
LoadKey	19 _{hex}	Reads a key from the FIFO buffer and puts it into the key buffer. <u>Note:</u> The key has to be prepared in a specific format (refer to 6.4.1, key format) <u>Note:</u> related to MIFARE® Classic Security	Byte0 (LSB) Byte1 ... Byte 10 Byte11 (MSB)	-	18.9.2
Authent1	0C _{hex}	Performs the first part of the Crypto1 card authentication. <u>Note:</u> related to MIFARE® Classic Security	Card's Auth-Command Card's Block Address Card's Serial Number LSB Card's Serial Number Byte1 Card's Serial Number Byte2 Card's Serial Number MSB	-	18.9.3
Authent2	14 _{hex}	Performs the second part of the card authentication using the Crypto1 algorithm. <u>Note:</u> related to MIFARE® Classic Security	-	-	18.9.4
LoadConfig	07 _{hex}	Reads data from E ² PROM and initialises the CL RC632 registers.	Start Address LSB Start Address MSB	-	18.7.1
CalcCRC	12 _{hex}	Activates the CRC-Coprocessor. <u>Note:</u> The result of the CRC calculation can be read from the registers <i>CRCResultLSB</i> and <i>CRCResultMSB</i>	Data Byte-Stream	-	18.7.2

Table 18-1: CL RC632 Command Overview

Multiple Protocol Contactless Reader IC**CL RC632****18.3.1 BASIC STATES****18.3.2 STARTUP COMMAND 3F_{HEX}**

Command	Code_{hex}	Action	Arguments and Data	Returned Data
StartUp	3F	Runs the Reset- and Initialisation Phase <u>Note:</u> This command can not be activated by software, but only by a Power-On or Hard Reset	-	-

The *StartUp-Command* runs the Reset- and Initialisation Phase. It does not need or return any data. It can not be activated by the µ-Processor but is started automatically after one of the following events:

- Power On Reset caused by power up at Pin DVDD
- Power On Reset caused by power up at Pin AVDD
- Negative Edge at Pin RSTPD

The Reset-Phase defines certain register bits by an asynchronous reset. The Initialisation-Phase defines certain registers with values taken from the E²PROM.

When the *StartUp-Command* has finished, the *Idle-Command* is entered automatically.

Notes:

- The µ-Processor must not write to the CL RC632 as long as the CL RC632 is busy executing the *StartUp-Command*. To ensure this, the µ-Processor shall poll for the *Idle-Command* to determine the end of the Initialisation Phase (see also chapter 11.4).
- As long as the *StartUp-Command* is active, only reading from page 0 of the CL RC632 is possible.
- The *StartUp-Command* can not be interrupted by the µ-Processor.

18.3.3 IDLE COMMAND 00_{HEX}

Command	Code_{hex}	Action	Arguments and Data	Returned Data
Idle	00	No action, cancels current command execution	-	-

The *Idle-Command* switches the CL RC632 to its inactive state. In this Idle-state it waits for the next command. It does not need or return any data. The device automatically enters the Idle-state when a command finishes. In this case the CL RC632 simultaneously initiates an interrupt request by setting bit *IdleRq*. Triggered by the µ-Processor, the *Idle-Command* may be used to stop execution of all other commands (except the *StartUp Command*). In that case no *IdleRq* is generated.

Remark: Stopping a command with the *Idle Command* does not clear the FIFO buffer content.

Multiple Protocol Contactless Reader IC

CL RC632

18.4 Commands for ISO14443 A Card Communication

The CL RC632 is a fully ISO 14443 and ISO15693 and I²CODE1 compliant reader IC. Therefore, the command set of this IC allows more flexibility and more generalised commands compared to MIFARE[®] or I²CODE1 dedicated reader ICs. The following chapter describes the command set for card communication for ISO14443 A related communication schemes.

18.4.1 TRANSMIT COMMAND 1A_{HEX}

Command	Code _{hex}	Action	Arguments and Data	Returned Data
Transmit	1A	Transmits data from FIFO buffer to the card	Data Stream	-

The *Transmit-Command* takes data from the FIFO buffer and forwards it to the transmitter. It does not return any data. The *Transmit-Command* can only be started by the µ-Processor.

18.4.1.1 Working with the Transmit Command

To transmit data one of the following sequences may be used:

1. All data, that shall be transmitted to the card is written to the FIFO while the *Idle-Command* is active. After that, the command code for the *Transmit-Command* is written to the *Command-Register*.
Note: This is possible for transmission of data with a length of up to 64 bytes.
2. The command code for the *Transmit-Command* is written to the *Command-Register* first. Since no data is available in the FIFO, the command is only enabled but transmission is not triggered yet.
Data transmission really starts with the first data byte written to the FIFO. To generate a continuous data stream on the RF-interface, the µ-Processor has to put the next data bytes to the FIFO in time.
Note: This allows transmission of data of any length but requires that data is available in the FIFO in time.
3. A part of the data, that shall be transmitted to the card is written to the FIFO while the *Idle-Command* is active. After that, the command code for the *Transmit-Command* is written to the *Command-Register*. While the *Transmit-Command* is active, the µ-Processor may feed further data to the FIFO, causing the transmitter to append it to the transmitted data stream.
Note: This enables transmission of data of any length but requires that data is available in the FIFO in time.

When the transmitter requests the next data byte to keep the data stream on the RF-interface continuous but the FIFO buffer is empty, the *Transmit-Command* automatically terminates. This causes the internal state machine to change its state from Transmit to Idle.

If data transmission to the card is finished, the CL RC632 sets the flag *TxIRq* to signal it to the µ-Processor.

Remark: If the µ-Processor overwrites the transmit code in the *Command-Register* with the *Idle-Command* or any other command, transmission stops immediately with the next clock cycle. This may produce output signals that are not according to ISO14443-A.

18.4.1.2 RF-Channel Redundancy and Framing

Each transmitted ISO14443 frame consists of a SOF (start of frame) pattern, followed by the data stream and is closed by an EOF (end of frame) pattern. These different phases of the transmit sequence may be monitored by watching *ModemState* of *PrimaryStatus-Register* (see 18.4.4).

Multiple Protocol Contactless Reader IC

CL RC632

Depending on the setting of bit *TxCRCEn* in the *ChannelRedundancy-Register* a CRC is calculated and appended to the data stream. The CRC is calculated according the settings in the *ChannelRedundancy Register*. Parity generation is handled according the settings in the *ChannelRedundancy-Register* (bits *ParityEn* and *ParityOdd*).

18.4.1.3 Transmission of Bit Oriented Frames

The transmitter may be configured to send an incomplete last byte. To achieve this *TxLastBits* has to be set to a value unequal zero. This is shown in the figure below.

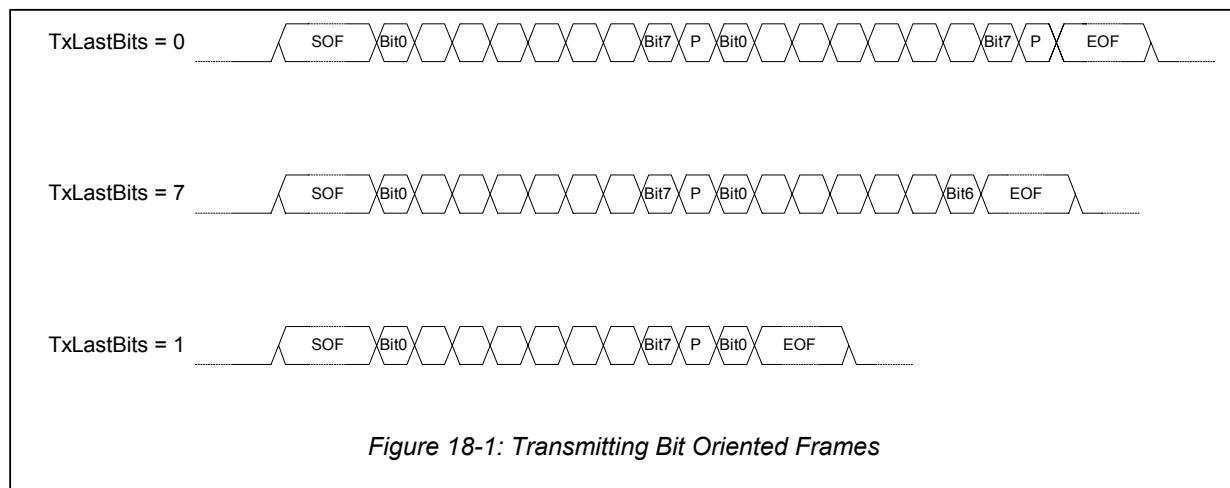


Figure 18-1: Transmitting Bit Oriented Frames

The figure shows the data stream if *ParityEn* is set in *ChannelRedundancy-Register*. All fully transmitted bytes are followed by a parity check bit, but the incomplete byte is not followed by a parity check bit. After transmission, *TxLastBits* is cleared automatically.

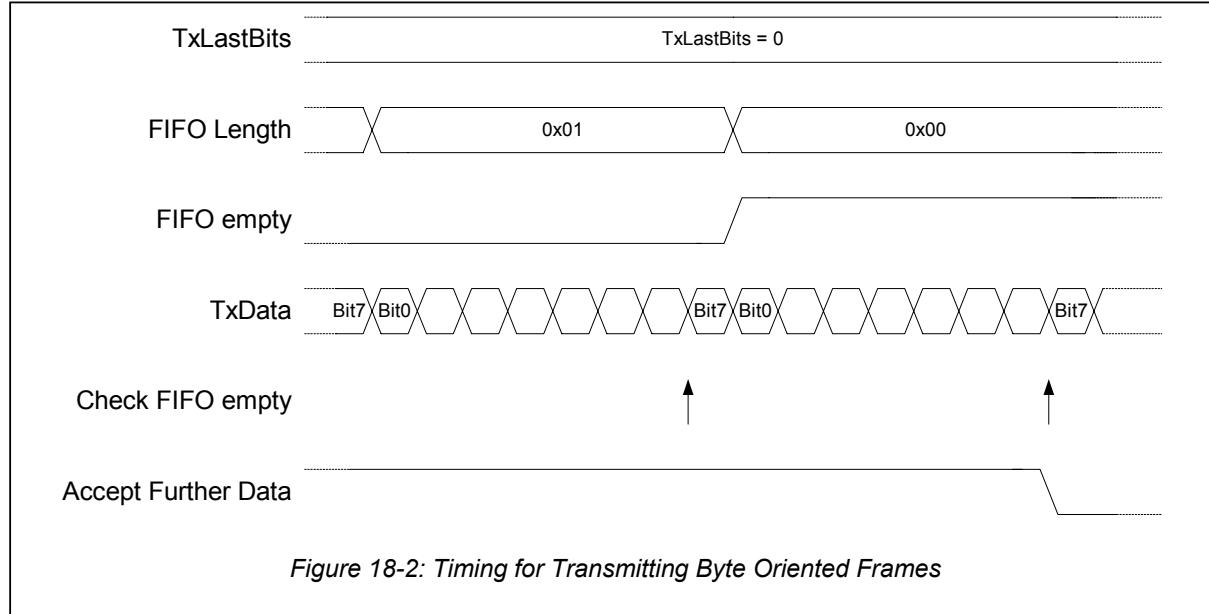
Note: If *TxLastBits* is not equal to zero CRC generation has to be disabled. This is done by clearing the bit *TxCRCEn* in the *ChannelRedundancy Register*.

Multiple Protocol Contactless Reader IC

CL RC632

18.4.1.4 *Transmission of Frames with more than 64 Bytes*

To generate frames with more than 64 bytes, the µ-Processor has to write data into the FIFO buffer while the *Transmit Command* is active. The state machine checks the FIFO status when it starts transmitting the last bit of the actual data stream (the check time is marked below with arrows).



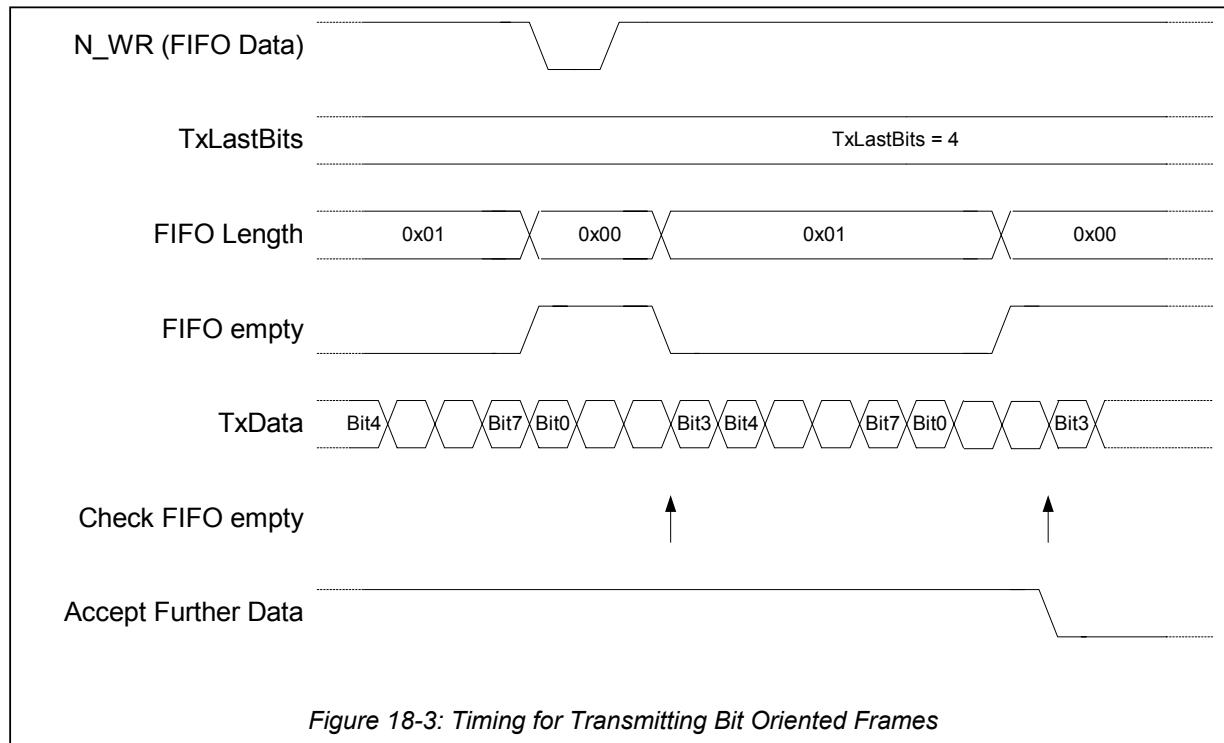
As long as the internal signal 'Accept Further Data' is 1 further data may be loaded to the FIFO. The CL RC632 appends this data to the data stream transmitted via the RF-interface.
If the internal signal 'Accept Further Data' is 0 the transmission will terminate. All data written into the FIFO buffer after 'Accept Further Data' went 0 will not be transmitted anymore, but remain in the FIFO buffer.

Remark: If parity generation is enabled (*ParityEn* bit is 1) the parity bit is the last bit to be transmitted. This delays the signal 'Accept Further Data' for one bit duration.

Multiple Protocol Contactless Reader IC

CL RC632

If *TxLastBits* is unequal zero the last byte is not transmitted completely, but only the number of bits set in *TxLastBits* are transmitted (starting with the least significant bit). Thus, the internal state machine has to check the FIFO status at an earlier point in time (shown in the figure below).



Since *TxLastBits* = 4 in this example, transmission stops after Bit 3 is transmitted. If configured, the frame is completed with an EOF.

The figure above also shows a write access to the *FIFOData Register* right before the FIFO's status is checked. This leads to 'FIFO empty' going to 0 again and therefore 'Accept Further Data' stays active. The new byte written is transmitted via the RF-interface.

'Accept Further Data' is changed only by the 'Check FIFO empty' function. This function verifies 'FIFO empty' one bit duration before the last expected bit transmission.

Frame Definition	Verification at:
8 Bit with Parity	8 th Bit
8 Bit without Parity	7 th Bit
x Bit without Parity	(x-1) th Bit

Multiple Protocol Contactless Reader IC

CL RC632

18.4.2 RECEIVE COMMAND 16_{HEX}

Command	Code _{hex}	Action	Arguments and Data	Returned Data
Receive	16	Activates Receiver Circuitry	-	Data Stream

The *Receive-Command* activates the receiver circuitry. All data received from the RF interface is returned via the FIFO buffer. The *Receive-Command* can be started either by the µ-Processor or automatically during execution of the *Transceive-Command*.

Note: This command may be used for test purposes only, since there is no timing relation to the *Transmit-Command*.

18.4.2.1 Working with the Receive Command

After starting the *Receive Command* the internal state machine decrements the value set in the *RxWait-Register* with every bit-clock. From 3 down to 1 the analog receiver circuitry is prepared and activated. When the counter reaches 0, the receiver starts monitoring the incoming signal at the RF-interface. If the signal strength reaches a level higher than the value set in the *MinLevel-Register* it finally starts decoding. The decoder stops, if no more signal can be detected on the receiver input pin Rx. The decoder indicates termination of operation by setting bit *RxIRq*.

The different phases of the receive sequence may be monitored by watching *ModemState* of the *PrimaryStatus-Register* (see 18.4.4).

Note: Since the counter values from 3 to 0 are necessary to initialise the analog receiver circuitry the minimum value for *RxWait* is 3.

18.4.2.2 RF-Channel Redundancy and Framing

The decoder expects a SOF pattern at the beginning of each data stream. If a SOF is detected, it activates the serial to parallel converter and gathers the incoming data bits. Every completed byte is forwarded to the FIFO. If an EOF pattern is detected or the signal strength falls below *MinLevel* set in the *RxThreshold-Register*, the receiver and the decoder stop, the *Idle-Command* is entered and an appropriate response for the µ-Processor is generated (interrupt request activated, status flags set).

If bit *RxCRCEn* in the *ChannelRedundancy Register* is set a CRC block is expected. The CRC block may be one byte or two bytes according to bit *CRC8* in the *ChannelRedundancy Register*.

Remark: The received CRC block is not forwarded to the FIFO buffer if it is correct. This is realised by shifting the incoming data bytes through an internal buffer of either one or two bytes (depending on the defined CRC). The CRC block remains in this internal buffer. As a consequence all data bytes are available in the FIFO buffer one or two bytes delayed.

If the CRC fails all received bytes are forwarded to the FIFO buffer (including the faulty CRC itself).

If *ParityEn* is set in the *ChannelRedundancy Register* a parity bit is expected after each byte. If bit *ParityOdd* is set to 1, the expected parity is an odd parity, otherwise an even parity is expected.

Multiple Protocol Contactless Reader IC**CL RC632****18.4.2.3 Collision Detection**

If more than one card is within the RF-field during the card selection phase, they will respond simultaneously. The CL RC632 supports the algorithm defined in ISO14443-A to resolve data-collisions of cards serial numbers by doing the so-called anti-collision procedure. The basis for this is the ability to detect bit-collisions.

Bit-collision detection is supported by the used bit-coding scheme, namely the Manchester-coding. If in the first and second half-bit of a bit a sub-carrier modulation is detected, instead of forwarding a 1 or a 0 a bit collision will be signalled. To distinguish a 1 or 0-bit from a bit-collision, the CL RC632 uses the setting of *CollLevel*. If the amplitude of the half-bit with smaller amplitude is larger than defined by *CollLevel*, the CL RC632 indicates a bit-collision.

If a bit-collision is detected, the error flag *CollErr* is set. If a bit-collision is detected in a parity bit, the flag *ParityErr* is set indicating a parity error.

Independent from the detected collision the receiver continues receiving the incoming data stream. In case of a bit-collision, the decoder forwards 1 at the collision position.

Note: As an exception, if bit *ZeroAfterColl* is set, all bits received after the first bit-collision are forced to zero, regardless whether a bit-collision or an unequivocal state has been detected. This feature eases for the software to carry out the anti-collision procedure defined in ISO14443-A.

When the first bit collision in a frame is detected, the bit position of this collision is stored in the *CollPos Register*.

The collision position follows the table below:

Collision in Bit	Value of CollPos
SOF	0
LSbit of LSByte	1
...	...
MSbit of LSByte	8
LSbit of second Byte	9
...	...
MSbit of second Byte	16
LSbit of third Byte	17
...	...

Table 18-2: Returned Values for Bit Collision Positions

The parity bits are not counted in *CollPos*, since a bit-collision in a parity bit per definition succeeds a bit-collision in the data bits. If a collision is detected in the SOF a frame error is reported and no data is forwarded to the FIFO buffer. In this case the receiver continues to monitor the incoming signal and generates the correct notifications to the µ-Processor when the ending of the faulty input stream is detected. This helps the µ-Processor to determine the time when it is allowed next to send anything to the card.

Multiple Protocol Contactless Reader IC

CL RC632

18.4.2.4 Receiving Bit Oriented Frames

The receiver can handle byte streams with incomplete bytes, resulting in bit oriented frames. To support this, the following values may be used:

- *RxAlign* selects a bit offset for the first incoming byte, e.g. if *RxAlign* is set to 3, the first 5 bits received are forwarded to the FIFO buffer. Further bits are packed into bytes and forwarded. After reception, *RxAlign* is cleared automatically.
If *RxAlign* is set to zero, all incoming bits are packed into one byte.
- *RxLastBits* returns the number of bits valid in the last received byte, e.g. if *RxLastBits* evaluates to 5 at the end of the receiving command, the 5 least significant bits are valid.
RxLastBits evaluates to zero if the last byte is complete.

RxLastBits is valid only, if no frame error is indicated by the flag *FrameErr*. If *RxAlign* is set to a value other than zero and also *ParityEn* is active, the first parity bit is not checked but ignored.

18.4.2.5 Communication Errors

The following table shows which event causes the setting of error flags:

Cause	Bit, that is set
Received data did not start with a SOF pattern.	FramingErr
The CRC block is not equal the expected value.	CRCErr
The received data is shorter than the CRC block.	CRCErr
The parity bit is not equal the expected value (e. g. a bit collision occurs when a parity is expected)	ParErr
A collision is detected.	CollErr

Table 18-3: Communication Error Table

Multiple Protocol Contactless Reader IC**CL RC632****18.4.3 TRANSCEIVE COMMAND 1E_{HEX}**

Command	Code_{hex}	Action	Arguments and Data	Returned Data
Transceive	1E	Transmits data from FIFO buffer to the card and then activates automatically the receiver	Data Stream	Data Stream

The *Transceive-Command* first executes the *Transmit-Command* (see 18.4.1) and then automatically starts the *Receive-Command* (see 18.4.2). All data that shall be transmitted is forwarded via the FIFO buffer and all data received is returned via the FIFO buffer. The *Transceive-Command* can be started only by the µ-Processor.

Note: To adjust the timing relation between transmitting and receiving, the *RxWait Register* is used to define the time delay from the last bit transmitted until the receiver is activated. Furthermore, the *BitPhase Register* determines the phase-shift between the transmitter and the receiver clock.

18.4.4 STATES OF THE CARD COMMUNICATION

The actual state of the transmitter and receiver state machine can be fetched from *ModemState* in the *PrimaryStatus Register*.

The assignment of *ModemState* to the internal action is shown in the following table:

ModemState	Name of State	Description
000	Idle	Neither the transmitter nor the receiver is in operation, since none of them is started or the transmitter has not got input data
001	TxSOF	Transmitting the 'Start Of Frame' Pattern
010	TxData	Transmitting data from the FIFO buffer (or redundancy check bits)
011	TxEOS	Transmitting the 'End Of Frame' Pattern
100	GoToRx1	Intermediate state passed, when receiver starts
	GoToRx2	Intermediate state passed, when receiver finishes
101	PrepareRx	Waiting until the time period selected in the <i>RxWait Register</i> has expired
110	AwaitingRx	Receiver activated; Awaiting an input signal at pin Rx
111	Receiving	Receiving data

Table 18-4: Meaning of ModemState

Multiple Protocol Contactless Reader IC

CL RC632

18.4.5 STATE DIAGRAM FOR THE CARD COMMUNICATION

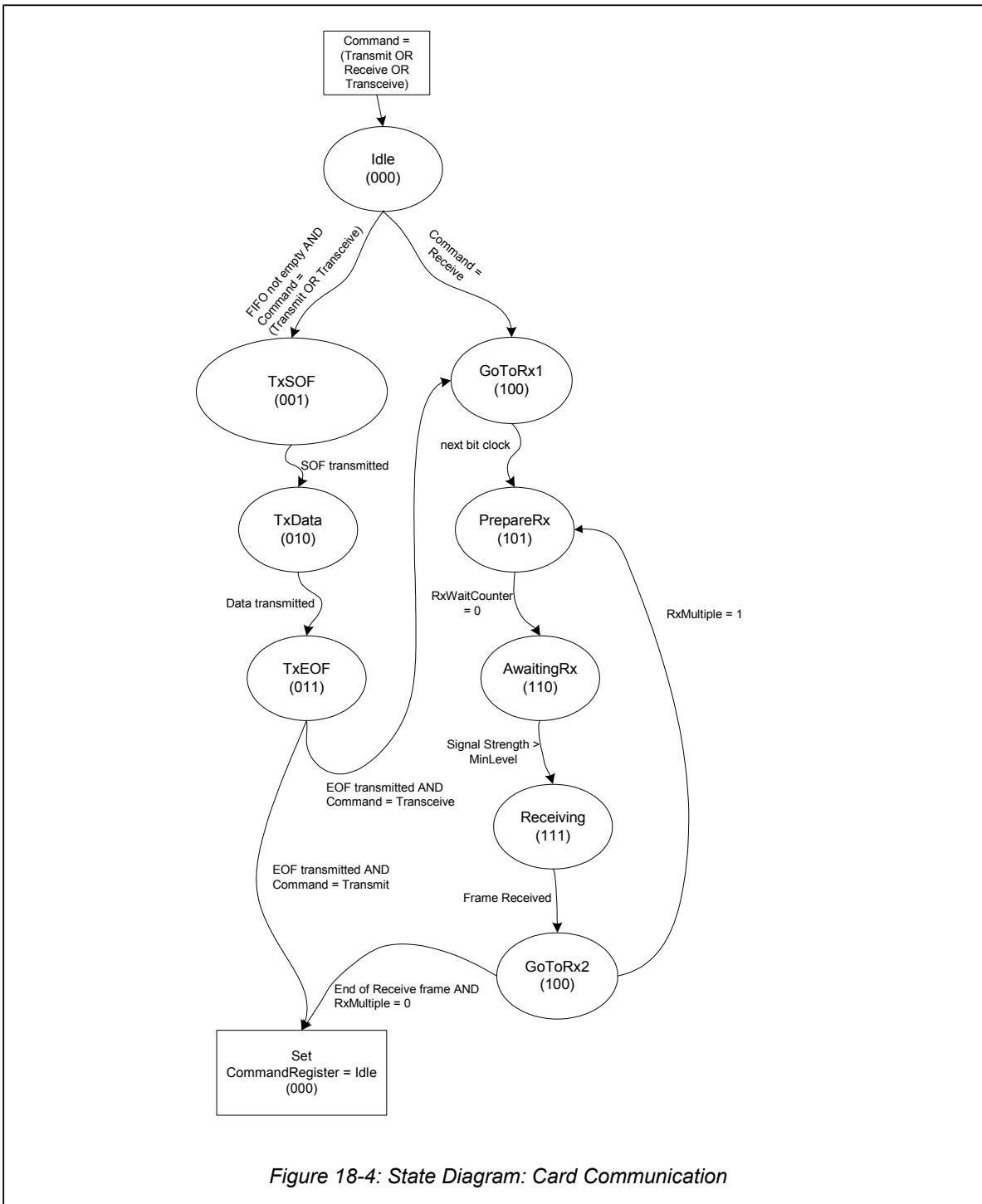


Figure 18-4: State Diagram: Card Communication

Multiple Protocol Contactless Reader IC

CL RC632

18.5 Commands for I•CODE1 and ISO15693 Label Communication

The CL RC632 is a fully ISO 14443 and ISO15693 and I•CODE1 compliant reader IC. Therefore, the command set of this IC allows more flexibility and more generalised commands compared to MIFARE® or I•CODE1 dedicated reader ICs. The following chapter describes the command set for card communication for I•CODE1 and ISO15693 related communication schemes in general.

18.5.1 TRANSMIT COMMAND 1A_{HEX}

Command	Code _{hex}	Action	Arguments and Data	Returned Data
Transmit	1A	Transmits data from FIFO buffer to the label	Data Stream	-

The *Transmit-Command* takes data from the FIFO buffer and forwards it to the transmitter. It does not return any data. The *Transmit-Command* can only be started by the µ-Processor.

18.5.1.1 Working with the Transmit Command

To transmit data one of the following sequences may be used:

1. All data, that shall be transmitted to the label is written to the FIFO while the *Idle-Command* is active. After that, the command code for the *Transmit-Command* is written to the *Command-Register*.
Note: This is possible for transmission of data with a length of up to 64 bytes.
2. The command code for the *Transmit-Command* is written to *Command-Register* first. Since no data is available in the FIFO, the command is only enabled but transmission is not triggered yet. Data transmission really starts with the first data byte written to the FIFO. To generate a continuous data stream on the RF-interface, the µ-Processor has to put the next data bytes to the FIFO in time.
Note: This allows transmission of data of any length but requires that data is available in the FIFO in time.
3. A part of the data, that shall be transmit to the label is written to the FIFO while the *Idle-Command* is active. After that, the command code for the *Transmit-Command* is written to the *Command-Register*. While the *Transmit-Command* is active, the µ-Processor may feed further data to the FIFO, causing the transmitter to append it to the transmitted data stream.
Note: This enables transmission of data of any length but requires that data is available in the FIFO in time.

When the transmitter requests the next data byte to keep the data stream on the RF-interface continuous but the FIFO buffer is empty, the *Transmit-Command* automatically terminates. This causes the internal state machine to change its state from *Transmit* to *Idle*.

If data transmission to the label is finished, the CL RC632 sets the flag *Tx/Rq* to signal it to the µ-Processor.

Remark: If the µ-Processor overwrites the transmit code in the *Command-Register* with the *Idle-Command* or any other command, transmission stops immediately with the next clock cycle. This may produce output signals that are not according to the standard ISO 15693 or the I•CODE1 protocol.

Multiple Protocol Contactless Reader IC

CL RC632

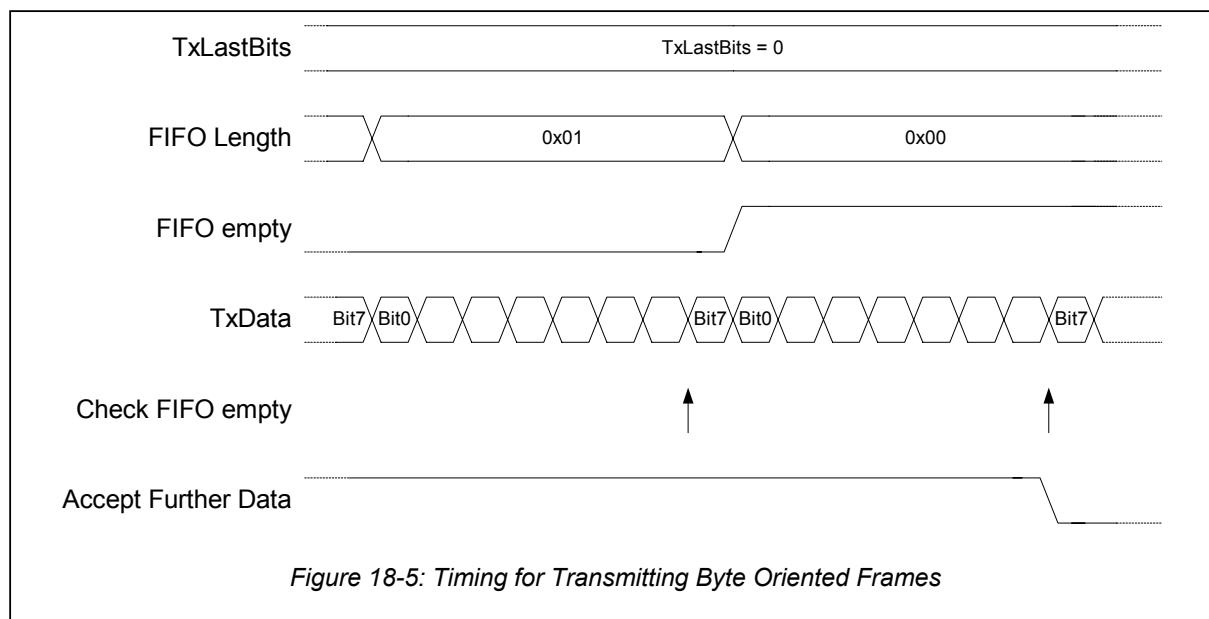
18.5.1.2 RF-Channel Redundancy and Framing

Each transmitted ISO 15693 frame consists of a SOF (start of frame) pattern, followed by the data stream and is closed by an EOF (end of frame) pattern. All I•CODE1 command frames consists of a START PULSE followed by the data stream. The I•CODE1 commands have a fix length and no EOF is needed. These different phases of the transmit sequence may be monitored by watching ModemState of *PrimaryStatus-Register* (see 18.4.4).

Depending on the setting of bit TxCRCEn in the *ChannelRedundancy-Register* a CRC is calculated and appended to the data stream. The CRC is calculated according the settings in the *ChannelRedundancy Register*.

18.5.1.3 Transmission of Frames with more than 64 Bytes

To generate frames with more than 64 bytes, the µ-Processor has to write data into the FIFO buffer while the *Transmit Command* is active. The state machine checks the FIFO status when it starts transmitting the last bit of the actual data stream (the check time is marked below with arrows).



As long as the internal signal 'Accept Further Data' is 1 further data may be loaded into the FIFO. The CL RC632 appends this data to the data stream transmitted via the RF-interface.

If the internal signal 'Accept Further Data' is 0 the transmission will terminate. All data written into the FIFO buffer after 'Accept Further Data' went 0 will not be transmitted anymore, but remain in the FIFO buffer.

Multiple Protocol Contactless Reader IC

CL RC632

18.5.2 RECEIVE COMMAND 16_{HEX}

Command	Code _{hex}	Action	Arguments and Data	Returned Data
Receive	16	Activates Receiver Circuitry	-	Data Stream

The *Receive-Command* activates the receiver circuitry. All data received from the RF interface is returned via the FIFO buffer. The *Receive-Command* can be started either by the µ-Processor or automatically during execution of the *Transceive-Command*.

Note: This command may be used for test purposes only, since there is no timing relation to the *Transmit-Command*.

18.5.2.1 Working with the Receive Command

After starting the *Receive Command* the internal state machine decrements the value set in the *RxWait-Register* with every bit-clock. From 3 down to 1 the analog receiver circuitry is prepared and activated. When the counter reaches 0, the receiver starts monitoring the incoming signal at the RF-interface. If the signal strength reaches a level higher than the value set in the *MinLevel-Register* it finally starts decoding. The decoder stops, if no more signal can be detected on the receiver input pin Rx. The decoder indicates termination of operation by setting bit *RxIRq*.

The different phases of the receive sequence may be monitored by watching ModemState of the *PrimaryStatus-Register* (see 18.4.4).

Note: Since the counter values from 3 to 0 are necessary to initialise the analog receiver circuitry the minimum value for *RxWait* is 3.

18.5.2.2 RF-Channel Redundancy and Framing

For ISO 15693 the decoder expects a SOF pattern at the beginning of each data stream. If a SOF is detected, it activates the serial to parallel converter and gathers the incoming data bits. For I•CODE1 the decoder do not expects a SOF pattern at the beginning of each data stream. It activates the serial to parallel converter with the first received bit of the data. Every completed byte is forwarded to the FIFO. If an EOF pattern (ISO15693) is detected or the signal strength falls below *MinLevel* set in the *RxThreshold Register*, the receiver and the decoder stop, the *Idle-Command* is entered and an appropriate response for the µ-Processor is generated (interrupt request activated, status flags set).

If bit *RxCRCEn* in the *ChannelRedundancy Register* is set a CRC block is expected. The CRC block may be one byte or two bytes according to bit *CRC8* in the *ChannelRedundancy Register*.

Remark: The received CRC block is not forwarded to the FIFO buffer if it is correct. This is realised by shifting the incoming data bytes through an internal buffer of either one or two bytes (depending on the defined CRC). The CRC block remains in this internal buffer. As a consequence all data bytes are available in the FIFO buffer one or two bytes delayed.

If the CRC fails all received bytes are forwarded to the FIFO buffer (including the faulty CRC itself).

Multiple Protocol Contactless Reader IC**CL RC632****18.5.2.3 Collision Detection**

If more than one label is within the RF-field during the label selection phase, they will respond simultaneously. The CL RC632 supports the algorithm defined in ISO 15693 as well as the I•CODE1 anti-collision algorithm to resolve data-collisions of label serial numbers by doing the so-called anti-collision procedure. The basis for this is the ability to detect bit-collisions.

Bit-collision detection is supported by the used bit-coding scheme, namely the Manchester-coding. If in the first and second half-bit of a bit a sub-carrier modulation is detected, instead of forwarding a 1 or a 0 a bit collision will be signalled. To distinguish a 1 or 0-bit from a bit-collision, the CL RC632 uses the setting of *CollLevel*. If the amplitude of the half-bit with smaller amplitude is larger than defined by *CollLevel*, the CL RC632 indicates a bit-collision.

If a bit-collision is detected, the error flag *CollErr* is set.

Independent from the detected collision the receiver continues receiving the incoming data stream. In case of a bit-collision, the decoder forwards 1 at the collision position.

Note: As an exception, if bit *ZeroAfterColl* is set, all bits received after the first bit-collision are forced to zero, regardless whether a bit-collision or an unequivocal state has been detected. This feature eases for the software to carry out the anti-collision procedure defined in ISO 15693.

When the first bit collision in a frame is detected, the bit position of this collision is stored in the *CollPos Register*.

The collision position follows the table below:

Collision in Bit	Value of CollPos
SOF	0
LSBit of LSByte	1
...	...
MSBit of LSByte	8
LSBit of second Byte	9
...	...
MSBit of second Byte	16
LSBit of third Byte	17
...	...

Table 18-5: Returned Values for Bit Collision Positions

If a collision is detected in the SOF a frame error is reported and no data is forwarded to the FIFO buffer. In this case the receiver continues to monitor the incoming signal and generates the correct notifications to the µ-Processor when the ending of the faulty input stream is detected. This helps the µ-Processor to determine the time when it is allowed next to send anything to the label.

Multiple Protocol Contactless Reader IC**CL RC632****18.5.2.4 Communication Errors**

The following table shows which event causes the setting of error flags:

Cause	Bit, that is set
Received data did not start with a SOF pattern.	FramingErr
The CRC block is not equal the expected value.	CRCErr
The received data is shorter than the CRC block.	CRCErr
A collision is detected.	CollErr

Table 18-6: Communication Error Table

18.5.3 TRANSCEIVE COMMAND 1E_{HEX}

Command	Code _{hex}	Action	Arguments and Data	Returned Data
Transceive	1E	Transmits data from FIFO buffer to the label and then activates automatically the receiver	Data Stream	Data Stream

The *Transceive-Command* first executes the *Transmit-Command* (see 18.4.1) and then automatically starts the *Receive-Command* (see 18.4.2). All data that shall be transmitted is forwarded via the FIFO buffer and all data received is returned via the FIFO buffer. The *Transceive-Command* can be started only by the µ-Processor.

Note: To adjust the timing relation between transmitting and receiving, the *RxWait Register* is used to define the time delay from the last bit transmitted until the receiver is activated. Furthermore, the *BitPhase Register* determines the phase-shift between the transmitter and the receiver clock.

18.5.4 STATES OF THE LABEL COMMUNICATION

The actual state of the transmitter and receiver state machine can be fetched from *ModemState* in the *PrimaryStatus Register*.

The assignment of *ModemState* to the internal action is shown in the following table:

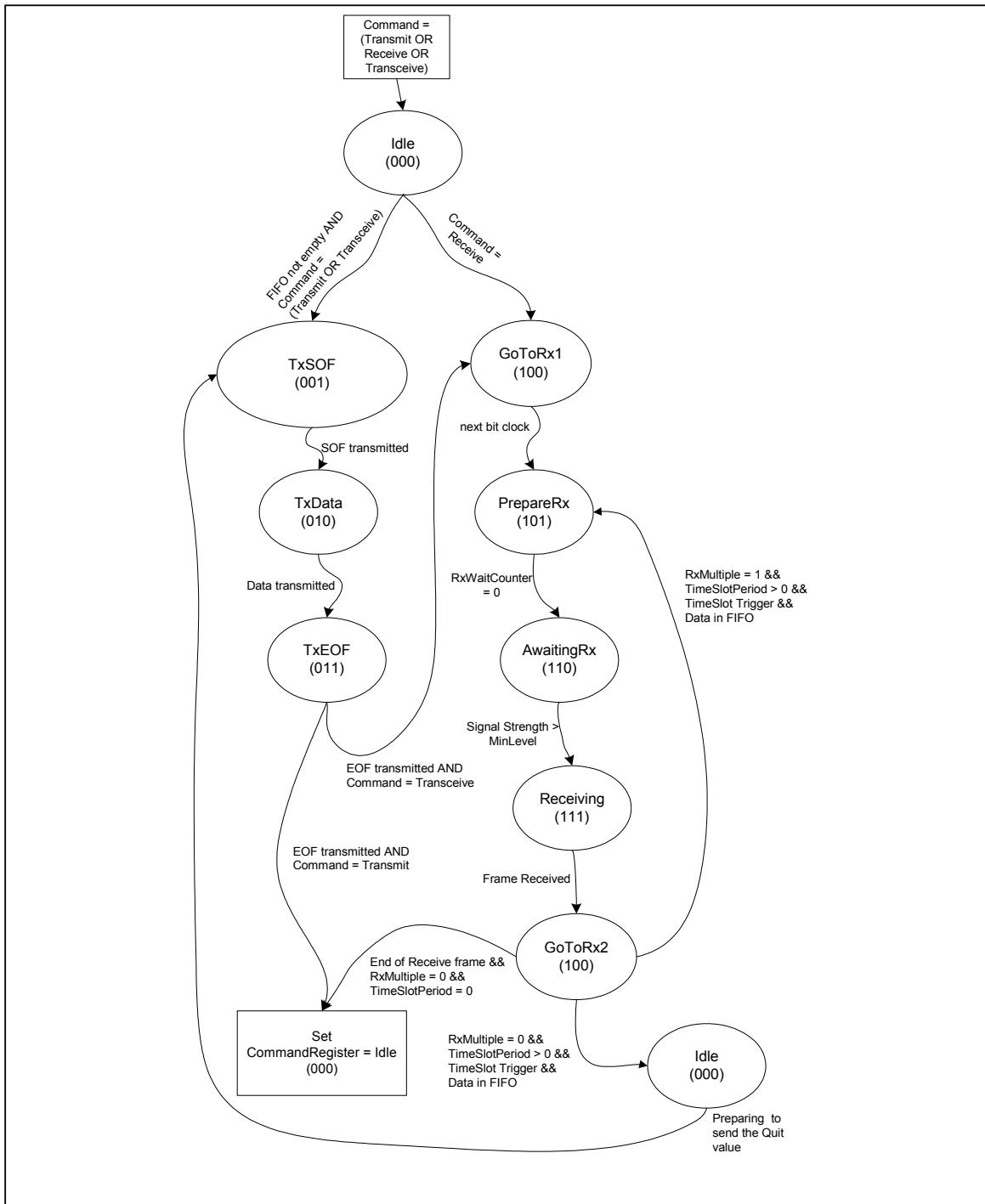
ModemState	Name of State	Description
000	Idle	Neither the transmitter nor the receiver is in operation, since none of them is started or the transmitter has not got input data
001	TxSOF	Transmitting the 'Start Of Frame' Pattern
010	TxDATA	Transmitting data from the FIFO buffer (or redundancy check bits)
011	TxEOP	Transmitting the 'End Of Frame' Pattern
100	GoToRx1	Intermediate state passed, when receiver starts
	GoToRx2	Intermediate state passed, when receiver finishes
101	PrepareRx	Waiting until the time period selected in the <i>RxWait Register</i> has expired
110	AwaitingRx	Receiver activated; Awaiting an input signal at pin Rx
111	Receiving	Receiving data

Table 18-7: Meaning of ModemState

18.5.5 STATE DIAGRAM FOR THE LABEL COMMUNICATION

Multiple Protocol Contactless Reader IC

CL RC632



Remark: I•CODE1 do not have a SOF and a EOF

Multiple Protocol Contactless Reader IC

CL RC632

18.6 Commands to Access the E²PROM

18.6.1 WRITEE2 COMMAND 01_{HEX}

18.6.1.1 Overview

Command	Code _{hex}	Action	Arguments and Data passed via FIFO	Returned Data via FIFO
WriteE2	01	Get data from FIFO buffer and write it to the E ² PROM	Start Address LSB Start Address MSB Data Byte Stream	-

The *WriteE2-Command* interprets the first two bytes in the FIFO buffer as E²PROM starting byte-address. Any further bytes are interpreted as data bytes and are programmed into the E²PROM, starting from the given E²PROM starting byte-address. This command does not return any data.

The *WriteE2-Command* can only be started by the µ-Processor. It will not stop automatically but has to be stopped explicitly by the µ-Processor by issuing the *Idle-Command*.

18.6.1.2 Programming Process

One byte up to 16 byte can be programmed into the EEPROM in one programming cycle. The time needed will be in any case about 5.8ms.

The state machine copies all data bytes prepared in the FIFO buffer to the E²PROM input buffer. The internal E²PROM input buffer is 16 byte long, which is equal the block size of the E²PROM. A programming cycle is started either if the last position of the E²PROM input buffer is written or if the last byte of the FIFO buffer has been fetched.

As long as there are unprocessed bytes in the FIFO buffer or the E²PROM programming cycle still is in progress, the flag *E2Ready* is 0. If all data from the FIFO buffer are programmed into the E²PROM, the flag *E2Ready* is set to 1. Together with the rising edge of *E2Ready* the interrupt request flag *TxIRq* indicates a 1. This may be used to generate an interrupt when programming of all data is finished.

After the *E2Ready* bit is set to 1, the *WriteE2-Command* may be stopped by the µ-Processor by issuing the *Idle-Command*.

Note: During the E2PROM programming indicated by *E2Ready* = 0 the WRITEE2 command cannot be stopped by any other command.

Multiple Protocol Contactless Reader IC

CL RC632

18.6.1.3 Timing Diagram

The following diagram shows programming of 5 bytes into the E²PROM:

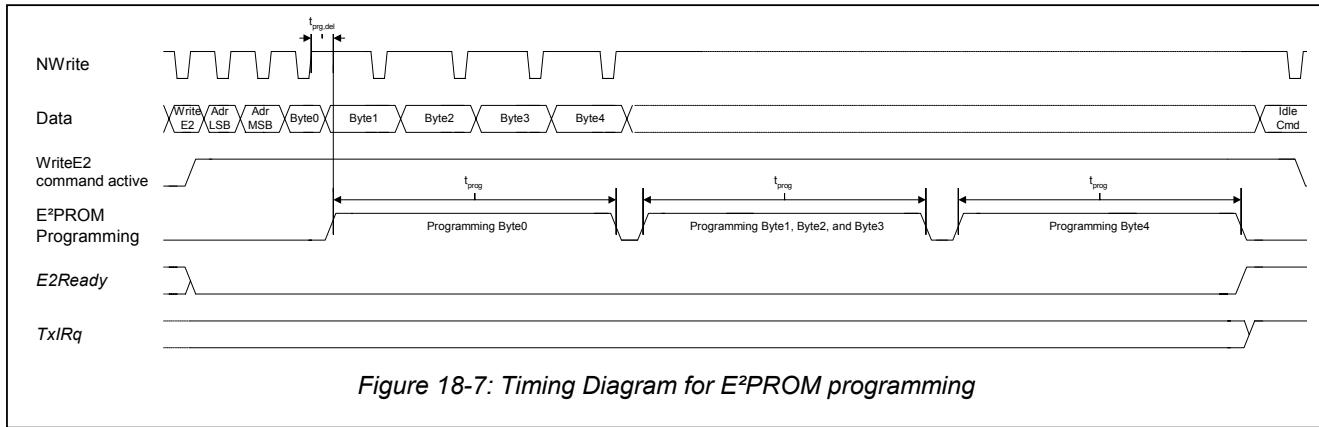


Figure 18-7: Timing Diagram for E²PROM programming

Explanation: It is assumed, that the CL RC632 finds and reads Byte 0 before the µ-Processor is able to write Byte 1 ($t_{prog,del} = 300$ ns). This causes the CL RC632 to start the programming cycle, which needs about $t_{prog} = 5.8$ ms. In the meantime the µ-Processor stores Byte 1 to Byte 4 to the FIFO buffer. Assuming, that the E²PROM starting byte-address is e.g. 16C_{hex} then Byte 0 is stored exactly there. The CL RC632 copies the following data bytes into the E²PROM input buffer. Copying Byte 3, it detects, that this data byte has to be programmed at the E²PROM byte-address 16F_{hex}. Since this is the end of the memory block, the CL RC632 automatically starts a programming cycle. In the next turn, Byte 4 will be programmed at the E²PROM byte-address 170_{hex}. Since this is the last data byte, the flags (E2Ready and TxIRq) that indicate the end of the E²PROM programming activity will be set.

Although all data has been programmed into the E2PROM, the CL RC632 stays in the *WriteE2-Command*. Writing further data to the FIFO would lead to further E²PROM programming, continuing at the E²PROM byte-address 171_{hex}. The command is stopped using the *Idle-Command*.

18.6.1.4 Error Flags for the WriteE2 Command

Programming is inhibited for the E²PROM blocks 0 (E²PROM's byte-address 00_{hex} to 0F_{hex}). Programming to these addresses sets the flag AccessErr. No programming cycle is started. Addresses above 1FF_{hex} are taken modulo 200_{hex} (for the E²PROM memory organisation, refer to chapter 6.).

Multiple Protocol Contactless Reader IC

CL RC632

18.6.2 READE2 COMMAND 03_{HEX}

18.6.2.1 Overview

Command	Code _{hex}	Action	Arguments	Returned Data
ReadE2	03	Reads data from E ² PROM and puts it to the FIFO buffer	Start Address LSB Start Address MSB Number of Data Bytes	Data Bytes

The *ReadE2-Command* interprets the first two bytes found in the FIFO buffer as E²PROM starting byte-address. The next byte specifies the number of data bytes that shall be returned. When all three argument-bytes are available in the FIFO buffer, the specified number of data bytes is copied from the E²PROM into the FIFO buffer, starting from the given E²PROM starting byte-address. The *ReadE2-Command* can be triggered only by the µ-Processor. It stops automatically when all data has been delivered.

18.6.2.2 Error Flags for the ReadE2 Command

Reading is inhibited for the E²PROM blocks 8_{hex} up to 1F_{hex} (key memory area). Reading from these addresses sets the flag *AccessErr* to 1. Addresses above 1FF_{hex} are taken modulo 200_{hex} (for the E²PROM memory organisation, refer to chapter 6).

18.7 Diverse Commands

18.7.1 LOADCONFIG COMMAND 07_{HEX}

18.7.1.1 Overview

Command	Code _{hex}	Action	Arguments and Data	Returned Data
LoadConfig	07	Reads data from E ² PROM and initialises the registers	Start Address LSB Start Address MSB	-

The *LoadConfig-Command* interprets the first two bytes found in the FIFO buffer as E²PROM starting byte-address. When the two argument-bytes are available in the FIFO buffer, 32 bytes from the E²PROM are copied into the CL RC632 control and configuration registers, starting at the given E²PROM starting byte-address. The *LoadConfig-Command* can only be started by the µ-Processor. It stops automatically when all relevant registers have been copied.

Multiple Protocol Contactless Reader IC

CL RC632

18.7.1.2 Register Assignment

The 32 bytes of E²PROM content, beginning with the E²PROM starting byte-address, is written to the CL RC632 register 10_{hex} up to register 2F_{hex} (for the E²PROM memory organisation see also 6).

Note: The procedure for the register assignment is the same as it is for the Start Up Initialisation (see 11.3). The difference is, that the E²PROM starting byte-address for the Start Up Initialisation is fixed to 10_{hex} (Block 1, Byte 0). With the *LoadConfig-Command* it can be chosen.

18.7.1.3 Relevant Error Flags for the LoadConfig-Command

Valid E²PROM starting byte-addresses are in the range from 10_{hex} up to 60_{hex}.

Copying from block 8_{hex} up to 1F_{hex} (keys) is inhibited. Reading from these addresses sets the flag *AccessErr* to 1. Addresses above 1FF_{hex} are taken modulo 200_{hex} (for the E²PROM memory organisation refer to chapter 6).

18.7.2 CALCCRC COMMAND 12_{HEX}

18.7.2.1 Overview

Command	Code _{hex}	Action	Arguments and Data	Returned Data
CalcCRC	12	Activates the CRC-Coprocessor	Data Byte-Stream	-

The *CalcCRC-Command* takes all data from the FIFO buffer as input bytes for the CRC-Coprocessor. All data stored in the FIFO buffer before the command is started will be processed. This command does not return any data via the FIFO buffer, but the content of the CRC-register can be read back via the *CRCResultLSB-register* and the *CRCResultMSB-register*. The *CalcCRC-Command* can only be started by the µ-Processor. It does not stop automatically but has to be stopped explicitly by the µ-Processor with the *Idle-Command*. If the FIFO buffer is empty, the *CalcCRC-Command* waits for further input from the FIFO buffer.

Multiple Protocol Contactless Reader IC

CL RC632

18.7.2.2 CRC-Coprocessor Settings

For the CRC-Coprocessor the following parameters may be configured:

Parameter	Value	Bit	Register
CRC Register Length	8 Bit or 16 Bit CRC	CRC8	ChannelRedundancy
CRC Algorithm	Algorithm according ISO14443-A or according ISO/IEC3309	CRC3309	ChannelRedundancy
CRC Preset Value	Any	CRCPresetLSB, CRCPresetMSB	CRCPresetLSB, CRCPresetMSB

Table 18-8: CRC-Coprocessor Parameters

The CRC polynomial for the 8-bit CRC is fixed to $x^8 + x^4 + x^3 + x^2 + 1$.

The CRC polynomial for the 16-bit CRC is fixed to $x^{16} + x^{12} + x^5 + 1$.

18.7.2.3 Status Flags of the CRC-Coprocessor

The status flag *CRCReady* indicates, that the CRC-Coprocessor has finished processing of all data bytes found in the FIFO buffer. With the *CRCReady* flag setting to 1, an interrupt is requested with *TxIRq* being set. This supports interrupt driven usage of the CRC-Coprocessor.

When *CRCReady* and *TxIRq* are set to 1, respectively, the content of the *CRCResultLSB*- and *CRCResultMSB*-register and the flag *CRCErr* is valid.

The *CRCResultLSB*- and *CRCResultMSB*-register hold the content of the CRC register, the *CRCErr* flag indicates CRC validity for the processed data.

18.8 Error Handling during Command Execution

If any error is detected during command execution, this is shown by setting the status flag *Err* in the *PrimaryStatus Register*. For information about the cause of the error, the µ-Processor may evaluate the status flags in the *ErrorFlag Register*.

Error Flag of the <i>ErrorFlag Register</i>	Related to Command
<i>KeyErr</i>	<i>LoadKeyE2, LoadKey</i>
<i>AccessError</i>	<i>WriteE2, ReadE2, LoadConfig</i>
<i>FIFOOverflow</i>	<i>No specific commands</i>
<i>CRCErr</i>	<i>Receive, Transceive, CalcCRC</i>
<i>FramingErr</i>	<i>Receive, Transceive</i>
<i>ParityErr</i>	<i>Receive, Transceive</i>
<i>CollErr</i>	<i>Receive, Transceive</i>

Table 18-9: Error Flags Overview

Multiple Protocol Contactless Reader IC

CL RC632

18.9 MIFARE® Classic Security Commands

18.9.1 LOADKEYE2 COMMAND 0B_{HEX}

18.9.1.1 Overview

Command	Code _{hex}	Action	Arguments and Data	Returned Data
LoadKeyE2	0B	Reads a key from the E ² PROM and puts it into the internal key buffer	Start Address LSB Start Address MSB	-

The *LoadKeyE2-Command* interprets the first two bytes found in the FIFO buffer as E²PROM starting byte-address. The E²PROM bytes starting from the given starting byte-address are interpreted as key, stored in the correct key format as described in chapter 6.4.1. When all two argument-bytes are available in the FIFO buffer, the command execution starts. The *LoadKeyE2-Command* can be started only by the µ-Processor. It stops automatically after having copied the key from the E²PROM into the key buffer.

18.9.1.2 Relevant Error Flags for the LoadKeyE2-Command

If the key format is not correct (see chapter 6.4.1) an undefined value is copied into the key buffer and the flag *KeyError* is set.

18.9.2 LOADKEY COMMAND 19_{HEX}

18.9.2.1 Overview

Command	Code _{hex}	Action	Arguments and Data	Returned Data
LoadKey	19	Reads a key from the FIFO buffer and puts it into the key buffer	Byte0 (LSB) Byte1 ... Byte10 Byte11 (MSB)	-

The *LoadKey-Command* interprets the first twelve bytes it finds in the FIFO buffer as key, stored in the correct key format as described in chapter 6.4.1.

When the twelve argument-bytes are available in the FIFO buffer they are checked and, if valid, are copied into the key buffer (see also 19.2).

The *LoadKey-Command* can only be started by the µ-Processor. It stops automatically after having copied the key from the FIFO buffer into the key buffer.

18.9.2.2 Relevant Error Flags for the LoadKey-Command

All bytes requested are copied from the FIFO buffer to the key buffer. If the key format is not correct (see chapter 6.4.1) an undefined value is copied into the key buffer and the flag *KeyError* is set.

Multiple Protocol Contactless Reader IC

CL RC632

18.9.3 AUTHENT1 COMMAND 0C_{HEX}

18.9.3.1 Overview

Command	Code _{hex}	Action	Arguments and Data	Returned Data
Authent1	0C	Performs the first part of the Crypto1 (MIFARE® Classic) card authentication	Card Auth-Command Card Block Address Card Serial Number LSB Card Serial Number Byte1 Card Serial Number Byte2 Card Serial Number MSB	-

The *Authent1-Command* is a special *Transceive-Command*: it takes six argument bytes which are sent to the card. The card's response is not forwarded to the µ-Processor, but is used to check the authenticity of the card and to prove authenticity of the CL RC632 to the card.

The *Authent1-Command* can be triggered only by the µ-Processor. The sequence of states for this command is the same as for the *Transceive-Command* (see 18.4.3).

18.9.4 AUTHENT2 COMMAND 14_{HEX}

18.9.4.1 Overview

Command	Code _{hex}	Action	Arguments and Data	Returned Data
Authent2	14	Performs the second part of the card authentication using the Crypto1 algorithm.	-	-

The *Authent2-Command* is a special *Transceive-Command*. It does not need any argument byte but all necessary data which has to be sent to the card is assembled by the CL RC632 itself. The card response is not forwarded to the µ-Processor, but is used to check the authenticity of the card and to prove authenticity of the CL RC632 to the card.

The *Authent2-Command* can only be started by the µ-Processor. The logical sequence for this command is the same as for the *Transceive-Command* (see 18.4.3).

18.9.4.2 Effect of the Authent2-Command

If the *Authent2-Command* was successful, authenticity of card and CL RC632 is proved. In this case, the control bit *Crypto1On* is set automatically. When bit *Crypto1On* is set, all further card communication is done encrypted, using the Crypto1 security algorithm. If the *Authent2-Command* fails, bit *Crypto1On* is cleared.

Note: The flag *Crypto1On* can not be set by the µ-Processor but only through a successfully performed *Authent2-Command*. The µ-Processor may clear the bit *Crypto1On* to continue with plain card communication.

Note: The *Authent2-Command* has to be executed immediately after a successful *Authent1-Command* (see 18.9.3). Furthermore, the keys stored in the key buffer and those on the card have to match.

Multiple Protocol Contactless Reader IC

CL RC632

19 MIFARE® CLASSIC AUTHENTICATION AND CRYPTO1

19.1 General

The security algorithm implemented in MIFARE® Classic products is called Crypto1. It is based on a proprietary stream cipher with a key length of 48 bits. To access data of a MIFARE® Classic card, the knowledge of the according key is necessary. For successful card authentication and subsequent access to the card's data stored in the EEPROM, the correct key has to be available in the CL RC632. After a card is selected as defined in ISO14443A the user may continue with the MIFARE® Classic protocol. In this case it is mandatory to perform a card authentication. The Crypto1 authentication is a 3-pass authentication. This procedure is done automatically with the execution of *Authent1-* (see 18.9.3) and the *Authent2-Commands* (see 18.9.4). During the card authentication procedure, the security algorithm is initialised. The communication with a MIFARE® Classic card following a successful authentication is encrypted.

19.2 Crypto1 Key Handling

During the authentication command the CL RC632 reads the key from the internal key buffer. The key is always taken from the key buffer. Therefore, the commands for Crypto1 authentication do not require addressing of a key. The user has to ensure, that the correct key is prepared in the key buffer before the card authentication is triggered.

The key buffer can be loaded

- from the E²PROM with the *LoadKeyE2-Command* (see 18.9.1)
- directly from the µ-Processor via the FIFO-Buffer with the *LoadKey-Command* (see 18.9.2)

This is shown in the following figure:

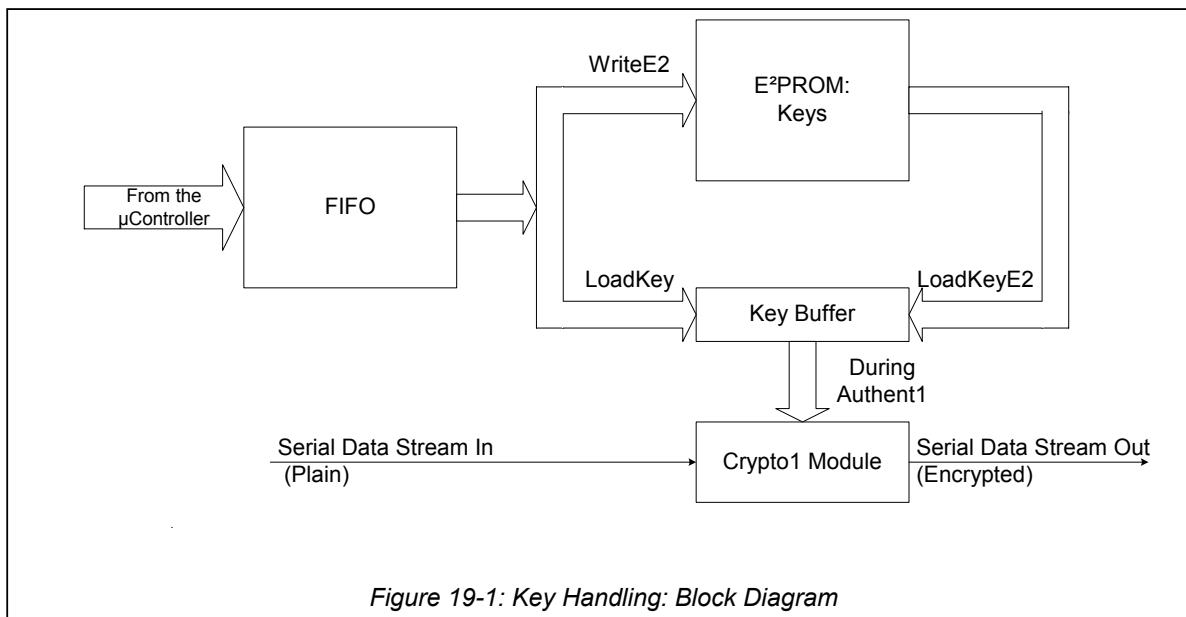


Figure 19-1: Key Handling: Block Diagram

Multiple Protocol Contactless Reader IC

CL RC632

19.3 Performing MIFARE® Classic Authentication

To enable authentication of MIFARE® Classic cards the Crypto1 security algorithm is implemented. To obtain valid authentication, the correct key has to be available in the key buffer of the CL RC632.

- ⇒ Step 1: Load the internal key buffer by means of the *LoadKeyE2-* (see 18.9.1) or the *LoadKey-Command* (see 18.9.2).
- ⇒ Step 2: Start the *Authent1-Command* (see 18.9.3). When finished, check the error flags to obtain the status of the command execution.
- ⇒ Step 3: Start the *Authent2-Command* (see 18.9.4). When finished, check the error flags and bit *Crypto1On* to obtain the status of the command execution.

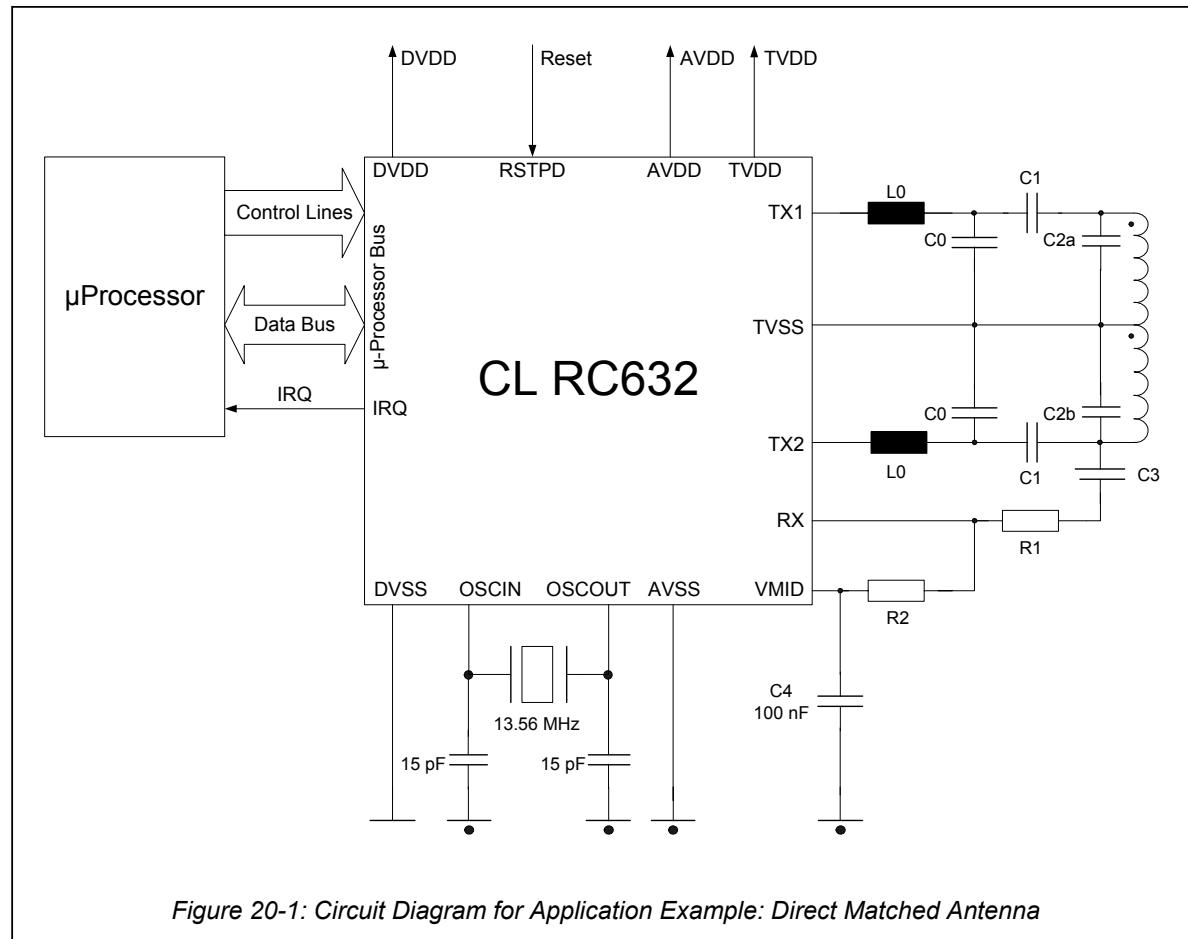
Multiple Protocol Contactless Reader IC

CL RC632

20 TYPICAL APPLICATION

20.1 Circuit Diagram

The figure below shows a typical application, where the antenna is directly connected to the CL RC632:



Multiple Protocol Contactless Reader IC

CL RC632

20.2 Circuit Description

The matching circuit consists of an EMC low pass filter (L0 and C0), a matching circuitry (C1 and C2), and a receiving circuit (R1, R2, C3 and C4), and the antenna itself.

For more detailed information about designing and tuning an antenna please refer to the Application Note '*MIFARE® and I CODE MICORE reader IC family; Directly Matched Antenna Design*' and '*MIFARE® (14443A) 13,56 MHz RFID Proximity Antennas*'.

20.2.1 EMC LOW PASS FILTER

The MIFARE® system operates at a frequency of 13.56 MHz. This frequency is generated by a quartz oscillator to clock the CL RC632 and is also the basis for driving the antenna with the 13.56 MHz energy carrier. This will not only cause emitted power at 13.56 MHz but will also emit power at higher harmonics. The international EMC regulations define the amplitude of the emitted power in a broad frequency range. Thus, an appropriate filtering of the output signal is necessary to fulfil these regulations.

A multi-layer board it is recommended to implement a low pass filter as shown in the circuit above. The low pass filter consists of the components L0 and C0. The recommended values are given in the above mentioned application notes.

Note: To achieve best performance all components shall have at least the quality of the recommended ones.

Note: The layout has a major influence on the overall performance of the filter.

20.2.2 ANTENNA MATCHING

Due to the impedance transformation of the given low pass filter, the antenna coil has to be matched to a certain impedance. The matching elements C1 and C2 can be estimated and have to be fine tuned depending on the design of the antenna coil.

The correct impedance matching is important to provide the optimum performance. The overall Quality factor has to be considered to guarantee a proper ISO14443 communication scheme. Environmental influences have to be considered as well as common EMC design rules.

For details refer to the above mentioned application notes.

Note: Do not exceed the current limits I_{TVDD} , otherwise the chip might be destroyed.

Note: The overall 13.56MHz RFID proximity antenna design with the CL RC632 chip is straight forward and doesn't require a special RF-know how. However, all relevant parameters have to be considered to guarantee an overall optimum performance together with international EMC compliance.

Multiple Protocol Contactless Reader IC**CL RC632****20.2.3 RECEIVING CIRCUIT**

The internal receiving concept of the CL RC632 makes use of both side-bands of the sub-carrier load modulation of the card response. No external filtering is required.

It is recommended to use the internally generated VMID potential as the input potential of pin RX. This DC voltage level of VMID has to be coupled to the Rx-pin via R2. To provide a stable DC reference voltage a capacitance C4 has to be connected between VMID and ground.

Considering the (AC) voltage limits at the Rx-pin the AC voltage divider of R1 + C3 and R2 has to be designed. Depending on the antenna coil design and the impedance matching the voltage at the antenna coil varies from antenna design to antenna design. Therefore the recommended way to design the receiving circuit is to use the given values for R1, R2, and C3 from the above mentioned application note, and adjust the voltage at the Rx-pin by varying R1 within the given limits.

Note: R2 is AC-wise connected to ground (via C4).

20.2.4 ANTENNA COIL

The precise calculation of the antenna coils' inductance is not practicable but the inductance can be **estimated** using the following formula. We recommend designing an antenna either with a circular or rectangular shape.

$$L_1 \text{ [nH]} = 2 \cdot I_1 \text{ [cm]} \cdot \left(\ln\left(\frac{I_1}{D_1}\right) - K \right) N_1^{1.8}$$

I₁.....Length of one turn of the conductor loop

D₁.....Diameter of the wire or width of the PCB conductor respectively

K.....Antenna Shape Factor (K = 1,07 for circular antennas and K = 1,47 for square antennas)

N₁.....Number of turns

InNatural logarithm function

The actual values of the **antenna inductance, resistance, and capacitance at 13.56 MHz** depend on various parameters like:

- antenna construction (Type of PCB)
- thickness of conductor
- distance between the windings
- shielding layer
- metal or ferrite in the near environment

Therefore a measurement of those parameters under real life conditions, or at least a rough measurement and a tuning procedure is recommended to guarantee the optimum performance. For details refer to the above mentioned application notes.

Multiple Protocol Contactless Reader IC

CL RC632

21 TEST SIGNALS

21.1 General

The CL RC632 allows different kind of signal measurements. These measurements can be used to check the internally generated and received signals using the possibilities of the serial signal switch as described in chapter 15.

Furthermore, with the CL RC632 the user may select internal analogue signals to measure them at pin AUX and internal digital signals to observe them on pin MFOUT by register selections. These measurements can be helpful during the design-in phase to optimise the receiver's behaviour or for test purpose.

21.2 Measurements Using the Serial Signal Switch

Using the serial signal switch at pin MFOUT the user may observe data send to the card or data received from the card. The following tables give an overview of the different signals available.

SignalToMFOUT	MFOUTSelect	Signal routed to MFOUT pin
0	0	LOW
0	1	HIGH
0	2	Envelope
0	3	Transmit NRZ
0	4	Manchester with Subcarrier
0	5	Manchester
0	6	RFU
0	7	RFU
1	X	Digital Test signal

Table 21-1 Signal routed to MFOUT pin

Note: The routing of the Manchester and the Manchester with Subcarrier signal to the MFOUT is only possible at 106 kbaud according to ISO14443A.

Multiple Protocol Contactless Reader IC

CL RC632

21.2.1 TX-CONTROL

The following plot shows as an example an ISO14443 A related communication.

The signal measured at MFOUT using the serial signal switch to control the data sent to the card .Setting the flag *MFOUTSelect* to 3 data sent to the card is shown NRZ coded. *MFOUTSelect* set to 2 shows the Miller coded signal.

The RFOut signal is measured directly on the antenna showing the pulse shape of the RF signal. For detail information concerning the pulse of the RF signal please refer to the application note '*MIFARE® Design of MF RC 500 Matching Circuits and Antennas*'

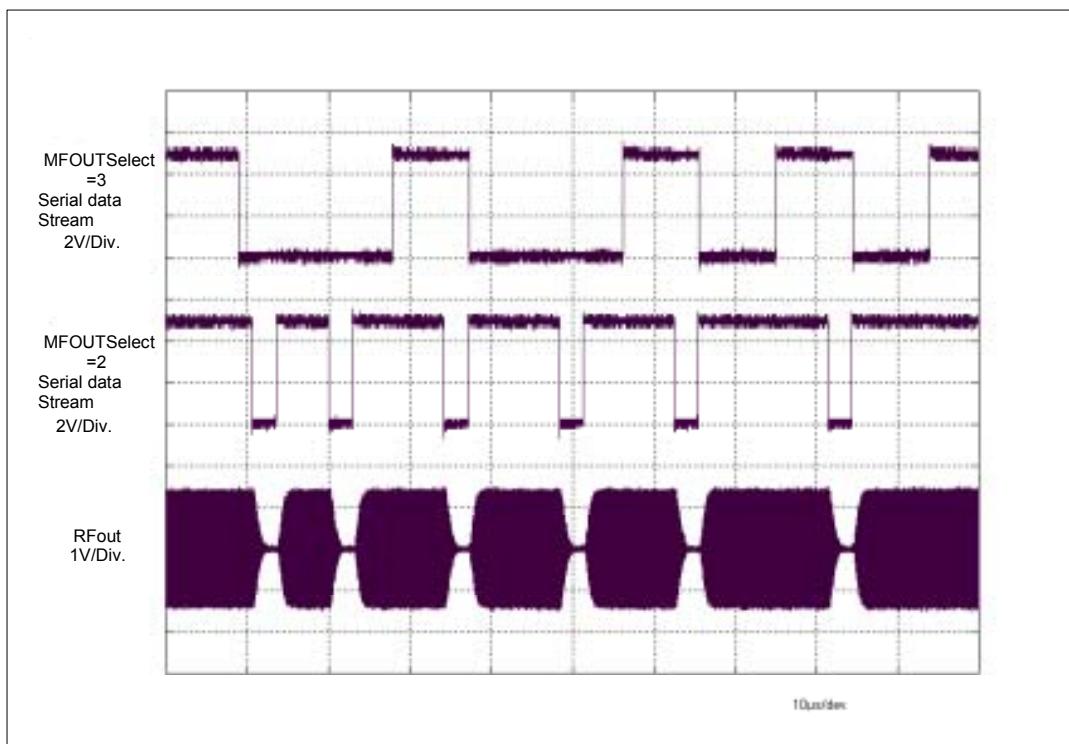


Figure 21 TX Control Signals

Multiple Protocol Contactless Reader IC

CL RC632

21.2.2 RX-CONTROL

The following plot shows as an example an ISO14443 A related communication.

The following plot shows the beginning of a cards answer to a request signal. The signal RF shows the RF voltage measured directly on the antenna so that the cards load modulation is visible. *MFOUTSelect* set to 4 shows the Manchester decoded signal with subcarrier. *MFOUTSelect* set to 5 shows the Manchester decoded signal.

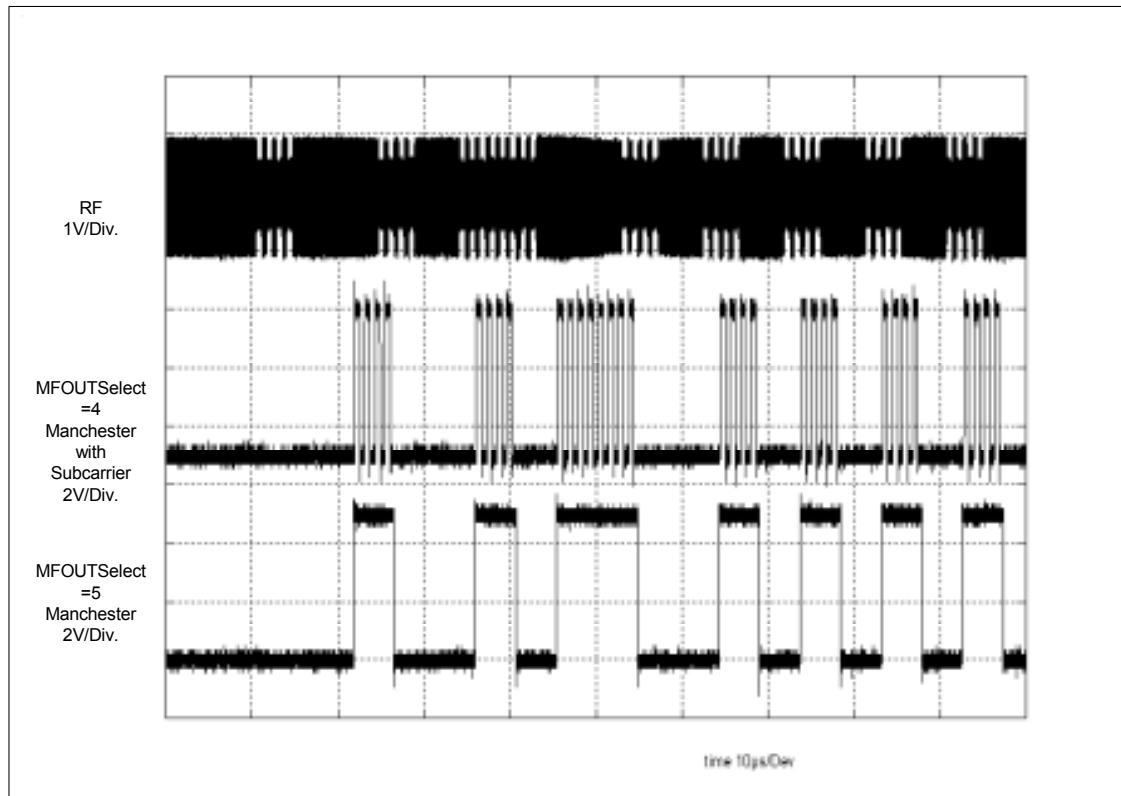


Figure 22 RX Control Signals

Multiple Protocol Contactless Reader IC

CL RC632**21.3 Analog Test-Signals**

The analog test signals may be routed to pin AUX by selecting them with the register bits *TestAnaOutSel*.

Value	Signal Name	Description
0	V_{mid}	Voltage at internal node V_{mid}
1	$V_{bandgap}$	Internal reference voltage generated by the band gap.
2	$V_{RxFollI}$	Output signal from the demodulator using the I-clock.
3	$V_{RxFollQ}$	Output signal from the demodulator using the Q-clock.
4	V_{RxAmpI}	I-channel subcarrier signal amplified and filtered.
5	V_{RxAmpQ}	Q-channel subcarrier signal amplified and filtered.
6	V_{CorrNI}	Output signal of N-channel correlator fed by the I-channel subcarrier signal.
7	V_{CorrNQ}	Output signal of N-channel correlator fed by the Q-channel subcarrier signal.
8	V_{CorrDI}	Output signal of D-channel correlator fed by the I-channel subcarrier signal.
9	V_{CorrDQ}	Output signal of D-channel correlator fed by the Q-channel subcarrier signal.
A	V_{EvalL}	Evaluation signal from the left half bit.
B	V_{EvalR}	Evaluation signal from the right half bit.
C	V_{Temp}	Temperature voltage derived from band gap.
D	rfu	Reserved for future use
E	rfu	Reserved for future use
F	rfu	Reserved for future use

Table 21-2: Analog Test Signal Selection

Multiple Protocol Contactless Reader IC**CL RC632****21.4 Digital Test-Signals**

Digital test signals may be routed to pin MFOUT by setting bit *SignalToMFOUT* to 1. A digital test signal may be selected via the register bits *TestDigiSignalSel* in Register *TestDigiSelect*.

The signals selected by a certain *TestDigiSignalSel* setting is shown in the table below:

TestDigiSignalSel	Signal Name	Description
F4 _{hex}	s_data	Data received from the card.
E4 _{hex}	s_valid	Shows with 1, that the signals s_data and s_coll are valid.
D4 _{hex}	s_coll	Shows with 1, that a collision has been detected in the current bit.
C4 _{hex}	s_clock	Internal serial clock: during transmission, this is the coder-clock and during reception this is the receiver clock.
B5 _{hex}	rd_sync	Internal synchronised read signal (derived from the parallel µ-Processor interface).
A5 _{hex}	wr_sync	Internal synchronised write signal (derived from the parallel µ-Processor interface).
96 _{hex}	int_clock	Internal 13.56 MHz clock.
83 _{hex}	BPSK_out	BPSK signal output
E2 _{hex}	BPSK_sig	BPSK signal's amplitude detected
00 _{hex}	no test signal	output as defined by <i>MFOUTSelect</i> are routed to pin MFOUT.

Table 21-3: Digital Test Signal Selection

If no test signals are used, the value for the *TestDigiSelect-Register* shall be 00_{hex}.

Note: All other values of *TestDigiSignalSel* are for production test purposes only.

Multiple Protocol Contactless Reader IC

CL RC632

21.5 Examples of ISO14443A Analog- and Digital Test Signals

Fig. 22 shows a MIFARE® Classic Card's answer to a request command using the Qclock receiving path.

RX –Reference is given to show the Manchester modulated signal at the RX pin. This signal is demodulated and amplified in the receiver circuitry VRxAmpQ shows the amplified side band signal having used the Q-Clock for demodulation. The signals VCorrDQ and VCorrNQ generated in the correlation circuitry are evaluated and digitised in the evaluation and digitizer circuitry. VEvalR and VEvalL show the evaluation signal of the right and left half bit. Finally, the digital test-signal S_data shows the received data which is send to the internal digital circuit and S_valid indicates that the received data stream is valid.

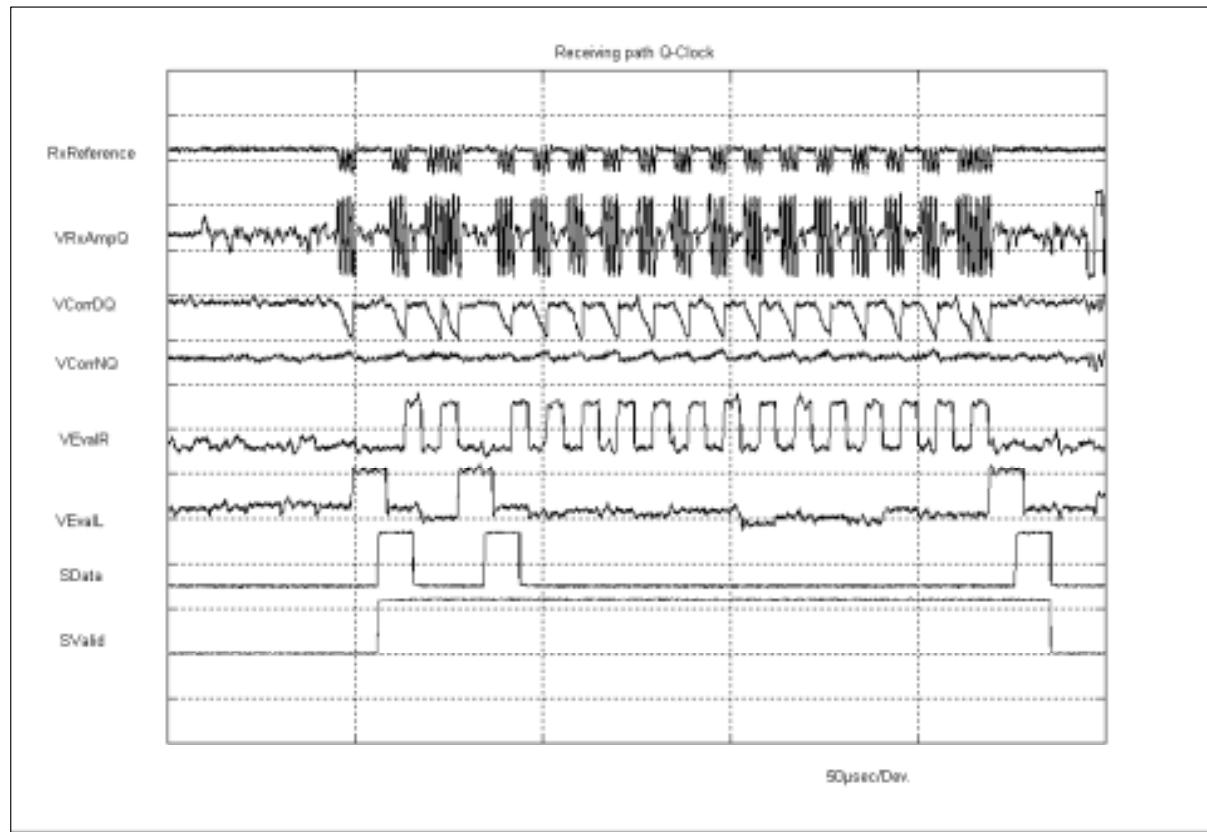


Figure 23. Receiving path Q-Clock

Multiple Protocol Contactless Reader IC

CL RC632

21.6 Examples of ICODE1 Analog- and Digital Test Signals

Fig. 17 shows the answer of an ICODE1 Label IC to a unselected read command using the Qclock receiving path.

RX –Reference is given to show the Manchester modulated signal at the RX pin. This signal is demodulated and amplified in the receiver circuitry VRXAmpQ shows the amplified side band signal having used the Q-Clock for demodulation. The signals VCorrDQ and VCorrNQ generated in the correlation circuitry are evaluated and digitised in the evaluation and digitizer circuitry. VEvalR and VEvalL show the evaluation signal of the right and left half bit. Finally, the digital test-signal S_data shows the received data which is send to the internal digital circuit and S_valid indicates that the received data stream is valid.

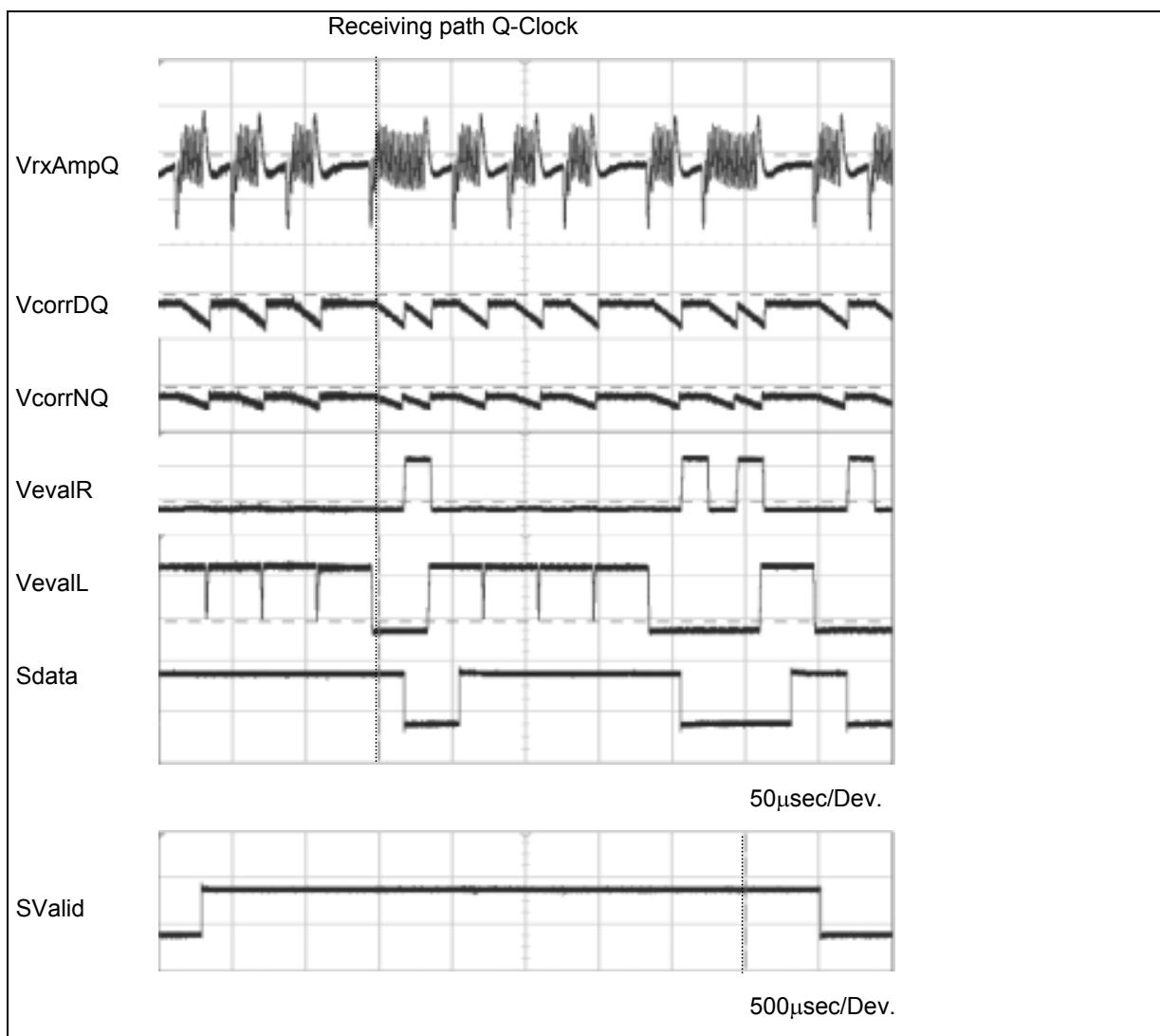


Figure 24. Receiving path Q-Clock

Multiple Protocol Contactless Reader IC

CL RC632

22 ELECTRICAL CHARACTERISTICS

22.1 Absolute Maximum Ratings

SYMBOL	PARAMETER	MIN	MAX	UNIT
T _{amb,abs}	Ambient or Storage Temperature Range	-40	+150	°C
DVDD				
AVDD	DC Supply Voltages	-0.5	6	V
TVDD				
V _{in,abs}	Absolute voltage on any digital pin to DVSS	-0.5	DVDD + 0.5	V
V _{RX,abs}	Absolute voltage on RX pin to AVSS	-0.5	AVDD + 0.5	V

Table 22-1: Absolute Maximum Ratings

22.2 Operating Condition Range

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
T _{amb}	Ambient Temperature	-	-25	+25	+85	°C
DVDD	Digital Supply Voltage	DVSS = AVSS = TVSS = 0V	3.0	3.3	3.6	V
			4.5	5.0	5.5	V
AVDD	Analog Supply Voltage	DVSS = AVSS = TVSS = 0V	4.5	5.0	5.5	V
TVDD	Transmitter Supply Voltage	DVSS = AVSS = TVSS = 0V	3.0	5.0	5.5	V

Table 22-2: Operating Condition Range

22.3 Current Consumption

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
I _{DVDD}	Digital Supply Current	Idle Command		6	9	mA
		Stand By Mode		3	5	mA
		Soft Power Down Mode		800	1000	μA
		Hard Power Down Mode		1	10	μA
I _{AVDD}	Analog Supply Current	Idle Command, Receiver On		25	40	mA
		Idle Command, Receiver Off		8	12	mA
		Stand By Mode		6.5	9	mA
		Soft Power Down Mode		1	10	μA
		Hard Power Down Mode		1	10	μA
I _{TVDD}	Transmitter Supply Current	Continuous Wave			150	mA
		TX1 and TX2 unconnected TX1RFEn, TX2RFEn = 1		4.5	6	mA
		TX1 and TX2 unconnected TX1RFEn, TX2RFEn = 0		65	130	μA

Table 22-3: Current Consumption

Multiple Protocol Contactless Reader IC

CL RC632

22.4 Pin Characteristics**22.4.1 INPUT PIN CHARACTERISTICS**

Pins D0 to D7, A0, and A1 have TTL input characteristics and behave as defined in the following table.

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
I_{Leak}	Input Leakage Current		-1.0	+1.0	μA
V_T	Threshold	CMOS: DVDD < 3.6 V	0.35 DVDD	0.65 DVDD	V
		TTL: 4.5 < DVDD	0.8	2.0	V

Table 22-4: Standard Input Pin Characteristics

The digital input pins NCS, NWR, NRD, ALE, A2, and MFIN have Schmitt-Trigger characteristics, and behave as defined in the following table.

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
I_{Leak}	Input Leakage Current		-1.0	+1.0	μA
V_{T+}	Positive-Going Threshold	TTL: 4.5 < DVDD	1.4	2.0	V
		CMOS: DVDD < 3.6 V	0.65 DVDD	0.75 DVDD	V
V_{T-}	Negative-Going Threshold	TTL: 4.5 < DVDD	0.8	1.3	V
		CMOS: DVDD < 3.6 V	0.25 DVDD	0.4 DVDD	V

Table 22-5: Schmitt-Trigger Input Pin Characteristics

Pin RSTPD has Schmitt-Trigger CMOS characteristics. In addition, it is internally filtered with an RC-low-pass filter, which causes a relevant propagation delay for the reset signal:

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
I_{Leak}	Input Leakage Current		-1.0	+1.0	μA
V_{T+}	Positive-Going Threshold	CMOS: DVDD < 3.6 V	0.65 DVDD	0.75 DVDD	V
V_{T-}	Negative-Going Threshold	CMOS: DVDD < 3.6 V	0.25 DVDD	0.4 DVDD	V
$t_{\text{RSTPD},p}$	Propagation Delay			20	μs

Table 22-6: RSTPD Input Pin Characteristics

The analog input pin RX has the following input capacitance:

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
C_{RX}	Input Capacitance			15	pF

Table 22-7: RX Input Capacitance

The analog input pin RX has the following input voltage range:

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
$V_{\text{IN,RX}}$	Dynamical Voltage input range	AVDD=5V, T=25°C	1,1V	4,4	V

Table 22-8: RX Input voltage range

Multiple Protocol Contactless Reader IC**CL RC632****22.4.2 DIGITAL OUTPUT PIN CHARACTERISTICS**

Pins D0 to D7, MFOUT and IRQ have CMOS output characteristics and behave as defined in the following table.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	Output Voltage HIGH	DVDD = 5 V, $I_{OH} = -1 \text{ mA}$	2.4	4.9		V
		DVDD = 5 V, $I_{OH} = -10 \text{ mA}$	2.4	4.2		V
V_{OL}	Output Voltage LOW	DVDD = 5 V, $I_{OL} = 1 \text{ mA}$		25	400	mV
		DVDD = 5 V, $I_{OL} = 10 \text{ mA}$		250	400	mV
I_O	Output Current source or sink	DVDD = 5 V			10	mA

Table 22-8: Digital Output Pin Characteristics

Note: IRQ pin may also be configured as open collector. In that case the values for V_{OH} do not apply.

22.4.3 ANTENNA DRIVER OUTPUT PIN CHARACTERISTICS

The source conductance of the antenna driver pins TX1 and TX2 for driving the HIGH level can be configured via *GsCfgCW* in the *CwConductance Register*, while their source conductance for driving the LOW level is constant.

For the default configuration, the output characteristic is specified below:

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	Output Voltage HIGH	TVDD = 5.0 V, $I_{OL} = 20 \text{ mA}$		4.97		V
		TVDD = 5.0 V, $I_{OL} = 100 \text{ mA}$		4.85		V
V_{OL}	Output Voltage LOW	TVDD = 5.0 V, $I_{OL} = 20 \text{ mA}$		30		mV
		TVDD = 5.0 V, $I_{OL} = 100 \text{ mA}$		150		mV
I_{TX}	Transmitter Output Current	Continuous Wave			200	mA_{peak}

Table 22-9: Antenna Driver Output Pin Characteristics

Multiple Protocol Contactless Reader IC

CL RC632

22.5 AC Electrical Characteristics

22.5.1 AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' for time. The other characters indicate the name of a signal or the logic state of that signal (depending on position):

Designation:	Signal:	Designation:	Logic Level:
A	address	H	HIGH
D	data	L	LOW
W	NWR or nWait	Z	high impedance
R	NRD or R/NW or nWrite	X	any level or data
L	ALE or AS	V	any valid signal or data
C	NCS	N	NSS
S	NDS or nDStrb and nAStrb, SCK		

Example: t_{AVLL} = time for address valid to ALE low

Multiple Protocol Contactless Reader IC

CL RC632

22.5.2 AC OPERATING SPECIFICATION

22.5.2.1 Bus Timing for Separated Read/Write Strobe

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{LHLL}	ALE pulse width	20		ns
t_{AVLL}	Multiplexed Address Bus valid to ALE low (Address Set Up Time)	15		ns
t_{LLAX}	Multiplexed Address Bus valid after ALE low (Address Hold Time)	8		ns
t_{LLWL}	ALE low to NWR, NRD low	15		ns
t_{CLWL}	NCS low to NRD, NWR low	0		ns
t_{WHCH}	NRD, NWR high to NCS high	0		ns
t_{RLDV}	NRD low to DATA valid		65	ns
t_{RHDZ}	NRD high to DATA high impedance		20	ns
t_{WLDV}	NWR low to DATA valid		35	ns
t_{WHDX}	DATA hold after NWR high (Data Hold Time)	8		ns
t_{WLWH}	NRD, NWR pulse width	65		ns
t_{AVWL}	Separated Address Bus valid to NRD, NWR low (Set Up Time)	30		ns
t_{WHAX}	Separated Address Bus valid after NWR high (Hold Time)	8		ns
t_{WHWL}	period between sequenced read / write accesses	150		ns

Table 22-10: Timing Specification for Separated Read/Write Strobe

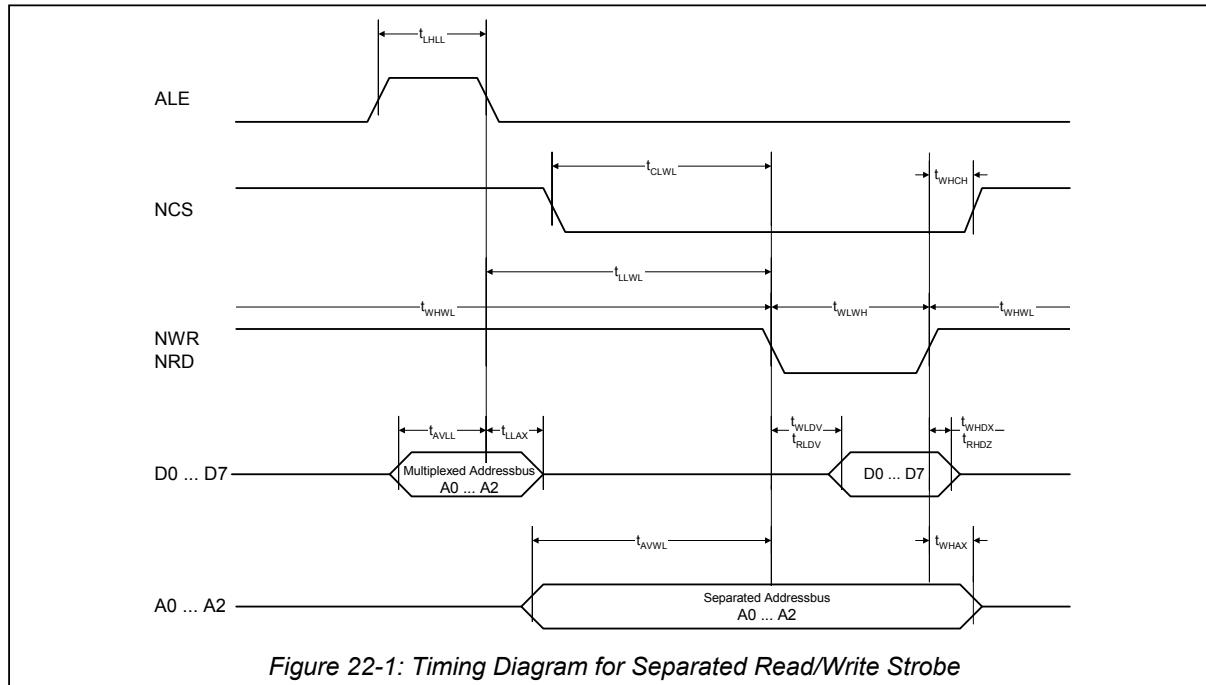


Figure 22-1: Timing Diagram for Separated Read/Write Strobe

Note: For separated address and data bus the signal ALE is not relevant and the multiplexed addresses on the data bus don't care.

For the multiplexed address and data bus the address lines A0 to A2 have to be connected as described in 4.3.

Multiple Protocol Contactless Reader IC

CL RC632

22.5.2.2 Bus Timing for Common Read/Write Strobe

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{LHLL}	AS pulse width	20		ns
t_{AVLL}	Multiplexed Address Bus valid to AS low (Address Set Up Time)	15		ns
t_{LLAX}	Multiplexed Address Bus valid after AS low (Address Hold Time)	8		ns
t_{LLSL}	AS low to NDS low	15		ns
t_{CLSL}	NCS low to NDS low	0		ns
t_{SHCH}	NDS high to NCS high	0		ns
$t_{SLDV,R}$	NDS low to DATA valid (for read cycle)		65	ns
t_{SHDZ}	NDS low to DATA high impedance (read cycle)		20	ns
$t_{SLDW,W}$	NDS low to DATA valid (for write cycle)		35	ns
t_{SHDX}	DATA hold after NDS high (write cycle, Hold Time)	8		ns
t_{SHRX}	R/NW hold after NDS high	8		ns
t_{SLSH}	NDS pulse width	65		ns
t_{AVSL}	Separated Address Bus valid to NDS low (Hold Time)	30		ns
t_{SHAX}	Separated Address Bus valid after NDS high (Set Up Time)	8		ns
t_{SHSL}	period between sequenced read/write accesses	150		ns
t_{RVSL}	R/NW valid to NDS low	8		ns

Table 22-11: Timing Specification for Common Read/Write Strobe

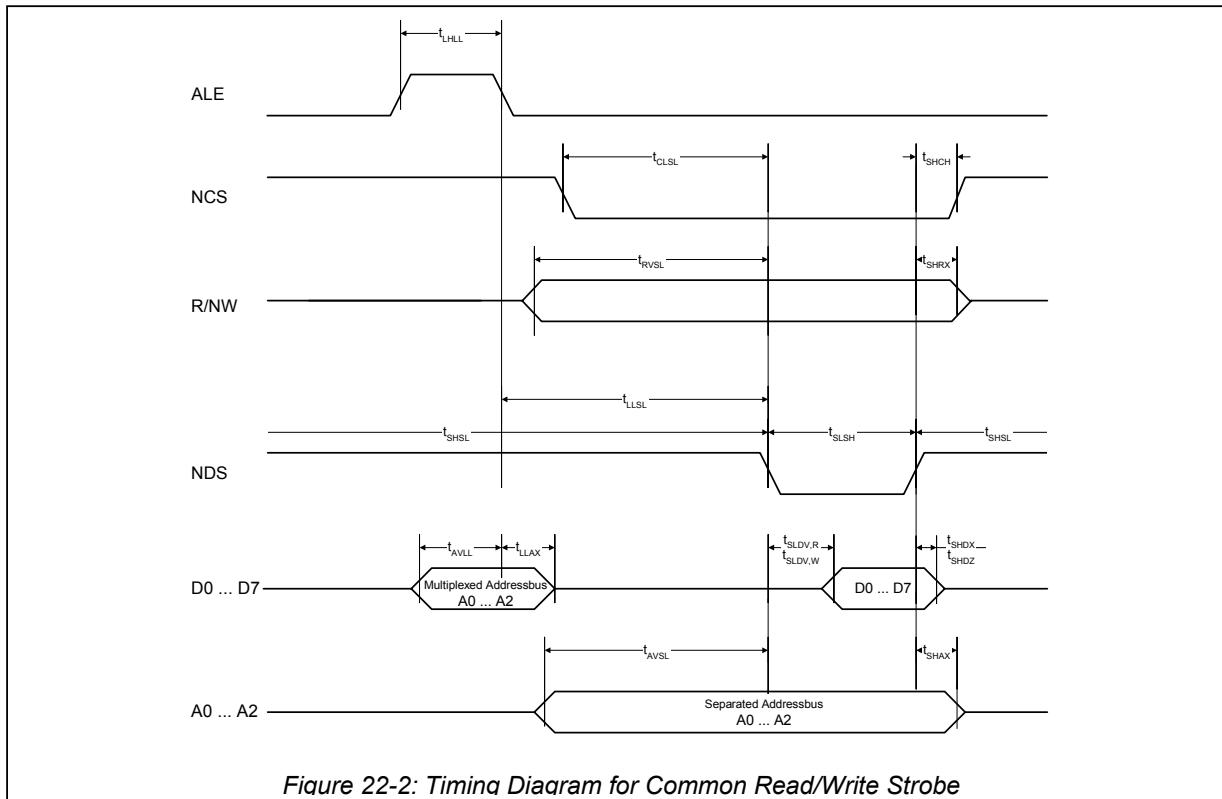


Figure 22-2: Timing Diagram for Common Read/Write Strobe

Note: For separated address and data bus the signal ALE is not relevant and the multiplexed addresses on the data bus don't care. For the multiplexed address and data bus the address lines A0 to A2 have to be connected as described in 4.3.

Multiple Protocol Contactless Reader IC

CL RC632

22.5.2.3 Bus Timing for EPP

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{LLLH}	nAStrb pulse width	20		ns
t_{AVLH}	Multiplexed Address Bus valid to nAStrb high (Set Up Time)	15		ns
t_{LHAX}	Multiplexed Address Bus valid after nAStrb high (Hold Time)	8		ns
t_{CLSL}	NCS low to nDStrb low	0		ns
t_{SHCH}	nDStrb high to NCS high	0		ns
$t_{SLDV,R}$	nDStrb low to DATA valid (read cycle)		65	ns
t_{SHDZ}	nDStrb low to DATA high impedance (read cycle)		20	ns
$t_{SLDV,W}$	nDStrb low to DATA valid (write cycle, Set up Time)		35	ns
t_{SHDX}	DATA hold after nDStrb high (write cycle, Hold Time)	8		ns
t_{SHRX}	nWrite hold after nDStrb high	8		ns
t_{SLSH}	nDStrb pulse width	65		ns
t_{RVSL}	nWrite valid to nDStrb low	8		ns
t_{SLWH}	nDStrb low to nWait high		75	ns
t_{SHWL}	nDStrb high to nWait low		75	ns

Table 22-12: Timing Specification for Common Read/Write Strobe

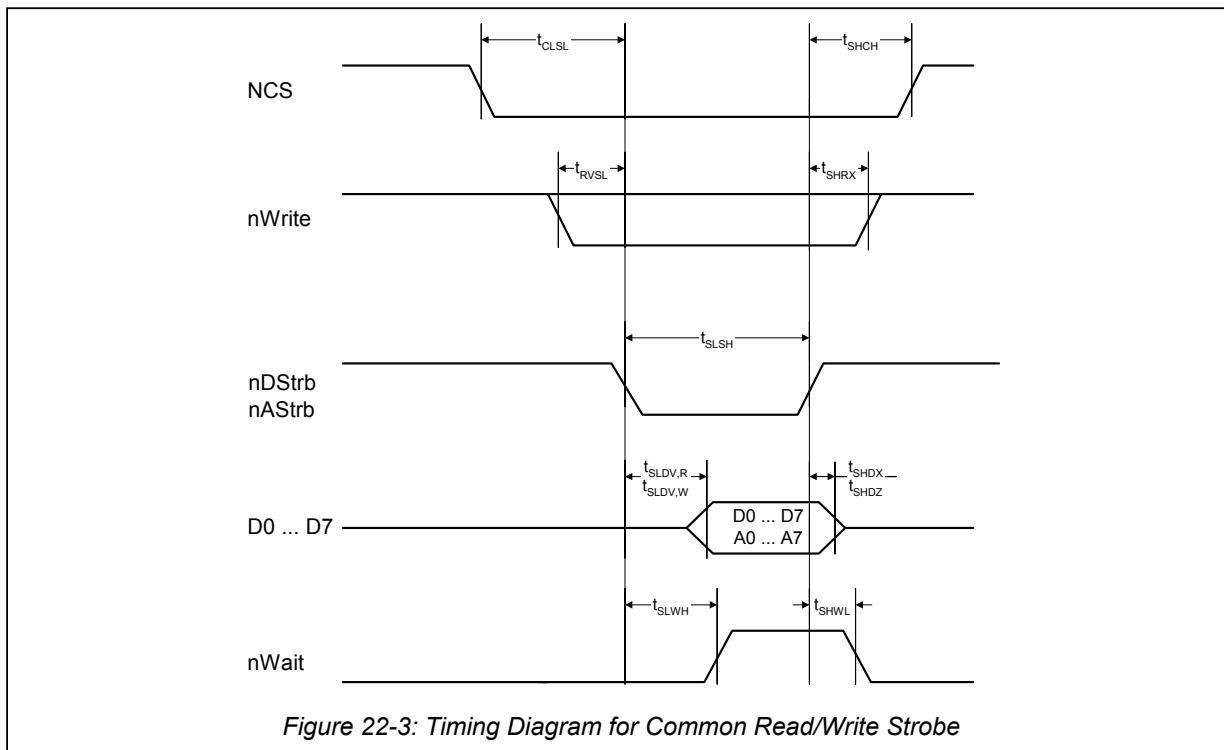


Figure 22-3: Timing Diagram for Common Read/Write Strobe

Remark: The figure does not distinguish between the Address Write Cycle and a Data Write Cycle. Take in account, that timings for the Address Write and Data Write Cycle are different. For the EPP-Mode the address lines A0 to A2 have to be connected as described in 4.3.

Multiple Protocol Contactless Reader IC

CL RC632

22.5.2.4 Timing for SPI compatible interface

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{SCKL}	SCK low pulse width	100		ns
t_{SCKH}	SCK high pulse width	100		ns
t_{SHDX}	SCK high to data changes	20		ns
t_{DXSH}	data changes to SCK high	20		ns
t_{SLDX}	SCK low to data changes		15	ns
t_{SLNH}	SCK low to NSS high	20		ns

Table 22-13 Timing Specification for SPI

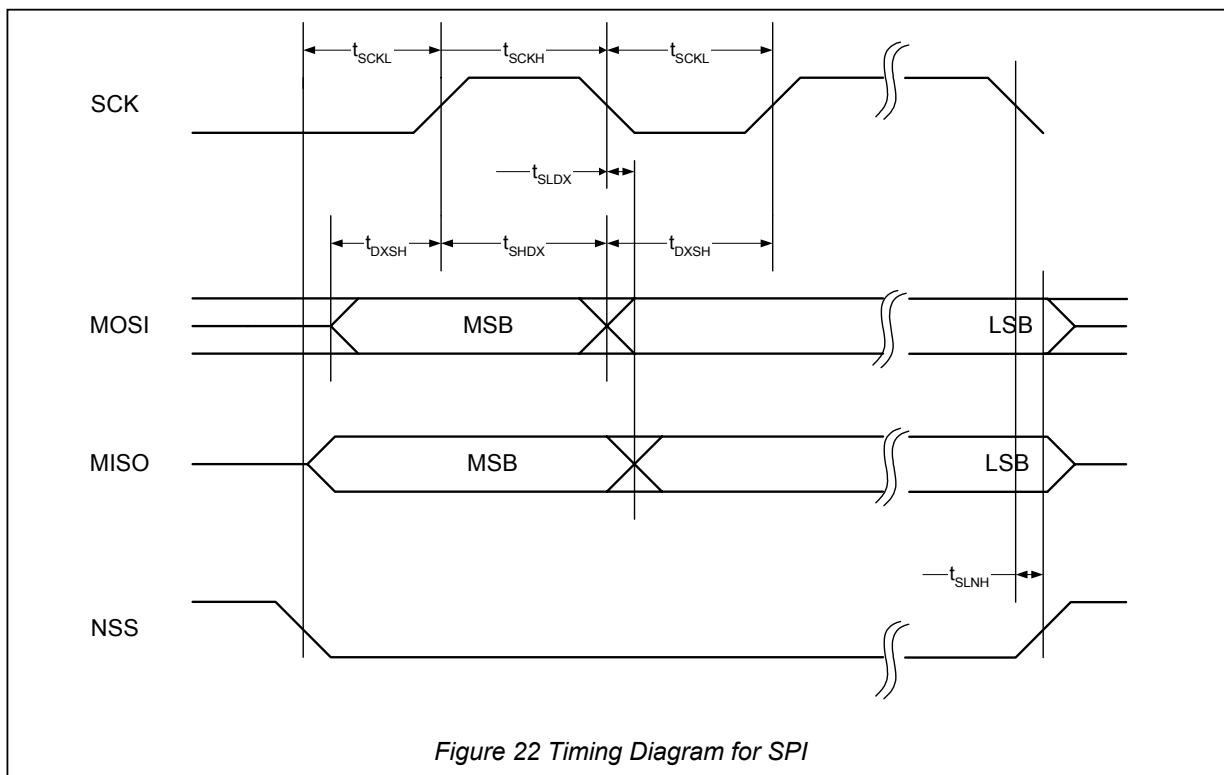


Figure 22 Timing Diagram for SPI

Note: To send more than bytes in one datastream the NSS signal has to low all the time.
 To send more than one datastream NSS has to be set to HIGH level in between the datastreams.

Multiple Protocol Contactless Reader IC**CL RC632**

22.5.3 CLOCK FREQUENCY

The clock input is pin 1, OSCIN.

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT
Clock Frequency (checked by the clock filter)	f_{OSCIN}		13.56		MHz
Duty Cycle of Clock Frequency	d_{FEC}	40	50	60	%
Jitter of Clock Edges	t_{jitter}			10	ps

The clock applied to the CL RC632 acts as time basis for the coder and decoder of the synchronous system. Therefore stability of clock frequency is an important factor for proper performance. To obtain highest performance, clock jitter shall be as small as possible. This is best achieved using the internal oscillator buffer with the recommended circuitry (see 12).

Multiple Protocol Contactless Reader IC**CL RC632**

23 E²PROM CHARACTERISTICS

The E²PROM has a size of 32x16x8 = 4.096 bit.

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
t _{EEEndurance}	Data Endurance		100.000		erase/write cycles
t _{EERetention}	Data Retention	T _{amb} ≤ 55°C	10		years
t <subeeerase< sub=""></subeeerase<>	Erase Time			2.9	ms
t <subeewrite< sub=""></subeewrite<>	Write Time			2.9	ms

Table 23-1:E²PROM Characteristics

Multiple Protocol Contactless Reader IC**CL RC632**

24 ESD SPECIFICATION

To ensure the usage of the CL RC632 during production the ICs is specified as described in the following table.

TEST	NAME	CONDITIONS	MAX
ESDH	ESD Susceptibility (Human body model)	1500 Ω, 100 pF	1 kV
ESDM	ESD Susceptibility (Machine model)	0.75 μH, 200 pF	100 V

Table 24-1. ESD Specification

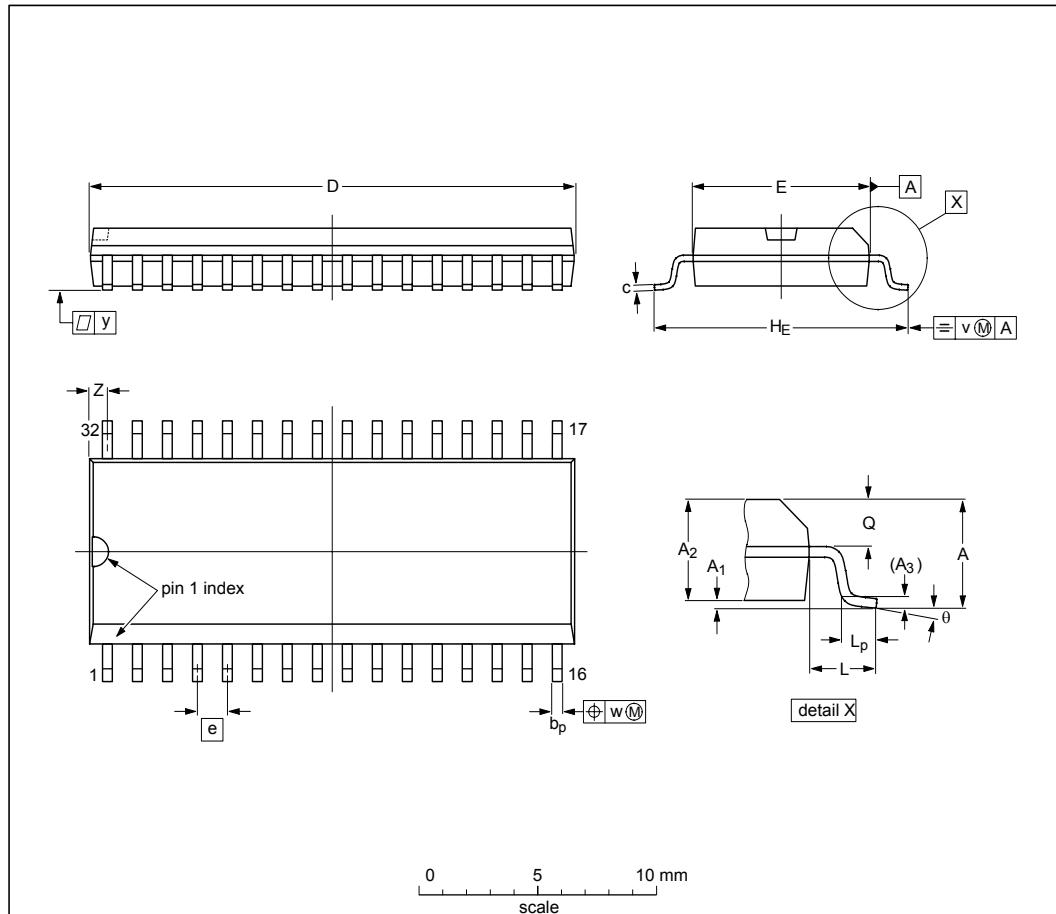
Multiple Protocol Contactless Reader IC

CL RC632

25 PACKAGE OUTLINES

25.1 SO32

SO32: plastic small outline package; 32 leads; body width 7.5 mm SOT287-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A ₁	A ₂	A ₃	b _p	c	D ⁽¹⁾	E ⁽¹⁾	e	H _E	L	L _p	Q	v	w	y	z ⁽¹⁾	θ
mm	2.65	0.3 0.1	2.45 2.25	0.25	0.49 0.36	0.27 0.18	20.7 20.3	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.2 1.0	0.25	0.25	0.1	0.95 0.55	8° 0°
inches	0.10	0.012 0.004	0.096 0.086	0.01	0.02 0.01	0.011 0.007	0.81 0.80	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.016	0.047 0.039	0.01	0.01	0.004 0.022	0.037 0.022	

Note

- Plastic or metal protrusions of 0.15 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES					EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ				
SOT287-1							95-01-25 97-05-22

Figure 255-1: Outline and Dimension of CL RC632 in SO32

Multiple Protocol Contactless Reader IC**CL RC632****Definitions**

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics section of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

26 DISCLAIMERS**26.1 Life support applications**

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

26.2 Licence Policy

Purchase of this Philips IC with a functionality according to ISO/IEC 15693 Standard does not convey an implied license under any patent right on this standard. A license for the Philips portfolio of patents on the ISO/IEC 15693 Standard can be obtained via the Philips Intellectual Property and Standards department. For more information please contact the nearest Philips Semiconductors sales office.

27 REVISION HISTORY

27.1 Update from Revision 2.0 to Revision 3.0

The whole document was editorial revised. New phrasings and additional descriptions have been added.
The table below refers to relevant changes in content.

Chapter	Description
5.2.4.2	Added Bits 4-3: 'ISO Selection'
22.4.1	Chapter 'Input Pin characteristics': dynamical input voltage range for RX pin added

Table 0-1: Update from Revision 2.0 to Revision 3.0

27.2 Versions Up to Revision 3.0

REVISION	DATE	CPCN	PAGE	DESCRIPTION
3.0	November2002	-		first published version
2.0	June 2002	-		second published version
1.0	January2002	-		internal version

Table 0-2: Document Revision History

Philips Semiconductors - a worldwide company

Contact Information

For additional information please visit <http://www.semiconductors.philips.com>. Fax: +31 40 27 24825
For sales offices addresses send e-mail to: sales.addresses@www.semiconductors.philips.com.

© Koninklijke Philips Electronics N.V. 2002

SCA74

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without any notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Let's make things better.

**Philips
Semiconductors**



PHILIPS

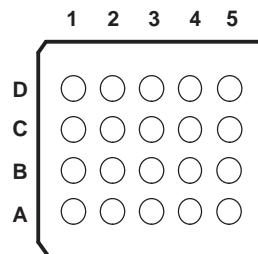
8-BIT BIDIRECTIONAL VOLTAGE-LEVEL TRANSLATOR WITH AUTO-DIRECTION SENSING AND ± 15 -kV ESD PROTECTION

Check for Samples: [TXB0108](#)

FEATURES

- 1.2 V to 3.6 V on A Port and 1.65 to 5.5 V on B Port ($V_{CCA} \leq V_{CCB}$)
- V_{CC} Isolation Feature – If Either V_{CC} Input Is at GND, All Outputs Are in the High-Impedance State
- OE Input Circuit Referenced to V_{CCA}
- Low Power Consumption, 4- μ A Max I_{CC}
- I_{off} Supports Partial-Power-Down Mode Operation
- Latch-Up Performance Exceeds 100 mA Per JESD 78, Class II
- ESD Protection Exceeds JESD 22
 - A Port
 - 2000-V Human-Body Model (A114-B)
 - 1000-V Charged-Device Model (C101)
 - B Port
 - ± 15 -kV Human-Body Model (A114-B)
 - 1000-V Charged-Device Model (C101)

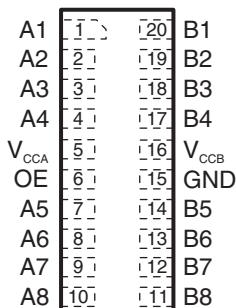
**GXY OR ZXY PACKAGE
(BOTTOM VIEW)**



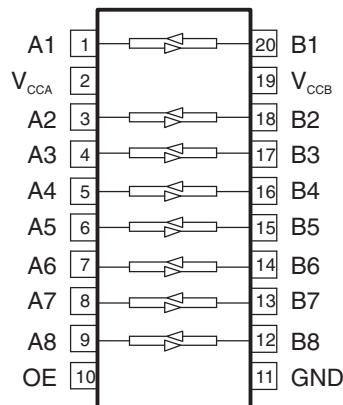
**TERMINAL ASSIGNMENTS
(20-Ball GXY/ZXY Package)**

	1	2	3	4	5
D	V_{CCB}	B2	B4	B6	B8
C	B1	B3	B5	B7	GND
B	A1	A3	A5	A7	OE
A	V_{CCA}	A2	A4	A6	A8

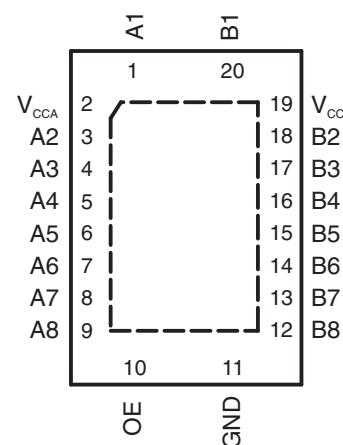
**DQS PACKAGE
(TOP VIEW)**



**PW PACKAGE
(TOP VIEW)**



**RGY PACKAGE
(TOP VIEW)**



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

DESCRIPTION/ORDERING INFORMATION

This 8-bit noninverting translator uses two separate configurable power-supply rails. The A port is designed to track V_{CCA} . V_{CCA} accepts any supply voltage from 1.2 V to 3.6 V. The B port is designed to track V_{CCB} . V_{CCB} accepts any supply voltage from 1.65 V to 5.5 V. This allows for universal low-voltage bidirectional translation between any of the 1.2-V, 1.5-V, 1.8-V, 2.5-V, 3.3-V, and 5-V voltage nodes. V_{CCA} should not exceed V_{CCB} .

When the output-enable (OE) input is low, all outputs are placed in the high-impedance state.

The TXB0101 is designed so that the OE input circuit is supplied by V_{CCA} .

This device is fully specified for partial-power-down applications using I_{off} . The I_{off} circuitry disables the outputs, preventing damaging current backflow through the device when it is powered down.

To ensure the high-impedance state during power up or power down, OE should be tied to GND through a pulldown resistor; the minimum value of the resistor is determined by the current-sourcing capability of the driver.

Table 1. ORDERING INFORMATION⁽¹⁾

T _A	PACKAGE ⁽²⁾		ORDERABLE PART NUMBER	TOP-SIDE MARKING
–40°C to 85°C	QFN – RGY	Reel of 1000	TXB0108RGYR	YE08
	SON – DQS	Reel of 2000	TXB0108DQSR	5MR
	TSSOP – PW	Reel of 2000	TXB0108PWR	YE08
	VFBGA – GXY	Reel of 2500	TXB0108GXYR	YE08
	VFBGA – ZXY (Pb-free)	Reel of 2500	TXB0108ZXYR	YE08

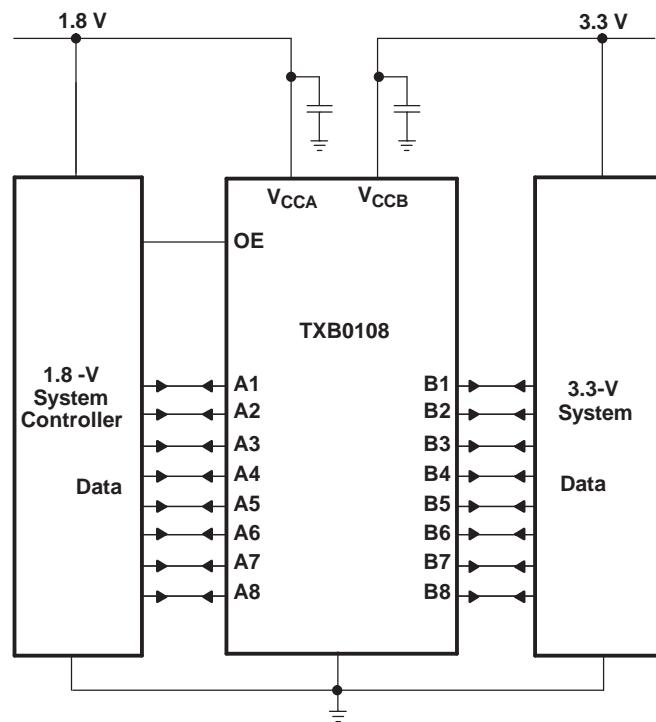
(1) For the most current package and ordering information, see the Package Option Addendum at the end of this document, or see the TI web site at www.ti.com.

(2) Package drawings, thermal data, and symbolization are available at www.ti.com/packaging.

PIN DESCRIPTION

NO. (DQS, PW, RGY)	NAME	FUNCTION
1	A1	Input/output 1. Referenced to V_{CCA} .
2	V_{CCA}	A-port supply voltage. $1.1 \text{ V} \leq V_{CCA} \leq 3.6 \text{ V}$, $V_{CCA} \leq V_{CCB}$.
3	A2	Input/output 2. Referenced to V_{CCA} .
4	A3	Input/output 3. Referenced to V_{CCA} .
5	A4	Input/output 4. Referenced to V_{CCA} .
6	A5	Input/output 5. Referenced to V_{CCA} .
7	A6	Input/output 6. Referenced to V_{CCA} .
8	A7	Input/output 7. Referenced to V_{CCA} .
9	A8	Input/output 8. Referenced to V_{CCA} .
10	OE	Output enable. Pull OE low to place all outputs in 3-state mode. Referenced to V_{CCA} .
11	GND	Ground
12	B8	Input/output 8. Referenced to V_{CCB} .
13	B7	Input/output 7. Referenced to V_{CCB} .
14	B6	Input/output 6. Referenced to V_{CCB} .
15	B5	Input/output 5. Referenced to V_{CCB} .
16	B4	Input/output 4. Referenced to V_{CCB} .
17	B3	Input/output 3. Referenced to V_{CCB} .
18	B2	Input/output 2. Referenced to V_{CCB} .
19	V_{CCB}	B-port supply voltage. $1.65 \text{ V} \leq V_{CCB} \leq 5.5 \text{ V}$.
20	B1	Input/output 1. Referenced to V_{CCB} .

TYPICAL OPERATING CIRCUIT



Absolute Maximum Ratings⁽¹⁾

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
V_{CCA}	Supply voltage range	-0.5	4.6	V
V_{CCB}	Supply voltage range	-0.5	6.5	V
V_I	Input voltage range ⁽²⁾	-0.5	6.5	V
V_O	Voltage range applied to any output in the high-impedance or power-off state ⁽²⁾	-0.5	6.5	V
V_O	Voltage range applied to any output in the high or low state ^{(2) (3)}	A inputs	$-0.5 \text{ } V_{CCA} + 0.5$	V
		B inputs	$-0.5 \text{ } V_{CCB} + 0.5$	
I_{IK}	Input clamp current	$V_I < 0$	-50	mA
I_{OK}	Output clamp current	$V_O < 0$	-50	mA
I_O	Continuous output current		± 50	mA
	Continuous current through V_{CCA} , V_{CCB} , or GND		± 100	mA
θ_{JA}	Package thermal impedance	DQS package	TBD	°C/W
		GXY/ZXY package ⁽⁴⁾	78	
		PW package ⁽⁴⁾	83	
		RGY package ⁽⁵⁾	37	
T_{stg}	Storage temperature range	-65	150	°C

- (1) Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) The input and output negative-voltage ratings may be exceeded if the input and output current ratings are observed.
- (3) The value of V_{CCA} and V_{CCB} are provided in the recommended operating conditions table.
- (4) The package thermal impedance is calculated in accordance with JESD 51-7.
- (5) The package thermal impedance is calculated in accordance with JESD 51-5.

Recommended Operating Conditions⁽¹⁾ (2)

		V _{CCA}	V _{CCB}	MIN	MAX	UNIT
V _{CCA}	Supply voltage			1.2	3.6	V
V _{CCB}				1.65	5.5	
V _{IH}	High-level input voltage	Data inputs	1.2 V to 3.6 V	1.65 V to 5.5 V	V _{CCI} × 0.65 ⁽³⁾	V
		OE			V _{CCI}	
V _{IL}	Low-level input voltage	Data inputs	1.2 V to 5.5 V	1.65 V to 5.5 V	0 V _{CCI} × 0.35 ⁽³⁾	V
		OE	1.2 V to 3.6 V		0 V _{CCA} × 0.35	
Δt/Δv	Input transition rise or fall rate	A-port inputs	1.2 V to 3.6 V	1.65 V to 5.5 V	40	ns/V
		B-port inputs	1.2 V to 3.6 V	1.65 V to 3.6 V	40	
				4.5 V to 5.5 V	30	
T _A	Operating free-air temperature				-40	85 °C

(1) The A and B sides of an unused data I/O pair must be held in the same state, i.e., both at V_{CCI} or both at GND.

(2) V_{CCA} must be less than or equal to V_{CCB} and must not exceed 3.6 V.

(3) V_{CCI} is the supply voltage associated with the input port.

Electrical Characteristics^{(1) (2)}

over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	V _{CCA}	V _{CCB}	T _A = 25°C			−40°C to 85°C		UNIT
				MIN	TYP	MAX	MIN	MAX	
V _{OHA}	I _{OH} = −20 μA	1.2 V			1.1				V
		1.4 V to 3.6 V					V _{CCA} − 0.4		
V _{OLA}	I _{OL} = 20 μA	1.2 V			0.9				V
		1.4 V to 3.6 V					0.4		
V _{OHB}	I _{OH} = −20 μA		1.65 V to 5.5 V				V _{CCB} − 0.4		V
V _{OLB}	I _{OL} = 20 μA		1.65 V to 5.5 V				0.4		V
I _I	OE	1.2 V to 3.6 V	1.65 V to 5.5 V			±1		±2	μA
I _{off}	A port	0 V	0 V to 5.5 V			±1		±2	μA
	B port	0 V to 3.6 V	0 V			±1		±2	
I _{oz}	A or B port	OE = GND	1.2 V to 3.6 V	1.65 V to 5.5 V		±1		±2	μA
I _{CCA}	V _I = V _{CCI} or GND, I _O = 0	1.2 V	1.65 V to 5.5 V		0.06				μA
		1.4 V to 3.6 V					5		
		3.6 V	0 V				2		
		0 V	5.5 V				−2		
I _{CCB}	V _I = V _{CCI} or GND, I _O = 0	1.2 V	1.65 V to 5.5 V		3.4				μA
		1.4 V to 3.6 V					5		
		3.6 V	0 V				−2		
		0 V	5.5 V				2		
I _{CCA} + I _{CCB}	V _I = V _{CCI} or GND, I _O = 0	1.2 V	1.65 V to 5.5 V		3.5				μA
		1.4 V to 3.6 V					10		
I _{CCZA}	V _I = V _{CCI} or GND, I _O = 0, OE = GND	1.2 V	1.65 V to 5.5 V		0.05				μA
		1.4 V to 3.6 V					5		
I _{CCZB}	V _I = V _{CCI} or GND, I _O = 0, OE = GND	1.2 V	1.65 V to 5.5 V		3.3				μA
		1.4 V to 3.6 V					5		
C _I	OE	1.2 V to 3.6 V	1.65 V to 5.5 V		5		5.5	pF	
C _{io}	A port		1.2 V to 3.6 V	1.65 V to 5.5 V	5		6.5	pF	
	B port				8		10		

(1) V_{CCI} is the supply voltage associated with the input port.

(2) V_{CCO} is the supply voltage associated with the output port.

Timing Requirements

T_A = 25°C, V_{CCA} = 1.2 V

		V _{CCB} = 1.8 V ± 0.15 V	V _{CCB} = 2.5 V ± 0.2 V	V _{CCB} = 3.3 V ± 0.3 V	V _{CCB} = 5 V ± 0.5 V	UNIT	
						TYP	TYP
Data rate		20	20	20	20		Mbps
t _w	Pulse duration	50	50	50	50		ns

Timing Requirements

over recommended operating free-air temperature range, V_{CCA} = 1.5 V ± 0.1 V (unless otherwise noted)

		V _{CCB} = 1.8 V ± 0.15 V	V _{CCB} = 2.5 V ± 0.2 V	V _{CCB} = 3.3 V ± 0.3 V	V _{CCB} = 5 V ± 0.5 V	UNIT	
						MIN	MAX
Data rate			50	50	50	50	Mbps
t _w	Pulse duration		20	20	20	20	ns

Timing Requirements

over recommended operating free-air temperature range, $V_{CCA} = 1.8 \text{ V} \pm 0.15 \text{ V}$ (unless otherwise noted)

		$V_{CCB} = 1.8 \text{ V} \pm 0.15 \text{ V}$		$V_{CCB} = 2.5 \text{ V} \pm 0.2 \text{ V}$		$V_{CCB} = 3.3 \text{ V} \pm 0.3 \text{ V}$		$V_{CCB} = 5 \text{ V} \pm 0.5 \text{ V}$		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
Data rate			52		60		60		60	Mbps
t_w	Pulse duration	Data inputs	19	17	17	17	17	17	ns	

Timing Requirements

over recommended operating free-air temperature range, $V_{CCA} = 2.5 \text{ V} \pm 0.2 \text{ V}$ (unless otherwise noted)

		$V_{CCB} = 2.5 \text{ V} \pm 0.2 \text{ V}$		$V_{CCB} = 3.3 \text{ V} \pm 0.3 \text{ V}$		$V_{CCB} = 5 \text{ V} \pm 0.5 \text{ V}$		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
Data rate			70		100		100	Mbps
t_w	Pulse duration	Data inputs	14	10	10	10	10	ns

Timing Requirements

over recommended operating free-air temperature range, $V_{CCA} = 3.3 \text{ V} \pm 0.3 \text{ V}$ (unless otherwise noted)

		$V_{CCB} = 3.3 \text{ V} \pm 0.3 \text{ V}$		$V_{CCB} = 5 \text{ V} \pm 0.5 \text{ V}$		UNIT
		MIN	MAX	MIN	MAX	
Data rate			100		100	Mbps
t_w	Pulse duration	Data inputs	10	10	10	ns

Switching Characteristics

$T_A = 25^\circ\text{C}$, $V_{CCA} = 1.2 \text{ V}$

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$V_{CCB} = 1.8 \text{ V}$		$V_{CCB} = 2.5 \text{ V}$		$V_{CCB} = 3.3 \text{ V}$		$V_{CCB} = 5 \text{ V}$		UNIT
			TYP	TYP	TYP	TYP	TYP	TYP	TYP		
t_{pd}	A	B	9.5	7.9	7.6	8.5					ns
	B	A	9.2	8.8	8.4	8					
t_{en}	OE	A	1	1	1	1					μs
		B	1	1	1	1					
t_{dis}	OE	A	20	17	17	18					ns
		B	20	16	15	15					
t_{rA} , t_{fA}	A-port rise and fall times		4.1	4.4	4.1	3.9					ns
t_{rB} , t_{fB}	B-port rise and fall times		5	5	5.1	5.1					ns
$t_{SK(O)}$	Channel-to-channel skew		2.4	1.7	1.9	7					ns
Max data rate			20	20	20	20					Mbps

Switching Characteristics

over recommended operating free-air temperature range, $V_{CCA} = 1.5 \text{ V} \pm 0.1 \text{ V}$ (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$V_{CCB} = 1.8 \text{ V} \pm 0.15 \text{ V}$		$V_{CCB} = 2.5 \text{ V} \pm 0.2 \text{ V}$		$V_{CCB} = 3.3 \text{ V} \pm 0.3 \text{ V}$		$V_{CCB} = 5 \text{ V} \pm 0.5 \text{ V}$		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
t_{pd}	A	B	1.4	12.9	1.2	10.1	1.1	10	0.8	9.9	ns
	B	A	0.9	14.2	0.7	12	0.4	11.7	0.3	13.7	
t_{en}	OE	A			1		1		1		μs
		B			1		1		1		
t_{dis}	OE	A	6.6	33	6.4	25.3	6.1	23.1	5.9	24.6	ns
		B	6.6	35.6	5.8	25.6	5.5	22.1	5.6	20.6	
t_{rA}, t_{fA}	A-port rise and fall times		0.8	6.5	0.8	6.3	0.8	6.3	0.8	6.3	ns
t_{rB}, t_{fB}	B-port rise and fall times		1	7.3	0.7	4.9	0.7	4.6	0.6	4.6	ns
$t_{SK(O)}$	Channel-to-channel skew			2.6		1.9		1.6		1.3	ns
Max data rate			50		50		50		50		Mbps

Switching Characteristics

over recommended operating free-air temperature range, $V_{CCA} = 1.8 \text{ V} \pm 0.15 \text{ V}$ (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$V_{CCB} = 1.8 \text{ V} \pm 0.15 \text{ V}$		$V_{CCB} = 2.5 \text{ V} \pm 0.2 \text{ V}$		$V_{CCB} = 3.3 \text{ V} \pm 0.3 \text{ V}$		$V_{CCB} = 5 \text{ V} \pm 0.5 \text{ V}$		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
t_{pd}	A	B	1.6	11	1.4	7.7	1.3	6.8	1.2	6.5	ns
	B	A	1.5	12	1.2	8.4	0.8	7.6	0.5	7.1	
t_{en}	OE	A			1		1		1		μs
		B			1		1		1		
t_{dis}	OE	A	5.9	26.7	5.6	21.6	5.4	18.9	4.8	18.7	ns
		B	6.1	33.9	5.2	23.7	5	19.9	5	17.6	
t_{rA}, t_{fA}	A-port rise and fall times		0.7	5.1	0.7	5	1	5	0.7	5	ns
t_{rB}, t_{fB}	B-port rise and fall times		1	7.3	0.7	5	0.7	3.9	0.6	3.8	ns
$t_{SK(O)}$	Channel-to-channel skew			0.8		0.7		0.6		0.6	ns
Max data rate			52		60		60		60		Mbps

Switching Characteristics

over recommended operating free-air temperature range, $V_{CCA} = 2.5 \text{ V} \pm 0.2 \text{ V}$ (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$V_{CCB} = 2.5 \text{ V} \pm 0.2 \text{ V}$		$V_{CCB} = 3.3 \text{ V} \pm 0.3 \text{ V}$		$V_{CCB} = 5 \text{ V} \pm 0.5 \text{ V}$		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
t_{pd}	A	B	1.1	6.4	1	5.3	0.9	4.7	ns
	B	A	1	7	0.6	5.6	0.3	4.4	
t_{en}	OE	A			1		1		μs
		B			1		1		
t_{dis}	OE	A	5	16.9	4.9	15	4.5	13.8	ns
		B	4.8	21.8	4.5	17.9	4.4	15.2	
t_{rA}, t_{fA}	A-port rise and fall times		0.8	3.6	0.6	3.6	0.5	3.5	ns
t_{rB}, t_{fB}	B-port rise and fall times		0.6	4.9	0.7	3.9	0.6	3.2	ns
$t_{SK(O)}$	Channel-to-channel skew				0.4		0.3		ns
Max data rate			70		100		100		Mbps

Switching Characteristics

over recommended operating free-air temperature range, $V_{CCA} = 3.3 \text{ V} \pm 0.3 \text{ V}$ (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$V_{CCB} = 3.3 \text{ V} \pm 0.3 \text{ V}$		$V_{CCB} = 5 \text{ V} \pm 0.5 \text{ V}$		UNIT	
			MIN	MAX	MIN	MAX		
t_{pd}	A	B	0.9	4.9	0.8	4	ns	
	B	A	0.5	5.4	0.2	4		
t_{en}	OE	A			1		μs	
		B			1			
t_{dis}	OE	A	4.5	13.9	4.1	12.4	ns	
		B	4.1	17.3	4	14.4		
t_{rA}, t_{fA}	A-port rise and fall times		0.5	3	0.5	3	ns	
t_{rB}, t_{fB}	B-port rise and fall times		0.7	3.9	0.6	3.2	ns	
$t_{SK(O)}$	Channel-to-channel skew				0.4		0.3	ns
Max data rate			100		100		Mbps	

Operating Characteristics

 $T_A = 25^\circ\text{C}$

PARAMETER		TEST CONDITIONS	V_{CCA}							UNIT	
			1.2 V	1.2 V	1.5 V	1.8 V	2.5 V	2.5 V	3.3 V		
			V_{CCB}								
			5 V	1.8 V	1.8 V	1.8 V	2.5 V	5 V	3.3 V to 5 V		
				TYP	TYP	TYP	TYP	TYP	TYP		
C_{pdA}	A-port input, B-port output	$C_L = 0$, $f = 10 \text{ MHz}$, $t_r = t_f = 1 \text{ ns}$, $OE = V_{CCA}$ (outputs enabled)	9	8	7	7	7	7	8	pF	
	B-port input, A-port output		12	11	11	11	11	11	11		
C_{pdB}	A-port input, B-port output		35	26	27	27	27	27	28		
	B-port input, A-port output		26	19	18	18	18	20	21		
C_{pdA}	A-port input, B-port output	$C_L = 0$, $f = 10 \text{ MHz}$, $t_r = t_f = 1 \text{ ns}$, $OE = GND$ (outputs disabled)	0.01	0.01	0.01	0.01	0.01	0.01	0.01	pF	
	B-port input, A-port output		0.01	0.01	0.01	0.01	0.01	0.01	0.01		
C_{pdB}	A-port input, B-port output		0.01	0.01	0.01	0.01	0.01	0.01	0.03		
	B-port input, A-port output		0.01	0.01	0.01	0.01	0.01	0.01	0.03		

PRINCIPLES OF OPERATION

Applications

The TXB0108 can be used in level-translation applications for interfacing devices or systems operating at different interface voltages with one another.

Architecture

The TXB0108 architecture (see [Figure 1](#)) does not require a direction-control signal to control the direction of data flow from A to B or from B to A. In a dc state, the output drivers of the TXB0108 can maintain a high or low, but are designed to be weak, so that they can be overdriven by an external driver when data on the bus starts flowing the opposite direction.

The output one shots detect rising or falling edges on the A or B ports. During a rising edge, the one shot turns on the PMOS transistors (T1, T3) for a short duration, which speeds up the low-to-high transition. Similarly, during a falling edge, the one shot turns on the NMOS transistors (T2, T4) for a short duration, which speeds up the high-to-low transition. The typical output impedance during output transition is 70Ω at $V_{CCO} = 1.2\text{ V}$ to 1.8 V , 50Ω at $V_{CCO} = 1.8\text{ V}$ to 3.3 V and 40Ω at $V_{CCO} = 3.3\text{ V}$ to 5 V .

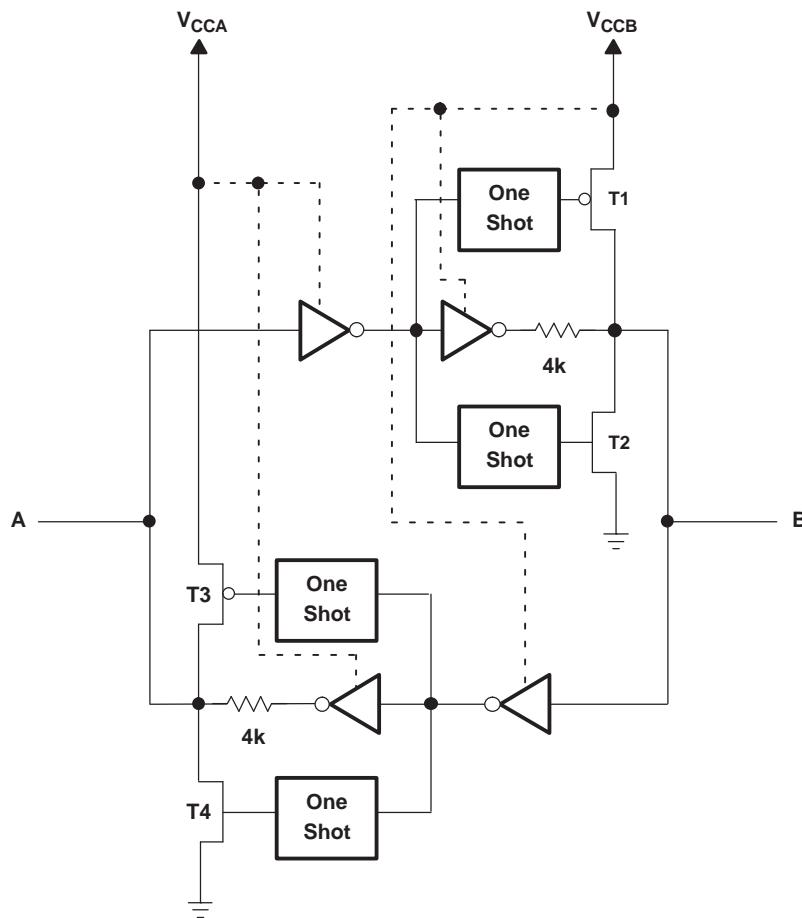
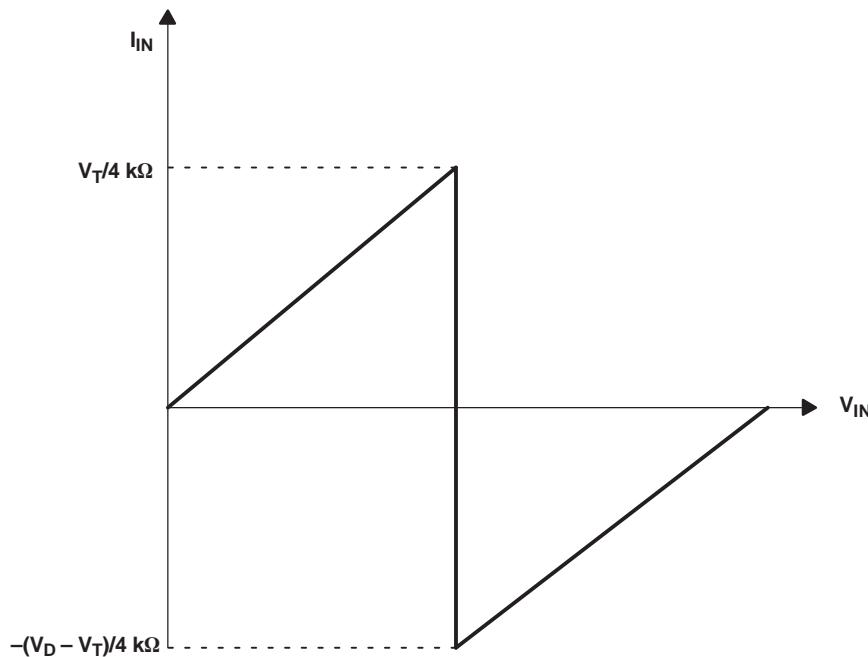


Figure 1. Architecture of TXB0108 I/O Cell

Input Driver Requirements

Typical I_{IN} vs V_{IN} characteristics of the TXB0108 are shown in [Figure 2](#). For proper operation, the device driving the data I/Os of the TXB0108 must have drive strength of at least $\pm 2\text{ mA}$.



- A. V_T is the input threshold voltage of the TXB0108 (typically $V_{CC1}/2$).
- B. V_D is the supply voltage of the external driver.

Figure 2. Typical I_{IN} vs V_{IN} Curve

Power Up

During operation, ensure that $V_{CCA} \leq V_{CCB}$ at all times. During power-up sequencing, $V_{CCA} \geq V_{CCB}$ does not damage the device, so any power supply can be ramped up first. The TXB0108 has circuitry that disables all output ports when either V_{CC} is switched off ($V_{CC/A/B} = 0$ V).

Enable and Disable

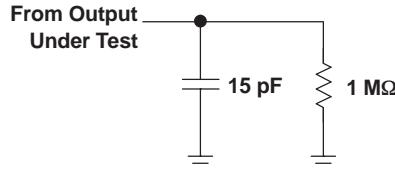
The TXB0108 has an OE input that is used to disable the device by setting OE = low, which places all I/Os in the high-impedance (Hi-Z) state. The disable time (t_{dis}) indicates the delay between when OE goes low and when the outputs actually get disabled (Hi-Z). The enable time (t_{en}) indicates the amount of time the user must allow for the one-shot circuitry to become operational after OE is taken high.

Pullup or Pulldown Resistors on I/O Lines

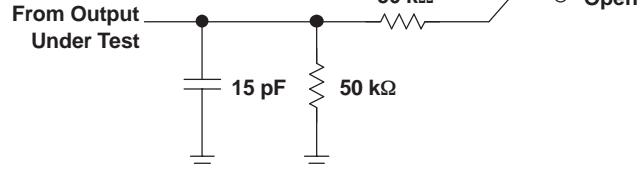
The TXB0108 is designed to drive capacitive loads of up to 70 pF. The output drivers of the TXB0108 have low dc drive strength. If pullup or pulldown resistors are connected externally to the data I/Os, their values must be kept higher than 50 k Ω to ensure that they do not contend with the output drivers of the TXB0108.

For the same reason, the TXB0108 should not be used in applications such as I²C or 1-Wire where an open-drain driver is connected on the bidirectional data I/O. For these applications, use a device from the TI TSSOP1xx series of level translators.

PARAMETER MEASUREMENT INFORMATION

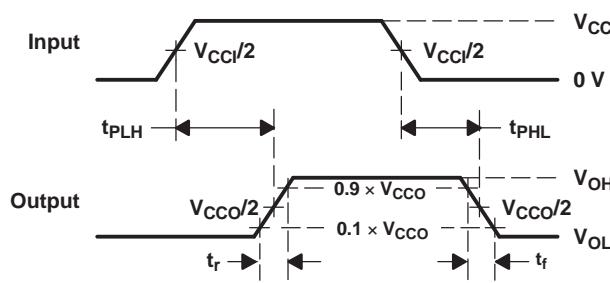


LOAD CIRCUIT FOR MAX DATA RATE,
PULSE DURATION PROPAGATION
DELAY OUTPUT RISE AND FALL TIME
MEASUREMENT

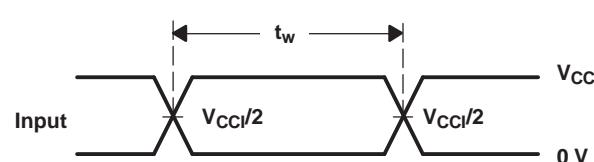


LOAD CIRCUIT FOR
ENABLE/DISABLE
TIME MEASUREMENT

TEST	S1
t_{PLH}/t_{PLZ} t_{PHL}/t_{PZH}	$2 \times V_{CCO}$ Open



VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES



VOLTAGE WAVEFORMS
PULSE DURATION

- A. C_L includes probe and jig capacitance.
- B. All input pulses are supplied by generators having the following characteristics: PRR ≤ 10 MHz, $Z_O = 50 \Omega$, $dv/dt \geq 1 \text{ V/ns}$.
- C. The outputs are measured one at a time, with one transition per measurement.
- D. t_{PLH} and t_{PHL} are the same as t_{pd} .
- E. V_{CCI} is the V_{CC} associated with the input port.
- F. V_{CCO} is the V_{CC} associated with the output port.
- G. All parameters and waveforms are not applicable to all devices.

Figure 3. Load Circuits and Voltage Waveforms



www.ti.com

PACKAGE OPTION ADDENDUM

14-Oct-2010

PACKAGING INFORMATION

Orderable Device	Status ⁽¹⁾	Package Type	Package Drawing	Pins	Package Qty	Eco Plan ⁽²⁾	Lead/Ball Finish	MSL Peak Temp ⁽³⁾	Samples (Requires Login)
TXB0108DQSR	ACTIVE	USON	DQS	20	3000	Green (RoHS & no Sb/Br)	Call TI	Level-1-260C-UNLIM	Request Free Samples
TXB0108PWR	ACTIVE	TSSOP	PW	20	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	Request Free Samples
TXB0108PWRG4	ACTIVE	TSSOP	PW	20	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	Request Free Samples
TXB0108RGYR	ACTIVE	VQFN	RGY	20	3000	Green (RoHS & no Sb/Br)	Call TI	Level-2-260C-1 YEAR	Request Free Samples
TXB0108RGYRG4	ACTIVE	VQFN	RGY	20	3000	Green (RoHS & no Sb/Br)	Call TI	Level-2-260C-1 YEAR	Request Free Samples
TXB0108ZXYR	ACTIVE	BGA MICROSTAR JUNIOR	ZXY	20	2500	Green (RoHS & no Sb/Br)	SNAGCU	Level-1-260C-UNLIM	Request Free Samples

⁽¹⁾ The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSOLETE: TI has discontinued the production of the device.

⁽²⁾ Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check <http://www.ti.com/productcontent> for the latest availability information and additional product content details.

TBD: The Pb-Free/Green conversion plan has not been defined.

Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.

Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

⁽³⁾ MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.



www.ti.com

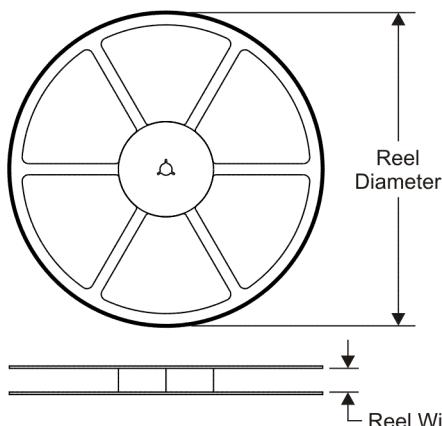
PACKAGE OPTION ADDENDUM

14-Oct-2010

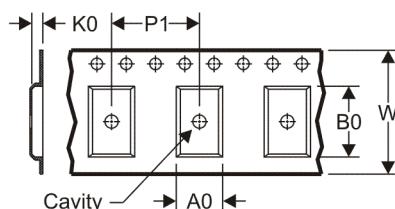
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

TAPE AND REEL INFORMATION

REEL DIMENSIONS

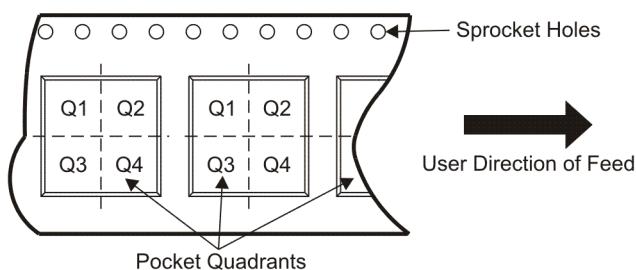


TAPE DIMENSIONS



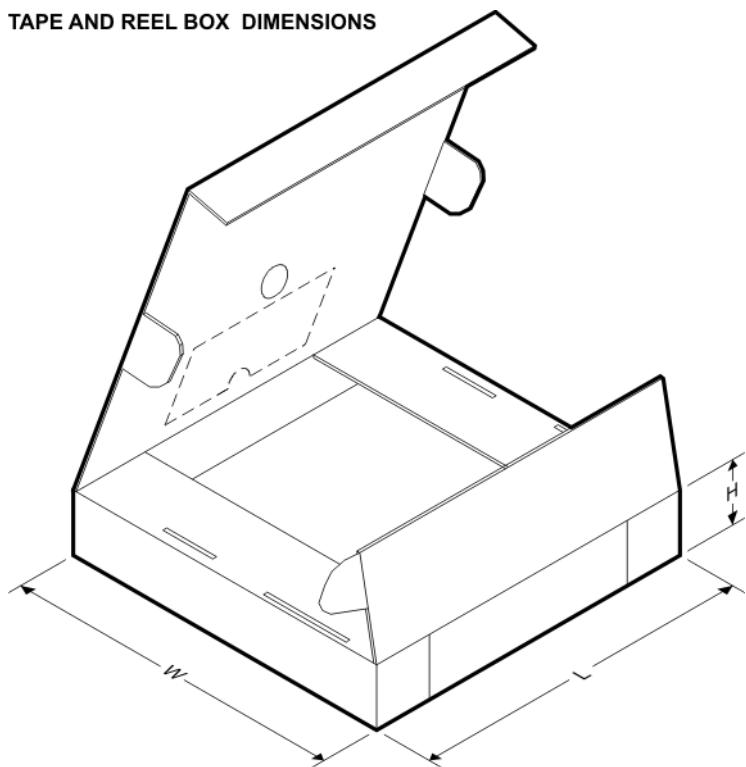
A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
TXB0108DQSR	USON	DQS	20	3000	177.8	12.4	2.21	4.22	0.81	4.0	12.0	Q1
TXB0108PWR	TSSOP	PW	20	2000	330.0	16.4	6.95	7.1	1.6	8.0	16.0	Q1
TXB0108RGYR	VQFN	RGY	20	3000	330.0	12.4	3.8	4.8	1.6	8.0	12.0	Q1
TXB0108RGYR	VQFN	RGY	20	3000	330.0	12.4	3.8	4.8	1.6	8.0	12.0	Q1
TXB0108ZXYR	BGA MICROSTAR JUNIOR	ZXY	20	2500	330.0	12.4	2.8	3.3	1.0	4.0	12.0	Q2

TAPE AND REEL BOX DIMENSIONS


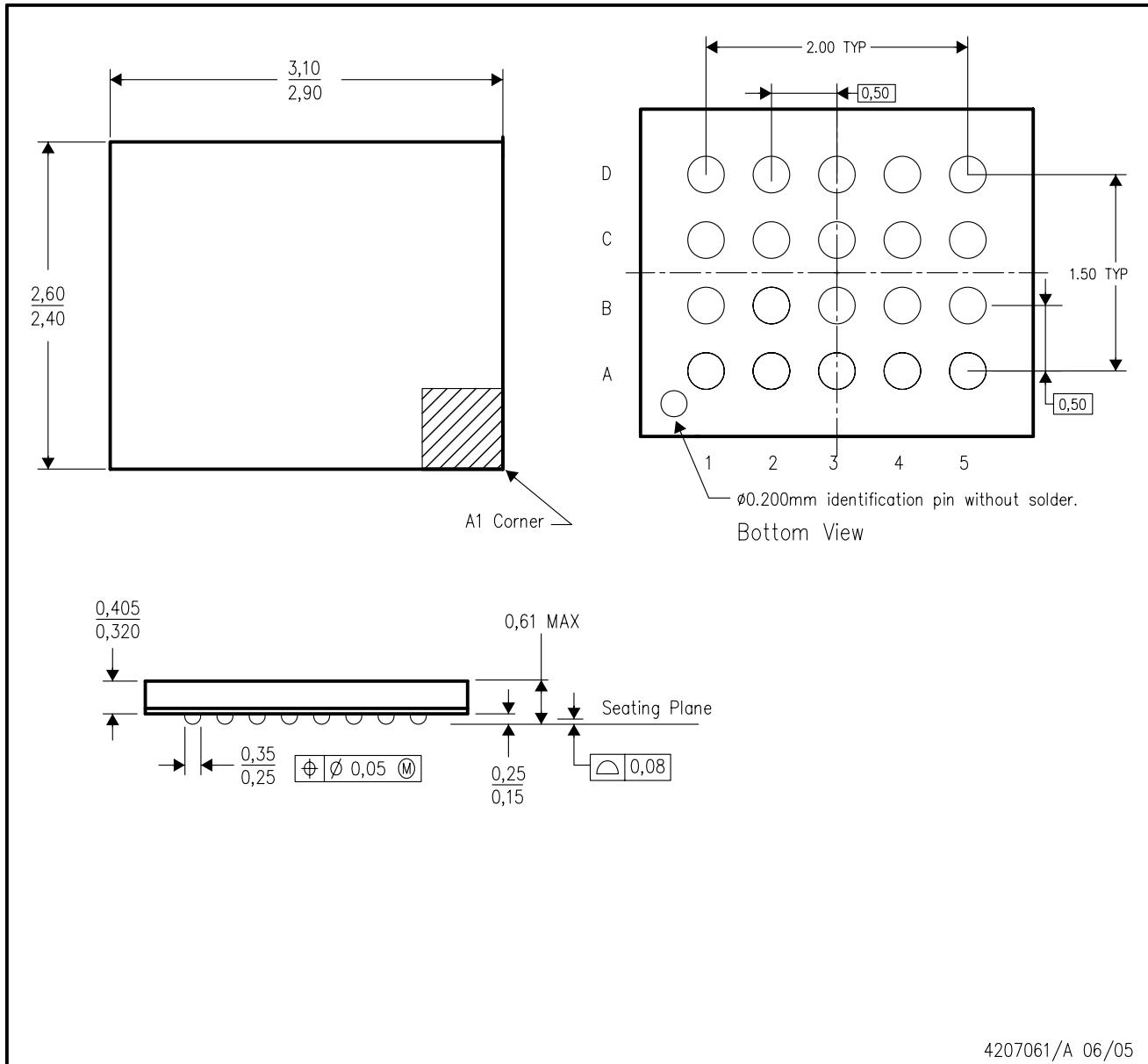
*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
TXB0108DQSR	USON	DQS	20	3000	202.0	201.0	28.0
TXB0108PWR	TSSOP	PW	20	2000	346.0	346.0	33.0
TXB0108RGYR	VQFN	RGY	20	3000	346.0	346.0	29.0
TXB0108RGYR	VQFN	RGY	20	3000	355.0	350.0	50.0
TXB0108ZXYR	BGA MICROSTAR JUNIOR	ZXY	20	2500	340.5	338.1	20.6

MECHANICAL DATA

ZXY (S-PBGA-N20)

PLASTIC BALL GRID ARRAY

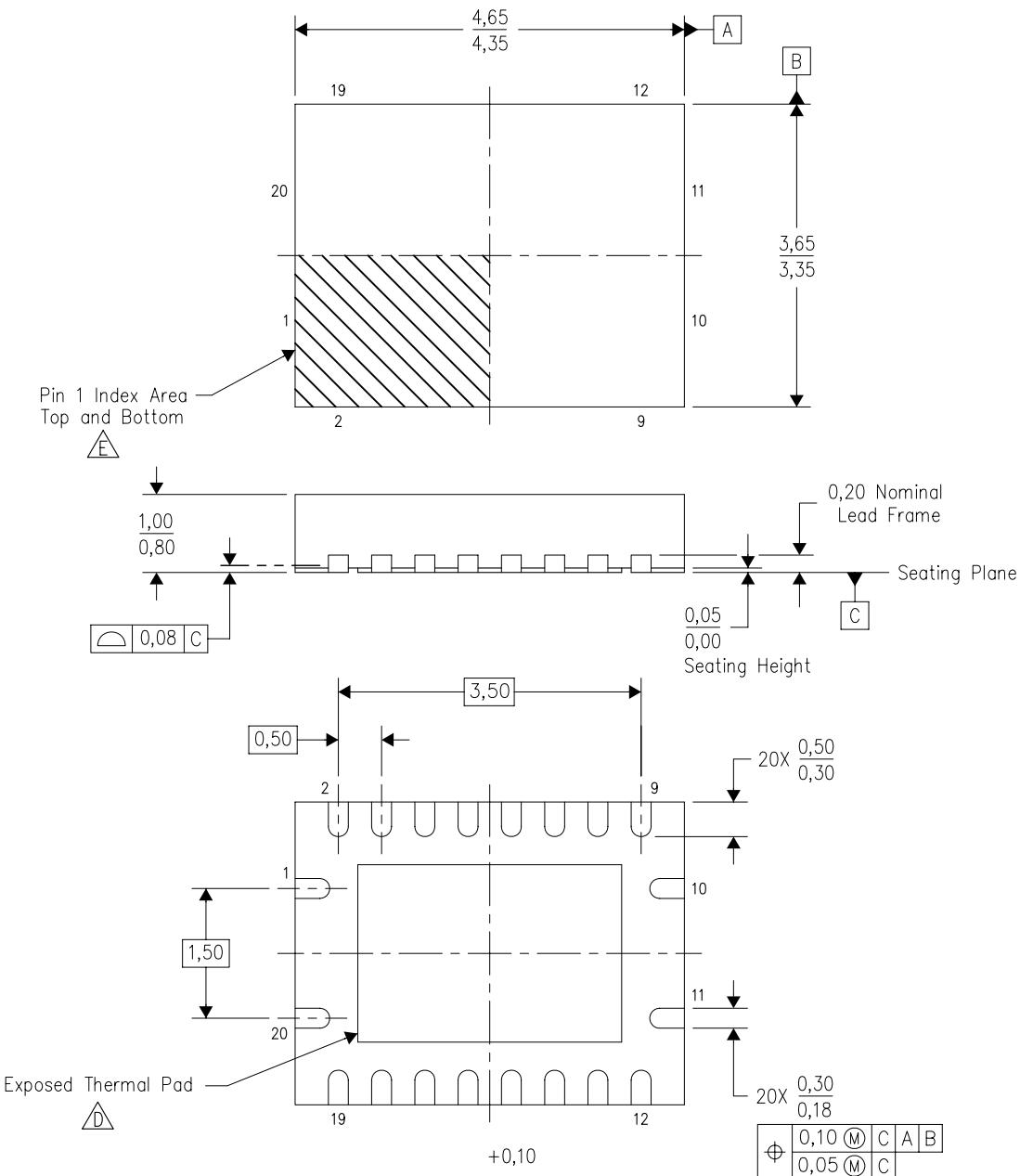


- NOTES:
- All linear dimensions are in millimeters.
 - This drawing is subject to change without notice.
 - This package is a lead-free solder ball design.

MECHANICAL DATA

RGY (R-PVQFN-N20)

PLASTIC QUAD FLATPACK NO-LEAD



Bottom View

4203539-4/H 06/2009

NOTES: A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M-1994.

B. This drawing is subject to change without notice.

C. QFN (Quad Flatpack No-Lead) package configuration.

D. The package thermal pad must be soldered to the board for thermal and mechanical performance. See the Product Data Sheet for details regarding the exposed thermal pad dimensions.

E. Pin 1 identifiers are located on both top and bottom of the package and within the zone indicated. The Pin 1 identifiers are either a molded, marked, or metal feature.

F. Package complies to JEDEC MO-241 variation BC.

THERMAL PAD MECHANICAL DATA

RGY (R-PVQFN-N20)

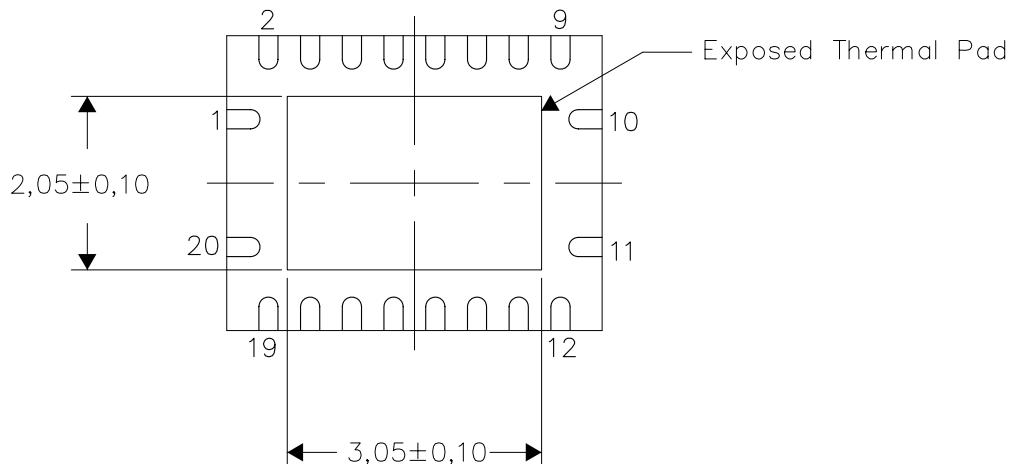
PLASTIC QUAD FLATPACK NO-LEAD

THERMAL INFORMATION

This package incorporates an exposed thermal pad that is designed to be attached directly to an external heatsink. The thermal pad must be soldered directly to the printed circuit board (PCB). After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For information on the Quad Flatpack No-Lead (QFN) package and its advantages, refer to Application Report, QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271. This document is available at www.ti.com.

The exposed thermal pad dimensions for this package are shown in the following illustration.



Bottom View

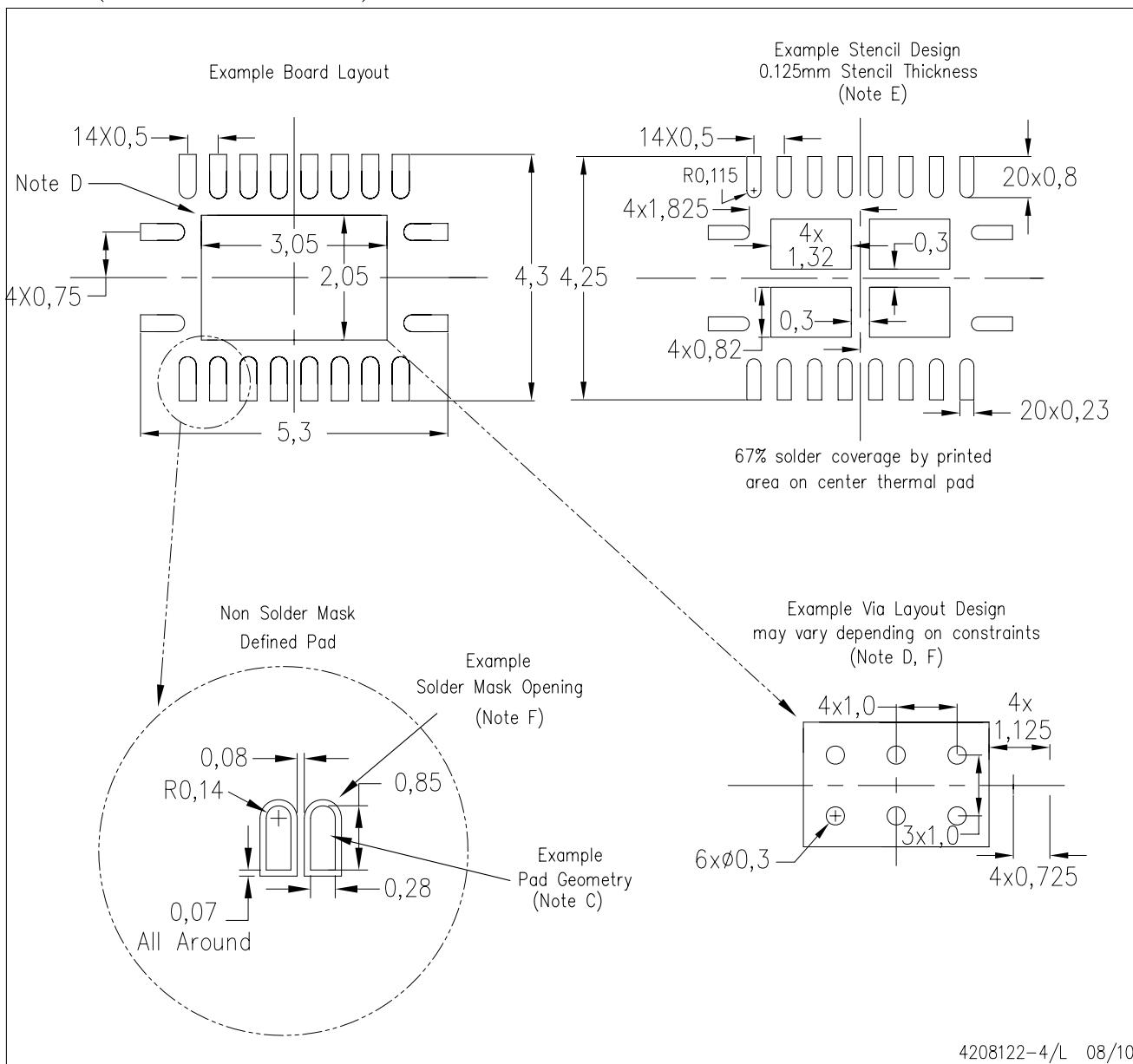
NOTE: All linear dimensions are in millimeters

Exposed Thermal Pad Dimensions

LAND PATTERN DATA

RGY (R-PVQFN-N20)

PLASTIC QUAD FLATPACK NO-LEAD



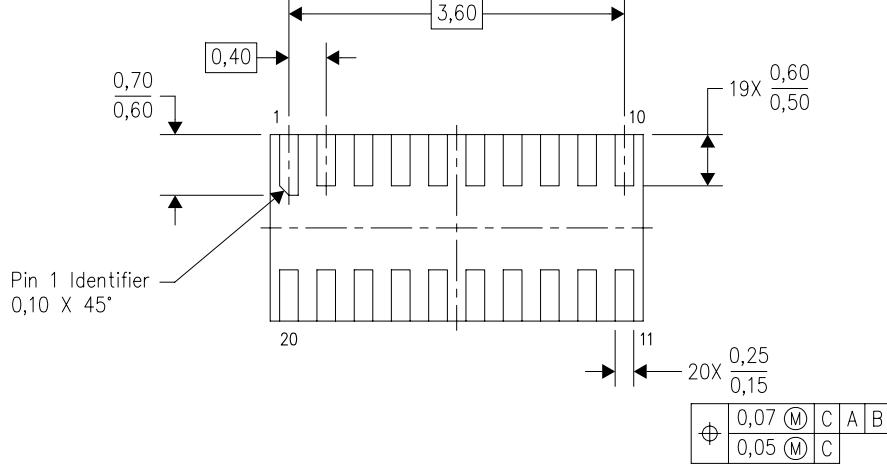
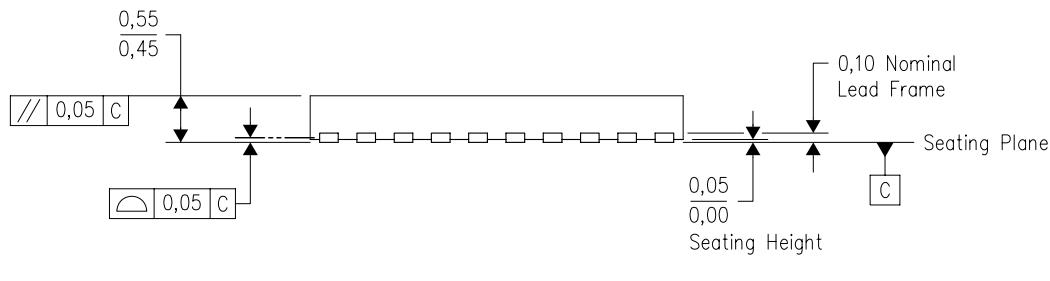
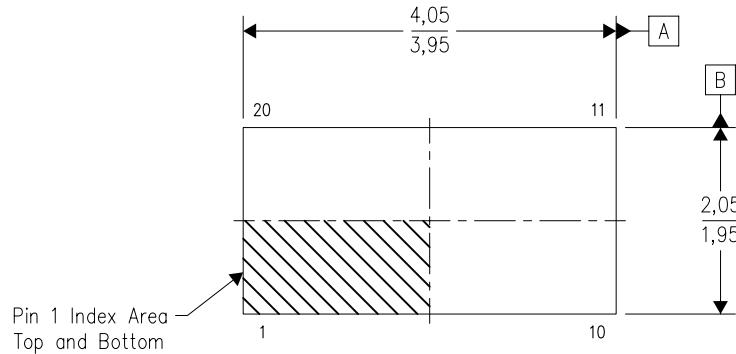
4208122-4/L 08/10

- NOTES:
- All linear dimensions are in millimeters.
 - This drawing is subject to change without notice.
 - Publication IPC-7351 is recommended for alternate designs.
 - This package is designed to be soldered to a thermal pad on the board. Refer to Application Note, Quad Flat-Pack QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271, and also the Product Data Sheets for specific thermal information, via requirements, and recommended board layout. These documents are available at www.ti.com <<http://www.ti.com>>.
 - Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC 7525 for stencil design considerations.
 - Customers should contact their board fabrication site for minimum solder mask web tolerances between signal pads.

MECHANICAL DATA

DQS (R-PSON-N20)

PLASTIC SMALL OUTLINE NO-LEAD



Bottom View

4210558/A 09/2009

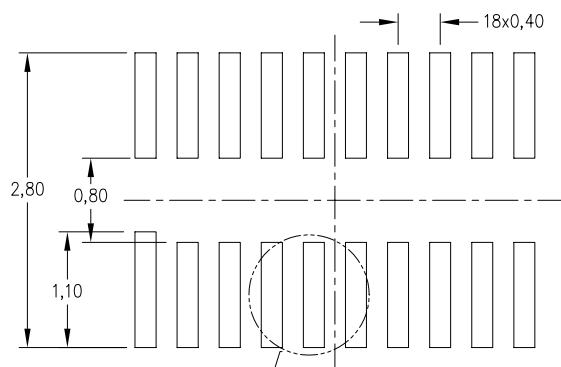
- NOTES:
- A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M-1994.
 - B. This drawing is subject to change without notice.
 - C. SON (Small Outline No-Lead) package configuration.

LAND PATTERN DATA

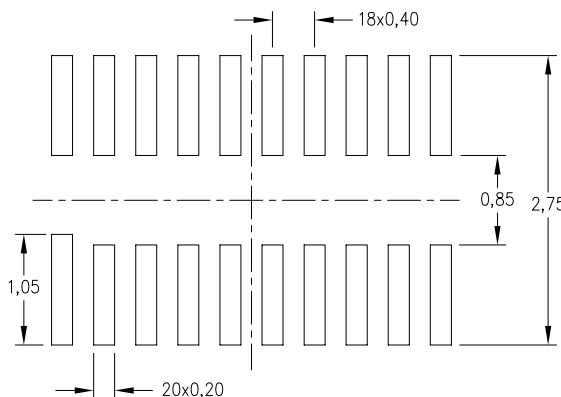
DQS (R-PSON-N20)

PLASTIC SMALL OUTLINE NO-LEAD

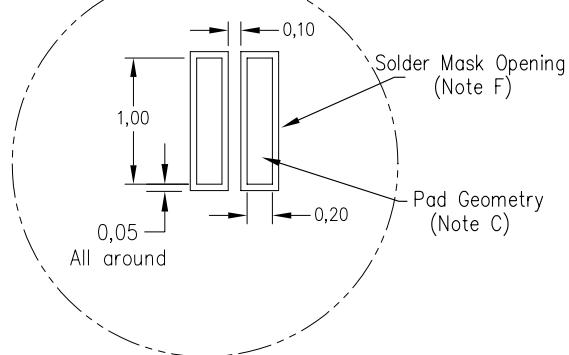
Example Board Layout



Example Stencil Design
0.125mm Stencil Thickness
(Note D)



Non Solder Mask
Defined Pad



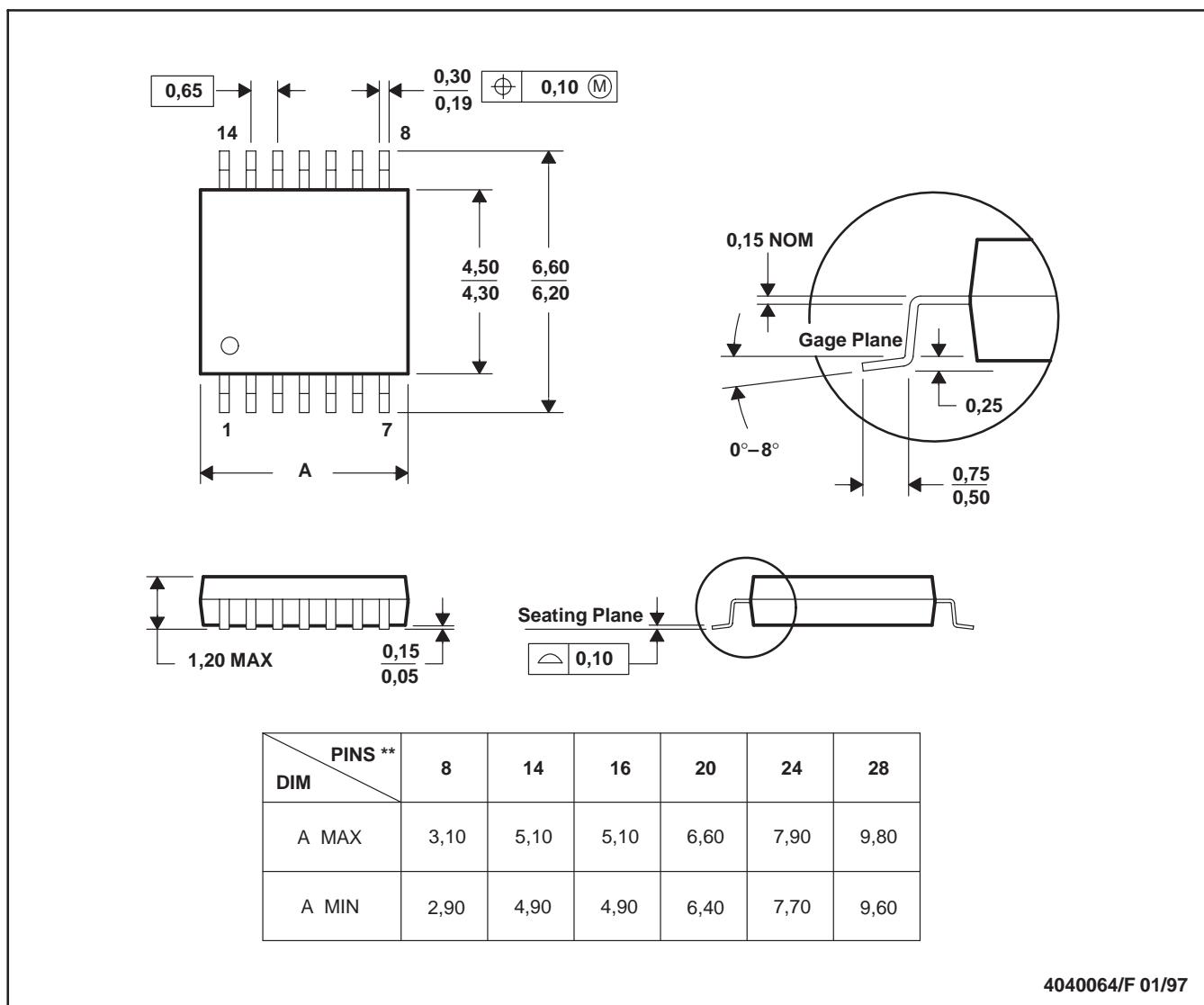
4211489/A 11/10

- NOTES:
- A. All linear dimensions are in millimeters.
 - B. This drawing is subject to change without notice.
 - C. Publication IPC-7351 is recommended for alternate designs.
 - D. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC 7525 for stencil design considerations.
 - E. Customers should contact their board fabrication site for minimum solder mask web tolerances between signal pads.

PW (R-PDSO-G**)

PLASTIC SMALL-OUTLINE PACKAGE

14 PINS SHOWN



- NOTES:
- All linear dimensions are in millimeters.
 - This drawing is subject to change without notice.
 - Body dimensions do not include mold flash or protrusion not to exceed 0,15.
 - Falls within JEDEC MO-153

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

LM1117/LM1117I

800mA Low-Dropout Linear Regulator

General Description

The LM1117 is a series of low dropout voltage regulators with a dropout of 1.2V at 800mA of load current. It has the same pin-out as National Semiconductor's industry standard LM317.

The LM1117 is available in an adjustable version, which can set the output voltage from 1.25V to 13.8V with only two external resistors. In addition, it is also available in five fixed voltages, 1.8V, 2.5V, 2.85V, 3.3V, and 5V.

The LM1117 offers current limiting and thermal shutdown. Its circuit includes a zener trimmed bandgap reference to assure output voltage accuracy to within $\pm 1\%$.

The LM1117 series is available in LLP, TO-263, SOT-223, TO-220, and TO-252 D-PAK packages. A minimum of $10\mu F$ tantalum capacitor is required at the output to improve the transient response and stability.

Features

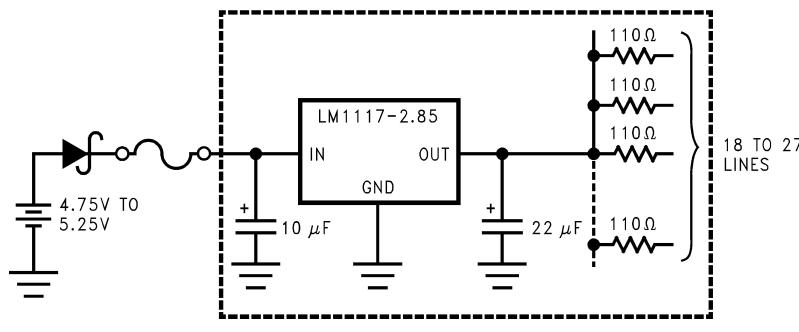
- Available in 1.8V, 2.5V, 2.85V, 3.3V, 5V, and Adjustable Versions
- Space Saving SOT-223 and LLP Packages
- Current Limiting and Thermal Protection
- Output Current 800mA
- Line Regulation 0.2% (Max)
- Load Regulation 0.4% (Max)
- Temperature Range
 - LM1117 0°C to 125°C
 - LM1117I -40°C to 125°C

Applications

- 2.85V Model for SCSI-2 Active Termination
- Post Regulator for Switching DC/DC Converter
- High Efficiency Linear Regulators
- Battery Charger
- Battery Powered Instrumentation

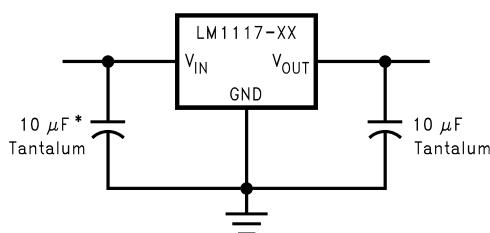
Typical Application

Active Terminator for SCSI-2 Bus



10091905

Fixed Output Regulator



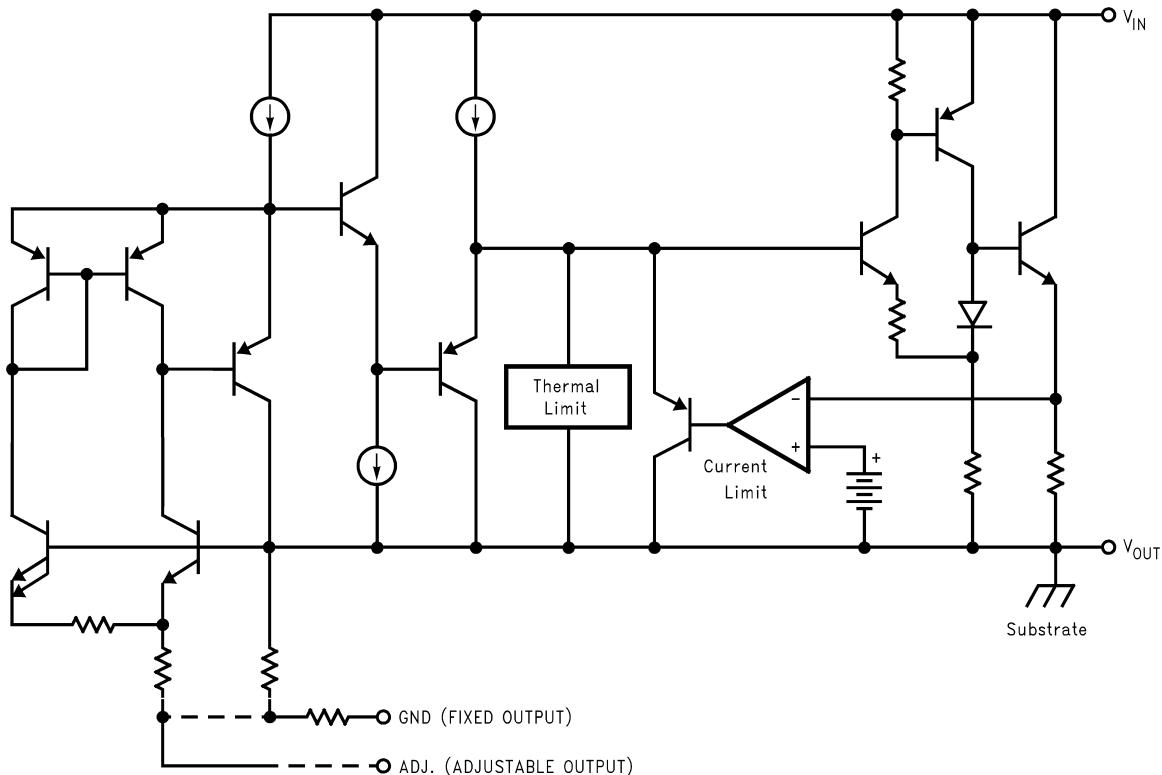
* Required if the regulator is located far from the power supply filter.

10091928

Ordering Information

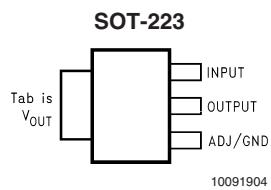
Package	Temperature Range	Part Number	Packaging Marking	Transport Media	NSC Drawing
3-lead SOT-223	0°C to +125°C	LM1117MPX-ADJ	N03A	Tape and Reel	MP04A
		LM1117MPX-1.8	N12A	Tape and Reel	
		LM1117MPX-2.5	N13A	Tape and Reel	
		LM1117MPX-2.85	N04A	Tape and Reel	
		LM1117MPX-3.3	N05A	Tape and Reel	
		LM1117MPX-5.0	N06A	Tape and Reel	
	-40°C to +125°C	LM1117IMPX-ADJ	N03B	Tape and Reel	
		LM1117IMPX-3.3	N05B	Tape and Reel	
		LM1117IMPX-5.0	N06B	Tape and Reel	
3-lead TO-220	0°C to +125°C	LM1117T-ADJ	LM1117T-ADJ	Rails	T03B
		LM1117T-1.8	LM1117T-1.8	Rails	
		LM1117T-2.5	LM1117T-2.5	Rails	
		LM1117T-2.85	LM1117T-2.85	Rails	
		LM1117T-3.3	LM1117T-3.3	Rails	
		LM1117T-5.0	LM1117T-5.0	Rails	
3-lead TO-252	0°C to +125°C	LM1117DTX-ADJ	LM1117DT-ADJ	Tape and Reel	TD03B
		LM1117DTX-1.8	LM1117DT-1.8	Tape and Reel	
		LM1117DTX-2.5	LM1117DT-2.5	Tape and Reel	
		LM1117DTX-2.85	LM1117DT-2.85	Tape and Reel	
		LM1117DTX-3.3	LM1117DT-3.3	Tape and Reel	
		LM1117DTX-5.0	LM1117DT-5.0	Tape and Reel	
	-40°C to +125°C	LM1117IDTX-ADJ	LM1117IDT-ADJ	Tape and Reel	
		LM1117IDTX-3.3	LM1117IDT-3.3	Tape and Reel	
		LM1117IDTX-5.0	LM1117IDT-5.0	Tape and Reel	
8-lead LLP	0°C to +125°C	LM1117LDX-ADJ	1117ADJ	Tape and Reel	LDC08A
		LM1117LDX-1.8	1117-18	Tape and Reel	
		LM1117LDX-2.5	1117-25	Tape and Reel	
		LM1117LDX-2.85	1117-28	Tape and Reel	
		LM1117LDX-3.3	1117-33	Tape and Reel	
		LM1117LDX-5.0	1117-50	Tape and Reel	
	-40°C to 125°C	LM1117ILDX-ADJ	1117IAD	Tape and Reel	
		LM1117ILDX-3.3	1117I33	Tape and Reel	
		LM1117ILDX-5.0	1117I50	Tape and Reel	
TO-263	0°C to +125°C	LM1117SX-ADJ	LM1117SADJ	Tape and Reel	TS3B
		LM1117SX-2.85	LM1117S2.85	Tape and Reel	
		LM1117SX-3.3	LM1117S3.3	Tape and Reel	
		LM1117SX-5.0	LM1117S5.0	Tape and Reel	

Block Diagram

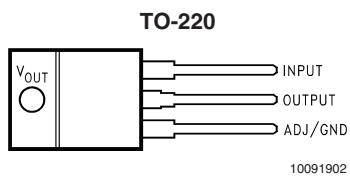


10091901

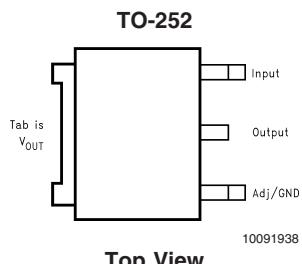
Connection Diagrams



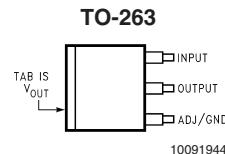
Top View



Top View



Top View

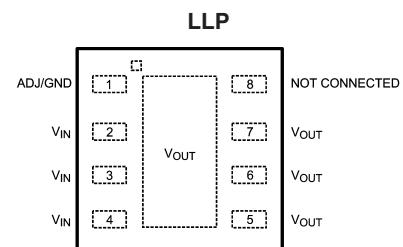


Top View



10091945

Side View



When using the LLP package
Pins 2, 3 & 4 must be connected together and
Pins 5, 6 & 7 must be connected together

Top View

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Maximum Input Voltage (V_{IN} to GND)	20V
Power Dissipation (Note 2)	Internally Limited
Junction Temperature (T_J) (Note 2)	150°C
Storage Temperature Range	-65°C to 150°C
Lead Temperature	

TO-220 (T) Package

260°C, 10 sec

SOT-223 (IMP) Package

260°C, 4 sec

ESD Tolerance (Note 3)

2000V

Operating Ratings (Note 1)

Input Voltage (V_{IN} to GND)	15V
Junction Temperature Range (T_J) (Note 2)	
LM1117	0°C to 125°C
LM1117I	-40°C to 125°C

LM1117 Electrical Characteristics

Typicals and limits appearing in normal type apply for $T_J = 25^\circ\text{C}$. Limits appearing in **Boldface** type apply over the entire junction temperature range for operation, 0°C to 125°C.

Symbol	Parameter	Conditions	Min (Note 5)	Typ (Note 4)	Max (Note 5)	Units
V_{REF}	Reference Voltage	LM1117-ADJ $I_{OUT} = 10\text{mA}$, $V_{IN}-V_{OUT} = 2\text{V}$, $T_J = 25^\circ\text{C}$ $10\text{mA} \leq I_{OUT} \leq 800\text{mA}$, $1.4\text{V} \leq V_{IN}-V_{OUT} \leq 10\text{V}$	1.238 1.225	1.250 1.250	1.262 1.270	V V
V_{OUT}	Output Voltage	LM1117-1.8 $I_{OUT} = 10\text{mA}$, $V_{IN} = 3.8\text{V}$, $T_J = 25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$, $3.2\text{V} \leq V_{IN} \leq 10\text{V}$	1.782 1.746	1.800 1.800	1.818 1.854	V V
		LM1117-2.5 $I_{OUT} = 10\text{mA}$, $V_{IN} = 4.5\text{V}$, $T_J = 25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$, $3.9\text{V} \leq V_{IN} \leq 10\text{V}$	2.475 2.450	2.500 2.500	2.525 2.550	V V
		LM1117-2.85 $I_{OUT} = 10\text{mA}$, $V_{IN} = 4.85\text{V}$, $T_J = 25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$, $4.25\text{V} \leq V_{IN} \leq 10\text{V}$ $0 \leq I_{OUT} \leq 500\text{mA}$, $V_{IN} = 4.10\text{V}$	2.820 2.790 2.790	2.850 2.850 2.850	2.880 2.910 2.910	V V V
		LM1117-3.3 $I_{OUT} = 10\text{mA}$, $V_{IN} = 5\text{V}$, $T_J = 25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$, $4.75\text{V} \leq V_{IN} \leq 10\text{V}$	3.267 3.235	3.300 3.300	3.333 3.365	V V
		LM1117-5.0 $I_{OUT} = 10\text{mA}$, $V_{IN} = 7\text{V}$, $T_J = 25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$, $6.5\text{V} \leq V_{IN} \leq 12\text{V}$	4.950 4.900	5.000 5.000	5.050 5.100	V V
ΔV_{OUT}	Line Regulation (Note 6)	LM1117-ADJ $I_{OUT} = 10\text{mA}$, $1.5\text{V} \leq V_{IN}-V_{OUT} \leq 13.75\text{V}$		0.035	0.2	%
		LM1117-1.8 $I_{OUT} = 0\text{mA}$, $3.2\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV
		LM1117-2.5 $I_{OUT} = 0\text{mA}$, $3.9\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV
		LM1117-2.85 $I_{OUT} = 0\text{mA}$, $4.25\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV
		LM1117-3.3 $I_{OUT} = 0\text{mA}$, $4.75\text{V} \leq V_{IN} \leq 15\text{V}$		1	6	mV
		LM1117-5.0 $I_{OUT} = 0\text{mA}$, $6.5\text{V} \leq V_{IN} \leq 15\text{V}$		1	10	mV

LM1117 Electrical Characteristics (Continued)

Typicals and limits appearing in normal type apply for $T_J = 25^\circ\text{C}$. Limits appearing in **Boldface** type apply over the entire junction temperature range for operation, 0°C to 125°C .

Symbol	Parameter	Conditions	Min (Note 5)	Typ (Note 4)	Max (Note 5)	Units
ΔV_{OUT}	Load Regulation (Note 6)	LM1117-ADJ $V_{\text{IN}} - V_{\text{OUT}} = 3\text{V}$, $10 \leq I_{\text{OUT}} \leq 800\text{mA}$		0.2	0.4	%
		LM1117-1.8 $V_{\text{IN}} = 3.2\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-2.5 $V_{\text{IN}} = 3.9\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-2.85 $V_{\text{IN}} = 4.25\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-3.3 $V_{\text{IN}} = 4.75\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-5.0 $V_{\text{IN}} = 6.5\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	15	mV
		$I_{\text{OUT}} = 100\text{mA}$		1.10	1.20	V
$V_{\text{IN}} - V_{\text{OUT}}$	Dropout Voltage (Note 7)	$I_{\text{OUT}} = 500\text{mA}$		1.15	1.25	V
		$I_{\text{OUT}} = 800\text{mA}$		1.20	1.30	V
I_{LIMIT}	Current Limit	$V_{\text{IN}} - V_{\text{OUT}} = 5\text{V}$, $T_J = 25^\circ\text{C}$	800	1200	1500	mA
	Minimum Load Current (Note 8)	LM1117-ADJ $V_{\text{IN}} = 15\text{V}$		1.7	5	mA
	Quiescent Current	LM1117-1.8 $V_{\text{IN}} \leq 15\text{V}$		5	10	mA
		LM1117-2.5 $V_{\text{IN}} \leq 15\text{V}$		5	10	mA
		LM1117-2.85 $V_{\text{IN}} \leq 10\text{V}$		5	10	mA
		LM1117-3.3 $V_{\text{IN}} \leq 15\text{V}$		5	10	mA
		LM1117-5.0 $V_{\text{IN}} \leq 15\text{V}$		5	10	mA
	Thermal Regulation	$T_A = 25^\circ\text{C}$, 30ms Pulse		0.01	0.1	%/W
	Ripple Regulation	$f_{\text{RIPPLE}} = 1\text{ }20\text{Hz}$, $V_{\text{IN}} - V_{\text{OUT}} = 3\text{V}$ $V_{\text{RIPPLE}} = 1\text{V}_{\text{PP}}$	60	75		dB
	Adjust Pin Current			60	120	μA
	Adjust Pin Current Change	$10 \leq I_{\text{OUT}} \leq 800\text{mA}$, $1.4\text{V} \leq V_{\text{IN}} - V_{\text{OUT}} \leq 10\text{V}$		0.2	5	μA
	Temperature Stability			0.5		%
	Long Term Stability	$T_A = 125^\circ\text{C}$, 1000Hrs		0.3		%
	RMS Output Noise	(% of V_{OUT}), $10\text{Hz} \leq f \leq 10\text{kHz}$		0.003		%
	Thermal Resistance Junction-to-Case	3-Lead SOT-223		15.0		$^\circ\text{C}/\text{W}$
		3-Lead TO-220		3.0		$^\circ\text{C}/\text{W}$
		3-Lead TO-252		10		$^\circ\text{C}/\text{W}$
	Thermal Resistance Junction-to-Ambient (No air flow)	3-Lead SOT-223 (No heat sink)		136		$^\circ\text{C}/\text{W}$
		3-Lead TO-220 (No heat sink)		79		$^\circ\text{C}/\text{W}$
		3-Lead TO-252 (Note 9) (No heat sink)		92		$^\circ\text{C}/\text{W}$
		3-Lead TO-263		55		$^\circ\text{C}/\text{W}$
		8-Lead LLP (Note 10)		40		$^\circ\text{C}/\text{W}$

LM1117I Electrical Characteristics

Typicals and limits appearing in normal type apply for $T_J = 25^\circ\text{C}$. Limits appearing in **Boldface** type apply over the entire junction temperature range for operation, -40°C to 125°C .

Symbol	Parameter	Conditions	Min (Note 5)	Typ (Note 4)	Max (Note 5)	Units
V_{REF}	Reference Voltage	LM1117I-ADJ $I_{\text{OUT}} = 10\text{mA}$, $V_{\text{IN}} - V_{\text{OUT}} = 2\text{V}$, $T_J = 25^\circ\text{C}$ $10\text{mA} \leq I_{\text{OUT}} \leq 800\text{mA}$, $1.4\text{V} \leq V_{\text{IN}} - V_{\text{OUT}} \leq 10\text{V}$	1.238 1.200	1.250 1.250	1.262 1.290	V V
V_{OUT}	Output Voltage	LM1117I-3.3 $I_{\text{OUT}} = 10\text{mA}$, $V_{\text{IN}} = 5\text{V}$, $T_J = 25^\circ\text{C}$ $0 \leq I_{\text{OUT}} \leq 800\text{mA}$, $4.75\text{V} \leq V_{\text{IN}} \leq 10\text{V}$	3.267 3.168	3.300 3.300	3.333 3.432	V V
		LM1117I-5.0 $I_{\text{OUT}} = 10\text{mA}$, $V_{\text{IN}} = 7\text{V}$, $T_J = 25^\circ\text{C}$ $0 \leq I_{\text{OUT}} \leq 800\text{mA}$, $6.5\text{V} \leq V_{\text{IN}} \leq 12\text{V}$	4.950 4.800	5.000 5.000	5.050 5.200	V V
ΔV_{OUT}	Line Regulation (Note 6)	LM1117I-ADJ $I_{\text{OUT}} = 10\text{mA}$, $1.5\text{V} \leq V_{\text{IN}} - V_{\text{OUT}} \leq 13.75\text{V}$		0.035	0.3	%
		LM1117I-3.3 $I_{\text{OUT}} = 0\text{mA}$, $4.75\text{V} \leq V_{\text{IN}} \leq 15\text{V}$		1	10	mV
		LM1117I-5.0 $I_{\text{OUT}} = 0\text{mA}$, $6.5\text{V} \leq V_{\text{IN}} \leq 15\text{V}$		1	15	mV
ΔV_{OUT}	Load Regulation (Note 6)	LM1117I-ADJ $V_{\text{IN}} - V_{\text{OUT}} = 3\text{V}$, $10 \leq I_{\text{OUT}} \leq 800\text{mA}$		0.2	0.5	%
		LM1117I-3.3 $V_{\text{IN}} = 4.75\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	15	mV
		LM1117I-5.0 $V_{\text{IN}} = 6.5\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	20	mV
$V_{\text{IN}} - V_{\text{OUT}}$	Dropout Voltage (Note 7)	$I_{\text{OUT}} = 100\text{mA}$		1.10	1.30	V
		$I_{\text{OUT}} = 500\text{mA}$		1.15	1.35	V
		$I_{\text{OUT}} = 800\text{mA}$		1.20	1.40	V
I_{LIMIT}	Current Limit	$V_{\text{IN}} - V_{\text{OUT}} = 5\text{V}$, $T_J = 25^\circ\text{C}$	800	1200	1500	mA
	Minimum Load Current (Note 8)	LM1117I-ADJ $V_{\text{IN}} = 15\text{V}$		1.7	5	mA
	Quiescent Current	LM1117I-3.3 $V_{\text{IN}} \leq 15\text{V}$		5	15	mA
		LM1117I-5.0 $V_{\text{IN}} \leq 15\text{V}$		5	15	mA
	Thermal Regulation	$T_A = 25^\circ\text{C}$, 30ms Pulse		0.01	0.1	%/W
	Ripple Regulation	$f_{\text{RIPPLE}} = 1\text{ }20\text{Hz}$, $V_{\text{IN}} - V_{\text{OUT}} = 3\text{V}$ $V_{\text{RIPPLE}} = 1\text{V}_{\text{PP}}$	60	75		dB
	Adjust Pin Current			60	120	μA
	Adjust Pin Current Change	$10 \leq I_{\text{OUT}} \leq 800\text{mA}$, $1.4\text{V} \leq V_{\text{IN}} - V_{\text{OUT}} \leq 10\text{V}$		0.2	10	μA
	Temperature Stability			0.5		%
	Long Term Stability	$T_A = 125^\circ\text{C}$, 1000Hrs		0.3		%
RMS Output Noise	Thermal Resistance Junction-to-Case	(% of V_{OUT}), $10\text{Hz} \leq f \leq 10\text{kHz}$		0.003		%
		3-Lead SOT-223		15.0		$^\circ\text{C}/\text{W}$
	Thermal Resistance Junction-to-Ambient No air flow)	3-Lead TO-252		10		$^\circ\text{C}/\text{W}$
		3-Lead SOT-223 (No heat sink)		136		$^\circ\text{C}/\text{W}$
		3-Lead TO-252 (No heat sink)(Note 9)		92		$^\circ\text{C}/\text{W}$
	8-Lead LLP(Note 10)			40		$^\circ\text{C}/\text{W}$

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but specific performance is not guaranteed. For guaranteed specifications and the test conditions, see the Electrical Characteristics.

Note 2: The maximum power dissipation is a function of $T_{J(\max)}$, θ_{JA} , and T_A . The maximum allowable power dissipation at any ambient temperature is $P_D = (T_{J(\max)} - T_A)/\theta_{JA}$. All numbers apply for packages soldered directly into a PC board.

Note 3: For testing purposes, ESD was applied using human body model, $1.5k\Omega$ in series with $100pF$.

Note 4: Typical Values represent the most likely parametric norm.

Note 5: All limits are guaranteed by testing or statistical analysis.

Note 6: Load and line regulation are measured at constant junction room temperature.

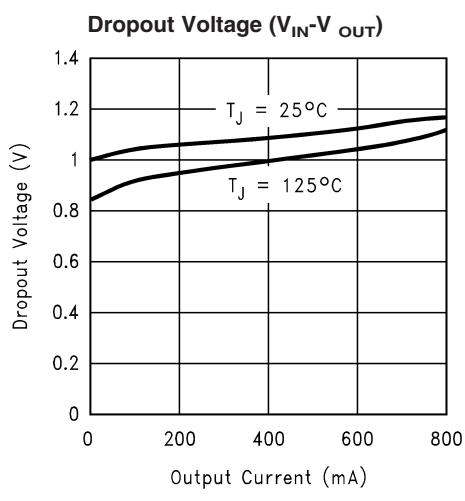
Note 7: The dropout voltage is the input/output differential at which the circuit ceases to regulate against further reduction in input voltage. It is measured when the output voltage has dropped $100mV$ from the nominal value obtained at $V_{IN} = V_{OUT} + 1.5V$.

Note 8: The minimum output current required to maintain regulation.

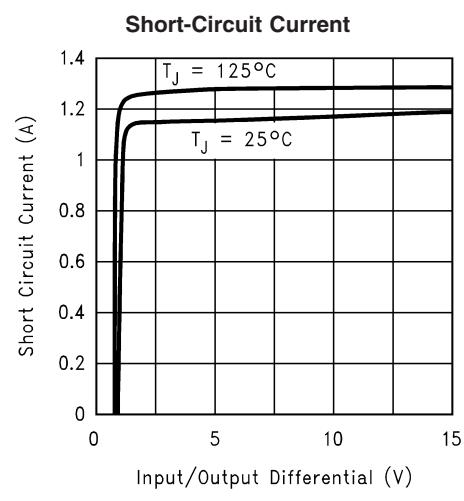
Note 9: Minimum pad size of $0.038in^2$

Note 10: Thermal Performance for the LLP was obtained using JESD51-7 board with six vias and an ambient temperature of $22^\circ C$. For information about improved thermal performance and power dissipation for the LLP, refer to Application Note AN-1187.

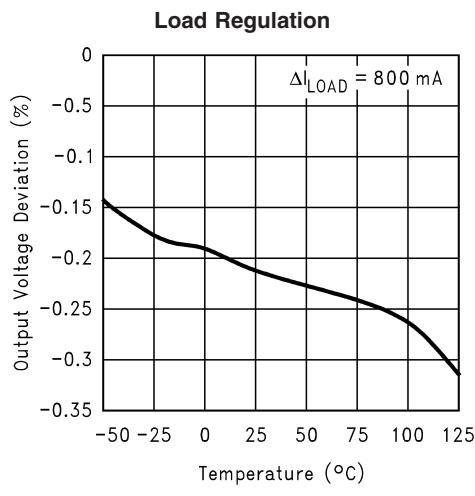
Typical Performance Characteristics



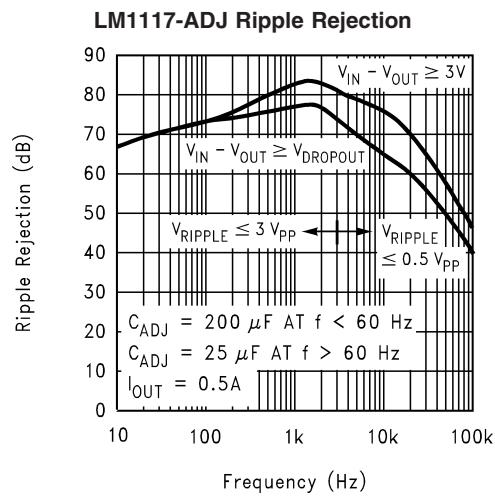
10091922



10091923



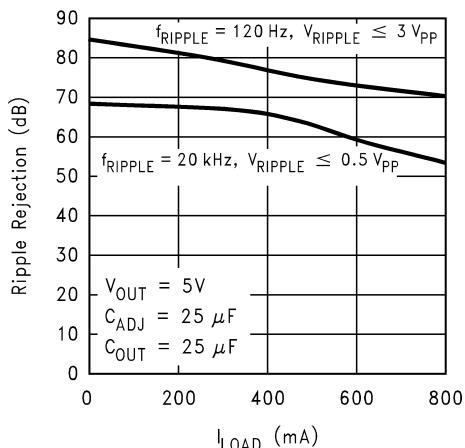
10091943



10091906

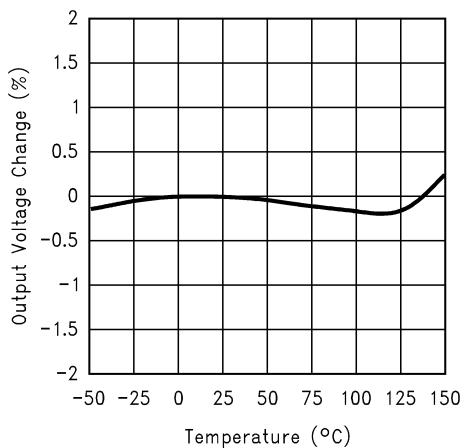
Typical Performance Characteristics (Continued)

LM1117-ADJ Ripple Rejection vs. Current



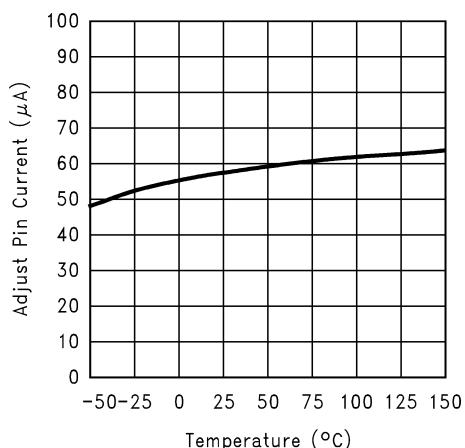
10091907

Temperature Stability



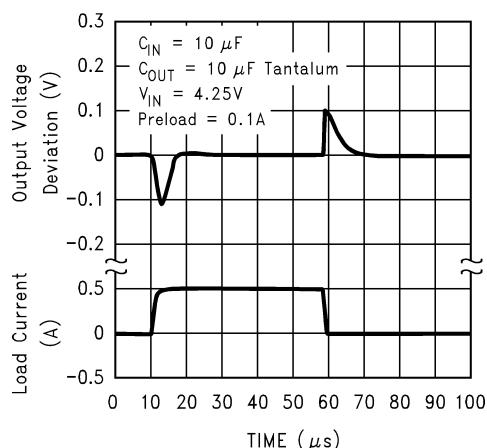
10091925

Adjust Pin Current



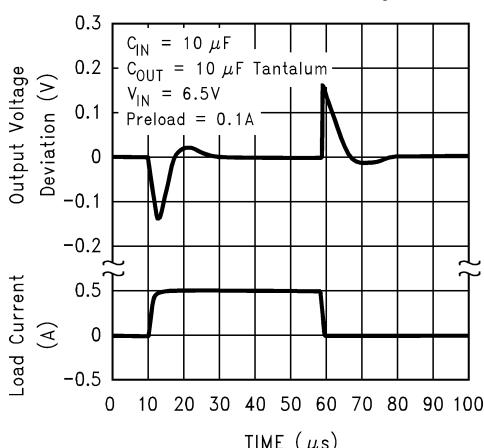
10091926

LM1117-2.85 Load Transient Response



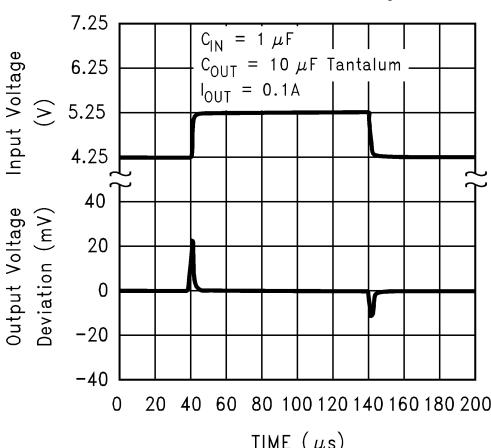
10091908

LM1117-5.0 Load Transient Response



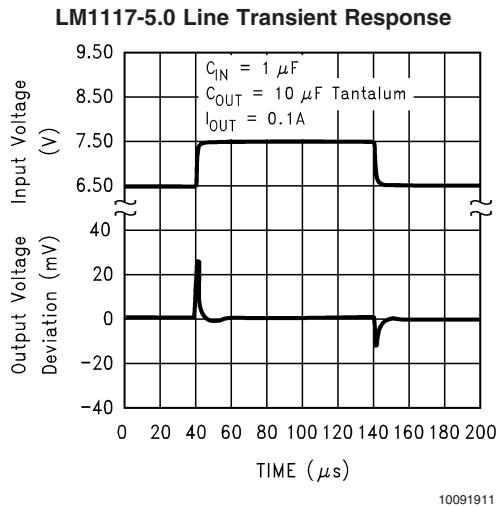
10091909

LM1117-2.85 Line Transient Response



10091910

Typical Performance Characteristics (Continued)



Application Note

1.0 EXTERNAL CAPACITORS/STABILITY

1.1 Input Bypass Capacitor

An input capacitor is recommended. A 10 μF tantalum on the input is a suitable input bypassing for almost all applications.

1.2 Adjust Terminal Bypass Capacitor

The adjust terminal can be bypassed to ground with a bypass capacitor (C_{ADJ}) to improve ripple rejection. This bypass capacitor prevents ripple from being amplified as the output voltage is increased. At any ripple frequency, the impedance of the C_{ADJ} should be less than R1 to prevent the ripple from being amplified:

$$1/(2\pi f_{RIPPLE} C_{ADJ}) < R1$$

The R1 is the resistor between the output and the adjust pin. Its value is normally in the range of 100-200 Ω . For example, with $R1 = 124\Omega$ and $f_{RIPPLE} = 120Hz$, the C_{ADJ} should be $> 11\mu F$.

1.3 Output Capacitor

The output capacitor is critical in maintaining regulator stability, and must meet the required conditions for both minimum amount of capacitance and ESR (Equivalent Series Resistance). The minimum output capacitance required by the LM1117 is 10 μF , if a tantalum capacitor is used. Any increase of the output capacitance will merely improve the loop stability and transient response. The ESR of the output capacitor should range between 0.3 Ω - 22 Ω . In the case of the adjustable regulator, when the C_{ADJ} is used, a larger output capacitance (22 μF tantalum) is required.

2.0 OUTPUT VOLTAGE

The LM1117 adjustable version develops a 1.25V reference voltage, V_{REF} , between the output and the adjust terminal. As shown in Figure 1, this voltage is applied across resistor R1 to generate a constant current $I1$. The current I_{ADJ} from the adjust terminal could introduce error to the output. But since it is very small (60 μA) compared with the $I1$ and very constant with line and load changes, the error can be ig-

nored. The constant current $I1$ then flows through the output set resistor R2 and sets the output voltage to the desired level.

For fixed voltage devices, R1 and R2 are integrated inside the devices.

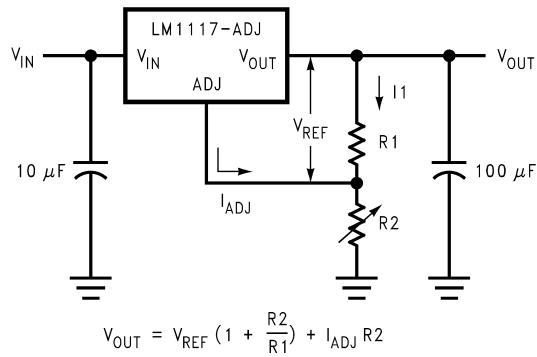


FIGURE 1. Basic Adjustable Regulator

3.0 LOAD REGULATION

The LM1117 regulates the voltage that appears between its output and ground pins, or between its output and adjust pins. In some cases, line resistances can introduce errors to the voltage across the load. To obtain the best load regulation, a few precautions are needed.

Figure 2, shows a typical application using a fixed output regulator. The R_{L1} and R_{L2} are the line resistances. It is obvious that the V_{LOAD} is less than the V_{OUT} by the sum of the voltage drops along the line resistances. In this case, the load regulation seen at the R_{LOAD} would be degraded from the data sheet specification. To improve this, the load should be tied directly to the output terminal on the positive side and directly tied to the ground terminal on the negative side.

Application Note (Continued)

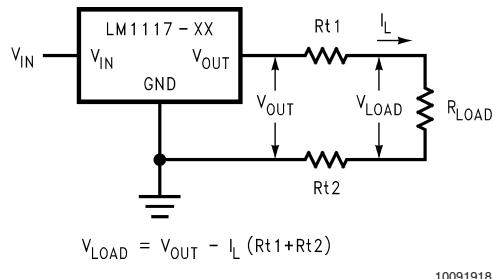


FIGURE 2. Typical Application using Fixed Output Regulator

When the adjustable regulator is used (Figure 3), the best performance is obtained with the positive side of the resistor R1 tied directly to the output terminal of the regulator rather than near the load. This eliminates line drops from appearing effectively in series with the reference and degrading regulation. For example, a 5V regulator with 0.05Ω resistance between the regulator and load will have a load regulation due to line resistance of $0.05\Omega \times I_L$. If R1 ($=125\Omega$) is connected near the load, the effective line resistance will be $0.05\Omega (1+R2/R1)$ or in this case, it is 4 times worse. In addition, the ground side of the resistor R2 can be returned near the ground of the load to provide remote ground sensing and improve load regulation.

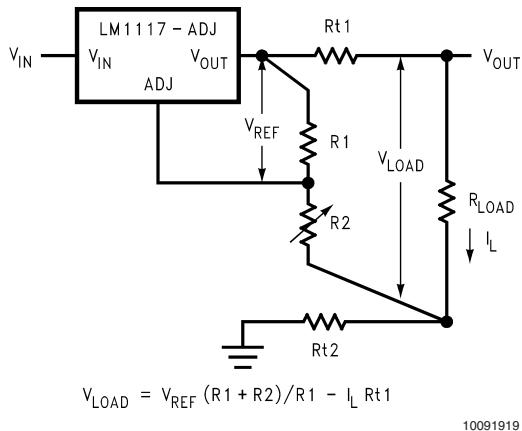


FIGURE 3. Best Load Regulation using Adjustable Output Regulator

4.0 PROTECTION DIODES

Under normal operation, the LM1117 regulators do not need any protection diode. With the adjustable device, the internal resistance between the adjust and output terminals limits the current. No diode is needed to divert the current around the regulator even with capacitor on the adjust terminal. The adjust pin can take a transient signal of $\pm 25V$ with respect to the output voltage without damaging the device.

When a output capacitor is connected to a regulator and the input is shorted to ground, the output capacitor will discharge into the output of the regulator. The discharge current depends on the value of the capacitor, the output voltage of the regulator, and rate of decrease of V_{IN} . In the LM1117 regulators, the internal diode between the output and input pins

can withstand microsecond surge currents of 10A to 20A. With an extremely large output capacitor ($\geq 1000 \mu F$), and with input instantaneously shorted to ground, the regulator could be damaged.

In this case, an external diode is recommended between the output and input pins to protect the regulator, as shown in Figure 4.

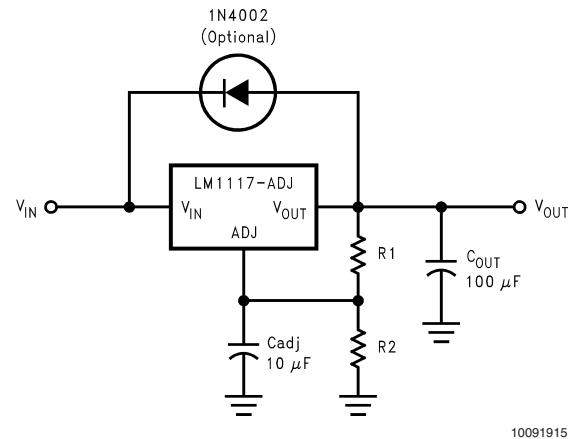


FIGURE 4. Regulator with Protection Diode

5.0 HEATSINK REQUIREMENTS

When an integrated circuit operates with an appreciable current, its junction temperature is elevated. It is important to quantify its thermal limits in order to achieve acceptable performance and reliability. This limit is determined by summing the individual parts consisting of a series of temperature rises from the semiconductor junction to the operating environment. A one-dimensional steady-state model of conduction heat transfer is demonstrated in Figure 5. The heat generated at the device junction flows through the die to the die attach pad, through the lead frame to the surrounding case material, to the printed circuit board, and eventually to the ambient environment. Below is a list of variables that may affect the thermal resistance and in turn the need for a heatsink.

R^0 JC (Component Variables)	R^0 CA (Application Variables)
Leadframe Size & Material	Mounting Pad Size, Material, & Location
No. of Conduction Pins	Placement of Mounting Pad
Die Size	PCB Size & Material
Die Attach Material	Traces Length & Width
Molding Compound Size and Material	Adjacent Heat Sources
	Volume of Air
	Ambient Temperature
	Shape of Mounting Pad

Application Note (Continued)

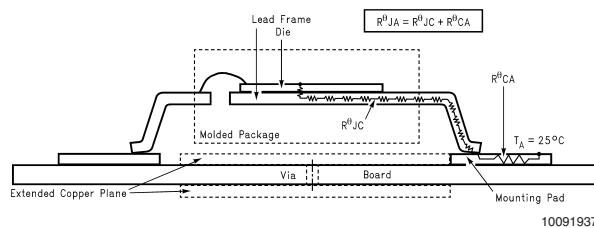


FIGURE 5. Cross-sectional view of Integrated Circuit Mounted on a printed circuit board. Note that the case temperature is measured at the point where the leads contact with the mounting pad surface

The LM1117 regulators have internal thermal shutdown to protect the device from over-heating. Under all possible operating conditions, the junction temperature of the LM1117 must be within the range of 0°C to 125°C . A heatsink may be required depending on the maximum power dissipation and maximum ambient temperature of the application. To determine if a heatsink is needed, the power dissipated by the regulator, P_D , must be calculated:

$$I_{IN} = I_L + I_G$$

$$P_D = (V_{IN} - V_{OUT})I_L + V_{IN}I_G$$

Figure 6 shows the voltages and currents which are present in the circuit.

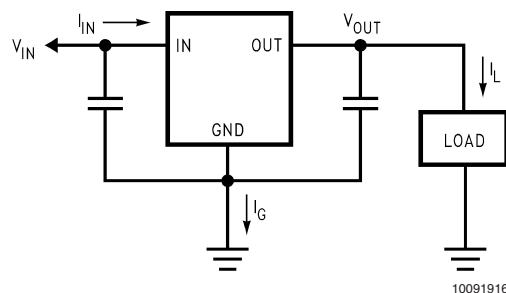


FIGURE 6. Power Dissipation Diagram

TABLE 1. θ_{JA} Different Heatsink Area

Layout	Copper Area		Thermal Resistance	
	Top Side (in^2)*	Bottom Side (in^2)	(θ_{JA} , $^{\circ}\text{C/W}$) SOT-223	(θ_{JA} , $^{\circ}\text{C/W}$) TO-252
1	0.0123	0	136	103
2	0.066	0	123	87
3	0.3	0	84	60
4	0.53	0	75	54
5	0.76	0	69	52
6	1	0	66	47
7	0	0.2	115	84
8	0	0.4	98	70
9	0	0.6	89	63
10	0	0.8	82	57
11	0	1	79	57
12	0.066	0.066	125	89
13	0.175	0.175	93	72

Application Note (Continued)**TABLE 1. θ_{JA} Different Heatsink Area (Continued)**

Layout	Copper Area		Thermal Resistance	
14	0.284	0.284	83	61
15	0.392	0.392	75	55
16	0.5	0.5	70	53

*Tab of device attached to topside copper

Application Note (Continued)

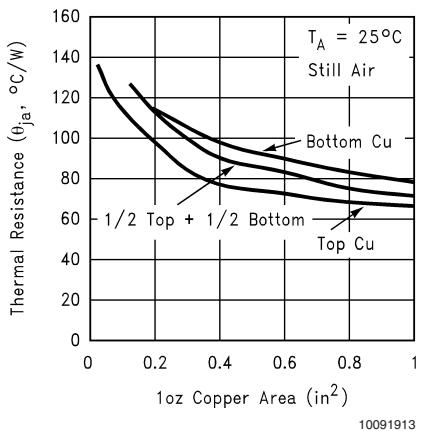


FIGURE 7. θ_{JA} vs. 1oz Copper Area for SOT-223

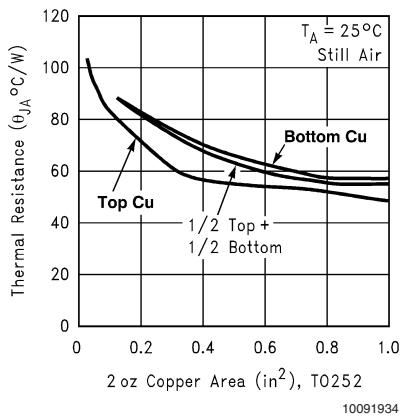


FIGURE 8. θ_{JA} vs. 2oz Copper Area for TO-252

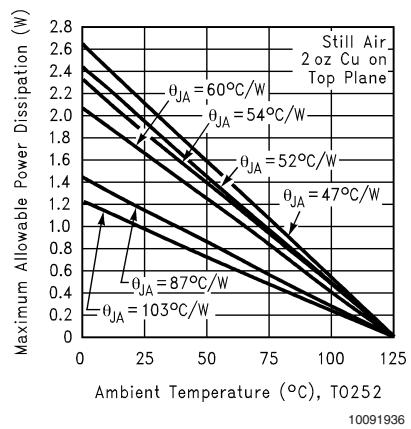


FIGURE 10. Maximum Allowable Power Dissipation vs. Ambient Temperature for TO-252

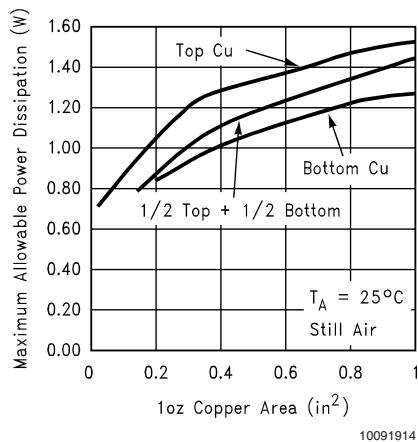


FIGURE 11. Maximum Allowable Power Dissipation vs. 1oz Copper Area for SOT-223

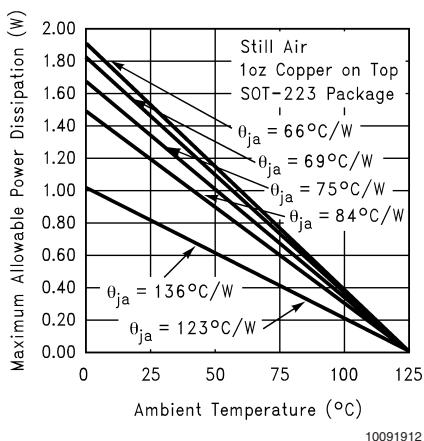


FIGURE 9. Maximum Allowable Power Dissipation vs. Ambient Temperature for SOT-223

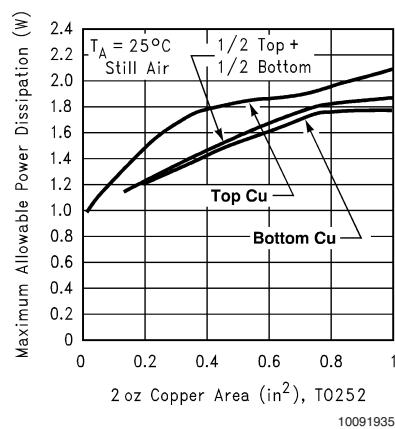
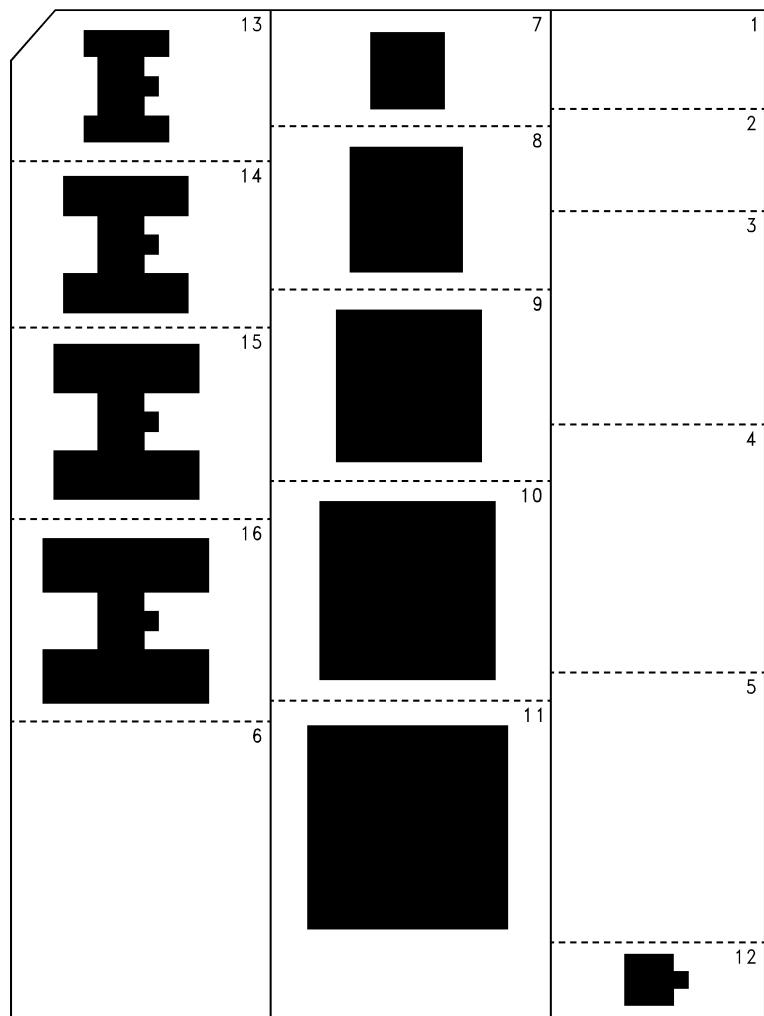
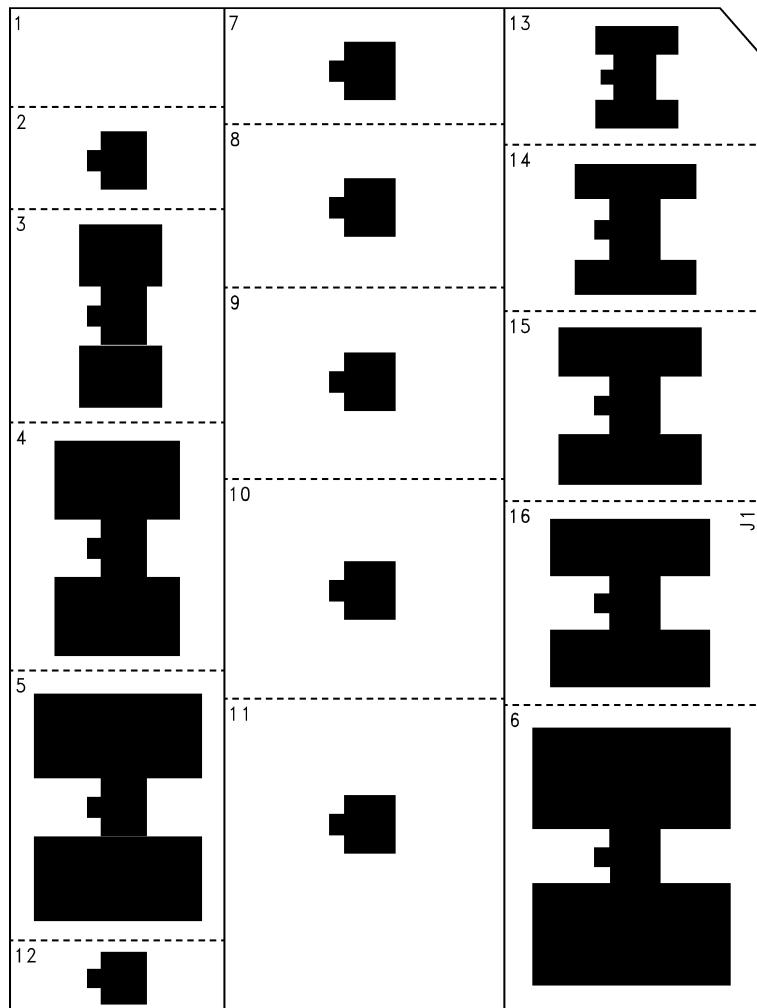


FIGURE 12. Maximum Allowable Power Dissipation vs. 2oz Copper Area for TO-252

Application Note (Continued)

10091941

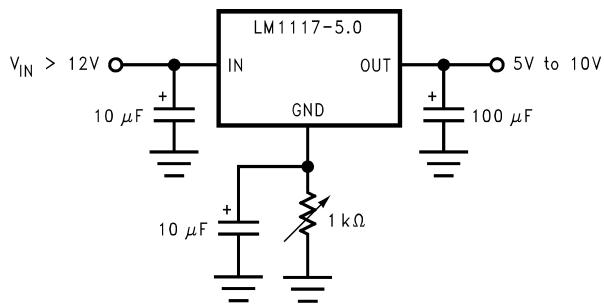
FIGURE 13. Top View of the Thermal Test Pattern in Actual Scale

Application Note (Continued)

10091942

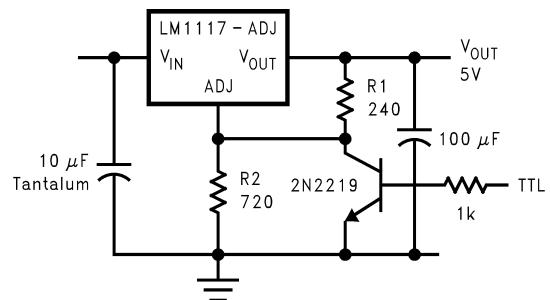
FIGURE 14. Bottom View of the Thermal Test Pattern in Actual Scale

Typical Application Circuits



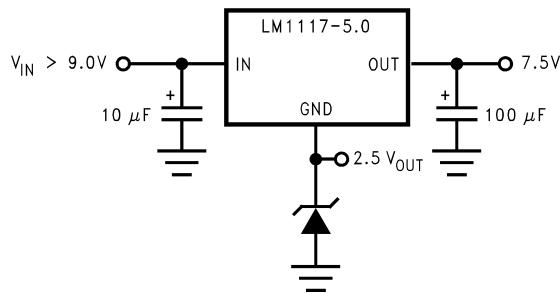
Adjusting Output of Fixed Regulators

10091930

* Min. output $\approx 1.25V$

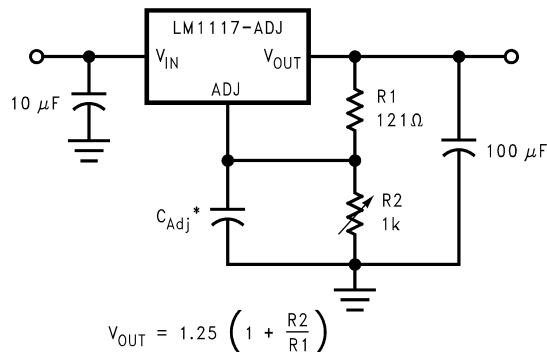
10091927

5V Logic Regulator with Electronic Shutdown*



Regulator with Reference

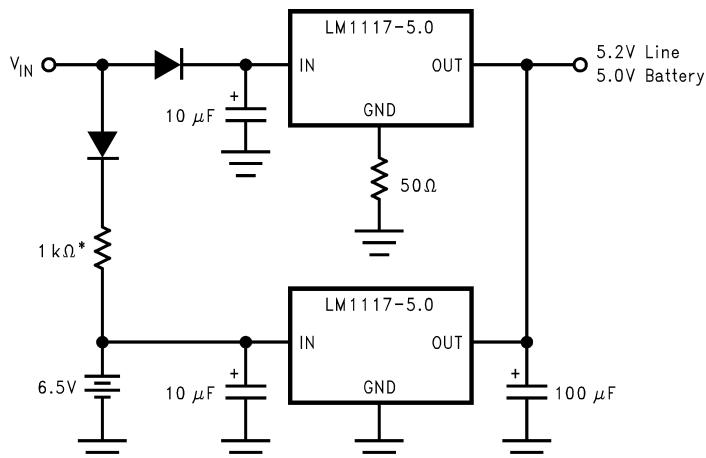
10091931

* C_{Adj} is optional, however it will improve ripple rejection.

10091929

1.25V to 10V Adjustable Regulator with Improved Ripple Rejection

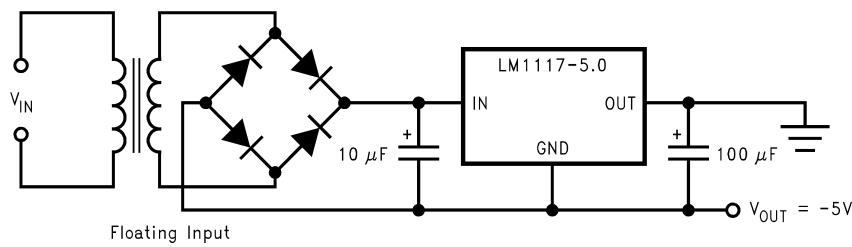
Typical Application Circuits (Continued)



* Select for charge rate.

10091932

Battery Backed-Up Regulated Supply

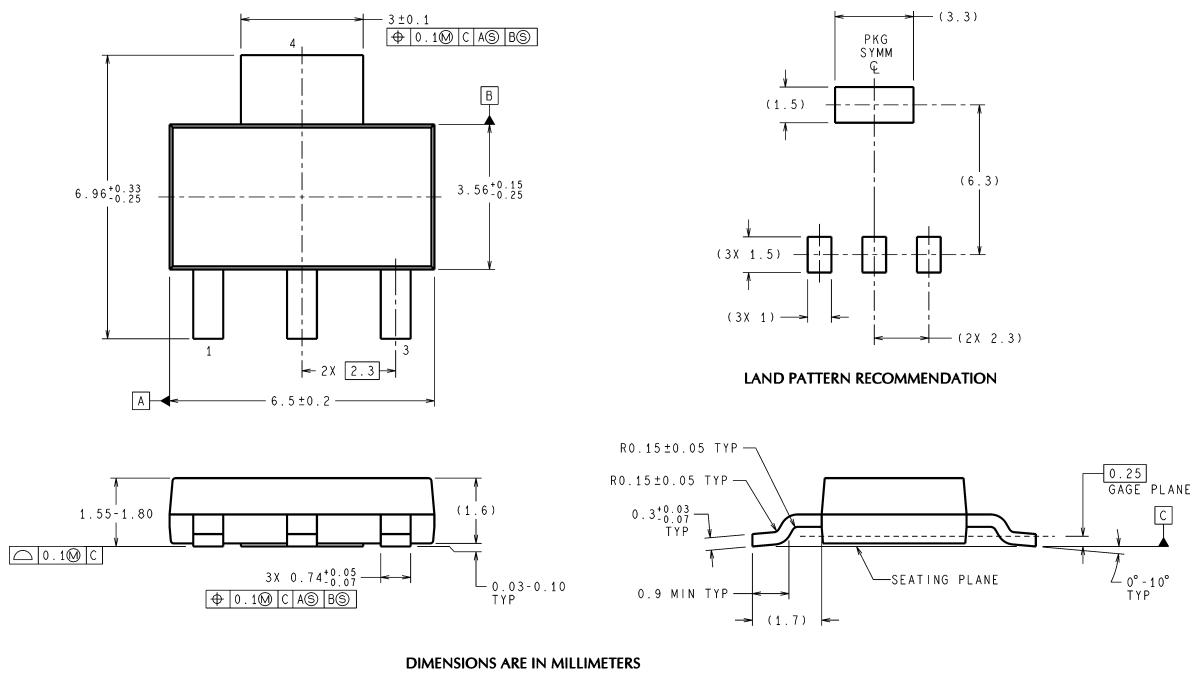


10091933

Low Dropout Negative Supply

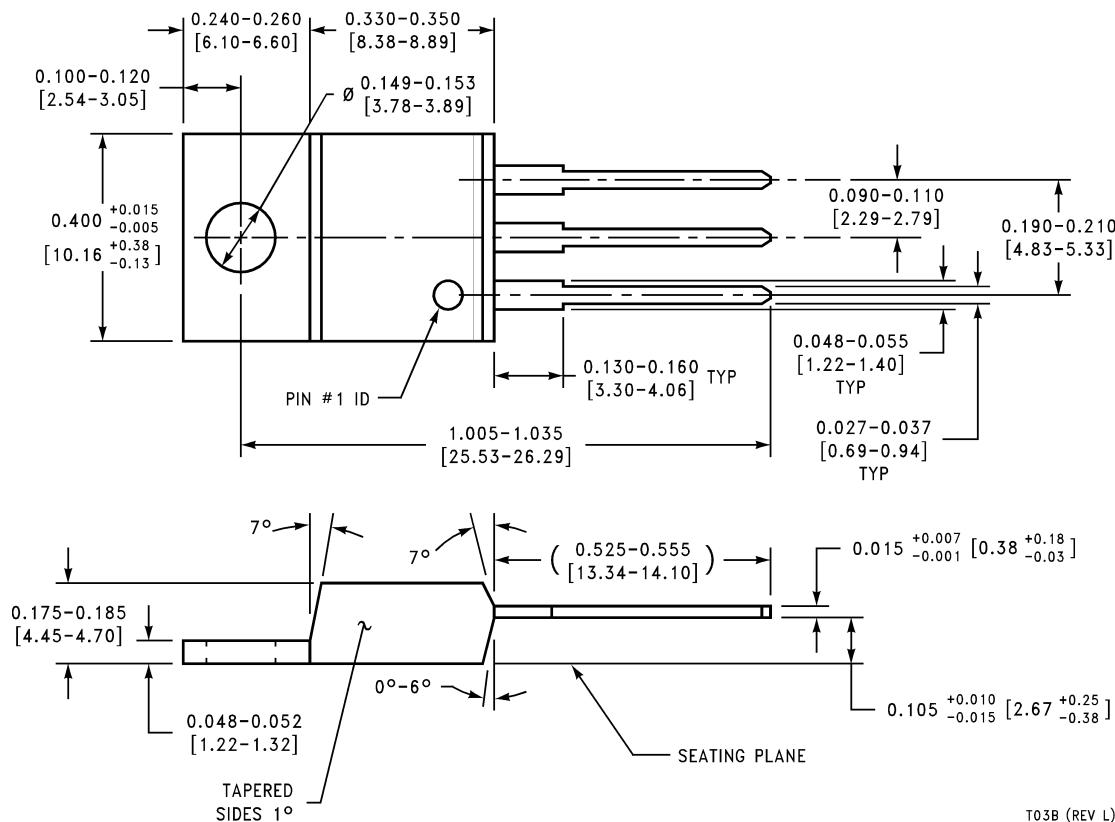
Physical Dimensions

inches (millimeters) unless otherwise noted



MP04A (Rev B)

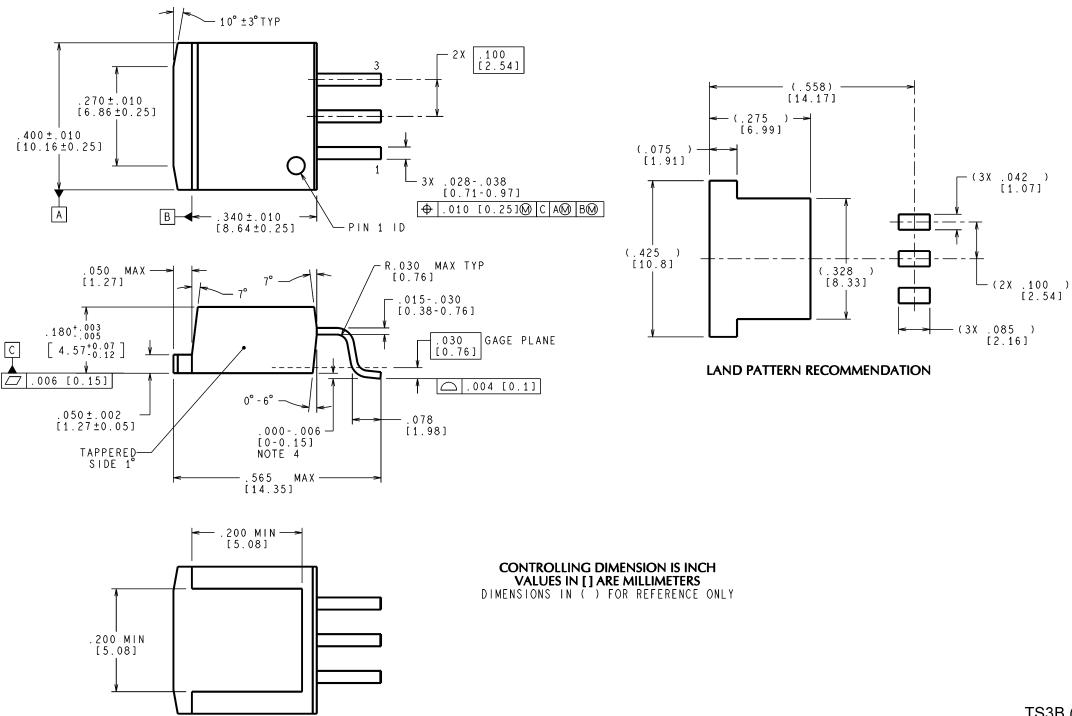
**3-Lead SOT-223
NS Package Number MP04A**



T03B (REV L)

**3-Lead TO-220
NS Package Number T03B**

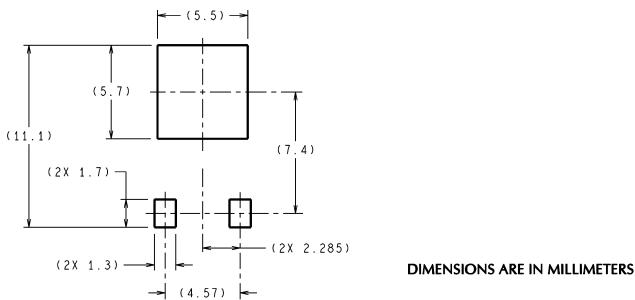
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



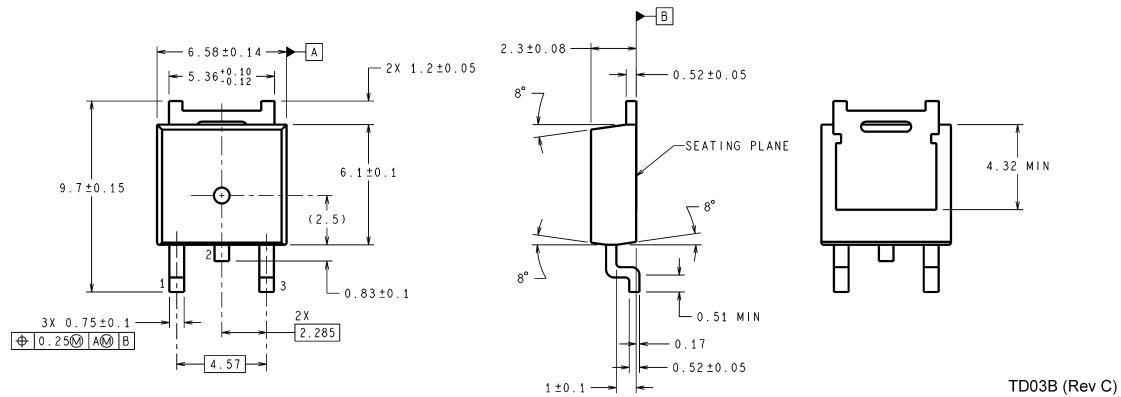
TS3B (Rev F)

**3-Lead TO-263
NS Package Number TS3B**

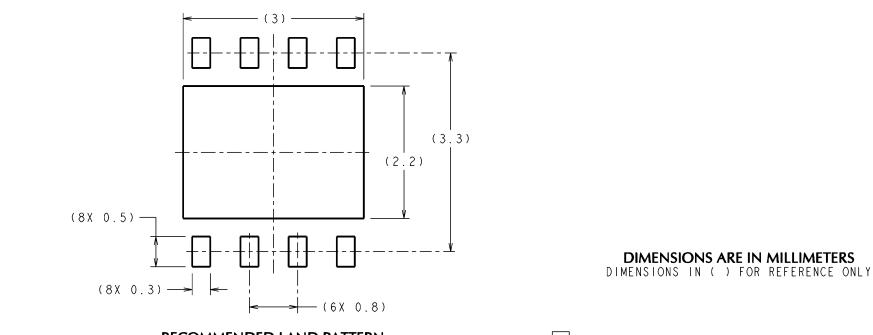
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



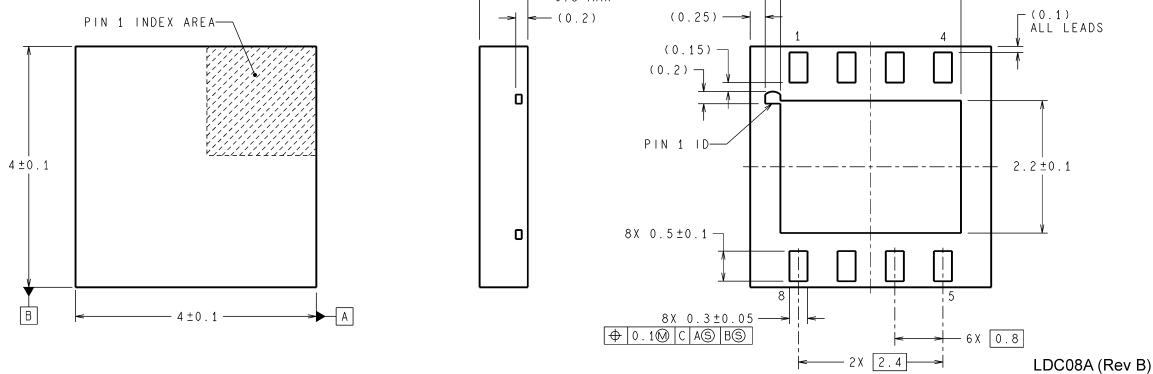
LAND PATTERN RECOMMENDATION



**3-Lead TO-252
NS Package Number TD03B**



RECOMMENDED LAND PATTERN



**8-Lead LLP
NS Package Number LDC08A**

Notes

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

For the most current product information visit us at www.national.com.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

BANNED SUBSTANCE COMPLIANCE

National Semiconductor manufactures products and uses packing materials that meet the provisions of the Customer Products Stewardship Specification (CSP-9-111C2) and the Banned Substances and Materials of Interest Specification (CSP-9-111S2) and contain no "Banned Substances" as defined in CSP-9-111S2.

Leadfree products are RoHS compliant.

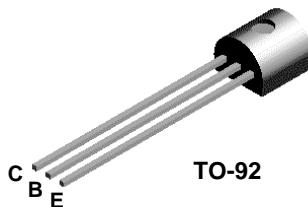
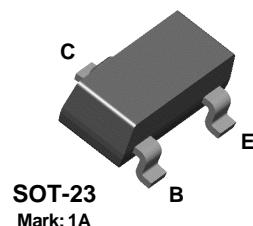
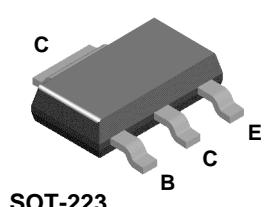


National Semiconductor
Americas Customer
Support Center
Email: new.feedback@nsc.com
Tel: 1-800-272-9959
www.national.com

National Semiconductor
Europe Customer Support Center
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor
Asia Pacific Customer
Support Center
Email: ap.support@nsc.com

National Semiconductor
Japan Customer Support Center
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com
Tel: 81-3-5639-7560

**2N3904****MMBT3904****PZT3904**

NPN General Purpose Amplifier

This device is designed as a general purpose amplifier and switch.
The useful dynamic range extends to 100 mA as a switch and to
100 MHz as an amplifier.

Absolute Maximum Ratings*

$T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	60	V
V_{EBO}	Emitter-Base Voltage	6.0	V
I_C	Collector Current - Continuous	200	mA
T_J, T_{stg}	Operating and Storage Junction Temperature Range	-55 to +150	°C

* These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:

- 1) These ratings are based on a maximum junction temperature of 150 degrees C.
- 2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Thermal Characteristics

$T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Characteristic	Max			Units
		2N3904	*MMBT3904	**PZT3904	
P_D	Total Device Dissipation Derate above 25°C	625 5.0	350 2.8	1,000 8.0	mW mW/°C
$R_{\theta JC}$	Thermal Resistance, Junction to Case	83.3			°C/W
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	200	357	125	°C/W

* Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06."

** Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6 cm².

NPN General Purpose Amplifier

(continued)

Electrical Characteristics

$T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Conditions	Min	Max	Units
OFF CHARACTERISTICS					
$V_{(\text{BR})\text{CEO}}$	Collector-Emitter Breakdown Voltage	$I_C = 1.0 \text{ mA}, I_B = 0$	40		V
$V_{(\text{BR})\text{CBO}}$	Collector-Base Breakdown Voltage	$I_C = 10 \mu\text{A}, I_E = 0$	60		V
$V_{(\text{BR})\text{EBO}}$	Emitter-Base Breakdown Voltage	$I_E = 10 \mu\text{A}, I_C = 0$	6.0		V
I_{BL}	Base Cutoff Current	$V_{CE} = 30 \text{ V}, V_{EB} = 3\text{V}$		50	nA
I_{CEX}	Collector Cutoff Current	$V_{CE} = 30 \text{ V}, V_{EB} = 3\text{V}$		50	nA

ON CHARACTERISTICS*

h_{FE}	DC Current Gain	$I_C = 0.1 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 1.0 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 10 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 50 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 100 \text{ mA}, V_{CE} = 1.0 \text{ V}$	40 70 100 60 30	300	
$V_{CE(\text{sat})}$	Collector-Emitter Saturation Voltage	$I_C = 10 \text{ mA}, I_B = 1.0 \text{ mA}$ $I_C = 50 \text{ mA}, I_B = 5.0 \text{ mA}$		0.2 0.3	V
$V_{BE(\text{sat})}$	Base-Emitter Saturation Voltage	$I_C = 10 \text{ mA}, I_B = 1.0 \text{ mA}$ $I_C = 50 \text{ mA}, I_B = 5.0 \text{ mA}$	0.65	0.85 0.95	V

SMALL SIGNAL CHARACTERISTICS

f_T	Current Gain - Bandwidth Product	$I_C = 10 \text{ mA}, V_{CE} = 20 \text{ V}, f = 100 \text{ MHz}$	300		MHz
C_{obo}	Output Capacitance	$V_{CB} = 5.0 \text{ V}, I_E = 0, f = 1.0 \text{ MHz}$		4.0	pF
C_{ibo}	Input Capacitance	$V_{EB} = 0.5 \text{ V}, I_C = 0, f = 1.0 \text{ MHz}$		8.0	pF
NF	Noise Figure	$I_C = 100 \mu\text{A}, V_{CE} = 5.0 \text{ V}, R_S = 1.0 \text{k}\Omega, f = 10 \text{ Hz to } 15.7 \text{ kHz}$		5.0	dB

SWITCHING CHARACTERISTICS

t_d	Delay Time	$V_{CC} = 3.0 \text{ V}, V_{BE} = 0.5 \text{ V},$		35	ns
t_r	Rise Time	$I_C = 10 \text{ mA}, I_{B1} = 1.0 \text{ mA}$		35	ns
t_s	Storage Time	$V_{CC} = 3.0 \text{ V}, I_C = 10 \text{ mA}$		200	ns
t_f	Fall Time	$I_{B1} = I_{B2} = 1.0 \text{ mA}$		50	ns

*Pulse Test: Pulse Width $\leq 300 \mu\text{s}$, Duty Cycle $\leq 2.0\%$

Spice Model

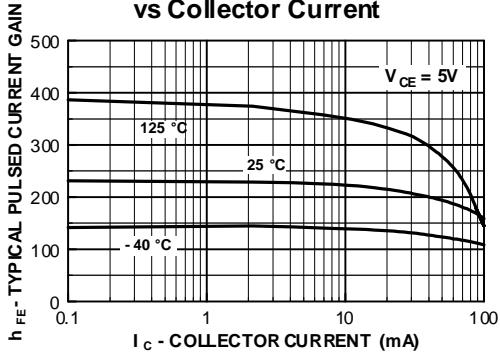
```
NPN (Is=6.734f Xti=3 Eg=1.11 Vaf=74.03 Bf=416.4 Ne=1.259 Ise=6.734 Ikf=66.78m Xtb=1.5 Br=.7371 Nc=2
Isc=0 Ikr=0 Rc=1 Cjc=3.638p Mjc=.3085 Vjc=.75 Fc=.5 Cje=4.493p Mje=.2593 Vje=.75 Tr=239.5n Tf=301.2p
Itf=.4 Vtf=4 Xtf=2 Rb=10)
```

NPN General Purpose Amplifier

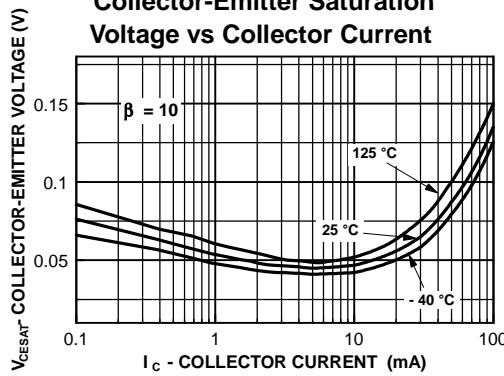
(continued)

Typical Characteristics

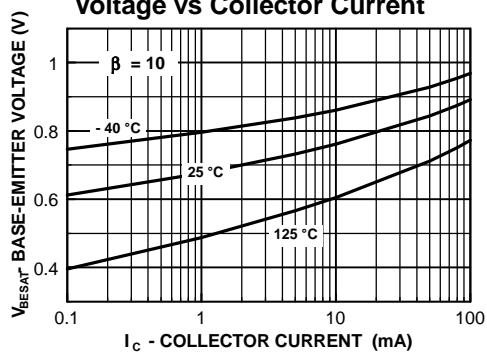
Typical Pulsed Current Gain vs Collector Current



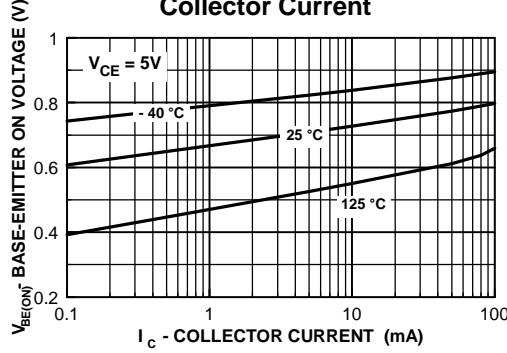
Collector-Emitter Saturation Voltage vs Collector Current



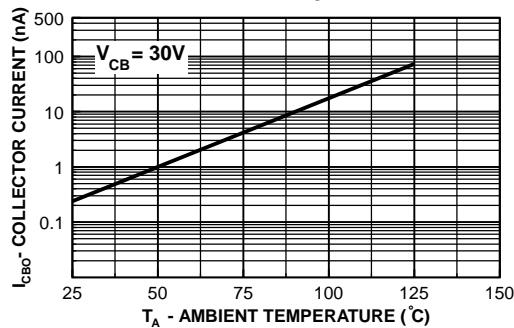
Base-Emitter Saturation Voltage vs Collector Current



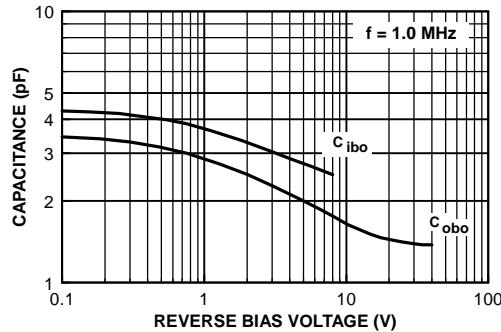
Base-Emitter ON Voltage vs Collector Current



Collector-Cutoff Current vs Ambient Temperature



Capacitance vs Reverse Bias Voltage

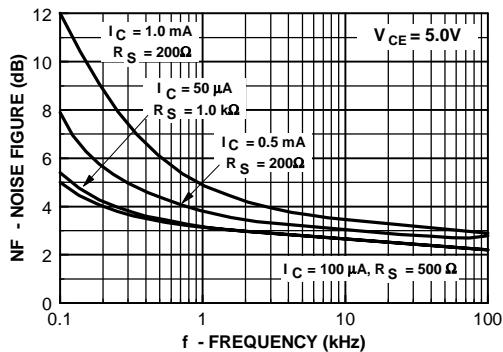


NPN General Purpose Amplifier

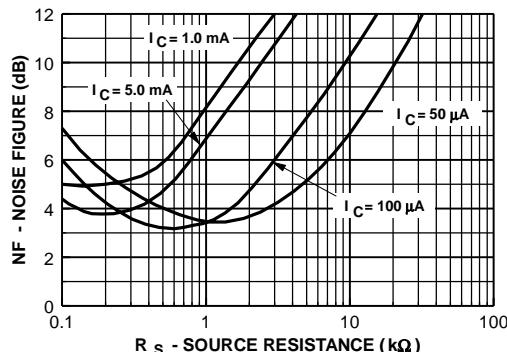
(continued)

Typical Characteristics (continued)

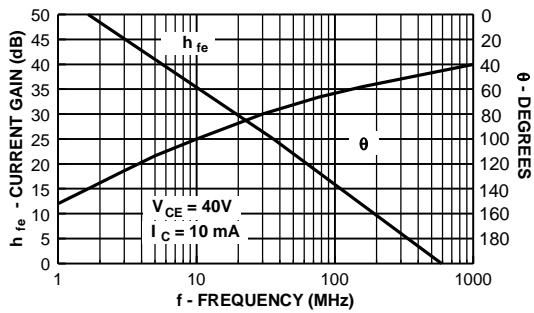
Noise Figure vs Frequency



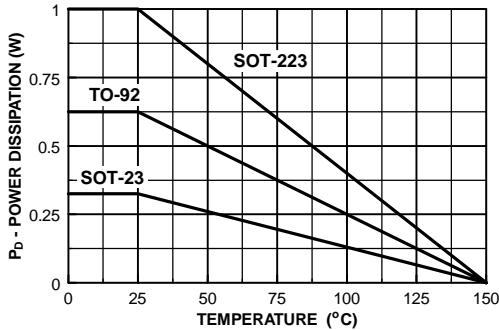
Noise Figure vs Source Resistance



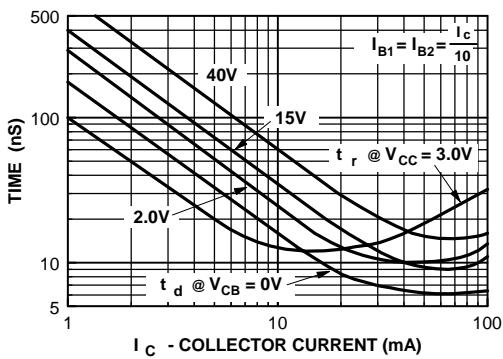
Current Gain and Phase Angle vs Frequency



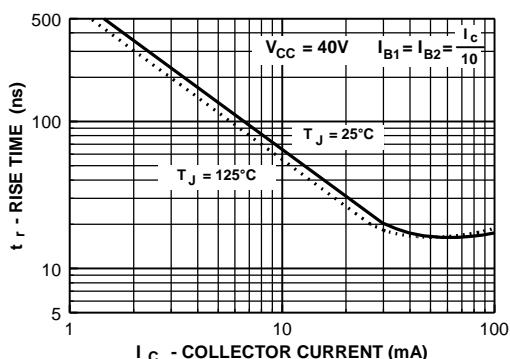
Power Dissipation vs Ambient Temperature



Turn-On Time vs Collector Current



Rise Time vs Collector Current

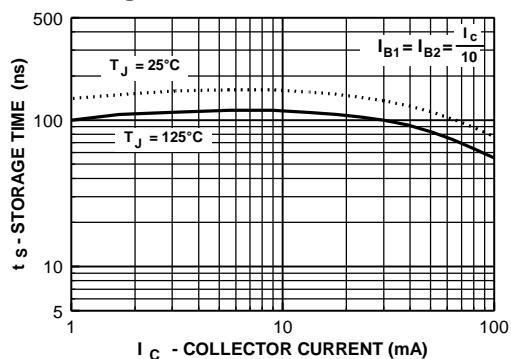


NPN General Purpose Amplifier

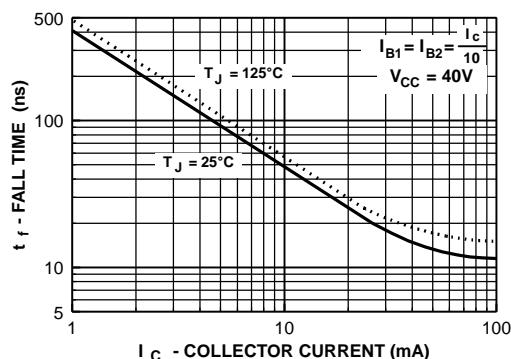
(continued)

Typical Characteristics (continued)

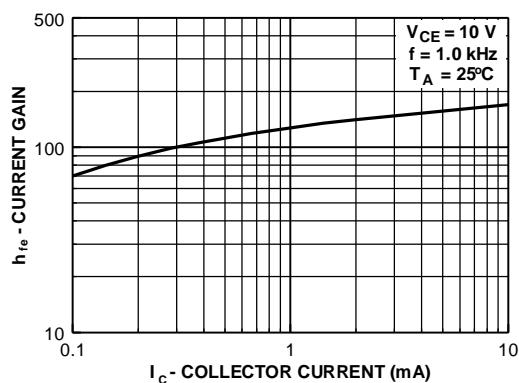
Storage Time vs Collector Current



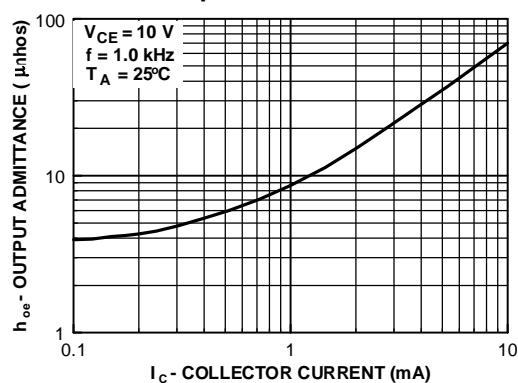
Fall Time vs Collector Current



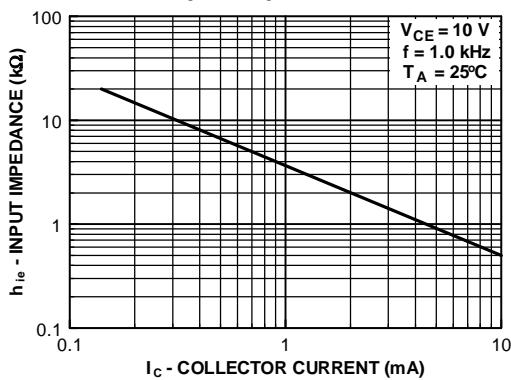
Current Gain



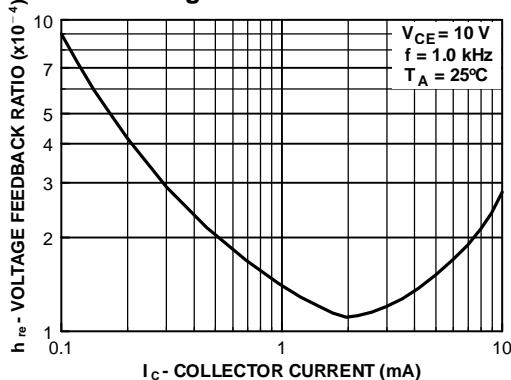
Output Admittance



Input Impedance



Voltage Feedback Ratio



NPN General Purpose Amplifier

(continued)

Test Circuits

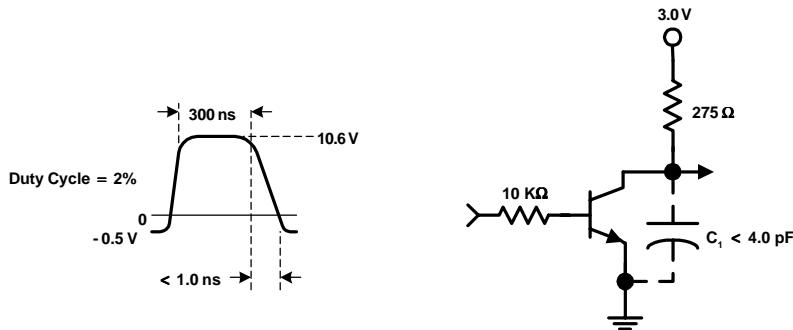


FIGURE 1: Delay and Rise Time Equivalent Test Circuit

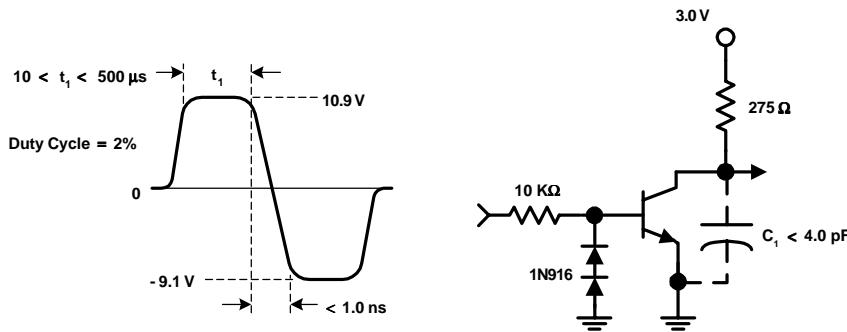
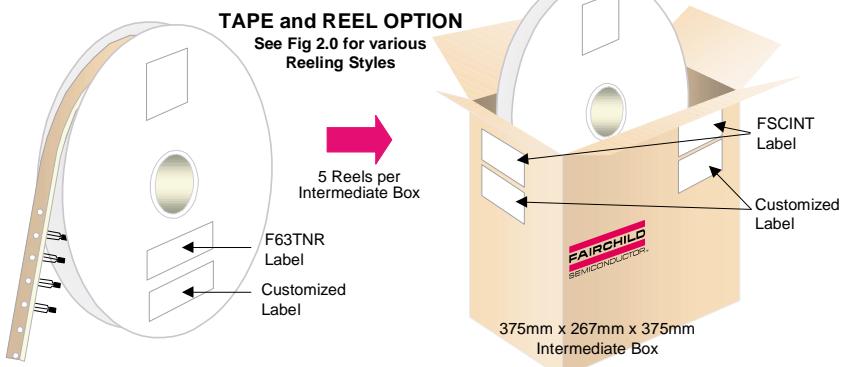
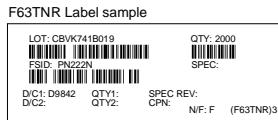


FIGURE 2: Storage and Fall Time Equivalent Test Circuit

TO-92 Tape and Reel Data

FAIRCHILD
SEMICONDUCTOR™

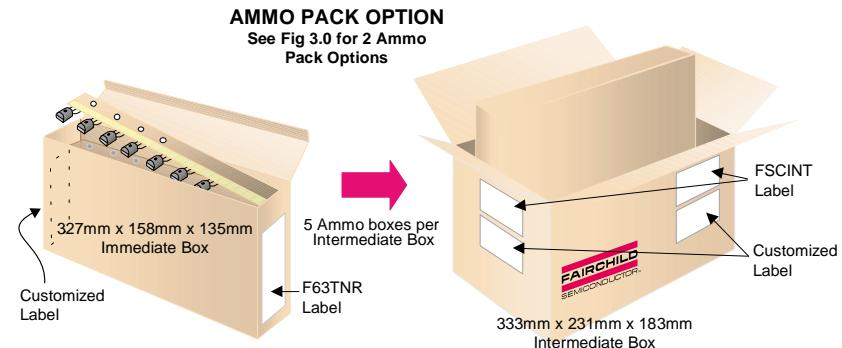
TO-92 Packaging Configuration: Figure 1.0



TO-92 TNR/AMMO PACKING INFORMATION

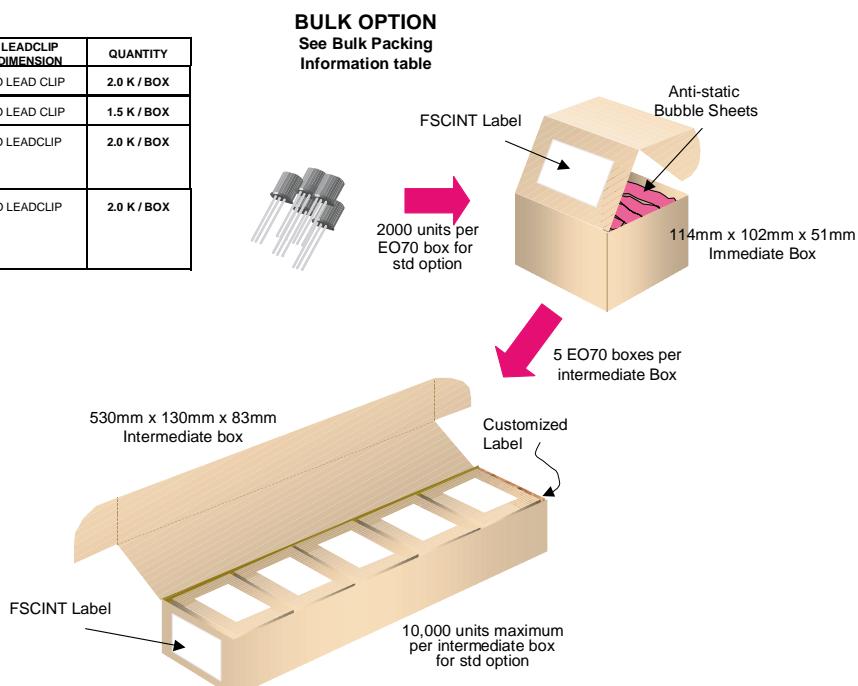
Packing	Style	Quantity	EOL code
Reel	A	2,000	D26Z
	E	2,000	D27Z
Ammo	M	2,000	D74Z
	P	2,000	D75Z

Unit weight = 0.22 gm
Reel weight with components = 1.04 kg
Ammo weight with components = 1.02 kg
Max quantity per intermediate box = 10,000 units



(TO-92) BULK PACKING INFORMATION

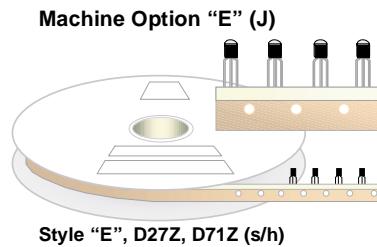
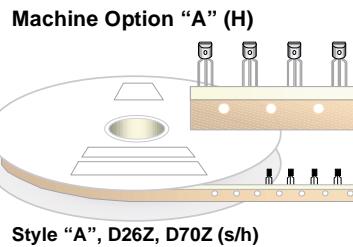
EOL CODE	DESCRIPTION	LEADCLIP DIMENSION	QUANTITY
J18Z	TO-18 OPTION STD	NO LEAD CLIP	2.0 K / BOX
J05Z	TO-5 OPTION STD	NO LEAD CLIP	1.5 K / BOX
NO EOL CODE	TO-92 STANDARD STRAIGHT FOR: PKG 92, 94 (NON PROELECTRON SERIES), 96	NO LEADCLIP	2.0 K / BOX
L34Z	TO-92 STANDARD STRAIGHT FOR: PKG 94 (PROELECTRON SERIES BCXXX, BFXXX, BSXXX), 97, 98	NO LEADCLIP	2.0 K / BOX



TO-92 Tape and Reel Data, continued

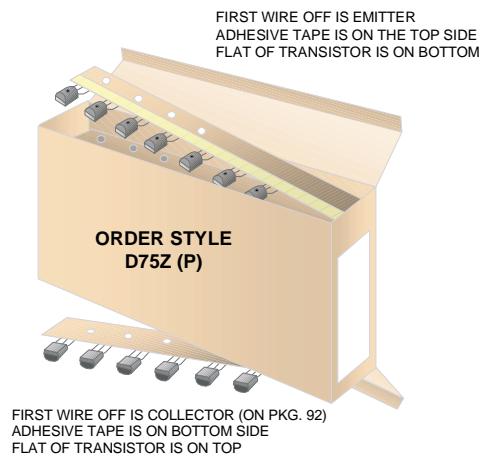
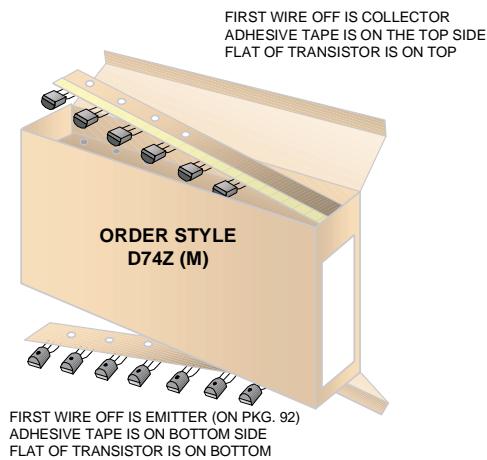
TO-92 Reeling Style

Configuration: Figure 2.0



TO-92 Radial Ammo Packaging

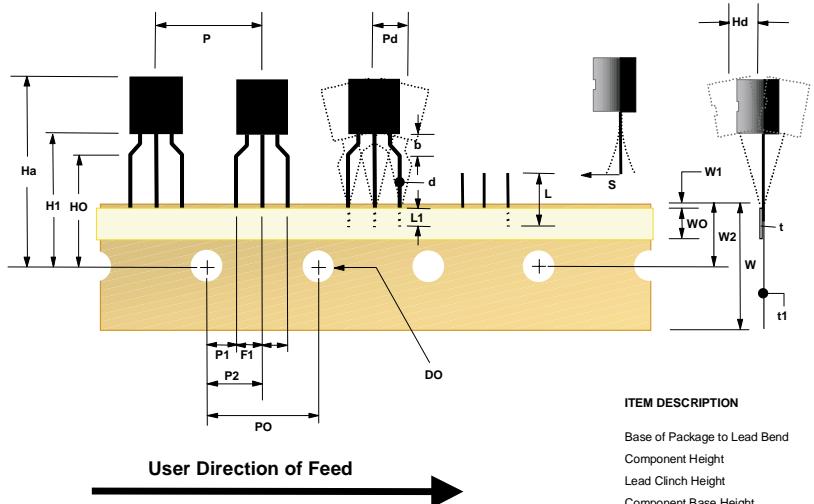
Configuration: Figure 3.0



TO-92 Tape and Reel Data, continued

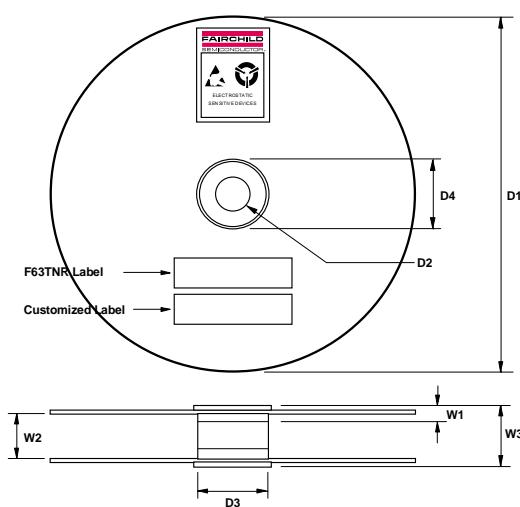
TO-92 Tape and Reel Taping

Dimension Configuration: Figure 4.0



ITEM DESCRIPTION	SYMBOL	DIMENSION
Base of Package to Lead Bend	b	0.098 (max)
Component Height	Ha	0.928 (+/- 0.025)
Lead Clinch Height	HO	0.630 (+/- 0.020)
Component Base Height	H1	0.748 (+/- 0.020)
Component Alignment (side/side)	Pd	0.040 (max)
Component Alignment (front/back)	Hd	0.031 (max)
Component Pitch	P	0.500 (+/- 0.020)
Feed Hole Pitch	PO	0.500 (+/- 0.008)
Hole Center to First Lead	P1	0.150 (+0.009, -0.010)
Hole Center to Component Center	P2	0.247 (+/- 0.007)
Lead Spread	F1/F2	0.104 (+/- 0.010)
Lead Thickness	d	0.018 (+0.002, -0.003)
Cut Lead Length	L	0.429 (max)
Taped Lead Length	L1	0.209 (+0.051, -0.052)
Taped Lead Thickness	t	0.032 (+/- 0.006)
Carrier Tape Thickness	t1	0.021 (+/- 0.006)
Carrier Tape Width	W	0.708 (+0.020, -0.019)
Hold - down Tape Width	WO	0.236 (+/- 0.012)
Hold - down Tape position	W1	0.035 (max)
Feed Hole Position	W2	0.360 (+/- 0.025)
Sprocket Hole Diameter	DO	0.157 (+0.008, -0.007)
Lead Spring Out	S	0.004 (max)

Note : All dimensions are in inches.



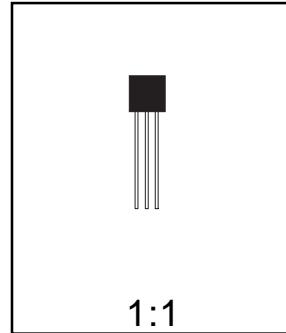
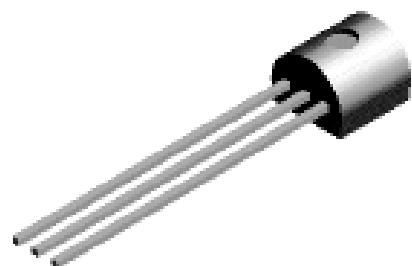
ITEM DESCRIPTION	SYSMBOL	MINIMUM	MAXIMUM
Reel Diameter	D1	13.975	14.025
Arbor Hole Diameter (Standard)	D2	1.160	1.200
(Small Hole)	D2	0.650	0.700
Core Diameter	D3	3.100	3.300
Hub Recess Inner Diameter	D4	2.700	3.100
Hub Recess Depth	W1	0.370	0.570
Flange to Flange Inner Width	W2	1.630	1.690
Hub to Hub Center Width	W3		2.090

Note: All dimensions are inches

TO-92 Package Dimensions

FAIRCHILD
SEMICONDUCTOR™

TO-92 (FS PKG Code 92, 94, 96)



1:1

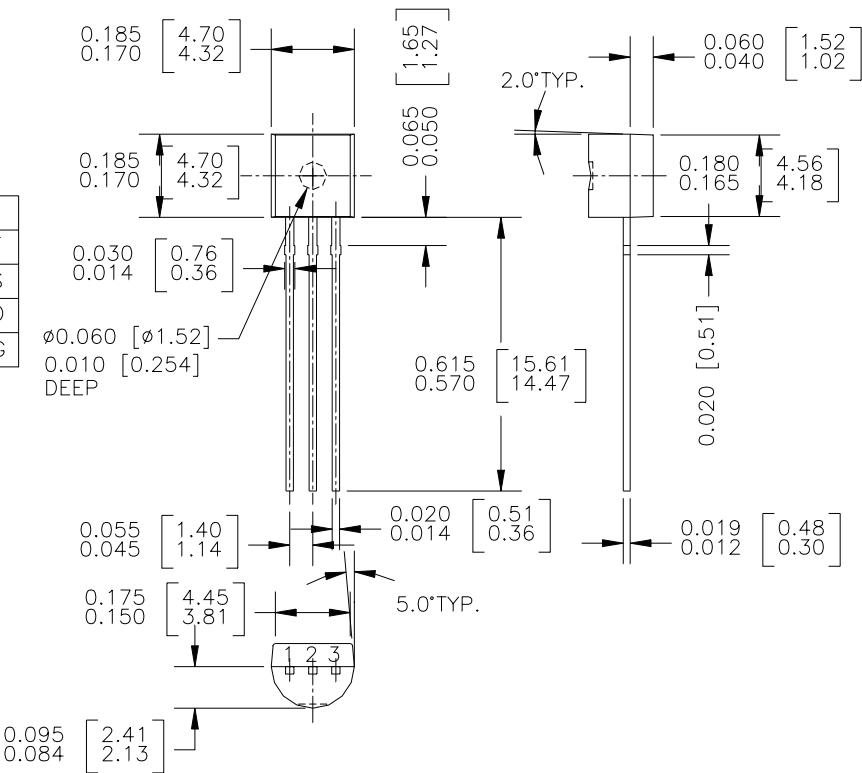
Scale 1:1 on letter size paper

Dimensions shown below are in:
inches [millimeters]

Part Weight per unit (gram): 0.1977

TO-92 (92,94,96)

PIN	92		94		96	
	B	F	B	F	B	F
1	E	D	E	D	B	S
2	B	S	C	G	E	D
3	C	G	B	S	C	G

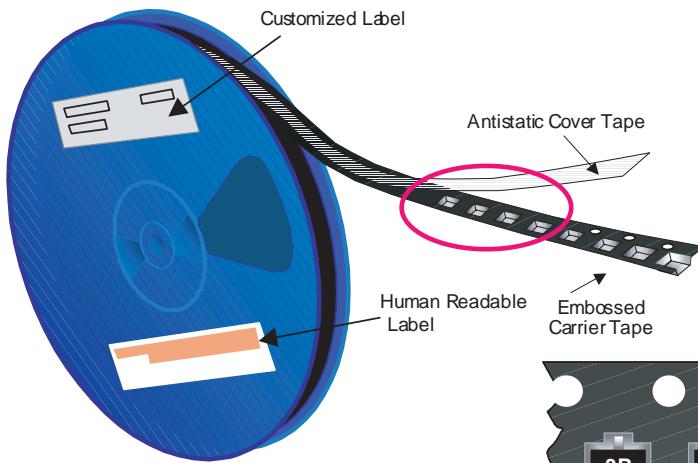


SOT-23 Tape and Reel Data

FAIRCHILD
SEMICONDUCTOR™

SOT-23 Packaging

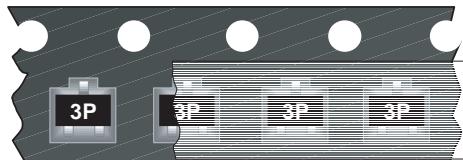
Configuration: Figure 10



Packaging Description:

SOT-23 parts are shipped in tape. The carrier tape is made from a dissipative (carbon filled) polycarbonate resin. The cover tape is a multilayer film (Heat Activated Adhesive in nature) primarily composed of polyester film, adhesive layer, sealant, and anti-static sprayed agent. These reeled parts in standard option are shipped with 3,000 units per 7" or 177mm diameter reel. The reels are dark blue in color and is made of polystyrene plastic (anti-static coated). Other option comes in 10,000 units per 13" or 330mm diameter reel. This and some other options are described in the Packaging Information table.

These full reels are individually labeled and placed inside a standard intermediate made of recyclable corrugated brown paper with a Fairchild logo printing. One pizza box contains eight reels maximum. And these intermediate boxes are placed inside a labeled shipping box which comes in different sizes depending on the number of parts shipped.



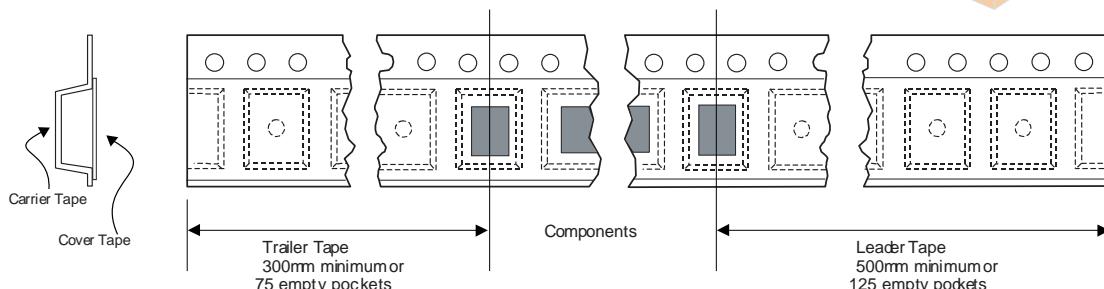
SOT-23 Unit Orientation

SOT-23 Packaging Information		
Packaging Option	Standard (no flow code)	D87Z
Packaging type	TNR	TNR
Qty per Reel/Tube/Bag	3,000	10,000
Reel Size	7" Dia	13"
Box Dimension (mm)	187x107x183	343x343x64
Max qty per Box	24,000	30,000
Weight per unit (gm)	0.0082	0.0082
Weight per Reel (kg)	0.1175	0.4006
Note/Comments		

Human Readable Label sample



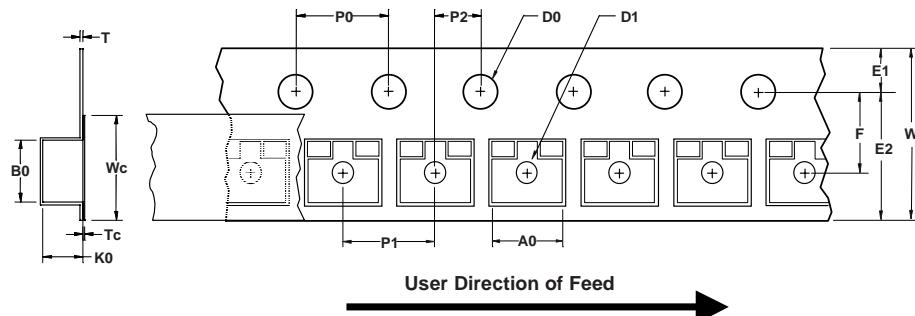
SOT-23 Tape Leader and Trailer Configuration: Figure 20



SOT-23 Tape and Reel Data, continued

SOT-23 Embossed Carrier Tape

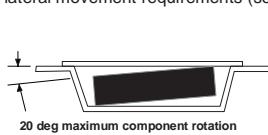
Configuration: Figure 3.0



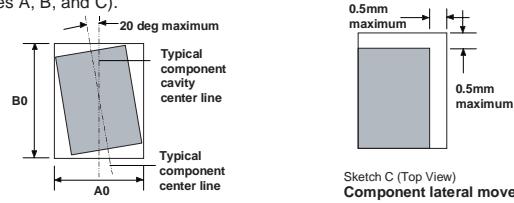
Dimensions are in millimeter

Pkg type	A_0	B_0	W	D_0	D_1	E_1	E_2	F	P_1	P_0	K_0	T	W_c	T_c
SOT-23 (8mm)	3.15 ± 0.10	2.77 ± 0.10	8.0 ± 0.3	1.55 ± 0.05	1.125 ± 0.125	1.75 ± 0.10	6.25 min	3.50 ± 0.05	4.0 ± 0.1	4.0 ± 0.1	1.30 ± 0.10	0.228 ± 0.013	5.2 ± 0.3	0.06 ± 0.02

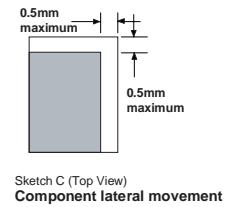
Notes: A_0 , B_0 , and K_0 dimensions are determined with respect to the EIA/Jedec RS-481 rotational and lateral movement requirements (see sketches A, B, and C).



Sketch A (Side or Front Sectional View)
Component Rotation

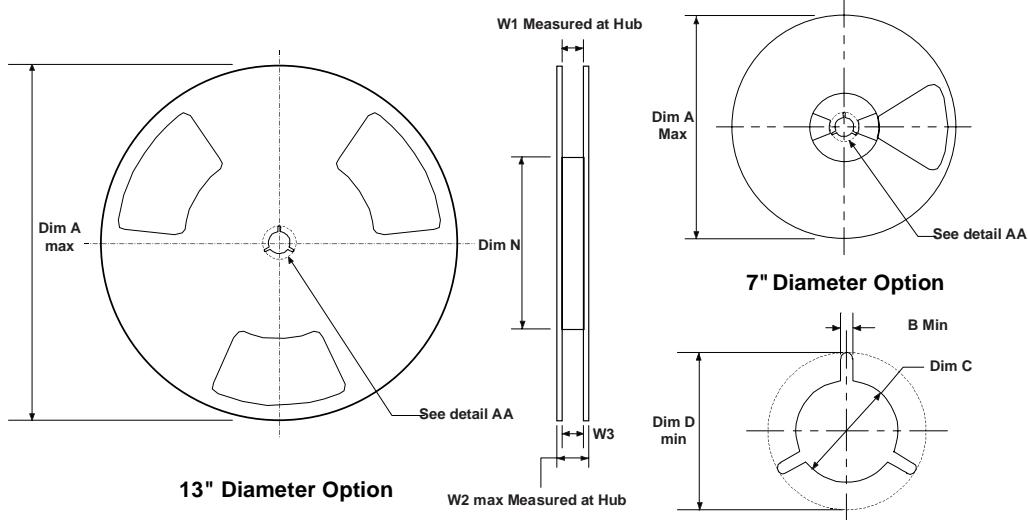


Sketch B (Top View)
Component Rotation



Sketch C (Top View)
Component lateral movement

SOT-23 Reel Configuration: Figure 4.0



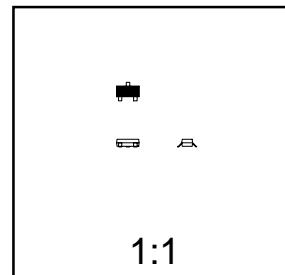
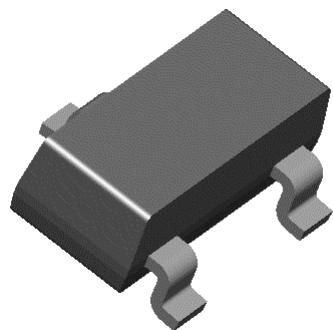
Dimensions are in inches and millimeters

Tape Size	Reel Option	Dim A	Dim B	Dim C	Dim D	Dim N	Dim W1	Dim W2	Dim W3 (LSL-USL)
8mm	7" Dia	7.00 177.8	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	2.165 55	0.331 +0.059/-0.000 8.4 +1.5/0	0.567 14.4	0.311 - 0.429 7.9 - 10.9
8mm	13" Dia	13.00 330	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	4.00 100	0.331 +0.059/-0.000 8.4 +1.5/0	0.567 14.4	0.311 - 0.429 7.9 - 10.9

SOT-23 Package Dimensions



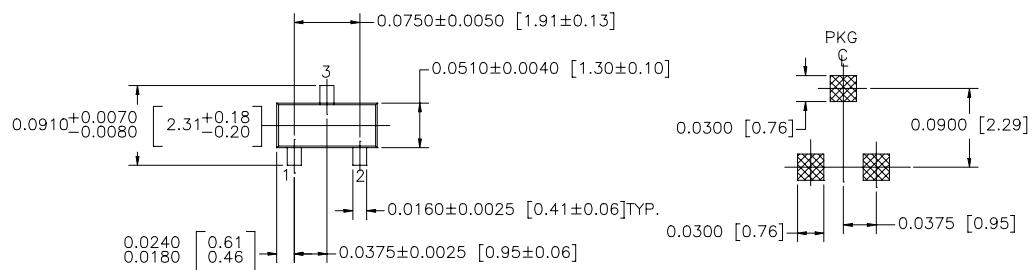
SOT-23 (FS PKG Code 49)



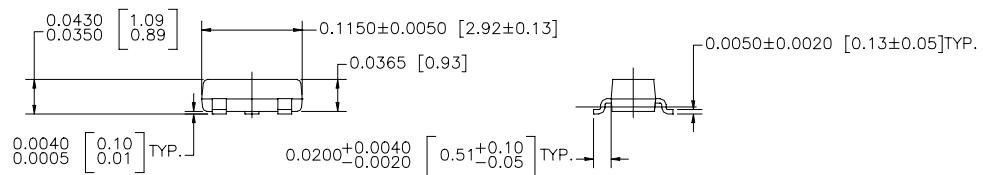
Scale 1:1 on letter size paper

Dimensions shown below are in:
inches [millimeters]

Part Weight per unit (gram): 0.0082



LAND PATTERN RECOMMENDATION



CONTROLLING DIMENSION IS INCH
VALUES IN [] ARE MILLIMETERS

SOT 23, 3 LEADS LOW PROFILE

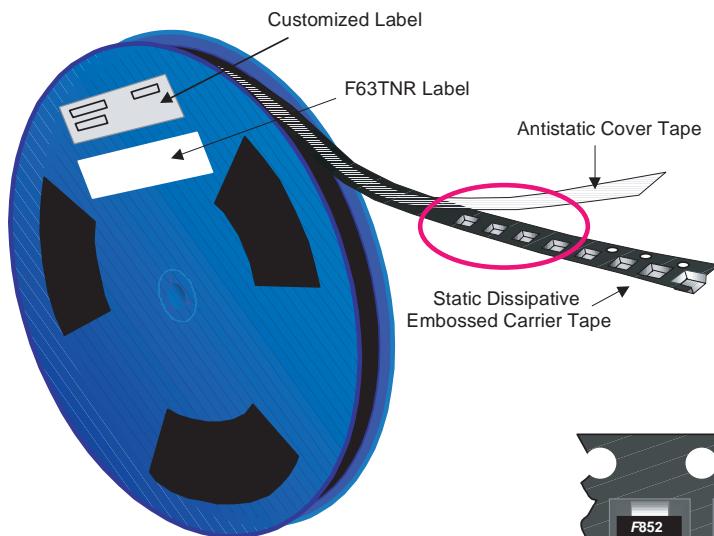
NOTE : UNLESS OTHERWISE SPECIFIED

1. STANDARD LEAD FINISH 150 MICROINCHES / 3.81 MICROMETERS
MINIMUM TIN / LEAD (SOLDER) ON ALLOY 42
2. REFERENCE JEDEC REGISTRATION TO-236, VARIATION AB, ISSUE G, DATED JUL 1993

SOT-223 Tape and Reel Data

FAIRCHILD
SEMICONDUCTOR™

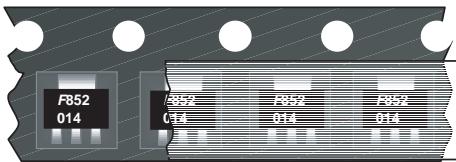
SOT-223 Packaging Configuration: Figure 1.0



Packaging Description:

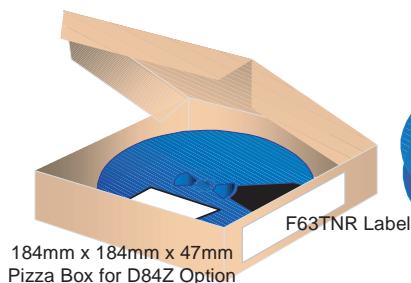
SOT-223 parts are shipped in tape. The carrier tape is made from a dissipative (carbon filled) polycarbonate resin. The cover tape is a multilayer film (Heat Activated Adhesive in nature) primarily composed of polyester film, adhesive layer, sealant, and anti-static sprayed agent. These reeled parts in standard option are shipped with 2,500 units per 13" or 330mm diameter reel. The reels are dark blue in color and is made of polystyrene plastic (anti-static coated). Other option comes in 500 units per 7" or 177mm diameter reel. This and some other options are further described in the Packaging Information table.

These full reels are individually barcode labeled and placed inside a standard intermediate box (illustrated in figure 1.0) made of recyclable corrugated brown paper. One box contains two reels maximum. And these boxes are placed inside a barcode labeled shipping box which comes in different sizes depending on the number of parts shipped.



SOT-223 Unit Orientation

SOT-223 Packaging Information		
Packaging Option	Standard (no flow code)	D84Z
Packaging type	TNR	TNR
Qty per Reel/Tube/Bag	2,500	500
Reel Size	13" Dia	7" Dia
Box Dimension (mm)	343x64x343	184x187x47
Max qty per Box	5,000	1,000
Weight per unit (gm)	0.1246	0.1246
Weight per Reel (kg)	0.7250	0.1532
Note/Comments		



343mm x 342mm x 64mm
Intermediate box for Standard

F63TNR Label

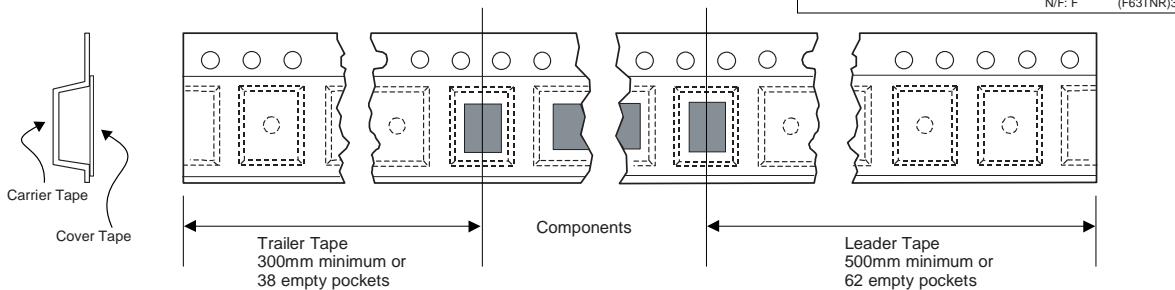
F63TNR Label sample

LOT: CBVK741B019	QTY: 3000
FSID: PN2222A	SPEC:
D/C1: D9842	QTY1:
D/C2:	QTY2:
	SPEC REV:
	CPN:
	N/F: F
	(F63TNR)3

184mm x 184mm x 47mm

Pizza Box for D84Z Option

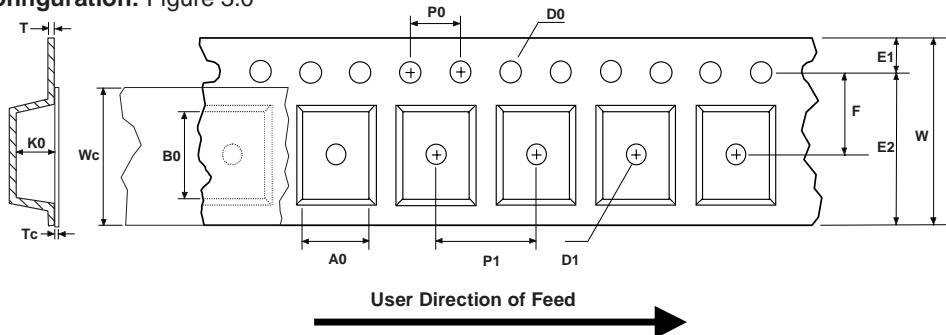
SOT-223 Tape Leader and Trailer Configuration: Figure 2.0



SOT-223 Tape and Reel Data, continued

SOT-223 Embossed Carrier Tape

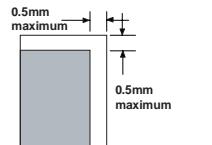
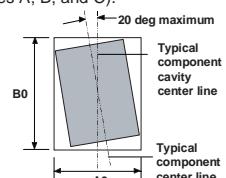
Configuration: Figure 3.0



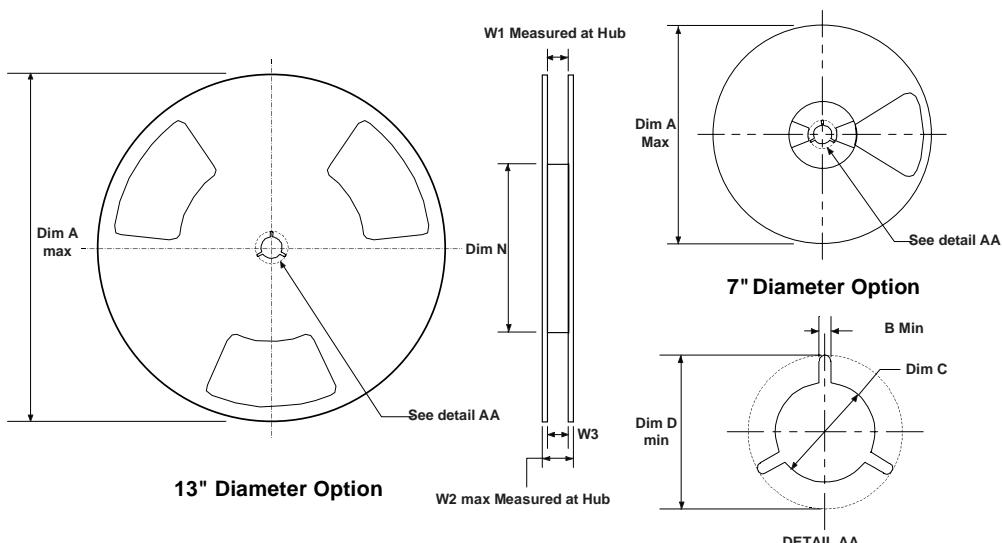
Dimensions are in millimeter

Pkg type	A_0	B_0	W	D_0	D_1	E_1	E_2	F	P_1	P_0	K_0	T	W_c	T_c
SOT-223 (12mm)	6.83 ± 0.10	7.42 ± 0.10	12.0 ± 0.3	1.55 ± 0.05	1.50 ± 0.10	1.75 ± 0.10	10.25 min	5.50 ± 0.05	8.0 ± 0.1	4.0 ± 0.1	1.88 ± 0.10	0.292 ± 0.0130	9.5 ± 0.025	0.06 ± 0.02

Notes: A_0 , B_0 , and K_0 dimensions are determined with respect to the EIA/Jedec RS-481 rotational and lateral movement requirements (see sketches A, B, and C).



SOT-223 Reel Configuration: Figure 4.0



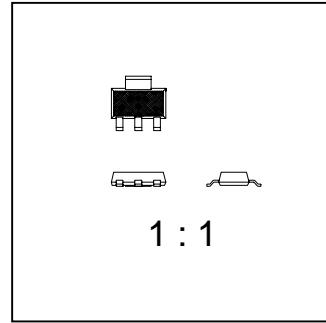
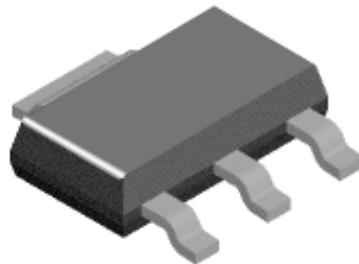
Dimensions are in inches and millimeters

Tape Size	Reel Option	Dim A	Dim B	Dim C	Dim D	Dim N	Dim W1	Dim W2	Dim W3 (LSL-USL)
12mm	7" Dia	7.00 177.8	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	5.906 150	0.488 +0.078/-0.000 12.4 +2/0	0.724 18.4	0.469 - 0.606 11.9 - 15.4
12mm	13" Dia	13.00 330	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	7.00 178	0.488 +0.078/-0.000 12.4 +2/0	0.724 18.4	0.469 - 0.606 11.9 - 15.4

SOT-223 Package Dimensions

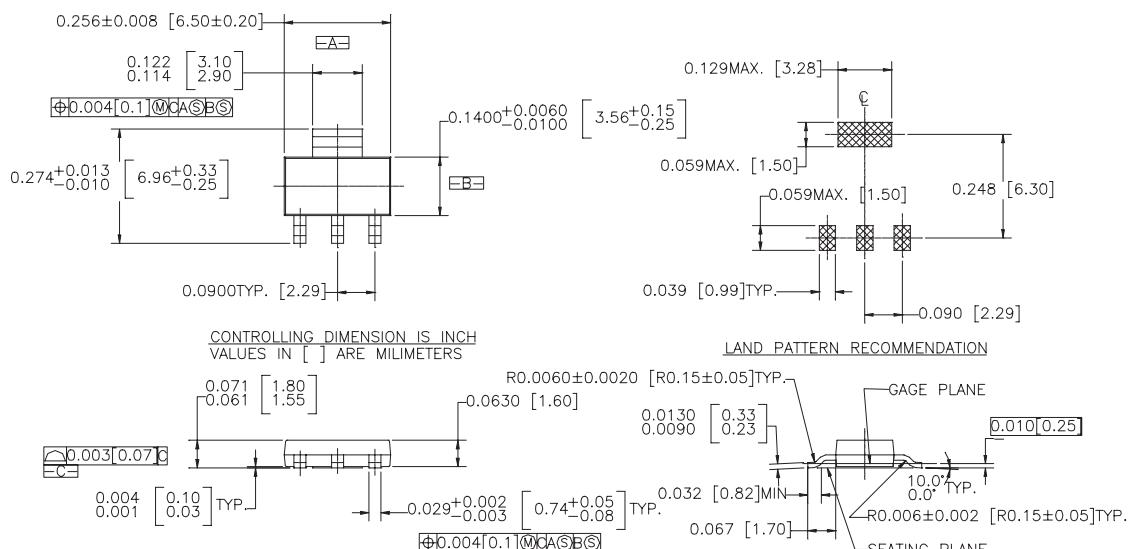
FAIRCHILD
SEMICONDUCTOR™

SOT-223 (FS PKG Code 47)



Scale 1:1 on letter size paper

Part Weight per unit (gram): 0.1246



TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™	FASTR™	PowerTrench®	SyncFET™
Bottomless™	GlobalOptoisolator™	QFET™	TinyLogic™
CoolFET™	GTO™	QS™	UHC™
CROSSVOLT™	HiSeC™	QT Optoelectronics™	VCX™
DOME™	ISOPLANAR™	Quiet Series™	
E ² CMOS™	MICROWIRE™	SILENT SWITCHER®	
EnSigna™	OPTOLOGIC™	SMART START™	
FACT™	OPTOPLANAR™	SuperSOT™-3	
FACT Quiet Series™	PACMAN™	SuperSOT™-6	
FAST®	POP™	SuperSOT™-8	

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

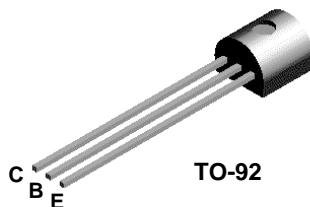
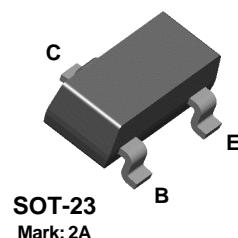
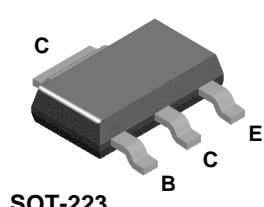
FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

**2N3906****MMBT3906****PZT3906**

PNP General Purpose Amplifier

This device is designed for general purpose amplifier and switching applications at collector currents of 10 μ A to 100 mA.

Absolute Maximum Ratings*

$T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	40	V
V_{EBO}	Emitter-Base Voltage	5.0	V
I_C	Collector Current - Continuous	200	mA
T_J, T_{stg}	Operating and Storage Junction Temperature Range	-55 to +150	$^\circ\text{C}$

* These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:

- 1) These ratings are based on a maximum junction temperature of 150 degrees C.
- 2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.
- 3) All voltages (V) and currents (A) are negative polarity for PNP transistors.

Thermal Characteristics

$T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Characteristic	Max			Units
		2N3906	*MMBT3906	**PZT3906	
P_D	Total Device Dissipation Derate above 25°C	625 5.0	350 2.8	1,000 8.0	mW mW/ $^\circ\text{C}$
$R_{\theta JC}$	Thermal Resistance, Junction to Case	83.3			$^\circ\text{C/W}$
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	200	357	125	$^\circ\text{C/W}$

* Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06."

** Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6 cm².

PNP General Purpose Amplifier

(continued)

Electrical Characteristics

$T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Conditions	Min	Max	Units
OFF CHARACTERISTICS					
$V_{(\text{BR})\text{CEO}}$	Collector-Emitter Breakdown Voltage*	$I_C = 1.0 \text{ mA}, I_B = 0$	40		V
$V_{(\text{BR})\text{CBO}}$	Collector-Base Breakdown Voltage	$I_C = 10 \mu\text{A}, I_E = 0$	40		V
$V_{(\text{BR})\text{EBO}}$	Emitter-Base Breakdown Voltage	$I_E = 10 \mu\text{A}, I_C = 0$	5.0		V
I_{BL}	Base Cutoff Current	$V_{CE} = 30 \text{ V}, V_{BE} = 3.0 \text{ V}$		50	nA
I_{CEX}	Collector Cutoff Current	$V_{CE} = 30 \text{ V}, V_{BE} = 3.0 \text{ V}$		50	nA

ON CHARACTERISTICS

h_{FE}	DC Current Gain *	$I_C = 0.1 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 1.0 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 10 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 50 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 100 \text{ mA}, V_{CE} = 1.0 \text{ V}$	60 80 100 60 30	300	
$V_{CE(\text{sat})}$	Collector-Emitter Saturation Voltage	$I_C = 10 \text{ mA}, I_B = 1.0 \text{ mA}$ $I_C = 50 \text{ mA}, I_B = 5.0 \text{ mA}$		0.25 0.4	V
$V_{BE(\text{sat})}$	Base-Emitter Saturation Voltage	$I_C = 10 \text{ mA}, I_B = 1.0 \text{ mA}$ $I_C = 50 \text{ mA}, I_B = 5.0 \text{ mA}$	0.65	0.85 0.95	V

SMALL SIGNAL CHARACTERISTICS

f_T	Current Gain - Bandwidth Product	$I_C = 10 \text{ mA}, V_{CE} = 20 \text{ V}, f = 100 \text{ MHz}$	250		MHz
C_{obo}	Output Capacitance	$V_{CB} = 5.0 \text{ V}, I_E = 0, f = 100 \text{ kHz}$		4.5	pF
C_{ibo}	Input Capacitance	$V_{EB} = 0.5 \text{ V}, I_C = 0, f = 100 \text{ kHz}$		10.0	pF
NF	Noise Figure	$I_C = 100 \mu\text{A}, V_{CE} = 5.0 \text{ V}, R_S = 1.0 \text{k}\Omega, f = 10 \text{ Hz to } 15.7 \text{ kHz}$		4.0	dB

SWITCHING CHARACTERISTICS

t_d	Delay Time	$V_{CC} = 3.0 \text{ V}, V_{BE} = 0.5 \text{ V}, I_C = 10 \text{ mA}, I_{B1} = 1.0 \text{ mA}$		35	ns
t_r	Rise Time			35	ns
t_s	Storage Time	$V_{CC} = 3.0 \text{ V}, I_C = 10 \text{ mA}$ $I_{B1} = I_{B2} = 1.0 \text{ mA}$		225	ns
t_f	Fall Time			75	ns

* Pulse Test: Pulse Width $\leq 300 \mu\text{s}$, Duty Cycle $\leq 2.0\%$

NOTE: All voltages (V) and currents (A) are negative polarity for PNP transistors.

Spice Model

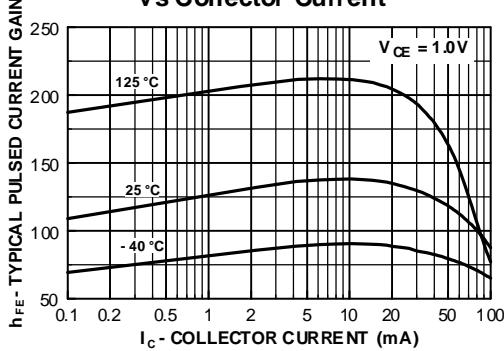
PNP (Is=1.41f Xti=3 Eg=1.11 Vaf=18.7 Bf=180.7 Ne=1.5 Ise=0 Ikf=80m Xtb=1.5 Br=4.977 Nc=2 Isc=0 Ikr=0 Rc=2.5 Cjc=9.728p Mjc=.5776 Vjc=.75 Fc=.5 Cje=8.063p Mje=.3677 Vje=.75 Tr=33.42n Tf=179.3p Itf=.4 Vtf=4 Xtf=6 Rb=10)

PNP General Purpose Amplifier

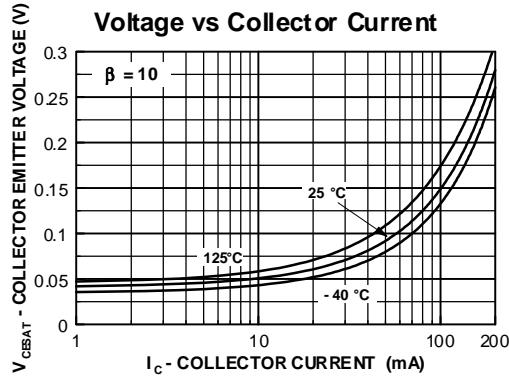
(continued)

Typical Characteristics

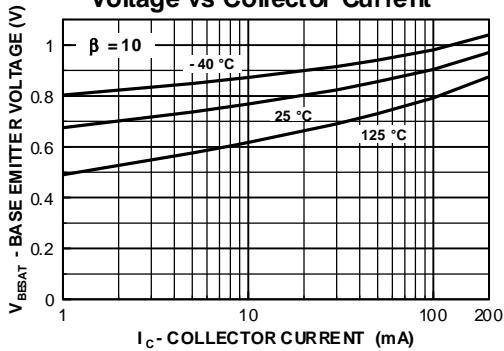
**Typical Pulsed Current Gain
vs Collector Current**



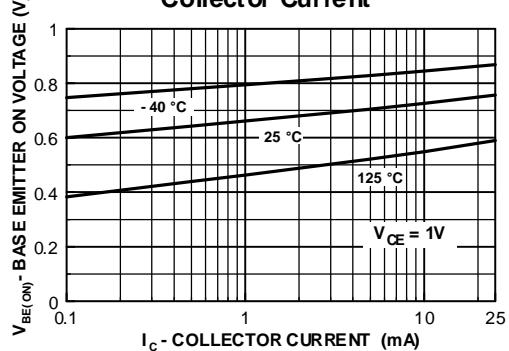
Collector-Emitter Saturation Voltage vs Collector Current



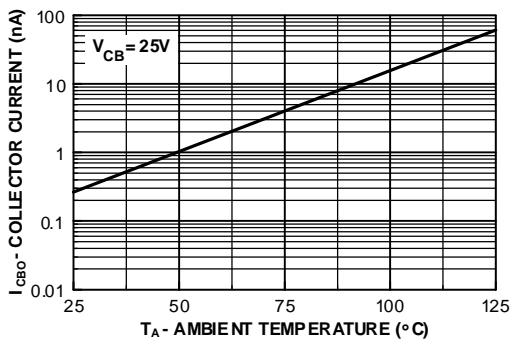
Base-Emitter Saturation Voltage vs Collector Current



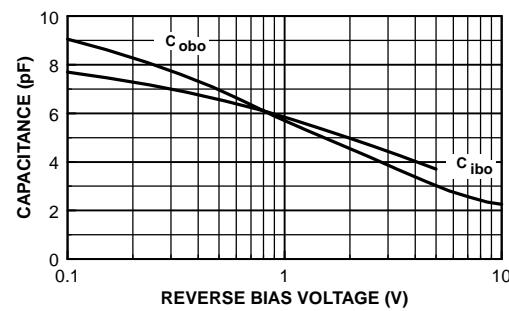
Base Emitter ON Voltage vs Collector Current



Collector-Cutoff Current vs Ambient Temperature



Common-Base Open Circuit Input and Output Capacitance vs Reverse Bias Voltage

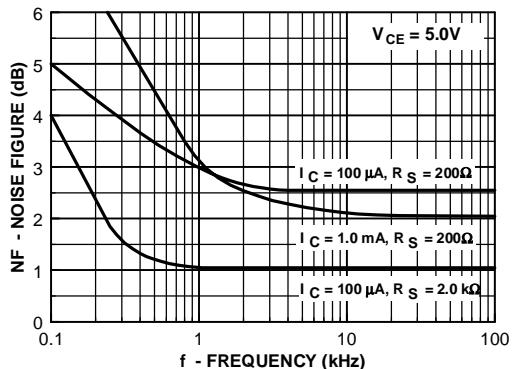


PNP General Purpose Amplifier

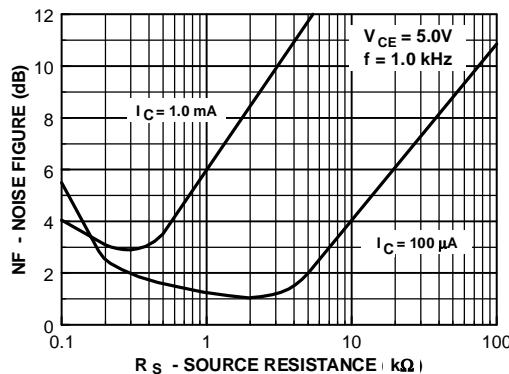
(continued)

Typical Characteristics (continued)

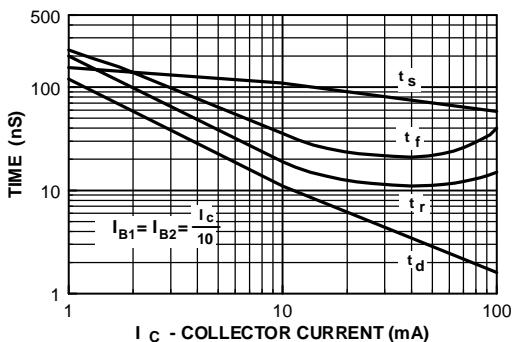
Noise Figure vs Frequency



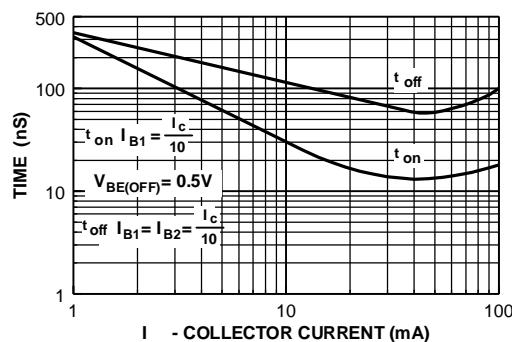
Noise Figure vs Source Resistance



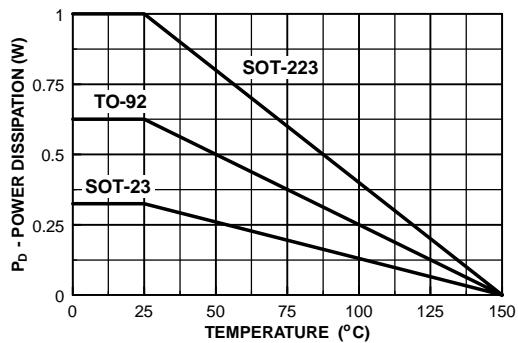
**Switching Times
vs Collector Current**

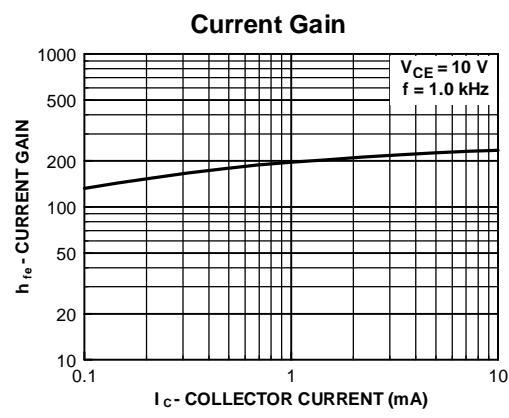
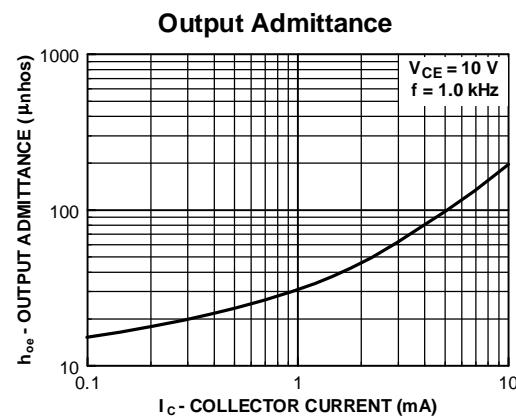
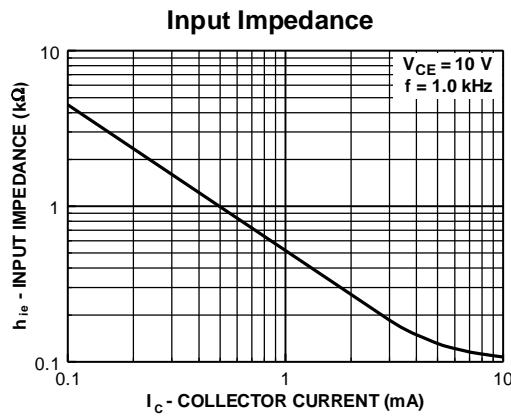
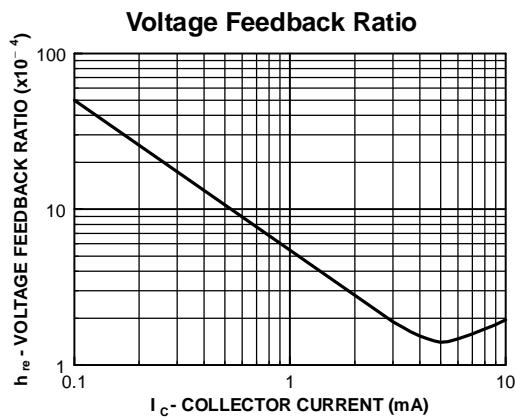


**Turn On and Turn Off Times
vs Collector Current**



**Power Dissipation vs
Ambient Temperature**

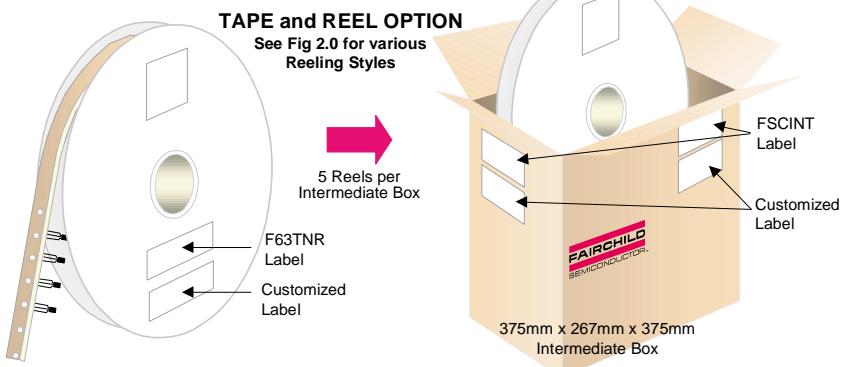
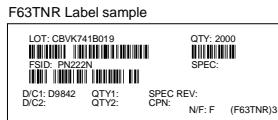


PNP General Purpose Amplifier
(continued)**Typical Characteristics** (continued)

TO-92 Tape and Reel Data

FAIRCHILD
SEMICONDUCTOR™

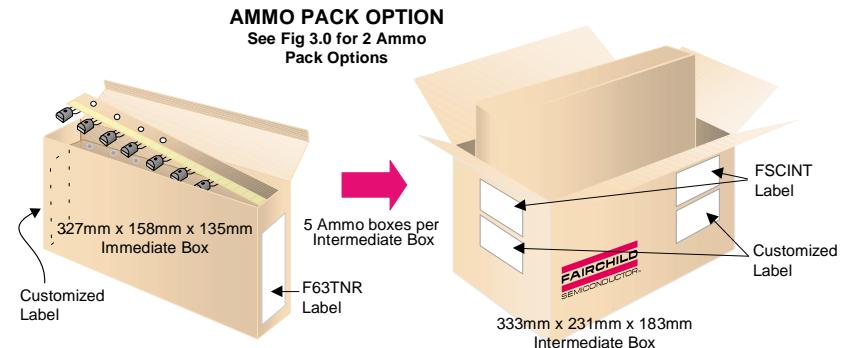
TO-92 Packaging Configuration: Figure 1.0



TO-92 TNR/AMMO PACKING INFORMATION

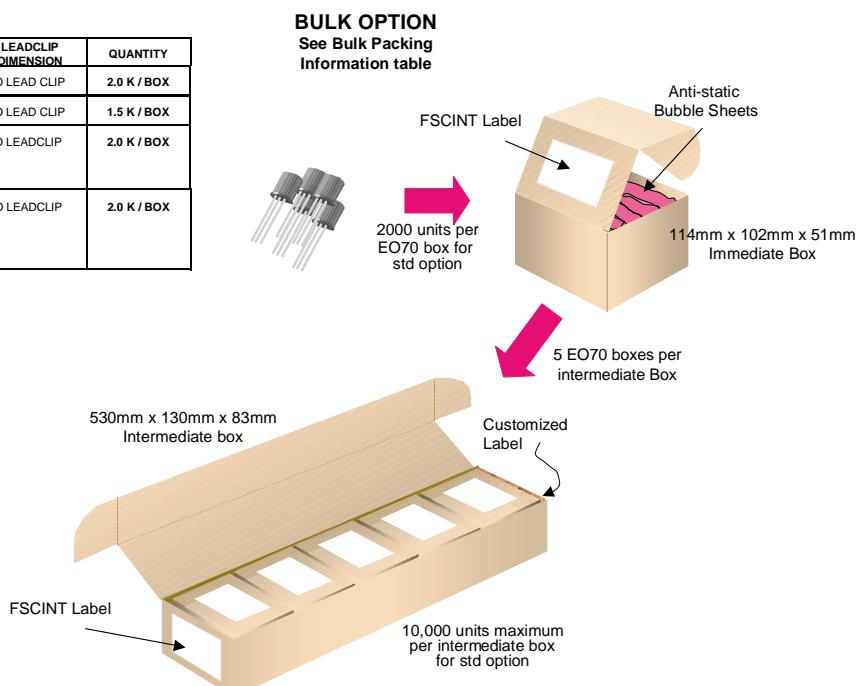
Packing	Style	Quantity	EOL code
Reel	A	2,000	D26Z
	E	2,000	D27Z
Ammo	M	2,000	D74Z
	P	2,000	D75Z

Unit weight = 0.22 gm
Reel weight with components = 1.04 kg
Ammo weight with components = 1.02 kg
Max quantity per intermediate box = 10,000 units



(TO-92) BULK PACKING INFORMATION

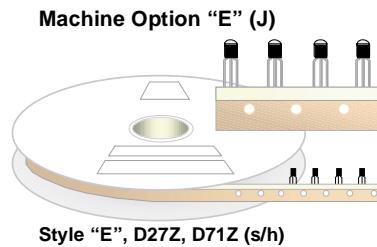
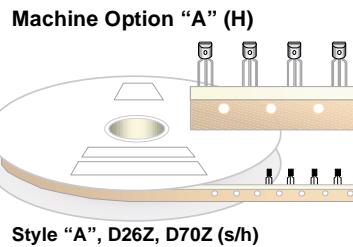
EOL CODE	DESCRIPTION	LEADCLIP DIMENSION	QUANTITY
J18Z	TO-18 OPTION STD	NO LEAD CLIP	2.0 K / BOX
J05Z	TO-5 OPTION STD	NO LEAD CLIP	1.5 K / BOX
NO EOL CODE	TO-92 STANDARD STRAIGHT FOR: PKG 92, 94 (NON PROELECTRON SERIES), 96	NO LEADCLIP	2.0 K / BOX
L34Z	TO-92 STANDARD STRAIGHT FOR: PKG 94 (PROELECTRON SERIES BCXXX, BFXXX, BSXXX), 97, 98	NO LEADCLIP	2.0 K / BOX



TO-92 Tape and Reel Data, continued

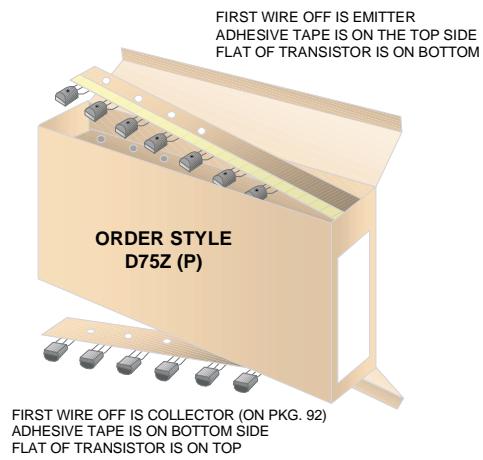
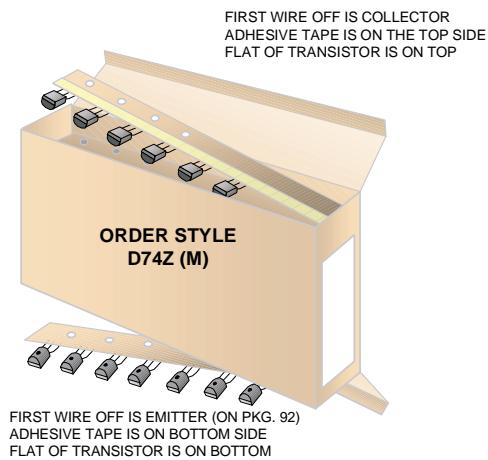
TO-92 Reeling Style

Configuration: Figure 2.0



TO-92 Radial Ammo Packaging

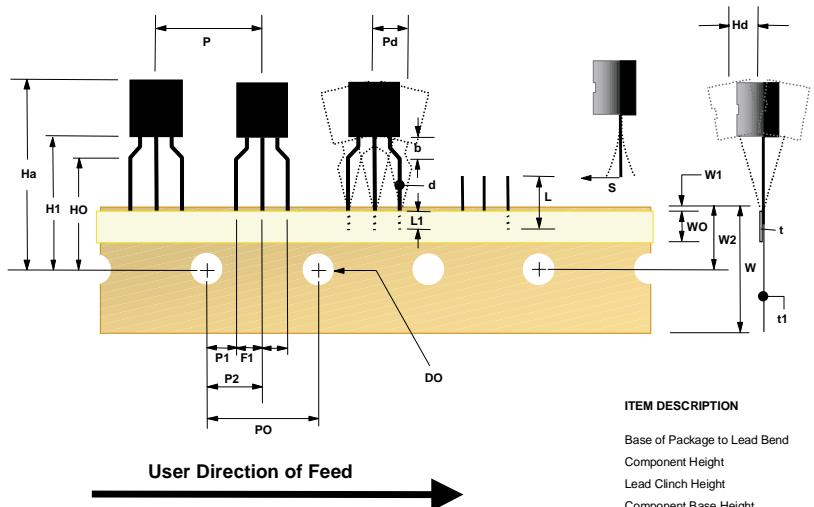
Configuration: Figure 3.0



TO-92 Tape and Reel Data, continued

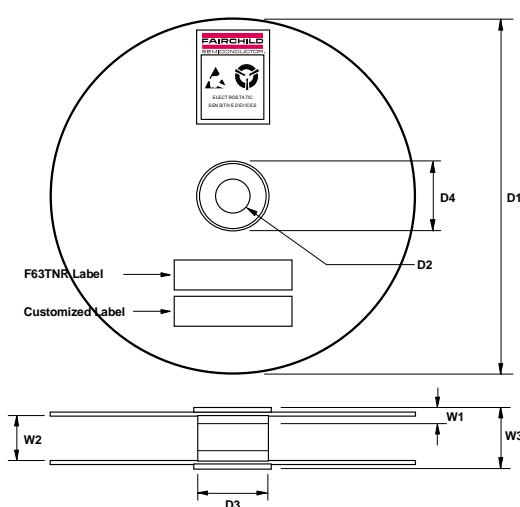
TO-92 Tape and Reel Taping

Dimension Configuration: Figure 4.0



ITEM DESCRIPTION	SYMBOL	DIMENSION
Base of Package to Lead Bend	b	0.098 (max)
Component Height	Ha	0.928 (+/- 0.025)
Lead Clinch Height	HO	0.630 (+/- 0.020)
Component Base Height	H1	0.748 (+/- 0.020)
Component Alignment (side/side)	Pd	0.040 (max)
Component Alignment (front/back)	Hd	0.031 (max)
Component Pitch	P	0.500 (+/- 0.020)
Feed Hole Pitch	PO	0.500 (+/- 0.008)
Hole Center to First Lead	P1	0.150 (+0.009, -0.010)
Hole Center to Component Center	P2	0.247 (+/- 0.007)
Lead Spread	F1/F2	0.104 (+/- 0.010)
Lead Thickness	d	0.018 (+0.002, -0.003)
Cut Lead Length	L	0.429 (max)
Taped Lead Length	L1	0.209 (+0.051, -0.052)
Taped Lead Thickness	t	0.032 (+/- 0.006)
Carrier Tape Thickness	t1	0.021 (+/- 0.006)
Carrier Tape Width	W	0.708 (+0.020, -0.019)
Hold - down Tape Width	WO	0.236 (+/- 0.012)
Hold - down Tape position	W1	0.035 (max)
Feed Hole Position	W2	0.360 (+/- 0.025)
Sprocket Hole Diameter	DO	0.157 (+0.008, -0.007)
Lead Spring Out	S	0.004 (max)

Note : All dimensions are in inches.



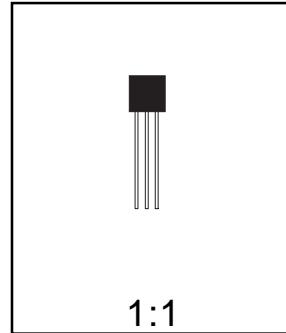
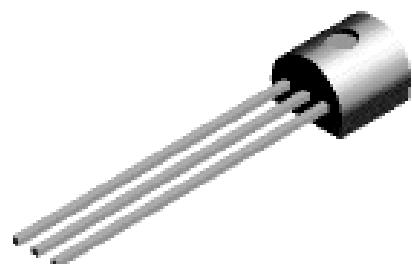
ITEM DESCRIPTION	SYSMBOL	MINIMUM	MAXIMUM
Reel Diameter	D1	13.975	14.025
Arbor Hole Diameter (Standard)	D2	1.160	1.200
(Small Hole)	D2	0.650	0.700
Core Diameter	D3	3.100	3.300
Hub Recess Inner Diameter	D4	2.700	3.100
Hub Recess Depth	W1	0.370	0.570
Flange to Flange Inner Width	W2	1.630	1.690
Hub to Hub Center Width	W3		2.090

Note: All dimensions are inches

TO-92 Package Dimensions

FAIRCHILD
SEMICONDUCTOR™

TO-92 (FS PKG Code 92, 94, 96)



1:1

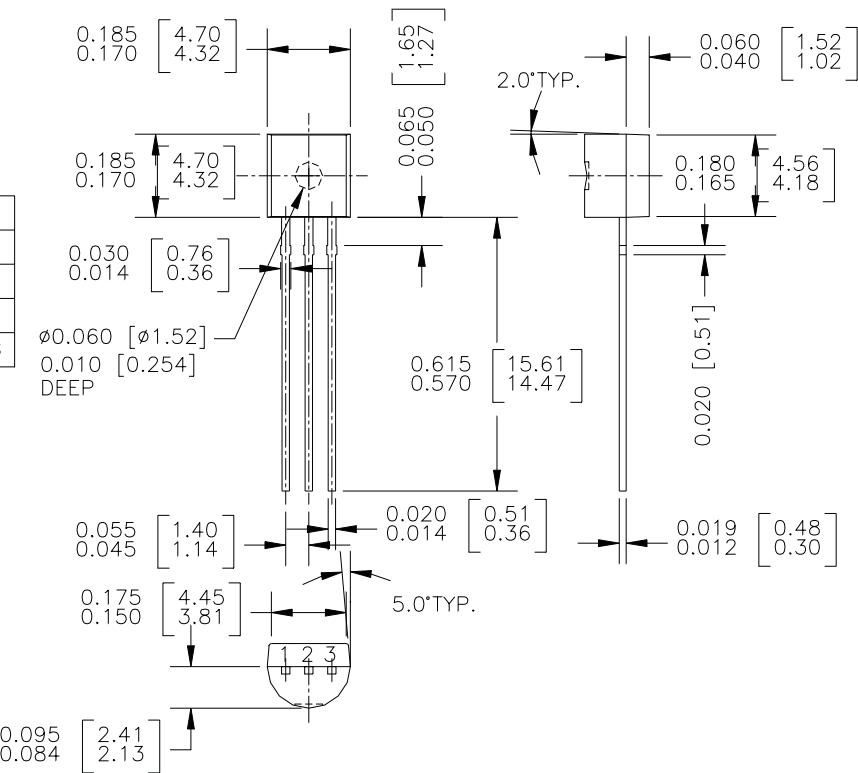
Scale 1:1 on letter size paper

Dimensions shown below are in:
inches [millimeters]

Part Weight per unit (gram): 0.1977

TO-92 (92,94,96)

PIN	92		94		96	
	B	F	B	F	B	F
1	E	D	E	D	B	S
2	B	S	C	G	E	D
3	C	G	B	S	C	G

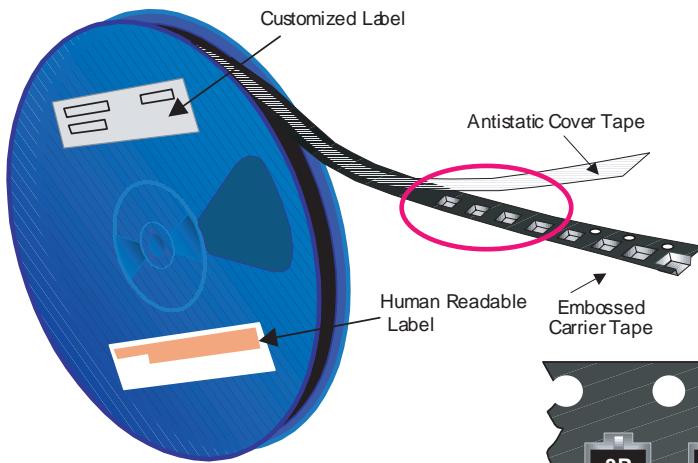


SOT-23 Tape and Reel Data

FAIRCHILD
SEMICONDUCTOR™

SOT-23 Packaging

Configuration: Figure 10

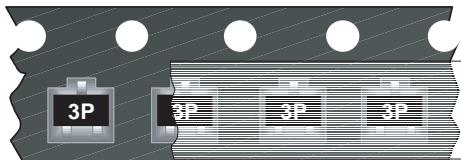


SOT-23 Packaging Information		
Packaging Option	Standard (no flow code)	D87Z
Packaging type	TNR	TNR
Qty per Reel/Tube/Bag	3,000	10,000
Reel Size	7" Dia	13"
Box Dimension (mm)	187x107x183	343x343x64
Max qty per Box	24,000	30,000
Weight per unit (gm)	0.0082	0.0082
Weight per Reel (kg)	0.1175	0.4006
Note/Comments		

Packaging Description:

SOT-23 parts are shipped in tape. The carrier tape is made from a dissipative (carbon filled) polycarbonate resin. The cover tape is a multilayer film (Heat Activated Adhesive in nature) primarily composed of polyester film, adhesive layer, sealant, and anti-static sprayed agent. These reeled parts in standard option are shipped with 3,000 units per 7" or 177cm diameter reel. The reels are dark blue in color and is made of polystyrene plastic (anti-static coated). Other option comes in 10,000 units per 13" or 330cm diameter reel. This and some other options are described in the Packaging Information table.

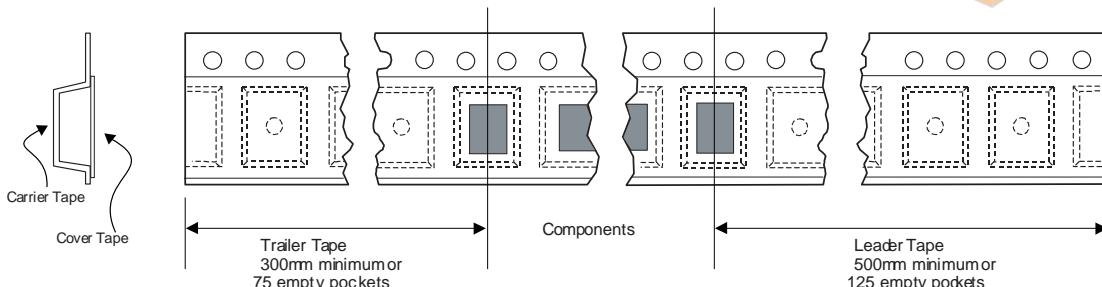
These full reels are individually labeled and placed inside a standard intermediate made of recyclable corrugated brown paper with a Fairchild logo printing. One pizza box contains eight reels maximum. And these intermediate boxes are placed inside a labeled shipping box which comes in different sizes depending on the number of parts shipped.



SOT-23 Unit Orientation



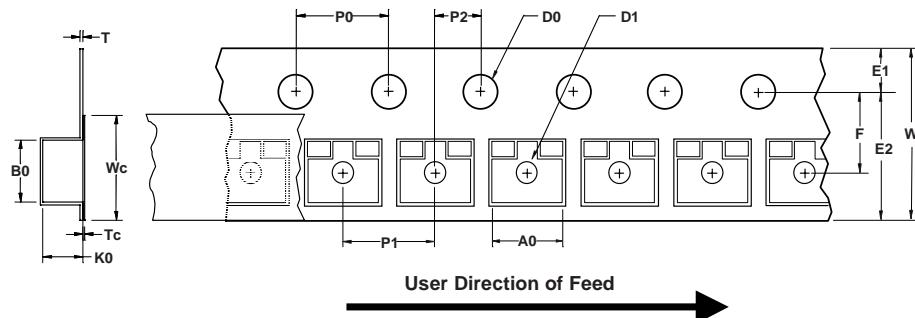
Configuration: Figure 20



SOT-23 Tape and Reel Data, continued

SOT-23 Embossed Carrier Tape

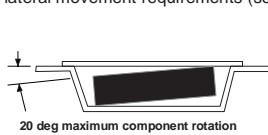
Configuration: Figure 3.0



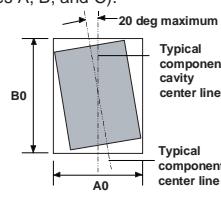
Dimensions are in millimeter

Pkg type	A_0	B_0	W	D_0	D_1	E_1	E_2	F	P_1	P_0	K_0	T	W_c	T_c
SOT-23 (8mm)	3.15 ± 0.10	2.77 ± 0.10	8.0 ± 0.3	1.55 ± 0.05	1.125 ± 0.125	1.75 ± 0.10	6.25 min	3.50 ± 0.05	4.0 ± 0.1	4.0 ± 0.1	1.30 ± 0.10	0.228 ± 0.013	5.2 ± 0.3	0.06 ± 0.02

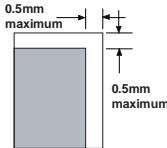
Notes: A_0 , B_0 , and K_0 dimensions are determined with respect to the EIA/Jedec RS-481 rotational and lateral movement requirements (see sketches A, B, and C).



Sketch A (Side or Front Sectional View)
Component Rotation

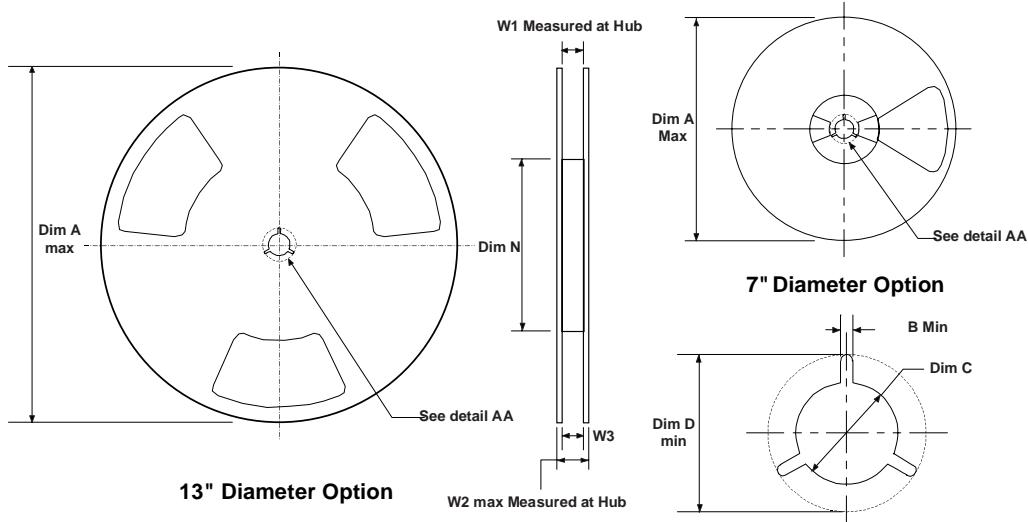


Sketch B (Top View)
Component Rotation



Sketch C (Top View)
Component lateral movement

SOT-23 Reel Configuration: Figure 4.0



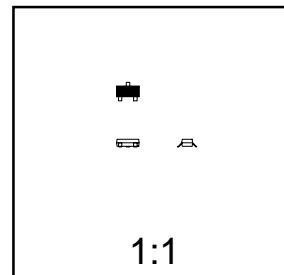
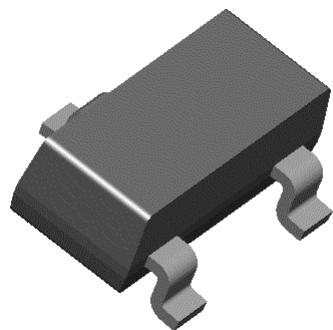
Dimensions are in inches and millimeters

Tape Size	Reel Option	Dim A	Dim B	Dim C	Dim D	Dim N	Dim W1	Dim W2	Dim W3 (LSL-USL)
8mm	7" Dia	7.00 177.8	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	2.165 55	0.331 +0.059/-0.000 8.4 +1.5/0	0.567 14.4	0.311 - 0.429 7.9 - 10.9
8mm	13" Dia	13.00 330	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	4.00 100	0.331 +0.059/-0.000 8.4 +1.5/0	0.567 14.4	0.311 - 0.429 7.9 - 10.9

SOT-23 Package Dimensions



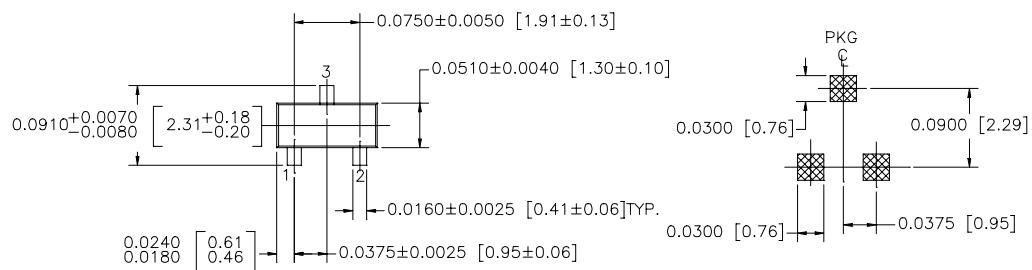
SOT-23 (FS PKG Code 49)



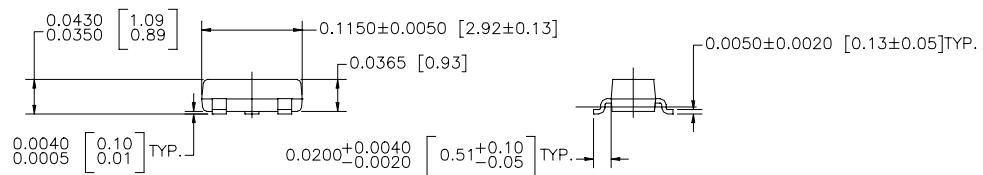
Scale 1:1 on letter size paper

Dimensions shown below are in:
inches [millimeters]

Part Weight per unit (gram): 0.0082



LAND PATTERN RECOMMENDATION



CONTROLLING DIMENSION IS INCH
VALUES IN [] ARE MILLIMETERS

SOT 23, 3 LEADS LOW PROFILE

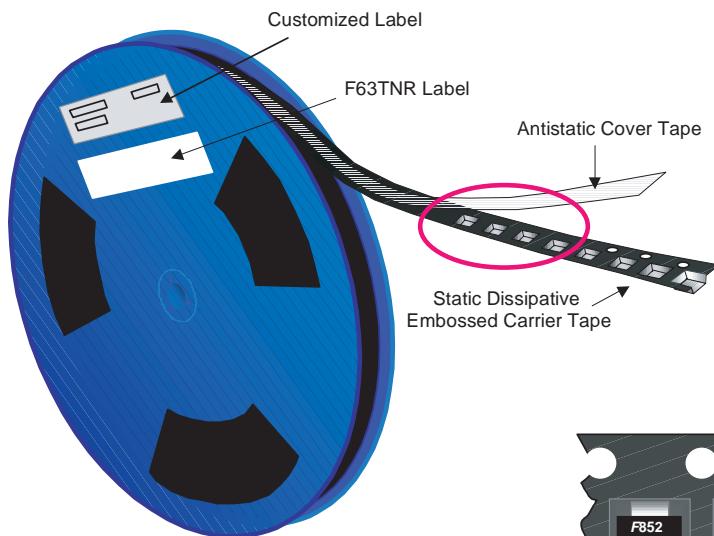
NOTE : UNLESS OTHERWISE SPECIFIED

1. STANDARD LEAD FINISH 150 MICROINCHES / 3.81 MICROMETERS
MINIMUM TIN / LEAD (SOLDER) ON ALLOY 42
2. REFERENCE JEDEC REGISTRATION TO-236, VARIATION AB, ISSUE G, DATED JUL 1993

SOT-223 Tape and Reel Data



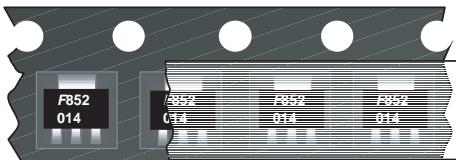
SOT-223 Packaging Configuration: Figure 1.0



Packaging Description:

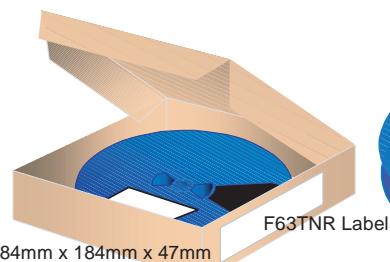
SOT-223 parts are shipped in tape. The carrier tape is made from a dissipative (carbon filled) polycarbonate resin. The cover tape is a multilayer film (Heat Activated Adhesive in nature) primarily composed of polyester film, adhesive layer, sealant, and anti-static sprayed agent. These reeled parts in standard option are shipped with 2,500 units per 13" or 330mm diameter reel. The reels are dark blue in color and is made of polystyrene plastic (anti-static coated). Other option comes in 500 units per 7" or 177mm diameter reel. This and some other options are further described in the Packaging Information table.

These full reels are individually barcode labeled and placed inside a standard intermediate box (illustrated in figure 1.0) made of recyclable corrugated brown paper. One box contains two reels maximum. And these boxes are placed inside a barcode labeled shipping box which comes in different sizes depending on the number of parts shipped.

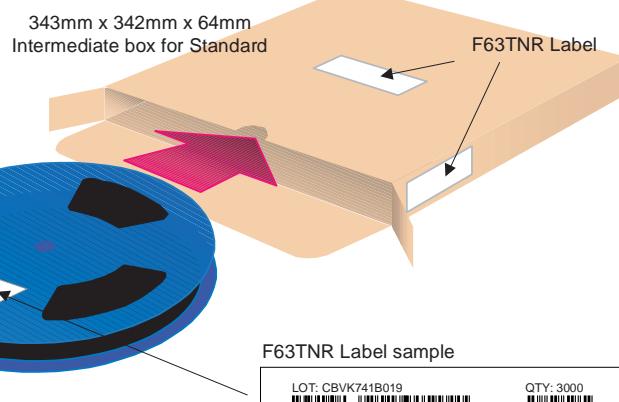


SOT-223 Unit Orientation

SOT-223 Packaging Information		
Packaging Option	Standard (no flow code)	D84Z
Packaging type	TNR	TNR
Qty per Reel/Tube/Bag	2,500	500
Reel Size	13" Dia	7" Dia
Box Dimension (mm)	343x64x343	184x187x47
Max qty per Box	5,000	1,000
Weight per unit (gm)	0.1246	0.1246
Weight per Reel (kg)	0.7250	0.1532
Note/Comments		



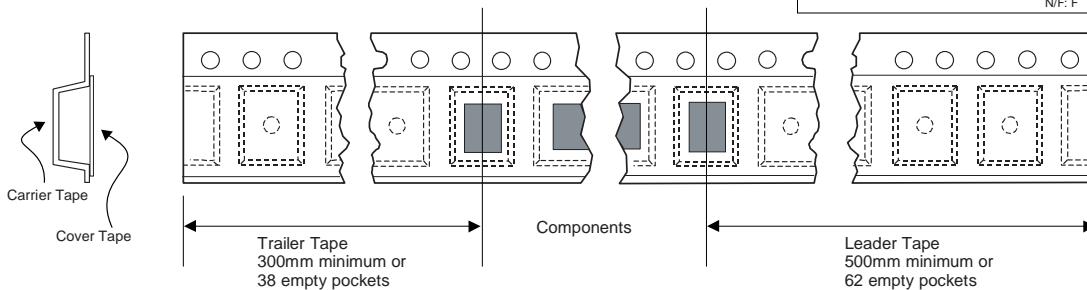
184mm x 184mm x 47mm
Pizza Box for D84Z Option



F63TNR Label sample

LOT: CBVK741B019	QTY: 3000
FSID: PN2222A	SPEC:
D/C1: D9842	QTY1:
D/C2:	QTY2:
	SPEC REV:
	CPN:
	N/F: F
	(F63TNR)3

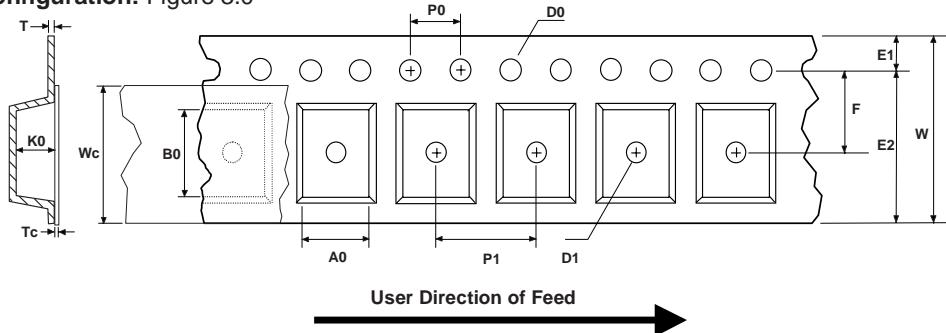
SOT-223 Tape Leader and Trailer Configuration: Figure 2.0



SOT-223 Tape and Reel Data, continued

SOT-223 Embossed Carrier Tape

Configuration: Figure 3.0



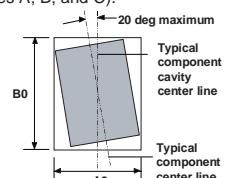
Dimensions are in millimeter

Pkg type	A_0	B_0	W	D_0	D_1	E_1	E_2	F	P_1	P_0	K_0	T	W_c	T_c
SOT-223 (12mm)	6.83 ± 0.10	7.42 ± 0.10	12.0 ± 0.3	1.55 ± 0.05	1.50 ± 0.10	1.75 ± 0.10	10.25 min	5.50 ± 0.05	8.0 ± 0.1	4.0 ± 0.1	1.88 ± 0.10	0.292 ± 0.0130	9.5 ± 0.025	0.06 ± 0.02

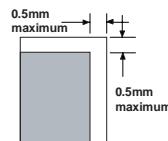
Notes: A_0 , B_0 , and K_0 dimensions are determined with respect to the EIA/Jedec RS-481 rotational and lateral movement requirements (see sketches A, B, and C).



Sketch A (Side or Front Sectional View)
Component Rotation

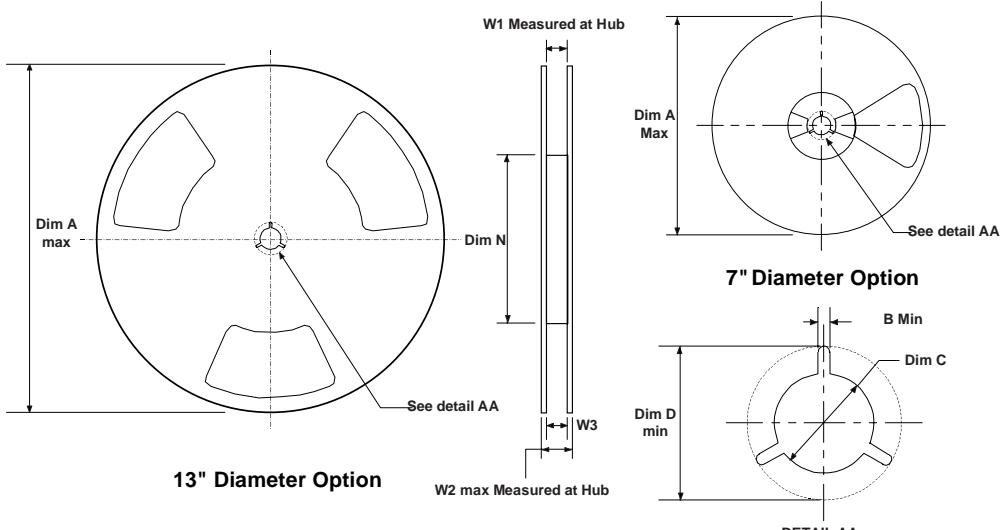


Sketch B (Top View)
Component Rotation



Sketch C (Top View)
Component lateral movement

SOT-223 Reel Configuration: Figure 4.0



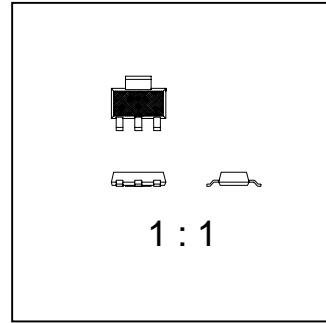
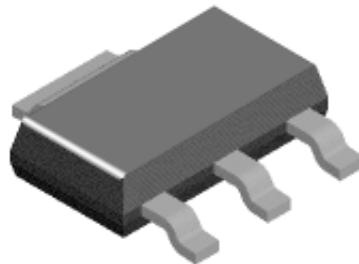
Dimensions are in inches and millimeters

Tape Size	Reel Option	Dim A	Dim B	Dim C	Dim D	Dim N	Dim W1	Dim W2	Dim W3 (LSL-USL)
12mm	7" Dia	7.00 177.8	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	5.906 150	0.488 +0.078/-0.000 12.4 +2/0	0.724 18.4	0.469 - 0.606 11.9 - 15.4
12mm	13" Dia	13.00 330	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	7.00 178	0.488 +0.078/-0.000 12.4 +2/0	0.724 18.4	0.469 - 0.606 11.9 - 15.4

SOT-223 Package Dimensions

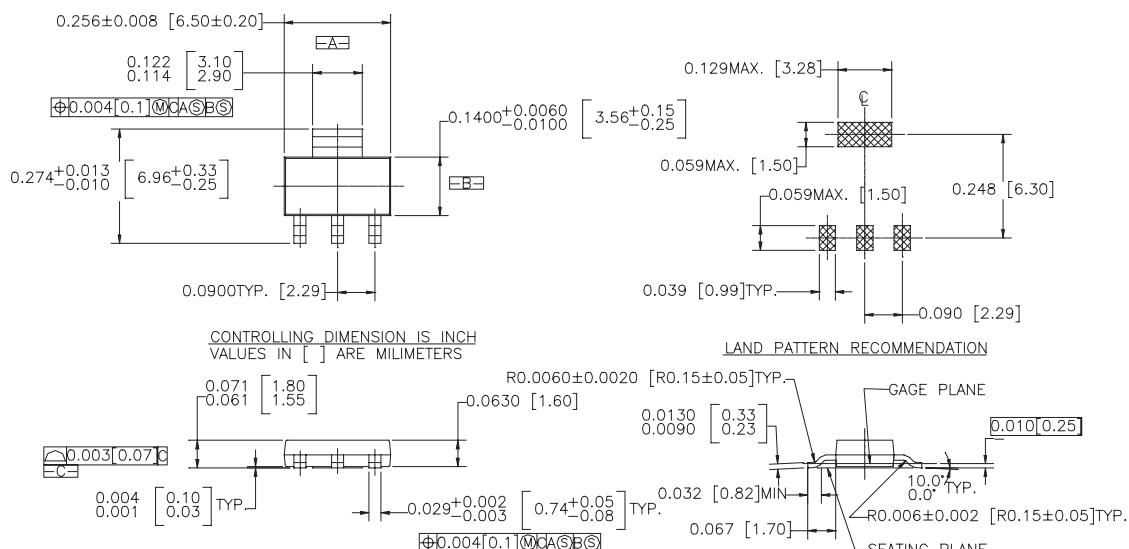
FAIRCHILD
SEMICONDUCTOR™

SOT-223 (FS PKG Code 47)



Scale 1:1 on letter size paper

Part Weight per unit (gram): 0.1246



TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™	FASTR™	PowerTrench®	SyncFET™
Bottomless™	GlobalOptoisolator™	QFET™	TinyLogic™
CoolFET™	GTO™	QS™	UHC™
CROSSVOLT™	HiSeC™	QT Optoelectronics™	VCX™
DOME™	ISOPLANAR™	Quiet Series™	
E ² CMOS™	MICROWIRE™	SILENT SWITCHER®	
EnSigna™	OPTOLOGIC™	SMART START™	
FACT™	OPTOPLANAR™	SuperSOT™-3	
FACT Quiet Series™	PACMAN™	SuperSOT™-6	
FAST®	POP™	SuperSOT™-8	

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

TRANSIL™ array for data protection

Main applications

Where transient overvoltage protection in ESD sensitive equipment is required, such as :

- Computers
- Printers
- Communication systems
- Cellular phones handsets and accessories
- Wireline and wireless telephone sets
- Set top boxes



Features

- 2 up to 5 Unidirectional Transil functions
- Breakdown voltage:
 $V_{BR} = 6.1 \text{ V min. and } 25 \text{ V min.}$
- Low leakage current: $< 1 \mu\text{A}$
- Very small PCB area $< 4.2 \text{ mm}^2$ typically

Description

The ESDAxxxWx are monolithic suppressors designed to protect components connected to data and transmission lines against ESD.

These devices clamp the voltage just above the logic level supply for positive transients, and to a diode drop below ground for negative transients.

Benefits

- High ESD protection level: up to 25 kV
- High integration

Complies with the following standards

IEC61000-4-2

Level 4 15 kV (air discharge)
 8 kV(contact discharge)

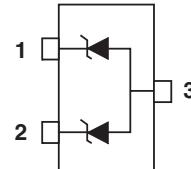
MIL STD 883E - Method 3015-7 Class 3

25 kV HBM (Human Body Model)

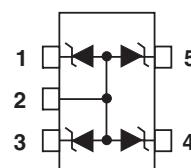
Order codes

Part Numbers	Marking
ESDA6V1W5	E61
ESDA6V1-5W6	E62
ESDA25W	E25
ESDA25W5	E25

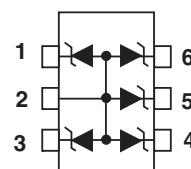
ESDA25W Functional diagram



ESDA6V1W5/ESDA25W5 Functional diagram



ESDA6V1-5W6 Functional diagram



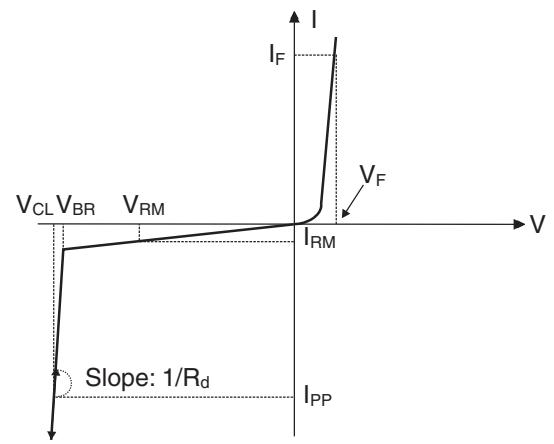
1 Characteristics

Table 1. Absolute Ratings ($T_{amb} = 25^{\circ}\text{C}$)

Symbol	Parameter	Value	Unit
P_{PP}	Peak pulse power (8/20 μs)	ESDA25W	W
		ESDA25W5 / ESDA6V1W5	
		ESDA6V1-5W6	
T_j	Junction temperature	125	$^{\circ}\text{C}$
T_{stg}	Storage temperature range	-55 to +150	$^{\circ}\text{C}$
T_L	Maximum lead temperature for soldering during 10s	260	$^{\circ}\text{C}$
T_{op}	Operating temperature range ⁽¹⁾	ESDA25W / ESDA25W5 / ESDA6V1W5	$^{\circ}\text{C}$
		ESDA6V1-5W6	

1. The values of the operating parameters versus temperature are given through curves and αT parameter.

1.1 Electrical Characteristics ($T_{amb} = 25^{\circ}\text{C}$)

Symbol	Parameter	 <p>The graph shows the relationship between forward current (I_F) and forward voltage (V_F). Key points marked on the curve are V_CL, V_BR, V_RM, I_RM, and I_PP. A dashed line from V_CL to the curve is labeled 'Slope: 1/R_d'.</p>
V_{RM}	Stand-off voltage	
V_{BR}	Breakdown voltage	
V_{CL}	Clamping voltage	
I_{RM}	Leakage current	
I_{PP}	Peak pulse current	
I_R	Reverse leakage current	
I_F	Forward current	
αT	Voltage temperature coefficient	
V_F	Forward voltage drop	
C	Capacitance	
R_d	Dynamic resistance	

Part Numbers	V _{BR}			I _{RM} @ V _{RM}	V _F @ I _F	R _d	αT	C		
	min.	max.	@ I _R	max.	typ. ⁽¹⁾	typ. ⁽²⁾	typ.			
	V	V	mA	μA	V	mA	Ω	10 ⁻⁴ /°C	pF	
ESDA25W	25	30	1	1	24	1.2	10	1.1	10	65
ESDA25W5	25	30	1	1	24	1.2	10	1.9	10	30
ESDA6V1-5W6	6.1	7.2	1	1	3	1.25	200	0.61	6	50
ESDA6V1W5	6.1	7.2	1	1	3	1.25	200	0.35	6	90

1. Square pulse $I_{pp} = 15 \text{ A}$, $t_p = 2.5 \mu\text{s}$
 2. $V_{BR} = aT^* (T_{amb} - 25^\circ\text{C}) * V_{BR} (25^\circ\text{C})$

Figure 1. Peak power dissipation versus initial junction temperature

Figure 2. Peak pulse power versus exponential pulse duration (T_j initial = 25°C) (ESDA25W)

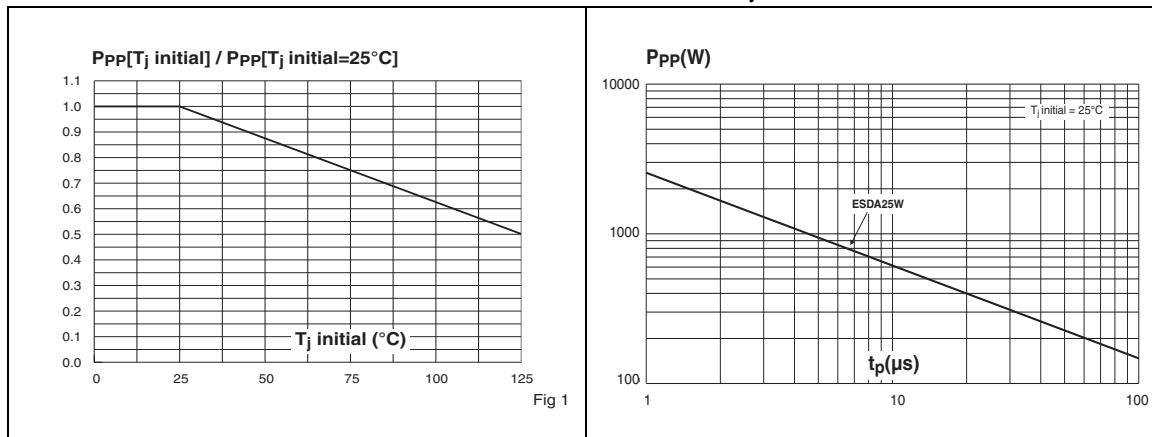


Figure 3. Peak pulse power versus exponential pulse duration (T_j initial = 25°C) (ESDA25W5 / ESDA6V1W5 / ESDA6V1-5W6)

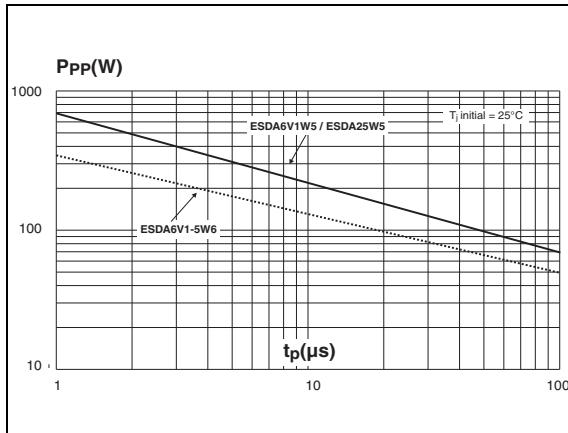


Figure 4. Clamping voltage versus peak pulse current (T_j initial = 25°C, rectangular waveform, t_p = 2.5 μs) (ESDA25W / ESDA25W5)

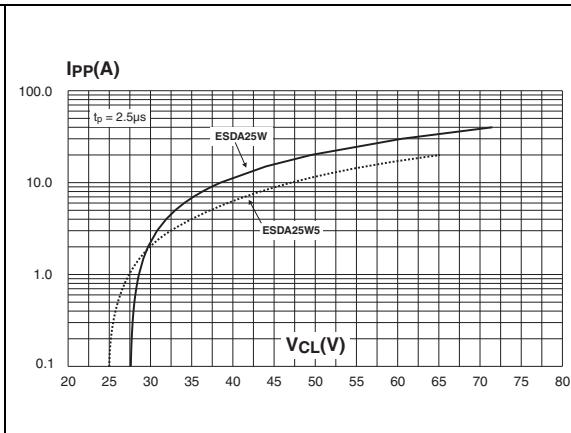


Figure 5. Clamping voltage versus peak pulse current (T_j initial = 25°C, rectangular waveform, t_p = 2.5 μs) (ESDA6V1W5 / ESDA6V1-5W6)

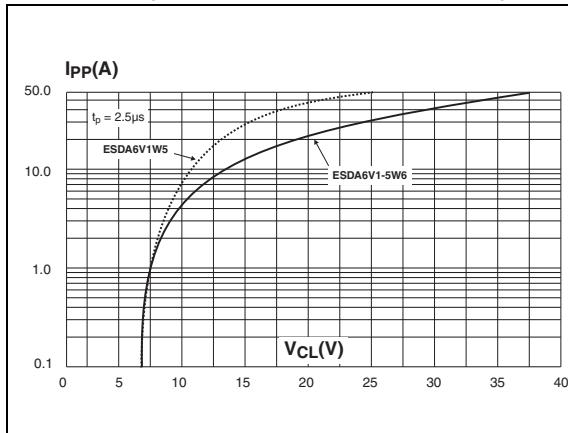


Figure 6. Capacitance versus reverse applied voltage (typical values) (ESDA25W / ESDA25W5)

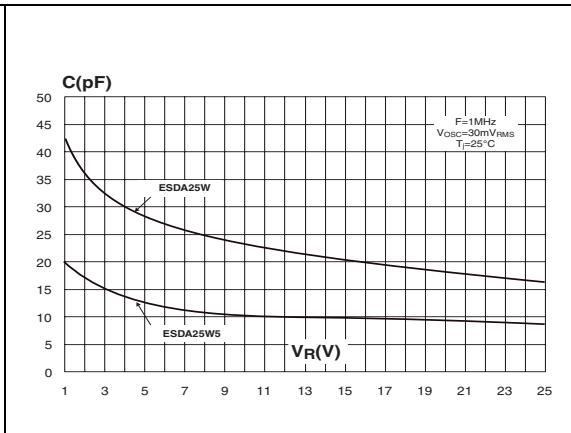


Figure 7. Capacitance versus reverse applied voltage (typical values) (ESDA6V1W5 / ESDA6V1-5W6)

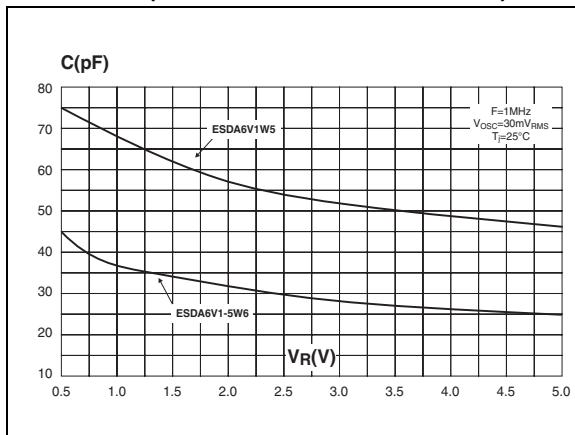


Figure 8. Relative variation of leakage current versus junction temperature (typical values)

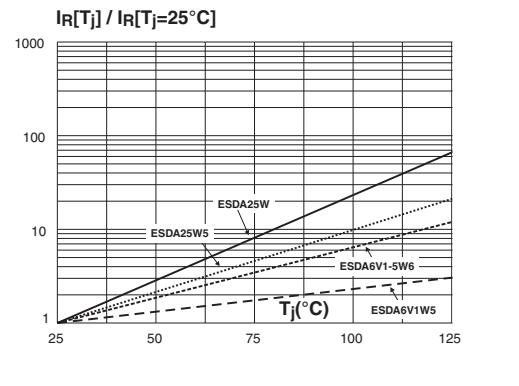


Figure 9. Peak forward voltage drop versus peak forward current (typical values) (ESDA25W / ESDA25W5)

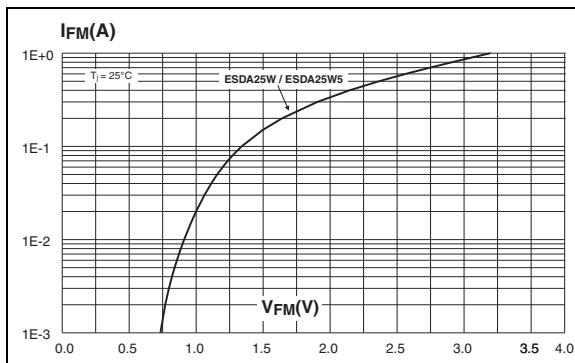


Figure 10. Peak forward voltage drop versus peak forward current (typical values) (ESDA6V1W5 / ESDA6V1-5W6)

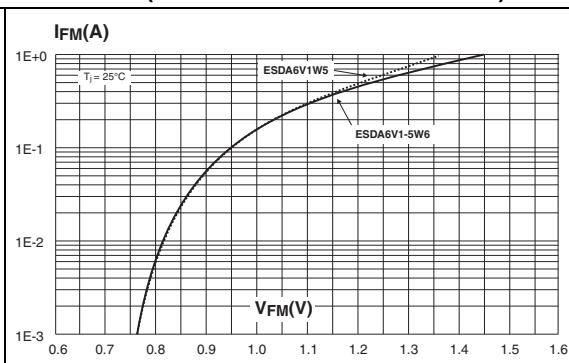
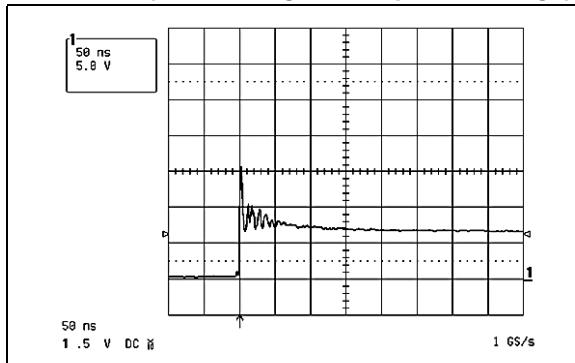
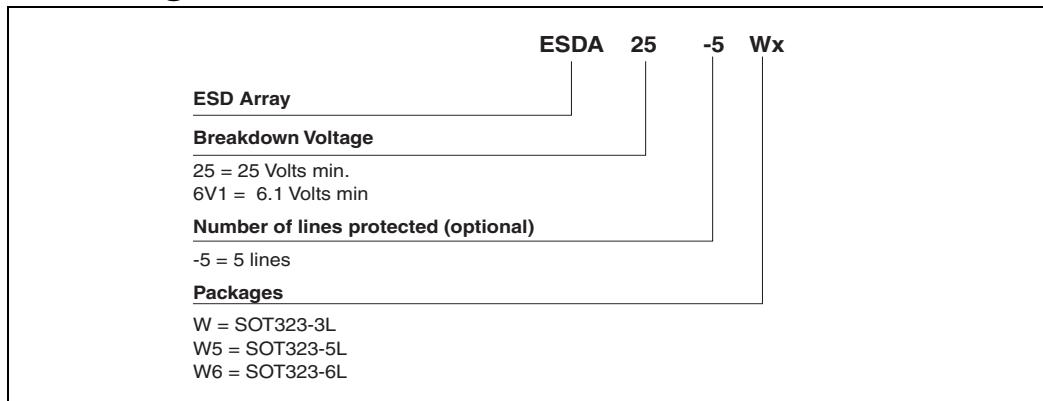


Figure 11. ESD response to IEC61000-4-2 (air discharge 15 kV, positive surge)

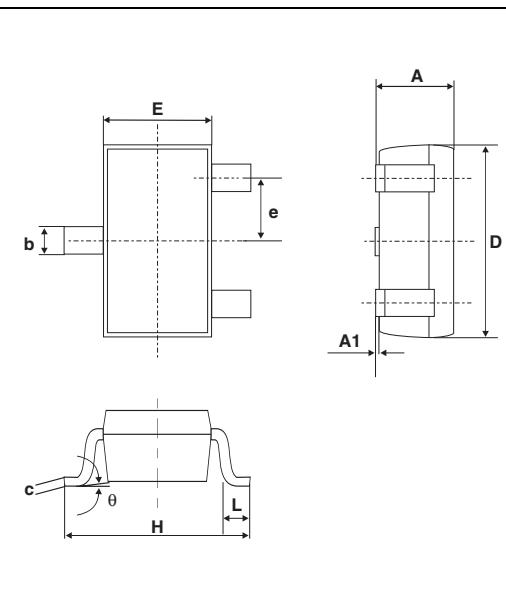


2 Ordering information scheme



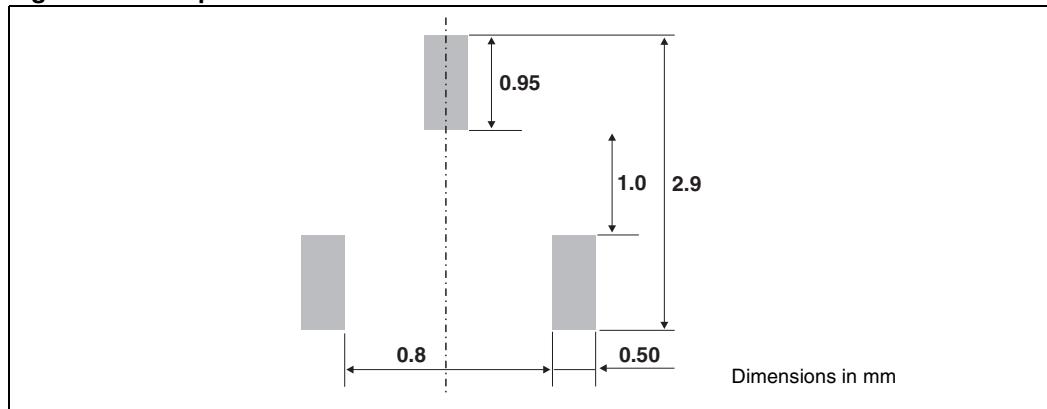
3 Package mechanical data

3.1 SOT323-3L package

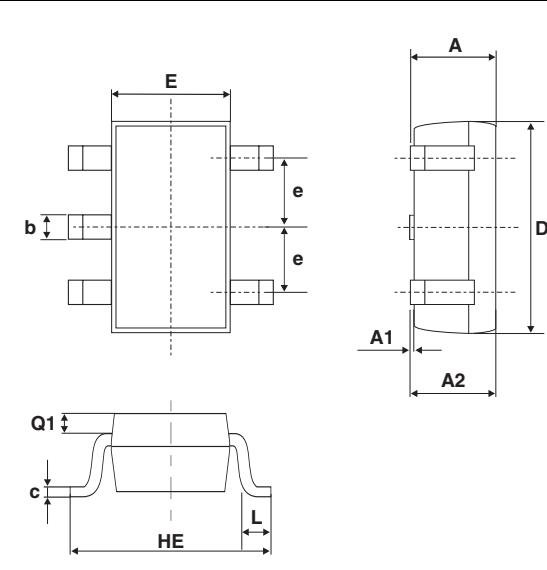


REF.	DIMENSIONS					
	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A	0.8		1.1	0.031		0.043
A1	0.0		0.1	0.0		0.004
b	0.25		0.4	0.010		0.016
c	0.1		0.26	0.004		0.010
D	1.8	2.0	2.2	0.071	0.079	0.086
E	1.15	1.25	1.35	0.045	0.049	0.053
e		0.65			0.026	
H	1.8	2.1	2.4	0.071	0.083	0.094
L	0.1	0.2	0.3	0.004	0.008	0.012
q	0		30°	0		30°

Figure 12. Footprint dimensions

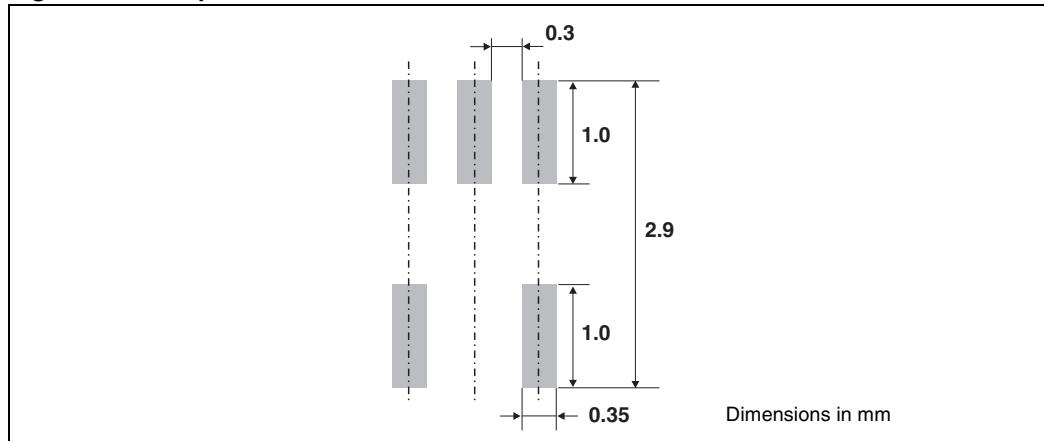


3.2 SOT323-5L package

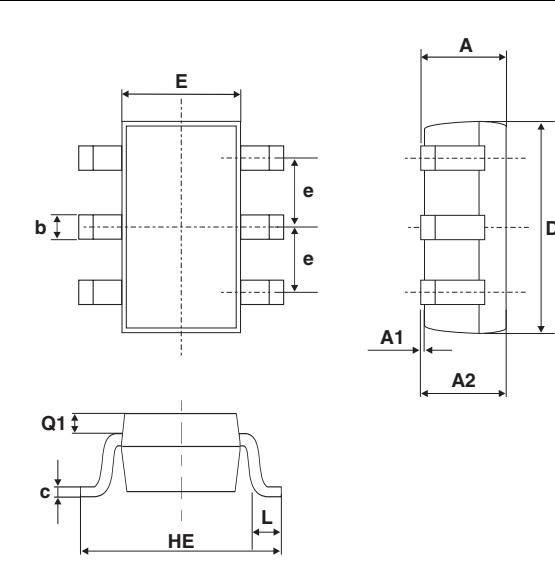


REF.	DIMENSIONS			
	Millimeters		Inches	
	Min.	Max.	Min.	Max.
A	0.8	1.1	0.031	0.043
A1	0	0.1	0	0.004
A2	0.8	1	0.031	0.039
b	0.15	0.3	0.006	0.012
c	0.1	0.18	0.004	0.007
D	1.8	2.2	0.071	0.086
E	1.15	1.35	0.045	0.053
e	0.65 Typ.		0.025 Typ.	
H	1.8	2.4	0.071	0.094
Q1	0.1	0.4	0.004	0.016

Figure 13. Footprint dimensions

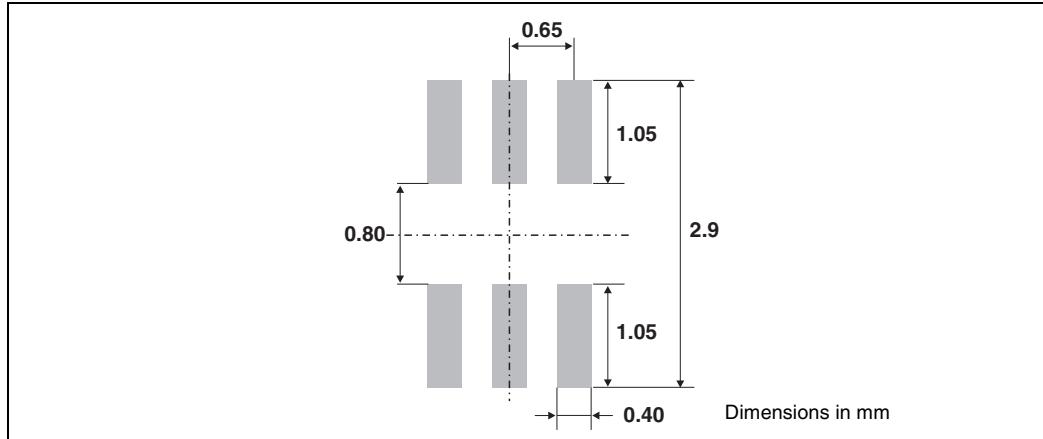


3.3 SOT323-6L package



REF.	DIMENSIONS			
	Millimeters		Inches	
	Min.	Max.	Min.	Max.
A	0.8	1.1	0.031	0.043
A1	0	0.1	0	0.004
A2	0.8	1	0.031	0.039
b	0.15	0.3	0.006	0.012
c	0.1	0.18	0.004	0.007
D	1.8	2.2	0.071	0.086
E	1.15	1.35	0.045	0.053
e	0.65 Typ.		0.025 Typ.	
H	1.8	2.4	0.071	0.094
Q	0.1	0.4	0.004	0.016

Figure 14. Footprint dimensions



4 Ordering information

Part Number	Marking	Package	Weight	Base qty	Delivery mode
ESDA6V1W5	E61	SOT323-5L	6 mg	3000	Tape & reel
ESDA6V1-5W6	E62	SOT323-6L			
ESDA25W	E25	SOT323-3L			
ESDA25W5	E25	SOT323-5L			

5 Revision history

Date	Revision	Changes
20-Jul-2005	1	Initial release
29-Aug-2005	2	Added notes to table on page2, removed annotations in Figure 1.

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics.
All other names are the property of their respective owners

© 2005 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -
Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com



HD44780U (LCD-II)

(Dot Matrix Liquid Crystal Display Controller/Driver)

HITACHI

ADE-207-272(Z)

'99.9

Rev. 0.0

Description

The HD44780U dot-matrix liquid crystal display controller and driver LSI displays alphanumerics, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

A single HD44780U can display up to one 8-character line or two 8-character lines.

The HD44780U has pin function compatibility with the HD44780S which allows the user to easily replace an LCD-II with an HD44780U. The HD44780U character generator ROM is extended to generate 208 5 × 8 dot character fonts and 32 5 × 10 dot character fonts for a total of 240 different character fonts.

The low power supply (2.7V to 5.5V) of the HD44780U is suitable for any portable battery-driven product requiring low power dissipation.

Features

- 5 × 8 and 5 × 10 dot matrix possible
- Low power operation support:
 - 2.7 to 5.5V
- Wide range of liquid crystal display driver power
 - 3.0 to 11V
- Liquid crystal drive waveform
 - A (One line frequency AC waveform)
- Correspond to high speed MPU bus interface
 - 2 MHz (when V_{CC} = 5V)
- 4-bit or 8-bit MPU interface enabled
- 80 × 8-bit display RAM (80 characters max.)
- 9,920-bit character generator ROM for a total of 240 character fonts
 - 208 character fonts (5 × 8 dot)
 - 32 character fonts (5 × 10 dot)

HD44780U

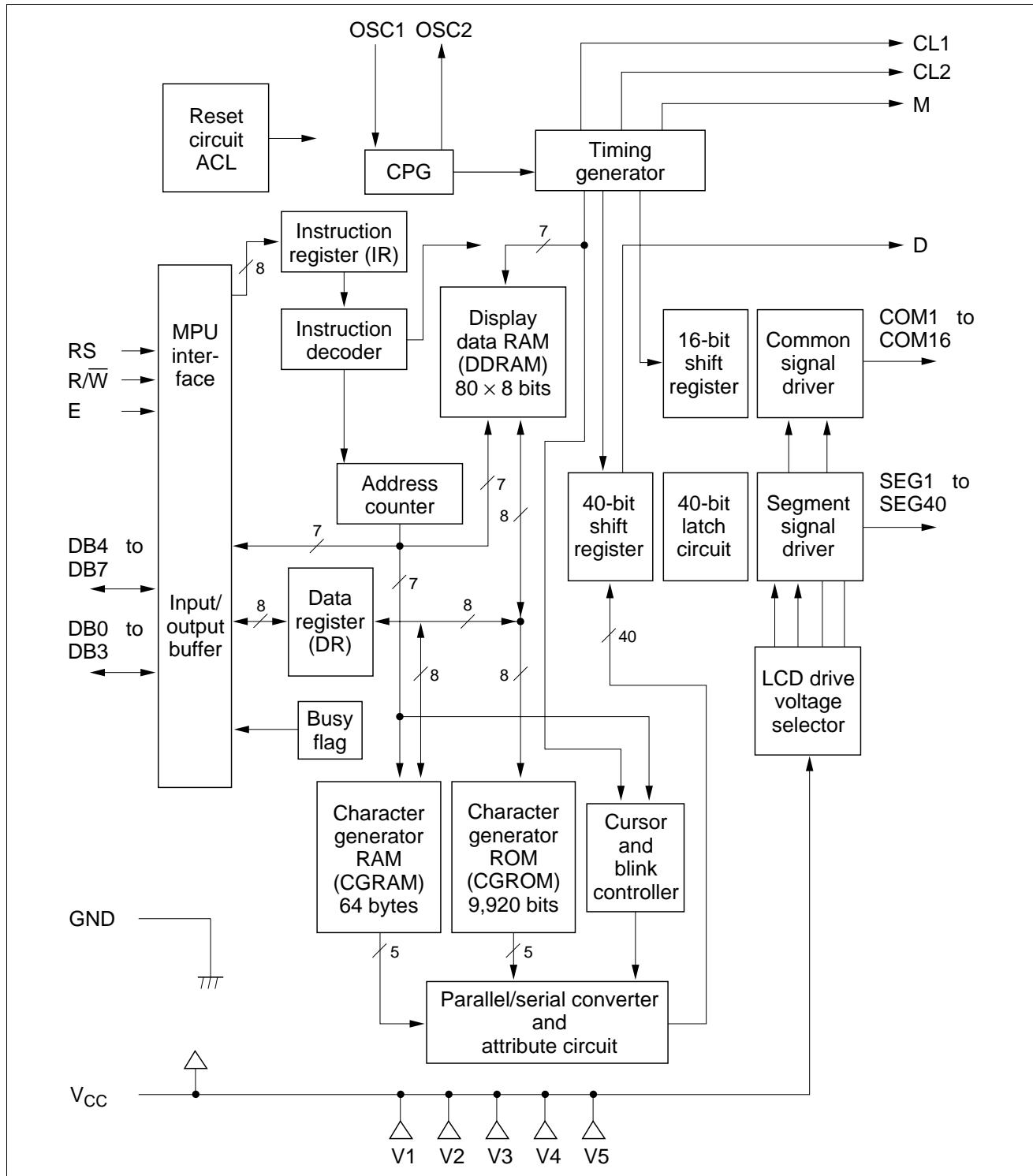
- 64 × 8-bit character generator RAM
 - 8 character fonts (5 × 8 dot)
 - 4 character fonts (5 × 10 dot)
- 16-common × 40-segment liquid crystal display driver
- Programmable duty cycles
 - 1/8 for one line of 5 × 8 dots with cursor
 - 1/11 for one line of 5 × 10 dots with cursor
 - 1/16 for two lines of 5 × 8 dots with cursor
- Wide range of instruction functions:
 - Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
- Pin function compatibility with HD44780S
- Automatic reset circuit that initializes the controller/driver after power on
- Internal oscillator with external resistors
- Low power consumption

Ordering Information

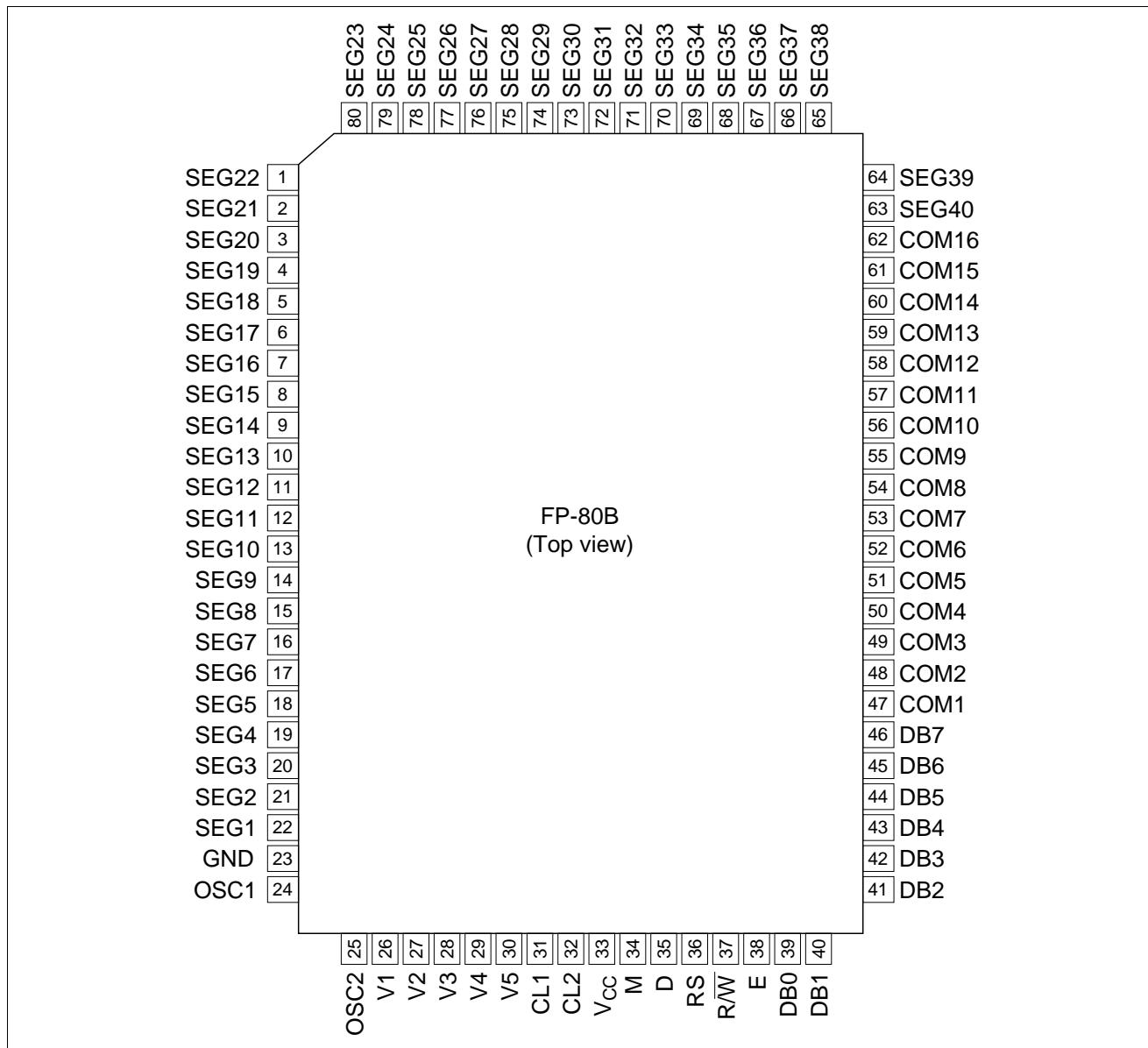
Type No.	Package	CGROM
HD44780UA00FS	FP-80B	Japanese standard font
HCD44780UA00	Chip	
HD44780UA00TF	TFP-80F	
HD44780UA02FS	FP-80B	European standard font
HCD44780UA02	Chip	
HD44780UA02TF	TFP-80F	
HD44780UBxxFS	FP-80B	Custom font
HCD44780UBxx	Chip	
HD44780UBxxTF	TFP-80F	

Note: xx: ROM code No.

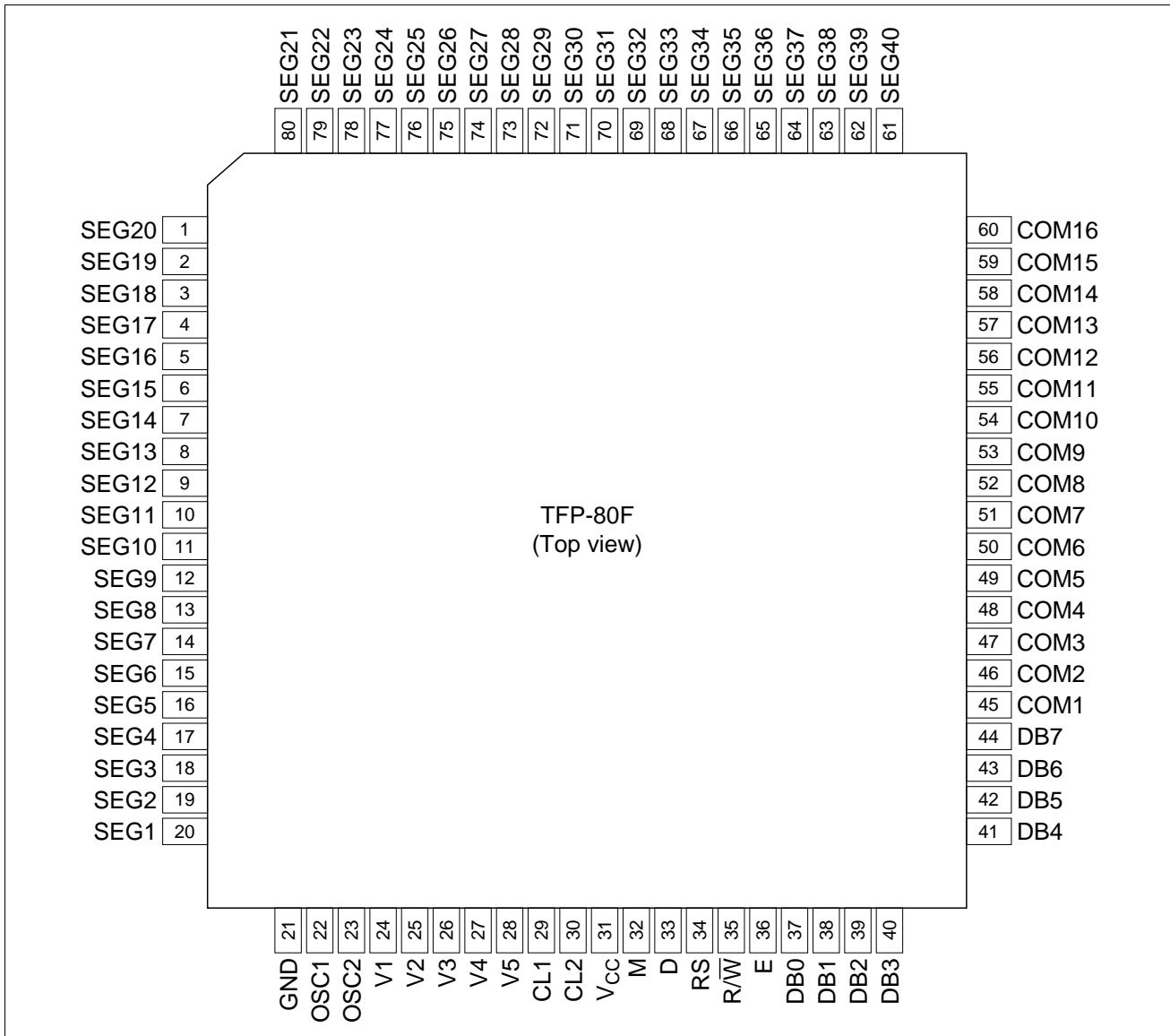
HD44780U Block Diagram



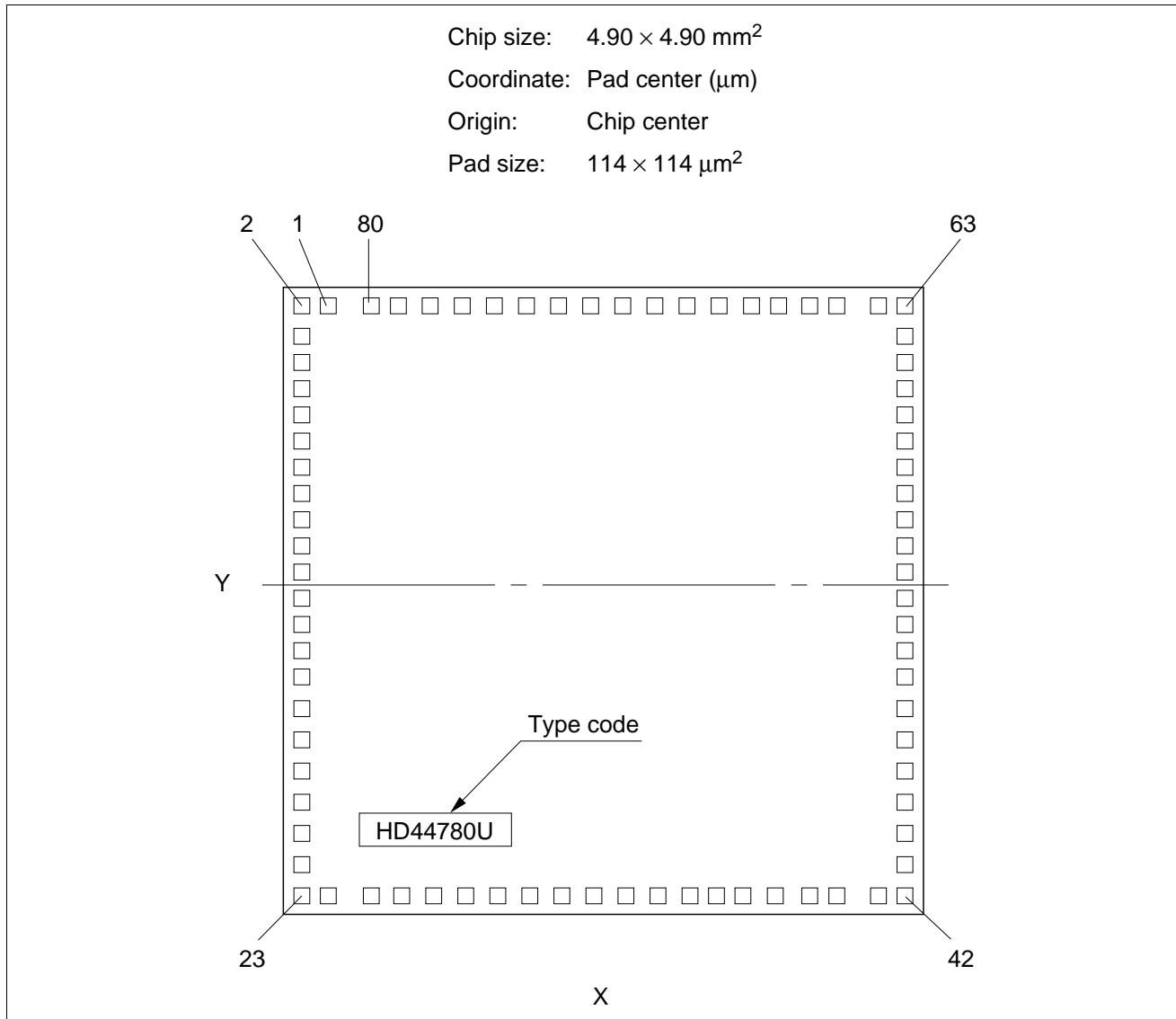
HD44780U Pin Arrangement (FP-80B)



HD44780U Pin Arrangement (TFP-80F)



HD44780U Pad Arrangement



HCD44780U Pad Location Coordinates

Pad No.	Function	Coordinate		Pad No.	Function	Coordinate	
		X (um)	Y (um)			X (um)	Y (um)
1	SEG22	-2100	2313	41	DB2	2070	-2290
2	SEG21	-2280	2313	42	DB3	2260	-2290
3	SEG20	-2313	2089	43	DB4	2290	-2099
4	SEG19	-2313	1833	44	DB5	2290	-1883
5	SEG18	-2313	1617	45	DB6	2290	-1667
6	SEG17	-2313	1401	46	DB7	2290	-1452
7	SEG16	-2313	1186	47	COM1	2313	-1186
8	SEG15	-2313	970	48	COM2	2313	-970
9	SEG14	-2313	755	49	COM3	2313	-755
10	SEG13	-2313	539	50	COM4	2313	-539
11	SEG12	-2313	323	51	COM5	2313	-323
12	SEG11	-2313	108	52	COM6	2313	-108
13	SEG10	-2313	-108	53	COM7	2313	108
14	SEG9	-2313	-323	54	COM8	2313	323
15	SEG8	-2313	-539	55	COM9	2313	539
16	SEG7	-2313	-755	56	COM10	2313	755
17	SEG6	-2313	-970	57	COM11	2313	970
18	SEG5	-2313	-1186	58	COM12	2313	1186
19	SEG4	-2313	-1401	59	COM13	2313	1401
20	SEG3	-2313	-1617	60	COM14	2313	1617
21	SEG2	-2313	-1833	61	COM15	2313	1833
22	SEG1	-2313	-2073	62	COM16	2313	2095
23	GND	-2280	-2290	63	SEG40	2296	2313
24	OSC1	-2080	-2290	64	SEG39	2100	2313
25	OSC2	-1749	-2290	65	SEG38	1617	2313
26	V1	-1550	-2290	66	SEG37	1401	2313
27	V2	-1268	-2290	67	SEG36	1186	2313
28	V3	-941	-2290	68	SEG35	970	2313
29	V4	-623	-2290	69	SEG34	755	2313
30	V5	-304	-2290	70	SEG33	539	2313
31	CL1	-48	-2290	71	SEG32	323	2313
32	CL2	142	-2290	72	SEG31	108	2313
33	V _{cc}	309	-2290	73	SEG30	-108	2313
34	M	475	-2290	74	SEG29	-323	2313
35	D	665	-2290	75	SEG28	-539	2313
36	RS	832	-2290	76	SEG27	-755	2313
37	R/W	1022	-2290	77	SEG26	-970	2313
38	E	1204	-2290	78	SEG25	-1186	2313
39	DB0	1454	-2290	79	SEG24	-1401	2313
40	DB1	1684	-2290	80	SEG23	-1617	2313

Pin Functions

Signal	No. of Lines	I/O	Device Interfaced with	Function
RS	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/W	1	I	MPU	Selects read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.
DB0 to DB3	4	I/O	MPU	Four low order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. These pins are not used during 4-bit operation.
CL1	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
CL2	1	O	Extension driver	Clock to shift serial data D
M	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
D	1	O	Extension driver	Character pattern data corresponding to each segment signal
COM1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveforms. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.
SEG1 to SEG40	40	O	LCD	Segment signals
V1 to V5	5	—	Power supply	Power supply for LCD drive $V_{cc} - V_5 = 11\text{ V (max)}$
V_{cc} , GND	2	—	Power supply	V_{cc} : 2.7V to 5.5V, GND: 0V
OSC1, OSC2	2	—	Oscillation resistor clock	When crystal oscillation is performed, a resistor must be connected externally. When the pin input is an external clock, it must be input to OSC1.

Function Description

Registers

The HD44780U has two 8-bit registers, an instruction register (IR) and a data register (DR).

The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM) and character generator RAM (CGRAM). The IR can only be written from the MPU.

The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR from the MPU is automatically written into DDRAM or CGRAM by an internal operation. The DR is also used for data storage when reading data from DDRAM or CGRAM. When address information is written into the IR, data is read and then stored into the DR from DDRAM or CGRAM by an internal operation. Data transfer between the MPU is then completed when the MPU reads the DR. After the read, data in DDRAM or CGRAM at the next address is sent to the DR for the next read from the MPU. By the register selector (RS) signal, these two registers can be selected (Table 1).

Busy Flag (BF)

When the busy flag is 1, the HD44780U is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/W = 1 (Table 1), the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)

The address counter (AC) assigns addresses to both DDRAM and CGRAM. When an address of an instruction is written into the IR, the address information is sent from the IR to the AC. Selection of either DDRAM or CGRAM is also determined concurrently by the instruction.

After writing into (reading from) DDRAM or CGRAM, the AC is automatically incremented by 1 (decremented by 1). The AC contents are then output to DB0 to DB6 when RS = 0 and R/W = 1 (Table 1).

Table 1 Register Selection

RS	R/W	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Display Data RAM (DDRAM)

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80×8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses and positions on the liquid crystal display.

The DDRAM address (A_{DD}) is set in the address counter (AC) as hexadecimal.

- 1-line display ($N = 0$) (Figure 2)
 - When there are fewer than 80 display characters, the display begins at the head position. For example, if using only the HD44780, 8 characters are displayed. See Figure 3.
 - When the display shift operation is performed, the DDRAM address shifts. See Figure 3.

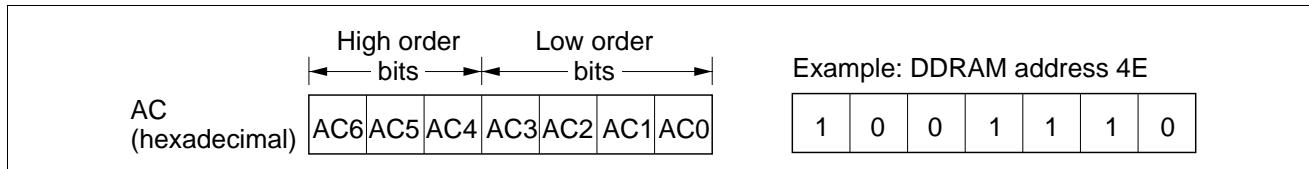


Figure 1 DDRAM Address

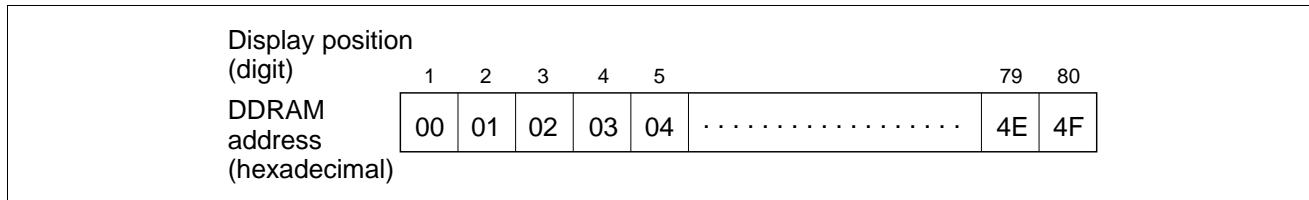


Figure 2 1-Line Display

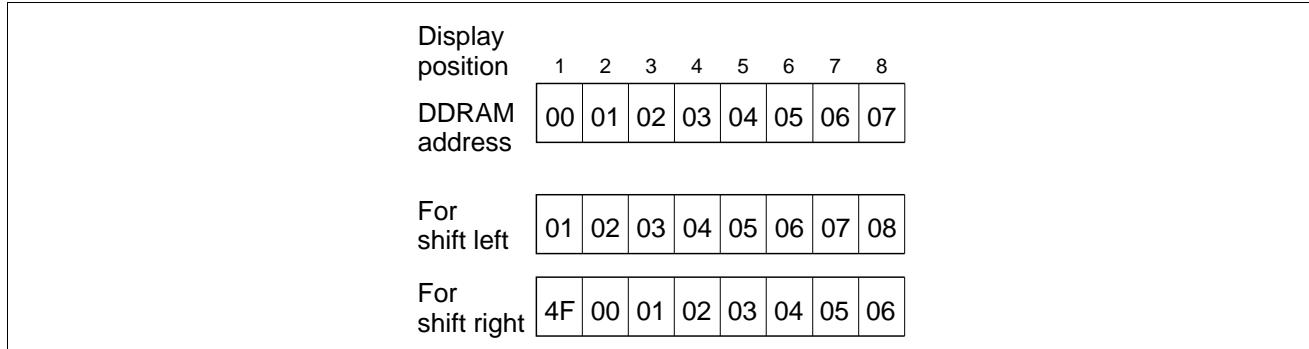


Figure 3 1-Line by 8-Character Display Example

- 2-line display ($N = 1$) (Figure 4)

— Case 1: When the number of display characters is less than 40×2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters \times 2 lines are displayed. See Figure 5.

When display shift operation is performed, the DDRAM address shifts. See Figure 5.

Display position	1	2	3	4	5		39	40
DDRAM address (hexadecimal)	00	01	02	03	04	26	27
	40	41	42	43	44	66	67

Figure 4 2-Line Display

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07
	40	41	42	43	44	45	46	47

For shift left	01	02	03	04	05	06	07	08
	41	42	43	44	45	46	47	48

For shift right	27	00	01	02	03	04	05	06
	67	40	41	42	43	44	45	46

Figure 5 2-Line by 8-Character Display Example

HD44780U

- Case 2: For a 16-character × 2-line display, the HD44780 can be extended using one 40-output extension driver. See Figure 6.

When display shift operation is performed, the DDRAM address shifts. See Figure 6.

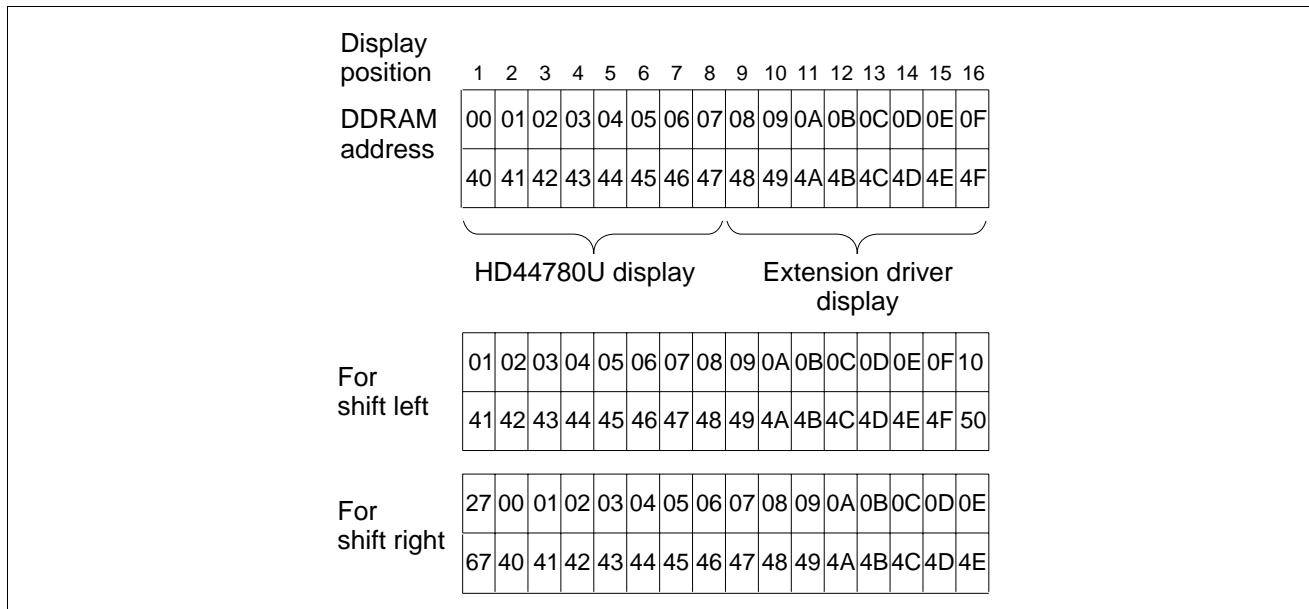


Figure 6 2-Line by 16-Character Display Example

Character Generator ROM (CGROM)

The character generator ROM generates 5×8 dot or 5×10 dot character patterns from 8-bit character codes (Table 4). It can generate 208 5×8 dot character patterns and 32 5×10 dot character patterns. User-defined character patterns are also available by mask-programmed ROM.

Character Generator RAM (CGRAM)

In the character generator RAM, the user can rewrite character patterns by program. For 5×8 dots, eight character patterns can be written, and for 5×10 dots, four character patterns can be written.

Write into DDRAM the character codes at the addresses shown as the left column of Table 4 to show the character patterns stored in CGRAM.

See Table 5 for the relationship between CGRAM addresses and data and display patterns.

Areas that are not used for display can be used as general data RAM.

Modifying Character Patterns

- Character pattern development procedure

The following operations correspond to the numbers listed in Figure 7:

1. Determine the correspondence between character codes and character patterns.
2. Create a listing indicating the correspondence between EPROM addresses and data.
3. Program the character patterns into the EPROM.
4. Send the EPROM to Hitachi.
5. Computer processing on the EPROM is performed at Hitachi to create a character pattern listing, which is sent to the user.
6. If there are no problems within the character pattern listing, a trial LSI is created at Hitachi and samples are sent to the user for evaluation. When it is confirmed by the user that the character patterns are correctly written, mass production of the LSI proceeds at Hitachi.

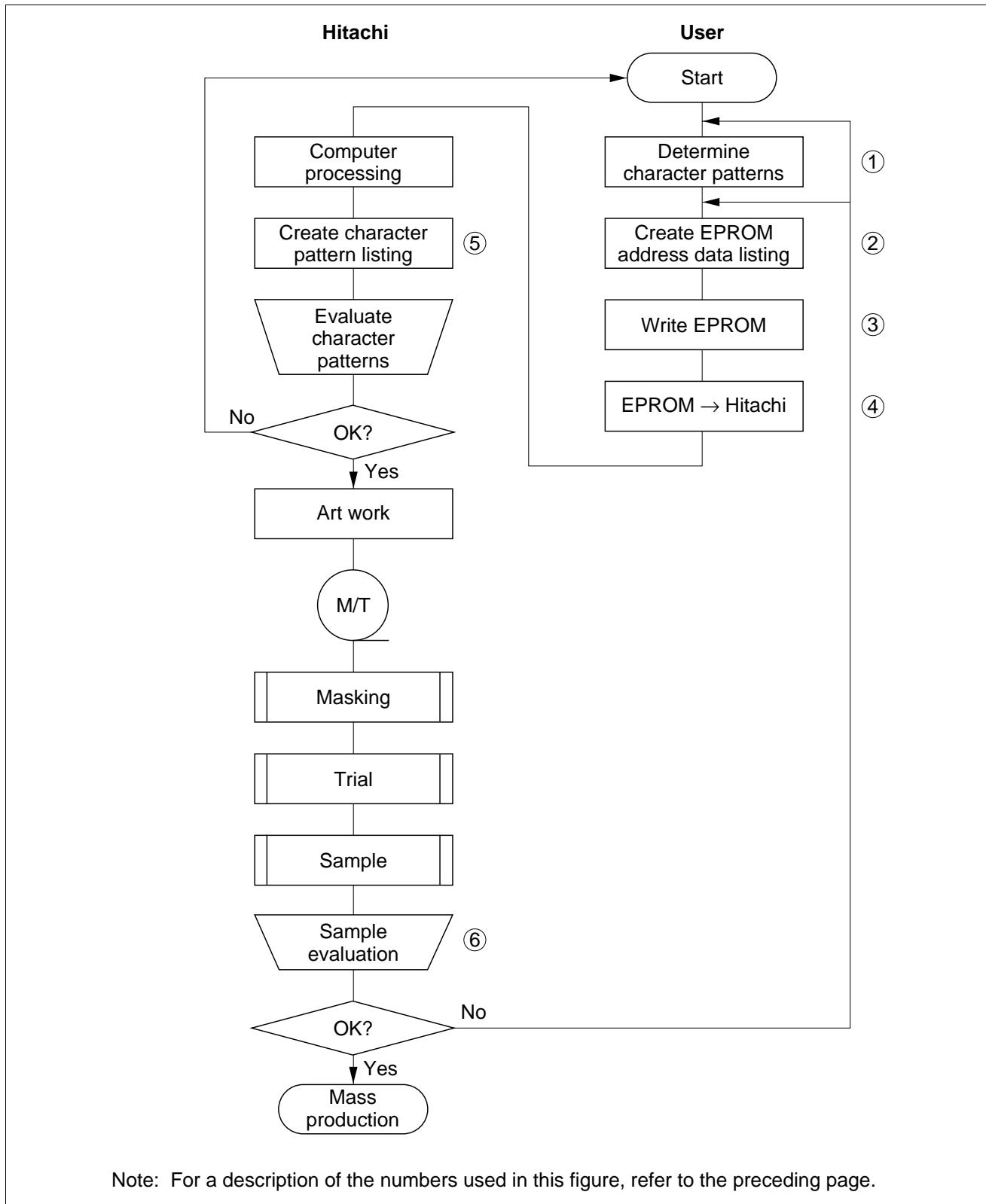


Figure 7 Character Pattern Development Procedure

- Programming character patterns

This section explains the correspondence between addresses and data used to program character patterns in EPROM. The HD44780U character generator ROM can generate 208 5×8 dot character patterns and 32 5×10 dot character patterns for a total of 240 different character patterns.

- Character patterns

EPROM address data and character pattern data correspond with each other to form a 5×8 or 5×10 dot character pattern (Tables 2 and 3).

Table 2 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 8 Dots)

EPROM Address										Data				
										LSB				
A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0										O4	O3	O2	O1	O0
										1	0	0	0	0
										1	0	0	0	0
										1	0	1	1	0
										1	1	0	0	1
										1	0	0	0	1
										1	0	0	0	1
										1	1	1	1	0
0	1	1	0	0	0	1	0	0	1	1	1	1	0	0
										0	0	0	0	0
										1	0	0	0	0
										1	0	0	0	0
										1	0	1	0	0
										1	0	1	1	0
										1	1	0	0	0
										1	1	0	1	0
										1	1	1	0	0
										1	1	1	1	1
										0	0	0	0	0

← Cursor position

Character code Line position

- Notes:
1. EPROM addresses A11 to A4 correspond to a character code.
 2. EPROM addresses A3 to A0 specify a line position of the character pattern.
 3. EPROM data O4 to O0 correspond to character pattern data.
 4. EPROM data O5 to O7 must be specified as 0.
 5. A lit display position (black) corresponds to a 1.
 6. Line 9 and the following lines must be blanked with 0s for a 5×8 dot character fonts.

— Handling unused character patterns

1. EPROM data outside the character pattern area: Always input 0s.
2. EPROM data in CGRAM area: Always input 0s. (Input 0s to EPROM addresses 00H to FFH.)
3. EPROM data used when the user does not use any HD44780U character pattern: According to the user application, handled in one of the two ways listed as follows.
 - a. When unused character patterns are not programmed: If an unused character code is written into DDRAM, all its dots are lit. By not programming a character pattern, all of its bits become lit. (This is due to the EPROM being filled with 1s after it is erased.)
 - b. When unused character patterns are programmed as 0s: Nothing is displayed even if unused character codes are written into DDRAM. (This is equivalent to a space.)

Table 3 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 10 Dots)

EPROM Address										Data						
										LSB						
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	O0
												0	0	0	0	0
												0	0	0	0	0
												0	1	1	0	1
												0	0	1	1	1
												1	0	0	1	1
												0	1	0	0	1
												1	0	0	0	1
												0	1	0	1	1
												0	1	1	1	1
0	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1
												0	1	1	1	1
												0	0	0	0	1
												1	0	0	0	1
												0	0	0	0	1
												1	0	1	0	0
												0	0	0	0	0
												1	0	1	1	1
												0	0	0	0	0
												1	1	0	0	0
												0	0	0	0	0
												1	1	0	1	0
												0	0	0	0	0
												1	1	1	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
												0	0	0	0	0
												1	1	1	1	1
			</td													

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Upper 4 Bits Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	0	P	P					—	9	E	x	p	
xxxx0001	(2)		!	1	A	Q	a	q			■	7	7	4	ä	q
xxxx0010	(3)		"	2	B	R	b	r			■	4	9	x	£	9
xxxx0011	(4)		#	3	C	S	c	s			■	0	T	E	€	8
xxxx0100	(5)		\$	4	D	T	d	t			■	I	ト	フ	μ	2
xxxx0101	(6)		%	5	E	U	e	u			■	オ	ナ	1	€	0
xxxx0110	(7)		&	6	F	V	f	v			■	カ	ニ	3	ρ	Σ
xxxx0111	(8)		*	7	G	W	g	w			■	†	‡	‡	g	π
xxxx1000	(1)		(8	H	X	h	x			■	○	＊	Y	×	X
xxxx1001	(2))	9	I	Y	i	y			■	†	J	l	~	y
xxxx1010	(3)		*	:	J	Z	j	z			■	□	△	▽	j	≠
xxxx1011	(4)		+	;	K	C	k	c			■	†	□	○	×	¤
xxxx1100	(5)		,	<	L	⌘	l	l			■	○	○	○	Φ	¤
xxxx1101	(6)		=	=	M]	m	}			■	○	○	○	±	±
xxxx1110	(7)		,	>	N	^	n	+			■	†	†	†	ñ	
xxxx1111	(8)		/	?	O	_	o	+			■	○	○	○	ö	█

Note: The user can specify any pattern for character-generator RAM.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A02)

Upper 4 Bits Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)															
xxxx0001	(2)		!	1	A	Q	a	q	A	J	i	t	A	N	A	N
xxxx0010	(3)		"	2	B	R	b	r	X	C	Q	2	A	O	á	ó
xxxx0011	(4)		#	3	C	S	c	s	3	X	E	3	Á	Ó	á	ó
xxxx0100	(5)		\$	4	D	T	d	t	H	Z	x	R	Ä	ö	ä	ö
xxxx0101	(6)		%	5	E	U	e	u	U	o	¥	P	Å	ß	ß	ß
xxxx0110	(7)		8	6	F	V	f	v	J	J	!	9	€	ö	æ	ö
xxxx0111	(8)		?	7	G	W	w	W	T	S	=	G	X	g	÷	
xxxx1000	(1)		(8	H	X	h	x	Y	*	+	o	É	£	é	£
xxxx1001	(2))	9	I	Y	i	y	U	8	1	É	0	é	ú	
xxxx1010	(3)		*	*	J	Z	j	z	4	9	9	É	0	é	ú	
xxxx1011	(4)		+	+	K	C	k	c	W	8	<	É	0	é	ó	
xxxx1100	(5)		,	,	L	\	l	\	W	~	0	§	í	0	í	ó
xxxx1101	(6)		-	-	M	M	m	m	2	b	*	§	i	í	í	í
xxxx1110	(7)		.	.	N	^	n	~	ы	с	2	и	í	í	í	í
xxxx1111	(8)		/	?	0	o	o	3	0	o	?	í	í	í	í	í

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

For 5 × 8 dot character patterns

Character Codes (DDRAM data)		CGRAM Address						Character Patterns (CGRAM data)	
7 6 5 4 3 2 1 0 High Low		5 4 3 2 1 0 High Low						7 6 5 4 3 2 1 0 High Low	
0 0 0 0 0 * 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0	Character pattern (1)
0 0 0 0 0 * 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	Character pattern (2)
0 0 0 0 0 * 1 1 1	1 1 1	1 1 1	1 0 0	1 0 1	1 1 0	1 1 1	1 1 1	* * *	Cursor position

- Notes:
1. Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
 2. CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor. Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display. If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.
 3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
 4. As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
 5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- * Indicates no effect.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data) (cont)

For 5 × 10 dot character patterns

Character Codes (DDRAM data)								CGRAM Address								Character Patterns (CGRAM data)							
7 6 5 4 3 2 1 0				5 4 3 2 1 0				7 6 5 4 3 2 1 0				High				Low							
High		Low		High		Low		High				High				High				Low			
*	*	*	*	*	*	*	*	0	0	0	0	*	*	*	*	0	0	0	0	0	0	0	0
0	0	0	0	*	0	0	*	0	0	0	1	*	*	*	*	0	0	0	0	0	0	0	0
0	0	0	0	*	0	0	*	0	0	1	0	*	*	*	*	1	0	1	1	0	0	0	0
0	0	0	0	*	0	0	*	0	1	0	0	*	*	*	*	1	1	0	0	1	1	0	0
0	0	0	0	*	0	0	*	0	1	1	0	*	*	*	*	1	0	0	0	1	1	1	0
0	0	0	0	*	0	0	*	0	1	1	1	*	*	*	*	1	1	0	0	0	0	0	0
0	0	0	0	*	0	0	*	1	0	0	0	*	*	*	*	1	0	0	0	0	0	0	0
0	0	0	0	*	0	0	*	1	0	0	1	*	*	*	*	1	0	1	1	0	0	0	0
0	0	0	0	*	0	0	*	1	0	1	0	*	*	*	*	1	1	0	0	1	1	1	0
0	0	0	0	*	0	0	*	1	1	0	1	*	*	*	*	1	1	1	0	0	0	0	0
0	0	0	0	*	0	0	*	1	1	1	0	*	*	*	*	1	1	1	1	1	0	0	0
0	0	0	0	*	0	0	*	1	1	1	1	*	*	*	*	1	0	0	0	0	0	0	0
0	0	0	0	*	0	0	*	0	0	0	0	*	*	*	*	0	0	0	0	0	0	0	0
0	0	0	0	*	0	0	*	0	0	0	1	*	*	*	*	0	0	0	0	0	0	0	0
0	0	0	0	*	1	1	*	1	1	1	0	*	*	*	*	1	0	1	1	0	0	0	0
0	0	0	0	*	1	1	*	1	0	1	0	*	*	*	*	1	1	0	0	1	1	1	0
0	0	0	0	*	1	1	*	1	1	0	1	*	*	*	*	1	1	1	0	0	0	0	0
0	0	0	0	*	1	1	*	1	1	1	0	*	*	*	*	1	1	1	1	0	0	0	0
0	0	0	0	*	1	1	*	1	1	1	1	*	*	*	*	1	1	1	1	1	0	0	0
0	0	0	0	*	1	1	*	1	1	1	1	*	*	*	*	1	1	1	1	1	1	0	0
0	0	0	0	*	1	1	*	1	1	1	1	*	*	*	*	1	1	1	1	1	1	1	0
0	0	0	0	*	1	1	*	1	1	1	1	*	*	*	*	1	1	1	1	1	1	1	1

- Notes:
1. Character code bits 1 and 2 correspond to CGRAM address bits 4 and 5 (2 bits: 4 types).
 2. CGRAM address bits 0 to 3 designate the character pattern line position. The 11th line is the cursor position and its display is formed by a logical OR with the cursor.
 - Maintain the 11th line data corresponding to the cursor display position at 0 as the cursor display. If the 11th line data is "1", "1" bits will light up the 11th line regardless of the cursor presence. Since lines 12 to 16 are not used for display, they can be used for general data RAM.
 3. Character pattern row positions are the same as 5 × 8 dot character pattern positions.
 4. CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bits 0 and 3 have no effect, the P display example above can be selected by character codes 00H, 01H, 08H, and 09H.
 5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- * Indicates no effect.

Timing Generation Circuit

The timing generation circuit generates timing signals for the operation of internal circuits such as DDRAM, CGROM and CGRAM. RAM read timing for display and internal operation timing by MPU access are generated separately to avoid interfering with each other. Therefore, when writing data to DDRAM, for example, there will be no undesirable interferences, such as flickering, in areas other than the display area.

Liquid Crystal Display Driver Circuit

The liquid crystal display driver circuit consists of 16 common signal drivers and 40 segment signal drivers. When the character font and number of lines are selected by a program, the required common signal drivers automatically output drive waveforms, while the other common signal drivers continue to output non-selection waveforms.

Sending serial data always starts at the display data character pattern corresponding to the last address of the display data RAM (DDRAM).

Since serial data is latched when the display data character pattern corresponding to the starting address enters the internal shift register, the HD44780U drives from the head display.

Cursor/Blink Control Circuit

The cursor/blink control circuit generates the cursor or character blinking. The cursor or the blinking will appear with the digit located at the display data RAM (DDRAM) address set in the address counter (AC).

For example (Figure 8), when the address counter is 08H, the cursor position is displayed at DDRAM address 08H.

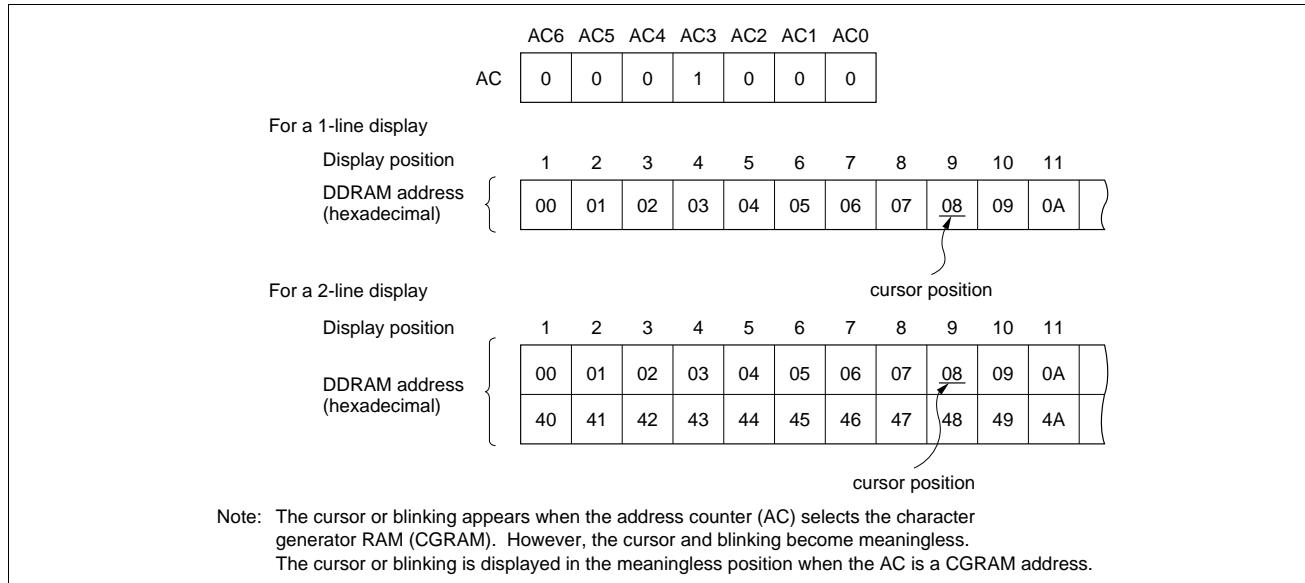


Figure 8 Cursor/Blink Display Example

Interfacing to the MPU

The HD44780U can send data in either two 4-bit operations or one 8-bit operation, thus allowing interfacing with 4- or 8-bit MPUs.

- For 4-bit interface data, only four bus lines (DB4 to DB7) are used for transfer. Bus lines DB0 to DB3 are disabled. The data transfer between the HD44780U and the MPU is completed after the 4-bit data has been transferred twice. As for the order of data transfer, the four high order bits (for 8-bit operation, DB4 to DB7) are transferred before the four low order bits (for 8-bit operation, DB0 to DB3).
The busy flag must be checked (one instruction) after the 4-bit data has been transferred twice. Two more 4-bit operations then transfer the busy flag and address counter data.
- For 8-bit interface data, all eight bus lines (DB0 to DB7) are used.

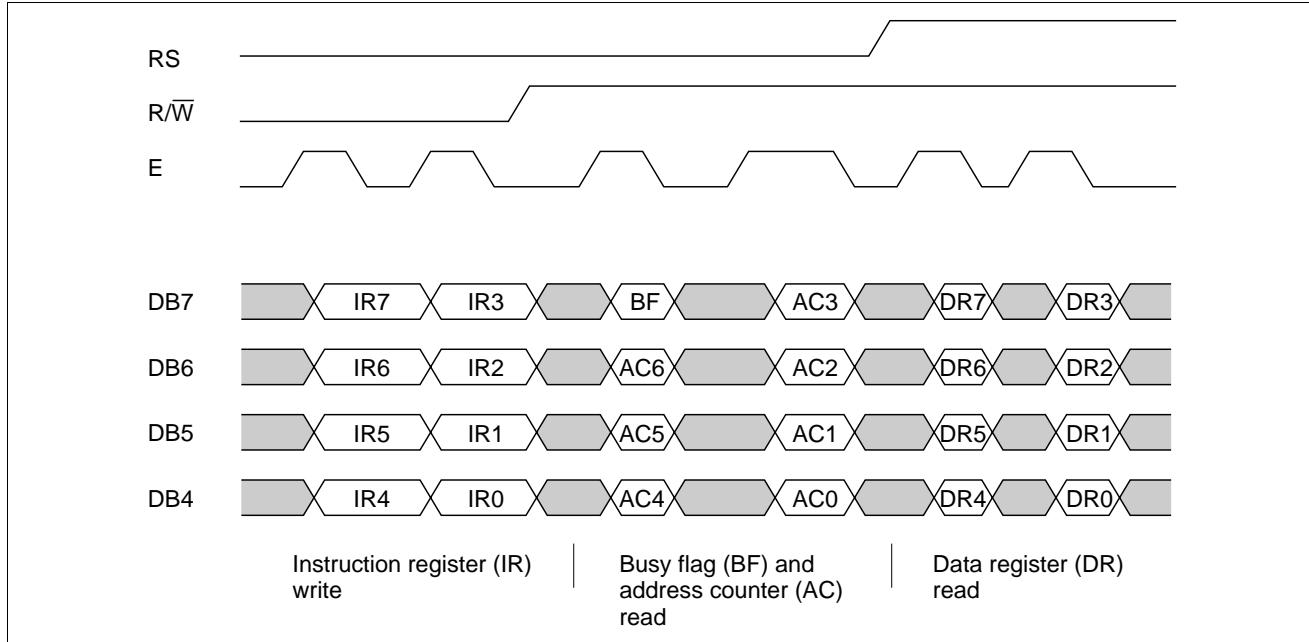


Figure 9 4-Bit Transfer Example

Reset Function

Initializing by Internal Reset Circuit

An internal reset circuit automatically initializes the HD44780U when the power is turned on. The following instructions are executed during the initialization. The busy flag (BF) is kept in the busy state until the initialization ends ($BF = 1$). The busy state lasts for 10 ms after V_{CC} rises to 4.5 V.

1. Display clear

2. Function set:

$DL = 1$; 8-bit interface data

$N = 0$; 1-line display

$F = 0$; 5×8 dot character font

3. Display on/off control:

$D = 0$; Display off

$C = 0$; Cursor off

$B = 0$; Blinking off

4. Entry mode set:

$I/D = 1$; Increment by 1

$S = 0$; No shift

Note: If the electrical characteristics conditions listed under the table Power Supply Conditions Using Internal Reset Circuit are not met, the internal reset circuit will not operate normally and will fail to initialize the HD44780U. For such a case, initialization must be performed by the MPU as explained in the section, Initializing by Instruction.

Instructions

Outline

Only the instruction register (IR) and the data register (DR) of the HD44780U can be controlled by the MPU. Before starting the internal operation of the HD44780U, control information is temporarily stored into these registers to allow interfacing with various MPUs, which operate at different speeds, or various peripheral control devices. The internal operation of the HD44780U is determined by signals sent from the MPU. These signals, which include register selection signal (RS), read/write signal (R/\overline{W}), and the data bus (DB0 to DB7), make up the HD44780U instructions (Table 6). There are four categories of instructions that:

- Designate HD44780U functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions

Normally, instructions that perform data transfer with internal RAM are used the most. However, auto-incrementation by 1 (or auto-decrementation by 1) of internal HD44780U RAM addresses after each data write can lighten the program load of the MPU. Since the display shift instruction (Table 11) can perform concurrently with display data write, the user can minimize system development time with maximum programming efficiency.

When an instruction is being executed for internal operation, no instruction other than the busy flag/address read instruction can be executed.

Because the busy flag is set to 1 while an instruction is being executed, check it to make sure it is 0 before sending another instruction from the MPU.

Note: Be sure the HD44780U is not in the busy state ($BF = 0$) before sending an instruction from the MPU to the HD44780U. If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Table 6 for the list of each instruction execution time.

Table 6 Instructions

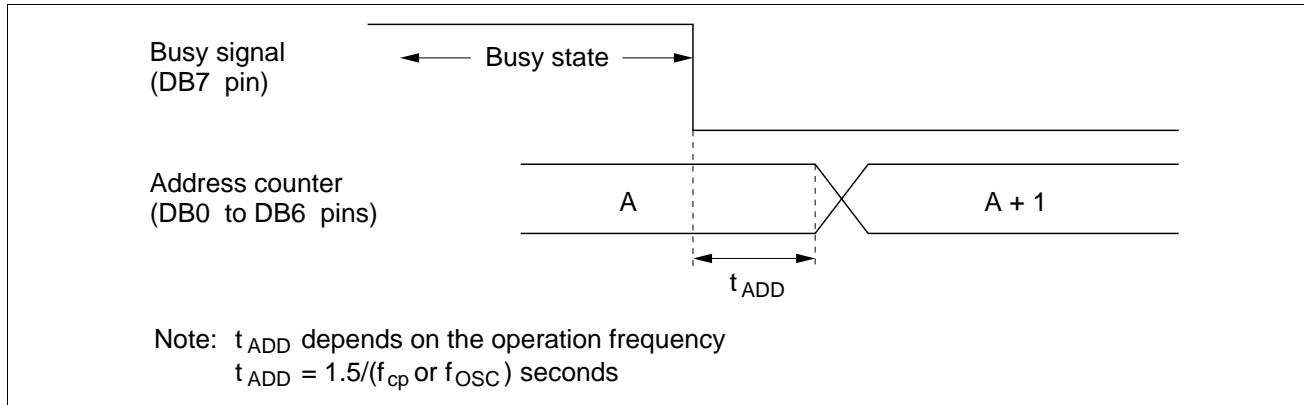
Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 μ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 μ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 μ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 μ s
Set DDRAM address	0	0	1	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 μ s						
Read busy flag & address	0	1	BF	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μ s						

Table 6 Instructions (cont)

Instruction	Code										Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)				
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0					
Write data to CG or DDRAM	1	0	Write data										Writes data into DDRAM or CGRAM. $t_{ADD} = 4 \mu s^*$	37 μs $t_{ADD} = 4 \mu s^*$	
Read data from CG or DDRAM	1	1	Read data										Reads data from DDRAM or CGRAM.	37 μs $t_{ADD} = 4 \mu s^*$	
	I/D	= 1:	Increment									DDRAM: Display data RAM	Execution time		
	I/D	= 0:	Decrement									CGRAM: Character generator	changes when		
	S	= 1:	Accompanies display shift									RAM	frequency changes		
	S/C	= 1:	Display shift									ACG: CGRAM address	Example:		
	S/C	= 0:	Cursor move									ADD: DDRAM address	When f_{cp} or f_{osc} is		
	R/L	= 1:	Shift to the right									(corresponds to cursor	250 kHz,		
	R/L	= 0:	Shift to the left									address)	$37 \mu s \times \frac{270}{250} = 40 \mu s$		
	DL	= 1:	8 bits, DL = 0: 4 bits									AC: Address counter used for	both DD and CGRAM		
	N	= 1:	2 lines, N = 0: 1 line									addresses			
	F	= 1:	5 × 10 dots, F = 0: 5 × 8 dots												
	BF	= 1:	Internally operating												
	BF	= 0:	Instructions acceptable												

Note: — indicates no effect.

- * After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.

**Figure 10 Address Counter Update**

Instruction Description

Clear Display

Clear display writes space code 20H (character pattern for character code 20H must be a blank pattern) into all DDRAM addresses. It then sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. In other words, the display disappears and the cursor or blinking goes to the left edge of the display (in the first line if 2 lines are displayed). It also sets I/D to 1 (increment mode) in entry mode. S of entry mode does not change.

Return Home

Return home sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. The DDRAM contents do not change.

The cursor or blinking go to the left edge of the display (in the first line if 2 lines are displayed).

Entry Mode Set

I/D: Increments (I/D = 1) or decrements (I/D = 0) the DDRAM address by 1 when a character code is written into or read from DDRAM.

The cursor or blinking moves to the right when incremented by 1 and to the left when decremented by 1. The same applies to writing and reading of CGRAM.

S: Shifts the entire display either to the right (I/D = 0) or to the left (I/D = 1) when S is 1. The display does not shift if S is 0.

If S is 1, it will seem as if the cursor does not move but the display does. The display does not shift when reading from DDRAM. Also, writing into or reading out from CGRAM does not shift the display.

Display On/Off Control

D: The display is on when D is 1 and off when D is 0. When off, the display data remains in DDRAM, but can be displayed instantly by setting D to 1.

C: The cursor is displayed when C is 1 and not displayed when C is 0. Even if the cursor disappears, the function of I/D or other specifications will not change during display data write. The cursor is displayed using 5 dots in the 8th line for 5×8 dot character font selection and in the 11th line for the 5×10 dot character font selection (Figure 13).

B: The character indicated by the cursor blinks when B is 1 (Figure 13). The blinking is displayed as switching between all blank dots and displayed characters at a speed of 409.6-ms intervals when f_{cp} or f_{osc} is 250 kHz. The cursor and blinking can be set to display simultaneously. (The blinking frequency changes according to f_{osc} or the reciprocal of f_{cp} . For example, when f_{cp} is 270 kHz, $409.6 \times 250/270 = 379.2$ ms.)

Cursor or Display Shift

Cursor or display shift shifts the cursor position or display to the right or left without writing or reading display data (Table 7). This function is used to correct or search the display. In a 2-line display, the cursor moves to the second line when it passes the 40th digit of the first line. Note that the first and second line displays will shift at the same time.

When the displayed data is shifted repeatedly each line moves only horizontally. The second line display does not shift into the first line position.

The address counter (AC) contents will not change if the only action performed is a display shift.

Function Set

DL: Sets the interface data length. Data is sent or received in 8-bit lengths (DB7 to DB0) when DL is 1, and in 4-bit lengths (DB7 to DB4) when DL is 0. When 4-bit length is selected, data must be sent or received twice.

N: Sets the number of display lines.

F: Sets the character font.

Note: Perform the function at the head of the program before executing any instructions (except for the read busy flag and address instruction). From this point, the function set instruction cannot be executed unless the interface data length is changed.

Set CGRAM Address

Set CGRAM address sets the CGRAM address binary AAAAAA into the address counter.

Data is then written to or read from the MPU for CGRAM.

		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear display	Code	0	0	0	0	0	0	0	0	1	
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Return home	Code	0	0	0	0	0	0	0	0	1	*
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Entry mode set	Code	0	0	0	0	0	0	0	1	I/D	S
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Display on/off control	Code	0	0	0	0	0	0	1	D	C	B
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Cursor or display shift	Code	0	0	0	0	0	1	S/C	R/L	*	*
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Function set	Code	0	0	0	0	1	DL	N	F	*	*
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Set CGRAM address	Code	0	0	0	1	A	A	A	A	A	A
		← Higher order bit				Lower order bit →					

Figure 11 Instruction (1)

Set DDRAM Address

Set DDRAM address sets the DDRAM address binary AAAAAAAA into the address counter.

Data is then written to or read from the MPU for DDRAM.

However, when N is 0 (1-line display), AAAAAAAA can be 00H to 4FH. When N is 1 (2-line display), AAAAAAAA can be 00H to 27H for the first line, and 40H to 67H for the second line.

Read Busy Flag and Address

Read busy flag and address reads the busy flag (BF) indicating that the system is now internally operating on a previously received instruction. If BF is 1, the internal operation is in progress. The next instruction will not be accepted until BF is reset to 0. Check the BF status before the next write operation. At the same time, the value of the address counter in binary AAAAAAAA is read out. This address counter is used by both CG and DDRAM addresses, and its value is determined by the previous instruction. The address contents are the same as for instructions set CGRAM address and set DDRAM address.

Table 7 Shift Function

S/C	R/L	
0	0	Shifts the cursor position to the left. (AC is decremented by one.)
0	1	Shifts the cursor position to the right. (AC is incremented by one.)
1	0	Shifts the entire display to the left. The cursor follows the display shift.
1	1	Shifts the entire display to the right. The cursor follows the display shift.

Table 8 Function Set

N	F	No. of Display Lines	Character Font	Duty Factor	Remarks
0	0	1	5 × 8 dots	1/8	
0	1	1	5 × 10 dots	1/11	
1	*	2	5 × 8 dots	1/16	Cannot display two lines for 5 × 10 dot character font

Note: * Indicates don't care.

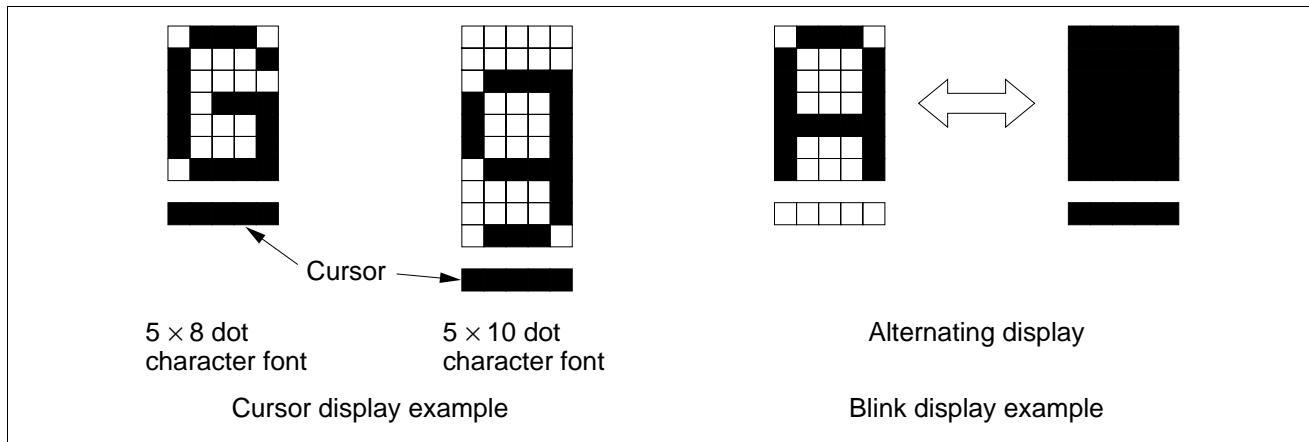


Figure 12 Cursor and Blinking

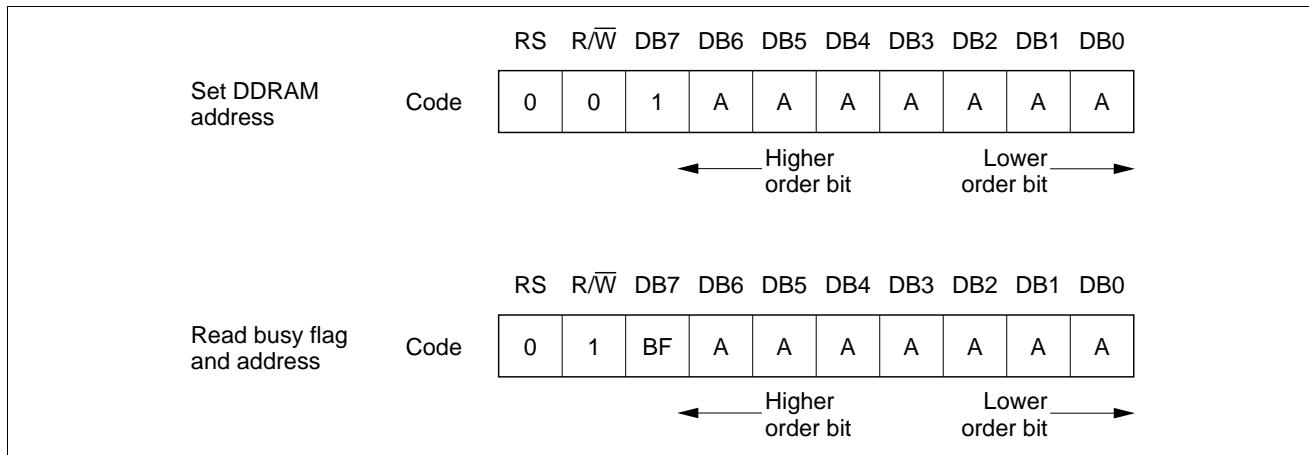


Figure 13 Instruction (2)

Write Data to CG or DDRAM

Write data to CG or DDRAM writes 8-bit binary data DDDDDDDDD to CG or DDRAM.

To write into CG or DDRAM is determined by the previous specification of the CGRAM or DDRAM address setting. After a write, the address is automatically incremented or decremented by 1 according to the entry mode. The entry mode also determines the display shift.

Read Data from CG or DDRAM

Read data from CG or DDRAM reads 8-bit binary data DDDDDDDDD from CG or DDRAM.

The previous designation determines whether CG or DDRAM is to be read. Before entering this read instruction, either CGRAM or DDRAM address set instruction must be executed. If not executed, the first read data will be invalid. When serially executing read instructions, the next address data is normally read from the second read. The address set instructions need not be executed just before this read instruction when shifting the cursor by the cursor shift instruction (when reading out DDRAM). The operation of the cursor shift instruction is the same as the set DDRAM address instruction.

After a read, the entry mode automatically increases or decreases the address by 1. However, display shift is not executed regardless of the entry mode.

Note: The address counter (AC) is automatically incremented or decremented by 1 after the write instructions to CGRAM or DDRAM are executed. The RAM data selected by the AC cannot be read out at this time even if read instructions are executed. Therefore, to correctly read data, execute either the address set instruction or cursor shift instruction (only with DDRAM), then just before reading the desired data, execute the read instruction from the second time the read instruction is sent.

		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Write data to CG or DDRAM	Code	1	0	D	D	D	D	D	D	D	D
		← Higher order bits					Lower order bits →				
Read data from CG or DDRAM	Code	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
		1	1	D	D	D	D	D	D	D	D
		← Higher order bits					Lower order bits →				

Figure 14 Instruction (3)

Interfacing the HD44780U

Interface to MPUs

- Interfacing to an 8-bit MPU

See Figure 16 for an example of using a I/O port (for a single-chip microcomputer) as an interface device.

In this example, P30 to P37 are connected to the data bus DB0 to DB7, and P75 to P77 are connected to E, R/W, and RS, respectively.

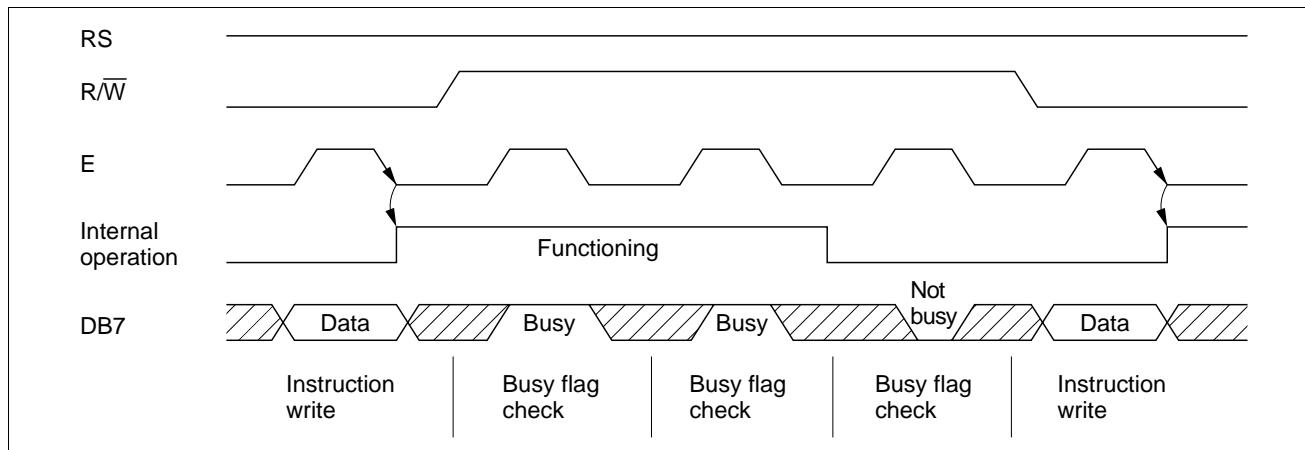


Figure 15 Example of Busy Flag Check Timing Sequence

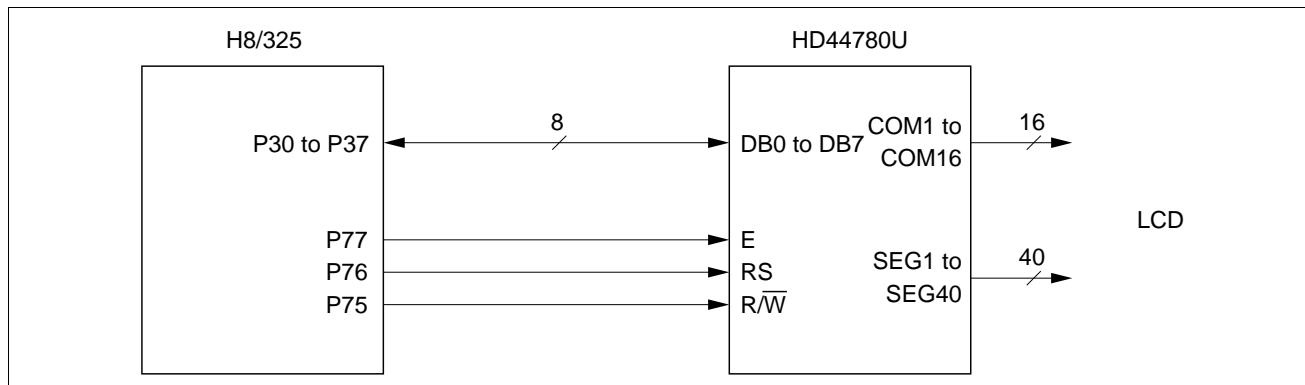


Figure 16 H8/325 Interface (Single-Chip Mode)

- Interfacing to a 4-bit MPU

The HD44780U can be connected to the I/O port of a 4-bit MPU. If the I/O port has enough bits, 8-bit data can be transferred. Otherwise, one data transfer must be made in two operations for 4-bit data. In this case, the timing sequence becomes somewhat complex. (See Figure 17.)

See Figure 18 for an interface example to the HMCS4019R.

Note that two cycles are needed for the busy flag check as well as for the data transfer. The 4-bit operation is selected by the program.

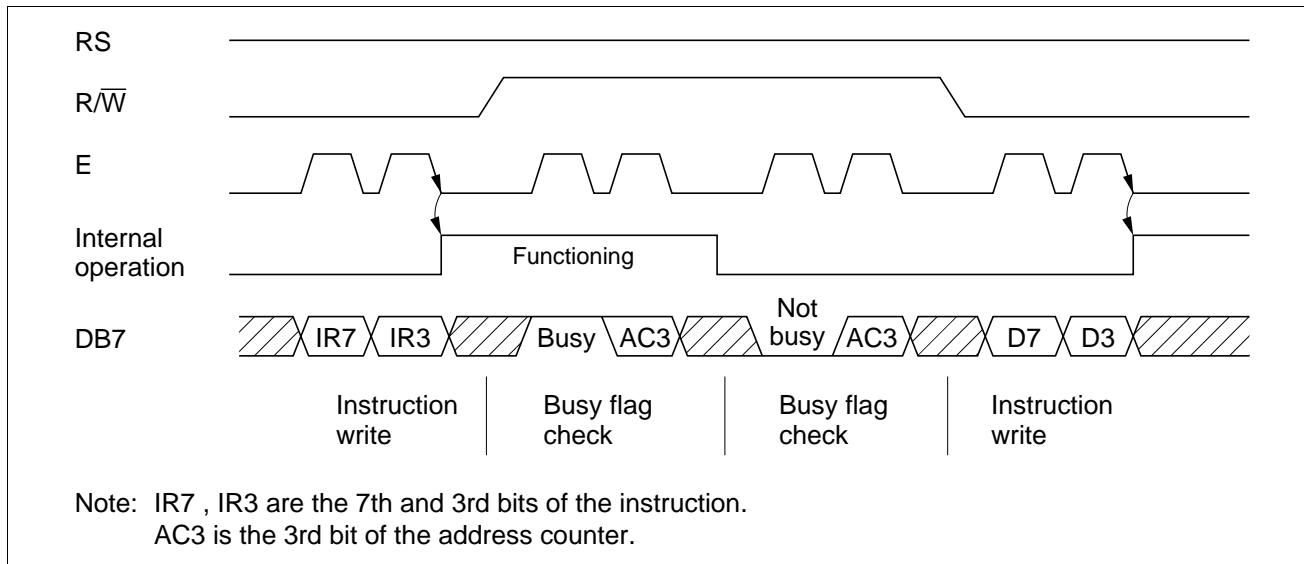


Figure 17 Example of 4-Bit Data Transfer Timing Sequence

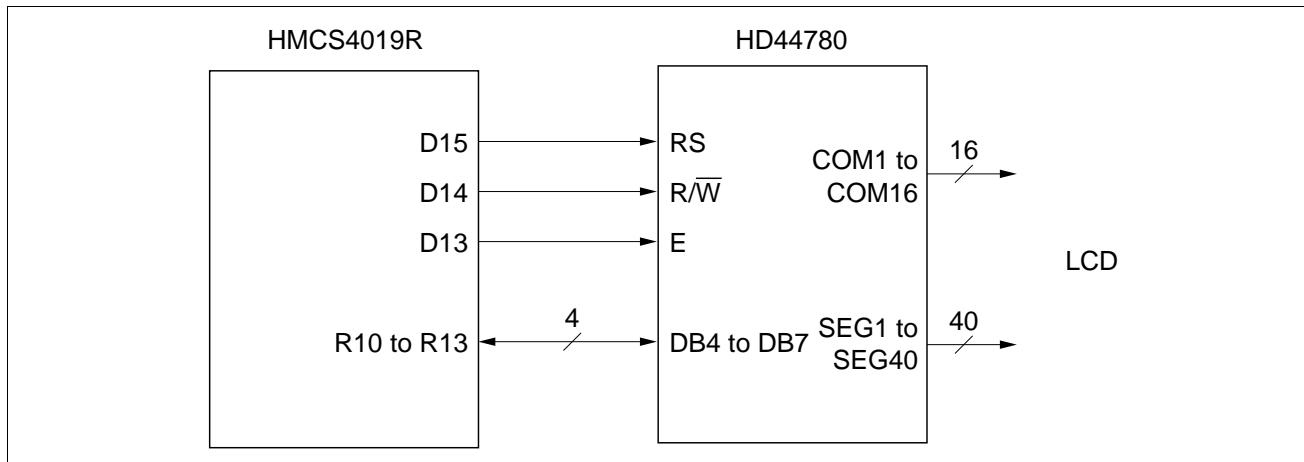


Figure 18 Example of Interface to HMCS4019R

Interface to Liquid Crystal Display

Character Font and Number of Lines: The HD44780U can perform two types of displays, 5×8 dot and 5×10 dot character fonts, each with a cursor.

Up to two lines are displayed for 5×8 dots and one line for 5×10 dots. Therefore, a total of three types of common signals are available (Table 9).

The number of lines and font types can be selected by the program. (See Table 6, Instructions.)

Connection to HD44780 and Liquid Crystal Display: See Figure 19 for the connection examples.

Table 9 Common Signals

Number of Lines	Character Font	Number of Common Signals	Duty Factor
1	5×8 dots + cursor	8	1/8
1	5×10 dots + cursor	11	1/11
2	5×8 dots + cursor	16	1/16

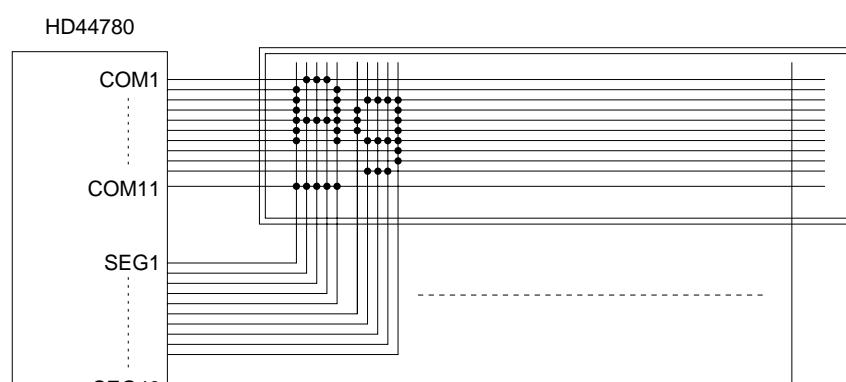
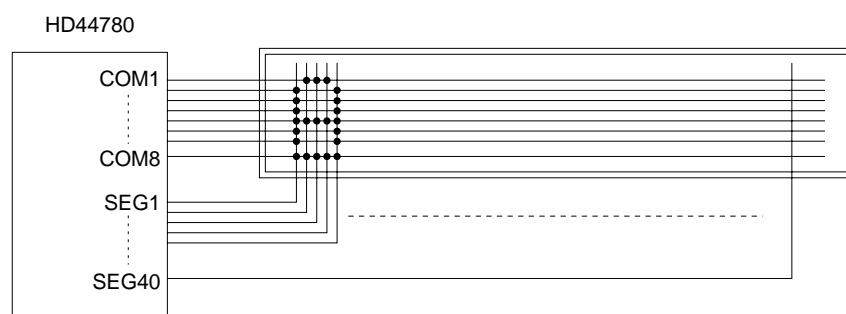
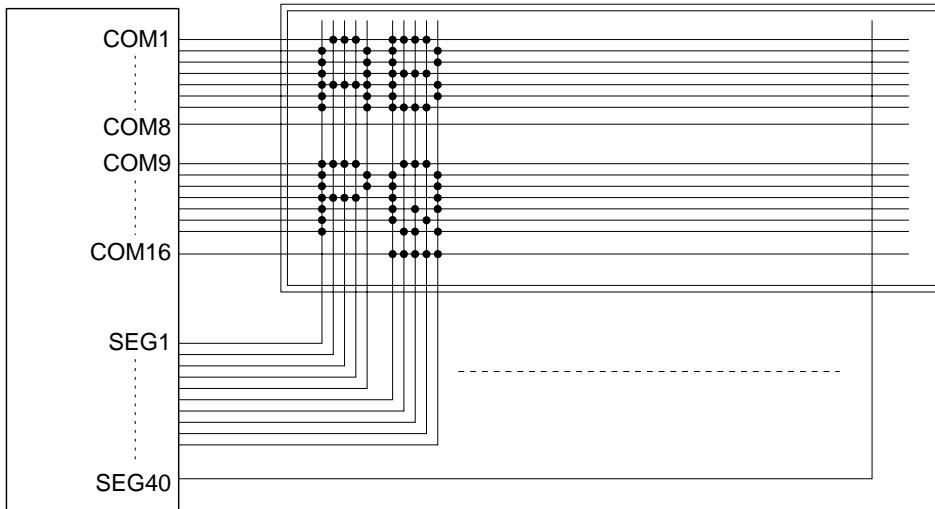


Figure 19 Liquid Crystal Display and HD44780 Connections

Since five segment signal lines can display one digit, one HD44780U can display up to 8 digits for a 1-line display and 16 digits for a 2-line display.

The examples in Figure 19 have unused common signal pins, which always output non-selection waveforms. When the liquid crystal display panel has unused extra scanning lines, connect the extra scanning lines to these common signal pins to avoid any undesirable effects due to crosstalk during the floating state.

HD44780



Example of a 5×8 dot, 8-character \times 2-line display (1/5 bias, 1/16 duty cycle)

Figure 19 Liquid Crystal Display and HD44780 Connections (cont)

Connection of Changed Matrix Layout: In the preceding examples, the number of lines correspond to the scanning lines. However, the following display examples (Figure 20) are made possible by altering the matrix layout of the liquid crystal display panel. In either case, the only change is the layout. The display characteristics and the number of liquid crystal display characters depend on the number of common signals or on duty factor. Note that the display data RAM (DDRAM) addresses for 4 characters \times 2 lines and for 16 characters \times 1 line are the same as in Figure 19.

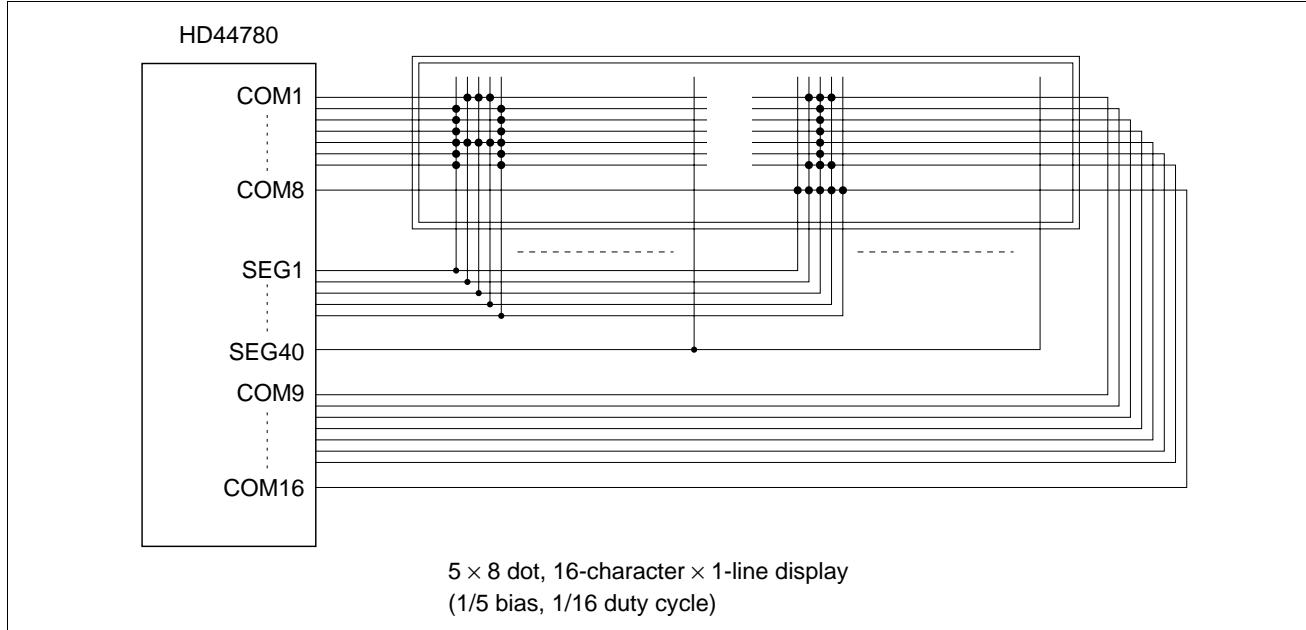


Figure 20 Changed Matrix Layout Displays

Power Supply for Liquid Crystal Display Drive

Various voltage levels must be applied to pins V1 to V5 of the HD44780U to obtain the liquid crystal display drive waveforms. The voltages must be changed according to the duty factor (Table 10).

VLCD is the peak value for the liquid crystal display drive waveforms, and resistance dividing provides voltages V1 to V5 (Figure 21).

Table 10 Duty Factor and Power Supply for Liquid Crystal Display Drive

Power Supply	Duty Factor	
	1/8, 1/11	1/16
	Bias	
V1	$V_{CC} - 1/4 VLCD$	$V_{CC} - 1/5 VLCD$
V2	$V_{CC} - 1/2 VLCD$	$V_{CC} - 2/5 VLCD$
V3	$V_{CC} - 1/2 VLCD$	$V_{CC} - 3/5 VLCD$
V4	$V_{CC} - 3/4 VLCD$	$V_{CC} - 4/5 VLCD$
V5	$V_{CC} - VLCD$	$V_{CC} - VLCD$

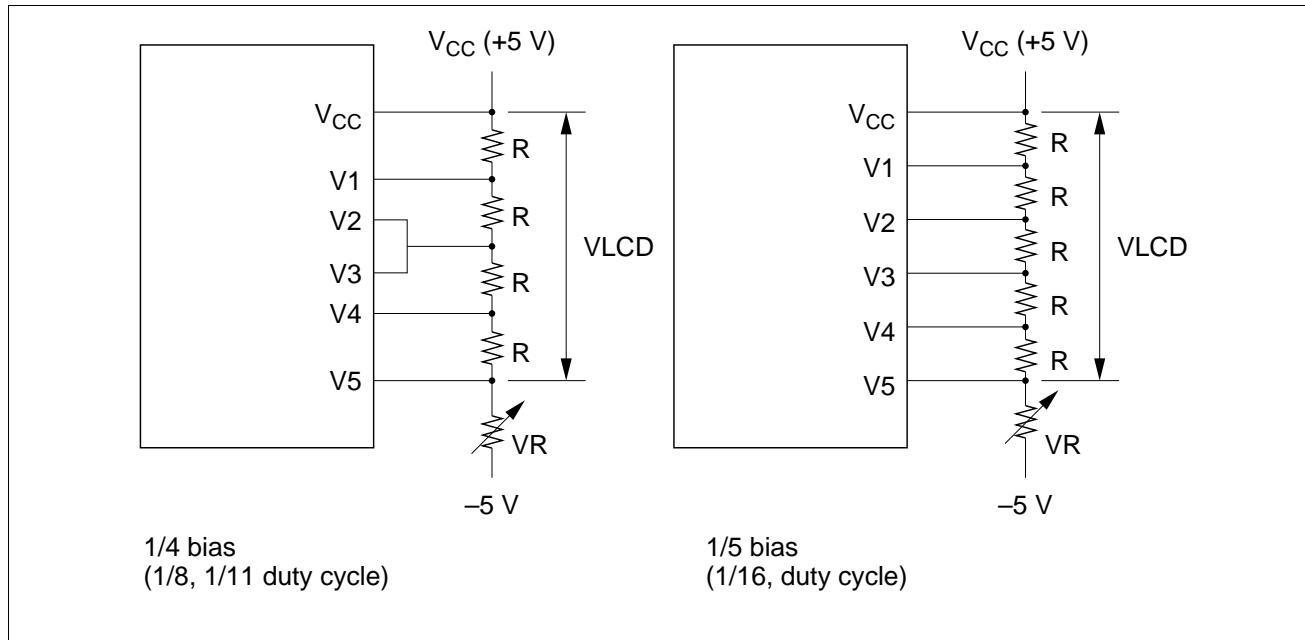


Figure 21 Drive Voltage Supply Example

Relationship between Oscillation Frequency and Liquid Crystal Display Frame Frequency

The liquid crystal display frame frequencies of Figure 22 apply only when the oscillation frequency is 270 kHz (one clock pulse of 3.7 µs).

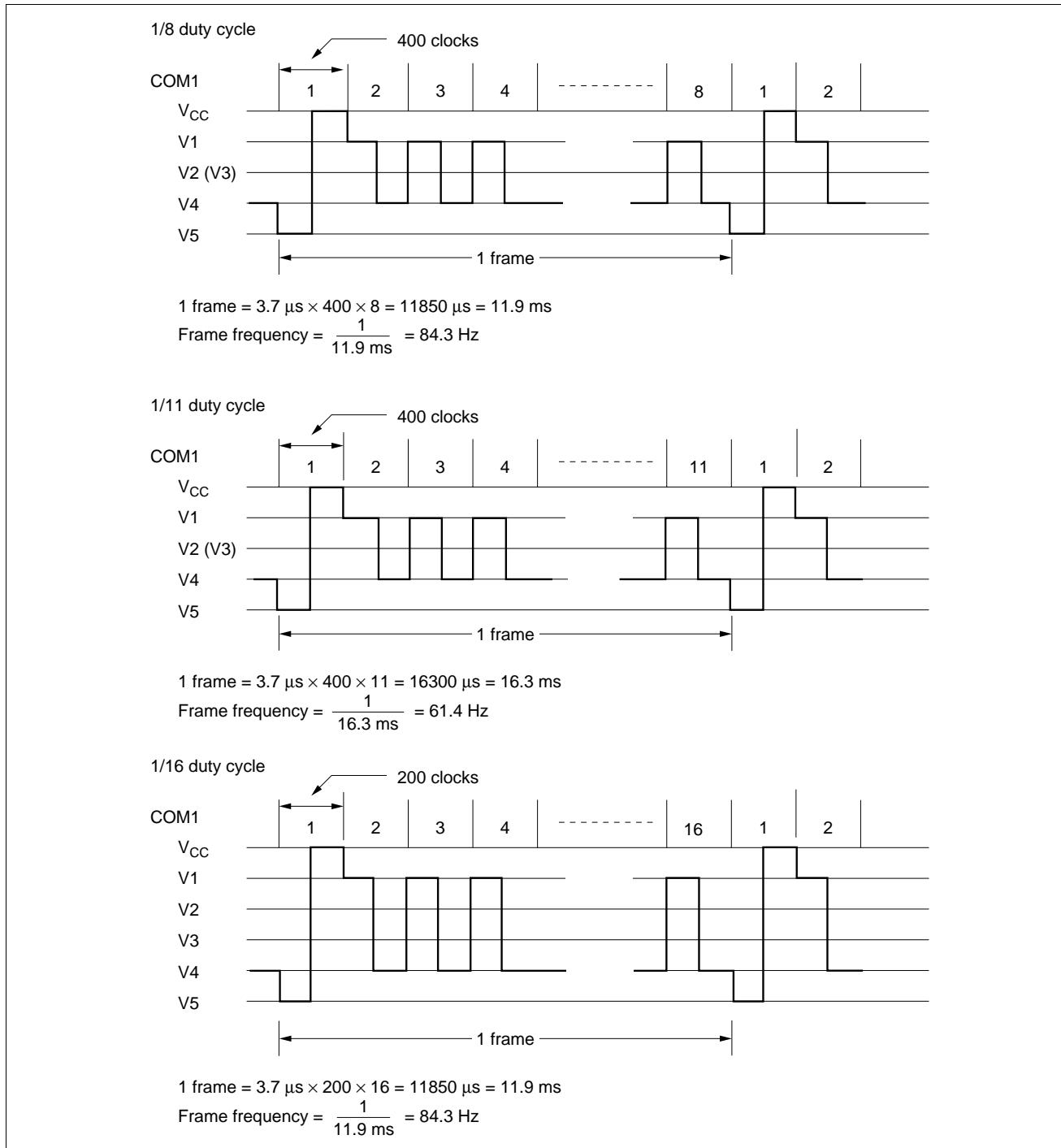


Figure 22 Frame Frequency

Instruction and Display Correspondence

- 8-bit operation, 8-digit × 1-line display with internal reset

Refer to Table 11 for an example of an 8-digit × 1-line display in 8-bit operation. The HD44780U functions must be set by the function set instruction prior to the display. Since the display data RAM can store data for 80 characters, as explained before, the RAM can be used for displays such as for advertising when combined with the display shift operation.

Since the display shift operation changes only the display position with DDRAM contents unchanged, the first display data entered into DDRAM can be output when the return home operation is performed.

- 4-bit operation, 8-digit × 1-line display with internal reset

The program must set all functions prior to the 4-bit operation (Table 12). When the power is turned on, 8-bit operation is automatically selected and the first write is performed as an 8-bit operation. Since DB0 to DB3 are not connected, a rewrite is then required. However, since one operation is completed in two accesses for 4-bit operation, a rewrite is needed to set the functions (see Table 12). Thus, DB4 to DB7 of the function set instruction is written twice.

- 8-bit operation, 8-digit × 2-line display

For a 2-line display, the cursor automatically moves from the first to the second line after the 40th digit of the first line has been written. Thus, if there are only 8 characters in the first line, the DDRAM address must be again set after the 8th character is completed. (See Table 13.) Note that the display shift operation is performed for the first and second lines. In the example of Table 13, the display shift is performed when the cursor is on the second line. However, if the shift operation is performed when the cursor is on the first line, both the first and second lines move together. If the shift is repeated, the display of the second line will not move to the first line. The same display will only shift within its own line for the number of times the shift is repeated.

Note: When using the internal reset, the electrical characteristics in the Power Supply Conditions Using Internal Reset Circuit table must be satisfied. If not, the HD44780U must be initialized by instructions. See the section, Initializing by Instruction.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step	Instruction											Operation
No.	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set											Sets to 8-bit operation and selects 1-line display and 5 × 8 dot character font. (Number of display lines and character fonts cannot be changed after step #2.)
3	Display on/off control											Turns on display and cursor. Entire display is in space mode because of initialization.
4	Entry mode set											Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM											Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6	Write data to CGRAM/DDRAM											Writes I.
7	.											.
8	Write data to CGRAM/DDRAM											Writes I.
9	Entry mode set											Sets mode to shift display at the time of write.
10	Write data to CGRAM/DDRAM											Writes a space.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset (cont)

Step No.	Instruction											Operation
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	
11	Write data to CGRAM/DDRAM											Writes M.
	1	0	0	1	0	0	1	1	0	1	TACHI M_	
12												
		
13	Write data to CGRAM/DDRAM											Writes O.
	1	0	0	1	0	0	1	1	1	1	MICROKO_	
14	Cursor or display shift											Shifts only the cursor position to the left.
	0	0	0	0	0	1	0	0	*	*	MICROKO	
15	Cursor or display shift											Shifts only the cursor position to the left.
	0	0	0	0	0	1	0	0	*	*	MICROKO	
16	Write data to CGRAM/DDRAM											Writes C over K. The display moves to the left.
	1	0	0	1	0	0	0	0	1	1	ICROCO	
17	Cursor or display shift											Shifts the display and cursor position to the right.
	0	0	0	0	0	1	1	1	*	*	MICROCO	
18	Cursor or display shift											Shifts the display and cursor position to the right.
	0	0	0	0	0	1	0	1	*	*	MICROCO_	
19	Write data to CGRAM/DDRAM											Writes M.
	1	0	0	1	0	0	1	1	0	1	ICROCOM_	
20												
		
21	Return home											Returns both display and cursor to the original position (address 0).
	0	0	0	0	0	0	0	0	1	0	HITACHI	

Table 12 4-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step No.	Instruction						Display	Operation
	RS	R/W	DB7	DB6	DB5	DB4		
1	Power supply on (the HD44780U is initialized by the internal reset circuit)						[]	Initialized. No display.
2	Function set 0 0 0 0 1 0						[]	Sets to 4-bit operation. In this case, operation is handled as 8 bits by initialization, and only this instruction completes with one write.
3	Function set 0 0 0 0 1 0 0 0 0 0 * *						[]	Sets 4-bit operation and selects 1-line display and 5 × 8 dot character font. 4-bit operation starts from this step and resetting is necessary. (Number of display lines and character fonts cannot be changed after step #3.)
4	Display on/off control 0 0 0 0 0 0 0 0 1 1 1 0						[]	Turns on display and cursor. Entire display is in space mode because of initialization.
5	Entry mode set 0 0 0 0 0 0 0 0 0 1 1 0						[]	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
6	Write data to CGRAM/DDRAM 1 0 0 1 0 0 1 0 1 0 0 0						[H]	Writes H. The cursor is incremented by one and shifts to the right.

Note: The control is the same as for 8-bit operation beyond step #6.

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset

Step No.	Instruction											Operation
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set											Sets to 8-bit operation and selects 2-line display and 5 × 8 dot character font.
3	Display on/off control											Turns on display and cursor. All display is in space mode because of initialization.
4	Entry mode set											Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM											Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6	.											.
7	.											.
7	Write data to CGRAM/DDRAM											Writes I.
7	1	0	0	1	0	0	1	0	0	1	HITACHI	
8	Set DDRAM address											Sets DDRAM address so that the cursor is positioned at the head of the second line.
8	0	0	1	1	0	0	0	0	0	0	HITACHI	
8											—	

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset (cont)

Step No.	Instruction											Operation
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	
9	Write data to CGRAM/DDRAM										HITACHI M_	Writes M.
10			.								.	
			.								.	
			.								.	
			.								.	
11	Write data to CGRAM/DDRAM										HITACHI MICROCO_	Writes O.
12	Entry mode set										HITACHI MICROCO_	Sets mode to shift display at the time of write.
13	Write data to CGRAM/DDRAM										HITACHI ICROCOM_	Writes M. Display is shifted to the left. The first and second lines both shift at the same time.
14			.								.	
			.								.	
			.								.	
			.								.	
15	Return home										HITACHI MICROCOM	Returns both display and cursor to the original position (address 0).

Initializing by Instruction

If the power supply conditions for correctly operating the internal reset circuit are not met, initialization by instructions becomes necessary.

Refer to Figures 23 and 24 for the procedures on 8-bit and 4-bit initializations, respectively.

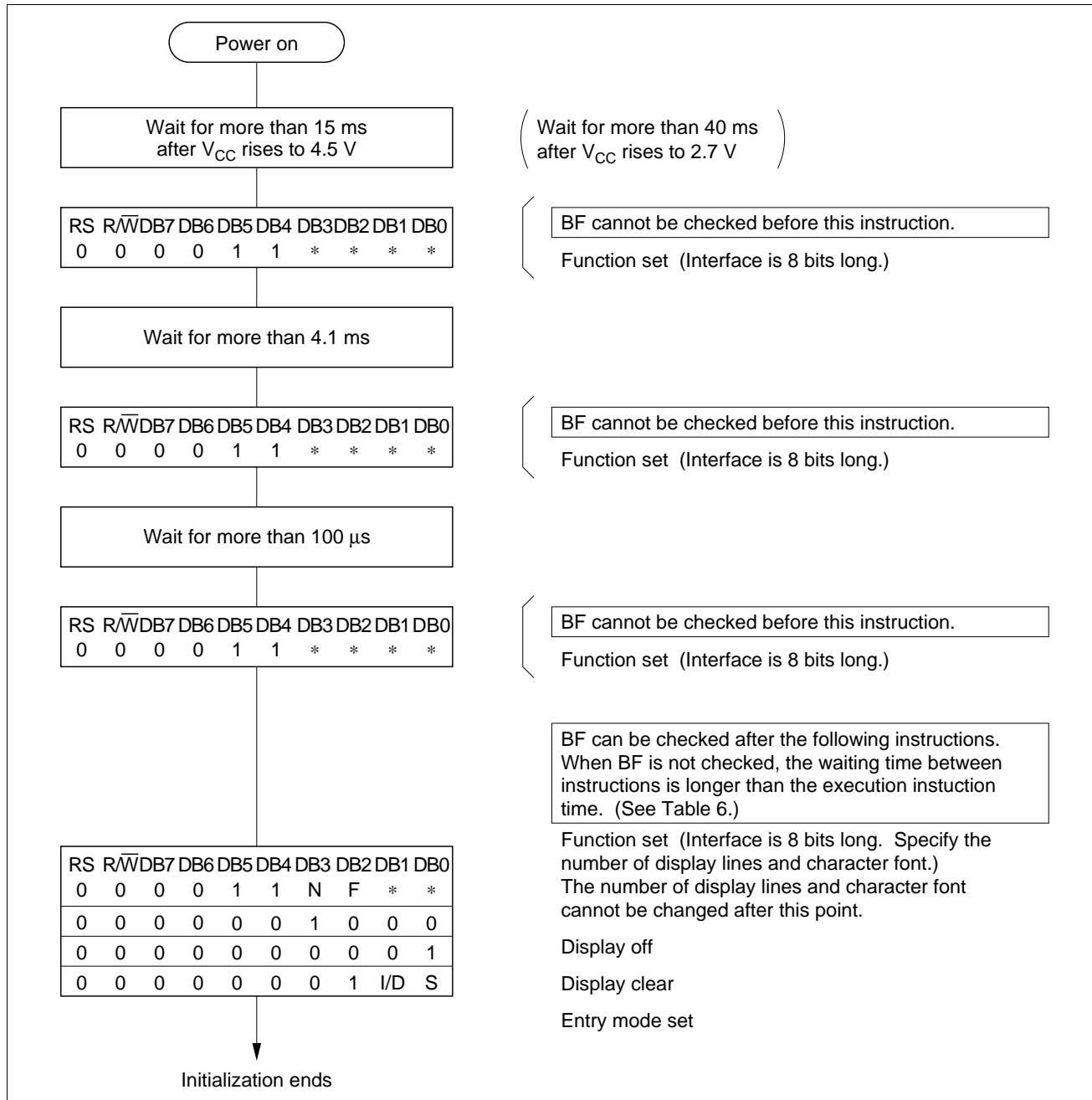


Figure 23 8-Bit Interface

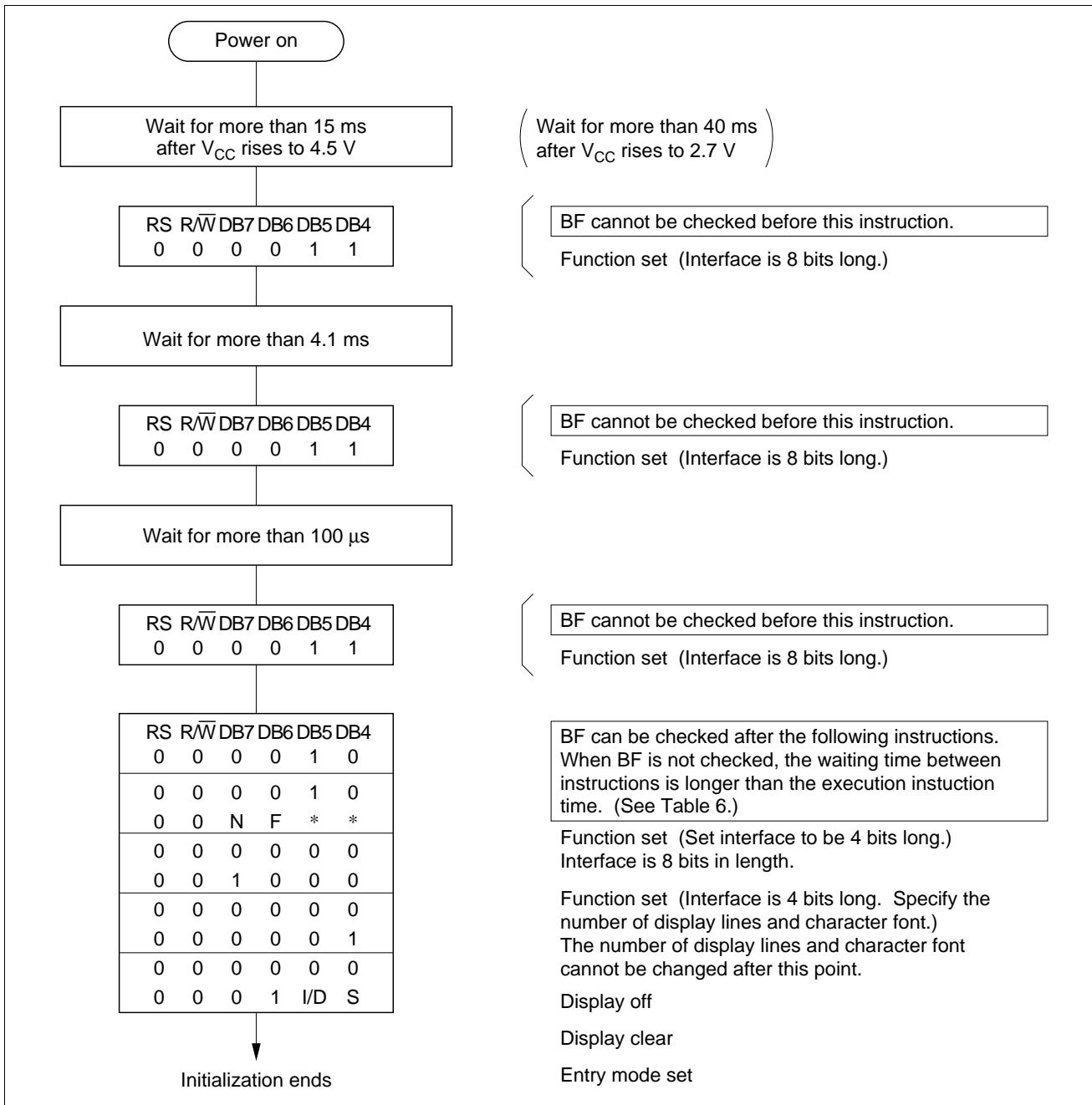


Figure 24 4-Bit Interface

Absolute Maximum Ratings*

Item	Symbol	Value	Unit	Notes
Power supply voltage (1)	V_{CC} -GND	-0.3 to +7.0	V	1
Power supply voltage (2)	V_{CC} -V5	-0.3 to +13.0	V	1, 2
Input voltage	V_t	-0.3 to V_{CC} +0.3	V	1
Operating temperature	T_{opr}	-30 to +75	°C	
Storage temperature	T_{stg}	-55 to +125	°C	4

Note: * If the LSI is used above these absolute maximum ratings, it may become permanently damaged. Using the LSI within the following electrical characteristic limits is strongly recommended for normal operation. If these electrical characteristic conditions are also exceeded, the LSI will malfunction and cause poor reliability.

DC Characteristics ($V_{CC} = 2.7$ to 4.5 V, $T_a = -30$ to $+75^\circ\text{C}$ *³)

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	$0.7V_{CC}$	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	-0.3	—	0.55	V		6
Input high voltage (2) (OSC1)	VIH2	$0.7V_{CC}$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	$0.2V_{CC}$	V		15
Output high voltage (1) (DB0–DB7)	VOH1	$0.75V_{CC}$	—	—	V	$-I_{OH} = 0.1$ mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	$0.2V_{CC}$	V	$I_{OL} = 0.1$ mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	$0.8V_{CC}$	—	—	V	$-I_{OH} = 0.04$ mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	$0.2V_{CC}$	V	$I_{OL} = 0.04$ mA	8
Driver on resistance (COM)	R_{COM}	—	2	20	kΩ	$\pm I_d = 0.05$ mA, VLCD = 4 V	13
Driver on resistance (SEG)	R_{SEG}	—	2	30	kΩ	$\pm I_d = 0.05$ mA, VLCD = 4 V	13
Input leakage current	I_{LI}	-1	—	1	μA	$V_{IN} = 0$ to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/W)	$-I_p$	10	50	120	μA	$V_{CC} = 3$ V	
Power supply current	I_{CC}	—	150	300	μA	R_f oscillation, external clock $V_{CC} = 3$ V, $f_{OSC} = 270$ kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	V_{CC} –V5, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	V_{CC} –V5, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics ($V_{CC} = 2.7$ to 4.5 V, $T_a = -30$ to $+75^\circ C$ ³)**Clock Characteristics**

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Note*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		
	External clock rise time	t_{rcp}	—	—	0.2	μs		
	External clock fall time	t_{fcp}	—	—	0.2	μs		
R_f oscillation	Clock oscillation frequency	f_{osc}	190	270	350	kHz	$R_f = 75$ k Ω , $V_{CC} = 3$ V	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics**Write Operation**

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time		t_{cycE}	1000	—	—	ns	Figure 25
Enable pulse width (high level)		PW_{EH}	450	—	—		
Enable rise/fall time		t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/W to E)		t_{AS}	60	—	—		
Address hold time		t_{AH}	20	—	—		
Data set-up time		t_{DSW}	195	—	—		
Data hold time		t_H	10	—	—		

Read Operation

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time		t_{cycE}	1000	—	—	ns	Figure 26
Enable pulse width (high level)		PW_{EH}	450	—	—		
Enable rise/fall time		t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/W to E)		t_{AS}	60	—	—		
Address hold time		t_{AH}	20	—	—		
Data delay time		t_{DDR}	—	—	360		
Data hold time		t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—	ns	
Clock set-up time		t_{CSU}	500	—	—	ns	
Data set-up time		t_{SU}	300	—	—	ns	
Data hold time		t_{DH}	300	—	—	ns	
M delay time		t_{DM}	−1000	—	1000	ns	
Clock rise/fall time		t_{ct}	—	—	200	ns	

Power Supply Conditions Using Internal Reset Circuit

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time		t_{rcc}	0.1	—	10	ms	Figure 28
Power supply off time		t_{OFF}	1	—	—	ms	

DC Characteristics ($V_{CC} = 4.5$ to 5.5 V, $T_a = -30$ to $+75^\circ\text{C}$ *³)

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	2.2	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	-0.3	—	0.6	V		6
Input high voltage (2) (OSC1)	VIH2	$V_{CC}-1.0$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	1.0	V		15
Output high voltage (1) (DB0–DB7)	VOH1	2.4	—	—	V	$-I_{OH} = 0.205$ mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	0.4	V	$I_{OL} = 1.2$ mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	0.9 V_{CC}	—	—	V	$-I_{OH} = 0.04$ mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	0.1 V_{CC}	V	$I_{OL} = 0.04$ mA	8
Driver on resistance (COM)	R _{COM}	—	2	20	kΩ	$\pm I_d = 0.05$ mA, VLCD = 4 V	13
Driver on resistance (SEG)	R _{SEG}	—	2	30	kΩ	$\pm I_d = 0.05$ mA, VLCD = 4 V	13
Input leakage current	I_{LI}	-1	—	1	μA	$V_{IN} = 0$ to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/W)	$-I_p$	50	125	250	μA	$V_{CC} = 5$ V	
Power supply current	I_{CC}	—	350	600	μA	R_f oscillation, external clock $V_{CC} = 5$ V, $f_{osc} = 270$ kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	$V_{CC}-V_5$, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	$V_{CC}-V_5$, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics ($V_{CC} = 4.5$ to 5.5 V, $T_a = -30$ to $+75^\circ C$ *³)**Clock Characteristics**

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		11
	External clock rise time	t_{rcp}	—	—	0.2	μs		11
	External clock fall time	t_{fcp}	—	—	0.2	μs		11
R_f oscillation	Clock oscillation frequency	f_{osc}	190	270	350	kHz	$R_f = 91$ kΩ $V_{CC} = 5.0$ V	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics**Write Operation**

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time		t_{cycE}	500	—	—	ns	Figure 25
Enable pulse width (high level)		PW_{EH}	230	—	—		
Enable rise/fall time		t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/W to E)		t_{AS}	40	—	—		
Address hold time		t_{AH}	10	—	—		
Data set-up time		t_{DSW}	80	—	—		
Data hold time		t_H	10	—	—		

Read Operation

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time		t_{cycE}	500	—	—	ns	Figure 26
Enable pulse width (high level)		PW_{EH}	230	—	—		
Enable rise/fall time		t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/W to E)		t_{AS}	40	—	—		
Address hold time		t_{AH}	10	—	—		
Data delay time		t_{DDR}	—	—	160		
Data hold time		t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

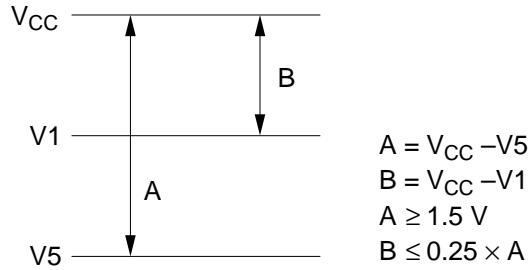
Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—		
Clock set-up time		t_{CSU}	500	—	—		
Data set-up time		t_{SU}	300	—	—		
Data hold time		t_{DH}	300	—	—		
M delay time		t_{DM}	—1000	—	1000		
Clock rise/fall time		t_{ct}	—	—	100		

Power Supply Conditions Using Internal Reset Circuit

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time		t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time		t_{OFF}	1	—	—		

Electrical Characteristics Notes

- All voltage values are referred to GND = 0 V.

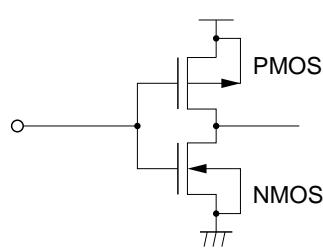


$A = V_{CC} - V_5$
 $B = V_{CC} - V_1$
 $A \geq 1.5$ V
 $B \leq 0.25 \times A$

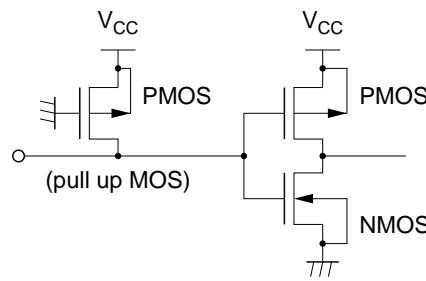
The conditions of V_1 and V_5 voltages are for proper operation of the LSI and not for the LCD output level. The LCD drive voltage condition for the LCD output level is specified as LCD voltage $VLCD$.

- $V_{CC} \geq V_1 \geq V_2 \geq V_3 \geq V_4 \geq V_5$ must be maintained.
- For die products, specified at 75°C.
- For die products, specified by the die shipment specification.
- The following four circuits are I/O pin configurations except for liquid crystal display output.

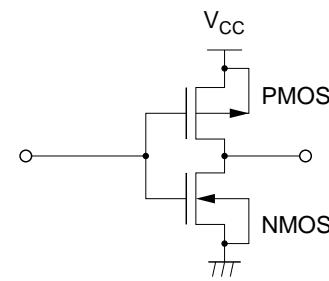
Input pin
Pin: E (MOS without pull-up)



Pins: RS, R/W (MOS with pull-up)

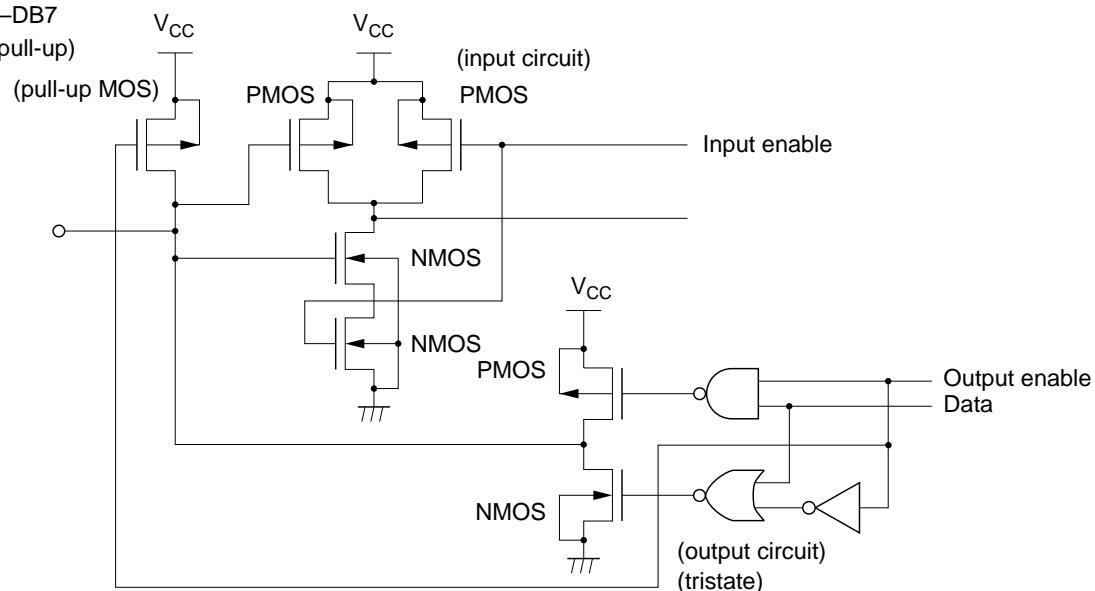


Output pin
Pins: CL1, CL2, M, D

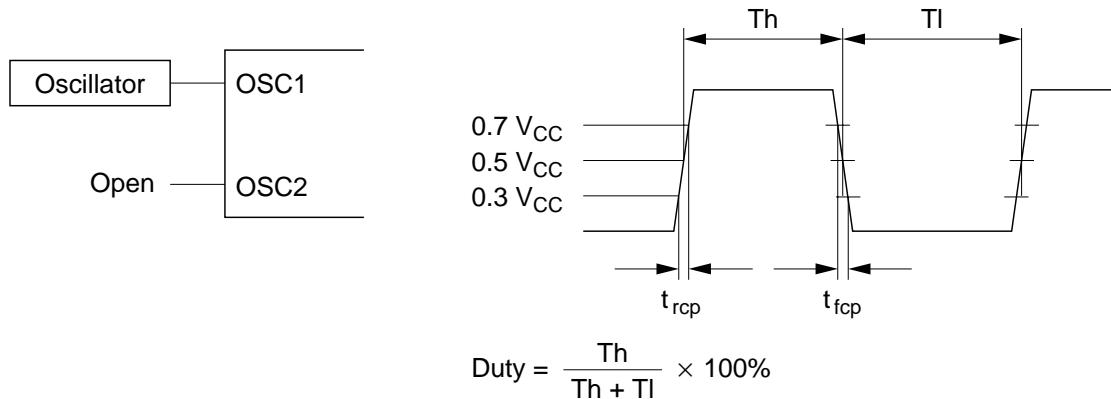


I/O Pin

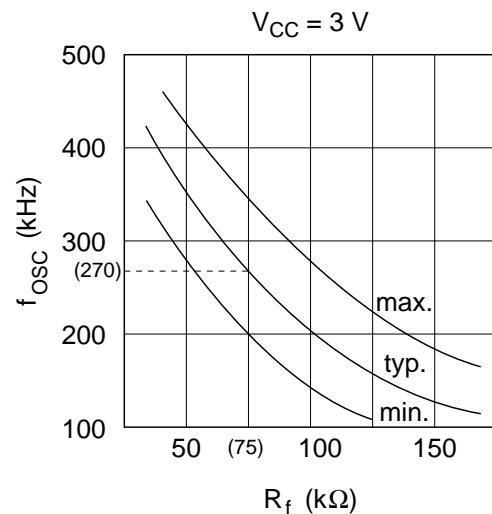
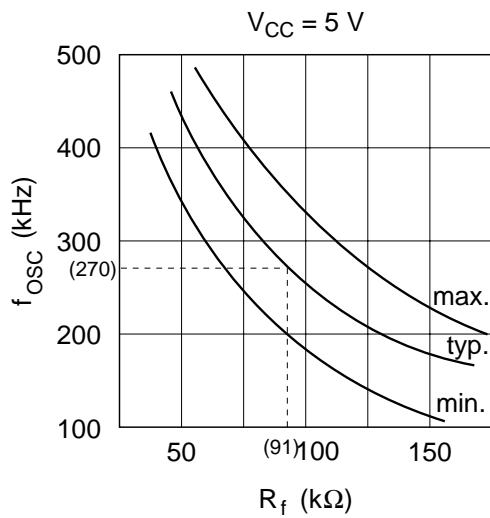
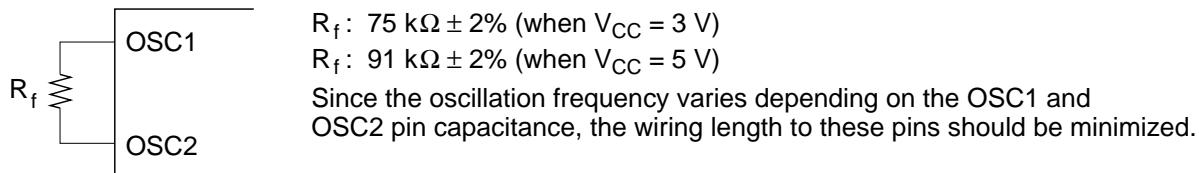
Pins: DB0 –DB7
(MOS with pull-up)



6. Applies to input pins and I/O pins, excluding the OSC1 pin.
7. Applies to I/O pins.
8. Applies to output pins.
9. Current flowing through pull-up MOSs, excluding output drive MOSs.
10. Input/output current is excluded. When input is at an intermediate level with CMOS, the excessive current flows through the input circuit to the power supply. To avoid this from happening, the input level must be fixed high or low.
11. Applies only to external clock operation.



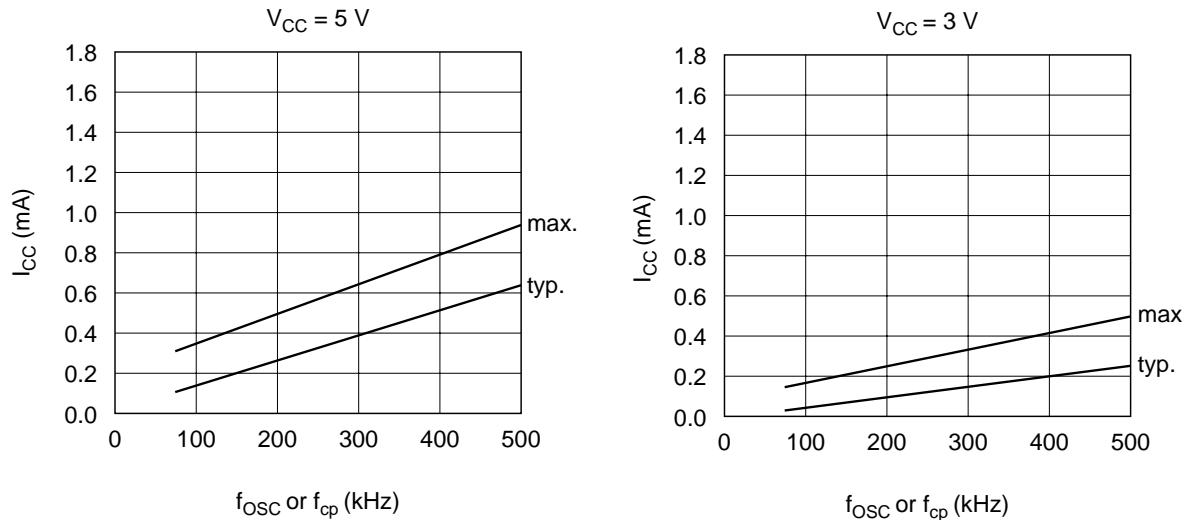
12. Applies only to the internal oscillator operation using oscillation resistor R_f .



13. RCOM is the resistance between the power supply pins (V_{CC} , V1, V4, V5) and each common signal pin (COM1 to COM16).

RSEG is the resistance between the power supply pins (V_{CC} , V2, V3, V5) and each segment signal pin (SEG1 to SEG40).

14. The following graphs show the relationship between operation frequency and current consumption.

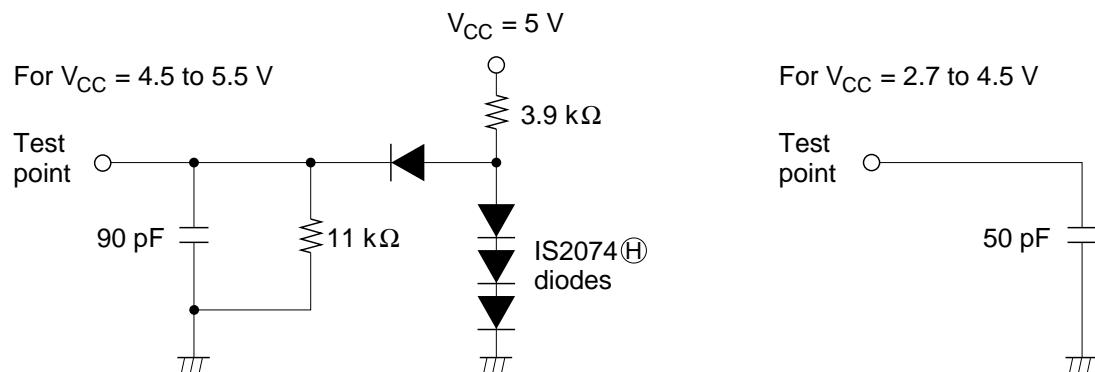


15. Applies to the OSC1 pin.

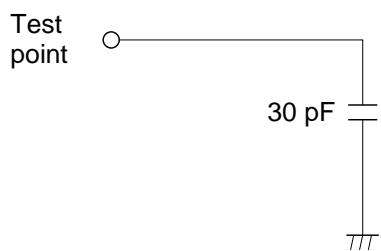
16. Each COM and SEG output voltage is within $\pm 0.15\text{ V}$ of the LCD voltage (V_{CC} , V1, V2, V3, V4, V5) when there is no load.

Load Circuits

Data Bus DB0 to DB7



External Driver Control Signals: CL1, CL2, D, M



Timing Characteristics

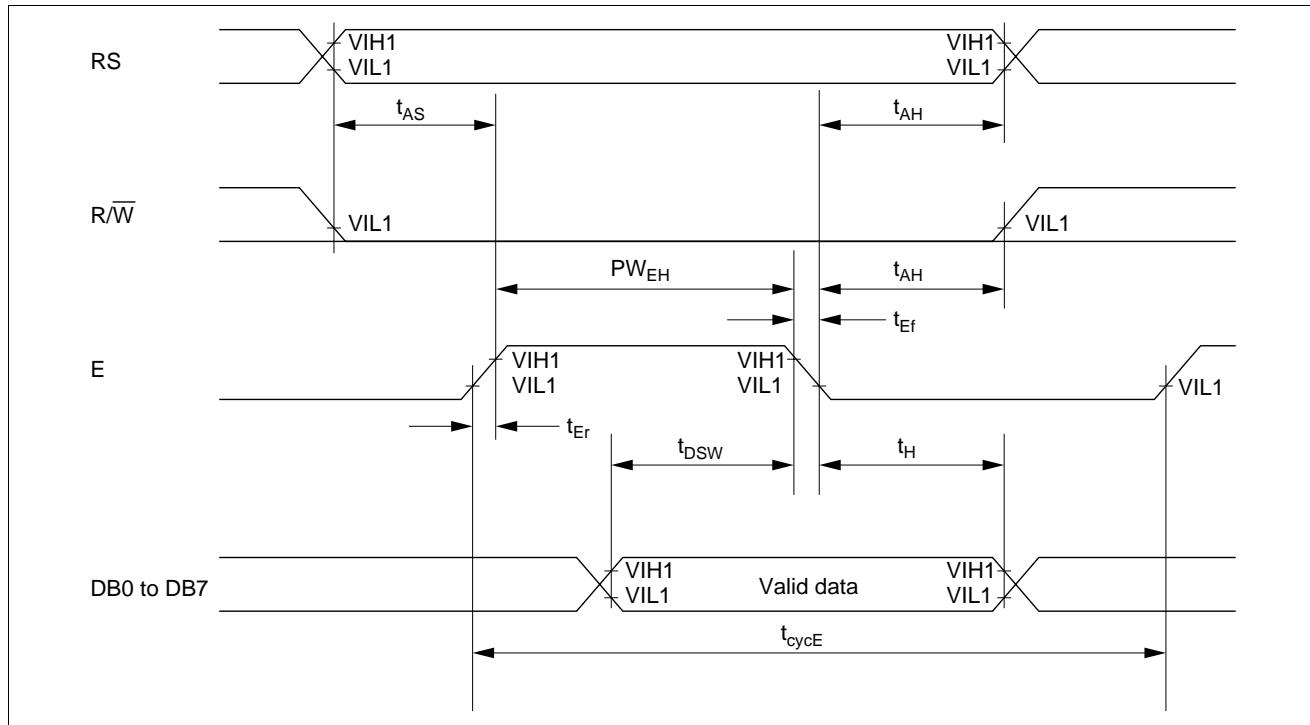


Figure 25 Write Operation

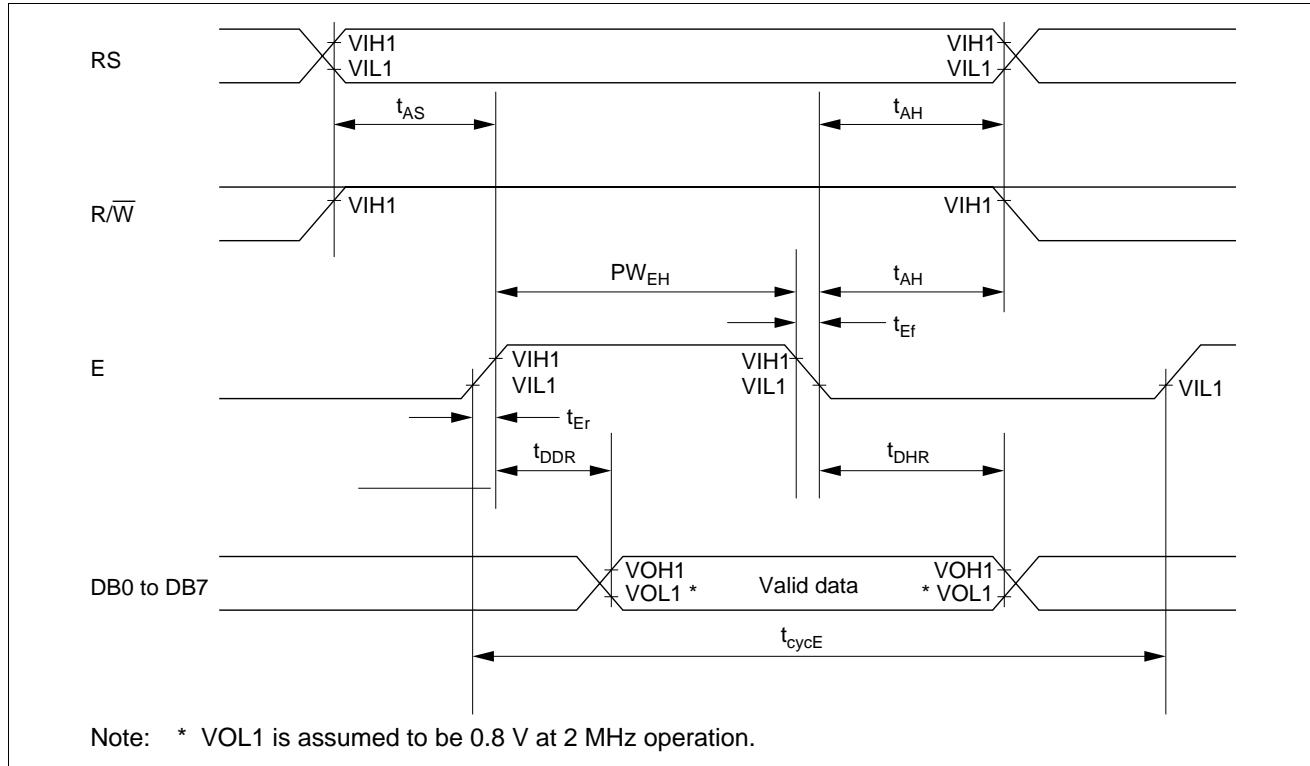


Figure 26 Read Operation

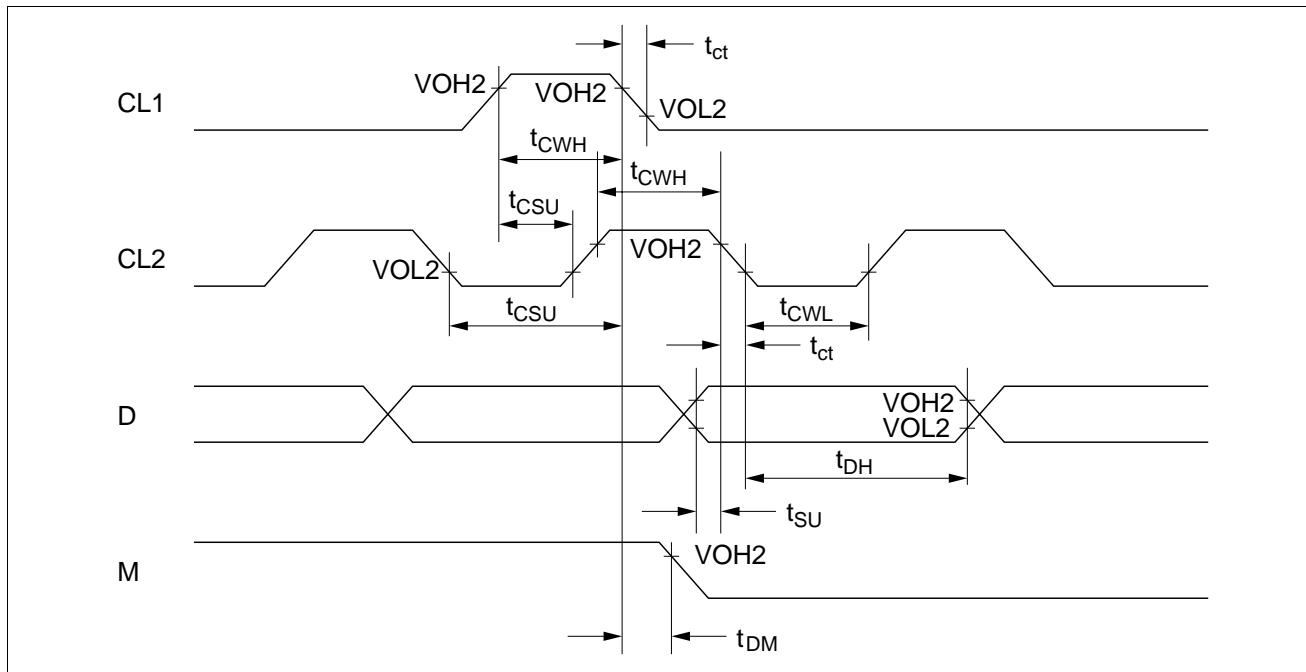


Figure 27 Interface Timing with External Driver

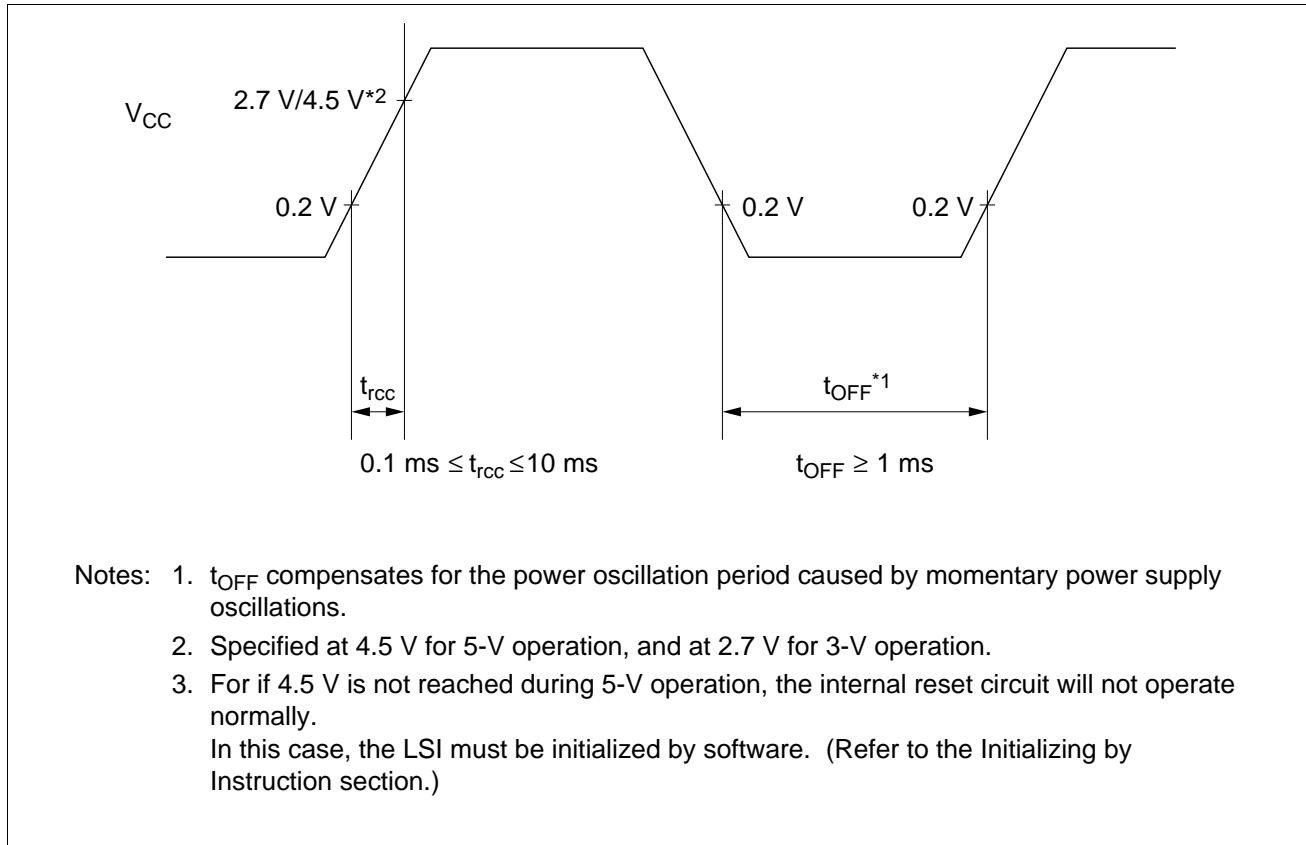


Figure 28 Internal Power Supply Reset

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

**Hitachi, Ltd.**

Semiconductor & Integrated Circuits.

Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan
Tel: Tokyo (03) 3270-2111 Fax: (03) 3270-5109

URL	NorthAmerica	:	http://semiconductor.hitachi.com/
	Europe	:	http://www.hitachi-eu.com/hel/ecg
	Asia (Singapore)	:	http://www.has.hitachi.com.sg/grp3/sicd/index.htm
	Asia (Taiwan)	:	http://www.hitachi.com.tw/E/Product/SICD_Frame.htm
	Asia (HongKong)	:	http://www.hitachi.com.hk/eng/bo/grp3/index.htm
	Japan	:	http://www.hitachi.co.jp/Sicd/indx.htm

For further information write to:

Hitachi Semiconductor (America) Inc. 179 East Tasman Drive, San Jose, CA 95134 Tel: <1>(408) 433-1990 Fax: <1>(408) 433-0223	Hitachi Europe GmbH Electronic Components Group Dornacher Straße 3 D-85622 Feldkirchen, Munich Germany Tel: <49>(89) 9 1980-0 Fax: <49>(89) 9 29 30 00 Hitachi Europe Ltd. Electronic Components Group. Whitebrook Park Lower Cookham Road Maidenhead Berkshire SL6 8YA, United Kingdom Tel: <44>(1628) 585000 Fax: <44>(1628) 778322	Hitachi Asia Pte. Ltd. 16 Collyer Quay #20-00 Hitachi Tower Singapore 049318 Tel: 535-2100 Fax: 535-1533 Hitachi Asia Ltd. Taipei Branch Office 3F, Hung Kuo Building, No.167, Tun-Hwa North Road, Taipei (105) Tel: <886>(2) 2718-3666 Fax: <886>(2) 2718-8180	Hitachi Asia (Hong Kong) Ltd. Group III (Electronic Components) 7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Tsim Sha Tsui, Kowloon, Hong Kong Tel: <852>(2) 735 9218 Fax: <852>(2) 730 0281 Telex: 40815 HITEC HX
---	---	--	--

Copyright © Hitachi, Ltd., 1998. All rights reserved. Printed in Japan.



UNCONTROLLED DOCUMENT

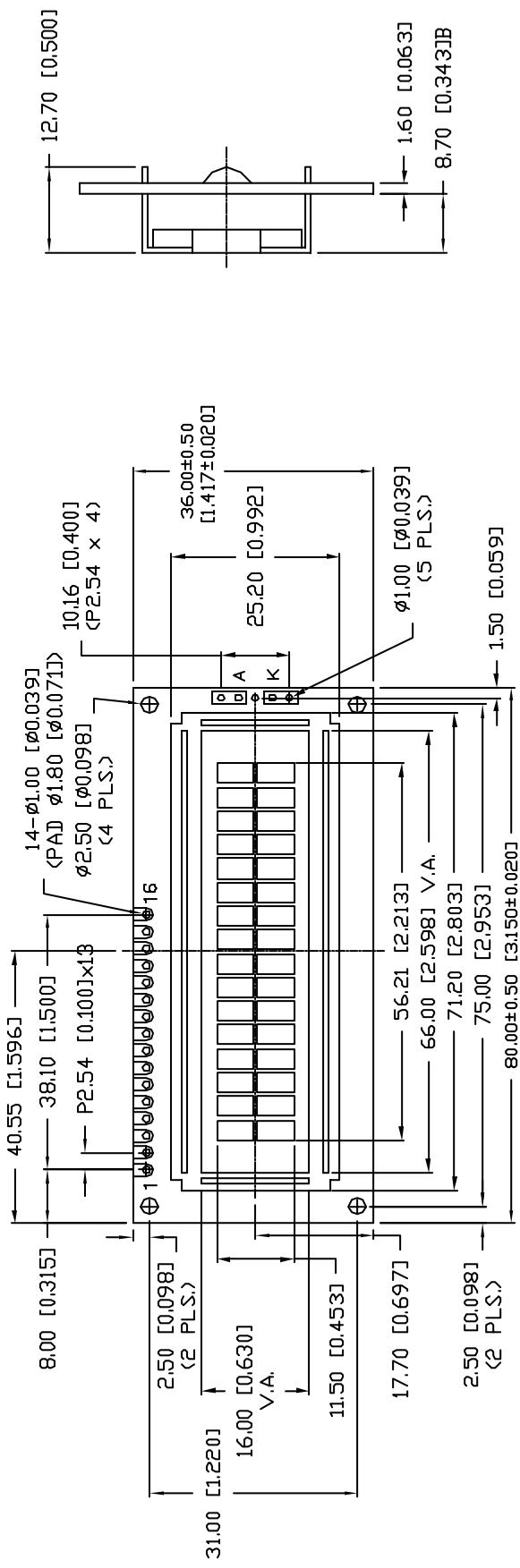
PART NUMBER REV.

LCM-S01602DSF/A

PART NUMBER REV.

CHARACTER DETAIL

CAUTION: STATIC SENSITIVE DEVICE
FOLLOW PROPER E.S.D. HANDLING PROCEDURES
WHEN WORKING WITH THIS PART.



*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DRAWING ARE ± 0.003 .

—DECIMAL FRACTION

MAINTAINED IN THIS DOCUMENT

LCM-S01602DSF/A
5.56mm CHARACTER HEIGHT, 5 x 8 DOT MATRIX
STN, TRANSELECTIVE WITH LED BACKLIGHT,
16 x 2 LCD MODULE, 1/16 DUTY, 1/5 BIAS

10

SARI NUMBER

LCM-S01602DSF/A

卷之三

IN PART FROM OBSOLETIVE AND DISSEMINATED

DATE: 1.22.02
PAGE: 1 OF 2
SCALE: N/A

UNCONTROLLED DOCUMENT



PART NUMBER

卷之三

15

二二

THEORY

HEIGHT, 3 X 8

E WITH LED

E. 1/16 DUTY

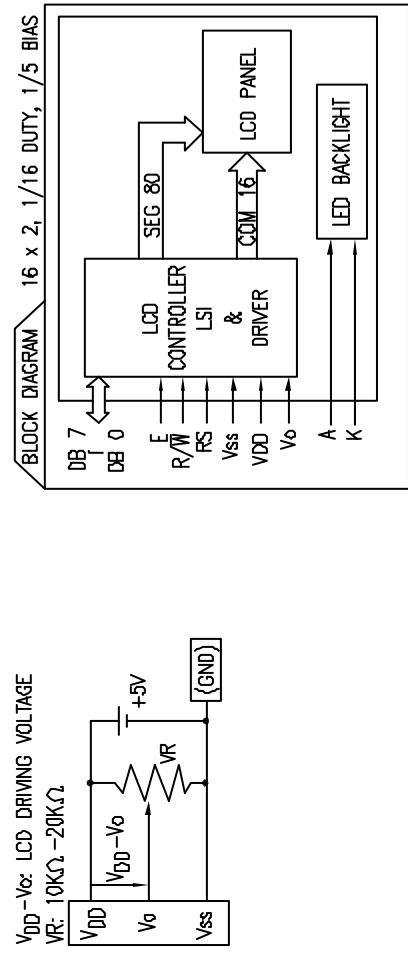
UNCONTROLLED DOCUMENT

PART NUMBER
REV.

LCM-S01602DSF/A

PIN CONFIGURATION

PIN NO.	SYMBOL	LEVEL	FUNCTION
1	V _{SS}	-	GND (0V)
2	V _{DD}	-	POWER SUPPLY FOR LCD DRIVE
3	V _O	-	REGISTER SELECT SIGNAL H: DATA INPUT L: INSTRUCTION INPUT
4	R/S	H/L	H: DATA READ (MODULE-->MPU) L: DATA WRITE (MODULE<-MPU)
5	R/W	H/L	H,L>L: ENABLE
6	E	H,H>L	DATA BUS SOFTWARE SELECTABLE 4 OR 8 BIT MODE.
7~14	DB0~DB7	H/L	DATA BUS ANODE LED BACKLIGHT
15	A	-	CATHODE LED BACKLIGHT
16	K	-	CATHODE LED BACKLIGHT



ELECTRICAL CHARACTERISTICS

ITEM	SYMBOL	CONDITION	STANDARD VALUE		UNIT
			MIN.	TYP.	MAX.
SUPPLY VOLTAGE FOR LOGIC	V _{DD} -V _{SS}	-	-	5.0	-
SUPPLY CURRENT FOR LOGIC	I _{DD}	V _{DD} =5V	-	2.0	3.0
INPUT VOLTAGE	V _{IH}	-	2.2	-	VDD
OUTPUT VOLTAGE	V _{IL}	-	0	-	0.6
VOLTAGE	V _F	f=160mA	-	4.2	4.6
CURRENT	I _f	-	-	160	mA
*LED BACKLIGHT POWER CONSUMPTION	P _D	-	-	656	nW
LUMINOUS COLOR	L	f=160mA	70	-	cd/m ²
	-	-	-	-	mm

*ONLY APPLIES TO MODULES WITH BACKLIGHT

*UNLESS OTHERWISE SPECIFIED TOLERANCES FOR DECIMAL PRECISION ARE: X=±1 (±0.039), XX=±0.5 (±0.020), XXX=±0.25 (±0.010), XXXX=±0.127 (±0.006).

LEAD SIZE=4.05 (±0.02), LEAD LENGTH=±0.75 (±0.002), MIN=+0.000, MAX=+0.000, NIN= -0.000

CONFIDENTIAL INFORMATION

THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF LUMEX INC., EXCEPT AS SPECIFICALLY AUTHORIZED IN WRITING BY LUMEX INC. THE HOLDER OF THIS DOCUMENT SHALL KEEP ALL INFORMATION CONTAINED HEREIN CONFIDENTIAL AND SHALL PROTECT SAME IN WHOLE OR IN PART FROM DISCLOSURE AND DISSEMINATION TO ALL THIRD PARTIES.

RELIABILITY NOTE

OUR MANY YEARS OF EXPERIENCE AND ACCUMULATION INDICATE THAT SOLDER HEAT IS A MAJOR CAUSE OF EARLY AND FUTURE FAILURE. PLEASE PAY ATTENTION TO YOUR SOLDERING PROCESS.

UNCONTROLLED DOCUMENT

REV.	PART NUMBER	CHECKED BY:	APPROVED BY:	DATE:
	LCM-S01602DSF/A	CT		1.22.02 PAGE: 2 OF 2 SCALE: N/A
	5.56mm CHARACTER HEIGHT, 5 x 8 DOT MATRIX, STN, TRANSFLECTIVE WITH LED BACKLIGHT, 16 x 2 LCD MODULE, 1/16 DUTY, 1/5 BIAS.			www.lumex.com.tw

Document information

Info	Content
Keywords	Mifare, ICode, Micore, RC400, RC500, RC530, RC531, RC632, RFID-Reader, Directly Matched Antenna Design
Abstract	This application note describes how to design an antenna for the MICORE contactless reader IC family.
BL-ID Doc Number	M077925 update

Revision history

Rev	Date	Description
01.01	20040501	Initial version of Application Note; Directly Matched Antenna Design for Micore Reader ICs
02.05	20060510	Change of layout, general update on the content, correction in formula for C2, Add the changes of the EMC filter, and the antenna example, correction of table 4

Contact information

For additional information, please visit: <http://www.semiconductors.philips.com>

For sales office addresses, please send an email to: sales.addresses@www.semiconductors.philips.com

1. Introduction

This document describes the Micore antenna design for applications where the antenna is directly connected to the reader board (without any cable in between the reader and the antenna).

The MIFORE reader IC family includes the MF RC500, MF RC530, MF RC531, SL RC 400 and the CL RC 632. The antenna design and matching is the same for all of these ICs.

1.1 How to use this document

For a good overview the **Micore Antenna Principle section** shows the blocks of each reader antenna design, how these blocks fit to Micore, and also the complete schematic of a Micore Antenna.

Because there are many parameters that influence the overall performance of an antenna, a basic RF knowledge is needed to design an antenna that takes **all** these parameters into account. Although this (specific) knowledge is important for understanding the background, a **complete antenna design** (as shown in [Fig 1](#) and [Fig 2](#)) **can be done without RF-specific knowledge**. Therefore the complete design of a directly matched antenna is divided into two design levels:

1. The **basic parameter design** is dedicated to those, who are not really familiar with RF design and its tools. Although a certain number of requirements and parameters are frozen, the design will yield an optimized antenna. A guideline, the required formulas, and a tool to support the design is described and provided. Depending on some simplifications this design might not be *completely* compliant to the ISO/IEC14443, even though it fully satisfies the Mifare and I-Code requirements.
2. The **full parameter design** is dedicated to those, who want to cover all the specific cases of various design requirements to make a perfect design for the application. This design describes how to build an antenna, which is fully ISO/IEC14443 compliant (including the higher bit rates). The complete antenna design covers all the open parameters, some background information for the design, and the general behaviour of such an antenna.

In [section 5](#) some useful antenna design hints are given.

Appendix [6.1](#) contains a checklist to help optimizing and finishing an antenna design for a proximity reader based on the Micore. It also contains a collection of relevant formulas and an antenna design example.

“Card” in this document means a contactless smart card according to the ISO14443 (or MIFARE[®]) or a contactless tag / label according to the ISO15693 (or I-Code[®]).

Please be aware: This application note cannot and does not replace any of the relevant datasheets.

It is required to have some general knowledge about RFID proximity antennas as given by the application note in [ref. \[1\]](#).

For higher bit rates refer to the application note in [ref. \[2\]](#).

2. Micore antenna principle

The Micore is a single reader IC family designed to achieve operating distances up to 100mm without external amplifiers. The design rules and parameters are basically the same for ISO14443, Mifare®, ISO15693 and I-Code®, i.e. the same antenna can be used to communicate with all products mentioned.

A complete antenna design always includes the antenna coil and resonance circuit design, the matching of the antenna circuit, the receiving circuitry and the EMC filtering (see [Fig.1](#))

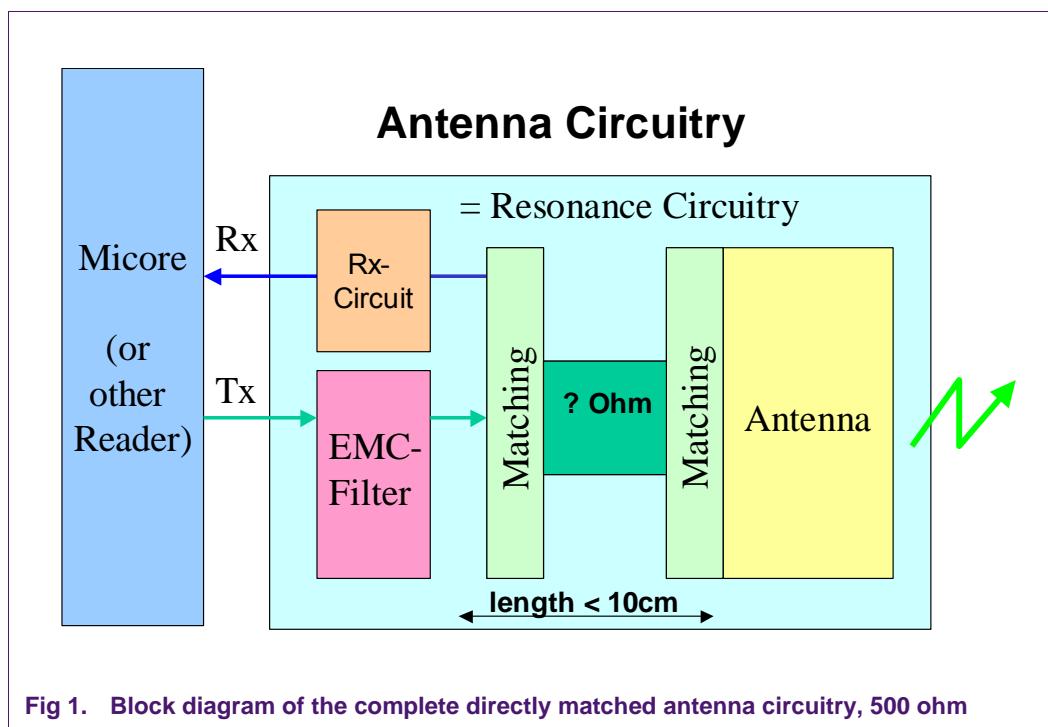


Fig 1. Block diagram of the complete directly matched antenna circuitry, 500 ohm

Although some of these blocks may contain only a few passive components, it is important to consider all these blocks and all their functionality to guarantee the proper working of the complete antenna.

The overall functionality can be separated into three basic functions:

1. **Transmit power:** The radiated magnetic field has to be maximized considering the radiation and datasheet limits, especially the limits for the radiation of the harmonics (up to 1GHz).
2. **Transmit data:** The 10% or 100% ASK modulated data signal has to be transmitted in such way, that every card is able to receive it. The signal shape and timing (i.e. the Q-factor) has to be considered.
3. **Receive data:** The card's answer has to be delivered to the receive input of the Micore considering the datasheet limits.

Basically this complete antenna circuitry consists of 8 capacitors, 2 inductors, 2 or 4 resistors and the symmetrical antenna coil ($L_a + L_b$) as shown in [Fig.2](#).

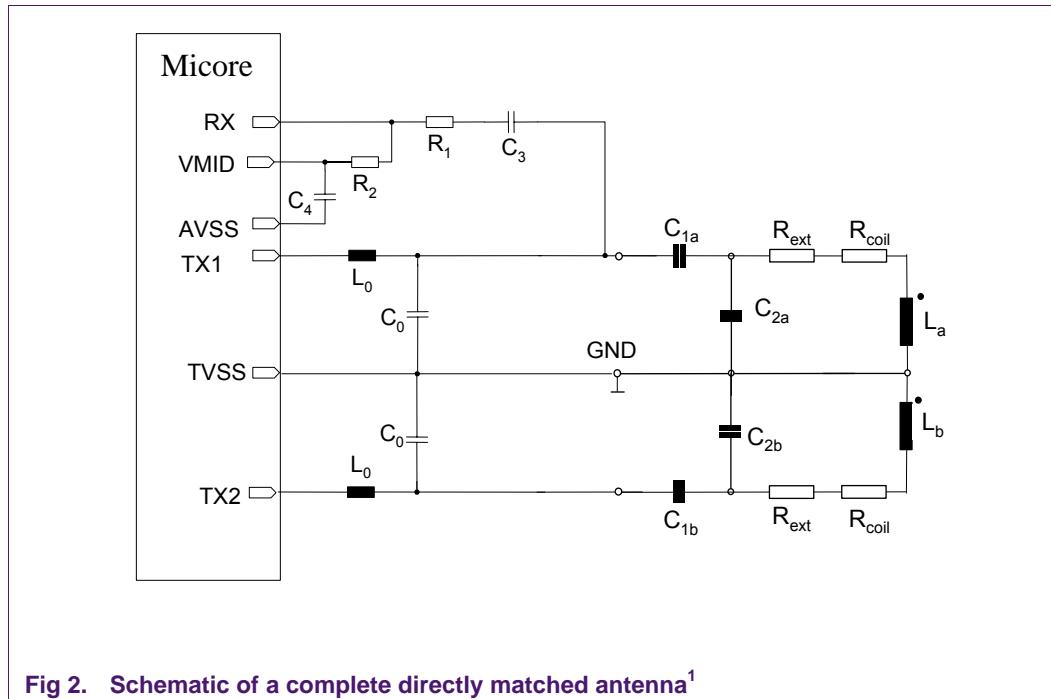


Fig 2. Schematic of a complete directly matched antenna¹

Table 1: Components for directly matched antenna

Abbreviation	Explanation
L_a , L_b and R_{coil}	Antenna coil components according to section 3.2
R_{ext}	External resistor to adjust the quality factor. Please also consider the power consumption.
C_0 , C_1 ... C_3	Typically 0402, 0603 or 0805 SMD parts with low tolerance ($< \pm 2\%$). NPO is required. The voltage limit has to be considered.
C_4	XR7 capacitor.
L_0	Typically a small inductance with high Q for general applications. The frequency range and the maximum allowed current have to be considered. This inductance should be magnetically shielded.
R_1 and R_2	Small 0402, 0603 or 0805 SMD parts.

Remark: Plan on placing two capacitors, C_{1a} and C_{2a} in parallel to achieve the required matching values. The tuning procedure most likely will return matching values, which cannot be realized with a single capacitor.

1. The GND connection of the antenna coil (center tapping) is not required.

3. Basic parameter design for 106kbit/s

A complete Micore reader antenna design can be done in steps based on the certain number of given parameters given below (see [section 3.1](#)). Every step is described in one of the following sections . The best way to design antenna is to follow the steps given below:

1. Design a coil, measure L and R or L and Q (see [section 3.2](#)).
2. Calculate the resonance capacitors to design a resonance circuitry together with the coil (see [section 3.3](#) to [3.5](#)).
3. Tune this resonance circuitry to the required impedance (see [section 3.6](#)).
4. Connect the resonance circuitry to the EMC low pass filter output, check the I_{TVDD} and if necessary retune the components for optimum performance.
5. Check & adjust the Q-factor (see [section 3.7](#)).
6. Check & adjust the receive circuitry (see [section 3.8](#)).

Remark: *This design might not be completely compliant to the ISO/IEC 14443 (with higher bit rates) due to some simplification in the design steps. For an ISO/IEC 14443 compliant design the complete design steps need to be taken into account (see section 4).*

3.1 Given parameters

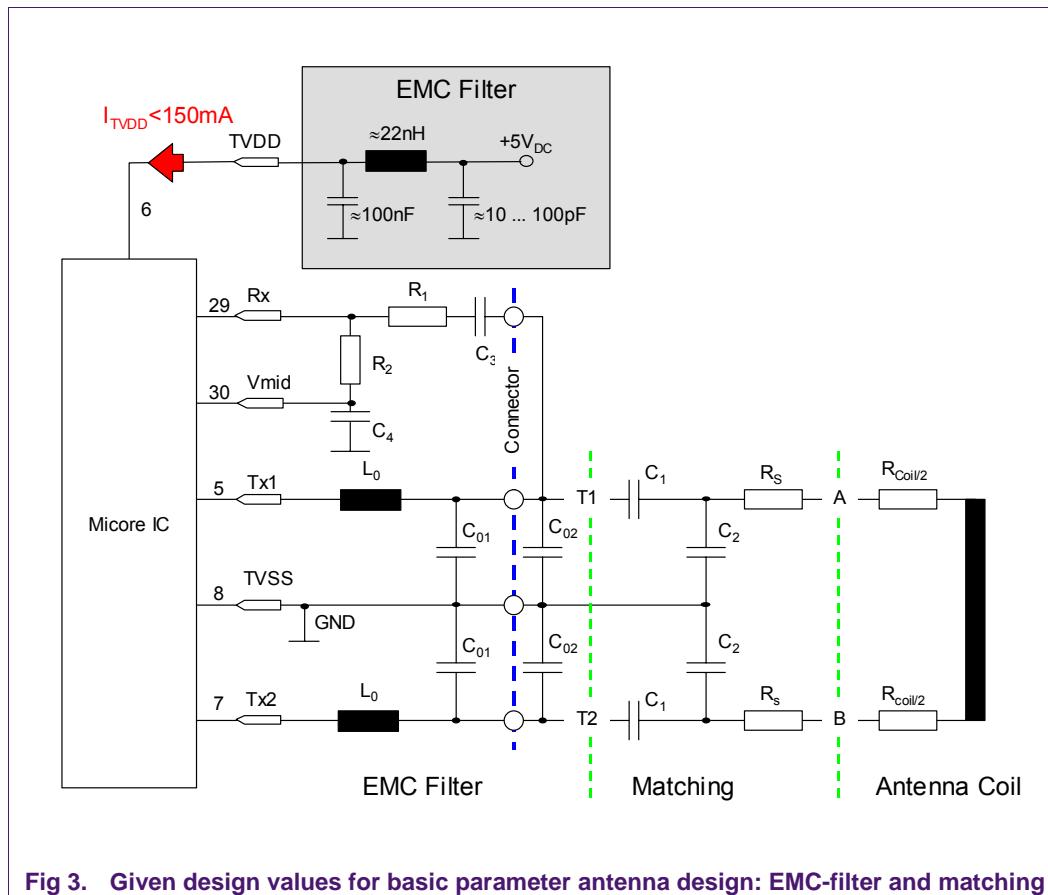


Fig 3. Given design values for basic parameter antenna design: EMC-filter and matching

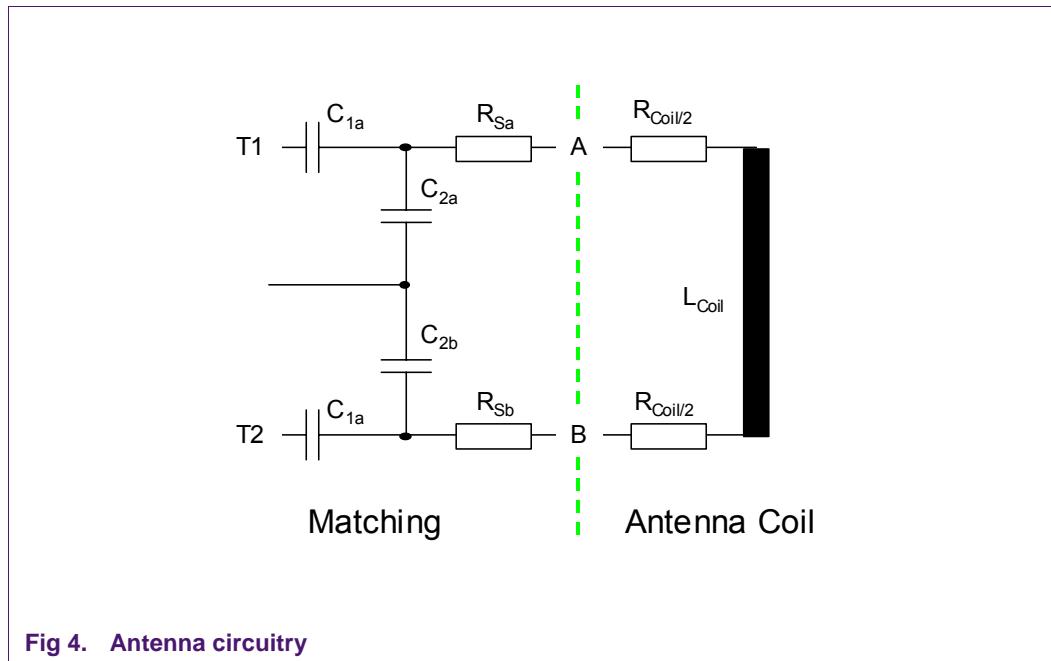
To reduce the number of parameters for the basic parameter antenna design, the following fixed component values shall be used:

Table 2: Fixed value of components of directly matched antenna

Component	Value
L_0	$1\mu\text{H}$ (e.g. TDK NL322522T-1R0J)
C_{01}, C_{02}	68pF each (Ceramic NP0, tolerance $\leq \pm 2\%$)
C_3	1nF (Ceramic NP0, tolerance $\leq \pm 10\%$)
C_4	100nF (Ceramic X7R, tolerance $\leq \pm 10\%$)
R_1	$470\Omega \dots 4.7\text{k}\Omega^2$
R_2	820Ω

Remark: GND should be one GND-plane for the Micore circuitry itself, and should be directly connected to the center pin of the shielding, if shielding is applicable. It is not required to connect the center tapping of the antenna coil.

² The exact value has to be chosen according to the value of the antenna coil ([section 3.8](#)).



The complete directly matched antenna consists of the parts as shown in [Fig 4³](#).

Table 3: Components of antenna circuitry

Component	Value
L_{Coil}	Antenna coil (see section 3.2)
R_{coil}	Internal resistor of the antenna coil (see section 3.2)
R_{Sa}, R_{Sb}	External resistor (see section 3.7)
$C_{1a}, C_{1b}, C_{2a}, C_{2b}$	Capacitors to match impedance and resonance (see section 3.5)

3. It is not required to connect a center tapping of the coil to GND.

3.2 The antenna coil

As a start the required antenna coil shall be designed within the mechanical application requirements and according to the general antenna design rules ([see \[1\]](#)).

The inductance $L = L_a + L_b$ and series resistor $R_L = 2 \cdot R_{coil}$ **shall be measured** between A and B in [Fig 5](#). This measurement could be done with an impedance analyzer or an LCR meter at a frequency of $f = 13.56\text{ MHz}$.

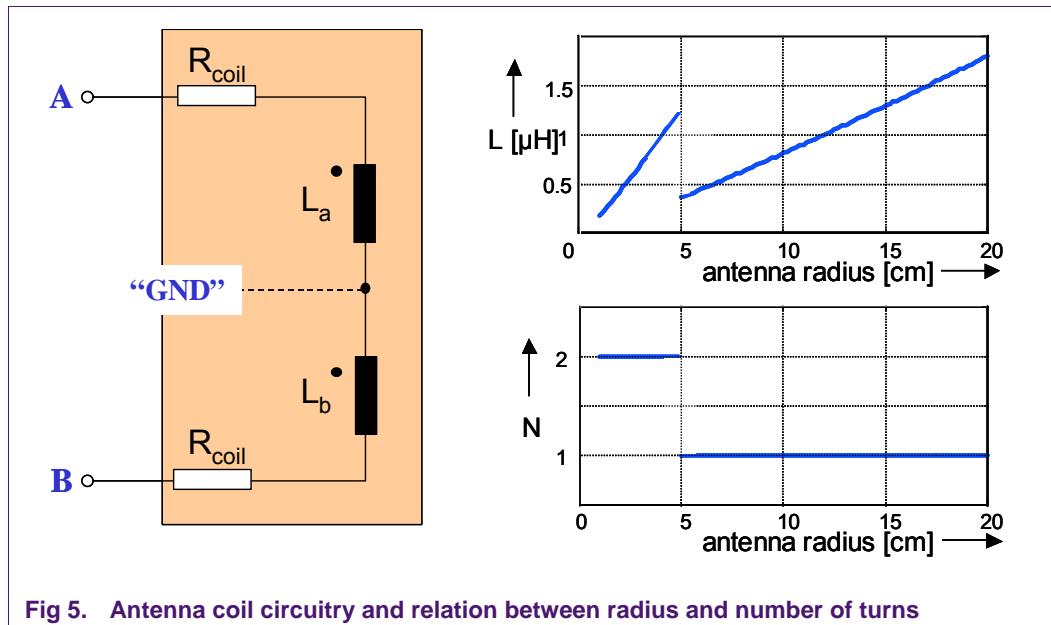
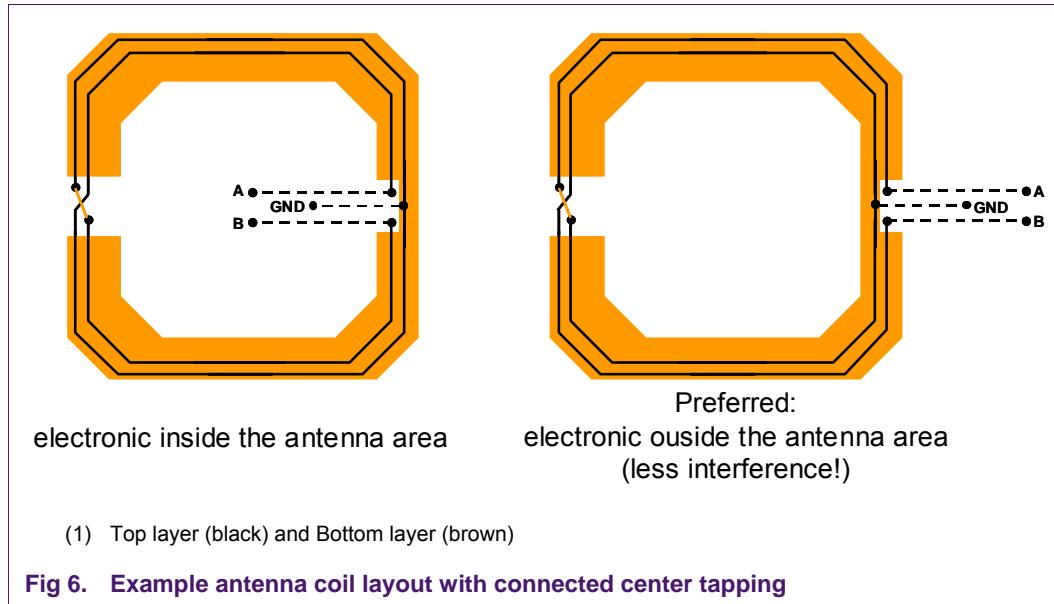


Fig 5. Antenna coil circuitry and relation between radius and number of turns

Depending on the antenna radius use either a single turn coil ($r \geq 5\text{ cm}$) or a double turn coil ($r < 5\text{ cm}$) for each coil L_a and L_b .

The antenna coil shall be symmetrical.

The (electrical) center point **can** be connected to GND. If this center point is connected to GND, it is suggested to use the same number of complete turns for both L_a and L_b (even number of turns for L), to keep this center point as near as possible to the antenna coil connector. If one and a half turns are used, no center tapping shall be used. The shielding (if applicable) shall be connected to GND directly at the antenna connector (center tapped). Two layout examples based on a 2-layer board are shown in [Fig 6](#).



According to the given design rules (coil radius \approx operating distance, and number of turns as shown in [Fig 5](#)) the inductance of the antenna coil usually is between

$$L = L_a + L_b = 300nH \dots 2\mu H$$

$$R_L = 2 \cdot R_{coil} = 0.5\Omega \dots 5\Omega$$

Remark: These values are only typical values.

The turn direction of the two parts of the coil (L_a and L_b) should be considered!

The antenna shall be measured completely ($L_a + L_b$ between A and B) because of the coupling between the two parts of the antenna coil (L_a and L_b).

The measured value of R_L is usually too high due to the high Q-factor of the coil. This has to be considered as follows. The effect is really low, as on one hand the influence of a slight change of R_L on the matching is not high, and on the other hand the Q-factor has to be checked later anyway.

To be measured at 13.56 MHz.

It is not required to connect the center tapping of the coil, but it is required to connect the shielding (if applicable) to GND.

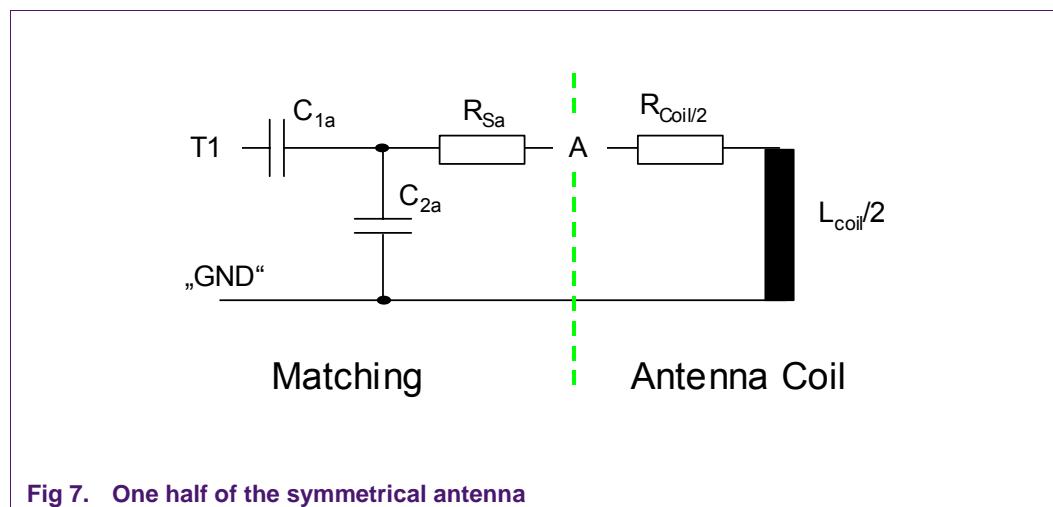
3.3 Simplification due to symmetry

Due to the symmetrical circuitry of the whole antenna for one coil it is required that:

$$L_a = L_b$$

$$Z_a = Z_b = \frac{Z_{ant}}{2} \text{ (as shown in Fig 7).}$$

As this symmetry is a general requirement for the proper function of the antenna, it can be used to simplify the whole circuitry. In the following only the upper half of the complete antenna circuit is used to calculate the matching network and the external resistor(s). The lower half looks and behaves exactly the same.



Although this simplification can be done in theory, the practical measurement cannot be done with this simplified model, because of the coupling between L_a and L_b . So all the impedance *measurements* have to be done with the complete antenna circuit with both sides (between T1 and T2 or A and B) and NOT between T1(or A) and GND or T2 (or B) and GND.

3.4 The external resistor

Together with the general definition of the Q-factor of the coil

$$Q = \frac{\omega L}{R_{Coil}} \quad \text{or} \quad R_{Coil} = \frac{\omega L}{Q}$$

the overall resistor (that specifies the overall Q)

$$R = 2 \cdot R_S + R_{Coil} = R_{Sa} + R_{Sb} + 2 \cdot \frac{R_{coil}}{2}$$

and the requirements for

$$Q = 30^4$$

the external resistor R_{Sa} and R_{Sb} can be estimated:

$$R_{Sa} = R_{Sb} = \frac{1}{2} \cdot (R - R_{Coil}) = \frac{\omega L}{2 \cdot Q} - \frac{R_{Coil}}{2} \quad \text{with } \omega = 2\pi \cdot 13.56MHz$$

Neglecting the influence of all the other components on the Q-factor, this calculation only gives an estimation of the later used value of R_S , but this estimation is necessary to do the calculation of the matching capacitors in the next step.

The Q-factor has to be checked and adjusted later on as described in [section 3.7](#) to get an exact value for the external resistor.

⁴ Although I-Code basically uses a higher Q-factor, the Micore requires a Q < 40. Micore is limited to the proximity range. For ISO/IEC 14443 higher bit rates even a lower Q is required.

3.5 The parallel and serial capacitors

With these values of the coil L_a , R (including the external resistor) and the required impedance Z_{ant} , now the parallel capacitor C_2 and serial capacitor C_1 can be calculated:

$$C_{2a} = C_{2b} = \frac{1}{\omega \cdot \sqrt{\left(\frac{\omega L}{1 - R/Z_a} \right)^2 - \frac{R^2 + \omega^2 L^2}{1 - R/Z_a} + \frac{\omega^2 L}{1 - R/Z_a}}}$$

$$C_{1a} = C_{1b} = \frac{\frac{R^2 + (\omega L - 1/\omega C_2)^2}{\omega L (1/\omega C_2 - \omega L) - R^2}}{C_2} \quad \text{with } Z_a = 250\Omega$$

Although the estimated external resistor may vary from the really needed one, the calculated capacitor values should be taken as start values for the following tuning procedure.

Remark: There is an Excel-Sheet available that performs this calculation.

Designing based on the given parameters, the value of $Z = 2 \cdot Z_{ant} = 500\Omega \cdot e^{0^\circ}$ may be increased up to $Z = 2 \cdot Z_{ant} = 800\Omega \cdot e^{0^\circ}$ to increase the output power (details see section 4), but of course the limit of the output current of Micore must not be exceeded!

3.6 Tuning procedure

With the given and calculated values for L , R_{Sa} , R_{Sb} , C_{1a} , C_{1b} , C_{2a} and C_{2b} the complete antenna circuit has to be tuned to equalize measurement and calculation uncertainty and tolerances. Fig 8 shows the tuning procedure. C1 and C2 each mean both the corresponding symmetrical values. The use of an impedance analyzer is recommended, but there is also an easy method described in the [appendix 6.2](#) to tune the antenna circuit (without impedance analyzer) using only an oscilloscope and a signal generator.

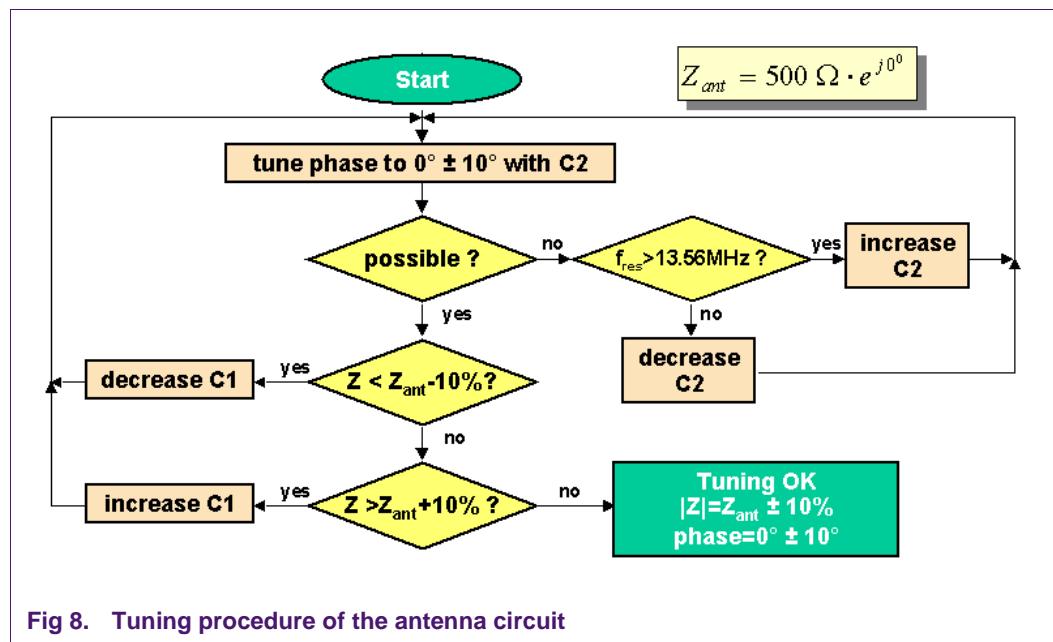


Fig 8. Tuning procedure of the antenna circuit

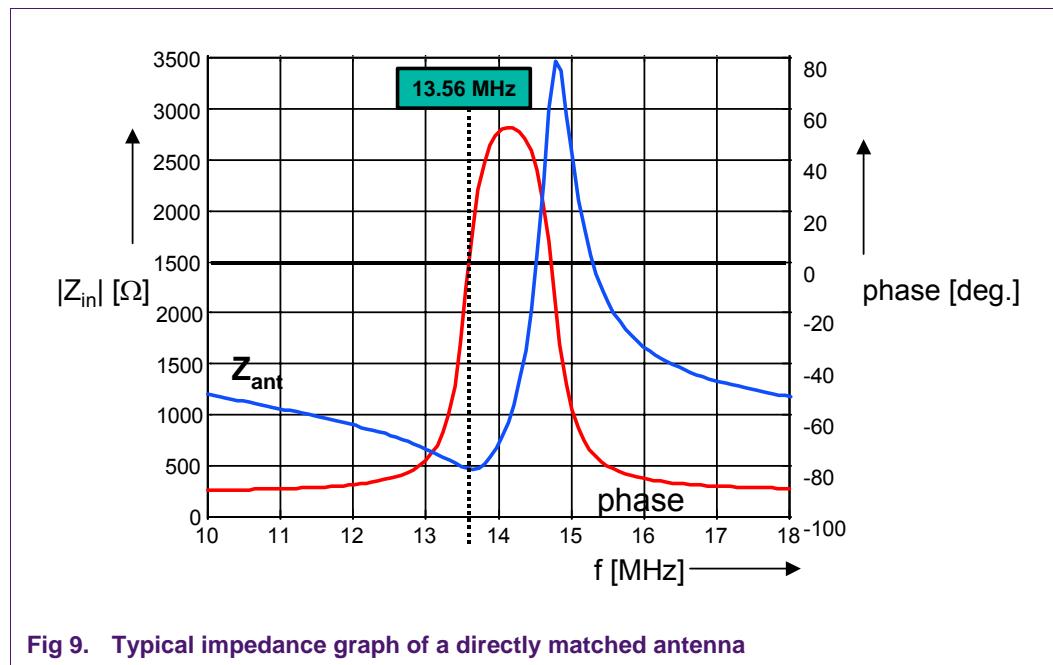


Fig 9. Typical impedance graph of a directly matched antenna

Remark: It is useful to start the tuning procedure with smaller capacitance values than calculated. On one hand measuring the antenna coil disregards stray capacitance also influencing the tuning. On the other hand it is much easier to increase capacitance (by adding extra capacitors) during the tuning procedure than reducing the capacitance.

The Excel sheet can also be used to check and “get a feeling” for the relation between capacitor value changes and the impedance changes. A few pF may change the whole matching!

3.7 Checking the Q-factor and output current

As the Q-factor has a direct influence on the edges of the modulation shape, this should be used to check the Q-factor.

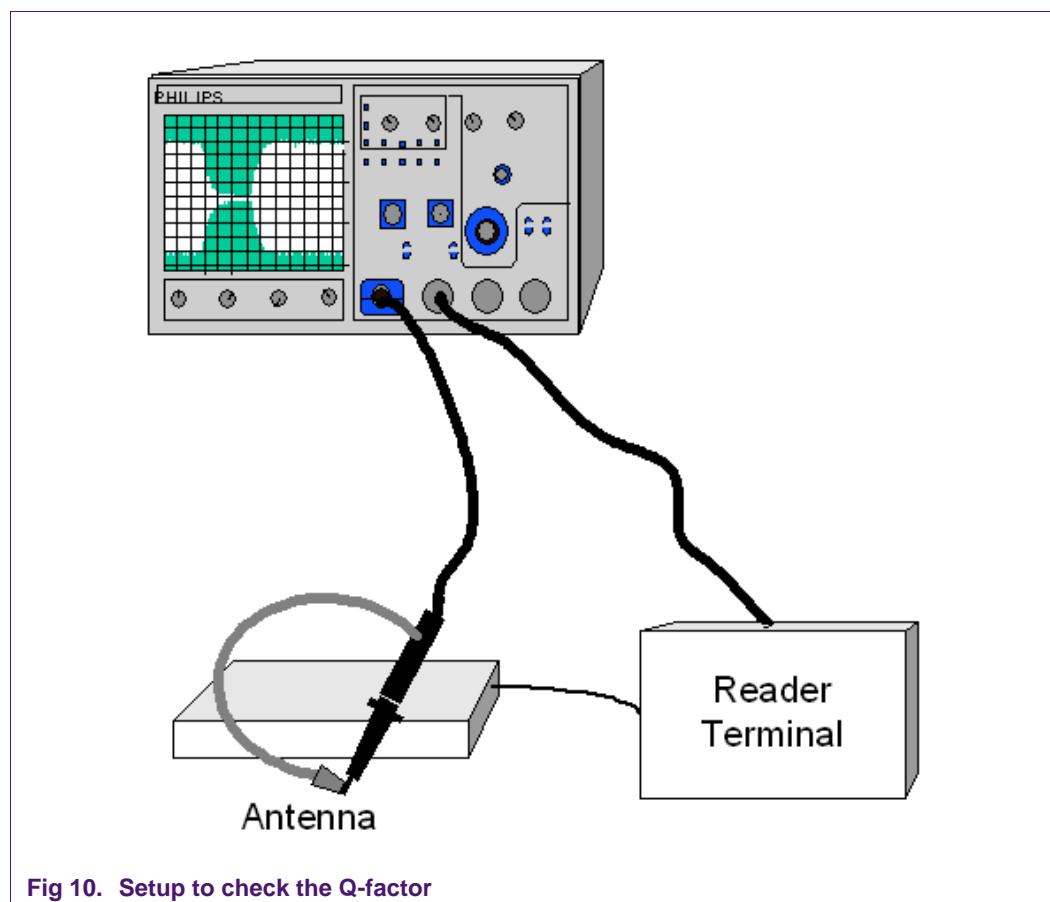


Fig 10. Setup to check the Q-factor

An oscilloscope with a bandwidth of at least 50 MHz shall be used and two probes shall be connected as shown in [Fig 10](#):

CH1: Form a loop with the ground line at the probe to enable inductive signal coupling.
Hold the probe loop closely above the antenna.

CH2: Connect probe to the MFout signal at Pin 4 of the reader IC, Trigger source = CH2.
The MFoutSelect register (26h) has to be set to
“2” (Modulation Signal (envelope) from internal coder, Miller coded)
or “3” (Serial data stream, not Miller coded)

For further details see the related Micore datasheets.

An example of these test signals is shown in [Fig 11.](#)

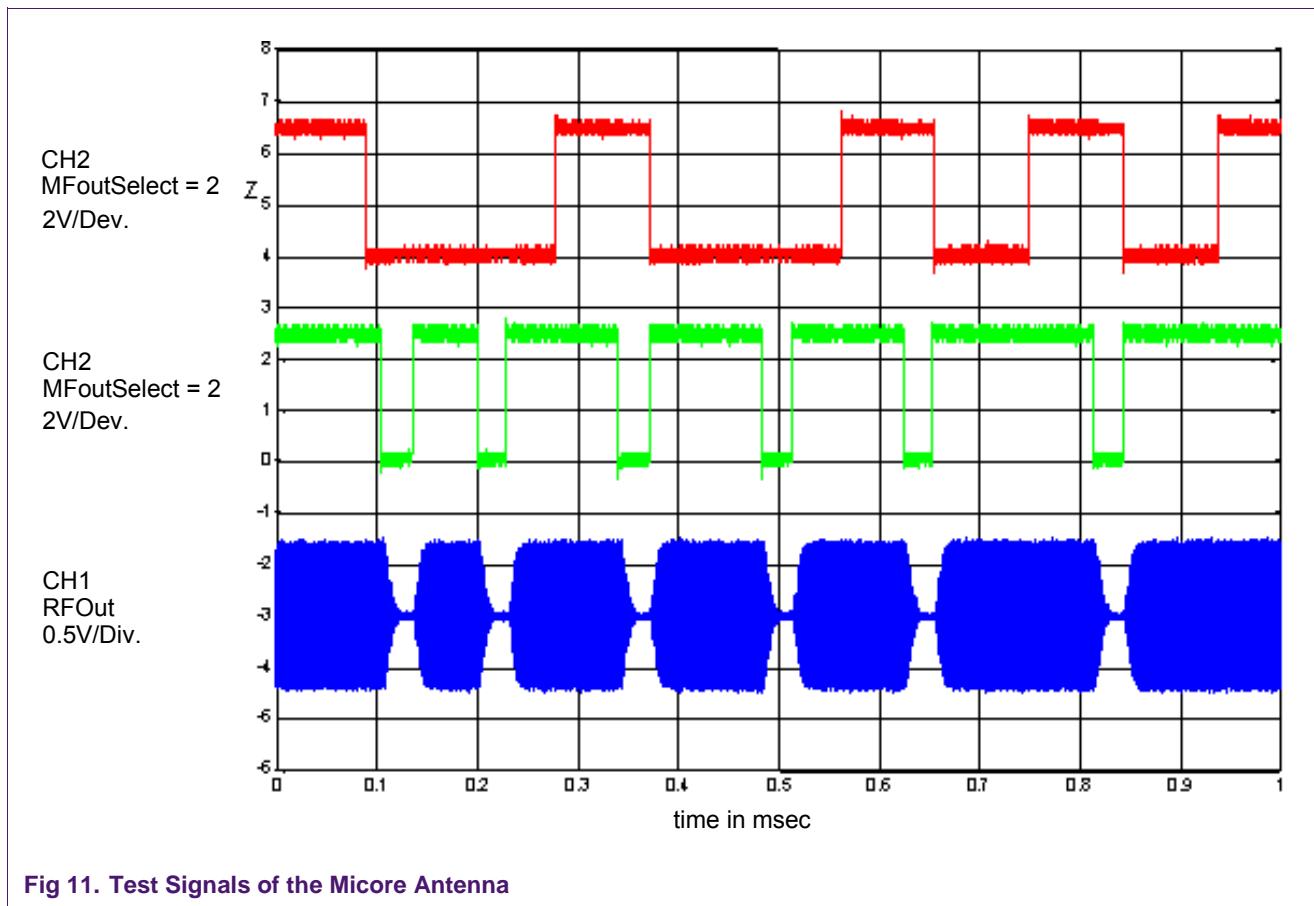


Fig 11. Test Signals of the Micore Antenna

It is recommended to check the pulse shape and compare the scope plot to [Fig 12](#). The related values are given in [Table 4](#).

Remark:

The absolute measured voltage in CH1 depends on the coupling (= distance) between the probe loop and the reader antenna.

The influence of the coupling on the shape can be neglected.

The complete antenna tuning and Q-checking is done without any card (unloaded). However, the complete PCD has to be checked against the ISO/IEC14443 (see ref [8]) using the reference PICCs acc. ISO10373-6 (see ref [9]). That requires e.g. to check the pulse shapes unloaded and under load conditions (using the reference PICC as defined in ref [9]).

3.7.1 Pulse shape according to ISO14443A

For the antenna design for the MF RC500, the MF RC530, the MF RC531 and the CL RC632 the pulse shape (Q-Factor) shall be checked according to the ISO14443A. For the SL RC400 the shape is described in [section 3.7.2](#)

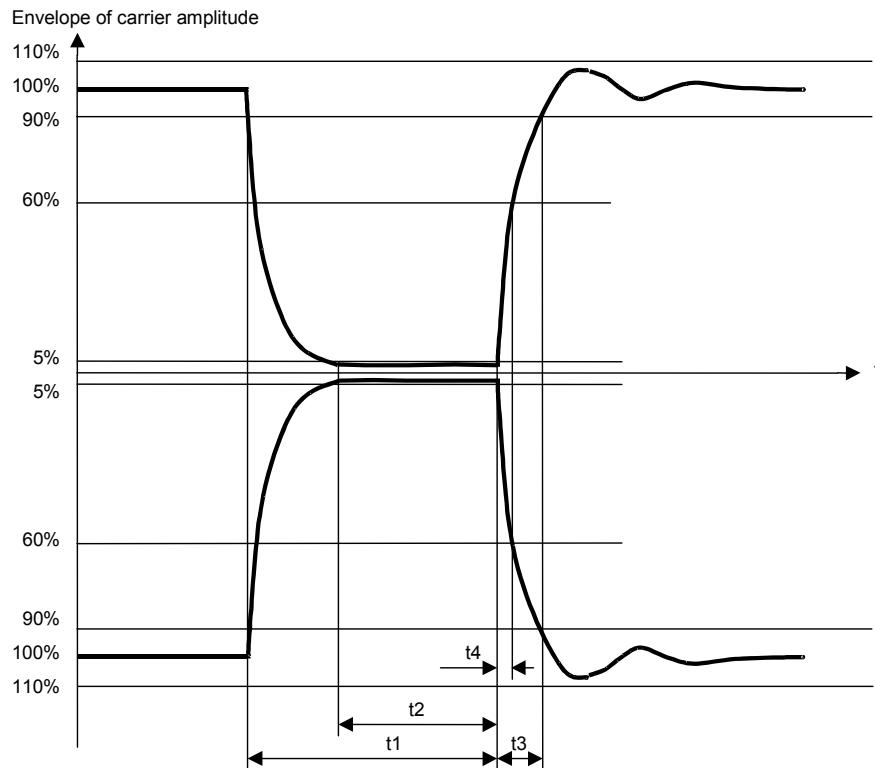


Fig 12. Pulse shape according to ISO/IEC 14443A

Table 4: Pulse duration in [μs] compliant with ISO/IEC 14443A

Pulses length	t_1	$t_2 \text{ min}$	$t_3 \text{ max}$	$t_4 \text{ max}$
T1 MAX	3.0	0.7	1.0	0.4
T1 MIN	2.0	0.7	1.0	0.4

The time t_1-t_2 describes the time span, in which the signal falls from 90% down below 5 % of the signal amplitude. As the pulse length of Micore is accurate enough, only the times t_2 and t_4 have to be checked: the signal has to remain below 5% for the time t_2 .

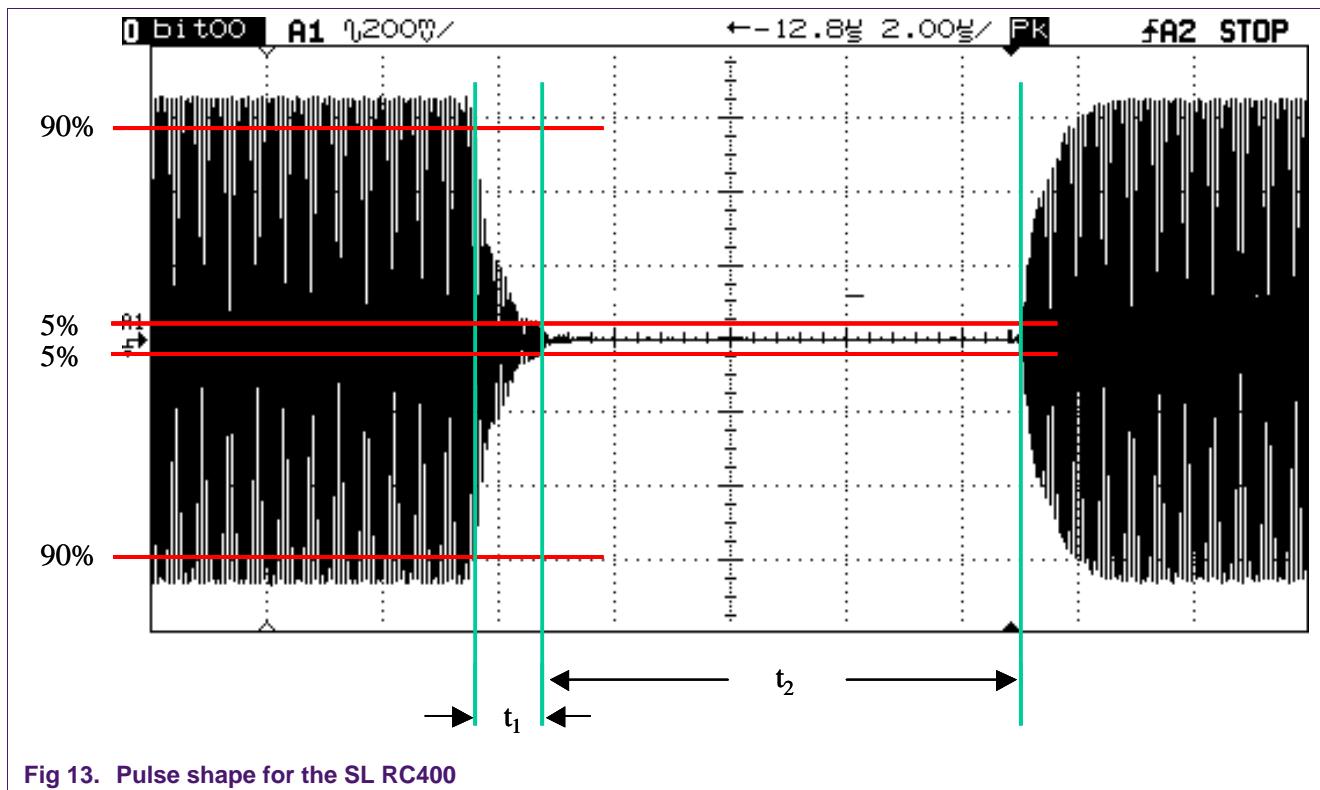
To guarantee a correct antenna tuning and Q-factor the following shall be checked:

- I) The signal has to fall below the 5 % value.
- II) The time t_2 shall be in the limit: $0.7 \mu\text{s} < t_2 < 1.4 \mu\text{s}$

If $t_2 < 0.7 \mu\text{s}$, the Q-factor is too high ($Q > 35$). R_{ext} has to be increased.

If $t_2 > 1.4 \mu\text{s}$, the Q-factor is too low and the operating distance will be dissatisfying. R_{ext} has to be decreased.

3.7.2 Pulse shape for the SL RC400 design



The I-Code pulse shall be switched to 100% ASK to check the Q-factor as shown in [Fig 13](#). The time t_1 describes the time span, in which the signal falls from 90% down below 5% of the signal amplitude. As the pulse length of Micore is accurate enough, only the time t_2 has to be checked: the signal has to remain below 5% for the time t_2 .

Table 5: Pulse duration for the SL RC400

Pulses length	t_2 [μs]
T MAX	8.7
T MIN	7.2

To guarantee a correct antenna tuning and Q-factor the following shall be checked:

- I) The signal has to fall below the 5 % value.
- II) The time t_2 shall be in the limit: $7.2\mu s < t_2 < 8.7\mu s$

If $t_2 < 7.2\mu s$, the Q-factor is too high ($Q > 35$). R_{ext} has to be increased.

If $t_2 > 8.7\mu s$, the Q-factor is too low and the operating distance will be dissatisfying. R_{ext} has to be decreased.

3.8 Receiving circuitry

When all the transmit design issues ([sections 3.2 to 3.7](#)) have been taken care of, the reader antenna radiates the maximum possible magnetic field and correctly transmits the TX-data according to the specified protocol and coding. Now the receive circuitry has to be connected and adjusted.

The antenna circuitry should be assembled with the components as given in [section 3.1](#):

$$C_3 = 1nF \quad (\text{Ceramic NP0, tolerance } \leq \pm 10\%)$$

$$C_4 = 100nF \quad (\text{Ceramic X7R, tolerance } \leq \pm 10\%)$$

$$R_1 = 470\Omega \dots 4.7k\Omega \quad (\text{For determining the exact value: see below})$$

$$R_2 = 820\Omega$$

As the matching of the antenna (TX-way) provides a maximum of power coupled into the antenna depending on its impedance, the voltage at the antenna (at node T1 as given in [Fig.3](#)) is slightly different from antenna coil to antenna coil. The Rx input pin (PIN29) of the Micore is high-impedance, so a voltage is coupled back into the Rx-input of the Micore.

So two rules have to be fulfilled:

- I) **DC-voltage level at the Rx input pin has to be kept at Vmid.**
(That is why R2 and C4 are required.)
- II) **AC-voltage level at the Rx input pin has to be kept within the following limit:**
 $1.5 \text{ Vpp} < V_{Rx} < 3 \text{ Vpp}$

If $V_{Rx} > 3 \text{ Vpp}$, R_1 has to be increased.

If $V_{Rx} < 1.5 \text{ Vpp}$, R_1 has to be decreased.

The Rx input voltage shall be checked with and without a card in the field with minimum and maximum operating distance.

Remarks: *Do not exceed the limit of $V_{Rx} = 3Vpp$ AC at the Rx-input pin!!*

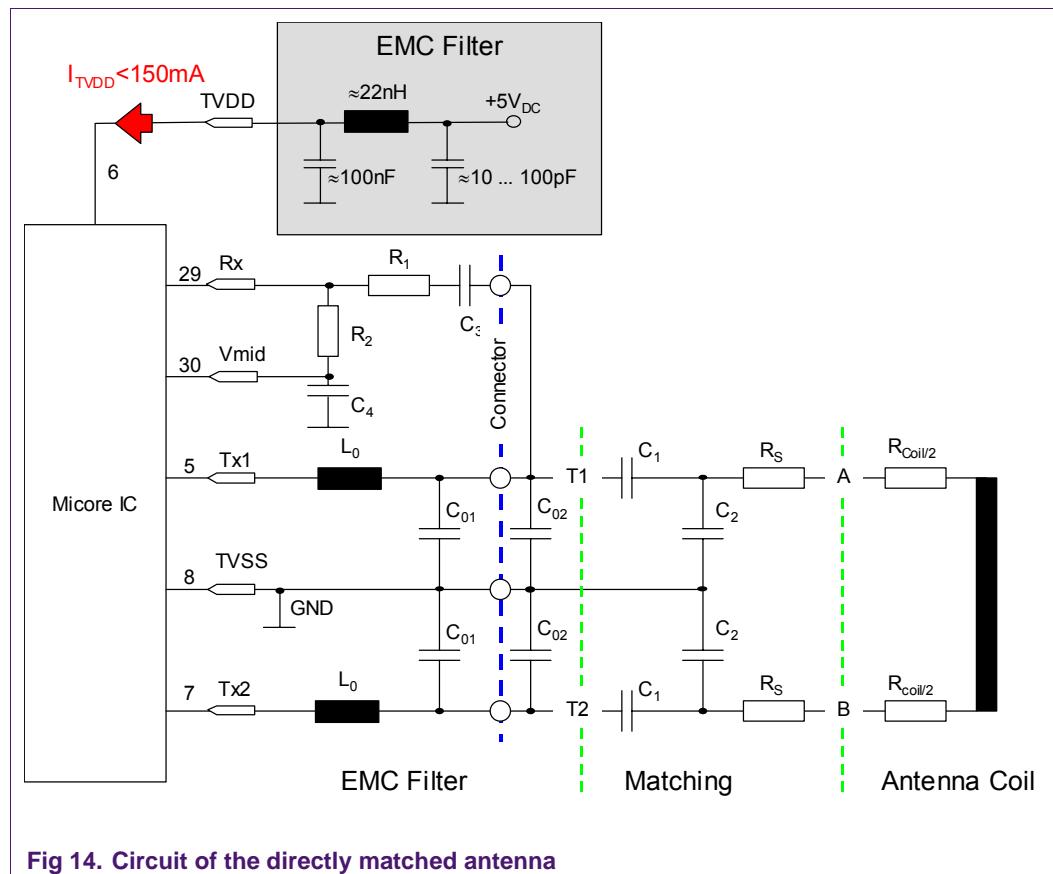
A higher input voltage may not destroy the chip, but results in a receiving failure.

The RX-Input may either be connected to node T1 or T2.

4. Full parameter design

The full parameter design needs some basic knowledge about RF design. Some parameters, that are fixed in [section 3](#), here are free to be changed or adapted to special requirements to achieve more flexibility, but of course the design needs some more design steps and detailed design work. The complete Micore reader antenna design principle is discussed with the same reference hardware as shown in [Fig 2](#).

So the antenna design principle is the same as given in [section 3](#), and the circuit (see [Fig 14](#)) itself is the same as above (compare with [Fig 3](#)). To get improve performance the EMC low pass filter is included into the matching.



4.1 Design requirements

4.1.1 Filtering the supply voltage

Even though it is not required, an EMC filter connected to the TVDD pin as shown in [Fig 14](#) might help to improve the overall performance:

- a) It suppresses noise coming from the supply voltage coupling into the analog part of the antenna circuit, and
- b) it suppresses harmonics coming from the transmitter to be radiated into the environment (and the rest of the circuit).

A similar filter might be used for the AVDD and even DVDD.

4.1.2 Resonance frequency of the EMC filter

In the basic design the resonance frequency of the given EMC low pass filter is approximately 13.56 MHz to make the design easy.

To get a better performance the resonance frequency of the EMC filter itself should be around 14.4MHz (= $f_c + 847.5\text{kHz}$). Proposed values for this EMC filter can be found in [Table 6](#): This should be for two reasons:

- a) It increases the signal to noise ratio for the receive signal, and improves the receive performance.
- b) It decreases the overshoots of the transmit pulses, and improves the signal quality of the transmit signal.

Table 6: Proposed value of EMC low pass filter components

Component	Value
L_0	1 μH (e.g. TDK NL322522T-1R0J)
C_{01}	68pF each (Ceramic NP0, tolerance $\leq \pm 2\%$)
C_{02}	56pF each (Ceramic NP0, tolerance $\leq \pm 2\%$)

Of course this resonance frequency requires the EMC low pass filter to be considered in the matching of the antenna.

4.1.3 Matching requirement

In addition to the general design rules based on the Mifare interface principle, as given in the application note in [ref. \[1\]](#), the basic requirement for the antenna design is reduced to the minimum requirement of the Micore.

The Micore delivers a square signal of

$U_{TxAC} \approx \pm 2.5 \text{ V pp (square)}^5$ with $f_0 = 13.56 \text{ MHz}$ and a maximum output current of $I_{TVDD} \leq 150\text{mA}$

That means, the TX output toggles between $V_L = 0\text{V}$ and $V_H \approx 5\text{V}$ at a frequency of 13.56 MHz. Tx1 and Tx2 usually have a 180 degrees phase shift, depending on the setting of

⁵ Additionally there is a mean DC voltage of $U_{TxDC} = 2.5 \text{ V!}$

TX2Inv (bit 3 of the Tx-Control Register). Please refer also to the related Micore datasheet.

Therefore four main requirements can be specified:

- I) The TX-output current must not exceed the given limit: $I_{TVDD} \leq 150mA$
- II) The harmonics have to be suppressed to meet the regulation rules.⁶
- III) The receive signal has to be coupled back into Rx input of the Micore.

Limiting the radiation of harmonics is not the specific goal of this application note. However, the basic guidance provided herein shows, that an easy design is possible that also meets the general EMC rules. The use of a low pass filter directly connected to the TX outputs is recommended.

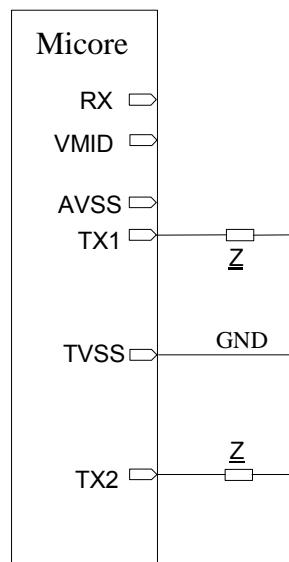
Remark: As mentioned in the beginning of the full parameter design, some knowledge is required to design a Micore reader antenna different from the recommendation in section 3. This includes the knowledge of the EMC behavior at RF outputs. Usually highly efficient RF outputs generate a great number of harmonics, which have to be suppressed sufficiently to meet government restrictions. Due to the limited Q-factor and some parasitic effects of the passive components the whole antenna circuitry might resonate at some frequencies above 100MHz, or behave like a second (unwanted) antenna at a certain frequency above 13.56 MHz. This has to be checked very carefully during the design.

Layout hint: The most critical part of the antenna circuit is the EMC low pass filter, so the component area of this filter shall be as small as possible, and a proper GND connection of this filter shall be directly connected to the TVSS pin.

⁶ EN (Europe) or FCC (USA) EMC regulations, mainly the radiation of electro-magnetic field <1GHz

With the given voltage and current, the first requirement can be formed into the following (see [Fig 15](#)):

- la) The minimum load impedance connected to a TX-output, shall be at least
 $Z = Z_{\text{loadTx}} = 20 \Omega$.⁷



[Fig 15. Micore minimum load requirement](#)

Remark: Of course all the general rules, like maximum power, a correct Q-factor, resonance, antenna size, receive circuit, etc. have to be considered, too.

The mean DC voltage at each TX pin has to be considered, too, but usually a matching capacitor decouples the DC voltage anyway.

Although the TX-output current basically is AC (@13.56 MHz), the specified supply current I_{TVDD} is DC that easily can be measured and checked at the TVDD pin of the Micore continuously during the whole design.

\underline{Z} always means a complex impedance, consisting of resistance and reactance:
 $\underline{Z} = R + jX = Z \cdot e^{j\phi}$

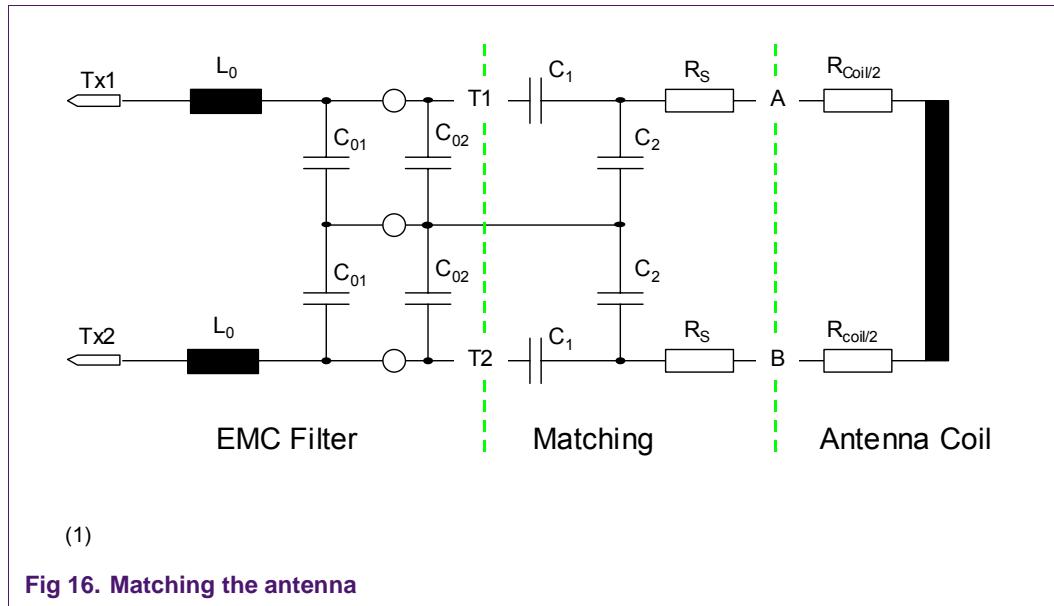
⁷ The impedance and the voltage are referred to GND. Phase = 0, if not specified.

4.2 Required design steps

1. Design a coil, measure L and R or L and Q (see [section 3.2](#)).
2. Calculate the resonance capacitors to design a resonance circuitry together with the coil (see [section 3.3](#) to [3.5](#)).
3. Tune this resonance circuitry together with the EMC low pass filter to the required impedance (see [section 4.3](#))
4. Connect the resonance circuitry to the Micore output, check the I_{TVDD} and if necessary retune the components for optimum performance.
5. Check & adjust the Q-factor (see [section 4.3.1](#)).
6. Check & adjust the receive circuitry (see [section 3.8](#)).

4.3 Impedance Matching & Resonance

The principle of the antenna matching is the same as shown before in [section 3.6](#), but now the EMC low pass filter has to be included into the matching and tuning procedure.



So based on the same design the whole circuit as shown in [Fig 16](#) has to be matched to an impedance of approximately 40Ω between Tx1 and Tx2, using the following values as proposed in [Table 6](#):

4.3.1 Q-factor

In any case of designing a Micore reader antenna, the Q-factor has to be checked. The overall Q-factor of a Micore antenna – supporting higher bit rates – is limited to

$$Q \leq 22$$

and shall be checked in principle as given in [section 3.7](#). This value is valid for mifare® and I-Code® (proximity) designs.

The lower Q factor compared to a standard mifare reader design is related to the pulse shape requirements of higher bit rates according to [\[8\]](#). In addition to that the lower Q-factor increases the overall stability and the robustness against environmental changes.

So in addition to [section 3.7](#) for higher bit rates the relevant pulse shapes shall be checked, too. Refer to the application note [\[2\]](#) for details.

5. Additional design hints

5.1 Antenna functionality

In each of the design steps the three functions of a reader antenna should be considered:

1. **Transmit power:** The radiated magnetic field has to be maximized considering the radiation and datasheet limits, especially the limits for the radiation of the harmonics (up to 1GHz).
2. **Transmit data:** The coded and 10% or 100% ASK modulated data signal has to be transmitted in a way, that every card is able to receive it. The signal shape and timing (i.e. the Q-factor) has to be considered.
3. **Receive data:** The card's answer has to be delivered to the receive input of the Micore considering the datasheet limits.

If one of these functions is not completely provided, the overall function of the antenna is disturbed or at least the performance is reduced. So if a supposed overall performance is not achieved with a specific design, each of these 3 functions shall be checked separately.

5.2 Layout

Even though this document does not replace any relevant RF design documents and it does not cover EMC related topics in detail, some general recommendations can be given to simplify a proper design.

The Micore IC itself drives the 13.56 MHz carrier with a signal, which is almost a square signal. This leads to many harmonics up to the GHz range, which have to be suppressed sufficiently to meet all the relevant EMC regulations⁸. The most critical part of the overall analog layout is the circuits directly connected to the Micore IC: the EMC low pass filter as well as the connection of the supply voltage pin TVDD.

So on one hand an additional EMC filter for the supply voltage might be usefull.

On the other hand the layout of L_0 and C_0 shall be considered carefully. The overall layout and placement area of TX1, TX2, L_0 , C_0 and TVSS shall be kept as small as possible. A proper and short GND connection is required! One proper GND plane is recommended!

A 2-layer board reference layout is shown in [Fig 17](#) and [Fig 18](#) with the corresponding schematic in [Fig 20](#) and [Table 7](#):

8. Like FCC in the USA and CE in Europe.

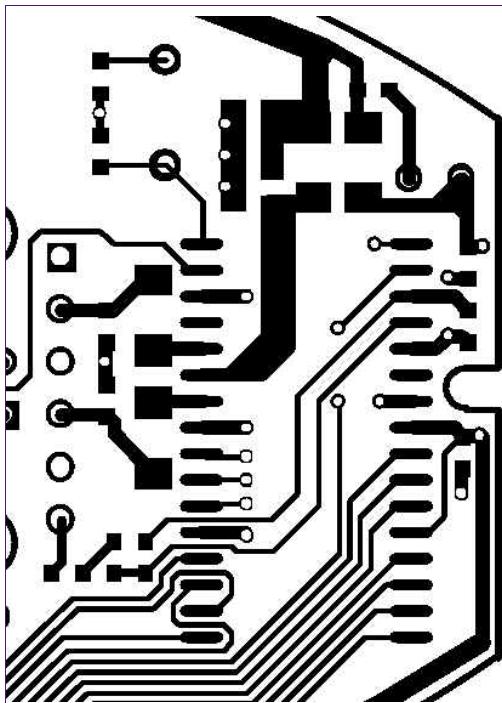


Fig 17. Example layout, top layer

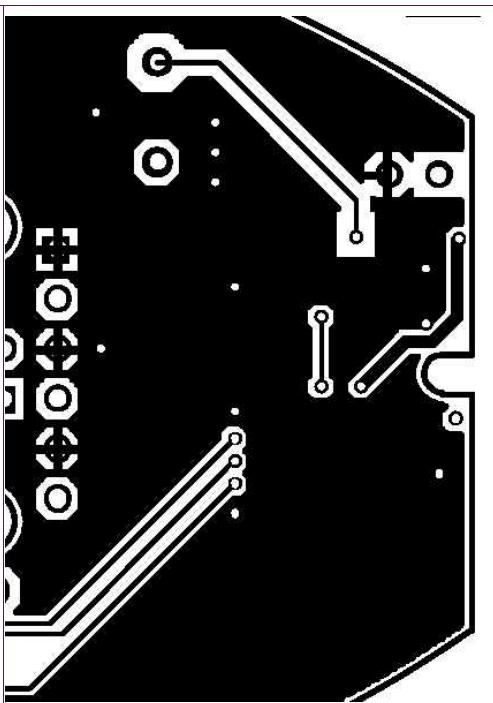


Fig 18. Example layout, bottom layer (mirror view)

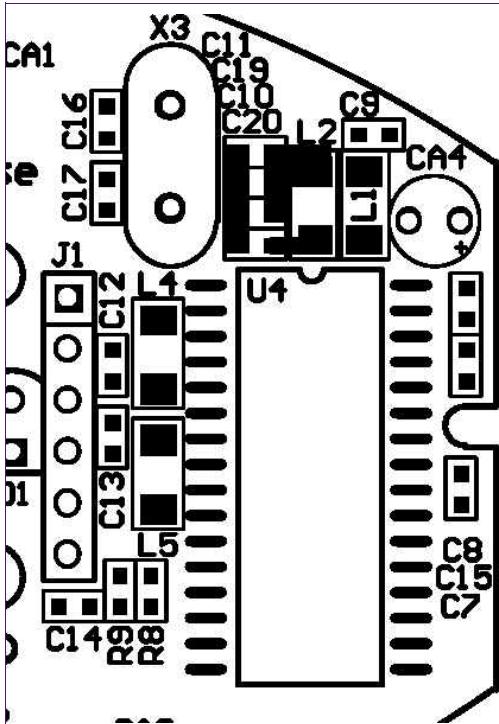


Fig 19. Example layout, placement

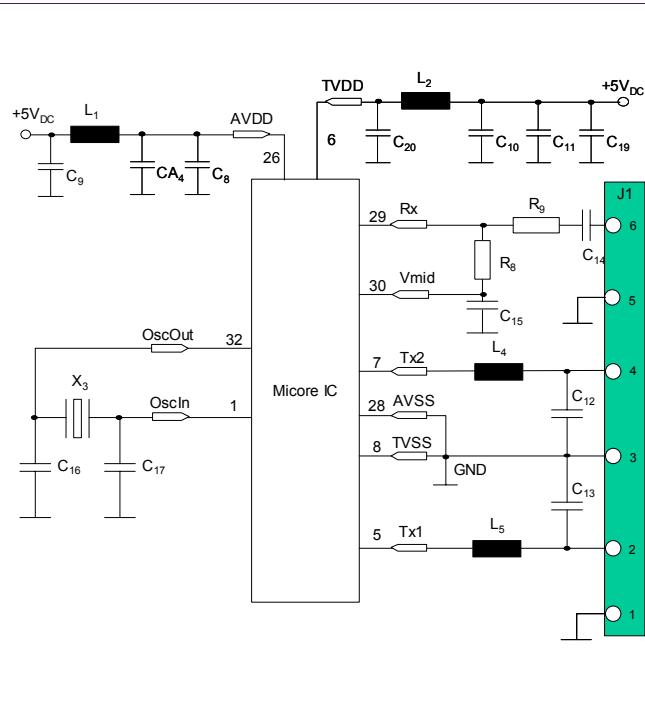


Fig 20. Example schematic

Table 7: Bill of Material

Part	Value	Remark
X ₃	13.56MHz	X-Tal
L ₁ , L ₂	22nH	Or 0Ω Jumper
L ₄ , L ₅	1 μH	Shielded, +/-5% tolerance
C ₁₂ , C ₁₃	68pF	NPO, +/-2% tolerance
C ₈ , C ₉ , C ₁₅ , C ₁₉	100nF	X7R
C ₁₄	1nF	NPO
C ₂₀	10pF	NPO
C ₁₀	100pF	NPO
C ₁₁	n.a.	
C ₁₆ , C ₁₇	15pF	NPO
CA ₄	10μF	
R ₈	820Ω	
R ₉	2.2kΩ	

6. Appendices

6.1 General Checklist for Micore antenna design

Table 8: Checklist

Parameter	Requirement	Comment	Check
Power	maximum	maximum operating distance	
I_{TVDD}	$I_{TVDD} < 150\text{mA DC}$	detuning has to be considered!	
U_{rxpin}	$1.5\text{Vpp} < U_{rxpin} < 3\text{Vpp}$	measured at pin 29	
Quality factor	$Q < 30$	check with oscilloscope	
Temperature influence	operating range	Min & max operating distance	
Temperature influence	$I_{TVDD} < 150\text{mA DC}$		
Environmental changes	operating range	Min & max operating distance	
Environmental changes	$I_{TVDD} < 150\text{mA DC}$		

6.2 Simple method for impedance measurement

If no impedance analyzer is available, the impedance measurement and tuning could be done with the following simple method.

The test setup consists of:

1. Signal generator (13.56 MHz)
2. Oscilloscope with low impedance probe
3. Measurement circuit as shown in [Fig 21](#)

The two probes of the oscilloscope ($C_{x\text{probe}}$ $C_{y\text{probe}}$) are connected to the function generator output and in parallel to the reference resistor. The oscilloscope displays a Lissajous figure, allowing us to derive the absolute magnitude and the phase. The magnitude is given by the angle of the Lissajous figure and the area as depicted in the figure below gives the phase.

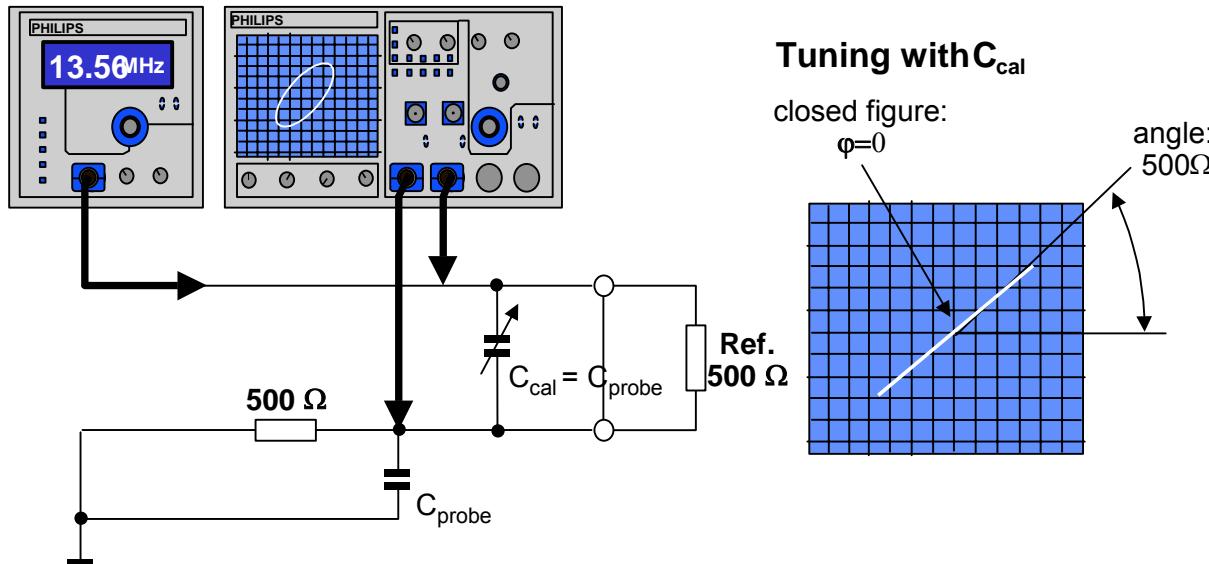


Fig 21. Simple impedance measurement, calibration

The x-probe capacitance $C_{x\text{probe}}$ only reduces the amplitude at the function generator output. This has no influence on the tuning results.

The y-probe capacitance $C_{y\text{probe}}$ affects a phase shift, which changes the area of the Lissajous figure. To compensate this effect, the capacitor C_{cal} is connected in parallel to the matching network.

The tuning procedure has to be done in two steps:

Step 1: Calibration

For the calibration a reference resistor of 500Ω has to be inserted instead of the antenna.

The calibration procedure is depicted in [Fig 21](#). The function generator shall be set to:

Table 9: Settings of function generator for calibration

Parameter	Value
Wave form:	Sinusoidal
Frequency:	13.56 MHz
Amplitude:	2V - 5V

The calibration capacitor has to be adjusted until the Lissajous figure is completely closed (phase = 0°). Then the calibration capacitance C_{cal} is equal to the capacitance C_y probe. The y-probe voltage is in phase and the amplitude is exactly half of the function generator voltage (x-probe).

Remark: If the scale for the x-probe is chosen twice the scale for the y-probe (e.g. x-scale: 2V/DIV and y-scale: 1V/DIV) the Lissajous figure angle shall be 45 degree.

A loop of the ground cable of the probe shall be avoided to minimize inductive coupling from the antenna. The use of a low capacitance, high frequency probe is recommended.

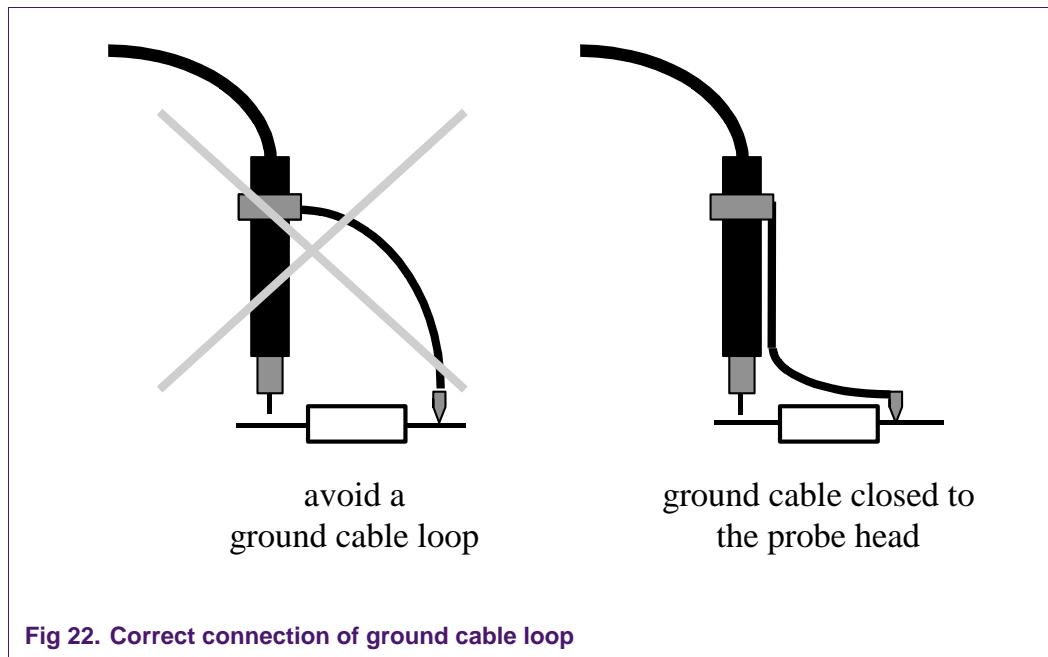
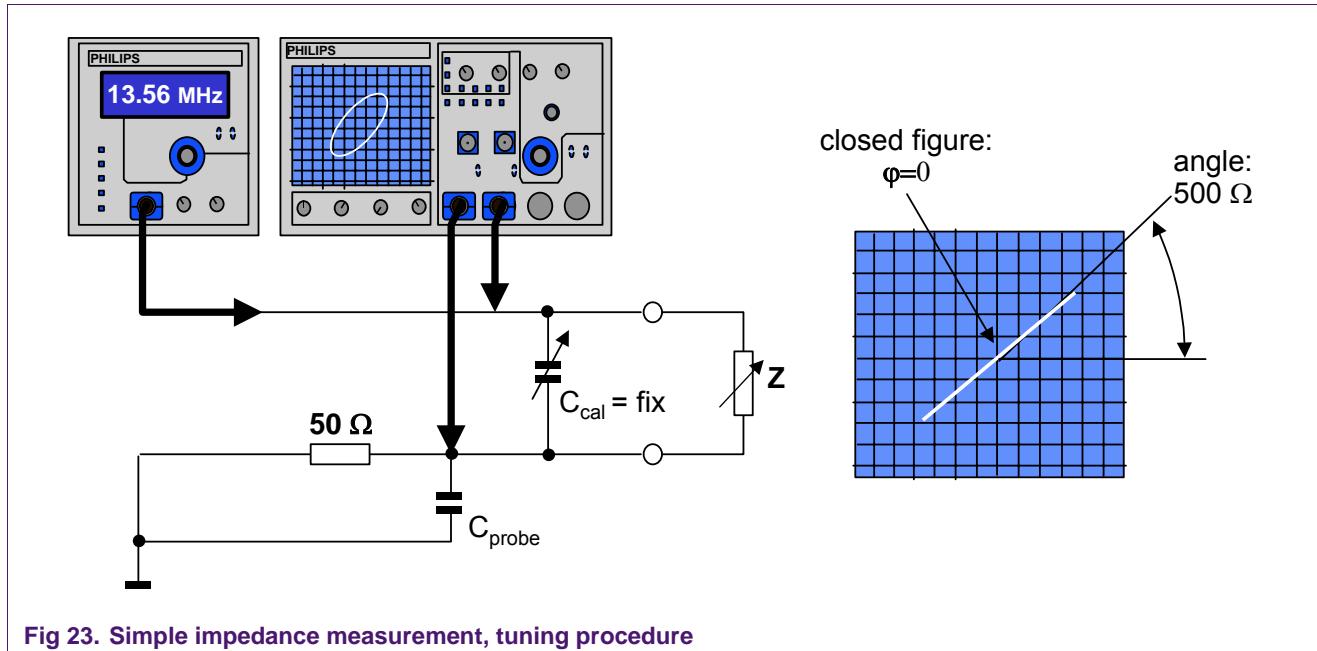


Fig 22. Correct connection of ground cable loop

Step 2: Tuning procedure:

After the calibration, the reference resistor has to be replaced by the antenna circuit (Z) as shown in [Fig 23](#). The matching network shall be tuned by the (variable) capacitors C_1 and C_2 until the Lissajou figure is completely closed. Now the Lissajou figure angle has to equal to the angle of the calibration step. In this case the impedance of the tuned antenna has $Z = 2 \cdot Z_{ant} = 500\Omega \cdot e^{0^\circ}$.



Notes to interpret the Lissajou figures:

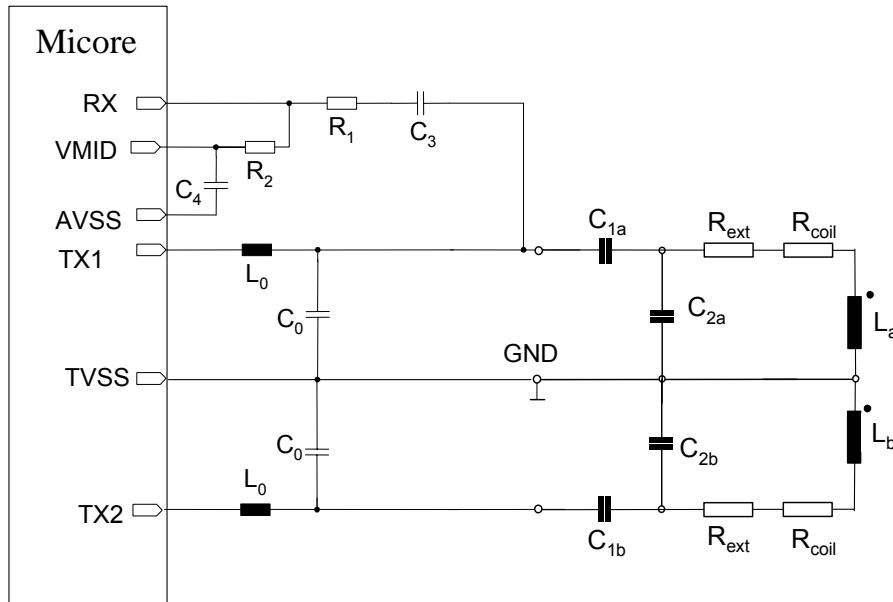
- If the figure is not closed the phase between x and y is unequal to zero.
- If the angle $\phi=0^\circ$, the Lissajou figure is closed completely.
- If the angle is greater than 45° , Z is greater than 500Ω .
- If the angle is smaller than 45° , Z is greater than 500Ω

Remark: This calibration and tuning procedure principally may be done with any required impedance value. In praxis the environmental influences have to be considered. Therefore this method typically is limited to impedance values $< 1k\Omega$.

The impedance curve of an antenna (as shown in [Fig 9](#)) has two points of resonance, where the phase is 0° . It is only possible to tune the lower one of both these resonance frequencies (f_{LOW}) to the required $Z = 2 \cdot Z_{ant} = 500\Omega \cdot e^{0^\circ}$.

To be sure that the tuning is done to the lower resonance frequency, it is recommended to reduce the calculated value for C_1 and C_2 by 40% and add tuning capacitors in that range: Start the tuning with the lowest values for the tuning capacitors.

6.3 All formulas on one page



$$L_0 = 1\mu H$$

$$C_3 = 15 \text{ pF}$$

$$C_4 = 100 \text{ nF}$$

$$R_1 = 470\Omega \dots 2.7k\Omega$$

$$R_2 = 820\Omega$$

$$Z_{ant} = Z = 500\Omega = (500 + j0)\Omega = 500\Omega \cdot e^{j0^\circ}$$

Fig 24. Schematic diagram and component values

Measured Coil: $L = L_a + L_b \quad R_L = 2 \cdot R_{coil}$

Symmetry: $L_a = L_b \quad Z_a = Z_b = \frac{Z_{ant}}{2}$

External resistor: $R_{ext} = \frac{1}{2} \cdot (R_L - R) = \frac{\omega L}{2 \cdot Q} - R_{coil} \quad \text{with } \omega = 2\pi \cdot 13.56 \text{ MHz}$

Parallel capacitor: $C_{2a} = C_{2b} = \frac{1}{\omega \cdot \sqrt{\left(\frac{\omega L}{1 - R/Z_a}\right)^2 - \frac{R^2 + \omega^2 L^2}{1 - R/Z_a} + \frac{\omega^2 L}{1 - R/Z_a}}}$

Serial capacitor: $C_{1a} = C_{1b} = \frac{R^2 + \left(\omega L - \frac{1}{\omega C_2}\right)^2}{\omega L \left(\frac{1}{\omega C_2} - \omega L\right) - \frac{R^2}{C_2}} \quad \text{with } Z_a = 250\Omega$

6.4 Antenna design example

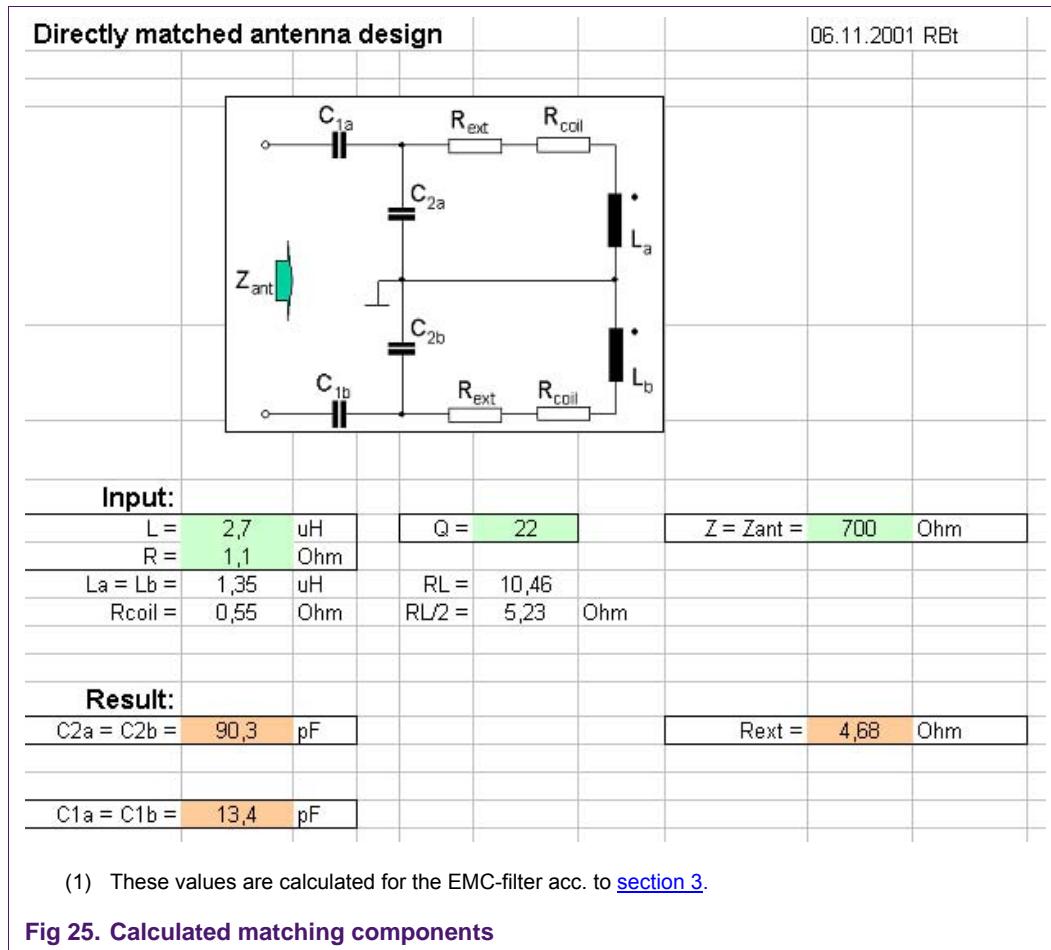
In the following pages an antenna examples is shown. First of all the antenna coil is designed. In this example a two turns antenna with a rectangular size of 160mm x 180mm is used (one turn per each L_a and L_b). This coil is measured with an impedance analyzer.

Table 10: Example antenna coil

Inductance and resistance measurement, using an impedance analyzer

	L_s and R_s	L_s , R_s , and C_p (resonance equivalent circuit)
L	2.7 μH	2.5 μH
R	1.1 Ω	0.95 Ω
C	-	4pF

With these values the matching components can be calculated, using the spreadsheet:



Based on these values the tuning procedure is started, including the EMC low pass filter as proposed in [section 4.1.2](#). The resulting circuit is shown in [Fig 26](#), the component values are shown in [Table 11](#); and the matching result is shown in [Fig 27](#).

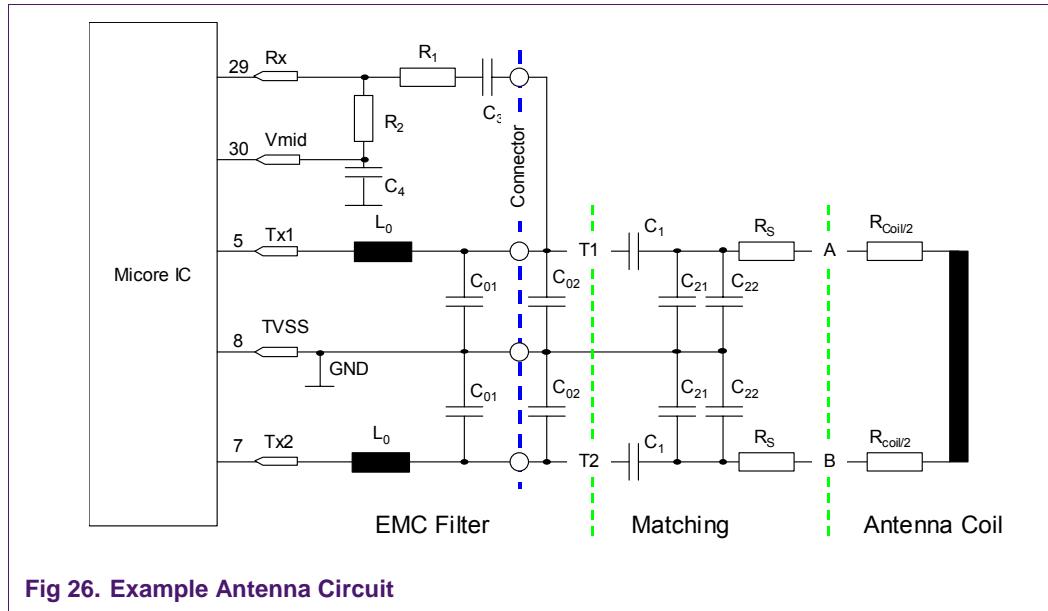
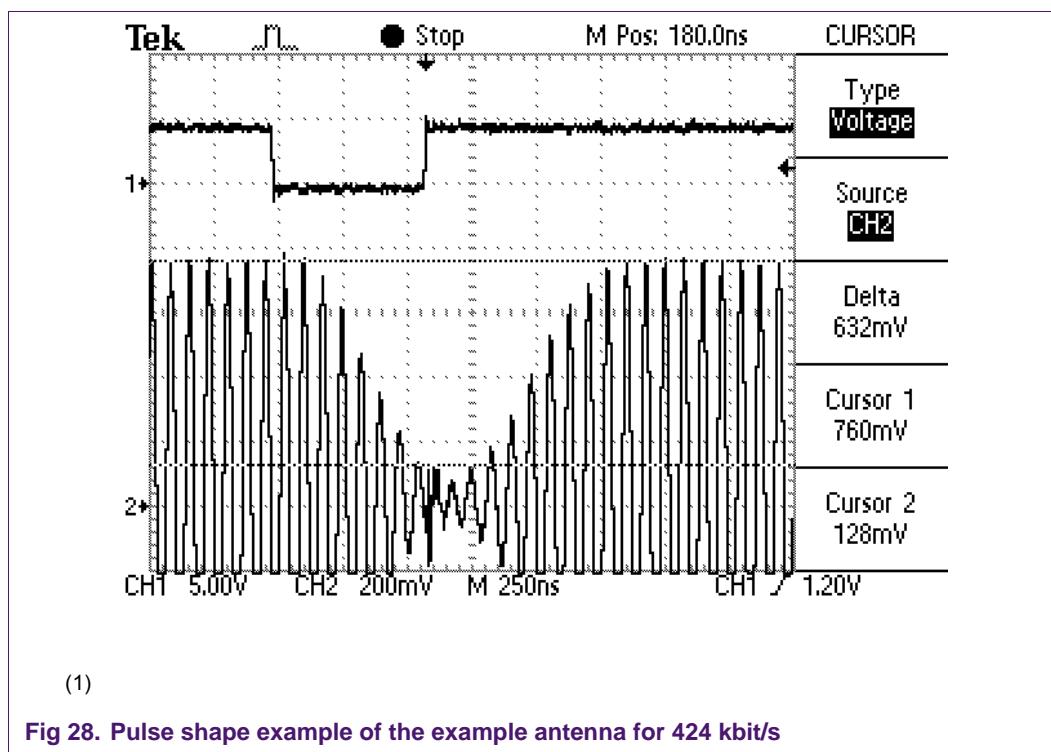
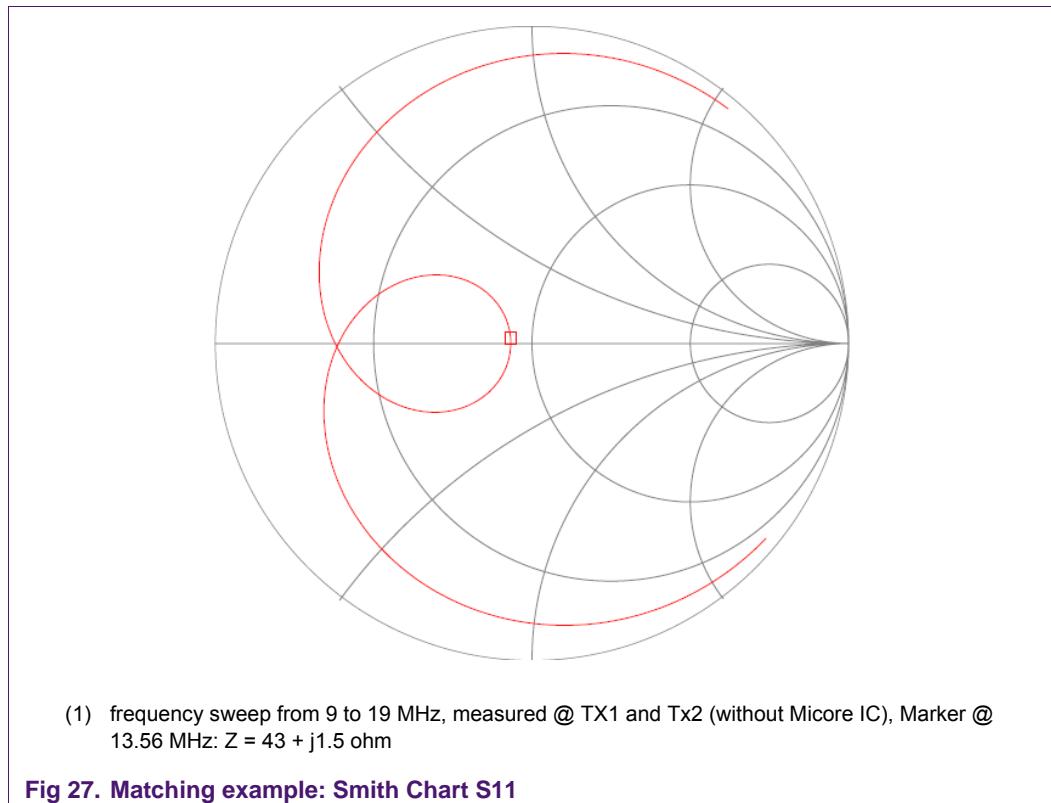


Table 11: Resulting component values of the example antenna

Component	Value
L_0	1uH (e.g. TDK NL322522T-1R0J)
C_{01}	68pF each (Ceramic NP0, tolerance $\leq \pm 2\%$)
C_{02}	56pF each (Ceramic NP0, tolerance $\leq \pm 2\%$)
C_1	12pF each (Ceramic NP0, tolerance $\leq \pm 2\%$)
C_{21}	82pF each (Ceramic NP0, tolerance $\leq \pm 2\%$)
C_{22}	6.8pF each (Ceramic NP0, tolerance $\leq \pm 2\%$)
C_3	1nF (Ceramic NP0, tolerance $\leq \pm 10\%$)
C_4	100nF (Ceramic X7R, tolerance $\leq \pm 10\%$)
R_1	2.2 k Ω
R_2	820 Ω
R_S	4.7 Ω



This antenna meets the Q-factor requirements and fulfills the pulse shapes requirements according to [8], and with a driving current of $I_{TVDD} = 112\text{mA}$ and a receive voltage of $U_{RxAC} = 2.9\text{Vpp}$ at the receive pin an operating distance of approximately 10cm for a typical mifare card can be measured:

- >12cm (DESfire, $f_{res} = 14.6\text{MHz}$)
- >10cm (Mifare 1K, $f_{res} = 16.5 - 17\text{MHz}$)
- >10cm (2x Mifare 1K, each $f_{res} = 16.5 - 17\text{MHz}$)

7. Abbreviations

Table 12: Abbreviations

Acronym	Description
ATQA	Answer To reQuest type A
ATS	Answer To Select
BCC	Block Check Character (checksum)
CBC	Cipher-Block Chaining
f_c	Carrier frequency (13.56 MHz)
f_{res}	Resonance frequency
FSCI	Frame Size for proximity Card Integer
PCD	Proximity Coupling Device (ISO/IEC 14443 term for reader/writer unit)
PICC	Proximity Integrated Circuit Card
PPS	Protocol and Parameter Selection
RATS	Request for Answer To Select
REQA	REQuest type A
RFU	Reserved for Future Use
SAK	Select AcKnowledge
UID	Unique IDentification number

8. References

- [1] Mifare®(14443A) 13,56 MHz RFID Proximity Antennas; Application Note
- [2] ISO/IEC 14443 higher bit rates with Micore; Application Note
- [3] Data Sheet; SL RC400 I Code Reader IC
- [4] MIFARE® MF RC500; Highly Integrated ISO 14443A Reader IC
- [5] MIFARE® MF RC 530 ISO14443A reader IC
- [6] MIFARE® MF RC531; ISO 14443 Reader IC
- [7] MIFARE® and I Code CL RC632 Multiple protocol contactless reader IC
- [8] ISO/IEC14443 Identification cards - Contactless integrated circuit(s) cards - Proximity cards
- [9] ISO10373-6 Identification cards — Test methods part 6: Proximity cards

9. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products

are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

10. Trademarks

ICode — is a trademark of Koninklijke Philips Electronics N.V.

Mifare — is a trademark of Koninklijke Philips Electronics N.V.

11. Contents

1.	Introduction	3
1.1	How to use this document.....	3
2.	Micore antenna principle	4
3.	Basic parameter design for 106kbit/s.....	6
3.1	Given parameters	7
3.2	The antenna coil	9
3.3	Simplification due to symmetry	11
3.4	The external resistor	12
3.5	The parallel and serial capacitors	13
3.6	Tuning procedure.....	14
3.7	Checking the Q-factor and output current	16
3.7.1	Pulse shape according to ISO14443A	18
3.7.2	Pulse shape for the SL RC400 design	19
3.8	Receiving circuitry.....	20
4.	Full parameter design	21
4.1	Design requirements	22
4.1.1	Filtering the supply voltage	22
4.1.2	Resonance frequency of the EMC filter	22
4.1.3	Matching requirement	22
4.2	Required design steps	25
4.3	Impedance Matching & Resonance	26
4.3.1	Q-factor.....	26
5.	Additional design hints.....	27
5.1	Antenna functionality.....	27
5.2	Layout	27
6.	Appendices	30
6.1	General Checklist for Micore antenna design ...	30
6.2	Simple method for impedance measurement....	31
6.3	All formulas on one page	34
6.4	Antenna design example	35
7.	Abbreviations	39
8.	References	40
9.	Disclaimers	41
10.	Trademarks	41
11.	Contents.....	42



© Koninklijke Philips Electronics N.V. 2006

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Date of release:10. May 2006
Document number:

Published in Austria

APPLICATION NOTE

mifare[®] (14443A)
13.56 MHz RFID Proximity Antennas

November 2002

Revision 1.0

PUBLIC

**Philips
Semiconductors**



PHILIPS

13.56 MHz RFID Proximity Antennas**mifare® (14443A)**

CONTENTS

1	INTRODUCTION.....	3
1.1	Purpose and Scope.....	3
1.2	Abbreviations	3
1.3	Reference Documents.....	3
2	HOW TO USE THIS DOCUMENT.....	4
3	THE MIFARE® RF INTERFACE	5
3.1	Energy Transmission.....	6
3.2	Data Transmission PCD → PICC	7
3.3	Data Transmission PICC → PCD	9
4	THE PCD ANTENNA	11
4.1	The PCD Antenna size	11
4.2	Environmental Influences.....	13
4.3	Antenna Shielding and Compensation	14
5	ANTENNA DECISION GUIDE	20
6	ANNEX	22
6.1	Theory for estimating the optimum PCD antenna radius	22
6.2	Parameters and Units.....	24
CONTACT INFORMATION.....		26

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

1 INTRODUCTION

1.1 Purpose and Scope

This application note is intended to support RF-related design-in of Micore reader ICs. The aim is to provide the required understanding of the MIFARE® RF interface (ISO 14443A) to design application specific antennas and matching circuits to achieve the best performance for a communication with a contactless MIFARE® card. This paper shall give a background on the system's RF part and an overview on the used antenna designs.

1.2 Abbreviations

ASK	Amplitude Shift Keying
OOK	On-Off Keying
UID	Unique IDentifier
RF	radio frequency
Micore	contactless reader IC family including : MF RC500 MF RC530 MF RC 531 SL RC400 CL RC632
PCD	13.56MHz Proximity Reader (Proximity Coupling Device according to the ISO14443)
PICC	MIFARE® Proximity Card

1.3 Reference Documents

- 1) "MIFARE® and I Code, Micore Directly Matched Antenna Design"
- 2) "Data Sheet; SL RC400 I Code Reader IC"
- 3) "MIFARE® MF RC500; Highly Integrated ISO 14443A Reader IC"
- 4) "MIFARE® MF RC 530 ISO14443A reader IC"
- 5) "MIFARE® MF RC531; ISO 14443 Reader IC"
- 6) "MIFARE® and I Code CL RC632 Multiple protocol contactless reader IC"

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

2 HOW TO USE THIS DOCUMENT

The chapter 3 of this document shows the basic principles of the RF interface as it used with MIFARE®. The general understanding of this principle helps to do a fast and reliable antenna design based on Micore reader ICs, but none of the given formulas (especially those in the annex) are necessary to design such an antenna.

The chapter 4 describes some basic facts about the PCD antenna, its optimum size, and how to minimize environmental influences. The given rules and design recommendations should generally be considered for a PCD antenna design. This is true for a Micore based antenna design, too.

The chapter 5 gives an overview and comparison of the two antenna design principles for Micore antennas. A help is given to find the best solution based on certain application requirements.

This application note cannot and does not replace any of the relevant datasheets.

“Card” in this document means a contactless smart card according to the ISO14443A (or MIFARE®) or a contactless tag / label according to the ISO15693 (or I-Code®).

“Micore” includes all contactless reader ICs as the MF RC500, MF RC530, MF RC531, SL RC400 and the CL RC632. For all of these ICs the antenna matching is the same.

Remarks and Comments are given in italic letters (blue marked).

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

3 THE MIFARE® RF INTERFACE

The MIFARE® technology describes an ISO 14443-Type A compliant RF interface for a communication between a PCD and a PICC.

Table 1 gives a short overview on the MIFARE® RF interface. Essentially the MIFARE® RF interface follows the transformer principle, although both the PCD and PICC antenna of course are resonance circuitries as antennas usually are. The PICC is passive with no onboard battery. Thus, an energy transmission from the PCD to the PICC is required in addition to the communication (data transmission) in both directions between the PCD and the PICC (see Figure 1).

Table 1: Overview MIFARE® RF interface

Energy transmission	Transformer principle; MIFARE® card is passive
Operating frequency	13.56 MHz
Communication structure	Half duplex, reader talks first
Data rate	105.9 kHz
Data transmission	
RWD → Card	100 % ASK, Modified Miller Code
Card → RWD	Load Modulation, Subcarrier 847.5 kHz, OOK, Manchester Code

In the following the fundamentals of the MIFARE® RF interface are described starting with the basic energy transmission. Finally, the data transmission and the used data coding in both directions will be shown.

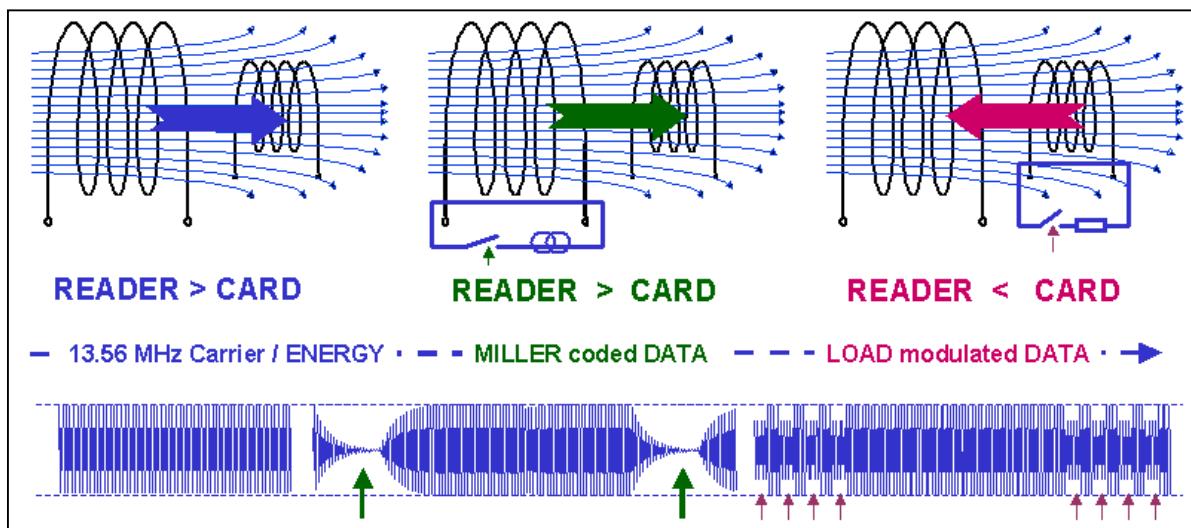


Figure 1: Mifare Interface Principle

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

3.1 Energy Transmission

The energy transmission from the PCD to the passive PICC is based on the transformer principle. At PCD side an antenna coil is required as well as a card coil implemented in the MIFARE® card (PICC). Figure 2 shows the basic principle and the equivalent electronic circuitry.

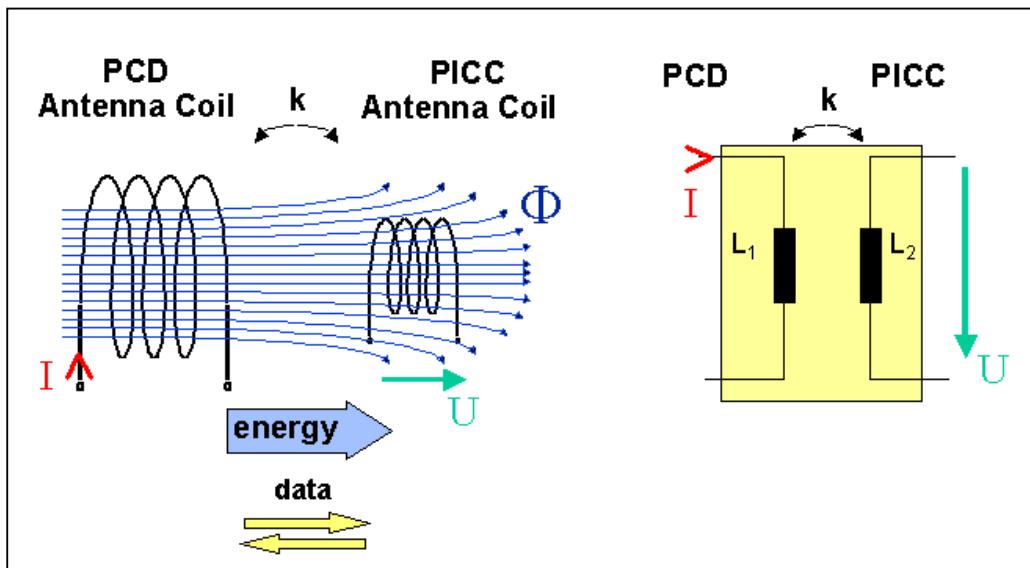


Figure 2: Transformer Model

The figure's left part describes the antennas and the energy transmission. The current I in the PCD antenna coil generates a magnetic flux Φ . Parts of this flux Φ flow through the card coil and induce a voltage U in the card coil itself. This voltage U is rectified and the card IC is activated when the operating voltage is reached. The induced voltage will vary within the distance between PCD antenna and the PICC. Due to that voltage variation, the achievable operating distance is limited by the transferred power.

The right part shows the equivalent electrical circuitry, the transformer model.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

3.2 Data Transmission PCD → PICC

To transfer data between the PCD and the PICC a half-duplex communication structure is used. The PCD always starts the communication (“reader talks first”). The data transmission from the PCD to the PICC uses a 100% ASK modulation according to the ISO14443 Type A. Figure 3 shows a typical signal shape.

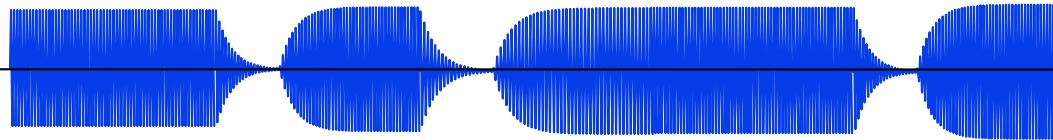


Figure 3: Data Transmission PCD → PICC, typical signal shape

Due to the quality factor Q of the antenna the transmitted signal is deformed to the shape shown in Figure 3. This shape can be used to measure the tuning of the antenna. For details refer to the “MIFARE® and I Code, Micore Directly Matched Antenna Design”.

As the PICC is passive, the energy for the PICC has to be provided during the communication between PCD and PICC. Therefore, MIFARE® (ISO14443A) uses an optimised coding to provide a constant level of energy independently from the data transmitted to the PICC. This is the modified Miller code, which is shown in Figure 4 in detail.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

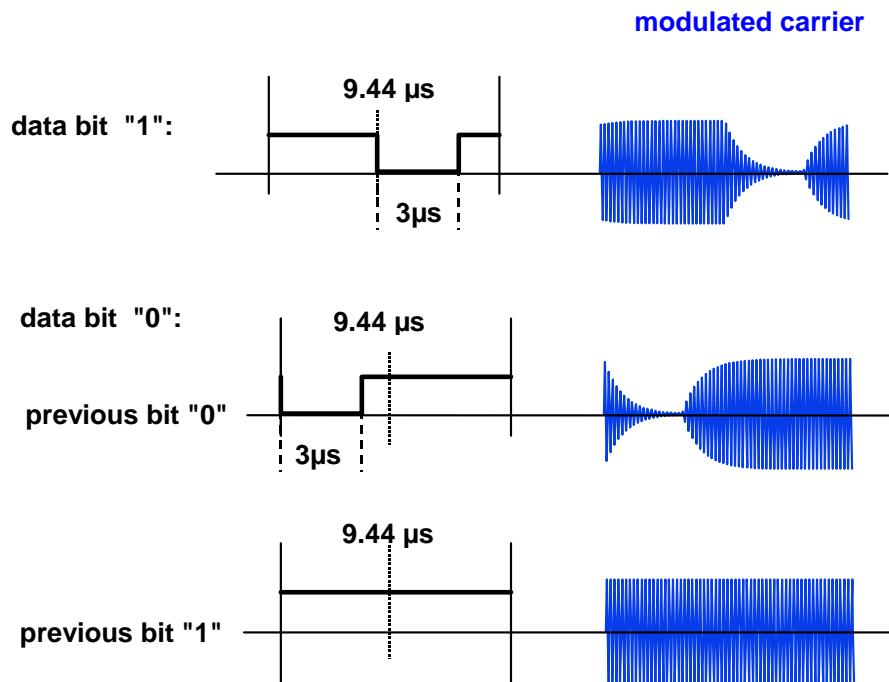


Figure 4: Data Transmission PCD → PICC, Miller Coding

The data rate of MIFARE® is 105.9KHz, so the length of a bit frame is 9.44μs. A pulse in the Miller coding has a length of 3μs.

A logical '1' is expressed with a pulse in the centre of the bit frame.

Two possibilities are given to code a logical '0'. This coding depends on the previous bit:

If the previous bit was a '0', the following '0' is expressed with a pulse of 3μs at in the first half of the next bit frame.

If the previous bit was a '1', the following '0' is expressed without a pulse in the next bit frame.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

3.3 Data Transmission PICC → PCD

3.3.1 SUBCARRIER LOAD MODULATION PRINCIPLE

The data transmission from the PICC back to the PCD uses the principle of load modulation shown in Figure 5. The PICC is designed as a resonance circuitry and consumes energy generated by the PCD. This energy consumption has a feedback effect as a voltage drop on PCD side. This effect is used to transfer data from the PICC back to the PCD by changing the load in the card IC.

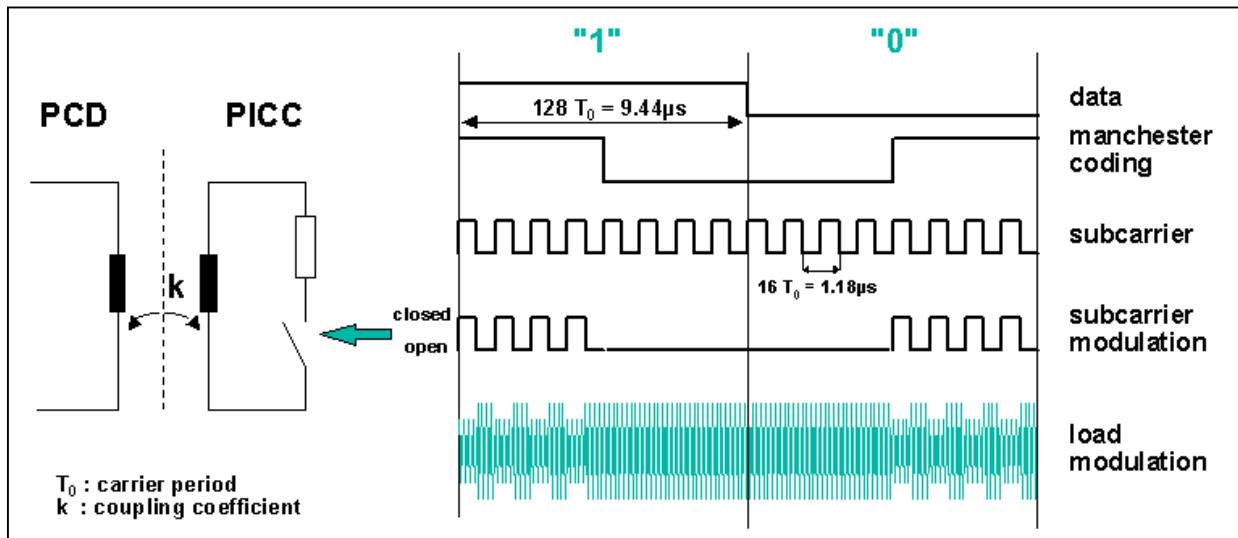


Figure 5: Subcarrier Load Modulation Principle

The PCD antenna is tuned to a resonance frequency $f_R = 13.56$ MHz. The time T_0 expresses the pulse length of the operating frequency $T_0 = \frac{1}{f_R} \approx 74\text{ns}$. In fact, this resonance circuit generates voltages at

the PCD antenna several times higher than the supply voltage. Due to the small coupling factor between the PCD and PICC antenna the PICC's response is up to 60dB below the voltage generated by the reader. To detect such a signal, it requires a well designed receiving circuit.

The PICC data transfer back to the PCD uses a data rate of 105.9kbit/s with Manchester coding. At Manchester Coding each bit is represented by either a raising or a falling edge in the centre of a bit frame. For the MIFARE® principle this is shown on the right side of Figure 5:

A logical '1' is expressed with a falling edge in the centre of the bit frame.

A logical '0' is expressed with a rising edge in the centre of the bit frame.

This Manchester coded data modulates a sub carrier $f_{SUB} = \frac{f_R}{16} = 847.5\text{kHz}$.

Finally, this modulated sub carrier switches the load of the PICC, which results in the load modulation as shown in the last line of Figure 5, and which is received and decoded again by the PCD.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

Figure 6 shows the relation between the time and the frequency domain of the load modulation. Due to the data rate of $v \approx 106kBd \approx \frac{1}{9.44\mu s}$ the Manchester code generates sidebands at both sides of the sub carrier frequency:

$$f_{mSUB} = 847.5\text{kHz}|_{Subcarrier} \pm 106\text{kHz}|_{Data}$$

The modulated sub carrier then generates sidebands at both sides of the carrier frequency:

$$f_{mR} = 13.56\text{MHz}|_{Carrier} \pm 847.5\text{kHz}|_{Subcarrier} \pm 106\text{kHz}|_{Data}$$

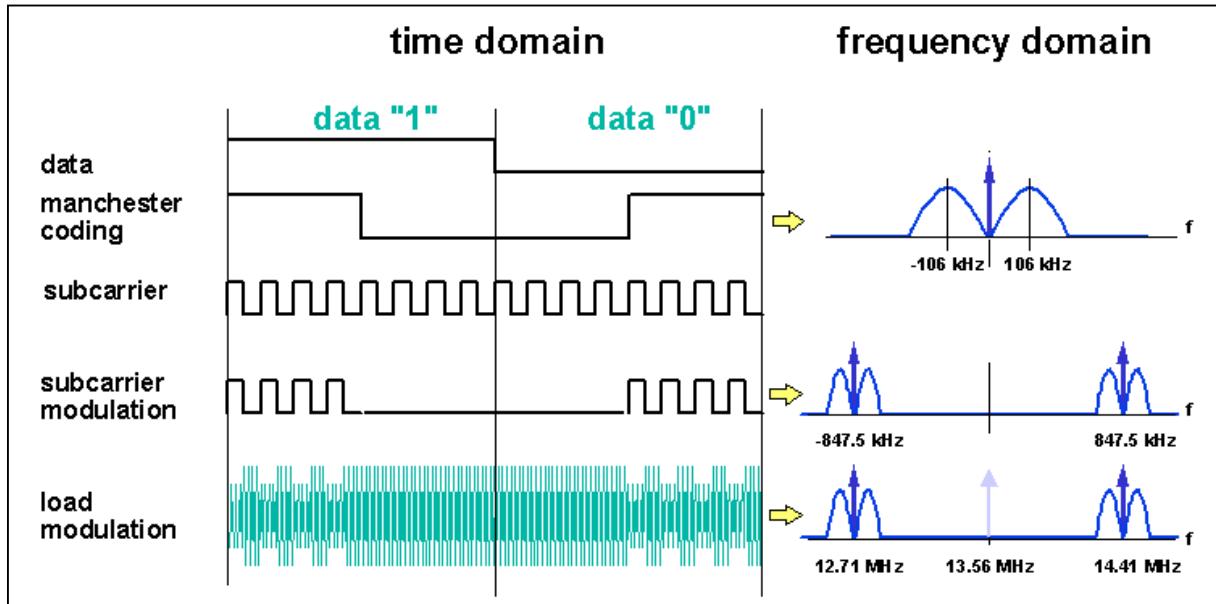


Figure 6: Principle of Data Coding PICC → PCD, Time and Frequency Domain

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

4 THE PCD ANTENNA

The achievable operating distance for a Proximity RFID system depends on:

- PCD (and PICC) Antenna size
- Matching of the antenna
- Quality factor of the antenna and matching circuit
- Power of the PCD
- Environmental influences

Assuming that the matching of the antenna is optimised, the Q-Factor, the PICC antenna, and the delivered power of the PCD are given and fix, only the PCD antenna size and the environmental influences have to be considered for the antenna design principle. For the details of the Micore antenna design refer to "MIFARE® and I Code, Micore Directly Matched Antenna Design".

In the first step the environmental influences are neglected. This can even be done in many practical cases. The environmental influence and the way, how to avoid it, is shown in chapter 4.2.

4.1 The PCD Antenna size

Based on the transformer principle (as shown in Figure 2) and with the help of the law of the electromagnetic induction, the law of Biot and Savart the mean flux density in dependence on the operating distance and the PCD antenna radius can be calculated. Based on the limit of the minimum required field strength as specified in the ISO14443-2:

$$H_{\min} = 1.5 \text{ A/m}$$

the coupling factor can be calculated in dependence on the operating distance and the PCD antenna radius.¹

As a rule of thumb the following can be shown (see Figure 7):

$$x = R = \frac{D}{2}$$

x: maximum operating distance

R: PCD antenna radius

D: PCD antenna diameter

¹ as shown in the Annex, chapter 6.1

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

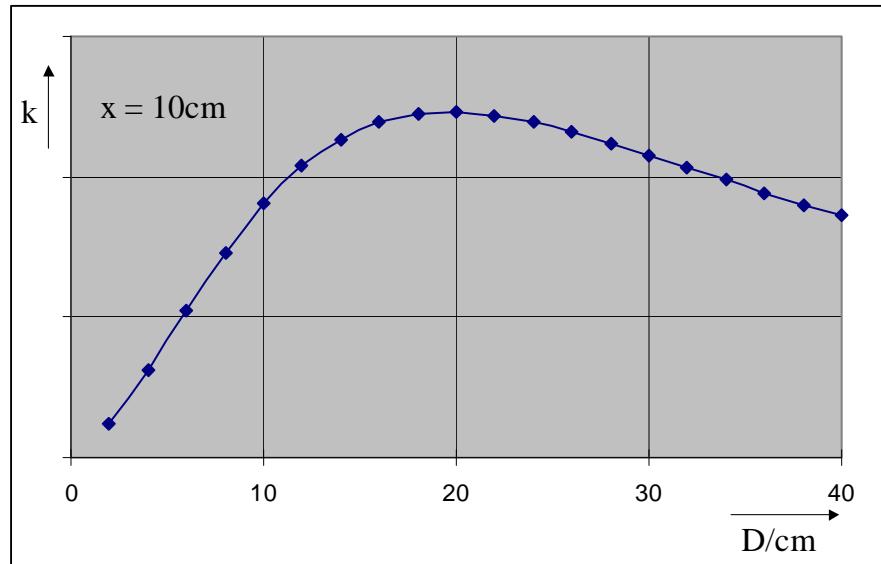


Figure 7: Coupling factor vs. Antenna diameter

Figure 7 shows the coupling factor k versus antenna diameter D based on the required operating distance of $x = 10\text{cm}$. The optimum antenna diameter is $D = 20\text{cm}$, which means bigger antennas give no better (but even less) performance. Although the curve is very flat on its top, it can be seen that an antenna diameter less than 16cm shows a significant decrease of the coupling factor (and performance). The coupling factor and performance goes dramatically down at antenna diameter below 12 cm.

Remark:

The above shown relation between antenna size and operating distance is independent on the number of turns (and the inductance) of the PCD antenna.

The above shown relation is only a first step for a successful antenna design. For a complete design, the environmental influences as well as antenna size limitations due to application related restrictions have to be considered.

The coupling coefficient is a limiting factor not only for the energy that is transported to the PICC, but also for the PICC's answer that has to be received by the PCD.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

4.2 Environmental Influences

4.2.1 METAL ENVIRONMENT

Any alternating magnetic field induces a voltage in metal components positioned nearby the reader antenna. This induced voltage generates eddy currents in the metal plane. These eddy currents cause loss combined with a detuning of the antenna and decreasing of the magnetic field. The result of these effects is a reduced operating distance as well as possible transmission errors.

It is recommended that the distance between antenna and massive metal components is at least as large as the operating distance (on both sides of the PCD antenna).

To avoid negative influences of a metallic environment a ferrite shielding should be used.

The antenna distance from massive metal should be at least 10cm for full R/W distance, 3cm for reduced R/W distance, and for close metal ferrite shielding is strictly recommended.

In all cases the tuning of the antenna has to be made with the metal placed in the finally intended position.

4.2.2 MULTIPLE PCD ANTENNAS

Antennas are resonance circuits with a high quality factor and tuned to the operating frequency. According to the reciprocity law a good transmitting antenna is also a good receiving antenna and vice versa. This means that an antenna positioned close to the used reader antenna and tuned to the same frequency dissipates energy from the field. This causes a detuning of the antenna and a reduced operating distance. If two active antennas for an MIFARE® application are positioned in a close distance a communication to the card will be disturbed.

Multiple Proximity PCD antennas should be at least 30 cm away from each other if they are magnetically shielded and 10 times of the antenna radius if they are not shielded!

4.2.3 TEMPERATURE

The PCD antenna may be detuned as a consequence of temperature drifts of the electrical parameters of the antenna itself and the matching circuit. This will result in a reduction of the transmitting power available at the antenna. The consequence will be a reduced operating distance.

Experiments can show that these influences can be neglected when appropriate components with low temperature coefficient for the matching circuit (SMD capacitors with NP0 dielectric medium) are used.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

4.3 Antenna Shielding and Compensation

Three topics shall be discussed.

- Electrical Shielding
The electrical shielding absorbs the electrical field generated by the antenna coil as well as the electrical field of the reader PCB.
- Compensation
Compensation should be used to reduce common mode earth currents.
- Ferrite Shielding
Ferrite Shielding should be used if metal has to be placed very close to the antenna itself. This metal, e.g. metal housing of the terminal generates eddy currents. The effect of the eddy currents is a dramatically reduced operating distance. A ferrite shielding should be used to reduce the generated eddy currents.
Note: Ferrite shielding will not increase the operating distance above values achievable in non metallic environment.

4.3.1 ELECTRICAL SHIELDING**4.3.1.1 Directly Matched Antennas**

An electrical shielding should be used to reduce the electrical field generated by the antenna coil itself. To build a shielded antenna on a PCB at least one with 4 layers should be used with the shielding loop on the top and the bottom layer. **These loops must not be closed**. The loops provide electrical shielding and improve the EMC behaviour. The shielding has to be connected **in one point** to system ground. The coil is routed in the first inner layer. The centre tap of the coil is done with the marked Via to GND. The connection of the coil ends to the matching circuit shall be routed close together, to avoid additional inductance.

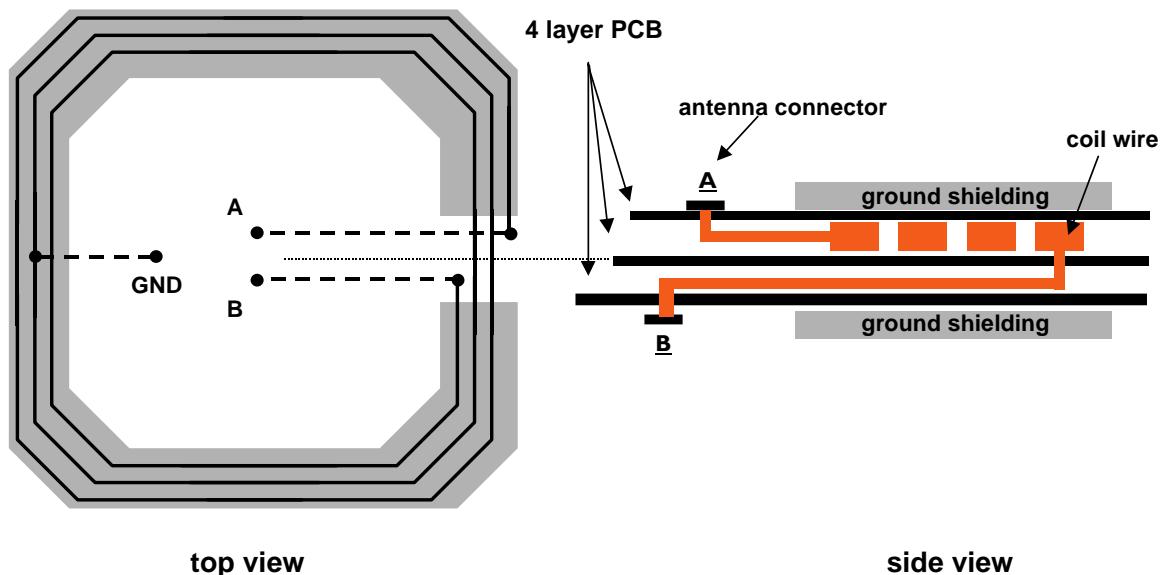


Figure 8: Electrical shielding for a directly matched antenna

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

4.3.1.2 50Ω Matched Antennas

An electrical shielding should be used to reduce the electrical field generated by the antenna coil itself. To build a shielded antenna on a PCB at least one with 4 layers should be used with the shielding loop on the top and the bottom layer. **These loops must not be closed**. The loops provide electrical shielding and improve the EMC behaviour. The shielding has to be connected **in one point** to system ground.

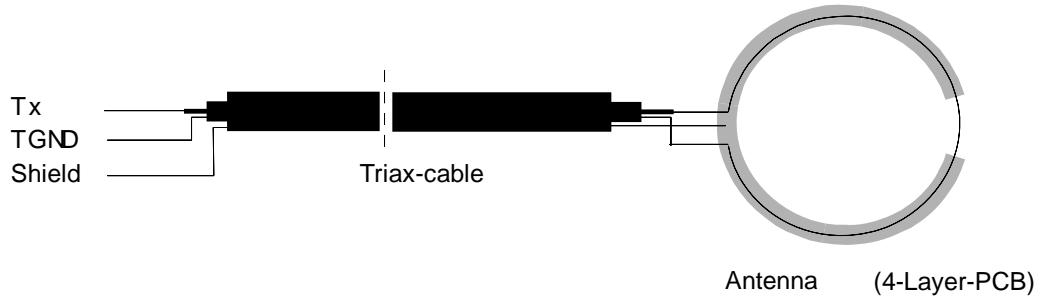


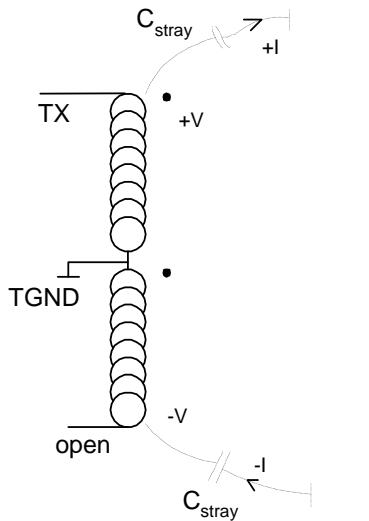
Figure 9: Electrical shielding for a 50Ω matched antenna using a triax cable.

On the top and bottom layers of the PCB a shielding plane is placed directly above the active antenna loop which is an inside layer of the PCB. These shielding planes must not be closed loops! The shielding should be connected with a triax cable.

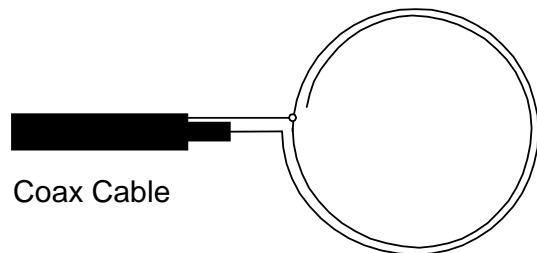
13.56 MHz RFID Proximity Antennas

mifare® (14443A)

4.3.2 COMPENSATION



Electrical Principle



Implementation

Figure 10: Compensated 50Ω antenna

To compensate the stray capacitance of the antenna another turn with an open end is added. Due to the transformer's principle the induced voltage in the open loop is inverted. The stray capacitance of the active and the compensation loop have nearly the same value. The effect will be, that the current across these capacitance has nearly the same magnitude but opposite direction. By that a compensation of these currents is done. These currents can reach values in a range of several mA at 13.56 MHz, so compensation is necessary to avoid problems with ground currents.

4.3.3 FERRITE SHIELDING

The benefit of a ferrite is to shield an antenna against the influence of metal. A metal plane could be part of the housing of the reader or a ground plane of the reader PCB itself, which has to be connected very near to the antenna. If metal is placed very near to the antenna the alternating magnetic field generates eddy currents in the metal. These eddy currents absorb power, and lead to detuning of the antenna due to a decreased inductance and quality factor. Therefore for operation of an antenna in metallic environment, it is necessary to shield the antenna with ferrite.

The following examples should give an impression on the influence of ferrite for the distribution of a magnetic field.

For easy simulation a circular antenna has been used in all case. A circular antenna is rotation symmetrical to the x-axis. Therefore the simulation can be reduced to a two dimensional mathematical problem. The simulations shows on the one hand the field distribution of a non disturbed antenna. Common for all examples: Radius of the RWD antenna 7.5 cm, 1 turn, wire width 1mm copper.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

Figure 11 shows the two dimensional field of the circular antenna. The right part shows the field distribution. The highest field strength is generated in the area of the coil. The left part shows the magnitude of the field strength H over the distance d . The line of a minimal field strength of $H_{MIN} = 1.5 \text{ A/m}$ according to ISO 14443 is marked.

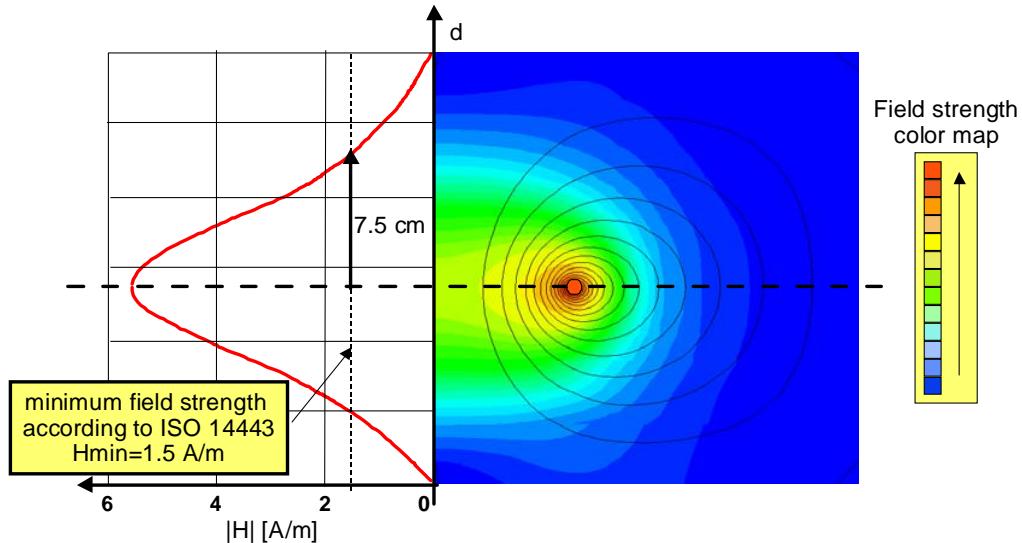


Figure 11: Non disturbed field distribution of a circular antenna

Figure 12 shows the field distribution of the same antenna with a metal plane near to the antenna. Compared to the disturbed field it is obvious that the magnitude of the field strength has decreased leading to a decreased operating distance.

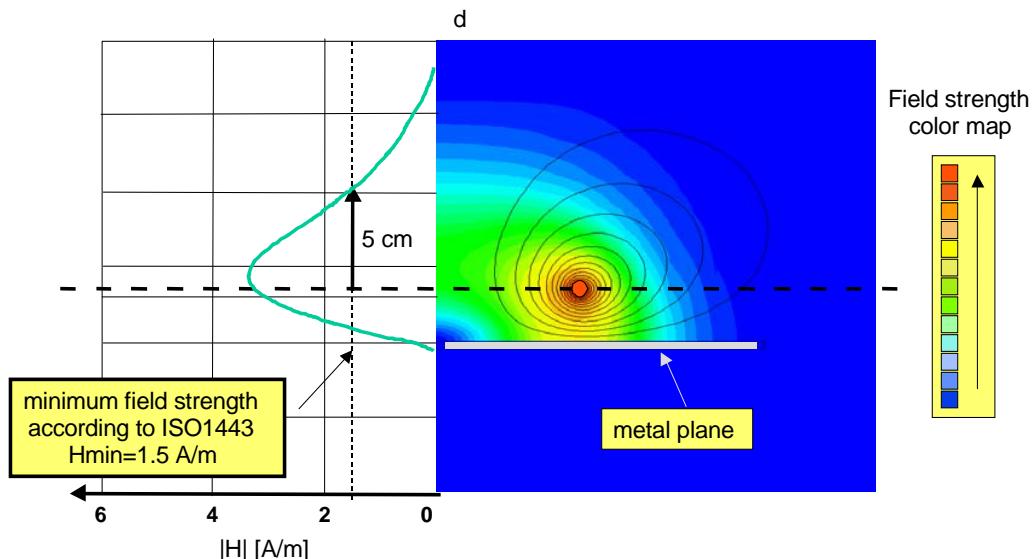


Figure 12: Field distribution of a circular antenna with a metal plane

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

Now, as shown in Figure 13 a ferrite plane ($\mu_R=40$) is positioned in between the metal plane and the antenna coil itself. The field strength very near to the ferrite increases, but this increasing of the magnitude is not combined with an increasing of the operating distance. This is marked once again with the H_{MIN} value according to ISO 14443.

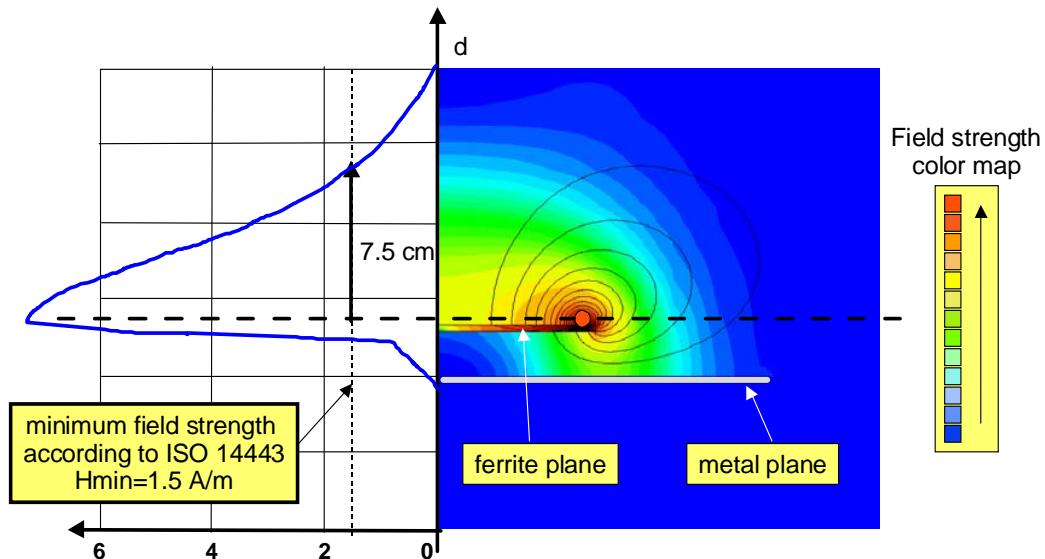


Figure 13: Ferrite shielded field distribution of a circular antennae

These simulations show how the use of ferrite reduces the generated eddy currents in a metal plane. The ferrite generates an additional field component and the effect for the design of the antenna is a fixed detuning of the antenna itself.

Figure 14 gives recommendations how to dimension the ferrite to find the optimum dimensions between ferrite plane and metal plane. To calculate the optimum dimensions of the ferrite plane and the optimum distance and overlapping is very hard and not recommended. Application specific tests have to be made to find the best ferrite dimensions.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

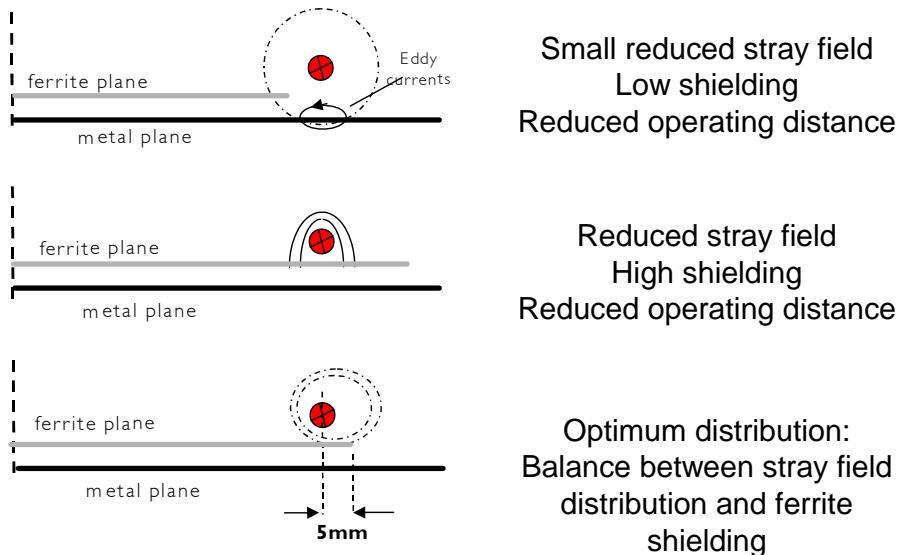


Figure 14: Estimation of the optimum ferrite dimensions

Tests have shown that the best performance is achieved when the overlapping of the antenna coil and the ferrite is in a range of 5 mm. That gives a balance between needed stray field to communicate to the card and the shielding of the ferrite.

Applying the distance estimation to specific applications, it is recommended to make test to find the best solution. Once again it has to be mentioned that ferrite does not increase the operating distance compared to a non-disturbed field.

13.56 MHz RFID Proximity Antennas

mifare® (14443A)**5 ANTENNA DECISION GUIDE**

Micore is a single reader IC family, which is designed to achieve operating distances up to 100mm without external amplifiers. The design of the remaining passive RF part is straightforward.

Two different concepts are possible to design an antenna and a matching circuit.

- **Directly matched antennas** can be used to build up small, complete terminals with a minimum distance between a reader and antenna. Possible applications could be an access control reader in a small housing or a handheld reader.
- **50Ω matched antennas** can be used for an easy solution to achieve long distances between the reader and the antenna using a coaxial cable. Using a coaxial cable between the reader matching circuit and the antenna itself, distances up to 10 m between these parts are possible. A high-end solution to achieve an operating distance up to 100mm as well as a low cost solution for operating distances lower than 50 mm is available.

At first it has to be decided, which of the possible basic concepts meets the application requirements best. The design help shown in Figure 15 and the comparison shown in Table 2 shall give a support for this decision.

Table 2: Comparison of antenna concepts

Concept		50 Ω matched		Directly matched
		Full range	Short range	
Reader	MF RC 500			
	EMC-Circuit		same circuitry and values	
	Receiving circuit		same circuit and values	
	Impedance Transformation	using T_{X1} and T_{X2}	using only T_{X1}	---
Antenna	Cable	50 Ω coaxial		Short wire or directly connected
	Antenna matching circuitry	same circuit, but different values depending on the antenna size		Same circuit, but different values depending on the antenna size
	Antenna coil	operating distance depends on the antenna size and environmental influences		operating distance depends on the antenna size and environmental influences
	Antenna shielding	Shielding depends on the application, e. g. the housing and environmental influences		

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

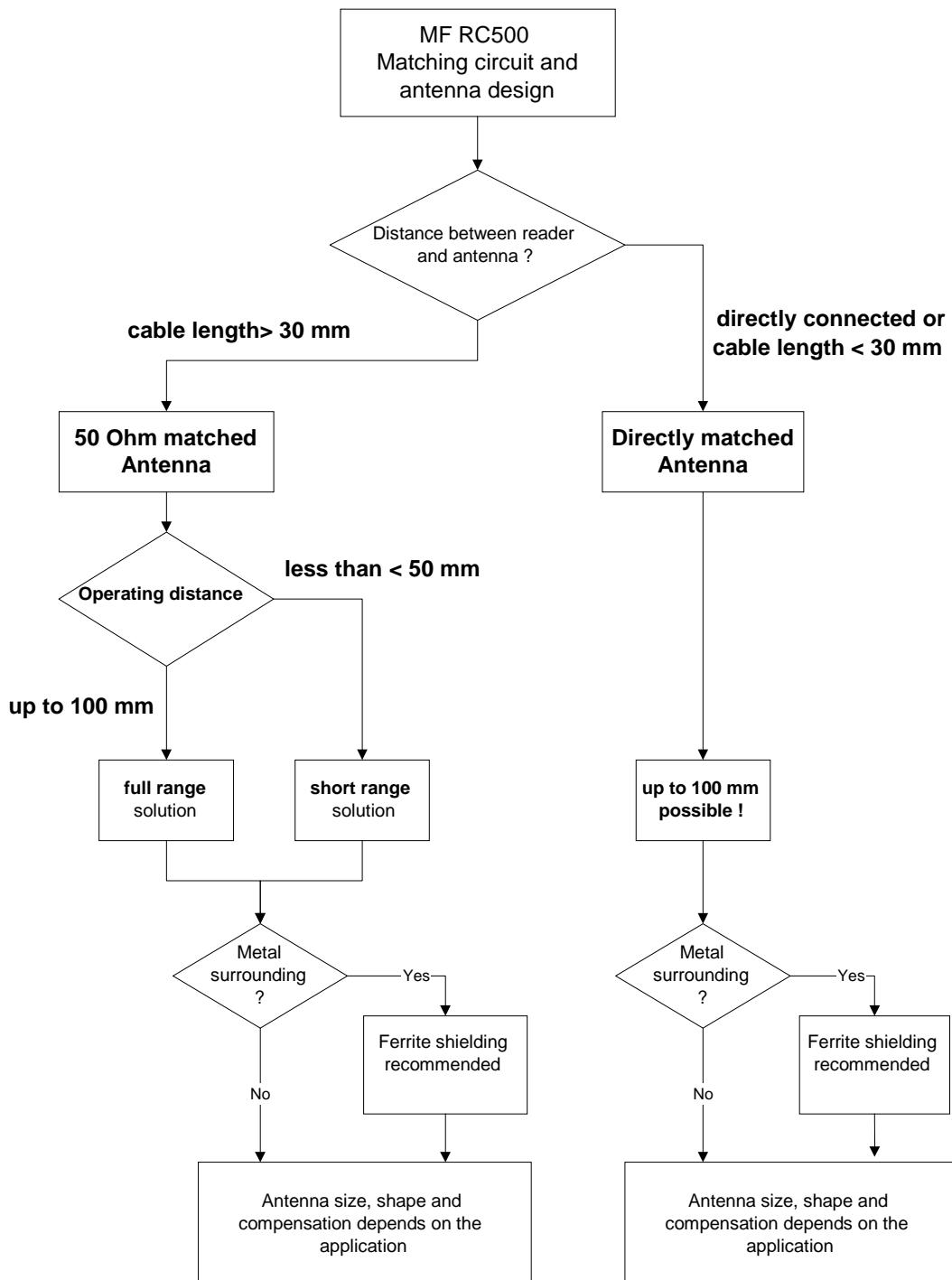


Figure 15: Design Help

13.56 MHz RFID Proximity Antennas

mifare® (14443A)

6 ANNEX

6.1 Theory for estimating the optimum PCD antenna radius

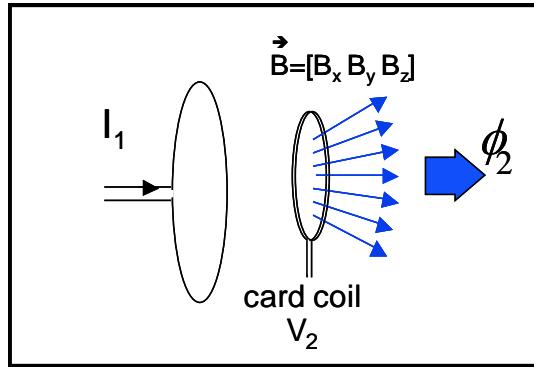
It's not necessary to know and use the below given formulas for the design of a Micore PCD antenna. The principle is described above and details for that design can be found in "MIFARE® and I Code, Micore Directly Matched Antenna Design".

Electromagnetic Induction:

$$V_{ind} = -N_2 \cdot \frac{d\phi}{dt} = -N_2 \cdot \frac{d}{dt} \int_A B_x \cdot dA$$

Induced open loop voltage:

$$\begin{aligned} v_{20}(t) &= N_2 \frac{d \hat{\phi}_2 \cdot \sin \omega t}{dt} \\ &= N_2 \cdot \omega \cdot \hat{\phi}_2 \cdot \cos \omega t \end{aligned}$$

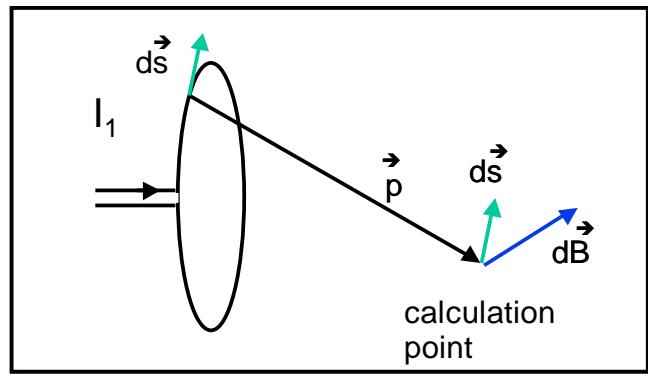


$$\begin{aligned} V_{20} &= N_2 \cdot \omega \cdot \hat{\phi}_2 \\ &= N_2 \cdot \omega \cdot \hat{B} \cdot A_2 \end{aligned}$$

card coil parameters:
 A_2 : area
 N_2 : number of turns

Law of Biot and Savart:

$$\vec{B} = \frac{\mu_0 \cdot I_1}{4\pi} \cdot \oint \frac{[ds \times p]}{p^3}$$



13.56 MHz RFID Proximity Antennas

mifare® (14443A)

Optimisation of the coupling coefficient:

$$\begin{aligned} V_{20} &= \omega \cdot M \cdot I_1 \\ M &= \frac{V_{20}}{\omega \cdot I_1} \\ M &= k \cdot \sqrt{L_1 \cdot L_2} \\ V_{20} &= N_2 \cdot \omega \cdot \hat{B} \cdot A_2 \\ \hat{B} &= \mu_0 \cdot I_1 \cdot \frac{N_1 \cdot r^2}{2 \cdot (r^2 + x^2)^{3/2}} \end{aligned}$$

↓

$$k = \mu_0 \cdot \frac{r^2}{2(r^2 + x^2)^{3/2}} \cdot \sqrt{\frac{N_1^2 \cdot N_2^2}{L_1 \cdot L_2} \cdot A_2}$$

$$L_1 = L_{01} \cdot N_1^2 \quad L_2 = L_{02} \cdot N_2^2$$

L_{01}, L_{02} : single turn inductance

$$L_{01} \approx \frac{2 \cdot 10^{-7}}{[\text{m}]} \cdot 2\pi \cdot r \cdot \ln\left(\frac{2\pi \cdot r}{d}\right)$$

Coupling coefficient $k = f(r, x)$:

$$k = \mu_0 \cdot \frac{r^2}{2(r^2 + x^2)^{3/2}} \cdot \frac{A_2}{\sqrt{L_{01} \cdot L_{02}}}$$

13.56 MHz RFID Proximity Antennas**mifare® (14443A)**

6.2 Parameters and UnitsV_{ind}: Induced voltage

N: Number of turns

Φ: Magnetical Flux

t: time

B: Magnetical Flux densitiy

A: (Coil) Area

V₂₀: Induced open loop voltage (card antenna coil)I₁: PCD antenna coil current

M: Mutual inductance (transformer model)

L: Inductance

r: PCD antenna radius

x: PCD-PICC operating distance

k: coupling coefficient

L₀₁, L₀₂: Single turn inductance

Index 1 is used for the PCD antenna.

Index 2 is used for the PICC antenna. Parameter with index 2 are fixed (according to the ISO14443-1).

Definitions

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics section of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

Life support applications

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

Philips Semiconductors - a worldwide company

Contact Information

For additional information please visit <http://www.semiconductors.philips.com>. Fax: +31 40 27 24825

For sales offices addresses send e-mail to: sales.addresses@www.semiconductors.philips.com.

© Koninklijke Philips Electronics N.V. 2002

SCA74

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

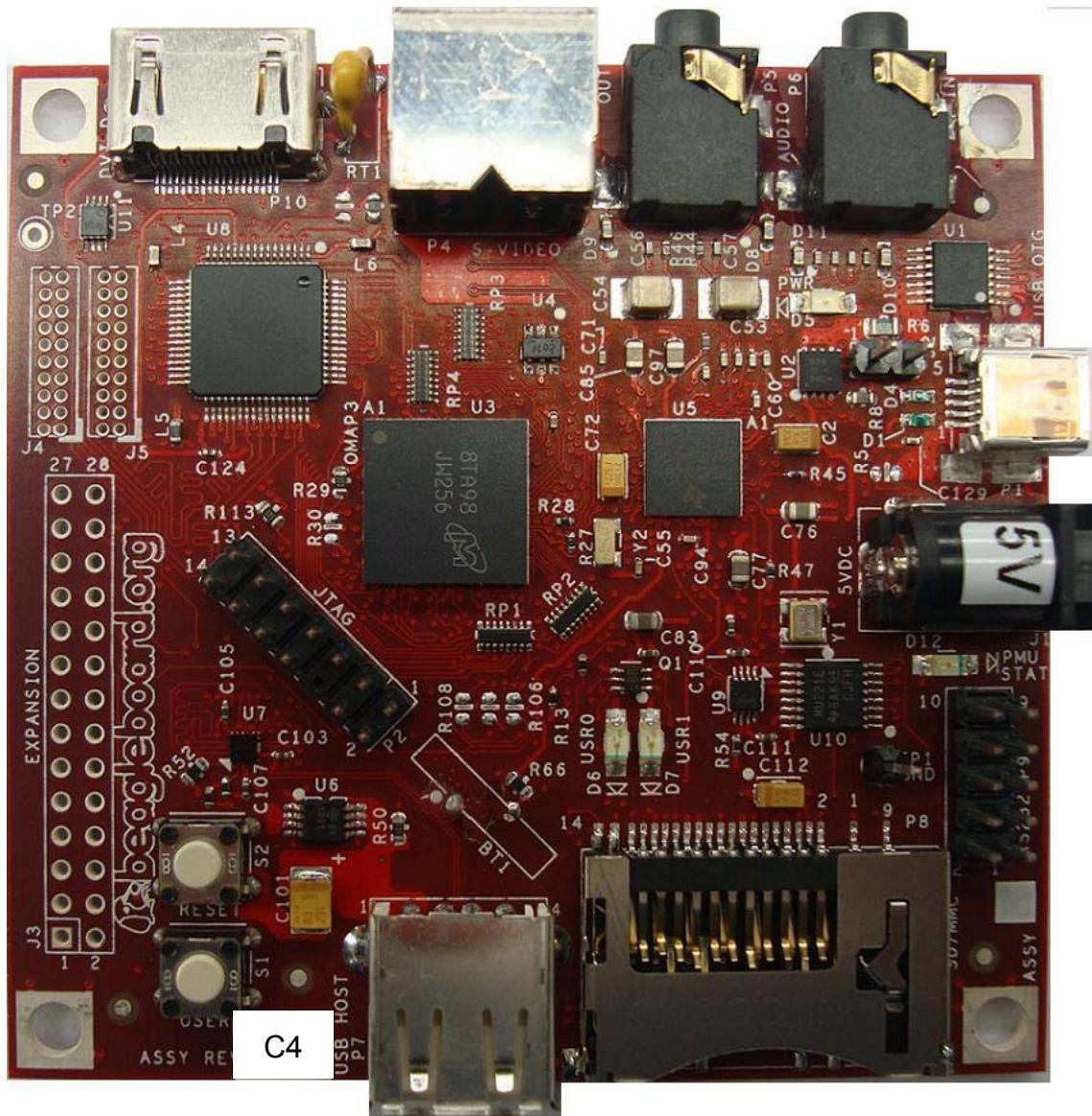
The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without any notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Let's make things better.

**Philips
Semiconductors**



PHILIPS



BeagleBoard

System Reference Manual

Rev C4

Revision 0.0
December 15, 2009

THIS DOCUMENT

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

All derivative works are to be attributed to Gerald Coley of BeagleBoard.org.

For more information, see http://creativecommons.org/licenses/results-one?license_code=by-sa

For any questions, concerns, or issues submit them to gerald@BeagleBoard.org

BEAGLEBOARD DESIGN

These design materials referred to in this document are ***NOT SUPPORTED*** and **DO NOT** constitute a reference design. Only “community” support is allowed via resources at BeagleBoard.org/discuss.

THERE IS NO WARRANTY FOR THE DESIGN MATERIALS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN MATERIALS “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN MATERIALS IS WITH YOU. SHOULD THE DESIGN MATERIALS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

We mean it; these design materials may be totally unsuitable for any purposes.

BeagleBoard.org provides the enclosed product(s) under the following conditions:

This evaluation board/kit is intended for use for **ENGINEERING DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY** and is not considered by BeagleBoard.org to be a finished end-product fit for general consumer use. Persons handling the product(s) must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards. This evaluation board/kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL, and therefore may not meet the technical requirements of these directives or other related directives.

Should this evaluation board/kit not meet the specifications indicated in the User's Guide, the board/kit may be returned within 30 days from the date of delivery for a full refund. THE FOREGOING WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY SELLER TO BUYER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies BeagleBoard.org from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

BeagleBoard.org currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. BeagleBoard.org assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

Please read the User's Guide and, specifically, the Warnings and Restrictions notice in the User's Guide prior to handling the product. This notice contains important safety information about temperatures and voltages. For additional information on BeagleBoard.org environmental and/or safety programs, please contact visit BeagleBoard.org.

No license is granted under any patent right or other intellectual property right of BeagleBoard.org covering or relating to any machine, process, or combination in which such BeagleBoard.org products or services might be or are used.

Mailing Address:

BeagleBoard.org
675 North Glenville #195
Richardson, TX 75081

WARRANTY: *The BeagleBoard is warranted against defects in materials and workmanship for a period of 90 days from purchase. This warranty does not cover any problems occurring as a result of improper use, modifications, exposure to water, excessive voltages, abuse, or accidents. All boards will be returned via standard mail if an issue is found. If no issue is found or express return is needed, the customer will pay all shipping costs.*

Before returning the board, please visit BeagleBoard.org/support

Please refer to sections 12 and 13 of this document for the board checkout procedures and troubleshooting guides.

To return a defective board, please request an RMA at <http://beagleboard.org/support/rma>.

Table of Contents

FIGURES.....	9
TABLES.....	11
1.0 INTRODUCTION.....	13
2.0 CHANGE HISTORY.....	14
2.1 CHANGE HISTORY.....	14
2.2 REVISION C3 VS. C4	14
3.0 DEFINITIONS AND REFERENCES.....	15
3.1 DEFINITIONS	15
4.0 BEAGLEBOARD OVERVIEW.....	15
4.1 BEAGLEBOARD USAGE SCENARIOS	15
5.0 BEAGLEBOARD SPECIFICATION.....	17
5.1 BEAGLEBOARD FEATURES.....	17
5.2 OMAP PROCESSOR.....	18
5.3 MEMORY.....	18
5.4 POWER MANAGEMENT.....	18
5.5 HS USB 2.0 OTG PORT.....	19
5.6 HS USB 2.0 HOST PORT	19
5.7 STEREO AUDIO OUTPUT CONNECTOR.....	20
5.8 STEREO AUDIO IN CONNECTOR	20
5.9 S-VIDEO CONNECTOR.....	20
5.10 DVI-D CONNECTOR.....	20
5.11 LCD HEADER	20
5.12 SD/MMC 6 IN 1 CONNECTOR	21
5.13 RESET BUTTON	21
5.14 USER/BOOT BUTTON	21
5.15 INDICATORS	22
5.16 POWER CONNECTOR	22
5.17 JTAG CONNECTOR	23
5.18 RS232 HEADER	23
5.19 EXPANSION HEADER.....	23
5.20 BEAGLEBOARD MECHANICAL SPECIFICATIONS.....	23
5.21 ELECTRICAL SPECIFICATIONS	24
6.0 PRODUCT CONTENTS.....	26
6.1 BEAGLEBOARD IN THE BOX REV C4	26
6.2 SOFTWARE ON THE BEAGLEBOARD	27
6.3 REPAIR	27
7.0 BEAGLEBOARD HOOKUP.....	28
7.1 CONNECTING USB OTG	28
7.2 CONNECTING USB Host	29
7.3 CONNECTING OPTIONAL POWER.....	30
7.4 CONNECTING JTAG	31
7.5 CONNECTING SERIAL CABLE	32
7.6 CONNECTING S-VIDEO.....	33
7.7 CONNECTING DVI-D CABLE.....	34

7.8	CONNECTING STEREO OUT CABLE.....	35
7.9	CONNECTING STEREO IN CABLE.....	36
7.10	INDICATOR LOCATIONS.....	37
7.11	BUTTON LOCATIONS	38
7.12	SD/MMC CONNECTION.....	39
7.13	LCD CONNECTION.....	40
8.0	BEAGLEBOARD SYSTEM ARCHITECTURE AND DESIGN	41
8.1	SYSTEM BLOCK DIAGRAM	41
8.2	INPUT POWER.....	44
8.2.1	<i>USB DC Source</i>	45
8.2.2	<i>Wall Supply Source</i>	45
8.2.3	<i>DC Source Control</i>	45
8.2.4	<i>3.3V Supply</i>	46
8.2.5	<i>Meter Current Measurement</i>	46
8.2.6	<i>Processor Current Measurement</i>	46
8.3	POWER CONDITIONING	47
8.4	TPS65950 RESET AND POWER MANAGEMENT	48
8.4.1	<i>Main Core Voltages</i>	49
8.4.2	<i>Main DC Input</i>	49
8.4.3	<i>OMAP3530 I2C Control</i>	49
8.4.4	<i>VIO_IV8</i>	49
8.4.5	<i>Main Core Voltages Smart Reflex</i>	51
8.4.6	<i>VOCORE_IV3</i>	51
8.4.7	<i>VDD2</i>	51
8.5	PERIPHERAL VOLTAGES.....	52
8.5.1	<i>VDD_PLL2</i>	52
8.5.2	<i>VDD_PLL1</i>	52
8.5.3	<i>VDAC_IV8</i>	54
8.5.4	<i>VDD_SIM</i>	54
8.5.5	<i>VMMC1</i>	54
8.5.6	<i>VAUX2</i>	54
8.5.7	<i>Boot Configuration</i>	54
8.5.8	<i>RTC Backup Battery</i>	55
8.5.9	<i>Power Sequencing</i>	56
8.5.10	<i>Reset Signals</i>	57
8.5.11	<i>mSecure Signal</i>	58
8.6	OMAP3530 PROCESSOR.....	59
8.6.1	<i>Overview</i>	59
8.6.2	<i>SDRAM Bus</i>	60
8.6.3	<i>GPMC Bus</i>	60
8.6.4	<i>DSS Bus</i>	61
8.6.5	<i>McBSP2</i>	61
8.6.6	<i>McBSP1</i>	61
8.6.7	<i>McBSP3</i>	62
8.6.8	<i>Pin Muxing</i>	62
8.6.9	<i>GPIO Mapping</i>	64
8.6.10	<i>Interrupt Mapping</i>	64
8.7	POP MEMORY DEVICE	65
8.8	SYSTEMCLOCKS.....	65
8.8.1	<i>32KHz Clock</i>	65
8.8.2	<i>26MHz Clock</i>	66
8.8.3	<i>McBSP_CLKS</i>	66
8.9	USB OTG PORT.....	67
8.9.1	<i>USB OTG Overview</i>	67

8.9.2	<i>USB OTG Design</i>	68
8.9.3	<i>OTG ULPI Interface</i>	68
8.9.4	<i>OTG Charge Pump</i>	69
8.9.5	<i>OTG USB Connector</i>	70
8.9.6	<i>OTG USB Protection</i>	70
8.10	USB HOST PORT.....	70
8.10.1	<i>Host USB OMAP3 Interface</i>	71
8.10.2	<i>Host USB PHY</i>	71
8.10.3	<i>Host USB Connector</i>	72
8.10.4	<i>Host USB Power Control</i>	73
8.11	SD/MMC.....	73
8.11.1	<i>MMC Power</i>	75
8.11.2	<i>OMAP3530 Interface</i>	76
8.11.3	<i>Card Detect</i>	78
8.11.4	<i>Write Protect</i>	78
8.11.5	<i>8 Bit Mode</i>	78
8.11.6	<i>Booting From SD/MMC Cards</i>	78
8.12	AUDIO INTERFACE	79
8.12.1	<i>OMAP3530 Audio Interface</i>	79
8.12.2	<i>TPS65950 Audio Interface</i>	80
8.12.3	<i>Audio Output Jack</i>	80
8.12.4	<i>Audio Input Jack</i>	80
8.13	DVI-D INTERFACE	80
8.13.1	<i>OMAP3530 LCD Interface</i>	81
8.13.2	<i>OMAP3530 LCD Power</i>	82
8.13.3	<i>TFP410 Framer</i>	83
8.13.4	<i>TFP410 Power</i>	84
8.13.5	<i>TFP410 Control Pins</i>	84
8.13.6	<i>DVI-D Connector</i>	85
8.14	LCD EXPANSION HEADERS	87
8.15	S-VIDEO.....	89
8.16	RS232 PORT	90
8.16.1	<i>OMAP3530 Interface</i>	90
8.16.2	<i>OMAP3530 Level Translator</i>	90
8.16.3	<i>RS232 Transceiver</i>	90
8.16.4	<i>Connector</i>	91
8.17	INDICATORS	91
8.17.1	<i>Power Indicator</i>	92
8.17.2	<i>PMU Status Indicator</i>	92
8.17.3	<i>User Indicators</i>	93
8.18	JTAG	93
8.18.1	<i>OMAP3530 Interface</i>	93
8.18.2	<i>Connector</i>	94
8.19	EXPANSION HEADER	94
8.19.1	<i>OMAP3530 Interface</i>	95
8.19.2	<i>Expansion Signals</i>	96
8.19.3	<i>Power</i>	98
8.19.4	<i>Reset</i>	98
8.19.5	<i>Power Control</i>	98
8.20	ADDITIONAL EXPANSION HEADER.....	98
9.0	CONNECTOR PINOUTS AND CABLES	100
9.1	POWER CONNECTOR	100
9.2	USB OTG	101
9.3	S-VIDEO.....	102

9.4	DVI-D	103
9.5	LCD.....	105
9.5.1	<i>Connector Pinout</i>	105
9.5.2	<i>Connector Suppliers</i>	106
9.5.3	<i>Dimensions</i>	107
9.5.4	<i>Mounting Scenarios</i>	108
9.6	AUDIO CONNECTIONS	110
9.7	AUDIO OUT	111
9.8	JTAG.....	112
9.9	RS232.....	114
9.10	EXPANSION	115
9.11	BATTERY INSTALLATION	116
9.11.1	<i>Battery</i>	116
9.11.2	<i>Battery Installation</i>	116
10.0	BEAGLEBOARD ACCESSORIES	118
10.1	DC POWER SUPPLY	119
10.2	SERIAL RIBBON CABLE	120
10.3	USB HUBS	121
10.4	DVI CABLES	121
10.5	DVI-D MONITORS	122
10.6	SD/MMC CARDS	122
10.7	USB TO ETHERNET	123
10.8	USB TO WiFi	123
10.9	USB TO BLUETOOTH	124
10.10	EXPANSION CARD DESIGN	125
11.0	MECHANICAL INFORMATION	126
11.1	BEAGLEBOARD DIMENSIONS	126
11.2	BEAGLEBOARD EXPANSION CARD DESIGN INFORMATION	127
11.2.1	<i>Mounting Method</i>	127
11.2.2	<i>Expansion EEPROM</i>	127
12.0	BOARD VERIFICATION	129
12.1	EQUIPMENT	129
12.2	OUT OF THE BOX	130
12.3	SD CARD CONFIGURATION	131
12.4	SETUP	132
12.5	FACTORY BOOT VERIFICATION	132
12.6	BOARD SD BOOT	133
12.7	FACTORY BOOT REINSTALL	134
12.8	BOOTING THE KERNEL	136
12.9	UBOOT TESTS	142
12.9.1	<i>EDID Test</i>	142
12.9.2	<i>LED Test</i>	142
12.9.3	<i>DVI-D Test</i>	143
12.10	KERNEL BASED TESTS	143
12.10.1	<i>DVI-D Test</i>	143
12.10.2	<i>S-Video Test</i>	143
12.10.3	<i>Audio Test</i>	144
13.0	TROUBLESHOOTING	148
13.1	ACCESS POINTS	148
13.1.1	<i>Voltage Points</i>	148
13.1.2	<i>Signal Access Points</i>	150

13.2	TROUBLESHOOTING GUIDE.....	151
13.3	SERIAL PORT ISSUES.....	152
13.3.1	<i>First Step</i>	153
13.3.2	<i>Second Step</i>	154
13.3.3	<i>Third Step</i>	155
13.3.4	<i>Fourth Step</i>	155
14.0	KNOWN ISSUES.....	157
15.0	PCB COMPONENT LOCATIONS	158
16.0	SCHEMATICS.....	160
17.0	BILLS OF MATERIAL	171
18.0	PCB INFORMATION.....	172

Figures

Figure 1.	BeagleBoard Usage Scenarios	16
Figure 2.	USB Y-Cable	19
Figure 3.	The Rev C4 Box.....	26
Figure 4.	Rev C4 Box Contents.....	27
Figure 5.	USB OTG Connection	28
Figure 6.	USB Host Connection.....	29
Figure 7.	DC Power Connection	30
Figure 8.	BeagleBoard JTAG Connection	31
Figure 9.	BeagleBoard Serial Cable Connection.....	32
Figure 10.	BeagleBoard S-Video Connection.....	33
Figure 11.	BeagleBoard DVI-D Connection.....	34
Figure 12.	BeagleBoard Audio Out Cable Connection.....	35
Figure 13.	BeagleBoard Audio In Cable Connection.....	36
Figure 14.	BeagleBoard Indicator Locations	37
Figure 15.	BeagleBoard Button Location.....	38
Figure 16.	BeagleBoard SD/MMC Location	39
Figure 17.	BeagleBoard LCD Header Location.....	40
Figure 18.	BeagleBoard High Level Block Diagram	41
Figure 19.	BeagleBoard Top Side Components.....	42
Figure 20.	BeagleBoard Backside Components.....	43
Figure 21.	Input Power Section	44
Figure 22.	Processor Current Measurement	47
Figure 23.	Power Conditioning	48
Figure 24.	Main Power Rails.....	50
Figure 25.	Peripheral Voltages	53
Figure 26.	Power Sequencing.....	56
Figure 27.	Reset Circuitry	57
Figure 28.	OMAP3530 Block Diagram	59
Figure 29.	McBSP2 Interface.....	61
Figure 30.	McBSP1 Interface.....	62

Figure 31.	McBSP3 Interface	62
Figure 32.	POP Memory	65
Figure 33.	System Clocks.....	65
Figure 34.	USB OTG Design	68
Figure 35.	USB Host Design.....	71
Figure 36.	Example of an SDIO Card	74
Figure 37.	RS-MMC and Card	75
Figure 38.	SD/MMC Interface	77
Figure 39.	Audio Circuitry	79
Figure 40.	DVI-D Interface	81
Figure 41.	S-Video Interface	89
Figure 42.	RS232 Interface Design	90
Figure 43.	RS232 Cable	91
Figure 44.	Indicator Design.....	92
Figure 45.	JTAG Interface.....	93
Figure 46.	Expansion Header	94
Figure 47.	Power Connector.....	100
Figure 48.	USB OTG Connector	101
Figure 49.	OTG Host Shorting Pads	101
Figure 50.	S-Video Connector.....	102
Figure 51.	DVI-D Connector.....	103
Figure 52.	DVI-D Cable	104
Figure 53.	Top Mount LCD Adapter.....	107
Figure 54.	Top Mount LCD Adapter.....	108
Figure 55.	Bottom Mount LCD Adapter	109
Figure 56.	Audio In Plug	110
Figure 57.	Audio In Plug	110
Figure 58.	Audio Out Plug	111
Figure 59.	Audio In Plug	111
Figure 60.	JTAG Connector Pinout.....	112
Figure 61.	JTAG 14 to 20 Pin Adapter	113
Figure 62.	JTAG Connector Pinout.....	113
Figure 63.	RS232 Header	114
Figure 64.	RS232 Flat Cable	114
Figure 65.	Expansion Sockets	115
Figure 66.	Optional Battery.....	116
Figure 67.	Optional Battery Location.....	117
Figure 68.	DC Power Supply	119
Figure 69.	RS232 Cable	120
Figure 70.	RS232 Cable Wiring	120
Figure 71.	HDMI to DVI-D Cable	122
Figure 72.	USB to Ethernet Adapters.....	123
Figure 73.	USB to WiFi	124
Figure 74.	USB to Bluetooth.....	125
Figure 75.	BeagleBoard Dimension Drawing	126
Figure 76.	BeagleBoard Bottom Stacked Daughter Card	127

Figure 77.	BeagleBoard Expansion Board EEPROM Schematic	128
Figure 78.	BeagleBoard Voltage Access Points.....	149
Figure 79.	BeagleBoard Signal Access Points	151
Figure 80.	BeagleBoard Serial Cable Orientation.....	153
Figure 81.	DB9 Male Connector	154
Figure 82.	DB9 Null Modem Cable	154
Figure 83.	Serial Cable Loopback	155
Figure 84.	BeagleBoard Top Side Components.....	158
Figure 85.	BeagleBoard Bottom Side Components	159

Tables

Table 1.	Change History	14
Table 2.	BeagleBoard Features	17
Table 3.	BeagleBoard Electrical Specification Rev C4	24
Table 4.	OMAP3530 Pin Muxing Settings	63
Table 5.	OMAP3530 GPIO Pins.....	64
Table 6.	OMAP3530 Interrupt Pins	64
Table 7.	OMAP3530 ULPI Interface	68
Table 8.	OMAP3530 ULPI Interface.....	69
Table 9.	USB OTG Charge Pump Pins.....	70
Table 10.	USB Host Port OMAP Signals	71
Table 11.	SD/MMC OMAP Signals	78
Table 12.	OMAP3530 Audio Signals	79
Table 13.	OMAP3530 Audio Signals	80
Table 14.	OMAP3530 LCD Signals	82
Table 15.	TFP410 Interface Signals.....	83
Table 16.	J4 LCD Signals	87
Table 17.	J5 LCD Signals	88
Table 18.	S-Video Interface Signals	89
Table 19.	JTAG Signals	94
Table 20.	Expansion Connector Signals	96
Table 21.	Expansion Connector Signal Groups	97
Table 22.	J4 GPIO Signals	99
Table 23.	J5 GPIO Signals	99
Table 24.	DVI-D to HDMI Cable	103
Table 25.	J4 LCD Signals	105
Table 26.	J5 LCD Signals	106
Table 27.	J4 and J5 Connector Sources	106
Table 28.	Connector Dimensions.....	107
Table 29.	JTAG Signals	112
Table 30.	DC Power Supply Specifications.....	119
Table 31.	DC Power Supplies	119
Table 32.	Cable Pinout.....	121
Table 33.	USB Hubs Tested.....	121
Table 34.	DVI-D Monitors Tested.....	122

Table 35.	SD/MMC Cards Tested.....	122
Table 36.	USB to Ethernet Adapters.....	123
Table 37.	USB to WiFi Adapters	124
Table 38.	USB to Bluetooth Adapters	125
Table 39.	Voltages	150
Table 40.	Troubleshooting	152
Table 41.	Known Issues	157

1.0 Introduction

This document is the System Reference Manual for the BeagleBoard, a low cost OMAP3530 based board supported through BeagleBoard.org. This document provides detailed information on the overall design and usage of the BeagleBoard from the system level perspective. It is not intended to provide detailed documentation of the OMAP3530 processor or any other component used on the board. It is expected that the user will refer to the appropriate documents for these devices to access detailed information.

The key sections in this document are:

Section 2.0– Change History

Provides tracking for the changes made to the System Reference Manual.

Section 3.0– Definitions and References

This section provides definitions for commonly used terms and acronyms.

Section 4.0– Overview

This is a high level overview of the BeagleBoard.

Section 5.0– Specification

Provided here are the features and electrical specifications of the BeagleBoard.

Section 6.0–Product Contents

Describes what the BeagleBoard package looks like and what is included in the box.

Section 7.0– Hookup

Covered here is how to connect the various cables to the BeagleBoard.

Section 8.0– System Architecture and Design

This section provides information on the overall architecture and design of the BeagleBoard. This is a very detailed section that goes into the design of each circuit on the board.

Section 9.0– Connector Pinouts and Cables

The section describes each connector and cable used in the system. This will allow the user to create cables, purchase cables, or to perform debugging as needed.

Section 10.0– BeagleBoard Accessories

Covered in this section are a few of the accessories that may be used with BeagleBoard. This is not an exhaustive list, but does provide an idea of the types of cables and accessories that can be supported and how to find them. It also provides a definition of what they need to be. It does not guarantee that these devices will work on all OS implementations.

Section 11.0 – Mechanical

Information is provided here on the dimensions of the BeagleBoard.

Section 12.0 – Board Verification

A description is provided on how to setup the board and using the verification process and SW to verify that the board is functional.

Section 13.0 – Troubleshooting

Here is where you can find tips on troubleshooting the setup of the BeagleBoard.

Section 14.0- Known Issues

This section describes the known issues with the current revision of the BeagleBoard and any workarounds that may be possible.

Section 15.0- BeagleBoard Components

This section provides information on the top and bottom side silkscreen of the BeagleBoard showing the location of the components.

Section 16.0- BeagleBoard Schematics

These are the schematics for the BeagleBoard and information on where to get the PDF and OrCAD files..

Section 17.0- Bill Of Material

This section describes where to get the latest Bill of Material for the BeagleBoard.

Section 18.0- BeagleBoard PCB Information

This section describes where to get the PCB file information for the BeagleBoard.

2.0 Change History

2.1 Change History

Table 1 tracks the changes made for each revision of this document.

Table 1. Change History

Rev	Changes	Date	By
C4	Initial release.	12/15/2009	GC

2.2 Revision C3 vs. C4

There are three key changes on the Rev C4 board versus the Rev C3 version.

- Use of the OMAP3530DCBB72 device which is the 720MHZ version of the OMAP3530.
- An updated version of the UBoot software. The following changes will affect the user experience:
 - The Beagle splash screen has been replaced with an orange only screen at boot up.
 - Turning on VAUX2 for the EHCI fix
- A more advanced fix for the EHCI noise issue on Rev C3 board. This involves a change in the power circuitry for the 1.8V rail supplied to the EHCI PHY

interface. The power is now derived from the VAUX2 on the TPS65950 through a filter circuit.

3.0 Definitions and References

3.1 Definitions

SD- Secure Digital

SDIO- Secure Digital Input Output

MMC- Multimedia Card

MDDR- Mobile Dual Data Rate

SDRAM- Synchronous Dual Access Memory

OMAP3530- The CortexA8 based System on a Chip from Texas Instruments.

4.0 BeagleBoard Overview

The BeagleBoard is an OMAP3530 platform designed specifically to address the Open Source Community. It has been equipped with a minimum set of features to allow the user to experience the power of the OMAP3530 and is not intended as a full development platform as many of the features and interfaces supplied by the OMAP3530 are not accessible from the BeagleBoard. By utilizing standard interfaces, the BeagleBoard is highly extensible to add many features and interfaces. It is not intended for use in end products. All of the design information is freely available and can be used as the basis for a product. BeagleBoards will not be sold for use in any product as this hampers the ability to get the boards to as many community members as possible and to grow the community.

4.1 BeagleBoard Usage Scenarios

The **Figure 1** provides an example of a few of the various usage scenarios for the BeagleBoard.

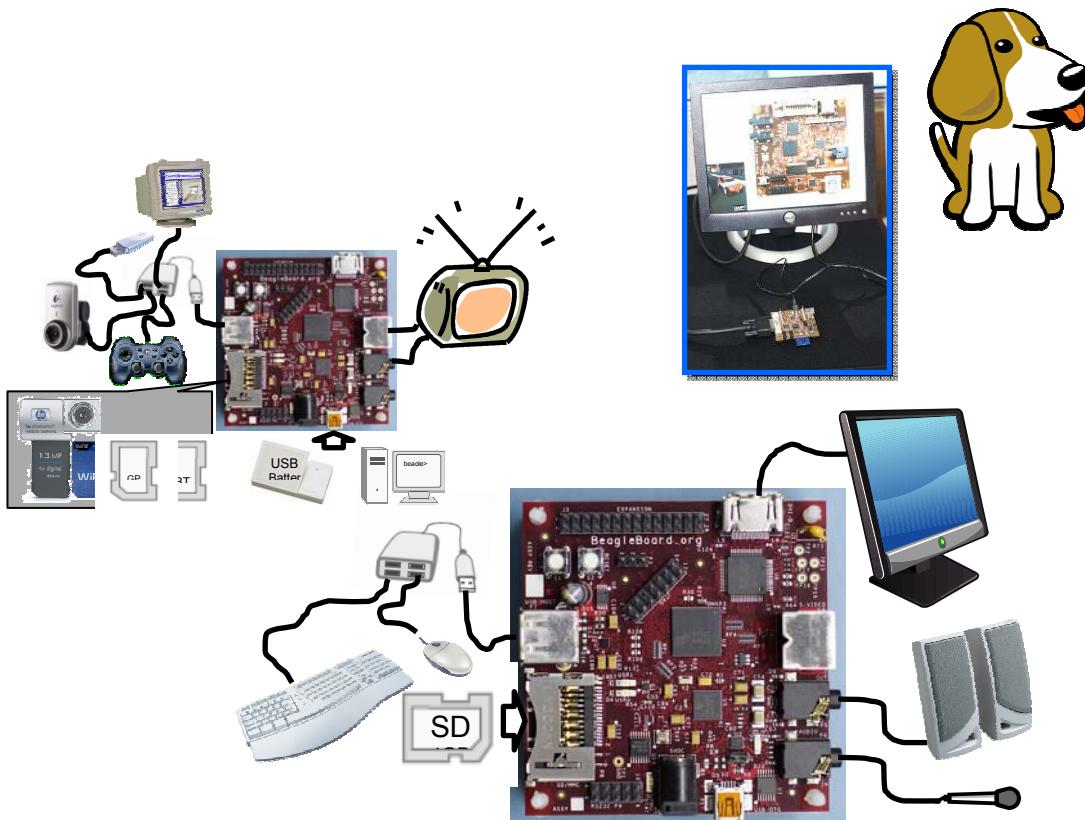


Figure 1. BeagleBoard Usage Scenarios

5.0 BeagleBoard Specification

This section covers the specifications of the BeagleBoard and provides a high level description of the major components and interfaces that make up the BeagleBoard.

5.1 BeagleBoard Features

Table 2 provides a list of the BeagleBoard's features.

Table 2. BeagleBoard Features

	Feature	
Processor	OMAP3530DCBB72 720MHz	
POP Memory	Micron	
	2Gb NAND (256MB)	2Gb MDDR SDRAM (256MB)
PMIC TPS65950	Power Regulators	
	Audio CODEC	
	Reset	
	USB OTG PHY	
Debug Support	14-pin JTAG	GPIO Pins
	UART	LEDs
PCB	3.1" x 3.0" (78.74 x 76.2mm)	6 layers
Indicators	Power	2-User Controllable
	PMU	
HS USB 2.0 OTG Port	Mini AB USB connector	
	TPS65950 I/F	
	MiniAB	
HS USB Host Port	Single USB HS Port	Up to 500ma Power
Audio Connectors	3.5mm	3.5mm
	L+R out	L+R Stereo In
SD/MMC Connector	6 in 1 SD/MMC/SDIO	4/8 bit support, Dual voltage
User Interface	1-User defined button	Reset Button
Video	DVI-D	S-Video
Power Connector	USB Power	DC Power
Expansion Connector (Not Populated)	Power (5V & 1.8V)	UART
	McBSP	McSPI
	I2C	GPIO
	MMC	PWM
2 LCD Connectors	Access to all of the LCD control signals plus I2C	
	3.3V, 5V, 1.8V	

The following sections provide more detail on each feature and sections of the BeagleBoard.

5.2 OMAP Processor

The BeagleBoard uses the OMAP3530DCBB72 720MHZ version and comes in a .4mm pitch POP package. POP (Package on Package) is a technique where the memory, NAND and SDRAM, are mounted on top of the OMAP3530. For this reason, when looking at the BeagleBoard, you will not find an actual part labeled OMAP3530.

5.3 Memory

The Micron POP memory is used on the Rev C4 BeagleBoard and is mounted on top of the processor as mentioned. The key function of the POP memory is to provide:

- 2Gb NAND x 16 (256MB)
- 2Gb MDDR SDRAM x32 (256MB @ 166MHz)

No other memory devices are on the BeagleBoard. It is possible however, that additional memory can be added to BeagleBoard by:

- Installing a SD or MMC in the SD/MMC slot
- Use the USB OTG port and a powered USB hub to drive a USB Thumb drive or hard drive.
- Install a thumbdrive into the EHCI USB port

Support for this is dependent upon driver support in the OS.

5.4 Power Management

The TPS65950 is used on the Rev C4 to provide power to the BeagleBoard with the exception of the 3.3V regulator which is used to provide power to the DVI-D encoder and RS232 driver. In addition to the power the TPS65950 also provides:

- Stereo Audio Out
- Stereo Audio in
- Power on reset
- USB OTG PHY
- Status LED

5.5 HS USB 2.0 OTG Port

The USB OTG port can be used as the primary power source and communication link for the BeagleBoard and derives power from the PC over the USB cable. The client port is limited in most cases to 500mA by the PC. A single PC USB port is sufficient to power the BeagleBoard. If additional devices are connected to the expansion bus and the 5V rail is used to power them or if a high powered USB device is connected to the EHCI port, then the power required could exceed that supplied by a USB port or Hub.

It is possible to take this to 1A by using a Y cable if additional power is needed for either the USB host port or an expansion card. **Figure 2** shows an example of the Y-Cable for the USB.

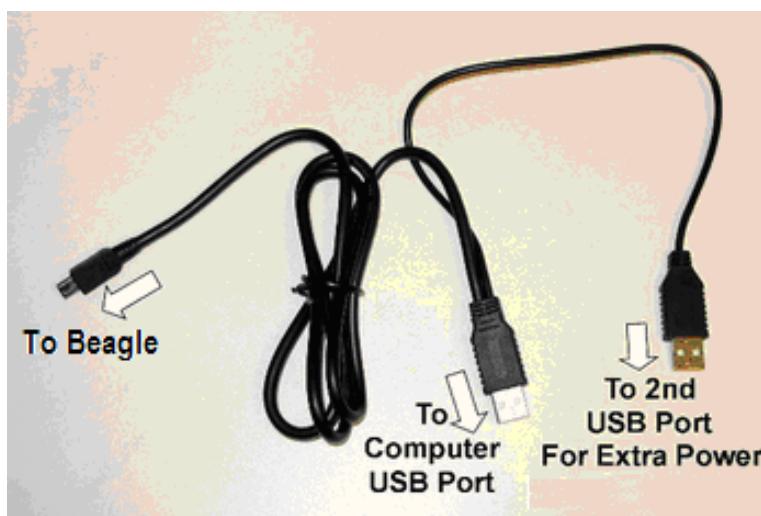


Figure 2. USB Y-Cable

The BeagleBoard requires a single minAB to USB A cable or as mentioned a Y-Cable can be used if needed.

There is an option to provide external power to the BeagleBoard using a 5V DC supply and is discussed later in this section.

5.6 HS USB 2.0 Host Port

On the Rev C4 board a single USB HS only Host port is provided via a USB Type A connector. It provides power on/off control and up to 500mA of current at 5V.

The HS USB Port is HS only. In order to support a FS/LS device, a HUB must be used.

5.7 Stereo Audio Output Connector

A 3.5mm standard stereo output audio jack is provided to access the stereo output of the onboard audio CODEC. The Audio CODEC is provided by the TPS65950.

5.8 Stereo Audio In Connector

A 3.5mm standard stereo audio input jack is provided to access the stereo output of the onboard audio CODEC.

5.9 S-Video Connector

A 4 pin DIN connector is provided to access the S-Video output of the BeagleBoard. This is a separate output from the OMAP processor and can contain different video output data from what is found on the DVI-D output if the software is configured to do it.

It will support NTSC or PAL format output to a standard TV. The default is NTSC, but can be changed via the Software.

5.10 DVI-D Connector

The BeagleBoard can drive a LCD panel equipped with a DVI-D digital input. This is the standard LCD panel interface of the OMAP3530 and will support 24b color output. DDC2B (Display Data Channel) or EDID (Enhanced Display ID) support over I2C is provided in order to allow for the identification of the LCD monitor type and the required settings.

The BeagleBoard is equipped with a DVI-D interface that uses an HDMI connector that was selected for its small size. [It does not support the full HDMI interface and is used to provide the DVI-D interface portion only.](#) The user must use a HDMI to DVI-D cable or adapter to connect to a LCD monitor. This cable or adapter is not provided with the BeagleBoard. A standard HDMI cable can be used when connecting to a monitor with an HDMI connector.

DO NOT PLUG IN THE DVI-D CONNECTOR TO A DISPLAY WITH THE BEAGLEBOARD POWERED ON. PLUG IN THE CABLE TO THE DISPLAY AND THEN POWER ON THE BEAGLEBOARD.

5.11 LCD Header

A pair of 1.27mm pitch 2x10 headers are provided to gain access to the LCD signals. This allows for the creation of LCD boards that will allow adapters to be made to provide the level translation to support different LCD panels.

5.12 SD/MMC 6 in 1 Connector

A 6 in 1 SD/MMC connector is provided as a means for expansion and can support such devices as:

- WiFi Cards
- Camera
- Bluetooth Cards
- GPS Modules
- SD Memory Cards
- MMC Memory Cards
- SDIO Cards
- MMCMobile cards
- RS-MMC Cards
- miniSD Cards

It supports the MMC4.0 (MMC+) standard and can boot from MMC or SD cards. It will support both 4 and 8 bit cards. It will also support most SDHC cards as well.

5.13 Reset Button

When pressed and released, causes a power on reset of the BeagleBoard.

5.14 User/Boot Button

A button is provided on the BeagleBoard to provide two functions:

- Force a change in the boot sequence of the OMAP3530.
- Used as an application button that can be used by SW as needed.

When used in conjunction with the RESET button, it will force a change to the order in which boot sources are checked as viable boot sources.

If the button is pressed while the RESET button is released, the sequence becomes:

- USB
- UART
- MMC1
- NAND

Even though the NAND may have a program in it, if a card is placed in the MMC slot, it will try to boot from it first. If it is not there, it will boot from NAND.

There is also the option to have a serial download application that will program the NAND if connected to the serial or USB ports. In this scenario the internal ROM will

stop on either the serial or USB port and start the download process from there. It does require an application to be run on the host PC in order to perform this function.

If the user button is not pressed at reset, the sequence in which the internal ROM looks for viable boot sources is as follows:

- NAND
- USB
- UART3
- MMC1

In this case, NAND overrides every option and will always boot from NAND if there is data in the NAND. If the NAND is empty, then the other sources are available to be used based on the boot order.

To force a boot from the SD/MMC card, the reset button must be pushed and the reset button pushed and released.

5.15 Indicators

There are three green LEDs on the BeagleBoard that can be controlled by the user.

- One on the TPS65950 that is programmed via the I2C interface
- Two on the OMAP3530 Processor controlled via GPIO pins

There is a fourth LED on the BeagleBoard that provides an indication that power is supplied to the board and is not controlled via software.

5.16 Power Connector

Power will be supplied via the USB OTG connector and if a need arises for additional power, such as when a board is added to the expansion connectors, a larger wall supply 5V can be plugged into the optional power jack. When the wall supply is plugged in, it will remove the power path from the USB connector and will be the power source for the whole board. The power supply is not provided with the BeagleBoard.

When using the USB OTG port in the host mode, the DC supply must be connected as the USB port will be used to provide limited power to the hub at a maximum of 100mA, so a hub must be powered. The 100mA is not impacted by having a higher amperage supply plugged into the DC power jack. The 100mA is a function of the OTG port itself.

WARNING: DO NOT PLUG IN ANYTHING BUT 5V TO THE DC CONNECTOR OR THE BOARD WILL BE DAMAGED!

Make sure the DC supply is regulated and a clean supply.

5.17 JTAG Connector

A 14 pin JTAG header is provided on the BeagleBoard to facilitate the SW development and debugging of the board by using various JTAG emulators. The interface is at 1.8V on all signals. Only 1.8V Levels are supported. **DO NOT expose the JTAG header to 3.3V.**

5.18 RS232 Header

Support for RS232 via UART3 is provided by a 10 pin header on the BeagleBoard for access to an onboard RS232 transceiver. It does require an IDC to DB9 flat cable, which is not provided, to access the serial port.

5.19 Expansion Header

An option for a single 28 pin header is provided on the board to allow for the connection of various expansion cards that could be developed by the users or other sources. Due to multiplexing, different signals can be provided on each pin. This header is not populated on the BeagleBoard so that based on the usage scenario it can be populated as needed.

5.20 BeagleBoard Mechanical Specifications

Size:	3.0" x 3.1"
Max height:	TBM
Layers:	6
PCB thickness:	.062"
RoHS Compliant:	Yes
Weight:	TBW

5.21 Electrical Specifications

Table 3 is the electrical specification of the external interfaces to the Rev C4 BeagleBoard.

Table 3. BeagleBoard Electrical Specification Rev C4

Specification	Min	Typ	Max	Unit
Power				
Input Voltage USB		5	5.2	V
Current USB		350		mA
Input Voltage DC	4.8	5	5.2	V
Current DC		350		mA
Expansion Voltage (5V)	4.8	5	5.2	V
Current (Depends on source current available)		1		A
Expansion Voltage (1.8V)	1.75	1.8	1.85	V
Current			30	mA
USB Host (Same as the DC supplied by the power plug or USB 5V)	4.8	5	5.2	V
Current (Depends on what the DC source can supply over what the board requires)		Varies		
USB OTG				
High Speed Mode			480	Mb/S
Full Speed Mode			12.5	Mb/S
Low Speed Mode			1.5	Mb/S
USB Host				
High Speed Mode			480	Mb/S
RS232				
Transmit				
High Level Output Voltage		5	5.4	V
Low Level output voltage		-5	-5.5	V
Output impedance		+/-35	+/-60	mA
Maximum data rate	250			Kbit/S
Receive				
High level Input Voltage	-2.7	-3.2		V
Lo Level Input Voltage			.4	
Input resistance	3	5	7	Kohms
JTAG				
Realview ICE Tool			30	MHz
XDS560			30	MHz
XDS510			30	MHz
Lauterbach(tm)			30	MHz
SD/MMC				
Voltage Mode 1.8V	1.71	1.8	1.89	V
Voltage Mode 3.0V	2.7	3.0		V
Current			220	mA
Clock			48	MHz
DVI-D				
Pixel Clock Frequency	25		65	MHz
High level output voltage		3.3		V
Swing output voltage	400		600	mVp-p
Maximum resolution			1024 x 768	

S-Video				
Full scale output voltage (75ohm load)	.7	.88	1	V
Offset voltage		50		mV
Output Impedance	67.5	75	82.5	Ohms
Audio In				
Peak-to-peak single-ended input voltage (0 dBFs)			1.5	Vpp
Total harmonic distortion (sine wave @ 1.02 kHz @ -1 dBFs)		-80	-75	dB
Total harmonic distortion (sine wave @ 1.02 kHz) 2 0 Hz to 20 kHz, A-weighted audio, Gain = 0 dB		-85	-78	dB
Audio Out				
Load Impedance @100 pF	14	16		ohms
Maximum Output Power (At 0.53 Vrms differential output voltage and load impedance = 16 Ohms)		17.56		mW
Peak-to-Peak output voltage			1.5	Vpp
Total Harmonic Distortion @ 0 dBFs		-80	-75	dB
Idle channel noise (20Hz to 20KHz)		-90	-85	dB

6.0 Product Contents

Under this section is a description of what comes in the box when the BeagleBoard is purchased.

6.1 BeagleBoard In the Box Rev C4

The final packaged Rev C4 product will contain the following:

- 1 Box
- 1 BeagleBoard in an ESD Bag

NO CABLES ARE PROVIDED WITH THE BEAGLEBOARD.

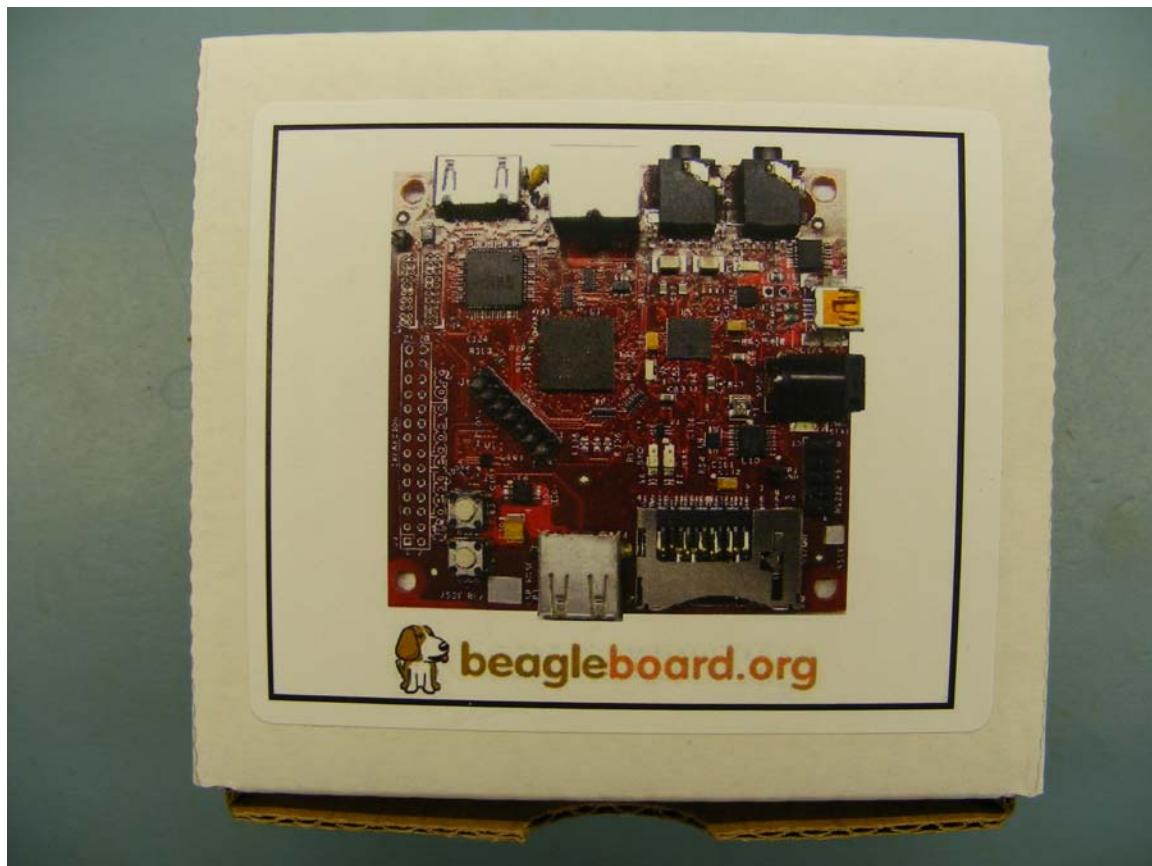


Figure 3. The Rev C4 Box



Figure 4. Rev C4 Box Contents

6.2 Software on the BeagleBoard

The board ships with U-Boot and X-Loader flashed onto the BeagleBoard.

6.3 Repair

If you feel the board is in need of repair, follow the RMA Request process found at
<http://beagleboard.org/support/rma>

7.0 BeagleBoard Hookup

This section provides an overview of all of the connectors on the BeagleBoard and how they should be used.

7.1 Connecting USB OTG

The USB OTG port connects to the PC host and uses a miniAB cable through which power is provided to the BeagleBoard. If desired, the BeagleBoard may also be connected to a self powered USB hub. **Figure 5** shows where the cable is connected to the BeagleBoard.

If the OTG Port is to be used as a Host, the ID pin must be grounded. This means that you must have a 5 pin cable connected to the OTG port on the BeagleBoard and you must use a USB powered HUB. There is also an option to ground the ID on the board and is discussed later.

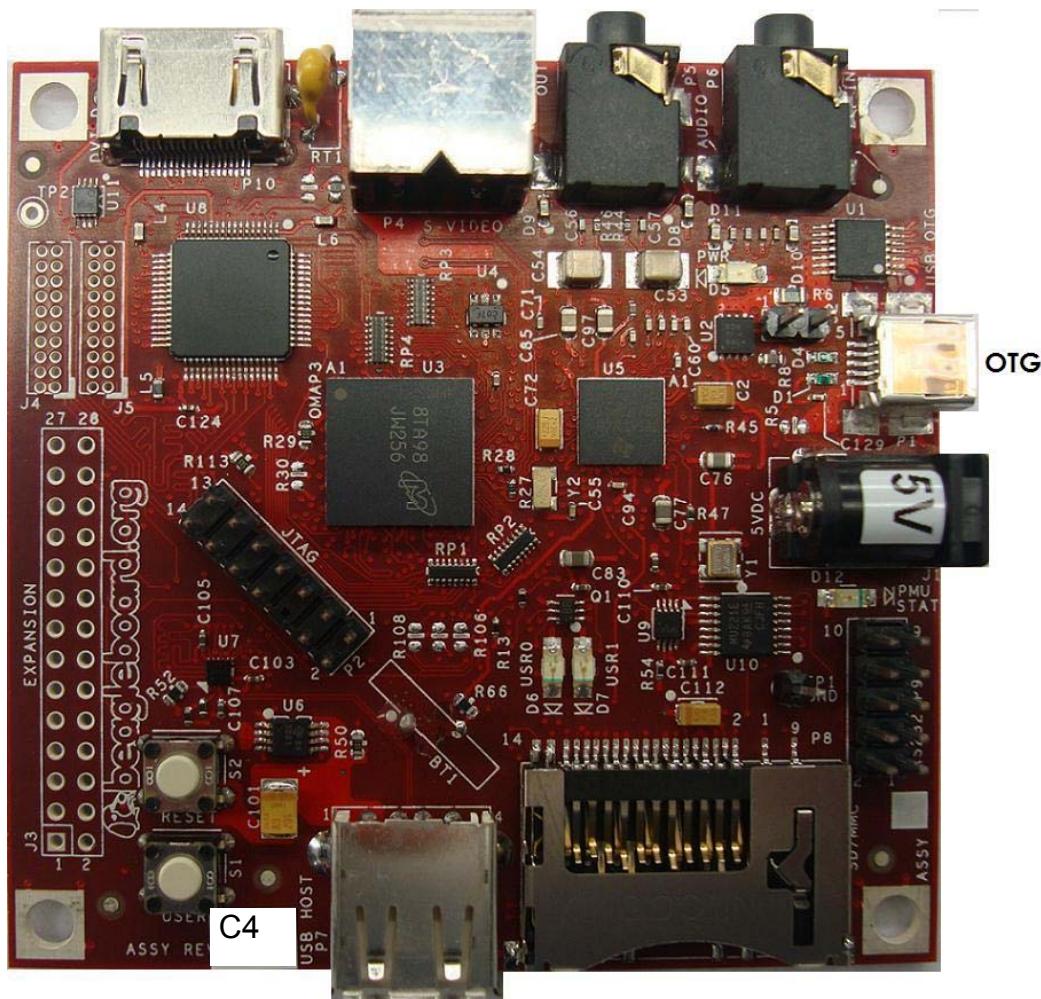


Figure 5. USB OTG Connection

7.2 Connecting USB Host

The Rev C4 Beagle is equipped with a USB Host only connector and can be used to support USB based devices. In order to connect multiple devices a Hub is required. The hub can be powered or un-powered if the total current on the devices connected to the hub do not exceed the available power from the DC power supply that is used. If the board is powered from the OTG connector, then the power available from this port is extremely limited and will not be able to provide sufficient power to run most USB devices. It may be possible to run a USB keyboard or mouse, but that is about all it will have the power to supply. The USB Host port is HS only and does not support LS or FS devices without a hub.

Figure 6 below shows the location of the USB Host connector.

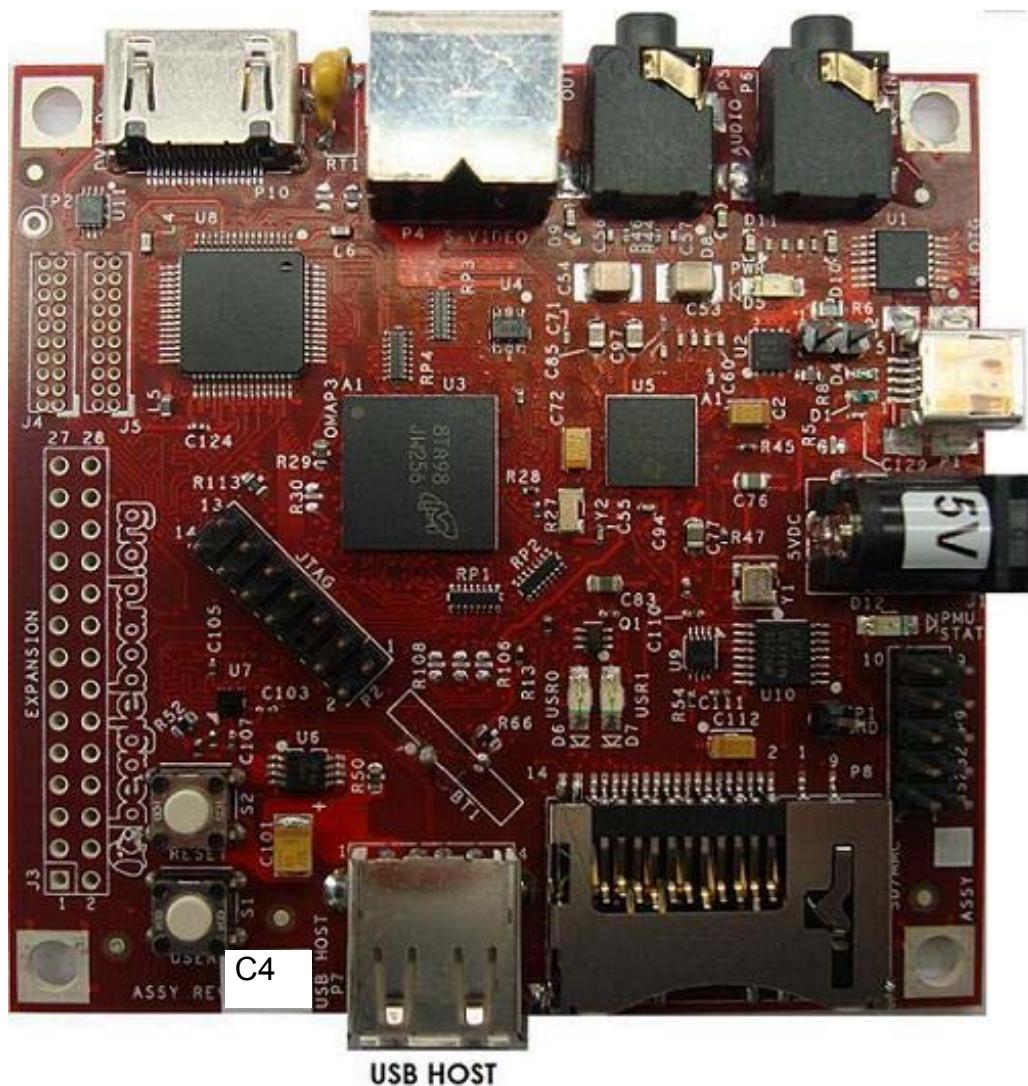
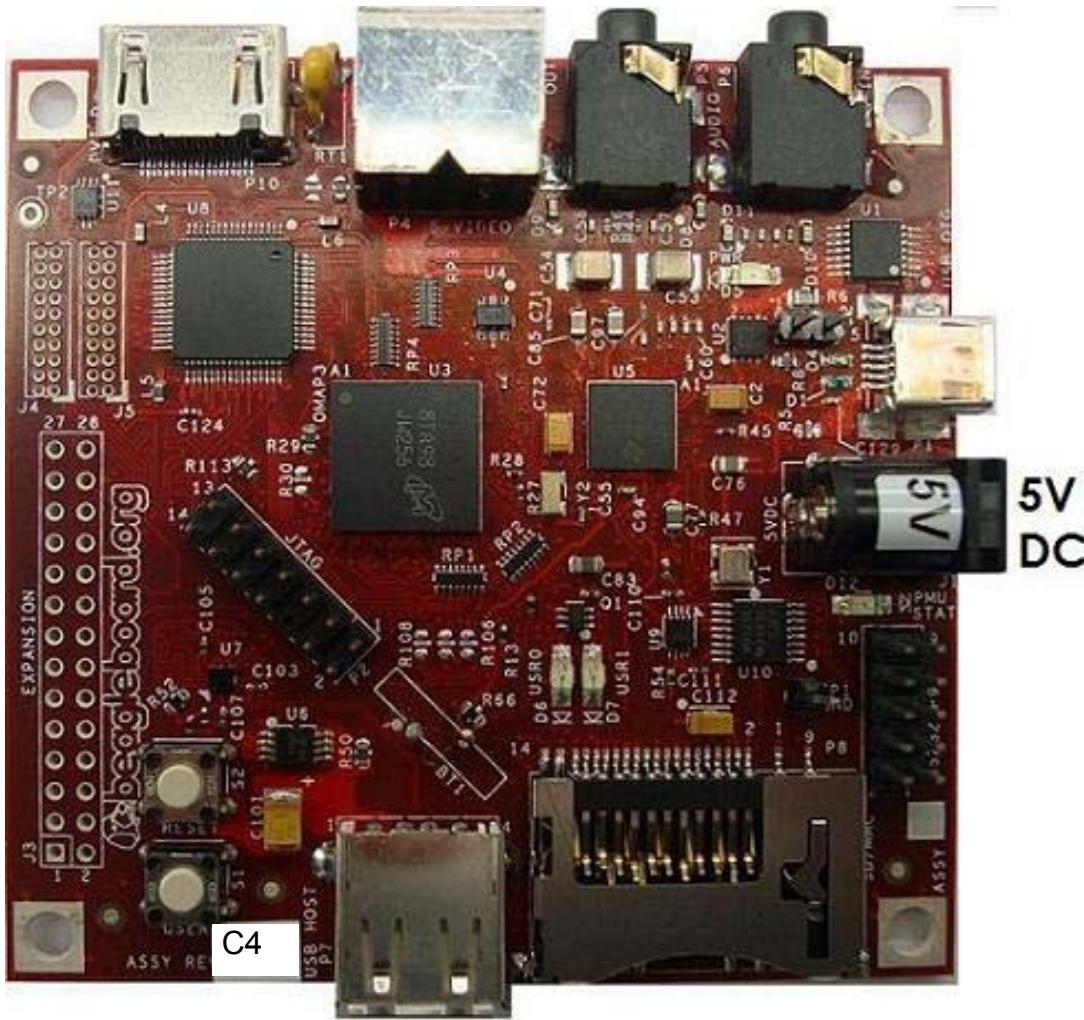


Figure 6. USB Host Connection

7.3 Connecting Optional Power

An optional DC supply can be used to power the BeagleBoard by plugging it into the power jack of the BeagleBoard. The power supply is not provided with the BeagleBoard, but can be obtained from various sources. You need to make sure the supply is a regulated 5V supply. **Figure 7** shows where to insert the power supply into the power jack.



7.4 Connecting JTAG

A JTAG emulator can be used for advanced debugging by connecting it to the JTAG header on the BeagleBoard. Only the 14pin version of the JTAG is supported and if a 20pin version is needed, you will need to contact your emulator supplier for the appropriate adapter. **Figure 8** shows the connection of the JTAG cable to the BeagleBoard.

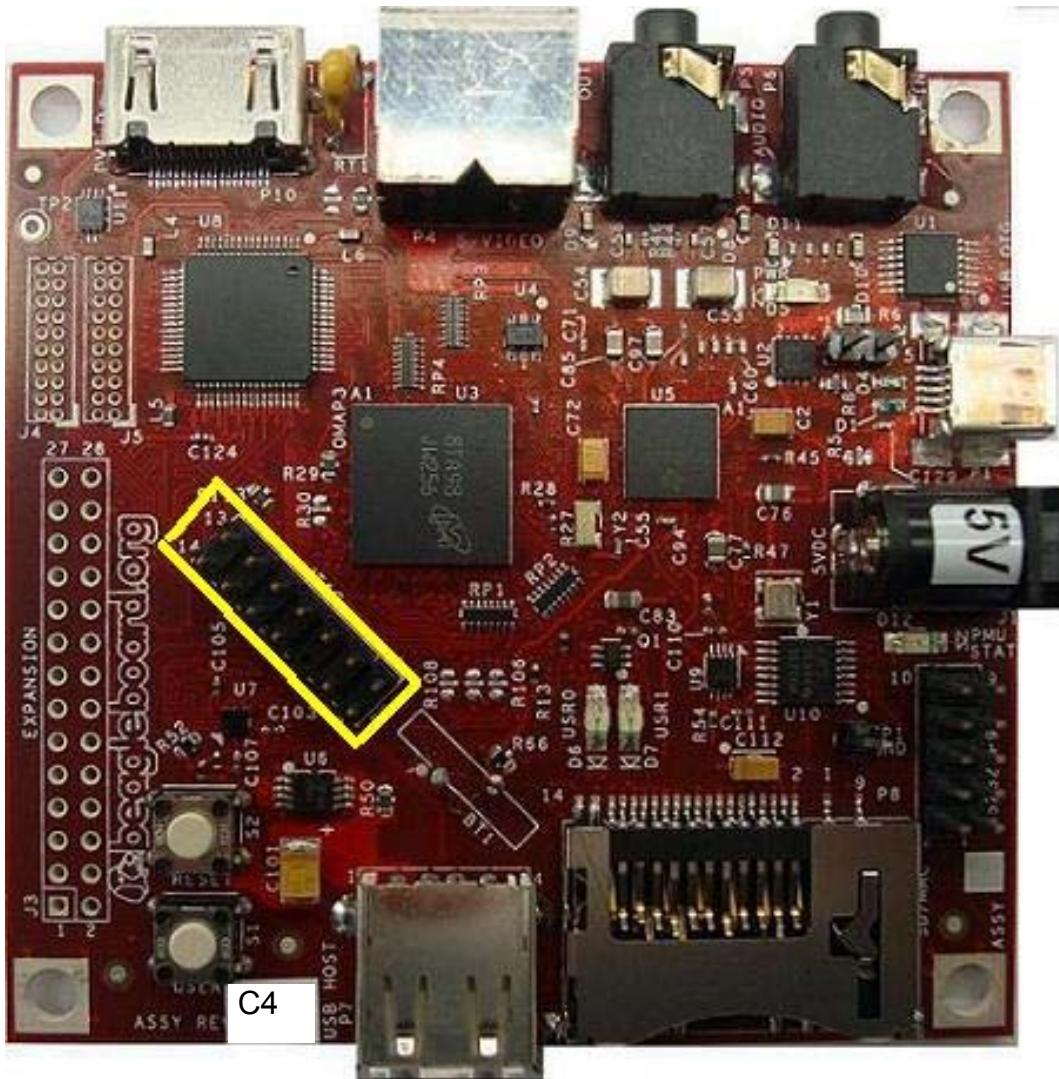


Figure 8. BeagleBoard JTAG Connection

DO NOT expose the JTAG header to 3.3V. It supports 1.8V only.

7.5 Connecting Serial Cable

In order to access the serial port of the BeagleBoard a flat cable is required to connect to a PC. The adapter will not plug directly into the PC and will require an external Female to Female twisted cable (Null Modem) in order to connect it to the PC. The ribbon cable is not supplied with the BeagleBoard but can be obtained from numerous sources. **Figure 9** shows where the ribbon cable is to be installed.

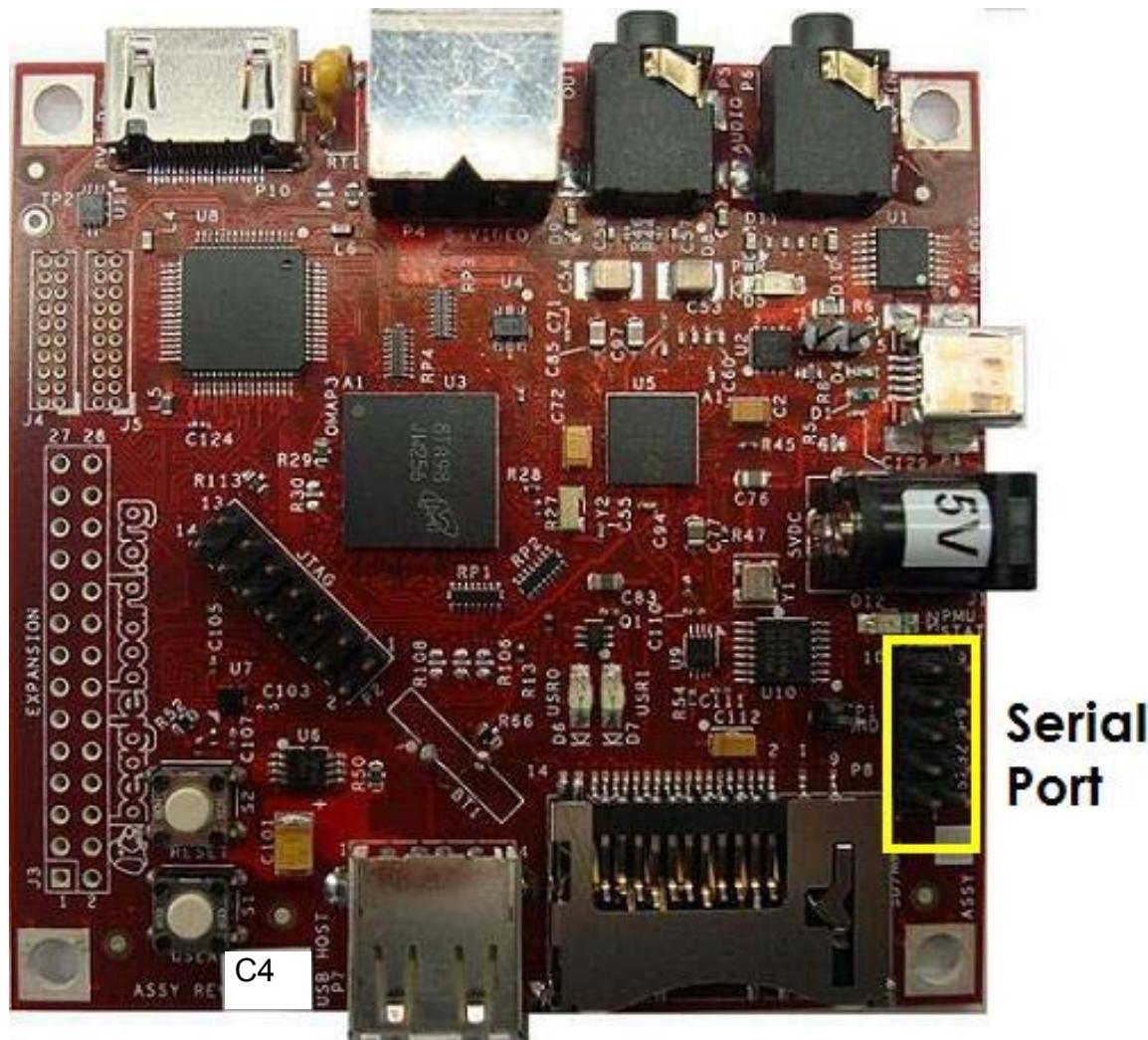


Figure 9. BeagleBoard Serial Cable Connection

7.6 Connecting S-Video

An S-Video cable can be connected to the BeagleBoard and from there is can be connected to a TV or monitor that supports an S-Video input. This cable is not supplied with the BeagleBoard. **Figure 10** shows the connector for the S-Video cable.

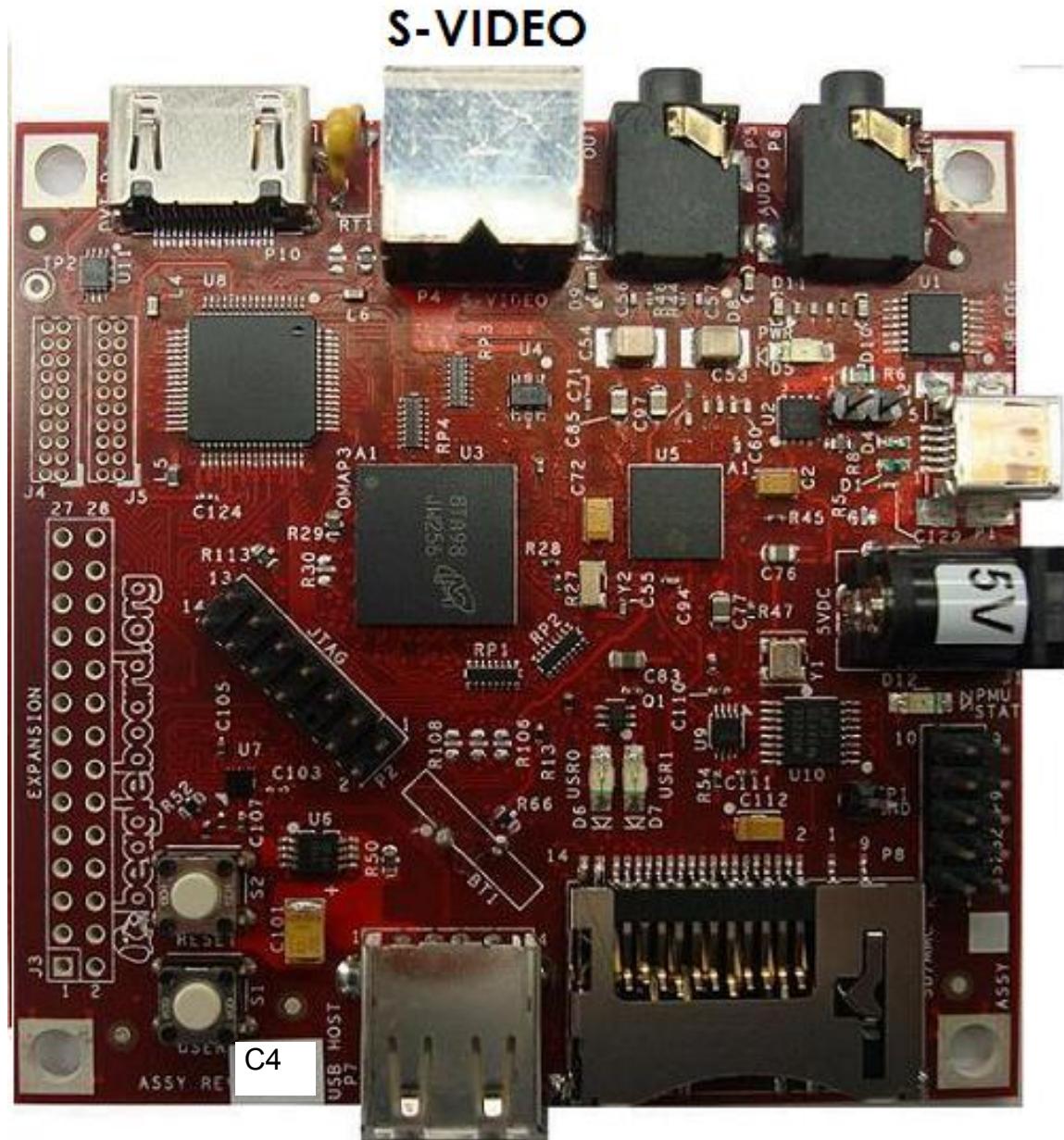


Figure 10. BeagleBoard S-Video Connection

7.7 Connecting DVI-D Cable

In order to connect the DVI-D output to a monitor, a HDMI to DVI-D cable is required. This cable is not supplied with BeagleBoard but can be obtained through numerous sources. **Figure 11** shows the proper connection point for the cable.

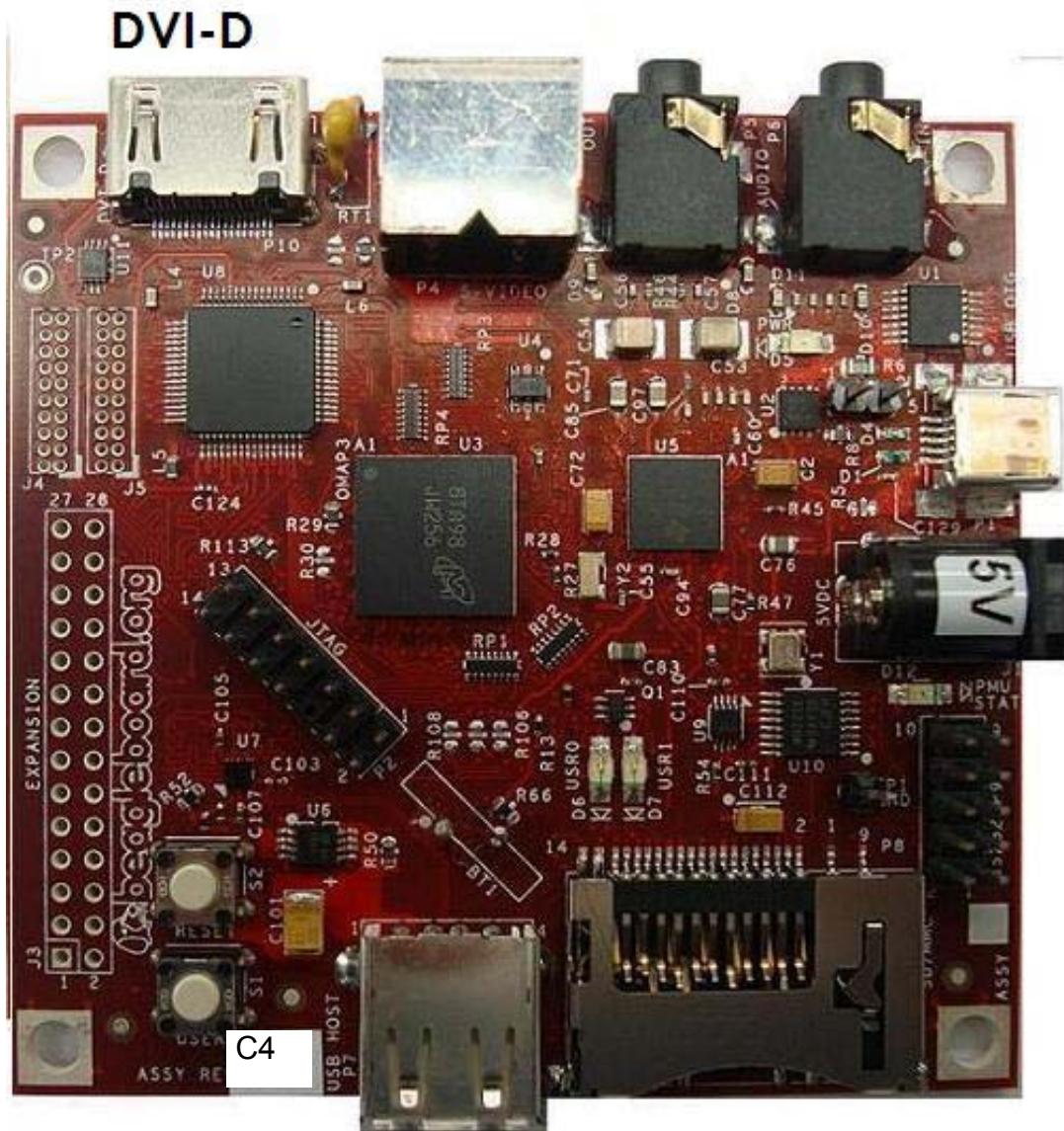


Figure 11. BeagleBoard DVI-D Connection

DO NOT PLUG IN THE DVI-D CONNECTOR TO A DISPLAY WITH THE BEAGLEBOARD POWERED ON. PLUG IN THE CABLE TO THE DISPLAY AND THEN POWER ON THE BEAGLEBOARD.

Only the digital portion of HDMI is supported on the BeagleBoard.

7.8 Connecting Stereo Out Cable

An external Audio output device, such as external stereo powered speakers, can be connected to the BeagleBoard via a 3.5mm jack. The audio cables are not provided with BeagleBoard, but can be obtained from just about anywhere. **Figure 12** shows how the cable connected to the stereo out jack.



Figure 12. BeagleBoard Audio Out Cable Connection

7.9 Connecting Stereo In Cable

External Audio input devices, such as a powered microphone or the audio output of a PC or MP3 player, can be connected to the via a 3.5mm jack. The audio cables are not provided with BeagleBoard, but can be obtained from just about any source. **Figure 13** shows how the cable is connected to the stereo input jack.

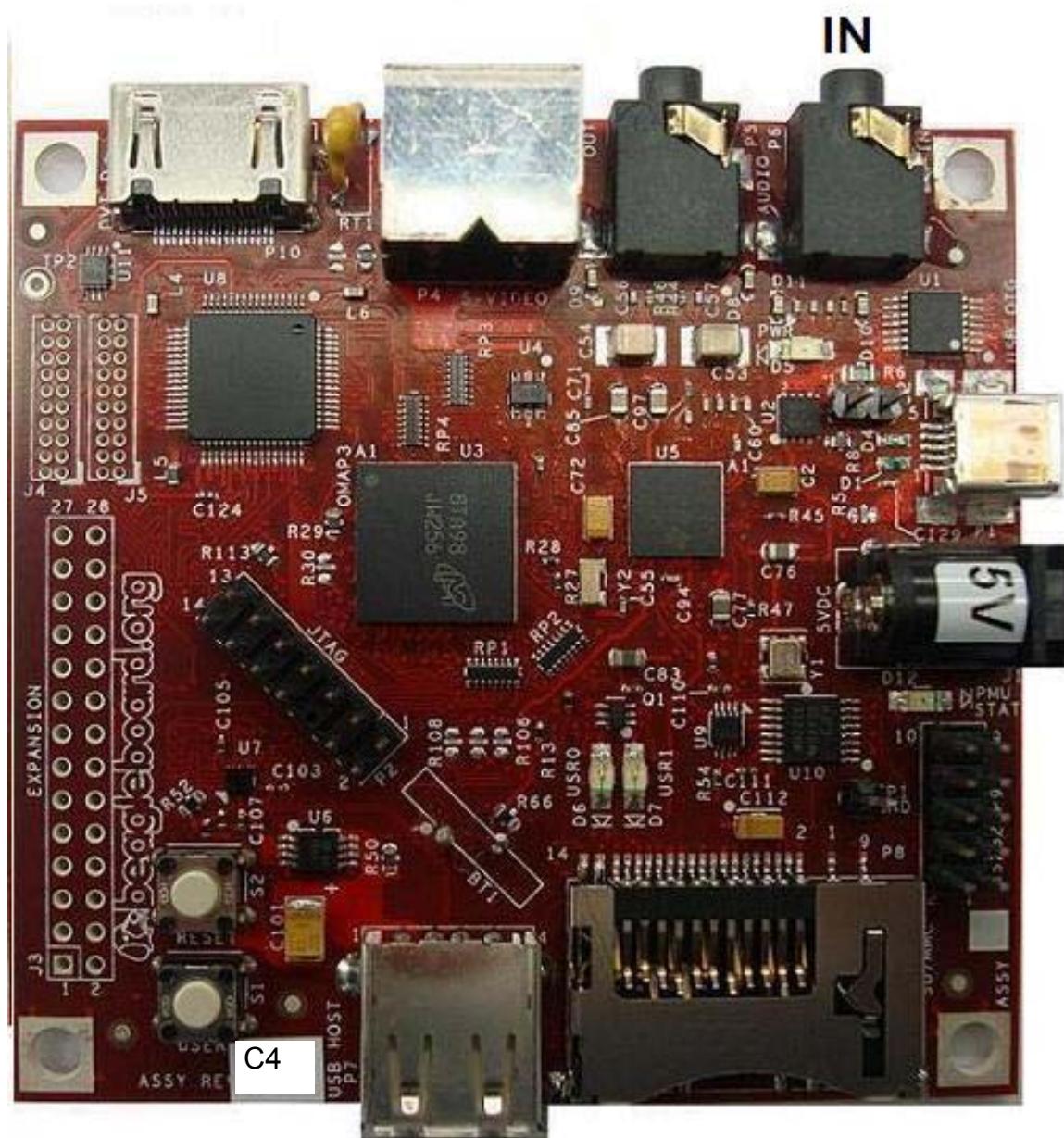


Figure 13. BeagleBoard Audio In Cable Connection

7.10 Indicator Locations

There are four green indicators on the BeagleBoard. One of them, POWER, indicates that the main supply is active. The other three can be controlled by the software. **Figure 14** shows the location of each indicator.

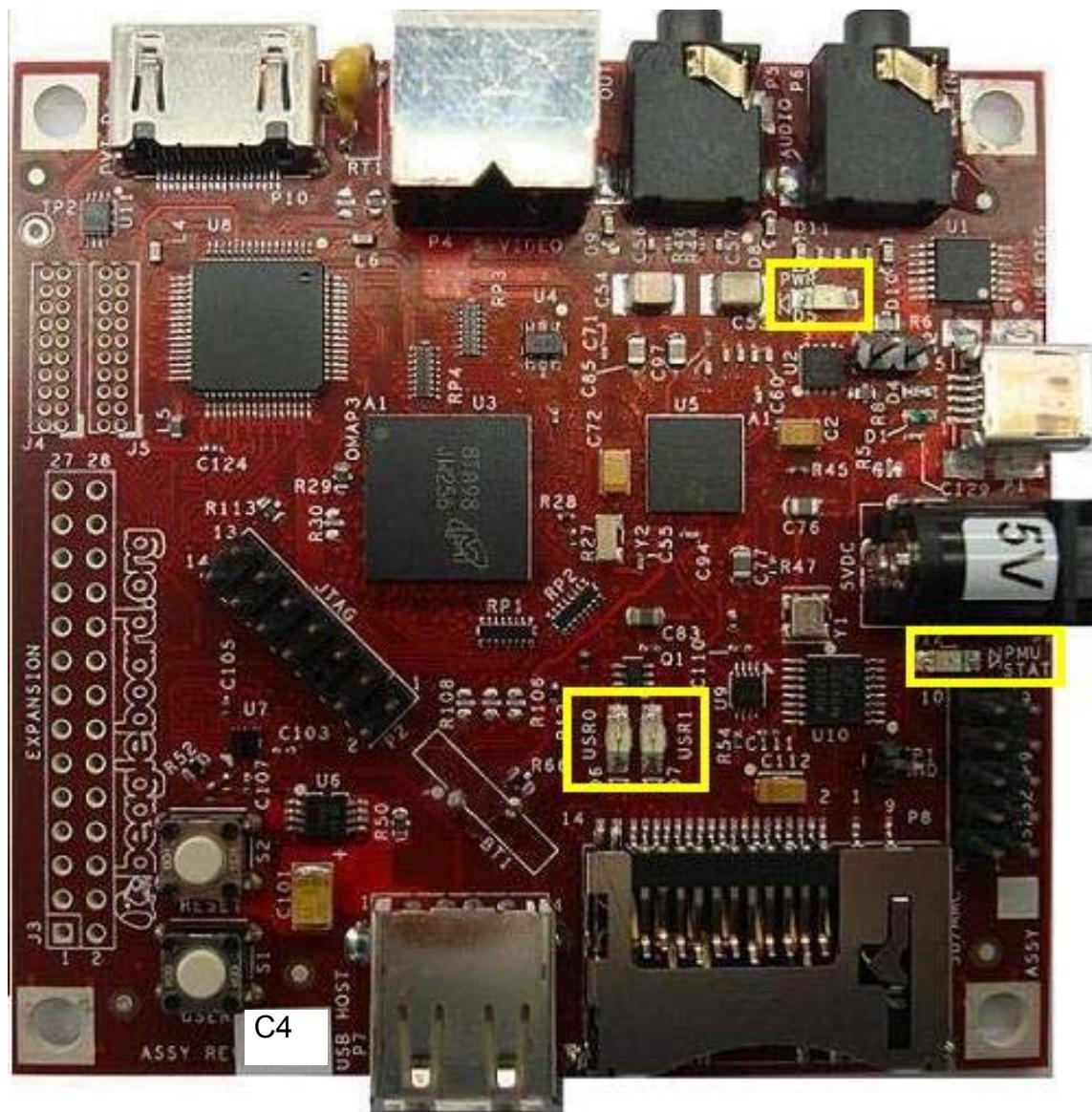


Figure 14. BeagleBoard Indicator Locations

7.11 Button Locations

There are two buttons on the BeagleBoard; the **RESET** button when pressed will force a board reset and the **USER** button which can be used by the SW for user interaction. If the user holds the **USER** button down while pressing and releasing the **RESET** button, the BeagleBoard will enter the ROM boot loader mode allowing it to boot from other sources than the onboard NAND. **Figure 15** shows the location of the buttons.

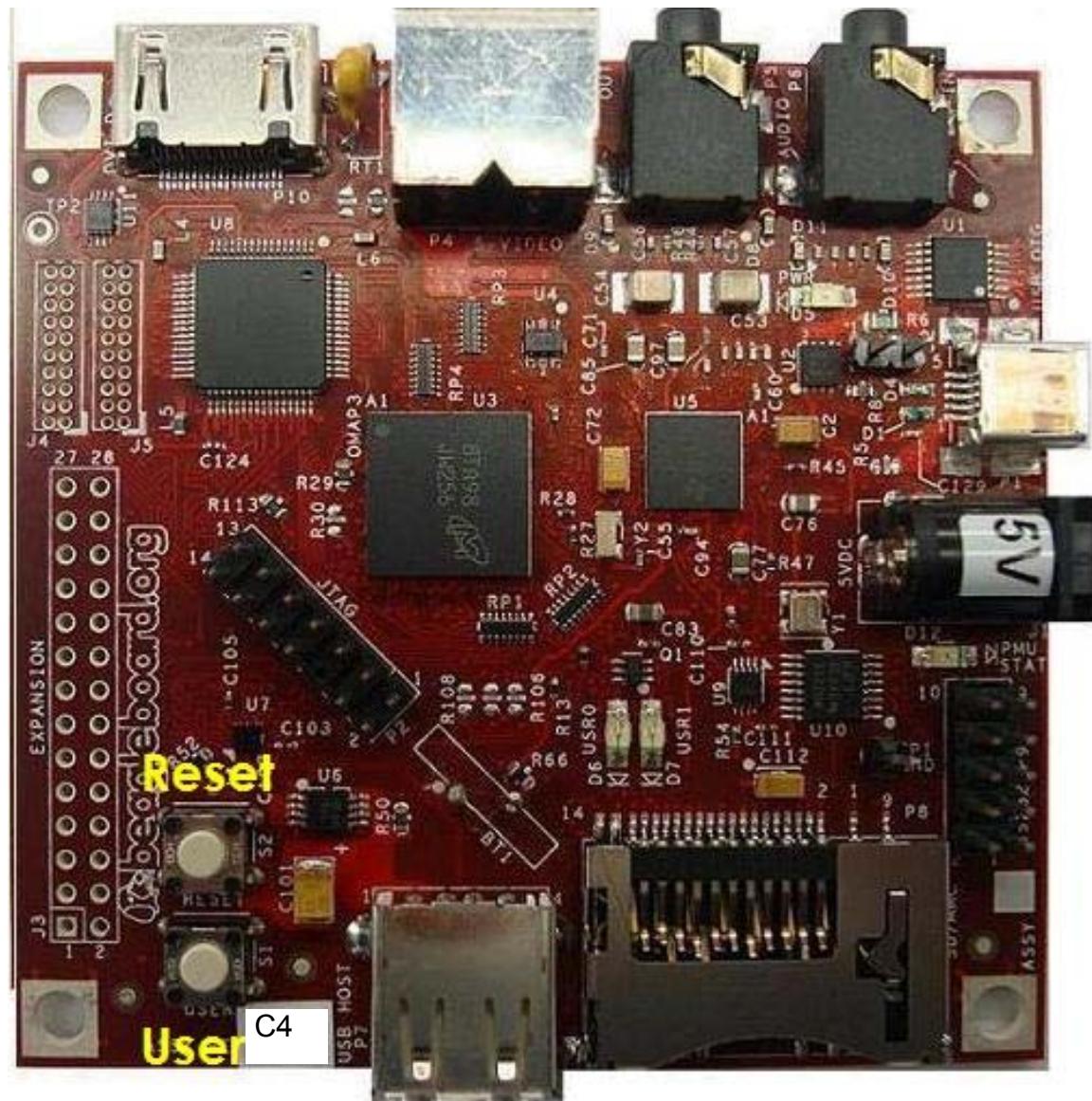


Figure 15. BeagleBoard Button Location

7.12 SD/MMC Connection

The SD/MMC connector can be used for Memory or SDIO type cards. This is a full size connector and will support various cards. Whether a particular card is supported or not, is dependent on the available software drivers. **Figure 16** shows the location of the SD/MMC connector.

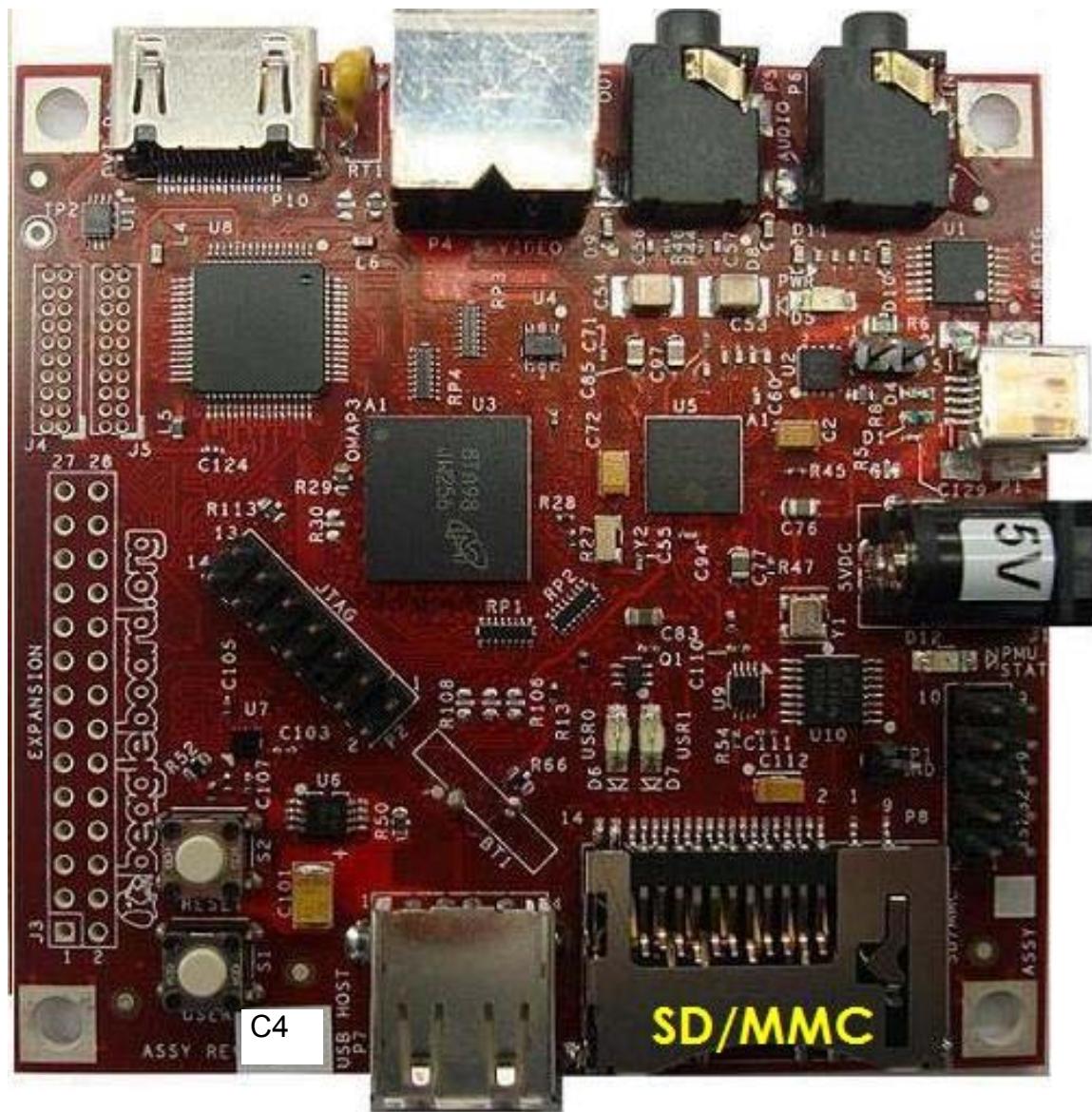


Figure 16. BeagleBoard SD/MMC Location

7.13 LCD Connection

There are two headers provided to allow access to the LCD signals on the BeagleBoard. These headers are 2x10 headers with a spacing of .05 (1.27mm) pitch. How these connectors are used is determined by the design of the adapter board. **Figure 17** shows the location of the LCD headers on the Beagle.

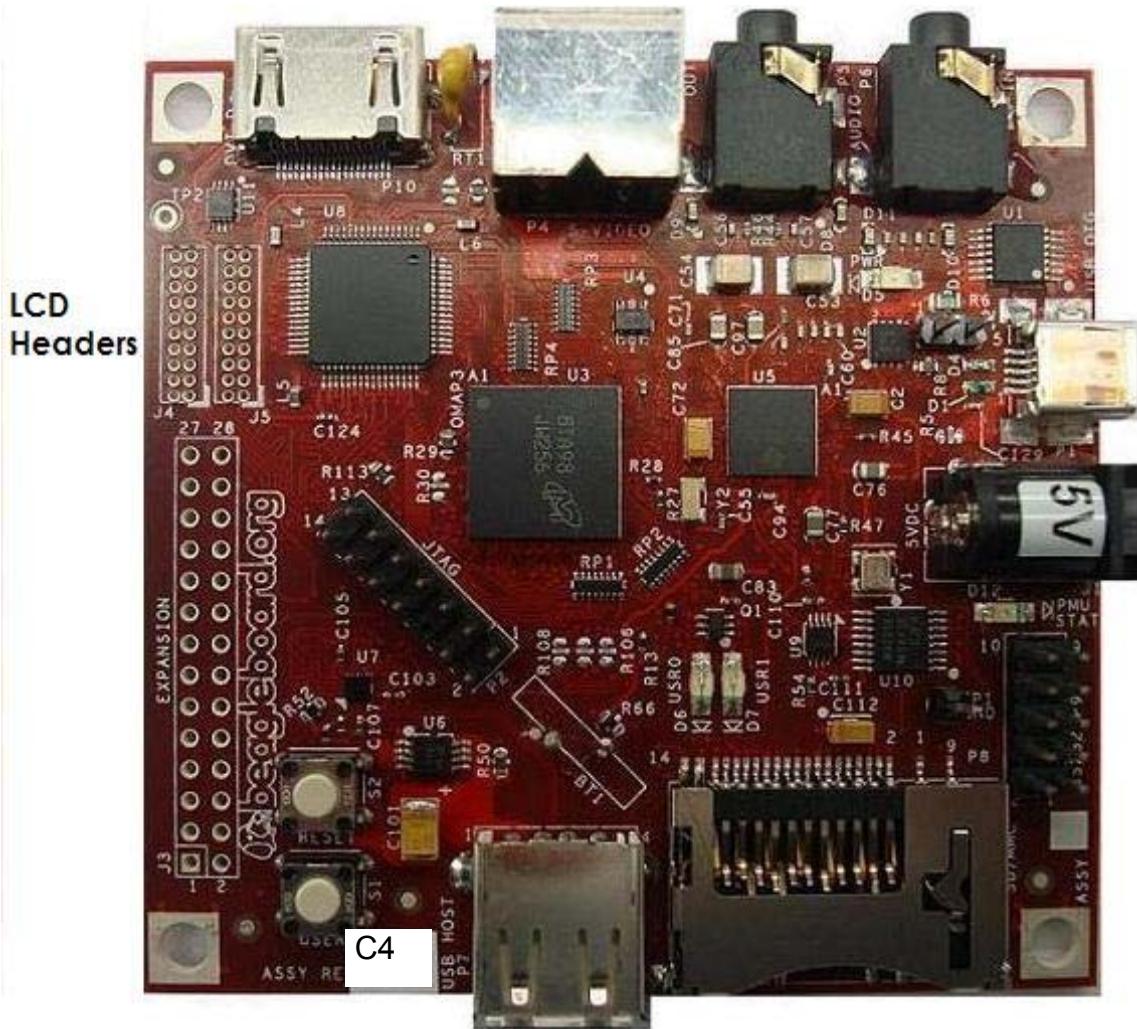


Figure 17. BeagleBoard LCD Header Location

8.0 BeagleBoard System Architecture and Design

This section provides a high level description of the design of the BeagleBoard and its overall architecture.

8.1 System Block Diagram

Figure 18 is the high level block diagram of the BeagleBoard. If you will notice, the block diagram is configured to match the component placement of the BeagleBoard.

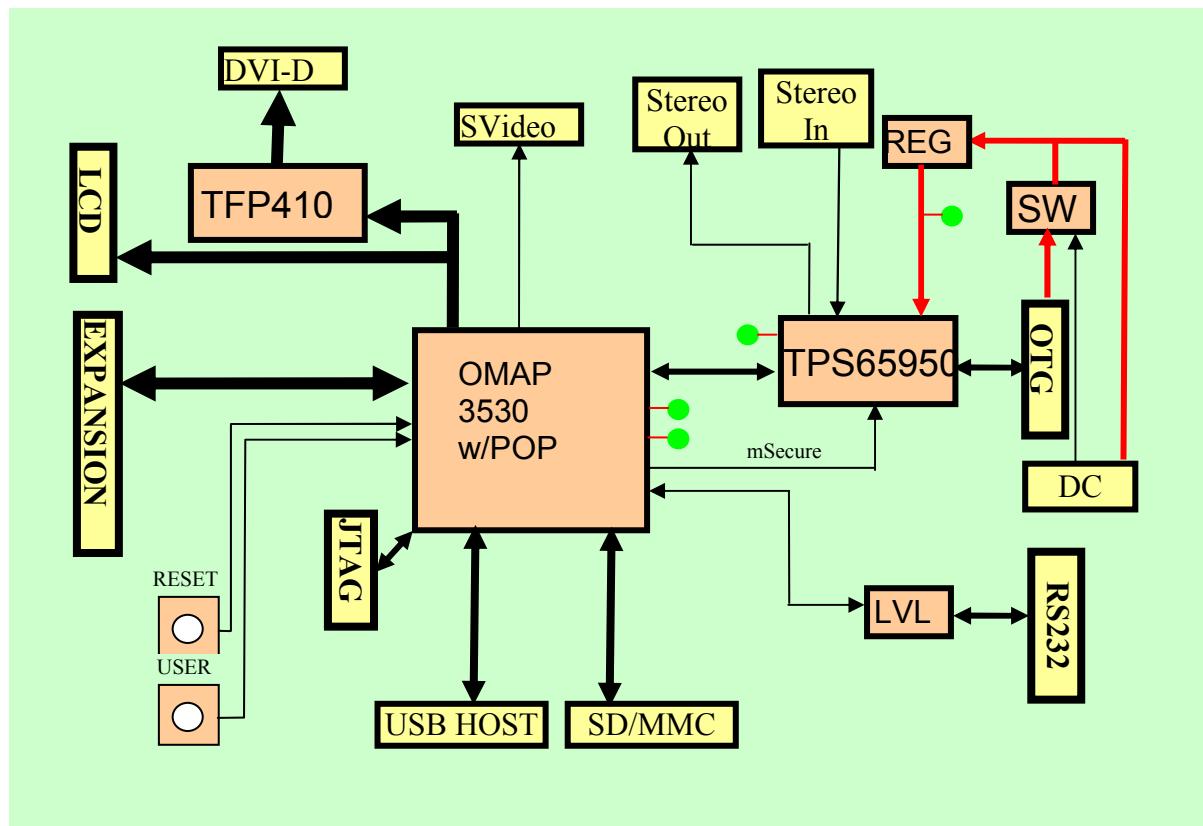


Figure 18. BeagleBoard High Level Block Diagram

Figure 19 shows the location of the components as shown in the block diagram.

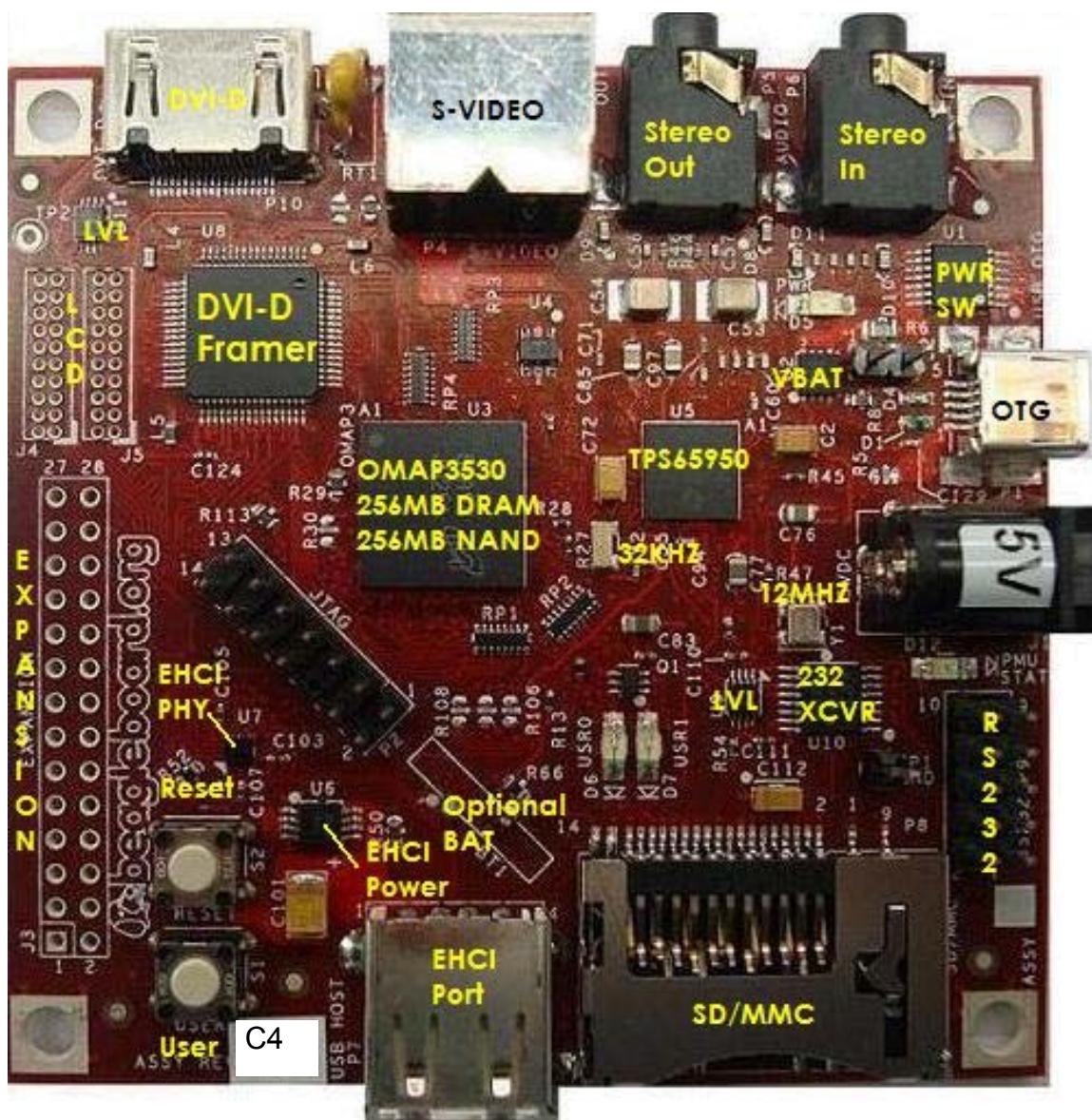


Figure 19. BeagleBoard Top Side Components

There are no key components on the back of the BeagleBoard, but **Figure 20** has been provided for completeness.

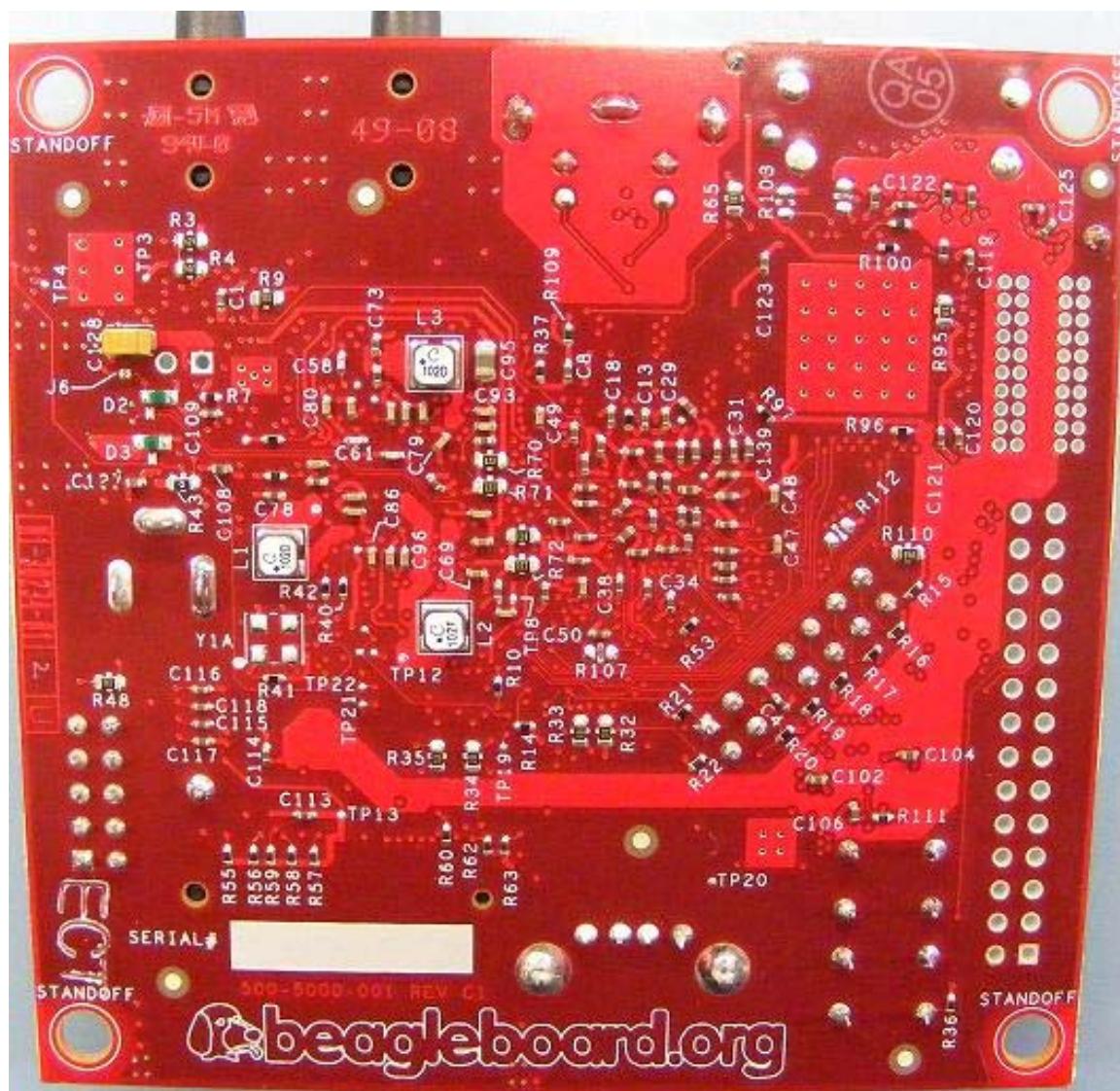


Figure 20. BeagleBoard Backside Components

This remainder of this section describes in detail the architecture and design of the BeagleBoard.

You will notice certain things in this section.

- The schematic has been created for each section showing only the pertinent components and their connections.
 - The pin names differ from the actual schematic. For ease of reading, the names have been truncated to only show the specific functions of that pin as used in the design.

8.2 Input Power

There are two possible sources of the 5V required by the BeagleBoard. It can come from the USB OTG port connected to a PC, powered USB HUB, or a 5V DC supply. The USB supply is sufficient to power the BeagleBoard. However, depending on the load needed by the expansion port on BeagleBoard, additional power may be required. This is where the DC supply comes in to play.

WARNING: DO NOT PLUG IN ANYTHING BUT 5V TO THE DC CONNECTOR OR THE BOARD WILL BE DAMAGED!

It should also be noted that if an OTG configuration is used, for example tying two BeagleBoards together via a UBS OTG cable, both of the BeagleBoards must be powered by the DC supply. If the OTG port is used as a Host port, then the DC supply must also be used.

Figure 21 is the design of the power input section.

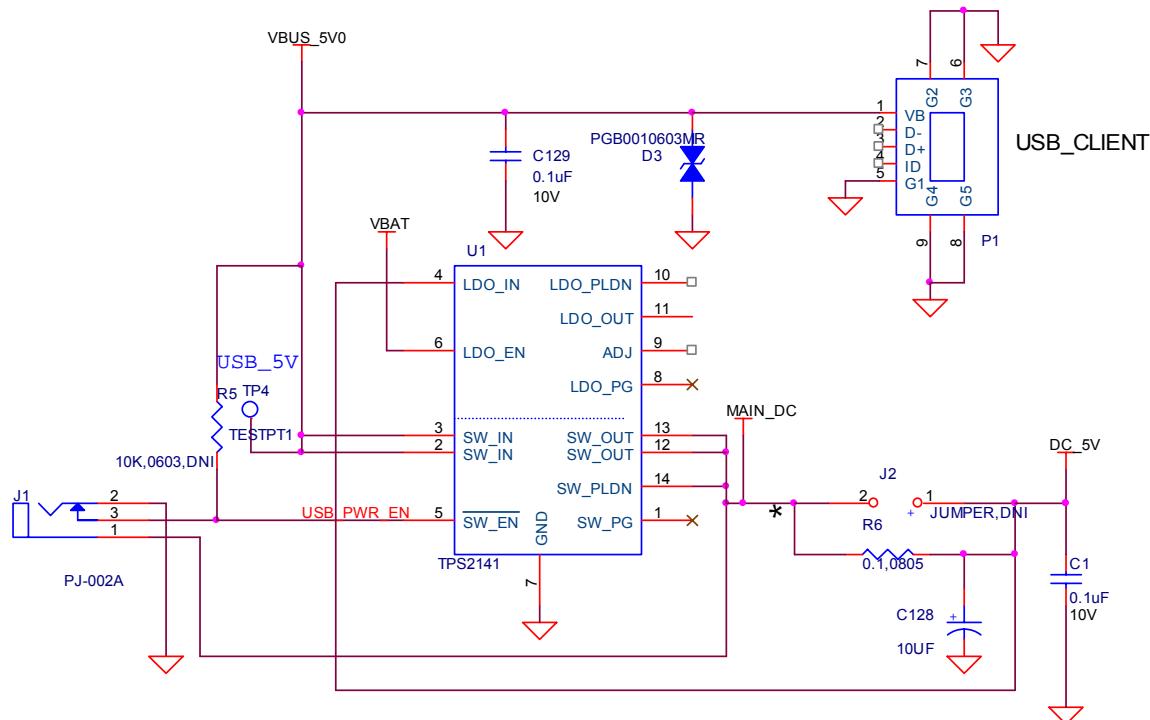


Figure 21. Input Power Section

8.2.1 USB DC Source

The USB specification requires that the current consumed prior to enumeration be limited to 100mA @ 5V (500mW). The 5V DC from the USB is routed through the **TPS2141** switch to insure that this requirement is met as uncharged capacitors on the BeagleBoard can exhibit a large current drain during start up that could exceed this requirement. The **TPS2141** is a USB 2.0 Specification-compatible IC containing a dual-current limiting power switch and an adjustable low dropout regulator (LDO). Both the switch and LDO limit inrush current by controlling the turn on slew rate. The dual-current-limiting feature of the switch allows USB peripherals to utilize high-value capacitance at the output of the switch, while keeping the inrush current low.

During turn on, the switch limits the current delivered to the capacitive load to less than 100 mA. When the output voltage from the switch reaches about 93% of the input voltage, the switch current limit increases to 800mA (minimum), at which point higher current loads can be turned on. The higher current limit provides short circuit protection while allowing the peripheral to draw maximum current from the USB bus.

When in the USB powered mode and no DC supply is connected, the **TPS2141** is enabled, allowing the power to be supplied to the board from the OTG port through the integrated switch inside the **TPS2141**.

8.2.2 Wall Supply Source

A wall supply can be used to provide power to the board. A regulated 5V DC supply of at least 500mA is required. It needs to have a 2.1mm plug with a center hot configuration.

WARNING: DO NOT PLUG IN ANYTHING BUT 5V TO THE DC CONNECTOR OR THE BOARD WILL BE DAMAGED!

In the event that a higher DC load is required due to the addition of a Daughtercard a higher current supply can be used. The maximum current should not exceed 2A.

8.2.3 DC Source Control

Unlike when powering from the USB OTG port, in the case of the DC voltage, the current limiting is not required. As long as the DC supply is not connected, the switch for the USB is enabled. When the DC supply is plugged in, the switch is disabled because the ground is removed from pin 5 of the **TPS2141**. This insures that the 5V from the USB is not connected by disabling the internal FET. In the case where there is no USB plugged in, there is no 5V available to be routed so the removal of the pullup in pin 5 has no affect.

When in the DC mode of operation, the USB OTG can be used in the Host or Client modes. The **TPS65950** will be responsible for handling the supply of the **VBUS_5V0** rail in the OTG or Host modes. As this is limited to 100mA, a powered hub must be used to support peripherals on the OTG port.

8.2.4 3.3V Supply

The **TPS2141** has an integrated 3.3V LDO which is being used to supply the **3.3V** as required on the BeagleBoard for the **DVI-D** interface and the **UART**. The input to the LDO is supplied by the main **DC_5V**. This insures that the power to the LDO can be supplied by either the USB or the DC wall supply and that the current measurement includes the 3.3V supply.

8.2.5 Meter Current Measurement

Jumper **J2** is a header that allows for the voltage drop across the resistor to be measured using a meter, providing a way to measure the current consumption of the BeagleBoard from the main voltage rails, either USB or DC. The resistor, **R6**, is a .1 ohm resistor across which the voltage is measured. The reading you get is .1mV per mA of current.

8.2.6 Processor Current Measurement

The resistor across **J2** can also be used to measure the current of the board by reading the voltage drop across **R6** from software. There are two pairs of resistors provided on the TPS65950 that measure the voltage on either side of R6. This is done via the I2C control bus to the TPS65950 from the OMAP3530 processor. These values along with resistance of R6, are used to calculate the current consumption of the board. **Figure 22** is the schematic of the measurement circuitry.

The maximum value that can be input to the ADC inputs is based on the setting of the **VINTANA2.OUT** voltage rail which defaults to 2.5V. In order to prevent the voltage levels from exceeding this value a pair of resistors of 12K and 10K is used to scale the voltage down.

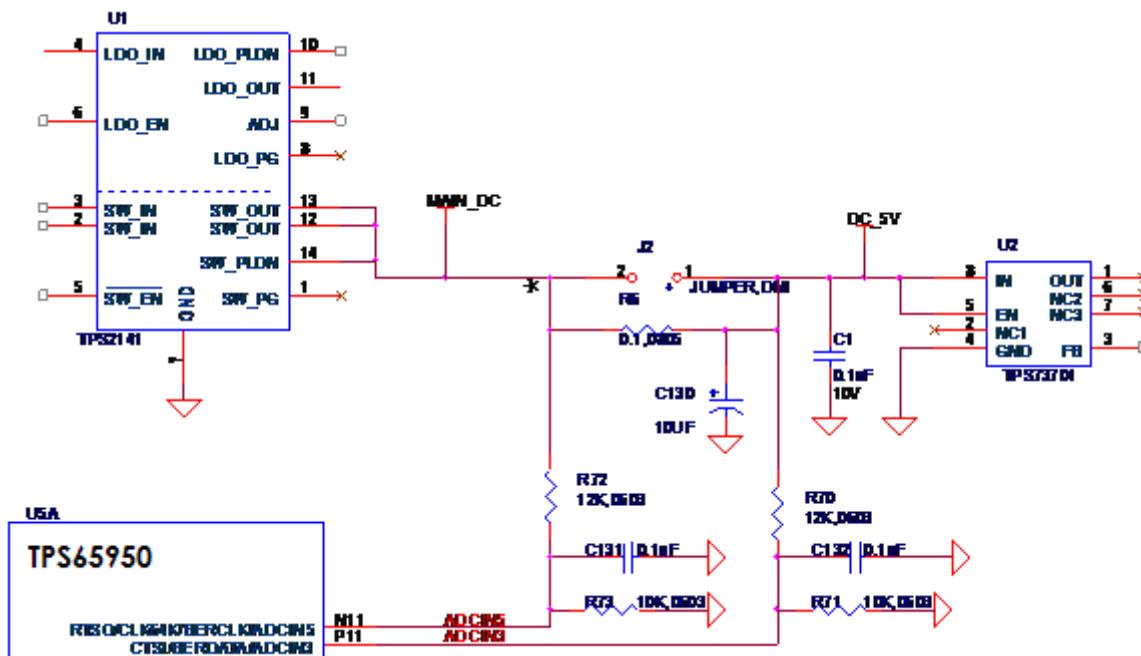


Figure 22. Processor Current Measurement

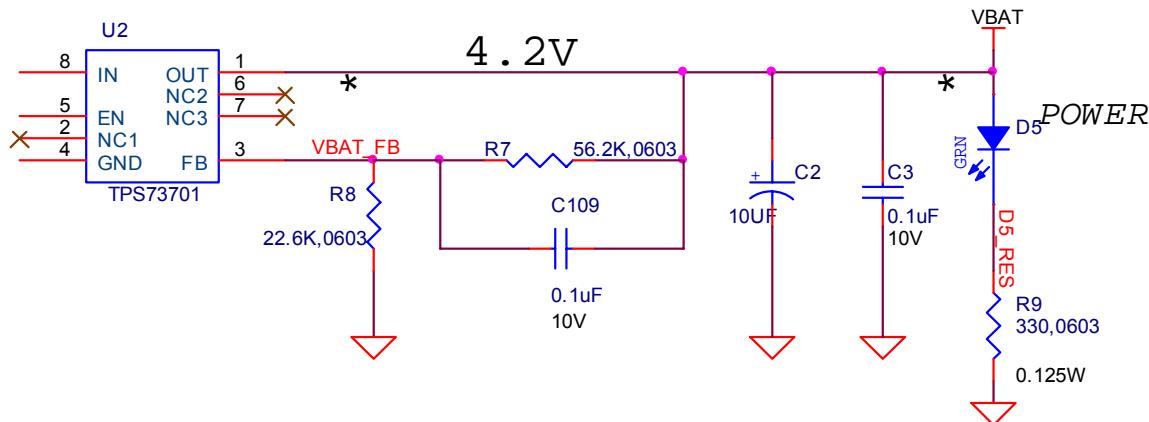
This results in a value that is 46% of the actual value. So, for a maximum value of 5.25V, the voltage read would be 2.415V which keeps it below the 2.5V limit.

The voltage drop across **R6** will be small as the value of the resistor is 0.1 ohms. For every 100 mA of current a voltage of .01V will be detected. In order to determine the actual power, the input voltage and the voltage drop must be measured.

8.3 Power Conditioning

This circuitry regulates the DC input to a nominal 4.2VDC level. This is required in order to meet the maximum DC voltage level as specified by the **TPS65950** Power Management device which is 4.7V. Using 4.2V gives us some margin and meets the nominal 4.2V rating of the **TPS65950**.

Figure 23 is the power conditioning section of the BeagleBoard.

**Figure 23. Power Conditioning**

The **TPS65950** provides the main power rails to the board and has a maximum limit of 4.7V on its VBAT input and a nominal of 4.2V. **U2**, the **TPS73701**, is used to convert the DC_5V, which can come from a DC wall supply or the USB, to 4.2V to meet this requirement. The **TPS73701** is a linear low-dropout (LDO) voltage regulator and is thermal shutdown and current limit protected. It has the ability to deliver 1A of current, although this is far and above the requirements of the board. By adjusting the values of **R7** and **R8**, the actual voltage can be adjusted if needed. The LED **D5** is an indication that the 4.2V is present.

8.4 TPS65950 Reset and Power Management

The **TPS65950** supplies several key functions on the BeagleBoard. This section covers a portion of those functions centered on the power and reset functions. Included in this section is:

- Main Core Voltages
- Peripheral Voltages
- Power Sequencing
- Reset
- Current measurement via SW

The other functions are covered in other sections in this document and are grouped by their overall board functions. The explanation of the various regulators found on the **TPS65950** is based upon how they are used in the board design and are not intended to reflect the overall capability of the **TPS65950** device. Please refer to the **TPS65950** documents for a full explanation of the device operation.

8.4.1 Main Core Voltages

The **TPS65950** supplies the three main voltage rails for the **OMAP3530** processor and the board:

- VOCORE_1V3 (1.2V, adjustable)
- VDD2 (1.3V)
- VIO_1V8 (1.8V)

The **VOCORE_1V3** defaults to **1.2V** at power up, but can be adjusted by software to the **1.3V** level. **Figure 24** is the interfacing of the **TPS65950** to the system as it provides the three main rails.

8.4.2 Main DC Input

The main supply to the **TPS65950** for the main rails is the **VBAT** rail which is a nominal 4.2V. Each rail has a filter cap of **10uF** connected to each of the three inputs. A **.1uF** cap is also provided for high frequency noise filtering.

8.4.3 OMAP3530 I2C Control

The various components in the **TPS65950** are controlled from the OMAP3530 via the I2C interface. I2C_0 is used to control the **TPS65950** device.

8.4.4 VIO_1V8

The **VIO_1V8** rail is generated by the **TPS65950 VIO** regulator. The **VIO** output is a stepdown converter with a choice of two output voltage settings: 1.8 V or 1.85 V. The voltage is set by configuring the VSEL bit (VIO_VSEL[0]). When the VSEL bit is set to 0, the output voltage is 1.8 V, and when it is set to 1, the output voltage is 1.85 V.

When the **TPS65950** resets, the default value of this LDO is 1.80 V; the OMAP3530 must write 1 to the VSEL field to change the output to 1.85 V. The default for the BeagleBoard is 1.8V. This regulator output is used to supply power to the system memories and I/O ports. It is one of the first power supplies to be switched on in the power-up sequence. VIO does not support the SmartReflex voltage control schemes. VIO can be put into sleep or off mode by configuring the SLEEP_STATE and OFF_STATE fields of the VIO_REMAP register.

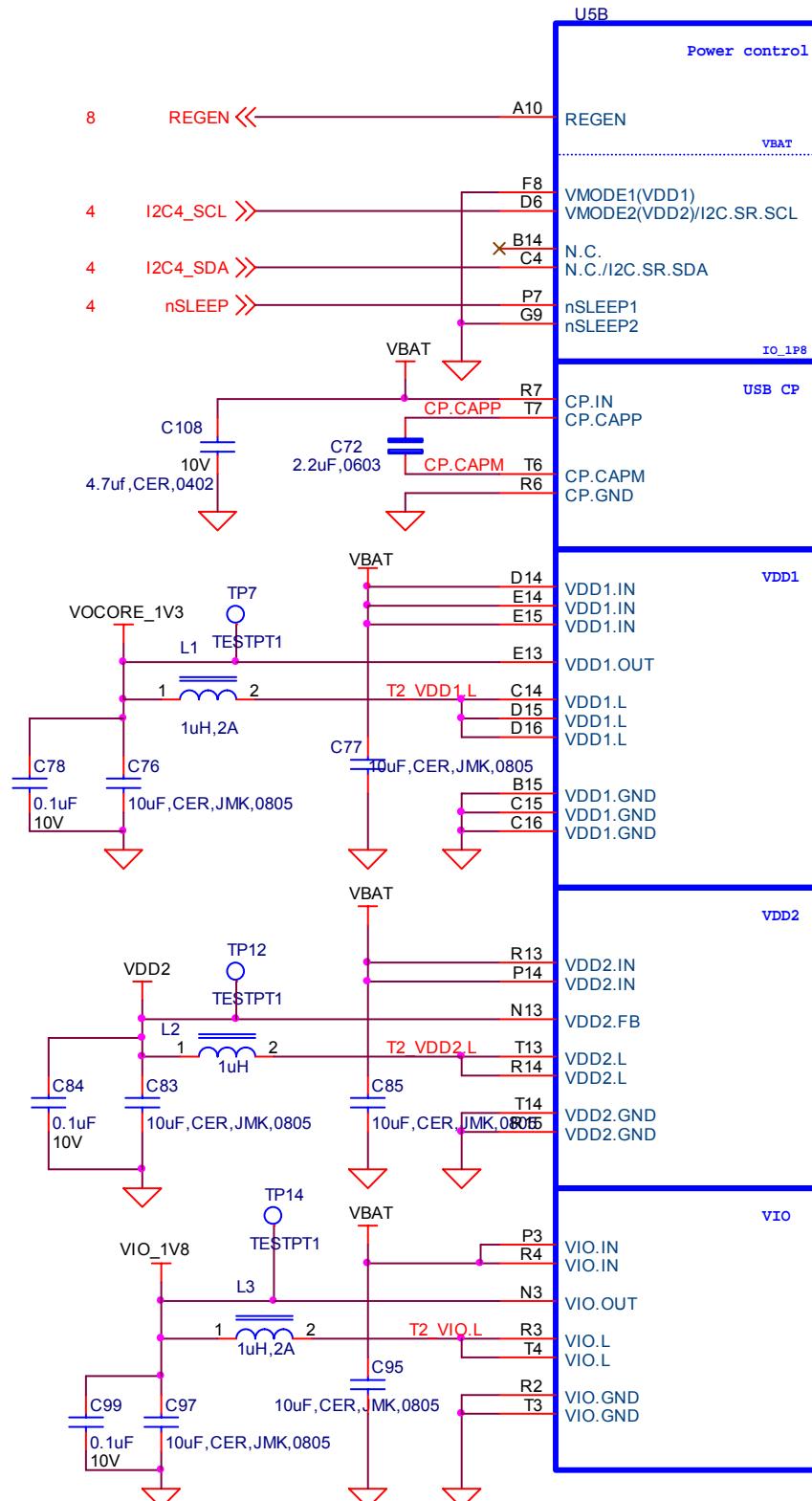


Figure 24. Main Power Rails

8.4.5 Main Core Voltages Smart Reflex

VDD1 and **VDD2** regulators on the **TPS65950** provide SmartReflex-compliant voltage management. The SmartReflex controller in the **OMAP3530** interfaces with the **TPS65950** counterpart through the use of a dedicated I²C bus. The **OMAP3530** computes the required voltage and informs the **TPS65950** using the SmartReflex I²C interface.

SmartReflex control of the **VDD1** and **VDD2** regulators can be enabled by setting the SMARTREFLEX_ENABLE bit (DCDC_GLOBAL_CFG[3]) to 1. To perform **VDD1** voltage control through the SmartReflex interface, the TPS65950 provides the VDD1_SR_CONTROL register. The MODE field of the VDD1_SR_CONTROL register can be set to 0 to put VDD1 in an ACTIVE state; setting the field to 1 moves **VDD1** to a SLEEP state. **VDD1** output voltage can be programmed by setting the VSEL field of the VDD1_SR_CONTROL register. The **VDD1** output voltage is given by $VSEL \times 12.5 \text{ mV} + 600 \text{ mV}$.

8.4.6 VOCORE_1V3

The **VOCORE_1V3** rail is supplied by the **VDD1** regulator of the **TPS65950**. The **VDD1** regulator is a 1.1A stepdown power converter with configurable output voltage between 0.6 V and 1.45 V in steps of 12.5 mV. This regulator is used to power the **OMAP3530** core.

The **OMAP3530** can request the **TPS65950** to scale the **VDD1** output voltage to reduce power consumption. The default output voltage at power-up depends on the boot mode settings, which in the case of the BeagleBoard is 1.2V. The output voltage of the **VDD1** regulator can be scaled by software or hardware by setting the ENABLE_VMODE bit (VDD1_VMODE_CFG[0]). In each of these modes, the output voltage ramp can be single-step or multiple-step, depending on the value of the STEP_REG field of the VDD1_STEP[4:0] register. The **VOCORE_1V3** rail should be set to 1.3V after boot up.

Apart from these modes, the **VDD1** output voltage can also be controlled by the **OMAP3530** through the SmartReflex I²C interface between the OMAP3530 and the **TPS65950**. The default voltage scaling method selected at reset is a software-controlled mode. Regardless of the mode used, **VDD1** can be configured to the same output voltage in sleep mode as in active mode by programming the DCDC_SLP bit of the VDD1_VMODE_CFG[2] register to 0. When the DCDC_SLP bit is 1, the sleep mode output voltage of **VDD1** equals the floor voltage that corresponds to the VFLOOR field (VDD1_VFLOOR[6:0]).

8.4.7 VDD2

The **VDD2** voltage rail is generated by the **TPS65950** using the **VDD2** regulator. The **VDD2** regulator is a stepdown converter with a configurable output voltage of between

0.6 V and 1.45 V and is used to power the OMAP3530 core. **VDD2** differs from **VDD1** in its current load capabilities with an output current rating of 600 mA in active mode.

The **VDD2** provides different voltage regulation schemes. When **VDD2** is controlled by the VMODE2 signal or with the SmartReflex interface, the range of output voltage is 0.6 V to 1.45 V. The use of the VMODE2 signal and the VDD2_VMODE_CFG, VDD2_STEP, VDD2_FLOOR, and VDD2_ROOF registers is similar to the use of the corresponding signals and registers for **VDD1**. **VDD2** shares the same SmartReflex I2C bus to provide voltage regulation. The VDD2_SR_CONTROL register is provided for controlling the **VDD2** output voltage in SmartReflex mode.

When the **VDD2** is used in software-control mode, the VSEL (VDD2_DEDICATED[4:0]) field can be programmed to provide output voltages of between 0.6 V and 1.45 V. The output voltage for a given value of the VSEL field is given by $VSEL \cdot 12.5 \text{ mV} + 600 \text{ mV}$. If the VSEL field is programmed so that the output voltage computes to more than 1.45 V, the TPS65950 sets the **VDD2** output voltage to 1.5 V.

8.5 Peripheral Voltages

There are six additional voltages used by the system that are generated by the **TPS65950**. These are:

- o VDD_PLL2
- o VDD_PLL1
- o VDAC_1V8
- o VDD_SIM
- o VMMC1
- o VAUX2

Figure 25 shows the peripheral voltages supplied by the **TPS65950**.

8.5.1 VDD_PLL2

This programmable LDO is used to power the OMAP3530 PLL circuitry. The **VPLL2** LDO can be configured through the I2C interface to provide output voltage levels of 1.0 V, 1.2 V, 1.3 V, or 1.8 V, based on the value of the VSEL field (VPLL2_DEDICATED[3:0]). On the board this rail is used to power DVI output for pins DSS_DATA(0:5), DSS_DATA(10:15) and DSS_DATA(22:23). The VPLL2 must be set to 1.8V for proper operation of the **DVI-D** interface.

8.5.2 VDD_PLL1

The VPLL1 programmable LDO regulator is low-noise, linear regulator used for the OMAP3530 PLL supply. The VDD_PLL1 rail is initialized to 1.8V.

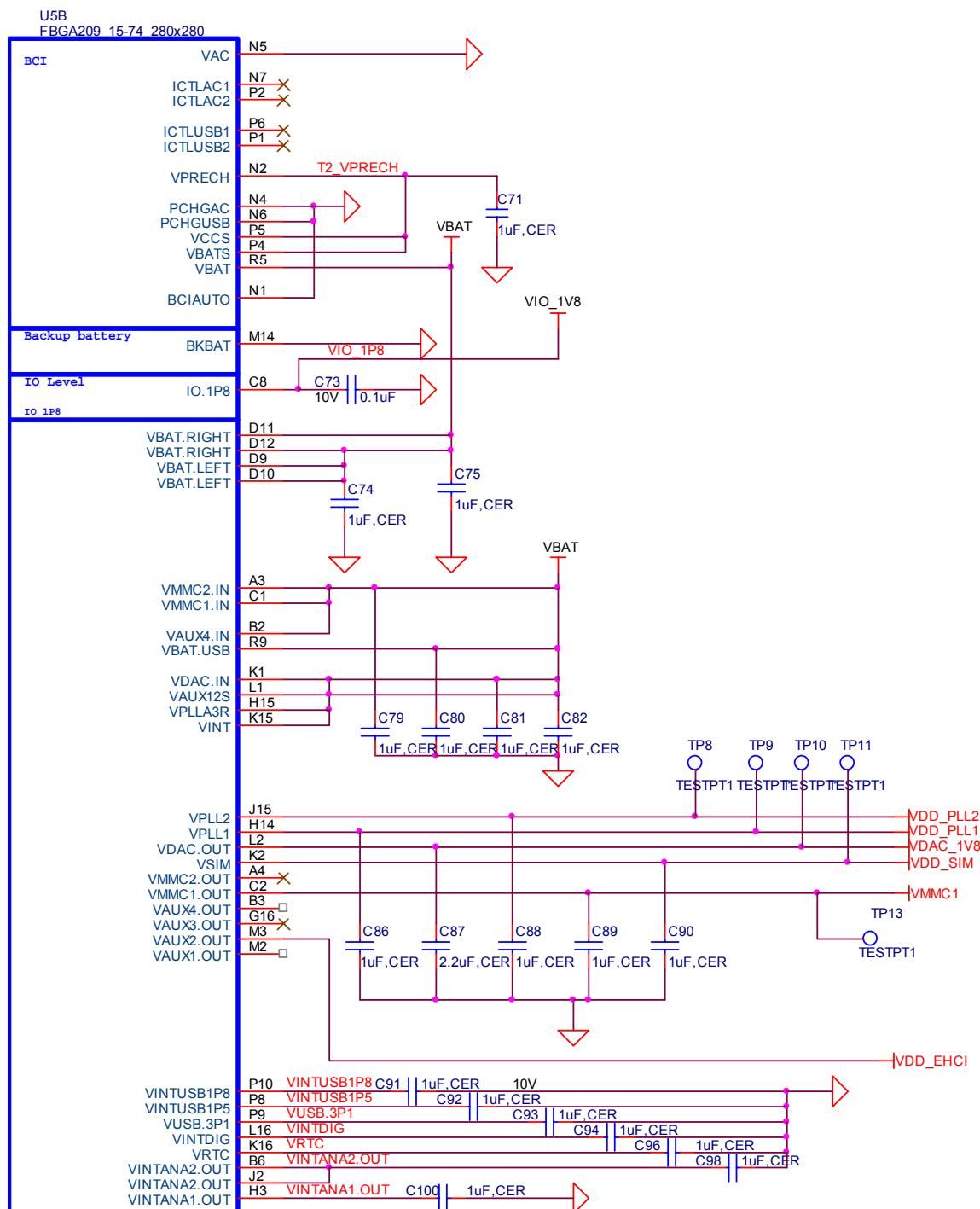


Figure 25. Peripheral Voltages

8.5.3 VDAC_1V8

The **VDAC** programmable LDO regulator is a high-PSRR, low-noise, linear regulator that powers the OMAP3530 dual-video DAC. It is controllable with registers via I2C and can be powered down if needed. The **VDAC** LDO can be configured to provide 1.2V, 1.3 V, or 1.8 V in on power mode, based on the value of the VSEL field (VDAC_DEDICATED[3:0]). The **VDAC_1V8** rail should be set to 1.8V for the BeagleBoard.

8.5.4 VDD_SIM

This voltage regulator is a programmable, low dropout, linear voltage regulator supplying the bottom 4 bits of the 8 bit **SD/MMC** card slot. The VSEL field (VSIM_DEDICATED[3:0]) can be programmed to provide output voltage of 1.0 V, 1.2 V, 1.3 V, 1.8 V, 2.8 V, or 3.0 V and can deliver up to 50mA. The default output voltage of this LDO as directed by the **TPS65950** boot pins is 1.8V.

8.5.5 VMMC1

The **VMMC1** LDO regulator is a programmable linear voltage converter that powers the MMC1 slot and includes a discharge resistor and overcurrent protection (short-circuit). This LDO regulator can also be turned off automatically when the MMC card extraction is detected. The VMMC1 LDO is powered from the main **VBAT** rail. The **VMMC1** rail defaults to 3.0V as directed by the **TPS65950** boot pins and will deliver up to 220mA. It can be set to 3.0V in the event 3V cards are being used.

8.5.6 VAUX2

The **VAUX2** LDO regulator is a programmable linear voltage converter that powers the 1.8V I/O rail of the USB PHY and includes a discharge resistor and overcurrent protection (short-circuit). The VAUX2 LDO is powered from the main **VBAT** rail. The **VMMC1** rail defaults to 3.0V as directed by the **TPS65950** boot pins and will deliver up to 220mA. The voltage rail is labeled **VDD_EHCI** on the schematic.

8.5.7 Boot Configuration

The boot configuration pins on the **TPS65950** determine the power sequence of the device. For the OMAP3530 support, the boot pin configuration is fixed at:

- **BOOT0** tied to **VBAT**
- **BOOT1** tied to Ground.

8.5.8 RTC Backup Battery

An optional battery to backup the Real Time Clock that is in the **TPS65950**. The board does not come equipped with the battery. The battery can be purchased from DigiKey or other component suppliers. When the battery is not installed, **R66** must be installed. You must make sure that prior to installing the battery that R66 is removed.

Refer to section **9.11** for information on the battery selection and installation.

8.5.9 Power Sequencing

Based on the boot configuration pins, the **TPS65950** knows the type of OMAP processor that it needs to support, in this case the OMAP3530. The voltages are ramped in a sequence that is compatible with the OMAP3530 processor. **Figure 26** is the sequence in which the power rails, clocks, and reset signal come up.

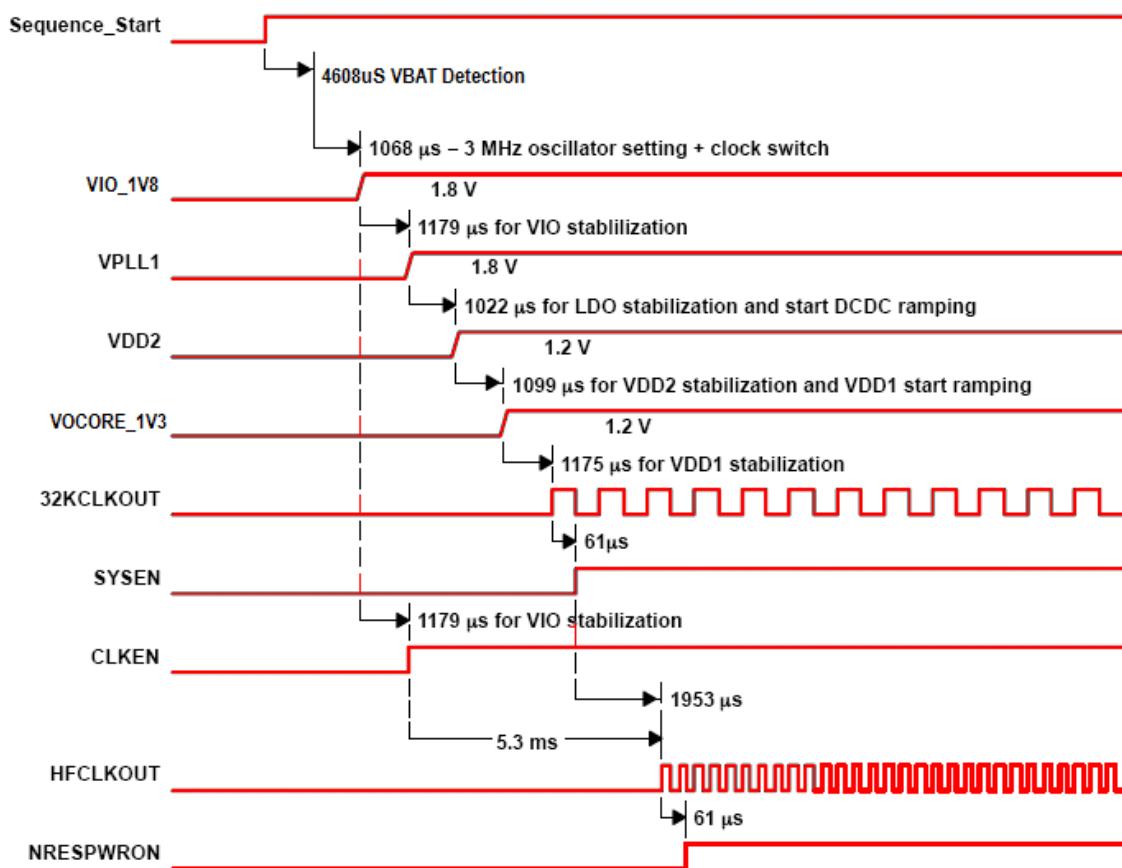


Figure 26. Power Sequencing

8.5.10 Reset Signals

The BeagleBoard uses three distinct reset circuits:

- Warm Reset
- Cold Reset
- User Reset

Figure 27 shows the connections for the Reset interfaces.

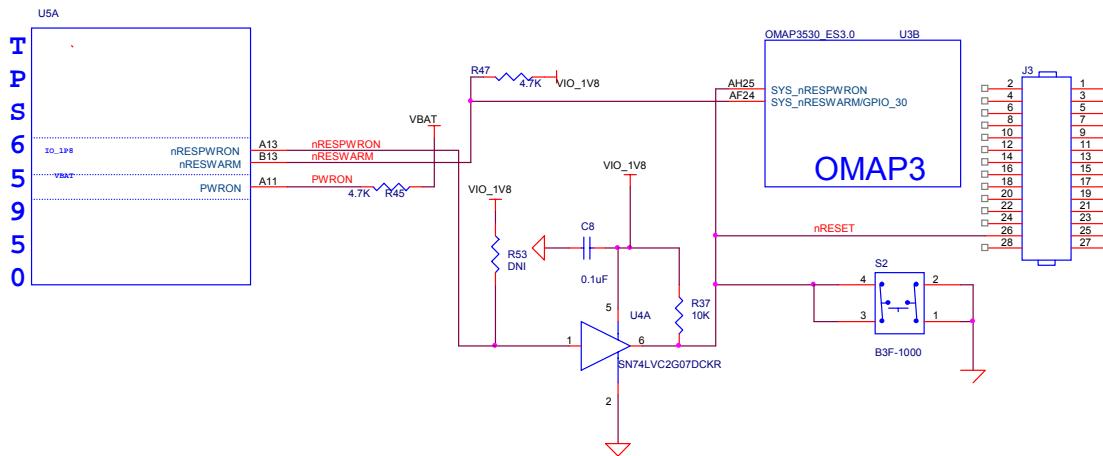


Figure 27. Reset Circuitry

8.5.10.1 Warm Reset

The warm reset is generated by the **OMAP3530** processor on power up. The **nRESWARM** signal is a bidirectional reset. When an internal reset occurs, **nRESWARM** goes low and resets all the peripherals and the **TPS65950**. The **TPS65950** can be configured to perform a warm reset of the device to bring it into a known defined state by detecting a request for a warm reset on the **NRESWARM** pin. The minimum duration of the pulse on the **nRESWARM** pin should be two 32-kHz clock cycles. The **nRESWARM** output is open-drain; consequently, an external pullup resistor is required. There is no way for the user to generate a warm reset on the BeagleBoard.

8.5.10.2 Cold Reset

On power up as shown in **Figure 27**, the **TPS65950** generates **nRESPWRON**, power on reset. The signal from the **TPS65950** is an output only and is not an open drain signal. By running the signal through a buffer, **SN74LVC2G07**, the signal becomes open drain, which requires a pullup on the signal. This will allow the **nRESPWRON** signal to be

pulled low, by pressing the reset switch S2, to force a reset to the **OMAP3530** processor and to any device on the expansion card that require a reset.

It also allows for the reset signal to be pulled low or held low for an extended time by circuitry on the expansion card if needed.

8.5.10.3 User Reset

The USER RESET button can be used to request a Warm Reset from the processor. After initialization, this pin becomes an input to the processor. By pushing the Reset button, an interrupt is generated into the OMAP3530 processor. The software that is run as a result of this can then do whatever housekeeping is required and then send the processor into a reset mode.

8.5.10.4 PWRON

You will notice another signal on the **TPS65950** called **PWRON**. This signal is referenced in the **TPS65950** documentation. In the BeagleBoard design it is not used but it is pulled high to insure the desired operation is maintained.

8.5.11 mSecure Signal

This signal provides for protection of the **RTC registers** in the **TPS65950** by disabling that function via a control signal from the **OMAP3530** processor.

For more information on the operation on the signal, please refer to the **OMAP3530** Technical Reference Manual.

8.6 OMAP3530 Processor

The heart of BeagleBoard is the OMAP3530 processor. **Figure 28** is a high level block diagram of the OMAP3530.

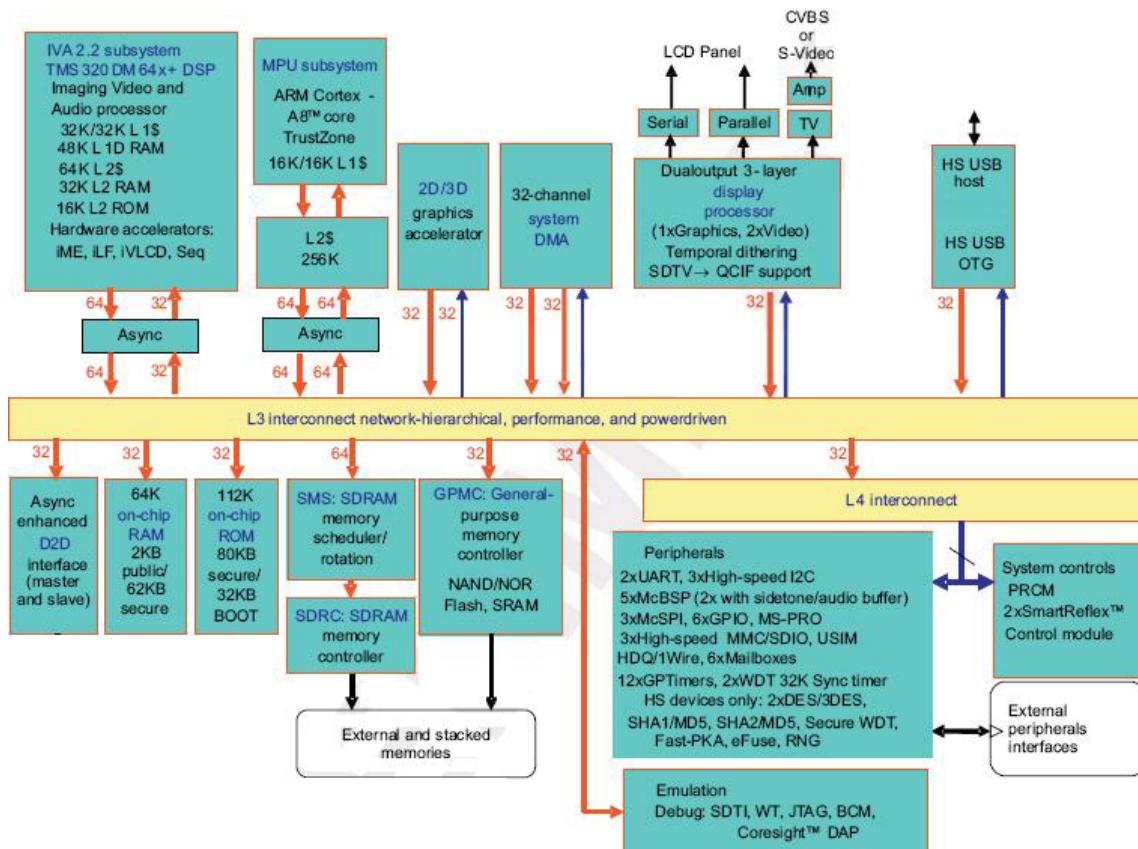


Figure 28. OMAP3530 Block Diagram

8.6.1 Overview

The OMAP3530 high-performance, multimedia application device is based on the enhanced OMAP™ 3 architecture and is integrated on TI's advanced 65-nm process technology. The OMAP3530 architecture is configured with different sets of features in different tier devices. Some features are not available in the lower-tier devices. For more information, refer to the OMAP3530 Technical Reference Manual (TRM). The architecture is designed to provide best-in-class video, image, and graphics processing sufficient to various applications.

The OMAP3530 supports high-level operating systems (OSs), such as:

- o Windows CE

- Linux
- QNX
- Symbian
- Others

This OMAP3530 device includes state-of-the-art power-management techniques required for high-performance low power products. The OMAP3530 supports the following functions and interfaces on the BeagleBoard:

- Microprocessor unit (MPU) subsystem based on the ARM Cortex-A8™ microprocessor
- POP Memory interface
 - 2Gb MDDR (256Mbytes)
 - 2Gb NAND Flash (256Mbytes)
- 24 Bit RGB Display interface (DSS)
- SD/MMC interface
- USB OTG interface
- NTSC/PAL/S-Video output
- Power management
- Serial interface
- I²C interface
- I²S Audio interface (McBSP2)
- Expansion McBSP1
- JTAG debugging interface

8.6.2 SDRAM Bus

The SDRAM bus is not accessible on the BeagleBoard. Its connectivity is limited to the POP memory access on the top of the OMAP3530 and therefore is only accessible by the SDRAM memory.

The base address for the DDR SDRAM in the POP device is **0x8000 0000**.

8.6.3 GPMC Bus

The GPMC bus is not accessible on the BeagleBoard. Its connectivity is limited to the POP memory access on the top of the OMAP3530 and therefore is only accessible by the NAND memory.

The memory on the GPMC bus is NAND and therefore will support the classical NAND interface. The address of the memory space is programmable.

8.6.4 DSS Bus

The display subsystem provides the logic to display a video frame from the memory frame buffer in SDRAM onto a liquid-crystal display (LCD) display via the DVI-D interface or to a standalone LCD panel via the LCD interface connectors. The logic levels of the LCD expansion connectors are 1.8V so it will require buffering of the signals to drive most LCD panels. The DSS is configured to a maximum of 24 bits, but can be used in lower bit modes if needed.

8.6.5 McBSP2

The multi-channel buffered serial port (McBSP) McBSP2 provides a full-duplex direct serial interface between the OMAP3530 and the audio CODEC in the **TPS65950** using the I2S format. Only four signals are supported on the McBSP2 port. **Figure 29** is a depiction of McBSP2.

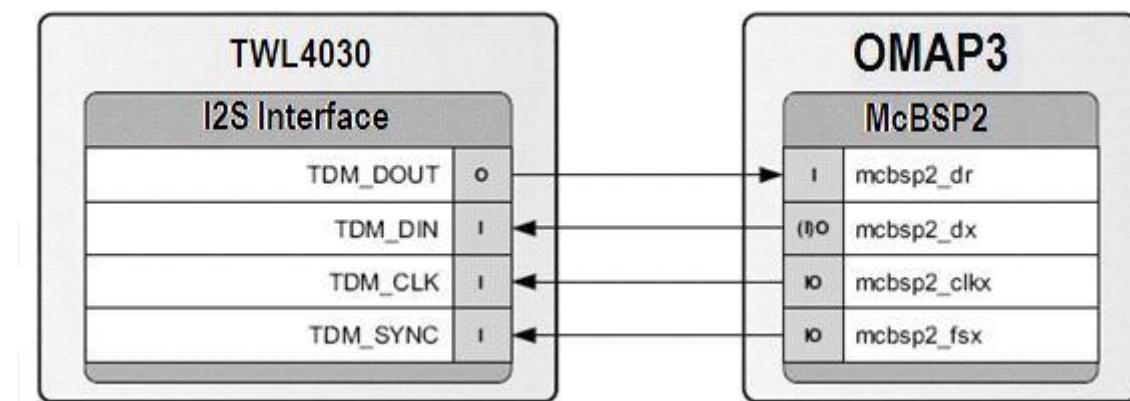


Figure 29. McBSP2 Interface

8.6.6 McBSP1

McBSP1 provides a full-duplex direct serial interface between the OMAP3530 and the expansion interface. There are 6 signals supported on McBSP1, unlike the 4 signals on the other ports. **Figure 30** is a diagram of McBSP1.

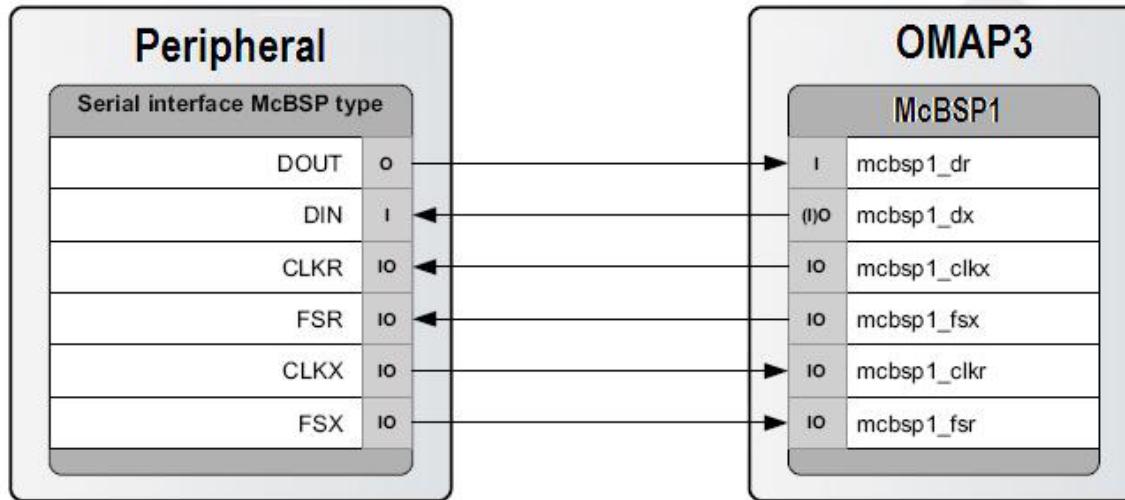


Figure 30. McBSP1 Interface

8.6.7 McBSP3

McBSP3 provides a full-duplex direct serial interface between the OMAP3530 and the expansion interface. **Figure 31** is a diagram of McBSP3.

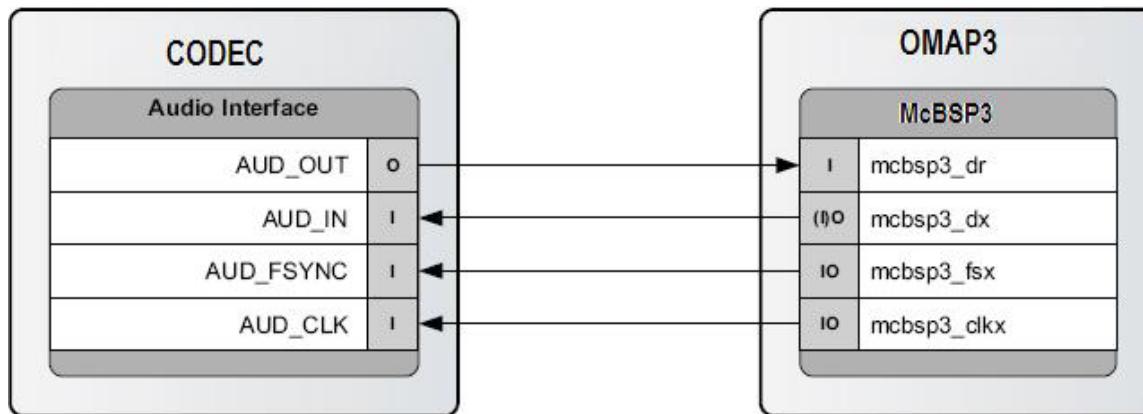


Figure 31. McBSP3 Interface

8.6.8 Pin Muxing

On the OMAP3530, the majority of pins have multiple configurations that the pin can be set to. In essence, the pin can become different signals depending on how they are set in the software. In order for the BeagleBoard to operate, the pins used must be set to the correct signal. In some cases, the default signal is the correct signal. Each pin can have a maximum of 8 options on the pin. This is called the pin mode and is indicated by a three

bit value (0:3). In the case of the signals going to the expansion connector, the settings required for those pins depends on how they are to be used. For an explanation of the options, please refer to the Expansion Header section. Each pin can be set to a different mode independent of the other pins on the connector.

Table 4 is a list of all of the signals used on the OMAP3530 for the BeagleBoard and the required mode setting for each pin. Where the default setting is needed, it will be indicated. The USER notation under mode indicates that this is an expansion signal and can be set at the discretion of the user. A FIXED indicates that there is only one function for that signal and that it cannot be changed,

Table 4. OMAP3530 Pin Muxing Settings

Signal	Mode
DSS	Default
MMC1	Default
MMC2	User
UART3	Default
GPMC	Default
UART1	Default
I2C1	Default
I2C2	Default
I2C3	Default
I2C4	Default
JTAG	FIXED
TV OUT	Default
SYS_nRESPWRON	Default
SYS_nRESWARM	Default
SYS_nIRQ	Default
SYS_OFF	Default
SYS_CLKOUT	Default
SYS_CLKOUT2	Default
SYS_CLKREQ	Default
SYS_XTALIN	FIXED
GPIO_149	4
GPIO_150	4
McBSP1	Default
McBSP2	User
McBSP3	Default
GPIO_171	4
GPIO_172	4

8.6.9 GPIO Mapping

There are a number of GPIO pins from the OMAP3530 that are used on the BeagleBoard design. **Table 5** shows which of these GPIO pins are used in the design and whether they are inputs or outputs. While GPIO pins can be used as interrupts, the table only covers the GPIO pin mode. If it is an interrupt, then it is covered in the interrupt section.

Table 5. OMAP3530 GPIO Pins

OMAP PIN	INT/GPIO	I/O	Signal	USAGE
AA9	GPIO_149	O	LED_GPIO149	Controls User LED0
W8	GPIO_150	O	LED_GPIO149	Controls User LED1
AG9	GPIO_23	I	MMC1_WP	SD/MMC card slot Write protect
J25	GPIO_170	O	DVI_PUP	Controls the DVI-D interface. A Hi = DVI-D enabled.
AE21	GPIO_7	I	SYSBOOT_5	Used to put the device in the boot mode or as a user button input

Other signals, such as those that connect to the expansion connector, may also be set as a GPIO pin. For information on those, refer to the Expansion Connector section.

8.6.10 Interrupt Mapping

There are a small number of pins on the OMAP3530 that act as interrupt. Some of these interrupts are connected to the TPS65950 and their status is reflected through the main TPS65950 interrupt. **Table 6** lists the interrupts.

Table 6. OMAP3530 Interrupt Pins

TPS65950 Pin	OMAP PIN	INT/GPIO	USAGE
	AF26	SYS_nIRQ	Interrupt from the TPS65950
	AH8	GPIO_29	SD Write protect lead. Can be polled or set to an interrupt.
P12		GPIO0	MMC1 card detect input. Goes to the OMAP3530 over the SYS_nIRQ pin.

8.7 POP Memory Device

The OMAP3530 uses what is called POP (Package-on-Package) memory. The memory is a MCP (Multi Chip Package) that contains both the Mobile DDR SDRAM and the NAND Flash. **Figure 32** shows the POP Memory concept.



Figure 32. POP Memory

The Memory device mounts on top of the OMAP3530 device. The configuration used on the board is a 2Gb NAND Flash plus 2Gb MDDR SDRAM device from Micron.

8.8 System Clocks

There are three clocks needed for the operation of the BeagleBoard, 32KHz, 26MHz and McBSP_CLKS. **Figure 33** shows the components that make up the System Clocks.

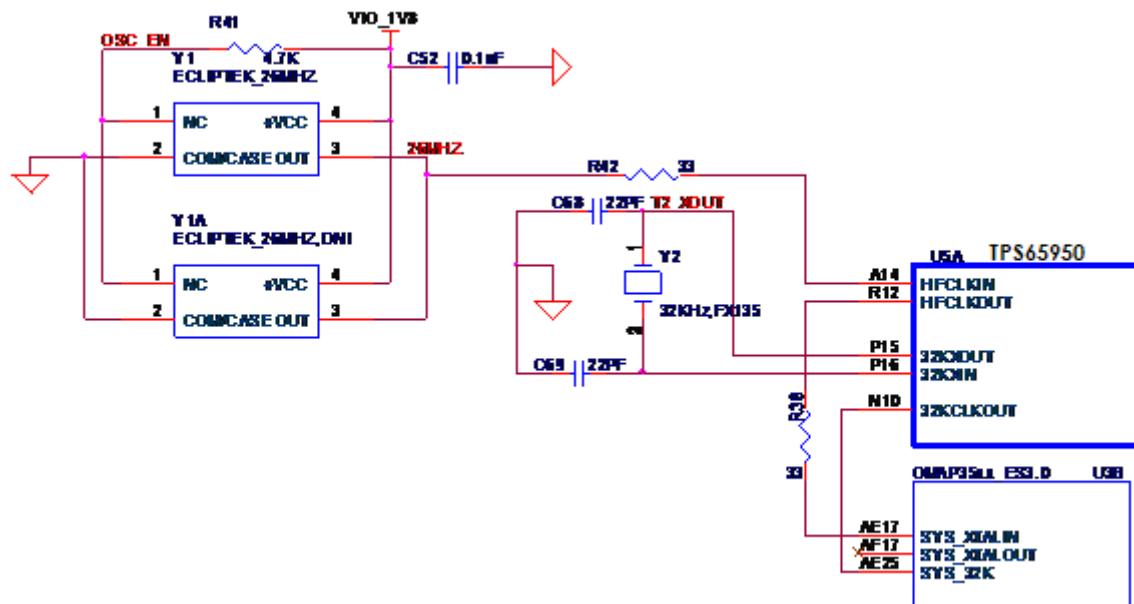


Figure 33. System Clocks

8.8.1 32KHz Clock

The 32KHz clock is needed for the TPS65950 and the **OMAP3530** and is provided by the **TPS65950** via the external 32KHz crystal, Y2. The **TPS65950** has a separate output

from the crystal to drive the OMAP3530 that buffers the resulting 32-kHz signal and provides it as **32KCLKOUT**, which is provided to the OMAP3530 on ball **AE25**. The default mode of the **32KCLKOUT** signal is active, but it can be disabled if desired under SW control.

The 32.768-kHz clock drives the RTC embedded in the **TPS65950**. The RTC is not enabled by default; the host processor must set the correct date and time to enable the RTC.

8.8.2 26MHz Clock

This section describes the 26MHz clock section of the BeagleBoard.

8.8.2.1 26MHz Source

The BeagleBoard is designed to support two suppliers of the 26MHz oscillator. The **26MHz** clock is provided by an onboard oscillator, **Y1**. The **TPS65950** receives the external **HFCLKIN** signal on ball **A14** and uses it to synchronize or generate the clocks required to operate the TPS65950 subsystems. The **TPS65950** must have this clock in order to function to the point where it can power up the BeagleBoard. This is the reason the **26MHz** clock is routed through the TPS65950.

8.8.2.2 TPS65950 Setup

When the TPS65950 enters an active state, the OMAP3530 must immediately indicate the **HFCLKIN** frequency (26 MHz) by setting the **HFCLK_FREQ** bit field (bits [1:0]) in the **CFG_BOOT** register of the TPS65950. **HFCLK_FREQ** has a default of being not programmed, and in that condition, the USB subsection does not work. The three DCDC switching supplies (**VIO**, **VDD1**, and **VDD2**) operate from their free-running 3-MHz (RC) oscillators, and the **PWR** registers are accessed at a default 1.5-M byte. **HFCLK_FREQ** must be set by the OMAP3530 during the initial power-up sequence. On the BeagleBoard, this is done by the internal boot ROM on startup.

8.8.2.3 OMAP3530 26MHz

The 26MHz clock for the **OMAP3530** is provided by the TPS65950 on ball **R12** through **R38**, a 33 ohm resistor is providing to minimize any reflections on the clock line. The clock signal enters via ball **AE17** on the **OMAP3530**.

8.8.3 McBSP_CLKS

An additional clock is also provided by the **TPS65950** called **McBSP_CLKS**. This clock is provided to the OMAP3530 in order to insure synchronization of the I2S interface between the **OMAP3530** and the **TPS65950**.

8.9 USB OTG Port

The BeagleBoard has a USB OTG (On-the-Go) port. It can be used as an OTG port, Client port, or Host port. The main use is as a client port, as that is the mode that will supply the power needed to power the BeagleBoard.

NOTE: In order to use the OTG in the Host mode, the BeagleBoard must be powered from the DC supply.

8.9.1 USB OTG Overview

USB OTG is a supplement to the USB 2.0 specification. The standard USB uses a master/slave architecture, a USB host acting as a master and a USB peripheral acting as a slave. Only the USB host can schedule the configuration and data transfers over the link. The USB peripherals cannot initiate data transfers, they only respond to instructions given by a host.

USB OTG works differently in that gadgets don't need to be pure peripherals because they can sometimes act as hosts. An example might be connecting a USB keyboard or printer to BeagleBoard or a USB printer that knows how to grab documents from certain peripherals and print them. The USB OTG compatible devices are able to initiate the session, control the connection and exchange Host/Peripheral roles between each other.

The USB OTG supplement does not prevent the use of a hub, but it describes role swapping only in the case of a one-to-one connection where two OTG devices are directly connected. If a standard hub is used, the supplement notes that using it will lead to losing USB OTG role-swap capabilities making one device as the Default-Host and the other as the Default-Peripheral until the hub is disconnected.

The combination of the **OMAP3530** and the **TPS65950** allows the BeagleBoard to work as an OTG device if desired. The primary mode of operation however, is intended to be a client mode in order to pull power from the USB host which is typically a PC. As the Rev B does not have a Host USB port, this port will be used as a Host port in many applications.

8.9.2 USB OTG Design

Figure 34 is the design of the USB OTG port on the BeagleBoard.

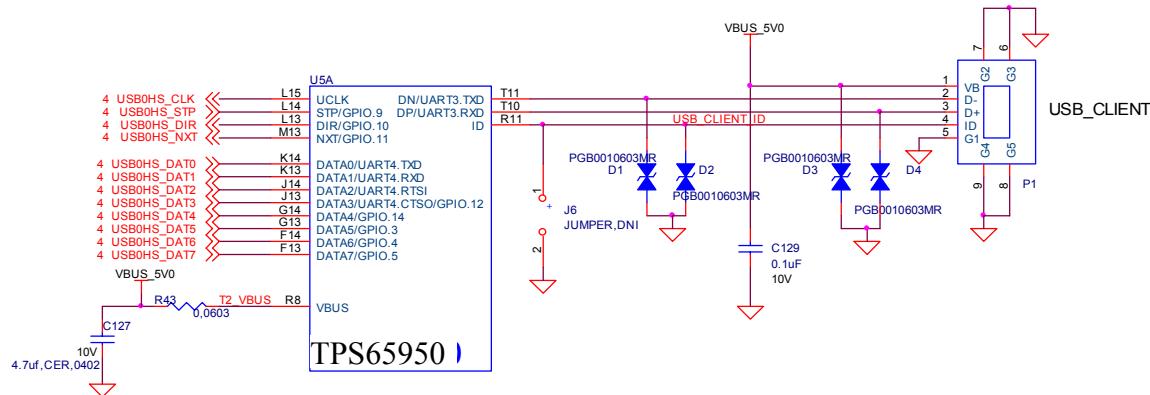


Figure 34. USB OTG Design

8.9.3 OTG ULPI Interface

ULPI is an interface standard for high-speed USB 2.0 systems. It defines an interface between USB link controller (**OMAP3530**) and the **TPS65950** that drives the actual bus. ULPI stands for UTMI+ low pin interface and is designed specifically to reduce the pin count of discrete high-speed USB PHYs. Pin count reductions minimize the cost and footprint of the PHY chip on the PCB and reduce the number of pins dedicated to USB for the link controller.

Unlike full- and low-speed USB systems, which utilize serial interfaces, high-speed requires a parallel interface between the controller and PHY in order to run the bus at 480Mbps. This leads to a corresponding increase in complexity and pin count. The ULPI used on the BeagleBoard keeps this down to only 12 signals because it combines just three control signals, plus clock, with an 8-bit bi-directional data bus. This bus is also used for the USB packet transmission and for accessing register data in the ULPI PHY.

8.9.3.1 OMAP3530 Interface

The controller for the ULPI interface is the OMAP3530. It provides all of the required signals to drive the interface. **Table 7** describes the signals from the OMAP3530 that are used for the USB OTG interface.

Table 7. OMAP3530 ULPI Interface

Signal	Description	Type	Ball
hsusb0_clk	Dedicated for external transceiver 60-MHz clock input from PHY	I	T28

hsusb0_stp	Dedicated for external transceiver Stop signal	O	T25
hsusb0_dir	Dedicated for external transceiver Data direction control from PHY	I	R28
hsusb0_nxt	Dedicated for external transceiver Next signal from PHY	I	T26
hsusb0_data0	Transceiver Bidirectional data bus	I/O	T27
hsusb0_data1	Transceiver Bidirectional data bus	I/O	U28
hsusb0_data2	Transceiver Bidirectional data bus	I/O	U27
hsusb0_data3	Transceiver Bidirectional data bus	I/O	U26
hsusb0_data4	Transceiver Bidirectional data bus	I/O	U25
hsusb0_data5	Transceiver Bidirectional data bus	I/O	V28
hsusb0_data6	Transceiver Bidirectional data bus	I/O	V27
hsusb0_data7	Transceiver Bidirectional data bus	I/O	V26

8.9.3.2 TPS65950 Interface

The TPS65950 USB interfaces to the OMAP3 over the ULPI interface. **Table 8** is a list of the signals used on the TPS65950 for the ULPI interface.

Table 8. OMAP3530 ULPI Interface

Signal	Description	Type	Ball
UCLK	High speed USB clock	I/O	L15
STP	High speed USB stop	I	L14
DIR	High speed USB dir	O	L13
NXT	High speed USB direction	O	M1
DATA0	High speed USB Data bit 0	I/O	K14
DATA1	High speed USB Data bit 0	I/O	K13
DATA2	High speed USB Data bit 0	I/O	J14
DATA3	High speed USB Data bit 0	I/O	J13
DATA4	High speed USB Data bit 0	I/O	G14
DATA5	High speed USB Data bit 0	I/O	G13
DATA6	High speed USB Data bit 0	I/O	F14
DATA7	High speed USB Data bit 0	I/O	F13

8.9.4 OTG Charge Pump

When the **TPS65950** acts as an A-device, the USB charge pump is used to provide 4.8 V/100 mA to the VBUS pin. When the **TPS65950** acts as a B-device, the USB charge pump is in high impedance. If used in the OTG mode as an A-device, the BeagleBoard will need to be powered from the DC supply. If acting as a B-device, there will not be a voltage source on the USB OTG port to drive the BeagleBoard. **Table 9** describes the charge pump pins.

Table 9. USB OTG Charge Pump Pins

Signal	Description	Type	Ball
CP.IN	The charge pump input voltage. Connected to VBAT.	Power	R7
CP.CAPP	The charge pump flying capacitor plus.	O	L14
CP.CAPM	The charge pump flying capacitor minus.	O	T6
CP.GND	The charge pump ground.	GND	R6

The charge pump is powered by the **VBAT** voltage rail. The charge pump generates a 4.8-V (nominal) power supply voltage to the **VBUS** pin. The input voltage range is 2.7 V to 4.5 V so the 4.2V VBAT is within this range. The charge pump operating frequency is 1 MHz. The charge pump integrates a short-circuit current limitation at 450 mA.

8.9.5 OTG USB Connector

The OTG USB interface is accessed through the miniAB USB connector. If you want to use the OTG port as a USB Host, **pin 4** of the connector must be grounded. The Rev C4 version of Beagle provides jumper pad, **J6** that allows for a small piece of solder to be placed on the pads to perform this function. It should be noted that with the USB Host port on the Rev C4 Beagle, the need to convert the OTG port to a host mode is greatly diminished.

8.9.6 OTG USB Protection

Each lead on the USB port has ESD protection. In order for the interface to meet the USB 2.0 Specification Eye Diagram, these protection devices must be low capacitance.

8.10 USB Host Port

The Rev C4 is equipped with a High Speed USB Host interface connected to the ULPI port 2 on the OMAP3530. It uses a SMSC PHY as the physical interface and provides power control to the USB connector. This port is a High Speed only port and will not support low speed or full speed devices plugged directly into the connector. **Figure 35** is the design of the USB Host port.

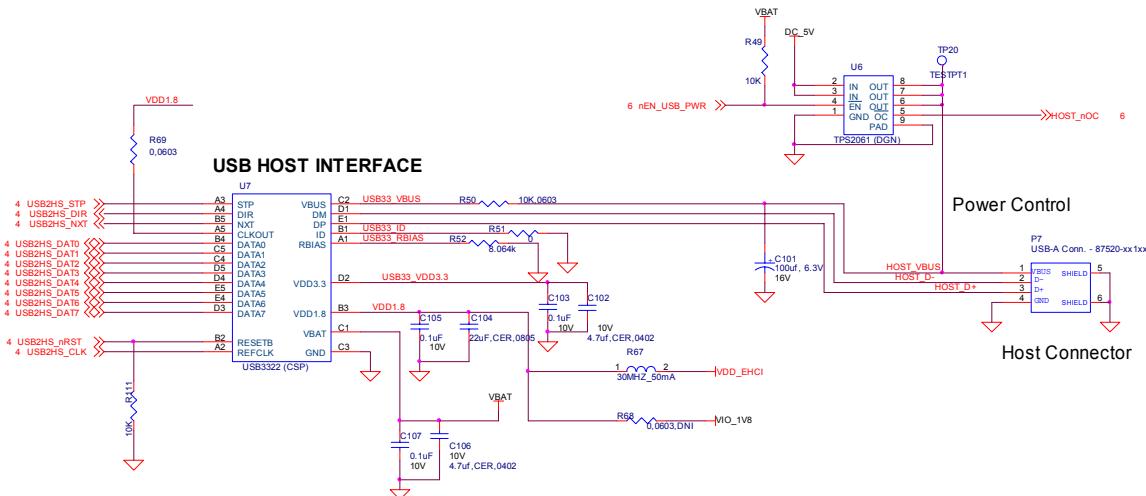


Figure 35. USB Host Design

8.10.1 Host USB OMAP3 Interface

The interface to the OMAP3 is the HSUSB2 interface. The signals used on this interface are contained in **Table 10**.

Table 10. USB Host Port OMAP Signals

Signal	Description	Input/Output
Hsusb2_clk	External transceiver 60-MHz clock output to PHY	O
Hsusb2_stp	External transceiver Stop signal	O
Hsusb2_dir	Transceiver data direction control from PHY	I
Hsusb2_nxt	Next signal from PHY	I
Hsusb2_data0	Bidirectional data bus signal for 12-pin ULPI operation	I/O
Hsusb2_data1	Bidirectional data bus signal for 12-pin ULPI operation	I/O
Hsusb2_data2	Bidirectional data bus signal for 12-pin ULPI operation	I/O
Hsusb2_data3	Bidirectional data bus signal for 12-pin ULPI operation	I/O
Hsusb2_data4	Bidirectional data bus signal for 12-pin ULPI operation	I/O
Hsusb2_data5	Bidirectional data bus signal for 12-pin ULPI operation	I/O
Hsusb2_data6	Bidirectional data bus signal for 12-pin ULPI operation	I/O
Hsusb2_data7	Bidirectional data bus signal for 12-pin ULPI operation	I/O
Gpio_147	Enable/reset line to the USB PHY.	O

The **husb2_clk** signal is an output only and is used to support a HS USB PHY that supports an input clock mode. The SMSC PHY device supports this mode and is used on the Beagle.

8.10.2 Host USB PHY

The PHY used in the design is a USB3322/26 series device from SMSC. The USB3322 is a highly integrated Hi-Speed USB2.0 Transceiver (PHY) that meets all of the electrical

requirements to be used as a Hi-Speed USB Host, Device, or an On-the-Go (OTG) transceiver. In this design, only the host mode of operation is being supported. The USB3322 uses the industry standard UTMI+ Low Pin Interface (ULPI) to connect the USB PHY to the OMAP3. ULPI uses a method of in-band signaling and status byte transfers between the Link and PHY to facilitate a USB session with only 12 pins.

In order to interface to the **OMAP3530**, the device must be used in the 60MHz clock mode. This is done by tying the **CLKOUT** signal on the USB PHY to **VIO_1V8**. On Rev C4, a zero ohm series resistor was added. This is not required, but was added as a “just in case” option if the CLKOUT signal was a source of noise in the PHY. It was proven not to be the case. The clock for the PHY is derived from the 60MHz signal generated by the **OMAP3530**. All of the signals and their functions align with the descriptions found in the **OMAP3530** interface section.

The USB3322 device requires two voltages, the **VIO_1V8** rail to power the I/O rails and the **VBAT**, which needs to be between 3.1V and 5.1V, to power the rest of the device. On the board the **VBAT** is a regulated 4.2V DC. The 3.3V rail for the device is generated internally and requires a filter and bypass cap to be connected externally. Unlike the Rev C3 version, the Rev C4 version derives its 1.8V from the **VAUX2** rail supplied by the **TPS65950** PMIC. It also uses a ferrite bead, R67, to provide additional filtering for noise. This is the fix for the EHCI noise issue. There is an option to connect the 1.8V rail to the **VIO_1V8** rail, but that has not been populated in the Rev C4 design.

The **RBIAS** block in the PHY consists of an internal bandgap reference circuit used for generating the driver current and the biasing of the analog circuits. This block requires an external $8.06\text{K}\Omega$, 1% tolerance, reference resistor connected from **RBIAS** to ground. The nominal voltage at **RBIAS** is 0.8V and therefore the resistor will dissipate approximately $80\mu\text{W}$ of power.

The USB3322 can detect **ID** grounded and **ID** floating to determine if an A or B cable has been inserted. The A plug will ground the **ID** pin while the B plug will float the **ID** pin. As we are not using this device to support the OTG protocol but instead as a host device, we ground the **ID** pin to force it into a Host mode at all times. The **ID** signal is not present on the USB connector.

The USB3322 transceiver fully integrates all of the USB termination resistors on both **DP** and **DM**. This includes $1.5\text{k}\Omega$ pull-up resistors, $15\text{k}\Omega$ pull-down resistors and the 45Ω high speed termination resistors. These resistors require no tuning or trimming.

8.10.3 Host USB Connector

The USB connector used is a Type A receptacle and provides connections for four signals, DP, DM, VBUS, and Ground. This is the same connector you will see on the back of a USB hub. You will notice that there are no external ESD devices on the connector. The ESD protection is integrated into the USB PHY.

8.10.4 Host USB Power Control

Power is provided through the USB Host connector to power devices that are plugged into the USB host connector. This power can be controlled by the **OMAP3530** via the **TPS2061** power switch.

The **TPS2061** power-distribution switch is intended for applications where heavy capacitive loads and short-circuits are likely to be encountered. This device incorporates a 70-mW N-channel MOSFET power switch. Gate drive is provided by an internal charge pump designed to control the power-switch rise times and fall times to minimize current surges during switching. The switch is controlled by the **TPS65950** using the **LED.A** signal. The **OMAP3530** uses the I2C interface to activate the signal in the **TPS65950**.

The amount of available current to be supplied depends on the remaining current available when in USB mode or the DC supply. The switch will not be able to supply more current than is available from the DC source being used.

The **TPS2061** also provides an overcurrent indicator and protection circuit. When the output load exceeds the current-limit threshold or a short is present, the device limits the output current to a safe level by switching into a constant-current mode, pulling the overcurrent (OC) logic output low. This is read by the **TPS65950** via the **CD2** pin. The **CD2** pin can be set to generate an interrupt to the **OMAP3530** to alert it of this condition.

When continuous heavy overloads and short-circuits increase the power dissipation in the switch, causing the junction temperature to rise, a thermal protection circuit shuts off the switch to prevent damage. Recovery from a thermal shutdown is automatic once the device has cooled sufficiently. Internal circuitry ensures that the switch remains off until valid input voltage is present. This power-distribution switch is designed to set current limit at 1.5 A typically. As mentioned, the amount of current available depends on the current source.

8.11 SD/MMC

The board provides an SD/MMC interface. Its primary use is for proving the boot source for SW but it can be used for other things such as cameras and Wireless LAN cards. Typical users prefer to use the USB port for these functions and as such, the SD card function is the primary us of this connector.

The connector supports 7 different types of cards.

- o **SD-** Secure Digital (SD) is a flash memory card format developed by Matsushita, SanDisk and Toshiba for use in portable devices. As of 2007, SD card capacities range from 8 MB to 16 GB. Several companies have announced SD cards with 32 GB. Cards with 4-32 GB are considered high-capacity. The format has proven to be very popular. However, compatibility issues between older devices and the

newer 4 GB and larger cards and the SDHC format have caused considerable confusion for some users. SD card have a write protect tab to prevent the data from being overwritten. SD supports 1-bit SD, 4-bit SD, and SPI modes.

- **miniSD-** Has the same features as the SD with the exceptions that it is in a smaller size and the support for 4-bit mode is optional amongst suppliers.
- **SDIO** - SDIO stands for Secure Digital Input Output. SD slots can actually be used for more than flash memory cards. Devices that support **SDIO** can use small devices designed for the SD form factor, like GPS receivers, Wi-Fi or Bluetooth adapters, modems, Ethernet adapters, barcode readers, IrDA adapters, FM radio tuners, TV tuners, RFID readers, digital cameras, or other mass storage media such as hard drives. SDIO cards are fully compatible with SD Memory Card host controller (including mechanical, electrical, power, signaling and software). When an SDIO card is inserted into a non SDIO-aware host, it will cause no physical damage or disruption to device or host controller. It should be noted that SPI bus topology is mandatory for SDIO, unlike SD Memory and most of the SD Memory commands are not supported in SDIO. **Figure 36** is an example of a SDIO camera card.



Figure 36. Example of an SDIO Card

- **MMC-** The Multi Media Card (**MMC**) is a flash memory card standard. Unveiled in 1997 by Siemens AG and SanDisk, it is based on Toshiba's NAND-based flash memory, and is therefore much smaller than earlier systems based on Intel NOR-based memory such as CompactFlash. MMC is about the size of a postage stamp:

24 mm x 32 mm x 1.4 mm. MMC originally used a 1-bit serial interface, but newer versions of the specification allow transfers of 4 at a time. MMCs are currently available in sizes up to and including 4 GB and 8 GB models.

- **MMCplus**- The version 4.x of the MMC standard, introduced in 2005, brought in two very significant changes to compete against SD cards. These were support for running at higher speeds (26MHz, 52MHz) than the original MMC (20MHz) or SD (25MHz, 50MHz). Version 4.x cards are fully backward compatible with existing readers but require updated hardware/software to use their new capabilities; even though the 4 bit wide bus and high-speed modes of operation are deliberately electrically compatible with SD, the initialization protocol is different, so firmware/software updates are required to allow these features to be enabled when the card is used in an SD reader.
- **MMCmobile** – Is basically the same as MMCplus except that it supports 8 bit data mode.
- **RS-MMC** – This alternate form factor is known as Reduced-Size MultiMediaCard, or RS-MMC, and was introduced in 2004. This form factor is a smaller form factor, of about half the size: 24 mm × 18 mm × 1.4 mm. RS-MMCs are simply smaller MMCs. RS-MMCs are currently available in sizes up to and including 4 GB. Nokia used to use RS-MMC in the Nokia 770 Internet Tablet. **Figure 37** is a side by side comparison of the RS-MMC and MMC card.



Figure 37. RS-MMC and Card

Figure 38 is the SD/MMC interface design on the BeagleBoard.

8.11.1 MMC Power

The SD/MMC connector is supplied power from the **TPS65950** using the **VMMC1** rail. The default setting on this rail is 3.0V as set by the Boot ROM and under SW control, can be set to 1.80V for use with 1.8V cards. The maximum current this rail can provide is

220mA as determined by the TPS65950 regulator. Maximum current can be limited by the overall current available from the USB interface of the PC.

8.11.2 OMAP3530 Interface

There are no external buffers required for the SD/MC operation. The **OMAP3530** provides all of the required interfaces for the SD/MMC interface.

The main features of the MMC/SD/SDIO host controller are:

- Full compliance with MMC command/response sets as defined in the *Multimedia Card System Specification*, v4.0
- Full compliance with SD command/response sets as defined in the *SD Memory Card Specifications*, v1.10d
- Full compliance with SDIO command/response sets and interrupt/read-wait mode as defined in the *SDIO Card Specification, Part E1*, v1.10
- Compliance with sets as defined in the *SD Card Specification, Part A2, SD Host Controller Standard Specification*, v1.00
- Full compliance with MMC bus testing procedure as defined in the *Multimedia Card System Specification*, v4.0
- Full compliance with CE-ATA command/response sets as defined in the *CE-ATA Standard Specification*
- Full compliance with ATA for MMCA specification
- Flexible architecture allowing support for new command structure
- Support:
 - 1-bit or 4-bit transfer mode specifications for SD and SDIO cards
 - 1-bit, 4-bit, or 8-bit transfer mode specifications for MMC cards
- Built-in 1024-byte buffer for read or write
- 32-bit-wide access bus to maximize bus throughput
- Single interrupt line for multiple interrupt source events
- Two slave DMA channels (1 for TX, 1 for RX)
- Programmable clock generation
- Support SDIO Read Wait and Suspend/Resume functions
- Support Stop at block gap
- Support command completion signal (CCS) and command completion signal disable (CCSD) management as specified in the *CE-ATA Standard Specification*

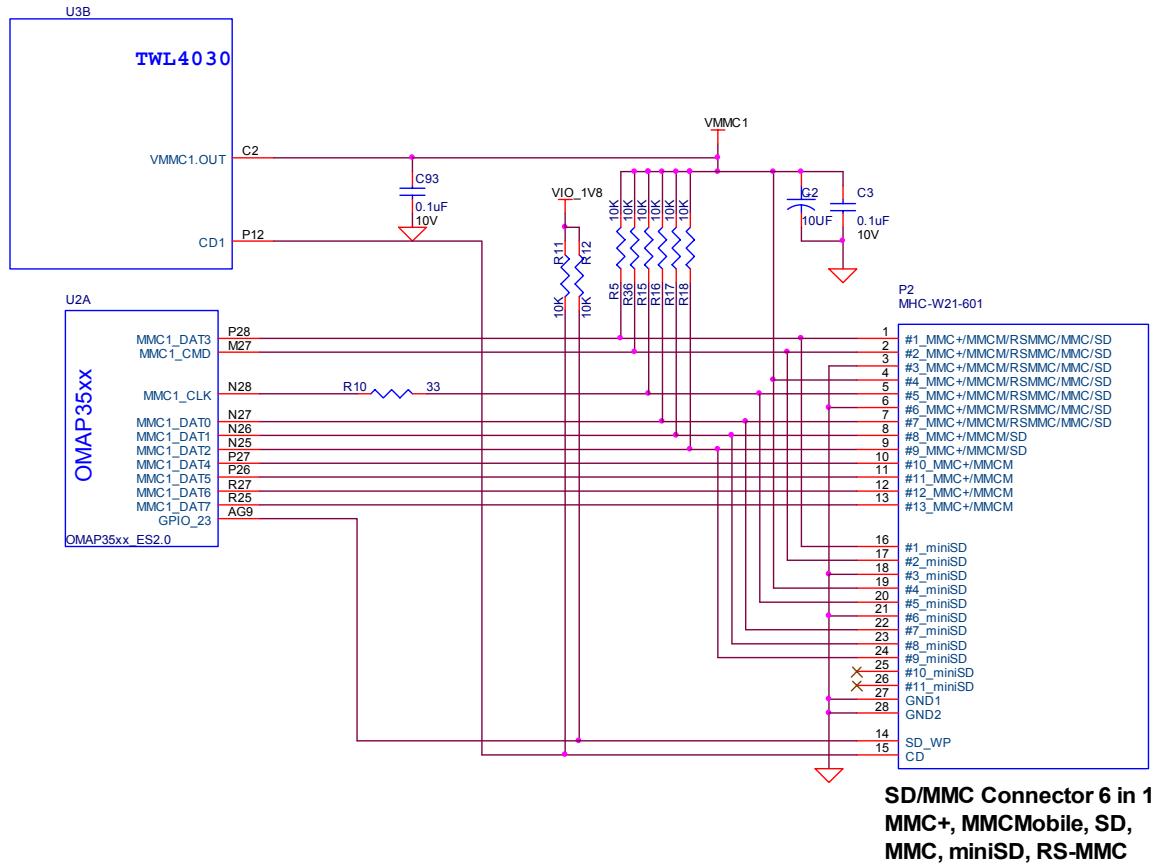


Figure 38. SD/MMC Interface

The known limitations are as follows:

- No built-in hardware support for error correction codes (ECC). See the *Multimedia Card System Specification*, v4.0, and the *SD Memory Card Specifications*, v1.10d, for details about ECC.
- The maximum block size defined in the *SD Memory Card Specifications*, v1.10d states that the host driver can read and write to the buffer in the host controller is 2048 bytes. MMC supports a maximum block size of 1024 bytes. Up to 512 byte transfers, the buffer in MMC is considered as a double buffering with ping-pong management; half of the buffer can be written while the other part is read. For 512 to 1024 byte transfers, the entire buffer is dedicated to the transfer (read only or write only).

Table 11 provides a description of the signals on the MMC card.

Table 11. SD/MMC OMAP Signals

Signal Name	Description	I/O	Pin
MMC1_CLK	SD/MMC Clock output.	O	N28
MMC1_CMD	SD/MMC Command pin	I/O	M27
MMC1_DAT(0..7)	SD/MMC Data pins	I/O	N27,N26,N25,P28,P27, P26,R27,R25
MMC_WP	Write Protect detect	I	AG9

8.11.3 Card Detect

When a card is inserted into the SD/MMC connector, the **Card Detect** pin is grounded. This is detected on pin **P12** of the **TPS65950**. An interrupt, if enabled, is sent to the **OMAP3530** via the interrupt pin. The SW can be written such that the system comes out of sleep or a reduced frequency mode when the card is detected.

8.11.4 Write Protect

If an SD card is inserted into the SD/MMC connector and the write protect pin is active, the Write Detect pin is grounded. This is detected **GPIO_29** of the OMAP3530. The SW can then determine if the card is write protected and act accordingly.

8.11.5 8 Bit Mode

The BeagleBoard also supports the new 8-bit cards. The upper 4 bits are supplied by the VDD_SIM power rail and as such the 8-bit mode is only supported in 1.8V modes. This requires that both the VMMC1 and VDD_SIM rails must be set to 1.8V when using 8 bit cards.

8.11.6 Booting From SD/MMC Cards

The ROM code supports booting from MMC and SD cards with some limitations:

- Support for MMC/SD cards compliant with the Multimedia Card System Specification v4.2 from the MMCA Technical Committee and the Secure Digital I/O Card Specification v2.0 from the SD Association. Including high-capacity (size >2GB) cards: HC-SD and HC MMC.
- 3-V power supply, 3-V I/O voltage on port 1
- Initial 1-bit MMC mode, 4-bit SD mode.
- Clock frequency:
 - Identification mode: 400 kHz
 - Data transfer mode: 20 MHz
- Only one card connected to the bus
- FAT12/16/32 support, with or without master boot sector (MBR).

The high-speed MMC/SD/SDIO host controllers handle the physical layer while the ROM code handles the simplified logical protocol layer (read-only protocol). A limited range of commands is implemented in the ROM code. The MMC/SD specification defines two operating voltages for standard or high-speed cards. The ROM code only supports standard operating voltage range (3-V) (both modes supported). The ROM code reads out a booting file from the card file system and boots from it.

8.12 Audio Interface

The BeagleBoard supports stereo in and out through the **TPS65950** which provides the audio CODEC.

Figure 39 is the Audio circuitry design on the BeagleBoard.

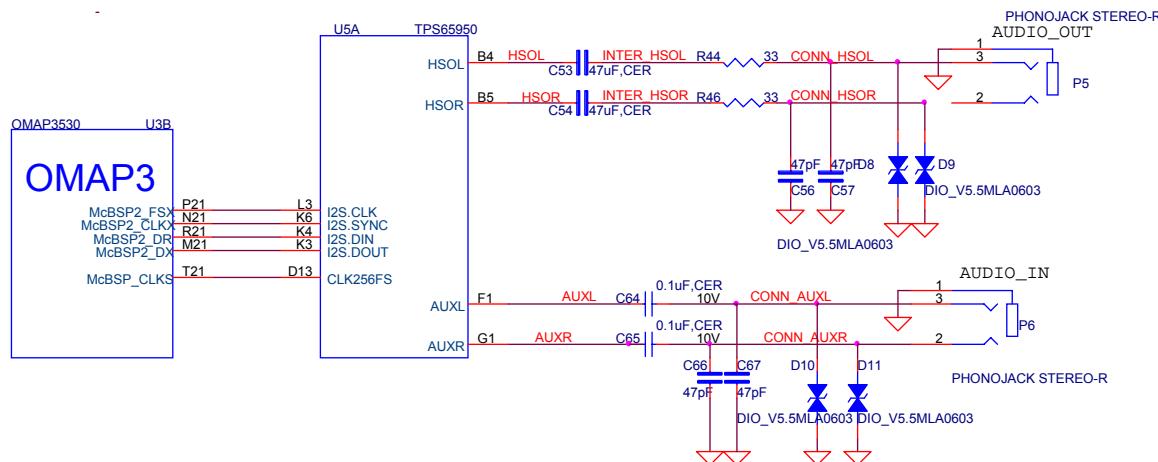


Figure 39. Audio Circuitry

8.12.1 OMAP3530 Audio Interface

There are five McBSP modules called McBSP1 through McBSP5 on the OMAP3530. **McBSP2** provides a full-duplex, direct serial interface between CODEC inside the **TPS65950**. It supports the I2S format to the TPS65950. In **Table 12** are the signals used on the **OMAP3530** to interface to the CODEC.

Table 12. OMAP3530 Audio Signals

Signal Name	Description	I/O	Pin
mcbsp2_dr	Received serial data	I	R21
mcbsp2_dx	Transmitted serial data	I/O	M21
mcbsp2_clkx	Combined serial clock	I/O	N21
mcbsp2_fsx	Combined frame synchronization	I/O	P21
Mcbsp_clks	External clock input. Used to synchronize with the TPS65950	I	T21

8.12.2 TPS65950 Audio Interface

The **TPS65950** acts as a master or a slave for the I2S interface. If the **TPS65950** is the master, it must provide the frame synchronization (I2S_SYNC) and bit clock (I2S_CLK) to the **OMAP3530**. If it is the slave, the **TPS65950** receives frame synchronization and bit clock. The TPS65950 supports the I2S left-justified and right-justified data formats, but doesn't support the TDM slave mode.

In **Table 13** are all the signals used to interface to the **OMAP3530**.

Table 13. OMAP3530 Audio Signals

Signal Name	Description	I/O	Pin
I2S.CLK	Clock signal (audio port)	I/O	L3
I2S.SYNC	Synchronization signal (audio port)	IO	K6
I2S.DIN	Data receive (audio port)	I	K4
I2S.DOUT	Data transmit (audio port)	O	K3
CLK256FS	Synchronization frame sync to the OMAP3530	O	D13

8.12.3 Audio Output Jack

A single 3.5mm jack is provided on BeagleBoard to support external stereo audio output devices such as headphones and powered speakers.

8.12.4 Audio Input Jack

A single 3.5mm jack is supplied to support external audio inputs including stereo or mono.

8.13 DVI-D Interface

The LCD interface on the **OMAP3530** is accessible from the **DVI-D** interface connector on the board. **Figure 40** is the DVI-D interface design.

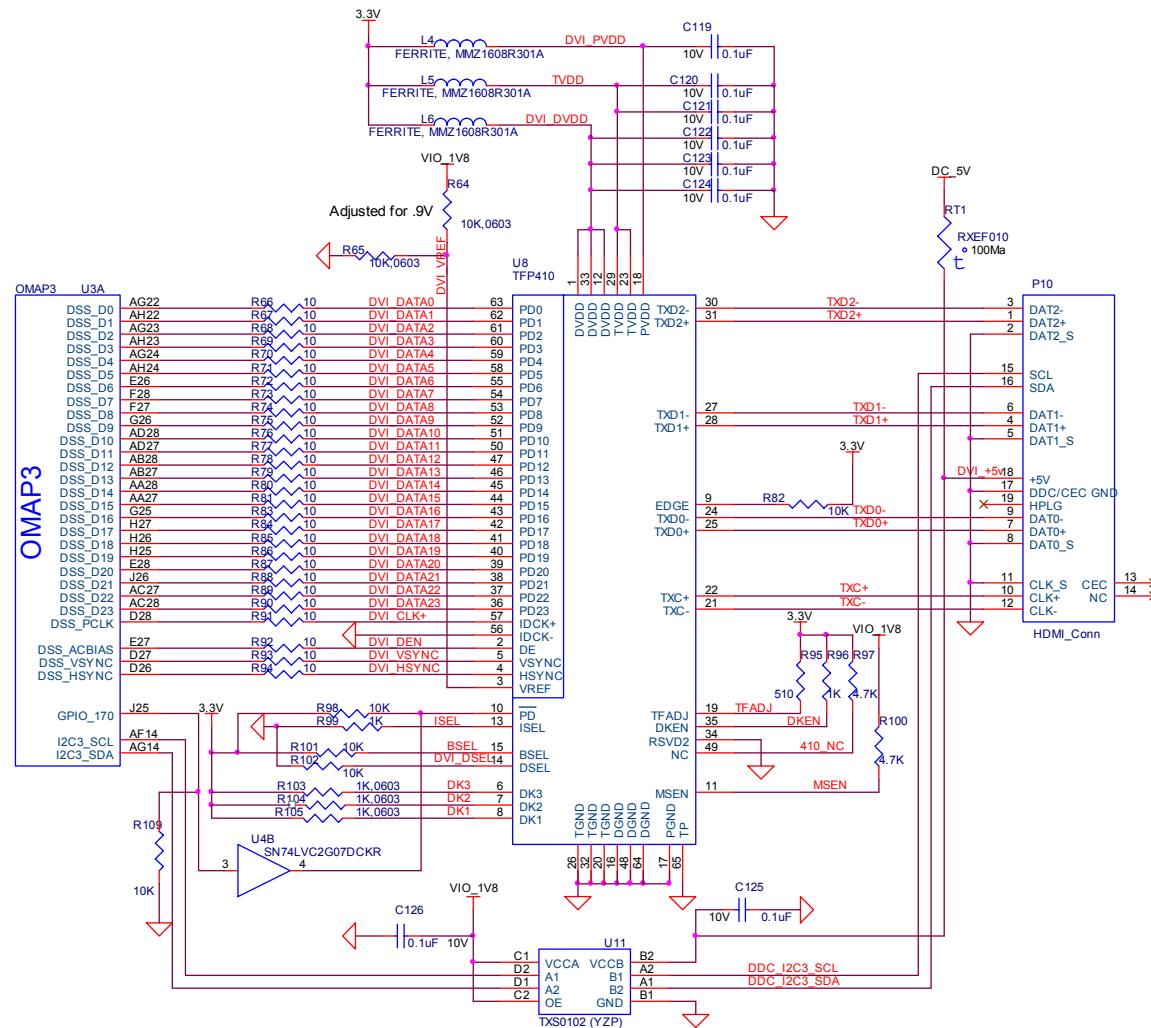


Figure 40. DVI-D Interface

8.13.1 OMAP3530 LCD Interface

The main driver for the DVI-D interface originates at the **OMAP3530** via the **DSS** pins. The **OMAP3530** provides 24 bits of data to the DVI-D framer chip, **TFP410**. There are three other signals used to control the DVI-D that originate at the **OMAP3530**. These are **I2C3_SCL**, **I2C3_SDA**, and **GPIO_170**. All of the signals used are described in **Table 14**.

Table 14. OMAP3530 LCD Signals

Signal Name	Description	Type	Ball
dss_pclk	LCD Pixel Clock	O	D28
dss_hsync	LCD Horizontal Synchronization	O	D26
dss_vsync	LCD Vertical Synchronization	O	D27
dss_acbias	Pixel data enable (TFT) output	O	E27
dss_data0	LCD Pixel Data bit 0	BLUE0	O
dss_data1	LCD Pixel Data bit 1	BLUE1	O
dss_data2	LCD Pixel Data bit 2	BLUE2	O
dss_data3	LCD Pixel Data bit 3	BLUE3	O
dss_data4	LCD Pixel Data bit 4	BLUE4	O
dss_data5	LCD Pixel Data bit 5	BLUE5	O
dss_data6	LCD Pixel Data bit 6	BLUE6	O
dss_data7	LCD Pixel Data bit 7	BLUE7	O
dss_data8	LCD Pixel Data bit 8	GREEN0	O
dss_data9	LCD Pixel Data bit 9	GREEN1	O
dss_data10	LCD Pixel Data bit 10	GREEN2	O
dss_data11	LCD Pixel Data bit 11	GREEN3	O
dss_data12	LCD Pixel Data bit 12	GREEN4	O
dss_data13	LCD Pixel Data bit 13	GREEN5	O
dss_data14	LCD Pixel Data bit 14	GREEN6	O
dss_data15	LCD Pixel Data bit 15	GREEN7	O
dss_data16	LCD Pixel Data bit 16	RED0	O
dss_data17	LCD Pixel Data bit 17	RED1	O
dss_data18	LCD Pixel Data bit 18	RED2	O
dss_data19	LCD Pixel Data bit 19	RED3	O
dss_data20	LCD Pixel Data bit 20	RED4	O
dss_data21	LCD Pixel Data bit 21	RED5	O
dss_data22	LCD Pixel Data bit 22	RED6	O
dss_data23	LCD Pixel Data bit 23	RED7	O
GPIO_170	Powers down the TFP410 when Lo. TFP410 is active when Hi.	O	J25
I2C3_SCL	I2C3 clock line. Used to communicate with the monitor to determine setting information.	I/O	AF14
I2C3_SDA	I2C3 data line. Used to communicate with the monitor to determine setting information.	I/O	AG14

10ohm series resistors are provide in the signal path to minimize reflections in the high frequency signals from the **OMAP3530** to the **TFP410**. These resistors are in the form of resistor packs on the BeagleBoard. The maximum clock frequency of these signals is 65MHz.

8.13.2 OMAP3530 LCD Power

In order for the DSS outputs to operate correctly out of the **OMAP3530**, two voltage rails must be active, **VIO_1V8** and **VDD_PLL2**. Both of these rails are controlled by the **TPS65950** and must be set to 1.8V. By default, **VDD_PLL2** is not turned and must be activated by SW. Otherwise some of the bits will not have power supplied to them.

8.13.3 TFP410 Framer

The **TFP410** provides a universal interface to allow a glue-less connection to provide the DVI-D digital interface to drive external LCD panels. The adjustable 1.1-V to 1.8-V digital interface provides a low-EMI, high-speed bus that connects seamlessly with the 1.8V and 24-bit interface output by the **OMAP3530**. The DVI interface on the BeagleBoard supports flat panel display resolutions up to XGA at 65 MHz in 24-bit true color pixel format.

Table 15 is a description of all of the interface and control pins on the **TFP410** and how they are used on BeagleBoard.

Table 15. TFP410 Interface Signals

Signal Name	Description	Type	Ball
DATA[23:12]	The upper 12 bits of the 24-bit pixel bus.	I	36–47
DATA[11:0]	The bottom 12 bits of the 24-bit pixel bus.	I	50–55, 56–53
IDCK+	Single ended clock input.	I	57
IDCK-	Tied to ground to support the single ended mode.	I	56
DE	Data enable. During active video (DE = high), the transmitter encodes pixel data, DATA[23:0]. During the blanking interval (DE = low), the transmitter encodes HSYNC and VSYNC.	I	2
HSYNC	Horizontal sync input	I	4
VSYNC	Vertical sync input	I	5
DK3	These three inputs are the de-skew inputs DK[3:1], used to adjust the setup and hold times of the pixel data inputs DATA[23:0], relative to the clock input IDCK±.	I	6
DK2		I	7
DK1		I	8
MSEN	A low level indicates a powered on receiver is detected at the differential outputs. A high level indicates a powered on receiver is not detected.	O	11
ISEL	This pin disables the I2C mode on chip. Configuration is specified by the configuration pins (BSEL, DSEL, EDGE, VREF) and state pins (PD, DKEN).	I	13
BSEL	Selects the 24bit and single-edge clock mode.	I	13
DSEL	Lo to select the single ended clock mode.	I	14
EDGE	A high level selects the primary latch to occur on the rising edge of the input clock IDCK	I	9
DKEN	A HI level enables the de-skew controlled by DK[1:3]	I	35
VREF	Sets the level of the input signals from the OMAP3530.	I	3
PD	A HI selects normal operation and a LO selects the powerdown mode.	I	10
TGADJ	This pin controls the amplitude of the DVI output voltage swing, determined by the value of the pullup resistor RTFADJ connected to 3.3V.	I	19

8.13.4 TFP410 Power

Power to the TFP410 is supplied from the 3.3V regulator in **U1**, the **TPS2141**. In order to insure a noise free signal, there are three inductors, **L4**, **L5**, and **L6** that are used to filter the 3.3V rail into the TFP410.

8.13.5 TFP410 Control Pins

There are twelve control pins that set up the TFP410 to operate with the **OMAP3530**. Most of these pins are set by HW and do not require any intervention by the **OMAP3530** to set them.

8.13.5.1 ISEL

The **ISEL** pin is pulled LO via **R99** to place the TFP410 in the control pin mode with the I2C feature disabled. This allows the other modes for the TFP410 to be set by the other control pins.

8.13.5.2 BSEL

The **BSEL** pin is pulled HI to select the 24 bit mode for the Pixel Data interface from the **OMAP3530**.

8.13.5.3 DSEL

The **DSEL** pin is pulled low to select the single ended clock mode from the **OMAP3530**.

8.13.5.4 EDGE

The **EDGE** signal is pulled HI through **R82** to select the rising edge on the IDCK+ lead which is the pixel clock from the **OMAP3530**.

8.13.5.5 DKEN

The **DKEN** signal is pulled HI to enable the de-skew pins. The de-skew pins, **DK1-DK3**, are pulled low by the internal pulldown resistors in the **TFP410**. This is the default mode of operation. If desired, the resistors can be installed to pull the signals high. However, it is not expected that any of the resistors will need to be installed. The DK1-DK3 pins adjust the timing of the clock as it relates to the data signals.

8.13.5.6 MSEN

The **MSEN** signal, when low, indicates that there is a powered monitor plugged into the DVI-D connector. This signal is not connected to the **OMAP3530** and is provided as a test point only.

8.13.5.7 VREF

The **VREF** signal sets the voltage level of the **DATA**, **VSYNC**, **HSYNC**, **DE**, and **IDCK+** leads from the OMAP3530. As the **OMAP3530** is 1.8V, the level is set to .9V by **R64** and **R65**.

8.13.5.8 PD

The **PD** signal originates from the **OMAP3530** on the **GPIO_170** pin. Because the **PD** signal on the **TFP410** is 3.3V referenced, this signal must be converted to **3.3V**. This is done by **U4**, **SN74LVC2G07**, a non-inverting open drain buffer. If the **GPIO_170** pin is HI, then the open drain signal is inactive, causing the signal to be pulled HI by **R98**. When **GPIO_170** is taken low, the output of **U4** will also go LO, placing the **TFP410** in the power down mode. Even though **U4** is running at 1.8V to match the **OMAP3530**, the output will support being pulled up to **3.3V**. On power up, the TFP410 is disabled by **R109**, a 10K resistor. When the **OMAP3530** powers on, pin **J25** comes in the safe mode, meaning it is not being driven. **R109** insures that the signal is pulled LO, putting the TFP410 in the power down mode.

8.13.5.9 TFADJ

The **TFADJ** signal controls the amplitude of the DVI output voltage swing, determined by the value of **R95**.

8.13.5.10 RSVD2

This unused pin is terminated to ground as directed by the TFP410 data manual.

8.13.5.11 NC

This unused pin is pulled HI as directed by the TFP410 data manual.

8.13.6 DVI-D Connector

In order to minimize board size, a HDMI connector was selected for the DVI-D connection. The BeagleBoard does not support HDMI but only the DVI-D component of HDMI. The Cable is not supplied with the BeagleBoard but is available from numerous cable suppliers and is required to connect a display to the BeagleBoard.

8.13.6.1 Shield Wire

Each signal has a shield wire that is used in the cable to provide signal protection for each differential pair. This signal is tied directly to ground.

8.13.6.2 DAT0+/DAT0-

The differential signal pair **DAT0+/DAT0-** transmits the 8-bit blue pixel data during active video and HSYNC and VSYNC during the blanking interval.

8.13.6.3 DAT1+/DAT1-

The differential signal pair **DAT1+/DAT1-** transmits the 8-bit green pixel data during active video.

8.13.6.4 DAT2+/DAT2-

The differential signal pair **DAT2+/DAT2-** transmits the 8-bit red pixel data during active.

8.13.6.5 TXC+/TXC-

The differential signal pair **TXC+/TXC-** transmits the differential clock from the TFP410.

8.13.6.6 DDC Channel

The **Display Data Channel** or **DDC** (sometimes referred to as EDID Enhanced Display ID) is a digital connection between a computer display and the **OMAP3530** that allows the display specifications to be read by the **OMAP3530**. The standard was created by the Video Electronics Standards Association (VESA). The current version of DDC, called DDC2B, is based on the I²C bus. The monitor contains a read-only memory (ROM) chip programmed by the manufacturer with information about the graphics modes that the monitor can display. This interface in the LCD panel is powered by the +5V pin on the connector through **RT1**, a resetable fuse. As the **OMAP3530** is 1.8V I/O, the I²C bus is level translated by **U11**, a **TXS0102**. It provides for a split rail to allow the signals to interface on both sides of the circuit. Inside of **TXS0102** is a pullup on each signal, removing the need for an external resistor.

8.13.6.7 HDMI Support

The digital portion of the DVI-D interface is compatible with HDMI and is electrically the same. A standard HDMI cable may be used to connect to the HDMI input of monitors or televisions. Whether or not the Beagle will support those monitors is dependent on the timings that are used on the BeagleBoard and those that are accepted by the monitor. This may require a change in the software running on the Beagle. The audio and encryption features of HDMI are not supported by the BeagleBoard.

8.13.6.8 DVI to VGA

The analog portion of DVI which provides RGB analog signals is **not supported** by the BeagleBoard. Buying a DVI to VGA adapter connector will not work on a VGA display. You will need an active DVI-D to VGA adapter. Another option for these signals is to find a board that connects to the J4 and J5 expansion connectors and generates the RGB signals for the VGA display.

8.14 LCD Expansion Headers

Access is provided on the Rev C4 to allow access to the LCD signals. **Table 16** shows the signals that are on the J4 connector. You will notice that the signals are not in a logical order or grouping. This is due to the routing on the PCB where we allowed the routing to take president to get it to route with no addition of layers to the design.

Table 16. J4 LCD Signals

Pin#	Signal	I/O	Description
1	DC_5V	PWR	DC rail from the Main DC supply
2	DC_5V	PWR	DC rail from the Main DC supply
3	DVI_DATA1	O	LCD Pixel Data bit
4	DVI_DATA0	O	LCD Pixel Data bit
5	DVI_DATA3	O	LCD Pixel Data bit
6	DVI_DATA2	O	LCD Pixel Data bit
7	DVI_DATA5	O	LCD Pixel Data bit
8	DVI_DATA4	O	LCD Pixel Data bit
9	DVI_DATA12	O	LCD Pixel Data bit
10	DVI_DATA10	O	LCD Pixel Data bit
11	DVI_DATA23	O	LCD Pixel Data bit
12	DVI_DATA14	O	LCD Pixel Data bit
13	DVI_DATA19	O	LCD Pixel Data bit
14	DVI_DATA22	O	LCD Pixel Data bit
15	I2C3_SDA	I/O	I2C3 Data Line
16	DVI_DATA11	O	LCD Pixel Data bit
17	DVI_VSYNC	O	LCD Vertical Sync Signal
18	DVI_PUP	O	Control signal for the DVI controller. When Hi, DVI is enabled. Can be used to activate circuitry on adapter board if desired.
19	GND	PWR	Ground bus
20	GND	PWR	Ground bus

The current available on the DC_5V rail is limited to the available current that remains from the DC supply that is connected to the DC power jack on the board. Keep in mind that some of that power is needed by the USB Host power rail and if more power is

needed for the expansion board, the main DC power supply current capability may need to be increased. All signals are 1.8V except the DVI_PUP which is a 3.3V signal.

Table 17 shows the signals that are on connector J5.

Table 17. J5 LCD Signals

Pin#	Signal	I/O	Description
1	3.3V	PWR	3.3V reference rail
2	VIO_1V8	PWR	1.8V buffer reference rail.
3	DVI_DATA20	O	LCD Pixel Data bit
4	DVI_DATA21	O	LCD Pixel Data bit
5	DVI_DATA17	O	LCD Pixel Data bit
6	DVI_DATA18	O	LCD Pixel Data bit
7	DVI_DATA15	O	LCD Pixel Data bit
8	DVI_DATA16	O	LCD Pixel Data bit
9	DVI_DATA7	O	LCD Pixel Data bit
10	DVI_DATA13	O	LCD Pixel Data bit
11	DVI_DATA8	O	LCD Pixel Data bit
12	NC		No connect
13	DVI_DATA9		LCD Pixel Data bit
14	I2C3_SCL	I/O	I2C3 Clock Line
15	DVI_DATA6	O	LCD Pixel Data bit
16	DVI_CLK+	O	DVI Clock
17	DVI_DEN	O	Data Enable
18	DVI_HSYNC	O	Horizontal Sync
19	GND	PWR	Ground bus
20	GND	PWR	Ground bus

The 1.8V rail is for level translation only and should not be used to power circuitry on the board. The 3.3V rail also has limited capacity on the power as well. If the **TFP410** is disabled on the Beagle, then 80mA is freed up for use on an adapter card connected to the LCD signals connectors. It is not required that the **TFP410** be disabled when running an adapter card, but the power should be taken into consideration when making this decision.

It is suggested that the 5V rail be used to generate the required voltages for an adapter card.

8.15 S-Video

A single S-Video port is provided on the BeagleBoard. **Figure 41** is the design of the S-Video interface.

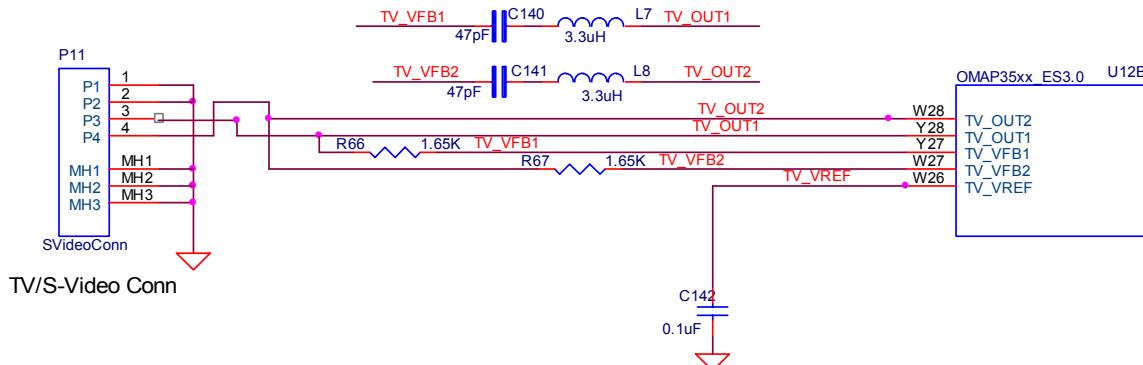


Figure 41. S-Video Interface

Table 18 is the list of the signals on the S-Video interface and their definitions.

Table 18. S-Video Interface Signals

Signal	I/O	Description
tv_out1	O	TV analog output composite
tv_out2	O	TV analog output S-VIDEO
tv_vref	I	Reference output voltage from internal bandgap
tv_vfb1	O	Amplifier feedback node
tv_vfb2	O	Amplifier feedback node

Power to the internal DAC is supplied by the **TPS65950** via the **VDAC_1V8** rail. **Figure 37** reflects the filtering that is used on these rails, including the input VBAT rail.

A **47pf** CAP and **3.3uh** inductor are across the feedback resistors to improve the quality of the S-Video signal.

8.16 RS232 Port

A single RS232 port is provided on the BeagleBoard and provides access to the TX and RX lines of **UART3** on the OMAP3530. **Figure 42** shows the design of the RS232 port.

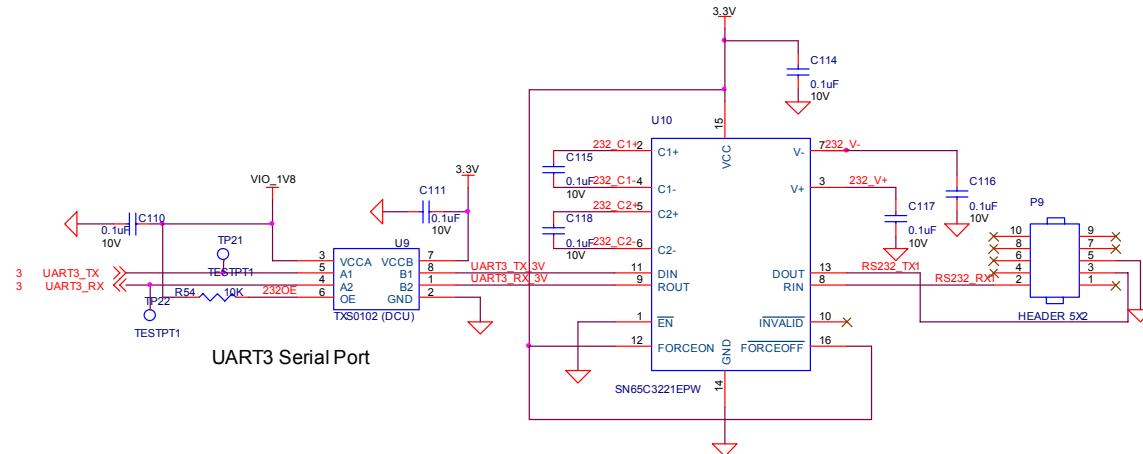


Figure 42. RS232 Interface Design

8.16.1 OMAP3530 Interface

Two lines, **UART3_Tx** and **UART3_Rx**, are provided by the **OMAP3530**. The **UART3** function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate and also supports auto bauding.

8.16.2 OMAP3530 Level Translator

All of the I/O levels from the **OMAP3530** are **1.8V** while the transceiver used runs at **3.3V**. This requires that the voltage levels be translated. This is accomplished by the **TXS0102** which is a two-bit noninverting translator that uses two separate configurable power-supply rails. The A port tracks VCCA, 1.8V and the B port tracks VCCB, 3.3V. This allows for low-voltage bidirectional translation between the two voltage nodes. When the output-enable (OE) input is low, all outputs are placed in the high-impedance state. In this design, the OE is tied high via a 10K ohm resistor to insure that it is always on.

8.16.3 RS232 Transceiver

The RS232 transceiver used is the **SN65C322** which consists of one line driver, one line receiver, and a dual charge-pump circuit with $\pm 15\text{-kV}$ IEC ESD protection pin to pin (serial-port connection pins, including GND). These devices provide the electrical interface between an asynchronous communication controller and the serial-port connector. The charge pump and four small external capacitors allow operation from a

single 3-V to 5.5-V supply. The **SN65C3221** operates at data signaling rates up to 1 Mbit/s and a driver output slew rate of 24 V/ms to 150 V/ms. While the **OMAP3530** can easily drive a 1Mbit/S rate, your results may vary based on cabling, distance, and the loads and drive capability on the other end of the RS232 port.

The transceiver is powered from the 3.3V rail and is active at power up. This allows the port to be used for UART based peripheral booting over the port.

8.16.4 Connector

Access to the RS232 port is through a 10pin header, **P9**. Connection to the header is through a 10 pin IDC to 9 pin D-sub cable. This header requires the use of an **ATI-Everex** type cable. This is the only cable that will work. This cable is readily available from a number of sources and is commonly found on many PC motherboards and is not supplied with the BeagleBoard. **Figure 43** is a picture of what the cable assembly looks like.

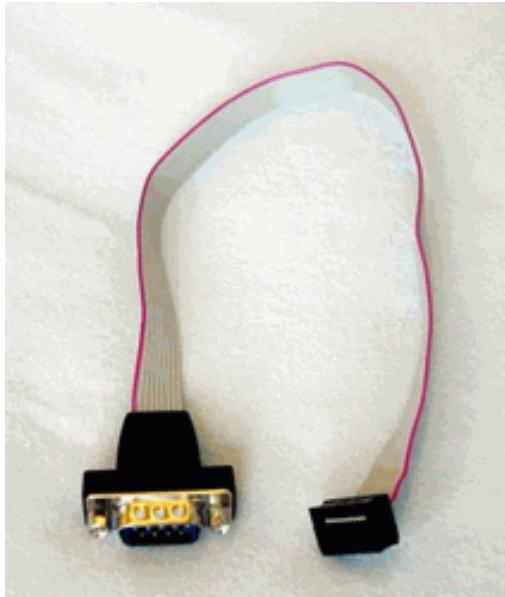


Figure 43. RS232 Cable

When purchasing, make sure the ATI-Everex or pass through cable is ordered.

8.17 Indicators

There are four green indicators on the BeagleBoard:

- Power
- PMU_STAT
- USER0
- USER1

Three of these are programmable under software control and the fourth one is tied to the main power rail. **Figure 44** shows the connection of all of these indicators.

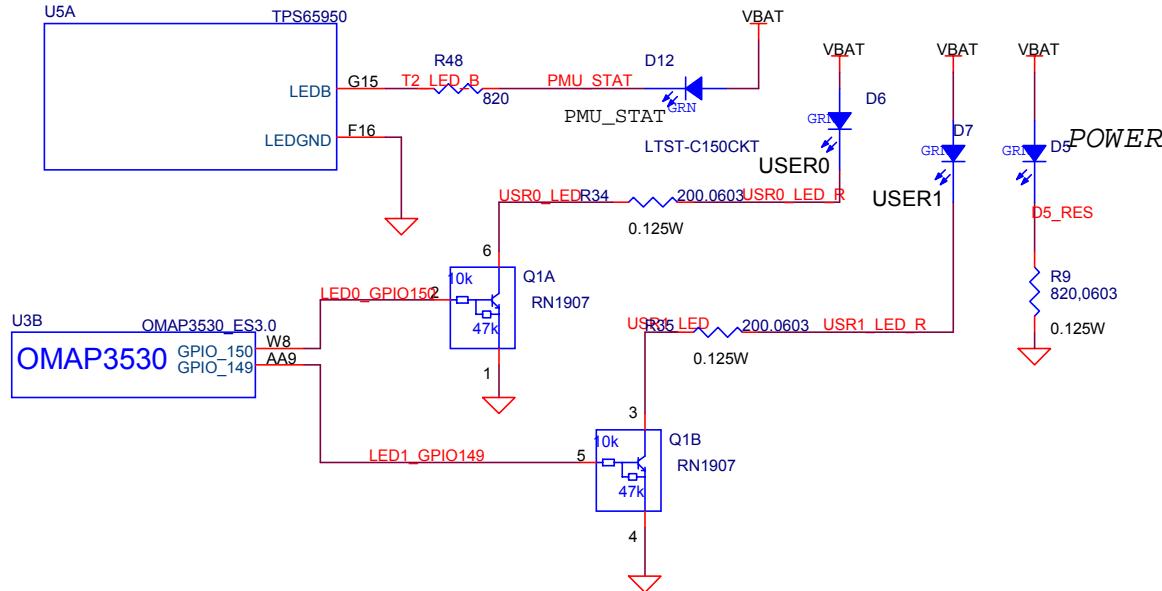


Figure 44. Indicator Design

8.17.1 Power Indicator

This indicator, **D7**, connects across the **VBAT** supply and ground. It indicates that the entire power path is supplying the power to the board. The **VBAT** regulator can be driven from either the USB Client port or an external 5VDC power supply. Indicator **D7** does not indicate which power source is being used to supply the main power to the board but only that it is active.

8.17.2 PMU Status Indicator

This output is driven from the **TPS65950** using the **LED.B** output. The **TPS65950** provides LED driver circuitry to power two LED circuits that can provide user indicators. The first circuit can provide up to 160 mA and the second, 50 mA. Each LED circuit is independently controllable for basic power (on/off) control and illumination level (using PWM). The second driver, **LED.B**, is used to drive an LED that is connected to the **VBAT** rail through a resistor.

The PWM inside the **TPS65950** can be used to alter the brightness of the LED if desired or it can be turned on or off by the **OMAP3530** using the I2C bus. The PWM is programmable, register-controlled, duty cycle based on a nominal 4-Hz cycle which is

derived from an internal 32-kHz clock. It is possible to set the LED to flash automatically without software control if desired.

8.17.3 User Indicators

There are two user LEDs that can be driven directly from a GPIO pin on the **OMAP3530**. These can be used for any purpose by the software. The output level of the **OMAP3530** is 1.8V and the current sink capability is not enough to drive an LED with any level of brightness. A transistor pair, **RN1907** is used to drive the LEDs from the **V_{BAT}** rail. A logic level of 1 will turn the LED on.

8.18 JTAG

A JTAG header is provided to allow for advanced debugging on the BeagleBoard by using a JTAG based debugger **Figure 45** shows the interconnection to the **OMAP3530** processor.

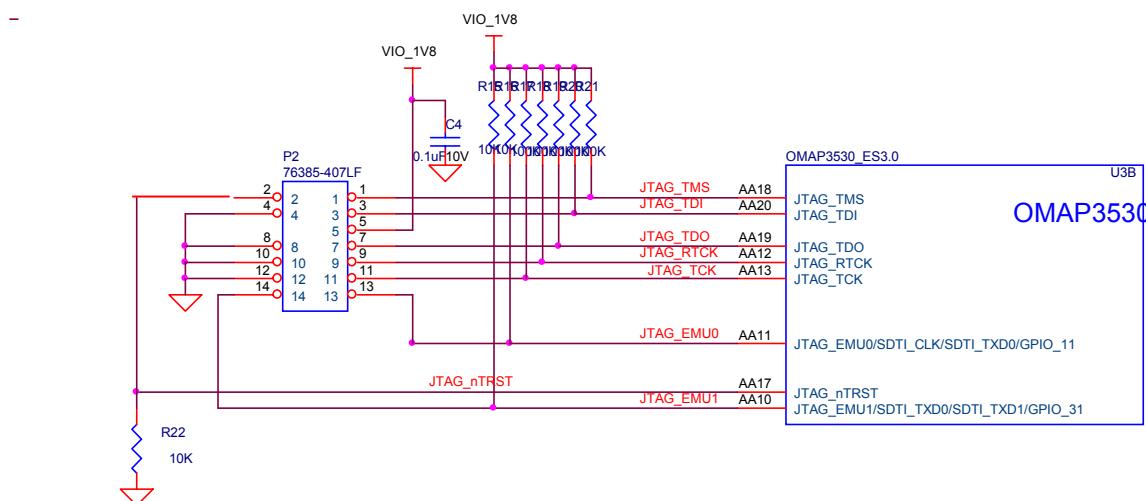


Figure 45. JTAG Interface

8.18.1 OMAP3530 Interface

The JTAG interface connects directly to the OMAP processor. All signals are a 1.8V level. **Table 19** describes the signals on the JTAG connector.

Table 19. JTAG Signals

Signal	Description	I/O
JTAG_TMS	Test mode select	I/O
JTAG_TDI	Test data input	I
JTAG_TDO	Test Data Output	O
JTAG_RTCK	ARM Clock Emulation	O
JTAG_TCK	Test Clock	I
JTAG_nTRST	Test reset	I
JTAG_EMU0	Test emulation 0	I/O
JTAG_EMU1	Test emulation 1	I/O

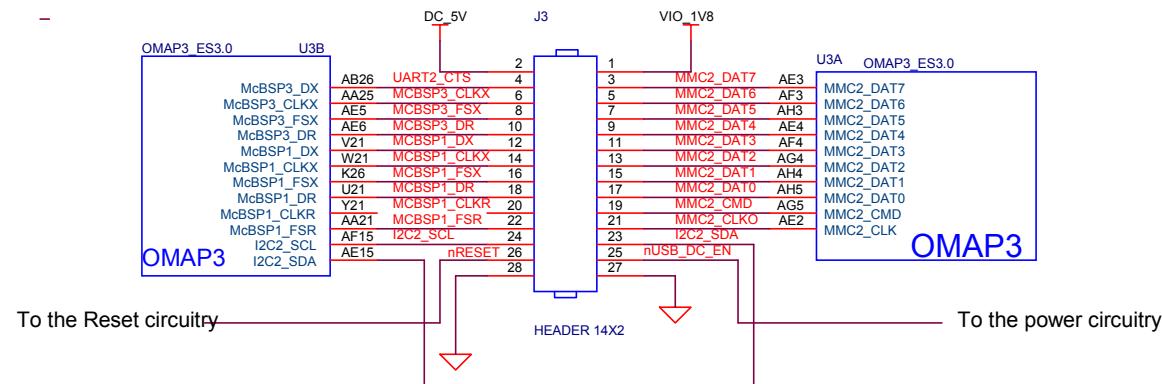
8.18.2 Connector

The JTAG interface uses a 14 pin connector. All JTAG emulator modules should be able to support this interface. Contact your emulator supplier for further information or if an adapter is needed.

8.19 Expansion Header

The expansion header is provided to allow a limited number of functions to be added to the BeagleBoard via the addition of a daughtercard.

Figure 46 is the design of the expansion connector and the interfaces to the OMAP3530.

**Figure 46. Expansion Header**

NOTE: The Expansion header itself is NOT provided on the BeagleBoard. This is user installed option. This header is not populated on the BeagleBoard so that based on the usage scenario the user chooses; it can be populated as needed (Top, Bottom, Top right angle, or Bottom Right angle). The user should take care in installing this header.

CAUTION: The voltage levels on the expansion header are 1.8V. Exposure of these signals to a higher voltage will result in damage to the board and a voiding of the warranty.

8.19.1 OMAP3530 Interface

The main purpose of the expansion connector is to route additional signals from the **OMAP3530** processor. **Table 20** shows all of the signals that are on the expansion header. As the **OMAP3530** has a multiplexing feature, multiple signals can be connected to certain pins to add additional options as it pertains to the signal available. Each pin can be set individually for a different mux mode. This allows any of the listed mux modes to be set on a pin by pin basis by writing to the pin mux register in software. Following is the legend for **Table 20**.

X= there is no signal connected when this mode is selected

Z= this is the safe mode meaning neither input to output. This is the default mode on power up.

*****= this indicates that there is a signal connected when this mode is selected, but it has no useful purpose without other pins being available. Access to these other pins is not provided on the expansion connector.

The first column is the pin number of the expansion connector.

The second column is the pin number of the OMAP3530 processor.

The columns labeled 0-7 represent each of the pin mux modes for that pin. By setting this value in the control register, this signal will be routed to the corresponding pin of the expansion connector. These setting are on a pin by pin basis. Any pin can be set with the mux register setting, and the applicable signal will be routed to the pin on the expansion connector.

Table 20. Expansion Connector Signals

EXP	OMAP	0	1	2	3	4	5	6	7
1				VIO_1V8					
2				DC_5V					
3	AE3	MMC2_DAT7	*	*	*	GPIO_139	*	*	Z
4	AB26	UART2_CTS	McBSP3_RX	GPT9_PWMEVT	X	GPIO_144	X	X	Z
5	AF3	MMC2_DAT6	*	*	*	GPIO_138	*	X	Z
6	AA25	UART2_TX	McBSP3_CLKX	GPT11_PWMEVT	X	GPIO_146	X	X	Z
7	AH3	MMC2_DAT5	*	*	*	GPIO_137	*	X	Z
8	AE5	McBSP3_FSX	UART2_RX	X	X	GPIO_143	*	X	Z
9	AE4	MMC2_DAT4	*	X	*	GPIO_136	X	X	Z
10	AB25	UART2_RTS	McBSP3_DR	GPT10_PWMEVT	X	GPIO_145	X	X	Z
11	AF4	MMC2_DAT3	McSPI3_CS0	X	X	GPIO_135	X	X	Z
12	V21	McBSP1_RX	McSPI4_SIMO	McBSP3_RX	X	GPIO_158	X	X	Z
13	AG4	MMC2_DAT2	McSPI3_CS1	X	X	GPIO_134	X	X	Z
14	W21	McBSP1_CLKX	X	McBSP3_CLKX	X	GPIO_162	X	X	Z
15	AH4	MMC2_DAT1	X	X	X	GPIO_133	X	X	Z
16	K26	McBSP1_FSX	McSPI4_CS0	McBSP3_FSX	x	GPIO_161	X	X	Z
17	AH5	MMC2_DAT0	McSPI3_SOMI	X	X	GPIO_132	X	X	Z
18	U21	McBSP1_DR	McSPI4_SOMI	McBSP3_DR	X	GPIO_159	X	X	Z
19	AG5	MMC2_CMD	McSPI3_SIMO	X	X	GPIO_131	X	X	Z
20	Y21	McBSP1_CLKR	McSPI4_CLK	X	X	GPIO_156	X	X	Z
21	AE2	MMC2_CLKO	McSPI3_CLK	X	X	GPIO_130	X	X	Z
22	AA21	McBSP1_FSR	X	*	Z	GPIO_157	X	X	Z
23	AE15	I2C2_SDA	X	X	X	GPIO_183	X	X	Z
24	AF15	I2C2_SCL	X	X	X	GPIO_168	X	X	Z
25	25			REGEN					
26	26			nRESET					
27	27			GND					
28	28			GND					

8.19.2 Expansion Signals

This section provides more detail on each of the signals available on the expansion connector. They are grouped by functions in **Table 21** along with a description of each signal and the MUX setting to activate the pin. If you use these signals in their respective groups and that is the only function you use, all of the signals are available. Whether or not the signals you need are all available, depends on the muxing function on a per-pin basis. Only one signal per pin is available at any one time.

Table 21. Expansion Connector Signal Groups

Signal	Description	I/O	EXP	OMAP	Mux
SD/MMC Port 2					
MMC2_DAT7	SD/MMC data pin 7.	I/O	3	AE3	1
MMC2_DAT6	SD/MMC data pin 6.	I/O	5	AF3	1
MMC2_DAT5	SD/MMC data pin 5.	I/O	7	AH3	1
MMC2_DAT4	SD/MMC data pin 4.	I/O	9	AE4	1
MMC2_DAT3	SD/MMC data pin 3.	I/O	11	AF4	1
MMC2_DAT2	SD/MMC data pin 2.	I/O	13	AG4	1
MMC2_DAT1	SD/MMC data pin 1.	I/O	15	AH4	1
MMC2_DATO	SD/MMC data pin 0.	I/O	17	AH5	1
MMC2_CMD	SD/MMC command signal.	I/O	19	AG5	1
MMC_CLKO	SD/MMC clock signal.	O	21	AE2	1
McBSP Port 1					
McBSP1_DR	Multi channel buffered serial port receive	I	18		
McBSP1_CLKS	-----	N/A	N/A		
McBSP1_FSR	Multi channel buffered serial port transmit frame sync RCV	I/O	22		
McBSP1_DX	Multi channel buffered serial port transmit	I/O	12		
McBSP1_CLKX	Multi channel buffered serial port transmit clock	I/O	14		
McBSP1_FSX	Multi channel buffered serial port transmit frame sync XMT	I/O	16		
McBSP1_CLKR	Multi channel buffered serial port receive clock	I/O	20		
I2C Port 2					
I2C2_SDA	I2C data line.	IOD	23		
I2C2_SCL	I2C clock line	IOD	24		
McBSP Port 3					
McBSP3_DR	Multi channel buffered serial port receive	I	10,18		
McBSP3_DX	Multi channel buffered serial port transmit	I/O	4,12		
McBSP3_CLKX	Multi channel buffered serial port receive clock	I/O	6,14		
McBSP3_FSX	Multi channel buffered serial port frame sync transmit	I/O	8,16		
General Purpose I/O Pins					
GPIO_130	GP Input/Output pin. Can be used as an interrupt pin.	I/O	21		
GPIO_131	GP Input/Output pin. Can be used as an interrupt pin.	I/O	19		
GPIO_132	GP Input/Output pin. Can be used as an interrupt pin.	I/O	17		
GPIO_133	GP Input/Output pin. Can be used as an interrupt pin.	I/O	15		
GPIO_134	GP Input/Output pin. Can be used as an interrupt pin.	I/O	13		
GPIO_135	GP Input/Output pin. Can be used as an interrupt pin.	I/O	11		
GPIO_136	GP Input/Output pin. Can be used as an interrupt pin.	I/O	9		
GPIO_137	GP Input/Output pin. Can be used as an interrupt pin.	I/O	7		
GPIO_138	GP Input/Output pin. Can be used as an interrupt pin.	I/O	5		
GPIO_139	GP Input/Output pin. Can be used as an interrupt pin.	I/O	3		
GPIO_143	GP Input/Output pin. Can be used as an interrupt pin.	I/O	8		
GPIO_144	GP Input/Output pin. Can be used as an interrupt pin.	I/O	4		
GPIO_145	GP Input/Output pin. Can be used as an interrupt pin.	I/O	10		
GPIO_146	GP Input/Output pin. Can be used as an interrupt pin.	I/O	6		
GPIO_156	GP Input/Output pin. Can be used as an interrupt pin.	I/O	20		
GPIO_158	GP Input/Output pin. Can be used as an interrupt pin.	I/O	12		
GPIO_159	GP Input/Output pin. Can be used as an interrupt pin.	I/O	18		
GPIO_161	GP Input/Output pin. Can be used as an interrupt pin.	I/O	16		
GPIO_162	GP Input/Output pin. Can be used as an interrupt pin.	I/O	14		
GPIO_168	GP Input/Output pin. Can be used as an interrupt pin.	I/O	24		
GPIO_183	GP Input/Output pin. Can be used as an interrupt pin.	I/O	23		
McSPI Port 3					
McSPI3_CS0	Multi channel SPI chip select 0	O	11		
McSPI3_CS1	Multi channel SPI chip select 1	O	13		
McSPI3_SIMO	Multi channel SPI slave in master out	I/O	19		
McSPI3_SOMI	Multi channel SPI slave out master in	I/O	17		
McSPI3_CLK	Multi channel SPI clock	I/O	21		
McSPI Port 4					
McSPI4_SIMO	Multi channel SPI slave in master out	I/O	12		
McSPI4_SOMI	Multi channel SPI slave out master in	I)	18		
McSPI4_CS0	Multi channel SPI chip select 0	O	16		
McSPI4_CLK	Multi channel SPI clock	I/O	20		
UART Port 2					
UART2_CTS	UART clear to send.	I/O	4		
UART2 RTS	UART request to send	O	10		

UART2_RX	UART receive	I	8		
UART2_TX	UART transmit	O	6		
GPT PWM					
GPT9_PWMEV	PWM or event for GP timer 9	O	4		
GPT11_PWMEV	PWM or event for GP timer 11	O	10		
GPT10_PWMEV	PWM or event for GP timer 10	O	8		

8.19.3 Power

The expansion connector provides two power rails. The first is the **VIO_1.8V** rail which is supplied by the **TPS65950**. This rail is limited in the current it can supply from the **TPS65950** and what remains from the current consumed by the BeagleBoard and is intended to be used to provide a rail for voltage level conversion only. It is not intended to power a lot of circuitry on the expansion board. All signals from the BeagleBoard are at 1.8V.

The other rail is the **DC_5V**. The same restriction exists on this rail as mentioned in the USB section. The amount of available power to an expansion board depends on the available power from the DC supply or the USB supply from the PC.

8.19.4 Reset

The **nRESET** signal is the main board reset signal. When the board powers up, this signal will act as an input to reset circuitry on the expansion board. After power up, a system reset can be generated by the expansion board by taking this signal low. This signal is a 1.8V level signal.

8.19.5 Power Control

There is an additional open-drain signal on the connector called **REGEN**. The purpose of this signal is to provide a means to control power circuitry on the expansion card to turn on and off the voltages. This insures that the power on the expansion board is turned on at the appropriate time. Depending on what circuitry is provided on the expansion board, an additional delay may be needed to be added before the circuitry is activated. Refer to the **OMAP3530** and **TPS65950** documentation for more information.

8.20 Additional Expansion Header

If you choose not to use the LCD headers for access to the LCD signals or for the DVI-D interface, they can also be used for other functions on the board based on the pin mux setting of each pin. **Table 22** shows the options for **J4** and **Table 23** shows the options for **J5**. The MUX: column indicates which MUX mode must be set for each pin to make the respective signals accessible on the pins of the **OMAP3530**.

Table 22. J4 GPIO Signals

Pin#	Signal	MUX:0	MUX:2	MUX:4
3	DVI_DATA1	DATA1	UART1_RTS	GPIO71
4	DVI_DATA0	DATA0	UART1_CTS	GPIO70
5	DVI_DATA3	DATA3	-	GPIO73
6	DVI_DATA2	DATA2	-	GPIO72
7	DVI_DATA5	DATA5	UART3_TX	GPIO75
8	DVI_DATA4	DATA4	UART3_RX	GPIO74
9	DVI_DATA12	DATA12		GPIO82
10	DVI_DATA10	DATA10	-	GPIO79
11	DVI_DATA23	DATA23	-	GPIO93
12	DVI_DATA14	DATA14	-	GPIO84
13	DVI_DATA19	DATA19	McSPI3_SIMO	GPIO89
14	DVI_DATA22	DATA22	McSPI3_CS1	GPIO92
15	I2C3_SDA	I2C3_SDA	-	-
16	DVI_DATA11	DATA11	-	GPIO81
17	DVI_VSYNC	VSYNC	-	GPIO68
18	DVI_PUP	DVI_PUP	-	-

Table 23. J5 GPIO Signals

Pin#	Signal	MUX:0	MUX:2	MUX:4
3	DVI_DATA20	DATA20	McSPI3_SOMI	GPIO90
4	DVI_DATA21	DATA21	McSPI3_CS0	GPIO91
5	DVI_DATA17	DATA17	-	GPIO87
6	DVI_DATA18	DATA18	McSPI3_CLK	GPIO88
7	DVI_DATA15	DATA15	-	GPIO85
8	DVI_DATA16	DATA16	-	GPIO86
9	DVI_DATA7	DATA7	UART1_RX	GPIO77
10	DVI_DATA13	DATA13	-	GPIO83
11	DVI_DATA8	DATA8	-	GPIO78
12	NC	-	-	-
13	DVI_DATA9	DATA9	-	GPIO79
14	I2C3_SCL	I2C3_SCL		-
15	DVI_DATA6	DATA6	UART1_TX	GPIO_76
16	DVI_CLK+	PCLK	-	GPIO66
17	DVI_DEN	DEN	-	GPIO69
18	DVI_HSYNC	HSYNC	-	GPIO67

9.0 Connector Pinouts and Cables

This section provides a definition of the pinouts and cables to be used with all of the connectors and headers on the BeagleBoard.

THERE ARE NO CABLES SUPPLIED WITH THE BEAGLEBOARD.

9.1 Power Connector

Figure 47 is a picture of the BeagleBoard power connector with the pins identified. The supply must have a 2.1mm center hot connector with a 5.5mm outside diameter.

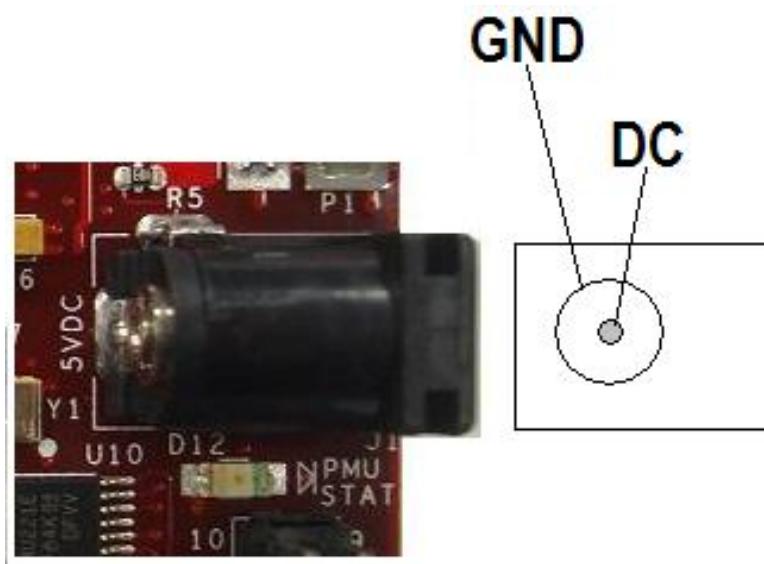


Figure 47. Power Connector

The supply must be at least 500mA with a maximum of 2A. If the expansion connector is used, more power will be required depending on the load of the devices connected to the expansion connector.

**WARNING: DO NOT PLUG IN ANYTHING BUT
5V TO THE DC CONNECTOR OR THE BOARD
WILL BE DAMAGED!!!!**

9.2 USB OTG

Figure 48 is a picture of the BeagleBoard USB OTG connector with the pins identified.



Figure 48. USB OTG Connector

The shorting pads to convert the OTG port to a Host mode are found in Figure 49.

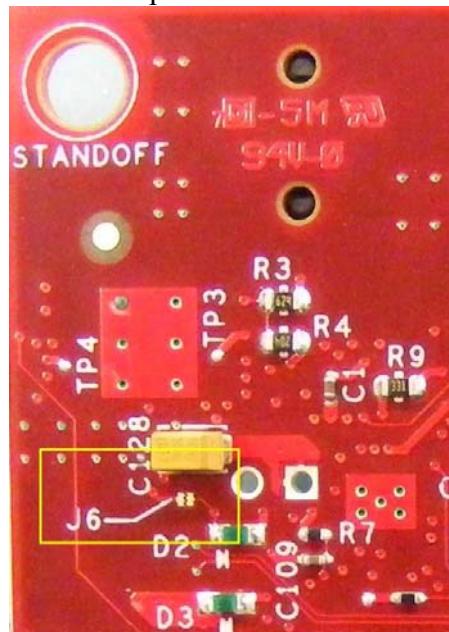


Figure 49. OTG Host Shorting Pads

9.3 S-Video

Figure 50 is the S-Video connector on the BeagleBoard.

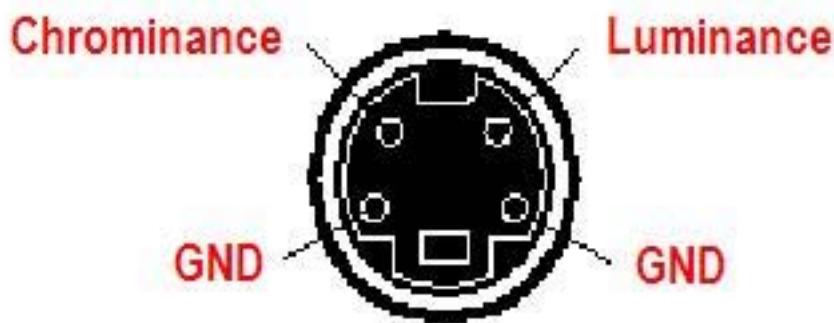
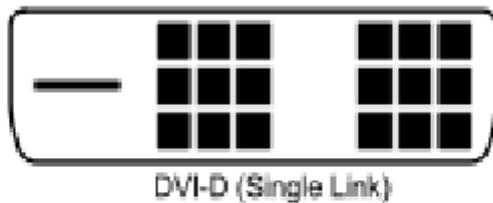


Figure 50. S-Video Connector

9.4 DVI-D

Figure 51 is the pinout of the DVI-D connector on BeagleBoard.

Cable End



BeagleBoard

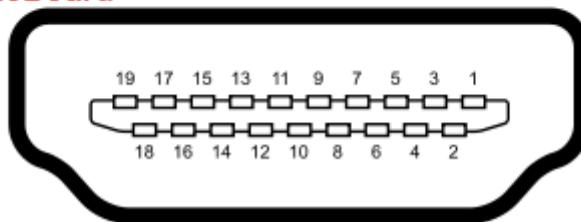


Figure 51. DVI-D Connector

Table 24 is the pin numbering of the two ends of the cable as it relates to the signals used in the DVI-D interface itself.

Table 24. DVI-D to HDMI Cable

SIGNAL	DVI-D PIN#	HDMI PIN#
DATA 2-	1	3
DATA 2+	2	1
SHIELD	3	2
	4	
	5	
DDS CLOCK	6	15
DDS DATA	7	16
	8	
DATA 1-	9	6
DATA 1+	10	4
SHIELD	11	5
	12	
	13	
5V	14	18
GROUND (5V)	15	17
	16	
DATA 0-	17	9
SIGNAL	DVI-D PIN#	DVI-D PIN#
DATA 0+	18	7

SHIELD	19	5
	20	
	21	
	22	
CLOCK+	23	10
CLOCK-	24	12

DO NOT PLUG IN THE DVI-D CONNECTOR TO A DISPLAY WITH THE BEAGLEBOARD POWERED ON. PLUG IN THE CABLE TO THE DISPLAY AND THEN POWER ON THE BEAGLEBOARD.

Figure 52 is the cable to be used to connect to an LCD monitor.



Figure 52. DVI-D Cable

9.5 LCD

This section covers the pair of headers that provide access to the raw 1.8V DSS signals from the OMAP3530 processor. This provides the ability to create adapters for such things as different LCD panels, LVDS interfaces, etc.

9.5.1 Connector Pinout

The **Table 25** and **26** define the pinout of the LCD connectors. All signal levels are 1.8V with the exception of DVI_PUP signal which is 3.3V.

Table 25. J4 LCD Signals

Pin#	Signal	I/O	Description
1	DC_5V	PWR	DC rail from the Main DC supply
2	DC_5V	PWR	DC rail from the Main DC supply
3	DVI_DATA1	O	LCD Pixel Data bit
4	DVI_DATA0	O	LCD Pixel Data bit
5	DVI_DATA3	O	LCD Pixel Data bit
6	DVI_DATA2	O	LCD Pixel Data bit
7	DVI_DATA5	O	LCD Pixel Data bit
8	DVI_DATA4	O	LCD Pixel Data bit
9	DVI_DATA12	O	LCD Pixel Data bit
10	DVI_DATA10	O	LCD Pixel Data bit
11	DVI_DATA23	O	LCD Pixel Data bit
12	DVI_DATA14	O	LCD Pixel Data bit
13	DVI_DATA19	O	LCD Pixel Data bit
14	DVI_DATA22	O	LCD Pixel Data bit
15	I2C3_SDA	I/O	I2C3 Data Line
16	DVI_DATA11	O	LCD Pixel Data bit
17	DVI_VSYNC	O	LCD Vertical Sync Signal
18	DVI_PUP	O	Control signal for the DVI controller. When Hi, DVI is enabled. Can be used to activate circuitry on adapter board if desired.
19	GND	PWR	Ground bus
20	GND	PWR	Ground bus

Table 26. J5 LCD Signals

Pin#	Signal	I/O	Description
1	3.3V	PWR	3.3V reference rail
2	VIO_1V8	PWR	1.8V buffer reference rail.
3	DVI_DATA20	O	LCD Pixel Data bit
4	DVI_DATA21	O	LCD Pixel Data bit
5	DVI_DATA17	O	LCD Pixel Data bit
6	DVI_DATA18	O	LCD Pixel Data bit
7	DVI_DATA15	O	LCD Pixel Data bit
8	DVI_DATA16	O	LCD Pixel Data bit
9	DVI_DATA7	O	LCD Pixel Data bit
10	DVI_DATA13	O	LCD Pixel Data bit
11	DVI_DATA8	O	LCD Pixel Data bit
12	NC		No connect
13	DVI_DATA9		LCD Pixel Data bit
14	I2C3_SCL	I/O	I2C3 Clock Line
15	DVI_DATA6	O	LCD Pixel Data bit
16	DVI_CLK+	O	DVI Clock
17	DVI_DEN	O	Data Enable
18	DVI_HSYNC	O	Horizontal Sync
19	GND	PWR	Ground bus
20	GND	PWR	Ground bus

9.5.2 Connector Suppliers

The actual connector to be used will be determined by the supplier of the board to be plugged into the Beagle. **Table 27** below lists a few of the part numbers and suppliers that can be used for the connectors on the LCD interface. All of the listed connectors are in a vertical mount configuration

Table 27. J4 and J5 Connector Sources

Supplier	Header	Socket (Thru Hole)	Socket (SMT)
Major League	TSHC-510-D-06-340-G-LF	SSHS-510-D-04-G-LF	LSSHS-510-D-06-G-LF
"	TSHC-510-D-06-340-T-LF	SSHS-510-D-04-T-LF	LSSHS-510-D-06-T-LF
"	TSHC-510-D-06-340-GT-LF	SSHS-510-D-04-TG-LF	LSSHS-510-D-06-F-LF
"	TSHC-510-D-06-340-H-LF	SSHS-510-D-04-H-LF	
"	TSHC-510-D-06-340-F-LF	SSHS-510-D-04-F-LF	
SAMTEC	FTS-110-01-L-D		FLE-110-01-G-DV
"	FTS-110-03-L-D		
Sullins	GRPB052VWVN-RC		

Major League <http://www.mlelectronics.com/>

Samtec <http://www.samtec.com>

Sullins <http://www.sullinscorp.com>

9.5.3 Dimensions

Figure 53 provides some of the dimensions that can assist in the location of the LCD headers. It is strongly recommended that the CAD data be used in order to determine their location exact. **Table 28** provides the values for each lettered dimension.

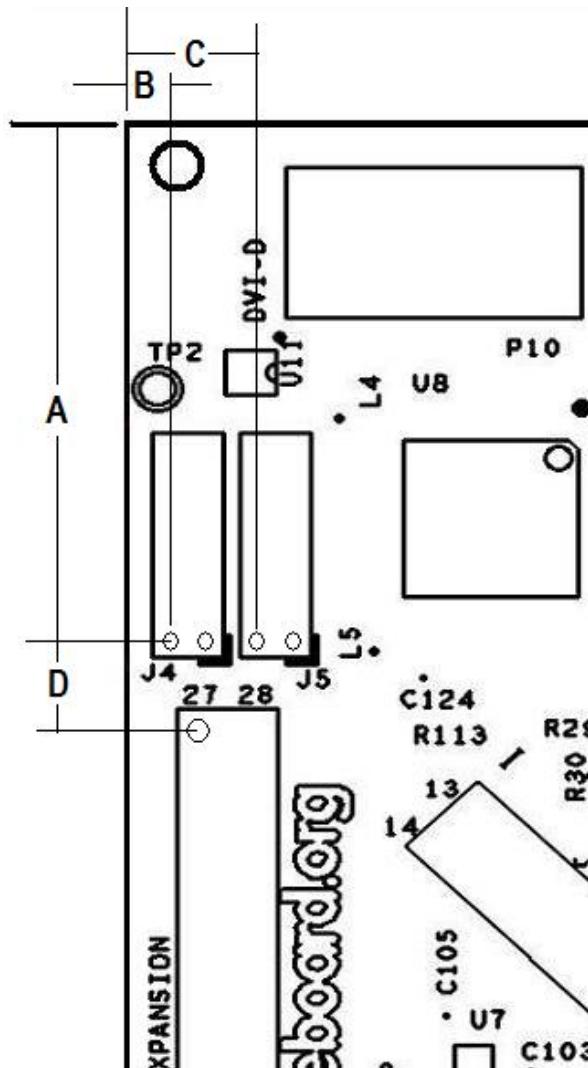


Figure 53. Top Mount LCD Adapter

Table 28. Connector Dimensions

Dimension	Inches	Millimeters
A	1.085	27.56
B	0.118	2.99
C	0.296	7.52
D	0.190	4.83

9.5.4 Mounting Scenarios

This section provides a few possible mounting scenarios for the LCD connectors. It should be noted that the voltage level of these signals are 1.8V. It will require that they be buffered in order to drive other voltage levels.

9.5.4.1 Top Mounting

Figure 54 shows the board being mounted on top of the BeagleBoard.

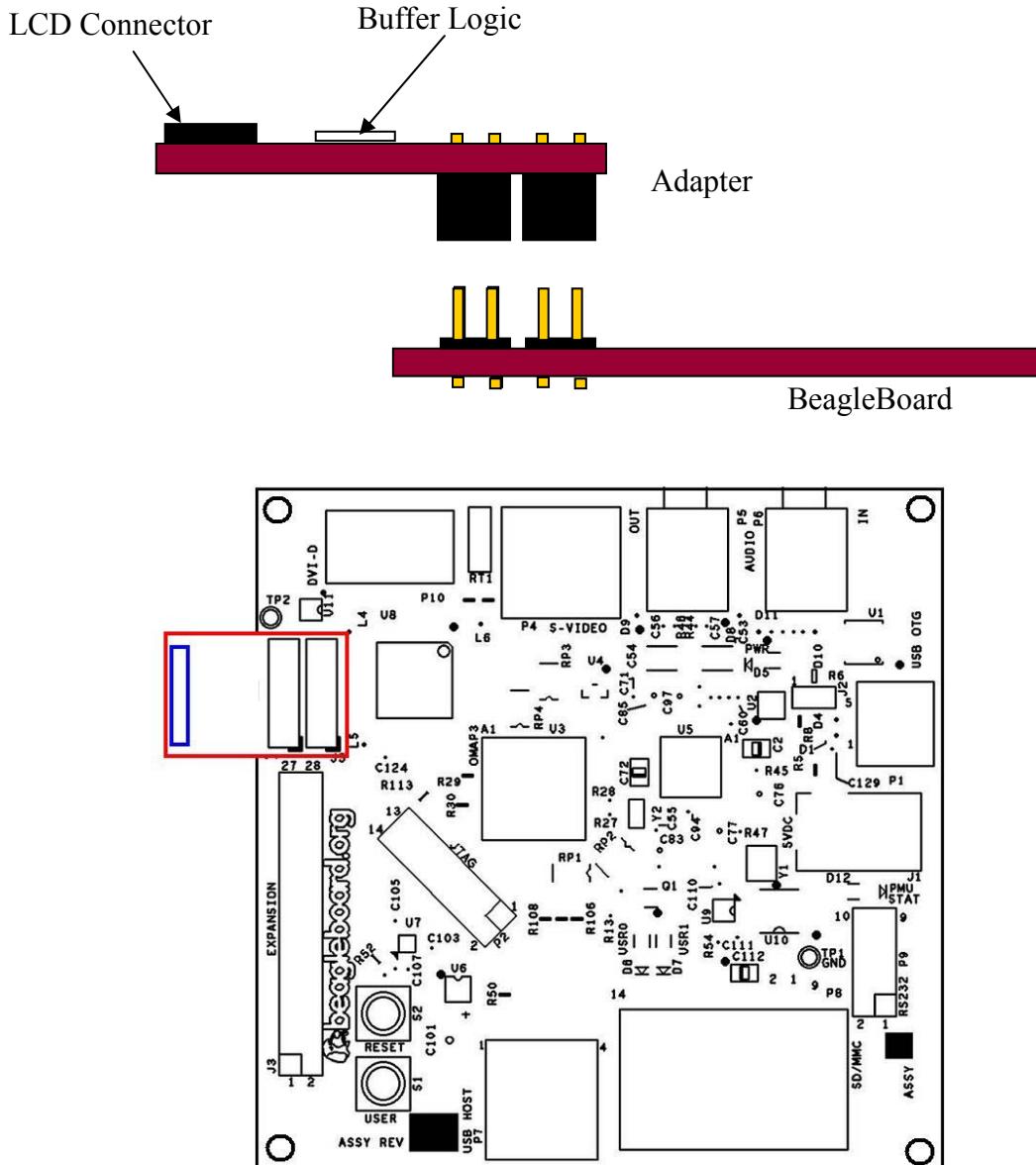


Figure 54. Top Mount LCD Adapter

9.5.4.2 *Bottom Mounting*

Figure 55 shows the board being mounted under the BeagleBoard.

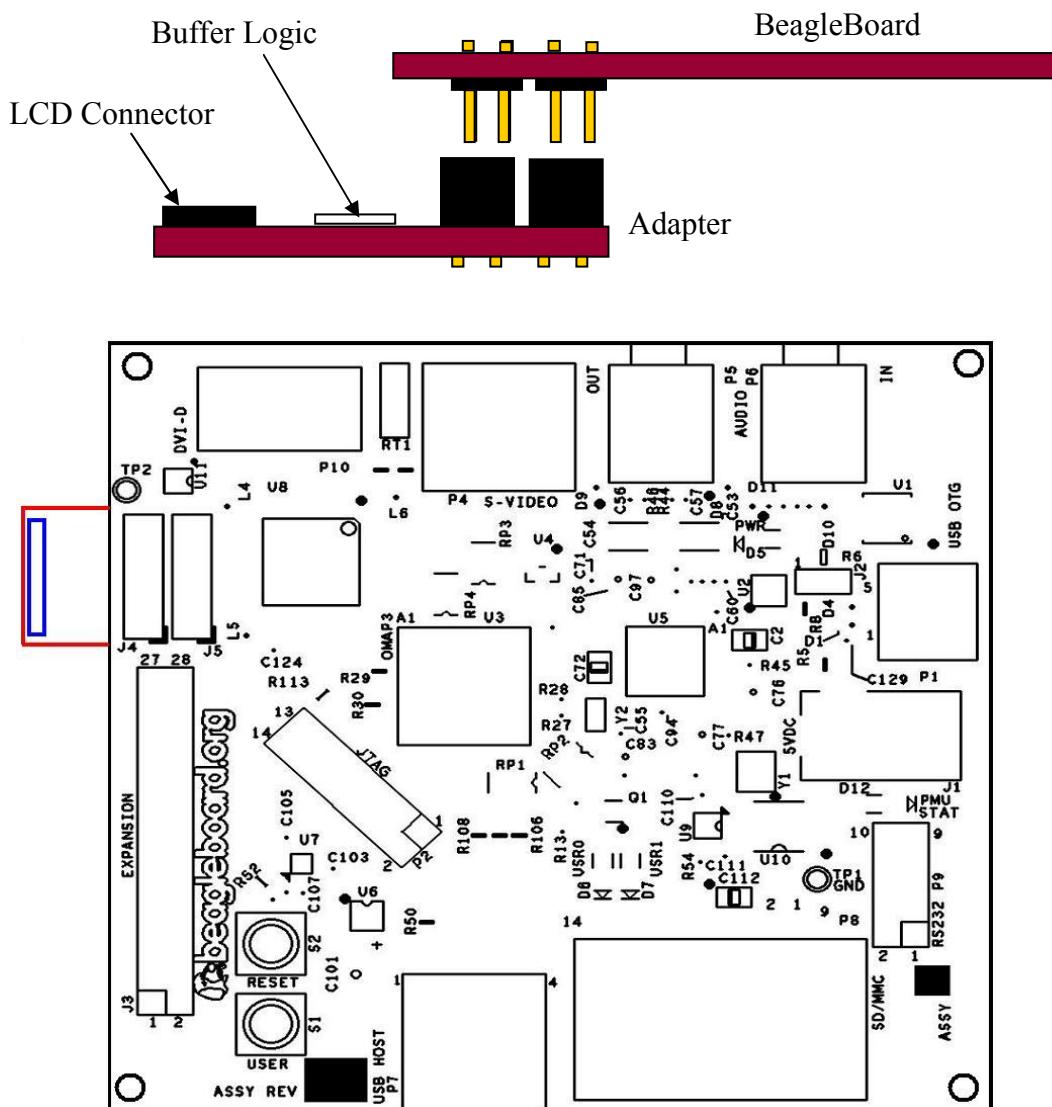


Figure 55. Bottom Mount LCD Adapter

9.6 Audio Connections

Figure 56 is the audio input jack required to connect to the BeagleBoard.

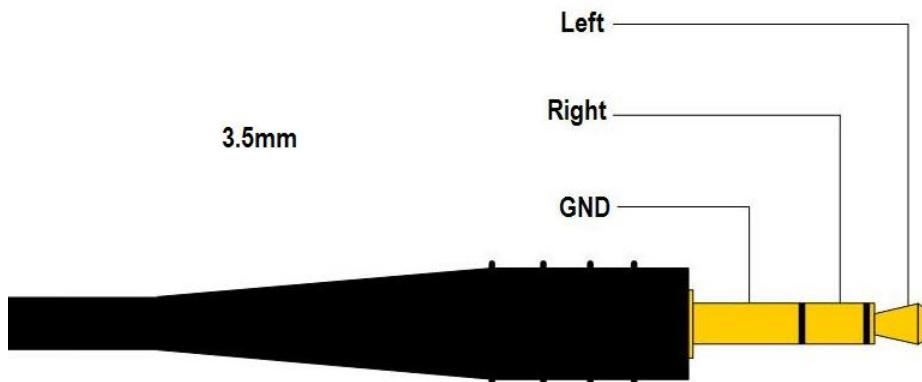


Figure 56. Audio In Plug

Figure 57 is the actual connector used on the BeagleBoard.



Figure 57. Audio In Plug

9.7 Audio Out

Figure 58 is the audio out jack required to connect to the BeagleBoard.

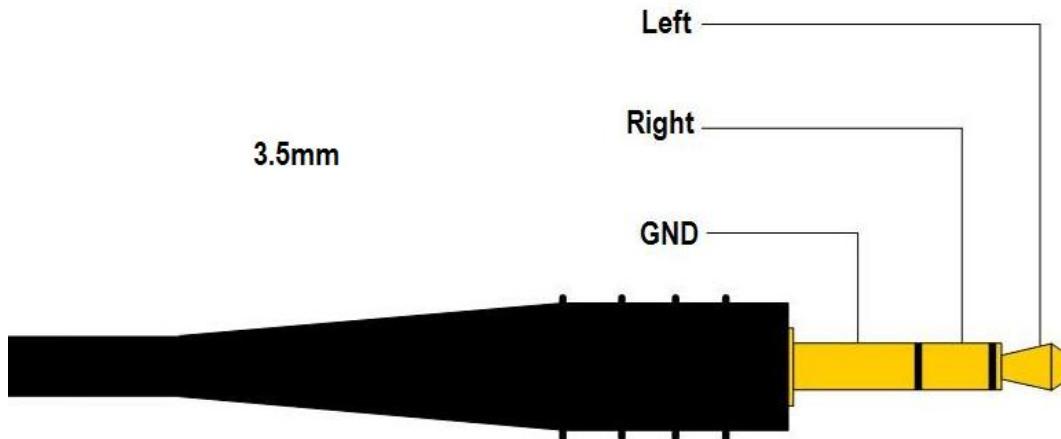


Figure 58. Audio Out Plug

Figure 59 is the actual connector used on the BeagleBoard.



Figure 59. Audio In Plug

9.8 JTAG

Figure 60 is the JTAG connector pin out showing the pin numbering.

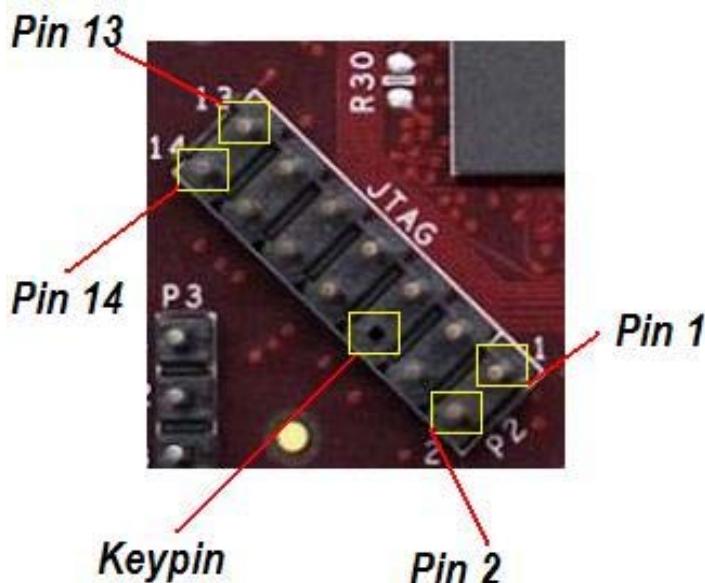


Figure 60. JTAG Connector Pinout

Table 29 gives a definition of each of the signals on the JTAG header.

Table 29. JTAG Signals

Pin	Signal	Description	I/O
1	JTAG_TMS	Test mode select	I/O
3	JTAG_TDI	Test data input	I
7	JTAG_TDO	Test Data Output	O
9	JTAG_RTCK	ARM Clock Emulation	O
11	JTAG_TCK	Test Clock	I
2	JTAG_nTRST	Test reset	I
13	JTAG_EMU0	Test emulation 0	I/O
14	JTAG_EMU1	Test emulation 1	I/O
5	VIO	Voltage pin	PWR
4,8,10,12,14	GND	Ground	PWR

All of the signals are 1.8V only. The JTAG emulator must support 1.8V signals for use on the BeagleBoard.

If a 20 pin connector is provided on the JTAG emulator, then a 20 pin to 14 pin adapter must be used. You may also use emulators that are either equipped with a 14 pin connector or are universal in nature.

Figure 61 shows an example of a 14 pin to 20 pin adapter.

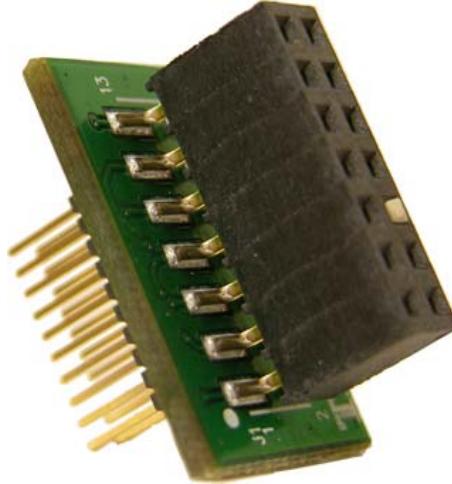


Figure 61. JTAG 14 to 20 Pin Adapter

Figure 62 shows how the JTAG cable is to be routed when connected to the BeagleBoard.

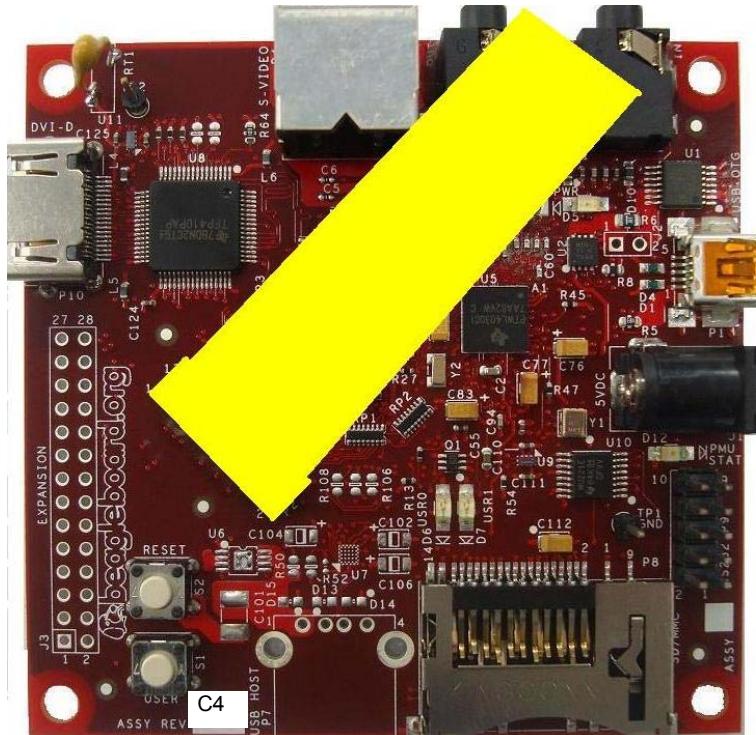


Figure 62. JTAG Connector Pinout

9.9 RS232

Figure 63 is the RS232 header on the BeagleBoard with the pin numbers identified.

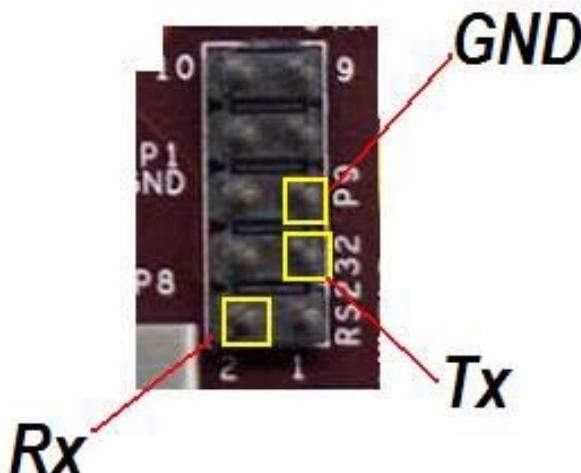


Figure 63. RS232 Header

Figure 64 is the cable that is required in order to access the RS232 header. This cable can be purchased from various sources and is referred to as the ATI/Everex type cable.

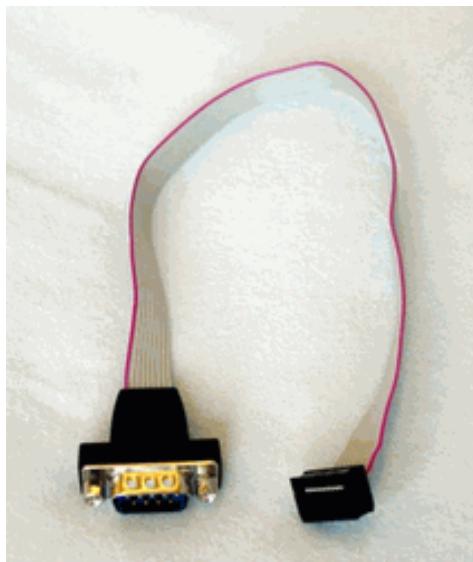


Figure 64. RS232 Flat Cable

9.10 Expansion

We are changing the configuration of the expansion header on all Beagle boards. This will standardize on the configuration for compatibility with expansion cards. The socket will be mounted on the BACKSIDE of the board.

Figure 65 is a drawing and picture of the socket that is specified.

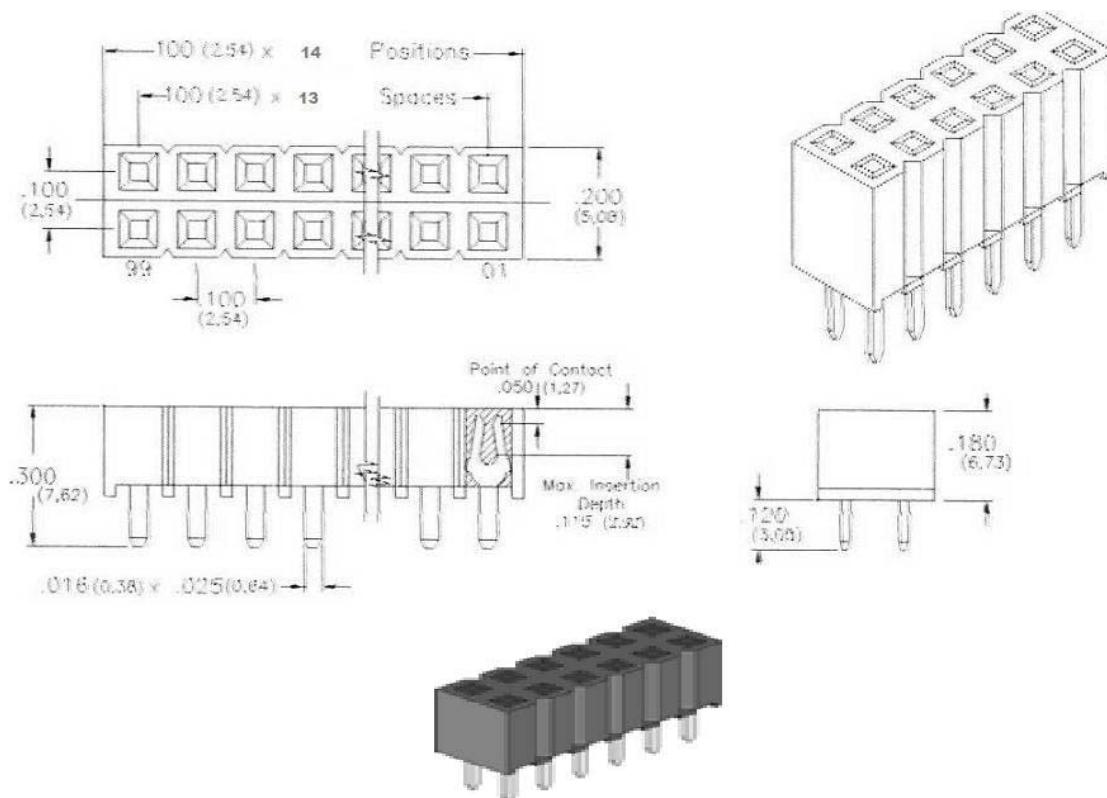


Figure 65. Expansion Sockets

The selected supplier for this socket is Major League Electronics <http://www.mlelectronics.com/> and the part number is **LSWSS-114-D-02-T-LF**. Other suppliers may be available as well and can be used as long as it is compatible with this socket. This socket is mounted on the BACK of the BeagleBoard.

9.11 Battery Installation

9.11.1 Battery

The board was designed to use the VL-1220/VCN battery from Panasonic-BSG. This is a Vanadium Pentoxide Lithium Rechargeable Battery with a 7mAH capacity. **Figure 66** is a picture of the battery.

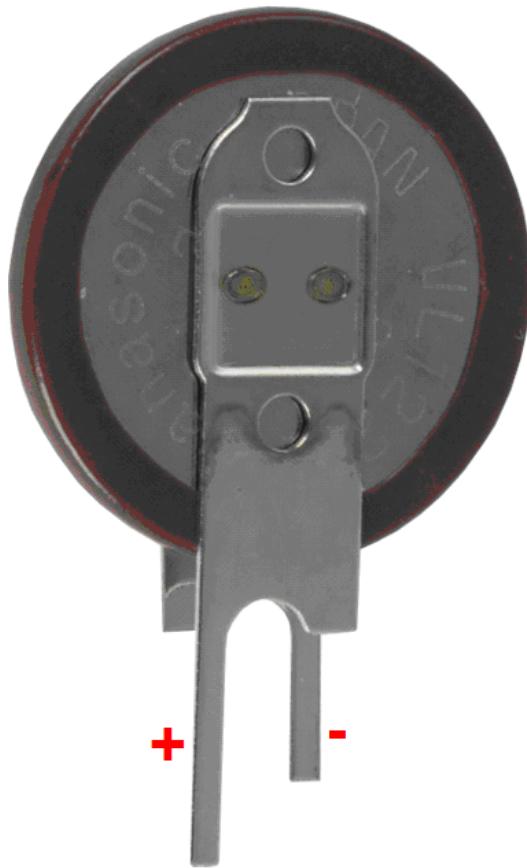


Figure 66. Optional Battery

9.11.2 Battery Installation

THE FOLLOWING INSTRUCTIONS ASSUME THE USER HAS PREVIOUS EXPERIENCE WITH BATTERIES. BATTERY INSTALLATION IS THE SOLE RESPONSIBILITY OF THE USER. INSTALLATION OF THE BATTERY BY THE USER IS AT THEIR OWN RISK. FAILURE TO FOLLOW THE INSTRUCTIONS CAN RESULT IN DAMAGE TO THE BOARD. THIS DAMAGE IS NOT COVERED UNDER THE WARRANTY.

Figure 67 shows the location of the battery on the Beagle Board.

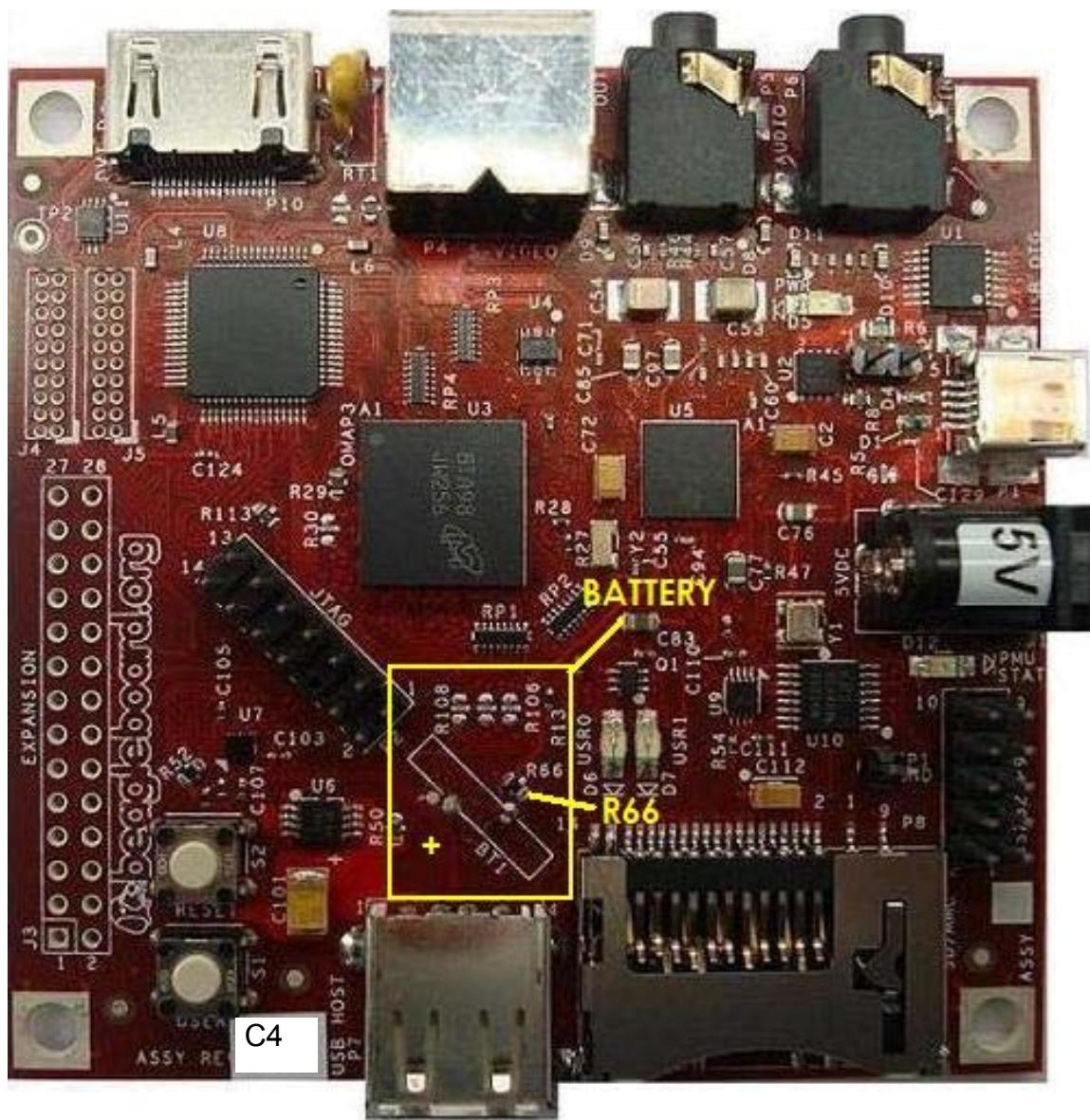


Figure 67. Optional Battery Location

Following are the steps required to install the battery.

- 1) Remove all cables from the board.
- 2) Remove R66 from the board as shown on **Figure 67**.
- 3) Using Figure 66, locate the positive (+) lead of the battery.
- 4) Insert the (+) lead into the hole that is marked (+) on **Figure 67**.

10.0 BeagleBoard Accessories

Throughout this manual various items are mentioned as not being provided with the standard BeagleBoard package or as options to extend the features of the BeagleBoard. The concept behind BeagleBoard is that different features and functions can be added to BeagleBoard by bringing your own peripherals. This has several key advantages:

- User can choose which peripherals to add.
- User can choose the brand of peripherals based on driver availability and ability to acquire the particular peripheral
- User can add these peripherals at a lower cost than if they were integrated into the BeagleBoard.

This section covers these accessories and add-ons and provides information on where they may be obtained. Obviously things can change very quickly as it relates to devices that may be available. Please check BeagleBoard.org for an up to date listing of these peripherals.

Inclusion of any products in this section does not guarantee that they will operate with all SW releases. It is up to the user to find the appropriate drivers for each of these products. Information provided here is intended to expose the capabilities of what can be done with the BeagleBoard and how it can be expanded.

All pricing information provided is subject to change and in most cases is likely to be lower depending on the products purchased and from where they are purchased.

Covered in this section are the following accessories:

- DC Power Supplies
- Serial Ribbon cable
- USB Hubs
- USB Thumb Drives
- DVI-D Cables
- DVI-D Monitors
- SD/MMC Cards
- USB to Ethernet
- USB to WiFi
- USB Bluetooth
- Expansion Cards

NO CABLES OR POWER SUPPLIES ARE PROVIDED WITH THE BEAGLEBOARD.

10.1 DC Power Supply

Tabletop or wall plug supplies can be used to power BeagleBoard. **Table 30** provides the specifications for the BeagleBoard DC supply. Supplies that provide additional current than what is specified can be used if additional current is needed for add on accessories. The amount specified is equal to that supplied by a USB port.

Table 30. DC Power Supply Specifications

Specification	Requirement	Unit
Voltage	5.0	V
Current	500mA (minimum)	mA
Connector	2.1mm x 5.5mm Center hot	

It is recommended that a supply higher than 500mA be used if higher current peripherals are expected to be used or if expansion boards are added.

Table 31 lists some power supplies that will work with the BeagleBoard.

Table 31. DC Power Supplies

Part #	Manufacturer	Supplier	Price
EPS050100-P6P	CUI	Digi-Key	\$7
DPS050200UPS-P5P-SZ	CUI	Digi-Key	\$16

Figure 68 is a picture of the type of power supply that will be used on the BeagleBoard.



Figure 68. DC Power Supply

10.2 Serial Ribbon Cable

Figure 69 is an example of the serial ribbon cable for the BeagleBoard. Other serial cables that will work on the board may have a different appearance.

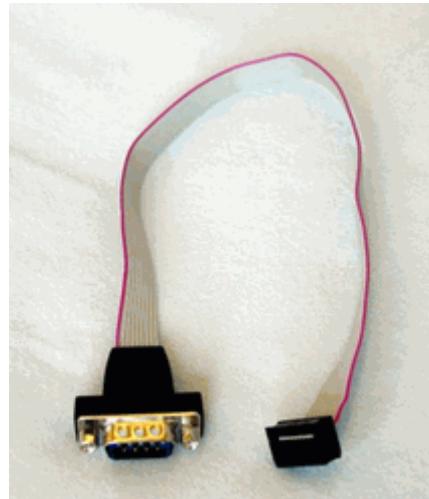


Figure 69. RS232 Cable

If you like, you can al71 67.

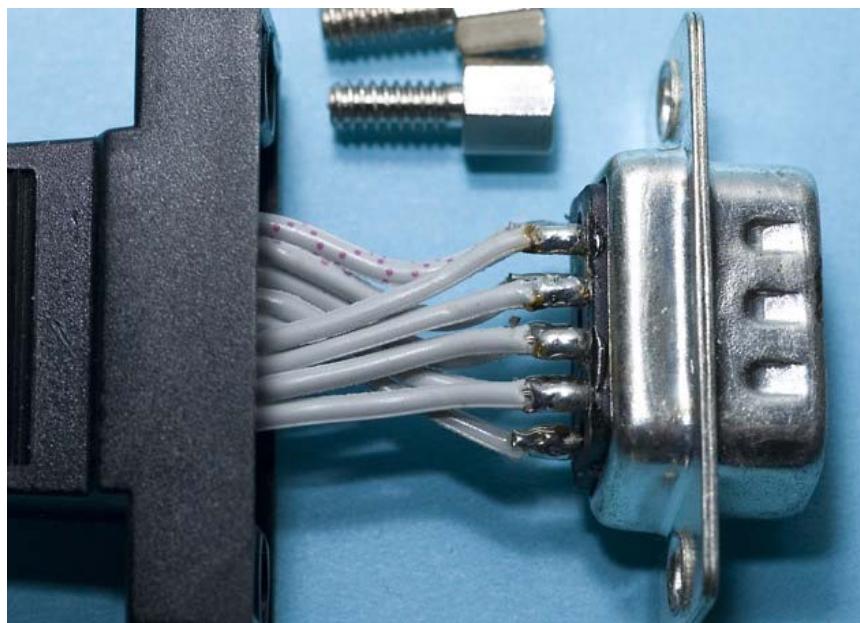


Figure 70. RS232 Cable Wiring

Table 32 shows the pinout of the ribbon cable connector.

Table 32. Cable Pinout

Ribbon Cable	DB9
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

10.3 USB Hubs

There are no known or anticipated issues with USB hubs. However, it should be noted that a self powered hub is highly recommended. **Table 34** is a list of Hubs that have been tested on the BeagleBoard.

Table 33. USB Hubs Tested

Supplier	Part Number
IOGEAR	GUH274
D-Link	DUB-H4 2.0
Vakoss	TC-204-NS

10.4 DVI Cables

In order to connect the DVI-D interface to a LCD monitor, a HDMI to DVI-D cable is required. **Figure 71** is a picture of a HDMI to DVI-D cable.



Figure 71. HDMI to DVI-D Cable

10.5 DVI-D Monitors

There are many monitors that can be used with the BeagleBoard. With the integrated EDID feature, timing data is collected from the monitor to enable the SW to adjust its timings. **Table 35** shows a short list of the monitors that have been tested to date on the BeagleBoard at the 1024x768 resolution. Please check on BeagleBoard.org for an up to date listing of the DVI-D monitors as well as information on the availability of drivers.

Table 34. DVI-D Monitors Tested

Manufacturer	Part Number	Status
Dell	2407WFPb	Tested
Insignia	NS-LCD15	Tested
Dell	1708FP	Tested

DO NOT PLUG IN THE DVI-D CONNECTOR TO A DISPLAY WITH THE BEAGLEBOARD POWERED ON. PLUG IN THE CABLE TO THE DISPLAY AND THEN POWER ON THE BEAGLEBOARD.

The digital portion of the DVI-D interface is compatible with HDMI and is electrically the same. A standard HDMI cable may be used to connect to the HDMI input of monitors. Whether or not the Beagle will support those monitors is dependent on the timings that are used on the Beagle and those that are accepted by the monitor. This may require a change in the software running on the Beagle. The audio and encryption features of HDMI are not supported by the Beagle.

The analog portion of DVI which provides RGB analog type signals is not supported by the Beagle. Buying a DVI to VGA adapter connector will not work on a VGA display. You will need an active DVI-D to VGA adapter.

10.6 SD/MMC Cards

Table 36 is a list of SD/MMC cards that have been tested on BeagleBoard. Please check BeagleBoard.org for an up to date listing of the SD/MMC cards that have been tested as well as information on the availability of drivers if required.

Table 35. SD/MMC Cards Tested

Manufacturer	Type	Part Number	Status
Patriot	SD	1GB	Tested
Microcenter	SD	1GB/2GB	Tested

10.7 USB to Ethernet

There are several USB to Ethernet adapters on the market and **Figure 72** shows a few of these devices. These devices can easily add Ethernet connectivity to BeagleBoard by using the USB OTG port in the host. This will require a special cable to convert the miniAB connector to a Type A or a hub can also be used. These are provided as examples only. Check BeagleBoard.org for information on devices that have drivers available for them.



Figure 72. USB to Ethernet Adapters

Table 37 provides examples of USB to Ethernet Adapters that **might** be used with the BeagleBoard. This list has not been verified. Inclusion of these items in the table does not guarantee that they will work, but is provided as examples only. Please check BeagleBoard.org for an up to date listing of the USB to Ethernet devices as well as information on the availability of drivers.

Table 36. USB to Ethernet Adapters

Product	Manufacturer	Status
ASOHOUSB	Airlink	Not Tested
TU-ET100C 10/100Mbps	TRENDnet	Not Tested
SABRENT	NB-USB20	Not Tested
Zonet	ZUN2210	Not Tested
StarTech	USB2105S	Not Tested

MOSCHIP is the silicon provider for USB to Ethernet devices. The product that has been tested uses the 7830 from MOSCHIP and has a vendor ID of 9710 and a product ID of 7830. The devices above that are based upon the MOSCHIP device are highlighted in red.

10.8 USB to WiFi

There are several USB to WiFi adapters on the market and **Figure 73** shows a few of these devices. These devices can easily add WiFi connectivity to BeagleBoard by using the USB OTG port in the host mode. This will require a special cable to convert the

miniAB connector to a Type A or a hub can also be used. These are provided as examples only. Check BeagleBoard.org for information on devices that have drivers available for them.



Figure 73. USB to WiFi

Table 38 provides a list of USB to WiFi adapters that could be used with the BeagleBoard. Inclusion of these items in the table does not guarantee that they will work, but is provided as examples only. Please check BeagleBoard.org for an up to date listing of the USB to WiFi devices as well as information on the availability of drivers.

Table 37. USB to WiFi Adapters

Product	Manufacturer	Status
4410-00-00AF	Zoom	Not Tested
HWUG1	Hawkins	Not Tested
TEW-429Uf	Trendnet	Not Tested

It should be noted that the availability of Linux drivers for various WiFi devices is limited. Before purchasing a particular device, please verify the availability of drivers for that device.

10.9 USB to Bluetooth

There are several USB to Bluetooth adapters on the market and **Figure 74** shows a few of these devices. These devices can easily add Bluetooth connectivity to BeagleBoard by using the USB OTG port in the host mode. This will require a special cable to convert the miniAB connector to a Type A or a hub can also be used. These are provided as examples only. Check BeagleBoard.org for information on devices that have drivers available for them and their test status.



Figure 74. USB to Bluetooth

Table 39 provides a list of USB to Bluetooth adapters that could be used with the BeagleBoard. Inclusion of these items in the table does not guarantee that they will work, but is provided as examples only. Please check BeagleBoard.org for an up to date listing of the USB to Bluetooth devices as well as information on the availability of drivers.

Table 38. USB to Bluetooth Adapters

Product	Manufacturer
TBW-105UB	Trendnet
ABT-200	Airlink
F8T012-1	Belkin

10.10 Expansion Card Design

This section covers the requirements for expansion cards that are designed to mount onto the Beagle Board..

11.0 Mechanical Information

11.1 BeagleBoard Dimensions

This section provides information on the mechanical aspect of the BeagleBoard. **Figure 75** is the dimensions of the BeagleBoard.

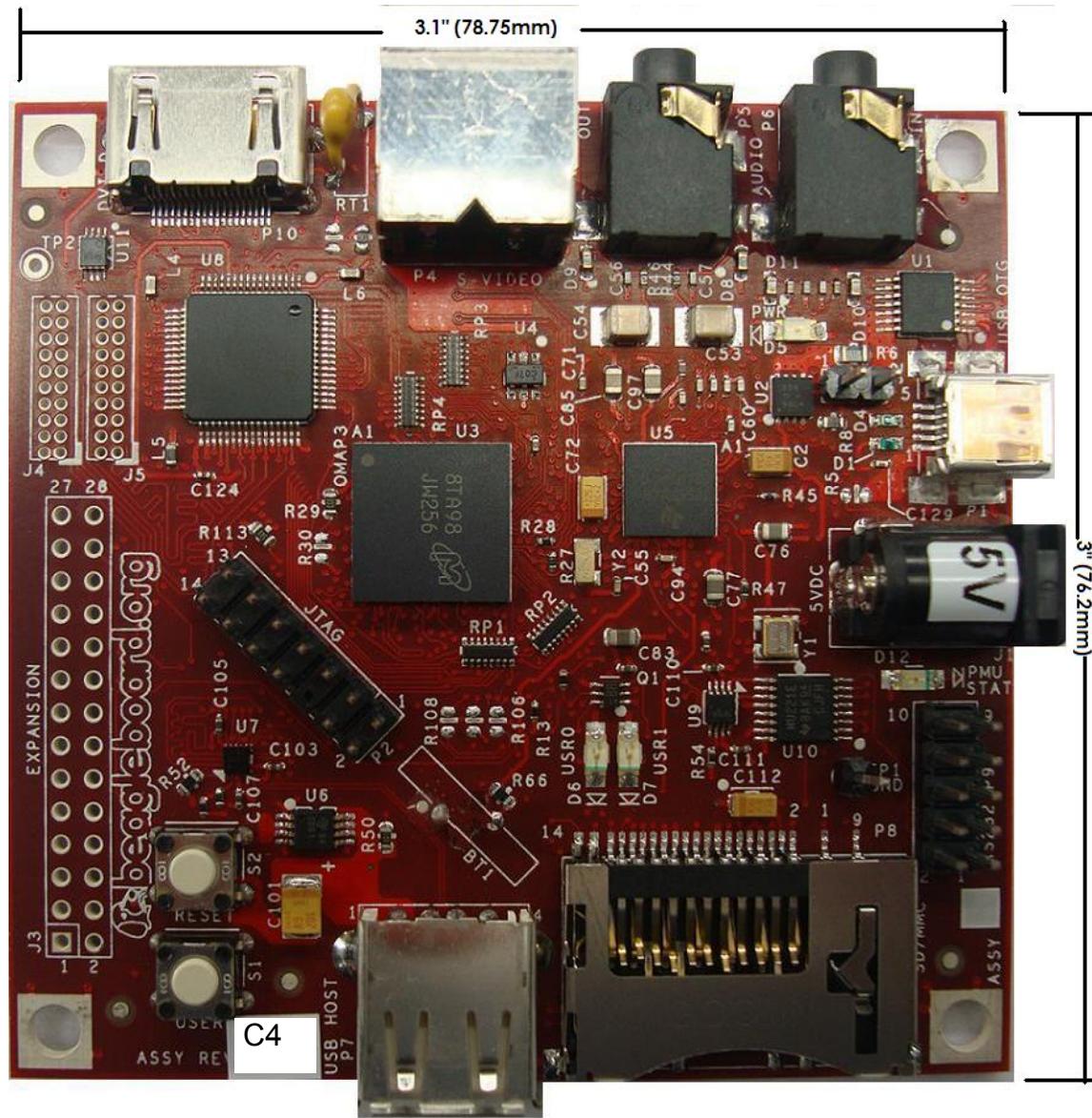


Figure 75. BeagleBoard Dimension Drawing

11.2 BeagleBoard Expansion Card Design Information

This section provides information on what is required from a mechanical and electrical aspect to create expansion cards for the BeagleBoard that are designed to connect to the Expansion header on the BeagleBoard. Users are free to create their own cards for private or commercial use, but in order to be supported by the Software they must conform to these standards if such support is desired.

11.2.1 Mounting Method

The standard method to provide a daughtercard for the BeagleBoard is for it to be mounted UNDER the Beagle Board as described in **Figure 76**.

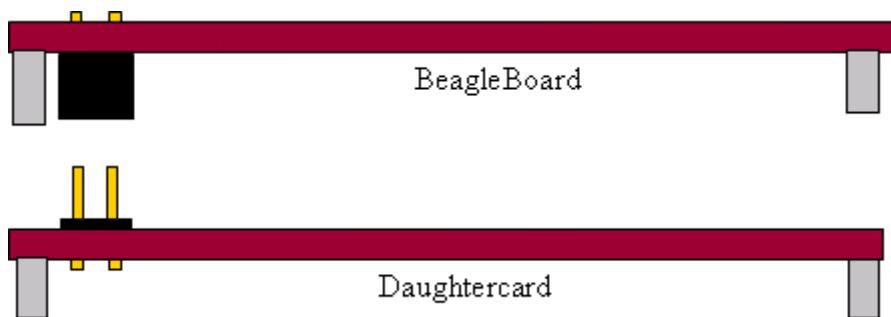


Figure 76. BeagleBoard Bottom Stacked Daughter Card

At the next letter revision of the board, all BeagleBoards produced will have the connectors pre mounted onto the bottom of the BeagleBoard.

11.2.2 Expansion EEPROM

All expansion cards designed for use with the BeagleBoard are required to have a EEPROM located on the board. This is to allow for the identification of the card by the Software in order to set the pin muxing on the expansion connector to be compatible with the expansion card.

The schematic for the EEPROM is in **Figure 77** below.

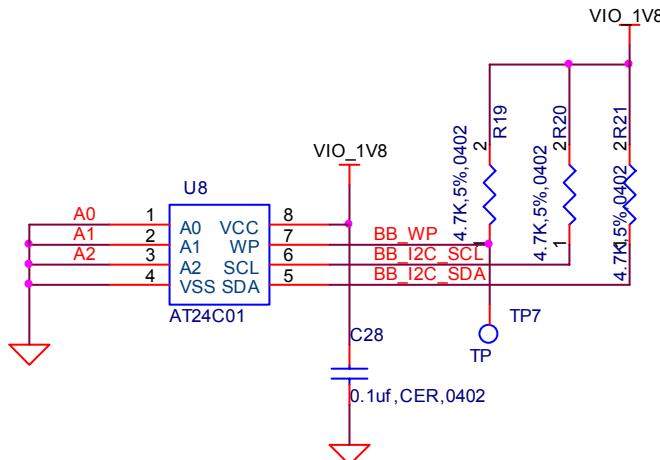


Figure 77. BeagleBoard Expansion Board EEPROM Schematic

The EEPROM must be write protected. It is suggested that a testpoint be used to allow for the WP to be disabled during test to allow the required data to be written to the EEPROM. The EEPROM is to be connected to I2C2 as found on the main expansion connector.

The EEPROM that is designated is the AT24C01 or ATC24C01B. The AT24C01 is designated as “Not Recommended for New Design” but can still be used. The AT24C01B is the replacement part and is available in several different packages, all of which can be used.

- TSSOP 8
- PDIP 8
- UDFN 8
- SOIC 8
- SOT23 5
- dBGA2 8

The contents of the EEPROM are not specified in this document.

12.0 Board Verification

This section provides a step by step process to be followed to verify that the hardware is working. This is the same basic process the board is taken through in production testing.

For an up to date listing of common questions and their answers, please refer to
<http://elinux.org/BeagleBoardFAQ>

12.1 Equipment

To run these tests you will need the following components:

- BeagleBoard
- 5V DC supply with a 2.1mm I.D. and 5.5mm O.D. connector
- SD Card
- PC
- USB miniA to A cable
- USB HUB
- DVI-D Monitor
- DVI-D to HDMI cable
- Speakers
- 3.5mm stereo cable with connectors on both ends
- DB9 Null-Modem Cable
- DB9 to IDC-10 cable ATI/Everex configuration

12.2 Out of the Box

Each BeagleBoard comes pre-loaded with the XLoader and UBoot in Flash. When powered up, it will do the following:

1. Plug in either a USB cable to the board and then to a PC or plug in a 5V power supply.
2. Power LED (D5) will turn on.
3. On the terminal window the following will be printed:

Texas Instruments X-Loader 1.4.2 (Feb 19 2009 - 12:01:24)
Loading u-boot.bin from nand

U-Boot 2009.01-dirty (Feb 19 2009 - 12:22:31)

I2C: ready
OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
OMAP3 Beagle board + LPDDR/NAND
DRAM: 256 MB
NAND: 256 MiB
*** Warning - bad CRC or NAND, using default environment

MUSB: using high speed
In: serial usbtt
Out: serial usbtt
Err: serial usbtt

The warning message is not an indication of an error condition and is normal. The UBoot is configured to look for a script file on the SD card for booting instructions. If not found, it will then look for the environment variables. As these are not loaded at the factory, you will see the warning message.

4. At this point the following LEDS will turn on:

- o USR0
- o USR1
- o PMU

5. Then the following will be sent to the terminal window and a countdown will commence. To stop the countdown, hit any key on the terminal

Board revision C
Serial #486000030000000004013f8a17019010
Hit any key to stop autoboot: 10

The revision of the board should be identified as a Rev C. The Serial# is NOT the board serial number, but a unique ID for the processor.

6. The BeagleBoard.org logo will be sent out of the DVI port and the color bars will appear on the S-Video Port.
7. If you do not stop the booting process by hitting a key, the following will be printed to the terminal.

No MMC card found

Booting from nand ...

NAND read: device 0 offset 0x280000, size 0x400000

4194304 bytes read: OK

Wrong Image Format for bootm command

ERROR: can't get kernel image!

OMAP3 beagleboard.org #

12.3 SD Card Configuration

In order to boot from the SD card, it must be formatted and the files loaded. The following steps explain that process.

1. Format the MMC/SD Card for FAT32 File System using the HP USB Disk Storage Format Tool 2.0.6: <http://selfdestruct.net/misc/usbboot/SP27213.exe>
2. Insert the Card writer/reader into the Windows machine.
3. Insert MMC/SD card into the card reader/writer
4. Open the HP USB Disk Storage Format Tool.
5. Select “FAT as File System”. Click on “Start”.
6. After formatting is done Click “OK”
7. Copy the following files on to MMC in the exact order listed. **COPY THE MLO FIRST!** Make sure you name the file as indicated in the **BOLD** type. These files can be found at
<http://code.google.com/p/beagleboard/wiki/BeagleboardRevCValidation>

[MLO](#) as **MLO**

[u-boot](#) as **u-boot.bin**

[u-boot for flash](#) as **u-boot-f.bin**

[ramdisk image](#) as **ramdisk.gz**

[Kernel \(ulimage\)](#) as **ulimage.bin**

[reset.scr](#) as **boot.scr**

[x-loader image](#) as **x-load.bin.ift**

[Regular script file](#) as **normal.scr**

12.4 Setup

This step sets up the board for the tests to follow.

1. Make sure Beagle power is in OFF state by removing the 5V supply and the USB host connection.
2. Connect the IDC UART cable the BeagleBoard and using a Null-Modem serial cable connect it to a SERIAL port on a Window/Linux/Mac machine
3. Have terminal program, such as [TeraTerm](#), HyperTerminal, or [Minicom](#), running on the host machine.
4. Configure the terminal program for (BAUD RATE - 115200, DATA - 8 bit, PARITY- none, STOP - 1bit, FLOW CONTROL - none)
5. Insert the MMC/SD card (that is prepared as described above) into MMC/SD slot on Beagle Board.
6. Connect a LCD Monitor to DVI/HDMI port on the Beagle Board.
7. Connect an externally powered speaker to audio out jack on Beagle Board.
8. Connect a Line-in cable from PC or any player to Audio In jack on Beagle Board.
9. Connect a TV (NTSC-M) to S-video port.
10. Power ON LCD, TV and audio speakers.

12.5 Factory Boot Verification

The BeagleBoard comes pre-Flashed with the Xloader and UBoot in Flash. This step verifies that the board will boot properly from NAND. If the board has been flashed and the default code removed or overwritten, then you should proceed to the next step.

1. Connect the USB cable to the Host PC,
2. The power LED should come on.
3. On the terminal window the following should be printed out by the BeagleBoard:

*Texas Instruments X-Loader 1.4.2 (Feb 19 2009 - 12:01:24)
Loading u-boot.bin from nand*

U-Boot 2009.01-dirty (Feb 19 2009 - 12:22:31)

*I2C: ready
OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
OMAP3 Beagle board + LPDDR/NAND
DRAM: 256 MB
NAND: 256 MiB
*** Warning - bad CRC or NAND, using default environment*

MUSB: using high speed

In: serial usbtty

Out: serial usbtty

Err: serial usbtty

*Board revision C
Serial #486000030000000004013f8a17019010
Hit any key to stop autoboot: 10
No MMC card found
Booting from nand ...*

*NAND read: device 0 offset 0x280000, size 0x400000
4194304 bytes read: OK
Wrong Image Format for bootm command
ERROR: can't get kernel image!
OMAP3 beagleboard.org #
OMAP3 beagleboard.org #*

4. The USER LEDS and the PMU LED should be on.
5. The S-Video output should display color bars.
6. The DVI-D monitor should display a solid orange screen.

12.6 Board SD Boot

This test will force the BeagleBoard to boot from the SD card instead of the onboard Flash.

1. Press and hold the USER button while pressing and releasing the RESET button.
2. The following should be printed to the terminal window:

*40V
Texas Instruments X-Loader 1.4.2 (Feb 19 2009 - 12:01:24)
Reading boot sector
Loading u-boot.bin from mmc*

U-Boot 2009.01-dirty (Feb 19 2009 - 12:23:21)

*I2C: ready
OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
OMAP3 Beagle board + LPDDR/NAND
DRAM: 256 MB
NAND: 256 MiB
Using default environment*

*MUSB: using high speed
In: serial usbtt
Out: serial usbtt
Err: serial usbtt
Board revision C
Serial #486000030000000004013f8a17019010
Hit any key to stop autoboot: 10*

3. The USER LEDS and the PMU LED should be on.

Hitting any key on the terminal before the countdown reaches 10, will stop the booting process. If no key is hit, it will continue and flash the SW into the NAND. Go to the next section for a description of this.

12.7 Factory Boot Reinstall

This section tells you how to restore the information in the Flash to the factory default. This is the same test that is run in production when the board is new. The erase process is automatically run before the Flashing process starts.

Prior to getting to this section, follow the process and section 12.6 and do not hit any key on the terminal.

1. As long as no key is pressed on the terminal, the following will be displayed:

```
reading boot.scr  
  
679 bytes read  
Running bootscript from mmc ...  
## Executing script at 80200000  
reading x-load.bin.ift  
  
20392 bytes read
```

2. The x-load.bin.ift contains the XLoader file that will be flashed into the NAND in the following steps. The following will be displayed:

```
***** Replacing x-load *****  
Usage:  
nand - NAND sub-system  
  
HW ECC selected  
  
NAND erase: device 0 offset 0x0, size 0x80000  
  
Erasing at 0x0 -- 25% complete.  
Erasing at 0x20000 -- 50% complete.  
Erasing at 0x40000 -- 75% complete.  
Erasing at 0x60000 -- 100% complete.  
OK  
  
NAND write: device 0 offset 0x0, size 0x20000  
131072 bytes written: OK  
  
NAND write: device 0 offset 0x20000, size 0x20000  
131072 bytes written: OK
```

NAND write: device 0 offset 0x40000, size 0x20000
131072 bytes written: OK

NAND write: device 0 offset 0x60000, size 0x20000
131072 bytes written: OK

3. The u-boot-f.bin contains the UBoot file that will be flashed into the NAND in the following steps. The following will be displayed:

reading u-boot-f.bin

275928 bytes read
***** Replacing u-boot *****
Usage:
nand - NAND sub-system

SW ECC selected

NAND erase: device 0 offset 0x80000, size 0x160000

Erasing at 0x80000 -- 9% complete.
Erasing at 0xa0000 -- 18% complete.
Erasing at 0xc0000 -- 27% complete.
Erasing at 0xe0000 -- 36% complete.
Erasing at 0x100000 -- 45% complete.
Erasing at 0x120000 -- 54% complete.
Erasing at 0x140000 -- 63% complete.
Erasing at 0x160000 -- 72% complete.
Erasing at 0x180000 -- 81% complete.
Erasing at 0x1a0000 -- 90% complete.
Erasing at 0x1c0000 -- 100% complete.
OK

NAND write: device 0 offset 0x80000, size 0x160000
1441792 bytes written: OK

4. After the XLoader and UBot are flashed, the environment variables are erased to insure proper booting of the Kernel image that I son the SD card. . The following will be displayed:

***** Erasing environment settings *****
Usage:
nand - NAND sub-system

NAND erase: device 0 offset 0x160000, size 0x20000

Erasing at 0x160000 -- 100% complete.
OK

- At this point you can remove power to stop the Kernel from booting, or just let it continue the Kernel boot process. For information on the Kernel booting process, proceed to the next section.

12.8 Booting the Kernel

This section describes how to boot the kernel from the SD card. In order to complete this section, you must have completed section 12.7 and do not hit any keys or remove power after the NAND has been flashed.

1. After the NAND has been flashed, the normal.scr script is read from the SD card and the first step after that is to load in the uImage.bin file into the SDRAM. Beagle will print the following to the terminal:

```
***** Executing normal.scr *****
## Executing script at 80200000
reading ulmage.bin

2578044 bytes read
***** Kernel: /dev/mmcblk0p1/ulmage.bin *****
```

2. Then the root filesystem is read into SDRAM. The BeagleBoard will output the following:

```
reading ramdisk.gz  
7999649 bytes read  
***** RootFS: /dev/mmcblk0p1/ramdisk.gz *****  
1856680 bytes read
```

- At this point, the booting process will start. The following will be printed to the terminal:

```
## Booting kernel from Legacy Image at 80200000 ...
Image Name: Linux-2.6.28-omap1
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 2577980 Bytes = 2.5 MB
Load Address: 80008000
Entry Point: 80008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
OK

Starting kernel ...

Uncompressing
Linux.....
..... done, booting the kernel.

Linux version 2.6.28-omap1 (root@tiross) (gcc version 4.2.1 (CodeSourcery Sourcery G++ Lite
2007q3-51)) #2 Thu Feb 19 12:45:34 IST 2009
CPU: ARMv7 Processor [411fc083] revision 3 (ARMv7), cr=10c5387f
CPU: VIRT: nonaliasing data cache, VIRT: nonaliasing instruction cache
```



Machine: OMAP3 Beagle Board
Memory policy: ECC disabled, Data cache writeback
OMAP3430 ES3.0
SRAM: Mapped pa 0x40200000 to va 0xd7000000 size: 0x100000
Reserving 15728640 bytes SDRAM for VRAM
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 65024
Kernel command line: console=ttyS2,115200n8 console=tty0 root=/dev/ram0 rw ramdisk_size=32768 initrd=0x81600000,32M
Clocking rate (Crystal/DPLL/ARM core): 26.0/332/500 MHz
GPMC revision 5.0
IRQ: Found an INTC at 0xd8200000 (revision 4.0) with 96 interrupts
Total of 96 interrupts on 1 active controller
OMAP34xx GPIO hardware version 2.5
PID hash table entries: 1024 (order: 10, 4096 bytes)
OMAP clockevent source: GPTIMER12 at 32768 Hz
Console: colour dummy device 80x30
console [tty0] enabled
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 128MB 128MB = 256MB total
Memory: 206080KB available (4776K code, 425K data, 168K init)
Calibrating delay loop... 473.71 BogomIPS (lpj=1851392)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
net_namespace: 532 bytes
regulator: core version 0.5
NET: Registered protocol family 16
Found NAND on CS0
Registering NAND on CS0
OMAP DMA hardware revision 4.0
USB: No board-specific platform config found
OMAP DSS rev 2.0
OMAP DISPC rev 3.0
OMAP VENC rev 2
OMAP DSI rev 1.0
i2c_omap i2c_omap.1: bus 1 rev3.12 at 2600 kHz
twl4030: PIH (irq 7) chaining IRQs 368..375
twl4030: power (irq 373) chaining IRQs 376..383
twl4030: gpio (irq 368) chaining IRQs 384..401
i2c_omap i2c_omap.3: bus 3 rev3.12 at 400 kHz
SCSI subsystem initialized
twl4030_usb twl4030_usb: Initialized TWL4030 USB module
usbcore: registered new interface driver usbfis
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
musb_hdrc: version 6.0, musb-dma, otg (peripheral+host), debug=0
musb_hdrc: USB OTG mode controller at d80ab000 using DMA, IRQ 92
regulator: VMMC1: 1850 <-> 3150 mV normal standby
regulator: VDAC: 1800 mV normal standby

regulator: VUSB1V5: 1500 mV normal standby
regulator: VUSB1V8: 1800 mV normal standby
regulator: VUSB3V1: 3100 mV normal standby
regulator: VSIM: 1800 <-> 3000 mV normal standby
Bluetooth: Core ver 2.13
NET: Registered protocol family 31
Bluetooth: HCI device and connection manager initialized
Bluetooth: HCI socket layer initialized
cfg80211: Using static regulatory domain info
cfg80211: Regulatory domain: US
(start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp)
(2402000 KHz - 2472000 KHz @ 40000 KHz), (600 mBi, 2700 mBm)
(5170000 KHz - 5190000 KHz @ 40000 KHz), (600 mBi, 2300 mBm)
(5190000 KHz - 5210000 KHz @ 40000 KHz), (600 mBi, 2300 mBm)
(5210000 KHz - 5230000 KHz @ 40000 KHz), (600 mBi, 2300 mBm)
(5230000 KHz - 5330000 KHz @ 40000 KHz), (600 mBi, 2300 mBm)
(5735000 KHz - 5835000 KHz @ 40000 KHz), (600 mBi, 3000 mBm)
cfg80211: Calling CRDA for country: US

NET: Registered protocol family 2
IP route cache hash table entries: 2048 (order: 1, 8192 bytes)
TCP established hash table entries: 8192 (order: 4, 65536 bytes)
TCP bind hash table entries: 8192 (order: 3, 32768 bytes)

TCP: Hash tables configured (established 8192 bind 8192)
TCP reno registered
NET: Registered protocol family 1
checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
Freeing initrd memory: 32768K
VFS: Disk quotas dquot_6.5.1
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
JFFS2 version 2.2. (NAND) (SUMMARY) © 2001-2006 Red Hat, Inc.
msgmni has been set to 467
alg: No test for stdrng (krng)
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered (default)
Serial: 8250/16550 driver4 ports, IRQ sharing enabled
serial8250.0: ttyS0 at MMIO 0x4806a000 (irq = 72) is a ST16654
serial8250.0: ttyS1 at MMIO 0x4806c000 (irq = 73) is a ST16654
serial8250.0: ttyS2 at MMIO 0x49020000 (irq = 74) is a ST16654
console [ttyS2] enabled
brd: module loaded
loop: module loaded
usbcore: registered new interface driver asix
usbcore: registered new interface driver cdc_ether
usbcore: registered new interface driver rndis_host
usbcore: registered new interface driver zd1211rw
usbcore: registered new interface driver rndis_wlan
usbcore: registered new interface driver zd1201
usbcore: registered new interface driver usb8xxx
usbcore: registered new interface driver rtl8187
usbcore: registered new interface driver rt2500usb
usbcore: registered new interface driver rt73usb
usbcore: registered new interface driver p54usb
i2c /dev entries driver
input: triton2-pwrbutton as /class/input/input0
triton2 power button driver initialized
Driver 'sd' needs updating - please use bus_type methods
Driver 'sr' needs updating - please use bus_type methods
omap2-nand driver initializing
NAND device: Manufacturer ID: 0x2c, Chip ID: 0xba (Micron NAND 256MiB 1,8V 16-bit)
cmdlinepart partition parsing not available
Creating 5 MTD partitions on "omap2-nand":
0x00000000-0x00080000 : "X-Loader"
0x00080000-0x00260000 : "U-Boot"
0x00260000-0x00280000 : "U-Boot Env"
0x00280000-0x00680000 : "Kernel"
0x00680000-0x10000000 : "File System"
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
ehci-omap ehci-omap.0: OMAP-EHCI Host Controller
ehci-omap ehci-omap.0: new USB bus registered, assigned bus number 1
ehci-omap ehci-omap.0: irq 77, io mem 0x48064800
ehci-omap ehci-omap.0: USB 2.0 started, EHCI 1.00
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 3 ports detected
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
g_ether gadget: using random self ethernet address
g_ether gadget: using random host ethernet address
usb0: MAC 22:46:4b:c2:b0:fb

usb0: HOST MAC a2:b4:44:63:f6:ae
g_ether gadget: Ethernet Gadget, version: Memorial Day 2008
g_ether gadget: g_ether ready

```
musb_hdrc musb_hdrc: MUSB HDRC host driver
musb_hdrc musb_hdrc: new USB bus registered, assigned bus number 2
usb usb2: configuration #1 chosen from 1 choice
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
mice: PS/2 mouse device common for all mice
input: gpio-keys as /class/input/input1
twl4030_rtc twl4030_rtc: rtc core: registered twl4030_rtc as rtc0
twl4030_rtc twl4030_rtc: Power up reset detected.
twl4030_rtc twl4030_rtc: Enabling TWL4030-RTC.
OMAP Watchdog Timer Rev 0x31: initial timeout 60 sec
Bluetooth: HCI USB driver ver 2.10
usbcore: registered new interface driver hcj_usb
Bluetooth: Broadcom Blutonium firmware driver ver 1.2
usbcore: registered new interface driver bcm203x
Bluetooth: Digianswer Bluetooth USB driver ver 0.10
usbcore: registered new interface driver bpa10x
Bluetooth: Generic Bluetooth SDIO driver ver 0.1
mmci-omap-hs mmci-omap-hs.0: Failed to get debounce clock
Registered led device: beagleboard::usr0
Registered led device: beagleboard::usr1
leds-gpio: probe of leds-gpio failed with error -22
usbcore: registered new interface driver ushhid
ushhid: v2.6:USB HID core driver
Advanced Linux Sound Architecture Driver Version 1.0.18rc3.
usbcore: registered new interface driver snd-usb-audio
ASoC version 0.13.2
OMAP3 Beagle SoC init
TWL4030 Audio Codec init
asoc: twl4030 <-> omap-mcbsp-dai-(link_id) mapping ok
ALSA device list:
#0: omap3beagle (twl4030)
oprofile: using arm/armv7
TCP cubic registered
NET: Registered protocol family 17

NET: Registered protocol family 15
Bluetooth: L2CAP ver 2.11
Bluetooth: L2CAP socket layer initialized
Bluetooth: SCO (Voice Link) ver 0.6
Bluetooth: SCO socket layer initialized
Bluetooth: RFCOMM socket layer initialized
Bluetooth: RFCOMM TTY layer initialized
Bluetooth: RFCOMM ver 1.10
Bluetooth: BNEP (Ethernet Emulation) ver 1.3
Bluetooth: BNEP filters: protocol multicast
Bluetooth: HIDP (Human Interface Emulation) ver 1.2
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
ieee80211: 802.11 data/management/control stack, git-1.1.13
ieee80211: Copyright (C) 2004-2005 Intel Corporation <jketreno@linux.intel.com>
ThumbEE CPU extension supported.
Power Management for TI OMAP3.
SmartReflex driver initialized
Disabling unused clock "sr2_fck"
Disabling unused clock "sr1_fck"
Disabling unused clock "mcbsp_fck"
Disabling unused clock "mcbsp_fck"
Disabling unused clock "mcbsp_fck"
Disabling unused clock "mcbsp_ick"
Disabling unused clock "mcbsp_ick"
Disabling unused clock "mcbsp_ick"
Disabling unused clock "gpt2_ick"
Disabling unused clock "gpt3_ick"
Disabling unused clock "gpt4_ick"
Disabling unused clock "gpt5_ick"
Disabling unused clock "gpt6_ick"
Disabling unused clock "gpt7_ick"
```

*Disabling unused clock "gpt8_ick"
Disabling unused clock "gpt9_ick"
Disabling unused clock "wdt3_ick"
Disabling unused clock "wdt3_fck"
Disabling unused clock "gpio2_dbck"
Disabling unused clock "gpio3_dbck"
Disabling unused clock "gpio4_dbck"
Disabling unused clock "gpio5_dbck"
Disabling unused clock "gpio6_dbck"
Disabling unused clock "gpt9_fck"
Disabling unused clock "gpt8_fck"
Disabling unused clock "gpt7_fck"
Disabling unused clock "gpt6_fck"
Disabling unused clock "gpt5_fck"
Disabling unused clock "gpt4_fck"
Disabling unused clock "gpt3_fck"
Disabling unused clock "gpt2_fck"
Disabling unused clock "gpt1_ick"
Disabling unused clock "wdt1_ick"
Disabling unused clock "wdt2_ick"
Disabling unused clock "wdt2_fck"
Disabling unused clock "gpio1_dbck"
Disabling unused clock "gpt1_fck"
Disabling unused clock "cam_ick"

Disabling unused clock "cam_mclk"
Disabling unused clock "des1_ick"

Disabling unused clock "sha11_ick"
Disabling unused clock "rng_ick"

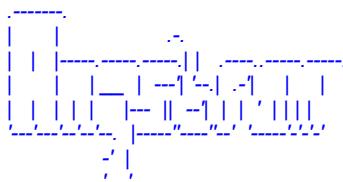
Disabling unused clock "aes1_ick"
Disabling unused clock "ssi_ick"
Disabling unused clock "mailboxes_ick"
Disabling unused clock "mcbsp_ick"
Disabling unused clock "mcbsp_ick"
Disabling unused clock "gpt10_ick"
Disabling unused clock "gpt11_ick"
Disabling unused clock "i2c_ick"
Disabling unused clock "mcspi_ick"
Disabling unused clock "mcspi_ick"
Disabling unused clock "mcspi_ick"
Disabling unused clock "mcspi_ick"
Disabling unused clock "hdq_ick"
Disabling unused clock "mspro_ick"
Disabling unused clock "des2_ick"
Disabling unused clock "sha12_ick"
Disabling unused clock "aes2_ick"
Disabling unused clock "icr_ick"
Disabling unused clock "pka_ick"
Disabling unused clock "ssi_ssrfck"
Disabling unused clock "hdq_fck"
Disabling unused clock "mcspi_fck"
Disabling unused clock "mcspi_fck"
Disabling unused clock "mcspi_fck"
Disabling unused clock "mcspi_fck"
Disabling unused clock "mcbsp_fck"
Disabling unused clock "mcbsp_fck"
Disabling unused clock "i2c_fck"
Disabling unused clock "mspro_fck"
Disabling unused clock "gpt11_fck"
Disabling unused clock "gpt10_fck"
Disabling unused clock "dpll4_m6x2_ck"
Disabling unused clock "dpll3_m3x2_ck"
Disabling unused clock "sys_clkout1"*

VFP support v0.3: implementor 41 architecture 3 part 30 variant c rev 1
Console: switching to colour frame buffer device 80x30
clock: clksel_round_rate_div: dpll4_m4_ck target_rate 48000000
clock: new_div = 9, new_rate = 48000000

```

omap-dss DISPC error: dispc irq error status 4024
omap-dss DISPC error: dispc irq error status 4000
omap-dss DISPC error: dispc irq error status 4000
omap-dss DISPC error: dispc irq error status 4022
omap-dss DISPC error: dispc irq error status 4000
omap-dss DISPC error: dispc irq error status 4000
omap-dss DISPC error: dispc irq error status 4022
omap-dss DISPC error: dispc irq error status 4000
omap-dss DISPC error: dispc irq error status 4022
omap-dss DISPC error: dispc irq error status 4000
omap-dss DISPC error: dispc irq error status 4000
omap-dss DISPC error: dispc irq error status 4022
omap-dss DISPC error: dispc irq error status 4000
omap-dss DISPC error: Excessive DISPC errors
Turning off lcd and digit
omap-dss DISPC error: Excessive DISPC errors
Turning off lcd and digit
omap-dss DISPC error: Excessive DISPC errors
Turning off lcd and digit
usb 2-1: new high speed USB device using musb_hdrc and address 2
twl4030 RTC twl4030_RTC: setting system clock to 2000-01-01 00:00:00 UTC (946684800)
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 168K
mmc0: new high speed SD card at address b368
mmcblk0: mmc0:b368 SD 970 MiB
mmcblk0: p1
usb 2-1: device v4146 pba01 is not supported
usb 2-1: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
udevd version 124 started
uncorrectable error : <3>end_request: I/O error, dev mtdblock0, sector 0
uncorrectable error : <3>end_request: I/O error, dev mtdblock0, sector 8
uncorrectable error : <3>end_request: I/O error, dev mtdblock0, sector 16
uncorrectable error : <3>end_request: I/O error, dev mtdblock0, sector 24
uncorrectable error : <3>end_request: I/O error, dev mtdblock0, sector 0
scsi 0:0:0:0: Direct-Access Pretec 256MB Tiny 1.30 PQ: 0 ANSI: 2
sd 0:0:0:0: [sda] 512000 512-byte hardware sectors: (262 MB/250 MiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 512000 512-byte hardware sectors: (262 MB/250 MiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
sd 0:0:0:0: Attached scsi generic sg0 type 0

```



The Angstrom Distribution beagleboard ttyS2

Angstrom 2008.1-test-20090127 beagleboard ttyS2

beagleboard login:

4. Type **root** and hit <enter>. You now are in the Linux kernel.

12.9 UBoot Tests

There are a series of tests that are run while in UBoot. This requires that the Beagle is not in the Kernel mode, but the UBoot mode. The UBoot mode is entered by hitting a key prior to the UBoot timeout reaching 10.

The following sections describe each test and how it is to be run.

12.9.1 EDID Test

This test will display the EDID (Enhanced Display ID) from the DVI-D monitor by using the I2C interface on the DVI-D connector. The DVI-D connector must be connected to a DVI-D compatible monitor in order to run this test..

1. Type the following commands:

```
OMAP3 beagleboard.org # ibus 2
OMAP3 beagleboard.org# imd 0x50 0 100
```

2. Something similar to the following will be displayed:

```
0000: 00 ff ff ff ff ff 00 10 ac 24 40 5a 39 41 41 .....$@Z9AA
0010: 1f 11 01 03 80 22 1b 78 ee ae a5 a6 54 4c 99 26 ....".x....TL.&
0020: 14 50 54 a5 4b 00 71 4f 81 80 01 01 01 01 01 01 .PT.K.qO.....
0030: 01 01 01 01 01 01 30 2a 00 98 51 00 2a 40 30 70 .....0*.Q.*@0p
0040: 13 00 52 0e 11 00 00 1e 00 00 00 ff 00 50 4d 30 ..R.....PM0
0050: 36 31 37 38 32 41 41 39 5a 0a 00 00 00 fc 00 44 61782AA9Z.....D
0060: 45 4c 4c 20 31 37 30 38 46 50 0a 20 00 00 00 fd ELL 1708FP. ....
0070: 00 38 4c 1e 51 0e 00 0a 20 20 20 20 20 20 00 36 .8L.Q... .6
```

Note the words "DELL 1708FP" which is the ID of the monitor in this example. It will be different based on the display manufacturer of your display. For more detailed information on the full EDID format, refer to: <http://en.wikipedia.org/wiki/EDID>

12.9.2 LED Test

This test checks out the PWM and USER0/1 LEDs on the Beagle.

1. Type the following commands followed by the <ENTER> key and verify that the correct results are seen on LEDs USR0 and USR1.

```
OMAP3 beagleboard.org # mw 0x49056090 0x00600000 [USR0 & USR1 OFF]
OMAP3 beagleboard.org # mw 0x49056094 0x00400000 [USR0 ON]
OMAP3 beagleboard.org # mw 0x49056094 0x00200000 [USR1 ON]
```

2. Type the following commands and verify that the correct results are seen on PMU LED.

```
OMAP3 beagleboard.org # ibus 0
```

```
OMAP3 beagleboard.org # imm 0x4A 0xEE <ENTER>
000000ee: 22 ? 00 <ENTER> [PMU LED OFF]
<CTRL-C> <CTRL-C>
OMAP3 beagleboard.org # imm 0x4A 0xEE
000000ee: 22 ? 22 <ENTER> [PMU LED ON]
<CTRL-C> <CTRL-C>
```

12.9.3 DVI-D Test

This test checks the DVI-D interface for proper operation. It sends various colors to the DVI-D monitor.

1. Type the following commands followed by the <ENTER> key and verify the correct results are seen.

```
OMAP3 beagleboard.org # mw 0x49058090 0x00000400 [DISPLAY TURNS OFF]
OMAP3 beagleboard.org # mw 0x49058094 0x00000400 [DISPLAY TURNS ON]
OMAP3 beagleboard.org # mw 0x80500000 07e007e0 7fffff [DISPLAY TURNS GREEN]
OMAP3 beagleboard.org # mw 0x80500000 001f001f 7fffff [DISPLAY TURNS BLUE]
OMAP3 beagleboard.org # mw 0x80500000 00000000 7fffff [DISPLAY TURNS BLACK]
OMAP3 beagleboard.org # mw 0x80500000 ffffffff 7fffff [DISPLAY TURNS WHITEN]
OMAP3 beagleboard.org # mw 0x80500000 f800f800 7fffff [DISPLAY TURNS RED]
```

12.10 Kernel Based Tests

The following tests require that the Kernel is loaded and that you have logged into the Kernel. [See section 12.8]

12.10.1 DVI-D Test

This test plays a short video clip to the DVI-D monitor.

1. Type the following command:

```
root@beagleboard:~# mplayer /sample_video.avi
```

2. It will display a 320x240 video on the DVI screen. The video has been downloaded from <https://garage.maemo.org/download.php/54/269/2380/bug.avi>

12.10.2 S-Video Test

1. Type the following command:

```
root@beagleboard:/mmc# svideo
```

2. Type the following command to start the video:

```
[root@beagleboard:/mmc# mplayer /sample_video.avi
```

3. It will display a 320x240 video on the DVI screen. The video has been downloaded from <https://garage.maemo.org/download.php/54/269/2380/bug.avi>

12.10.3 Audio Test

The audio test is divided into two test, one for audio in and one for audio out. Audio is recorded into the audio in port and then played out the audio out port.

12.10.3.1 *Audio In*

1. Make Sure your player is running and Audio Line in is connected to board.
2. Make sure that you are in the MMC directory. If you are, proceed to step 4. If not, then type the following command:

```
root@beagleboard:~# mkdir /mmc
root@beagleboard:~# mount -t vfat /dev/mmcblk0p1 /mmc
```

3. Change the directory by typing:

```
root@beagleboard:/mmc# cd /mmc
```

4. Type the following command:

```
root@beagleboard:/mmc# arecord -t wav -c 2 -r 44100 -f S16_LE -v /mmc/k
```

5. The following output is expected on the terminal window:

```
Recording WAVE '/mmc/k' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Plug PCM: Hardware PCM card 0 'omap3beagle' device 0 subdevice 0
Its setup is:
stream    : CAPTURE
access    : RW_INTERLEAVED
format    : S16_LE
subformat : STD
channels  : 2
rate      : 44100
exact rate: 44100 (44100/1)
msbits    : 16
buffer_size: 22052
period_size: 5513
period_time: 125011
tstamp_mode: NONE
period_step: 1
avail_min : 5513
period_event: 0
```

```
start_threshold : 1
stop_threshold : 22052
silence_threshold: 0
silence_size : 0
boundary    : 1445199872
```

- When you want to stop the recording process just press <CONTROL+C>.

12.10.3.2 Audio Out

NOTE: It is expected that you have previously recorded an audio file to be played and that you are still in the MMC directory.

- Type the following command:

```
root@beagleboard:/mmc# aplay -t wav -c 2 -r 44100 -f S16_LE -v k
```

- The recorded audio should be heard on the Speakers,
- The following output is expected on terminal window:

```
Playing WAVE '/mmc/k' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Plug PCM: Hardware PCM card 0 'omap3beagle' device 0 subdevice 0
Its setup is:
stream      : PLAYBACK
access      : RW_INTERLEAVED

format      : S16_LE
subformat   : STD
channels    : 2
rate        : 44100
exact rate  : 44100 (44100/1)
msbits      : 16
buffer_size : 22052
period_size : 5513
period_time : 125011
tstamp_mode : NONE
period_step : 1
avail_min   : 5513
period_event: 0
start_threshold : 22052
stop_threshold : 22052
silence_threshold: 0
silence_size : 0
boundary    : 1445199872
```

- To stop the audio playback just press <CONTROL+C>. If you choose, you can let the recorded audio play out. It will stop when it reaches the end of the recorded file.

12.10.3.3 Keyboard Test

This test runs on the OTG port in the Host mode. It requires that a Powered USB hub be used, and that the Hub and device (Keyboard or mouse) be connected when the Linux OS is booted. This section is broken down into two sections, one for the mouse and the other for the keyboard.

NOTE: This test is run after the OS is booted with the Hub and Keyboard plugged in.

1. Make sure that you are in the MMC directory. If you are, proceed to step 3. If not, then type the following command:

```
root@beagleboard:~# mkdir /mmc
root@beagleboard:~# mount -t vfat /dev/mmcblk0p1 /mnt/mmc/cd /mnt/mmc/
```

2. Change the directory by typing:

```
root@beagleboard:~# cd /mmc
```

3. Type the following command:

```
root@beagleboard:/mmc]# evtest /dev/input/event1
```

4. Press a Key on USB Keyboard and look for a printout in the terminal window.

Example if "a" is pressed the following output is seen on Console:

```
Event: time 1657.754638, type 1 (Key), code 30 (A), value 1
Event: time 1657.754638, ----- Report Sync -----
Event: time 1657.964599, type 1 (Key), code 30 (A), value 0
Event: time 1657.964599, ----- Report Sync -----
```

5. Press <CONTROL+C> to stop the test.

12.10.3.4 Mouse Test

NOTE: This test is run after the Kernel booted with the Hub and mouse plugged in.

1. Make sure that you are in the MMC directory. If you are, proceed to step 3. If not, then type the following command:

```
root@beagleboard:~# mount -t vfat /dev/mmcblk0p1 /mmc/
```

2. Change the directory by typing:

```
root@beagleboard:~# cd /mnt/mmc/
```

3. Type the following command:

```
root@beagleboard:/mmc# evtest /dev/input/event0
```

4. Press mouse button and look for a printout in the terminal window.

Example if Left button is pressed and released the following lines should get displayed on console

```
Event: time 1871.724792, ----- Report Sync -----  
Event: time 1873.804687, type 1 (Key), code 272 (LeftBtn), value 1  
Event: time 1873.804687, ----- Report Sync -----  
Event: time 1873.964660, type 1 (Key), code 272 (LeftBtn), value 0  
Event: time 1873.964660, ----- Report Sync -----
```

5. Moving the Mouse also results in Console messages

```
Event: time 1959.120635, ----- Report Sync -----  
Event: time 1959.130676, type 2 (Relative), code 0 (X), value -21  
Event: time 1959.130676, ----- Report Sync -----  
Event: time 1959.140625, type 2 (Relative), code 0 (X), value -16
```

6. Press <CONTROL+C> to stop the test

12.10.3.5 USB EHCI Test

The following test will copy a file from the SD card to the USB EHCI port and back. The file name can be changed to anything on the SD card. You must have a USB ThumbDrive installed in the EHCI port at power up.

Start in the root directory and make sure the MMC directory is already mounted.

1. Type the following commands to set up the test:

```
root@beagleboard:~# mkdir /usb1  
root@beagleboard:~# mount /dev/sda1 /usb1
```

2. Type the following commands to copy from the SD card to the USB Drive:

```
root@beagleboard:~# cp /mmc/u-boot.bin /usb1/test.bin
```

3. Type the following command to make sure the file was copied to the USB drive:

```
root@beagleboard:~# ls -al /usb1
```

The file should be listed in the directory.

4. Type the following command to copy the file from the USB drive to the SD card.

```
root@beagleboard:~# cp /usb1/test.bin /mmc1/test.bin
```

5. Type the following command to copy to make sure the file was copied to the SD card.

```
root@beagleboard:~# ls -al /mmc
```

The file should be listed in the directory. Larger files can be used to create a longer test if desired.

13.0 Troubleshooting

This section will provide assistance in troubleshooting the BeagleBoard in the event there are questions raised as to what the state of the BeagleBoard is. This may be due to a HW failure or the SW not initializing things properly during development. Also provided is a section of known issues. Be sure and check with BeagleBoard.org for any updates.

For an up to date listing of common questions and their answers, please refer to <http://elinux.org/BeagleBoardFAQ>

13.1 Access Points

This section covers the various access points where various signals and voltages can be measured.

13.1.1 Voltage Points

Figure 81 shows the test points for the various voltages on BeagleBoard.

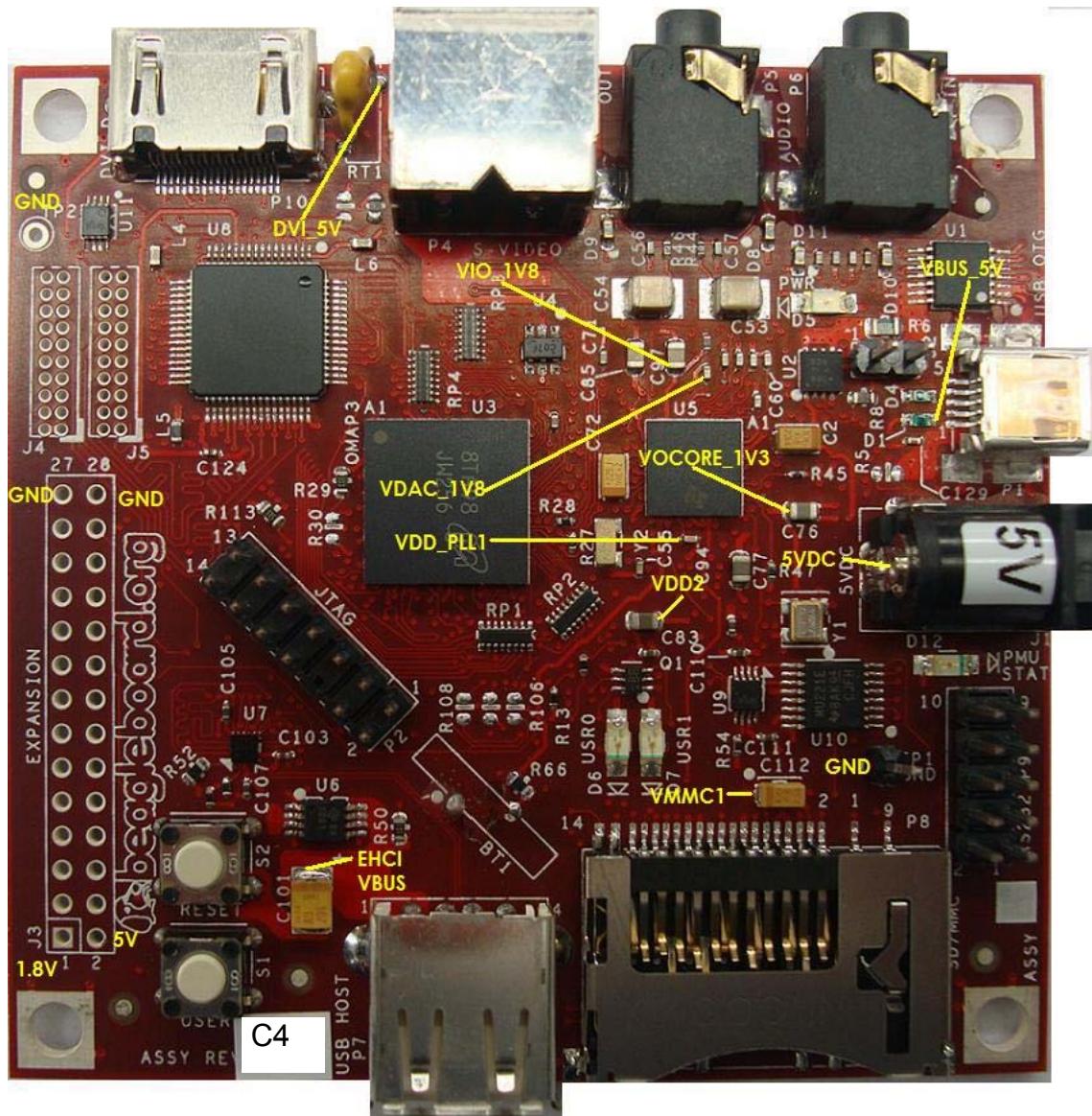


Figure 78. BeagleBoard Voltage Access Points

Some of these voltages may not be present depending on the state of the TWL4030 as set by the OMAP3530. Others may be at different voltage levels depending on the same factor.

Table 40 provides the ranges of the voltages and the definition of the conditions as applicable.

Table 39. Voltages

Voltage	Min	Nom	Max	Conditions
VIO_1V8	1.78	1.8	1.81	
VDD_SIM	1.78	1.8	1.81	
VBUS_5V0	4.9	5.0	5.2	From the host PC. May be lower or higher.
VOCORE_1V3	1.15	1.2	1.4	Can be set via SW. Voltage levels may vary.
VBAT	4.1	4.2	4.3	
VDAC_1V8	1.78	1.8	1.81	
VDD_PLL1	1.78	1.8	1.81	
VDD_PLL2	1.78	1.8	1.81	
VDD2	1.15	1.2	1.25	
3.3V	3.28	3.3	3.32	
VMMC1 (3V)	2.9	3.0	3.1	3.0V at power up. Can be set to via SW.
VMMC1(1.8V)	1.78	1.8	1.81	

13.1.2 Signal Access Points

Figure 82 shows the access points for various signals on BeagleBoard.

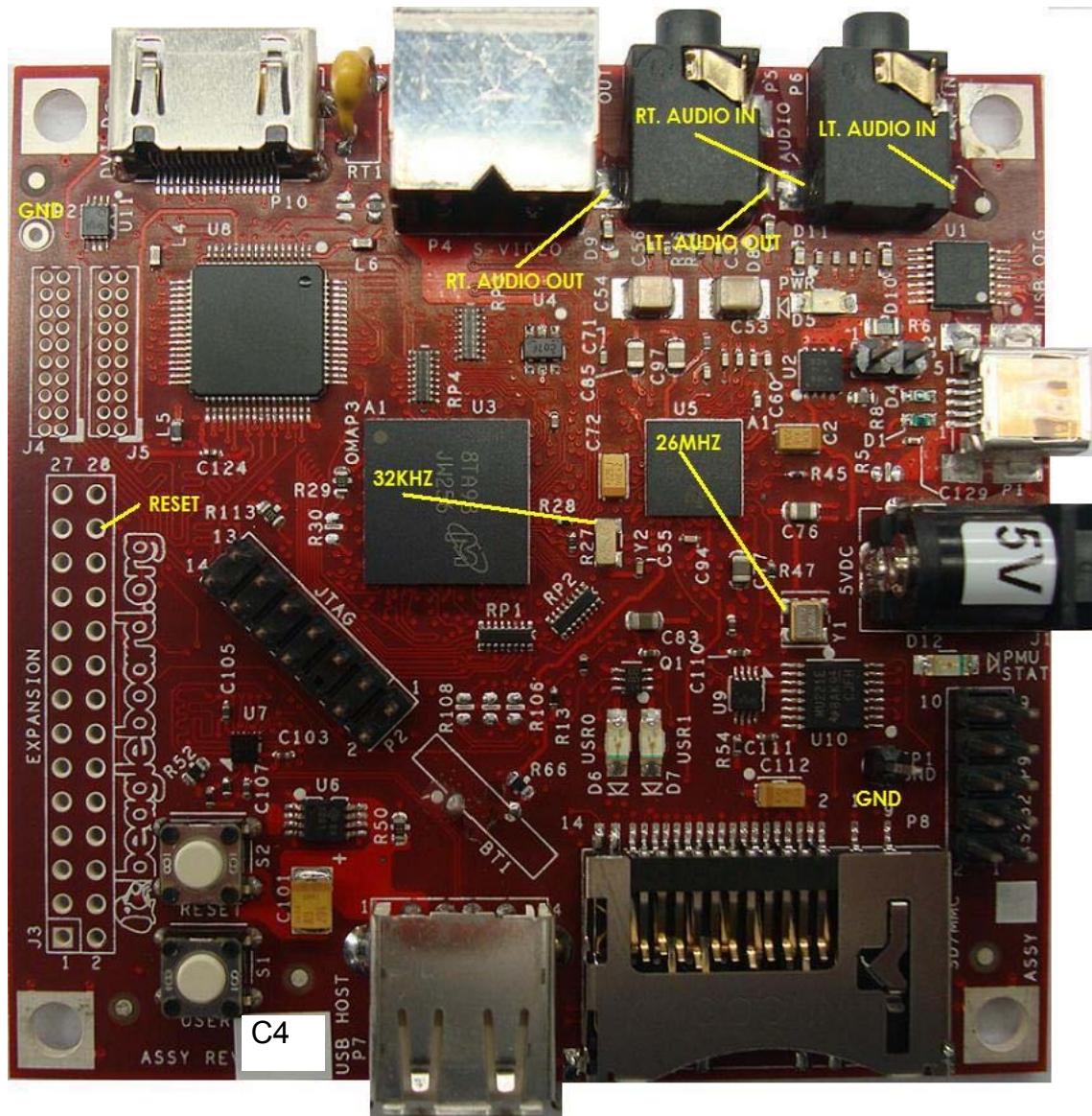


Figure 79. BeagleBoard Signal Access Points

13.2 Troubleshooting Guide

Table 41 provides a list of possible failure modes and conditions and suggestions on how to diagnose them and ultimately determine whether the HW is operational or not.

Table 40. Troubleshooting

Symptoms	Possible Problem	Action
JTAG does not connect.	Verify that the Power LED is on.	If off and running over USB, the PC may have shut down the voltage due to excessive current as related to what it is capable of providing. Remove the USB cable and re insert. If running on a DC supply make sure that voltage is being supplied.
	JTAG interface needs to be reset	Reset the BeagleBoard.
UBoot does not start, and no activity on the RS232 monitor.	Incorrect serial cable configuration.	Verify orientation of the RS232 flat cable Check for the right null modem cable.
	If a 40V is displayed over the serial cable, processor is booting. Issue could be the SD/MMC card.	Make sure the SD/MMC card is installed all the way into the connector. Make sure the card is formatted correctly and that the MLO file is the first file written to the SD card.
USB Host Connection Issues via OTG.	Cheap USB Cable. OTG cables are typically not designed for higher current. The expect 100mA max.	Measure the voltage at the card to determine the voltage drop across the cable. If it the level is below 4.35V, the USB power is not guaranteed to work,

13.3 Serial Port Issues

We have had several serial port issues in the field caused by different issues. This section attempts to provide a step by step process to identify what the issue is.

The main thing to keep in mind is that the PC and the BeagleBoard connectors are wired the same. In order for them to talk, they must have a null modem cable to connect them.

The following sections provide steps to help identify the issue.

For additional help on debugging serial issues, refer to the FAQ at
http://elinux.org/BeagleBoardFAQ#Serial_connection_231

13.3.1 First Step

1. Review the wiring of your IDC10 to DB9M serial adapter. Only the TX, RX and GND signals are used.
2. Make sure that the cable is plugged in correctly. The red stripe should be at the bottom next to pin1 of P9. Some cables may have the flat cable extending away from the BeagleBoard and others may be extending toward the middle of the BeagleBoard. **Figure 83** shows the proper orientation of the IDC serial cable.

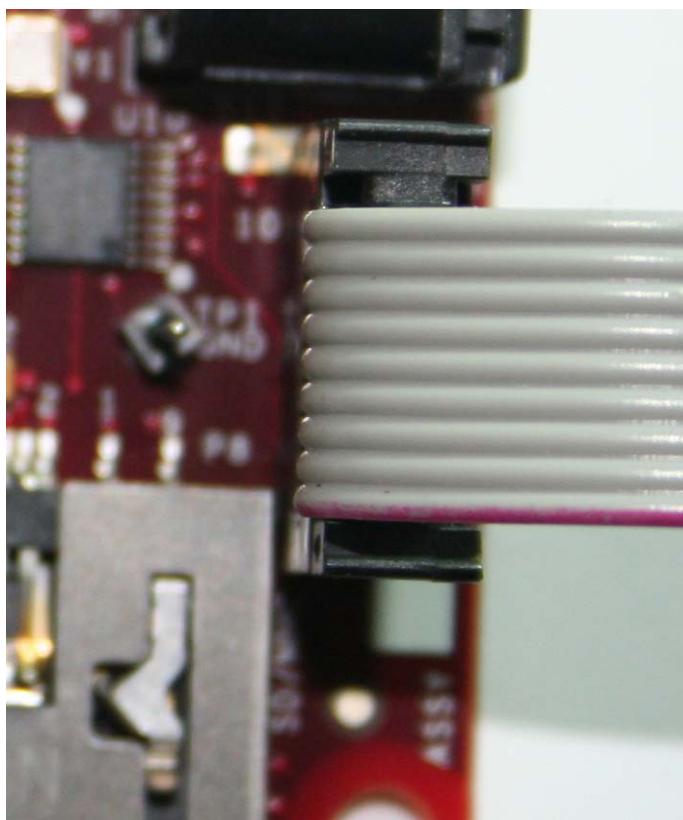


Figure 80. BeagleBoard Serial Cable Orientation

3. You must have a Null Modem cable to connect to a PC. This results in the TX and RX leads being swapped, connecting the TX of the BeagleBoard to RX of the PC and RX of the BeagleBoard to TX of the PC. This cable also must be a female to female cable as the connectors on the BeagleBoard and PC are male. **Figure 84** shows the DB9 male connector and **Figure 85** shows the Null Modem Cable.



Figure 81. DB9 Male Connector



Figure 82. DB9 Null Modem Cable

4. If you have an ohmmeter, you can measure to see if the pins are swapped between pins 2 and 3 from each end of the cable.

13.3.2 Second Step

A simple test to verify that the cables you are using are correct to create a loopback on the cable. This checks the IDC cable and the null modem cable for connections.

1. Connect a wire across the TX and RX leads (Pins 2 and Pins 3) of the cable that plugs into the BeagleBoard (IDC Cable). **Figure 86** shows how this is done.

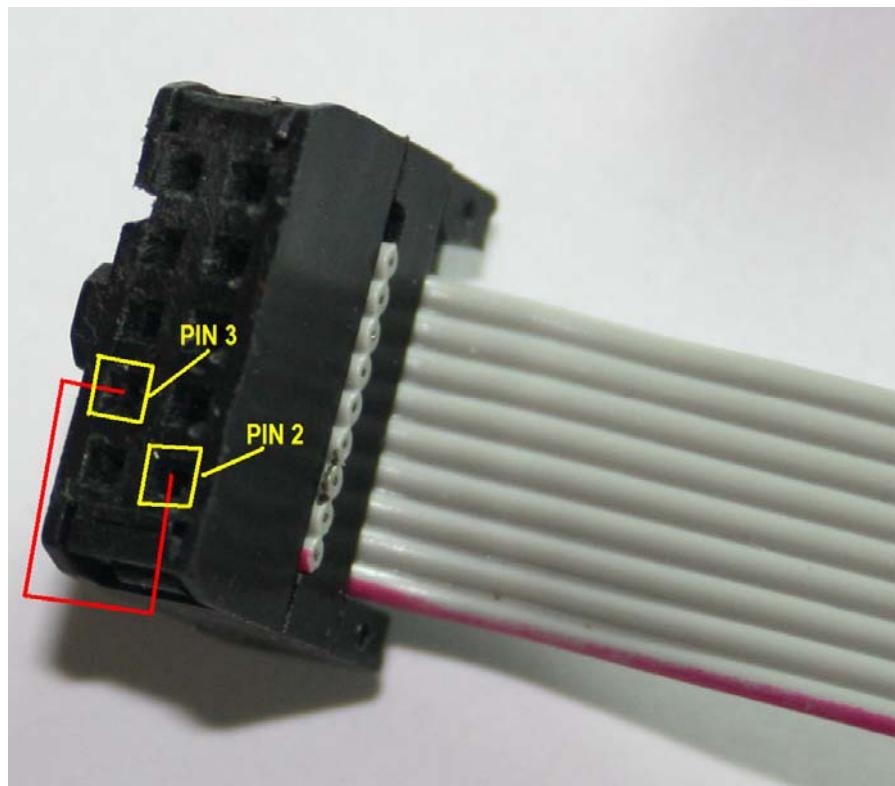


Figure 83. Serial Cable Loopback

2. On your terminal start typing.
3. If the correct characters are echoed back, then the cables are in the proper configuration. Note that this checks the electrical connection only. If the terminal program is set wrong, then serial port will still not work.

13.3.3 Third Step

1. Make sure your terminal settings are correct.

- BAUD RATE: 115200
- DATA: 8 bit
- PARITY: none
- STOP: 1bit
- FLOW CONTROL: none (**Critical**)

Make sure that the Flow Control is set to none.

13.3.4 Fourth Step

If everything checks out OK on the previous steps, then the issue may be on the BeagleBoard. Follow the steps below to determine that state of the BeagleBoard.

1. Apply power to the board.
2. LED D5 should come on indicating that power is on.
3. LEDs USR0 and USR1 will come on once the board runs UBOOT.
4. By this time data should be printed to the terminal window.
5. Below are a couple of scenarios we have seen:
 - o BeagleBoard sends data but cannot receive data
 - o No data is sent at all
 - o No data is sent, but it can be received.

If any of these issues are present, then there is a chance that the serial driver has failed. This is an issue with the level shifter, U9, on the board that we have seen fail after 48 hours of operation. The vast majority of boards with this issue are being screened out at the manufacturing stages, but some of the early shipment of boards could still exhibit this issue.

If this is the case, complete the RMA process at <http://beagleboard.org/support/rma>

14.0 Known Issues

This section provides information on any known issues with the BeagleBoard HW and the overall status. **Table 42** provides a list of the known issues on the BeagleBoard. The Rev C2 and C3 are provided for reference purposes.

Table 41. Known Issues

Affected Revision	Issue	Description	Workaround	Final Fix
C2 and C3	Random USB Host Disconnects	There is a small number of boards that are shown to have random disconnects when transferring large amounts of data via the EHCI USB Host port. This requires that the board be power cycled to restore the USB port. The issue is related to noise in a specific frequency range locking up the USB PHY. Some boards are noisier than others based on the current draw of the various components on the 1.8V rail.	Some boards can be fixed by placing a 22uf capacitor in parallel with C97. This fix does not work for all boards. Others have had success by adding the 22uF capacitor to the expansion header.	C4
C4	None	None		

15.0 PCB Component Locations

Figures 85 and **Figure 86** contain the bottom and top side component locations of the BeagleBoard.

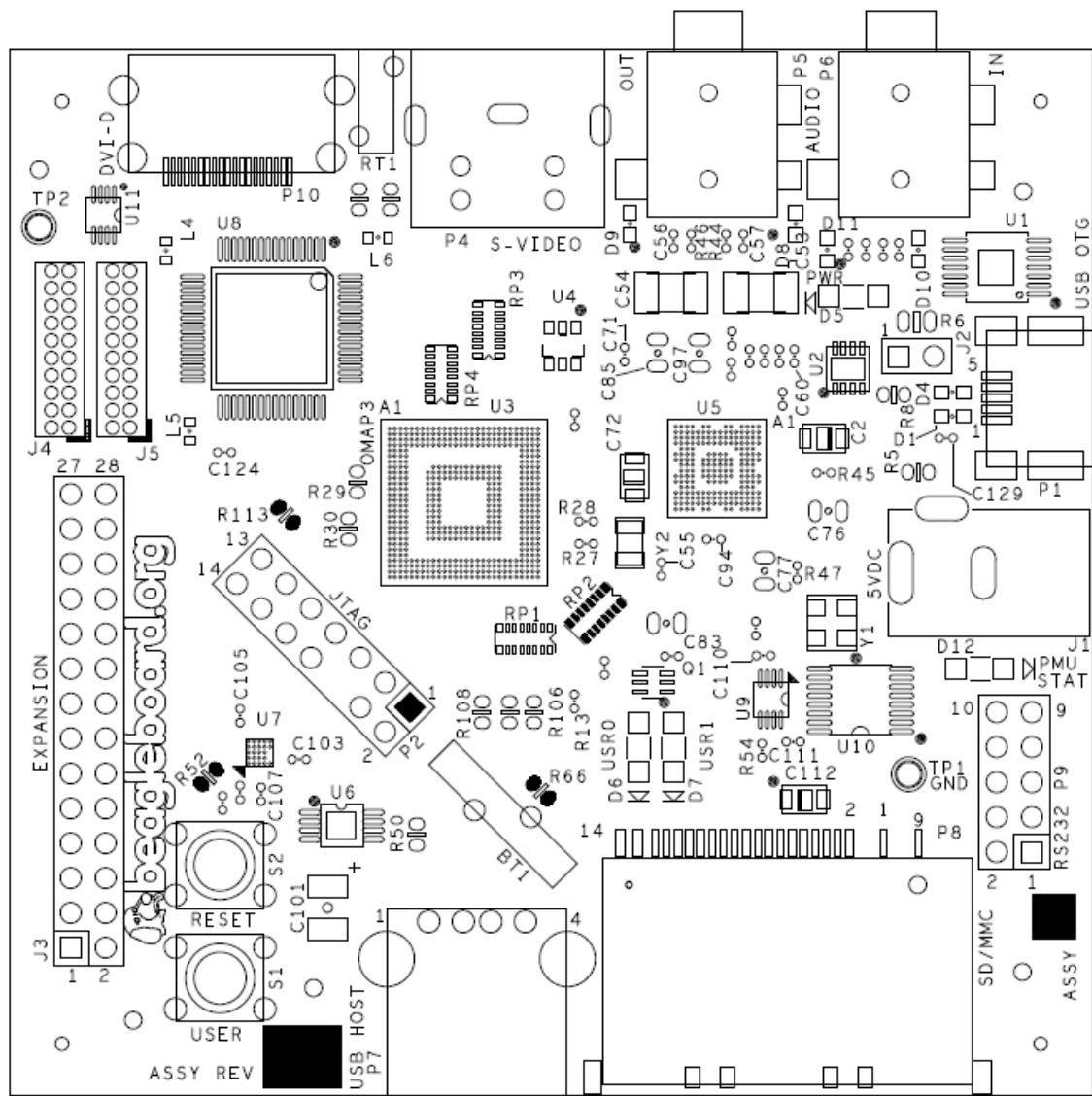


Figure 84. BeagleBoard Top Side Components

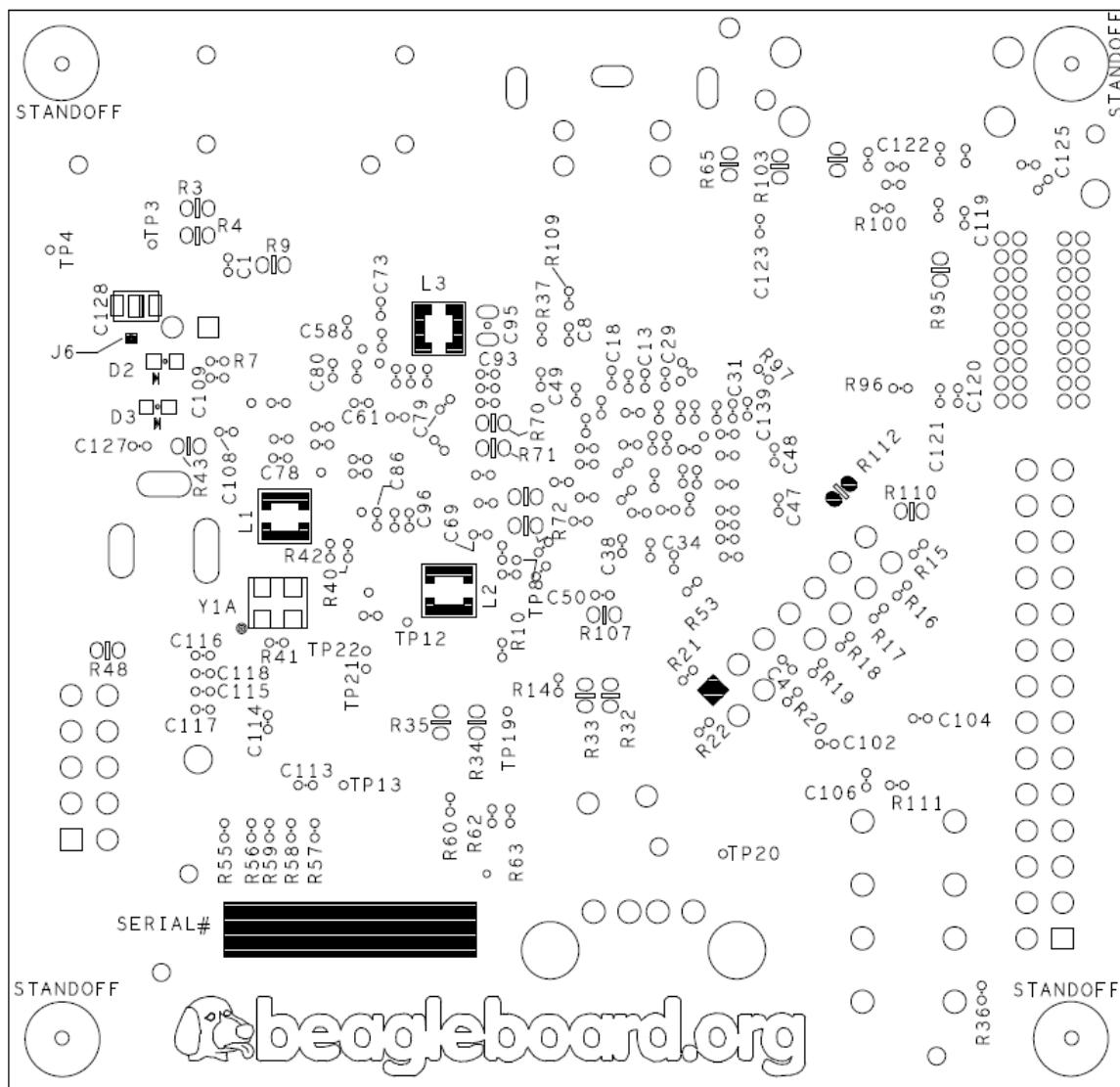


Figure 85. BeagleBoard Bottom Side Components

16.0 Schematics

The following pages contain the PDF schematics for the BeagleBoard. This manual will be periodically updated, but for the latest documentations be sure and check BeagleBoard.org for the latest schematics.

OrCAD source files are provided for BeagleBoard on BeagleBoard.org at the following link.

<http://beagleboard.org/hardware/design>

*These design materials are *NOT SUPPORTED* and DO NOT constitute a reference design. Only “community” support is allowed via resources at BeagleBoard.org/discuss. THERE IS NO WARRANTY FOR THE DESIGN MATERIALS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN MATERIALS “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN MATERIALS IS WITH YOU. SHOULD THE DESIGN MATERIALS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.*

We mean it, these design materials may be totally unsuitable for any purposes.

REV	Description	DATE	BY	3	2	1
C	1. Improved layout for the USB PHY. 2. Removed unused parts from the design. 3. Added current measurement function to the TWL4030. 4. Added filter caps to the VBUS rail input and output. 5. Changed U9 & U11 package to the QFN.	8/14/08	GC			
D	1. Added J12 and J13 to provide access to the RGB TTL signals on the LCD. 2. Added 5 filter caps. 3. Moved the USB Host port from Port1 to Port2. 4. Deleted R1. 5. Added 10K pulldown to USB reset signal. 6. Added 10K pulldown resistors as ID function to determine board type by reading these pins. 7. Added series resistor, R53, in the CLK line of the HSUSB clock line. May be removed after testing.	10/1/08	GC			
C1	1. Moved the McBSP3_DX signal to pin AB26. 2. Moved the McBSP3_DR signal to pin AB25. 3. Moved the McBSP3_CLKX signal to pin AD25. 4. Changes were to allow access to three PWM signals from OMAP3530.	12/16/08	GC			
C2	1. Added series resistor, BKBAT. 2. Added TP to BKBAT to allow access for battery. 3. Added a 4.7μF CAP and 3.3uH inductor to the S-Video feedback resistors.	2/11/2009	GC			
C3	1. Switched to TPS65960 based on the availability of the parts. 2. Made the battery an installed component. Removed parallel resistor.	4/21/2009	GC			
C3A	1. Corrected J4 and J5 symbol for the RGB interface. No electrical changes were made. 2. Removed battery as an installed component due to availability issues.	4/30/2009	GC			
C3B	1. Added C141, 22uF in parallel with C9. 2. Added option to allow the USB PHY and CLKOUT to be powered from the VIO_1V8 rail or the VAUX2 rail from the TPS65960 Default is VIO_1V8 rail. 3. Changed 1.8V filter CAP on USB PHY to 22uF. 4. Made R113 a DNL and installed R112.	10/5/2009	GC			
C4	1. Made R67 an installed inductor and made R68 a DNL. Switched to LDO powered EHCI USB Phy.	11/5/2009	GC			
B-C4						

CONTENTS

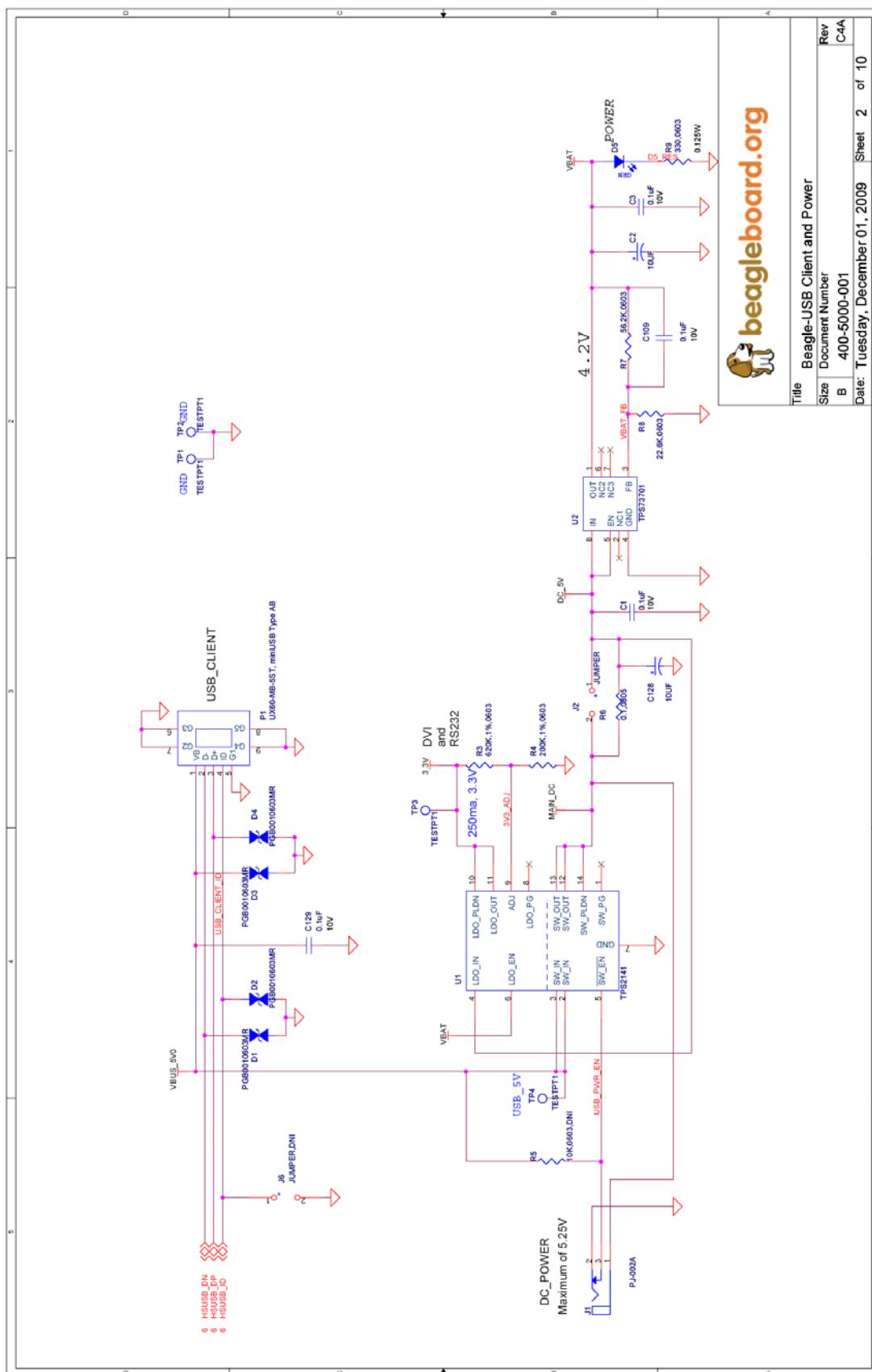
PAGE NO.	SCHEMATIC PAGE
1	COVER PAGE
2	USB OTG CONNECTOR AND MAIN POWER
3	OMAP3 1 OF 3
4	OMAP3 2 OF 3, JTAG, SWITCHES, LEDs, SVIDEO
5	OMAP3 3 OF 3
6	TPS65950 1 of 2, AUDIO JACKS, LED, 26MHz, 32KHz
7	TPS65950 2 of 2, Power Rails
8	USB HOST AND EXPANSION
9	SD/MMC, SERIAL HEADER
10	DVI-D

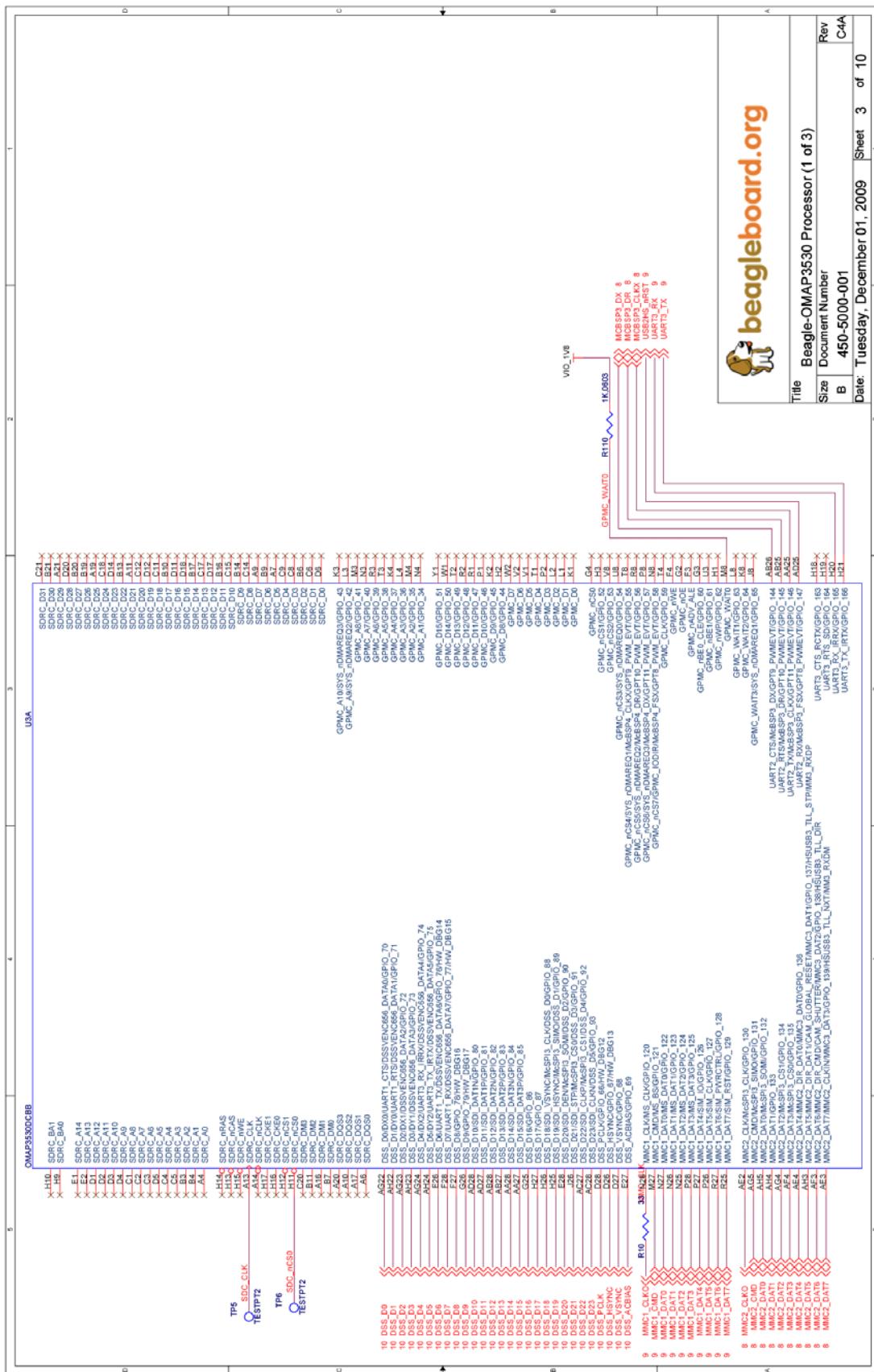
This schematic is *NOT SUPPORTED* and DOES NOT constitute a reference design. Only "community" support is allowed via resources at BeagleBoard.org/discuss.

THERE IS NO WARRANTY FOR THIS DESIGN, TO THE EXTENT PERMITTED BY APPLICABLE LAW, EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN IS WITH YOU. SHOULD THE DESIGN PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

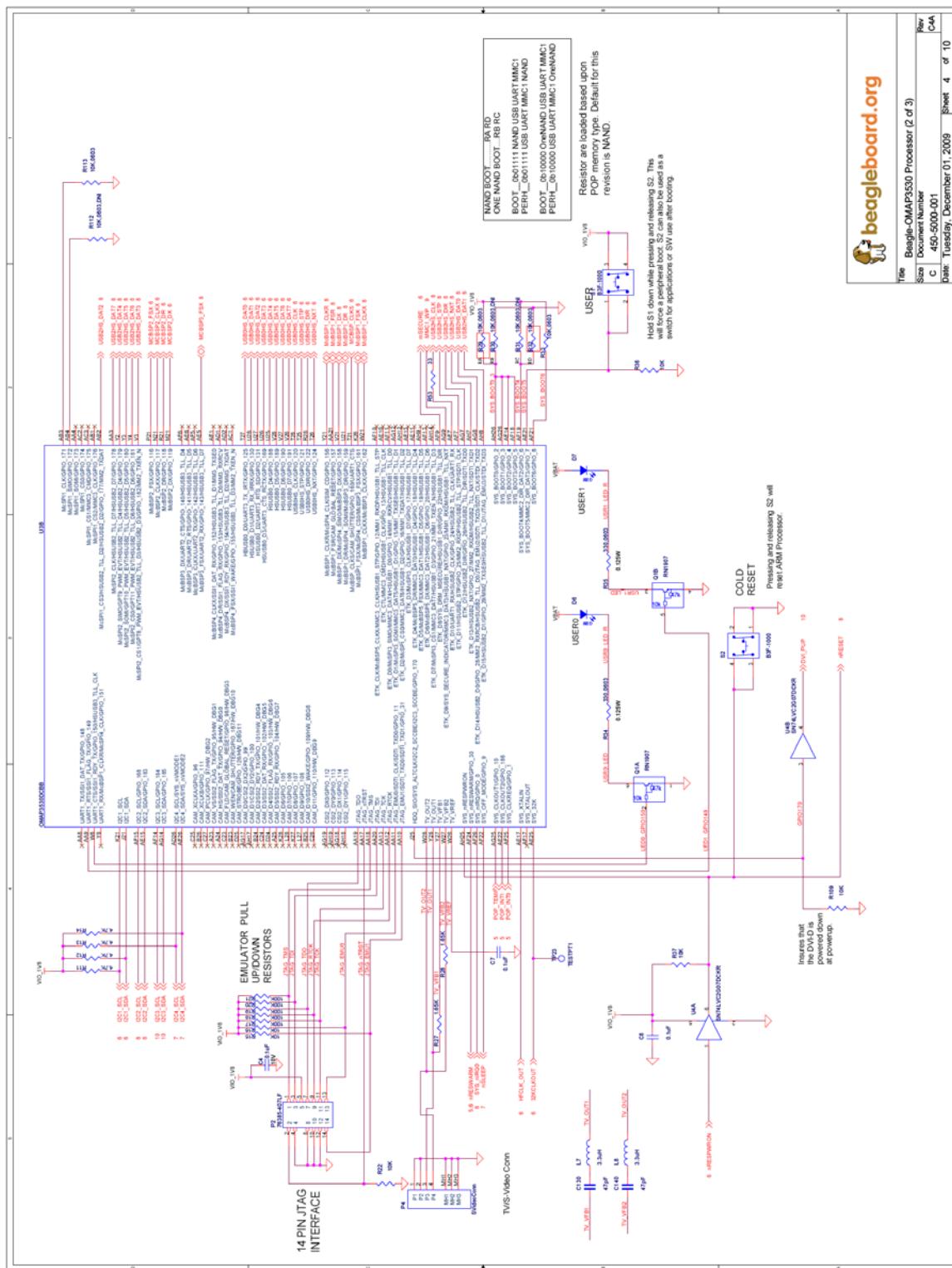


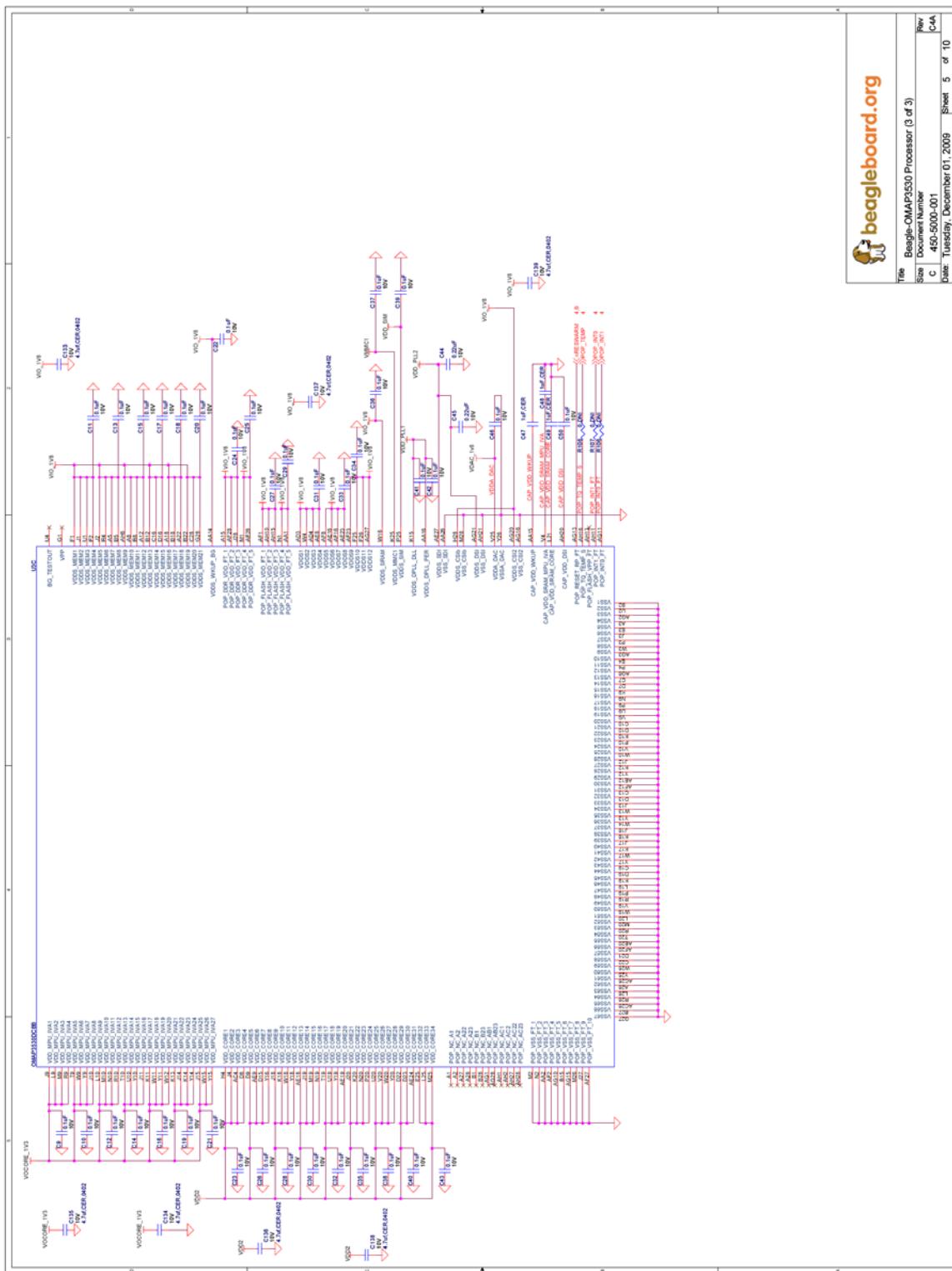
Title	Beagle-Cover Page
Size	Document Number
B	450-5000-001
Date:	Tuesday, December 01, 2009
Rev	C4A
Sheet	1 of 10

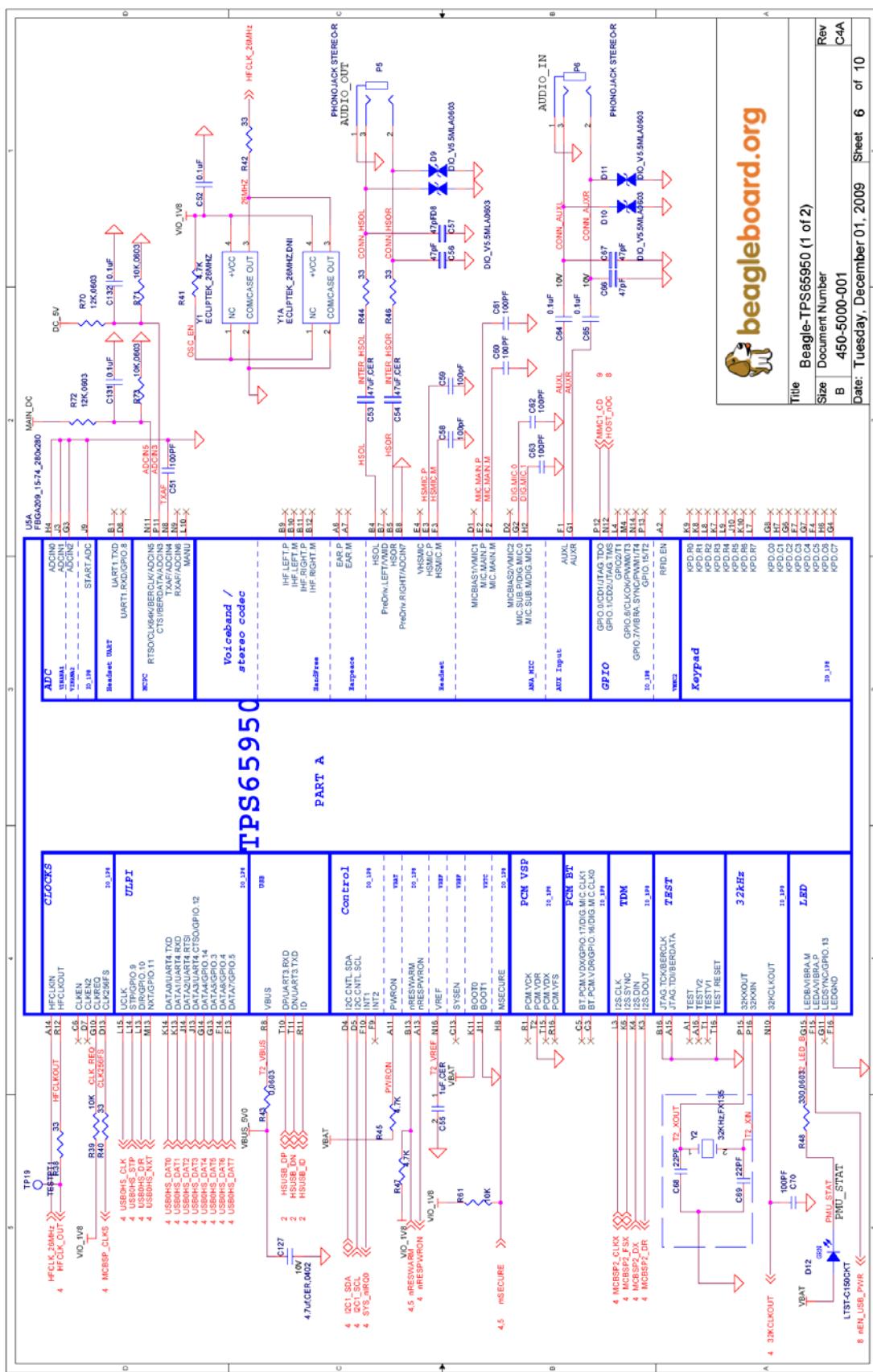


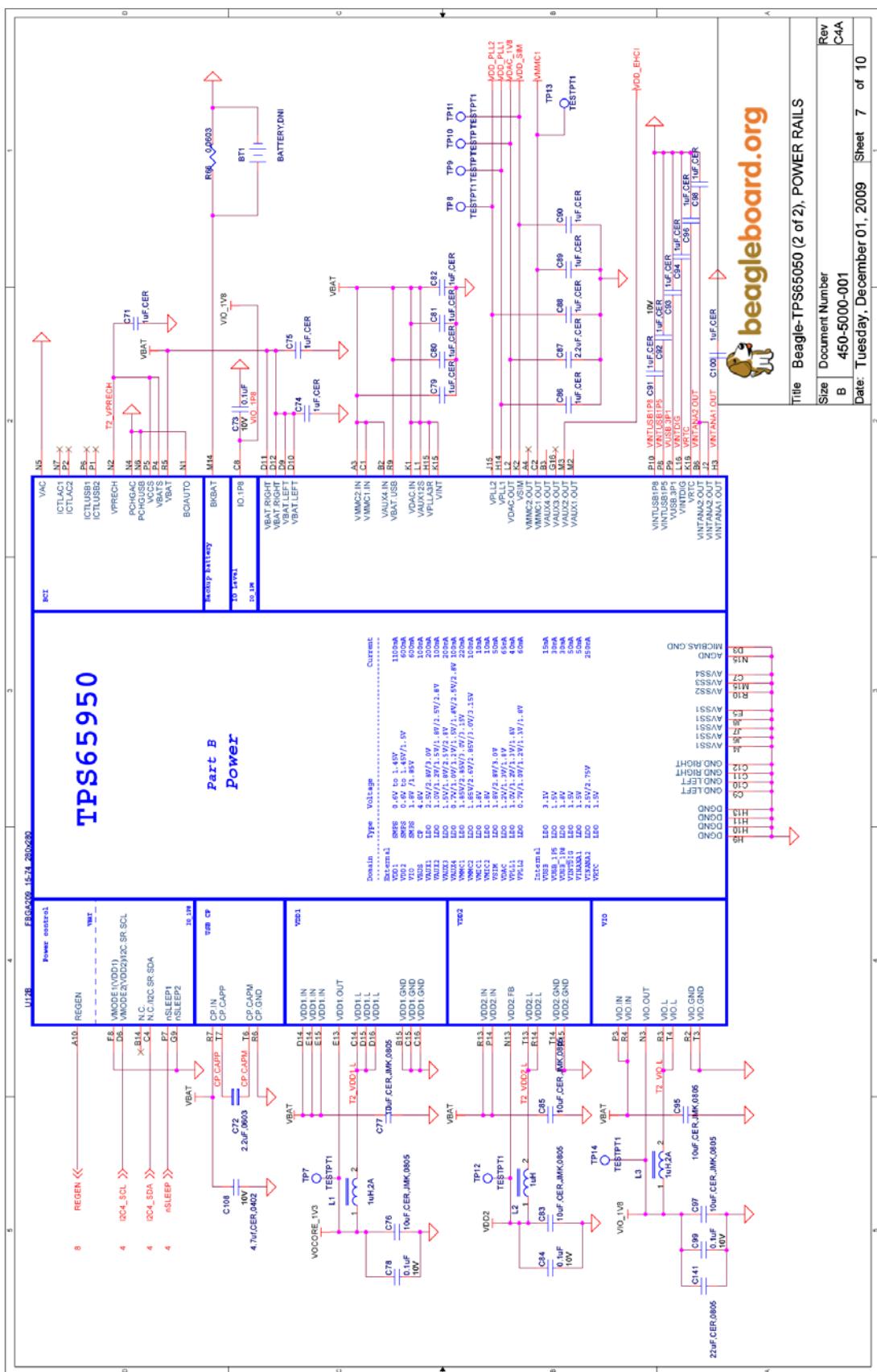


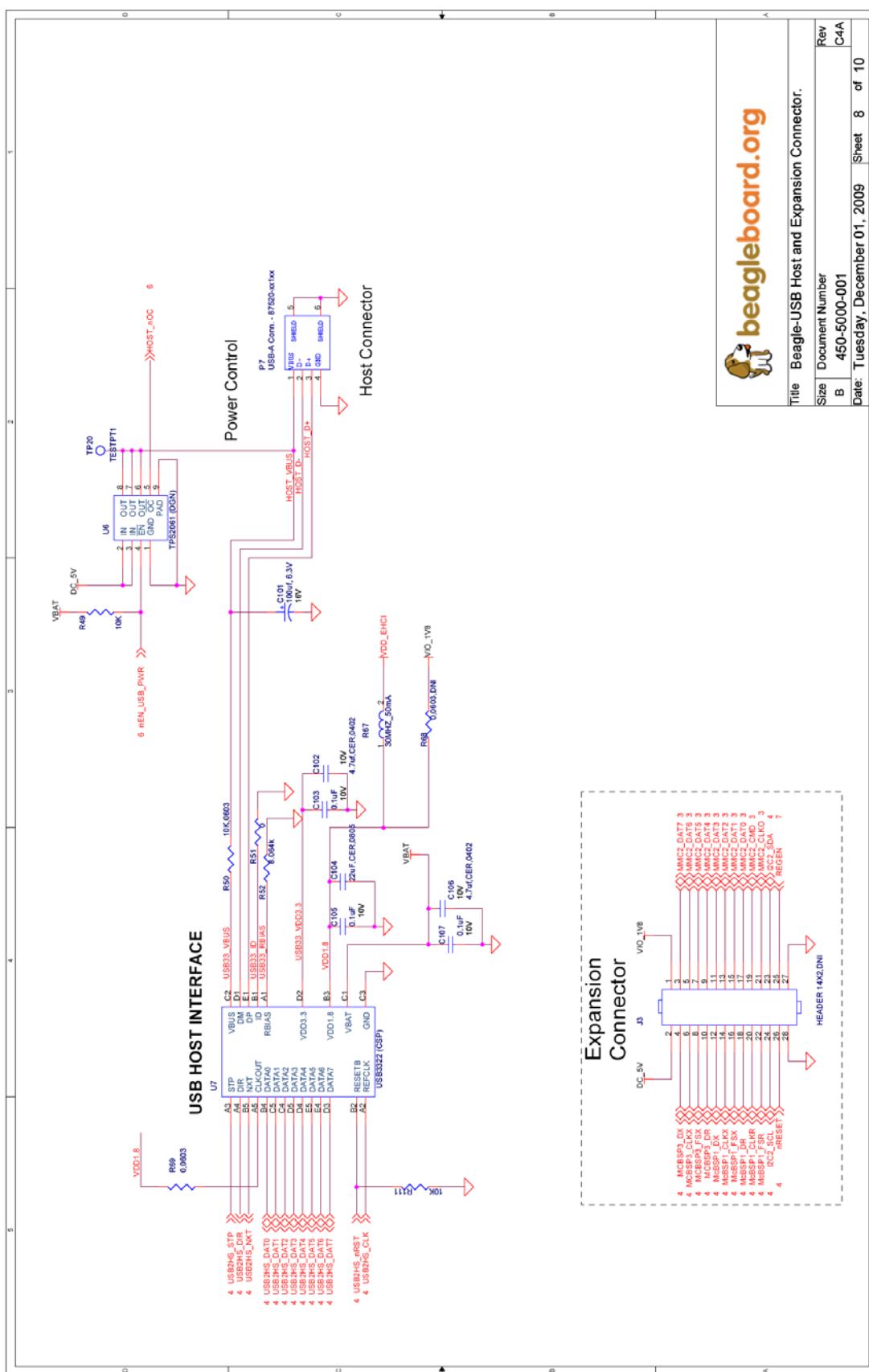
beagleboard.org
Title: Beagle-OMAP3530 Processor (1 of 3)
Size: Document Number: B
Rev: 450-5000-001
Date: Tuesday, December 01, 2009
Sheet: 3 of 10

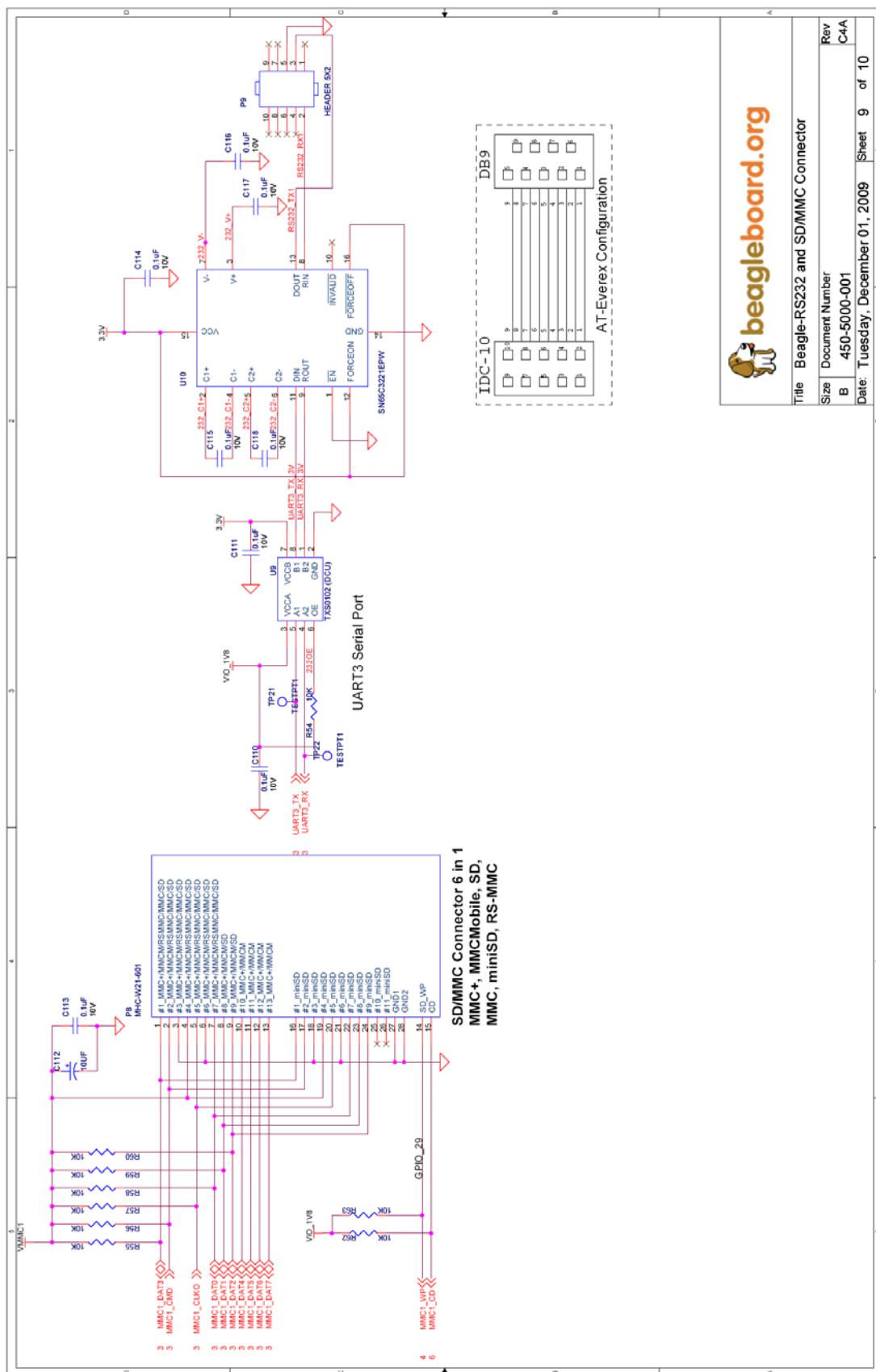


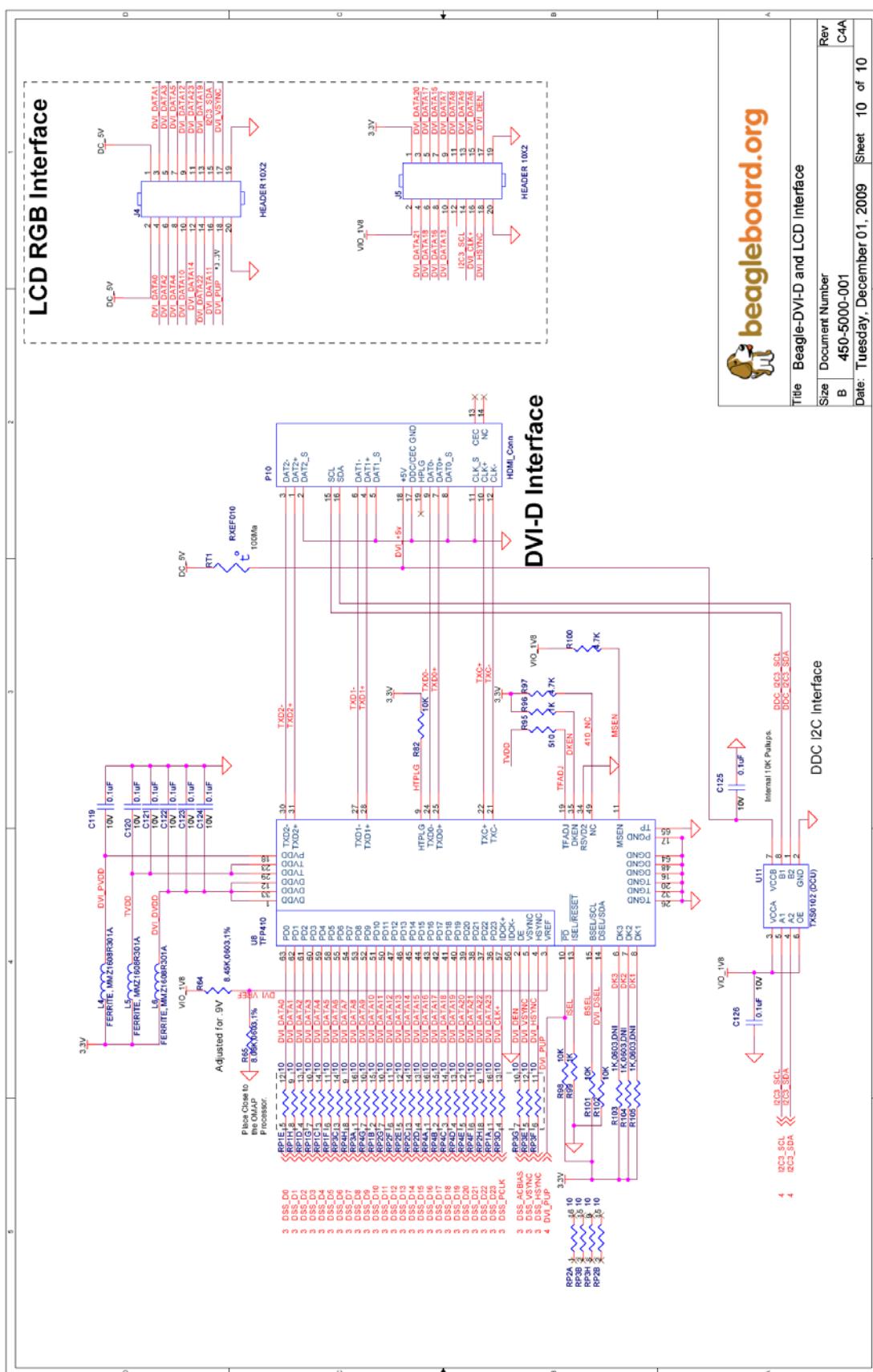












17.0 Bills of Material

The Bill of Material for the Beagle Board is provided at BeagleBoard.org at the following location:

<http://beagleboard.org/hardware/design>

*These design materials are *NOT SUPPORTED* and DO NOT constitute a reference design. Only “community” support is allowed via resources at BeagleBoard.org/discuss. THERE IS NO WARRANTY FOR THE DESIGN MATERIALS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN MATERIALS “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN MATERIALS IS WITH YOU. SHOULD THE DESIGN MATERIALS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.*

We mean it; these design materials may be totally unsuitable for any purposes.

18.0 PCB Information

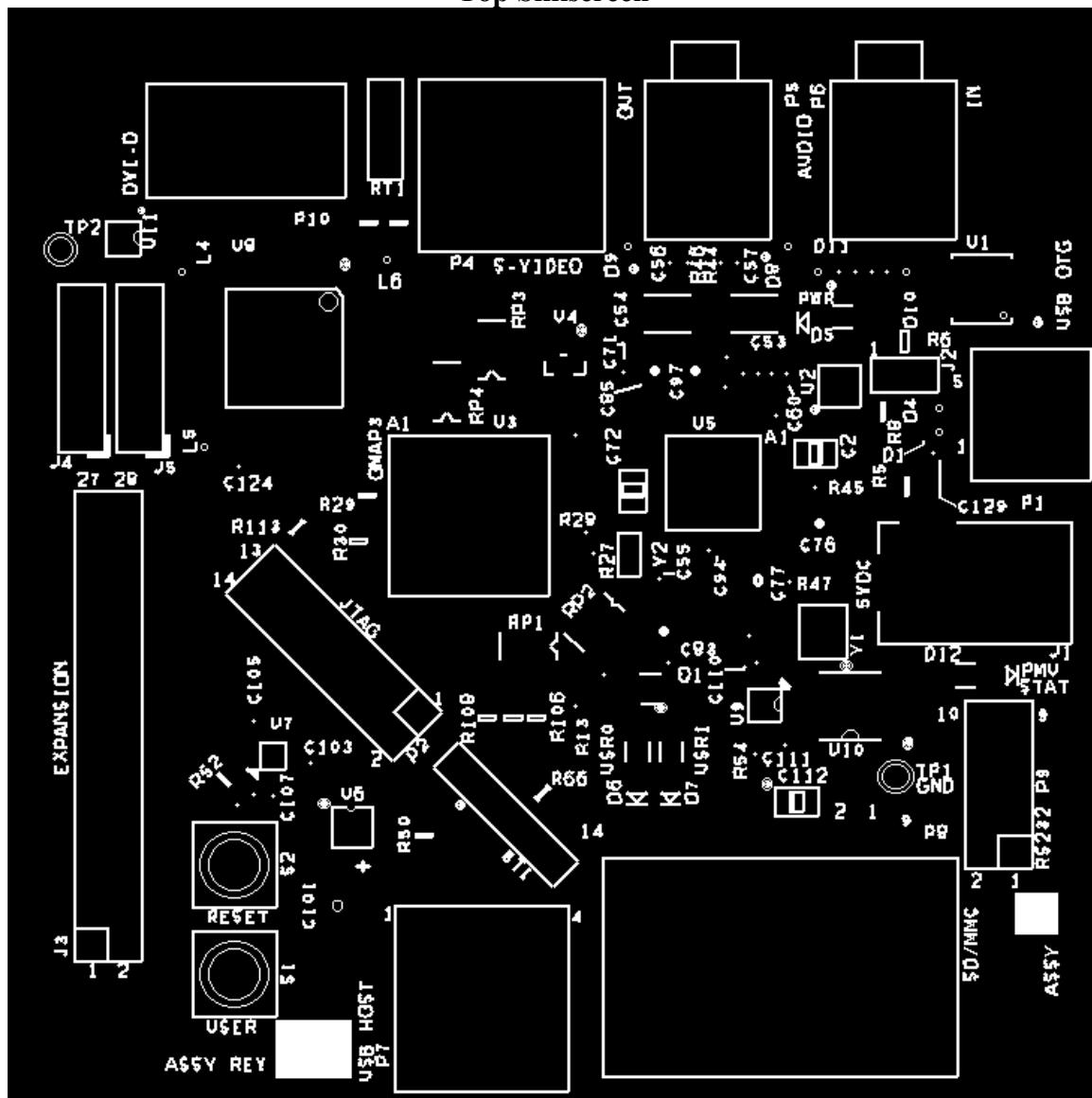
The following pages contain the PDF PCB layers for the BeagleBoard. Gerber files and Allegro source files are available on BeagleBoard.org at the following address.

<http://beagleboard.org/hardware/design>

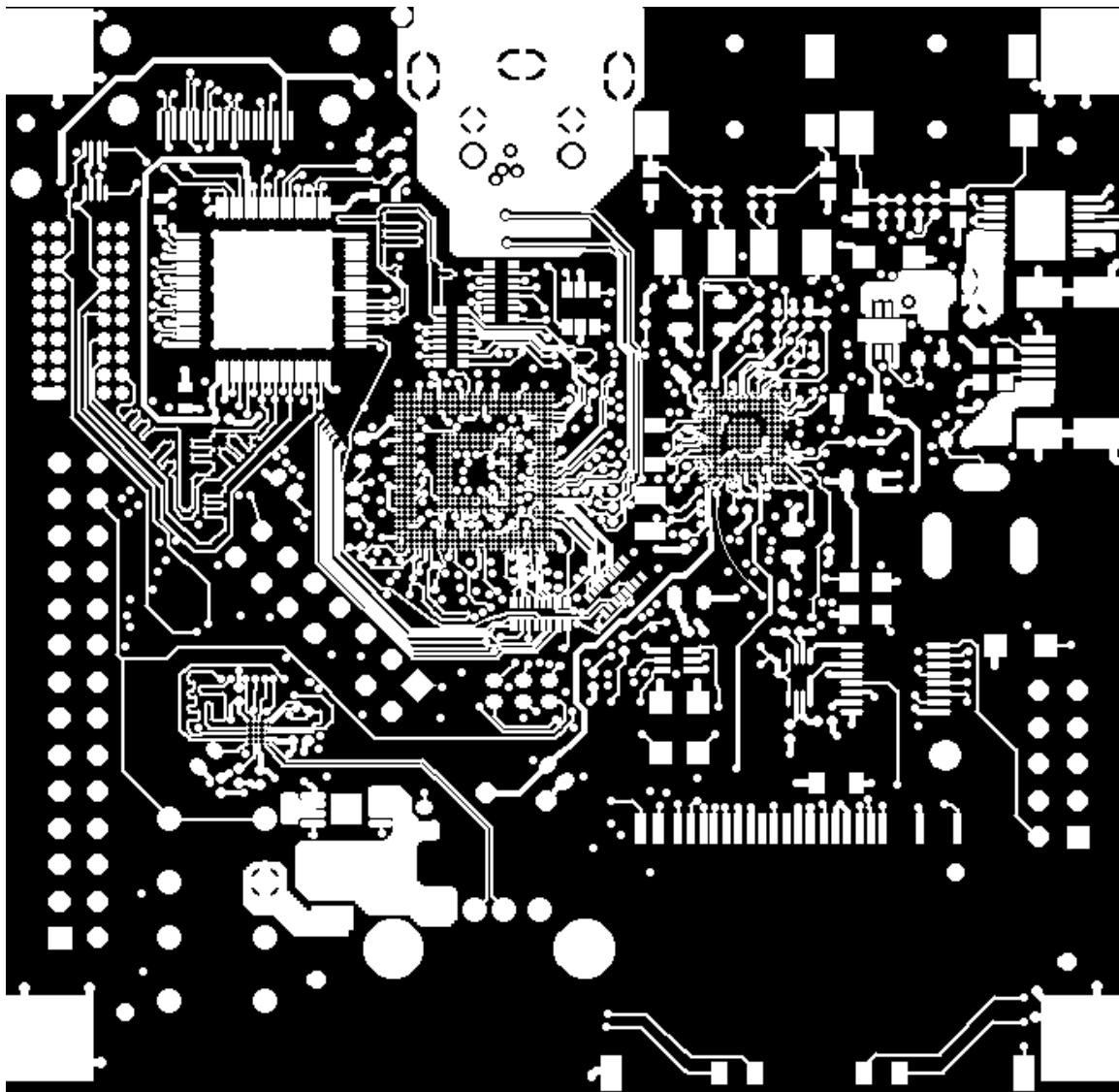
*These design materials are *NOT SUPPORTED* and DO NOT constitute a reference design. Only “community” support is allowed via resources at BeagleBoard.org/discuss. THERE IS NO WARRANTY FOR THE DESIGN MATERIALS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN MATERIALS “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN MATERIALS IS WITH YOU. SHOULD THE DESIGN MATERIALS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.*

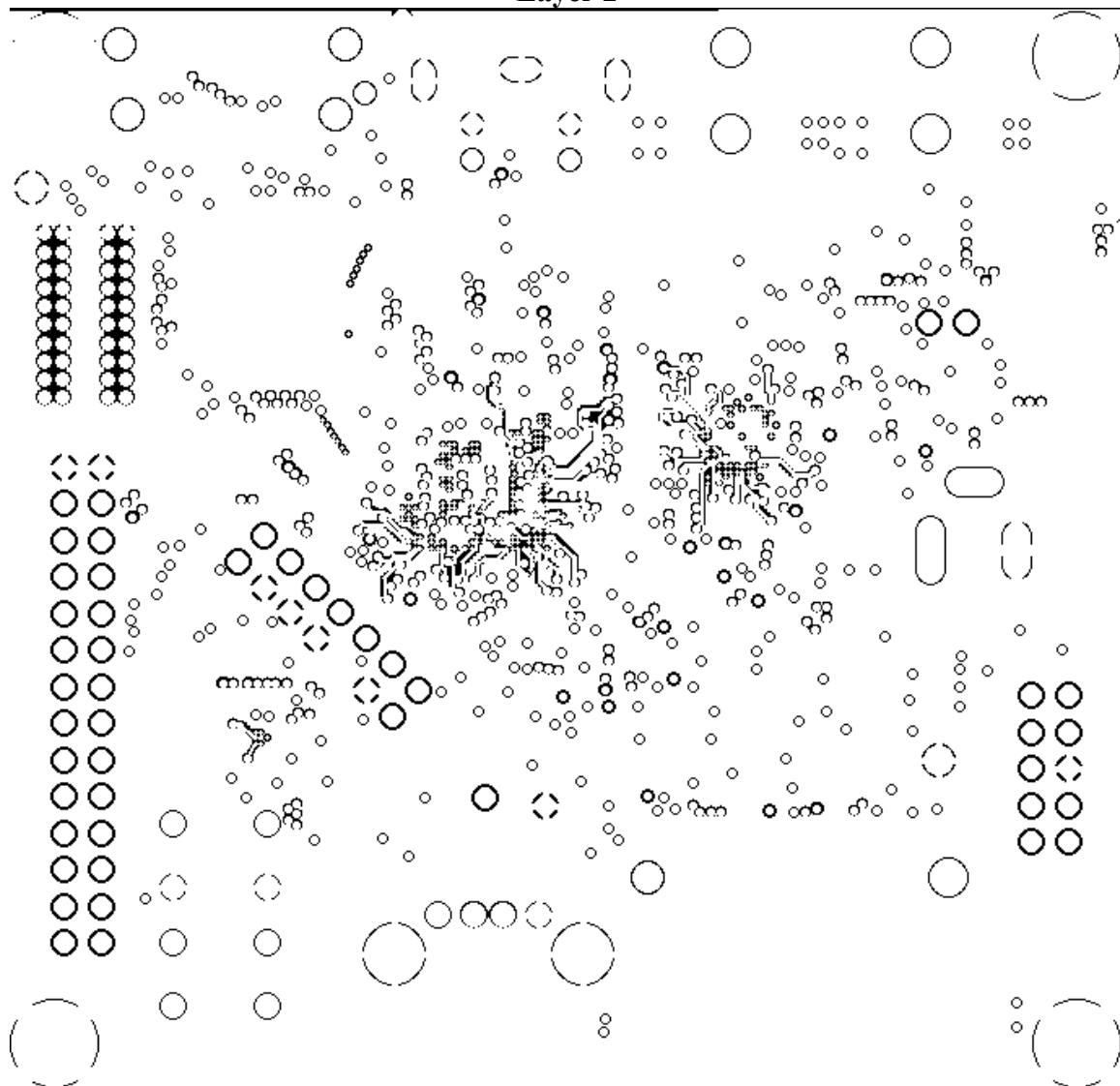
We mean it; these design materials may be totally unsuitable for any purposes.

Top Silkscreen

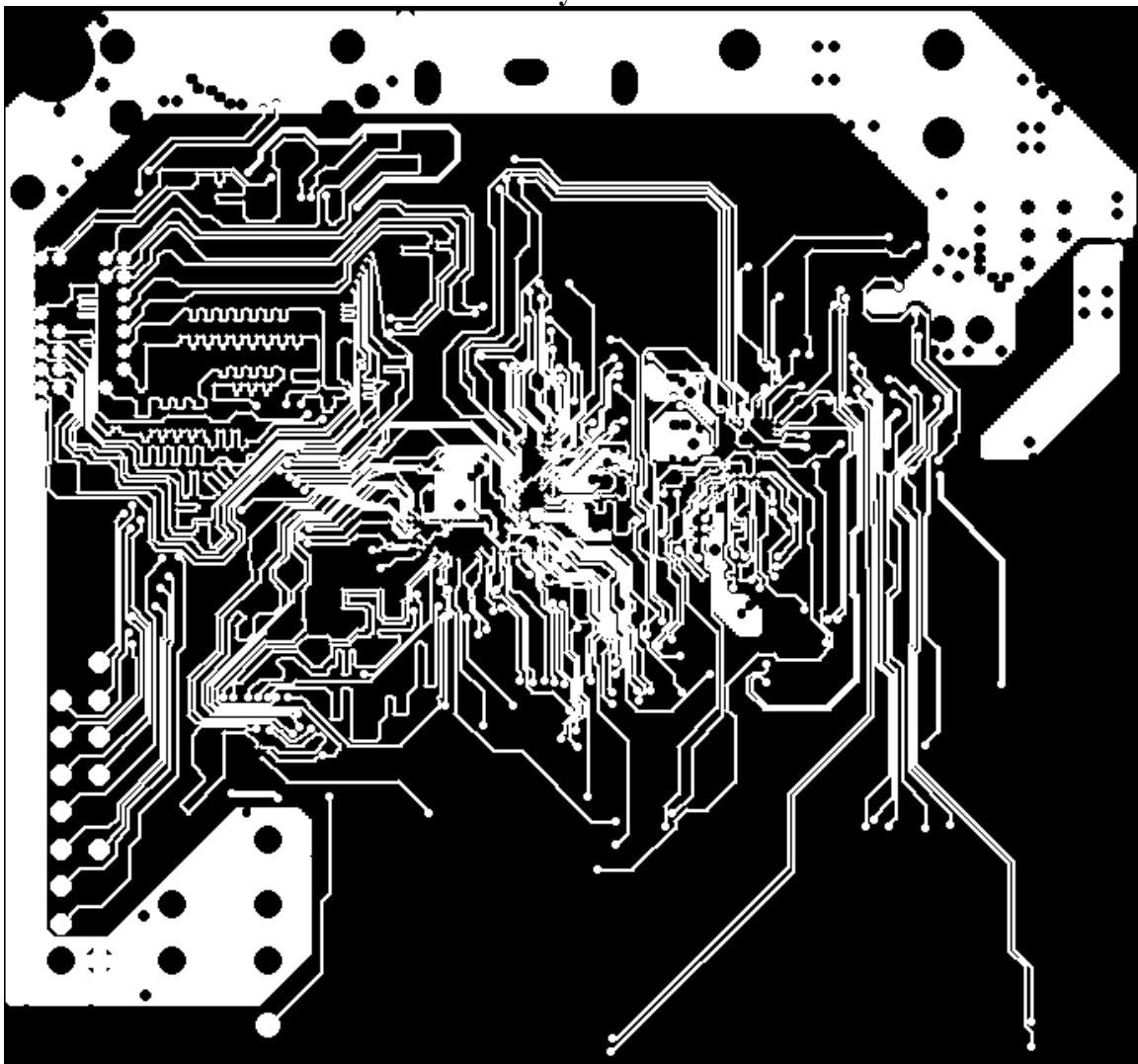


Layer 1

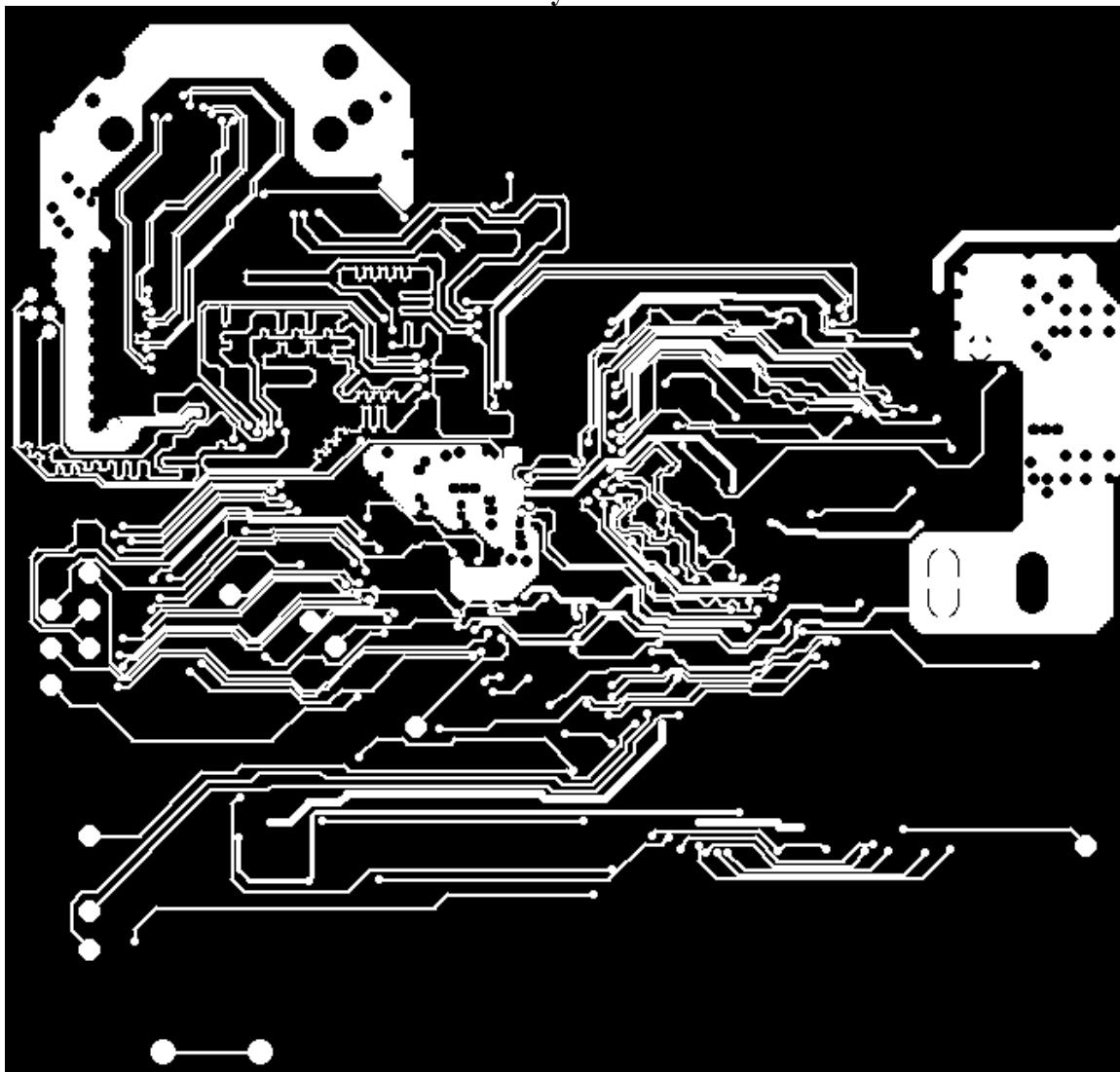


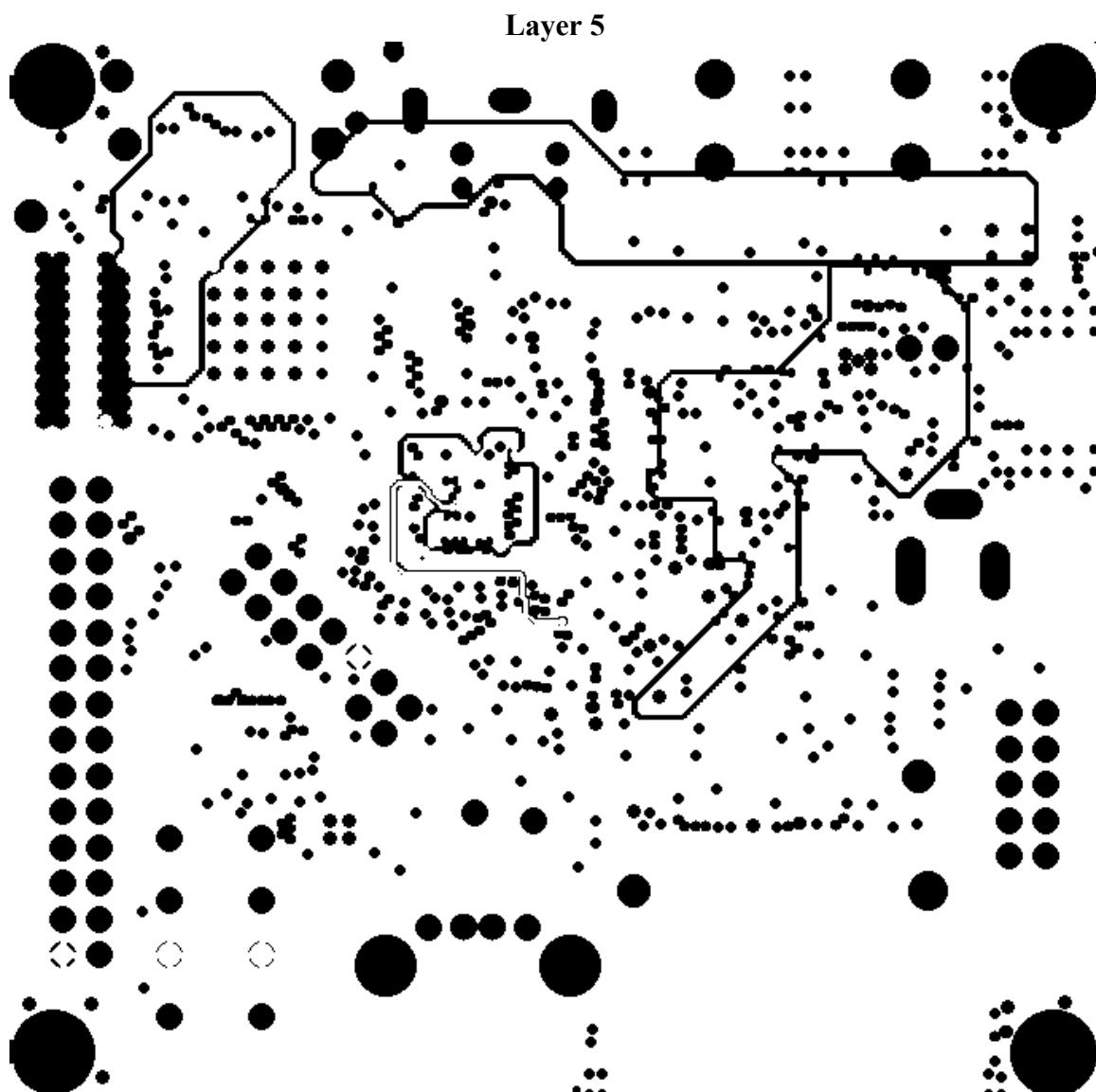
Layer 2

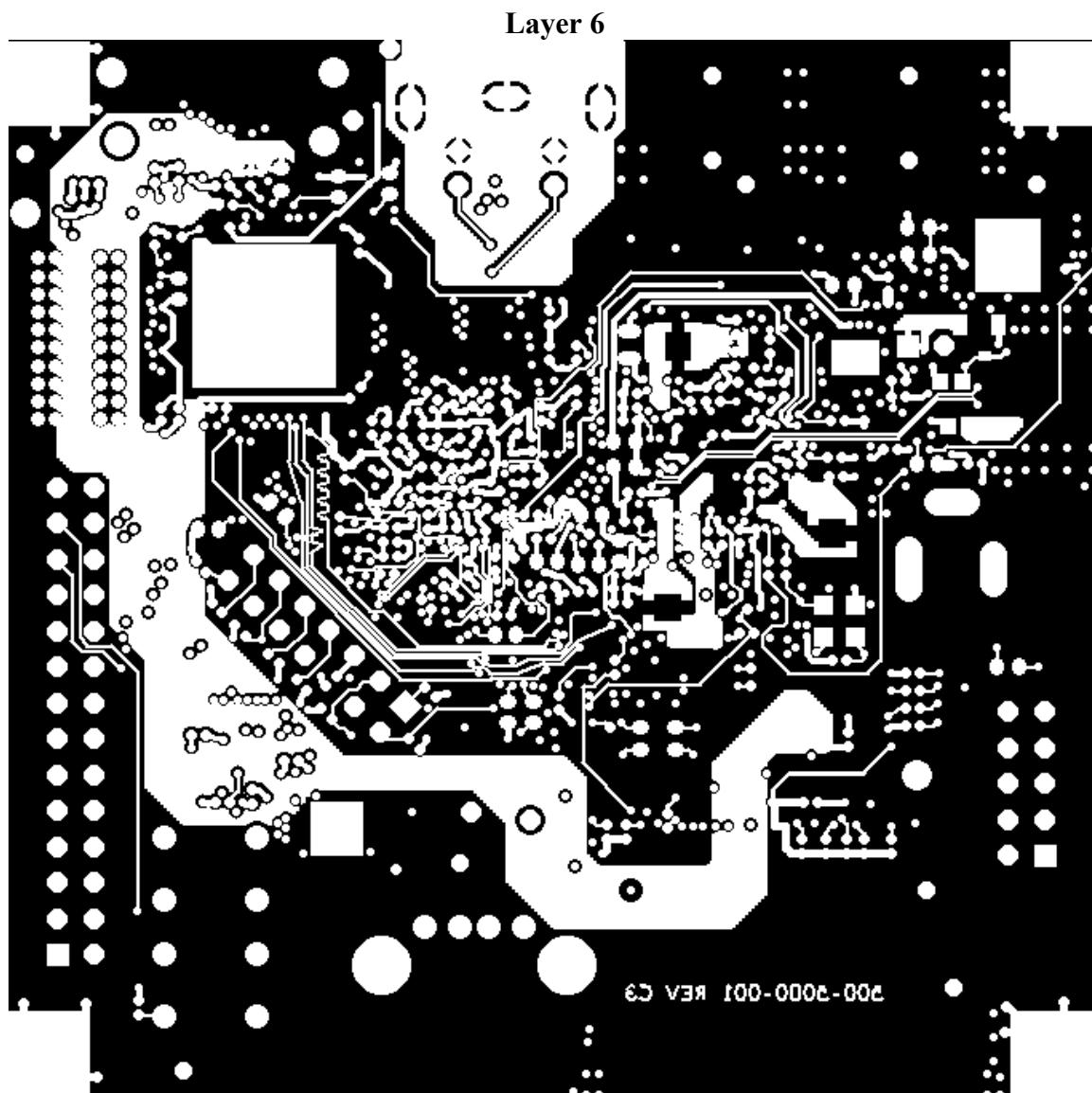
Layer 3



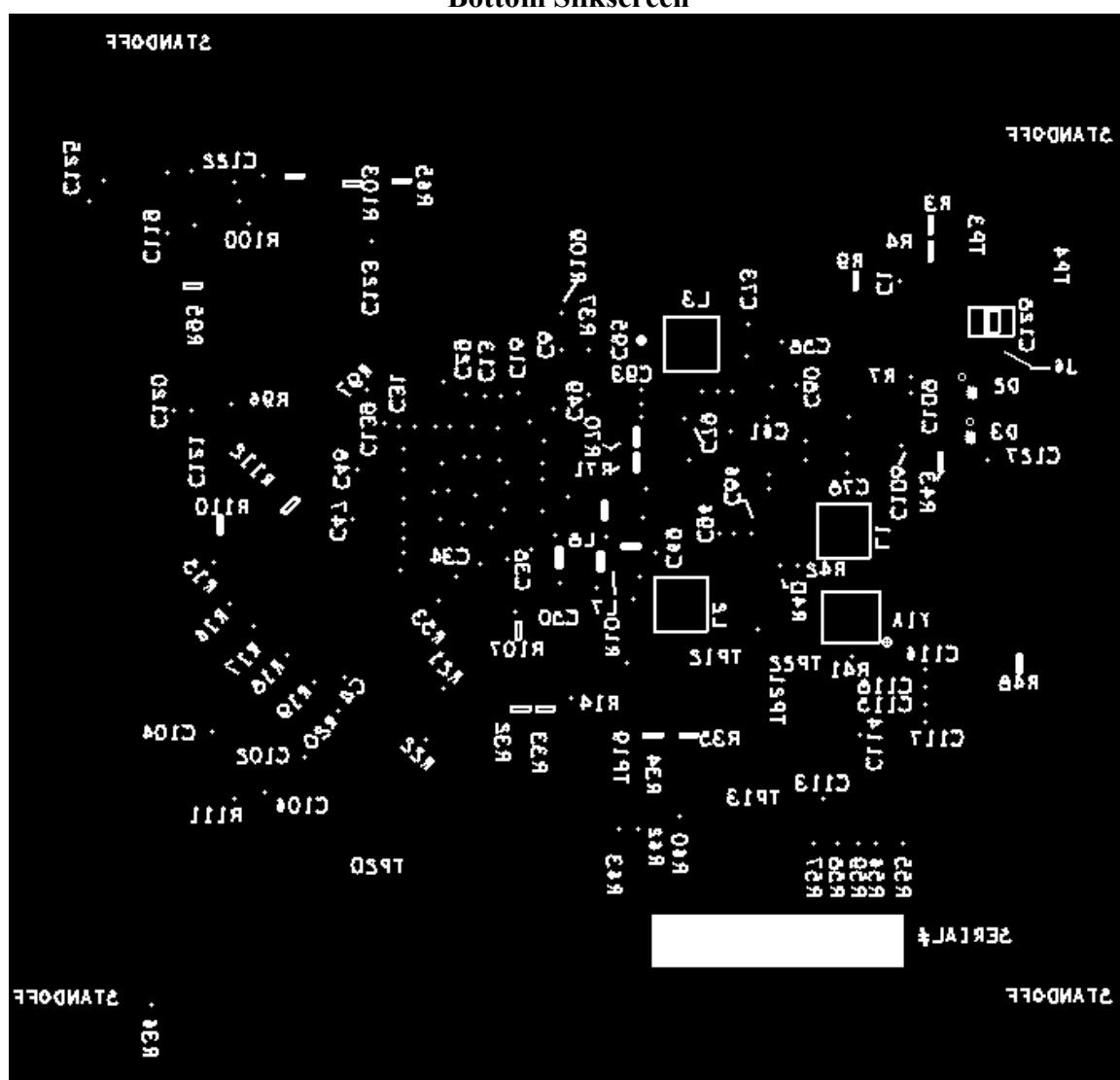
Layer 4







Bottom Silkscreen



7.4.2 SCM Initialization

The SCM responds only to the internal power-on reset and to the device type. At power-on, reset values for the registers define the safe state for the device. In the initialization mode, only modules used at boot time are associated with the pads. Other module inputs are internally tied, and outputs pads are turned off each time the feature is available.

For the pad configuration, pull-up/pull-down fields are set according to the device pin list.

General-purpose devices include features that are inaccessible or unavailable. These inaccessible registers define the default or fixed device configuration or behavior.

7.4.3 Wake-up Control Module

The wake-up control module in the SCM belongs to the WKUP power domain. It contains a 1K-byte memory in which to save the pad configuration registers in the core control module before going to off mode. Pad configuration registers driving the I/O pad control in the WKUP power domain are also instantiated in the wake-up control module.

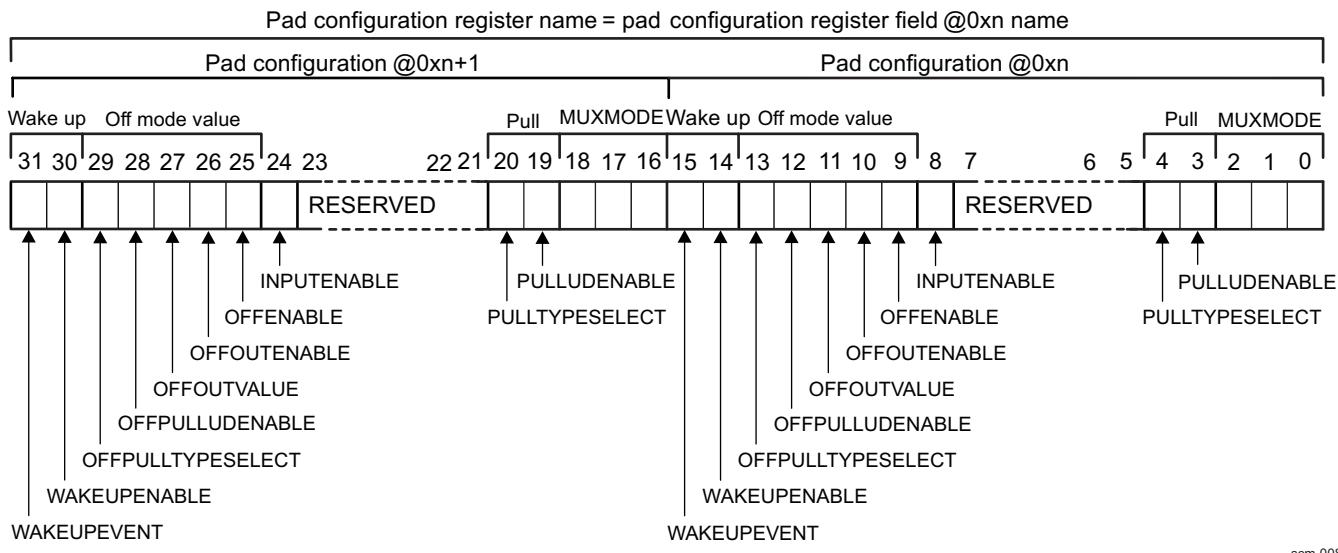
The wake-up control module is configured through the L4-Core interface in the core control module and is accessed from the core control module through a dedicated interface. This interface between the core control module and the wake-up control module uses the CORE_L4_ICLK clock divided by 4 or 2 to reduce power consumption in the WKUP power domain.

7.4.4 Pad Functional Multiplexing and Configuration

After power-on reset, the software sets the pad functional multiplexing and configuration registers to the requested device pad configurations. Data written in these registers command directly the multiplexing of the pad configuration logic.

Each pin is configurable by software using its associated pad configuration register field, which is 16 bits wide (see Figure 7-7).

Figure 7-7. Pad Configuration Register Functionality



One pad configuration register field is available for each pin. Each 32-bit pad configuration register is grouped into two 16-bit pad configuration register fields. One pad configuration register provides control for two different pins.

Some pad configuration registers control the configuration of pads in the CORE power domain. These registers are instantiated in the CORE power domain of the SCM (core control module, physical addresses 0x4800 2030 to 0x4800 2260). Pad configuration registers also control the configuration of pads in the WKUP power domain. These registers are instantiated in the WKUP power domain of the SCM (wake-up control module, physical addresses 0x4800 2A00 to 0x4800 2A4C).

NOTE: These registers can be accessed using 8-, 16-, and 32-bit operations.

The functional bits of a pad configuration register field are divided into the following five fields:

- MUXMODE (3 bits) defines the multiplexing mode applied to the pin. A mode corresponds to the selection of the functionality mapped on the pin with six (0 to 5) possible functional modes for each pin.
- PULL (2 bits) for combinational pullup/pulldown configuration:
 - PULLTYPESELECT: Pullup/pulldown selection for the pin.
 - PULLUDENABLE: Pullup/pulldown enable for the pin.
- INPUTENABLE (1 bit) drives an input enable signal to the I/O CTRL.
 - INPUTENABLE = 0: Input Disable. Pin is configured in output only mode.
 - INPUTENABLE = 1: Input Enable. Pin is configured in bidirectional mode.
- Off mode values (5 bits) override the pin state when the OFFENABLE bit CONTROL.
CONTROL_PADCONF_X is set and off mode is active. This feature allows having separate configurations for the pins when in off mode:
 - OFFENABLE: Off mode pin state override control. Set to 1 to enable the feature and to 0 to disable it.
 - OFFOUTENABLE: Off mode output enable value. Set to 0 to enable the feature and to 1 to disable it.
 - OFFOUTVALUE: Off mode output value.
 - OFFPULLUDENABLE: Off mode pullup/pulldown enable
 - OFFPULLTYPESELECT: Off mode pullup/pulldown selection

CAUTION

The OFFOUTENABLE and OFFOUTVALUE bits are functional only if the pad configuration supports output mode on at least one MUXMODE. For a pad that supports only the input feature, the OFFOUTENABLE and OFFOUTVALUE bits cannot be configured (they are don't care and read always returns 0).

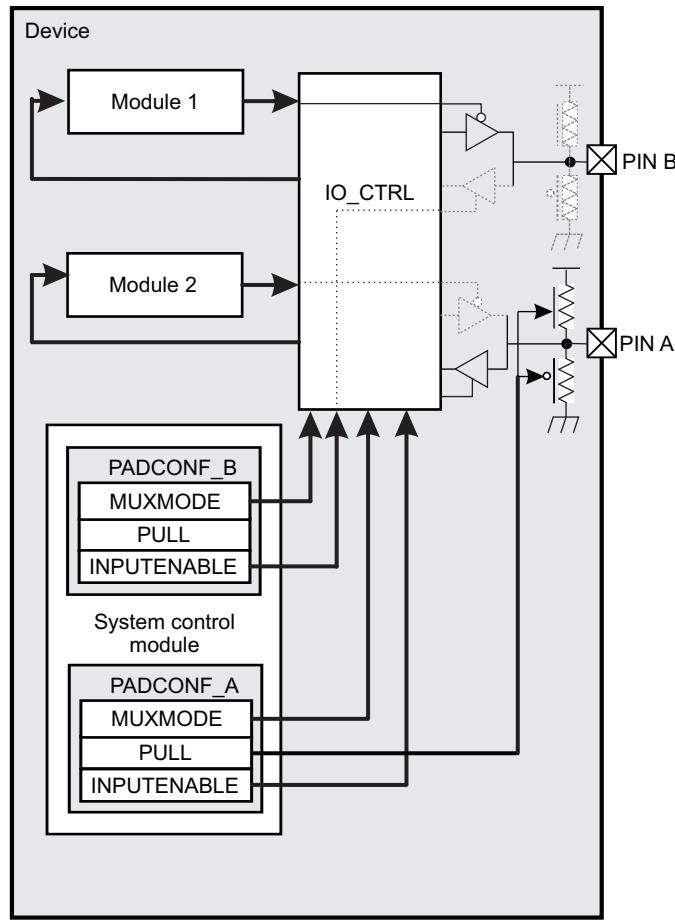
- Wake-up bits (2 bits):
 - WAKEUPENABLE: Enable wake-up detection on input. It is also the off mode input enable value.
 - WAKEUPEVENT: Wake-up event status for the pin.

CAUTION

The software must configure the OFF mode pads. It must ensure that the input/output capability is enabled for each pin.

Figure 7-8 shows the pad configuration functionality when off mode is inactive. For more information about off mode, see [Section 7.4.4.4, Off Mode](#).

Figure 7-8. Pad Configuration Diagram



scm-009

7.4.4.1 Mode Selection

Table 7-2 lists the multiplexing modes and settings.

Table 7-2. Mode Selection

MUXMODE	Selected Mode
0b000	Primary mode = Mode 0
0b001	Mode 1
0b010	Mode 2
0b011	Mode 3
0b100	Mode 4
0b101	Mode 5
0b110	Mode 6
0b111	Safe mode = Mode 7

The MUXMODE field CONTROL.[CONTROL_PADCONF_X](#) defines the multiplexing mode applied to the pad. Modes are referred to by their decimal (from 0 to 7) or binary (from 0b000 to 0b111) representation. Functional modes are defined from 0b000 to 0b101; mode 0b111 is referred to as the safe mode.

For most pads, the reset value for the MUXMODE field CONTROL.[CONTROL_PADCONF_X](#) is 0b111. The exceptions are pads to be used at boot time to transfer data from selected peripherals to the external flash memory.

Mode 0 is the primary mode. When mode 0 is set, the function mapped to the pin corresponds to the name of the pin.

Mode 1 to mode 6 are possible modes for alternate functions. On each pin, some modes are used effectively for alternate functions, while other modes are unused and correspond to no functional configuration.

The safe mode avoids any risk of electrical contention by configuring the pin as an input with no functional interface mapped to it. The safe mode is used mainly as the default mode for all pins containing no mandatory interface at the release of power-on reset.

For more information about the configurable mode on each pin, see [Table 7-4](#) through [Table 7-6](#).

7.4.4.2 Pull Selection

Whichever pull value is configured, pulls are automatically disabled when a pin is configured as an output (see [Table 7-3](#)).

Table 7-3. Pull Selection

PULL		Pin Behavior
PULLTYPESELECT	PULLUDENABLE	
0b0	0b0	Pull-down selected but not activated
0b0	0b1	Pull-down selected and activated if pin is NOT configured as OUTPUT
0b1	0b0	Pull-up selected but not activated
0b1	0b1	Pull-up selected and activated if pin is NOT configured as OUTPUT

For more information on the pull available on each pin, see [Table 7-4](#) through [Table 7-6](#).

7.4.4.3 Pad Multiplexing Register Fields

Table 7-4 through **Table 7-6** provide for each pad configuration register field the address offset, reset values, and associated signal name for each multiplexing mode (as set by bit field MUXMODE). Mode 0 is always defined. Modes with no signal name are undefined for the given pad.

NOTE:

- Pad configuration registers are split into three types, which correspond to the following three tables:
 - **Table 7-4** lists the pad configuration registers instantiated in the CORE power domain that drive the pads in the CORE power domain.
 - **Table 7-5** lists the pad configuration registers instantiated in the CORE power domain that drive the D2D pads. These pads are available in stacked mode only.
 - **Table 7-6** lists the pad configuration registers instantiated in the WKUP power domain that drive the pads in the WKUP power domain.
- In **Table 7-4** through **Table 7-6**, an empty cell indicates that the mode or pull is not available for this pin.

Table 7-4. Core Control Module Pad Configuration Register Fields

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_SDRC_D0[15:0]	0x4800 2030	sdrc_d0							
CONTROL_PADCONF_SDRC_D0[31:16]	0x4800 2030	sdrc_d1							
CONTROL_PADCONF_SDRC_D2[15:0]	0x4800 2034	sdrc_d2							
CONTROL_PADCONF_SDRC_D2[31:16]	0x4800 2034	sdrc_d3							
CONTROL_PADCONF_SDRC_D4[15:0]	0x4800 2038	sdrc_d4							
CONTROL_PADCONF_SDRC_D4[31:16]	0x4800 2038	sdrc_d5							
CONTROL_PADCONF_SDRC_D6[15:0]	0x4800 203C	sdrc_d6							
CONTROL_PADCONF_SDRC_D6[31:16]	0x4800 203C	sdrc_d7							
CONTROL_PADCONF_SDRC_D8[15:0]	0x4800 2040	sdrc_d8							
CONTROL_PADCONF_SDRC_D8[31:16]	0x4800 2040	sdrc_d9							
CONTROL_PADCONF_SDRC_D10[15:0]	0x4800 2044	sdrc_d10							
CONTROL_PADCONF_SDRC_D10[31:16]	0x4800 2044	sdrc_d11							
CONTROL_PADCONF_SDRC_D12[15:0]	0x4800 2048	sdrc_d12							
CONTROL_PADCONF_SDRC_D12[31:16]	0x4800 2048	sdrc_d13							
CONTROL_PADCONF_SDRC_D14[15:0]	0x4800 204C	sdrc_d14							
CONTROL_PADCONF_SDRC_D14[31:16]	0x4800 204C	sdrc_d15							
CONTROL_PADCONF_SDRC_D16[15:0]	0x4800 2050	sdrc_d16							
CONTROL_PADCONF_SDRC_D16[31:16]	0x4800 2050	sdrc_d17							
CONTROL_PADCONF_SDRC_D18[15:0]	0x4800 2054	sdrc_d18							
CONTROL_PADCONF_SDRC_D18[31:16]	0x4800 2054	sdrc_d19							
CONTROL_PADCONF_SDRC_D20[15:0]	0x4800 2058	sdrc_d20							

Table 7-4. Core Control Module Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_SDRC_D20[31:16]	0x4800 2058	sdrc_d21							
CONTROL_PADCONF_SDRC_D22[15:0]	0x4800 205C	sdrc_d22							
CONTROL_PADCONF_SDRC_D22[31:16]	0x4800 205C	sdrc_d23							
CONTROL_PADCONF_SDRC_D24[15:0]	0x4800 2060	sdrc_d24							
CONTROL_PADCONF_SDRC_D24[31:16]	0x4800 2060	sdrc_d25							
CONTROL_PADCONF_SDRC_D26[15:0]	0x4800 2064	sdrc_d26							
CONTROL_PADCONF_SDRC_D26[31:16]	0x4800 2064	sdrc_d27							
CONTROL_PADCONF_SDRC_D28[15:0]	0x4800 2068	sdrc_d28							
CONTROL_PADCONF_SDRC_D28[31:16]	0x4800 2068	sdrc_d29							
CONTROL_PADCONF_SDRC_D30[15:0]	0x4800 206C	sdrc_d30							
CONTROL_PADCONF_SDRC_D30[31:16]	0x4800 206C	sdrc_d31							
CONTROL_PADCONF_SDRC_CLK[15:0]	0x4800 2070	sdrc_clk							
CONTROL_PADCONF_SDRC_CLK[31:16]	0x4800 2070	sdrc_dqs0							
CONTROL_PADCONF_SAD2D_SBUSFLAG[31:16]	0x4800 2260	sdrc_cke0							safe_mode_out1 ⁽¹⁾
CONTROL_PADCONF_SDRC_CKE1[15:0]	0x4800 2264	sdrc_cke1							safe_mode_out1
CONTROL_PADCONF_SDRC_DQS1[15:0]	0x4800 2074	sdrc_dqs1							
CONTROL_PADCONF_SDRC_DQS1[31:16]	0x4800 2074	sdrc_dqs2							
CONTROL_PADCONF_SDRC_DQS3[15:0]	0x4800 2078	sdrc_dqs3							
CONTROL_PADCONF_SDRC_DQS3[31:16]	0x4800 2078	gpmc_a1				gpio_34			safe_mode
CONTROL_PADCONF_GPMC_A2[15:0]	0x4800 207C	gpmc_a2				gpio_35			safe_mode
CONTROL_PADCONF_GPMC_A2[31:16]	0x4800 207C	gpmc_a3				gpio_36			safe_mode
CONTROL_PADCONF_GPMC_A4[15:0]	0x4800 2080	gpmc_a4				gpio_37			safe_mode
CONTROL_PADCONF_GPMC_A4[31:16]	0x4800 2080	gpmc_a5				gpio_38			safe_mode
CONTROL_PADCONF_GPMC_A6[15:0]	0x4800 2084	gpmc_a6				gpio_39			safe_mode
CONTROL_PADCONF_GPMC_A6[31:16]	0x4800 2084	gpmc_a7				gpio_40			safe_mode
CONTROL_PADCONF_GPMC_A8[15:0]	0x4800 2088	gpmc_a8				gpio_41			safe_mode
CONTROL_PADCONF_GPMC_A8[31:16]	0x4800 2088	gpmc_a9	sys_ndmareq2			gpio_42			safe_mode
CONTROL_PADCONF_GPMC_A10[15:0]	0x4800 208C	gpmc_a10	sys_ndmareq3			gpio_43			safe_mode
CONTROL_PADCONF_GPMC_A10[31:16]	0x4800 208C	gpmc_d0							
CONTROL_PADCONF_GPMC_D1[15:0]	0x4800 2090	gpmc_d1							
CONTROL_PADCONF_GPMC_D1[31:16]	0x4800 2090	gpmc_d2							
CONTROL_PADCONF_GPMC_D3[15:0]	0x4800 2094	gpmc_d3							
CONTROL_PADCONF_GPMC_D3[31:16]	0x4800 2094	gpmc_d4							
CONTROL_PADCONF_GPMC_D5[15:0]	0x4800 2098	gpmc_d5							

⁽¹⁾ Pad initialized as an output in this specific safe mode implementation (buffer in output mode, drive 1).

Table 7-4. Core Control Module Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_GPMC_D5[31:16]	0x4800 2098	gpmc_d6							
CONTROL_PADCONF_GPMC_D7[15:0]	0x4800 209C	gpmc_d7							
CONTROL_PADCONF_GPMC_D7[31:16]	0x4800 209C	gpmc_d8				gpio_44			safe_mode
CONTROL_PADCONF_GPMC_D9[15:0]	0x4800 20A0	gpmc_d9				gpio_45			safe_mode
CONTROL_PADCONF_GPMC_D9[31:16]	0x4800 20A0	gpmc_d10				gpio_46			safe_mode
CONTROL_PADCONF_GPMC_D11[15:0]	0x4800 20A4	gpmc_d11				gpio_47			safe_mode
CONTROL_PADCONF_GPMC_D11[31:16]	0x4800 20A4	gpmc_d12				gpio_48			safe_mode
CONTROL_PADCONF_GPMC_D13[15:0]	0x4800 20A8	gpmc_d13				gpio_49			safe_mode
CONTROL_PADCONF_GPMC_D13[31:16]	0x4800 20A8	gpmc_d14				gpio_50			safe_mode
CONTROL_PADCONF_GPMC_D15[15:0]	0x4800 20AC	gpmc_d15				gpio_51			safe_mode
CONTROL_PADCONF_GPMC_D15[31:16]	0x4800 20AC	gpmc_ncs0							
CONTROL_PADCONF_GPMC_NCS1[15:0]	0x4800 20B0	gpmc_ncs1				gpio_52			safe_mode
CONTROL_PADCONF_GPMC_NCS1[31:16]	0x4800 20B0	gpmc_ncs2				gpio_53			safe_mode
CONTROL_PADCONF_GPMC_NCS3[15:0]	0x4800 20B4	gpmc_ncs3	sys_ndmreq0			gpio_54			safe_mode
CONTROL_PADCONF_GPMC_NCS3[31:16]	0x4800 20B4	gpmc_ncs4	sys_ndmreq1	mcbsp4_clkx	gpt9_pwm_evt	gpio_55			safe_mode
CONTROL_PADCONF_GPMC_NCS5[15:0]	0x4800 20B8	gpmc_ncs5	sys_ndmreq2	mcbsp4_dr	gpt10_pwm_evt	gpio_56			safe_mode
CONTROL_PADCONF_GPMC_NCS5[31:16]	0x4800 20B8	gpmc_ncs6	sys_ndmreq3	mcbsp4_dx	gpt11_pwm_evt	gpio_57			safe_mode
CONTROL_PADCONF_GPMC_NCS7[15:0]	0x4800 20BC	gpmc_ncs7	gpmc_io_dir	mcbsp4_fsx	gpt8_pwm_evt	gpio_58			safe_mode
CONTROL_PADCONF_GPMC_NCS7[31:16]	0x4800 20BC	gpmc_clk				gpio_59			safe_mode
CONTROL_PADCONF_GPMC_NADV_ALE[15:0]	0x4800 20C0	gpmc_nadv_ale							
CONTROL_PADCONF_GPMC_NADV_ALE[31:16]	0x4800 20C0	gpmc_noe							
CONTROL_PADCONF_GPMC_NWE[15:0]	0x4800 20C4	gpmc_nwe							
CONTROL_PADCONF_GPMC_NWE[31:16]	0x4800 20C4	gpmc_nbe0_cle				gpio_60			safe_mode
CONTROL_PADCONF_GPMC_NBE1[15:0]	0x4800 20C8	gpmc_nbe1				gpio_61			safe_mode
CONTROL_PADCONF_GPMC_NBE1[31:16]	0x4800 20C8	gpmc_nwp				gpio_62			safe_mode
CONTROL_PADCONF_GPMC_WAIT0[15:0]	0x4800 20CC	gpmc_wait0							
CONTROL_PADCONF_GPMC_WAIT0[31:16]	0x4800 20CC	gpmc_wait1				gpio_63			safe_mode
CONTROL_PADCONF_GPMC_WAIT2[15:0]	0x4800 20D0	gpmc_wait2				gpio_64			safe_mode
CONTROL_PADCONF_GPMC_WAIT2[31:16]	0x4800 20D0	gpmc_wait3	sys_ndmreq1			gpio_65			safe_mode
CONTROL_PADCONF_DSS_PCLK[15:0]	0x4800 20D4	dss_pclk				gpio_66	hw_dbg12		safe_mode
CONTROL_PADCONF_DSS_PCLK[31:16]	0x4800 20D4	dss_hsync				gpio_67	hw_dbg13		safe_mode
CONTROL_PADCONF_DSS_VSYNC[15:0]	0x4800 20D8	dss_vsync				gpio_68			safe_mode
CONTROL_PADCONF_DSS_VSYNC[31:16]	0x4800 20D8	dss_acbias				gpio_69			safe_mode
CONTROL_PADCONF_DSS_DATA0[15:0]	0x4800 20DC	dss_data0	dsi_dx0	uart1_cts	dssvenc656_data0	gpio_70			safe_mode
CONTROL_PADCONF_DSS_DATA0[31:16]	0x4800 20DC	dss_data1	dsi_dy0	uart1_rts	dssvenc656_data1	gpio_71			safe_mode

Table 7-4. Core Control Module Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_DSS_DATA2[15:0]	0x4800 20E0	dss_data2	dsi_dx1		dssvenc656_data2	gpio_72			safe_mode
CONTROL_PADCONF_DSS_DATA2[31:16]	0x4800 20E0	dss_data3	dsi_dy1		dssvenc656_data3	gpio_73			safe_mode
CONTROL_PADCONF_DSS_DATA4[15:0]	0x4800 20E4	dss_data4	dsi_dx2	uart3_rx_irrx	dssvenc656_data4	gpio_74			safe_mode
CONTROL_PADCONF_DSS_DATA4[31:16]	0x4800 20E4	dss_data5	dsi_dy2	uart3_rx_irtx	dssvenc656_data5	gpio_75			safe_mode
CONTROL_PADCONF_DSS_DATA6[15:0]	0x4800 20E8	dss_data6		uart1_tx	dssvenc656_data6	gpio_76	hw_dbg14		safe_mode
CONTROL_PADCONF_DSS_DATA6[31:16]	0x4800 20E8	dss_data7		uart1_rx	dssvenc656_data7	gpio_77	hw_dbg15		safe_mode
CONTROL_PADCONF_DSS_DATA8[15:0]	0x4800 20EC	dss_data8				gpio_78	hw_dbg16		safe_mode
CONTROL_PADCONF_DSS_DATA8[31:16]	0x4800 20EC	dss_data9				gpio_79	hw_dbg17		safe_mode
CONTROL_PADCONF_DSS_DATA10[15:0]	0x4800 20F0	dss_data10	sdi_dat1n			gpio_80			safe_mode
CONTROL_PADCONF_DSS_DATA10[31:16]	0x4800 20F0	dss_data11	sdi_dat1p			gpio_81			safe_mode
CONTROL_PADCONF_DSS_DATA12[15:0]	0x4800 20F4	dss_data12	sdi_dat2n			gpio_82			safe_mode
CONTROL_PADCONF_DSS_DATA12[31:16]	0x4800 20F4	dss_data13	sdi_dat2p			gpio_83			safe_mode
CONTROL_PADCONF_DSS_DATA14[15:0]	0x4800 20F8	dss_data14	sdi_dat3n			gpio_84			safe_mode
CONTROL_PADCONF_DSS_DATA14[31:16]	0x4800 20F8	dss_data15	sdi_dat3p			gpio_85			safe_mode
CONTROL_PADCONF_DSS_DATA16[15:0]	0x4800 20FC	dss_data16				gpio_86			safe_mode
CONTROL_PADCONF_DSS_DATA16[31:16]	0x4800 20FC	dss_data17				gpio_87			safe_mode
CONTROL_PADCONF_DSS_DATA18[15:0]	0x4800 2100	dss_data18	sdi_vsync	mcspi3_clk	dss_data0	gpio_88			safe_mode
CONTROL_PADCONF_DSS_DATA18[31:16]	0x4800 2100	dss_data19	sdi_hsync	mcspi3_simo	dss_data1	gpio_89			safe_mode
CONTROL_PADCONF_DSS_DATA20[15:0]	0x4800 2104	dss_data20	sdi_den	mcspi3_somi	dss_data2	gpio_90			safe_mode
CONTROL_PADCONF_DSS_DATA20[31:16]	0x4800 2104	dss_data21	sdi_stp	mcspi3_cs0	dss_data3	gpio_91			safe_mode
CONTROL_PADCONF_DSS_DATA22[15:0]	0x4800 2108	dss_data22	sdi_clkp	mcspi3_cs1	dss_data4	gpio_92			safe_mode
CONTROL_PADCONF_DSS_DATA22[31:16]	0x4800 2108	dss_data23	sdi_clkn		dss_data5	gpio_93			safe_mode
CONTROL_PADCONF_CAM_HS[15:0]	0x4800 210C	cam_hs				gpio_94	hw_dbg0		safe_mode
CONTROL_PADCONF_CAM_HS[31:16]	0x4800 210C	cam_vs				gpio_95	hw_dbg1		safe_mode
CONTROL_PADCONF_CAM_XCLKA[15:0]	0x4800 2110	cam_xclka				gpio_96			safe_mode
CONTROL_PADCONF_CAM_XCLKA[31:16]	0x4800 2110	cam_pcclk				gpio_97	hw_dbg2		safe_mode
CONTROL_PADCONF_CAM_FLD[15:0]	0x4800 2114	cam_fld		cam_global_rese		gpio_98	hw_dbg3		safe_mode
CONTROL_PADCONF_CAM_FLD[31:16]	0x4800 2114	cam_d0		csi2_dx2		gpio_99			safe_mode
CONTROL_PADCONF_CAM_D1[15:0]	0x4800 2118	cam_d1		csi2_dy2		gpio_100			safe_mode
CONTROL_PADCONF_CAM_D1[31:16]	0x4800 2118	cam_d2				gpio_101	hw_dbg4		safe_mode
CONTROL_PADCONF_CAM_D3[15:0]	0x4800 211C	cam_d3				gpio_102	hw_dbg5		safe_mode

Table 7-4. Core Control Module Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_CAM_D3[31:16]	0x4800 211C	cam_d4				gpio_103	hw_dbg6		safe_mode
CONTROL_PADCONF_CAM_D5[15:0]	0x4800 2120	cam_d5				gpio_104	hw_dbg7		safe_mode
CONTROL_PADCONF_CAM_D5[31:16]	0x4800 2120	cam_d6				gpio_105			safe_mode
CONTROL_PADCONF_CAM_D7[15:0]	0x4800 2124	cam_d7				gpio_106			safe_mode
CONTROL_PADCONF_CAM_D7[31:16]	0x4800 2124	cam_d8				gpio_107			safe_mode
CONTROL_PADCONF_CAM_D9[15:0]	0x4800 2128	cam_d9				gpio_108			safe_mode
CONTROL_PADCONF_CAM_D9[31:16]	0x4800 2128	cam_d10				gpio_109	hw_dbg8		safe_mode
CONTROL_PADCONF_CAM_D11[15:0]	0x4800 212C	cam_d11				gpio_110	hw_dbg9		safe_mode
CONTROL_PADCONF_CAM_D11[31:16]	0x4800 212C	cam_xclkb				gpio_111			safe_mode
CONTROL_PADCONF_CAM_WEN[15:0]	0x4800 2130	cam_wen		cam_shutter		gpio_167	hw_dbg10		safe_mode
CONTROL_PADCONF_CAM_WEN[31:16]	0x4800 2130	cam_strobe				gpio_126	hw_dbg11		safe_mode
CONTROL_PADCONF_CSII_RX0[15:0]	0x4800 2134	csii_rx0				gpio_112			safe_mode
CONTROL_PADCONF_CSII_RX0[31:16]	0x4800 2134	csii_dy0				gpio_113			safe_mode
CONTROL_PADCONF_CSII_RX1[15:0]	0x4800 2138	csii_dx1				gpio_114			safe_mode
CONTROL_PADCONF_CSII_RX1[31:16]	0x4800 2138	csii_dy1				gpio_115			safe_mode
CONTROL_PADCONF_MCBSP2_FSX[15:0]	0x4800 213C	mcbsp2_fsx				gpio_116			safe_mode
CONTROL_PADCONF_MCBSP2_FSX[31:16]	0x4800 213C	mcbsp2_clkx				gpio_117			safe_mode
CONTROL_PADCONF_MCBSP2_DR[15:0]	0x4800 2140	mcbsp2_dr				gpio_118			safe_mode
CONTROL_PADCONF_MCBSP2_DR[31:16]	0x4800 2140	mcbsp2_dx				gpio_119			safe_mode
CONTROL_PADCONF_MMC1_CLK[15:0]	0x4800 2144	mmc1_clk	ms_clk			gpio_120			safe_mode
CONTROL_PADCONF_MMC1_CLK[31:16]	0x4800 2144	mmc1_cmd	ms_bs			gpio_121			safe_mode
CONTROL_PADCONF_MMC1_DATO[15:0]	0x4800 2148	mmc1_dat0	ms_dat0			gpio_122			safe_mode
CONTROL_PADCONF_MMC1_DATO[31:16]	0x4800 2148	mmc1_dat1	ms_dat1			gpio_123			safe_mode
CONTROL_PADCONF_MMC1_DAT2[15:0]	0x4800 214C	mmc1_dat2	ms_dat2			gpio_124			safe_mode
CONTROL_PADCONF_MMC1_DAT2[31:16]	0x4800 214C	mmc1_dat3	ms_dat3			gpio_125			safe_mode
CONTROL_PADCONF_MMC1_DAT4[15:0]	0x4800 2150	mmc1_dat4		sim_io		gpio_126			safe_mode
CONTROL_PADCONF_MMC1_DAT4[31:16]	0x4800 2150	mmc1_dat5		sim_clk		gpio_127			safe_mode
CONTROL_PADCONF_MMC1_DAT6[15:0]	0x4800 2154	mmc1_dat6		sim_pwrctrl		gpio_128			safe_mode
CONTROL_PADCONF_MMC1_DAT6[31:16]	0x4800 2154	mmc1_dat7		sim_RST		gpio_129			safe_mode
CONTROL_PADCONF_MMC2_CLK[15:0]	0x4800 2158	mmc2_clk	mcsipi3_clk			gpio_130			safe_mode
CONTROL_PADCONF_MMC2_CLK[31:16]	0x4800 2158	mmc2_cmd	mcsipi3_simo			gpio_131			safe_mode
CONTROL_PADCONF_MMC2_DATO[15:0]	0x4800 215C	mmc2_dat0	mcsipi3_somi			gpio_132			safe_mode
CONTROL_PADCONF_MMC2_DATO[31:16]	0x4800 215C	mmc2_dat1				gpio_133			safe_mode
CONTROL_PADCONF_MMC2_DAT2[15:0]	0x4800 2160	mmc2_dat2	mcsipi3_cs1			gpio_134			safe_mode
CONTROL_PADCONF_MMC2_DAT2[31:16]	0x4800 2160	mmc2_dat3	mcsipi3_cs0			gpio_135			safe_mode
CONTROL_PADCONF_MMC2_DAT4[15:0]	0x4800 2164	mmc2_dat4	mmc2_dir_dat0		mmc3_dat0	gpio_136			safe_mode

Table 7-4. Core Control Module Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_MMC2_DAT4[31:16]	0x4800 2164	mmc2_dat5	mmc2_dir_dat1	cam_global_rese t	mmc3_dat1	gpio_137	hsusb3_tll_stp	mm3_rxrdp	safe_mode
CONTROL_PADCONF_MMC2_DAT6[15:0]	0x4800 2168	mmc2_dat6	mmc2_dir_cmd	cam_shutter	mmc3_dat2	gpio_138	hsusb3_tll_dir		safe_mode
CONTROL_PADCONF_MMC2_DAT6[31:16]	0x4800 2168	mmc2_dat7	mmc2_clkin		mmc3_dat3	gpio_139	hsusb3_tll_nxt	mm3_rxrdm	safe_mode
CONTROL_PADCONF_MCBSP3_RX[15:0]	0x4800 216C	mcbsp3_dx	uart2_cts			gpio_140	hsusb3_tll_data4		safe_mode
CONTROL_PADCONF_MCBSP3_RX[31:16]	0x4800 216C	mcbsp3_dr	uart2_rts			gpio_141	hsusb3_tll_data5		safe_mode
CONTROL_PADCONF_MCBSP3_CLKX[15:0]	0x4800 2170	mcbsp3_clkx	uart2_tx			gpio_142	hsusb3_tll_data6		safe_mode
CONTROL_PADCONF_MCBSP3_CLKX[31:16]	0x4800 2170	mcbsp3_fsx	uart2_rx			gpio_143	hsusb3_tll_data7		safe_mode
CONTROL_PADCONF_UART2_CTS[15:0]	0x4800 2174	uart2_cts	mcbsp3_dx	gpt9_pwm_evt		gpio_144			safe_mode
CONTROL_PADCONF_UART2_CTS[31:16]	0x4800 2174	uart2_rts	mcbsp3_dr	gpt10_pwm_evt		gpio_145			safe_mode
CONTROL_PADCONF_UART2_TX[15:0]	0x4800 2178	uart2_tx	mcbsp3_clkx	gpt11_pwm_evt		gpio_146			safe_mode
CONTROL_PADCONF_UART2_TX[31:16]	0x4800 2178	uart2_rx	mcbsp3_fsx	gpt8_pwm_evt		gpio_147			safe_mode
CONTROL_PADCONF_UART1_RX[15:0]	0x4800 217C	uart1_tx				gpio_148			safe_mode
CONTROL_PADCONF_UART1_RX[31:16]	0x4800 217C	uart1_rts				gpio_149			safe_mode
CONTROL_PADCONF_UART1_CTS[15:0]	0x4800 2180	uart1_cts				gpio_150	hsusb3_tll_clk		safe_mode
CONTROL_PADCONF_UART1_CTS[31:16]	0x4800 2180	uart1_rx		mcbsp1_clkr	mcsipi4_clk	gpio_151			safe_mode
CONTROL_PADCONF_MCBSP4_CLKX[15:0]	0x4800 2184	mcbsp4_clkx				gpio_152	hsusb3_tll_data1	mm3_txse0	safe_mode
CONTROL_PADCONF_MCBSP4_CLKX[31:16]	0x4800 2184	mcbsp4_dr				gpio_153	hsusb3_tll_data0	mm3_rxrcv	safe_mode
CONTROL_PADCONF_MCBSP4_RX[15:0]	0x4800 2188	mcbsp4_dx				gpio_154	hsusb3_tll_data2	mm3_txdat	safe_mode
CONTROL_PADCONF_MCBSP4_RX[31:16]	0x4800 2188	mcbsp4_fsx				gpio_155	hsusb3_tll_data3	mm3_txen_n	safe_mode
CONTROL_PADCONF_MCBSP1_CLKR[15:0]	0x4800 218C	mcbsp1_clk	mcsipi4_clk	sim_cd		gpio_156			safe_mode
CONTROL_PADCONF_MCBSP1_CLKR[31:16]	0x4800 218C	mcbsp1_fsr	adpllv2d_dither ing_en1	cam_global_ reset		gpio_157			safe_mode
CONTROL_PADCONF_MCBSP1_RX[15:0]	0x4800 2190	mcbsp1_dx	mcsipi4_simo	mcbsp3_dx		gpio_158			safe_mode
CONTROL_PADCONF_MCBSP1_RX[31:16]	0x4800 2190	mcbsp1_dr	mcsipi4_somi	mcbsp3_dr		gpio_159			safe_mode
CONTROL_PADCONF_MCBSP_CLKS[15:0]	0x4800 2194	mcbsp_clocks		cam_shutter		gpio_160	uart1_cts		safe_mode
CONTROL_PADCONF_MCBSP_CLKS[31:16]	0x4800 2194	mcbsp1_fsx	mcsipi4_cso	mcbsp3_fsx		gpio_161			safe_mode
CONTROL_PADCONF_MCBSP1_CLKX[15:0]	0x4800 2198	mcbsp1_clkx		mcbsp3_clkx		gpio_162			safe_mode
CONTROL_PADCONF_MCBSP1_CLKX[31:16]	0x4800 2198	uart3_cts_rctx				gpio_163			safe_mode
CONTROL_PADCONF_UART3_RTS_SD[15:0]	0x4800 219C	uart3_rts_sd				gpio_164			safe_mode
CONTROL_PADCONF_UART3_RTS_SD[31:16]	0x4800 219C	uart3_rx_irrx				gpio_165			safe_mode
CONTROL_PADCONF_UART3_TX_IRTX[15:0]	0x4800 21A0	uart3_tx_irtx				gpio_166			safe_mode
CONTROL_PADCONF_UART3_TX_IRTX[31:16]	0x4800 21A0	hsusb0_clk				gpio_120			safe_mode
CONTROL_PADCONF_HSUSBO_STP[15:0]	0x4800 21A4	hsusb0_stp				gpio_121			safe_mode
CONTROL_PADCONF_HSUSBO_STP[31:16]	0x4800 21A4	hsusb0_dir				gpio_122			safe_mode
CONTROL_PADCONF_HSUSBO_NXT[15:0]	0x4800 21A8	hsusb0_nxt				gpio_124			safe_mode
CONTROL_PADCONF_HSUSBO_NXT[31:16]	0x4800 21A8	hsusb0_data0		uart3_tx_irtx		gpio_125			safe_mode

Table 7-4. Core Control Module Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_HSUSB0_DATA1[15:0]	0x4800 21AC	hsusb0_data1		uart3_rx_irrx		gpio_130			safe_mode
CONTROL_PADCONF_HSUSB0_DATA1[31:16]	0x4800 21AC	hsusb0_data2		uart3_rts_sd		gpio_131			safe_mode
CONTROL_PADCONF_HSUSB0_DATA3[15:0]	0x4800 21B0	hsusb0_data3		uart3_cts_rctx		gpio_169			safe_mode
CONTROL_PADCONF_HSUSB0_DATA3[31:16]	0x4800 21B0	hsusb0_data4				gpio_188			safe_mode
CONTROL_PADCONF_HSUSB0_DATA5[15:0]	0x4800 21B4	hsusb0_data5				gpio_189			safe_mode
CONTROL_PADCONF_HSUSB0_DATA5[31:16]	0x4800 21B4	hsusb0_data6				gpio_190			safe_mode
CONTROL_PADCONF_HSUSB0_DATA7[15:0]	0x4800 21B8	hsusb0_data7				gpio_191			safe_mode
CONTROL_PADCONF_HSUSB0_DATA7[31:16]	0x4800 21B8	i2c1_scl							
CONTROL_PADCONF_I2C1_SDA[15:0]	0x4800 21BC	i2c1_sda							
CONTROL_PADCONF_I2C1_SDA[31:16]	0x4800 21BC	i2c2_scl				gpio_168			safe_mode
CONTROL_PADCONF_I2C2_SDA[15:0]	0x4800 21C0	i2c2_sda				gpio_183			safe_mode
CONTROL_PADCONF_I2C2_SDA[31:16]	0x4800 21C0	i2c3_scl				gpio_184			safe_mode
CONTROL_PADCONF_I2C3_SDA[15:0]	0x4800 21C4	i2c3_sda				gpio_185			safe_mode
CONTROL_PADCONF_I2C3_SDA[31:16]	0x4800 21C4	hdq_sio	sys_altclk	i2c2_sccbe	i2c3_sccbe	gpio_170			safe_mode
CONTROL_PADCONF_MCSPI1_CLK[15:0]	0x4800 21C8	mcspi1_clk	mmc2_dat4			gpio_171			safe_mode
CONTROL_PADCONF_MCSPI1_CLK[31:16]	0x4800 21C8	mcspi1_simo	mmc2_dat5			gpio_172			safe_mode
CONTROL_PADCONF_MCSPI1_SOMI[15:0]	0x4800 21CC	mcspi1_somi	mmc2_dat6			gpio_173			safe_mode
CONTROL_PADCONF_MCSPI1_SOMI[31:16]	0x4800 21CC	mcspi1_cs0	mmc2_dat7			gpio_174			safe_mode
CONTROL_PADCONF_MCSPI1_CS1[15:0]	0x4800 21D0	mcspi1_cs1	adpll2d_dithering_en2		mmc3_cmd	gpio_175			safe_mode
CONTROL_PADCONF_MCSPI1_CS1[31:16]	0x4800 21D0	mcspi1_cs2			mmc3_clk	gpio_176			safe_mode
CONTROL_PADCONF_MCSPI1_CS3[15:0]	0x4800 21D4	mcspi1_cs3		hsusb2_tll_data2	hsusb2_data2	gpio_177	mm2_txdat		safe_mode
CONTROL_PADCONF_MCSPI1_CS3[31:16]	0x4800 21D4	mcspi2_clk		hsusb2_tll_data7	hsusb2_data7	gpio_178			safe_mode
CONTROL_PADCONF_MCSPI2_SIMO[15:0]	0x4800 21D8	mcspi2_simo	gpt9_pwm_evt	hsusb2_tll_data4	hsusb2_data4	gpio_179			safe_mode
CONTROL_PADCONF_MCSPI2_SIMO[31:16]	0x4800 21D8	mcspi2_somi	gpt10_pwm_evt	hsusb2_tll_data5	hsusb2_data5	gpio_180			safe_mode
CONTROL_PADCONF_MCSPI2_CS0[15:0]	0x4800 21DC	mcspi2_cs0	gpt11_pwm_evt	hsusb2_tll_data6	hsusb2_data6	gpio_181			safe_mode
CONTROL_PADCONF_MCSPI2_CS0[31:16]	0x4800 21DC	mcspi2_cs1	gpt8_pwm_evt	hsusb2_tll_data3	hsusb2_data3	gpio_182	mm2_txen_n		safe_mode
CONTROL_PADCONF_SYS_NIRQ[15:0]	0x4800 21E0	sys_nirq				gpio_0			safe_mode
CONTROL_PADCONF_SYS_NIRQ[31:16]	0x4800 21E0	sys_clkout2				gpio_186			safe_mode
CONTROL_PADCONF_ETK_CLK[15:0]	0x4800 25D8	etk_clk	mcbsp5_clkx	mmc3_clk	hsusb1_stp	gpio_12	mm1_rxdp	hsusb1_tll_stp	hw_dbg0
CONTROL_PADCONF_ETK_CLK[31:16]	0x4800 25D8	etk_ctl		mmc3_cmd	hsusb1_clk	gpio_13		hsusb1_tll_clk	hw_dbg1
CONTROL_PADCONF_ETK_D0[15:0]	0x4800 25DC	etk_d0	mcspi3_simo	mmc3_dat4	hsusb1_data0	gpio_14	mm1_rxrcv	hsusb1_tll_data0	hw_dbg2
CONTROL_PADCONF_ETK_D0[31:16]	0x4800 25DC	etk_d1	mcspi3_somi		hsusb1_data1	gpio_15	mm1_txse0	hsusb1_tll_data1	hw_dbg3
CONTROL_PADCONF_ETK_D2[15:0]	0x4800 25E0	etk_d2	mcspi3_cs0		hsusb1_data2	gpio_16	mm1_txdat	hsusb1_tll_data2	hw_dbg4
CONTROL_PADCONF_ETK_D2[31:16]	0x4800 25E0	etk_d3	mcspi3_clk	mmc3_dat3	hsusb1_data7	gpio_17		hsusb1_tll_data7	hw_dbg5
CONTROL_PADCONF_ETK_D4[15:0]	0x4800 25E4	etk_d4	mcbsp5_dr	mmc3_dat0	hsusb1_data4	gpio_18		hsusb1_tll_data4	hw_dbg6

Table 7-4. Core Control Module Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_ETK_D4[31:16]	0x4800 25E4	etk_d5	mcbsp5_fsx	mmc3_dat1	hsusb1_data5	gpio_19		hsusb1_tll_data5	hw_dbg7
CONTROL_PADCONF_ETK_D6[15:0]	0x4800 25E8	etk_d6	mcbsp5_dx	mmc3_dat2	hsusb1_data6	gpio_20		hsusb1_tll_data6	hw_dbg8
CONTROL_PADCONF_ETK_D6[31:16]	0x4800 25E8	etk_d7	mcsipi3_cs1	mmc3_dat7	hsusb1_data3	gpio_21	mm1_txen_n	hsusb1_tll_data3	hw_dbg9
CONTROL_PADCONF_ETK_D8[15:0]	0x4800 25EC	etk_d8	Reserved for non-GP devices	mmc3_dat6	hsusb1_dir	gpio_22		hsusb1_tll_dir	hw_dbg10
CONTROL_PADCONF_ETK_D8[31:16]	0x4800 25EC	etk_d9	Reserved for non-GP devices	mmc3_dat5	hsusb1_nxt	gpio_23	mm1_rxmdm	hsusb1_tll_nxt	hw_dbg11
CONTROL_PADCONF_ETK_D10[15:0]	0x4800 25F0	etk_d10		uart1_rx	hsusb2_clk	gpio_24		hsusb2_tll_clk	hw_dbg12
CONTROL_PADCONF_ETK_D10[31:16]	0x4800 25F0	etk_d11			hsusb2_stp	gpio_25	mm2_rxdp	hsusb2_tll_stp	hw_dbg13
CONTROL_PADCONF_ETK_D12[15:0]	0x4800 25F4	etk_d12			hsusb2_dir	gpio_26		hsusb2_tll_dir	hw_dbg14
CONTROL_PADCONF_ETK_D12[31:16]	0x4800 25F4	etk_d13			hsusb2_nxt	gpio_27	mm2_rxmdm	hsusb2_tll_nxt	hw_dbg15
CONTROL_PADCONF_ETK_D14[15:0]	0x4800 25F8	etk_d14			hsusb2_data0	gpio_28	mm2_rxrcv	hsusb2_tll_data0	hw_dbg16
CONTROL_PADCONF_ETK_D14[31:16]	0x4800 25F8	etk_d15			hsusb2_data1	gpio_29	mm2_txse0	hsusb2_tll_data1	hw_dbg17

Table 7-5. Core Control Module D2D Pad Configuration Register Fields

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_SAD2D_MCAD0[15:0]	0x4800 21E4	sad2d_mcad0	mad2d_mcad0						
CONTROL_PADCONF_SAD2D_MCAD0[31:16]	0x4800 21E4	sad2d_mcad1	mad2d_mcad1						
CONTROL_PADCONF_SAD2D_MCAD2[15:0]	0x4800 21E8	sad2d_mcad2	mad2d_mcad2						
CONTROL_PADCONF_SAD2D_MCAD2[31:16]	0x4800 21E8	sad2d_mcad3	mad2d_mcad3						
CONTROL_PADCONF_SAD2D_MCAD4[15:0]	0x4800 21EC	sad2d_mcad4	mad2d_mcad4						
CONTROL_PADCONF_SAD2D_MCAD4[31:16]	0x4800 21EC	sad2d_mcad5	mad2d_mcad5						
CONTROL_PADCONF_SAD2D_MCAD6[15:0]	0x4800 21F0	sad2d_mcad6	mad2d_mcad6						
CONTROL_PADCONF_SAD2D_MCAD6[31:16]	0x4800 21F0	sad2d_mcad7	mad2d_mcad7						
CONTROL_PADCONF_SAD2D_MCAD8[15:0]	0x4800 21F4	sad2d_mcad8	mad2d_mcad8						
CONTROL_PADCONF_SAD2D_MCAD8[31:16]	0x4800 21F4	sad2d_mcad9	mad2d_mcad9						
CONTROL_PADCONF_SAD2D_MCAD10[15:0]	0x4800 21F8	sad2d_mcad10	mad2d_mcad10						
CONTROL_PADCONF_SAD2D_MCAD10[31:16]	0x4800 21F8	sad2d_mcad11	mad2d_mcad11						
CONTROL_PADCONF_SAD2D_MCAD12[15:0]	0x4800 21FC	sad2d_mcad12	mad2d_mcad12						
CONTROL_PADCONF_SAD2D_MCAD12[31:16]	0x4800 21FC	sad2d_mcad13	mad2d_mcad13						
CONTROL_PADCONF_SAD2D_MCAD14[15:0]	0x4800 2200	sad2d_mcad14	mad2d_mcad14						
CONTROL_PADCONF_SAD2D_MCAD14[31:16]	0x4800 2200	sad2d_mcad15	mad2d_mcad15						
CONTROL_PADCONF_SAD2D_MCAD16[15:0]	0x4800 2204	sad2d_mcad16	mad2d_mcad16						
CONTROL_PADCONF_SAD2D_MCAD16[31:16]	0x4800 2204	sad2d_mcad17	mad2d_mcad17						
CONTROL_PADCONF_SAD2D_MCAD18[15:0]	0x4800 2208	sad2d_mcad18	mad2d_mcad18						

Table 7-5. Core Control Module D2D Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_SAD2D_MCAD18[31:16]	0x4800 2208	sad2d_mcad19	mad2d_mcad19						
CONTROL_PADCONF_SAD2D_MCAD20[15:0]	0x4800 220C	sad2d_mcad20	mad2d_mcad20						
CONTROL_PADCONF_SAD2D_MCAD20[31:16]	0x4800 220C	sad2d_mcad21	mad2d_mcad21						
CONTROL_PADCONF_SAD2D_MCAD22[15:0]	0x4800 2210	sad2d_mcad22	mad2d_mcad22						
CONTROL_PADCONF_SAD2D_MCAD22[31:16]	0x4800 2210	sad2d_mcad23	mad2d_mcad23						
CONTROL_PADCONF_SAD2D_MCAD24[15:0]	0x4800 2214	sad2d_mcad24	mad2d_mcad24						
CONTROL_PADCONF_SAD2D_MCAD24[31:16]	0x4800 2214	sad2d_mcad25	mad2d_mcad25						
CONTROL_PADCONF_SAD2D_MCAD26[15:0]	0x4800 2218	sad2d_mcad26	mad2d_mcad26						
CONTROL_PADCONF_SAD2D_MCAD26[31:16]	0x4800 2218	sad2d_mcad27	mad2d_mcad27						
CONTROL_PADCONF_SAD2D_MCAD28[15:0]	0x4800 221C	sad2d_mcad28	mad2d_mcad28						
CONTROL_PADCONF_SAD2D_MCAD28[31:16]	0x4800 221C	sad2d_mcad29	mad2d_mcad29						
CONTROL_PADCONF_SAD2D_MCAD30[15:0]	0x4800 2220	sad2d_mcad30	mad2d_mcad30						
CONTROL_PADCONF_SAD2D_MCAD30[31:16]	0x4800 2220	sad2d_mcad31	mad2d_mcad31						
CONTROL_PADCONF_SAD2D_MCAD32[15:0]	0x4800 2224	sad2d_mcad32	mad2d_mcad32						
CONTROL_PADCONF_SAD2D_MCAD32[31:16]	0x4800 2224	sad2d_mcad33	mad2d_mcad33						
CONTROL_PADCONF_SAD2D_MCAD34[15:0]	0x4800 2228	sad2d_mcad34	mad2d_mcad34						
CONTROL_PADCONF_SAD2D_MCAD34[31:16]	0x4800 2228	sad2d_mcad35	mad2d_mcad35						
CONTROL_PADCONF_SAD2D_MCAD36[15:0]	0x4800 222C	sad2d_mcad36	mad2d_mcad36						
CONTROL_PADCONF_SAD2D_MCAD36[31:16]	0x4800 222C	sad2d_clk26mi							safe_mode
CONTROL_PADCONF_SAD2D_NRESPWRON[15:0]	0x4800 2230	sad2d_nrespwron							
CONTROL_PADCONF_SAD2D_NRESPWRON[31:16]	0x4800 2230	sad2d_nreswarm							
CONTROL_PADCONF_SAD2D_ARMMIRQ[15:0]	0x4800 2234	sad2d_armmirq							
CONTROL_PADCONF_SAD2D_ARMMIRQ[31:16]	0x4800 2234	sad2d_umafiq							
CONTROL_PADCONF_SAD2D_SPINT[15:0]	0x4800 2238	sad2d_spint				gpio_187			
CONTROL_PADCONF_SAD2D_SPINT[31:16]	0x4800 2238	sad2d_frint				gpio_32			
CONTROL_PADCONF_SAD2D_DMAREQ0[15:0]	0x4800 223C	sad2d_dmareq0		uart2_dma_tx	mmc1_dma_tx				
CONTROL_PADCONF_SAD2D_DMAREQ0[31:16]	0x4800 223C	sad2d_dmareq1		uart2_dma_rx	mmc1_dma_rx				
CONTROL_PADCONF_SAD2D_DMAREQ2[15:0]	0x4800 2240	sad2d_dmareq2	uart1_dma_tx		uart3_dma_tx				
CONTROL_PADCONF_SAD2D_DMAREQ2[31:16]	0x4800 2240	sad2d_dmareq3	uart1_dma_rx		uart3_dma_rx				
CONTROL_PADCONF_SAD2D_NTRST[15:0]	0x4800 2244	sad2d_ntrst							
CONTROL_PADCONF_SAD2D_NTRST[31:16]	0x4800 2244	sad2d_tdi							
CONTROL_PADCONF_SAD2D_TDO[15:0]	0x4800 2248	sad2d_tdo							
CONTROL_PADCONF_SAD2D_TDO[31:16]	0x4800 2248	sad2d_tms							
CONTROL_PADCONF_SAD2D_TCK[15:0]	0x4800 224C	sad2d_tck							
CONTROL_PADCONF_SAD2D_TCK[31:16]	0x4800 224C	sad2d_rtck							
CONTROL_PADCONF_SAD2D_MSTDBY[15:0]	0x4800 2250	sad2d_mstdby							

Table 7-5. Core Control Module D2D Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_SAD2D_MSTDBY[31:16]	0x4800 2250	sad2d_idlereq							
CONTROL_PADCONF_SAD2D_IDLEACK[15:0]	0x4800 2254	sad2d_idleack							
CONTROL_PADCONF_SAD2D_IDLEACK[31:16]	0x4800 2254	sad2d_mwrite	mad2d_swrite						
CONTROL_PADCONF_SAD2D_SWRITE[15:0]	0x4800 2258	sad2d_swrite	mad2d_mwrite						
CONTROL_PADCONF_SAD2D_SWRITE[31:16]	0x4800 2258	sad2d_mread	mad2d_sread						
CONTROL_PADCONF_SAD2D_SREAD[15:0]	0x4800 225C	sad2d_sread	mad2d_mread						
CONTROL_PADCONF_SAD2D_SREAD[31:16]	0x4800 225C	sad2d_mbustflag	mad2d_sbustflag						
CONTROL_PADCONF_SAD2D_SBUSTFLAG[15:0]	0x4800 2260	sad2d_sbustflag	mad2d_mbustflag						

CAUTION

The D2D pads are available in stacked mode only.

Table 7-6. Wake-Up Control Module Pad Configuration Register Fields

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_I2C4_SCL[15:0]	0x4800 2A00	i2c4_scl	sys_nvmode1						safe_mode
CONTROL_PADCONF_I2C4_SCL[31:16]	0x4800 2A00	i2c4_sda	sys_nvmode2						safe_mode
CONTROL_PADCONF_SYS_32K[15:0]	0x4800 2A04	sys_32k							
CONTROL_PADCONF_SYS_32K[31:16]	0x4800 2A04	sys_clkreq				gpio_1			safe_mode
CONTROL_PADCONF_SYS_NRESWARM[15:0]	0x4800 2A08	sys_nreswarm				gpio_30			safe_mode
CONTROL_PADCONF_SYS_NRESWARM[31:16]	0x4800 2A08	sys_boot0				gpio_2			safe_mode
CONTROL_PADCONF_SYS_BOOT1[15:0]	0x4800 2A0C	sys_boot1				gpio_3			safe_mode
CONTROL_PADCONF_SYS_BOOT1[31:16]	0x4800 2A0C	sys_boot2				gpio_4			safe_mode
CONTROL_PADCONF_SYS_BOOT3[15:0]	0x4800 2A10	sys_boot3				gpio_5			safe_mode
CONTROL_PADCONF_SYS_BOOT3[31:16]	0x4800 2A10	sys_boot4	mmc2_dir_dat2			gpio_6			safe_mode
CONTROL_PADCONF_SYS_BOOT5[15:0]	0x4800 2A14	sys_boot5	mmc2_dir_dat3			gpio_7			safe_mode
CONTROL_PADCONF_SYS_BOOT5[31:16]	0x4800 2A14	sys_boot6				gpio_8			safe_mode
CONTROL_PADCONF_SYS_OFF_MODE[15:0]	0x4800 2A18	sys_off_mode				gpio_9			safe_mode
CONTROL_PADCONF_SYS_OFF_MODE[31:16]	0x4800 2A18	sys_clkout1				gpio_10			safe_mode
CONTROL_PADCONF_JTAG_NTRST[15:0]	0x4800 2A1C	jtag_nrst							
CONTROL_PADCONF_JTAG_NTRST[31:16]	0x4800 2A1C	jtag_tck							
CONTROL_PADCONF_JTAG_TMS_TMSC[15:0]	0x4800 2A20	jtag_tms_tmcs							
CONTROL_PADCONF_JTAG_TMS_TMSC[31:16]	0x4800 2A20	jtag_tdi							

Table 7-6. Wake-Up Control Module Pad Configuration Register Fields (continued)

Register Name	Physical Address	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_PADCONF_JTAG_EMU0[15:0]	0x4800 2A24	jtag_emu0				gpio_11			safe_mode
CONTROL_PADCONF_JTAG_EMU0[31:16]	0x4800 2A24	jtag_emu1				gpio_31			safe_mode
CONTROL_PADCONF_SAD2D_SWAKEUP[15:0] ⁽¹⁾	0x4800 2A4C	sad2d_swakeup							
CONTROL_PADCONF_SAD2D_SWAKEUP[31:16]	0x4800 2A4C	jtag_rck							
CONTROL_PADCONF_JTAG_TDO[15:0]	0x4800 2A50	jtag_tdo							

⁽¹⁾ The D2D_swakeup pad is available in stacked mode only.

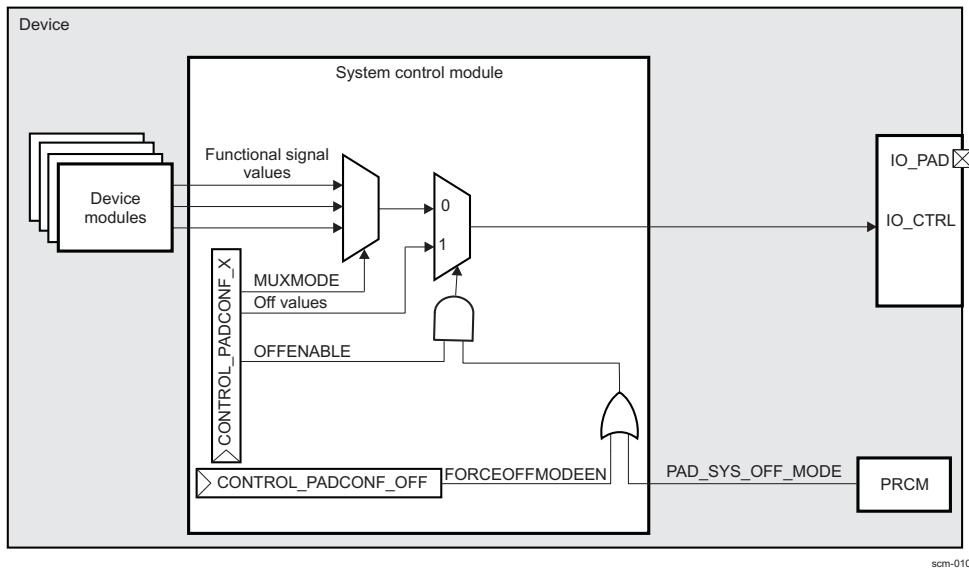
NOTE: Pad names are signal names available in mode 0.

7.4.4.4 System Off Mode

When off mode is active (PAD_SYS_OFF_MODE = 0b1 from PRCM or FORCEOFFMODEENABLE bit CONTROL_CONTROL_PADCONF_OFF[0] = 0b1), the off mode values field CONTROL_CONTROL_PADCONF_X overrides the pad state when OFFENABLE bit CONTROL_CONTROL_PADCONF_X is set.

Figure 7-9 shows the off mode pad control.

Figure 7-9. System Off Mode Pad Control Overview



If off mode is active and the OFFENABLE bit is set to disable, the pad keeps the AND value of the configuration (input/output, PU/PD) it had before going into off mode:

- For an input, the pad is isolated and the pull remains active.
- For an output, the value is latched before going into off mode, to drive the same value in off mode

For more information about off mode, see [Chapter 4, Power, Reset, and Clock Management](#).

For more information about the preliminary settings that must be done before performing OFF <-> ON transitions, see [Section 7.5.3, Off Mode Preliminary Settings](#).

7.4.4.4.1 Save-and-Restore Mechanism

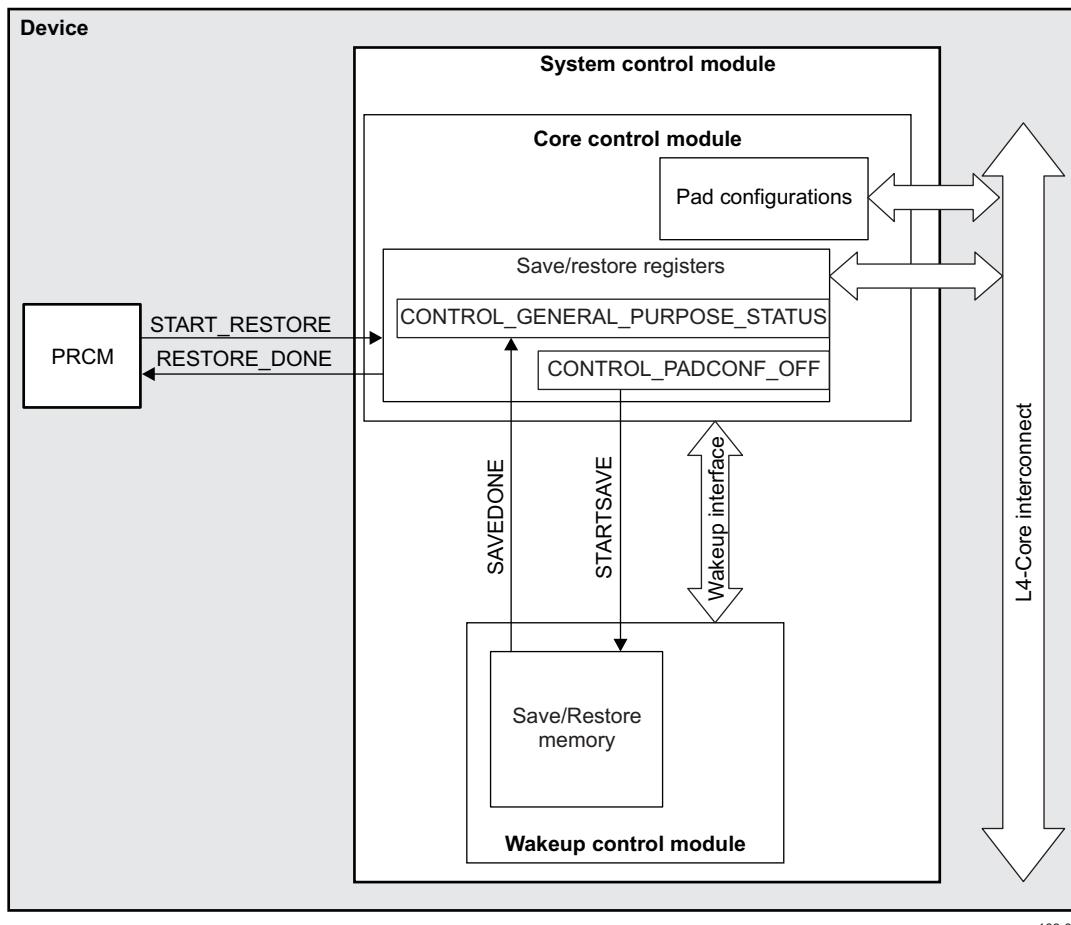
Before going to off mode there is a context saving of the device. The save-and-restore mechanism saves the pad configuration registers (in the CORE power domain) in a WKUP power domain memory (physical addresses 0x4800 2600 to 0x4800 29FC) before going to off mode, and restores those registers when returning from off mode. This mechanism uses the dedicated wake-up interface between the core control module and the wake-up control module.

The save mechanism in the pad configuration registers is activated by setting the STARTSAVE bit CONTROL.CONTROL_PADCONF_OFF[1]. When all pad configuration registers have been saved to the wake-up memory, the status SAVEDONE bit CONTROL.CONTROL_GENERAL_PURPOSE_STATUS[0] is set. In smart-idle mode, the idleAck is returned when the save process is complete.

When returning from off mode, the registers are restored after the PRCM module asserts the START_RESTORE signal. The SCM returns a RESTORE_DONE signal to the PRCM module when the restore process completes.

Figure 7-10 shows an overview of the save-and-restore mechanism in off mode.

Figure 7-10. Save-and-Restore Mechanism Overview

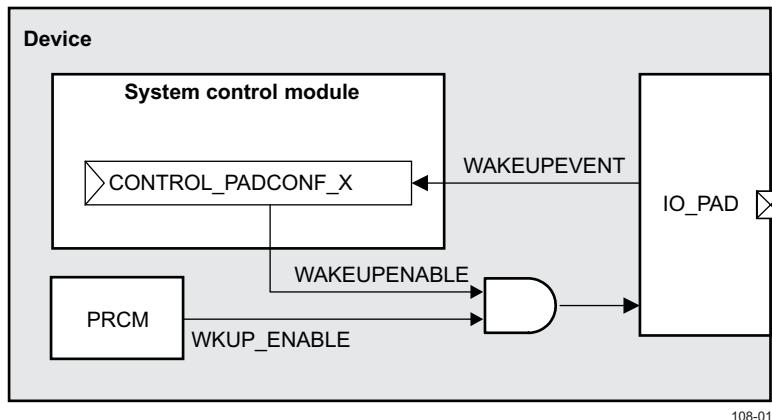


108-011

7.4.4.4.2 Wake-Up Event Detection

In off mode, wake-up event detection can also be enabled on an input pad. The pad wake-up event is latched in the WAKEUPEVENT bit CONTROL. **CONTROL_PADCONF_X**.

The off mode I/O pads wake-up scheme is enabled by setting the EN_I/O bit PRCM.PM._WKEN_WKUP[8]. The wake-up scheme status is transmitted by the WKUP_ENABLE signal. The wake-up event detection capability of each I/O pad of the device is individually enabled/disabled by writing WAKEUPENABLE bit CONTROL. **CONTROL_PADCONF_X**.

Figure 7-11. Wake-Up Event Detection Overview


For more information about the wake-up sequences, see [Chapter 4, Power, Reset, and Clock Management](#).

NOTE: When wake-up detection is enabled for a pad, the pad must be configured as input to avoid contention between the OMAP output buffer and an external driver. If this pin is already configured as an input in active mode, hardware automatically retains the same input state; no software action is required.

If this pin is configured as an output in active mode, the OFF override function must be enabled to set the pin as an input during off mode: in the CONTROL.CONTROL_PADCONF register, set OFFENABLE to 0x1 to enable the OFF override function, and set OFFOUTENABLE to 0x1 to switch this pin to input mode.

[Table 7-7](#) lists the bit directions of the CONTROL_PADCONF_x registers.

Table 7-7. Bit Directions for CONTROL_PADCONF_x Registers

CONTROL_PADCONF_x Bit	Bit Direction	
	0	1
PULLUDENABLE	Not activated	Activated
PULLTYPESELECT	Pulldown	Pullup
INPUTENABLE	Input enable signal inactive	Input enable signal active
OFFENABLE	Off mode values are invalid.	Off mode values are valid.
OFFOUTENABLE	Output	Input
OFFOUTVALUE	Low	High
OFFPULLUDENABLE	Not activated	Activated
OFFPULLTYPESELECT	Pulldown	Pullup
WAKEUPENABLE	Disable I/O wake-up function.	Enable I/O wake-up function.
WAKEUPEVENT	Wake-up event not detected	Detect wake-up event.

7.4.5 Extended-Drain I/O Pin and PBIAS Cell

The device mainly supports 1.8-V I/O voltage on its interfaces, with the exception of MMC/SD/SDIO1 interface, which support both 1.8-V and 3.0-V voltages via internal PBIAS generation and extended-drain I/Os.

The need for embedded extended-drain I/Os on MMC/SD/SDIO1 interface imposes the use of embedded PBIAS cells to provide 1.8-V or 3.0-V reference voltage. The PBIAS cells and the extended-drain I/Os are software-controlled by bits located in the CONTROL.CONTROL_PBIAS_LITE register of the SCM. See [Section 7.6.3, Register Descriptions](#), for the description of this register.

Bangalore –Nov 8th 2010 -Innovate Software Solutions Pvt Ltd today announced that plan of action for Hawkboard and Hawkboard LITE Second lot (identification is DDR with marking D9LHR, processor being production version of OMAP L 138).

The problems experienced by some users in TFTP hanging , Linux kernel not booting were noticed in the second lot (identification is DDR with marking D9LHR, processor being production version of OMAP L 138) of Hawkboard & Hawkboard LITE shipped between August 1st 2010 to Oct 20th 2010, approximately 300 in number.

Innovate Software Solutions Pvt Ltd is currently working to develop a remedy for this condition. Until this remedy is developed, customers will receive an interim notice instructing them to try the current fix (software) by themselves.

Please follow this link for the fix

<http://www.innovatesolutions.net/products/hawkboard>

Once the remedy has been developed, customers will receive a secondary notice advising the customer of the remedy availability.

If the Problem turns out to in the Software – The new fixed Software will be uploaded in Innovate's website with complete instructions –including source code.

If the problem turn out to in the Hardware -The boards must be shipped to the distributors from where it has been purchased and Innovate will collect the same from the distributors, fix the issue and ship it back to distributors who in turn which ship to individual customers.

If the above solution is not fixed within 60 days from this press release we will be recalling the entire lot of Hawkboard, Hawkboard Lite with the above said problem & issue new completely tested boards.

For any information on the above

Ms.Kashma

Email : kashma@innovatesolutions.net

Tel:+91-80-41105384/9162

Bangalore –Dec 18th 2010 -Innovate Software Solutions Pvt Ltd today announced the solution for Hawkboard and Hawkboard LITE Second lot (identification is DDR with marking D9LHR, processor being production version of OMAP L 138).

The problems experienced by some users in TFTP hanging , Linux kernel not booting were noticed in the second lot (identification is DDR with marking D9LHR, processor being production version of OMAP L 138) of Hawkboard & Hawkboard LITE shipped between August 1st 2010 to Oct 20th 2010, approximately 300 in number.

Solution is as follows

On the Backside of Hawkboard or Hawkboard LITE the 2 Ferrite Beads –FB12 and FB13 need to be removed and shorted as shown

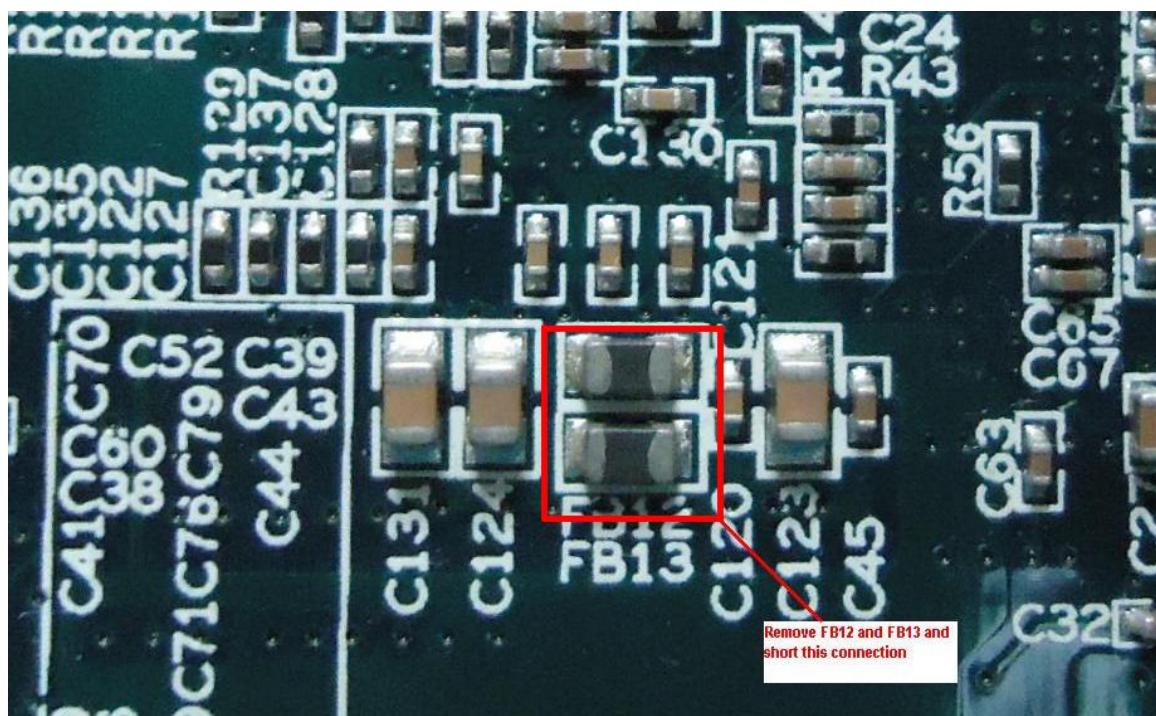


Fig1 : above shows the current position.

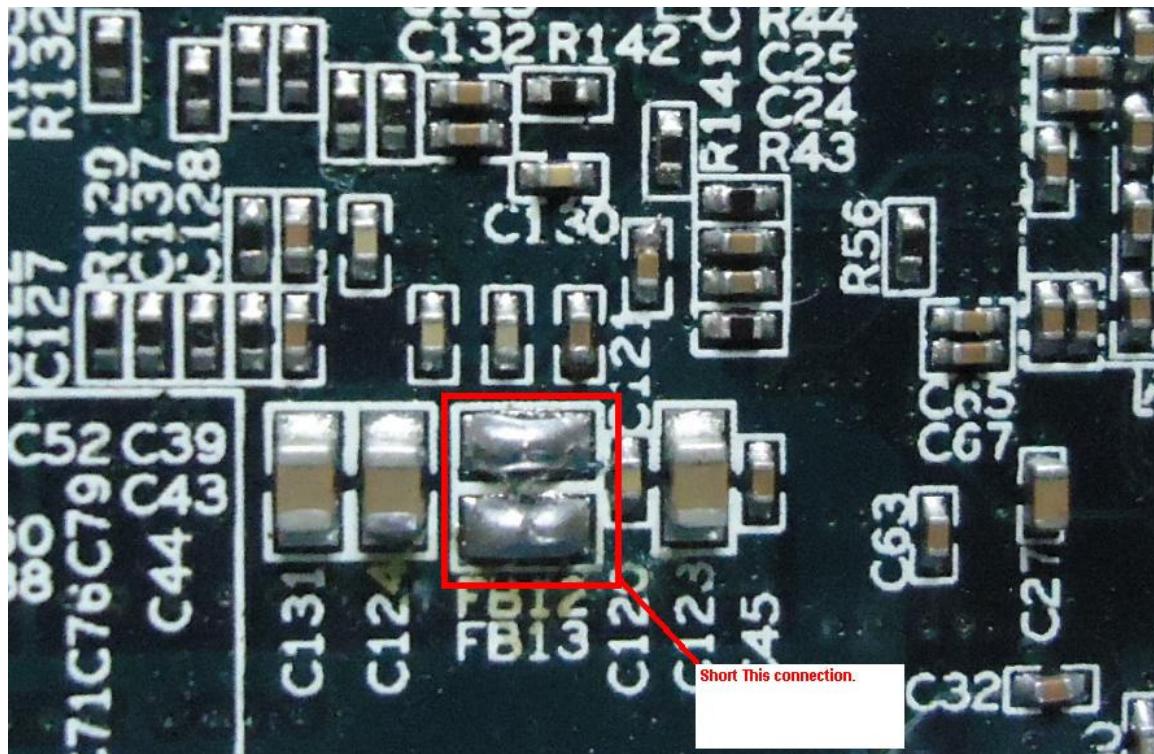


Fig 2: Make the change as this picture

Fix., Recall Policy

Those of you who can remove the 2 Ferrite Beads and short the connection as shown in the solution above, it would be great. The warranty will be applicable for this change by users.

For OTHERS who cannot solve it. : Please send the Boards to your local distributors who will then collect the same and send it to us for the FIX. One side shipping charges is to be borne by users and one side will be done by Innovate. Local customs duties will also be borne by Users

Distributors need to maintain a board identification number for each customer's boards. Distributors must send us minimum of 25boards at once to Innovate for the FIX. There will be NO repair cost for the users.

Time duration

The Recall date has been fixed for a period from 20th Dec 2010 to Feb 5th 2011. Only during this time the board will be accepted. No boards will be taken after this time since the Hawkboard platform would be revised and support for this at local site would not be available.

For any information on the above

Ms.Kashma

Email : kashma@innovatesolutions.net or sales@innovatesolutions.net

Tel:+91-80-41105384/9162

Proxmark.org - Build a cheap 13.56MHz MIFARE antenna for the Proxmark - v1.0

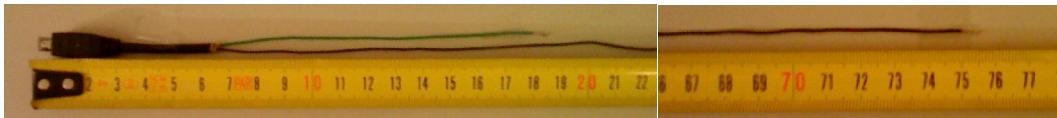


To make a very cheap High-Frequency (HF) antenna for all 13.56 MHz modulations (MIFARE, Felica, etc.), you only need a cheap USB<->Hirose cable. On the left you see a Hirose connector, there are multiple mini-USB cables available on the market, so make sure you buy the correct one.

First you have to strip the cable. You can cut off the USB side; we do not need this part. Make sure the total size of the cable is bigger than 80cm. After cutting the cable you make an incision at ~6.5cm and remove the isolation. Do the same to the shielding that is underneath the plastic isolation. You will see 4 wires appear in different colors.



We do not need the red and white wire for a HF antenna. Those are for connecting a LF antenna. So cut away the shielding, red and white wire. We have left, the connector and 2 wires, black and green, which are at least 80cm. Cut the green wire at ~19cm from the connector and strip it a bit. Do the same to the black wire, but use the length ~76cm.



Make an antenna coil of 3 windings using the green cable. Connect the green cable with the black one and tape them together so they won't unwind.



Plug it into the Proxmark and you will get an antenna with 13V. You can even optimize this value by adjusting the length of the wires. Note that small changes (1cm) can already have a big impact. If you found out better values than we describe, please drop a note so we update this manual.

Roel Verdult
<http://www.proxmark.org>

Ångström Manual

– *Embedded Power* –

MERCIADRI¹ LUCA, KOËN² Kooi

June 29, 2010

¹Luca.Merciadri@student.ulg.ac.be

²k.kooi@student.utwente.nl

Contents

Thanks	1
License	2
I History	3
II Installation	5
 1 Warnings	6
 2 Installation options	7
 3 Loopback Image	8
3.1 Use	8
3.2 Interest	8
 4 LiveRam Disks	10
4.1 Interest	10
4.2 The Linux Boot Process	10
4.3 Easy Versions	11
4.4 Hard Versions	11
4.4.1 In Theory	11
4.4.2 In Practice	12
4.4.2.1 The Python Script	12
4.4.2.2 The Outfile	13
4.4.2.3 HaRET	13
4.4.2.4 zImage	13
4.4.2.5 Initrd	13
4.4.2.6 The Startup Script	13
4.4.2.7 Automatizing The Task	13
4.4.3 Summary	14
4.5 Kexecboots	15
4.5.1 Credits	15
4.5.2 Building	15
4.5.2.1 Overriding the Default Version	15
4.5.2.2 Remarks	16
4.5.3 Role	16

4.5.4	Way of Work	17
4.5.5	Consequences	17
4.5.6	Screenshots	17
5	Diff. Img. for Diff. Purp.	19
5.1	Formats	19
6	Hardware	21
6.1	Supported Devices	21
6.1.1	Installation Procedure	21
6.1.2	List	22
7	After Installation	24
7.1	First Steps	24
7.1.1	Security Before Everything	24
7.1.2	Internationalization and Localization	24
7.1.2.1	Two Main Definitions	24
7.1.2.2	Locale and Packages	25
8	Bug Tracking System	26
III	Building	27
IV	The Ångström System	30
9	Introduction	31
10	Packaging System	32
10.1	Introduction	32
10.2	Core Principle	32
10.3	Commands	33
10.4	Selecting Packages	34
10.5	Format of Packages	34
10.6	Sw. fr. Cons. Img. to X Serv.	34
11	Use	35
11.1	The User	35
11.2	Checking the Version	35
11.3	Global Networking	35
11.3.1	Basic Info	35
11.3.2	With USB	36
11.3.2.1	On the Server	36
11.3.2.2	On the Slave	37
11.3.3	With Bluetooth	38
11.3.4	With WiFi	39
11.3.4.1	Preliminaries	39
11.3.4.2	Commands	40

11.3.4.3	WEP	40
11.3.4.4	WPA	40
11.3.4.5	Drivers	41
11.3.5	Firewall	41
11.4	Errors You Could Encounter	41
12	Getting More Help	42

Thanks

I (Luca Merciadri) would sincerely want to thanks **Andrea Adami** for his information about kexec-stuff, **Yuri Bushmelev** for his details on kexec-stuff, **Gregory Graeme** for his first answer, **Kooi Koen** for his idea of setting up a seperate git repo on the Ångström site or gitorious, and modifying it slightly, **Lex Landa** for his investment about hx4700, and for his precious help (in particular: Section 10.2, p. 32, Subsections 4.5.1, 4.5.2, 4.5.2.1, resp. 15, p. 15, p. 15), **Kelly Price** for his info about the kexecboot kernels.

License

THE Angström Manual is under Creative Commons (CC) BY-SA license. In this, *BY* and *SA* globally mean ([3, 4]) that others can remix, tweak, and build upon our work, even for commercial reasons, as long as others credit us and license their creations under the identical terms. All new works based on this will carry the same license, so any derivatives will also allow commercial use.

You are thus **free to**

1. *share*: copy, distribute and transmit the work,
2. *remix*: adapt the work

under the following assumptions:

1. You must attribute the work in the manner specified by Luca Merciadri,
2. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

Any of the above conditions can be waived if you get permission from Luca Merciadri, as Luca Merciadri holds the copyright.

In **no way are any of the following rights affected** by the license:

- Your fair dealing or fair use rights,
- The author's moral rights,
- Rights other persons may have either in the work itself or in how the work is used, such as publicity or private rights.

For a more detailed description of the license, please read

<http://creativecommons.org/licenses/by-sa/3.0/legalcode>.

If you need further information about this, please [contact me](#).

© Merciadri Luca 2009. Please see Page [1](#) for thanks.

Part I

History

Ångström was started by a small group of people who worked on the OpenEmbedded, OpenZaurus and OpenSimpad projects to unify their effort to make a stable and user-friendly distribution for embedded devices like handhelds, set top boxes and network-attached storage devices and more.

OpenEmbedded is used by Ångström but Ångström is *not* OpenEmbedded Ångström.

Part II

Installation

CHAPTER 1

Warnings

BEFORE beginning to speak about Ångström, here are some problems you could encounter when trying it:

1. Installing Ångström could result in a bricked device, with the consequences of this: your device could be completely useless.
2. Modifying the internal configuration of your device voids warranty.
3. Trying Ångström on older PocketPC, Windows Mobile and Windows CE devices is not without any danger: it erases the content of the RAM. If you want to exit Ångström, you are obliged to proceed to a hard-reset. The result of this hard-reset is that you lose all your personal data, programs, *etc*. After hard-resetting your device, you will find your system, in the same state as when you first booted your device after having bought it.

These things being said, enjoy reading this, and try Ångström on your embedded device!

CHAPTER 2

Installation options

FOR a given device, there are different possibilites to try Ångström on it:

1. **Using the internal flash memory.** This method is potentially dangerous, as it could result in a bricked device.
2. **Using external storage.** Running Ångström of an SD or CF card is relatively safe
3. **Using a loopback image.** Like the external storage option, but using a monolithic file.
4. **Using a liveram disk.**

These options are highly device dependent, so the instructions are broken down per device group.

CHAPTER 3

Loopback Image

Contents

3.1	Use	8
3.2	Interest	8

3.1 Use

ACCORDING to [6] and to my tests, the use of a *loopback image* for Ångström is done by following these steps:

1. Once you are in <http://www.angstrom-distribution.org/releases/2007.12/images/>,
2. Download two files:
 - (a) a .img.bz2 one;
 - (b) a .exe one,
3. Place them on an extension card, in the same (root) directory,
4. Launch from your OS the .exe file,
5. Choose the .img.bz2 image once Ångström will have asked you which image it should use, by
 - (a) Naviguating between images with the “physical” arrows of your device (if so),
 - (b) Pressing between the four “physical” arrows of your device (to confirm, if so).

3.2 Interest

Once these steps have been performed, Ångström will be *executed*. If you choose to read the image from your extension card (and it should be so), all the things you will modify under Ångström (*i.e.* new sessions, *etc.*), will be written on the extension card.

Consequently, even if you hard-reset the device, doing all the given steps will, if you have already tried Ångström at least once, using the same extension card as the one you are using now, let you use the session you defined, all your parameters being written in the memory of your extension card.

CHAPTER 4

LiveRam Disks

Contents

4.1	Interest	10
4.2	The Linux Boot Process	10
4.3	Easy Versions	11
4.4	Hard Versions	11
4.5	Kexecboots	15

4.1 Interest

LiveRam Disks is an interesting way to try Ångström without any great danger¹ (as you only lose personal data and programs, *i.e.* internal configuration). However, all the changes you will do will *not* be kept.

A LiveRam Disk is simply a .exe file which can be executed from the File Explorer of your Windows Mobile Edition.

It should be clear in your mind that these LiveRam Disks are less interesting for the developers; as a result, they are less “up-to-date.” If you want to use up-to-date LiveRam Disks, you will have to do some tricky things which will be given in 4.4, p. 11.

We shall give the important info about these two possibilities. To simplify the presentation, we shall denote by “easy version” a version which is directly available on a website, and which is ready to execute, on the opposite of “hard versions,” whose file is the result of a compilation of different files.

We begin by a rough explanation about how Linux boots.

4.2 The Linux Boot Process

The *Linux boot process* is simple [10]:

¹More info about this at Chapter 1, p. 6, point ??.

1. A boot loader finds the kernel image on the disk, loads it into memory, and starts it,
2. The kernel initializes the devices and its drivers,
3. The kernel mounts the root filesystem,
4. The kernel starts a program called `init`,
5. `init` sets the rest of the processes in motion,
6. The last processes that `init` starts as part of the boot sequence allow you to log in.

4.3 Easy Versions

The *easy versions* date back to the year 2007. A hx4700 version can be found² with the name

`Angstrom-x11-image-liveramdisk-2007.12-hx4700.exe`

You simply use such a file by launching it in the WM File Explorer. That is really straightforward.

4.4 Hard Versions

The *hard versions* are as up-to-date as you expect them to be, because you constitute them. After having spoke with Lex Landa because of a WiFi problem, some info emerged from this thread³, becoming increasingly big.

4.4.1 In Theory

To build this exe file, there is a Python script. It is called `make-bootbundle.py`. Its usage is described by:

```
make-bootbundle.py -o <outfile> <path to haret.exe> <zImage> <initrd> <script>
```

Let's describe these elements in their order:

1. `make-bootbundle.py`: the Python script.
2. `<outfile>`: the name of the file you will launch in the WM File Explorer.
3. `<haret.exe>`: the name of the HaRET bootloader.
4. `<zImage>`: the kernel.
5. `<initrd>`: the initrd.
6. `<script>`: the startup script.

To understand what they mean, you need some basic knowledge in Linux/UNIX systems which will not be detailed here. That is a prerequisite.

²In the <http://www.angstrom-distribution.org/releases/2007.12/images/hx4700/> directory.

³(See <http://www.mail-archive.com/angstrom-distro-users@linuxtogo.org/msg02710.html>.)

4.4.2 In Practice

Practically, you may wonder how all these files can be found, as, even with some searches on the Internet, you may still be wondering where they are.

4.4.2.1 The Python Script

You can use the following Python script (thanks to Lex Landa for this):

```
#!/usr/bin/env python

import sys
import os
import stat
import struct
import getopt
import getopt
import re

optlist, args = getopt.gnu_getopt(sys.argv[1:], "o:vh?")
opts = {}
opts.update(optlist)

if len(args) != 4:
    print "make-bootbundle - Make a standalone HaRET boot bundle with kernel and initrd"
    print "Usage: make-bootbundle.py -o <outfile> <path to haret.exe> <zImage> <initrd> <script>"
    sys.exit(0)

outfile = opts["-o"]

os.system("cat %s %s %s %s> %s" % (args[0], args[1], args[2], args[3], outfile))

exe = open(outfile, "r+b")
kernelSt = os.stat(args[1])
initrdSt = os.stat(args[2])
scriptSt = os.stat(args[3])

exe.seek(0, 2)
exe.write("HARET1\0\0")
exe.write(struct.pack("i", kernelSt[stat.ST_SIZE]))
exe.write(struct.pack("i", initrdSt[stat.ST_SIZE]))
exe.write(struct.pack("i", scriptSt[stat.ST_SIZE]))
exe.write(struct.pack("i", 0))
exe.write(struct.pack("i", 0))
exe.write(struct.pack("i", 0))
exe.close()

if opts.has_key("-v"):
    haretSt = os.stat(args[0])
    print "HaRET:\t", haretSt[stat.ST_SIZE]
    print "Kernel:\t", kernelSt[stat.ST_SIZE]
    print "Initrd:\t", initrdSt[stat.ST_SIZE]
    print "Script:\t", scriptSt[stat.ST_SIZE]
    print "Header:\t", 8 + 4*6
    print "-----"
    print "Total:\t", \
        haretSt[stat.ST_SIZE] + kernelSt[stat.ST_SIZE] + \
        initrdSt[stat.ST_SIZE] + scriptSt[stat.ST_SIZE] + 8 + 4*6
```

4.4.2.2 The Outfile

There is nothing to say about the *outfile*, except that giving it a representative name is a nice thing. For this, you may use the script given at point 4.4.2.7, p. 13.

4.4.2.3 HaRET

The *HaRET* bootloader is already an old thing: it was also used in Familiar. The latest one⁴ is *haret-0.5.2.exe*.

4.4.2.4 zImage

The easiest thing is to use the *Narcissus* image generator⁵ from Ångström's website. It will generate a file like⁶ *hx4700-full-x11-image-hx4700.tar.bz2* for the *hx4700*. You then go to its folder named *boot*, and extract the corresponding *zImage*. For example, it is here *zImage-2.6.21-hh20*. Its numbering is the kernel's numbering, as the *zImage* is the kernel.

4.4.2.5 Initrd

The *initrd* file is under the form of a *.cpio.gz*. For example, its current name is *initramfs.cpio.gz*. You can use the following bash script to make the *initrd* (thanks to Lex Landa for this):

```
#!/bin/sh
# Execute this from the initrd root dir.
su
find . | cpio -o -H newc | gzip -9 > ../../initramfs.cpio.gz
```

4.4.2.6 The Startup Script

You can use the following *startup script*:

```
# initramfs-bootmenu HaRET script
# (c) 2007 Paul Sokolovsky
msgbox "Attention!" "This will boot Linux. All memory will be overwritten and hard reset will be
required to return to Windows Mobile.
You *MUST BACKUP* your data first. Continue booting (Cancel - quit to make backup)??" 0x40031
if result == 2 exit
set cmdline "root=/dev/ram0 rootdelay=3
ip=192.168.2.202:192.168.2.200:192.168.2.200:255.255.255.0:ipaq:usb0
console=tty0 console=ttyS0,115200n8 rdinit=/sbin/init"
ramboot
```

It is named *startup.txt.bootimage* in our example.

4.4.2.7 Automatizing The Task

If you want to make the task of producing the bootbundle more automatic, you can use the following bash script (thanks to Lex Landa for this):

⁴Which can be found in the <http://handhelds.org/~koconnor/haret/> directory.

⁵That you can find at <http://www.angstrom-distribution.org/narcissus/>.

⁶The name of this file has actually been decided only by me, but it represents the content of it.

```
#!/bin/sh

# Run this to build the boot bundle
# L. 01-Nov-09

dt=`date +%d%m%y-%H%M%S`

./make-bootbundle.py -o bootbundle-${dt}.exe haret-0.5.2.exe
zImage-2.6.21-hh20 initramfs.cpio.gz startup.txt.bootimage
```

What it actually does is use the predefined files, and give a date-related name to the boot bundle file (*i.e.* the .exe that you will execute from the WM File Explorer).

You can evidently modify it as you want.

4.4.3 Summary

You will find here a summary of all these files. I suggest you to download them at the place of copying the code of this booklet, as, for Python scripts, indentation is important, and it could be broken by the output of this document.

File	Role	Download at
<i>Python script</i>	Produce the bootbundle	http://www.student.montefiore.ulg.ac.be/~merciadri/angstrom/files/make-bootbundle.py
<i>Outfile</i>	The bootbundle	http://www.student.montefiore.ulg.ac.be/~merciadri/angstrom/files/console-zImage-2.6.21-hh20-hx4700.exe
<i>HARET</i>	The bootloader	http://handhelds.org/~koconnor/haret/haret-0.5.2.exe or http://www.student.montefiore.ulg.ac.be/~merciadri/angstrom/files/haret-0.5.2.exe
<i>zImage</i>	The kernel	http://www.student.montefiore.ulg.ac.be/~merciadri/angstrom/files/zImage-2.6.21-hh20
<i>initrd</i>	The initrd	http://www.student.montefiore.ulg.ac.be/~merciadri/angstrom/files/initramfs.cpio.gz
<i>Startup script</i>	The startup script	http://www.student.montefiore.ulg.ac.be/~merciadri/angstrom/files/startup.txt.bootimage

Table 4.1: The different files for the “Hard version.”

If you use such a LiveRam Disk, Linux should boot in something like 10 seconds.

4.5 Kexecboots

4.5.1 Credits

Yuri Bushmelev, Thomas Kunze and Matthew Allum are the authors of the kexecboot.

4.5.2 Building

The OpenEmbedded (<http://www.openembedded.org>) system is used to build kexecboot. Please see the "Getting Started" for this, for example at

http://wiki.openembedded.net/index.php/Getting_started

document for instructions when setting up OpenEmbedded. Please ensure that you have updated your OpenEmbedded tree recently (do

```
git pull
```

from the openembedded / directory).

4.5.2.1 Overriding the Default Version

The default kexecboot version can be overridden by specifying a BitBake recipe file (.bb) when invoking bitbake, or by setting the variable PREFERRED_VERSION_linux-kexecboot in local.conf, according to [1], for example with

```
bitbake linux-kexecboot -b \
  ../../openembedded/recipes/kexecboot/linux-kexecboot_2.6.32.bb
```

for 2.6.32. This will build kexecboot with a 2.6.32 kernel. Depending on the options specified in conf/local.conf, the kexecboot files may be written in a variety of formats.

There are other kexecboot recipe files, such as (2010):

```
initramfs-kexecboot-image.bb
kexecboot-cfg_0.1.bb
kexecboot_git.bb
linux-kexecboot_2.6.21-hh20.bb
linux-kexecboot_2.6.24.bb
linux-kexecboot_2.6.26.bb
linux-kexecboot_2.6.29.bb
linux-kexecboot_2.6.32+2.6.33-rc3.bb
linux-kexecboot_2.6.32.bb
```

More specifically, the bitbake process consists of downloading and building the cross-compiler toolchain, native toolchain and kernel, and building the cpio-format initramfs image (linux-kexecboot-image) for the kexecboot system. This image is used as a ramdisk-based root filesystem, and its purpose is to mount one or more filesystems, locate the replacement kernel (and root filesystem) and call the kexecboot kernel routine to replace the kernel.

Before the kernel is built, any patches are downloaded and applied, depending on the machine type which is selected. The kernel image is compressed and written, along with the initramfs filesystem.

The kexecboot kernel's command line arguments are set in the default recipe file, *i.e.* `recipes/kexecboot/linux-kexecboot.inc`. The `CMDLINE` variable is set conditionally, depending on the platform (any, poodle, akita, collie, spitz and hx4700). The default kernel configuration is read from file `defconfig` and in the `do_configure_append()` section, some edits are made to remove various configure options, and add others. `CONFIG_BLK_DEV_INITRD` and `CONFIG_KEXEC` are enabled, and the `initramfs` source file is set to `initramfs.cpio.gz`. Kernel modules are disabled and the kernel is built with the configuration in file `.config`.

The `require` line in `linux-kexecboot.inc` includes `../linux/linux.inc`, which builds the kexecboot kernel. Conditional tests decide whether OABI or EABI support should be included, set the Linux logo type, Thumb instruction set support, endianness and other options.

The kernel name is set with the following lines:

```
KERNEL_IMAGE_BASE_NAME = "${KERNEL_IMAGETYPE}-kexecboot-${PV}-${PR}-${MACHINE}"
KERNEL_IMAGE_SYMLINK_NAME = "${KERNEL_IMAGETYPE}-kexecboot-${MACHINE}"
```

The `do_stage()` and `do_install()` steps are set to empty functions, meaning that they will have no effect: the kernel and initramfs are built only, and no other actions occur. The generated images will be written in various locations

*Where is KERNEL_IMA
set?*

Which locations?

For example, `build/tmp/deploy/glibc/images/c7x0/` dir contains:

```
Angstrom-initramfs-kexecboot-image-glibc-ipk-2009.X-test-20100106-c7x0.rootfs.cpio.gz
Angstrom-initramfs-kexecboot-image-glibc-ipk-2009.X-test-20100106-c7x0.rootfs.cpio.lzma
initramfs-kexecboot-image-c7x0.cpio.gz -> Angstrom-initramfs-kexecboot-image-glibc-ipk-2009.X-test-20100106-c7x0.rootfs.cpio.g
initramfs-kexecboot-image-c7x0.cpio.lzma -> Angstrom-initramfs-kexecboot-image-glibc-ipk-2009.X-test-20100106-c7x0.rootfs.cpio
modules-2.6.26-r12-c7x0.tgz
zImage-2.6.26-r12-c7x0.bin
zImage-c7x0.bin -> zImage-2.6.26-r12-c7x0.bin
zImage-kexecboot-2.6.26-r15-c7x0.bin
zImage-kexecboot-2.6.32-r15-c7x0.bin
zImage-kexecboot-c7x0.bin -> zImage-kexecboot-2.6.32-r15-c7x0.bin
```

If `USE_MACHINE_KERNEL` is defined when building kexecboot, `/proc/cpuinfo` is scanned for a `Hardware`: line and if present, the `Hardware` result is converted to lower case, spaces are replaced with underscores and `/mnt/boot/zImage-` is prepended to it. For example, in the case of the Sharp SL-5500 ('Collie'), `/proc/cpuinfo` has the line

`Hardware : Sharp-Collie`

The line is checked for a colon and if found, `Sharp-Collie` would be appended to the `zImage-` string, so `/mnt/boot/zImage-Sharp-Collie` is looked for.

4.5.2.2 Remarks

4.5.3 Role

A *kexecboot kernel* is, in a more precise way, a kernel that has the `kexec` feature called; this feature can load other, normal kernels in and boot into them. This allows the main kernel used in Ångström to be upgraded without flashing the device whole. The kexecboot kernel needs to be small and slim enough that it can fit into the NAND, yet be able to handle the managing RAM initdisk used to find and select the new kernel to boot off of. It should also (in minimal variant) have support all 'bootable' storages (NAND, CF, SD/MMC, etc).

Similar projects include "Two Kernel Monte" and CoreBoot.

Kexecboot development was started to solve following problems:

1. Zauruses with default bootloader and MTD partitioning have only 1.2 Mb of free space in NAND to hold kernel. Modern kernels are too big to fit into that constraints without significant loss in features (even with separate modules),
2. Some devices (such as Zauruses) have no possibility to boot from other media (SD/MMC/CF) with default bootloader.

Ångström was based on top of OE (OpenEmbedded), OpenZaurus and OpenSimpad. Zaurus' OS was the most supported platform, at that time. To address these problems, Thomas Kunze wrote first versions of kexecboot.

4.5.4 Way of Work

When kexecboot's kernel is loaded, it runs kexecboot binary as init process. This process does the following things (for the current development version, *i.e.* 0.6):

1. scan all known partitions on devices (from `/proc/partitions`),
2. check that the partition's filesystem is known (according to `fstype.c` from `klbc` and `/proc/filesystems`),
3. mount that partition and check for `/boot/boot.cfg` there:
 - (a) if `boot.cfg` is found, then parse it and do some additional things (load custom icon for example);
 - (b) if `boot.cfg` is not found, then check for default kernels (`/boot/zImage`, `/zImage`),
4. check for `zImage`, the replacement Linux kernel (compressed),
5. create and show GUI with collected partitions,
6. wait for selection and then kexec selected kernel.

Kexecboot binary can be started as non-init process too, so you can run it in your own image at any time too.

Latest kexecboot release was 0.5. The development version is 0.6.

4.5.5 Consequences

As the kexecboot kernel embeds a very limited initramfs, it contains just a couple of static binaries (kexec and kexecboot). Kexec is the tool (see kexec-tools), kexecboot is the "init," a custom-purposed binary launching kexec with appropriate args. The name "kexecboot" was given by the author of the project.

So, if you want to create a standard "liveramdisk" image, just enable kexec during the config of the kernel and deploy kexec-tools in your image.

4.5.6 Screenshots

Here are pictures (taken from [2]) of a v0.4 *kexecboot*.

(4)
Two
portable
par-
titions
from
fSDad.
card.

Figure 4.5.1: Kexecboots' screenshots.

CHAPTER 5

Different Images for Different Purposes

ACCORDING to [5], there are different types of images. They are represented at Table 5.1.

Name	Description
base-image	Very minimal system without GUI. SSH login over built-in networking (usbnet, ethernet port, ...). This is intended for headless devices with very small amount of flash (like routers) or for special-purpose installs (such as standalone portable servers, automation controllers, data loggers, etc.).
console-image	Usable system without GUI. Can connect to net via BT/WiFi/USB. Should handle devices connected via CF or USB Host.
minimal-gpe-image	Very minimal system with X11 GUI, on which you may install software you need. Suitable for devices with minuscule amount of flash and for DIY fans. NOT suitable for first-time users.
x11-image	Like console-image but with X11 environment.
x11-gpe-image	x11-image with GPE PIM suite preinstalled.
x11-pimlico-image	x11-image with OpenedHand Pimlico PIM suite preinstalled (VGA screen is recommended).
x11-office-image	x11-image with GNOME Office preinstalled (Abiword, Gnumeric, etc.).

Table 5.1: Different types of images for Ångström, for different purposes.

Images like x11-gpe and big are provided for selected devices which come with large secondary storage (like Microdrives). If there are no such images for your device, you do not lose anything: you can easily install any software you need from the feeds.

If you are in doubt which image to choose, use x11-image: it is intended as a starting point for majority of users, offering basic GUI tools to control the device and devoid of application software which not every user may need, thus offering good compromise between usability, size required, and free space available.

5.1 Formats

There are also *different formats* for each image. They are summarized at Table 5.2.

Extension	Description
.tar.gz, .tar.bz2, .cpio.gz	Archived root filesystem (<code>rootfs</code>), suitable for installing to a secondary storage by simply unpacking (the secondary storage must be already formatted appropriately, usually with ext2 filesystem)
.ext2, .ext2.bz2, .img, img.bz2	Integral filesystem image, which can be copied (e.g. using <code>dd</code>) to a secondary storage partition directly.
.jffs2	Flash image, which can be flashed to internal flash ROM of the specific device
.summary.jffs2	Same as .jffs2, but with extended “summary” information allowing to mount flash filesystem quicker. (Suggested if space allows.)
.exe	Executables for WinCE devices. Bigger files are LiveRam Disks,
zImage	Smaller files are bootmanagers (see Chapter 3, p. 8) Standalone kernel image for ad-hoc boot-loaders

Table 5.2: Different formats of images for Ångström, for different purposes.

CHAPTER 6

Hardware

UNFORTUNATELY, embedded devices vary greatly in their architecture and system software, and as of now, no generic installation procedure suitable for all of them is available.

6.1 Supported Devices

6.1.1 Installation Procedure

Here are the devices which are known to be supported:

1. Sharp Zaurus family

- (a) *kexecboot*: For the Zaurus, a special “kernel+initramfs” has been developed. This special-purpose kernel is small enough to be flashed on NAND and features a framebuffer graphical menu for the selection of boot media containing the kernel to be kexec’ed. Multi-partition cards and lots of filesystems are supported. The kernels to be launched are searched for in /boot of each available partition. There is also a configuration file (/boot/boot.cfg) which allows selection of other kernels, custom kernel command lines, and specifies the label and the icon for each instance;
- (b) *How to install*: Just unpack zaurus-installer-YOURMACHINE.tar.gz on a free (FAT or EXT2/3 formatted) partition: the package contains the kexecboot-kernel and the updater.sh utility. Then proceed as usual for flashing (switch off, pull battery out, insert battery again, before inserting AC plug, press OK and Power On. In the Japanese Menu select “4” and finally “3” for SD card or “4” for CF). See Figure 6.1.1 for a screenshot;
- (c) *For the SL-5500 (collie)* and optionally for other Zaurus too, flashing is possible just using the routines of the original firmware: reset the unit, Press “C” and “D” buttons during restart. That is specialized. For this, please read <http://linuxtogo.org/gowiki/CollieInstall>;
- (d) If you want a **rootfs in NAND**, this must be in the .jffs2 format. Just rename your-image-rootfs.jffs2 to initrd.bin and copy it on the card with updater.sh. The image will be flashed on mtd2,

2. **Sharp Zaurus family (old method, kernel in NAND)**. Please follow the following links, according to your device:
 - (a) *Sharp Spitz* (*c3xxx*): <http://www.angstrom-distribution.org/angstrom-installation-spitz>;
 - (b) *Sharp Akita* (*c1000*): <http://www.angstrom-distribution.org/c1000-install-instructions>;
 - (c) *Sharp Corgi, Shepherd, Husky, Boxer* (*c7xo-c8xo*): <http://www.angstrom-distribution.org/zaurus-c7x0-c8x0>;
 - (d) *Sharp Collie* (*SL-5500*): <http://linuxtogo.org/gowiki/CollieInstall>;
 - (e) *Sharp Poodle* (*SL-5600*): <http://linuxtogo.org/gowiki/PoodleInstall>;
 - (f) *Sharp Tosa* (*SL-6000*): <http://linuxtogo.org/gowiki/TosaInstall>,
3. **PocketPC and Windows Mobile family**,
4. **Generation6 devices** such as Archos 5 and Archos 7.

Figure 6.1.1: Sharp Zaurus family's screen.

6.1.2 List

Roughly speaking, the following devices are supported:

1. Acer n50,
2. Beagle Boards,
3. Dell Axim X50/X51,
4. HP's
 - (a) h1940;
 - (b) h2200, but you may read this: <http://www.angstrom-distribution.org/ipaq-h2200-series>;
 - (c) h3900;
 - (d) h4000;
 - (e) h63xx;
 - (f) hx2000;
 - (g) hx4700 (one of the devices amongst the most supported ones);
 - (h) rx1950;
 - (i) rx3000;
 - (j) rx3115,
5. HTC Alpine, Apache, Athena, Beetles, Blueangel, Magician, Titan, Universal,
6. iMate Jasjar,
7. Nokia 770 internet tablet, and N800 (but see <http://linuxtogo.org/gowiki/AngstromAndN800>),

8. Pandora Handheld (see <http://openpandora.org/>).

The h2000 create random problems because of WM 2005. See <http://www.handhelds.org/hypermail/hx2000-port/0/0083.html> and <http://www.handhelds.org/hypermail/hx2000-port/0/0085.html> for this. If your device is listed here: <http://linuxtogo.org/gowiki/OeDeviceList> or here: <http://linuxtogo.org/gowiki/LinuxDevices>, there may be chances that it works, as this project is closely related to OpenEmbedded.

After Installation

7.1 First Steps

7.1.1 Security Before Everything

ONCE you have installed Ångström and booted into the system, you are highly recommended to upgrade system from the feed to get the latest security and bug fixes since the release has been made. To do this, you need to run the following commands in terminal:

```
opkg update  
opkg upgrade
```

However, for this to work, you must be connected to the Internet to be able to download updates. For more info about the package manager, please have a look at Chapter 10, p. 32. You may also check the packages of Ångström's repositories at <http://www.angstrom-distribution.org/repo/>.

7.1.2 Internationalization and Localization

7.1.2.1 Two Main Definitions

Here are two definitions from [5]:

1. *Internationalization (i18n)*: Having support in the system for different character sets (both for input and output), fonts, and using well-known locale-neutral standards for representing data like time/date, and numbers,
2. *Localization (l19n)*: Having support in the system for particular locality, from time and monetary data to messages output by applications.

Ångström is targetted at the wide user community throughout the world, and consequently provides good i18n and l19n support, with basic i18n support available out of the box with standard X11 GUI images. As a contemporary Linux system, Ångström uses UTF-8 encoding. Ångström's images come with English UTF-8 locale (en_GB) which is suitable to get good i18n support. Make sure that you select "English(Great Britain)" when logging into X11 session. You select the locale for

session when logging into X, “Language” dropdown. After Ångström install, it will contain two choices: “English(C locale)” (just “English” in older builds) and “English(Great Britain).” Make sure you avoid “English(C locale)” as it does not actually offer any i18n, limiting used charset to ASCII.

To get further regarding 119n support than generic i18n above, one needs to configure proper locale for libc(C runtime library). As said earlier, Ångström’s images come with en_GB locale preinstalled (and this was apparently not the best choice, as most people would expect en_US to be default-available).

To install more libc locales, use packages with names `locale-base-LL-VV`, where LL is the language code and VV is the variant code, both converted to lowercase. For example, you may use

```
opkg install locale-base-en-us
```

Once the package has been installed, a new choice will be available in X11 login.

Some bugs are known with the Shell and the encodings.

7.1.2.2 Locale and Packages

If you want *some given packages to be in your language*, you must install locale for the corresponding language. Then, for each application you are interested in, you should install package `<app>-locale-LL` to get translated messages, where LL is the language code, as stated before.

CHAPTER 8

Bug Tracking System

For the *bug tracking system*, please use BugZilla (<http://bugs.openembedded.org/>). That is the easiest way to solve problems together.

Part III

Building

ALL Ångström images are built using OpenEmbedded. We here describe the steps (coming from [8]) which are necessary to setup an environment where you can build images and packages yourself. You may find more info for the needed packages on your host and possible tweaks (SELinux, ...) at [9].

Here are the steps you need to do:

1. Get OE metadata. Use

```
export OETREE="${HOME}/OE"
mkdir -p ${OETREE} && cd ${OETREE}
git clone git://git.openembedded.org/openembedded.git openembedded
cd openembedded
git checkout origin/stable/2009 -b stable/2009
```

2. Update OE metadata. Use

```
cd ${OETREE}/openembedded
git pull
```

3. Setup the environment.

(a) Download

<http://www.angstrom-distribution.org/files/source-me.txt>

to \${OETREE};

(b) Setup local.conf for our needs:

```
mkdir -p ${OETREE}/build/conf
cp ${OETREE}/openembedded/contrib/angstrom/local.conf ${OETREE}/build/conf/
```

Optionally, you can open \${OETREE}/build/conf/local.conf in your favourite editor and add:

```
MACHINE ?= "<your machine>"
```

where

```
"<your machine>"
```

is replaced with the machine you want to build for, such as

- i. beagleboard for the OMAP3 beagleboard.org board,
- ii. c7x0 for pxa25x zaurus clamshells (SL-C700, SL-C750, SL-C760, SL-C860, SL-7500),
- iii. spitz for pxa27x zaurus clamshells with a microdrive (SL-C3000, SL-C3100, SL-C3200),
- iv. akita for pxa27x zaurus clamshells without a microdrive (SL-C1000),
- v. tosa for the SL-C6000,
- vi. h2200 for the iPAQ h2200 series,
- vii. hx4700 for the iPAQ hx4700,
- viii. fic-gta01 for the neo1973 phone;

You may also check

<http://cgit.openembedded.org/cgit.cgi/openembedded/tree/conf/machine?h=stable/2009>.

(c) **Start building.** Use

```
# set environment variables
source source-me.txt
#Go to the OE tree
cd ${OETREE}/openembedded
#Make sure it's up to date
git pull --rebase
#Start building
# you can specify machine on the cmdline:
MACHINE=yourmachine bitbake base-image ; MACHINE=yourmachine bitbake console-image x11-image
# If you have set it in local.conf you can do:
bitbake base-image ; bitbake console-image x11-image
```

Building for a different machine or C library is just a matter of changing the MACHINE= or ANGSTROMLIBC statement in local.conf to a new value. There is ABSOLUTELY no need for using different directories for that; Angstrom takes care of all the details, it was specifically designed for this.

Part IV

The Ångström System

CHAPTER 9

Introduction

THE Ångström system¹ is a Linux distribution (distro) for embedded devices (PDAs, Beagleboards, Mobile Phones, etc.). The Ångström community was started by a group of people who worked on the OpenEmbedded, OpenZaurus and OpenSimpad projects. They aim to create a stable, user friendly linux distribution for embedded devices.

Using Linux on an embedded device is an interesting thing. Users of such devices are often unsatisfied by the bad quality of the built-in OS'es and softwares on their embedded devices. Or a Linux *aficionados* is always unhappy to use something else than an UNIX-based system. The advantage of UNIX-based systems is well-known, and does not enter in the scope of this booklet. For these reasons, there are now Linux-based distros for embedded devices.

¹ And not *OpenEmbedded* Ångström!

CHAPTER 10

Packaging System

Some info comes from [5].

10.1 Introduction

As any modern distribution, Ångström is fully package-based. However, as interaction with the device during initial setup phase is difficult, Ångström is bootstrapped using an “image,” which is essentially a set of core packages already merged into archive or filesystem image. This image is installed using device-specific means, and provides basic Ångström functionality.

Adhering with the Ångström’s aim of flexibility and best practices of Linux system management, the images provide only core packages, allowing users to customize system to their needs by easily installing any required software from a wealth of Ångström’s feeds. However, as a convenience to users, different kinds of images are provided, for example, with (core packages of) different GUI environments pre-installed.

10.2 Core Principle

The *packaging system* was firstly based on ipkg (Itsy Packaging system), which was developed a long time ago for the Itsy handheld project, by Compaq. The original ipkg may¹ have been developed by Kernel Concepts (<http://www.kernelconcepts.de>). For example, its .conf files are stored in /etc/ipkg/. When that project closed down and <http://www.handhelds.org> came to be, ipkg was ported to the ‘Familiar’ system, running on the iPAQ hx4700. opkg appears to have been developed by OpenedHand Ltd. (<http://www.openedhand.com>) The home of opkg is <http://opkg.googlecode.com>. The preconfigured feeds are located at /etc/opkg, and, according to [?], opkg is replacing ipkg, for different reasons; opkg has the following advantages:

- maintained, when ipkg is not,
- two companies behind it,

¹ According to a comment in src/opkg-frontend.c in opkg-0.1.7.

- GPG signed packages and feeds.

Consequently, we shall speak only about opkg here.

The feeds for the 2007 image are located at

<http://www.angstrom-distribution.org/feeds/2007/>.

10.3 Commands

- To look for a given package, you may have a look at

`http://www.angstrom-distribution.org/repo/`

if you do not have an Internet access on your device,

- To update the list of the packages which is on your device, you could use

`opkg update`

- To automatically upgrade every package which needs to be upgraded on your device, you could use

`opkg upgrade`

- To see the log file of ipkg, you may look in

`/var/log/opkgupgrade.log`

- To look for a packagename package, you could use

`opkg search packagename`

(Regexp are also accepted),

- To look for what is available as packages, you could use

`opkg list`

- Once you know the name of the package you want to install, such as packagename, you then use

`opkg install packagename`

to install it. If you have it on your disk, you can then use

`opkg install dir1/dir2/.../packagename.ipk`

Be careful with dependencies,

- To remove a package, you can use

`opkg remove packagename`

If you want all its dependencies to be removed too, you then use

`opkg remove --recursive packagename`

being very careful.

10.4 Selecting Packages

As with any Linux distro, there are oftentimes different packages available to perform some given task. However, they often require different levels of resources (e.g. command line mail vs. GUI, command line mp3 playback vs. XMMS); the key thing with Ångström is whether you have opted for any GUI support at all (GPE, Opie or Qtopia) or whether you are strictly using command line. The key therefore is to read the package description and do not use packages with the word "opie" in them when you are GPE based and *vice-versa!*

10.5 Format of Packages

The software packages are stored in files whose name ends by .ipk; this is a compressed format, containing files to be installed as well as metadata and support scripts. Actually, the .ipk format is the *de facto* package format for Embedded Linux, and many other distros use it for their packages too. However, packages from different systems are not compatible. In particular, Ångström maintainers cannot guarantee system stability if packages which do not come from the official feeds are used.

10.6 Switch from Console Image to a X Server

If you are using a Console Image, you may install, once you have updated your lists of packages, `xserver-kdrive`, or `xserver-kdrive-imageon` (for Zaurus C7xo image users). Next, install `task-base-extended`, `angstrom-x11-base-depends`, `angstrom-gpe-task-base`, `angstrom-gpe-task-settings`, `libesd0`. It will give you a fully working X GUI and a base GPE environment.

Use

11.1 The User

ONCE you have installed Ångström, you can modify your owner's info by editing
`/etc/gpe/gpe-ownerinfo.data`

11.2 Checking the Version

You can check the current Ångström version you are running by reading the content of (the file):

`/etc/angstrom-version`

You can also use

`uname -a`

just like you would do on every normal UNIX station.

11.3 Global Networking

Some info comes from [5].

One of the first things you want to do after installing an Ångström release is to connect to Ångström's website and get the latest updates for security and important bug fixes. If you connect to the Internet "directly" (for example, via GPRS) or via a dedicated router (for example, using WiFi or Ethernet), you usually do not need to do additional configuration "on the other end," assuming your GPRS account is enabled and router already setup to serve other computers.

11.3.1 Basic Info

You may use, to

- List available interfaces:

`ifconfig`

- Bring interface `interfacename` up:

`ifconfig interfacename up`

or

`ifup interfacename`

- Bring interface `interfacename` down:

`ifconfig interfacename down`

or

`ifdown interfacename`

11.3.2 With USB

However, many people with conventional PDA devices will connect using USB to the desktop or laptop computer. In this scenario, your host computer will serve as a router for your Ångström device, and so must be configured accordingly: you must enable IP forwarding and NAT on it. Please consult your Linux distribution manual on how to make such changes permanent, to be able to connect the device to the Internet without manual setup in the future. We here assume that you are using a Linux server. If you are using a Windows one, you need to enable ICS(Internet Connection Sharing) to achieve the similar functionality. If you are already using ICS for a different interface than the one you projected to work with, consider the use of a simple proxy such as AnalogX's, for giving other cards access to the Internet. It is due to the fact that ICS can only deal with a couple (of two) devices.

11.3.2.1 On the Server

To activate NAT, you could do

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

assuming `eth0` is your external interface.

To activate IP forwarding, you then edit `/etc/sysctl.conf`, adding

```
net.ipv4.ip_forward = 1
```

to it, or replacing

```
net.ipv4.ip_forward = 0
```

by

```
net.ipv4.ip_forward = 1
```

if

```
net.ipv4.ip_forward = 0
```

is already there. You can then reload the config using

```
sysctl -p /etc/sysctl.conf
```

which should show you the line

```
net.ipv4.ip_forward = 1
```

(there could be other lines before, evidently).

You then need to load the modules `usbnet`, `cdc_ether` and `g_ether` (the last one being done if your host is also a device). You also need to add an entry to `/etc/network/interfaces` like

```
iface usb0 inet static
    address 192.168.129.1
    network 192.168.129.0
    netmask 255.255.255.0
    broadcast 192.168.129.255
```

if you are running Debian, or

```
config_usb0=( "192.168.129.1" )
```

to `/etc/conf.d/net` if you are using Gentoo, or even create a file

```
/etc/sysconfig/network/ifcfg-usb0
```

with

```
BOOTPROTO='static'
STARTMODE='auto'
USERCONTROL='yes'
NAME='Sharp Zaurus'
IPADDR='192.168.129.1'
BROADCAST='192.168.129.255'
NETMASK='255.255.255.0'
NETWORK='192.168.129.0'
PREFIXLEN=''
```

if you use SuSE.

11.3.2.2 On the Slave

On the slave (the computer running Ångström), you will then

```
modprobe g_ether
```

(if the command

```
lsmod | grep g_ether
```

gives nothing as result). Consequently, using then

```
ifconfig -a
```

should display some devices, but at least `usb0`. If necessary, you may run

```
ifup usb0
```

(if `usb0` is already there, no need¹ for it). You can then modify the IP stuff with either the GUI of your front-end (X11, GPE, ...), or by appending

```
iface usb0 inet static
    pre-up modprobe g_ether
    address 192.168.129.201
    netmask 255.255.255.0
    gateway 192.168.129.1
    post-down rmmod g_ether
```

for example, to your `/etc/network/interfaces`.

11.3.3 With Bluetooth

Ångström uses the standard 2.6 kernel's Bluetooth modules (which were derived from <http://www.bluez.org/>). Userspace commands used to control/access Bluetooth hardware are `hciconfig`, `hcitool`, `rfcomm`, `l2ping` and other ones. (It is also implemented in GUIs.) Before establishing a Bluetooth connection with another computer, either for `rfcomm` or `pand`, it is necessary to pair the two. Bluetooth uses a PIN, or password, for incoming pairing requests. This one is set in the `/etc/bluetooth/pin` file, and for outgoing requests by the utility "passkey-agent."

There are two ways to make a network using Bluetooth: one is to use point-to-point protocol (PPP) over the `rfcomm`(serial-like link); the other is to use PAN. This is a very brief summary to give you an understanding of the processes involved:

1. PPP over `rfcomm` was previously popular with PDAs like Palms and some WindowsMobile/PocketPC/WinCE devices. It is not so efficient, but is useful if you are connecting the device to a Windows computer. On the Ångström device you need to create the appropriate files in `/etc/ppp` and then trigger the PPP connection using the command

```
pppd call XXX
```

(`XXX` being the profile); this then results in a `ppp0` device appearing.

If the called device is correctly set up, the `ppp0` device will end up with the correct IP address and default route, and the local DNS resolver will probably also be set correctly, otherwise use

```
ifconfig ppp0
route add default gw x.x.x.x
```

to tweak settings,

2. PAN/PAND is best for making a Linux-to-Linux connection. Simply run

```
pand -listen
```

on one end, and

¹Note that issuing `ifup usb0` would do nothing, in this particular case: it would not duplicate a device!

```
pand -connect xx:xx:xx:xx:xx:xx
```

on the other where the xx's are the parts of the MAC address of the other device (use

```
hciconfig -a
```

to find out the MAC address). When connected, use

```
ifconfig bnep0
```

to set the IP address. Use

```
route add
```

to make routes (*e.g.*

```
route add default gw 192.168.129.1
```

), just as with any other network connection. You might also need to tweak /etc/resolv.conf on the Ångström device so it can resolve names into IP addresses. It should be possible to create a bnep0 entry in /etc/network/interfaces for this.

In both cases, it is likely that you are using the other end of the Bluetooth connection as a gateway to allow the Ångström device to access the internet (*e.g.* access mail, browse the web, install from software feeds). If the remote box is running Linux you may want to set up masquerading on its external/outbound interface and thus allow the Ångström device to access hosts and networks other than the gateway. Do not forget neither to turn on ipv4forwarding just as in Subsubsection 11.3.2.1, p. 36. The equivalent in Windows is “Internet Connection Sharing,” accessed through the properties for the external interface.

11.3.4 With WiFi

11.3.4.1 Preliminaries

On some devices, you may need to load specific modules to allow WiFi. It depends on the chipsets.

- For the hx4700, you need to type

```
modprobe hx4700-acx
modprobe acx debug=0
```

in a terminal before being able to use WiFi. If you do not want to modprobe specific modules manually, you can add them in /etc/modules, just like in every normal UNIX system.

11.3.4.2 Commands

Commands for WiFi are supposed to be known (`iwconfig`, `iwlist`, ...), as they are universal. Here are some basic commands for you, if you are too lazy to have a look at `man` pages:

- **List networks:**

```
iwlist interface scan
```

where `scan` can be replaced by `scanning` and where `interface` is often `wlan0`,

- **Connect to network** with a ESSID `essidname`:

```
iwconfig interface essid essidname
```

As in the other methods, you are also able to **bring up the interface** issuing

```
ifup interface
```

and, to bring it down,

```
ifdown interface
```

where `interface` is often `wlan0` (it can be verified using `ifconfig`). If `iwconfig` and related commands are missing (your `$PATH` being correct, or after having tried with `/sbin/iwconfig`), be sure to install the `wireless-tools` package.

For WEP/WPA/PSK, you need to edit `/etc/network/interfaces`, as explained in the following lines.

11.3.4.3 WEP

Here is an example for *WEP encryption*:

```
iface wlan0 inet dhcp
    wireless_mode managed
    wireless_essid youressid
    wireless_key yourwepkey
```

11.3.4.4 WPA

Here is an example for *WPA/PSK encryption*:

```
iface wlan0 inet dhcp
# start wpa_supplicant
    wpa-conf /etc/wpa_supplicant.conf
    wpa-driver hostap
```

If `wpa_supplicant.conf` does not exist in `/etc`, create one with

```
#/etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=1
ap_scan=1
fast_reauth=1
network={
    ssid="(your SSID)"
    proto=WPA
    pairwise=TKIP
    psk= (quoted text or hexadecimal number)
    priority=1
}
```

(You may have a look at http://linux.die.net/man/5/wpa_supplicant.conf.) You also need to install the package `wpa_supplicant`.

11.3.4.5 Drivers

There are roughly two different drivers:

1. `hostap`: supports WEP and WPA,
2. `acx`: supports WEP.

11.3.5 Firewall

Just like in every normal Linux distro, you may use the following trivial firewall:

```
iptables -P OUTPUT ACCEPT
iptables -P INPUT DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

To make this happen on boot-up, you may create a script with these lines, put it into `/etc/init.d/firewall`, and make soft links to the script from the `rc3.d` and `rc5.d` directories.

11.4 Errors You Could Encounter

It is possible that you *encounter different errors*, such as, in the console:

`xxx: command not found.`

In this case, please check that `/sbin/` is in the environment variables. If it is not the case, simply add it, or always use `/sbin/` before each app, *i.e.*

`/sbin/modprobe mymodule`

or

`/sbin/ifconfig myadapter`

If you do not want to do this every time you want to use such a command, you may simply add `sbin` to `$PATH`(it is actually only necessary if you are using a liveram disk):

`PATH=$PATH:/sbin`

`export PATH`

CHAPTER 12

Getting More Help

THE objective of this booklet was to provide an easy way to find fresh info about Ångström. It is particularly difficult to find *recent* info on such a subject when the community is not really big. Fortunately, despite being a small community, users are active. The principal concepts are evoked here, and feel free to send an e-mail to any contributor (*and* to me) if there are any interesting things to say which could be nice to put here. That being said, there are other websites which you can find any other info on. For example, one can find some interesting information in [5]. There is also the principal <http://www.angstrom-distribution.org/> website, and the mailinglist (see <http://www.angstrom-distribution.org/contact> and <http://lists.linuxtogo.org/pipermail/angstrom-distro-users/> for this).

Index

.cpio.gz, 20
.exe, 20
.ext2, 20
.ext2.bz2, 20
.img, 20
.img.bz2, 20
.jffs2, 20
.summary.jffs2, 20
.tar.bz2, 20
.tar.gz, 20

base-image, 19
bootbundle, 14
bootmanager, 20
BugZilla, 26

console-image, 19

HaRET, 13, 14

i18n, 24
initramfs.cpio.gz, 13, 17
Initrd, *see* initramfs.cpio.gz
initrd, *see* initramfs.cpio.gz

kexecboot, 15

l19n, 24
LiveRam Disk, 10, 11, 20
Loopback image, 8

minimal-gpe-image, 19

Startup script, *see* startup.txt.bootimage
startup.txt.bootimage, 13, 14

x11-gpe-image, 19
x11-image, 19

Bibliography

- [1] BITBAKE, *BitBake (Example 4.7)*, 2010. <http://bitbake.berlios.de/manual/ch04s02.html#id870480>.
- [2] Y. BUSHMELEV, *Picasa Albums Web – Jay*. <http://picasaweb.google.ru/yuri.bushmelev>, 2009.
- [3] C. COMMONS, *Creative Commons – Attribution-Share Alike 3.0 Unported*, 2009. <http://creativecommons.org/licenses/by-sa/3.0/>.
- [4] ——, *Creative Commons Legal Code: BY-SA*, 2009. <http://creativecommons.org/licenses/by-sa/3.0/legalcode>.
- [5] LINUXToGo, *Ångström Manual*, 2009. <http://www.linuxtogo.org/gowiki/AngstromManual>.
- [6] ——, *WinCeQuickInstall*, 2009. <http://linuxtogo.org/gowiki/WinCeQuickInstall>.
- [7] L. MERCIADRI, *Using Linux on a Handheld Device: an Obstacle Course?* 2009.
- [8] OPENEMBEDDED, *Building Ångström — The Ångström Distribution*, 2006. <http://www.angstrom-distribution.org/building-angstrom>.
- [9] ——, *OEandYourDistro - OpenEmbedded*, 2010. <http://wiki.openembedded.org/index.php/OEandYourDistro>.
- [10] T. S. PRESS, *Chapter 3 – How Linux Boots*, 2004. http://nostarch.com/download/howlinuxworks_ch3.pdf.

Parte VII

Bibliografía

Bibliografía

- [1] Antenna Circuit Design for RFID Applications, AN710.
- [2] HF Antenna Design Notes, Literature Number: 11-08-26-003.
- [3] HF Antenna Cook Book, Lit. Number 11-08-26-001.
- [4] Micore Reader IC Family; Directly Matched Antenna Design, Rev. 2.05, 10 May 2006.
- [5] Mifare®(14443A) 13,56 MHz RFID Proximity Antennas, Application Note.
- [6] MIFARE Type Identification Procedure; AN10833 Rev. 3.1 — 07 July 2009.
- [7] MIFARE ISO/IEC 14443 PICC Selection; AN10834 Rev. 3.0 — 26 June 2009.
- [8] MIFARE Classic Commands; AN10856 Rev. 1.0 — 31 July 2009.
- [9] MIFARE and handling of UIDs; AN10927 Rev. 2.0 — 01 September 2010.
- [10] Mainstream contactless smart card IC for fast and easy; MF1S5009 Rev. 3 — 27 July 2010.
- [11] MIFARE Standard Card IC MF1 IC S50 Functional Specification; Revision 5.1 May 2001.
- [12] MIFARE® and I Code CL RC632 Multiple protocol contactless reader IC.
- [13] 8-BIT BIDIRECTIONAL VOLTAGE-LEVEL TRANSLATOR.
- [14] HD44780U (LCD-II), Dot Matrix Liquid Crystal Display Controller/Driver, Rev. 0.0 1998.

- [15] LM1117/LM1117I 800mA Low-Dropout Linear Regulator.
- [16] LCM-S01602DSF/A - hoja de datos del LCD16x2.
- [17] BeagleBoard System Reference Manual Rev C4, Revision 0.0.
- [18] Smart Card Handbook, Third Edition, Wolfgang Rankl and Wolfgang Effing, 2003, ISBN: 0-470-85668-8.
- [19] RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition, Klaus Finkenzeller, 2003, ISBN: 0-470-84402-7.
- [20] A Practical Attack on the MIFARE Classic, Gerhard de Koning Gans, Jaap-Henk Hoepman and Flavio D. Garcia, 2008.
- [21] Angström Manual - Embedded Power -, Luca Merciadri and Kooi Koën, 29 June 2010.
- [22] Application Independent CardTerminal Application Programming Interface for ICC Applications; Deutsche Telekom AG / PZ Telesec, GMD - Forschungszentrum Informationstechnik GmbH, TÜV Informationstechnik GmbH, TELETRUST Deutschland e.V., 30.10.1996.
- [23] Application independent CardTerminal Basic Command Set for ICC applications; TeleTrusT Deutschland e.V., GMD Forschungszentrum Informationstechnik GmbH, Version 0.9, 28.07.1995.
- [24] IFDHandler 3.0 MUSCLE PC/SC IFD Driver API; David Corcoran & Ludovic Rousseau, July 28, 2004.
- [25] Diseño y desarrollo de circuitos impresos con KiCad, Miguel Pareja Aparicio, 2010, ISBN: 978-84-937769-1-6.
- [26] El lenguaje de programación C, Brian W. Kernighan y Dennis M. Ritchie, 1991, ISBN: 968-880-205-0.
- [27] Norma ISO/IEC 7816.
- [28] Norma ISO/IEC 14443.

- [29] OpenPCD, <http://www.openpcd.org/>, agosto 2011.
- [30] librfid, <http://openmrtd.org/projects/librfid/>, agosto 2011.
- [31] GES9G20, http://www.gloimationinc.com/product_9G20.html, agosto 2011.
- [32] Hawkboard, <http://www.hawkboard.org/>, agosto 2011.
- [33] Página oficial Beagleboard, <http://beagleboard.org/>, agosto 2011.
- [34] Foro de Beagleboard, <https://groups.google.com/forum/#!forum/beagleboard>, agosto 2011.
- [35] Consumo de Beagleboard, http://elinux.org/BeagleBoardFAQ#Power_usage, agosto 2011.
- [36] Consumo de Beagleboard, <http://www.h-online.com/news/forum/S-Power-consumption-under-2W/forum-110345/msg-14369911/read/>, agosto 2011.
- [37] Openembedded-Bitbake, http://wiki.openembedded.net/index.php/BitBake_%28dev%29, agosto 2011.
- [38] KiCad, <http://www.lis.inpg.fr/realise.au.lis/kicad/>, agosto 2011.
- [39] gEDA, <http://www.geda.org/index.html>, agosto 2011.
- [40] Proxmark, <http://www.proxmark.org/proxmark>, agosto 2011.
- [41] Angström, <http://www.angstrom-distribution.org/>, agosto 2011.
- [42] MLO, <http://www.angstrom-distribution.org/demo/beagleboard/MLO/>, agosto 2011.
- [43] Narcissus, <http://narcissus.angstrom-distribution.org/>, agosto 2011.
- [44] termios, <http://pubs.opengroup.org/onlinepubs/007908799/xsh/termios.h.html>, agosto 2011.
- [45] Proceso de booteo de linux, <http://www.ibm.com/developerworks/linux/library/l-linuxboot/index.html>, agosto 2011.

- [46] Make, <http://es.wikipedia.org/wiki/Make>, agosto 2011.
- [47] DDD, <http://www.gnu.org/s/ddd/>, agosto 2011.
- [48] GPIO, http://www.avrfreaks.net/wiki/index.php/Documentation:Linux/GPIO#Interfaces_explained, agosto 2011.
- [49] Kernel GPIO, <http://kernel.org/doc/Documentation/gpio.txt>, agosto 2011.
- [50] Configuración SDK, <http://www.electrosonradio.com/2011/04/intro-to-basic-cross-compiling-for-the-beagleboard/>, agosto 2011.
- [51] Grupo mina - INCO, <http://www.fing.edu.uy/inco/grupos/mina/>, agosto 2011.
- [52] LCD4Linux, <http://ssl.bulix.org/projects/lcd4linux/>, agosto 2011.
- [53] Foro Texas Instruments, http://e2e.ti.com/support/interface/etc_interface/f/391/t/114719.aspx, agosto 2011.
- [54] Audiofingerprint, http://iie.fing.edu.uy/cursos/file.php/14/proyectos/2009/audio_fingerprint/index.html, agosto 2011.
- [55] Tarjetas inteligentes, http://es.wikipedia.org/wiki/Tarjeta_inteligente, agosto 2011.
- [56] Beagleboards beginners, <https://groups.google.com/forum/#!forum/beagleboard>, agosto 2011.
- [57] Desarrollo en Beagleboard, <http://members.cox.net/ebrombaugh1/embedded/beagle/index.html>, agosto 2011.
- [58] Wiki Beagleboard, <http://code.google.com/p/beagleboard/w/list>, agosto 2011.
- [59] NAND de la Beagleboard, <http://elinux.org/BeagleBoardNAND>, agosto 2011.
- [60] Croscompilación en Beagleboard, <http://garrenblog.blogspot.com/2008/10/adventures-in-beagleboard-land-compiler.html>, agosto 2011.
- [61] Primeros pasos Beagleboard, <http://www.mipixel.com/electronica/beagleboard>, agosto 2011.

- [62] Arranque de Linux en la Beagleboard, <http://www.ibm.com/developerworks/library/l-beagle-board/index.html>, agosto 2011.
- [63] Internet en Beagleboard, <http://thoughtshubham.blogspot.com/2010/03/internet-over-usb-otg-on-beagleboard.html>, agosto 2011.
- [64] OpenEmbedded en Ubuntu 10.04, <http://maltanar.blogspot.com/2010/05/openembedded-on-ubuntu-1004.html>, agosto 2011.
- [65] Solución de problemas con Beagleboard, <http://www.electronsonradio.com/category/bb/>, agosto 2011.
- [66] Foro Beagleboard - Testeo SPI, http://groups.google.com/group/beagleboard/browse_thread/thread/d6a4e0703033cada%7CSPI, agosto 2011.
- [67] SPI en Beagleboard, <http://elinux.org/BeagleBoard/SPI>, agosto 2011.
- [68] Parche SPI en Beagleboard con kernel 2.6.32, <http://elinux.org/BeagleBoard/SPIPatch-2.6.32>, agosto 2011.
- [69] Foro Beagleboard – SPI, <https://groups.google.com/forum/#!topic/beagleboard/p7Ua8jmILkM>, agosto 2011.
- [70] Foro Beagleboard - Configuración SPI, <https://groups.google.com/forum/#!topic/beagleboard/QpiPDhTbDwE>, agosto 2011.
- [71] Foro Beagleboard - Configuración SPI, <https://groups.google.com/forum/#!topic/beagleboard/3v3qsRQNQzA>, agosto 2011.
- [72] Foro Beagleboard - Aplicar parche SPI, [https://groups.google.com/forum/#!searchin/beagleboard/%22bitbake\\$20-c\\$20rebuild%22/beagleboard/TF6TCJxXL1Y/joHWHiPZEJ](https://groups.google.com/forum/#!searchin/beagleboard/%22bitbake$20-c$20rebuild%22/beagleboard/TF6TCJxXL1Y/joHWHiPZEJ), agosto 2011.
- [73] Foro Beagleboard - SPI en Beagleboard, [https://groups.google.com/forum/#!searchin/beagleboard/applying\\$20patches/beagleboard/s340Ss-GoJU/KAeSfH8vqb8J](https://groups.google.com/forum/#!searchin/beagleboard/applying$20patches/beagleboard/s340Ss-GoJU/KAeSfH8vqb8J), agosto 2011.
- [74] Foro Beagleboard - Problemas con SPI, https://groups.google.com/forum/?show_docid=9db5bcf44392d4a0#!topic/beagleboard/MgzrowFy-6U, agosto 2011.

- [75] Foro Beagleboard - Construyendo un kernel y habilitar SPI, <https://groups.google.com/forum/#topic/beagleboard/XrMz-9Byum0>, agosto 2011.
- [76] Foro Beagleboard - SPI en Beagleboard, <https://groups.google.com/forum/#topic/beagleboard/s340Ss-GoJU>, agosto 2011.
- [77] Foro Beagleboard - Compilación de kernel, https://groups.google.com/forum/?pli=1#!searchin/beagleboard/mkimage/beagleboard/Od_tCEHwUvk/DxoPO8Yv9t0J, agosto 2011.
- [78] Foro Beagleboard - Problemas construyendo uImage, <https://groups.google.com/forum/?pli=1#!msg/beagleboard/MDxilRJKIIQ/r0RVzZdIdYsJ>, agosto 2011.
- [79] Foro Beagleboard - Sobre las versiones generadas de uImage, <https://groups.google.com/forum/#topic/beagleboard/DR-ruujKu9U>, agosto 2011.
- [80] Foro Beagleboard - Aplicar parche al kernel linux, [https://groups.google.com/forum/#topic/beagleboard/applying\\$20patches/beagleboard/bAjYSVKAyAM/jCTXIbfn40wJ](https://groups.google.com/forum/#topic/beagleboard/applying$20patches/beagleboard/bAjYSVKAyAM/jCTXIbfn40wJ), agosto 2011.
- [81] Openmbedded, http://www.openembedded.org/index.php/Main_Page, agosto 2011.
- [82] OpenEmbedded , uso, <http://www.uv-ac.de/openembedded/>, agosto 2011.
- [83] Foro Beagleboard - configuración de kernel con OpenEmbedded, https://groups.google.com/forum/#msg/beagleboard/UIL4e0_O_UE/_NefIFsj7pkJ, agosto 2011.
- [84] Bitbake, tutorial, <http://www.gumstix.org/software-development/open-embedded/160-bitbake-tutorial.html>, agosto 2011.
- [85] Bitbake, tutorial, <http://bitbake.berlios.de/manual/>, agosto 2011.
- [86] Foro Beagleboard - OpenEmbedded/Bitbake, como forzar regeneración de uImage, https://groups.google.com/forum/?pli=1#!searchin/beagleboard/uimage/beagleboard/FibI_LXNVGk/mNHp6ZPNOAYJ, agosto 2011.

- [87] Foro Beagleboard - U-boot, fuentes, [https://groups.google.com/forum/#!searchin/beagleboard/u-boot\\$20source/beagleboard/XD82Fd54ABg/xCUaE5W_OwQJ](https://groups.google.com/forum/#searchin/beagleboard/u-boot$20source/beagleboard/XD82Fd54ABg/xCUaE5W_OwQJ), agosto 2011.
- [88] Beagleboard, desarrollo de software, <http://wh1t3s.com/tag/beagleboard/>, agosto 2011.
- [89] Foro Beagleboard - Línea de 1.8V en el bloque de expansión, [https://groups.google.com/forum/#!topic/beagleboard/00urlSRNrXA](https://groups.google.com/forum/#topic/beagleboard/00urlSRNrXA), agosto 2011.
- [90] PinMux en el bloque de expansión, <http://elinux.org/BeagleBoardPinMux>, agosto 2011.
- [91] Guía de programación Serial, <http://www.easysw.com/~mike/serial/serial.html>, agosto 2011.
- [92] Chequear y utilizar puertos seriales bajo Linux, <http://www.cyberciti.biz/faq/find-out-linux-serial-ports-with-setserial/>, agosto 2011.
- [93] Programación Serial, <http://tldp.org/HOWTO/Serial-Programming-HOWTO/x115.html>, agosto 2011.
- [94] Programación Serial, <http://www.linux.com/learn/docs/ldp/745-Serial-Programming-HOWTO>, agosto 2011.
- [95] Interfaz GPIO, <http://www.avrfreaks.net/wiki/index.php/Documentation:Linux/GPIO>, agosto 2011.
- [96] Encendido y apagado de pines de la Beagleboard, <http://blog.makezine.com/archive/2009/02/blinking-leds-with-the-beagle-board.html?CMP=OTC-0D6B48984890>, agosto 2011.
- [97] Uso de diff y patch, <http://www.linuxtutorialblog.com/post/introduction-using-diff-and-patch-tutorial>, agosto 2011.
- [98] CodeSourcery 2009q3-67, <http://www.codesourcery.com/sgpp/lite/arm/portal/release1039>, agosto 2011.

- [99] Librfid, instalación, <http://wiki.yobi.be/wiki/RFID#librfid>, agosto 2011.
- [100] Proyectos con Beagleboard, <http://www.hy-research.com/projects.html>, agosto 2011.
- [101] PCSC-Lite, <http://pcsclite.alioth.debian.org/>, agosto 2011.
- [102] PCSC Workgroup, <http://www.pcscworkgroup.com/>, agosto 2011.
- [103] Descargas PCSC-Lite, https://alioth.debian.org/frs/?group_id=30105, agosto 2011.
- [104] <http://ludovic.rousseau.free.fr/softwares/pcsc-tools/>, agosto 2011.
- [105] Blog de Ludovic Rousseau - autor de PCSC-Lite, <http://ludovicrousseau.blogspot.com/>, agosto 2011.
- [106] Newark, <http://www.newark.com/>, agosto 2011.
- [107] Digikey, <http://www.digikey.com/>, agosto 2011.
- [108] Special Computing, <http://specialcomp.com/>, agosto 2011.