

Advanced Electronic Systems, Bengaluru

ALS-SDA-ARMCTXM3-01

ARM Cortex M3 Evaluation Board

LPC1768 is an ARM Cortex M3 Based microcontroller

On board hardwares related to many onchip peripherals

Ethernet MAC, USB Host, USB Device, CAN_I2S etc

2014



USER MANUAL

CONTENTS

1. INTRODUCTION TO ARM CORTEX M3 PROCESSORS

- 1.1 Introduction
- 1.2 Background of ARM architecture
- 1.3 Architecture versions
- 1.4 Instruction Set Development
- 1.5 Advantages of Cortex M3 processors
- 1.6 Applications of ARM Cortex M3 processors
- 1.7 The Cortex-M3 Processor versus Cortex-M3-Based Micro Controllers

2. INTRODUCTION TO MICRO CONTROLLER LPC1768

- 2.1 Architectural Overview
- 2.2 Block Diagram of LPC1768FBD100
- 2.3 A brief description of the blocks

3. TECHNICAL SPECIFICATIONS

- 3.1 Specifications of LPC1768
- 3.2 Specifications of the product

4. HARDWARE DESCRIPTION

- 4.1 Block diagram
- 4.2 Description of the different blocks

5. PLACEMENT DIAGRAMS

6. SOFTWARE/FIRMWARE

- 6.1 KEIL uvision4 IDE Installation
- 6.2 Flash magic 6.01 installation
- 6.3 Project Creation in KEIL uvision4 IDE

7. TEST SET UP & TEST PROCEDURE

- 7.1 Test set up requirements
- 7.2 Test set up
- 7.3 Flash programming
- 7.4 Testing of sample programs

8. TROUBLE SHOOTING

9. BIBILOGRAPHY

USER MANUAL

1. INTRODUCTION TO ARM Cortex M3 PROCESSOR

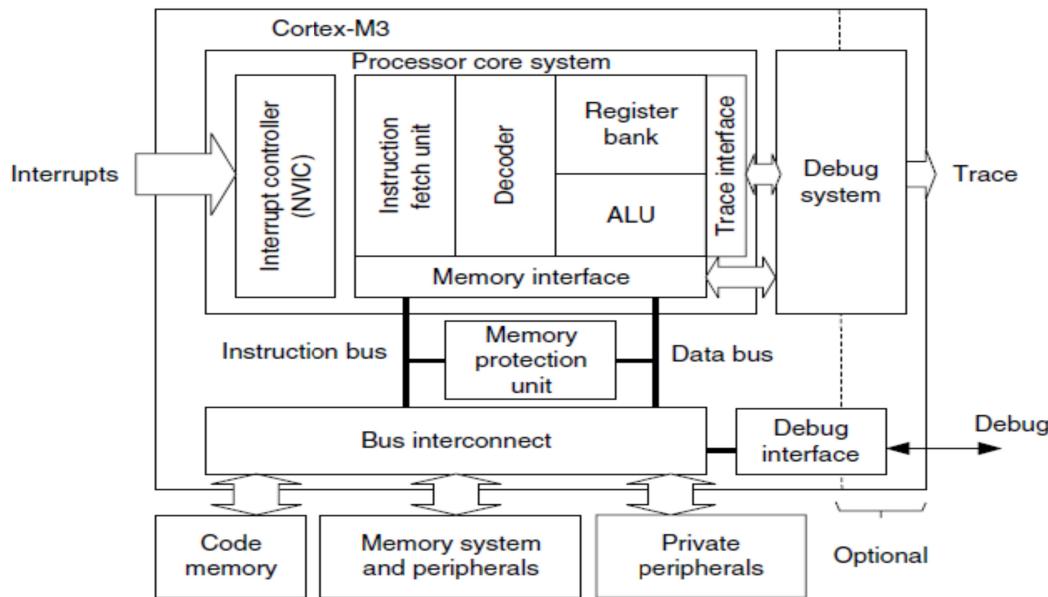
1.1 Introduction

The ARM Cortex-M3 is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The Cortex-M3 offers many new features, including a Thumb-2 instruction set, low interrupt latency, hardware divide, interruptible/continuable multiple load and store instructions, automatic state save and restore for interrupts, tightly integrated interrupt controller with Wake-up Interrupt Controller, and multiple core buses capable of simultaneous accesses.

Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory.

The processor has a Harvard architecture, which means that it has a separate instruction bus and data bus. This allows instructions and data accesses to take place at the same time, and as a result of this, the performance of the processor increases because data accesses do not affect the instruction pipeline. This feature results in multiple bus interfaces on Cortex-M3, each with optimized usage and the ability to be used simultaneously. However, the instruction and data buses share the same memory space (a unified memory system). In other words, you cannot get 8 GB of memory space just because you have separate bus interfaces.

A simplified block diagram of the Cortex-m3 architecture is shown below



It is worthwhile highlighting that the Cortex-M3 processor is not the first ARM processor to be used to create generic micro controllers. The venerable ARM7 processor has been very successful in this market, The Cortex-M3 processor builds on the success of the ARM7 processor to deliver devices that are significantly easier to program and debug and yet deliver a higher processing capability.

USER MANUAL

1.2 Background of ARM architecture

ARM was formed in 1990 as Advanced RISC Machines Ltd., a joint venture of Apple Computer, Acorn Computer Group, and VLSI Technology. In 1991, ARM introduced the ARM6 processor family, and VLSI became the initial licensee. Subsequently, additional companies, including Texas Instruments, NEC, Sharp, and ST Microelectronics, licensed the ARM processor designs, extending the applications of ARM processors into mobile phones, computer hard disks, personal digital assistants (PDAs), home entertainment systems, and many other consumer products.

Nowadays, ARM partners ship in excess of 2 billion ARM processors each year. Unlike many semiconductor companies, ARM does not manufacture processors or sell the chips directly. Instead, ARM licenses the processor designs to business partners, including a majority of the world's leading semiconductor companies. Based on the ARM low-cost and power-efficient processor designs, these partners create their processors, micro controllers, and system-on-chip solutions. This business model is commonly called intellectual property (IP) licensing.

1.3 Architecture versions

Over the years, ARM has continued to develop new processors and system blocks. These include the popular ARM7TDMI processor and, more recently, the ARM1176TZ(F)-S processor, which is used in high-end applications such as smart phones. The evolution of features and enhancements to the processors over time has led to successive versions of the ARM architecture. Note that architecture version numbers are independent from processor names. For example, the ARM7TDMI processor is based on the ARMv4T architecture (the *T* is for *Thumb* instruction mode support).

The ARMv5E architecture was introduced with the ARM9E processor families, including the ARM926E-S and ARM946E-S processors. This architecture added "Enhanced" Digital Signal Processing (DSP) instructions for multimedia applications. With the arrival of the ARM11 processor family, the architecture was extended to the ARMv6. New features in this architecture included memory system features and Single Instruction–Multiple Data (SIMD) instructions. Processors based on the ARMv6 architecture include the ARM1136J(F)-S, the ARM1156T2(F)-S, and the ARM1176JZ(F)-S.

Over the past several years, ARM extended its product portfolio by diversifying its CPU development, which resulted in the architecture version 7 or v7. In this version, the architecture design is divided into three profiles:

- The **A profile** is designed for high-performance open application platforms.
- The **R profile** is designed for high-end embedded systems in which real-time performance is needed.
- The **M profile** is designed for deeply embedded micro controller-type systems.

Bit more details on these profiles

A Profile (ARMv7-A): Application processors which are designed to handle complex applications such as high-end embedded operating systems (OSs) (e.g., Symbian, Linux, and Windows Embedded). These processors requiring the highest processing power,

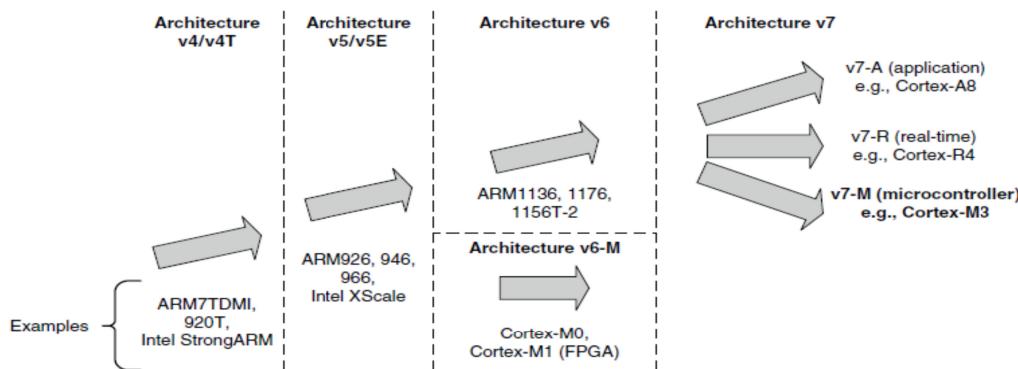
USER MANUAL

virtual memory system support with memory management units (MMUs), and, optionally, enhanced Java support and a secure program execution environment. Example products include high-end mobile phones and electronic wallets for financial transactions.

R Profile (ARMv7-R): Real-time, high-performance processors targeted primarily at the higher end of the real-time market, those applications, such as high-end breaking systems and hard drive controllers, in which high processing power and high reliability are essential and for which low latency is important.

M Profile (ARMv7-M): Processors targeting low-cost applications in which processing efficiency is important and cost, power consumption, low interrupt latency, and ease of use are critical, as well as industrial control applications, including real-time control systems. The Cortex processor families are the first products developed on architecture v7, and the Cortex- M3 processor is based on one profile of the v7 architecture, called ARM v7-M, an architecture specification for micro controller products.

Below figure shows the development stages of ARM versions



1.4 Instruction Set Development

Enhancement and extension of instruction sets used by the ARM processors has been one of the key driving forces of the architecture's evolution. Historically (since ARM7TDMI), two different instruction sets are supported on the ARM processor: the ARM instructions that are 32 bits and Thumb instructions that are 16 bits. During program execution, the processor can be dynamically switched between the ARM state and the Thumb state to use either one of the instruction sets. The Thumb instruction set provides only a subset of the ARM instructions, but it can provide higher code density. It is useful for products with tight memory requirements.

The Thumb-2 Technology and Instruction Set Architecture

The Thumb-2 technology extended the Thumb Instruction Set Architecture (ISA) into a highly efficient and powerful instruction set that delivers significant benefits in terms of ease of use, code size, and performance. The extended instruction set in Thumb-2 is a super set of the previous 16-bit Thumb instruction set, with additional 16-bit instructions alongside 32-bit instructions. It allows more complex operations to be carried out in the Thumb state, thus allowing higher efficiency by reducing the number of states switching between ARM state and Thumb state.

USER MANUAL

Focused on small memory system devices such as micro controllers and reducing the size of the processor, the Cortex-M3 supports only the Thumb-2 (and traditional Thumb) instruction set. Instead of using ARM instructions for some operations, as in traditional ARM processors, it uses the Thumb-2 instruction set for all operations. As a result, the Cortex-M3 processor is not backward compatible with traditional ARM processors.

Nevertheless, the Cortex-M3 processor can execute almost all the 16-bit Thumb instructions, including all 16-bit Thumb instructions supported on ARM7 family processors, making application porting easy. With support for both 16-bit and 32-bit instructions in the Thumb-2 instruction set, there is no need to switch the processor between Thumb state (16-bit instructions) and ARM state (32-bit instructions). For example, in ARM7 or ARM9 family processors, you might need to switch to ARM state if you want to carry out complex calculations or a large number of conditional operations and good performance is needed, whereas in the Cortex-M3 processor, you can mix 32-bit instructions with 16-bit instructions without switching state, getting high code density and high performance with no extra complexity.

The Thumb-2 instruction set is a very important feature of the ARMv7 architecture. Compared with the instructions supported on ARM7 family processors (ARMv4T architecture), the Cortex-M3 processor instruction set has a large number of new features. For the first time, hardware divide instruction is available on an ARM processor, and a number of multiply instructions are also available on the Cortex-M3 processor to improve data-crunching performance. The Cortex-M3 processor also supports unaligned data accesses, a feature previously available only in high-end processors.

1.5 Advantages of Cortex M3 processors

The Cortex-M3 addresses the requirements for the 32-bit embedded processor market in the following ways:

Greater performance efficiency: allowing more work to be done without increasing the frequency or power requirements

Low power consumption: enabling longer battery life, especially critical in portable products including wireless networking applications

Enhanced determinism: guaranteeing that critical tasks and interrupts are serviced as quickly as possible and in a known number of cycles

Improved code density: ensuring that code fits in even the smallest memory footprints

Ease of use: providing easier programming and debugging for the growing number of 8-bit and 16-bit users migrating to 32 bits

Lower cost solutions: reducing 32-bit-based system costs close to those of legacy 8-bit and 16-bit devices and enabling low-end, 32-bit micro controllers to be priced at less than US\$1 for the first time

Wide choice of development tools: from low-cost or free compilers to full-featured development suites from many development tool vendors

Cost savings can be achieved by improving the amount of code reuse across all systems. Because Cortex-M3 processor-based micro controllers can be easily programmed using the C language and are based on a well-established architecture, application code can be ported and reused easily, reducing development time and testing costs.

1.6 Applications of Cortex M3 processors

Low-cost micro controllers: The Cortex-M3 processor is ideally suited for low-cost micro controllers, which are commonly used in consumer products, from toys to electrical appliances. It is a highly competitive market due to the many well-known 8-bit and 16-bit micro controller products on the market. Its lower power, high performance, and ease-of-use advantages enable embedded developers to migrate to 32-bit systems and develop products with the ARM architecture.

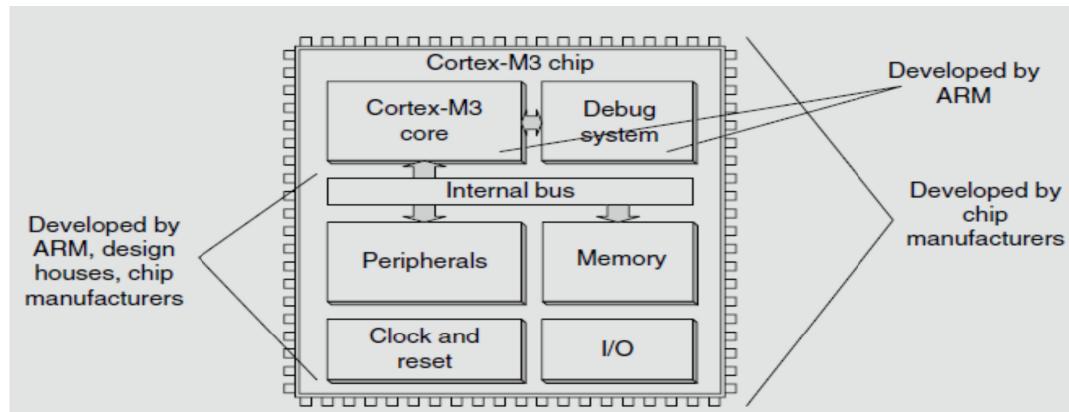
Automotive: Another ideal application for the Cortex-M3 processor is in the automotive industry. The Cortex-M3 processor has very high-performance efficiency and low interrupt latency, allowing it to be used in real-time systems. The Cortex-M3 processor supports up to 240 external vectored interrupts, with a built-in interrupt controller with nested interrupt supports and an optional MPU, making it ideal for highly integrated and cost-sensitive automotive applications.

Data communications: The processor's low power and high efficiency, coupled with instructions in Thumb-2 for bit-field manipulation, make the Cortex-M3 ideal for many communications applications, such as Bluetooth and ZigBee.

Industrial control: In industrial control applications, simplicity, fast response, and reliability are key factors. Again, the Cortex-M3 processor's interrupt feature, low interrupt latency, and enhanced fault-handling features make it a strong candidate in this area.

Consumer products: In many consumer products, a high-performance microprocessor (or several of them) is used. The Cortex-M3 processor, being a small processor, is highly efficient and low in power and supports an MPU enabling complex software to execute while providing robust memory protection.

1.7 The Cortex-M3 Processor versus Cortex-M3-Based Micro Controllers



The Cortex-M3 processor is the central processing unit (CPU) of a micro controller chip. In addition, a number of other components are required for the whole Cortex-M3 processor-based micro controller. After chip manufacturers license the Cortex-M3 processor, they can put the Cortex-M3 processor in their silicon designs, adding memory, peripherals, input/output (I/O), and other features. Cortex-M3 processor-based chips from different manufacturers will have different memory sizes, types, peripherals, and features.

2. INTRODUCTION TO MICRO CONTROLLER LPC1768

2.1 Architectural Overview

The LPC1768FBD100 is an ARM Cortex-M3 based micro controller for embedded applications requiring a high level of integration and low power dissipation. The ARM Cortex-M3 is a next generation core that offers system enhancements such as modernized debug features and a higher level of support block integration. LPC1768 operate up to 100 MHz CPU frequency.

The peripheral complement of the LPC1768 includes up to 512 kilo bytes of flash memory, up to 64KB of data memory, Ethernet MAC, a USB interface that can be configured as either Host, Device, or OTG, 8 channel general purpose DMA controller, 4 UARTs, 2 CAN channels, 2 SSP controllers, SPI interface, 3 I2C interfaces, 2-input plus 2-output I2S interface, 8 channel 12-bit ADC, 10-bit DAC, motor control PWM, Quadrature Encoder interface, 4 general purpose timers, 6-output general purpose PWM, ultra-low power RTC with separate battery supply, and up to 70 general purpose I/O pins.

The LPC1768 use a multi layer AHB(Advanced High Performance Bus) matrix to connect the ARM Cortex-M3 buses and other bus masters to peripherals in a flexible manner that optimizes performance by allowing peripherals that are on different slaves ports of the matrix to be accessed simultaneously by different bus masters.

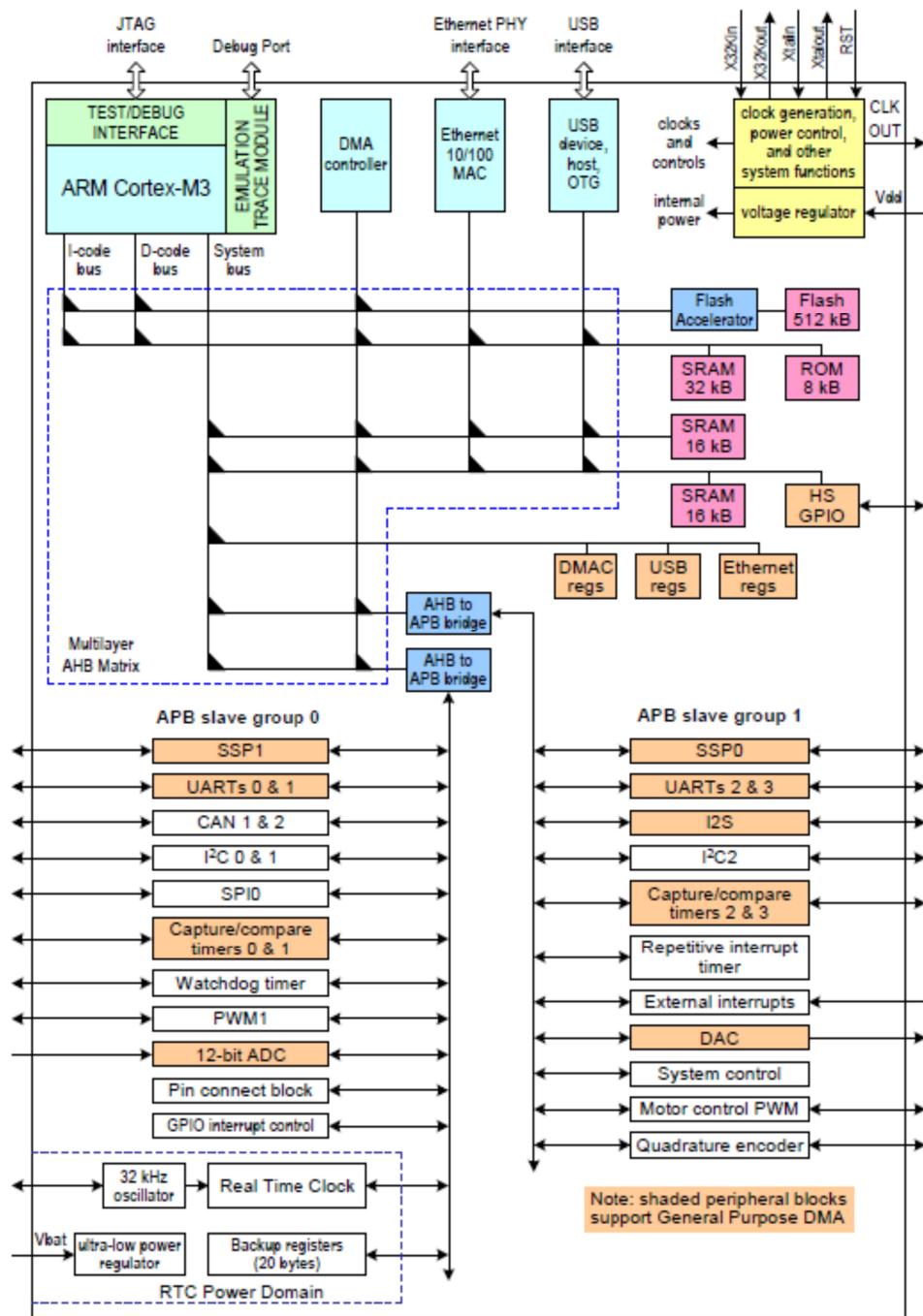
On-chip flash memory system

The LPC1768 contains up to 512 KB of on-chip flash memory. A flash memory accelerator maximizes performance for use with the two fast AHB Lite buses. This memory may be used for both code and data storage. Programming of the flash memory may be accomplished in several ways. It may be programmed In System via the serial port. The application program may also erase and/or program the flash while the application is running, allowing a great degree of flexibility for data storage field firmware upgrades, etc.

On-chip Static RAM

The LPC1768 contains up to 64 KB of on-chip static RAM memory. Up to 32 KB of SRAM, accessible by the CPU and all three DMA controllers are on a higher-speed bus. Devices containing more than 32 KB SRAM have two additional 16 KB SRAM blocks, each situated on separate slave ports on the AHB multilayer matrix. This architecture allows the possibility for CPU and DMA accesses to be separated in such a way that there are few or no delays for the bus masters.

2.2 Block Diagram of LPC1768FBD100



2.3 A brief description of the blocks:

Nested vector interrupt controller

The NVIC is an integral part of the Cortex-M3. The tight coupling to the CPU allows for low interrupt latency and efficient processing of late arriving interrupts.

Features

- Controls system exceptions and peripheral interrupts
- In the LPC1768, the NVIC supports 33 vectored interrupts
- 32 programmable interrupt priority levels, with hardware priority level masking
- Relocatable vector table
- Non-Maskable Interrupt (NMI)
- Software interrupt generation

Interrupt sources

Each peripheral device has one interrupt line connected to the NVIC but may have several interrupt flags. Individual interrupt flags may also represent more than one interrupt source.

Any pin on Port 0 and Port 2 (total of 42 pins) regardless of the selected function, can be programmed to generate an interrupt on a rising edge, a falling edge, or both.

General purpose DMA controller

The GPDMA(General Purpose Direct Memory Access) is an AMBA AHB (Advanced Micro controller Bus Architecture Advance high performance bus) compliant peripheral allowing selected peripherals to have DMA support.

The GPDMA enables peripheral-to-memory, memory-to-peripheral, peripheral-to-peripheral, and memory-to-memory transactions. The source and destination areas can each be either a memory region or a peripheral, and can be accessed through the AHB master. The GPDMA controller allows data transfers between the USB and Ethernet controllers and the various on-chip SRAM areas. The supported APB peripherals are SSP0/1, all UARTs, the I2S-bus interface, the ADC, and the DAC. Two match signals for each timer can be used to trigger DMA transfers.

Pin connect block

The pin connect block allows selected pins of the micro controller to have more than one function. Configuration registers control the multiplexers to allow connection between the pin and the on-chip peripherals. Peripherals should be connected to the appropriate pins prior to being activated and prior to any related interrupt(s) being enabled.

Activity of any enabled peripheral function that is not mapped to a related pin should be considered undefined. Most pins can also be configured as open-drain outputs or to have a pull-up, pull-down, or no resistor enabled.

USER MANUAL

Fast general purpose parallel I/O

Device pins that are not connected to a specific peripheral function are controlled by the GPIO registers. Pins may be dynamically configured as inputs or outputs. Separate registers allow setting or clearing any number of outputs simultaneously. The value of the output register may be read back as well as the current state of the port pins.

Ethernet

The Ethernet block contains a full featured 10 Mbit/s or 100 Mbit/s Ethernet MAC(Media Access Controller) designed to provide optimized performance through the use of DMA hardware acceleration. Features include a generous suite of control registers, half or full duplex operation, flow control, control frames, hardware acceleration for transmit retry, receive packet filtering and wake-up on LAN activity. Automatic frame transmission and reception with scatter-gather DMA off-loads many operations from the CPU.

The Ethernet block interfaces between an off-chip Ethernet PHY using the Reduced Media Independent Interface (RMII) protocol and the on-chip Media Independent Interface Management (MIIM) serial bus.

USB interface

The Universal Serial Bus (USB) is a 4-wire bus that supports communication between a host and one or more (up to 127) peripherals. The host controller allocates the USB bandwidth to attached devices through a token-based protocol. The bus supports hot plugging and dynamic configuration of the devices. All transactions are initiated by the host controller.

The USB interface includes a device, Host, and OTG controller with on-chip PHY for device and Host functions. The OTG switching protocol is supported through the use of an external controller.

USB device controller enables 12 Mbit/s data exchange with a USB Host controller. It consists of a register interface, serial interface engine, endpoint buffer memory, and a DMA controller. The serial interface engine decodes the USB data stream and writes data to the appropriate endpoint buffer. The status of a completed USB transfer or error condition is indicated via status registers. An interrupt is also generated if enabled. When enabled, the DMA controller transfers data between the endpoint buffer and the on-chip SRAM.

USB host controller enables full- and low-speed data exchange with USB devices attached to the bus. It consists of a register interface, a serial interface engine, and a DMA controller. The register interface complies with the OHCI specification.

USB OTG controller is a supplement to the USB 2.0 specification that augments the capability of existing mobile devices and USB peripherals by adding host functionality for connection to USB peripherals.

The OTG Controller integrates the host controller, device controller, and a master-only I2C-bus interface to implement OTG dual-role device functionality. The dedicated I2C-bus interface controls an external OTG transceiver.

USER MANUAL

CAN controller and acceptance filters

The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time control with a very high level of security. Its domain of application ranges from high-speed networks to low cost multiplex wiring.

The CAN block is intended to support multiple CAN buses simultaneously, allowing the device to be used as a gateway, switch, or router among a number of CAN buses in industrial or automotive applications.

12-bit ADC

The LPC1768 contain a single 12-bit successive approximation ADC with eight channels and DMA support.

10-bit DAC

The DAC allows to generate a variable analog output. The maximum output value of the DAC is VREFP.

UART's

The LPC1768 contain four UART's. In addition to standard transmit and receive data lines, UART1 also provides a full modem control handshake interface and support for RS-485/9-bit mode allowing both software address detection and automatic address detection using 9-bit mode.

The UART's include a fractional baud rate generator. Standard baud rates such as 115200 Baud can be achieved with any crystal frequency above 2 MHz.

SPI serial I/O controller

The LPC1768 contain one SPI controller. SPI is a full duplex serial interface designed to handle multiple masters and slaves connected to a given bus. Only a single master and a single slave can communicate on the interface during a given data transfer. During a data transfer the master always sends 8 bits to 16 bits of data to the slave, and the slave always sends 8 bits to 16 bits of data to the master.

SSP serial I/O controller

The LPC1768 contain two SSP controllers. The SSP controller is capable of operation on a SPI, 4-wire SSI, or Microwire bus. It can interact with multiple masters and slaves on the bus. Only a single master and a single slave can communicate on the bus during a given data transfer. The SSP supports full duplex transfers, with frames of 4 bits to 16 bits of data flowing from the master to the slave and from the slave to the master. In practice, often only one of these data flows carries meaningful data.

I2C-bus serial I/O controllers

The LPC1768 each contain three I2C-bus controllers. The I2C-bus is bidirectional for inter-IC control using only two wires: a Serial Clock line (SCL) and a Serial DAta line (SDA). Each device is recognized by a unique address and can operate as either a

receiver-only device or a transmitter with the capability to both receive and send information (such as memory). Transmitters and/or receivers can operate in either master or slave mode, depending on whether the chip has to initiate a data transfer or is only addressed. The I2C is a multi-master bus and can be controlled by more than one bus master connected to it.

I2S-bus serial I/O controllers

The I2S(Inter IC Sound)-bus provides a standard communication interface for digital audio applications. The *I2S-bus specification* defines a 3-wire serial bus using one data line, one clock line, and one word select signal. The basic I2S-bus connection has one master, which is always the master, and one slave. The I2S-bus interface provides a separate transmit and receive channel, each of which can operate as either a master or a slave.

General purpose 32-bit timers/external event counters

The LPC1768 include four 32-bit timer/counters. The timer/counter is designed to count cycles of the system derived clock or an externally-supplied clock. It can optionally generate interrupts, generate timed DMA requests, or perform other actions at specified timer values, based on four match registers. Each timer/counter also includes two capture inputs to trap the timer value when an input signal transitions, optionally generating an interrupt.

Pulse width modulator

The PWM is based on the standard Timer block and inherits all of its features, although only the PWM function is pinned out on the LPC1768. The Timer is designed to count cycles of the system derived clock and optionally switch pins, generate interrupts or perform other actions when specified timer values occur, based on seven match registers. The PWM function is in addition to these features, and is based on match register events.

Watchdog timer

The purpose of the watchdog is to reset the micro controller within a reasonable amount of time if it enters an erroneous state. When enabled, the watchdog will generate a system reset if the user program fails to 'feed' (or reload) the watchdog within a predetermined amount of time.

RTC and backup registers

The RTC is a set of counters for measuring time when system power is on, and optionally when it is off. The RTC on the LPC1768 is designed to have extremely low power consumption, i.e. less than 1 uA. The RTC will typically run from the main chip power supply, conserving battery power while the rest of the device is powered up. When operating from a battery, the RTC will continue working down to 2.1 V. Battery power can be provided from a standard 3 V Lithium button cell.

An ultra-low power 32 kHz oscillator will provide a 1 Hz clock to the time counting portion of the RTC, moving most of the power consumption out of the time counting function.

Clocking and power control

Crystal oscillators

The LPC1768 include three independent oscillators. These are the main oscillator, the IRC oscillator, and the RTC oscillator. Each oscillator can be used for more than one purpose as required in a particular application. Any of the three clock sources can be chosen by software to drive the main PLL and ultimately the CPU.

Following reset, the LPC1768 will operate from the Internal RC oscillator until switched by software. This allows systems to operate without any external crystal and the boot loader code to operate at a known frequency.

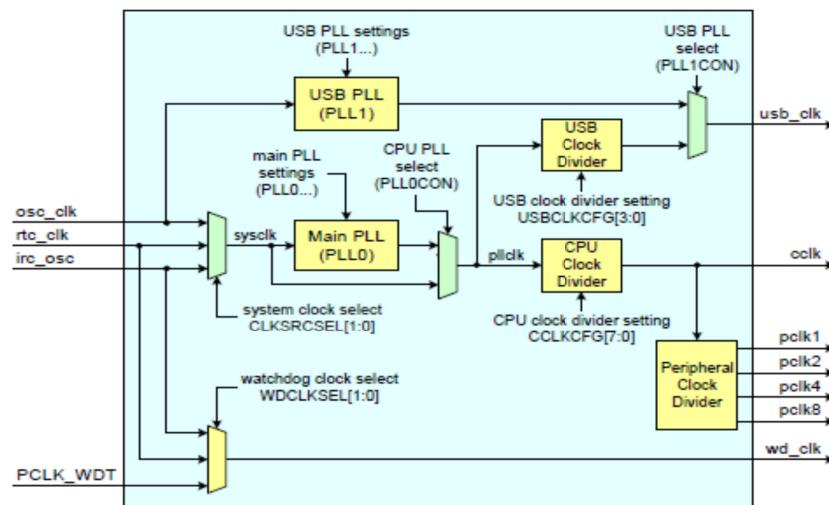
Power control

The LPC1768 support a variety of power control features. There are four special modes of processor power reduction: Sleep mode, Deep-sleep mode, Power-down mode, and Deep power-down mode. The CPU clock rate may also be controlled as needed by changing clock sources, reconfiguring PLL values, and/or altering the CPU clock divider value. This allows a trade-off of power versus processing speed based on application requirements. In addition, Peripheral Power Control allows shutting down the clocks to individual on-chip peripherals, allowing fine tuning of power consumption by eliminating all dynamic power use in any peripherals that are not required for the application. Each of the peripherals has its own clock divider which provides even better power control.

Integrated PMU (Power Management Unit) automatically adjust internal regulators to minimize power consumption during Sleep, Deep sleep, Power-down, and Deep power-down modes.

The LPC1768 also implement a separate power domain to allow turning off power to the bulk of the device while maintaining operation of the RTC and a small set of registers for storing data during any of the power-down modes.

Clock generation block diagram for LPC1768 is shown below



System Control**Reset**

Reset has four sources on the LPC1768: the RESET pin, the Watchdog reset, power-on reset (POR), and the Brown Out Detection (BOD) circuit. The RESET pin is a Schmitt trigger input pin. Assertion of chip Reset by any source, once the operating voltage attains a usable level, causes the RSTOUT pin to go LOW. Once reset is de-asserted, or, in case of a BOD-triggered reset, once the voltage rises above the BOD threshold, the RSTOUT pin goes HIGH. In other words RSTOUT is high when the controller is in its active state.

Emulation and debugging

Debug and trace functions are integrated into the ARM Cortex-M3. Serial wire debug and trace functions are supported in addition to a standard JTAG debug and parallel trace functions. The ARM Cortex-M3 is configured to support up to eight breakpoints and four watch points.

Note: For further details on Controller blocks refer the User manual of LPC176x/5x – UM10360 available at www.nxp.com

3. TECHNICAL SPECIFICATIONS

3.1 Specifications of LPC1768:

- ARM Cortex-M3 processor runs up to 100 MHz frequency.
- ARM Cortex-M3 built-in Nested Vectored Interrupt Controller (NVIC).
- Up to 512kB on-chip flash program memory with In-System Programming (ISP) and In-Application Programming (IAP) capabilities. The combination of an enhanced flash memory accelerator and location of the flash memory on the CPU local code/data bus provides high code performance from flash.
- Up to 64kB on-chip SRAM includes:
 - Up to 32kB of SRAM on the CPU with local code/data bus for high-performance CPU access.
 - Up to two 16kB SRAM blocks with separate access paths for higher throughput. These SRAM blocks may be used for Ethernet, USB, and DMA memory, as well as for general purpose instruction and data storage.
- Eight channel General Purpose DMA controller (GPDMA) on the AHB multilayer matrix that can be used with the SSP, I2S, UART, the Analog-to-Digital and Digital-to-Analog converter peripherals, timer match signals, GPIO, and for memory-to-memory transfers.
- Serial interfaces:
 - Ethernet MAC with RMII interface and dedicated DMA controller.
 - USB 2.0 full-speed controller that can be configured for either device, Host, or OTG operation with an on-chip PHY for device and Host functions and a dedicated DMA controller.
 - Four UART's with fractional baud rate generation, internal FIFO, IrDA, and DMA support. One UART has modem control I/O and RS-485/EIA-485 support.
 - Two-channel CAN controller.
 - Two SSP controllers with FIFO and multi-protocol capabilities. The SSP interfaces can be used with the GPDMA controller.
 - SPI controller with synchronous, serial, full duplex communication and programmable data length. SPI is included as a legacy peripheral and can be used instead of SSP0.
 - Three enhanced I2C-bus interfaces, one with an open-drain output supporting the full I2C specification and Fast mode plus with data rates of 1Mbit/s, two with standard port pins. Enhancements include multiple address recognition and monitor mode.
 - I2S (Inter-IC Sound) interface for digital audio input or output, with fractional rate control. The I2S interface can be used with the GPDMA. The I2S interface supports 3-wire data transmit and receive or 4-wire combined transmit and receive connections, as well as master clock output.
- Other peripherals:
 - 70 General Purpose I/O (GPIO) pins with configurable pull-up/down resistors, open drain mode, and repeater mode. All GPIOs are located on an AHB bus for fast access, and support Cortex-M3 bit-banding. GPIOs can be accessed by the General Purpose DMA Controller. Any pin of ports 0 and 2 can be used to generate an interrupt.

- 12-bit Analog-to-Digital Converter (ADC) with input multiplexing among eight pins, conversion rates up to 200 kHz, and multiple result registers. The 12-bit ADC can be used with the GPDMA controller.
- 10-bit Digital-to-Analog Converter (DAC) with dedicated conversion timer and DMA support.
- Four general purpose timers/counters, with a total of eight capture inputs and ten compare outputs. Each timer block has an external count input. Specific timer events can be selected to generate DMA requests.
- One motor control PWM with support for three-phase motor control.
- Quadrature encoder interface that can monitor one external quadrature encoder.
- One standard PWM/timer block with external count input.
- Real-Time Clock (RTC) with a separate power domain. The RTC is clocked by a dedicated RTC oscillator. The RTC block includes 20 bytes of battery-powered backup registers, allowing system status to be stored when the rest of the chip is powered off. Battery power can be supplied from a standard 3 V Lithium button cell. The RTC will continue working when the battery voltage drops to as low as 2.1 V. An RTC interrupt can wake up the CPU from any reduced power mode.
- Watchdog Timer (WDT). The WDT can be clocked from the internal RC oscillator, the RTC oscillator, or the APB clock.
- Cortex-M3 system tick timer, including an external clock input option.
- Repetitive interrupt timer provides programmable and repeating timed interrupts.
- Standard JTAG test/debug interface as well as Serial Wire Debug and Serial Wire Trace Port options.
- Emulation trace module supports real-time trace.
- Four reduced power modes: Sleep, Deep-sleep, Power-down, and Deep power-down.
- Single 3.3 V power supply (2.4 V to 3.6 V). Temperature range of -40 °C to 85 °C.
- Four external interrupt inputs configurable as edge/level sensitive. All pins on PORT0 and PORT2 can be used as edge sensitive interrupt sources.
- Non-maskable Interrupt (NMI) input.
- Clock output function that can reflect the main oscillator clock, IRC clock, RTC clock, CPU clock, or the USB clock.
- The Wake-up Interrupt Controller (WIC) allows the CPU to automatically wake up from any priority interrupt that can occur while the clocks are stopped in deep sleep, Power-down, and Deep power-down modes
- Processor wake-up from Power-down mode via any interrupt able to operate during Power-down mode (includes external interrupts, RTC interrupt, USB activity, Ethernet wake-up interrupt, CAN bus activity, PORT0/2 pin interrupt, and NMI).
- Each peripheral has its own clock divider for further power savings.
- Brownout detect with separate threshold for interrupt and forced reset.
- On-chip Power-On Reset (POR).

USER MANUAL

- On-chip crystal oscillator with an operating range of 1 MHz to 25 MHz.
- 4 MHz internal RC oscillator trimmed to 1% accuracy that can optionally be used as a system clock.
- An on-chip PLL allows CPU operation up to the maximum CPU rate without the need for a high-frequency crystal. May be run from the main oscillator, the internal RC oscillator, or the RTC oscillator.
- A second, dedicated PLL may be used for the USB interface in order to allow added flexibility for the Main PLL settings.
- Versatile pin function selection feature allows many possibilities for using on-chip peripheral functions.

3.2 SPECIFICATIONS OF THE PRODUCT

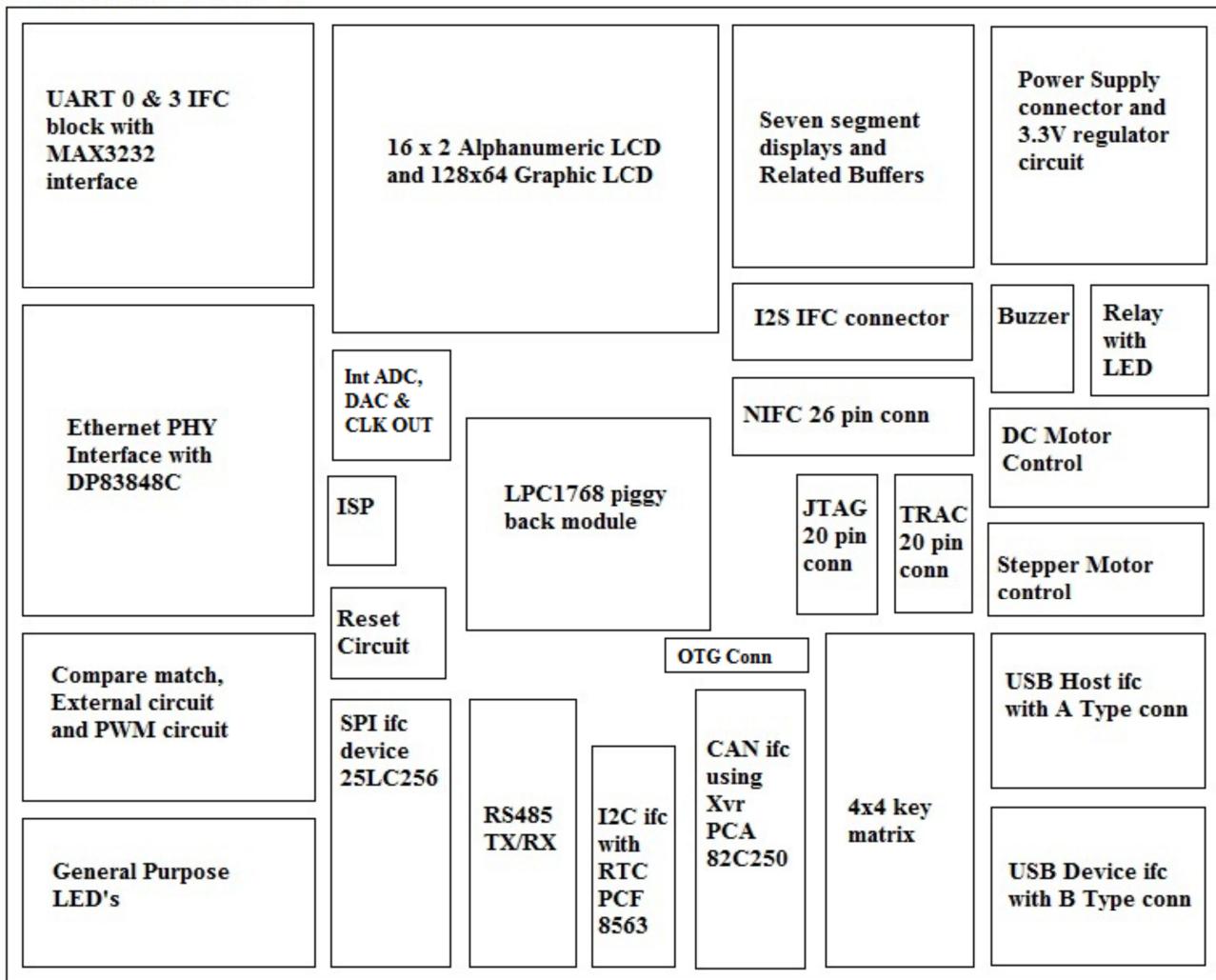
- **LPC1768** 32 Bit ARM Cortex-M3 based controller with 512K bytes Program Flash, 64K bytes RAM
- 12MHz Crystal allows easy communication setup
- One on board voltage regulator for generating 3.3V. Input to this will be from External +5V DC Power supply through a 9-pin DSUB connector
- Reset push-button for resetting the controller
- Three of the four controller UART's are made available to user for external interfaces
 - UART0: General purpose RS232 terminated in a 9 pin female DSUB connector which is also used to program **LPC1768** Flash memory without external Programmer
 - UART1: General purpose RS485 terminated in a 3 pin male reliamate
 - UART3: General purpose RS232 terminated in a 3 pin male reliamate
- Interface to 2 channels of Controller internal ADC
- Interface to Controller internal DAC
- 16×2 alphanumeric LCD and 128×64 Graphic LCD display are provided – one of them can be used at a time
- 4×4 Key-Matrix
- SPI interface: Flash NVROM memory circuit
- Simple I2S demonstration circuit
- I2C interface: RTC IC
- Eight general purpose LED's
- Four-digit multiplexed 7-segment display interface
- Interface circuit for external Interrupt
- Interface circuit for Capture & match
- Interface circuit for PWM with a monitor LED
- On-board stepper motor, DC motor control, Relay interface
- Interface circuit for on board Buzzer
- Ethernet interface with RJ45 female connector
- USB 2.0 full-speed controller that can be configured for either device or Host along with USB link and connect LED's
- Two CAN interfaces with reliamate connector
- Some of the on board interfaces connect to the Controller using short standard 10 pin FRC cables while the others are connected directly

- Standard JTAG connector with ARM 2×10 pin layout for programming/debugging with ARM-JTAG
- Standard Emulation Trace connector with 2×10 pin layout for CPU Execution tracing
- One standard 26-pin FRC connector to connect to some of **ALS standard External Interfaces**
- A number of software examples in ‘C-language’ to illustrate the functioning of the interfaces. The software examples are compiled using an evaluation version of KEIL4 ‘C’ compiler for ARM

4. HARDWARE DESCRIPTION

4.1 BLOCK DIAGRAM

ALS-SDA-ARMCTXM3-01



4.2 DESCRIPTION OF THE DIFFERENT BLOCKS

Power supply:

This board works with a DC +5V supply. On board regulator generates the +3.3v from 5V for controller and other peripherals. Before inserting the piggy back module in to the board, test the voltage levels in this block. TP1 and TP3 in the same block shows the +5V and ground respectively. After shorting the jumper JP6 +3.3V supply flows to board. Test the voltage at TP2. After checking the available voltages at these test points, insert the piggyback module.

Note: Permanently short JP6. Remove to measure current only.

CN8 – Male 9 pin DSUB connector. Used power the board from power supply

Pin Number	Description
1,2,3,6,7,8	No Connection
4,5	GND
9	+5V



Inserting a Piggy back:

CN1 to CN4 are the connectors support the mounting of the piggy back module on base board. While inserting the piggy back connectors 1 to 4 of the piggy back board are to be inserted to connectors 1 to 4 of the base board respectively. Piggy back – 99-05-14 REV00

CN1 – 28 pin connector 14x2 header

Pin No	Description
1	TDO-JTAG
2	TDI-JTAG
3	TMS-JTAG
4	TRST-JTAG
5	TCK-JTAG
6	P0.26
7	P0.25
8	P0.24
9	P0.23
10, 12	3.3V
11, 15	GROUND
13,19,22,23,26,27,28	NC
14	RSTOUT
16	RTCX1
17	RESET
18	RTCX2
20	P1.31
21	P1.30
24	P0.28
25	P0.27

CN2 – 28 pin connector 14x2 header

Pin No	Description
1,2,28	NC
3	P3.26
4	P3.25
5,19	3.3V



USER MANUAL

6	P0.29
7	P0.30
8,18	GROUND
9	P1.18
10	P1.19
11	P1.20
12	P1.21
13	P1.22
14	P1.23
15	P1.24
16	P1.25
17	P1.26
20	P1.27
21	P1.28
22	P1.29
23	P0.0
24	P0.1
25	P0.10
26	P0.11
27	P2.13

CN3 – 28 pin connector 14×2 header

Pin No	Description
1,2,28	NC
3	P2.12
4	P2.11
5	P2.10
6,23	3.3V
7,24	GROUND
8	P0.22
9	P0.21
10	P0.20
11	P1.19
12	P0.15



USER MANUAL

13	P0.17
14	P0.15
15	P0.16
16	P2.9
17	P2.8
18	P2.7
19	P2.6
20	P2.5
21	P2.4
22	P2.3
25	P2.2
26	P2.1
27	P2.0

CN4 – 28 pin connector 14×2 header

Pin no	Description
1	P0.9
2	P0.8
3	P0.7
4	P0.6
5	P0.5
6	P0.4
7	P4.28
8, 22	GROUND
9, 21	+3.3V
10	P4.29
11	P1.17
12	P1.16
13	P1.15
14	P1.14
15	P1.10
16	P1.9
17	P1.8
18	P1.4



19	P1.1
20	P1.0
23	P0.2
24	P0.3
25	RTCK
26, 27, 28	NC

GPIO extension connectors:

There are four 10 pin FRC type male connectors, they extends the controllers general purpose port lines for the use of user requirements. Details on each connector is given below:

CNA – 10 pin male box type FRC connector. Port lines P0.4 to P0.11 from controller are terminated in this connector. They can be extended to interface few on board or external peripherals.

Pin CNA	PIN LPC1768	Description
1	81	P0.4/I2SRX_CLK/RD2/CAP2.0
2	80	P0.5/I2SRX_WS/TD2/CAP2.1
3	79	P0.6/I2SRX_SDA/SSEL1/MAT2.0
4	78	P0.7/I2STX_CLK/SCK1//MAT2.1
5	77	P0.8/I2STX_WS/MISO1/MAT2.2
6	76	P0.9/I2STX_SDA/MOSI1/MAT2.3
7	48	P0.10/TXD2/SDA2/MAT3.0
8	49	P0.11/RXD2/SCL2/MAT3.1
9	-	No connection
10	-	Ground

The pins mentioned in the above table are configured to work as a GPIO's at power on. Other alternate functions on those pins needs to be selected using respective PINSEL registers.

In the given example projects, On board peripherals I2S, GP LED, Relay, stepper motor, DC motor, Graphic LCD Data lines, seven segment data lines are interfaced with this connector.

USER MANUAL

CNB – 10 pin male box type FRC connector. Port lines from P1.23 to P1.26 and P2.10 to P2.13 are terminated in this connector.

Description of the connector CNB:

Pin CNB	Pin LPC1768	Description
1	37	P1.23/MCI1/PWM1.4/MISO0
2	38	P1.24/MCI2/PWM1.5/MOSI0
3	39	P1.25/MCOA1/MAT1.1
4	40	P1.26/MCOB1/PWM1.6/CAP0.0
5	53	P2.10/EINT0/NMI
6	52	P2.11/EINT1/I2STX_CLK
7	51	P2.12/EINT2/I2STX_WS
8	50	P2.13/EINT3/I2STX_SDA
9	-	No connection
10	-	Ground

Short JP4(1, 2) to use CNB pin 8. In the given example projects, On board peripherals key pad, PWM, External interrupt, Compare and match circuit, Enable lines of seven segment displays are interfaced with this connector.

CNC – 10 pin male box type FRC connector. Port lines from P0.15 to P0.22 and P2.13 are terminated in this connector.

Pin CNC	Pin LPC1768	Description
1	62	P0.15/TXD1/SCK0/SCK
2	63	P0.16/RXD1/SSEL0/SSEL
3	61	P0.17/CTS1/MISO0/MISO
4	60	P0.18/DCD1/MOSI0/MOSI
5	59	P0.19/DSR1/SDA1
6	58	P0.20/DTR1/SCL1
7	57	P0.21/RI1/RD1
8	56	P0.22/RTS1/TD1
9	50	P2.13/I2STX_SDA
10	-	Ground

Short JP4(2, 3) to work CNC pin 9. In the given example projects, On board peripherals SPI, I2C, RS485 are interfaced with this connector.

CND – 10 pin male box type FRC connector. Port lines from P0.23 to P0.28 and P2.0 to P2.1 are terminated in this connector.

USER MANUAL

Pin CND	Pin LPC1768	Description
1	9	P0.23/AD0.0/I2SRX_CLK/CAP3.0
2	8	P0.24/AD0.1/I2SRX_WS/CAP3.1
3	7	P0.25/AD0.2/I2SRX_SDA/TXD3
4	6	P0.26/AD0.3/AOUT/RXD3
5	25	P0.27/SDA0/USB/SDA
6	24	P0.28/SCL0/USB_SCL
7	75	P2.0/PWM1.1/TXD1
8	74	P2.1/PWM1.2/RXD1
9	-	No connection
10	-	Ground

Short JP5 to use the pin 4 of this connector. In the given examples LCD is interfaced with this connector. While using USB OTG pin 5 and 6 of this connector not available.

Note: Other than mentioned in the manual, connections from these GPIO connectors to peripherals in different ways are also allowed only when the port lines are used as a GPIO. Care needs to be taken while interfacing to the peripherals related to internal features of the controller. Because above GPIO connectors and related connector at other blocks of the board are 1:1 mapped. So using other than a specified connector will create a problem.

NIFC Interface block:

This is a 26 pin connector, in which 24 port lines are terminated here along with optional **+5V via JP1** and ground. Some of ALS standard external NIFC's can be interfaced through this connector or any other interface as per user need.

CN6 – 26 pin box type FRC male connector. ALS standard NIFC interfaces can be done with this connector.

Pin no	Description
1, 2	P2.0, P2.1
3, 4	P0.25, P0.26
5, 6	P0.23, P0.24
7,8	P0.10, P0.11
9, 10	P0.8, P0.9
11, 12	P0.6, P0.5
13, 14	P0.4, P0.5
15, 16	P0.21, P0.22
17, 18	P0.19, P0.20



USER MANUAL

19, 20	P0.17, P0.18
21, 22	P0.15, P0.16
23, 24	P2.2, P2.3
25	+5V via JP1
26	Ground

Note: Short JP5 to use pin 4 of this connector.

JTAG Interface:

CN5 – 20 pin box type FRC male connector. An ARM parallel port JTAG interface can be done using this connector.

Pin no	Description
1,2	+3.3V
3	TRST-JTAG
4,6,8,10,12,14,16,18,20	Ground
5	TDI-JTAG
6	TMS-JTAG
7	TCK-JTAG
9	TCK-JTAG
11	RTCK-JTAG
13	TDO-JTAG
15	RESET
17,18	Resistive ground

JP2 – Allows to ground the RTCK line via 10K resistor

Trace Interface block:

Trace is a function which provides CPU debugging capability. Trace can be done with this connector interface.

CN7 – 20 pin box type FRC male connector. Trace port connector.

Pin no	Description
1	+3.3V
2	TMS-JTAG
3,5,7,9,11,13,15,17,19	TCK-JTAG
4	TCK-JTAG
6	TDO-JTAG



8	TDI-JTAG
10	RESET
12	P2.6 – TRACE CLOCK
14	TDO-JTAG/TDATA0
16	TRST-JTAG/TDATA1
18	TDATA2
20	TDATA3

UART0 and UART3 interface block:

If you are connecting, **ALS-SDA-ARMCTXM3-01** Evaluation board to a computer, you have to use Hyper Terminal communication package. This package allows the user to use computer as a simple display terminal for **ALS-SDA-ARMCTXM3-01** and to transfer data between computer and **ALS-SDA-ARMCTXM3-01**. To set this communication package refer the following steps.

Follow the sequence as **Start -> All Programs -> Accessories -> Communications -> Hyper Terminal**.

1. Give the name for Hyper Terminal and then press **OK**.
2. The window “**Connect To**” will appear, Select the COM port you are using in “**Connect Using**” option.
3. Go to **FILE MENU** click on **properties** then click on **Configure**.
Select the “**Port Settings**”
Bits per Sec : 9600 (Depends on application software)
Data bits : 8
Parity : None
Stop bits : 1
Flow control : None
Then **Press -> OK**
Go to “**File**” -> “**Save**”
4. Bring the Shortcut icon for Hyper Terminal on the Desktop of your computer.
Use serial **Cross Cable** from 9-pin DSUB Female connector to the PC Com port.

UART0 is used for the boot loader purpose. Also can be used for interfacing with PC for any communication purpose. CN9 has the TXD and RXD of UART0 along with the handshaking lines RTS and DTR. UART3 lines are terminated to RM1 – a 3 pin male connector can be used for any communication. MAX3232 is used between these connectors and controller. Hence the signals at CN9 and RM1 will be in RS232 level. These lines cannot be used directly for TTL interface. A conversion must be done from RS232 to TTL level at other side if they have to be used with TTL lines.

CN9 - (UART0) 9-Pin D-type Female connector connects to the COM port of host PC for In System Programming (ISP) application and transferring the data between controller device and host computer. If UART0 is used for the flash programming on the switch SW21. Off the switch21 if UART0 is used for communication purpose. Note that there is an interfacing RS232C device between the connector and the device pins listed below



Pin Number at CN9	Pin LPC1768	Description
1,4,7,9	-	NC
2	99(P0.3)	R0IN – RXD0
3	98(P0.2)	T0OUT – TXD0
5	-	GND
6	17(RESET)	DTR
8	53(P2.10)	RTS

RM1 – 3 pin male reliamate. Pin 1 – TXD3, Pin 2 – RXD3. Pin 3 – Ground. If UART3 is used through this reliamate JP7(1, 2) and JP8(1, 2) has to be shorted.

SW1 – is to isolate the hand shaking signals from board signals (RTS, DTR) and connector. Keep this switch ON before programming the controller flash memory with the application code. Keep it open to Run the loaded program and reset. Especially if UART0 is using for any communication purpose, user must keep these switch open to execute UART0 related code.

RTS – Controls the ISP line of the controller P2.10

DTR – Used to interface controller reset.

Ethernet Interface block:

Interface to local area network is possible from the controller thorough a Physical layer Transceiver – U14. U14 interfaces between LAN connector CN13 and controller. A 50MHz crystal oscillator X1 is used for PHY.

LD12 indicates the mode of communication. It will be on when feature is not working or if communication speed is 10MBit/s. During 100MBit /s communication mode it will be off. Left LED(Yellow color) toggles during the communication. Right LED(Green color) will be on always during the cable connection.

CN13 – Pulse Jack for 1:1 LAN cable connection with hub. Pin details at PCB side to Physical layer transceiver are given in the table.

Pin no	Description
1	TD+
2	TD-
3	RD+
4	CT
5	CT
6	RD-
7, 8	External ground
9	Left LED anode
10	Left LED cathode
11	Right LED cathode
12	Right LED anode



USER MANUAL

Cable side:

Pin no	Description
1	TX+
2	TX-
3	RX+
6	RX-
4,5,7,8	Ground

USB Device Interface Block:

Controller works as a USB device to be interfaced with PC or any other Host controllers. Interface is done using B type Male USB connector CN15. LED LD14 glows if the host device is connected. LD13 toggles during the communication. Switch On pins 1 & 2 at Switch 20. Switch off SW19 and keep CN18 free. Short JP18(1, 2) to use this feature.

CN15 – B type male right angle USB connector

Pin no	Description
1	VBUS
2	Data -
3	Data +
4	Ground

USB Host interface block:

LPC1768 also works as a Host controller to read or to write the data from any connected device. An 'A type' male connector is used for connecting the USB device to the board. LD14 glows if the USB device is connected. Switch On the 1st and 2nd pin at SW19. U15 acts as a Serial Bus standard power switch and over-current protection for all host port applications. Short JP19 to use this feature.

CN14 - A type male right angle USB connector.

Pin no	Description
1	Power U15
2	Data -
3	Data +
4	Ground

OTG Connector:

compatible lines for OTG are terminated at the connector CN18 – a male straight berg of 8 pin. To interface any external OTG hardware, these lines can be extended

USER MANUAL

using connection leads. While using this connector, keep SW19 and 20 at off position. Keep CND and pin 9 at CNC free. Short JP4(2, 3). below table shows the description of the pins at this connector. Note that pin 1 is at diode D6 side and last pin is at CND side.

Pin no	Description
1	3.3V
2	USB_D+
3	USB_D-
4	P0.27(USB_SDA)
5	P0.28(USB_SCL)
6	EXT INT3
7	RSTOUT
8	GROUND

Alphanumeric and Graphic LCD's:

A **16x2 alphanumeric LCD** can be used to display the message from controller. 16 pin small LCD has to be mounted to the connector CN11. 10 pin connector CNAD is used to interface this LCD from controller. Only higher 4 data lines are used among the 8 LCD data lines.

Use POT3 for contrast adjustment and Short the jumper JP16 to use this LCD. LCD connector CN11 is described in this table. CN11 is Single row 16 pin female berg.

Pin no CN11	Description
1	Ground
2	+5V
3	LCD contrast
4	RS
5,7,8,9,10	NC
6	En
11 to 14	Data 4 to 7
15	Back light anode
16	Back light cathode

Connection from CNAD to LCD connector CN11 is shown in the below table.

Pin no at CNAD	Description	Pin no at CN11
1	L0 – Data line 4 of LCD	11
2	L1 – Data line 5 of LCD	12



3	L2 – Data line 6 of LCD	13
4	L3 – Data line 7 of LCD	14
5	L5 – Command line of LCD	4
6	L5 – Enable line of LCD	6

At a given time we can either use AN LCD or graphic LCD. Using both at a time is not allowed.

128x64 Graphic LCD has to be mounted on the 20x2 female straight connector CN10. Graphic LCD has got single row connector, but board is having 2 rows. This is just for providing the mounting grip for LCD. User can select any 20 pin row in the connector CN10 to insert LCD connector. Both the rows on the connector CN10 of the board are shorted from pin to pin respectively.

This LCD connector CN10 can be interfaced from the controller using the connectors CNAD and CND1. Connection between CN10 and CNAD and CND1 is given in the below table.

Pin no CN10	Description	CNAD	CND1
1	Ground (0V)	-	-
2	DC +5V	-	-
3	Contrast Adjust	-	-
4	RS – Data/Instruction	-	1
5	R/W – Read / Write	-	2
6	EN – Enable Signal	-	3
7	BD0 – Data bus line	1	-
8	BD1 – Data bus line	2	-
9	BD2 – Data bus line	3	-
10	BD3 – Data bus line	4	-
11	BD4 – Data bus line	5	-
12	BD5 – Data bus line	6	-
13	BD6 – Data bus line	7	-
14	BD7 – Data bus line	8	-
15	CS1 – Chip 1 enable signal	-	4
16	CS2 – Chip 2 enable signal	-	5
17	Reset Signal	-	6
18	Negative voltage output (DC -5V)	-	-
19	Back light positive	-	-
20	Back light negative	-	-

Relay buzzer and Motors control block:

It is a high Current applications block where a stepper motor, a Dc motor and a relay are interfaced through the high current driver ULN2803 (U13). These lines will have high current (max 300 mA) with low voltage level of 0.7V.

USER MANUAL

The Stepper motor can be interfaced to the board by connecting it to the Power Mate PM1. The direction of the rotation can be changed through software. The DC Motor can also be interfaced to the board by connecting it to the Reliamate RM5. The direction of the rotation can be changed through software.

The Relay K2 is switched between ON and OFF state. The LED L12 will toggle for every relay switch over. The contact of NO & NC of the relay can be checked at the MKDSN connector CN12 pins 1 & 2 using a Digital Multimeter – these contacts can be connected to external devices.

Using connector CNA5 micro controller can interface with this block. Description of the connector pins are given in below table.

Pin @ CNA5	Description
1 to 4	Buffered from U13 used for stepper motor control
5	Buffered from U13 and connected to relay k2 coil. coil other end is connected to +5V
6	Connected to both 1 & 2 of U13; corresponding outputs of U13 are taken to NO and NC contacts of relay K1
7	One end of coil of relay K1
8	Controls the buzzer

CN12 – 2 pin MKDSN connector provides the K2 relay output for external use. This is controlled using 5th pin of CNA5.

Pin 1 – COM of the relay.
pin 2 – NO of the relay.

RM5 – it's a 2 pin straight male reliamate. COM pins of relay K1 are terminated here. It is used to run and direction control of DC motor.

PM1 – it's a 5 pin straight male power mate.

Pin no	Description
1	+5v supply
2	Phase A
3	Phase B
4	Phase C
5	Phase D

Pin 2 to 5 are phase A to D output for the stepper motor respectively.

Internal ADC interface:

On board there are two interfaces for internal ADC's. AD0.5 (pin P1.31) of controller is used to convert the analog input voltage varied using POT1 to digital value. AD0.4(Pin 1.30) used convert the analog voltage varied using POT4. A 0 to 3.3V is the input voltage

USER MANUAL

range. 000 to FFF is the converted digital voltage range here. Short JP18(2, 3) to use AD0.4.

Internal DAC:

An analog output from the controller can be observed in this block at TP8. Open JP5 to use this feature and use CRO or multimeter to watch analog output value.

Internal clock out:

Pin P1.27 output the clock frequency when this pin is configured as Clock out pin. To check the clock use TP9 and open JP19.

Reset Circuit:

It's an active high input to the controller. A low level on this pin cause the controller to be in reset mode. Switch-SW1 to be used for the purpose of resetting the controller.

ISP Circuit:

A LOW level after reset at pin P2.10 is considered an external hardware request to start the ISP command handler. Assuming that power supply pins are on their nominal levels when the rising edge on RESET pin is generated, it may take up to 3 ms before P2.10 is sampled and the decision on whether to continue with user code or ISP handler is made.

During programming the controller using Flash magic software, jumper – JP3 needs to be shorted. This jumper connects the ISP line P2.10 to ground level during the Flash magic attempt to program the flash.

CAN Interface:

CAN 1 and CAN 2 are interfaced externally via CAN transceivers U3, U4. U3 is used for CAN1 and U4 is used for CAN2. Transceivers inputs and outputs are terminated at the relamate RM2 for external use. While using CAN1 short JP7(2, 3) and JP8(2, 3). JP7 pin 2 is P0.1 of the controller which is TD1 and JP7 pin 3 is TD1 of U3. JP8 pin 2 is P0.0 (RD1) of the controller and JP8 pin 3 is RD1 of U3.

Pin descriptions of RM2

Pin no	Description
1	CANL2 from U4
2	CANH2 from U4
3, 4	Ground
5	CANL1 from U3
6	CANH1 from U3

General purpose LED's:

There are 8 general purpose LED's on the board – LD2 to LD9. This block can be interfaced to the controller by connecting a 10 pin FRC cable from any of the GPIO connector to CNA1 in this block. A high level from the controller will turn on the LED.



USER MANUAL

CNA1 – 10 pin FRC box type male connector. Pin 1 to 8 of this connector are connected to LED's LD2 to LD9 respectively.

Pin no	Connects to
1	Anode of LD2
2	Anode of LD3
3	Anode of LD4
4	Anode of LD5
5	Anode of LD6
6	Anode of LD7
7	Anode of LD8
8	Anode of LD9
9	No connection
10	Ground

Cathodes of all the LED's are grounded through resistors.

SPI, I2C and RS485 Interface:

This block has SPI device U5 – EEPROM, RS485 transceiver – U6 and I2C device U16 – RTC. An LED LD15 is provided for general use. This block can be accessed by the controller using a connector CNC1. Pin description of the connector is given below. CNC is the compatible connector to work the features in this block.

Pin no CNC	Description
1	SCK/TXD1 for U5 or U6
2	SSEL/RXD1 for U5 or U6
3	MISO for U5
4	MOSI for U5
5	SDA for U16
6	SCL for U16
7	No connection
8	RTS1 for U6
9	Interrupt of U16
10	Ground

Since there are two functions on Pin 1 and 2, jumpers JP12 and JP13 needs to be used for selecting a required function. While using SPI, short JP12(2, 3) and JP13(2, 3) and while using RS485 short JP12(1, 2) and JP13(1, 2).

USER MANUAL

U5 – is a 256 K bit Serial Electrically Erasable Programmable Read only memory device. This memory can be accessed by the controller using a SPI bus.

U6 – MAX485 is a low-power transceiver for RS-485 communication. Using RTS1 pin a transmission can be enabled or disabled. Shorting JP10 enables the receiver and opening it disables the receiver. Short JP11 always. The external interfacing lines of this IC are terminated at RM3 – a 3 pin male straight reliamate.

Pin no RM3	Description
1	TX+ - Non inverting Receiver Input and Non inverting Driver Output
2	TX- – Inverting Receiver Input and Inverting Driver Output
3	Ground

U16 – Real-Time Clock (RTC) and calendar optimized for low power consumption. A 32.768K Hz tubular crystal is used as oscillator input to the U16. CR2032 battery is used a backup for the IC to retain the configuration of internal registers of RTC. To use the clock generated by the RTC as an interrupt to the controller, short JP15.

4x4 key matrix Interface:

The switches SW3 to SW18 are organized as 4 rows X 4 columns matrix. One end of all the switches are configured as columns. The other end of the matrix configured as rows. A row line will be always an output from the controller. Column lines are pulled to ground. A high level sent from the row will appear at column end if the switch is pressed.

Connector CNB3 is used for interfacing this block with controller. At controller end any connector can be used to interact this connector CNB3.

PWM, External Interrupt and Compare match Interface:

A PWM output from the controller can be observed as an intensity variation of the LED LD10. Pulses on the CRO can be observed at TP5. Voltage level of this pin can be observed at TP6.

A match output from the controller can be observed at TP4 and the same match output can be used as a capture input to the controller by shorting JP14. Or on external capture signal can also be interfaced by providing the input at pin 1 of RM4. During this condition open JP14.

An external interrupt to the controller is generated using the switch SW2. Pressing this switch generates edge or low level sensitive interrupt to the controller.

CNB1 has to be used for interfacing this block with the controller. At controller side CNB needs to be used for interfacing this block.

Pin no CNB1	Description
1	PWM output from controller



USER MANUAL

2, 5, 6, 8, 9	No connection
3	Match output from controller
4	Capture input to the controller
9	External interrupt to the controller

Seven segment display interface:

There are four multiplexed 7-segment display units (**U8, U9, U10 and U11**) on the board. Each display has 8-inputs SEG_A (Pin-7), SEG_B (Pin-6), SEG_C (Pin-4), SEG_D (Pin-2), SEG_E (Pin-1), SEG_F (Pin-9), SEG_G (Pin-10) and SEG_H (Pin-5) and the remaining pins pin-3 & pin-8 are Common Cathode CC. These segments are common cathode type hence active high devices.

At power on all the segments are pulled up. A four bits input through CNB2 is used for multiplexing operation. A 1-of-10 Decoder/Driver U7 is used to accept BCD inputs and provide appropriate outputs for enabling the required display.

8 bits data is provided to this block using CNA2. All the data lines are taken buffered at U12 before giving to the displays.

Pin no CNB2	Description
1	Input A to U7
2	Input B to U7
3	Input C to U7
4	Input D to U7
5 to 9	No connection
10	Ground

At controller end any 2 connector are required for interfacing this block.

I2S Interface:

This hardware is made for simple demonstration of I2S. Here receive, clock and word select lines of receive and transmitter are terminated at CN17 - 7 pin male berg. CNA4 is used to interact this block with controller. At controller side CNA needs to be used to interface this block. LED LD16 is provided at this block for general use.

Pin no CNA4	Description	Pin @ CN17
1	Receive clock	1
2	Receive word select	2
3	Receive SDA	3
4	Transmit clock	4
5	Transmit word select	5

6	Transmit SDA	6
7	LED Anode	-
10	Ground	-
8,9	No connection	-

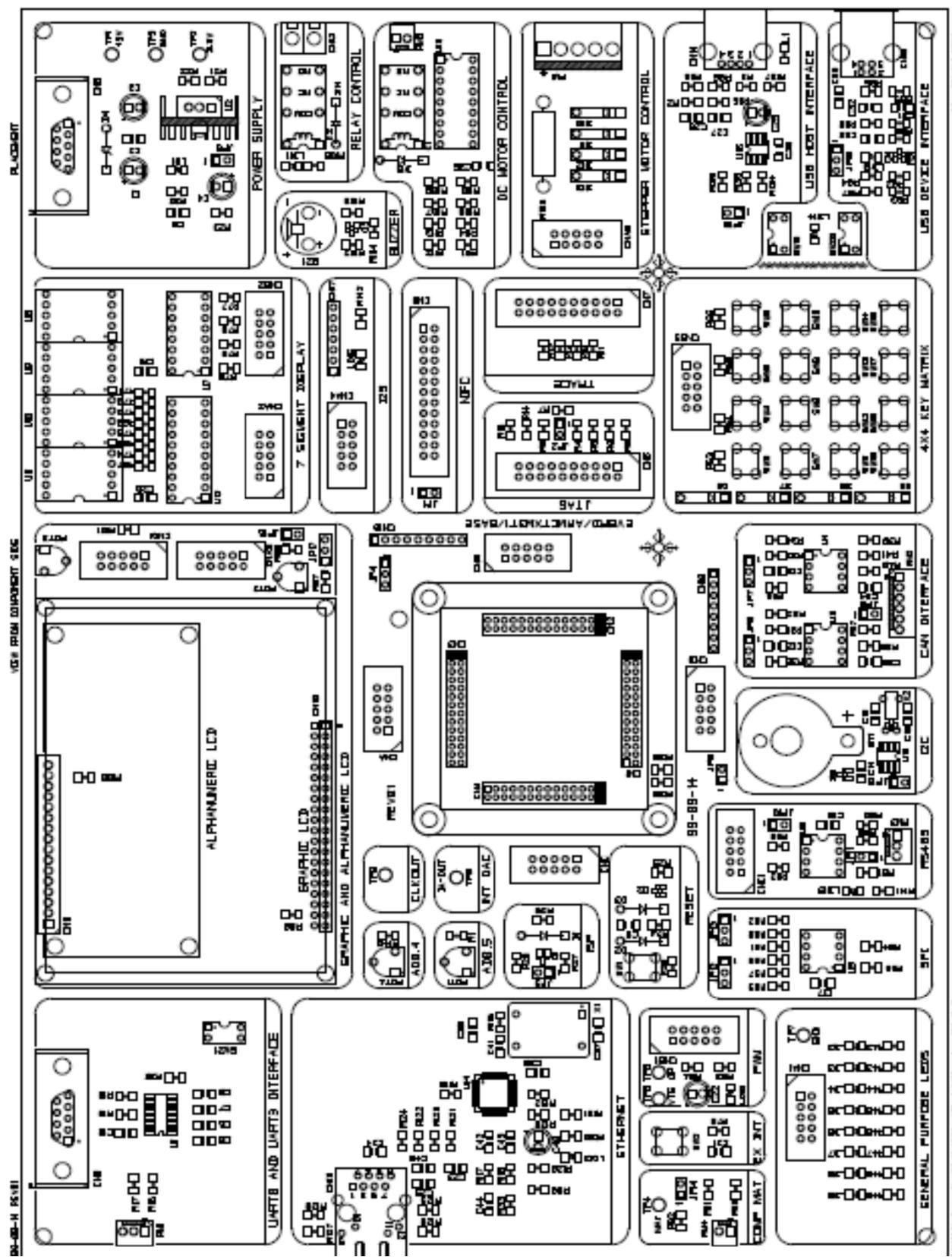
7th pin of CN17 and LED cathode are grounded. At CN17 pin number 1 is at CNA4 side and pin 7 is at buzzer side.

Spare connector:

Few unused lines of the which are not used anywhere on the board are terminated at a connector – CN16. It's a 9 pin male straight berg. These lines can be used for any external interface if required. Pin descriptions of this connector is given below. Note that CN16 has pin number 1 at JP17 side and last pin at CNB side.

Pin no	Description
1	P1.20/MCI0/PWM1.2/SCK0
2	P1.21/MCABORT/PWM1.3/SSEL0
3	P1.28/MCOA2/PCAP1.0/MAT0.0
4	P1.29/MCOB2/PCAP1.1.MAT0.1
5	P3.25/MAT0.0/PWM1.2
6	P3.26/STCLK/MAT0.1/PWM1.3
7	P4.28/RX_MCLK/MAT2.0/TXD3
8	P4.29/TX_MCLK/MAT2.1/RXD3
9	VBAT – Battery backup for internal RTC

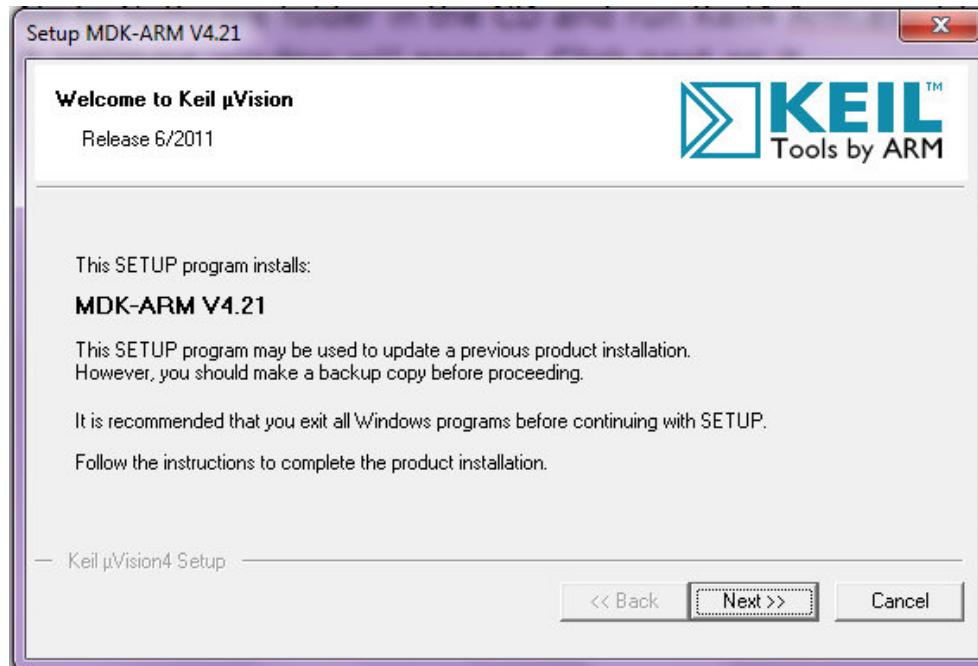
5. PLACEMENT DIAGRAM



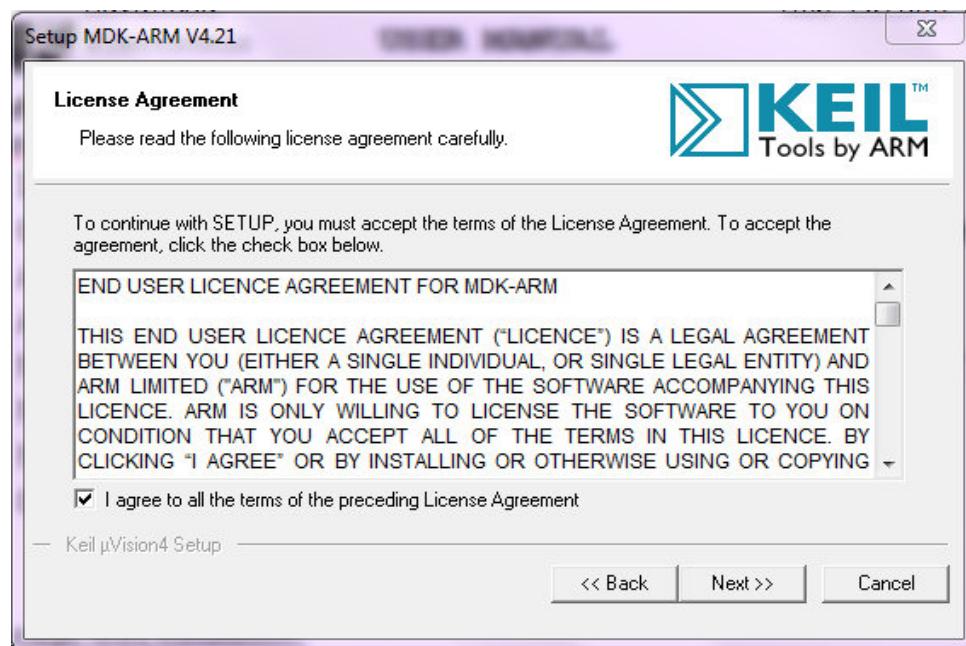
6. Software/Firmware

6.1 Keil uvision4 ide Installation:

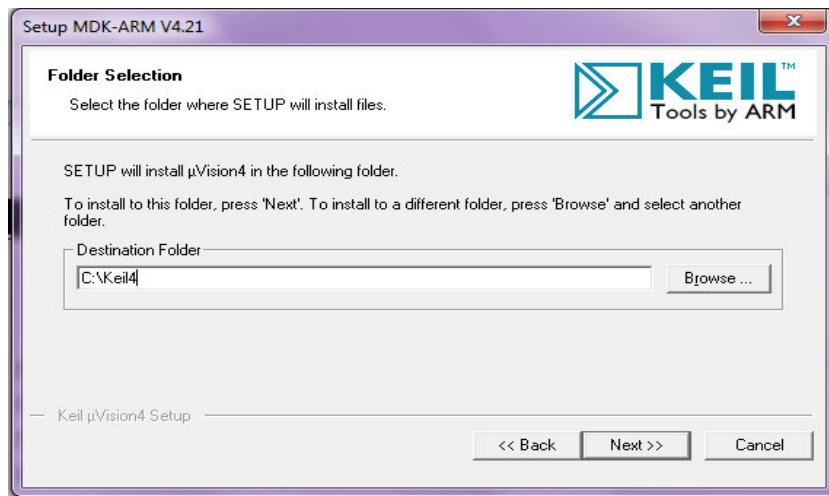
- Installation of keiluvision4 as follows.
- Go to Software folder in the CD and run Keil4Arm.exe file.
- A welcome window will appear. Click next on it



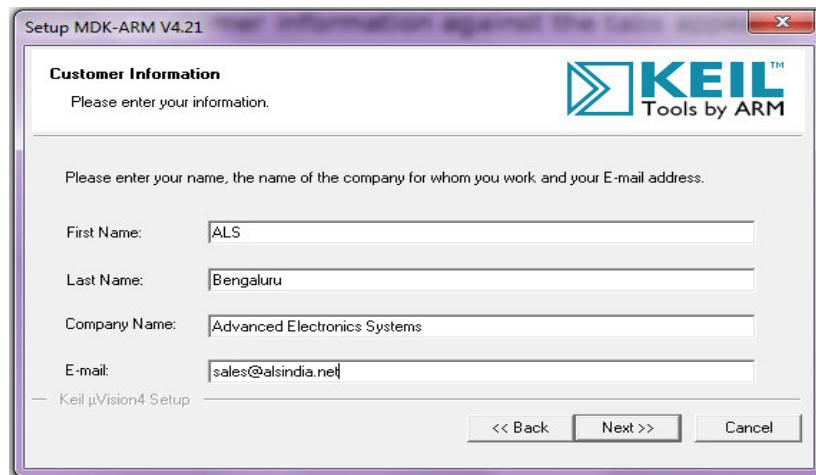
- A license window will appear. Click next



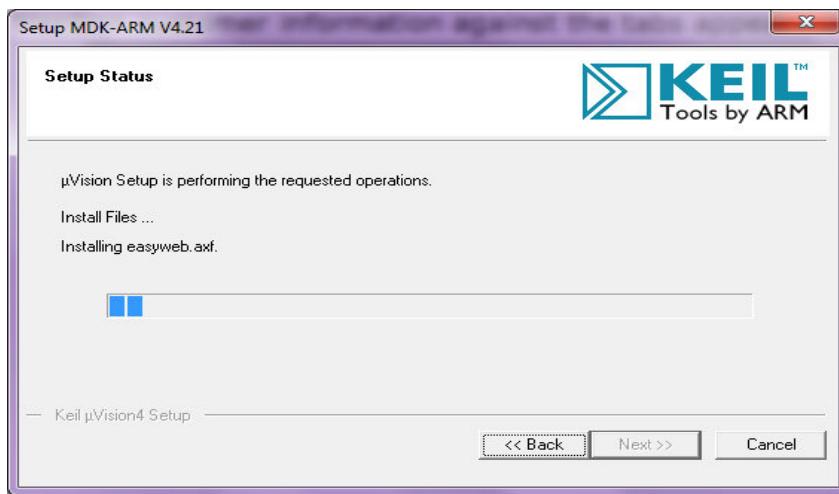
- Create a folder Keil4 at C drive to install software



- Mention customer information against the tabs appear



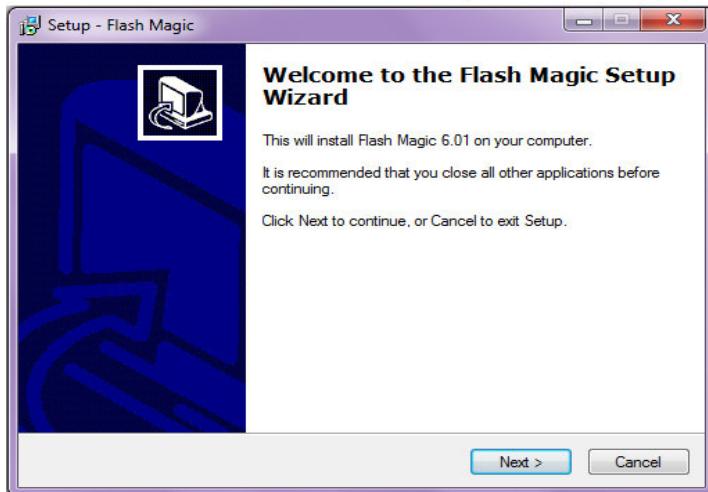
- After clicking next a setup status will appear



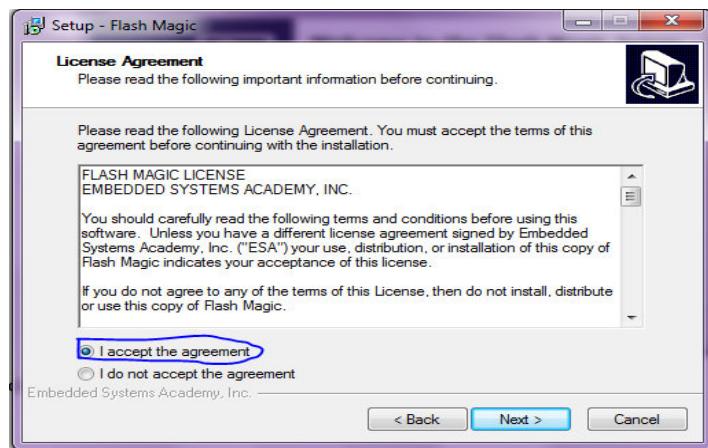
- Click next and finish

6.2 Flash magic 6.01 installation:

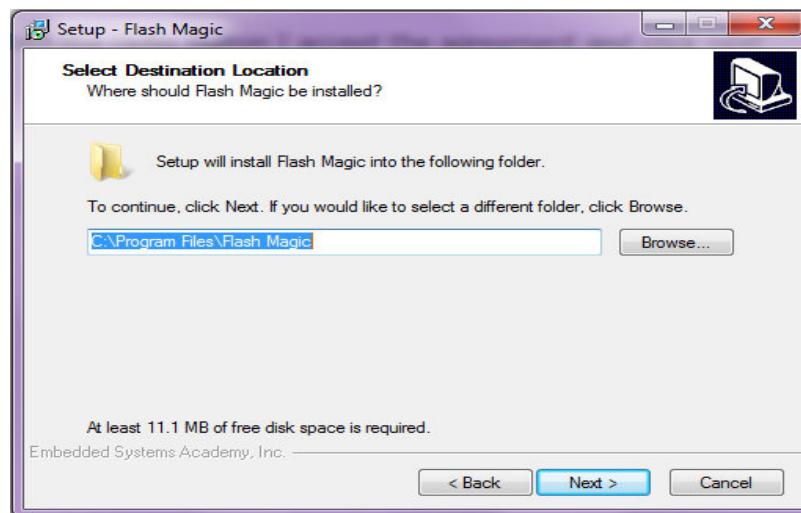
- Go to Software folder in the CD and run FlashMagic.exe file.



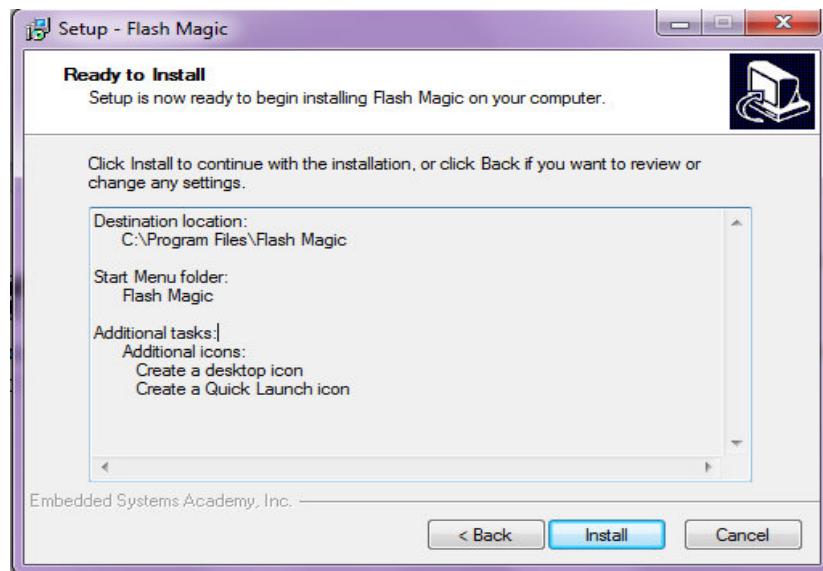
- Click next on Welcome wizard



- Select the radio button I accept the agreement and click next



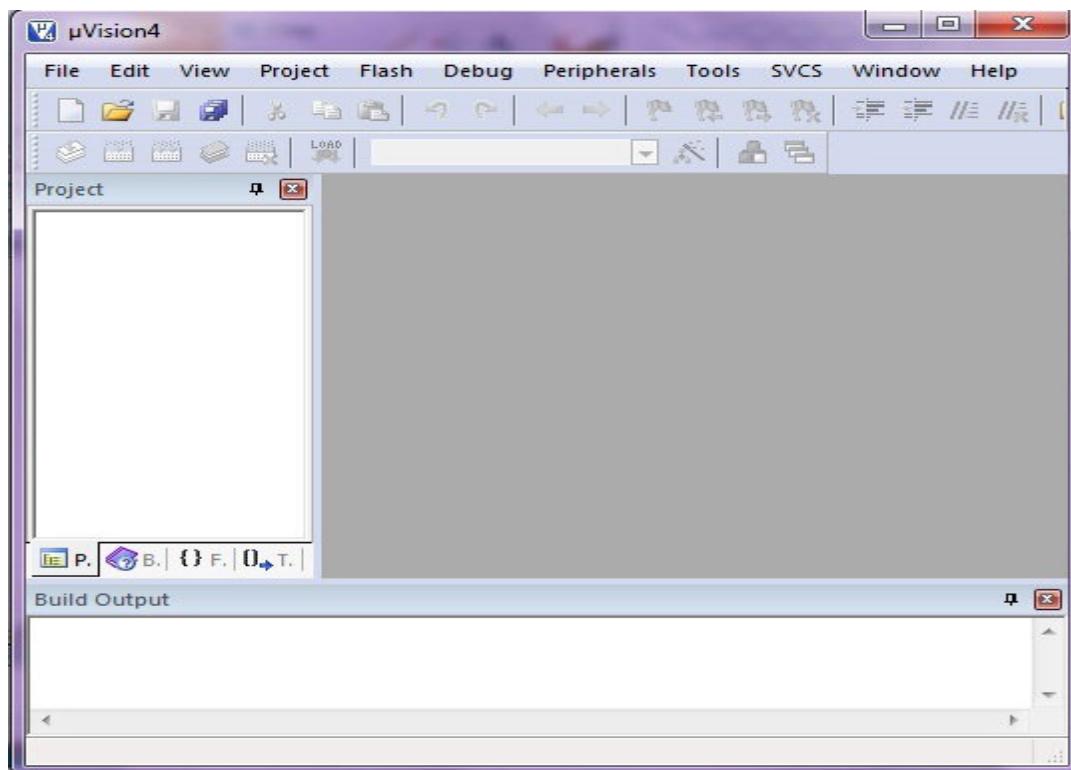
- Choose a folder to install the files. Click next and choose the option create short cut icon and click next. Displays the options we have selected



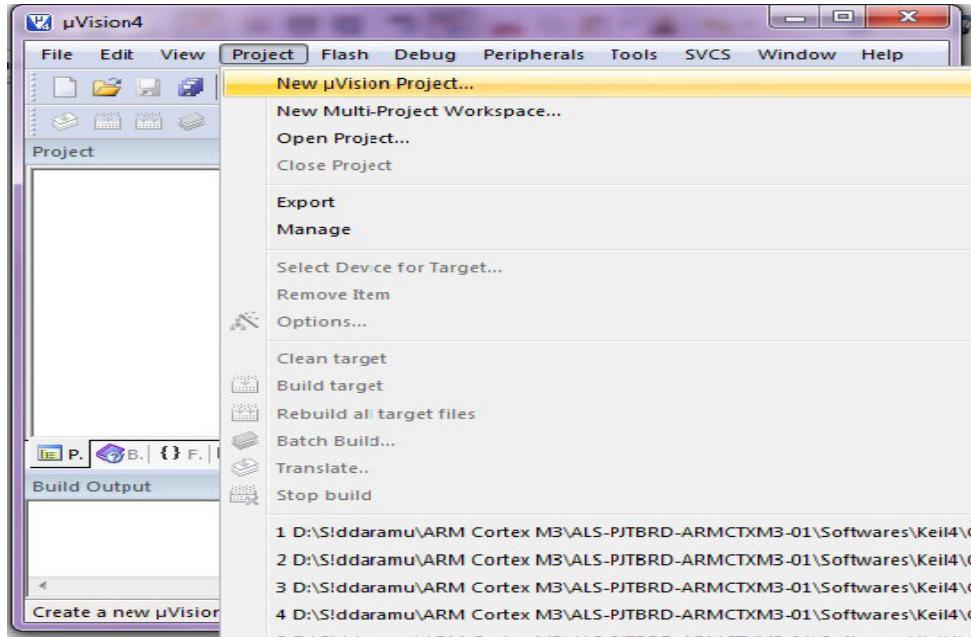
- Click on install and finish

6.3 Project Creation in Keil uvision4 IDE:

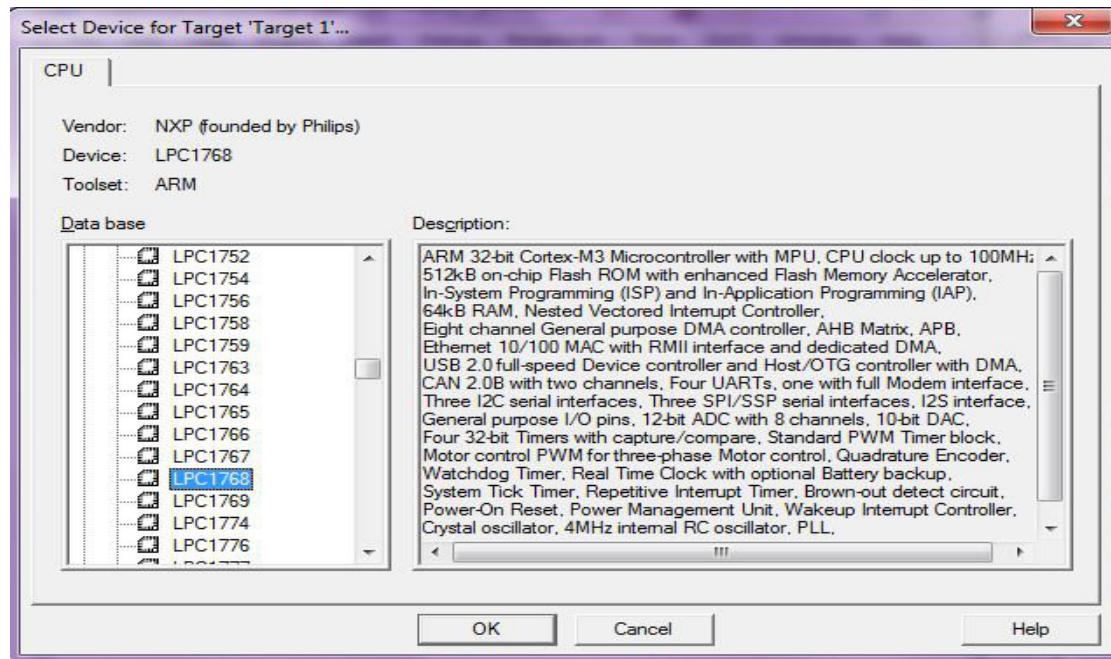
- Create a project folder before creating NEW project.
- Use separate folder for each project
- Open Keil uVision4 IDE software by double clicking on "Keil Uvision4" icon.

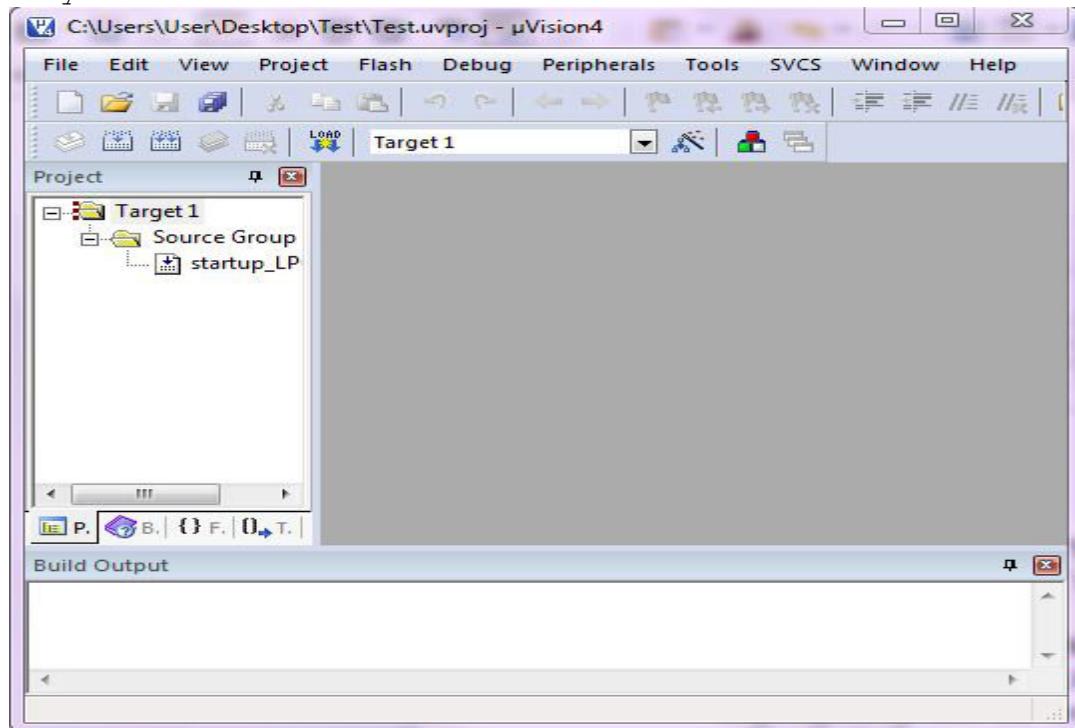


- Go to “Project” then to “New Project” and save it with a name in the Respective Project folder, already you created.

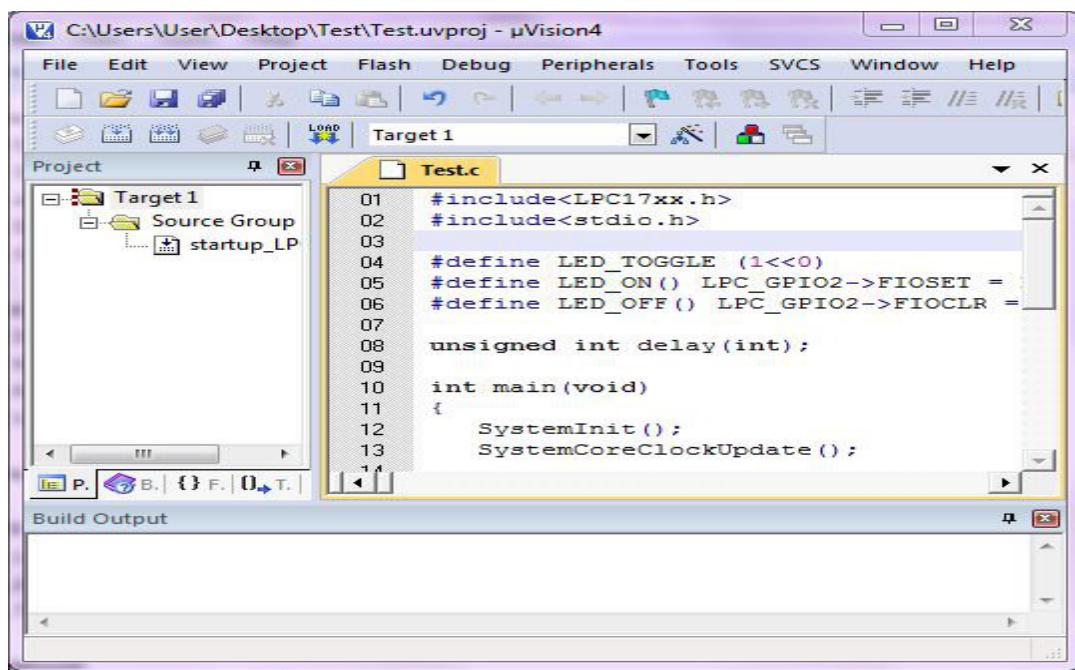


- Select the device as “NXP (founded by Philips)” In that “LPC1768” then Press OK and then press “YES” button to add “system_LPC17xx.s” file.





- Go to “File” In that “New” to open an editor window. Create your source file And use the header file “LPC17xx.h” in the source file and save the file. Color syntax highlighting will be enabled once the file is saved with a Recognized extension such as “.C ”.



USER MANUAL

- Right click on "Source Group 1" and select the option "Add Files to Group 'Source Group 1' "add the. C source file(s) to the group.
 - Again right click on Source Group 1 and select the option "Add Files to Group 'Source Group 1' "add the file –
C:Keil(4)\ARM\startup\NXP\LPC17xx\system_LPC17xx.c
 - Any changes made to this file at current project will directly change the source system_LPC17xx.c file. So that other project settings may get altered. So **it is recommended** to copy the file
C:Keil(4)\ARM\startup\NXP\LPC17xx\system_LPC17xx.c to the project folder and add to the source group.
- Important:** This file should be added at each project creation.
- Then go to "Project" in that "Translate" to compile the File (s).
 - Go to "Project" in that "Build Target" for building all source files such as ".C", ".ASM", ".h", files, etc...This will create the hex file if no warnings & no Errors.

6.3.1 Some Settings to be done in KEILUV4 for Executing C programs:

- In Project Window Right click "TARGET1" and select "options for target 'TARGET1'
- Then go to option "Target" in that
 1. Xtal 12.0MHz
 2. Select IROM1 (starting 0x0 size 0x8000).
 3. Select IRAM1 (starting 0x10000000 size 0x8000).
- Then go to option "Output"
Select "Create Hex file".
- Then go to option "Linker"
Select use memory layout from target dialog

6.3.2 Settings to be done at configuration wizard of system_LPC17xx.c file

- Before configuring the clock registers study the block diagram which is described in the Clocking and power control in chapter 2.
- There are three clock sources for CPU. Select Oscillator clock out of three. This selection is done by CLKSRCSEL register.
- If we disable the PLL0 System clock will be bypassed directly into CPU clock divider register.
- Use CCLKCFG register for choosing the division factor of 4 to get 3M Hz out of 12 M Hz Oscillator frequency.
- For any other peripherals use the PCLK same as CCLK.

Follow the below mentioned procedure to do these settings.

- Double click on system_LPC17xx.c file at project window
- Select the configuration wizard at the bottom
- Expand the icons

- Select Clock configuration
- Under System controls and Status registers
 - OSCRANGE: Main Oscillator range select 1MHz to 20MHz
 - OSCEN: Main oscillator enable ✓
- Under Clock source select register (CLKSRCSEL)
 - CLKSRC: PLL clock source selection Main oscillator
- Disable PLL0 configuration and PLL1 configuration
- Under CPU Clock Configuration register(CCLKCFG)
 - CCLKSEL: Divide value for CPU clock for PLL0 4
- Under USB Clock configuration register (USBCLKCFG)
 - USBSEL: Divide value for USB clock for PLL0 4
- Under Peripheral clock selection register 0 (PCLKSEL0) and 1 (PCLKSEL1)
 - select Pclk = Cclk for all.
- Under Power control for peripherals (PCONP)
 - Enable the power for required peripherals
- If CLKOUT to be studied configure the Clock output configuration register as below

CLKOUTSEL	: Main Oscillator
CLKOUTDIV	: 1
CLKOUT_EN	: ✓
- Call the functions
 - SystemInit();
 - SystemCoreClockUpdate();

at the beginning of the main function without missing. These functions are defined in `system_LPC17xx.c` where actual clock and other system control registers configuration takes place.

- A small change is required in the file `system_LPC17xx.c` after installation. Go to text editor:


```
#define PLL0_SETUP      0
#define PLL1_SETUP      0
```

if the above #defines are 1 then make 0
- In void SystemInit (void) function, in the condition
`#if (CLOCK_SETUP)`
 After the instructions
`LPC_SC->PCLKSEL0 = PCLKSEL0_Val; /* Peripheral Clock Selection */`
`LPC_SC->PCLKSEL1 = PCLKSEL1_Val;`
 add the below mentioned instruction.
`LPC_SC->CLKSRCSEL = CLKSRCSEL_Val;`
 Note: this instruction is written under the condition `#if PLL0_SETUP` by default.

7. TEST SET UP & TEST PROCEDURE:

7.1 TEST SET UP REQUIREMENTS:

- ALS-SDA-ARMCTXM3-01 : 1 No.
- Power supply (+5V) : 1 No.
- Cross cable for programming and serial communication : 1 No
- Few flying leads
- Kiel uvision4 and flash magic 6.01
- One working COM port (Ex: COM1) in the host computer system and PC for downloading the software.
- 2 numbers 10 core FRC cables of 8 inch length
- USB to B type cable
- 2/4GB pen drive
- 1:1 Ethernet enhanced cable.

7.2 TEST SET UP:

- Connect +5V power cable to CN8.
- Do not insert the piggyback module into the board before testing the voltage levels.
- **Remove JP6** and then switch on the power supply.
- Check 3.3V, +5V at TP1 and TP2.
- If voltages are at right level then check if any IC's getting warm.
- Then switch off the power supply, insert the piggyback module, short JP6 and switch on the power.
- Also make sure the regulator output which is flowing into the controller is 3.3V

7.3 FLASH PROGRAMMING

1. Connect 9 pin DSUB cross cable from PC to CN9 at the board.
2. On the 2 way dip switch SW21.
3. Short jumper JP3
4. Open flash magic 6.01

5. Some Settings in FLASH MAGIC:

Step1. Communications:

Device : LPC1768
Com Port : COM1
Baud Rate : 9600
Interface : None(ISP)
Oscillator : 12MHz

Step2. ERASE:

Select "Erase Blocks Used By Hex File".

Step3. Hex file:

Browse and select the Hex file which you want to download.

Step4. Options:

Select "Verify After Programming".

Go to **Options -> Advanced Options->communications**

Do not select **High Speed Communications**

keep baud rate 115200.

Options -> Advanced Options->Hardware config



Advanced
Electronic
Systems

ALS-SDA-ARMCTXM3-01-UM1

USER MANUAL

Select **Use DTR & RTS to control RST & ISP Pin.**

Select **Keep RTS asserted while COM Port open.**

T1 = 50ms. T2 = 100ms.

Step5. Start:

Click **Start** to download the hex file to the controller.

USER MANUAL

7.4 TESTING OF SAMPLE PROGRAMS:

- Do the connection as mentioned above test set up.
- Switch ON power supply.
- Download the respective source code.hex file using UART0 (CN9).

Port Lines Test:

Software: This application is for testing the working condition of port lines. Download the file PORT_TEST.hex.

Hardware Setup: Connect one end of 10 pin FRC cable to CNA1. Short other end to any one among CNA, CNB, CNC, CND. Short JP4(1, 2) & JP5.

Working procedure: Connect a cable from all the (GPIO) connectors to CNA1. At all 4 different connections LED's from LD2 to LD9 will toggle.

UART0 TEST:

Software: Download the file **UART0_Test.hex**.

This project is for studying the UART0 communication. Code is written in such a way that, a data is received from UART0 and same data is sent back to the same port.

Hardware setup: Connect a 9 pin DSUB cross cable from CN9 of the board to PC. Off the switch SW21. Use default properties for the hyper terminal at PC end.

Working procedure: Use PC communication terminal to see the working. A character typed at the PC key board is sent to the project board via Hyper terminal. And same character is sent back from the controller to hyper terminal. The data is displayed on Hyper terminal.

Alphanumeric LCD Test:

Software: Download the file **AN_LCD.hex**

Hardware setup: insert 16×2 small size LCD into CN11. Connect 10 pin FRC cable from CND to CNAD. Short the jumper JP16 & JP5. Use POT3 for contrast adjustment.

Working procedure: After software download and hardware setup, press the reset. A fixed message will display on LCD.

General Purpose LED's:

Software: Download the file **GP_LED.hex**

Hardware setup: Connect 10 core FRC cable from CNA to CNA1

Working procedure: LED's from LD2 to LD9 serially toggles.

Graphic LCD Test:

Software: Download the file **GLCD.hex**

Hardware setup: Connect 10 core FRC cable from CNA to CNAD and CND to CND1. Open JP16, short JP17(2, 3) and JP5. Open JP16. use POT2 for contrast adjustment. Insert 128×64 Graphic LCD into CN10.

Working procedure: After downloading the software and hardware setup, press reset. A fixed message will display on Graphic LCD.

UART3 Test:

Software: Download the file **UART3_Test.hex**

Hardware setup: Short the jumpers JP7(1, 2) and JP8(1, 2). Connect reliamate to DSUB cable from RM1 to 9 pin DSUB cross cable.

Pin no reliamate	Pin no DSUB female connector
1	3



2	2
3	5

Connect other end of female DSUB connector to male end of 9 pin DSUB cross cable. Connect other end of the cross cable to PC. Use default properties at for PC hyper terminal.

Working procedure: Use PC communication terminal to see the working. A character typed at the PC key board is sent to the project board via Hyper terminal. And same character is sent back from the controller to hyper terminal. The data is displayed on Hyper terminal.

Seven Segment Display:

Software: Download the file **Seven_seg.hex**

Hardware setup: Connect a 10 core FRC cable from CNA to CNA2 and CNB to CNB2.

Working procedure: Press the reset after downloading the code. Observe the count from 0000 to 9999 on the display.

4x4 key matrix:

Software: Download the code **KEY_LCD.hex**

Hardware setup: Connect 10 core FRC cable from CNB to CNB3, short JP4(1, 2) and also do the hardware setup related to AN LCD.

Working procedure: Identity of key pressed (0 to F) will be displayed on LCD.

ADC0.5 Test:

Software: Download the file **Int_ADC5.hex**

Hardware Setup: Do the setup related to Alphanumeric LCD

Working procedure: Vary POT1 and observe the corresponding analog and digital voltage values on LCD.

ADC0.4 Test:

Software: Download the file **Int_ADC4.hex**

Hardware Setup: Do the setup related to Alphanumeric LCD. Short JP18(2, 3) at USB device block.

Working procedure: Vary POT4 and observe the corresponding analog and digital voltage values on LCD.

Internal DAC:

Software: Download the file **Int_DAC.hex**

Hardware setup: Open the jumper JP5

Working procedure: observe the analog output at TP8 using multimeter or CRO with reference to ground.

Internal Clock test:

Software: Download the file **CLK_Test.hex**

Hardware setup: Open the jumper JP19

Working procedure: Observe the clock pulse at TP9 with reference to ground.

Internal RTC Test:

Software: Download the file **Int_RTC.hex**

Hardware setup: Make UART0 communication setup.

Working: According to this software, Internal RTC is operated based on the commands sent from PC terminal through UART0. A read and write operation is done and read values are sent to serial port UART0. Until **Esc** key pressed at PC key board a read operation will continue.

USER MANUAL

BUZZER TEST:

Software: Download the file **Buzzer.hex**

Hardware setup: Connect 10 pin FRC cable from CNA to CNA5

Working procedure: After pressing reset, buzzer starts sounding.

RELAY CONRTOL Test:

Software: Download the file **Relay.hex**

Hardware setup: Connect 10 pin FRC cable from CNA to CNA5. If this control output is needed externally, use CN12.

Working procedure: Observe the toggling of Relay K2 and LED LD11.

DC motor Direction control:

Software: Download the file **DCM.hex**

Hardware setup: 10 pin FRC cable from CNA to CNA5. Connect DC motor supply lines to the reliamate RM5.

Working procedure: Alternatively DC motor will rotate in clockwise and anticlockwise direction.

Stepper Motor Control:

Software: Download the file **STPM.hex**

Hardware setup: Connect 10 pin FRC cable from CNA to CNA5. Connect the stepper motor to PM1.

Working procedure: Stepper motor will rotate clockwise and in anti clock wise direction automatically after reset.

External Interrupt Test:

Software: Download the file **EXINT.hex**

Hardware setup: Connect 10 pin FRC cable from CNB to CNB1

Working procedure: Press switch SW2 to generate external interrupt. At each press LD10 will toggle.

PWM Test:

Software: Download the file **PWM.hex**

Hardware setup: Connect 10 pin FRC cable from CNB to CNB1.

Working procedure: As the pulse width varies, intensity of LED LD10 varies. Observe the pulses at TP5. Observe the amplitude level at TP6

Capture and Match Test:

Software: Download the file **COMP_MAT.hex**.

Hardware setup: Connect 10 pin FRC cable from CNB to CNB1. Short JP14. Optionally Use RM4 to feed external pulses for counting and open JP14. Use TP4 to watch the pulses on CRO. Also do the LCD related Hardware setup.

Working procedure: Observe the Pulse width on LCD. Change the MR1 value of timer1 to vary the pulse width.

SPI EEPROM Test:

Software: Download the file **SPI_Test.hex**

Hardware setup: Connect 10 pin FRC cable from CNC to CNC1 and short JP12(2,3) and JP13(2, 3). Also do the hardware setup related to UART0 communication test.

Working procedure: After resetting the controller, a message will display on the PC hyperterminal. Depending on the selection from the user, operation will get performed in the controller. For read operation, read data will display on the hyperterminal. For write operation, an acknowledgment message will get displayed after write.

USER MANUAL

I2C RTC Test:

Software: Download the file **I2C_RTC.hex**

Hardware setup: Connect 10 pin FRC cable from CNC to CNC1. Short JP15(1, 2) and JP4(2, 3). If clock out pin of the RTC is using as an interrupt, short JP15 and JP4(2, 3). Keep CN18 free. Also do the settings required for UART0 communication.

Working procedure: After downloading the hex file, do the hardware setup and reset the controller. A message will display on hyperterminal. Depending upon the selection, operation is being done at controller. If time is read it continuously update on the hyper terminal at each interrupt, till **Esc** key is pressed. If time is set, an acknowledgment message will get displayed.

RS485 Protocol Test:

RS485 project setup has two ends; Transmitter and Receiver. Keep two ALS-SDA-ARMCTXM3-01 for testing. Assume any one as a master and other one as a slave.

Software: Download the file **RS485_Tx.hex** for master and **RS485_Rx.hex** for slave.

Hardware setup: Connect 10 pin FRC cable from CNC to CNC1 at both the boards. Short JP13(1, 2) and JP12(1, 2) at both the boards. Connect 1:1 3 pin relamate between RM3 of both the boards or connect 3 flying leads from RM3 of master to RM3 of slave in 1:1 format. Short JP11 at both the boards. Open JP10 at master end and short JP10 at slave end.

Working procedure: After above steps, press reset at master. After sending the data from master, LD15 gets on. After receiving the data at slave, LD15 gets on.

Note: For details on the operation of RS485 protocol, refer **15.4.21 RS-485/EIA-485 modes of operation** in LPC17xx user manual **UM10360**.

I2S_Test:

Software: Download the file **I2S.hex**

Hardware setup: Connect 10 pin FRC cable from CNA to CNA4. At CN17 connect flying leads from pin 1 to 4, 2 to 5, 3 to 6. Count the pin numbers at CN17 from CNA4 side in ascending order.

Working procedure: After executing a code LD16 glows if there is a successful communication.

CAN Test:

Software: Download the file **CAN.hex**

Hardware setup: Connect 10 pin FRC cable from CNA to CNA1. Short JP7(2, 3) and JP8(2, 3). Short JP9. Connect a flying leads in RM2 from pin1 to pin5 and from pin2 to pin6.

Working procedure: This application check the TX and RX channel of the CAN 1 and 2. Send one message from CAN1(TX) and verify received message on CAN2(RX), if it's a match, both CAN TX and RX are working. If the message sent from CAN1 matches the received message of CAN2, then LD2 to LD9 gets ON If the Doesn't matches LED's will be off.

USB Device Test:

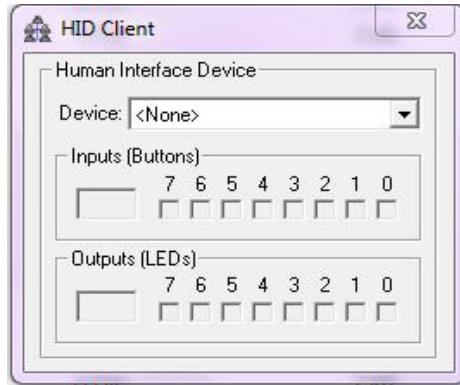
Note: We can use either Device or Host or OTG at a time. We cant use more than one USB feature at same time.

Software: Download the file **USBHID.hex**

Hardware setup: On pin 1 & 2 at switches SW20. Connect USB to B type cable from CN15 to PC USB port. Check LD13 and LD14 are On. Connect 10 pin FRC cable from CNA to CNA1 and from CNB to CNB3. Short JP18(1, 2) and JP4(1, 2).

Working procedure: After the above steps, a device will get detected from the PC and device driver software get automatically installed in the PC. And the device (controller) get identification as a **HID – compliant device** under **Human Interface Devices**. Check at Computer → Manage.

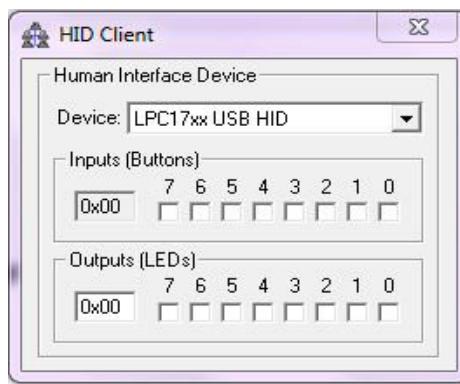
Then open **HIDClient.exe** software in the USBHID project folder.



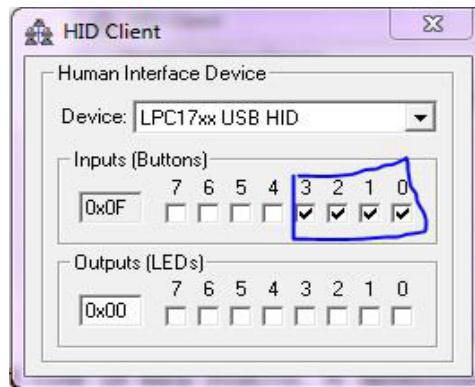
In this utility select the device **LPC17xx USB HID**.



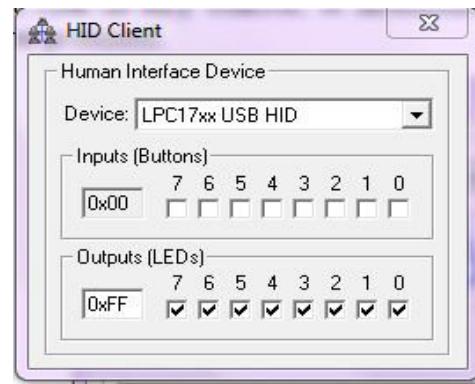
After selection of the device, the window looks like this.



Now press a key at first row of key matrix. A check appears at the corresponding I/p of Input buttons while the key is pressed as shown below

USER MANUAL

Now check a box from 0 to 7 in the above window corresponding to Output(LED's). A corresponding LED from LD2 to LD9 will glow on the board.



The above two steps indicates simple communication of the controller operation as a Human interface input/output device.

USB Host Test:

Software: Download the file **USBHost.hex**

Note: A pen drive used in this test can be FAT16 formatted. A pen drive with less than or equal to 4GB only supports the FAT16 format.

Procedure to format the pen drive to FAT16:

If PC operating system is Windows7:

Connect a pen drive. Open **my computer**. Right click on pen drive icon. Go to **Format**. Under **File system** select **FAT(Default) or FAT16**. Under **Allocation unit size** select **Default allocation size** (16 kilobytes). Click **start**.

If PC operating system is windows XP:

Connect a pen drive. Open **my computer** and note weather the pen drive has been detected under Removable disk drives or Hard disk drives. Also note the Volume name and Volume label. If pen drive is detected as ALS(G:), ALS – Volume label and G – is Volume name.

If pen drive has been detected under the Removable Hard Drives, follow the below steps to format the pen drive. Open the command window. Type the command →

FORMAT <Volume name>:/FS:FAT16

A message will appear at command prompt

**Insert new disk for drive <Volume>:
and press enter when ready...**

Press **ENTER**.

Sometime it may shows the message

The new file system is FAT16.

FORMAT is not available for FAT16 drivers.

In such cases use the command:

FORMAT <Volume name>:/FS:FAT

Press **ENTER** twice.

A message – **Volume label** will appear. Type a name to the device and press enter, or else just press enter.

A new format information will display.

Below snap shows an example for formatting 2GB pen drive. Volume name – I.

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\als>FORMAT I:/FS:FAT16
Insert new disk for drive I:
and press ENTER when ready...
The type of the file system is FAT.
The new file system is FAT16.
FORMAT is not available for FAT16 drives.

C:\Documents and Settings\als>FORMAT I:/FS:FAT
Insert new disk for drive I:
and press ENTER when ready...
The type of the file system is FAT.
Verifying 1926M
Initializing the File Allocation Table (FAT)...
Volume label (11 characters, ENTER for none)?
Format complete.

2,019,262,464 bytes total disk space.
2,019,262,464 bytes available on disk.

    32,768 bytes in each allocation unit.
    61,623 allocation units available on disk.

        16 bits in each FAT entry.

Volume Serial Number is 9CF7-27ED

C:\Documents and Settings\als>
```

If pen drives get detected under Hard Disk Drives(Next after the C, D drives) follow the below steps. Assume we are formatting the drive **ALS(H:)**

Open a command window and type a commands

FORMAT <Volume name>:/FS:FAT

Below message will appear

The type of the file system is FAT.

Enter current volume label for drive H:

Mention the label of the volume. Here it is ALS. The press enter. A warning will appear:

WARNING, ALL DATA ON NON-REMOVABLE DISK

DRIVE H: WILL BE LOST!**Proceed with Format (Y/N)?**

Once again open the My Computer and make sure, Entered details are right.
Press **Y**. and press Enter. A message will Appear.

Verifying 3818M

WARNING! The cluster size for this volume, 64K bytes, may cause application compatibility problems, particularly with setup applications. The volume must be less than 2048 MB in size to change this if the default cluster size is being used.

Proceed with Format using a 64K cluster (Y/N)?

Type **Y** and Press Enter. Below shown message will appear.

Initializing the File Allocation Table (FAT)...**Volume label (11 characters, ENTER for none)?**

Provide any label for the pen drive and Press enter. A details of newly formatted disk are shown.

Format complete.**4,004,184,064 bytes total disk space.****4,004,184,064 bytes available on disk.****65,536 bytes in each allocation unit.****61,099 allocation units available on disk.****16 bits in each FAT entry.****Volume Serial Number is BCB5-3E58**

After formatting, open a notepad. Type any message and save in the pen drive. Give the file name as **MSREAD.txt**. Remove the pen drive from PC.

Hardware setup: Switch on both the pins at SW19. Connect lesser than or equal 4GB size pen drive to CN14 which Is FAT16 formatted. Check LD14 is On and LED at pen drive(if available) is also On. Also do the communication setup for UART0 with baudrate of 57600. Short JP19.

Working procedure: Controller is host and accesses the pen drive. After flash programming and reset, controller copies the data in the file **MSREAD.txt**. Then controller creates the file **MSWRITE.txt** and puts the data read from MSREAD.txt to MSWRITE.txt. Open hyperterminal and press reset at board. A below shown log information will appear.

*Initializing Host Stack**Host Initialized**Connect a Mass Storage device*



USER MANUAL

Mass Storage device connected

Copying from MSREAD.TXT to MSWRITE.TXT...

Copy completed

Then remove the pen drive from board, insert to PC and verify.

Ethernet Test:

Software: A Local Area Network is needed to test this application. Note down the available IP address range of LAN. To find the LAN addresses range and Mask address open the command window in PC that is connected to the LAN you are working under. Type the command **ipconfig**. Choose any IP address which not currently being used by any PC's. And also note the subnet Mask. Open **EMAC** project. Open **tcpip.h** in project. There are following #defines.

```
#define MYIP_1
#define MYIP_2
#define MYIP_3
#define MYIP_4

#define SUBMASK_1
#define SUBMASK_2
#define SUBMASK_3
#define SUBMASK_4
```

If the IP address 90.0.0.70 is available, then alter the code like this

```
#define MYIP_1      90          // our internet protocol (IP) address
#define MYIP_2      0
#define MYIP_3      0
#define MYIP_4      70
```

Usually Ipv4 subnet mask will be 255.255.255.0. If there is change in MASK address edit it. If subnet mask is 255.255.254.0 then change like below

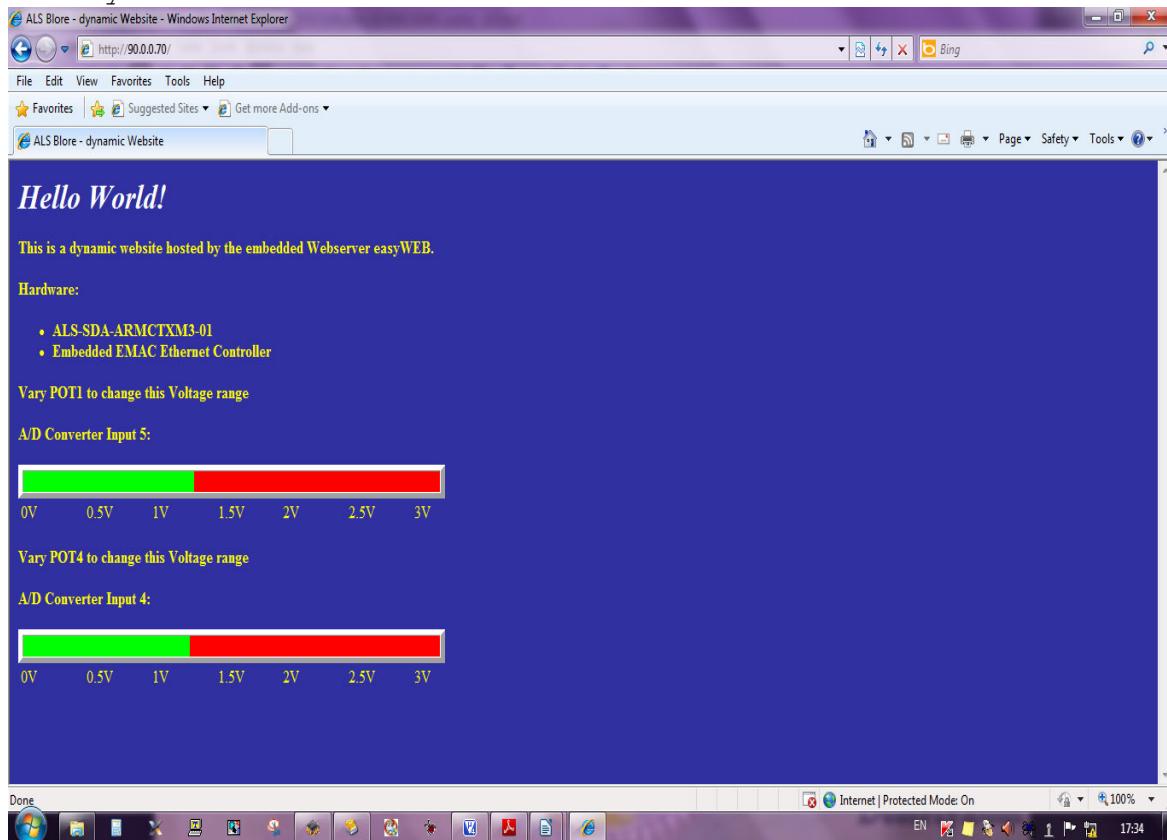
```
#define SUBMASK_1    255
#define SUBMASK_2    255
#define SUBMASK_3    254
#define SUBMASK_4    0
```

Save, build and rebuild all the project.

Download the file **EMAC.hex** to controller.

Hardware setup: Connect Ethernet cable from CN13 of the board to available port at hub. Short JP18(2, 3).

Working procedure: Use the PC which is connected to the LAN from same hub. Open the browser internet explorer. Type the IP address which has been entered in the file **tcpip.h** and press Enter. A HTML page will pop up. Vary the POT1 and POT5. As its analog voltage varies to the controller, same voltage information updates on the HTML page. Page will refreshes once in 5 seconds. Below snap is an example of the page which is accessing the IP 90.0.0.70



8. TROUBLE SHOOTING

1. Power Supply: Short jumper (**JP6**) +3.3V to Board.

2. In System Programming / Download (ISP):

In System Programming or download could not be established properly then check out whether the following conditions are met

The cable used for communication should be **cross cable**.

On both the pins of **SW21** for serial communication.

Short jumper **JP3 (ISP)**.

Open jumper **JP2 (RTCK)**.

IC MAX3232 is in good condition.

Check settings at flash magic.

3. JTAG Programming / Download:

Short jumper **JP2 (RTCK)** for communication.

Off both the pins of SW21.

Open jumper **JP3 (ISP)**.

4. General Problems:

Make Proper Jumper Connections as mentioned in Hardware Details.

Make Proper Connector Connections as mentioned in Demo Programs Setup.

9. BIBILOGRAPHY

- 1]. www.nxpsemiconductors.com
- 2]. User manual of LPC1768 – UM10306.pdf
- 3]. Application notes of LPC1768FBD100 at www.nxpsemiconductors.com
- 4]. www.keil.com
- 5]. Product data sheet of LPC1768 – DS_LPC1769_68_67_66_65_64_63.pdf
- 6]. Data sheets and application notes related to all the components from corresponding manufacturer's.