CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION TO PROJECT :

Traffic congestion deteriorates the quality of life of citizens and contributes significantly to environmental pollution and economic loss. Traffic information systems (TISs) aim at solving this problem by collecting traffic data, producing traffic estimates, and providing drivers with location-specific information. The increasing smart phone penetration, along with the wide coverage of cellular networks, defines an unprecedented large-scale network of sensors, with extensive spatial and temporal coverage, able to serve as traffic probes for TISs. To reap the benefits of smart phone-based TISs, users must participate in large numbers. Ideally, anyone possessing a smart phone should contribute to the TIS. Nevertheless, this very openness of such systems renders them vulnerable to adversaries and malicious users. It is thus necessary to secure the collection of data and render the contributing users (smart phones) accountable.

This is a task that cannot be achieved only by relying on the security of the mobile-to-cellular infrastructure communication. At the same time, as TISs require fine-grained location information, the privacy of the contributing participants must be protected. This need for privacy is intensified in the context of smart phone-based TISs. Smart phones already reveal a great deal of, possibly sensitive, information to the cellular operators (e.g., user identity, coarse-grained location, and calling/ messaging actions among others). Thus, it is important that the introduction of smart phone-based TISs does not, under any circumstances, deteriorate user privacy.

These points define a challenging tradeoff; although users should be able to participate in the system in an anonymous manner, they should be held, at the same time, fully accountable for their actions. Furthermore, the introduction of security and privacy protection mechanisms should neither deplete the user platform resources (i.e., computation resources, battery, and bandwidth) nor should it come at the expense of the TIS's efficiency and accuracy. Balancing security, privacy, effectiveness and efficiency is not straightforward. In most cases, the literature considers the aforementioned aspects separately, either overlooking security and privacy and focusing on the traffic estimation aspects of TISs or considering security and privacy without evaluating their effect on the efficiency and the accuracy of the TIS. This sets the challenge

ahead: *Can we leverage smart phones and build efficient, secure, and privacy-preserving TISs of unprecedented*

spatial coverage?

Contributions: We meet this challenge by addressing security and privacy protection aspects of smart phone-based TISs. Moreover, we assess their effect on the accuracy of traffic estimation. More specifically, building on our prior work [1]– [3], we present a smart phone-based TIS and assess its accuracy through Global Positioning System (GPS) traces in the presence of traffic estimation errors and for different values of location reporting rates and accumulation frames. Furthermore, by leveraging cellular providers, existing telecommunication standards and state-of-the-art cryptographic schemes, we propose a comprehensive security and privacy-preserving architecture, resilient against offending users and TISs entities. We formally assess the security and privacy properties of the system and demonstrate its efficiency through extensive evaluations.

CHAPTER 2
EMBEDDED SYSTEMS

2.1 INTRODUCTION TO EMBEDDED SYSTEMS :

Many embedded systems have substantially different design constraints than desktop computing applications. No single characterization applies to the diverse spectrum of embedded systems. However, some combination of cost pressure, long life-cycle, real-time requirements, reliability requirements, and design culture dysfunction can make it difficult to be successful applying traditional computer design methodologies and tools to embedded applications. Embedded systems in many cases must be optimized for life-cycle and business-driven factors rather than for maximum computing throughput. There is currently little *tool* support for expanding embedded computer design to the scope of holistic embedded system design. However, knowing the strengths and weaknesses of current approaches can set expectations appropriately, identify risk areas to tool adopters, and suggest ways in which tool builders can meet industrial needs. If we look around us, today we see numerous appliances which we use daily, be it our refrigerator, the microwave oven, cars, PDAs etc. Most appliances today are powered by something beneath the sheath that makes them do what they do. These are tiny microprocessors, which respond to various keystrokes or inputs. These tiny microprocessors, working on basic assembly languages, are the heart of the appliances. We call them embedded systems. Of all the semiconductor industries, the embedded systems market place is the most conservative, and engineering decisions here usually lean towards established, low risk solutions. Welcome to the world of embedded systems, of computers that will not look like computers and won't function like anything we are familiar with.

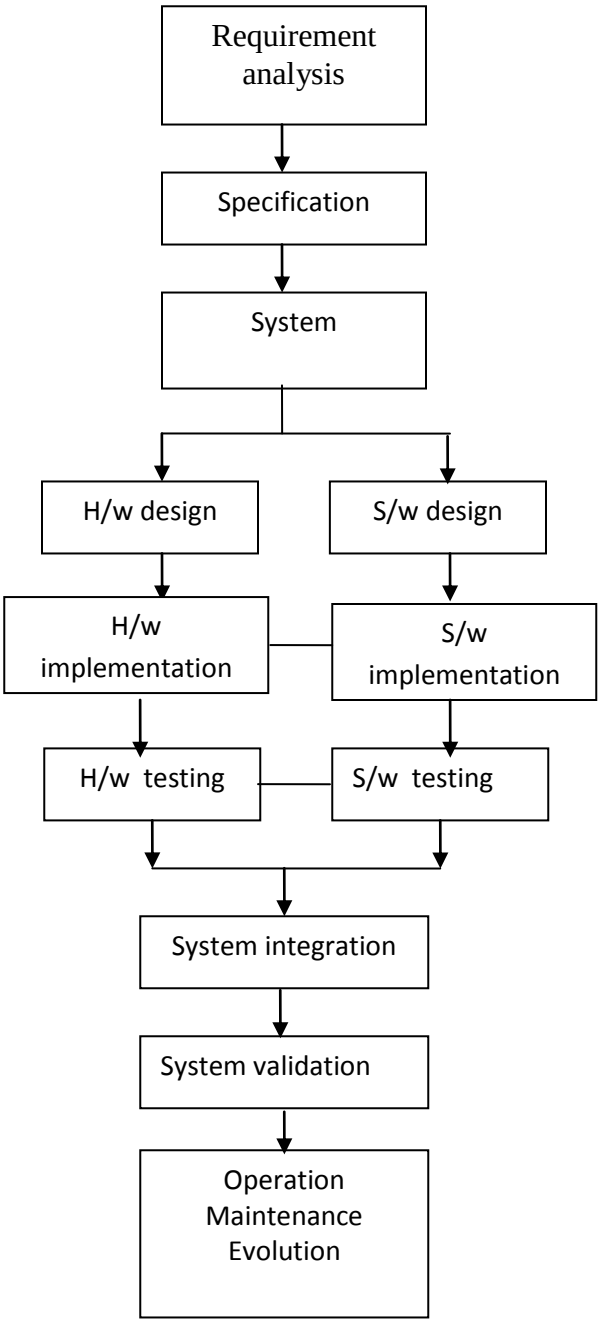


Fig 2.1: Embedded Development Life Cycle

2.2 CLASSIFICATION

Embedded systems are divided into autonomous, realtime, networked & mobile categories.

Autonomous systems

They function in standalone mode. Many embedded systems used for process control in manufacturing units& automobiles fall under this category.

Real-time embedded systems

These are required to carry out specific tasks in a specified amount of time. These systems are extensively used to carry out time critical tasks in process control.

Networked embedded systems

They monitor plant parameters such as temperature, pressure and humidity and send the data over the network to a centralized system for on line monitoring.

Mobile gadgets

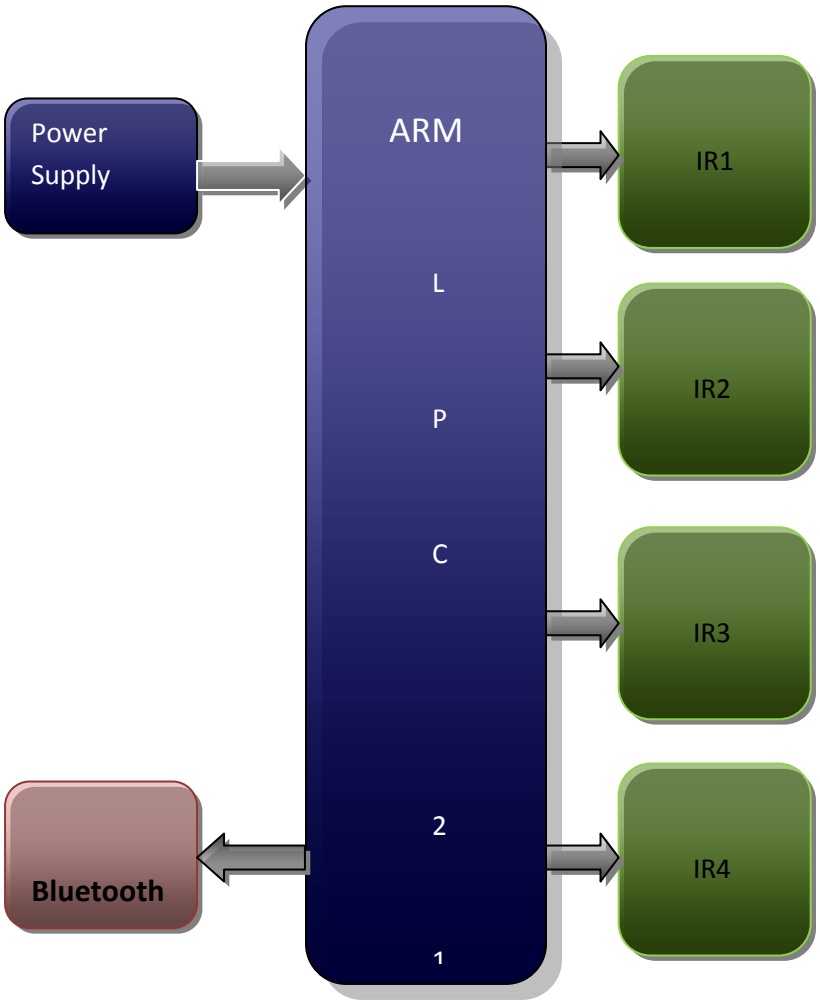
Mobile gadgets need to store databases locally in their memory. These gadgets imbibe powerful computing & communication capabilities to perform realtime as well as nonrealtime tasks and handle multimedia applications. The embedded system is a combination of computer hardware, software, firmware and perhaps additional mechanical parts, designed to perform a specific function. A good example is an automatic washing machine or a microwave oven. Such a system is in direct contrast to a personal computer, which is not designed to do only a specific task. But an embedded system is designed to do a specific task with in a given timeframe, repeatedly, endlessly, with or without human interaction.

Hardware

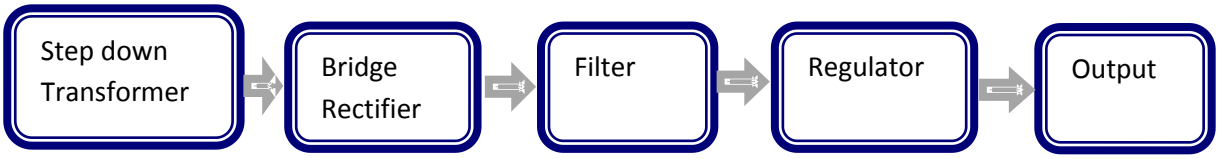
Good software design in embedded systems stems from a good understanding of the hardware behind it. All embedded systems need a microprocessor, and the kinds of microprocessors used in them are quite varied. A list of some of the common microprocessors families are: ARM family, The Zilog Z8 family, Intel 8051/X86 family, Motorola 68K family

and the power PC family. For processing of information and execution of programs, embedded system incorporates microprocessor or micro- controller. In an embedded system the microprocessor is a part of final product and is not available for reprogramming to the end user. An embedded system also needs memory for two purposes, to store its program and to store its data. Unlike normal desktops in which data and programs are stored at the same place, embedded systems store data and programs in different memories. This is simply because the embedded system does not have a hard drive and the program must be stored in memory even when the power is turned off. This type of memory is called ROM. Embedded applications commonly employ a special type of ROM that can be programmed or reprogrammed with the help of special devices.

2.3BLOCK DIAGRAM:



POWER SUPPLY BLOCKDIAGRAM:



2.4 OTHER COMMON PARTS FOUND ON MANY EMBEDDED SYSTEMS

- UART& RS232
- PLD
- ASIC's& FPGA's
- Watch dog timer etc.

2.5 DESIGN PROCESS

Embedded system design is a quantitative job. The pillars of the system design methodology are the separation between function and architecture, is an essential step from conception to implementation. In recent past, the search and industrial community has paid significant attention to the topic of hardware-software (HW/SW) codesign and has tackled the problem of coordinating the design of the parts to be implemented as software and the parts to be implemented as hardware avoiding the HW/SW integration problem marred the electronics system industry so long. In any large scale embedded systems design methodology, concurrency must be considered as a first class citizen at all levels of abstraction and in both hardware and software. Formal models & transformations in system design are used so that verification and synthesis can be applied to advantage in the design methodology. Simulation tools are used for exploring the design space for validating the functional and timing behaviors of embedded systems. Hardware can be simulated at different levels such as electrical circuits, logic gates, RTL e.t.c. using VHDL description. In some environments software development tools can be coupled with hardware simulators, while in others the software is executed on the simulated hardware. The later approach is feasible only for small parts of embedded systems. Design of an embedded system using Intel's 80C188EB chip is shown in the figure. Inorder to reduce complexity, the design process is divided in four major steps: specification, system synthesis, implementation synthesis and performance evaluation of the prototype.

2.5.1 SPECIFICATION

During this part of the design process, the informal requirements of the analysis are transformed to formal specification using SDL.

2.5.2 SYSTEM-SYNTHESIS

For performing an automatic HW/SW partitioning, the system synthesis step translates the SDL specification to an internal system model switch contains problem graph& architecture graph. After system synthesis, the resulting system model is translated back to SDL.

2.5.3 IMPLEMENTATION-SYNTHESIS

SDL specification is then translated into conventional implementation languages such as VHDL for hardware modules and C for software parts of the system.

2.5.4 PROTOTYPING

On a prototyping platform, the implementation of the system under development is executed with the software parts running on multiprocessor unit and the hardware part running on a FPGA board known as phoenix, prototype hardware for Embedded Network Interconnect Accelerators.

2.5.5 APPLICATIONS

Embedded systems are finding their way into robotic toys and electronic pets, intelligent cars and remote controllable home appliances. All the major toy makers across the world have been coming out with advanced interactive toys that can become our friends for life. 'Furby' and 'AIBO' are good examples at this kind. Furbies have a distinct life cycle just like human beings, starting from being a baby and growing to an adult one. In AIBO first two letters stands for Artificial Intelligence. Next two letters represents robot. The AIBO is robotic dog. Embedded systems in cars also known as Telematic Systems are used to provide navigational security communication & entertinment services using GPS, satellite. Home appliances are going the embedded way. LG electronics digital DIOS refrigerator can be used for surfing the net, checking e-mail, making video phone calls and watching TV.IBM is developing an air conditioner that we can control over the net. Embedded systems cover such a broad range of products that generalization is difficult. Here are some broad categories.

- **Aerospace and defence electronics**: Fire control, radar, robotics/sensors, sonar.
- **Automotive**: Autobody electronics, auto power train, auto safety, car information systems.
- **Broadcast & entertainment**: Analog and digital sound products, camaras, DVDs, Set top boxes, virtual reality systems, graphic products.
- **Consumer/internet appliances**: Business handheld computers, business network computers/terminals, electronic books, internet smart handheld devices, PDAs.
- **Data communications:** Analog modems, ATM switches, cable modems, XDSL modems, Ethernet switches, concentrators.
- **Digital imaging**: Copiers, digital still cameras, Fax machines, printers, scanners.
- **Industrial measurement and control:** Hydro electric utility research & management traffic management systems, train marine vessel management systems.
- **Medical electronics:** Diagnostic devices, real time medical imaging systems, surgical devices, critical care systems.
- **Server I/O:** Embedded servers, enterprise PC servers, PCI LAN/NIC controllers, RAID devices, SCSI devices.
- **Telecommunications**: ATM communication products, base stations, networking switches, SONET/SDH cross connect, multiplexer.
- **Mobile data infrastructures**: Mobile data terminals, pagers, VSATs, Wireless LANs, Wireless phones.

CHAPTER 3
ARM PROCESSOR

3.1 INTRODUCTION TO ARM

The ARM7 family includes the ARM7TDMI, ARM7TDMI-S, ARM720T, and ARM7EJ-S processors. The ARM7TDMI core is the industry's most widely used 32-bit embedded RISC microprocessor solution. Optimized for cost and power-sensitive applications, the ARM7TDMI solution provides the low power consumption, small size, and high performance needed in portable, embedded applications.

The ARM7EJ-S processor is a synthesizable core that provides all the benefits of the ARM7TDMI low power consumption, small size, and the thumb instruction set while also incorporating ARM's latest DSP extensions and enabling acceleration of java-based applications. Compatible with the ARM9™, ARM9E™, and ARM10™ families, and Strong-Arm® architecture software written for the ARM7TDMI processor is 100% binary-compatible with other members of the ARM7 family and forwards-compatible with the ARM9, ARM9E, and ARM10 families, as well as products in Intel's Strong ARM and x scale architectures. This gives designers a choice of software-compatible processors with strong price-performance points. Support for the ARM architecture today includes:

- Operating systems such as Windows CE, Linux, palm and SYMBIAN OS.
- More than 40 real-time operating systems, including qnx, Wind River's vxworks and mentor graphics' vrtx.
- Co simulation tools from leading eda vendors
- A variety of software development tools.

3.2 ARM7 TDMI

Figure 2.2 shows the ARM7TDMI Core Diagram. The ARM7TDMI core is based on the Von- Neumann architecture with a 32-bit data bus that carries both instructions and data. Load, store, and swap instructions can access data from memory. Data can be 8-bit, 16-bit, and 32-bit.

Instruction pipeline

The ARM7TDMI core uses a three-stage pipeline to increase the flow of instructions to the processor. This allows multiple simultaneous operations to take place and continuous operation of the processing and memory systems. The instructions are executed in three stages: fetch, decode and execute.

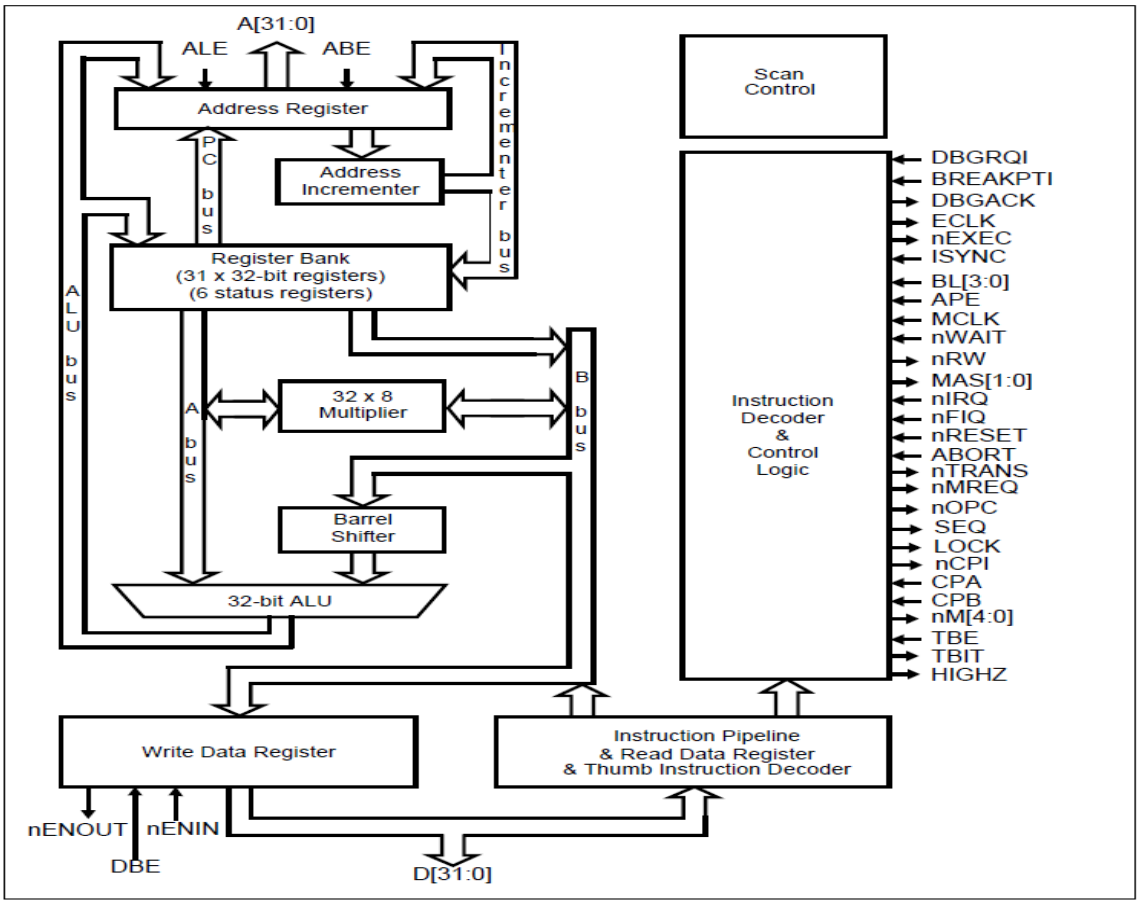


Fig 3.1: ARM7TDMI Core Diagram.

Memory interface

The ARM7TDMI memory interface is designed to allow optimum performance potential and minimize memory usage. Speed critical control signals are pipelined to allow system control functions to exploit the fast-burst access modes supported by many memory technologies. The ARM7TDMI has four basic types of memory cycle: Internal, Non sequential, Sequential, Coprocessor registers transfer. There is also the option to use either a single bidirectional data bus or two separate unidirectional data input and output buses.

Memory formats

The ARM7TDMI can be configured to treat stored words in either big-endian or little-endian format.

Performance, code density and operating states

The ARM7TDMI core supports two operating states and instruction set

- ARM state for 32-bit, word-aligned instructions
- Thumb state for 16-bit, half word-aligned instructions.

The ARM instruction set allows a program to achieve maximum performance with the minimum number of instructions. The simpler thumb instruction set offers much increased code density reducing memory requirement. Code can switch between the ARM and thumb instruction sets on any procedure call.

Operating modes

The ARM7TDMI core has seven modes of operation:

- User mode is the usual program execution state
- allow very fast interrupt processing and to preserve values across
- interrupt calls
- System mode is a privileged user mode for the operating system
- Undefined mode is entered when an undefined instruction is executed.

Coprocessors

Up to 16 coprocessors can be connected to an ARM7TDMI system.

Debug features

Internal state of the ARM core can be examined using a JTAG interface to allow the insertion of instructions into core pipeline and avoid using external data bus. ARM7TDMI core includes an internal functional unit known as the Embedded ICE logic.

ARM7TDMI processor core

The ARM7TDMI processor core implements the ARMv4T Instruction Set Architecture (ISA).This is a superset of the ARMv4 ISA which adds support for the 16-bit Thumb instruction

set. Software using the Thumb instruction set is compatible with all members of the ARM Thumb family, including ARM9, ARM9E, and ARM10 families.

Registers

The ARM7TDMI core consists of a 32-bit data path and associated control logic. This data path contains 31 general-purpose 32-bit registers, 7 dedicated 32-bit registers coupled to a barrel-shifter, Arithmetic Logic Unit, and multiplier.

Modes and exceptions

The ARM7TDMI supports seven modes of operation:

- User mode
- Fast Interrupt (FIQ)
- Interrupt (IRQ)
- Supervisor mode
- Abort mode
- Undefined mode and System mode.

All modes other than User are privileged modes. These are used to service hardware interrupts, exceptions, and software interrupts. Each privileged mode has an associated Saved Program Status Register (SPSR). This register is use to save the state of the Current Program Status Register (CPSR) of the task immediately before the exception occurs. System mode does not have any banked registers. It uses the User mode registers. System mode runs tasks that require a privileged processor mode and allows them to invoke all classes of exception.

Processor states

The ARM7TDMI processor can be in one of two states:

- **ARM state**

  In ARM state, 16 general registers and one or two status registers are accessible at any one time. The ARM state register set contains 16 directly accessible registers: R0 to R15. All of these except R15 are general-purpose, and may be used to hold either data or address values. The registers available to the programmer in each mode, in ARM state, are illustrated in Figure 3.3.3.1 Register Organization in ARM state.

- **THUMB state**

  The THUMB state register set is a subset of the ARM state set. The programmer has direct access to eight general registers, R0-R7, as well as the Program Counter (PC), a stack pointer register (SP), a link register (LR), and the CPSR. There are banked Stack Pointers, Link Registers and Saved Process Status Registers (SPSRs) for each privileged mode. The registers available to the programmer in each mode, in THUMB state, are illustrated in Figure 2.3.3.2 Register organization in THUMB state.

Exceptions

The ARM7TDMI supports seven types of exception:

- FIQ – fast interrupt
- IRQ – normal interrupt
- Data abort
- Pre fetch abort
- Undefined instruction
- Reset.

All exceptions have banked registers for R14 and R13. After an exception, R14 holds the return address for exception processing. This address is used both to return after the exception is processed and to address the instruction that caused the exception. R13 is banked across exception modes to provide each exception handler with a private stack pointer. The fast interrupt mode also banks registers 8 to 12 so that interrupt processing can begin without the need to save or restore these registers.

Status registers

All other processor states are held in status registers. The current operating processor status in the CPSR. The CPSR holds:

- Four ALU flags (Negative, Zero, Carry, and Overflow)
- An interrupt disable bit for each of the FIQ and IRQ interrupts
- A bit to indicate ARM or Thumb execution state
- Five bits to encode the current processor mode.

Conditional execution

All ARM instructions are conditionally executed and can optionally update the four condition code flags (Negative, Zero, Carry, and Overflow) according to their result. Fifteen conditions are implemented.

Classes of instructions

The ARM and Thumb instruction sets can be divided into four broad classes of instruction:

- Data processing instructions
- Load and store instructions
- Branch instructions

Data processing instructions

The data processing instructions operate on data held in general-purpose registers of the two source operands, one is always a register.

The other has two basic forms:

- An immediate value
- A register value optionally shifted.

If the operand is a shifted register the shift amount can have an immediate value or the value of another register. Four types of shift can be specified. Most data processing instructions can perform a shift followed by a logical or arithmetic operation.

Multiply instructions come in two classes:

- Normal, 32-bit result
- Long, 64-bit result variants.

Both types of multiply instruction can optionally perform an accumulate operation.

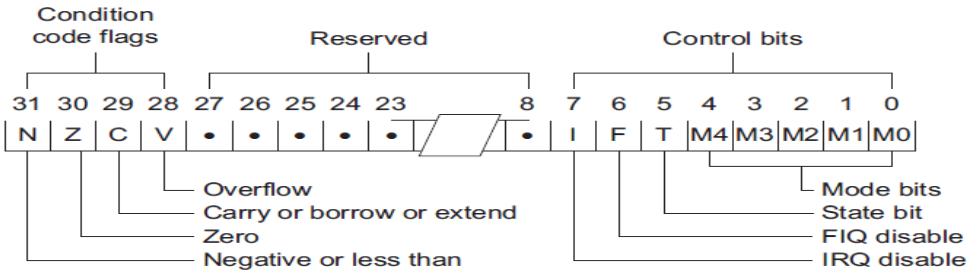
Load and store instructions

Single or multiple registers can be loaded and stored at one time. Load and store single register instructions can transfer a 32-bit word, a 16-bit half word, or an 8-bit byte between memory and a register. Byte and half word loads can be automatically zero extended or sign extended as they are loaded. Load and store instructions have three primary addressing modes:

- offset
- pre-indexed
- post-indexed.

The address is formed by adding or subtracting an immediate or register- based offset to or from a base register. Register-based offsets can also be scaled with shift operations. Pre-indexed and post-indexed addressing modes update the base register with the result of the offset calculation. As the PC is a general-purpose register, a 32-bit value can be loaded directly into the PC to perform a jump to any address in the 4GB memory space. Load and store multiple instructions perform a block transfer of any number of the general purpose registers to or from memory. Four addressing modes are provided:

- Pre-increment addressing
- Post-increment addressing
- Pre-decrement addressing
- Post-decrement addressing



.Fig 3.2:

Load and store instructions.

Branch instructions

As well as allowing any data processing or load instruction to change control flow (by modifying the PC) a standard branch instruction is provided with 24-bit signed offset, allowing forward and backward branches of up to 32MB. Branch with Link (BL) allows efficient subroutine calls, and preserves the address of the instruction after the branch in R14 (the Link Register or LR)

Advantages

- Simple hardware
- Small die size

- Low power consumption
- Simple decoding
- Higher performance
- Easy to implement an effective pipelined structure.

Disadvantages

- Performance depends on compiler
- Poor code density
- RISC has a fixed size of instruction format

Applications

Using the ARMv7 architecture, ARM can strengthen its position as a low-power/performance leader while conquering new markets to carry its cores up in high performance and down in the low-cost high-volume domain of the microcontroller ARM designs the technology that lies at the heart of advanced digital products, from wireless, networking and consumer entertainment solutions to imaging, automotive, security and storage devices. ARM's comprehensive product offering includes 16/32-bit RISC microprocessors, data engines, 3D processors, digital libraries, embedded memories, peripherals, software and development tools, as well as analog functions and high-speed connectivity products.

3.3 LPC2148 MICROCONTROLLER

LPC2148 microcontroller board based on a 16-bit/32-bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combine microcontrollers with embedded high-speed flash memory ranging from 32 KB to 512 KB. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30% with minimal performance penalty. The meaning of LPC is Low Power Low Cost microcontroller. This is 32 bit microcontroller manufactured by Philips semiconductors (NXP). Due to their tiny size and low power consumption, LPC2148 is ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale.

3.3.1 FEATURES OF LPC2148 MICROCONTROLLER

- 16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
- 8 KB to 40 KB of on-chip static RAM and 32 KB to 512 KB of on-chip flash memory; 128-bit wide interface/accelerator enables high-speed 60 MHz operation.
- USB 2.0 Full-speed compliant device controller with 2 KB of endpoint RAM. In addition, the LPC2148 provides 8 KB of on-chip RAM accessible to USB by DMA.
- One or two (LPC2141/42 Vs, LPC2144/46/48) 10-bit ADCs provide a total of 6/14 analog inputs, with conversion times as low as 2.44 ms per channel.
- Single 10-bit DAC provides variable analog output (LPC2148 only)
- Two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog.
- Low power Real-Time Clock (RTC) with independent power and 32 kHz clock input

3.3.2 LPC2148 MICROCONTROLLER ARCHITECTURE

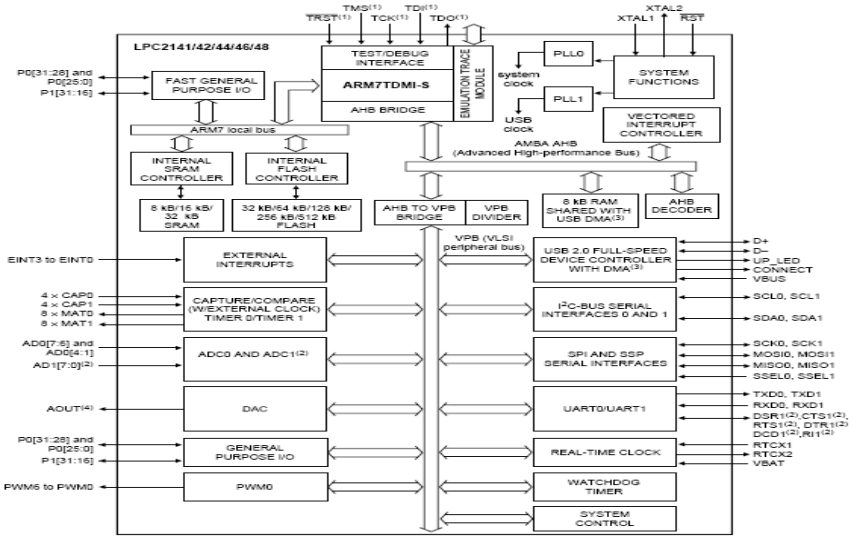


Fig 3.3: LPC2148 Microcontroller Architecture.

3.3.3 PIN DIAGRAM

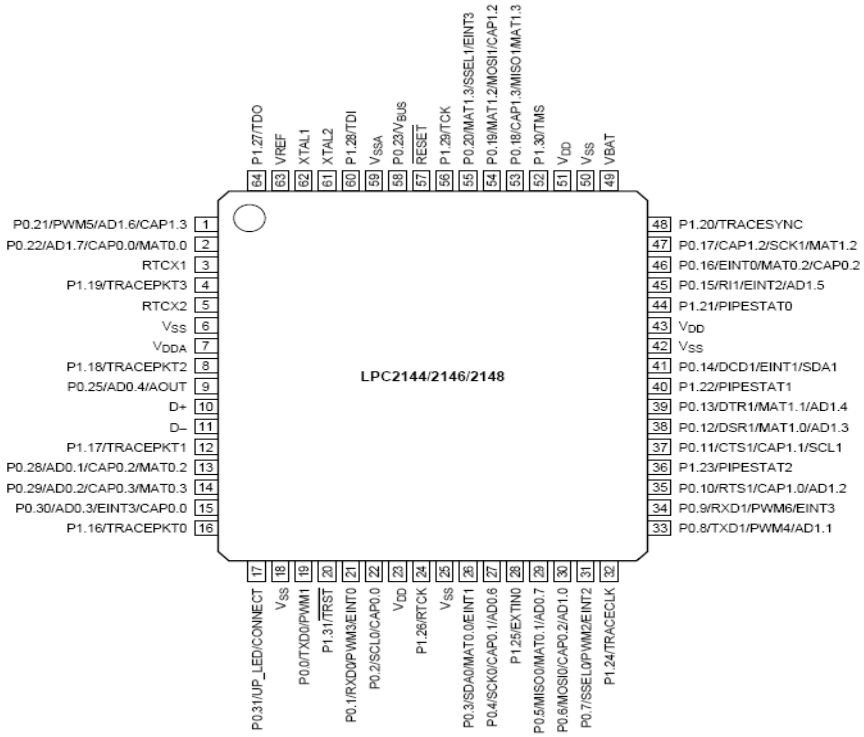


Fig 3.4: LPC2148 Microcontroller Pin Diagram.

3.3.4 ARCHITECTURAL OVERVIEW

Figure 3.3 shows the LPC2148 Microcontroller Architecture. The ARM7TDMI-S is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of micro programmed Complex Instruction Set Computers (CISC). This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective processor core. Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. The ARM7TDMI-S processor also employs a unique architectural strategy known as Thumb, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue.

On-chip flash program memory

The LPC2141/42/44/46/48 incorporates a 32 KB, 64 KB, 128 KB, 256 KB and 512 KB flash memory system respectively. 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 Due to the architectural solution chosen for an on-chip boot loader, flash memory available for user's code on LPC2141/42/44/46/48 is 32 KB, 64 KB, 128 KB, 256 KB and 500 KB respectively.

On-chip static RAM

On-chip static RAM may be used for code and/or data storage. The SRAM may be accessed as 8-bit, 16-bit, and 32-bit. The LPC2141, LPC2142/44 and LPC2146/48 provide 8 KB, 16 KB and 32 KB of static RAM respectively. In case of LPC2146/48 only, an 8 KB SRAM block intended to be utilized mainly by the USB can also be used as a general purpose RAM for data storage and code storage and execution.

Memory map

The LPC2141/42/44/46/48 memory map incorporates several distinct regions, as shown in Figure 3.5 Memory map. In addition, the CPU interrupt vectors may be remapped to allow them to reside in either flash memory (the default) or on-chip static RAM.

Pin connect block

The pin connect block allows selected pins of the microcontroller 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.

Fast general purpose parallel I/O (GPIO)

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.LPC2141/42/44/46/48 introduces accelerated GPIO functions over prior LPC2000 devices:

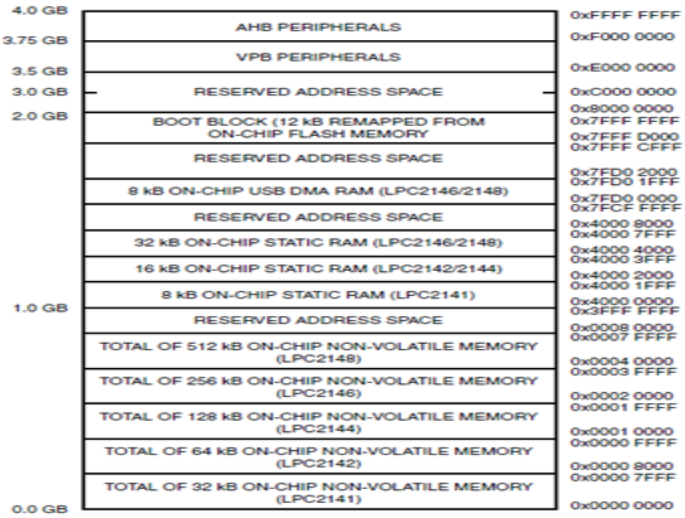


Fig 3.5: Memory map.

Interrupt controller

The Vectored Interrupt Controller (VIC) accepts all of the interrupt request inputs and categorizes them as Fast Interrupt Request (FIQ), vectored Interrupt Request (IRQ), and non-vectored IRQ as defined by programmable settings. If more than one request is assigned to the FIQ class, the FIQ service routine will read a word from the. If any of the vectored IRQs are pending, the VIC provides the address of the highest-priority requesting IRQs service routine, otherwise it provides the address of a default routine that is shared by all the non-vectored IRQs. The default routine can read another VIC register to see what IRQs are active.

10-bit ADC

The LPC2141/42 contains one and the LPC2144/46/48 contains two analog to digital converters. These converters are single 10-bit successive approximation analog to digital converters.

Features

- 10 bit successive approximation analog to digital converter.
- Measurement range of 0 V to VREF (2.0 V ≤ VREF ≤ VDDA).
- Every analog input has a dedicated result register to reduce interrupt overhead.
- Burst conversion mode for single or multiple inputs.

10-bit DAC

The DAC enables the LPC2141/42/44/46/48 to generate a variable analog output. The maximum DAC output voltage is the VREF voltage.

Features

- 10-bit DAC.
- Buffered output.
- Power-down mode available.

USB 2.0 device controller

The USB is a 4-wire serial bus that supports communication between a host and a number (127 max) of peripherals. The host controller allocates the USB bandwidth to attached devices through a token based protocol. The bus supports hot plugging, unplugging, and dynamic configuration of the devices. All transactions are initiated by the host controller

Features

- Fully compliant with USB 2.0 Full-speed specification.
- Supports 32 physical (16 logical) endpoints.
- Scalable realization of endpoints at run time.
- RAM message buffer size based on endpoint realization and maximum packet size.

UARTs

The LPC2141/42/44/46/48 each contains two UARTs. Compared to previous LPC2000 microcontrollers, UARTs in LPC2141/42/44/46/48 introduce a fractional baud rate generator for both UARTs, enabling these microcontrollers to achieve standard baud rate such as 115200 with any crystal frequency above 2 MHz In addition, auto-CTS/RTS flow-control functions are fully implemented in hardware (UART1 in LPC2144/46/48 only).

Features

- 16 byte Receive and Transmit FIFO.
- Register locations conform to ‘550 industry standard.
- Receiver FIFO triggers points at 1, 4, 8, and 14 bytes
- Transmission FIFO control enables implementation of software (XON/XOFF) Flow control on both UARTs.

I2C-bus serial I/O controller

The LPC2141/42/44/46/48 each contains two 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 (e.g., an LCD driver or a transmitter with the capability to both receive and send information (such as memory)).

Features

- Compliant with standard I2C-bus interface.
- Bidirectional data transfer between masters and slaves.
- Multi-master bus (no central master).

SPI serial I/O controller

The LPC2141/42/44/46/48 each contains one SPI controller. The SPI is a full duplex serial interface, designed to handle multiple masters and slaves connected to a given bus. During a data transfer the master always sends a byte of data to the slave, and the slave always sends a byte of data to the master.

Features

- Compliant with Serial Peripheral Interface (SPI) specification.
- Synchronous, Serial, Full Duplex, Communication.
- Maximum data bit rate of one eighth of the input clock rate.

SSP serial I/O controller

The LPC2141/42/44/46/48 each contains one SSP. The SSP controller is capable of operation on a SPI, 4-wire SSI, or Micro wire bus. It can interact with multiple masters and slaves on the bus. However, 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 data frames of 4 bits to 16 bits of data flowing from the master to the slave and from the slave to the master.

Features

- Synchronous serial communication.
- Master or slave operation.

General purpose timers/external event counters

The Timer/Counter is designed to count cycles of the peripheral clock (PCLK) or an externally supplied clock and optionally generate interrupts or perform other actions at specified timer values, based on four match registers

Watchdog timer

The purpose of the watchdog is to reset the microcontroller 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.

Features

- Internally resets chip if not periodically reloaded.
- Debug mode.
- Programmable 32-bit timer with internal pre-scalar.

Real-time clock

The RTC is designed to provide a set of counters to measure time when normal or idle operating mode is selected. The RTC has been designed to use little power, making it suitable for battery powered systems where the CPU is not running continuously.

Features

- Measures the passage of time to maintain a calendar and clock.
- Provides Seconds, Minutes, Hours, Day of Month, Month, Year, Day of Week, and Day of Year.

CHAPTER 4

SERIAL COMMUNICATION

4.1 INTRODUCTION

Computers transfer data in two ways: parallel and serial. In parallel data transfers, often 8 or more lines (wire conductors) are used to transfer data to a device that is only a few feet away. Examples of parallel transfers are printers and hard disk; each uses cables with many wire strips. Although in such cases a lot of data can be transferred in a short amount of time by using many wires in parallel, the distance cannot be great. To transfer to a device located at many meters away, the serial method is used. In serial communication, the data is sent one bit at a time, in contrast to parallel communication, in which the data is sent a byte or more at a time.

For serial data communication to work the byte of data must be converted to serial bits using a parallel-in-serial-out shift register; then it can be transmitted over a single data line. This also means that at the receiving end there must be a serial-in-parallel-out shift register to receive the serial data and pack them into a byte. Of course, if data is to be transferred on the telephone line, it must be converted from 0s and 1s to audio tones, which are sinusoidal-shaped signals. This conversion is performed by a peripheral device called a modem, which stands for "modulator/demodulator".

4.2 RS232 STANDARDS

To allow compatibility among data communication equipment made by various manufacturers, an interfacing standard called RS232 was set by the Electronics Industries Association (EIA) in 1960. Today, RS232 is the most widely used serial I/O interfacing standard. However, since the standard was set long before the advent of TTL logic family, its input and output voltage levels are not TTL compatible. In RS232, a 1 is represented by -3 to -25V, while a 0 bit is +3 to +25V, making -3 to +3 undefined. For this reason, to connect any RS232 to a microcontroller system we must use voltage converters such as MAX232 to convert the TTL logic levels to the RS232 voltage levels, and vice versa. MAX232 IC chips are commonly referred to as line drivers.

4.2.1 DB-9 CONNECTOR

Since not all the pins are used in PC cables, IBM introduced the DB-9 version of the serial I/O standard, which uses 9 pins only, as shown in the following table:

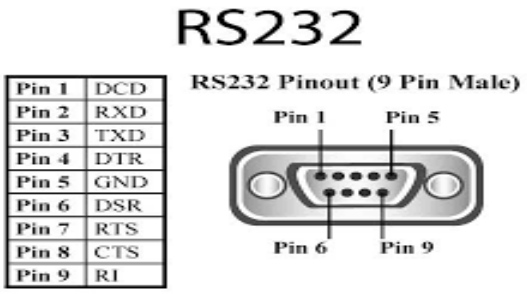


Fig 4.1: RS232 pin configuration

Table 4.1: Pin Description of DB-9 Connector.

PIN	DESCRIPTION
1	Data carrier detect (DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (DSR)
7	Request to send (RTS)
8	Clear to send (CTS)
9	Ring indicator (RI)

The D-subminiature or D-sub is a common type of electrical connector used particularly in computers. At the time of introduction they were some of the smaller connectors used on computer systems. A D-sub contains two or more parallel rows of pins or sockets usually surrounded by a D-shaped metal shield that provides mechanical support, some screening against electromagnetic interference, and ensures correct orientation.

4.3 MAX232

The MAX232 is an integrated circuit that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The drivers provide RS-232 voltage level outputs (approx. ± 7.5 V) from a single + 5 V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0 V to + 5 V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case.

4.3.1 PIN DIAGRAM OF MAX232

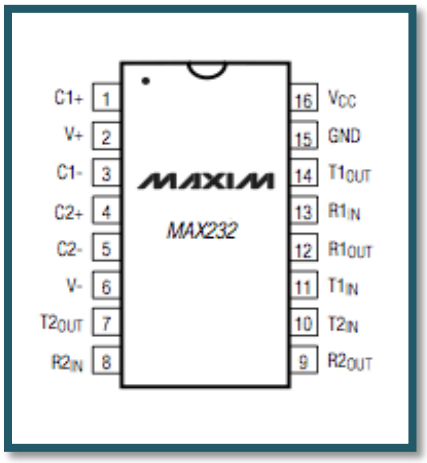


Fig 4.2: Pin Diagram of MAX232.

RS232 Interfaced to MAX 232

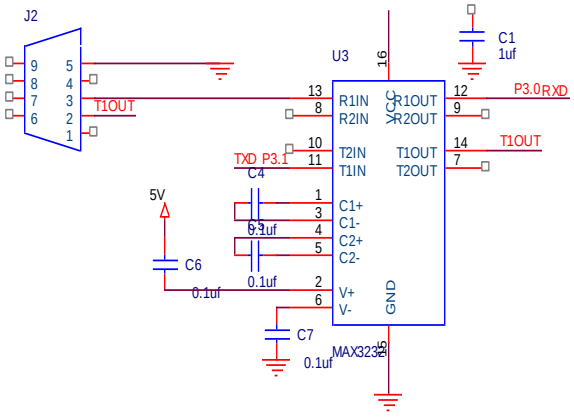


Fig 4.3: RS232 Interfaced to MAX232.

RS232 is 9 pin db connector, only three pins of this are used ie 2,3,5 the transmit pin of RS232 is connected to Rx pin of MAX232.

4.3.2 VOLTAGE LEVELS

It is helpful to understand what occurs to the voltage levels. When a MAX232 IC receives a TTL level to convert, it changes a TTL Logic 0 to between +3 and +15V, and changes TTL Logic 1 to between -3 to -15V, and vice versa for converting from RS232 to TTL. This can be confusing when you realize that the RS232 Data Transmission voltages at a certain logic state are opposite from the RS232 Control Line voltages at the same logic state. To clarify the matter, see the table below.

Table 4.2: TTL Logic Levels

RS232 Line Type & Logic Level	RS232 Voltage	TTL Voltage to/from MAX232
Data Transmission (Rx/Tx) Logic 0	+3V to +15V	0V
Data Transmission (Rx/Tx) Logic 1	-3V to -15V	5V
Control Signals (RTS/CTS/DTR/DSR) Logic 0	-3V to -15V	5V
Control Signals (RTS/CTS/DTR/DSR) Logic 1	+3V to +15V	0V

4.4 SERIAL COMMUNICATION IN LPC2148

Uart uses TxD(Transmit) Pin for sending Data and RxD(Receive) Pin to get data. UART sends & receives data in form of chunks or packets. These chunks or packets are also referred to as 'transmission characters'. The structure of a UART data packet is as shown below:

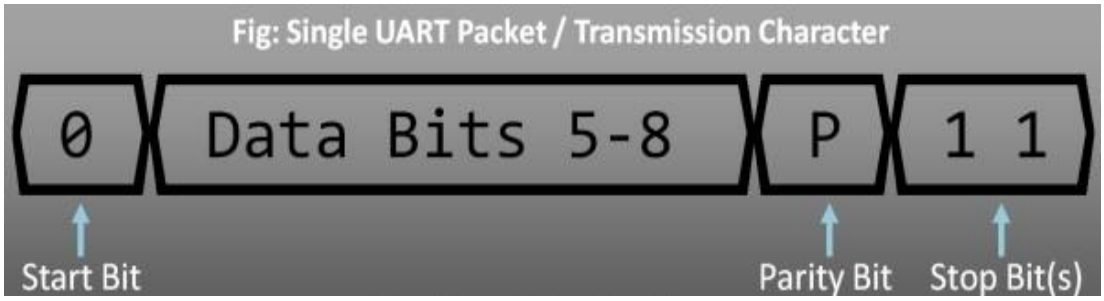


Fig 4.4: Single UART packet/Transmission Character

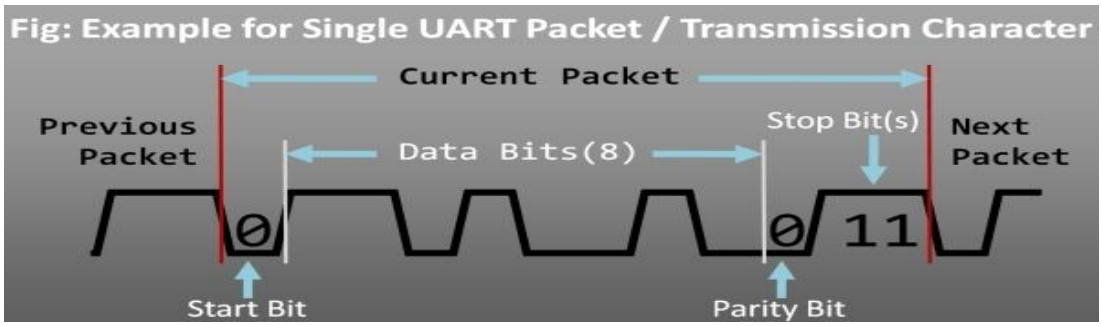


Fig 4.5: Example for single UART packet

LPC214x has 2 UART blocks which are UART0 and UART1. For UART0 the TxD pin is P0.0 and RxD pin is P0.1 and similarly for UART 1 the TxD pin is P0.8 and RxD pin is P0.9 as shown in the table below:

Pins:	TxD	RxD
UART0	P0.0	P0.1
UART1	P0.8	P0.9

Both UART0 & UART1 blocks internally have a 16-byte FIFO (First In First Out) structure to hold the Rx and Tx data. Each byte in this FIFO represents a character which was sent or received in order. Both blocks also contain 2 registers each, for data access and assembly.

Tx has THR(Transmit Holding Register) and TSR(Transmit Shift Register)

When we write Data to be sent into THR it is then transferred to TSR which assembles the data to be transmitted via Tx Pin.

Rx has RSR(Receive Shift Register) and RBR(Receive Buffer Register)

When a valid data is Received at Rx Pin it is first assembled in RSR and then passed in to Rx FIFO which can be then accessed via RBR.

Data Related Registers

1) U0RBR – Receiver Buffer Register (READ ONLY!)

This register contains the top most byte(8-bit data chunk) in the Rx FIFO i.e the oldest received data in FIFO. To properly read data from U0RBR , the DLAB(Divisor Latch Access) bit in U0LCR register must be first set to 0. Also , as per the user manual “The right approach for

fetching the valid pair of received byte and its status bits is first to read the content of the U0LSR register, and then to read a byte from the U0RBR."

2) U0THR – Transmit Holding Register (WRITE ONLY!)

U0THR contains the top most byte in Tx FIFO and in this case it's the newest (latest) transmitted data. As in the case with U0RBR, we must set DLAB=0 to access U0THR for write operation.

Baud Rate Setup related registers

1) U0DLL and U0DLM – Divisor Latch registers

Both of them hold 8-bit values. These register together form a 16-bit divisor value which is used in baud rate generation which we will see in later section. U0DLM holds the upper 8-bits and U0DLL holds the lower 8-bits and the formation is "[U0DLM:U0DLL]". Since these form a divisor value and division by zero is invalid, the starting value for U0DLL is 0×01 (and not 0×00) i.e the starting value in combined formation is "[0x00:0x01]" i.e 0×0001.

2) U0FDR – Fractional Divider Register

This register is used to set the prescale value for baud rate generation. The input clock is the peripheral clock and output is the desired clock defined by this register. This register actually holds to different 4-bit values (a divisor and a multiplier) for prescaling which are:

1. **Bit [3 to 0] – DIVADDVAL** This is the prescale divisor value. If this value if 0 then fractional baud rate generator wont have any effect on Uart Baud rate.
2. **Bit [7 to 4] – MULVAL** This is prescale multiplier value. Even if fractional baud rate generator is not used the value in this register must be more than or equal to 1 else UART0 will not operate properly.
3. Other Bits reserved.

Control and Status Registers

1) U0FCR – FIFO Control Register

Used to control Rx/Tx FIFO operations.

1. **Bit 0 – FIFO Enable** 1 to enable both Rx and Tx FIFOs and 0 to disable.
2. **Bit 1 – Rx FIFO Reset** Writing a 1 will clear and reset Rx FIFO.
3. **Bit 2 – Tx FIFO Reset** Writing a 1 will clear and reset Tx FIFO.

4. **Bits [7 to 6]** Used to determine that how many UART0 Rx FIFO characters must be written before an interrupt is activated.
   [00] (i.e trigger level 0) for 1 character.
   [01] (i.e trigger level 1) for 4 characters.
   [10] (i.e trigger level 2) for 8 characters.
   [11] (i.e trigger level 3) for 14 characters.
5. Others bits are reserved.

2) U0LCR – Line Control Register

Used to configure the UART block (i.e the data format used in transmission).

1. **Bit [1 to 0] – Word Length Select:** Used to select the length of an individual data chunk. [00] for 5 bit character length. Similarly [01] , [10] , [11] for 6 , 7 , 8 bit character lengths respectively.
2. **Bit 2 – Stop bit select:** 0 for using 1 stop bit and 1 for using 2 stop bits.
3. **Bit 3 – Parity Enable:** 0 to disabled Partiy generation & checking and 1 to enable it.
4. **Bit [5 to 4] – Parity Select:** [00] to Odd-parity, [01] for Even-parity, [10] for forced "1″ (Mark) parity and [11] for forced "0″ (Space) parity.
5. **Bit 6 – Break Control:** 0 to disable break transmission and 1 to enable it. TxD pin will be forced to logic 0 when this bit is 1!
6. **Bit 7 – Divisior Latch Access bit:** 0 to disable access to divisor latches and 1 to enable access.

3) U0LSR – Line Status Register: used to read the status of Rx and Tx blocks.

1. **Bit 0 – Receiver Data Ready (RDR):** 0 means U0RBR is empty(i.e Rx FIFO is empty) and 1 means U0RBR contains valid data.
2. **Bit 1 – Overrun Error (OE):** 0 means Overrun hasn't occured and 1 means Overrun has occured. Overrun is the condition when RSR (Receive Shift Register)[See note 1] has new character assembled but the RBR FIFO is full and the new assembled character is eventually lost since no data is written into FIFO if its full. (**Note:** *Reading U0LSR clears this bit*)
3. **Bit 2 – Parity Error (PE):** 0 mean no parity error and 1 mean a parity error has occured. When the value of the parity bit in the recieved character is in wrong state then a parity error occurs. (Note: Reading U0LSR clears this bit)

4. **Bit 3 – Framing Error (FE):** 0 means no framing error has occured and 1 means that a framing error has taken place. Framing error occurs when the stop bit of a received character is zero. (**Note:** *Reading U0LSR clears this bit*)
5. **Bit 4 – Break Interrupt:** 0 means no Break Interrupt occures and 1 means that it has occured. A Break Interrupt occurs when the RxD line is pulled low (i.e all 0s) i.e held in spacing state for 1 full character after which Rx Block goes into Idle state. Rx Block gets back to active state when RxD pin is pulled high (i.e all 1s) i.e held in marking state for 1 full character. (**Note:** *Reading U0LSR clears this bit*)
6. **Bit 5 – Transmit Holding Register Empty (THRE):** 0 means U0THR contains valid data and 1 means it's empty.
7. **Bit 6 – Transmitter Empty (TEMT):** 0 means U0THR and/or U0RSR contains valid data and 1 means that both U0THR and U0RSR are empty.
8. **Bit 7 – Error in RX FIFO (RXFE):** 0 means that U0RBR has no Rx Errors or Rx FIFO is disabled (i.e 0th bit in U0FCR is 0) and 1 means that U0RBR has atleast one error. (**Note:** *This bit is cleared only if U0LSR is read and there are no other subsequent errors in Rx FIFO .. else this bit will stay 1*)

4) U0TER – Transmit Enable Register: This register is used to enable UART transmission. When bit-7 (i.e TXEN) is set to 1 Tx block will be enabled and will keep on transmitting data as soon as its ready. If bit-7 is set to 0 then Tx will stop transmission. Other bits are reserved.

Interrupt Related Registers

1) U0IER – Interrupt Enable Register: Set a bit to 0 to disable and 1 to enable the corresponding interrupt.

1. **Bit 0** – RBR Interrupt Enable
2. **Bit 1** - THRE Interrupt Enable
3. **Bit 2** – RX Line Status Interrupt Enable
4. **Bit 3** – ATBOInt Enable
5. **Bit 4** – ATEOInt Enable

Where ATBOInt = Auto Baud Time-Out Interrupt, ATEO = End of Auto Baud Interrupt and rest of the bits are reserved.

2) U0IIR – Interrupt Identification Register: Refer User Manual when in doubt. In some application the usage of this register might get a bit complicated.

This register is organized as follows:

1. **Bit 0 – Interrupt Pending:** 0 means atleast one interrupt is pending, 1 means no interrupts are pending. Note: This bit is ACTIVE LOW!
2. **Bits [3 to 1] – Interrupt Identification:** [011] is for Receive Line Status (RLS), [010] means Receive Data Available (RDA) , 110 is for Character Time-out Indicator(CTI), [001] is for THRE Interrupt.
3. **Bits [7 to 6] – FIFO Enable.**
4. **Bit 8 – ABEOInt:** 1 means Auto Baud Interrupt has successfully ended and 0 otherwise.
5. **Bit 9 – ABTOInt:** 1 means Auto Baud Interrupt has Timed-out.
6. All others bits are reserved.

CHAPTER 5
HARDWARE COMPONENTS

5.1 LCD (Liquid Cristal Display)

Introduction:

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Features:

(1) Interface with either 4-bit or 8-bit microprocessor.

(2) Display data RAM

(3) 80x□8 bits (80 characters).

(4) Character generator ROM

(5). 160 different 5 □□7 dot-matrix character patterns.

(6).□ Character generator RAM

(7) □8 different user programmed 5 □□7 dot-matrix patterns.

(8).Display data RAM and character generator RAM may be Accessed by the microprocessor.

(9) Numerous instructions

(10) .Clear Display, Cursor Home, Display ON/OFF, Cursor ON/OFF, Blink Character, Cursor Shift, Display Shift.

(11). Built-in reset circuit is triggered at power ON.

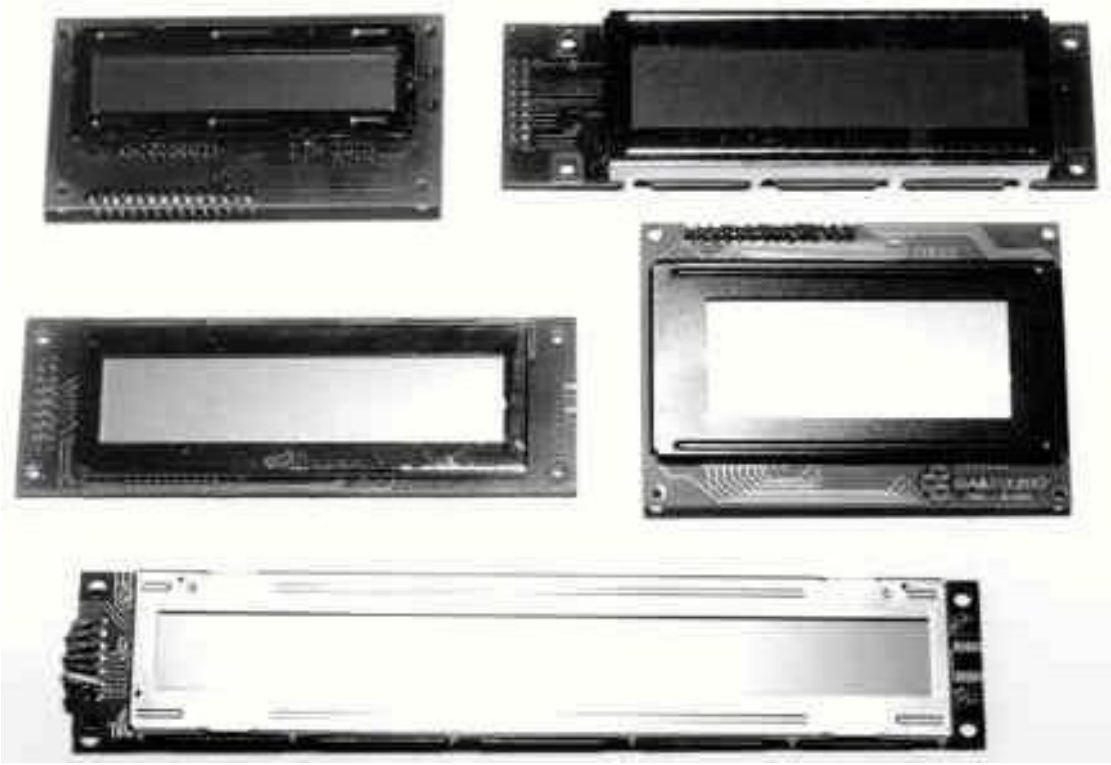
(12). Built-in oscillator.

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Fig : Address locations for a 1x16 line LCD

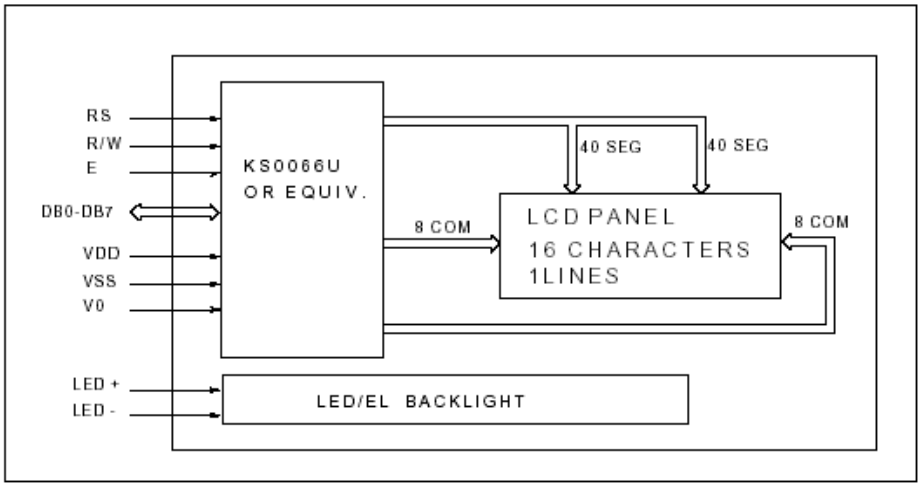
5.2 Shapes and sizes:



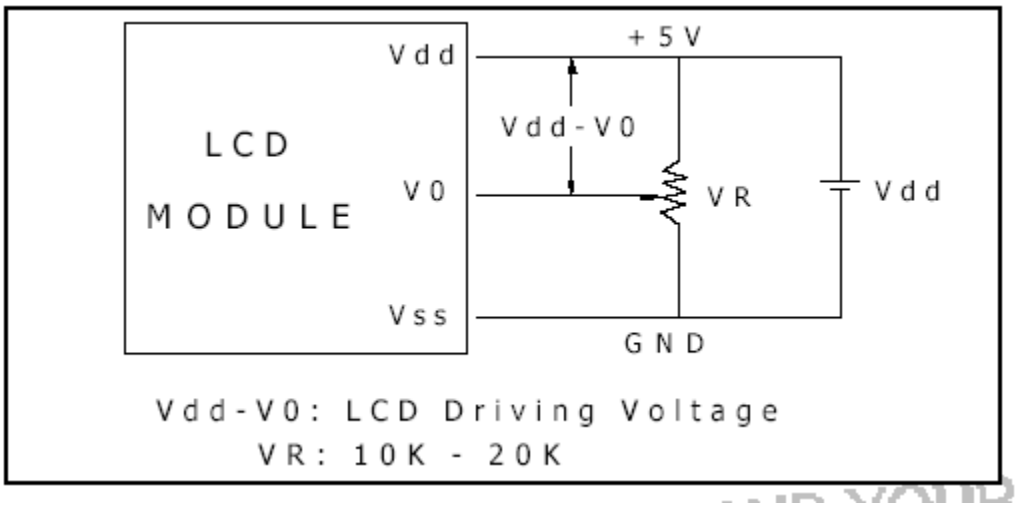
Even limited to character based modules,there is still a wide variety of shapes and sizes available. Line lenghs of 8,16,20,24,32 and 40 charecters are all standard, in one, two and four line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

5.3 Electrical blockdiagram:



5.4 Power supply for lcd driving:



5.5 PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

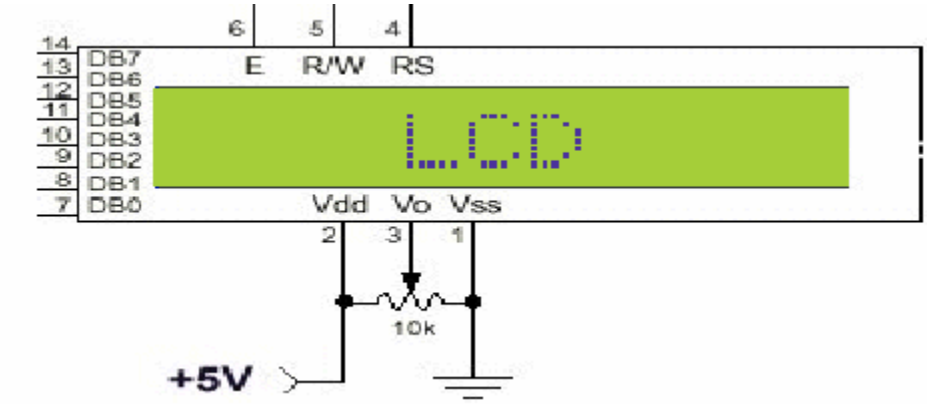


Fig: pin diagram of 1x16 lines lcd

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

CONTROL LINES:

EN:

Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS:

Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which sould be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW:

Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Read data from data lines (if it is reading)on LCD:

1) Set R/W bit to high

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high

5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

The switches must be the type where On = 0, so that when they are turned to the zero position, all four outputs are shorted to the common pin, and in position "F", all four outputs are open circuit.

All the available characters that are built into the module are shown in Table 3. Studying the table, you will see that codes associated with the characters are quoted in binary and hexadecimal, most significant bits ("left-hand" four bits) across the top, and least significant bits ("right-hand" four bits) down the left.

Most of the characters conform to the ASCII standard, although the Japanese and Greek characters (and a few other things) are obvious exceptions. Since these intelligent modules were designed in the "Land of the Rising Sun," it seems only fair that their Katakana phonetic symbols should also be incorporated. The more extensive Kanji character set, which the Japanese share

with the Chinese, consisting of several thousand different characters, is not included!

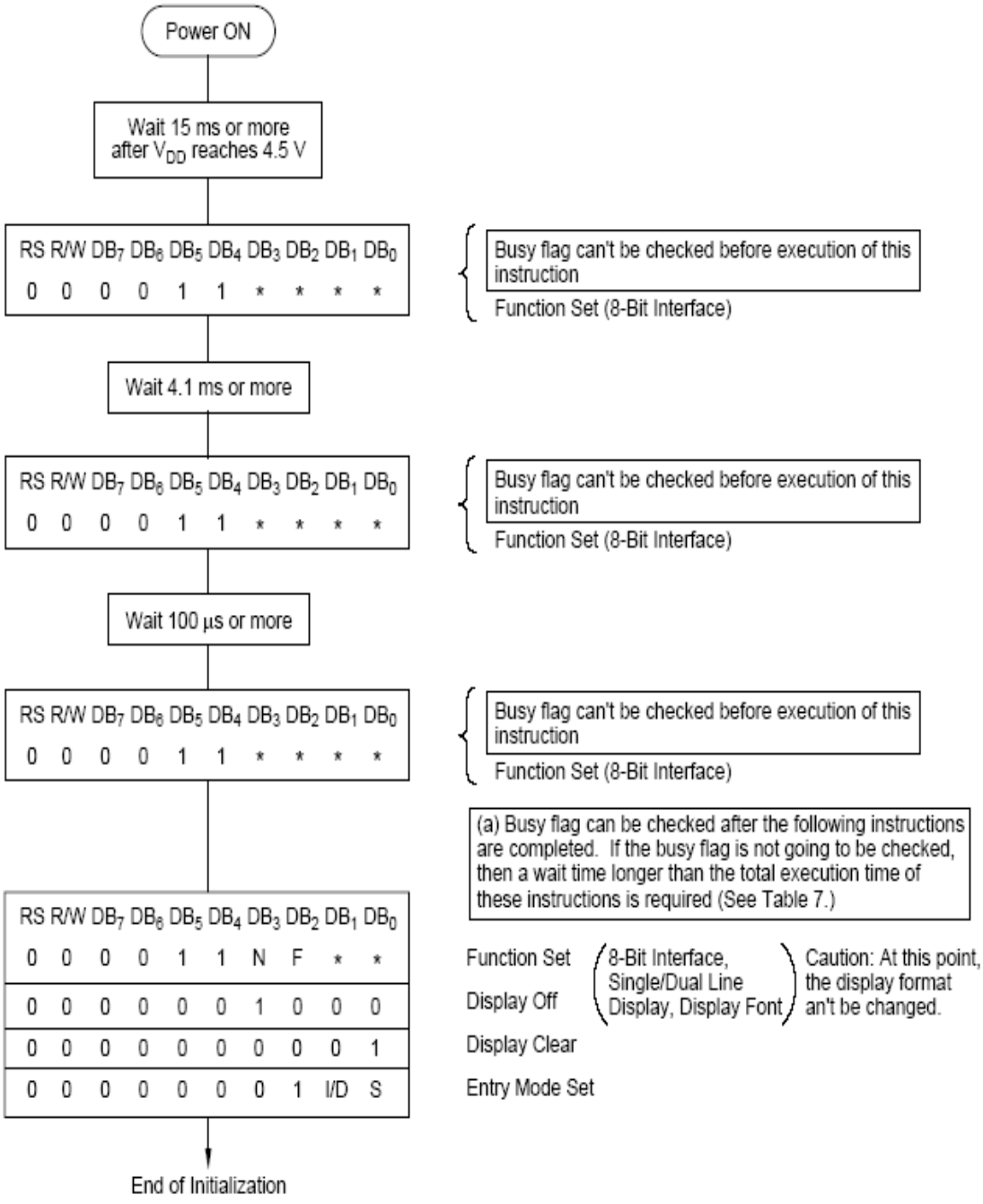
Using the switches, of whatever type, and referring to Table 3, enter a few characters onto the display, both letters and numbers. The RS switch (S10) must be “up” (logic 1) when sending the characters, and switch E (S9) must be pressed for each of them. Thus the operational order is: set RS high, enter character, trigger E, leave RS high, enter another character, trigger E, and so on.

The first 16 codes in Table 3, 00000000 to 00001111, ($00 to $0F) refer to the CGRAM. This is the Character Generator RAM (random access memory), which can be used to hold user-defined graphics characters. This is where these modules really start to show their potential, offering such capabilities as bar graphs, flashing symbols, even animated characters. Before the user-defined characters are set up, these codes will just bring up strange looking symbols.

Codes 00010000 to 00011111 ($10 to $1F) are not used and just display blank characters. ASCII codes “proper” start at 00100000 ($20) and end with 01111111 ($7F). Codes 10000000 to 10011111 ($80 to $9F) are not used, and 10100000 to 11011111 ($A0 to $DF) are the Japanese characters.


Upper 4 bits
Lower 4 bits
0 0000
1 0001
2 0010
3 0011
4 0100
5 0101
6 0110
7 0111
8 1000
9 1001
A 1010
B 1011
C 1100
D 1101
E 1110
F 1111
CG RAM (1)
CG RAM (2)
CG RAM (3)
CG RAM (4)
CG RAM (5)
CG RAM (6)
CG RAM (7)
CG RAM (8)

Initialization by Instructions:



If the power conditions for the normal operation of the internal reset circuit are not satisfied, then executing a series of instructions must initialize LCD unit. The procedure for this initialization process is as above show.

5.6 REGULATED POWER SUPPLY:

Introduction:

Power supply is a supply of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

A power supply may include a power distribution system as well as primary or secondary sources of energy such as

- Conversion of one form of electrical power to another desired form and voltage, typically involving converting AC line voltage to a well-regulated lower-voltage DC for electronic devices. Low voltage, low power DC power supply units are commonly integrated with the devices they supply, such as computers and household electronics.
- Batteries.
- Chemical fuel cells and other forms of energy storage systems.
- Solar power.
- Generators or alternators.

Block Diagram:

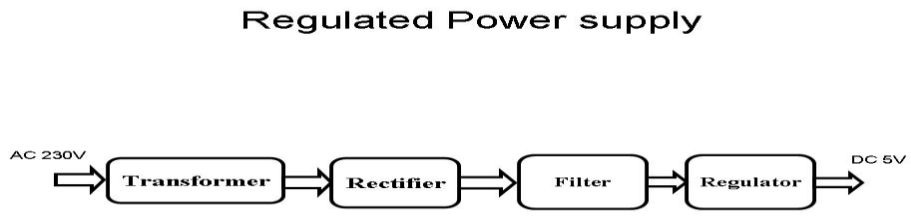


Fig .Regulated Power Supply

The basic circuit diagram of a regulated power supply (DC O/P) with led connected as load is shown in fig:

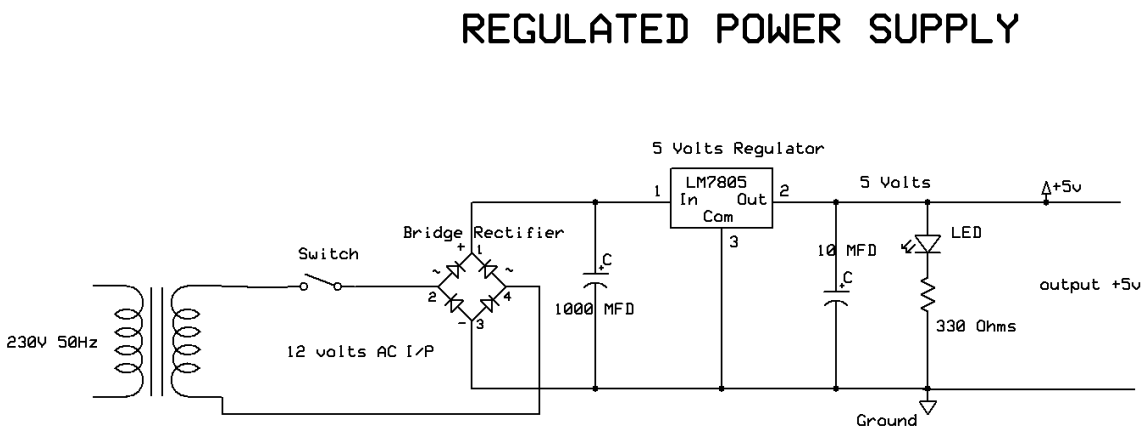


Fig Circuit diagram of Regulated Power Supply with Led connection

The components mainly used in above figure are

- 230V AC MAINS
- TRANSFORMER
- BRIDGE RECTIFIER(DIODES)
- CAPACITOR
- VOLTAGE REGULATOR(IC 7805)
- RESISTOR
- LED(LIGHT EMITTING DIODE)

The detailed explanation of each and every component mentioned above is as follows:

Step 1: Transformation: The process of transforming energy from one device to another is called transformation. For transforming energy we use transformers.

5.7 Transformers:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors without changing its frequency. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

If a load is connected to the secondary, an electric current will flow in the secondary winding and electrical energy will be transferred from the primary circuit through the transformer to the load. This field is made up from lines of force and has the same shape as a bar magnet.

If the current is increased, the lines of force move outwards from the coil. If the current is reduced, the lines of force move inwards.

If another coil is placed adjacent to the first coil then, as the field moves out or in, the moving lines of force will "cut" the turns of the second coil. As it does this, a voltage is induced in the second coil. With the 50 Hz AC mains supply, this will happen 50 times a second. This is called MUTUAL INDUCTION and forms the basis of the transformer.

The input coil is called the PRIMARY WINDING; the output coil is the SECONDARY WINDING. Fig: 3.3.4 shows step-down transformer.

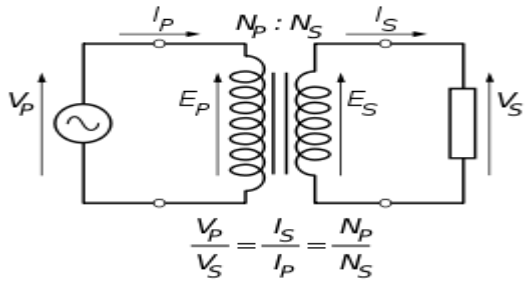


Fig 3.3.3: Step-Down Transformer

The voltage induced in the secondary is determined by the TURNS RATIO.

$$\frac{\text{primary voltage}}{\text{secondary voltage}} = \frac{\text{number of primary turns}}{\text{number of secondary turns}}$$

For example, if the secondary has half the primary turns; the secondary will have half the primary voltage.

Another example is if the primary has 5000 turns and the secondary has 500 turns, then the turn's ratio is 10:1.

If the primary voltage is 240 volts then the secondary voltage will be x 10 smaller = 24 volts. Assuming a perfect transformer, the power provided by the primary must equal the power taken by a load on the secondary. If a 24-watt lamp is connected across a 24 volt secondary, then the primary must supply 24 watts.

To aid magnetic coupling between primary and secondary, the coils are wound on a metal CORE. Since the primary would induce power, called EDDY CURRENTS, into this core, the core is LAMINATED. This means that it is made up from metal sheets insulated from each other. Transformers to work at higher frequencies have an iron dust core or no core at all.

Note that the transformer only works on AC, which has a constantly changing current and moving field. DC has a steady current and therefore a steady field and there would be no induction.

Some transformers have an electrostatic screen between primary and secondary. This is to prevent some types of interference being fed from the equipment down into the mains supply, or in the other direction. Transformers are sometimes used for IMPEDANCE MATCHING.

We can use the transformers as step up or step down.

Step Up transformer:

In case of step up transformer, primary windings are every less compared to secondary winding.

Because of having more turns secondary winding accepts more energy, and it releases more voltage at the output side.

Step down transformer:

Incase of step down transformer, Primary winding induces more flux than the secondary winding, and secondary winding is having less number of turns because of that it accepts less number of flux, and releases less amount of voltage.

Battery power supply:

A battery is a type of linear power supply that offers benefits that traditional line-operated power supplies lack: mobility, portability and reliability. A battery consists of multiple electrochemical cells connected to provide the voltage desired. Fig: 3.3.4 shows Hi-Watt 9V battery

Fig : Hi-Watt 9V Battery

The most commonly used dry-cell battery is the carbon-zinc dry cell battery. Dry-cell batteries are made by stacking a carbon plate, a layer of electrolyte paste, and a zinc plate alternately until the desired total voltage is achieved. The most common dry-cell batteries have one of the following voltages: 1.5, 3, 6, 9, 22.5, 45, and 90. During the discharge of a carbon-zinc battery, the zinc metal is converted to a zinc salt in the electrolyte, and magnesium dioxide is reduced at the carbon electrode. These actions establish a voltage of approximately 1.5 V.

The lead-acid storage battery may be used. This battery is rechargeable; it consists of lead and lead/dioxide electrodes which are immersed in sulfuric acid. When fully charged, this type of battery has a 2.06-2.14 V potential (A 12 volt car battery uses 6 cells in series). During discharge, the lead is converted to lead sulfate and the sulfuric acid is converted to water. When the battery is charging, the lead sulfate is converted back to lead and lead dioxide A nickel-cadmium battery has become more popular in recent years. This battery cell is completely sealed and rechargeable. The electrolyte is not involved in the electrode reaction, making the voltage constant over the span of the batteries long service life. During the charging process, nickel oxide is oxidized to its higher oxidation state and cadmium oxide is reduced. The nickel-cadmium batteries have many benefits. They can be stored both charged and uncharged. They have a long service life, high current availabilities, constant voltage, and the ability to be recharged. Fig: 3.3.5 shows pencil battery of 1.5V.

Fig : Pencil Battery of 1.5V

Step 2: Rectification

The process of converting an alternating current to a pulsating direct current is called as rectification. For rectification purpose we use rectifiers.

Rectifiers:

A rectifier is an electrical device that converts alternating current (AC) to direct current (DC), a process known as rectification. Rectifiers have many uses including as components of power supplies and as detectors of radio signals. Rectifiers may be made of solid-state diodes, vacuum tube diodes, mercury arc valves, and other components.

A device that it can perform the opposite function (converting DC to AC) is known as an inverter.

When only one diode is used to rectify AC (by blocking the negative or positive portion of the waveform), the difference between the term diode and the term rectifier is merely one of usage, i.e., the term rectifier describes a diode that is being used to convert AC to DC. Almost all rectifiers comprise a number of diodes in a specific arrangement for more efficiently converting AC to DC than is possible with only one diode. Before the development of silicon semiconductor rectifiers, vacuum tube diodes and copper (I) oxide or selenium rectifier stacks were used.

Bridge full wave rectifier:

The Bridge rectifier circuit is shown in figure, which converts an ac voltage to dc voltage using both half cycles of the input ac voltage. The Bridge rectifier circuit is shown in the figure. The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between the other two ends of the bridge.

For the positive half cycle of the input ac voltage, diodes D1 and D3 conduct, whereas diodes D2 and D4 remain in the OFF state. The conducting diodes will be in series with the load resistance R_L and hence the load current flows through R_L.

For the negative half cycle of the input ac voltage, diodes D2 and D4 conduct whereas, D1 and D3 remain OFF. The conducting diodes D2 and D4 will be in series with the load resistance R_L and hence the current flows through R_L in the same direction as in the previous half cycle. Thus a bi-directional wave is converted into a unidirectional wave.

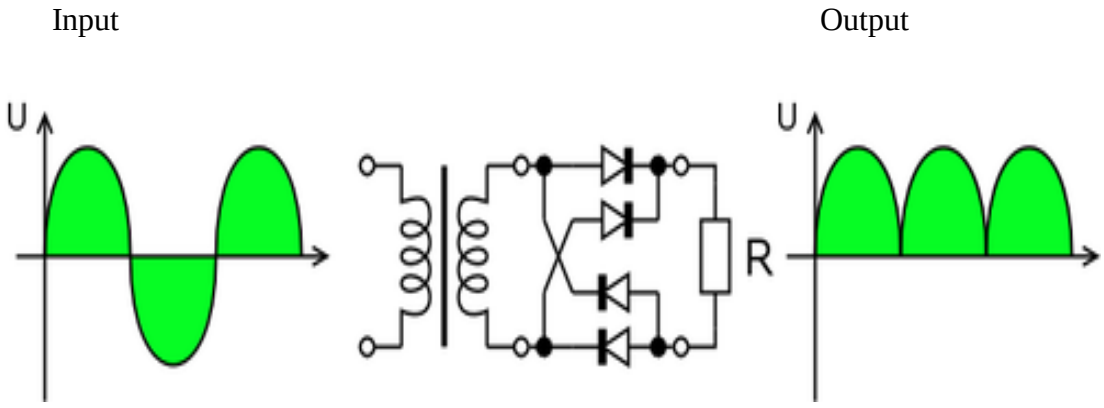


Fig : Bridge rectifier: a full-wave rectifier using 4 diodes

DB107:

Now -a -days Bridge rectifier is available in IC with a number of DB107. In our project we are using an IC in place of bridge rectifier.

Features:

- Good for automation insertion
- Surge overload rating - 30 amperes peak
- Ideal for printed circuit board
- Reliable low cost construction utilizing molded
- Glass passivated device
- Polarity symbols molded on body
- Mounting position: Any
- Weight: 1.0 gram

Fig : DB107

Step 3: Filtration

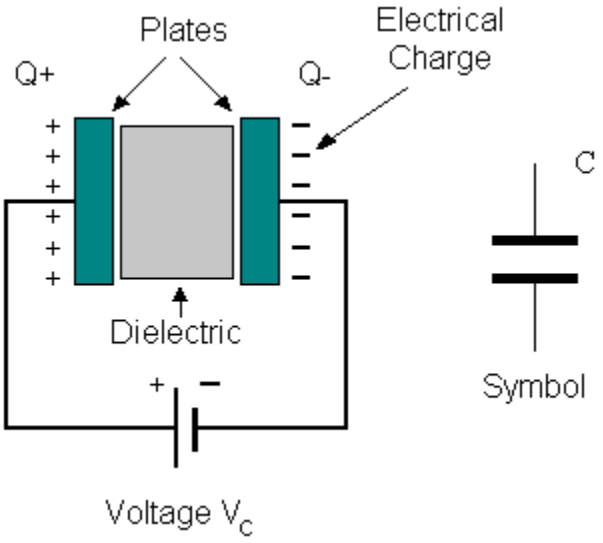
The process of converting a pulsating direct current to a pure direct current using filters is called as filtration.

Filters:

Electronic filters are electronic circuits, which perform signal-processing functions, specifically to remove unwanted frequency components from the signal, to enhance wanted ones.

5.8 Introduction to Capacitors:

The **Capacitor** or sometimes referred to as a Condenser is a passive device, and one which stores energy in the form of an electrostatic field which produces a potential (static voltage) across its plates. In its basic form a capacitor consists of two parallel conductive plates that are not connected but are electrically separated either by air or by an insulating material called the Dielectric. When a voltage is applied to these plates, a current flows charging up the plates with electrons giving one plate a positive charge and the other plate an equal and opposite negative charge. This flow of electrons to the plates is known as the Charging Current and continues to flow until the voltage across the plates (and hence the capacitor) is equal to the applied voltage Vcc. At this point the capacitor is said to be fully charged and this is illustrated below.



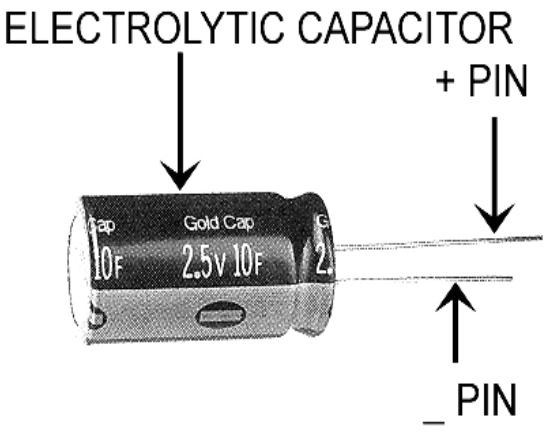


Fig:Construction Of a Capacitor **Fig :Electrolytic Capaticor**

Units of Capacitance:

Microfarad (μF) 1μF = 1/1,000,000 = 0.000001 = 10^{-6} F

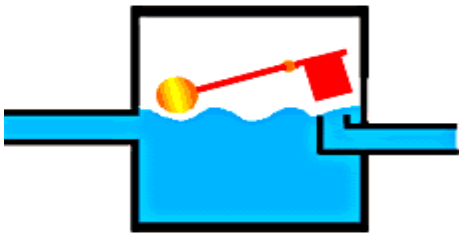
Nanofarad (nF) 1nF = 1/1,000,000,000 = 0.000000001 = 10^{-9} F

Pico farad (pF) 1pF = 1/1,000,000,000,000 = 0.000000000001 = 10^{-12} F

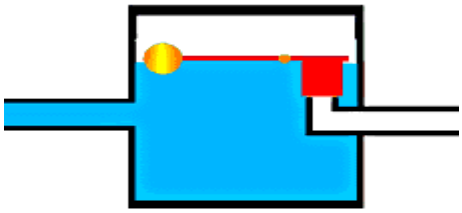
Operation of Capacitor:

Think of water flowing through a pipe. If we imagine a capacitor as being a storage tank with an inlet and an outlet pipe, it is possible to show approximately how an electronic capacitor works.

First, let's consider the case of a "coupling capacitor" where the capacitor is used to connect a signal from one part of a circuit to another but without allowing any direct current to flow.



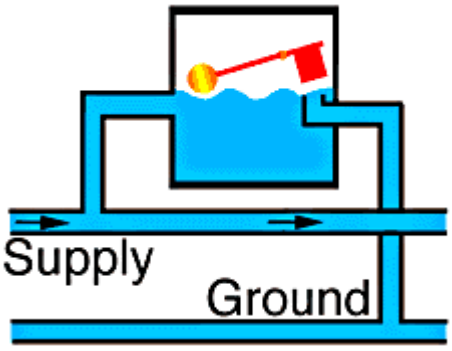
If the current flow is alternating between zero and a maximum, our "storage tank" capacitor will allow the current waves to pass through.



However, if there is a steady current, only the initial short burst will flow until the "floating ball valve" closes and stops further flow.

So a coupling capacitor allows "alternating current" to pass through because the ball valve doesn't get a chance to close as the waves go up and down. However, a steady current quickly fills the tank so that all flow stops.

A capacitor will pass alternating current but (apart from an initial surge) it will not pass d.c.



Where a capacitor is used to decouple a circuit, the effect is to "smooth out ripples". Any ripples, waves or pulses of current are passed to ground while d.c. Flows smoothly.

Step 4: Regulation

The process of converting a varying voltage to a constant regulated voltage is called as regulation. For the process of regulation we use voltage regulators.

Voltage Regulator:

A voltage regulator (also called a 'regulator') with only three terminals appears to be a simple device, but it is in fact a very complex integrated circuit. It converts a varying input voltage into a constant 'regulated' output voltage. Voltage Regulators are available in a variety of outputs like 5V, 6V, 9V, 12V and 15V. The LM78XX series of voltage regulators are designed for positive input. For applications requiring negative input, the LM79XX series is used. Using a pair of 'voltage-divider' resistors can increase the output voltage of a regulator circuit.

It is not possible to obtain a voltage lower than the stated rating. You cannot use a 12V regulator to make a 5V power supply. Voltage regulators are very robust. These can withstand over-current draw due to short circuits and also over-heating. In both cases, the regulator will cut off before any damage occurs. The only way to destroy a regulator is to apply reverse voltage to its input. Reverse polarity destroys the regulator almost instantly. Fig: 3.3.10 shows voltage regulator.

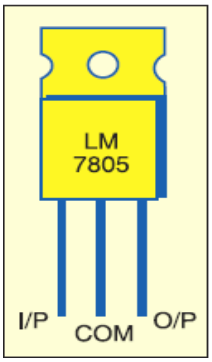


Fig : Voltage Regulator

Resistors:

A resistor is a two-terminal electronic component that produces a voltage across its terminals that is proportional to the electric current passing through it in accordance with Ohm's law:

$$V = IR$$

Resistors are elements of electrical networks and electronic circuits and are ubiquitous in most electronic equipment. Practical resistors can be made of various compounds and films, as well as resistance wire (wire made of a high-resistivity alloy, such as nickel/chrome).

The primary characteristics of a resistor are the resistance, the tolerance, maximum working voltage and the power rating. Other characteristics include temperature coefficient, noise, and inductance. Less well-known is critical resistance, the value below which power dissipation limits the maximum permitted current flow, and above which the limit is applied voltage. Critical resistance is determined by the design, materials and dimensions of the resistor.

Resistors can be made to control the flow of current, to work as Voltage dividers, to dissipate power and it can shape electrical waves when used in combination of other components. Basic unit is ohms.

Theory of operation: Ohm's law: The behavior of an ideal resistor is dictated by the relationship specified in Ohm's law. V = IR

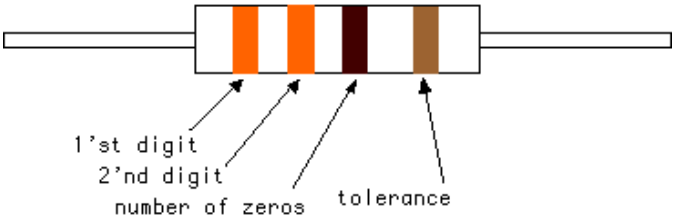
Ohm's law states that the voltage (V) across a resistor is proportional to the current (I) through it where the constant of proportionality is the resistance (R).

Power dissipation:The power dissipated by a resistor (or the equivalent resistance of a resistor network) is calculated using the following:

$$P = I^2R = IV = \frac{V^2}{R}$$

Fig : Resistor

Resistor color Coding



Digit	color	Tolerance	color
0	Black	20%	nothing
1	Brown	10%	Silver
2	Red	5%	Gold
3	Orange	2%	Red
4	Yellow	1%	Brown
5	Green		
6	Blue		
7	Violet		
8	Grey		
9	White		

Fig : Color Bands In Resistor

LED:

A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices, and are increasingly used for lighting. Introduced as a practical electronic component in 1962, early LEDs emitted low-intensity red light, but modern versions are available across the visible, ultraviolet and infrared wavelengths, with very high brightness. The internal structure and parts of a led are shown below.

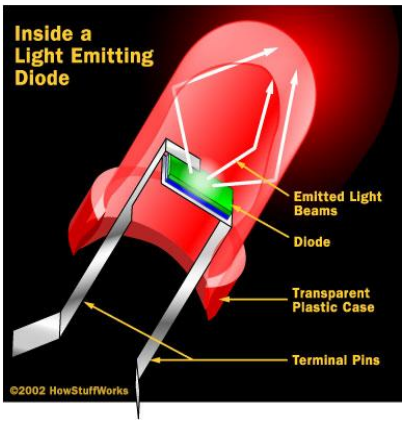


Fig : Inside a LED

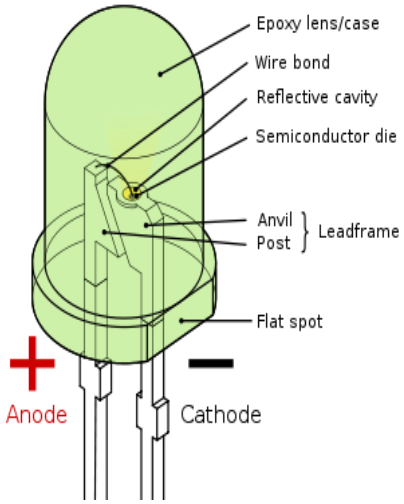


Fig : Parts of a LED

Working:

The structure of the LED light is completely different than that of the light bulb. Amazingly, the LED has a simple and strong structure. The light-emitting semiconductor material is what determines the LED's color. The LED is based on the semiconductor diode.

When a diode is forward biased (switched on), electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence and the color of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor. An LED is usually small in area (less than

1 mm^2), and integrated optical components are used to shape its radiation pattern and assist in reflection. LEDs present many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved robustness, smaller size, faster switching, and greater durability and reliability. However, they are relatively expensive and require more precise current and heat management than traditional light sources. Current LED products for general lighting are more expensive to buy than fluorescent lamp sources of comparable output. They also enjoy use in applications as diverse as replacements for traditional light sources in automotive lighting (particularly indicators) and in traffic signals. The compact size of LEDs has allowed new text and video displays and sensors to be developed, while their high switching rates are useful in advanced communications technology. The electrical symbol and polarities of led are shown in fig:

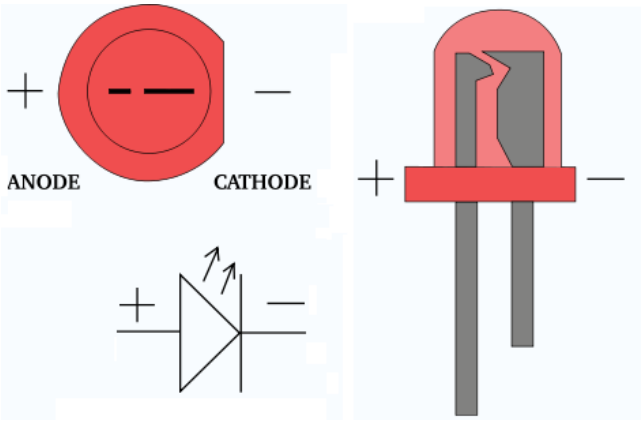


Fig : Electrical Symbol & Polarities of LED

LED lights have a variety of advantages over other light sources:

- High-levels of brightness and intensity
- High-efficiency
- Low-voltage and current requirements
- Low radiated heat
- High reliability (resistant to shock and vibration)
- No UV Rays
- Long source life

- Can be easily controlled and programmed

Applications of LED fall into three major categories:

- Visual signal application where the light goes more or less directly from the LED to the human eye, to convey a message or meaning.
- Illumination where LED light is reflected from object to give visual response of these objects.
- Generate light for measuring and interacting with processes that do not involve the human visual system.

5.9 GSM (Global System for Mobile communications)

Introduction:

GSM (Global System for Mobile communications) is a cellular network, which means that mobile phones connect to it by searching for cells in the immediate vicinity. GSM networks operate in four different frequency ranges. Most GSM networks operate in the 900 MHz or 1800 MHz bands. Some countries in the Americas use the 850 MHz and 1900 MHz bands because the 900 and 1800 MHz frequency bands were already allocated.

The rarer 400 and 450 MHz frequency bands are assigned in some countries, where these frequencies were previously used for first-generation systems.

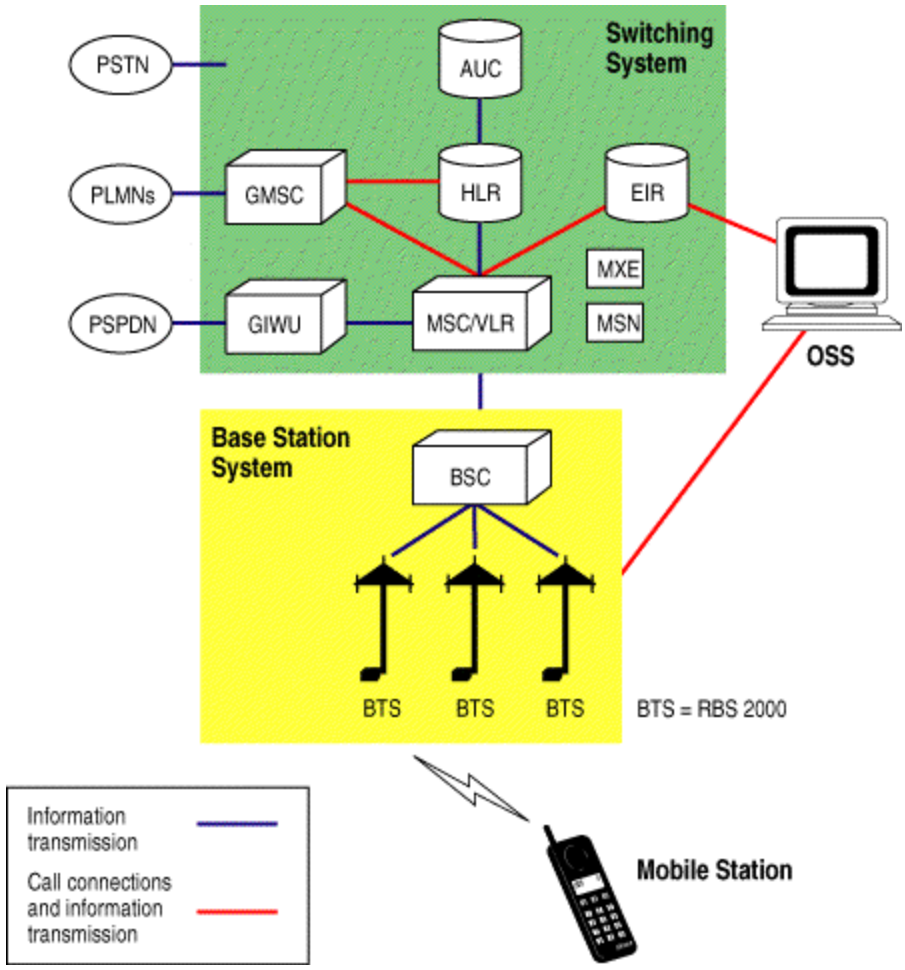
GSM-900 uses 890–915 MHz to send information from the mobile station to the base station (uplink) and 935–960 MHz for the other direction (downlink), providing 124 RF channels (channel numbers 1 to 124) spaced at 200 kHz. Duplex spacing of 45 MHz is used. In some countries the GSM-900 band has been extended to cover a larger frequency range. This 'extended GSM', E-GSM, uses 880–915 MHz (uplink) and 925–960 MHz (downlink), adding 50 channels (channel numbers 975 to 1023 and 0) to the original GSM-900 band. Time division multiplexing is used to allow eight full-rate or sixteen half-rate speech channels per radio frequency channel. There are eight radio timeslots (giving eight burst periods) grouped into what is called a TDMA frame. Half rate channels use alternate frames in the same timeslot. The channel data rate is 270.833 kbit/s, and the frame duration is 4.615 ms.

GSM Advantages:

GSM also pioneered a low-cost, to the network carrier, alternative to voice calls, the Short t message service (SMS, also called "text messaging"), which is now supported on other mobile standards as well. Another advantage is that the standard includes one worldwide Emergency telephone number, 112. This makes it easier for international travelers to connect to emergency services without knowing the local emergency number.

The GSM Network:

GSM provides recommendations, not requirements. The GSM specifications define the functions and interface requirements in detail but do not address the hardware. The GSM network is divided into three major systems: the switching system (SS), the base station system (BSS), and the operation and support system (OSS).



The Switching System:

The switching system (SS) is responsible for performing call processing and subscriber-related functions. The switching system includes the following functional units.

- **Home location register (HLR)**: The HLR is a database used for storage and management of subscriptions. The HLR is considered the most important database, as it stores permanent data about subscribers, including a subscriber's service profile, location

information, and activity status. When an individual buys a subscription from one of the PCS operators, he or she is registered in the HLR of that operator.

- **Mobile services switching center (MSC):** The MSC performs the telephony switching functions of the system. It controls calls to and from other telephone and data systems. It also performs such functions as toll ticketing, network interfacing, common channel signaling, and others.
- **Visitor location register (VLR)**: The VLR is a database that contains temporary information about subscribers that is needed by the MSC in order to service visiting subscribers. The VLR is always integrated with the MSC. When a mobile station roams into a new MSC area, the VLR connected to that MSC will request data about the mobile station from the HLR. Later, if the mobile station makes a call, the VLR will have the information needed for call setup without having to interrogate the HLR each time.
- **Authentication center (AUC)**: A unit called the AUC provides authentication and encryption parameters that verify the user's identity and ensure the confidentiality of each call. The AUC protects network operators from different types of fraud found in today's cellular world.
- **Equipment identity register (EIR)**: The EIR is a database that contains information about the identity of mobile equipment that prevents calls from stolen, unauthorized, or defective mobile stations. The AUC and EIR are implemented as stand-alone nodes or as a combined AUC/EIR node.

The Base Station System (BSS):

All radio-related functions are performed in the BSS, which consists of base station controllers (BSCs) and the base transceiver stations (BTSs).

- **BSC**: The BSC provides all the control functions and physical links between the MSC and BTS. It is a high-capacity switch that provides functions such as handover, cell configuration data, and control of radio frequency (RF) power levels in base transceiver stations. A number of BSCs are served by an MSC.

- **BTS**: The BTS handles the radio interface to the mobile station. The BTS is the radio equipment (transceivers and antennas) needed to service each cell in the network. A group of BTSs are controlled by a BSC.

The Operation and Support System

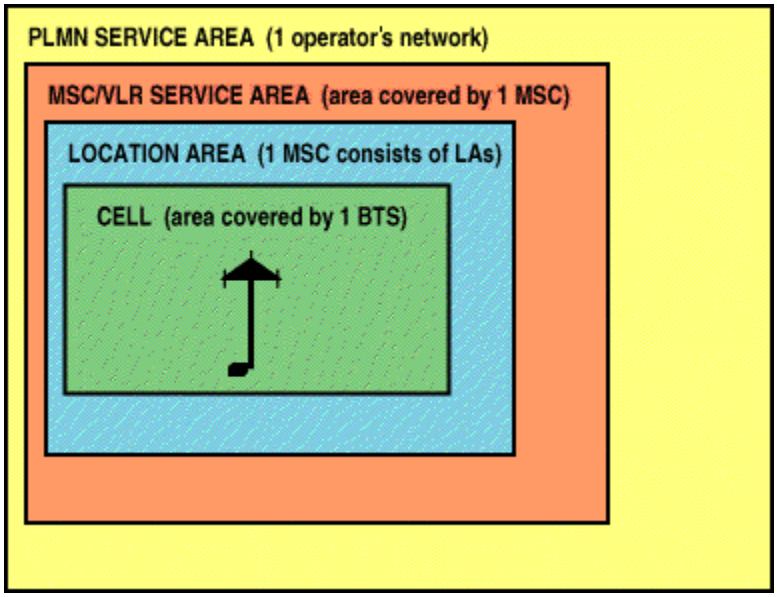
The operations and maintenance center (OMC) is connected to all equipment in the switching system and to the BSC. The implementation of OMC is called the operation and support system (OSS). The OSS is the functional entity from which the network operator monitors and controls the system. The purpose of OSS is to offer the customer cost-effective support for centralized, regional and local operational and maintenance activities that are required for a GSM network. An important function of OSS is to provide a network overview and support the maintenance activities of different operation and maintenance organizations.

Additional Functional Elements

- **Message center (MXE)**: The MXE is a node that provides integrated voice, fax, and data messaging. Specifically, the MXE handles short message service, cell broadcast, voice mail, fax mail, e-mail, and notification.
- **Mobile service node (MSN)**: The MSN is the node that handles the mobile intelligent network (IN) services.
- **Gateway mobile services switching center (GMSC)**: A gateway is a node used to interconnect two networks. The gateway is often implemented in an MSC. The MSC is then referred to as the GMSC.
- **GSM inter-working unit (GIWU)**: The GIWU consists of both hardware and software that provides an interface to various networks for data communications. Through the GIWU, users can alternate between speech and data during the same call. The GIWU hardware equipment is physically located at the MSC/VLR.

GSM Network Areas:

The GSM network is made up of geographic areas. As shown in bellow figure, these areas include cells, location areas (LAs), MSC/VLR service areas, and public land mobile network (PLMN) areas.



Location Areas:

The cell is the area given radio coverage by one base transceiver station. The GSM network identifies each cell via the cell global identity (CGI) number assigned to each cell. The location area is a group of cells. It is the area in which the subscriber is paged. Each LA is served by one or more base station controllers, yet only by a single MSC Each LA is assigned a location area identity (LAI) number.

MSC/VLR service areas:

An MSC/VLR service area represents the part of the GSM network that is covered by one MSC and which is reachable, as it is registered in the VLR of the MSC.

PLMN service areas:

The PLMN service area is an area served by one network operator.

GSM Specifications:

Specifications for different personal communication services (PCS) systems vary among the different PCS networks. Listed below is a description of the specifications and characteristics for GSM.

- **Frequency band**: The frequency range specified for GSM is 1,850 to 1,990 MHz (mobile station to base station).
- **Duplex distance**: The duplex distance is 80 MHz. Duplex distance is the distance between the uplink and downlink frequencies. A channel has two frequencies, 80 MHz apart.
- **Channel separation**: The separation between adjacent carrier frequencies. In GSM, this is 200 kHz.
- **Modulation**: Modulation is the process of sending a signal by changing the characteristics of a carrier frequency. This is done in GSM via Gaussian minimum shift keying (GMSK).
- **Transmission rate**: GSM is a digital system with an over-the-air bit rate of 270 kbps.
- **Access method**: GSM utilizes the time division multiple access (TDMA) concept. TDMA is a technique in which several different calls may share the same carrier. Each call is assigned a particular time slot.
- **Speech coder**: GSM uses linear predictive coding (LPC). The purpose of LPC is to reduce the bit rate. The LPC provides parameters for a filter that mimics the vocal tract. The signal passes through this filter, leaving behind a residual signal. Speech is encoded at 13 kbps.

GSM Subscriber Services:

Dual-tone multifrequency (DTMF): DTMF is a tone signaling scheme often used for various control purposes via the telephone network, such as remote control of an answering machine. GSM supports full-originating DTMF.

Facsimile group III—GSM supports CCITT Group 3 facsimile. As standard fax machines are designed to be connected to a telephone using analog signals, a special fax converter connected to the exchange is used in the GSM system. This enables a GSM–connected fax to communicate with any analog fax in the network.

Short message services: A convenient facility of the GSM network is the short message service. A message consisting of a maximum of 160 alphanumeric characters can be sent to or from a mobile station. This service can be viewed as an advanced form of alphanumeric paging with a number of advantages. If the subscriber's mobile unit is powered off or has left the coverage area, the message is stored and offered back to the subscriber when the mobile is powered on or has reentered the coverage area of the network. This function ensures that the message will be received.

Cell broadcast: A variation of the short message service is the cell broadcast facility. A message of a maximum of 93 characters can be broadcast to all mobile subscribers in a certain geographic area. Typical applications include traffic congestion warnings and reports on accidents.

Voice mail: This service is actually an answering machine within the network, which is controlled by the subscriber. Calls can be forwarded to the subscriber's voice-mail box and the subscriber checks for messages via a personal security code.

Fax mail: With this service, the subscriber can receive fax messages at any fax machine. The messages are stored in a service center from which they can be retrieved by the subscriber via a personal security code to the desired fax number

Supplementary Services:

GSM supports a comprehensive set of supplementary services that can complement and support both telephony and data services.

Call forwarding: This service gives the subscriber the ability to forward incoming calls to another number if the called mobile unit is not reachable, if it is busy, if there is no reply, or if call forwarding is allowed unconditionally.

Barring of outgoing calls: This service makes it possible for a mobile subscriber to prevent all outgoing calls.

Barring of incoming calls: This function allows the subscriber to prevent incoming calls. The following two conditions for incoming call barring exist: baring of all incoming calls and barring of incoming calls when roaming outside the home PLMN.

Advice of charge (AoC): The AoC service provides the mobile subscriber with an estimate of the call charges. There are two types of AoC information: one that provides the subscriber with an estimate of the bill and one that can be used for immediate charging purposes. AoC for data calls is provided on the basis of time measurements.

Call hold: This service enables the subscriber to interrupt an ongoing call and then subsequently reestablish the call. The call hold service is only applicable to normal telephony.

Call waiting: This service enables the mobile subscriber to be notified of an incoming call during a conversation. The subscriber can answer, reject, or ignore the incoming call. Call waiting is applicable to all GSM telecommunications services using a circuit-switched connection.

Multiparty service: The multiparty service enables a mobile subscriber to establish a multiparty conversation—that is, a simultaneous conversation between three and six subscribers. This service is only applicable to normal telephony.

Calling line identification presentation/restriction: These services supply the called party with the integrated services digital network (ISDN) number of the calling party. The restriction service enables the calling party to restrict the presentation. The restriction overrides the presentation.

Closed user groups (CUGs): CUGs are generally comparable to a PBX. They are a group of subscribers who are capable of only calling themselves and certain numbers

Main AT commands:

"AT command set for GSM Mobile Equipment" describes the Main AT commands to communicate via a serial interface with the GSM subsystem of the phone.

AT commands are instructions used to control a modem. AT is the abbreviation of Attention. Every command line starts with "AT" or "at". That's why modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems, such as ATD (Dial), ATA (Answer), ATH (Hook control) and ATO (Return to online data state), are also supported by GSM/GPRS modems and mobile phones. Besides this common AT command set, GSM/GPRS modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS-related commands like AT+CMGS (Send SMS message), AT+CMSS (Send SMS message from storage), AT+CMGL (List SMS messages) and AT+CMGR (Read SMS messages).

Note that the starting "AT" is the prefix that informs the modem about the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD and +CMGS is the actual AT command name in AT+CMGS. However, some books and web sites use them interchangeably as the name of an AT command.

Here are some of the tasks that can be done using AT commands with a GSM/GPRS modem or mobile phone:

- Get basic information about the mobile phone or GSM/GPRS modem. For example, name of manufacturer (AT+CGMI), model number (AT+CGMM), IMEI number (International Mobile Equipment Identity) (AT+CGSN) and software version (AT+CGMR).
- Get basic information about the subscriber. For example, MSISDN (AT+CNUM) and IMSI number (International Mobile Subscriber Identity) (AT+CIMI).
- Get the current status of the mobile phone or GSM/GPRS modem. For example, mobile phone activity status (AT+CPAS), mobile network registration status (AT+CREG), radio signal strength (AT+CSQ), battery charge level and battery charging status (AT+CBC).
- Establish a data connection or voice connection to a remote modem (ATD, ATA, etc).
- Send and receive fax (ATD, ATA, AT+F*).

- Send (AT+CMGS, AT+CMSS), read (AT+CMGR, AT+CMGL), write (AT+CMGW) or delete (AT+CMGD) SMS messages and obtain notifications of newly received SMS messages (AT+CNMI).
- Read (AT+CPBR), write (AT+CPBW) or search (AT+CPBF) phonebook entries.
- Perform security-related tasks, such as opening or closing facility locks (AT+CLCK), checking whether a facility is locked (AT+CLCK) and changing passwords (AT+CPWD).

  (Facility lock examples: SIM lock [a password must be given to the SIM card every time the mobile phone is switched on] and PH-SIM lock [a certain SIM card is associated with the mobile phone. To use other SIM cards with the mobile phone, a password must be entered.])
- Control the presentation of result codes / error messages of AT commands. For example, you can control whether to enable certain error messages (AT+CMEE) and whether error messages should be displayed in numeric format or verbose format (AT+CMEE=1 or AT+CMEE=2).
- Get or change the configurations of the mobile phone or GSM/GPRS modem. For example, change the GSM network (AT+COPS), bearer service type (AT+CBST), radio link protocol parameters (AT+CRLP), SMS center address (AT+CSCA) and storage of SMS messages (AT+CPMS).
- Save and restore configurations of the mobile phone or GSM/GPRS modem. For example, save (AT+CSAS) and restore (AT+CRES) settings related to SMS messaging such as the SMS center address.

5.10 BLUE TOOTH:

The RN42 is a small form factor, low power, highly economic Bluetooth radio for OEM's adding wireless capability to their products. The RN42 supports multiple interface protocols, is simple to design in and fully certified, making it a complete embedded Bluetooth solution. The RN 42 is functionally compatible with RN 41. With its high performance on chip antenna and support for Bluetooth® Enhanced Data Rate (EDR), the RN42 delivers up to 3 Mbps data rate for distances to 20M. The RN-42 also comes in a package with no antenna (RN-42-N). Useful when the application requires an external antenna, the RN-42-N is shorter in length and has RF pads to route the antenna signal.

Features

- Fully qualified Bluetooth 2.1/2.0/1.2/1.1 module
- Bluetooth v2.0+EDR support
- Available with on board chip antenna (RN- 42) and without antenna (RN-42-N)
- Postage stamp sized form factor, 13.4mm x 25.8 mm x 2mm (RN-42) and 13.4mm x 20 mm x 2 mm (RN-42-N)
- Low power (26uA sleep, 3mA connected, 30mA transmit)
- UART (SPP or HCI) and USB (HCI only) data connection interfaces.
- Sustained SPP data rates - 240Kbps (slave), 300Kbps (master)
- HCI data rates - 1.5Mbps sustained, 3.0Mbps burst in HCI mode
- Embedded Bluetooth stack profiles included (requires no host stack): GAP, SDP, RFCOMM and L2CAP protocols, with SPP and DUN profile support.
- Bluetooth SIG certified • Castellated SMT pads for easy and reliable PCB mounting
- Certifications: FCC, ICS, CE
- Environmentally friendly, RoHS compliant

Applications

- Cable replacement
- Barcode scanners
- Measurement and monitoring systems
- Industrial sensors and controls
- Medical devices
- Barcode readers

- Computer accessories

IR:

What is an Infrared Receiver?

An infrared receiver, or IR receiver, is hardware that sends information from an infrared remote control to another device by receiving and decoding signals. In general, the receiver outputs a code to uniquely identify the infrared signal that it receives. This code is then used in order to convert signals from the remote control into a format that can be understood by the other device. It is the part of a device that receives infrared commands from a remote control. Because infrared is light, it requires line-of-sight visibility for the best possible operation, but can however still be reflected by items such as glass and walls. Poorly placed IR receivers can result in what is called "tunnel vision", where the operational range of a remote control is reduced because they are set so far back into the chassis of a device.

Types of Infrared Receivers

There are many different kinds of infrared receivers and at Future Electronics we stock many of the most common types categorized by supply voltage, carrier frequency, transmission distance, power dissipation, packaging type and supply current. The parametric filters on our website can help refine your search results depending on the required specifications.

The most common sizes for carrier frequency are 36 kHz, 37.9 kHz, 38 kHz and 40 kHz. We also carry infrared receivers with carrier frequency up to 56 kHz. The transmission distance can range from 8 m to 45 m, with the most common infrared receivers having a transmission distance of 45 m.

Infrared Receivers from Future Electronics

Future Electronics has a full selection of programmable infrared receivers from several manufacturers where you can find the right chip that can be used for an IR receiver circuit, USB IR receiver, IR receiver for PC, IR receiver cable, IR remote receiver, wireless IR receiver, IR receiver IC or any IR module. Simply choose from the infrared receiver technical attributes below and your search results will quickly be narrowed to match your specific infrared receiver application needs.

If you have a preferred brand, we deal with several manufacturers such as Everlight, New Japan Radio, Optek, ROHM Semiconductor, Vishay or Sharp. You can easily refine your infrared receiver product search results by clicking your preferred infrared receiver brand below from our list of manufacturers.

Applications for Infrared Receivers:

Infrared receivers can often be found in consumer products such as television remote controls or infrared ports such as PDAs, laptops, and computers. They are also present in devices such as home theatres, cable or satellite receivers, VCRs, DVD and Blu-Ray players and audio amplifiers. Infrared receivers can also be found in the industrial, military, aerospace and photography markets.

Choosing the Right Infrared Receiver:

When you are looking for the right infrared receivers, with the FutureElectronics.com parametric search, you can filter the results by various attributes: by Supply Current (5 uA, 450 uA, 1.5 mA,…), Transmission Distance (8m, 12 m, 35 m,…) and Supply Voltage (up to 32 V) to name a few. You will be able to find the right chip for your USB IR receiver, IR receiver circuit, IR receiver for PC, IR receiver cable, IR receiver IC, IR remote receiver, wireless IR receiver or any other IR module.

Infrared Receivers in Production Ready Packaging or R&D Quantities

If the quantity of infrared receivers required is less than a full reel, we offer customers many of our programmable infrared receiver products in tube, tray or individual quantities that will avoid unneeded surplus.

In addition, Future Electronics offers clients a unique bonded inventory program that is designed to eliminate potential problems that could arise from an unpredictable supply of products containing raw metals and products with long or erratic lead times..

CHAPTER-6

IMPLEMENTATION PHASE & RESULTS

6.1 Operation:

In this project LPC2148 plays major role. It was belongs to ARM7 architecture. Traffic signal lights are connected to microcontroller. Microcontroller runs traffic lights with delay. With the help of IR sensors microcontroller can detects density of vehicles. This density information transmitted to mobile phone through GSM and Bluetooth. Bluetooth data useful for smart phones.

This project uses regulated 5V, 500mA power supply. 7805 three terminal voltage regulator is used for voltage regulation. Bridge type full wave rectifier is used to rectify the ac output of secondary of 230/12V step down transformer.

6.2 INTERFACING OF HARDWARE COMPONENTS

POWER SUPPLY UNIT

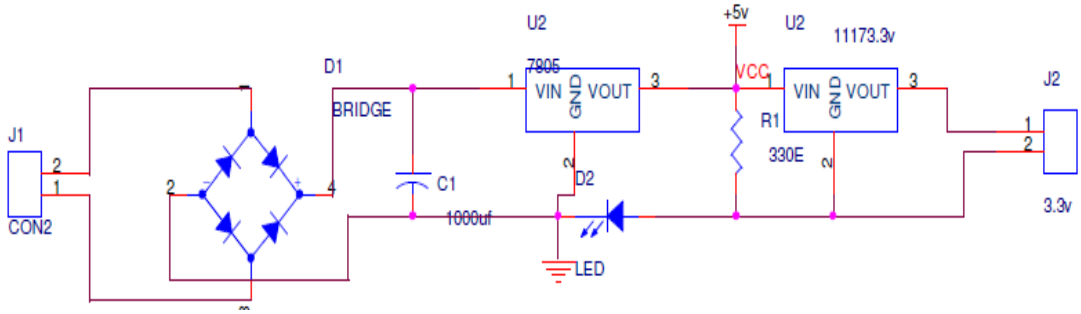


Fig : Power supply unit

- As soon as the system is ON, the AC Voltage Received from the Supply is Converted to DC voltage using a Bridge Rectifier .
- 7805 IC voltage regulator gets an Input of 12V and outputs 5V . This 5V is supplied to all Peripheral devices which needs a supply of 5V.

- 1117S Voltage Regulator takes input of 5V and outputs 3.3V which is needed voltage for Micro controller LPC2148.

EM-18 GSM INTERFACING WITH ARM7 (LPC2148)

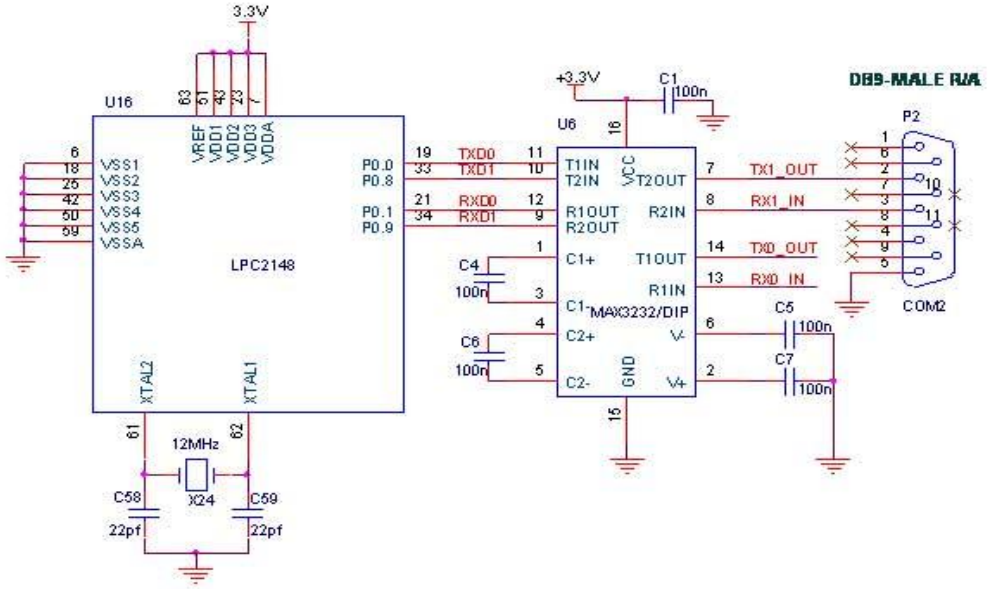


Fig : Interfacing GSM with LPC2148

- MAX 232 is the voltage level converter, converts voltages from TTL to wire logic or vice versa.
- MAX 232 is connected to UART1 ports p0.8-p0.9 of LPC2148 uc Through RS 232 cable.
- UART1 receives the data and Received data will be displayed on LCD .

6.3 INTERFACING LCD WITH ARM7 (LPC2148)

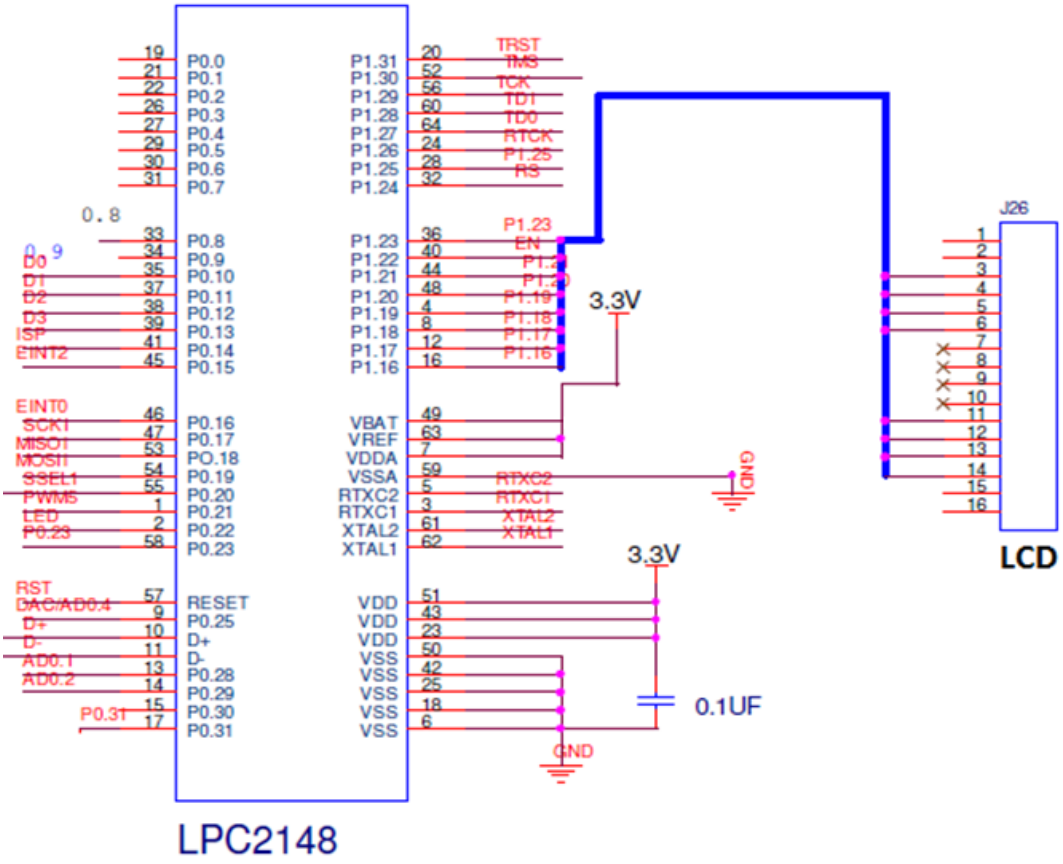


Fig : LCD Interfacing with LPC2148

- A simple schematic for interfacing a 16x2 LCD with LPC2148 is shown above.
- As you can see in the following circuit, eight pins are needed for interfacing connected to Port pins P1.16-P1.23.
- As we are operating the LCD in 4 bit mode, we require only four data pins saving the other four pins for any other use of microcontroller

6.4 SOFTWARE TOOLS -6

6.4.1 INTRODUTION TO ORCAD

OrCAD Circuit Design

- This tool is used to design the schematic of the hardware.
- Using Orcad the PCB layout is designed Keil IDE's:
- This tool is used to develop the source code needed for the design.
- The tool helps us not only to develop but also compile the code and simulate the code.
- The Keil tool is also used to convert the compiled Embedded C code to its equivalent hex code.

Flash Programmer

- Flash programmer is used to fuse the built hex code into the Microcontroller AT89c51 (here).
- **Language**: Embedded C.

ORCAD CAPTURE CIS

OrCAD Capture CIS is designed to reduce production delays and cost overruns through efficient management of components. It reduces the time spent searching existing parts for reuse, manually entering part information content, and maintaining component data. Users search parts based on their electrical characteristics and OrCAD Capture CIS automatically retrieves the associated part.

Flexible and scalable, the solution is quickly implemented. OrCad Capture CIS is ideal for individual design teams or multi-site teams who need to collaborate across multiple locations, OrCAD Capture CIS gives designers access to correct part data early in the design process and enables complete component specifications to be passed to board designers and other members of the design team, reducing the potential for downstream errors.

It provides access to cost information so designers can use preferred, lower cost, and instock parts. The embedded part selector accesses information stored in MRP/ERP systems and engineering databases and synchronizes externally sourced data with the schematic design database, so bills of materials can be automatically generated.

BENEFITS

- Provides fast, intuitive schematic editing.
- Boosts schematic editing efficiency by design reuse
- Automates the integration of FPGA and PLD devices
- Makes changes quickly through a single spreadsheet editor
- Imports and exports virtually every commonly used design file format
- Reduces delays caused by out-of-stock parts (CIS)
- Promotes reuse of preferred components (CIS)
- Encourages reuse of known good part data (CIS)
- Makes reuse of duplicate circuitry easy through hierarchical blocks (CIS)

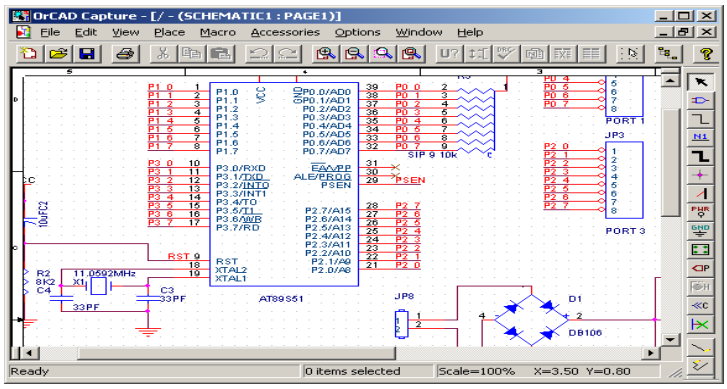


Fig 6.1: Orcad Schematic capture window

6.5 KEIL C COMPILER

Keil Software publishes one of the most complete development tool suites for 8051 software, which is used throughout industry. For development of C code, their Developer's Kit product includes their C51 compiler, as well as an integrated 8051 simulator for debugging. A demonstration version of this product is available on their website, but it includes several limitations.

The C programming language was designed for computers, though, and not embedded systems. It does not support direct access to registers, nor does it allow for the reading and setting of single bits, two very important requirements for 8051 software. In addition, most software developers are accustomed to writing programs that will by executed by an operating system, which provides system calls the program may use to access the hardware. However, much code for the 8051 is written for direct use on the processor, without an operating system.

To support this, the Keil compiler has added several extensions to the C language to replace what might have normally been implemented in a system call, such as the connecting of interrupt handlers.

The purpose of this manual is to further explain the limitations of the Keil compiler, the modifications it has made to the C language, and how to account for these in developing software for the 8051 microcontroller.

Keil Limitations

There are several very important limitations in the evaluation version of Keil's Developer's Kit that users need be aware of when writing software for the 8051.

Object code must be less than 2 Kbytes

The compiler will compile any-sized source code file, but the final object code may not exceed 2 Kbytes. If it does, the linker will refuse to create a final binary executable (or HEX file) from it. Along the same lines, the debugger will refuse any files that are over 2Kbytes, even if they were compiled using a different software package.

Few student projects will cross this 2Kbyte threshold, but programmers should be aware of it to understand why code may no longer compile when the project grows too large.

Program code starts at address 0x4000

All C code compiled and linked using the Keil tools will begin at address 0x4000 in code memory. Such code may not be programmed into devices with less than 16Kbytes of Read-Only Memory. Code written in assembly may circumvent this limitation by using the "origin" keyword to set the start to address 0x0000. No such work-around exists for C programs, though. However, the integrated debugger in the evaluation software may still be used for testing code. Once tested, the code may be compiled by the full version of the Keil software, or by another compiler that supports the C extensions used by Keil.

C Modifications

The Keil C compiler has made some modifications to another wise ANSI-compliant implementation of the C programming language. These modifications were made solely to facilitate the use of a higher-level language like C for writing programs on microcontrollers.

Variable Types

The Keil C compiler supports most C variable types and adds several of its own.

Standard Types

The evaluation version of the Keil C compiler supports the standard ANSI C variable types, with the exception of the floating-point types. These types are summarized below.

Table 6.1: Range of Datatypes

Type	Bits	Bytes	Range
Char	8	1	-128 to +127
Unsigned char	8	1	0 to 255
Enum	16	2	-32,768 to +32,767
Short	16	2	-32,768 to +32,767
Unsigned short	16	2	0 to 65,535
Int	16	2	-32,768 to +32,767
Unsigned int	16	2	0 to 65,535
Long	32	4	-2,147,483,648 to +2,147,483,647
Unsigned long	32	4	0 to 4,294,697,295

In addition to these variable types, the compiler also supports the **struct** and **union** data structures, as well as type redefinition using **typedef**.

Keil Types

To support a microcontroller and embedded systems applications, Keil added several new types to their compiler. These are summarized in the table below.

Table : Different Keil Types and ranges

	Bits	Bytes	Range
Bit	1	0	0 to 1
Sbit	1	0	0 to 1
Sfr	8	1	0 to 255
sf16	16	2	0 to 65,535

Of these, only the **bit** type works as a standard variable would. The other three have special behavior that a programmer must be aware of.

Bit

This is a data type that gets allocated out of the 8051's bit-addressable on-chip RAM. Like other data types, it may be declared as either a variable. However, unlike standard C types, if may not be used as a pointer.

Sbit, sfr, and sf16

These are special types for accessing 1-bit, 8-bit, and 16-bit special function registers. Because there is no way to indirectly address registers in the 8051, addresses for these variables must be declared outside of functions within the code. Only the data addressed by the variable may be manipulated in the code.

Conveniently, the standard special function registers are all defined in the reg51.h file that any developer may include into their source file. Only registers unique to the particular

8051-derivative being used for the project need have these variable declared, such as registers and bits related to a second on-chip serial port.

Keil Variable Extensions

In writing applications for a typical computer, the operating system handles manages memory on behalf of the programs, eliminating their need to know about the memory structure of the hardware. Even more important, most computers having a unified memory space, with the code and data sharing the same RAM. This is not true with the 8051, which has separate memory spaces for code, on-chip data, and external data.

Accommodate for this when writing C code, Keil added extensions to variable declarations to specify which memory space the variable is allocated from, or points to. The most important of these for student programmers are summarized in the following table.

Table : Memory Types and spaces

Extension	Memory Type	Related ASM
Data	Directly-addressable data memory (data memory addresses 0x00-0x7F)	MOV A, 07Fh
Idata	Indirectly-addressable data memory (data memory addresses 0x00-0xFF)	MOV R0, #080h MOV A, R0
Xdata	External data memory	MOVX @DPTR
Code	Program memory	MOVC @A+DPTR

These extensions may be used as part of the variable type in declaration or casting by placing the extension after the type, as in the example below. If the memory type extension is not specified, the compiler will decide which memory type to use automatically, based on the memory model.

Keil Function Extensions

Keil provides two important extensions to the standard function declaration to allow for creation of interrupt handlers and reentrant functions.

CREATING A PROJECT IN KEIL µvision4

1. Start up Keil µVision4
2. Under the "Project" menu, select "New µVision Project…"
3. Navigate to a folder where the project will be created, and give it a name. Click "OK" to continue. If you have downloaded ValvanoWare, you can put the new project in there. For most systems you will need access to the **driverlib** and the **inc** folders.

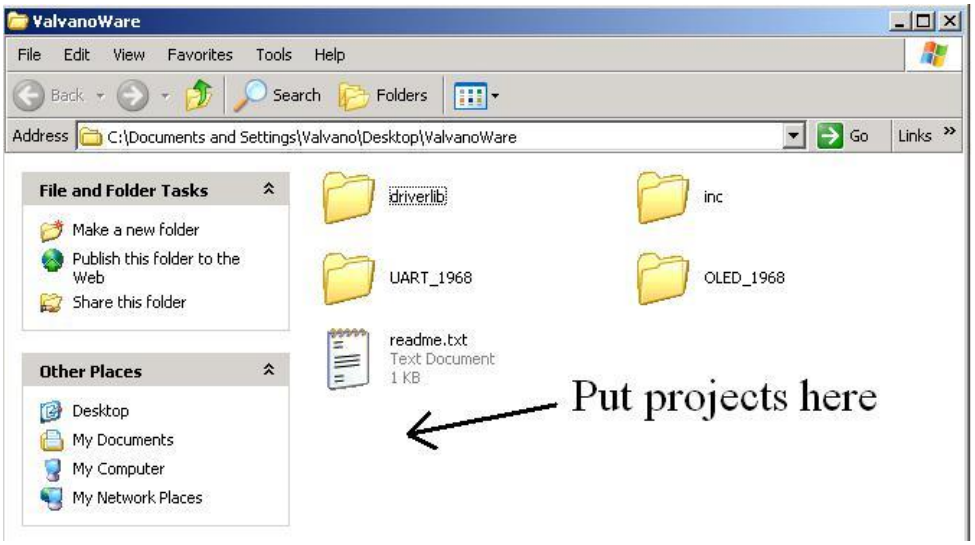


Fig : Select a folder to save the project

4. Click the [+] next to the manufacturer of your device and find your device from the resulting list. For example, click "[+] Texas Instruments" and then click on "LM3S1968". For older versions of the compiler, Stellaris microcontrollers can be found under "Luminary Micro". Your selection should look something like this:

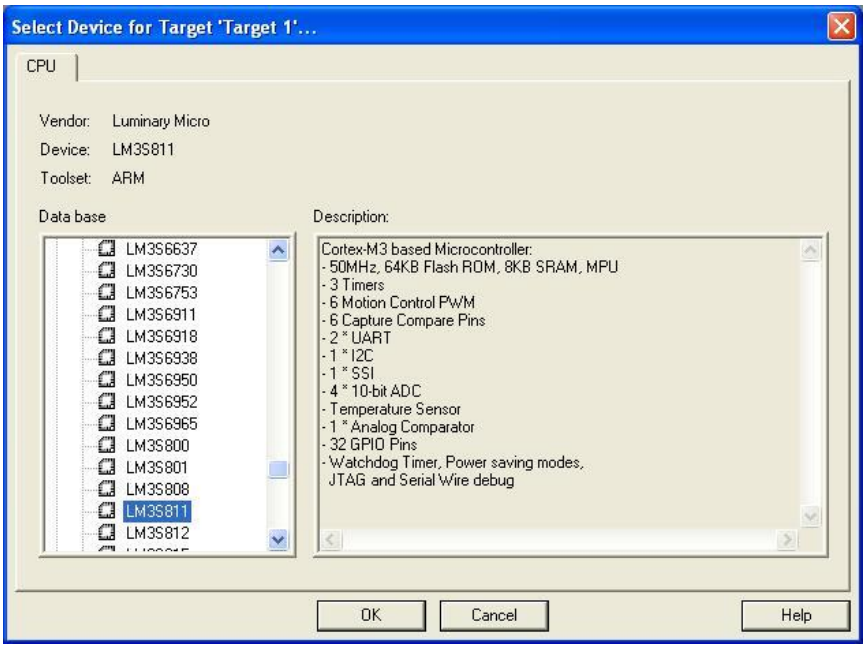


Fig 6.3: Select a device for the project

Click "OK" to continue.

5. Click "Yes" when asked to copy Luminary Startup Code to Project Folder and Add File to Project.

Fig 6.4: Selection window to add the startup code

6. Under the "File" menu, select "New…" to create a blank text space.
7. Under the "File" menu, select "Save As…" to save your new text space. Give it the extension ".c" to begin programming or ".txt" to provide documentation.
8. To the left of the screen, you will see a space that looks like this:

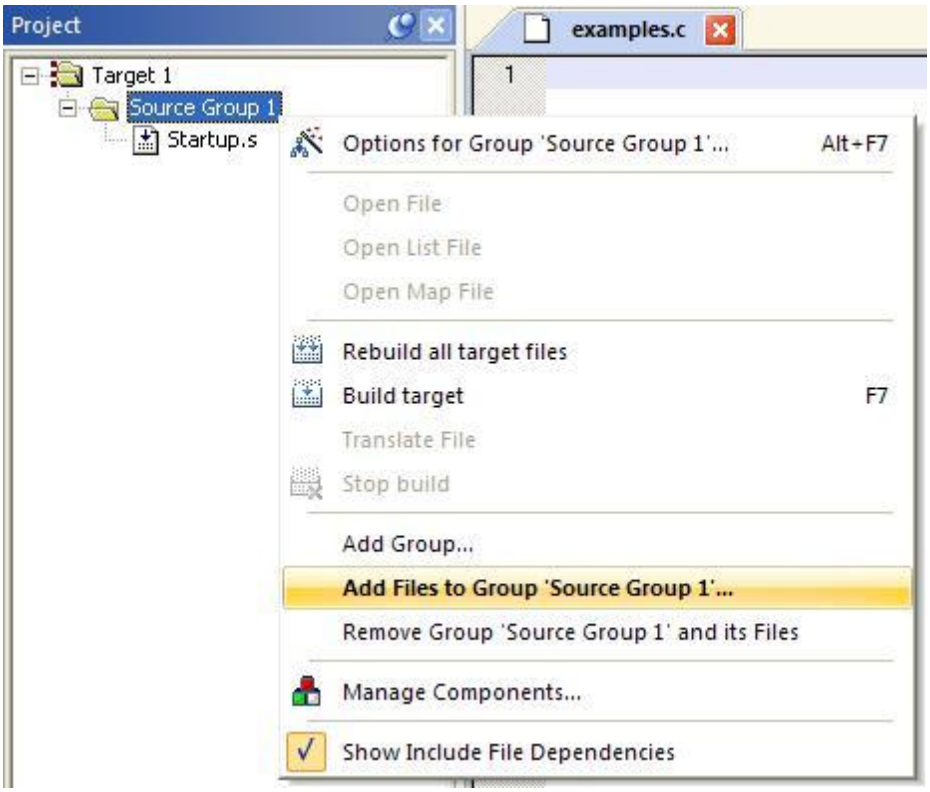


Fig 6.5: Shows how to add files to the source group

Right-click on "Source Group 1" and select "Add Files to Group 'Source group 1'…".

9. Select the file(s) that you saved in Step 5 and click "Add" for each one. Click "Close" when finished. If you want, you can re-name "Target 1" and "Source Group 1" by selecting them and pressing F2. You can also create additional sub-groups by right-clicking on "Target 1" and selecting "Add Group…". The new group can be populated with files and re-named in the same way as "Source Group 1". For example, you may want to add library files to your Project. Libraries give you easy access to common, useful functions without needing to write them yourself. Right-click on "Target 1", select "Add Group…", and give the new group a descriptive name such as "Library". Right-click on the "Library" group and select "Add Files to

Group ‘Library’…”. Navigate to the folder that contains library files, such as “…driverlib\rvmdk\driverlib.lib” and add them to this group. In ValvanoWare, the **driverlib.lib** in is in the **driverlib\rvmdk** directory

10. Under the “Flash” menu, select “Configure Flash Tools…”.
11. In the “Options” window that pops up, go to the “Utilities” sub-tab. Select the hardware that will be used to program the flash from the pull-down menu. For example, select “Stellaris ICDI” (In Circuit Debug Interface). Your selection should look something like this:

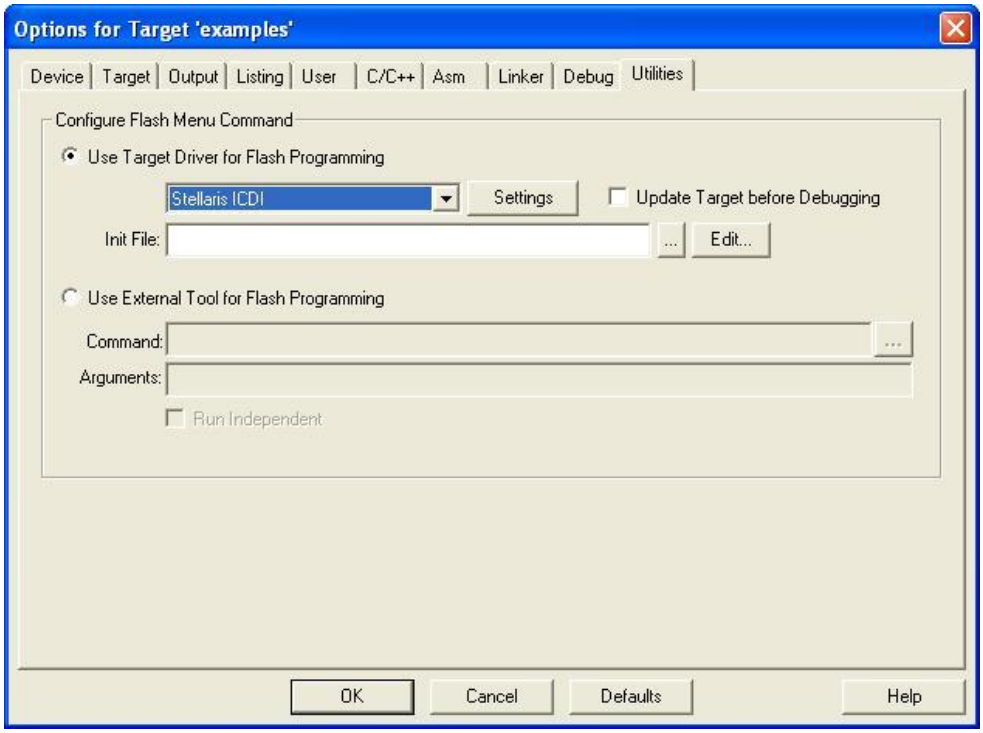


Fig 6.6: Device selection to create a target file

12. While still in the “Options” window, go to the “Target” sub- tab. It is important to check that the crystal frequency and memory page locations are correct for your device. Change the settings to this configuration: (The LM3S1968 board has an 8 MHz crystal)

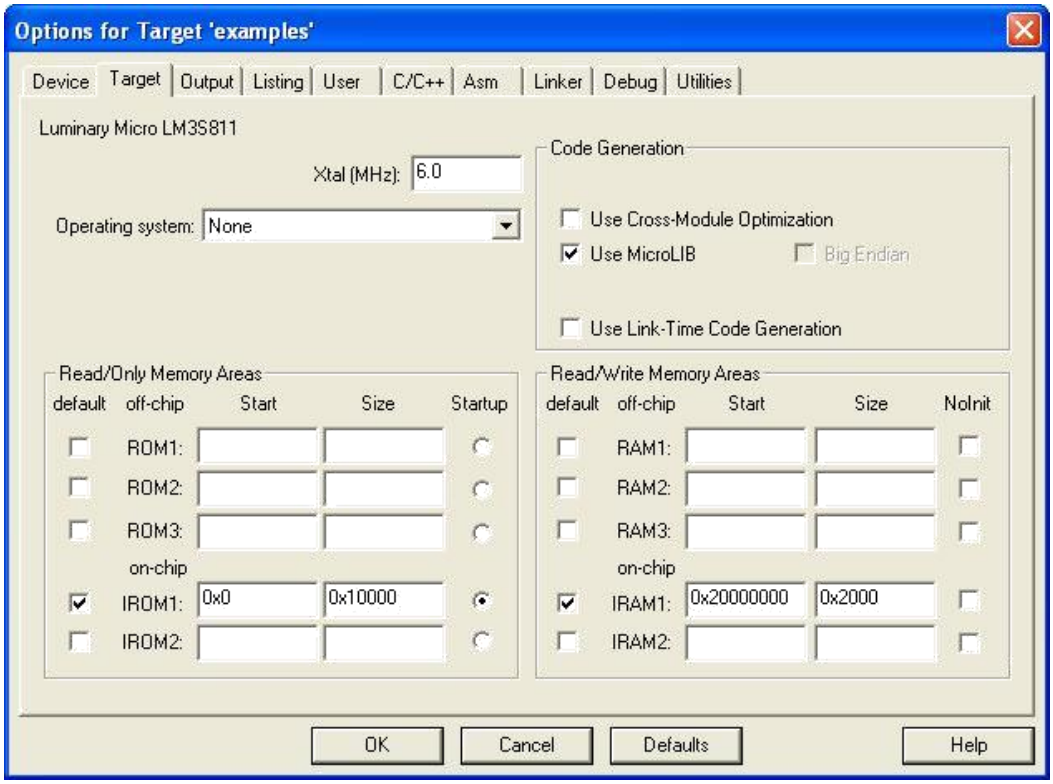


Fig 6.7: Set oscillator frequency

Click “OK” to continue. Setup should be complete, so you are now ready to program.

6.6 FLASH MAGIC

Flash Magic is an application developed by Embedded Systems Academy to allow you to easily access the features of a microcontroller device. With this program you can erase individual

blocks or the entire Flash memory of the microcontroller. This application is very useful for those who work in the electronics field. The main window of the program is composed of five sections where you can find the most common functions in order to program a microcontroller device. Using the "Communications" section you will be able to choose the way a specific device connects to your computer. Select the COM port to be used and the baud rate. It is recommended that you choose a low baud rate first and increase it afterwards. This way you will determine the highest speed with which your system works. In order to select which parts of the memory to erase, choose from the items in the "Erase" section. The third section is optional. It offers you the possibility to program a HEX file. In the next section you will be able to find different programming options, such as "verify after programming", "gen block checksums", "execute" and others. When you're done, click the Start button that can be found in the "Start" section. The program will start the device, and you will able to see the progress of the operations at the bottom of the main window.

Using Flash Magic, you are able to perform different operations to a microcontroller device, operations like erasing, programming and reading the flash memory, modifying the Boot Vector, performing a blank check on a section of the Flash memory and many others.

Flash Magic is the standard tool for programming NXP microcontrollers, and it's free! It has a straightforward and intuitive user interface and with it's five simple steps you can erase and program a device and setting key options. After programming is has an automatic verifying option.

Fig 6.8: Start up screen of the Flash Magic software

Features of software

- Read and write the Intel Hex file
- Read signature, lock and fuse bits
- Clear and Fill memory buffer
- Verify with memory buffer
- Reload current Hex file
- Display buffer checksum
- Program selected lock bits & fuses
- Auto detection of hardware

6.3.1 DUMPING A PROJECT IN FLASH MAGIC

1. Start up Flash Magic

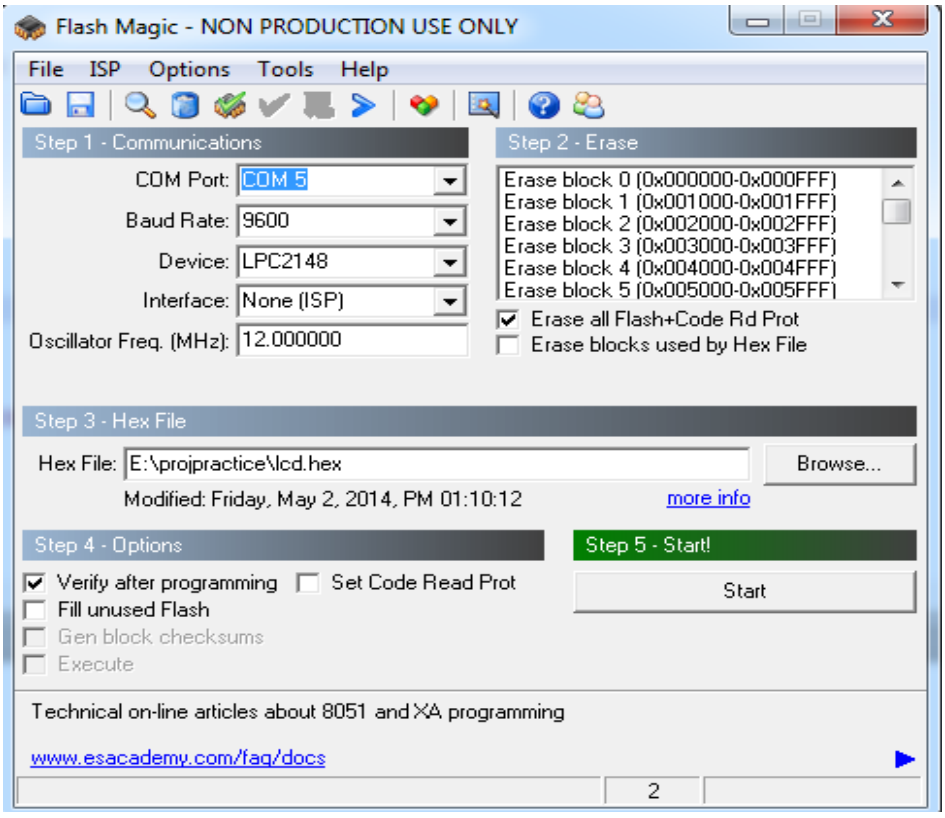


Fig 6.9: Main screen of the software

2. Select the COM port to which UART cable is connected (by default it is COM1)
3. Select Baud Rate as 9600 and also select the device we used (LPC2148)
4. Enable “Erase all Flash+Code Rd Prot” & “Verify after programming” buttons (so that all the flash & code memory of controller will be erased & code will be dumped and verified after programming)
5. Browse the required .hex (file which we are going to be dumped), For example: C:\Users\ashok\Documents\proj\voice-playback.hex
6. Before going to press “Start” button press & hold the reset button of controller board until programming is done
7. Clicking the Start button will result in all the selected operations in the main window taking place. They are:
    - Erasing Flash
    - Programming the Hex File
    - Verifying the Hex File
    - Filling Unused Flash
    - Generating Checksums
    - Programming the clocks bit
    - Programming the Security Bits
    - Executing the firmware

CONCLUSION AND FUTURE WORK

This paper has shown an extensive analysis on the feasibility of deploying smartphone-based TISs. We presented a localization algorithm, suitable for GPS location samples, and evaluated it through realistic simulations. Furthermore, leveraging state-of-the-art cryptographic and telecommunication schemes, we presented a comprehensive security and privacy-preserving architecture for smartphone-based TIS. Our results confirm it is feasible to build accurate and trustworthy smartphone-based TIS. Nevertheless, there are still challenges ahead: Security and privacy cannot, alone, incentivize uses to participate in large numbers. Toward this, it is interesting to provide fair and privacy-preserving incentive mechanisms.

REFERENCES

[1] S. Tao, V. Manolopoulos, S. Rodriguez, and A. Rusu, "Real-time urban traffic state estimation with A-GPS mobile phones as probes," *J. Transp. Technol.*, vol. 2, no. 1, pp. 22–31, Jan. 2011.

[2] V. Manolopoulos, P. Papadimitratos, T. Sha, and A. Rusu, "Securing smartphone based ITS," in *Proc. 11th Int. Conf. ITST*, 2011, pp. 201–206.

[3] V. Manolopoulos, S. Tao, A. Rusu, and P. Papadimitratos, "Smartphonebased traffic information system for sustainable cities," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 16, no. 4, pp. 30–31, Feb. 2013.

[4] Y. Wang, M. Papageorgiou, and A. Messmer, "Real-time freeway traffic state estimation based on extended Kalman filter: A case study," *Transp. Sci.*, vol. 41, no. 2, p. 167, May 2007.

[5] J. Guo, J. Xia, and B. Smith, "Kalman filter approach to speed estimation using single loop detector measurements under congested conditions," *J. Transp. Eng.*, vol. 135, no. 12, pp. 927–934, Dec. 2009.

[6] M. A. Ferman, D. E. Blumenfeld, and X. Dai, "An analytical evaluation of a real-time traffic information system using probe vehicles," *J. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 23–34, 2005.

[7] Y. Chen, L. Gao, Z. Li, and Y. Liu, "A new method for urban traffic state estimation based on vehicle tracking algorithm," in *Proc. ITSC*, 2007, pp. 1097–1101.

[8] R. Clayford and T. Johnson, "Operational parameters affecting use of anonymous cell phone tracking for generating traffic information," in *Proc. 82nd TRB Annu. Meet.*, 2003, pp. 1–20.

[9] R. L. Cheu, C. Xie, and D. Lee, "Probe vehicle population and sample size for arterial speed estimation," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 17, no. 1, pp. 53–60, Jan. 2002.

[10] M. A. Bacchus, B. Hellinga, and M. P. Izadpanah, "An opportunity assessment of wireless monitoring of network-wide road traffic conditions," Dept. Civil Eng., Univ. Waterloo, Waterloo, ON, Canada, 2007.

[11] "ICT facts and figures," Geneva, Switzerland, Feb. 2013. [Online]. Available: http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ ICTFactsFigures2013-e.pdf

[12] "Mobile future in focus,"Reston,VA, USA, Feb. 2012. [Online].Available: https://www.comscore.com/Insights/Presentations-and-Whitepapers/ 2012/2012-Mobile-Future-in-Focus

[13] Y. Yim, "The state of cellular probes," Inst. Transp. Studies, Univ. Calif., Berkeley, CA, USA, Jul. 2003, Research Reports.

[14] M. Fontaine, B. Smith, A. Hendricks, and W. Scherer, "Wireless location technology-based traffic monitoring: preliminary recommendations to transportation agencies based on synthesis of experience and simulation results," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1993, pp. 51–58,
2007.

[15] J. C. Herrera *et al.*, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment," *Transp. Res. C, Emerg. Technol.*, vol. 18, no. 4, pp. 568–583, Aug. 2010.

[16] B. Hellinga, "Reducing bias in probe-based arterial link travel time estimates," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 257–273, Aug. 2002.