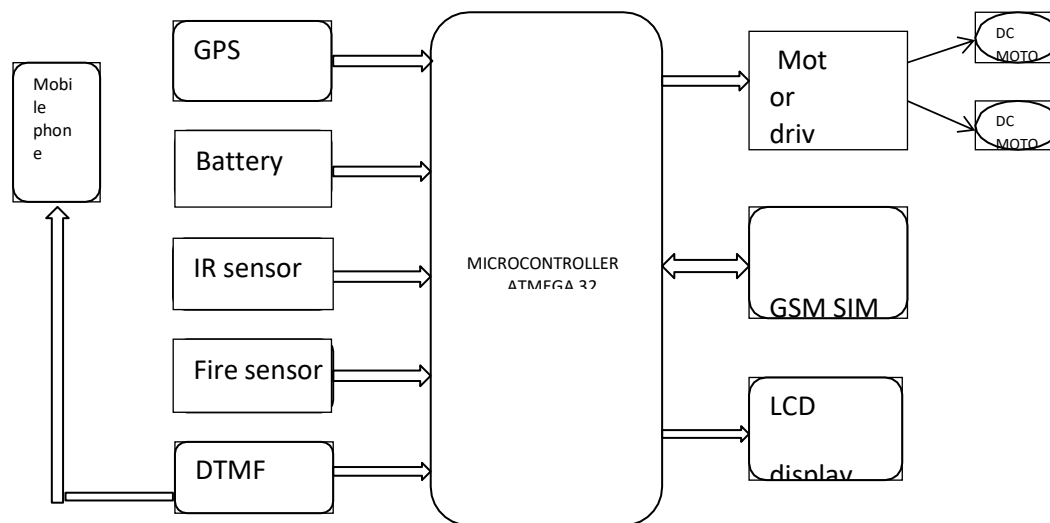# DESIGN OF A MULTITASKING ROBOT FOR ENHANCEDSURVEILLANCE AND MONITORING USING DTMF ANDGSM TECHNOLOGY

M. Aditi Reddy, L. Neha, K. Ankith, Koustubh Kulkarni

Student, Student, Student, Assistant Professor

NNRG

CHAPTER 2: EMBEDDED SYSTEMS

2.1 Embedded Systems:

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems are controlled by one or more main processing cores that are typically either microcontrollers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors. For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites. (Each radar probably includes one or more embedded systems of its own.)

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. For example, handheld computers share some elements with embedded systems such as the operating systems and microprocessors which power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems which don't expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions", and is thus appropriate to call "embedded". A modern example of embedded system is shown in fig: 2.1.
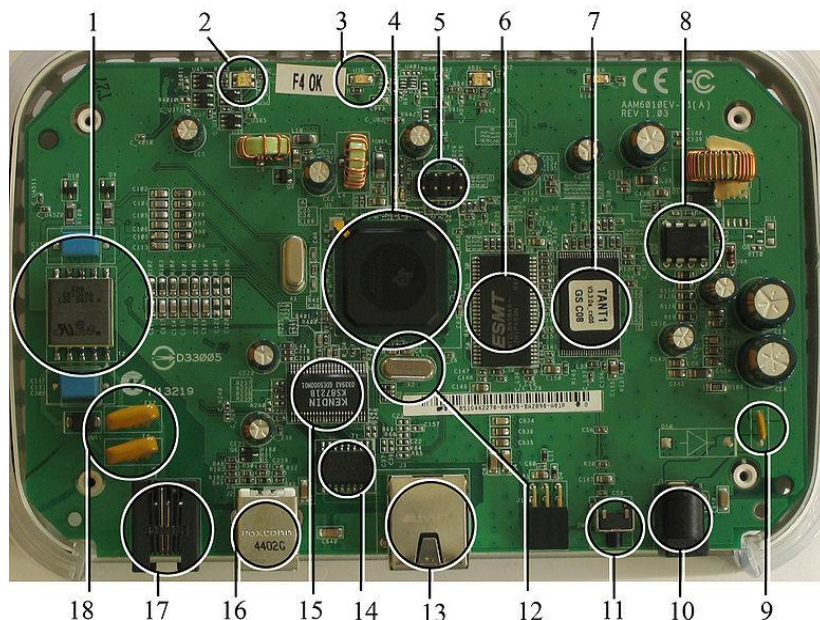

Fig 2.1:A modern example of embedded system

Labeled parts include microprocessor (4), RAM (6), flash memory (7).Embedded systems programming is not like normal PC programming. In many ways, programming for an embedded system is like programming PC 15 years ago. The hardware for the system is usually chosen to make the device as cheap as possible. Spending an extra dollar a unit in order to make things easier to program can cost millions. Hiring a programmer for an extra month is cheap in comparison. This means the programmer must make do with slow processors and low memory, while at the same time battling a need for efficiency not seen in most PC applications. Below is a list of issues specific to the embedded field.

2.1.1 History:

In the earliest years of computers in the 1930–40s, computers were sometimes dedicated to a single task, but were far too large and expensive for most kinds of tasks performed by embedded computers of today. Over time however, the concept of programmable controllers evolved from traditional electromechanical sequencers, via solid state devices, to the use of computer technology.

One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in 1961. It was built from transistor logic and had a hard disk for main memory. When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high-volume use of integrated circuits.

2.1.2 Tools:

Embedded development makes up a small fraction of total programming. There's also a large number of embedded architectures, unlike the PC world where 1 instruction set rules, and the Unix world where there's only 3 or 4 major ones. This means that the tools are more expensive. It also means that they're lowering featured, and less developed. On a major embedded project, at some point you will almost always find a compiler bug of some sort.

Debugging tools are another issue. Since you can't always run general programs on your embedded processor, you can't always run a debugger on it. This makes fixing your program difficult. Special hardware such as JTAG ports can overcome this issue in part. However, if you stop on a breakpoint when your system is controlling real world hardware (such as a motor), permanent equipment damage can occur. As a result, people doing embedded programming quickly become masters at using serial IO channels and error message style debugging.

2.1.3 Resources:

To save costs, embedded systems frequently have the cheapest processors that can do the job. This means your programs need to be written as efficiently as possible. When dealing with large data sets, issues like memory cache misses that never matter in PC programming can hurt you. Luckily, this won't happen too often- use reasonably efficient algorithms to start, and optimize only when necessary. Of course, normal profilers won't work well, due to the same reason debuggers don't work well.

Memory is also an issue. For the same cost savings reasons, embedded systems usually have the least memory they can get away with. That means their algorithms must be memory efficient (unlike in PC programs, you will frequently sacrifice processor time for memory, rather than the reverse). It also means you can't afford to leak memory. Embedded applications generally use deterministic memory techniques and avoid the default "new" and "malloc" functions, so that leaks can be found and eliminated more easily. Other resources programmers expect may not even exist. For example, most embedded processors do not have hardware FPUs (Floating-Point Processing Unit). These resources either need to be emulated in software, or avoided altogether.

2.1.4 Real Time Issues:

Embedded systems frequently control hardware, and must be able to respond to them in real time. Failure to do so could cause inaccuracy in measurements, or even damage hardware such as motors. This is made even more difficult by the lack of resources available. Almost all embedded systems need to be able to prioritize some tasks over others, and to be able to put off/skip low priority tasks such as UI in favor of high priority tasks like hardware control.

2.2 Need For Embedded Systems:

The uses of embedded systems are virtually limitless, because every day new products are introduced to the market that utilizes embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and FPGA chips have become much cheaper. So when implementing a new form of control, it's wiser to just buy the generic chip and write your own custom software for it. Producing a custom-made chip to handle a particular task or set of tasks costs far more time and money. Many embedded computers even come with extensive libraries, so that "writing your own software" becomes a very trivial task indeed. From an implementation viewpoint, there is a major difference between a computer and an embedded system. Embedded systems are often required to provide Real-Time response. The main elements that make embedded systems unique are its reliability and ease in debugging.

2.2.1 Debugging:

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

- Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g. Forth and Basic)

- External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi core systems.

- An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or Nexus interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.

- An in-circuit emulator replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.

- A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.

- Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as assembly code or source-code.

Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software(and microprocessor) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, co-processor). An increasing number of embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronization of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the processor cores, which requires very low-level debugging, at signal/bus level, with a logic analyzer, for instance.

2.2.2 Reliability:

Embedded systems often reside in machines that are expected to run continuously for years without errors and in some cases recover by them selves if an error occurs. Therefore the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

- The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.

- The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backup s are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals, engines on single-engine aircraft.
- The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.

A variety of techniques are used, sometimes in combination, to recover from errors—both software bugs such as memory leaks, and also soft errors in the hardware:

- Watchdog timer that resets the computer unless the software periodically notifies the watchdog
- Subsystems with redundant spares that can be switched over to
- software "limp modes" that provide partial function
- Designing with a Trusted Computing Base (TCB) architecture[6] ensures a highly secure & reliable system environment
- An Embedded Hypervisor is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software. This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
- Immunity Aware Programming

2.3 Explanation of Embedded Systems:

2.3.1 Software Architecture:

There are several different types of software architecture in common use.

- Simple Control Loop:

In this design, the software simply has a loop. The loop calls subroutines, each of which manages a part of the hardware or software.

- Interrupt Controlled System:

Some embedded systems are predominantly interrupt controlled. This means that tasks performed by the system are triggered by different kinds of events. An interrupt could be generated for example by a timer in a predefined frequency, or by a serial port controller receiving a byte. These kinds of systems are used if event handlers need low latency and the event handlers are short and simple.

Usually these kinds of systems run a simple task in a main loop also, but this task is not very sensitive to unexpected delays. Sometimes the interrupt handler will add longer tasks to a queue structure. Later, after the interrupt handler has finished, these tasks are executed by the main loop. This method brings the system close to a multitasking kernel with discrete processes.

- Cooperative Multitasking:

A non-preemptive multitasking system is very similar to the simple control loop scheme, except that the loop is hidden in an API. The programmer defines a series of tasks, and each task gets its own environment to "run" in. When a task is idle, it calls an idle routine, usually called "pause", "wait", "yield", "nop" (stands for no operation), etc. The advantages and disadvantages are very similar to the control loop, except that adding new software is easier, by simply writing a new task, or adding to the queue-interpreter.

- Primitive Multitasking:

In this type of system, a low-level piece of code switches between tasks or threads based on a timer (connected to an interrupt). This is the level at which the system is generally considered to have an "operating system" kernel. Depending on how much functionality is required, it introduces more or less of the complexities of managing multiple tasks running conceptually in parallel.

As any code can potentially damage the data of another task (except in larger systems using an MMU) programs must be carefully designed and tested, and access to shared data must be controlled by some synchronization strategy, such as message queues, semaphores or a non-blocking synchronization scheme.

Because of these complexities, it is common for organizations to buy a real-time operating system, allowing the application programmers to concentrate on device functionality rather than operating system services, at least for large systems; smaller systems often cannot afford the overhead associated with a generic real time system, due to limitations regarding memory size, performance, and/or battery life.

- Microkernels And Exokernels:

A microkernel is a logical step up from a real-time OS. The usual arrangement is that the operating system kernel allocates memory and switches the CPU to different threads of execution. User mode processes implement major functions such as file systems, network interfaces, etc.

In general, microkernels succeed when the task switching and intertask communication is fast, and fail when they are slow. Exokernels communicate efficiently by normal subroutine calls. The hardware and all the software in the system are available to, and extensible by application programmers. Based on performance, functionality, requirement the embedded systems are divided into three categories:

2.3.2 Stand Alone Embedded System:

These systems takes the input in the form of electrical signals from transducers or commands from human beings such as pressing of a button etc.., process them and produces desired output. This entire process of taking input, processing it and giving output is done in standalone mode. Such embedded systems comes under stand alone embedded systems

Eg: microwave oven, air conditioner etc..

2.3.3 Real-time embedded systems:

Embedded systems which are used to perform a specific task or operation in a specific time period those systems are called as real-time embedded systems. There are two types of real-time embedded systems.

- Hard Real-time embedded systems:

These embedded systems follow an absolute dead line time period i.e.., if the tasking is not done in a particular time period then there is a cause of damage to the entire equipment.

Eg: consider a system in which we have to open a valve within 30 milliseconds. If this valve is not opened in 30 ms this may cause damage to the entire equipment. So in such cases we use embedded systems for doing automatic operations.

- Soft Real Time embedded systems:

Eg: Consider a TV remote control system, if the remote control takes a few milliseconds delay it will not cause damage either to the TV or to the remote control. These systems which will not cause damage when they are not operated at considerable time period those systems comes under soft real-time embedded systems.

2.3.4 Network communication embedded systems:

A wide range network interfacing communication is provided by using embedded systems.

Eg:

- Consider a web camera that is connected to the computer with internet can be used to spread communication like sending pictures, images, videos etc.., to another computer with internet connection throughout anywhere in the world.

- Consider a web camera that is connected at the door lock.

Whenever a person comes near the door, it captures the image of a person and sends to the desktop of your computer which is connected to internet. This gives an alerting message with image on to the desktop of your computer, and then you can open the door lock just by clicking the mouse. Fig: 2.2 show the network communications in embedded systems.

Fig 2.2: Network communication embedded systems

2.3.5 Different types of processing units:

The central processing unit (c.p.u) can be any one of the following microprocessor, microcontroller, digital signal processing.

- Among these Microcontroller is of low cost processor and one of the main advantage of microcontrollers is, the components such as memory, serial communication interfaces, analog to digital converters etc.., all these are built on a single chip. The numbers of external components that are connected to it are very less according to the application.

- Microprocessors are more powerful than microcontrollers. They are used in major applications with a number of tasking requirements. But the microprocessor requires many external components like memory, serial communication, hard disk, input output ports etc.., so the power consumption is also very high when compared to microcontrollers.

- Digital signal processing is used mainly for the applications that particularly involved with processing of signals

2.4 APPLICATIONS OF EMBEDDED SYSTEMS:

2.4.1 Consumer applications:

At home we use a number of embedded systems which include microwave oven, remote control, vcd players, dvd players, camera etc….

Fig2.3: Automatic coffee makes equipment

2.4.2 Office automation:

We use systems like fax machine, modem, printer etc…

Fig2.4: Fax machine   **Fig2.5: Printing machine**

2.4.3. Industrial automation:

Today a lot of industries are using embedded systems for process control. In industries we design the embedded systems to perform a specific operation like monitoring temperature, pressure, humidity ,voltage, current etc.., and basing on these monitored levels we do control other devices, we can send information to a centralized monitoring station.

Fig2.6: Robot

In critical industries where human presence is avoided there we can use robots which are programmed to do a specific operation.

2.4.5 Computer networking:

Embedded systems are used as bridges routers etc..

Fig2.7: Computer networking

2.4.6 Tele communications:

Cell phones, web cameras etc.

Fig2.8: Cell Phone

Fig2.9: Web camera

ARDUINO NANO (Micro controller)

Introduction to the Arduino NANO Board

The Arduino Nano, as the name suggests is a compact, complete and bread-board friendly microcontroller board. The Nano board weighs around 7 grams with dimensions of 4.5 cms to 1.8 cms (L to B). This article discusses about the technical specs most importantly the pinout and functions of each and every pin in the Arduino Nano board.

Arduino Nano has similar functionalities as Arduino Duemilanove but with a different package. The Nano is inbuilt with the ATmega328P microcontroller, same as the Arduino UNO. The main difference between them is that the UNO board is presented in PDIP (Plastic Dual-In-line Package) form with 30 pins and Nano is available in TQFP (plastic quad flat pack) with 32 pins. The extra 2 pins of Arduino Nano serve for the ADC functionalities, while UNO has 6 ADC ports but Nano has 8 ADC ports. The Nano board doesn't have a DC power jack as other Arduino boards, but instead has a mini-USB port. This port is used for both programming and serial monitoring. The fascinating feature in Nano is that it will choose the strongest power source with its potential difference, and the power source selecting jumper is invalid.
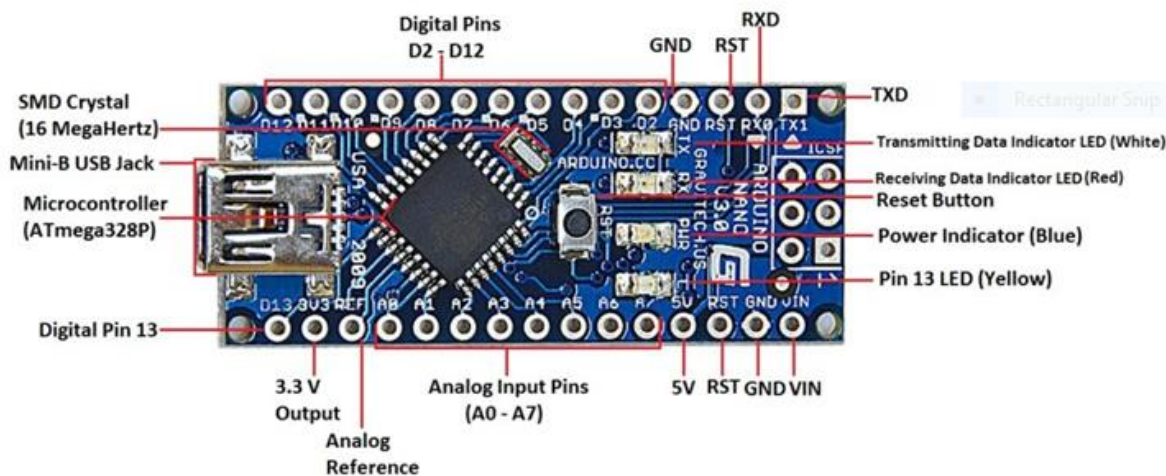
Figure 3.3.2 Arduino nano Board

Arduino Nano – Specification

Arduino Nano	Specifications
Analog I/O Pins	8
Architecture	AVR
Clock Speed	16 MHz
DC Current per I/O Pins	40 milliAmps
Digital I/O Pins	22
EEPROM	1 KB
Flash Memory	32 KB of which 2 KB used by Bootloader
Input Voltage	(7-12) Volts
Microcontroller	ATmega328P
Operating Voltage	5 Volts
PCB Size	18 x 45 mm
Power Consumption	19 milliAmps
PWM Output	6
SRAM	2KB
Weight	7 gms

3.2 Pin diagram

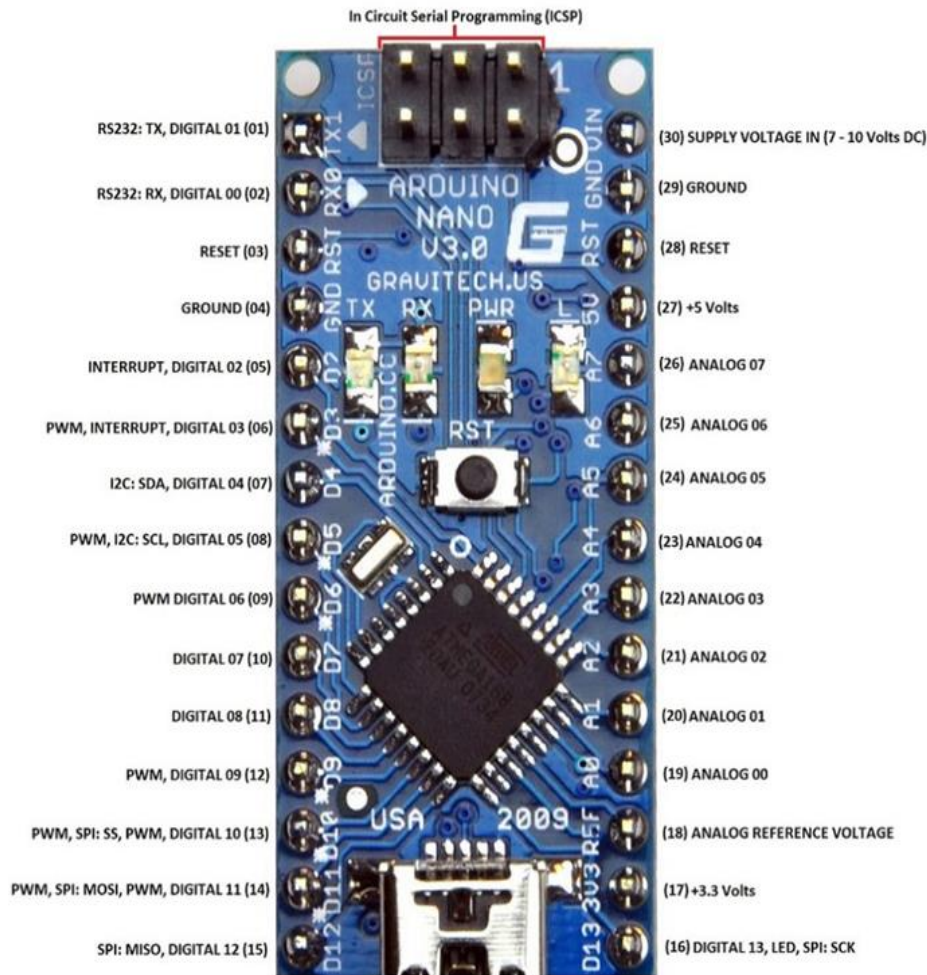Figure 3.3.3 Pin Configuration of Atmega328

Pin Description

Arduino Nan0 – Pin Description

Pins 1 to 30

Arduino Nano Pin	Pin Name	Type	Function		
1	D1/TX	I/O	Digital	I/O	Pin
			Serial TX Pin		
2	D0/RX	I/O	Digital	I/O	Pin
			Serial RX Pin		

Arduino Nano Pin	Pin Name	Type	Function
3	RESET	Input	Reset (Active Low)
4	GND	Power	Supply Ground
5	D2	I/O	Digital I/O Pin
6	D3	I/O	Digital I/O Pin
7	D4	I/O	Digital I/O Pin
8	D5	I/O	Digital I/O Pin
9	D6	I/O	Digital I/O Pin
10	D7	I/O	Digital I/O Pin
11	D8	I/O	Digital I/O Pin
12	D9	I/O	Digital I/O Pin
13	D10	I/O	Digital I/O Pin
14	D11	I/O	Digital I/O Pin
15	D12	I/O	Digital I/O Pin
16	D13	I/O	Digital I/O Pin

Arduino Nano Pin	Pin Name	Type	Function
17	3V3	Output	+3.3V Output (from FTDI)
18	AREF	Input	ADC reference
19	A0	Input	Analog Input Channel 0
20	A1	Input	Analog Input Channel 1
21	A2	Input	Analog Input Channel 2
22	A3	Input	Analog Input Channel 3
23	A4	Input	Analog Input Channel 4
24	A5	Input	Analog Input Channel 5
25	A6	Input	Analog Input Channel 6
26	A7	Input	Analog Input Channel 7
27	+5V	Output or Input	+5V Output (From On-board Regulator) or +5V (Input from External Power Supply
28	RESET	Input	Reset (Active Low)
29	GND	Power	Supply Ground

Arduino Nano Pin	Pin Name	Type	Function
30	VIN	Power	Supply voltage

ICSP Pins

Arduino Nano ICSP Pin Name	Type	Function
MISO	Input or Output	Master In Slave Out
Vcc	Output	Supply Voltage
SCK	Output	Clock from Master to Slave
MOSI	Output or Input	Master Out Slave In
RST	Input	Reset (Active Low)
GND	Power	Supply Ground

Arduino Nano Digital Pins

Pins - 1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16

As mentioned earlier, Arduino Nano has 14 digital I/O pins that can be used either as digital input or output. The pins work with 5V voltage as maximum, i.e., digital high is 5V and digital low is 0V. Each pin can provide or receive a current of 20mA, and has a pull-up resistance of about 20-50k ohms. Each of the 14 digital pins on the Nano pinout can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions.

Other than the digital input and output functions, the digital pins have some additional functionality as well.

Serial Communication Pins

Pins - 1, 2

1 - RX and 2 - TX

These two pins RX- receive and TX- transmit are used for TTL serial data communication. The pins RX and TX are connected to the corresponding pins of the USB-to-TTL Serial chip.

PWM Pins

Pins - 6, 8, 9, 12, 13, and 14

Each of these digital pins provide a Pulse Width Modulation signal of 8-bit resolution. The PWM signal can be generated using analogWrite () function.

External Interrupts

Pins - 5, 6

When we need to provide an external interrupt to other processor or controller we can make use of these pins. These pins can be used to enable interrupts INT0 and INT1 respectively by using the attachInterrupt () function. These pins can be used to trigger three types of interrupts such as interrupt on a low value, a rising or falling edge interrupt and a change in value interrupt.

SPI Pins

Pins - 13, 14, 15, and 16

When you don't want the data to be transmitted asynchronously you can use these Serial Peripheral Interface pins. These pins support synchronous communication with SCK as the synchronizing clock. Even though the hardware has this feature, the Arduino software doesn't have this by default. So you have to include a library called SPI Library for using this feature.

LED

Pin - 16

If you remember your first Arduino code, blinking LED, then you'll definitely came across this Pin16. The pin 16 is being connected to the blinking LED on the board.

Arduino Nano Analog Pins

Pins - 18, 19, 20, 21, 22, 23, 24, 25, and 26

As mentioned earlier UNO got 6 analog input pins but Arduino Nano has 8 analog inputs (19 to 26), marked A0 through A7. This means you can connect *8 channel analog sensor inputs for processing. Each of these analog pins has a inbuilt ADC of resolution of 1024 bits (so it will give 1024 values). By default, the pins are measured from ground to 5V. If you want the reference voltage to be 0V to 3.3V, we can give 3.3V to AREF pin (18th Pin) by using the analogReference () function.

Similar to digital pins in Nano, analog pins also got some other functions as well.

I2C

Pins 23, 24 as A4 and A5

Since SPI communication also has its disadvantages such as 4 essential pins and limited within a device. For long distance communication we use the I2C protocol. I2C supports multi master and multi slave with only two wires. One for clock (SCL) and another for data (SDA). For using this I2C feature we need to import a library called Wire library.
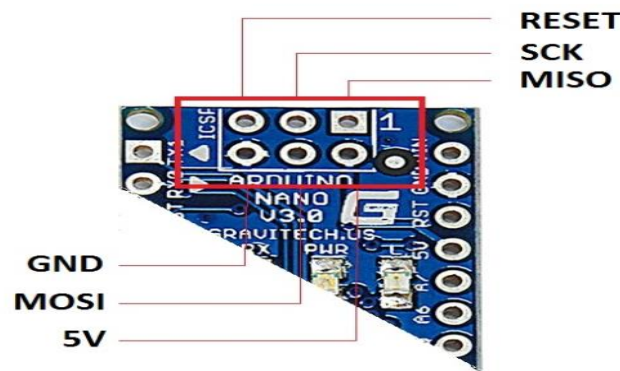
AREF

Pin 18

As mentioned already the AREF- Analog Reference pin is used as a reference voltage for analog input for the ADC conversion.

Reset

Pin 28

Reset pins in Arduino are active LOW pins which means if we make this pin value as LOW i.e., 0v, it will reset the controller. Usually used to be connected with switches to use as reset button.

ICSP

Arduino Nano ISCP

ICSP stands for *In Circuit Serial Programming*, which represents one of the several methods available for programming Arduino boards. Ordinarily, an Arduino bootloader program is used to program an Arduino board, but if the bootloader is missing or damaged, ICSP can be used instead. ICSP can be used to restore a missing or damaged bootloader.

Each ICSP pin usually is cross-connected to another Arduino pin with the same name or function. For example, MISO on Nano's ICSP header is connected to MISO / digital pin 12 (Pin 15); MOSI on the ISCP header is connected to MOSI / digital pin 11 (Pin 16); and so forth. Note, MISO, MOSI, and SCK pins taken together make up most of an SPI interface.

We can use one Arduino to program another Arduino using this ICSP.

Arduino as ISP	ATMega328
Vcc/5V	Vcc
GND	GND
MOSI/D11	D11

Arduino as ISP	ATMega328
MISO/D12	D12
SCK/D13	D13
D10	Reset

RESET

Pins 3, 28 and 5 in ICSP

Power

Pins 4, 17, 27, 28, 30 and 2 & 6 in ICSP

FEATURES

- ➢ 1.8-5.5V operating range
- ➢ Up to 20MHz
- ➢ Part: ATMEGA328P-AU
- ➢ 32kB Flash program memory
- ➢ 1kB EEPROM
- ➢ 2kB Internal SRAM
- ➢ 2 8-bit Timer/Counters
- ➢ 16-bit Timer/Counter
- ➢ RTC with separate oscillator
- ➢ Master/Slave SPI interface
- ➢ 2-wire (I2C) interface
- ➢ Watchdog timer
- ➢ 23 IO lines
- ➢ Data retention: 20 years at 85C/ 100 years at 25C
- ➢ Digital I/O Pins are 14 (out of which 6 provide PWM output)
- ➢ Analog Input Pins are 6.

➢ DC Current per I/O is 40 mA

➢ DC Current for 3.3V Pin is 50mA

3.3.3 AVR CPU Core

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.
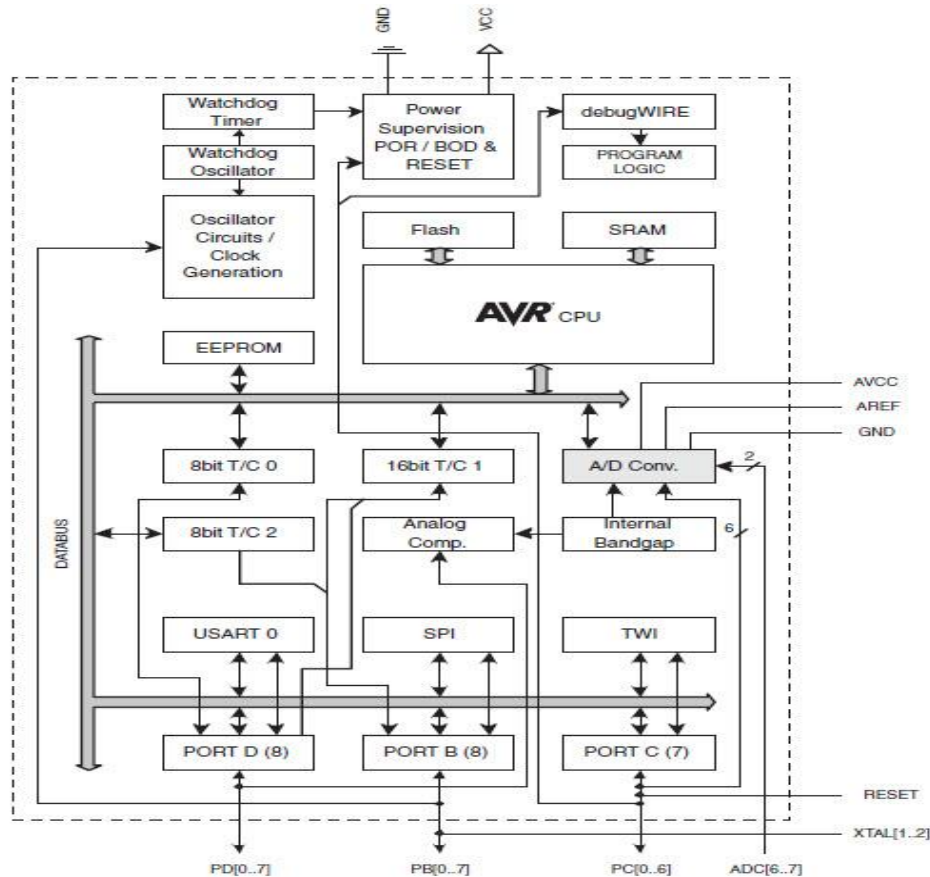

Figure 3.3.3 AVR Block Diagram

Overview

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle   interrupts.
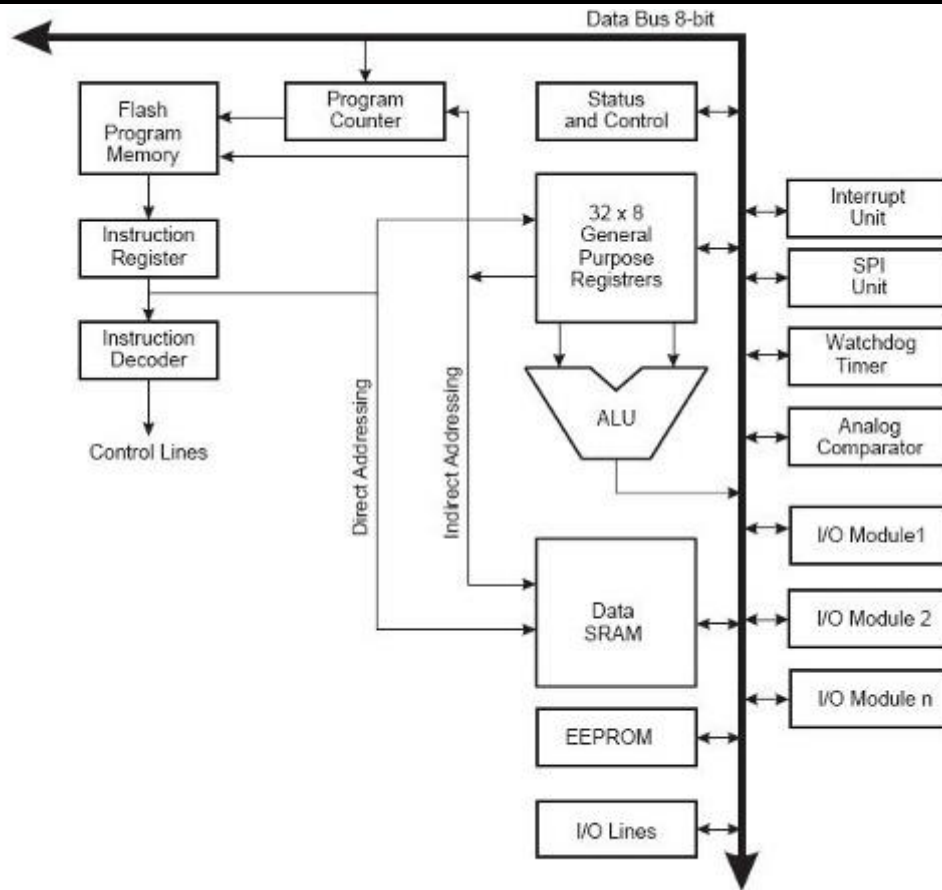
Figure 3.3.4 AVR core architecture

In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory. The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File– in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section. The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation. Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

3.2.4 Arduino with ATmega328

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to versionR2) programmed as a USB-to-serial converter.

➢ Pin out: Added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino. Due that operates with 3.3V. The second one is a not connected pin that is reserved for future purposes.

➢ Stronger RESET circuit.

➢ Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Arduino Characteristics

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

➢ **VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

➢ **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- ➢ **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- ➢ **GND.** Ground pins.
- ➢ **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory:

The ATmega328 has 32 KB (with 0.5 KB used for the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Serial Communication:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

3.3 REGULATED POWER SUPPLY:

3.3.1 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.

3.3.2 Block Diagram:

Regulated Power supply

AC 230V   **Transformer** → **Rectifier** → **Filter** → **Regulator**   DC 5V
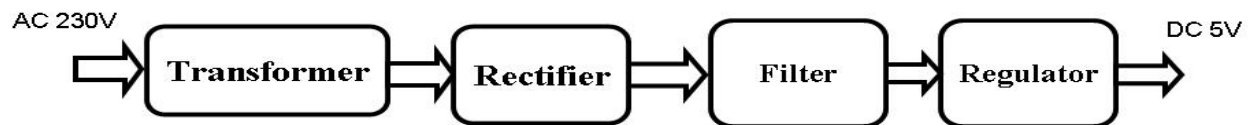
Fig 3.3.2 Regulated Power Supply

The basic circuit diagram of a regulated power supply (DC O/P) with led connected as load is shown in fig: 3.3.3.
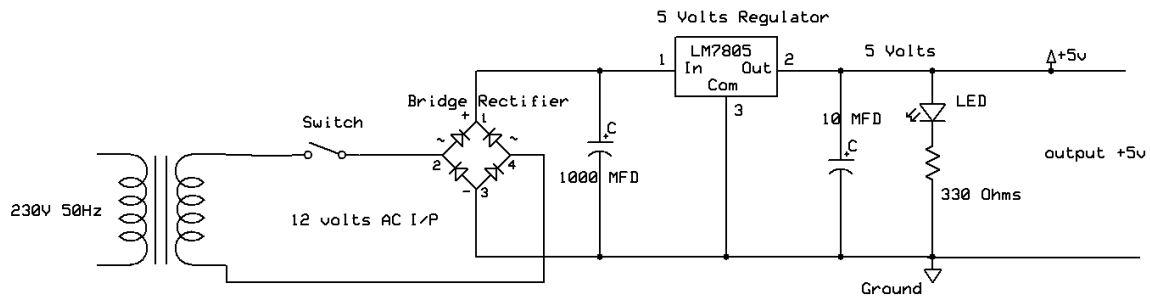
REGULATED POWER SUPPLY

Fig 3.3.3 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:

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

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.
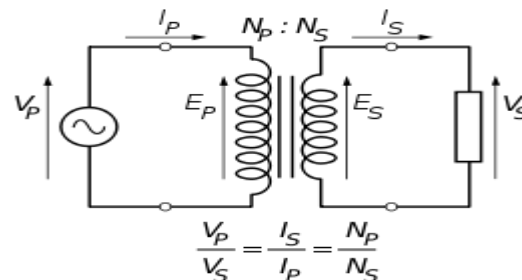

Fig 3.3.4: Step-Down Transformer

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: 3.3.5 shows Hi-Watt 9V battery

Fig 3.3.5: 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: 3.3.6 shows pencil battery of 1.5V.

Fig 3.3.6: Pencil Battery of 1.5V

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 fig: 3.3.7, 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.
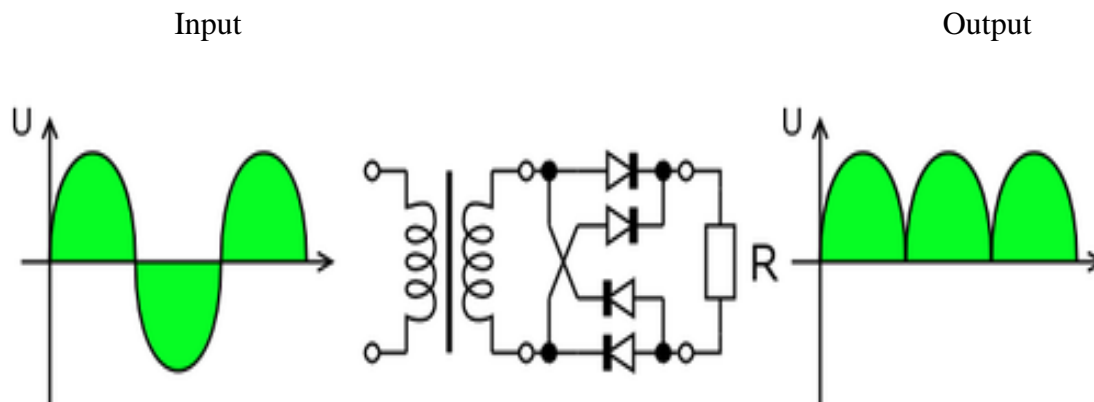
Input   Output

Fig 3.3.7: 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. The picture of DB 107 is shown in fig: 3.3.8.

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 3.3.8: DB107

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.

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. The construction of capacitor and an electrolytic capacitor are shown in figures 3.3.9 and 3.3.10 respectively.
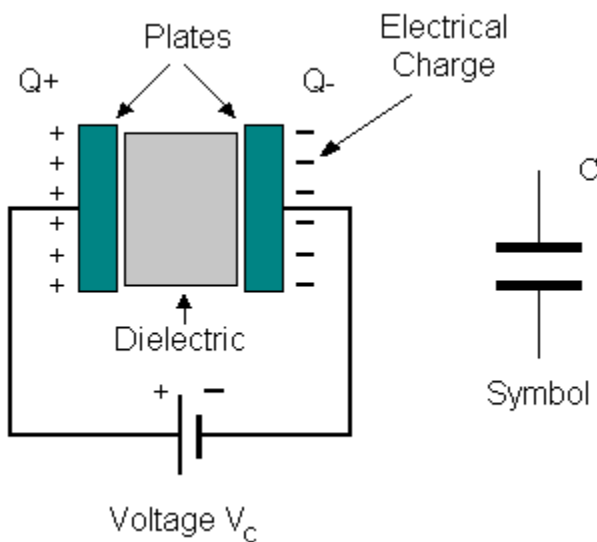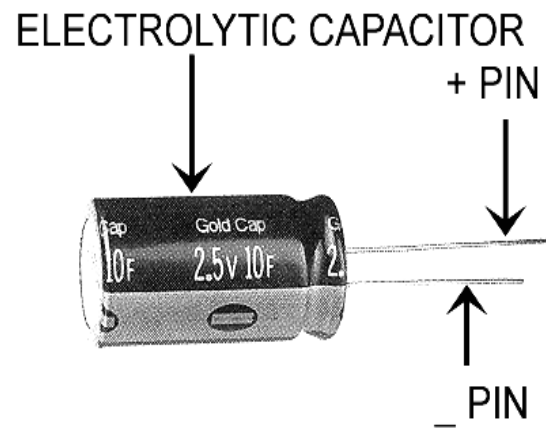

Fig 3.3.9:Construction Of a Capacitor   Fig 3.3.10:Electrolytic Capaticor

Units of Capacitance:

Microfarad (μF) $1\mu F = 1/1,000,000 = 0.000001 = 10^{-6}$ 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.
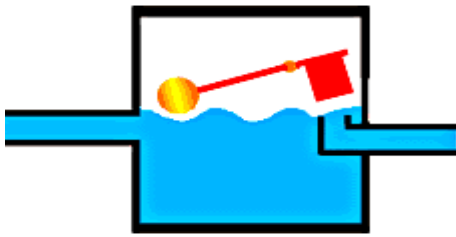
If the current flow is alternating between zero and a maximum, our "storage tank" capacitor will allow the current waves to pass through.
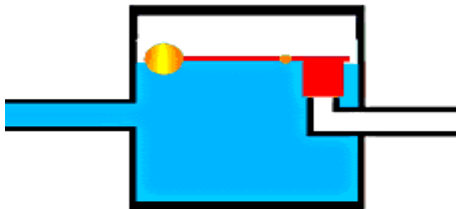
However, if there is a steady current, only the initial short burst will flow until the "floating ball valve" closes and stops further flow.

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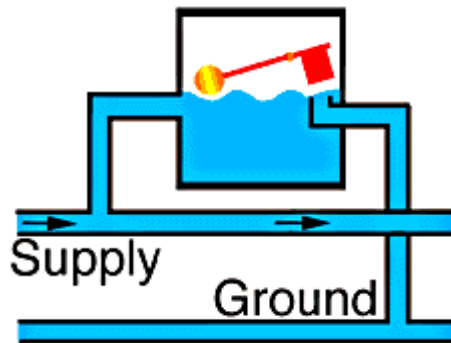
Where a capacitor is used to decouple a circuit, the effect is to "smooth out ripples". Any ripples, waves or pulses of current are passed to ground while d.c. Flows smoothly.

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: 3.3.11 shows voltage regulator.
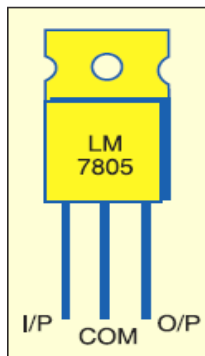
Fig 3.3.11: Voltage Regulator

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:

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

Resistor color Coding

1'st digit
2'nd digit
number of zeros
tolerance
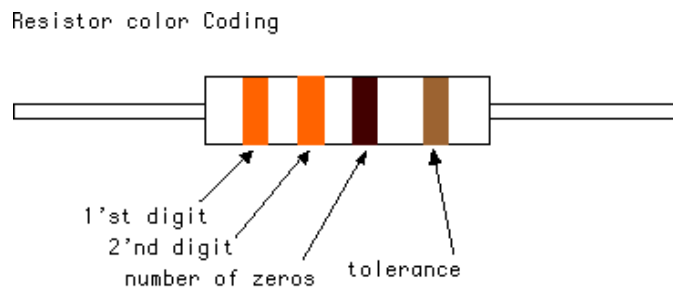
Digit	color		Tolerance	color
0	Black		20%	nothing
1	Brown		10%	Silver
2	Red		5%	Gold
3	Orange		2%	Red
4	Yellow		1%	Brown
5	Green			
6	Blue			
7	Violet			
8	Grey			
9	White			

Fig 3.3.12: Resistor   **Fig 3.3.13: Color Bands In Resistor**

3.4. LED:

A light-emitting diode (LED) is a semiconductor light source. LED's are used as indicator lamps in many devices, and are increasingly used for lighting. Introduced as a practical electronic component in 1962, early LED's 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 in figures 3.4.1 and 3.4.2 respectively.

Fig 3.4.1: Inside a LED
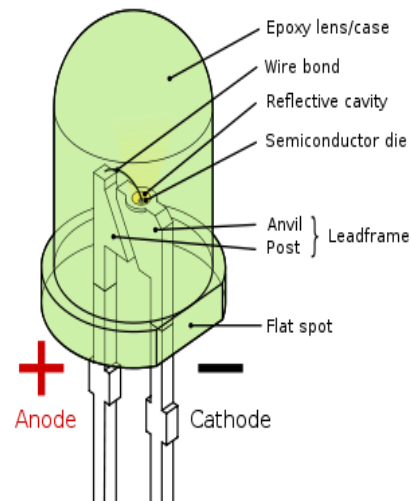
Fig 3.4.2: Parts of a LED

Working:

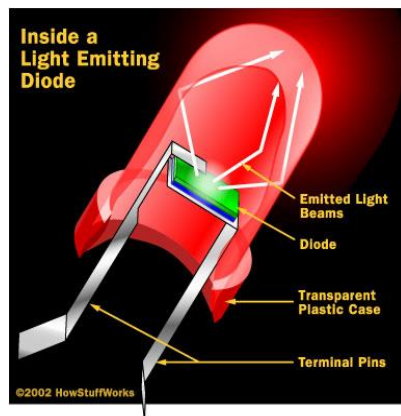The structure of the LED light is completely different than that of the light bulb. Amazingly, the LED has a simple and strong structure. The light-emitting semiconductor material is what determines the LED's color. The LED is based on the semiconductor diode.

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. LED's 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 LED's 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: 3.4.3.
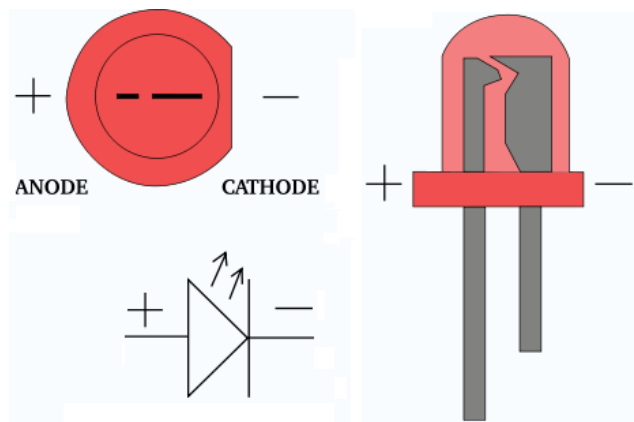
Fig 3.4.3: Electrical Symbol & Polarities of LED

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.

LCD MODULE

To display interactive messages we are using LCD Module. We examine an intelligent LCD display of two lines,16 characters per line that is interfaced to the controllers. The protocol (handshaking) for the display is as shown. Whereas D0 to D7th bit is the Data lines, RS, RW and EN pins are the control pins and remaining pins are +5V, -5V and GND to provide supply. Where RS is the Register Select, RW is the Read Write and EN is the Enable pin.

The display contains two internal byte-wide registers, one for commands (RS=0) and the second for characters to be displayed (RS=1). It also contains a user-programmed RAM area (the character RAM) that can be programmed to generate any desired character that can be formed using a dot matrix. To distinguish between these two data areas, the hex command byte 80 will be used to signify that the display RAM address 00h will be chosen.Port1 is used to furnish the command or data type, and ports 3.2 to3.4 furnish register select and read/write levels.

The display takes varying amounts of time to accomplish the functions as listed. LCD bit 7 is monitored for logic high (busy) to ensure the display is overwritten.

Liquid Crystal Display also called as LCD is very helpful in providing user interface as well as for debugging purpose. The most common type of LCD controller is HITACHI 44780 which provides a simple interface between the controller & an LCD. These LCD's are very simple to interface with the controller as well as are cost effective.

2x16 Line Alphanumeric LCD Display
The most commonly used *ALPHANUMERIC* displays are *1x16* (Single Line & 16 characters), *2x16* (Double Line & 16 character per line) & *4x20* (four lines & Twenty characters per line).

The LCD requires 3 control lines (RS, R/W & EN) & 8 (or 4) data lines. The number on data lines depends on the mode of operation. If operated in 8-bit mode then 8 data lines + 3 control lines i.e. total 11 lines are required. And if operated in 4-bit mode then 4 data lines + 3 control lines i.e. 7 lines are required. How do we decide which mode to use? It's simple if you have sufficient data lines you can go for 8 bit mode & if there is a time constrain i.e. display should be faster then we have to use 8-bit mode because basically 4-bit mode takes twice as more time as compared to 8-bit mode.

Pin	Symbol	Function
1	Vss	Ground
2	Vdd	Supply Voltage
3	Vo	Contrast Setting
4	RS	Register Select
5	R/W	Read/Write Select
6	En	Chip Enable Signal
7-14	DB0-DB7	Data Lines
15	A/Vee	Gnd for the backlight
16	K	Vcc for backlight

When *RS* is low (0), the data is to be treated as a command. When RS is high (1), the data being sent is considered as text data which should be displayed on the screen.

When *R/W* is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively reading from the LCD. Most of the times there is no need to read from the LCD so this line can directly be connected to Gnd thus saving one controller line.

The *ENABLE* pin is used to latch the data present on the data pins. A HIGH - LOW signal is required to latch the data. The LCD interprets and executes our command at the instant the EN line is brought low. If you never bring EN low, your instruction will never be executed.
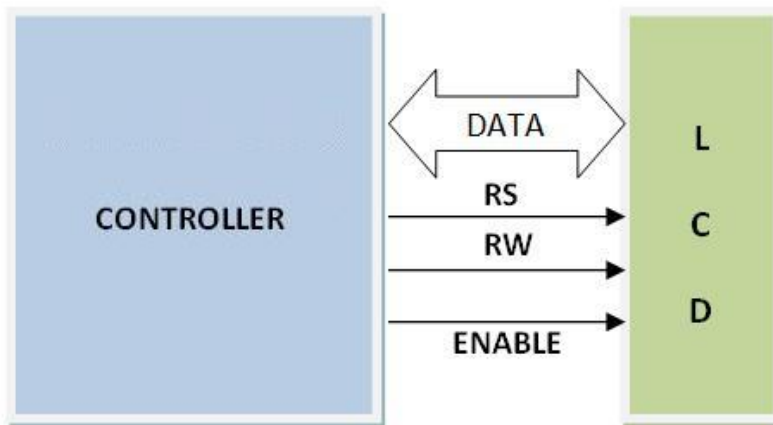
COMMANDS USED IN LCD

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1

Basic 16x 2 Characters LCD - Black on Green 5V:

Description:

This is a basic 16 character by 2 line display. Black text on Green background. Utilizes the extremely common HD44780 parallel interface chipset. Interface code is freely available. We will need ~11 general I/O pins to interface to this LCD screen. **Includes LED backlight.**

Pin Description:

The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, whereas LCDs supporting more than 80 characters make use of 2 HD44780 controllers.

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). Pin description is shown in the table below.
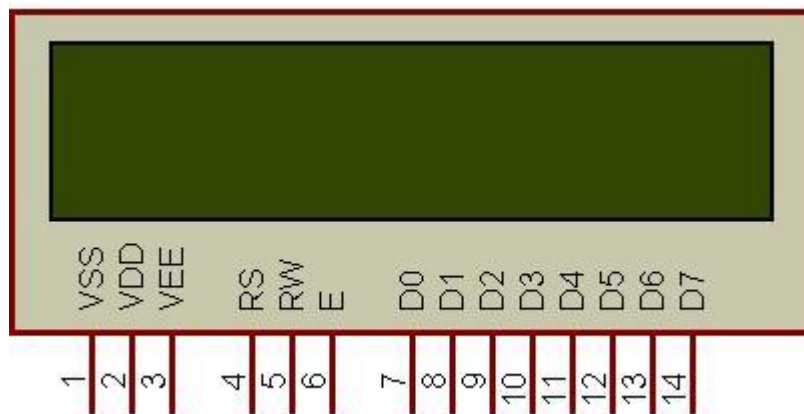
Figure 1: Character LCD type HD44780 Pin diagram

Pin description:

Pin No.	Name	Description
Pin no. 1	VSS	Power supply (GND)
Pin no. 2	VCC	Power supply (+5V)
Pin no. 3	VEE	Contrast adjust
Pin no. 4	RS	0 = Instruction input 1 = Data input
Pin no. 5	R/W	0 = Write to LCD module 1 = Read from LCD module
Pin no. 6	EN	Enable signal
Pin no. 7	D0	Data bus line 0 (LSB)
Pin no. 8	D1	Data bus line 1
Pin no. 9	D2	Data bus line 2
Pin no. 10	D3	Data bus line 3
Pin no. 11	D4	Data bus line 4
Pin no. 12	D5	Data bus line 5
Pin no. 13	D6	Data bus line 6
Pin no. 14	D7	Data bus line 7 (MSB)

Table 1: Character LCD pins with 1 Controlle

Pin No.	Name	Description
Pin no. 1	D7	Data bus line 7 (MSB)
Pin no. 2	D6	Data bus line 6
Pin no. 3	D5	Data bus line 5
Pin no. 4	D4	Data bus line 4
Pin no. 5	D3	Data bus line 3
Pin no. 6	D2	Data bus line 2
Pin no. 7	D1	Data bus line 1
Pin no. 8	D0	Data bus line 0 (LSB)

Pin no. 9	EN1	Enable signal for row 0 and 1 (1stcontroller)
Pin no. 10	R/W	0 = Write to LCD module 1 = Read from LCD module
Pin no. 11	RS	0 = Instruction input 1 = Data input
Pin no. 12	VEE	Contrast adjust
Pin no. 13	VSS	Power supply (GND)
Pin no. 14	VCC	Power supply (+5V)
Pin no. 15	EN2	Enable signal for row 2 and 3 (2ndcontroller)
Pin no. 16	NC	Not Connected

Table 2: Character LCD pins with 2 Controller

LCD Background:

Frequently, an 8051 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 8051 is an LCD display. Some of the most common LCDs connected to the 8051 are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Fortunately, a very popular standard exists which allows us to communicate with the vast majority of LCDs regardless of their manufacturer. The standard is referred to as HD44780U, which refers to the controller chip which receives data from an external source (in this case, the 8051) and communicates directly with the LCD.

44780 Background:

The 44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred to as **EN**, **RS**, and **RW**.

The **EN** line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, our 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.

The **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 should be displayed on the screen. For example, to display the letter "T" on the screen we would set RS high.

The **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.

Interfacing Example - 16 Characters x 2 Lines LCD:

Description:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if no all Parallel Ports.

It however doesn't show the use of the Status Port as an input. A 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required running them is on board.

Schematic:
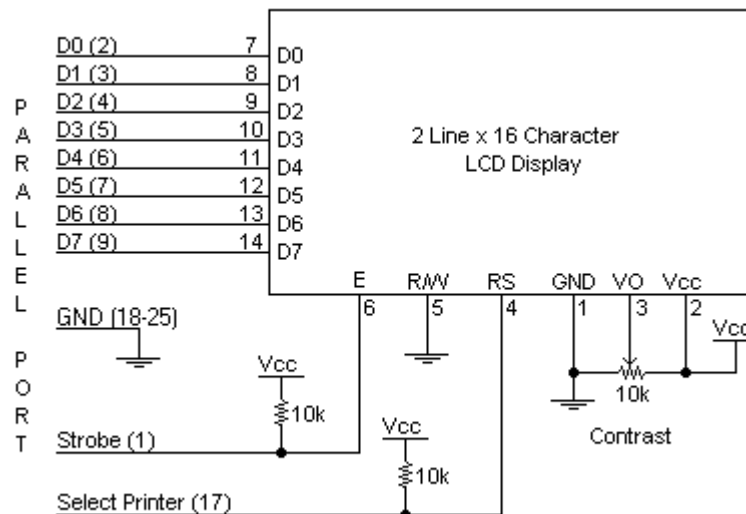
Circuit Description:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there is a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. We can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if we have trouble with the circuit working properly.

An example hardware configuration:

As we've mentioned, the LCD requires either 8 or 11 I/O lines to communicate with. For the sake of this tutorial, we are going to use an 8-bit data bus--so we'll be using 11 of the 8051's I/O pins to interface with the LCD.

Let's draw a sample pseudo-schematic of how the LCD will be connected to the 8051.
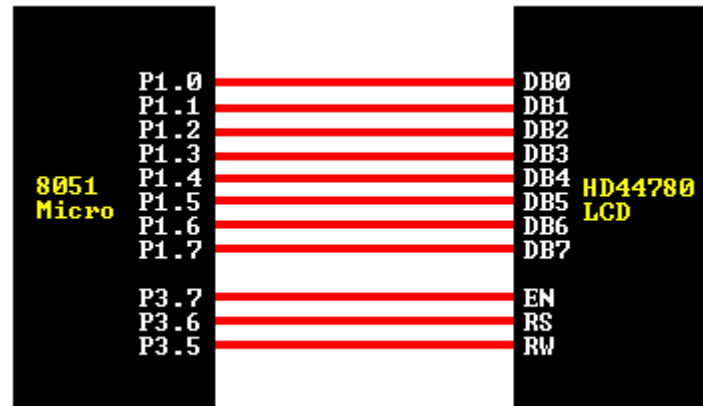
As we can see, we've established a 1-to-1 relation between a pin on the 8051 and a line on the 44780 LCD. Thus as we write our assembly program to access the LCD, we are going to equate constants to the 8051 ports so that we can refer to the lines by their 44780 name as opposed to P0.1, P0.2, etc. Let's go ahead and write our initial equates:

DB0	**EQU**	**P1.0**
DB1	**EQU**	**P1.1**
DB2	**EQU**	**P1.2**
DB3	**EQU**	**P1.3**
DB4	**EQU**	**P1.4**
DB5	**EQU**	**P1.5**
DB6	**EQU**	**P1.6**
DB7	**EQU**	**P1.7**
EN	**EQU**	**P3.7**
RS	**EQU**	**P3.6**
RW	**EQU**	**P3.5**

DATA EQU P1　　　Having established the above equates, we may now refer to our I/O lines by their 44780 name. For example, to set the RW line high (1), we can execute the following instruction:

SETB RW

Handling the EN control line:

As we mentioned above, the EN line is used to tell the LCD that we are ready for it to execute an instruction that we've prepared on the data bus and on the other control lines. Note that the EN line must be raised/ lowered before/after each instruction sent to the LCD regardless of whether that instruction is read or write text or instruction. In short, we must always manipulate EN when communicating with the LCD. EN is the LCD's way of knowing that we are talking to it. If we don't raise/lower EN, the LCD doesn't know we're talking to it on the other lines.

Thus, before we interact in any way with the LCD we will always bring the **EN** line low with the following instruction:

CLR EN

And once we've finished setting up our instruction with the other control lines and data bus lines, we'll always bring this line high:

SETB EN

The line must be left high for the amount of time required by the LCD as specified in its datasheet. This is normally on the order of about 250 nanoseconds, but checks the datasheet. In the case of a typical 8051 running at 12 MHz, an instruction requires 1.08 microseconds to execute so the EN line can be brought low the very next instruction. However, faster microcontrollers (such as the DS89C420 which executes an instruction in 90 nanoseconds given an 11.0592 MHz crystal) will require a number of NOPs to create a delay while EN is held high. The number of NOPs that must be inserted depends on the microcontroller we are using and the crystal we have selected.

The instruction is executed by the LCD at the moment the EN line is brought low with a final CLR EN instruction.

Checking the busy status of the LCD:

As previously mentioned, it takes a certain amount of time for each instruction to be executed by the LCD. The delay varies depending on the frequency of the crystal attached to the oscillator input of the 44780 as well as the instruction which is being executed.

While it is possible to write code that waits for a specific amount of time to allow the LCD to execute instructions, this method of "waiting" is not very flexible. If the crystal frequency is changed, the software will need to be modified. Additionally, if the LCD itself is changed for another LCD which, although 44780 compatible, requires more time to perform its operations, the program will not work until it is properly modified.

A more robust method of programming is to use the "Get LCD Status" command to determine whether the LCD is still busy executing the last instruction received.

The "Get LCD Status" command will return to us two tidbits of information; the information that is useful to us right now is found in DB7. In summary, when we issue the "Get LCD Status" command the LCD will immediately raise DB7 if it's still busy executing a command or lower DB7 to indicate that the LCD is no longer occupied. Thus our program can query the LCD until DB7 goes low, indicating the LCD is no longer busy. At that point we are free to continue and send the next command.

Since we will use this code every time we send an instruction to the LCD, it is useful to make it a subroutine. Let's write the code:

```
WAIT_LCD:

CLR    EN                          ;Start LCD command
CLR    RS                          ;It's a command
SETB   RW                          ;It's a read command
MOV    DATA, #0FFh                 ;Set all pins to FF initially
SETB   EN                          ;Clock out command to LCD
MOV    A,DATA                      ;Read the return value
JB     ACC.7,WAIT_LCD              ;If bit 7 high, LCD still busy
CLR    EN                          ;Finish the command
CLR    RW                          ;Turn off RW for future commands
RET
```

Thus, our standard practice will be to send an instruction to the LCD and then call our **WAIT_LCD** routine to wait until the instruction is completely executed by the LCD. This will assure that our program gives the LCD the time it needs to execute instructions and also makes our program compatible with any LCD, regardless of how fast or slow it is.

The last byte we need to send is used to configure additional operational parameters of the LCD. We must send the value 06h.

CLR		RS
MOV	DATA,	#06h
SETB		EN
CLR		EN

LCALL WAIT_LCD

So, in all, our initialization code is as follows:

INIT_LCD:

CLR		RS
MOV	DATA,	#38h
SETB		EN
CLR		EN
LCALL		WAIT_LCD
CLR		RS
MOV	DATA,	#0Eh
SETB		EN
CLR		EN
LCALL		WAIT_LCD
CLR		RS
MOV	DATA,	#06h
SETB		EN
CLR		EN
LCALL		WAIT_LCD
RET		

Having executed this code the LCD will be fully initialized and ready for us to send display data to it.

CLEARING THE DISPLAY:

When the LCD is first initialized, the screen should automatically be cleared by the 44780 controller. However, it's always a good idea to do things our self so that we can be completely sure that the display is the way we want it. Thus, it's not a bad idea to clear the screen as the very first operation after the LCD has been initialized.

An LCD command exists to accomplish this function. Not surprisingly, it is the command 01h. Since clearing the screen is a function we very likely will wish to call more than once, it's a good idea to make it a subroutine:

CLEAR_LCD:

```
CLR                                                             RS
MOV                        DATA,                              #01h
SETB                                                            EN
CLR                                                             EN
LCALL                                                     WAIT_LCD
RET
```

How that we've written a "Clear Screen" routine, we may clear the LCD at any time by simply executing an **LCALL CLEAR_LCD**

APPLICATIONS:

- Medical equipment
- Electronic test equipment
- Industrial machinery Interface
- Serial terminal
- Advertising system
- EPOS
- Restaurant ordering systems
- Gaming box
- Security systems
- R&D Test units
- Climatizing units
- PLC Interface
- Simulators
- Environmental monitoring
- Lab development
- Student projects
- Home automation

- PC external display

- HMI operator interface.

GSM

Global System for Mobile Communication (GSM)

Definition:

GSM, which stands for Global System for Mobile communications, reigns (important) as the world's most widely used cell phone technology. Cell phones use a cell phone service carrier's GSM network by searching for cell phone towers in the nearby area. Global system for mobile communication (GSM) is a globally accepted standard for digital cellular communication.

GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard that would formulate specifications for a pan-European mobile cellular radio system operating at 900 MHz. It is estimated that many countries outside of Europe will join the GSM partnership.
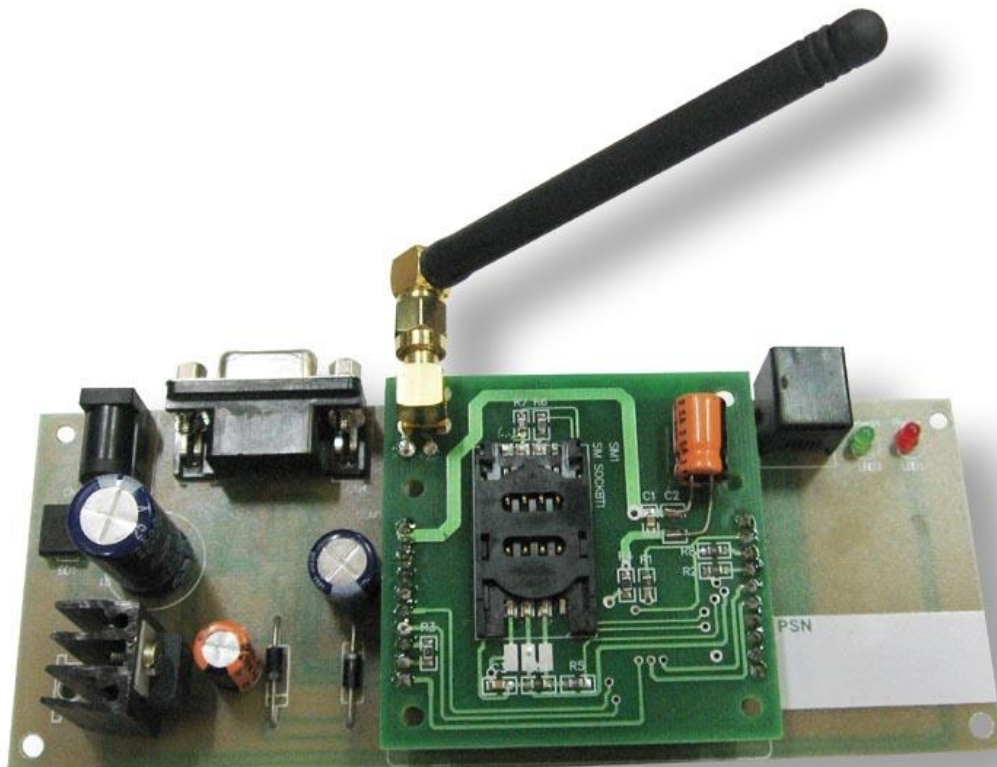
MODEM SPECIFICATIONS:

The SIM300 is a complete Tri-band GSM solution in a compact plug-in module.

Featuring an industry-standard interface, the SIM300 delivers GSM/GPRS900/1800/1900Mhz performance for voice, SMS, data and Fax in a small form factor and with low power consumption.

The leading features of SIM300 make it deal fir virtually unlimited application, such as WLL applications (Fixed Cellular Terminal), M2M application, handheld devices and much more.

1. Tri-band GSM/GPRS module with a size of 40x33x2.85
2. Customized MMI and keypad/LCD support
3. An embedded powerful TCP/IP protocol stack
4. Based upon mature and field proven platform, backed up by our support service, from definition to design and production.

General Features:

- Tri-band GSM/GPRS900/1800/1900Mhz
- GPRS multi-slot class 10
- GPRS mobile station class –B
- Complaint to GSM phase 2/2+
    i. -class 4(2W @900MHz)
    ii. -class 1(1W @/18001900MHz)
- Dimensions: 40x33x2.85 mm
- Weight: 8gm
- 7. Control via AT commands
- (GSM 07.07, 07.05 and SIMCOM enhanced AT commands)
- SIM application tool kit
- supply voltage range 3.5…….4.5 v
- Low power consumption
- Normal operation temperature: -20 'C to +55 'C
- Restricted operation temperature : -20 'C to -25 'C and +55 'C to +70 'C
- storage temperature: -40 'C to +80 'C

Specifications for Fax:

Group 3 and class 1

Specifications for Data:

- GPRS class 10: max 85.6 kbps (downlink)
- PBCCH support
- coding schemes Cs 1,2,3,4
- CSD upto 14.4 kbps
- USSD
- Non transperant mode
- PPP-stack
-

Specifications for SMS via GSM/GPRS:

- Point to point MO and MT
- SMS cell broadcast
- Text and PDU mode

Compatibility:

At cellular command interface

Specifications for voice:

1. Tricodec

-Half rate (HR)

-Full rate (FR)

-Enhanced full rate (EFR)

2. Hands free operation

(Echo cancellation)

Drivers:

Microsoft windows mobile RIL driver

MUX driver

Interfaces:

- Interface to external SIM 3v 1.8v
- 60 pins board-to-board connector
- Two analog audio interfaces
- Keypad interfaces
- LCD interface
- RTC backup
- AT commands via serial interface
- Dual-Serial interfaces
- Antenna connector and antenna pad

Approvals:

- FTA
- Local type approval
- CE

Need of GSM:

The GSM study group aimed to provide the followings through the GSM:

- Improved spectrum efficiency.
- International roaming.
- Low-cost mobile sets and base stations (BS)
- High-quality speech
- Compatibility with Integrated Services Digital Network (ISDN) and other telephone company services.
- Support for new services.

GSM – Architecture:

A GSM network consists of several functional entities whose functions and interfaces are defined. The GSM network can be divided into following broad parts.

- The Mobile Station (MS)
- The Base Station Subsystem (BSS)
- The Network Switching Subsystem (NSS)
- The Operation Support Subsystem (OSS)

Following fig shows the simple architecture diagram of GSM Network.
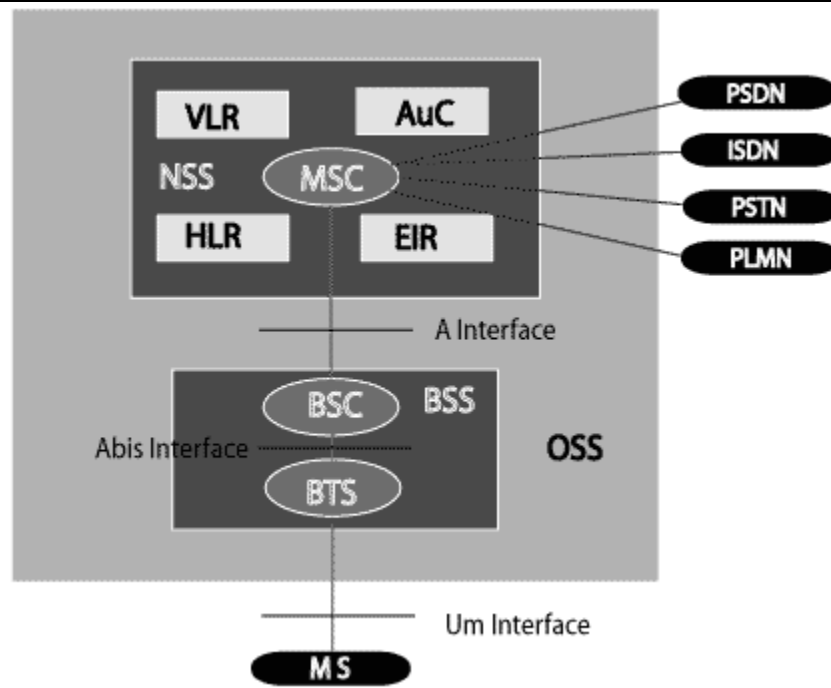
Fig: GSM Network.

The added components of the GSM architecture include the functions of the databases and messaging systems:

- Home Location Register (HLR)
- Visitor Location Register (VLR)
- Equipment Identity Register (EIR)
- Authentication Center (AuC)
- SMS Serving Center (SMS SC)
- Gateway MSC (GMSC)
- Chargeback Center (CBC)
- Transcoder and Adaptation Unit (TRAU)

Following fig shows the diagram of GSM Network along with added elements.
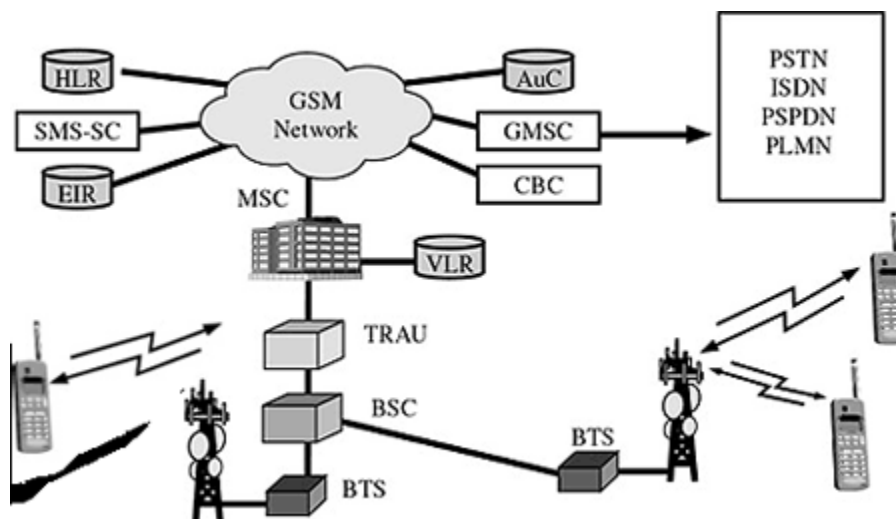
Fig: GSM Network along with added elements.

The MS and the BSS communicate across the Um interface, also known as the air interface or radio link. The BSS communicates with the Network Service Switching center across the A interface.

GSM network areas:

In a GSM network, the following areas are defined:

Cell: Cell is the basic service area, one BTS covers one cell. Each cell is given a Cell Global Identity (CGI), a number that uniquely identifies the cell.

Location Area: A group of cells form a Location Area. This is the area that is paged when a subscriber gets an incoming call. Each Location Area is assigned a Location Area Identity (LAI). Each Location Area is served by one or more BSCs.

MSC/VLR Service Area: The area covered by one MSC is called the MSC/VLR service area.

PLMN: The area covered by one network operator is called PLMN. A PLMN can contain one or more MSCs.

The GSM networks parts are explained as follows:

1) Mobile Station:

The mobile station (MS) consists of the physical equipment, such as the radio transceiver, display and digital signal processors, and a smart card called the Subscriber Identity Module (SIM). The SIM provides personal mobility, so that the user can have access to all subscribed services irrespective of both the location of

the terminal and the use of a specific terminal. By inserting the SIM card into another GSM cellular phone, the user is able to receive calls at that phone, make calls from that phone, or receive other subscribed services.

The mobile equipment is uniquely identified by the International Mobile Equipment Identity (IMEI). The SIM card contains the International Mobile Subscriber Identity (IMSI), identifying the subscriber, a secret key for authentication, and other user information. The IMEI and the IMSI are independent, thereby providing personal mobility. The SIM card may be protected against unauthorized use by a password or personal identity number.

2) Base Station Subsystem:

The Base Station Subsystem is composed of two parts, the Base Transceiver Station (BTS) and the Base Station Controller (BSC). These communicate across the specified Abis interface, allowing (as in the rest of the system) operation between components made by different suppliers.

The Base Transceiver Station houses the radio transceivers that define a cell and handles the radio link protocols with the Mobile Station. In a large urban area, there will potentially be a large number of BTSs deployed. The requirements for a BTS are ruggedness, reliability, portability, and minimum cost.

The Base Station Controller manages the radio resources for one or more BTSs. It handles radio channel setup, frequency hopping, and handovers, as described below. The BSC is the connection between the mobile and the Mobile service Switching Center (MSC). The BSC also translates the 13 kbps voice channel used over the radio link to the standard 64 kbps channel used by the Public Switched Telephone Network or ISDN.

3) Network Subsystem:

The central component of the Network Subsystem is the Mobile services Switching Center (MSC). It acts like a normal switching node of the PSTN or ISDN, and in addition provides all the functionality needed to handle a mobile subscriber, such as registration, authentication, location updating, handovers, and call routing to a roaming subscriber. These services are provided in conjunction with several functional entities, which together form the Network Subsystem. The MSC provides the connection to the public fixed network (PSTN or ISDN), and signaling between functional entities uses the ITUT Signaling System Number 7 (SS7), used in ISDN and widely used in current public networks.
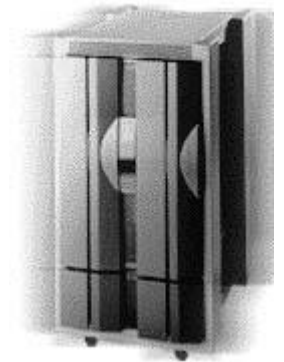
The Home Location Register (HLR) and Visitor Location Register (VLR), together with the MSC, provide the call routing and (possibly international) roaming capabilities of GSM. The HLR contains all the administrative information of each subscriber registered in the corresponding GSM network, along with the current location of the mobile. The current location of the mobile is in the form of a Mobile Station Roaming Number (MSRN) which is a regular ISDN number used to route a call to the MSC where the mobile is currently located. There is logically one HLR per GSM network, although it may be implemented as a distributed database.

The Visitor Location Register contains selected administrative information from the HLR, necessary for call control and provision of the subscribed services, for each mobile currently located in the geographical area controlled by the VLR. Although each functional entity can be implemented as an independent unit, most manufacturers of switching equipment implement one VLR together with one MSC, so that the geographical area controlled by the MSC corresponds to that controlled by the VLR, simplifying the signaling required. Note that the MSC contains no information about particular mobile stations - this information is stored in the location registers.

The other two registers are used for authentication and security purposes. The Equipment Identity Register (EIR) is a database that contains a list of all valid mobile equipment on the network, where each mobile station is identified by its International Mobile Equipment Identity (IMEI). An IMEI is marked as invalid if it has been reported stolen or is not type approved. The Authentication Center is a protected database that stores a copy of the secret key stored in each subscriber's SIM card, which is used for authentication and ciphering of the radio channel.

GSM - The Base Station Subsystem (BSS):

The BSS is composed of two parts:

- The Base Transceiver Station (BTS)
- The Base Station Controller (BSC)

The BTS and the BSC communicate across the specified Abis interface, enabling operations between components that are made by different suppliers. The radio components of a BSS may consist of four to seven or nine cells. A BSS may have one or more base stations. The BSS uses the Abis interface between the BTS and the BSC. A separate high-speed line (T1 or E1) is then connected from the BSS to the Mobile MSC.
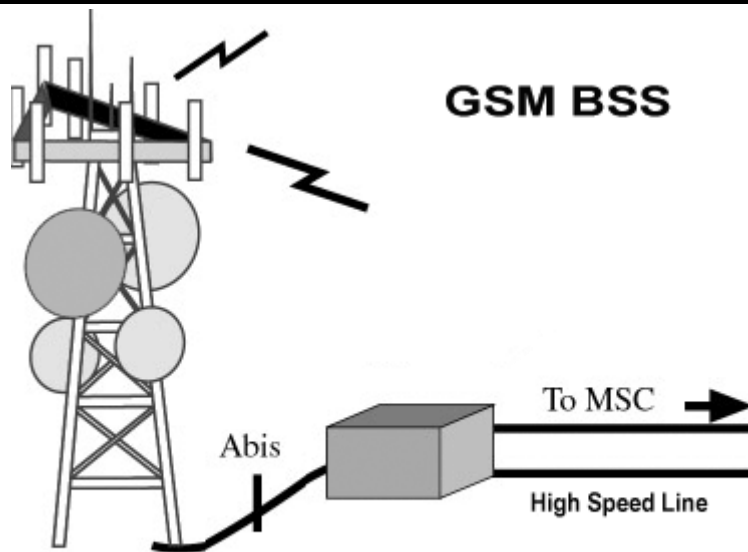
The Base Transceiver Station (BTS):

The BTS houses the radio transceivers that define a cell and handles the radio link protocols with the MS. In a large urban area, a large number of BTSs may be deployed.
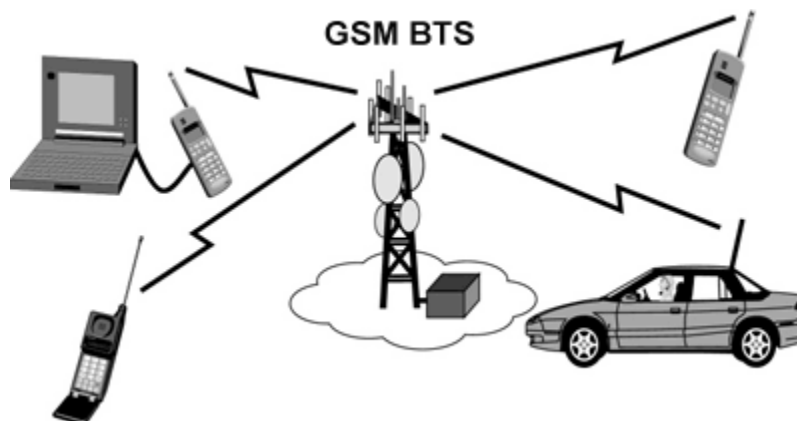
The BTS corresponds to the transceivers and antennas used in each cell of the network. A BTS is usually placed in the center of a cell. Its transmitting power defines the size of a cell. Each BTS has between 1 and 16 transceivers, depending on the density of users in the cell. Each BTS serves a single cell. It also includes the following functions:

- Encoding, encrypting, multiplexing, modulating, and feeding the RF signals to the antenna.
- Transcoding and rate adaptation
- Time and frequency synchronizing
- Voice through full- or half-rate services
- Decoding, decrypting, and equalizing received signals
- Random access detection

- Timing advances

- Uplink channel measurements

The Base Station Controller (BSC):

The BSC manages the radio resources for one or more BTSs. It handles radio channel setup, frequency hopping, and handovers. The BSC is the connection between the mobile and the MSC. The BSC also translates the 13 Kbps voice channel used over the radio link to the standard 64 Kbps channel used by the Public Switched Telephone Network (PSDN) or ISDN.

It assigns and releases frequencies and time slots for the MS. The BSC also handles intercell handover. It controls the power transmission of the BSS and MS in its area. The function of the BSC is to allocate the necessary time slots between the BTS and the MSC. It is a switching device that handles the radio resources. Additional functions include:

- Control of frequency hopping

- Performing traffic concentration to reduce the number of lines from the MSC

- Providing an interface to the Operations and Maintenance Center for the BSS

- Reallocation of frequencies among BTSs

- Time and frequency synchronization

- Power management

- Time-delay measurements of received signals from the MS

The Network Switching Subsystem (NSS):

The Network switching system (NSS), the main part of which is the Mobile Switching Center (MSC), performs the switching of calls between the mobile and other fixed or mobile network users, as well as the management of mobile services such as authentication.
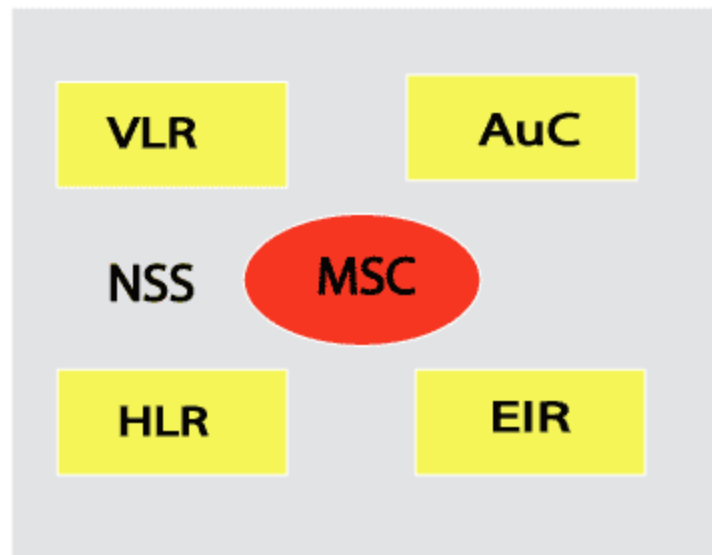
The switching system includes the following functional elements.

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 in the form of SIM then all the information about this subscription is registered in the HLR of that operator.

Mobile Services Switching Center (MSC):

The central component of the Network Subsystem is the MSC. The MSC performs the switching of calls between the mobile and other fixed or mobile network users, as well as the management of mobile services such as such as registration, authentication, location updating, handovers, and call routing to a roaming subscriber. It also performs such functions as toll ticketing, network interfacing, common channel signaling, and others. Every MSC is identified by a unique ID.

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):

The Authentication Center is a protected database that stores a copy of the secret key stored in each subscriber's SIM card, which is used for authentication and ciphering of the radio channel. The AUC protects network operators from different types of fraud found in today's cellular world.

Equipment Identity Register (EIR):

The Equipment Identity Register (EIR) is a database that contains a list of all valid mobile equipment on the network, where its International Mobile Equipment Identity (IMEI) identifies each MS. An IMEI is marked as invalid if it has been reported stolen or is not type approved.

4) The Operation Support Subsystem (OSS):

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).

Here are some of the OMC functions:

- Administration and commercial operation (subscription, end terminals, charging and statistics).
- Security Management.
- Network configuration, Operation and Performance Management.
- Maintenance Tasks.

The operation and Maintenance functions are based on the concepts of the Telecommunication Management Network (TMN) which is standardized in the ITU-T series M.30.

Following is the figure which shows how OMC system covers all the GSM elements.
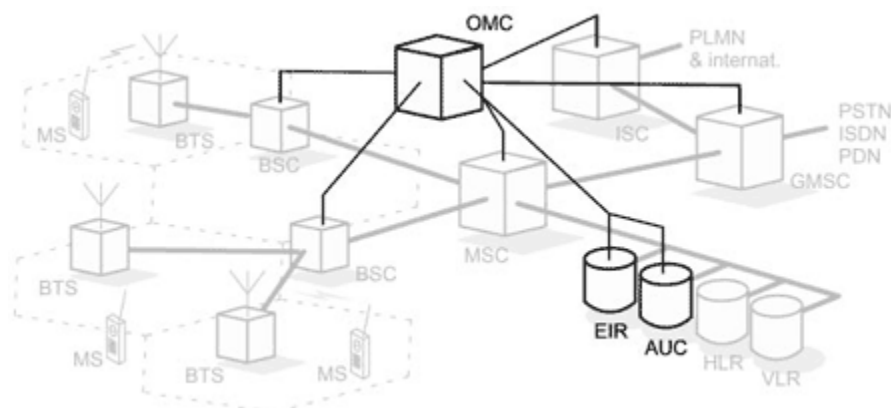
The OSS is the functional entity from which the network operator monitors and controls the system. The purpose of OSS is to offer the customer cost-effective support for centralized, regional and local operational and maintenance activities that are required for a GSM network. An important function of OSS is to provide a network overview and support the maintenance activities of different operation and maintenance organizations.

The GSM Specifications:

Specifications for different Personal Communication Services (PCS) systems vary among the different PCS networks. The GSM specification is listed below with important characteristics.

Modulation:

Modulation is a form of change process where we change the input information into a suitable format for the transmission medium. We also changed the information by demodulating the signal at the receiving end.

The GSM uses Gaussian Minimum Shift Keying (GMSK) modulation method.

Access Methods:

Because radio spectrum is a limited resource shared by all users, a method must be devised to divide up the bandwidth among as many users as possible.

GSM chose a combination of TDMA/FDMA as its method. The FDMA part involves the division by frequency of the total 25 MHz bandwidth into 124 carrier frequencies of 200 kHz bandwidth.

One or more carrier frequencies are then assigned to each BS. Each of these carrier frequencies is then divided in time, using a TDMA scheme, into eight time slots. One time slot is used for transmission by the mobile and one for reception. They are separated in time so that the mobile unit does not receive and transmit at the same time.

Transmission Rate:

The total symbol rate for GSM at 1 bit per symbol in GMSK produces 270.833 K symbols/second. The gross transmission rate of the time slot is 22.8 Kbps.

GSM is a digital system with an over-the-air bit rate of 270 kbps.

Frequency Band:

The uplink frequency range specified for GSM is 933 - 960 MHz (basic 900 MHz band only). The downlink frequency band 890 - 915 MHz (basic 900 MHz band only).

Channel Spacing: This indicates separation between adjacent carrier frequencies. In GSM, this is 200 kHz.

Speech Coding:

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.

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.

Misc:

*   Frame duration: 4.615 ms
*   Duplex Technique: Frequency Division Duple Xing (FDD) access mode previously known as WCDMA.
*   Speech channels per RF channel: 8.

Advantages of GSM:

*   GSM is already used worldwide with over 450 million subscribers.
*   International roaming permits subscribers to use one phone throughout Western Europe. CDMA will work in Asia, but not France, Germany, the U.K. and other popular European destinations.
*   GSM is mature, having started in the mid-80s. This maturity means a more stable network with robust features. CDMA is still building its network.
*   GSM's maturity means engineers cut their teeth on the technology, creating an unconscious preference.
*   The availability of Subscriber Identity Modules, which are smart cards that provide secure data encryption give GSM m-commerce advantages.

GSM COMMANDS:

Commands always start with AT (which means Attention) and finish with a <CR> character.

Information responses and result codes

Responses start and end with <CR><LF>, except for the ATV0 DCE response format) and the ATQ1 (result code suppression) commands.

_ If command syntax is incorrect, an **ERROR** string is returned.

_ If command syntax is correct but with some incorrect parameters, the **+CME ERROR:**

<Err> or **+CMS ERROR:** <Sms Err> strings are returned with different error codes.

_ If the command line has been performed successfully, an **OK** string is returned.

In some cases, such as "AT+CPIN?" or (unsolicited) incoming events, the product does not return the **OK** string as a response.

In the following examples <CR> and <CR><LF> are intentionally omitted.

1. Manufacturer identification +CGMI

2. Request model identification +CGMM

3. Request revision identification +CGMR

4. Product Serial Number +CGSN

5. Dial command D

ATD<nb> where <nb> is the destination phone number.

Please note that for an **international number**, the local international prefix does not need to be set (usually 00) but does need to be replaced by the **'+'** character.

Example: to set up a voice call to Wavecom offices from another country, the AT command is:

"ATD+33146290800;"

Note that some countries may have specific numbering rules for their GSM handset numbering. The response to the ATD command is one of the following:

6. Hang-Up command H

Description:

The ATH (or ATH0) command disconnects the remote user. In the case of multiple calls, all calls are released (active, on-hold and waiting calls). The specific Wavecom ATH1 command has been appended to disconnect the current outgoing call, only in dialing or alerting state (ie. ATH1 can be used only after the ATD command, and before its terminal response (OK, NO CARRIER, ...). It can be useful in the case of multiple calls.

Syntax:

Command syntax: ATH

7. Answer a call A

Description:

When the product receives a call, it sets the **RingInd** signal and sends the ASCII "**RING**" or

"**+CRING: <type>**" string to the application (+CRING if the cellular result code +CRC is enabled). Then it waits for the application to accept the call with the ATA command.

Syntax:

Command syntax: ATA

8. Redial last telephone number ATDL

Description:

This command redials the last number used in the ATD command. The last number dialed is displayed followed by ";" for voice calls only

Syntax:

Command syntax: ATDL

9. Preferred Message Format +CMGF

Description:

The message formats supported are text mode and PDU mode. In PDU mode, a complete SMS Message including all header information is given as a binary string (in hexadecimal format). Therefore, only the following set of characters is allowed: {'0','1','2','3','4','5','6','7','8','9', 'A', 'B','C','D','E','F'}. Each pair or characters are converted to a byte (e.g.: '41' is converted to the ASCII character 'A', whose ASCII code is 0x41 or 65). In Text mode, all commands and responses are in ASCII characters. The format selected is stored in EEPROM by the +CSAS command.

Syntax:

Command syntax: AT+CMGF

10. Read message +CMGR

Description:

This command allows the application to read stored messages. The messages are read from the memory selected by **+CPMS** command.

Syntax:

Command syntax: AT+CMGR=<index>

11. Send message +CMGS

Description:

The <address> field is the address of the terminal to which the message is sent. To send the message, simply type, <ctrl-Z> character (ASCII 26). The text can contain all existing characters except <ctrl-Z> and <ESC> (ASCII 27). This command can be aborted using the <ESC> character when entering text. In PDU mode, only hexadecimal characters are used ('0'…'9','A'…'F').

Syntax:

AT+CMGS= <length> <CR>

PDU is entered <ctrl-Z / ESC >

12. Delete message +CMGD

Description:

This command deletes one or several messages from preferred message storage ("BM" SMS CB 'RAM storage', "SM" SMSPP storage 'SIM storage' or "SR" SMS Status-Report storage).

Syntax:

Command syntax: AT+CMGD=<Index> [,<DelFalg>]

GPS (GLOBAL POSITIONING SYSTEM)

The Global Positioning System (GPS) is a U.S. space-based radio navigation system that provides reliable positioning, navigation, and timing services to civilian users on a continuous worldwide basis -- freely available to all. For anyone with a GPS receiver, the system will provide location and time. GPS provides accurate location and time information for an unlimited number of people in all weather, day and night, anywhere in the world.

The GPS is made up of three parts:

1. Satellites orbiting the Earth

2. Control and monitoring stations on Earth

 3. The GPS receivers owned by users.

GPS satellites broadcast signals from space that are picked up and identified by GPS receivers. Each GPS receiver then provides three-dimensional location (latitude, longitude, and altitude) plus the time.

1. SPACE SEGMENT
   - 24+ satellites
   - 20,200 km altitude
   - 55 degrees inclination
   - 12 hour orbital period
   - 5 ground control stations
   - Each satellite passes over a ground monitoring station every 12 hours
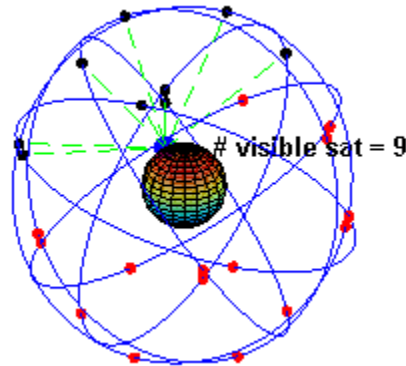
The GPS satellite system

The space segment is composed of the orbiting GPS satellites or Space Vehicles (SV) in GPS parlance. The GPS design originally called for 24 SVs, this was modified to six planes with four satellites each. The orbital planes are centered on the Earth, not rotating with respect to the distant stars. The six planes have approximately 55° inclination (tilt relative to Earth's equator) and are separated by 60° right ascension of the ascending node (angle along the equator from a reference point to the orbit's intersection). The orbits are arranged so that at least six satellites are always within line of sight from almost everywhere on Earth's surface.

The full constellation of 24 satellites that make up the GPS space segment are orbiting the earth about 20,200 km above us. They are constantly moving, making two complete orbits in less than 24 hours. These satellites are travelling at speeds of roughly 7,000 miles an hour.

GPS satellites are powered by solar energy. They have backup batteries onboard to keep them running in the event of a solar eclipse, when there's no solar power. Small rocket boosters on each satellite keep them flying in the correct path.

Here are some other interesting facts about the GPS satellites (also called NAVSTAR, the official U.S. Department of Defense name for GPS):

- The first GPS satellite was launched in 1978.
- A full constellation of 24 satellites was achieved in 1994.

- Each satellite is built to last about 10 years. Replacements are constantly being built and launched into orbit.

- A GPS satellite weighs approximately 2,000 pounds and is about 17 feet across with the solar panels extended.

- Transmitter power is only 50 watts or less.

- The orbits are arranged so that at any time, anywhere on Earth, there are at least four satellites "visible" in the sky.

- All satellites broadcast at the same two frequencies, 1.57542 GHz (L1 signal) and 1.2276 GHz (L2 signal).

- The satellite network uses a CDMA spread-spectrum technique where the low-bitrate message data is encoded with a high-rate pseudo-random (PRN) sequence that is different for each satellite.

The receiver must be aware of the PRN codes for each satellite to reconstruct the actual message data. The C/A code, for civilian use, transmits data at 1.023 million chips per second, whereas the P code, for U.S. military use, transmits at 10.23 million chips per second. The L1 carrier is modulated by both the C/A and P codes, while the L2 carrier is only modulated by the P code. The P code can be encrypted as a so-called P(Y) code which is only available to military equipment with a proper decryption key. Both the C/A and P(Y) codes impart the precise time-of-day to the user

2. Control and monitoring stations on Earth

Ground Stations (also known as the "Control Segment")

These stations monitor the GPS satellites, checking both their operational health and their exact position in space. The master ground station transmits corrections for the satellite's ephemeris constants and clock offsets back to the satellites themselves. The satellites can then incorporate these updates in the signals they send to GPS receivers.

There are five monitor stations: Hawaii, Ascension Island, Diego Garcia, Kwajalein, and Colorado Springs.

Each GPS satellite regularly with a navigational update using dedicated or shared ground antennas (GPS dedicated ground antennas are located at Kwajalein, Ascension Island, Diego Garcia, and Cape Canaveral). These updates synchronize the atomic clocks on board the satellites to within a few nanoseconds of each other, and adjust the ephemeris of each satellite's internal orbital model. The updates are created by a Kalman filter, which uses inputs from the ground monitoring stations, space weather information, and various other inputs. Satellite maneuvers are not precise by GPS standards. So to change the orbit of a satellite, the satellite must be marked unhealthy, so receivers will not use it in their calculation. Then the maneuver can be carried out, and the resulting orbit tracked from the ground. Then the new ephemeris is uploaded and the satellite marked healthy again.

3. THE GPS receivers

- Receiver determines location, speed, direction, and time
- 3 satellite signals are necessary to locate the receiver in 3D space
- 4th satellite is used for time accuracy
- Position calculated within sub-centimeter scale

Individuals may purchase GPS handsets that are readily available through commercial retailers. Equipped with these GPS receivers, users can accurately locate where they are and easily navigate to where they want to go, whether walking, driving, flying, or boating.

Today's GPS receivers are extremely accurate, thanks to their parallel multi-channel design. Garmin's 12 parallel channel receivers are quick to lock onto satellites when first turned on and they maintain strong locks, even in dense foliage or urban settings with tall buildings. Certain atmospheric factors and other sources of error can affect the accuracy of GPS receivers. Garmin® GPS receivers are accurate to within 15 meters on average.

-
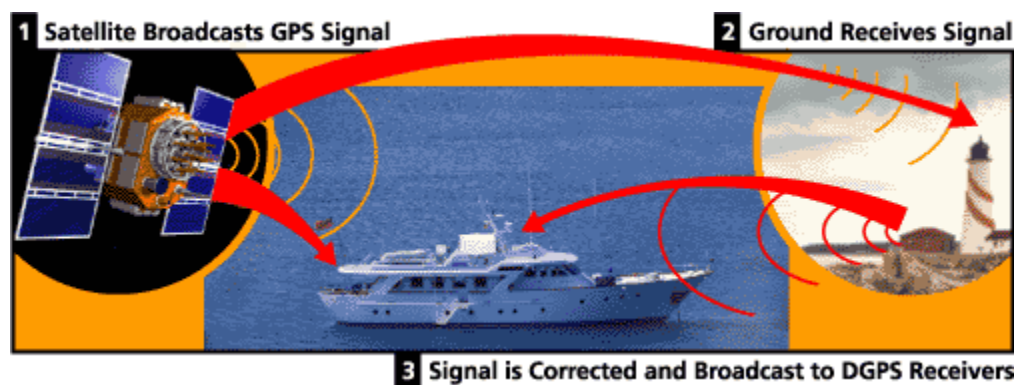
Newer Garmin GPS receivers with WAAS (Wide Area Augmentation

System) capability can improve accuracy to less than three meters on average. No additional equipment or fees are required to take advantage of WAAS. Users can also get better accuracy with Differential GPS (DGPS), which corrects GPS signals to within an average of three to five meters. The U.S. Coast Guard operates the most common DGPS correction service. This system consists of a network of towers that receive GPS signals and transmit a corrected signal by beacon transmitters. In order to get the corrected signal, users must have a differential beacon receiver and beacon antenna in addition to their GPS.

Our ancestors had to go to pretty extreme measures to keep from getting lost. They erected monumental landmarks, laboriously drafted detailed maps and learned to read the stars in the night sky.

Things are much, much easier today. For less than $100, you can get a pocket-sized gadget that will tell you exactly where you are on Earth at any moment. As long as you have a GPS receiver and a clear view of the sky, you'll never be lost again.
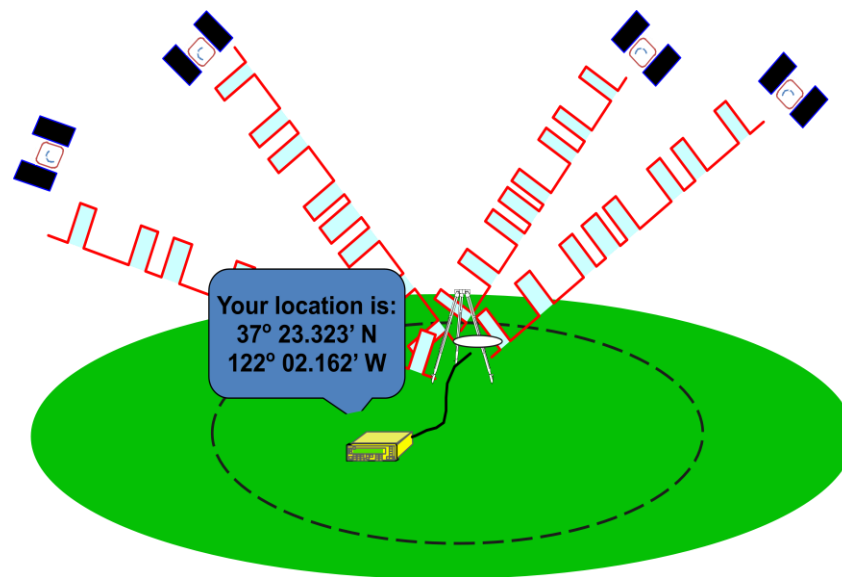
The user segment is composed of hundreds of thousands of U.S. and allied military users of the secure GPS Precise Positioning Service, and tens of millions of civil, commercial and scientific users of the Standard Positioning Service. In general, GPS receivers are composed of an antenna, tuned to the frequencies transmitted by the satellites, receiver-processors, and a highly stable clock (often a crystal oscillator). They may also include a display for providing location and speed information to the user. A receiver is often described by its number of channels: this signifies how many satellites it can monitor simultaneously. Originally limited to four or five, this has progressively increased over the years so that, as of 2007, receivers typically have between 12 and 20 channels.

The Global Positioning System is vast, expensive and involves a lot of technical ingenuity, but the fundamental concepts at work are quite simple and intuitive.

When people talk about "a GPS," they usually mean a GPS receiver. The Global Positioning System (GPS) is actually a constellation of 24 Earth-orbiting satellites The U.S. military developed and implemented this satellite network as a military navigation system, but soon opened it up to everybody else.

Each of these 3,000- to 4,000-pound solar-powered satellites circles the globe making two complete rotations every day. The orbits are arranged so that at any time, anywhere on Earth, there are at least four satellites "visible" in the sky.
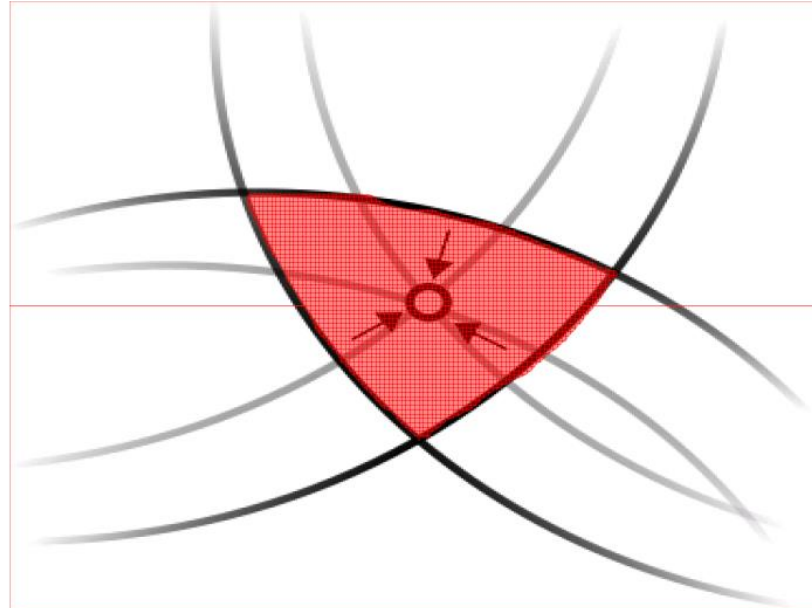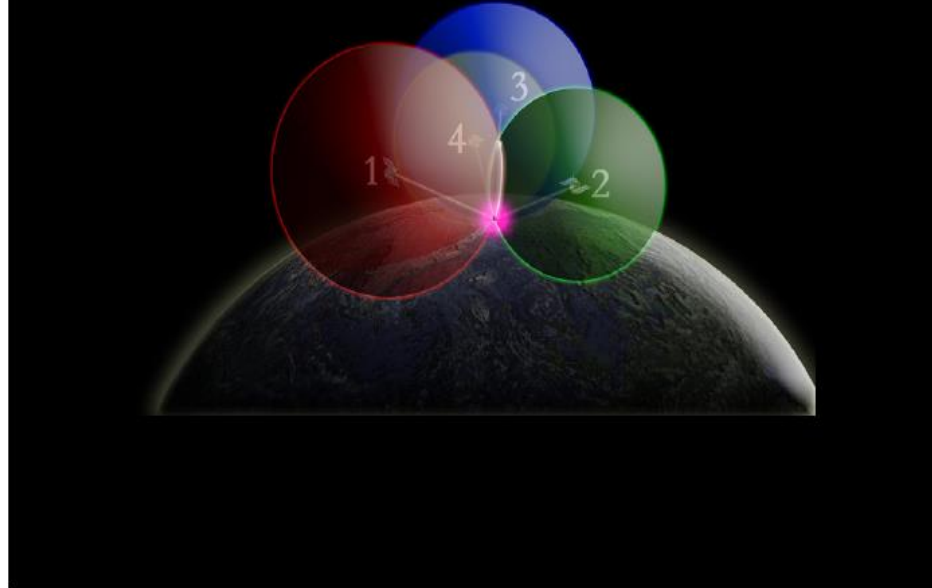
A GPS receiver's job is to locate four or more of these satellites, figure out the distance to each, and use this information to deduce its own location. This operation is based on a simple mathematical principle called trilateration. Trilateration in three-dimensional space can be a little tricky, so we'll start with an explanation of simple two-dimensional trilateration.

APPLICATIONS OF GPS

- GPS has become a mainstay of transportation systems worldwide,

- Providing navigation for aviation, ground, and maritime operations.

- Disaster relief and emergency services depend upon GPS for location and timing capabilities in their life-saving missions.

- Everyday activities such as banking,

- Mobile phone operations, and even

- The control of power grids, are facilitated by the accurate timing provided by GPS.

- Farmers, surveyors, geologists and countless others perform their work more efficiently, safely, economically, and accurately using the free and open GPS signals.

COMMANDS IN GPS

NMEA record	Description
GGA	Global positioning system fixed data
GLL	Geographic position - latitude/longitude
GSA	GNSS DOP and active satellites
GSV	GNSS satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed

- • **GGA--- Global Positioning System Fixed Data**
- Table 5-2 contains the values for the following example:
- $GPGGA,053740.000,2503.6319,N,12136.0099,E,1,08 ,1.1,63.8,M,15.2,M,,0000*64 *Table5- 2 GGA Data Format*

Name	Example	Units	Description
Message ID	$GPGGA		GGA protocol header
UTC Time	053740.000		hhmmss.sss
Latitude	2503.6319		ddmm.mmmm
N/S indicator	N		N=north or S=south
Longitude	12136.0099		dddmm.mmmm
E/W Indicator	E		E=east or W=west
Position Fix Indicator	1		See Table 5-3
Satellites Used	08		Range 0 to 12
HDOP	1.1		Horizontal Dilution of Precision
MSL Altitude	63.8	mters	

Units	M	mters	
Geoid Separation	15.2	mters	
Units	M	mters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*64		
<CR> <LF>			End of message termination

- *Table 5-3* Position Fix Indicators

-

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix
3-5	Not supported
6	Dead Reckoning Mode, fix valid

- • **GLL--- Geographic Position - Latitude/Longitude**

- Table 5-4 contains the values for the following example:

- $GPGLL,2503.6319,N,12136.0099,E

  ,053740.000,A,A*52 *Table 5-4* GLL

  Data Format

-

Name	Example	Units	Description
Message ID	$GPGLL		GLL protocol header
Latitude	2503.6319		ddmm.mmmm
N/S indicator	N		N=north or S=south
Longitude	12136.0099		dddmm.mmmm
E/W indicator	E		E=east or W=west
UTC Time	053740.000		hhmmss.sss
Status	A		A=data valid or V=data not valid
Mode	A		A=autonomous, D=DGPS, E=DR
Checksum	*52		
<CR> <LF>			End of message termination

- • **GSA---GNSS DOP and Active Satellites**

- Table 5-5 contains the values for the following example:

- $GPGSA,A,3,24,07,17,11,28,08,20,0

   4,,,,,2.0,1.1,1.7*35 *Table 5-5 GSA*

   Data Format

-

Name	Example	Units	Description
Message ID	$GPGSA		GSA protocol header
Mode 1	A		See Table 5-6
Mode 2	3		See Table 5-7
ID of satellite used	24		Sv on Channel 1
ID of satellite used	07		Sv on Channel 2
….			….
ID of satellite used			Sv on Channel 12
PDOP	2.0		Position Dilution of Precision
HDOP	1.1		Horizontal Dilution of Precision
VDOP	1.7		Vertical Dilution of Precision
Checksum	*35		
<CR> <LF>			End of message termination

- *Table 5-6 Mode 1*

-

Value	Description
M	Manual- forced to operate in 2D or
A	Automatic-allowed to automatically

- *Table 5-7 Mode 2*

-

Value	Description
1	Fix not available
2	2D
3	3D

- • **GSV---GNSS Satellites in View**

   - Table 5-8 contains the values for the following example:

- $GPGSV,3,1,12,28,81,285,42,24,67,302,46,31,54,354,,20,51,

   077,46*73

   $GPGSV,3,2,12,17,41,328,45,07,32,315,45,04,31,250,40,11,2

   5,046,41*75

- $GPGSV,3,3,12,08,22,214,38,27,08,190,16,19,05,092,

33,23,04,127,*7B *Table 5-8* GSV Data Format

Name	Example	Units	Description
Message ID	$GPGSV		GSV protocol header
Total number of Message number[1]	3		Range 1 to 3
Message number[1]	1		Range 1 to 3
Satellites in view	12		
Satellite ID	28		Channel 1 (Range 01 to 32)
Elevation	81	degrees	Channel 1 (Range 00 to 90)
Azimuth	285	degrees	Channel 1 (Range 000 to 359)
SNR (C/No)	42	dB-Hz	Channel 1 (Range 00 to 99, null when not
Satellite ID	20		Channel 4 (Range 01 to 32)
Elevation	51	degrees	Channel 4 (Range 00 to 90)
Azimuth	077	degrees	Channel 4 (Range 000 to 359)
SNR (C/No)	46	dB-Hz	Channel 4 (Range 00 to 99, null when not
Checksum	*73		
<CR> <LF>			End of message termination

- 1. Depending on the number of satellites tracked multiple messages of GSV data may be required.

- • **RMC---Recommended Minimum Specific GNSS Data**

- Table 5-9 contains the values for the following example:

- $GPRMC,053740.000,A,2503.6319,N,12136.0099,E,2.69,79.65,

  100106,,,A*53 *Table 5-9* RMC Data Format

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol header
UTC Time	053740.000		hhmmss.sss
Status	A		A=data valid or V=data not valid
Latitude	2503.6319		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12136.0099		dddmm.mmmm
E/W Indicator	E		E=east or W=west
Speed over ground	2.69	knots	True
Course over ground	79.65	degrees	
Date	100106		Ddmmyy
Magnetic variation		degrees	Not shown
Variation sense			E=east or W=west (Not shown)
Mode	A		A=autonomous, D=DGPS, E=DR
Checksum	*53		
<CR> <LF>			End of message termination

Introduction to IR communication:

As next-generation electronic information systems evolve, it is critical that all people have access to the information available via these systems. Examples of developing and future information systems include interactive television, touch screen-based information kiosks, and advanced Internet programs. Infrared technology, increasingly present in mainstream applications, holds great potential for enabling people with a variety of disabilities to access a growing list of information resources. Already commonly used in remote control of TVs, VCRs and CD players, infrared technology is also being used and developed for remote control of environmental control systems, personal computers, and talking signs.

For individuals with mobility impairments, the use of infrared or other wireless technology can facilitate the operation of information kiosks, environmental control systems, personal computers and associated peripheral devices. For individuals with visual impairments, infrared or other wireless communication technology can enable users to locate and access talking building directories, street signs, or other assistive navigation devices. For individuals using augmentative and alternative communication (AAC) devices, infrared or other wireless technology can provide an alternate, more portable, more independent means of accessing computers and other electronic information systems.

A discussion specific to infrared technology then follows, with advantages and disadvantages of the technology presented along with the infrared applications.

Infrared (IR) is a type of light that is not visible to the human eye. Our eyes are detectors which are designed to detect visible light waves (or visible radiation). Visible light is one of the few types of radiation that can penetrate our atmosphere and be detected on the Earth's surface. Actually we can only see a very small part of the entire range of radiation called the electromagnetic spectrum.
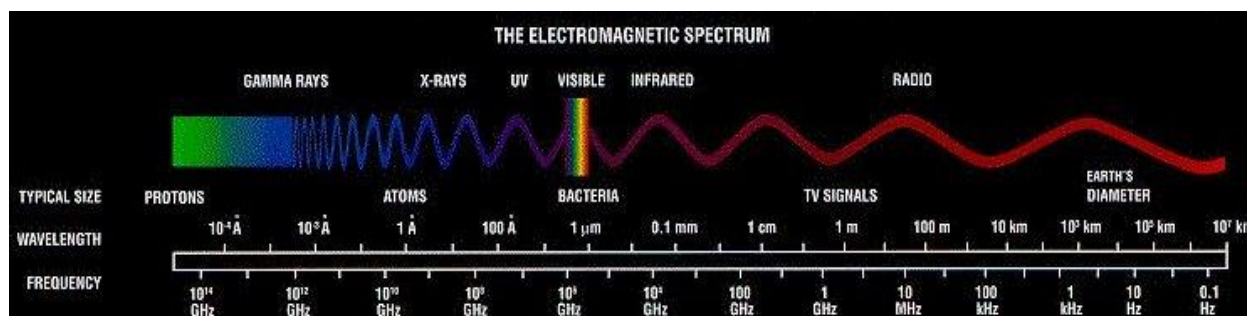

Figure (1.1): Electromagnetic spectrum

The electromagnetic spectrum includes gamma rays, X-rays, ultraviolet, visible, infrared, microwaves, and radio waves. The only difference between these different types of radiation is their wavelength or frequency. Wavelength increases and frequency decreases from gamma rays to radio waves. All of these forms of radiation

travel at the speed of light (186,000 miles or 300,000,000 meters per second in a vacuum). Infrared radiation lies between the visible and microwave portions of the electromagnetic spectrum.

Infrared waves have wavelengths longer than visible and shorter than microwaves, and have frequencies which are lower than visible and higher than microwaves. With wavelengths from 750 nm to 1 mm, infrared starts at the end of the microwave spectrum and ends at the beginning of visible light. Infrared transmission typically requires an unobstructed line of sight between transmitter and receiver.

Infrared is broken into three categories: near, mid and far-infrared. Near-infrared refers to the part of the infrared spectrum that is closest to visible light and far-infrared refers to the part that is closer to the microwave region. Mid-infrared is the region between these two. The primary source of infrared radiation is heat or thermal radiation. This is the radiation produced by the motion of atoms and molecules in an object.

The higher the temperature, the more the atoms and molecules move and the more infrared radiation they produce. Even objects that we think of as being very cold, such as an ice cube, emit infrared. When an object is not quite hot enough to radiate visible light, it will emit most of its energy in the infrared. For example, hot charcoal may not give off light but it does emit infrared radiation which we feel as heat. The warmer the object, the more infrared radiation it emits.

The following figure shows the transmitter and receiver of IR communication
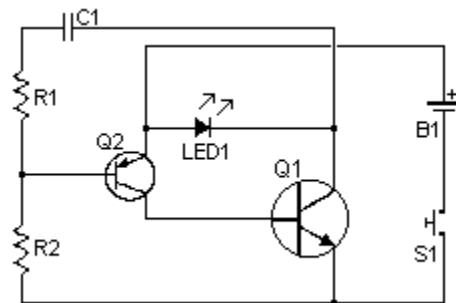
Fig: Schematic for Transmitter
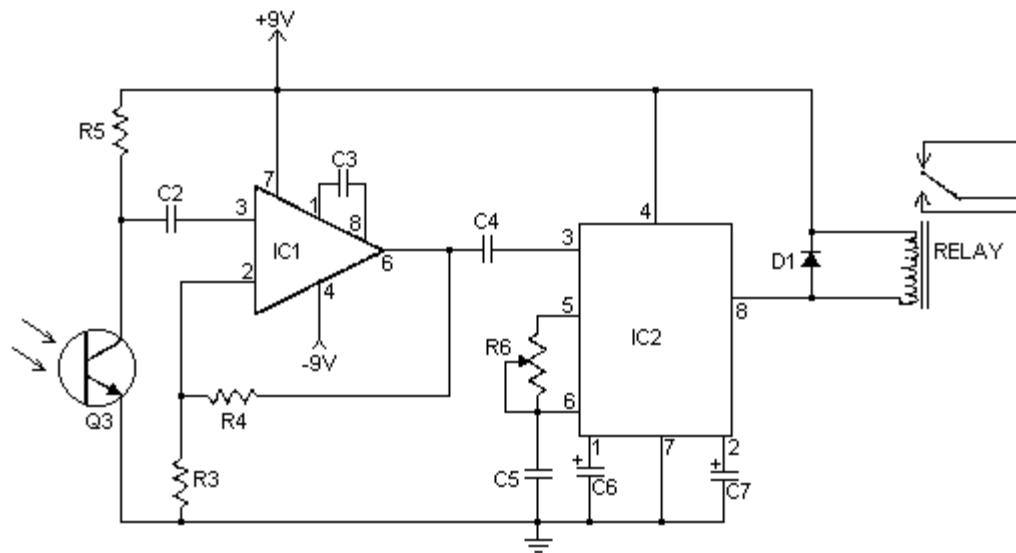
Fig: Schematic for Receiver

Working of infrared communication:

Various types of infrared based applications are available in the market. The circuit for infrared based applications is designed along with the transmitter and receiver sections i.e. we can't use it for other application. But the infrared communication project which we have done here can be used in any application just by replacing the application at the place of infrared LED in the circuit diagram of infrared communication. By using this project we can design infrared based applications easily. The entire circuit consists of two sections named as

1. Transmitter section and
2. Receiver section

1. Transmitter section:

The transmitter section consists of a 555 timer IC functioning in astable mode. It is wired as shown in figure. The output from astable mode is fed to an IR LED via resistor which limits its operating current. Infrared LED in the transmitter section emits IR radiation which is focused by a plastic lens (optics) in to a narrow beam.

2. Receiver section:

The receiver section consists of a silicon phototransistor to convert the infrared radiation to an electric current. It responds only to the rapidly pulsing signal created by the transmitter, and filters out slowly changing infrared radiation from ambient light. The receiver section comprises an infrared receiver module, and a led indicator. When the signals are interrupted, the IR Led goes off after a few seconds depending upon the value of RC combination.

We can increase the distance between the IR transmitter and receiver just by placing the lens between them. After connecting the IR transmitter and receiver circuit, we can get the output by applying 6V Power supply to the circuit. We can use this circuit with any application very simply. For example a buzzer circuit is placed at the output of IR circuit, when the signals are interrupted, the buzzer produces sound. Both the transmitter and receiver parts can be mounted on a single bread board or PCB. The infrared receiver must be placed behind the IR Led to avoid false indication due to infrared leakage. An object moving nearby actually reflects the IR rays emitted by the IR Led.

Photo Diodes:

A photodiode is a semiconductor diode that functions as a photo detector. Photodiodes are packaged with either a window or optical fiber connection, to let in the light to the sensitive part of the device. They may also be used without a window to detect vacuum UV or X-rays.

A phototransistor is in essence nothing more than a bipolar transistor that is encased in a transparent case so that light can reach the base-collector junction. The phototransistor works like a photodiode, but with a much higher responsivity for light, because the electrons that are generated by photons in the base-collector junction are injected into the base, and this current is then amplified by the transistor operation.

Fig (3.13) Photodiode schematic symbol

Principle of operation:

A photodiode is a p-n junction or p-i-n structure. When a photon of sufficient energy strikes the diode, it excites an electron thereby creating a mobile electron and a positively charged electron hole. If the absorption occurs in the junction's depletion region, or one diffusion length away from it, these carriers are swept from the junction by the built-in field of the depletion region, producing a photocurrent.

Photodiodes can be used under either zero bias (photovoltaic mode) or reverse bias (photoconductive mode). In zero bias, light falling on the diode causes a current across the device, leading to forward bias which in turn induces "dark current" in the opposite direction to the photocurrent. This is called the photovoltaic effect, and is the basis for solar cells in fact; a solar cell is just a large number of big photodiodes. Reverse bias induces only little current (known as saturation or back current) along its direction.

But a more important effect of reverse bias is widening of the depletion layer (therefore expanding the reaction volume) and strengthening the photocurrent. Circuits based on this effect are more sensitive to light than ones based on the photovoltaic effect and also tend to have lower capacitance, which improves the speed of their time response. On the other hand, the photovoltaic mode tends to exhibit less electronic noise.

Avalanche photodiodes have a similar structure, but they are operated with much higher reverse bias. This allows each photo-generated carrier to be multiplied by avalanche breakdown, resulting in internal gain within the photodiode, which increases the effective responsivity of the device.

Features:

Critical performance parameters of a photodiode include:

1. Responsivity:

The responsivity may also be expressed as quantum efficiency, or the ratio of the number of photo generated carriers to incident photons and thus a unit less quantity.

2. Dark current:

The dark current includes photocurrent generated by background radiation and the saturation current of the semiconductor junction. Dark current must be accounted for by calibration if a photodiode is used to make an accurate optical power measurement, and it is also a source of noise when a photodiode is used in an optical communication system.

3. Noise-equivalent power:

(NEP) The minimum input optical power to generate photocurrent, equal to the RMS noise current in a 1 hertz bandwidth. The related characteristic directivity (D) is the inverse of NEP, 1/NEPThe NEP is roughly the minimum detectable input power of a photodiode.

Applications:

1. P-N photodiodes are used in similar applications to other photo detectors, such as photoconductors, charge-coupled devices, and photomultiplier tubes.
2. Photodiodes are used in consumer electronics devices such as compact disc players, smoke detectors, and the receivers for remote controls in VCRs and televisions.
3. PIN diodes are much faster and more sensitive than ordinary p-n junction diodes, and hence are often used for optical communications and in lighting regulation.

P-N vs. P-I-N Photodiodes:

1. Due to the intrinsic layer, a PIN photodiode must be reverse biased (Vr). The Vr increases the depletion region allowing a larger volume for electron-hole pair production, and reduces the capacitance thereby increasing the bandwidth.

2. The Vr also introduces noise current, which reduces the S/N ratio. Therefore, a reverse bias is recommended for higher bandwidth applications and/or applications where a wide dynamic range is required.

3. A PN photodiode is more suitable for lower light applications because it allows for unbiased operation.

FIRE SENSOR

Flame detector is a <u>sensor</u> designed to detect and respond to the presence of a <u>flame</u> or <u>fire</u>, allowing flame detection. Responses to a detected flame depend on the installation, but can include sounding an alarm, deactivating a fuel line (such as a <u>propane</u> or a <u>natural gas</u> line), and activating a fire suppression system. When used in applications such as industrial furnaces, their role is to provide confirmation that the furnace is properly; in these cases they take no direct action beyond notifying the operator or control system. A flame detector can often respond faster and more accurately than a <u>smoke</u> or <u>heat detector</u> due to the mechanisms it uses to detect the flame.

Description

The Fire sensor is used to detect fire flames . The module makes use of Fire sensor and comparator to detect fire up to a range of 1 meters.

Feature

- Allows your robot to detect flames from upto 1 M away
- Typical Maximum Range :1 m .
- Calibration preset for range adjustment.
- Indicator LED with 3 pin easy interface connector.
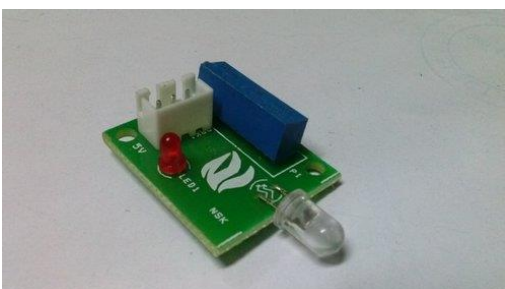- Input Voltage +5VDC

FIG . Fire sensor

Fire Sensor circuit

This fire sensor circuit exploits the temperature sensing property of an ordinary signal diode IN 34 to detect heat from fire. At the moment it senses heat, a loud alarm simulating that of Fire brigade will be produced. The circuit is too sensitive and can detect a rise in temperature of 10 degree or more in its vicinity. Ordinary signal diodes like IN 34 and OA 71 exhibits this property and the internal resistance of these devices will decrease when temperature rises.

The fire sensor circuit is too sensitive and can detect a rise in temperature of 10 degree or more in its vicinity. Ordinary signal diodes like IN 34 and OA 71 exhibits this property and the internal resistance of these devices will decrease when temperature rises. In the reverse biased mode, this effect will be more significant. Typically the diode can generate around 600 milli volts at 5 degree centigrade. For each degree rise in temperature; the diode generates 2 mV output voltage. That is at 5 degree it is 10 mV and when the temperature rises to 50 degree, the diode will give 100 milli volts. This voltage is used to trigger the remaining circuit. Transistor T1 is a temperature controlled switch and its base voltage depends on the voltage from the diode and from VR and R1. Normally T1 conducts (due to the voltage set by VR) and LED glows. This indicates normal temperature.

When T1 conducts, base pf T2 will be grounded and it remains off to inhibit the Alarm generator. IC UM 3561 is used in the circuit to give a Fire force siren. This ROM IC has an internal oscillator and can generate different tones based on its pin connections. Here pin 6 is shorted with the Vcc pin 5 to get a fire force siren. When the temperature near the diode increases above 50 degree, it conducts and ground the base of T1. This makes T1 off and T2 on. Alarm generator then gets current from the emitter of T2 which is regulated by ZD to 3.1 volt and buffered by C1.Resistor R4 (220K) determines the frequency of oscillation and the value 220K is a must for correct tone. To set the fire sensor circuit, keep a lighted candle near the diode and wait for 1 minute. Slowly adjust VR till the alarm sounds. Remove the heat .After one minute, alarm will turns off. VR can be used for further adjustments for particular temperature levels.

Fire Sensor Circuit Diagram

Fig: Fire sensor circuit diagram

HT9170B

DTMF RECIEVER

The HT9170B/D are Dual Tone Multi Frequency (DTMF) receivers integrated with digital decoder and band split filter functions as well as power-down mode and inhibit mode operations. Such devices use digital counting techniques to detect and decode all the 16 DTMF tone pairs into a 4-bit code output.

Highly accurate switched capacitor filters are implemented to divide tone signals into low and high group signals. A built-in dial tone rejection circuit is provided to eliminate the need for pre-filtering.

The DTMF (Dual Tone Multiple Frequency) application is associated with digital telephony, and provides two selected output frequencies (one high band, one low band) for a duration of 100 ms. A benchmark subroutine has been written for the COP820C/840C microcontrollers, and is outlined in detail in this application note. This DTMF subroutine takes 110 bytes of COP820C/840C code, consisting of 78 bytes

of program code and 32 bytes of ROM table. The timings in this DTMF subroutine are based on a 20 MHz

COP820C/840C clock, giving an instruction cycle time of 1 ms.The matrix for selecting the high and low band frequencies associated with each key is shown in Figure 1 . Each key is uniquely referenced by selecting one of the four low band frequencies associated with the matrix rows, coupled with selecting one of the four high band frequencies associated with the matrix columns. The low band frequencies are 697,

770, 852, and 941 Hz, while the high band frequencies are 1209, 1336, 1477, and 1633 Hz. The DTMF subroutine assumes that the key decoding is supplied as a low order hex digit in the accumulator. The COP820C/840C DTMF subroutine will then generate the selected high band and low

band frequencies on port G output pins G3 and G2 respectively for a duration of 100 ms. The COP820C/840C each contain only one timer. The problem is that three different times must be generated to satisfy the DTMF application. These three times are the periods of the two selected frequencies and the 100 ms duration period. Obviously the single timer can be used to generate any one (or possibly two) of the required times, with the program having to generate the other two (or one) times. The solution to the DTMF problem lies in dividing the 100 ms time duration by the half periods (rounded to the nearest
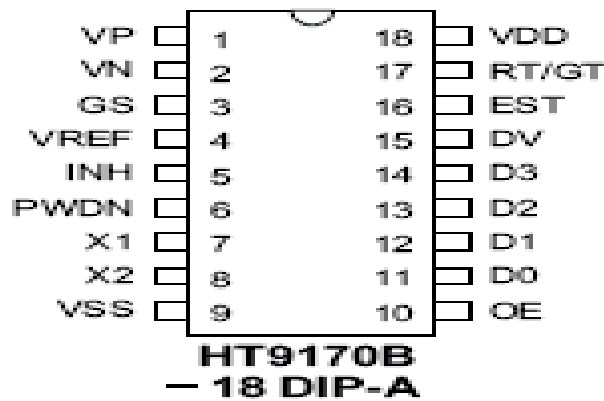
micro second) for each of the eight frequencies, and then examining the respective high band and low band quotients and remainders. The results of these divisions are detailed in Table I. The low band frequency quotients range from 139 to 188, while the high band quotients range from 241 to 326.

The observation that only the low band quotients will each fit in a single byte dictates that tshe high band frequency be produced by the 16 bit (2 byte) COP820C/840C timer running in PWM (Pulse Width Modulation) Mode. TL/

2.8.1 FEAUTURES:

- Operating voltage: 2.5V~5.5V

- Minimal external components

- No external filter is required

- Low standby current (on power down mode)

- Excellent performance

- Tristate data output for MCU interface

- 3.58MHz crystal or ceramic resonator

- 1633Hz can be inhibited by the INH pin

- HT9170B: 18-pin DIP package

PIN ASSIGMENT:

```
          VP  ┌1          18┐ VDD
          VN  ┌2          17┐ RT/GT
          GS  ┌3          16┐ EST
        VREF  ┌4          15┐ DV
         INH  ┌5          14┐ D3
        PWDN  ┌6          13┐ D2
          X1  ┌7          12┐ D1
          X2  ┌8          11┐ D0
         VSS  ┌9          10┐ OE
```

HT9170B
— 18 DIP-A

PIN DESCRIPTION:

| Pin Name	I/O	Internal Connection	Description
VP	I	Operational Amplifier	Operational amplifier non-inverting input
VN	I		Operational amplifier inverting input
GS	O		Operational amplifier output terminal
VREEF	O	VREF	Reference voltage output, normally $V_{DD}/2$
X1	I	oscillator	The system oscillator consists of an inverter, a bias resistor and the necessary load capacitor on chip.
X2	O		A standard 3.579545MHz crystal connected to X1 and X2 terminals implements the oscillator function.
PWDN	I	CMOS IN Pull-low	Active high. This enables the device to go into power down mode and inhibits the oscillator. This pin input is internally pulled down.
INH	I	CMOS IN Pull-low	Logic high. This inhibits the detection of tones representing characters A, B, C and D. This pin input is internally pulled down.
VSS	—	—	Negative power supply, ground
OE	I	CMOS IN Pull-high	D0~D3 output enable, high active
D0~D3	O	CMOS OUT Tristate	Receiving data output terminals OE="H": Output enable OE="L": High impedance
DV	O	CMOS OUT	Data valid output When the chip receives a valid tone (DTMF) signal, the DV goes high; otherwise it remains low.
EST	O	CMOS OUT	Early steering output (see Functional Description)
RT/GT	I/O	CMOS IN/OUT	Tone acquisition time and release time can be set through connection with external resistor and capacitor.
VDD	—	—	Positive power supply, 2.5V~5.5V for normal operation

BLOCK DIAGRAM:
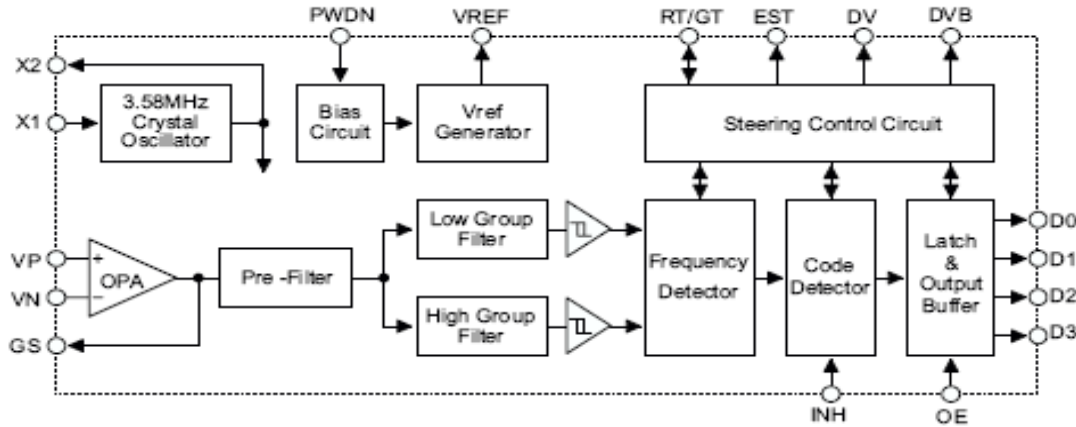
FUNCTIONAL DESCRIPTION:

The HT9170B/D tone decoders consist of three band pass filters and two digital decode circuits to convert a tone (DTMF) signal into digital code output.

An operational amplifier is built-in to adjust the input signal. The pre-filter is a band rejection filter, which reduces the dialing tone from 350Hz to 400Hz.

The low group filter filters low group frequency signal output whereas the high group filter filters high group Frequency signal output. A zero-crossing detector with follows each filters output hysteretic. When each signal amplitude at the output exceeds the specified level, it is transferred to full swing logic signal.

When input signals are recognized to be effective, DV becomes high, and the correct tone code (DTMF) digit is transferred.
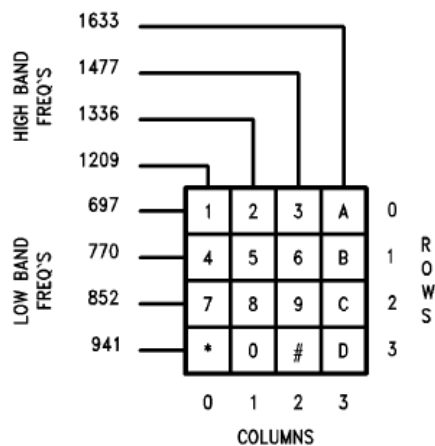

Steering control circuit:

The steering control circuit is used for measuring the effective signal duration and for protecting against drop out of valid signals. It employs the analog delay by external RC time-constant controlled by EST.

The EST pin is normally low and draws the RT/GT pin to keep low through discharge of external RC. When a valid tone input is detected, EST goes high to charge RT/GT through RC.

When the voltage of RT/GT changes from 0 to VTRT (2.35V for 5V supply), the input signal is effective, and the code detector will create the correct code. After D0~D3 are completely latched, DV output becomes high. When the voltage of RT/GT falls down from VDD to VTRT (i.e. when there is no input tone), DV output becomes Low, and D0~D3 keeps data until a next valid tone input is produced. By selecting adequate external RC value, the minimum acceptable input tone duration (tACC) and the minimum acceptable inter-tone rejection (tIR) can be set. External

Components (R, C) are chosen by the formula.

TIMING DIAGRAM
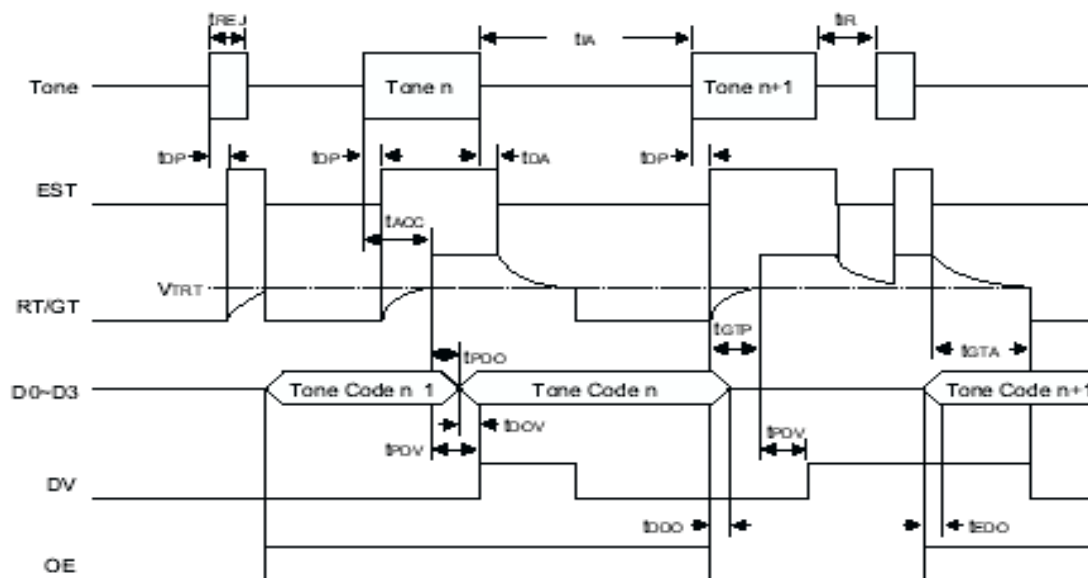
2.8.4 Applications

• PABX

• Central office

• Mobile radio

• Remote control

• Remote data entry

• Call limiting