CHAPTER 1

INTRODUCTION TO PROJECT

1.1 INTRODUCTION

The idea of designing a new system for the street-light that do not consume huge amount of electricity and illuminate large areas with the highest intensity of light is concerning each engineer working in this field. Providing street lighting is one of the most important and expensive responsibilities of a city. Lighting can account for 10–38% of the total energy bill in typical cities worldwide. Street lighting is a particularly critical concern for public authorities in developing countries because of Its strategic importance for economic and social stability. Inefficient lighting wastes significant financial resources every year, and poor lighting creates unsafe conditions. Energy efficient technologies and design mechanism can reduce cost of the street lighting drastically. Manual control is prone to errors and leads to energy wastages and manually dimming during mid-night is impracticable. Also, dynamically tracking the light level is manually impracticable. The current trend is the introduction of automation and remote management solutions to control street lighting.

In this Document LDR and IR is used as sensor. The main objective is to provide an efficient & energy saving lighting system by evaluating the outside lighting condition and then adjusting the lights accordingly. The circuit is consisting of ARDUINO UNO, IR sensor, LDR sensor, LCD, LED.

1.2 Real Time Systems:

One subclass of embedded is worthy of an introduction at this point. As commonly defined, a real-time system is a computer system that has timing constraints. In other words, a real-time system is partly specified in terms of Its ability to make certain calculations or decisions in a timely manner. These important calculations are said to have deadlines for completion. And, for all practical purposes, a missed deadline is just as bad as a wrong answer. The issue of what if a deadline is missed is a crucial one. For example, if the real-time system is part of an airplane's flight control system, it is possible for the lives of the passengers and crew to be endangered by a single missed deadline. However, if instead the system is involved in satellite communication, the damage could be limited to a single corrupt data packet. The more severe the consequences, the more likely it will be said that the deadline is "hard" and thus, the system is a hard-real-time system. Real-time systems at the other end of this discussion are said to have "soft" deadlines.

All the topics and examples presented in this book are applicable to the designers of real-time system who is more delight in his work. He must guarantee reliable operation of the software and hardware under all the possible conditions and to the degree that human lives depend upon three systems' proper execution, engineering calculations and descriptive paperwork.

1.3 Application Areas

Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation, military and so on. Applications of embedded systems are applicable in areas like space, communication, transportation, robotic systems, home appliances, etc. This article is intended to give information about the embedded system applications. Based on the performance requirements, these systems are categorized into four types such as standalone, networked, mobile and real time embedded systems.

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.

1.4 Office automation:

The office automation products using embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

1.5 Industrial automation:

Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence must be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

1.6 Security:

Security of persons and information has always been a major issue. We need to protect our homes and offices; and, the information we transmit and store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are embedded systems. Encryption devices are nearly 99 per cent of the processors that are manufactured end up in~ embedded systems. Embedded systems find applications in every industrial segment- consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defence, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines. Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

1.5 What are microcontrollers and what are they used for?

A microcontroller is a tiny, affordable and self-contained computer-on-a-chip which will be used as an embedded system. It’s a pc on-a-chip optimized to manage electrical gadgets. It is meant significantly for specific tasks like an exact system. A microcontroller is often abbreviated as µC, or MCU. Also, a micro-controller could be a fraction of a set-in system that is essentially a whole card. A fixed-in system could be a computing system supposed to hold out one or a lot of functions over and all over again with real-time estimate limits. It is embedded as a part of a full machine typically enumeration hardware and motorized parts in addition.

Examples of microcontrollers are the Arduino, Intel, PIC and Motorola. Microcontrollers that are usually incorporated in toys, cars, appliances and workplace machines are gears that amalgamate variety of constituents of a micro-processor system on a solo chip.

A few microcontrollers might utilize four-bit expressions and work on clock rate frequencies that typically include:

- An 8- or 16-bit chip
- tiny amount of RAM.
- Programmable computer memory (ROM) and non-volatile storage (Flash memory)
- Serial & parallel I/O.
- Timers.
- Analog to Digital and Digital to Analog conversion
- Programmable computer memory and non-volatile storage.

Chapter-2

PROPOSED SYSTEM

The main aim of this project is that it saves energy by putting on the lights of the system only when the system detects movement of vehicle. The system switches on the street light ahead of the vehicle and switches off the trailing lights simultaneously. In order to detect movement of vehicles, sensors are used. The system automatically puts on the lights that are ahead of the vehicle detected and as soon as the vehicle moves ahead, the trailing lights are switched off. This is better than the existing system where the street lights are kept on always unlike this system where the street lights are put on only when movement of vehicle is detected which helps in saving lot of energy. During day time these lights are dim as this system has the capability to sense external lights. Thus, this system senses the external light and then accordingly puts ON or OFF the street lights through microcontroller. The sensors sense the vehicle movements and send it to an Arduino that initiates commands for switching the lights ON/OFF. The system basically consists of a LDR, power supply, IR sensor, LCD, and Arduino Uno. The pictorial representation of system is given below

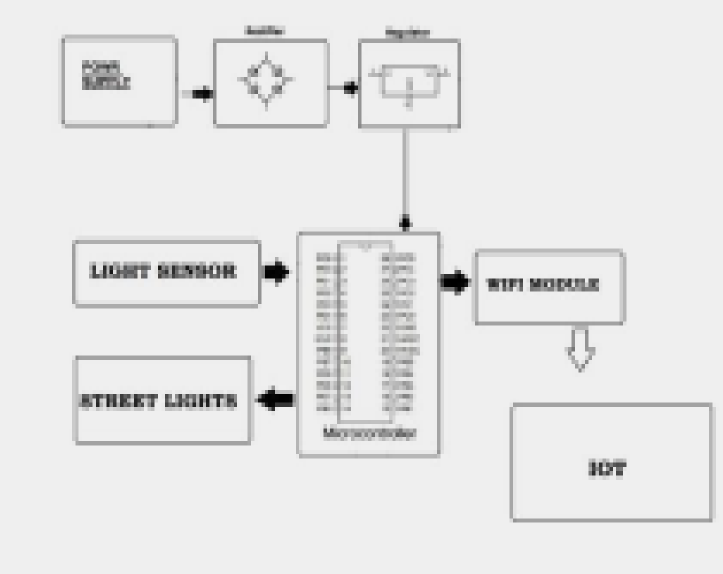


Fig:2.0 Proposed Circuit Diagram

CHAPTER-3

PROJECT DESCRIPTION

3.1: BLOCK DIAGRAM:

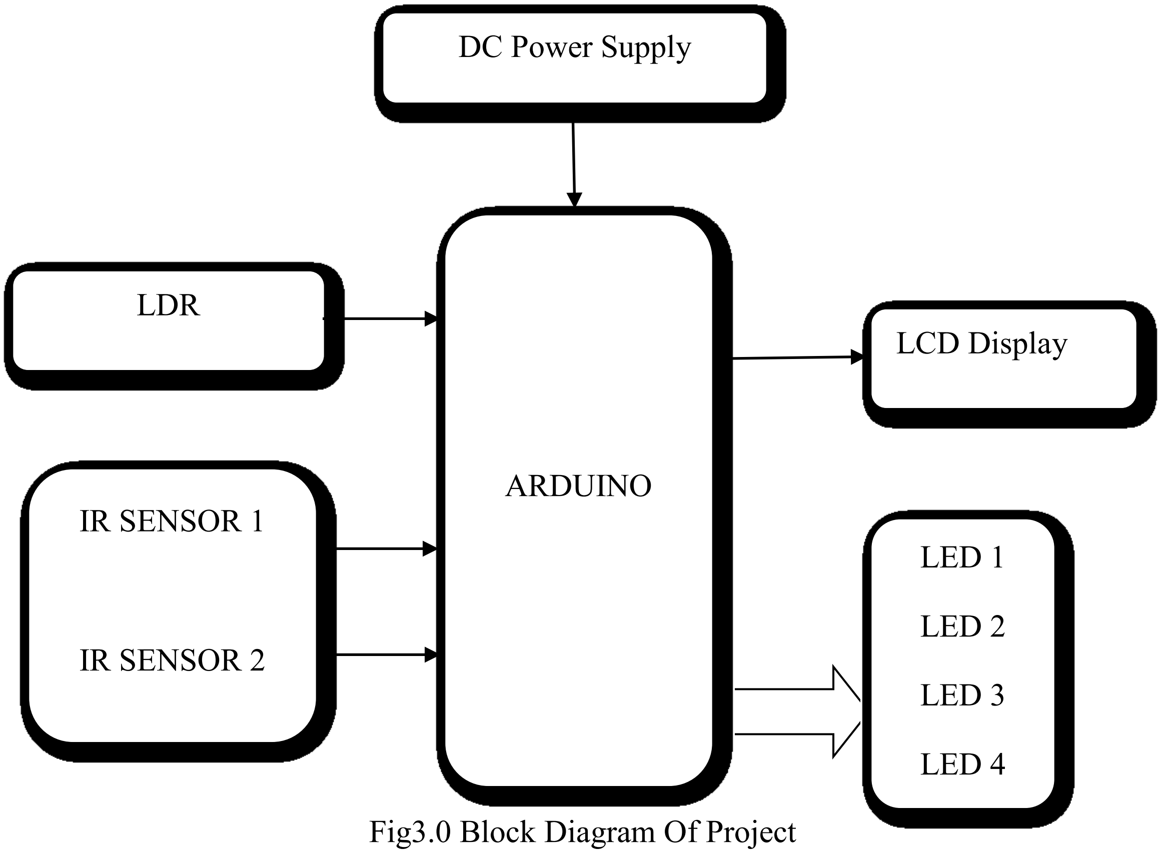


Fig3.0 Block Diagram Of Project

PROJECT WORKING:

The working of the model is very simple. The supply is given Through the power jack. From the Arduino we take 5v supply and connect it to one of the terminals of photo resistor and other end is connected to a resistor of 10k which acts as a voltage divider and then final connected to ground. The output is given by output pin13 of the Arduino which is connected to the led through a220ohm resistor. The other end of LED is perfectly grounded. The LDR senses the amount of light in the atmosphere at that moment of time and accordingly sends the data is to Arduino. The Arduino converts the data received into various discrete levels .For example from 0 to 1023 discrete levels for a given data then 0 represents maximum darkness and 1023 represents maximum brightest so light is received is converted into one of the discrete value from 0 to 1023.Now depending upon the discrete value that we get (0 to 1023) we adjust the output voltage accordingly from 0 to 5v.So, when complete darkness (night time) that is discrete level 0 than the output is 5v as a result LED is brightest or when partially dark (dawn/evening) that is discrete level of 512 then the output is 2.5 v as a result LED is half of the maximum brightest or when completely bright that is discrete level 1023 then the output voltage is 0v as a result LED switched off. Thus, the LED not only just automatically switches on and off but also adjusts the amount of light emitted according to the outside condition. The usage of such kind of application in the headlights of cars, park lights, street lights is very useful.

3.2 Schematic Diagram

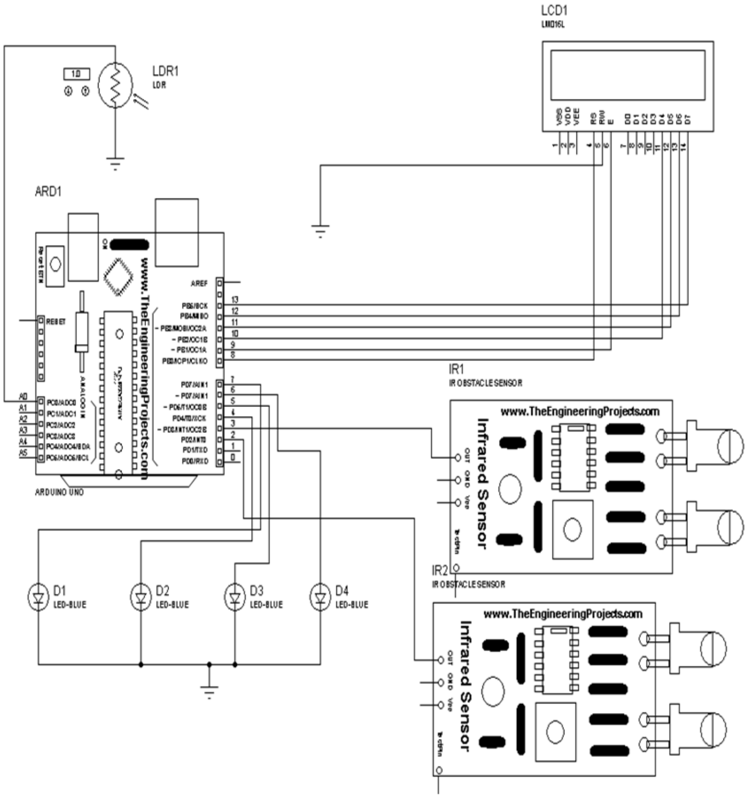


Fig:3.1 Schematic Circuit Diagram

CHAPTER-4

HARDWARE REQUIREMENT

4.1 EMBEDED SYSTEM

An embedded system is one kind of a computer system mainly designed to perform several tasks like to access, process, and store and also control the data in various electronics-based systems. Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits. Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too. The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighborhood traffic control systems, etc. Please follow the below link for Embedded system basics; block diagram, types, and applications.

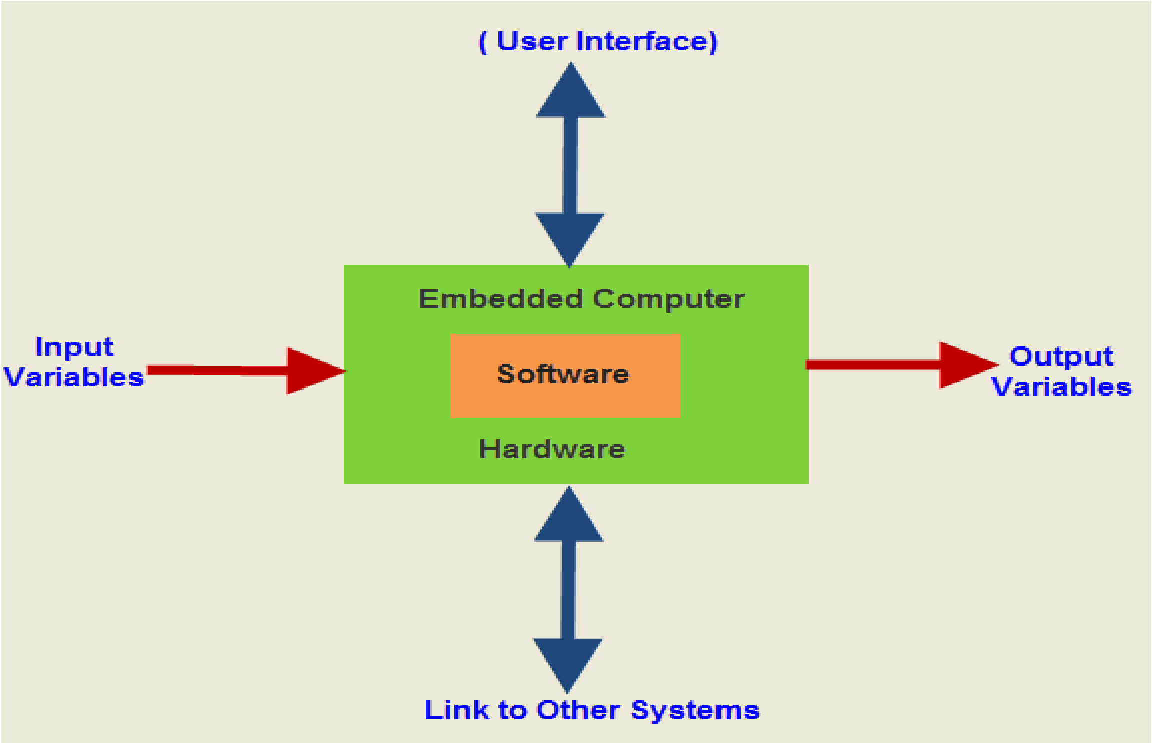


Fig 4.1: Embedded systems

Embedded gadget can be a mixture of hardware and software program system won't to deliver the goods one specific assignment. accomplice degree embedded gadget may be a microcontroller-primarily based, software gadget driven, reliable, time period gadget, self-sustaining, or human or network interactive, in operation on several bodily variables and in several environments and offered into an aggressive and fee acutely conscious market.

An embedded gadget is not a computing system that's used commonly for processing, now not a code on laptop or UNIX operating device, not a trendy enterprise or clinical utility. Excessive-end embedded & lower end embedded structures. High-stop embedded gadget - usually thirty-two, sixty-four Bit Controllers used with OS. Examples non-public digital Assistant and cellular telephones and so forth. decrease finish embedded systems - usually eight, 16-bit Controllers used with companion degree lowest in operation structures and hardware format designed for the unique reason. Examples little controllers and gadgets in our way of existence like washer, Microwave Ovens, anyplace they're embedded in this region we would like to discuss the function of simulation code, term systems and records acquisition in dynamic take a look at applications. Ancient trying out is remarked as "static" testing anyplace practicality of components is examined by using imparting illustrious inputs and degree outputs. These days there's additional stress to urge product to plug faster and cut back style cycle times.

This has caused a necessity for "dynamic" testing anyplace parts are tested whereas in use with the whole system – either real or simulated. Due to fee and safety issues, simulating the remainder of the device with term hardware is maximum well-liked to testing elements within the real machine.

The diagram proven in this slide is that the "V Diagram" it is commonly accustomed describe the event cycle. In the beginning advanced to encapsulate the appearance approach of code applications, absolutely extraordinary many diverse many alternatives' versions of this diagram may be found to provide an explanation for one of a kind product style cycle. here we've got were given shown one instance of any such diagram representing the look cycle of embedded control programs common to automotive, part and defence programs.

The coordinate axis of this diagram will be notion of due to the fact the level at that the system elements are concept of. Timely inside the improvement, the needs of the machine must be concept of. Due to the fact the gadget is cut up into sub-systems and elements, the approach turns into terribly low-degree all the manner right down to the cause of loading code onto man or woman processors. After elements are integrated and examined along till such time that the complete system will input final manufacturing checking out. Accordingly, the very best of the diagram represents the excessive-degree system study and also the bottom of the diagram represents an absolutely low-degree examine. Notes:

- V diagram describes several applications—derived from code development.
- Reason for form, each part of style needs a complimentary take a look at par. High-level to low-level read of application.
- This could be a simplified version.
- Loop back/ unvaried method, coordinate axis is time (sum up).

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the ‘firmware’. The embedded system architecture can be represented as a layered architecture. As Its name suggests, embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system, or it can be a part of a large system. An embedded system is a microcontroller or microprocessor-based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke. The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighborhood traffic control systems, etc. Please follow the below link for Embedded system basics; block diagram, types, and applications.

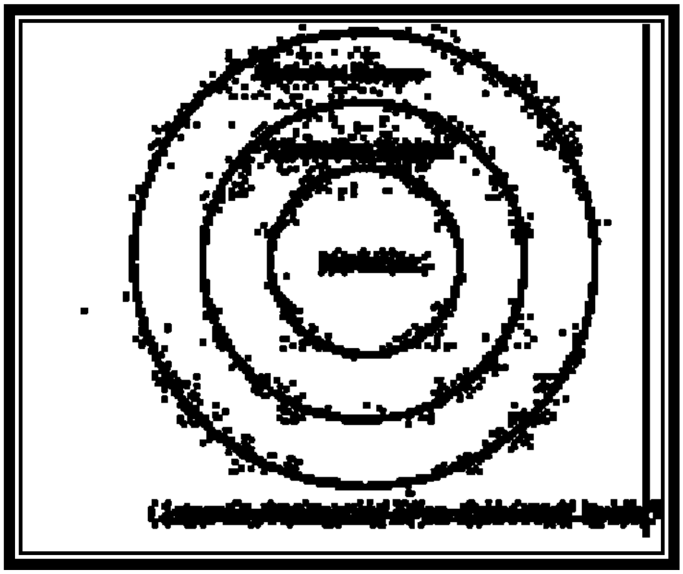


Fig 4.2: Layered architecture of an embedded system

The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system. For small appliances such as remote-control unit's, air conditioners, toys etc., there is no need for an operating system, and you can write only the software specific to that application. For applications involving complex processing, it is advisable to have an operating system. In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run for a long time you do not need to reload new software.

Now, let us see the details of the various building blocks of the hardware of an embedded system. As shown in Fig. the building blocks are.

- Central Processing Unit (CPU)
- Memory (Read-only Memory and Random-Access Memory)
- Input Devices
- Output devices
- Communication interfaces
- Application-specific circuitry.

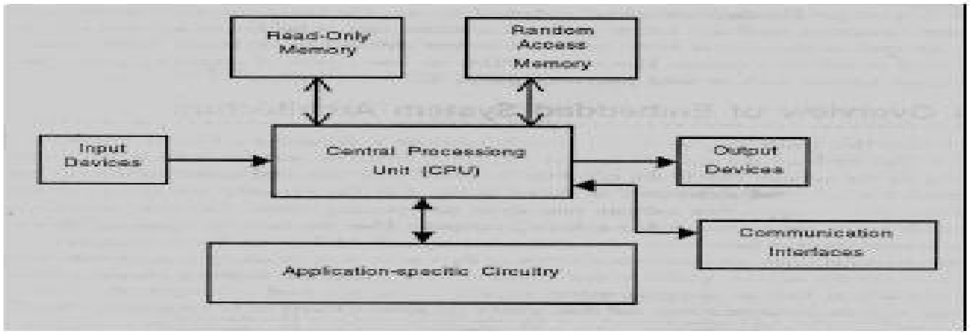


Fig: 4.3 Block diagram of hardware components of embedded system

4.1.1 Embedded System Basics

The embedded system basics are the combination of embedded system hardware and embedded system software.

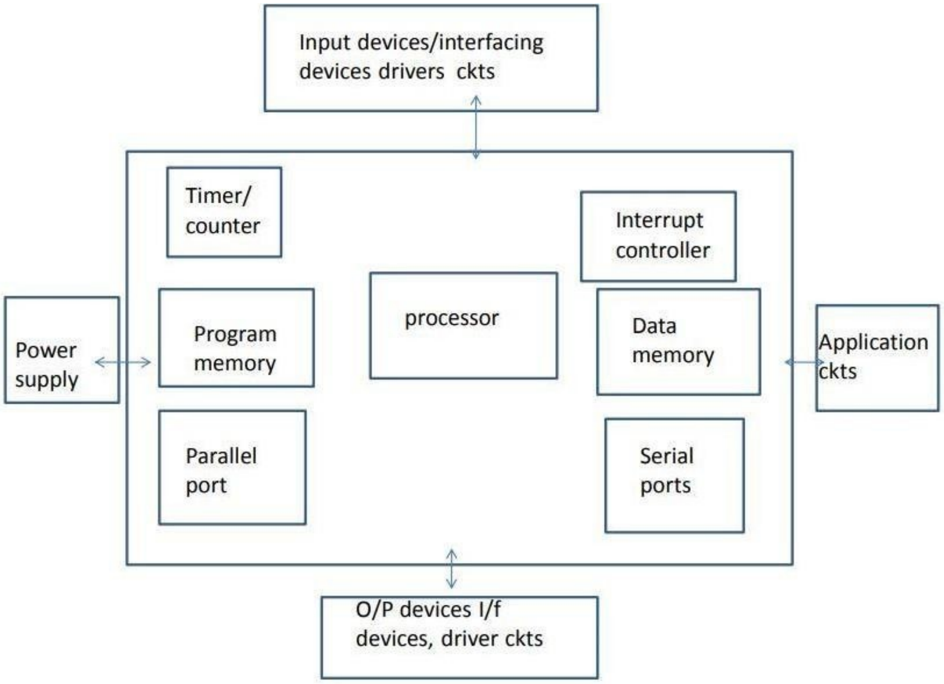


Fig 4.4: Block diagram of embedded system

4.1.2 Embedded System Hardware

An embedded system uses a hardware platform to perform the operation. Hardware of the embedded system is assembled with a microprocessor/microcontroller. It has the elements such as input/output interfaces, memory, user interface and the display unit. Generally, an embedded system comprises of the following

- Power Supply
- Timers
- Output/Output circuits
- Serial communication ports
- SASC (System application specific circuits)

4.1.3 Embedded System Software

The software of an embedded system is written to execute a particular function. It is normally written in a high-level setup and then compiled down to offer code that can be stuck within a non-volatile memory in the hardware. An embedded system software is intended to keep in view of the following three limits

- Convenience of system memory
- Convenience of processor's speed
- When the embedded system runs constantly, there is a necessity to limit power dissipation for actions like run, stop and wake up.

4.1.4 RTOS (Real Time Operating System)

A system which is essential to finish its task and send its service on time, then only it said to be a real time operating system. RTOS controls the application software and affords a device to allow the processor run. It is responsible for managing the different hardware resources of a personal computer and also host applications which run on the PC.

This operating system is specially designed to run various applications with an exact timing and a huge amount of consistency. Particularly, this can be significant in measurement & industrial automation systems where a delay of a program could cause a safety hazard.

4.1.5 Memory and Processors

The different kinds of processors used in an embedded system include Digital Signal Processor (DSP), microprocessor, RISC processor, microcontroller, ASSP processor, ASIP processor, and ARM processor. The different types of memories of an embedded system are given in the below chart.

4.1.6 Embedded System Characteristics

Generally, an embedded system executes a particular operation and does the similar continually. For instance: A pager is constantly functioning as a pager.

All the computing systems have limitations on design metrics, but those can be especially tight. Design metric is a measure of an execution features like size, power, cost and also performance.

It must perform fast enough and consume less power to increase battery life.

Several embedded systems should constantly react to changes in the system and also calculate particular results in real time without any delay. For instance, a car cruise controller; it continuously displays and responds to speed & brake sensors. It must calculate acceleration/de-accelerations frequently in a limited time; a delayed computation can consequence in let-down to control the car.

- It must be based on a microcontroller or microprocessor based.
- It must require a memory, as its software generally inserts in ROM. It does not require any secondary memories in the PC.
- It must need connected peripherals to attach input & output devices.
- An Embedded system is inbuilt with hardware and software where the hardware is used for security and performance and Software is used for more flexibility andfeatures.

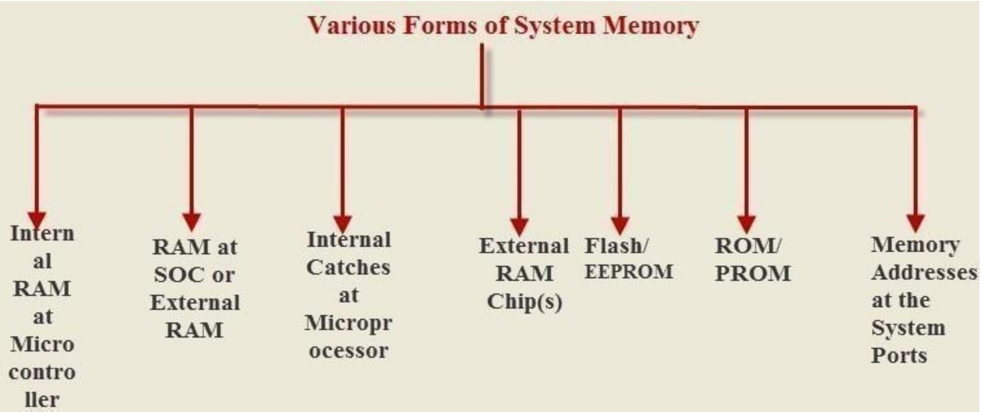

Various Forms of System Memory
Internal RAM at Microcontroller
RAM at SOC or External RAM
Internal Catches at Microprocessor
External RAM Chip(s)
Flash/ EEPROM
ROM/ PROM
Memory Addresses at the System Ports

Fig 4.5: various forms of system Memory

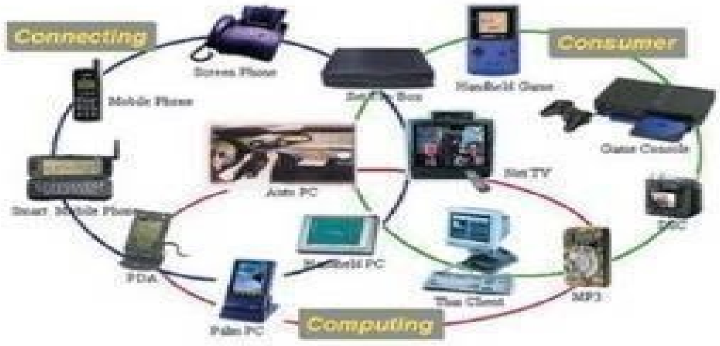


Fig 4.6: Embedded system application

4.1.7 Embedded System Applications

The applications of an embedded system basics include smart cards, computer networking, satellites, telecommunications, digital consumer electronics, missiles, etc.

- Embedded systems in automobiles include motor control, cruise control, body safety, engine safety, robotics in an assembly line, car multimedia, car entertainment, E-com access, mobiles etc.
- Embedded systems in telecommunications include networking, mobile computing, and wireless communications, etc.
- Embedded systems in smart cards include banking, telephone and security systems.
- Embedded Systems in satellites and missiles include defense, communication, and aerospace.
- Embedded systems in computer networking & peripherals include image processing, networking systems, printers, network cards, monitors and displays.
- Embedded Systems in digital consumer electronics include set-top boxes, DVDs, high definition TVs and digital cameras.

Thus, this is all about the basics of embedded system basics and applications. We all know that embedded systems are extremely fabulous systems that play avital role in many applications like equipment, industrial instrumentation, etc.

4.2 Power Supply:

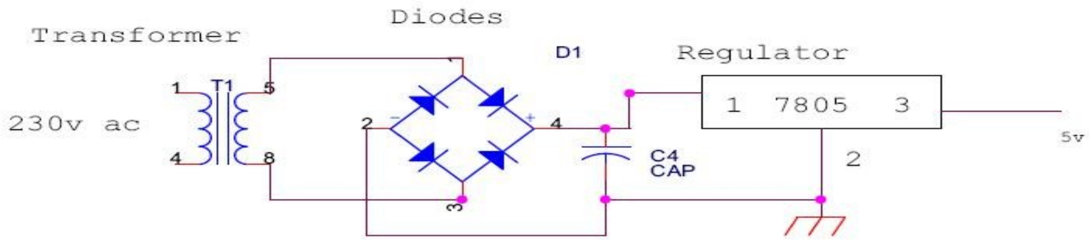


Figure 4.8: Power Supply

Power provides unit consists of following unit’s:

i) Step down Transformer (Electrical device).
ii) Rectifier unit
iii) Input filter
iv) Regulator unit
v) Output filter

4.2.1 Step-down Transformer (Electrical device):

The Step-down electrical device is employed to step down the most provide voltage from 230V AC to lower price. This 230 AC voltage cannot be used directly, so it's stepped down. The electrical device consists of primary and secondary coils. to scale back or step down the voltage, the electrical device is meant to contain less variety of turns in its secondary core. The output from the secondary winding is additionally AC undulation. So, the conversion from AC to DC is important. This conversion is achieved by mistreatment the Rectifier Circuit/Unit.

Step down transformers will step down incoming voltage that permit’s you to possess the proper voltage input for your electrical wants. For instance, if our instrumentality has been given for input voltage of twelve volts, and therefore the main power provide is 230 volts, we are going to want a step-down electrical device that decreases the incoming electrical Voltage to be compatible together with your twelve-volt instrumentality.

4.2.2 Rectifier Unit:

The Rectifier circuit is employed to convert the AC voltage into its corresponding DC voltage. The foremost vital and easy device employed in Rectifier circuit is that the diode. The straightforward operate of the diode is to conduct once forward biased and to not conduct in reverse bias. Currently we have a tendency to are mistreatment 3 sorts of rectifiers. They are

1. Half-wave rectifier
2. Full-wave rectifier
3. Bridge rectifier

Half-wave rectifier: In full wave rectification, either the positive or negative half the AC wave is passed, whereas the opposite full wave rectification is blocked. As a result of just one half the input undulation reaches the output, it's terribly inefficient if used for power transfer. Half-wave rectification is achieved with one diode in a very one section provides, or with 3 diodes in a very three-phase provide.

Full-wave rectifier: A rectifier converts the complete of the input undulation to at least one of constant polarity (positive or negative) at its output. Full-wave rectification converts each polarities of the input undulation to DC (direct current), and is a lot of economical. However, in a very circuit with a non-center broach electrical device, four diodes are needed rather than the one required for half-wave rectification. A rectifier uses a diode bridge, made from four diodes, like this:

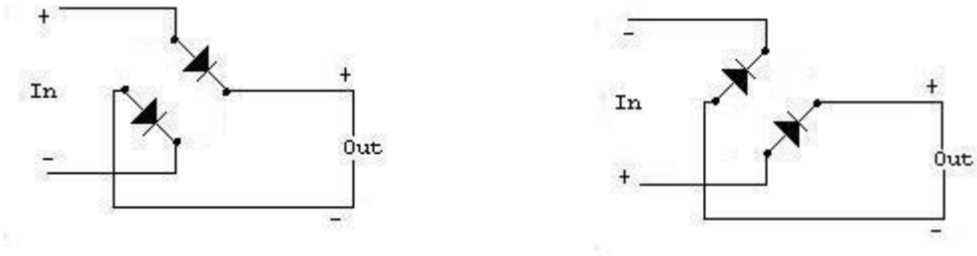


Figure: 4.9: Full Wave Rectifier

So, if we have a tendency to feed our AC signal into a full wave rectifier, we'll see each half of the wave higher than zero Volts. Since the signal passes through 2 diodes, the voltage out is lower by 2 diode drops, or 1.2 Volts.

At first, this could look even as confusing because the unidirectional streets of Bean Town. The factor to appreciate is that the diodes add pairs. because the voltage of the signal flips back and forth, the diodes Shepard this to invariably flow within the same direction for the output.

Here's what the circuit appears like to the signal because it alternates:

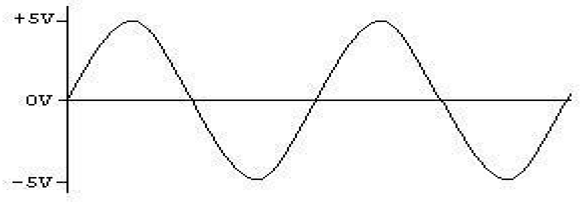


Figure: 4.10: Input wave form (AC wave)

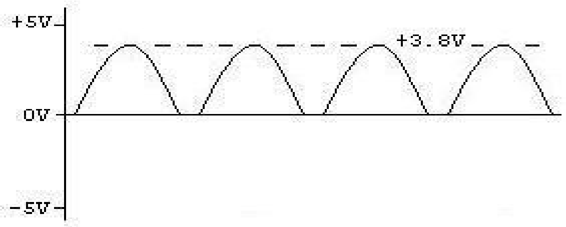
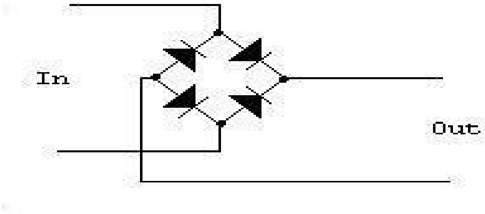


Figure: 4.11: Output wave form (Pulsating DC)

If we're interested by practice the rectifier as a DC power offer, we'll add a smoothing condenser to the output of the diode bridge. Bridge rectifier: A bridge rectifier makes use of 4 diodes in an exceedingly bridge arrangement to realize full-wave rectification. This can be a wide used configuration, each with individual diodes wired as shown and with single element bridges wherever the diode bridge is wired internally.



4.12: Bridge Rectifier with diodes back-to-back connection

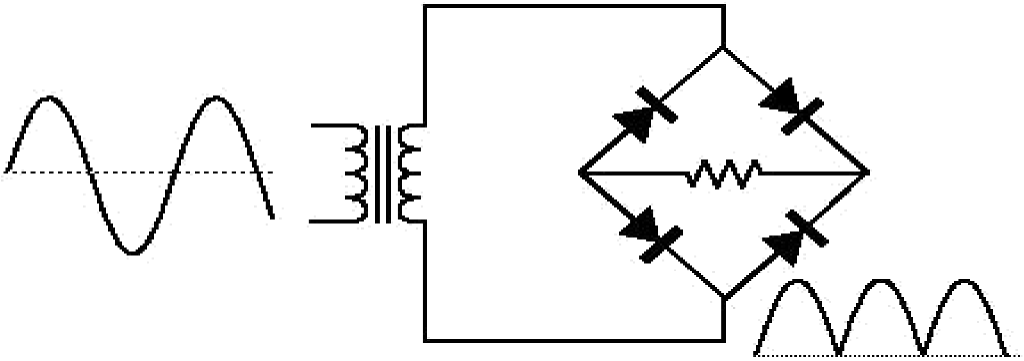


Figure: 4.13: Bridge Rectifier with AC input

A diode bridge or bridge rectifier is a briefing of 4 diodes in an exceedingly bridge configuration that gives similar/same polarity of output voltage for either terminal/polarity of input voltage. Once utilized in Its commonest application, for conversion of electricity (AC) input into DC (DC) output, it's called a bridge rectifier. A bridge rectifier provides full-wave rectification from a two-wire AC input, leading to lower value and weight as compared to a center-tapped electrical device style.The Forward Bias is achieved by connecting the diode’s positive with positive of the battery and negative with battery’s negative. The economical circuit used is that the Full wave Bridge rectifier circuit. The output voltage of the rectifier is in rippled type, the ripples from the obtained DC voltage area unit removed exploitation different circuits offered. The circuit used for removing the ripples is termed Filter circuit.

4.2.3 Input Filter:

Capacitor’s area unit used as filter. The ripples from the DC voltage area unit removed and pure DC voltage is obtained. And additionally, these capacitors area unit won’t to cut back the harmonics of the input voltage. The first action performed by capacitance is charging and discharging. It charges in positive 0.5 cycle of the AC voltage and it'll discharge in negative 0.5 cycles. Thus, it permit’s solely AC voltage and doesn't permit the DC voltage. This filter is fastened before the regulator. So, the output is free from ripples. There are unit 2 kinds of filters. They are

1. Low pass filter

2. High pass filter

Low pass filter:

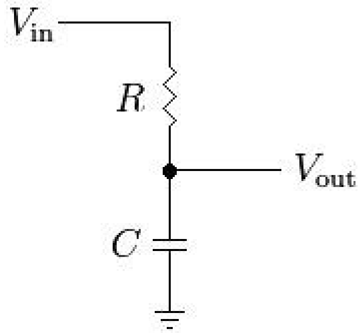


Figure: 4.14: Low Pass Filter

One straightforward electrical device which will function a low-pass filter consists of an electrical device asynchronous with a load, and a capacitance in parallel with the load. The capacitance exhibit's electrical phenomenon, and blocks low-frequency signals, inflicting them to travel through the load instead. At higher frequencies the electrical phenomenon drops, and therefore the capacitance effectively functions as a brief circuit. The mix of resistance and capacitance provides you the time constant of the filter τ = RC (represented by the Greek letter tau). The break frequency, additionally referred to as the turnover frequency or cut-off frequency (in hertz), is set by the time constant: or equivalently (in radians per second):

One way to grasp this circuit is to target the time the capacitance takes to charge. It takes time to charge or discharge the capacitance through that resistor:

● At low frequencies, there's lots of time for the capacitance to rouse to much a similar voltage because the input voltage.

● At high frequencies, the capacitance solely has time to rouse tiny low quantity before the input switches direction. The output goes up and down solely tiny low fraction of the quantity the input goes up and down.

● It ought to be noted that the capacitance isn't associate degree "on/off" object (like the block or pass fluidic clarification above). The capacitance can variably act between these 2 extremes. It's the betoken plot and frequency response that show this variability.

High pass filter:

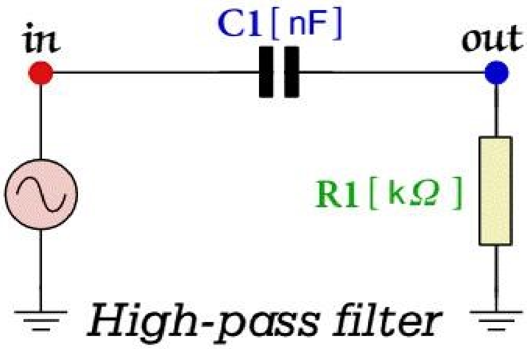


Figure: 4.15: High Pass Filter

The higher than circuit diagram illustrates a straightforward 'RC' high-pass filter. we must always realize that the circuit passes 'high' frequencies fairly well, however attenuate 'low' frequencies. thus, it's helpful as a filter to dam any unwanted low frequency elements of a posh signal while passing higher frequencies. Circuits like this square measure used quite a ton in natural philosophy as a 'D.C. Block' - i.e., to pass a.m. signals however forestall any D.C. voltages from obtaining through.

The basic quantities that describe this circuit square measure the same as those used for the Low Pass Filter. In effect, this circuit is simply a straightforward low-pass filter with the elements swapped over.

$$T=R1C1 \longrightarrow \textbf{eqn1}$$

The action of the circuit may also be delineated in terms of a connected amount, the flip over Frequency, f0, that encompasses a worth as with the low-pass filter, the circuit's behaviour we will be understood as arising thanks to the time taken to alter the capacitor's charge after we alter the applied input voltage. It continually takes a finite (i.e., non-zero) time to alter the quantity of charge keep by the condenser. Thus, it takes time to alter the electric potential across the condenser. As a result, any abrupt amendment within the input voltage produces an analogous abrupt amendment on the opposite aspect of the condenser. This produces a voltage across the resistance and causes a current to flow through it, charging the condenser till all the voltage falls across it rather than the resistance. The result's that steady (or slowly varying) voltages seem largely across the condenser and fast changes seem largely across the resistance. The Voltage Gain:

eqn2

$$\longrightarrow$$

The phase/section Delay:

$$\phi = \mathrm{ArcTan}\{ f_0/f \}$$

⟶ **eqn3**

Try mistreatment the higher than experimental system to gather results and plot a graph of however the voltage gain, Av, (and the section change) rely on the input frequency and if we have a tendency to check result agrees with the higher than formulae. Compare this with a low-pass filter that uses a similar element value and you ought to see that they provide 'opposite' results. within the high-pass filter, the output wave form 'leads' the input wave form - i.e., it peaks before the input.

4.2.4 Regulator Unit:

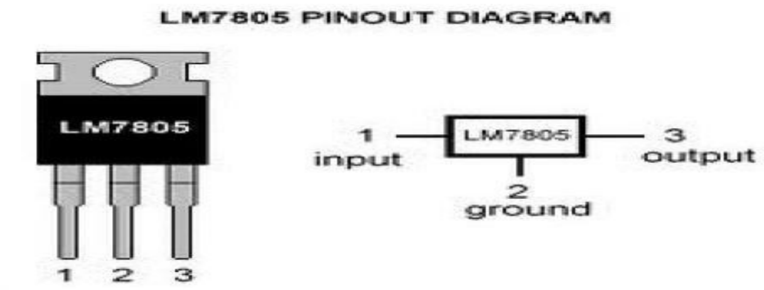


Figure: 4.16: 7805 Regulator along with pin diagram

Regulator regulates the output voltage to be continually constant. The output voltage is maintained regardless of the fluctuations within the input AC voltage. As and so the AC voltage changes, the DC voltage conjointly changes. So, to avoid this Regulators square measure used. Conjointly once the interior resistance of the facility provide is larger than thirty ohms, the output gets affected. So, this may be with success reduced here. The regulators square measure chiefly classified for low voltage and for top voltage. Additional they'll even be classified as:

i) Positive
regulator
1) Input pin/First
pin

2) Ground pin /Middle pin

3) Output pin/last or 3rd pin

It regulates the positive voltage.

ii) Negative regulator

1) Ground pin /first pin

2) Input pin/ middle pin

3) Output pin/last pin

It regulates the negative voltage.

4.3 Arduino Uno

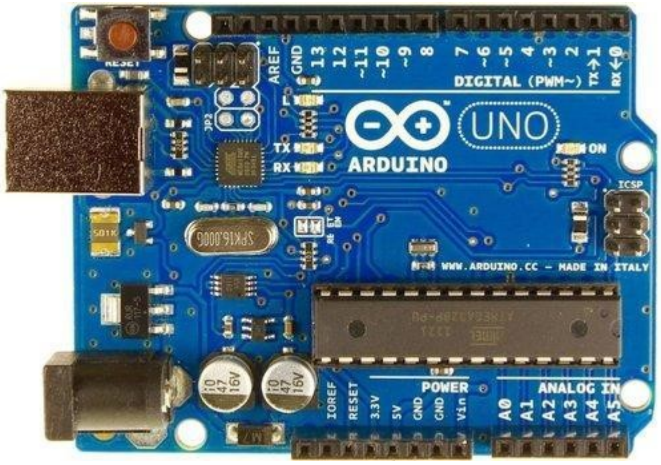


Fig4.17: Arduino Uno

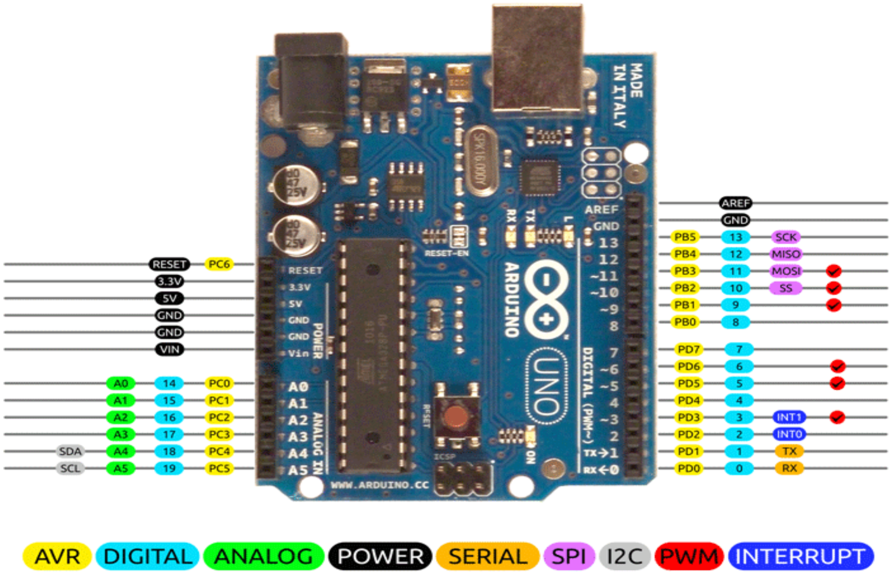
The **Arduino Uno** is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo.The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software The Uno board is the first in a series of

USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

4.3.1General pin functions

- **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.
- **VIN**: The input voltage to the Arduino/Genuino board when it is 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 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- **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/Genuino 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 to work with the 5V or 3.3V.
- **Reset**: Typically used to add a reset button to shields that block the one on the board



2014 by Bouni
Photo by Arduino.cc

Fig 4.18: Arduino pin function

4.3.2: Technical Application

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 can provide PWM output)
- PWM Pins: 6 (Pin # 3, 5, 6, 9, 10 and 11)
- UART: 1
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g
- ICSP Header: Yes

- Power Sources: DC Power Jack & USB Port

4.4 LCD (LIQUID CRYSTAL DISPLAY)

4.4.1 LIQUID CRYSTAL DISPLAY:

- LCD stands for liquid crystal displays. Digital display is finding wide unfold use substitution LEDs (seven phase LEDs or different multi-phase LEDs) thanks to the subsequent reasons:

  1.The declining costs of LCDs.

  2.The power to show numbers, characters and graphics. This is often in distinction to LEDs, that area unit restricted to numbers and a couple of characters.

  3.Incorporation of a refreshing controller into the digital display, thereby relieving the processor of the task of refreshing the digital display. In distinction, the crystal rectifier should be reinvigorated by the processor to stay displaying the info.

  4.Simple programming for characters and graphics.These parts area unit “specialized” for being employed with the microcontrollers, which implies that they can't be activated by customary IC circuits. They’re used for writing completely different messages on a miniature digital display.

  5.Incorporation of a refreshing controller into the digital display, thereby relieving the processor of the task of refreshing the digital display. In distinction, the crystal rectifier should be reinvigorated by the processor to stay displaying the info.

6.Simple programming for characters and graphics.These parts area unit “specialized” for being employed with the microcontrollers, which implies that they can't be activated by customary IC circuits. They’re used for writing completely different messages on a miniature digital display.

7.The power to show numbers, characters and graphics. This is often in distinction to LEDs, that area unit restricted to numbers and a couple of characters.

8.Incorporation of a refreshing controller into the digital display, thereby relieving the processor of the task of refreshing the digital display. In distinction, the crystal rectifier should be reinvigorated by the processor to stay displaying the info.

Function	Pin Number	Name	Logic State	Description
Ground	1	Vss	-	0V
Power supply	2	Vdd	-	+5V
Contrast	3	Vee	-	0 –Vdd
Control of operating	4	RS	0 1	D0 – D7 are interpreted as commands D0 – D7 are interpreted as data
	5	R/W	0 1	Write data (from controller to LCD) Read data (from LCD to controller)
	6	E	0 1 From 1 to 0	Access to LCD disabled Normal operating Data/commands are transferred to LCD
Data / commands	7	D0	0/1	Bit 0 LSB
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7 MSB

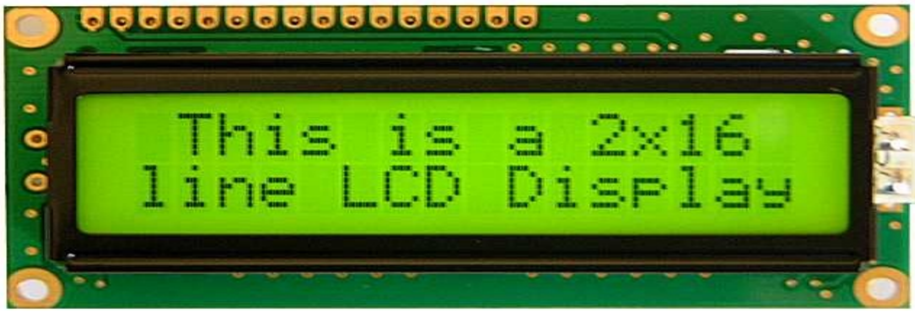


Fig4.14: Liquid crystal display

- A model represented here is for Its low value and nice potentialities most often utilized in follow. it's supported the HD44780 microcontroller (Hitachi) and might show messages in 2 lines withsixteencharacters every. It displays all the alphabets, Greek letters, and punctuation marks, mathematical symbols etc. additionally; it's attainable to show symbols that user makes informed Its own. Automatic shifting message on show (shift left and right), look of the pointer, backlight etc. area unit thought of as helpful characteristics

4.5 LCD screen:

LCD screen consists of 2 lines with sixteen characters every. Every character consists of 5x7 matrix.

Distinction on show depends on the facility provide voltage and whether or not messages area unit displayed in one or 2 lines. For that reason, variable voltage 0-Vdd is applied on pin marked as Vee. Trimmer potentiometer is typically used for that purpose. Some versions of displays have in-built backlight (blue or inexperienced diodes). Once used throughout operative, a resistance for current limitation ought to be used (like with any autoimmune disease diode).

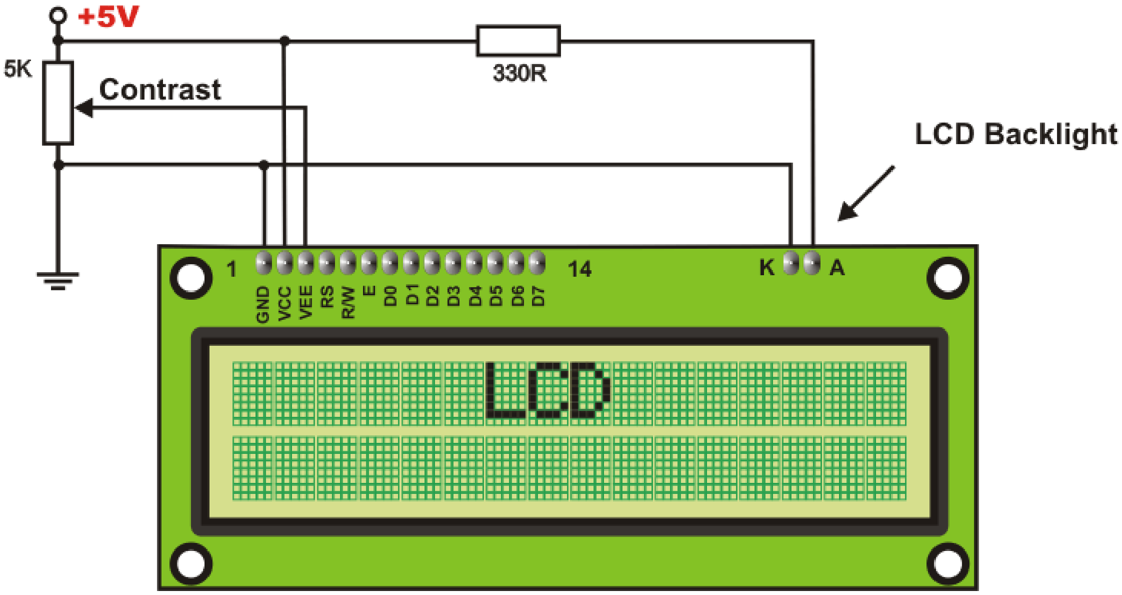


Fig4.15 : LCD pin connection

4.5.1 LCD Basic Commands

- All information transferred to LCD through outputs D0-D7 is going to be taken as commands or as information, that depends on logic state on pin RS:
- RS = one – B It's D0 - D7 area unit addresses of characters that ought to be displayed. In-built processor addresses in-built "map of characters" and displayscorresponding symbols. Displaying position is set by DDRAM address. This address is either antecedently outlined or the address of antecedental transferred character is mechanically incremented
- RS = zero – B It's D0 - D7 area unit commands that confirm show mode. List of commands that LCD acknowledges area unit given within the table below:

Command	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Execution Time
Clear display	0	0	0	0	0	0	0	0	0	1	1.64mS
Cursor home	0	0	0	0	0	0	0	0	1	x	1.64mS
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	40uS
Display on/off control	0	0	0	0	0	0	1	D	U	B	40uS
Cursor/Display Shift	0	0	0	0	0	1	D/C	R/L	x	x	40uS
Function set	0	0	0	0	1	DL	N	F	x	x	40uS
Set CGRAM address	0	0	0	1	CGRAM address						40uS
Set DDRAM address	0	0	1	DDRAM address							40uS
Read "BUSY" flag (BF)	0	1	BF	DDRAM address							-
Write to CGRAM or DDRAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	40uS
Read from CGRAM or DDRAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	40uS

4.5.2 Connection of LCD:

Depending on what percentage lines are used for affiliation/connection to the microcontroller, there are 8-bit and 4-bit alphanumeric display modes. The suitable mode is decided at the start of the method in an exceedingly part known as "initialization". Within the initial case, the info is transferred through outputs D0-D7 because it has been already explained. just in case of 4-bit crystal rectifier mode, for the sake of saving valuable I/O pins of the microcontroller, there are solely four higher b It's (D4-D7) used for communication, whereas different could also be left unconnected.

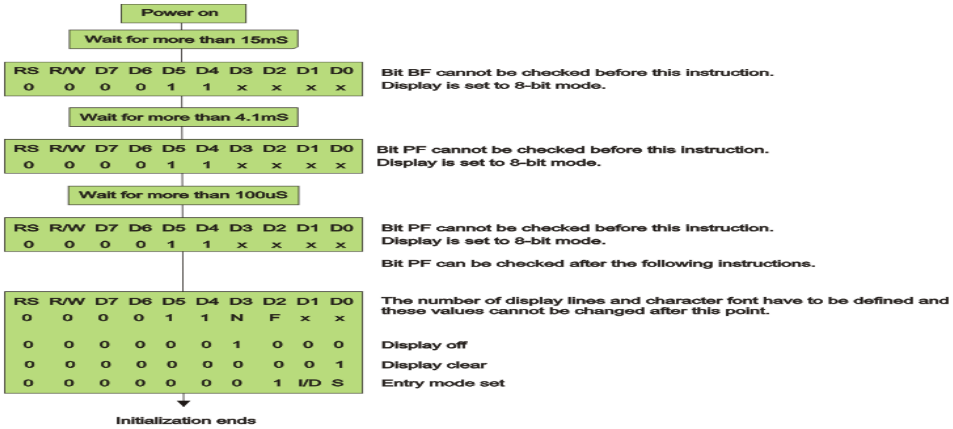
Consequently, every information is shipped to alphanumeric display in 2 steps: four higher b It’s are sent initial (that usually would be sent through lines D4-D7), four lower b It’s are sent after. With the assistance of initialization, alphanumeric display can properly connect and interpret every information received. Besides, with regards to the very fact that information is seldom scanning from alphanumeric display (data chiefly are transferred from microcontroller to LCD) an additional I/O pin could also be saved by easy connecting R/W pin to the bottom. Such saving has It’s worth. Even if message displaying is usually performed, it'll not be attainable to scan from busy flag since it's impractical to scan from show.

4.5.3 LCD INITIALIZATION FORMAT:

Once the facility offer is turned on, alphanumeric display is mechanically cleared. This method lasts for roughly 15mS. After that, show is prepared to control. The mode of operative is ready by default. This implies that:

- 1. Display/screen is cleared
- 2. Mode
- decilitre = one Communication through 8-bit interface
- N = zero Messages area unit displayed in one line
- F = zero Character font five x eight dots
- 3. Display/Cursor on/off
- D = zero show off
- U = zero pointer off
- B = zero pointer blink off
- 4. Character entry
- ID = one Addresses on show area unit mechanically incremented by one
- S = zero show shift off

Automatic reset is especially performed with none issues. Primarily however not always! If for any reason power offer voltage doesn't reach full price within the course of 10mS, show can begin perform utterly unpredictably? If voltage offer unit cannot meet this condition or if it's required to supply utterly safe operative, the method of format by that a brand-new reset sanctionative show to control usually should be applied. Algorithm in line with the format is being performed depends on whether or not affiliation to the microcontroller is thru 4- or 8-bit interface. All left over to be done then is to provide basic commands and of course- to show messages.



Power on

Wait for more than 15mS

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	1	x	x	x	x

Bit BF cannot be checked before this instruction.
Display is set to 8-bit mode.

Wait for more than 4.1mS

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	1	x	x	x	x

Bit PF cannot be checked before this instruction.
Display is set to 8-bit mode.

Wait for more than 100uS

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	1	x	x	x	x

Bit PF cannot be checked before this instruction.
Display is set to 8-bit mode.

Bit PF can be checked after the following instructions.

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	1	1	N	F	x	x	The number of display lines and character font have to be defined and these values cannot be changed after this point.
0	0	0	0	0	0	1	0	0	0	Display off
0	0	0	0	0	0	0	0	0	1	Display clear
0	0	0	0	0	0	0	1	I/D	S	Entry mode set

Initialization ends

Fig 4.16 : Procedure on 8-bit initialization.

4.5.4 CONTRAST CONTROL:

To have a transparent read of the characters on the alphanumeric display, distinction ought to be adjusted. To regulate the distinction, the voltage ought to be varied. For this, a planned is employed which may behave sort of a variable voltage device. Because the voltage of this planned is varied, the distinction of the alphanumeric display will be adjusted.

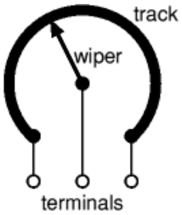


Fig4.17: Variable resistor for contrast control

4.5.5POTENTIOMETER:

Variable resistors used as potentiometers have all 3 terminals connected.

This arrangement is generally wont to vary voltage, as an example to line the switch purpose of a circuit with a device, or management the degree (loudness) in Associate in Nursing electronic equipment circuit. If the terminals at the ends of the track area unit connected across the facility offer, then the wiper terminal can give a voltage which may be varied from zero up to the most of the provision.

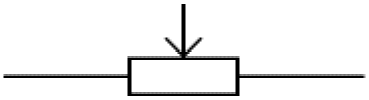


Fig4.18: Potentiometer image

4.5.6PRESETS

These area unit miniature versions of the quality resistor. They're designed to be mounted directly onto the printed circuit and adjusted only the circuit is constructed. {Forexample|forinstance as Associate in Nursing example} to line the frequency of an alarm tone or the sensitivity of a sensitive circuit. little screwdriver or similar tool is needed to regulate pre-sets.

- Pre-sets area unit less expensive than customary electrical device|rheostat|resistor|resistance} s so that they area unit typically employed in comes wherever a customary variable resistor would usually be used.
- Multiturn pre-sets area unit used wherever terribly precise changes should be created. The screw should be turned repeatedly (10+) to manoeuvre the slider from one finish of the track to the opposite, giving terribly fine management.

4.6 IR SENSORS

An infrared detector is an electronic radiation instrument that's accustomed sense sure characteristics of Its surroundings by either emitting and/or detection infrared. it's conjointly capable of measure heat of an object and detective work motion. Infrared waves aren't visible to the human eye. In the spectrum, infrared is that the region having wavelengths longer than visible radiation wavelengths, however shorter than microwaves. The infrared region is or so demarcated from zero.75 to 1000µm. The wavelength region from zero.75 to 3µm is termed as close to infrared, the region from three to 6µm is termed mid-infrared, and therefore the region beyond 6µm is termed as so much infrared.Infrared technology is found in several of our everyday product. as an example, TV has an IR detector for decoding the signal from the remote. Key advantages of infrared sensors embody low power necessities, straightforward electronic equipment, and their transportable feature.

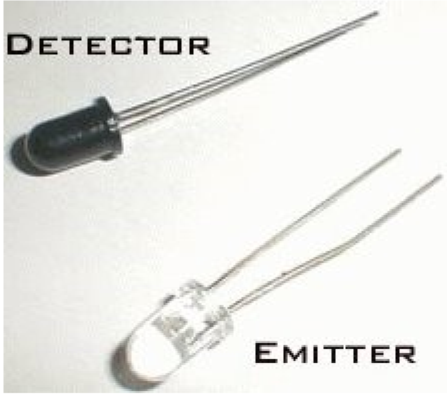


Fig4.19 : IR Sensor

4.6.1Types of Infra-Red Sensors

Infra-red sensors are generally classified into 2 types:

Thermal infrared sensors – These use infrared energy as heat. Their having some photo (light) sensitivity is freelance of wavelength. Thermal detectors don't need cooling; but they need slow response times and low detection capability.

Quantum infrared sensors – These offer higher detection performance and quicker response speed. Their image sensitivity relies on wavelength. Quantum detectors need to be cooled thus on acquire correct measurements. The sole exception is for detectors that square measure employed in the close to infrared region.

- **Working Principle**

A typical system for detection infrared emission using infrared sensors includes the infrared supply like black body radiators, W lamps, and carbide. Just in case of active IR sensors, the sources are infrared lasers and LEDs of specific IR wavelengths. Next is that the transmission medium used for infrared transmission, which has vacuum, the atmosphere, and optical fibres.Thirdly, optical parts like optical lenses made up of quartz, CaF2, Ge and Si, polythene physicist lenses, and Al or Au mirrors, are accustomed converge or focus infrared emission. Likewise, to limit spectral response, band-pass filters are ideal.Finally, the infrared detector completes the system for sleuthing infrared emission. The output from the detector is sometimes terribly tiny, and thus pre-amplifiers not to mention electronic equipment are supplementary to additional method the received signals.

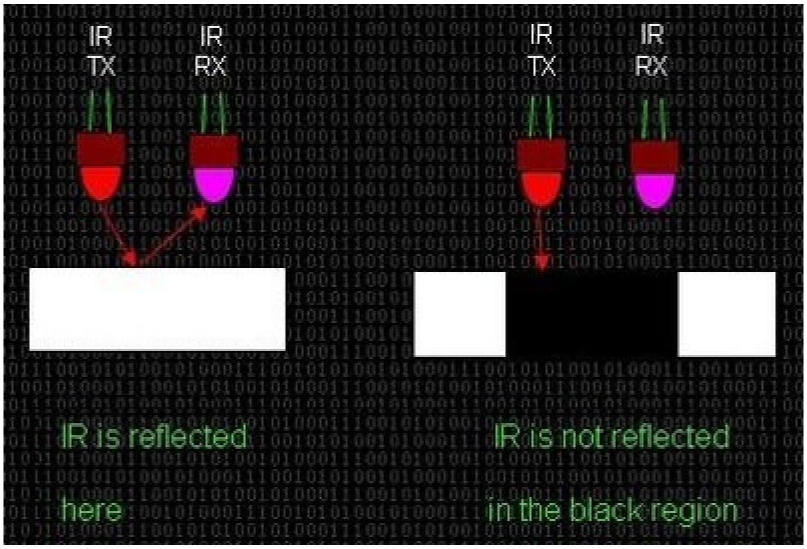


Fig 4.20: Working Of IR Sensor

4.6.2 Applications

The following are the key application areas of infrared sensors:

- Tracking and humanistic discipline
- Climatology, meteorology, and physical science
- Thermographs, communications side, and alcohol testing
- Heating, hyper spectral imaging, and vision

- Biological systems, pic bio modulation, and plant health
- Gas detectors/gas leak detection
- Water and steel analysis, flame detection
- Anaesthesiology testing and spectrographic analysis
- Petroleum exploration and underground answer
- Rail safety.
- Heating, hyper spectral imaging, and vision
- Biological systems, pic bio modulation, and health condition of plant
- Gas detectors/gas leak detection
- Water and steel analysis, flame detection

  Anaesthesiology testing and spectrographic analysis
- Petroleum exploration and underground answer
- Rail safety

4.7 LIGHT DEPENDENT RESISTORS

The Light Dependent resistance (LDR) is formed exploitation the semiconductor mineral. The device is employed up to the mark circuits to show lights on and in photographic exposure meters.The resistance of the LDR resistance will be decreases because the intensity of the sunlight falling on that will increase. Incident photons drive electrons from that valency band into that conductivity of bonding.

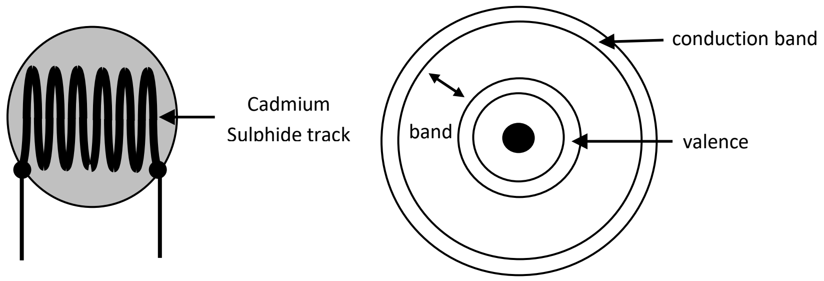


Figure 4.21 : Structure of a light-weight Dependent resistance, showing cadmium sulphide track and an atom for example electrons within the valence and physical phenomenon bands

Measure the resistance of the LDR within the dark and fully lightweight conditions,

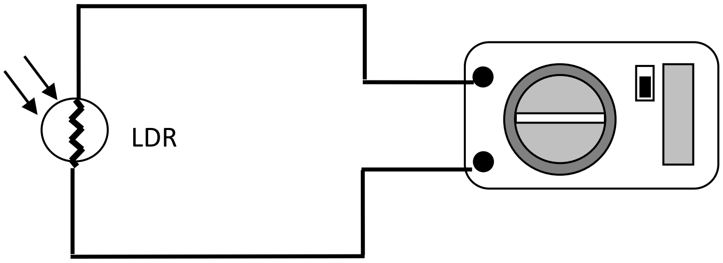


Figure4.22: measuring the resistance of the LDR

Resistance of LDR in the dark	High
Resistance of LDR in bright light	Low

Build the circuit shown in figure 3.15 and observe that circuit shown current flowing through the LDR at too different light levels. Note the response of the LED. R1 =330Ω

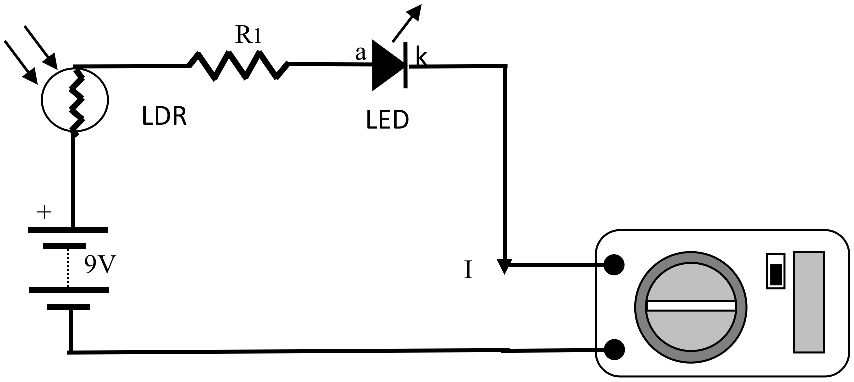


Figure4.23: LDR and LED circuit.

Investigate the dimensionality of the response from the LDR by angling the LDR thus it faces the white diode on the setup. Leave house, figure 16, to position the Neutral Density (ND) filters between the diode and also the LDR. The bread board is on a magnetic block and elements will be designed to be at a similar height because the alternative parts.

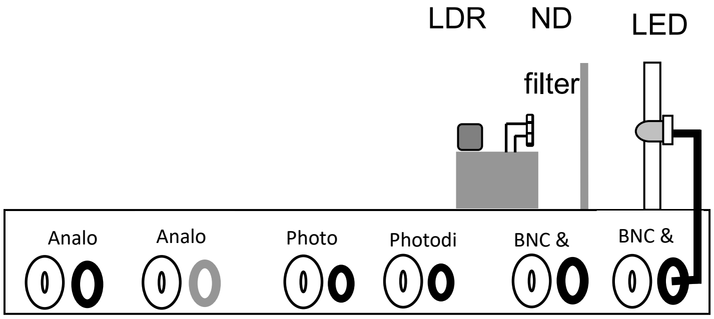


Figure4.24: mensuration theresponseoftheLDRto completelydifferent lightweight levelsT heOpticalDensityoftheNeutralDensityfilters is calculatedusing:I=transmitted lightweight andIo=Incident lightweight.Calculate the OD values for every filter and fill within the table. Live and record this flowing through the circuit with the various filters.

4.8 LED

A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The colour of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

Fig 4.25: LED

CHAPTER-5

SOFTWARE REQUIREMENT

- Arduino software
- Proteus software

5.1 Proteus Software:

Proteus is computer code for microchip simulation, schematic capture, and computer circuit board (PCB) style. it's developed by Lab centre physical science.

The XGameStation small Edition was designed victimisation PCB layout tools and Proteus schematic entry.

System Components

ISIS Schematic Capture - a tool for getting into styles.

PROSPICE Mixed Mode SPICE Simulation – business normal SPICE3F5 machine combined with a digital machine.

ARES PCB Layout – PCB style system with automatic part alluvial sediment, rip-up and rehear auto-router and interactive style rule checking.

VSM – Virtual System Modelling lets co-simulate embedded computer code for well-liked micro-controllers aboard hardware style.

System edges Integrated package with common interface and absolutely context sensitive facilitate.

PROTUES combines advanced schematic capture, mixed mode SPICE simulation, PCB layout and motorcar routing to create a whole electronic style system

The PROTUES product vary conjointly includes our revolutionary VSM technology to perform the system desired task.

Product Features:

ISIS Schematic Capture a straightforward to use however and intensely powerful tool for getting into your style

- PROSPICE Mixed mode SPICE Simulation business normal SPICE3F5 machine
- ARES – layout designing in this tool
- All modules are standardised Graphical interface.
- Runs on Windows 98/ME/2000/XP or Later
- Technical Support direct kind the author
- Rated best overall product

5.1.1 Intelligent Schematic Input System (ISIS):

ISIS lies right at the guts of the PROTUES system and is way over simply another schematic package. it's powerful setting to regulate most aspects of the drawing look. whether or not your demand is that the speedy entry of complicated style for simulation & PCB layout, Or the creation of engaging Schematic for publication ISIS is that the right tool for the work Product

Features:

- Produces publication quality schematic
- Style templates enable customization of equipped library
- Mouse driven context sensitive interface
- Automatic wire routing and junction dot placement
- Full support for buses as well as sub- circuit ports and bus pins
- Large and growing part library of over 8000 elements

5.1.2 VSM (Virtual System Modelling):

Proteus VSM is associate degree extension of the PROSPICE machine that facilities co-simulation of microchip primarily based style as well as all the associated physical science. what is more, you'll move with the microcontroller computer code through the employment of animated keypads, switches, buttons, LEDs, lamps and even alphanumeric display displays.

Features:

CPU models offered for several well-liked microcontrollers as well as PIC, AVR, HC11 and ArduinoInteractive device models embody semiconductor diode and alphanumeric display displays, RS232 terminal, universal input device and a spread of switches, buttons, pots, Extensive debugging facilities as well as register and memory contents, breakpoints and single step modes.Source level debugging for select development tools as well as IAR C-SPY and

Keil Vision.

Embedded "C" Compiler.

- ANSI C - full featured and moveable
- Reliable - mature, field-proven technology
- Multiple C optimisation levels
- An optimizing computer programme.
- Full linker, with overlaying of native variables to attenuate RAM usage
- Comprehensive C library with all ASCII text file provided
- Includes support for 24-bit and 32-bit IEEE floating purpose and 32-bit long information sorts
- Mixed C and computer programme programming
- Unlimited range of supply files
- Listings showing generated computer programme

- Compatible - integrates into the MPLAB IDE, MPLAB ICD and most 3rd-party development tools

- Runs on multiple platforms: Windows, Linux, UNIX, Mac OS X, Solaris

5.1.3 Embedded System Tools:

A computer program is a translating programming language basically, illustration of machine language into computer code. A cross computer program produces code for one variety of processor, however runs on another. The process step wherever a computer program is run is thought as assembly time. Translating assembly instruction method into opcodes, assemblers give the flexibility to use symbolic names for memory locations and macro facilities for performing.arts matter substitution usually won't to inscribe common short sequences of directions to run inline rather than in an exceedingly subprogram. Assemblers are so much easier to write down than compilers for high-level languages. Assembly Language has many Benefits.

Speed:

Assembly language programs are usually the quickest programs around.

Space:

Assembly language programs are typically the littlest.

5.2 "Arduino" what does it mean?

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating It's offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open- source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

5.2.1 Why Arduino?

Thanks to It's simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than$50

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version.

- Of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

- Getting Started with Arduino and Genuine products: -

Install the Arduino Software (IDE) on Windows PCs-

…This document explains how to install the Arduino Software (IDE) on Windows machines.

✓ Download the Arduino Software (IDE)

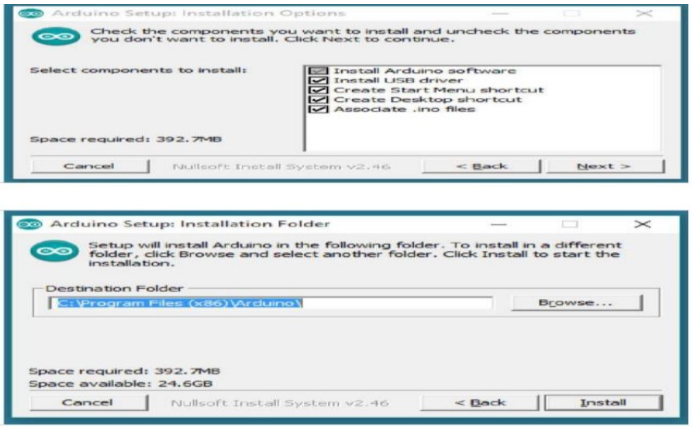
✓ Proceed with board specific instructions.

5.2.2 How to Download the Arduino Software (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.

5.2.3 Proceed with board specific instructions:

When the Arduino Software (IDE) is properly installed you can go back to the Getting Started Home and choose your board from the list on the right of the page.



Arduino Setup: Installation Options

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ☑ Install Arduino software
- ☑ Install USB driver
- ☑ Create Start Menu shortcut
- ☑ Create Desktop shortcut
- ☑ Associate .ino files

Space required: 392.7MB

Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino\

Space required: 392.7MB
Space available: 24.6GB

Fig: 5.1: software program selection

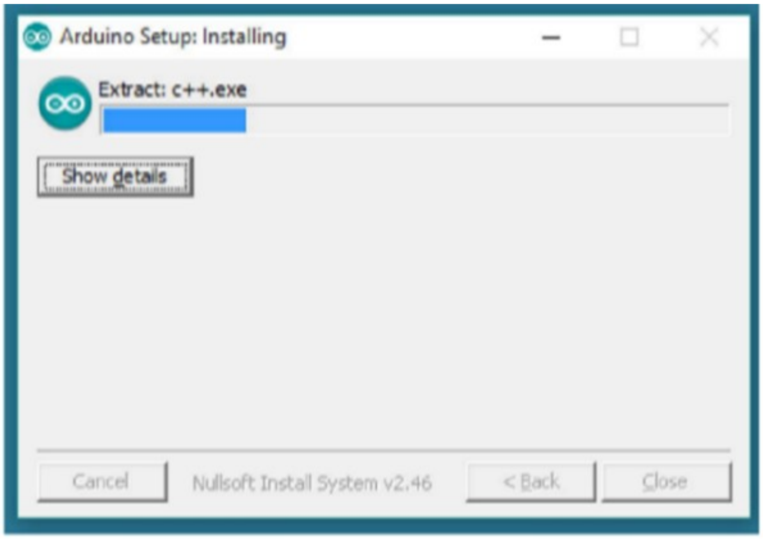


Fig: 5.2: program detailed check

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main () into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

In October 2019 the Arduino organization began providing early access to a new

Arduino Pro IDE with debugging and other advanced features

The IDE environment is mainly distributed into three sections

- Menu Bar
- Text Editor
- Output Pane

5.2.4Arduino IDE Introduction:

The bar appearing on the top is called Menu Bar that comes with five different options as follows.

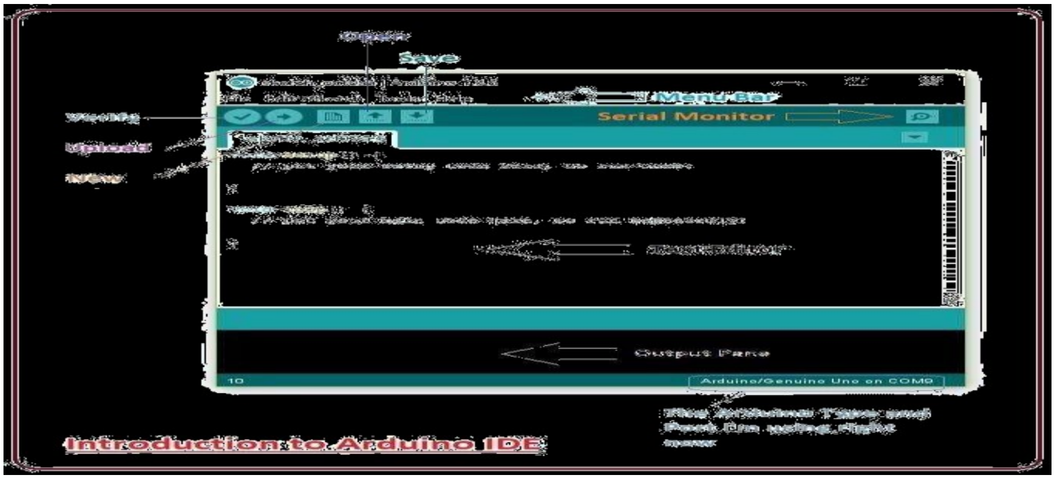
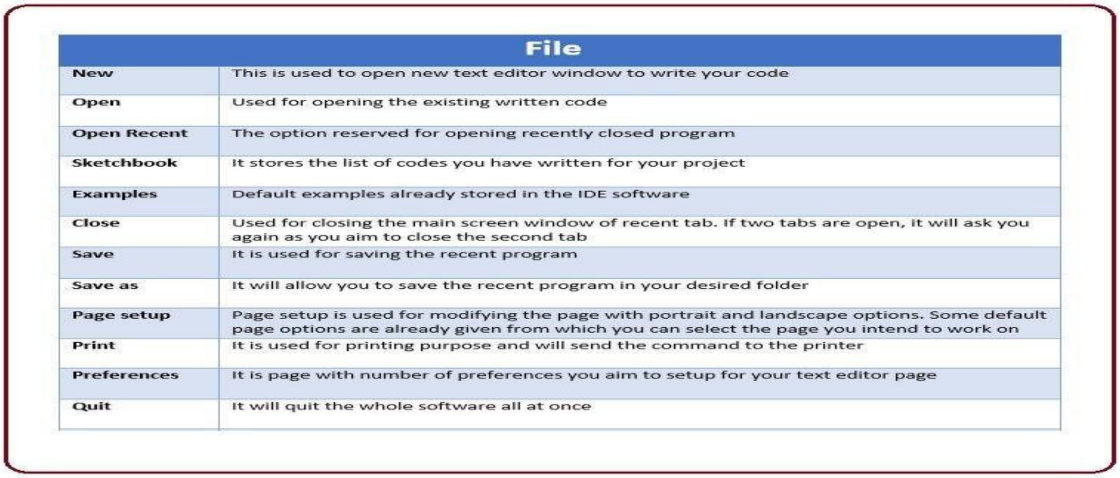


Fig.5.3: Menu Bar of Arduino IDE

5.2.1 File Description:

You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into. Creating file descriptors

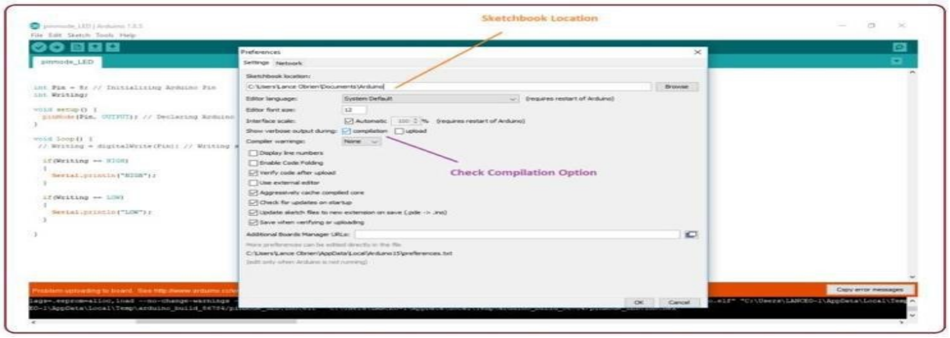
Open (), create () socket (), accept (), socketpair (), pipe (), epoll_create (), (Linux), signalfd () (Linux), eventfd () (Linux)timerfd_create () (Linux) etc…,



File	
New	This is used to open new text editor window to write your code
Open	Used for opening the existing written code
Open Recent	The option reserved for opening recently closed program
Sketchbook	It stores the list of codes you have written for your project
Examples	Default examples already stored in the IDE software
Close	Used for closing the main screen window of recent tab. If two tabs are open, it will ask you again as you aim to close the second tab
Save	It is used for saving the recent program
Save as	It will allow you to save the recent program in your desired folder
Page setup	Page setup is used for modifying the page with portrait and landscape options. Some default page options are already given from which you can select the page you intend to work on
Print	It is used for printing purpose and will send the command to the printer
Preferences	It is page with number of preferences you aim to setup for your text editor page
Quit	It will quit the whole software all at once

Fig.5.4: File Description

As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.



```
int Pin = 8; // Initializing Arduino Pin
int Writing;

void setup() {
  pinMode(Pin, OUTPUT); // Declaring Arduino
}

void loop() {
  // Writing = digitalWrite(Pin); // Writing

  if(Writing == HIGH)
  {
    Serial.println("HIGH");
  }

  if(Writing == LOW)
  {
    Serial.println("LOW");
  }

}
```

Preferences

Settings Network

Sketchbook location: C:\Users\Lance Obrien\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during: ☑ compilation ☐ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

☑ Verify code after upload

☐ Use external editor

☑ Aggressively cache compiled core

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs:

More preferences can be edited directly in the file

C:\Users\Lance Obrien\AppData\Local\Arduino15\preferences.txt

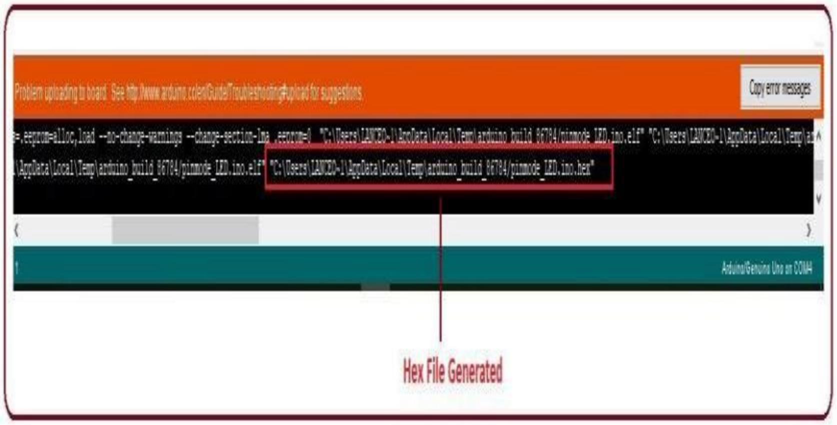
(edit only when Arduino is not running)

Problem uploading to board. See http://www.arduino.cc/en/

```
lags=.eeprom=alloc,load --no-change-warnings
EO-1\AppData\Local\Temp\arduino_build_86784/pi
```

Fig.5.5: Compilation Screen

And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.



Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions.

```
=.eeprom=alloc,load --no-change-warnings --change-section-lma .eeprom=0 "C:\Users\LANCEO-1\AppData\Local\Temp\arduino_build_86784/pinmode_LED.ino.elf" "C:\Users\LANCEO-1\AppData\Local\Temp\a
\AppData\Local\Temp\arduino_build_86784/pinmode_LED.ino.elf" "C:\Users\LANCEO-1\AppData\Local\Temp\arduino_build_86784/pinmode_LED.ino.hex"
```

Fig.5.6: Hex File Generation

5.2.2 Edit Description:

Used for copying and pasting the code with further modification for font

5.2.3 Sketch Description:

For compiling programming and

5.2.4 Tools Description:

Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller

5.2.5 Help Description:

In case you are feeling sceptical about software, complete help is available from getting started to troubleshooting.

The Six Buttons appearing under the Menu tab are connected with the running program as follows

While Arduino IDE is highly-rated by users according to ease of use, it is also capable of performing complex processes without taxing computing resources.

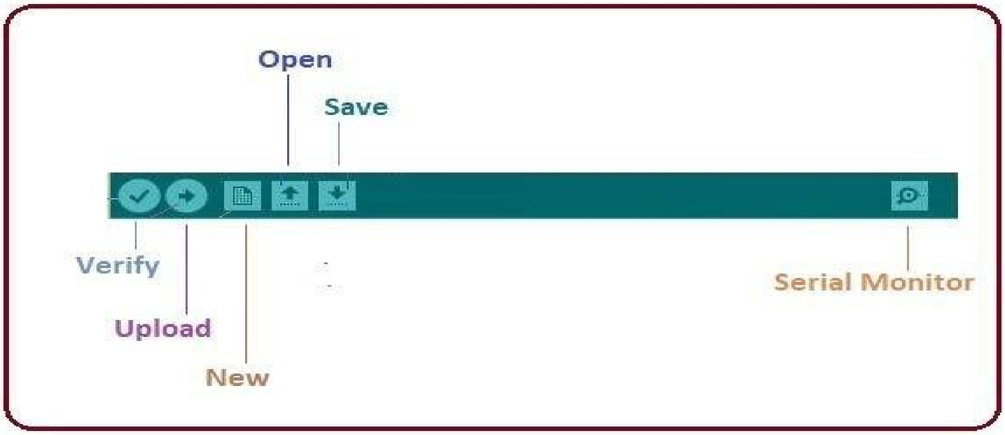


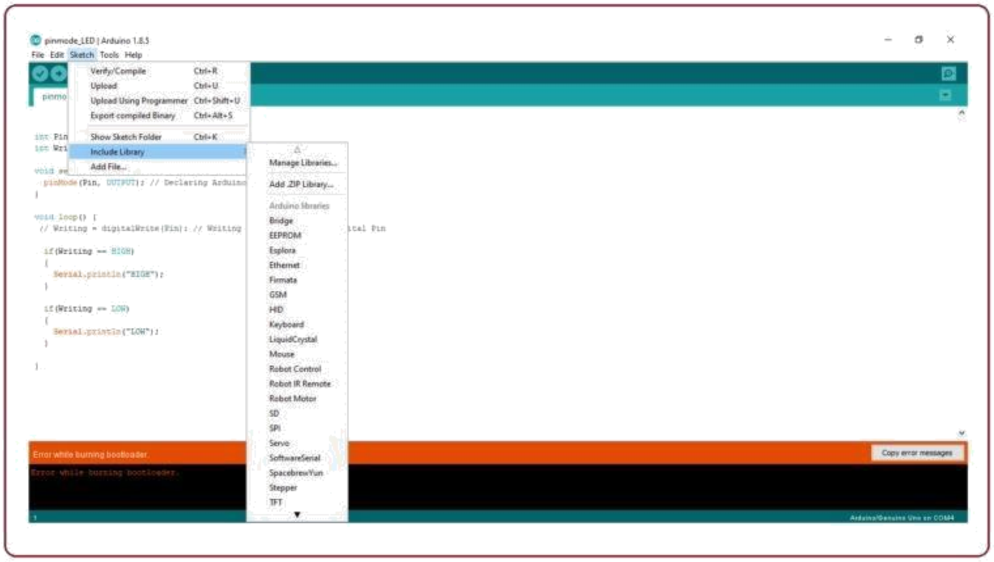
Fig.5.7: Menu Tab

- The check mark appearing in the circular button is used to verify the code. Click this once you have written your code.
- The arrow key will upload and transfer the required code to the Arduino board.
- The dotted paper is used for creating a new file.
- The upward arrow is reserved for opening an existing Arduino project.

- The downward arrow is used to save the current running code.

- The button appearing on the top right corner is a Serial Monitor – A separate pop-up window that acts as an independent terminal and plays a vital role for sending and receivingthe Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressingCtrl+Shift+M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.

You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below:



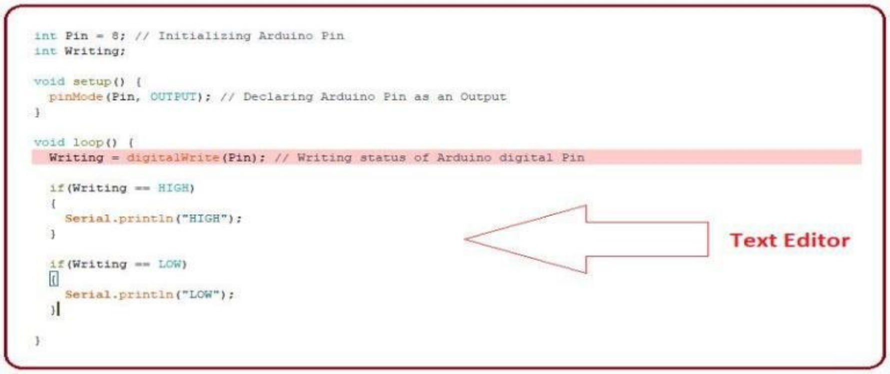
```
int Pin
int Wri

void se
  pinMode(Pin, OUTPUT); // Declaring Arduino
}

void loop() {
 // Writing = digitalWrite(Pin); // Writing                     ital Pin

  if(Writing == HIGH)
  {
    Serial.println("HIGH");
  }

  if(Writing == LOW)
  {
    Serial.println("LOW");
  }

}
```

```
Error while burning bootloader.
Error while burning bootloader.
```

```
1                                                    Arduino/Genuino Uno on COM4
```

Fig.5.8: Connecting to Board

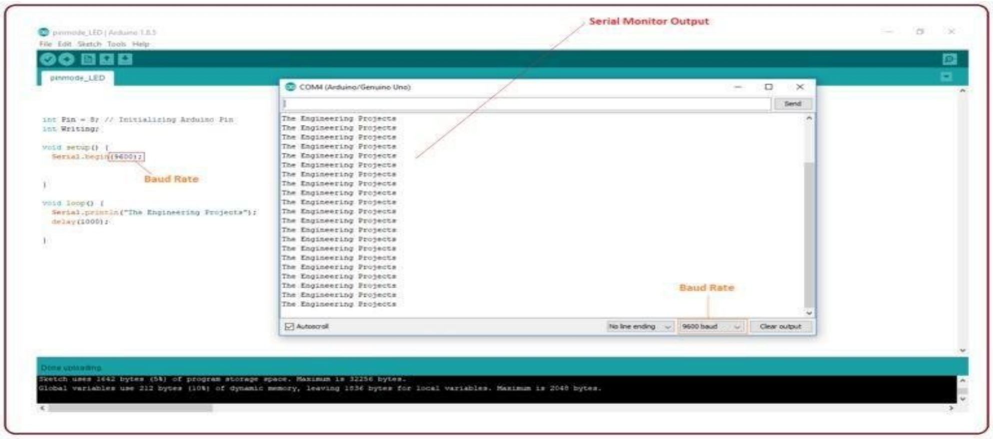
The main screen below the Menu board is known as a simple text editor used for writing the registration code.



```
int Pin = 8; // Initializing Arduino Pin
int Writing;

void setup() {
  pinMode(Pin, OUTPUT); // Declaring Arduino Pin as an Output
}

void loop() {
  Writing = digitalWrite(Pin); // Writing status of Arduino digital Pin

  if(Writing == HIGH)
  {
    Serial.println("HIGH");
  }

  if(Writing == LOW)
  {
    Serial.println("LOW");
  }

}
```

Fig.5.9: Text Editor

The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.



```
int Pin = 8; // Initializing Arduino Pin
int Writing;

void setup() {
  Serial.begin(9600);

}

void loop() {
  Serial.println("The Engineering Projects");
  delay(1000);

}
```

```
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
The Engineering Projects
```

```
Sketch uses 1642 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 212 bytes (10%) of dynamic memory, leaving 1836 bytes for local variables. Maximum is 2048 bytes.
```

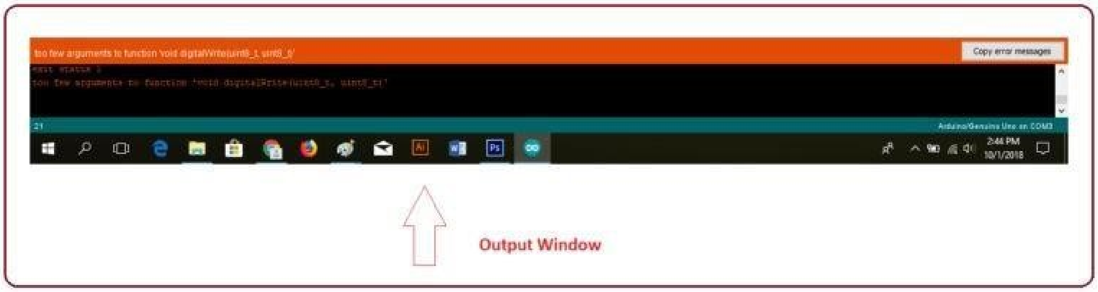
Fig.5.10: Output Window

More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.

5.3 Libraries:

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library. The Arduino environment can be extended through the use of libraries, just like most programming platforms. Libraries provide extra functionality for use in sketches, e.g., working with hardware or manipulating data. To use a library in a sketch, select it from Sketch > Import Library.

A number of libraries come installed with the IDE, but you can also download or create your own. See these instructions for details on installing libraries. There's also a tutorial on writing your own libraries. See the API Style Guide for information on making a good Arduino-style API for your library.



```
too few arguments to function 'void digitalWrite(uint8_t, uint8_t)'
exit status 1
too few arguments to function 'void digitalWrite(uint8_t, uint8_t)'
```

Fig.5.11: Arduino IDE Library

As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

#include <EEPROM.h>

Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from the external sources.

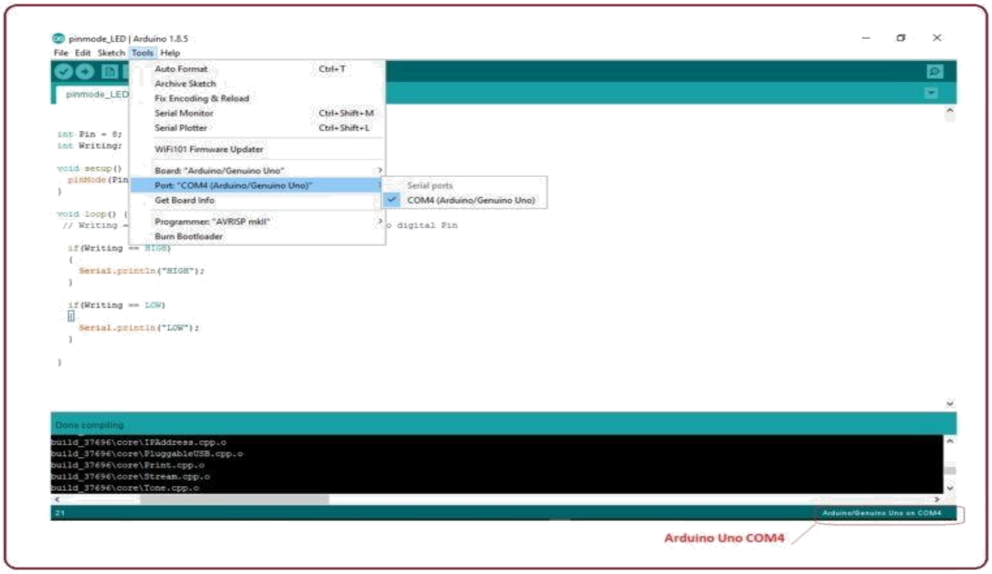
5.4 Making Pins Input or Output:

The digitalRead and digitalWrite commands are used for addressing and making the Arduino pins as an input and output respectively.

These commands are text sensitive i.e., you need to write them down the exact way they are given like digitalWrite starting with small “d” and write with capital “W”. Writing it down with Digitalwrite or digitalwrite won’t be calling or addressing any function.

5.5 How to Select the Board:

On the online IDE we are able to automatically detect the kind of board and the port it is connected to without you having to individually select them. In order to upload sketch, you need to select the relevant board you are using and the ports for that operating system. As you click the Tools on the Menu; it will open like the figure below. You’ll need to select the entry in the Tools > Board menu that corresponds to your Arduino board. Select the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.


pinmode_LED | Arduino 1.8.5
File Edit Sketch Tools Help
pinmode_LED
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM4 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
✓ COM4 (Arduino/Genuino Uno)
21
Arduino Uno COM4

```
int Pin = 8;
int Writing;

void setup()
  pinMode(Pin
}

void loop() {
 // Writing =                                    o digital Pin

  if(Writing == HIGH)
  {
    Serial.println("HIGH");
  }

  if(Writing == LOW)
  {
    Serial.println("LOW");
  }

}
```

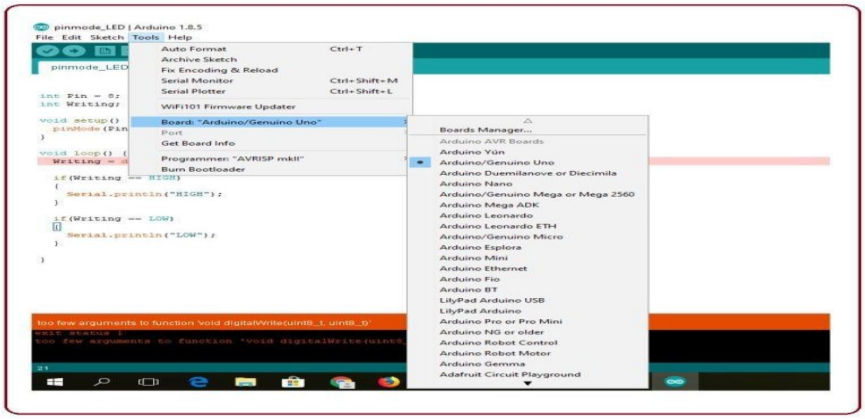
Done compiling

```
build_37696\core\IPAddress.cpp.o
build_37696\core\PluggableUSB.cpp.o
build_37696\core\Print.cpp.o
build_37696\core\Stream.cpp.o
build_37696\core\Tone.cpp.o
```

Arduino/Genuino Uno on COM4

Fig.5.12: Selection of Board

Just go to the “Board” section and select the board you aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USBboard. You can look for the USB serial device in the port section of the Windows Device Manager.Following figure shows the COM4 that I have used for my project, indicating the Arduino Uno with COM4 port at the bottom right corner of the screen. you are using and the ports for that operating system. As you click the Tools on the Menu; it will open like the figure below. You’ll need to select the entry in the Tools > Board menu that corresponds to your Arduino board. Select the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.



```
int Pin = 8;
int Writing;

void setup()
  pinMode(Pin
}

void loop() {
  Writing = d

  if(Writing == HIGH)
  {
    Serial.println("HIGH");
  }

  if(Writing == LOW)
  {
    Serial.println("LOW");
  }

}
```

too few arguments to function 'void digitalWrite(uint8_t, uint8_t)'

```
exit status 1
too few arguments to function 'void digitalWrite(uint8_
```

Fig.5.13: Example

- After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six-button section or you can go to the Sketch section and press verify/compile and then upload.

- The sketch is written in the text editor and is then saved with the file extension

It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, older version may require the physical reset on the board.

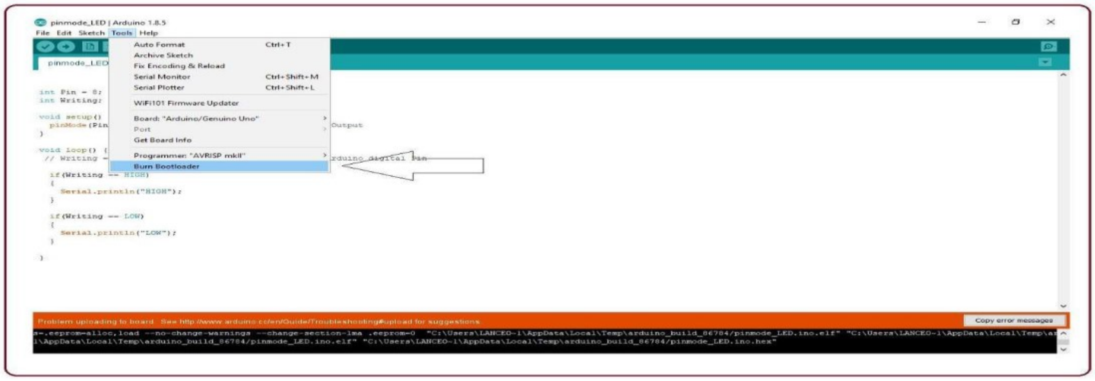
- Once you upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

Note: The port selection criteria mentioned above is dedicated for Windows operating system only, you can check this Guide if you are using a MAC or Linux.

- The amazing thing about this software is that no prior arrangement or bulk of mess is required to install this software, you will be writing your first program within 2 minutes after the installation of the IDE environment.

5.6 Bootloader:

As you go to the Tools section, you will find a bootloader at the end. It is very helpful to burn the code directly into the controller, setting you free from buying the external burner to burn the required code.



```
int Pin = 8;
int Writing;

void setup()
  pinMode(Pin                                   Output
}

void loop() {
 // Writing -                                  rduino digital Pin

  if(Writing == HIGH)
  {
    Serial.println("HIGH");
  }

  if(Writing == LOW)
  {
    Serial.println("LOW");
  }

}
```

```
Problem uploading to board.  See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions.
s=.eeprom=alloc,load --no-change-warnings --change-section-lma .eeprom=0  "C:\Users\LANCEO~1\AppData\Local\Temp\arduino_build_86784/pinmode_LED.ino.elf" "C:\Users\LANCEO~1\AppData\Local\Temp\ar
1\AppData\Local\Temp\arduino_build_86784/pinmode_LED.ino.elf" "C:\Users\LANCEO~1\AppData\Local\Temp\arduino_build_86784/pinmode_LED.ino.hex"
```

Fig.5.14: Bootloader

When you buy the new Arduino Module, the bootloader is already installed inside the controller. However, if you intend to buy a controller and put in the Arduino module, you need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader.

5.7 Android Application:

Android application: The multi terminal is intended for simultaneous management of multiple accounts, such as WIFI and Bluetooth for which is mostly helpful transmitting messages to the display. The new terminal successfully combines great functionalities that allow effective transferring with many accounts and with exceptional usability Terminal can easily get acquainted to this new program within a few minutes. After installing the application in mobile-phone it needs to configure with password.

BT Terminal is a terminal app with UART serial communication protocol that transmits & receives data wirelessly through Bluetooth connections. The app can be used for Robotics Communication, Configuring Bluetooth Modules (using AT Commands), Home Automation, etc.

5.8 Arduino History:

Colombian student Hernando Barraging created the development platform wiring as his Master's thesis project in 2004 at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. Massimo Banzi and Casey Reas(known for his work on Processing) were supervisors for his thesis. The goal was to create low cost, simple tools for non-engineers to create digital projects. The Wiring platform consisted of a hardware PCB with an ATmega128 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2005, Massimo Banzi, with David Mellis (then an IDII student) and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they forke (or copied) the Wiring source code and started running it as a separate project, called Arduino. The Arduino's initial core team consisted of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis. The name Arduino comes from a bar in Ivrea, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.Following the completion of the Wiring platform, its lighter, lower cost versions were created and made available to the open-source community. Associated researchers, including David Cuartielles, promoted the idea. The first prototype board, made in 2005, was a simple design, and it wasn’t called Arduino.

5.8.1 Types of Arduino

There are mainly 3 types of Arduinos

Major types are discussed below:

1. Entry Level

Get started with Arduino using Entry Level products: easy to use and ready to power your first creative projects. These boards and modules are the best to start learning and tinkering with electronics and coding. The Starter Kit includes a book with 15 tutorials that will walk you through the basics up to complex projects.

5.8.2 PROTEUS SOFTWARE:

Proteus is computer code for microchip simulation, schematic capture, and computer circuit board (PCB) style. it's developed by Lab center physical science.

The XGameStation small Edition was designed victimization PCB layout tools and Proteus schematic entry.

System Components

ISIS Schematic Capture - a tool for getting into styles.

PROSPICE Mixed Mode SPICE Simulation – business normal SPICE3F5 machine combined with a digital machine.

ARES PCB Layout – PCB style system with automatic part alluvial sediment, rip-up and rehear auto-router and interactive style rule checking.

VSM – Virtual System Modelling lets co-simulate embedded computer code for well-liked micro-controllers aboard hardware style.

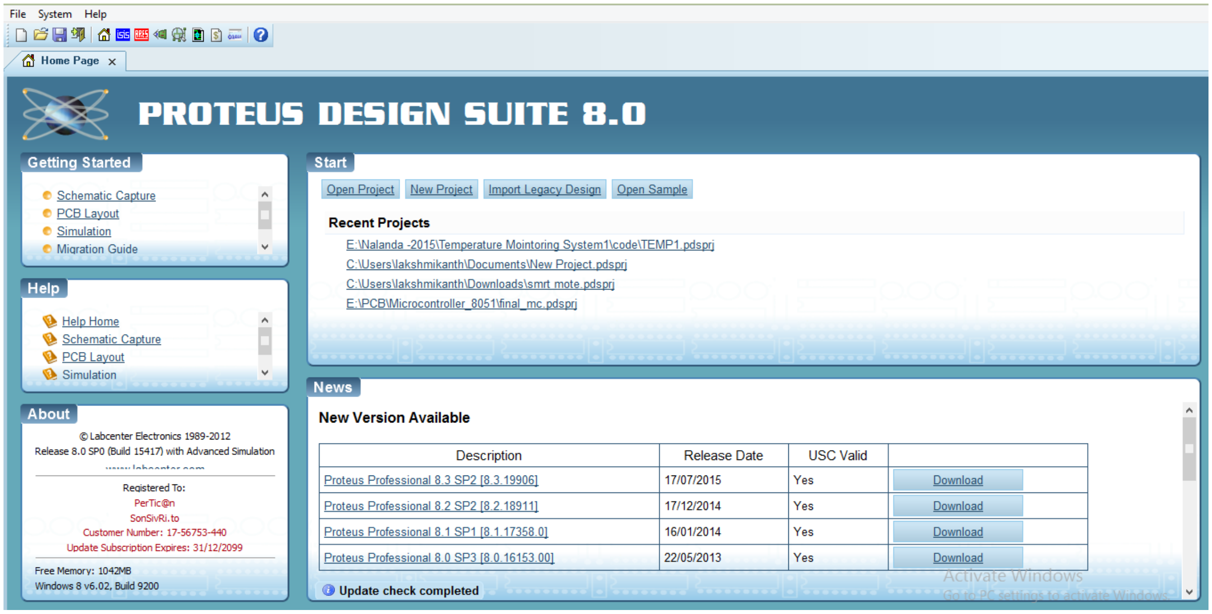
System edges integrated package with common interface and absolutely context sensitive facilitate.

PROTUES combines advanced schematic capture, mixed mode SPICE simulation, PCB layout and motorcar routing to create a whole electronic style system

The PROTUES product vary conjointly includes our revolutionary VSM technology to perform the system desired task.

Product Features:

- ISIS Schematic Capture a straightforward to use however and intensely powerful tool for getting into your style
- PROSPICE Mixed mode SPICE Simulation business normal SPICE3F5 machine
- ARES – layout designing in this tool
- All modules are standardised Graphical interface.
- Runs on Windows 98/ME/2000/XP or Later
- Technical Support direct kind the author
- Rated best overall product



Recent Projects

E:\Nalanda -2015\Temperature Mointoring System1\code\TEMP1.pdsprj
C:\Users\lakshmikanth\Documents\New Project.pdsprj
C:\Users\lakshmikanth\Downloads\smrt mote.pdsprj
E:\PCB\Microcontroller_8051\final_mc.pdsprj

© Labcenter Electronics 1989-2012
Release 8.0 SP0 (Build 15417) with Advanced Simulation
www.labcenter.com

Registered To:
PerTic@n
SonSivRi.to
Customer Number: 17-56753-440
Update Subscription Expires: 31/12/2099

Free Memory: 1042MB
Windows 8 v6.02, Build 9200

New Version Available

Description	Release Date	USC Valid	
Proteus Professional 8.3 SP2 [8.3.19906]	17/07/2015	Yes	Download
Proteus Professional 8.2 SP2 [8.2.18911]	17/12/2014	Yes	Download
Proteus Professional 8.1 SP1 [8.1.17358.0]	16/01/2014	Yes	Download
Proteus Professional 8.0 SP3 [8.0.16153.00]	22/05/2013	Yes	Download

Update check completed

4.9 Intelligent Schematic Input System (ISIS):

ISIS lies right at the guts of the PROTUES system and is way over simply another schematic package. it's powerful setting to regulate most aspects of the drawing look. Whether or not your demand is that the speedy entry of complicated style for simulation & PCB layout, Or the creation of engaging Schematic for publication ISIS is that the right tool for the work Product

Features:

- Produces publication quality schematic
- Style templates enable customization of equipped library
- Mouse driven context sensitive interface
- Automatic wire routing and junction dot placement
- Full support for buses as well as sub- circuit ports and bus pins
- Large and growing part library of over 8000 elements

VSM (Virtual System Modelling):

Proteus VSM is associate degree extension of the PROSPICE machine that facilities co-simulation of microchip primarily based style as well as all the associated physical science. what is more, you'll move with the microcontroller computer code through the employment of animated keypads, switches, buttons, LEDs, lamps and even alphanumeric display displays.

Features:

- CPU models offered for several well-liked microcontrollers as well as PIC, AVR, HC11 and Arduino
- Interactive device models embody semiconductor diode and alphanumeric display displays, RS232 terminal, universal input device and a spread of switches, buttons, pots, LEDs, seven -segment displays and additional.
- Extensive debugging facilities as well as register and memory contents, breakpoints and single step modes.

CHAPTER-6

RESULT

Here is the project we successfully executed the output and the whole procedure is working as per the result. The result of the project is given below:

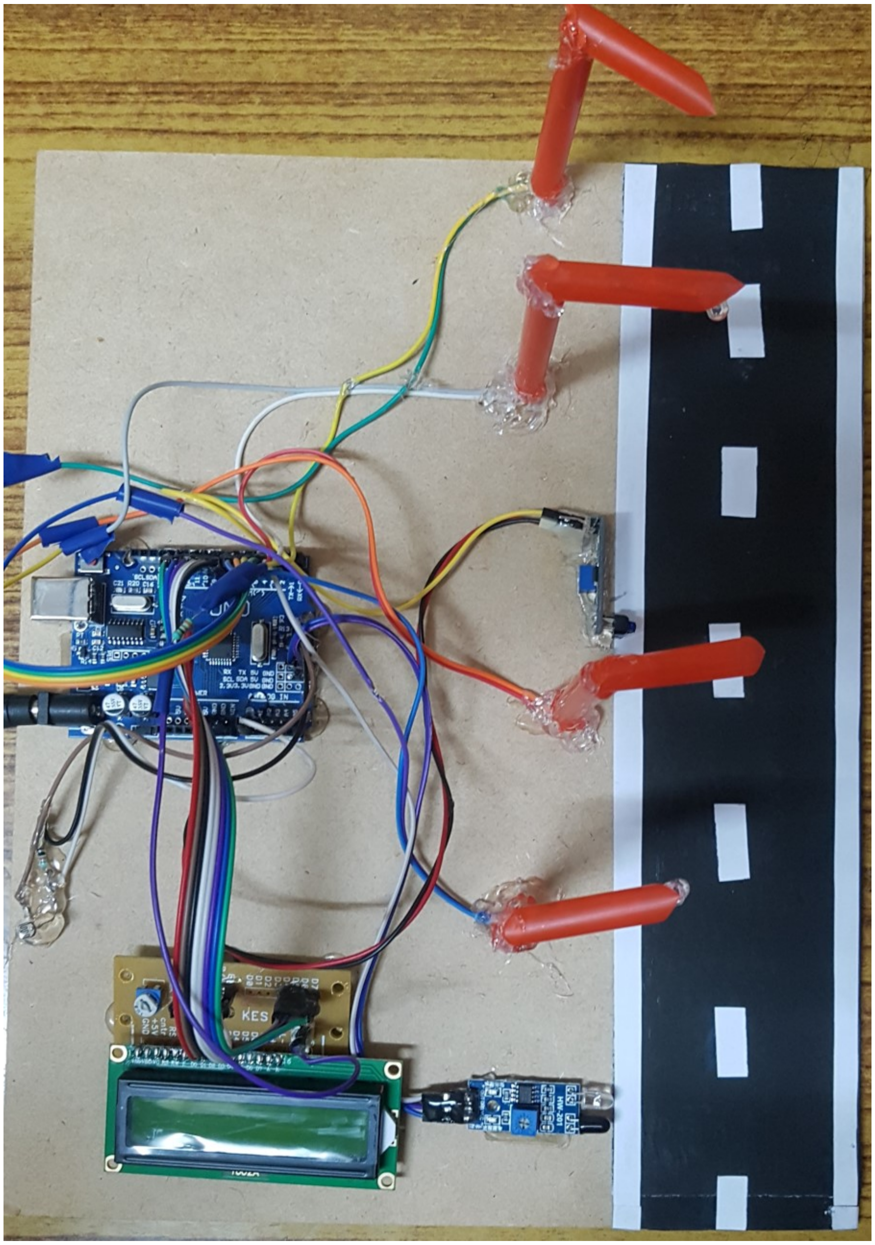


Fig 6.1: Project Figure

6.2 Source Code

```
#include<LiquidCrystal.h>

LiquidCrystal lcd(8,9,10,11,12,13);
```

```
int ldr=A0;
int ir1=2;
int ir2=3;
int led1=4;
int led2=5;
int led3=6;
int led4=7;

void setup()
{
  lcd.begin(16,2);
  lcd.setCursor(0,0);lcd.print("HIGHWAY LIGHTS");
  lcd.setCursor(0,1);lcd.print("SMART CONTROL SYS");
   pinMode(led1,OUTPUT); pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT); pinMode(led4,OUTPUT);
  digitalWrite(led1,HIGH);digitalWrite(led2,HIGH);
  digitalWrite(led3,HIGH);digitalWrite(led4,HIGH);
   pinMode(ir1,INPUT);pinMode(ir2,INPUT);
    pinMode(ldr,INPUT);
  delay(3000);
  lcd.clear();
    lcd.setCursor(0,0);lcd.print("MODE  :");delay(50);
    lcd.setCursor(0,1);lcd.print("LIGHTS:");delay(50);
}
```

```
void loop()
{
  int sensorValue = analogRead(ldr);
 if(sensorValue<=200)
 {
 lcd.setCursor(7,0);lcd.print("NIGHT");delay(50);
     if(digitalRead(ir1)==LOW)
     {
        lcd.setCursor(7,1);lcd.print("ON  ");
      digitalWrite(led1,LOW);digitalWrite(led2,LOW);
      //digitalWrite(led3,HIGH);digitalWrite(led4,HIGH);
      delay(2000);
     }
     else if(digitalRead(ir2)==LOW)
     {
        lcd.setCursor(7,1);lcd.print("ON  ");
      //digitalWrite(led1,HIGH);digitalWrite(led2,HIGH);
      digitalWrite(led3,LOW);digitalWrite(led4,LOW);delay(2000);
     }
     else
     {
      lcd.setCursor(7,1);lcd.print("OFF  ");
      digitalWrite(led1,HIGH);digitalWrite(led2,HIGH);
```

```
        digitalWrite(led3,HIGH);digitalWrite(led4,HIGH);delay(100);
       }
  }
  else
  {
    lcd.setCursor(7,0);lcd.print("DAY   ");
    lcd.setCursor(7,1);lcd.print("OFF  ");
        digitalWrite(led1,HIGH);digitalWrite(led2,HIGH);
        digitalWrite(led3,HIGH);digitalWrite(led4,HIGH);delay(100);
  }
}
```

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

This Arduino based project will provide a competent method for lighting systems and make the whole process of energy saving easier and efficient. With a capability to change the amount of light emitted depending upon the outside condition is no doubt an innovation with many future applications apart from the fact that it can also be used in many present-day tech such as head lights, street light, park lights, industrial lights and many more. The usage of the smart lighting system will undoubtedly change the world that we see today.

FUTURE SCOPE:

- Reduced energy cost and usage with flexible dimming controls.
- Increased pedestrian satisfaction through improved safety measures.
- Lowered repair and maintenance costs with the monitoring software.
- Reduced carbon emissions and light pollution.
- Increased lamp life and shorter response times to outages.
- Improved architectural planning based on real traffic patterns and insights. and

Increased revenue opportunities, such as leasing poles for digital signage or other services.

REFERENCE

[1].Lee, J.; Baik, S.; Lee, C. Building an Integrated Service Management Platform for Ubiquitous Cities. Computer 2011, 44, 56–63.
[2]. Li, Z.; Chen, C.; Wang, K. Cloud Computing for Agent-Based Urban Transportation Systems. IEEE Intell. Syst. 2011, 26, 73–79.
[3]. Mitton, N.; Papavassiliou, S.; Puliafito, A.; Trivedi, K.S. Combining Cloud and sensors in a smart city environment. EURASIP J. Wirel. Commun. Netw. 2012, 2012, doi:10.1186/1687-1499-2012-247.

[4].Bonomi,F.;Milito,R.;Zhu,J.;Addepalli,S.FogComputingandItsRoleintheInternetof Things. InProceedingsoftheFirstEditionoftheMCCWorkshoponMobile nProceedingsoftheFirstEditionoftheMCCWorkshopon mobile CloudComputing(MCC'12), Helsinki, Finland, 13–17 August 2012; pp. 13–16.
[5]. Fox, G.C.; Hartman, R.; Kamburugamuva, S. Open SourceIoT Cloud.Available online: https://sites.google.com/site/opensourceiotcloud (accessed on 1 July 2015). IoT using Smart Street Light System www.jst.org.in 311Page
[6]. Li, F.; Voegler, M.; Claessens, M.; Dustdar, S. Efficient and Scalable IoT Service Delivery on Cloud. In Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD), Santa Clara, CA, USA, 28 June–3 July 2013; pp. 740–747.
[7]. PTC. ThingWorx—Internet of Things and M2M Application Platform. Available online: http://www.thingworx.com/ (accessed on 1 July 2015).
[8]. SmartThings, Inc. SmartThings Open Cloud. Available online: http://www.smartthings. com/opencloud/ (accessed on 1 July 2015). Leccese, F.; Cagnetti, M.; Trinca, D. A Smart City Application: A Fully Controlled Street Lighting Isle Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX. Sensors 2014, 14, 24408–24424.
[9]. Zaidi, S.; Imran, A.; McLernon, D.; Ghogho, M. Enabling IoT empowered smart lighting solutions: A communication theoretic perspective. In Proceedings of the 2014 IEEE
[10]. Wireless Communications and Networking Conference Workshops (WCNCW), Istanbul, Turkey, 6–9 April 2014; pp.140-144
[11]. Archana. G, Aishwarya N, Anitha J "Intelligent Street Light System" International Journal of Recent Advances in Engineering & Technology, Vol-3, Issue- 4, 2015
[12]. DeepanshuKhandelwal, Bijo M Thomas, KritikaMehndiratta, Nitin Kumar "Sensor Based Automatic Street Lighting system" International Journal of Education and Science Research Review Volume-2, Issue-2 April- 2015.
[13]. Soledad Escolar, JesúsCarretero, Maria-Cristina Marinescu and Stefano Chessa "Estimating Energy Savings in Smart Street Lighting byUsing an Adaptive Control System" International Journal of Distributed Sensor NetworksVolume 2014, Article ID 971587
[14]. Samir A. ElsagheerMohamed "Smart Street Lighting Control and Monitoring System for Electrical Power Saving by Using VANET",Int. J. Communications, Network and System Sciences, 2013, 6, 351-360.
[15]. Andrea Zanella, Senior Member, IEEE, Nicola Bui, Angelo Castellani "Internet of Things for Smart Cities" IEEE Internet Of Things Journal, vol. 1, no. 1, Feb. 2014.
[16] H.N. Lokhande, S.D. Markande "Adaptive Street Light Controlling For Smart Cities" in International Journal of Applied Engineering Research, Volume: 13 Issue:10 Nov-2018, pp. 7719-7723
[17]. J.Sherly1,D.Somasundareswari "INTERNET OF THINGS BASED SMART TRANSPORTATION SYSTEMS" International Research Journal of Engineering and Technology Volume: 02 Issue: 07 | Oct-2015.