SENSOR BASED IDENTIFICATION SYSTEM FOR TRAIN COLLISION AVOIDANCE

A Project Report submitted in Partial fulfilment of the requirement for the Award Degree of

BACHELOR OF TECHNOLOGY
IN
ELECTONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY

B. VIJAY KUMAR REDDY **20HM1A0405**

Under the esteemed guidance of
K. SANJEEVRAO MTech
Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

A.M. REDDYMEMORIALCOLLEGE OFENGINEERING AND TECHNOLOGY

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada)

Vinukonda Road, Petlurivaripalem(V), Narasaraopet(M), Guntur (Dt.)522601, AP.

2020-2024

A.M. REDDYMEMORIAL COLLEGEOF ENGINEERING ANDTECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

This is to certify that B. VIJAYA KUMAR REDDY bearing the Registration numbers (20HM1A0405) has carried out the project work present in this report SENSOR BASED IDENTIFICATION SYSTEM FOR TRAIN COLLISION AVOIDANCE for the award of the degree of bachelor of technology in Electronic and Communication Engineering JNTUK in Kakinada under our super vision. The report embodies results of original work and studies are carried out by the student him/her self and the contents of report do not copy from any other report or any other university/institute

Project guide **EXTERNAL EXAMINAR**

Head of the department **Principal**

ACKNOWLEDGEMENT

I wish to express my thanks to various personalities who are responsible for the completion of Project work.

First of all, I would like to thank the **Dr. G. SAMBHA SIVA RAO MTech, Ph.D. Principal,** for providing the online resource and allowing me to carry out this work.

I express my deepfelt gratitude to **Mr. P.S. GANESH KUMAR** MTech., (Ph.D.) Head of the Department of Electronics and Communication Engineering for extend his cooperation and providing the required and for extending his helping hands and valuable suggestion when in need.

I would likely to gratefully acknowledge my guide and encouragement given by my guide **Mr. K. Sanjeev Rao,** M. Tech, Assistant Professor Department of ECE who boosted my morale all the time. His vast experience, profound knowledge and willingness have been a constant source of inspiration for me throughout this project work. My sincere thanks to him/her for evincing keen interest in my endeavor and for very amicable.

I extend my sincere thanks to all other Teaching and Non-Teaching Staff of ECE department for cooperation and encouragement during the course. We have no words to acknowledge the warm affection, constant in spiration and encouragement that I receive from my parents.

I express my special thanks to Librarian and the library staff, for providing the necessary Library facilities.

BY

B. VIJAY KUMAR REDDY **20HM1A0405**

CONTENTS

LIST OF FIGURES

FIGURES **PAGE NO**

CHAPTER-1

INTRODUCTION

The Internet of Things (IoT) is an arrangement of interrelated computing gadgets, mechanical and digital machines, objects, animals or individuals that are given one kind of an identifiers and the capacity to exchange information over a system without requiring human-to-human or human-to-PC communication. IoT is a new concept that has evolved from the convergence of wireless technologies. Wireless communication is the transfer of information or signal between two or more points that are not connected by an electrical conductor. In IoT devices equipped with Wi-Fi allow the machine-to-machine communication. 1978. In the early years, the technology was not yet operational, due to an insufficient number of satellites orbiting the earth. On Jan. 17, 1994, after years of gradual growth, the final of the first 24 satellites was launched, and the GPS system was considered fully operational. Early GPS was designed primarily only for military but in 1996, President Bill Clinton determined that the system would be an asset to civilians as well as the military. This policy change made GPS technology available to the average individual, including fleet managers, who could see the benefit of using the technology to keep tabs on their vehicles. In the early days of fleet tracking, in order to properly track a fleet, each vehicle had to be enabled with a costly GPS.

Nowadays, there is an increase in the number of accidents that happen in the world. As the population is increasing, there is the number of cars increasing on the road that contributes to severe accidents that happen daily. Around 80 per cent of accidents contribute to the loss of many lives. Mostly, the growing countries are being targeted by the day-to-day road accidents. The major reason is the lack of infrastructure, lack of traffic control and accident management. Out of all the developing countries, India has been listed as the country with a higher number of accidents. The most prominent reason for the loss of a life during an accident is the unavailability of immediate help that can save a person's life by a few seconds. The moment an accident has occurred, the life of all passengers travelling in the vehicle is at stake. It all depends on response time that can save their lives by a few minutes or seconds.

According to the statistics, reducing accident delay time by even 1 minute can save 6 per cent of lives. Hence, this response time is very crucial, and it needs to be reduced or at least either improved to save their lives.

To contribute to our society and reduce the number of accidents happening in our day-to-day life, there are several techniques and mechanisms that can drop down the rate of accidents and can save lot lives. Living in a tech world that is growing day by day with new technologies, we can apply these techniques in our society and help them overcome such problems. The Vision of the Internet of Things (IoT) has come out to reach unexpected bounds of today's computing world. It is a concept that not only can impact human's life but also how they function. The heart of IOT is smart sensors without which it would not have existed. These sensors form a vast network for their communication.

They capture minute details of their surroundings and pass this important information to each other. Based on the received information, relevant actions are performed accordingly. It is the latest communication model that imagines the proximate future, in which objects of day-to-day life will be incorporated with microcontrollers for digital communication with the help of appropriate protocol stacks that will make them capable of communication with one another. It is a technology that aims to impart intelligence to devices so that they can smartly connect and perform the necessary actions to eliminate human labor.

It gives an image of the future where non-living objects will be communicating with each other and doing the needful work. In this way, human labor will be eliminated to an extent and the devices will be performing necessary actions. The significance of accident detection and notification system is very prominent for our society. Imagine a situation where an accident happened, it is immediately notified to the emergency services. This will result in the rescue of injured people involved in the accident. As the Internet of Things has witnessed fast growth these days, it has the power to bridge these two situations.

OVER VIEW:

Car accidents that happen daily are the major social problems towards which serious action must be taken. One of the solutions for this domain is the Internet of Things which is the current trend in technology. For this purpose, many authors have worked in this domain by applying this technology. Akriti et al., found many tradeoffs while working with the accidental management system such as high cost, non-portability, false delivery etc. The system faced many shortcomings due to lack of resources. In their technique, they used severity scale to measure the impact of an accident. This reduced load on the cloud server by 30 per cent. Chatrapathi et al., designed a framework that has two components. First one is accident detection and alerting system. The second one is traffic management for the ambulance. The efficient routing algorithm is used to route the ambulance. The technique is feasible for the road junctions with signals. However, it is not applicable to the segments without signals. In paper, Raut and Sachdev proposed a call notification system that consists of X Bee WIFI Module, X Bee Shield, GPS Module and See Arduino. The accident is detected using only crash sensors because of which it gives less accurate results. Ali and Alwan proposed a system that consists of several cases to detect low speed and high-speed car accidents. In case of high speed, if the smartphone's acceleration > 4G, then there is an accident which is identified by the smartphone's application. However, it leads to triggering of false alarm in few cases since mobiles are the subject. In paper, S. R et al. discussed the driver's behavior by analyzing eye blinking with the help of IR sensors. The head movements of the driver are monitored by the accelerometer which is fixed onto the fore head to measure the angles made by the head. This technique is not feasible since it would be uncomfortable for the driver to attach an accelerometer to the forehead every time. Moreover, driver behavior is the only factor that is considered for accident detection. Sandeep et al., introduced a solution for the accidents that are majorly caused by drink and drive case. For this purpose, they used few sensors like touch sensor, heartbeat sensor, alcohol sensor interfaced with Raspberry Pi. In their work, they only considered the situation for the drink and drive cases.

CHAPTER-2

BLOCK DIAGRAM

2.1 BLOCK DIAGRAM OVERVIEW:

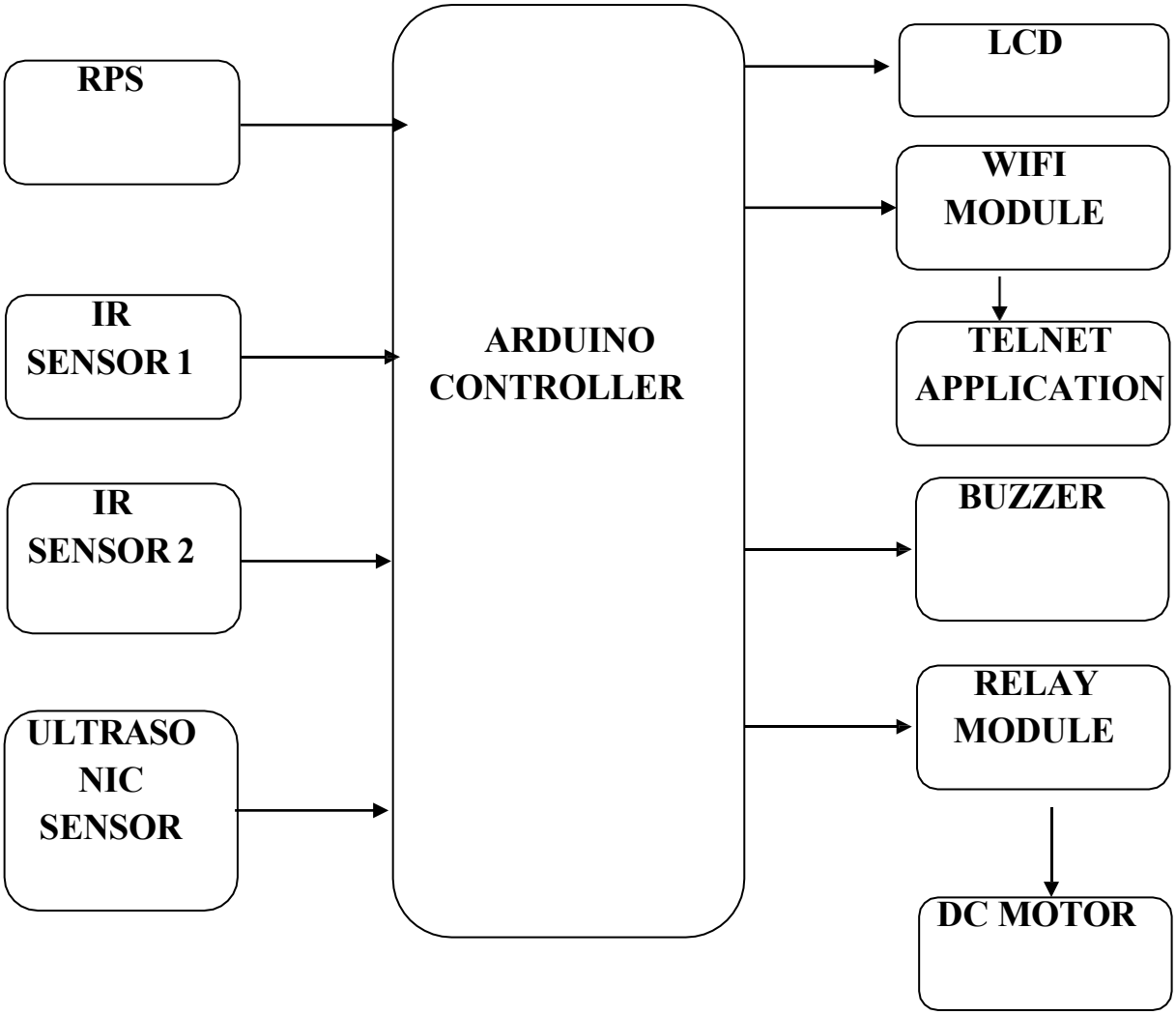


Fig.2.1. block diagram

2.2 POWER SUPPLY:

All digital circuits require regulated power supply. In this article we are going to learn how to get a regulated positive supply from the mains supply.

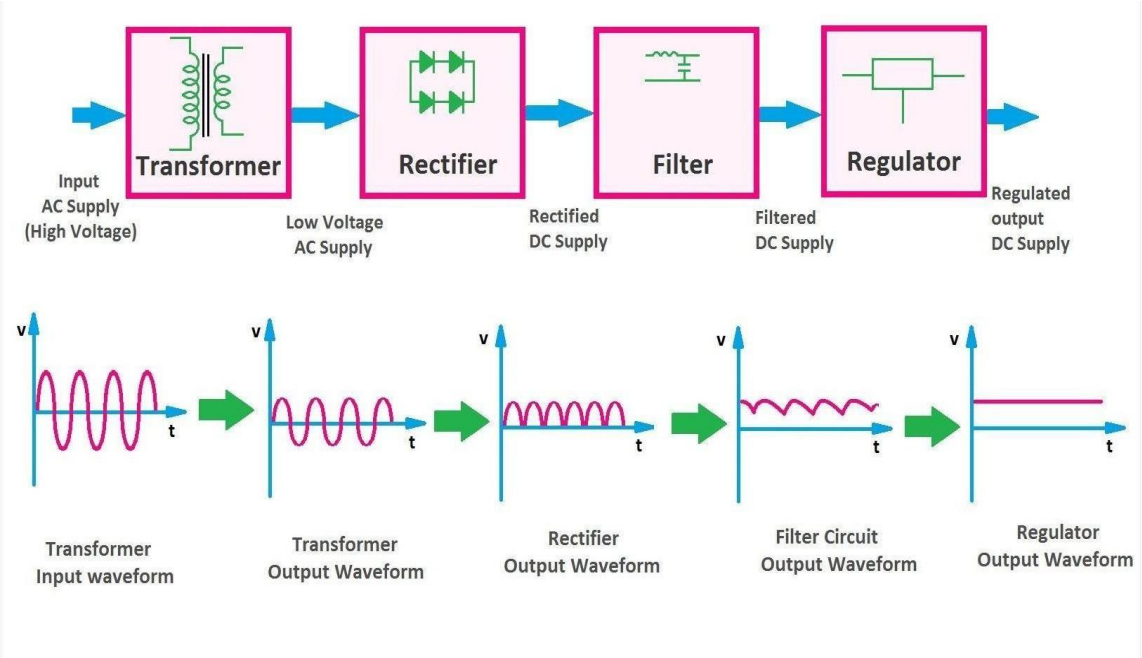


Fig:2.2 shows the basic block diagram of a fixed regulated power supply. Let us go through each block.

TRANSFORMER:

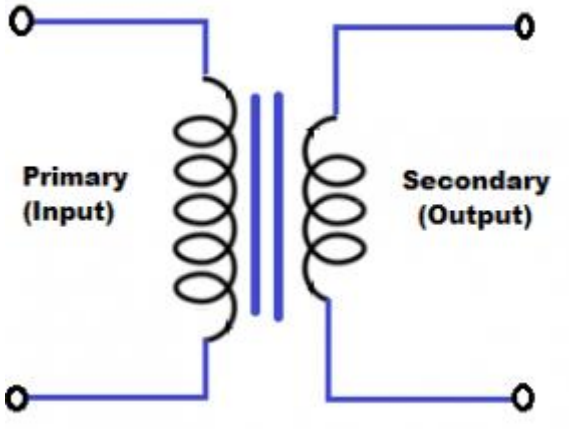


Fig:2.3 Transformer

A transformer consists of two coils also called as "WINDINGS" namely PRIMARY & SECONDARY. They are linked together through inductively coupled electrical conductors also called as CORE. A changing current in the primary causes a change in the Magnetic Field in the core & this in turn induces an alternating voltage in the secondary coil. If load is applied to the secondary then an alternating current will flow through the load.

If we consider an ideal condition then all the energy from the primary circuit will be transferred to the secondary circuit through the magnetic field.

$$P_{primary} = P_{secondary}$$

So

$$I_p V_p = I_s V_s$$

The secondary voltage of the transformer depends on the number of turns in the Primary as well as in the secondary.

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

Rectifier:

A rectifier is a device that converts an AC signal into DC signal. For rectification purpose we use a diode, a diode is a device that allows current to pass only in one direction i.e. when the anode of the diode is positive with respect to the cathode also called as forward biased condition & blocks current in the reversed biased condition.

Rectifier can be classified as follows:

1) Half Wave rectifier:

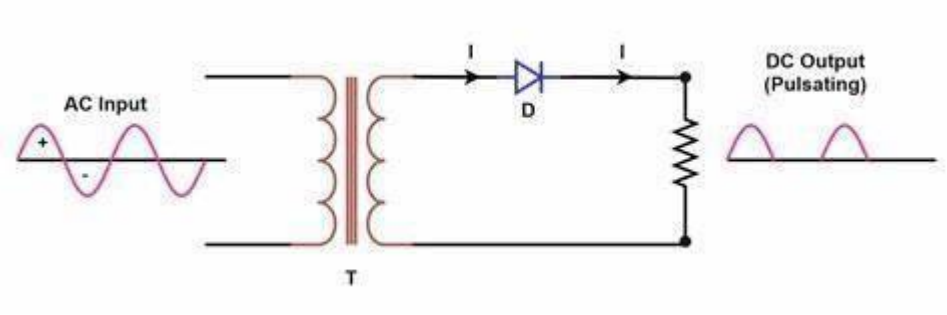


Fig: 2.4: Half wave rectifier

This is the simplest type of rectifier as you can see in the diagram a half wave rectifier consists of only one diode. When an AC signal is applied to it during the positive half cycle the diode is forward biased & current flows through it.

But during the negative half cycle diode is reverse biased & no current flows through it. Since only one half of the input reaches the output, it is very inefficient to be used in power supplies.

2) Full wave rectifier:

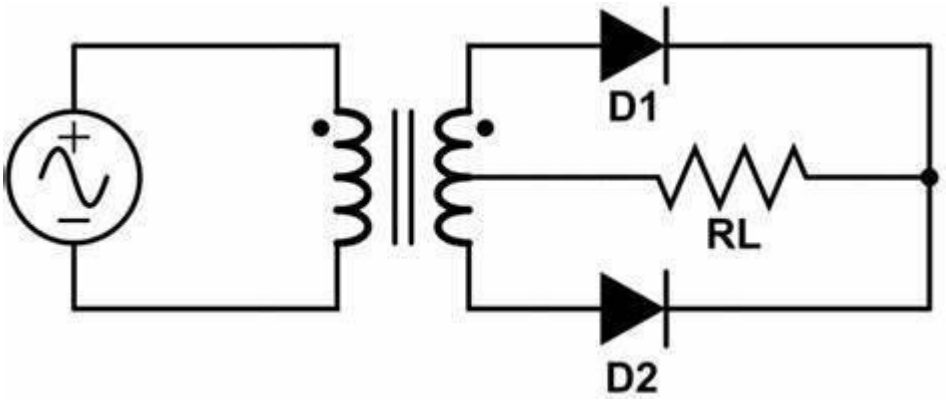


Fig: 2.5 :Full wave rectifier

Half wave rectifier is quite simple but it is very inefficient, for greater efficiency we would like to use both the half cycles of the AC signal. This can be achieved by using a center tapped transformer i.e. we would have to double the size of secondary winding & provide connection to the center. So, during the positive half cycle diode D1 conducts & D2 is in reverse biased condition. During the negative half cycle diode D2 conducts & D1 is reverse biased. Thus, we get both the half cycles across the load.

One of the disadvantages of Full Wave Rectifier design is the necessity of using a center tapped transformer, thus increasing the size & cost of the circuit. This can be avoided by using the Full Wave Bridge Rectifier.

Operation:

a. During the positive half-cycle of the input AC voltage, the upper end of the center-tapped transformer secondary winding becomes positive and the lower end becomes negative.
b. Diodes D1 and D2 are forward biased, allowing current to flow through the load resistor (RL) in the direction of the arrow.
c. During the negative half-cycle of the input AC voltage, the polarity across the transformer secondary winding reverses.

3) Bridge Rectifier:

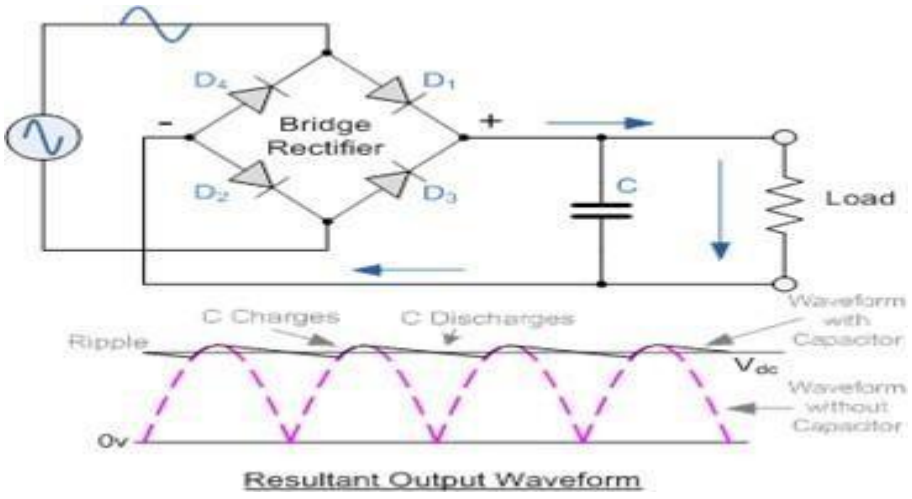


Fig:2.6:Bridge Rectifier

As the name suggests it converts the full wave i.e. both the positive & the negative half cycle into DC thus it is much more efficient than Half Wave Rectifier & that too without using a center tapped transformer thus much more cost effective than Full Wave Rectifier.

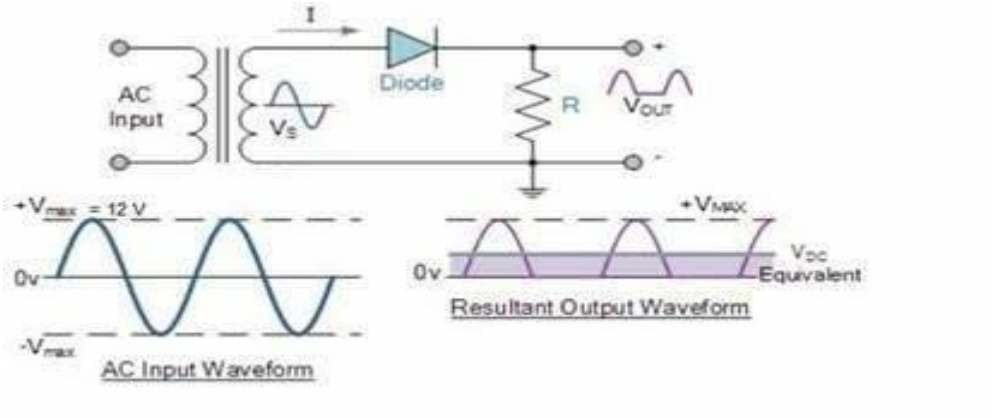


Fig:2.7:Bridge Rectifier Half wave Rectifier

Full Bridge Wave Rectifier consists of four diodes namely D1, D2, D3 and D4. During the positive half cycle diodes D1 & D4 conduct whereas in the negative half cycle diodes D2 & D3 conduct thus the diodes keep switching the transformer connections so we get positive half cycles in the output.

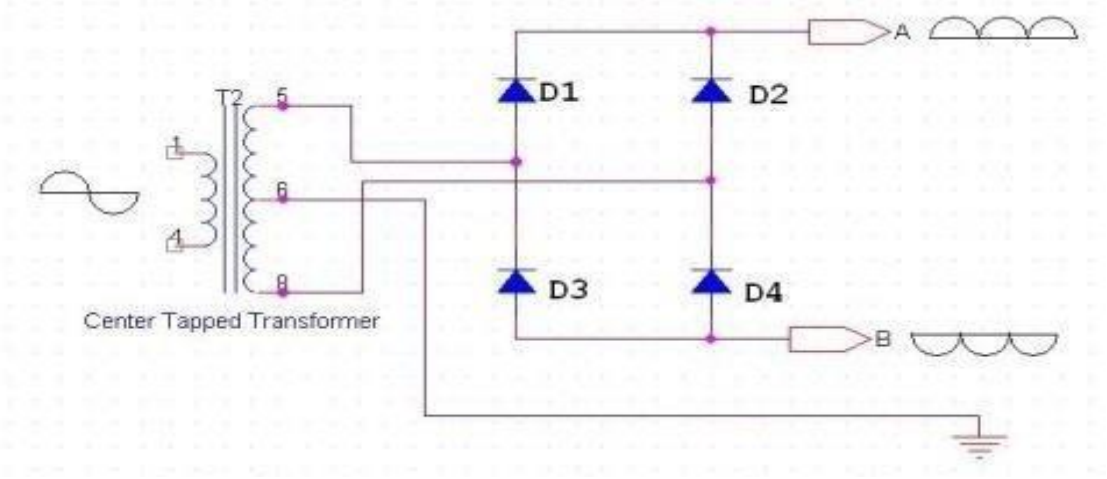


Fig: 2.8: Bridge rectifier Negative half wave rectifier

If we use a center tapped transformer for a bridge rectifier, we can get both positive & negative half cycles which can thus be used for generating fixed positive & fixed negative voltages.

2.3 FILTER CAPACITOR:

Even though half wave & full wave rectifier give DC output, none of them provides a constant output voltage. For this we require to smoothen the waveform received from the rectifier. This can be done by using a capacitor at the output of the rectifier this capacitor is also called as "FILTER CAPACITOR" or "SMOOTHING CAPACITOR" or "RESERVOIR CAPACITOR". Even after using this capacitor a small amount of ripple will remain.

We place the Filter Capacitor at the output of the rectifier the capacitor will charge to the peak voltage during each half cycle then will discharge its stored energy slowly through the load while the rectified voltage drops to zero, thus trying to keep the voltage as constant as possible.

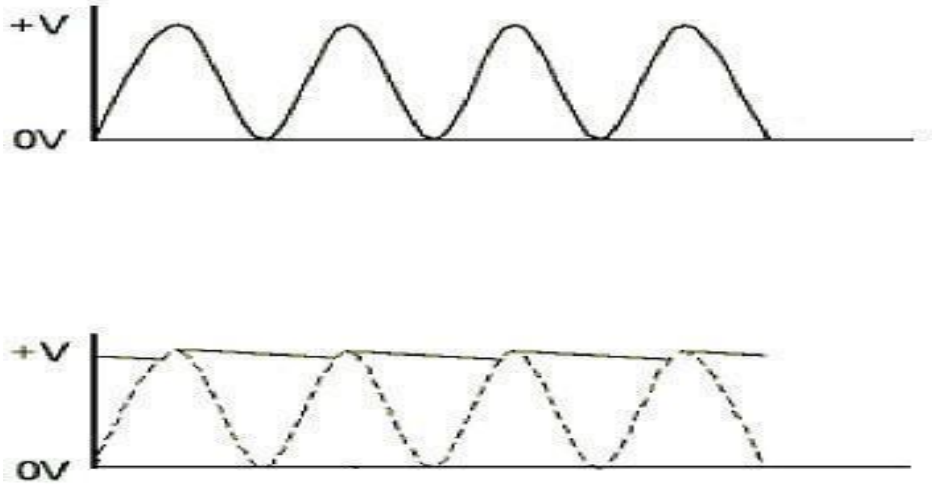


Fig: 2.9 :Filter capacitor wave forms

If we go on increasing the value of the filter capacitor then the Ripple will decrease. But then the costing will increase. The value of the Filter capacitor depends on the current consumed by the circuit, the frequency of the waveform & the accepted ripple.

$$C = \frac{V_r F}{I}$$

Where,

Vr= accepted ripple voltage. (should not be more than 10% of the voltage)

I= current consumed by the circuit in Amperes.

F= frequency of the waveform. A half wave rectifier has only one peak in one cycle so F=25hz

Whereas a full wave rectifier has Two peaks in one cycle so F=100hz.

VOLTAGE REGULATOR:

A Voltage regulator is a device which converts varying input voltage into a constant regulated output voltage. Voltage regulator can be of two types

1) Linear Voltage Regulator

Also called as Resistive Voltage regulator because they dissipate the excessive voltage resistively as heat.

2) Switching Regulators.

They regulate the output voltage by switching the Current ON/OFF very rapidly. Since their output is either ON or OFF it dissipates very low power thus achieving higher efficiency as compared to linear voltage regulators. But they are more complex & generate high noise due to their switching action. For low level of output power switching regulators tend to be costly but for higher output wattage they are much cheaper than linear regulators.

The most commonly available Linear Positive Voltage Regulators are the 78XX series where the XX indicates the output voltage. And 79XX series is for Negative Voltage Regulators.

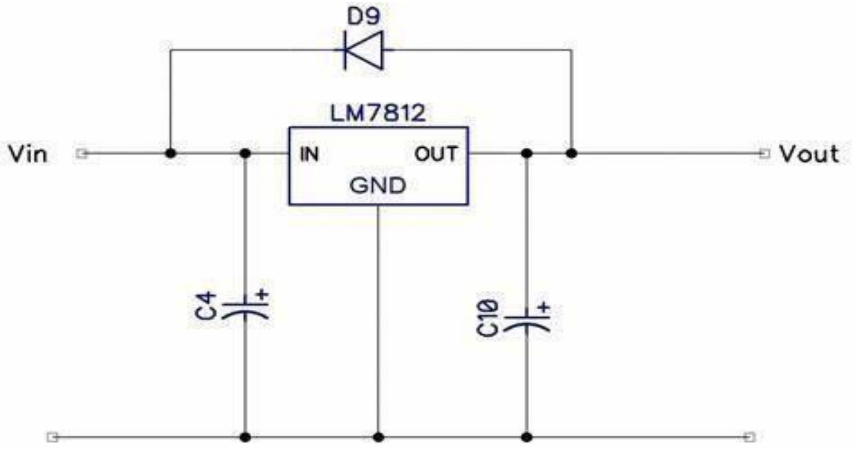


Fig:2.10: Voltage Regulator

After filtering the rectifier output the signal is given to a voltage regulator. The maximum input voltage that can be applied at the input is 35V.Normally there is a 2-3 Volts drop across the regulator so the input voltage should be at least 2-3 Volts higher than the output voltage. If the input voltage gets below the Vmin of the regulator due to the ripple voltage or due to any other reason the voltage regulator will not be able to produce the correct regulated voltage.

Circuit diagram:

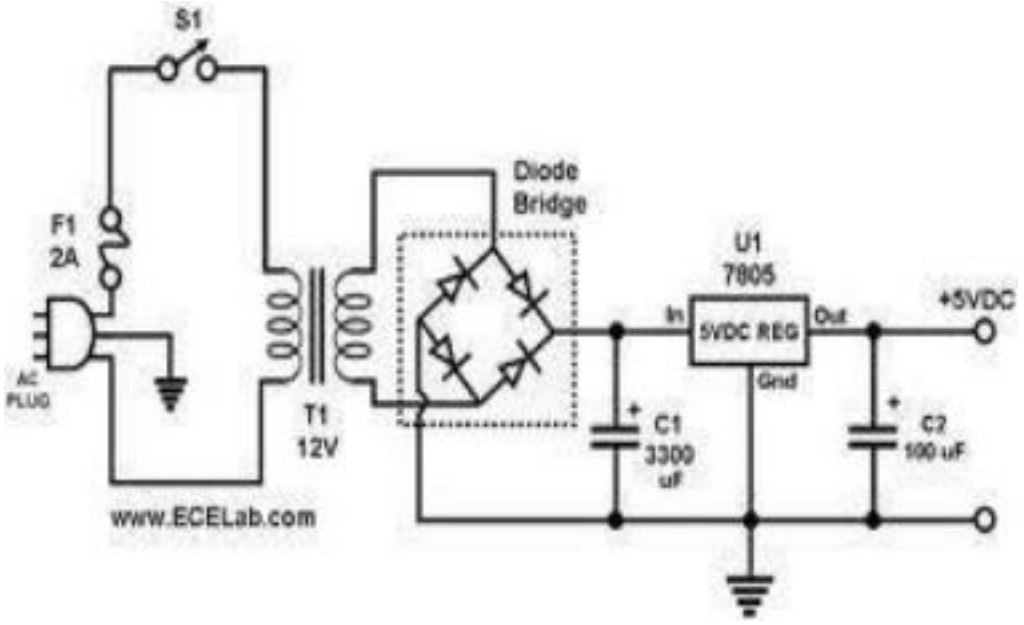


Fig: 2.11. Circuit Diagram of power supply

IC 7805:

7805 is an integrated three-terminal positive fixed linear voltage regulator. It supports an input voltage of 10 volts to 35 volts and output voltage of 5 volts. It has a current rating of 1 amp although lower current models are available. Its output voltage is fixed at 5.0V. The 7805 also has a built-in current limiter as a safety feature. 7805 is manufactured by many companies, including National Semiconductors and Fairchild Semiconductors.

The 7805 will automatically reduce output current if it gets too hot. The last two digits represent the voltage; for instance, the 7812 is a 12-volt regulator. The 78xx series of regulators is designed to work in complement with the 79xx series of negative voltage regulators in systems that provide both positive and negative regulated voltages, since the 78xx series can't regulate negative voltages in such a system.

The 7805 & 78 is one of the most common and well-known of the 78xx series regulators, as it's small component count and medium-power regulated 5V make it useful for powering TTL devices.

SPECIFICATIONS	IC 7805
Vout	5V
V_{ein} - V_{out} Difference	5V - 20V
Operation Ambient Temp	0 - 125°C
Output I_{max}	1A

Table 2.1. Specifications of IC7805

2.4. IR SENSOR:

IR transmitter and receiver:

Basics of IR transmitter and receiver transmitter and receiver are commonly used in engineering projects for remote control of objects. In particularly, in Robotic system uses transmitter and receiver. Here i would like to describe the basics if IR transmitter and receiver

Basics of IR transmitter:

An electroluminescent IR LED is a product which requires care in use. IR LEDs are fabricated from narrow band hetero structures with energy gap from 0.25 to 0.4 eV. Infra red transmitter emits IR rays in planar wave front manner. Even though infra red rays spread in all directions, it propagates along straight line in forward direction. IR rays have the characteristics of producing secondary wavelets when it collides with any obstacles in its path. This property of IR is used here.

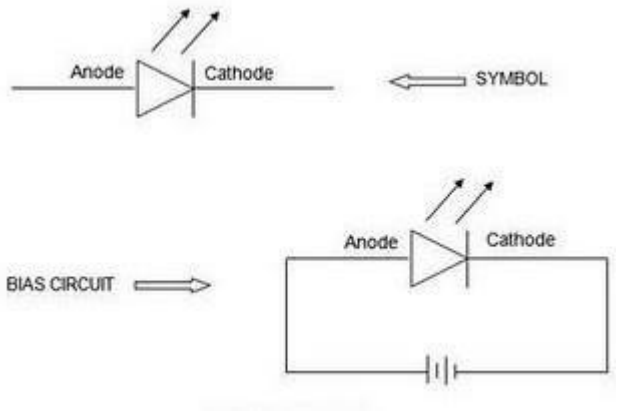


Fig:2.12:IR Transmitter

When IR rays gets emitted from LED, it moves in the direction it is angled. When any obstacle interferes in the path, the IR rays get cut and it produces secondary wavelets which propagates mostly in return direction or in a direction opposite to that of the primary waves, which produces the net result like reflection of IR rays.

Basics of IR receiver:

Infrared photo receiver is a two terminal PN junction device, which operates in a reverse bias. It has a small transparent window, which allows light to strike the PN junction. A photodiode is a type of photo detector capable of converting light into either current or voltage, depending upon the mode of operation. Most photodiodes will look similar to a light emitting diode.

They will have two leads, or wires, coming from the bottom. The shorter end of the two is the cathode, while the longer end is the anode.

A photodiode consists of PN junction or PIN structure. When a photon of sufficient energy strikes the diode, it excites an electron thereby creating a mobile electron and a positively charged electron hole. If the absorption occurs in the junction's depletion region, or one diffusion length away from it, these carriers are swept from the junction by the built-in field of the depletion region. Thus, holes move toward the anode, and electrons toward the cathode, and a photocurrent is produced.

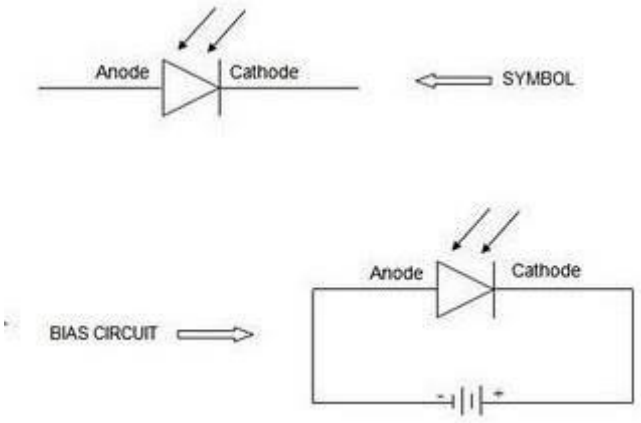


Fig:2.13:IR Receiver

Working of infrared communication:

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

The entire circuit consists of two sections named as

1. Transmitter section and
2. Receiver section

1. Transmitter section:

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

2. Receiver section:

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

We can increase the distance between the IR transmitter and receiver just by placing the lens between them. After connecting the IR transmitter and receiver circuit, we can get the output by applying 6V Power supply to the circuit.

We can use this circuit with any application very simply. For example, a buzzer circuit is placed at the output of IR circuit, when the signals are interrupted, the buzzer produces sound. Both the transmitter and receiver parts can be mounted on a single bread board or PCB. The infrared receiver must be placed behind the IR Led to avoid false indication due to infrared leakage. An object moving nearby actually reflects the IR rays emitted by the IR Led.

PHOTO DIODE:

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

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

Fig :2.14: Photodiode schematic symbol

Principle of operation:

A photodiode is a p-n junction or p-i-n structure. When a photon of sufficient energy strikes the diode, it excites an electron thereby creating a mobile electron and a positively charged electron hole. If the absorption occurs in the junction's depletion region, or one diffusion length away from it, these carriers are swept from the junction by the built-in field of the depletion region, producing a photocurrent. Photodiodes can be used under either zero bias (photovoltaic mode) or reverse bias (photoconductive mode). In zero bias, light falling on the diode causes a current across the device, leading to forward bias which in turn induces "dark current" in the opposite direction to the photocurrent. This is called the photovoltaic effect, and is the basis for solar cells in fact; a solar cell is just a large number of big photodiodes. Reverse bias induces only little current (known as saturation or back current) along its direction. But a more important effect of reverse bias is widening of the depletion layer (therefore expanding the reaction volume) and strengthening the photocurrent. Circuits based on this effect are more sensitive to light than ones based on the photovoltaic effect and also tend to have lower capacitance, which improves the speed of their time response. On the other hand, the photovoltaic mode tends to exhibit less electronic noise.

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

Features: Critical performance parameters of a photodiode include:

1. Responsivity:

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

3. **Dark current:**

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

3. Noise-equivalent power:

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

Applications:

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

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

1. Due to the intrinsic layer, a PIN photodiode must be reverse biased (Vr). The Vr increases the depletion region allowing a larger volume for electron-hole pair production, and reduces the capacitance thereby increasing the bandwidth.
2. The Vr also introduces noise current, which reduces the S/N ratio. Therefore, a reverse bias is recommended for higher bandwidth applications and/or applications where a wide dynamic range is required.
3. A PN photodiode is more suitable for lower light applications because it allows for unbiased operation.

2.5. WIFI MODULE:

The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability, produced by Express if Systems[1] in Shanghai, China. The chip first came to the attention of Western makers in August 2014 with the ESP-01 module, made by a third-party manufacturer Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands. However, at first there was almost no English-language documentation on the chip and the commands it accepted.[2] The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, the chip, and the software on it, as well as to translate the Chinese documentation. The ESP8285 is an ESP8266 with 1 MiB of built-in flash, allowing the building of single-chip devices capable of connecting to Wi-Fi.

P8266 comes with capabilities of

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16 GPIO),
- Inter-Integrated Circuit (I^2C) serial communication protocol,
- analog-to-digital conversion (10-bit ADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I^2S (Inter-IC Sound) interfaces with DMA (Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and
- pulse-width modulation (PWM).

It employs a 32-bit RISC CPU based on the Ten silica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External flash memory can be accessed through SPI.

ESP8266 module is low-cost standalone wireless transceiver that can be used for end-point IoT developments. To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having specified Baud rate.

There are many third-party manufacturers that produce different modules based on this chip. So, the module comes with different pin availability options like,

- ESP-01 comes with 8 pins (2 GPIO pins) – PCB trace antenna. (shown in above figure)
- ESP-02 comes with 8 pins, (3 GPIO pins) – U-FL antenna connector.
- ESP-03 comes with 14 pins, (7 GPIO pins) – Ceramic antenna.
- ESP-04 comes with 14 pins, (7 GPIO pins) – No ant etc.

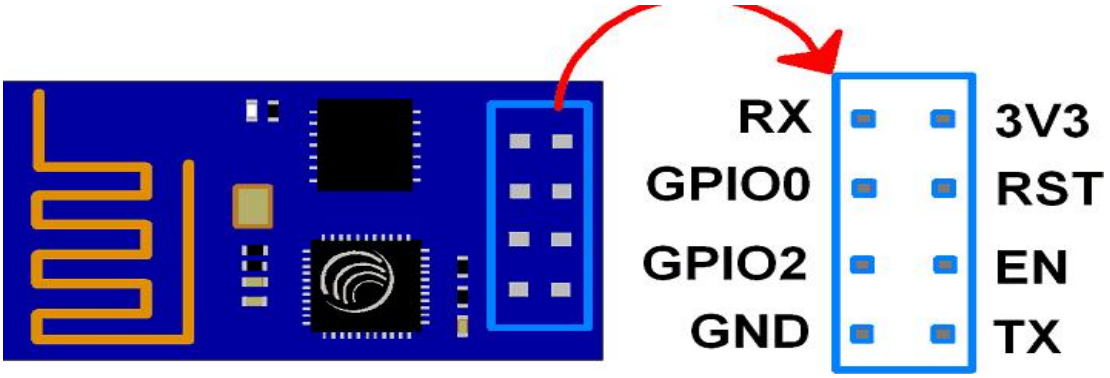


Fig: 2.15: ESP8266-01 Module pins and WIFI Module

3V3: - 3.3 V Power Pin.

GND: - Ground Pin.

RST: - Active Low Reset Pin.

EN: - Active High Enable Pin.

TX: - Serial Transmit Pin of UART.

RX: - Serial Receive Pin of UART.

GPIO0 & GPIO2:

General Purpose I/O Pins. These pins decide what mode (boot or normal) the module starts up in. It also decides whether the TX/RX pins are used for Programming the module or for serial I/O purpose.

To program the module using UART, Connect GPIO0 to ground and GPIO2 to VCC or leave it open. To use UART for normal Serial I/O leave both the pins open (neither VCC nor Ground).

THE CHIPS:

The ESP8266 series, or family, of Wi-Fi chips is produced by Espressif Systems, a fabless semiconductor company operating out of Shanghai, China. The ESP8266 series presently includes the ESP8266EX and ESP8285 chips.

ESP8266EX (simply referred to as ESP8266) is a system-on-chip (SoC) which integrates a 32-bit Tensilica microcontroller, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules into a small package. It provides capabilities for 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2), general-purpose input/output (16 GPIO), Inter-Integrated Circuit (I^2C), analog-to-digital conversion (10-bit ADC), Serial Peripheral Interface (SPI), I^2S interfaces with DMA (sharing pins with GPIO), UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and pulse-width modulation (PWM). The processor core, called "L106" by Espressif, is based on Tensilica's Diamond Standard 106Micro 32-bit processor controller core and runs at 80 MHz (or over clocked to 160 MHz). It has a 64 KiB boot ROM, 32 KiB instruction RAM, and 80 KiB user data RAM. (Also, 32 KiB instruction cache RAM and 16 KiB ETS system data RAM.).

- ESP8266EX Datasheet
- ESP8266 Technical Reference

Vendors have consequently created a multitude of compact printed circuit board modules based around the ESP8266 chip. Some of these modules have specific identifiers, including monikers such as "ESP-WROOM-02" and "ESP-01" through "ESP-14".

2.6. LCD:

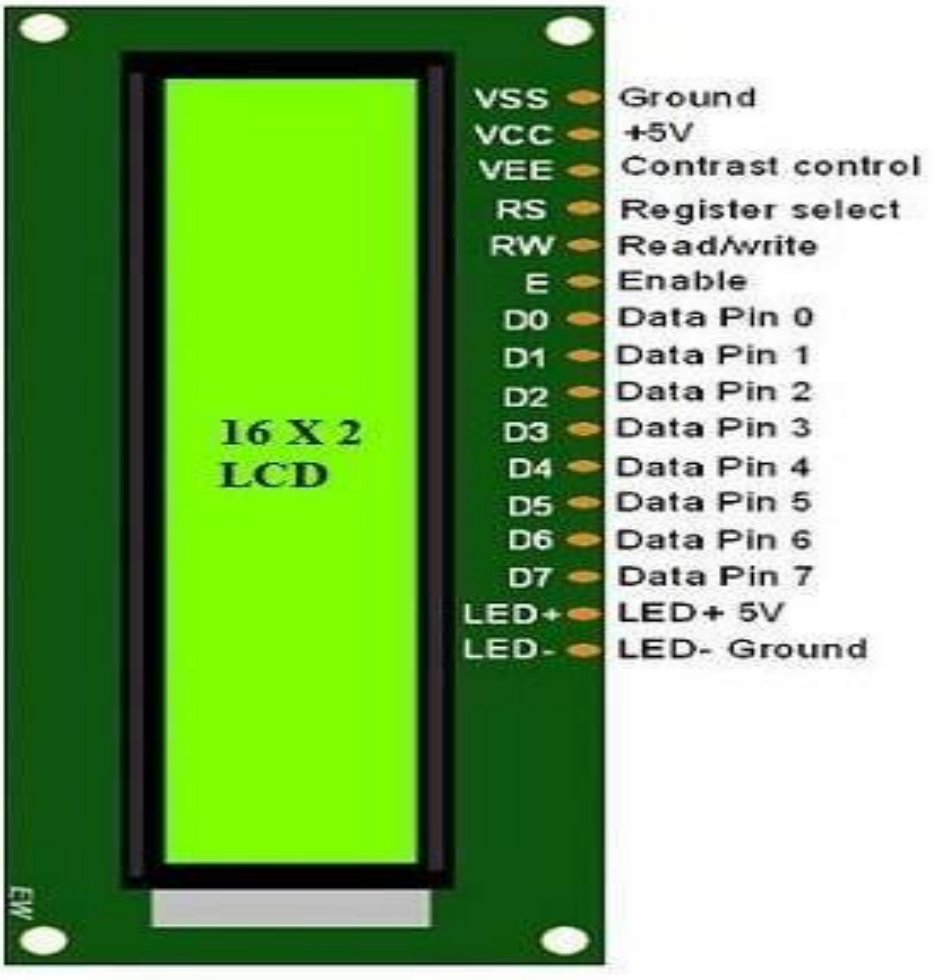


Fig: 2.16.LCD (LIDUID CRYSTAL DISPLAY)

1. VSS (Ground): Ground connection.
2. VDD (+5V): Power supply voltage.
3. V0: Contrast adjustment (varying voltage for adjusting display contrast).
4. RS (Register Select): Selects between command/data input.
5. RW (Read/Write): Selects between read and write mode. Often tied to ground for write-only operation.
6. E (Enable): Strobe signal for initiating data transfer.
7. 7-14. D0-D7: Data bus for transferring data to/from the LCD.
8. A (Anode): Backlight anode (positive).
9. K (cathode): Backlight cathode (negative).

Here's a brief explanation of the use of each pin:

1. VSS (Ground): This pin is connected to ground (0V) and completes the circuit for electrical current to flow.
2. VDD (+5V): This pin is connected to a +5V power supply and provides the operating voltage for the LCD.
3. V0: This pin is used to adjust the contrast of the display. By applying a varying voltage to this pin, you can control the contrast of the characters displayed on the screen.
4. RS (Register Select): This pin selects between command mode and data mode. When RS is low (0), the data sent to the LCD is treated as a command (such as clear screen, move cursor, etc.). When RS is high (1), the data sent to the LCD is treated as character data to be displayed on the screen.

5. RW (Read/Write): This pin selects the direction of data transfer between the microcontroller and the LCD. When RW is low (0), data is written to the LCD (write mode). When RW is high (1), data is read from the LCD (read mode). In many applications, this pin is tied to ground to force write-only operation.

6. E (Enable): This pin initiates data transfer. When E transitions from low to high, the LCD reads the data present on its data bus.

7. 7-14. D0-D7: These pins constitute the data bus and are used to transfer data between the microcontroller and the LCD. When writing data to the LCD, the microcontroller sends character codes or commands to be displayed or executed.

8. A (Anode): This pin is connected to the positive terminal of the backlight LED. It provides power to the backlight, allowing the characters on the display to be visible in low-light conditions.

9. K (Cathode): This pin is connected to the negative terminal of the backlight LED. It completes the circuit for the backlight, allowing current to flow through the LED and produce light

CHAPTER-3

ARDUINO CONTROLLER

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control both physically and digitally. Its products are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors.

The name Arduino comes from a bar in Ivrea, Italy, 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.

Fig.3.1. Hardware image.

History:

The Arduino project was started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller at a cost of $50, a considerable expense for many students. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas. Casey Reas is known for co-creating, with Ben Fry, the Processing development platform. The project goal was to create simple, low-cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartillas, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they forked the project and renamed it Arduino.

The initial Arduino core team consisted of Massimo Banzi, David Cuartillas, Tom Igoe, Gianluca Martino, and David Mellis, but Barragán was not invited to participate.

Following the completion of the Wiring platform, lighter and less expensive versions were distributed in the open-source community.

It was estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands. In October 2016, Federico Musto, Arduino's former CEO, secured a 50% ownership of the company. In April 2017, Wired reported that Musto had "fabricated his academic record.... On his company's website, personal LinkedIn accounts, and even on Italian business documents, Musto was until recently listed as holding a PhD from the Massachusetts Institute of Technology. In some cases, his biography also claimed an MBA from New York University." Wired reported that neither University had any record of Musto's attendance, and Musto later admitted in an interview with Wired that he had never earned those degrees. Around that same time, Massimo Banzi announced that the Arduino Foundation would be "a new beginning for Arduino." But a year later, the Foundation still hasn't been established, and the state of the project remains unclear. The controversy surrounding Musto continued when, in July 2017, he reportedly pulled many Open source licenses, schematics, and code from the Arduino website, prompting scrutiny and outcry. In October 2017, Arduino announced its partnership with ARM Holdings (ARM). The announcement said, in part, "ARM recognized independence as a core value of Arduino ... without any lock-in with the ARM architecture." Arduino intends to continue to work with all technology vendors and architectures.

OPERATION WITH PINS:

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in -Arduino.

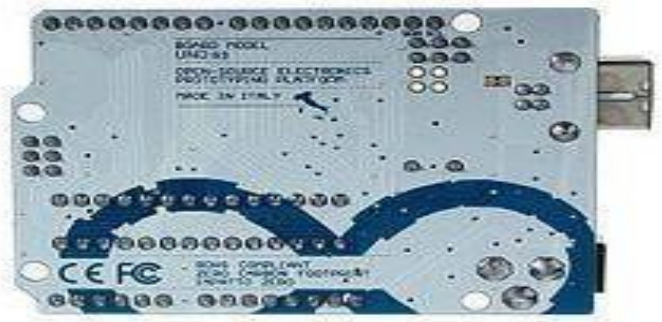


Fig.3.2. Back side of module.

Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8,[24] ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lilypad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino UNO is the Opti boot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header.

Other variants, such as the Arduino Mini and the unofficial Board Arduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods.

When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used. The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Decimal iLab, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Board Arduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

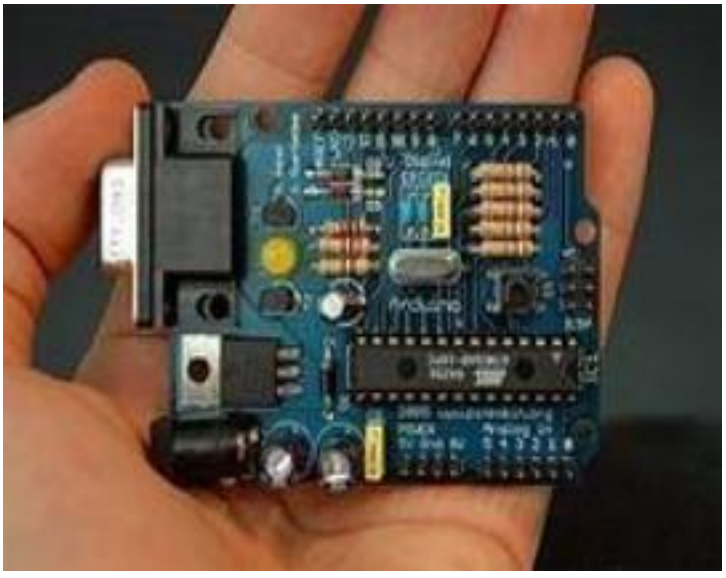


Fig.3.3. Arduino board.

1	**1. Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works

	fine with 3.3 volt and 5 volts. • GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If

this light does not turn on, then there is something wrong with the connection.

TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

Digital I/O

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.

AREF

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

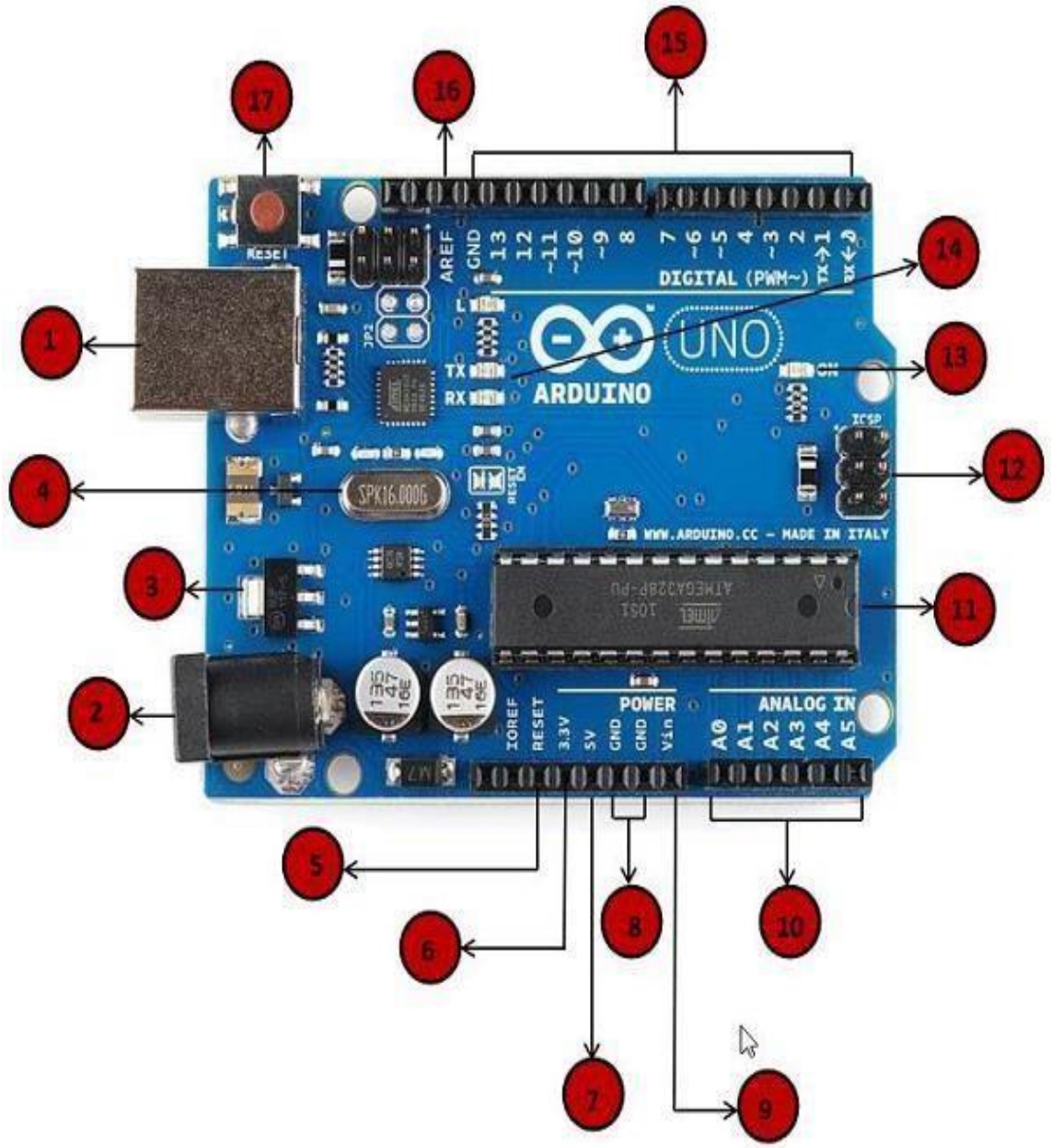


Fig.3.4. Pin explanation.

CHAPTER-4

SOFTWARE EXPLANATION

4.1 : Introduction:

This project is implemented using following software's:

- Express PCB – for designing circuit
- Arduino IDE compiler - for compilation part
- Proteus 7 (Embedded C) – for simulation part

PCB BOARD:

a. User-Friendly Interface: Express PCB offers an intuitive and easy-to-use interface, making it accessible to users with varying levels of experience in PCB design.

b. Schematic Capture: Users can draw schematics of their electronic circuits using the built-in schematic capture tool. This feature allows for the creation of circuit diagrams to visualize the connections between components.

c. PCB Layout: Express PCB enables users to create PCB layouts by placing components onto the board and routing traces to connect them. The software includes tools for arranging components, routing traces, and defining board dimensions.

d. Component Library: The software provides a library of pre-made components, including resistors, capacitors, ICs, connectors, and more. Users can also create custom components and add them to the library.

e. Design Rule Checks (DRC): Express PCB includes design rule checks to ensure that the PCB design meets certain criteria and specifications, such as minimum trace width, clearance, and hole sizes.

f. Output Generation: Once the PCB design is complete, users can generate output files for manufacturing. This typically includes Gerber files for fabrication and a Bill of Materials (BOM) for sourcing components.

g. Affordability: Express PCB is known for its affordability, making it a popular choice for hobbyists and small-scale projects.

The software offers a free version with limited features and a paid version with additional capabilities.

Arduino IDE compiler:

The term "Arduino IDE compiler" typically refers to the compilation process that occurs within the Arduino Integrated Development Environment (IDE) when you verify or upload your code to an Arduino board. Here's a breakdown of what happens during this process:

1. Code Compilation: When you write your code in the Arduino IDE, you're essentially writing in the Arduino programming language, which is based on C and C++. The IDE includes a built-in compiler that translates your human-readable code into machine-readable instructions (binary code) that the microcontroller can understand.

2. Error Checking: Before compilation, the Arduino IDE performs syntax checking and error detection on your code. If there are any syntax errors or other issues in your code, the IDE will highlight them and display error messages in the console at the bottom of the window.

3. Translation to Machine Code: Once the code is error-free, the Arduino IDE compiles it into machine code specific to the target microcontroller. This process involves translating your high-level code into low-level instructions that the microcontroller can execute.

4. Generating HEX File: After successful compilation, the Arduino IDE generates a HEX file, which contains the compiled machine code in a format that can be understood by the microcontroller. This HEX file is what gets uploaded to the Arduino board.

Upload to Arduino: Finally, if you choose to upload your code to an Arduino board, the Arduino IDE uses a bootloader (a small piece of software pre-loaded on the Arduino board) to transfer the compiled HEX file from your computer to the microcontroller's flash memory.

This process typically occurs over a USB connection between your computer and the Arduino board.

4.2 : The Interface:

When a project is first started you will be greeted with a yellow outline. This yellow outline is the dimension of the PCB. Typically, after positioning of parts and traces, move them to their final position and then crop the PCB to the correct size. However, in designing a board with a certain size constraint, crop the PCB to the correct size before starting.

Fig: 4.1 show the toolbar in which each button has the following functions:

Fig: 4.1 Tool bar necessary for the interface

The select tool: It is fairly obvious what this does. It allows you to move and manipulate parts. When this tool is selected the top toolbar will show buttons to move traces to the top / bottom copper layer, and rotate buttons. **The zoom to selection tool:** does just that.

The place pad: button allows you to place small soldier pads which are useful for board connections or if a part is not in the part library but the part dimensions are available. When this tool is selected the top toolbar will give you a large selection of round holes, square holes and surface mount pads.

The place component: tool allows you to select a component from the top toolbar and then by clicking in the workspace places that component in the orientation chosen using the buttons next to the component list. The components can always be rotated afterwards with the select tool if the orientation is wrong.

The place trace: tool allows you to place a solid trace on the board of varying thicknesses. The top toolbar allows you to select the top or bottom layer to place the trace on.

The Insert Corner in trace: button does exactly what it says. When this tool is selected, clicking on a trace will insert a corner which can be moved to route around components and other traces. The remove a trace button is not very important since the delete key will achieve the same result.

4.3 : Design Considerations:

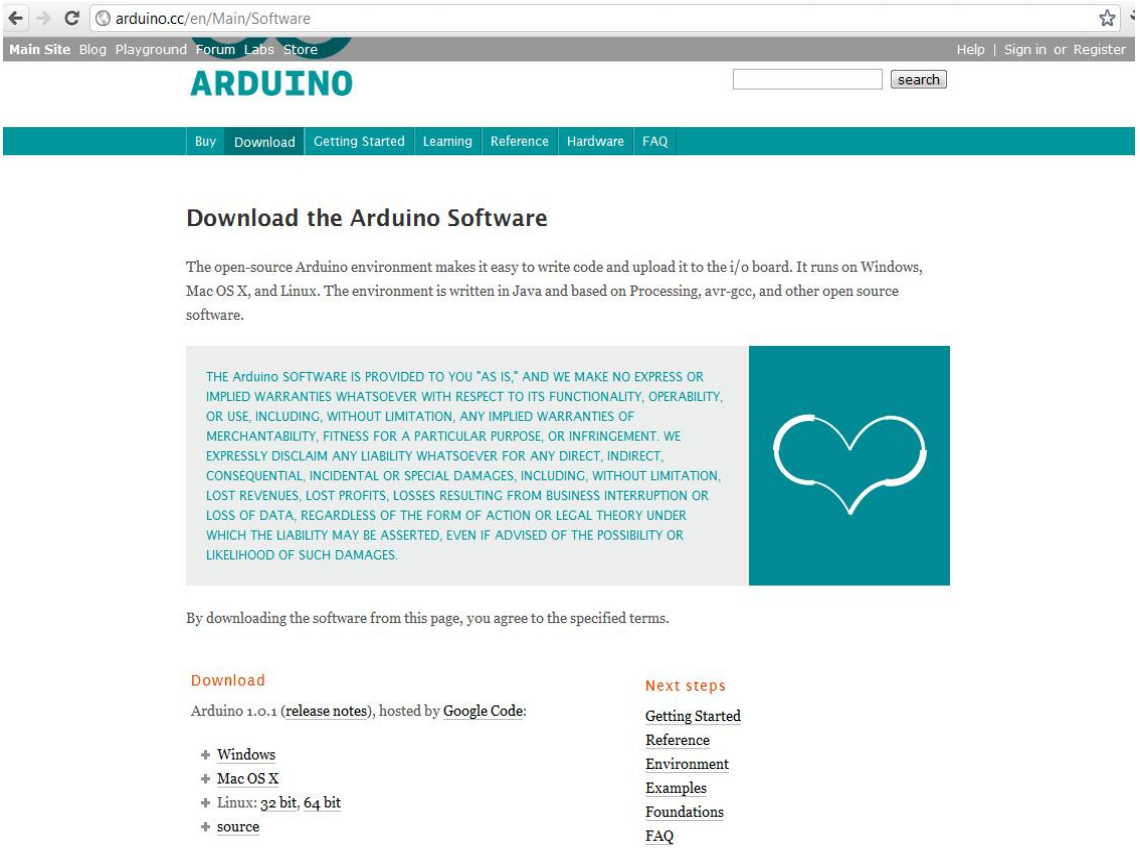
Before starting a project there are several ways to design a PCB and one must be chosen to suit the project's needs. Single sided, or double sided?

When making a PCB you have the option of making a single sided board, or a double-sided board. Single sided boards are cheaper to produce and easier to etch, but much harder to design for large projects. If a lot of parts are being used in a small space it may be difficult to make a single sided board without jump ring over traces with a cable. While there's technically nothing wrong with this, it should be avoided if the signal travelling over the traces is sensitive (e.g. audio signals).

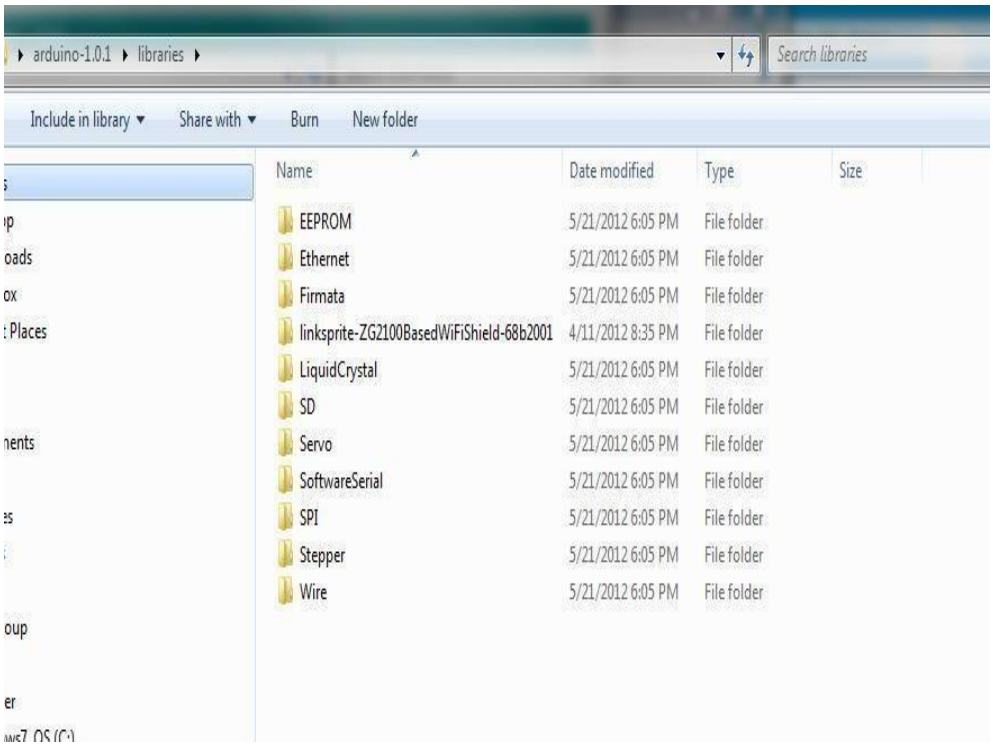
A double-sided board is more expensive to produce professionally, more difficult to etch on a DIY board, but makes the layout of components a lot smaller and easier. It should be noted that if a trace is running on the top layer, check with the components to make sure you can get to its pins with a soldering iron. Large capacitors, relays, and similar parts which don't have axial leads can NOT have traces on top unless boards are plated professionally.

CHAPTER-5

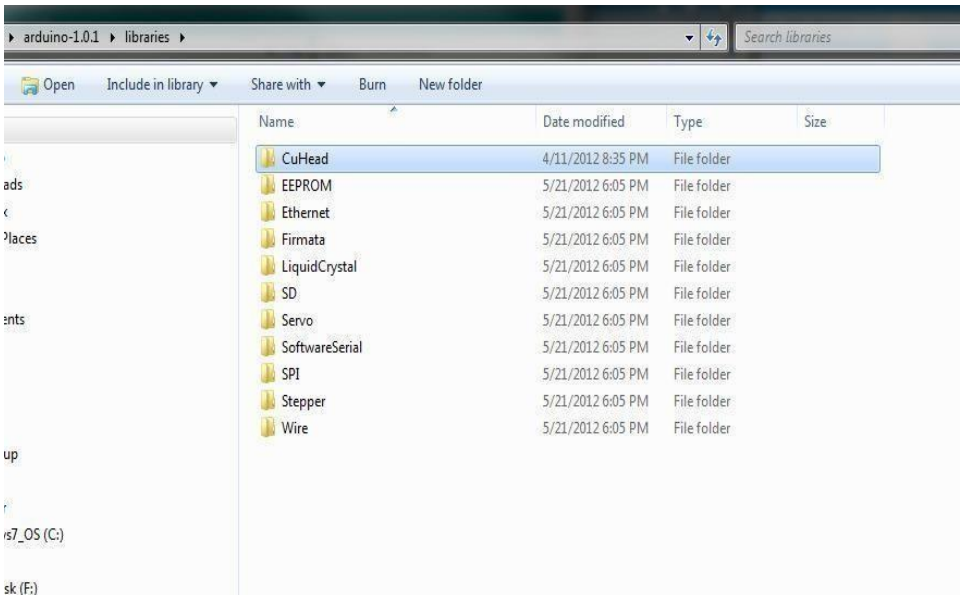
AURDINO COMPILING:



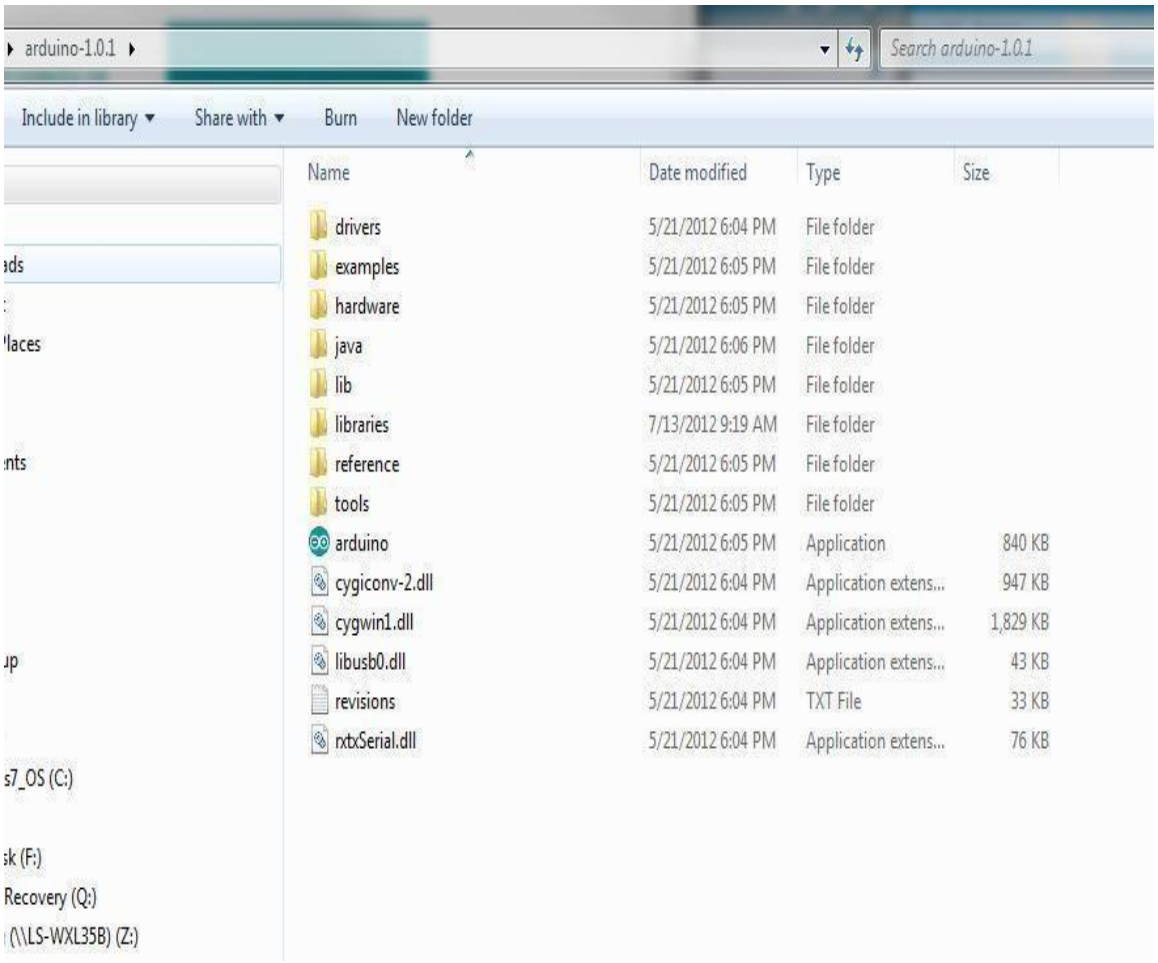
In next step download library



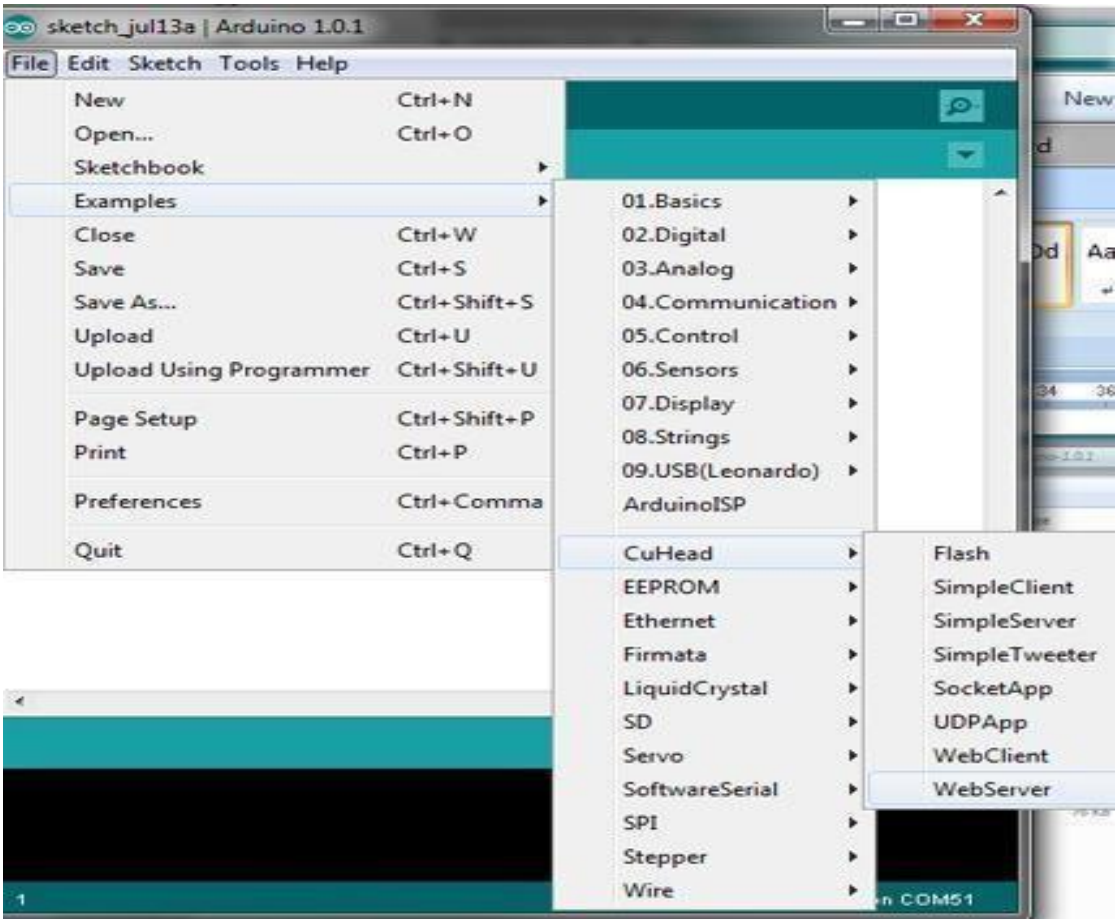
As Arduino doesn't recognize the directory name, please rename it



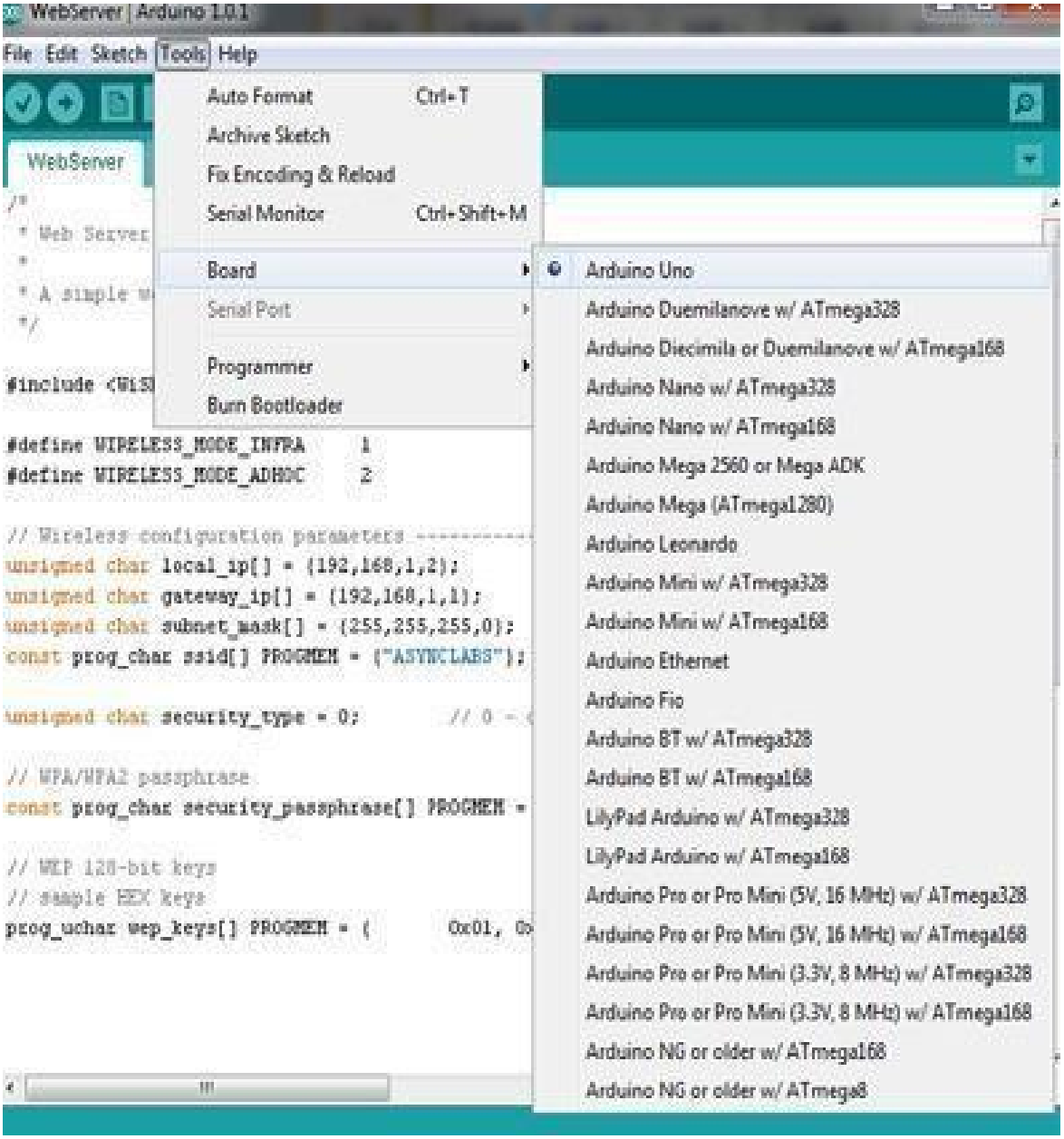
Launch Arduino by double click "Arduino" below



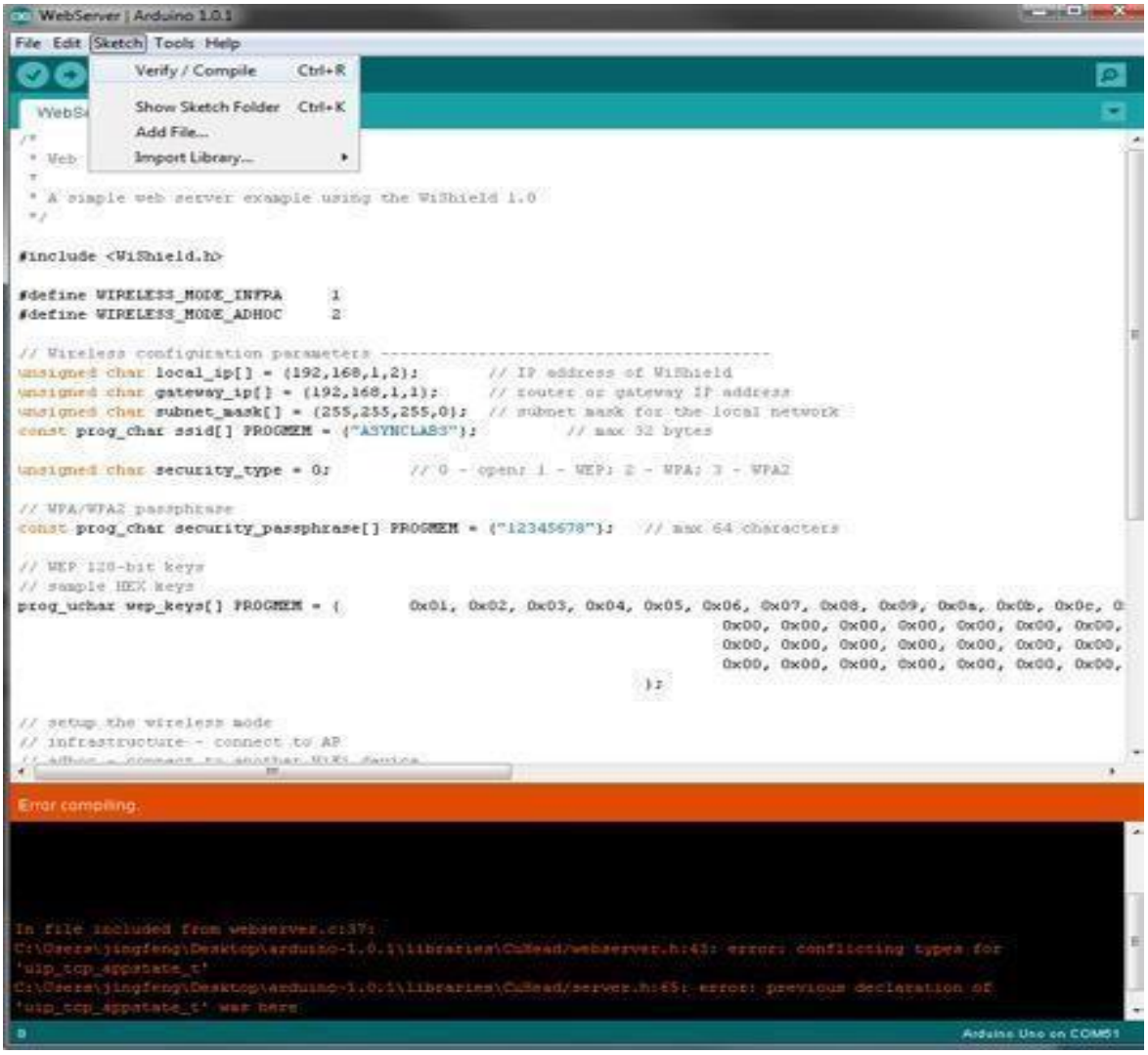
One example


sketch_jul13a | Arduino 1.0.1
File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
01.Basics
02.Digital
03.Analog
04.Communication
05.Control
06.Sensors
07.Display
08.Strings
09.USB(Leonardo)
ArduinoISP
CuHead
EEPROM
Ethernet
Firmata
LiquidCrystal
SD
Servo
SoftwareSerial
SPI
Stepper
Wire
Flash
SimpleClient
SimpleServer
SimpleTweeter
SocketApp
UDPApp
WebClient
WebServer
COM51

Select the target board as "Arduino Uno":



Click Sketch-> Verify/Compile:



CHAPTER 6

CODEING:

```
#include <LiquidCrystal.h>
#include <stdio.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(8, 9);

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
// defines pins numbers
//New GPS GY-GPS6MV2

#include <Wire.h>
#include "dht.h"
#define dht_apin A5
dht DHT;

int ir1 = 2;
int ir2 = 3;
int relay = 4;

int buzzer  = 7;
const int trigPin1 = A4;
const int echoPin1 = A5;
char cntt=0;
char pd=0,pwd[5];
```

```
int tempc=0,humc=0;
unsigned char rcv,count,gchr;

 char rcvmsg[10],pastnumber[11];
 //char pastnumber1[11],pastnumber2[11];//pastnumber3[11];

 char gpsval[50];
// char dataread[100] = "";
// char lt[15],ln[15];

int i=0,k=0,lop=0;
int gps_status=0;
float latitude=0;
float logitude=0;
String Speed="";
String gpsString="";
char *test="$GPRMC";

 int ii=0,rchkr=0;
int rtr1=0;
int dist1=0,dist2=0,dist3,dist4=0,sts1=0,sts2=0;
long duration1;
int distanceCm1, distanceInch1;
long duration2;
```

```
int distanceCm2, distanceInch2;
//int hbtc=0,hbtc1=0,rtrl=0;

unsigned char gv=0,msg1[10],msg2[11];
 float lati=0,longi=0;
unsigned int lati1=0,longi1=0;
unsigned char flat[5],flong[5];
unsigned char finallat[10]="17.5616\0",finallong[10]="078.4539\0";
//17.3544,78.5935

float vout=0;

unsigned int ultra_dist1()
{int ud=0; digitalWrite(trigPin1,
  LOW); delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distanceCm1 = duration1*0.034/2;
  ud = distanceCm1;
  return ud;
}
```

```
String inputString = "";         // a string to hold incoming data
boolean stringComplete = false;  // whether the string is complete


void okcheck()
{
 unsigned char rcr;
 do{
    rcr = Serial.read();
  }while(rcr != 'K');
}
//17.6050421,78.4834139
//https://www.google.co.in/search?client=opera&q=17.6050421%2C78.4834139
void send_link()
 {
   Serial.write("AT+CMGS=\"");
   Serial.write(pastnumber);
   Serial.write("\"\r\n"); delay(2500);
   Serial.write("https://www.google.co.in/search?client=opera&q=");
   for(ii=0;ii<=6;ii++){Serial.write(finallat[ii]);}
   Serial.write("%2C");
   for(ii=0;ii<=7;ii++){Serial.write(finallong[ii]);}
   Serial.write(0x1A);delay(4000);delay(4000);
 }
```

```
 void beep()
 {

digitalWrite(buzzer,LOW);delay(1000);delay(1000);digitalWrite(buzzer,HI
GH);
 }
 void setup()
 {
 // unsigned char rcr;


 Serial.begin(115200);//serialEvent();
 mySerial.begin(9600);


 pinMode(ir1, INPUT);
 pinMode(ir2, INPUT);
 pinMode(relay, OUTPUT);


 pinMode(buzzer, OUTPUT);


 digitalWrite(buzzer, HIGH);
 digitalWrite(relay, LOW);


 lcd.begin(16, 2);lcd.cursor();
 lcd.print(" WELCOME TO ");
 lcd.setCursor(0,1);
```

```
lcd.print("    THE PROJECT");
delay(2000);


  lcd.clear();


lcd.print("Wifi init");


Serial.write("AT\r\n");
delay(500); okcheck();
Serial.write("ATE0\r\n");          okcheck();
Serial.write("AT+CWMODE=3\r\n"); delay(500);
Serial.write("AT+CIPMUX=1\r\n"); delay(500);         okcheck();
Serial.write("AT+CIPSERVER=1,23\r\n");       okcheck();
lcd.clear();
lcd.print("Waiting For");
lcd.setCursor(0, 1);
lcd.print("Connection");
do {
  rcv = Serial.read();
} while (rcv != 'C');
lcd.clear();
lcd.print("Connected");
delay(2000);
lcd.clear();
```

```
  delay(1500);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("IR1:"); //2-3-4,0
    lcd.setCursor(0,1);
    lcd.print("IR2:");  //7-8-9,0


}
int dista=0;
int distb=0;
void loop()
{
  if(digitalRead(ir1) == HIGH)
     {
       lcd.setCursor(4,0);
       lcd.print("OFF");
       digitalWrite(buzzer, HIGH);
       digitalWrite(relay, LOW);
     }

  if(digitalRead(ir1) == LOW)
     {
       lcd.setCursor(4,0);
```

```
      lcd.print("ON ");
   digitalWrite(buzzer, LOW);
   digitalWrite(relay, HIGH);
       Serial.write("AT+CIPSEND=0,16\r\n");delay(500);
    Serial.write(" Train Detected ");
    }
 if(digitalRead(ir2) == HIGH)
    {
     lcd.setCursor(4,1);
     lcd.print("OFF");
     digitalWrite(buzzer, HIGH);
     digitalWrite(relay, LOW);
    }

  if(digitalRead(ir2) == LOW)
    {
     lcd.setCursor(4,1);
     lcd.print("ON ");
   digitalWrite(buzzer, LOW);
    digitalWrite(relay, HIGH);
       Serial.write("AT+CIPSEND=0,16\r\n");delay(500);
    Serial.write(" Train Detected ");
    }
```

```
 delay(200);
}
void serialEvent()
{
  while (Serial.available())
        {

         char inChar = (char)Serial.read();
           if(inChar == '*')
             {
             // gchr = Serial.read();
             gchr = 's';
             }

        }
}

/*
void serialEvent()
{
  while (Serial.available())
        {

         char inChar = (char)Serial.read();
```

```
            if(inChar == '*')
              {
               gchr = Serial.read();
              }
            if(inChar == '#')
              {
               gchr1 = Serial.read();
              }
         }
}*/
int readSerial(char result[])
{
  int i = 0;
  while (1)
  {
    while (Serial.available() > 0)
    {
      char inChar = Serial.read();
      if (inChar == '\n')
         {
          result[i] = '\0';
          Serial.flush();
          return 0;
         }
```

```
      if (inChar != '\r')
         {
          result[i] = inChar;
          i++;
         }
    }
  }
}



int readSerial1(char result[])
{
  int i = 0;
  while (1)
  {
    while (Serial.available() > 0)
    {
      char inChar = Serial.read();
      if (inChar == '*')
         {
          result[i] = '\0';
          Serial.flush();
          return 0;
         }
```

```
        if (inChar != '*')
           {
            result[i] = inChar;
            i++;
           }
      }
    }
  }
  void gpsEvent()
  {
    gpsString="";
    while(1)
    {
     //while (gps.available()>0)           //Serial incoming data from GPS


      while (mySerial.available() > 0)
     {
      //char inChar = (char)gps.read();
      char inChar = (char)mySerial.read();
       gpsString+= inChar;                 //store incoming data from GPS to
temparary string str[]
       i++;
      // Serial.print(inChar);
       if (i < 7)
```

```
        {
         if(gpsString[i-1] != test[i-1])         //check for right string
         {
           i=0;
           gpsString="";
         }
        }
       if(inChar=='\r')
       {
        if(i>60)
        {
          gps_status=1;
          break;
        }
        else
        {
          i=0;
        }
       }
     }
     if(gps_status)
      break;
     }
   }
```

```
void get_gps()
{

 lcd.clear();
 lcd.print("Getting GPS Data");
 lcd.setCursor(0,1);
 lcd.print("Please Wait.....");

  gps_status=0;
  int x=0;
  while(gps_status==0)
  {
   gpsEvent();
   int str_lenth=i;
   coordinate2dec();
   i=0;x=0;
   str_lenth=0;
  }
}

void coordinate2dec()
{
 String lat_degree="";
```

```
  for(i=17;i<=18;i++)
    lat_degree+=gpsString[i];


String lat_minut="";
   for(i=18;i<=19;i++)
   lat_minut+=gpsString[i];
   for(i=21;i<=22;i++)
   lat_minut+=gpsString[i];
String log_degree="";
for(i=29;i<=31;i++)
log_degree+=gpsString[i];
String log_minut="";
for(i=32;i<=33;i++)
    log_minut+=gpsString[i];
  for(i=35;i<=36;i++)
  log_minut+=gpsString[i];



  Speed="";
  for(i=42;i<45;i++)          //extract longitude from string
    Speed+=gpsString[i];


   float minut= lat_minut.toFloat();
   minut=minut/60;
```

```
    float degree=lat_degree.toFloat();
    latitude=degree+minut;


    minut= log_minut.toFloat();
    minut=minut/60;
    degree=log_degree.toFloat();
    logitude=degree+minut;
}


/*
void coordinate2dec()
{
  String lat_degree="";
    for(i=19;i<=20;i++)
    lat_degree+=gpsString[i];


  String lat_minut="";
     for(i=21;i<=22;i++)
     lat_minut+=gpsString[i];
     for(i=24;i<=25;i++)
     lat_minut+=gpsString[i];
  String log_degree="";
    for(i=32;i<=34;i++)
    log_degree+=gpsString[i];
```

```
  String log_minut="";
    for(i=35;i<=36;i++)
    log_minut+=gpsString[i];
    for(i=38;i<=39;i++)
    log_minut+=gpsString[i];
      Speed="";
    for(i=45;i<48;i++)          //extract longitude from string
     Speed+=gpsString[i];


    float minut= lat_minut.toFloat();
    minut=minut/60;
    float degree=lat_degree.toFloat();
    latitude=degree+minut;


    minut= log_minut.toFloat();
    minut=minut/60;
    degree=log_degree.toFloat();
    logitude=degree+minut;
}
*/
void gps_convert()
{
 if(gps_status)
  {
```

```
    Serial.println(gpsString);


    if(gpsString[0] == '$' && gpsString[1] == 'G' && gpsString[2] == 'P'
&& gpsString[3] == 'R' && gpsString[4] == 'M' && gpsString[5] == 'C')
      {
       //
Serial.println("Don1111111111111111111111111111111111111111111111
11111111\r\n");
      //
Serial.write(gpsString[18]);Serial.write(gpsString[19]);Serial.write(gpsStri
ng[20]);Serial.write(gpsString[21]);Serial.write(gpsString[22]);
       //lcd.setCursor(0,0);
       for(ii=0;ii<9;ii++)
         {
          //lcd.write(gpsString[19+ii]);
          msg1[ii] = gpsString[19+ii];
          //Serial.write(msg1[ii]);
         }
         //Serial.println("\r\n");
       //lcd.setCursor(0,1);
        for(ii=0;ii<10;ii++)
         {
          //lcd.write(gpsString[32+ii]);
          msg2[ii] = gpsString[32+ii];
         // Serial.write(msg2[ii]);
         }
```

```
 // Serial.println(msg1);

 // Serial.println(msg2);


      //lati = (((msg1[2]-48)*100000) +((msg1[3]-48)*10000) + ((msg1[5]-
48)*1000) + ((msg1[6]-48)*100) + ((msg1[7]-48)*10) + (msg1[8]-48));

      //longi = (((msg2[3]-48)*100000) + ((msg2[4]-48)*10000) +
((msg2[6]-48)*1000) + ((msg2[7]-48)*100) + ((msg2[8]-48)*10) +
(msg2[9]-48));


      lati = (((msg1[2]-48)*1000) + ((msg1[3]-48)*100) + ((msg1[5]-
48)*10) + (msg1[6]-48));

      longi = (((msg2[3]-48)*1000) + ((msg2[4]-48)*100) + ((msg2[6]-
48)*10) + (msg2[7]-48));


     //  converts(lati);Serial.write("-");

     // converts(longi);Serial.write("\r\n");


      lati = (lati/60);  longi = (longi/60);


      lati = (lati*100); longi = (longi*100);

      lati1 = lati;      longi1 = longi;


 // Serial.write("After ");

  //      converts(lati1);Serial.write("-");

    //   converts(longi1);Serial.write("\r\n");

          convlat(lati); convlong(longi);
```

```
        finallat[0] = msg1[0];
        finallat[1] = msg1[1];
        finallat[2] = '.';
        finallat[3] = flat[0]; finallat[4] = flat[1];finallat[5] = flat[2];finallat[6]
= flat[3];finallat[7] = '\0';



        finallong[0] = msg2[0];
        finallong[1] = msg2[1];
        finallong[2] = msg2[2];
        finallong[3] = '.';
        finallong[4] = flong[0];finallong[5] = flong[1];finallong[6] =
flong[2];finallong[7] = flong[3];finallong[8] = '\0';


    }
   }
  }


  void convlat(unsigned int value)
      {
           unsigned int a,b,c,d,e,f,g,h;


      a=value/10000;
      b=value%10000;
      c=b/1000;
```

```
    d=b%1000;
    e=d/100;
    f=d%100;
    g=f/10;
    h=f%10;
    a=a|0x30;
   c=c|0x30;
   e=e|0x30;
   g=g|0x30;
   h=h|0x30;

  // dlcd(a);
// dlcd(c);dlcd(e); dlcd(g);dlcd(h);//lcddata('A');//lcddata(' ');lcddata(' ');
          flat[0] = c;
          flat[1] = e;
         flat[2]  =  g;
         flat[3] = h;
       }

 void convlong(unsigned int value)
    {
        unsigned int a,b,c,d,e,f,g,h;

    a=value/10000;
```

```
      b=value%10000;
      c=b/1000;
      d=b%1000;
      e=d/100;
      f=d%100;
      g=f/10;
      h=f%10;
      a=a|0x30;
    c=c|0x30;
    e=e|0x30;
    g=g|0x30;
    h=h|0x30;

  // dlcd(a);
// dlcd(c);dlcd(e); dlcd(g);dlcd(h);//lcddata('A');//lcddata(' ');lcddata(' ');
             flong[0] = c;
             flong[1] = e;
           flong[2]  =  g;
           flong[3] = h;
         }
/*
void coordinate2dec()
{
  String lat_degree="";
```

```
  for(i=20;i<=21;i++)
    lat_degree+=gpsString[i];

String lat_minut="";
   for(i=22;i<=28;i++)
   lat_minut+=gpsString[i];
String log_degree="";
for(i=32;i<=34;i++)
log_degree+=gpsString[i];
String log_minut="";
for(i=35;i<=41;i++)
    log_minut+=gpsString[i];

  Speed="";
  for(i=45;i<48;i++)          //extract longitude from string
    Speed+=gpsString[i];

   float minut= lat_minut.toFloat();
   minut=minut/60;
   float degree=lat_degree.toFloat();
   latitude=degree+minut;

   minut= log_minut.toFloat();
   minut=minut/60;
```

```
    degree=log_degree.toFloat();
    logitude=degree+minut;
}*/

void gsminit()
{
 Serial.write("AT\r\n");                okcheck();
 Serial.write("ATE0\r\n");              okcheck();
Serial.write("AT+CMGF=1\r\n");          okcheck();
Serial.write("AT+CNMI=1,2,0,0\r\n");    okcheck();
Serial.write("AT+CSMP=17,167,0,0\r\n"); okcheck();

 lcd.clear();
 lcd.print("SEND MSG STORE");
 lcd.setCursor(0,1);
 lcd.print("MOBILE NUMBER");
 do{
   rcv = Serial.read();
  }while(rcv != '*');
   readSerial(pastnumber);pastnumber[10] = '\0';

   /*
  pastnumber1[0] = pastnumber[0];pastnumber1[1] =
pastnumber[1];pastnumber1[2] = pastnumber[2];pastnumber1[3] =
```

```
pastnumber[3];pastnumber1[4] = pastnumber[4];pastnumber1[5] =
pastnumber[5];

   pastnumber1[6] = pastnumber[6];pastnumber1[7] =
pastnumber[7];pastnumber1[8] = pastnumber[8];pastnumber1[9] =
pastnumber[9];pastnumber1[10] = '\0';

    */

 /*

   pastnumber3[0] = pastnumber[20];pastnumber3[1] =
pastnumber[21];pastnumber3[2] = pastnumber[22];pastnumber3[3] =
pastnumber[23];pastnumber3[4] = pastnumber[24];pastnumber3[5] =
pastnumber[25];

   pastnumber3[6] = pastnumber[26];pastnumber3[7] =
pastnumber[27];pastnumber3[8] = pastnumber[28];pastnumber3[9] =
pastnumber[29];pastnumber3[10] = '\0';

 */

   lcd.clear();

   lcd.print(pastnumber);


    Serial.write("AT+CMGS=\"");

    Serial.write(pastnumber);

    Serial.write("\"\r\n"); delay(3000);

    Serial.write("Reg\r\n");

    Serial.write(0x1A);

    delay(4000);  delay(4000);


 }
```

```
void converts(unsigned int value)
{
  unsigned int a,b,c,d,e,f,g,h;

     a=value/10000;
     b=value%10000;
     c=b/1000;
     d=b%1000;
     e=d/100;
     f=d%100;
     g=f/10;
     h=f%10;

     a=a|0x30;
     c=c|0x30;
     e=e|0x30;
     g=g|0x30;
     h=h|0x30;

    Serial.write(a);
    Serial.write(c);
    Serial.write(e);
    Serial.write(g);
```

```
   Serial.write(h);
}

void convertl(unsigned int value)
{
  unsigned int a,b,c,d,e,f,g,h;

      a=value/10000;
      b=value%10000;
      c=b/1000;
      d=b%1000;
      e=d/100;
      f=d%100;
      g=f/10;
      h=f%10;



      a=a|0x30;
      c=c|0x30;
      e=e|0x30;
      g=g|0x30;
      h=h|0x30;
```

```
    //lcd.write(a);
    //lcd.write(c);
    lcd.write(e);
    lcd.write(g);
    lcd.write(h);
}
void convertk(unsigned int value)
{
  unsigned int a,b,c,d,e,f,g,h;

     a=value/10000;
     b=value%10000;
     c=b/1000;
     d=b%1000;
     e=d/100;
     f=d%100;
     g=f/10;
     h=f%10;

     a=a|0x30;
     c=c|0x30;
     e=e|0x30;
     g=g|0x30;
     h=h|0x30;
```

```
    //lcd.write(a);
    //lcd.write(c);
   // lcd.write(e);
   // lcd.write(g);
    lcd.write(h);
}
/*
   sensorValue = analogRead(analogInPin);
   sensorValue = (sensorValue/9.31);
   lcd.setCursor(1,1); //rc
   lcd.print(sensorValue);
   Serial.print(sensorValue);
 */
```

CHAPTER-7

PROPOSED METHOD

The significance of defining the research problem is to address the gaps in the literature. The purpose is to contribute to the existing work to enhance the quality of the overall framework so that it can benefit the end society in future. This can be achieved by adding more functionalities and features that can improve the working of the end system. Dhana Lakshmi and Leni [21] designed a system that monitors the condition of the car during its journey. The parameters that are addressed in their work are, gas leakage which is monitored by using an MQ2 gas sensor, vehicle speed which is recorded by hall-effect sensors, GPS and WIFI modules for communication and tracking location of vehicles.

However, for an accident detection case, only speed has been considered by making use of hall -effect sensors. Moreover, Pin and Wang proposed a vehicle collision detection algorithm which works well for T-intersection road design. The parameters that are considered for the design of the algorithm are, curvature area of T-intersection junctions and the predicted time for the two cars to meet at the junction. We feel that the algorithm is effective for the specific case of T-intersection and not for general road accidents. Therefore, there is a need for modifying the existing work done by authors to support the general road accidents.

In our approach, we are addressing the gaps by adding an accelerometer, vibration sensor and most importantly heartrate sensor. These components contribute to the hardware setup of the system. Also, we would like to introduce an algorithm for general road accidents that is appropriate for this hardware setup. We have considered a few parameters which are helpful for accident detection and notification. These parameters are vehicle acceleration, retardation, crash impact, the value of heart rate sensor (embedded within the belt) and information of accident location which is tracked by GPS. It is then sent to emergency services/family members by WIFI communication.

EXPLANATION OF RESULT

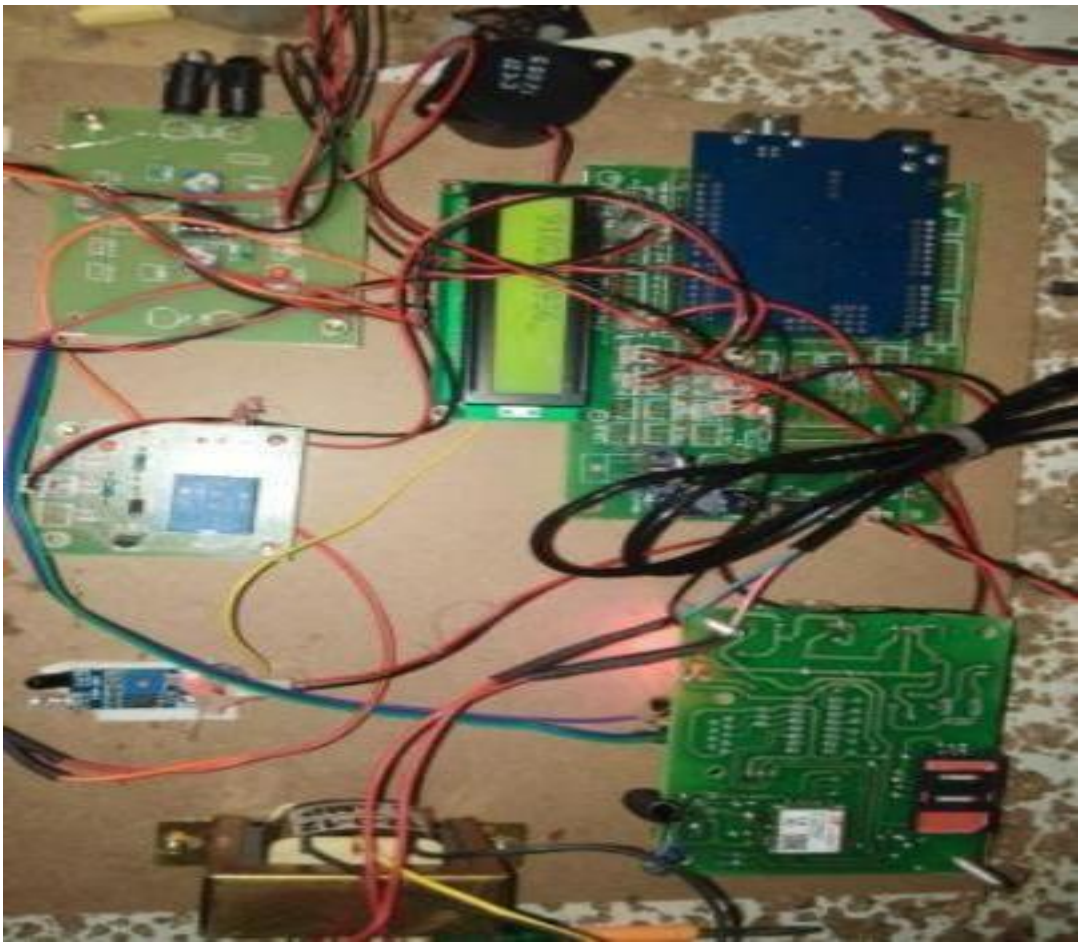
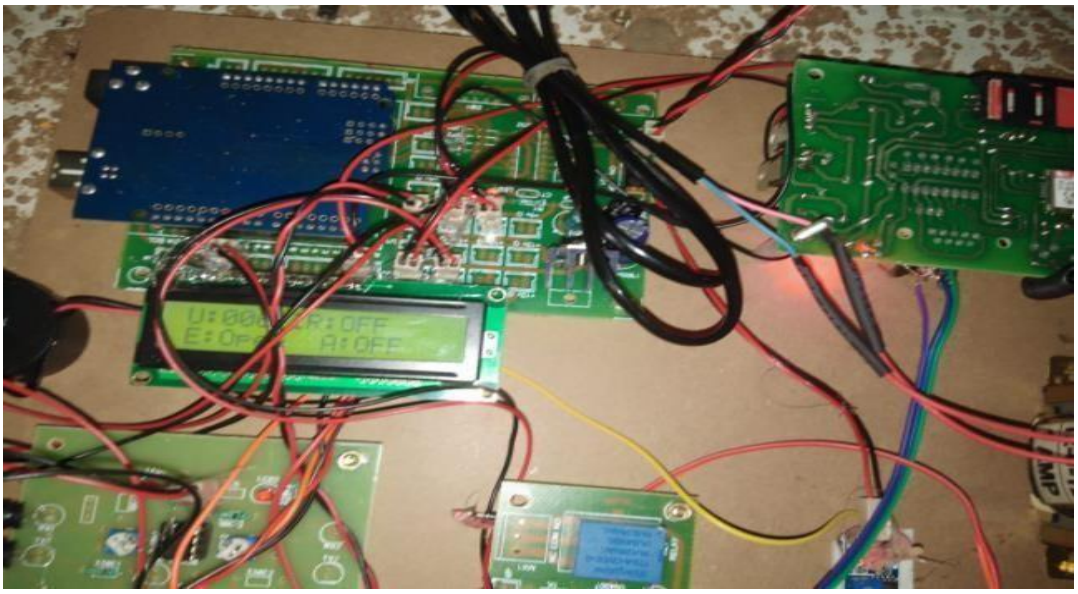


Fig.7.1. Hardware kit image.



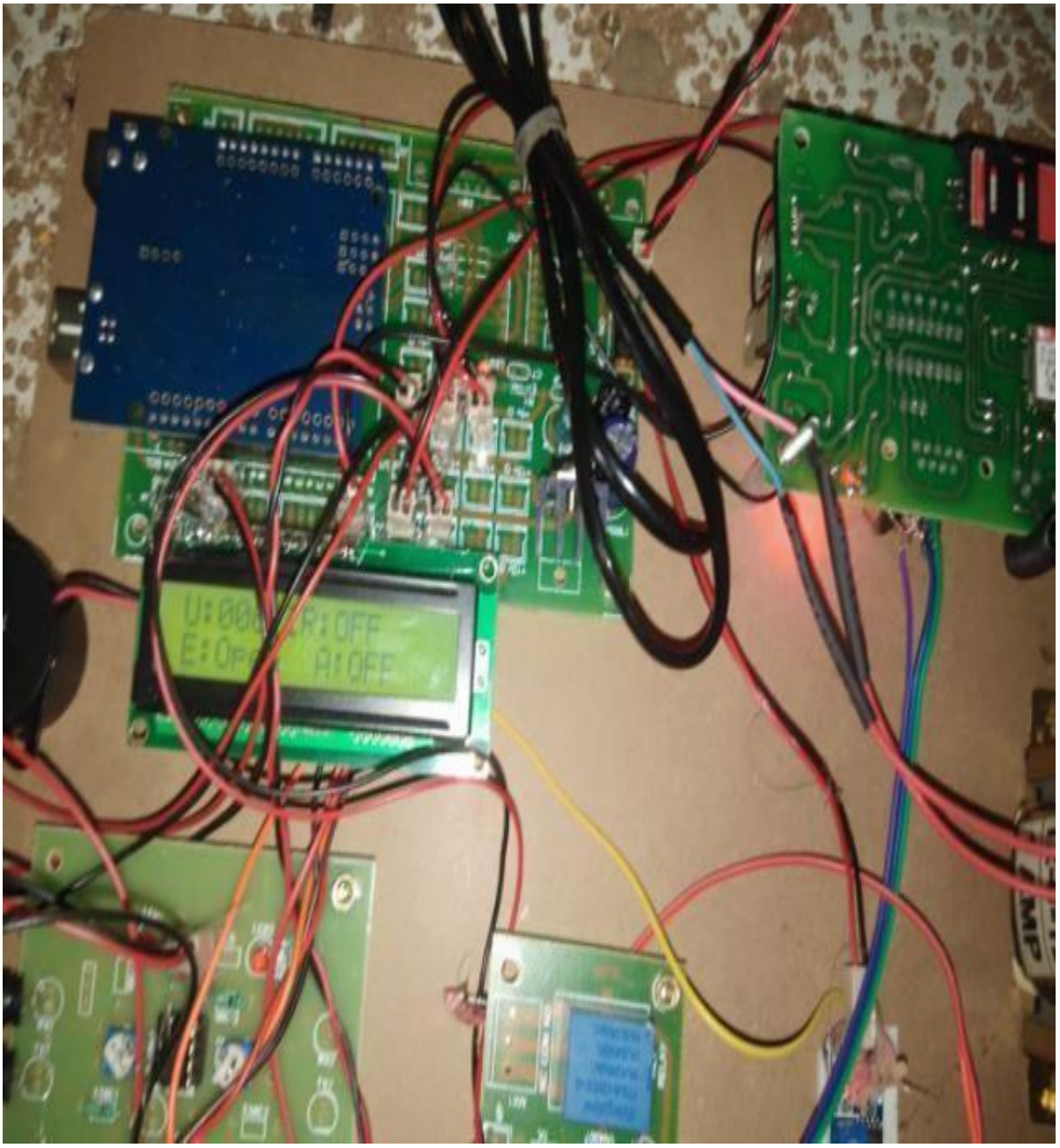


Fig.7.2. Accident detected time indicated in LCD.

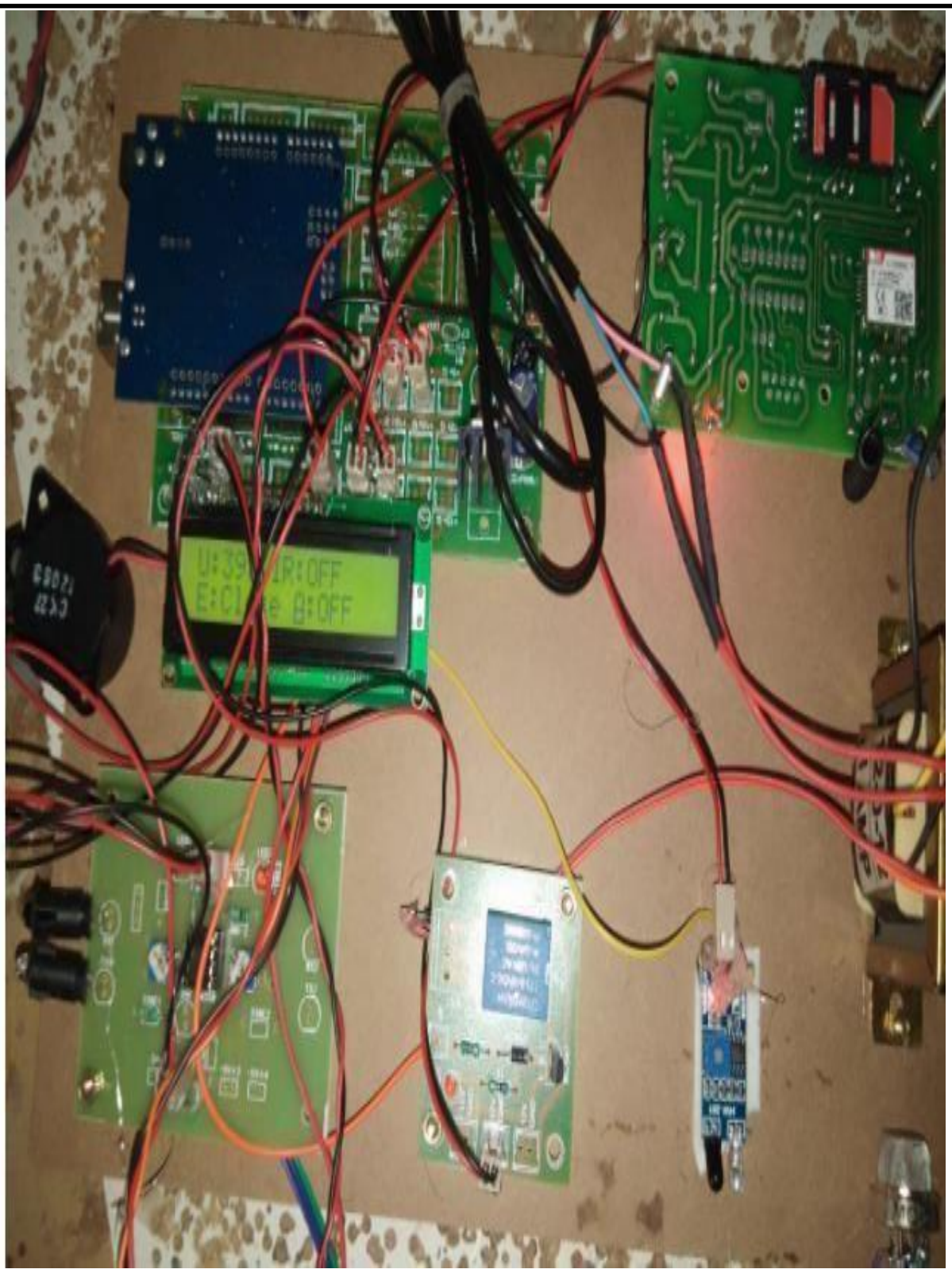


Fig.7.3. Eyes closed indication.

CHAPTER-8

ADVANTAGES AND LIMITATIONS

ADVANTAGES:

1) Highly Sensitive, Low cost and Reliable circuit.
2) Easily identify the accident occurs time.
3) Long distance also possible for data communication
4) Can handle heavy loads up to 7A.
5) System can be switched into manual mode when even required.
6) Train Collison avoidance system helps prevent head- on, Rear-end, and Side- on collisions
7) To improve safety of trackside maintenance workers and vehicles.
8) Train Collision Prevention.
9) Level crossing safety.
10) Automatic Train Control (ATC).
11) Applications aimed at enhancing railway safety, improving operational efficiency, and minimizing the risk of accidents and disruptions in railway operations

LIMITATIONS:

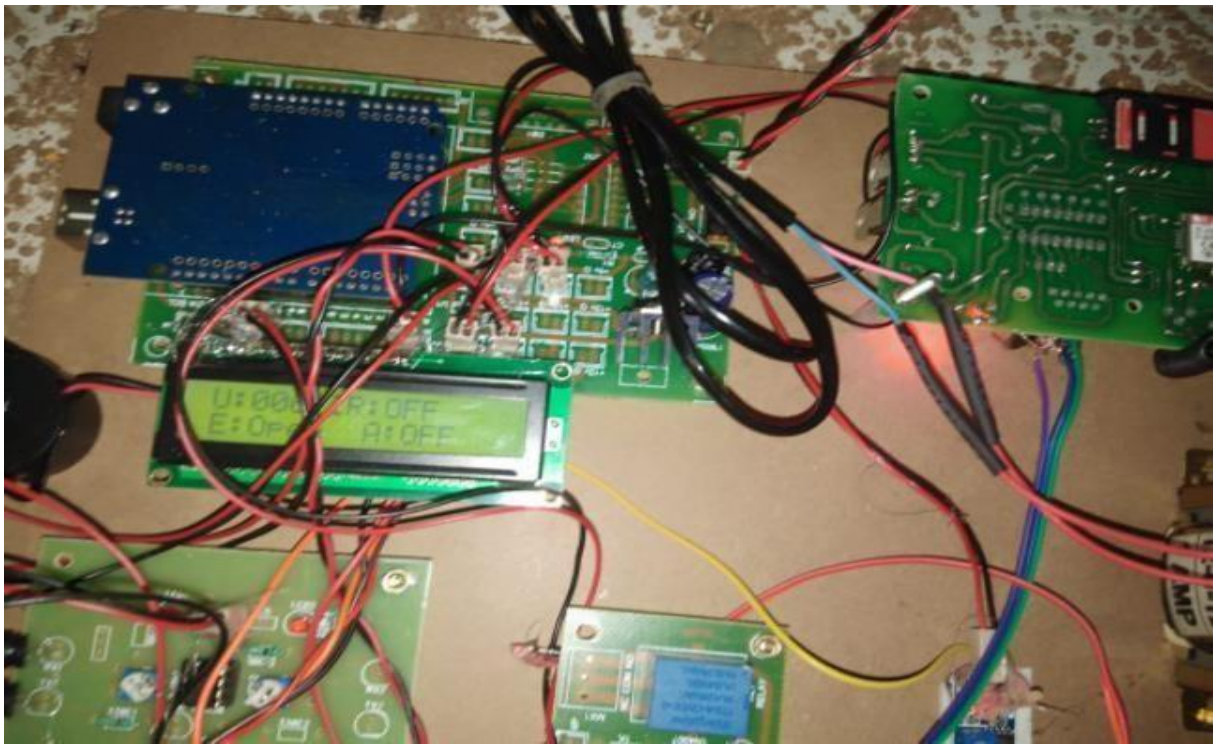
1) This is applicable for only large farms.
2) Have limited life after installation.

CHAPTER-9

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

The main idea of this paper is to notify the concerned authorities about an accident only if the passengers are injured. The proposed framework is intended to solve the same by incorporating more features in the already existing work done by the authors. With the addition of above discussed functionalities, this system can resolve most of the accident scenarios by detecting accidents on time and triggering immediate help from emergency services without wasting any time. Moreover, the driver's health is being tracked by heart rate sensor (embedded in seatbelt) which serves as the added advantage. If implemented with proper planning and resources, this framework could serve to be a great help to the society. Hence, there is need of such systems that could save the lives involved with accidents.



CHAPTER-10

BIBLIOGRAPHY

REFERENCE:

1. Andrea Z and Lorenzo V., "Internet of Things for Smart Cities," IEEE Internet of Things Journal, vol/issue: 1(1), Feb 2014.
2. Isan K. and S. D. Sawant, "Integration of Cloud Computing and Internet of Things," International Journal of Advanced Research in Computer and Communication Engineering, vol/issue: 5(4), Apr 2016.
3. Sonali D. T., "Cloud Computing and Software-Based Internet of Things," International Journal of Advanced Research in Computer Science and Software Engineering, vol/issue: 6(4), Apr 2014.
4. Jonathan K., "Using Active Queue Management to Assist IOT Application Flows in Home Broad band Networks," 2017 IEEE Internet of Things Journal, vol/issue: 4(5), Oct 2017.
5. Penafiel Z., et al., "Secure Location of Things (SLOT): Mitigating Local Spoofing Attacks in Internet of Things," IEEE Internet of Things Journal, vol. 4, Dec 2017.
6. Akriti S., et al., "Intelligent Accident Management System using IoT and Cloud Computing," 2nd International Conference on Next Generation Computing Technologies, Oct 2016.
7. C. Chatrapathi and N. R. Venkat Sasikumar, "VANET based Integrated Framework for Smart Accident Management System," International Conference on Soft-Computing and Network Security, Feb 2015.
8. Priyal R. and Vanthana S., "Car Accident Notification System based on Internet of Things," International Journal of Computer Applications, vol/issue: 107(17), Dec 2014.
9. H. M. Ali and Z. S. Alwan, "Car Accident Detection and Notification System Using Smartphone," International Journal of Computer Science and Mobile Computing, vol/issue: 4(4), pp. 620-635, Apr 2015.

10. Aishwarya S. R., et al., "An IoT Based Accident Prevention and Tracking System for Night Drivers," International Journal of Innovative Research in Computer and Communication Engineering, vol/issue: 3(4), Apr 2015