A MAJOR PROJECT REPORT ON

IOT BASED THEFT CONTROL AUTO ALERTING SYSTEM FOR AUTOMOBILE

Submitted

In partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted By

SAGRI ANVITH	(16D41A05H5)
RATHNA NAGARAJU	(16D41A05H1)
PALLAPU JAI SHRAVAN KUMAR	(16D41A05F1)

Under the guidance of

Mr. NATARAJA SHEKHAR
(ASST PROFESSOR)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution under UGC, Accredited by NBA, Affiliated to JNTUH)
Sheriguda, Ibrahimpatnam (2019-2020)

SRI INDU COLLEGE OF ENGINEERING AND

TECHNOLOGY

(An Autonomous Institution under UGC, Accredited by NBA, Affiliated to JNTUH)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Major Project entitled "**IOT BASED THEFT CONTROL AUTO ALERTING FOR AUTOMOBILES**"is a bonafide work carried out by **SAGRI ANVITH(16D41A05H5) ,RATHNA NAGARAJU(16D41A05H1), PALLAPU JAI SHRAVAN KUMAR(16D 41A05F1)**in partial fulfillment for the award of **Bachelor of Technology** in **Computer Science and Engineering** of SICET, Hyderabad for the academic year **2019-2020**. The Project has been approved as it satisfies academic requirements in respect of the work pres- -cribed for **IV Year, II Semester** of **B.Tech** course.

Internal Guide **HOD**

(Dept of CSE)

External Examiner

ACKNOWLEDGMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project work a success. We thank the almighty for giving us the courage & perseverance in completing the project.

We are thankful to principal **Prof.Dr.P.MALLESHAM**, for giving us the permission to carry out this project and for providing necessary infrastructure and labs.

We are highly indebted to, **Dr.CH.G.V.N.PRASAD,** Head of the Department of Computer Science & Engineering, Project Guide, for providing valuable guidance at every stage of this project.

We are grateful to our internal project guide **Mr.NATARAJA SHEKHAR** for his constant motivation and guidance given by him during the execution of this project work.

We would like to thank the Teaching & Non-Teaching staff of Department of Computer Science & Engineering for sharing their knowledge with us last but not the least we express our sincere thanks to everyone who helped directly or indirectly for the completion of this project.

SAGRI ANVITH (16D41A05H5)

RATHNA NAGARAJU (16D41A05H1)

PALLAPU JAI SHRAVAN KUMAR (16D41A05F1)

TABLE OF CONTENTS

1.ABSTRACT

As the amount of urban vehicle grows rapidly, vehicle theft has become a shared concern for all citizens. Security and safety have always become a necessity for urban population. However, present anti-theft systems lack the tracking and monitoring function. Internet of things(IOT) has been governing the electronics era with cloud services dominating the ever-increasing electronics product segment. Thus, there is a need to develop a system for providing security to the vehicle from problems like theft and towing using IOT for security of automobiles and passengers. Our system proposes a novel security system based on wireless communication and a lowcost Bluetooth module. This paper illustrates a model in which the GSM is used for sending messages. the user can control the engine/ignition and turn it off if needed. The system also employs a password through keypad (with maximum 3 chances) which controls the opening of a safety locker door as well as wearing of a seat belt. If there is a window intruder, the IR module/sensor detects the intruder, or any obstacle and it sends a signal to the micro controller. The controller is connected to a Bluetooth module and to an alarm system. The System transmits an alert signal to the dashboard (which is nothing but a mobile handset) which sends an alert signal to the user's mobile phone. The prototype also provides a solution to the problem like Towing. Thus, the system uses Bluetooth module and controller to control the security system from the user's mobile phone by means of any device with a potential Internet connection..

2. INTRODUCTION

In recent years vehicle theft has become a major issue which should be traced and detected. The safety and security of the vehicle is essential. Even there are many existing mechanisms they have some limitations and high cost. So, an efficient security mechanism is needed. This project detects vehicle theft. Arduino is the main component which is used to interface dc motor and GPS, GSM. The place of the vehicle identified using Global positioning system (GPS) and Global system mobile communication (GSM) with the help of Wi-Fi module ESP 8266. GPS is a space-based navigation system used to track the vehicle and it gives the location of the robbed device in all weather conditions. It gives the latitude and longitude of the device using GPS antenna. GSM is a specialized type of modem which accepts a SIM card, and operates just like a mobile phone. It is used to provide information to the owner and alert him with a message having latitude and longitude of the vehicle. This total system is operated with a switch which is made on when we park the vehicles out. Now if the vehicle theft happened, dc motor starts and the above procedure continues and the information is posted using internet of things. This vehicle theft prevention and tracking system is used in client's vehicle as a theft prevention and rescue device

2.1 EXISTING SYSTEM:

The current conventional vehicles does not come with any security other than lock only high end cars having maps support.

2.2 PROPOSED SYSTEM:

The Main Aim of this Project presents an IOT based vehicle theft detection system. In this mechanism as soon as the motor starts with duplicate keys i.e., vehicle theft occurs, Arduino activates GPS, GSM and sends an alert message to the owner and the longitude and latitude readings of vehicle are posted using internet of things with the help of GSM module and Also gives Buzzer Sound At Vehicle As well. The entire mechanism can be operated with a switch for user convenience.

Advantages:

- IOT based user -friendly interfacing.
- Low power consumption can work on with vehicle Battery.
- Long Life.
- Wireless Transmission.
- Fast Response to the Owner.
- System can be Turned off whenever not Required.
- Easy to Fetch Vehicle Location without any assistance of Police
- Less Space for Device

Disadvantages:

- If user did not turn-off IOT device he/she will receives the notification
- If Theft finds out IOT device there is a Higher chance of Disconnecting Device.

2. REQUIREMENTS:

3.1 HARDWARE REQUIREMENTS:

1. PIR SENSOR
2. LCD
3. BUZZER
4. GSM MODULE
5. GPS 6. RPS

3.2 SOFTWARE REQUIREMENTS:

1. Express PCB – for designing circuit
2. Arduino IDE compiler - for compilation part
3. Proteus 8 (Embedded C) – for simulation part

APPLICATIONS:

It can be used in all types of vehicles with further changes can be used for homes to detect Intruder.

With further Enhancements of Hardware it is capable of turning off the engine of a Vehicle .

4. MODULES

4.1 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.

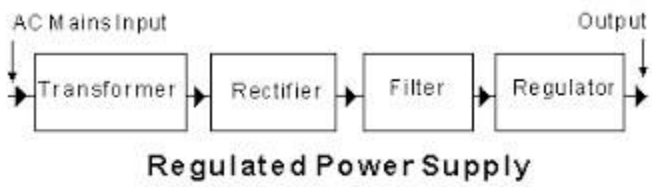
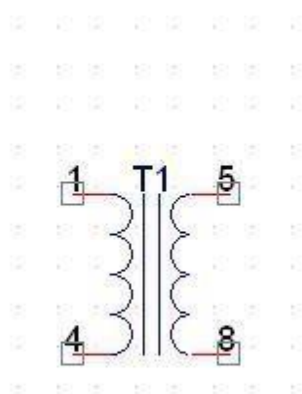


Fig: shows the basic block diagram of a fixed regulated power supply. Let us go through each block.

4.1.1 Transformer:



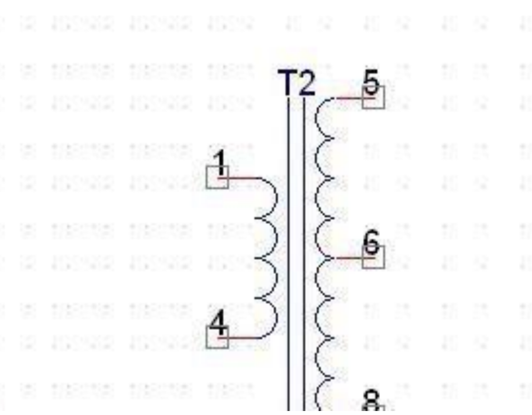


Fig 3.3: Transformer

Fig 3.4:Center tapped 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.

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

4.1.2 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:

A) Half Wave rectifier.

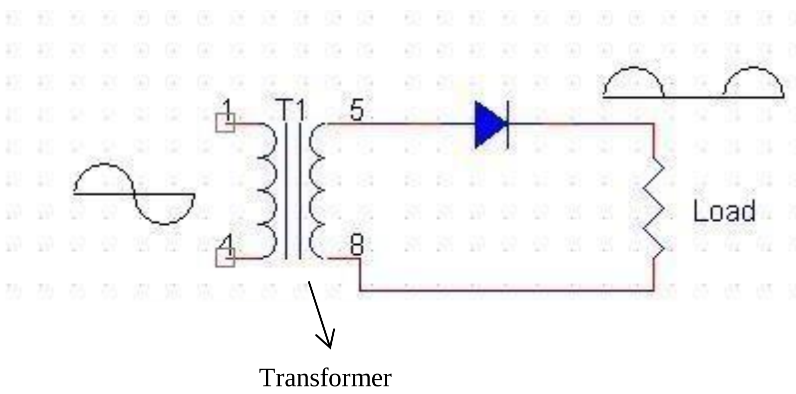


Fig 3.5: Half wave rectifier

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.

B) Full wave rectifier:

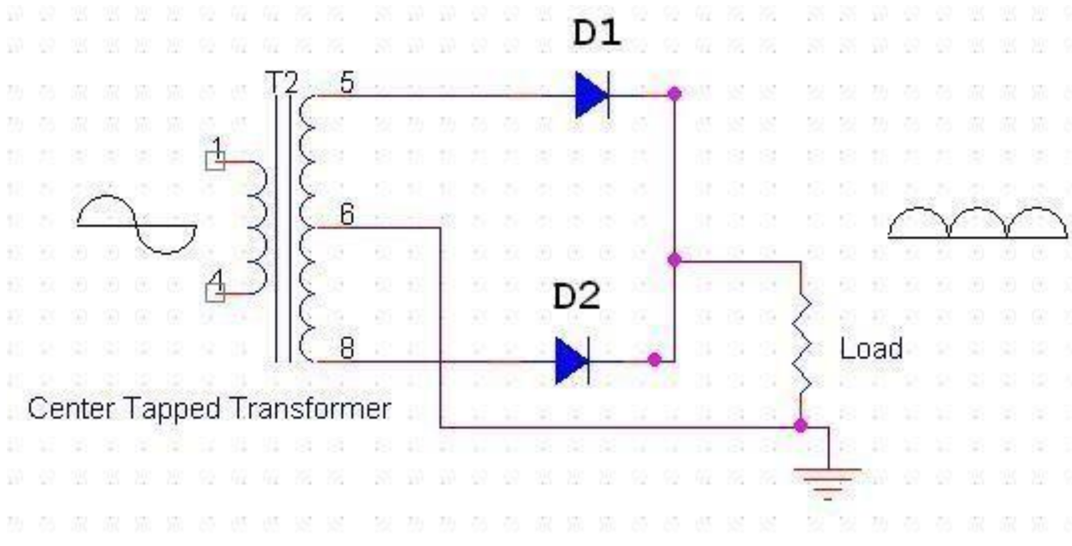


Fig Full wave Rectifier

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.

C) BridgeRectifier:

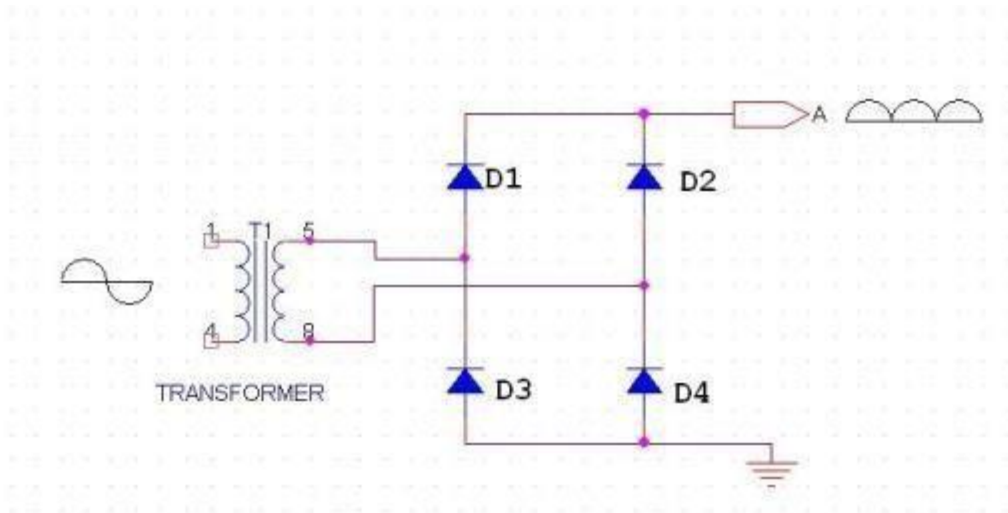


Fig 3.7: Bridge Rectifier with transformer

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.

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.

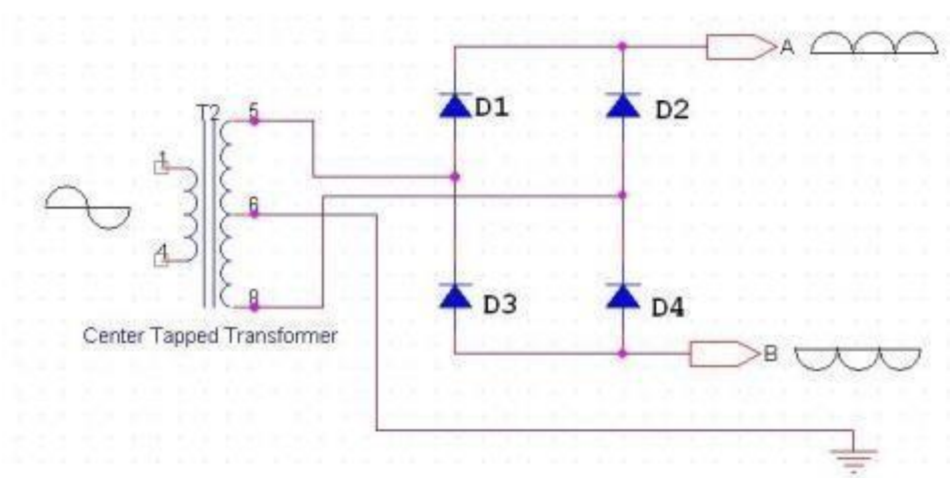


Fig : Bridge Rectifier with center tapped transformer

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.

4.1.3 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.

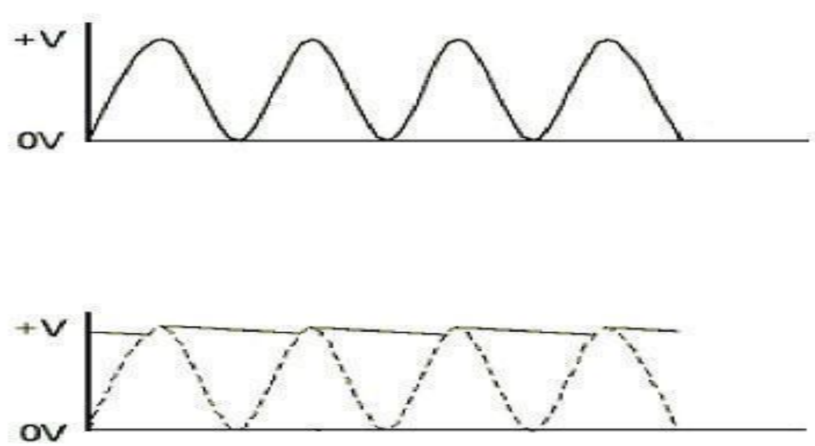


Fig : Output of Rectifier due to filter capacitor

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 ca0pacitor 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.

4.1.4 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.

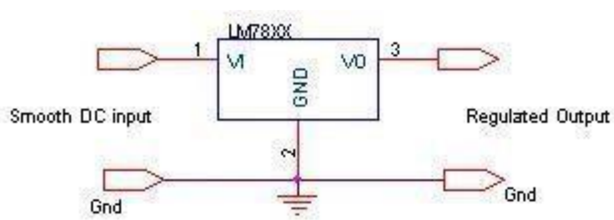


Fig 3.10: Voltage regulator

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:

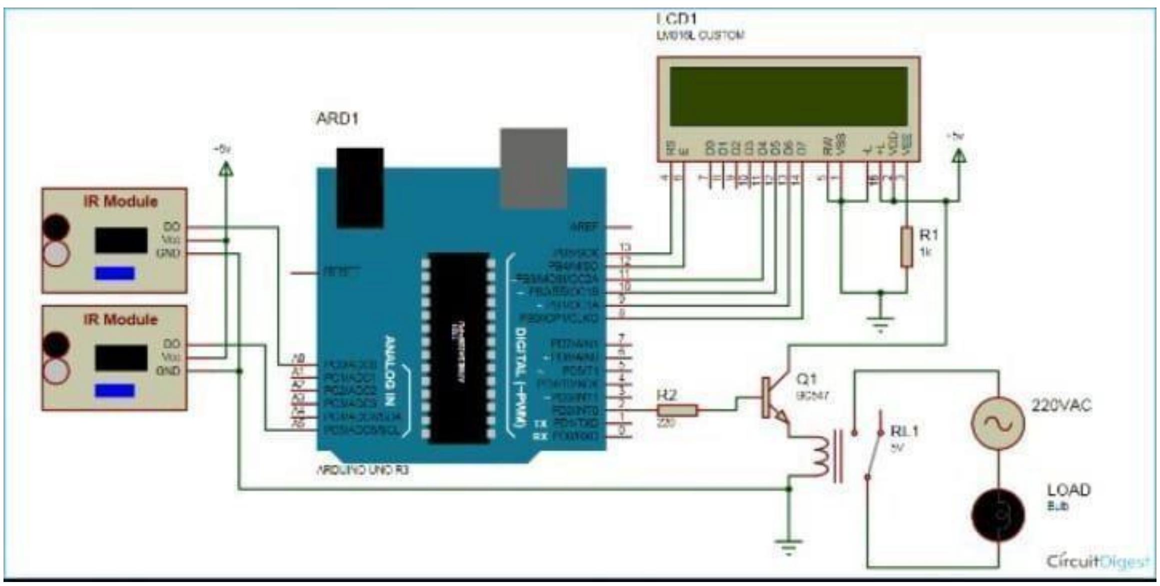


Fig 3.11: Circuit Diagram

4.1.5 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.

Table 3.1. Specifications of IC7805

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

4.2.4. PIR SENSOR:

PIR sensor:

The PIR Sensor detects motion up to 20 feet away by using a Fresnel lens and infraredsensitive element to detect changing patterns of passive infrared emitted by objects in its vicinity. Inexpensive and easy to use, it's ideal for alarm systems, motion-activated lighting, and holiday props. The PIR Sensor is compatible with all Parallax microcontrollers. BASIC Stamp and SX/B code is available under Downloads below, and Spin code is posted to the Propeller Object Exchange.

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses.

They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors. PIRs are basically made of a pyroelectric sensor (which you can see above as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

Working of PIR sensor:

PIR sensors are more complicated than many of the other sensors explained in these tutorials (like photocells, FSRs and tilt switches) because there are multiple variables that affect the sensors input and output. To begin explaining how a basic sensor works, we'll use the rather nice diagram below (if anyone knows where it originates plz let me know).

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. The lens used here is not really doing much and so we see that the two slots can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors.

When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.

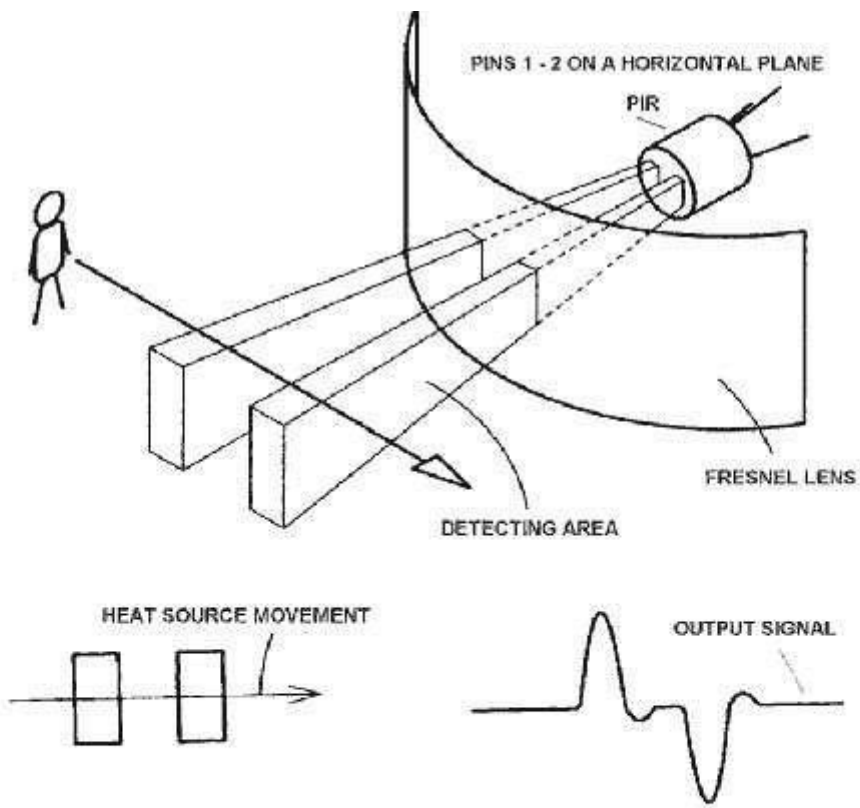
Fig: PIR sensor

Theory of Operation:

Pyroelectric devices, such as the PIR sensor, have elements made of a crystalline material that generates an electric charge when exposed to infrared radiation. The changes in the amount of infrared striking the element change the voltages generated, which are measured by an on-board amplifier. The device contains a special filter called a Fresnel lens, which focuses the infrared signals onto the element. As the ambient infrared signals change rapidly, the on-board amplifier trips the output to indicate motion.

The PIR (Passive Infra-Red) Sensor is a pyroelectric device that detects motion by measuring changes in the infrared (heat) levels emitted by surrounding objects. This motion can be detected by checking for a sudden change in the surrounding IR patterns. When motion is detected the PIR sensor outputs a high signal on its output pin. This logic signal can be read by a microcontroller or used to drive a transistor to switch a higher current load.

The IR sensor itself is housed in a hermetically sealed metal can to improve noise/temperature/humidity immunity. There is a window made of IR-transmissive material (typically coated silicon since that is very easy to come by) that protects the sensing element behind the window are the two balanced sensors.



Lenses:

PIR sensors are rather generic and for the most part vary only in price and sensitivity. Most of the real magic happens with the optics. This is a pretty good idea for manufacturing: the PIR sensor and circuitry is fixed and costs a few dollars. The lens costs only a few cents and can change the breadth, range, sensing pattern, very easily.

In the diagram above, the lens is just a piece of plastic, but that means that the detection area is just two rectangles. Usually we'd like to have a detection area that is much larger. To do that, we use a simple lens such as those found in a camera: they condenses a large area (such as a landscape) into a small one (on film or a CCD sensor). For reasons that will be apparent soon, we would like to make the PIR lenses small and thin and moldable from cheap plastic, even though it may add disto

WIFI MODULE:

The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability, produced by Espressif 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 WiFi 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 Tensilica 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.

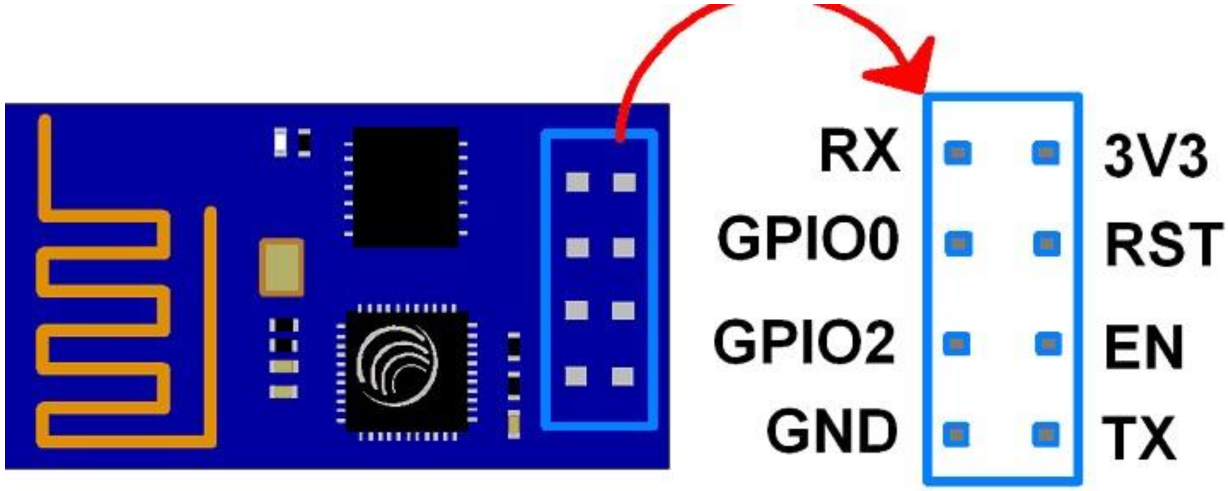
ESP8266 module is low cost standalone wireless transceiver that can be used for end-point IoT developments.

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.



ESP8266-01 Module Pins

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 (I²C), analog-to-digital conversion (10-bit ADC), Serial Peripheral Interface (SPI), I²S 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.) External flash memory can be accessed through SPI. The silicon chip itself is housed within a 5 mm × 5 mm Quad Flat No-Leads package with 33 connection pads — 8 pads along each side and one large thermal/ground pad in the center.

- ESP8266EX Overview
- 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"; while other modules might be ill-labeled and merely referred to by a general description — e.g., "ESP8266 Wireless Transceiver." ESP8266-based modules have demonstrated themselves as a capable, low-cost, networkable foundation for facilitating end-point IoT developments. Espressif's official modules are presently ESP-WROOM-02 and ESP-WROOM-S2. The Ai-Thinker modules are succinctly labeled

ESP01 through ESP-14. (Note: many people refer to the Ai-Thinker modules with the unofficial monikers of "ESP8266-01" through "ESP8266-14" for clarity.) See the ESP8266 article on Wikipedia for more information about popular ESP8266 modules.

4.3 Arduino Controller

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board micro controllers and micro controller 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: Hardware image.

4.3.1 Operation of 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 Arduinocompatible products commercially released have avoided the project name by using various names ending in -duino.

Fig: 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 addon 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.

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 Bo 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.

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: Arduino board.

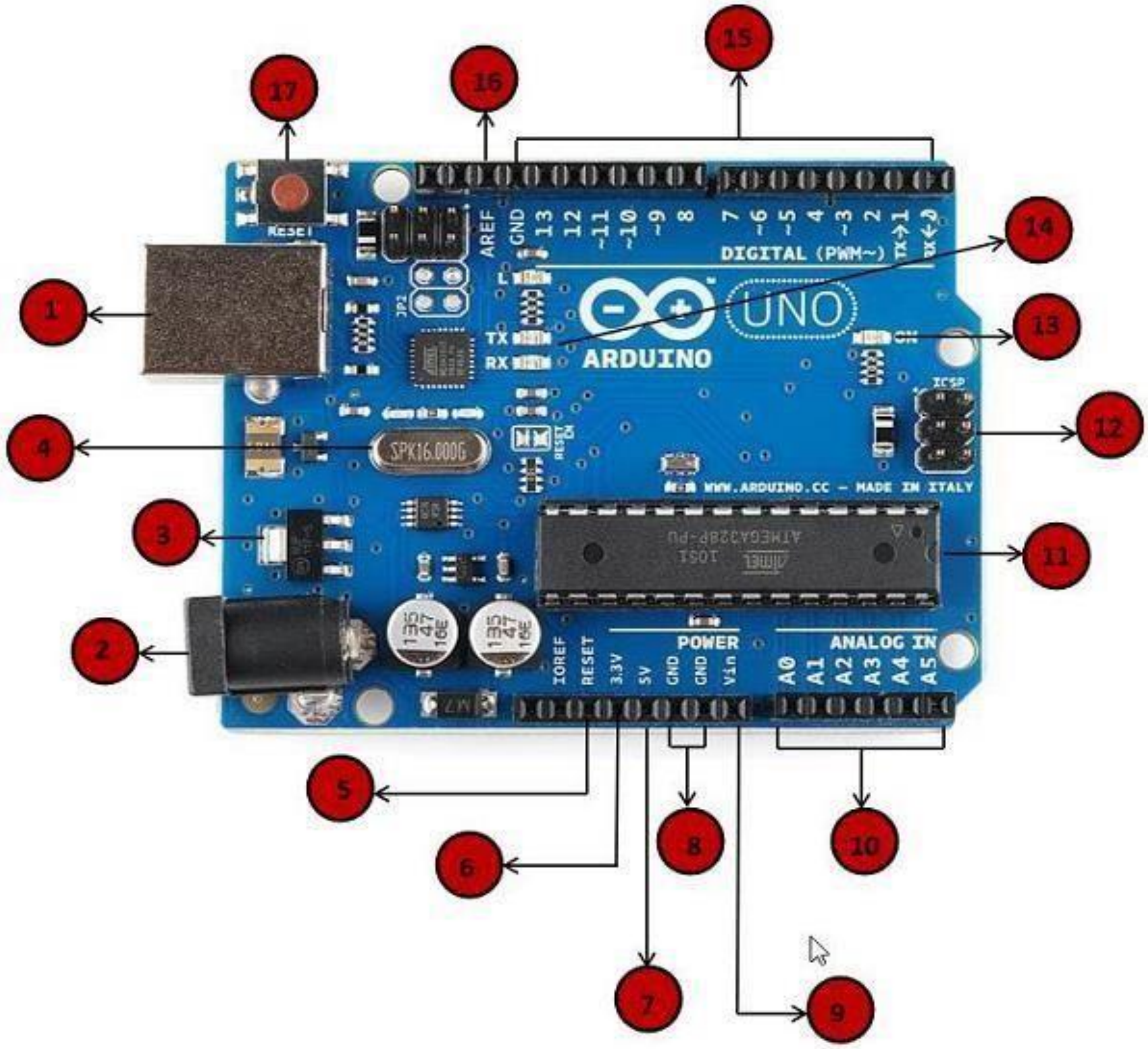


Fig. Pin explanation.

Table. Pin Diagram Description

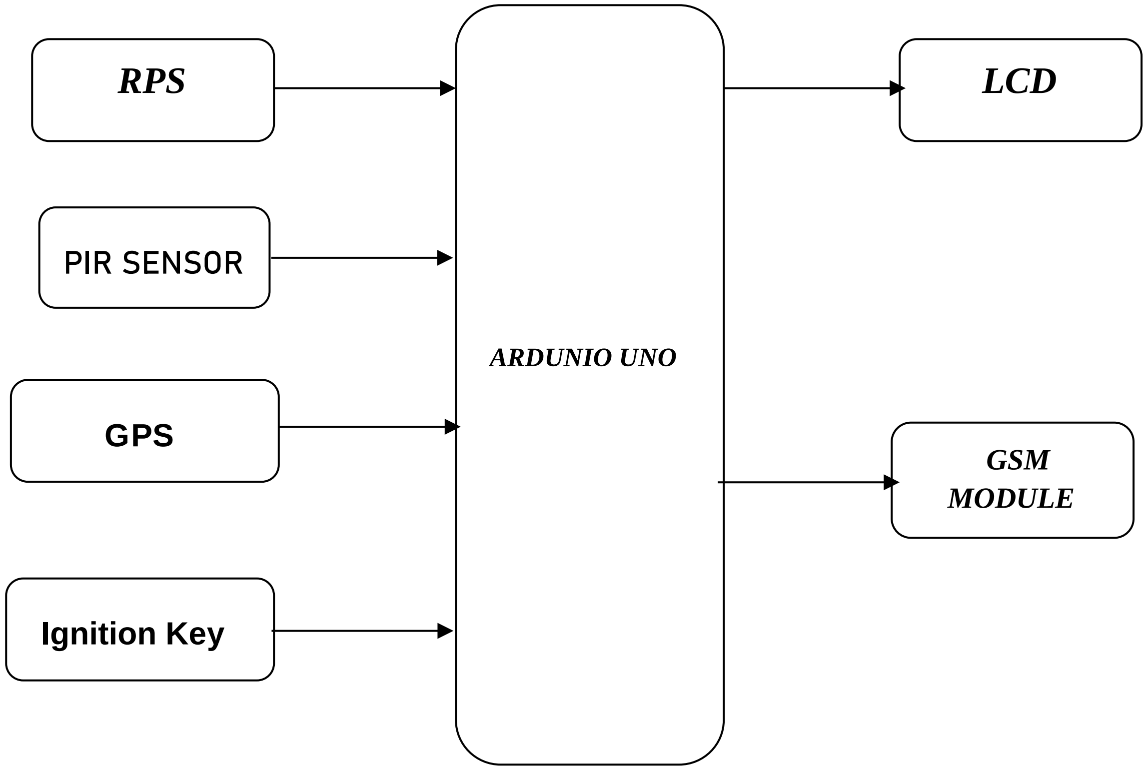
1	POWER USB: Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	POWER (BARREL JACK): Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	VOLTAGE REGULATOR: 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.
4	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.
5,17	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).

6,7,8,9	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 volt. • 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.
14	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.
15	DIGITAL INPUT-OUTPUT: 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.
16	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.

5.DESIGN

BLOCK DIAGRAM:



6.IMPLEMENTATION

6.0: 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

6.0: 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 the each button has the following functions:

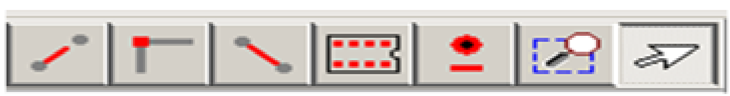


Fig: 4.1 Tool bar necessary for the interface

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 pad

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.

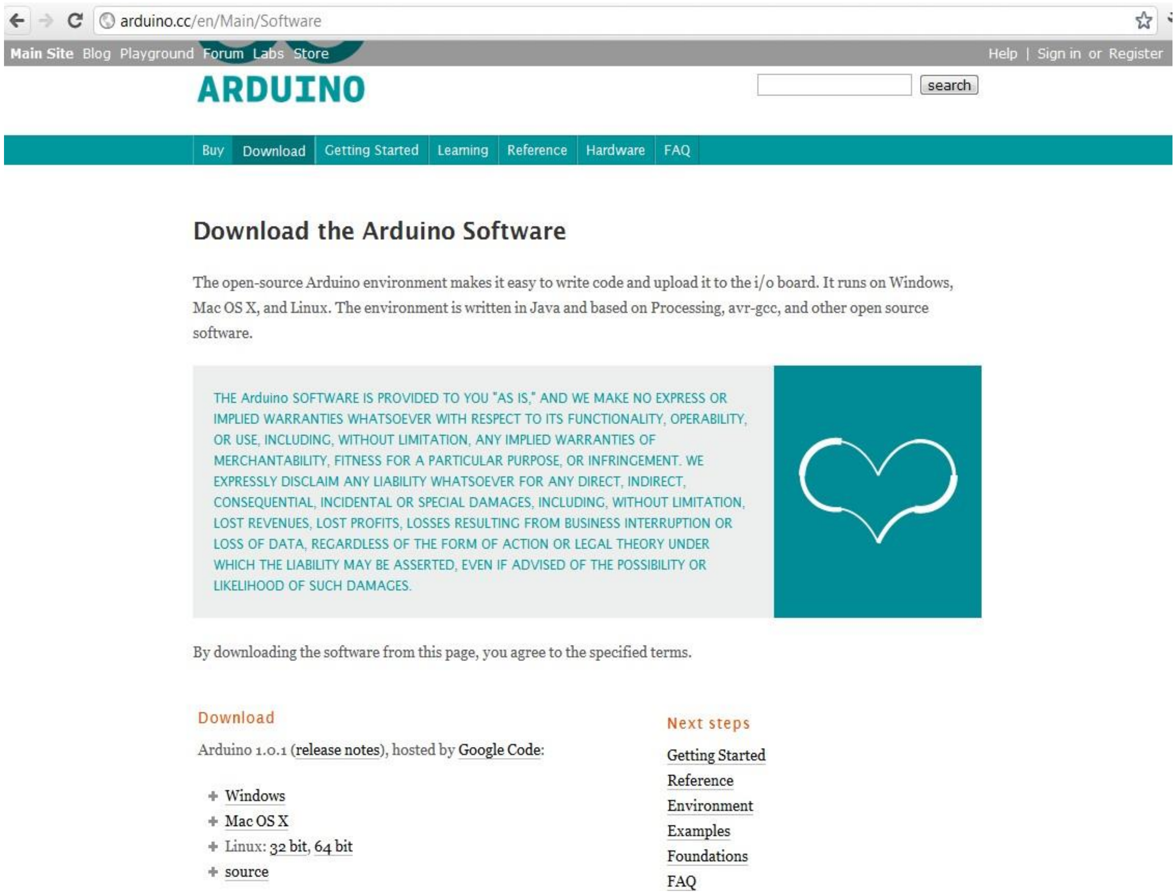
4.0 : Design Considerations

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?

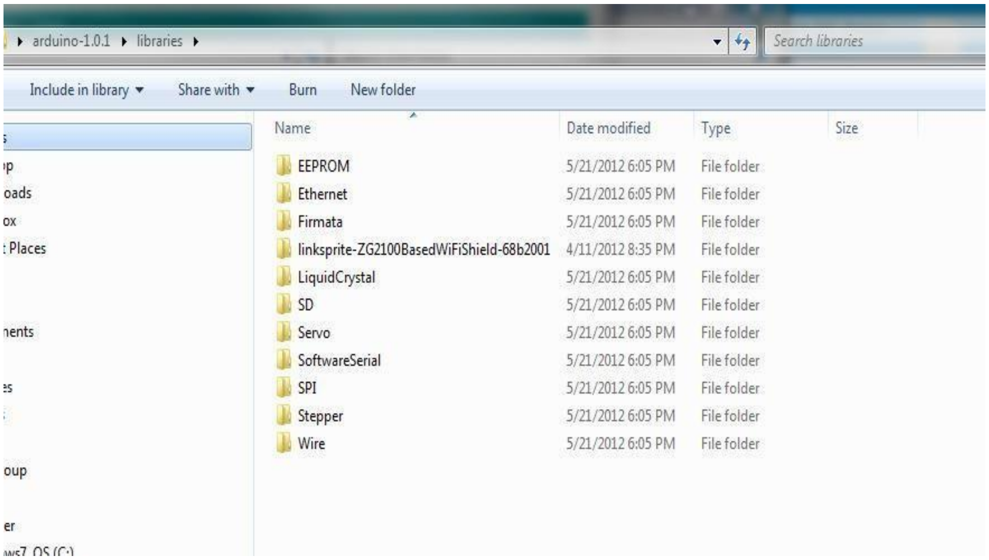
When making a PCB you have the option of making a single sided board, or a double sided board. Single sided boards are cheaper to produce and easier to etch, but much harder to design for large projects. If a lot of parts are being used in a small space it may be difficult to make a single sided board without jumpering 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.

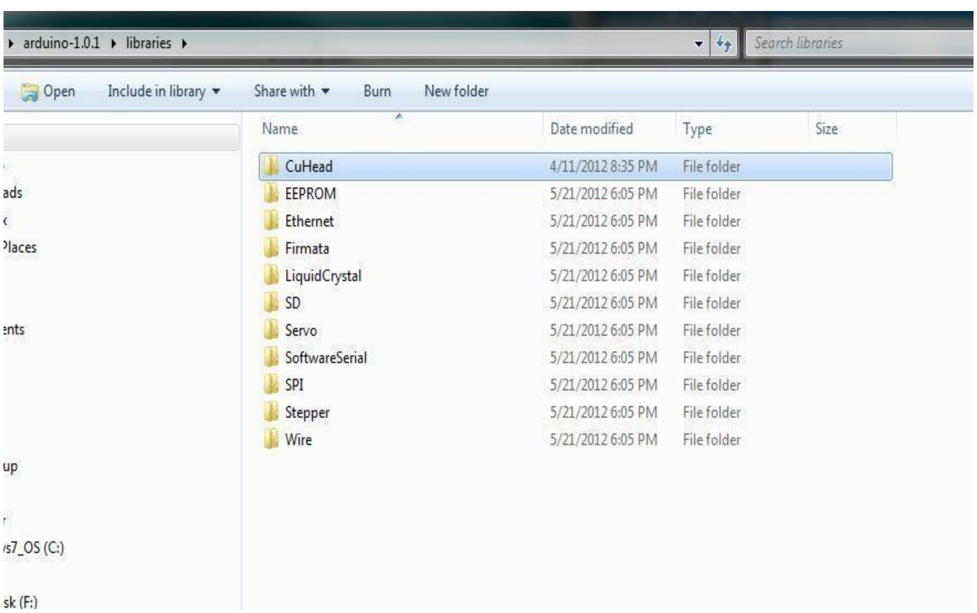
6.0 AURDINO COMPILING



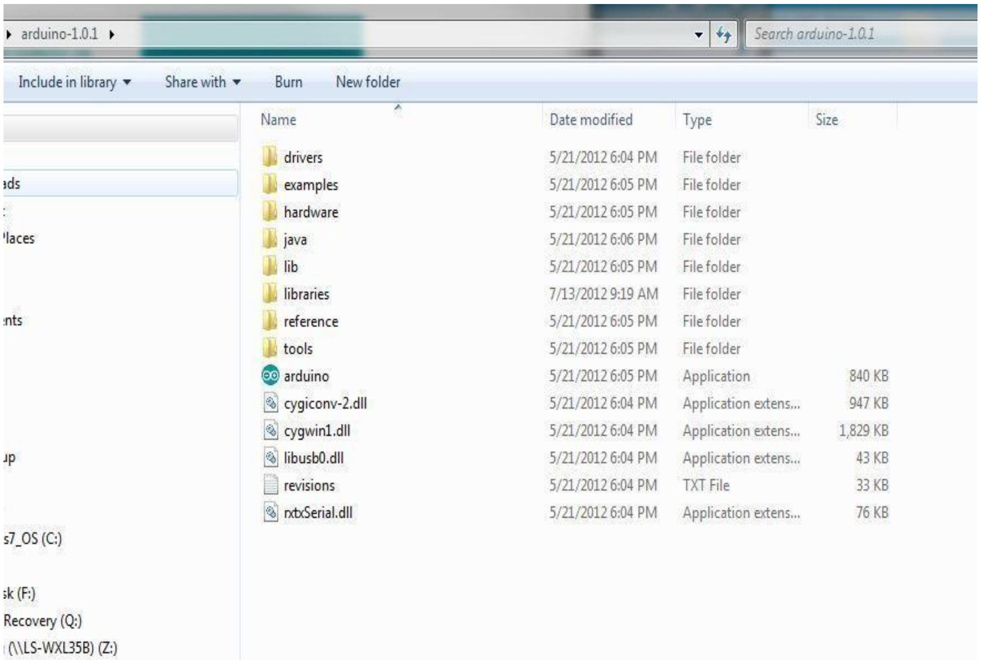
In next step download library



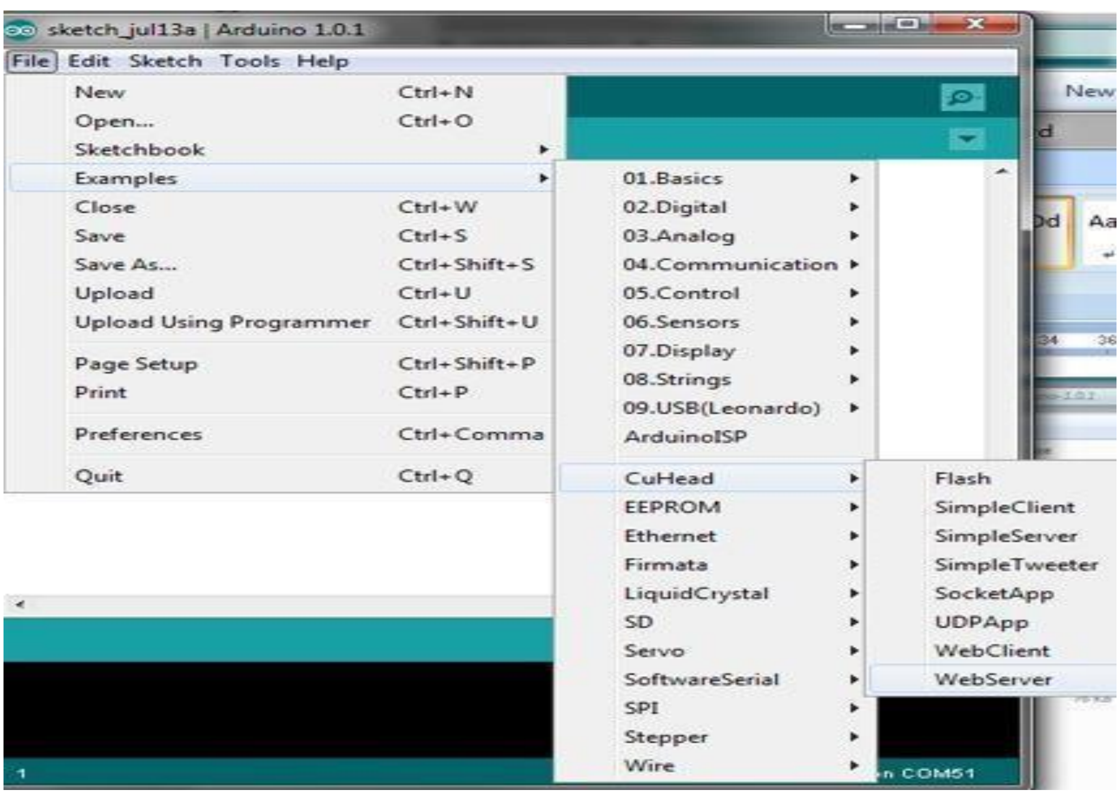
As Arduino doesn’t recognize the directory name, please rename it



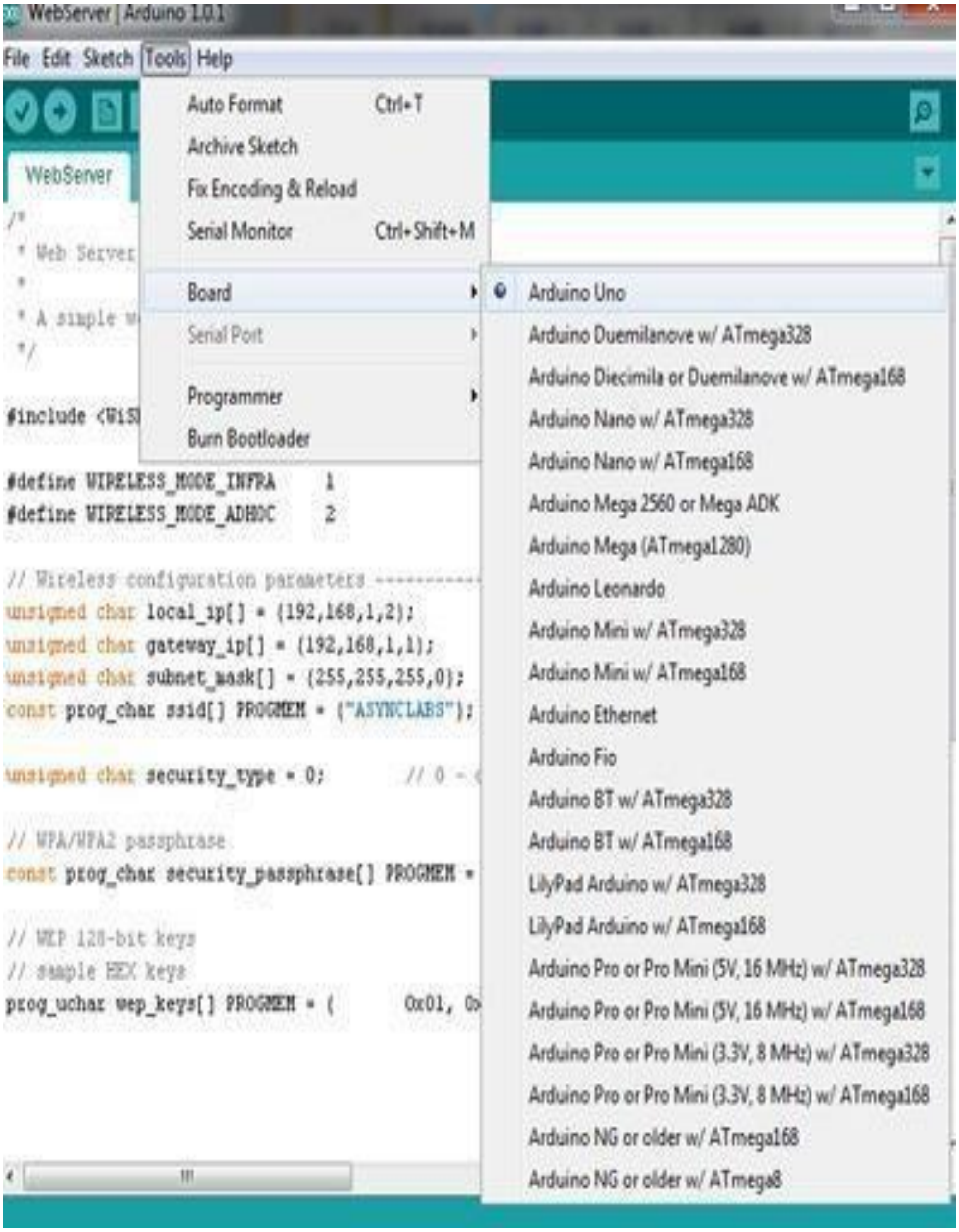
Launch Arduino by double click “Arduino” below


arduino-1.0.1
Search arduino-1.0.1
Include in library
Share with
Burn
New folder
Name
Date modified
Type
Size
drivers
examples
hardware
java
lib
libraries
reference
tools
arduino
cygiconv-2.dll
cygwin1.dll
libusb0.dll
revisions
rxtxSerial.dll
Application
Application extens...
TXT File
File folder
840 KB
947 KB
1,829 KB
43 KB
33 KB
76 KB
Recovery (Q:)

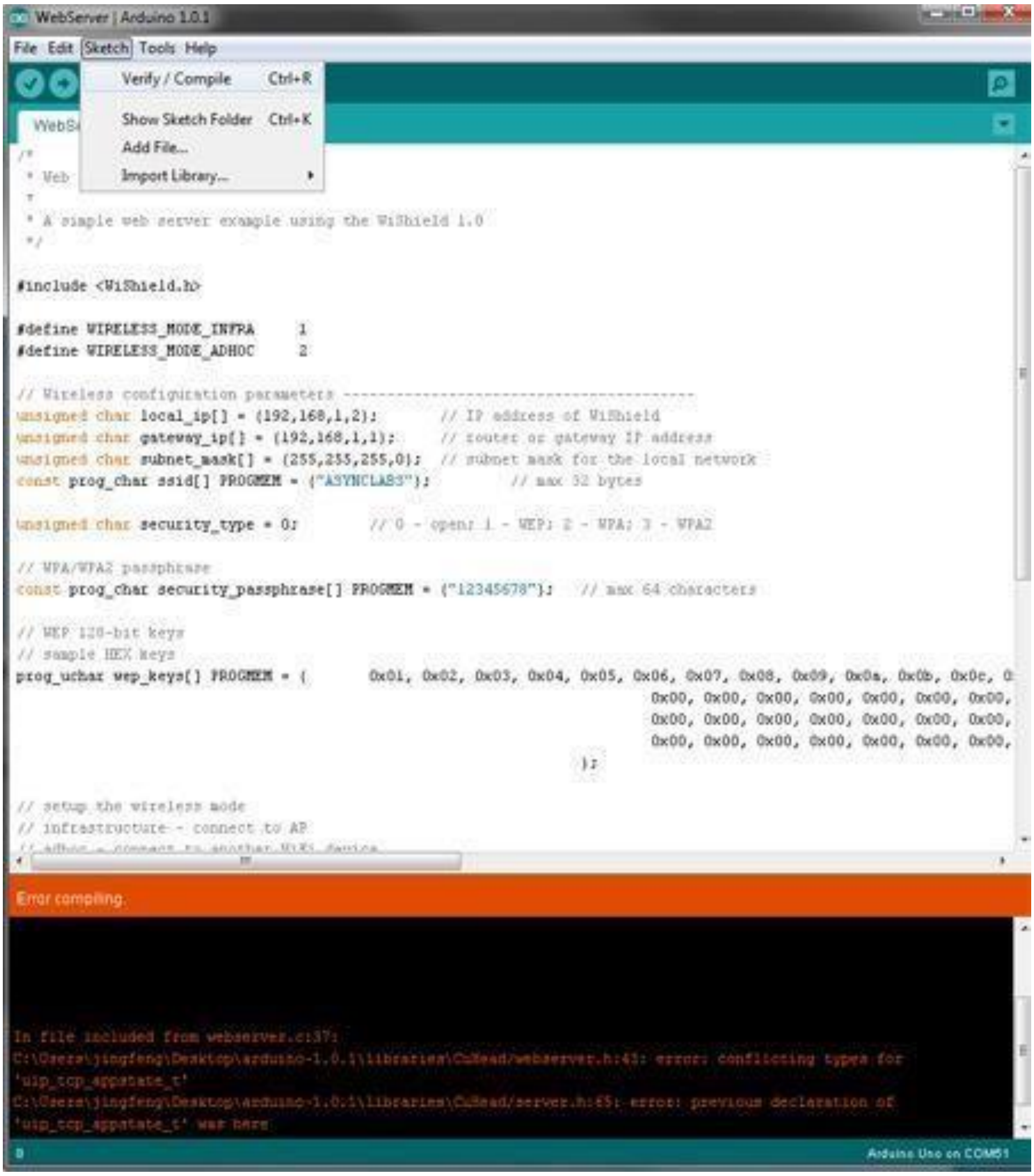
One example


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

Select the target board as “Arduino Uno”:



Click Sketch-> Verify/Compile:



FLOW CHART:

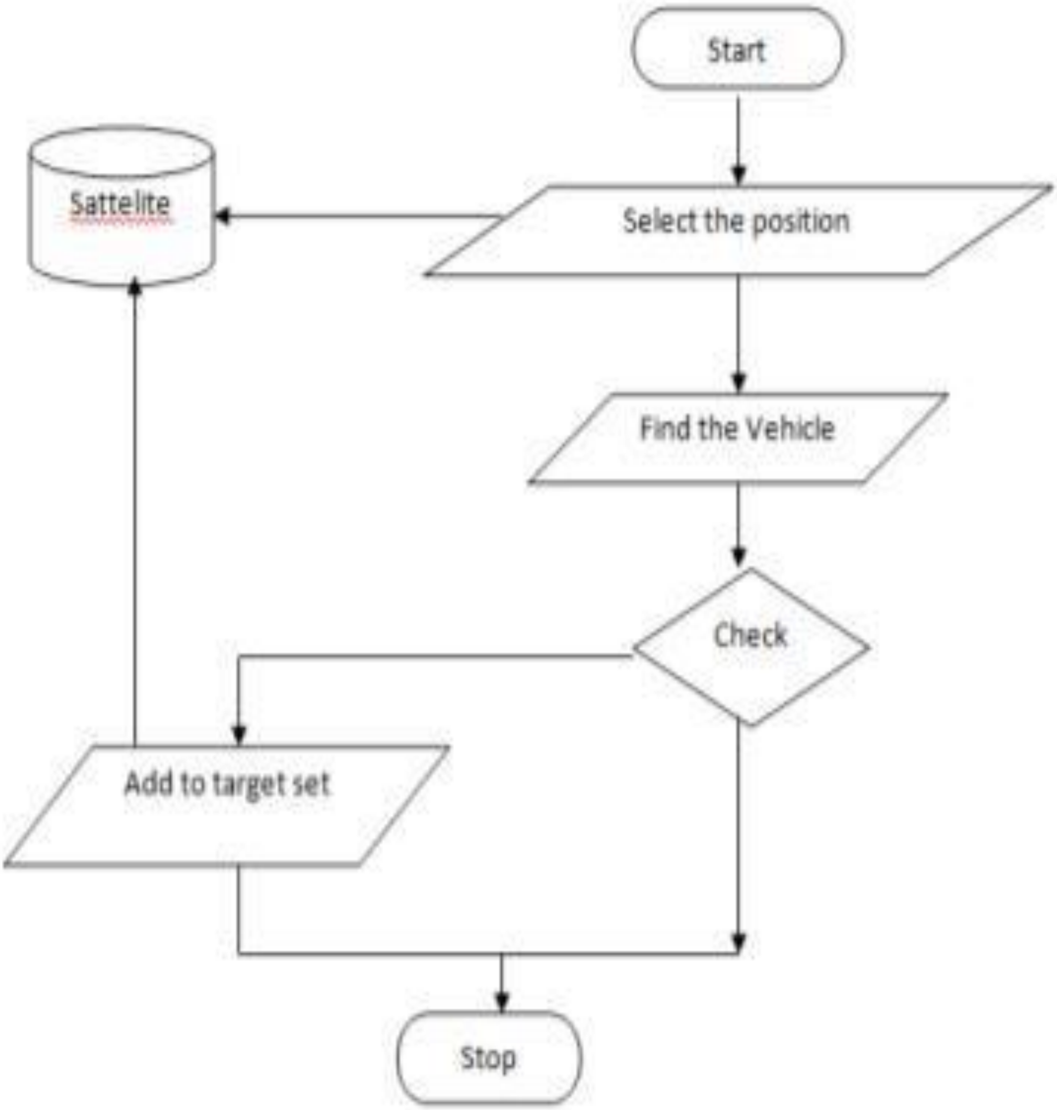


Fig. 3 GPS tracking flow chart

7. SAMPLE CODE

```
#include <LiquidCrystal.h>
```

```
#include <stdio.h>
//#include "dht.h"



LiquidCrystal lcd(6, 7, 5, 4, 3, 2);
unsigned char gchr='x',gchr1='x';  char
rcv,count;
 //char pastnumber[11]="";



int ir1     = 10; int
ir2     = 11;

int light  = 8; int
fan    = 9;



int mqv=0,soundv=0,soundv1=0; float phv=0,turbv=0;

int cntl=0,tempv=0,tempv1=0,x=0,y=0,humc=0; int st_count=0;

String inputString = "";          // a string to hold incoming data boolean
stringComplete = false;  // whether the string is complete /* void beep()
{
  digitalWrite(buzzer, LOW);delay(1500);digitalWrite(buzzer, HIGH);
}*/             void okcheck()
{
  unsigned             char     rcr;
do{
      rcr = Serial.read();
```

```
   }while(rcr != 'K');
}
void things_send()
{
unsigned char recr;
 Serial.write("AT+CIPMUX=1\r\n");delay(2000);
 Serial.write("AT+CIPSTART=4,\"TCP\",\"184.106.153.149\",80\r\n");          delay(4000);
//OK  LINKED
 Serial.write("AT+CIPSEND=4,70\r\n");  delay(3000);
 Serial.write("GET https://api.thingspeak.com/update?api_key=7Y2LX89AXYTX8G94&");
}

void things_rcv()
{
unsigned char recr;
  Serial.write("AT+CIPSTART=4,\"TCP\",\"184.106.153.149\",80\r\n");  delay(4000);
  Serial.write("AT+CIPSEND=4,73\r\n");    delay(3000);
  Serial.write("GET https://api.thingspeak.com/channels/449069/fields/3.json?results=1"); }

void things_done()
{
 Serial.write("\r\n\r\n");   delay(4000);
}

void device_control()
{   if(st_count ==
0)
{
      digitalWrite(light,LOW);digitalWrite(fan,LOW);
    }
```

```
  if(st_count >= 1 && st_count <= 5)
    {
      digitalWrite(light,HIGH);digitalWrite(fan,LOW);
    }
  if(st_count >= 6)
    {
      digitalWrite(light,HIGH);digitalWrite(fan,HIGH);
    }
}   void setup()
{
  // initialize serial:
  Serial.begin(9600);//serialEvent();


  pinMode(light,                    OUTPUT);
pinMode(fan, OUTPUT);   pinMode(ir1,
INPUT);           pinMode(ir2,    INPUT);
digitalWrite(light,            LOW);
digitalWrite(fan, LOW);


  lcd.begin(16, 2);   lcd.print("IOT
Smart Class Room");   delay(1500);

  wifiinit();

  lcd.clear();
```

```
   lcd.setCursor(0,           0);                              //column,row
lcd.print("Count:");          //3,0

}

void loop()  {   if(digitalRead(ir1) ==
LOW)     {while(digitalRead(ir1) ==
LOW);
                st_count++;
lcd.setCursor(6,0);convertl(st_count);
device_control();
lcd.setCursor(13,1);lcd.print("1               ");
things_send();
Serial.write("field3=");converts(st_count
);//00100");         things_done();
lcd.setCursor(13,1);lcd.print("    ");
for(cntl=0;cntl<35;cntl++)
         {lcd.setCursor(13,1);convertl1(cntl);delay(900);}         lcd.setCursor(13,1);lcd.print("
");
    }

  if(digitalRead(ir2) == LOW)
{while(digitalRead(ir2) == LOW);
st_count--;       if(st_count <=
0){st_count=0;}
lcd.setCursor(6,0);convertl(st_count);       device_control();
lcd.setCursor(13,1);lcd.print("1   ");         things_send();
        Serial.write("field3=");converts(st_count);//00100");         things_done();
lcd.setCursor(13,1);lcd.print("    ");
```

```
for(cntl=0;cntl<35;cntl++)
  {lcd.setCursor(13,1);convertl1(cntl);delay(900);}     lcd.setCursor(13,1);lcd.print("
");
  }



}



/*
 Read input serial  */ 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++;
}
  }
 }
}
```

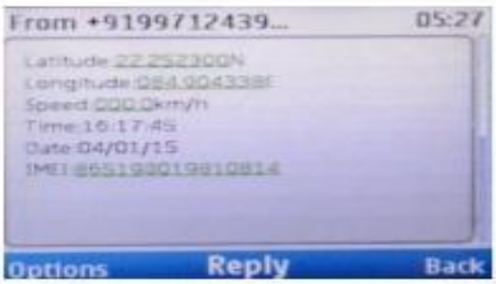
```
void wifiinit()
{
  Serial.write("AT\r\n");             delay(2000);
  Serial.write("ATE0\r\n");  okcheck();delay(2000);
  Serial.write("AT+CWMODE=3\r\n");             delay(2000);
  Serial.write("AT+CWJAP=\"iotserver\",\"iotserver@123\"\r\n"); okcheck();
Serial.write("AT+CIPMUX=1\r\n");delay(3000);//        okcheck();   lcd.clear();
lcd.print("Connected");   delay(1000);
}



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;
```

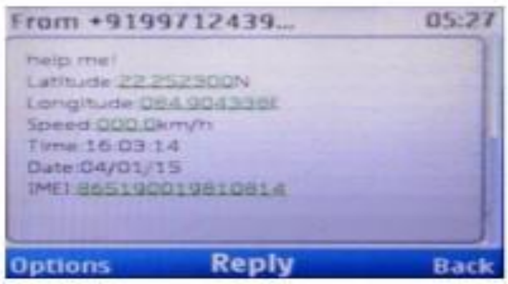
```
  Serial.write(a);
  Serial.write(c);
  Serial.write(e);
  Serial.write(g);
  Serial.write(h);
}

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(c);   lcd.write(e);
lcd.write(g);   lcd.write(h); } void
convertl1(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(g);
lcd.write(h);
}
```

8. EXPERIMENTAL RESULT

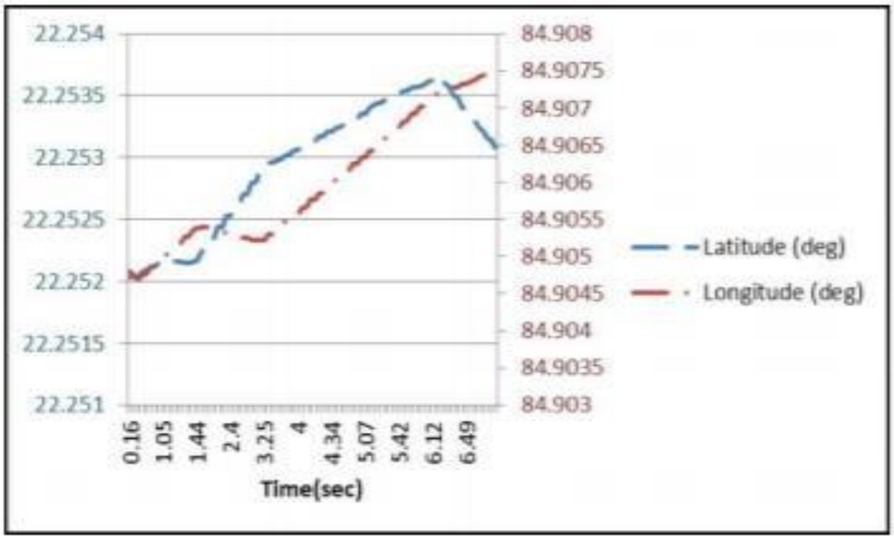
. Generate history report of vehicle.





Offline tracking using GSM.

. MSG sent in case of accident to one of the family member.



Experimental result.

9. CONCLUSION AND FUTURE SCOPE

CONCLUSION:

The project has been successfully designed and tested. Integrating features of all the hardware components used have developed it. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced IC's and with the help of growing technology the project has been successfully implemented. Embedded systems are emerging as a technology with high potential. In the past decades microprocessor based embedded system ruled the market. The last decade witnessed the revolution of Microcontroller based embedded systems.. With regards to the requirements gathered the manual work and the complexity in counting can be achieved with the help of electronic devices.

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] Isna 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] Pengfie 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. Venkatesakumar, "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