Chapter-1

INTRODUCTION TO THE PROJECT

1.1 Introduction:

Agriculture is major source of income for the largest population in India and is major contributor to Indian economy. In past decade it is observed that there is not much crop development in agriculture sector. Food prices are continuously increasing because crop rate declined. There are number of factors which is responsible for this it may be due to water waste, low soil fertility, Fertilizer abuse, climate change or diseases etc. It is very essential to make effective intervention in agriculture and the solution is IOT in integration with wireless sensor network. Internet of things (IOT) is a method of connecting everything to the internet- it is connecting object or things (such as car, home, electronic devices, etc. ...) which are previously not connected with each other main purpose of IOT is ensuring delivery of right information to right people at right time. In agriculture irrigation is the important factor as the monsoon rain falls are unpredictable and uncertain.

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today. Embedded systems are controlled by one or more main processing cores that are typically either micro controllers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors. For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites. (Each radar probably includes one or more embedded systems of its own).

1.2 Aim:

Internet of Things (IoT), a well-known branch of computer science has introduced smart farming to each farmer’s neighborhood while offering constructive green agriculture. IoT depicts a self-configuring chain of components. The efficient implementation helps agriculture, a self-discipline as nicely as reducing human work and increasing crop cultivations. This paper endorses sensible IoT based Agriculture Stick as farmers aid by obtaining Live knowledge (Temperature, Soil Moisture) of farm data. These live readings help the farmers to try clever farming and to increase their average crop yields, also the quality of plants. The Smart Agriculture with Arduino Technology supports the farmers to control the live farm data and get the desired crop cultivation results

1.3 Motivation:

The agricultural growth is enhanced with the increase in the productivity and upgradation of the plantation systems. The application of Internet of Things (IoT) technology in agriculture could have the greatest impact for increase the productivity Smart farming based on IoT technologies will enable growers and farmers to reduce waste and enhance productivity ranging from the quantity of fertilizer utilized to the number of journeys the farm vehicles have made. In IoT-based smart farming, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, PIR Sensor) and automating the irrigation system. The farmers can monitor the field conditions from anywhere. IoT-based smart farming is highly efficient when compared with the conventional approach

1.4 Conclusion:

The uses of embedded systems are virtually limitless, because every day new products are introduced to the market that utilizes embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and ARDUINO UNO chips have become much cheaper. Many embedded computers even come with extensive libraries, so that "writing your own software" becomes a very trivial task indeed. The main elements that make embedded systems unique are its reliability and ease in debugging.

Chapter-2

BLOCK DIAGRAM AND CIRCUIT DIAGRAM

2.1 Introduction:

In this chapter Block diagram of automatic room light control system is explained. The various parts of the power supply unit for the system is explained in this chapter.

2.2 Block Diagram Overview:

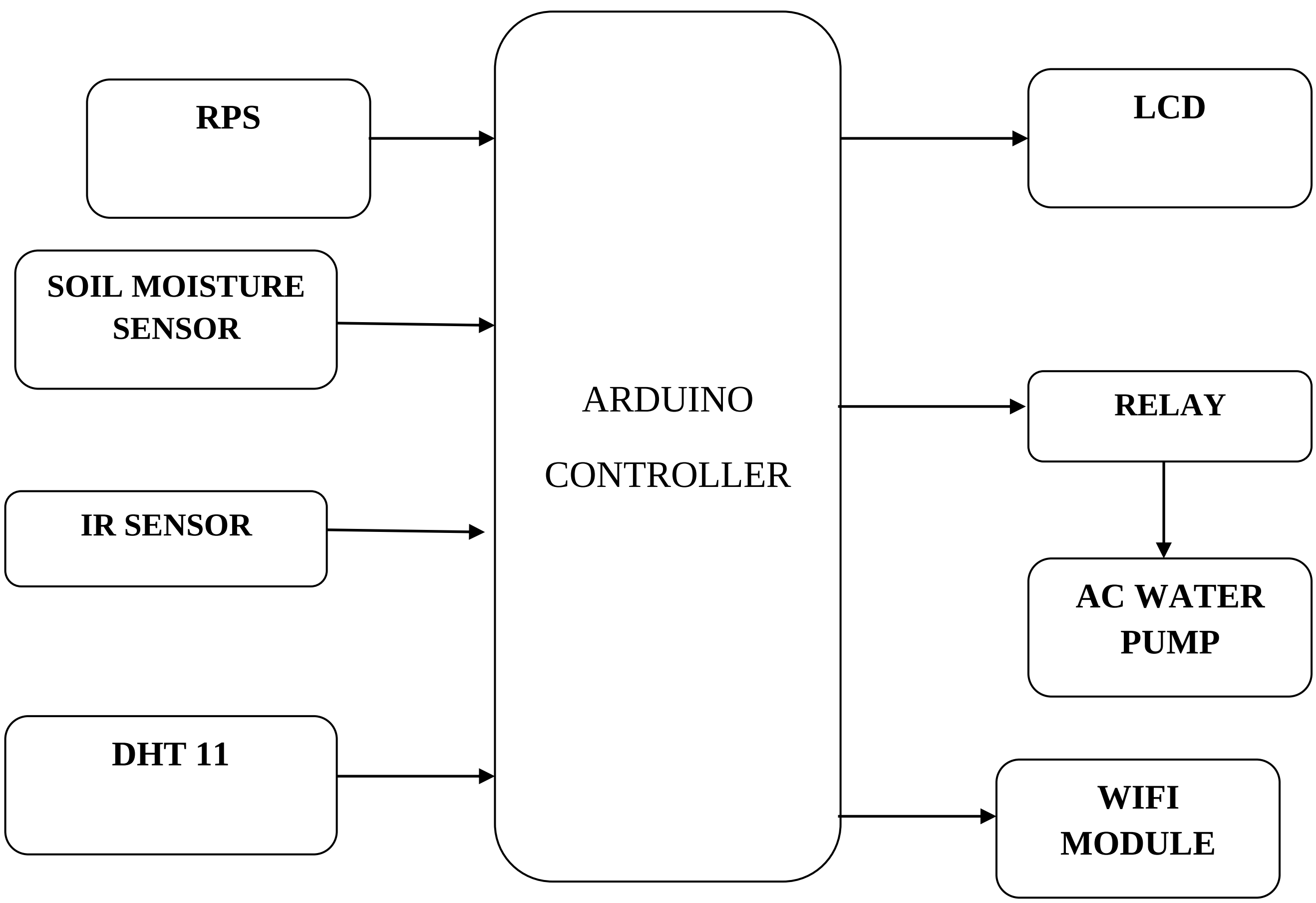


Fig.2.1 block diagram

2.3 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.

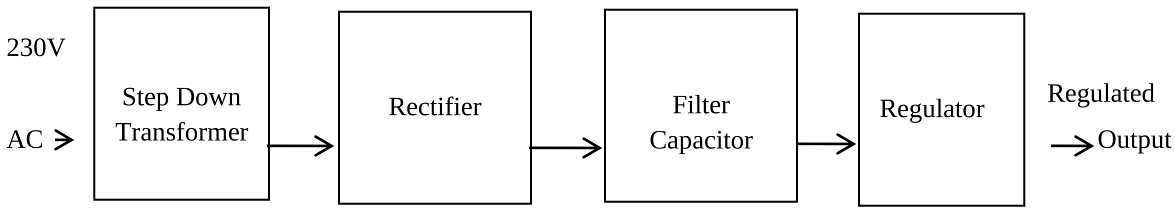
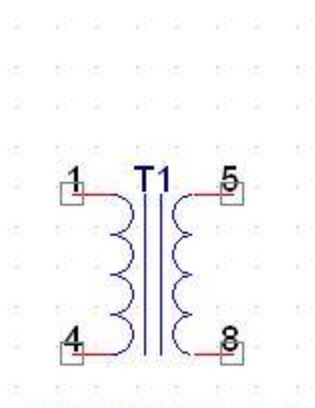


Fig:2.2 basic block diagram of a fixed regulated power supply

2.3.1 Transformer:



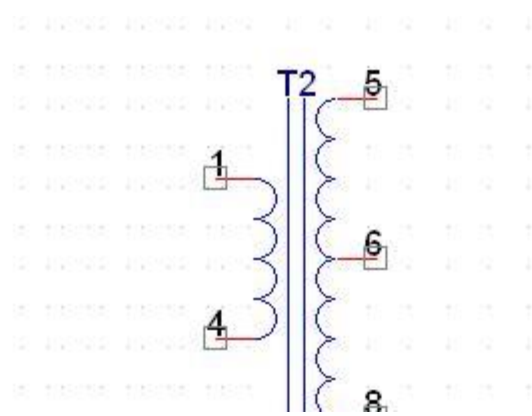


Fig 2.3: Transformer

Fig 2.4: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.

So

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

$$I_p V_p = I_s V_s$$

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

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

2.3.2 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.

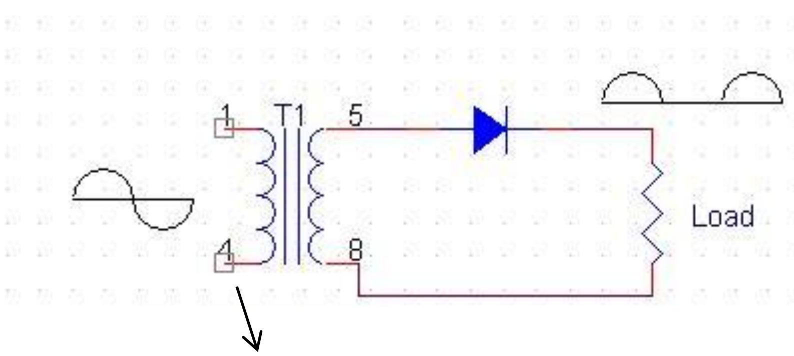


Fig 2.5: Half wave rectifier

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:

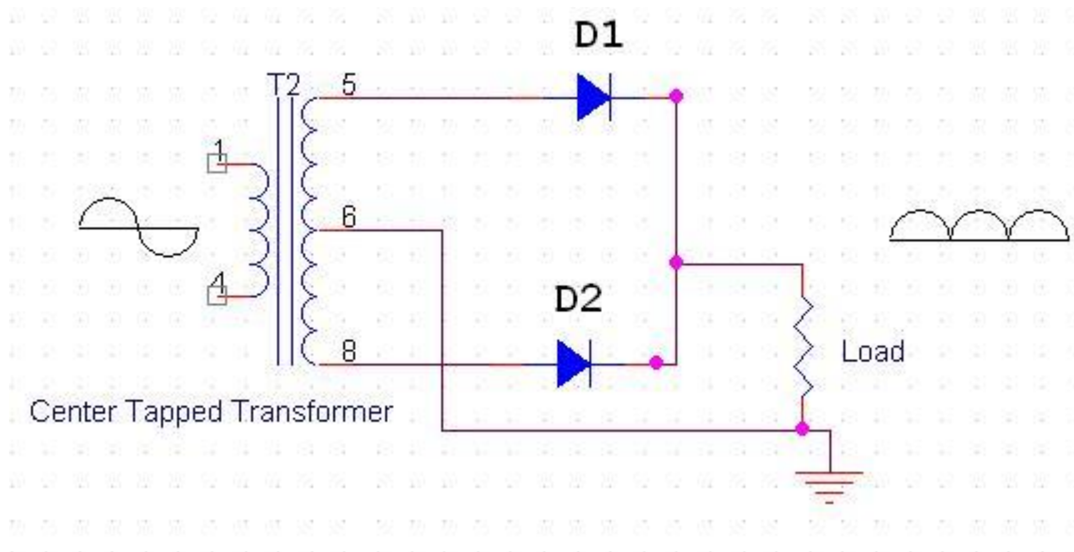


Fig 3.6: Full wave Rectifier

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:

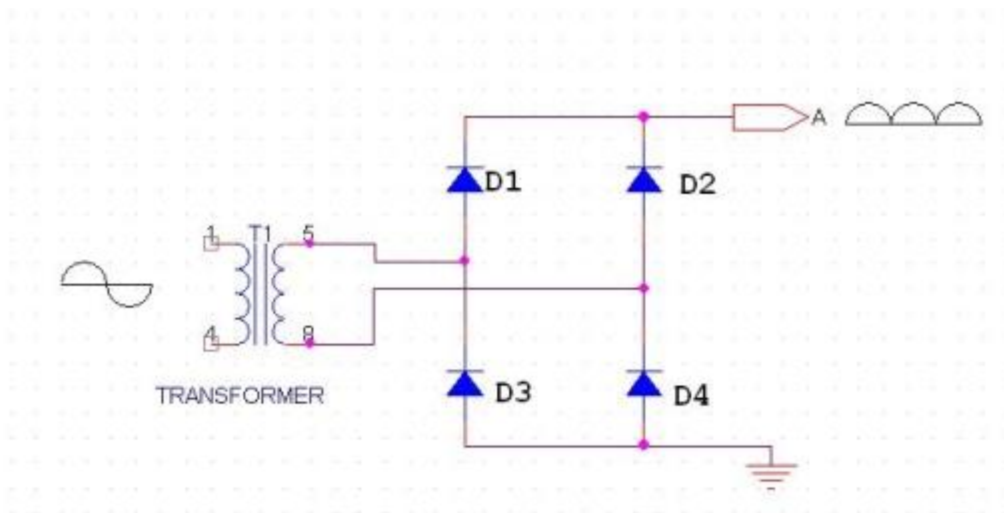


Fig 2.7: Bridge Rectifier with transformer

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.

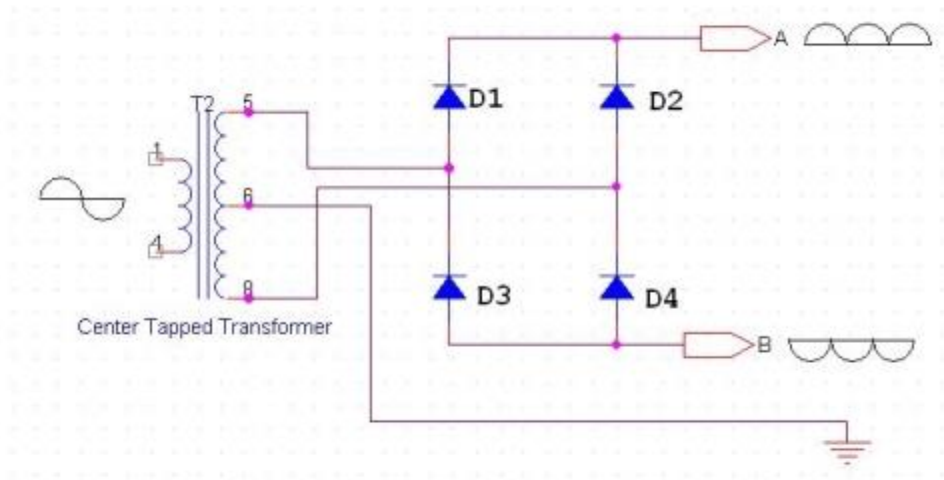


Fig 2.8 : Bridge Rectifier with center tapped transformer

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.

2.3.3 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.

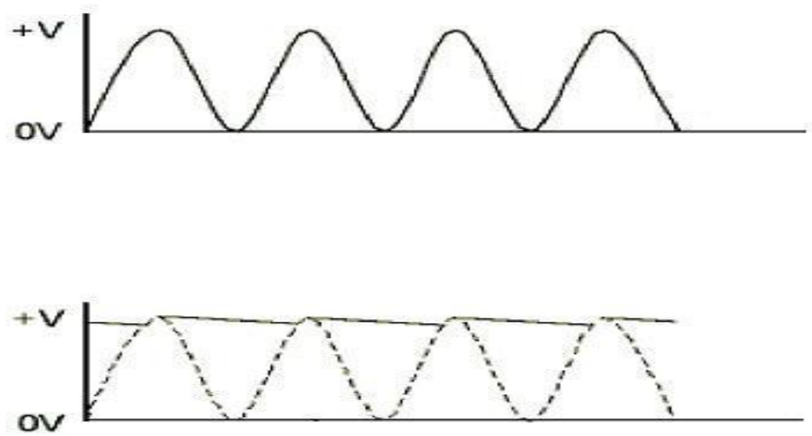


Fig 2.9: Output of Rectifier due to filter capacitor

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.

2.3.4 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.

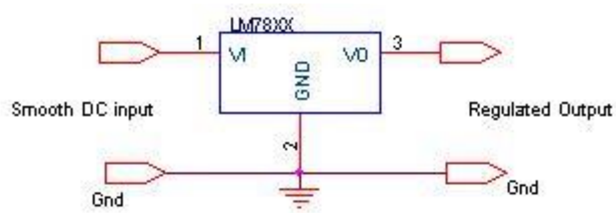


Fig 2.10 :Voltage regulator

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.

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

2.4 SOIL MOITURE SENSOR

General Description:

The LM324 series consists of four independent, high gains; internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, DC gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM124 series can be directly operated off of the standard +5V power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional ±15V power supplies.

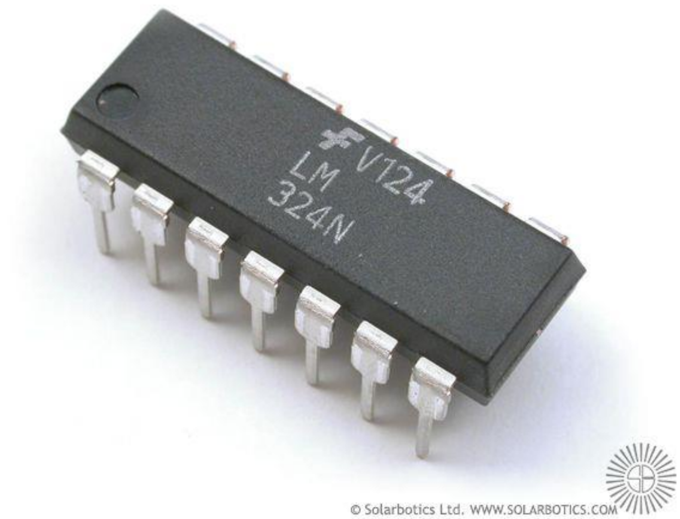
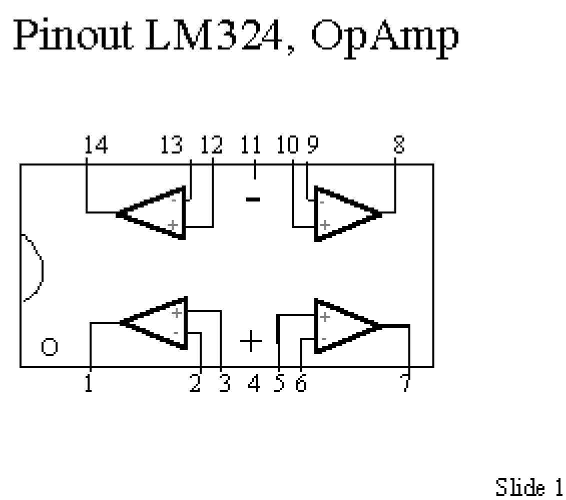


Fig 2.11 .soil moisture sensor LM324



Unique Characteristics

In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage. The unity gain cross frequency is temperature compensated. The input bias current is also temperature compensated.

PIN Diagram of LM324:

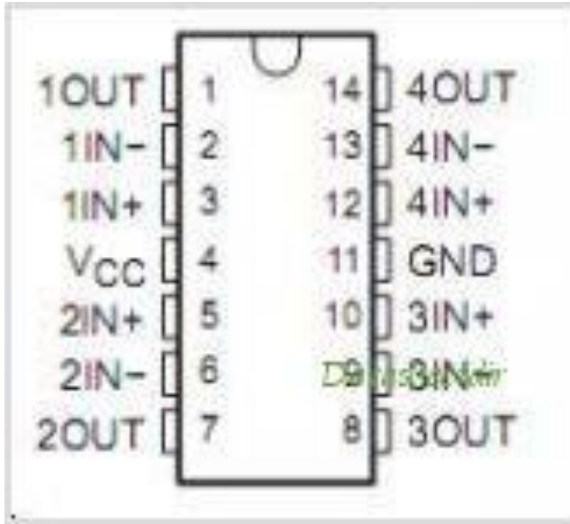
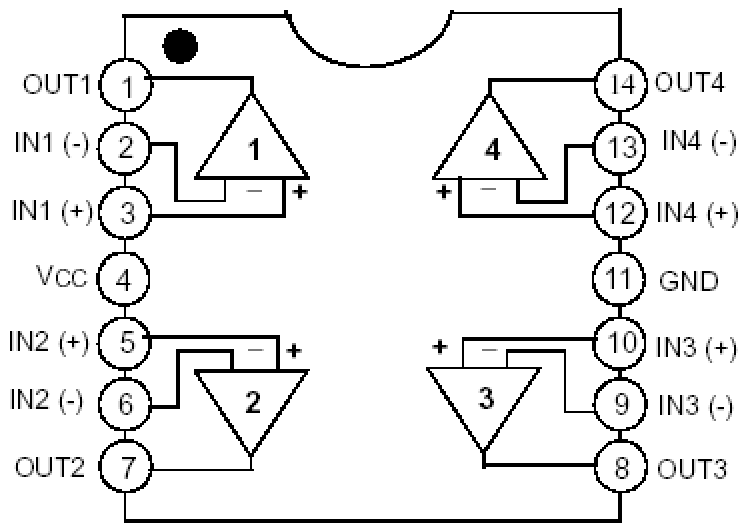


Fig.2.12 Pin diagram of LM324

Theory:

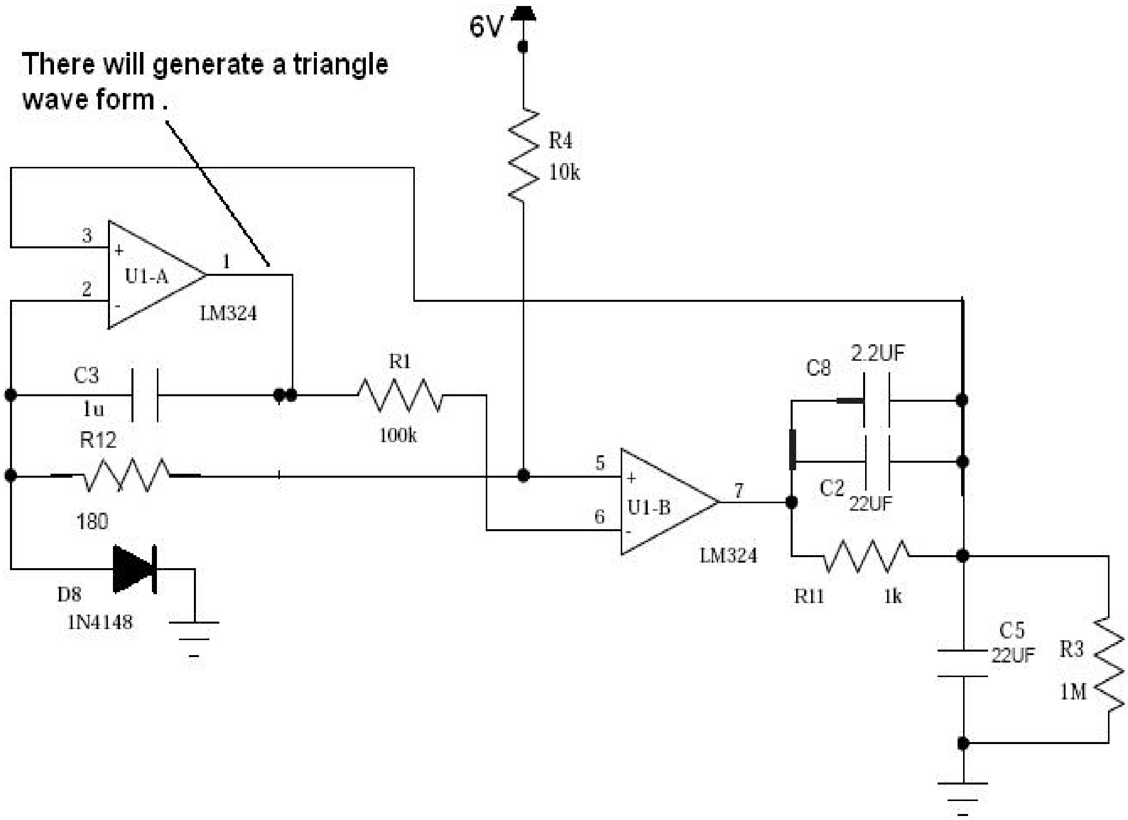
The LM124LM124/LM224/LM324/LM2902 Low Power Quad Operational Amplifiers series are op amps which operate with only a single power supply voltage, have true-differential inputs, and remain in the linear mode with an input common-mode voltage of 0 VDC. These amplifiers operate over a wide range of power supply voltage with little change in performance Characteristics. At 25°C amplifier operation is possible down to a minimum supply voltage of 2.3 VDC.

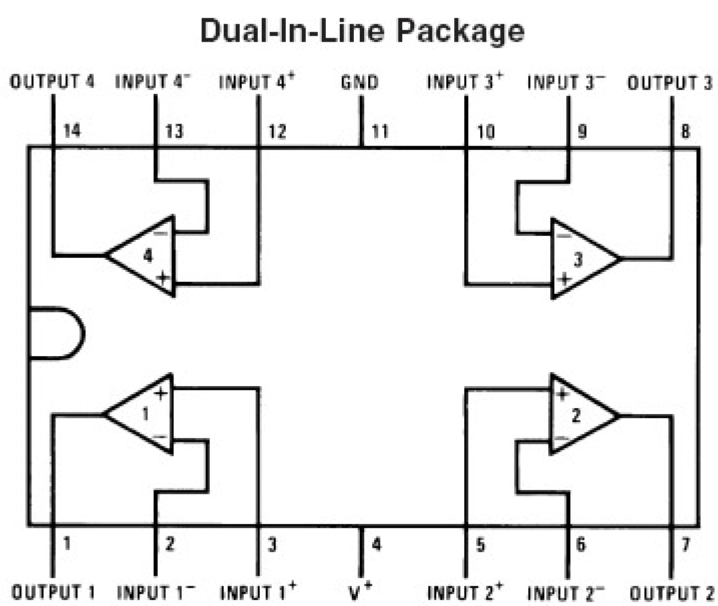
The pinouts of the package have been designed to simplify PC board layouts. Inverting inputs are adjacent to outputs for all of the amplifiers and the outputs have also been placed at the corners of the package (pins 1, 7, 8, and 14). Precautions should be taken to insure that the power supply for the integrated circuit never becomes reversed in polarity or that the unit is not inadvertently installed backwards in a test socket as an unlimited current surge through the resulting forward diode within the IC could cause fusing of the internal conductors and result in a destroyed unit. Large differential input voltages can be easily accommodated and, as input differential voltage protection diodes are not needed, no large input currents result from large differential input voltages. The differential input voltage may be larger than V+ without damaging the device. Protection should be provided to prevent the input voltages from going negative more than −0.3 VDC (at 25°C). An input clamp diode with a resistor to the IC input terminal can be used.



To reduce the power supply drain, the amplifiers have a class an output stage for small signal levels which converts to class B in a large signal mode. This allows the amplifiers to both source and sinks large output currents. Therefore both NPN and PNP external current boost transistors can be used to extend the power capability of the basic amplifiers. The output voltage needs to raise approximately 1 diode drop above ground to bias the on-chip vertical PNP transistor for output current sinking applications. For ac applications, where the load is capacitive coupled to the output of the amplifier, a resistor should be used, from the output of the amplifier to ground to increase the class a bias current and prevent crossover distortion. Where the load is directly coupled, as in dc applications, there is no crossover distortion. Capacitive loads which are applied directly to the output of the amplifier reduce the loop stability margin. Values of

50 pF can be accommodated using the worst-case non inverting unity gain connection. Large closed loop gains or resistive isolation should be used if larger load capacitance must be driven by the amplifier.





LM124/LM224/LM324/LM2902

The bias network of the LM124 establishes a drain current which is independent of the magnitude of the power supply voltage over the range of from 3 VDC to 30 VDC. Output short circuits either to ground or to the positive power supply should be of short time duration. Units can be destroyed, not as a result of the short circuit current causing metal fusing, but rather due to the large increase in IC chip dissipation which will cause eventual failure due to excessive junction temperatures. Putting direct short-circuits on more than one amplifier at a time will increase the total IC power dissipation to destructive levels, if not properly protected with external dissipation limiting resistors in series with the output leads of the amplifiers. The larger value of output source current which is available at 25°C provides a larger output current capability at elevated temperatures (see typical performance characteristics) than a standard IC op amp.

The circuits presented in the section on typical applications emphasize operation on only a single power supply voltage. If complementary power supplies are available, all of the standard op amp circuits can be used. In general, introducing a pseudo-ground (a bias voltage reference of V+/2) will allow operation above and below this value in single power supply systems. Many application circuits are shown which take advantage of the wide input common-mode voltage range which includes ground. In most cases, input biasing is not required and input voltages which range to ground can easily be accommodated.

Features:

1. Internally frequency compensated for unity gain
2. Large DC voltage gain 100 dB
3. Wide bandwidth (unity gain) 1 MHz (temperature compensated)
4. Wide power supply range: Single supply 3V to 32V or dual supplies ±1.5V to ±16V
5. Very low supply current drain (700 μA)—essentially independent of supply voltage
6. Low input biasing current 45 nA (temperature compensated)
7. Low input offset voltage 2 mV and offset current: 5 nA
8. Input common-mode voltage range includes ground
9. Differential input voltage range equal to the power supply voltage

10. Large output voltage swing 0V to V+ − 1.5V

Advantages:

1. Eliminates need for dual supplies
2. Four internally compensated op amps in a single package
3. Allows directly sensing near GND and VOUT also goes to GND
4. Compatible with all forms of logic
5. Power drain suitable for battery operation

2.5. ESP8266

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

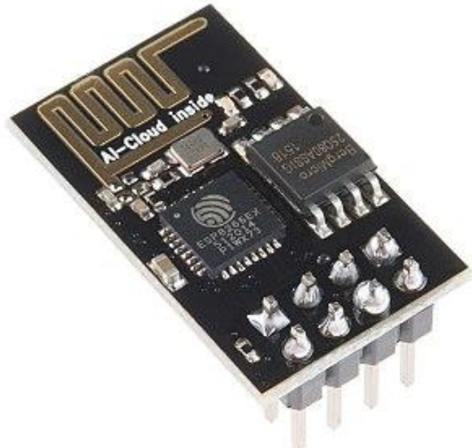


Fig.2.13 ESP 8266

- Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz[5]
- Memory:
  - 32 KiB instruction RAM
  - 32 KiB instruction cache RAM
  - 80 KiB user-data RAM
  - 16 KiB ETS system-data RAM
- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)
- IEEE 802.11 b/g/n Wi-Fi
  - Integrated TR switch, balun, LNA, power amplifier and matching network
  - WEP or WPA/WPA2 authentication, or open networks
- 16 GPIO pins
- SPI
- I²C (software implementation)[6]
- I²S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 10-bit ADC (successive approximation ADC)

OPERATIONS:

In October 2014, Espressif Systems released a software development kit (SDK) for programming the chip directly, which removed the need for a separate microcontroller.[7] Since then, there have been many official SDK releases from Espressif; Espressif maintains two versions of the SDK — one that is based on FreeRTOS and the other based on callbacks.[8]

An alternative to Espressif's official SDK is the open-source ESP-Open-SDK[9] that is based on the GNU Compiler Collection (GCC) toolchain, maintained by Max Filippov.[10] Another alternative is the "Unofficial Development Kit" by Mikhail Grigorev.[11][12]

Other SDKs, mostly open-source, include:

- Arduino — A C++-based firmware. With this core, the ESP8266 CPU and its Wi-Fi components can be programmed like any other Arduino device. The ESP8266 Arduino Core is available through GitHub.
- ESP8266 BASIC — An open-source BASIC-like interpreter specifically tailored for the Internet of Things (IoT). Self-hosting browser-based development environment.
- ESP Easy — Developed by home automation enthusiasts.
- ESPHome — ESPHome is a system to control your ESP8266/ESP32 by simple yet powerful configuration files and control them remotely through home automation systems.
- ESP-Open-RTOS — Open-source FreeRTOS-based ESP8266 software framework.
- ESP-Open-SDK — Free and open (as much as possible) integrated SDK for ESP8266/ESP8285 chips.
- Espruino — An actively maintained JavaScript SDK and firmware, closely emulating Node.js. Supports a few MCUs, including the ESP8266.
- ESPurna — Open-source ESP8285/ESP8266 firmware.
- Forthright — Port of Jones Forth to the ESP8266 microcontroller.
- MicroPython — A port of MicroPython (an implementation of Python for embedded devices) to the ESP8266 platform.

- Mongoose OS — An open-source operating system for connected products. Supports ESP8266 and ESP32. Develop in C or JavaScript.[13]
- NodeMCU — A Lua-based firmware.
- PlatformIO — A cross-platform IDE and unified debugger, which sits on top of Arduino code and libraries.
- Punyforth — Forth-inspired programming language for the ESP8266.
- Sming — An actively developed asynchronous C/C++ framework with superb performance and multiple network features.
- uLisp — A version of the Lisp programming language specifically designed to run on processors with a limited amount of RAM.
- ZBasic for ESP8266 — A subset of Microsoft's widely-used Visual Basic 6, which has been adapted as a control language for the ZX microcontroller family and the ESP8266.
- Zerynth — IoT framework for programming ESP8266 and other microcontrollers in Python.

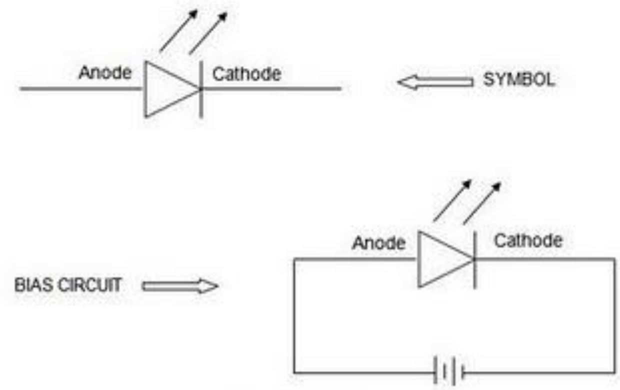
2.6. IR SENSOR:

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 LED’s 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.



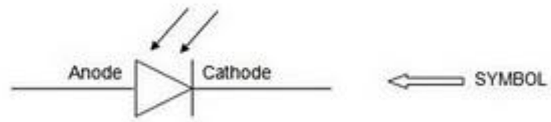
IR TRANSMITTER

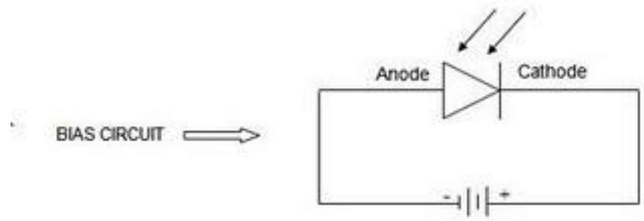
When IR rays gets emitted from LED, it moves in the direction it is angled. When any obstacle interferes in the path, the IR rays get cut and it produces secondary wavelets which propagates mostly in return direction or in a direction opposite to that of the primary waves, which produces the net result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





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 cannot use it for other application. But the infrared communication project which we have done here can be used in any application just by replacing the application at the place of infrared LED in the circuit diagram of infrared communication. By using this project we can design infrared based applications easily. The entire circuit consists of two sections named as

1. Transmitter section and
2. Receiver section

1. Transmitter section:

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

2. Receiver section:

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

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

Photo Diodes:

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

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

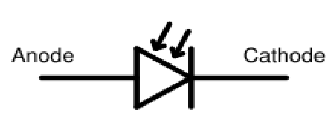


Fig (3.13) Photodiode schematic symbol

Principle of operation:

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

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

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

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

Features:

Critical performance parameters of a photodiode include:

1. Responsivity:

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

2. Dark current:

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

3. Noise-equivalent power:

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

Applications:

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

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

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

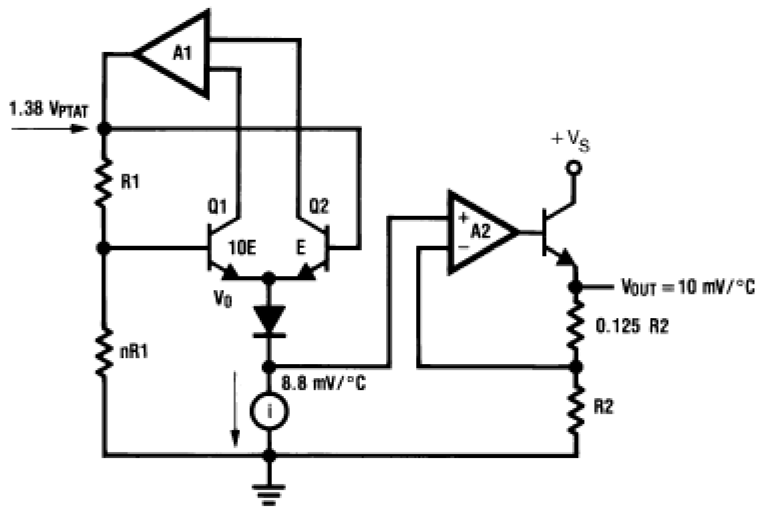
2.7. TEMPERATURE SENSOR:

The Temperature Sensor LM35 sensor series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature.

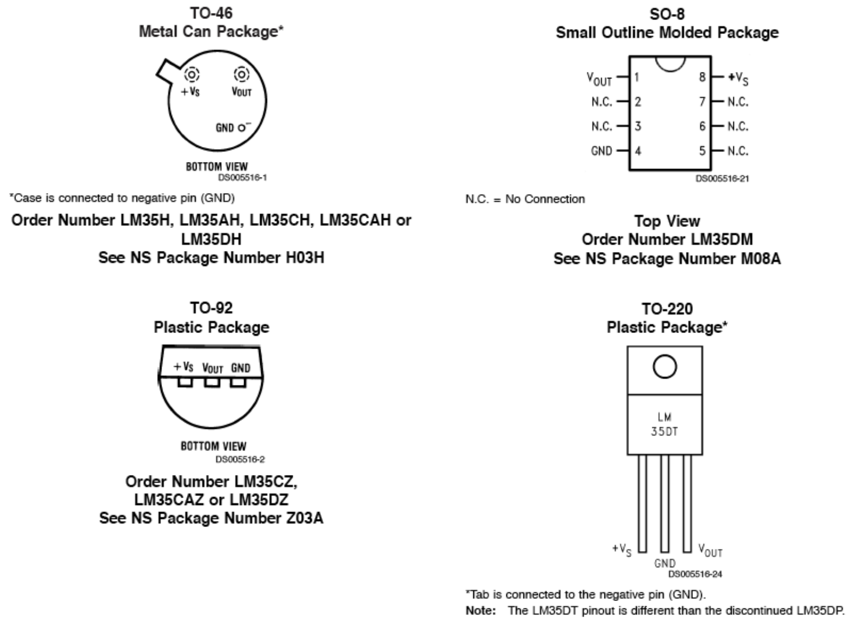
LM35 Sensor Specification:

The LM35 series are precision integrated-circuit LM35 temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 sensor thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 sensor does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full -55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55° to +150°C temperature range, while the LM35C sensor is rated for a -40° to +110°C range (-10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D sensor is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

LM35 Sensor Circuit Schematic



LM35 Sensor Pin outs and Packaging:



LM35 Sensor Sources:

There are several manufacturers of this popular part and each has LM35 sensor specs, datasheets and other free LM35 downloads. This amplifier is available from the following manufacturers.

- National Semiconductor
- On Semiconductor
- Texas Instruments
- Fairchild Semiconductor
- STMicroelectronics
- Jameco Electronics
- Analog Devices

Temperature Recorder using LM35:

Here is how you can make an LM35 a temperature recorder by using the 12F675 PIC microcontroller as the controller and data store. It generates serial output so that you can view the results on a PC and it also calculates the temperature reading in Fahrenheit sending both to the serial port at half second intervals.

LM35 Sensor Applications:

Most commonly-used electrical temperature sensors are difficult to apply. For example, thermocouples have low output levels and require cold junction compensation. Thermistors are nonlinear. In addition, the outputs of these sensors are not linearly proportional to any temperature scale. Early monolithic sensors, such as the LM3911, LM134 and LM135, overcame many of these difficulties, but their outputs are related to the Kelvin temperature scale rather than the more popular Celsius and Fahrenheit scales. Fortunately, in 1983 two IC's, the LM34 Precision Fahrenheit Temperature Sensor and the LM35 Precision Celsius Temperature Sensor, were introduced. This application note will discuss the LM34, but with the proper scaling factors can easily be adapted to the LM35.

The LM35/LM34 has an output of 10 mV/°F with a typical nonlinearity of only ±0.35°F over a −50 to +300°F temperature range, and is accurate to within ±0.4°F typically at room temperature (77°F). The LM34's low output impedance and linear output characteristic make interfacing with readout or control circuitry easy. An inherent strength of the LM34 sensor over other currently available temperature sensors is that it is not as susceptible to large errors in its output from low level leakage currents. For instance, many monolithic temperature sensors have an output of only 1 µA/°K. This leads to a 1°K error for only 1 µ-Ampere of leakage current. On the other hand, the LM34 sensor may be operated as a current mode device providing 20 µA/° F of output current. The same 1 µA of leakage current will cause an error in the LM34's output of only 0.05°F (or 0.03°K after scaling).

Low cost and high accuracy are maintained by performing trimming and calibration procedures at the wafer level. The device may be operated with either single or dual supplies. With less than 70 μA of current drain, the LM34 sensor has very little self-heating (less than 0.2°F in still air), and comes in a TO-46 metal can package, a SO-8 small outline package and a TO-92 plastic package.

The LM35/LM34 is a versatile device, which may be used for a wide variety of applications, including oven controllers and remote temperature sensing. The device is easy to use (there are only three terminals) and will be within 0.02°F of a surface to which it is either glued or cemented. The TO-46 package allows the user to solder the sensor to a metal surface, but in doing so, the GND pin will be at the same potential as that metal. For applications where a steady reading is desired despite small changes in temperature, the user can solder the TO-46 package to a thermal mass. Conversely, the thermal time constant may be decreased to speed up response time by soldering the sensor to a small heat fin.

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.[2] 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.[4] In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, 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 Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis,[2] 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 -duino.

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 optiboot 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 Boarduino, 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 Diecimila,[a] Duemilanove,[b] and current Uno[c] 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 Boarduino 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.

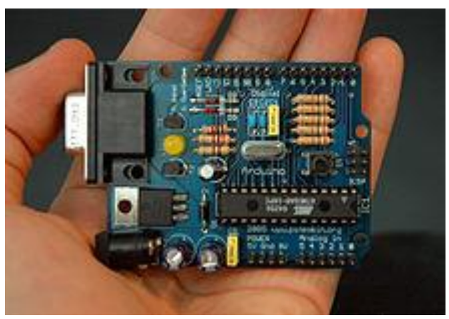


Fig.3.3. Arduino board.

1. **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 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.
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.

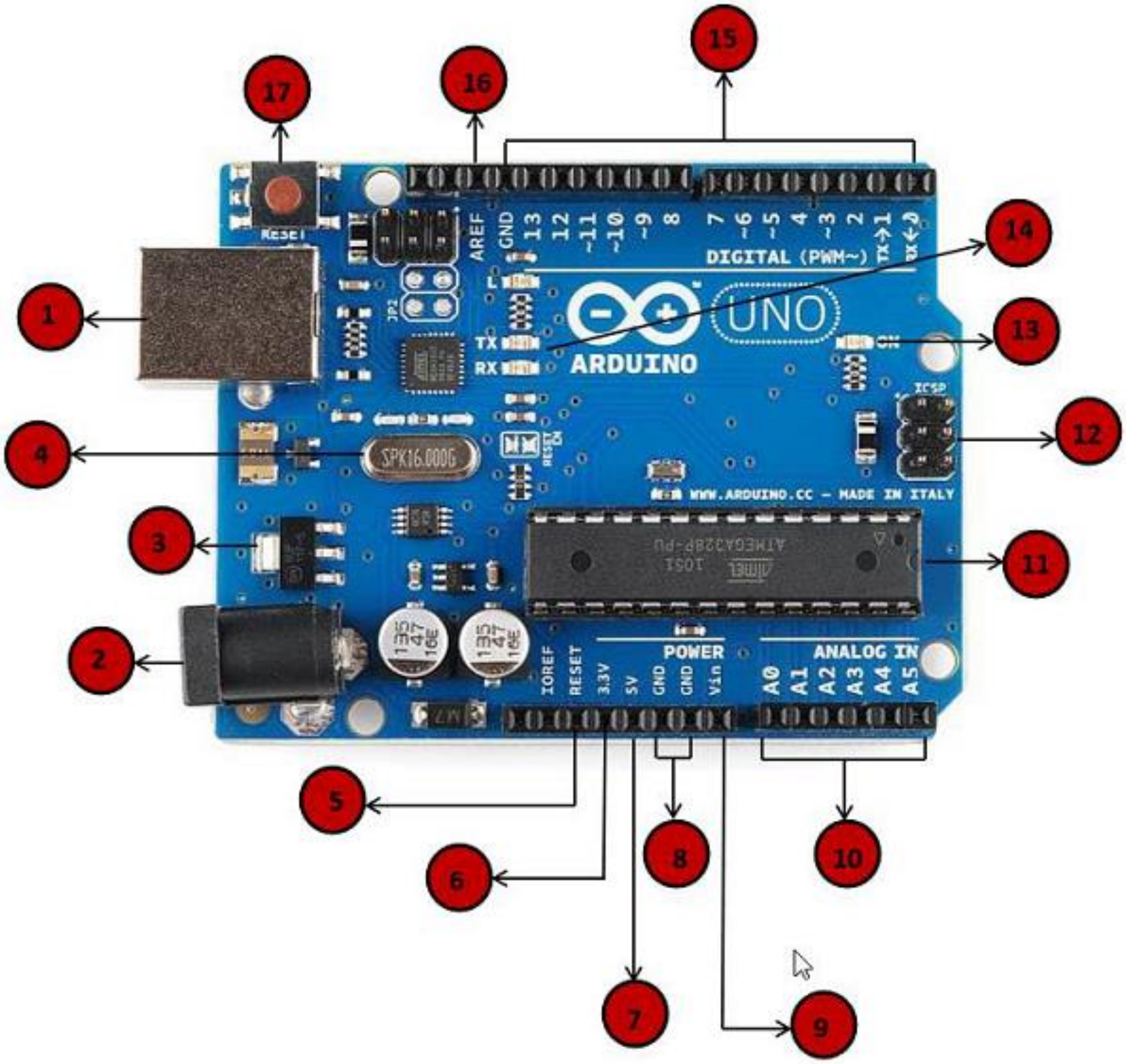


Fig.3.4. Pin explanation.

CHAPTER-4

SOFTWARE EXPLANATION

4.0 : 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

4.1 : 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:

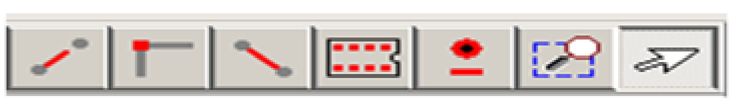


Fig: 4.1 Tool bar necessary for the interface

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 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.

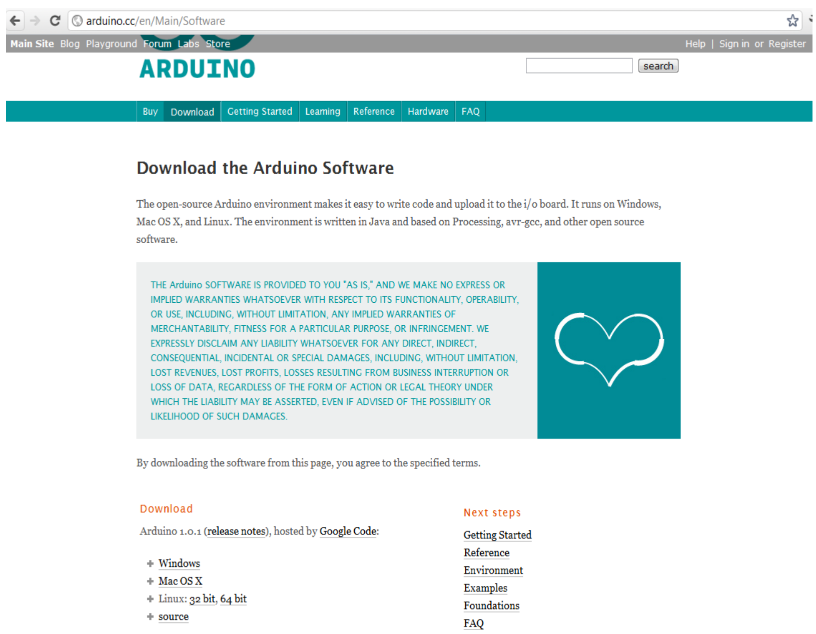
4.2 : Design Considerations

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?

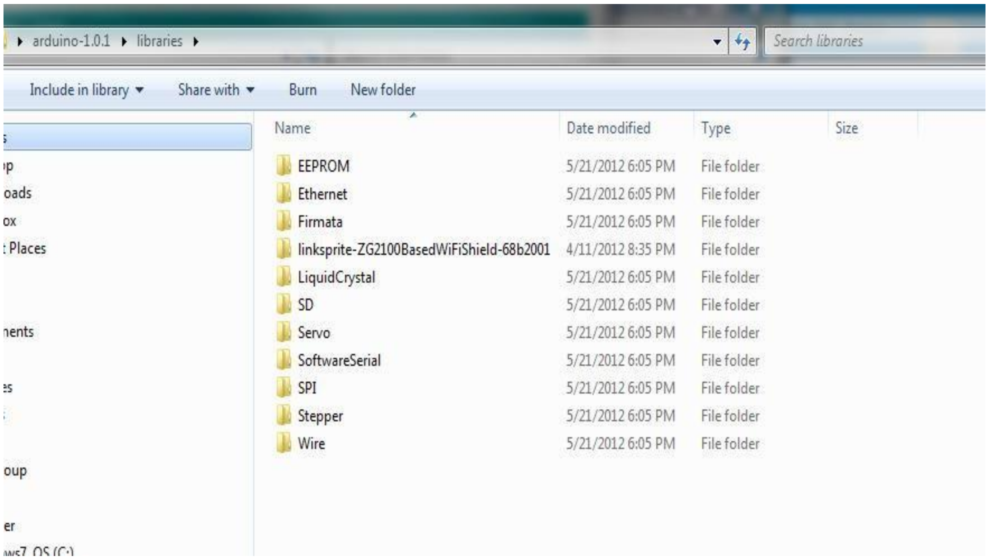
When making a PCB you have the option of making a single sided board, or a double sided board. Single sided boards are cheaper to produce and easier to etch, but much harder to design for large projects. If a lot of parts are being used in a small space it may be difficult to make a single sided board 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.

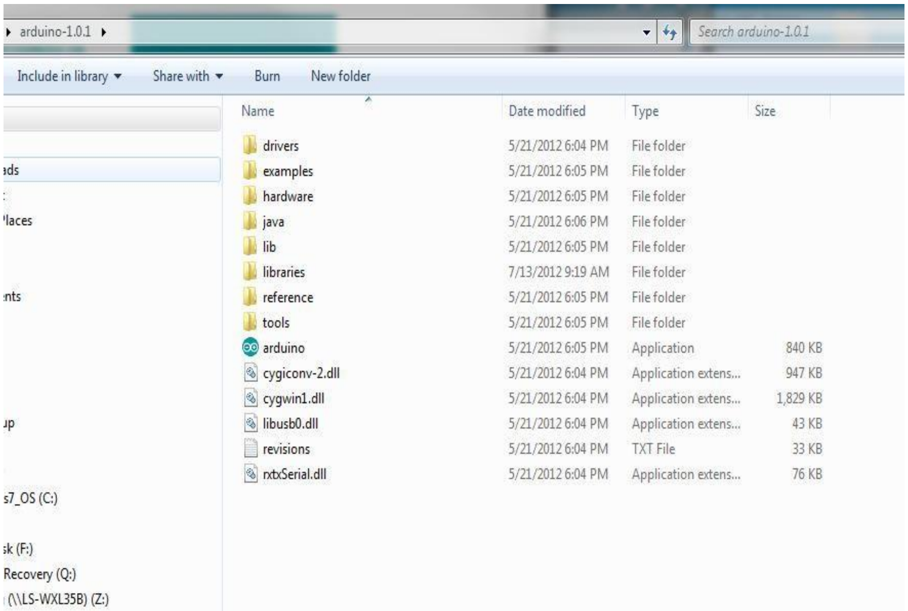
4.3 AURDINO COMPILING



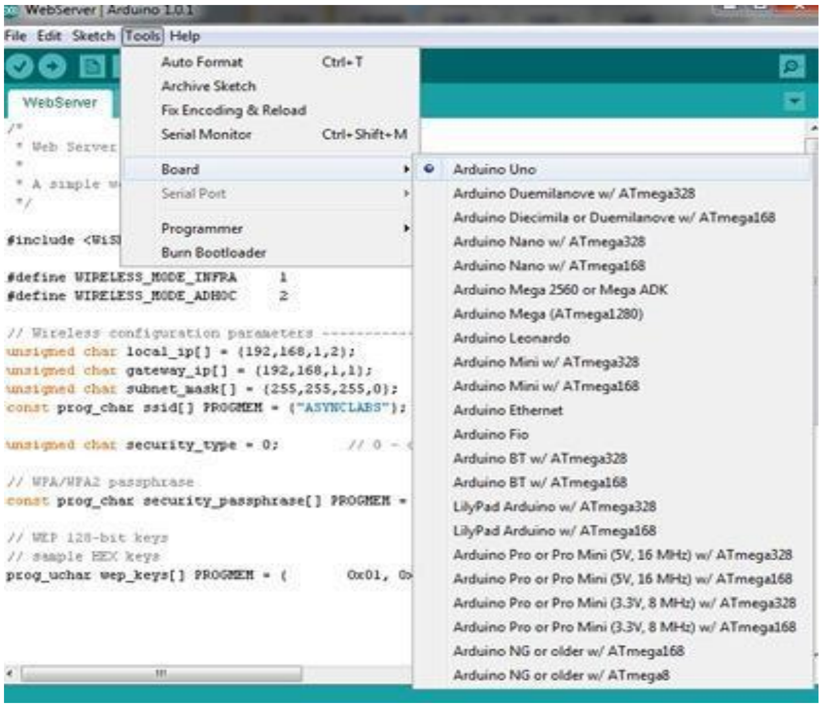
In next step download library



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

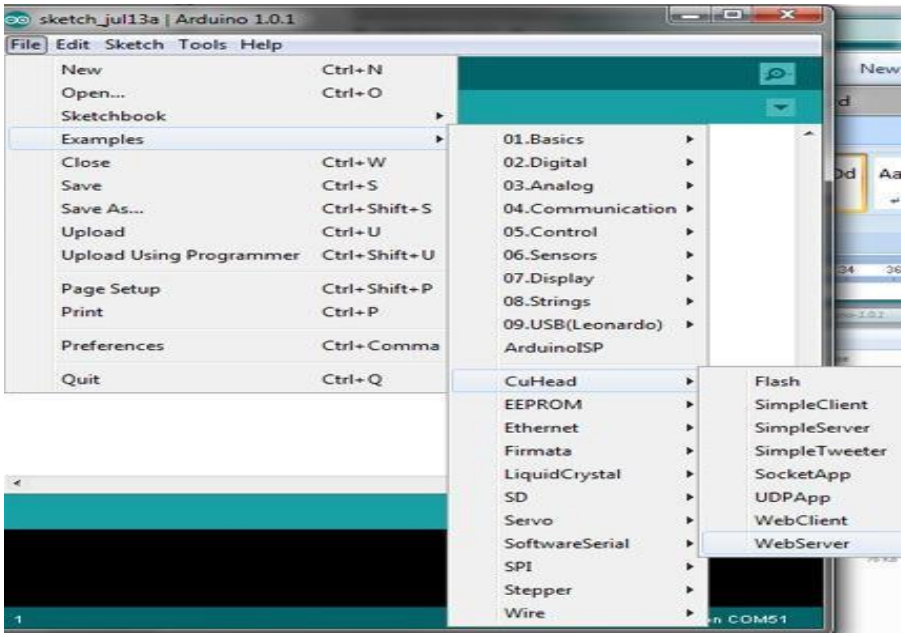

arduino-1.0.1
Search arduino-1.0.1
Include in library
Share with
Burn
New folder
Name
Date modified
Type
Size
drivers 5/21/2012 6:04 PM File folder
examples 5/21/2012 6:05 PM File folder
hardware 5/21/2012 6:05 PM File folder
java 5/21/2012 6:06 PM File folder
lib 5/21/2012 6:05 PM File folder
libraries 7/13/2012 9:19 AM File folder
reference 5/21/2012 6:05 PM File folder
tools 5/21/2012 6:05 PM File folder
arduino 5/21/2012 6:05 PM Application 840 KB
cygiconv-2.dll 5/21/2012 6:04 PM Application extens... 947 KB
cygwin1.dll 5/21/2012 6:04 PM Application extens... 1,829 KB
libusb0.dll 5/21/2012 6:04 PM Application extens... 43 KB
revisions 5/21/2012 6:04 PM TXT File 33 KB
rxtxSerial.dll 5/21/2012 6:04 PM Application extens... 76 KB
Recovery (Q:)

Launch Arduino by double click “Arduino” below

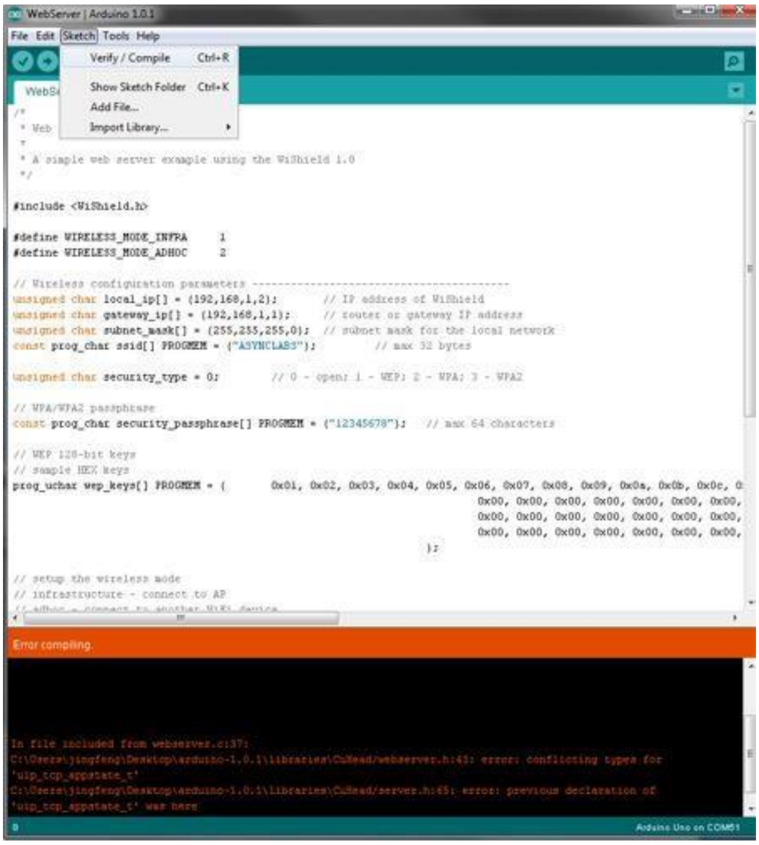


One example

45


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

Click Sketch-> Verify/Compile:



4.3 : CODE

```
#include <LiquidCrystal.h>

#include <stdio.h>

#include <SoftwareSerial.h>

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

// defines pins numbers

//New GPS GY-GPS6MV2


int SMS = 2;

int IR = 3;

int WL = 4;

int relay = 5;

int buzzer    = 7;

void beep()

{

  digitalWrite(buzzer, LOW);delay(1500);digitalWrite(buzzer, HIGH);delay(200);

}

char pwds='f';

int dist1,dist2,dist3,sts1=0;

long duration;

int distanceCm, distanceInch;
```

```
int memsx=0,memsy=0;
unsigned char rcv,count,gchr='x',gchr1='x',robos='s';
char pastnumber[31];
char pastnumber1[11],pastnumber2[11],pastnumber3[11];
 char gpsval[50];
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 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[8],finallong[9];
char keypads='x';
int ii=0;
float tempc=0,weight=0;
float vout=0;
```

```
int sti=0;
String inputString = "";         // a string to hold incoming data
boolean stringComplete = false;  // whether the string is complete
char cntt=0;
char pd=0,pwd[5];
void okcheck()
{
  unsigned char rcr;
  do{
      rcr = Serial.read();
    }while(rcr != 'K');
}
void send_link()
 {
    Serial.write("AT+CMGS=\"");
    Serial.write(pastnumber);
    Serial.write("\"\r\n");  delay(2500);
     Serial.write(0x1A);delay(4000);delay(4000);
 }
```

```
void setup()
{
   Serial.begin(115200);serialEvent();
   pinMode(SMS, INPUT);
   pinMode(IR, INPUT);
   pinMode(WL, INPUT);
   pinMode(relay, OUTPUT);
   pinMode(buzzer, OUTPUT);

digitalWrite(buzzer, HIGH);
digitalWrite(relay, HIGH);
  lcd.begin(16, 2);lcd.cursor();
  lcd.print("IOT based Smart");
  lcd.setCursor(0,1);
  lcd.print("Agriculture System");
  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();
 lcd.setCursor(0,0);
 lcd.print("SM:");
 lcd.setCursor(7,0);
 lcd.print("IR:");
 lcd.setCursor(0,1);
 lcd.print("WL:");

serialEvent();
```

```
}


int voltage=0;

int current=0;


void loop()

{

if(digitalRead(SMS) == LOW)

 {

    lcd.setCursor(4, 0);

    lcd.print(" ON");

    beep();

digitalWrite(relay, HIGH);


    Serial.write("AT+CIPSEND=0,22\r\n");delay(2000);

    Serial.write(" Water full Motor Off ");

   delay(3000);

 }

     if(digitalRead(SMS) == HIGH)

{

    lcd.setCursor(4, 0);

    lcd.print("OFF");
```

```
digitalWrite(relay, LOW);
 }


if(digitalRead(IR) == LOW)
 {
   lcd.setCursor(10, 0);
   lcd.print(" ON");
   beep();


   Serial.write("AT+CIPSEND=0,17\r\n");delay(2000);
   Serial.write(" Animal Detected ");
  delay(3000);
 }
    if(digitalRead(IR) == HIGH)
{
   lcd.setCursor(10, 0);
   lcd.print("OFF");
 }


if(digitalRead(WL) == LOW)
 {
   lcd.setCursor(3, 1);
```

```
    lcd.print(" ON");
    beep();

    Serial.write("AT+CIPSEND=0,18\r\n");delay(2000);
    Serial.write(" Water level full ");
   delay(3000);
 }
    if(digitalRead(WL) == HIGH)
{
    lcd.setCursor(3, 1);
    lcd.print("OFF");
 }

}

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

         char inChar = (char)Serial.read();
           //sti++;
```

```
        //inputString += inChar;
        if(inChar == '*')
          {sti=1;
            inputString += inChar;
           //  stringComplete = true;
           // gchr = inputString[sti-1]
          }
        if(sti == 1)
          {
              inputString += inChar;
          }
        if(inChar == '#')
          {sti=0;
            stringComplete = true;
          }
      }
}

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 == '*')
        {
         return 0;
        }
      if (inChar != '*')
        {
         result[i] = inChar;
         i++;
        }
    }
  }
}
void gpsEvent()
{
  gpsString="";
  while(1)
  {
```

```
{
 string str[]
    i++;
    if (i < 7)
    {
     if(gpsString[i-1] != test[i-1])         //check for right string
     {
       i=0;
       gpsString="";
     }
    }
//    if(inChar=='\r')
   {
    if(i>60)
    {
      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 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')

  {

  for(ii=0;ii<9;ii++)

    {

     msg1[ii] = gpsString[19+ii];

   for(ii=0;ii<10;ii++)

    {

     msg2[ii] = gpsString[32+ii];

    }



      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;


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;

    flong[0] = c;
    flong[1] = e;
    flong[2] = g;
   flong[3] = h;
  }
```

```
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(pastnumber1);
      pastnumber1[10]='\0';



   Serial.write("AT+CMGS=\"");
   Serial.write(pastnumber1);
```

```
   Serial.write("\"\r\n"); delay(2500);

   Serial.write("Reg-1\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(h);

                                  }
```

Chapter-5

Results

Generally, a moisture/temperature range which is already specified in datasheet of module, and whenever the actual values are out of this range, the microcontroller automatically turn ON the water pump, which is mounted on at output pins. The microcontroller also has solenoid valve attached to it to make sure that the pipes are actually watering the fields uniformly so that no area gets clogged or is left too dry. The entire system can be monitored by the end-user through a TELNET and Telegram application. Smart irrigation system makes it possible for farmers to monitor and irrigate their fields remotely, without any hassles.

Iot based smart agriculture system using TELNET which is capable of automating the irrigation process by analyzing the moisture of soil and the climate condition that is temperature and humidity. Also the data of sensors will be displayed in graphical form on TELNET cloud page as well as in text format in TELNET or thing speak application. When power supply is on then microcontroller checks the soil moisture content, temperature and humidity. If the moisture content is not up to the threshold level then it makes the motor to get on automatically and turns off automatically if reaches the threshold level and according valve movement on the pipe When the weather condition is such that it is raining then the microcontroller puts off the motor till then raining. After the raining it checks for threshold value set in system and makes the necessary action. All the data from the sensors and water is graphically shown in the TELNET iot cloud page which is used for monitoring. We can see the sensor data on telegram application in text form on the android mobile phone. Advantages of these system, it is a cost effective irrigation controller, increase efficiency and decrease wastage, easy to monitor, reduces man cost, reduced runoff water and nutrients.

We can develop the proposed system according to system architecture diagram. Soil moisture sensor connected with Arduino was dipped in soil and we get resultant valve on TELNET channel as you in fig. we can also see temperature data and humidity data on TELNET channel using LM35 attached to Arduino as you can see in fig

The hardware is interfaced with all the sensors in the board. The hardware components include the microcontroller, a water pump, relay,12 V battery, Wi-fi sensor and the soil moisture sensor is interfaced and power supply has provided. The system has been tested for watering a plant in a garden. In the field section, sensors are deployed in the field like soil moisture. The data collected from these sensors are sent to the Database via the android application. In control section, the system is turned on using the application, this is done using the on or off buttons in the application. Also, this system is turned on automatically when the moisture of the soil is low, the pump is turned on and depending on the moisture content. The application has a future feature of taking the time from the user and irrigates the field when the time comes. In manual mode, there is a manual switch in the field to make sure that if the system fails, one can turn off the water supply manually.

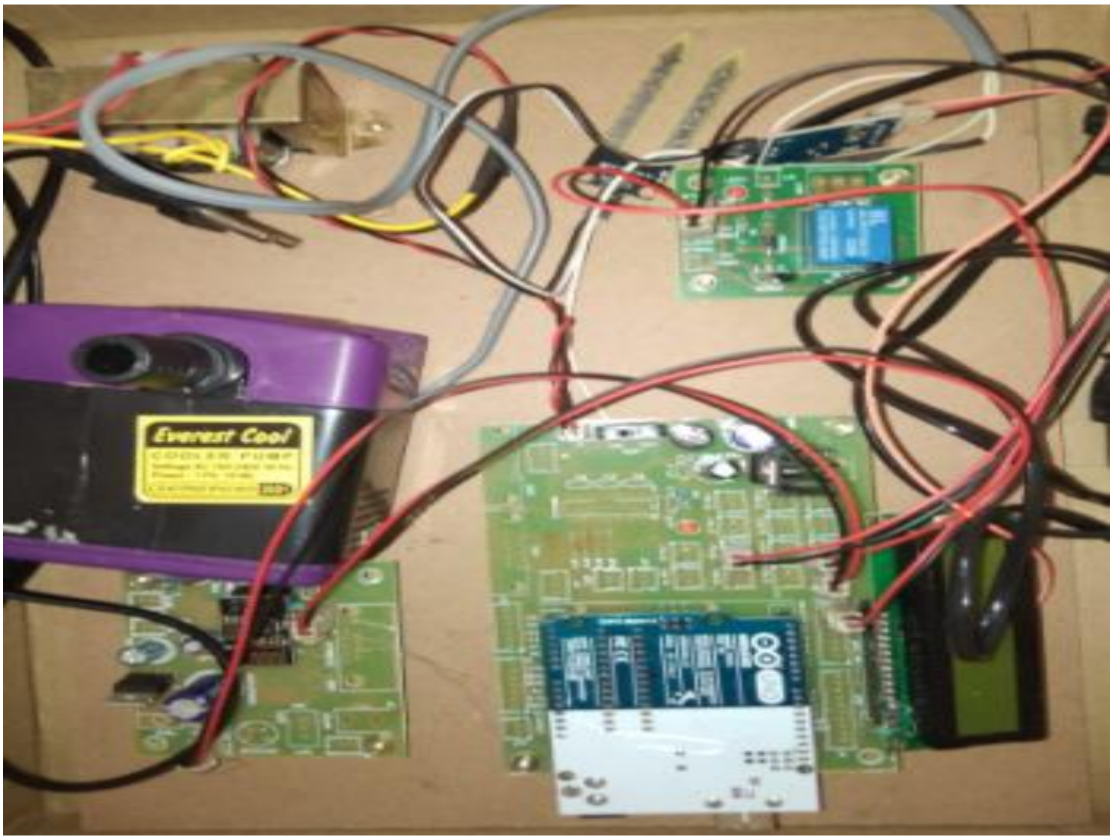


Fig.5.1. Hardware kit.

Fig.5.2. IOT irrigation display.

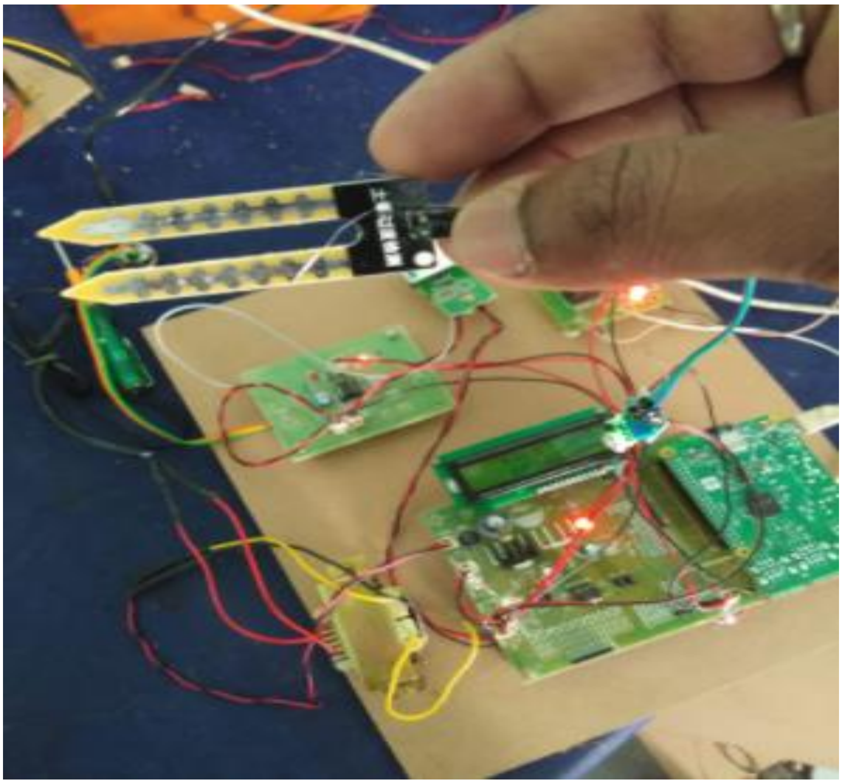


Fig.5.3. Soil moisture sensor.

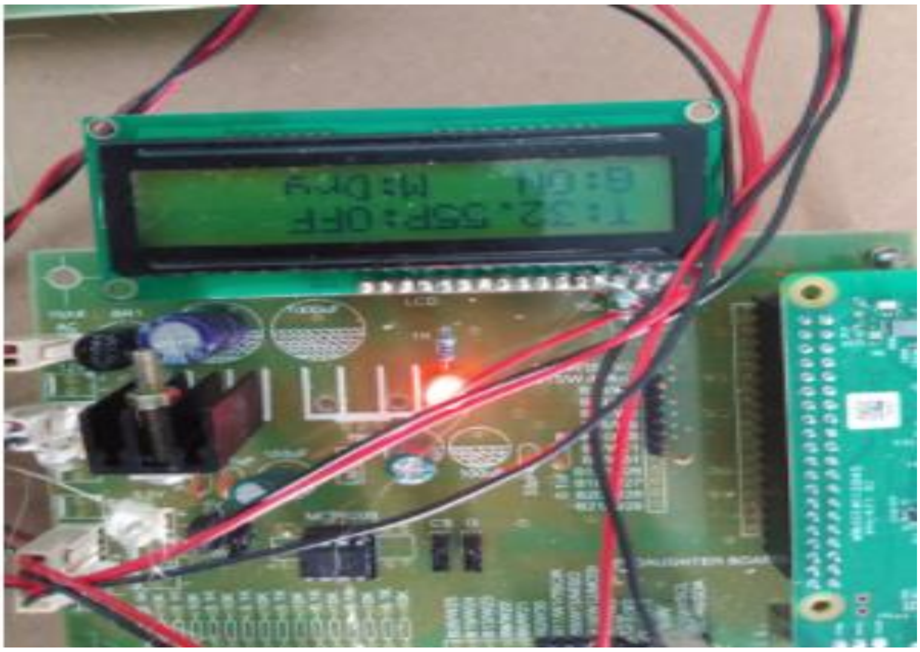


Fig.5.4. Input Parameters display in LCD.

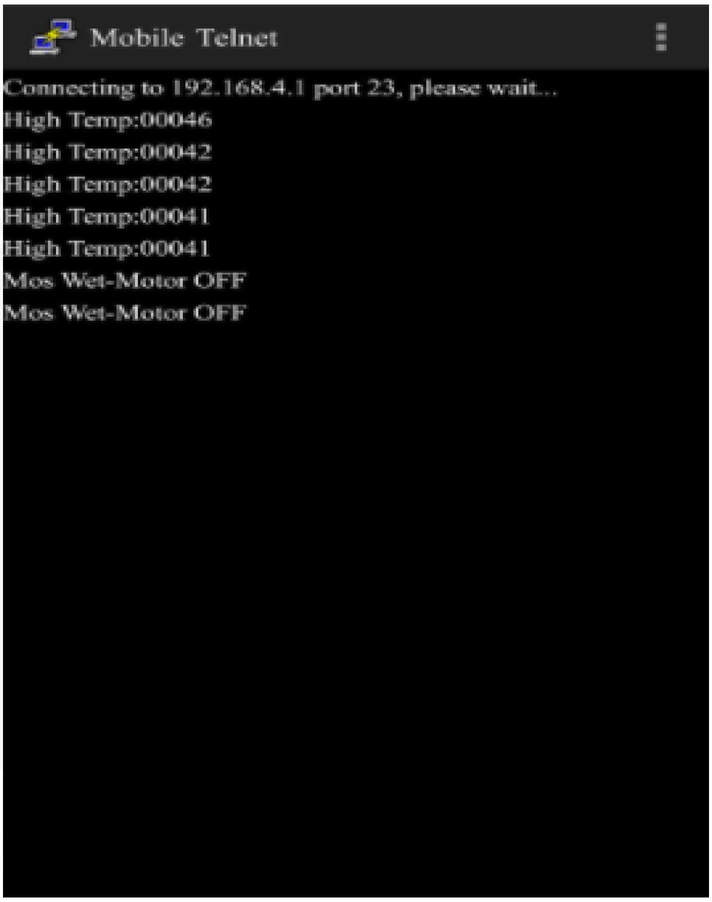


Fig.5.5. OUTPUT results

Chapter-6

Conclusion and Future Scope

6.1 Conclusion:

Internet of Things will help to enhance smart farming. Using IoT we can predict the soil moisture level and humidity. Irrigation system can be monitored and controlled by IoT technology. The crop damage using predators is reduced. IoT works in different domains of farming to improve time efficiency, water management, crop monitoring, soil management, control of insecticides and pesticides. It also minimizes human efforts, simplifies techniques of farming and helps to gain smart farming. Along with these features smart farming can help to grow the market for farmer with single touch and minimum efforts.

6.2 Future Scope:

For further enhancement, this system is used for large acres of land. Also, the system can be integrated to check the soil nutrient and crop growth in each soil. Also, the system can be further improved by adding machine learning algorithms, which are able to learn and understand the requirements of the crop, this would help the field be an automatic system.

References

[1]. Rajalakshmi.P, Mrs.S.Devi Mahalakshmi "IOT Based Crop-Field Monitoring And Irrigation Automation" 10th International conference on Intelligent systems and control (ISCO), 7-8 Jan 2016 published in IEEE Xplore Nov 2016.

[2]. Prof. K. A. Patil And Prof N. R. Kale proposes "A Model For Smart Agriculture Using IOT" 2016 International Conference on Global Trends in signal Processing, Information Computing And Communication.

[3]. Dr.N.Suma, Sandra Rhea Samson, S. Saranya, G. Shanmugapriya, R. Subhashri „IOT Based Smart Agriculture Monitoring System‟ 2017 International Journal on Recent and Innovation Trends in Computing and Communication.

[4]. Mahammad shareef Mekala, Dr.P.Viswanathan „A Survey: Smart agriculture IoT with cloud Computing ‟ 978-1-5386-1716-8/17/$31.00 ©2017 IEEE

[5]. Prathibha S R1, Anupama Hongal 2, Jyothi M P3‟ IOT BASED MONITORING SYSTEM IN SMART AGRICULTURE‟ 2017 International Conference on Recent Advances in Electronics and Communication Technology

[6]. Ibrahim Mat, Mohamed Rawidean Mohd Kassim, Ahmad Nizar Harun, Ismail Mat Yusoff "IOT in Precision Agriculture Applications Using Wireless Moisture Sensor Network" 2016 IEEE Conference on Open Systems (ICOS), October 10-12- 2016, Langkaw, Malaysia.

[7]. Zhaochan Li, JinlongWang, Russell Higgs, LiZhou WenbinYuan4 "Design of an Intelligent Management System for Agricultural Green houses based on the Internet of Things" IEEE International Conference on Embedded and Ubiquitous Computing (EUC) 2017.