1.1 INTRODUCTION:

Generally, the current irrigation systems are manually operated. India is the country of village and agriculture plays an important role in the development of the country. In our country, agriculture depends on the monsoons which have an insufficient source of water. So, irrigation is used in the agriculture field. In Irrigation system, depending upon the soil type, which is how much water is provided to plant. In agriculture, two things are very important, first to get information about the fertility of the soil and second to measure moisture content in the soil. Nowadays, for irrigation, the different techniques are available in the country which is used to reduce the dependency of rain such as water pump etc. And mostly this technique is driven by electrical power and on/off schedule. In this technique, water level indicator placed in the water reservoir and soil moisture sensors are placed root zone of the plant and near the module and gateway unit handles the sensor information and transmit data to the controller which in turns in the control the flow of water through the valves.

The Internet of things (IoT) [1] emerges as the naturals choice for irrigation monitoring and controller system using IoT applications, even though the integration of different technologies required for making it work seamlessly in practice is still not fully accomplished. Internet of Things (IoT) is widely used in connecting devices and collecting data information. IoT(Internet of Things) is used with IoT frameworks to handle and interact with data and information about sensor which is sense the data. Advanced tools and technology can be used to increase farm yield. Developing IoT technologies can help to collect a large amount of ecological and crop recital data. "IoT encompasses many new intelligent concepts for using in the near future such as smart home, smart city, smart transportation, and smart farming" [2]. The technique can be used for the application of an accurate amount of fertilizer, water, pesticide etc. to enhance productivity and excellence. The emergence of IoT is a phenomenon that owes to the conjunction of several factors such as inexpensive devices, low-power wireless technologies, availability of cloud data centers for storage and processing, management frameworks for dealing with unstructured data from social networks, high-performance computing resources in commodity platforms, and computational intelligence algorithms to deal with this monumental amount of data.

This paper presents an irrigation monitoring and controller system. The system uses the sensor network to monitor the environmental conditions such as soil moisture content and water level of agriculture land for controlling the irrigation. The system has an automatic and manual mode. The real-time sensed data is stored on the android app. for decision making and controlling actions. The user can monitor the controlling actions taken at the farm as well as control the irrigation via an android app on the farmer's mobile phone.

1.1 MOTIVATION:

For continuously increasing demand and decrease in supply of food necessities, it's important to rapid improvement in production of food technology. Agriculture Is only the source to provide this. This is the important factor in human societies to growing and dynamic demand in food production. Agriculture plays the important role in the economy and development, like India. Due to lack of water and scarcity of land water result the decreasing volume of water on earth, the farmer use irrigation. Irrigation may be defined as the science of artificial application of water to the land or soil that means depending on the soil type, plant is to be provided with water.

1.2 AREA OF UTILITY:

The primary focus of this project is to help the farmers and reduce their work. This module can be implemented in perennial plant irrigation land and gardening land.

2.1 Literature Survey

In irrigation field, soil moisture sensor, temperature sensors are placed in root of plant and microcontroller handles the sensor information and transmits data. One algorithm was developed to measure threshold values of temperature sensor and soil moisture sensor that was programmed into a microcontroller to control water quantity.[1]

A model of automatic irrigation system which is based on microcontroller and solar power was used only for source of power supply. Various sensor is placed in paddy field. Sensors sense water level continuously and give the information to farmer through cellular phone. Farmer controls the motor using cellular phone without going in paddy field. If the water level reaches at danger level, automatically motor will be off without conformation of farmer.[1]

This system proposes low cost moisture sensor-based data acquisition system required for automated irrigation system. The authors have developed an impedance-based moisture sensor. Sensors works on the change of impedance between two electrodes kept in soil [2].

This system is smart irrigation techniques using internet of things (IOT). In this system sensors are placed in the agriculture field, measures the soil moisture value, water level in the tank and well-water through mobile data communication network. The web servers use intelligent software to analyze the data and act according to the result obtained to perform desired action [4].

2.2 Related Work

In this paper, soil moisture sensor, temperature and humidity sensors placed in the root zone of the plant and transmit data to android application. The threshold value of a soil moisture sensor that was programmed into a microcontroller to control water quantity. soil moisture values are displayed on the android application.

This system developed an irrigation control and monitoring system for the farmer on the basis of the internet of things which is used wi-fi modules, Arduino Uno and Android app. This system continuously monitors the parameters moisture of the soil. An algorithm was used with threshold values of soil moisture to be maintained continuously. The system starts or stops irrigation based on the moisture content of the soil. This system proposes a low-cost moisture sensor-based data acquisition system required for the automated irrigation system. The authors have developed an impedance-based moisture sensor. Sensors works on the change of impedance between two electrodes kept in soil [2].

This paper on "IoT based Irrigation Control and Monitoring System on Sensing Soil Moisture Content" is intended to create an automated irrigation mechanism which turns the pumping motor ON and OFF on detecting the state or condition of being slightly wet content of the earth. In this paper, only soil moisture value is considered but the proposed project provided an extension to this existed project by adding temperature and humidity values. [3]

This proposed paper is IoT based irrigation control and monitoring system developed for the agricultural plantation, which is placed at the remote location and required water provides for plantation when the moisture of land of the soil goes below the set-point value. But in this, we did not aware about the soil moisture level so to overcome this drawback proposed system included with extra feature soil moisture value and temperature value which displayed on the farmer mobile application [4].

The system supports water management decision, used for monitoring the whole system using Wi-fi module. The system continuously monitors the moisture - the level of soil and provide an accurate amount of water required for plant or crop. The system checks the moisture-level of soil to retain the nutrient composition of the soil managed for growth of plant [5].

Plenty of research work has been done to improve the performance of agriculture field. In [1] the system uses Arduino technology to control watering and roofing of the green house. It uses statistical data acquired from sensors (like temperature, humidity, moisture and light intensity sensors) compared with the weather forecast for decision making. Kalman filter is used to eliminate noise from the sensors.

3.1 Proposed Work

The proposed irrigation monitoring and controller system using IoT (Internet of Thing) consists of the Arduino Uno, water pump, and Soil-moisture Sensor and Wi-fi Modules. Smartphones module is used for communication. In the proposed work, crops or plants are considered along with their water requirement at different stages. The crops or plants are irrigated with respect to the water requirements at different stages of their growth.

Fig 1 shows the architectural design of the project. The smartphone is connected to Arduino Uno through wi-fi modules. The motor is controlled by the smartphone by the values moisture level of soil which is turn ON and OFF.

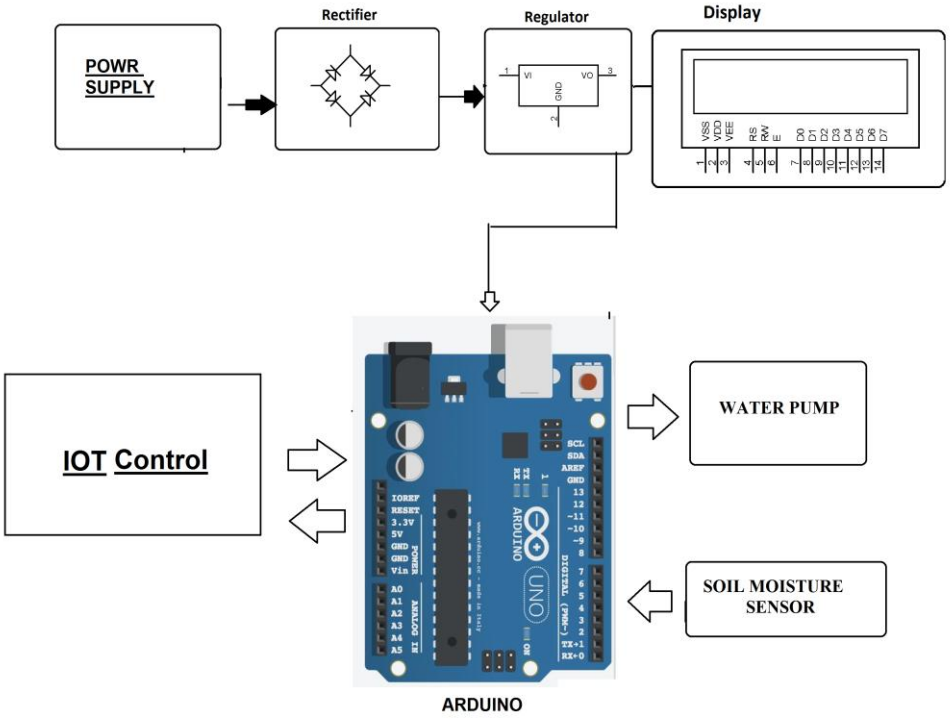


Fig.1 Proposed System Design

They continuously monitor the field and send it to the web server using NRF24LO1 transmitter and receiver and Ethernet connection at receiver ends. The sensor data are stored in database. The web application is designed in such a way to analyze the data received and to check with the threshold values of moisture. The decision making is done at server to automate irrigation. If soil moisture is less than the threshold value the motor is switched ON and if the soil moisture exceeds the threshold value the motor is switched Off.

4.1 Definition

The Internet of things (IoT) is the extension of internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

The definition of the Internet of things has evolved due to convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded system. Traditional fields of embedded system wireless sensor networks, control system, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such smartphones and smart packets.

The Internet of Things (IoT) is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure. The “Internet of Things” connects devices and vehicles using electronic sensors and the Internet.

4.2 Introduction

The Internet of Things (IoT) is the network of physical objects devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data. The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer based systems, and resulting in improved efficiency, accuracy and economic benefit, when IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.

So, Internet of Things or IoT is an architecture that comprises specialized hardware boards, Software systems, web APIs, protocols which together creates a seamless environment which allows smart embedded devices to be connected to internet such that sensory data can be accessed and control system can be triggered over internet.

Also, devices could be connected to internet using various means like Wi-Fi, Ethernet and so on. Furthermore, devices may not need to be connected to internet independently. Rather a cluster of devices could be created (for example a sensor network) and the base station or the cluster head could be connected to internet. This leads to more abstract architecture for communication protocols which ranges from high level to low level.

Most interestingly, these devices must be uniquely discovered. For unique discovery of the devices in a Network, they need to have unique IP address. IoT devices essentially have IPv6 addressing scheme. All these devices have either fixed or Subnet masked IP addresses of type v6. Unique IP addresses makes IoT devices discoverable in the internet as independent node. This is the most important concept to have in mind to understand IoT.

Following figure.4.1 explain what IoT is all about.

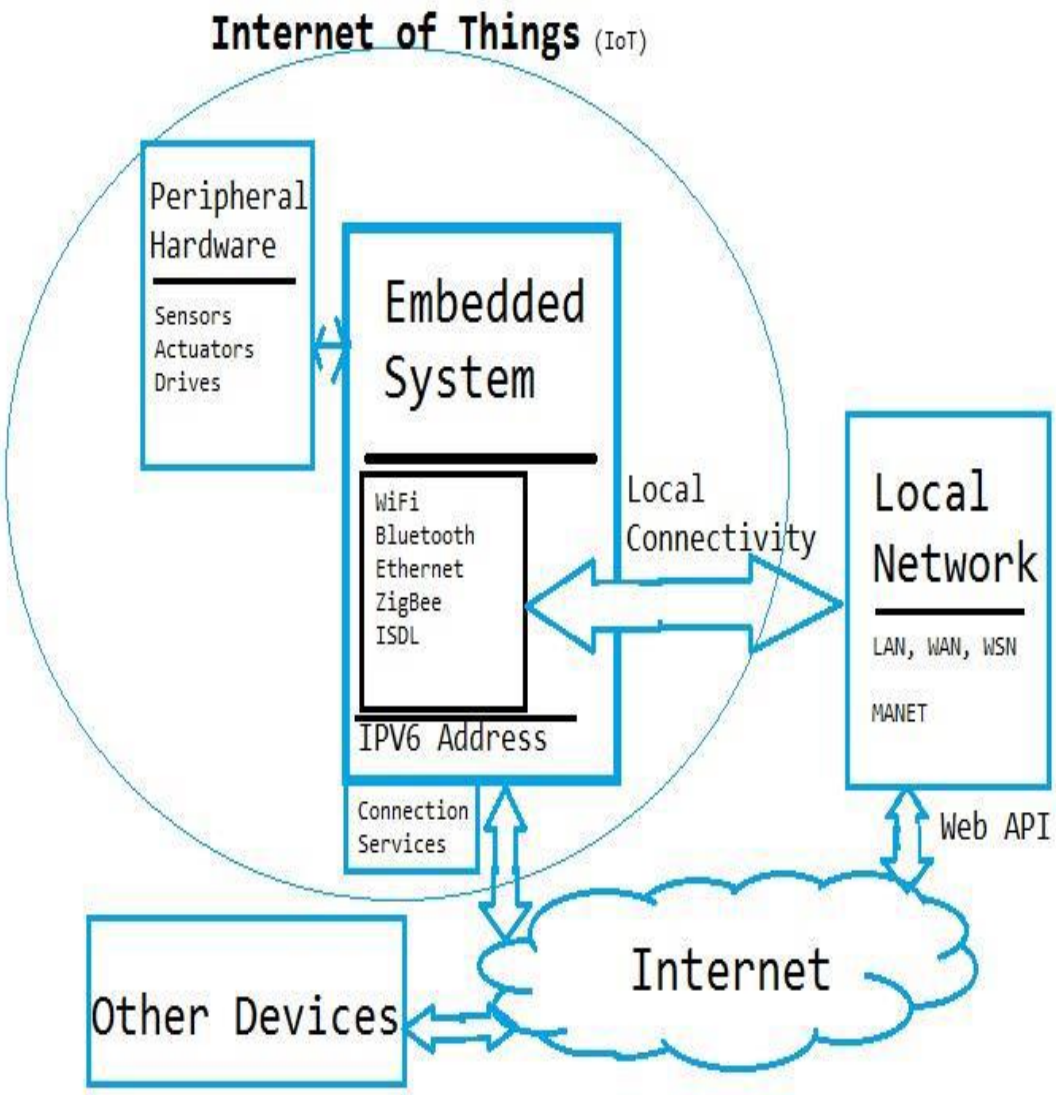


Figure 4.1 Internet of Things (IoT) Basic Architecture

4.3 What Devices Makes it to IoT

Since IoT are essentially embedded systems and smart objects connected to internet with unique IP address which can be discovered and communicated over internet. We have also seen that the IoT devices may have external peripheral like Actuators and Sensors.

4.3.1 Are Mobile Phones are IoT Devices

One of the most common in day to day life are mobile phones. Mobile phone is essentially an embedded system with a processor at the core having display and keypad. They support wide variety of sensors like ambient light Sensors, Accelerometer, Gyroscope and so on.

They are connected to internet. Mobile phones get IP addresses, can access internet. In other words, it virtually fits every description of IoT. So, can we call mobile phones IoT devices? This doubt was clarified at a keynote event during Sept 2011's Mobile World Congress in Barcelona by Qualcomm Chairman and CEO Dr. Paul Jacobs.

Paul Jacobs talked about how mobile technology could be used to **connect non-phone, non-tablet devices called IoT devices** and objects to the Internet. In this future where everything is Web-connected, **mobile phones will serve as the hub, or the remote control**, for Internet of Things.

So IoT is internet connectivity of smart objects and embedded system other than mobile phones which can be connected with external hardware and Mobiles, Tablets, Laptops and PCs are remote control/access center of IoT.

4.3.2 IoT Devices

The most common and popular technologies in IoT will give an overview device. The IoT devices into two broad categories: The wearable ones and Microcontroller /Microprocessor driven embedded IoT devices. Some of the embedded devices like Arduino Lilly pad are Monique and it can further utilize them to make wearable solution. But wearable includes hardware which are pretty standard and IoT has only software scope for the developer. Some peripheral hardware is which might require are in IoT hardware in embedded level. Apps can be used with popular wearable platforms, Embedded IoT platform may include broader technologies like Raspberry Pi, Arduino or Galileo, etc.

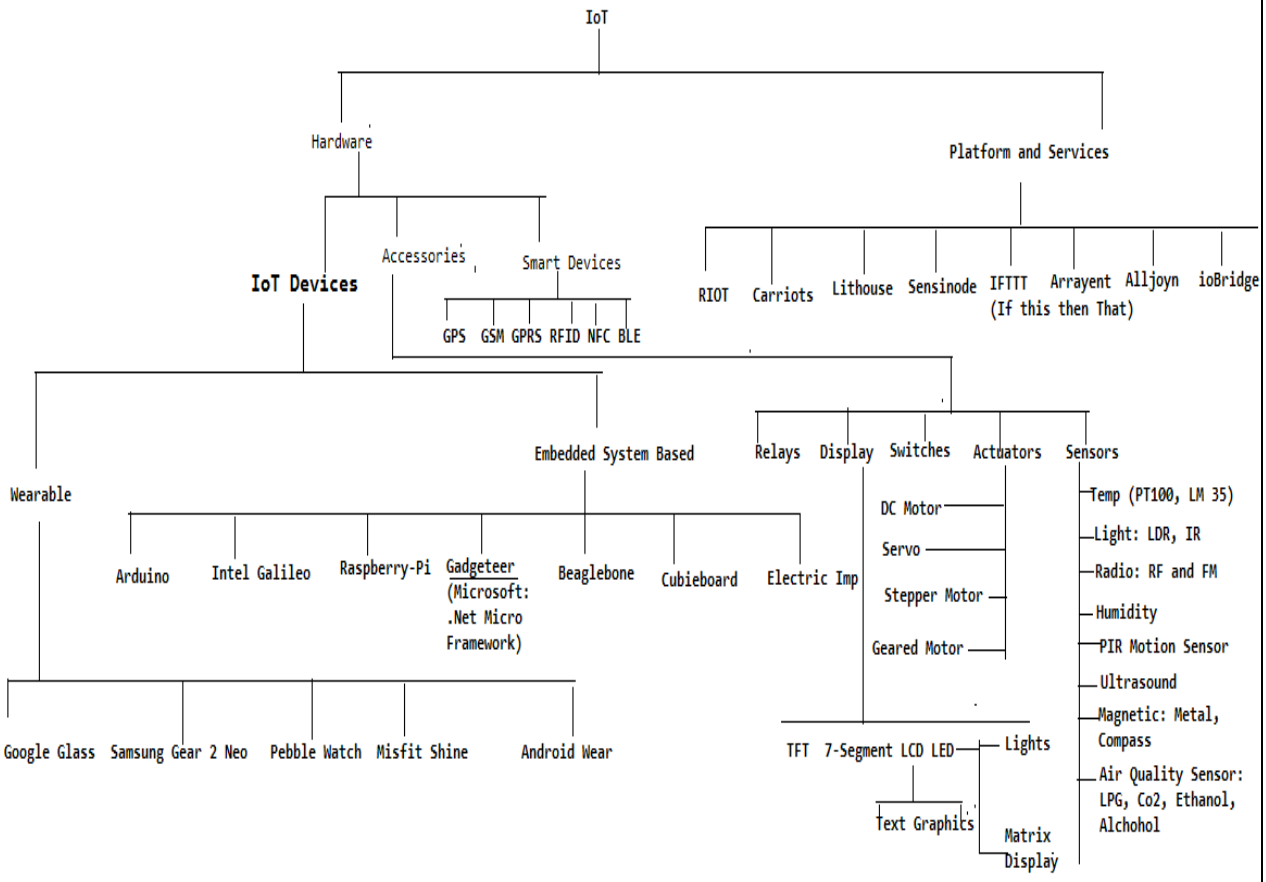


Figure 4.2 Common IoT Devices and Technologies

4.3.3 IoT Platforms

IoT development can be divided into two parallel technologies: Wearable and Embedded. Developers can build apps for custom Wearable devices like Pebble, Samsung Gear or can often create their own platform using Embedded solution and then can develop app for that platform.

IoT platform is an essential component of a huge IoT ecosystem that supports and connects all components within the system. It helps to facilitate device management, handle hardware/software communication protocols, collect/analyze data, enhance data flow and functionality of smart applications.

The overall IoT system includes:

- hardware (devices and sensors)
- connectivity through a router, gateway, wi-fi, satellite, ethernet, etc.
- software
- user interface

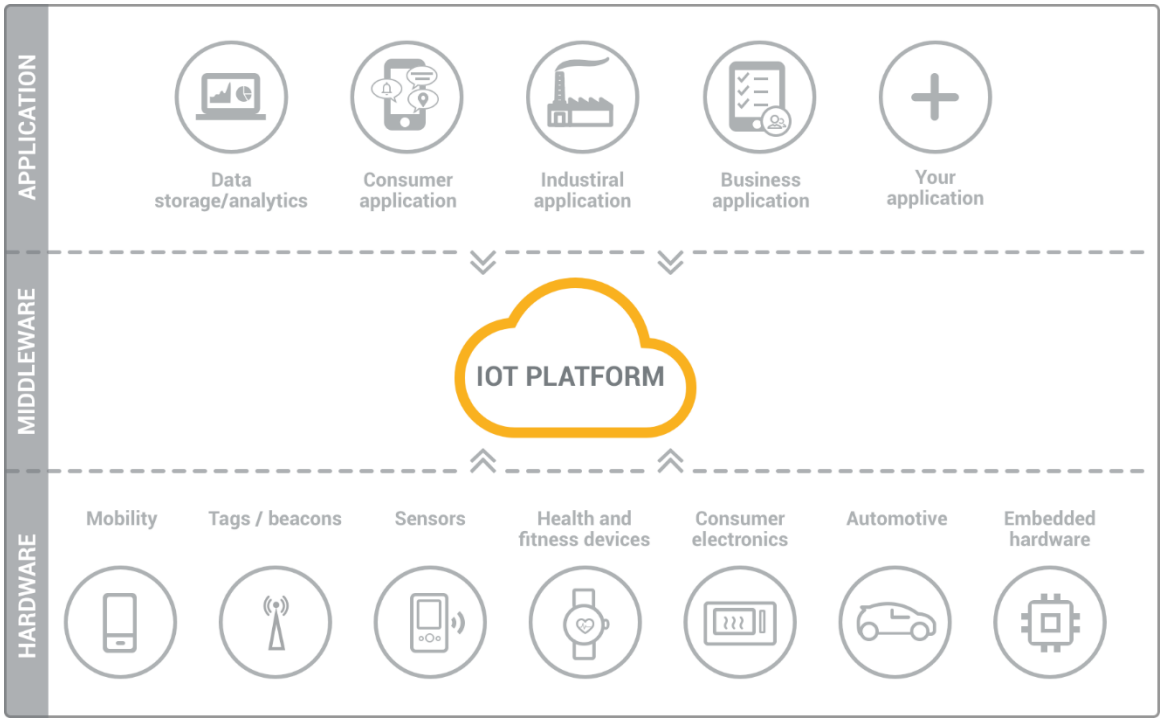


Figure 4.3 IoT platform

4.3.3.1 Wearable Platform

Tizen is fast becoming one of the most popular platforms for Mobile and wearable devices. Tizen SDK comes ported with wearable emulator which makes it easier to develop wearable solutions for Tizen platform.

As figure 4.2 suggests, a large Android Wear device are now being made and marketed. Smart watches are getting popular by every day. Android Wear apps can be developed and tested in Eclipse.

This Android Developer Guide helps you in setting up Android Wear development environment in Eclipse. Salesforce is another platform which is coming up with awesome development environment, APIs in wearable technologies. Their solution is extended from Pebble to Google glass.

4.3.3.2 Embedded Platform

Arduino is probably the best starting point for embedded based IoT. Basic Arduino boards don't come with Ethernet shield or Wi-Fi shield and for Arduino to be able to work as IoT device, their need to select Arduino with Ethernet shield or Wi-Fi shield. Arduino Yun on the other hand is a board that comes ported with Ethernet shield.

Raspberry Pi is probably one of the best things to happen in DIY IoT. A wide range of Data driven applications like Home Automation Server to Home Multimedia server, File Server can be developed with Pi. PI like Arduino has general purpose IO pins. But seamless working with sensors is bit tedious in Pi.

Another efficient IoT board is **Intel Edition** which has integrated BLE, Wi-Fi among host of other features. It supports wide range of Industry standard hardware (over 30) through 70-pin interface.

Intel Galileo is another good offering by Intel which supports the same shielding that of Arduino Uno. So, it can be said to be first Intel powered device which is Arduino compatible. It has among other thing a USB host controller like Raspberry Pi which makes this an attractive hardware. Galileo also has ethernet shield in built.

4.3.3.3 Cloud Platform

IoT really can bring several services (like online payment gateway), several hardware platforms (like embedded board of the vending machine) and smart objects and data like NFC, GPS into a seamless environment. Now it can integrate online payment into beverage vending machine, if one is using location service for beverage machine, then utilizing the location and payment service can be done.

One can get the data of a medical diagnosis like ECG (acquired through another embedded board pertaining to medical electronics) into cloud such that several doctors can view it and form a comprehensive opinion about the patient's state.

Well, in fact all of them are possible. A little understanding of web and software design would take your mind towards cloud. Just like Web of Machines, in a Machine to Machine (M2M) or Machine to Objects (M2O) or any similar communication several modules will be common and several modules demands data to be available for sharing. Cloud APIs comes in handy in this regard.

For instance, to make a device discoverable in web, then assign a fixed IP address, maintain a router and follow several networking skills.

Yaler is a great example of what services and cloud can bring to table. This provides connection as a service such that your device is easily discoverable and communicable over the web without much hassle and take care of underneath security.

Axeda Provides infrastructure for M2M architecture.

OpenIoT is an open source IoT platform that provides out of other services a unique Sensing as a Service.

Google has already integrated location services with its cloud. Location extracted from your devices are silently put in your status updates in Facebook and twitter and are also used for more personalized searches.

So, cloud APIs has a great potential in IoT in all levels of architecture starting from firmware to hardware to more top-level architecture.

4.4 Implementation using IoT

This project uses concept of IoT for monitoring and controlling the system using a public server called **MQTT** server. It uses an android app called **MyMQTT**. In this app, one has to subscribe a topic and publish a message of specific function. The server will call-back to perform the function.

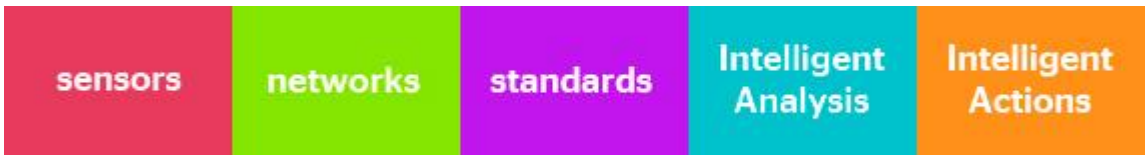


Figure 4.4 component of IoT implementation

4.4.1 MQTT

MQTT stands for Message Queue Telemetry Transport. It is a publish /subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium.

4.4.2 MQTT Architecture

MQTT has a client/server model, where every sensor is a client and connects to a server, known as a broker, over TCP.

MQTT is message oriented. Every message is a discrete chunk of data, opaque to the broker.

Every message is published to an address, known as a topic. Clients may subscribe to multiple topics. Every client subscribed to a topic receives every message published to the topic.

MQTT defines methods (sometimes referred to as *verbs*) to indicate the desired action to be performed on the identified resource. What this resource represents, whether pre-existing data or data that is generated dynamically, depends on the implementation of the server.

Often, the resource corresponds to a file or the output of an executable residing on the server.

Connect: Waits for a connection to be established with the server.

Disconnect: Waits for the MQTT client to finish any work it must do, and for the TCP/IP session to disconnect.

Subscribe: Waits for completion of the Subscribe or Unsubscribe method.

Unsubscribe: Requests the server unsubscribe the client from one or more topics.

Publish: Returns immediately to the application thread after passing the request to the MQTT client

4.4.3 MQTT Ports

The server listens on the following ports:

- 1883: MQTT, unencrypted
- 8883: MQTT, encrypted
- 8884: MQTT, encrypted, client certificate required
- 8080: MQTT over WebSocket's, unencrypted
- 8081: MQTT over WebSocket's, encrypted

This project uses 1883 an unencrypted MQTT port.

4.4.4 MQTT Example

Imagine a simple network with three clients and a central broker. All three clients open TCP connections with the broker. Clients B and C subscribe to the topic temperature (Figure 4.5).

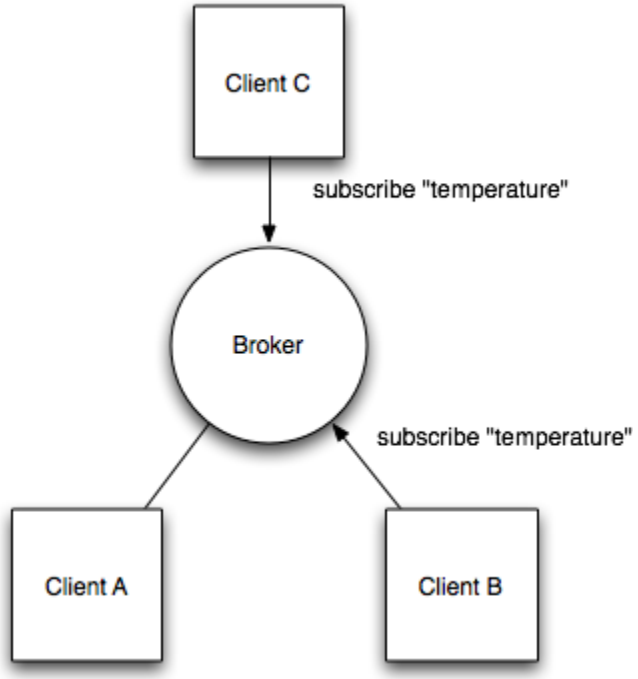


Figure 2.3 Client B and C Subscribing Topic temperature.

The publisher subscriber model allows MQTT clients to communicate one-to-one, one-to many and many-to-one.

5.1 Arduino

Arduino is an open source physical computing platform based on simple input/output board and a development environment that implements the Processing language (www.processing.org). Arduino can be used to develop standalone interactive objects or can be connected to software on your computer. The boards can be assembled by hand or purchased preassembled; the open source IDE (Integrated Development Environment) can be downloaded for free from www.arduino.cc.

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.

5.1.1 Introduction to Arduino Boards

Arduino is an architecture that combines Atmel microcontroller family with standard hardware into a board with inbuilt bootloader for plug and play embedded programming. Arduino Software comes with an IDE that helps writing, debugging and burning program into Arduino. The IDE also comes with a Serial Communication window through which can easily get the serial data from the board.

5.1.2 Arduino UNO

The Arduino Uno is a microcontroller board developed by Arduino.cc and based on the ATmega328P. Microcontrollers are widely used in embedded systems and make devices work according to our needs and requirements. Arduino Uno is a very valuable addition in the electronics that consists of USB interface, 14 digital I/O pins, 6 analogue pins, and Atmega328 microcontroller. It also supports serial communication using Tx and Rx pins. Each of the 14 digital pins can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. The Arduino Uno is a small single board computer which is been used to teach computer science. The Arduino Uno is been used as a computer where external memory can be used and it has four ports where any input devices can be connected. This project uses Arduino Uno for easy process and installation.

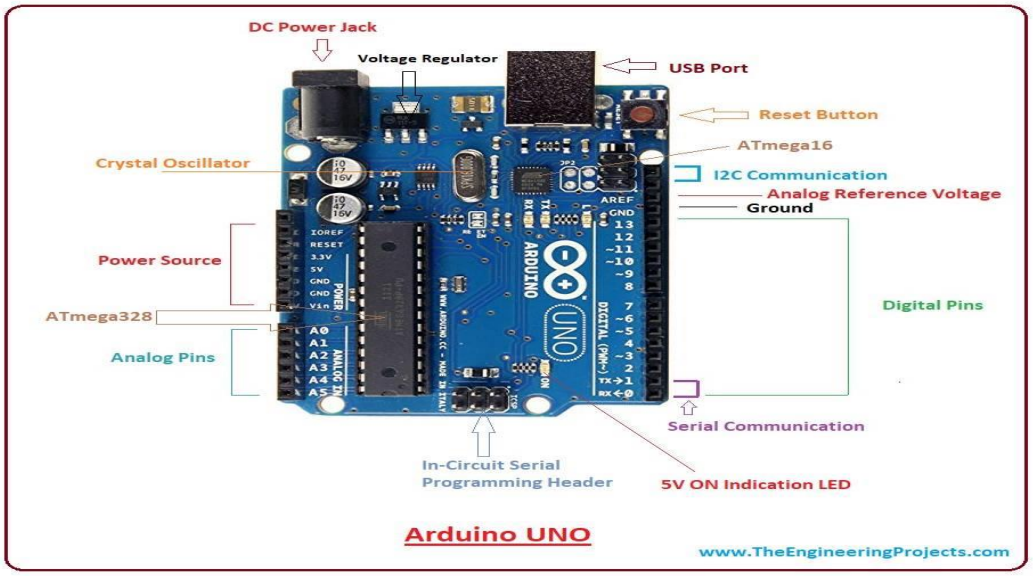


Fig 5.1 Arduino Uno

They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED driven by digital pin 13
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the
analogReference () function.

There are a couple of other pins on the board,

- AREF Reference voltage for the analog inputs. Used with analogReference ().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

5.1.3 Arduino UNO Technical Specifications

Microcontroller	ATmega 328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 5.1 Technical Specification

5.2 ESP8266

5.2.1 Introduction to ESP8266

The ESP8266 Arduino compatible module is a low-cost Wi-Fi chip with full TCP/IP capability, and the amazing thing is that this little board has an MCU (Micro Controller Unit) integrated which gives the possibility to control I/O digital pins via simple and almost pseudo-code like a programming language. ESP8266 is a complete and self-contained Wi-Fi network solution that can support and move (someone or something) from one place to another software applications, or through another application, processor uninstalls all Wi-Fi networking capabilities. Built-in cache memory will help improve system performance of software application and reduce memory requirements. Another situation is when wireless Internet access assumes the task of the Wi-Fi adapter, you can add it to any microcontroller-based design, and the connection is simple.

Another situation is when wireless Internet access assume the task of Wi-Fi adapter, you can add it to any microcontroller-based design, and the connection is simple, just by SPI / SDIO interface or central processor AHB bridge interface. Processing and storage capacity on ESP8266 powerful piece, it can be integrated via GPIO ports sensors and other applications specific equipment to achieve the lowest early in the development and operation of at least occupy system resources. The ESP8266 highly integrated chip, including antenna switch balun, power management converter, so with minimal external circuitry, and includes front-end module, including the entire solution designed to minimize the space occupied by PCB. The system is equipped with ESP8266 manifested leading features are: energy saving VoIP quickly switch between the sleep / wake patterns, with low-power operation adaptive radio bias, front-end signal processing functions, troubleshooting and radio systems coexist characteristics eliminate cellular / Bluetooth / DDR / LVDS / LCD interference.

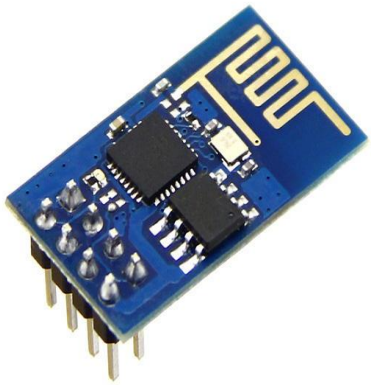
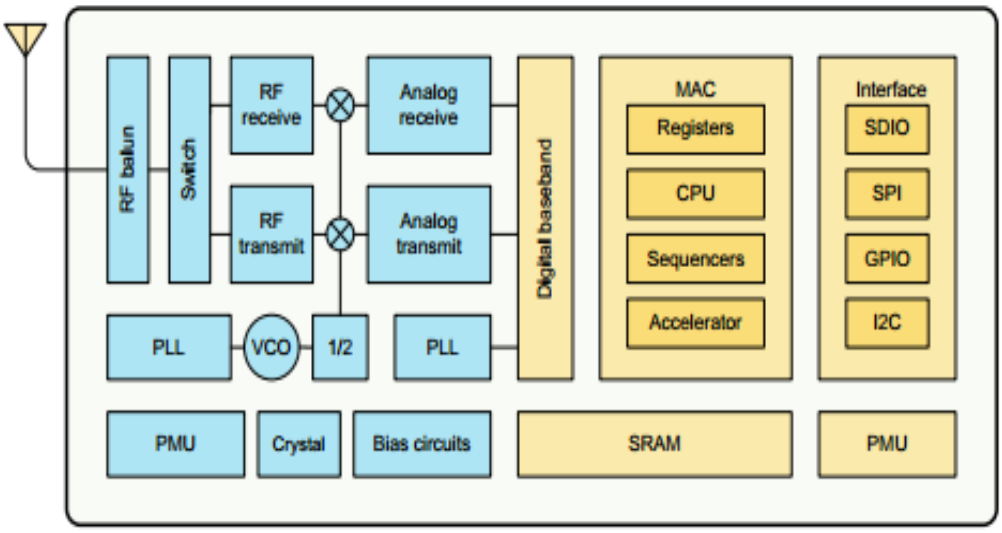


Fig. 5.2 wi-fi modules

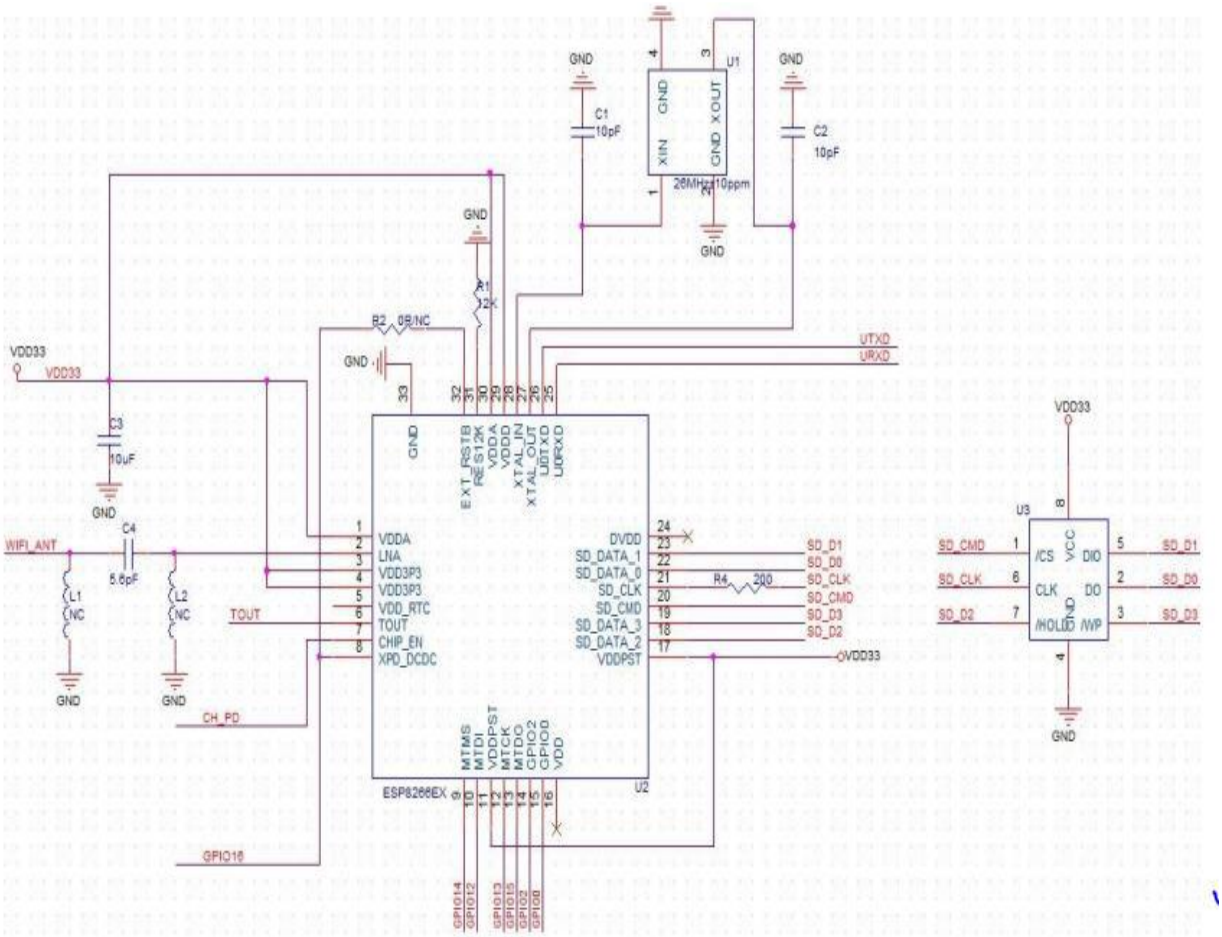
5.2.2 Block Diagram of ESP8266



5.2.3 Characteristics of ESP8266

- 802.11 b / g / n
- Wi-Fi Direct (P2P), soft-AP
- Built-in TCP / IP protocol stack
- Built-in TR switch, balun, LNA, power amplifier and matching network
- Built-in PLL, voltage regulator and power management components
- 802.11b mode + 19.5dBm output power
- Built-in temperature sensor
- Support antenna diversity
- off leakage current is less than 10uA
- Built-in low-power 32-bit CPU: can double as an application processor
- SDIO 2.0, SPI, UART
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU, A-MSDU aggregation and the 0.4 Within wake
- 2ms, connect and transfer data packets
- standby power consumption of less than 1.0mW (DTIM3)

5.2.4 Schematic Diagram of ESP8266-EX



5.2.5 ESP Modules

The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack and microcontroller capability produced by Shanghai-based Chinese manufacturer, **Espressif**. These were the first series of modules made by third-party manufacturer, **AI-Thinker** with the ESP8266 and remain the most widely available.

Board ID Pins Pitch LEDs Antenna Dimensions mm
ESP-01 8 .1“ Yes Etched-on PCB 14.3 x 24.8
ESP-02 8 .1” No None 14.2 x 14.2
ESP-03 14 2mm No Ceramic 17.3 x 12.1
ESP-04 14 2mm No None 14.7 x 12.1
ESP-05 5 .1“ No None 14.2 x 14.2
ESP-06 12+GND misc No None ?
ESP-07 16 2mm Yes Ceramic 20.0 x 16.0
ESP-08 14 2mm No None 17.0 x 16.0
ESP-09 12+GND misc No None 10.0 x 10.0
ESP-10 5 2mmm? No None 14.2 x 10.0
ESP-11 8 1.27mm No Ceramic 17.3 x 12.1
ESP-12 16 2mm Yes Etched-on PCB 24.0 x 16.0
ESP-12-E 22 2mm Yes Etched-on PCB 24.0 x 16.0
ESP-13 18 1.5mm - Etched-on PCB -
ESP-14 22 2mm 1 Etched-on PCB 24.3 x 16.2
WROOM-02 18 1.5mm No Etched on PCB 20.0 x 18.0
WT8266-S1 18 1.5mm 1 Etched on PCB 15.0 x 18.6

5.2.6 ESP8266 Applications

- Smart Power Plug
- Home Automation
- Industrial wireless control
- Baby Monitor
- Network Camera
- Wireless location-aware devices and positioning system signals

5.2.7 Explore ESP8266 Wi-Fi Module

The ESP8266 ESP12E Wi-Fi Module is more user friendly with the Explore ESP8266 Wi-Fi Module. It fits on a breadboard with all pins taken out. The module goes into programming mode with a single reset switch.

Features:

- Fits on a breadboard.
- Single button 'Reset' switch for programming. Uses MOSFET's to put the module in
- programming mode.
- All pins of ESP12E taken out.
- Separate serial pins breakout compatible with FTDI cable layout.
- On-board LM1117-3.3V regulator.
- Works with Arduino IDE for ESP8266.
- Programs can easily dump using USB to TTL converter.
- ESP8266 ESP12E features.

5.2.8 Schematic Diagram of Explore ESP8266 Wi-Fi Module

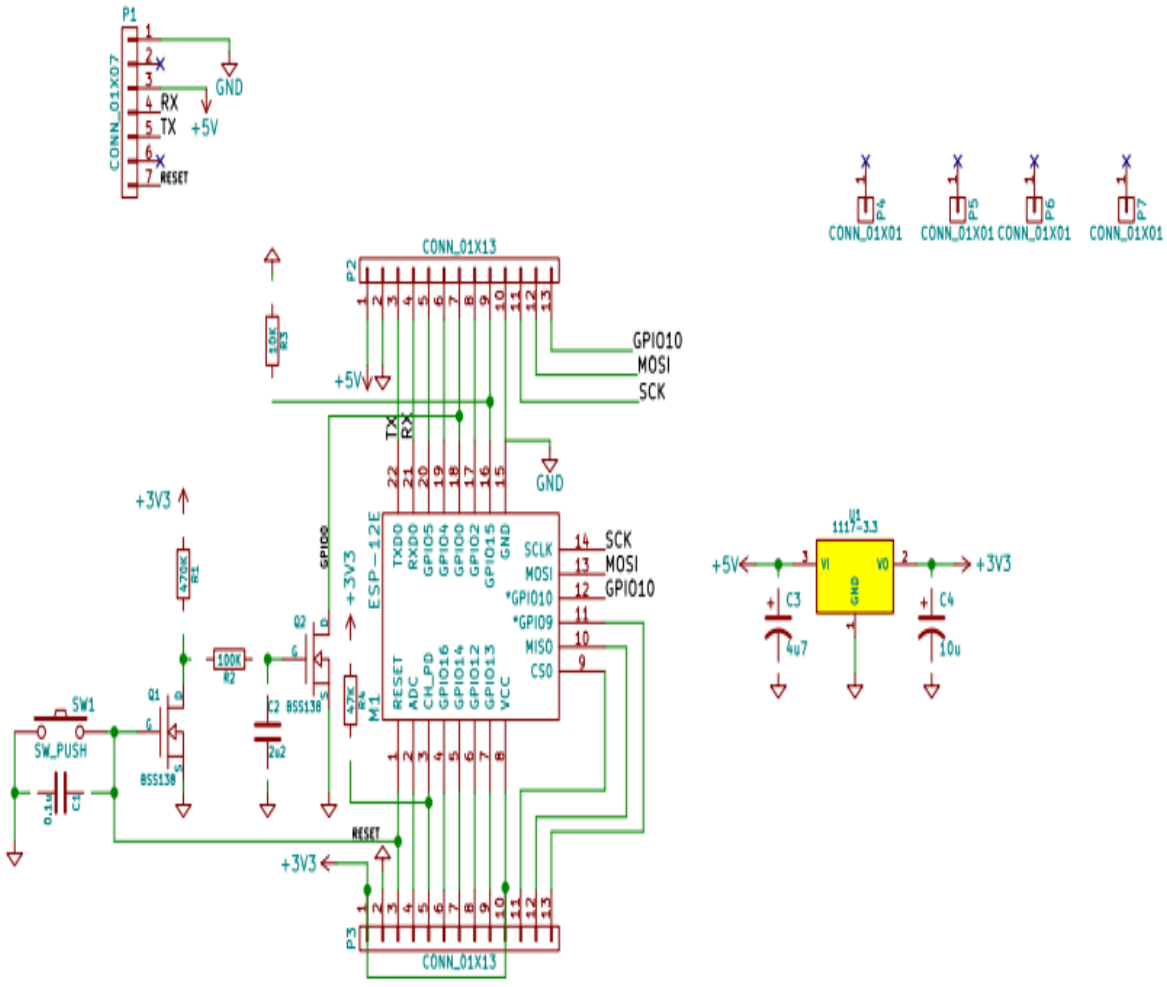


Figure 5.5 Schematic Diagram of Explore ESP8266 Wi-Fi Module.

5.2.9 AT Commands

ESP8266, in its default configuration, boots up into the serial modem mode. In this mode you can communicate with it using a set of **AT commands**. AT commands are based on the Hayes Command Set.

Basic	**Wi-Fi layer**	**TCPIP Layer**
AT	AT+CWMODE	AT+CIPSTATUS
AT+RST	AT+CWJAP	AT+CIPSTART
AT+GMR	AT+CWLAP	AT+CIPSEND
AT+GSLP	AT+CWQAP	AT+CIPCLOSE
ATE	AT+CWSAP	AT+CIFSR
	AT+CWLIF	AT+CIPMUX
	AT+CWDHCP	AT+CIPSERVER
	AT+CIPSTAMAC	AT+CIPMODE
	AT+CIPAPMAC	AT+CIPSTO
	AT+CIPSTA	AT+CIUPDATE
		AT+CIPAP +IPD

5.3 Soil Moisture Sensor

The sensors are the device which converts the physical parameter into the electric signal. The system consists of soil moisture sensor Fig 3. The output of the sensor is analogue signal; the signal is converted into the digital signal and then fed to the processor. The soil moisture sensor is a flow sensor technology, but ideal for monitoring an urban garden or your pet plant's water level and must-have tool for a connected garden. This sensor uses the two physically explore to pass current through the soil, and then it reads that resistance to get the moisture level of the soil. In soil moisture sensor, Copper electrodes are used to sense the moisture content of the soil.

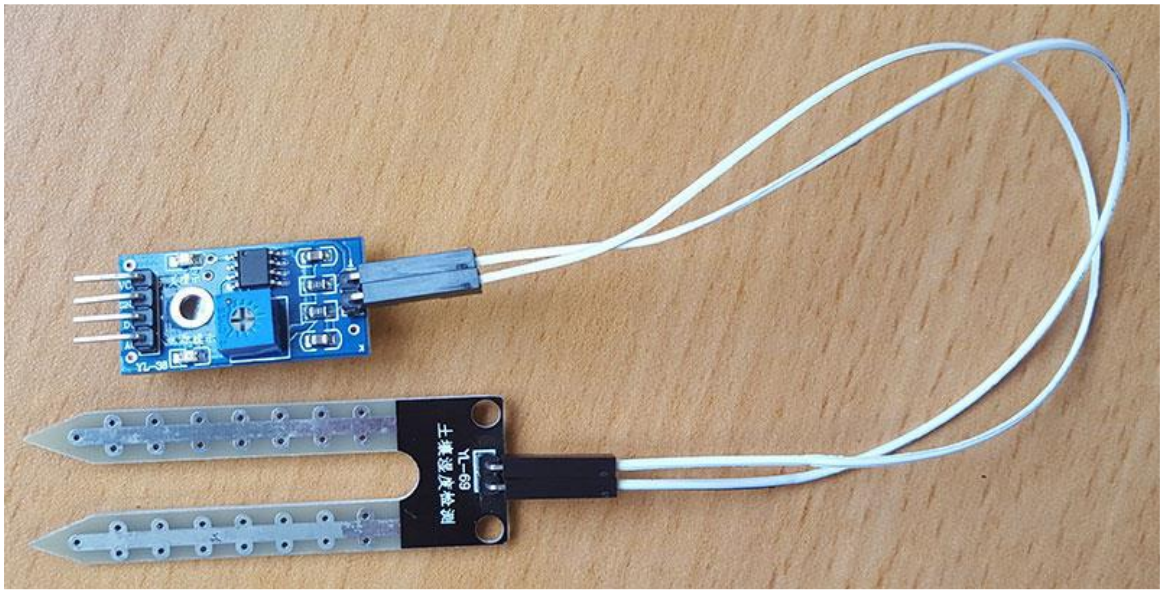


Fig. 5.6 Soil Moisture Sensor

5.4 Submersible Motor Pump

A submersible pump (or sub pump, electric submersible pump) (figure3.8) is a device which has a hermetically sealed motor close-coupled to the pump body. The whole assembly is submerged in the fluid to be pumped. The main advantage of this type of pump is that it prevents pump cavitation's, a problem associated with a high elevation difference between pump and the fluid surface. Small DC Submersible water pumps push fluid to the surface as opposed to jet pumps having to pull fluids. Submersibles are more efficient than jet pumps. It is usually operated between 3v to 12v.

Figure 3.8 Submersible Water Pump.

Specifications:

- Voltage : 2.5-10V
- Maximum lift : 40-110cm / 15.75"-43.4"
- Flow rate : 80-120L/H
- Outside diameter : 7.5mm / 0.3"
- Inside diameter : 5mm / 0.2"
- Diameter : Approx. 24mm / 0.95"
- Length : Approx. 45mm / 1.8"
- Height : Approx. 30mm / 1.2"
- Material : Engineering plastic
- Driving mode : DC design, magnetic driving
- Continuous working life for 500 hours

5.5 Relay Switch

Relay is an electromagnetic device which is used to isolate two circuits electrically and connect them magnetically. They are very useful devices and allow one circuit to switch another one while they are completely separate. They are often used to interface an electronic circuit (working at a low voltage) to an electrical circuit which works at very high voltage. For example, a relay can make a 5V DC battery circuit to switch a 230V AC mains circuit. Thus, a small sensor circuit can drive, say, a fan or an electric bulb. A relay switch can be divided into two parts: input and output. The input section has a coil which generates magnetic field when a small voltage from an electronic circuit is applied to it. This voltage is called the operating voltage. Commonly used relays are available in different configuration of operating voltages like 6V, 9V, 12V, 24V etc. The output section consists of contactors which connect or disconnect mechanically. In a basic relay there are three contactors: normally open (NO), normally closed (NC) and common (COM). At no input state, the COM is connected to NC. When the operating voltage is applied the relay coil gets energized and the COM changes contact to NO.

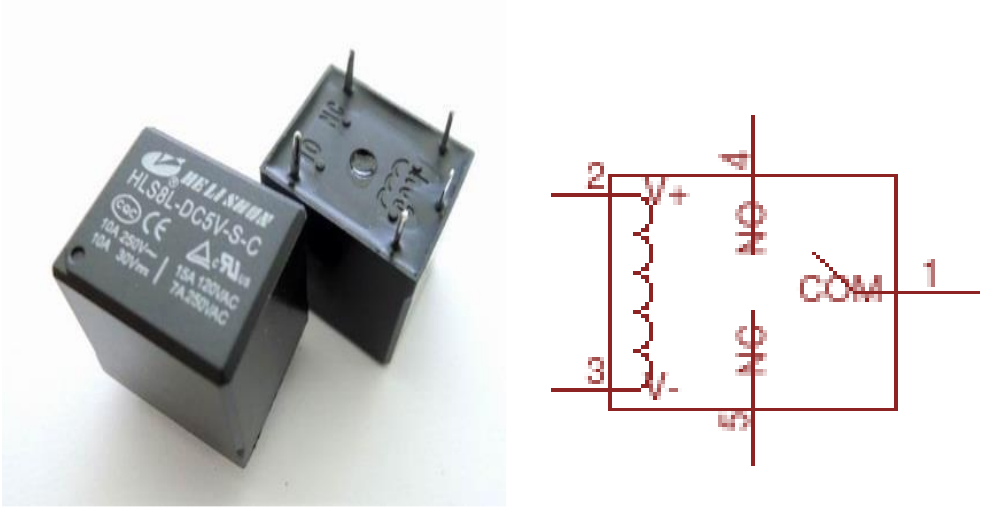


Figure 5.7 Relay switch

6.1 Initial Setups in Arduino IDE Software

Step 1: Install the Arduino 1.6.7 IDE.

Step 2: Go to File>>Preferences>>Additional Boards Manager URLs:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
>>Ok

Step 3: Go to Tools>>Boards>>Boards Manager>>
Download the **"esp8266** by **ESP8266 Community** version **2.2.0"**

Step 4: Go to Tools>>Boards>>Generic ESP8266 Module

Step 5: Go to Tools>>Upload Speed>>115200
Port>>choose preferred COM ports.

6.2 How to Flash ESP8266-12

Step 1: First upload BareMinimum code to Arduino Uno board.

Step 2: Connect the pins of Arduino Uno to ESP8266-12 pins as mention below Arduino Pins ESP8266 pins

- 5v → 5v
- 3.3v → 3.3v, CH_PD
- Gnd → Gnd (both)
- Tx → Tx
- Rx → Rx

Note: Program code is directly uploaded into ESP8266 module. In this case, Arduino board is used as a Flash Burner, i.e. code is directly uploaded to ESP8266 module.

Step 3: Reset the ESP8266 by connecting RESET pin to GND→3.3v→GND and disconnect. But Explore ESP8266 Wi-Fi module has inbuilt Reset button. Press the reset button to reset the module.

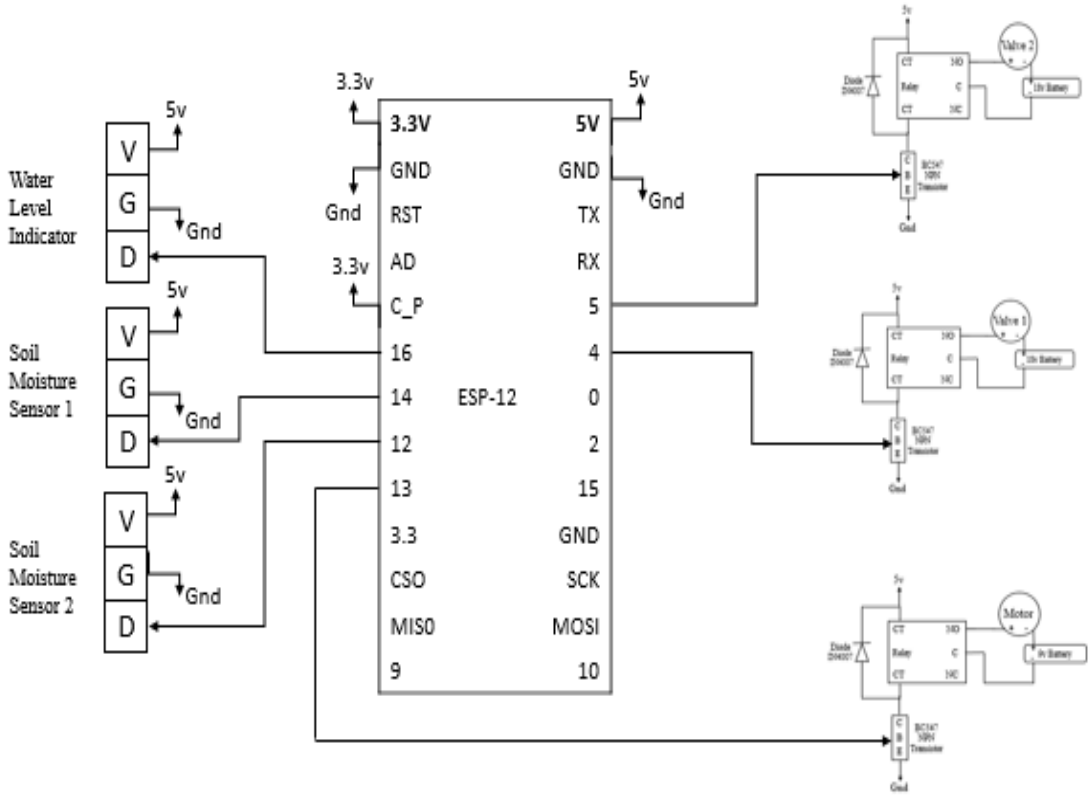
Step 4: While uploading the program code connect the GPIO 0 to GND.

Step 5: Once upload is successful, disconnect the GPIO 0 from GND.

6.3 Experimental Setup
6.3.1 List of Components

Components	Quantity
Explore ESP8266 Wi-Fi Module	1
Relay Switch	1
Power Supply –	
5v	1 (from Arduino)
3.3v	1 (from Arduino)
Gnd	From Arduino

6.3.2 Circuit Connection Procedure



6.3.3 Physical Connection

Physical connections include the placing the sensors and actuators in small model of agriculture field and includes proper connections. Detail is given below

- The field includes two Regions: 1, 2 and a water reservoir.
- Submerse the submersible motor pump in the reservoir.
- Place the valve 1 in Region 1.

- Place the valve 2 in Region 2.
- Make proper pipeline connections from motor pump to the valves.
- Extend the pipeline connection to respective fields to supply water.
- Make some arrangements to supply the water like making the holes to pipes.
- Place the water level indicator in the water reservoir.
- Place the soil moisture sensor 1 in Region 1 near the roots of the plants.
- Place the soil moisture sensor 2 in Region 2 near the roots of the plants.

Give all required supply voltages.

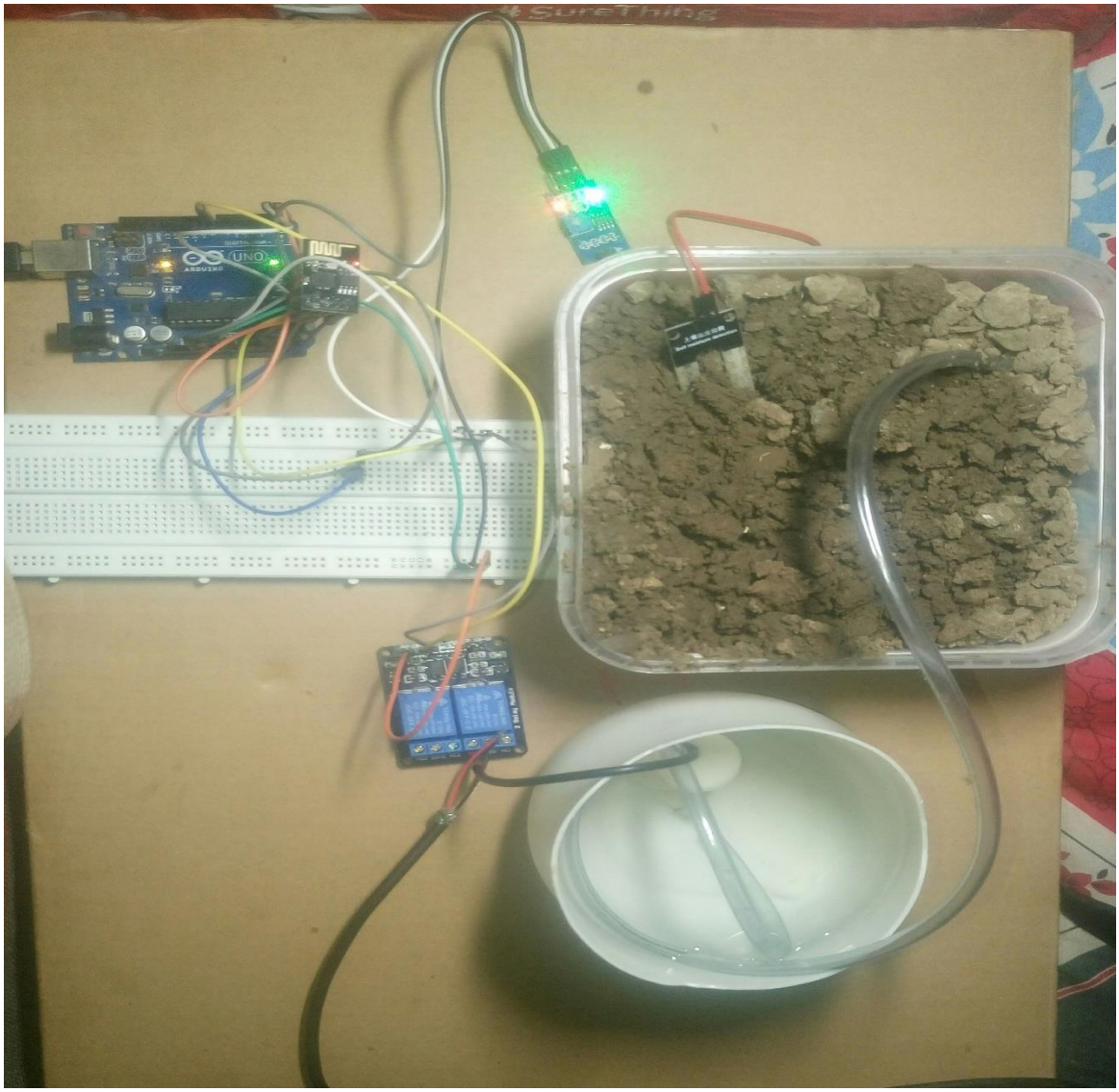


Figure 6.3 Project Setup

6.4 Results

6.4.1 How Module Works?

The installation of the irrigation monitoring and controlling system using IoT is done. It is found that the system works properly and the water is passed to the plants as when required. If the soil is dry, an alert message is sent" MOTOR ON" to the mobile and through text, water pump starts which leads to water to flow. If the soil is wet, an alert message is sent" MOTOR ON" to the mobile and through text, the water pump is turned off and water flow stop. We have used an android application i.e. Blue term. These applications work totally on wi-fi module. To interface the android application and the master robot we require wi-fi module. The application Blue term is used for coding and writing programming instructions and this programming data is sent via wi-fi through the internet. This application acts like an emulator which then is given as the input to the microcontroller Arduino Uno. This Set of codes is then given to the input of the motor driver which is responsible for the movement of the motor. As a result of which the Motor will start and water will be supplied to plants. The same codes are simultaneously sent to the output pin of the microcontroller.

Table 1
MOISTURE REQUIREMENTS

Moisture level	*Motor Status*
1020	*ON*
850	*ON*
220	*OFF*
300	*ON*
32	*OFF*
245	*OFF*

6.4.2 Controlling the Module using MyMQTT Android App

MyMQTT is a simple Message Queue Telemetry Transport (MQTT) client for Android. The service is being provided by **Mosquitto** of **eclipse** a service provider.

Features: Connect to MQTT v3.1 Broker (optional with username and password) Subscribe to various topics.
Publish messages to a topic
Save messages
Filter received Messages

Code:-

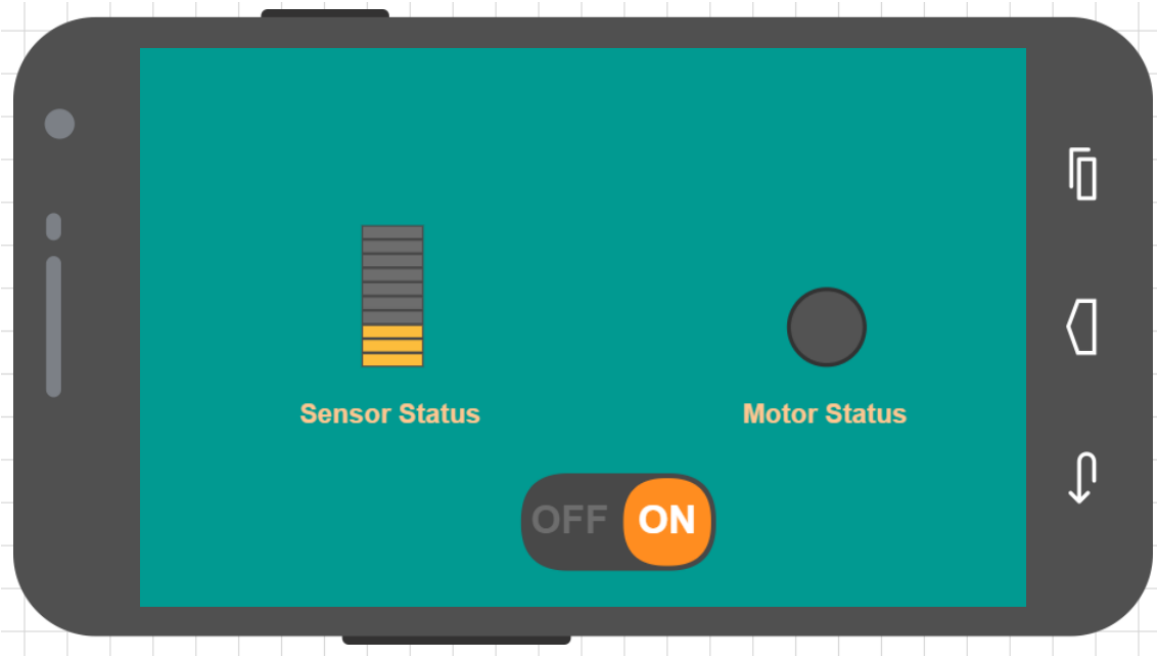
```
/*
-- New project --
This source code of graphical user interface
has been generated automatically by
RemoteXY editor.
To compile this code using RemoteXY library
2.3.3 or later version
download by link http://remotexy.
com/en/library/
To connect using RemoteXY mobile app by
link http://remotexy.
com/en/download/
- for ANDROID 4.1.1 or later version;
- for iOS 1.2.1 or later version;
This source code is free software; you can
redistribute it and/or
modify it under the terms of the GNU Lesser
General Public
License as published by the Free Software
Foundation; either
version 2.1 of the License, or (at your
option) any later version.
*/
//////////////////////////////////////////////
/
// RemoteXY include library //
//////////////////////////////////////////////
/
// RemoteXY select connection mode and
include
library

#defineREMOTEXY_MODE__ESP8266_HARDSERIAL_POIN
T
#include <RemoteXY.h>
```

```
// RemoteXY connection settings

#define REMOTEXY_SERIAL Serial
#define REMOTEXY_SERIAL_SPEED 115200
#define REMOTEXY_WIFI_SSID "RemoteXY"
#define REMOTEXY_WIFI_PASSWORD "12345678"
#define REMOTEXY_SERVER_PORT 6377

// RemoteXY configurate

#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
{ 255,2,0,4,0,58,0,8,13,0,
2,1,9,23,22,11,2,26,31,31,
79,78,0,79,70,70,0,1,0,73,
23,12,12,2,31,88,0,66,130,45,
8,13,10,16,26,65,4,38,24,9,
9,65,2,56,24,9,9,66,0,48,
37,7,16,2,26 };

struct {

// input variable

uint8_t switch_1; // =1 if switch ON and =0
if OFF

// output variable

uint8_t led_1_r; // =0..255 LED Red
brightness
uint8_t led_1_g; // =0..255 LED Green
brightness
int8_t level_2; // =0..100 level position

// other variable
uint8_t connect_flag; // =1 if wire
connected, else =0
```

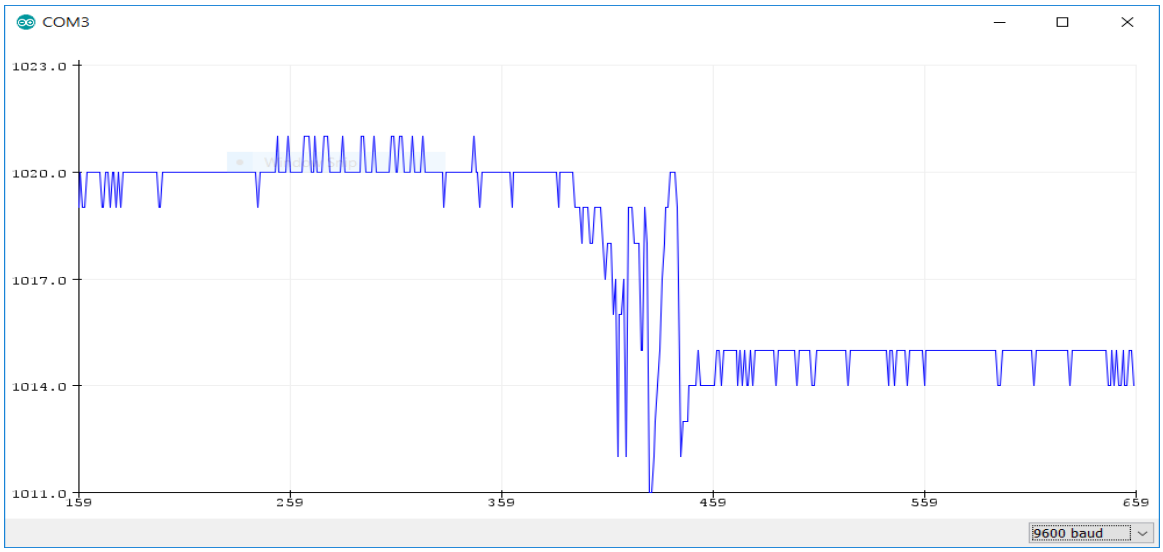
```
} RemoteXY;
#pragma pack(pop)

//////////////////////////////////////////////
// END RemoteXY include //
//////////////////////////////////////////////

#define PIN_SWITCH_1 13
#define PIN_SWITCH_2 11
int sensorPin = A2; // select the input pin
for the potentiometer

// select the pin for the LED

int sensorValue = 0;
void setup()
{
RemoteXY_Init ();
Serial.begin(9600);
pinMode (PIN_SWITCH_1, OUTPUT);
// TODO you setup code
}
void loop()
{
sensorValue=analogRead(sensorPin);
Serial.print("Value Sensed By Soil Moisture
Sensor");
Serial.println(sensorValue);
RemoteXY_Handler ();
sensorValue = analogRead(sensorPin);
RemoteXY.level_2=sensorValue;
if(sensorValue > 250)
{
digitalWrite(PIN_SWITCH_1,HIGH);
digitalWrite(PIN_SWITCH_2,HIGH);
RemoteXY.led_1_g=255;
}
else
```

```
{
digitalWrite(PIN_SWITCH_1,LOW);
digitalWrite(PIN_SWITCH_2,LOW);
RemoteXY.led_1_r=255;
}

//digitalWrite(PIN_SWITCH_1, (RemoteXY.

switch_1==0)?LOW:HIGH);

//digitalWrite(PIN_SWITCH_1, (RemoteXY.

switch_1==0)?LOW:HIGH);

// TODO you loop code
// use the RemoteXY structure for data

transfer
}
```

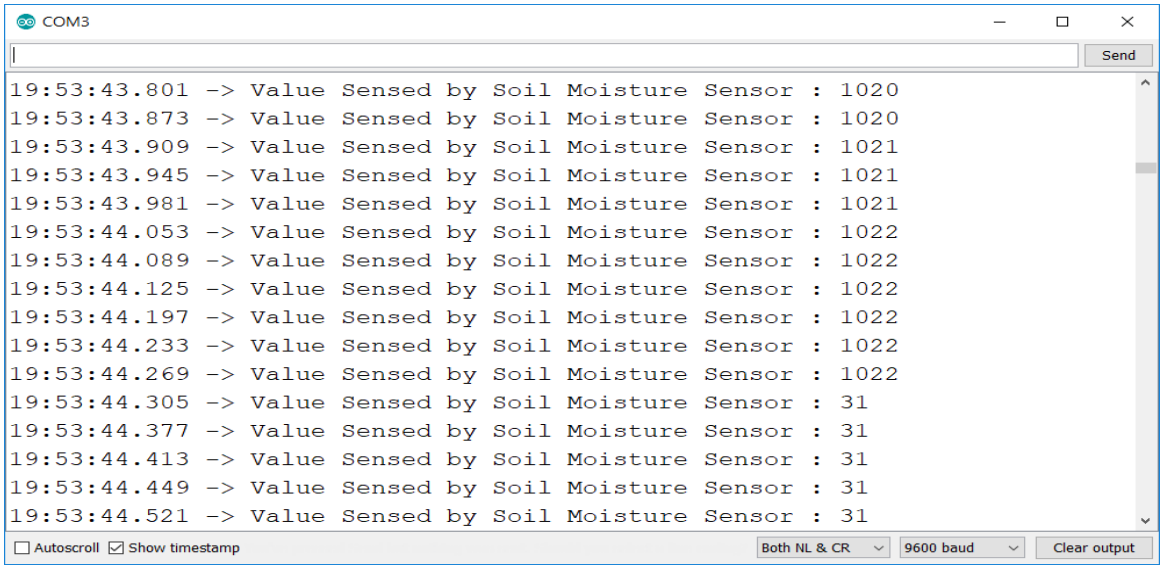
Screenshot:



2.


COM3
1023.0
1020.0
1017.0
1014.0
1011.0
159
259
359
459
559
659
9600 baud

3.


COM3
Send
19:53:43.801 -> Value Sensed by Soil Moisture Sensor : 1020
19:53:43.873 -> Value Sensed by Soil Moisture Sensor : 1020
19:53:43.909 -> Value Sensed by Soil Moisture Sensor : 1021
19:53:43.945 -> Value Sensed by Soil Moisture Sensor : 1021
19:53:43.981 -> Value Sensed by Soil Moisture Sensor : 1021
19:53:44.053 -> Value Sensed by Soil Moisture Sensor : 1022
19:53:44.089 -> Value Sensed by Soil Moisture Sensor : 1022
19:53:44.125 -> Value Sensed by Soil Moisture Sensor : 1022
19:53:44.197 -> Value Sensed by Soil Moisture Sensor : 1022
19:53:44.233 -> Value Sensed by Soil Moisture Sensor : 1022
19:53:44.269 -> Value Sensed by Soil Moisture Sensor : 1022
19:53:44.305 -> Value Sensed by Soil Moisture Sensor : 31
19:53:44.377 -> Value Sensed by Soil Moisture Sensor : 31
19:53:44.413 -> Value Sensed by Soil Moisture Sensor : 31
19:53:44.449 -> Value Sensed by Soil Moisture Sensor : 31
19:53:44.521 -> Value Sensed by Soil Moisture Sensor : 31
Autoscroll
Show timestamp
Both NL & CR
9600 baud
Clear output

7.1 PROJECT EXPENDITURE

1. Arduino UNO Board	-	Rs.425/-
2. Wi-Fi Module	-	Rs.625/-
3. Soil Moisture Sensors	-	Rs.360/-
4. Submersible water Pump	-	Rs.160/-
5. Water Supply Pipes	-	Rs.15/-
6. Relay Switch	-	Rs.140/-
7. Battery 9V (2)	-	Rs.40/-
8. PCB Board	-	Rs.40/-
9. Connecting Wires	-	Rs.90/-
10. 2.54mm female header pins	-	Rs.40/-
11. Field Model	-	Rs.250/-
TOTAL COST	-	Rs.2185/-

8.1 CONCLUSION AND FUTURE SCOPE

In this work, we successfully develop a system that can help in an irrigation monitoring and controlling system using IoT by analyzing the moisture level of the land. The irrigation monitoring and controlling system prove to be a useful system as it automates and regulates the watering without any automatically or electronically intervention. The primary applications for this project are for farmers who do not have enough time to give water in proper time crops/plants. The farmers are facing major problems in watering their agriculture fields. It is because they have no proper idea about when the current available so that they can pump water. The moisture sensors measure the moisture level (water content) of the plants. If the moisture level is found to be below the desired level, the moisture sensor sends the signal to the Arduino Uno and sends an alert message which alerts the Water Pump to turn ON and supply the water to the respective plant. Also, without visiting will get the status of the motor on the smartphone. The system features a sensor is a design for power efficiency, cost effectiveness, cheap components, as well as scalability end ease of use. In future, there are some tasks that should be done and would develop the system to a more fully developed physically state. The system may be further lasting longer than is usual for outdoor utilization.

9.1 REFERENCE

[1] Atzori, L.; Iera, A.; Morabito, G. The Internet of Thing: A survey. Comput. Netw. 2010, 54, 2787-2805

[2] C. Arun, K. Lakshmi Sudha "Agricultural Management using Wireless Sensor Networks – A Survey"2nd International Conference on Environment Science and Biotechnology IPCBEE vol.48(IACSIT Press, Singapore 2012

[3] Jeonghwan Hwang, Changsun Shin, and Hyun Yoe "Study on an Agricultural Environment Monitoring Server System using Wireless Sensor Networks", 2010

[4] Sneha angle "Raspberry Pi and Arduino Based Automated irrigation System" International Journal of Science and Research (IJSR) ISSN(Online):2319-7064

[5] Yogesh G..Gawali, Devendra S. Chaudhari, Hitendra C. Chaudhari "Automated Irrigation System using Wireless sensor networks" International Journal of Advanced research in electronics and communication engineering(IJARECE)ISSN:2278 – 909X volume 5, Issue 6, June 2016