Table of contents

ABSTRACT

Indian agriculture is diverse ranging from impoverished farm villages to developed farms utilizing modern agricultural technologies. Facility agriculture area in China is expanding and is leading the world. However, its ecosystem control technology is still immature, with low level of intelligence. Promoting application of modern information technology in agriculture will solve a series of problems facing by farmers. Lack of exact information and communication leads to the loss in production. Our project is designed to overcome these problems. This system provides an intelligent monitoring platform framework and system structure for facility agriculture ecosystem based on IOT. This will be a catalyst for the transition from traditional farming to modern farming. This also provides opportunity for creating new technology and service development in IOT (internet of things) farming application. The Internet of Things makes everything connected. Over 50 years since independence, India has made immense progress towards food productivity. The Indian population has tripled, but food grain production more than quadrupled there has thus been a substantial increase in available food grain per ca-pita. Modern agriculture practices have a great promise for the economic development of a nation. So we have brought-in an innovative project for the welfare of farmers and also for the farms.

Many advances in technology have made the agriculture business a much less labor intensive industry to be a part of. If we think back even only 50 years, farmers were just beginning to incorporate technologies into their farming techniques. It has been said that individuals that are involved in the farming industry are some of the least susceptible to change. They are very set in the ways of those came before them. When we take a look at the farming industry now, we can see that this is rapidly changing. Farmers are looking for new ways to implement technology to cut costs and reduce labor hours.

INTRODUCTION TO PROJECT

Smart Agriculture developing model is a real time monitoring system. It monitor the soil properties like temperature, humidity soil moisture etc. It is possible to control many operations of the field remotely from anywhere, anytime by IOT. It offers a futuristic way of life in which an individual gets to control his electronic devices using a smart phone, it also offers an efficient use of energy. It applied in all areas of industry, including smart agriculture, smart parking, smart building environmental monitoring, healthcare transportation and many more.

Literature Survey:

In the existing system of agriculture, the crops are being monitored with the help of Arduino boards and GSM technology where in Arduino board's acts as a microcontroller but not as a server. Hence in order to overcome all these features Arduino Nano boards or renesas microcontrollers are being included with the NodeMCU which a latest version is and also acts both as a microcontroller as well as server. Main feature of this methodology is its cheap cost for installation and multiple advantages. Here one can access as well as control the agriculture system in laptop, cell phone or a computer.

Problem Statement:

The proposed project aims to supply water when farm is dry without human presence and avoiding water wastage in irrigation process. Also monitor the soil parameters like temperature, humidity and soil moisture level. This gives signals to the controller whether to send water (that is when farm is dry) to the field or not.

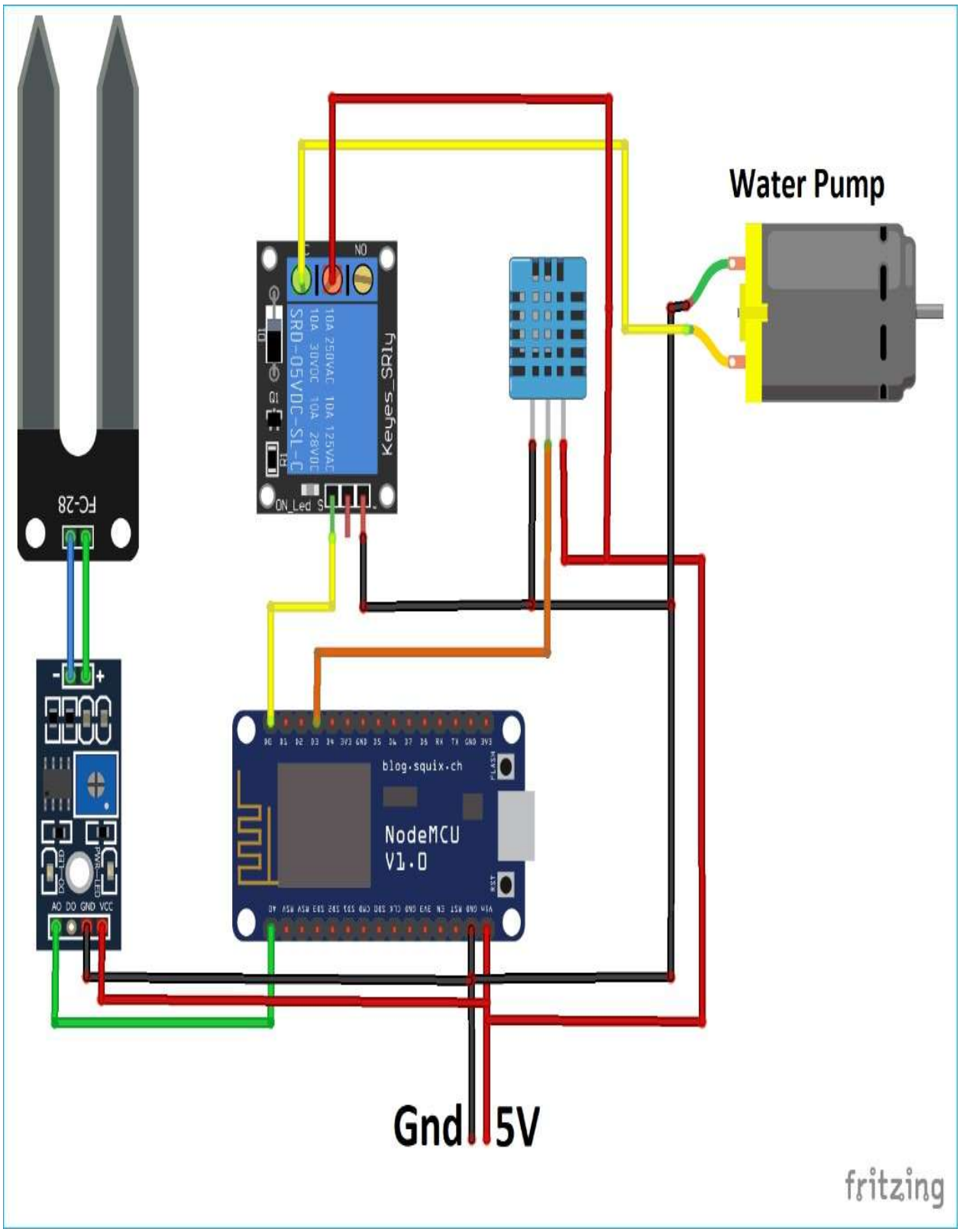
Proposed System:

The smart agriculture model main aim to avoid water wastage in the irrigation process. It is low cost and efficient system. It includes NodeMCU, sensors like soil moisture and Dht11, Water Pump, relays etc.

Components Required:

- NodeMCU ESP8266
- Soil Moisture Sensor Module
- Water Pump Module
- Relay Module
- DHT11

CIRCUIT DIAGRAM


Water Pump
FC-28
SRD-05VDC-SL-C
Keyes_SR1y
NO
ON_Led S
blog.squix.ch
NodeMCU
V1.0
AO DO GND VCC
Gnd
5V
fritzing

HARDWARE

NodeMCU:

NodeMCU is an open source IOT platform it includes firmware which runs on the ESP8266 Wi-Fi SoC from Expressive Systems, and hardware which is based on the ESP-12 module.

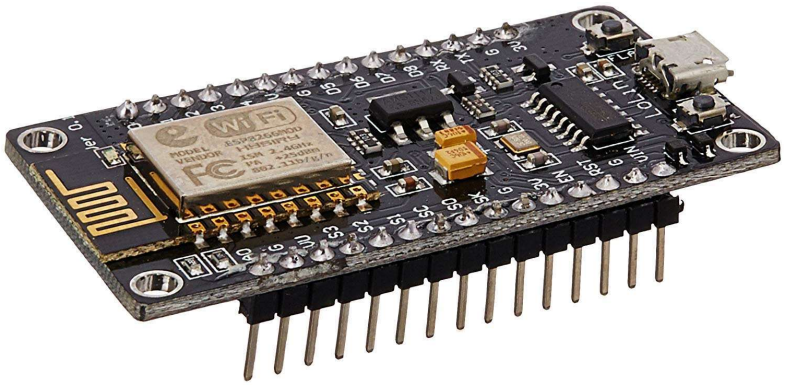


Fig: NodeMCU

The term "NodeMCU" by default refers to the firmware rather than the kits. The firmware uses the Lua scripting language. The programming code is being written for ESP8266 Wi-Fi chip using Arduino IDE, for which installation of ESP8266 library is required. We designed to make working with this chip super easy and a lot of fun. We took a certified module with an onboard antenna, and plenty of pins, and soldered it onto our designed breakout PCBs. While this chip has been very popular, it’s also been very difficult to use. Most of the low-cost modules.

Temperature and humidity sensor:

Humidity Sensor is one of the most important devices that has been widely in consumer, industrial, biomedical, and environmental etc. applications for measuring and monitoring Humidity values. For monitoring the temperature and humidity we use the DHT11 sensor. The DHT11 detects the water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate

with electrodes applied to the surface. When water vapor is absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes. The principle in the humidity sensor is they consist of a humidity sensing component, a NTC temperature sensor (or thermistor) and an IC on the back side of the sensor. For measuring humidity, they use the humidity sensing component which has two electrodes with Humidity moisture holding substrate between them. Humidity indicates the likelihood of predications, dew, or fog. Higher humidity reduces the effectiveness of sweating in cooling the body by reducing the rate of evaporation of moisture from the skin. This effect is calculated in a heat index table or humidex the amount of water vapor that is needed to achieve saturation increases as the temperature increases. As the temperature of a parcel of water becomes lower it will eventually reach the point of saturation without adding or losing water.

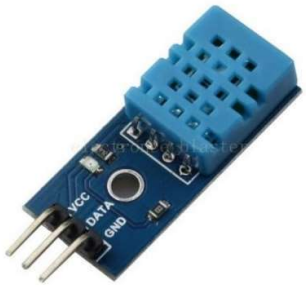


Fig: DHT11

Moisture Sensor:

A simple soil moisture sensor for gardeners. Soil moisture sensors measure the volumetric water content in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighting of a sample, soil moisture sensors measure the

volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content. The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity. Reflected microwave radiation is affected by the soil moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or gardeners.

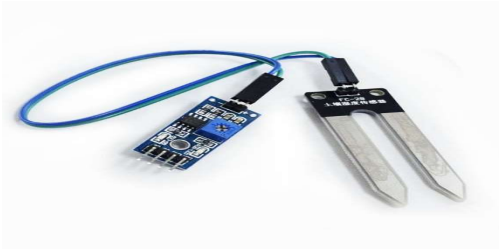


Fig: Moisture sensor

Water Pump:

Water pump is a machine that delivers or pressurizes a liquid. It transfers the mechanical energy of the prime mover or other external energy to the liquid, increasing the energy of the liquid. While mini water motor pump is mini type to transfer water from lower place to higher place or to far place.

Fig: Water Pump

The 16 x 2 character LCD display:

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

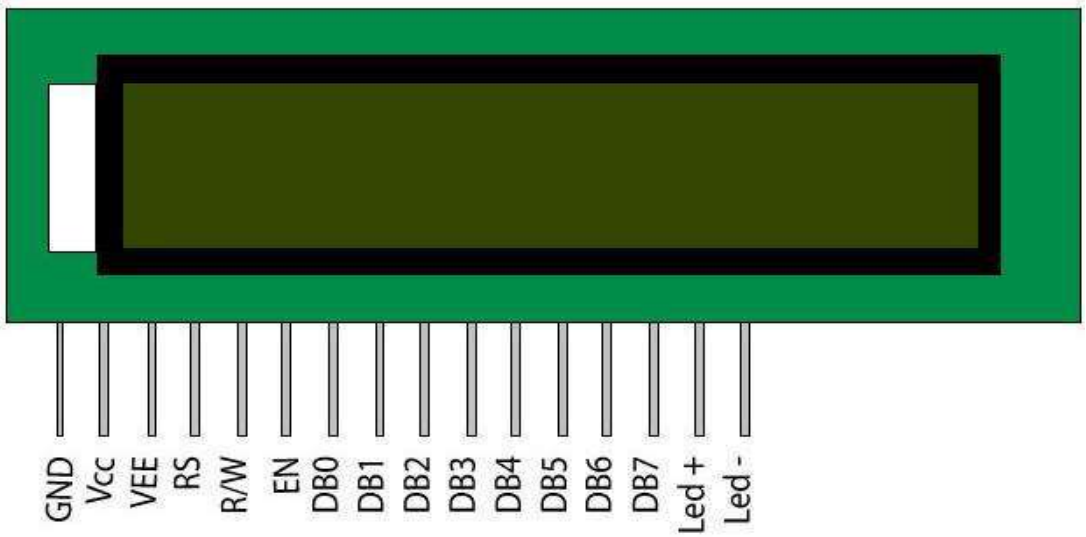


Fig: **LCD display**

PIN. No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	VEE
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the Register	Read/write
6	Sends data to data pins when a high to low pulse is Given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight VCC (5V)	Led+
16	Backlight Ground (0V)	Led-

INTRODUCTION TO WIRELESS COMMUNICATION

E-BOOK

In the world today, everything would be incredibly different if it were not for wireless communication devices. The fact that we can communicate with people in other parts of our own country and the world is amazing and has led to lots of changes in human history. There are various kinds of wireless communication tools and here we will look at a few different kinds as well as the benefits of having them.

Short Distances

Sometimes wireless communication does not happen across thousands of miles but only across a few feet. For instance, if you have a television with a remote control, there is communication happening between the two. The remote control tells the television what to do and this information is transferred without the use of any wires.

You could also use walkie talkies for this purpose. These are little devices that work like telephones except with different methods and will allow two people or a group of people to talk to each other from a small distance away, such as at a festival or celebration.

Benefits of Short Distance Wireless Communication

Having these short distance devices allows many things to happen. You do not have to get up to change the channel, for one thing. With walkie talkies you are able to communicate with a group of people, which means that people who need to work together such as police men or fire fighters or security guards can do so at special events without having to take the time to dial a phone number. This can save time and sometimes lives.

Medium and Long Distances

Anything that can connect to the internet is capable of going through medium or long distances. Many forms of wireless communication such as cell phones and laptop computers can send information from ten feet to thousands of miles because they can connect to the internet. Wireless communication of this form has changed the world dramatically.

One of the most widely acknowledged benefits of long distance wireless communication is that people can perform their jobs at a distance. If they are trying to contact their boss or write a report, they can do so from almost any part of the globe without having to be physically present. This reduces travel costs and the impact of travel on the environment.

Another benefit of long distance wireless communication is that families can stay in touch with each other even if one of them is far away. It used to be that if someone was on a business trip, they would have to write a letter to communicate with their loved ones. Now they can communicate through video and show their family exactly where they are so that the family can connect about these experiences and maintain closer relationships with each other.

The types and forms of wireless communication are changing at a rapid pace to include ever more increasing advances in technology. The benefits of these devices are many and range from benefits that relate to our jobs to those that relate to our loving connection to our families. There is virtually no end to the amount that human kind can advance given the incredible leaps and bounds we are making with technology!

Applications of Wireless Data Communications:

Wireless data communications are an essential component of mobile computing. The various available technologies differ in local availability, coverage range and performance, and in some circumstances, users must be able to employ multiple connection types and switch between them. To simplify the experience for the user, connection manager software can be used, or a mobile VPN deployed to handle the multiple connections as a secure, single virtual network. Supporting technologies include:

Wi-Fi is a wireless local area network that enables portable computing devices to connect easily to the internet. Standardized as IEEE 802.11a,b,g,n, Wi-Fi approaches speeds of some types of wired Ethernet. Wi-Fi has become the de facto standard for access in private homes, within offices, and at public hotspots. Some businesses charge customers a monthly fee for service, while others have begun offering it for free in an effort to increase the sales of their goods.

Cellular data service offers coverage within a range of 10-15 miles from the nearest cellsite. Speeds have increased as technologies have evolved, from earlier technologies such as GSM,CDMA and GPRS, to3G networks such as W-CDMA,EDGE or CDMA2000

Mobile Satellite Communications may be used where other wireless connections are unavailable, such as in largely rural areas or remote locations Satellite Communication are especially important for transportation, aviation, maritime and military.

MICROCONTROLLER OVERVIEW

A **microcontroller** is a small and low-cost microcomputer, which is designed to perform the specific tasks of embedded systems like displaying microwave's information, receiving remote signals, etc.

The general microcontroller consists of the processor, the memory (RAM, ROM, EPROM), Serial ports, peripherals (timers, counters), etc.

Difference between Microprocessor and Microcontroller

The following table highlights the differences between a microprocessor and a microcontroller −

Microcontroller	Microprocessor
Microcontrollers are used to execute a single task within an application.	Microprocessors are used for big applications.
Its designing and hardware cost is low.	Its designing and hardware cost is high.
Easy to replace.	Not so easy to replace.
It is built with CMOS technology, which requires less power to operate.	Its power consumption is high because it has to control the entire system.
It consists of CPU, RAM, ROM, I/O ports.	It doesn't consist of RAM, ROM, I/O ports. It uses its pins to interface to peripheral devices.

Types of Microcontrollers

Microcontrollers are divided into various categories based on memory, architecture, bits and instruction sets. Following is the list of their types −

Bit

Based on bit configuration, the microcontroller is further divided into three categories.

- **8-bit microcontroller** − This type of microcontroller is used to execute arithmetic and logical operations like addition, subtraction, multiplication division, etc. For example, Intel 8031 and 8051 are 8 bits microcontroller.
- **16-bit microcontroller** − This type of microcontroller is used to perform arithmetic and logical operations where higher accuracy and performance is required. For example, Intel 8096 is a 16-bit microcontroller.
- **32-bit microcontroller** − This type of microcontroller is generally used in automatically controlled appliances like automatic operational machines, medical appliances, etc.

Memory

Based on the memory configuration, the microcontroller is further divided into two categories.

- **External memory microcontroller** − This type of microcontroller is designed in such a way that they do not have a program memory on the chip. Hence, it is named as external memory microcontroller. For example: Intel 8031 microcontroller.
- **Embedded memory microcontroller** − This type of microcontroller is designed in such a way that the microcontroller has all programs and data memory, counters and timers, interrupts, I/O ports are embedded on the chip. For example: Intel 8051 microcontroller.

Instruction Set

Based on the instruction set configuration, the microcontroller is further divided into two categories.

- **CISC** − CISC stands for complex instruction set computer. It allows the user to insert a single instruction as an alternative to many simple instructions.
- **RISC** − RISC stands for Reduced Instruction Set Computers. It reduces the operational time by shortening the clock cycle per instruction.

Applications of Microcontrollers

Microcontrollers are widely used in various different devices such as −

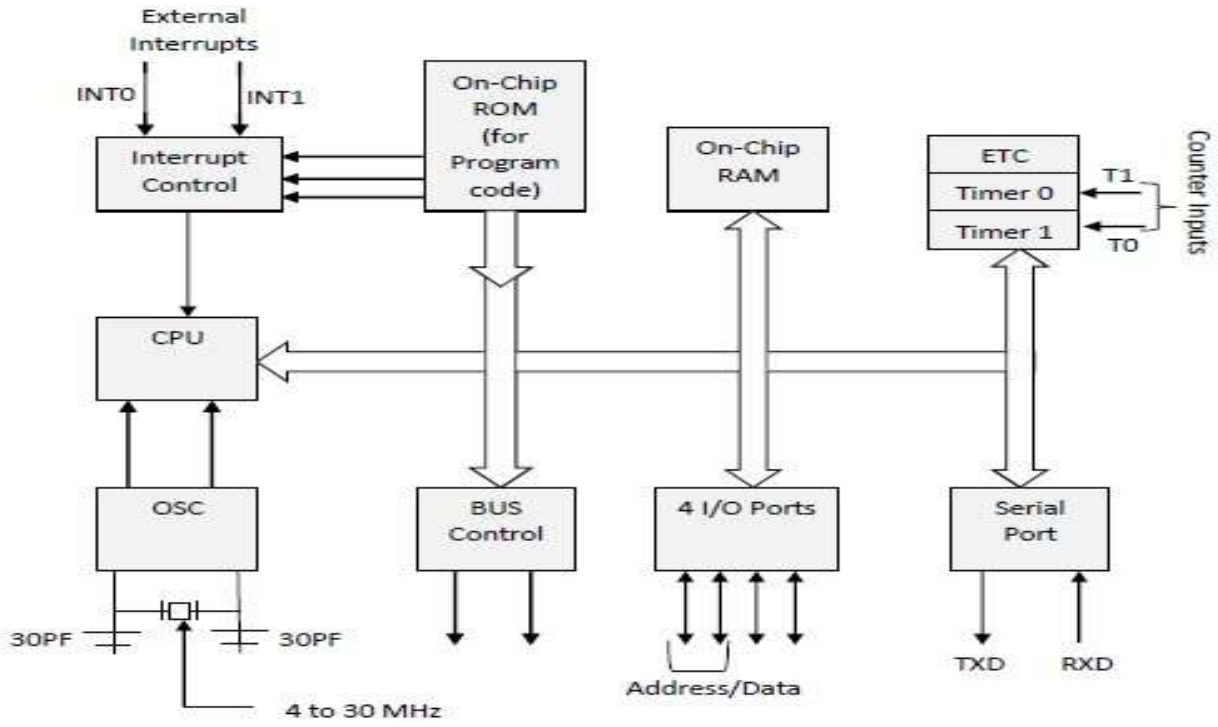
- Light sensing and controlling devices like LED.
- Temperature sensing and controlling devices like microwave oven, chimneys.
- Fire detection and safety devices like Fire alarm.
- Measuring devices like Volt Meter.

8051

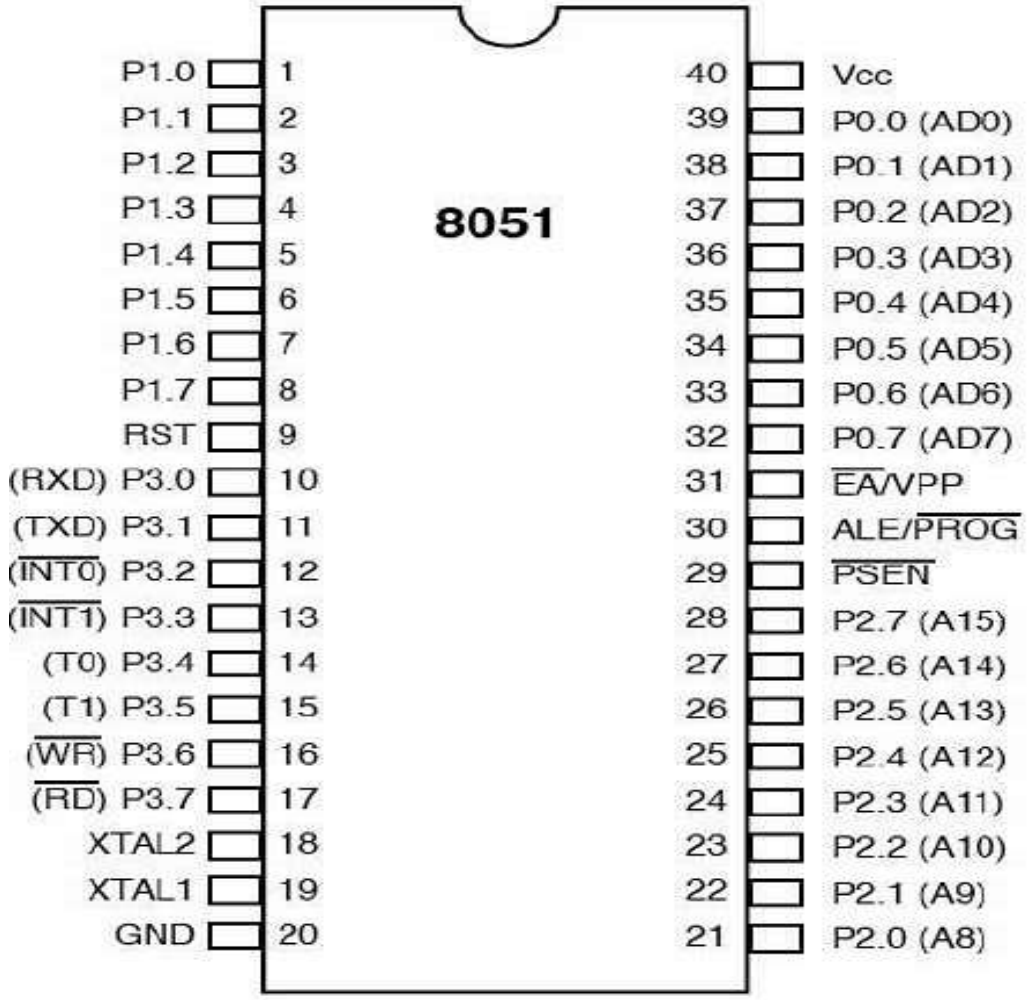
8051 microcontroller is designed by Intel in 1981. It is an 8-bit microcontroller. It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, 2 16-bit timers. It consists of are four parallel 8-bit ports, which are programmable as well as addressable as per the requirement. An on-chip crystal oscillator is integrated in the microcontroller having crystal frequency of 12 MHz.

Let us now discuss the architecture of 8051 Microcontroller.

In the following diagram, the system bus connects all the support devices to the CPU. The system bus consists of an 8-bit data bus, a 16-bit address bus and bus control signals. All other devices like program memory, ports, data memory, serial interface, interrupt control, timers, and the CPU are all interfaced together through the system bus.



The pin diagram of 8051 microcontroller looks as follows −



- **Pins 1 to 8** − These pins are known as Port 1. This port doesn't serve any other functions. It is internally pulled up, bi-directional I/O port.
- **Pin 9** − It is a RESET pin, which is used to reset the microcontroller to its initial values.
- **Pins 10 to 17** − These pins are known as Port 3. This port serves some functions like interrupts, timer input, control signals, serial communication signals RxD and TxD, etc.
- **Pins 18 & 19** − These pins are used for interfacing an external crystal to get the system clock.
- **Pin 20** − This pin provides the power supply to the circuit.
- **Pins 21 to 28** − These pins are known as Port 2. It serves as I/O port. Higher order address bus signals are also multiplexed using this port.

- **Pin 29** − This is PSEN pin which stands for Program Store Enable. It is used to read a signal from the external program memory.
- **Pin 30** − This is EA pin which stands for External Access input. It is used to enable/disable the external memory interfacing.
- **Pin 31** − This is ALE pin which stands for Address Latch Enable. It is used to demultiplex the address-data signal of port.
- **Pins 32 to 39** − These pins are known as Port 0. It serves as I/O port. Lower order address and data bus signals are multiplexed using this port.
- **Pin 40** − This pin is used to provide power supply to the circuit.

8051 microcontrollers have 4 I/O ports each of 8-bit, which can be configured as input or output. Hence, total 32 input/output pins allow the microcontroller to be connected with the peripheral devices.

- **Pin configuration**, i.e. the pin can be configured as 1 for input and 0 for output as per the logic state.
    - **Input/Output (I/O) pin** − All the circuits within the microcontroller must be connected to one of its pins except P0 port because it does not have pull-up resistors built-in.
    - **Input pin** − Logic 1 is applied to a bit of the P register. The output FE transistor is turned off and the other pin remains connected to the power supply voltage over a pull-up resistor of high resistance.
- **Port 0** − The P0 (zero) port is characterized by two functions −
    - When the external memory is used then the lower address byte (addresses A0A7) is applied on it, else all bits of this port are configured as input/output.
    - When P0 port is configured as an output then other ports consisting of pins with built-in pull-up resistor connected by its end to 5V power supply, the pins of this port have this resistor left out.

Input Configuration

If any pin of this port is configured as an input, then it acts as if it “floats”, i.e. the input has unlimited input resistance and in-determined potential.

Output Configuration

When the pin is configured as an output, then it acts as an “open drain”. By applying logic 0 to a port bit, the appropriate pin will be connected to ground (0V), and applying logic 1, the external output will keep on “floating”.

In order to apply logic 1 (5V) on this output pin, it is necessary to build an external pullup resistor.

Port 1

P1 is a true I/O port as it doesn’t have any alternative functions as in P0, but this port can be configured as general I/O only. It has a built-in pull-up resistor and is completely compatible with TTL circuits.

Port 2

P2 is similar to P0 when the external memory is used. Pins of this port occupy addresses intended for the external memory chip. This port can be used for higher address byte with addresses A8-A15. When no memory is added then this port can be used as a general input/output port similar to Port 1.

Port 3

In this port, functions are similar to other ports except that the logic 1 must be applied to appropriate bit of the P3 register.

Pins Current Limitations

- When pins are configured as an output (i.e. logic 0), then the single port pins can receive a current of 10mA.
- When these pins are configured as inputs (i.e. logic 1), then built-in pull-up resistors provide very weak current, but can activate up to 4 TTL inputs of LS series.
- If all 8 bits of a port are active, then the total current must be limited to 15mA (port P0: 26mA).
- If all ports (32 bits) are active, then the total maximum current must be limited to 71mA.

Interrupts are the events that temporarily suspend the main program, pass the control to the external sources and execute their task. It then passes the control to the main program where it had left off.

8051 has 5 interrupt signals, i.e. INT0, TFO, INT1, TF1, RI/TI. Each interrupt can be enabled or disabled by setting bits of the IE register and the whole interrupt system can be disabled by clearing the EA bit of the same register.

IE (Interrupt Enable) Register

This register is responsible for enabling and disabling the interrupt. EA register is set to one for enabling interrupts and set to 0 for disabling the interrupts. Its bit sequence and their meanings are shown in the following figure.

EA	-	-	ES	ET1	EX1	ET0	EX0

EA	IE.7	It disables all interrupts. When EA = 0 no interrupt will be acknowledged and EA = 1 enables the interrupt individually.
-	IE.6	Reserved for future use.
-	IE.5	Reserved for future use.
ES	IE.4	Enables/disables serial port interrupt.
ET1	IE.3	Enables/disables timer1 overflow interrupt.
EX1	IE.2	Enables/disables external interrupt1.
ET0	IE.1	Enables/disables timer0 overflow interrupt.
EX0	IE.0	Enables/disables external interrupt0.

IP (Interrupt Priority) Register

We can change the priority levels of the interrupts by changing the corresponding bit in the Interrupt Priority (IP) register as shown in the following figure.

- A low priority interrupt can only be interrupted by the high priority interrupt, but not interrupted by another low priority interrupt.
- If two interrupts of different priority levels are received simultaneously, the request of higher priority level is served.
- If the requests of the same priority levels are received simultaneously, then the internal polling sequence determines which request is to be serviced.

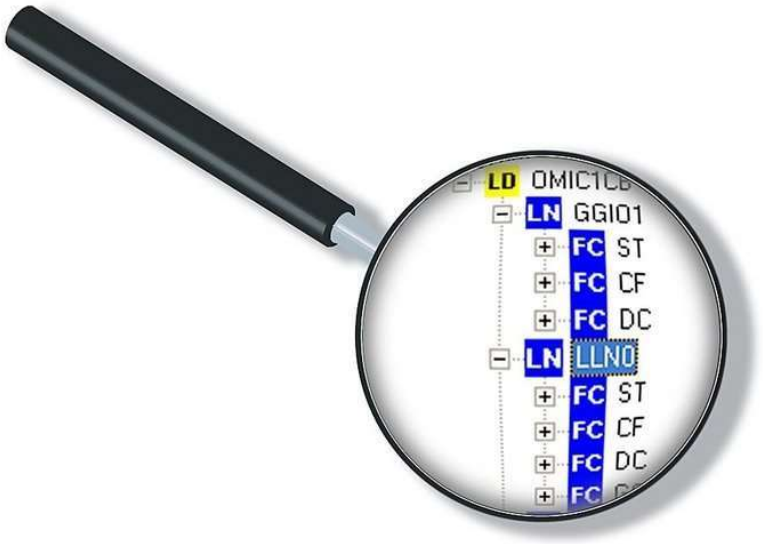
-	-	PT2	PS	PT1	PX1	PT0	PX0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	

-	IP.6	Reserved for future use.
-	IP.5	Reserved for future use.
PS	IP.4	It defines the serial port interrupt priority level.
PT1	IP.3	It defines the timer interrupt of 1 priority.
PX1	IP.2	It defines the external interrupt priority level.
PT0	IP.1	It defines the timer0 interrupt priority level.
PX0	IP.0	It defines the external interrupt of 0 priority level.

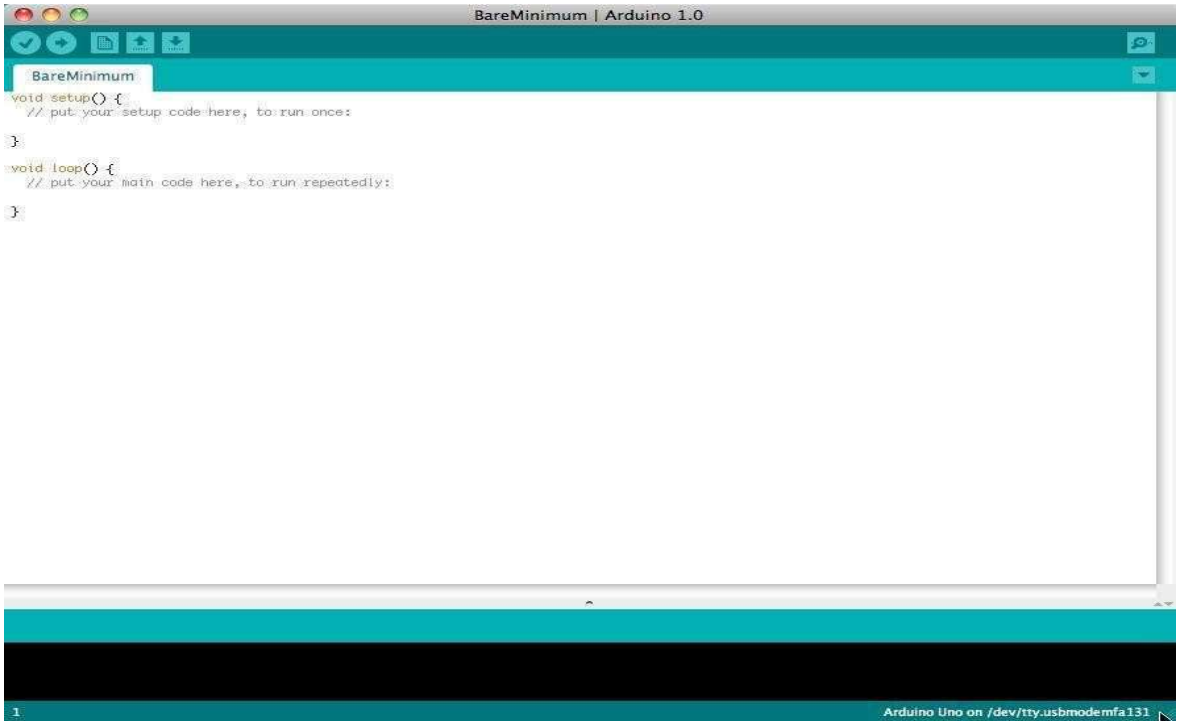
TCON Register

TCON register specifies the type of external interrupt to the microcontroller.

SOFTWARE TOOLS



The project uses Arduino IDE as software to program microcontroller. The program is written in Arduino IDE and compiled and fed into the microcontroller. The following steps are involved into programming a microcontroller using Arduino IDE.



Before you can start doing anything with the Arduino, you need to download and install the Arduino IDE (integrated development environment). From this point on we will be referring to the Arduino IDE as the Arduino Programmer.

The Arduino Programmer is based on the Processing IDE and uses a variation of the C and C++ programming languagesPlug your Arduino to your computer using the programmer as shown before.

Select the board:

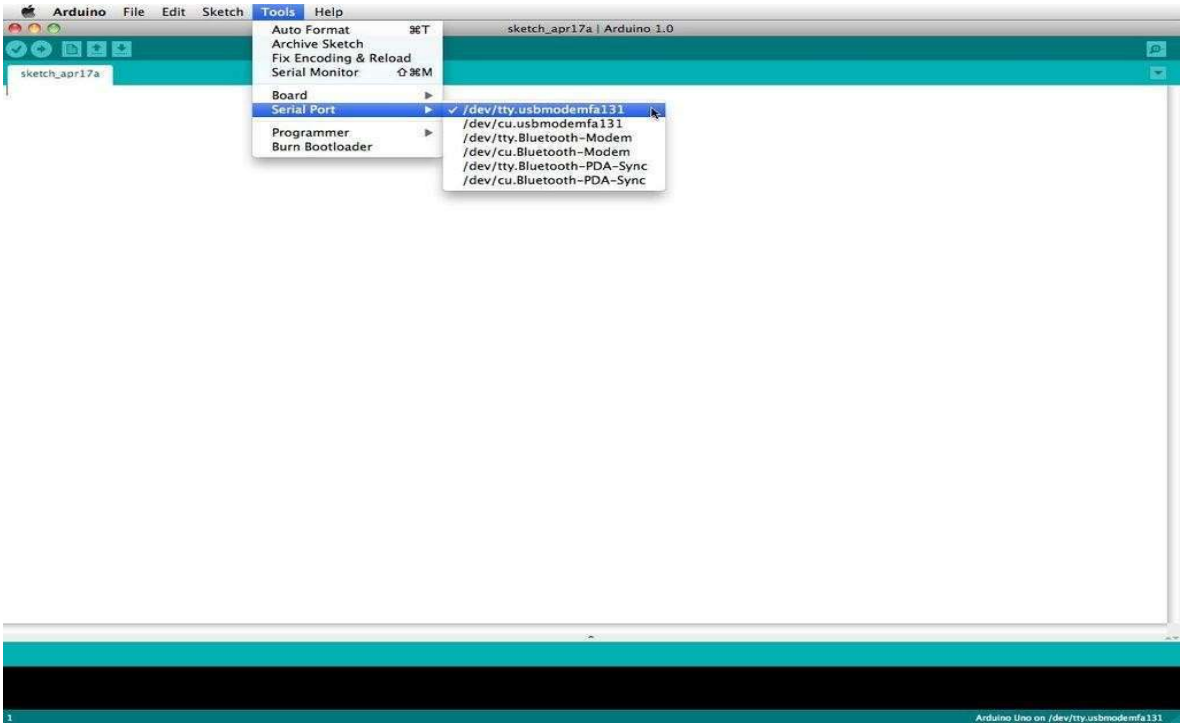
Before compiling the programmer and feeding it onto the arduino board you need to select the appropriate board into which you are feeding the program.

To set the board, go to the following:

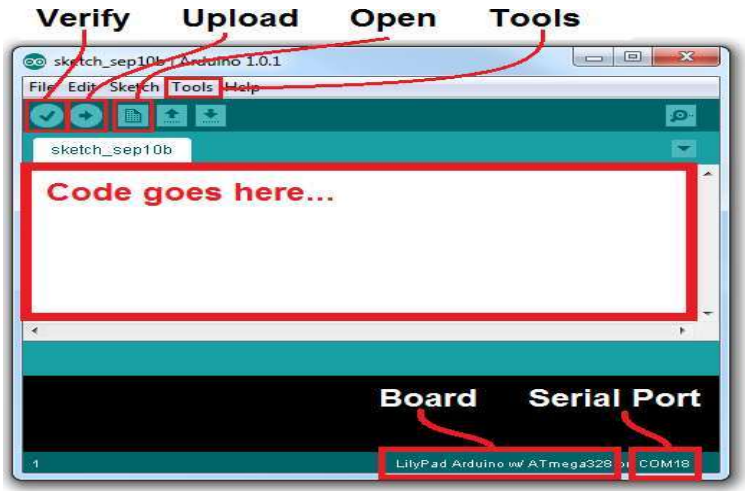
Tools --> Boards

Select the version of board that you are using. Since I have an Arduino Mega plugged in, I obviously selected "Arduino Mega."

To set the serial port, go to the following: Tools --> Serial Port



Compile and upload the sketch to arduino board:



STEPS TO WRITE AN ASSEMBLY LANGUAGE PROGRAM IN Arduino AND HOW TO COMPILE IT:

1. Install the Arduino Software in the PC in any of the drives.
2. After installation, an icon will be created with the name “Arduino Uno”. Just drag this icon onto the desktop so that it becomes easy whenever you try to write programs in Arduino.
3. Double click on this icon to start the Arduino compiler.
4. A page opens with different options in it showing the project workspace at the leftmost corner side, output window in the bottom and an ash coloured space for the program to be written.
5. Now to start using the keil, click on the option “project”.
6. A small window opens showing the options like new project, import project, open project etc. Click on “New project”.
7. A small window with the title bar “Create new project” opens. The window asks the user to give the project name with which it should be created and the destination location. The project can be created in any of the drives available. You can create a new folder and then a new file or can create directly a new file.
8. After the file is saved in the given destination location, a window opens where a list of vendors will be displayed and you have to select the device for the target you have created.
9. The most widely used vendor is Atmel. So click on Atmel and now the family of microcontrollers manufactured by Atmel opens. You can select any one of the microcontrollers according to the requirement.
10. When you click on any one of the microcontrollers, the features of that particular microcontroller will be displayed on the right side of the page. The most appropriate microcontroller with which most of the projects can be implemented is the AT89S52. Click on this microcontroller and have a look at its features. Now click on “OK” to select this microcontroller.

11. A small window opens asking whether to copy the startup code into the file you have created just now. Just click on “No” to proceed further.
12. Now you can see the TARGET and SOURCE GROUP created in the project workspace.
13. Now click on “File” and in that “New”. A new page opens and you can start writing program in it.
14. After the program is completed, save it with any name but with the .asm extension. Save the program in the file you have created earlier.
15. You can notice that after you save the program, the predefined keywords will be highlighted in bold letters.
16. Now add this file to the target by giving a right click on the source group. A list of options open and in that select “Add files to the source group”. Check for this file where you have saved and add it.
17. Right click on the target and select the first option “Options for target”. A window opens with different options like device, target, output etc. First click on “target”.
18. Since the set frequency of the microcontroller is 11.0592 MHz to interface with the PC, just enter this frequency value in the Xtal (MHz) text area and put a tick on the Use on-chip ROM. This is because the program what we write here in the keil will later be dumped into the microcontroller and will be stored in the inbuilt ROM in the microcontroller.
19. Now click the option “Output” and give any name to the hex file to be created in the “Name of executable” text area and put a tick to the “Create HEX file” option present in the same window. The hex file can be created in any of the drives. You can change the folder by clicking on “Select folder for Objects”.
20. Now to check whether the program you have written is errorless or not, click on the icon exactly below the “Open file” icon which is nothing but Build Target icon. You can even use the shortcut key F7 to compile the program written.
21. To check for the output, there are several windows like serial window, memory window, project window etc. Depending on the program you have written, select the appropriate window to see the output by entering into debug mode.

22. The icon with the letter “d” indicates the debug mode.
23. Click on this icon and now click on the option “View” and select the appropriate window to check for the output.
24. After this is done, click the icon “debug” again to come out of the debug mode.
25. The hex file created as shown earlier will be dumped into the microcontroller.

PROGRAM

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
String apiKey = "IQBK3DVZRC8WHH9J";     //  Enter your Write API key here
const char* server = "api.thingspeak.com";
const char *ssid =  "chirag";     // Enter your WiFi Name
const char *pass =  "12345678"; // Enter your WiFi Password
#define DHTPIN D3          // GPIO Pin where the dht11 is connected
DHT dht(DHTPIN, DHT11);
WiFiClient client;
LiquidCrystal_I2C lcd(0x3F,16,2);

const int rain = D5;
const int soil = D0;
const int motorPin = D6;
const int motorPin1 = D7;
const int sw1 = D5;
 int sensorVal;
unsigned long interval = 10000;
unsigned long previousMillis = 0;
unsigned long interval1 = 1000;
unsigned long previousMillis1 = 0;

float h;                 // humidity reading
float t;                //temperature reading
float r;
float rainsensor;
float soilsensor;
void setup()
{
  Serial.begin(115200);
  delay(10);


  lcd.init(); //initialize the lcd
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Smt Agriculture");
  lcd.setCursor(0,1);
  lcd.print("System using IOT ");
  delay(500);
  pinMode(motorPin, OUTPUT);
```

```
  pinMode(motorPin1, OUTPUT);
   pinMode(sw1,INPUT_PULLUP);
  pinMode(r, INPUT);
  digitalWrite(motorPin, LOW); // keep motor off initally
  dht.begin();

  digitalWrite(motorPin, HIGH);
     digitalWrite(motorPin1, HIGH);
     delay(10000);
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
   delay(500);
   Serial.print(".");              // print ... till not connected
  }
  Serial.println("");
  Serial.println("WiFi connected");
}

void loop()
{
  unsigned long currentMillis = millis(); // grab current time

  h = dht.readHumidity();     // read humiduty
  t = dht.readTemperature();     // read temperature

  if (isnan(h) || isnan(t))
  {
   Serial.println("Failed to read from DHT sensor!");
   return;
  }


if ((unsigned long)(currentMillis - previousMillis) >= interval) {

  sendThingspeak();           //send data to thing speak
  previousMillis = millis();
  client.stop();
}
int rainsensor = digitalRead(rain);
  // print out the state of the button:
  Serial.println(rain);
```

```
  delay(1);
  if (rain == 1)
  {
    rainsensor = 1;
    Serial.println("Its Raining");
    delay(100);
    lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("  Its Raining  ");
  }
  else
  {
    rainsensor = 0;
    Serial.println("Its not Raining");
    delay(100);
  }
//////////////////////////////////////////////////
 sensorVal = digitalRead(soil);
  //print out the value of the pushbutton
  Serial.println(sensorVal);

  // Keep in mind the pull-up means the pushbutton's logic is inverted. It goes
  // HIGH when it's open, and LOW when it's pressed. Turn on pin 13 when the
  // button's pressed, and off when it's not:
  if (sensorVal == HIGH) {
    digitalWrite(13, LOW);

  } else {
    digitalWrite(13, HIGH);
  }
//////////////////////////////////////////////////
if (digitalRead(sw1) == LOW)
      {
      lcd.setCursor (0,0);
      lcd.print("Motor ON");
      digitalWrite(motorPin, LOW);
      delay(10000);
      }
      else
      {
        digitalWrite(motorPin, HIGH);
      delay(100);
      }
```

```
////////////////////////////////////
}

void sendThingspeak() {
  if (client.connect(server, 80))
  {
    String postStr = apiKey;              // add api key in the postStr string
    postStr += "&field1=";
    postStr += String(sensorVal);    // add mositure readin
    postStr += "&field2=";
    postStr += String(t);                // add tempr readin
    postStr += "&field3=";
    postStr += String(h);                 // add humidity readin
     postStr += "&field4=";
    postStr += String(rainsensor);
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());          //send lenght of the string
    client.print("\n\n");
    client.print(postStr);                     // send complete string
    Serial.print("Moisture Percentage: ");
    Serial.print(sensorVal);
    Serial.print("%. Temperature: ");
    Serial.print(t);
    Serial.print(" C, Humidity: ");
    Serial.print(h);
    Serial.print(" Rain: ");
    Serial.print(rainsensor);
    Serial.println("%. Sent to Thingspeak.");
    lcd.clear();
    lcd.setCursor(0,0);       //display latitude and longitude on 16X2 lcd display
    lcd.print("Soil: ");
    lcd.print(sensorVal);
    lcd.setCursor(8,0);       //display latitude and longitude on 16X2 lcd display
    lcd.print(" Rain: ");
    lcd.print(rainsensor);
    lcd.setCursor(0,1);
    lcd.print("T:");
```

```
    lcd.print(t);
    lcd.setCursor(7,1);
    lcd.print("  H:");
    lcd.print(h);
    delay(100);
  }
}
```

Thingspeak:

ThingSpeak is a platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs. We will consider each of these features in detail below.

The core element of ThingSpeak is a 'ThingSpeak Channel'. A channel stores the data that we send to ThingSpeak and comprises of the below elements:

- 8 fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.
- 3 location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
- 1 status field - A short message to describe the data stored in the channel.

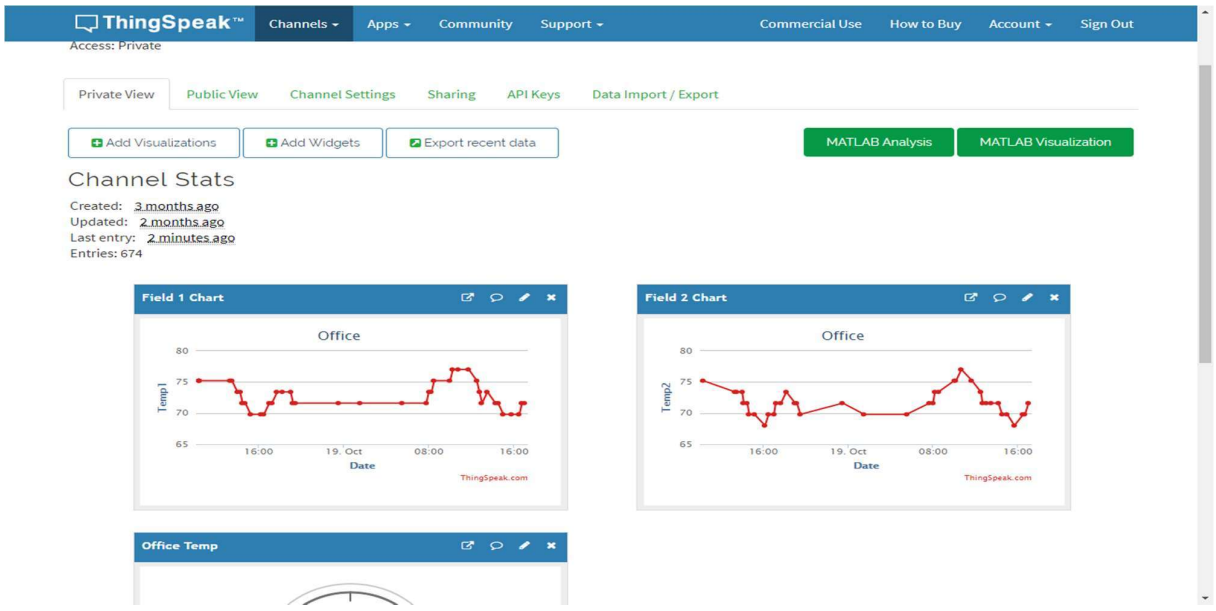


FIG:THINGSPEAK

ADVANTAGES:

- More Labors not required
- It allows farmers to maximize yields using minimum resources such as water, fertilizer, seeds etc.
- Solar power and mobile operated pumps save cost and electricity
- He can Monitor the agriculture field by any where
- Best irrigation system

DISADVANTAGES:

- The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.
- The smart farming based equipments require farmers to understand and learn the use of technology. This is major challenge in adopting smart agriculture farming at large scale across the countries.

CONCLUSION

The smart agriculture using IOT has been experimentally proven to work satisfactorily by monitoring the values of humidity and temperature successfully. It also stores the sensor parameters in the timely manner. This will help the user to analyse the conditions of various parameters in the field anytime anywhere. Then control or maintain the parameters of field properly. Finally, we conclude that automatic irrigation system is more efficient than scheduled irrigation process.