Table of contents

Sr No.		Contents
		Abstract
1.		Introduction
2.		Methodology
	2.1	Objectives
3.		Implementation
	3.1	Hardware Devices
	3.2	Software Devices
4.		Testing and Verification
	4.1	Results
5.		Applications and Challenges Faced
6		Conclusion and future work
7		Acknowledgement
8.		References
9.		Code

IoT Based Smart Pots

Sheethal Halandur Nagaraja (15), Shreya Shah (39), Megha Nair (32)
Software Engineering Department, College of Engineering
San Jose State University, San Jose, CA 95112

***Abstract*- Smart home automation is an emerging Internet of Things (IoT) application domain which aims at providing a comfortable living atmosphere with minimum human intervention along with optimizing the consumption of energy resources. As a part of home automation, Smart Pots project provides an efficient solution for remotely monitoring the plant condition by sensing soil moisture, temperature, humidity, and light intensity using DTH11, BH1750FVI, and soil moisture sensors. This data is uploaded to cloud using NodeMCU with the ESP8266 Wi-Fi module and can be accessed through visualization dashboards configured using IoT Analytics platform from anywhere using smartphones or laptops. Sensed soil moisture data is then used by the automatic watering system with a submersible water pump to start watering the plants when soil dryness is more than the set threshold. This project provides a feasible cost-effective solution to help people to remotely take care of their plants in their physical absence.**

Keywords- Home Automation, IoT Plant Monitoring, IoT plant watering, Automation using NodeMCU

1 Introduction

Plants add beauty to the environment and purify the air, enhancing the quality of life. But many times, we forget to take care of our plants due to our busy lifestyles or due to frequent trips. The Smart Flower Pot device helps us to communicate with our plants. This device can collect data about the moisture of the soil, light intensity, temperature and humidity of the air with the help of sensors, then display the results on the LCD display. It enables us to upload data to network with the help of a wireless communication module. The data will be available online and can be viewed anytime and anywhere.
This system alarms the user about soil moisture through LED along with measuring outside temperature and humidity. This tool can automatically provide water to the soil with a vinyl tube attached to the water pump whenever the soil moisture is low by making the modification to the potentiometer to control the threshold of soil humidity.

2 Methodology

2.1 Objectives

The objective of "Smart Pot" includes measurement of Humidity of soil, light intensity as well as the humidity and temperature of the air at a regular interval (5 seconds). The data collected is then displayed on the 1602 LCD display. The LED should light up as soon as the humidity of the soil becomes low. The soil moisture, humidity, light intensity must be consistently uploaded to thing speak and the graphs should display the data at a regular interval (5 seconds). This plant monitoring system helps the user to automatically water the plant because as soon as the moisture content of the soil reaches beyond the threshold value, water through the vinyl tube starts pouring into the soil till moisture level comes up to our max dryness threshold (700).

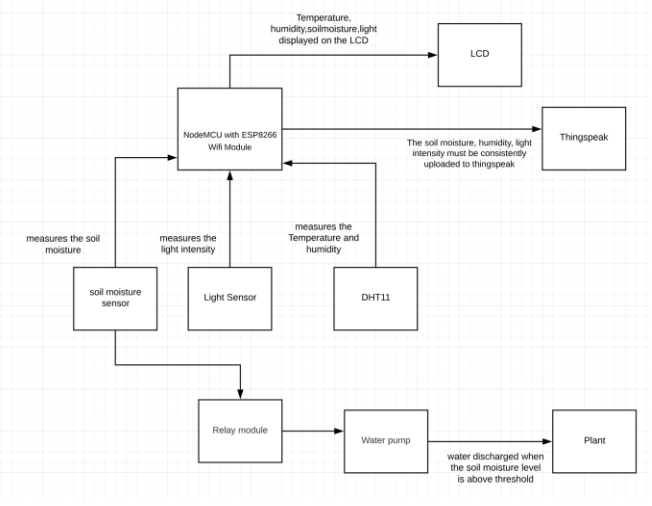


Figure 1: Process Flow Diagram for Smart Pots

3 Implementation

In this section, we describe the various hardware and software components used in this project.

3.1 Hardware Design

The following are the major hardware components used in the project:

3.1.1 NodeMCU 1.0 with ESP8266

NodeMCU is a single board microcontroller which runs on the ESP8266 Wifi SoC. It's hardware is based on an ESP-12 module. It is a breadboard friendly microcontroller with LUA based firmware. The following are the specifications of the microcontroller [1]:

1. Wi-Fi Module – ESP-12E module similar to ESP-12 module but with 6 extra GPIOs.
2. USB – micro USB port for power, programming and debugging
3. Headers – 2x 2.54mm 15-pin header with access to GPIOs, SPI, UART, ADC, and power pins
4. Misc – Reset and Flash buttons
5. Power – 5V via micro USB port
6. Dimensions – 49 x 24.5 x 13mm

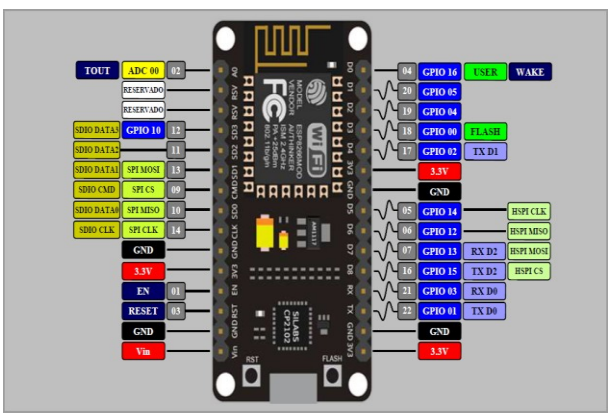


Figure 2: NodeMCU with ESP8266 Wifi Module

3.1.2 DHT11 Temperature Sensor

It is a reduced cost sensor that features adjusted digital output with humidity and temperature sensing capabilities. It consists of two major components: capacitive humidity sensor and a thermistor which identifies the surrounding temperature and air and reflects out on the data pins. It also contains 8- bit high-performance microcontroller which ensures high stability and reliability.

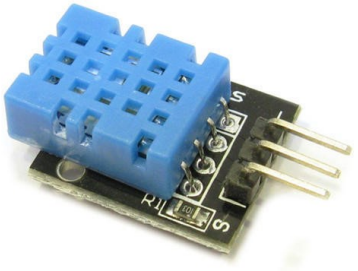


Figure 3: DHT11 Temperature & Humidity Sensor

The following are the specifications of the sensor:

1. Supply Voltage: +5 V
2. Temperature range: 0-50 °C error of ± 2 °C
3. Humidity :20-90% RH ± 5% RH error
4. Interface: Digital

3.1.3 BH1750FVI Light Sensor

It is an effective light sensor with generated digital output signal Integrated Circuit for I2C bus interface. Wide range of high-resolution output can be detected. Following are some of the important features of this sensor [2].

1. Illuminance to Digital Converter
2. Wide range and High resolution. (1 - 65535 lx)
3. Low Current by power down function
4. 50Hz / 60Hz Light noise reject-function
5. I2C bus Interface (f / s Mode Support)
6. It is possible to select 2 types of I2C slave-address.
7. It is possible to detect min. 0.11 lx, max. 100000 lx by using this function.

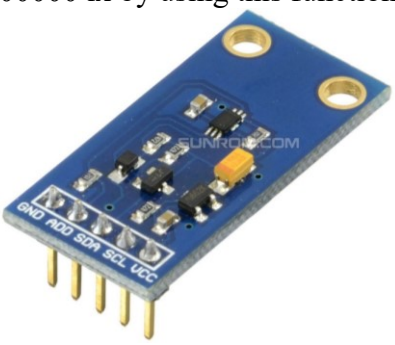


Figure 4: BH1750FVI Light Sensor

3.1.4 Soil Moisture Sensor with Probe

The soil moisture sensor consists of two probes which are used to measure the volumetric content of water. The two probes allow the current to pass through the soil and then it gets the resistance value to measure the moisture value. [4]

The moisture in the soil is high when the soil conducts a good amount of electricity meaning less resistance. So when the amount of water content is less in soil (Dry Soil), it conducts very less electricity implying more resistance.

This sensor can be connected in two modes; Analog mode and digital mode. First, we will connect it in Analog mode and then we will use it in Digital mode.[4]

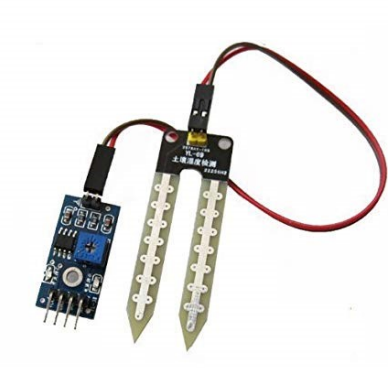


Figure 5: Soil Moisture Sensor with Probe

The following are the interface specifications: [4]

1. VCC: .3 V-5V
2. GND: GND
3. DO: digital output interface (0 and 1)
4. AO: Analog Output Interface

3.1.5 1602 LCD Display

It is a display with two rows which holds 16 characters in each row. It contains LED backlight which shows the text in white color with a blue background. It is compatible with almost all the type of microcontrollers like Arduino board, Raspberry Pi, NodeMCU, etc.

It uses the Liquid Crystal library which allows any board to control these LCDs based on a compatible chipset. The library works within either 4- or 8-bit mode (i.e. using 4 or 8 data lines.[3]

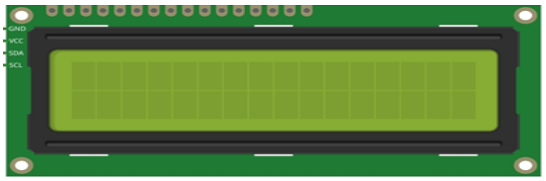


Figure 6: 1602 LCD

It is a dot matrix module which supports all the formats like character, digits, and letters. The following are the characteristics [6]:

1. Display Mode: STN, BLUB
2. Display Format: 16 Character x 2 Line
3. Viewing Direction: 6 O'Clock
4. Input Data: 4-Bits or 8-Bits interface available
5. Display Font: 5 x 8 Dots
6. Power Supply: Single Power Supply (5V±10%)
7. Driving Scheme: 1/16Duty, 1/5Bias
8. Backlight (SIDE)：LED (WHITE)

3.1.6 Relay Module

A relay is an electrically operated switch that can be turned on or off, letting the current go through or not, and can be controlled with low voltages, like the 5V provided by the Arduino pins [5]. We are using a 5V single channel relay module with status LED, diode and transistor trigger to control the flow of current to water pump based on the readings of the soil moisture sensor.

Figure 7: Relay Module

Below are the interface specifications [7]:

1. Type: Digital
2. Single relay board
3. Rated through-current: 10A (NO) 5A (NC)
4. Maximum switching voltage: 5V
5. Control signal: TTL level

6. Contact Rating (Res. Load):10A 277VAC/24VDC
7. Max. switching voltage 250VAC/30VDC 250VAC/30VDC
8. Max. switching current: 10A
9. Operate time: 10ms
10. Release time: 5ms

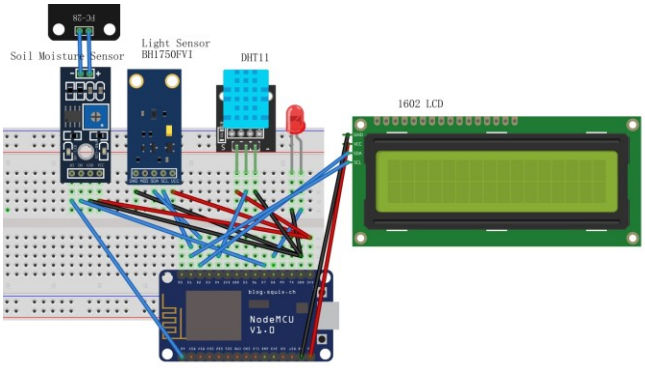


Figure 8: Circuit Diagram for Monitoring sensor

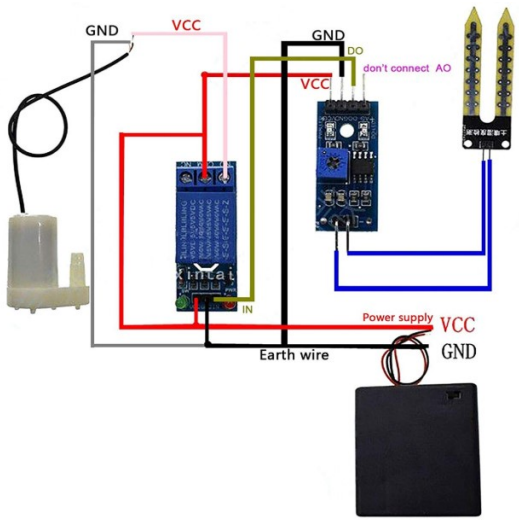


Figure 9: Circuit Diagram for Watering Plants

ESP8266-12E	Light Sensor
3.3V	VCC
GND	GND
D1	SCL
D2	SDA

ESP8266-12E	DHT11
3.3V	VCC
GND	GND
D4	OUT

ESP8266-12E	1602 LCD
5V	VCC
GND	GND
D1	SCL
D2	SDA

ESP8266-12E	BH1750FVI
3.3V	VCC
GND	GND
D7	DO
AO	AO

3.2 Software Design

Software is an integral part of this system because it is the only communication channel between each module. The interaction between each module is done through various libraries. The libraries are the package bundle providing different sets of classes and functions for individual operations. The major components and libraries are explained below:

3.2.1 BH1750FVI Light Sensor

To connect the sensor with the NodeMCU ESP8266 module we are using Wire library to transmit the data into the channel.

First, before requesting or sending data to any other medium the wire module provides a function Wire. available () to check whether the wire is available for transmission. If you get s successful response, then we can send or receive data to medium with the functions Wire.read().

The data is transmitted in 'L' mode with 7-bit addressing mechanism. The general code for writing data to channel is:

```
Wire.beginTransmission(ADDR_LightSensor);
Wire.write(0b00000111);
Wire.endTransmission();
```

We use VCC - 3.3V, GND – GND, DO-D7 AO-AO connections to connect them to the microcontroller board.

3.2.2 DHT11 Temperature Sensor

This temperature and humidity sensor is directly connected to our microcontroller ESP2866 communicating through two main libraries:

```
RPi.GPIO
dht11
```

The DHT11 link is using this VCC - 3.3V, GND - GND, OUT - D4 connection. The input pin is set to 2 to read sensor data using a method like dht11.read() and digitalRead(DHT11PIN). we are reading the value of bit1-40 from the response signal. Now the first 8 bits are used for humidity reading and later from 16-23 bits are used for temperature reading converting them into specific value format.

3.2.3 Soil Moisture Sensor with Probe

This sensor identifies the dielectric volume content of the soil and generates the analog output.
It has two main threshold values HIGH and LOW. whenever the sensor is in the air the reading values returns 1024 which indicates extreme dryness. The high reading values mean more dryness and vice versa.

We are using this sensor for two purposes. One is to get the reading of the soil moisture through the probe and another is sending the signal based on the output to the water pumping circuit to water the plants.

We have set the pins for moisture at A0 and for DO at 13 number on the microcontroller. Here 13 number indicates D7 port of ESP8266. The serial ports return the measurement data using function analogRead(Moisture). After reading we send data to alarm the system through the LED.

```
if(digitalRead(DO) == 1 ){
   digitalWrite(LED, HIGH);   // light LED
}else{
   digitalWrite(LED, LOW);
}
```

3.2.4 1602 LCD display

The LCD displays the overall reading of all the sensors. We have used the LiquidCrystal_I2C library with microcontroller to control the LCD display. we have set the LCD address to 0x27 which has 16 characters each row and 2 line display**.** The addressing scheme is 'L' mode 7bit addressing mechanism.

```
Wire.beginTransmission(ADDR_1602LCD);
lcd.init();
lcd.backlight();
lcd.setCursor(0,0);
```

Using Wire and LCD module of liquid crystal we display the data on the screen.

3.2.5 ThinkSpeak Analytics platform

We are uploading the sensing data on this analytics platform every 5 seconds to generate real-time data analytics charts. We are uploading the data through REST Apis passing a private token and secret channel identifier. Using Wi-Fi client we are uploading the sensor data on the cloud.

```
client.connect(host, httpPort)
while(client.available()){
  String line = client.readStringUntil('\r');
  Serial.print(line);
}
```

3.2.6 Relay Module

Relay Module we are connecting with the moisture sensor to receive the sensor analog data identifying the threshold value and upon that providing the signal to the water pump for automatic watering.

We have set pin 0 of the microcontroller board to the relay module. We have set the max dryness threshold to 700 to water the plants when above that. We will be using a digital signal to send the signal to the submersible water pump.

```
if(tempSoilMoistur >= MAXDRYNESS)
 {
  Serial.println("Soil dry start watering");
  digitalWrite(WATERPIN,HIGH);
  delay(WATERDELAY);
  digitalWrite(WATERPIN,LOW);
  delay(WATERPOSTDELAY);
 }
```

4 Testing and Verification

We performed unit testing of all our circuit sensors to ensure that components are working as expected. Temperature collected from the sensor was cross verified with the smartphone weather app. Readings were matching with an error rate of ± 2 °C. Humidity sensor readings were matched with the readings from Google engine based on current location. Soil moisture readings were recorded for multiple soil types like very dry sand, dry soil, moderately humid soil, and extremely humid soil. Reading from soil moisture sensor decreased as moisture level in the soil increased as we are using electrode probes and sensors which measures the resistance between them. So as soil water content increases, conductivity increases and resistance decreases leading to lower readings. Reading from light sensors were recorded in different lighting conditions like a dark room, under sunlight, normal room lighting, bright lights around circuits. Variations in readings based on lighting were found satisfactory.

Integration testing was performed by connecting all modules of the circuit and power supply. Readings from all the sensors are displayed in LCD screen connected to the circuit. On modifying the testing conditions LCD screen refreshes and displays new output values in every 5 seconds as configured. Also, data is uploaded in real time (every 5 seconds) to cloud using WiFi module. Data which is displayed in the ThingSpeak visualization dashboard matched with the LCD display values. We tested the module by setting different time duration for data read and upload and decided to keep 5 seconds as results were optimal. The automatic watering module was configured to pump water when soil sensor readings go above 700. We tested this with different types of soils and results were as expected.

Sr No.	Components	Notes
1	BreadBoard(2)	2.2" x 3.4"
2	NodeMCU 1.0 with ESP8266	Single board microcontroller
3	DHT11 Temperature Sensor	measures the temperature
4	Relay module	An On-Off switch that allows the signal to pass through the pump.
5	1602 LCD Display	Displays the temperature, soil moisture, humidity and light intensity

6	BH1750FVI Light Sensor	Measures the light intensity
7	Soil Moisture Sensor with Probe	Measures the moisture content in the soil
8	Plastic Battery Storage Case Holder and AA battery	Supplies power to the water pump
9	Water Pump	Automatically discharges water to the soil
10	Jumper Cables	Connects the components
11	USB power cable	Connects the microcontroller with the power supply
12	Vinyl Tubing	0.5m
13	LED light	Alarms when the soil moisture is low

Table 1: Bill of Materials

4.1 Results

For data visualization, we have used ThingSpeak, a free IoT Analytics platform service application by Mathworks. Four output parameter collected from sensors is displayed in four visualization charts as a continuous line graph (Figure).

Figure 10: Think Speak Analytics Result

The same output is displayed on the LCD screen connected to the circuit(Figure) and on the Arduino IDE terminal(Figure).

Figure 11: LCD Output of the sensor data

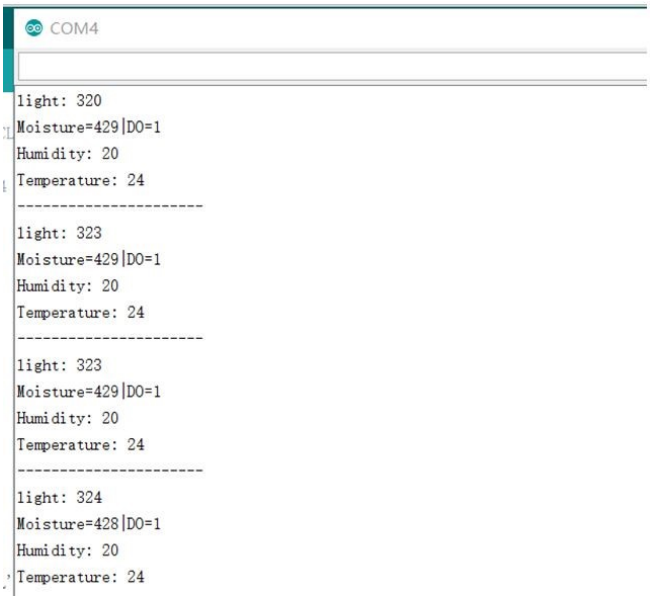


Figure 12: Port Output in Arduino IDE

5 Applications and Challenges Faced

The main application of this project is to monitor

environmental parameters that play a major role in plant growth and automatically water plant based on programmed conditions to nurture it without much human intervention. This is a part of home automation making life more easy for frequent travelers and plant lovers. This solution can ideally be used in offices, apartments, public parks to automatically water plants especially in hot summer days.

Initially choice of microcontroller and sensors were crucial to decide as we wanted to develop a low-cost module with compact size. We decided on the present components after some research and study based on features needed. Deciding the feature-rich and free analytics cloud service was another challenge we faced. Since the module we developed is not soldered end to end we faced connection issues which caused readings to jump quite a times.

6 Conclusion and Future Work

We have built a cost-effective plant monitoring system using IoT that can help people to monitor temperature, humidity, soil moisture and light intensity around plants and automatically water the plants when needed in their absence. Data collected by the sensors are displayed on the LCD display connected to the circuit. Data is also uploaded to cloud storage and displayed as visualization dashboard which can be accessed from anywhere at any time using any of the internet-enabled devices. The compact size of the circuit and low-cost deployment makes it an efficient and feasible solution as a part of home automation.

Future work can include utilizing output from the light sensor to automatically adjust light intensity around indoor plants. Data from temperature sensor can be used to control the room temperature to customize the model for sensitive plants which requires specific environmental conditions for growth. The model can be programmed to pump a specific amount of water depending on the quantity of soil in flower pot or size of the plant. Additionally, with slight modifications in the present circuit, sensors data can be used for other home automation like room temperature control and automatic room lighting system.

7 Acknowledgement

Prof. Ammar Rayes helped us to understand the principles, process, components, concepts, and protocols behind IoT framework. This knowledge helped us in implementing this project successfully.

8 References

1. Cnxsoft, 'NodeMCU is both a Breadboard-Friendly ESP8266 Wi-Fi Board and an LUA based Firmware', 2015. [Online].Available:https://www.cnx-software.com/2015/04/18/nodemcu-is-both-a-breadboard-friendly-esp8266-wi-fi-board-and-a-lua-based-firmware/
2. M. Rawashdeh, 'BH1750 Digital Light Sensor'.[Online].Available:https://www.instructables.com/id/BH1750-Digital-Light-Sensor/
3. 'Liquidcrystal Documentation - Read The Docs', [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/arduinoliquidcrystal/latest/arduinoliquidcrystal.pdf.
4. 'Arduino and Soil Moisture Sensor - Interfacing', [Online]. Available: http://www.circuitstoday.com/arduino-soil-moisture-sensor
5. 'Guide for Relay Module with Arduino'.[Online].Available:https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/
6. "Specification for LCD Module". [Online]. Available:https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf
7. '5V relay module with Status LED, diode and transistor trigger'. [Online]. Available: https://www.botshop.co.za/product/5v-relay-module-with-status-led-diode-and-transistor-trigger/
8. nodemcu-flasher [Online]. Available: https://github.com/nodemcu/nodemcu-flasher

9 Code

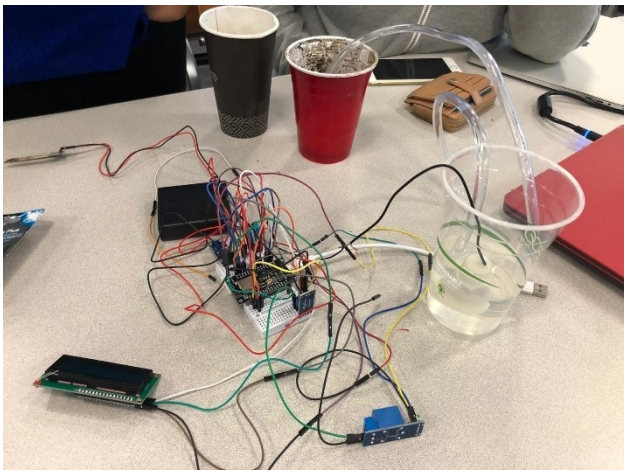
```
void showOnLCD1602(){
  Wire.beginTransmission(ADDR_1602LCD);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);

  String line1 = "sm:";
  line1 += tempSoilMoistur;
  line1 += " tem:";
  line1 += tempTemp;

  String line2 = "li:";
  line2 += tempLight;
  line2 += " hum:";
  line2 += tempHum;
  Serial.println(line1);
  lcd.print(line1);
  lcd.setCursor(0,1);
  lcd.print(line2);
  Wire.endTransmission();
}
void readLight(){
   // reset
  Wire.beginTransmission(ADDR_LightSensor);
  Wire.write(0b00000111);
  Wire.endTransmission();

  Wire.beginTransmission(ADDR_LightSensor);
  Wire.write(0b00100000);
  Wire.endTransmission();
  // typical read delay 120ms
  delay(120);
  Wire.requestFrom(ADDR_LightSensor, 2); // 2byte every time
  for (tempLight = 0; Wire.available() >= 1; ) {
    char c = Wire.read();
    //Serial.println(c, HEX);
    tempLight = (tempLight << 8) + (c & 0xFF);
  }
  tempLight = tempLight / 1.2;
  Serial.print("light: ");
  Serial.println(tempLight);
}


//BH1750FVI Setting
#define ADDR_LightSensor 0b0100011
//1602 Setting
#define ADDR_1602LCD 0b0100111  //
LiquidCrystal_I2C lcd(0x27,16,2); /
//SoliMoisturesensor Setting
#define Moisture A0  //Define the A
#define DO 13          //ESP8266 D7
#define LED 15        // ESP8266 D8
//DHT11 Setting
#define DHT11PIN 2     // ESP8266 D4
#define WATERPIN 0
#define MAXDRYNESS 700
#define WATERDELAY 750
#define WATERPOSTDELAY 5000
```

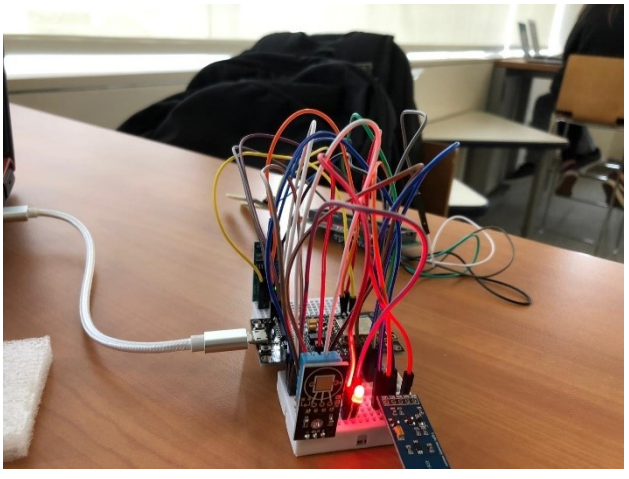
```
void readDHT11(){
  int j;
  unsigned int loopCnt;
  int chr[40] = {0};
  unsigned long time1;
bgn:
  pinMode(DHT11PIN, OUTPUT);
  digitalWrite(DHT11PIN, LOW);
  delay(20);
  digitalWrite(DHT11PIN, HIGH);
  delayMicroseconds(40);
  digitalWrite(DHT11PIN, LOW);
  //Set interface mode 2: input
  pinMode(DHT11PIN, INPUT);
  //High level response signal
  loopCnt = 10000;
  while (digitalRead(DHT11PIN) != HIGH){
    if (loopCnt-- == 0){
      //If don't return to high level for a long time, output a prompt and start over
      Serial.println("HIGH");
      goto bgn;
    }
  }
  //Low level response signal
  loopCnt = 30000;
  while (digitalRead(DHT11PIN) != LOW){
    if (loopCnt-- == 0){
      //If don't return low for a long time, output a prompt and start over
      Serial.println("LOW");
      goto bgn;

void readSoilMoisture(){
  //Serial port returns measurement data
  tempSoilMoistur = analogRead(Moisture);
  Serial.print("Moisture=");
  Serial.print(tempSoilMoistur);//read AO data
  Serial.print("|DO=");
  Serial.println(digitalRead(DO));//read DO data
  if(digitalRead(DO) == 1 ){
     digitalWrite(LED, HIGH);   // light LED
  }else{
     digitalWrite(LED, LOW);
  }
  if(tempSoilMoistur >= MAXDRYNESS)
  {
    Serial.println("Soil dry start watering");
    digitalWrite(WATERPIN,HIGH);
    delay(WATERDELAY);
    digitalWrite(WATERPIN,LOW);
    delay(WATERPOSTDELAY);
  }
}
```

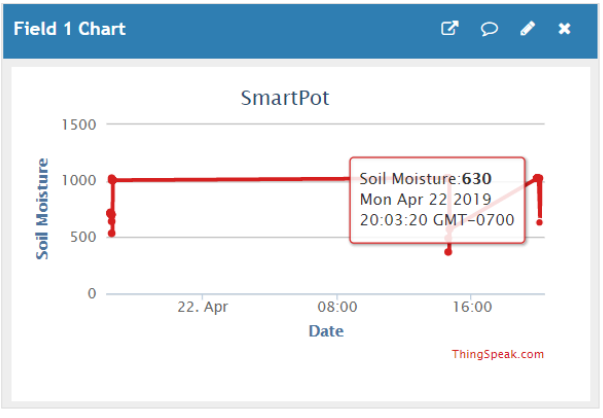
Appendix

1. Configuration Steps

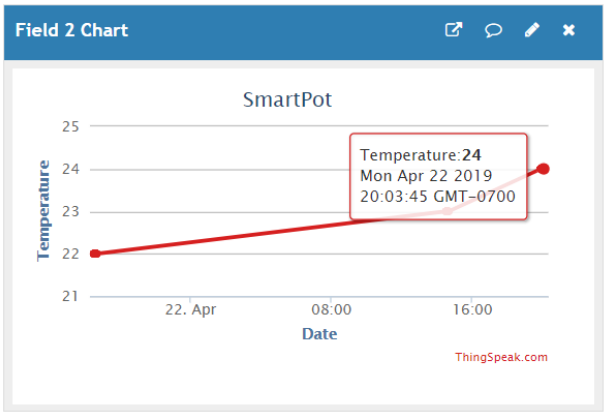
2. Install USB to the serial driver based on the operating system.
3. Connect the NodeMCU microcontroller with the laptop using a usb cable. Connection can be verified under Device Manager.
4. Install the Arduino IDE based on the operating system.
5. Import and install packages for ESP8266 module from the URL:http://arduino.esp8266.com/stable/package_esp8266com_index.json
6. Select NodeMCU 1.0 (ESP-12E Module) as a board from Tools and select USB port to which NodeMCU is connected.
7. Burn firmware into ESP8266 [8]
8. Connect all the components of the module.
9. Create a free account in https://thingspeak.com which is IoT Analytics platform from MathWorks.
10. Create a new channel and modify the privacy and dashboard settings as required. We have configured four charts for four parameters sensed in our project.
11. Generate the Write API and configure them in the Arduino sketch program
12. Program is written in Arduino IDE for working of the modules.
13. Liquid Crystal library is imported for integrating LCD display with Arduino.
14. Wi-Fi credentials are configured to upload data to cloud storage.
15. Verify and upload the project from the IDE to burn software to ESP8266.

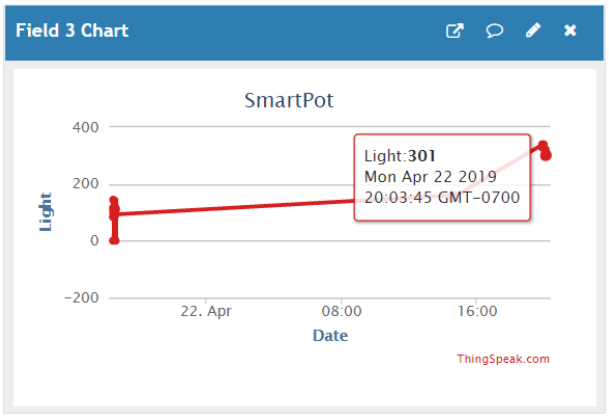


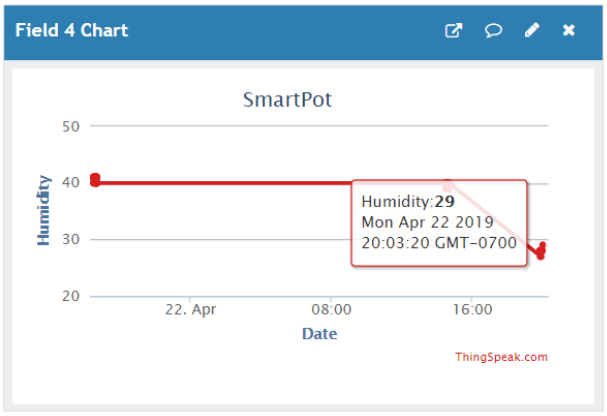




2. Application Snapshot


Field 2 Chart
SmartPot
Temperature
25
24
23
22
21
Temperature:24
Mon Apr 22 2019
20:03:45 GMT-0700
22. Apr
08:00
16:00
Date
ThingSpeak.com


Field 3 Chart
SmartPot
Light
400
200
0
-200
Light:301
Mon Apr 22 2019
20:03:45 GMT-0700
22. Apr
08:00
16:00
Date
ThingSpeak.com


Field 4 Chart
SmartPot
Humidity
50
40
30
20
Humidity:29
Mon Apr 22 2019
20:03:20 GMT-0700
22. Apr
08:00
16:00
Date
ThingSpeak.com