ABSTRACT

Weather forecast these days is unpredictable to be exact because the climate changes drastically over weather. In cause of that, Weather Reporting System is mostly used to monitor the continuously changing climatic and weather conditions over controlled areas likes house, industry, agriculture and etc. in real time monitoring. Internet of Things (IoT) platform use is ThingSpeak it should be able in displaying the weather parameters and the information will be visible wherever in the world with two-way microcontroller communication via Wi-Fi hotspots. The condition of some particular place that be reported by satellite weather report system does not give the exact condition. However, the problem occurs when needed the accurate weather report for current time. With weather reporting system all weather parameters sensor will be controlled by ESP8266 microcontroller as the server that will send all the data collected by sensors to the database by ThingSpeak and will visible anywhere in the world. This system will monitor the changes of weather condition happening over the environment and then provides the users fastest way to access the information from anywhere.

ACKNOWLEDGEMENT

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

We would like to express our sincere gratitude and heartfelt thanks to Dr. Naman Garg for their unflinching support and guidance, valuable suggestions and expert advice. Their words of wisdom and expertise in subject matter were of immense help throughout the duration of this project.

We also take the opportunity to thank our Director and all the faculty of School of Computing, IIIT Una for helping us by providing necessary knowledge base and resources.

We would also like to thank our parents and friends for their constant support.

Rhythm Kothari (21138)

TABLE OF CONTENTS

LIST OF ACRONYMS

IoT	Internet of Things
API	Application Programme Interface
DHT	Digital Humidity and Temperature
DO	Digital Output
AO	Analog Output
SDA	Serial Data
SCL	Serial Clock
VCC	Voltage Common Collector
GND	Ground

LIST OF FIGURES

Chapter 3

Chapter 4

Chapter 1

Introduction

Climate plays an important role in human life the unprecedented growth of industries and vehicular traffic have seriously affected the purity of clean air and environment. Satellite weather report system gives condition of present which does not give the exact condition of the particular place. The building sector offers a great potential for the energy savings, where it is necessary to have accurate weather data in the exact location where the building is being built in order to improve the calibration of energy simulation programs, such is the with IIIT Una. By develop a controlling local weather reporting system with ESP8266 and Arduino Uno microcontroller can minimize the error in weather forecast system at exact location.

A precision agriculture and farming can be defined as the art and science of using technology to improve crop production. Even though water is a scarce resource, overall 50% of water is wasted in agriculture due to the improper scheduling of irrigation. In this context, the real-time monitoring of water usage in the fields can prevent misuse of water.

Use of technology in the field of agriculture plays important role in increasing the production as well as in reducing the extra man power efforts, some of the researches tried for betterment of farmers and provides the systems that use technologies which are helpful for increasing the agriculture yield. Difficulty to monitor weather parameters through offline system such as agriculture zone during certain hazardous envy and critical situations where the people need to check manually the weather condition at the places and it will take time unless it is online system.

In the evolving generation of wireless technology, the concept of smart cities and IoT has given a new remark in the world. One such remark leads towards the online smart weather station system. The weather parameters should be able displaying, analysing and monitoring system using ThingSpeak that connect user with internet that visible anywhere in the world. To analyse and monitoring system using ThingSpeak that connect user with internet that visible anywhere in the world. IoT is playing a leading role in providing solutions to many applications with the support of software, internet and embedded systems.

Chapter 2

Review of Literature

The author in [1], presents an IoT-based weather monitoring system. In this research, the environmental parameter can be retrieved through sensors. The author uses a different sensor to scale the various parameter like humidity, temperature, pressure & rain value. The temperature sensor can be used to measure the value of the particular area, room, or any place. The author in this system uses node MCU 8266, and various sensors.

The author [2], implement an IoT-based weather monitoring system, in this research paper, the author describes that how with the help of IoT technology, the weather can be monitored. And which provide the info of climate-changing conditions. With the help of this project, people can be aware of the climate condition changes. It gives an accurate and efficient output and the algorithm as the swarm is used to implement for further improving the accuracy. So, in this project, the author aims to make a weather monitoring with the help of IoT. In this project, the hardware and software are used which makes it easy to implement. In the project, the author uses a different sensor to collect the information of the climate and stored it in the cloud. For this storage, the website www.thingspeak.com is commonly used for Internet of things projects. And from the cloud storage space, it extracts the whole weather data and uploads it to the android mobile application using an API key. Tools which detect the rain drops, is called rain sensor. Once the plague reveals the raindrops on the strips and the voltage is considered from that. And there is no short circuit condition that occurs because water is a bad conductor and the sensor acts like variable resistance. Once a measurement of voltage is completed then the circuit takes the output.

The author [3], in this research paper have cleared the objective to build a system that can be monitored weather parameter by wireless system and IoT. The Sensor station and Weather station will be communicated by hotspot Wi-Fi and it is limited in areas covered but still better in communication via wireless.

The proposed system by **the author in [4]** works on the client-side architecture model. The proposed approach observed various environmental information using multiple sensors. The main aim of our proposed model is to make the system cost-effective, affordable. So that everyone can use it freely. In our proposed system, capturing multiple data from multiple sensors and send all the data to the webpage by HTTP request protocol on the webserver.

Chapter 3

Methodology

We will see all working principle of all types of sensors.

3.1 Block Diagram

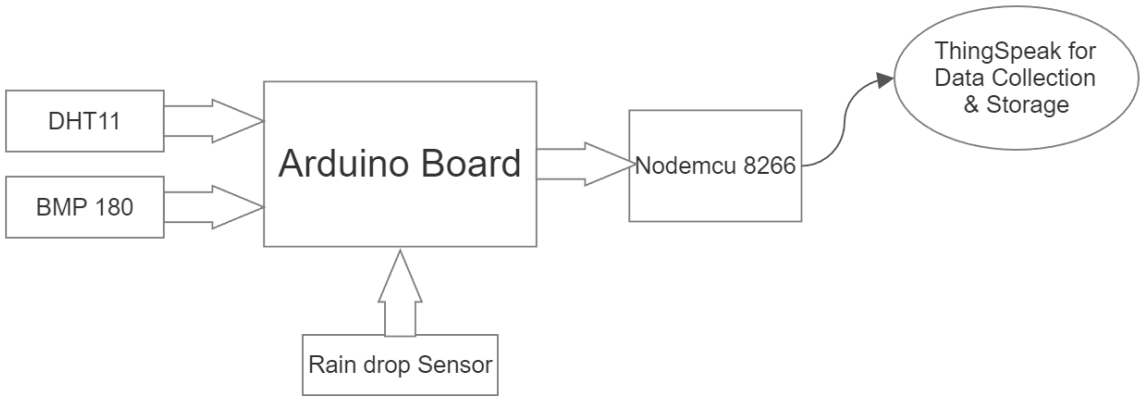

Figure 3.1 : Block Diagram

3.2 Detailed Description

Arduino Uno: The Arduino UNO is an ATmega328p (data sheet)-centred microcontroller device. It has fourteen physical outputs and ports, six of them PMW output. It is possible to use 6 analog inputs and 16 MHz quartz crystal. The moisture sensor consists of two samples of wire that rely on the particular water resistance to sensing the water in contact. The RF transmitter was interfaced with the Arduino design codes on the microscope and the RF transmitter was attached to the microscope.

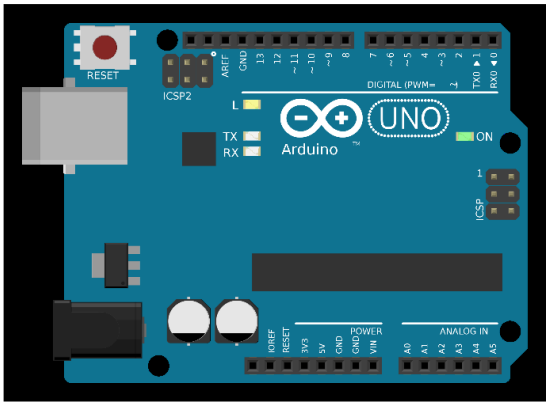

Figure 3.2 : Arduino Uno

Sensors:

DHT11 temperature and humidity module:

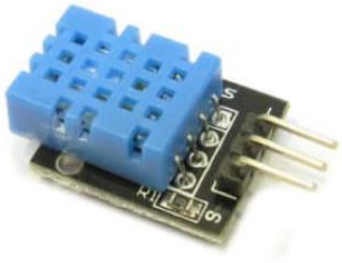

Figure 3.3 : DHT11 Sensor

DHT11 is a digital sensor responsible for collecting temperature and humidity data from your surroundings. It has three terminals namely:

- Vcc
- GND
- Data

Vcc connects to 5V supply, GND connects to GND and data pin connects to D9 of Arduino.

BMP180 Barometric Sensor:

Figure 3.4 : BMP180 Sensor

The above illustrated module is a barometric sensor which is capable of measuring atmospheric data; it can give out data like, atmospheric pressure at ground level, atmospheric pressure at sea level and altitude.

We will be only extracting atmospheric pressure data at ground and sea level to display it on the LCD, but we will be sending only the atmospheric pressure data at ground level to Thingspeak server, which is the relevant data for your locality.

It has the following pins:

- SDA – Serial data.
- SCL – Serial clock.
- Vcc – 3.3V.
- GND – ground.

Care must be taken while connecting the supply to this module as it operates on 3.3V DC and 5V will kill the module.

Rain drop Sensor:

Figure 3.5 : Rain drop Sensor

Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance.

Rain Sensor module allows to measure moisture via analog output pins and it provides a digital output when a threshold of moisture exceeds.

When rain drop present it reduces the resistance because water is a conductor of electricity and presence of water connects nickel lines in parallel so reduces resistance and reduces voltage drop across it.

Nodemcu ESP8266:

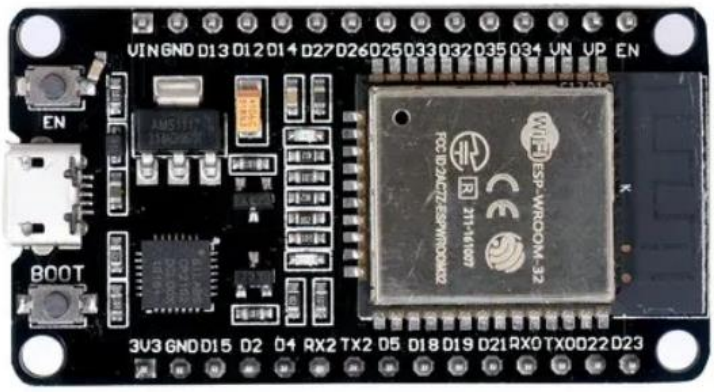

Figure 3.6 : Nodemcu ESP8266

The above illustrated module is called generic ESP8266 which is responsible for connecting the weather monitoring system to internet.

ESP8266 is not a just another ordinary module, it has a full-fledged 32-bit microcontroller which requires a program code to function. We will be using a programmer to upload the code to this ESP8266 module which we will see in the later part of this article. It operates on 3.3V and communicates on serial interface with Arduino.

3.3 Setting up ThingSpeak Account

- Create a new channel and do the following to your Thingspeak channel:

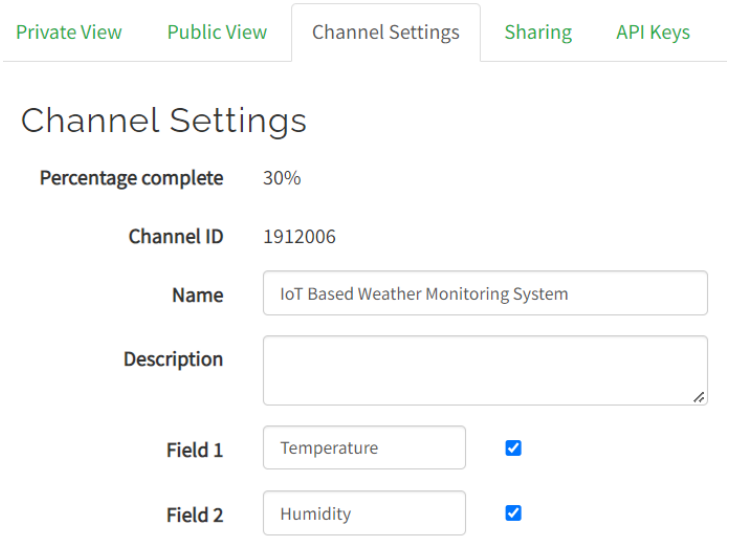

Figure 3.7 : ThingSpeak channel settings

- Scroll down and press save.
- Please take note of your channel ID and you need to copy and insert the ID to the ESP8266 code.
- Now go to API key tab and you will see “write API” and “read API” keys. Write API key is a secret code for writing data to your Thingspeak channel.

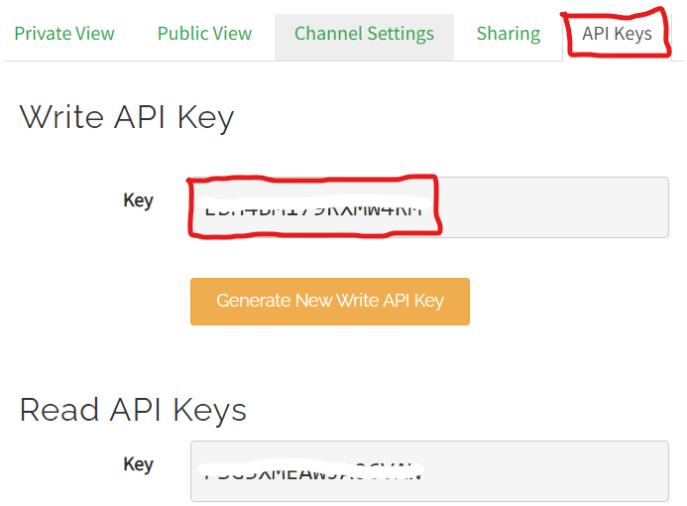

Figure 3.8 : API Key

- You need copy and paste the write API key to the ESP8266 code.

Chapter 4

Result

First the circuit of control unit system have been made that Arduino Uno microcontroller control all weather parameters sensor, that are BMP180 (Temperature and Pressure) sensor, DHT11 (Temperature and Humidity) and Rain sensor. Then it powered by USB cable also to upload the sketch of coding in ESP8266 microcontroller. The sensor data can be display on serial monitor in Arduino IDE software. ESP8266 will connect with the Wi-Fi hotspot that have applied to this system so that the web server can be create to display all the sensor data.

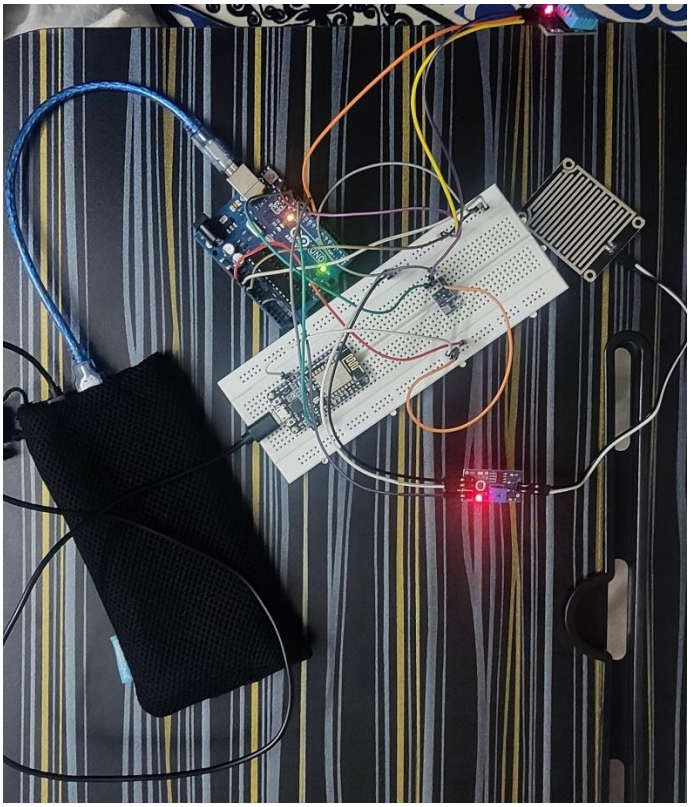

Figure 4.1 : Propose Weather Monitoring System

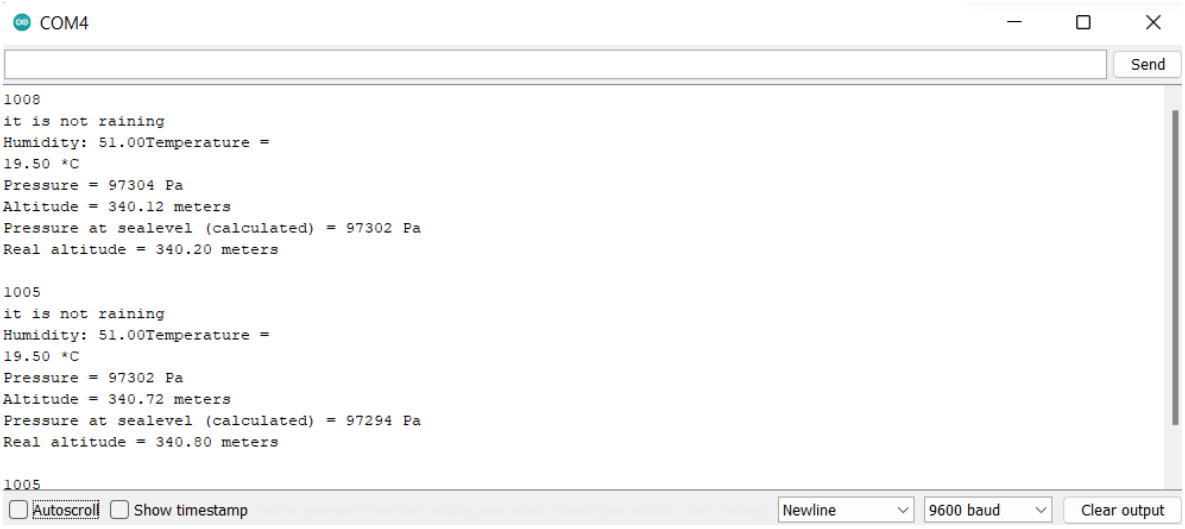

Figure 4.2 : Serial Monitor Output

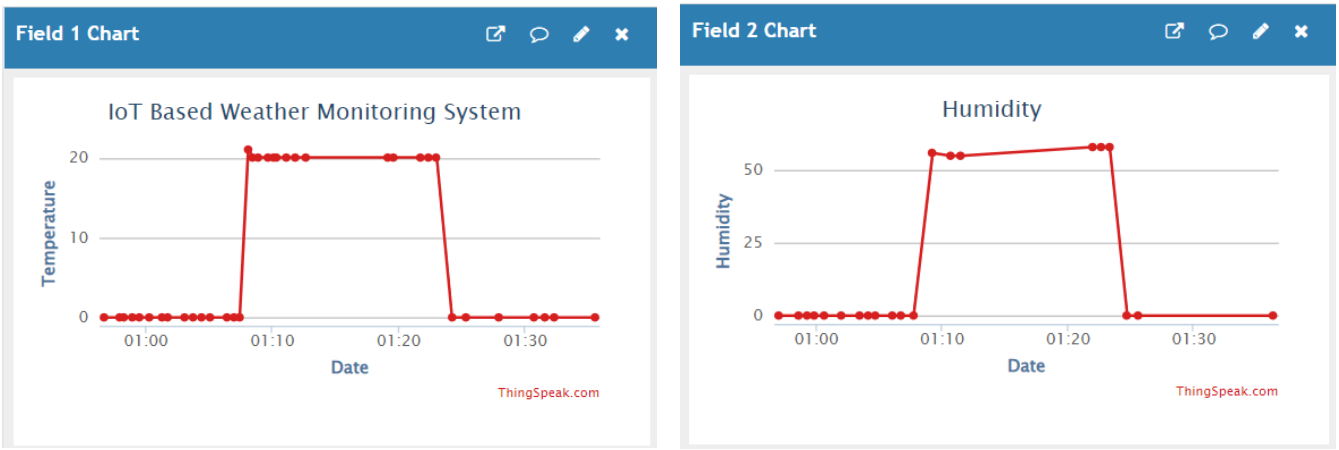

Figure 4.3 : Data collected on ThingSpeak

After that it will read all the sensor value and then send to the cloud data where ThingSpeak has been uses for this. ThingSpeak will stored the sensor value and display that data to the channel create there. The user can check the weather parameter via ThingSpeak websites. The data has collected from the reading of ESP8266 for all sensor and send that data to the ThingSpeak as the results of this project objective.

Appendices

Appendix A

Code Attachments

The following is the partial / subset of the code. Code of some module(s) have been wilfully supressed.

A.1 Arduino Code

```
#include <DHT.h>
#include<Wire.h>
#include <SoftwareSerial.h>
#include <Adafruit_BMP085.h>
#define seaLevelPressure_hPa 1013.25
#define Type DHT11
SoftwareSerial mySerial(10, 11);
void ReadBMP(void);
void ReadDHT(void);
void ReadRain(void);
void send_data(void);
int sensePin = 9;
DHT HT(sensePin,Type);
float humidity;
Adafruit_BMP085 bmp;

void setup(){
  mySerial.begin(115200);
  Serial.begin(9600);
  HT.begin();
  delay(2000);
}

void loop(){
  ReadRain();
  ReadDHT();
  ReadBMP();
  send_data();
}

void ReadDHT(void)
{
  humidity = HT.readHumidity();

  Serial.print("Humidity: ");
  Serial.print(humidity);
  delay(2000);
}

void ReadRain(void)
```

```
{
  int sensorValue = analogRead(A1);
  // print out the value you read:
  Serial.println(sensorValue);
  if (sensorValue<900&&sensorValue>600)
  {
      Serial.println("it is Drizzling!");

   }
   else if(sensorValue<600 && sensorValue>300)
   {
      Serial.println("It is Raining!");
   }
   else if(sensorValue<300)
   {
      Serial.println("It is Raining heavily!");
   }
   else
   {
      Serial.println("it is not Raining");

   }
   delay(2000);
}

void ReadBMP(void)
{

  if (!bmp.begin()) {
  Serial.println("Could not find a valid BMP085 sensor, check wiring!");
  while (1) {}
  }
  else
  {
    Serial.println("Temperature = ");
    Serial.print(bmp.readTemperature());
    Serial.println(" *C");

    Serial.print("Pressure = ");
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");

    Serial.print("Altitude = ");
    Serial.print(bmp.readAltitude());
    Serial.println(" meters");

    Serial.print("Pressure at sealevel (calculated) = ");
    Serial.print(bmp.readSealevelPressure());
    Serial.println(" Pa");

    Serial.print("Real altitude = ");
    Serial.print(bmp.readAltitude(seaLevelPressure_hPa * 100));
    Serial.println(" meters");

    Serial.println();
    delay(500);
  }
```

```
  delay(2000);
}

void send_data()
{
  int sensorValue = analogRead(A1);
  mySerial.print('*'); // Starting char
  mySerial.print(bmp.readTemperature(), 0); //2 digit data
  mySerial.print(HT.readHumidity(), 0); //2 digit data
  mySerial.print(bmp.readPressure()); //5 digit data
  mySerial.print(bmp.readAltitude(seaLevelPressure_hPa * 100) * 100); //5
digit data
  mySerial.print(sensorValue); //4 digit data
  mySerial.println('#'); // Ending char
}
```

A.2 Nodemcu ESP8266 code

```
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>

//------- WI-FI details ----------//
char ssid[] = ""; //SSID here
char pass[] = ""; // Password here
//--------------------------------//

//----------- Channel details ----------------//
unsigned long Channel_ID = 1912006; // Your Channel ID
const char * myWriteAPIKey = "EBH4BM179RXMW4RM"; //Your write API key
//--------------------------------------------//

const int Field_Number_1 = 1;
const int Field_Number_2 = 2;
const int Field_Number_3 = 3;
const int Field_Number_4 = 4;
const int Field_Number_5 = 5;
String value = "";
int value_1 = 0, value_2 = 0, value_3 = 0, value_4 = 0, value_5 = 0;
WiFiClient  client;

void setup()
{
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
  internet();
}

void loop()
{
  internet();
  if (Serial.available() > 0)
  {
    delay(100);
    while (Serial.available() > 0)
```

```
    {
      value = Serial.readString();
      if (value[0] == '*')
      {
        if (value[19] == '#')
        {
          value_1 = ((value[1] - 0x30) * 10 + (value[2] - 0x30));
          value_2 = ((value[3] - 0x30) * 10 + (value[4] - 0x30));
          value_3 = ((value[5] - 0x30) * 10000 + (value[6] - 0x30) * 1000
+ (value[7] - 0x30) * 100 + (value[8] - 0x30) * 10 + (value[9] - 0x30));
          value_4 = ((value[10] - 0x30) * 10000 + (value[11] - 0x30) *
1000 + (value[12] - 0x30) * 100 + (value[13] - 0x30) * 10 + (value[14] -
0x30));
          value_5 = ((value[15] - 0x30) * 1000 + (value[16] - 0x30) * 100
+ (value[17] - 0x30) * 10 + (value[18] - 0x30)););
        }
      }
    }
  }
  upload();
}

void internet()
{
  if (WiFi.status() != WL_CONNECTED)
  {
    while (WiFi.status() != WL_CONNECTED)
    {
      WiFi.begin(ssid, pass);
      delay(5000);
    }
  }
}

void upload()
{
  ThingSpeak.writeField(Channel_ID,        Field_Number_1,        value_1,
myWriteAPIKey);
  delay(15000);
  ThingSpeak.writeField(Channel_ID,        Field_Number_2,        value_2,
myWriteAPIKey);
  delay(15000);
  ThingSpeak.writeField(Channel_ID,        Field_Number_3,        value_3,
myWriteAPIKey);
  delay(15000);
  ThingSpeak.writeField(Channel_ID,        Field_Number_4,        value_4,
myWriteAPIKey);
  delay(15000);
  ThingSpeak.writeField(Channel_ID,        Field_Number_5,        value_5,
myWriteAPIKey);
  delay(15000);
  value = "";
}
```

CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL3 SL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL L306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL30 SL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL3 L306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL30 SL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL L306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL3 SL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306
CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306 CSL306