**Introduction:**

In this IoT project we are going to Monitor Humidity and Temperature over the internet using ThingSpeak where we will show the current Humidity & Temperature data over the Internet using the ThingSpeak server. It is accomplished by the data communications between Arduino, DHT11 Sensor Module, ESP8266 WIFI module and LCD. Celsius scale thermometer and percentage scale humidity meter displays the ambient temperature and humidity through a LCD display and also sends it to ThingSpeak server for live monitoring from anywhere in the world. Moreover, we can read a data from any sensor and analyse it graphically from anywhere in the world. It is one of the leading platform for Internet of Things. It can transfer a data to IoT cloud.

**Equipments:**

Arduino Uno

ESP8266-01

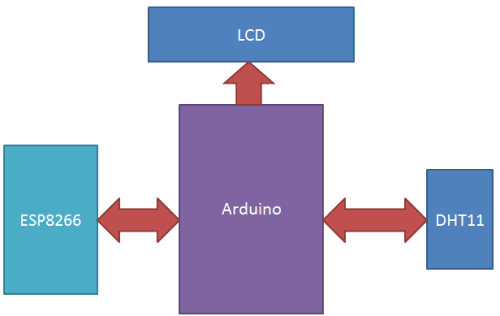
DHT11

AMS1117-3.3V

9V battery

**Working and ThingSpeak Setup:**

This IoT based project having four sections, firstly Humidity and Temperature sensor DHT11 senses the Humidity and Temperature Data. Secondly Arduino Uno extracts the DHT11 sensor’s data as suitable number in percentage and Celsius scale, and sends it to Wi-Fi module. Thirdly Wi-Fi module ESP8266 sends the data to ThingSpeak’s sever. And finally ThingSpeak analyses the data and shows it in a graph form. Optional LCD is also used to display the temperature and humidity. For reset the module pull down the RST pin of ESP8266 to Gnd.ESP8266 have 2 GPIO pins GPIO 0 and GPIO 2. ThingSpeak provides very good tool for IoT based projects for Arduino. By using ThingSpeak site, we can monitor our data over the Internet from anywhere, and we can also control our system over the internet, using the channels and webpages provided by ThingSpeak. ThingSpeak collects the data from the sensors, ‘Analyze and Visualize’ the data and ‘Acts’ by triggering a reaction.



**Circuit Description:**

Connections for this ThingSpeak temperature and humidity monitoring project are very simple.

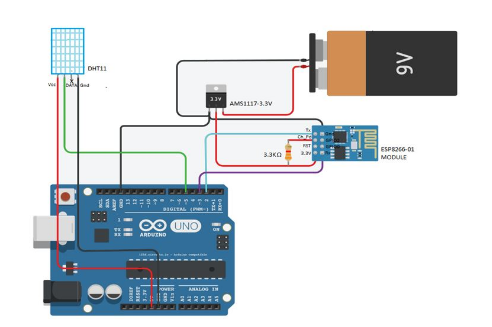


Figure: Circuit diagram for monitoring Humidity and Temperature in IOT cloud

First make the connection as shown in figure. The second pin is of DHT11 is a data pin, it can send a temperature and humidity value to the 5th pin of Arduino Uno.1st and 4th pin of DHT11 is a Vcc and Gnd and 3rd pin is no connection. The Arduino Uno process a temperature and humidity value and send it to a ESP8266 WiFi module. The Tx and Rx pin of ESP8266 is connected to the 2nd (Rx) and 3rd (Tx) of Arduino Uno. Make sure that input voltage of ESP8266 must be 3.3V, not a 5V (otherwise it would damage a device). For that, we are using AMS1117 Voltage regulator circuit. It can regulate a voltage from 9V to 3.3V and will give it to Vcc pin of ESP8266.The Ch\_Pd is a chip enable pin of ESP8266 and should be pullup to 3.3V through 3.3KΩ resistor. For reset the module pull down the RST pin of ESP8266 to Gnd.ESP8266 have 2 GPIO pins GPIO 0 and GPIO 2.

**Programming Part:**

Programming part of this project plays a very important role to perform all the operations.

#include "DHT.h"

#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 13, d7 = 10;

LiquidCrystal lcd(rs,en , d4, d5, d6, d7);

int buzz = 9;

#define dhtDataPin 7 // Digital Pin 7

#define dhtModel DHT11 // We are Using DHT11

String WriteApiKey = "NSVLA0XPANHGPHDY"; // Edit this API key according to your Account

String Wifi\_Name = "Sirus"; // Edit Host\_Name

String wifiPassword = "55555221"; // Edit Password

SoftwareSerial ser(2, 3); // RX, TX

DHT dht(dhtDataPin, dhtModel); // Initialising Pin and Type of DHT

void setup() {

Serial.begin(115200); // enable software serial

ser.begin(115200); // reset ESP8266

ser.println("AT+RST"); // Resetting ESP8266

dht.begin(); // Enabling DHT11

char k='"';

String s1 = "AT+CWJAP";

s1+= "=";

s1+= k;

s1+= Wifi\_Name;

s1+= k;

s1+= ",";

s1+= k;

s1+= wifiPassword;

s1+= k;

ser.println(s1); // Connecting ESP8266 to your WiFi Router

lcd.begin(16, 2);

lcd.print("Temperature:" );

lcd.setCursor(0,1);

lcd.print("Humidity:" ); }

void loop() {

int humidity = dht.readHumidity(); // Reading Humidity Value

int temperature = dht.readTemperature(); // Reading Temperature Value

String state1=String(humidity); // Converting them to string

String state2=String(temperature); // as to send it through URL

Serial.print("Temperature = ");

Serial.println(temperature);

Serial.print("Humidity = ");

Serial.println(humidity);

lcd.setCursor(13, 0);

if (isnan(humidity) || isnan(temperature)) {

lcd.print("Failed to Read");

return;

}

lcd.print(temperature);

lcd.setCursor(10,1);

lcd.print(humidity);

pinMode(buzz, OUTPUT);

if(temperature >=32 || humidity >=65) {

int i=0;

do { i++;

tone(buzz, 1000);

delay(200);

noTone(buzz);

delay(200);

}while(i<2); }

String s2 = "AT+CIPSTART=\"TCP\",\""; // Establishing TCP connection

s2 += "184.106.153.149"; // api.thingspeak.com

s2 += "\",80"; // port 80

ser.println(s2);

if(ser.find((char\*)"Error")){

return; }

String s3 = "GET /update?api\_key="; s3 += WriteApiKey;

s3 +="&field1=";

s3 += String(state1);

s3 +="&field2=";

s3 += String(state2);

s3 += "\r\n\r\n";

s2 = "AT+CIPSEND=";

s2 += String(s3.length()); // Total Length of data

ser.println(s2);

if(ser.find((char\*)">")){

ser.print(s3);}

else{

ser.println("AT+CIPCLOSE"); // closing connection

}

}

**Discussion:**

This project presents real-time monitoring of temperature and humidity of surround-ing environment. The sensed data is sent through Wi-Fi to the cloud where both real-time data and its graphical analyses can be viewed. Working of this project is based on single wire serial communication for fetching data from DHT11. First Arduino sends a start signal to DHT module and then DHT gives a response signal with containing data. Arduino collects and extracts the data in two parts first is humidity and second is temperature and then send it to LCD and ThingSpeak server. This system can be extended to implement a home automation system where the monitored values of temperature and humidity can be used to trigger some action and control the devices for heating or cooling via the mobile application. This system is a crucial step in understanding the IoT applications development and implementation and serves as a building block for a number of useful innovations in this direction.