Home-Made Weather Station

This is a project overview for module S3-EE1953 - Engineering Design

Department of Electrical Engineering
University of Moratuwa

Table of Contents

Project Description

We introduce a small device using Arduino module to monitor the environmental parameters such as temperature, humidity and air pressure and update the collected data (weather statistics) to the cloud storage.

It can be used

- by farmers for agricultural purposes
- by fishermen for marine fields
- for some utility companies to estimate the demands
- by students for simple practical sessions in school laboratories

Image Reference : https://www.pinterest.com/pin/1151795673420402350/

Project Description (Continued...)

We are planning to implement this system through having the device as a small box. And we can put these boxes in multiple places (each district) and able to monitor the weather all over Sri Lanka.

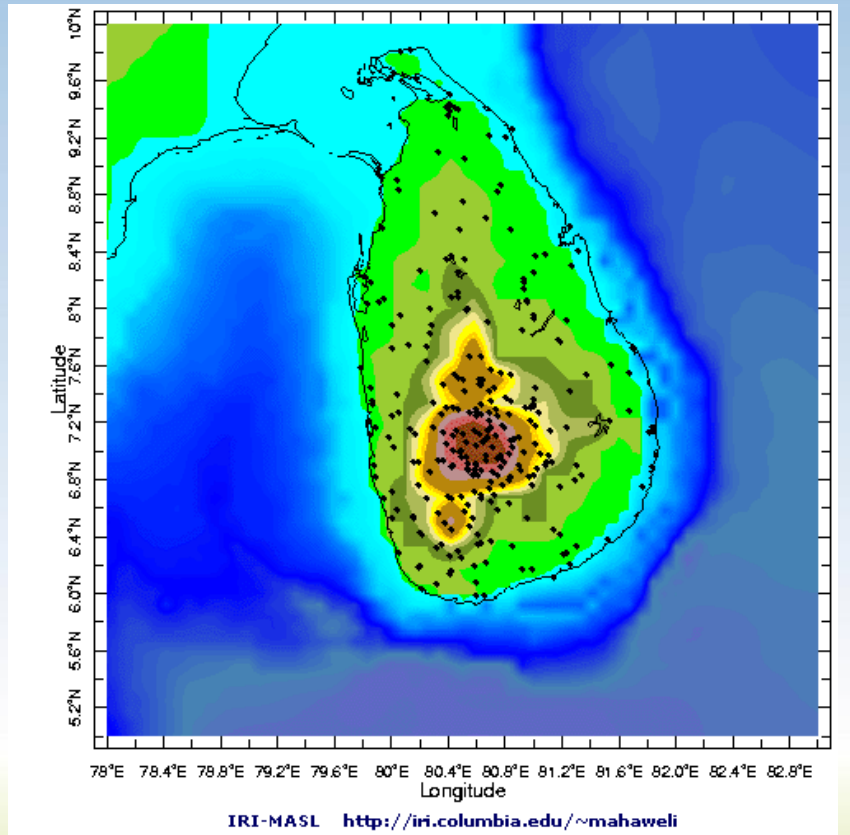


Image reference : https://tropicalclimate.org/~mahaweli/RiverBasinMahaweli/climate.html

Sensors and Components & their Specifications

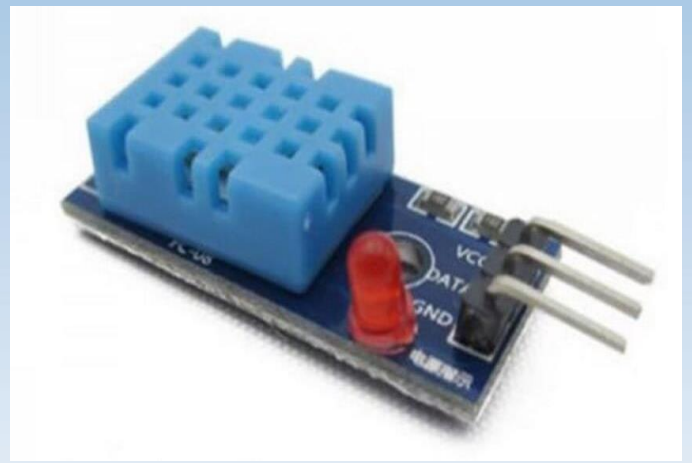


Image Reference : https://www.daraz.lk/products/dht11-humidity-and-temperature-sensor-for-arduino-i104749069.html

Sensor : **Humidity and Temperature Module**
Specification Type : DHT11

Image Reference : https://microchip.lk/product/gy-68-bmp180-digital-barometric-pressure-sensor-board-module/

Sensor : **Digital Barometric Pressure Module**
Specification Type : GY 68 - BMP180

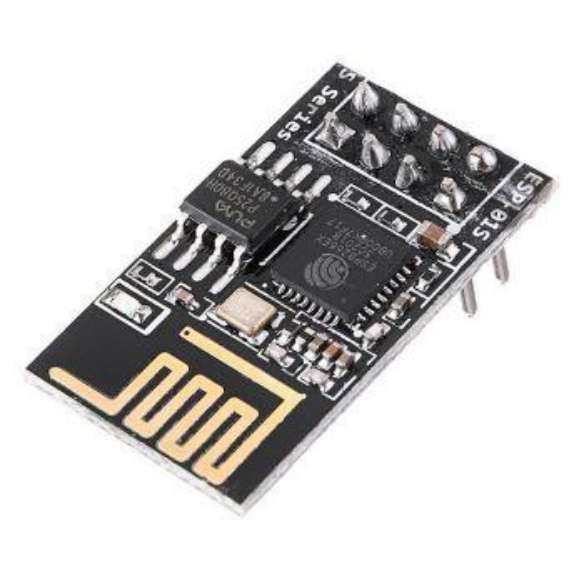
Image Reference : https://www.daraz.lk/products/rain-drop-sensor-module-for-arduino-i104745215.html

Sensor : **Rain Drop Module**
Specification Type : YL 83

Sensors and Components & their Specifications (Continued...)

https://nilambaraelectronics.com/product/mq-135-air-quality-detection-gas-sensor-module/

Sensor : **Air Quality Detection module**
Specification Type : MQ 135



https://grobotronics.com/esp8266-wifi-module.html?sl=en

Sensor : ESP8266 **Wi-Fi module**
Specification Type :
Part Number - ESP8266-01

Software

- Arduino
  - ✓ For handling the sensor data

- Proteus Design Suite
  - ✓ For the automation and simulation of the circuit

- ThingSpeak IoT
  - ✓ To communicate with internet enabled devices.

Image Reference :
https://commons.wikimedia.org/wiki/File:Arduino_Logo.svg

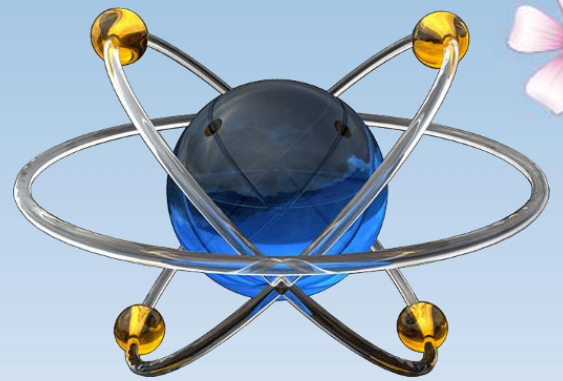
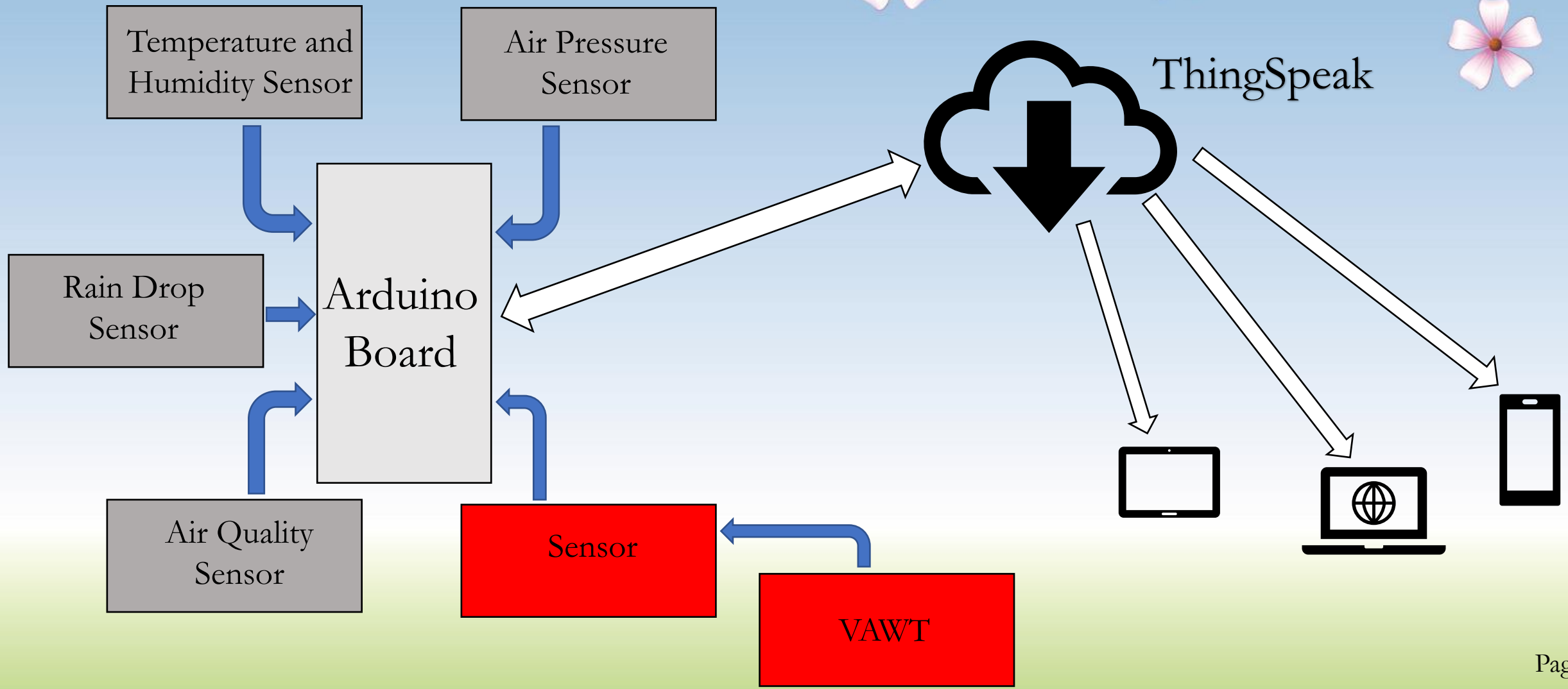


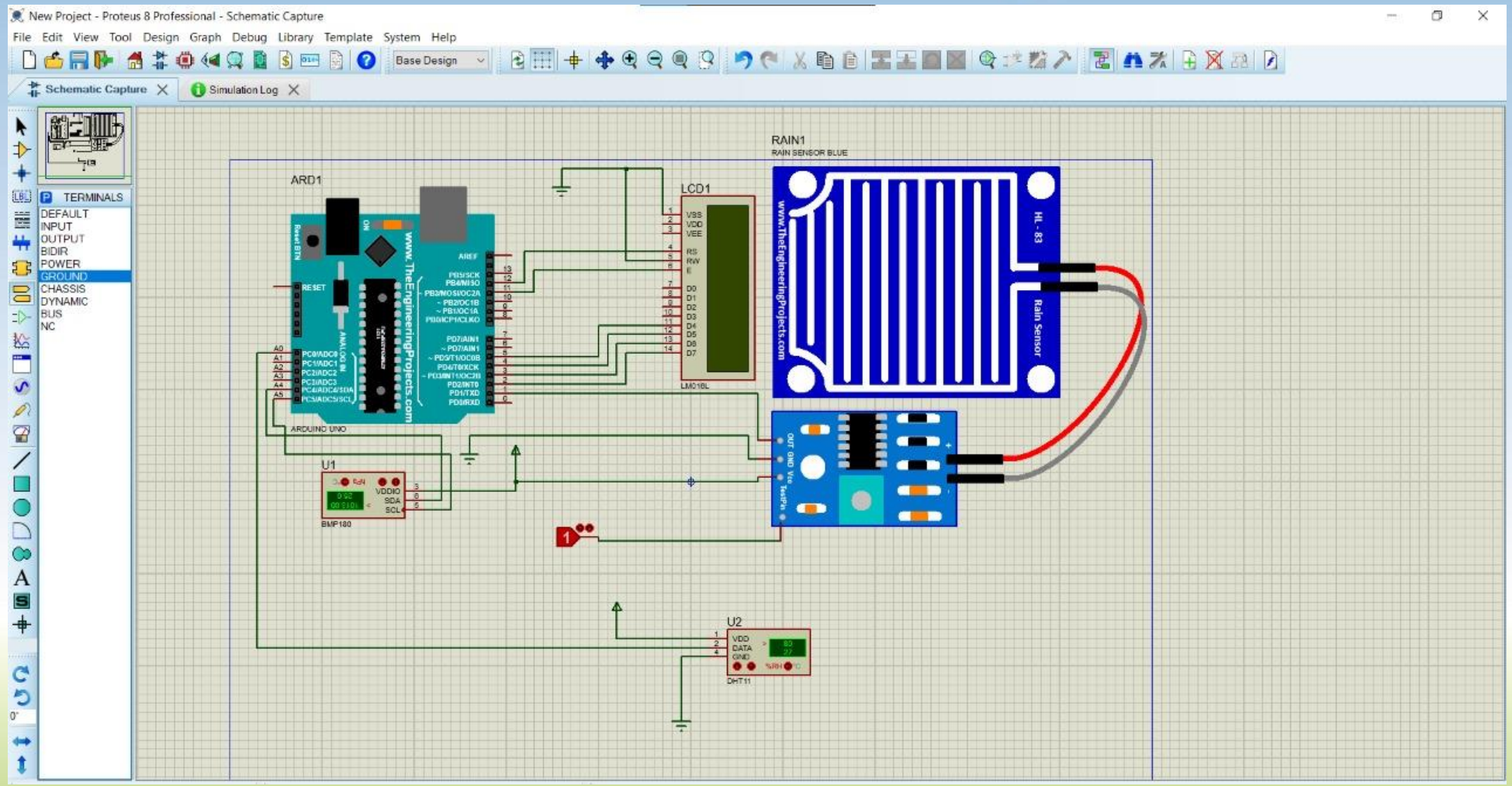
Image Reference :
https://www.labcenter.com/

Image Reference :
https://ww2.mathworks.cn/hardware-support/thingspeak.html

Block Diagram of the Project



Circuit Design of the Project



Note :

The Design at the left is sketched using Proteus 8 Professional Software

Final__ | Arduino 1.8.16

File Edit Sketch Tools Help

Final__ §

```
#include <SoftwareSerial.h>
#include <dht.h>
#define RX 2
#define TX 3
dht DHT;
float humidity;
float temp;
#define dht_apin A0
String AP = "Yathu Cat";       // AP NAME
String PASS = "IamYathu"; // AP PASSWORD
String API = "GVDMM1ZC55MCQDDG";   // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
int valSensor = 1;
const int capteur_D = 0;
#include "TimerOne.h"
unsigned int counter=0;
void docount()  // counts from the speed sensor
{
  counter++;  // increase +1 the counter value
}
int val_analogique;
#include <SFE_BMP180.h>
#include <Wire.h>
SFE_BMP180 pressure;
#define ALTITUDE 8.0
SoftwareSerial esp8266(RX,TX);
//-------------------------------------------------------------------------------------------------------------------------------------------------------------------
void setup() {
  Serial.begin(9600);
  esp8266.begin(115200);
```

Final__ | Arduino 1.8.16

File Edit Sketch Tools Help

Final__ §

```
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
pressure.begin();

}
//-----------------------------------------------------------------------------------------------------------------------------------------
void loop() {

 String getData = "GET /update?api_key="+ API +"&field1="+getTemperatureValue()+"&field2="+getHumidityValue()+"&field3="+getGasvalue()+"&field4="+getpressurevalue()+"&field5="+getrainvalue();
 sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
 esp8266.println(getData);delay(1500);countTrueCommand++;
 sendCommand("AT+CIPCLOSE=0",5,"OK");

}

//-----------------------------------------------------------------------------------------------------------------------------------------
String getTemperatureValue(){

    DHT.read11(dht_apin); //A0,5v,gnd
  float temp=DHT.temperature;

   return String(temp);

}

//-----------------------------------------------------------------------------------------------------------------------------------------
String getHumidityValue(){
    DHT.read11(dht_apin); //A0,5v,gnd
  float humidity=DHT.humidity;

   return String(humidity);
```

Arduino Code - 2

Final__ | Arduino 1.8.16

File Edit Sketch Tools Help

Final__ §

```
}
//-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
String getGasvalue(){

  float sensorvalue = analogRead(A3);

  return String(sensorvalue);
  }
//-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
String getrainvalue(){

  val_analogique=analogRead(A2);

  return String(val_analogique);
}
//-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
String getpressurevalue() {

  char status;
  double T,P,p0,a;
  Serial.println();
  status = pressure.startPressure(4);
  status = pressure.getPressure(P,T);
  return String(P);
}

//-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
```

Final__ | Arduino 1.8.16

File Edit Sketch Tools Help

Final__ §

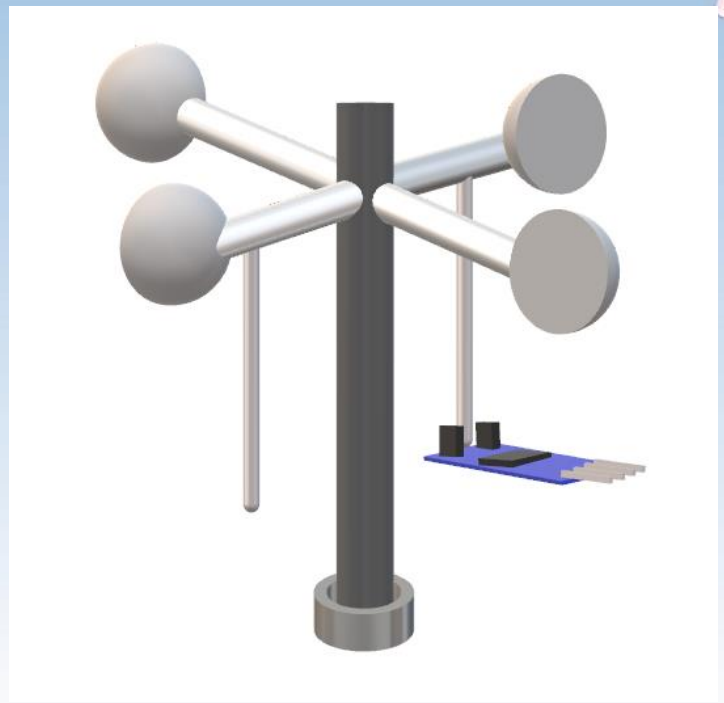
```
  return String(P);
}

//--------------------------------------------------------------------------------------------------------------------------------------------------------------
void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
      found = true;
      break;
    }

    countTimeCommand++;
  }

  if(found == true)
  {
    Serial.println("Done");
    countTrueCommand++;
    countTimeCommand = 0;
  }
  if(found == false)
  {
    Serial.println("Fail");
    countTrueCommand = 0;
    countTimeCommand = 0;
  }
  found = false;
 }
```

Presentation Video

VAWT (Vertical Axis Wind Turbine)

- We planned to prepare a small-scale wind turbine which can rotate in vertical axis, to monitor the speed of wind.

- We had tried several options to measure the rotational speed, but all ended up in vain, leading the project to set aside the VAWT.
    1. Using IR sensor
    2. Using Voltage sensor
    3. Using Ultra sonic sensor

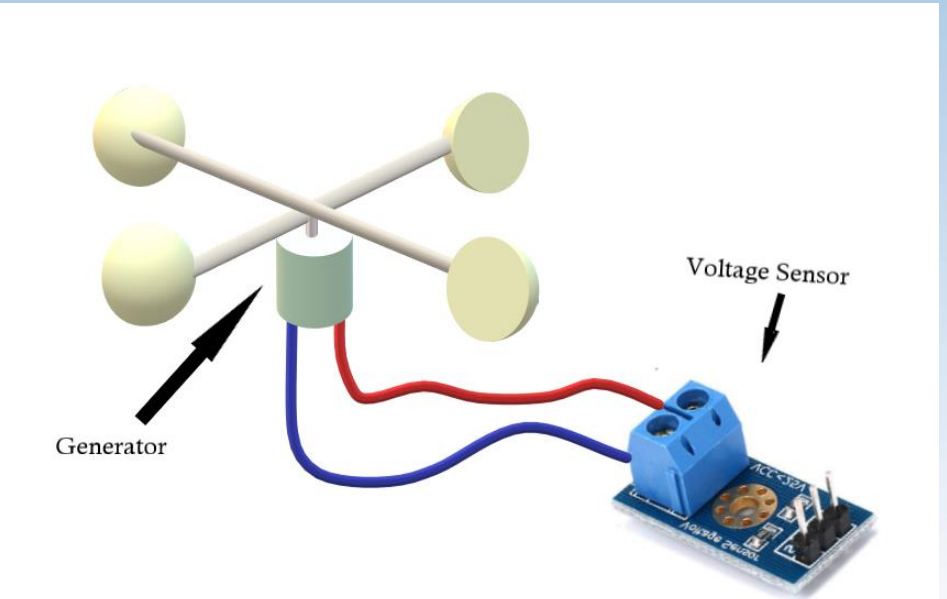


Rough Sketch of Method using IR Sensor

- Unavailability of Anemometer which is of High cost. (So, we are unable to do the mapping of windspeeds with the values of sensor data).

- The materials we used for VAWT are of high mass and the turbine failed to rotate at enough speed. (So, the readings obtained from the voltage sensor seemed to be very low)(0.06 V to 0.2V).

- The system with less-weight materials are not stable.



Rough Sketch of Method using Voltage Sensor

IoT (ThingSpeak) Output Results

The Outputs from each sensor which we obtained in the ThingSpeak platform.

- Date and Time
- Entry ID
- Temperature
- Humidity
- Air Quality
- Pressure
- Rain

	A	B	C	D	E	F	G	H
531	2022-02-12T12:34:40+05:30	530	28	84	54	933.37	970	
532	2022-02-12T12:34:57+05:30	531	28	84	53	933.76	971	
533	2022-02-12T12:35:13+05:30	532	28	84	51	933.81	969	
534	2022-02-12T12:35:33+05:30	533	28	84	52	933.79	972	
535	2022-02-12T12:35:49+05:30	534	28	84	52	933.7	971	
536	2022-02-12T12:36:06+05:30	535	28	84	87	933.59	966	
537	2022-02-12T12:36:30+05:30	536	28	84	70	933.56	967	
538	2022-02-12T12:36:55+05:30	537	28	84	68	933.51	967	
539	2022-02-12T12:37:13+05:30	538	28	84	81	933.59	960	
540	2022-02-12T12:37:29+05:30	539	28	84	77	933.59	963	
541	2022-02-12T12:38:37+05:30	540	27	84	83	933.51	966	
542	2022-02-12T12:38:53+05:30	541	27	84	81	933.48	966	
543	2022-02-12T12:39:20+05:30	542	27	84	82	933.37	966	
544	2022-02-12T12:39:36+05:30	543	27	84	85	933.45	967	
545	2022-02-12T12:39:52+05:30	544	27	84	84	933.37	966	
546	2022-02-12T12:40:19+05:30	545	27	84	85	933.48	967	
547	2022-02-12T12:55:41+05:30	546	28	81	103	657.78	973	
548	2022-02-12T12:56:14+05:30	547	28	81	92	931.05	972	
549	2022-02-12T12:57:02+05:30	548	28	81	92	930.97	973	
550	2022-02-12T12:57:50+05:30	549	28	81	92	931.11	974	
551	2022-02-12T12:58:39+05:30	550	28	81	91	931.36	973	
552	2022-02-12T13:41:33+05:30	551	26	86	89	937.19	973	
553	2022-02-12T13:41:49+05:30	552	26	86	91	936.88	970	
554	2022-02-12T13:42:05+05:30	553	26	86	74	937.16	962	
555	2022-02-12T13:42:44+05:30	554	26	86	75	937.58	975	
556	2022-02-12T13:43:00+05:30	555	26	86	74	937.33	980	
557	2022-02-12T13:43:16+05:30	556	26	86	74	936.35	989	

feeds

Results and Observations

Sensor Name	Measuring Quantity	Results	
		Before Testing	After Testing
Rain Drop Module YL 83	Logic Value of Raining (ON/OFF)	OFF	ON
Air Quality Detection module MQ 135	Prescence of Flammable Gases, CO2, Smoke, NH3, NOx, Alcohol and Benzene in the environment in ppm (Parts Per Million)	Test 1 - Without Flame: 64 ppm Test 2 - Without Alcohol: 66 ppm	Test 1 - With Flame: 102 ppm Test 2 - With Alcohol: 379 ppm
Humidity and Temperature Module DHT 11	Temperature in °C	27 °C	With Hot Water : 35 °C With Cold Ice : 26°C
	Humidity in Percentage (%)	87%	With Hot Water : 91% With Cold Ice : 86%
Digital Barometric Pressure Module GY 68 - BMP180	Pressure of the surrounding environment in hPa (Hecto-Pascals/Millibars)	936 hPa (No Proper Methods available to change the pressure)	

Sample Testing Carried out and its results.

Budget

Components	Cost
Temperature & Humidity Sensor	Rs. 350
Barometric Pressure Sensor	Rs. 250
Rain Sensor	Rs. 200
Air quality sensor	Rs. 360
Arduino UNO	Rs. 2000
Wi-Fi Module	Rs. 360
Breadboard	Rs. 250
Jumper Wires	Rs. 350
Courier Services	Rs. 600
Other Expenses	Rs. 680
Total Expense	**Rs. 5400**

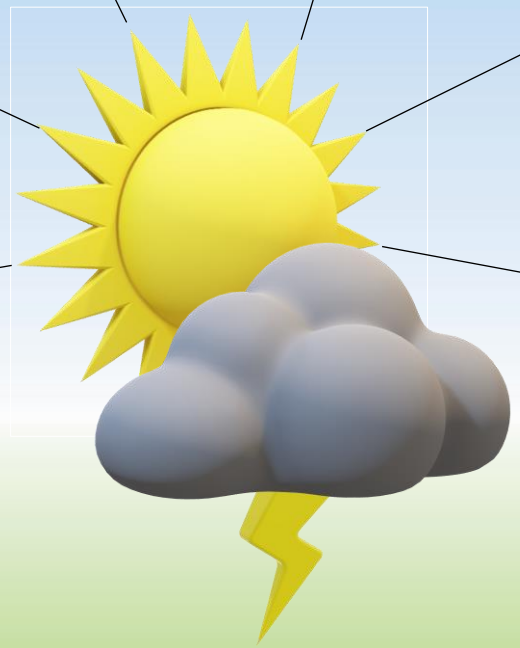
Timeline

Weeks 4, 5 and 6
Gathering Sensors and Implementation

Week 7, 8 and 9
Vertical Axis Wind Turbine Making

Week 3
Project Proposal

Week 10, 11 and 12
Testing and Rechecking Feedbacks



Week 1 and 2
Project topics Research

Week 13 and 14
Final prototype

Group Members

Barathraj M.	(190091K)
Nanthaluxsan E.	(190411U)
Pragalathanan A.	(190468A)
Yathunanthanasarma B.	(190722A)

THANK YOU

Background Image Reference : http://www.freeppt.net/1024x768/april-weather-1026.html