# Autonomous Monitoring System using Wi-Fi Economic

Michael Ames Ccoa Garay[1]
Electronics and Telecommunication Engineering Department, Universidad Nacional Tecnologica de Lima Sur, Lima, Perú

Avid Roman-Gonzalez[2]
Aerospace Sciences and Health Research Laboratory (INCAS-Lab), Universidad Nacional Tecnologica de Lima Sur, Lima, Perú

***Abstract*—In this project, it is presented the implementation of an autonomous monitoring system using solar panels and connecting to the network through Wi-Fi. The system will collect meteorological data and transmit in real-time to the web for the visualization and analysis of the results over temperature, humidity, and atmospheric pressure. The system will allow saving time and money, employing decision making and efficiency. For the development of this device, a small platform "Wemos D1" for the internet of things allows easy programming in the platform "Arduino IDE".**

***Keywords*—Wemos d1 mini-skirt; Wi-Fi; sensor; internet**

## I. Introduction

An autonomous meteorological station is a sophisticated device that helps to reduce weather uncertainties. This task is accomplished through a data history that helps making decisions by saving time and money.

Worldwide projects exist on meteorological stations autonomous for cultivation fields.

SENCROP is a technology company to cultivation fields in France where it includes collecting data and also shares them through the internet.

Has a lot of characteristics but among which it stands out is the simplicity of the devices.

It is light because it allows moving the stone from one side to another without difficulty.

The data obtained in the fields of cultivation are processed to an algorithm developed by them for the predictions of pests and diseases, giving an import value in the decision making [1].

Local levels there are projects on automatic meteorological stations.

For example, the project of an automatic meteorological station of the "Reserva Biologica Alberto Maberes Brenes" in costa Rica, you will see the value of obtaining data with quality and improvement in the care of a resource of a state [2].

The project of the autonomous monitoring system uses Wi-Fi for collecting meteorological data. The idea is to visualize, analysis, and storage in a simple way, record, and at a low cost, all data. This system can help recognize the type of microclimate that exists in the different areas to know the composition of the place flora and improve the use of vegetable species [3].

The document consists of the implementation of a standing monitoring system using development cards like the "Wemos D1 mini" for the data collections at a low cost.

In the implementation development platform is used as the Wemo D1 mini, a precision sensor, a module of TP4056, solar panel, and a rechargeable lithium battery. The system can measure atmospheric pressure, temperature, and relative humidity. These data are transmitted in real-time to the internet through the connection Wi-Fi to visualize and analyze the meteorological data.

Also, use panels that provide an autonomy of the device.

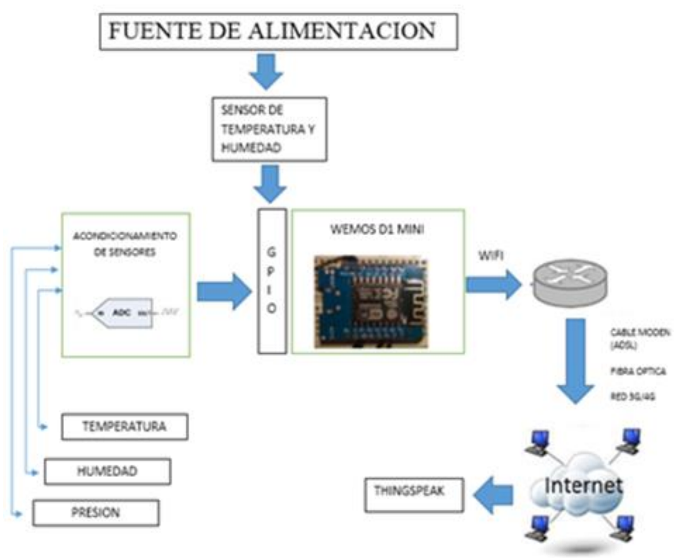


Fig. 1. Diagram Blocks.

## II. Methodology

The main components of the system are:

In Fig. 1, a summary of sensor connections is displayed BMP 280, the modules “Wemos” and “TP4056” with the solar panel and the shipping to the internet using the Wi-Fi.

### A. Description of main Components

*1) Wemos D1 mini-skirt:* The Wemos (Fig. 2): Is a development card that incorporates on its printed circuit to ESP-12F. Is small and includes functions useful as a tension regulator of 5V to 3.3V with which the module is fed with 5V. The power source of 5V allows a current of 500mA. All needed consumption is covered, including micro USB and an integrated circuit model CH340G. The CH340G is a converter USB series to connect the plate directly to the computer without the need to through Arduino.

Tension regulator allows feeding directly with 5V without the need of an additional extreme source.

The tension is taken directly of the existence of 5V of the module T4056 that in turn feeds on energy that transfers solar panels with the module “Wemos” connects to the Wi-Fi network and take measures through the sensor [4]. All these features can be seen in Table I.

TABLE. I. Features of the Module Wemos D1 Mini

| Parameter | Value |
|---|---|
| **Voltage of nutrition** | **5V AD** |
| Voltage of prominent entrance | 3.3V AD |
| Digital pins GPIO | 11 |
| Analogical pins ADC | 1 |
| Average of consumed current | 70 mA |
| Memory flash external | 4MB |
| Date RAM | 96kB |
| I mince current | 400mA |
| Frequency of clock | 80 Mhz 160 Mhz |
| Microcontroller | ESP8266 |
| Dimensions | 34.2mm x 25,6 mm |
| I weigh | 10g |

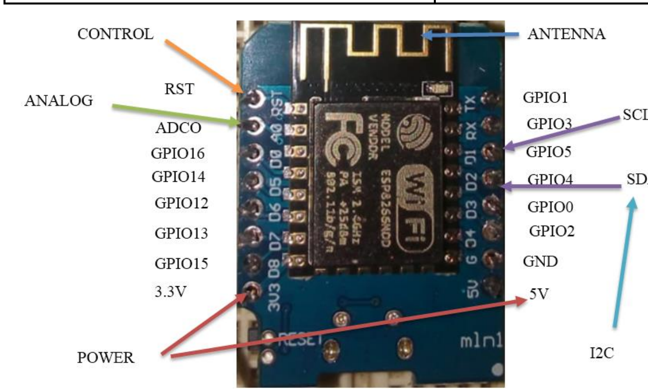


Fig. 2. Module Wemos D1 Mini.

*2) I modulate TP4056:* The modulate is a battery charger of lithium TP4056 that account with an entry micro USB and an additional two contactors for direct connection IN+ and IN-. At the other end are the terminals B+ and B- that is responsible for connecting to the battery to change. The modulate loads the constant current 1A until the moment the current starts to decrease, activating the mode to change at a constant voltage. All these features can be seen in Table II and in Fig. 3.

*3) Sensor BMP280:* The sensor BMP 280 will allow us measuring atmospheric pressure in the status of 300 to 1100hpa with +/- average error 40 to 85 C. With the error one can measure 1hpa regarding temperature from -1,0 C. The interface allows us connecting the Wemos by SPI or I2C that needs a source of 3.3V [5]. All these features can be seen in Table III and in Fig. 4.

*4) Power supply:* For this project, it was used a battery of 2800 mah. That connect through the module TP40556 will get energy from solar panels. It can be seen in Fig. 5.

TABLE. II. Characteristics of the I Modulate TP4056

| Parameter | Value |
|---|---|
| Voltage from the start | 4.5V 5.5V |
| Charging voltage full | 4.2V |
| Entrance Microcomputer Usb | If |
| Operating temperature | - 10 C to +85 C |
| Inverse polarity | No |
| Charging precision | 1,5 % |
| Charging mode | Linear load |
| Current of maximum charging exit | 1.2A |

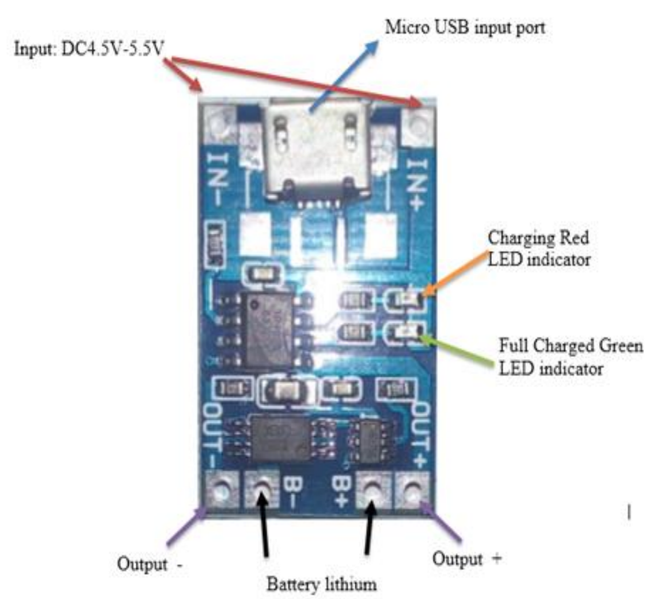


Fig. 3. Module TP4056.

TABLE. III. CHARACTERISTICS OF THE SENSOR BMP 280

| **Parameter** | **Value** |
|---|---|
| Voltage of operation | 1.8V 3.3V AD |
| Interface of communication | I2C or SPI 3.3V |
| Status of pressure | 300 to 1100hPa |
| Absolute precision | 1 hPa |
| Measurement of temperature | - 40 C to +85 C |
| Precision of temperature | 1 C |
| Frequency of sampling | 157 Hz |
| I decrease consumption of energy | If |


Fig. 4. Sensor BMP280.

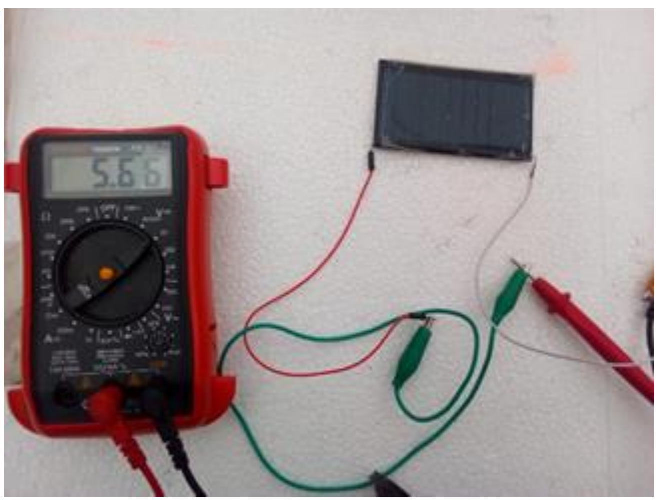

Fig. 5. Solar Panel of 5.5V.

## B. Implementation of the Autonomous Monitoring System

The monitoring system is autonomous; there must be a continuous power supply. The best way to provide uninterrupted power to the circuit is through the use of a battery. After a few days, the battery charge would be depleted, and it is very complicated to get the energy of the sun to charge the batteries and provide power to all circuit. For this project, it was used a battery of lithium 2800 mah. It can be seen in Fig. 6.

The system can measure temperature, humidity, barometric, pressure. Also, it can monitor the meteorological parameters due to a record stored from the web.

The battery is charged using a solar panel through a charging module TP4056.

Fig. 6. Lithium Battery of 2800mA.

The module TP4056 is ideal for loading cells (Solar panels) from 3.7V and 1A. This module will offer a constant charging current of 1A and then cut when the load is also finished. When the tension of the battery descends below 2.4V, the IC of protection will reduce the load to protect her cells of the battery against the low voltage. Also, protect against the excessive energy and the connection of inverse polarity. It can be seen in Fig. 7.

*1) Installation of the solar panel and the battery:* A cable is welded to the solar panel's negative terminal to the positive terminal and the black wire. Next, insert the battery's support in the slot in the protoboard's part. It can be seen in Fig. 8.

*2) Programming:* After the implementation of the plate of development Wemos, the sensor should position itself. Then, it will send the instructions for the sensor of ultrasounds to accomplish the measurement of temperature, humidity, and atmospheric pressure for which washed out to stub the following code. The Wemos module gets connected to net road Wi-Fi and besides sends the data for its visualization in real-time. The inserted code is shown from Fig. 10 to Fig. 15.

To use Wemos D1 with the library, Arduino, you will have to use the ÍDE ARDUINO with support ESP8266. Other forms of plate seriously install the plate's support ESP8266 in the ÍDE of Arduino [6]. It can be seen in Fig. 9.

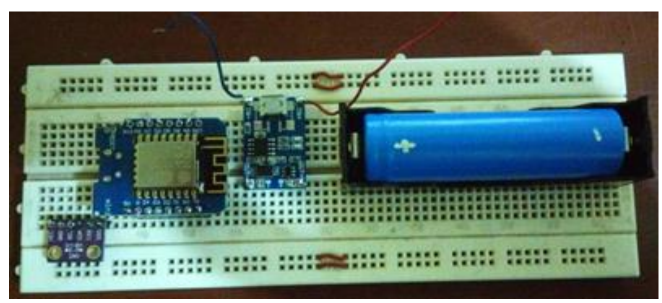

Fig. 7. Prototype Implementad unrelated.

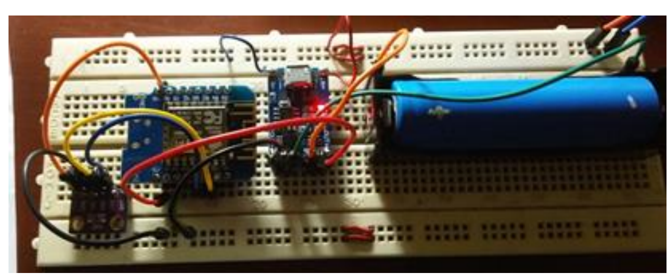

Fig. 8. Prototype Implemented without Connections.

The following adjustments are preferable:

- The PU's frequency:

  80MHz 160MHz

- Size of the flash: The archival system's (3M SPIFFS) - the archival system's Size 3M 4M (1M SPIFFS) - So Big a 4M 1M

- Charging velocity: 921600 bps [7]

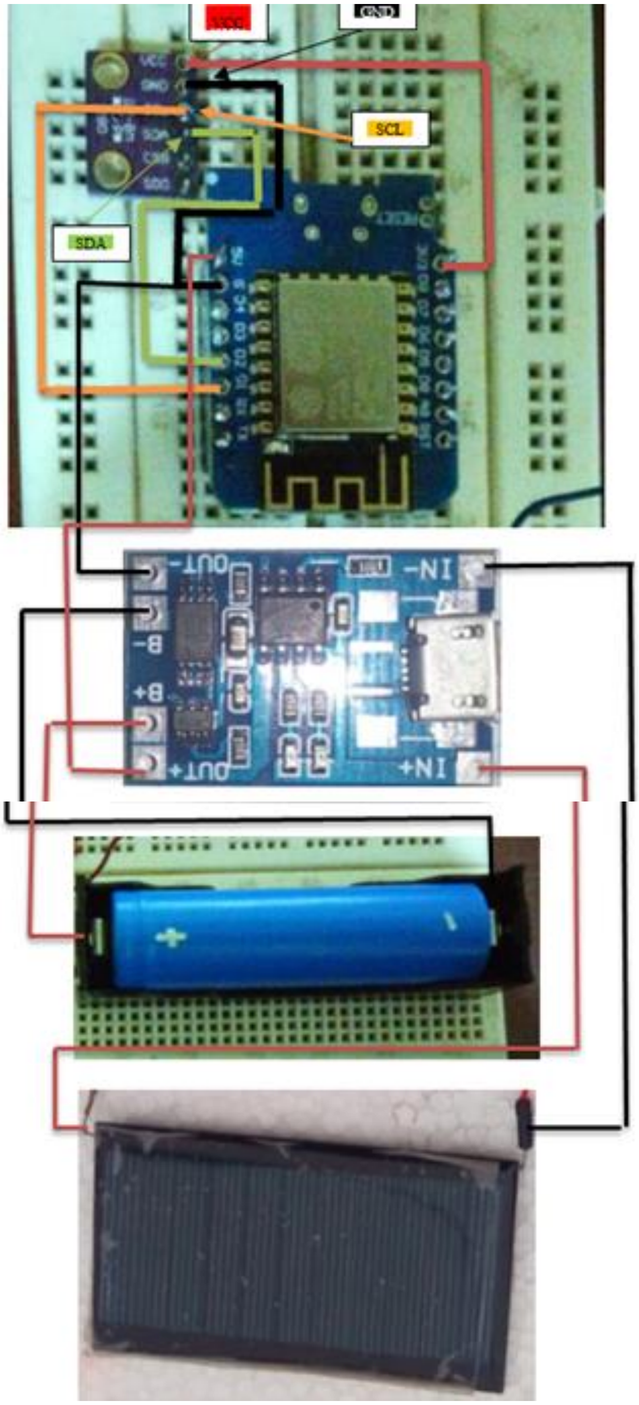


Fig. 9. Wire Diagram.

```
#include <BME280_MOD-1022.h>
#include <Wire.h>

// Wifi and ThingSpeak settings
#include <ESP8266WiFi.h>

const char* ssid = "UNTELS wifi";
const char* password = "ygdo5q00bm";

const char* server = "api.thingspeak.com";
const char* api_key = "Z95RSV977I166TYR";

// Measurement interval (seconds)
const int interval = 300; //5 mins

#define LED D4

WiFiClient client;

void printFormattedFloat(float x, uint8_t precision) {
char buffer[10];

  dtostrf(x, 7, precision, buffer);
  Serial.print(buffer);

}

void printCompensatedMeasurements(void) {
```

Fig. 10. Network Configuration that One is Going to Connect.

```
void printCompensatedMeasurements(void) {

float temp, humidity,  pressure, pressureMoreAccurate;
double tempMostAccurate, humidityMostAccurate, pressureMostAccurate;
char buffer[80];

  temp      = BME280.getTemperature();
  humidity  = BME280.getHumidity();
  pressure  = BME280.getPressure();

  pressureMoreAccurate = BME280.getPressureMoreAccurate();  // t_fine a

  tempMostAccurate     = BME280.getTemperatureMostAccurate();
  humidityMostAccurate = BME280.getHumidityMostAccurate();
  pressureMostAccurate = BME280.getPressureMostAccurate();

  Serial.print("Temperature: ");
  printFormattedFloat(tempMostAccurate, 2);
  Serial.println();

  Serial.print("Humidity: ");
  printFormattedFloat(humidityMostAccurate, 2);
  Serial.println();

  Serial.print("Pressure: ");
  printFormattedFloat(pressureMostAccurate, 2);
  Serial.println();
```

Fig. 11. Declaring Variables.

```
  // Post data to ThingSpeak
  postData(tempMostAccurate, humidityMostAccurate, pressureMostAccurate);
  Serial.println();
}

void postData(float temperature, float humidity, float pressure){
  // Send data to ThingSpeak
  if (client.connect(server,80)) {
  Serial.println("Connect to ThingSpeak - OK");

  String dataToThingSpeak = "";
  dataToThingSpeak+="GET /update?api_key=";
  dataToThingSpeak+=api_key;

  dataToThingSpeak+="&field1=";
  dataToThingSpeak+=String(temperature);

  dataToThingSpeak+="&field2=";
  dataToThingSpeak+=String(humidity);

  dataToThingSpeak+="&field3=";
  dataToThingSpeak+=String(pressure);

  dataToThingSpeak+=" HTTP/1.1\r\nHost: a.c.d\r\nConnection: close\r\n\r\n";
  dataToThingSpeak+="";
  client.print(dataToThingSpeak);

  int timeout = millis() + 5000;
  while (client.available() == 0) {
```

Fig. 12. A Code is Generated for the Website.

```
    if (timeout - millis() < 0) {
      Serial.println("Error: Client Timeout!");
      client.stop();
      return;
    }
  }
}
 while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
}


// Setup wire and serial
void setup()
{
  Wire.begin();
  Serial.begin(115200);
  pinMode(LED, OUTPUT);
  delay(10);
  Serial.println("Connecting to wifi...");
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED){

    // Blink LED when connecting to wifi
    digitalWrite(LED, LOW);
    delay(250);
```

Fig. 13. Showing Wi-Fi Connection.

```
  while (WiFi.status() != WL_CONNECTED){

    // Blink LED when connecting to wifi
    digitalWrite(LED, LOW);
    delay(250);
    digitalWrite(LED, HIGH);
    delay(250);
  }
  Serial.println("WiFi connected");


  // Prepare LED to turn on when measuring and send data

}

// main loop
void loop()
{
  // need to read the NVM compensation parameters
  BME280.readCompensationParams();

  // We'll switch into normal mode for regular automatic samples
  BME280.writeStandbyTime(tsb_0p5ms);         // tsb = 0.5ms
  BME280.writeFilterCoefficient(fc_16);       // IIR Filter coefficient 16
  BME280.writeOversamplingPressure(os16x);    // pressure x16
  BME280.writeOversamplingTemperature(os2x);  // temperature x2
  BME280.writeOversamplingHumidity(os1x);     // humidity x1
```

Fig. 14. Showing Network Connection.

```
{
  // need to read the NVM compensation parameters
  BME280.readCompensationParams();

  // We'll switch into normal mode for regular automatic samples
  BME280.writeStandbyTime(tsb_0p5ms);         // tsb = 0.5ms
  BME280.writeFilterCoefficient(fc_16);       // IIR Filter coefficient 16
  BME280.writeOversamplingPressure(os16x);    // pressure x16
  BME280.writeOversamplingTemperature(os2x);  // temperature x2
  BME280.writeOversamplingHumidity(os1x);     // humidity x1

  BME280.writeMode(smNormal);

  while (1) {
    //digitalWrite(LED, LOW);
    while (BME280.isMeasuring()) {
      //Serial.println("Measuring...");
      //delay(100);
    }

    // read out the data - must do this before calling the getxxxxx routines
    BME280.readMeasurements();
    printCompensatedMeasurements();
  //  digitalWrite(LED, HIGH);

    delay(interval*1000);
    Serial.println();
  }
}
```

Fig. 15. Constant Repetition with the Command.

*3) Sending the data of the sensor BMP 280 to the Web Thingspeak*

First, an account in Thingspeak is created. Next, a new canal in the account of Thingspeak is created. It gets stung with the data at the

- Field 1: Temperature
- Field 2: Humidity
- Field 3: Pressure

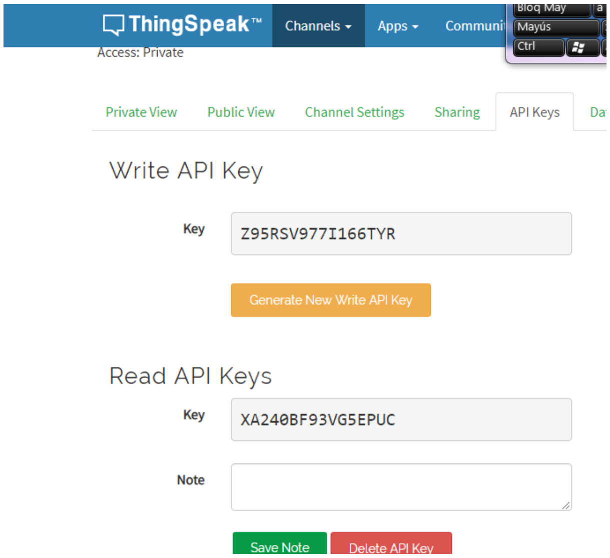


Fig. 16. Nail Down API.

The channel is selected after in Thingspeak's new account. It stops next looking for eyelash password API and imitates the keyboard write.

The net which one wants to connect itself the SSID to is written in the interface of the Arduino Íde once the code was opened stops next placing the password on the code. API replaces the WRITE itself and API copies the write itself key that Thingspeak's page provides us. It is essential to have installed the BMP's bookstores 280 [8]. It can be seen in Fig. 16.

## III. Results

ThingSpeak compiles information about temperature, humidity, and atmospheric pressure himself next after being sent to the platform the data in graphs could be visualized.

Data of temperature he shows the following values shown in Fig. 17 and Fig. 18.

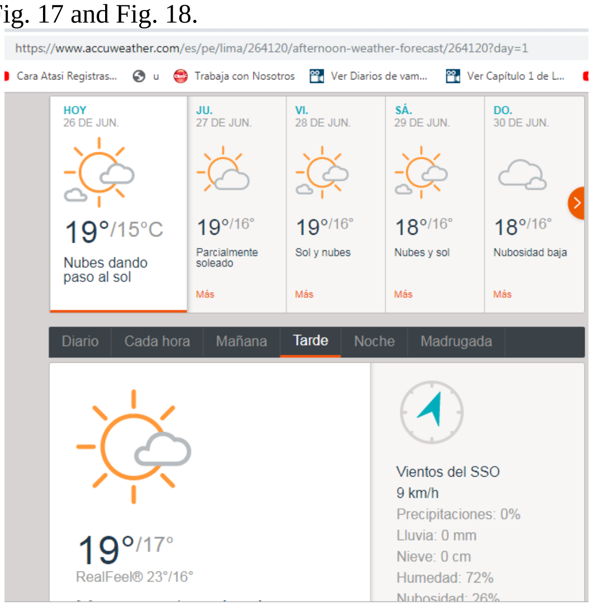


Fig. 17. The Platform's Graph AccuWeather.

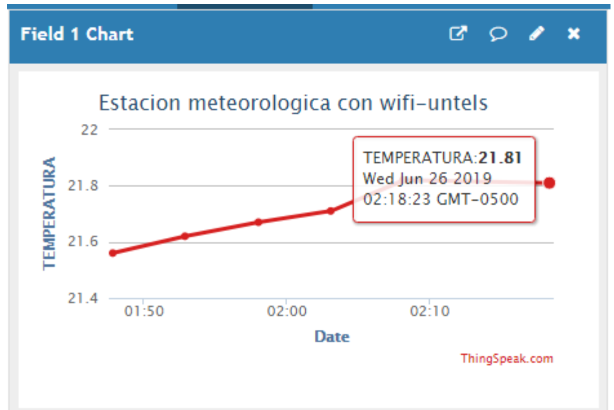


Fig. 18. Graph ThingSpeak of Temperature.

## IV. Discussion

The system's implementation is simple and economical. The objective to gather meteorological data and to be able to visualize, utilizing connection Wi-Fi to the internet, from any place thanks to the card of development fulfills Wemos. Wemos has the individual capacity of no. to control to transfer the data to the net giving one a great variety of application software. The sensor BMP280 is the precise sensor that if not one, also obtains the meteorological measures.

The application software that one can get from the module Wemos was joined of ThingSpeak. It is one of the platforms of software enlarged for the ones that wants to start-up in the world of the internet of thing (IoT). It is a simple way due to his compatibility with the card Arduino.

The recommendation is to take into account the bringing up to date of the bookstores of the Arduino. It would have the ones one come than by default that to make some modifications, but it is easy to obtain the necessary bookstores in the web.

The system has wholes the elements to be able to improve from their web interface in the designing improvement as in the structure and components. They allow protecting of the storms of the climate once the external use was given to and your portability.

Data of humidity shows the following values shown in Fig. 19.

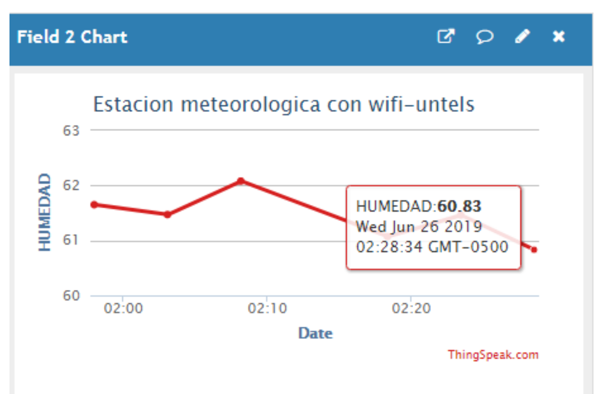


Fig. 19. Graph ThingSpeak of Relative Humidity.

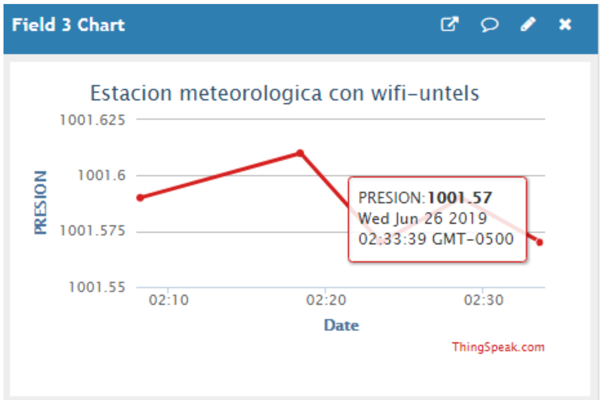


Fig. 20. Graph ThingSpeak of Atmospheric Pressure.

Data of atmospheric pressure shows values shown in Fig. 20.

## V. Conclusions

ThingSpeak compiles information about temperature, humidity, and atmospheric pressure. This itself is next after being sent to the platform and could compare with the data in AccuWeather, a North American company, and renders commercial services of weather forecast all over the world. One can become evident in the image than the sensor Bmp 280 is within range regarding the company of meteorological. They are strong values that the project gives us but unlike the rest of meteorological devices. In general, a technical service that one can help when a flaw exists does not have. This system is a very cost-reducing and easy project of implementation; besides that it is straightforward since one can move without worrying about the reserve of energy. One can pick up data that are very important for the study of farm cultivation for deeply letting us know the acquaintance microclimate that the Peruvian ground has.

## References

[1] Tripathi, Harish Kumar, and B. K. Sen. "Crop science literature and Bradford law." Annals of Library and Information Studies (ALIS) 63.2 (2016): 85-90.

[2] Sandoval, Chapoñan, and Julio Cesar. "Caracterización temporal del viento registrado en el borde costero de la ciudad de Santa Rosa y en las Islas Lobos de Afuera, durante los años 2005 al 2012." (2016).

[3] Sala, Juan Ignacio, et al. "Estación de medición para análisis y control de parámetros ambientales." IX Congreso Argentino de AgroInformática (CAI 2017)-JAIIO 46-CLEI 43 (Córdoba, 2017). 2017.

[4] Girón, Luis Diego Maldonado. "Diseño De Una Estación Meteorológica Con Control De Accionamientos Electromecánicos Y Monitoreo De Sensores."

[5] Medina Marín, Jairo José, and Alexis Oldemar Valle Ruiz. Diseño de una estación meteorológica para la medición de temperatura, humedad, presión atmosférica y radiación solar en el área del Recinto Universitario Rubén Darío, la cual estaría ubicado en el Centro de Investigaciones Geo Científicas (CIGEO), en la UNAN Managua. Diss. Universidad Nacional Autónoma de Nicaragua, Managua, 2017.

[6] Claudio P.Millahual,"Arduino-De cero a Experto:Proyectos Practicos-electronica , hardware y programacion",pp-110-114.

[7] Ben Akka, Youssef, et al. "Control and Command Of Several Greenhouses Via Telegram Messenger." Proceedings of the 2nd International Conference on Networking, Information Systems & Security. ACM, 2019.

[8] Loureiro Garrido, Rubén. "Estudio Plataformas IoT."