

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 Oficina 1105
Providencia, Santiago, Chile
cotizaciones@mcielectronics.cl
Tel: +56 2 23339579

Pasaje Antonio Varas 307 Local 8
Providencia, Santiago, Chile
showroom@mcielectronics.cl
Tel: +56 2 32249067

Viana 405, local 7
Viña del mar, Valparaíso, Chile
vinadelmar@mcielectronics.cl
Tel: +56 3 22710559

www.mcielectronics.cl
info@mcielectronics.cl

® MCI Ltda. 2019

Atención: cambios y modificaciones hechas en el dispositivo, no autorizados expresamente por MCI, anularán su garantía.

Código Manual: MCI-MA-2304

CONTENIDO

| | |
|--|----|
| Contenido..... | 3 |
| INTRODUCCIÓN | 4 |
| CARACTERÍSTICAS GENERALES | 4 |
| PARTES DEL DISPOSITIVO | 5 |
| EJEMPLO N°1: FUNCIONAMIENTO BÁSICO | 6 |
| EJEMPLO N°2: SENSADO Y MONITOREO | 8 |
| CARACTERÍSTICAS ELÉCTRICAS | 15 |
| CARACTERÍSTICAS MECÁNICAS | 15 |
| HISTORIA DEL DOCUMENTO | 16 |

INTRODUCCIÓN

La ESP WIFI SHIELD es una tarjeta que cuenta con un módulo WiFi para ser integrada a un a PICARO o Arduino Uno, de esta manera es ideal para desarrollar proyectos donde se requiera acceso a internet, excelente para ser implementados en IoT. Además, puede ser usado con un FTDI sin necesidad de ser conectado a una PICARO o Arduino UNO.

CARACTERÍSTICAS GENERALES

- Módulo ESP8266
- Diseñado para ser usado con PICARO o Arduino UNO.
- Botones de Reset y Flash para configuración del módulo ESP8266
- Leds de comunicación
- Pines reflejados del módulo
- Pines tipo protoboard para soldar componentes externos
- Pines para conectar un FTDI
- Pines reflejados del Arduino.

PARTES DEL DISPOSITIVO

En la siguiente imagen se muestra las diferentes partes que compone la ESP WIFI SHIELD, como pines, módulos, botones entre otros.

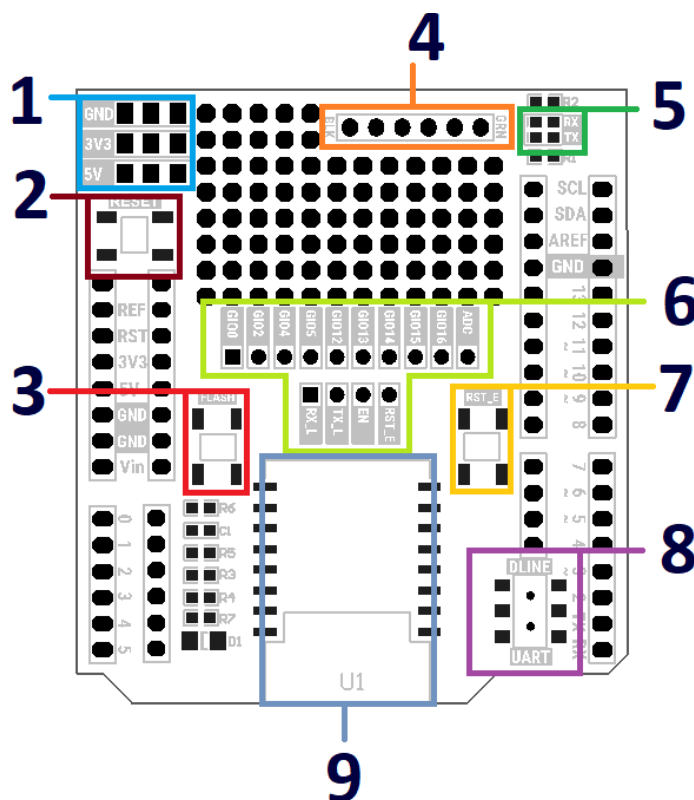


Figura 1. Partes de ESP WIFI SHIELD.

1. **PWR:** Pines de alimentación 5VDC y 3.3VDC.
2. **RESET:** Botón para reiniciar a la PICARO o Arduino UNO.
3. **FLASH:** Botón para entrar en modo carga de firmware junto con el botón RST_E.
4. **FTDI:** Conector para conectar un módulo FTDI Basic para modificar el firmware del ESP8266.
5. **LEDs:** Luces indicadores de comunicación serial.
6. **PINES:** Reflejo de todos pines del módulo ESP8266.
7. **RST_E:** Botón que reinicia el módulo.
8. **SWITCH:** Cambia la comunicación del Arduino con el módulo. UART (0 y 1), DLINE (2 y 3).
9. **ESP8266:** Módulo WiFi.

EJEMPLO N°1: FUNCIONAMIENTO BÁSICO

El siguiente ejemplo realiza la verificación de conexión de la ESP WIFI SHIELD, para esto antes de realizar esta prueba se requiere conectar sobre la PICARO la tarjeta, así como se muestra en la siguiente imagen.

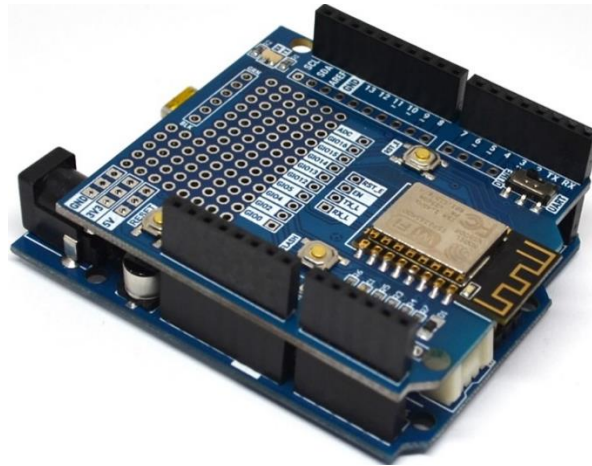


Figura 2. Conexión de ESP WIFI SHIELD sobre PICARO.

Una vez conectada la tarjeta sobre la PICARO, se conecta mediante un cable USB al computador y se procede a cargar el siguiente código de ejemplo.

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(3, 2); // RX, TX
void setup()
{
    Serial.begin(9600);
    while (!Serial) { ; }
    mySerial.begin(9600);
}
void loop()
{ // run over and over
    if (mySerial.available()) {
        Serial.write(mySerial.read());
    }
    if (Serial.available()) {
        mySerial.write(Serial.read());
    }
}
```

Código 1. Código de ejemplo 1.

Este código convierte a la PICARO en un puente, recibiendo los comandos desde el PC y enviándolos al módulo WiFi de la ESP WIFI SHIELD y viceversa. Para ello, se debe abrir el monitor serial a 9.600 baudios y se envía un comando AT, obteniendo como respuesta un OK, así como se muestra en la siguiente imagen.



Figura 3. Respuesta a los comandos AT del módulo Esp8266.

Además, este mismo código de prueba permite la configuración y manejo del módulo ESP8266 de forma manual, a través del ingreso de los distintos comandos AT usando el monitor serial del Arduino. Estos comandos AT del módulo ESP8266 se encuentran disponible en la pestaña de descargas del producto en nuestra página web.

EJEMPLO N°2: SENSADO Y MONITOREO

En el siguiente ejemplo, la PICARO realiza la lectura de un sensor de temperatura y humedad ([HDC1080](#)). Estos datos obtenidos son enviados a la plataforma Thing Speak, la cual permite visualizar los datos de forma gráfica en cualquier parte del mundo donde exista internet. La ESP WIFI SHIELD, permite realizar el envío de los datos a la plataforma Things Speak, ya que proporciona a la PICARO conexión a la red.

Ahora bien, para poder visualizar los datos en Thing Speak se debe crear una cuenta en el siguiente enlace: https://thingspeak.com/users/sign_up.

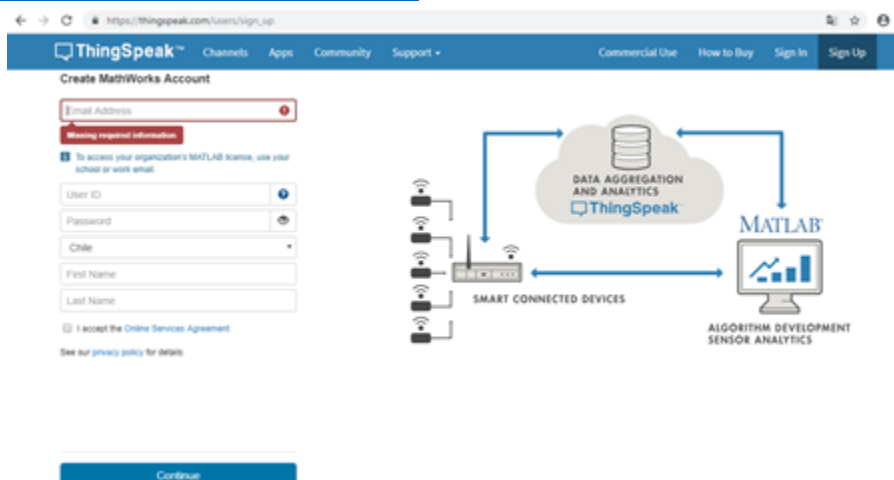


Figura 4: Formulario cuenta en Thing Speak.

Luego de obtenida la cuenta en Thing Speak, se procede a crea un canal. Cada canal contiene hasta ocho campos de datos y cada campo almacena los datos recibidos de algún dispositivo o aplicación. Para este caso se crea un canal con dos campos, un campo para almacenar los datos de temperatura y otro para los datos de humedad, ambos datos son suministrados por el sensor DHC1080.

En la siguiente imagen se puede ver los pasos para la creación del canal y sus campos:

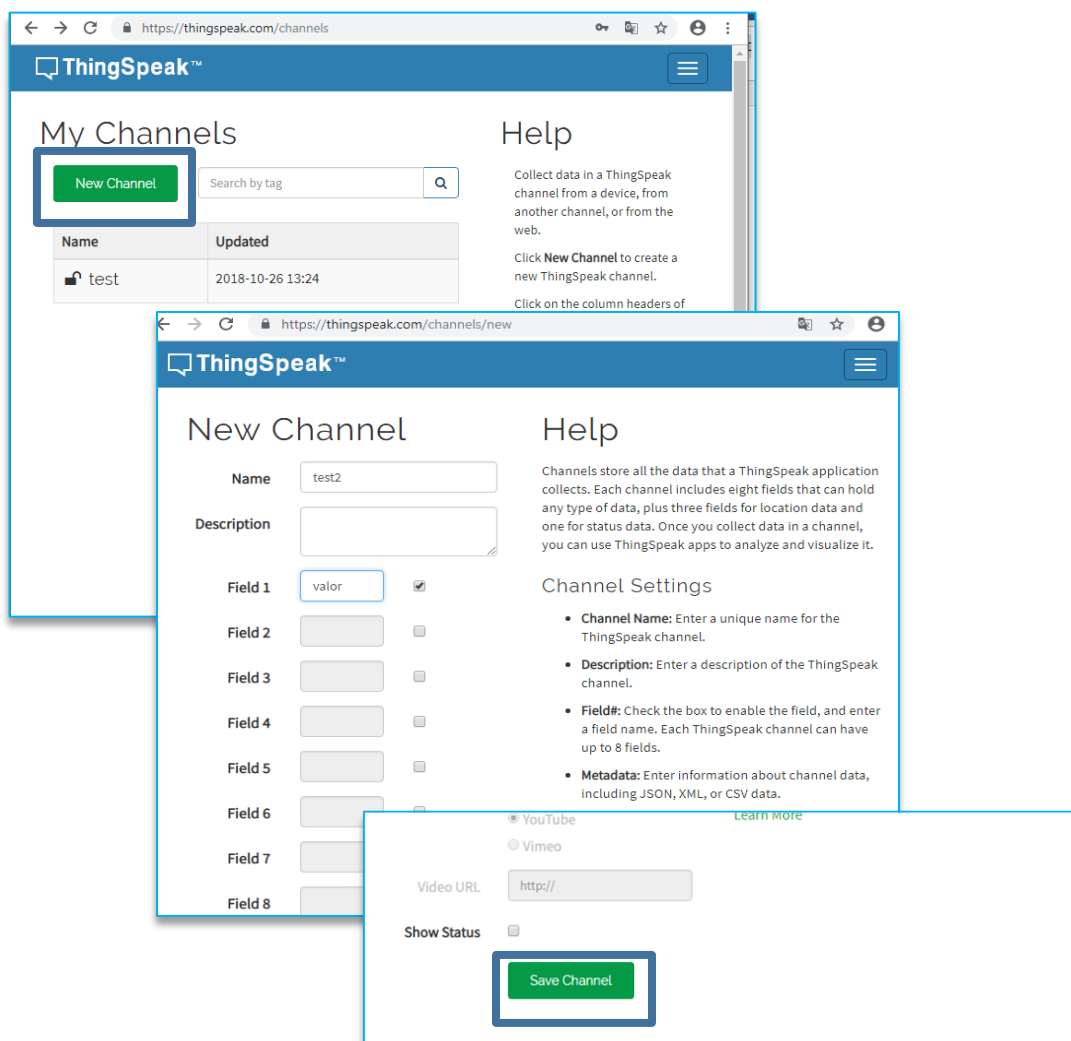


Figura 5: crear canal en Thing Speak.

Una vez creado el canal con los campos, se procede a conectar el sensor HDC1080 a los pines SDA, SCL, 5V y GND de la PICARO, así como se muestra en la siguiente figura:

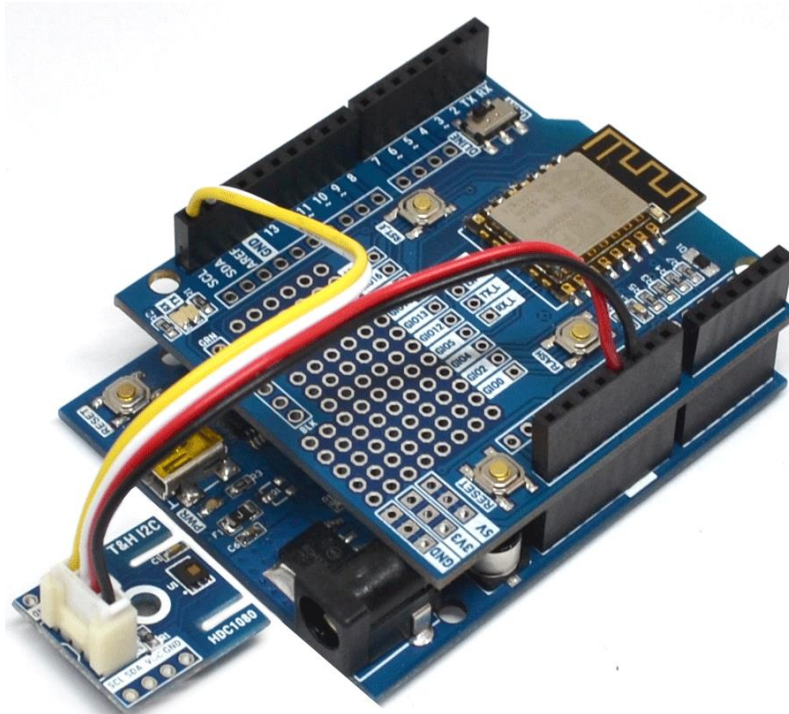


Figura 6. Conexión del sensor HDC1080.

Una vez conectado el sensor, se conecta el cable micro USB y se carga el siguiente código de ejemplo a la PICARO:

```
float Temperatura = 0;
float Humedad = 0;

String ID = "SSID_RED_WIFI"; // SSID de red Wifi
String Clave = "Password_Red_Wifi"; // Password de Red Wifi
String Key= "KEYS_THING_SPEAK"; // Keys api de ThingSpeak.com

int cont = 0;

#include <Wire.h> // Librería comunicación I2C
#include "ClosedCube_HDC1080.h" //Librería Sensor HDC1080
ClosedCube_HDC1080 hdc1080;
#include <SoftwareSerial.h>
SoftwareSerial mySerial(3, 2); // RX, TX
```

```
void setup()
{
  Serial.begin(9600); // configurar Baudios del Serial que se comunica con la PC
  while (!Serial) {
    ;
  }
  mySerial.begin(9600); // Configurar Baudios del serial que se comunica con
                        //el módulo ESP8266

  mySerial.println("AT+CWMODE=1"); // Modo Cliente
  delay(100);
  mySerial.println(AT+CWJAP=""+ID+"\",""+Clave+""); // ingresa datos de red WiFi
  delay(100);
  mySerial.println ("AT+CIPMUX=0"); // deshabilita múltiples conexiones
  delay(100);

  hdc1080.begin(0x40);
  Serial.print("Manufacturer ID=0x");
  Serial.println(hdc1080.readManufacturerId(), HEX); // 0x5449 ID of Texas Instruments
  Serial.print("Device ID=0x");
  Serial.println(hdc1080.readDeviceId(), HEX); // 0x1050 ID of the device
  delay(5000);
}

void loop()
{
  Temperatura=hdc1080.readTemperature();
  Serial.println("TEMPERATURA= "+String(Temperatura)+"°C");
  Humedad= hdc1080.readHumidity();
  Serial.println("HUMEDAD= "+String(Humedad)+"%");
  delay(300);
  mySerial.println AT+CIPSTART="TCP","\","api.thingspeak.com","\",80");
  delay(200);
  mySerial.println ("AT+CIPSEND=51");
  delay(200);
  mySerial.println ("GET /update?api_key="+Key+"&field1="+String(Temperatura));
  delay(200);
  mySerial.println ("AT+CIPCLOSE");
  delay(15000);
}
```

```
mySerial.println ("AT+CIPSTART=\\"TCP\\","\\"api.thingspeak.com\\",80");
delay(200);
mySerial.println ("AT+CIPSEND=51");
delay(200);
mySerial.println ("GET /update?api_key="+Key+"&field2="+String(Humedad));
delay(200);
mySerial.println ("AT+CIPCLOSE");
delay(15000);
contador();
chequeo();
}

void contador()
{
  if(analogRead(A0)<600)
  {
    cont=cont + 1;
  }
  else
  {
    cont=0;
  }
}

void chequeo()
{
  if (cont==60)
  {
    digitalWrite(4,LOW);
  }
}
```

Código 2. Código ejemplo 2.

NOTA: es importante resaltar que existen tres parámetros que se deben modificar del código antes de ser cargado a la PICARO y estos son los siguientes:

- **KEYS_THING_SPEAK:** este parámetro se sustituye por la llave que proporciona Thing Speak para que los datos sean cargados a la plataforma. Se debe resaltar que cada canal creado en Thing Speak, proporciona una llave distinta. Esta llave se encuentra en la sección **API Keys**, del canal creado. En la siguiente imagen se muestra la sección que proporciona la llave:

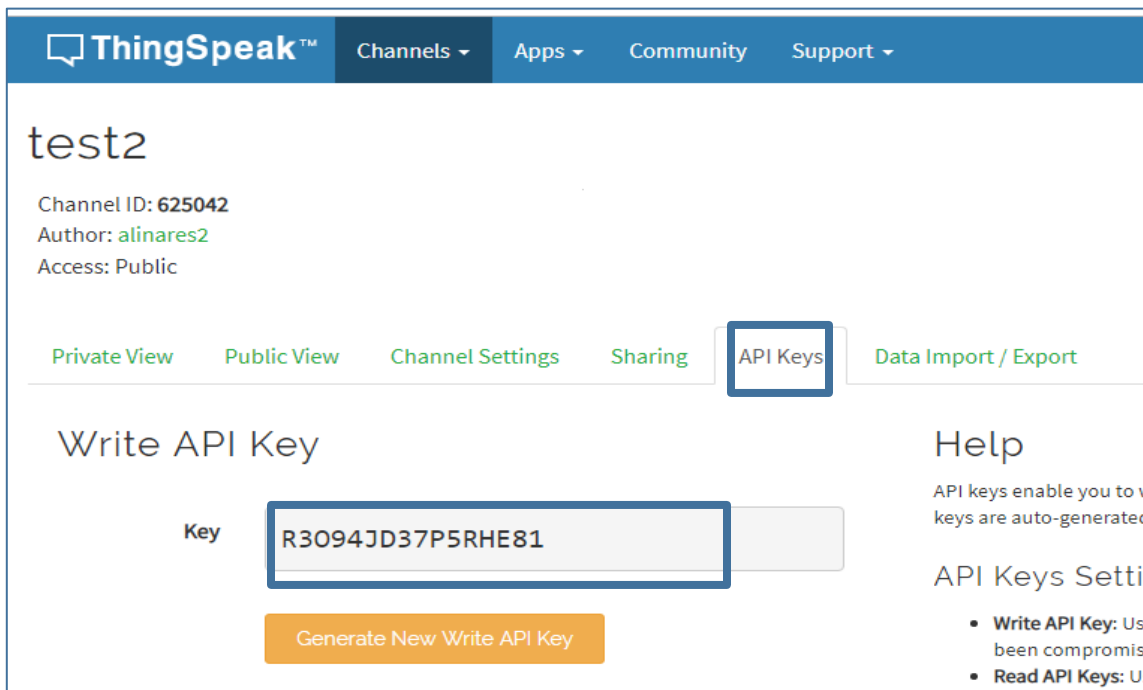


Figura 7. key de Thing Speak.

- **SSID_RED_WIFI:** este parámetro se sustituye por el SSID de la red WiFi, a la que se desea conectar la PICARO.
- **PASSWORD_RED_WIFI:** este parámetro corresponde al password de la red WiFi.

Luego de cargar el código de ejemplo a la PICARO, se puede visualizar la lectura de las variables del sensor en el monitor serial del Arduino, así como se muestra en la siguiente figura:

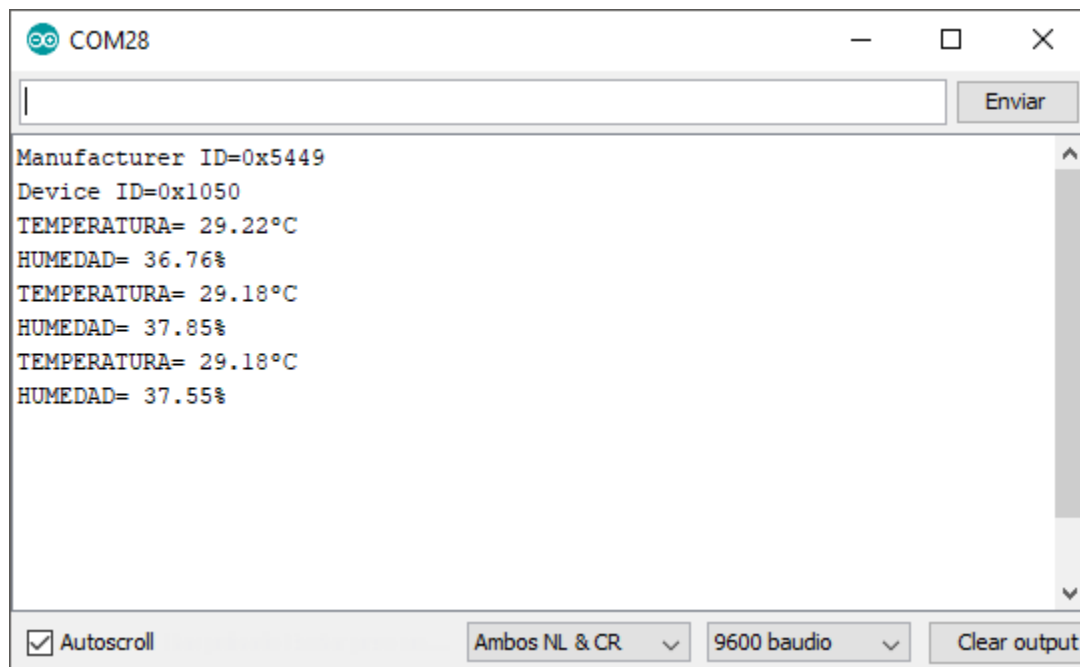


Figura 8. Visualización de los datos en monitor serial.

De igual forma, se puede visualizar los valores graficados en la plataforma de Thing Speak y compararlos con los mostrados en el monitor serial.

La siguiente imagen representa los campos del canal creado en Thing Speak, donde se grafican los valores de temperatura y humedad, que la PICARO recopila del sensor HDC1080.

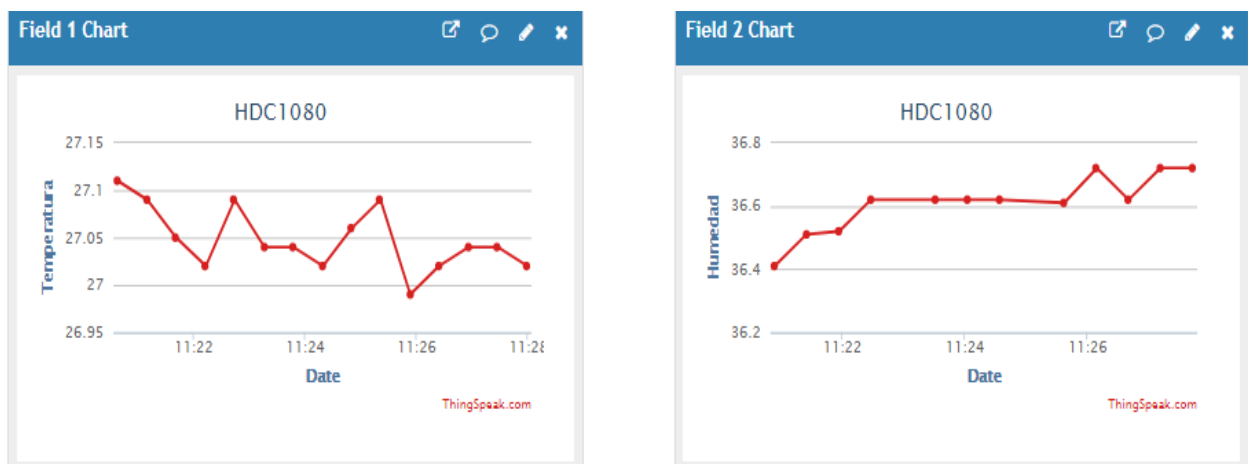


Figura 9. Visualización de los datos en thing speak.

CARACTERÍSTICAS ELÉCTRICAS

- Tensión de alimentación 5 [VDC]
- Consumo máximo de corriente 130.5 [mA]
- Consumo mínimo de corriente 33 [mA]

CARACTERÍSTICAS MECÁNICAS

Dimensiones: 59.59X53.42 [mm]

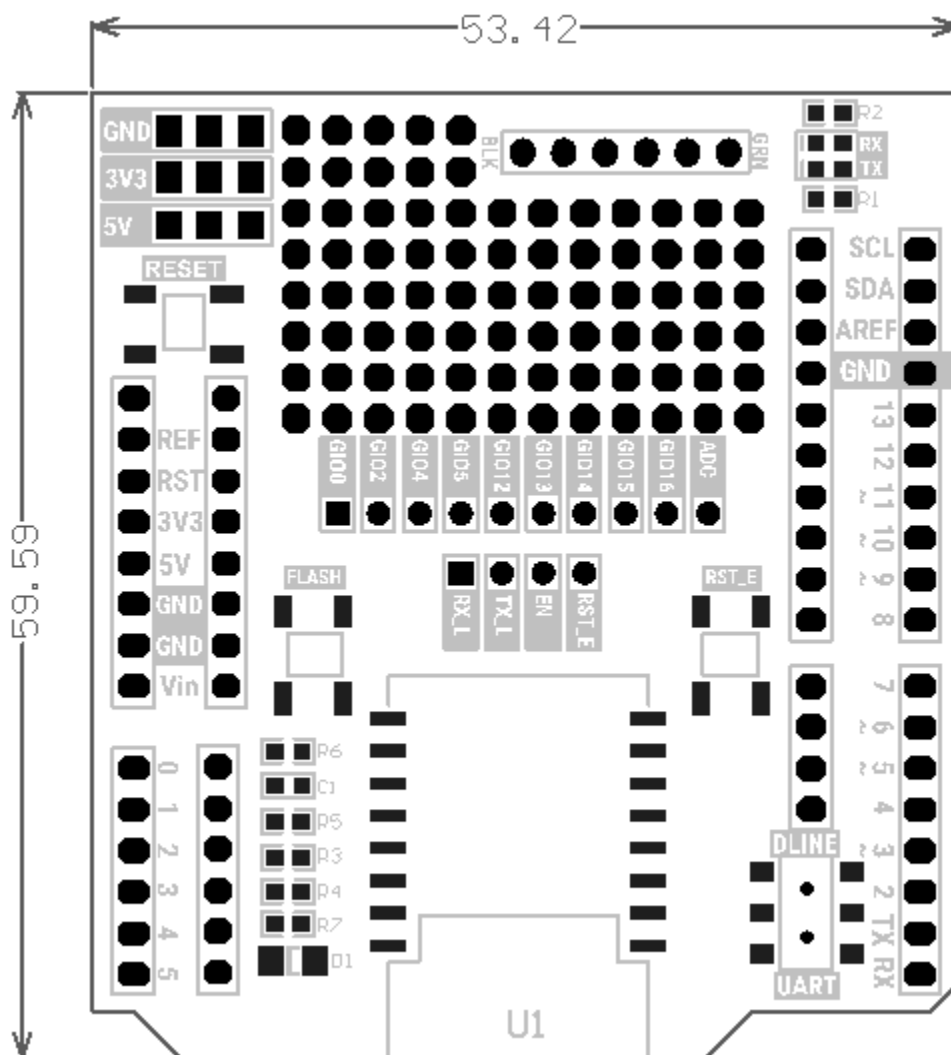


Figura 17. Dimensiones de la tarjeta.

HISTORIA DEL DOCUMENTO

| Revisión | Fecha | Editado por | Descripción/Cambios |
|----------|----------------------|----------------------|-------------------------|
| 1.0 | 21 de abril de 2016 | Nicolás Vásquez | Versión inicial |
| 2.0 | 15 de enero del 2018 | Álvaro A. Linares G. | Ejemplo con Thing Speak |