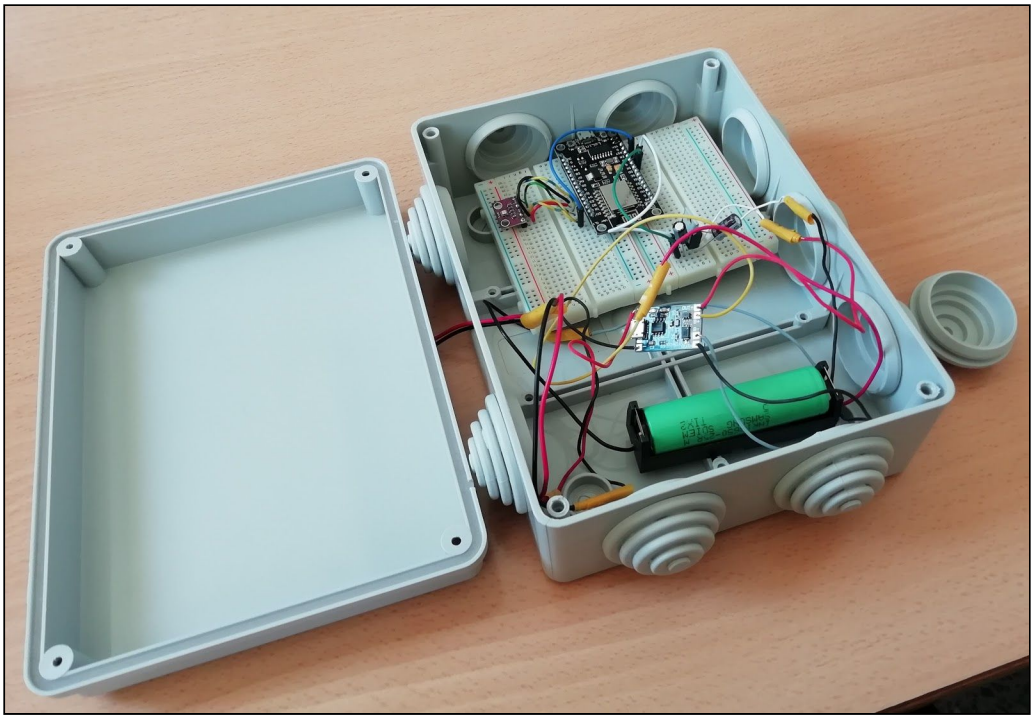
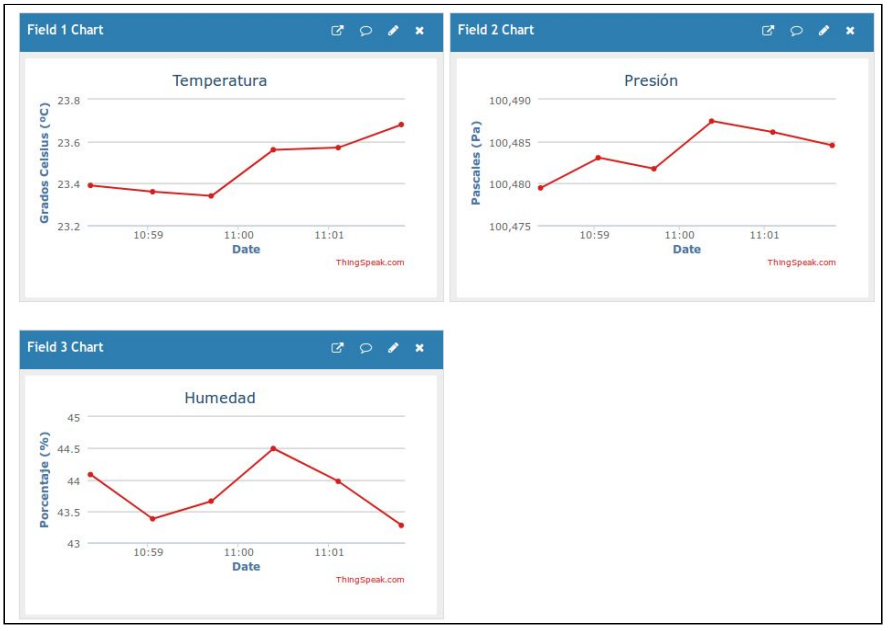
Estación Meteorológica





Grupo 4

Nuria Galve Contreras
Manolo García Martínez
Daniel Alexis Jurado Ortiz

IES VICENTE ALEIXANDRE
Departamento TIND - TIC
1º Bachillerato A y B

ÍNDICE

1. Finalidad del sistema

2. Búsqueda de información

3. Hardware
   Esquema de entradas y salidas
   Lista de materiales
   Esquema protoboard
   Esquema electrónico

4. Software

5. Evaluación
   Resultado final
   Problemas y soluciones
   Qué funciona bien y qué se puede mejorar
   Propuestas de mejora y ampliación

1. FINALIDAD DEL SISTEMA

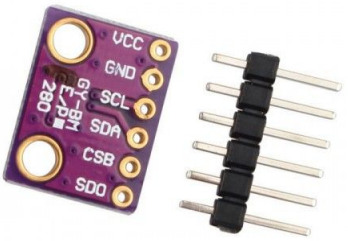
Realización de una estación meteorológica para el instituto que mida magnitudes como temperatura, humedad, presión atmosférica o velocidad del viento, y que se conecte a internet y pueda publicar los datos. Fases:

- **Fase 1 (PMV).** Construir una estación meteorológica con un sensor BME280 (que mide temperatura, humedad y presión) dentro de una cajita, y que envíe los datos a un Thingspeak.
- **Fase 2.** Solucionar problemas de energía del Node MCU ESP8266: modo sueño y placa solar.
- **Fase 3.** Mejorar el diseño de la caja, añadir más sensores (pluviómetro y anemómetro).

2. BÚSQUEDA DE INFORMACIÓN

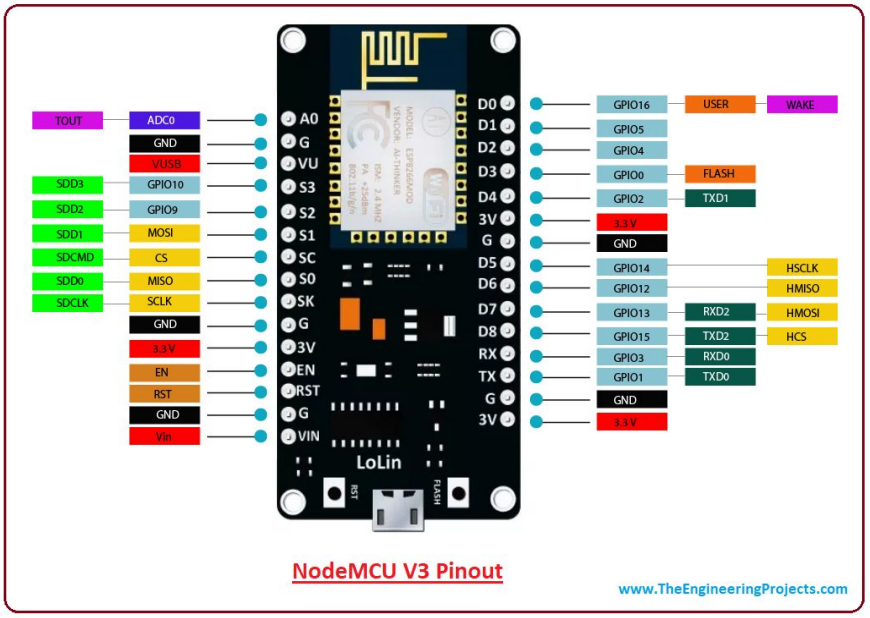
SENSOR BME280

El sensor BME280 es un sensor ambiental integrado desarrollado específicamente para aplicaciones móviles que consta de tres sensores: humedad, temperatura y presión atmosférica. Su reducido tamaño y bajo consumo de energía son puntos clave. Es programable con el IDE de Arduino mediante el uso de librerías específicas: "Adafruit_BME280 library".



NODE MCU ESP8266

El Node MCU es una placa que que lleva integrado el chip ESP8266, con conexión WiFi y compatible con el protocolo TCP/IP cuya función es dar acceso a cualquier microcontrolador a una red; es importante tener en cuenta que consume mucha energía. Nuestra placa Node MCU ESP8266 es una lolin, que tiene la siguiente disposición de pines:



Podemos hacer uso del Nodemcu copiando este link en el gestor de URLs de tarjetas del IDE de arduino. El ESP8266 consume mucha energía debido a su conexión WI-FI, por lo que se le puede aplicar tres tipos de modo sueño para reducirlo:

- **modem-sleep**: este modo de ahorro permite desactivar la conexión WiFi, establecida con un router, cuando no sea necesario su uso y volver a activarla cuando se necesite. El consumo típico en este modo es de 15mA.
- **light-sleep**: este modo de ahorro permite mantener la conexión WiFi, pero reduce el consumo de energía en los momentos en los que no hay envío de información. El consumo típico pasa a ser de unos 0,5 mA.
- **deep-sleep**: es el modo que genera mayor ahorro, pero a costa de dejar la placa en suspenso. La única parte de la placa que funciona durante este modo es reloj en tiempo real (Real Time Clock o RTC) para poder reiniciarla cuando haya finalizado el tiempo de reposo. El consumo típico pasa a ser de unos 10 uA. (Es el que hemos usado).

Hay cuatro posibilidades con respecto al modo de reinicio

- **WAKE_RF_DEFAULT**: cuando se reinicia el microprocesador, únicamente se calibra la señal de radio si el chequeo da error (init data byte 108>0). (Es el que hemos usado).
- **SLEEP_TIME, WAKE_RFCAL**: cuando se reinicia el microprocesador siempre se calibra la señal de radio. Esto incrementa el consumo.
- **SLEEP_TIME, WAKE_NO_RFCAL:** cuando se reinicia el microprocesador no se calibra la señal de radio. Esto reduce el consumo.
- **WAKE_RF_DISABLED**: cuando se reinicia el microprocesador se deshabilita la señal de radio (como en el modo modem sleep). Este es el modo con menor consumo, pero no permite ni enviar ni recibir datos vía WiFi.

PLACA SOLAR - GH165X135

Cada panel presenta celdas monocristalinas de alta eficiencia al 19% con un recubrimiento de uretano impermeable con respaldo de tablero duro. Con un sellado robusto para aplicaciones exteriores. Características:

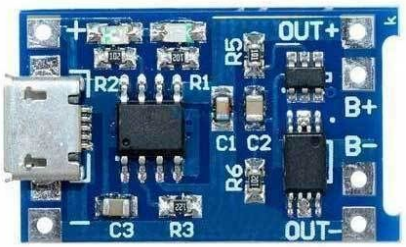
- Voltaje funcionamiento: 6V
- Potencia: 3.5W
- Corriente de Trabajo: 0 - 580mA
- Peso neto: alrededor de 100 g
- Tamaño: 165 * 135 * 2 mm

MÓDULO DE CARGA - TP4056

Módulo cargador basado en el chip TP4056. Perfecto para cargar una batería LiPo de tipo 18650 de 3.7V 1000mAh. Se puede alimentar a través del conector USB o con una tensión de 5V a los conectores IN+ IN-. Conectar la batería a los terminales B+ B-. Tiene un led rojo para indicar que la carga está en proceso y uno verde para indicar que el proceso de carga ha terminado. El módulo se auto desconecta cuando la carga está completa.

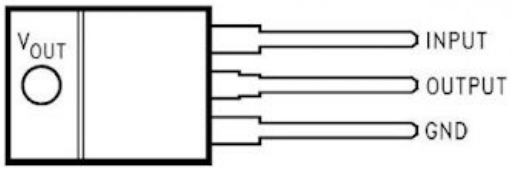
- Tensión de alimentación: 5V o USB
- Voltaje de carga completa: 4.2V

- Corriente de carga: 1A por defecto, no obstante podemos modificar este parámetro cambiando la resistencia de 1k2 que está junto al terminal IN-.



REGULADOR DE VOLTAJE - LM111T

El LM1117T-3.3/NOPB es un regulador de tensión de baja caída con una caída de tensión de 1’2v para una corriente de carga de 800mA. Su rendimiento oscila los 0 y 125°C y requiere un condensador de tantalio de un mínimo de 10µF en la salida para mejorar la estabilidad y respuesta a transitorios. Más datos en su: Datasheet. Presenta la siguiente configuración de pines:

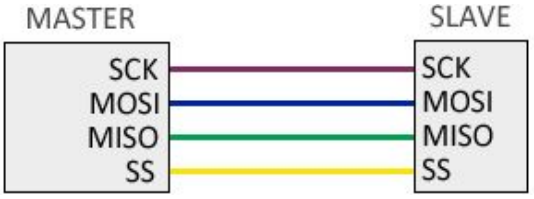


BUS SPI

El bus SPI (Serial Peripheral Interface) tiene una arquitectura de tipo maestro-esclavo. El dispositivo maestro (master) puede iniciar la comunicación con uno o varios dispositivos esclavos (slave), y enviar o recibir datos de ellos. Los dispositivos esclavos no pueden iniciar la comunicación, ni intercambiar datos entre ellos directamente.

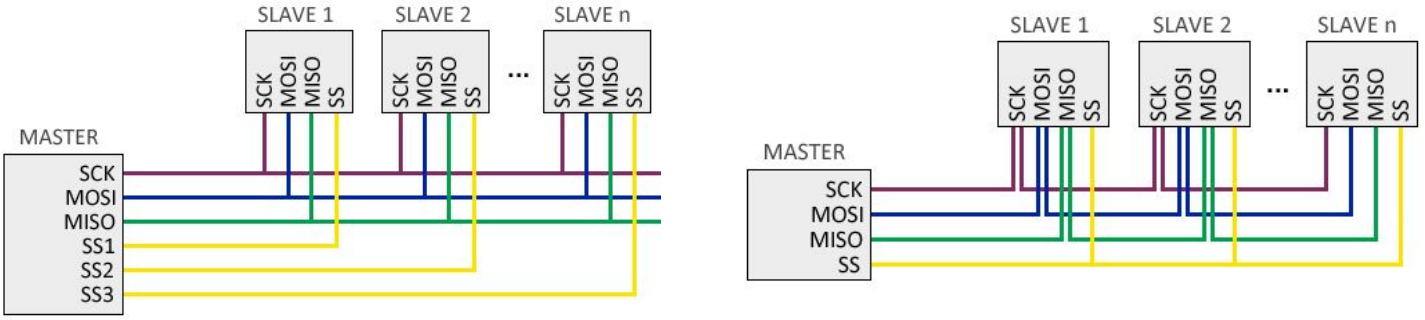
La comunicación entre maestro y esclavo se hace por líneas independientes, y el maestro puede enviar y recibir datos simultáneamente (comunicación Full Duplex). Es un bus síncrono lo que reduce la complejidad del sistema frente a los sistemas asíncronos. Por tanto, el bus SPI requiere un mínimo de 3 líneas:

- **MOSI** (Master-out, slave-in) para la comunicación del maestro al esclavo.
- **MISO** (Master-in, slave-out) para comunicación del esclavo al maestro.
- **SCK** (Clock) señal de reloj enviada por el maestro.



Además, se requiere una línea adicional **SS** (Slave Select) para cada dispositivo esclavo conectado, para seleccionar el dispositivo con el que se va a realizar la comunicación. Esto

es poco práctico cuando se tienen muchos dispositivos esclavos, por lo que es posible adoptar una conexión en cascada, donde cada esclavo transmite datos al siguiente.



Funcionamiento

Por defecto el maestro mantiene en estado HIGH todas las líneas **SS**. Cuando el maestro quiere establecer comunicación con esclavo pone a LOW la línea SS correspondiente, lo que indica al esclavo que debe iniciar la comunicación.

En cada pulso de la señal de reloj (que dispone el maestro), normalmente en el flanco de subida, el dispositivo maestro envía un bit del esclavo y a la vez recibe un bit del esclavo seleccionado.

La trama (los datos enviados) no sigue ninguna regla, es decir, podemos enviar cualquier secuencia arbitraria de bits. Esto hace que los dispositivos conectados necesiten tener pre-acordado la longitud y significado de los que van a enviar y recibir.

THINGSPEAK

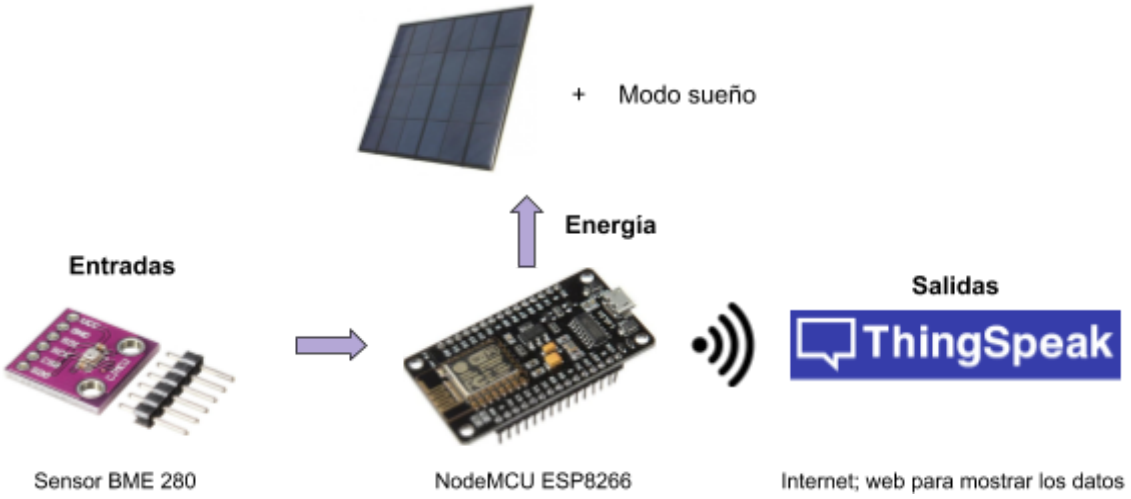
Thingspeak es una plataforma donde encontraremos múltiples opciones para recoger y almacenar datos con el fin de desarrollar aplicaciones IOT. Para usarla, nos ofrecen la posibilidad de ir guardando diferentes tipos de datos, cada cual organizado en un canal distinto. Para crear este canal tendremos que rellenar un formulario donde como máximo podremos registrar hasta 8 campos diferentes. Además de esto, permite exportar e importar datos, de manera que podemos almacenarlos y mirarlos cuando queramos.

LOCALIZACIÓN DE LA ESTACIÓN

- Debe estar situado sobre un terreno plano a poder ser sobre una superficie común en el área.
- El sensor debe estar colocado en un lugar ventilado y apartado. Nunca debe recibir directamente la luz del sol ya que si nos daría valores extremos.y debería estar alejado al menos 15 metros de fuentes de humedad como pueden ser fuentes, lagos, ríos, etc.

3. HARDWARE

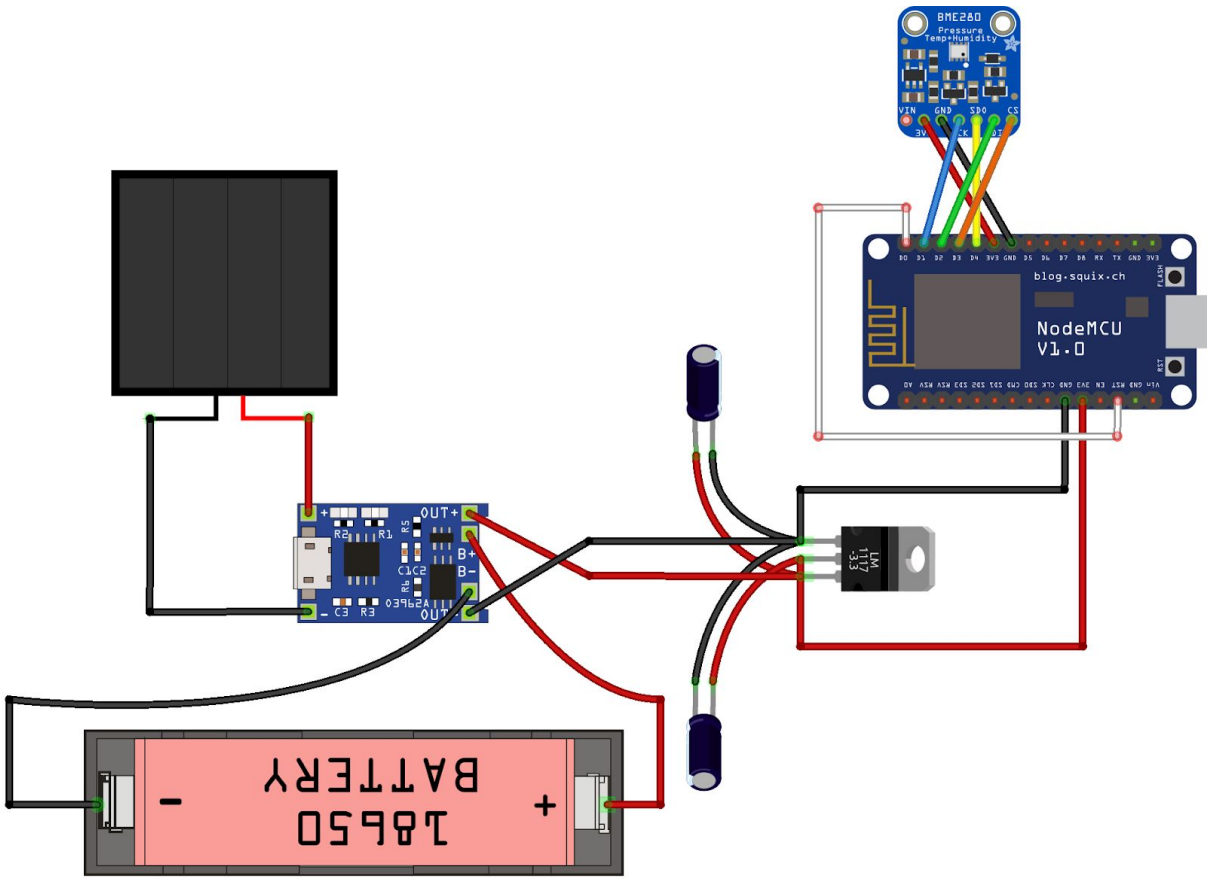
ESQUEMA ENTRADAS Y SALIDAS



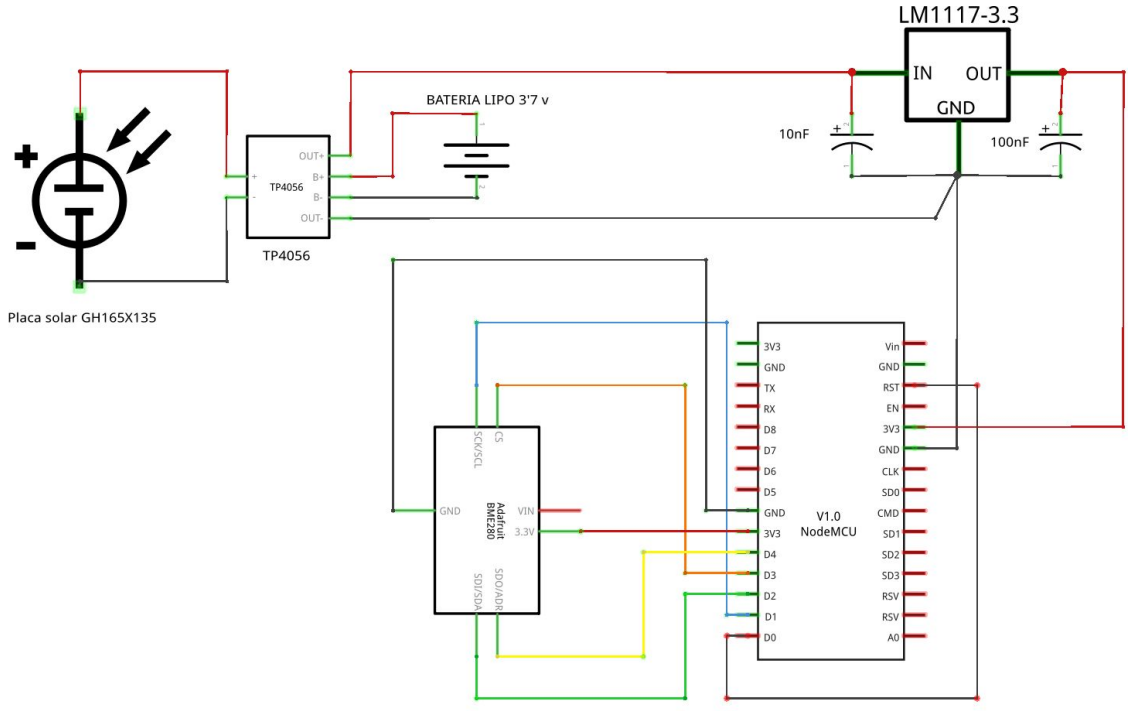
LISTA MATERIALES

Nº	Nombre	Cantidad	Precio
1	Caja Estanca	1	10.22 €
2	Placa Solar GH165X135	1	12 €
3	Módulo de carga TP4056	1	1.39 €
4	Batería Lipo 18650	1	5 €
5	Portapilas	1	1.48 €
6	Capacitor electrolítico 100µF	1	0.04 €
7	Capacitor electrolítico 10µF	1	0.01 €
8	Regulador de voltaje LM117T	1	0.52 €
9	Sensor BME 280	1	4.25 €
10	Node Mcu ESP8266	1	7.95 €
11	Cables de colores	17	1.26 €
		TOTAL	**44.12 €**

ESQUEMA PROTOBOARD



ESQUEMA ELECTRÓNICO



4. SOFTWARE

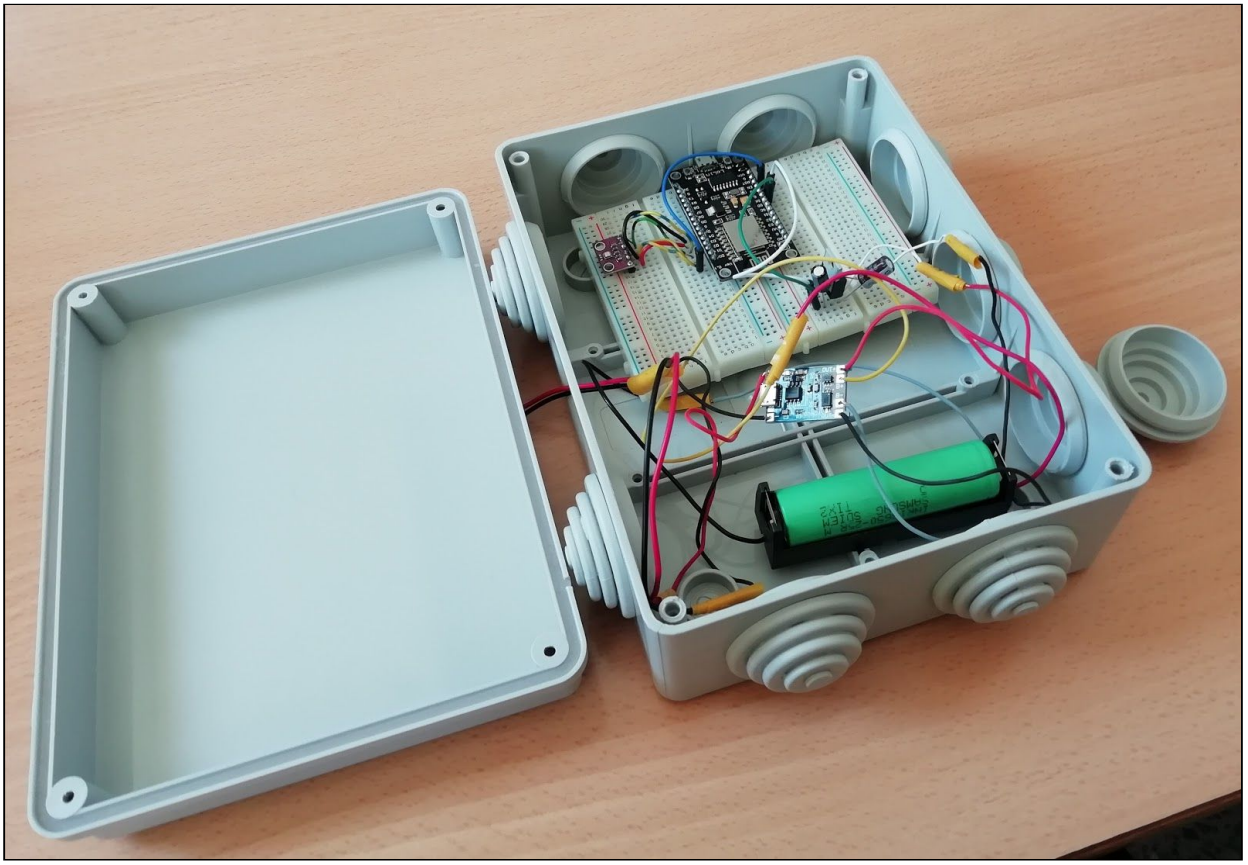
```
/*
 Estación Meteorológica
 TIND-TIC
 2018/19
 1º bachillerato A y B

 GRUPO 4
 Nuria Galve Conteras
 Manolo Garcia Martinez
 Daniel Alexis Jurado Ortiz
 */

//incluimos las librerías necesarias
#include <Wire.h>
#include <Adafruit_BME280.h>
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>

// asignar el bus del SPI a los pines
#define BME_SCK D1
#define BME_MISO D4
#define BME_MOSI D2
#define BME_CS D3

Adafruit_BME280 bme(BME_CS, BME_MOSI, BME_MISO, BME_SCK); //
software SPI

unsigned long myChannelNumber = 764210; //número del canal del
Thingspeak
const char * myWriteAPIKey = "CN82Z1DRLWJT83DU"; // APIKey del
Thingspeak
const char* ssid = "Andared"; //nombre del WiFi
const char* password = "llevalatararaunvestidoblancollenodecascabeles";
//contraseña del WiFi

const char* server = "api.thingspeak.com"; //dirección del servidor al que se
van a enviar los datos
WiFiClient client;

uint32_t TIEMPO_DeepSleep = 9e8;  // Tiempo en modo deep-sleep en
microsegundos
uint8_t TIEMPO_Referencia;        // Tiempo de referencia para encender 1
segundo el LED
```

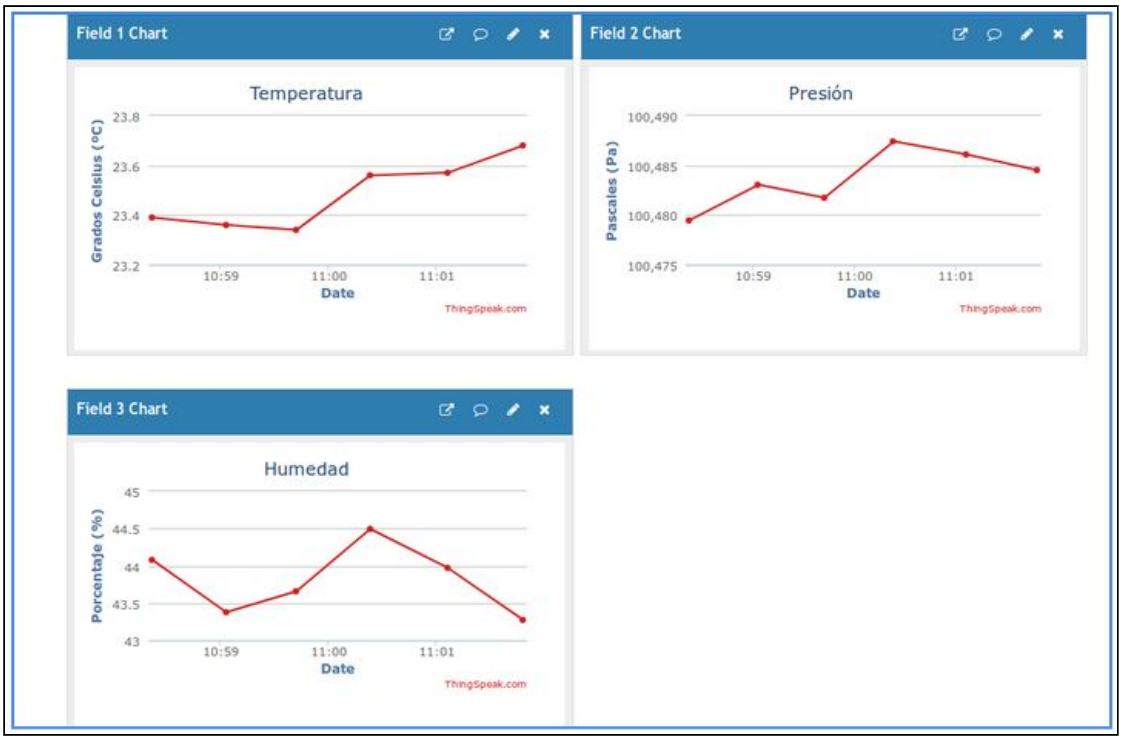
```
void setup() {
  Wire.begin(0, 2);
  Serial.begin(115200);

  WiFi.begin(ssid, password);

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  if (!bme.begin()) {
    Serial.println("No se encuentra un sensor BME280 válido, ¡comprobar el
cableado!");
    while (1);
  }
  ThingSpeak.begin(client);

  TIEMPO_Referencia = millis();
}

void loop() {
  float t = bme.readTemperature(); //lectura de la temperatura
  float p = bme.readPressure(); //lectura de la presión atmosférica
  float h = bme.readHumidity(); //lectura de la humedad

  //imprimimos en el puerto serie los datos
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" degrees Celcius Pressure: ");
  Serial.print(p);
  Serial.print(" Humidity: ");
  Serial.print(h);
  Serial.println("% send to Thingspeak");
```

```
  //Asignamos cada valor del sensor a su campo en el Thingspeak
  ThingSpeak.setField(1, t);
  ThingSpeak.setField(2, p);
  ThingSpeak.setField(3, h);
  ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

  //seleccionar tiempo para el modo sueño
  if (millis() - TIEMPO_Referencia > 1000) {
    ESP.deepSleep(TIEMPO_DeepSleep, WAKE_RF_DEFAULT); // Calibración de
señal de radio si es necesario
  }
}
```

5. EVALUACIÓN

RESULTADO FINAL





PROBLEMAS Y SOLUCIONES

PROBLEMAS	SOLUCIONES
La web Cayenne mostraba más inconvenientes de los que encontramos inicialmente.	Empezamos a utilizar Thingspeak, que nos resultó mucho más cómoda y que además, tenía una aplicación propia para móviles.
El módulo de carga TP4056 se quemó y dejó de funcionar.	Por suerte, contábamos con dos ejemplares, y usamos el otro.
Entre la placa solar y la caja quedaba un pequeño hueco por donde podía pasar aire, agua,etc	Utilizamos silicona para sellar la placa por los cuatro lados
La caja estanca quedaba totalmente cerrada y no entraba nada de aire	Retiramos varios de los tapones de goma para que el sensor estuviera a temperatura ambiente
La placa solar no alimentaba correctamente el sistema	Colocamos la estación meteorológica en un lugar en el que le daba correctamente la luz solar.
Se desconectaron los cables del portapilas	Los volvimos a soldar

QUÉ FUNCIONA BIEN Y QUÉ SE PUEDE MEJORAR

Hemos completado exitosamente la fase 1, del producto mínimo viable, y la fase 2, del ahorro de energía. El prototipo final de la estación meteorológica cumple con todas la funciones encomendadas por el equipo, dando valores exactos y con un grado de fiabilidad muy alta.

Respecto a la alimentación del sistema hemos superado nuestras expectativas que teníamos al comenzar el proyecto, ya que la continua realimentación por medio de la placa solar y la ventaja del modo sueño hace que la batería sea una fuente de alimentación eficaz y muy duradera.

Las gráficas del Thingspeak se ven perfectamente, tanto en el móvil como en el ordenador, con una continuidad adecuada que nos permite ver la evolución del tiempo. Hubiese sido un buen punto ser capaz de elegir que datos quieres ver en la gráfica, desafortunadamente, esa aplicación no existe en Thingspeak.

Como posible mejora, podríamos buscar una mejor localización del proyecto, el la que se aprovechen adecuadamente las horas de luz solar del día para una correcta y eficiente carga de la batería.

PROPUESTAS DE MEJORA Y AMPLIACIÓN

- Continuar con la fase 3 del proyecto, en la que se añadían más sensores a la estación meteorológica. Podríamos imprimir un anemómetro o un pluviómetro con la impresora 3D, e implementarlo al sistema.

  En la planificación del proyecto, al decidir que tipos de magnitudes medir, nos decantamos por las más comunes e importantes, y también propusimos varias mejoras en la estación dependiendo del tiempo que dispusieramos. Finalmente al determinar una fase de alimentación del microcontrolador muy exigente nos quedamos sin tiempo para seguir optimizando el proyecto. La intención de mejora que teníamos, la que sería la fase tres, consistía en añadir un pluviómetro y un anemómetro.

  El pluviómetro se podría haber realizado con la impresora 3D que disponemos en la clase y su función hubiese sido recolectar una cantidad de agua en un espacio determinado para posteriormente, con unos cálculos, poder constar de una aproximación fiable de la cantidad caída por metro cuadrado.

  Respecto al anemómetro, que calcula la velocidad del viento, no teníamos claro cómo hacerlo sin tener que comprar el aparato en sí, por lo tanto tendríamos que buscar más información o decantarnos por comprarlo para no enrevesar tanto el proyecto y obtener los valores de forma más precisa.

- Mejor distribución del circuito. Podríamos haber organizado mejor los cables y elementos dentro de la caja estanca, que están un poco desordenados.