



Manual de prototipo: Adquisidor de datos basado en Arduino

Imagen del prototipo



Figura 1: Adquisidor de datos basado en Arduino



Descripción general del prototipo

El adquisidor de datos mostrado en la Figura 1 está diseñado con la intención de sustituir un equipo comercial en la tarea de medir la temperatura a través de sensores. A su vez, se diseñó una carcasa que ofrece protección ante los elementos para facilitar su instalación en distintas ubicaciones.



Figura 2: Vista interior del adquisidor de datos

Los datos de temperatura obtenidos serán de utilidad para dimensionar e instalar sistemas de intercambiadores de calor tierra-aire. En la fase de operación de dichos sistemas y con adaptaciones menores, el prototipo puede expandir su funcionamiento a monitorear y/o controlar las condiciones de los sistemas intercambiadores de calor.



Instrumentación y calibración del prototipo

Para poder programar y utilizar el dispositivo de manera correcta es necesario contar con los siguientes materiales:

- Una computadora con Arduino IDE, conexión a internet y un programa para correr código python .
- Credenciales de acceso a un servidor con ThingsBoard donde se almacenarán los datos.
- Acceso a un baño térmico o equipo similar para calibración de los sensores de temperatura.

Una vez se cuente con la computadora es necesario instalar las siguientes librerías en Arduino IDE:

Librería	Autor
BlueDot BME280 Library	BlueDot
DallasTemperature	Miles Burton, Tim Bewsome, Et al.
OneWire	Jim Studt, Tom Pollard, Et al.
WifiEsp	bportaluri
Thingsboard	Thingsboard Team
PubSubCliente	Nick O'Leary
arduinohttpCliente	Arduino
arduinoJson	Benoit Blanchon

Tabla 1: Librerías utilizadas

Junto con este manual se entregan un total de 5 programas esenciales para configurar e instalar la sonda, en forma de comentarios, cada uno contiene sus respectivas instrucciones y consideraciones.

A continuación, se describe la preparación del prototipo y brevemente el papel que tienen los programas anexos:

Primeramente se debe determinar la cantidad de sensores de temperatura (DS18B20) que se utilizarán y conectarlos todos al mismo tiempo en las terminales de tornillo correspondientes de la sonda (ver figura 3), posteriormente se debe desconectar la entrada usb señalada con el número 2 en la figura anterior, conectarlo a la computadora y cargar el programa "01-nombrador-de-sensores-ds18b20", el cual contabilizará e identificará los sensores de manera individual, este programa y los siguientes 2 nos generarán líneas de código e instrucciones en la terminal de arduino IDE que deberemos seguir para configurar completamente el dispositivo.

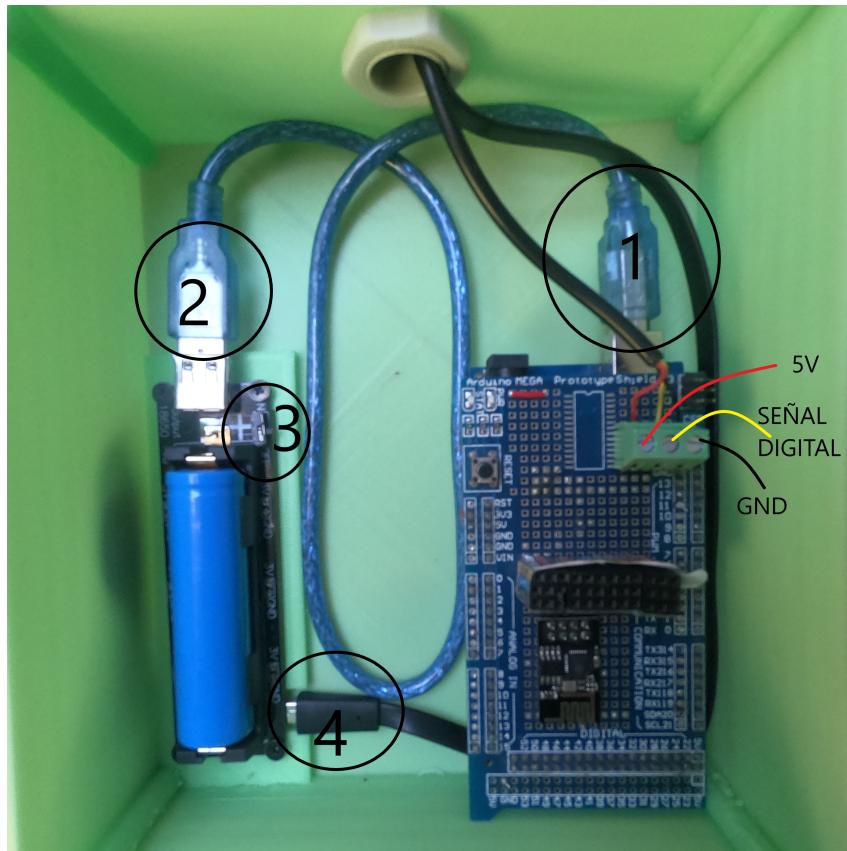


Figura 3: Guía de conexiones

En este punto el equipo tiene identificados todos los sensores y sin importar las veces que sean desconectados, reordenados o utilizados para otras pruebas, sin embargo, es necesario etiquetarlos físicamente para facilitar la calibración e instalación de los sensores, para esto se debe cargar el programa "02-Definicion-ds18" y podremos ver cada 3 segundos en la terminal de arduino IDE el número y la temperaturas que miden todos los sensores configurados anteriormente, podremos tomar los sensores con nuestros dedos individualmente, observar el número del sensor afectado y etiquetarlo de manera correspondiente con un pedazo de cinta y papel.



Posteriormente para la calibración continuaremos utilizando las temperaturas que el programa 02 nos provee. Se recomienda utilizar un baño térmico (ver figura 4), al menos 3 temperaturas conocidas y la función de regresión lineal en Excel para obtener la ecuación de cada sensor, sin embargo, se puede optar por una metodología distinta a consideración del usuario.



Figura 4: Proceso de calibración.

Una vez obtenidas las ecuaciones en formato $y = mx + b$ se debe cargar el programa "03-Monitoreo-Sonda" e introducir los valores encontrados de m y b para cada sensor en las líneas de código correspondiente (ver figura 5), una vez que agregamos todas las líneas de código generadas anteriormente en su respectivo lugar (ver figura 6), el prototipo está listo para desconectarse de la computadora e instalarse.



PARTE 1 DEL CODIGO

```
char coma=34;
float m_sensor_1=1;
float b_sensor_1=0;
float T1;
float m_sensor_2=1;
float b_sensor_2=0;
float T2;
```

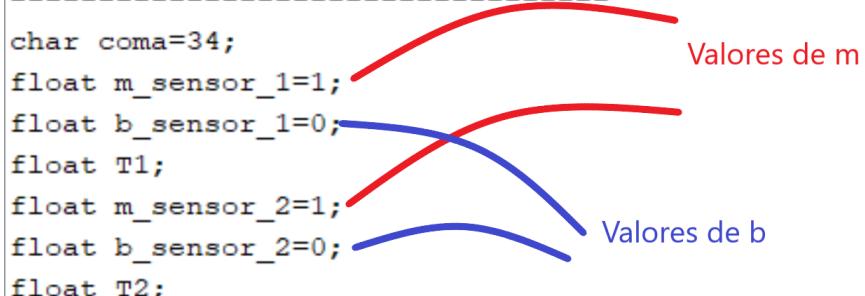


Figura 5: Ejemplo de líneas generadas por el programa 02.

```
-----CODIGO GENERADO POR PROGRAMA 01-----//  
uint8_t sensor1[8] = { 0x28, 0x8A, 0xC9, 0x03, 0x00, 0x80, 0x80, 0xEE};  
uint8_t sensor2[8] = { 0x28, 0xE9, 0xA2, 0x07, 0x00, 0x00, 0x80, 0x1B};  
int Nsensores=2;  
//-----PARTE 1 DEL CODIGO GENERADO POR PROGRAMA 02-----//  
char coma=34;
float m_sensor_1=1;
float b_sensor_1=0;
float T1;
float m_sensor_2=1;
float b_sensor_2=0;
float T2;
//  
void setup() {
  Serial.begin(9600);
  //Serial1 es la comunicación wifi
  Serial.println("Iniciando todo el programa");
  Serial1.begin(9600);
  InitWiFi();
  client.setServer(thingsboardServer, 1883);
  lastSend = 0;
  //Inicia la comunicación Serial
  sensors.begin(); //inicia los sensores de temperatura ds18b20
  Serial.println("Sensores iniciados");
  inicio=0;
}  
//-----PARTE 2 DEL CODIGO GENERADO POR PROGRAMA 02-----//  
void MedirTemp() {
  sensors.requestTemperatures();
  T1=m_sensor_1*sensors.getTempC(sensor1) + b_sensor_1;
  Serial.print((String)"T1="+(T1+"°C"));
  tb.sendTelemetryFloat("T1", T1);
}
```



Figura 6: Ejemplo del programa 03 terminado y listo para operar.

Utilizando de referencia la figura 3 el cable usb debe volver a su lugar inicial, el interruptor señalado con el número 3 debe pasar de OFF a ON y el cable al exterior de la carcasa se enchufa a un contacto regular de corriente alterna, hay tres indicadores LED que nos permiten saber si todo está conectado correctamente (ver figura 7) una vez confirmado esto se puede atornillar la tapadera y colocar el equipo en el lugar más conveniente, la batería de respaldo permitirá desconectar la carcasa y cambiar el equipo de posición sin preocuparse por perder datos.



Figura 7: Indicadores de funcionamiento.



Una vez instalado el equipo los datos pueden ser revisados directamente en ThingsBoard o a través del programa "04-importacion-datos" el cual debe ser ejecutado en python, siguiendo las instrucciones de dicho programa podremos obtener en formato CSV los datos de los sensores que deseemos y podremos generar gráficas para fechas específicas (ver figura 8).

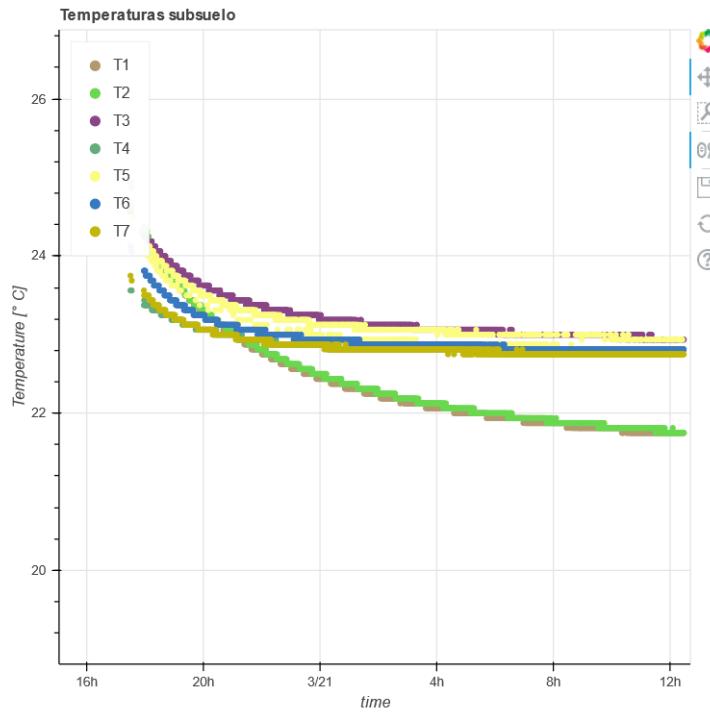


Figura 8: Datos de temperatura obtenidos con el prototipo.



Precauciones y recomendaciones

Es importante señalar que los programas 02 y 03 se limitarán a presentar errores si las líneas de código generadas por el programa 01 no son correctamente copiadas (ver figura 9).

Siempre que se inicie la operación del prototipo es necesario constatar que se están enviando los datos correctamente al servidor Thingsboard, la primera revisión se recomienda esperar 10 minutos después de conectar el equipo, en caso de no tener respuesta en el servidor es necesario volver a correr el programa 03 y leer con atención la terminal de arduino IDE para identificar algún posible error de conexión con la red wifi o servidor de ThingsBoard.

Posteriormente se recomienda vigilar los datos recibidos durante la primera semana de instalación, si después de esta semana no se detecta un comportamiento anormal en los datos se puede considerar una instalación exitosa, en caso contrario es necesario revisar las conexiones de los sensores al equipo y el aislamiento de los mismos.

En caso de la primera lluvia desde la instalación del prototipo es necesario esperar por lo menos 2 semanas y revisar nuevamente los datos, ya que con la humedad podrían fallar los sensores si no fueron correctamente aisladas las conexiones.

```
for (int i = 0; i < Nsensores; i++)
'Nsensores' was not declared in this scope
        sensors
exit status 1
'Nsensores' was not declared in this scope
```

Figura 9: Error por falta de código.



Descripción de las partes del prototipo

Tarjeta Arduino Mega 2560

Placa principal seleccionada debido a su costo, amplia compatibilidad con módulos (todos los compatibles con Arduino UNO) y canales de comunicación (4 RX y 4 TX).



Figura 10: Tarjeta Arduino Mega 2560.

Módulo WIFI ESP8266

Tarjeta de comunicación wifi confiable para establecer comunicación entre los sensores y el servidor ThingsBoard donde se almacenarán los datos.



Figura 11: Módulo WIFI ESP8266.



Tarjeta ProtoShield

Tarjeta utilizada para concentrar los componentes a modo de prueba y obtener un diseño para un pcb final.



Figura 12: ProtoShield.

Modulo XCSOURCE wemos 18650 batería Shield V3

Es el módulo encargado de suministrar un suministro continuo de energía al módulo en caso de que haya una desconexión por falla, mantenimiento o reubicación del módulo sin necesidad de perder datos.



Figura 13: Modulo XCSOURCE wemos 18650.



Sensores de temperatura DS18B20

El DS18B20 es un sensor digital de temperatura que utiliza el protocolo 1-Wire para comunicarse, este protocolo necesita solo un pin de datos para comunicarse y permite conectar más de un sensor en el mismo bus. Estos sensores nos permiten conectar por lo menos 100 simultáneamente sin ningún problema e identificarlos individualmente utilizando 1 solo cable hacia nuestra tarjeta Arduino son ideales para medir cambios graduales de temperatura como los del suelo, ambiente y líquidos, sin embargo, son catalogados como sensores “lentos” y por lo tanto se recomienda no utilizarlos para medir temperaturas instantáneas como la de una ráfaga de viento.



Figura 14: DS18B20.