

# testing workshop7

## **Materiales**

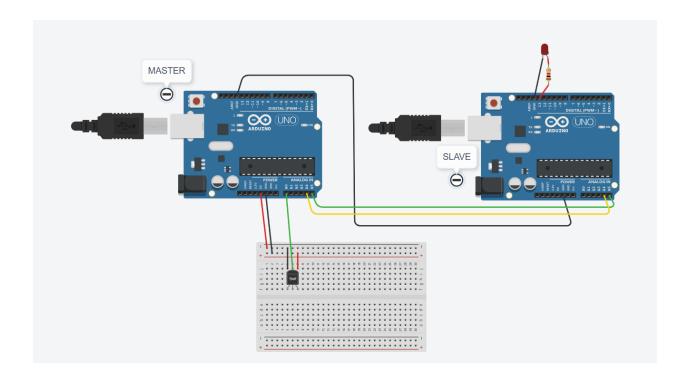
- protoboard
- arduino nano
- modulo esp32 wroom 32
- sensor lm35 o similares
- cables de alimentacion para cada modulo
- led
- resistencia

#### software

- IDE de programacion en este caso arduino IDE
- ThingSpeak plaform: dashboard

## **Ensamblar**

En tinkercad se realizo de la siguiente manera:



Como se requiere enviar la informacion a internet y contamos de ante mano con un modulo arduino nano que funcionara como esclavo. Usamos un modulo esp32 como maestro el cual leera la temperatura desde el pin 33 y la enviara por el bus de transmision con direccion 8 a esclavo. esto ya que este cuenta con un modulo Wifi para realizar el envio de informacion.

#### **MAESTRO**

```
#include <Wire.h> // use this library to connect and transmit data bia I2C
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
String URL = "http://api.thingspeak.com/update?api_key=K00XX903BUAHFNDV&field1=";

const char *ssid = "Familia_Cifu";
const char *pass = "77Familia9";

WiFiClient client;
HTTPClient http;

void setup() {
    Wire.begin(); // init communication
    Serial.begin(9600); // set data transfer rate
    WiFi.disconnect();
    delay(2000);
    Serial.print("Start connection");
```

```
WiFi.begin(ssid, pass);
 while((!(WiFi.status() == WL_CONNECTED))){
    delay(200);
   Serial.print("..");
 }
  Serial.println("Connected");
}
void loop() {
  int temperatura = getTemp_SENSOR(); // get temp
 if (temperatura >= 30.0) { // send temp if greater than 30
   Wire.beginTransmission(8); // transmit data by serial bus 8
   Wire.write('1'); // send 1
   Wire.endTransmission(); // close transmission
 } else {
   Wire.beginTransmission(8); // transmit data by serial bus 8
   Wire.write('0'); // send 0
   Wire.endTransmission(); // close transmission
 }
  Serial.println("Temperatura: "+ (String)temperatura);
 // send data
  String URL_ = URL + (String)temperatura;
 http.begin(client, URL_);
 int responsecode = http.GET();
  String data = http.getString();
 http.end();
  delay(1000);
}
float getTemp_SENSOR() {
  int lectura = analogRead(33); // read tension from A0
 float temperatura = lectura/10;
  return temperatura;
}
```

#### **ESCLAVO**

```
#include <Wire.h> // use this library to connect and transmit data bia I2C

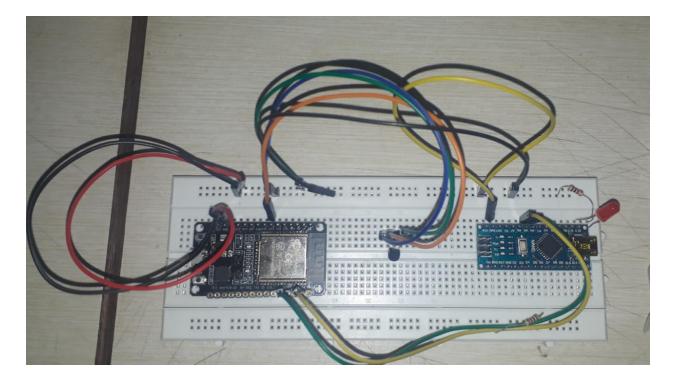
const int led_Pin = 13; // declare pin 13 as output warning

void setup() {
    Wire.begin(8); // listen data by serial bus 8
    Wire.onReceive(receiveEvent); // recive data
    pinMode(led_Pin, OUTPUT);
}

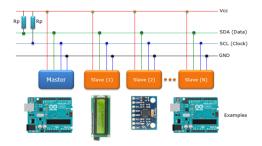
void loop(){// as slave just listen, slave does no need loop
}
```

```
void receiveEvent(int bytes) {
  if (Wire.available()) { // if exist data on serial bus 8
    char LED = Wire.read(); // LED is setted as 1 or 0

  if (LED == '1') {
    digitalWrite(led_Pin, HIGH); // temp greater than 30, led is on
  } else if (LED == '0') {
    digitalWrite(led_Pin, LOW); // temp lower than 30. led is off
  }
  }
}
```



En esta imagen se puede apreciar las conexiones realizadas, en la protoboard



Comunicacion I2C

- 1. Tener las conexiones hechas en la protoboard teniendo en cuenta la comunicación I2C
- 2. realizar la creacion de un canal en ThingSpeak
  - a. para este punto es necesario copiar el Chanel\_ID y el APIkey de escritura para realizar la conexion asi como el nombre de la red y la contraseña de esta.
- 3. Alimentar ambos modulos por separado
- 4. cargar al modulo esp32 por el puerto que coresponda COM# el codigo mostrado antes de master
- 5. cargar al modulo arduino nano por el puerto que corresponda COM# el codigo mostrado antes de esclavo
- 6. Ya con lo anterior se debe poder visualizar en ThingSpeak el la lectura realizada por el sensor