**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA**

**MECATRÓNICA**

****

**NOMBRE:** CHILIQUINGA CINTHYA, MASABANDA ALICIA

**DOCENTE:** ING. DAVID RIVAS

**FECHA DE ENTREGA:** 05 DE MARZO DEL 2021.

**TEMA:** MANUAL DE OPERACIÓN DEL MONITOR DE TEMPERATURA Y ENVIÓ DE DATOS A LA PLATAFORMA UBIDOTS.

|  |  |
| --- | --- |
| **ASIGNATURA** | **NRC** |
| DISEÑO ELECTRÓNICO | 3636 |

**NOVIEMBRE 2020 – ABRIL 2021**

# INTRODUCCION

El presente circuito tiene como finalidad medir la temperatura con la ayuda de un sensor y enviar datos a través del módulo WIFI ESP32 a la plataforma IoT (Ubidots).

# FUNCIONAMIENTO

Para utilizar este circuito solo se requiere una fuente de alimentación de 3.3 VDC y poder variar la temperatura. La fuente de alimentación se conecta en los pines especificados para el sensor de temperatura. El sensor permitirá receptar los datos de temperatura de un entorno y esta temperatura sensada se puede apreciar en la plataforma IOT pertinente.

# MATERIALES

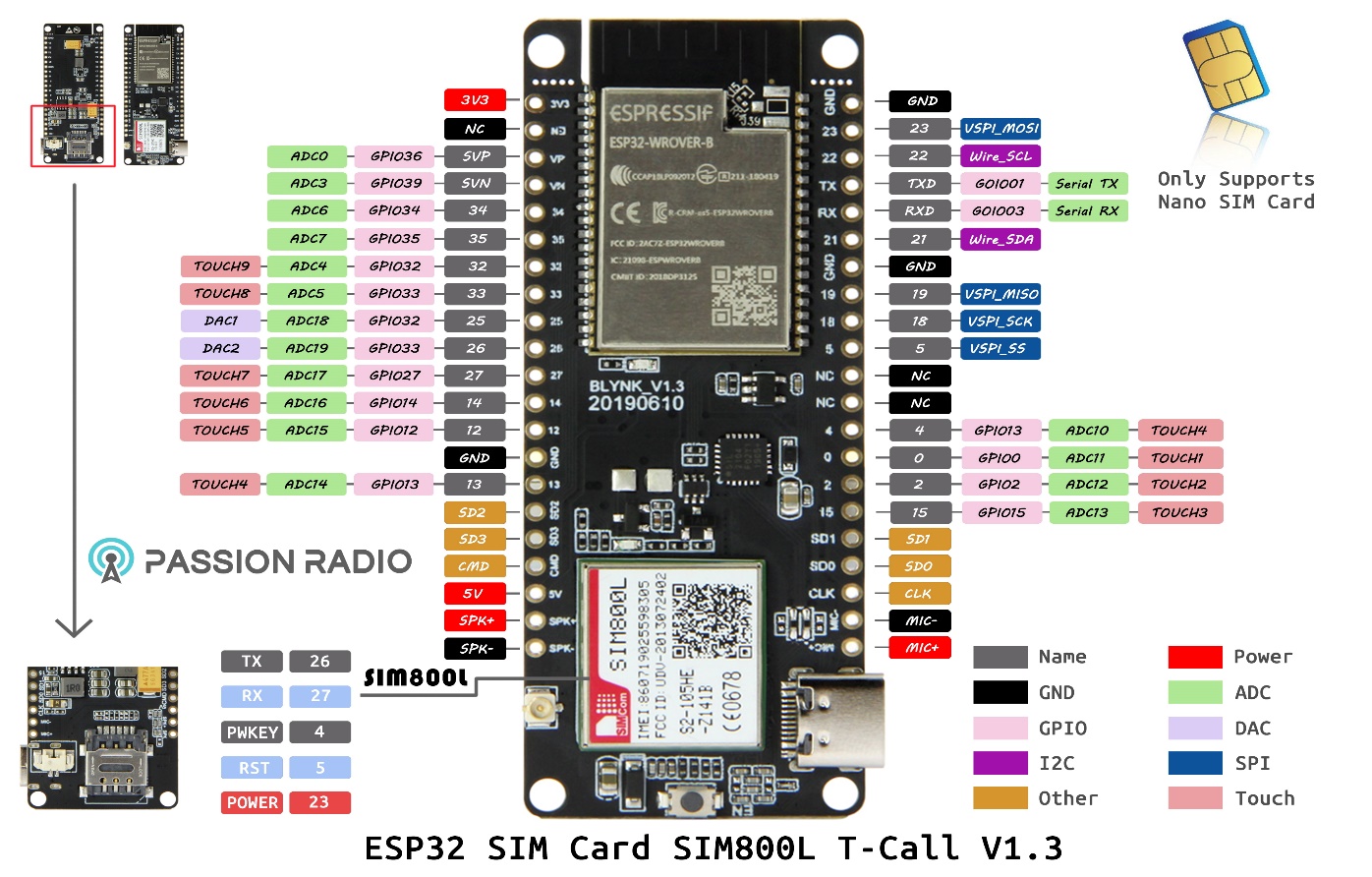
* 1 módulo ESP32

- 1 sensor lm35

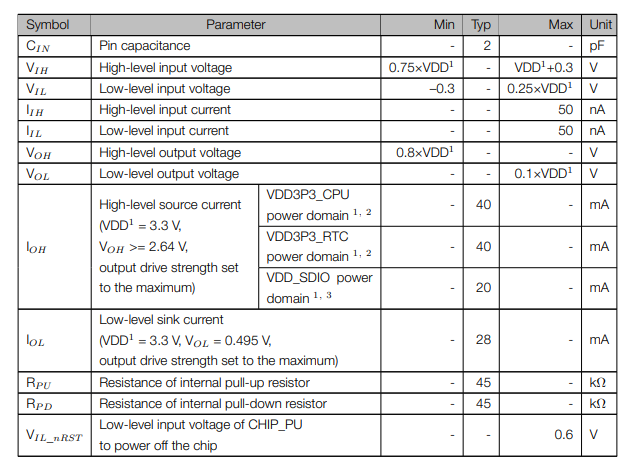
* Protoboard
* Cables de conexión
* Computadora
* LEDS

**DATOS TECNICOS DE LOS COMPONENTES**

**Módulo ES32**



**Especificaciones:**



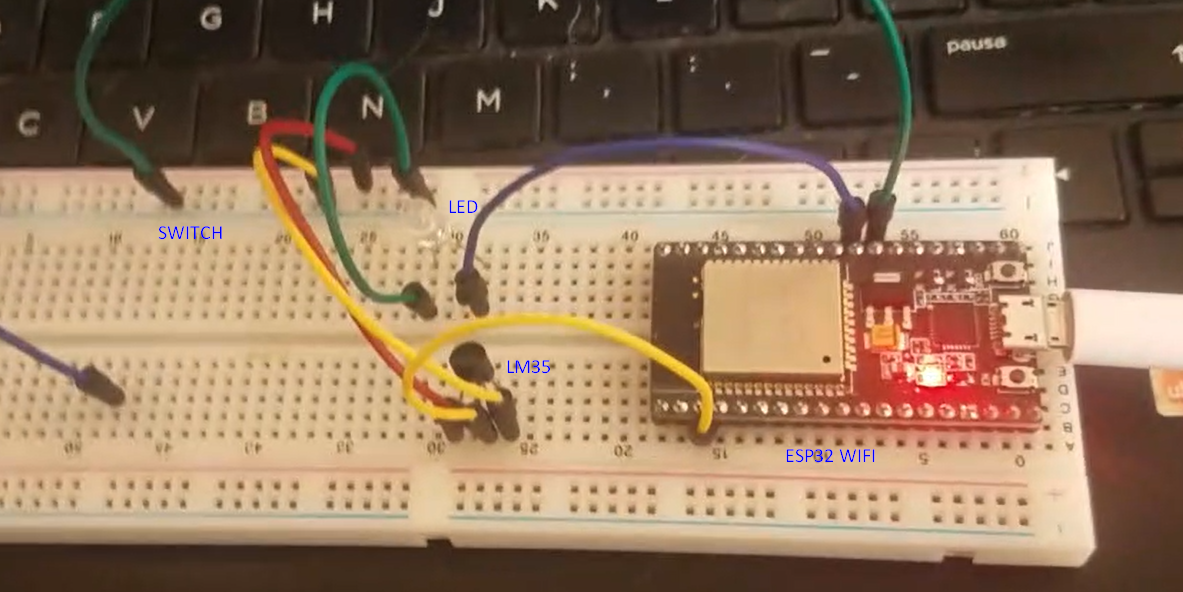
# LM35

# 

**Especificaciones:**

# 

**UBICACIÓN DE LOS COMPONENTES**



# CÓDIGO ARDUINO PARA UBIDOTS

# #include <WiFi.h>

# #include <PubSubClient.h>

# #define WIFISSID "PATRICIA Ajnet" // usuario de red

# #define PASSWORD "0502116510" // contraseña

# #define TOKEN "BBFF-9mml3ZMJRtDvE7xoVyIar2YRXqIEFd" // token de ubidots

# #define MQTT\_CLIENT\_NAME "Alexander\_Tello" // cliente

# #define SENSOR 0

# int ledd=17;

# int ventilador=16;

# //\*\*\*\*\*\*\*\*\*

# //defino dispositivo y variables de ubidots\*\*\*\*

# 

# #define VARIABLE\_LABEL "sensor" // variable de ubidots

# #define VARIABLE\_LABEL2 "lectura\_humedad" // variable de ubidots

# #define VARIABLE\_LABEL3 "led" // variable de ubidots

# #define VARIABLE\_LABEL4 "led\_virtual" // variable de ubidots

# #define DEVICE\_LABEL "esp32" // dispositivo de ubidots

# //\*\*\*\*\*\*\*\*\*

# //pins para el lcd\*\*\*\*\*\*\*

# char mqttBroker[] = "things.ubidots.com";

# char payload[100];

# char topic[150];

# char topicSubscribe[100];

# // Space to store values to send

# char str\_sensor[10];

# /\*\*\*\*\*\*

# \* Auxiliar Functions

# \*\*\*\*\*\*/

# WiFiClient ubidots;

# PubSubClient client(ubidots);

# /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# \* callback\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

# void callback(char\* topic, byte\* payload, unsigned int length) {

# char p[length + 1];

# memcpy(p, payload, length);

# p[length] = NULL;

# String message(p);

# int mensaje = message.toFloat();

# Serial.println(mensaje);

# 

# // encendido de led\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# if (message == "1.0") {

# digitalWrite(ledd, HIGH);

# }else{

# digitalWrite(ledd, LOW);

# }

# Serial.write(payload, length);

# Serial.println();

# }

# void reconnect() {

# 

# while (!client.connected()) {

# Serial.println("Attempting MQTT connection...");

# 

# // Attemp to connect

# if (client.connect(MQTT\_CLIENT\_NAME, TOKEN, "")) {

# Serial.println("Connected");

# client.subscribe(topicSubscribe);

# } else {

# Serial.print("Failed, rc=");

# Serial.print(client.state());

# Serial.println(" try again in 2 seconds");

# // Wait 2 seconds before retrying

# delay(2000);

# }

# }

# }

# /\*\*\*\*\*\*

# \* Main Functions

# \*\*\*\*\*\*/

# void setup() {

# 

# Serial.begin(115200);

# WiFi.begin(WIFISSID, PASSWORD);

# 

# pinMode(SENSOR, INPUT);

# pinMode(ledd,OUTPUT);

# pinMode(ventilador,INPUT);

# Serial.println();

# Serial.print("Wait for WiFi...");

# 

# while (WiFi.status() != WL\_CONNECTED) {

# Serial.print(".");

# delay(500);

# }

# 

# Serial.println("");

# Serial.println("WiFi Connected");

# Serial.println("IP address: ");

# Serial.println(WiFi.localIP());

# client.setServer(mqttBroker, 1883);

# client.setCallback(callback);

# 

# sprintf(topicSubscribe, "/v1.6/devices/%s/%s/lv", DEVICE\_LABEL, VARIABLE\_LABEL3);

# client.subscribe(topicSubscribe);

# }

# void loop() {

# if (!client.connected()) {

# reconnect();

# client.subscribe(topicSubscribe);

# }

# // lectura de señales analogas\*\*\*\*

# float sensor = analogRead(A0);

# float sensor\_humedad = analogRead(A3);

# // conversiones\*\*\*\*\*\*\*

# sensor=((sensor\*(150-55))/1023);

# sensor\_humedad=(sensor\_humedad\*(1023))/2000;

# 

# if (sensor\_humedad>=1000){

# Serial.println("El sensor se ha desconectado o esta fuera del suelo");

# }else if(sensor\_humedad<1000 && sensor\_humedad>=600){

# Serial.println("El suelo esta seco");

# }else if(sensor\_humedad<600 && sensor\_humedad>=370){

# Serial.println("El suelo esta humedo");

# }else if(sensor\_humedad<370){

# Serial.println("El sensor esta en el agua");

# }

# sensor\_humedad=(sensor\_humedad\*(100))/1023;

# 

# Serial.println("la temperatura es de: ");

# Serial.println(sensor);

# Serial.println("el sensor tiene una humedad de: ");

# Serial.print(sensor\_humedad);

# Serial.println("%");

# 

# 

# dtostrf(sensor, 4, 2, str\_sensor);

# sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE\_LABEL);

# sprintf(payload, "%s", "");

# sprintf(payload, "{\"%s\":", VARIABLE\_LABEL);

# sprintf(payload, "%s {\"value\": %s}}", payload, str\_sensor);

# client.publish(topic, payload);

# 

# int var=0;

# Serial.println("el ed esta en estado:");

# Serial.print("apagado");

# var=0;

# dtostrf(var, 4, 2, str\_sensor);

# sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE\_LABEL);

# sprintf(payload, "%s", "");

# sprintf(payload, "{\"%s\":", VARIABLE\_LABEL4);

# sprintf(payload, "%s {\"value\": %s}}", payload, str\_sensor);

# client.publish(topic, payload);

# int led2=digitalRead(16);

# while(led2==HIGH){

# led2=digitalRead(16);

# var=1;

# Serial.println("el ed esta en estado:");

# Serial.print("encendido");

# dtostrf(var, 4, 2, str\_sensor);

# sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE\_LABEL);

# sprintf(payload, "%s", "");

# sprintf(payload, "{\"%s\":", VARIABLE\_LABEL4);

# sprintf(payload, "%s {\"value\": %s}}", payload, str\_sensor);

# Serial.println("Publishing data to Ubidots Cloud");

# client.publish(topic, payload);

# }

# var=0;

# dtostrf(var, 4, 2, str\_sensor);

# sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE\_LABEL);

# sprintf(payload, "%s", "");

# sprintf(payload, "{\"%s\":", VARIABLE\_LABEL4);

# sprintf(payload, "%s {\"value\": %s}}", payload, str\_sensor);

# client.publish(topic, payload);

# client.loop();

# delay(1000);

# }

# INTERFAZ PLATAFORMA UBIDOTS

# 