

Resumen

En esta sección del proyecto Habeetat se construye el módulo encargado de medir la temperatura, humedad y concentración de CO2 dentro de la colmena. Se usa un microcontrolador ESP32-WROOM, un sensor de temperatura-Humedad DHT11 y un sensor de gases MQ135. El microcontrolador se programa para que mida las 3 variables y publique mediante el protocolo MQTT en un broker local, al mismo tiempo, se crea una base de datos con MySQL donde se guardan las lecturas de los sensores usando un flow creado en NodeRed; este flow nos ayuda a guardar las lecturas de los sensores en la base de datos y enviar notificaciones a un chat de Telegram usando un bot, el bot se programa para que reciba instrucciones y regrese consultas de la bases de datos, además de enviar notificaciones del estado de la colmena (alertas y datos) cada cierto periodo de tiempo. Los datos históricos y actuales recopilados son mostrados en gráficos en un dashboard de Grafana.

En este apartado se describe cómo se pueden implementar un sistema de monitoreo de la temperatura, humedad y concentración de CO2; en el interior de la colmena.

Software y bibliotecas a utilizar

- IDE de Arduino 1.8.20 <https://www.arduino.cc/en/software>
- NodeRed v3.0.2 <https://nodered.org/>
- Mosquitto 2.0.15 <https://mosquitto.org/>
- MySQL 8.0.30 - Ubuntu 22.04.1 (Ubuntu) <https://ubuntu.com/server/docs/databases-mysql>
- Grafana v9.1.7 <https://grafana.com/>
- Python 3.8 <https://www.python.org/downloads/release/python-380/>

Bibliotecas de Arduino para programar el microcontrolador y los sensores

- Wifi <https://github.com/arduino-libraries/WiFi>
- PubSubClient <https://github.com/knolleary/pubsubclient>
- DHT11 <https://github.com/adafruit/DHT-sensor-library>

Consideraciones para la realización del circuito, base de datos, flow de NodeRed, uso de la cámara, predicción y panel de Grafana.

Esta parte del proyecto hace uso de los siguientes archivos para su buen funcionamiento:

- [Programa del ESP32 WROOM](#)

- [Flow de node red para recibir, guardar y enviar datos, así como imágenes](#)
- [Programa para el ESP32 CAM para el envío de imágenes y vídeo al flow de Nodered](#)
- [Programa en Python para la predicción del peso de la colmena](#)

En las lecciones posteriores se explica con más detalle el funcionamiento ya que cada una de las partes interactúa entre sí para enviar y recibir la información.

Material necesario

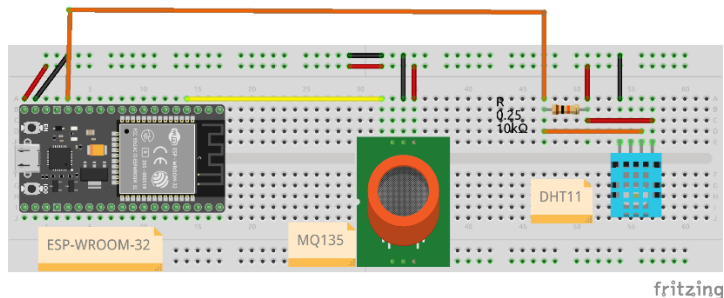
- Computadora
- 1 microcontrolador ESP32-WROOM
- 1 sensor DHT11
- 1 sensor MQ135
- 1 Protoboard
- Cables jumper MM
- 1 resistencia de 10 kilohms
- 1 cable microusb a USB

Material de referencia y servicios a usar

- Guía de instalación de Nodos de Telegram para Nodered (<https://flows.nodered.org/node/node-red-contrib-telegrambot>)
- Telegram bot API <https://core.telegram.org/bots/api>

Instrucciones para construir el circuito de medición de temperatura, humedad y concentración de CO2

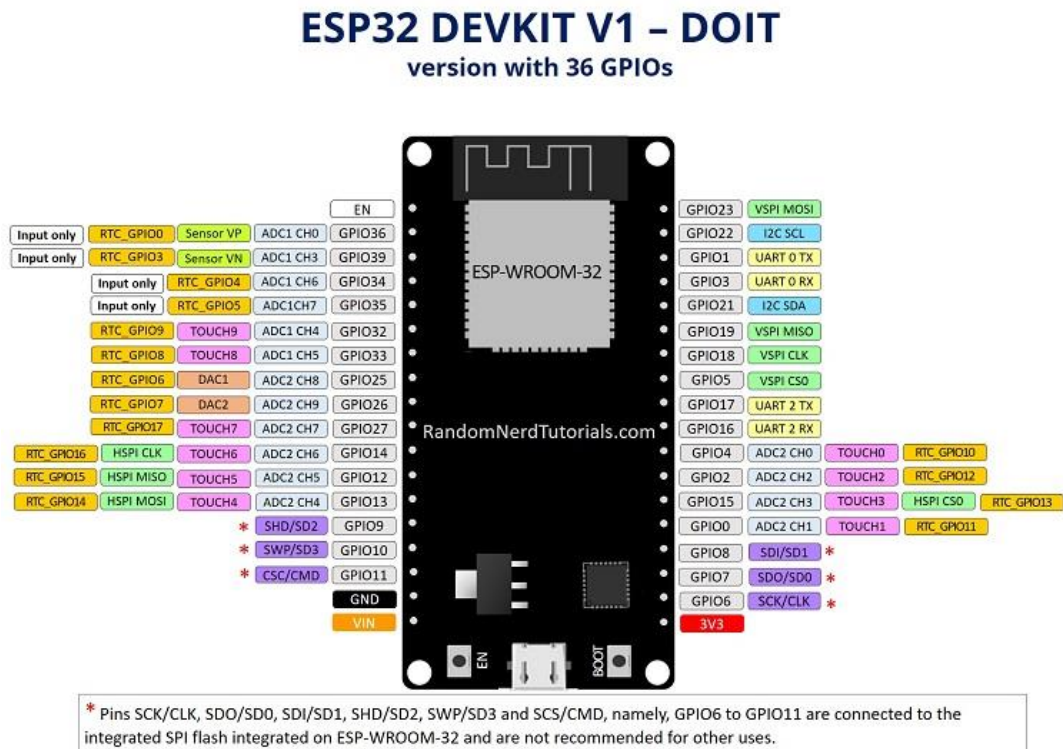
El circuito a crear es el siguiente



ATENCIÓN: Deben tomarse en cuenta las siguientes consideraciones para el armado del circuito:

No energizar el circuito hasta que todo esté debidamente conectado.

Los pines del ESP32 se consideran así de acuerdo a la siguiente nomenclatura

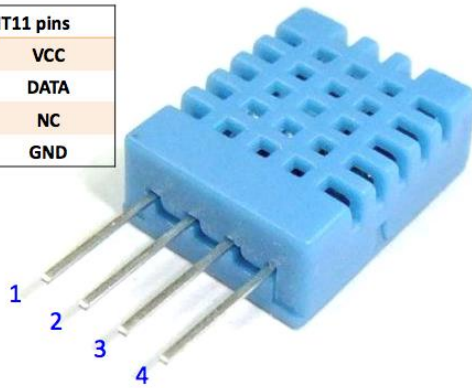


Los pines del sensor MQ135 son los siguientes:



Los pines del sensor DHT11 son los siguientes

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



El sensor MQ135 debe conectarse al microcontrolador a través de los siguientes pines

ESP32 -- MQ135

Vcc (5V) -- Vcc

GND -- GND

GPIO 34 -- A0

El sensor DHT11 debe conectarse al microcontrolador través de los siguientes pines

ESP32 -- DHT11

VCC (5V) -- VCC

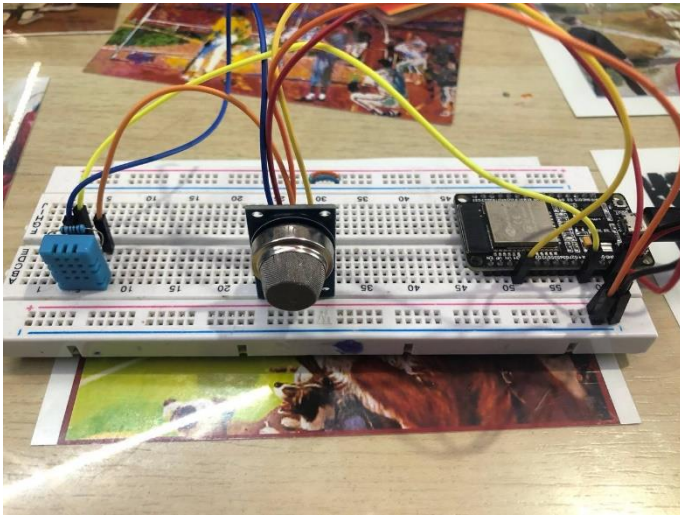
GND -- GND

GPIO 25 (14) -- DATA (2)

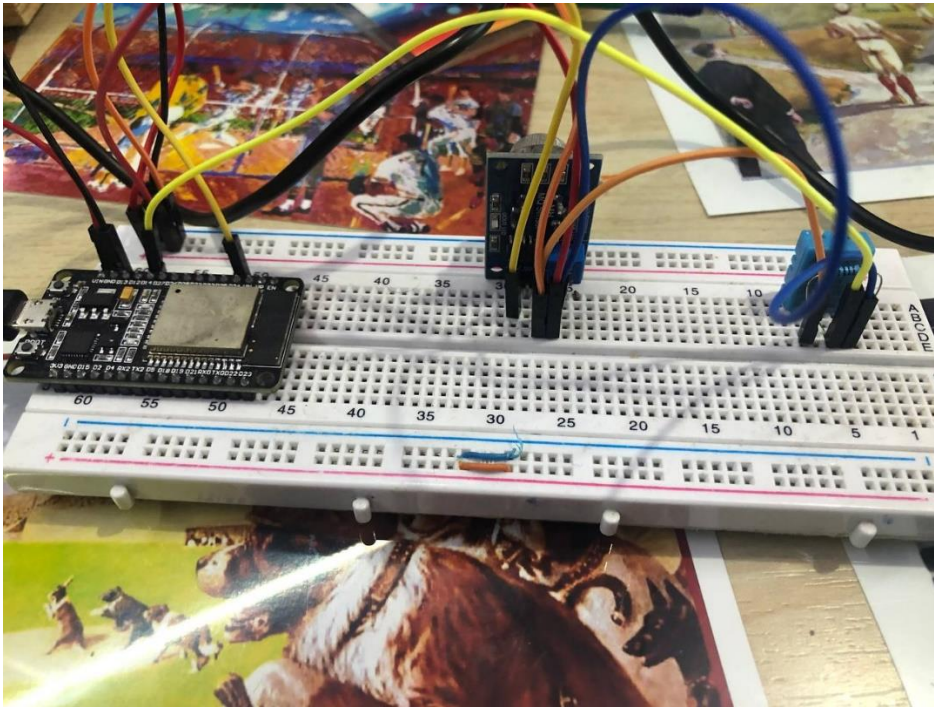
La tierra y la corriente se pueden conectar en paralelo para ser comunes.

La siguiente imagen muestra el circuito del MQ135 y el DHT11 conectados al ESP32 WROOM.

Vista frontal



Vista Trasera



Instrucciones para la programación del ESP32 WROOM

El programa para leer los sensores se encuentra en la siguiente carpeta del [repositorio de Github](#) . Este programa hace uso de las bibliotecas Wifi, PubSubClient y DHT11 por lo que es necesario instalarlas en el IDE de Arduino antes de subir el programa al ESP32, así como de preparar todo el entorno en sistema operativo (Ubuntu) para que se conecte el ESP32 WROOM a la computadora ([IDE de Arduino y ESP32](#)). Una vez que se realizan las lecturas del sensor éstas son publicadas en un broker local mediante el protocolo MQTT usando [Mosquitto](#) por lo que hay que configurarlo también en la computadora.

Una vez construido el circuito anterior se carga el programa usando el IDE de arduino,

El programa del ESP32 WROOM permite las siguientes acciones:

- Conexión del microcontrolador a Wifi
- Lectura del Voltaje analógico del sensor MQ-135 (concentración de CO₂)
- Lectura de Temperatura y Humedad con el sensor DHT11
- Publicación y suscripción a tópicos en el broker local usando MQTT
- Envío de los datos en formato JSON al broker local

Una vez conectado el ESP32 adquiriendo los datos de temperatura, humedad y CO2 creamos un flow en NodeRed ([Link del flow de NodeRed](#)) que realiza las siguientes acciones:

- Lectura y adquisición del archivo JSON con los datos desde el broker local hecho en MQTT para:
 1. Guardar los datos en una base de datos local hecha con MySQL ([Link detalles de la creación de base de datos](#)).
 2. Generar reportes con el bot de telegram desde nodered ([Link de los detalles para creación del bot de telegram con node red](#)).
 - El bot envía reportes programados o permite realizar consultas de los datos más recientes.
 3. Envío de imágenes y vídeo con la esp32CAM y el bot de telegram desde nodered ([link del programa del esp32CAM](#)).
 4. El flow de node-red también tiene un panel donde podemos visualizar vídeo en tiempo real desde la colmena, esta cámara se sitúa en el exterior de la colmena para vigilancia.
 5. Envío de la predicción del peso de la colmena en los siguientes días ([Link del código de predicción](#)).
 6. Cuando las condiciones de Temperatura, humedad o concentración de CO2 no son las favorables para la colmena también se envía un mensaje de alerta.

Creación de la base de datos MySQL para los datos del interior de la colmena

La base de datos se creó en una máquina virtual de Ubuntu 20.04 usando MySQL. Los datos obtenidos de los sensores MQ135 y DHT11 son enviados por MQTT en formato json al broker local donde son leídos por el flow de NodeRed y guardados en nuestra base de datos local usando el nodo MySQL. Posteriormente esta base de datos es usada como fuente de datos para un panel de Grafana donde visualizamos el comportamiento local.

Software a utilizar

Máquina virtual con Ubuntu 20.04

My Sql para Linux

Instrucciones para creación de base de datos

1. Abrir una terminal e instalar mysql server con los comandos

- *sudo apt update*
- *sudo apt install mysql-server*

2. Ejecutar mysql

- *sudo mysql*

3. Crear una nueva base de datos para guardar los datos de los sensores

- *CREATE DATABASE sensor_MQ_DHT;*

4. Seleccionar base de datos

- *use sensor_MQ_DHT;*

5. Crear una tabla llamada Datos_Sensores que contenga todos los campos necesarios

- *CREATE TABLE Datos_Sensores (n_registro INT(7) UNSIGNED AUTO_INCREMENT PRIMARY KEY, fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP, id_sensor CHAR (248) NOT NULL, ValorAnalogico FLOAT(7,3) NOT NULL, VoltajeAnalogico FLOAT(7,3) NOT NULL, Temperatura FLOAT(7,3) NOT NULL, Humedad FLOAT(7,3) NOT NULL);*

Esto creará la base de datos que guardará nuestros datos que son

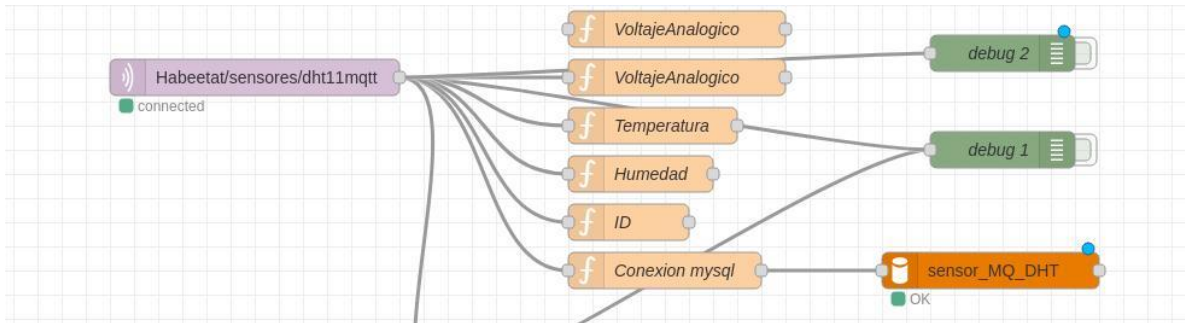
- id_sensor
- Timestamp
- ValorAnalogico
- VoltajeAnalogico
- Temperatura
- Humedad

6. Se requiere crear un usuario y contraseña para poder ingresar posteriormente desde Grafana a nuestra base de datos, lo creamos usando las instrucciones

- `CREATE USER 'raymundo_soto'@'localhost' IDENTIFIED BY 'contrasena';`
- `GRANT ALL PRIVILEGES ON *.* TO 'raymundo_soto'@'localhost';`

Esto nos permitirá el acceso a nuestra base de datos local.

7. Para guardar desde Node se usan los siguientes Nodos mqtt, función y MySQL



- El nodo mqtt se conecta al tópico *Habeetat/sensores/dht11mqtt* del broker local donde publica el ESP32
- El nodo función *Conexión mysql* se programa para que publique los datos en la base de datos creada anteriormente, el código del nodo es:

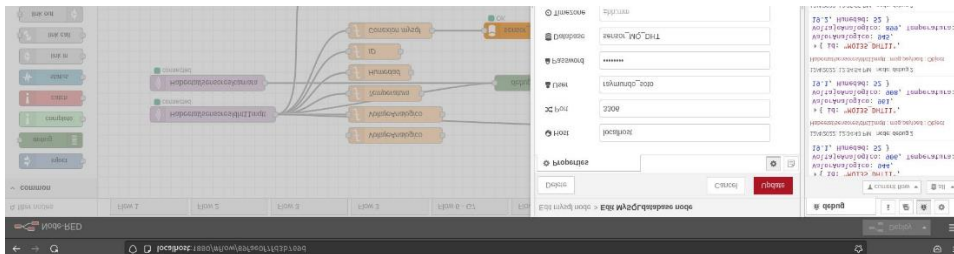
```
msg.topic = "INSERT INTO Datos_Sensores (id_sensor, ValorAnalogico, VoltajeAnalogico,
Temperatura, Humedad) VALUES (" + msg.payload.id + ",
```

```
" + msg.payload.ValorAnalogico + "," + msg.payload.VoltajeAnalogico + "," +
msg.payload.Temperatura + "," + msg.payload.Humedad + ");"
```

```
return msg;
```

Este nodo estará enviando los datos a la tabla *Datos_Sensores* para guardarlos en la base de datos.

* El nodo mysql *sensor_MQ_DHT* se configura para conectarse la base creada en el punto 3; la database debe ser *sensor_MQ_DHT*, el Host localhost, en el puerto 3306 y el usuario y contraseña del punto 6.



Una vez creada la base de datos podemos usarla para visualizar datos en Grafana o tomarlos para hacer predicciones de comportamiento.

Resultados

La base de datos con la tabla creada se muestra a continuación

```
raymundo@raymundo-VirtualBox: ~  
(base) raymundo@raymundo-VirtualBox:~$ sudo mysql;  
[sudo] password for raymundo:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 11  
Server version: 8.0.31-0ubuntu0.20.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| RFID |  
| codigoIoT |  
| datosclima |  
| detectorsintomas |  
| information_schema |  
| mysql |  
| performance_schema |  
| sensor_MQ_DHT |  
| sensormq135 |  
| sys |  
| tiktok |  
+-----+  
11 rows in set (0.02 sec)  
  
mysql> use sensor_MQ_DHT  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_sensor_MQ_DHT |  
+-----+  
| Datos_Sensores |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

Los datos guardados desde el flow de nodered se muestra en esta imagen

```
+-----+  
| 16402 | 2022-12-04 12:32:08 | MQ135_DHT11 | 976.000 | 893.000 | 19.000 | 53.000 |  
| 16403 | 2022-12-04 12:32:19 | MQ135_DHT11 | 971.000 | 940.000 | 19.000 | 52.000 |  
| 16404 | 2022-12-04 12:32:30 | MQ135_DHT11 | 983.000 | 951.000 | 19.100 | 52.000 |  
| 16405 | 2022-12-04 12:32:41 | MQ135_DHT11 | 962.000 | 910.000 | 19.100 | 52.000 |  
| 16406 | 2022-12-04 12:32:52 | MQ135_DHT11 | 959.000 | 899.000 | 19.100 | 51.000 |  
| 16407 | 2022-12-04 12:33:03 | MQ135_DHT11 | 965.000 | 908.000 | 19.100 | 51.000 |  
| 16408 | 2022-12-04 12:33:14 | MQ135_DHT11 | 962.000 | 906.000 | 19.100 | 51.000 |  
| 16409 | 2022-12-04 12:33:25 | MQ135_DHT11 | 931.000 | 910.000 | 19.100 | 52.000 |  
| 16410 | 2022-12-04 12:33:36 | MQ135_DHT11 | 947.000 | 909.000 | 19.100 | 51.000 |  
| 16411 | 2022-12-04 12:33:47 | MQ135_DHT11 | 959.000 | 910.000 | 19.100 | 51.000 |  
| 16412 | 2022-12-04 12:33:58 | MQ135_DHT11 | 956.000 | 901.000 | 19.100 | 52.000 |  
| 16413 | 2022-12-04 12:34:10 | MQ135_DHT11 | 947.000 | 911.000 | 19.100 | 52.000 |  
| 16414 | 2022-12-04 12:34:21 | MQ135_DHT11 | 957.000 | 893.000 | 19.100 | 52.000 |  
| 16415 | 2022-12-04 12:34:32 | MQ135_DHT11 | 944.000 | 906.000 | 19.100 | 52.000 |  
| 16416 | 2022-12-04 12:34:43 | MQ135_DHT11 | 961.000 | 908.000 | 19.100 | 52.000 |  
| 16417 | 2022-12-04 12:34:54 | MQ135_DHT11 | 945.000 | 899.000 | 19.200 | 52.000 |  
| 16418 | 2022-12-04 12:35:05 | MQ135_DHT11 | 938.000 | 899.000 | 19.200 | 52.000 |  
+-----+  
16418 rows in set (0.17 sec)  
  
mysql> SELECT * FROM Datos_Sensores;
```

Ya tenemos lista la base de datos.