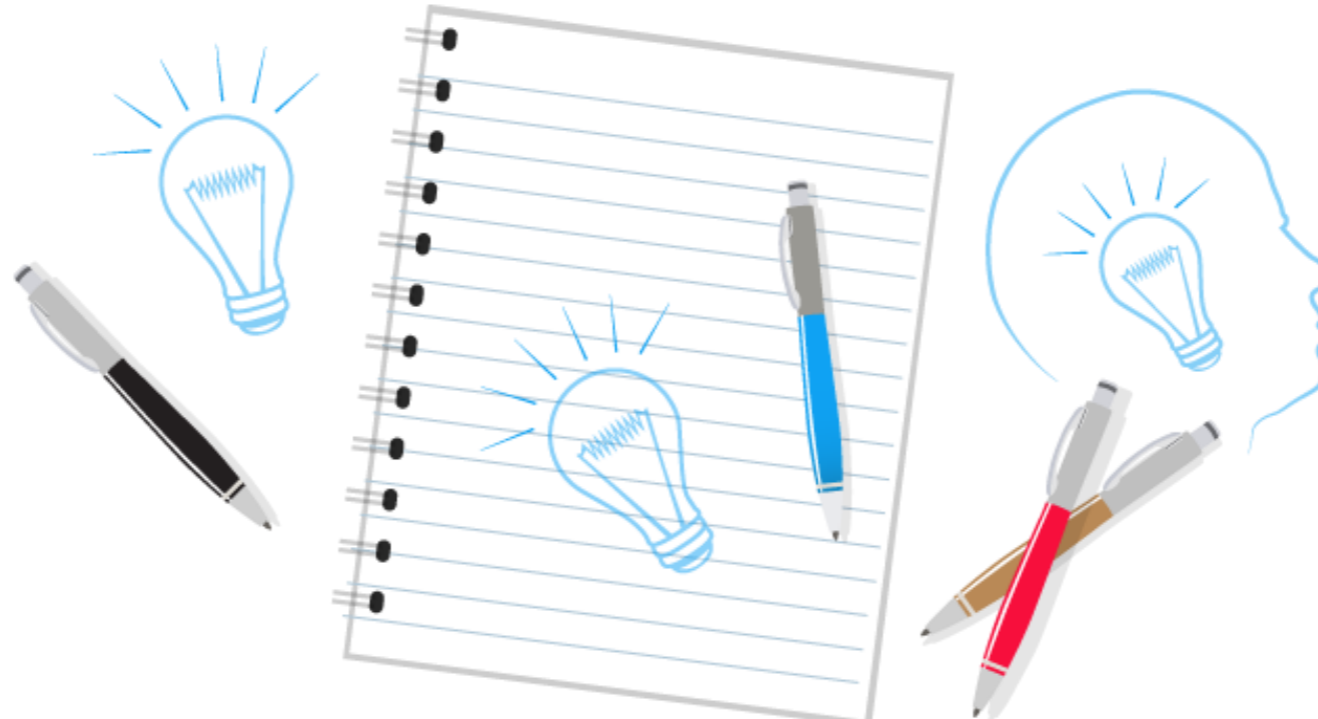


# CAPÍTULO 12. M5STACK

## v.1.1 MARZO 2024

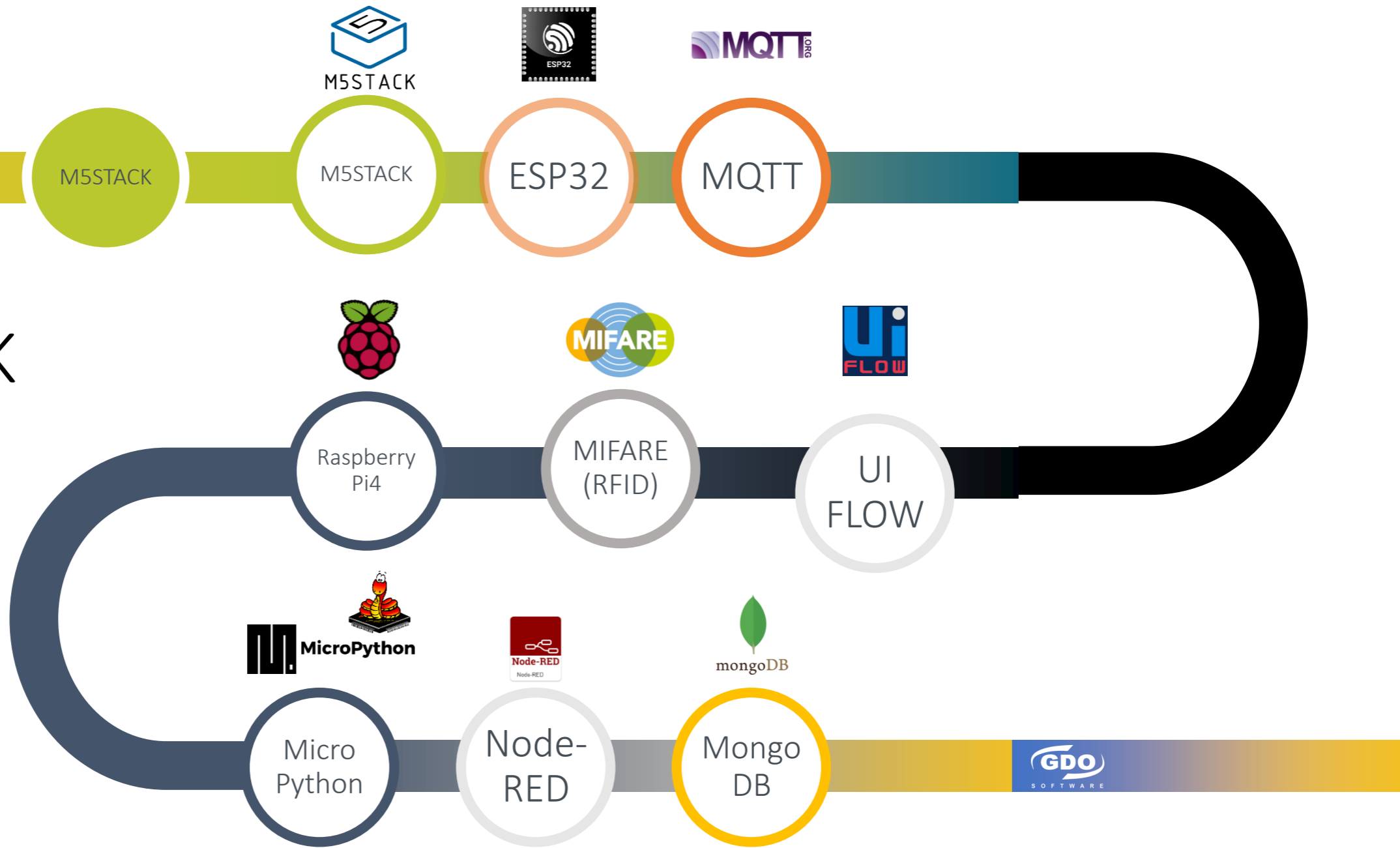
**Ricardo Moraleda Gareta**

[Director departamento de software de GDO Software]



# M5STACK

v.1.1 MARZO 2024





# M5STACK

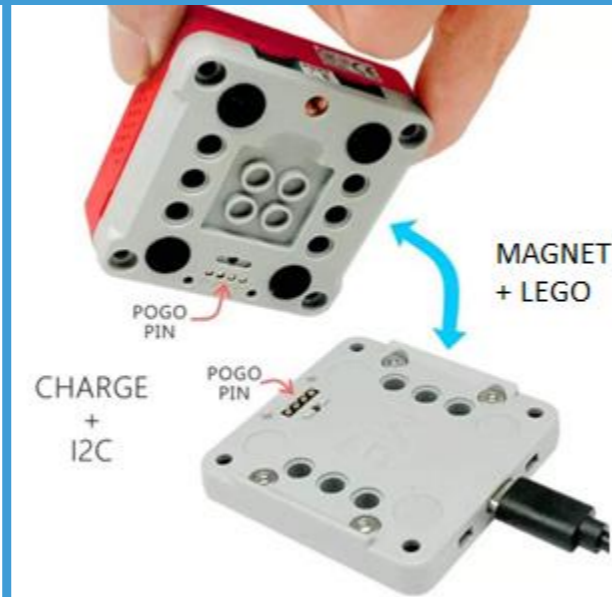


## M5STACK FIRE

## Componentes y sensores

FIRE es un dispositivo central M5. Su dispositivo modular, apilable, escalable y de mano se ejecuta en un núcleo **ESP32**, lo que permite el uso de código abierto, bajo costo, funcionalidad completa y facilidad de desarrollo.

Dispone de LCD, RGB bars, buttons, speaker, TF card.  
Dispone de una batería extra apilable (magnética)  
También 3 tipos de puertos para sensores/actuadores (units) con conector GROVE



- A: I2C
- B: I/O
- C: UART

GROVE PORT	CABLE				
A I2C	Black	Red	Yellow	GND	5V SDA SCL
B I/O	Black	Red	Blue	GND	5V G26 G36 In/Out Input
C UART	Black	Red	White	GND	5V TXD RXD



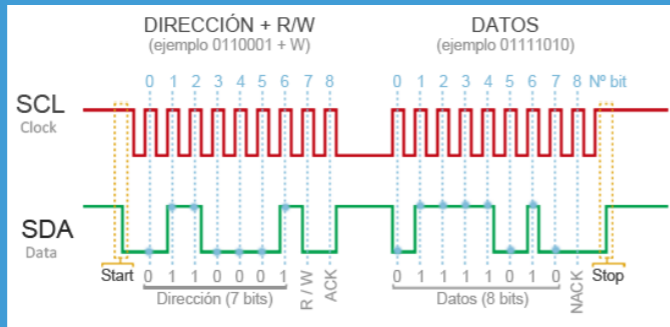
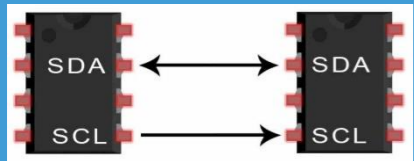


# PUERTOS



## I2C (Inter-Integrated Circuit)

Es un bus serie de datos desarrollado en 1982 por Philips Semiconductors.

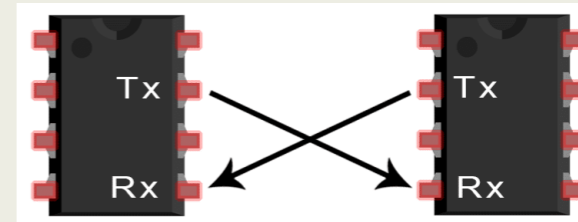


Algunos sensores de M5STACK (I2C)



## UART – I/O

UART (Universal Asynchronous Receiver-Transmitter). Su función principal es convertir los datos serie a paralelos cuando se trata de datos recibidos (de entrada) y de convertir datos paralelos a serie para transmisión (de salida). Es el puerto COM.



I/O (Input/Output). 1 o 0 (binarios).







# MÓDULOS



## Módulos

La ventaja del M5STACK es que es apilable. Puedes añadir tantos módulos como desees haciendo una pila, dejando arriba el core.

Existen varios módulos disponibles como:

LORAWAN, GPS, GSM, LTE, SERVO/STEP MOTOR, etc.



## Faces

Dispone de hardware adicional para control de interfaz de usuario.

Calculator, Finger, GameBoy, Joystick, Keyboard, RFID, etc.





# FIRMWARE



## M5STACK FIRE

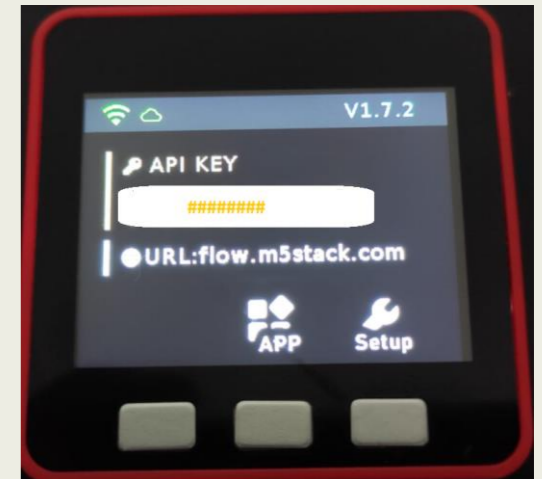
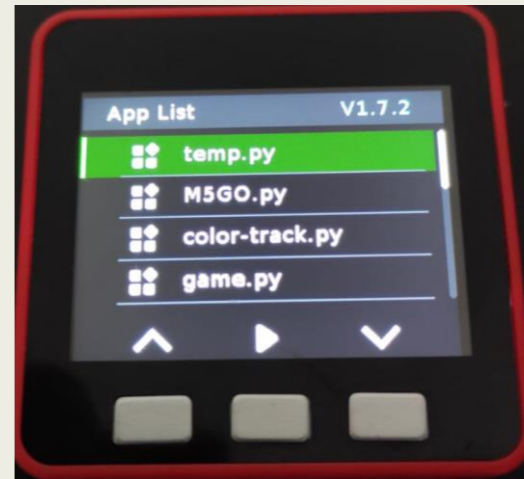
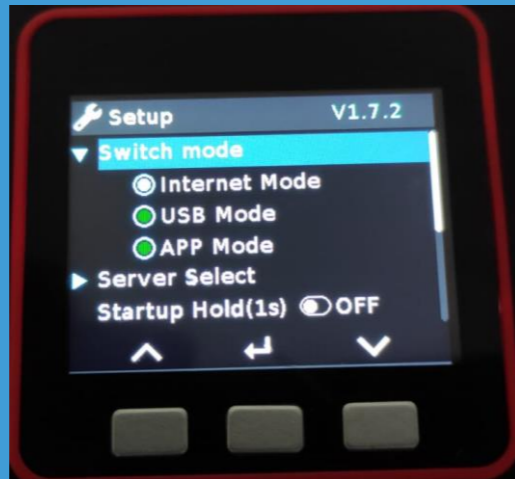
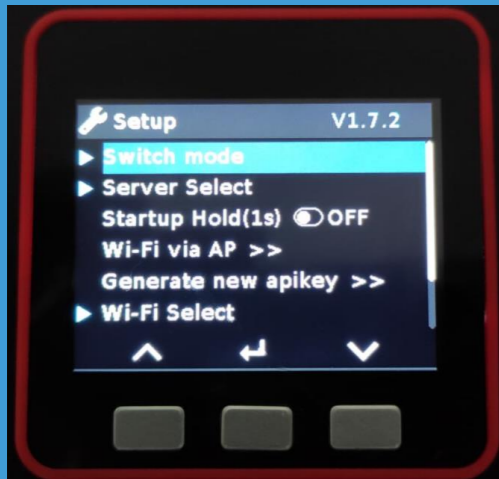
## Firmware menu

El firmware 1.7.2 Fire dispone de un menú para:

- Configurar la conectividad (Internet / USB)
- Seleccionar y ejecutar la app descargada en el dispositivo
- Otras opciones

**APP:** Listado de aplicaciones descargadas

**Internet Mode:** Se conecta a la Wi-Fi y mediante la url [flow.m5stack.com](http://flow.m5stack.com) y configurando la API KEY del dispositivo puedes abrir el IDE (UI FLOW) para programarlo.





# UI FLOW

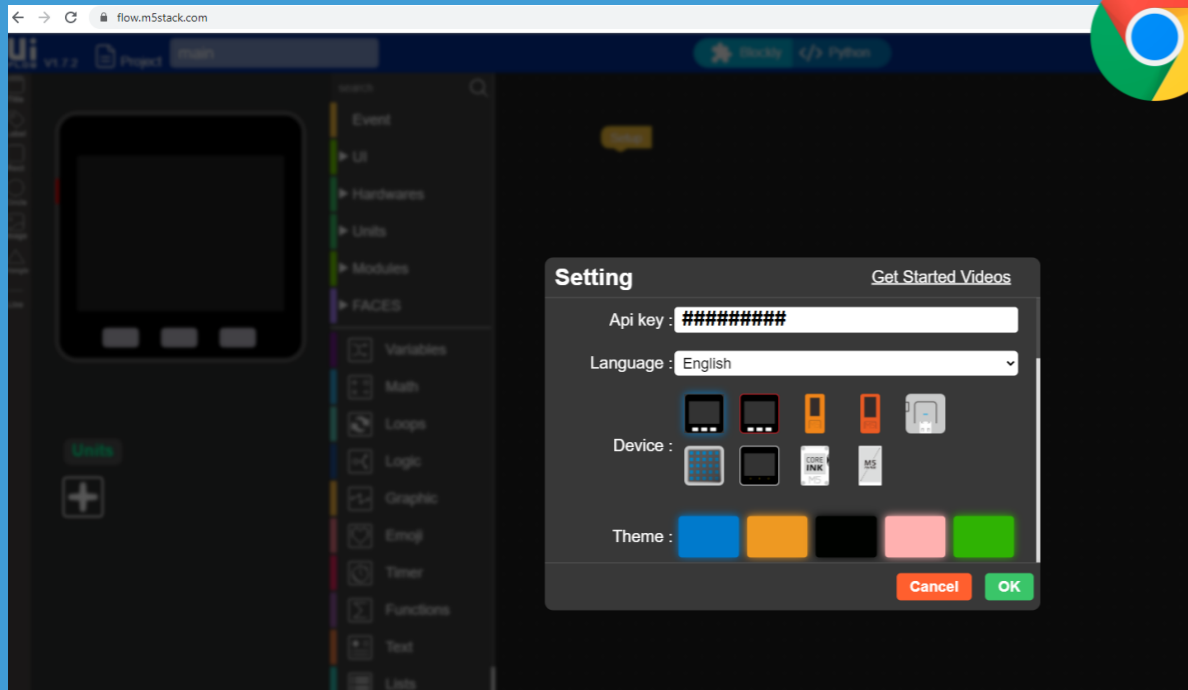


## UI FLOW online

## Blockly Graphic coding IDE Python

Desde cualquier navegador web puedes programar.  
URL: [flow.m5stack.com](http://flow.m5stack.com) y escribiendo la API KEY del dispositivo.

Muestra de la interfaz de usuario web del UI FLOW. En este caso hay un programa blockly.



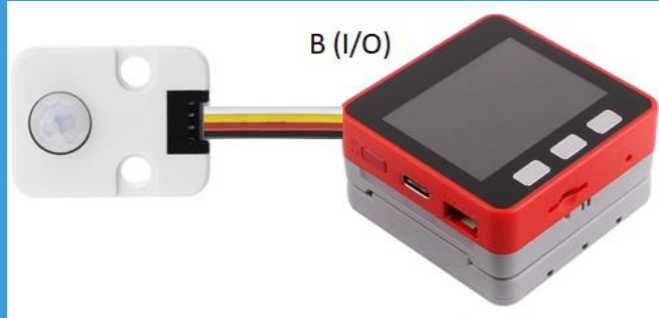


# PIR

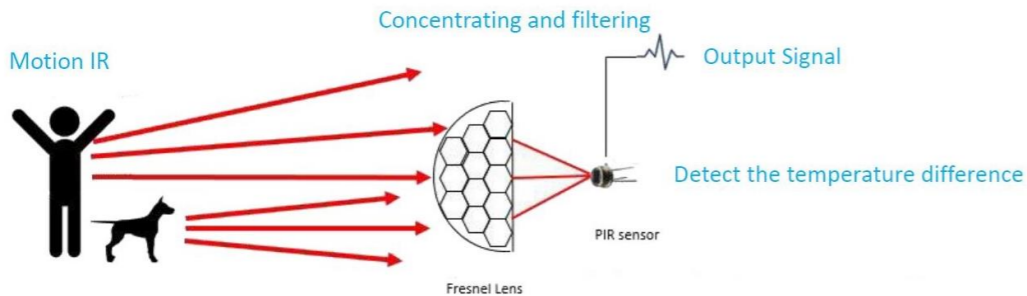


## MOTION SENSOR (PIR)

AS312 sensor PIR (Passive InfraRed)



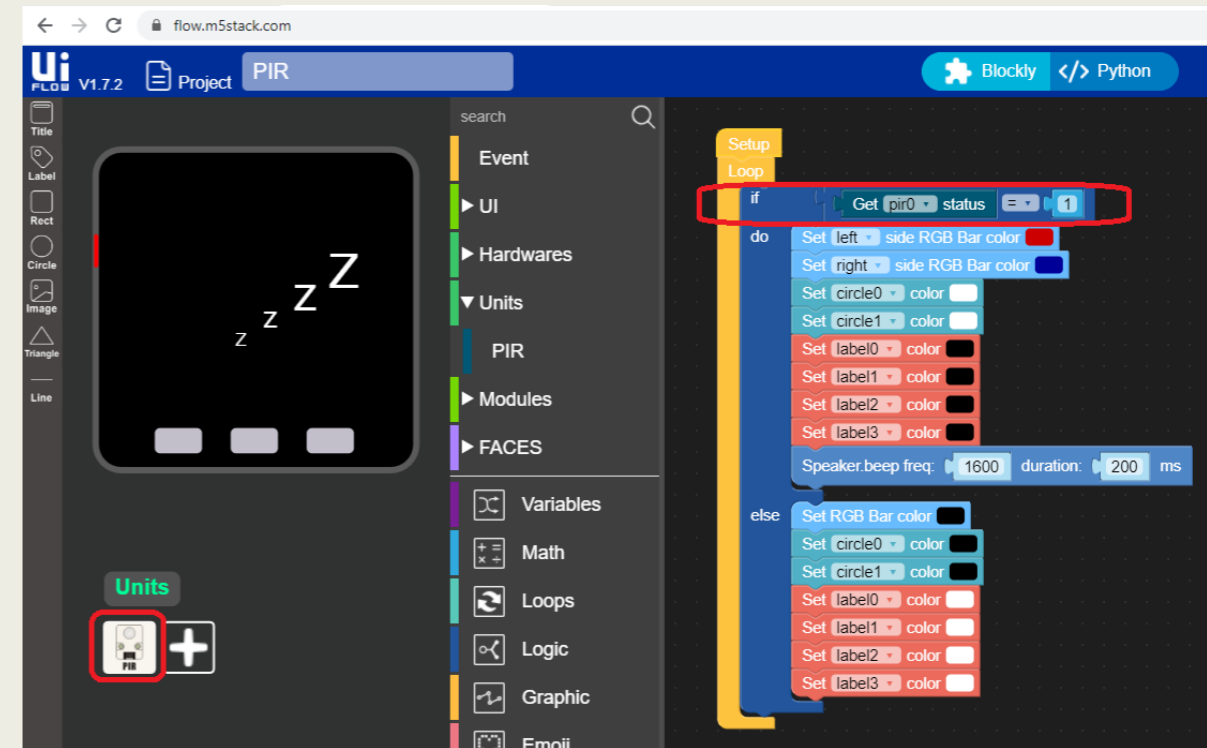
Motion IR



## Graphic coding IDE



Demo example: When the human body is detected, the RGB bar is lit and the speaker emits a sound







# ENV. II



## ENV. II SENSOR



## Graphic coding IDE

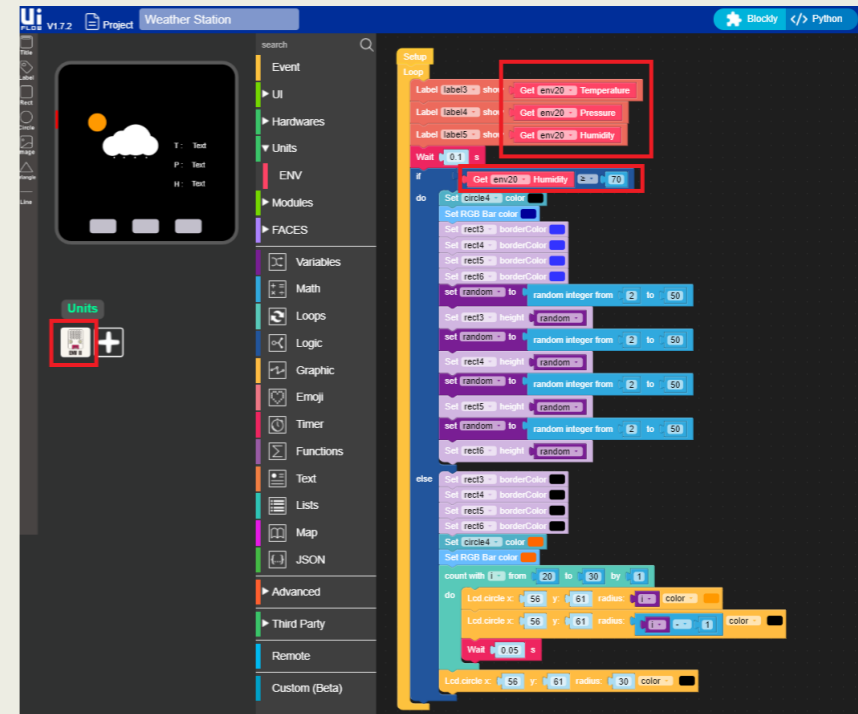


SHT30 sensor de temperatura (T) y humedad (HR)

BMP280 sensor de presión atmosférica (P)



Demo example: Display temperature, humidity, atmospheric pressure values, different animations based on humidity values





# RFID



## RFID

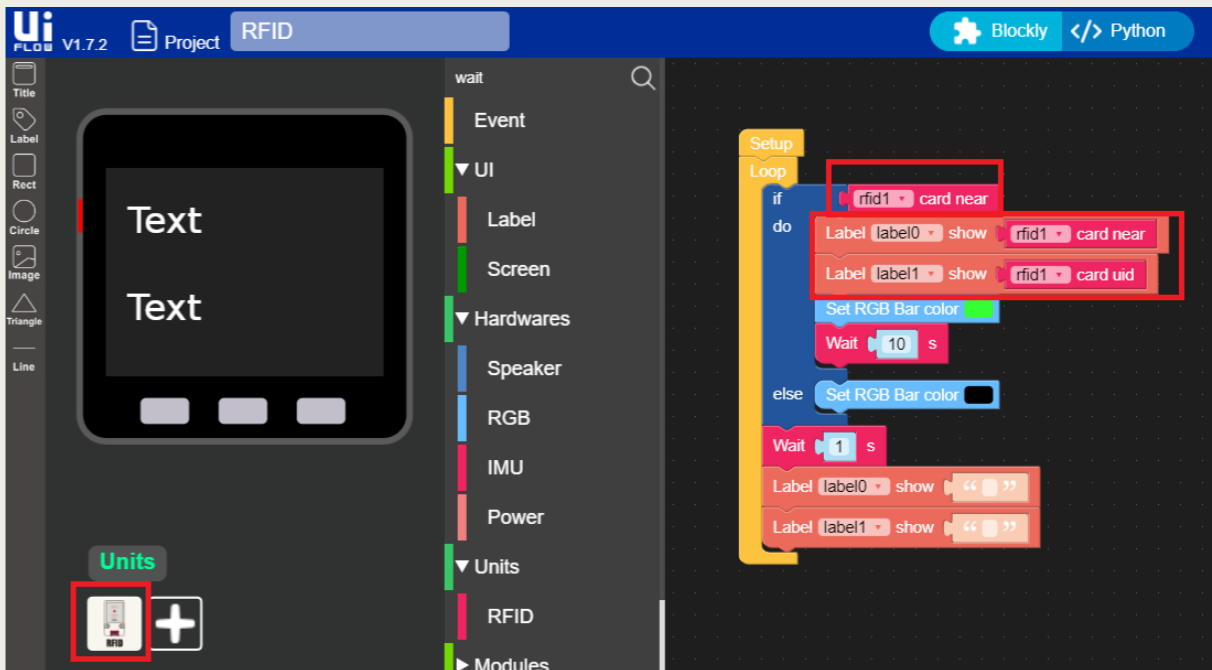


## Graphic coding IDE



El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio.

Sensor **RC522** (tecnología MIFARE) que trabaja a 13.56 MHz (High Frequency)



Demo example: Detect if a chip or card is near and write True and UID and light green RGB bar.



# RFID



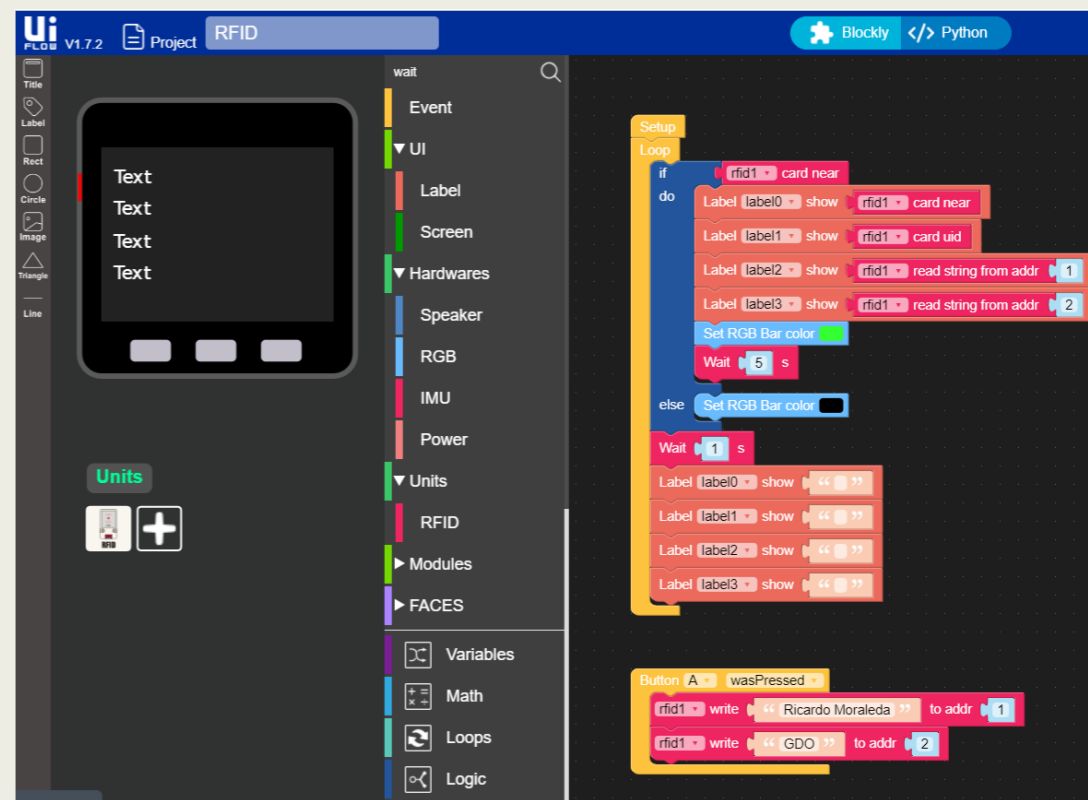
## RFID



## Graphic coding IDE



### Demo example: write and read



Los chips usados (pasivos) tienen 1K bytes de memoria (1.024 bytes)

El UID es imborrable y es único (4 bytes + fabricante).

<https://programarfacil.com/blog/arduino-blog/lector-rfid-rc522-con-arduino/>

En el ejemplo, cuando dejamos pulsado el botón A del M5STACK escribimos en las direcciones 1 y 2 (nombre y empresa). Luego al pasar el chip leemos en pantalla los datos almacenados.





# RFID



## RFID



## MicroPython

## RFID



## MicroPython

### El mismo ejemplo con MicroPython

```

Setup
Loop
  if rfid1 card near
  do
    Label label0 show rfid1 card near
    Label label1 show rfid1 card uid
    Label label2 show rfid1 read string from addr 1
    Label label3 show rfid1 read string from addr 2
    Set RGB Bar color
    Wait 6 s
  else
    Set RGB Bar color
  Wait 1 s
  Label label0 show
  Label label1 show
  Label label2 show
  Label label3 show
Button A wasPressed
  rfid1 write Ricardo Moraleda to addr 1
  rfid1 write GDO to addr 2
  
```

```

1 from m5stack import *
2 from m5sui import *
3 from uiflow import *
4 import time
5 import unit
6
7 setScreenColor(0x222222)
8 rfid1 = unit.get(unit.RFID, unit.PORTA)
9
10 label0 = M5TextBox(17, 30, "Text", lcd.FONT_DejaVu24, 0xFFFFFF, rotate=0)
11 label1 = M5TextBox(17, 74, "Text", lcd.FONT_DejaVu24, 0xFFFFFF, rotate=0)
12 label2 = M5TextBox(17, 119, "Text", lcd.FONT_DejaVu24, 0xFFFFFF, rotate=0)
13 label3 = M5TextBox(17, 162, "Text", lcd.FONT_DejaVu24, 0xFFFFFF, rotate=0)
14
15 def buttonA_wasPressed():
16     # global params
17     rfid1.writeBlock(1, 'Ricardo Moraleda')
18     rfid1.writeBlock(2, 'GDO')
19     pass
20 btnA.wasPressed(buttonA_wasPressed)
21
22
23 while True:
24     if rfid1.isCardOn():
25         label0.setText(str(rfid1.isCardOn()))
26         label1.setText(str(rfid1.readUid()))
27         label2.setText(str(rfid1.readBlockStr(1)))
28         label3.setText(str(rfid1.readBlockStr(2)))
29         rgb.setColorAll(0x33ff33)
30         wait(5)
31     else:
32         rgb.setColorAll(0x000000)
33     wait(1)
34     label0.setText('')
35     label1.setText('')
36     label2.setText('')
37     label3.setText('')
38     wait_ms(2)
  
```





# RFID



## RFID-NFC

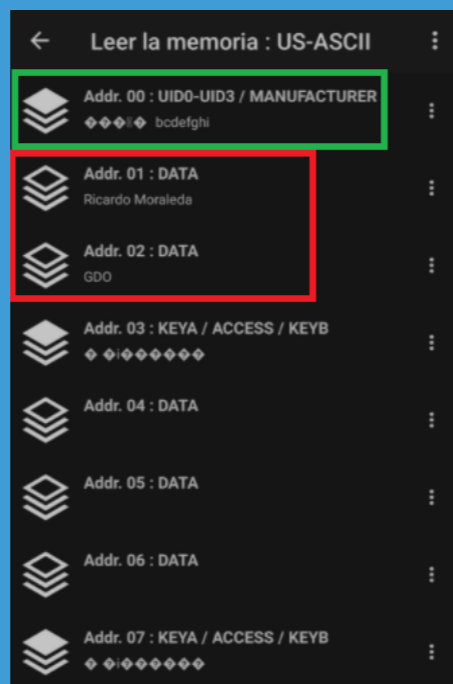
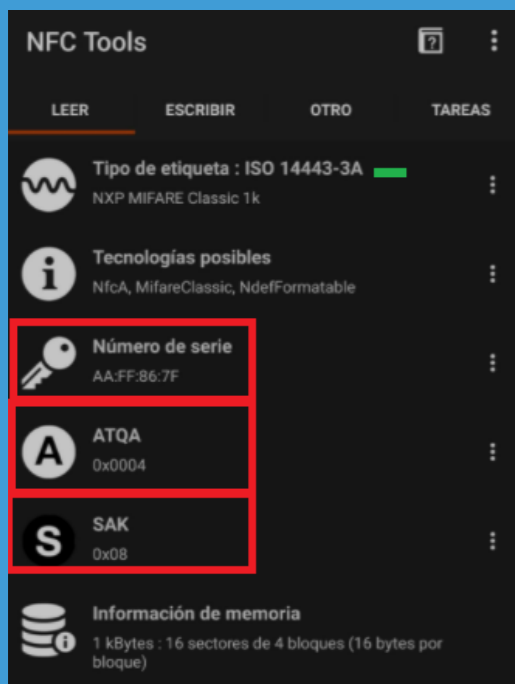


## Mapa memoria chip



Los chips usados (pasivos) tienen 1K de memoria (1.024 bytes)

Para ver el mapa de memoria uso mi Smartphone con NFC activado y en concreto la app en Android llamada **NFC Tools**. Acercando el chip al móvil me aparece esta información.



Tecnología MIFARE CLASSIC (ISO 1443-3A) de NXP Semiconductors.

1.024 bytes: 16 sectores de 4 bloques (16 bytes / bloque). Útiles de cada sector son 3 bloques y del sector 1 sólo 2, ya que el 1º es el UID-Manufacturer. En total disponemos de 47 bloques (752 bytes) de datos.

Addr. 00:

**AA:FF:86:7F:AC:08:04:00:62:63:64:65:66:67:68:69**

Número de serie UID (**AA:FF:86:7F**) – 4 bytes y 1 byte de control sobre el UID (**AC**). Después otros 10 bytes del fabricante, etc.: **08:04:00:62:63:64:65:66:67:68:69**

SAK ← ↓ → ATQA

<http://nfc-tools.org/index.php?title=ISO14443A>

<https://www.nxp.com/docs/en/application-note/AN10833.pdf>





# RFID



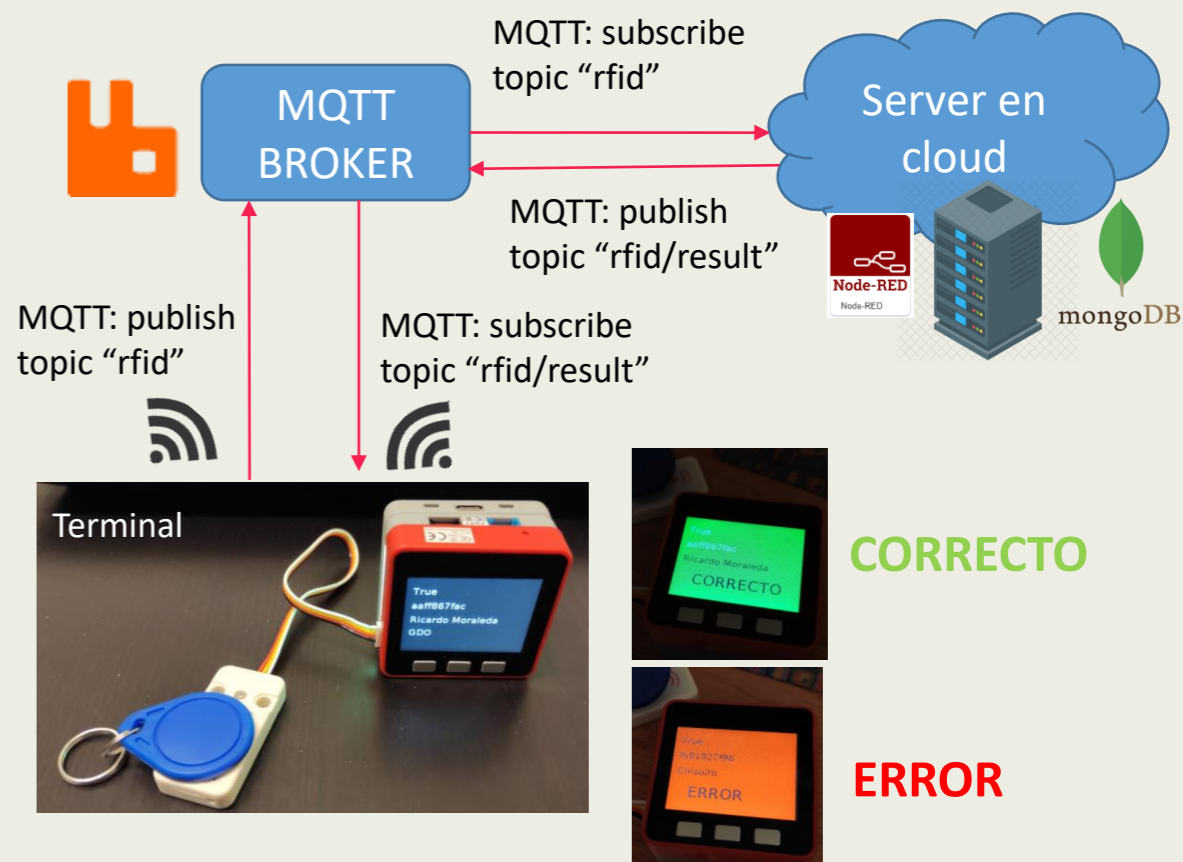
## #1. Control de acceso

Ejemplo completo con validación del usuario.

1. El M5 lee el chip RFID y envía el UID (o número de serie) por MQTT a un broker MQTT en Amazon WS (Rabbit MQ en cloud).
2. Hay un servicio (Node-RED + Mongo DB) en otro servidor de la nube que valida ese UID en una base de datos.
3. Si es correcto publica un mensaje = 1 y si no = 0.
4. El M5 recibe el resultado y lo muestra en la pantalla coloreando el fondo de verde si es CORRECTO o rojo si es ERROR.

Además se podría llegar a implementar que si es correcto abra una puerta, por ejemplo.

## Arquitectura





# RFID



## Terminal



## Servicio validador



### Código en Blockly del M5 STACK

```

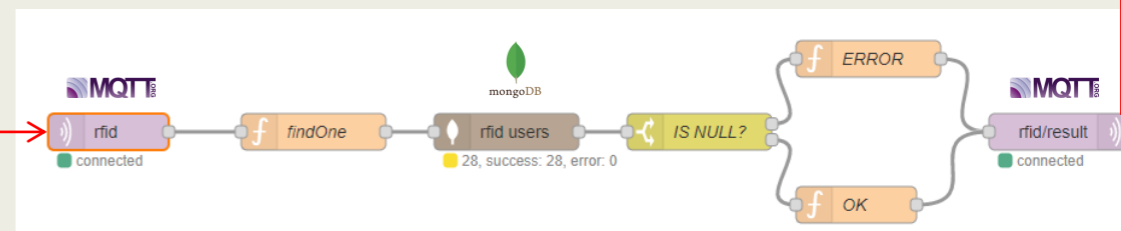
Setup
set client id "M5"
server "cloudamqp.com"
port 1883
user ""
password ""
keepalive 300
SSL False

mqtt start

Loop
if rfid1 card near
do
set codigo to rfid1 card uid
Set RGB Bar color
Label label0 show rfid1 card near
Label label1 show codigo
Label label2 show rfid1 read string from addr 1
publish topic rfid msg codigo
else
Set RGB Bar color
Wait 1 s
Label label0 show ""
Label label1 show ""
Label label2 show ""
Label label3 show ""

mqtt subscribe rfid/result with topic_data
Label label0 show ""
Label label3 show ""
set resultado to get topic_data
if resultado == "1"
do
Set RGB Bar color
Lcd fill
Label label3 show "CORRECTO"
Wait 10 s
Label label3 show ""
else
Set RGB Bar color
Lcd fill
Label label3 show "ERROR"
Wait 10 s
Label label3 show ""
Lcd fill

```



```

1 var tag = msg.payload;
2 var newMsg = {};
3 newMsg.operation = 'findOne';
4 newMsg.payload = { 'tag' : tag, 'active' : '1' };
5 newMsg.projection = { 'tag' : 1, '_id' : 0 };
6 return newMsg;

```



Service: External service

Server: rfid

Collection: users

Operation: Dynamic (msg.operation)

Name: Name

rfid users

Busca el UID (tag) enviado y si está activo (active = 1) Colección "rfid" y Documentos "users".

#	_id ObjectId	tag String	name String	active String
1	6065e85f46b3b6eb3d5b593e	"aaff867fac"	"Ricardo"	"1"



# METEO



## #2. Visor METEO

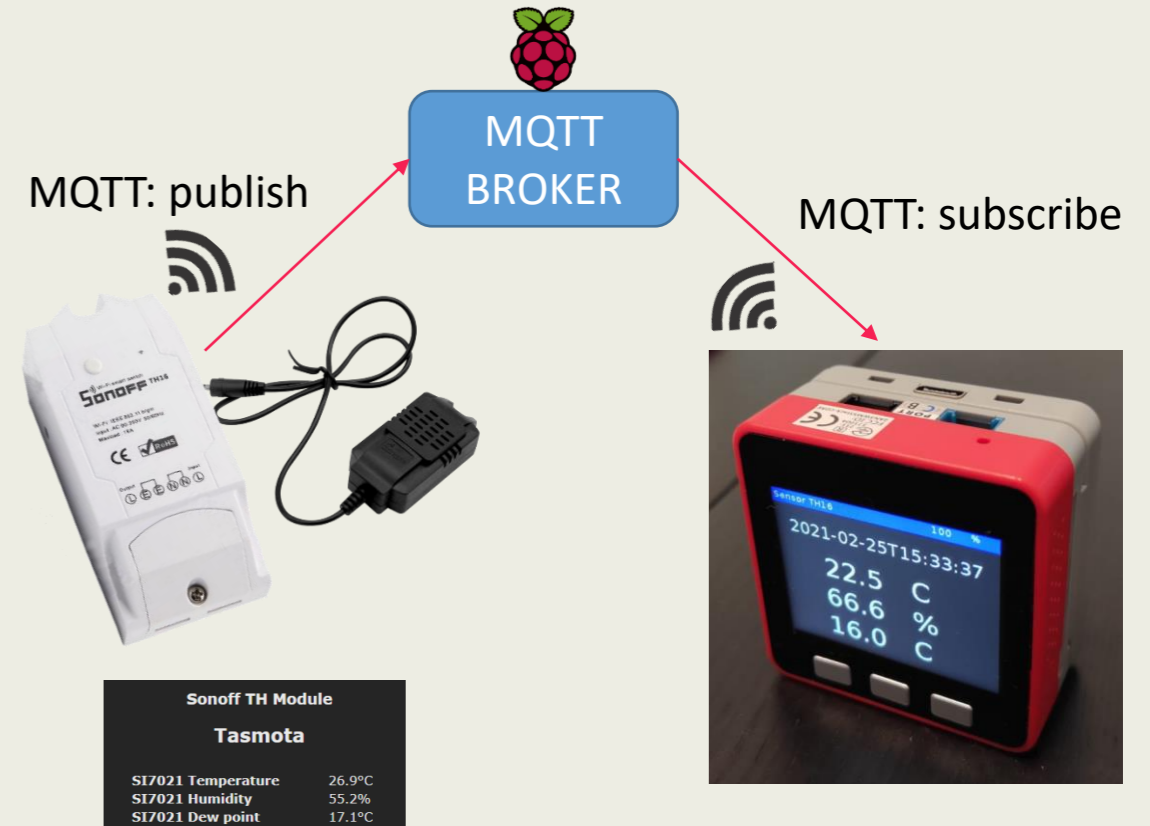
## Arquitectura

En el Capítulo 8 – Home Automation IoT enviábamos los datos de temperatura, humedad y punto de rocío de un sensor SI7021a través de la placa TH16 de Sonoff por MQTT (vía WI-FI) a una Raspberry Pi4 que hacía de MQTT Broker.

En lugar de visualizar datos con un dashboard (navegador web) haremos que M5STACK se suscriba y muestre los datos por pantalla como terminal.

También cuando se actualizan los datos se encienden las dos barras de led en verde.

Además mostramos el nivel de batería del M5STACK. Los posibles valores son 0-25-50-75-100%.





# METEO



## MQTT

## Datos MQTT

### Suscripción al broker MQTT y representación datos

```

Setup
set temperaturaint to ""
set data to ""
set SI7021 to ""
set datajson to dumps "" to json
set date to ""
set client id to "m5"
server to "192.168.1.129"
port to 1883
user to ""
password to ""
keepalive to 300
SSL False
mqtt start
Title title0 show "Sensor TH16"
Set title0 backgroundColor
Label label0 show "C"
Label label1 show ""
Label label4 show "%"
Label label5 show "C"
Label fechahora show ""
Label temp show ""
Label hum show ""
Label rocio show ""
Label label2 show ""

mqtt subscribe "tele/tasmota/SENSOR" with topic_data
Set left side RGB Bar color
Set right side RGB Bar color
set data to get topic_data
set datajson to loads json data
set date to get key "Time" in map datajson
set SI7021 to get key "SI7021" in map datajson
set temperaturaint to get key "Temperature" in map SI7021
set humedadint to get key "Humidity" in map SI7021
set rocioint to get key "DewPoint" in map SI7021
Label fechahora show ""
Label rocio show ""
Label hum show ""
Label temp show ""
Label fechahora show date
Label temp show temperaturaint
Label hum show humedadint
Label rocio show rocioint
Wait 2 s
Set left side RGB Bar color
Set right side RGB Bar color
Label label1 show get Battery Level

```

- Conexión con el broker
- Suscripción al topic: **tele/tasmota/sensor**
- Convierto los datos recibidos en JSON
- Obtengo las claves:
  - Time
  - SI7021.Temperature
  - SI7021.Humidity
  - SI7021.DewPoint
- Nuevos datos RGB Bar color (Green)
- get Battery Level (del M5STACK)

```

tele/tasmota/SENSOR : msg.payload : Object
  object
    Time: "2021-03-21T18:03:58"
  SI7021: object
    Temperature: 22.9
    Humidity: 50.2
    DewPoint: 12
    TempUnit: "C"

```





# REMOTE



## #3. Remote example

Control remoto del hardware del propio M5STACK.

Lectura código QR con el móvil o escribiendo la URL generada en el UI flow en un navegador web en PC.

Control de las dos barras de led ON/OFF

Brillo de las dos barras de led (0-100)

El QR genera una URL:

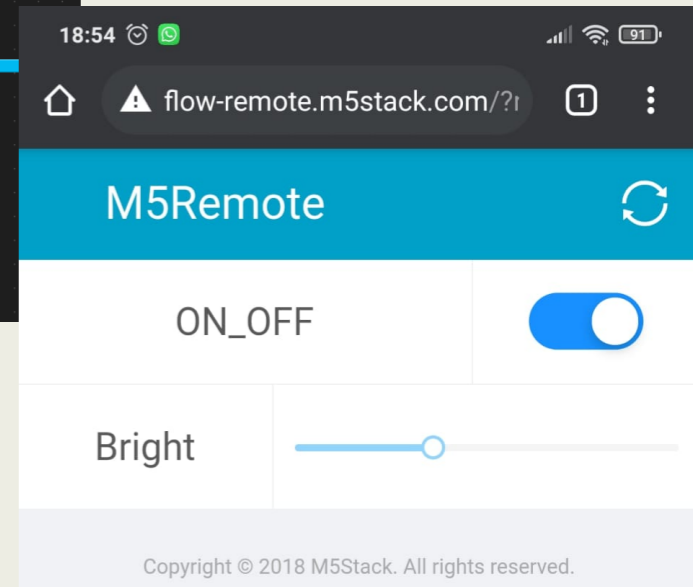
<http://flow-remote.m5stack.com/?remote=<id>>



## Remote example

```

Setup
  Remote qrcode show in x 72 y 32 size 176
  Add Remote Switch Button ON_OFF with: x index 0
    if
      x = 1
    do
      Set RGB Bar color
    else
      Set RGB Bar color
  Add Remote Slider Bright with: x index 0
    Set RGB brightness x
  
```

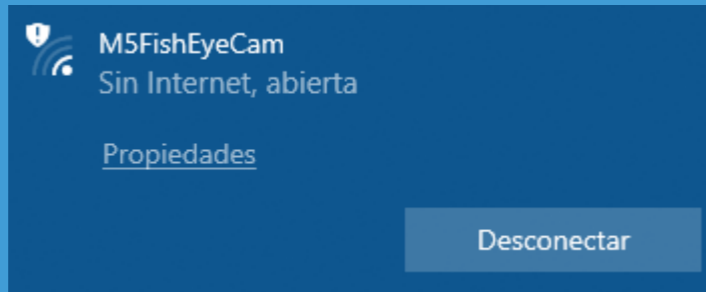




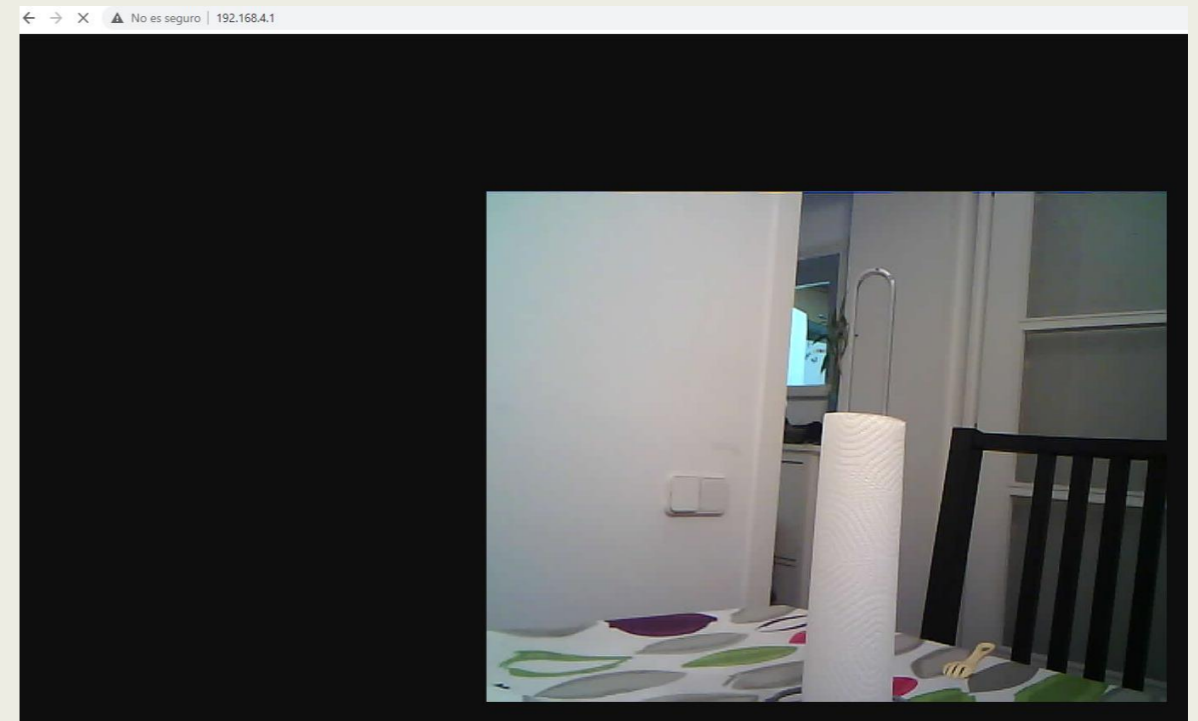


## #4. M5CAMERA

Conexión Wi-Fi a la red "M5FishEyeCam" de la propia cámara y en un navegador poner 192.168.4.1



## M5CAMERA



Óptica: OV2640 de 2 M pixels



# FIRMWARE



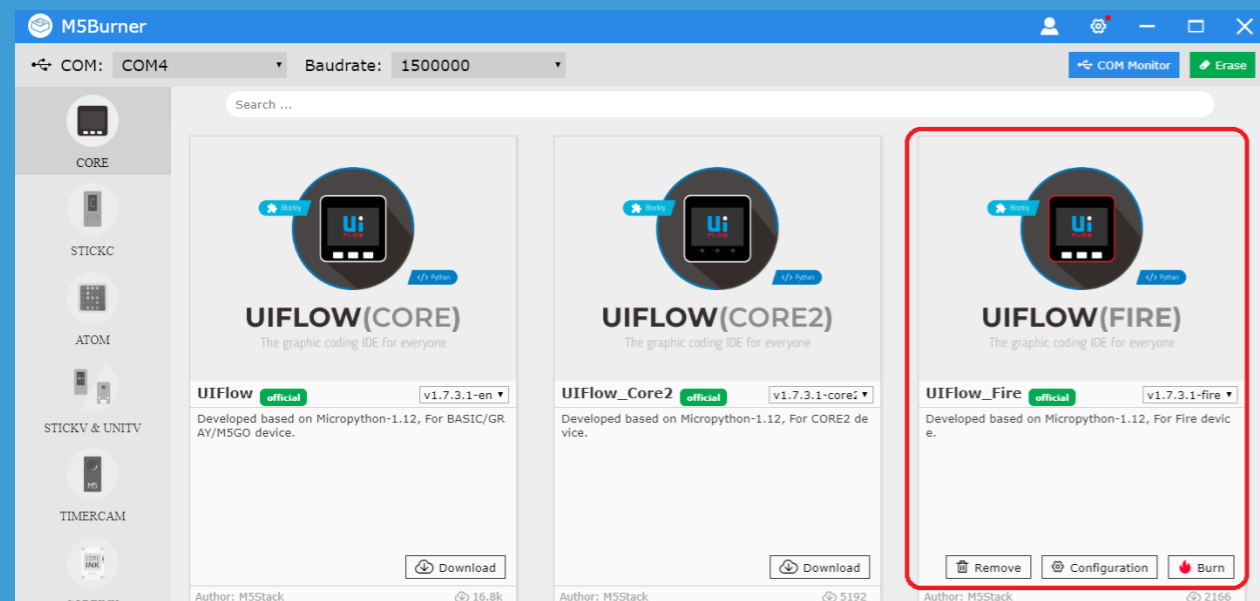
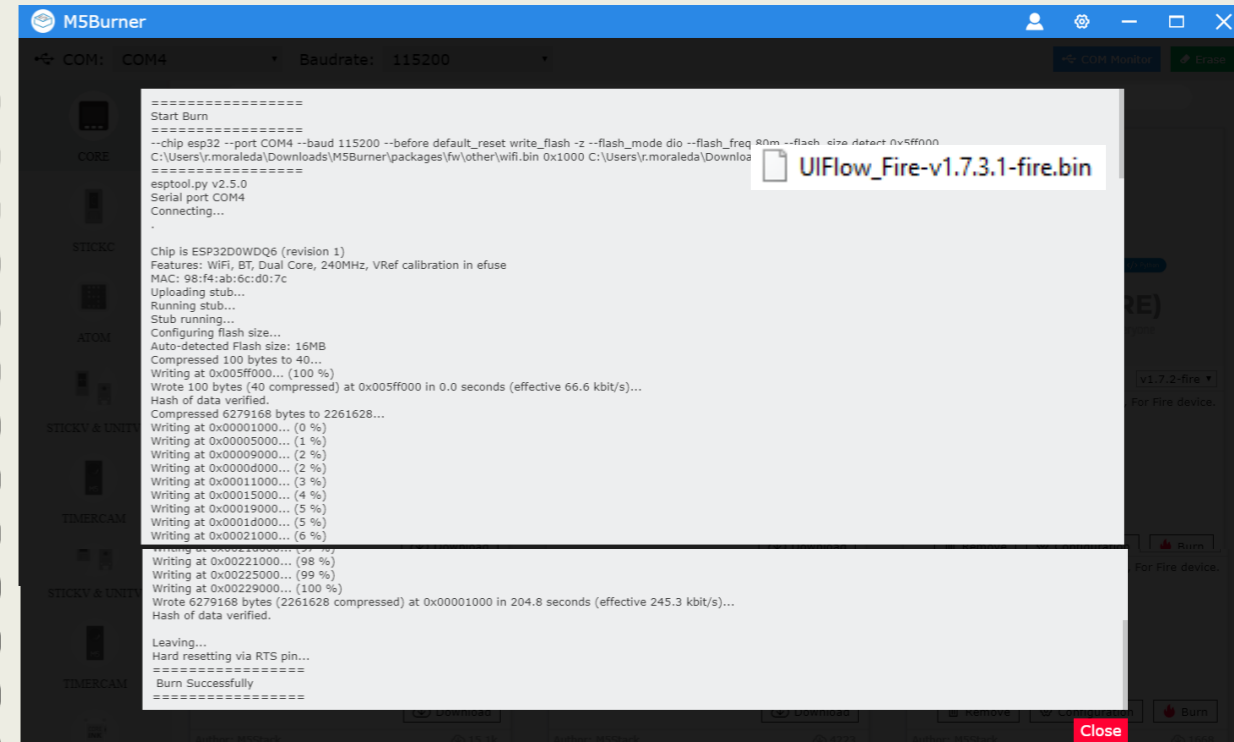
## Upgrade firmware UIFLOW

## Download + Config + Burn

Enlace:  
[https://docs.m5stack.com/#/en/quick\\_start/m5stickc/m5stickc\\_quick\\_start\\_with\\_uiflow](https://docs.m5stack.com/#/en/quick_start/m5stickc/m5stickc_quick_start_with_uiflow)

Download firmware, configure (WiFi/mode) and burn.

Con el software M5Burner y el driver USB CP210x previamente instalado. Elegir el puerto COM y modelo de placa. En mi caso FIRE y la versión v1.7.3.1 (latest)



# M5STACK

v.1.1 MARZO 2024



<https://www.linkedin.com/in/ricardo-moraleda-gareta-9421099>

<https://www.linkedin.com/company/gdo-electric1996/>

RICARDO MORALEDA GARETA