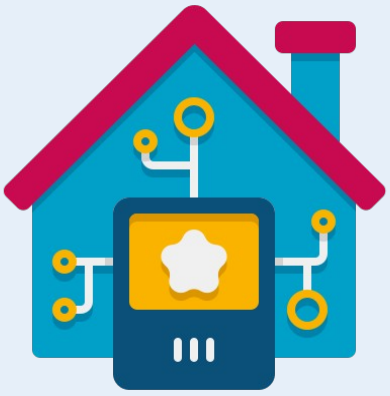
CFR de Vigo

Integración da IoT e da Domótica

Prácticas Node-Red MQTT Versión Arduinoblocks



Flaticon

Versión de Arduinoblocks para as prácticas NODE-RED - MQTT

As prácticas poden realizarse en Arduinoblocks, con algún que outra modificación menor. O maior problema atopado, que poida que teña solución, aínda que eu non a atopei, é que non se poden asignar variables booleanas a un topic, co que teríamos que cambiar a forma de configurar os nodos e as variables asociadas a sensores e actuadores binarios.

1. Práctica 1. Conexión do ESP32 á rede WIFI e ao servidor MQTT

Nesta primeira práctica comprobaremos que o ESP32 se conecta correctamente á wifi e ao servidor MQTT.

Pasos a seguir:

1. Creamos o programa cos bloques de Arduinoblocks.
2. Descargamos o código e o abrimos co IDE de Arduino.
3. Cargamos o programa no ESP32, tal como vimos na práctica 1 anterior.
4. No exemplo da imaxe tamén facemos que o ESP teña unha IP fixa, isto non o faremos neste curso, pero si que é adecuado facelo na aula co alumnado, pero sempre asignando unha IP distinta a cada ESP.
5. MOI IMPORTANTE: se queremos conectar máis dunha ESP ao mesmo servidor teremos que cambiar o nome do cliente MQTT, no caso do código obtido de Arduinoblocks:

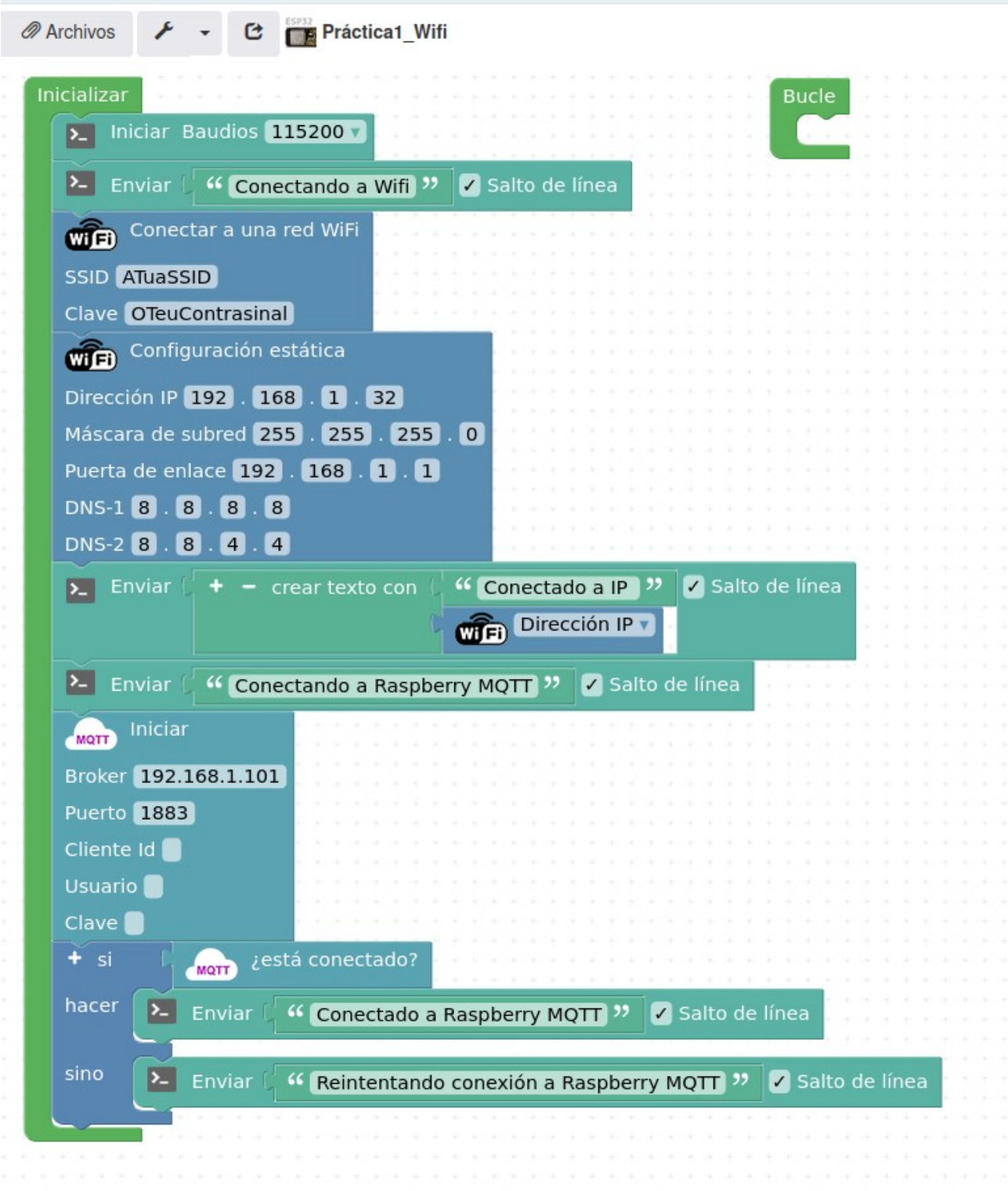
   PubSubClient espmqtt_client(espmqtt_wifiClient);

   PubSubClient espmqtt_client(espmqtt_wifiClient2);

   .

   PubSubClient espmqtt_client(espmqtt_wifiClient10);

6. Abrimos o monitor serie e comprobamos que o ESP se conecta correctamente. Para unha comunicación correcta temos que seleccionar un"Baud rate" de 115200



Temos que substituír a IP do broker MQTT pola IP do noso broker, a SSID e contrasinal da nosa WiFi e se non queremos unha IP estática para cada ESP teremos que eliminar o bloque “Configuración estática”

2. Práctica 2. Control dun LED

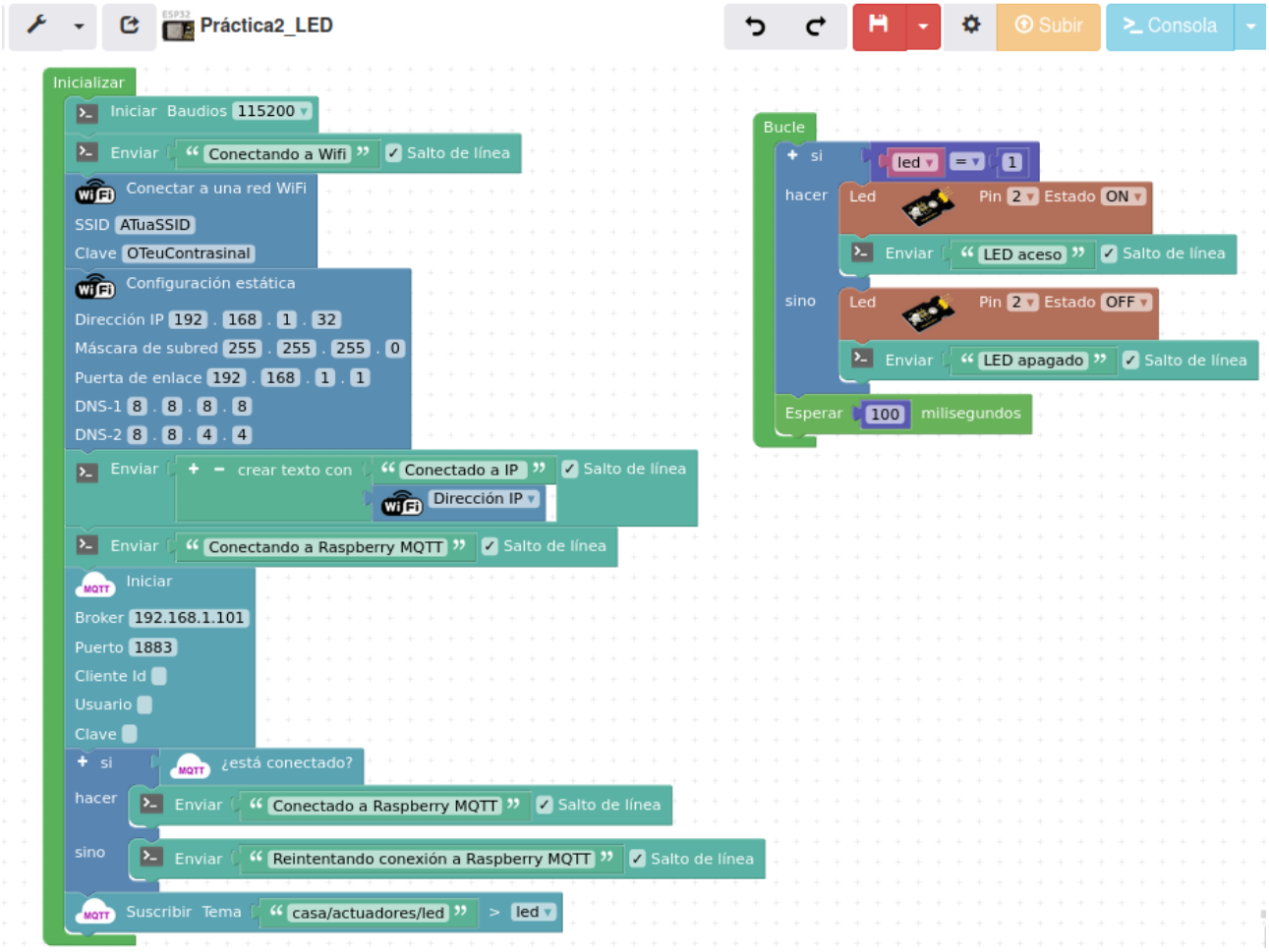
O Ola Mundo! Canónico, acender e apagar un LED.

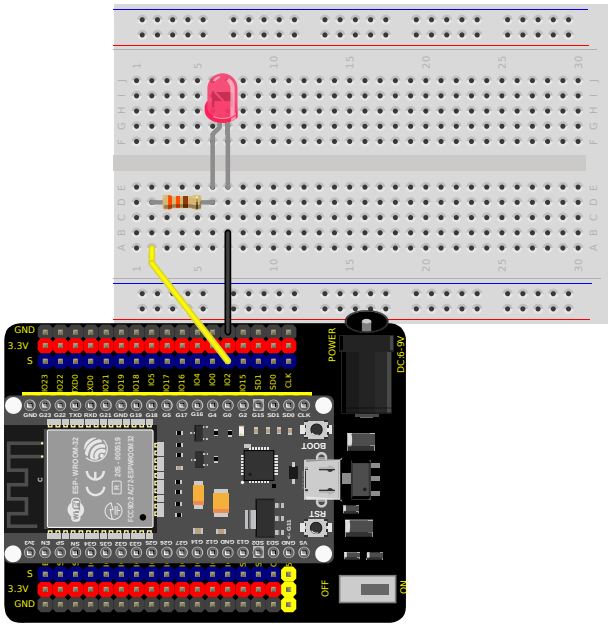
Material necesario:

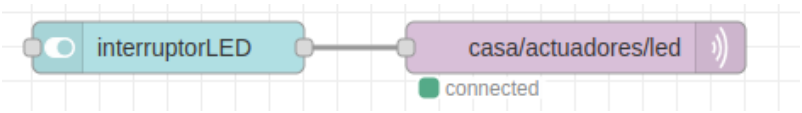
- ESP32 WROOM + shield
- LED 5mm
- Resistencia de 330 Ω
- Placa de conexións
- Cables M-F

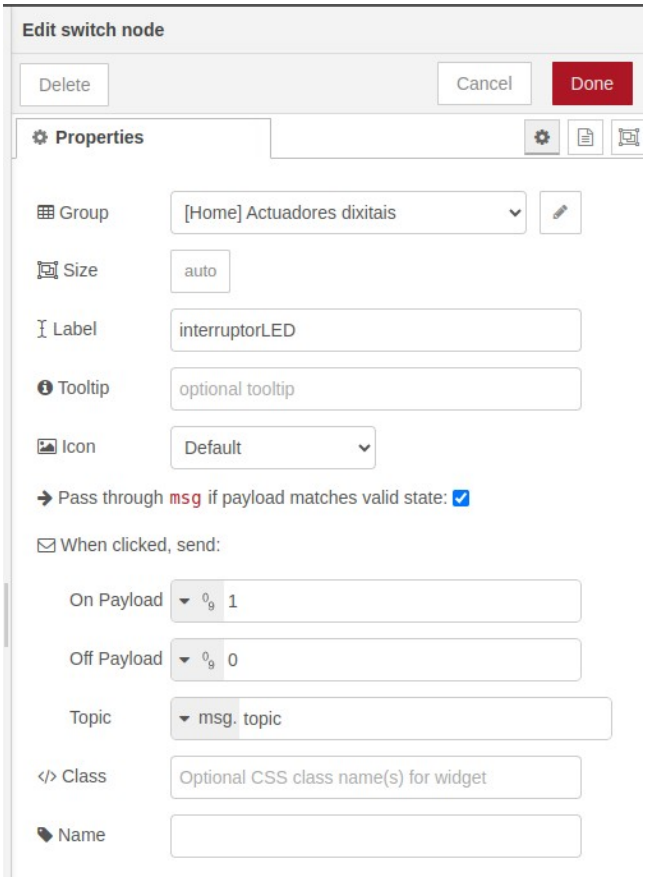
Pasos a seguir:

1. Creamos o programa da imaxe inferior en Arduinoblocks e seguimos os pasos da práctica do boletín anterior:
2. Para poder controlar un LED desde o panel de control temos que crear o fluxo en Node-Red. Como temos que telo preparado para o resto de prácticas xa facemos o fluxo completo.
3. Todos os topics teremos que utilizalos nos programas, polo que é importante estruturalos correctamente e non equivocarse ao escribilos nos programas.
4. No documento Mapa_Topics podes ver a proposta de topics.
5. Neste caso creamos o topic casa/actuadores/led e utilizamos os nodos **switch** e **mqtt out**, que chamaremos **interruptorLED** e **casa/actuadores/led** para que no fluxo teñamos claro o que fan eses nodos.
6. Descargamos o código desde Arduinoblocks e o abrimos co IDE Arduino, modificamos os nomes dos clientes MQTT, no caso de conectar máis dun ESP en cada broker e subimos o programa ao ESP.
7. Comprobamos que a configuración dos nodos do fluxo de Node Red se corresponde coa que aparece no código do programa. Que estamos a usar o mesmo topic e as mesmas mensaxes.
8. O esquema de conexións, a configuración do nodo **switch** e o fluxo de Node-Red podedes velos máis abaixo. LED conectado ao GPIO 2.
9. Abrimos no navegador o panel (Dashboard): ip do servidor:1880/ui
10. Comprobamos que ao accionar o interruptor o LED acende e apaga.


Práctica2_LED
Subir
Consola
Inicializar
Iniciar Baudios 115200
Enviar "Conectando a Wifi" Salto de línea
Conectar a una red WiFi
SSID ATuaSSID
Clave OTeuContrasinal
Configuración estática
Dirección IP 192 . 168 . 1 . 32
Máscara de subred 255 . 255 . 255 . 0
Puerta de enlace 192 . 168 . 1 . 1
DNS-1 8 . 8 . 8 . 8
DNS-2 8 . 8 . 4 . 4
Enviar crear texto con "Conectado a IP" Salto de línea
Dirección IP
Enviar "Conectando a Raspberry MQTT" Salto de línea
Iniciar
Broker 192.168.1.101
Puerto 1883
Cliente Id
Usuario
Clave
si ¿está conectado?
hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
sino Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
Suscribir Tema "casa/actuadores/led" > led
Bucle
si led = 1
hacer Led Pin 2 Estado ON
Enviar "LED aceso" Salto de línea
sino Led Pin 2 Estado OFF
Enviar "LED apagado" Salto de línea
Esperar 100 milisegundos




interruptorLED
casa/actuadores/led
connected


Edit switch node
Delete
Cancel
Done
Properties
Group
[Home] Actuadores dixitais
Size
auto
Label
interruptorLED
Tooltip
optional tooltip
Icon
Default
Pass through msg if payload matches valid state:
When clicked, send:
On Payload
1
Off Payload
0
Topic
msg. topic
Class
Optional CSS class name(s) for widget
Name

3. Práctica 3. Relé

O relé funcionará, a efectos prácticos como o LED, será un interruptor que activaremos e desactivaremos para que este á súa vez acenda ou apague un actuador que necesita máis potencia que a que pode proporcionar o ESP32. No noso exemplo utilizaremos un ventilador que funciona con 5 V.

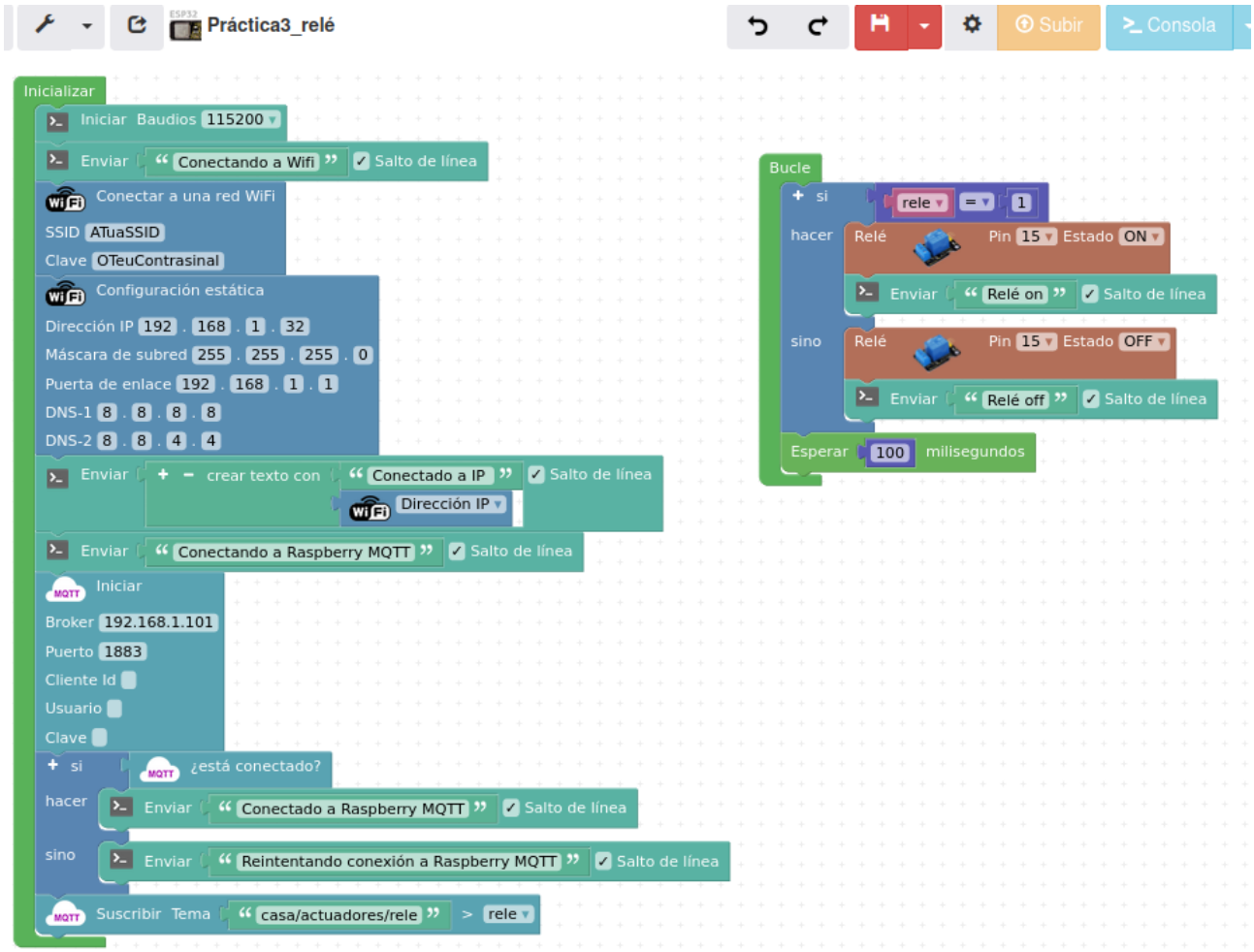
Material necesario:

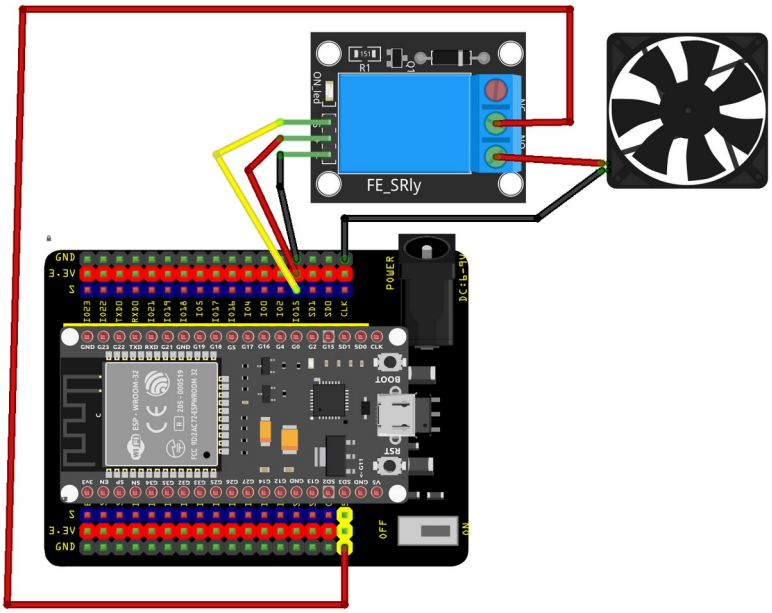
- ESP32 WROOM + shield
- Módulo relé.
- Ventilador 5V.
- Cables M-M, M-F, F-F

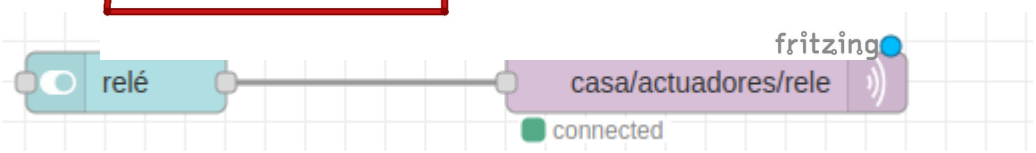
Pasos a seguir:

1. Seguimos os mesmos pasos que na práctica do LED, pero co programa de Arduinoblocks da imaxe inferior. Ao nodo switch que creamos en Node-Red chamarémolo **relé** e ao módulo mqtt out chamámolo **casa/actuadores/rele**
2. O esquema de conexións está máis abaixo, o relé está conectado ao GPIO 15. O relé conectámolo a un dos pins do shield que proporcionan 5V.
3. Unha vez teñamos creados os nodos en Node-Red, conectados os compoñentes na placa e cargado o programa no ESP32 abrimos o panel de control e comprobamos que o relé funciona correctamente, facendo que o ventilador acenda e apague.

   Non esquecer modificar os nomes dos clientes MQTT, se fora necesario, antes de cargar o programa no ESP.


Práctica3_relé
Subir
Consola
Inicializar
Iniciar Baudios 115200
Enviar "Conectando a Wifi" Salto de línea
Conectar a una red WiFi
SSID ATuaSSID
Clave OTeuContrasinal
Configuración estática
Dirección IP 192 . 168 . 1 . 32
Máscara de subred 255 . 255 . 255 . 0
Puerta de enlace 192 . 168 . 1 . 1
DNS-1 8 . 8 . 8 . 8
DNS-2 8 . 8 . 4 . 4
Enviar crear texto con "Conectado a IP" Salto de línea
Dirección IP
Enviar "Conectando a Raspberry MQTT" Salto de línea
Iniciar
Broker 192.168.1.101
Puerto 1883
Cliente Id
Usuario
Clave
si ¿está conectado?
hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
sino Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
Suscribir Tema "casa/actuadores/rele" > rele
Bucle
si rele = 1
hacer Relé Pin 15 Estado ON
Enviar "Relé on" Salto de línea
sino Relé Pin 15 Estado OFF
Enviar "Relé off" Salto de línea
Esperar 100 milisegundos


FE_SRly


relé
casa/actuadores/rele
fritzing
connected

4. Práctica 4. Zoador.

O zoador será activo e o activamos e desactivamos igual que ao LED e o relé. A patilla + do zoador conectámola ao GPIO12, a través dunha resistencia de 220Ω e a outra ao GND.

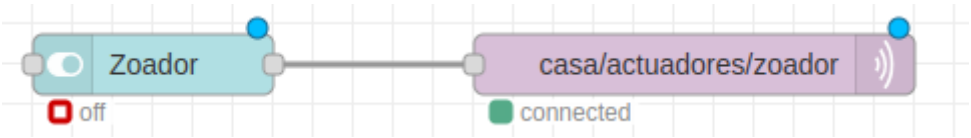
Material necesario:

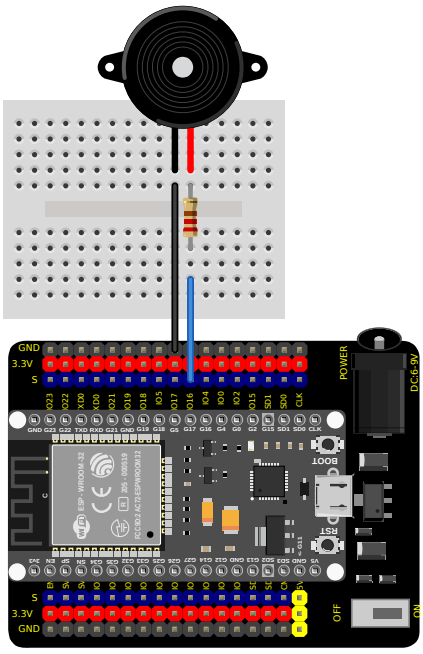
- ESP32 WROOM + shield
- Zoador.
- Cables F-F

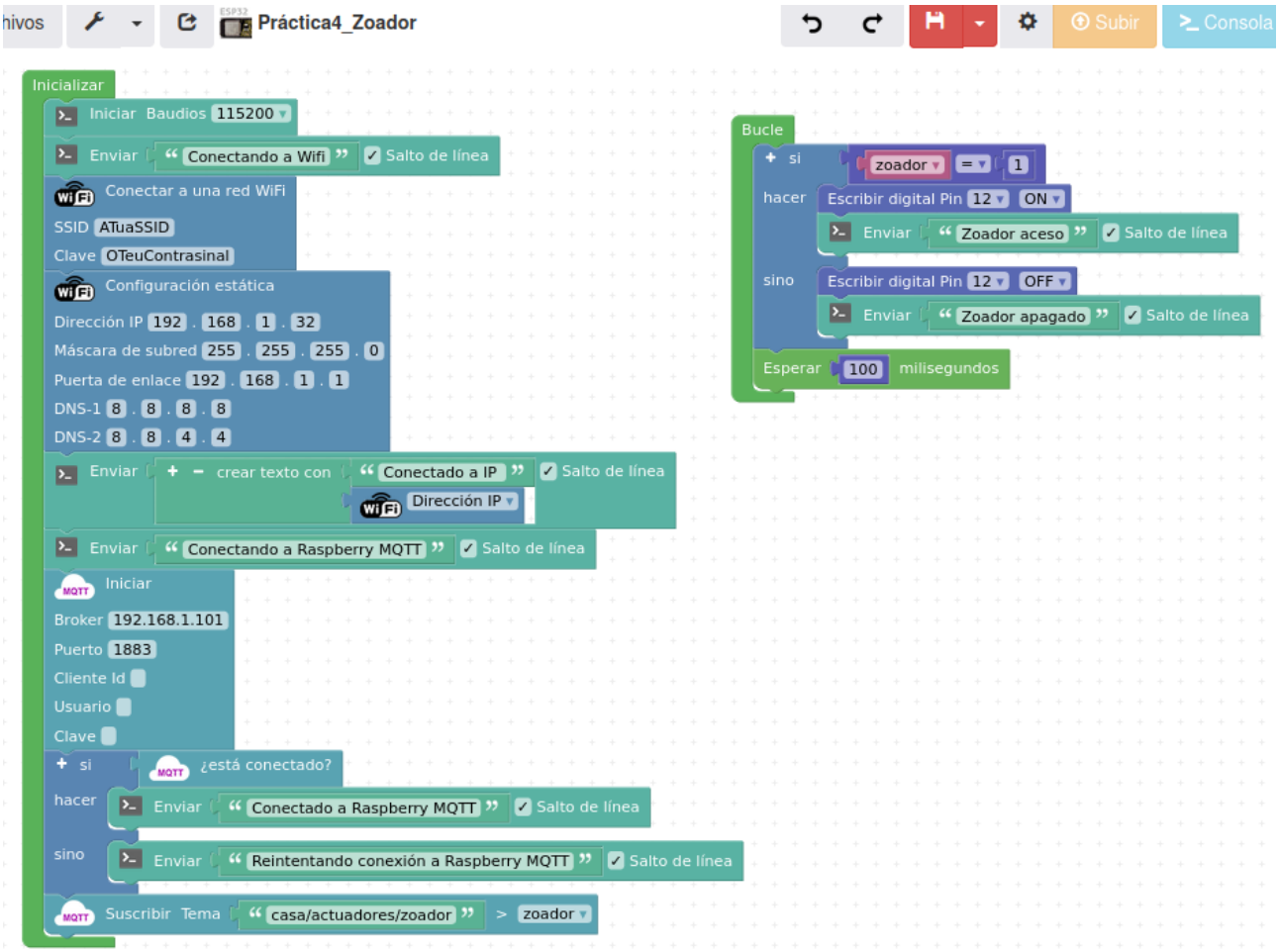
Pasos a seguir:

1. Seguimos os mesmos pasos que na práctica do LED, pero co programa da imaxe. Ao nodo switch que creamos en Node-Red chamarémolo **Zoador** e ao módulo mqtt out chamámolo **casa/actuadores/zoador**
2. O esquema de conexións e o fluxo de Node-Red están máis abaixo.
3. Unha vez teñamos creado os nodos en Node-Red, conectados os compoñentes na placa e cargado o programa no ESP32 abrimos o panel de control e comprobamos que o zoador funciona correctamente, facendo ruído ao activalo.

   Non esquecer modificar os nomes dos clientes MQTT, se fora necesario, antes de cargar o programa no ESP.






Práctica4_Zoador
Subir
Consola
Inicializar
Iniciar Baudios 115200
Enviar "Conectando a Wifi" Salto de línea
Conectar a una red WiFi
SSID ATuaSSID
Clave OTeuContrasinal
Configuración estática
Dirección IP 192 . 168 . 1 . 32
Máscara de subred 255 . 255 . 255 . 0
Puerta de enlace 192 . 168 . 1 . 1
DNS-1 8 . 8 . 8 . 8
DNS-2 8 . 8 . 4 . 4
Enviar crear texto con "Conectado a IP" Salto de línea
Dirección IP
Enviar "Conectando a Raspberry MQTT" Salto de línea
Iniciar
Broker 192.168.1.101
Puerto 1883
Cliente Id
Usuario
Clave
si ¿está conectado?
hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
sino Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
Suscribir Tema "casa/actuadores/zoador" > zoador
Bucle
si zoador = 1
hacer Escribir digital Pin 12 ON
Enviar "Zoador aceso" Salto de línea
sino Escribir digital Pin 12 OFF
Enviar "Zoador apagado" Salto de línea
Esperar 100 milisegundos

5. Práctica 5. Microservo

O control dun servo será moi útil para poder mover múltiples mecanismos. No exemplo da casa utilizamos unha porta accionada por un mecanismo de piñón – cremalleira.

Na placa que utilizamos podemos conectar a alimentación do servo a 3,3V ou a 5V, nos dous casos funcionará, pero xa que o utilizamos para mover unha porta conectarémolo ao pin de 5V, e ao GPIO5.

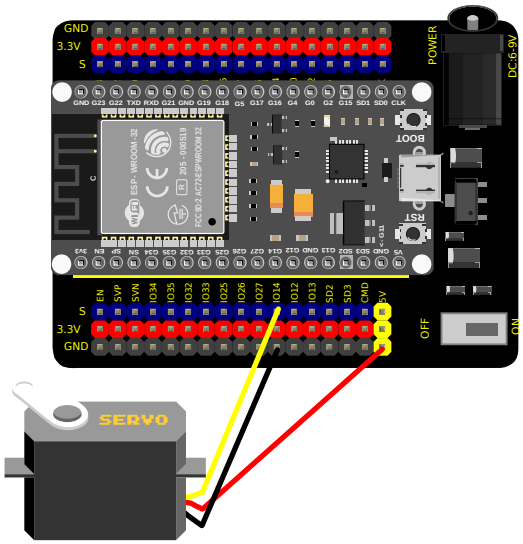
Para manexar o servo coa biblioteca ESP32Servo enviamos un valor ao servo de entre 0 e 100, que se traduce nun movemento do servo de entre 0 e 180º.

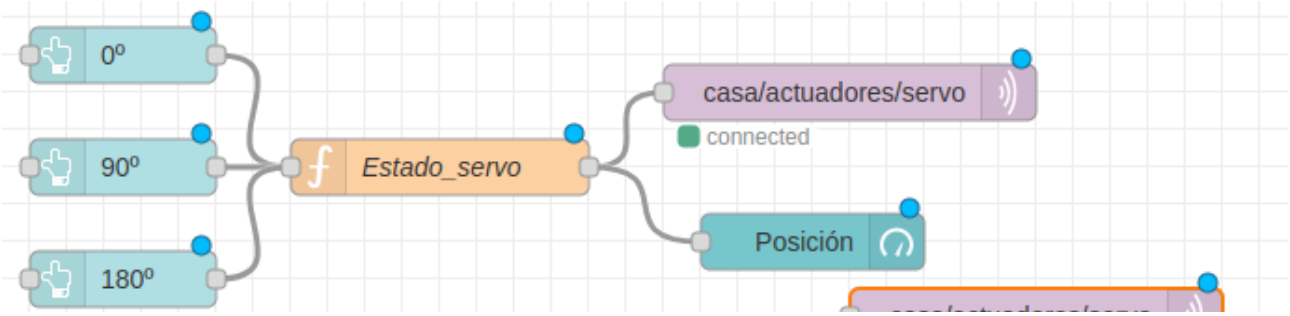
En Node-Red podemos crear botóns para posicións concretas.

- ESP32 WROOM + shield Microservo 9g Cable M-F

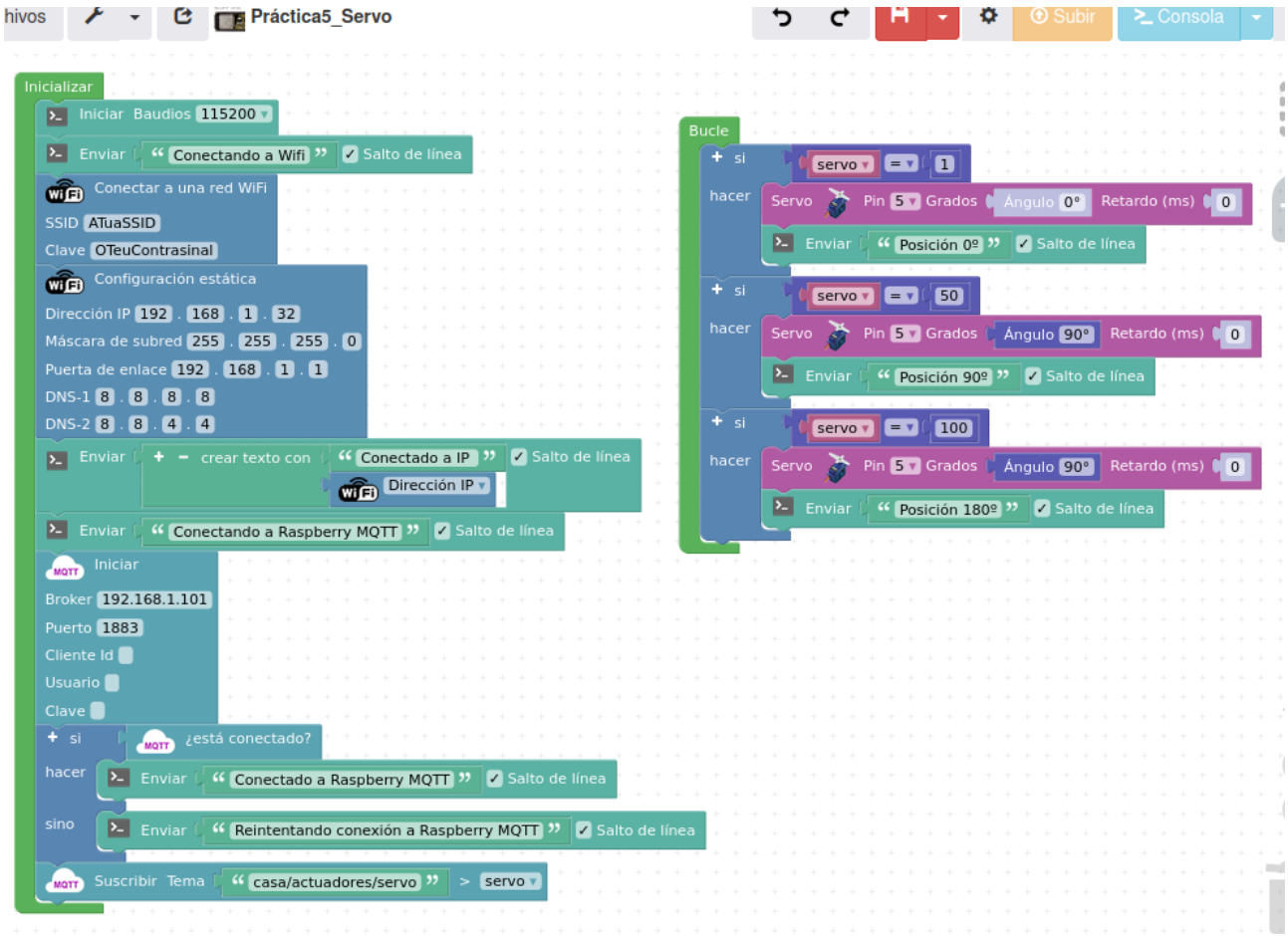
Pasos a seguir:

1. O esquema de conexións e os nodos en Node-Red son os das imaxes.
2. Creamos 3 botóns para as posicións 0, 90 e 180º.
3. O módulo **function** serve para poder enviar a orde que enviamos ao módulo **mqtt out** e a un indicador de posición. Ao módulo **mqtt out** ten a etiqueta **casa/actuadores/servo**, que o mesmo que o topic.
4. Creamos en Arduinoblocks o programa da imaxe inferior.
5. Instalamos no ESP32 o programa que descargamos de Arduinoblocks , desde o IDE Arduino. Cambiamos os nomes dos clientes MQTT, se fora necesario.
6. Unha vez teñamos creado os nodos en Node-Red, conectados os compoñentes na placa e cargado o programa no ESP32 abrimos o panel de control e comprobamos que o servo funciona correctamente, movéndose á posición que ordenamos.




0º
90º
180º
Estado_servo
casa/actuadores/servo
connected
Posición


0º
90º
180º


hivos
Práctica5_Servo
Subir
Consola
Inicializar
Iniciar Baudios 115200
Enviar "Conectando a Wifi" Salto de línea
Conectar a una red WiFi
SSID ATuaSSID
Clave OTeuContrasinal
Configuración estática
Dirección IP 192 . 168 . 1 . 32
Máscara de subred 255 . 255 . 255 . 0
Puerta de enlace 192 . 168 . 1 . 1
DNS-1 8 . 8 . 8 . 8
DNS-2 8 . 8 . 4 . 4
Enviar crear texto con "Conectado a IP" Salto de línea
Dirección IP
Enviar "Conectando a Raspberry MQTT" Salto de línea
Iniciar
Broker 192.168.1.101
Puerto 1883
Cliente Id
Usuario
Clave
si ¿está conectado?
hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
sino Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
Suscribir Tema "casa/actuadores/servo" > servo
Bucle
si servo = 1
hacer Servo Pin 5 Grados Ángulo 0º Retardo (ms) 0
Enviar "Posición 0º" Salto de línea
si servo = 50
hacer Servo Pin 5 Grados Ángulo 90º Retardo (ms) 0
Enviar "Posición 90º" Salto de línea
si servo = 100
hacer Servo Pin 5 Grados Ángulo 90º Retardo (ms) 0
Enviar "Posición 180º" Salto de línea

6. Práctica 6. LED RGB

Este programa de Arduinoblocks e distinto ao do boletín anterior, olo que o fluxo de Node-Red ten que ser distinto, agora enviaremos un valor para R, G e B, respectivamente para crear a cor que precisemos, polo que teremos que crear 3 novos topics:

casa/actuadores/ledr

casa/actuadores/ledg

casa/actuadores/ledb

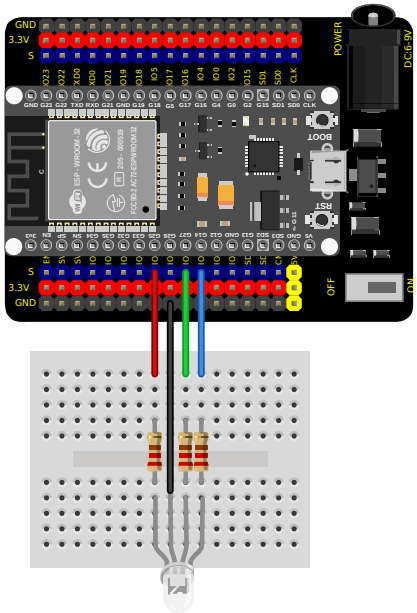
Utilizamos 3 GPIO: 25 para R, 26 para G e 27 para B.

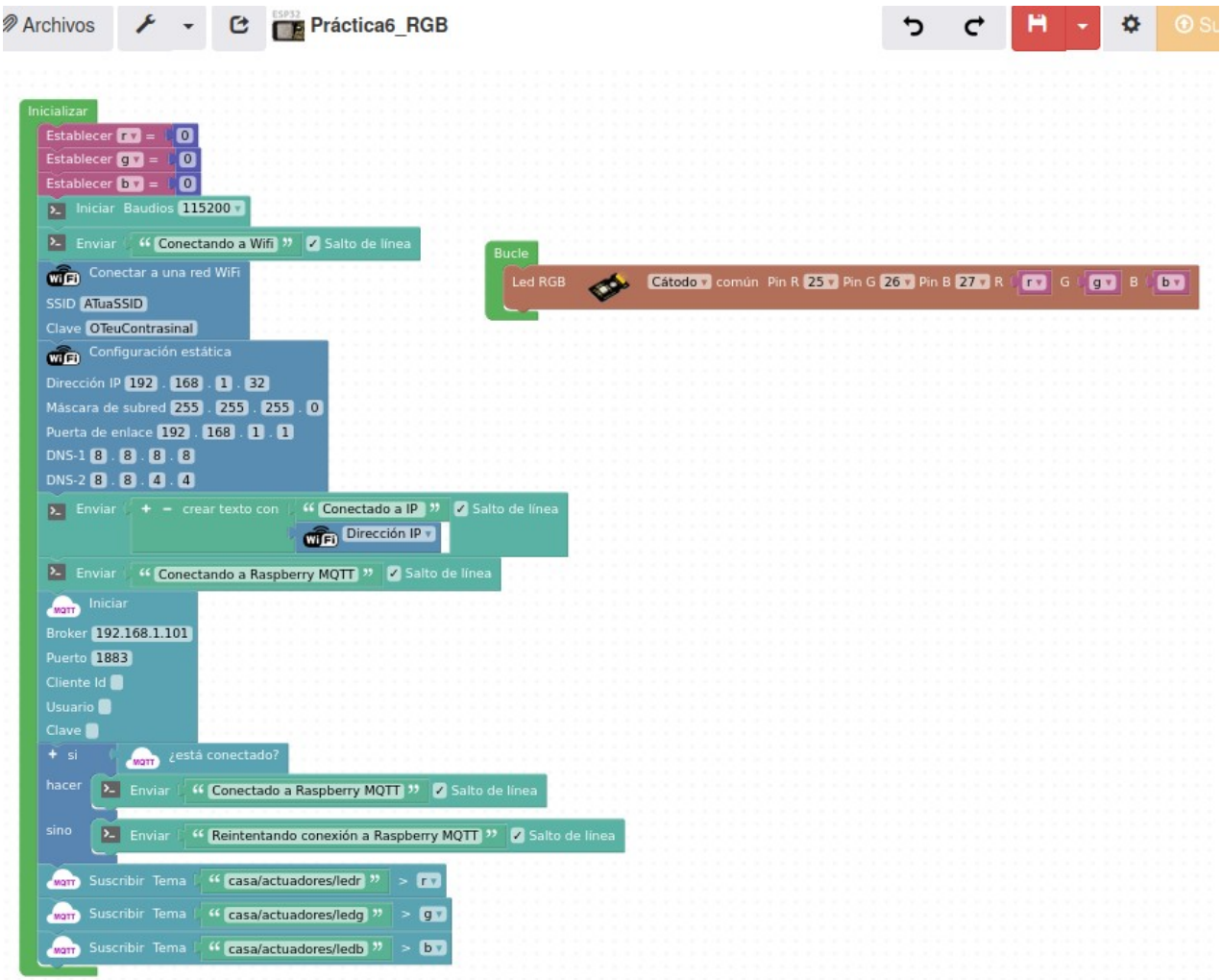
Material necesario:

- ESP32 WROOM + shield
- 3 resistencias 220Ω
- LED RGB 5mm. Importante saber se é de ánodo ou de cátodo común.
- Placa protoboard
- Cables M-F

Pasos a seguir:

1. O esquema de conexións e os nodos en Node-Red son os das imaxes.
2. Na configuración do nodo **mqtt out** escribir **casa/actuadores/ledr g ou b** segundo corresponda.
3. Engadimos un nodo **debug** para poder comprobar a mensaxe enviada cando seleccionamos unha cor no panel.
4. Instalamos no ESP32 o programa descargado de Arduinoblocks, desde o IDE Arduino.
5. Unha vez teñamos creado o fluxo en Node-Red, conectados os compoñentes na placa e cargado o programa no ESP32 abrimos o panel de control e comprobamos que podemos seleccionar a cor e a intensidade.




Archivos
Práctica6_RGB
Inicializar
Establecer r = 0
Establecer g = 0
Establecer b = 0
Iniciar Baudios 115200
Enviar "Conectando a Wifi" Salto de línea
Conectar a una red WiFi
SSID ATuaSSID
Clave OTeuContrasinal
Configuración estática
Dirección IP 192 . 168 . 1 . 32
Máscara de subred 255 . 255 . 255 . 0
Puerta de enlace 192 . 168 . 1 . 1
DNS-1 8 . 8 . 8 . 8
DNS-2 8 . 8 . 4 . 4
Enviar crear texto con "Conectado a IP" Salto de línea
Dirección IP
Enviar "Conectando a Raspberry MQTT" Salto de línea
Iniciar
Broker 192.168.1.101
Puerto 1883
Cliente Id
Usuario
Clave
si ¿está conectado?
hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
sino Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
Suscribir Tema "casa/actuadores/ledr" > r
Suscribir Tema "casa/actuadores/ledg" > g
Suscribir Tema "casa/actuadores/ledb" > b
Bucle
Led RGB Cátodo común Pin R 25 Pin G 26 Pin B 27 R r G g B b

7. Práctica 7. Sensor Temperatura e humidade DHT11.

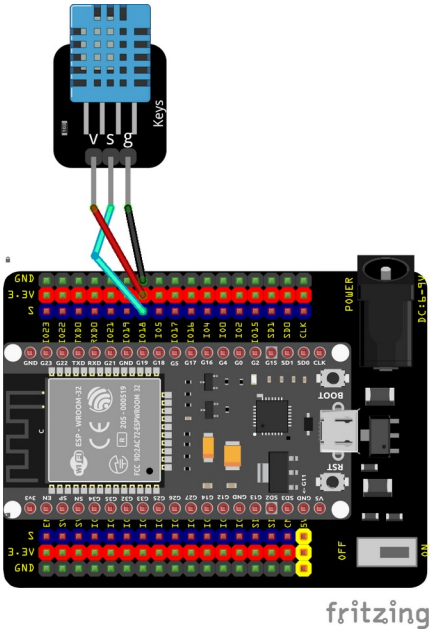
Conectamos o sensor ao GPIO 18.

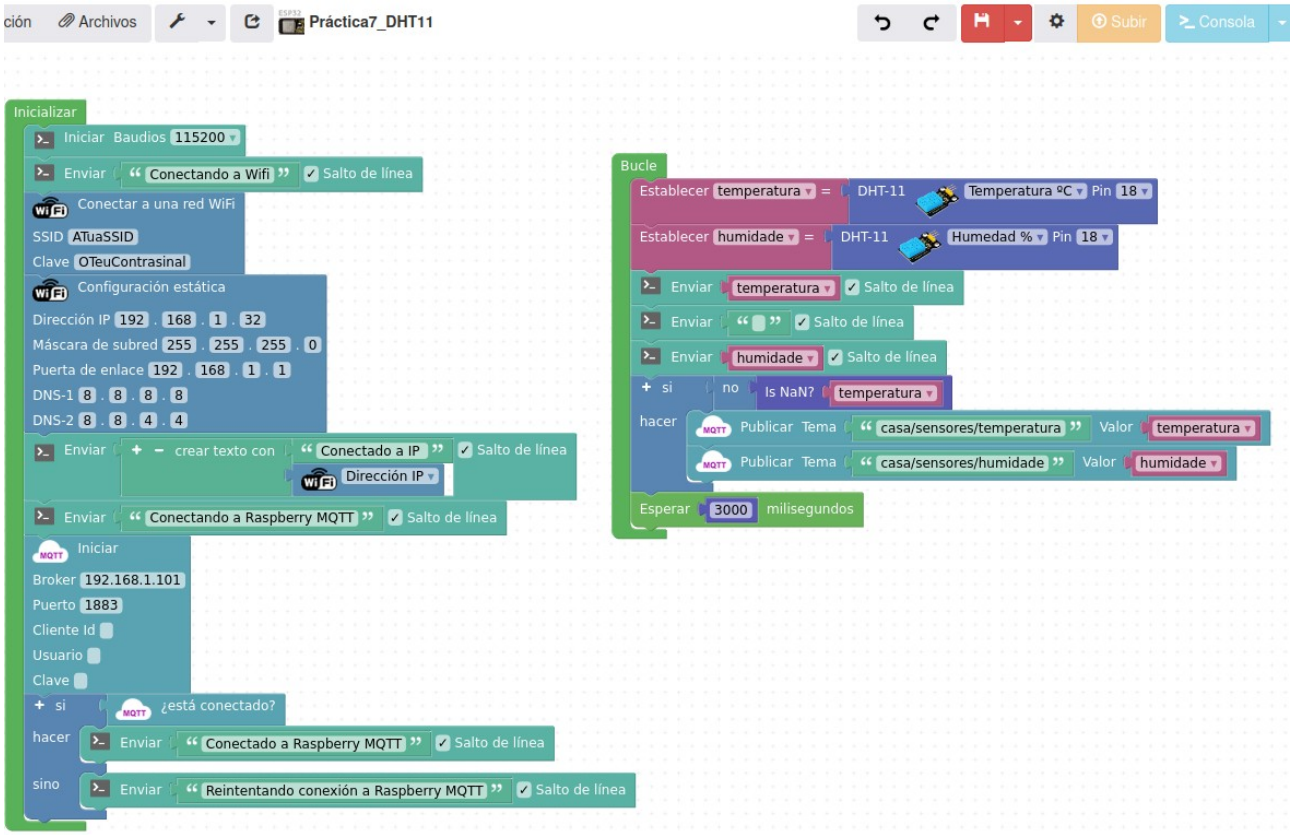
Material necesario:

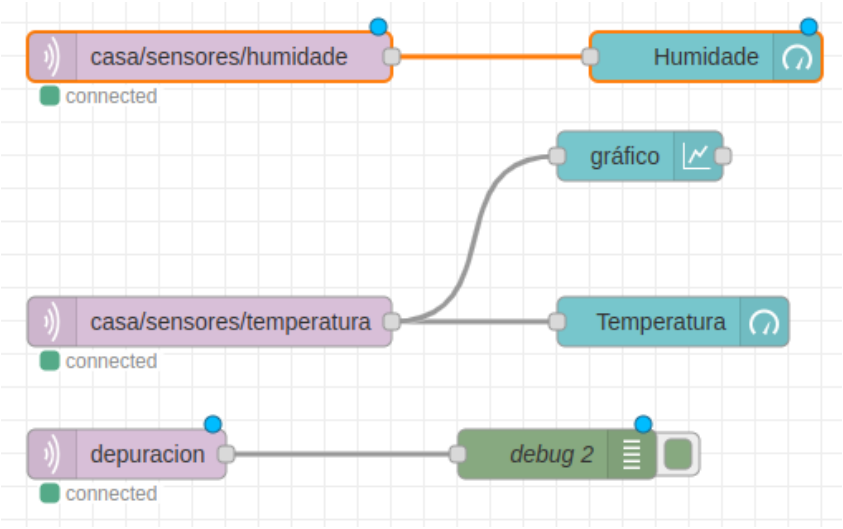
- ESP32 WROOM + shield
- Sensor DHT11
- Cables F-F

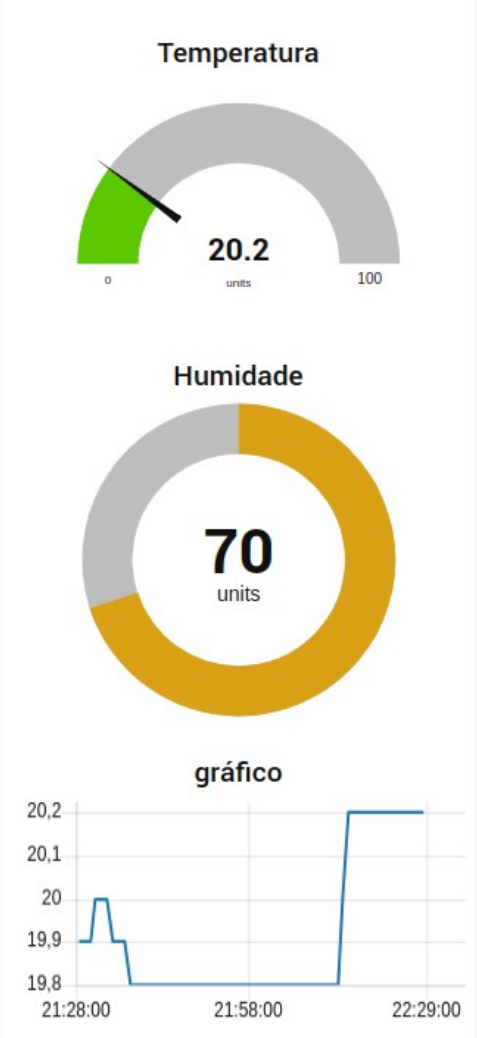
Pasos a seguir:

1. O esquema de conexións e os nodos en Node-Red son os das imaxes.
2. Na configuración do nodo **mqtt out,** que utilizamos para a humidade,escribir **casa/sensores/humidade** como **Topic**
3. Na configuración do nodo **mqtt out,** que utilizamos para a temperatura,escribir **casa/sensores/temperatura** como **Topic**
4. Engadimos un nodo **debug** para poder comprobar se o sensor está enviando mensaxes ao broker. Na configuración do nodo **mqtt in** que conectamos ao **debug** so temos que escribir correctamente o topic (depuracion) e seleccionar o noso broker mqtt.
5. Para visualizar os datos utilizamos nodo tipo **gauge** para a temperatura e tipo **donut** para a humidade, e os chamaremos temperatura e humidade. Tamén cambiamos o rango de unidades, de 0 a 100.
6. No caso da temperatura engadimos un nodo **chart**, que chamamos **gráfico**, como exemplo de visualización da evolución dos datos dun sensor.
7. Creamos un programa, como o da imaxe, en Arduinoblocks, o descargamos, abrimos desde IDE Arduino e o cargamos na ESP32.
8. Unha vez teñamos creado o fluxo en Node-Red, conectados os compoñentes na placa e cargado o programa no ESP32 abrimos o panel de control e comprobamos que podemos ver a temperatura e humidade.


V S g
Keys
GND
3.3V
POWER
BOOT
RST
OFF
ON
fritzing


ción
Archivos
Práctica7_DHT11
Subir
Consola
Inicializar
Iniciar Baudios 115200
Enviar "Conectando a Wifi" Salto de línea
Conectar a una red WiFi
SSID ATuaSSID
Clave OTeuContrasinal
Configuración estática
Dirección IP 192 . 168 . 1 . 32
Máscara de subred 255 . 255 . 255 . 0
Puerta de enlace 192 . 168 . 1 . 1
DNS-1 8 . 8 . 8 . 8
DNS-2 8 . 8 . 4 . 4
Enviar crear texto con "Conectado a IP" Salto de línea
Dirección IP
Enviar "Conectando a Raspberry MQTT" Salto de línea
Iniciar
Broker 192.168.1.101
Puerto 1883
Cliente Id
Usuario
Clave
si ¿está conectado?
hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
sino Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
Bucle
Establecer temperatura = DHT-11 Temperatura ºC Pin 18
Establecer humidade = DHT-11 Humedad % Pin 18
Enviar temperatura Salto de línea
Enviar Salto de línea
Enviar humidade Salto de línea
si no Is NaN? temperatura
hacer Publicar Tema "casa/sensores/temperatura" Valor temperatura
Publicar Tema "casa/sensores/humidade" Valor humidade
Esperar 3000 milisegundos


casa/sensores/humidade
connected
Humidade
gráfico
casa/sensores/temperatura
connected
Temperatura
depuracion
connected
debug 2


Temperatura
20.2
0
units
100
Humidade
70
units
gráfico
20,2
20,1
20
19,9
19,8
21:28:00
21:58:00
22:29:00

8. Práctica 8. Botón.

O botón e o exemplo dun sensor dixital, que pode adoptar dous estados pulsado - non pulsado; 0 – 1; 0 – 3,3V, ...

Para o noso exemplo conectaremos o botón en modo pull-down. (https://tecnoloxia.org/robotica/pull-up-e-pull-down/)

Para poder comprobar como funciona o botón utilizaremos un LED tal como o fixemos na práctica 1, no GPIO2, pero esta vez o LED acenderá e apagará cando pulsemos o botón conectado ao GPIO4.

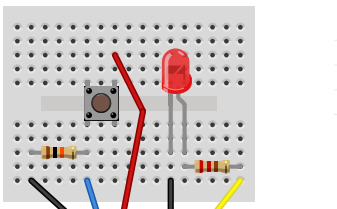
Neste caso, ante a imposibilidade de ligar o topic “casa/sensores/boton” a unha variable booleana facemos un programa que farña que o LED acenda ao pulsar o botón e un nodo **switch** ligado ao LED cambiará de posición.

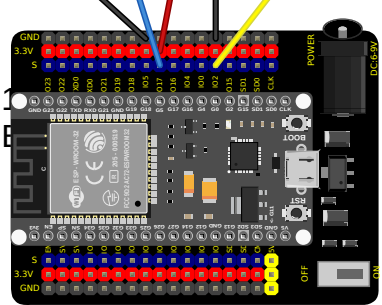
Material necesario:

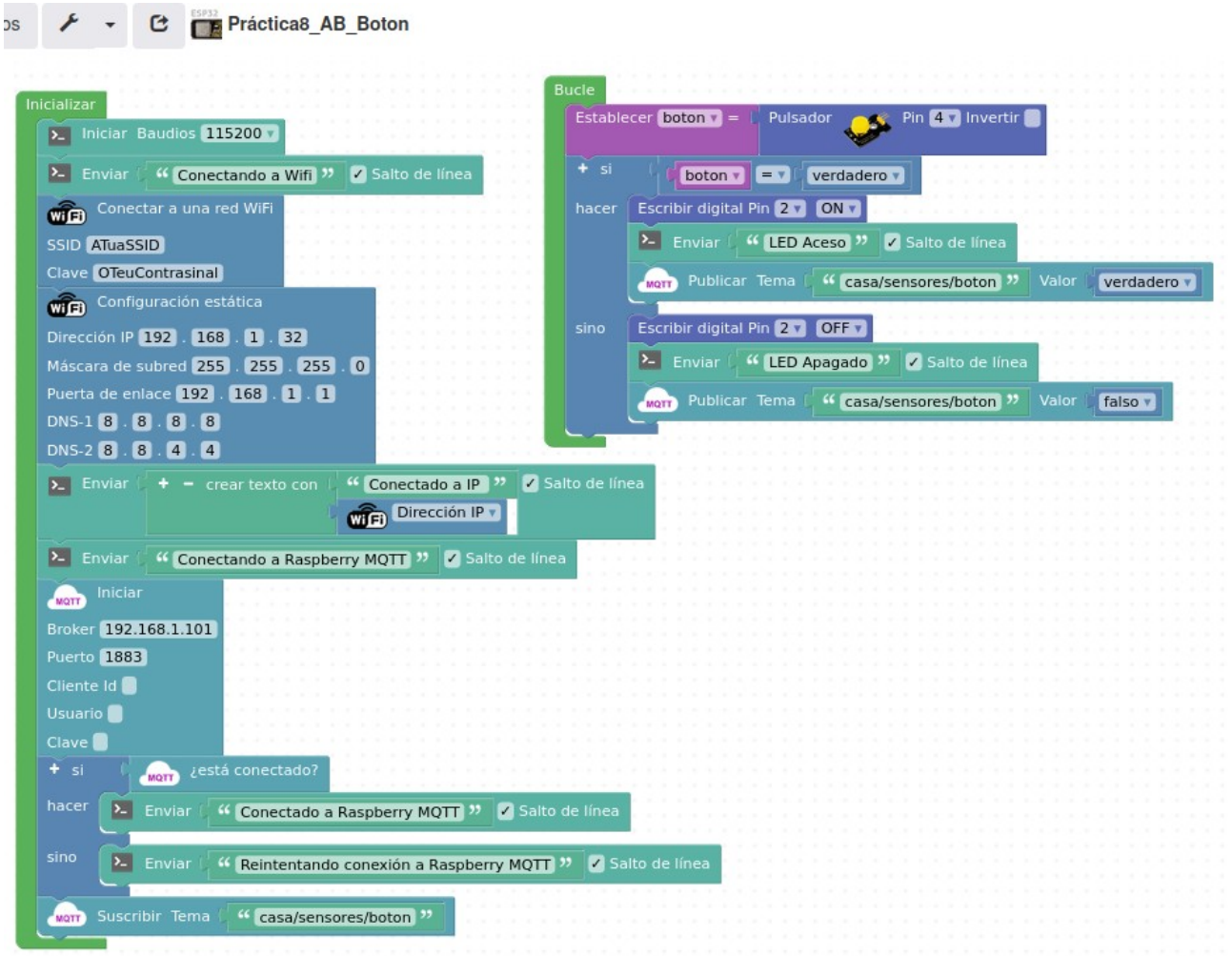
- ESP32 WROOM + shield
- Pulsador
- Resistencia de 10kΩ
- LED
- Resistencia de 220Ω
- Protoboard
- Cables M-F

Pasos a seguir:

1.
2.
3. Creamos un programa, como o da imaxe, en Arduinoblocks, o descargamos, abrimos desde IDE Arduino e o cargamos na ESP32.
4. Unha vez teñamos conectados os compoñentes na placa e cargado o programa no ESP32 comprobamos que ao pulsar o botón acende o LED e cando non o pulsamos o LED permanece apagado.






Práctica8_AB_Boton
Inicializar
Iniciar Baudios 115200
Enviar "Conectando a Wifi" Salto de línea
Conectar a una red WiFi
SSID ATuaSSID
Clave OTeuContrasinal
Configuración estática
Dirección IP 192 . 168 . 1 . 32
Máscara de subred 255 . 255 . 255 . 0
Puerta de enlace 192 . 168 . 1 . 1
DNS-1 8 . 8 . 8 . 8
DNS-2 8 . 8 . 4 . 4
Enviar crear texto con "Conectado a IP" Salto de línea
Dirección IP
Enviar "Conectando a Raspberry MQTT" Salto de línea
Iniciar
Broker 192.168.1.101
Puerto 1883
Cliente Id
Usuario
Clave
si ¿está conectado?
hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
sino Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
Suscribir Tema "casa/sensores/boton"
Bucle
Establecer boton = Pulsador Pin 4 Invertir
si boton = verdadero
hacer Escribir digital Pin 2 ON
Enviar "LED Aceso" Salto de línea
Publicar Tema "casa/sensores/boton" Valor verdadero
sino Escribir digital Pin 2 OFF
Enviar "LED Apagado" Salto de línea
Publicar Tema "casa/sensores/boton" Valor falso

9. Práctica 9. Sensor PIR + zoador.

Tedes información sobre o funcionamento do sensor na práctica do boletín anterior.

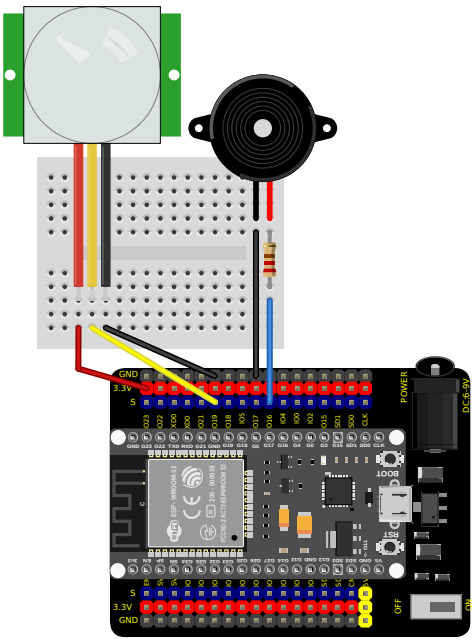
Sensor PIR conectado ao GPIO19, e zoador conectado ao GPIO16.

Material necesario:

- ESP32 WROOM + shield
- Sensor PIR
- Zoador
- Resistencia de 220Ω
- Protoboard
- Cables M-F

Pasos a seguir:

1. O esquema de conexións e os nodos en Node-Red son os das imaxes.
2. Na configuración do nodo **mqtt out,** escribir **casa/sensores/PIR** como **Topic**
3. No nodo **function** cargamos o programa en JavaScript **function_pir.js**, que fará que cando chegue a mensaxe de botón pulsado ao nodo mqtt in se acenda o LED.
4. O nodo **debug** que chamamos enviado serve para comprobar que mensaxe chega ao topic **casa/sensores/PIR**
5. Instalamos no ESP32 o programa **Practica9_PIR.ino**, desde o IDE Arduino.
6. Unha vez teñamos creado o fluxo en Node-Red, conectados os compoñentes na placa e cargado o programa no ESP32 abrimos o panel de control e comprobamos que ao pulsar o botón acende o LED e cando non o pulsamos o LED permanece apagado.



Práctica9_AB_PIR

```
Inicializar
  Iniciar Baudios 115200
  Enviar "Conectando a Wifi" Salto de línea
  Conectar a una red WiFi
  SSID ATuaSSID
  Clave OTeuContrasinal
  Configuración estática
  Dirección IP 192 . 168 . 1 . 32
  Máscara de subred 255 . 255 . 255 . 0
  Puerta de enlace 192 . 168 . 1 . 1
  DNS-1 8 . 8 . 8 . 8
  DNS-2 8 . 8 . 4 . 4
  Enviar + - crear texto con "Conectado a IP" Salto de línea
                              Dirección IP
  Enviar "Conectando a Raspberry MQTT" Salto de línea
  Iniciar
  Broker 192.168.1.101
  Puerto 1883
  Cliente Id
  Usuario
  Clave
  + si ¿está conectado?
  hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
  sino  Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
  Suscribir Tema "casa/sensores/PIR"

Bucle
  Establecer pir = Detector de movimiento (PIR) Pin 19
  + si pir = verdadero
  hacer Escribir digital Pin 12 ON
        Enviar "Alarma" Salto de línea
        Publicar Tema "casa/sensores/PIR" Valor verdadero
  sino  Escribir digital Pin 12 OFF
        Enviar "Seguro" Salto de línea
        Publicar Tema "casa/sensores/PIR" Valor falso
```

10. Práctica 10. Sensor de luz.

Utilizaremos un LDR conectado a un divisor resistivo.

Todos os modelos de ESP32 teñen dispoñibles dous ADC (analog-to-dixital converter) SAR (Successive Approximation Register) de 12 bits, denominados ADC1 e ADC2.

Cunha resolución de 12bits, unha resolución de 3.3 voltios / 4096 unidades, é equivalente a 0.8 mV por paso. Ademais, podemos configurar mediante programación a resolución do ADC e o rango da canle.

A tensión máxima admisible é 3V3. Aínda que moitos GPIO do ESP32 son tolerantes a 5V, o ADC non o é. Así que non metades máis de 3.6V nun pin que usedes como entrada analóxica, ou danaredes o ADC. (Fonte: Luis Llamas)

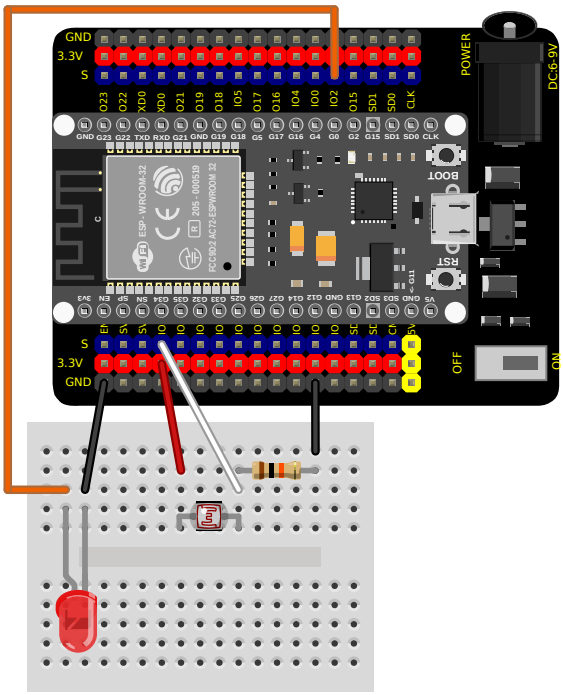
Conectamos o LDR ao GPIO 34.

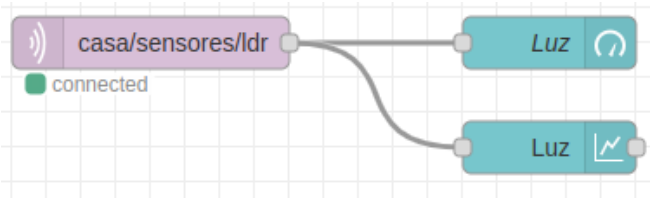
Material necesario:

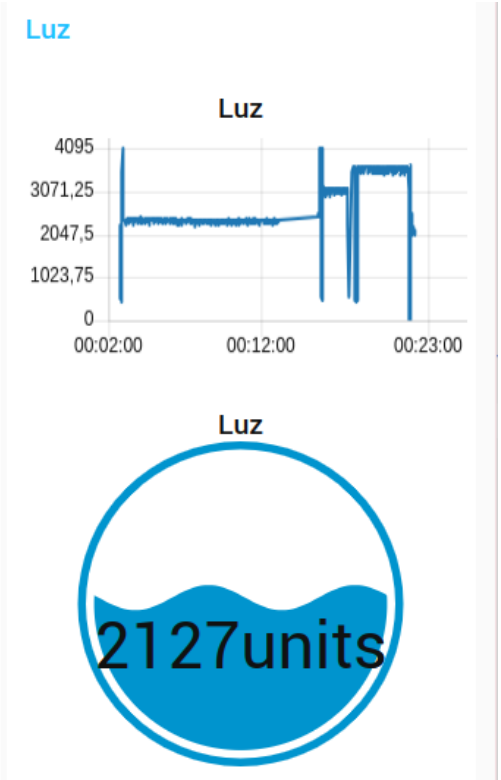
- ESP32 WROOM + shield
- LDR
- Resistencia 10kΩ
- Protoboard
- Cables M-F

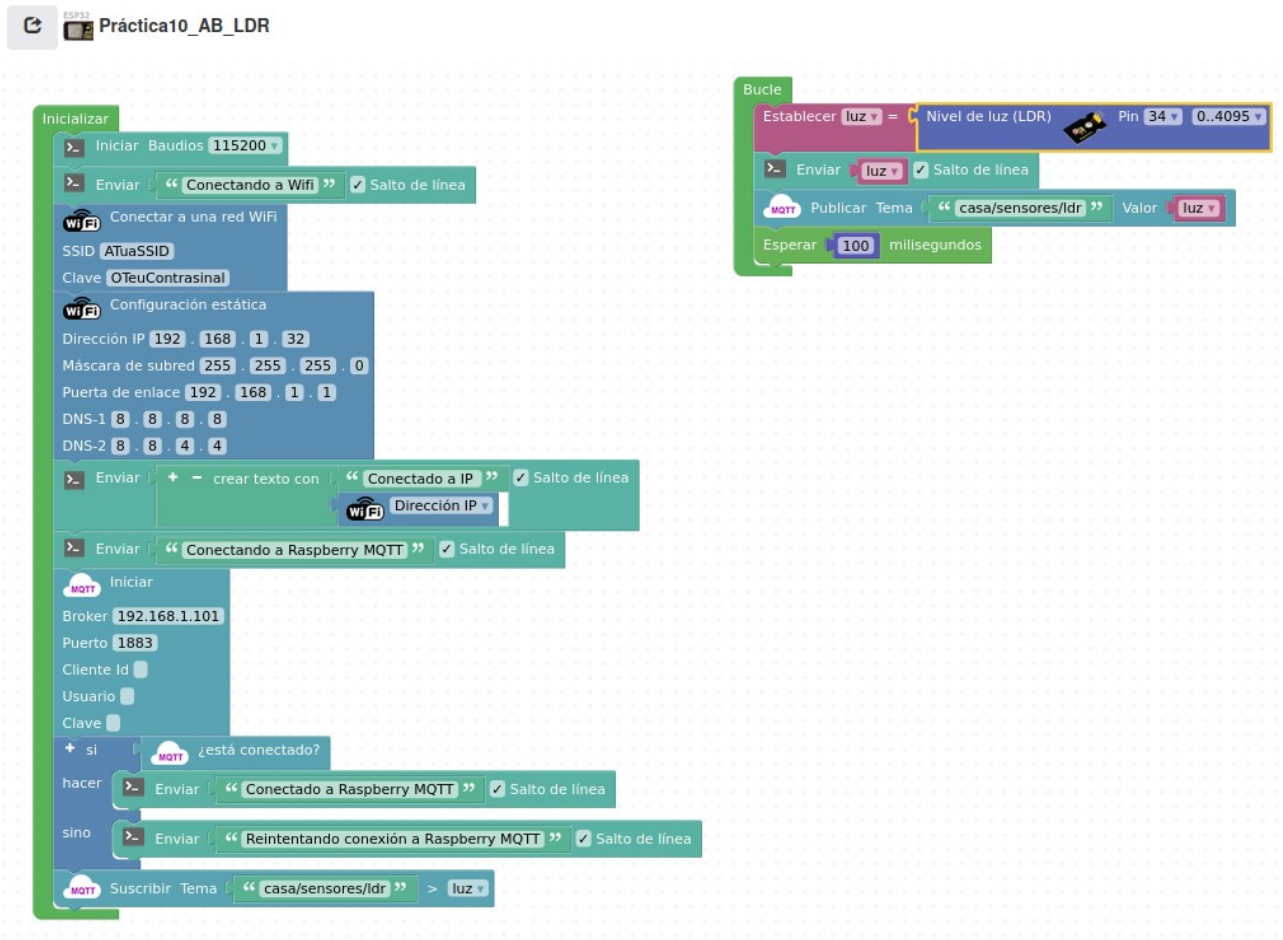
Pasos a seguir:

1. O esquema de conexións e os nodos en Node-Red son os das imaxes.
2. Na configuración do nodo **mqtt out,** escribir **casa/sensores/ldr** como **Topic**
3. Configuramos un nodo **gauge** e outro **grafic**, igual que o sensor de humidade, pero neste caso indicará a luz, que será de entre 0 e 4095, xa que o conversor ADC e de 12 dixitos.
4. Instalamos no ESP32 o programa Practica10_ldr.ino, desde o IDE Arduino.
5. Unha vez teñamos creado o fluxo en Node-Red, conectados os compoñentes na placa e cargado o programa no ESP32 abrimos o panel de control e comprobamos que aparecen indicadas as variacións de luz nos indicadores gráficos.


GND
3.3V
POWER
DC 6-9V
BOOT
RST
OFF
ON


casa/sensores/ldr
connected
Luz
Luz


Luz
Luz
4095
3071,25
2047,5
1023,75
0
00:02:00
00:12:00
00:23:00
Luz
2127units


Práctica10_AB_LDR
Inicializar
Iniciar Baudios 115200
Enviar "Conectando a Wifi" Salto de línea
Conectar a una red WiFi
SSID ATuaSSID
Clave OTeuContrasinal
Configuración estática
Dirección IP 192 . 168 . 1 . 32
Máscara de subred 255 . 255 . 255 . 0
Puerta de enlace 192 . 168 . 1 . 1
DNS-1 8 . 8 . 8 . 8
DNS-2 8 . 8 . 4 . 4
Enviar crear texto con "Conectado a IP" Salto de línea
Dirección IP
Enviar "Conectando a Raspberry MQTT" Salto de línea
Iniciar
Broker 192.168.1.101
Puerto 1883
Cliente Id
Usuario
Clave
si ¿está conectado?
hacer Enviar "Conectado a Raspberry MQTT" Salto de línea
sino Enviar "Reintentando conexión a Raspberry MQTT" Salto de línea
Suscribir Tema "casa/sensores/ldr" > luz
Bucle
Establecer luz = Nivel de luz (LDR) Pin 34 0..4095
Enviar luz Salto de línea
Publicar Tema "casa/sensores/ldr" Valor luz
Esperar 100 milisegundos

11. Práctica 11. Potenciómetro.

Esta práctica é igual que a anterior, só teremos que cambiar o LDR por unha resistencia variable, xa que nos dous casos o que temos é un sensor analóxico que envía unha tensión entre 0 e 3,3 voltios a un GPIO ADC do ESP32. Como o conversor analóxico-dixital é de 12 díxitos, os valores que obteremos serán de entre 0 e 4095.