

Programación de ESP32 e integración con Amazon Alexa.



AIDA SCHOOL



2nd International School on Artificial Intelligence and Data Science Applications Centro ESenCIA

2da Escuela Internacional sobre Aplicaciones en Inteligencia Artificial y Ciencia de Datos

Tutorial – Dr. Miguel Solís



Universidad
Andrés Bello®
Conectar • Innovar • Liderar

14 de octubre de 2025

Material disponible en www.miguelsolis.info

Identificación del Relator



Chile Centro
Section



NEURO-ROBOTICS SYSTEMS

IEEE TECHNICAL COMMITTEE ON

Director Ing. en Automatización y Robótica

Doctor. en Ing. Informática

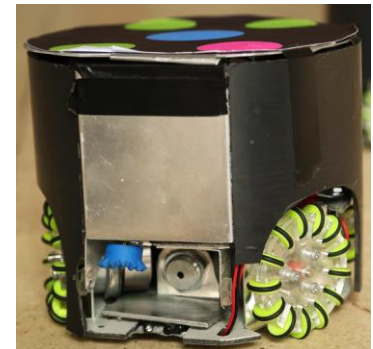
Magister en Ciencias de la Ing. Electrónica

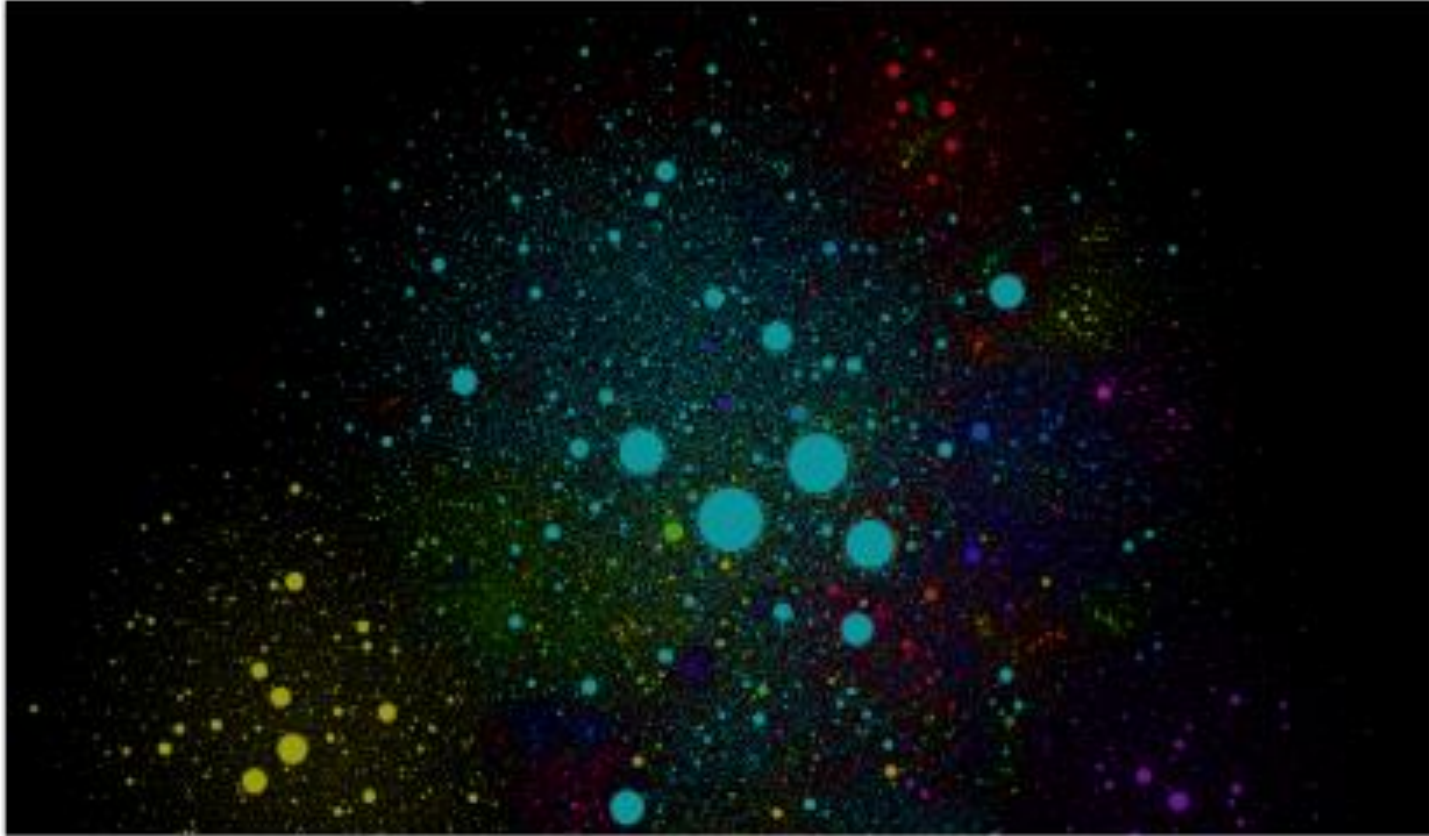


Miguel Solis
msolisc@ieee.org
www.miguelsolis.info

ICDL-EPIROB

IEEE International Conference on Development and Learning and Epigenetic Robotics.





internet-map.net

Internet (de las cosas?)

- Internet normalmente conecta personas unas con otras
- El objetivo ahora es interconectar objetos, con personas y con objetos entre sí



Imagen extraída de linnesoft.com

- Es la red formada por objetos físicos (o cosas) que integran electrónica, software, sensores y conectividad para obtener un mejor valor y servicio al intercambiar datos con el proveedor, operador y/u otros dispositivos conectados.

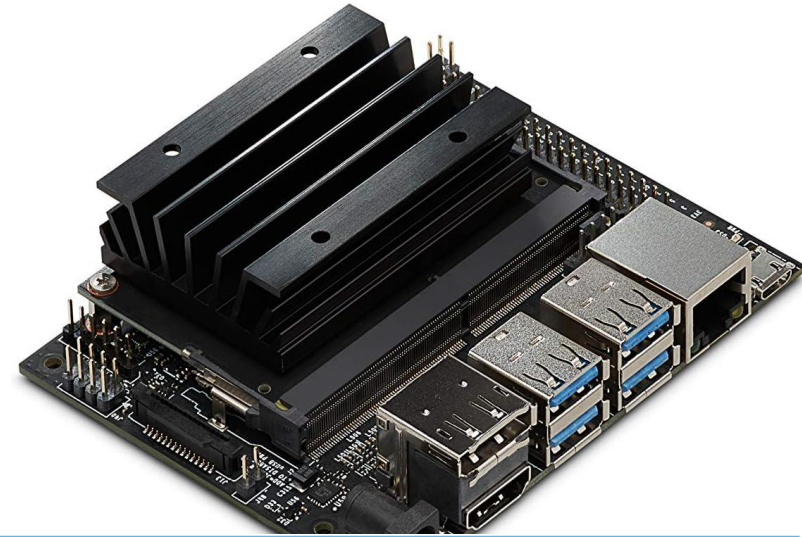
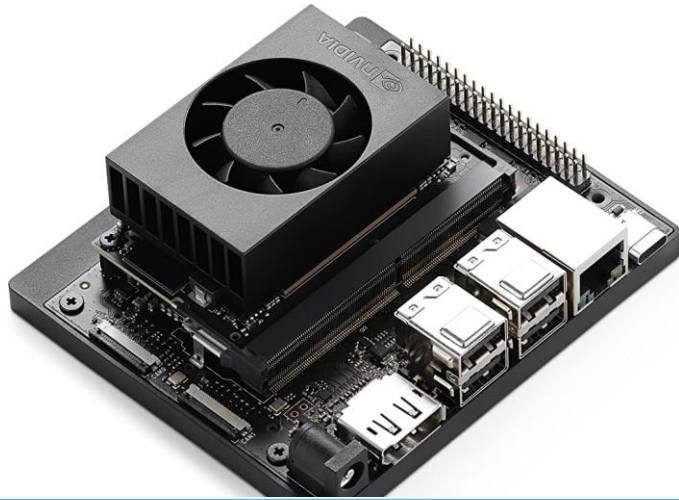
— Wikipedia (inglés)

- Se logra al incluir transmisores-receptores móviles de corto alcance, en un arreglo más amplio de dispositivos y objetos del día a día, permitiendo nuevas formas de comunicación entre personas y objetos, y entre objetos por sí mismo.

— WSIS (World Summit on the Information Society)

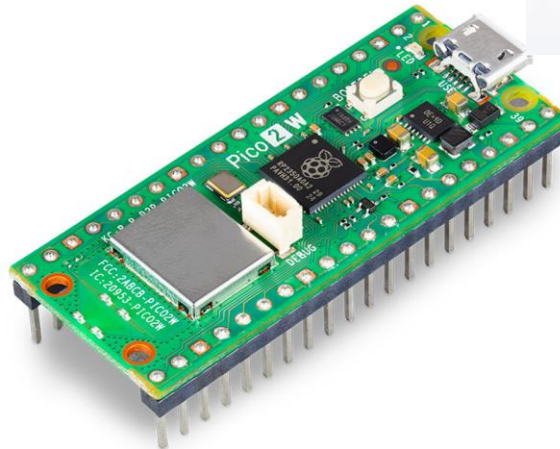
- WSN: Wireless Sensor Networks
- IoT $\neq$ WSN
- IoT $\geq$ WSN
- Sensores son necesarios para detectar cambios en el estado físico del entorno y sus objetos

Tarjetas de Desarrollo – SBC (Jetson nVidia)

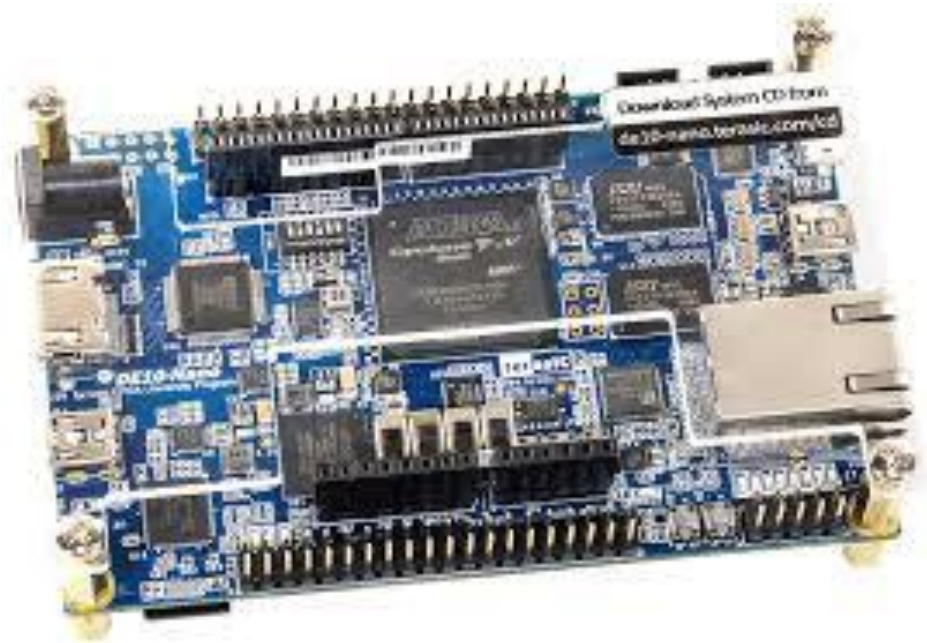
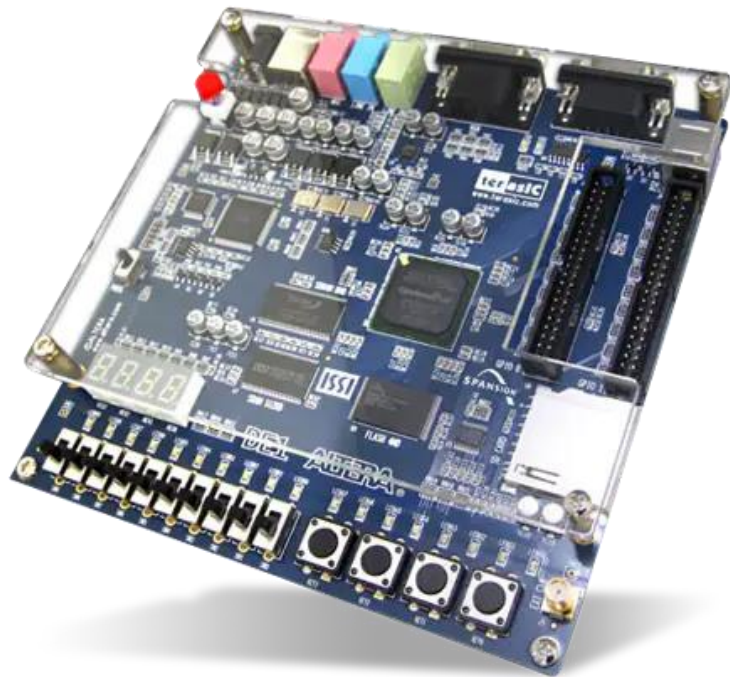


Característica	Jetson Orin Nano de 4 GB	Jetson Orin Nano de 8 GB	Jetson Orin NX de 8 GB	Jetson Orin NX de 16 GB	Jetson AGX Orin 32 GB	Jetson AGX Orin Industrial	Jetson AGX Orin de 64 GB
Rendimiento de la IA	20 TOPS	40 TOPS	70 TOPS	100 TOPS	200 TOPS	248 TOPS	275 TOPS
GPU	Ampere de 512 núcleos, con 16 núcleos Tensor	Ampere de 1024 núcleos, con 32 núcleos Tensor	1024 amperios de núcleo, con 32 núcleos tensores	1024 amperios de núcleo, con 32 núcleos tensores	1792 amperios de núcleo, con 56 núcleos tensores	2048 amperios de núcleo, con 64 núcleos tensores	2048 amperios de núcleo, con 64 núcleos tensores
UPC	Arm® Cortex®-A78AE de 6 núcleos	Arm® Cortex®-A78AE de 6 núcleos	Arm® Cortex®-A78AE de 6 núcleos	Arm® Cortex®-A78AE de 8 núcleos	Procesador Arm® Cortex®-A78AE de 8 núcleos	Arm® Cortex®-A78AE de 12 núcleos	Arm® Cortex®-A78AE de 12 núcleos
Memoria	LPDDR5 de 4 GB y 64 bits a 34 GB/s	LPDDR5 de 8 GB y 128 bits a 68 GB/s	LPDDR5 de 8 GB y 128 bits a 102,4 GB/s	LPDDR5 de 16 GB y 128 bits a 102,4 GB/s	32 GB, 256 bits, LPDDR5, 205 GB/s	64 GB, 256 bits, LPDDR5, 205 GB/s	64 GB, 256 bits, LPDDR5, 205 GB/s

Tarjetas de Desarrollo – SBC (Raspberry Pi)



Tarjetas de Desarrollo - FPGA

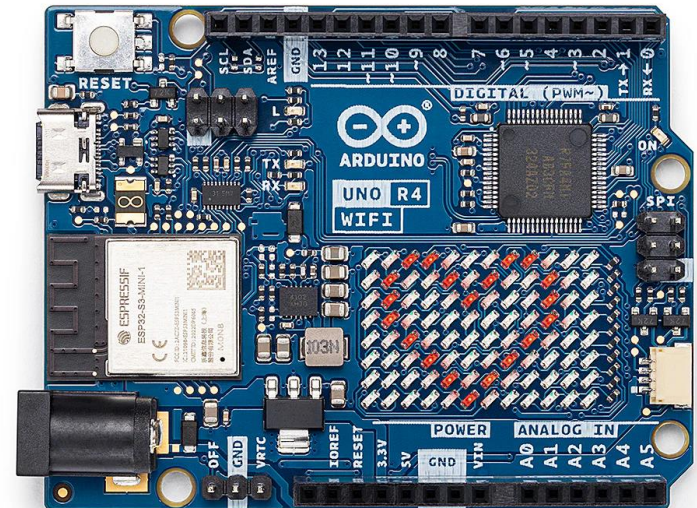
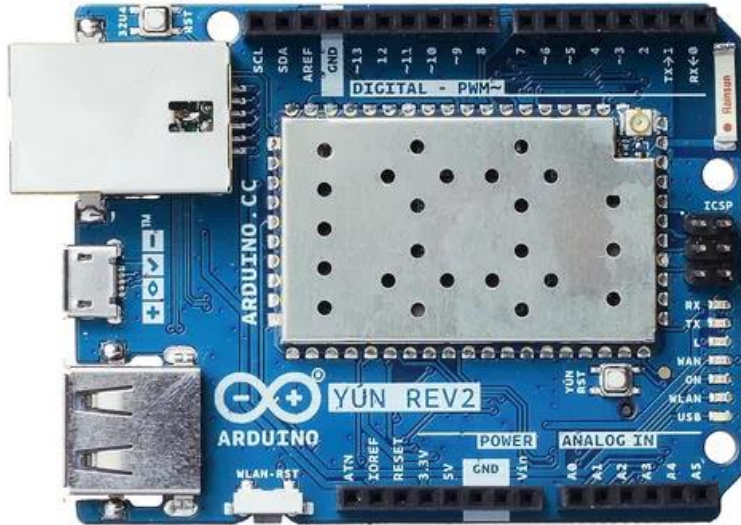


```
signal seqcount :      unsigned(10 downto 0) := "000000000000"  
signal Eng_Sck_en :    STD_LOGIC := '0';  
signal clkdiv2 :       STD_LOGIC := '0';  
entity acqeng is  
    Port (  
        Mclk : in  STD_LOGIC;  
        Rst  : in  STD_LOGIC;  
        Sclk : in  STD_LOGIC;  
        SDI  : in  STD_LOGIC;  
        SDO  : out STD_LOGIC;
```

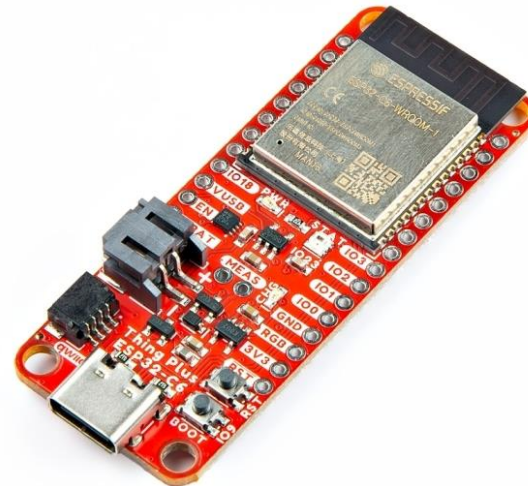
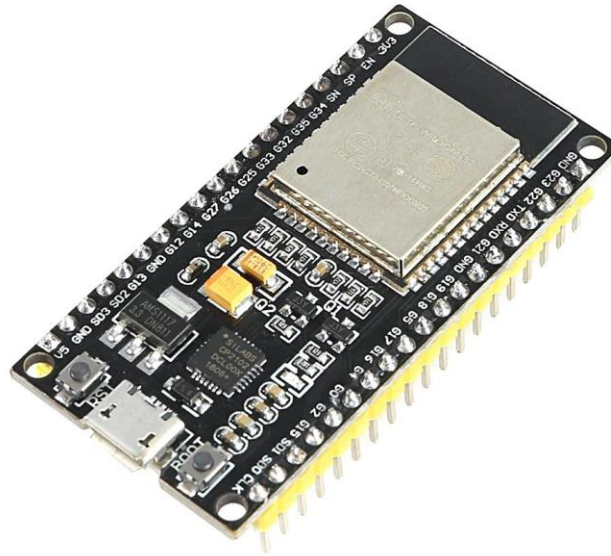
Tarjetas de Desarrollo - Arduino



Universidad
Andrés Bello®



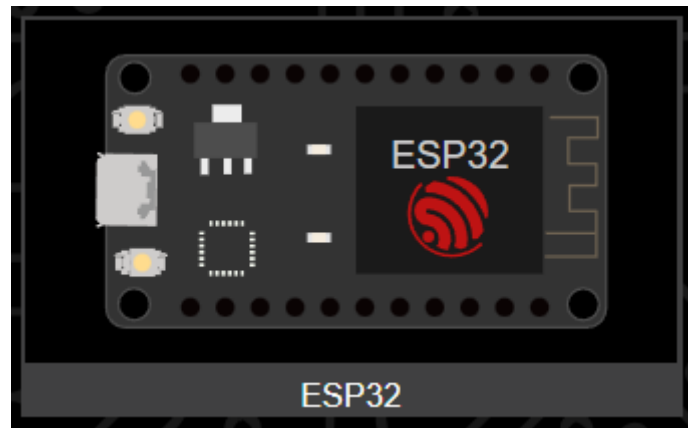
Tarjetas de Desarrollo – ESP32





<https://wokwi.com/projects/new/esp32>

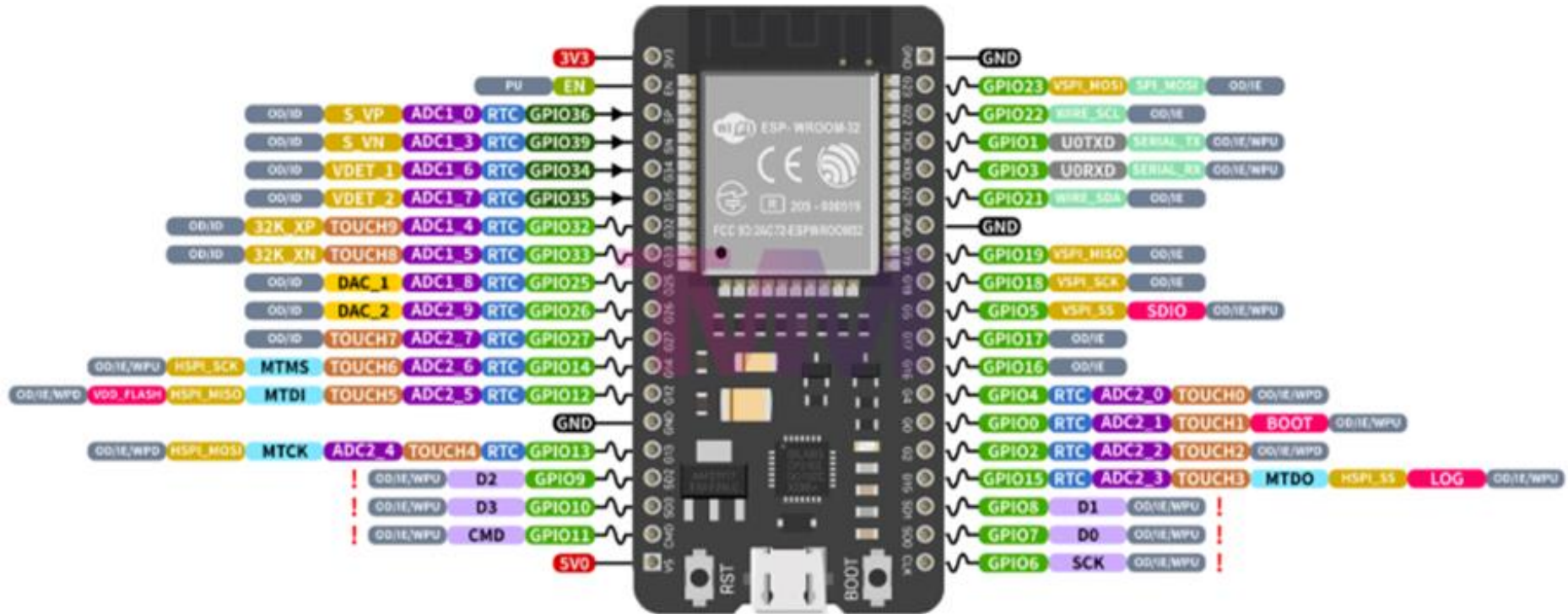
(no es necesario crear una cuenta ni realizar ningún tipo de suscripción)



Agregar componentes en simulación

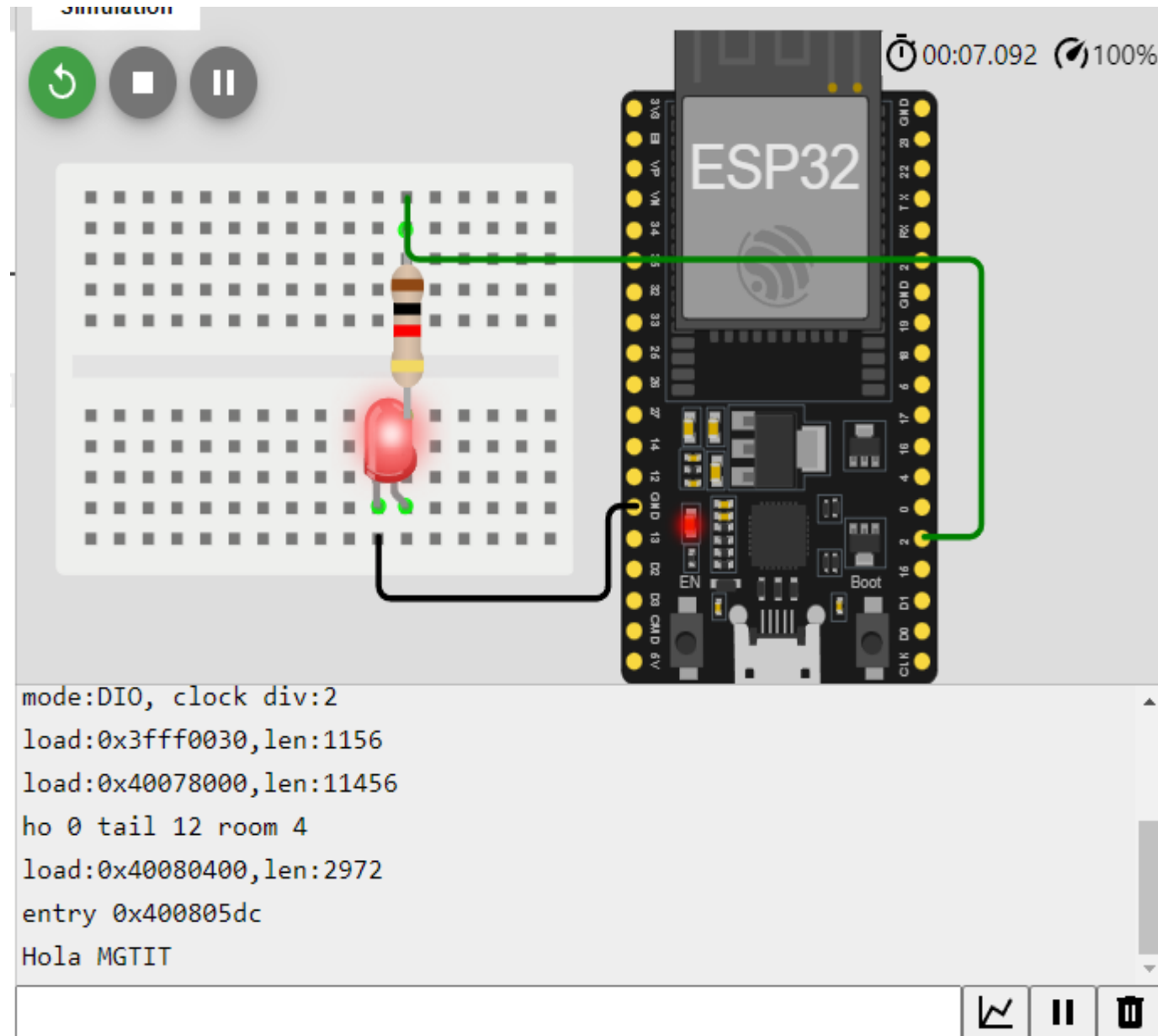


ESP32-WROOM-32 Pinout

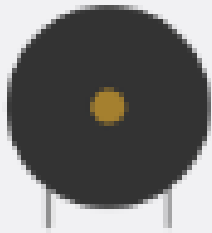


Agregar componentes en simulación

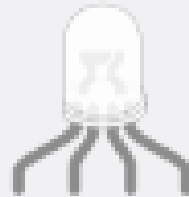
```
void setup() {  
  pinMode(2, OUTPUT);  
  Serial.begin(9600);  
  Serial.println("Hola MGTIT");  
}  
  
void loop() {  
  digitalWrite(2,HIGH);  
  delay(1000);  
  digitalWrite(2,LOW);  
  delay(1000);  
}
```



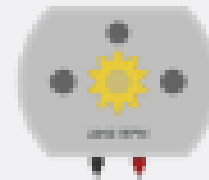
Otras salidas (actuadores eléctricos)



Piezo

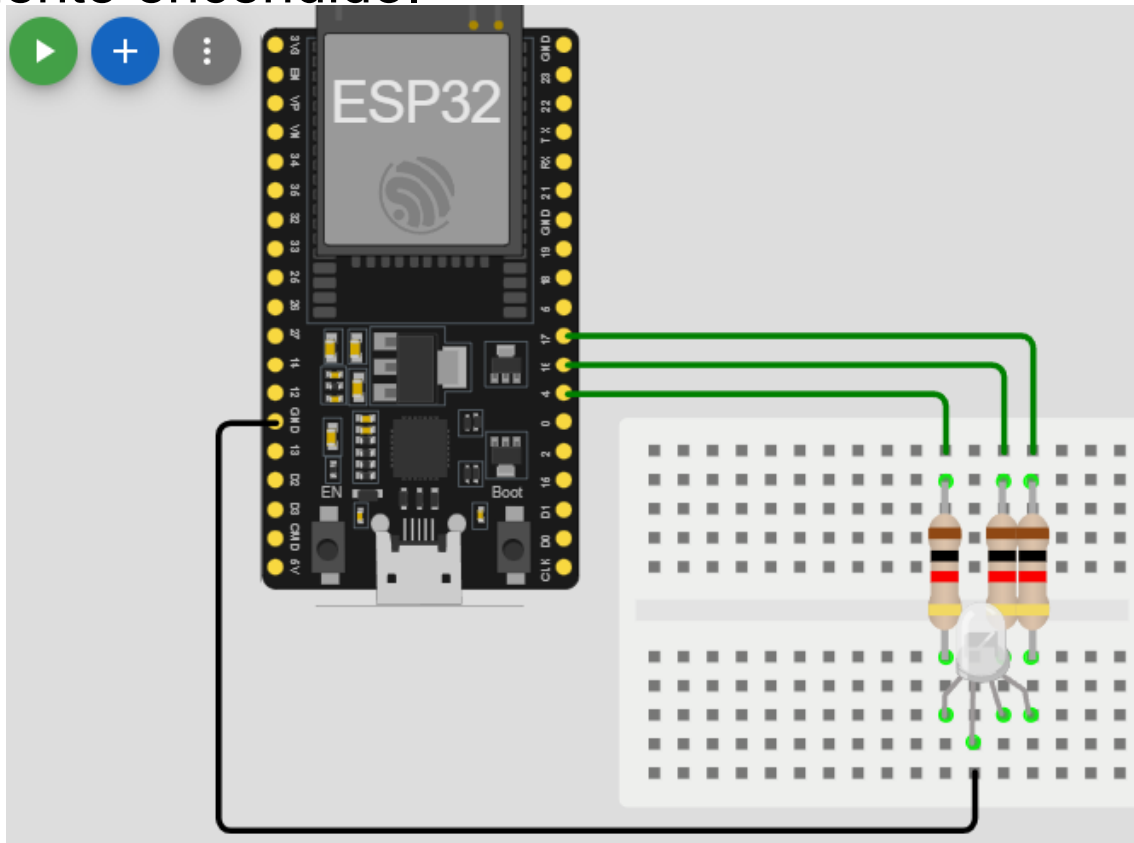


LED RGB



Motor de CC

- Considere el circuito indicado, y construya un programa que encienda de manera alternada cada 1 segundo el color rojo, verde y azul de un LED RGB, indicando por el monitor serie el nombre del color que se encuentra actualmente encendido.



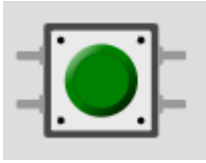
<https://wokwi.com/projects/444779037493342209>

```
int rojo = 4, verde = 16, azul = 17, tiempo = 1000;
```

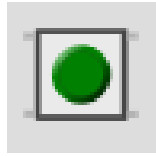
```
void setup() {  
    pinMode(rojo, OUTPUT);  
    pinMode(verde, OUTPUT);  
    pinMode(azul, OUTPUT);  
    Serial.begin(9600);  
    Serial.println("Hello AIDA SCHOOL");  
}
```

```
void loop() {  
    digitalWrite(rojo, LOW);  
    digitalWrite(verde, HIGH);  
    delay(tiempo);  
    digitalWrite(verde, LOW);  
    digitalWrite(azul, HIGH);  
    delay(tiempo);  
    digitalWrite(azul, LOW);  
    digitalWrite(rojo, HIGH);  
    delay(tiempo);  
}
```

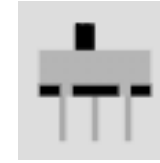
Entradas digitales



Pulsador



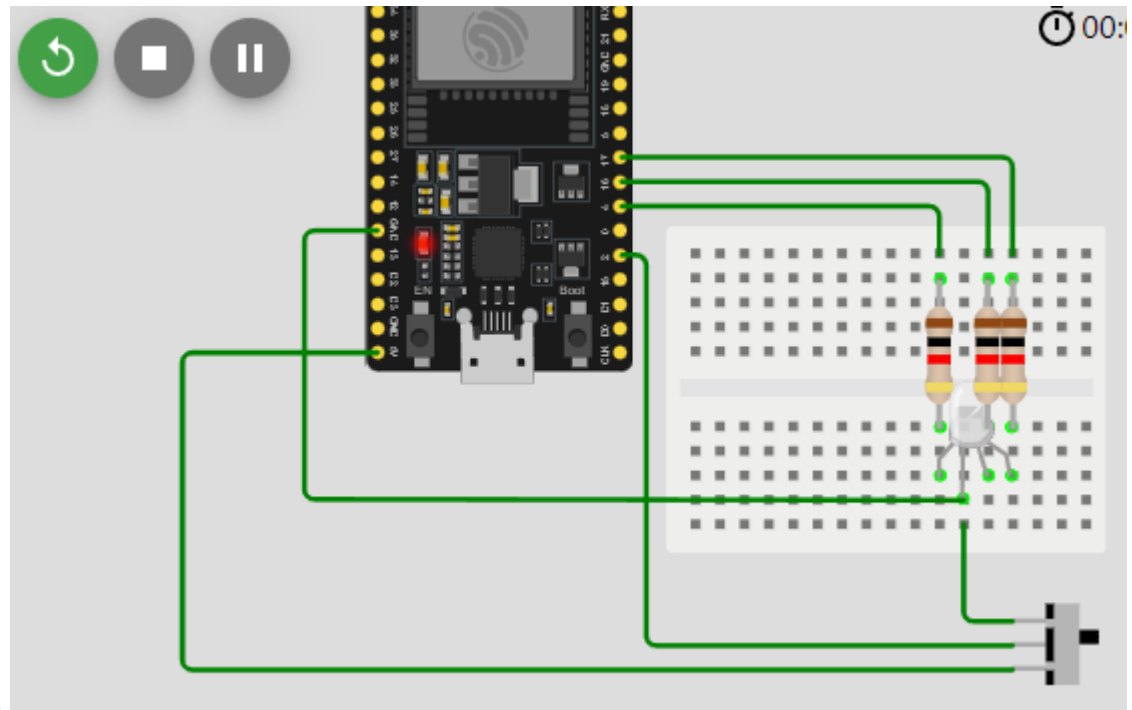
Pulsador 6mm



Interruptor deslizable

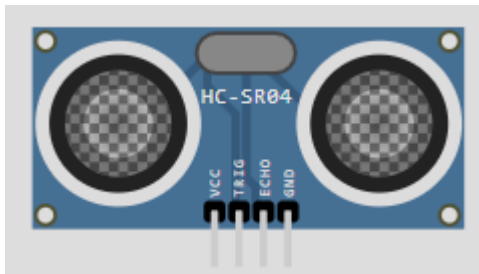
```
int rojo = 4, verde = 16, azul = 17, tiempo = 1000;  
int interruptor = 2;
```

```
void setup() {  
  pinMode(rojo, OUTPUT);  
  pinMode(verde, OUTPUT);  
  pinMode(azul, OUTPUT);  
  pinMode(interruptor, INPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  if(digitalRead(interruptor) == 1){  
    Serial.println("Encendido");  
    digitalWrite(rojo, LOW);  
    digitalWrite(verde, HIGH);  
    delay(tiempo);  
    digitalWrite(verde, LOW);  
    digitalWrite(azul, HIGH);  
    delay(tiempo);  
    digitalWrite(azul, LOW);  
    digitalWrite(rojo, HIGH);  
    delay(tiempo);  
  }else{  
    Serial.println("Apagado");  
  }  
}
```

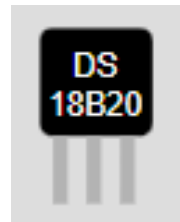


ADC (Analog-to-Digital Converter) de Arduino Uno tiene resolución de 12 bits.

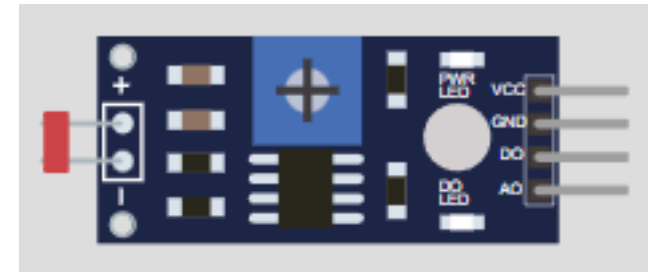
→ Números representables: 000000000000 – 111111111111
0 - 4096



Sensor ultrasónico

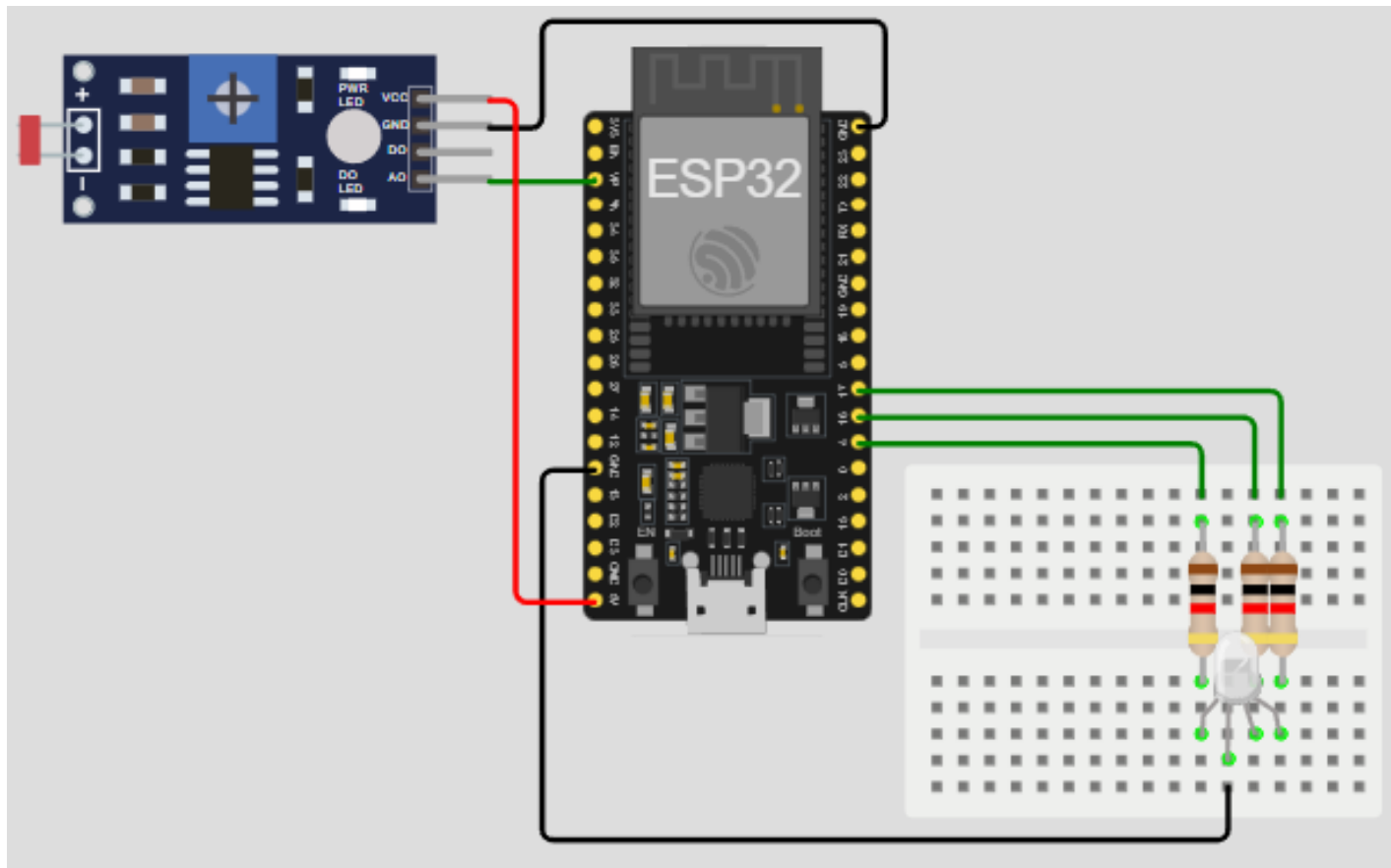


Sensor de
Temperatura



Sensor de luz

- Considere el circuito indicado a continuación, y construya un programa que encienda un LED rojo cada vez que el sensor de luz detecta menos de 100 lux de intensidad lumínica en el ambiente.



<https://wokwi.com/projects/444779232030473217>

```
int rojo = 4, tiempo = 1000;
int sensor = A0, valor;

void setup() {
  pinMode(rojo, OUTPUT);
  pinMode(sensor, INPUT);
  Serial.begin(9600);
}

void loop() {
  valor = analogRead(sensor);
  valor = map(valor, 32, 4063, 100, 0);
  if(valor <= 50){
    digitalWrite(rojo,HIGH);
  }else{
    digitalWrite(rojo, LOW);
  }
  Serial.println(valor);
  delay(tiempo);
}
```

- Es posible utilizar esta tarjeta en modo
 - estación WiFi
 - punto de acceso
- `const char WiFiSSID[] = "nombre_red";`
- `const char WiFiPSK[] = "clave_red";`
- para modo AP:
`const char WiFiAPPSK[] = "clave";`

- Conectar a una red: `WiFi.begin(ssid, password);`
- Chequear estado de conexión: `WiFi.status()`
- Obtener IP asignada: `WiFi.localIP()`

```
WiFiServer server(80);
```

```
server.begin();
```

Chequear si hay algún cliente conectado:

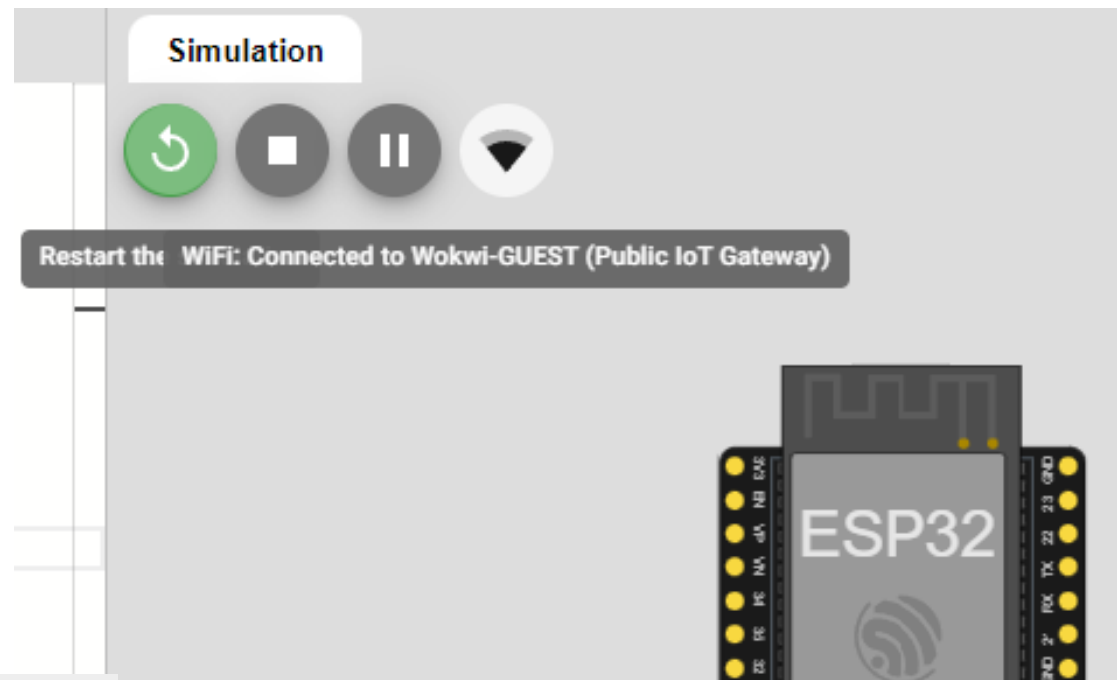
```
void loop() {  
    WiFiClient client = server.available();  
    if (!client){  
        return;  
    }  
    :  
}
```

Leer solicitud (primera línea):

```
void loop() {  
  :  
  String req = client.readStringUntil('\r');  
  Serial.println(req);  
  client.flush();  
  :  
}
```

Modo Access Point (AP)

- `WiFi.mode(WIFI_AP);` en lugar de `WiFi.mode(WIFI_STA);`
- `WiFi.softAP(nombre, clave);`
- esta vez el dispositivo cliente se debe conectar a la red



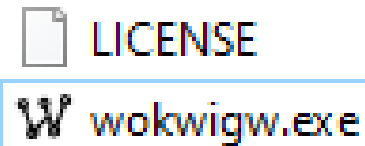
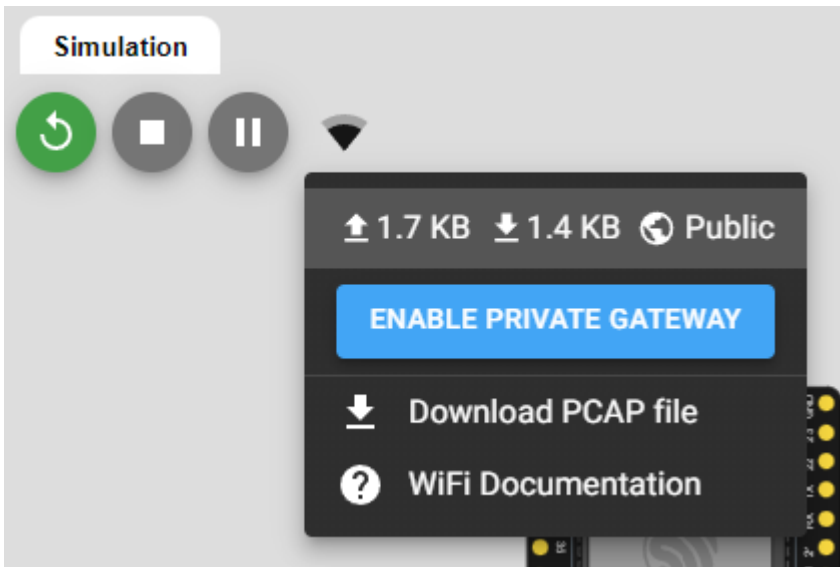
Conectando.....Conectado

3

10.10.0.2

RRSI: -94

Wokwi – Gateway privado



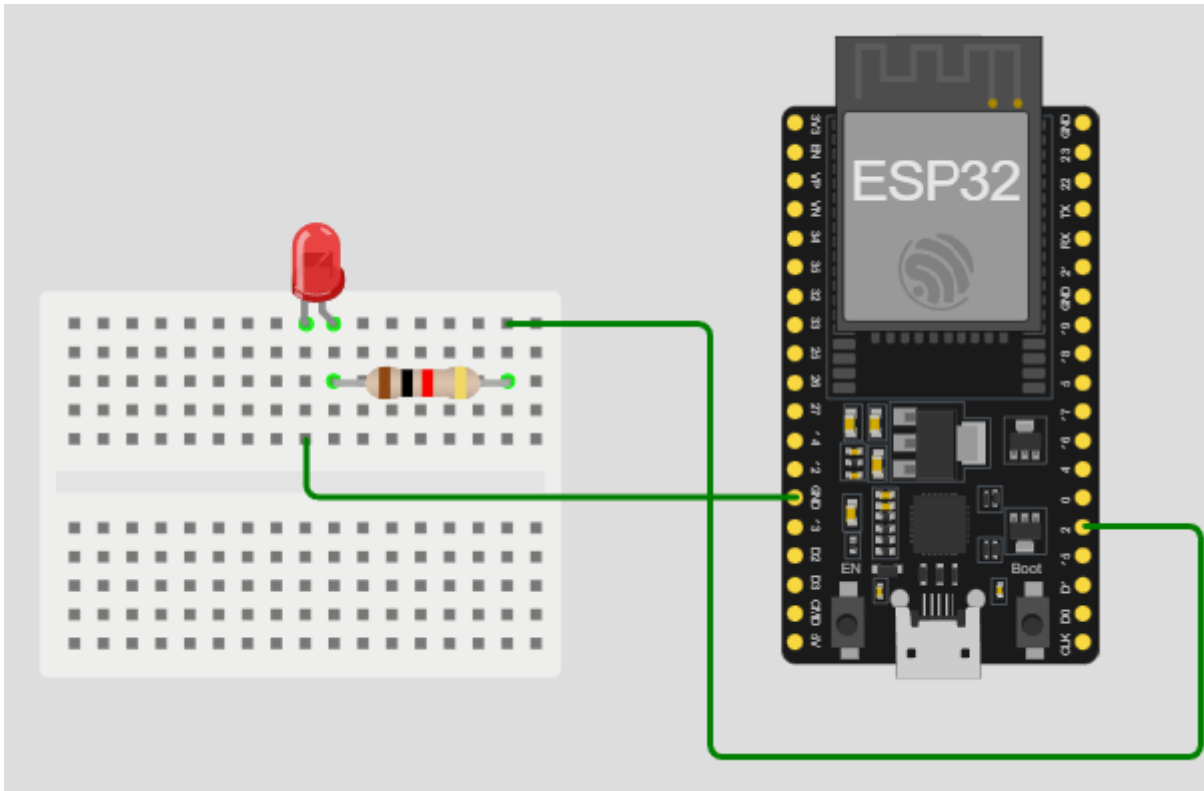
<https://github.com/wokwi/wokwigw/releases/tag/v1.2.0>

```
WOKWI
Wokwi IoT Gateway
Version: v0.1.2
Git revision: 664f0cf291af522dd3acdc7a563628cc68d50189
Built: Mon, 31 Jan 2022 20:05:22 +0000
Listening on TCP port 9011
Client connected: 127.0.0.1:60699
```

10.13.37.2

RRSI: -89

- Considere el circuito indicado a continuación, para atender solicitud de encendido/apagado de LED estando conectado al servidor levantado por la tarjeta ESP32.



<https://wokwi.com/projects/444779488814208001>



Conecte sus datos

Elija la base de datos correcta para almacenar sus datos
[Más información »](#)



Servicios de IA y ML

Integre servicios de IA y ML con su aplicación
[Más información »](#)



Infraestructura como código

Defina su infraestructura en la nube con lenguajes de programación conocidos
[Más información »](#)



Entrega continua

Cree, integre y entregue de forma continua con una canalización de extremo a extremo
[Más información »](#)



Monitoreo y registros

Observación de sus recursos y aplicaciones de AWS
[Más información »](#)



AWS DeepRacer

Adquiera experiencia práctica sobre machine learning con un simulador de carrera en 3D, un coche de carrera completamente autónomo en escala 1/18 y de la Global Racing League.

[Comience a competir »](#)



AWS DeepComposer

Ínciese en la IA generativa mediante la creación de una melodía que se transforma en una canción completamente original en segundos gracias a la IA.

[Comience a componer »](#)



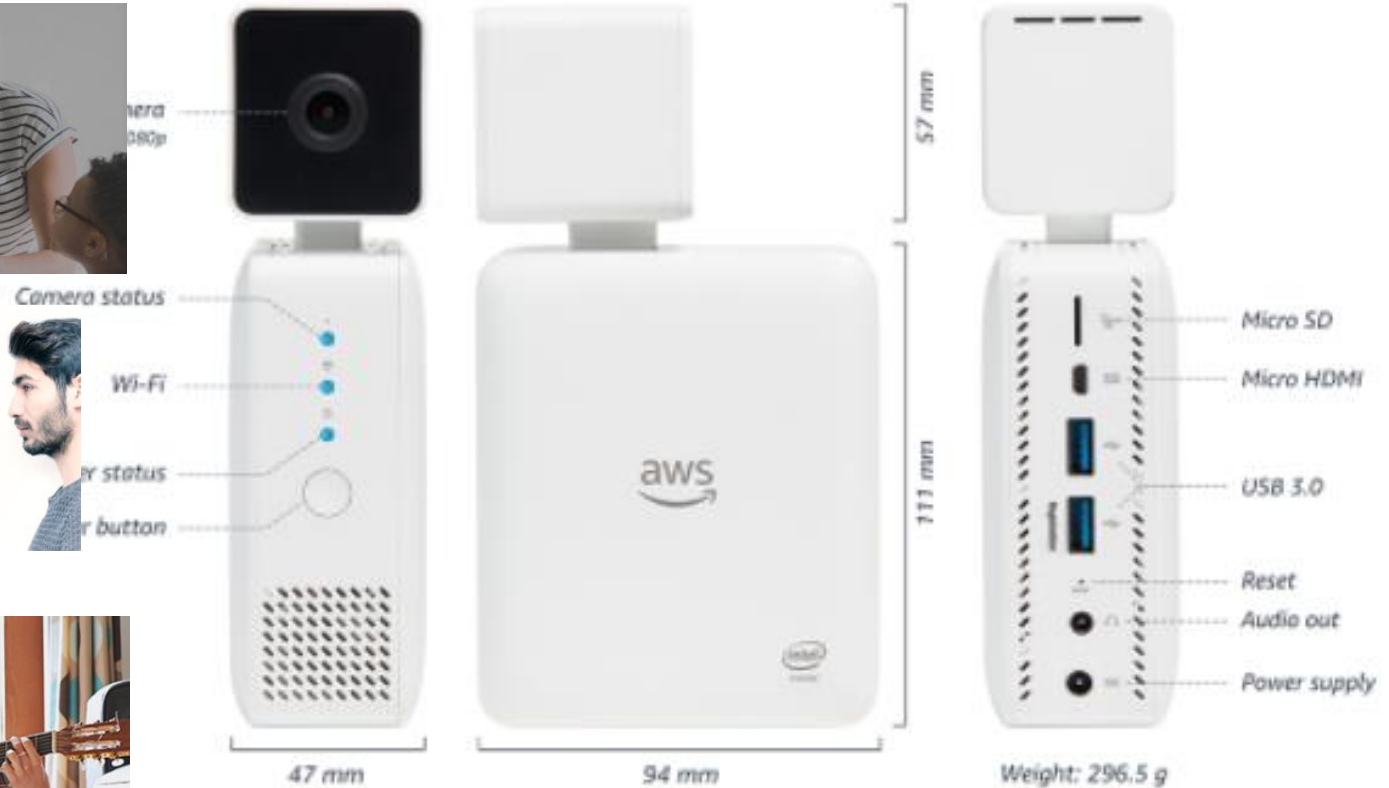
AWS DeepLens

Aprenda los fundamentos básicos del aprendizaje profundo mediante proyectos de visión artificial, tutoriales y exploración práctica del mundo real.

[Comience a crear »](#)



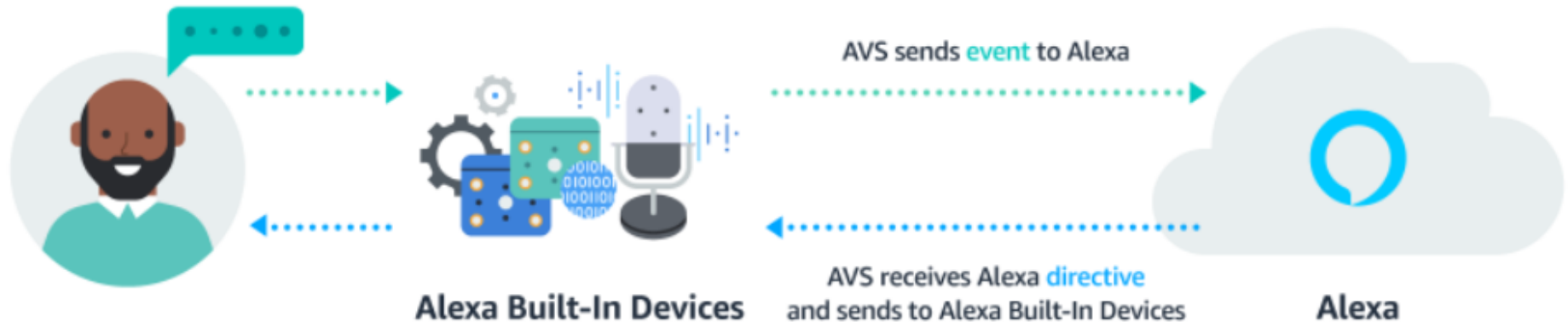
Amazon Web Services



Amazon Echo Dots



Alexa Voice Service (AVS)



Para acceder al AVS API se necesita un token de acceso LWA (Login with Amazon)

Conexión HTTP/2 con Alexa

URLs base

Region	Supported Countries/Regions	URL
Asia	Australia, Japan, New Zealand, India	https://alexa.fe.gateway.devices.a2z.com
Europe	Austria, France, Germany, Italy, Spain, United Kingdom	https://alexa.eu.gateway.devices.a2z.com
North America	Canada, Mexico, United States	https://alexa.na.gateway.devices.a2z.com

Alexa Voice Service (AVS)

- Directivas: mensajes desde AVS al dispositivo para acción específica (p.ej. reproducir audio, poner alarma).
- Eventos: mensajes desde el dispositivo a AVS notificando a Alexa que algo ha ocurrido. El evento más común es solicitud de voz del usuario.

Alerts

AudioActivityTracker

AudioPlayer

Bluetooth

EqualizerController

InputController

Notifications

PlaybackController

Settings

Speaker

SpeechRecognizer

SpeechSynthesizer

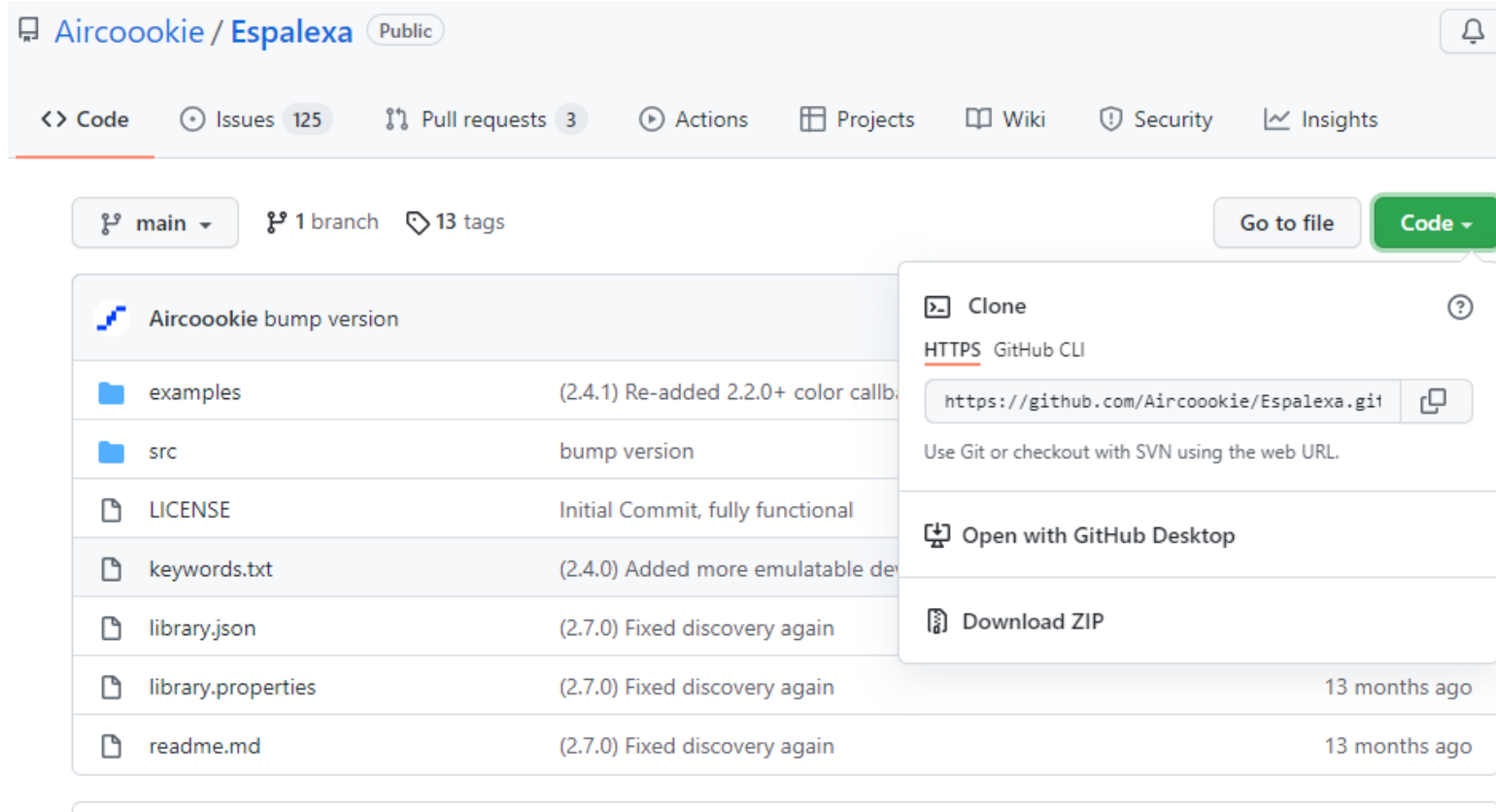
System

TemplateRuntime

VisualActivityTracker

Alexa Voice Service (AVS) con ESP32

Instalar librería ESPAlexa para Arduino IDE:
<https://github.com/Aircoookie/Espalexa>



Aircoookie / Espalexa Public

<> Code Issues 125 Pull requests 3 Actions Projects Wiki Security Insights

main 1 branch 13 tags Go to file Code

Aircoookie bump version

examples	(2.4.1) Re-added 2.2.0+ color callb...	
src	bump version	
LICENSE	Initial Commit, fully functional	
keywords.txt	(2.4.0) Added more emulatable dev	
library.json	(2.7.0) Fixed discovery again	
library.properties	(2.7.0) Fixed discovery again	13 months ago
readme.md	(2.7.0) Fixed discovery again	13 months ago

Clone ?

HTTPS GitHub CLI

`https://github.com/Aircoookie/Espalexa.git`

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

Declaración global de dispositivo:

```
#include <Espalexa.h>

void firstDeviceChanged(uint8_t brightness);

Espalexa espalexa;
```

Definir función que será llamada con el dispositivo:

```
void firstDeviceChanged(uint8_t brightness) {
    //brightness parameter contains the new device state (0:off,255:on,1-254:dimmed)

    //do what you'd like to happen here (e.g. control an LED)
}
```

Agregar dispositivo después de realizar conexión WiFi:

```
espalexa.addDevice("Alexa name of the device", firstDeviceChanged);
```

o

```
EspalexaDevice* d;
```

```
d = new EspalexaDevice("Alexa name of the device", firstDeviceChanged);  
espalexa.addDevice(d);
```

finalmente en setup:

```
espalexa.begin();
```

en loop:

```
espalexa.loop();
```

Librería espalexa usa un servidor web interno, en `http://[dir_ip]/espalexa` se pueden ver todos los dispositivos y su estado actual.



`http://localhost:9080/espalexa`

Hello from Espalexa!

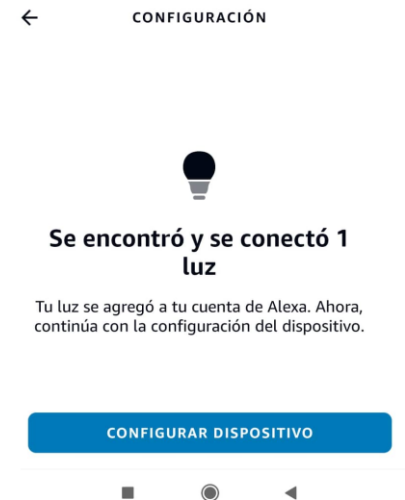
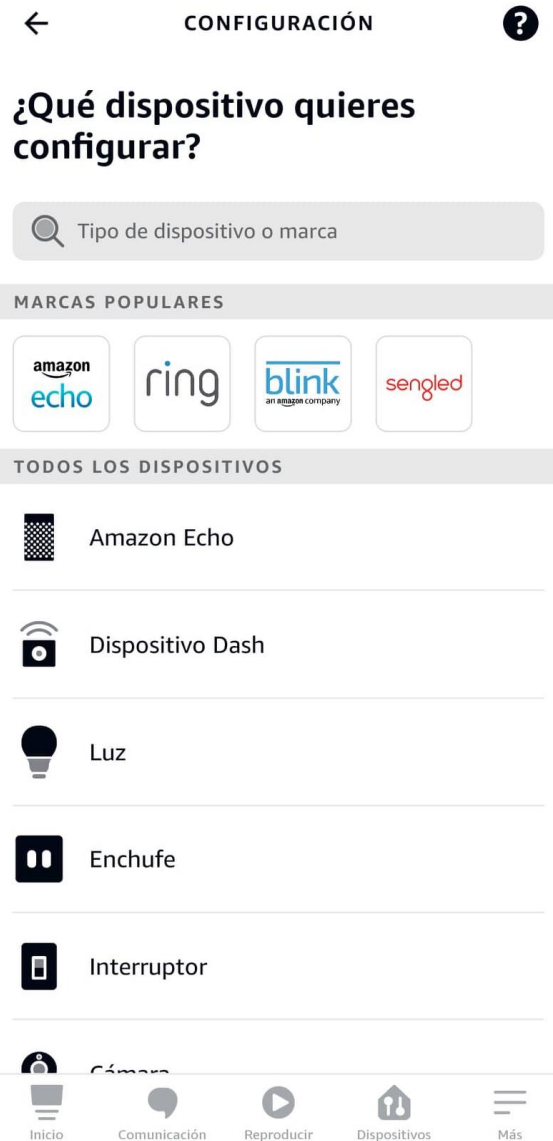
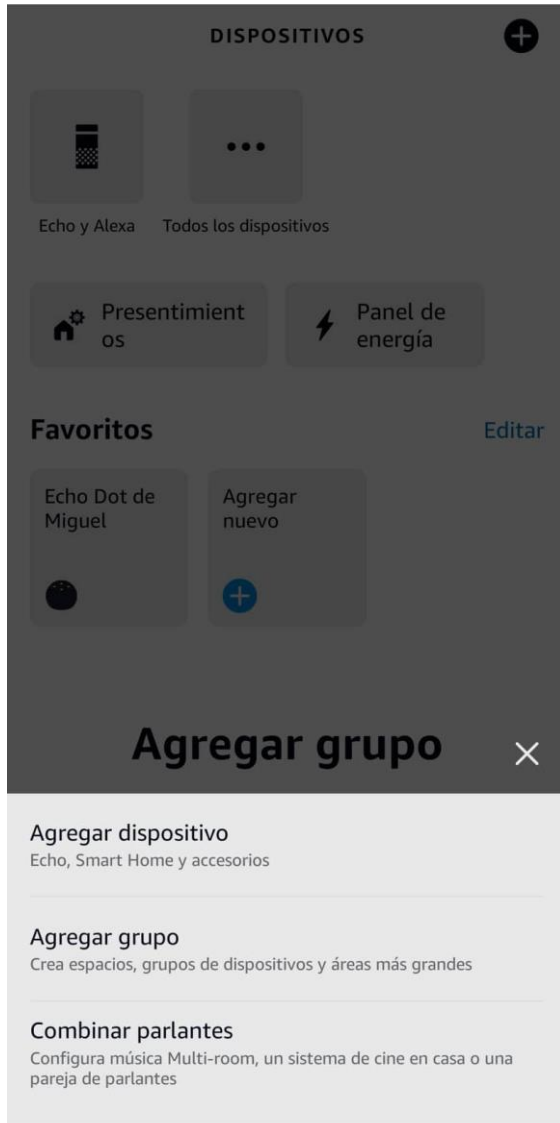
Value of device 1 (lucecita): 0 (Dimmable light)

Free Heap: 238476

Uptime: 15759

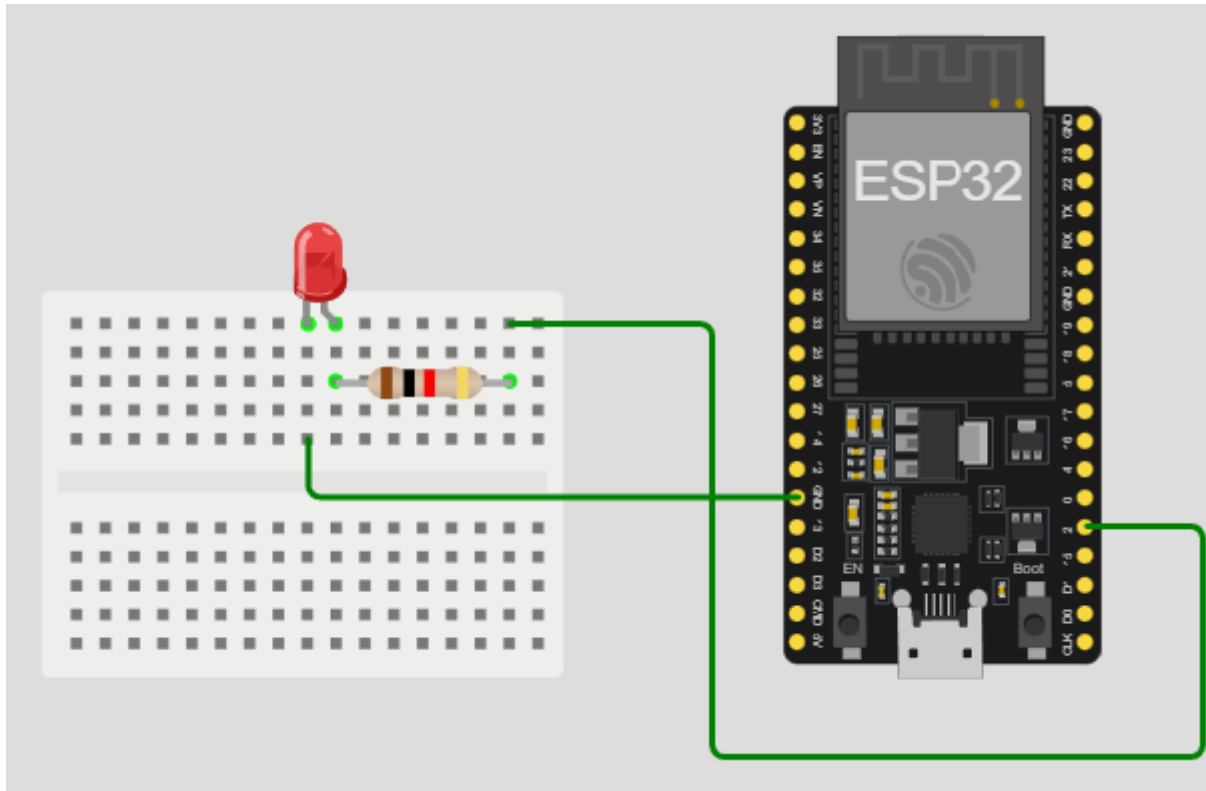
Espalexa library v2.7.0 by Christian Schwinne 2021

Alexa Voice Service (AVS) con ESP32



Actividad (requiere hardware físico)

- Considere el circuito indicado a continuación, para atender solicitud de encendido/apagado de LED usando Alexa (mediante app móvil o mediante echo dot).



<https://wokwi.com/projects/444781320622489601>