

Introducción

Sergio Fco. Hernández Machuca

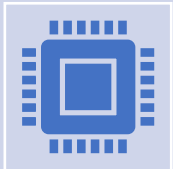
¿Para quién es este curso?

- 1. Personas familiarizadas con Python
- 2. Personas con poca experiencia en programación
- 3. Personas que conocen un dispositivo microcontrolador, el seleccionado para este curso en particular, en cuanto a arquitectura, recursos, aplicación básica
- 4. Personas que hubieran ensayado el desarrollo de diseño de aplicaciones en plataformas como Arduino

¿Qué es Micropython?



Micropython (de aquí en adelante se denominará uP), es un lenguaje de programación basado en la sintaxis, atributos y esquemas desarrollados para Python, en particular la versión 3.4



Es modelado para permitir el uso de Python a personas que quieren escribir programas para dispositivos con recursos limitados (pero no tan limitados como las tarjetas Arduino UNO y similares)

```
83 # To rotate the servo motor to 170 degrees
84 servo(pwm, 170)
85 time.sleep(1)
86
87 while True:
88     # To rotate the servo from 20 to 170 degrees
89     # by 10 degrees increment
90     for i in range(0, 170, 10):
91         print("20 to 170, step ", i)
92         servo(pwm, i)
93         time.sleep(0.5)
94
95     # To rotate the servo from 170 to 20 degrees
96     # by 10 degrees decrement
97     for i in range(170, 0, -10):
98         print("170 to 20, step ", i)
99         servo(pwm, i)
100         time.sleep(0.5)
101
```

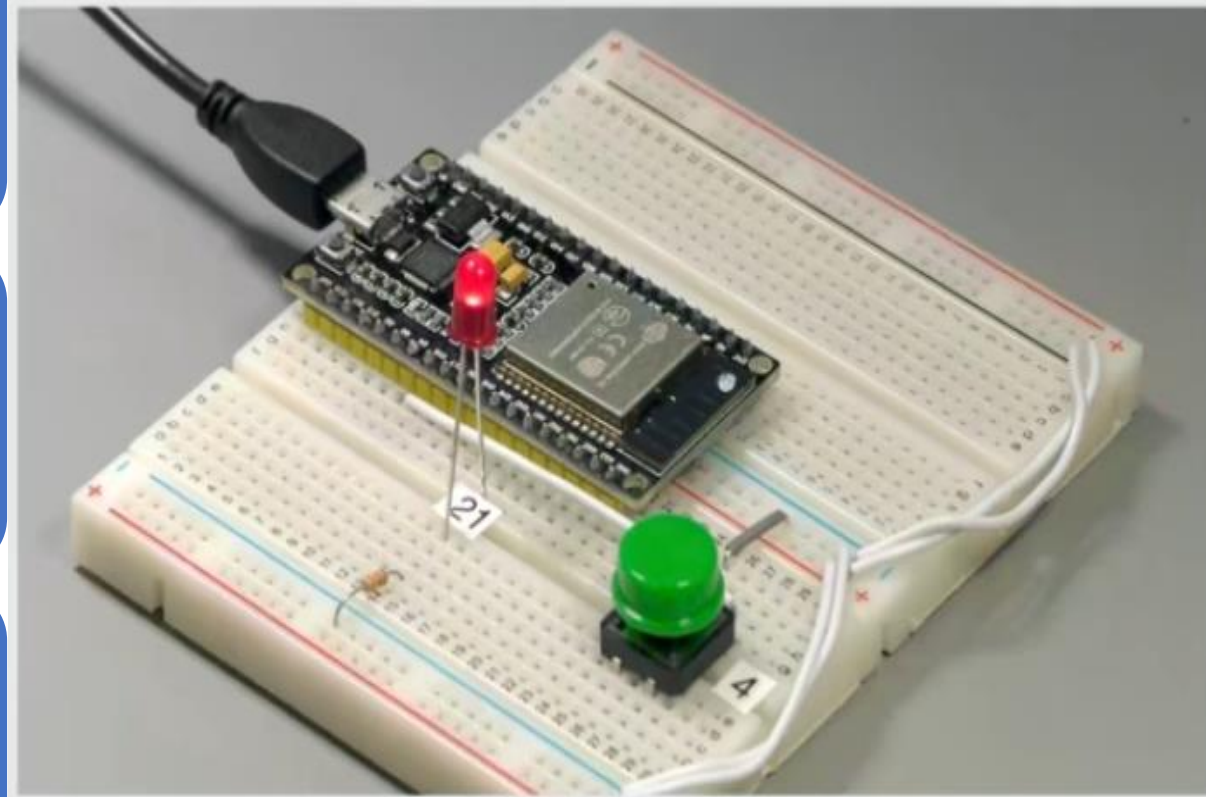


¿Qué es el Microcontrolador ESP32?

Un dispositivo microcontrolador (de aquí en adelante denominado uC) de bajo costo, construido por la compañía Espressif

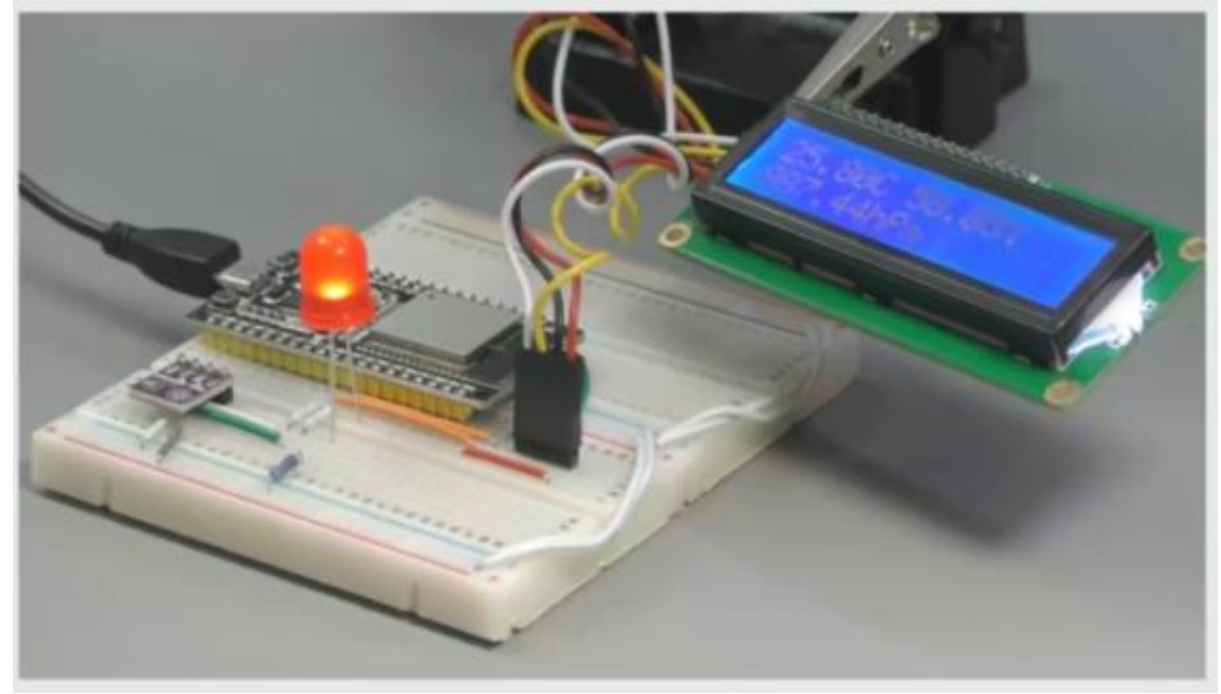
Es fácil de encontrar en diversas versiones, módulos y sistemas, en el mercado nacional

Posiblemente ya hubieran ensayado con este elemento, o estarán próximos a emplearlo en la elaboración de prácticas, laboratorios o proyectos



¿Cuáles son los objetivos del curso?

1. Conocer el lenguaje Python
2. Emplear el lenguaje Micropython (uPython) empleándole en una plataforma basada en un dispositivo uC
3. Elaborar prácticas en donde se ejerciten las características básicas de uPython en el diseño de aplicaciones basadas en un dispositivo uC
4. Implementar laboratorios que resuelvan sistemas electrónicos básicos
5. Desarrollar un proyecto aplicando lo aprendido en este curso, dando solución a un problema real



¿Cuáles son los prerequisites para el curso?

- Entendimiento de los principios de la electrónica digital (sistemas combinacionales, secuenciales, algorítmicos)
- Conocimiento de las herramientas para la construcción de prototipos electrónicos (tableta prototipo, cableado, fuentes de alimentación, circuitos, elementos pasivos)
- Experiencia con el uso de un lenguaje de programación, edición, compilación / interpretación, depuración, ejecución
- Capacidad de uso de una computadora para diversas tareas de búsqueda, edición, compilación, programación de dispositivos, etcétera.

Organización básica del curso - 1

- 1 Introducción. Condiciones del curso
- 2. Microprocesadores y Microcontroladores
- 3. uC ESP32. Características. Arquitectura. Recursos
- 4. uC ESP32. Elementos de E/S. Comunicaciones. Conversores A/D. Conversor D/A, Canales Touch, Otros elementos
- 5. uC ESP32. Sistemas y módulos. Ciclo de diseño.
- 6. uC ESP32. Aplicaciones

Organización básica del curso - 2

- 7. Python básico. Palabras reservadas. Comentarios. Variables. Tipos. Operadores. Cadenas y Formateo. Listas. Tuplas. Rango. Diccionarios. Lazos. Estructuras de Control de Flujo. Funciones. Objetos. Clases. Bibliotecas.
- 8. Herramientas. Visual Studio Code, Thonny
- 9. Micropython en el ESP32. Ensayos
- 10. Comparaciones entre Python y Micropython

Organización básica del curso - 3

- 11. ESP32. Operaciones de Entrada Salida
- 12. ESP32. Sensores
- 13. ESP32. Desplegadores de información. LCD. TFT. OLED. Matrices de LEDs
- 14. ESP32. Motores
- 15. ESP32. Actuadores
- 16.ESP32. Otras características

Organización básica del curso - 4

- 17. Proyecto Integrador. Diseño. Simulación. Desarrollo
- 18. Proyecto Integrador. Demostración de características. Documentación. Evaluación

Requerimientos

- Software
 - Lenguaje Python
 - Firmware de Micropython para la versión del uC seleccionado
 - Una IDE (Medioambiente Integrado de Desarrollo), Visual Studio Code, Thonny
- Hardware (circuitaría)
 - Módulo de desarrollo basado en el uC ESP32
 - Fuente de alimentación
 - Tarjeta prototipo y elementos (resistencias, condensadores, LEDs, interruptores, LCD (20x4), entre otros elementos
 - Elementos para el desarrollo del proyecto integrador

¿Qué es Micropython?

- “... una implementación simple, ágil y eficiente del lenguaje de programación Python3 que incluye un pequeño subconjunto de la librería estándar de Python que ha sido optimizada para ejecutarse en dispositivos microcontroladores en medio ambientes restringidos en cuanto a recursos”



Micropython no es igual a Python

- Python y Micropython son dos lenguajes de programación distintos, que tienen características comunes y comparten elementos similares, que “parecen lo mismo” en cuanto a cómo se escriben y desarrollan.
- Pero... uno está enfocado a desarrollar aplicaciones de escritorio y la Web, para el manejo de grandes conjuntos de datos y el otro debe trabajar en ambientes con poca cantidad de memoria, recursos restringidos y elementos de entrada y salida diversos.



Facilidad de aprendizaje

- Se ha dicho que Python (y Micropython) son fáciles de aprender para empezar a programar computadoras
- Existe mucha documentación de soporte de muchos medios y autores
- Tienen herramientas de desarrollo gratuitas (como los lenguajes mismos) fáciles de emplear
- Se pueden emplear distintos entornos (Windows, Mac, Unix, por un lado, y diversas plataformas de uCs, por el otro)

Características de Micropython

- Empleado para implementar el lenguaje, sintaxis y estándar de la versión 3.4 de Python regular (CPython)
- La librería estándar de uPython implementa un subconjunto de la librería estándar de Cpython
- La opción REPL (read-eval-print-loop) está disponible en uPython
- Se pueden instalar en uPython paquetes de diversos proveedores
- uPython soporta esquema de archivos en el dispositivo uC
- Existen distintos “puertos” para una variedad de uCs, tarjetas y módulos

Micropython en diversos dispositivos

- El lenguaje uPython y sus principales librerías trabajan en distintos ambientes, para varios uCs, tarjetas y módulos
- Debido a la variedad de recursos que tienen diversas plataformas, es común tener que hacer ajustes para la denominación de terminales, recursos, y diversos elementos entre los dispositivos uCs y tarjetas

Micropython es (relativamente) fácil de leer

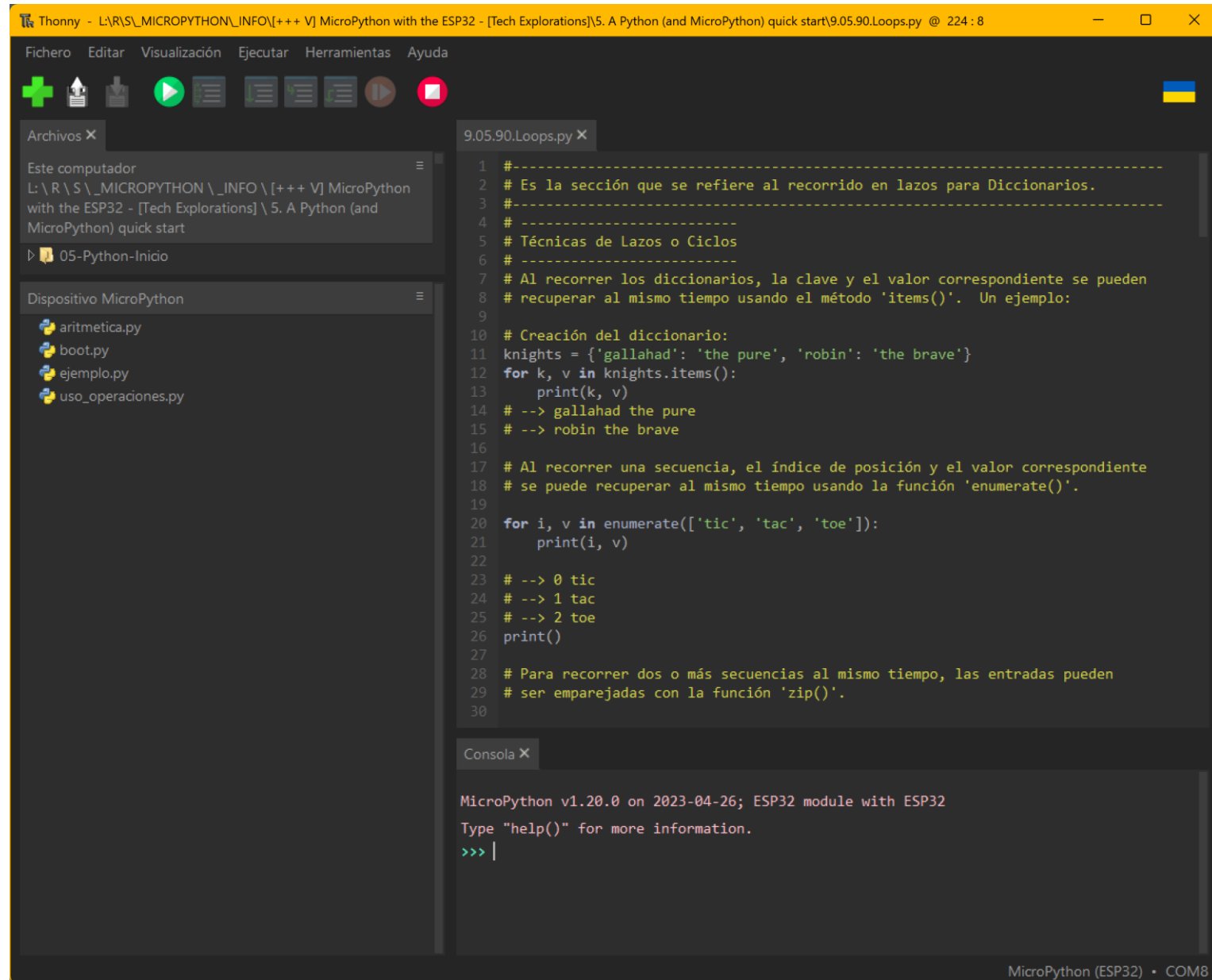
- Como con CPython, uPython es un lenguaje de alto nivel fácil de interpretar y con sintaxis simple
- Aun cuando no se hubiera programado previamente, es simple el entendimiento de los segmentos de código en estos lenguajes

```
led = Pin(21, Pin.OUT)
button_pin4 = Pin(4, Pin.IN, Pin.PULL_UP)

while True:
    if button_pin4.value() == 0:
        led.on()
        sleep(0.1)
    else:
        led.off()
```

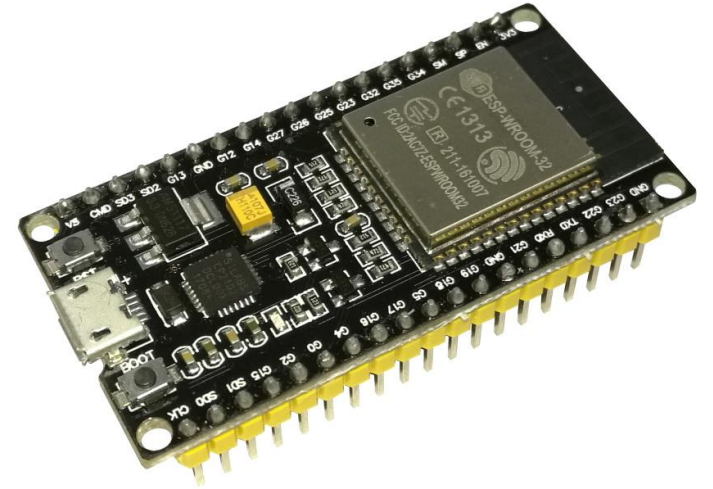
El editor Thonny

- Gratuito
- De código abierto
- Usado para proyectos en Python y Micropython
- Simple y fácil de usar
- Perfecto para quienes inician
- <https://tyhony.org>



Firmware de Micropython para ESP32

- Gratuito
- De código abierto
- Permite que el uC ESP32 ejecute código Micropython desde su REPL o a través de un IDE
- Se encuentra en:
<https://micropython.org/download/esp32/>



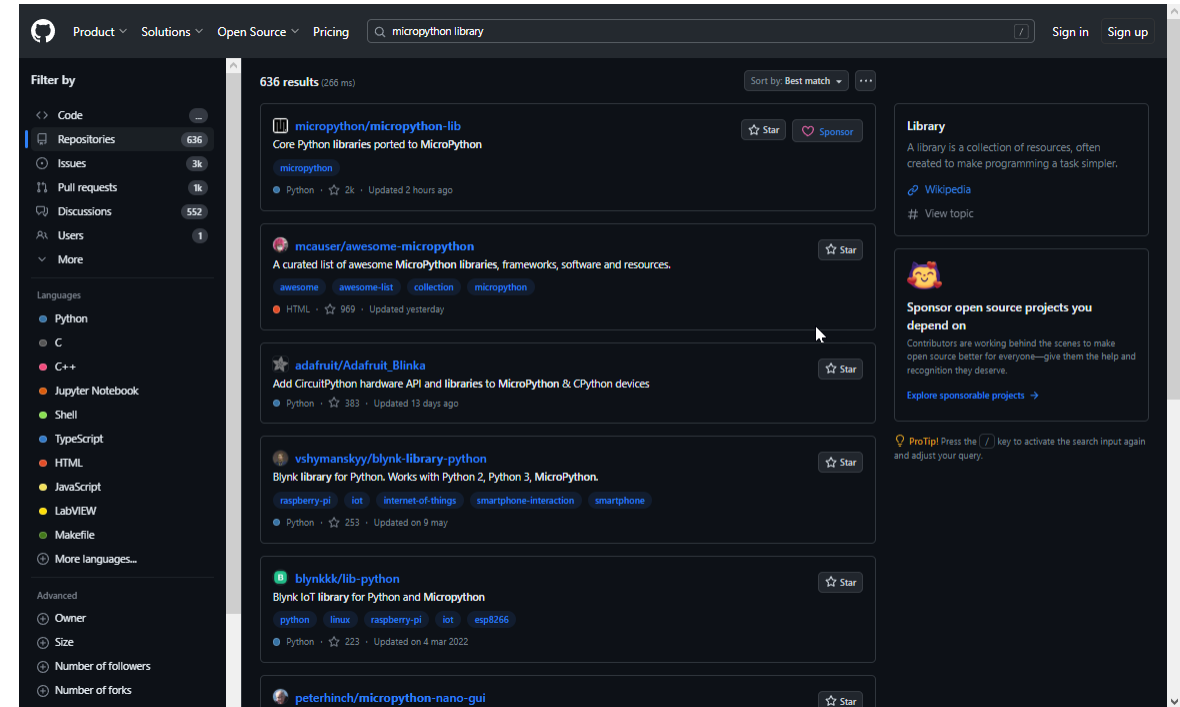
Firmware

Releases

v1.20.0 (2023-04-26) .bin [.elf] [.map] [Release notes] (latest)
v1.19.1 (2022-06-18) .bin [.elf] [.map] [Release notes]
v1.18 (2022-01-17) .bin [.elf] [.map] [Release notes]
v1.17 (2021-09-02) .bin [.elf] [.map] [Release notes]
v1.16 (2021-06-23) .bin [.elf] [.map] [Release notes]
v1.15 (2021-04-18) .bin [.elf] [.map] [Release notes]
v1.14 (2021-02-02) .bin [.elf] [.map] [Release notes]
v1.13 (2020-09-02) .bin [.elf] [.map] [Release notes]
v1.12 (2019-12-20) .bin [.elf] [.map] [Release notes]

Librerías externas de Micropython

- Gratuitas
- De código abierto, muchas de ellas
- Soportan periféricos, como pantallas, sensores, actuadores, motores
- Fáciles de instalar empleando Thonny
- De diversas fuentes



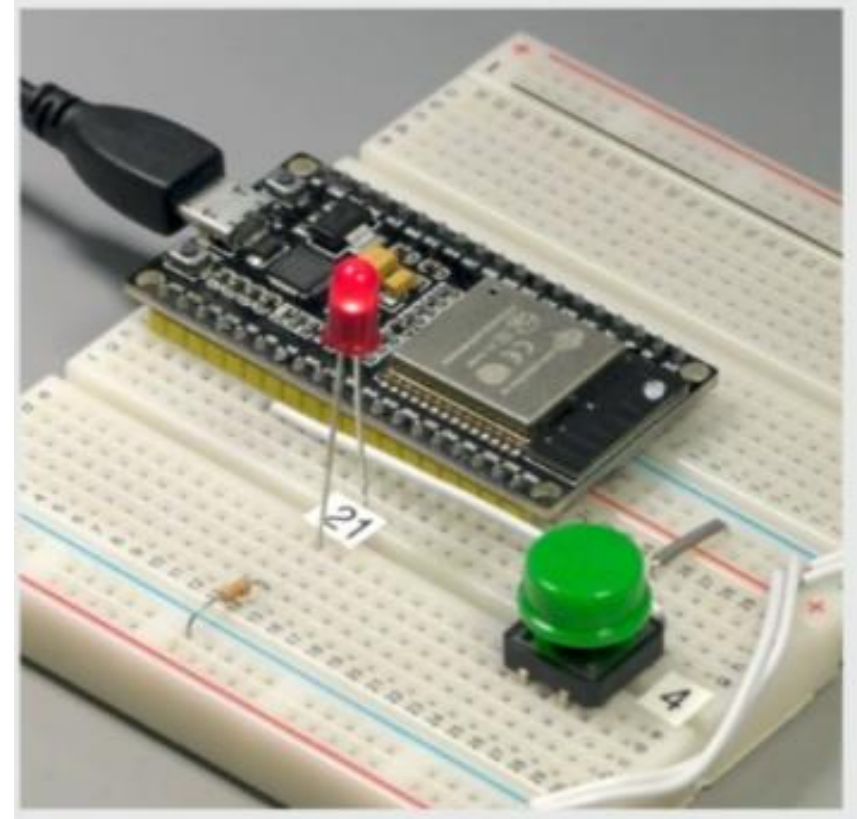
Hardware que se utilizará

- Una tarjeta o módulo ESP32
- Básicamente, cualquier tarjeta ESP32 puede ser útil
- Establecer el balance entre la cantidad de recursos accesibles (pines), elementos interconstruidos (TFT, OLED), esquema de *Reset*, etcétera,
- Verificar que se pueda instalar el firmware
- Ensayar el ciclo de desarrollo para verificar la facilidad de operación



Componentes simples para iniciar

- LEDs
- Interruptores
- Resistencias
- Potenciómetros
- Cables



Sensores

- Para variables físicas de diversa naturaleza
- Algunos ejemplos son:
 - DHT22 (humedad y temperatura)
 - BME280 Presión barométrica
 - HC-SR04 Ultrasonico, para distancia
- También pueden emplearse:
 - De color
 - Aceleración y posición
 - Magnéticos
 - Otros



Desplegadores y pantallas

- Pantalla LCD 20x4 con módulo I2C PCF8574
- Pantalla OLED I2C SSD1306 (0.96")
 - OLED I2C SH1106 (1.3")
 - OLED I2C SSD1315 (0.96")
- Tira de LEDs Neopixel
- Matrices de LEDs MAX7219
- Módulo Teclas y LEDs SPI TM1638
- Otros elementos



Motores

- Motor Servo de 5 volts
- Motor de Corriente Directa 5 volts
- Módulo para control de motores
 - L298n, puente H
 - DRV9971
 - Similar
- Motor de Pasos para 5 volts



Cuestiones generales para el curso

- Teoría, Prácticas, Laboratorios, Proyectos
- Atender la exposición de los conceptos teóricos expuestos
 - Dispositivo. Características. Elementos. Pines.
 - Programa. Lógica. Algoritmo. Código
- Completar cada *práctica* empleando la teoría mostrada
- Desarrollar y presentar los *laboratorios* aplicando lo ensayado en las prácticas
- Construir y presentar el *proyecto* con todo lo expuesto en el curso
- Planificar – Desarrollar – Revisar – Enmendar – Revisar - Presentar