



INGENIERÍA
ELECTRÓNICA
FACULTAD DE INGENIERÍA



RAMA IEEE UNIQUINDIO

Primeros Pasos con MicroPython

Alexander López-Parrado, PhD



Armenia - Quindío
COLOMBIA

Abril 26 - 2024

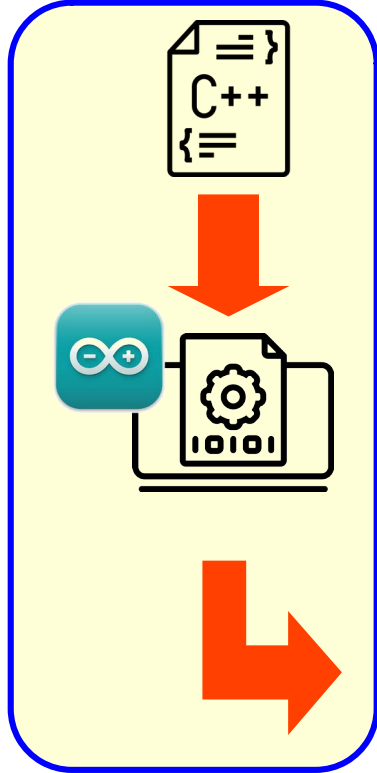
MOTIVACIÓN



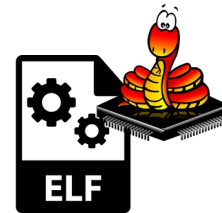
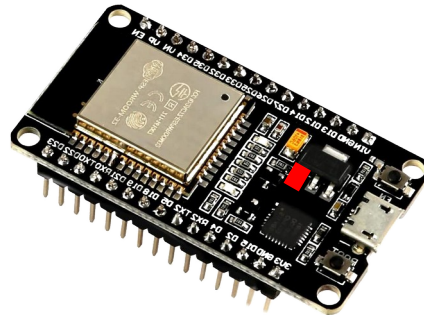


Compilado de tipos estáticos

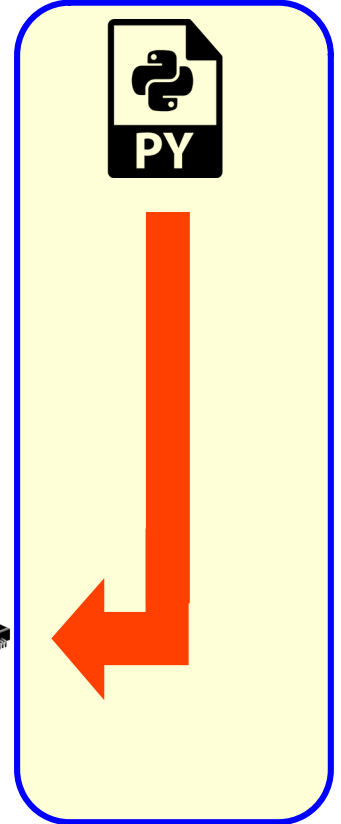
Interpretado de tipos dinámicos



Compilador



Intérprete



“Hola Mundo” en Microcontroladores

```
int LED_BUILTIN = 2;

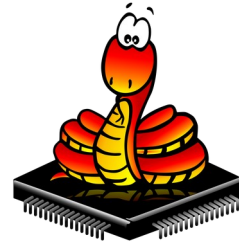
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

```
from machine import Pin
from utime import sleep_ms

ON_BOARD_PIN = 2
led_pin = Pin(ON_BOARD_PIN, Pin.OUT)

while True:
    led_pin.value(not led_pin.value())
    sleep_ms(1000)
```



INTRODUCCIÓN



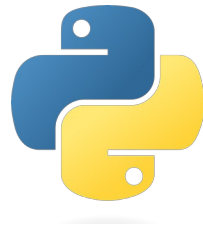
UNIVERSIDAD
DEL QUINDÍO

INGENIERÍA
ELECTRÓNICA
FACULTAD DE INGENIERÍA



UQ
RAMA IEEE UNIQUINDIO

MicroPython

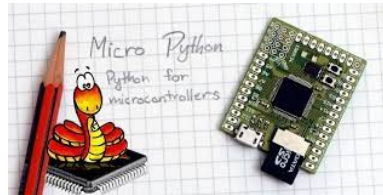


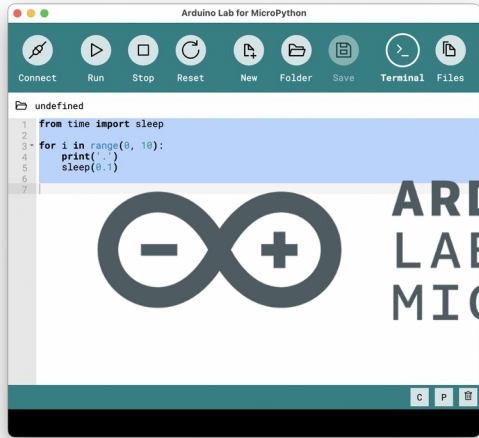
➤ ¿Qué es?

- Implementación básica y eficiente de Python 3
- Incluye un subconjunto pequeño de la biblioteca estándar de Python.
- Optimizado para correr en microcontroladores y entornos restringidos.

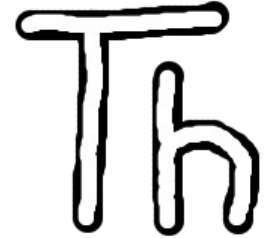
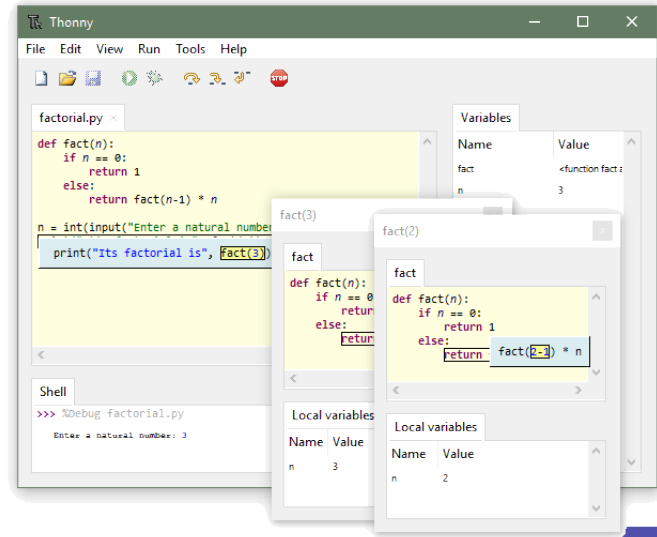
➤ **El firmware (interprete de Python) debe ser almacenado en la memoria flash del microcontrolador.**

➤ **Incluye el REPL (*Read Evaluate Print Loop*) a través de una conexión serial (USB) o WEB.**





ARDUINO[®]
LAB FOR
MICROPYTHON



```
Thonny - MicroPython device :: /boot.py @ 13:1
File Edit View Run Tools Help
[boot.py] x
1 # This file is executed on every boot (including wake-boot from deepsleep)
2 #import esp
3 #esp.osdebug(None)
4 #import webrepl
5 #webrepl.start()
6 from machine import deepsleep
7 from machine import Pin
8 from machine import RTC
9 from time import sleep
10
11 led = Pin (2, Pin.OUT)
12
13 rtc = RTC()
14 rtcData=rtc.memory()
15
16 if len(rtcData)==0:
17
Shell x
load:0x10000000, len:12031
ho 0 tail 12 room 4
load:0x40080400, len:4124
entry 0x40080680
1
Traceback (most recent call last):
  File "boot.py", line 37, in <module>
KeyboardInterrupt:
MicroPython v1.19.1 on 2022-06-18; ESP32 module with ESP32
Type "help()" for more information.

MicroPython v1.19.1 on 2022-06-18; ESP32 module with ESP32
Type "help()" for more information.
>>> |
```

Th

<https://thonny.org/>

ACCESO A PERIFÉRICOS



Lectura de un pin digital

Configuración
de los pines

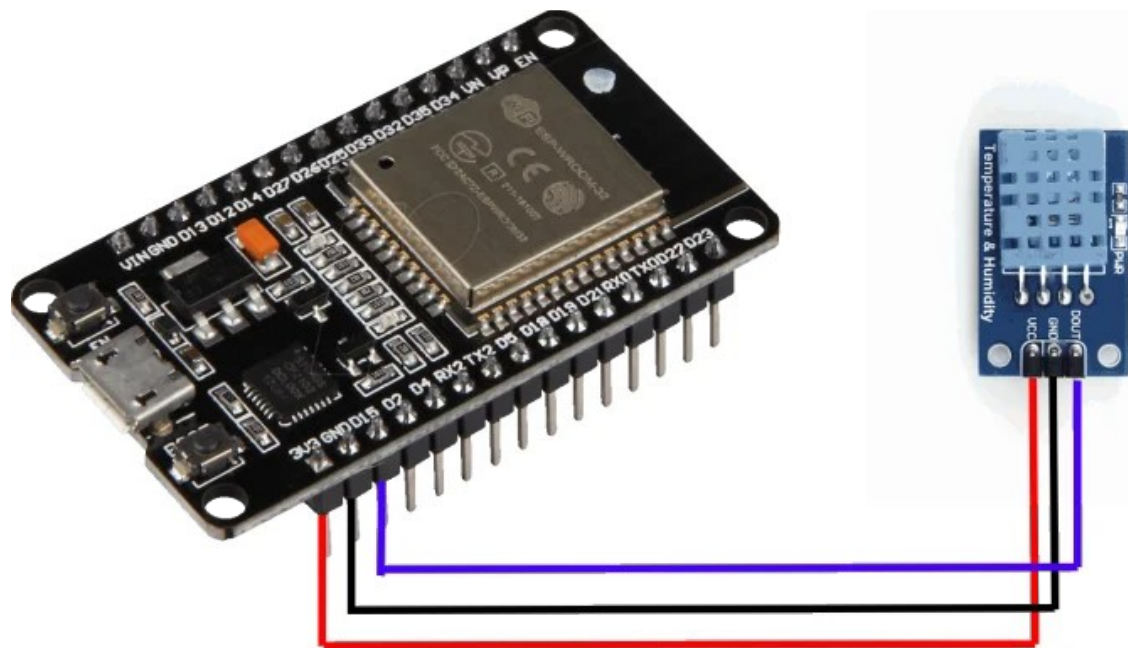
```
from machine import Pin  
from utime import sleep_ms
```

```
LED = 2  
BUTTON=0  
led_pin = Pin(LED, Pin.OUT)  
button_pin = Pin(BUTTON, Pin.IN)
```

Parpadeo más
lectura de
botón en ciclo
infinito

```
while True:  
    print(button_pin.value())  
    led_pin.value(not led_pin.value())  
    sleep_ms(1000)
```

Sensor de Temperatura y Humedad



SIGUIENTES PASOS



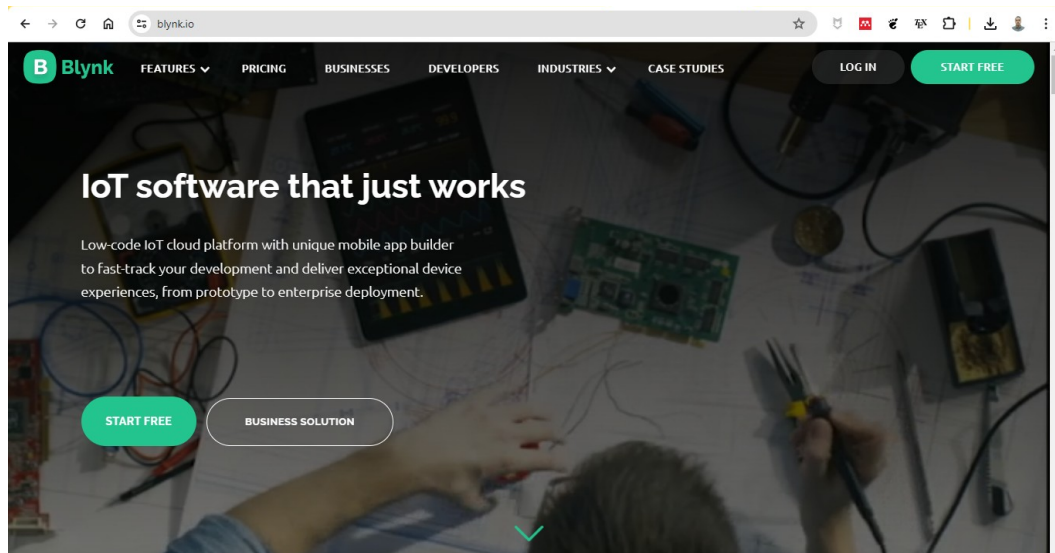
UNIVERSIDAD
DEL QUINDÍO

INGENIERÍA
ELECTRÓNICA
FACULTAD DE INGENIERÍA

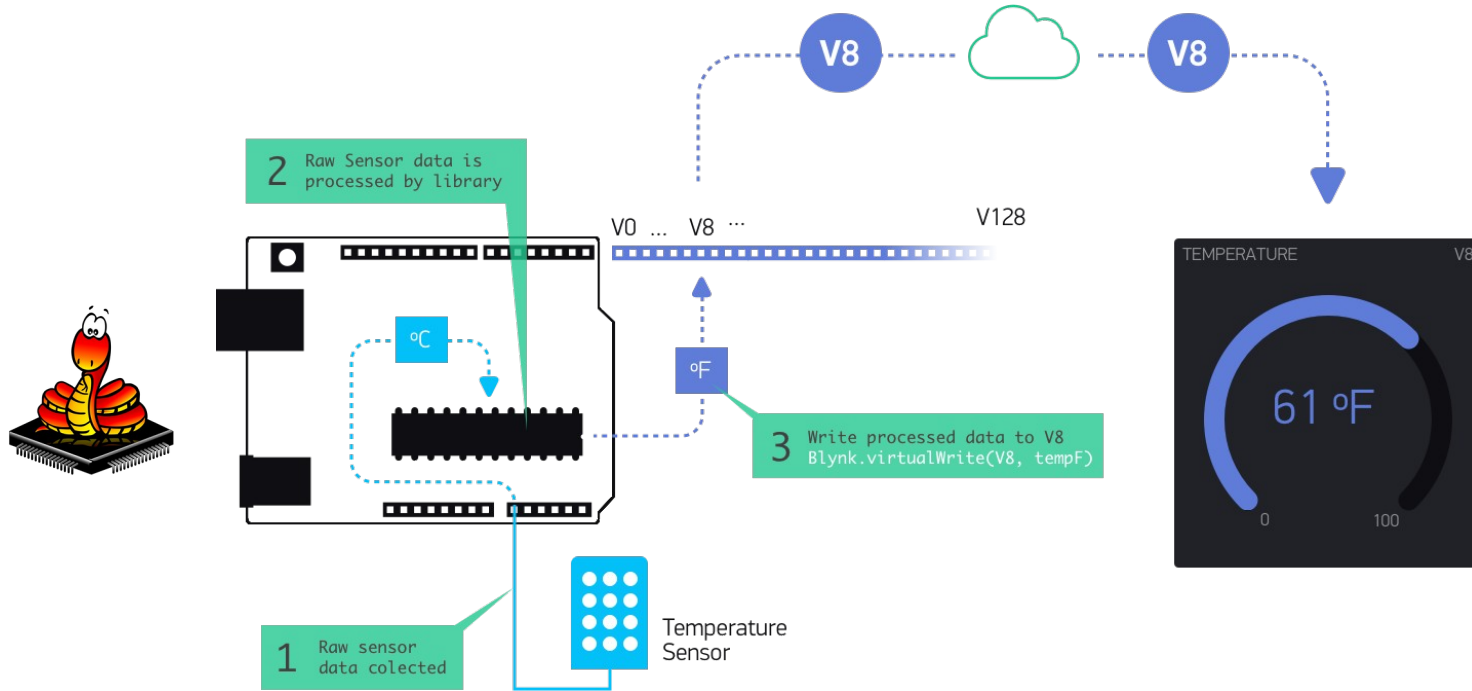


RAMA IEEE UNIQINDÍO

IoT con Blynk



IoT con Blynk



<https://github.com/vshymanskyi/blynk-library-python/tree/master>

Alexander López Parrado
parrado@uniquindio.edu.co

GRACIAS!



Armenia - Quindío
COLOMBIA

Abril 26 - 2024

