

# ESP32 CON C++ Y MICROPYTHON



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

Tobias Briones  
tobias.briones@unah.hn  
Universidad Nacional Autónoma de Honduras  
Ingeniería de Sistemas  
I PAC 2022  
IS911-MICROPROCESADORES

11 de diciembre de 2022

**Resumen**—En este marco teórico se investigan la información de entrada para empezar a emplear la tarjeta microcontrolador ESP32 así como ejemplos de código en C++ Arduino IDE y MicroPython.

## ÍNDICE

<b>I.</b>	<b>Introducción</b>	<b>1</b>
I-A.	Características . . . . .	2
<b>II.</b>	<b>Desarrollo en Arduino IDE y C++</b>	<b>3</b>
II-A.	Programa . . . . .	3
<b>III.</b>	<b>Desarrollo con MicroPython</b>	<b>3</b>
III-A.	Programa . . . . .	3
<b>IV.</b>	<b>Conclusión</b>	<b>4</b>

## I. INTRODUCCIÓN

ESP32 es una tarjeta microcontrolador de bajo costo [1] similar a las demás tarjetas conocidas como Arduino y Raspberry Pi Pico. Esta puede ser programada mediante Arduino IDE al hacer la configuración correspondiente y además mediante MicroPython también. Cuenta con una gran cantidad de características similares a las de este tipo de tarjetas microcontrolador de entrada.

Copyright (c) 2022 Tobias Briones. All rights reserved.  
This work is licensed under the Creative Commons Attribution Share Alike 4.0 International License (CC-BY-SA-4.0).  
Third-party content is available under their respective source, copyright, and/or license.  
For more details go to the [GitHub Repository](#).

La tarjeta ESP32 tiene una gran capacidad de conexión inalámbrica tanto WiFi como Bluetooth, posee grandes integraciones en su diseño, bajo consumo de potencia y un diseño robusto [2]. Por lo tanto, tenemos la siguiente denominación del producto:

Un MCU rico en funciones con Wi-Fi integrado y Conectividad Bluetooth para una amplia gama de aplicaciones.  
Fuente: *ESP32 Wi-Fi & Bluetooth MCU* | Espressif Systems  
[2] (traducido de inglés a español, bajo uso justo)

ESP32 es chip integrado con Wi-Fi y Bluetooth de 2,4GHz con la tecnología TSMC de ultra bajo poder de 40nm, consiguiendo la mejor potencia y rendimiento de conectividad o de radiofrecuencia, así mismo como otros atributos tales como: robustez, versatilidad, y fiabilidad. Diseñado para ser un dispositivo móvil IoT con características específicas con estado del arte en este aspecto. [3].

En la siguiente imagen se ve una tarjeta ESP32 montada en la tarjeta NodeMCU <sup>1</sup>.

<sup>1</sup>Una tarjeta NodeMCU es una plataforma IoT de bajo costo y de código abierto que corre tarjetas ESP como la ESP32 [4]



muchos lenguajes de programación y plataformas, comunidades en línea sobre este dispositivo en específico, lecturas y videos, plataformas de hardware, información sobre proveedores que venden este dispositivo y otros accesorios [5].

## II. DESARROLLO EN ARDUINO IDE Y C++

En esta sección se tratará de abordar como poder crear un programa para la tarjeta ESP32 mediante Arduino IDE y C++. En la siguiente sección, se verá como es el proceso en MicoPython.

Lo que se necesita hacer para establecer esta configuración es lo siguiente [6]:

- Descargar e instalar Arduino IDE.
- Ir a File - Preferences - Additional Boards Manager URLs e ingresar la dirección oficial para instalar la tarjeta ESP32 [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json).
- Ir a Tools - Board - Boards manager.
- Seleccionar la tarjeta ESP32 ya que se ha agregado al repositorio con el paso anterior y proceder a instalarla.
- Seleccionar la tarjeta que se tiene, de entre las tantas opciones.
- Seleccionar el puerto USB ESP32 al conectar el módulo ESP.

Se usará la herramienta esptool para crear el firmware y flashearla en la tarjeta ESP32.

La tarjeta puede estar desacoplada, esto es, solamente el chip, o en una tarjeta NodeMCU como se ilustró arriba. Utilizar Arduino IDE para flashear esta tarjeta es más complicado ya que lleva varios pasos más. Claro está, es Arduino IDE y no ESP32 IDE. Para esto, se tiene que resetear el microcontrolador y empezar lo en modo flash usando GPIO-0 a Tierra. También tener en cuenta sobre el programador que se va a emplear como se mencionó al principio. Por último, para una forma visual y más profunda sobre este proceso se puede consultar [ESP32 - Program a ESP32 | Studio Pieters](#) [6], así como también esta otra guía de mayor calidad [How to Program ESP32 with Arduino IDE?](#) [7].

### II-A. Programa

El siguiente programa puede ser usado para la ESP32. Es lo mismo que el C++ para Arduino, y se explica por si mismo.

```
1 #define ledPin 2
2
3 void setup()
4 {
5   pinMode(ledPin, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(ledPin, HIGH);
11  delay(1000);
12
13  digitalWrite(ledPin, LOW);
14  delay(1000);
```

Listing 1. Programa que hace parpadear el LED en el pin 2

Para este programa, es probable que el LED (SMD) ya se encuentre en la tarjeta, de lo contrario se tendrá que conectar un LED con su resistencia a esta salida. La tarea de flashear la tarjeta con Arduino IDE es un poco complicada, por lo que si no queda claro este aspecto, es muy recomendable visitar los enlaces proveídos para entrar a más detalles.

Para utilizar las entradas discretas, de igual forma, se utiliza “pinMode (ledPin, INPUT);” para emplearla igual que cualquier programa Arduino.

Se recomienda emplear MicroPython que se desarrolla en la siguiente sección ya que funciona mejor para la tarjeta ESP32.

## III. DESARROLLO CON MICROPYTHON

Lo que nos proporciona la documentación de MicroPython para la tarjeta ESP32 es la siguiente información [8]:

- Se obtienen los mejores beneficios al usar MicroPython en ESP32 (recordar que con Arduino IDE no fue tan bonito).
- El software MicroPython soporta ESP32 y hay que tener en cuenta la disposición de los pines, y si incluye un serial USB para tener el UART disponible en el PC.
- Verificar la alimentación de la tarjeta si no tiene USB.
- Se debe descargar el [firmware](#) de MicroPython para cargarlo en la tarjeta ESP32.
- Para cargar el firmware se debe de primero, poner la tarjeta en modo bootloader, y segundo, copiar el firmware. Esto depende de cada tarjeta por lo cual se deberá consultar la documentación correspondiente para mayor detalle.
- Para copiar el firmware, [esptool](#) es soportada por MicroPython y se debe tener instalada (pip install esptool).
- Se borra el flash: `esptool.py -port /dev/ttyUSB0 erase_flash`.
- Y se despliega el firmware: `esptool.py -chip esp32 -port /dev/ttyUSB0 write_flash -z 0x1000 esp32-20180511-v1.9.4.bin`.

### III-A. Programa

Para la creación del programa se recomienda utilizar Thonny IDE. Entonces se deberá de seleccionar en el IDE el intérprete y el puerto del dispositivo. Ahí mismo en la pestaña de Intérprete, se selecciona en la parte de Firmware para flashear el firmware que se descargó. El nombre del archivo debe ser “main.py” ya que se carga “boot.py” de primero y luego “main.py”.

```
1 import machine
2 import sys
3 import utime
4
```

```

5 repl_button = machine.Pin(0, machine.Pin.IN,
    machine.Pin.PULL_UP)
6 led = machine.Pin(5, machine.Pin.OUT)
7
8 while True:
9     if repl_button.value() == 0:
10        print("Dropping to REPL")
11        sys.exit()
12
13 led.value(1)
14 utime.sleep_ms(500)
15 led.value(0)
16 utime.sleep_ms(500)

```

Listing 2. Programa que hace parpadear un LED hasta que se presiona el botón de salir en MicroPython. Fuente: Sparkfun Electronics [9].

También se pudo ver el otro ejemplo más detalladamente: Utilizar PWM para controlar un LED de acuerdo a cuando se presione un botón pulsador.

El código es el siguiente:

```

1 import machine
2 import sys
3 import utime
4
5 repl_button = machine.Pin(0, machine.Pin.IN,
    machine.Pin.PULL_UP)
6 repl_led = machine.Pin(5, machine.Pin.OUT)
7 button = machine.Pin(14, machine.Pin.IN, machine.
    Pin.PULL_UP)
8 pwm_pin = machine.Pin(27, machine.Pin.OUT)
9 pwm = machine.PWM(pwm_pin)
10
11 while True:
12     if repl_button.value() == 0:
13         print("Dropping to REPL")
14         repl_led.value(1)
15         sys.exit()
16
17 for i in range(1024):
18     if button.value() == 0:
19         pwm.duty(i)
20         utime.sleep_ms(2)
21     else:
22         pwm.duty(0)

```

Listing 3. Programa que utiliza PWM en MicroPython para ESP32. Fuente: Sparkfun Electronics [9].

El diagrama es el siguiente:

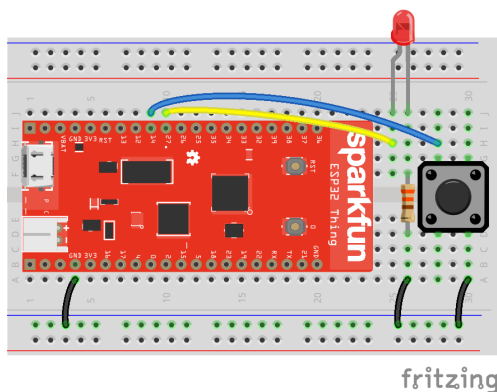


Figura 5. Circuito para programa PWM con MicroPython Fuente: Sparkfun | MicroPython Programming Tutorial: Getting Started with the ESP32 Thing [9] (under the CC BY-SA 4.0 license)

Y se ve algo así:

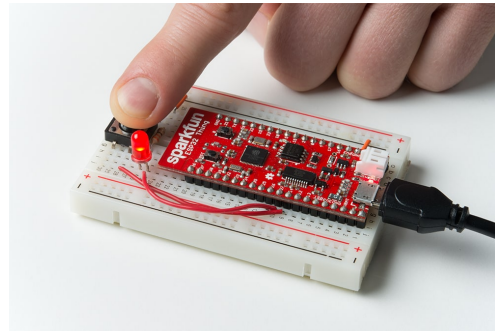


Figura 6. Resultado para programa PWM con MicroPython Fuente: Sparkfun | MicroPython Programming Tutorial: Getting Started with the ESP32 Thing [9] (under the CC BY-SA 4.0 license)

## IV. CONCLUSIÓN

Se explicó brevemente las características y pasos principales de la tarjeta microcontrolador ESP32 para poder emplear ya sea Arduino IDE o MicroPython para el desarrollo de aplicaciones para ESP32. Se vio el uso de entradas y salidas digitales tanto en C++ Arduino y en MicroPython, adicionando un proyecto PWM para MicroPython. Por último, se recomendó emplear MicroPython para un mejor uso de la tarjeta ESP32.

## REFERENCIAS

- [1] W. contributors. «ESP32». inglés. (26 de mar. de 2022), dirección: <https://en.wikipedia.org/wiki/ESP32> (visitado 03-04-2022).
- [2] L. ESPRESSIF SYSTEMS (SHANGHAI) CO. «ESP32 Wi-Fi Bluetooth MCU I Espressif Systems». inglés americano. (2022), dirección: <https://www.espressif.com/en/products/socs/esp32> (visitado 03-04-2022).
- [3] —, «esp32\_datasheet\_en.pdf». inglés americano. (mar. de 2022), dirección: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf) (visitado 03-04-2022).
- [4] W. contributors. «NodeMCU». inglés. (16 de feb. de 2022), dirección: <https://en.wikipedia.org/wiki/NodeMCU> (visitado 03-04-2022).
- [5] esp32.net. «The Internet of Things with ESP32». inglés. (), dirección: <http://esp32.net> (visitado 03-04-2022).
- [6] A. . Pieters. «ESP32 – Program a ESP32». inglés americano. (14 de mar. de 2022), dirección: <https://www.studiopieters.nl/esp32-program-a-esp32> (visitado 03-04-2022).
- [7] R. . Teja. «How to Program ESP32 with Arduino IDE?». Inglés americano. (7 de sep. de 2021), dirección: <https://www.electronicshub.org/esp32-arduino-ide> (visitado 03-04-2022).
- [8] MicroPython. «1. Getting started with MicroPython on the ESP32 — MicroPython 1.18 documentation». inglés. (2022), dirección: <https://docs.micropython.org/en/latest/esp32/tutorial/intro.html> (visitado 03-04-2022).

- [9] Hymel. «MicroPython Programming Tutorial: Getting Started with the ESP32 Thing - learn.sparkfun.com». inglés. (), dirección: <https://learn.sparkfun.com/tutorials/micropython-programming-tutorial-getting-started-with-the-esp32-thing/all> (visitado 03-04-2022).