UNIVERSIDAD
DE LIMA

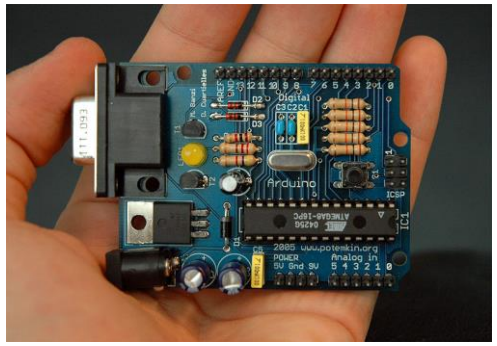
FACULTAD DE INGENIERÍA Y ARQUITECTURA
CARRERA DE INGENIERÍA DE SISTEMAS

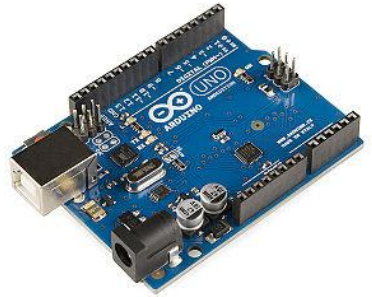
MICROCONTROLADORES

UNIVERSIDAD DE LIMA

Open hardware básico: arduino

- Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar
- Tomar información del entorno a través de sus pines de entrada de toda una gama de sensores
- Las placas pueden ser hechas a mano o compradas montadas de fábrica
- El software puede ser descargado de forma gratuita.

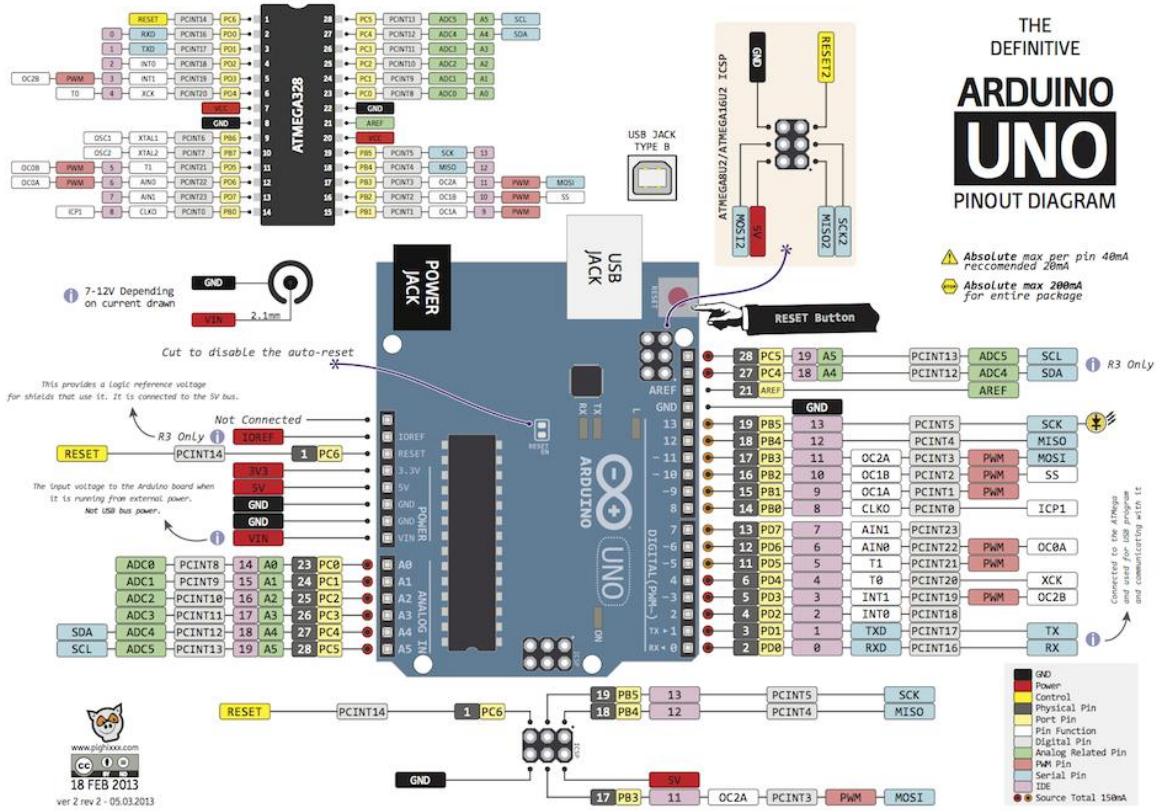




UNIVERSIDAD DE LIMA

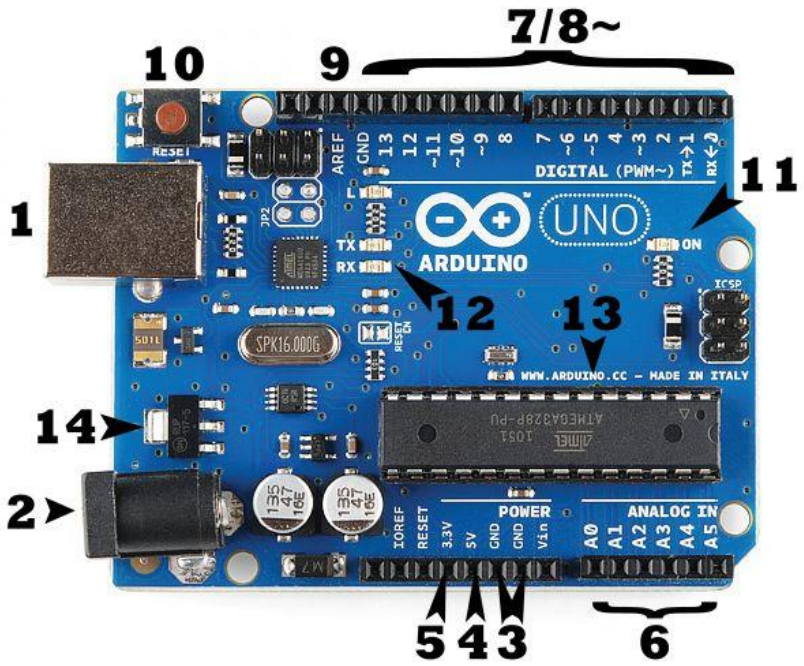
Introducción al arduino

- Partes



UNIVERSIDAD DE LIMA

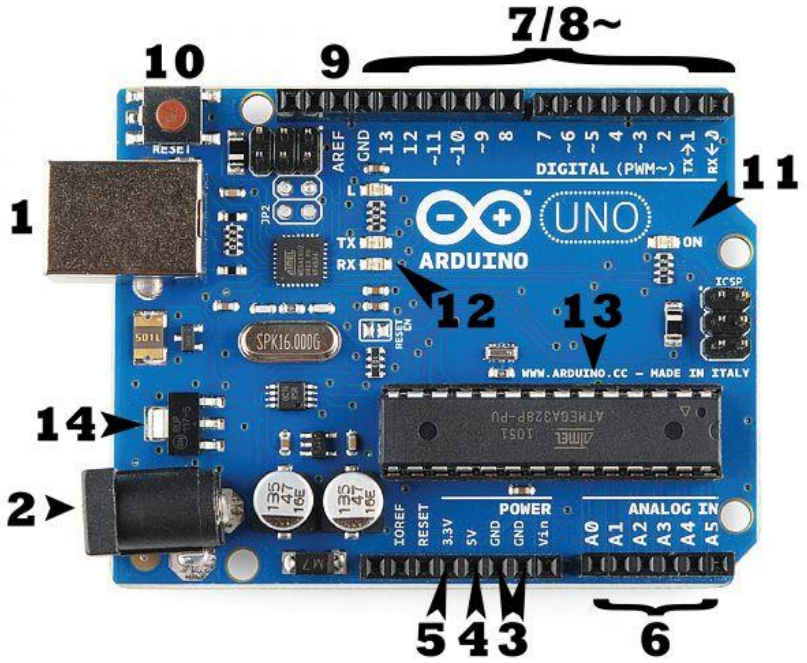
Componentes del arduino



1. USB
2. Alimentación (AC)
3. Tierra GND
4. 5V (DC)
5. 3.3V (DC)
6. Entradas analógicas
7. Entradas digitales
8. Entradas PWM
9. Referencia analógica
10. Botón de reinicio
11. Led encendido
12. Led transmisión data
13. Microcontrolador
14. Regulador de voltaje

UNIVERSIDAD DE LIMA

Señales analógicas y digitales



6 Entradas analógicas

7 Entradas digitales

8 Entradas PWM

UNIVERSIDAD DE LIMA

El ESP32

- El ESP32 es una serie de microcontroladores System on a Chip (SoC) de bajo costo y bajo consumo
- Desarrollados por Espressif
- Incluyen capacidades inalámbricas Wi-Fi y Bluetooth
- Posee un procesador de doble núcleo.

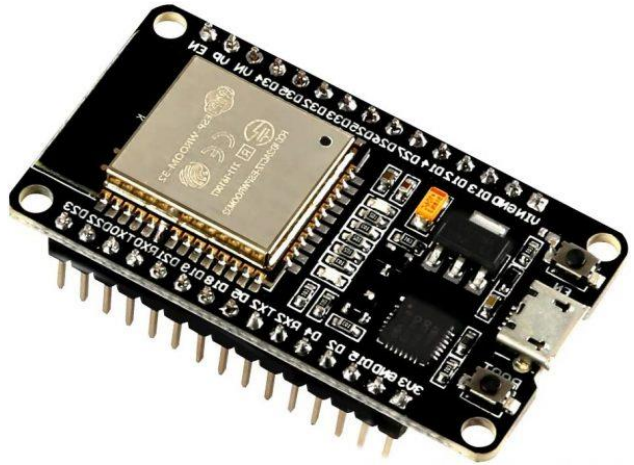
ESPRESSIF

UNIVERSIDAD DE LIMA

Características ESP32

- **Bajo costo**: El costo aproximado en el mercado peruano es de S/. 60
- **Bajo consumo**: el consumo en funcionamiento normal es de 50mA y de 1mA en sleep mode;
- **Conectividad Wi-Fi**: el ESP32 puede conectarse fácilmente a una red Wi-Fi para conectarse a Internet (modo estación) o crear su propia red inalámbrica Wi-Fi (modo punto de acceso);
- **Bluetooth**: utiliza el BLE (bluetooth low energy);
- **Doble núcleo**: vienen con 2 microprocesadores Xtensa LX6 de 32 bits: núcleo 0 y núcleo 1.
- **Interfaces de entrada/salida periféricas**: pines táctiles capacitivos, ADC, DAC, UART, SPI, I2C. , PWM.
- **Compatible con el "lenguaje de programación" Arduino**
- **Compatible con MicroPython**

Especificaciones técnicas ESP32



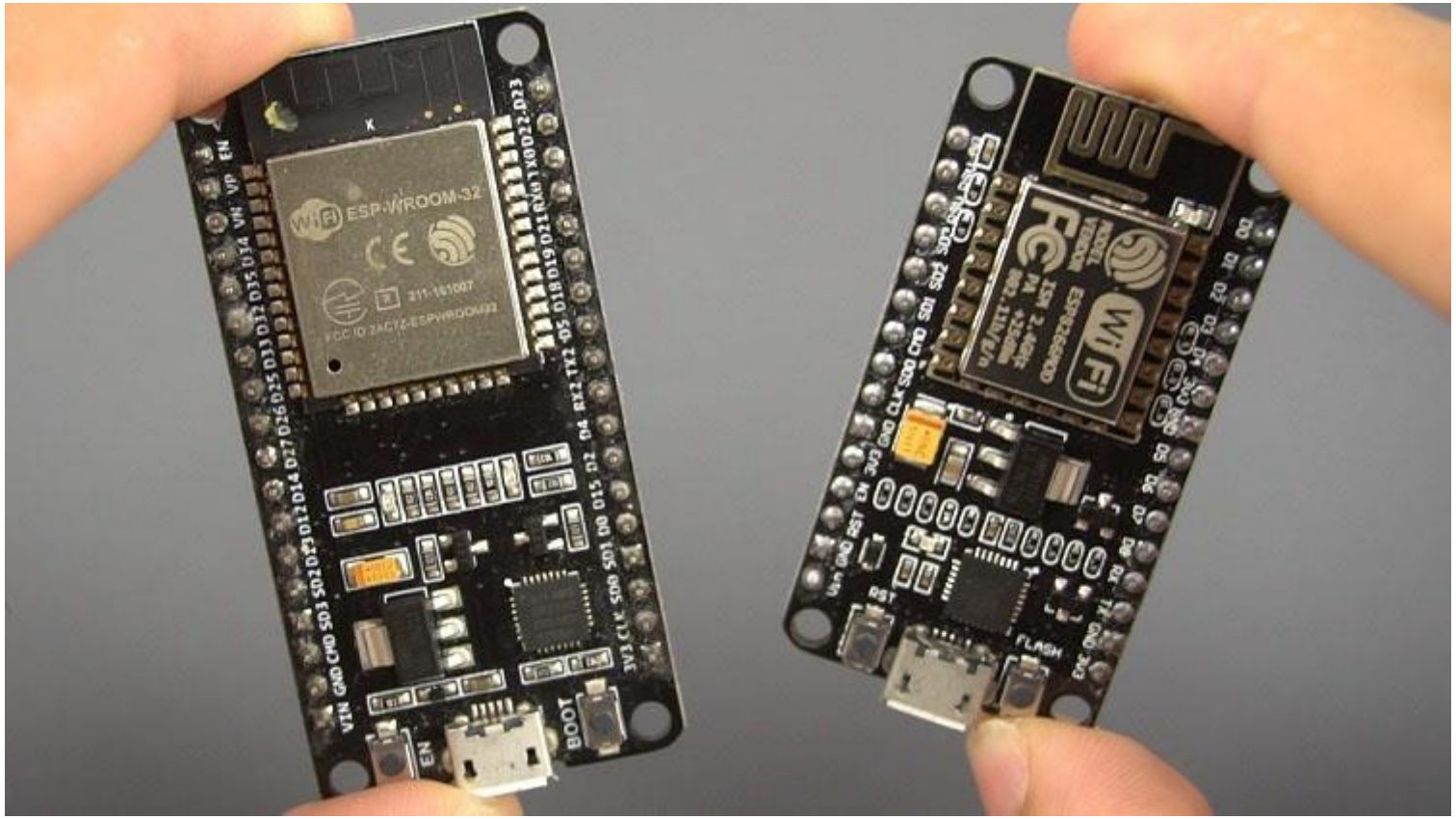
- Procesador Tensilica Xtensa 32bits LX6 hasta 240MHz.
- Wi-Fi: 802.11b/g/n/e/i (802.11n @ 2.4 Ghz hasta 150 Mbit/s).
- Bluetooth: v4.2 BR/EDR y bluetooth Low Energy (BLE).
- Rom:448 KiB.
- SRAM: 520 KiB.
- RTC slow SRAM: 8 KiB.
- RTC fast SRAM: 8 KiB.
- eFuse: 1 Kbit.
- Flash embebida: 0 MiB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips); 2 MiB (ESP32-D2WD chip); 4 MiB (ESP32-PICO-D4 SIP module).
- Periféricos compatibles: ADC, DAC, I2C, UART, Interfaz CAN 2.0, SPI, I2S, RMII y PWM entre otros.
- Seguridad tipo IEEE 802.11, WFA, WPA/WPA2 y WAPI.
- Encriptación de memoria Flash.
- Criptografía soportada por acelerador de hardware: AES, SHA-2, RSA, ECC, RNG.
- Voltaje de trabajo 3.3VDC.
- Energía y datos via conector microUSB 5VDC.

UNIVERSIDAD DE LIMA

Diferencias entre el ESP32 y ESP8266

- El ESP32 es más rápido que el ESP8266
- El ESP32 viene con más GPIO con múltiples funciones
- El ESP32 admite mediciones analógicas en 18 canales (pines habilitados para analógicos) en comparación con solo un pin ADC de 10 bits en el ESP8266
- El ESP32 admite Bluetooth mientras que el ESP8266 no
- El ESP32 es de doble núcleo (la mayoría de los modelos) y el ESP8266 es de un solo núcleo
- El ESP32 es un poco más caro que el ESP8266.

Diferencias entre el ESP32 y ESP8266



UNIVERSIDAD DE LIMA

Tipos de ESP32

- ESP32 se refiere al chip ESP32 integrado.
- El término "ESP32" también se utiliza para referirse a las placas de desarrollo ESP32.
- Usar chips básicos ESP32 no es fácil ni práctico, especialmente cuando se aprende, se prueba y se crean prototipos.

UNIVERSIDAD DE LIMA

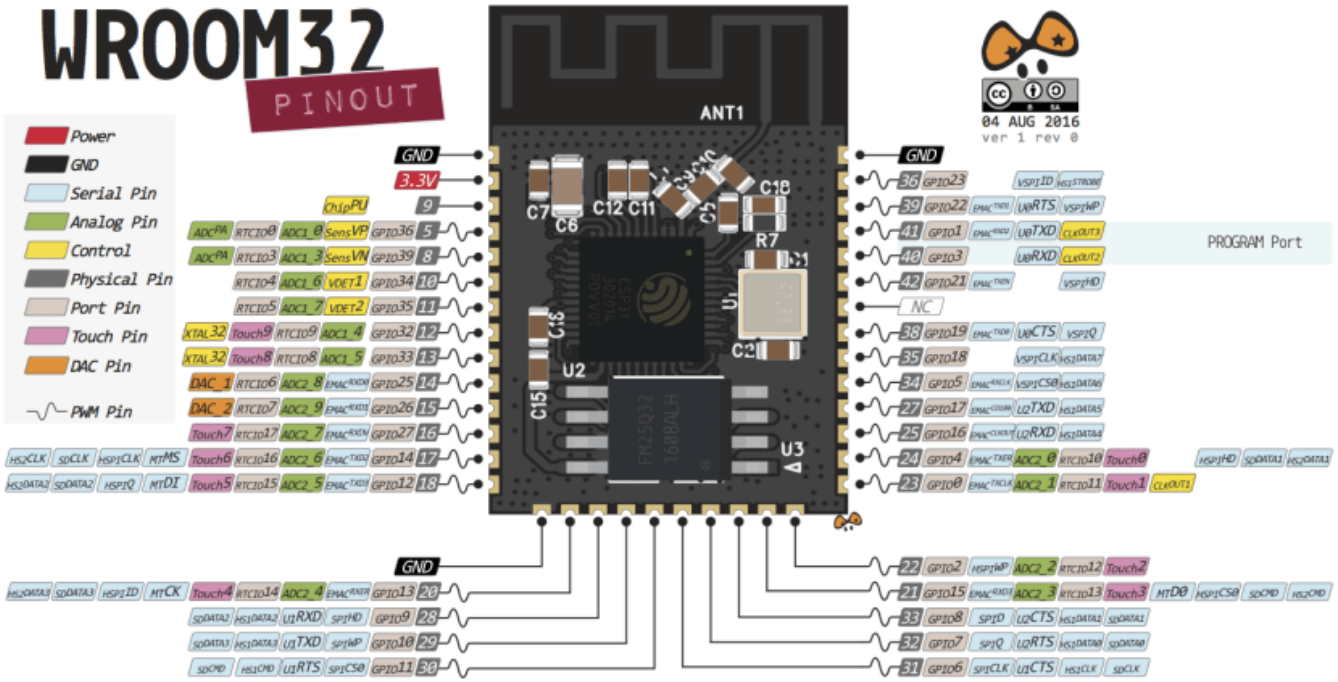
ESP32 Devkit V1

Number of cores	2 (dual core)
Wi-Fi	2.4 GHz up to 150 Mbits/s
Bluetooth	BLE (Bluetooth Low Energy) and legacy Bluetooth
Architecture	32 bits
Clock frequency	Up to 240 MHz
RAM	512 KB
Pins	30, 36, or 38 (depending on the model)
Peripherals	Capacitive touch, ADC (analog to digital converter), DAC (digital to analog converter), I2C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Networkr), SPI (Serial Peripheral Interface), I2S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.
Built-in buttons	RESET and BOOT buttons
Built-in LEDs	built-in blue LED connected to GPIO2; built-in red LED that shows the board is being powered
USB to UART bridge	CP2102

UNIVERSIDAD DE LIMA

PINOUT ESP32

- El circuito ESP32 es un chipset único, a partir de este se desarrollan distintos modelos de tarjetas



UNIVERSIDAD DE LIMA

PINOUT DEVKIT V1 38 PINES

- El modelo usado en la asignatura tiene 38 pines

UNIVERSIDAD DE LIMA

PINES ESPECIALES

- Los pines 6 7 8 9 10 y 11 están conectados a la memoria flash

- NO SE DEBEN DE USAR ESTOS PINES

UNIVERSIDAD DE LIMA

PINES ESPECIALES

- Algunos pines tienen una función única al iniciar el ESP32. Estos se llaman Strapping Pins.

Voltage of Internal LDO (VDD_SDIO)			
Pin	Default	3.3 V	1.8 V
MTDI	Pull-down	0	1
Booting Mode			
Pin	Default	SPI Boot	Download Boot
GPIO0	Pull-up	1	0
GPIO2	Pull-down	Don't-care	0
Enabling/Disabling Debugging Log Print over U0TXD During Booting			
Pin	Default	U0TXD Active	U0TXD Silent
MTDO	Pull-up	1	0
Timing of SDIO Slave			
Pin	Default	FE Sampling FE Output	FE Sampling RE Output	RE Sampling FE Output	RE Sampling RE Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Los Strapping Pins son **GPIO0, GPIO2, GPIO4, GPIO5 GPIO12 y GPIO15.**
Se pueden usar, pero se debe tener cuidado al configurar un estado lógico (3,3 V o 0 V) con una resistencia pull-up o pull-down externa

UNIVERSIDAD DE LIMA

PINES ESPECIALES

Voltage of Internal LDO (VDD_SDIO)			
Pin	Default	3.3 V	1.8 V
MTDI	Pull-down	0	1
Booting Mode			
Pin	Default	SPI Boot	Download Boot
GPIO0	Pull-up	1	0
GPIO2	Pull-down	Don't-care	0
Enabling/Disabling Debugging Log Print over U0TXD During Booting			
Pin	Default	U0TXD Active	U0TXD Silent
MTDO	Pull-up	1	0

Timing of SDIO Slave					
Pin	Default	FE Sampling FE Output	FE Sampling RE Output	RE Sampling FE Output	RE Sampling RE Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

GPIO 0 (debe estar LOW para ingresar al modo de boot)
GPIO 2 (o BAJO durante el boot)
GPIO 4
GPIO 5 (debe estar HIGH durante el boot)
GPIO 12 (debe estar LOW durante el boot)
GPIO 15 (debe estar HIGH durante el boot)

UNIVERSIDAD DE LIMA

PINES SOLO DE ENTRADA

- Los pines 34 a 39 son GPI: pines de solo entrada. Estos pines no tienen resistencias internas pull-up o pull-down.
- **No se pueden usar como salidas**

UNIVERSIDAD DE LIMA

PINES CAPACITIVOS

- El ESP32 tiene 10 sensores táctiles capacitivos internos.
- Estos pueden detectar variaciones en cualquier cosa que tenga carga eléctrica, como la piel humana.
- Pueden detectar variaciones inducidas al tocar los GPIO con un dedo

- **T0 (GPIO 4)**
- **T1 (GPIO 0)**
- **T2 (GPIO 2)**
- **T3 (GPIO 15)**
- **T4 (GPIO 13)**

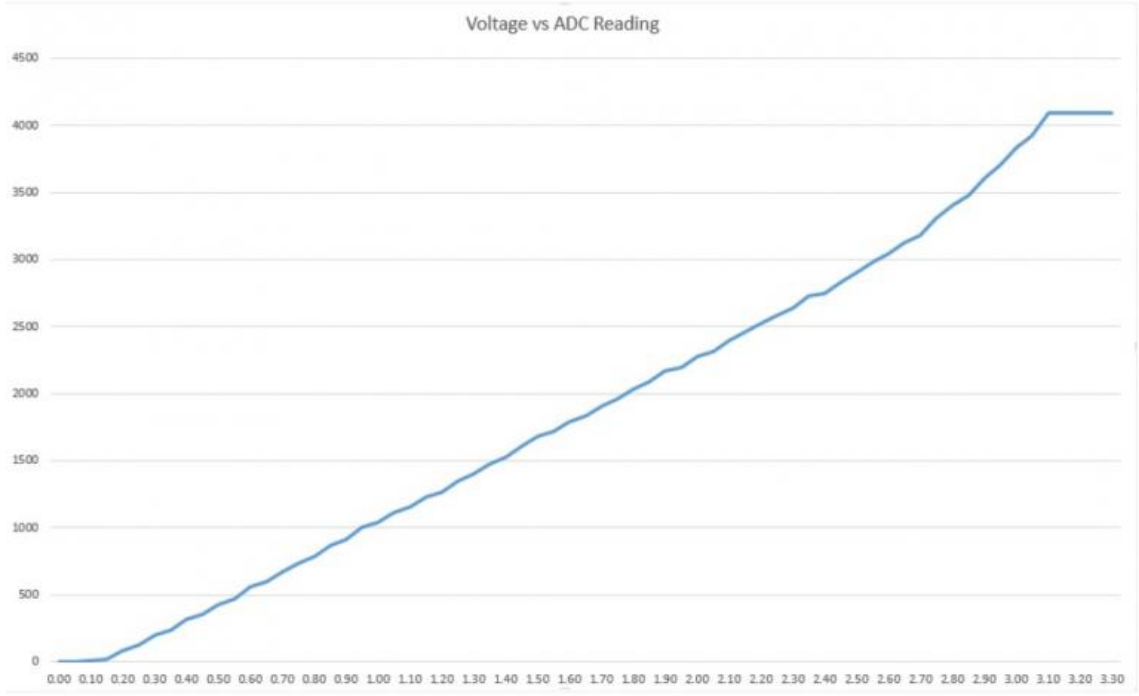
- **T5 (GPIO 12)**
- **T6 (GPIO 14)**
- **T7 (GPIO 27)**
- **T8 (GPIO 33)**
- **T9 (GPIO 32)**

CONVERSORES ANALOGICOS-DIGITALES ADC

- El ESP32 tiene canales de entrada ADC de 18 x 12 bits (mientras que el ESP8266 solo tiene 1 ADC de 10 bits).
- Estos son los GPIO que se pueden utilizar como ADC y sus respectivos canales:

- ADC1_CH0 (GPIO 36)
- ADC1_CH1 (GPIO 37)
- ADC1_CH2 (GPIO 38)
- ADC1_CH3 (GPIO 39)
- ADC1_CH4 (GPIO 32)
- ADC1_CH5 (GPIO 33)
- ADC1_CH6 (GPIO 34)
- ADC1_CH7 (GPIO 35)

- ADC2_CH0 (GPIO 4)
- ADC2_CH1 (GPIO 0)
- ADC2_CH2 (GPIO 2)
- ADC2_CH3 (GPIO 15)
- ADC2_CH4 (GPIO 13)
- ADC2_CH5 (GPIO 12)
- ADC2_CH6 (GPIO 14)
- ADC2_CH7 (GPIO 27)
- ADC2_CH8 (GPIO 25)
- ADC2_CH9 (GPIO 26)



UNIVERSIDAD DE LIMA

CONVERSORES ANALOGICOS-DIGITALES ADC

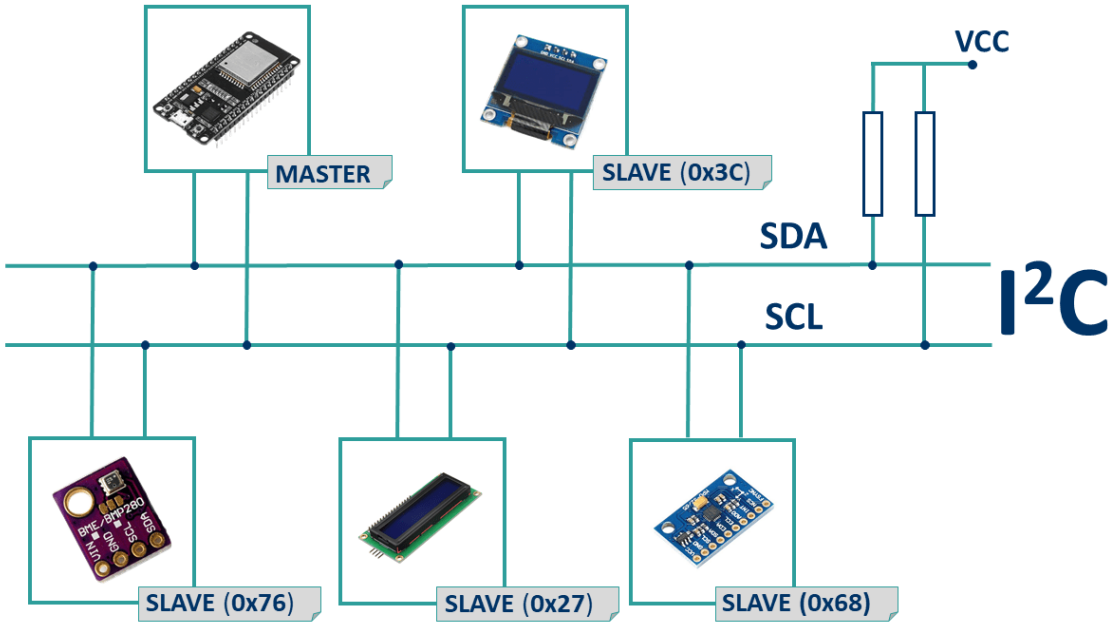
- **RESTRICCIONES**
  - Los pines **ADC2 no se pueden utilizar cuando se utiliza Wi-Fi**.
  - Los canales de entrada del ADC tienen una resolución de 12 bits. Esto significa que puede obtener lecturas analógicas que van de 0 a 4095, en las que 0 corresponde a 0 V y 4095 a 3,3 V. También puede configurar la resolución de sus canales en el código y el rango de ADC
  - Los pines del ESP32 ADC no tienen un comportamiento lineal. Probablemente no podrás distinguir entre 0 y 0,1 V, o entre 3,2 y 3,3 V. Debes tener esto en cuenta al utilizar los pines ADC.

PINES I2C

- El ESP32 tiene dos canales I2C y cualquier pin se puede configurar como SDA o SCL.
- los pines I2C predeterminados son:
  - **GPIO 21 (SDA)**
  - **GPIO 22 (SCL)**

Si desea utilizar otros pines cuando utilice la biblioteca de cables, solo necesita llamar:

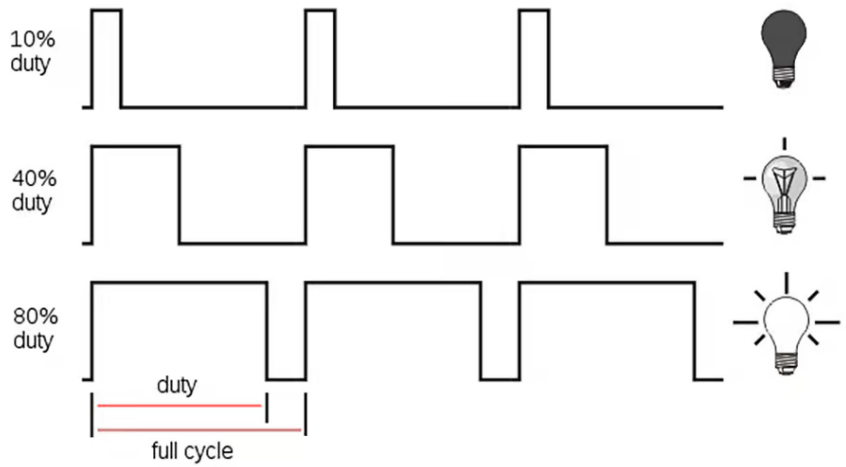
Wire.begin(SDA, SCL);



UNIVERSIDAD DE LIMA

PWM

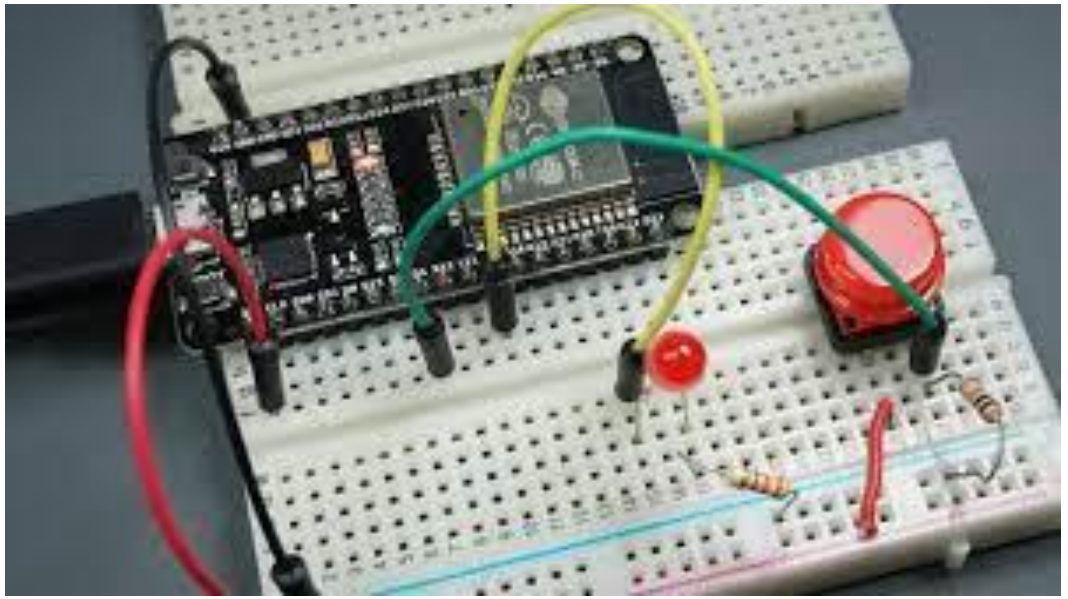
- El ESP32 tiene 16 canales que se pueden utilizar para generar señales PWM
- Se pueden tener hasta 16 salidas PWM diferentes.
- El uso de PWM difiere del de Arduino y puedes configurar más parámetros.



UNIVERSIDAD DE LIMA

PINES DIGITALES

- La gran mayoría de pines del ESP32 funcionan como entradas y salidas digitales
- No todos los pines tienen una resistencia interna, se recomienda usar el comando INPUT simple



UNIVERSIDAD DE LIMA

Tabla resumen

GPIO	INPUT	OUTPUT	Comentarios especiales
0	SI (pull-up interno)	SI	Debe estar a 0V durante el FLASH
1 (TX0)	NO	SI	Comunicación UART con la PC
2	SI (pull-down interno)	SI	Debe estar a 0V durante el FLASH/LED INTERNO
3 (RX0)	SI	NO	Comunicación UART con la PC
4	SI	SI	
5	SI	SI	
6	NO	NO	Conectado internamente al flash del esp32
7	NO	NO	Conectado internamente al flash del esp32
8	NO	NO	Conectado internamente al flash del esp32
9	NO	NO	Conectado internamente al flash del esp32
10	NO	NO	Conectado internamente al flash del esp32
11	NO	NO	Conectado internamente al flash del esp32
12 (MTDI)	SI (pull-down interno)	SI	Debe estar a 0V durante el BOOT
13	SI	SI	
14	SI	SI	
15 (MTDO)	SI (pull-up interno)	SI	Pin especial
16	SI	SI	

GPIO	INPUT	OUTPUT	Comentarios especiales
17	SI	SI	
18	SI	SI	
19	SI	SI	
21	SI	SI	
22	SI	SI	
23	SI	SI	
25	SI	SI	
26	SI	SI	
27	SI	SI	
32	SI	SI	
33	SI	SI	
34	SI	NO	Sin resistencia pull-up/pull-down interno
35	SI	NO	Sin resistencia pull-up/pull-down interno
36 (VP)	SI	NO	Sin resistencia pull-up/pull-down interno
39 (VN)	SI	NO	Sin resistencia pull-up/pull-down interno
EN	NO	NO	Conectado con el botón EN (ESP32 Reset)

UNIVERSIDAD DE LIMA

Configuración PWM

- La configuración del PWM en el esp32 depende de diversos parámetros:
  - El canal del microcontrolador: del 0 a 15
  - La frecuencia base del ciclo del reloj interno
  - La resolución del conversor analógico digital

Ejemplo PWM

- Los parámetros que usaremos serán los siguientes:
  - int freq = 5000; //frecuencia de 5000Hz
  - const int ledChannel = 0; //canal 0
  - const int resolution = 8; //resolución de 8bits (de 0 a 255)

- La configuración del pin PWM en el **setup** será de la siguiente manera:
  - ledcSetup(ledChannel, freq, resolution); //similar al pinMode
  - ledcAttachPin(ledPin, ledChannel); //definición del pin PWM

- En lugar de usar el comando analogWrite del Arduino se usará el siguiente comando:
  - ledcWrite(ledChannel, “valor de 0 a 255”);