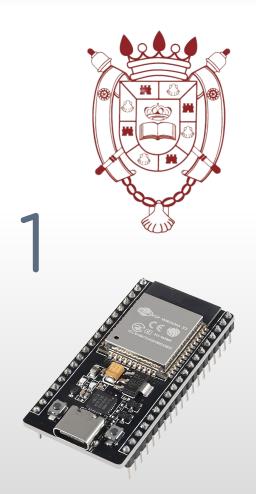
Internet de las Cosas con ESP32

Introducción al IoT, microcontroladores y tu primer programa

Ing. Fernando Raúl Vera Suasnávar



Universidad Nacional de Santiago del Estero - 2025

¿Qué es loT?

Red de dispositivos conectados

Recogen datos, procesan y actúan

Ejemplos

casas inteligentes relojes autos conectados



"Cualquier cosa que pueda conectarse, se conectará."

¿Por qué es importante loT?

Conecta el mundo físico con el digital

Automatización y mejores decisiones

Presente en hogares, industria, salud, agricultura.

ESP32

Ejemplo perfecto de loT barato y potente



IoT y microcontroladores

Microcontroladores

El corazón del loT

IoT =

- + MCU
- + conectividad
- + sensores/actuadores



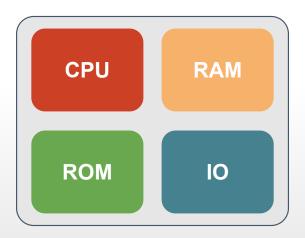


¿Qué es un microcontrolador?

Computador en un chip

- + CPU
- + memoria
- + periféricos

Diseñado para tareas específicas



[&]quot;Los microcontroladores nacieron para simplificar lo complejo: hacer mucho con poco."

¿Por qué nacen los microcontroladores?

Antes:

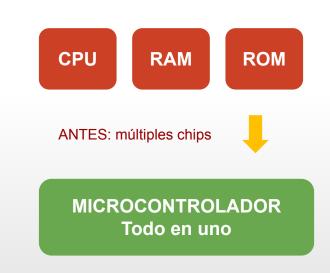
- microprocesadores
- + memorias externas
- → caros y complejos

Necesidad:

control barato y compacto

Solución:

- + integrar CPU
- + memoria
- + periféricos en un solo chip



Orígenes (1970-1980)

1974 → Texas Instruments TMS1000 primer microcontrolador comercial



1976 → Intel 8048 populariza el concepto de MCU



1980 → Intel 8051 estándar histórico, aún vigente



"La ciencia de hoy es la tecnología del mañana." – Edward Teller

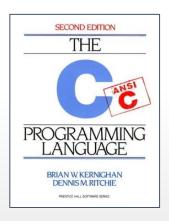
Expansión (1980-1990)

PIC (Microchip)

Clones del 8051

Transición de Assembly a C





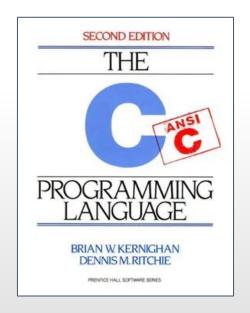
Democratización (1990-2000)

AVR (Atmel)

MSP430 (TI)

68HC05 (Freescale)

C se vuelve estándar global



"El lenguaje C unió a programadores de todo el mundo en un mismo idioma."

Profesionalización (2000-2010)

ARM Cortex-M, STM32

HAL

Hardware Abstraction Layer

CMSIS

Cortex Microcontroller Software Interface Standard

RTOS





Arduino (2005-2015)

Arduino con AVR (ATmega)

Lenguaje C++ simplificado

Democratización y comunidad maker





"Cuando la herramienta es simple, la creatividad se multiplica."

IoT y ESP32 (2014-hoy)

ESP8266 (2014)

WiFi barato

ESP32 (2016)

WiFi + BT, mayor potencia

Ecosistema

Arduino, MicroPython, TinyML







"Conectar es el comienzo; colaborar es el éxito." – Henry Ford

Tendencias actuales

RISC-V: arquitectura abierta

IA embebida: TinyML

Seguridad (incluye hardware criptográfico y mejores prácticas)

Rust (lenguaje moderno diseñado para evitar errores de memoria y mejorar la confiabilidad del software embebido).







Práctica

Pasos:

- → Instalar Visual Studio Code
- → Agregar extensión PlatformIO
- → Crear proyecto con ESP32
- → Conectar la placa ESP32
- → Subir el programa Blink

```
#include <Arduino.h>

void setup() {
   pinMode(2, OUTPUT);
}

void loop() {
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
}
```



Enlaces útiles

Visual Studio Code: https://code.visualstudio.com/

PlatformIO: https://platformio.org/install/ide?install=vscode

Consideraciones GPIO / DEVKIT V1

Nivel lógico

3.3 V (no tolera 5 V)

→ Corriente

12 mA recomendado (40 mA máx.)

Restricciones

Boot: GPIO0, 2, 15

Flash: GPIO6-11

Solo entrada: GPIO34-39

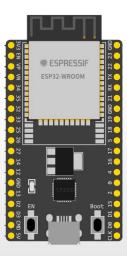
LED interno: GPIO2

Seguros para práctica

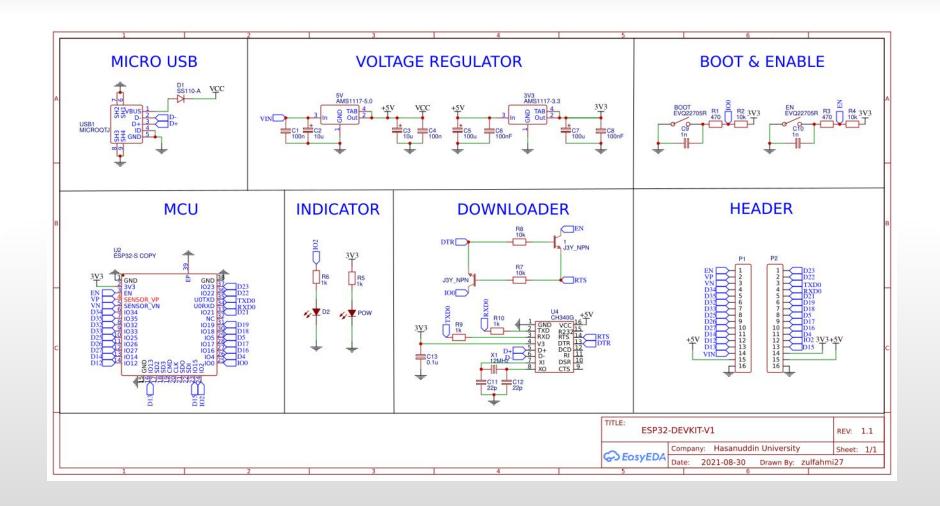
GPIO2, 4, 5,

12-19, 21-23,

25–27, 32–33

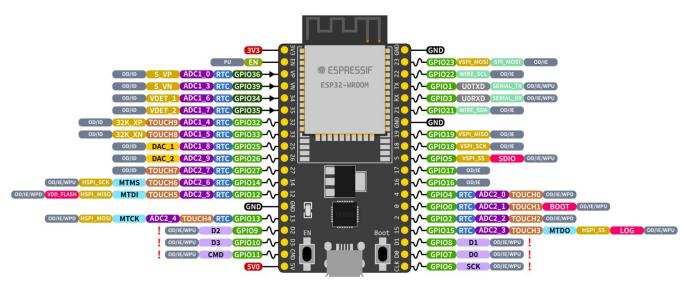


GPIO: General Purpose Input/Output



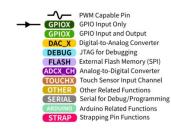
ESP32-DevKitC





ESP32 Specs

32-bit Xtensa® dual-core @240MHz
Wi-Fi IEEE 802.11 b/g/n 2.4GHz
BLuetooth 4.2 BR/EDR and BLE
520 KB SRAM (16 KB for cache)
448 KB ROM
34 GPIOs, 4x SPI, 3x UART, 2x I2C,
2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,
1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet



GPIO STATE

WPU: Weak Pull-up (Internal)
WPD: Weak Pull-down (Internal)
PU: Pull-up (External)
IE: Input Enable (After Reset)
ID: Input Disabled (After Reset)
OE: Output Enable (After Reset)
OD: Output Disabled (After Reset)

RTC Power Domain (VDD3P3_RTC)

Power Rails (3V3 and 5V)

Pin Shared with the Flash Memory

Can't be used as regular GPIO

GND

Ground

Consideraciones GPIO / SUPERMINI

Nivel lógico

3.3 V (no tolera 5 V)

← Corriente

12 mA recomendado (20 mA máx por pin, ~100 mA total)

Restricciones

Boot: GPIO0, GPIO9

USB interno: GPIO18, GPIO19

LED interno: GPIO8

Seguros para práctica

GPIO2, 3, 4, 5, 6, 7, 8, 10, 20, 21



GPIO: General Purpose Input/Output

ESP32C3 Supermini

