ESP32-WROOM-32E module and ESP32-DevKitC kit

Arquiteturas para Sistemas Embutidos

João Gameiro, 93097
Pedro Abreu, 93240

Turma TP1
Grupo 3

22 Março 2022

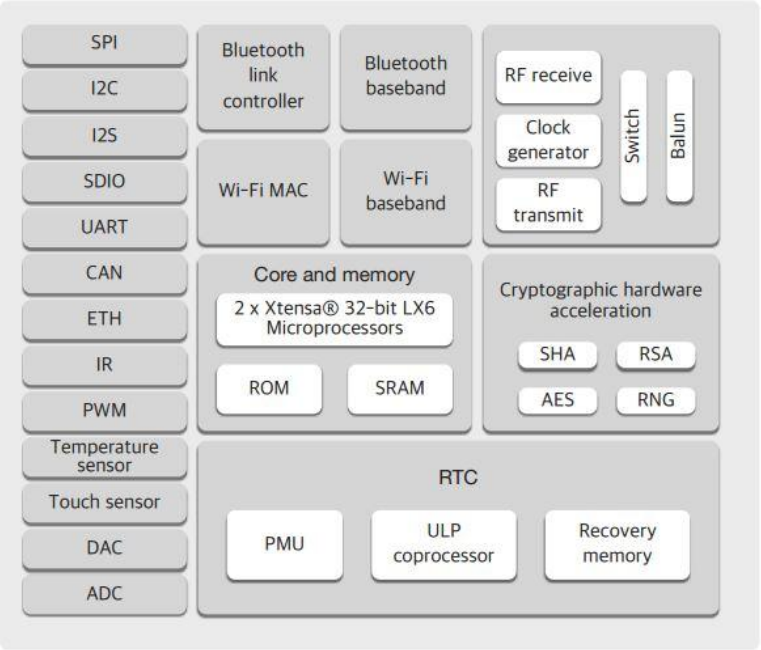
universidade de aveiro

What is ESP32-WROOM-32E?

- Is a powerful, generic Wi-Fi + Bluetooth + Bluetooth LE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

Overall view of the architecture

- Xtensa Dual-Core 32-bit LX6 microprocessor up to 240MHz (x2)
- ROM (448kB) and SRAM(520kB) internal memories
- Support for WiFi and Bluetooth
- RTC module
- Cryptographic hardware acceleration
- RF antenna
- A set of peripherals
  - SPI, I2C, UART,CAN,...
  - Temperature and Touch sensors
  - DAC & ADC

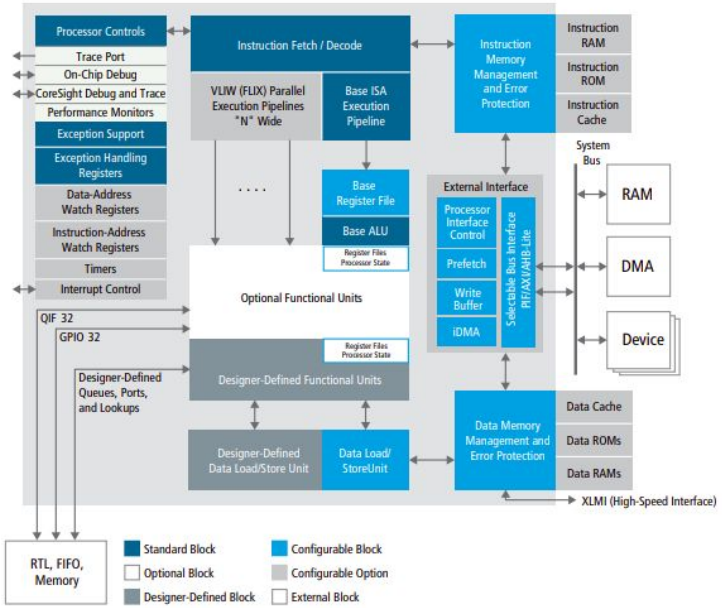


ESP32 - Xtensa LX6 Microprocessor

- General purpose processors lack the bandwidth needed to perform complex tasks
  - Network packet processing
  - Video processing
  - Digital cryptography
- Xtensa LX6 are ideal for dealing with complex compute-intensive digital signal processing (DSP) applications
  - **Configurable**: users are offered with pre-verified options ranging from memory size and width to complex DSP functions.
  - **Extensible**: It is possible to implement datapath elements in the processor pipeline and add more I/Os.

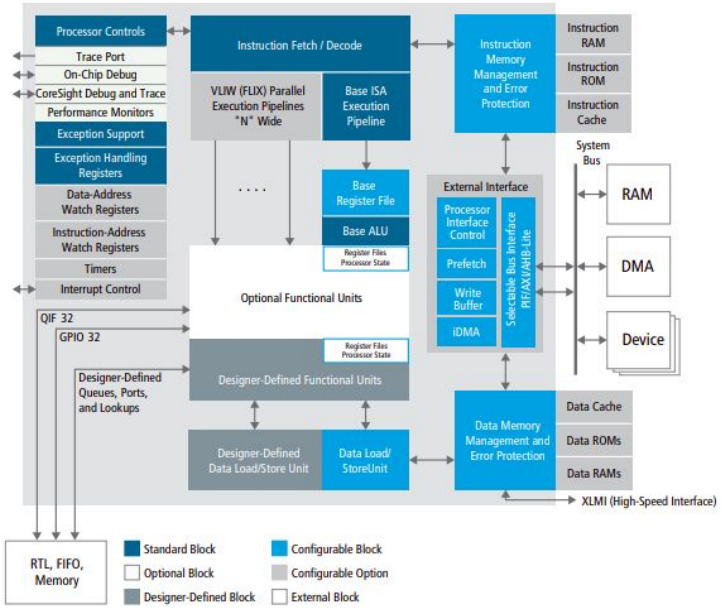
ESP32 - Xtensa LX6 Microprocessor

- Highly efficient, small, low-power 32-bit base architecture
- Harvard memory model: two separated memories one for data and another for instructions
- Selectable 5-or-7-stage pipeline to accommodate different memory speeds
- VLIW (FLIX) execution
    - Very Long Instruction Word
    - Flexible Length Instruction eXtensions



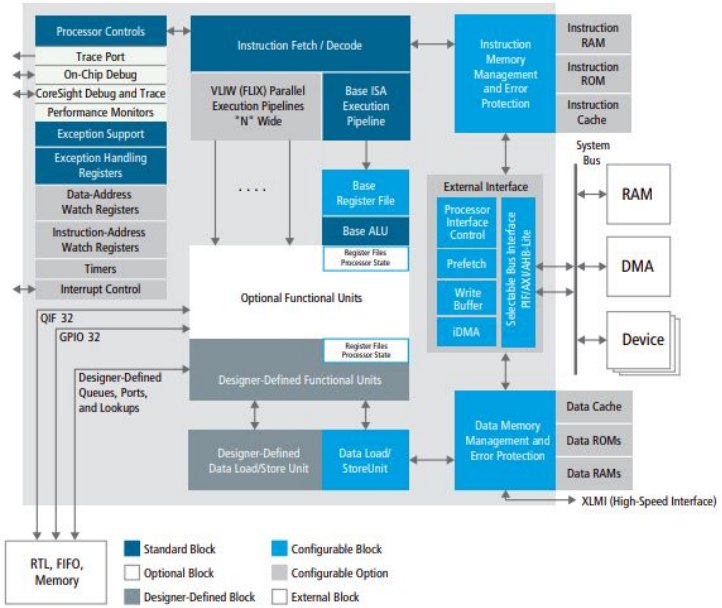
ESP32 - Xtensa LX6 Microprocessor

- 32-bit ALU
- Up to 64 general-purpose physical registers and 6 special purpose registers
- 80 base instructions, including 16- and 24-bit (rather than 32-bit) RISC instruction encoding
- ISA are backward compatible, because it was developed for extensibility
- IEEE 754-compliant single-/double-precision scalar floating-point option



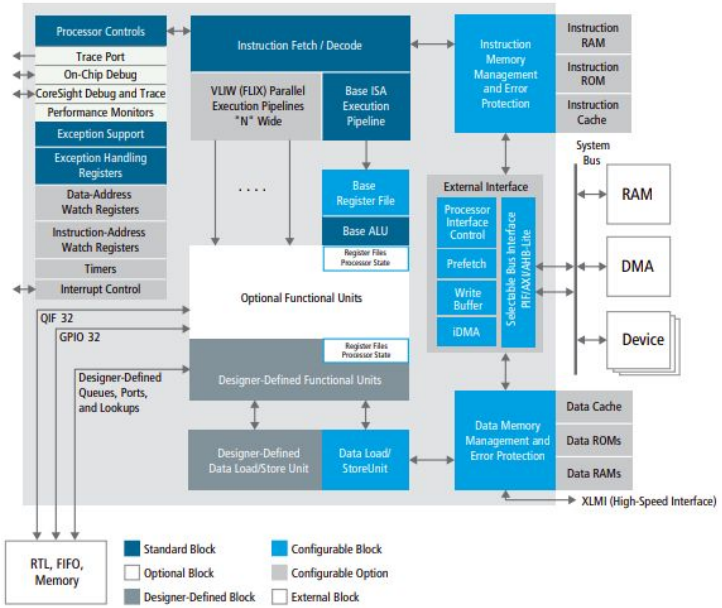
ESP32 - Xtensa LX6 Microprocessor

- Data and instruction caches
  - Up to 4-way set associative
  - Up to 128 kB
  - Write-back and write-through cache write policy
- One or two 32-/64-/128-/256-/512-bit-wide load/store units
- Up to six local memory banks can be connected for instruction and data accesses (up to 12 in total). Memory banks may be local ROM, RAM, or cache ways
- Memory management unit (MMU) with translation look-aside buffers (TLBs)



ESP32 - Xtensa LX6 Microprocessor

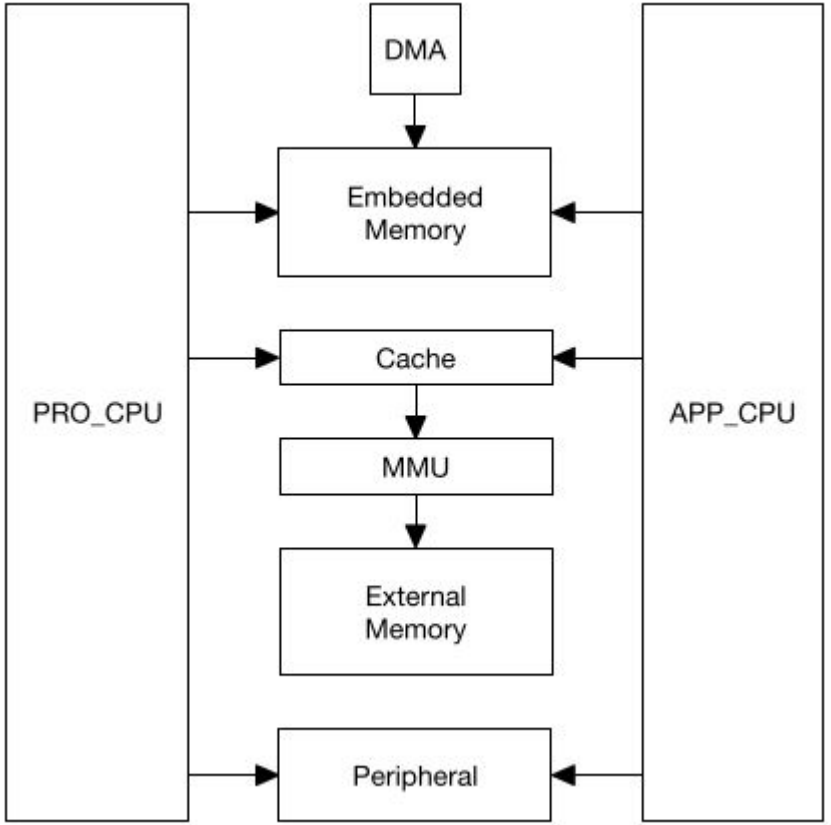
- Optional hardware prefetch to reduce memory latencies
- Up to 32 interrupts with up to seven levels of priority and specific registers to deal with exceptions
- Multiple custom-width GPIO ports for direct control and monitoring of peripherals
- Multiple custom-width queue interfaces for streaming data into and out of the processor via FIFOs



ESP32 - System Structure

- PRO_CPU is for the protocols
  - Wifi, Bluetooth
  - Internal Peripherals (SPI, I2C, etc)
- APP_CPU is used for the application code
- For most purposes the two CPUs are interchangeable

- Processors are mapped symmetrically to this address space
- This differentiation is done in the Espressif Internet Development Framework (ESP-IDF).
- ESP-IDF uses freeRTOS for switching between the processors and data exchange between them

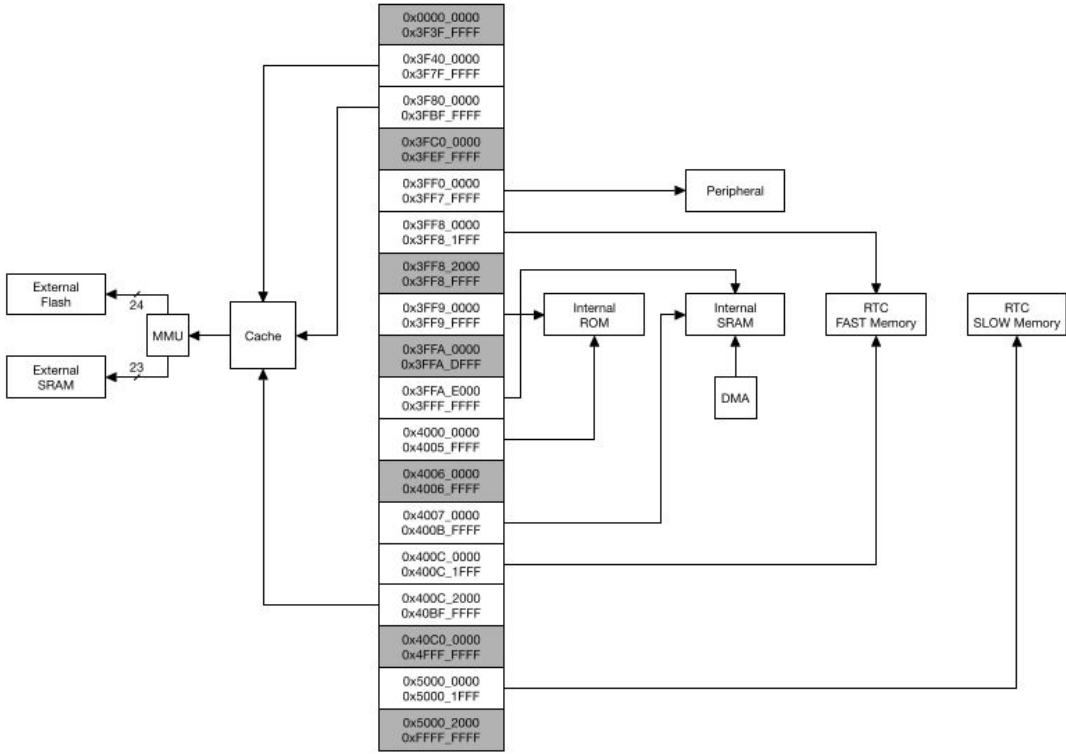


ESP32 - Memory

- Embedded Memory
  - 448 kB Internal ROM
  - 520 kB Internal SRAM
  - 8 kB RTC FAST Memory
    - Can be accessed as both instruction and data memory by the PRO_CPU
  - 8 kB RTC SLOW Memory
    - Accessed by the coprocessor during the Deep-sleep mode
  - Both of the last two are SRAMs and contained in the RTC Module
- External Memory
  - Supports up to 16 MB off-Chip SPI Flash
  - Supports up to 8 MB off-Chip SPI SRAM

ESP32 - Address Space

- When CPU accesses external memory through the Cache and MMU, the cache will map the CPU's address to an external physical memory address (in the external memory address space), according to the MMU settings



ESP32 - RTC module

Designed to handle the entry into, and exit from, the low-power mode, and control the clock sources and power switch to generate power-gating, clock-gating, and reset signals.

- RTC FSM to record the power state
- Sleep & wake up controllers to handles the entry into and exit from the low-power mode
- It includes RTC main timer, ULP coprocessor timer and touch timer
- Digital & analog power controllers

ESP32 - RTC module (CoProcessor)

- Ultra-low-power (ULP) processor that remains powered on during the Deep-sleep mode
- Contains four 16-bit general-purpose registers for manipulating data and accessing memory
- Is able to wake up the digital core or send an interrupt to the CPU.
- It can be a good supplement to, or replacement of, the CPU, especially for power-sensitive applications.
- ULP coprocessor can be started by software or a periodically-triggered timer and get into sleep mode, as well.

ESP32 - Cryptographic hardware acceleration

- 3 cryptographic hardware accelerators:
  - AES, SHA and RSA Accelerator
- RNG (Random Number Generator):
  - Used to generate 32 bit random numbers that can be used in cryptographic operations
  - The random number generator generates true random numbers, which means random number generated from a physical process, rather than by means of an algorithm.
- Flash Encryption/Decryption block to encrypt/decrypt code and write/read to off-chip flash memory for enhanced hardware security

ESP32 - Peripherals

- 3 UART controllers, for exchanging data that need minimal system resources
- 2 I2S interfaces for streaming digital data in multimedia applications
- I2C bus for communication with several external devices connected to the same bus as ESP32
- 4 SPI controllers to communicate with external devices that use the SPI protocol
- 3 Watchdog Timers (one in the RTC module)
- 5 controllers dedicated to operating ADCs
- 2 8-bit DAC channels

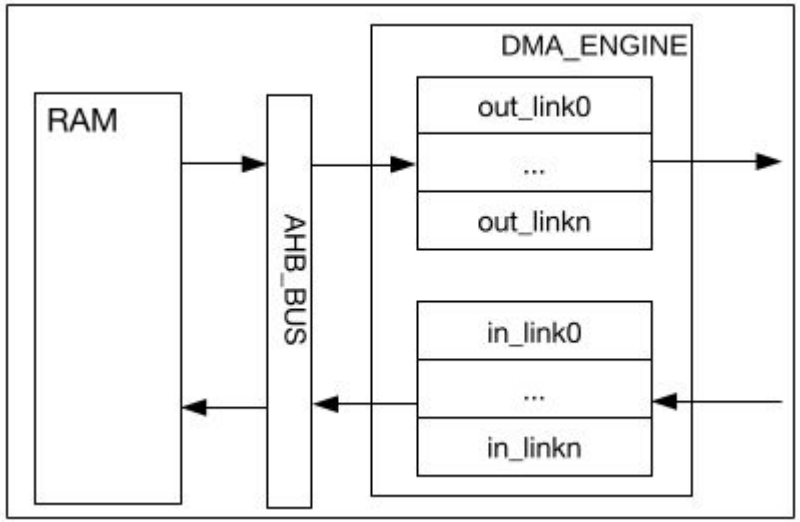
ESP32 - Peripherals

- 4 General purpose 64 bit timers based in 16 bit prescalers
- PWM peripheral for motor and power control. It provides six PWM outputs
- Ethernet PHY to send and receive data via Ethernet MAC (Media Access Controller) according to the IEEE 802.3 standard
- Two sensors:
  - Touch (10 capacitive touch pads / GPIOs)
  - Hall effect sensor

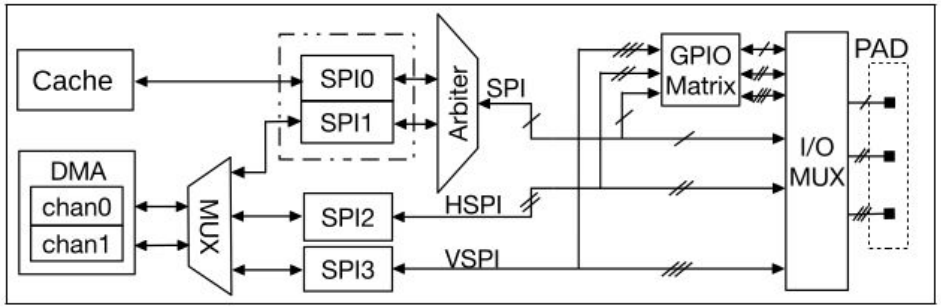
ESP32 - DMA

- Direct Memory Access (DMA) is used for high-speed data transfer between peripherals and memory, as well as from memory to memory. Data can be quickly moved with DMA without any CPU intervention, thus allowing for more efficient use of the cores when processing data.

- In the ESP32, 13 peripherals are capable of using DMA for data transfer (SPI, I2S, UART, etc)
- AHB bus architecture
- Supports full-duplex and half-duplex data transfers



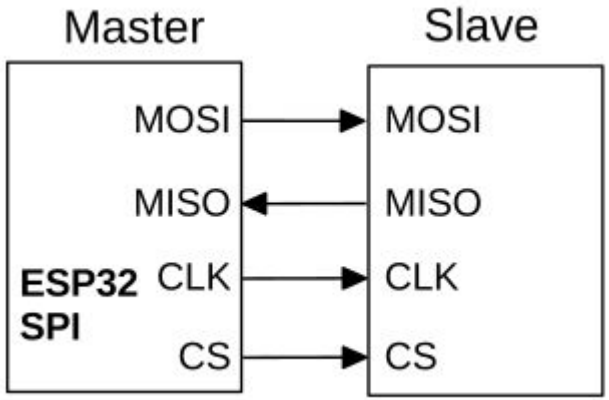
ESP32 - SPI



- ESP32 integrates four SPI controllers which can be used to communicate with external devices that use the SPI protocol.
  - SPI0 is used as buffer for accessing external memory
  - SPI1 can be used as master
  - SPI2 and SPI3 can be configured as either a master or a slave
- SPI1, SPI2 and SPI3 share two DMA channels
- SPI1 and SPI0 share one signal bus through an arbiter
- The I/O lines connected to the controllers can be mapped to pins via either the IO_MUX module or the GPIO matrix
- Programmable clock

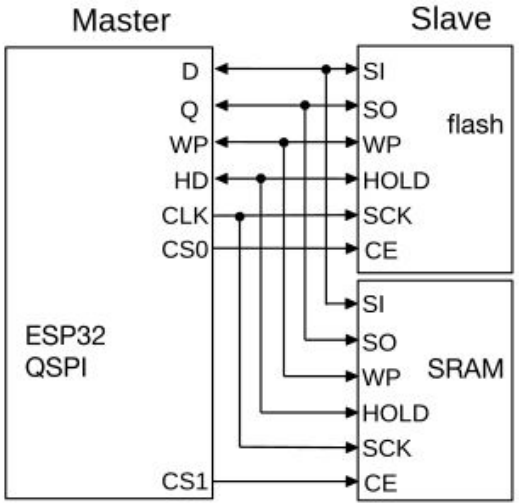
ESP32 - SPI

- The SPI controller supports four-line full-duplex/half-duplex communication
  - MOSI, MISO, CS, and CLK lines
- Supports three-line half-duplex-only communication
  - DATA, CS, and CLK lines
- It is also featured hardware support for SPI interruptions
  - SPI interruptions
  - SPI DMA interrupt



ESP32 - SPI

- ESP32 SPI controllers support SPI bus memory devices (such as flash and SRAM).
- SPI1, SPI2 and SPI3 controllers can also be configured as QSPI master to connect to external memory.
- ESP32 SPI has 16 × 32 bits of data buffer to buffer data-send and data-receive operations.



ESP32 - SPI

- Command phase - >address phase -> dummy phase -> data phase

Command	Description
0x1	Received by slave; writes data sent by the master into the slave status register via MOSI.
0x2	Received by slave; writes data sent by the master into the slave data buffer via MOSI.
0x3	Sent by slave; sends data in the slave buffer to master via MISO.
0x4	Sent by slave; sends data in the slave status register to master via MISO.
0x6	Writes master data on MOSI into data buffer and then sends the date in the slave data buffer to MISO.

What is ESP32-DevKitC kit?

- ESP32-DevKitC is a small-sized ESP32-based microcontroller produced by Espressif.
- One of many:
  - ESP-WROVER-KIT
  - ESP32-PICO-KIT
  - ESP32-Ethernet-Kit
  - ESP32-DevKit-S(-R)
  - ESP32-PICO-KIT-1
  - ...

ESP32-DevKitC - Applications

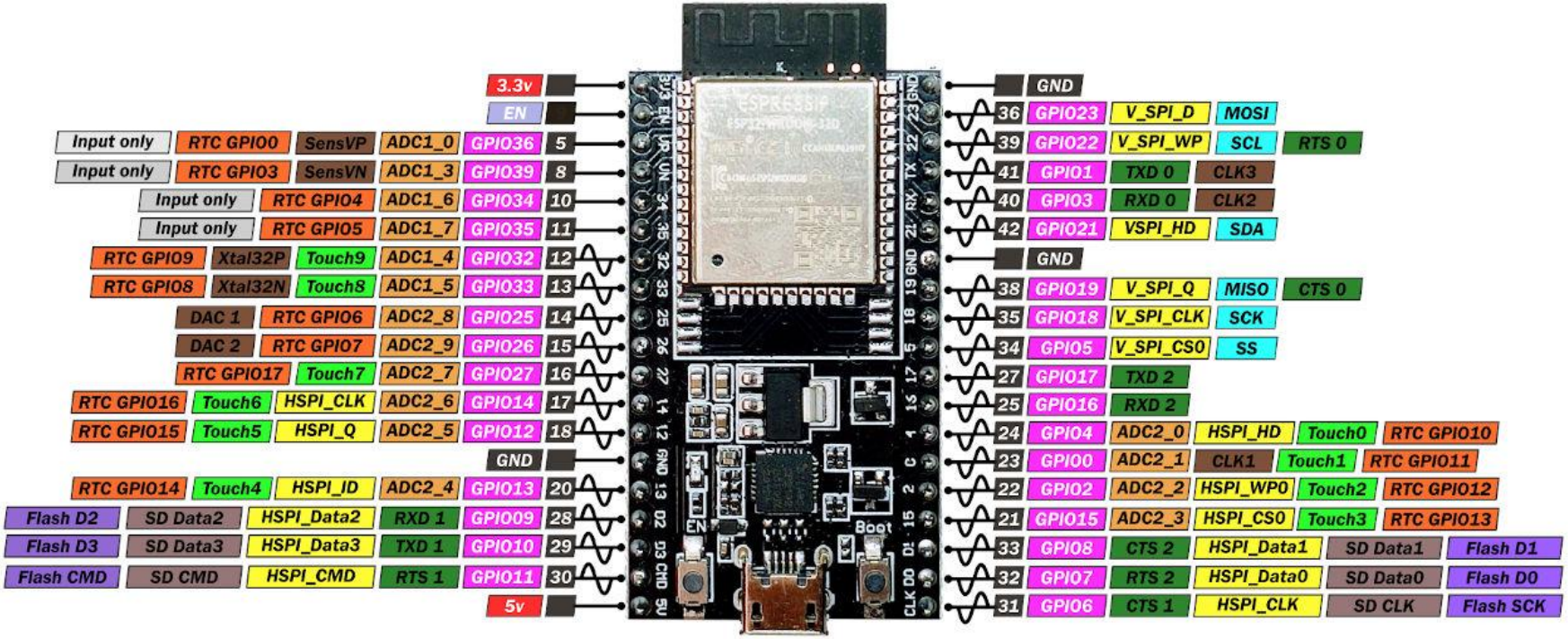
- Prototyping of IoT devices
- Low power battery operated applications
- Network projects
- Easy to use for beginner level DIYers and makers.
- Projects requiring Multiple I/O interfaces with Wi-Fi and Bluetooth functionalities

ESP32-DevKitC - GPIO

- The ESP32 chip features 34 physical GPIO pads
- Each pad can be used as a general-purpose I/O, or be connected to an internal peripheral signal

GPIO pins are 0-19, 21-23, 25-27, 32-39. But GPIO pads 34-39 are input only

ESP32 DevKitC V4 PINOUT



ESP32-DevKitC - Extra memory (Support for external RAM)

- ESP32 has the ability to also use up to 4 MB of external SPI RAM memory.
- PSRAM connected in parallel with the SPI flash of ESP32 chip.

Is Possible to:

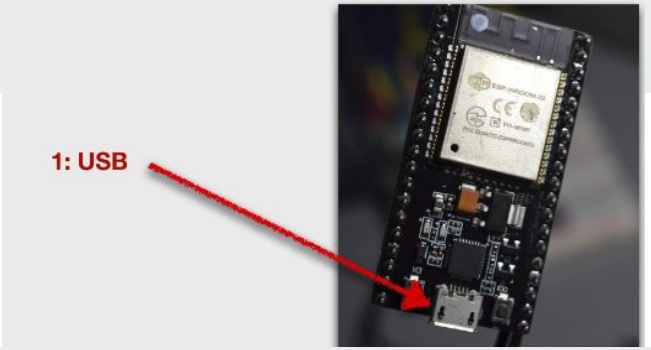
- Provide External RAM via malloc()
- Integrate RAM into the ESP32 Memory Map

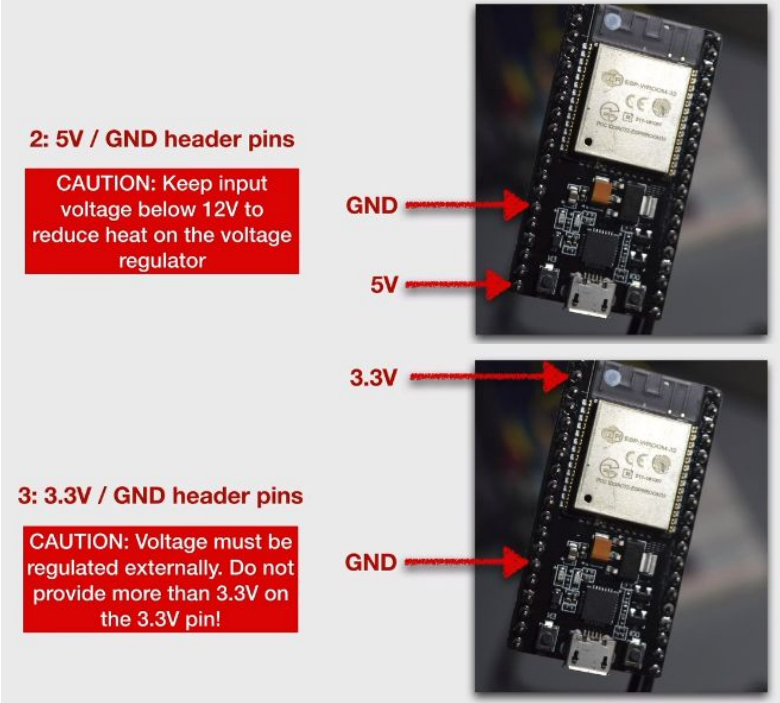
It as some Restrictions!

ESP32-DevKitC - Power Supply options

- USB (preferably)
- Unregulated 5V
- Regulated 3.3V

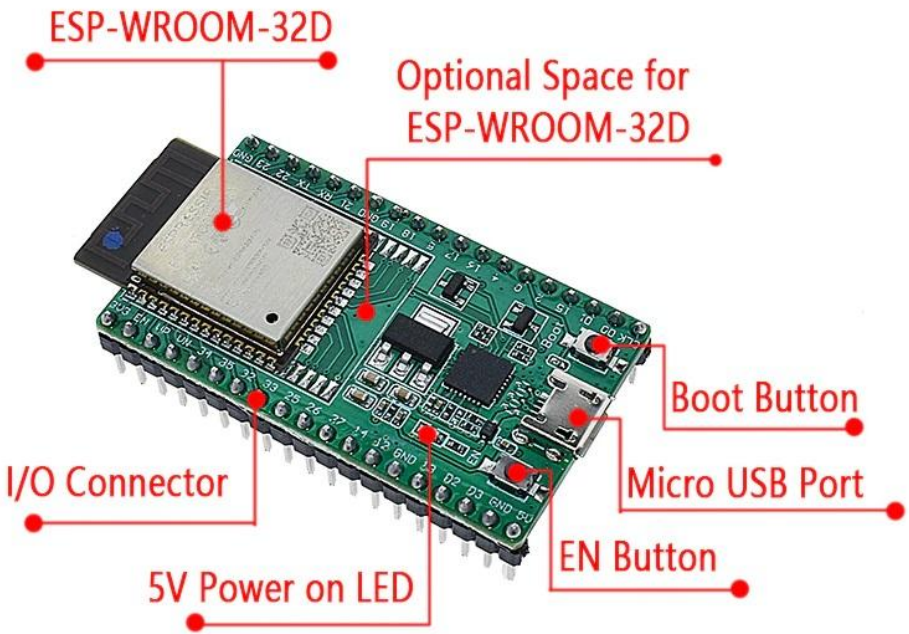
Use only one at a time!





ESP32-DevKitC - Buttons & LEDs

- Two buttons one for booting the kit and the other for the reset
- One LED that supports 5v



Bibliografia - Imagens

Por ordem de aparecimento:

- https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.robotshop.com%2Fde%2Fen%2Fesp32-wroom-mcu-module-16mb-pcb-antenna.html&psig=AOvVaw22LKGy2MNNFR9UChUIofUu&ust=1647718382512000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCICzmLmz0PYCFQAAAAAdAAAAABAf
- https://www.exploreembedded.com/wiki/images/0/01/FeaturesESP32.JPG
- https://mirrobo.ru/wp-content/uploads/2016/11/Cadence_Tensillica_Xtensa_LX6_ds.pdf (Página 4)
- https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (Página 25)
- https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (Página 25)
- https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (Página 112)
- https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (Página 117)
- https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (Página 118)
- https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (Página 123)
- https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (Página 119)
- https://www.botnroll.com/10966-medium_default/placa-de-desenvolvimento-esp32-espressif.jpg
- https://www.mischianti.org/wp-content/uploads/2021/07/ESP32-DEV-KIT-DevKitC-v4-pinout-mischianti.jpg
- https://techexplorations.com/guides/esp32/begin/power/
- https://www.botnroll.com/img/cms/ARD02088-ESP32-DEVKitC-WROOM32-32D-layout.jpg

Bibliografia

- ESP32
  - https://mirrobo.ru/wp-content/uploads/2016/11/Cadence_Tensillica_Xtensa_LX6_ds.pdf
  - https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
  - https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
  - https://www.exploreembedded.com/wiki/Overview_of_ESP32_features._What_do_they_practically_mean%3F
- ESP32-DevKitC
  - https://components101.com/microcontrollers/esp32-devkitc
  - https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/external-ram.html
  - https://techexplorations.com/guides/esp32/begin/power/