

Introduction

Specifications

Processors:

Clock frequency: up to 240 MHz

Performance: up to 600 DMIPS

Ultra low power co-processor

Wireless connectivity:

Memory:

Security:

Hardware and Software Requirements

Installation

Prerequisites

Download

Setup the tools

Environment Variables

Install VSCode and ESP-IDF plugin

Tips

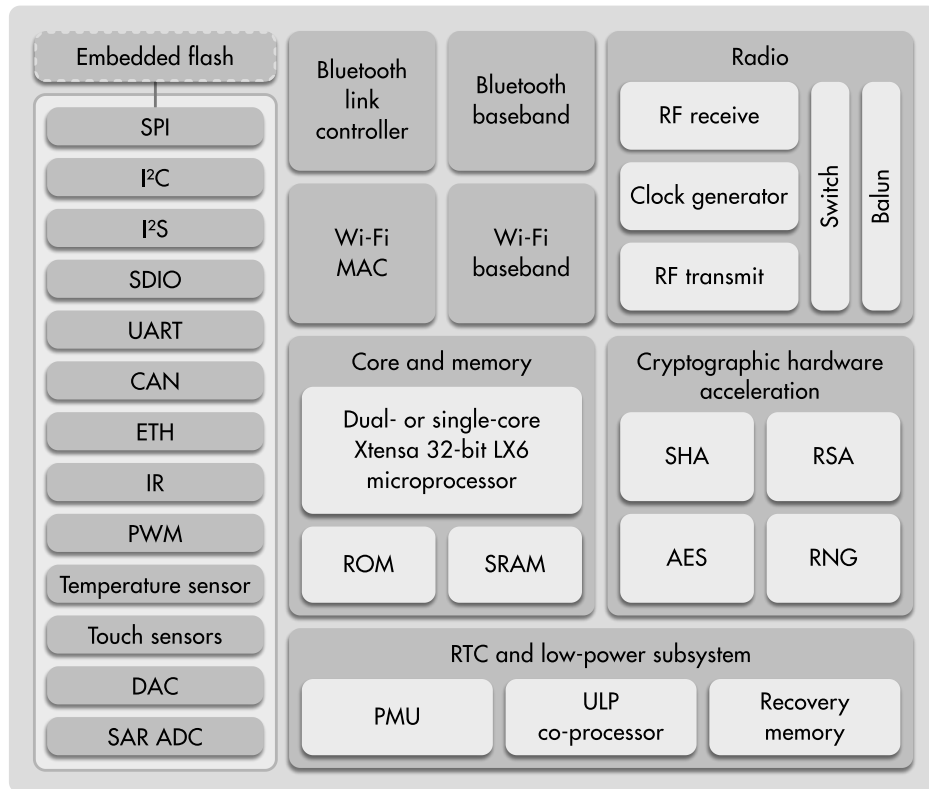
Introduction

ESP32 is a system on a chip that integrates the following features:

- Wi-Fi (2.4 GHz band)
- Bluetooth
- Dual high performance cores
- Ultra Low Power co-processor
- Multiple peripherals

Espressif provides basic hardware and software resources to help application developers realize their ideas using the ESP32 series hardware. The software development framework by Espressif is intended for development of Internet-of-Things (IoT) applications with Wi-Fi, Bluetooth, power management and several other system features.

ESP32 FUNCTION BLOCK DIAGRAM



Specifications

Processors:

- Main processor: Tensilica Xtensa 32-bit LX6 microprocessor
- Cores: 2 or 1 (depending on variation)
- All chips in the ESP32 series are dual-core except for ESP32-S0WD, which is single-core.

Clock frequency: up to 240 MHz

Performance: up to 600 DMIPS

Ultra low power co-processor

- allows to do ADC conversions, computation, and level thresholds while in deep sleep

Wireless connectivity:

- Wi-Fi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz up to 150 Mbit/s)
- Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)

Memory:

- Internal memory: ROM: 448 KiB
For booting and core functions.
- SRAM: 520 KiB
For data and instruction

Security:

- IEEE 802.11 standard security features all supported, including WPA, WPA/WPA2 and WAPI
- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

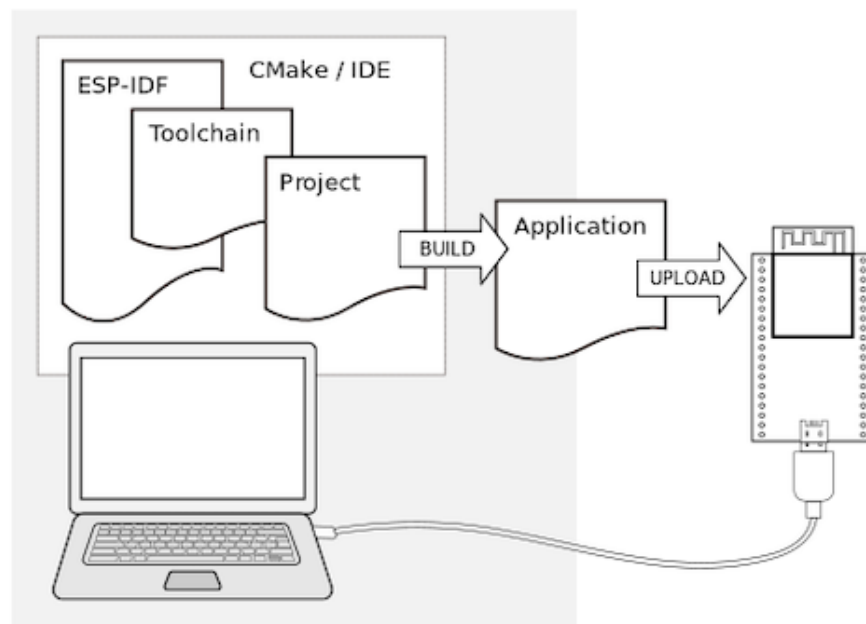
Hardware and Software Requirements

Hardware:

- An **ESP32** board
- **USB cable** - USB A / micro USB B
- **Computer** running Windows, Linux, or macOS

Software:

- **Toolchain** to compile code for ESP32
- **Build tools** - CMake and Ninja to build a full **Application** for ESP32
- **ESP-IDF** that essentially contains API (software libraries and source code) for ESP32 and scripts to operate the **Toolchain**



Installation

Prerequisites

```
1 sudo apt-get install git wget flex bison gperf python3 python3-pip python3-setuptools  
cmake ninja-build ccache libffi-dev libssl-dev dfu-util
```

Download

```
1 mkdir -p ~/esp  
2 cd ~/esp  
3 git clone --recursive https://github.com/espressif/esp-idf.git
```

Setup the tools

```
1 cd ~/esp/esp-idf  
2 ./install.sh
```

Environment Variables

```
1 . $HOME/esp/esp-idf/export.sh
```

For permanent setup

File name: `~/.bashrc`

```
1 alias get_idf='. $HOME/esp/esp-idf/export.sh'
```

Any time you want to setup IDF use the command

```
1 get_idf
```

Install VSCode and ESP-IDF plugin

Download VSCode from <https://code.visualstudio.com/>

Launch VSCode Quick Open Ctrl+P on Linux/ Windows) and then paste the following command and press enter

```
1 ext install esp-idf-extension
```

Tips

```
1 sudo usermod -a -G dialout $USER
```

```
1  #define LOG_LOCAL_LEVEL ESP_LOG_VERBOSE
2  #include "esp_log.h"
3
4  #include <stdio.h>
5  #include "sdkconfig.h"
6  #include "freertos/FreeRTOS.h"
7  #include "freertos/task.h"
8  #include "freertos/semphr.h"
9  #include "freertos/queue.h"
10
11 void app_main(void)
12 {
13     while (1)
14     {
15         vTaskDelay(1000/portTICK_PERIOD_MS);
16     }
17 }
18
```