PADO

Labs

Microcontroladores Aplicados a IoT

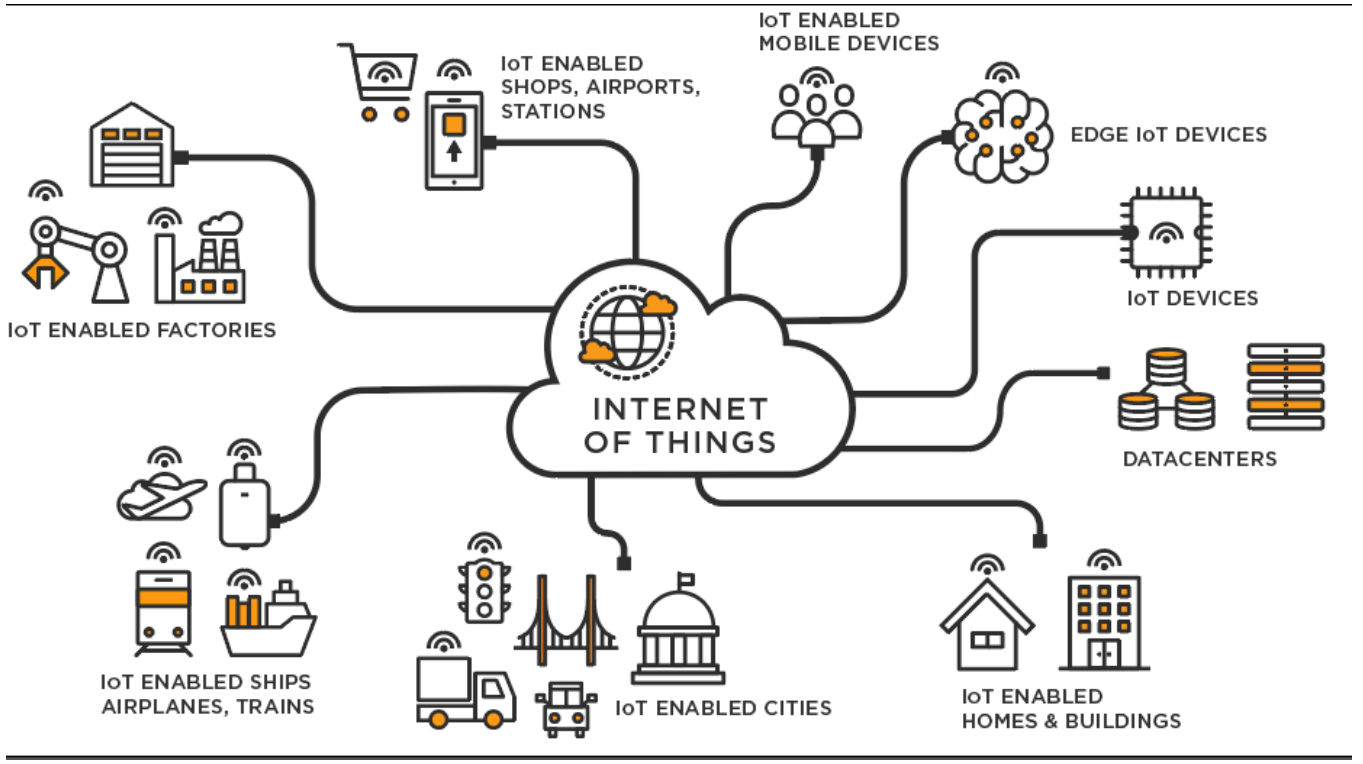
Dilson Liukiti Ito

01 – IoT e ESP32

Microcontroladores Aplicados a IoT

O que é IoT?

- Internet of Things
- Comunicação M2M (machine to machine)

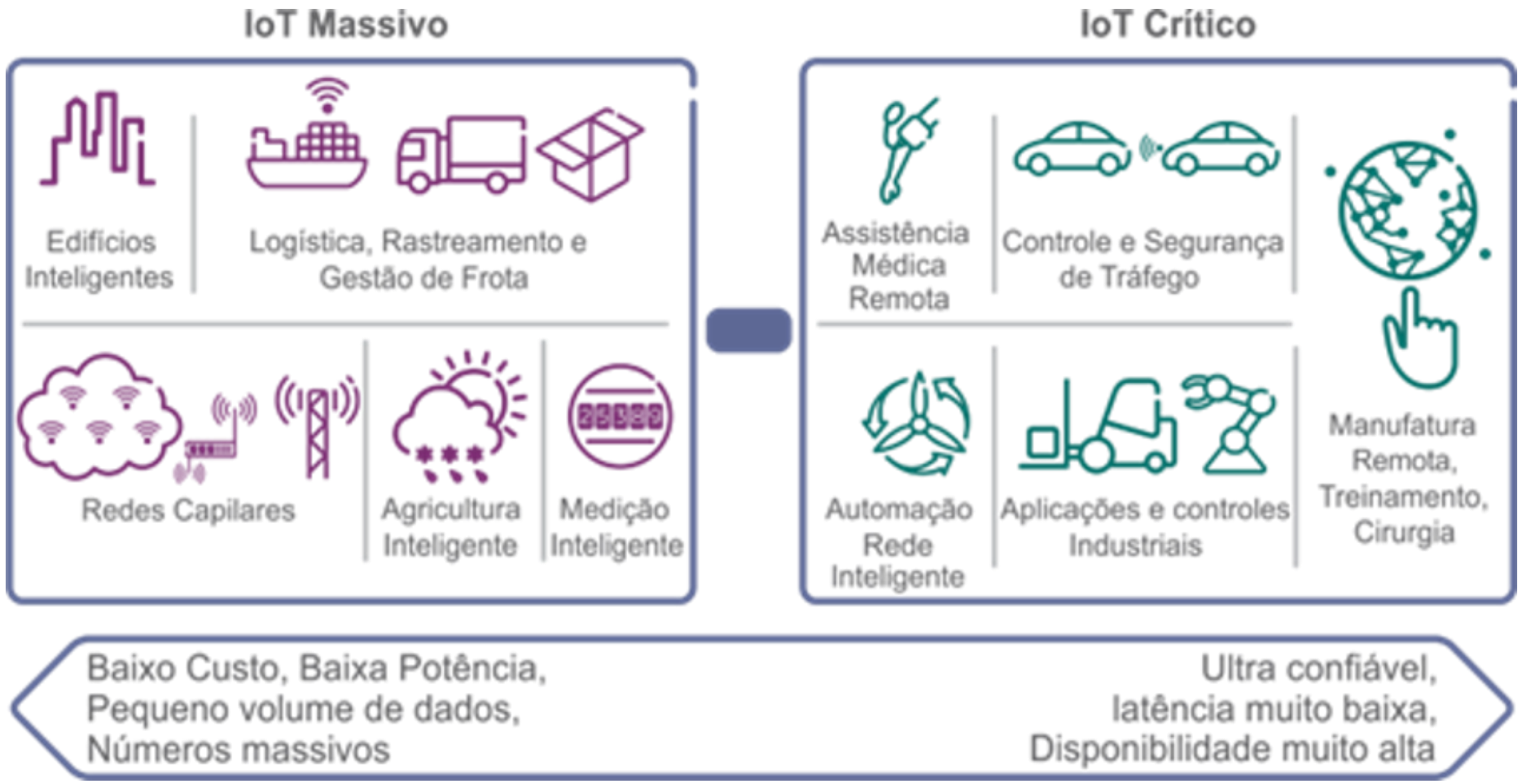


https://www.tibco.com/pt-br/reference-center/what-is-the-internet-of-things-iot

Categorias de IoT

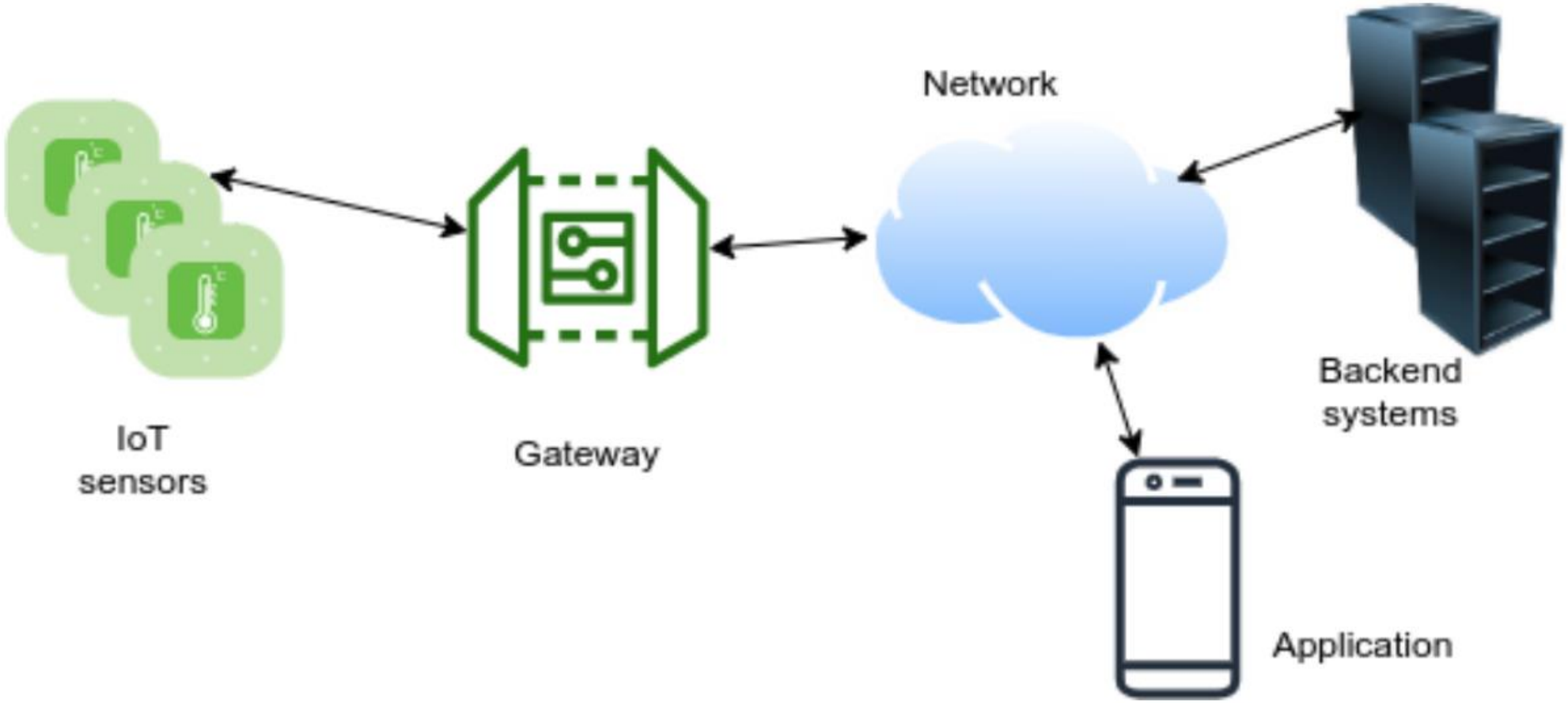
- IoT **Massivo**: tipicamente **sensores** que enviam informações periodicamente e implementados em **larga escala**. Para cumprir com requisitos comerciais também devem apresentar **custos reduzidos, baixo consumo de energia e ampla cobertura**;
- IoT **Crítico**: aplicações que demandam **grande confiabilidade, disponibilidade e baixas taxas de latência**. Apresentam volumes menores porém com **valores comerciais mais altos**.

Categorias de IoT (cont.)



Fonte: Adaptado e Traduzido de Ericsson (2016, p. 4)

Estrutura Básica



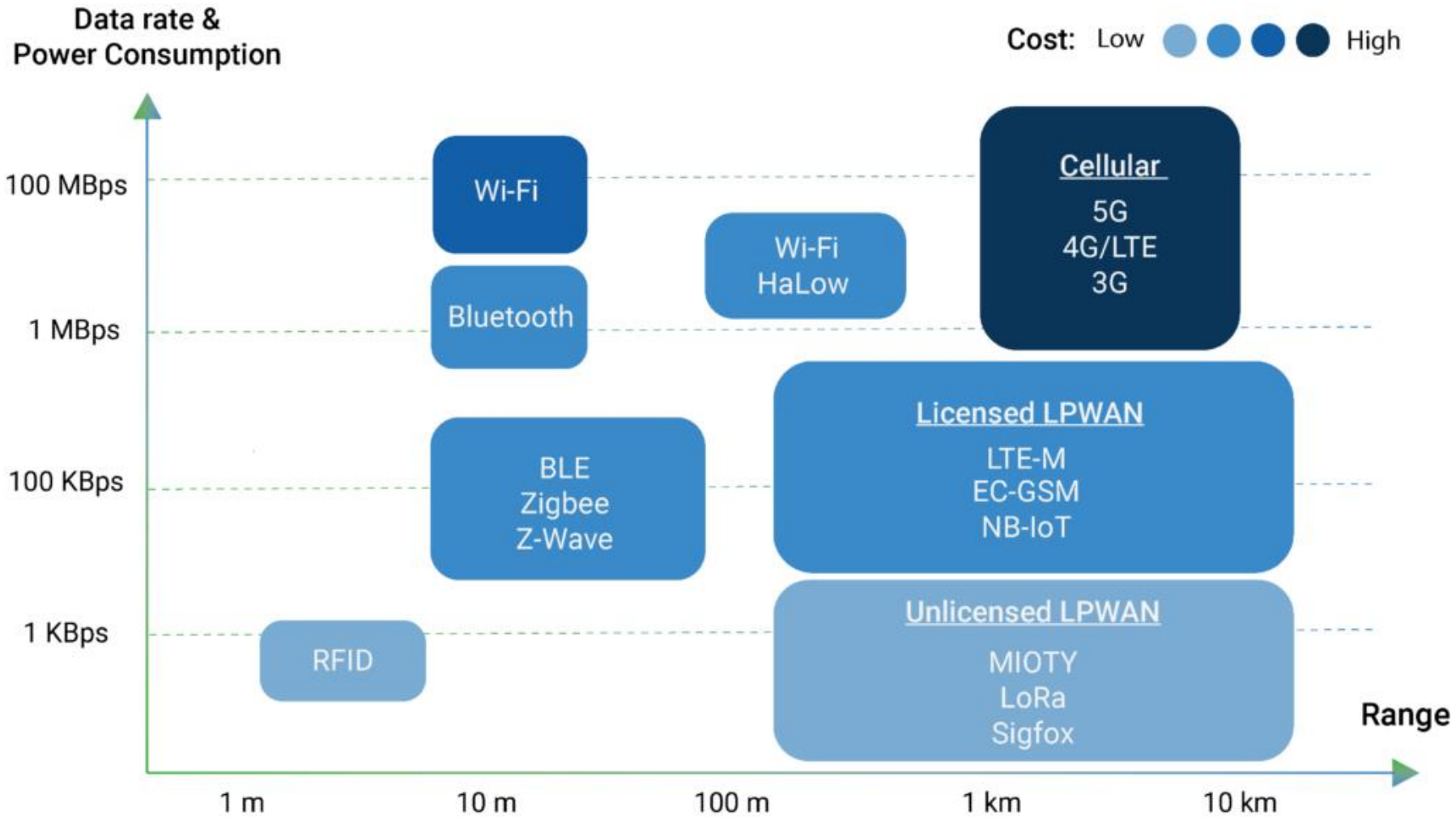
Developing IoT Projects with ESP32 (2021)

Capacidades necessárias para implementar IoT

- Cíber segurança
- Integração
- Analytics
- Rede e comunicações
- Gestão de dados
- Gerenciamento de dispositivo
- Desenvolvimento de aplicativos

https://www.tibco.com/pt-br/reference-center/what-is-the-internet-of-things-iot

Redes sem fio IoT



https://iot-industrial-devices.com/wp-content/uploads/2019/11/iot-wireless-1030x578.png

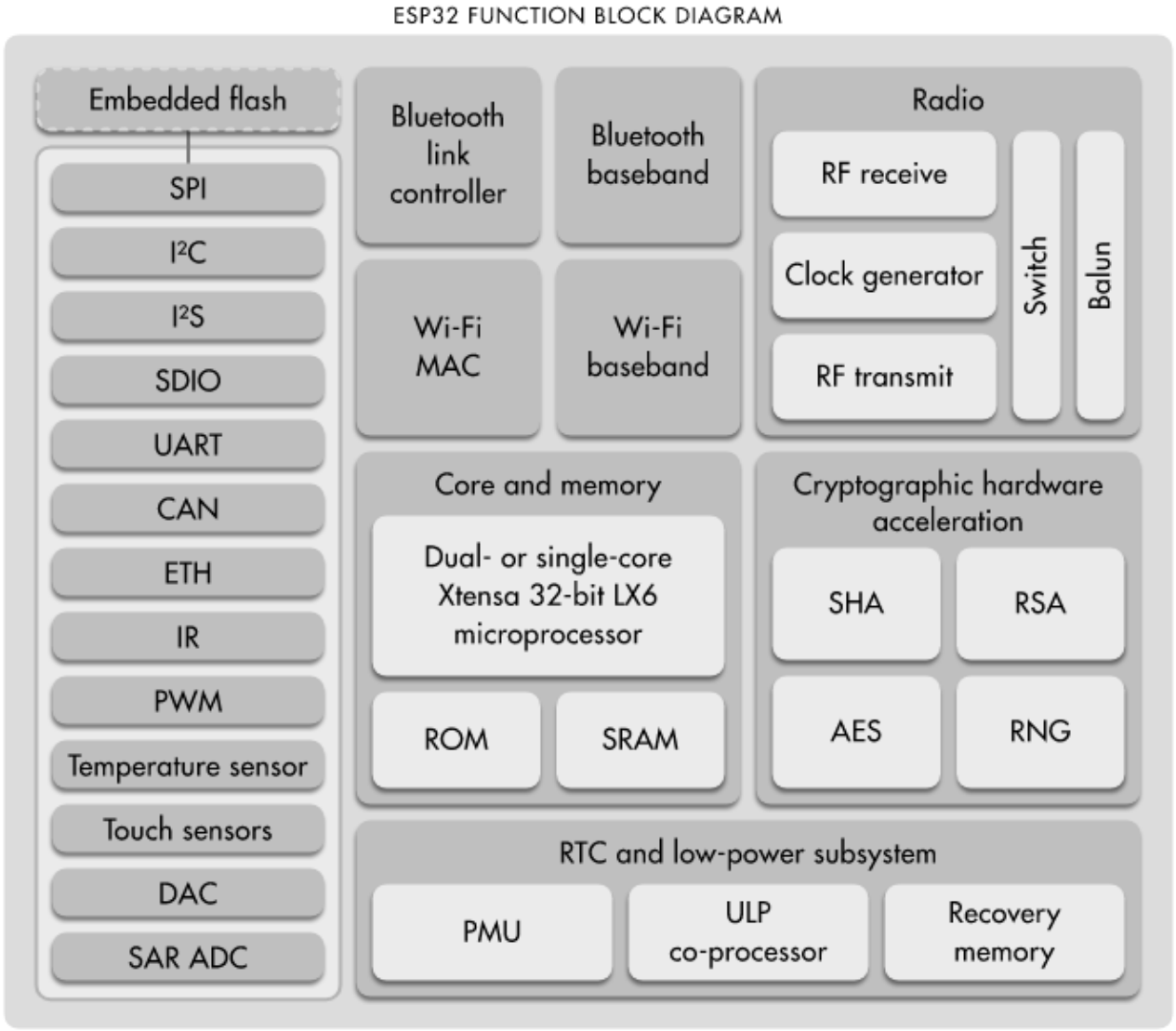
Aplicações X Tecnologias IoT

Key IoT Verticals	LPWAN (Star)	Cellular (Star)	Zigbee (Mostly Mesh)	BLE (Star & Mesh)	Wi-Fi (Star & Mesh)	RFID (Point-to-point)
Industrial IoT	●	○	○			
Smart Meter	●					
Smart City	●					
Smart Building	●		○	○		
Smart Home			●	●	●	
Wearables	○			●		
Connected Car					○	
Connected Health		●		●		
Smart Retail		○		●	○	●
Logistics & Asset Tracking	○	●				●
Smart Agriculture	●					

● Highly applicable ○ Moderately applicable

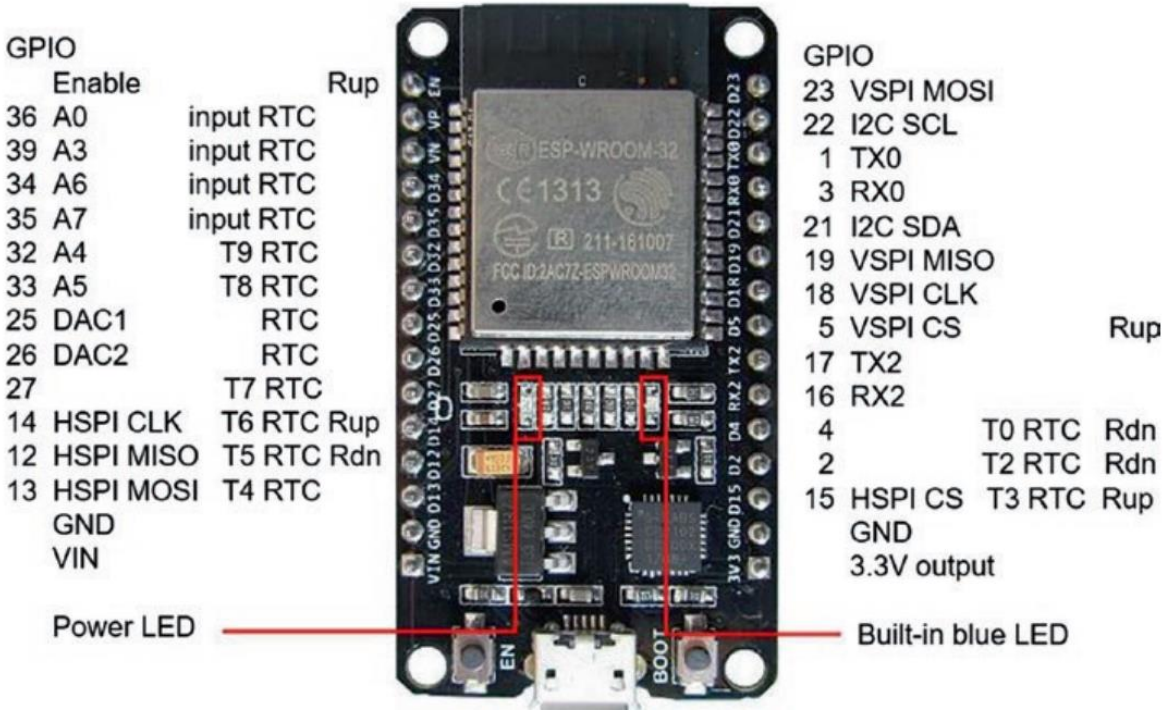
https://behrtech.com/wp-content/uploads/2020/11/IoT-Wireless-Technologies-Applications.jpg

ESP32

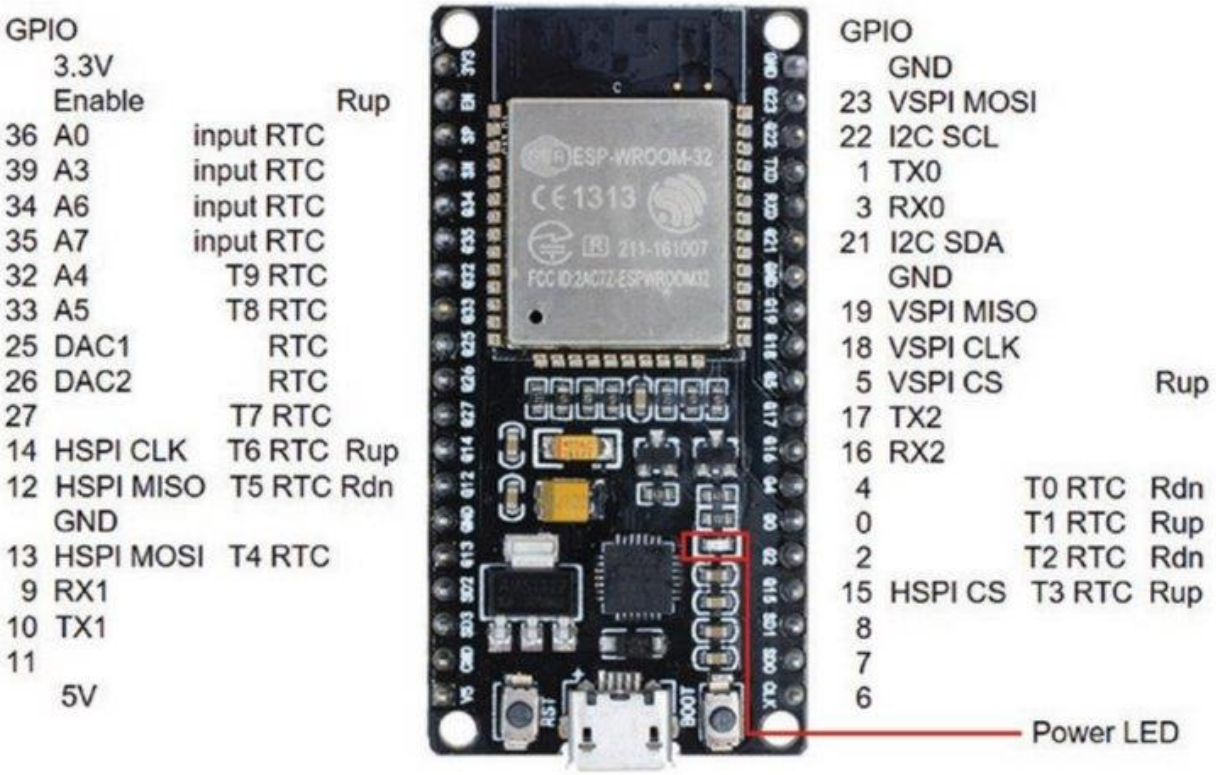


http://electronupdate.blogspot.com/2018/08/espressif-esp32-teardown.html

Kits ESP32



30 pinos



38 pinos

Plataformas de Desenvolvimento

Arduino: facilidade na utilização, menor flexibilidade de desenvolvimento comparado ao ESP-IDF

ESP-IDF : Espressif IoT Development Framework, plataforma oficial

PlatformIo: é uma boa opção entre facilidade e flexibilidade de desenvolvimento.

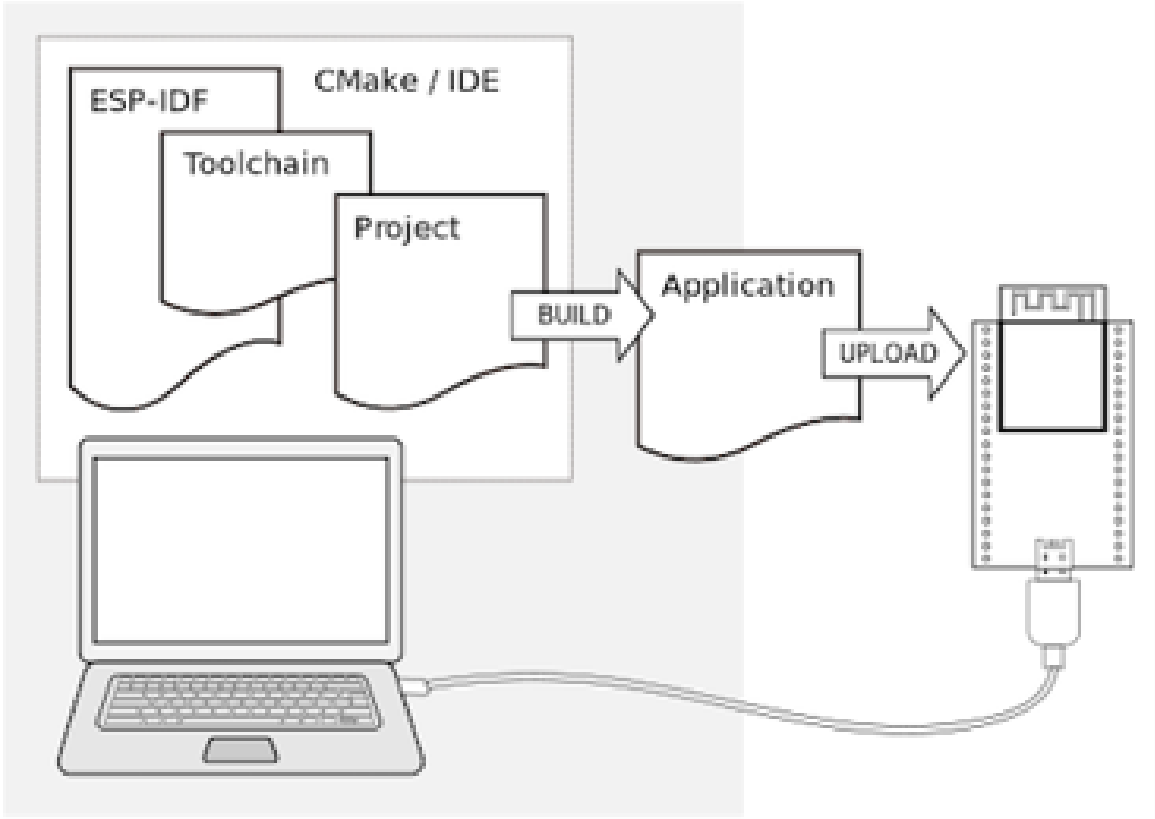
https://support.content.office.net/pt-br/media/8b7c7670-c3b8-4506-ac1f-a70e4d81b55c.png

https://espressif.gallerycdn.vsassets.io/extensions/espressif/esp-idf-extension/1.4.0/1646139386704/Microsoft.VisualStudio.Services.Icons.Default

https://cdn.platformio.org/images/platformio-logo.17fdc3bc.png

Utilizando ESP-IDF

- Necessário instalar:
  - *Toolchain*: compilar o código
  - Ferramentas de build: CMake e Ninja para realizar o *build*
  - ESP-IDF: bibliotecas de software e códigos fonte



https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html

CMAKE

- Gerencia o processo de build de maneira independente do compilador
- Arquivos CmakeList.txt são usados para gerar arquivos de compilação
- Quando CMake é executado, localiza arquivos, bibliotecas e executáveis
- https://cmake.org/cmake/help/book/mastering-cmake/

https://cmake.org/overview/

Curso CMake

This CMake training covers how to efficiently write CMake scripts for small to larger projects along with best practices. This training also provides an overview of unit testing with CTest, packaging with CPack and continuous integration with CDash.

This training mixes theory and application with a set of tutorials and exercises.

Objectives

- Understand the basics of CMake (variables, cache, flow, etc.)
- Learn the best practice of Modern CMake
- Configure simple and complex projects with CMake
- Learn about the companion CMake tools: CPack, CTest and CDash

Prerequisites

- C/C++ : Basic knowledge

Practical Information

Date: April 25-26, 2022 - 1pm to 5pm EST
Duration: 8h over 2 afternoons (1-5pm)
Location: Remote
Language: English
Price: $950

REGISTER Need Assistance?

Company Training

All our training can be offered on site with a custom agenda. Contact us for more information.

CONTACT US

https://www.kitware.com/courses/cmake-training/

CMakeLists.txt projeto blink

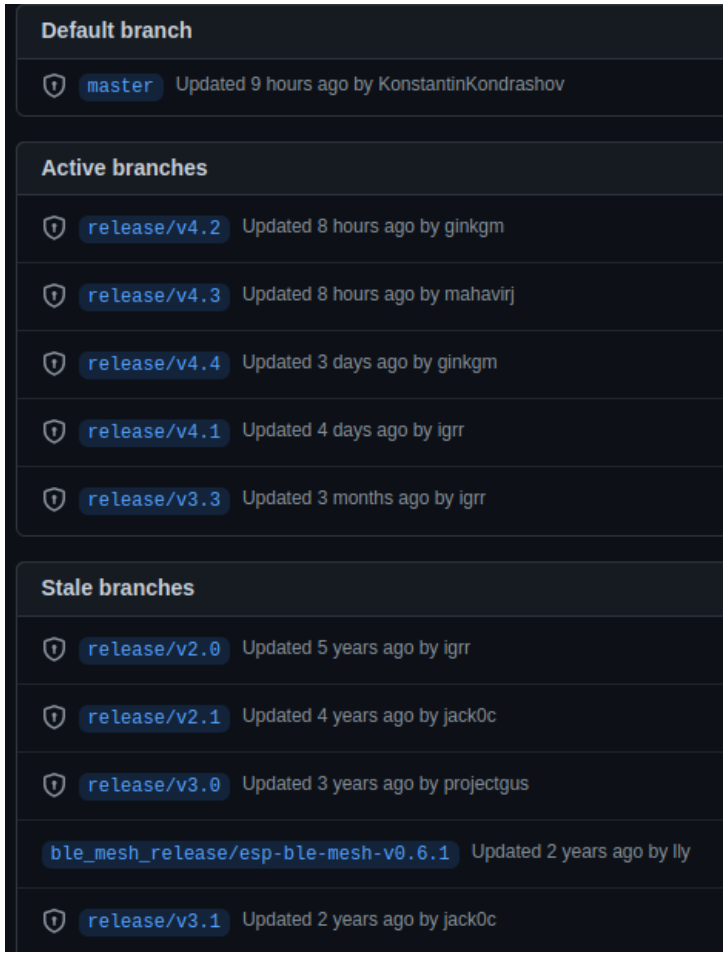
#/blink/CMakeLists.txt

```
# The following five lines of boilerplate have to be in your project's
# CMakeLists in this exact order for cmake to work correctly
cmake_minimum_required(VERSION 3.5)

include($ENV{IDF_PATH}/tools/cmake/project.cmake)
project(blink)
```

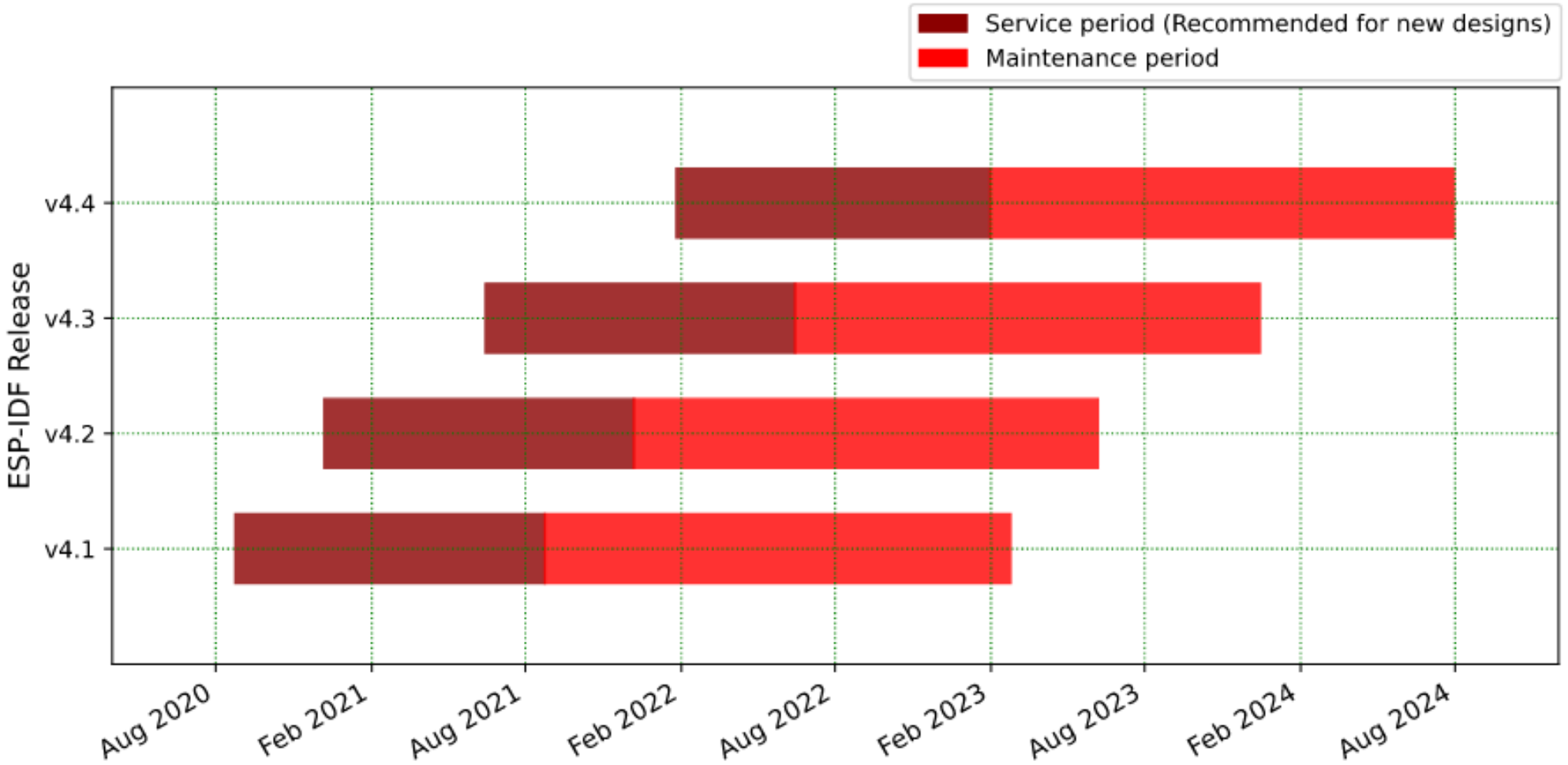
#/blink/main/CMakeLists.txt

```
idf_component_register(SRCS "blink.c"
INCLUDE_DIRS ".")
```

Github ESP-IDF



Período de Suporte ESP-IDF



https://camo.githubusercontent.com/916b474a6b33b73c38b6706fa51aed57a0bb334272df3b09703fc7d88a4daad8/68747470733a2f2f646c2e657370726573369662e636f6d2f646c2f6573702d6964662f737570706f72742d706572696f64732e737667

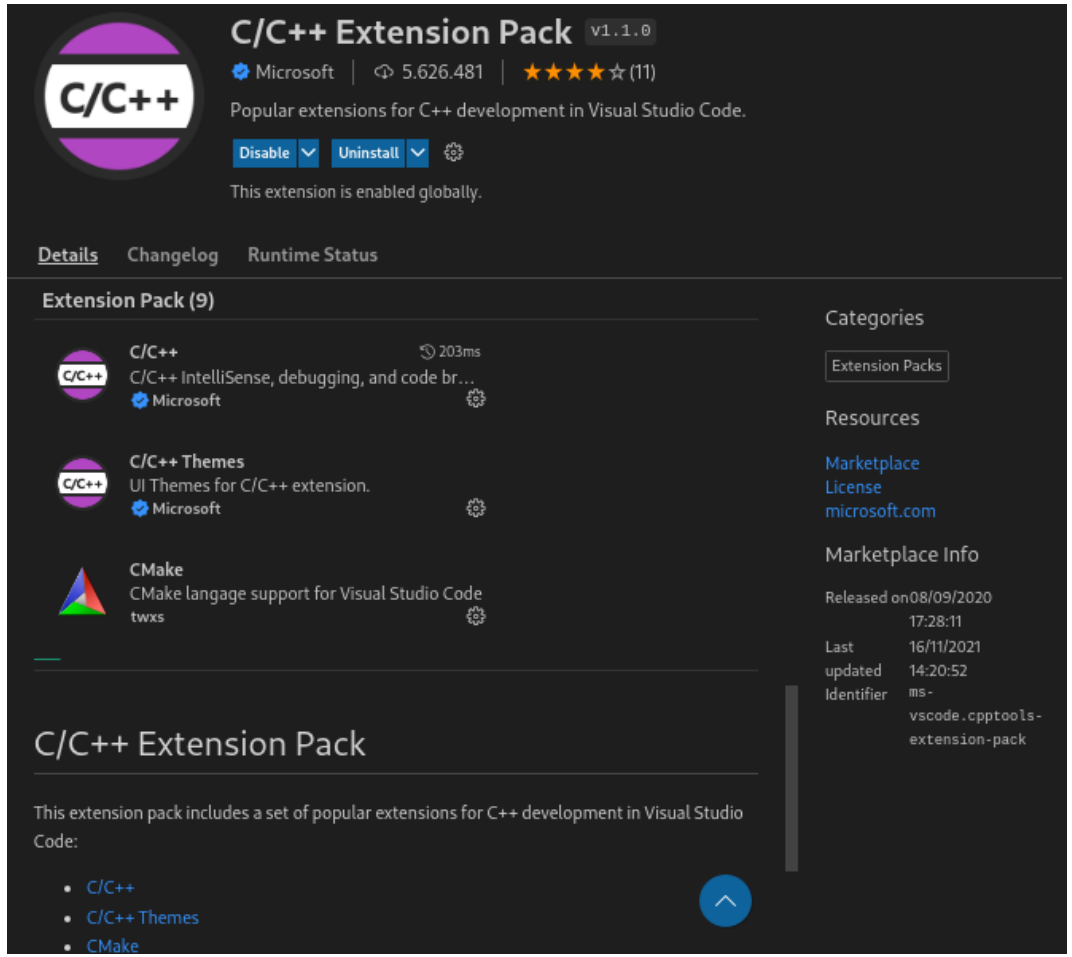
Instalando pré-requisitos

- Abra um terminal

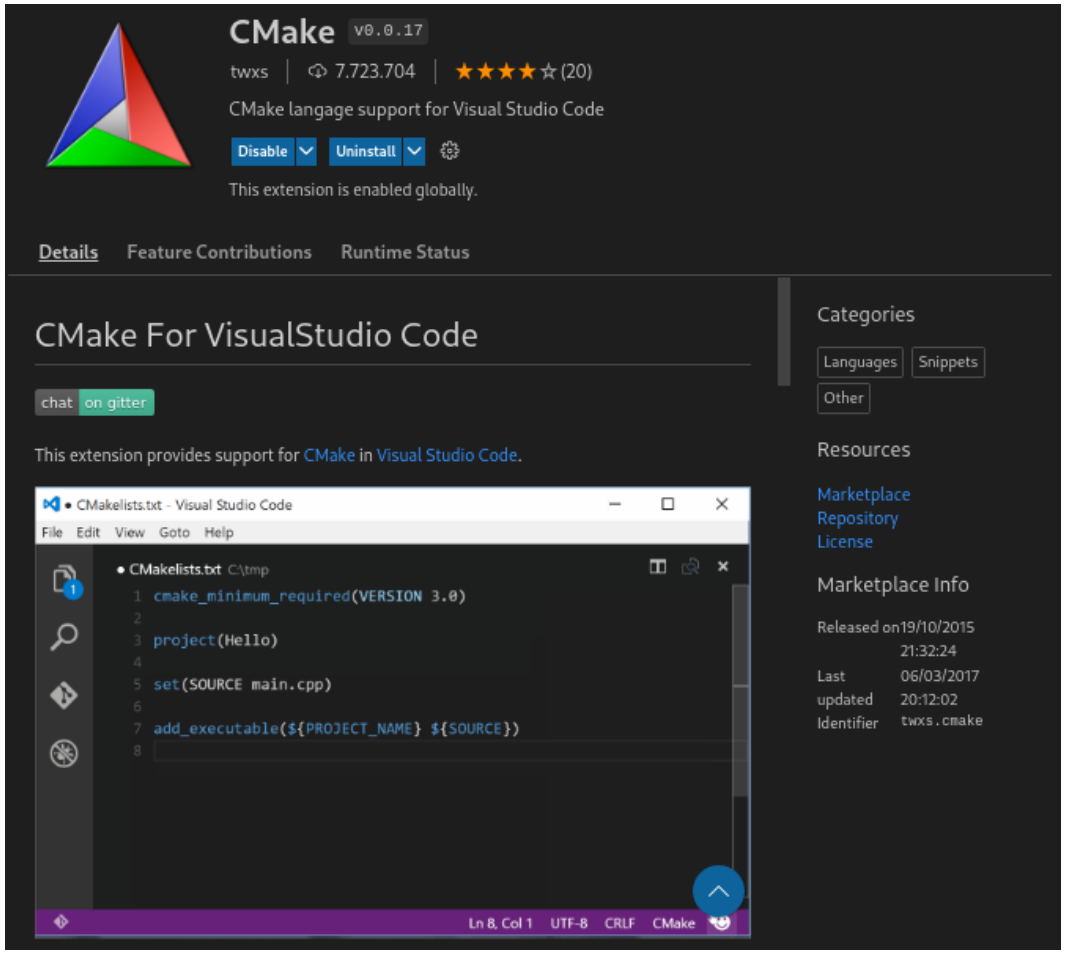
```
$ sudo apt-get install git wget flex bison gperf python3 python3-pip python3-setuptools cmake ninja-build ccache libffi-dev libssl-dev dfu-util libusb-1.0-0
```

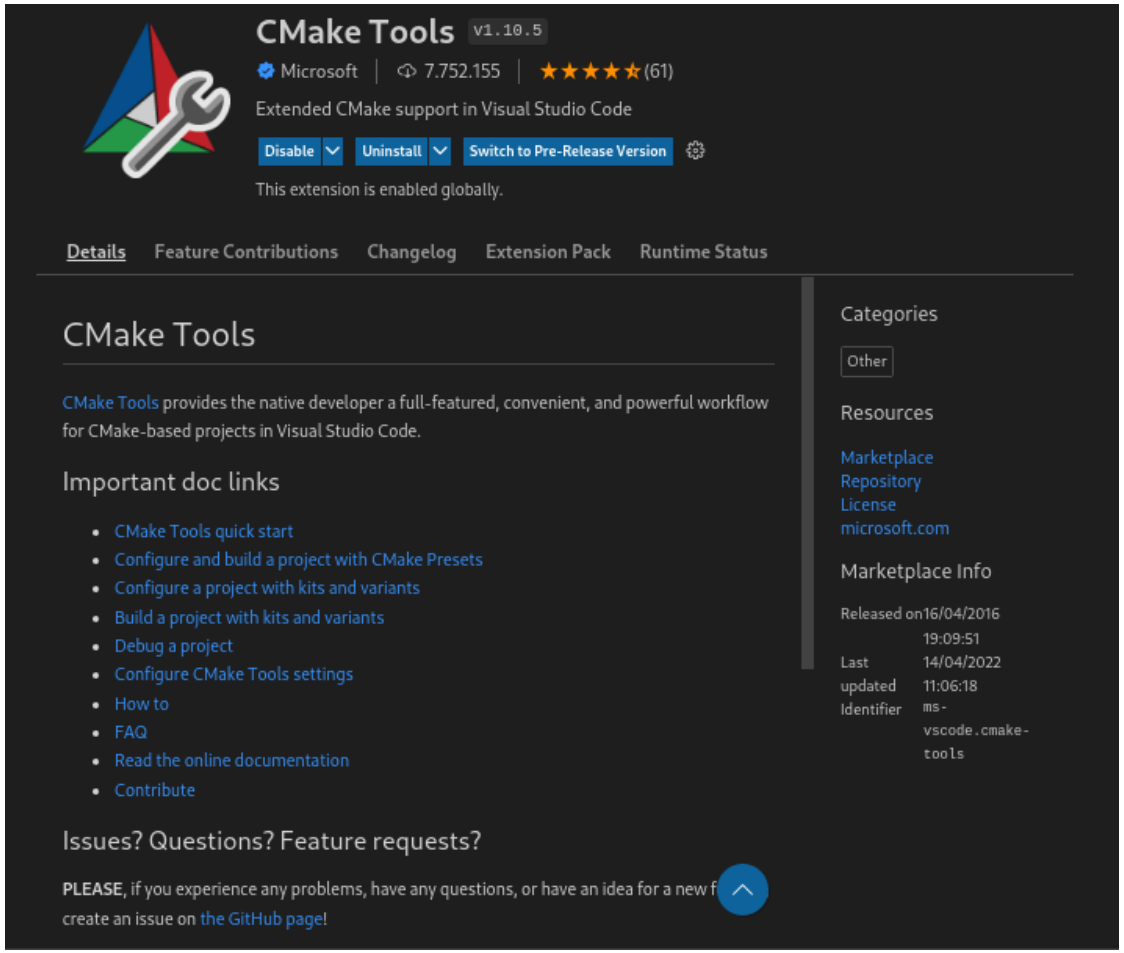
https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/linux-macos-setup.html

Instalando pré-requisitos (cont.)



Instalando pré-requisitos (cont.)





Instalação

- Faça o fork do repositório
  - https://github.com/liukiti/esp32
- Clone o repositório ESP32 de sua conta

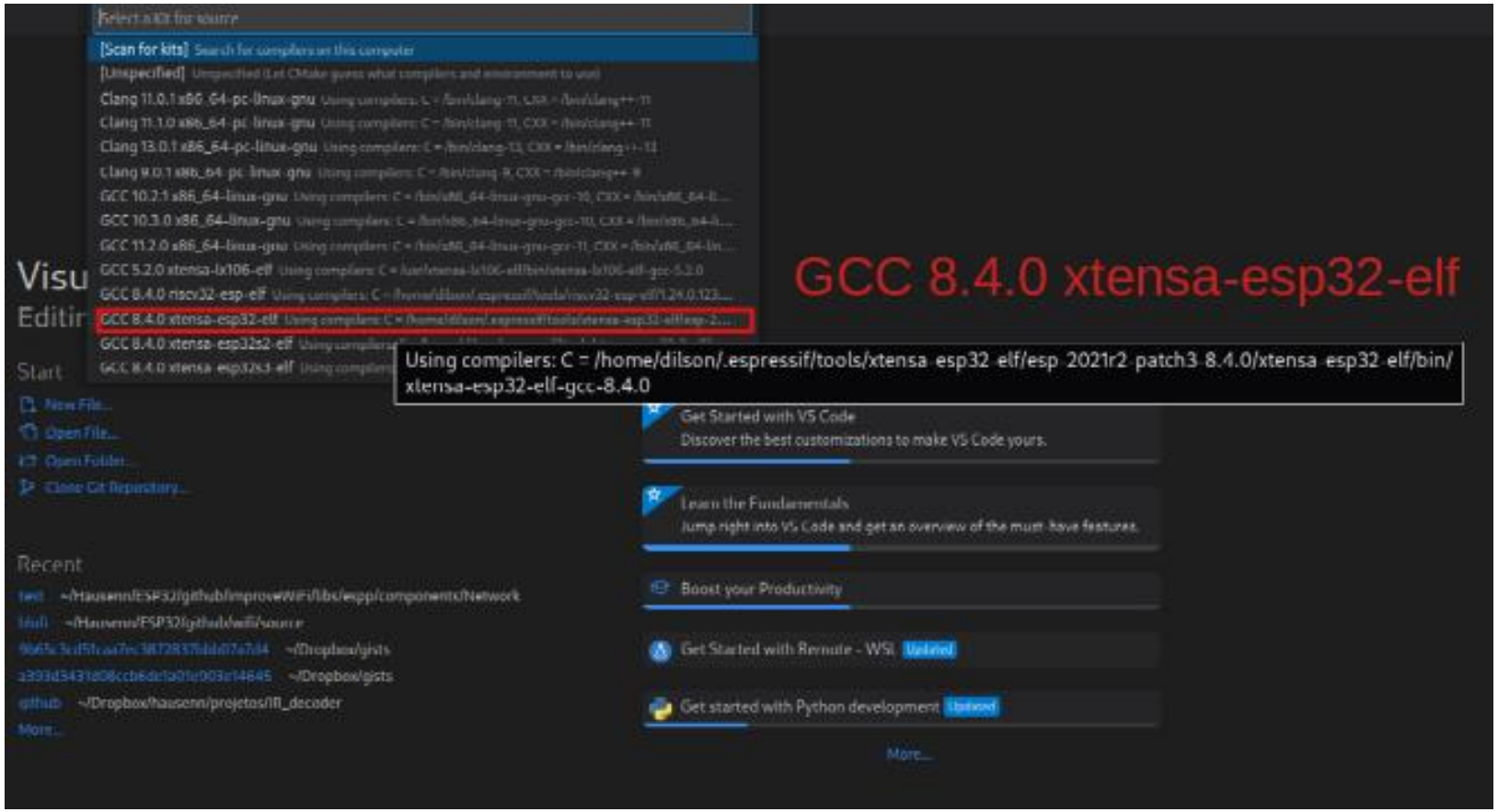
```
$ git clone git@github.com:USUARIO/ESP32.git

$ cd ESP32
$ git submodule update --init --recursive

$ cd esp-idf
$ ./install.sh esp32
```

Instalação (cont.)

- Navegue até /ESP32/esp-idf/examples/get-started/
- Copie a pasta blink para o ESP32/myCodes/

- Navegue até /ESP32/
- Dê um duplo clique em edit.desktop

Configurar CMake



Build do projeto blink

- Abra um terminal no VSCode para o projeto blink
- Execute

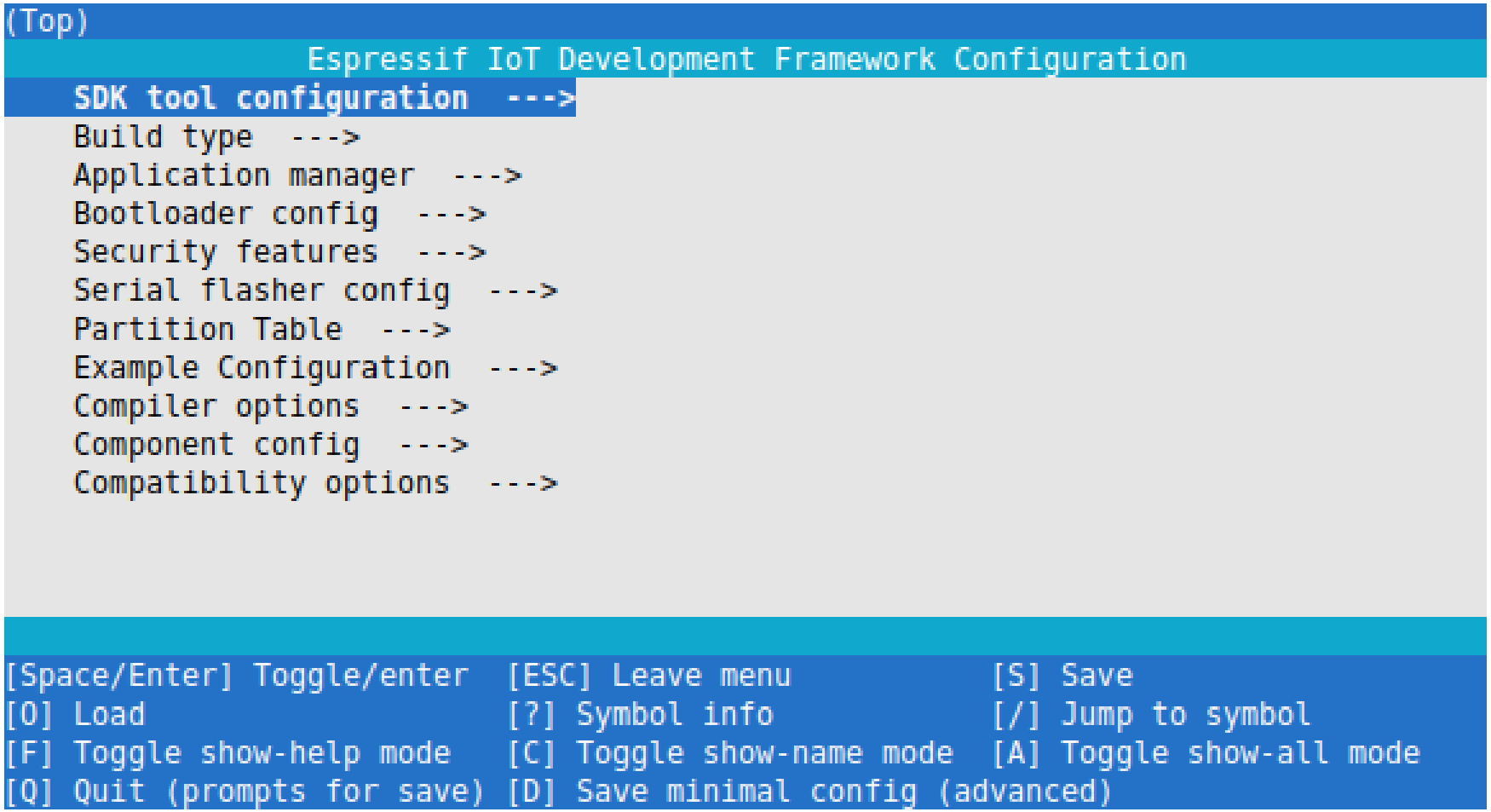
```
$ idf.py all
```

idf.py

- $ idf.py monitor -p /dev/ttyUSB*

- $ idf.py fullclean

idf.py menuconfig

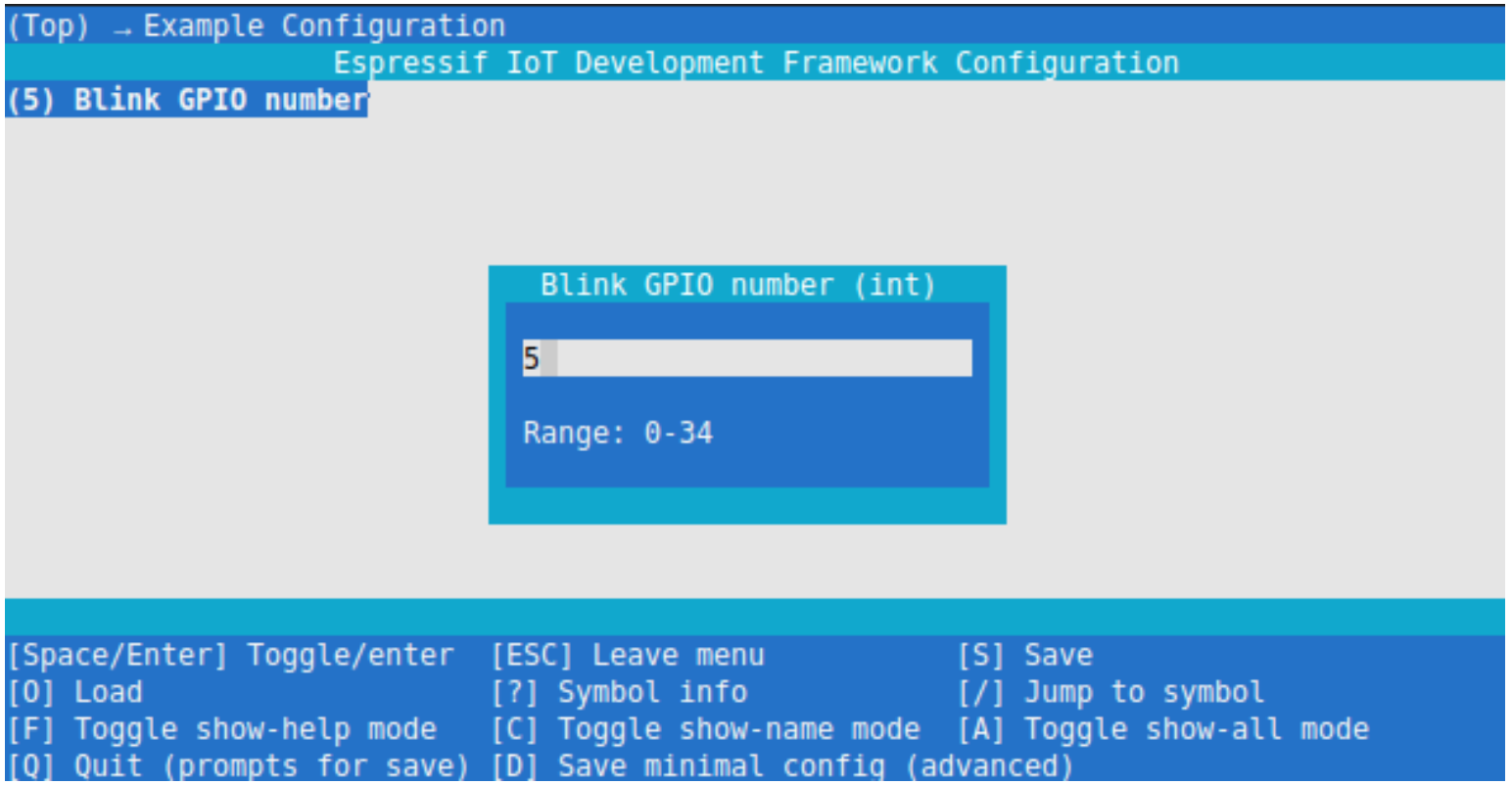


Kconfig

- Setar configurações do projeto

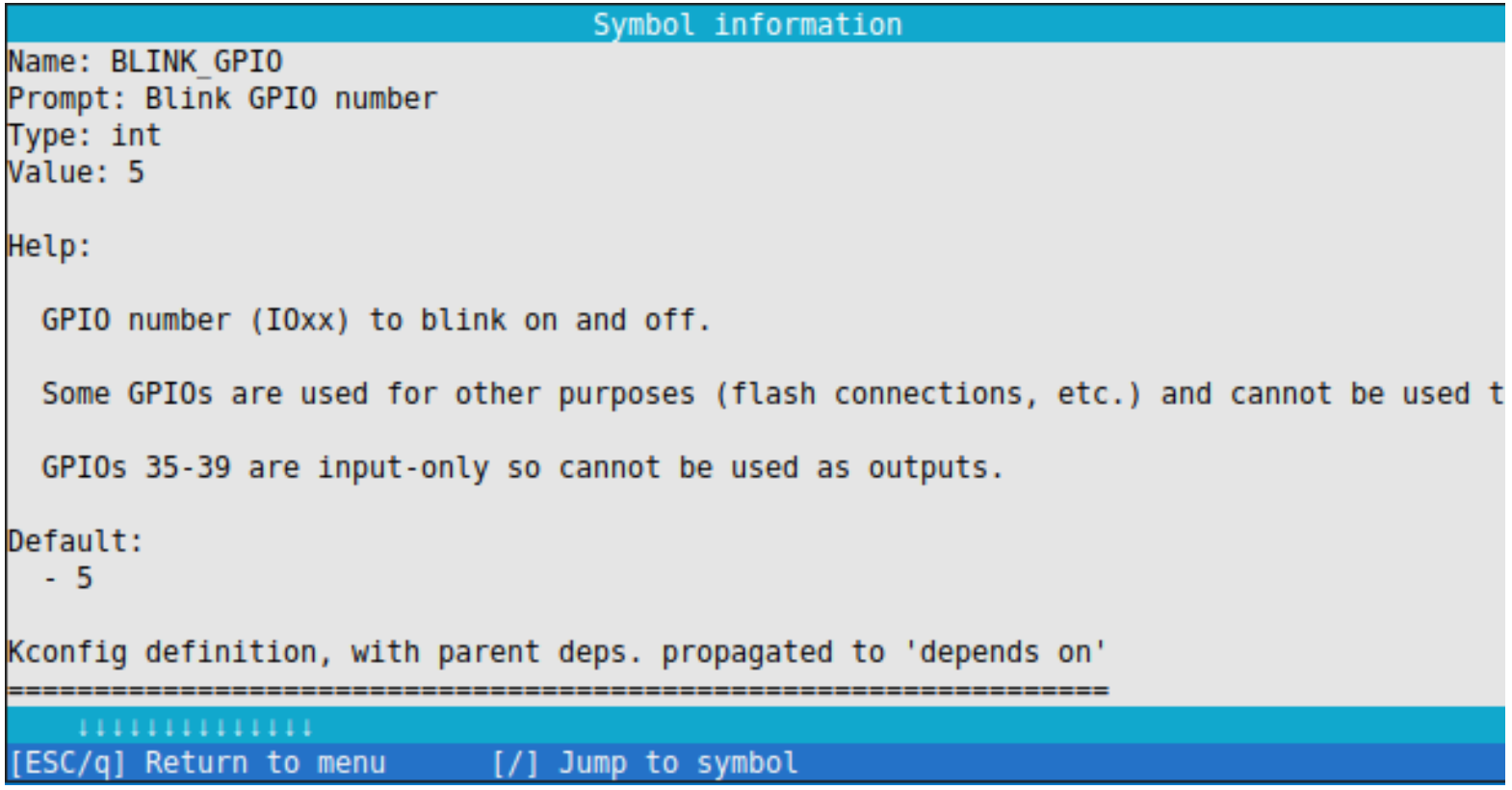
- https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html

```
menu "Example Configuration"
  config BLINK_GPIO
    int "Blink GPIO number"
    range 0 34
    default 5
    help
     GPIO number (IOxx) to blink on and off.

     Some GPIOs are used for other purposes
    (flash connections, etc.) and cannot be
     used to blink.

     GPIOs 35-39 are input-only so cannot
     be used as outputs.

endmenu
```



Kconfig (cont.)

```
menu "Example Configuration"
  config BLINK_GPIO
    int "Blink GPIO number"
    range 0 34
    default 5
    help
     GPIO number (IOxx) to blink on and off.

     Some GPIOs are used for other purposes
    (flash connections, etc.) and cannot be
     used to blink.

     GPIOs 35-39 are input-only so cannot
     be used as outputs.

endmenu
```



edit.desktop


```
#!/usr/bin/env xdg-open
[Desktop Entry]
Name=Edit Project
Comment=Edit project source code
Exec=bash -c 'cd "$(dirname "$1")/tools" && ./vscode.sh' dummy %k
Icon=/usr/share/pixmaps/com.visualstudio.code.png
Terminal=false
Type=Application
```

.clang-format

- Ctrl + Shift + I

```
{
  BasedOnStyle: LLVM,
  IndentWidth: 4,
  TabWidth: 4,
  UseTab: Never,
  BreakBeforeBraces: Allman,
  AllowShortIfStatementsOnASingleLine: false,
  AllowShortFunctionsOnASingleLine: false,
  AllowShortBlocksOnASingleLine: true,
  IndentCaseLabels: false,
  ColumnLimit: 80,
  AccessModifierOffset: -4,
  SortIncludes: false,
}
```

Recomendados

- Kolban's Book on ESP32
  - https://leanpub.com/kolban-ESP32
- Andreas Spiess
  - https://www.youtube.com/playlist?list=PL3XBzmAj53RnZPeWe799F-uoXERBldhn9
- Projetos com ESP32
  - https://hackaday.com/tag/ESP32/
  - https://www.hackster.io/search?i=projects&q=ESP32
  - https://www.instructables.com/howto/ESP32/

PADO

Labs