Arduino no ESP-IDF: simplificando o desenvolvimento com flexibilidade

Rodrigo Garcia

Software Platforms Department

Tópicos

1. ESP32 Arduino Core ⇒ IDF
2. Prática - Hands On
3. Q/A

ESP32 Arduino Core ⇒ IDF

ESP32 Arduino é…

Uma camada de software codificada em cima do SDK ESP-IDF.

Vantagens:

- Compatível com todos os SoC existentes.
- Simplifica e acelera a prototipação de aplicações.
- Total flexibilidade: arduino-CLI, IDF Component, IDEs
- Bibliotecas em Arduino API ou em C/C++ com IDF puro.
- Alto nível de abstração com Bibliotecas e Exemplos.
- 1 versão Arduino Core para cada versão de IDF.
- Conversão para IDF via camada Arduino HAL.

Arduino

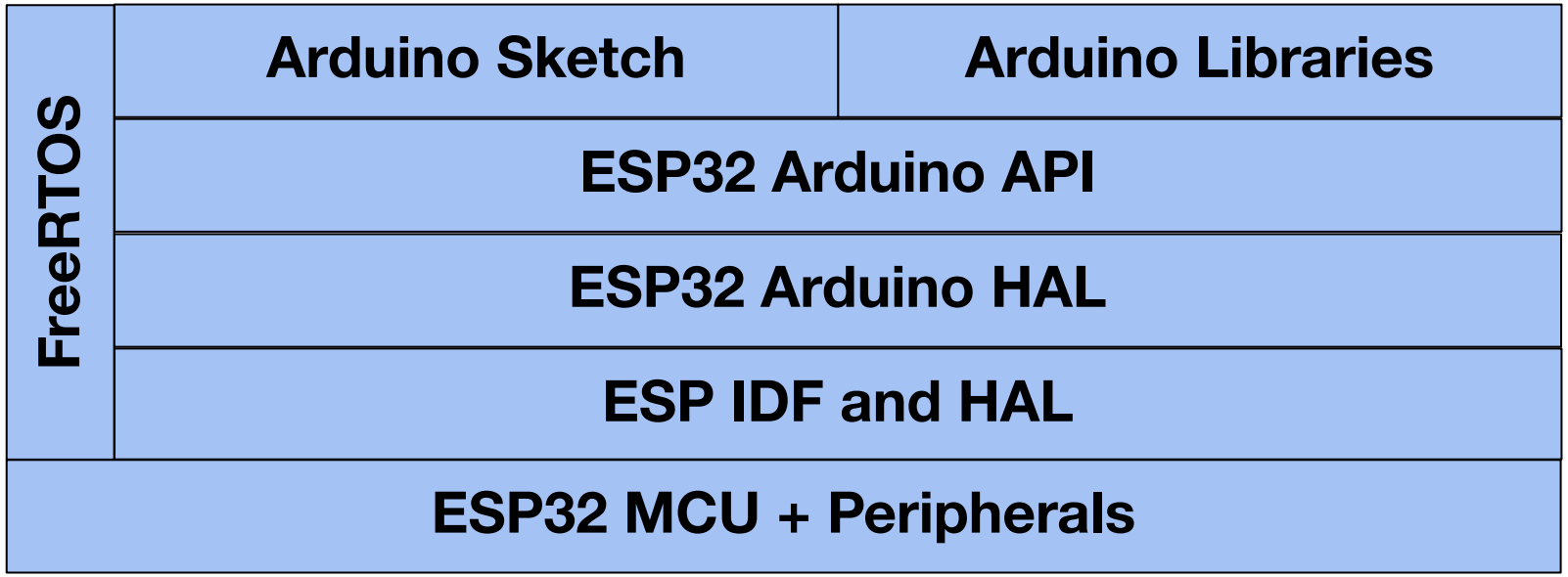
Principal uso: Prototipação, Projetos / PoC em Geral.
Licença: GNU General Public License (GPL)
[support.arduino.cc Licensing-for-products-based-on-Arduino]

```
// define LED_PIN 13
int LED_PIN = 13;

void setup () {
    pinMode (LED_PIN, OUTPUT);      // habilita o pino 13 para saída digital (OUTPUT).
}

void loop () {
    digitalWrite (LED_PIN, HIGH);  // liga o LED.
    delay (1000);                  // espera 1 segundo (1000 milissegundos).
    digitalWrite (LED_PIN, LOW);   // desliga o LED.
    delay (1000);                  // espera 1 segundo.
}
```

ESP32 Arduino

SoCs:

- v1.x - ESP32
- v2.x - ESP32-S2, ESP32-C3 e ESP32-S3
- v3.x - ESP32-C6, ESP32-H2, ESP32-P4, ESP32-C61, ESP32-H4, ESP32-C2



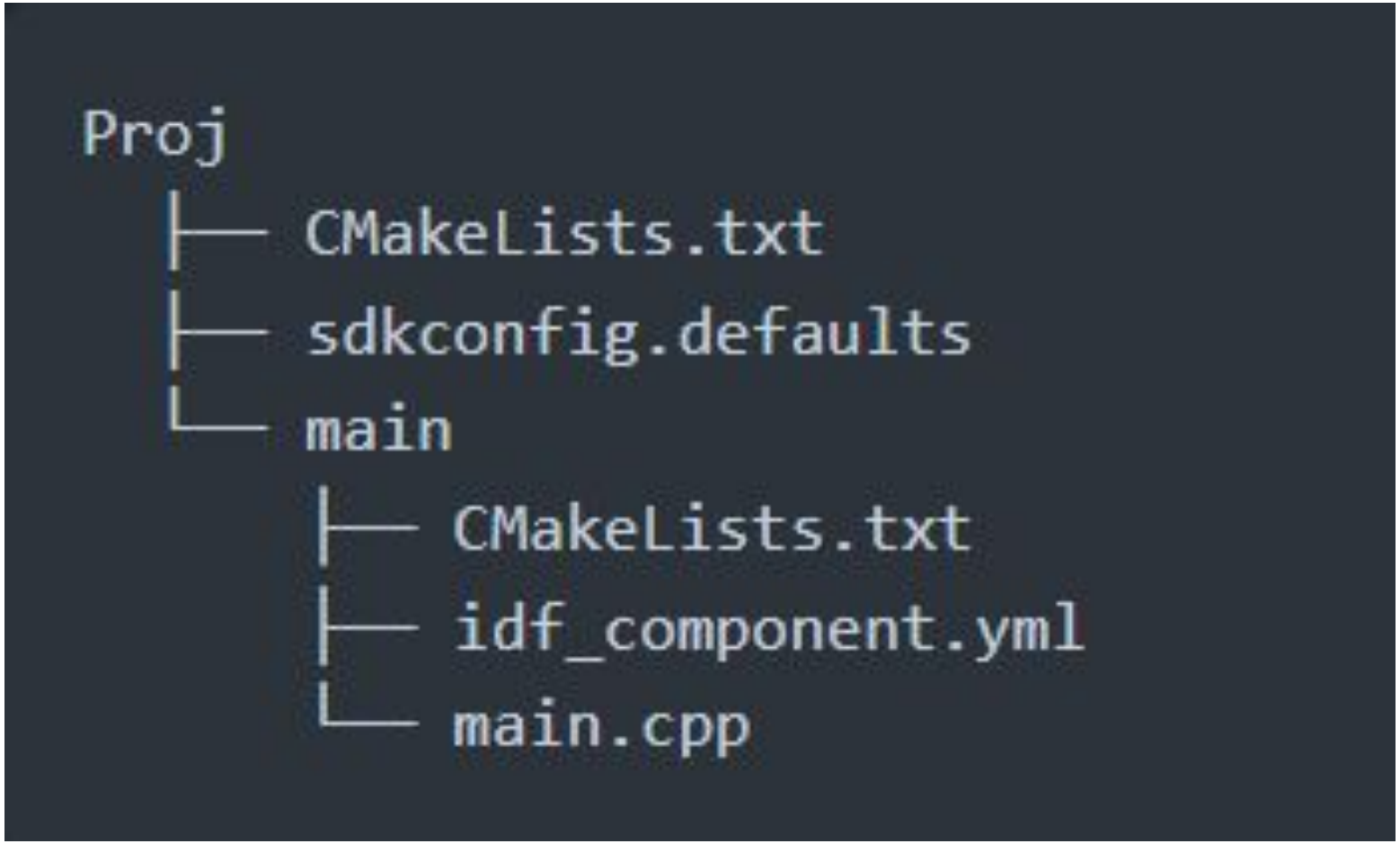
Arduino como Componente IDF

Versões do Arduino x versões do IDF

Informações:
https://github.com/espressif/arduino-esp32/releases

Arduino Core 3.3.0 usa IDF v5.5
Arduino Core 3.2.0 .. 3.2.1 usa IDF v5.4.1 a v5.4.2
Arduino Core 3.1.0 .. 3.1.3 usa IDF v5.3.2 a v5.3.3
Arduino Core 3.0.0 .. 3.0.7 usa IDF v5.1.4 a v5.1.4+
Arduino Core 2.0.1 .. 2.0.17 usa IDF v4.4 a v4.4.7

Estrutura de um Projeto IDF



Adicionando Componentes

```
├── CMakeLists.txt          Arduino Project Description and Settings
├── sdkconfig.defaults      Arduino and IDF Project Settings
├── main
│   ├── CMakeLists.txt      Arduino Sketch Description
│   ├── idf_component.yml   ESP32 Arduino Core version and necessary components
│   └── main.cpp            Main Sketch, equivalent to a .ino file
├── components
│   ├── user_library_1
│   │   ├── CMakeLists.txt   This will describe the Lib_1 source code files
│   │   ├── ...              Regular Library_1 files
│   ├── user_library_2
│   │   ├── CMakeLists.txt   This will describe the Lib_2 source code files
│   │   ├── ...              Regular Library_2 files
```

Uso da ferramenta **idf.py**:

```
idf.py --version
idf.py --help
idf.py --list-targets
```

```
idf.py fullclean  || rmdir /s/q build || rm -rf build
idf.py set-target esp32s3 || del or rm sdkconfig
idf.py menuconfig
idf.py -p <COM Port | DEV File> flash monitor
```

Prática - Hands On

Hands ON! problemas…

1. VSCode + ESP IDF plugin com IDF 5.5
2. Apresentação da placa ESP32-C3 esp-rust-board v1.2
3. Exemplo IDF Blink LED - GPIO e LED endereçável
4. Mesmo exemplo usando agora Arduino 3.3.0
5. Usando a porta USB CDC
6. Componente IDF Button
7. Bibliotecas Arduino para Sensores
8. Exemplo Arduino com WiFi HTTP Client

Q&A