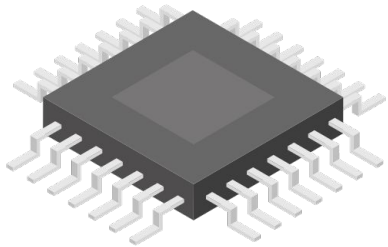
PADO

Labs

Microcontroladores



Prof.º: Pablo Jean Rozário

pablo.jean@padotec.com.br

/in/pablojeanrozario

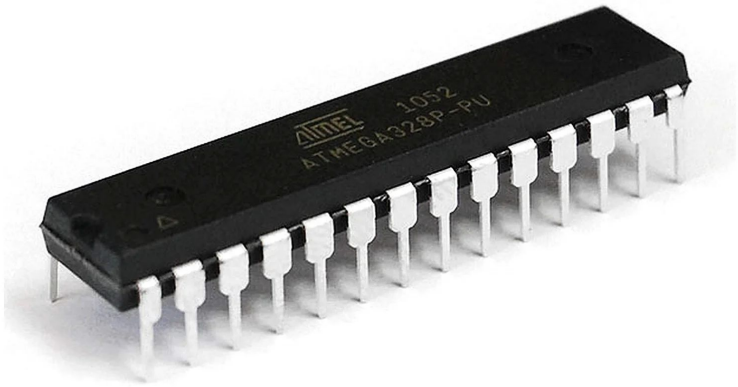
https://github.com/Pablo-Jean

Introdução aos Microcontroladores

Índice da Aula #1

- Exemplos de Aplicação
- O que são Microcontroladores
- Microcontroladores x Microprocessadores
- Arquitetura de Microcontroladores
- Registradores
- Program Counter e a Stack
- Set de Instruções
- Kit de Desenvolvimento
- STM32CubeIDE
- Lista de Exercícios #1

INTRODUÇÃO



Ementa

Aula	Descrição	Aula	Descrição
1	Apresentação de Conceitos Apresentação do kit STM32	7	Atividades
2	Tipos especiais de (u)int Registradores e Clock do uC Entradas e Saídas IOs	8	Comunicação Serial I²C
3	Estruturas em C Interrupções em IOs	9	Timers e RTC
4	Conversor A/D e Interrupções	10	PWM
5	Comunicação Serial UART	11	Desenvolvimento de Projetos
6	Comunicação Serial SPI	12	

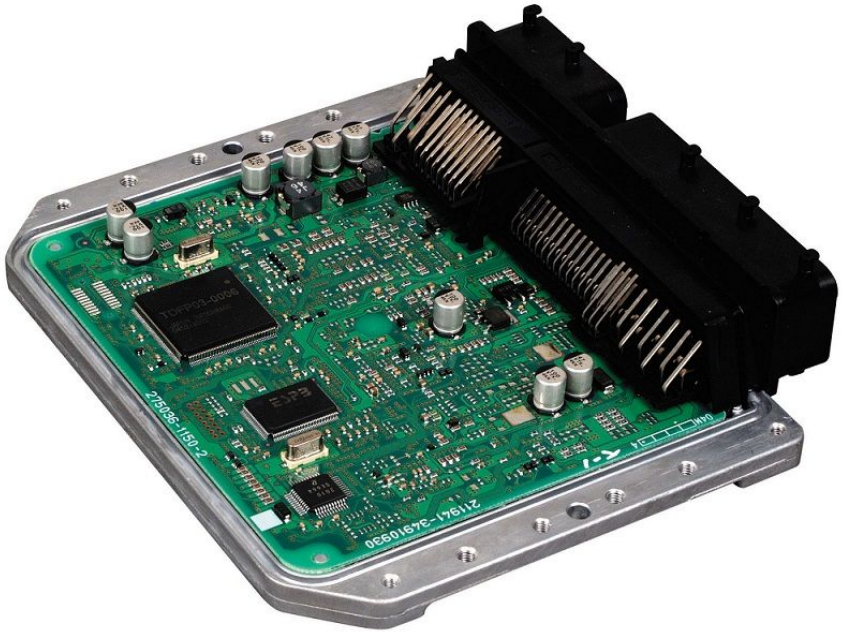
Exemplos de Aplicação

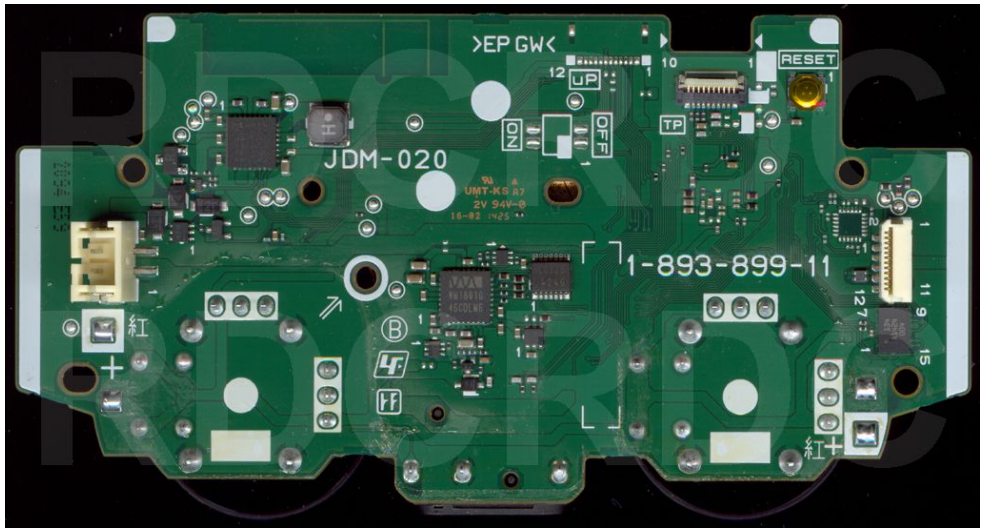
Como Exemplos de sistemas embarcados temos:

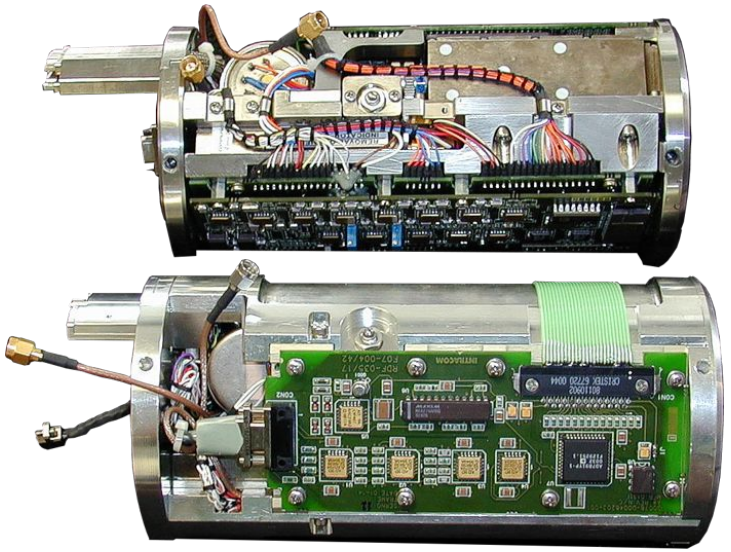
- Brinquedos eletrônicos;
- **Periféricos de computadores**: Mouse, teclado, controles de vídeo game;
- **Consoles**: Sony Playstation, Microsoft Xbox, Nintendo, etc;
- **Automotivo**: ECU, central multimídia, painel de instrumentos;
- **Eletrodomésticos**: Cafeteiras expressas, micro-ondas;
- **Bélico**: Armamentos, sistemas de controle;
- **Industrial**: CLPs, sensores;

Exemplos

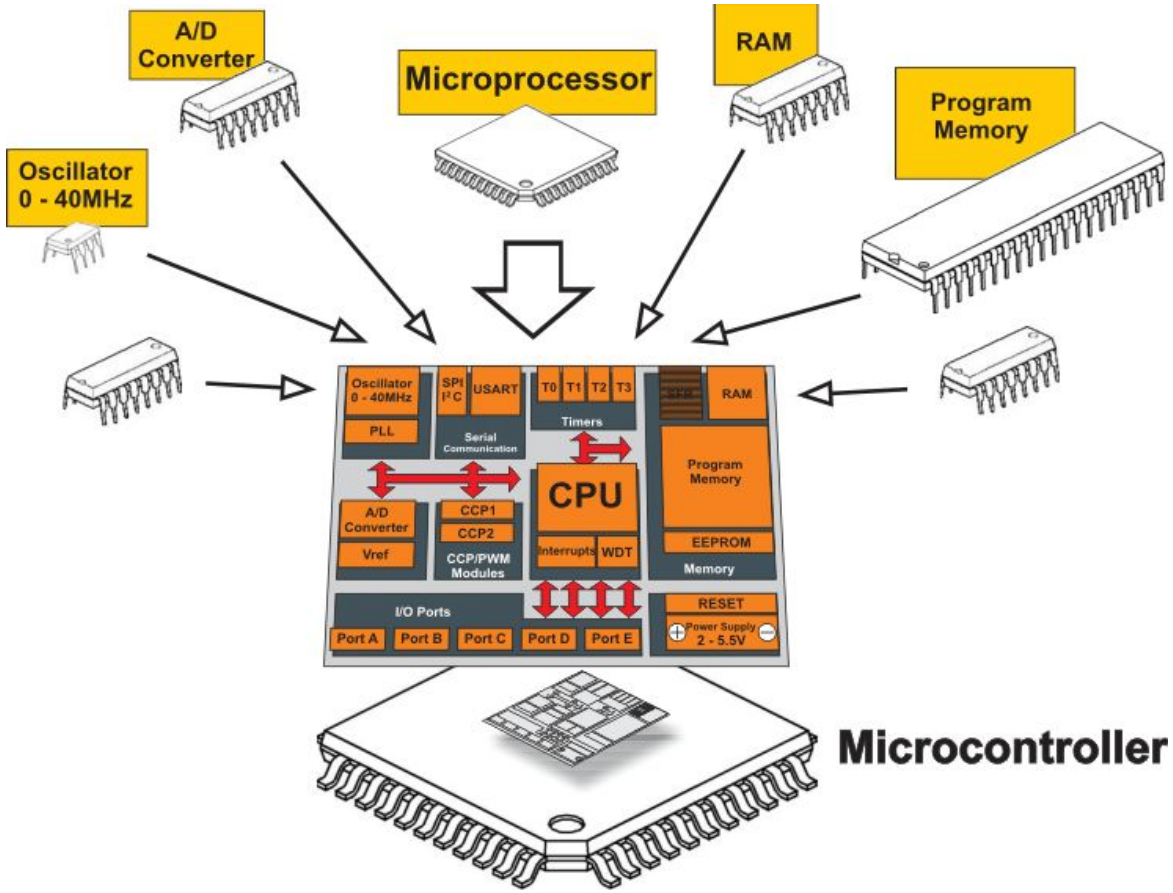
PADO Labs





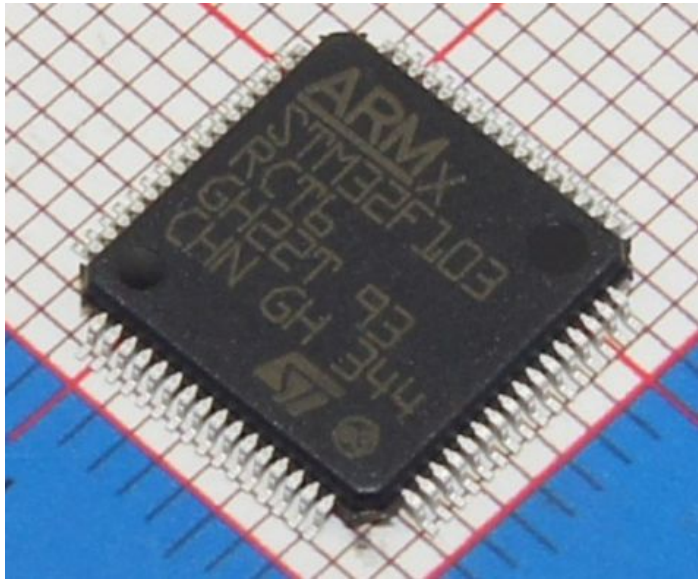


MICROCONTROLADORES

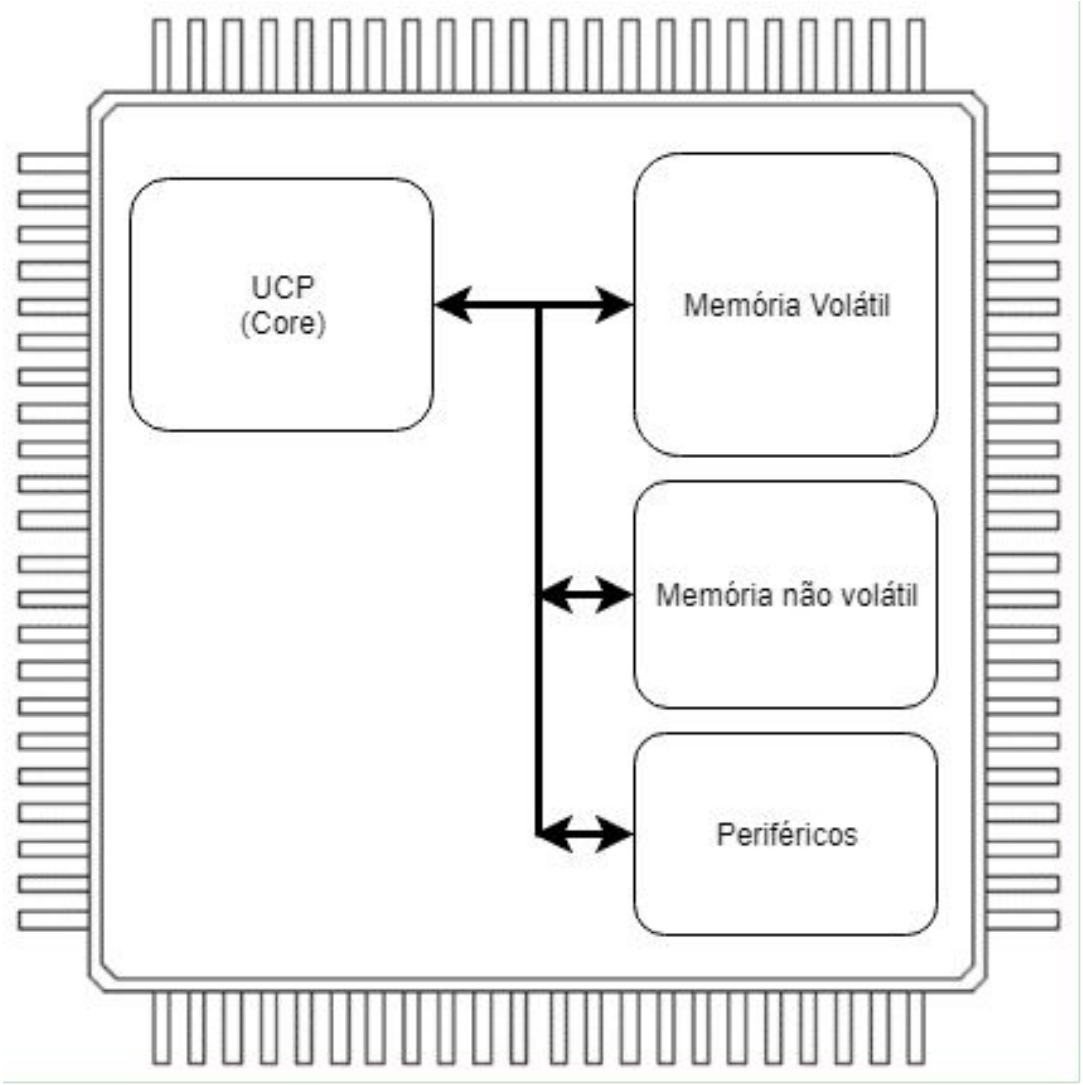


Microcontrolador - O Que É

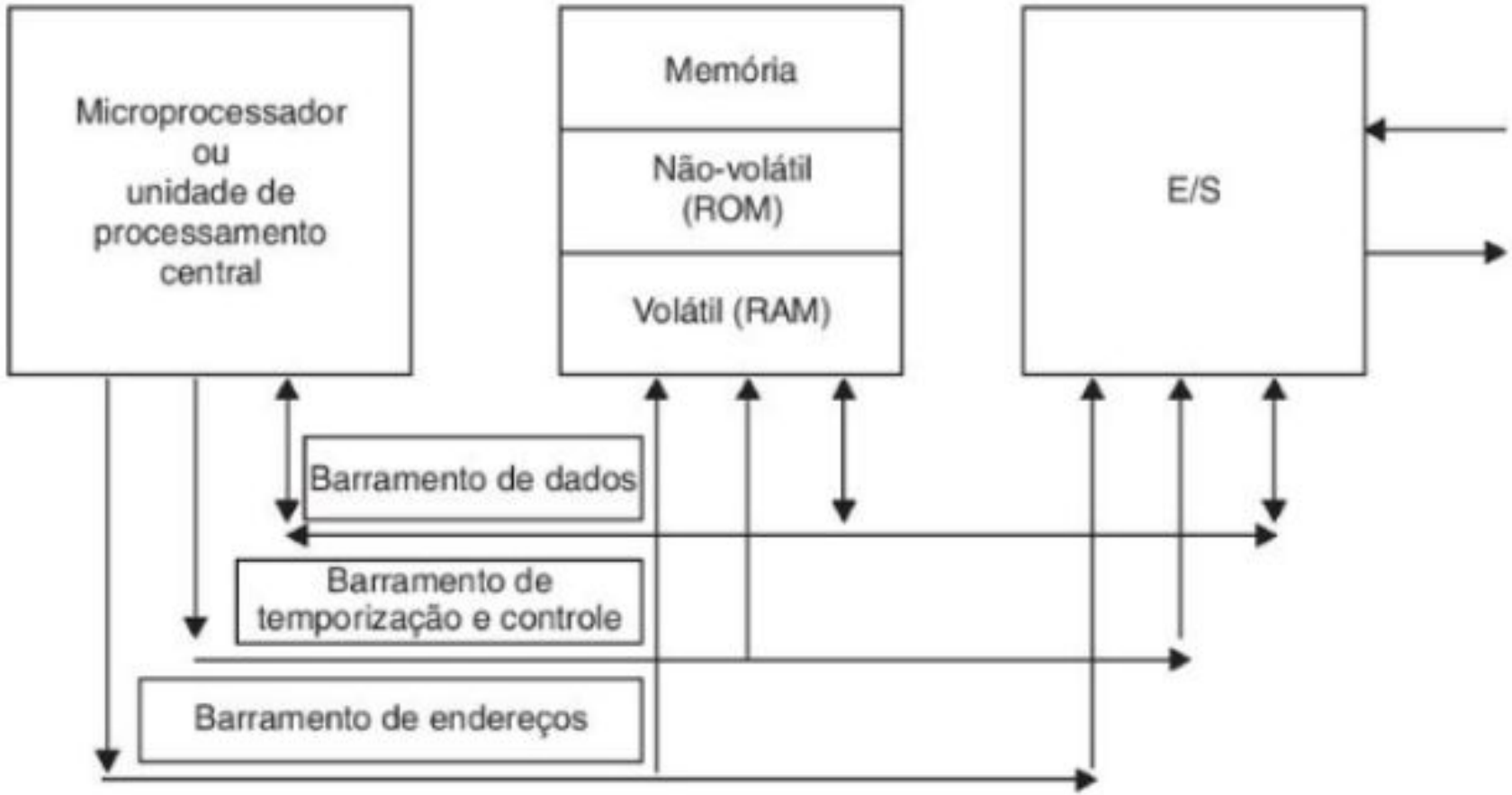
- Um microcontrolador é um componente eletrônico SoC (System on Chip), que possui capacidade de processamento, memórias, periféricos, interfaces de comunicação, entre outros.
- Em geral microcontroladores são muito menos potentes que computadores comuns.
- No entanto estas capacidades variam muito entre modelos e fabricantes.



Microcontroladores



Microcontroladores

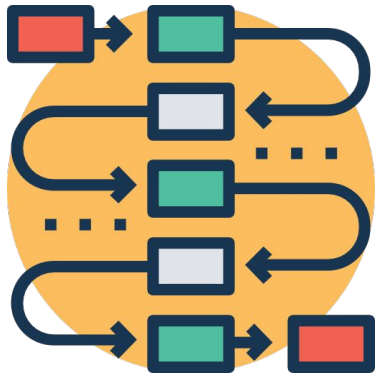


Microcontroladores - Estrutura

Barramento de Temporização e Controle : Define o dispositivo de IO no barramento de endereços por um período de tempo definido, além do sentido:

Barramento de endereços : Informa ao periférico qual endereço da informação que deseja ser alterado ou ler.

Barramento de Dados : Por onde é transferida a informação, para escrita ou leitura.



Microcontrolador ou Microprocessador

Característica	Microprocessadores	Microcontrolador
Periféricos	Necessita de periféricos externos	Periféricos integrados no chip
Memória	Permite vários formatos de dados	Poucos tipos de dados (8, 16 ou 32 bits)
Processamento	ALU Complexa e possui coprocessador	ALU limitada e ausência de coprocessamento
Custo	Custo elevado	Baixo custo, a depender da plataforma
Consumo	Alto consumo de enegeria	Possui métodos para economia de energia

Características

Os microcontroladores variam muito de modelos e fabricantes.

Com diferentes capacidades e diferentes periféricos.

Sempre de forma a atender determinados projetos, pois cada um possui suas peculiaridades.

É sempre importante levantar as necessidades do projeto.

Modelos de Microcontroladores

	PIC16F1824	MSP430FR2433	STM32G0B1RE	CC2642R
Fabricante	Microchip	Texas	ST	Texas
Core	PIC16	MSP430	ARM M0+	ARM M4F
Bits	8bits	16bits	32bits	32bits
Flash	7KB	15.5K	512KB	352KB
RAM	256B	4K	144KB	80KB
I/Os	12	19	60	31
ADC	10bits	10bits	12bits	12bits
SPI	X	X	X	X
I2C	X	X	X	X
UART	X	X	X	X
EEPROM	X	-	-	-
Extra	-	FRAM	USB OTG	Bluetooth

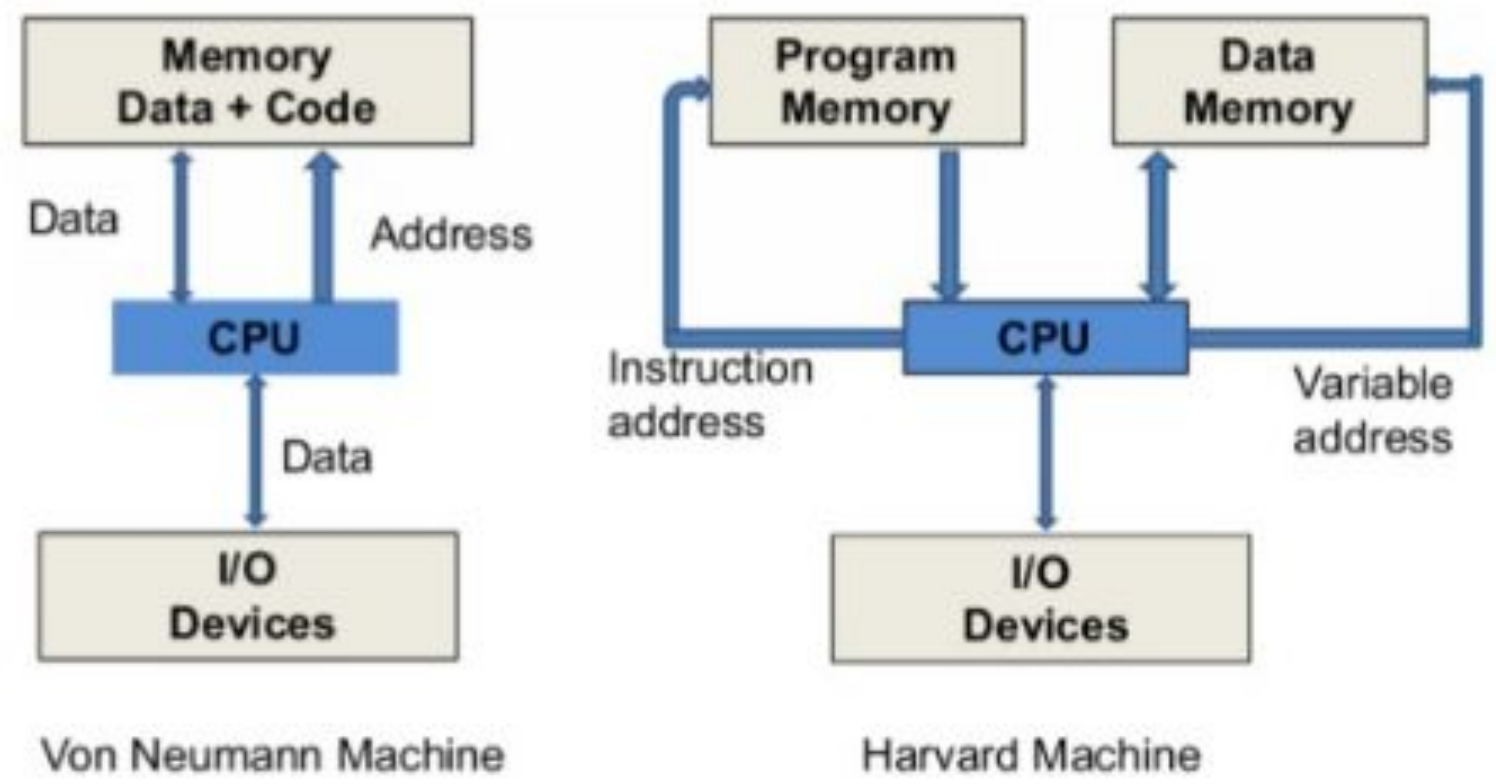
Arquiteturas

Existem duas principais arquiteturas base para os microcontroladores:

Von Neumann
Desenvolvida por John Von Neumann em 1945. Segue o conceito de armazenar instruções e dados em uma mesma memória.

Harvard
Desenvolvida para superar um gargalo causado pela Von Neumann, separa o barramento de dados do barramento de programa.

Von Neumann X Harvard



Von Neumann X Harvard

Von Neumann	Harvard
O mesmo endereço físico é utilizado para memória de programa e memória de dados	Diferentes endereços físicos são utilizados pelas memórias
O barramento da memória de programa e da memória de dados é compartilhado	O barramentos das memórias é isolado
É necessário pelo menos dois ciclos de clock para executar uma única instrução	Uma instrução pode ser executada em um ciclo de clock

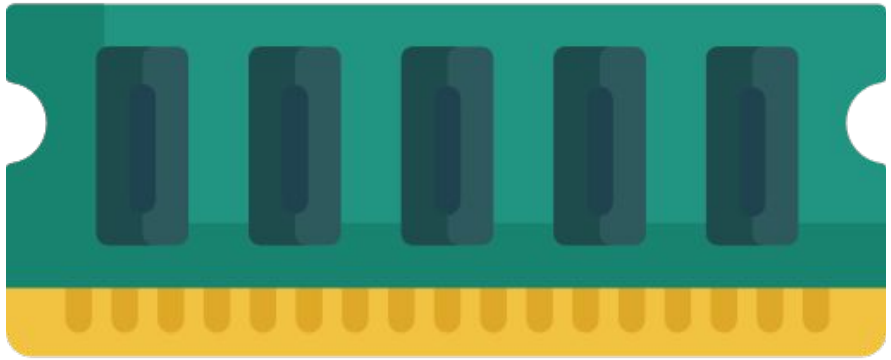
Von Neumann X Harvard

Von Neumann	Harvard
Possui custo menor	Possui um custo mais elevado se comparado a Von Neumann
Não é possível acessar uma instrução e ler/escrever ao mesmo tempo	E possível ler/escrever ao mesmo tempo que acessa uma instrução
Muito utilizado em computadores e microcontroladores ARM	Utilizado em microcontroladores e processamento de sinais

Registradores

São locais de memória onde é possível realizar operação de escrita e/ou leitura (*read/write*).

Como exemplo a memória RAM, que tem por característica altíssima velocidade e perde os dados ao ser desenergizado.



SFRs

Microcontroladores possuem um tipo especial de registradores, *Special Function Registers* (SFR), que são locais de memória que podem ser lidos e/ou escritos **mas** que estão conectados diretamente ao *hardware*.

Cada *bit* (ou mais) de um SFR é designado a uma função, podendo ser de dois tipos:

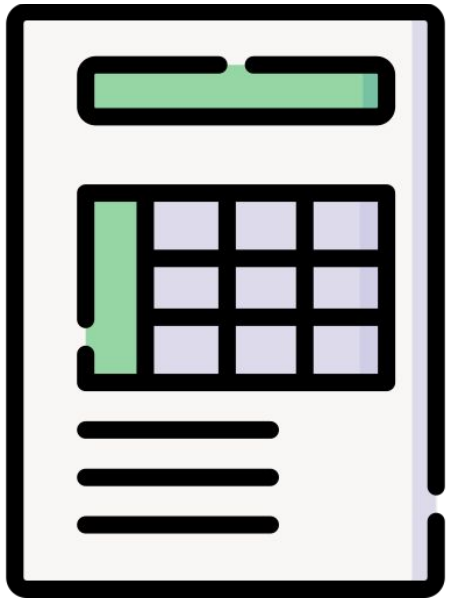
Control bit: acionam um determinado elemento de hardware

Flag bit: Controlado pelo hardware, indica status ou eventos.

SFRs

Todas as informações dos SRFs podem ser encontradas nos documentos, explicando em detalhes suas funcionalidades, endereço, valor inicial, etc.

A seguir, veremos um dos registradores de controle das GPIOs de um STM32.



Program Counter - PC

O Program Counter soluciona o problema de dizer onde o programa está.

Nada mais é do que um ponteiro que aponta para a próxima instrução que deverá ser executada.

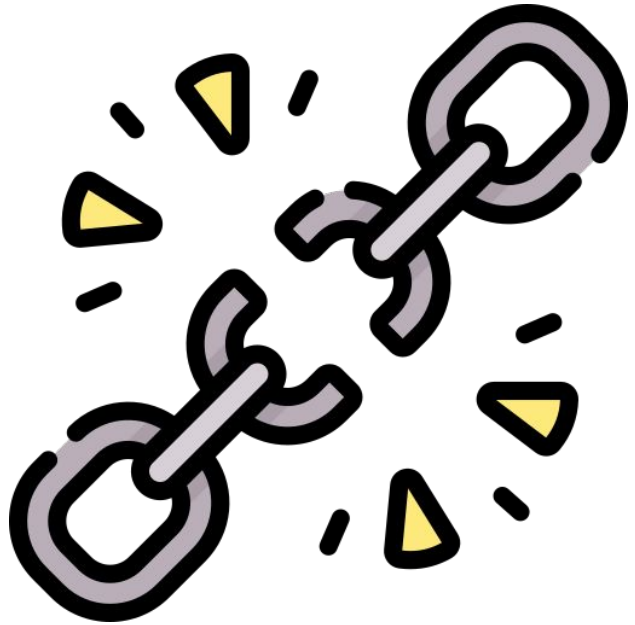
No **POR** (*Power On Reset*) o PC inicia sempre no endereço 0x0000, e é incrementado automaticamente ao executar a próxima instrução.

- MOVLW
- ADDLW
- SBL

Program Counter - PC

O contador é incrementado a cada instrução até o momento em que:

- Seja alterado por uma instrução (causando um *jump* na aplicação);
- Em uma chamada de função;
- Na ocorrência de um evento de interrupção.



Stack

A stack é uma pilha que é utilizada em chamadas de função e desvios causados por interrupções.

Armazena o endereço da próxima instrução que seria executada na ocorrência da chamada de função ou interrupção.

Stack - Passos

1 - É feito o **PUSH** na *stack* do valor do *Program Counter* (PC) no momento do desvio;

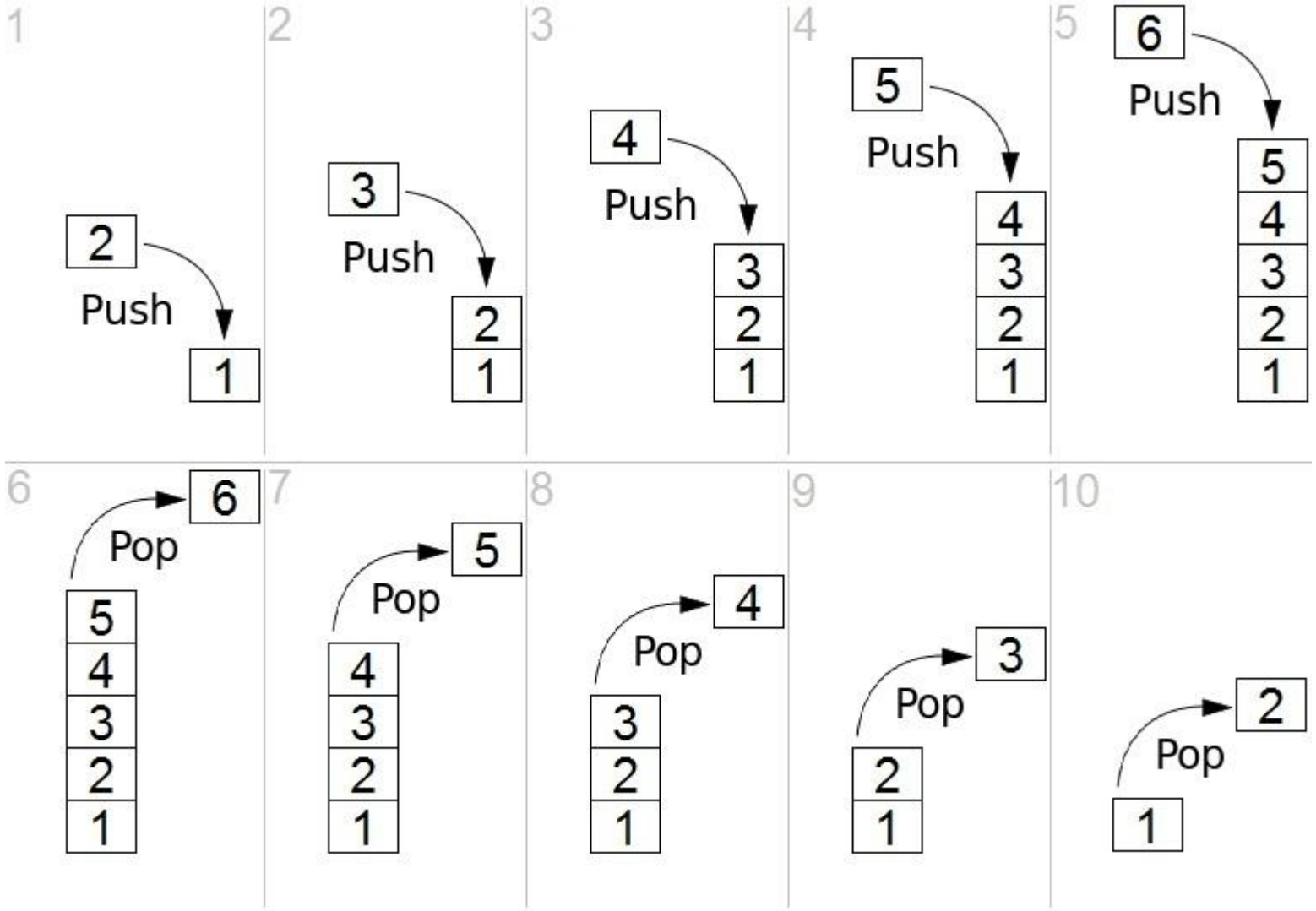
2 - O PC é modificado com o endereço de início da subrotina;

3 - O programa então pula (*jump*) e executa a subrotina;

4 - Quando ocorre a chamada de *return*, é feito um **POP** na *stack*, e o valor atribuído ao PC;

5 - O programa retorno a próximo instrução que seria executada antes da chamada da função.

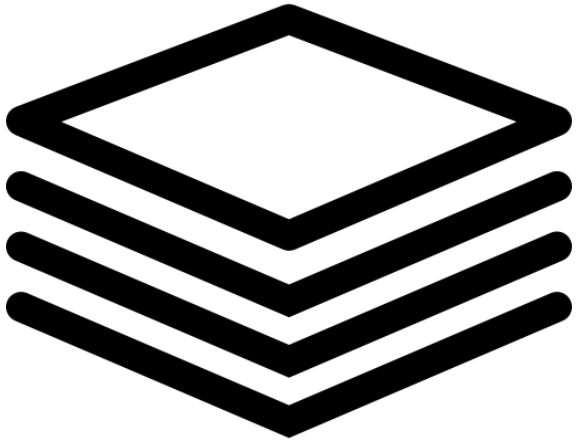
Stack - Exemplo



Stack - Implementações

A *stack* pode ser implementada por:

- Software : (STM8, STM32) a *stack* é armazenada na memória RAM e pode ser definido pelo desenvolvedor no programa. Porém, parte da memória RAM é perdida, apesar da flexibilidade.
- Hardware : (PIC16, PIC18) a *stack* possui um registrador em hardware dedicado.



Set de Instruções

O *Instruction set* ou set de instruções é o conjunto de todas instruções que são compreendidas e executadas pelo microcontrolador.

Quando um programa é compilado e gravado no microcontrolador, o binário gerado nada mais é do que a sequência de instruções geradas para atingir a finalidade desejada.

Set de Instruções - Exemplo

Exemplo de instruções de um core PIC18.

```
MOVLW 10H          ; Set 0x10 to the WREG
MOVF  20H, 1, 0    ; Move 0x10 to the address 0x20
MOVLW 5H           ; Set 0x5 to the WREG
ADDWF 20H, 1, 0    ; Sum WREG with value on 0x20
                   ; (0x10+0x5) and stores in 0x20
```

Set de Instruções - Categorias

Os sets de instruções variam de cada *core* de microcontroladores. E estes possuem duas categorias.

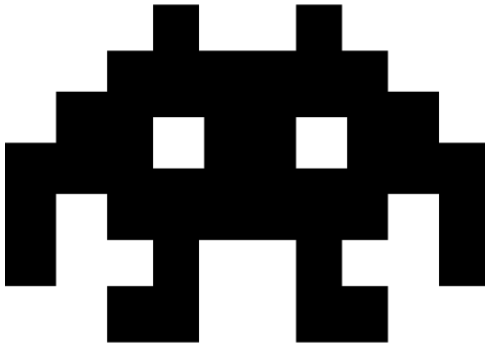
- **CISC** (*Complex Instruction Set Computer*) : Grande gama de instruções, sendo algumas realizadas por *microprogramas* gravados na CPU. Geralmente associado a arquitetura Von Neumann.
- **RISC** (*Reduced Instruction Set Computer*) : Pequeno conjunto de instruções mais simples, sendo mais barato de produzir e permite frequência de operação mais alto.

8 ou 32 Bits

Apesar de a linguagem C abstrair diversas peculiaridades, existem diferenças consideráveis entre a arquitetura dos microcontroladores.

Notável principalmente quando se trabalha em baixo nível, o *assembly*. Que é uma linguagem de programação que envolve apenas as instruções do microcontrolador.

O que define os bits é o tamanho do barramento de dados.



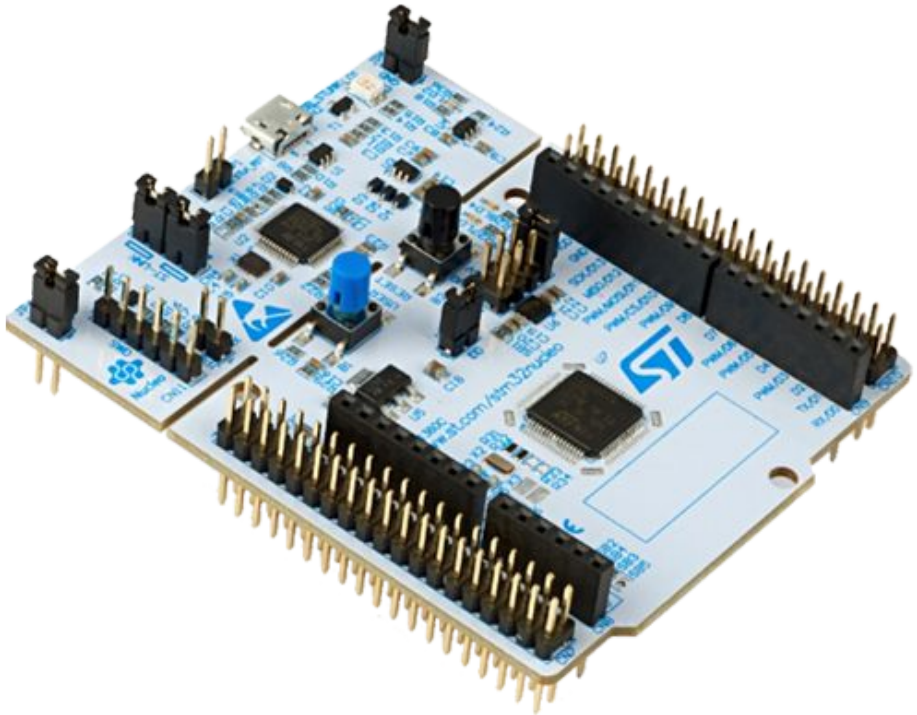
Mão na Massa

PADO Labs

NUCLEO-G0B1RE

Utilizaremos o kit **NUCLEO-G0B1RE** da ST.

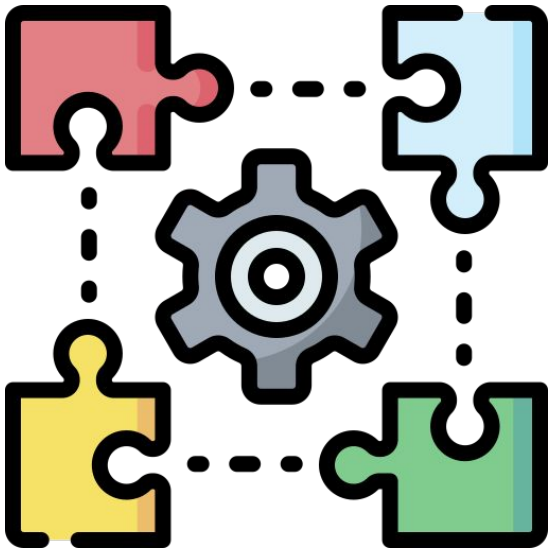
Utiliza um **STM32G0B1RE** como microcontrolador, 1 botão e 1 LED na placa, debugger integrado e conectores para conexões externas.



NUCLEO-G0B1RE

Principais recursos do microcontrolador:

- Unidade de cálculo CRC
- 12 timers + RTC alimentado por bateria
- Controlador DMA
- ADC e DAC de 12 bits
- três interfaces I2C
- Seis USART + 2 LPUART
- Três SPI
- Interface HDMI
- USB OTG
- Dois I2S
- Entre outros



Instalar o STM32CubeIDE

Neste momento, vamos instalar a IDE que iremos utilizar para desenvolver nosso kit.

O download do software está no Classroom.

E também materiais relacionados

ao microcontrolador.

Código exemplo

Está disponível também um código exemplo para testarmos a placa e a instalação assim que concluído.

Para acessar, basta acessar o github da Pado Labs

https://github.com/padolabs/firmware-A1-Test

Dúvidas ??

Referências

GEEKS, Geeks for. **Difference between Von Neumann and Harvard Architecture**. 2021. https://www.geeksforgeeks.org/difference-between-von-neumann-and-harvard-architecture . Acesso em 10 de Dezembero de 2021.

GIMENEZ, Salvador Pinillos. **Microcontroladores 8051: teoria do Hardware e do Software: aplicações em controle digital: laboratório e simulação**. 1. ed. [S.l.], 2002. ISBN 9788587918284.

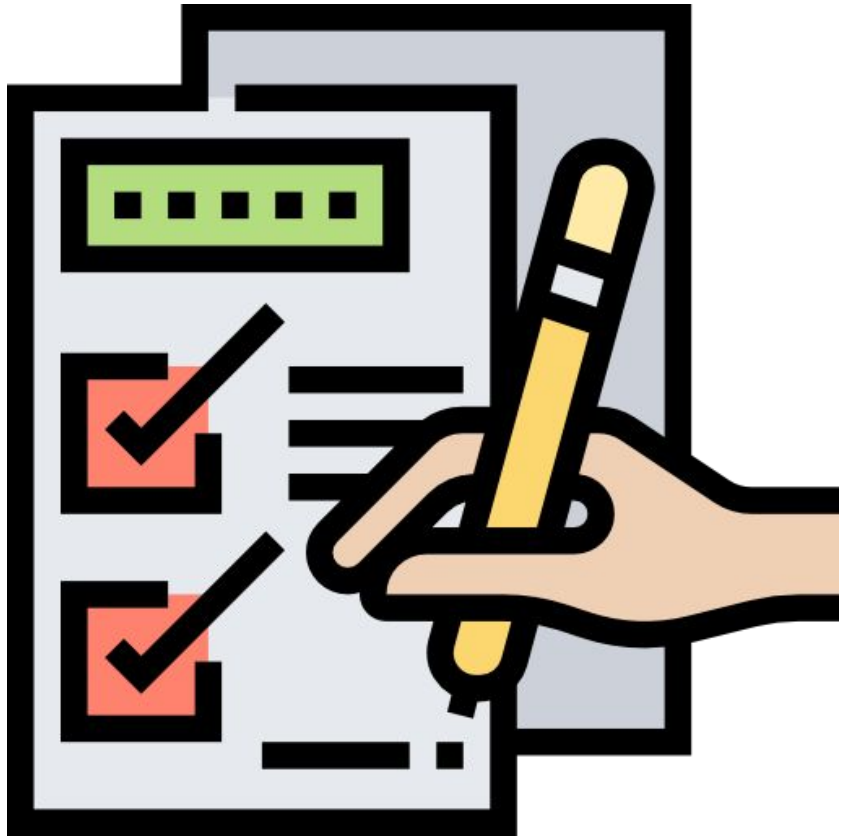
STMICROELETRONICS. **RM0444 - Reference Manual**. 5. ed. [S.l.], 2020. STM32G0x1 advanced Arm ® -based 32-bit MCUs.

____ . **UM2324 - User Manual**. 4. ed. [S.l.], 2021. STM32 Nucleo-64 boards (MB1360).

The Cortex-M0 Unstruction Set. 2021. https://developer.arm.com/documentation/dui0497/a/the-cortex-m0-instruction-set. Acesso em 10 de Dezembro de 2021.

WILDER, Jon. **A beginner's guide to microcontrollers**. 2015. https://www.microcontrollertips.com/a-beginners-guide-to-microcontrollers-faq/. Acesso em 10 de Dezembro de 2021.

Lista de Exercícios #1



Introdução aos microcontroladores

PADO

Labs