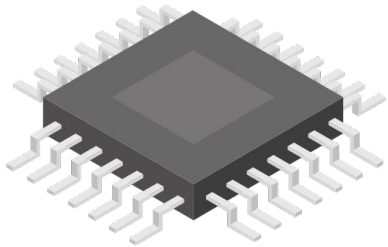
PADO

Labs

Microcontroladores



Inteiros e IOs

Prof.º: Pablo Jean Rozário

pablo.jean@padotec.com.br

/in/pablojeanrozario

https://github.com/Pablo-Jean

Índice da Aula #2

- Variáveis primitivas C
- stdint.h
- IOs de uso geral
- Registradores da GPIO
- Configurar GPIO no STM32CubeIDE
- Lista de Exercícios #2

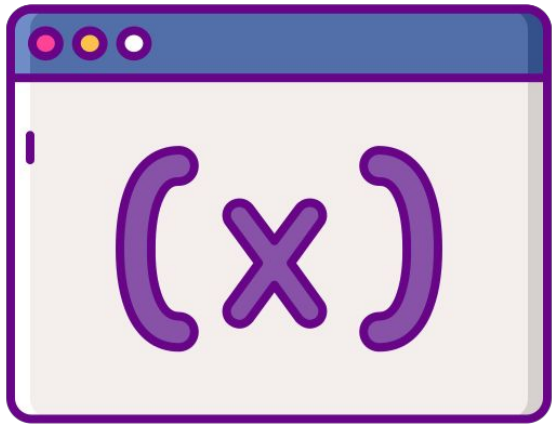
VARIÁVEIS PRIMITIVAS

C

Variáveis primitivas C

- **char** : Armazena um único caractere e requer apenas um byte de memória em quase todos compiladores
- **int** : Utilizado para armazenar números inteiros
- **float** : Utilizado para armazenar números decimais (números com valor em ponto flutuante) com precisão singular
- **double** : Igual *float*, mas possui precisão dupla.

No entanto, quanto bytes ocupam estas variáveis?



Variáveis em C - Bytes

Tipo	Bytes
char	1
int	4
float	4
double	8

Mas estes são valores padrões, dependendo da arquitetura e implementação, podem ser diferentes.

Variáveis em C - Exemplo

Tamanhos utilizados pelo MikroC PIC18.

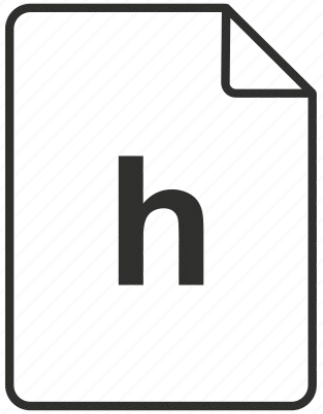
Type	Size in bytes
(unsigned) char	1
signed char	1
(signed) int	2
unsigned (int)	2
(signed) long (int)	4
unsigned long (int)	4

stdint.h

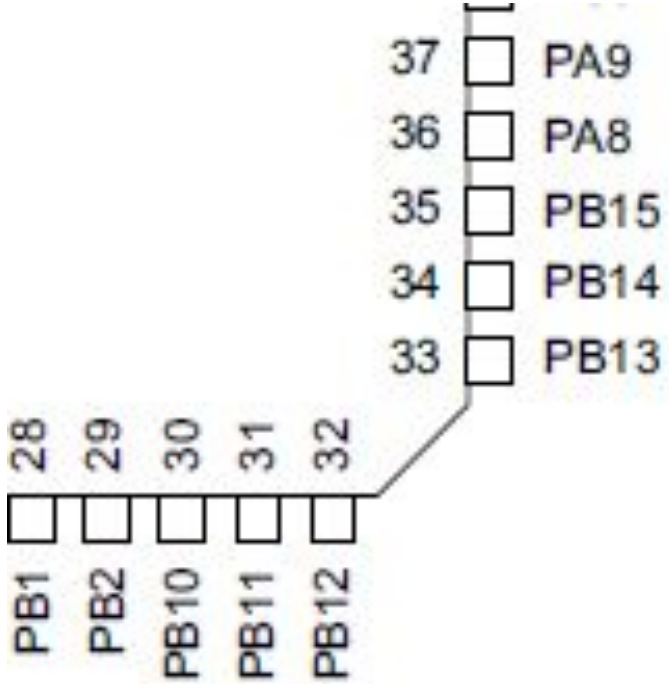
Uma biblioteca que pode ser utilizada para definir valores específicos de largura para números inteiros é a *stdint.h*.

Com ela é possível definir tipos como:
(u)intX_t :Variável inteira que ocupa o número X de bits
(u)int_fastX_t : Variável inteira que irá alocar o número de bits que implicará em melhor performance do chipset.
(u)int_leastX_t : Variável inteira que ocupará o menor número de bits possível que contemple os X bits.

Sendo X podendo ser **8**, **16**, **32** e **64**.



GENERAL PURPOSE IO



GPIO

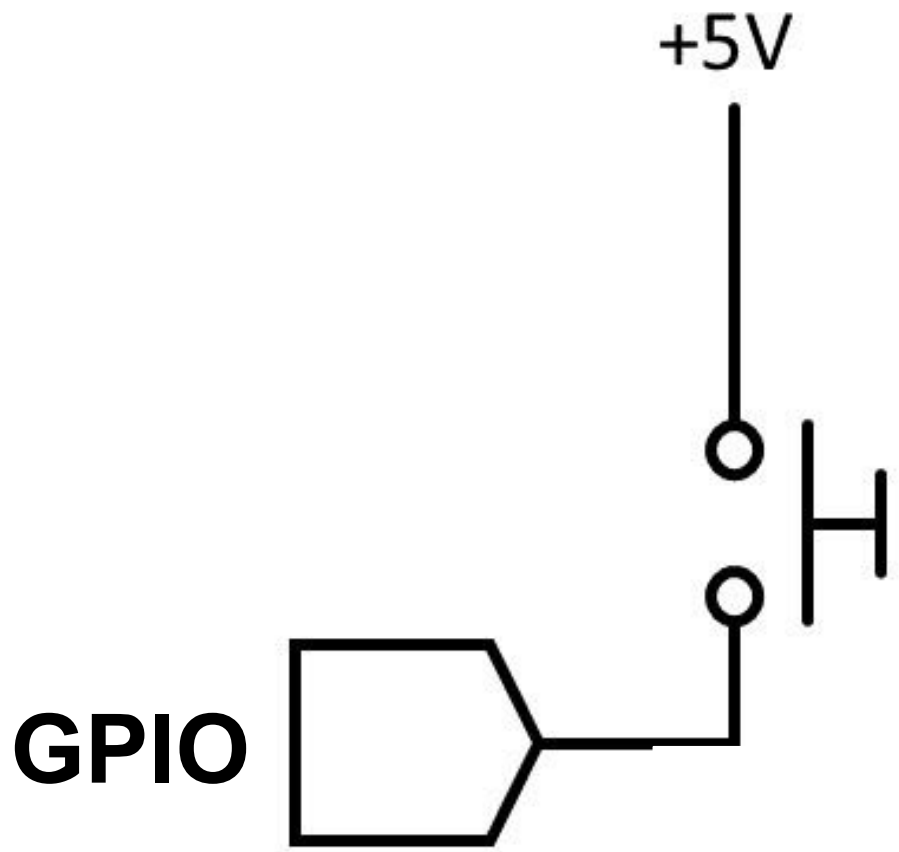
General Purpose Input-Outputs, ou GPIO, são terminais digitais utilizados pelo microcontrolador para interagir com o mundo externo.
Podem assumir função de **entrada** ou **saída**.

As GPIOs também são multiplexadas com outros periféricos do microcontrolador.

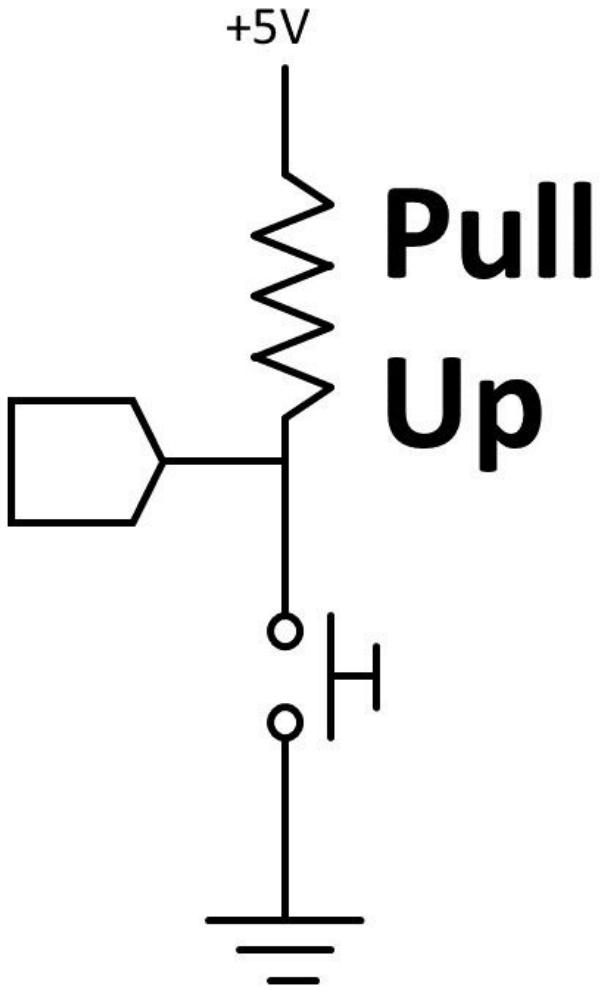
Hi-Z, Floating Pin, Pull-Up/Down

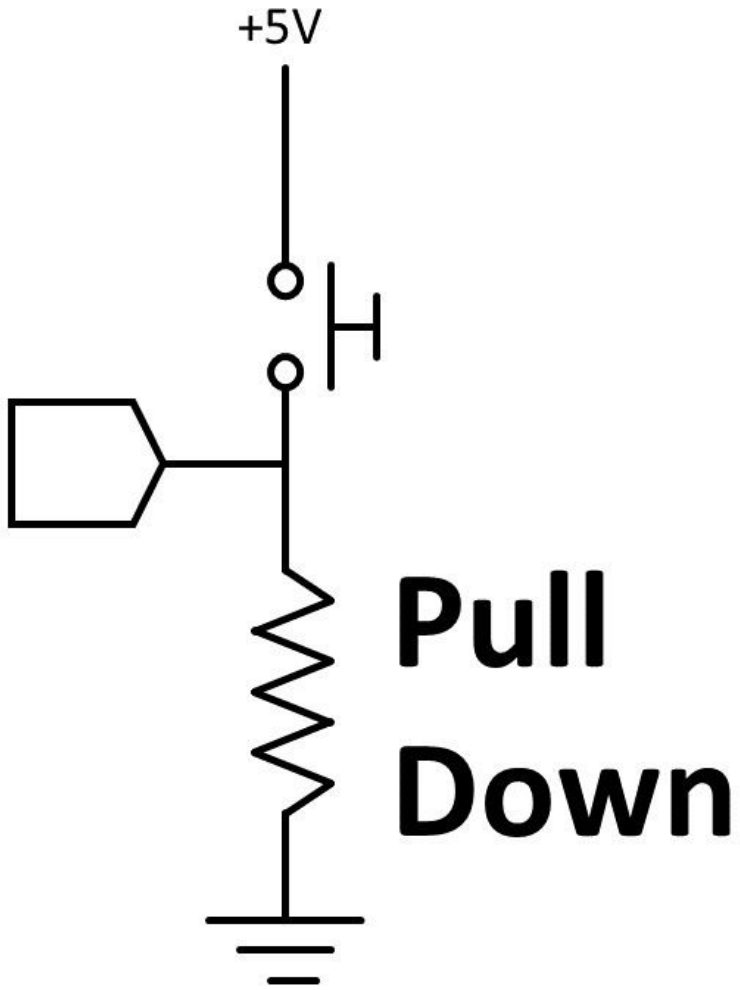
Hi-Z : Terminal caracterizado por ter uma alta impedância, de tal forma que mitiga uma influência no circuito externo.
Floating Pin : Um terminal deixado desconectado, no qual seu nível de tensão e lógico é indefinido e imprevisível.
Pull-Up/Down : Resistores utilizados para definir um nível lógico em um terminal flutuante. Resistor de *Pull-Up* é conectado ao barramento de alimentação e define o nível lógico 1. Resistor de *Pull-Down* é conectado ao referencial (*ground*) e defino o valor 0.

Sem Pull-Up/Down

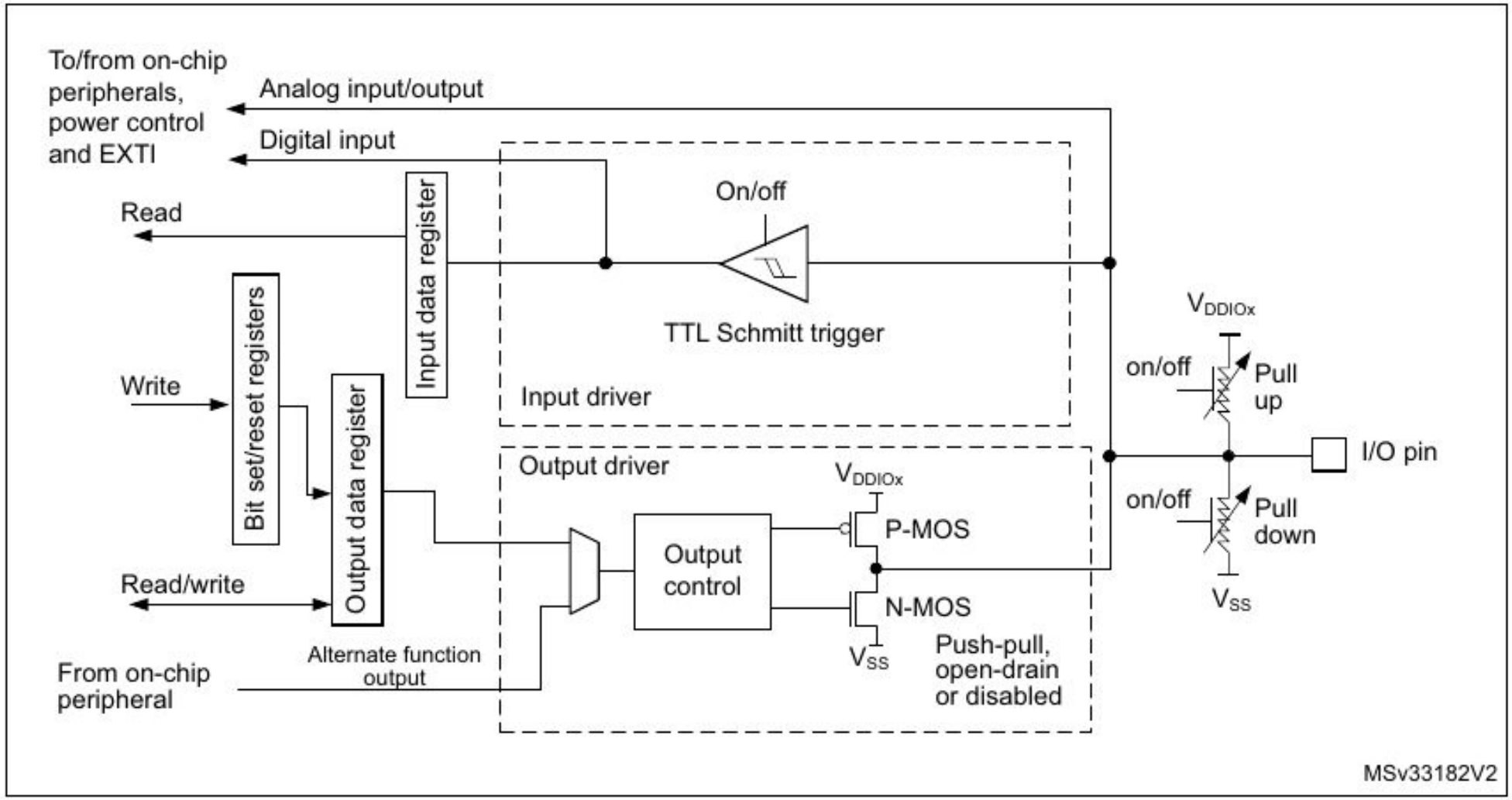


Pull-Up e Pull-Down

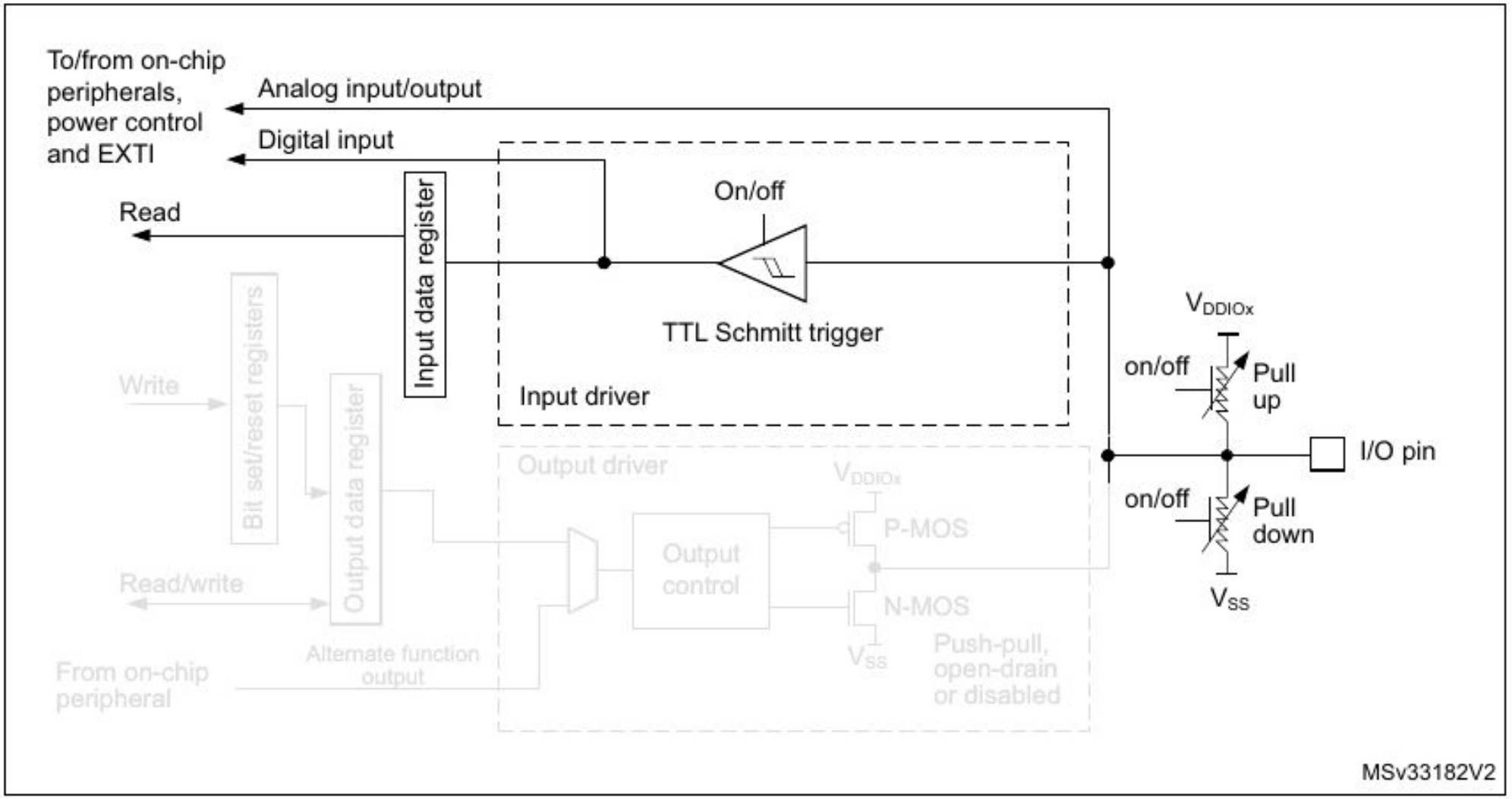




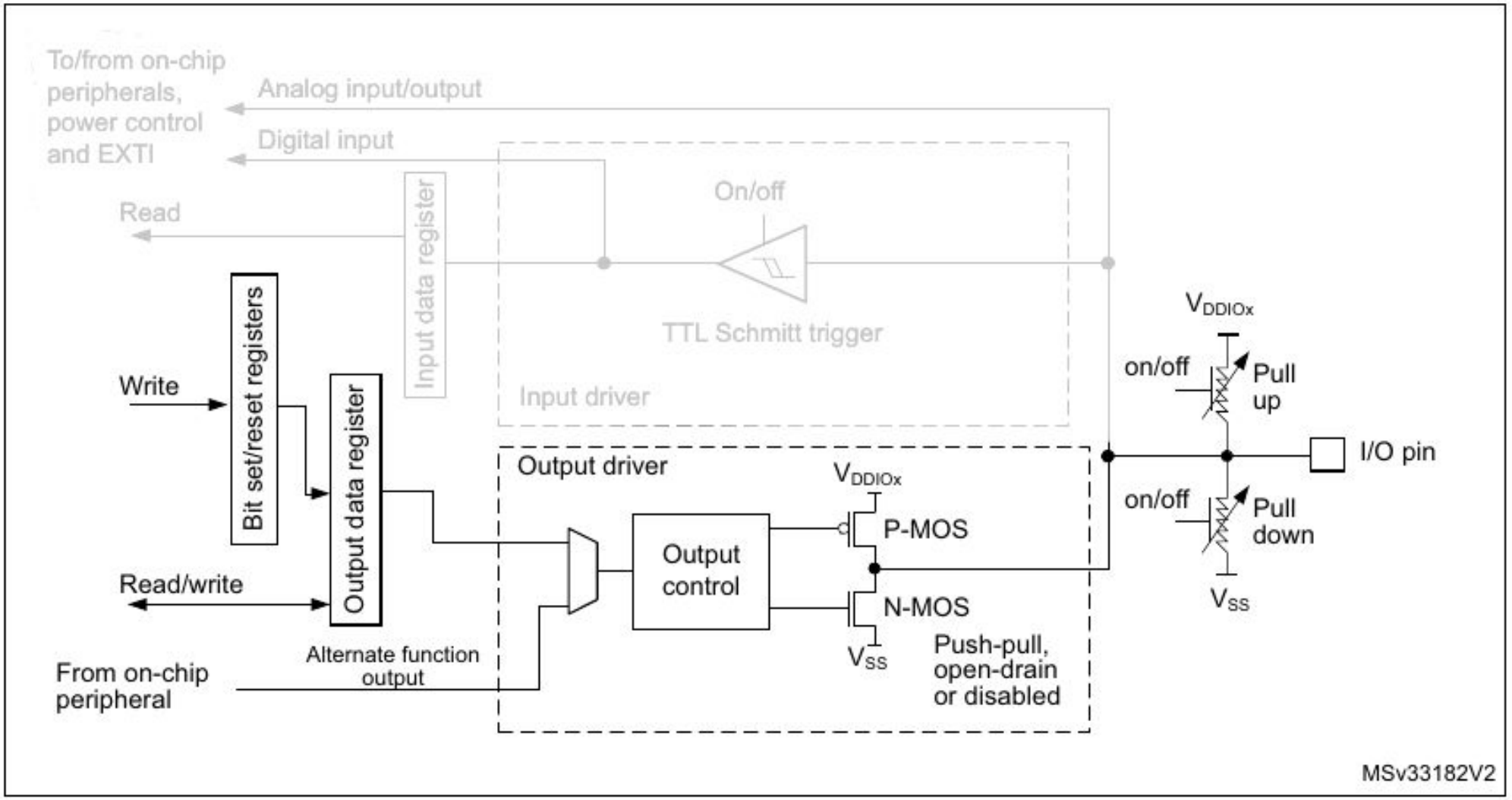
GPIO do STM32G0B1RE



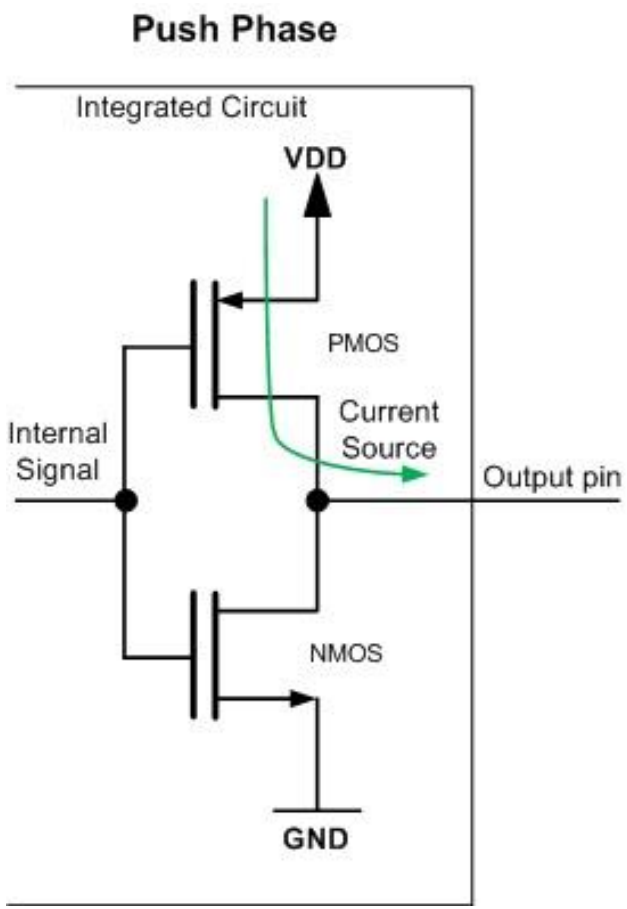
Input do STM32G0B1RE

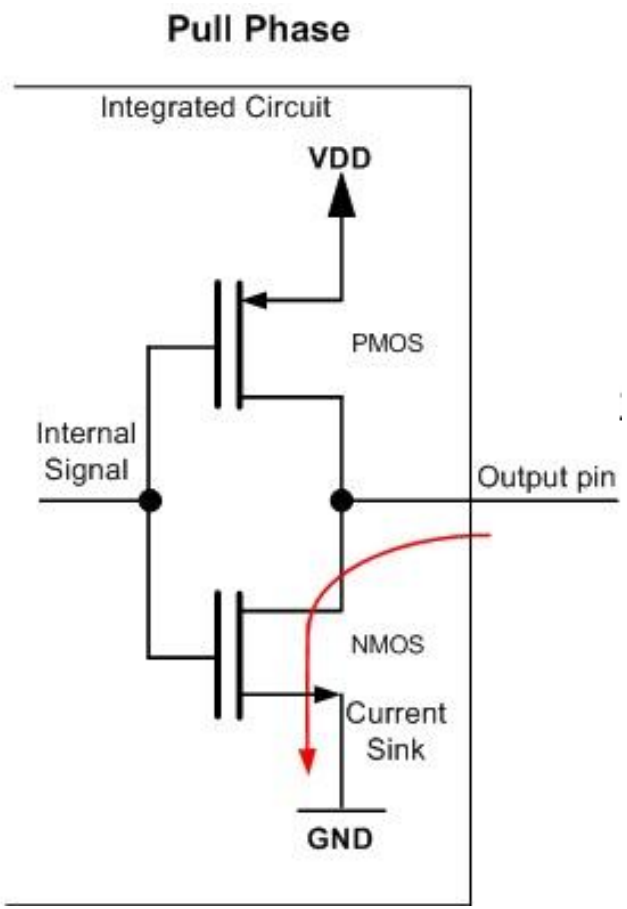


Output do STM32G0B1RE



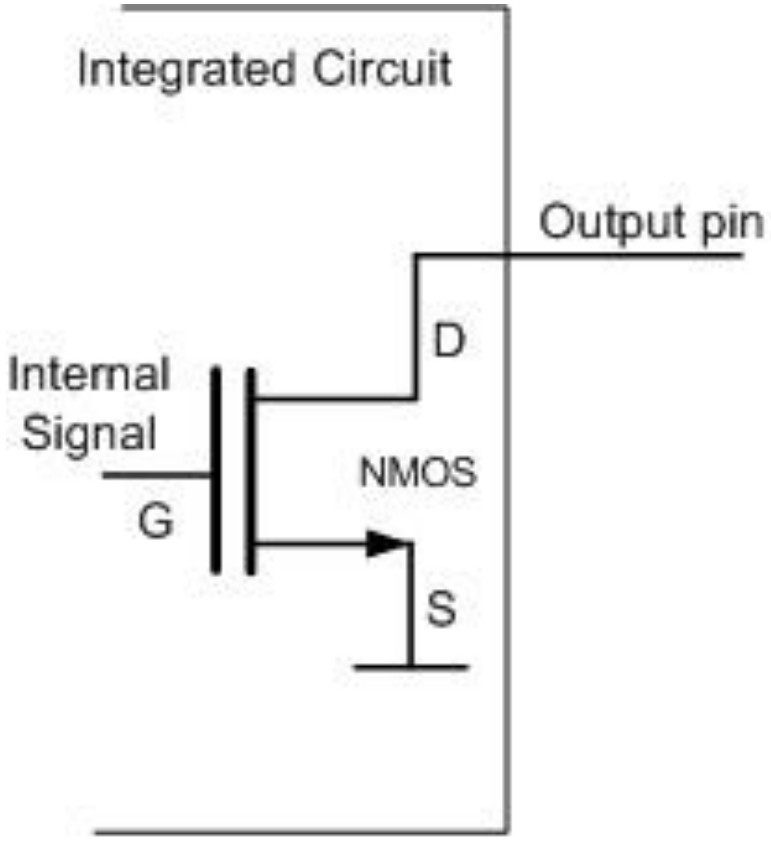
Output - Modo Push-Pull





Sinal	Saída
0	GND
1	VDD

Output - Modo Open Drain



Sinal	Saída
0	GND
1	Hi-Z

Output - Aplicações

Modo Push-Pull : Muito utilizado em interfaces unidirecionais como SPI e UART. Possui melhor tempo de resposta que o *open-drain*.

Modo Open-Drain : Utilizado em comunicação bi-direcionais (I2C, OneWire), mas possui maior corrente de consumo quanto acionado (quando inserido resistor de pull-up).

Registradores

Vamos falar um pouco sobre os *Special Function Registers* (SFRs) do STM32 para as GPIOs.

Para isto vamos abrir novamente o documento.

RM0444 : STM32G0x1 advanced Arm®-based 32-bit MCUs

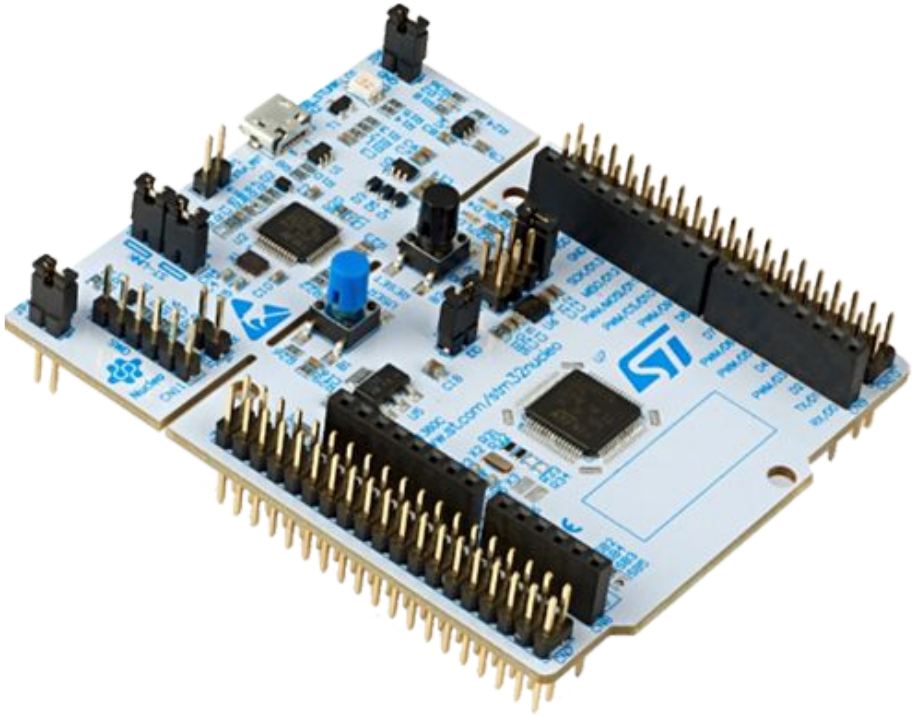
Mão na Massa

PADO Labs

GPIO NUCLEO-G0B1RE

Utilizaremos o kit **NUCLEO-G0B1RE**!

Vamos criar um novo projeto e configurar a GPIO.

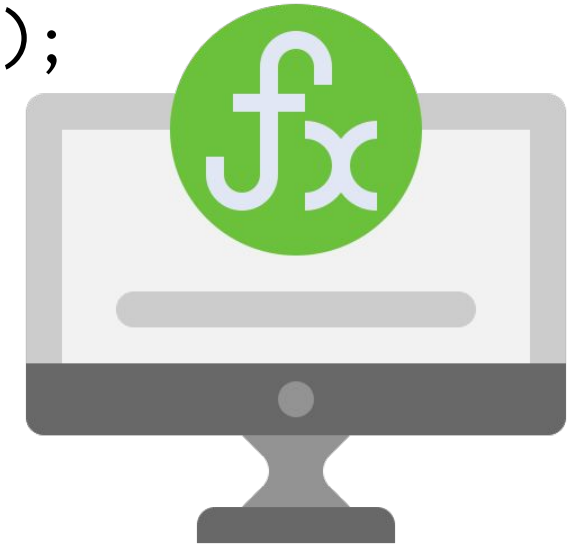


Funções - Output

Utilizamos:

```
// Escreve um valor GPIO_PIN_SET ou GPIO_PIN_RESET em uma
// porta configurada como output
HAL_GPIO_WritePin(GPIO_Port, GPIO_Pin, Value);

// Inverte o valor de uma output
HAL_GPIO_TogglePin(GPIO_Port, GPIO_Pin);
```

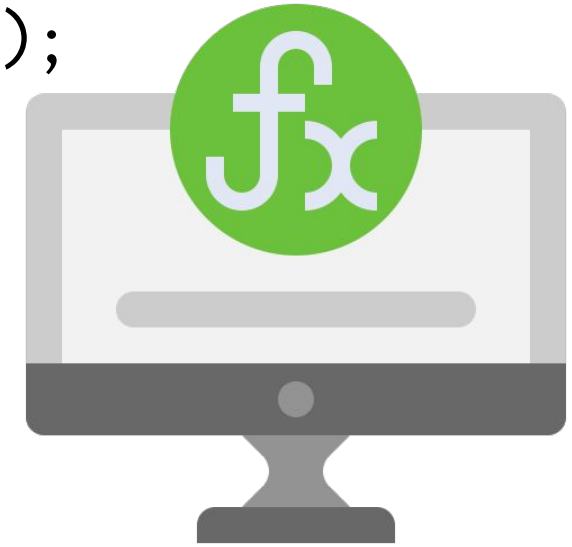


Funções - Input

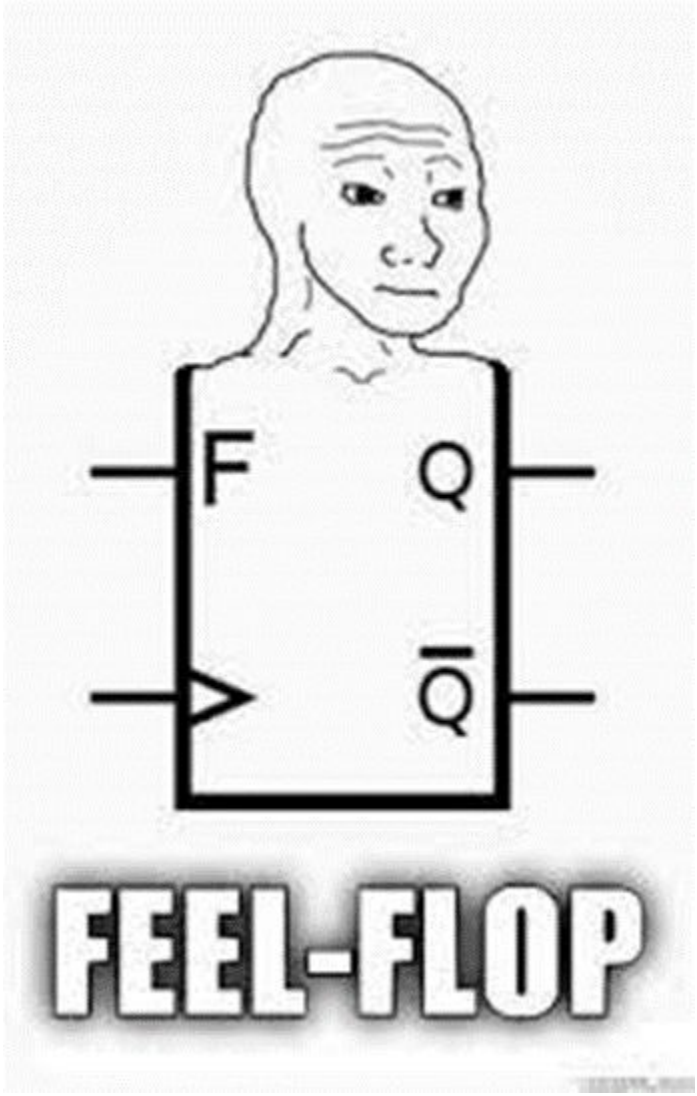
Utilizamos:

```
// Le o valor da porta, retornando o valor GPIO_PIN_SET ou
// GPIO_PIN_RESET
Value = HAL_GPIO_ReadPin(GPIO_Port, GPIO_Pin);
```

Existem outras funções, que podem ser consultadas no **UM2319** no capítulo *24 - HAL GPIO Generic Driver*.



Dúvidas ??



Referências

GEEKS, Geeks For. Data Types in C. 2021. https://www.geeksforgeeks.org/data-types-in-c/. Acesso 20 de Dezembro de 2021.

MIKROELEKTRONIKA. Arithmetic Types. [S.l.], 2012. Acesso em 20 de Dezembro de 2021.

NEWBEDEV. Difference between uint8_t, uint_fast8_t and uint_least8_t. 2021. https://newbedev.com/difference-between-uint8-t-uint-fast8-t-and-uint-least8-t . Acesso 24 de Dezembro de 2021.

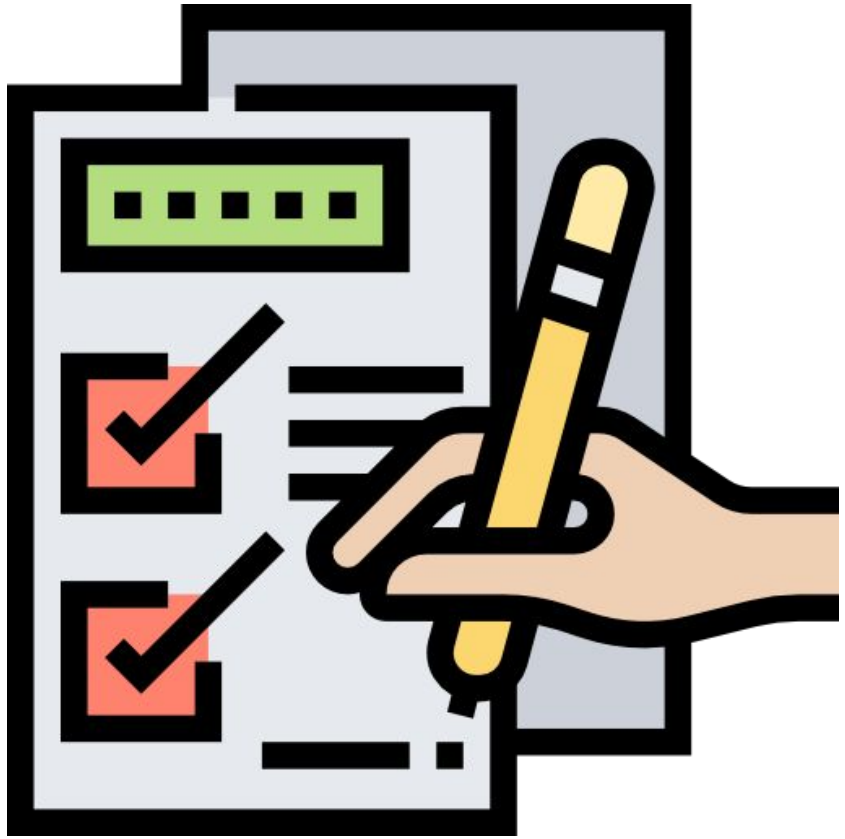
OPEN4TECH. Open Drain Output vs. Push-Pull Output. 2019. https://open4tech.com/open-drain-output-vs-push-pull-output/. Acesso em 23 de Dezembro de 2021.

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

STMICROELETRONICS. UM2319 Description of STM32G0 HAL and low-layer drivers.

Rev 2. [S.l.], 2020.

Lista de Exercícios #2



Registradores e IOs

PADO

Labs