

# Entrega 3 - (OLD) 2018-1

## Warning

semestre passado cada aluno teve que desenvolver um periférico para um hardware externo a FPGA, por diversas razões eles gastaram muito tempo nisso, nesse semestre eu quero facilitar um pouco para que possamos avançar em outras coisas, vou deixar a entrega aqui de referência para vocês verem o que foi feito!!!

Criar um IP dedicado para interfacear com um hardware externo a FPGA, cada aluno (ou dupla) deverá escolher um periférico e criar tanto o componente em HDL que conecta-se ao barramento Avalon quanto uma camada de abstração de hardware (especificada a seguir) para comandar esse periférico.

Os periféricos que podem ser escolhidos são:

- ☐ Teclado matricial de membrana 12 teclas
  - Sabrina
- ☒ Motor de passo:
  - Implementado por: **Gustavo / Astur**
- ☒ Motor DC com ponte H
  - Implementado por: **Cunial**
- ☒ Sensor ultrassônico HC-SR04
  - Implementado por: **Raphael**
- ☒ Encoder rotacional KY 040
  - Implementado por: **Marcelo**
- ☐ Sensor de temperatura e umidade AM2302:

## Driver software

- <https://www.intel.com/content/www/us/en/programmable/documentation/iga1430779693885.html>
- [https://www.intel.com/content/dam/altera-www/global/ja\\_JP/pdfs/literature/hb/nios2/n2sw\\_nii52003.pdf](https://www.intel.com/content/dam/altera-www/global/ja_JP/pdfs/literature/hb/nios2/n2sw_nii52003.pdf)

Para cada sensor deve ser desenvolvido um driver capaz de interagir com o periférico. Iremos padronizar algumas funções a fim de definirmos um padrão de interface:

```
int NAME_init( ..... );           // Inicializa o periférico
int NAME_config( ..... );         // Configura o periférico
int NAME_halt( ..... );           // Desativa o periférico
int NAME_en_irq( ..... );         // Habilita interrupção
int NAME_disable_irq( ..... );    // Desabilita interrupção
int NAME_read_xxxx( ..... );      // read data xxxx from device
int NAME_write_xxxx( ..... );     // write data to xxxx device
```

Esse driver deve estar distribuído em dois arquivos: `NAME.c` e `NAME.h`.

## Rubricas gerais do projeto:

- **I**
  - Não entregou
- **D**
  - Entregou o tutorial
- **C (individual)**
  - Fornecido exemplo que utiliza o periférico
  - Desenvolvido biblioteca em C para abstração do periférico (de acordo com documentação)
  - Periférico possui registrador de configuração que o possibilita ligar e desligar
  - • rubrica específica

- **B (individual) / C (dupla)**
  - rubrica específica
- **A (individual) / B (dupla)**
  - um item da rubrica específica
- **A+ (individual/dupla)**
  - dois itens do A

## Sensores

Detalhamento dos sensores e das rubricas.

### Teclado matricial

O periférico deve fazer a varredura automática das teclas e gravar em um registrador o último valor referente ao botão pressionado.

Rubrica:

- C
  - Periférico faz varredura das teclas e salva botão em registrador
- B
  - Sempre que uma tecla nova for detectada, gera uma interrupção
- A
  - Suporta diferentes configurações (frequência de varredura, diferentes teclados: 12, 16, ...)
  - Implementa debouncing
  - Armazena um vetor de teclas pressionadas

## Motor de passo

O periférico deve receber uma quantidade de pulsos e direção e realizar o controle das fases do motor de passo a fim de realizar tal operação.

Rubrica:

- C
  - Aciona motor com base em passos fornecidos
- B
  - Sempre que uma ação de movimentação for finalizada, gera uma interrupção.
- A
  - Velocidade de rotação é configurável
  - Periférico pode ser acionado tanto por passos quanto por graus

## Motor DC (Ponte H)

O periférico deve controlar a velocidade e direção de um motor DC modulando um sinal PWM. Esse periférico deve ser capaz de receber um valor de duty cycle, esse motor deve ficar acionado por um tempo determinado.

Rubrica:

- C
  - Aciona motor com base em duty cycle e direção fornecido.
  - Possui um timer interno que especifica por quanto tempo o motor ficará acionado.
- B
  - Gera uma interrupção quando o timer estourar.
- A (um item)
  - Implementa um controle em **malha aberta**.

# HC

Faz a leitura da distância de um objeto gerando o sinal de trigger no módulo e lendo o tempo de resposta para o echo. Deve possuir dois modos: Leitura periódica / Single shot.

Rubrica:

- C
  - Armazena em um registrador o valor do tempo da chegada do echo dado o inicio do trigger
  - Possui dois modos de operação: Periódico / Single shot
  - Período deve ser configurável por reg.
- B
  - Gera interrupção a cada nova leitura
- A
  - Armazena a distância em metros, não o tempo percorrido.
  - Armazena um buffer de leituras.
  - Filtra os dados de leitura para remover ruído

## Encoder de quadratura rotacional

Contabiliza a leitura dos pulsos de um encoder de quadratura para um valor, indicando o sentido de rotação. Deve possuir uma maneira de zerar a quantia de pulsos já contabilizada.

- C
  - Faz a leitura da quantidade de pulsos e a direção
- B
  - Gera interrupção a cada novo pulso detectado
- A
  - Armazena a informação em graus

- Faz a leitura por quadratura (sobida e descida) para aumentar a resolução

## Sensor de temperatura e umidade AM2302

Periférico responsável por estabelecer comunicação com o sensor AM2302 e extrair dados de temperatura e umidade. Deve possuir dois modos de operação: Periódico e Single Shot.

- C
  - Faz a leitura da temperatura e umidade
  - Período deve ser configurável por reg.
- B
  - Dois modos de operação: Periódico / Single Shot
  - Gera interrupção a cada novo valor
- A
  - Valida checksum e gera erro caso detectado algo errado