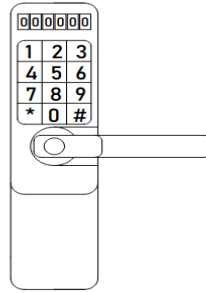


# Projeto – Fechadura Eletrônica



Nosso sistema se trata de uma fechadura eletrônica, ela possui a capacidade de armazenar até 4 senhas diferentes para realizar o destravamento da fechadura. É dotada da capacidade de monitorar e alertar o usuário caso a porta tenha sido deixada aberta emitindo um bip sonoro. O sistema também realiza seu próprio travamento após um curto período em que se encontre fechada.

## Os componentes do sistema:

1. Um teclado matricial 4×4 - Usado para inserir os pins que liberam o acesso
2. Seis displays de 7 segmentos
3. Uma chave de contato - responsável por indicar para o sistema se a porta se encontra aberta ou fechada.
4. Um botão para BLOQUEIO - função não perturbe, esta opção serve para desabilitar a abertura da fechadura pela unidade externa.
5. Um botão para fechar ou abrir a porta pelo lado de dentro.
6. Um botão CONFIG - utilizado para configurar o sistema.
7. Um LED - Para indicar se a fechadura está trancada ou não.

## Sobre o funcionamento do sistema:

### Modo operacional:

- Durante o modo operacional nosso sistema segue um fluxo simples de operações, ele sempre fica a espera da inserção de uma senha válida vindo pelo teclado matricial.
- A senha que foi inserido só será verificado após o pressionamento da tecla “ \* ”.
- Uma vez que se faz o recebimento de uma senha padrão válida, a fechadura deve ser destrancada, permitindo com que o usuário possa abrir sua porta.

- OBS: para evitar que outras pessoas descubram a sua senha, seja por marcas dos dedos deixadas no teclado ou por estarem próximas a você quando digitar sua senha, a fechadura permite o acréscimo de dígitos aleatórios, antes e/ou depois da senha. São permitidos 21 dígitos no total (somando a senha + dígitos aleatórios + \*). Lembrando que a senha pode ser de 4 a 12 dígitos. Exemplo: considerando que a senha cadastrada seja 1234, a abertura da fechadura poderá ser conforme a seguir:
  - Dígitos aleatórios + senha + \* = (4365944679437972 + 1234 + \*);
  - Senha + dígitos aleatórios + \* = (1234 + 4365944679437972 + \*);
  - Dígitos aleatórios + senha + dígitos aleatórios + \* = (44679437972 + 1234 + 25478 \*);
- OBS: Uma vez que o usuário digite uma tecla, ele possui 5 segundos para digitar uma nova tecla ou confirmar o conjunto de teclas que já digitou com “\*”, caso contrário a fechadura emitirá um bip, desconsiderando os dígitos atualmente inseridos.
- OBS: A tecla “#” atuará como um botão de clear, ou seja ele descartará a senha atual.
- Uma vez com a tranca aberta, caso a porta se encontre aberta, será iniciado uma contagem de 5 segundos (tempo padrão, que pode ser configurável), e ao fim desta contagem a mesma emitirá um bipe contínuo alertando o usuário sobre o fato que sua porta se encontra aberta.
- Caso a porta esteja fechada, será iniciado uma contagem crescente até 5 segundos (tempo padrão, que pode ser configurável), ao final a tranca da porta será fechada. Caso a porta seja aberta durante a contagem, o contador será reiniciado.
- Caso a porta seja aberta antes de ultrapassar o período de 5 segundos para realizar seu trancamento automático, ela começa a contagem para o acionamento do bip indicando que a porta está aberta. Quando a porta voltar a ser fechada, a contagem em relação ao trancamento automático será resetada e será iniciada novamente a contagem para o trancamento automático.
- Se a fechadura receber uma senha inválida, ela ficará inoperante por um tempo fixo de 1 segundo, durante esse segundo, deverá ser mostrado no display de 7 segmentos HEX0 o traço, por exemplo “-”. A cada tentativa incorreta, o display seguinte deverá mostrar o traço, por exemplo, se o usuário errar 3 vezes, deverá ser mostrado um traço nos displays ( HEX2, HEX1 e HEX0 ). Após 5 tentativas incorretas a fechadura ficará bloqueada por 30 segundos e o traço deverá ser mostrado em todos os displays. Obviamente, durante o bloqueio, a fechadura deve desconsiderar qualquer entrada realizada pelo teclado matricial.
- A fechadura pode ser destravada ou travada pelo lado de dentro mediante acionamento do botão interno. Observação, só pode travar se a porta estiver fechada.
- Função não perturbe, esta opção serve para desabilitar a abertura da fechadura pela unidade externa, com a porta fechada, mantenha o botão **BLOQUEIO** pressionado por 3s. Para desabilitar essa opção, basta abrir a fechadura pelo botão de fechamento/abertura.
- E por fim, mediante o pressionamento do botão **CONFIG**, a fechadura deverá sair de seu modo de operação padrão e entrar no modo de configuração. OBS: A porta deverá estar destravada e aberta.
- O sistema deve funcionar com um clock de 1KHz.

## Modo de configuração:

O modo de configuração é responsável por alterar alguns parâmetros da fechadura eletrônica. Uma vez no modo de configuração, o usuário passará por todas as opções sequencialmente, mesmo que não deseje realizar alguma operação específica.

As opções são ordenadas da seguinte forma:

- 1 - Ativar BIP (padrão: ativado)
- 2 - Tempo BIP (padrão: 5s)
- 3 - Tempo Fechamento Automático (padrão: 5s)
- 4 - Senha MASTER (padrão: "1234")
- 5 - Senha Padrão 1 (padrão: "FFFFFFFFFFFF")
- 6 - Senha Padrão 2 (padrão: "FFFFFFFFFFFF")
- 7 - Senha Padrão 3 (padrão: "FFFFFFFFFFFF")
- 8 - Senha Padrão 4 (padrão: "FFFFFFFFFFFF")

### Para entrar no modo configuração:

Aperte o botão de **CONFIG**. Ao apertar o botão, o display HEX5 presente na unidade externa irá apresentar o valor 0, indica que o sistema está apto a entrar no modo configuração. Nesse momento as configurações estão bloqueadas. É preciso digitar a senha MASTER + (\*). Caso o produto seja recém-instalado, digite a senha padrão 1234 e pressione (\*). O display HEX5 irá mostrar o valor 1, indicando que o sistema está apto para receber os parâmetros de configuração. **Caso o usuário deseje sair do modo de configuração, deve ser pressionado a tecla de jogo da velha "#". OBS: As informações que foram atualizadas antes de pressionar "#" deverá ser atualizado no Operacional.**

### 1 - Ativar BIP:

- Ativar ou desativar o bip é a primeira opção do nosso modo de configuração, sua função é ativar ou desativar a função de bipar da fechadura quando ela se encontra destrancada e aberta.
- Nos displays HEX5, deve aparecer o valor 1. O display HEX0 deverá apresentar o valor atual em relação a configuração da função. O usuário ativa ou desativa a função via teclado matricial, a tecla "1" será usada como indicador de que o bip deve estar ativado. Por sua vez, a tecla "0" deverá desativar o bip. Se alguma das duas teclas 0 ou 1 for pressionada, o valor deverá ser atualizado no display HEX0.
- A confirmação da escolha deverá ser feita pela tecla "\*" do teclado matricial. Uma vez que a escolha foi confirmada, o sistema deverá encaminhar para a próxima configuração. Obviamente os valores alteram de acordo com a mudança do usuário, somente a última mudança deverá afetar o sistema.
- Se qualquer outra tecla for pressionada, o sistema deverá desconsiderar.

### 2 ou 3 - Definir o tempo para bipar ou para realizar trancamento:

- As duas configurações compartilham os mesmos limites e comportamentos.
- O tempo tem um valor mínimo de 5 segundos (padrão) e um valor máximo de 60 segundos. O tempo deve ser inserido em segundos.

- No display HEX5, deve aparecer o valor (2 ou 3) respectivamente. Os displays HEX1 e HEX0 deverão apresentar o valor atual em relação a configuração da função.
- Caso deseje 8 segundos, deverá ser digitado “0” e “8”. Caso seja pressionado mais de 2 teclas, por exemplo, 1, 2 e 3, o sistema irá considerar apenas 2 e 3, considerando o valor 23 segundos.
- Supondo que ao entrar na função, o valor que está presente seja 0 e 5. Se for pressionado a tecla 6, será deslocado os valores nos displays, mostrando os dígitos 5 e 6, se for pressionado a tecla “\*”, o sistema irá armazenar o valor 56 segundos.
- A confirmação da escolha deverá ser feita pela tecla “\*” do teclado matricial. Uma vez que a escolha foi confirmada, o sistema deverá encaminhar para a próxima configuração.
- OBS. Na hora da confirmação, caso o display mostre valores fora da faixa ( 5 à 60 ), o sistema deverá considerar os limites. Por exemplo, se estiver mostrando os valores 0 e 3, após a confirmação será armazenado o valor 0 e 5. Caso, esteja mostrando os valores 9 e 9, após a confirmação será armazenado o valor 6 e 0.

## Senhas ( 4 a 8 )

- O gerenciador de senhas é responsável por alterar as 5 senhas. As operações assim como no menu anterior são executadas de forma sequencial, ou seja, para alterar o quarta senha, deve-se passar pelo outros três.
- O display HEX5 deverá mostrar a respectiva função.
- Deverá ser armazenado um valor de 4 a 12 dígitos. Não serão aceitas senhas com menos de 4 dígitos, se for mais de 12 dígitos, será considerado as últimas 12 teclas digitadas. Se for digitado menos de 4 dígitos e pressionado “\*”, o sistema deverá desconsiderar e apagar os dígitos.
- A tecla “\*” por sua vez é responsável por confirmar as modificações feitas. Uma vez confirmado, você é encaminhado para a próxima opção do menu e assim por diante.

## Sobre o reset:

- O sistema só será resetado após o botão do reset se mantiver pressionado por pelo menos 5 segundos, o reset funciona com lógica positiva. Ao ser resetado obviamente o sistema irá para os parâmetros padrão.
- Se a porta estiver fechada e travada, após o reset, permanecerá na mesma situação. Se a porta estiver fechada e destravada, após o reset a trava será acionada. Se a porta estiver aberta, após o reset o sistema só entrará no modo operacional quando a porta for fechada e travada automaticamente.

## Tipos pré-definidos

Os tipos pré-definidos são estruturas que assim como as interfaces dos módulos serão comuns a todos e não devem ser alteradas. No geral são pequenos pacotes de dados utilizados para facilitar a comunicação entre os módulos do sistema.

## SenhaPacked\_t:

Uma struct cuja função é armazenar até 20 dígitos. Seus valores variam de 0 a 9, qualquer valor fora desta faixa deve ser considerado invalido/sem dígito para com o sistema.

```
typedef struct packed {  
    logic [19:0] [3:0] digits;  
} senhaPac_t;
```

Nome	Tipo	Significado
digits[n]	logic [3:0]	Valor no formato BCD que representa um dígito.

## bcdPacked\_t:

Tipo de dado usado para empacotar os valores dos displays no formato BCD. Para mais informações sobre os valores consulte o módulo \*\*\*\*\*

```
typedef struct packed {  
    logic [3:0] BCD0;  
    logic [3:0] BCD1;  
    logic [3:0] BCD2;  
    logic [3:0] BCD3;  
    logic [3:0] BCD4;  
    logic [3:0] BCD5;  
} bcdPac_t;
```

Nome	Tipo	Significado
BCD*	logic [3:0]	Valor no formato BCD que a informação a ser exibida no display.

## SetupPacked\_t:

Tipo de dado usado para empacotar as informações transmitidas entre o módulo de setup e o módulo operacional.

```
typedef struct packed {  
    logic bip_status;  
    logic [5:0] bip_time;  
    logic [5:0] tranca_aut_time;  
    senhaPac_t senha_master;
```

```

senhaPac_t senha_1;
senhaPac_t senha_2;
senhaPac_t senha_3;
senhaPac_t senha_4;
} setupPac_t;

```

Nome	Tipo	Significado
bip_status	logic	indica se sistema deve ou não emitir algum bipe quando a porta se encontrar aberta (0 não / 1 sim)
bip_time	logic[5:0]	espaço reservado para informar a dimensão da contagem realizada antes de começar a bipar com a porta aberta
tranca_aut_time	logic[5:0]	Tempo referente a espera para trancar a porta quando aberta e não fechada.
senha_master	senhaPac_t	Pacote de dígitos contendo a senha master
senha_*	senhaPac_t	Pacote de dígitos contendo uma senha padrão

## Os módulos do sistema

### FechaduraTop:

A FechaduraTop é a casca do nosso sistema, ela é quem recebe todas as entradas do sistema e devolve as saídas, sendo responsável por rotear os sinais para os respectivos submódulos.

### Definição da FechaduraTop:

```

module FechaduraTop (
    input  logic clk, rst, sensor_de_contato, botao_interno, botao_bloqueio, botao_config,
    input  logic [3:0] matricial_col,
    output logic [3:0] matricial_lin,
    output logic [6:0] dispHex0, dispHex1, dispHex2, dispHex3, dispHex4, dispHex5,
    output logic tranca, bip );

endmodule

```

tabela de entradas e saídas:

*****	tipo	significado
sensor_de_contato	logic	entrada do sistema referente ao sensor de contato da fechadura (1 porta aberta / 0 porta fechada)
botao_interno	logic	entrada do sistema referente ao botão interno que destrava a fechadura.

botao_bloqueio	logic	Função não perturbe, este botão serve para desabilitar a abertura da fechadura pela unidade externa (Teclado)
botao_config	logic	Possui a função de entrada para configurações do sistema
matricial_col	logic[3:0]	entrada de dados do teclado matricial referente às colunas
matricial_lin	logic[3:0]	saída de dados do teclado matricial referente às linhas
dispHex*	logic[6:0]	Saídas referentes ao display de 7 segmentos.
tranca	logic	Saída referente ao estado da tranca (0 aberta / 1 fechada).
bip	logic	Saída referente ao estado do bip (0 desativado / 1 ativado).

## Módulo Teclado:

Módulo responsável por lidar com o teclado matricial, configurando sua leitura, realizando debounce para evitar ruídos no sistema e decodificar o resultado de forma interpretável ao sistema. O módulo entrega para seus consumidores um conjunto de 20 dígitos cujos valores são apresentados na tabela abaixo (código BCD). Sempre que o teclado decodificar o pressionar de uma tecla ele deve atualizar o barramento com o atual conjunto de dígitos, caso seja a tecla “\*” o pino *digitos\_valid* deve ter seu valor como 1. Toda vez que esse processo for realizado, o barramento *digitos\_value* deve ser completado com o valor “0xF” em todo o barramento, indicando não haver dígitos ali.

Tecla Pressionada	Valor
0	0x0
1	0x1
2	0x2
3	0x3
4	0x4
5	0x5
6	0x6
7	0x7
8	0x8
9	0x9
*	0xA
#	0xB

Caso o usuário não digite as 20 teclas, ou seja não complete todo o array de dígitos, os dígitos faltantes devem ser completados com o valor “0xF”. Exemplo:

Foi digitado: 12345678 + \*  
ARRAY: [ FFFFFFFF12345678 ]

A cada nova tecla digitada o usuário possui até 5 segundos para inserir uma nova tecla, ou confirmar com o “\*”, caso contrário o barramento *digito\_value* deve ser preenchido com o valor “0xE” e pino de saída *digitos\_valid* deve ter seu nível lógico em 1. Após um pulso seu sinal deve ser derrubado para 0 e então o barramento *digito\_value* deve ter seu valor preenchido por “0xF”.

A tecla “#” possui um comportamento especial, uma vez que o teclado decodifique tal tecla, ele preencherá o barramento com o valor “0xB” e o pino de saída *digitos\_valid* deverá ter seu nível lógico em 1 por um pulso de clock. Após tal pulso o barramento *digitos\_value* deverá ser preenchido pelo valor “0xF”

O sinal de *enable* serve para informar ao módulo se ele deve ou não estar lendo e tratando o teclado, quando desabilitado o teclado deve parar seu funcionamento, ficando com seus valores padrão, já quando o sinal de *enable* estiver em nível alto, ele deve operar de forma normal.

Definição do teclado:

```
module decodificador_de_teclado (  
    input        logic        clk,  
    input        logic        rst,  
    input        logic        enable,  
    input        logic [3:0]   col_matriz,  
    output       logic [3:0]   lin_matriz,  
    output       senhaPac_t    digitos_value,  
    output       logic         digitos_valid  
);  
  
endmodule
```

Entradas e Saídas do módulo:

Nome	Tipo	Significado
enable	logic	Informa ao módulo se ele deve estar lendo e tratando as teclas ou não.
col_matrix	logic[3:0]	Identificar a coluna
lin_matrix	logic[3:0]	Identificar a linha
digitos_value	senhaPac_t	Barramento de entrada com os últimos 20 dígitos do teclado, só deve ser lido quando ocorrer uma transição ascendente de <i>digitos_valid</i> .
digitos_valid	logic	Sinal para informar que os dados do teclado podem ser lidos, esse sinal é uma transição ascendente.
clk	logic	Clock do sistema
rst	logic	Reset do sistema



## Módulo Display:

O módulo Display é responsável por armazenar e exportar os valores dos displays de sete segmentos. Uma vez que o módulo é habilitado recebendo um nível lógico alto em *enable*, ele atualiza os barramentos de saída *HEX\** conforme o valores empacotados em *bcd\_packet*. Os valores em *bcd\_packet* serão codificados conforme a tabela abaixo.

Símbolo a ser exibido	Valor esperado
0	0x0
1	0x1
2	0x2
3	0x3
4	0x4
5	0x5
6	0x6
7	0x7
8	0x8
9	0x9
'_'	0xA
apagardisplay	0xB

## Definição do Display:

```
module display (  
    input      logic      clk,  
    input      logic      rst,  
    input      logic      enable_o, enable_s,  
    input      bcdPac_t    bcd_packet_operacional, bcd_packet_setup,  
    output     logic [6:0]  HEX0, HEX1, HEX2, HEX3, HEX4, HEX5  
);  
  
endmodule
```

entradas e saídas

*****	tipo	significado
-------	------	-------------

enable_o	logic	habilita o módulo, indicado que ele pode atualizar um bcd conforme os valores em bcd_packet_operacional
enable_s	logic	habilita o módulo, indicado que ele pode atualizar um bcd conforme os valores em bcd_packet_setup
bcd_packet_operacional	bcdPac_t	valores empacotados no formato bcd que vem do módulo operacional
bcd_packet_setup	bcdPac_t	valores empacotados no formato bcd que vem do módulo setup
HEX*	logic [5:0]	Saída que vai ser direcionada a um display.

## Módulo Setup:

O módulo de Setup é responsável por lidar com todo o comportamento não operacional. Sua responsabilidade é atualizar os dados de configuração do sistema. Ou seja, atualizar os parâmetros de tempo do sistema e todas as suas senhas como previamente descrito no documento.

## Definição da Setup:

```
module setup (
    input      logic      clk,
    input      logic      rst,
    input      logic      setup_on,
    input      senhaPac_t  digitos_value,
    input      logic      digitos_valid,
    output     logic      display_en,
    output     bcdPac_t    bcd_pac,
    output     setupPac_t  data_setup_new,
    output     logic      data_setup_ok
);
```

endmodule

Entradas e saídas do módulo:

Nome	Tipo	Significado
clk	logic	Sinal de clock do sistema
rst	logic	Sinal de reset do sistema
setup_on	logic	Sinal para informar que habilita o modulo setup para realizar suas operações
digitos_value	senhaPac_t	Barramento de entrada com os últimos 20 dígitos do teclado, só deve ser lido quando ocorrer uma transição ascendente de digitos_valid

## Módulo Operacional:

O módulo operacional é responsável por executar todo o comportamento funcional do sistema. Sendo ele quem controla a **tranca**, o **bip** e define quando o módulo setup deve ser ativado de acordo com o comportamento descrito previamente no módulo setup. A forma como ele deve se comportar em relação à tranca e o bip são definidos pelos valores recebidos no barramento `data_setup_new`, obviamente após um reset, o módulo deve operar com os valores padrões previamente definidos no documento.

Quando o módulo operacional estiver esperando por senhas, os valores “0xF” e “0xE” no barramento `digitos_value` tem um significado especial. O valor “0xF” representa a ausência de dígito, já o valor “0xE” serve como indicativo de que o usuário demorou para inserir uma senha, ou desistiu de fazê-la, de toda forma o módulo operacional deve emitir um pulso de bip nesses casos.

## Definição do Operacional:

```
module operacional(
    input      logic      clk,
    input      logic      rst,
    input      logic      sensor_contato,
    input      logic      botao_interno,
    input      logic      botao_bloqueio,
    input      logic      botao_config,
    input      setupPac_t  data_setup_new,
    input      logic      data_setup_ok,
    input      senhaPac_t  digitos_value,
    input      logic      digitos_valid,
    output     bcdPac_t    bcd_pac,
    output     logic       teclado_en,
    output     logic       display_en,
    output     logic       setup_on,
    output     logic       tranca,
    output     logic       bip
);
endmodule
```

Entradas e saídas do módulo:

Nome	Tipo	Significado
clk	logic	clock do sistema
rst	logic	Reset do sistema
sensor_contato	logic	Pino de entrada para o sinal do sensor de contato
botao_interno	logic	Pino de entrada para o sinal vindo do botao_interno para travar ou destravar a porta
botao_bloqueio	logic	Pino de entrada para o sinal vindo do botão de Bloqueio
botao_config	logic	Pino de entrada para o sinal vindo do botão de Config
data_setup_new	setupPac_t	Barramento com os dados de configuração do funcionamento operacional
data_setup_ok	logic	Pino de entrada que informa que os novos dados já podem ser lidos
digitos_value	senhaPac_t	Barramento de entrada com os últimos 20 dígitos do teclado, só deve ser lido quando ocorrer uma transição ascendente de digitos_valid
digitos_valid	logic	Sinal para informar que os dados do teclado podem ser lidos.
bcd_pac	bcdPac_t	Barramento com os valores de atualização do módulo Display
display_en	logic	Pino de saída para atualização do módulo Display
teclado_en	logic	Pino de saída que define se o teclado deve tratar as entradas do teclado ou não
setup_on	logic	Pino para ativar o módulo setup
tranca	logic	Pino para controlar a trava
bip	logic	Pino para controlar o bip

## ResetHold5s:

Tem como função garantir que o reset foi mantido ao nível lógico alto por ao menos 5 segundos antes de enviá-lo ao sistema.

## Definição do ResetHold5s:

```
module resetHold5s #(parameter TIME_TO_RST = 5)
    (input logic clk, reset_in,
    output logic reset_out);
```

```
endmodule
```

Entradas e saídas do módulo:

*****	tipo	significado
reset_in	logic	entrada do reset do sistema que deve ser contado até atingir a marca pré-declarada
reset_out	logic	Saída do sistema com função de resetar todos os outros módulos do sistema.