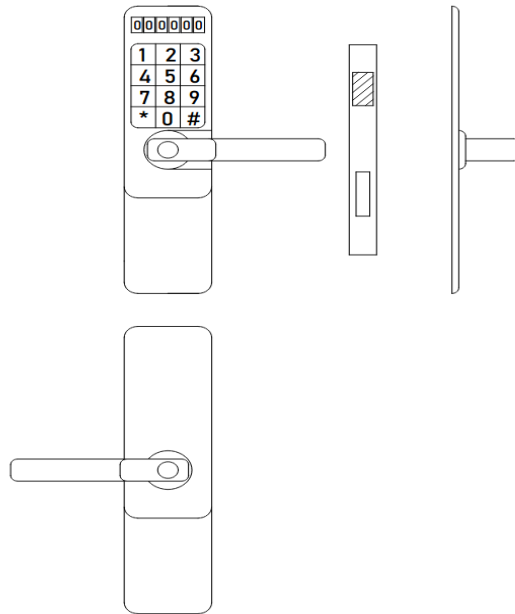


Projeto – Fechadura Eletrônica

Diagrama do Sistema:



Resumo:

Nosso sistema se trata de uma fechadura eletrônica no modelo sobrepor. A fechadura possui a capacidade de armazenar até 4 pins 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. Podendo até mesmo realizar seu próprio travamento após um curto período em que se encontre fechada.

Os componentes do sistema:

1. Um teclado matricial 4×4
 - a. Usado para inserir os pins que liberam o acesso
2. Seis displays de 7 segmentos
 - a. No modo padrão são responsáveis por exibir os dígitos inseridos pelo teclado matricial
 - b. No modo de configuração é responsável por exibir a opção atual ou seu estado e consequentemente as alterações.
3. Uma chave de contato
 - a. Responsável por indicar para o sistema se a porta se encontra aberta ou fechada.
4. Um botão
 - a. Responsável por liberar a tranca pelo lado de dentro.
5. Um LED
 - a. Para indicar se a fechadura está trancada ou não.

Sobre o funcionamento do sistema:

O sistema é baseado em dois modos simples, um modo operacional e um modo de configuração.

Modo operacional:

- Durante o modo operacional nosso sistema segue um fluxo simples de operações, ele sempre fica a espera da inserção de um pin (Padrão / Master) vindo pelo teclado matricial, cada tecla pressionada é exibida em sequência, começando no display HEX0 e indo até o display HEX4, em outras palavras da direita para esquerda. Os displays HEX4 e HEX5 ficaram apagados.
- A pin que foi inserido só será verificado após o pressionamento da tecla “*”.
- Uma vez que se faz o recebimento de um pin padrão válido, a fechadura deve ser destrancada, permitindo com que o usuário possa abrir sua porta.
- Uma vez com a tranca aberta, caso a porta se encontre aberta, será iniciado uma contagem de 5 segundos (tempo padrão, 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, 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 resetado e iniciado novamente a contagem para o trancamento automático.
- Se a fechadura receber um pin inválido, ela ficará inoperante por um tempo fixo de 1 segundo, durante esse segundo, deverá ser mostrado nos displays de 7 segmentos o traço, por exemplo “- - - - -”. Após 3 tentativas incorretas a fechadura exige uma pausa crescente entre os grupos de tentativas, com os tempos de 10, 20 e 30 segundos. Observação, após totalizar 5 tentativas erradas, o sistema deve aguardar 30 segundos entre cada tentativa futura. Obviamente, durante as pausas entre cada erro, a fechadura deve desconsiderar qualquer entrada realizada pelo teclado matricial.
- O tempo de espera entre cada tentativa será resetado, somente se, a inserção de um pin válido, ou reset for inserido.
- 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.
- E por fim, mediante a inserção do Master pin, a fechadura deverá sair de seu modo de operação padrão e entrar no modo de configuração.

Modo de setup:

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/desativar o bip
2. Definir tempo para bipar
3. Definir o tempo para realizar o trancamento da fechadura automaticamente.
4. Gerenciar senhas.

Ordem de entrada dos dados de configuração:

- 01 - Ativar BIP (padrão: ativado)
- 02 - Tempo BIP (padrão: 5s)
- 03 - Tempo Fechamento Automático (padrão: 5s)
- 04 - Senha Padrão 1 (padrão: ativado, valor '0000')
- 05 - Ativar senha 2 (padrão: desativado)
- 06 - Senha Padrão 2 (padrão: desativado)
- 07 - Ativar senha 3 (padrão: desativado)
- 08 - Senha Padrão 3 (padrão: desativado)
- 09 - Ativar senha 4 (padrão: desativado)
- 10 - Senha Padrão 4 (padrão: desativado)

01 - 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 e HEX4, devem aparecer os valores 0 e 1 respectivamente. 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.

02 ou 03 - 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.
- Nos displays HEX5 e HEX4, devem aparecer os valores (0 e 2) ou (0 e 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.

Gerenciar senhas:

- O gerenciador de senhas é responsável por ativar, desativar e alterar os 4 pins. As operações assim como no menu anterior são executadas de forma sequencial, ou seja, para alterar o quarto pin, deve-se passar pelo outros três.
- O usuário ativa ou desativa a função via teclado matricial, a tecla “1” será usada como indicador de que o pin deve estar ativado. Por sua vez, a tecla “0” deverá desativar o pin. Se alguma das duas teclas 0 ou 1 for pressionada, o valor deverá ser atualizado no display HEX0.
- O pin 1, não pode ser desativado.
- Os displays HEX5 e HEX4 deverão mostrar a respectiva função.
- Será armazenado sempre o valor que estiver sendo mostrado no displays de HEX3 à HEX0.
- 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.
- Uma vez resetado, se for digitado o pin Master padrão (1234) o sistema solicitará que seja atualizado a senha Master. Observação, é possível manter a mesma senha (1234). Vale ressaltar que a única forma de alterar o master pin é mediante ao reset. Nos displays HEX3 a HEX0 deverá ficar apagada aguardando a nova senha. E nos displays HEX5 e HEX4 deverão mostrar – e –. Obrigatoriamente o usuário deverá digitar 4 números, se for uma quantidade menor de número, o sistema deve permanecer aguardando a digitação dos demais números até completar os 4 dígitos.
- A tecla “*” por sua vez é responsável por confirmar as modificações feitas. Uma vez confirmado.
- Haverá um único pin padrão ativo, no caso o pin 1, contendo o valor (0000). O bip estará ativado, e os tempos para bipar e trancar a fechadura estarão em seus valores mínimos, no caso, 5 segundos.

Tipos pré-definidos

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

PinPacked_t:

Tipo de dado usado para empacotar as informações do pin, ou seja seu valor e se o mesmo se encontra ativo, inativo e seu valor.

```
typedef struct packed {  
    logic status;  
    logic [3:0] digit1;  
    logic [3:0] digit2;  
    logic [3:0] digit3;  
    logic [3:0] digit4;  
} pinPac_t;
```

Nome	Tipo	Significado
status	logic	indica se o pin está ativo ou inativo.
digit*	logic[3:0]	valor em hexadecimal dos dígitos do pin

bcdPacked_t:

Tipo de dado usado para empacotar os valores dos displays no formato BCD.

```
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;
```

SetupPacked_t:

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

```
typedef struct packed {  
    logic bip_status;  
    logic [6:0] bip_time;  
    logic [6:0] tranca_aut_time;  
    pinPac_t master_pin;  
    pinPac_t pin1;  
    pinPac_t pin2;  
    pinPac_t pin3;  
    pinPac_t pin4;
```

```
} setupPac_t
```

Nome	Tipo	Significado
bip_status	logic	indica se sistema deve ou não imitar algum bipe quando a porta se encontrar aberta (0 não / 1 sim)
bip_time	logic[5:0]	espaço reservado para informar o tamanho 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.
master_pin	pin_pac_t	pin_packet do master pin, verificar tipo Pin_packet_t para mais informações
pin*	pin_pac_t	pin_packet dos pin padrões verificar tipo Pin_packet_t para mais informações

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,  
    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.
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.
matricial_conf_out	logic[3:0]	Saída referente a configuração do teclado matricial
tranca	logic	Saída referente ao estado da tranca (0 aberta / 1 fechada).
bip	logic	Saída referente ao estado do bip (0 desativado / 1 ativado).

tabela.1

Operacional:

O módulo operacional é o bloco do sistema responsável por lidar com as responsabilidades de funcionamento operacional padrão, já descrito anteriormente neste documento. De certa forma, assim como o FechaduraTop ele é uma casca para subsistemas internos que trabalhando juntos devem retornar o que é esperado do sistema.

Definição do operacional:

```
module operacional (  
    input  logic      clk,  
    input  logic      rst,  
    input  logic      sensor_de_contato,  
    input  logic      botao_interno,  
    input  logic      key_valid,  
    input  logic [3:0] key_code,  
    output bcdPac_t    bcd_out,  
    output logic       bcd_enable,  
    output logic       tranca,  
    output logic       bip,  
    output logic       setup_on,  
    input  logic       setup_end,  
    output setupPac_t  data_setup_old,  
    input  setupPac_t  data_setup_new  
);  
  
endmodule;
```

entradas e saídas do módulo:

Nome	tipo	significado
Sensor_de_contato	logic	entrada referente ao sensor de contato da fechadura
botao_interno	logic	entrada referente ao botão interno da fechadura
key_valid	logic	indica que o valor no barramento key_code é um valor valido.
key_code	logic [3:0]	indica o valor em bcd decodificado pelo MatrixKeyDecoder, consultar modulo MatrixKeyDecoder para mais informações
bcd_out	bcdPac_t	Saída referente aos displays em formato bcd, para mais informações consultar tabela em SixDigit7SegCtrl e o tipo bcdpPacked_t
bcd_enable	logic	indica que se deve atualizar os hexadecimais com os valores do bcdPacked_t
tranca	logic	Saída referente ao estado em que a fechadura se encontra (0 fechada / 1 aberta)

bip	logic	Saída referente ao bip emitido pelo sistema (1 bipando / 0 desligado)
data_setup_new	setupPac_t	entrada referente aos valores atualizados pelo módulo setup, consultar tipo setupPacked_t para mais informações
data_setup_old	setupPac_t	saída referente aos valores atuais para o módulo setup
setup_on	logic	Saída referente a ativação do módulo de setup
setup_end	logic	entrada referente ao fim das operações de setup

Verificar senha:

Sub módulo responsável por verificar o pin entregue por montar pin é de alguma forma válido, seja um pin padrão ou master, ele também é responsável por saber se o sistema já atualizou o pin master após o reset. O módulo em questão recebe um pinPac_t do submodulo montar pin.

```

module verificar_senha (
    input logic          clk,
    input logic          rst,
    input pinPac_t       pin_in,
    input setupPac_t     data_setup
    output logic          senha_fail,
    output logic          senha_padrao,
    output logic          senha_master,
    output logic          senha_master_update
);

endmodule

```

Nome	tipo	significado
data_setup	setupPac_t	input responsável por receber o pacote de setup com os valores das senhas padrões e senha master atuais do sistema.
pin_in.digit*	logic [3:0]	valor em hexadecimal dos dígitos do pin
pin_in.status	logic	indica que os valores em pin_in.digit* foram confirmados e são um pin a ser verificado.
senha_fail	logic	indica que a senha recebida é inválida
senha_padrao	logic	indica que a senha recebida e uma senha padrão
senha_master	logic	indica que a senha recebida é uma senha master
senha_master_update	logic	flag que indica que a senha master ainda não foi atualizada (ainda não foi atualizado 1 / foi atualizado 0)

Update master:

Submódulo responsável por atualizar o master após a inserção do master pin(padrão) após um reset, ele só deve atualizar o PIN se receber um PIN com 4 dígitos válidos.

```
module update_master (
    input logic          clk,
    input logic          rst,
    input logic          enable,
    input pinPac_t       pin_in,
    output pinPac_t       new_master_pin
);

endmodule
```

Nome	tipo	significado
enable	logic	ativa o módulo para fazer suas respectivas operações
pin_in.digit*	logic[3:0]	dígitos do pin recebidos pelo submódulo montar pin
pin_in.status	logic	indica que a tecla “*” foi pressionada.
new_master_pin	pinPac_t	novo master pin quando ele for validado.
new_master_pin.status	logic	nesse caso estará sendo usado para indicar que a tecla “*” foi pressionada e que o valor do pin deve ser lido pelos demais submódulos.

Montar pin:

Sub módulo responsável por ficar recebendo as teclas do teclado já previamente decodificadas pelo submodulo **MatrixKeyDecoder**, montá-las no formato de pin e realizar o *shift* das teclas, para que os módulos operacional e setup consumam os dados que lhe interessam. Quando for pressionado a tecla “*”, o *pin_out.status* deve ser igual a 1 para sinalizar ao módulo **verificar_senha** e demais submodulos que utilizam o **montar_pin**, que deve ser lido o pinPac, e após um pulso de clock o pinPac.status deverá ser colocado em 0 e deverá ser aplicado o valor “E” nos pin_out.digit* .

definição do montar pin:

```
module montar_pin (
    input logic          clk,
    input logic          rst,
    input logic          key_valid,
    input logic [3:0]    key_code,
    output pinPac_t       pin_out
);

endmodule
```

Nome	tipo	significado
key_valid	logic	indica que o valor no barramento key_code é um valor válido.
key_code	logic [3:0]	indica o valor em bcd decodificado pelo MatrixKeyDecoder, consultar módulo MatrixKeyDecoder para mais informações
pin_out.digit*	logic [3:0]	valor em hexadecimal dos dígitos do pin
pin_out.status	logic	nesse caso estará sendo usado para indicar que a tecla “*” foi pressionada e que o valor do pin deve ser lido pelos demais submódulos.

Setup:

O módulo setup é o módulo responsável por todo o comportamento referente ao modo de configurações e suas funcionalidades já compreendidas anteriormente no documento.

Definição do setup:

```
module setup(
    input  logic      clk,
    input  logic      rst,
    input  logic      key_valid,
    input  logic [3:0] key_code,
    output bcdPac_t    bcd_out,
    output logic       bcd_enable,
    output setupPac_t  data_setup_new,
    input  setupPac_t  data_setup_old,
    input  logic       setup_on,
    output logic       setup_end
);

endmodule;
```

Nome	tipo	significado
key_valid	logic	indica que o valor no barramento key_code é um valor válido.
key_code	logic [3:0]	indica o valor em bcd decodificado pelo MatrixKeyDecoder, consultar módulo MatrixKeyDecoder para mais informações
bcd_out	bcdPac_t	Saída referente aos displays em formato bcd, para mais informações consultar tabela em SixDigit7SegCtrl e o tipo bcdPac_t
bcd_enable	logic	indica que se deve atualizar os hexadecimais com os valores do bcdPac_t

data_setup_new	setupPac_t	saída dos valores de setup atualizados para o módulo operacional, consultar tipo setupPacked_t para mais informações
setup_on	logic	entrada que indica a ativação do módulo setup
setup_end	logic	indica que o módulo setup terminou suas rotinas.

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.

tabela.2

MatrixKeyDecoder:

Módulo responsável por lidar com o teclado matricial, configurando sua leitura, realizando debounce para evitar ruídos no sistema e decodificando o resultado de forma interpretável ao sistema. Segue os valores retornados ao sistema.

Tecla Pressionada	Valor
0	0x0
1	0x1
2	0x2

3	0x3
4	0x4
5	0x5
6	0x6
7	0x7
8	0x8
9	0x9
A	0xA
B	0xB
C	0xC
D	0xD
#	0xE
*	0xF

Definição do MatrixKeyDecoder:

```

module matrixKeyDecoder (
    input clk, reset,
    input  logic [3:0] col_matrix,
    output logic [3:0] lin_matrix,
    output logic [3:0] tecla_value,
    output logic tecla_valid);

```

```
endmodule;
```

entradas e saídas do módulo:

*****	tipo	significado
col_matrix	logic[3:0]	Identificar a coluna
lin_matrix	logic[3:0]	Identificar a linha
tecla_valid	logic	Será gerado uma transição ascendente sempre que uma tecla for pressionada.
tecla_value	logic[3:0]	Código da tecla pressionada
clk	*	Clock do sistema
reset	*	Reset do sistema

tabela.3

SixDigit7SegCtrl:

SixDigit7SegCtrl é o módulo responsável por armazenar e exportar os valores dos displays de sete segmentos. Uma vez que o módulo é habilitado, ele atualiza os hexadecimais de acordo com o valores empacotados em bcd_packet,

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 SixDigit7SegCtrl:

```
module SixDigit7SegCtrl (  
    input logic clk,  
    input logic rst,  
    input logic enable,  
    input bcdPac_t bcd_packet,  
    output logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5  
);  
  
endmodule
```

entradas e saídas

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

enable	logic	habilita o modulo, indicado que ele pode atualizar um bcd consoante os valores das entradas
bcd_packet	bcdPac_t	valores empacotados no formato bcd
HEX*	logic [5:0]	Saída que vai ser direcionada a um display.