Curso de Ciência da Computação
CE4411 – Arquitetura de Computadores
Relatório de Projeto

Nome:	João Pedro Rosa Cezarino	R.A.: 22.120.021-5
Nome:	Vítor Martins Oliveira	R.A.: 22.120.067-8

Projeto de Arquitetura de Computadores

1. Descrição do Projeto:

O Projeto em questão tem o objetivo de reproduzir o conteúdo aprendido durante as aulas de Arquitetura de Computadores no Centro Universitário FEI. Durante o curso os alunos tiveram contato com a linguagem de programação Assembly, amplamente utilizada para manipulação direta de hardware, acesso a instruções especializadas do processador ou para resolver problemas críticos de desempenho. Para compreender e entender a linguagem foi utilizado um microcontrolador 8051 e o software EDSIM51D, excelente para emular as funções de um 8051 real.

O tema escolhido para aplicar os conceitos aprendidos em aula foi um Sistema de Desbloqueio de automóveis, baseado em senha. Essa implementação foi escolhida por possibilitar um melhor entendimento de algumas funções do Software de Simulação utilizado (EDSIM51d), como por exemplo o uso dos LEDs embarcados, Keypad, Displays e Motores.

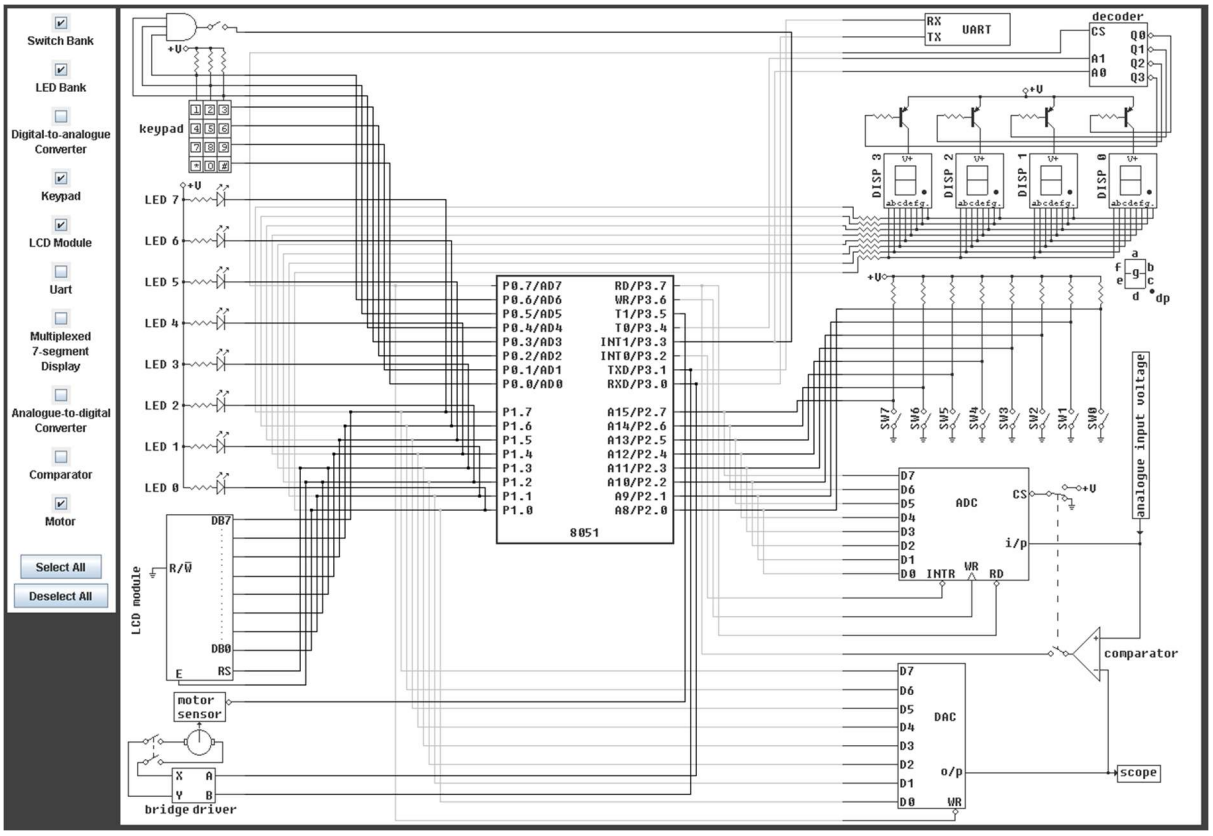
O funcionamento do sistema se dá da seguinte forma: Primeiramente, o sistema requisita ao usuário a inserção da senha (previamente definida no código-fonte), logo após a entrada, o sistema verifica se a senha inserida pelo usuário corresponde à que está registrada na memória. Caso a senha esteja correta, o sistema imprime “Acesso Liberado” no display e o carro é aberto, o motor presente no simulador é acionado representando o movimento das rodas do carro em geral. Se a senha inserida estiver incorreta, o sistema emitirá a mensagem “Acesso Negado” no display e um dos LEDs vermelho será acesso, imitando a ação de um alarme antifurto de um automóvel real.

Este sistema pode ser muito útil nos veículos atuais, já que com o crescente índice de roubos de carros, possuir um sistema desse nível dificulta a ação dos ladrões e acaba evitando esse tipo de situação. Além de adicionar uma camada extra de segurança ao veículo e aos passageiros.

Link do Projeto: https://youtu.be/RWwI3KKA5Cw

Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo “Pulse”

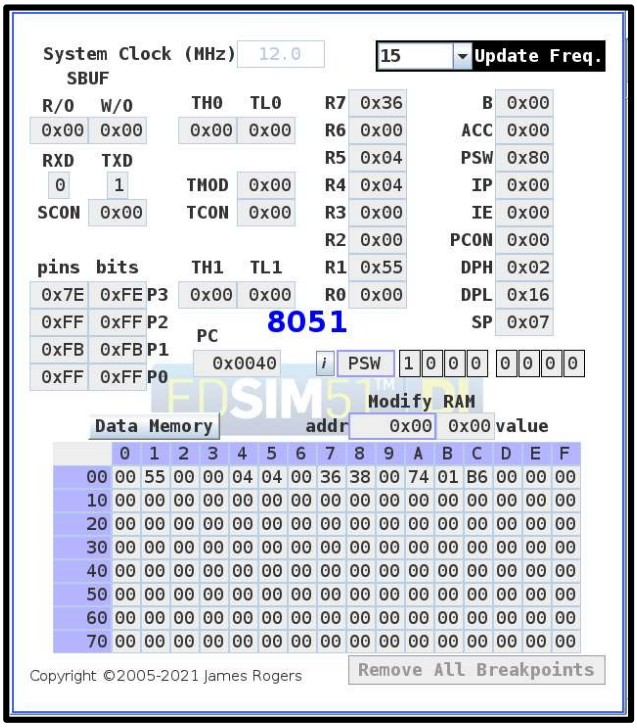
2. Desenho esquemático:



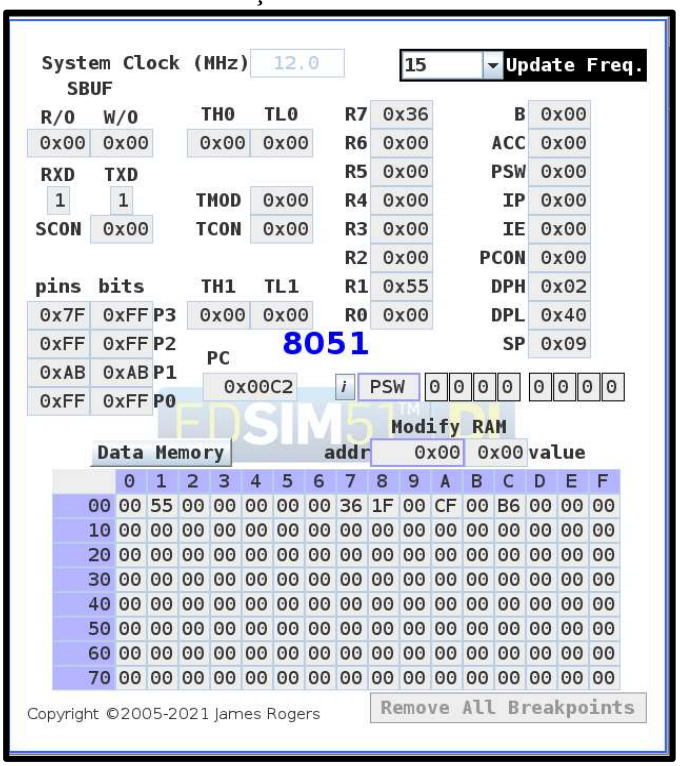
Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo "Pulse"

3. Fluxograma ou Diagrama:

- Posições de memória após a inserção da senha e com o acesso liberado:

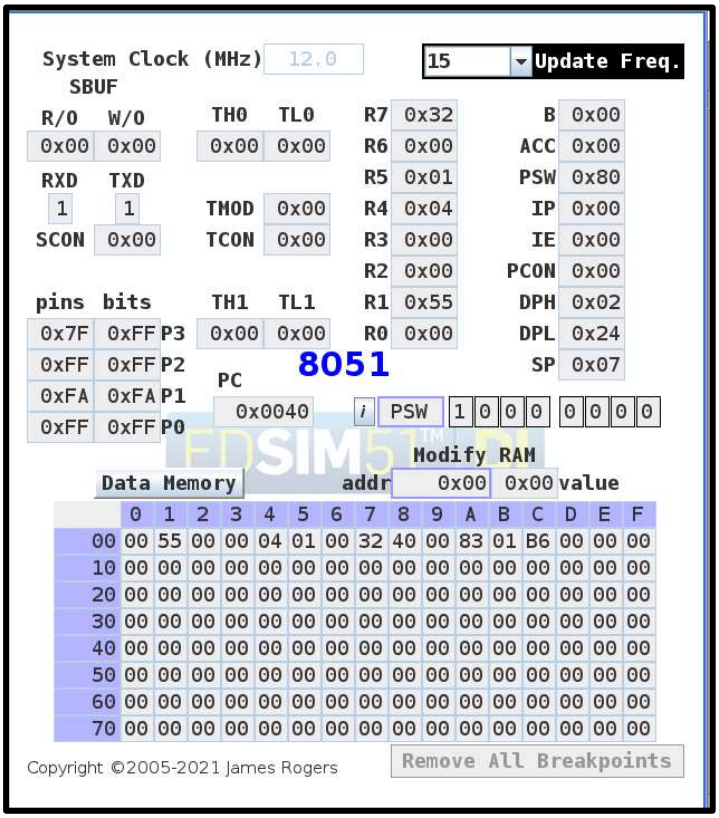


- Posições de memória antes da inserção da senha:



Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo “Pulse”

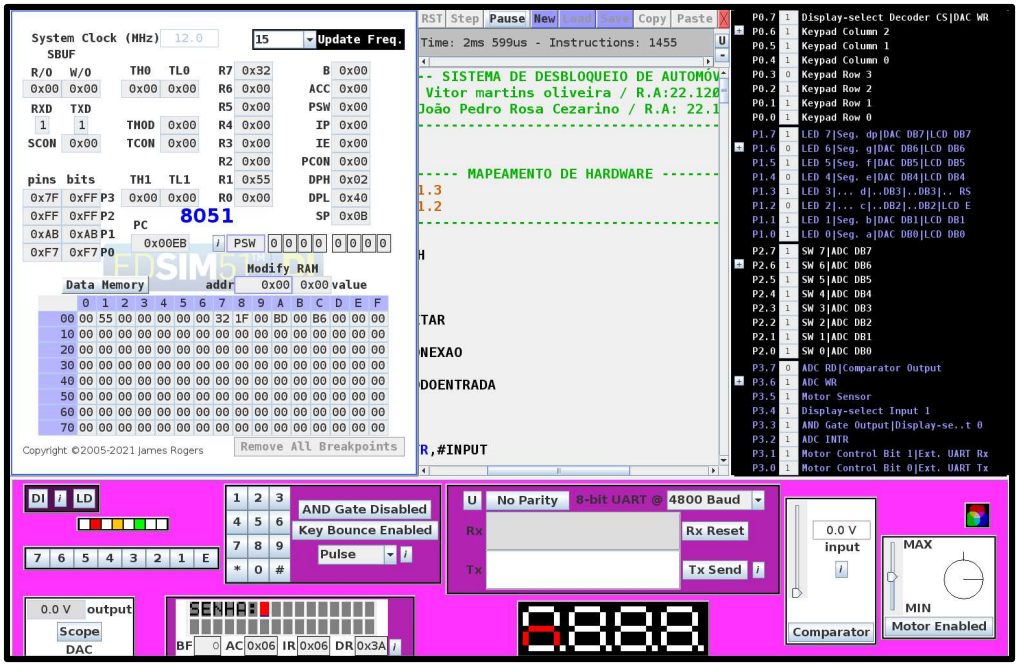
- Posições de memória após a inserção da senha e com o acesso Negado:



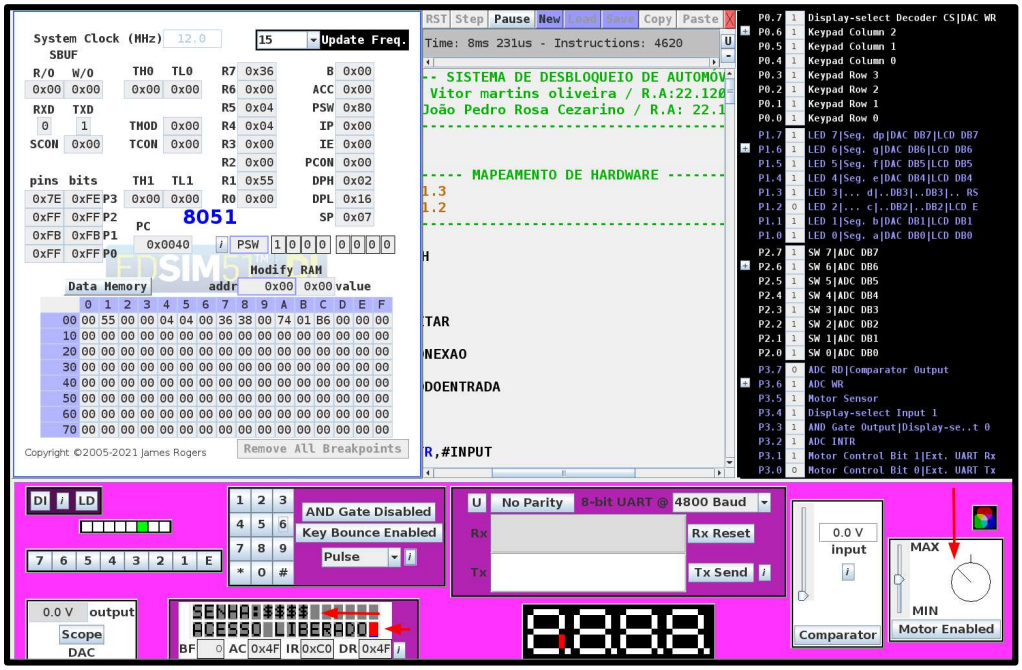
<u>Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo “Pulse”</u>

4. Imagens da simulação realizada na IDE:

- Sistema requisitando a senha para o desbloqueio do carro:

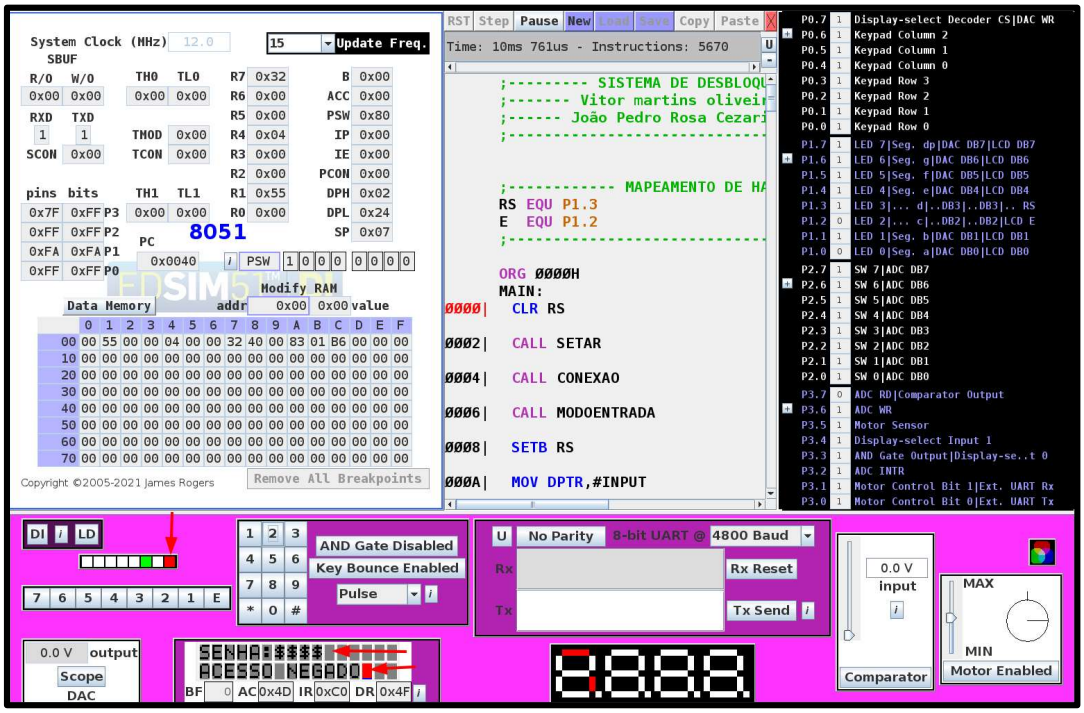


- Sistema após a inserção da senha correta (repare no motor girando e na mensagem impressa no display):



Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo "Pulse"

- Sistema após a inserção da senha incorreta (O motor permanece parado e o LED vermelho acende, imitando um alarme antifurto). O display exibe a mensagem “Acesso Negado”.



<u>Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo “Pulse”</u>

5. Discussões e conclusões

O projeto possibilitou à dupla um entendimento mais profundo a respeito de uma linguagem de baixa abstração e bem próxima ao hardware em si. Durante o desenvolvimento do projeto houve alguns impasses no decorrer da criação dos códigos. A ideia inicial para o projeto era de um Semáforo, porém, com o decorrer do projeto, percebemos que não seria possível programá-lo da forma desejada, isso porque não seria possível utilizar os LEDs necessários em conjunto com o display, já que o display utiliza alguns dos LEDs que seriam chave para o nosso projeto.

Após alguns dias, resolvemos trocar o tema do projeto e o escolhido foi o Sistema de Desbloqueio para automóveis. O tema chamou nossa atenção e decidimos seguir em frente com ele. Após realizar a troca, o projeto correu de forma natural e sem impasses. Foi possível implementar e utilizar grande parte das instruções aprendidas em aula no programa. A inspiração para a criação de um Sistema de desbloqueio surgiu a partir de carros norte americanos que já possuem uma tecnologia deste tipo, como por exemplo o Ford Fusion.

Após o desenvolvimento deste projeto, ambos entenderam como a linguagem Assembly é de fato utilizada no nosso dia a dia e como e ocorrem a movimentação dos bits e bytes dentro de um processador/microcontrolador e como as tarefas chegam até ele. aprender linguagens de baixa abstração é extremamente importante para que possamos compreender como as linguagens atuais realizam tantas operações e facilitam o desenvolvimento de código em geral. A linguagem ensina o que a CPU está fazendo e como funciona a execução de suas operações em cada instante. Certamente, entender um pouco de Assembly dá uma visão muito interessante de como são executadas as tarefas para as quais estão programadas.

Acreditamos que a linguagem Assembly, apesar de antiga, não está defasada. Isso porque, ela ainda é extremamente necessária para o desenvolvimento de hardwares e processadores, que nos auxiliam no dia a dia. Para entender a linguagem, não é necessário conhecer “bem” o hardware, mas sim dominar a linguagem e suas instruções (que variam de microcontrolador para microcontrolador). Apesar de não muito requisitada no mercado de trabalho, aprender o funcionamento das instruções de um processador pode ser um conhecimento diferencial e agregador para um profissional de T.I. Além disso, a grande maioria das vagas que requerem um conhecimento nesta linguagem possuem boas oportunidades e benefícios em empresas reconhecidas internacionalmente.

Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo “Pulse”

6. Código-fonte

```
;--------- SISTEMA DE DESBLOQUEIO DE AUTOMÓVEIS ---------;
;------- Vitor martins oliveira / R.A:22.120.067-8 ------;
;------ João Pedro Rosa Cezarino / R.A: 22.120.021-5 ----;
;-------------------------------------------------------;


;------------ MAPEAMENTO DE HARDWARE -------------;
RS EQU P1.3
E  EQU P1.2
;-------------------------------------------------------;

ORG 0000H
MAIN:
        CLR RS


        CALL SETAR

        CALL CONEXAO

        CALL MODOENTRADA


        SETB RS


        MOV DPTR,#INPUT

DENOVO:
        CLR A
        MOVC A,@A+DPTR
        JZ PROXIMA

        CALL ENVIA
        INC DPTR
        JMP DENOVO

PROXIMA:
        MOV R4,#00H
        MOV R5,#00H

        MOV DPTR,#PASSWORD

REPETIR:
        CALL ESCANEIA
        SETB RS

        CLR A
        MOV A,#'$'
        CALL ENVIA

        CLR A
        MOVC A,@A+DPTR
        CALL VERIFICAENTRADA

        INC DPTR
        INC R4
        CJNE R4,#04H,REPETIR
        CJNE R5,#04H,ERRADO

;----------------------- SENHA CERTA ----------------------;
CERTO:
        CALL CURSORPOS
        SETB RS

        CALL PERMITIDO
        JMP TERMINA

;----------------------- SENHA ERRADA ---------------------;
ERRADO:
        CALL CURSORPOS
        SETB RS

        CALL NEGADO

TERMINA:
        JMP $

SETAR:
        CLR  P1.7
        CLR  P1.6
        SETB P1.5
        CLR  P1.4
```

Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo "Pulse"

```
        CALL PISCA

        CALL DELAY

        CALL PISCA


        SETB P1.7
        CLR  P1.6
        CLR  P1.5
        CLR  P1.4

        CALL PISCA

        CALL DELAY
        RET

CONEXAO:
        CLR P1.7
        CLR P1.6
        CLR P1.5
        CLR P1.4

        CALL PISCA

        SETB P1.7
        SETB P1.6
        SETB P1.5
        SETB P1.4
        CALL PISCA

        CALL DELAY
        RET

MODOENTRADA:
        CLR P1.7
        CLR P1.6
        CLR P1.5
        CLR P1.4

        CALL PISCA

        CLR  P1.7
        SETB P1.6
        SETB P1.5
        CLR  P1.4

        CALL PISCA

        CALL DELAY
        RET

PISCA:
        SETB E
        CLR  E
        RET

ENVIA:
        MOV C, ACC.7
        MOV P1.7, C
        MOV C, ACC.6
        MOV P1.6, C
        MOV C, ACC.5
        MOV P1.5, C
        MOV C, ACC.4
        MOV P1.4, C

        CALL PISCA

        MOV C, ACC.3
        MOV P1.7, C
        MOV C, ACC.2
        MOV P1.6, C
        MOV C, ACC.1
        MOV P1.5, C
        MOV C, ACC.0
        MOV P1.4, C

        CALL PISCA

        CALL DELAY


        MOV R1,#55H
        RET

;---------------- MAPEANDO O KEYPAD
------------------------;

ESCANEIA:
        CLR P0.3
        CALL IDTECLA0
        SETB P0.3
        JB F0,FEITO
```

Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo "Pulse"

```
        CLR P0.2
        CALL IDTECLA1
        SETB P0.2
        JB F0,FEITO

        CLR P0.1
        CALL IDTECLA2
        SETB P0.1
        JB F0,FEITO

        CLR P0.0
        CALL IDTECLA3
        SETB P0.0
        JB F0,FEITO



        JMP ESCANEIA


FEITO:
        CLR F0
        RET

IDTECLA0:
        JNB P0.4, TECLA03
        JNB P0.5, TECLA02
        JNB P0.6, TECLA01
        RET


TECLA03:
        SETB F0
        MOV R7,#'3'
        RET

TECLA02:
        SETB F0
        MOV R7,#'2'
        RET
TECLA01:
        SETB F0
        MOV R7,#'1'
        RET

IDTECLA1:
        JNB P0.4, TECLA06
        JNB P0.5, TECLA05
        JNB P0.6, TECLA04
        RET

TECLA06:
        SETB F0
        MOV R7,#'6'
        RET

TECLA05:
        SETB F0
        MOV R7,#'5'
        RET

TECLA04:
        SETB F0
        MOV R7,#'4'
        RET

IDTECLA2:
        JNB P0.4, TECLA09
        JNB P0.5, TECLA08
        JNB P0.6, TECLA07
        RET


TECLA09:
        SETB F0
        MOV R7,#'9'
        RET

TECLA08:
        SETB F0
        MOV R7,#'8'
        RET

TECLA07:
        SETB F0
        MOV R7,#'7'
        RET

IDTECLA3:
        JNB P0.4, TECLAHASHTAG

        JNB P0.5, TECLA00
        JNB P0.6, TECLAAST
```

Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo “Pulse”

```
        RET


TECLAHASHTAG:
        SETB F0
        MOV R7,#'#'
        RET

TECLA00:
        SETB F0
        MOV R7,#'0'
        RET

TECLAAST:
        SETB F0
        MOV R7,#'*'
        RET
;-------------------------------------------------
----------;

VERIFICAENTRADA:
        CJNE A,07H,SAIDA
        INC R5

SAIDA:

        RET

CURSORPOS:
        CLR RS
        SETB P1.7
        SETB P1.6
        CLR P1.5
        CLR P1.4


        CALL PISCA

        CLR P1.7
        CLR P1.6
        CLR P1.5
        CLR P1.4


        CALL PISCA

        CALL DELAY
        RET

DELAY:
        MOV R0, #50
        DJNZ R0, $
        RET

PERMITIDO:
        MOV DPTR,#ACSLDO
        CLR P3.0

VOLTAR:
        CLR A
        MOVC A,@A+DPTR
        JZ COMECO
        CALL ENVIA
        INC DPTR
        JMP VOLTAR

COMECO:
        RET

NEGADO:
        MOV DPTR,#ACSNDO
        CLR P1.0

OUTRAVEZ:
        CLR A
        MOVC A,@A+DPTR
        JZ VOLTACOMECO
        CALL ENVIA
        INC DPTR
        JMP OUTRAVEZ

VOLTACOMECO:
        RET


;------------------------- ARMAZENANDO
AS STRINGS ---------------------------------
-;
ORG 0200H
INPUT: DB 'S', 'E', 'N', 'H', 'A', ':',0
ACSLDO: DB 'A', 'C', 'E', 'S', 'S', 'O', ' ',
'L', 'I', 'B', 'E', 'R', 'A', 'D', 'O', 0
ACSNDO:     DB 'A', 'C', 'E', 'S', 'S', 'O', '
', 'N', 'E', 'G', 'A', 'D', 'O', 0
```

<u>Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo “Pulse”</u>

```
;--------------------------- SENHA ------------
---------------------------------------;
ORG 0240H
PASSWORD: DB '9', '8', '7', '6',0
```

PARA:

```
        JMP $

END
```

Importante: Para usar o projeto da melhor forma, deixe a frequência de update em 15 e o botão em modo “Pulse”