|  |  |
| --- | --- |
| **Curso |** | Técnico de Eletrónica, Automação e Comando |
| **Ano letivo |** | 2018/2019 |

|  |  |
| --- | --- |
| Relatório Final de PAP | |
| **Projeto>** | Controlo de acesso RFID com teclado |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do aluno | | Tiago Filipe Gonçalves Lima | | |
| Turma | | R | Nº | | 22 |
| Ano | | 12.° |
| Orientador de PAP | | Hélder Nuno Oliveira Cerqueira | | | |

Ponte da Barca, 30 de abril de 2019

Agradecimentos

Em primeiro lugar, agradeço ao meu orientador de PAP, professor Hélder Cerqueira, por me ter ajudado ao longo da minha Prova de Aptidão Profissional.

Um grande agradecimento, ainda, ao meu diretor de curso, professor Marco Lima, pela, também, total disponibilidade e apoio. À minha diretora de turma, professora Renata Duro pela preocupação.

A todos os professores que, ao longo destes três anos de formação, me proporcionaram e transmitiram todos os conhecimentos que me permitiram o meu enriquecimento a nível profissional e humano e, aos professores António Campos e Nelson Cerqueira pela disponibilidade que demonstraram em qualquer momento para esclarecimento de dúvidas que me foram surgindo.

Agradeço, também, a todos os meus colegas e família pela compreensão e paciência que demonstraram ao longo de várias etapas deste projeto.

A todos, o meu mais sincero obrigado!

Índice

[Introdução 4](#_Toc7954995)

[Objetivos 5](#_Toc7954996)

[Gerais 5](#_Toc7954997)

[Específicos 5](#_Toc7954998)

[Recursos necessários 6](#_Toc7954999)

[Ferramentas 6](#_Toc7955000)

[Equipamentos 6](#_Toc7955001)

[Componentes 6](#_Toc7955002)

[Software 7](#_Toc7955003)

[Descrição do Projeto 8](#_Toc7955004)

[Desenvolvimento 9](#_Toc7955005)

[Metodologia 9](#_Toc7955006)

[Princípio de funcionamento da PAP 9](#_Toc7955007)

[Enquadramento teórico 9](#_Toc7955008)

[Arduíno Mega 2560 9](#_Toc7955009)

[Módulo Leitor RFID MFRC522 10](#_Toc7955010)

[Teclado Matricial 4x3 11](#_Toc7955011)

[Display 16x2 12](#_Toc7955012)

[Cadsoft Eagle 12](#_Toc7955013)

[3D Builder 13](#_Toc7955014)

[Ultimaker CURA 13](#_Toc7955015)

[Enquadramento prático 14](#_Toc7955016)

[Processo de revelação da PCI 15](#_Toc7955017)

[Insolação 15](#_Toc7955018)

[Revelação 15](#_Toc7955019)

[Corrosão 15](#_Toc7955020)

[Limpeza 15](#_Toc7955021)

[Impressão 3D 15](#_Toc7955022)

[Orçamento 16](#_Toc7955023)

[Conclusão 17](#_Toc7955024)

[Bibliografia 18](#_Toc7955025)

[Anexos 19](#_Toc7955026)

**Índice de figuras**

[Figura 1 – Arduíno Mega 2560 10](#_Toc7681617)

[Figura 2 – Módulo leitor RFID MFRC522 11](#_Toc7681618)

[Figura 3 – Teclado matricial membrana 4x3 12](#_Toc7681619)

[Figura 4 – Display 16x2 13](file:///C:\Users\Utilizador\MEOCloud\EPRALIMA%202018-19\R%20Eletrónica\R%20PAP\_Relatórios\2019-05-02\22_Tiago_Lima_relatorio%20final%20pap.docx#_Toc7681620)

[Figura 5 – Software Eagle 13](file:///C:\Users\Utilizador\MEOCloud\EPRALIMA%202018-19\R%20Eletrónica\R%20PAP\_Relatórios\2019-05-02\22_Tiago_Lima_relatorio%20final%20pap.docx#_Toc7681621)

[Figura 6 – Software 3D Builder 14](#_Toc7681622)

[Figura 7 – Software Cura 14](#_Toc7681623)

[Figura 8 – Ligações feitas 15](#_Toc7681624)

Introdução

O presente projeto corresponde à última fase do Curso Profissional de Técnico de Eletrónica, Automação e Comando, a Prova de Aptidão Profissional tem como principal objetivo abranger todos os conhecimentos adquiridos ao longo dos três anos de formação. É, sem dúvida, a oportunidade única de demonstrar, sobretudo a nível prático, muitas das competências adquiridas nas áreas da eletrónica, programação, entre outras.

A minha Prova de Aptidão Profissional tem por base abrir uma porta com uma fechadura elétrica magnética por um leitor de cartões RFID, também integrei um teclado 4x3 e um botão de pressão.

No meu relatório pretendo apresentar toda a parte teórica do meu trabalho mostrando, assim, o material utilizado e os softwares utilizados.

Também vou falar e apresentar todo o trabalho prático que fiz ao longo dos últimos meses.

Objetivos

Gerais

* Aplicar os conhecimentos e das competências que adquiri ao longo do curso;
* Desenvolver a minha PAP de acordo com as normas de higiene aplicáveis ao contexto do meu projeto;
* Elaborar o relatório de PAP;
* Pesquisar informação sobre o tema.

Específicos

* Aprofundar os conhecimentos na área de programação Arduíno;
* Realização da programação do Arduíno;
* Realizar uma placa de circuito impresso;
* Realização da apresentação em PowerPoint;
* Construir peças em 3D.

Recursos necessários

Ferramentas

* Alicate de corte;
* Ferro de solda;
* Chupa-soldas;
* Chave de cruz;
* Chave de fendas;
* Parafusos;
* Fita métrica;
* Paquímetro;
* Régua.

Equipamentos

* Voltímetro;
* Arduíno Mega 2580;
* Módulo relé 5v;
* Módulo Leitor RFID MFRC522;
* Impressora 3D;
* Teclado matricial 4x3;
* Display 16x2;
* Fechadura elétrica magnética NC;
* Computador;
* Fonte de alimentação.

Componentes

* Botão de pressão;
* Resistências (10kΩ);
* Potenciómetro (25kΩ);
* Led.

Software

* Arduíno;
* 3D Builder;
* Ultimaker Cura;
* Microsoft Word;
* Microsoft PowerPoint;
* Microsoft Excel;
* Cadsoft Eagle.

Descrição do Projeto

A minha Prova de Aptidão Profissional consiste em abrir uma porta por controlo de acesso RFID e/ou por um teclado.

Após ter determinado, de forma definitiva, o tema da minha PAP, iniciei a fase de pesquisa de conteúdos sobre o rumo que iria dar ao meu projeto e todos os materiais que iriam ser necessários para a concluir.

Depois de ter realizado toda a pesquisa necessária, optei por iniciar as ligações e a programação. Primeiramente, para perceber mais acerca da programação em Arduíno, fiz pequenos testes em *breadboard*. Como primeiro programa, acendi um led com um botão.

Em seguida fiz uma programação de teste para o módulo RFID, para a identificação das ID dos cartões a utilizar.

Desenvolvimento

Metodologia

Para a concretização do meu projeto, baseei-me, sobretudo, na pesquisa de conteúdos de forma a complementar conhecimentos para a execução do meu projeto. Para tal, tive, também, a ajuda dos professores que me orientaram e ajudaram a tomar as melhores decisões.

Em seguida, depois de traçado o rumo que iria dar ao meu projeto, elaborei todas as montagens e programações necessárias.

Princípio de funcionamento da PAP

O princípio de funcionamento da minha Prova de Aptidão Profissional, é um controlo de acesso RFID com Teclado e/ou botão de pressão que realizam o controlo de um fecho por eletroíman.

Enquadramento teórico

Arduíno Mega 2560



Figura 1 – Arduíno Mega 2560

O Arduíno Mega é um microcontrolador baseado em ATMega1280, tem 54 pinos de entrada / saídas digitais (dos quais 14 podem ser usados como saídas PWM), 16 entradas analógicas, 4 UARTs (porta serial de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, um conetor de energia ICSP, e um botão de *reset*. Ele contém tudo o que é necessário para suportar o microcontrolador, basta conetá-lo a um computador com um cabo USB ou ligá-lo a um adaptador de CA-CC ou bateria para iniciar.

O Arduíno Mega pode ser alimentado via conexão USB ou com uma fonte de alimentação externa. A fonte de energia é selecionada automaticamente.

Alimentação

A alimentação externa é feita através do conector Jack, e deve ser alimentado entre 6V a 20V, porém, se for alimentada com uma tensão abaixo de 7V, a tensão de funcionamento da placa, que no Arduíno Uno é 5V, pode ficar instável e quando alimentada com tensão acima de 12V, o regulador de tensão da placa pode sobreaquecer e danificar a placa. Desta forma, é recomendado ser alimentada entre 7V a 12V.

Programação

O Arduíno Mega pode ser programado com o software Arduíno que se pode descarregar no site oficial da marca para qualquer sistema operativo Microsoft Windows, Mac ou Linux

Módulo Leitor RFID MFRC522

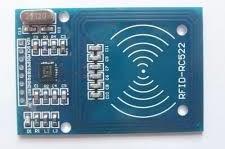


Figura 2 – Módulo leitor RFID MFRC522

Na interligação com o Arduíno foram utilizados os seguintes pinos:

3,3V – 3,3V;

RST – 8 Arduíno;

GND – GND;

IRQ – Não ligado;

MISO – 50 Arduíno;

MOSI – 51 Arduíno;

SCK – 52 Arduíno;

SDA – 9 Arduíno.

Para que o leitor RFID do modelo MFRC522 funcionasse tive de pesquisar o *datasheet* correspondente.

Este módulo de leitor RFID baseado no chip MFRC522 da empresa NXP é altamente utilizado em comunicação sem contato a uma frequência de 13,56MHz. Este chip, de baixo consumo e pequeno tamanho, permite sem contato ler e escrever em cartões que seguem o padrão *Mifare*, muito usado no mercado.

Este módulo de leitor RFID possui as ferramentas necessárias para um projeto de controlo de acesso ou sistemas de segurança. As *tags* (ou etiquetas) RFID, podem conter vários dados sobre o proprietário do cartão, como nome e endereço e, no caso de produtos, informações sobre procedência e data de validade, apenas para citar alguns exemplos.

Teclado Matricial 4x3



Figura 3 – Teclado matricial membrana 4x3

Na interligação com o Arduíno foram utilizados os seguintes pinos:

Pino 1 – Pino 34 do Arduíno;

Pino 2 – Pino 35 do Arduíno;

Pino 3 – Pino 36 do Arduíno;

Pino 4 – Pino 37 do Arduíno;

Pino 5 – Pino 38 do Arduíno;

Pino 6 – Pino 39 do Arduíno;

Pino 7 – Pino 40 do Arduíno.

Display 16x2

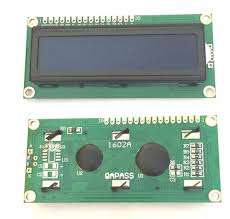
Na interligação com o Arduíno foram utilizados os seguintes pinos:

Figura 4 – Display 16x2

VSS – GND;

VDD - +5v;

VO – centro potenciómetro;

RS – 2 Arduíno;

RW – 3 Arduíno;

E – 3 Arduíno;

D4 – 5 Arduíno;

D5 - 6 Arduíno;

D6 - 7 Arduíno;

D7 - 8 Arduíno;

A – Resistência – polo positivo;

K – GND.

Cadsoft Eagle

O *Cadsoft* EAGLE é um software de automação de projeto eletrónico (APE). Permite que os projetistas de placa de circuito impresso (PCI) conetem perfeitamente diagramas esquemáticos, posicionamento de componentes, roteamento de PCB e conteúdo abrangente da biblioteca.



Figura 5 - Software Eagle

3D Builder

O 3D Builder é um software da Microsoft, grátis para Windows 10, 8.1 e Windows 10 mobile. Este é um programa que permite criar peças em 3D de forma simples, dado que inclui um catálogo de formas e objetos prontos a imprimir.



Figura 6 – Software 3D Builder

Ultimaker CURA

O Cura é um Software livre que prepara os objetos 3D para impressão. Para inexperientes, este facilita a obtenção de excelentes resultados. Para especialistas, existem mais de 200 configurações que se adequam às necessidades de cada um.



Figura 7 – Software Cura

Enquadramento prático

Em primeiro lugar tive de instalar os softwares Arduíno e Cadsoft Eagle 9.3.0 no meu computador.

Para que conseguisse fazer a minha Prova de Aptidão Profissional tive que aprender mais acerca da linguagem de programação C++, que é a linguagem que o Arduíno usa.hH

Fiz a soldagem dos pinos dos módulos Display 16x2, teclado matricial 4x3 e RFID MFRC522.

Depois da soldagem dos módulos, fiz a ligação dos pinos do display aos pinos 22, 23, 24, 25, 26, 27 e 28 do Arduíno; do módulo RFID pinos 3,3v, 8, 9, GND, 50, 51 e 52 do Arduíno.

Integrei um LED que quando se prime uma tecla, e a porta esteja aberta, o LED acende.

Quando as ligações já estavam todas nos devidos lugares, estive a fazer uma placa shield para o Arduíno no software *Cadsoft* Eagle.

No fim integrei um botão de reset externo, para que quando o programa travasse, faz o reset sem ter de abrir a caixa.

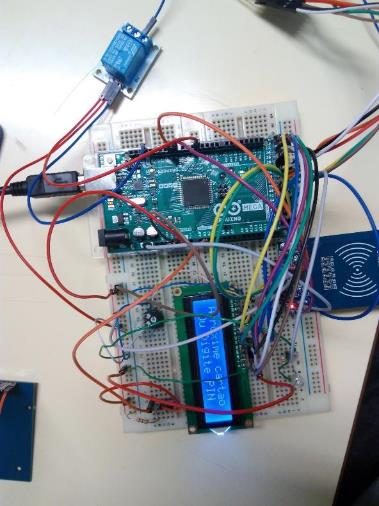


Figura 8 – Ligações feitas

Processo de revelação da PCI

Insolação

O processo de insolação consiste em aplicar a pelicula transparente sobre a placa de cobre pré-sensibilizada colocando-a numa máquina de insolação ultravioleta (U.V.) durante 90 segundos aproximadamente. Este tipo de luz provoca uma reação química com a parte fotossensível exposta.

Revelação

A revelação consiste no primeiro processo químico “Positive Stripper” durante alguns segundos. A solução irá desfazer o verniz que foi sensibilizado, isto é, o verniz exterior às pistas.

Corrosão

Finalmente, coloca-se a placa na solução de percloreto de ferro, até que o percloreto destrua o cobre que já não tem verniz, isto é, o cobre exterior às pistas.

Limpeza

Lava-se então a placa com água para remover todos os resíduos.

Limpa-se a placa com decapante, álcool ou acetona para retirar o verniz que cobre as pistas. Deixa-se secar

Finalmente, procede-se à abertura dos furos nos pontos indicados na matriz.

Limpa-se a placa e podemos ainda cobrir com o spray de verniz protetor, para evitar a oxidação.

Procede-se à respetiva soldadura dos componentes na outra face da placa.

Impressão 3D

Depois de construir a placa e fazer os devidos testes passei ao desenho em 3D para a construção da caixa. Para o desenho da caixa utilizei o Software 3D Builder e para a impressão da mesma precisei da impressora 3D Prusa i3.

Orçamento

Para a realização da minha Prova de Aptidão Profissional tive de fazer um orçamento com tudo o que precisava para a realização da mesma.

O orçamento da minha Prova de Aptidão Profissional ficou no total de 153€.

|  |  |  |  |
| --- | --- | --- | --- |
| **Qtd.** | **Designação** | **Preço** | |
| **Unitário** | **Total** |
| 2 | Teclado matricial 3x4 | 3.25 € | 6.50 € |
| 2 | Conjunto de cabos jumper M/M | 2.45 € | 4.90 € |
| 2 | Conjunto de cabos jumper F/M | 2.50 € | 5.00 € |
| 1 | Arduíno Mega2560 REV3 | 51.25 € | 51.25 € |
| 1 | Módulo Leitor RFid | 4.90 € | 4.90 € |
| 1 | Display LCD Alfanumérico Azul 16x2 - STN Negativo | 26.00 € | 26.00 € |
| 1 | Trinco Elétrico Magnético com Estado em LED | 42.50 € | 42.50 € |
| 1 | Filamento PLA | 9.50 € | 9.50 € |
| 1 | Módulo Relé 1 canal 5V compatível com Arduíno | 2.45 € | 2.45 € |
| **Total 2** | | | 153.00 € |

Conclusão

O projeto que desenvolvi no âmbito da Prova de Aptidão Profissional permitiu-me demonstrar a mim mesmo e a todas as pessoas que me rodeiam, nomeadamente à comunidade escolar e família, os meus conhecimentos adquiridos durante o meu percurso escolar na Epralima.

Na minha opinião, a realização deste projeto foi bastante positiva, tanto a nível pessoal como profissional, pois aprendi bastante, sobretudo, no que diz respeito às dificuldades do mundo do trabalho, pois a eletrónica é uma área bastante complexa que só com determinação e persistência temos lugar num mercado cada vez mais competitivo.

Bibliografia

https://www.arduino.cc

https://www.arduino.cc/en/Reference/LiquidCrystal

https://www.youtube.com

https://playground.arduino.cc/code/keypad/

Anexos

**Anexo 1 – Programação da Identificação ID dos cartões**

**Anexo 2 – Programação sem teclado**

**Anexo 3 – Programação teclado**

**Anexo 4 – Programação final**

**Anexo 1 – Programação da Identificação ID dos cartões**

*#include <SPI.h>*

*#include <MFRC522.h>*

*#define RST\_PIN 8*

*#define SDA\_PIN 9*

*MFRC522 mfrc522(SDA\_PIN, RST\_PIN);*

*void setup() {*

*Serial.begin(9600);*

*SPI.begin();*

*mfrc522.PCD\_Init();*

*Serial.println("Passe cartao para ver seu ID...");*

*}*

*void loop() {*

*if (mfrc522.PICC\_IsNewCardPresent() && mfrc522.PICC\_ReadCardSerial()) {*

*Serial.print("ID tarjeta: ");*

*for (byte i = 0; i < mfrc522.uid.size; i++) {*

*if(mfrc522.uid.uidByte[i] < 0x10) {*

*Serial.print(" 0");*

*}*

*else {*

*Serial.print(" ");*

*}*

*Serial.print(mfrc522.uid.uidByte[i], HEX);*

*}*

*Serial.println();*

*delay(1000);*

*}*

*}*

**Anexo 2 – Programação sem teclado**

*#include <SPI.h>*

*#include <MFRC522.h>*

*#include <LiquidCrystal.h>*

*#define SS\_PIN 9*

*#define RST\_PIN 8*

*#define rele 44*

*#define interruptor 3*

*int estadorele = 0;*

*int estadointerruptor = 0;*

*MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Criar instancia MFRC522*

*LiquidCrystal lcd(22,23,24,25,26,27,28);*

*void setup()*

*{*

*pinMode(rele, OUTPUT);*

*pinMode(interruptor, INPUT);*

*digitalWrite(rele, LOW);*

*Serial.begin(9600); // Inicia a serial comunicaçao*

*lcd.begin(16,2);*

*SPI.begin(); // Inicia SPI bus*

*mfrc522.PCD\_Init(); // Inicia MFRC522*

*Serial.println("Aproxime o cartão...");*

*Serial.println();*

*digitalWrite(rele, LOW);*

*lcd.setCursor(0,0);*

*lcd.print("Aproxime cartao");*

*delay(3000);*

*}*

*void loop()*

*{*

*estadointerruptor = digitalRead(interruptor);*

*if(estadointerruptor == 1){*

*digitalWrite(rele, HIGH);*

*Serial.println("Porta aberta");*

*lcd.print("Porta aberta");*

*delay(3000);*

*digitalWrite(rele, LOW);*

*}*

*else{*

*digitalWrite(rele, LOW);*

*Serial.println("zero");*

*// delay(5000);*

*}*

*lcd.setCursor(0,0);*

*lcd.leftToRight();*

*// novos cartoes*

*if ( ! mfrc522.PICC\_IsNewCardPresent())*

*{*

*return;*

*}*

*// Select one of the cards*

*if ( ! mfrc522.PICC\_ReadCardSerial())*

*{*

*return;*

*}*

*//Mostrar UID no serial monitor*

*Serial.print("UID tag : ");*

*String content= "";*

*byte letter;*

*for (byte i = 0; i < mfrc522.uid.size; i++)*

*{*

*Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");*

*Serial.print(mfrc522.uid.uidByte[i], HEX);*

*content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));*

*content.concat(String(mfrc522.uid.uidByte[i], HEX));*

*}*

*Serial.println();*

*Serial.print("Message : ");*

*content.toUpperCase();*

*if (content.substring(1) == "96 E3 C8 73") //UID Tags de acesso*

*{*

*digitalWrite(rele, HIGH);*

*Serial.println("Acesso Autorizado");*

*Serial.println();*

*lcd.clear();*

*lcd.setCursor(5,0);*

*lcd.print("Cartao");*

*lcd.setCursor(5,1);*

*lcd.print("Valido");*

*delay(1000);*

*lcd.clear();*

*lcd.setCursor(5,0);*

*lcd.print("Acesso");*

*lcd.setCursor(3,1);*

*lcd.print("Autorizado");*

*delay(3000);*

*lcd.clear();*

*setup();*

*digitalWrite(rele, LOW);*

*}*

*else {*

*Serial.println("Acesso Recusado");*

*digitalWrite(rele, LOW);*

*lcd.clear();*

*lcd.setCursor(4,0);*

*lcd.print("Cartao");*

*lcd.setCursor(3,1);*

*lcd.print("Invalido");*

*delay(1000);*

*lcd.clear();*

*lcd.setCursor(4,0);*

*lcd.print("Acesso");*

*lcd.setCursor(4,1);*

*lcd.print("Recusado");*

*delay(3000);*

*lcd.clear();*

*setup();*

*}*

*}*

**Anexo 3 – Programação teclado**

*#include <Password.h> //Libraria Password incluida*

*#include <Keypad.h> //Libraria Keypad incluida*

*#include <LiquidCrystal.h> //Libraria LiquidCrystal incluida*

*Password password = Password("4528");*

*int dlugosc = 4;*

*LiquidCrystal lcd(22,23,24,25,26,27,28); //Definimos los pines del LCD*

*#define rele 44*

*#define led 45*

*int ilosc; //Numero de Cliques*

*const byte ROWS = 4; // Quatro Linhas*

*const byte COLS = 3; // Tres Colunas*

*// Definos a KEYMAP*

*char keys[ROWS][COLS] = {*

*{'1','2','3'},*

*{'4','5','6'},*

*{'7','8','9'},*

*{'\*','0','#'}*

*};*

*byte rowPins[ROWS] = { 34, 35, 36, 37 };// Conectar los keypads ROW1, ROW2, ROW3 y ROW4 a esos Pines de Arduino.*

*byte colPins[COLS] = { 38, 39, 40 };// Conectar los keypads COL1, COL2, COL3 y COL4 a esos Pines de Arduino.*

*Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );*

*void setup()*

*{*

*Serial.begin(9600);*

*keypad.addEventListener(keypadEvent);*

*pinMode(rele, OUTPUT);*

*pinMode(led, OUTPUT);*

*digitalWrite(rele, LOW);*

*lcd.begin(16, 2);*

*lcd.setCursor(0,0);*

*lcd.print(" Bem vindo");*

*lcd.setCursor(0,1);*

*lcd.print("FAVOR Digite PIN");*

*}*

*void loop()*

*{*

*keypad.getKey();*

*}*

*void keypadEvent(KeypadEvent eKey)*

*{*

*switch (keypad.getState())*

*{*

*case PRESSED:*

*int i;*

*for( i = 1; i <= 1; i++ )*

*{*

*digitalWrite(led, HIGH);*

*delay(100);*

*digitalWrite(led, LOW);*

*}*

*Serial.print("Pressed: ");*

*Serial.println(eKey);*

*switch (eKey)*

*{*

*/\**

*case '#':*

*break;*

*case '\*':*

*break;*

*\*/*

*default:*

*ilosc=ilosc+1;*

*password.append(eKey);*

*}*

*//Serial.println(ilosc);*

*if(ilosc == 1)*

*{*

*lcd.clear();*

*lcd.setCursor(1,0);*

*lcd.print(" < PIN >");*

*lcd.setCursor(0,1);*

*lcd.print("\*\_");*

*}*

*if(ilosc == 2)*

*{*

*lcd.clear();*

*lcd.setCursor(1,0);*

*lcd.print(" < PIN >");*

*lcd.setCursor(0,1);*

*lcd.print("\*\*\_");*

*}*

*if(ilosc == 3)*

*{*

*lcd.clear();*

*lcd.setCursor(1,0);*

*lcd.print(" < PIN >");*

*lcd.setCursor(0,1);*

*lcd.print("\*\*\*\_");*

*}*

*if(ilosc == 4)*

*{*

*lcd.clear();*

*lcd.setCursor(1,0);*

*lcd.print(" < PIN >");*

*lcd.setCursor(0,1);*

*lcd.print("\*\*\*\*");*

*}*

*if(ilosc == dlugosc)*

*{*

*delay(50);*

*checkPassword();*

*ilosc = 0;*

*}*

*}*

*}*

*void checkPassword()*

*{*

*if (password.evaluate())*

*{*

*int i;*

*for( i = 1; i <= 3; i++ )*

*{*

*digitalWrite(led, HIGH);*

*delay(100);*

*digitalWrite(led, LOW);*

*}*

*ilosc = 0;*

*password.reset();*

*Serial.println("Correto");*

*lcd.clear();*

*lcd.setCursor(0,1);*

*lcd.print("<<PIN CORRETO>>");*

*digitalWrite(rele, HIGH);*

*delay(3000);*

*digitalWrite(rele, LOW);*

*lcd.clear();*

*lcd.setCursor(0,0);*

*lcd.print(" Bem vindo");*

*lcd.setCursor(0,1);*

*lcd.print("FAVOR ENTRE PIN");*

*}*

*else*

*{*

*int i;*

*for( i = 1; i <= 1; i++ )*

*{*

*digitalWrite(led, HIGH);*

*delay(100);*

*digitalWrite(led, LOW);*

*delay(100);*

*}*

*ilosc = 0;*

*password.reset();*

*Serial.println("Erro");*

*digitalWrite(led, HIGH);*

*delay(400);*

*digitalWrite(led, LOW);*

*digitalWrite(led, HIGH);*

*delay(400);*

*digitalWrite(led, LOW);*

*lcd.clear();*

*lcd.setCursor(0,1);*

*lcd.print("<<PIN INCORRETO>>");*

*delay(2000);*

*lcd.clear();*

*lcd.setCursor(0,0);*

*lcd.print(" Bem vindo");*

*lcd.setCursor(0,1);*

*lcd.print("FAVOR Digite PIN");*

*}*

*}*

**Anexo 4 – Programação final**

*#include <Password.h>*

*#include <SPI.h>*

*#include <MFRC522.h>*

*#include <LiquidCrystal.h>*

*#include <Keypad.h>*

*#define SS\_PIN 9 //SDA Arduino*

*#define RST\_PIN 8 //RST Arduino*

*#define rele 44 //rele ligado ao pino 4 Arduino*

*#define led 45 //led ligado ao pino 45 Arduino*

*#define interruptor 3*

*#define displaylight 5*

*int estadointerruptor = 0; //define o estado do interruptor em 0*

*MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Criar instancia MFRC522*

*LiquidCrystal lcd(22, 23, 24, 25, 26, 27, 28); //Definir os pinos RS,RW,E,D4,D5,D6,D7 do display ao Arduino correspondentes*

*char newPassword[4]; //charater string of newPasswordString*

*byte maxPasswordLength = 4; //valor maximo de carateres PASSWORD*

*byte currentPasswordLength = 0;*

*byte data\_count = 0;*

*char Data[4]; //numero de carateres PASSWORD*

*Password password = Password("4528"); //Password correspondente abrir porta*

*int dlugosc = 4; //Password com 4 carateres*

*int ilosc; //*

*//Definicao da quantidade de linhas e colunas*

*const byte ROWS = 4; //4 Linhas*

*const byte COLS = 3; //3 Colunas*

*//Matriz de carateres*

*char keys[ROWS][COLS] =*

*{*

*{'1', '2', '3'},*

*{'4', '5', '6'},*

*{'7', '8', '9'},*

*{'\*', '0', '#'}*

*};*

*//Definicao dos pinos das linhas*

*byte rowPins[ROWS] = {34, 35, 36, 37}; //Linhas*

*byte colPins[COLS] = {38, 39, 40}; // Colunas*

*byte currentLength = 0;*

*Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );*

*void setup()*

*{*

*Serial.begin(9600); // Inicia a serial comunicaçao*

*lcd.begin(16, 2); //Inicio o display*

*SPI.begin(); // Inicia SPI bus*

*mfrc522.PCD\_Init(); //Inicia o RFID MFRC522*

*keypad.addEventListener(keypadEvent);*

*pinMode(led, OUTPUT); //led como saida*

*pinMode(rele, OUTPUT);*

*pinMode(displaylight, OUTPUT);*

*digitalWrite(displaylight, HIGH);*

*digitalWrite(rele, LOW);*

*digitalWrite(led, LOW);*

*Serial.println("Aproxime o cartão...");*

*Serial.println();*

*digitalWrite(rele, LOW);*

*lcd.setCursor(0, 0);*

*lcd.print("Aproxime cartao");*

*lcd.setCursor(1, 1);*

*lcd.print("Ou Digite PIN");*

*digitalWrite(displaylight, LOW);*

*}*

*void loop()*

*{*

*keypad.getKey();*

*estadointerruptor = digitalRead(interruptor);*

*if(estadointerruptor == 1){*

*lcd.clear();*

*digitalWrite(displaylight, HIGH);*

*digitalWrite(rele, HIGH);*

*Serial.println("Porta aberta");*

*lcd.print("Porta aberta");*

*delay(3000);*

*digitalWrite(rele, LOW);*

*setup();*

*}*

*else{*

*digitalWrite(rele, LOW);*

*}*

*lcd.setCursor(0, 0);*

*lcd.leftToRight();*

*// novos cartoes*

*if ( ! mfrc522.PICC\_IsNewCardPresent())*

*{*

*return;*

*}*

*// Select one of the cards*

*if ( ! mfrc522.PICC\_ReadCardSerial())*

*{*

*return;*

*}*

*//Mostrar UID no serial monitor*

*Serial.print("UID tag : ");*

*String content = "";*

*byte letter;*

*for (byte i = 0; i < mfrc522.uid.size; i++)*

*{*

*Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");*

*Serial.print(mfrc522.uid.uidByte[i], HEX);*

*content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));*

*content.concat(String(mfrc522.uid.uidByte[i], HEX));*

*}*

*Serial.println();*

*Serial.print("Message : ");*

*content.toUpperCase();*

*if (content.substring(1) == "46 D6 C3 93" || content.substring(1) == "46 CD C7 93") //UID Tags de acesso para a fechadura eletrica magnetica*

*{*

*digitalWrite(displaylight, HIGH);*

*digitalWrite(rele, HIGH);*

*digitalWrite(led, HIGH);*

*Serial.println("Acesso Autorizado");*

*Serial.println();*

*lcd.clear();*

*lcd.setCursor(5, 0);*

*lcd.print("Cartao");*

*lcd.setCursor(5, 1);*

*lcd.print("Valido");*

*delay(2000);*

*lcd.clear();*

*lcd.setCursor(5, 0);*

*lcd.print("Acesso");*

*lcd.setCursor(3, 1);*

*lcd.print("Autorizado");*

*delay(3000);*

*lcd.clear();*

*setup();*

*}*

*else {*

*digitalWrite(displaylight, HIGH);*

*Serial.println("Acesso Recusado");*

*digitalWrite(rele, LOW);*

*lcd.clear();*

*lcd.setCursor(4, 0);*

*lcd.print("Cartao");*

*lcd.setCursor(3, 1);*

*lcd.print("Invalido");*

*digitalWrite(led, HIGH);*

*delay(1000);*

*digitalWrite(led, LOW);*

*delay(1000);*

*digitalWrite(led, HIGH);*

*delay(1000);*

*digitalWrite(led, LOW);*

*lcd.clear();*

*lcd.setCursor(4, 0);*

*lcd.print("Acesso");*

*lcd.setCursor(4, 1);*

*lcd.print("Recusado");*

*delay(3000);*

*lcd.clear();*

*}*

*//Para cartao Master*

*if (content.substring(1) == "59 BF 8B AB") //ID Cartao Master*

*{*

*digitalWrite(displaylight, HIGH);*

*Serial.println("Area de configuracao");*

*Serial.println();*

*lcd.setCursor(5, 0);*

*lcd.print("Area de");*

*lcd.setCursor(2, 1);*

*lcd.print("Configuracao");*

*menu();*

*}*

*}*

*void menu()*

*{*

*digitalWrite(displaylight, HIGH);*

*lcd.clear();*

*lcd.setCursor(5, 0);*

*lcd.print("Menu de");*

*lcd.setCursor(0, 1);*

*lcd.print("Programacao");*

*delay(4000);*

*lcd.clear();*

*lcd.setCursor(0, 0);*

*lcd.print("Configuracao");*

*lcd.clear();*

*}*

*void keypadEvent(KeypadEvent eKey)*

*{*

*switch (keypad.getState())*

*{*

*case PRESSED:*

*int i;*

*for ( i = 1; i <= 1; i++ )*

*{*

*digitalWrite(led, HIGH);*

*delay(100);*

*digitalWrite(led, LOW);*

*}*

*Serial.print("Pressed: ");*

*Serial.println(eKey);*

*switch (eKey)*

*{*

*case '#':*

*menu();*

*break;*

*case '\*':*

*setup();*

*break;*

*default:*

*ilosc = ilosc + 1;*

*password.append(eKey);*

*}*

*//Serial.println(ilosc);*

*if (ilosc == 1)*

*{*

*digitalWrite(displaylight, HIGH);*

*lcd.clear();*

*lcd.setCursor(1, 0);*

*lcd.print("< PIN >");*

*lcd.setCursor(0, 1);*

*lcd.print("\*\_");*

*}*

*if (ilosc == 2)*

*{*

*lcd.noBlink();*

*lcd.clear();*

*lcd.setCursor(1, 0);*

*lcd.print("< PIN >");*

*lcd.setCursor(0, 1);*

*lcd.print("\*\*\_");*

*}*

*if (ilosc == 3)*

*{*

*lcd.noBlink();*

*lcd.clear();*

*lcd.setCursor(1, 0);*

*lcd.print("< PIN >");*

*lcd.setCursor(0, 1);*

*lcd.print("\*\*\*\_");*

*}*

*if (ilosc == 4)*

*{*

*lcd.noBlink();*

*lcd.clear();*

*lcd.setCursor(1, 0);*

*lcd.print("< PIN >");*

*lcd.setCursor(0, 1);*

*lcd.print("\*\*\*\*");*

*}*

*if (ilosc == dlugosc)*

*{*

*delay(50);*

*checkPassword();*

*ilosc = 0;*

*}*

*}*

*}*

*void checkPassword()*

*{*

*if (password.evaluate())*

*{*

*int i;*

*for ( i = 1; i <= 3; i++ )*

*{*

*digitalWrite(led, HIGH);*

*delay(100);*

*digitalWrite(led, LOW);*

*}*

*ilosc = 0;*

*password.reset();*

*Serial.println("Correto");*

*digitalWrite(displaylight, HIGH);*

*lcd.clear();*

*lcd.setCursor(0, 1);*

*lcd.print("<<PIN CORRETO>>");*

*digitalWrite(rele, HIGH);*

*delay(3000);*

*digitalWrite(rele, LOW);*

*lcd.clear();*

*lcd.setCursor(0, 0);*

*lcd.print(" Bem vindo");*

*lcd.setCursor(0, 1);*

*lcd.print("FAVOR Digite PIN");*

*setup();*

*}*

*else*

*{*

*int i;*

*for ( i = 1; i <= 1; i++ )*

*{*

*digitalWrite(led, HIGH);*

*delay(100);*

*digitalWrite(led, LOW);*

*delay(100);*

*}*

*ilosc = 0;*

*password.reset();*

*Serial.println("Error");*

*digitalWrite(led, HIGH);*

*delay(400);*

*digitalWrite(led, LOW);*

*digitalWrite(led, HIGH);*

*delay(400);*

*digitalWrite(led, LOW);*

*digitalWrite(displaylight, HIGH);*

*lcd.clear();*

*lcd.setCursor(2, 0);*

*lcd.print("PIN INCORRETO");*

*delay(2000);*

*lcd.clear();*

*setup();*

*}*

*}*

|  |  |
| --- | --- |
| **Assinatura |** | |
| **Aluno |** |  |