#include <Wire.h>

#include <LiquidCrystal\_I2C.h> // declara a utilização da biblioteca LiquidCrystal

#include <SoftwareSerial.h>

#define MAX 4 //define o numero máximo de IMEI's e senhas armazenadas

#define PENDENTE 0 //define a autorização de acesso como pendente (caso o tamanho da senha ou do IMEI seja incorreto)

#define SIM 1 //permite o acesso

#define NAO 2 //nega o acesso

#define ALARME 3 //SITUAÇÃO DE EMERGÊNCIA, SOA UM ALARME

SoftwareSerial bluetooth(10,11); //instancia o bluetooth nos pinos 10 e 11 do arduino

String list\_IMEI[MAX] = {"356111061414319","356111061414327","2222222222","3333333333"};//lista de IMEI's com permissão de acesso

String list\_senha[MAX] = {"1234","4321","5555","9999"}; //lista de senhas para os IMEI's

String list\_senha\_emergencia[MAX] = {"6666","7777","6969","1313"};

**int** permissao = PENDENTE;

// Inicializa o display no endereco 0x27 srial do Arduino

LiquidCrystal\_I2C lcd(0x27,2,1,0,4,5,6,7,3, POSITIVE);

//faz o LED piscar 7x

**void** piscar7(){

**for**(**int** i = 0;i < 7;i++){ //exececuta 7 vezes

digitalWrite(7, HIGH); // acende o led no pino 7

delay(200); // espara 200 milisegundos

digitalWrite(7, LOW); // apaga o led

delay(200); // espara 200 milisegundos

}

}

//piscar 1x por 2 segundos

**void** piscar1(){

digitalWrite(7, HIGH); // acende o led no pino 7

delay(2000); // espera 2 segundos

digitalWrite(7, LOW); // apaga o led

}

//beep de acesso permitido

**void** beep1(){

tone(8,1000,200); //no pino 8, 1000Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,2000,200); //no pino 8, 2000Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,3000,200); //no pino 8, 3000Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,4000,200); //no pino 8, 4000Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,5000,200); //no pino 8, 5000Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

}

//beep de acesso negado

**void** beep2(){

tone(8,500,200); //no pino 8, 500Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,500,200); //no pino 8, 500Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,500,200); //no pino 8, 500Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,500,200); //no pino 8, 500Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,500,200); //no pino 8, 500Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

}

//sirene de emergência

**void** siren(**int** number){

**for**(**int** i = 0;i < number; i++){

tone(8,500,200); //no pino 8, 500Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

tone(8,1000,200); //no pino 8, 500Hz durante 200 milisegundos

delay(200); //aguarda 200 milisegundos

}

}

// verificar acesso

**int** verif\_permissao(String arg0){ //recebe como parâmetro o IMEI+senha recebidos por Bluetooth e retorna um inteiro referente a permissão

**for**(**int** i = 0; i < MAX; i++){ //percorre todos os IMEI's e senha cadastrados para verificar acesso

**if**(arg0.length() < 10){ //se o IMEI+senha não tiverem o tamanho correto

arg0 = "";

**return** PENDENTE; //o acesso é pendente (o hardware não executa nada)

}

**else** **if**(arg0 == (list\_IMEI[i] + list\_senha[i])){ //Se o argumento recebido for igual a combinação de IMEI+senha cadastrado

**return** SIM; //se encontrar o IMEI e a senha correspondente libera o acesso

}

**else** **if**(arg0 == (list\_IMEI[i] + list\_senha\_emergencia[i])){ //Se o argumento recebido for igual a combinação de IMEI+senha cadastrado

**return** ALARME; //se encontrar o IMEI e a senha de emergencia for digitada trava o acesso e soa um alarme

}

**else**{

arg0 = "";

**return** NAO; //se não encontrar nega o acesso

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EXECUÇÃO DO PROGRAMA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** run(){

lcd.setCursor(0,0); //seta o cursor na 1ª coluna da 1ª linha do LCD

lcd.print("Aguardando... "); //escreve no LCD

//se a porta serial Bluetooth estive disponível

**if**(bluetooth.available()){

//verifica a permissão através do IMEI+senha recebido

permissao = verif\_permissao(bluetooth.readString());

}

//verifica a permissão e executa ações a partir desta permissão

**switch**(permissao){

//autorizar acesso

**case** SIM:

Serial.println("Acesso liberado"); //escreve na porta serial

Serial.println(" Bem Vindo"); //escreve na porta seria

bluetooth.write("Acesso liberado \n"); //envia a string via bluetooth

bluetooth.write(" Bem Vindo\n"); //envia a string via bluetooth

bluetooth.write("#autorizado~\n");

lcd.setCursor(0,0); //seta o cursor na 1ª coluna da 1ª linha do LCD

lcd.print("Acesso liberado "); //escreve no LCD

lcd.setCursor(0,1); //seta o cursor na 1ª coluna da 2ª linha do LCD

lcd.print(" Bem Vindo"); //escreve no LCD

//toca beep

beep1(); //toca o beep

piscar1(); //pisca o LED

lcd.clear(); //limpa o LCD

permissao = PENDENTE; //a permissão torna-se pendente para a próxima iteração

**break**;

//negar acesso

**case** NAO:

Serial.println("Acesso negado "); //escreve na serial

bluetooth.write(" Acesso negado \n"); //envia via bluetooth

bluetooth.write("Tente novamente\n"); //envia via bluetooth

bluetooth.write("#negado~\n");

lcd.setCursor(0,0); //seta o cursor na 1ª coluna da 1ª linha do LCD

lcd.print(" Acesso negado"); //escreve no LCD

lcd.setCursor(0,1); //seta o cursor na 1ª coluna da 2ª linha do LCD

lcd.print("Tente novamente"); //escreve no LCD

//faz 5 beeps

beep2();

//pisca 7 vezes

piscar7();

//aguarda 3 segundos

delay(5000);

lcd.clear(); //limpa o LCD

permissao = PENDENTE; //a permissão torna-se pendente para a próxima iteração

**break**;

//caso a permissão seja pendente não executa nada

**case** PENDENTE:

**break**;

//caso a senha de emergência for digitada

**case** ALARME:

siren(10); //soa uma sirene de emergencia

**break**;

**default**:

**break**;

}//fim switch

}//fim main

**void** setup()

{

bluetooth.begin(9600); //inicializa o bluetooth

pinMode(7, OUTPUT); //inicializa o LED no pino 7 do Arduino

Serial.begin(9600); //inicializa a porta serial 9600 bits/seg.

lcd.begin (16,2); //inicializao LCD de 2 linhas x 16 colunas

}

//executa o programa em um loop infinito

**void** loop()

{

run();

}