

Combination Lock

DOKUMENTACJA

JĘZYK POLSKI

AUTORZY: KRZYSZTOF POKORA & NORBERT LIGAS

AKADEMIA GÓRNICZO- HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ INFORMATYKI,
ELEKTRONIKI I TELEKOMUNIKACJI
KIERUNEK ELEKTRONIKA

SPIS TREŚCI

Spis treści	1
Wstęp	2
Opis	2
Funkcjonalność	2
Analiza problemu	3
Projekt techniczny	4
Schemat klawiatury	4
Schemat układu	5
Opis realizacji	6
Opis wykonanych testów	7
Podręcznik użytkownika	8
Opis urządzenia	8
Otwieranie zamka	8
Zmiana hasła	8
Metodologia rozwoju i utrzymania oprogramowania	9
Rozwój oprogramowania	9
Kod oprogramowania	10

WSTĘP

OPIS

Zamek szyfrowy do drzwi, bram lub furtek otwierany za pomocą pilota bądź klawiatury z dostępem hasłem deweloperskim lub ustawionym użytkownika. Po naciśnięciu dowolnego klawisza ekran aktywuje się po czym należy wprowadzić hasło. Otwarcie zamka imitowane świecącą diodą. Projekt został złożony na platformie Arduino Mega z oprogramowaniem napisanym w programie Arduino w języku Arduino C.

FUNKcjONALNOŚĆ

Do funkcjonalności urządzenia można zaliczyć:

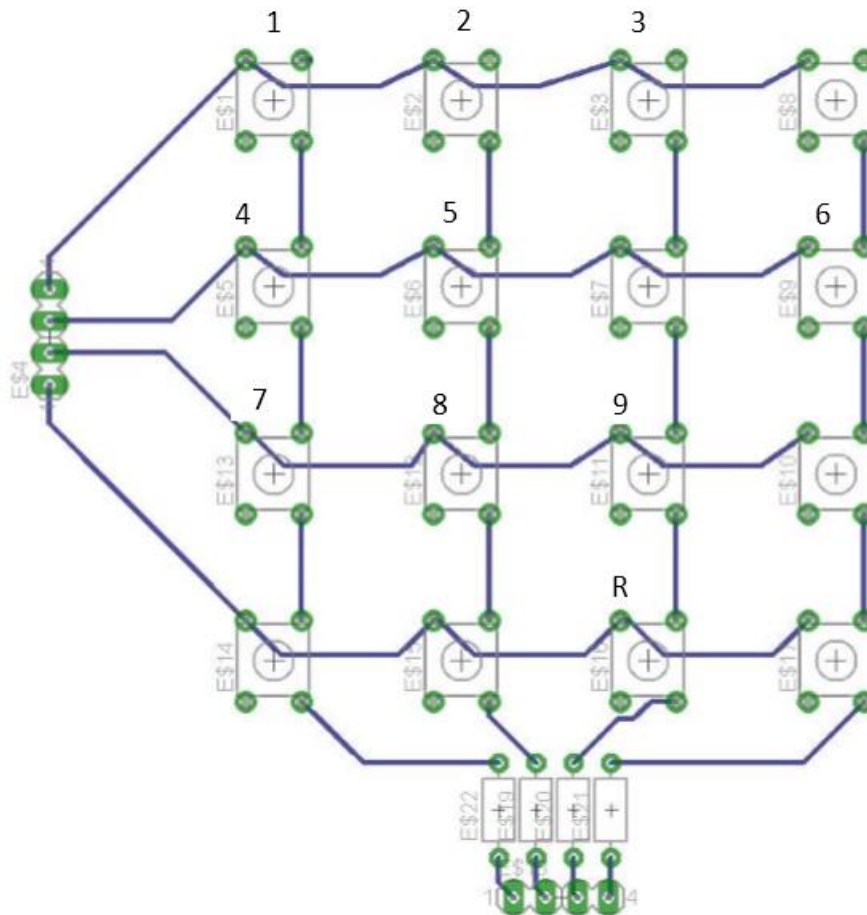
- Wygaszanie ekranu
- Jednopoziomowe zabezpieczenie hasłem
- Dostęp hasłem deweloperskim w przypadku awarii, bądź zapomnienia hasła użytkownika
- Możliwość zmiany hasła
- Dostęp z klawiatury
- Dostęp z pilota z klawiaturą

ANALIZA PROBLEMU

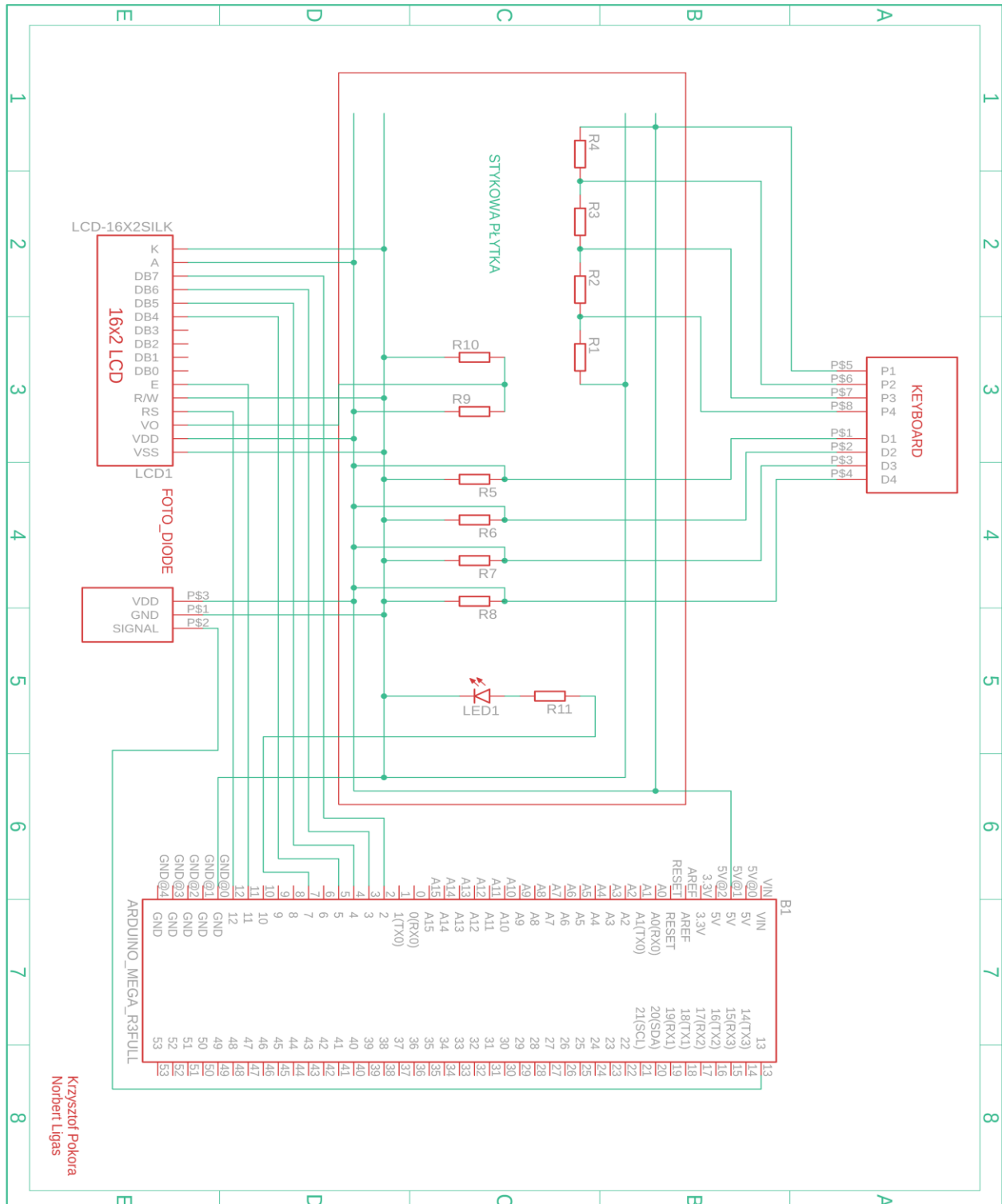
- Wygaszanie ekranu
- Hasło użytkownika
 - Zapisane startowo łatwe hasło użytkownika
 - Kontrola poprawności wpisanego hasła
- Zapisane w oprogramowaniu hasło deweloperskie
 - Zabezpieczenie przed możliwością zmiany
 - Dostęp powyższym hasłem do zmiany hasła użytkownika
- Możliwość zmiany hasła
 - Hasło startowe z możliwością zmiany na nowe
 - Zabezpieczenie starym hasłem lub deweloperskim
- Dostęp z klawiatury
 - Zczytywanie napięć z wyjścia klawiatury (każdy wiersz osobne wyjście)
 - Wysterowanie napięć rezystorami (każda kolumna inne napięcie)
 - Odczyt napięć w oprogramowaniu
 - Analiza napięć w celu identyfikacji wciśniętego przycisku
 - Wyświetlanie wprowadzanego hasła
- Dostęp z pilota z klawiaturą
 - Podłączenie do układu odbiornika
 - Zczytywanie odbieranego kodu
 - Nadanie odpowiedniego opóźnienia w wykonywaniu programu w celu uniknięcia przeczytania kilkakrotnie raz klikniętego przycisku
 - Analiza odebranego kodu w celu identyfikacji klikniętego przycisku
 - Wyświetlanie wprowadzanego hasła

PROJEKT TECHNICZNY

SCHEMAT Klawiatury



SCHEMAT UKŁADU



OPIS REALIZACJI

Projekt został złożony na platformie Arduino Mega z oprogramowaniem napisanym w programie Arduino w języku Arduino C. Głównymi problemami i ich rozwiązaniami są:

- **Wygaszanie ekranu:** Ekran po podłączeniu zasilania jest wygaszony (nie wyświetla żadnego tekstu). Po kliknięciu dowolnego przycisku dopiero wyświetlany jest odpowiedni tekst. System podczas wygaszenia ekranu jest gotowy na odebranie sygnału z pilota bądź klawiatury, lecz nie analizuje go, gdyż na tym etapie byłoby to bezcelowe.
- **Hasło użytkownika:** Po podłączeniu zasilania hasło użytkownika to 0000. Przy każdym odłączeniu systemu od zasilania hasła resetują się. Hasło jest z możliwością zmiany (zawsze 4 cyfrowe).
- **Zapisane w oprogramowaniu hasło deweloperskie:** Hasło deweloperskie bez możliwości jego zmiany dostępne jest zawsze takie samo. Nie ma możliwości jego zmiany. Za jego pomocą można otworzyć zamek lub zmienić hasło użytkownika.
- **Możliwość zmiany hasła:** Po wybudzeniu urządzenia możliwe jest zmienianie hasła użytkownika przez kliknięcie „#” na klawiaturze urządzenia lub „EQ” na pilocie. Żeby wprowadzić nowe hasło należy najpierw wpisać poprzednie hasło użytkownika, bądź dewelopera. Po wprowadzeniu nowego hasła użytkownika można je od razu wpisać i tym samym otworzyć zamek.
- **Dostęp z klawiatury:** Klawiatura nie jest bezpośrednio połączona z platformą. Przyciski są połączone z rezystorami, które podłączone tworzą dzielnik napięcia, co sprawia że każda kolumna przycisków jest reprezentowana innym napięciem. Każdy wiersz przycisków ma osobne wyjście, dzięki temu możliwa jest identyfikacja przycisku w oprogramowaniu. Odpowiednia funkcja interpretuje z odpowiednich zakresów napięć, który przycisk został kliknięty, a następnie wprowadza go jako następną wprowadzoną cyfrę hasła i wyświetla. Zostało także wprowadzone specjalne opóźnienie w programie, żeby ominąć błędy powodowane drganiem zestyków itp.
- **Dostęp z pilota z klawiaturą:** Po platformy podłączone jest wyjście odbiornika, którym jest IR RECEIVER CHQ 1838, który odbiera wysyłany z pilota kod. Odpowiednia funkcja interpretuje, który przycisk został kliknięty (niezależnie czy z pilota czy klawiatury), a następnie wprowadza go jako następną wprowadzoną cyfrę hasła i wyświetla. Zostało także wprowadzone specjalne opóźnienie w programie, żeby ominąć błędy zbyt szybkiego kilkukrotnego odczytu jednorazowo wysłanego kodu.

OPIS WYKONANYCH TESTÓW

Kod	Data	Autor	Opis	Stan
0.001	05.06.2019	Norbert Ligas	Reaguje na pilot wyłącznie przy wybudzaniu wyświetlacza.	Naprawiony
0.002	06.06.2019	Krzysztof Pokora	Wyświetla wcześniej wprowadzone hasło i nadpisuje kolejne cyfry.	Naprawiony
0.003	06.06.2019	Krzysztof Pokora	Wprowadza raz klikniętą cyfrę kilkakrotnie.	Naprawiony
0.004	07.06.2019	Krzysztof Pokora	Wyświetlacz zamienił się w licznik, który się inkrementuje o wartość klikniętej liczby	Naprawiony
0.005	07.06.2019	Norbert Ligas	Słabo świecąca dioda.	Naprawiony
0.006	08.06.2019	Norbert Ligas	Wyświetlanie zera po uruchomieniu wyświetlacza w miejscu wpisywanego hasła. Przy wprowadzaniu jednak znika pod pierwszą wprowadzoną cyfrą.	Zdarza się rzadko. Nierozwiązany.

PODRĘCZNIK UŻYTKOWANIA

OPIS URZĄDZENIA

Urządzenie składa się z 5 elementów głównych tj. wyświetlacza, odbiornika z pilotem, klawiatury z dzielnikiem napięcia, diody imitującej zamek, platformy arduino. Urządzenie należy podłączyć do zasilania przez kabel USB lub kabel zasilający.

OTWIERANIE ZAMKA

1. Wybudź urządzenie kliknięciem dowolnego klawisza na klawiaturze lub pilocie.
2. Wprowadź hasło użytkownika (serwis ma dostęp również za pomocą hasła deweloperskiego).
3. Dioda imitująca otwarcie zamka powinna się zaświecić jeżeli hasło zostało wpisane poprawnie. Jeżeli nie należy hasło wprowadzić jeszcze raz.

ZMIANA HASŁA

1. Wybudź urządzenie kliknięciem dowolnego klawisza na klawiaturze lub pilocie.
2. Kliknij „EQ” na pilocie, bądź „#” na klawiaturze.
3. Wprowadź stare hasło, lub poproś serwis o wpisanie hasła deweloperskiego.
4. Wprowadź nowe hasło użytkownika.
5. Otwórz zamek nowym hasłem użytkownika.
6. Dioda imitująca otwarcie zamka powinna się zaświecić jeżeli hasło zostało wpisane poprawnie. Jeżeli nie należy hasło wprowadzić jeszcze raz.

METODOLOGIA ROZWOJU I UTRZYMANIA OPROGRAMOWANIA

ROZWÓJ OPROGRAMOWANIA

- Podłączenie odpowiednio sprzętu
- Napisanie funkcji interpretującej napięcia klawiatury
- Napisanie funkcji interpretującej z napięć kliknięty przycisk
- Wywołanie opóźnienia po odczytaniu wartości przycisku
- Napisanie funkcji wyświetlającej wpisywane hasło i dodającej kolejną cyfrę do zmiennej przechowującej wpisywane hasło
- Dopisanie obsługi diody i odpowiednich dla niej reakcji względem użytkownika w danej sytuacji.
- Dopisanie funkcji odpowiedzialnych za zmianę hasła
- Dopisanie obsługi pilota na poziomie wybudzania wyświetlacza
- Dopisanie obsługi pilota z możliwością wprowadzania z niego cyfr hasła
- Usprawnienie działania oprogramowania
- Poukładanie odpowiednio połączeń, urządzeń i okablowania

KOD OPROGRAMOWANIA

```
#include <SPI.h>
#include <LiquidCrystal.h>
#include <IRremote.h>

#define receiverpin 13

IRrecv irrecv(receiverpin);
decode_results results;

LiquidCrystal lcd(12,11,5,4,3,2);
int portA1 = A1, portA2 = A2, portA3 = A3, portA4 = A4;
int V1, V2, V3, V4;
int keyCode, counter = 0, password = 0000;
const int developerPassword = 9999;
int writtenPassword, oldPassword, newPassword;
int remoteValue = 0;
boolean activity = false, changingState = false;
unsigned long time_;
#define waitTime 15000

void setup()
{
  Serial.begin(9600);
  SPI.begin();
  lcd.begin(16,2);
  pinMode(7,OUTPUT);
  irrecv.enableIRIn();
}

void loop()
{
  VoltageMeasurement();
  if (irrecv.decode(&results) || (V1 > 100 || V2 > 100 || V3 > 100 || V4 > 100))
  {
    lcd.display();
    activity = true;
    time_ = millis() + waitTime;
  }

  if (activity && time_ < millis())
    activity = false;

  if(activity==true)
  {
    lcd.clear();
    Registration();
  }
  else
  {
    lcd.noDisplay();
  }
  irrecv.resume();
}
```

```
void check()
{
    lcd.clear();
    lcd.setCursor(4,0);
    lcd.print("CHECKING");
    delay(2000);
    lcd.clear();
    if(writtenPassword == password)
    {
        lcd.setCursor(4,0);
        lcd.print("WELCOME!");
        digitalWrite(7,HIGH);
        delay(6000);
        digitalWrite(7,LOW);
        lcd.clear();
    }
    else if (writtenPassword == developerPassword)
    {
        lcd.setCursor(2,0);
        lcd.print("DEVELOPER'S");
        lcd.setCursor(8,1);
        lcd.print("ACCESS");
        digitalWrite(7,HIGH);
        delay(6000);
        digitalWrite(7,LOW);
        lcd.clear();
    }
    else
    {
        lcd.setCursor(2,0);
        lcd.print("INCORRECT");
        lcd.setCursor(6,1);
        lcd.print("PASSWORD");
        delay(3000);
    }
    activity=false;
    counter=0;
}

void Registration()
{
    irrecv.resume();
    do
    {
        lcd.setCursor(1,0);
        lcd.print("PASSWORD:");
        lcd.setCursor(counter+11,1);
        delay(300);
        if(irrecv.decode(&results))
        {
            Serial.print(results.value);
            remoteValue = results.value;
        }
        else
        {
            VoltageMeasurement();
        }
        KeyIdentification();
        Display(keyCode);
    }
```

```
delay(200);
if(keyCode=="#")
{
    keyCode = "noCode";
    changingState = true;
    counter=0;
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("PASSWORD");
    lcd.setCursor(6,1);
    lcd.print("CHANGING");
    delay(3000);
    lcd.clear();
    do
    {
        lcd.setCursor(1,0);
        lcd.print("LAST PASSWORD:");
        lcd.setCursor(counter+11,1);
        delay(300);
        if(irrecv.decode(&results))
        {
            Serial.print(results.value);
            remoteValue = results.value;
        }
        else
        {
            VoltageMeasurement();
        }
        KeyIdentification();
        Display2(keyCode);
        keyCode = "noCode";
        delay(200);
        irrecv.resume();
    } while(counter<4);

    ResetPassword();
}

keyCode = "noCode";
irrecv.resume();

}while(counter<4);
lcd.clear();

if(changingState == false)
{
    check();
}
}

void VoltageMeasurement()
{
    delay(200);
    V1 = analogRead(portA1);
    V2 = analogRead(portA2);
    V3 = analogRead(portA3);
    V4 = analogRead(portA4);
}
```

```
void KeyIdentyfication()
{
    if((V1<100 && V2<100 && V3<100 && V4<100) && (remoteValue > 40000000 || remoteValue == 0))
    {
        keyCode = "noCode";
    }
    else if (V4>100 && V4<300)
    {
        keyCode = "*";
        counter++;
    }
    else if ((V4>300 && V4<550) || remoteValue == 26775 )
    {
        keyCode = 0;
        counter++;
    }
    else if ((V4>550 && V4<800) || remoteValue == -28561 )
    {
        keyCode = "#";
    }
    else if ((V3>100 && V3<300) || remoteValue == 17085 )
    {
        keyCode = 7;
        counter++;
    }
    else if ((V3>300 && V3<550) || remoteValue == 19125 )
    {
        keyCode = 8;
        counter++;
    }
    else if ((V3>550 && V3<800) || remoteValue == 21165 )
    {
        keyCode = 9;
        counter++;
    }
    else if ((V2>100 && V2<300) || remoteValue == 4335 )
    {
        keyCode = 4;
        counter++;
    }
    else if ((V2>300 && V2<550) || remoteValue == 14535 )
    {
        keyCode = 5;
        counter++;
    }
    else if (V2>960 || remoteValue == 23205 )
    {
        keyCode = 6;
        counter++;
    }
    else if ((V1>100 && V1<300) || remoteValue == 12495 )
    {
        keyCode = 1;
        counter++;
    }
    else if ((V1>300 && V1<550) || remoteValue == 6375 )
    {
        keyCode = 2;
        counter++;
    }
}
```

```
else if ((V1>550 && V1<800) || remoteValue == 31365 )
{
    keyCode = 3;
    counter++;
}
remoteValue = 0;
}

void Display(int character)
{
    if (character != "noCode" && (character==1 || character==2 || character==3 || character==4 || character==5 || character==6 ||
character==7 || character==8 || character==9 || character==0)){
        lcd.print(character);
        if(counter==1)
            writtenPassword=1000*character;
        if(counter==2)
            writtenPassword=writtenPassword+100*character;
        if(counter==3)
            writtenPassword=writtenPassword+10*character;
        if(counter==4){
            writtenPassword=writtenPassword+1*character;
            delay(800);}
        }
    }

void Display2(int character)
{
    if (character != "noCode" && (character==1 || character==2 || character==3 || character==4 || character==5 || character==6 ||
character==7 || character==8 || character==9 || character==0)){
        lcd.print(character);
        if(counter==1)
            oldPassword=1000*character;
        if(counter==2)
            oldPassword=oldPassword+100*character;
        if(counter==3)
            oldPassword=oldPassword+10*character;
        if(counter==4){
            oldPassword=oldPassword+1*character;
            delay(800);}
        }
    }

void Display3(int character)
{
    if (character != "noCode" && (character==1 || character==2 || character==3 || character==4 || character==5 || character==6 ||
character==7 || character==8 || character==9 || character==0))
    {
        lcd.print(character);
        if(counter==1)
            newPassword=1000*character;
        if(counter==2)
            newPassword=newPassword+100*character;
        if(counter==3)
            newPassword=newPassword+10*character;
        if(counter==4){
            newPassword=newPassword+1*character;
            delay(800);}
        }
    }
```



```
void ResetPassword()
{
  if( oldPassword == password || oldPassword == developerPassword )
  {
    counter=0;
    lcd.clear();
    lcd.setCursor(4,0);
    lcd.print("CHANGING");
    delay(2000);
    do{

      irrecv.resume();
      lcd.setCursor(1,0);
      lcd.print("NEW PASSWORD:");
      lcd.setCursor(counter+11,1);
      delay(300);
      if(irrecv.decode(&results))
      {
        remoteValue = results.value;
      }
      else
      {
        VoltageMeasurement();
      }
      KeyIdentification();
      Display3(keyCode);
      keyCode = "noCode";
      delay(200);

    } while(counter<4);
    SetNewPassword();
  }
  else
  {
    activity=false;
    counter=0;
    changingState = false;
    lcd.clear();
  }
}

void SetNewPassword()
{
  password=newPassword;

  changingState = false;
  activity=false;
  counter=0;
  lcd.clear();
}
```