

Střední průmyslová škola elektrotechnická Havířov	Zpráva z MIT	Třída: 4.C
		Skupina:
Protokol z úlohy Kódový zámek		Zpráva číslo:
		Den:
		Učitel: Petr Madecki
		Jméno: Adam Gawlas
		Známka:

Zadání:

Kodový zamek:

Obecné info:

- mrizka # znamená Enter (potvrzení)
- místo vypisu hesla zobrazovat zastupný znak *
- všechna hesla jsou 4 místná
- všechny vypisy a přechody mezi nimi volit smysluplně a vstřícně, stejně tak jako dobu, po kterou budou zobrazeny, ať je možno je v klidu přecíst a reagovat na ně

Test LCD

- vypis textu na 2 řádky

Hlavní program:

a) Overení hesla:

- max. 3 pokusy na zadání hesla
- v případě úspěchu vypsat "OK", jinak vypsat info o chybě spolu s počtem zbývajících pokusů
- po 3 neúspěšných pokusech zablokovat přívarek (vypsat info)
- hesla nebudou overována automaticky, ale až po stisku #

b) Odblokování přípravku:

- Po zablokování mít možnost odblokovat pomocí servisního kodu (opět potvrdit pomocí #)

c) Menu:

- Po úspěšném přihlášení vypsat uživateli pozdrav a nabídnout možnost odhlášení (vyber jednoho z písmenek na klávesnici)

Rozšíření:

- Obohatit menu po přihlášení o možnost změny hesla
 - Změna probíhá stylem, že je nutno nejprve zadat 1x původní heslo a poté 2x nové (vždy potvrdit pomocí #)
 - V případě úspěchu bude další přihlášení již s novým heslem
 - V případě neúspěchu vypsat chybu a vrátit se do menu po přihlášení
- Pro odblokování přípravku použít RFID
 - a) Po přiložení čipu se nic nestane, ale po přiložení karty dojde k automatickému odblokování
 - b) Po přiložení čipu nebo karty dojde k zobrazení obrazovky pro zadání servisního kodu. Po jeho zadání se přívarek

Teoretický rozbor:

Použitý přípravek: Arduino Uno

Mozkem této desky je 8bitový mikrokontrolér ATmega328P o flash paměti 32 kB, paměti SRAM 2 kB a paměti EEPROM 1 kB. Arduino UNO obsahuje 14 digitálních pinů všechny mohou být použity jako vstup či výstup užitím funkcí `pinMode()`, `digitalWrite()` a `digitalRead()`.

Maximální dovolený proud těmito svorkami je 20 mA, pokud proud dosáhne dvojnásobku tohoto proudu, může dojít k poškození vývojového kitu. Dále obsahuje 6 analogových vstupů o šířce 10 bitů, ty disponují měřicím rozsahem 5 V. Toto napětí lze upravit užitím funkce `analogReference()` na pin AREF.

Přesný krystal generuje pro celý obvod hodinový signál 16 MHz.

Arduino UNO disponuje USB konektorem pro komunikaci s PC, piny umožňující pulzně šířkovou modulaci, napájecí konektor, resetovací tlačítko atp.

Teplotní čidlo DS18B20

Čidlo DS18B20 je digitální teploměr s kalibrovanou přesností $\pm 0,5$ °C a rozlišením až 12 bitů.

Informace o teplotě se z čidla do Arduina přenáší digitálně, nedochází tak ke zkreslení naměřené hodnoty ani změně v důsledku rušení. DS18B20 využívá jednoduché komunikační rozhraní 1-Wire, které pro přenos dat potřebuje pouze jeden vodič. Teplotní rozsah je $-55 \div +125$ °C a rozlišovací schopnost je až $\pm 0,0625$ °C. Teplotní čidlo zvládne také „alarm“ funkci s volitelnou horní a spodní úrovní teploty. Teplotní čidlo DS18B20 má pro adresaci jedinečný 64-bitový kód. To spolu s vlastnostmi sběrnice 1-Wire umožňuje použít v Arduino projektu až 50 teplotních čidel pospojovaných za sebou (paralelně) a komunikovat s nimi.

Displej I2C (16x2)

Displej je vybaven převodníkem na sběrnici I2C, který je zapojen pouhými čtyřmi piny.

Dvěma napájecími a dvěma datovými. Tím je zredukován počet použitých výstupů z kitu, aby jich zbylo dostatečné množství pro další periferie. Displej je napájen napětím 5 V.

Obsahuje 3 paměti:

- DDRAM - Slouží pro zápis dat pro zobrazení na displeji
- CGRAM - Slouží pro deklaraci vlastních znaků (po každém resetování je nutno zadeklarovat znovu)
- CGROM - zde jsou zapsány defaultní znaky

RC522 RFID čtečka

Modul RFID čtečky 13,56MHz pro mikrokontroléry s čipem a kartou. Komunikace probíhá na sběrnici SPI.

Řídící čip je PHILIPS MFRC522

Detekce chyb v přenosu ISO14443A. Podpora šifrování CRYPTO1.

Obousměrná rychlost přenosu dat až 424 kbit/s. SPI komunikační sběrnice.

Integrovaná anténa

Provozní proud: 13 - 26 mA / 3,3V. Klidový proud: 10 - 13 mA / 3,3V

Provozní teplota: -20 až 80 °C

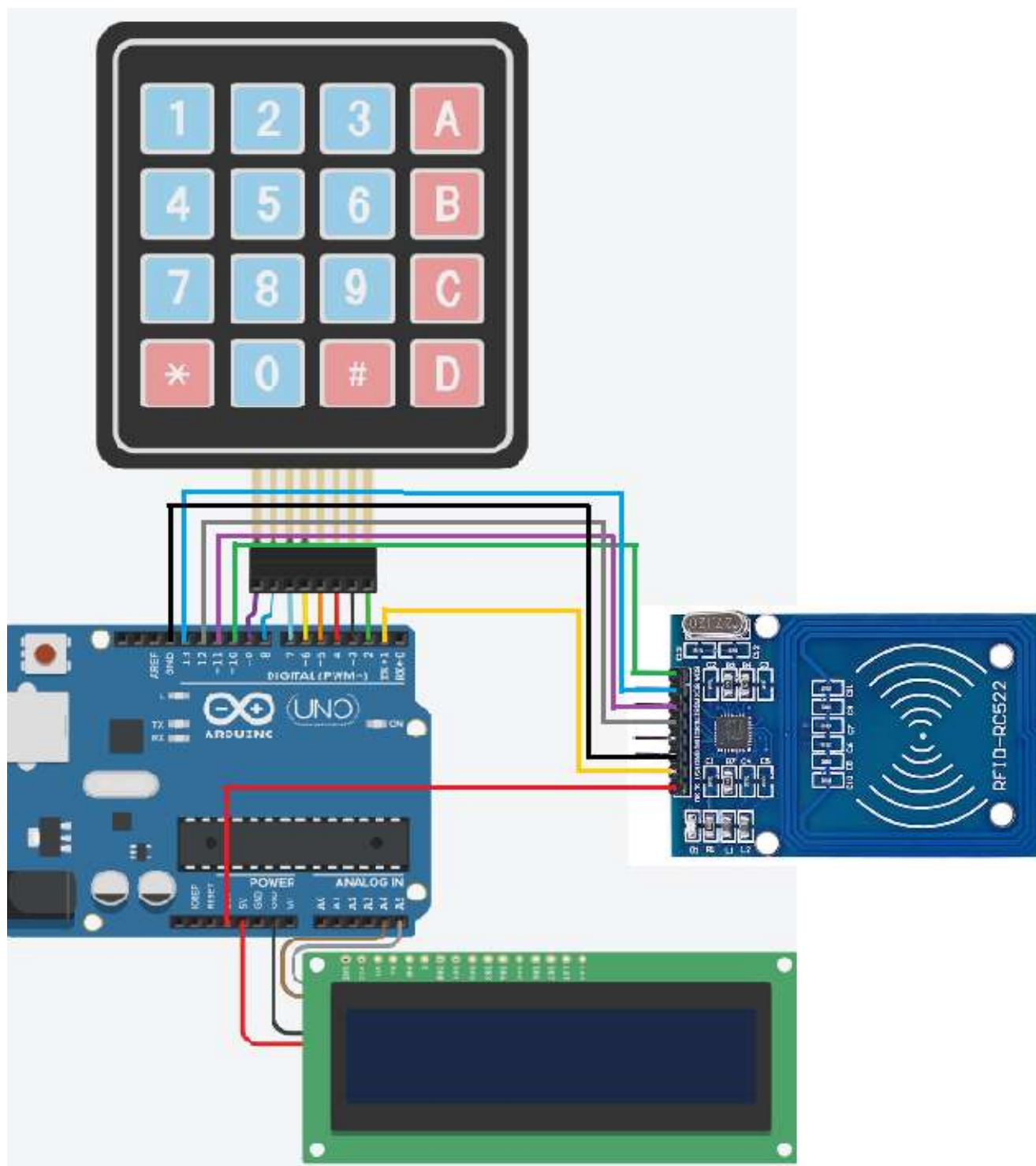
Napájení: 3.3 VDC

Klávesnice 4x4 16 tlačítek

Klávesnice se skládá z 16 tlačítek. K arduinu se připojuje pomocí 8 pinů - 4 řádky a 4 sloupce. Používá knihovnu Keypad.h

Schéma zapojení

(Ve skutečném zapojení mám použitý displej se sběrnici I2C)



Zdrojový kód:

```
#include <Keypad.h> //knihovna pro klavensici
#include <LiquidCrystal_I2C.h> //knihovan pro I2C
#include <Wire.h> //knihovna pro I2C
#include <SPI.h> // připojení knihoven SPI a MFRC522. Knihovna pro RFID
#include <MFRC522.h>

#define SDA_PIN 10 // definování pinů pro SDA a RST
#define RST_PIN 1

int i, y; //Pomocné proměnné
int pokusy[1]; //pocet pokusů
int volba[1]; //Volba z menu
int x[1]; //pomocná proměnná ve funkci srovnavani_ser_kod

char ser_kod[4] = {'1', '5', '5', '1'}; //Zde je ulozeny servisni kod
char novy_kod[4]; //Zde se bude ukládat nový kód
char heslo[4] = {'1', '2', '3', '4'}; //Zde je uložený přístupový kód
char zadano[4]; //Zde se budou ukládat znaky zmacknute na klavesnici
char keys[4][4] = { //piny klavesnice 2-9, displej scl-A5
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

byte pinyRadku[4] = {5, 4, 3, 2}; //čísla pinů s řádkem 1 2 3 4
byte pinySloupce[4] = {6, 7, 8, 9}; //čísla pinu se sloupcem 1 2 3 4

MFRC522 rfid(SDA_PIN, RST_PIN); // vytvoření instance RFID čtečky z knihovny
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);
//inicializace klavesnice
Keypad klavesnice = Keypad( makeKeymap(keys), pinyRadku, pinySloupce, 4, 4);

void setup() {
    Serial.begin(9600); // komunikace přes sériovou linku rychlostí 9600 baud
    SPI.begin(); // inicializace komunikace přes SPI
    rfid.PCD_Init(); // inicializace komunikace s RFID čtečkou
    lcd.init(); //nastavení displeje
    lcd.backlight(); //zapnutí podsvícení displeje
    lcd.print("--Kodovy zamek--"); //pocatecni vypis
    //Vypis napovedy na seriový monitor
    Serial.println("-----Nápověda-----\nTlačítka na klávesnici: ");
    Serial.println("\t\t\t '1' '2' '3' 'A' |\n\t\t\t '4' '5' '6' 'B' |\n\t\t\t '7' '8' '9' 'C' |\n\t\t\t '*' '0' '#' 'D' |");
    Serial.println("Zadej 4-místný kód a potvrď '#'.");
    delay(4000);
}

void loop() //v loopu máme 3 funkce, které následně obsahují další funkce
{
```

```

//uvodní funkce odkud se muzeme dostat do menu, nebo zablockovaneho stavu
uvod(zadano,heslo,pokusy);
//Funkce, kdy je pripravek zablockovan
pripravek_je_zablockovan(pokusy, zadano, ser_kod, x);
//funkce menu
vstup_do_menu(pokusy, zadano, heslo, volba, novy_kod);
}

```

```

char uvod(char zadano[4], char heslo[4], int pokusy[1]) //hlavni funkce uvod
{
    //do zavorek wpisujeme promenne, se kterymi chceme pracovat
    pokusy[0]=3;                //nastaveni hodnoty na 3 pokusy
    lcd.clear();
    lcd.print("Zadej heslo pro");
    lcd.setCursor(0, 1);
    lcd.print(" prihlaseni...");
    delay(3000);                //vypis na lcd
}

```

```

do{
    prihlaseni_text();          //volani funkce prihlaseni_text
    //volani funkce pro zadavani kodu
    zadavani_kodu(zadano);
    //volani funkce pro srovnani zadaneho kodu s prihlasovacim heslem
    srovnavani_hesel(zadano, heslo, pokusy);
}while((pokusy[0] != 3)&&(pokusy[0] != 0));    //smycka pocita 3 pokusy
return pokusy[1];              //funkce return vraci hodnotu poctu pokusu
}

```

```

char pripravek_je_zablockovan(int pokusy[1], char zadano[4], char ser_kod[4], int x[1])
{
    //funkce pripravek je zablockovan realizuje stav, kdy pripravek se zablokuje, a pro jeho odblokovani je
    //nutno prilozit kartu nebo cip a zadat servisni kod
    if(pokusy[0] == 0)          //podminka vyhodnocuje stav z predesle funkce uvod
    {
        zablockovano();//volani funkce pro zablockovani pripravku, v teto funkci je take realizuje odblokovani
                                //pomoci RFID ctecky

        do{                      //smycka pro overovani spravnosti kodu
            ser_heslo_text();      //volani funkce pro vypis ser_heslo_text
            zadavani_kodu(zadano);  //volani funkce pro zadavani kodu
            srovnavani_ser_kodu(zadano, ser_kod, x); //volani funkce pro srovnani zadaneho a servisniho kodu
        }while(x[1] != 4);
    }
}

```

```

char vstup_do_menu(int pokusy[0], char zadano[4], char heslo[4], int volba[1], char novy_kod[4])
{
    //funkce vstup do menu nabizi moznost odhlaseni a zmenu prihlasovaciho hesla
    int i=0;                    //Nastaveni pomocne promenne
    if(pokusy[0] == 3)          //podminka vyhodnocuje stav z predesle funkce uvod
}

```

```

{
    uspech();                //volani funkce pro vypis uspesneho prihlaseni
    do{
        menu(volba);        //funkce pro zobrazeni nabidky menu
        if(volba[1] == 2)    //podminka vyhodnocuje volbu z funkce menu
        {
            zmena_hesla_text();    //volani funkce pro zemnu hesla text
            zadavani_kodu(zadano);    //volani funkce pro zadani prihlasovaciho kodu
            srovnani_hesel(zadano, heslo, pokusy); //volani funkce pro overeni spravnosti prihlasovaciho kodu

            if(pokusy[0]==3) //podminka vyhodnocuje spravnost hesla srovnanou ve funkci srovnani_hesel
            {
                lcd.clear();
                lcd.print("Zadej nove:");
                lcd.setCursor(0, 1);
                lcd.print("HESLO: ");    //vypis na lcd

                zadavani_kodu(zadano);    //funkce pro prvni zadani noveho kodu
                zadavani_noveho_kodu( novy_kod);    //funkce pro druhe zadani noveho kodu
                srovnani_novych_hesel(zadano, novy_kod); //funkce, která overuje spravnost novych kodu
            }
        }

        if(volba[1]==1)
        {
            odhlaseni_text(); i=1;    //volani funkce pro odhlaseni z menu
        }

    }while(i != 1);    //smyčka pro ukončení funkce menu
    }    //smyčka se ukončí jen pokud se odhlasíme
}

```

```

void prihlaseni_text()    //funkce pro uvodni vypis na lcd
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Volne pokusy: ");
    lcd.print(pokusy[0]);    //vypis poctu pokusu
    lcd.setCursor(0, 1);
    lcd.print("HESLO: ");
    Serial.print("\nHeslo: ");    //vypis na seriový monitor
}

```

```

void odhlaseni_text()    //funkce pro odhlaseni
{
    lcd.clear();
    lcd.print(" Odhlasovani...");
    delay(2000);
    lcd.clear();
}

```

```
lcd.print(" Jste odhlášen");  
delay(1500);  
}
```

```
void zmena_hesla_text() //funkce pro vypis zmenu hesla na lcd  
{  
  lcd.clear();  
  lcd.print(" Zvolili jste");  
  lcd.setCursor(0, 1);  
  lcd.print(" zmenu hesla");  
  delay(2000);  
  lcd.clear();  
  lcd.print("Zadej puvodni ");  
  lcd.setCursor(0, 1);  
  lcd.print("HESLO: ");  
  Serial.print("\nNové heslo: "); //vypis na seriový monitor  
}
```

```
void ser_heslo_text() //funkce pro vypis ser hesla na lcd  
{  
  lcd.clear();  
  lcd.print("Servisni heslo:");  
  lcd.setCursor(0, 1);  
  lcd.print("HESLO: ");  
  Serial.print("\nServisni heslo: "); //vypis na seriový monitor  
}
```

```
void zablokovano() //funkce pro zablokovaný stav přípravku  
{  
  int i=0; //deklarace a nastavení pomocné proměnné  
  lcd.clear();  
  lcd.print(" Zámek je ");  
  lcd.setCursor(0, 1);  
  lcd.print(" zablokovan... ");  
  delay(3000);  
  lcd.clear();  
  lcd.print("Odblokuj pomoci");  
  lcd.setCursor(0, 1);  
  lcd.print("karty nebo cipu ");  
  delay(4000);  
  lcd.clear(); //vypis na lcd  
  
  do{  
    i=0;  
    lcd.setCursor(0, 0);  
    lcd.print(" Přilož kartu");  
    lcd.setCursor(0, 1);  
    lcd.print(" nebo cip ");  
    delay(50); //vypis na lcd
```



```

rfid.PICC_IsNewCardPresent();      // kontrola RFID tagů v okolí modulu,
rfid.PICC_ReadCardSerial();        // kontrola správného přečtení RFID tagu

if(rfid.uid.uidByte[0] == 0xC9 & rfid.uid.uidByte[1] == 0x90 & rfid.uid.uidByte[2] == 0x7C &
                                                                    rfid.uid.uidByte[3] == 0xB3)

    { //srovnani spravnosti karty
        Serial.println("\nDetekovana bila karta!");i=1;    //vypis na seriový monitor hodnota i se zapise
    }

else if(rfid.uid.uidByte[0] == 0x09 & rfid.uid.uidByte[1] == 0x34 &
                                                                    rfid.uid.uidByte[2] == 0x04 & rfid.uid.uidByte[3] == 0xA4)

    { //srovnani spravnosti karty
        Serial.println("\nDetekovan modry privesek!");i=1; //vypis na seriový monitor, hodnota i se zapise
    }

}while(i != 1);    //nekonecna smyčka čeka na přiložení karty nebo cipu
}

void uspech()                //funkce pro výpis úspěcha na lcd
{
    lcd.clear();
    lcd.print("Přihlasovací kód");
    lcd.setCursor(0, 1);
    lcd.print("  PRIJAT");
    delay(2000);
}

char menu(int volba[1])      //je to funkce char, protože vrácí hodnotu
{
    //funkce menu nabízí odhlášení a změnu hesla
    volba[1] = 0;
    lcd.clear();
    lcd.print(" Vítejte v menu");
    lcd.setCursor(0, 1);
    lcd.print("-----");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Odhlášení ---> *");
    lcd.setCursor(0, 1);
    lcd.print("Změna hesla -> A");
    delay(4000);
    lcd.clear();                //výpis na lcd
    lcd.print("Zmákně klávesu:");

    do{
        char klavesa = klavesnice.getKey();    //nahrát hodnotu z klávesnice
        if (klavesa)
        {
            lcd.setCursor(0, 1);
            lcd.print(klavesa);                //výpis klávesy na lcd
        }
    }
}

```

```

    if(klaveses == '*'){volba[1]=1;delay(300);}    //Podminky pro vyhodnoceni volby
    if(klaveses == 'A'){volba[1]=2;delay(300);}

}while((volba[1] != 1)&&(volba[1] != 2));    //smyčka konci, az zmackneme spravne tlačitko
return volba[1];    //funkce vraci hodnotu volba[1]
}

```

```

char zadavani_kodu(char zadano[4])    //funkce pro zadavani kodu na klavesnici
{
    int y, misto=7;    //deklarace promennych
    for(y=0;y<4;y++)    //smyčka for pro 4 znaky
    {
        char klaveses = klavesnice.getKey();    //nacita hodnotu z klavesnice
        if (klaveses)
        {
            if(klaveses != '#')    //kontrola, jestli jsme nezmackli potvrzovací znak
            {
                lcd.setCursor(misto, 1);
                lcd.print("*");    //vypis na lcd
                Serial.print(klaveses);    //vypis na seriový monitor
                zadano[y] = klaveses;    //ulozeni hodnoty z klavesy do pole
                misto++;
            }
            else{y--;}    //pokud jsme nic nezmackli smyčka odečte y--;
        }
        else {y--;}
    }
    do{    //smyčka pro zadani '#' = ukončuje zadavani kodu
        y=2;
        char klaveses = klavesnice.getKey();
        if (klaveses == '#'){y=1;}
    }while(y != 1);

    return zadano[4];    //funkce return vraci pole zadano
}

```

```

char zadavani_noveho_kodu(char novy_kod[4])    //funkce pro zadavani noveho kodu
{
    int y, misto=7;    //deklarace promennych
    lcd.clear();
    lcd.print("Zadej znovu:");
    lcd.setCursor(0, 1);
    lcd.print("HESLO: ");    //vypis na lcd

    for(y=0;y<4;y++)    //smyčka pro zadani noveho kodu
    {
        char klaveses = klavesnice.getKey();
        if (klaveses)
        {

```

```

    if(klavesa != '#')
    {
        lcd.setCursor(misto, 1);
        lcd.print("*");
        Serial.print(klavesa);
        novy_kod[y] = klavesa;      //kod je stejný jak funkce zadavani_kodu, jen hodnotu z klavesy
        misto++;                    //ukladame do novy_kod[4]
    }
    else{y--;}
}
else {y--;}
}
do{
    y=2;
    char klavesa = klavesnice.getKey();
    if (klavesa == '#'){y=1;}
}while(y != 1);
return novy_kod[4];
}

```

```

char srovnani_hesel(char zadano[4], char heslo[4], int pokusy[1]) //funkce pro srovnani zadaneho
{
    int i=0;                //deklarace promennych

    for(int y=0;y<4;y++)    //smyčka pro vyhodnocení stejnosti hesel
    {                        kodu z přihlasovacím
        if(heslo[y] == zadano[y]) { i++;}
    }

    if(i<4)                //pokud nejsou všechny znaky shodné, na lcd se vypíše neúspěch a ubude nám pokus
    {
        lcd.clear();
        lcd.print("špatně!!!");
        delay(1000);
        pokusy[0]--;
    }

    if(i == 4){pokusy[0]=3;} //pokud jsme zadali dobrý kód až na potřetí, do proměnné pokusy se uloží
                            //hodnota 3 pro jednoduchost programu

    return pokusy[1];      //funkce return vrací počet pokusu.
}

```

```

char srovnani_novych_hesel(char zadano[4], char novy_kod[4]) //funkce pro srovnání dvou nových kódů
{
    int i=0, y;            //deklarace promennych

    for(y=0;y<4;y++)
    {
        if(novy_kod[y] == zadano[y]) {i++;}    //smyčka pro vyhodnocení stejnosti hesel
    }
}

```

```

if(i<4)
{
    lcd.clear();
    lcd.print("Hesla nesouhlasí ");    //pokud nejsou všechny znaky shodné, na lcd se vypíše neúspěch
    delay(1000);
}

if(i == 4)                                //pokud jsou hesla stejné, nové heslo se uloží
{
    lcd.clear();
    lcd.print("Nové heslo");
    lcd.setCursor(0,1);
    lcd.print("uloženo");
    delay(3000);
    for(y=0;y<4;y++)
    {
        heslo[y] = zadano[y] ;          //ukládání nového hesla
    }
}
}

char srovnávání_ser_kodu(char zadano[4], char ser_kod[4], int x[1])
{//funkce sloužící ke srovnání zadaného kódu se servisním
int y;                                //deklarace proměnných
x[1]=0;

for(y=0;y<4;y++)
{
    //srovnávání zadaného kódu s servisním
    if(ser_kod[y] == zadano[y]) { x[1]++;}
    Serial.print(zadano[y]);          //použitím jednoduchých proměnných řídím smyčku do-while
}

if(x[1]<4)
{
    //zadaný kód se neshoduje se servisním, vypis na lcd
    lcd.clear();
    lcd.print("Špatné!!!");
    delay(1000);
}

else
{
    //zadaný kód se shoduje se servisním, vypis na lcd
    lcd.clear();
    lcd.print(" Servisní kód");
    lcd.setCursor(0,1);
    lcd.print(" PRIJAT");
    delay(3000);
}

return x[1];    //funkce return vrací hodnotu, která je pak použita ve funkci přípravek_je_zablokována
}

```

Závěr:

Úkolem bylo vytvořit funkční kódový zámek, který budeme ovládat klávesnicí, a výsledek se nám bude zobrazovat na lcd displeji. Po nahrání programu do přípravku se zobrazí na displeji možnost přihlášení do menu.

První možnost je, že zadáme 3x chybně přihlašovací kód, zámek se zablokuje a ocitneme se v nabídce pro servisní heslo. Musíme přiložit na čtečku RFID modrý čip nebo bílou kartu, jinak zámek nereaguje. Po přiložení se objeví možnost zadání servisního kódu. Zde není omezen počet pokusů, takže až tehdy, když zadáme správné servisní heslo, tak se zámek otevře a znovu se ocitneme v nabídce přihlášení do menu.

Druhá možnost je, že jsme zadali správné přihlašovací heslo a dostali jsme se do nabídky menu. Zde si můžeme vybrat mezi odhlášením a změnou hesla. Pokud zvolíme změnu hesla, následně musíme zadat původní přihlašovací kód. Pokud jsme ho zapsali nesprávně, vrátí nás program zpět do menu. Pokud jsme ho napsali správně, následně nás program dvakrát vyzve pro zapsání nového hesla. Pokud byly nové hesla totožná, heslo se uložilo, pokud ne, znovu nás program vrátil do menu. Po výběru odhlásit nás program automaticky odhlásí a přesune do nabídky přihlášení do menu.

Úloha byla nejnáročnější ze všech zadaných úkolů.

Nejnáročnější pro mě bylo přijít na princip funkcí, a pochopit řídicí příkazy klávesnice a čtečky. Kvůli rozsáhlosti programu je bohužel trochu nepřehledný.

Tato úloha může být použita lehce v praxi. Díky klávesnici můžeme zadat kód a zámek se nám otevře. Zámek pojištěn proti zlodějům a po zablokování lze zámek otevřít jen pokud máme přístupový čip nebo kartu. Po doplnění houkačky by mohl zámek aktivně reagovat na případné zloděje.