

```

//deklarace, nastavení wifi - esp8266 - ↓↓↓

#define BLYNK_PRINT Serial // posílá info. na sériový monitor (lze vymazat)
#include <ESP8266_Lib.h> // načtení knihovny ESP modulu
#include <BlynkSimpleShieldEsp8266.h> // načtení knihovny Blynk
char auth[] = "c58f18e7065e411783c1545bedde339c"; //token Blynk aplikace
char ssid[] = "HOTSPOT_DUM"; // název Wi-Fi
char pass[] = "123456789"; // heslo Wi-Fi
char server[] = "blynk-cloud.com"; // vzdálený server
int port = 8080; // tcp port
#define EspSerial Serial1 // sériová komunikace na pinech RX1 a TX1
#define ESP8266_BAUD 115200 // rychlost přenosu
ESP8266 wifi(&EspSerial);
BlynkTimer timer; // Blynk časovač
unsigned long cas_reconnect; // čas dalšího pokusu o připojení k Wi-Fi

//deklarace, nastavení wifi - esp8266 - ↑↑↑

//deklarace - alarm ↓↓↓
#include <PCF8574.h> // načte knihovnu PCF expandéru (I2C)
#include <Keypad_I2C.h> // načte knihovnu klávesnice (I2C)
#include <LiquidCrystal_I2C.h> // načte knihovnu displeje (I2C)
#include <Keypad.h>
#include <Wire.h>
#define I2CADDR 0x38 // adresa I2C klávesnice je 0x38
unsigned long timeBacklight; // časovač, zhasnutí displeje (1 min)
boolean stavBacklight; // aby neběžela furt dokola funkce: "backlight()"
String heslo = "1234"; // standartní heslo
String zadHeslo; // zadávané heslo
String noveHeslo; // změněné heslo
int cidloAlarm; // čidlo pohybu DHT
unsigned long timeCidloAlarm = 0; // odešle alarm na Blynk max jednou za 5s
boolean alarmActivated = true; // stav alarmu (zapnutý/vypnutý) (S1/S2)
boolean hesloOvereno = false; // bylo zadáno správné heslo (ano/ne)
boolean modeAktivaceAlarmu; // stav - alarm off - aktivace alarmu
boolean modeMenu; // stav - alarm off - hlavní menu
boolean modeZmenaHesla = false; // stav - alarm off - změna hesla
boolean nastaveniProZadaniHesla = 0; // inicializuje prostředí pro zadání hesla
boolean nastaveniProZmenuHesla = 0; // inicializuje prostředí pro změnu hesla
boolean doubleClickCounter = 0; // zabraňuje dvojitisku * nebo # při změně stavu
int k; // určuje pozici zadávaného hesla
int j; // určuje pozici nového hesla
char keypressed; // detekce stisku klávesy
const byte ROWS = 4; //four rows // klávesnice má 4 řádky
const byte COLS = 3; //three columns // klávesnice má 3 sloupce

```

```

char keys[ROWS][COLS] = {           // nastaví znakové pole klávesnice
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte rowPins[ROWS] = {5, 0, 1, 3}; // přidělení řádků k pinům (PCF8574)
byte colPins[COLS] = {4, 6, 2};    // přidělení sloupců k pinům (PCF8574)
                                   // spojení všech parametrů klávesnice
Keypad_I2C kpd( makeKeymap(keys), rowPins, colPins, ROWS, COLS, I2CADDR, PCF8574);

LiquidCrystal_I2C lcd(0x27, 20, 4); // nastavení adresy I2C (0x27), počet řádků a sloupců, LCD (20x4)

//deklarace - alarm ↑↑↑

//deklarace - teplotní čidlo - ↓↓↓
#include "DHT.h"                    // načte knihovnu DHT
#define pinDHT 23                   // DHT čidlo (pin)
DHT dht(pinDHT, DHT22);
float teplota;                      // ukládá teplotu
float vlhkost;                      // ukládá vlhkost
unsigned long timeDHT = 0;          // časovač, načte údaje ze senzoru jednou za 5 sec
boolean ukazTeplotu;               // ukáže teplotu na displeji

//deklarace - teplotní čidlo - ↑↑↑

//deklarace - čidlo pohybu - ↓↓↓

const int sensorPin = 13;           // PIR senzor (pin)
const int buzcak = 11;              // bzučák (pin)

//deklarace - čidlo pohybu - ↑↑↑

//deklarace - osvětlení - ↓↓↓

// vnitřní osvětlení 1 (inteligentní elektroinstalace)
int prepinaPin1 = 3;                // reálný přepínač (pin)
int ledPin1 = 2;                    // LED dioda (pin)
boolean prepinaState1 = 0;          // proměnná která uchovává stav přepínače
boolean lastPrepinaState1 = 0;      // proměnná aktuálního stavu přepínače
boolean counterPrepinaState1 = 0;   // brání znovunačtení - prepinaState1
int pinValue1 = 0;                  // proměnná stavu virt. přepínače (Blynk)

// vnitřní osvětlení 2 (inteligentní elektroinstalace)
int prepinaPin2 = 4;                // reálný přepínač (pin)
int ledPin2 = 5;                    // LED dioda (pin)
boolean prepinaState2 = 0;          // proměnná která uchovává stav přepínače
boolean lastPrepinaState2 = 0;      // proměnná aktuálního stavu přepínače

```

```

boolean counterPrepinacState2 = 0;    // brání znovunačtení - prepinacState2
int pinValue2 = 0;                    // proměnná stavu virt. přepínače (Blynk)

// vnitřní osvětlení 3 (inteligentní elektroinstalace)
int prepinacPin3 = 7;                  // reálný přepínač (pin)
int ledPin3 = 6;                       // LED dioda (pin)
boolean prepinacState3 = 0;           // proměnná která uchovává stav přepínače
boolean lastPrepinacState3 = 0;       // proměnná aktuálního stavu přepínače
boolean counterPrepinacState3 = 0;    // brání znovunačtení - prepinacState3
int pinValue3 = 0;                    // proměnná stavu virt. přepínače (Blynk)

//venkovní osvětlení (inteligentní elektroinstalace)
int buttonPin = 8;                    // reálné tlačítko (pin)
int ledPin4 = 10;                     // LED dioda (pin)
boolean buttonState = 0;              // proměnná stavu tlačítka
int pinValue4 = 0;                    // proměnná stavu virtuálního tlačítka (Blynk)
int R = 0;                             // proměnná počtu stisků tlačítka R(1-3)
unsigned long time1;                  // časovač
int odpocet = 0;                      // odpočet na displeji

// stavy pro přepínač-ledka 1
boolean sa1 = true;                   // stav sa1 (S1 pro 1. pokoj) - zhasnuto
boolean sa2 = false;                  // stav sa2 (S2 pro 1. pokoj) - rozsvíceno

// stavy pro přepínač-ledka 2
boolean sb1 = true;                   // stav sb1 (S1 pro 2. pokoj) - zhasnuto
boolean sb2 = false;                  // stav sb2 (S2 pro 2. pokoj) - rozsvíceno

// stavy pro přepínač-ledka 3
boolean sc1 = true;                   // stav sc1 (S1 pro 3. pokoj) - zhasnuto
boolean sc2 = false;                  // stav sc2 (S2 pro 3. pokoj) - rozsvíceno

// stavy pro tlačítko-ledka
boolean sd1 = true;                   // (S1)zhasnuto a nesepnuté tlačítko
boolean sd2 = false;                  // (S2)rozsvíceno a sepnuté tlačítko
boolean sd3 = false;                  // (S3)rozsvíceno a uvolněné tlačítko, neznáme R(1-3)
boolean sd4 = false;                  // (S4)rozsvíceno, známe R, neznáme t(10,60,600 sec)
boolean sd5 = false;                  // (S5)rozsvíceno a známe t
boolean sd6 = false;                  // (S6)zhasnuto a sepnuté tlačítko

BLYNK_WRITE(V1)                       // načte data z V1 (virtuální přepínač 1)
{ pinValue1 = param.asInt(); }

BLYNK_WRITE(V2)                       // načte data z V2 (virtuální přepínač 2)
{ pinValue2 = param.asInt(); }

BLYNK_WRITE(V3)                       // načte data z V3 (virtuální přepínač 3)
{ pinValue3 = param.asInt(); }

```

```

BLYNK_WRITE(V4)                                // načte data z V4 (virtuální tlačítko)
{ pinValue4 = param.asInt(); }

//deklarace - osvětlení - ↑↑↑

//deklarace - topení - ↓↓↓

int mosfet = 12;                               // mosfet tranzistor (pin)
int potenciometr = A13;                       // potenciometr (analogový pin)
int prepočetState;                            // přepoččet analogových hodnot (uložených)
int prepočet = 0;                             // přepoččet analogových hodnot (aktuálních)
boolean counterPrepočetState = 0;            // zabrání přepsání - prepočetState
boolean counterBlynk = 0;                   // zabrání přepsání - pinValue8State
int pinValue8State;                          // hodnota na Blynku (uložená)
int pinValue8 = 0;                           // hodnota na Blynku (aktuální)
int rizeniTopeni = 0;                       // kdo řídí: 1 - potenciometr, 2 - Blynk
unsigned long casSynchro = 0;               // čas synchronizace mezi potenciometry
BLYNK_WRITE(V8)                               // načte data z V8 (virtuální potenciometr)
{ pinValue8 = param.asInt(); }

//deklarace - topení - ↑↑↑

void setup(){

  Serial.begin(9600);

  // lcd, keypad, dht a teplotní čidlo
  Wire.begin();
  kpd.begin( makeKeymap(keys) );
  lcd.begin();
  lcd.backlight();
  timeBacklight = millis() + 60000;
  dht.begin();
  lcd.setCursor(0,0);
  lcd.print("***INICIALIZATION***");

  //senzor pohybu a bzučák
  pinMode(senzorPin, INPUT);
  pinMode(bzucak, OUTPUT);

  //wifi
  EspSerial.begin(ESP8266_BAUD);              // start sériové komunikace s ESP
  delay(10);
  Blynk.config(wifi, auth, server, port);     // nastavení připojení
  Serial.println("A");
  Blynk.connectWiFi(ssid, pass);             // připojení k Wi-Fi
  Serial.println("B");
  Blynk.connect();                           // připojení přes vzdálený server
  Serial.println("C");

```

```

timer.setInterval(1000L, myTimerEvent);           // využívá se k odpočtu osvětlení
cas_reconnect = millis() + 600000;               // interval pokusů o připojení

//osvětlení
pinMode (ledPin1, OUTPUT);
pinMode (ledPin2, OUTPUT);
pinMode (ledPin3, OUTPUT);
pinMode (ledPin4, OUTPUT);
pinMode (prepinacPin1, INPUT_PULLUP);
pinMode (prepinacPin2, INPUT_PULLUP);
pinMode (prepinacPin3, INPUT_PULLUP);
pinMode (buttonPin, INPUT_PULLUP);
digitalWrite(ledPin1, LOW);
digitalWrite(ledPin2, LOW);
digitalWrite(ledPin3, LOW);
digitalWrite(ledPin4, LOW);
Blynk.virtualWrite(V1, LOW);
Blynk.virtualWrite(V2, LOW);
Blynk.virtualWrite(V3, LOW);
}

void loop(){

// wifi
if (Blynk.connected()){                          // pokud je připojeno k Wi-Fi:
  Blynk.run();                                    // Blynk běží
  timer.run();                                    // čas běží
}
else if (millis() > cas_reconnect)               // není připojeno a doběhl časovač:
{
  Blynk.connectWiFi(ssid, pass);                 // pokus se připojit
  Blynk.connect();                               // pokus se zahájit komunikaci
  cas_reconnect = millis() + 60000;             // nastav časovač
}

// vnitřní osvětlení 1 (inteligentní elektroinstalace)
//stav 1 - zhasnuto
if (sa1) {                                        //stav 1 - zhasnuto
  if ( counterPrepinacState1 == 0) {            // pokud není zabráněno načtení přepínače:
    prepinacState1 = digitalRead(prepinacPin1); // načti stav přepínače
    counterPrepinacState1 = 1;                  // zabraň znovunačtení a přepsání stavu
  }
  lastPrepinacState1 = digitalRead(prepinacPin1); // načti poslední stav přepínače
  if (prepinacState1 != lastPrepinacState1) {  // pokud se stavy nerovnejí:
    digitalWrite(ledPin1, HIGH);               // rozsvítí LED
    counterPrepinacState1 = 0;                  // povolí znovunačtení a přepsání stavu
    sa1 = false;                               // ukončí stav S1
    sa2 = true;                                // povolí přechod do stavu S2
    Blynk.virtualWrite(V1, HIGH);              // nastaví virtuální přepínač na HIGH
  }
}

```

```

    pinValue1 = 1; // stav virtuálního přepínače
}
if (pinValue1 == 1 && sa1 == true ) { // pokud byl stisknut virt. přepínač:
    digitalWrite(ledPin1, HIGH); // rozsvítí LED
    counterPrepinacState1 = 0; // povolí znovunačtení a přepsání stavu
    sa1 = false; // ukončí stav S1
    sa2 = true; // povolí přechod do stavu S2
}
}
// stav - 2 rozsvíceno
if (sa2) { //stav 2 - rozsvíceno
    if ( counterPrepinacState1 == 0) { // pokud není zabráněno načtení přepínače:
        prepinacState1 = digitalRead(prepinacPin1); // načti stav přepínače
        counterPrepinacState1 = 1; // zabraň znovunačtení a přepsání stavu
    }
    lastPrepinacState1 = digitalRead(prepinacPin1); // načti poslední stav přepínače
    if (prepinacState1 != lastPrepinacState1) { // pokud se stavy nerovnejí:
        digitalWrite(ledPin1, LOW); // zhasne LED
        counterPrepinacState1 = 0; // povolí znovunačtení a přepsání stavu
        sa1 = true; // povolí přechod do stavu S1
        sa2 = false; // ukončí stav S2
        Blynk.virtualWrite(V1, LOW); // nastaví virtuální přepínač na LOW
        pinValue1 = 0; // stav virtuálního přepínače
    }
    if (pinValue1 == 0 && sa2 == true ) { // pokud byl stisknut virt. přepínač:
        digitalWrite(ledPin1, LOW); // zhasne LED
        counterPrepinacState1 = 0; // povolí znovunačtení a přepsání stavu
        sa1 = true; // povolí přechod do stavu S1
        sa2 = false; // ukončí stav S2
    }
}
}

// vnitřní osvětlení 2 (inteligentní elektroinstalace)
//stav 1 - zhasnuto
if (sb1) {
    if ( counterPrepinacState2 == 0) {
        prepinacState2 = digitalRead(prepinacPin2);
        counterPrepinacState2 = 1;
    }
    lastPrepinacState2 = digitalRead(prepinacPin2);
    if (prepinacState2 != lastPrepinacState2) {
        digitalWrite(ledPin2, HIGH);
        counterPrepinacState2 = 0;
        sb1 = false;
        sb2 = true;
        Blynk.virtualWrite(V2, HIGH);
        pinValue2 = 1;
    }
}
if (pinValue2 == 1 && sb1 == true ) {

```

```

    digitalWrite(ledPin2, HIGH);
    counterPrepinacState2 = 0;
    sb1 = false;
    sb2 = true;
}
}
// stav - 2 rozsvíceno
if (sb2) {
    if ( counterPrepinacState2 == 0) {
        prepinacState2 = digitalRead(prepinacPin2);
        counterPrepinacState2 = 1;
    }
    lastPrepinacState2 = digitalRead(prepinacPin2);
    if (prepinacState2 != lastPrepinacState2) {
        digitalWrite(ledPin2, LOW);
        counterPrepinacState2 = 0;
        sb1 = true;
        sb2 = false;
        Blynk.virtualWrite(V2, LOW);
        pinValue2 = 0;
    }
    if (pinValue2 == 0 && sb2 == true ) {
        digitalWrite(ledPin2, LOW);
        counterPrepinacState2 = 0;
        sb1 = true;
        sb2 = false;
    }
}

// vnitřní osvětlení 3 (inteligentní elektroinstalace)
//stav 1 - zhasnuto
if (sc1) {
    if ( counterPrepinacState3 == 0) {
        prepinacState3 = digitalRead(prepinacPin3);
        counterPrepinacState3 = 1;
    }
    lastPrepinacState3 = digitalRead(prepinacPin3);
    if (prepinacState3 != lastPrepinacState3) {
        digitalWrite(ledPin3, HIGH);
        counterPrepinacState3 = 0;
        sc1 = false;
        sc2 = true;
        Blynk.virtualWrite(V3, HIGH);
        pinValue3 = 1;
    }
    if (pinValue3 == 1 && sc1 == true ) {
        digitalWrite(ledPin3, HIGH);
        counterPrepinacState3 = 0;
        sc1 = false;
    }
}

```

```

        sc2 = true;
    }
}
// stav - 2 rozsvíceno
if (sc2) {
    if ( counterPrepinacState3 == 0) {
        prepinacState3 = digitalRead(prepinacPin3);
        counterPrepinacState3 = 1;
    }
    lastPrepinacState3 = digitalRead(prepinacPin3);
    if (prepinacState3 != lastPrepinacState3) {
        digitalWrite(ledPin3, LOW);
        counterPrepinacState3 = 0;
        sc1 = true;
        sc2 = false;
        Blynk.virtualWrite(V3, LOW);
        pinValue3 = 0;
    }
    if (pinValue3 == 0 && sc2 == true ) {
        digitalWrite(ledPin3, LOW);
        counterPrepinacState3 = 0;
        sc1 = true;
        sc2 = false;
    }
}

// venkovni chytne osvetleni
if (sd1) {buttonState = digitalRead (buttonPin); // stav 1 - zhasnuto
    if (!buttonState == HIGH || pinValue4 == 1) { // stisk tlačítka:
        digitalWrite(ledPin4, HIGH); // rozsvítí LED
        sd1 = false; // ukončí stav S1
        sd2 = true; // povolí přechod do stavu S2
    }
}
if (sd2) {buttonState = digitalRead (buttonPin); // stav 2 - rozsvíceno
    if (!buttonState == LOW && pinValue4 == 0) { // uvolnění tlačítka:
        R = R+1; // čítač stisků
        time1 = millis() + 2000; // nastaví časovač na 2 sec.
        sd2 = false; // ukončí stav S2
        sd3 = true; // povolí přechod do stavu S3
    }
}
if (sd3) { // stav 3 - rozsvíceno
    if (millis() > time1 || R == 3) { // pokud R = 3 nebo doběhl
        sd3 = false; // ukončí stav S3
        sd4 = true; // povolí přechod do stavu S4
    }
}
else { buttonState = digitalRead (buttonPin); // načte stav tlačítka
    if (!buttonState == HIGH || pinValue4 == 1) { // stisk tlačítka:

```



```

        sd3 = false;           // ukončí stav S3
        sd2 = true;           // navrátí stav S2
    }
}
}
if (sd4) {                   // stav 4 - rozsvíceno
    if (R == 1) {            // pokud R = 1, tak:
        time1 = millis() + 10000; // nastaví časovač na 10 ses
        odpocet = 10;        // nastaví odpočet na displeji
    }
    else if (R == 2) {       // pokud R = 2, tak:
        time1 = millis() + 60000; // nastaví časovač na 60 sec
        odpocet = 60;        // nastaví odpočet na displeji
    }
    else {                   // jinak:
        time1 = millis() + 600000; // nastaví časovač na 600 ses
        odpocet = 600;       // nastaví odpočet na displeji
    }
    sd4 = false;             // ukončí stav S4
    sd5 = true;              // povolí přechod do stavu S5
}
if (sd5) {buttonState = digitalRead (buttonPin); // stav 5 - rozsvíceno
    if (millis() > time1) { // pokud doběhne časovač:
        digitalWrite(ledPin4, LOW); // zhasne led
        R = 0; // nastaví R na 0
        sd5 = false; // ukončí stav S5
        sd1 = true; // navrátí stav S1
    }
    if (!buttonState == HIGH || pinValue4 == 1) { // stisk tlačítka:
        digitalWrite(ledPin4, LOW); // zhasne led
        R = 0; // nastaví R na 0
        odpocet = 0; // ukončí odpočet na displeji
        sd5 = false; // ukončí stav S5
        sd6 = true; // povolí přechod do stavu S5
    }
}
if (sd6) {buttonState = digitalRead (buttonPin); // stav 6 - zhasnuto
    if (!buttonState == LOW && pinValue4 == 0) { // uvolnění tlačítka:
        sd6 = false; // ukončí stav S6
        sd1 = true; // navrátí stav S1
    }
}
}

//dht_senzor
if (millis() > timeDHT) { // po doběhnutí časovače:
    teplota = dht.readTemperature(); // uloží teplotu do proměnné
    vlhkost = dht.readHumidity(); // uloží vlhkost do proměnné
    timeDHT = millis()+ 5000; // nastaví časovač na 5 sec.
    ukazTeplotu = true; // ukáže/změní teplotu na displeji
}

```



```

stavBacklight = 1; // nastaví stav na 1 (svítí)
}

// pokud je alarm - on ... S1
if (alarmActivated == true) { // pokud je alarm aktivován, tak:
    cidloAlarm = digitalRead(senzorPin); // načte stav pohybového senzoru
    digitalWrite(bzucak, cidloAlarm); // při pohybu spustí poplach
    if (cidloAlarm == 1 && millis() > timeCidloAlarm ) { // pokud je detekován pohyb, tak:
        Blynk.notify("POZOR ALARM!"); // pošle informaci na mobilní telefon
        timeCidloAlarm = millis() + 5000; // nastaví časovač
    }
}

if (hesloOvereno) { // pokud je heslo ověřeno, tak:
    digitalWrite(bzucak, LOW); // vypne bzučák
    alarmActivated = false; // deaktivuje alarm
    hesloOvereno = false; // vrátí ověření hesla do stavu - neověřeno
    lcd.clear(); // vyčistí displej
    lcd.setCursor(0,0); // nastaví kurzor do polohy (0,0)
    lcd.print(" *** alarm off *** "); // zobrazí se na displeji text
    delay (2000); // počká dvě vteřiny
    ukazMenu(); // volání funkce (inicializace hlavního menu)
}
else {
    enterPassword(); // funkce, která ověřuje heslo
}
}

// pokud je alarm - off... S2
if (alarmActivated == false) { // pokud je alarm deaktivován:
    if (keypressed == '*' && modeMenu == true ) { // stisk hvězdičky v hlavním menu:
        modeAktivaceAlarmu = true; // povolí přechod do menu aktivace alarmu
        modeMenu = false; // opustí hlavní menu
        doubleClickCounter = 1; // zabrání nechtěnému proklikání
    }
    if (keypressed == '#' && modeMenu == true ) { // stisk křížku v hlavním menu
        modeZmenaHesla = true; // povolí přechod do menu změny hesla
        modeMenu = false; // opustí hlavní menu
        doubleClickCounter = 1; // zabrání nechtěnému proklikání
    }
}
if (modeMenu == true && ukazTeplotu == true) { // každých 5 sec. je zobrazena aktuální teplota
    if (isnan(teplota) || isnan(vlhkost)){
        lcd.setCursor(9,2);
        lcd.print("no input");
        lcd.setCursor(9,3);
        lcd.print("no input");
    }
    else {
        lcd.setCursor(9,2);
        lcd.print(teplota);
    }
}

```

```

        lcd.print(" C ");
        lcd.setCursor(9,3);
        lcd.print(vlhkost);
        lcd.print(" % ");
    }
    ukazTeplotu = false;
}

if (modeZmenaHesla == true) { // pokud bylo zažádáno o změnu hesla:
    zmenaHesla(); // přejde do funkce na změnu hesla
}

if (modeAktivaceAlarmu == true) { // pokud bylo zažádáno o aktivaci alarmu:
    if (hesloOvereno) { // pokud je heslo ověřeno
        alarmActivated = true; // aktivuje alarm
        hesloOvereno = false; // vrátí ověření hesla do stavu - neověřeno
        modeAktivaceAlarmu = false; // odejde z menu pro nastavení alarmu
        lcd.clear(); // vyčistí displej
        lcd.setCursor(0,0); // nastaví kurzor do polohy (0,0)
        lcd.print(" *** alarm on *** "); // zobrazí se na displeji text
        delay (2000); // počká dvě vteřiny
    }
    else { // funkce, která ověřuje heslo
        enterPassword();
    }
}
}

// funkce co ověřuje heslo (aktivaci i deaktivaci)
void enterPassword() {
    if (nastaveniProZadaniHesla == 0){ // nastaví menu pro zadávání hesla:
        k=5; // heslo se zadává od 6. sloupce
        zadHeslo = ""; // smaže zadané heslo
        lcd.clear(); // vyčistí displej
        lcd.setCursor(0,0); // nastaví kurzor do polohy (0,0)
        lcd.print(" *** ALARM *** "); // zobrazí se na displeji text
        lcd.setCursor(0,1); // nastaví kurzor do polohy (0,1)
        lcd.print("Pass>"); // zobrazí se na displeji text
        nastaveniProZadaniHesla = 1; // brání znovunastavení tohoto menu
    }
    if (hesloOvereno == false) { // pokud heslo není ověřeno
        if (keypressed != NO_KEY){ // pokud bylo stisknuté tlačítko:
            if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
                keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
                keypressed == '8' || keypressed == '9' ) {
                zadHeslo += keypressed; // uloží číslo a posune se o 1 pozici
                lcd.setCursor(k,1); // nastaví kurzor na pozici - k
                lcd.print("*"); // zobrazuje místo čísla hvězdičku
                k++; // k = k+1;
            }
        }
    }
}

```

```

    }
}
// stisk křížku, nezadané heslo, alarm neaktivní:
if (keypressed == '#' && zadHeslo == "" && alarmActivated == false ) {
    modeAktivaceAlarmu = false; // odejde z menu pro nastavení alarmu
    nastaveniProZadaniHesla = 0;
    ukazMenu(); // volání funkce (inicializace hlavního menu)
}
// stisk křížku, zadané heslo:
if (k > 9 || (keypressed == '#' && zadHeslo != "" )) {
    nastaveniProZadaniHesla = 0; //obnoví menu pro zadávání hesla
}
// stisk hvězdičky:
if ( keypressed == '*' && doubleClickCounter == 0 ) {
    if ( zadHeslo == heslo ) { // pokud se hesla shodují:
        hesloOvereno = true; // heslo je ověřeno
        nastaveniProZadaniHesla = 0;
    }
    else if (zadHeslo != heslo) { // pokud se hesla neshodují:
        lcd.setCursor(0,1); // nastaví kurzor do polohy (0,1)
        lcd.print("Wrong! Try Again"); // zobrazí se na displeji text
        delay(2000); // počká 2 sec.
        nastaveniProZadaniHesla = 0; //obnoví menu pro zadávání hesla
    }
}
doubleClickCounter = 0;
}
}
//funkce co změní heslo
void zmenaHesla() {

    if (nastaveniProZmenuHesla == 0) { // nastaví menu pro změnu hesla:
        lcd.clear(); // vyčistí displej
        lcd.setCursor(0,0); // nastaví kurzor do polohy (0,0)
        lcd.print(" *** ZMENA HESLA *** "); // zobrazí se na displeji text
        lcd.setCursor(0,1); // nastaví kurzor do polohy (0,1)
        lcd.print("Old Pass>"); // zobrazí se na displeji text
        lcd.setCursor(0,2); // nastaví kurzor do polohy (0,2)
        lcd.print("new Pass>"); // zobrazí se na displeji text
        k=9; // původní heslo se zadává od 10. sloupce
        j=9; // nové heslo se zadává od 10. sloupce
        zadHeslo = ""; // smaže zadané původní heslo
        noveHeslo = ""; // smaže zadané nové heslo
        nastaveniProZmenuHesla =1; // brání znovunastavení tohoto menu
    }

    if (keypressed != NO_KEY){ // pokud bylo stisknuté tlačítko:
        if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
            keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
            keypressed == '8' || keypressed == '9' ) {
            if (k < 13) { // pokud bylo zadáno míň než pět znaků
                zadHeslo += keypressed; // uloží číslo a posune se o 1 pozici
            }
        }
    }
}

```

```

    lcd.setCursor(k,1);           // nastaví kurzor na pozici - k
    lcd.print("*");             // zobrazuje místo čísla hvězdičku
    k++;                        // k = k+1;
}
else {                          // pokud byl zadán již pátý znak
    noveHeslo += keypressed;    // uloží číslo a posune se o 1 pozici
    lcd.setCursor(j,2);         // nastaví kurzor na pozici - j
    lcd.print(keypressed);      // zobrazuje zadané číslo
    j++;                        // j = j+1;
}
}
}
// stisk křížku, nezadané heslo:
if (keypressed == '#' && zadHeslo == "" && doubleClickCounter == 0) {
    modeZmenaHesla = false;     // opustí menu pro změnu hesla
    nastaveniProZmenuHesla =0;
    ukazMenu();                // navrátí hlavní menu
}
// stisk křížku, zadané heslo:
if (keypressed == '#' && zadHeslo != "" && doubleClickCounter == 0) {
    nastaveniProZmenuHesla =0;  //obnoví menu pro zadávání hesla
}
if (j > 13) {                  // zadáno příliš mnoho znaků:
    nastaveniProZmenuHesla =0;  //obnoví menu pro zadávání hesla
}
if (keypressed == '*') {      // stisk hvězdičky:
    if (zadHeslo == heslo && j == 13) { // hesla se shodují a jsou zadány 4 znaky:
        heslo = noveHeslo;         // původní heslo nahrazeno novým
        lcd.clear();               // vyčistí displej
        lcd.setCursor(0,0);        // nastaví kurzor do polohy (0,0)
        lcd.print(" *** pw changed *** "); // zobrazí se na displeji text
        delay (2000);              // počká 2 sec.
        nastaveniProZmenuHesla = 0;
        modeZmenaHesla = false;    // opustí menu pro změnu hesla
        ukazMenu();                // přejde do hlavního menu
    }
    else {                          // pokud je heslo zadáno nesprávně:
        lcd.clear();
        lcd.setCursor(0,0);
        if (zadHeslo == heslo) {
            lcd.setCursor(0,0);
            lcd.print(" ** wrong new pw ** ");
        }
        if (zadHeslo != heslo) {
            lcd.setCursor(0,0);
            lcd.print(" ** wrong old pw ** ");
        }
        delay (2000);
        nastaveniProZmenuHesla = 0;
    }
}
}
}

```

```

    doubleClickCounter = 0;
}

// funkce co inicializuje hlavní menu
void ukazMenu()
{
    lcd.clear(); // vyčistí displej
    lcd.setCursor(0,0); // nastaví kurzor do polohy (0,0)
    lcd.print("* aktivace alarmu "); // zobrazí na daném místě text
    lcd.setCursor(0,1); // nastaví kurzor do polohy (0,1)
    lcd.print("# zmena hesla "); // zobrazí na daném místě text
    lcd.setCursor(0,2); // nastaví kurzor do polohy (0,2)
    lcd.print("teplota="); // zobrazí na daném místě text
    lcd.setCursor(9,2); // nastaví kurzor do polohy (9,2)
    lcd.print(teplota); // na daném místě zobrazí teplotu
    lcd.print(" C "); // pokračuje v textu za teplotou
    lcd.setCursor(0,3); // nastaví kurzor do polohy (0,3)
    lcd.print("vlhkost="); // zobrazí na daném místě text
    lcd.setCursor(9,3); // nastaví kurzor do polohy (9,3)
    lcd.print(vlhkost); // na daném místě zobrazí vlhkost
    lcd.print(" % "); // pokračuje v textu za vlhkostí
    modeMenu = true; // umožní přechod do hlavního menu
}

// funkce - odpočítává jak dlouho bude svítit venkovní světlo a posílá informaci na Blynk
void myTimerEvent()
{
    if ( odpocet > 0 ) {
        odpocet = odpocet - 1;
        Blynk.virtualWrite(V7, odpocet);
    }
    else {
        Blynk.virtualWrite(V7, " ");
    }
}
}

```