/\*

Kočica Filip, ME 3

16.5.2015

Program pro řízení zavlažovacího systému.

Podrobný popis funkce programu:

Na začátku programu se vypíší na 5 sekund základní informace o systému. Na displayi (20\*4) se lze pohybovat kurzorem ve tvaru šipky pomocí tlačítek nahoru, dolu, doleva a doprava na membránové klávesnici. Výběr a uložení dat se provedete tlačítkem Enter a výstup z funkce tlačítkem Escape, nastavuje se pomocí čísel 0-9. Tlačítka ‚F2‘, ‚#‘ a ‚\*‘ se nevyužívají v žádných částech programu. Při zmáčknutí ‚F1‘ kdekoli v programu se systém zablokuje. Pokud je u kteréhokoli vstupu/výstupu plný čtverec, znamená to, že je sepnutý (log. 1), pokud je u něj prázdný čtverec, znamená to, že je vypnutý (log. 0). Po každém nastavení je nutno uložit data do EEPROM / ORC stisknutím tlačítka Enter!,

Na prvním řádku LCD se nachází dny Pondělí až Neděle, kde můžete ve funkci „den“ nastavit který den, v jakou hodinu, minutu a na jak dlouho budou sepnuty okruhy (výstupy) 1, 2 a buď zapnout, nebo vypnout funkci současný běh výstupů. Také se tam zobrazuje aktuální čas pro usnadnění nastavování. Při zmáčknutí ‚F1‘ kdekoli v programu se systém zablokuje!

Na druhém řádku nastavujeme vlhkost (0-100%), hladinu (0-100%), hysterezi (0-10%), limit displaye (0-60 Minut) a režim (Normal – Vlhko – Sucho). Nastavení vlhkosti a hladiny bude porovnáváno s hodnotami ze vstupů (při porovnávání je nutno vzít v potaz i hysterezi) a na základě toho se bude program dále chovat, pokud NV>MV (nastavená vlhkost bude větší než vlhkost naměřená na elektrodách) a zároveň NH<MH (nastavená hladina bude menší než hladina snímaná ultrazvukovým senzorem), bude možno výstupy sepnout, v opačném případě pokud NV<MV nebo NH>MH výstupy nebude možno sepnout a pokud už sepnuty byly, neprodleně se rozepnou. Nastavení hystereze je ošetření proti náhlému vypínání a spínaní relé. Limit displaye znamená, po jaké době se vypne podsvícení displaye, opětovné zapnutí podsvícení displaye je možno provést pomocí libovolného tlačítka. Výběr režimu usnadňuje práci se systémem, aby se nemusel přenastavovat celý při náhlé změně počasí, ale stačilo jen přepnout režim (Normal – Vlhko – Sucho).

Na třetím řádku je zobrazena MV (měřená vlhkost), MH (měřená hladina) a P (porucha). Vstupní hodnoty se čtou pouze minutu před sepnutím výstupů, při stlačení tlačítka a při běhu výstupů, kvůli minimálnímu opotřebení elektrod a snímače. Pokud se na analogovém vstupu (A10) objeví více než 25% Ucc, vrátí funkce „prectiPort“ hodnotu 0 a sepne se hlášení poruchy, to se projeví pískáním piezo bzučáku, blikáním podsvícení displaye a nemožností manipulace se systémem. Při poruše nelze sepnout žádný z výstupů. Poruchu lze „pozastavit“ tím, že stlačíte jakékoli tlačítko na klávesnici, vypne se piezo bzučák, blikání podsvícení displaye a bude možný přístup ke všem funkcím programu, nebude ovšem možno v žádném případě sepnout výstupy a na displayi se bude pořád zobrazovat sepnutá porucha. Při výběru na třetím řádku se spustí funkce pro kalibraci vlhkosti a hladiny. Dolní mez je konstantní a kalibruje se pouze horní mez a pak se hodnota přepočítává podle tohoto rozsahu pomocí funkce map (vstup, 0, kalibrace, 0, 100); na rozsah 0 – 100%, který potom vidíme na displayi.

Na čtvrtém řádku jsou zobrazeny výstupy a aktuální den + čas. Výstupy lze simulovat výběrem v menu sepnutím požadovaného výstupu a vrácením se do hlavní nabídky (Ale pozor! Tato simulace není limitována podmínkami nastavených a měřených parametrů a výstupy se sami nikdy nevypnou (pouze při aktivní poruše, nebo blokaci systému)!). Taktéž nastavení obvodu relného času lze nastavit výběrem v menu. Nastavování probíhá stejně jako u nastavování dnů Po-Ne Po.

\*/

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// Directiva include pro zahrnutí knihoven do programu

#include <EEPROM.h> //knihovna pro zápis a čtení z/do EEPROM

#include <Wire.h> //knihvna pro práci s čtením/zápisem z/do ORC

#include "Wire.h" //knihvna pro práci s čtením/zápisem z/do ORC

#include <Keypad.h> //knihovna pro komunikaci s membránovou klávesnicí 5x4

#include <LiquidCrystal\_I2C.h> //práce s displayem přes i2c

#include <SPI.h> //sériové periferní rozhraní, používá se pro komunikaci

#include <Ethernet.h> //ethernetové rozhraní

#define ORC\_I2C\_ADRESA 0x68 //definice preprocesorových maker

#define NULL 0

// Piny použité pro připojení LCD displaye (20x4, přes i2c)

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

typedef unsigned bez\_znamenka; //bez\_znamenka je nyni datovym typem, zastupujici unsigned, neboli beznaménkový

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// Znaky

byte ctverec1[8] = { //plný čtverec

B00000,

B11111,

B11111,

B11111,

B11111,

B11111,

B11111,

B00000,

};

byte ctverec2[8] = { //prázdný čtverec

B00000,

B11111,

B10001,

B10001,

B10001,

B10001,

B11111,

B00000,

};

byte kurzor [8] = { //kurzor

B00000,

B00100,

B11110,

B11111,

B11110,

B00100,

B00000,

B00000

};

byte objekt [8] = { // ve funkci nastavcas problikava s nastavovanym cislem

B00000,

B00000,

B00000,

B00000,

B00000,

B11111,

B11111,

B11111

};

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

const byte radky = 5; // 5 řádků

const byte sloupce = 4; // 4 sloupce

//více rozměrné pole

char klavesnice[radky][sloupce] = {

{

'A', 'B', '#', '\*'

}

,

{

'1', '2', '3', '^'

}

,

{

'4', '5', '6', 'v'

}

,

{

'7', '8', '9', 'C'

}

,

{

'<', '0', '>', 'E'

}

};

byte pinyRadku[radky] = {

46, A0, A1, A2, A3

}; //čísla pinů s řadkem 1 2 3 4 5

byte pinySloupcu[sloupce] = {

A7, A6, A5, A4

}; //čísla pinu se sloupcem 1 2 3 4

//inicializuje objekt klávesnice s názvem stisknutiTlacitka

Keypad stisknutiTlacitka = Keypad(makeKeymap(klavesnice), pinyRadku, pinySloupcu, radky, sloupce);

char posledniZmackle;

bool pusteniTlacitka;

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // mac adresa ethernet shieldu - NEMĚNIT

IPAddress ip(192, 168, 10, 145); // sem zapsat volnou IP adresu ve vnitřní síti a změnit ji v celém programu

EthernetServer server(80);

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// Zadeklarování prototypů fcí a definice parametrů.

byte DECnaBCD(byte hodnota); //fce pro převod decimální hodnoty do kodu BCD pro obvod reálného času -

byte BCDnaDEC(byte hodnota); //- a z BCD na decimální hodnotu

void cteniORC(byte \*sekundy, byte \*minuty, byte \*hodiny, byte \*denVTydnu, byte \*denVMesici); //procedura s parametry pro čtení z obvodu reálného času

void zapisORC(byte sekundy, byte minuty, byte hodiny, byte denVTydnu, byte denVMesici); //procedura s parametry pro uložení do obvodu reálného času

void zpozdeni(int doba); //procedura má za úkol zpozdit program na takovou dobu, jaký se jí dá parametr v ms(milisekundách)

void problikavani(); //procedura zajišťující problikávání dvojtečky a šipky

void zkontrolujEEPROM(); //procedura, která přepíše všechna místa v EEPROM,kde je 255 na 0.

void privitani(int doba\_trvani); //na začátku programu vypíše info o prg a IP adr

void nastaveniNaEthernetu(); //při uložení na ethernetu se zobrazí na displayi

void blokaceSystemu(); //při zmáčknutí F1 se systém zablokuje dokud se znovu nezmáčkne F1

bool nactiPort(byte port); //z portu 'port' předávaného jako parametr načte hodnotu, pokud je > 25% z 5V, vrátí 0, jinak vrátí 1.

int pocetZnaku(String &retezec); //zjisti pocet znaku z retezce tridy String, zadaného jako parametr a vrátí počet.

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// Globální proměnné

short x, y, i; // x,y = kurzor, i = se používá ve všech cyklech for

byte sekundy, minuty, hodiny, denVTydnu, denVMesici; // globální proměnné pro obvod reálného času

bool nastavvystupy1 = false, nastavvystupy2 = false, vypisMozny = true, spustVystupy = true, soucasnyBeh = false, pozastavPoruchu = false, porucha = false, refresh[2] {false, false};

bez\_znamenka long cekej, cekej01; //časovače

int identifik = NULL, kteryVystup = 2, nedele2 = NULL, oba\_vystupy = NULL, druhy\_vystup = NULL, soucasnyBehVystupy = NULL, AN1, AN2, Binarnivstup1, zhasnuti = NULL, Menu = NULL, rezim = NULL, rezimEEPROM = NULL, minuty\_osetreniVystupu = NULL;

// binarni vstup na desce (BV1) , analogové vstupy 1,2 a binární vstup 1, kam se ukládají přepočítané hodnoty ze vstupů

double cas\_vystupu = NULL, cas2 = NULL;

int nastav\_vystup[] = {0, 0, 0, 0, 0, 0, 0}, rele[] = { 47, 49, 48, 53 }; // static array (pole se statickou vazbou)

class Zapouzdreni\_dat

{

private:

int limitdisplaye,

nastvlhkosti,

nasthladiny,

nasthystereze,

kalibraceVlhkost,

kalibraceHladina;

//struktura pole

typedef struct {

int hodiny1;

int hodiny2;

int hodinycelk;

int minuty1; // PRIVATE: (ZAPOUZDŘENÉ PROMĚNNÉ)

int minuty2; // LZE S NIMI OPEROVAT POUZE POMOCÍ ČLENSKÝCH FCÍ / METOD

int minutycelk;

int min1;

int min10;

int min1celk;

int min2;

int min20;

int min2celk;

}

nastaveniDny;

//strukturové pole

nastaveniDny dny[6];

int nastavORC[10] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

public: // METODY / ČLENSKÉ FCE TŘÍDY

Zapouzdreni\_dat(); //KONSTRUKTOR TŘÍDY

int nastaveniParametru(int limitdisplaye, int nasthladiny, int nastvlhkosti, int nasthystereze); //fce pro nastavení hladiny,vlhkosti,limitu displaye

int cas(int limitdisplaye); //fce pro nastavení Po-Ne (hodiny,minuty,min1,min2)

int nastavcas(int limitdisplaye); //fce pro nastavení obvodu reálného času

int nastavvystupy(int limitdisplaye); //fce pro nastavení výstupu (LOW/HIGH)

int vyberRezim(); //fce pro vyber rezimu 1 - 2 - 3

int hlaseniPoruchy(bool nastavvystupy1, bool nastavvystupy2, int Binarnivstup1, int i); //fce pro hlášení poruchy (binární vstup == NULL)

void nactiVstupy(int analogvstup1, int analogvstup2, int cteniVstupu1, int cteniVstupu2); //procedura na načtení hodnot ze vstupů

void kalibrace(); //nakalibruje dolní a horní mez při přepočtu analogových vstupů

void limit(); //procedura hlídající vypnutí podsvícení displaye

void prectiEEPROM(); //procedura, která načte data z EEPROM do proměnných

void zapisEEPROM(); //procedura, která zapíše data do EEPROM

void prace\_s\_vystupy(); //procedura, která zajišťuje správný běh výstupů (kdy se sepnou,na jak dlouho,kdy se rozepnou)

void zobrazNaDisplayi(); //procedura má za úlohu vykreslit celé menu, se všemi hodnotami

void stlaceniTlacitkaLoop(); //procedura, která zajišťuje správnou fci tlačítek

void ethernet(); //tato procedura zajištuje celou sitovou komunikaci

~Zapouzdreni\_dat(); //DESTRUKTOR TŘÍDY

};

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void setup()

{

Serial.begin(9600); // metoda begin pro objekt Serial,definuje 9600 baudů

Wire.begin(); // metoda begin pro objekt Wire (zápis a čtení z ORČ)

lcd.begin(20, 4); // metoda begin pro objekt lcd, o velikosti 20x4

Ethernet.begin(mac, ip); // metoda begin pro objekt Ethernet, inicializuje síť

server.begin(); // metoda begin pro objekt server, SPI

lcd.backlight(); // podsvícení displaye

/\*

for (int i = NULL; i < 1000; i++) // při mazání paměti se musí znovu nastavit obvod reálného času

EEPROM.write(i, 0); // Vycisti EEPROM - 1kB

\*/

//zapisORC(0,0,0,0,0); // formát (vteřiny,minuty,hodiny,denVTydnu,denVMesici) - nastavení Obvodu Reálného Času, při deklaraci smazat poznámku (//) mimochodem čas lze nastavit na displayi vpravo dole

pinMode(A10, OUTPUT); // napajeni Binarnich vstupu

digitalWrite(A10, HIGH); // A11 = binarni vstup, definuje ho jako INPUT a napájení pro něj A10

pinMode(A11, INPUT);

pinMode(A15, OUTPUT);

digitalWrite(A15, HIGH);

for (i = NULL; i < 4; i++) {

pinMode(rele[i], OUTPUT);

if (i == 3) digitalWrite(rele[i], LOW); // nastaví relátka jako výstupy a 4. relé definuje jako LOW

}

lcd.createChar(1, ctverec1);

lcd.createChar(2, ctverec2);

lcd.createChar(3, kurzor); // vytvoření objektů znaků

lcd.createChar(4, objekt);

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void loop()

{

Zapouzdreni\_dat \* den = new Zapouzdreni\_dat; // VYTVOŘENÍ DYNAMICKÉ INSTANCE TŘÍDY NEBOLI OBJEKTU POMOCÍ OPERÁTORU NEW, KVŮLI NEPŘÍMENÉ ADRESACI.

privitani(5000); // vypíše info o PRG a IP adr, pak pocka 3s bez možnosti přeskočení

den->nactiVstupy(A14, A12, A15, A13); // funkce která načte hodnoty z analog. a binar. vstupů a uloží je do globálních proměnných

zkontrolujEEPROM(); // procedura zapíše nulu na místo,kde je v EEPROM 255

for (;;) { // nekonečná smyčka (stejně jako void loop(), aby se nevytvářely pořád statické proměnné a objekt den

// -> = nepřímá adresace

den->prectiEEPROM(); // načte data z EEPROM

den->prace\_s\_vystupy(); // kontroluje (zapíná/vypíná) výstupy

den->zobrazNaDisplayi(); // Zobrazí na Displayi

problikavani(); // zajištuje blikani kurzoru a dvojtečky

den->limit(); // vypnutí podsvícení displaye po uplynutí nastaveného času bez stlačení tlačítka

Binarnivstup1 = nactiPort(A11); // fce vrátí 1 pokud je na portu méně než 25% z 5V a 0 pokud je více nebo rovno 25%.

if ((!Binarnivstup1) && (!pozastavPoruchu))

den->hlaseniPoruchy(nastavvystupy1, nastavvystupy2, Binarnivstup1, i); // zavolá fci hlášení poruchy, když (binární vstup 1 == NULL) a zároveň pozastav poruchu je neaktivni

den->stlaceniTlacitkaLoop(); // Snímá stlačení tlačítka a pro přísnušlé tlačítko vykoná příslušné operace

den->ethernet(); // zajišťuje síťovou komunikaci

}

delete[] den; // uvolnění paměti

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// FUNKCE

int Zapouzdreni\_dat::nastaveniParametru(int limitdisplaye, int nasthladiny, int nastvlhkosti, int nasthystereze)

{

refresh[2] = true;

cas2 = millis() / 1000;

if (limitdisplaye != 0) lcd.backlight();

int a = NULL;

porucha = pozastavPoruchu;

for (;;) {

if (porucha) pozastavPoruchu = true;

Binarnivstup1 = nactiPort(A11);

if ((!Binarnivstup1) && (!pozastavPoruchu)) break;

ethernet();

if (refresh[2]) {

nasthladiny = EEPROM.read(0 + rezimEEPROM);

nastvlhkosti = EEPROM.read(1 + rezimEEPROM);

nasthystereze = EEPROM.read(5 + rezimEEPROM);

limitdisplaye = EEPROM.read(2 + rezimEEPROM);

refresh[2] = false;

}

if ((millis() / 1000) > (cas2 + (60 \* limitdisplaye))) lcd.noBacklight();

lcd.setCursor(0, a);

lcd.write(byte(3));

lcd.setCursor(1, 0);

lcd.write("NAST. VLHKOSTI=");

if ((nastvlhkosti >= 10) && (nastvlhkosti < 100)) {

lcd.setCursor(16, 0);

lcd.print(" ");

lcd.setCursor(17, 0);

lcd.print(nastvlhkosti);

}

else if (nastvlhkosti < 10) {

lcd.setCursor(16, 0);

lcd.write(" ");

lcd.setCursor(18, 0);

lcd.print(nastvlhkosti);

}

else if (nastvlhkosti == 100) {

lcd.setCursor(16, 0);

lcd.print(nastvlhkosti);

}

lcd.setCursor(1, 1);

lcd.write("NAST. HLADINY =");

if ((nasthladiny >= 10) && (nasthladiny < 100)) {

lcd.setCursor(16, 1);

lcd.print(" ");

lcd.setCursor(17, 1);

lcd.print(nasthladiny);

}

else if (nasthladiny < 10) {

lcd.setCursor(16, 1);

lcd.write(" ");

lcd.setCursor(18, 1);

lcd.print(nasthladiny);

}

else if (nasthladiny == 100) {

lcd.setCursor(16, 1);

lcd.print(nasthladiny);

}

lcd.setCursor(1, 3);

lcd.write("LIMIT DISPLAYE=");

if ((limitdisplaye >= 10) && (limitdisplaye < 100)) {

lcd.setCursor(16, 3);

lcd.print(" ");

lcd.setCursor(17, 3);

lcd.print(limitdisplaye);

lcd.print("m");

}

else if (limitdisplaye < 10) {

lcd.setCursor(17, 3);

lcd.write(" ");

lcd.setCursor(18, 3);

lcd.print(limitdisplaye);

lcd.print("m");

}

lcd.setCursor(1, 2);

lcd.write("NAST. HYSTEREZ=");

if ((nasthystereze >= 10) && (nasthystereze < 100)) {

lcd.setCursor(16, 2);

lcd.print(" ");

lcd.setCursor(17, 2);

lcd.print(nasthystereze);

}

else if (nasthystereze < 10) {

lcd.setCursor(17, 2);

lcd.write(" ");

lcd.setCursor(18, 2);

lcd.print(nasthystereze);

}

else if (nasthystereze == 100) {

lcd.setCursor(16, 2);

lcd.print(nasthystereze);

}

for (int procenta = NULL; procenta < 3; procenta++) {

lcd.setCursor(19, procenta);

lcd.print("%");

}

char tlacitko = stisknutiTlacitka.getKey();

KeyState stav = stisknutiTlacitka.getState();

if (stav == PRESSED && tlacitko != NO\_KEY) {

if (limitdisplaye != 0) lcd.backlight();

cas2 = millis() / 1000;

posledniZmackle = tlacitko;

pusteniTlacitka = false;

if (tlacitko == 'A') {

blokaceSystemu(); // zablokuje systém dokud se nezmáčkne F1

}

if (posledniZmackle == 'C') //konec (ESC)

{

lcd.clear();

if (limitdisplaye != 0) lcd.backlight();

cas2 = millis() / 1000;

if (porucha) pozastavPoruchu = true;

break;

}

if (posledniZmackle == 'E')

{

lcd.clear();

if (porucha) pozastavPoruchu = true;

EEPROM.write(0 + rezimEEPROM, nasthladiny);

EEPROM.write(1 + rezimEEPROM, nastvlhkosti);

EEPROM.write(2 + rezimEEPROM, limitdisplaye); // zápis proměnných do eeprom!

EEPROM.write(5 + rezimEEPROM, nasthystereze);

lcd.setCursor(0, 0);

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

lcd.setCursor(7, 1);

lcd.print("ULOZENO");

lcd.setCursor(9, 2);

lcd.print("Do");

lcd.setCursor(3, 3);

lcd.print("Pameti EEPROM!");

zpozdeni(2000);

lcd.clear();

if (limitdisplaye != 0) lcd.backlight();

cas2 = millis() / 1000;

}

if (posledniZmackle == '^')

{

if (limitdisplaye != 0) lcd.backlight();

cas2 = millis() / 1000;

if (a == NULL) {

a = 3;

zpozdeni(5);

lcd.setCursor(0, 0);

lcd.write(" ");

continue;

}

if (a == 1) {

a = NULL;

zpozdeni(5);

lcd.setCursor(0, 1);

lcd.write(" ");

continue;

}

if (a == 2) {

a = 1;

lcd.setCursor(0, 2);

lcd.print(" ");

continue;

}

if (a == 3) {

a = 2;

lcd.setCursor(0, 3);

lcd.print(" ");

continue;

}

}

if (posledniZmackle == 'v')

{

if (limitdisplaye != 0) lcd.backlight();

cas2 = millis() / 1000;

if (a == NULL) {

a = 1;

zpozdeni(5);

lcd.setCursor(0, 0);

lcd.write(" ");

continue;

}

if (a == 1) {

a = 2;

zpozdeni(5);

lcd.setCursor(0, 1);

lcd.write(" ");

continue;

}

if (a == 2) {

a = 3;

zpozdeni(5);

lcd.setCursor(0, 2);

lcd.write(" ");

continue;

}

if (a == 3) {

a = NULL;

lcd.setCursor(0, 3);

lcd.print(" ");

continue;

}

}

if (posledniZmackle == '>')

{

if (a == NULL) {

if (nastvlhkosti <= 100) {

nastvlhkosti = nastvlhkosti + 1;

if (nastvlhkosti == 101) nastvlhkosti = 0;

}

}

if (a == 1) {

if (nasthladiny <= 100) {

nasthladiny = nasthladiny + 1;

if (nasthladiny == 101) nasthladiny = 0;

}

}

if (a == 2) {

if (nasthystereze <= 10) {

nasthystereze = nasthystereze + 1;

if (nasthystereze == 11) nasthystereze = 0;

}

}

if (a == 3) {

if (limitdisplaye == 0) limitdisplaye = 1;

else if (limitdisplaye == 1) limitdisplaye = 2;

else if (limitdisplaye == 2) limitdisplaye = 3;

else if (limitdisplaye == 3) limitdisplaye = 5;

else if (limitdisplaye == 5) limitdisplaye = 10;

else if (limitdisplaye == 10) limitdisplaye = 15;

else if ((limitdisplaye >= 15) && (limitdisplaye <= 45)) limitdisplaye += 15;

else if (limitdisplaye == 60) limitdisplaye = 0;

}

}

if (posledniZmackle == '<')

{

if (a == NULL) {

if (nastvlhkosti >= 0) {

nastvlhkosti = nastvlhkosti - 1;

if (nastvlhkosti == -1) nastvlhkosti = 100;

}

}

if (a == 1) {

if (nasthladiny >= 0) {

nasthladiny = nasthladiny - 1;

if (nasthladiny == -1) nasthladiny = 100;

}

}

if (a == 2) {

if (nasthystereze >= 0) {

nasthystereze = nasthystereze - 1;

if (nasthystereze == -1) nasthystereze = 10;

}

}

if (a == 3) {

if (limitdisplaye == 1) limitdisplaye = 0;

else if (limitdisplaye == 2) limitdisplaye = 1;

else if (limitdisplaye == 3) limitdisplaye = 2;

else if (limitdisplaye == 5) limitdisplaye = 3;

else if (limitdisplaye == 10) limitdisplaye = 5;

else if (limitdisplaye == 15) limitdisplaye = 10;

else if ((limitdisplaye >= 30) && (limitdisplaye <= 60)) limitdisplaye -= 15;

else if (limitdisplaye == 0) limitdisplaye = 60;

}

}

}

else if (stav == RELEASED && !pusteniTlacitka) {

pusteniTlacitka = true;

}

else if (stav == HOLD) {

if (limitdisplaye != 0) lcd.backlight();

cas2 = millis() / 1000;

if (posledniZmackle == '>')

{

if (a == NULL) {

if (nastvlhkosti <= 100) {

nastvlhkosti = nastvlhkosti + 1;

if (nastvlhkosti == 101) nastvlhkosti = 0;

}

}

if (a == 1) {

if (nasthladiny <= 100) {

nasthladiny = nasthladiny + 1;

if (nasthladiny == 101) nasthladiny = 0;

}

}

if (a == 2) {

if (nasthystereze <= 10) {

nasthystereze = nasthystereze + 1;

if (nasthystereze == 11) nasthystereze = 0;

}

}

if (a == 3) {

if (limitdisplaye == 0) limitdisplaye = 1;

else if (limitdisplaye == 1) limitdisplaye = 2;

else if (limitdisplaye == 2) limitdisplaye = 3;

else if (limitdisplaye == 3) limitdisplaye = 5;

else if (limitdisplaye == 5) limitdisplaye = 10;

else if (limitdisplaye == 10) limitdisplaye = 15;

else if ((limitdisplaye >= 15) && (limitdisplaye <= 45)) limitdisplaye += 15;

else if (limitdisplaye == 60) limitdisplaye = 0;

}

}

if (posledniZmackle == '<')

{

if (a == NULL) {

if (nastvlhkosti >= 0) {

nastvlhkosti = nastvlhkosti - 1;

if (nastvlhkosti == -1) nastvlhkosti = 100;

}

}

if (a == 1) {

if (nasthladiny >= 0) {

nasthladiny = nasthladiny - 1;

if (nasthladiny == -1) nasthladiny = 100;

}

}

if (a == 2) {

if (nasthystereze >= 0) {

nasthystereze = nasthystereze - 1;

if (nasthystereze == -1) nasthystereze = 10;

}

}

if (a == 3) {

if (limitdisplaye == 1) limitdisplaye = 0;

else if (limitdisplaye == 2) limitdisplaye = 1;

else if (limitdisplaye == 3) limitdisplaye = 2;

else if (limitdisplaye == 5) limitdisplaye = 3;

else if (limitdisplaye == 10) limitdisplaye = 5;

else if (limitdisplaye == 15) limitdisplaye = 10;

else if ((limitdisplaye >= 30) && (limitdisplaye <= 60)) limitdisplaye -= 15;

else if (limitdisplaye == 0) limitdisplaye = 60;

}

}

}

}

lcd.clear();

return 0;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// Proměnná "i" se nastaví podle dne.

int Zapouzdreni\_dat::cas(int limitdisplaye)

{ //funkce cas

if (limitdisplaye != 0) lcd.backlight();

int promenna = 1;

lcd.setCursor(6, 2);

lcd.print(dny[i].min1);

lcd.print(dny[i].min10);

lcd.setCursor(12, 2);

lcd.print(dny[i].min2);

if (i == 6) lcd.print(nedele2);

else lcd.print(dny[i].min20);

for (;;) {

ethernet();

if (refresh[1]) {

prectiEEPROM();

refresh[1] = false;

lcd.setCursor(0, 0);

lcd.print(dny[i].hodiny1);

lcd.setCursor(0, 1);

lcd.print(dny[i].hodiny2);

lcd.setCursor(0, 3);

lcd.print(dny[i].minuty1);

lcd.setCursor(0, 4);

lcd.print(dny[i].minuty2);

lcd.setCursor(6, 2);

lcd.print(dny[i].min1);

lcd.setCursor(7, 2);

lcd.print(dny[i].min10);

lcd.setCursor(12, 2);

lcd.print(dny[i].min2);

lcd.setCursor(13, 2);

if (i == 6) lcd.print(nedele2);

else lcd.print(dny[i].min20);

}

Binarnivstup1 = nactiPort(A11);

if ((!Binarnivstup1) && (!pozastavPoruchu)) break;

// Nacteni hodin a minut

cteniORC(&sekundy, &minuty, &hodiny, &denVTydnu, &denVMesici);

if ((millis() / 1000) > (cas2 + (60 \* limitdisplaye))) lcd.noBacklight();

else if (limitdisplaye != 0) lcd.backlight();

lcd.setCursor(0, 1);

if (hodiny < 10) {

lcd.print("0");

}

lcd.print(hodiny, DEC); //vypsani casu

lcd.print(":");

if (minuty < 10) {

lcd.print("0");

}

lcd.print(minuty, DEC);

lcd.setCursor(5, 1);

switch (denVTydnu) {

case 1:

lcd.print(" Pondeli");

break;

case 2:

lcd.print(" Utery");

break;

case 3:

lcd.print(" Streda");

break;

case 4:

lcd.print(" Ctvrtek");

break;

case 5:

lcd.print(" Patek");

break;

case 6:

lcd.print(" Sobota");

break;

case 7:

lcd.print(" Nedele");

break;

}

lcd.setCursor(2, 0);

lcd.print(":");

lcd.setCursor(1, 3);

lcd.print("Soucasny beh - ");

if (soucasnyBeh) lcd.print("Zap");

else lcd.print("Vyp");

if (promenna == 1) {

lcd.setCursor(1, 0);

lcd.print(dny[i].hodiny2);

lcd.setCursor(3, 0);

lcd.print(dny[i].minuty1);

lcd.setCursor(4, 0);

lcd.print(dny[i].minuty2);

lcd.setCursor(13, 2);

if (i == 6) lcd.print(nedele2);

else lcd.print(dny[i].min20);

}

if (promenna == 2) {

lcd.setCursor(0, 0);

lcd.print(dny[i].hodiny1);

lcd.setCursor(3, 0);

lcd.print(dny[i].minuty1);

lcd.setCursor(4, 0);

lcd.print(dny[i].minuty2);

}

if (promenna == 3) {

lcd.setCursor(0, 0);

lcd.print(dny[i].hodiny1);

lcd.setCursor(1, 0);

lcd.print(dny[i].hodiny2);

lcd.setCursor(4, 0);

lcd.print(dny[i].minuty2);

}

if (promenna == 4) {

lcd.setCursor(0, 0);

lcd.print(dny[i].hodiny1);

lcd.setCursor(1, 0);

lcd.print(dny[i].hodiny2);

lcd.setCursor(3, 0);

lcd.print(dny[i].minuty1);

lcd.setCursor(6, 2);

lcd.print(dny[i].min1);

}

if (promenna == 5) {

lcd.setCursor(3, 0);

lcd.print(dny[i].minuty1);

lcd.setCursor(7, 2);

lcd.print(dny[i].min10);

}

if (promenna == 6) {

lcd.setCursor(12, 2);

lcd.print(dny[i].min2);

lcd.setCursor(6, 2);

lcd.print(dny[i].min1);

}

if (promenna == 7) {

lcd.setCursor(13, 2);

if (i == 6) lcd.print(nedele2);

else lcd.print(dny[i].min20);

lcd.setCursor(7, 2);

lcd.print(dny[i].min10);

}

if (promenna == 8) {

lcd.setCursor(12, 2);

lcd.print(dny[i].min2);

lcd.setCursor(1, 0);

lcd.print(dny[i].hodiny2);

}

if ((promenna == 6) || (promenna == 1)) {

lcd.setCursor(11, 2);

lcd.write(" ");

}

if ((promenna == 7) || (promenna == 4)) {

lcd.setCursor(5, 2);

lcd.write(" ");

}

if ((promenna == 1) || (promenna == 2)) {

lcd.setCursor(6, 0);

lcd.write("Nast. Hodiny ");

lcd.setCursor(0, 3);

lcd.write(" ");

}

if ((promenna == 3) || (promenna == 4)) {

lcd.setCursor(6, 0);

lcd.write("Nast. Minuty ");

}

if ((promenna == 5) || (promenna == 6)) {

lcd.setCursor(6, 0);

lcd.write("Nast. 1.Okruh ");

lcd.setCursor(5, 2);

lcd.write(byte(3));

}

if ((promenna == 7) || (promenna == 8)) {

lcd.setCursor(6, 0);

lcd.write("Nast. 2.Okruh ");

lcd.setCursor(11, 2);

lcd.write(byte(3));

lcd.setCursor(0, 3);

lcd.write(" ");

}

if (promenna == 9) {

lcd.setCursor(6, 0);

lcd.write("Nast. Souc.Beh");

lcd.setCursor(0, 3);

lcd.write(byte(3));

lcd.setCursor(0, 0);

lcd.print(dny[i].hodiny1);

lcd.setCursor(11, 2);

lcd.print(" ");

lcd.setCursor(13, 2);

if (i == 6) lcd.print(nedele2);

else lcd.print(dny[i].min20);

}

if ((millis() > cekej + 500) && (millis() < cekej + 1000)) {

switch (promenna) {

case 1:

lcd.setCursor(0, 0);

lcd.write(byte(4));

break;

case 2:

lcd.setCursor(1, 0);

lcd.write(byte(4));

break;

case 3:

lcd.setCursor(3, 0);

lcd.write(byte(4));

break;

case 4:

lcd.setCursor(4, 0);

lcd.write(byte(4));

break;

case 5:

lcd.setCursor(6, 2);

lcd.write(byte(4));

break;

case 6:

lcd.setCursor(7, 2);

lcd.write(byte(4));

break;

case 7:

lcd.setCursor(12, 2);

lcd.write(byte(4));

break;

case 8:

lcd.setCursor(13, 2);

lcd.write(byte(4));

break;

}

}

if ((millis() > cekej + 1) && (millis() < cekej + 500)) {

switch (promenna) {

case 1:

lcd.setCursor(0, 0);

lcd.print(dny[i].hodiny1);

break;

case 2:

lcd.setCursor(1, 0);

lcd.print(dny[i].hodiny2);

break;

case 3:

lcd.setCursor(3, 0);

lcd.print(dny[i].minuty1);

break;

case 4:

lcd.setCursor(4, 0);

lcd.print(dny[i].minuty2);

break;

case 5:

lcd.setCursor(6, 2);

lcd.print(dny[i].min1);

break;

case 6:

lcd.setCursor(7, 2);

lcd.print(dny[i].min10);

break;

case 7:

lcd.setCursor(12, 2);

lcd.print(dny[i].min2);

break;

case 8:

lcd.setCursor(13, 2);

if (i == 6) lcd.print(nedele2);

else lcd.print(dny[i].min20);

break;

}

}

if (millis() > cekej + 1000) cekej = millis();

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko) {

cas2 = millis() / 1000;

if (limitdisplaye != 0) lcd.backlight();

if (tlacitko == 'A') {

blokaceSystemu(); // zablokuje systém dokud se nezmáčkne F1

}

for (int ch = 0; ch <= 9; ch++) {

if ((int(tlacitko) - 48) == ch) {

switch (promenna) {

case 1:

if ((dny[i].hodiny2 < 4) && (ch < 3)) dny[i].hodiny1 = ch;

else if (ch < 2) dny[i].hodiny1 = ch;

break;

case 2:

if (dny[i].hodiny1 < 2) dny[i].hodiny2 = ch;

else if ((dny[i].hodiny1 == 2) && (ch < 4)) dny[i].hodiny2 = ch;

break;

case 3:

if (ch < 6) dny[i].minuty1 = ch;

break;

case 4:

dny[i].minuty2 = ch;

break;

case 5:

dny[i].min1 = ch;

break;

case 6:

dny[i].min10 = ch;

break;

case 7:

dny[i].min2 = ch;

break;

case 8:

if (i == 6) nedele2 = ch;

else dny[i].min20 = ch;

break;

}

}

}

if (tlacitko == '>')

{

if (promenna < 10) {

promenna++;

zpozdeni(10);

if (promenna == 5) {

lcd.setCursor(4, 0);

lcd.print(dny[i].minuty2);

}

if (promenna == 10) {

promenna = 1;

}

}

}

if (tlacitko == '<')

{

if (promenna >= 1) {

promenna--;

zpozdeni(10);

if (promenna == NULL) {

promenna = 9;

}

}

}

if (tlacitko == '^')

{

if (promenna == 9) {

if (soucasnyBeh == false) {

soucasnyBeh = true;

zpozdeni(10);

continue;

}

if (soucasnyBeh == true) {

soucasnyBeh = false;

zpozdeni(10);

continue;

}

break;

}

}

if (tlacitko == 'v')

{

if (promenna == 9) {

if (soucasnyBeh == false) {

soucasnyBeh = true;

zpozdeni(10);

continue;

}

if (soucasnyBeh == true) {

soucasnyBeh = false;

zpozdeni(10);

continue;

}

break;

}

}

if (tlacitko == 'C')

break;

if (tlacitko == 'E')

{

dny[i].hodinycelk = (dny[i].hodiny1 \* 10) + dny[i].hodiny2;

dny[i].minutycelk = (dny[i].minuty1 \* 10) + dny[i].minuty2;

dny[i].min1celk = ((dny[i].min1 \* 10) + dny[i].min10);

if (i == 6) dny[i].min2celk = ((dny[i].min2 \* 10) + nedele2);

else dny[i].min2celk = ((dny[i].min2 \* 10) + dny[i].min20);

zapisEEPROM();

lcd.clear();

lcd.setCursor(0, 0);

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

lcd.setCursor(7, 1);

lcd.print("ULOZENO");

lcd.setCursor(9, 2);

lcd.print("Do");

lcd.setCursor(3, 3);

lcd.print("Pameti EEPROM!");

zpozdeni(2000);

lcd.clear();

promenna = 1;

lcd.setCursor(6, 2);

lcd.print(dny[i].min1);

lcd.print(dny[i].min10);

lcd.setCursor(12, 2);

lcd.print(dny[i].min2);

if (i == 6) lcd.print(nedele2);

else lcd.print(dny[i].min20);

cas2 = millis() / 1000;

continue;

}

}

}

lcd.write(" ");

return 0;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

int Zapouzdreni\_dat::nastavcas(int limitdisplaye)

{

int promenna = 1;

// Nacteni hodin a minut

cteniORC(&sekundy, &minuty, &hodiny, &denVTydnu, &denVMesici);

nastavORC[7] = denVTydnu;

nastavORC[2] = minuty / 10;

nastavORC[8] = hodiny / 10;

nastavORC[4] = minuty % 10;

nastavORC[3] = hodiny % 10;

if (limitdisplaye != 0) lcd.backlight();

cekej = millis();

zpozdeni(100);

for (;;) {

ethernet();

Binarnivstup1 = nactiPort(A11);

if ((!Binarnivstup1) && (!pozastavPoruchu)) break;

if ((millis() / 1000) > (cas2 + (60 \* limitdisplaye))) lcd.noBacklight();

lcd.setCursor(0, 2);

if (hodiny < 10) {

lcd.print("0");

}

lcd.print(hodiny, DEC); //vypsani casu

lcd.print(":");

if (minuty < 10) {

lcd.print("0");

}

lcd.print(minuty, DEC);

lcd.setCursor(9, 2);

lcd.print("NAST.DNE");

lcd.setCursor(6, 3);

switch (nastavORC[7]) {

case 1:

lcd.print("Pondeli ");

break;

case 2:

lcd.print("Utery ");

break;

case 3:

lcd.print("Streda ");

break;

case 4:

lcd.print("Ctvrtek ");

break;

case 5:

lcd.print("Patek ");

break;

case 6:

lcd.print("Sobota ");

break;

case 7:

lcd.print("Nedele ");

break;

}

lcd.setCursor(2, 0);

lcd.write(":");

switch (promenna) {

case 1:

lcd.setCursor(1, 0);

lcd.print(nastavORC[3]);

lcd.setCursor(3, 0);

lcd.print(nastavORC[2]);

lcd.setCursor(4, 0);

lcd.print(nastavORC[4]);

lcd.setCursor(6, 0);

lcd.write("Nast. Hodiny");

lcd.setCursor(5, 3);

lcd.write(" ");

break;

case 2:

lcd.setCursor(0, 0);

lcd.print(nastavORC[8]);

lcd.setCursor(3, 0);

lcd.print(nastavORC[2]);

lcd.setCursor(4, 0);

lcd.print(nastavORC[4]);

lcd.setCursor(6, 0);

lcd.write("Nast. Hodiny");

lcd.setCursor(5, 3);

lcd.write(" ");

break;

case 3:

lcd.setCursor(0, 0);

lcd.print(nastavORC[8]);

lcd.setCursor(1, 0);

lcd.print(nastavORC[3]);

lcd.setCursor(4, 0);

lcd.print(nastavORC[4]);

lcd.setCursor(6, 0);

lcd.write("Nast. Minuty");

break;

case 4:

lcd.setCursor(0, 0);

lcd.print(nastavORC[8]);

lcd.setCursor(1, 0);

lcd.print(nastavORC[3]);

lcd.setCursor(3, 0);

lcd.print(nastavORC[2]);

lcd.setCursor(6, 0);

lcd.write("Nast. Minuty");

break;

case 5:

lcd.setCursor(0, 0);

lcd.print(nastavORC[8]);

lcd.setCursor(6, 0);

lcd.write("Nast. Dne ");

lcd.setCursor(5, 3);

lcd.write(byte(3));

break;

}

lcd.setCursor(0, 1);

lcd.write("Konec= Esc,Uloz= Ent");

nastavORC[5] = (nastavORC[8] \* 10) + nastavORC[3];

nastavORC[6] = (nastavORC[2] \* 10) + nastavORC[4];

if ((millis() > cekej + 500) && (millis() < cekej + 1000)) {

switch (promenna) {

case 1:

lcd.setCursor(0, 0);

lcd.write(byte(4));

break;

case 2:

lcd.setCursor(1, 0);

lcd.write(byte(4));

break;

case 3:

lcd.setCursor(3, 0);

lcd.write(byte(4));

break;

case 4:

lcd.setCursor(4, 0);

lcd.write(byte(4));

break;

}

}

if ((millis() > cekej + 1) && (millis() < cekej + 500)) {

switch (promenna) {

case 1:

lcd.setCursor(0, 0);

lcd.print(nastavORC[8]);

break;

case 2:

lcd.setCursor(1, 0);

lcd.print(nastavORC[3]);

break;

case 3:

lcd.setCursor(3, 0);

lcd.print(nastavORC[2]);

break;

case 4:

lcd.setCursor(4, 0);

lcd.print(nastavORC[4]);

break;

}

}

if (millis() > cekej + 1000) cekej = millis();

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko) {

cas2 = millis() / 1000;

if (limitdisplaye != 0) lcd.backlight();

if (tlacitko == 'A') {

blokaceSystemu(); // zablokuje systém dokud se nezmáčkne F1

}

if (tlacitko == '^')

{

if (promenna == 5) {

if (nastavORC[7] <= 6) {

nastavORC[7]++;

}

zpozdeni(20);

}

}

if (tlacitko == 'v')

{

if (promenna == 5) {

if (nastavORC[7] >= 2) {

nastavORC[7]--;

}

zpozdeni(20);

}

}

for (int ch = 0; ch <= 9; ch++) {

if ((int(tlacitko) - 48) == ch) {

switch (promenna) {

case 1:

if ((nastavORC[3] < 4) && (ch < 3)) nastavORC[8] = ch;

else if (ch < 2) nastavORC[8] = ch;

break;

case 2:

if (nastavORC[8] < 2) nastavORC[3] = ch;

else if ((nastavORC[8] == 2) && (ch < 4)) nastavORC[3] = ch;

break;

case 3:

if (ch < 6) nastavORC[2] = ch;

break;

case 4:

nastavORC[4] = ch;

break;

}

}

}

if (tlacitko == '>')

{

if (promenna < 6) {

promenna++;

if (promenna == 5) {

lcd.setCursor(4, 0);

lcd.print(nastavORC[4]);

}

zpozdeni(20);

if (promenna == 6) {

promenna = 1;

}

}

}

if (tlacitko == '<')

{

if (promenna >= 1) {

promenna--;

zpozdeni(20);

if (promenna == 4) {

lcd.setCursor(5, 3);

lcd.print(" ");

}

if (promenna == NULL) {

promenna = 5;

}

}

}

if (tlacitko == 'C')

break;

if (tlacitko == 'E')

{

zapisORC(00, nastavORC[6], nastavORC[5], nastavORC[7], 12);

lcd.clear();

lcd.setCursor(0, 0);

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

lcd.setCursor(7, 1);

lcd.print("ULOZENO");

lcd.setCursor(9, 2);

lcd.print("Do");

lcd.setCursor(0, 3);

lcd.print("Obvodu Realneho Casu");

zpozdeni(2000);

lcd.setCursor(0, 0);

lcd.clear();

promenna = 1;

continue;

}

}

}

lcd.write(" ");

return 0;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

int Zapouzdreni\_dat::nastavvystupy(int limitdisplaye)

{

int promenna = 1;

x = NULL;

y = 2;

for (;;) {

ethernet();

Binarnivstup1 = nactiPort(A11);

if ((!Binarnivstup1) && (!pozastavPoruchu)) break;

promenna = y;

lcd.setCursor(x, y);

lcd.write(byte(3));

zpozdeni(20);

if ((millis() / 1000) > (cas2 + (60 \* limitdisplaye))) lcd.noBacklight();

lcd.setCursor(0, 0);

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

lcd.setCursor(1, 1);

lcd.print("OUT1=");

if (((nastavvystupy1 == true) || ((soucasnyBehVystupy == 2) && ((kteryVystup == NULL) || (kteryVystup == 1))) || (soucasnyBehVystupy == 1) || (nastavvystupy2 == true)) || ((nastav\_vystup[0] == 1) || (nastav\_vystup[1] == 1) || (nastav\_vystup[2] == 1) || (nastav\_vystup[3] == 1) || (nastav\_vystup[4] == 1) || (nastav\_vystup[5] == 1) || (nastav\_vystup[6] == 1)) || ((nastav\_vystup[0] == 2) || (nastav\_vystup[1] == 2) || (nastav\_vystup[2] == 2) || (nastav\_vystup[3] == 2) || (nastav\_vystup[4] == 2) || (nastav\_vystup[5] == 2) || (nastav\_vystup[6] == 2))) {

lcd.write(byte(1));

lcd.setCursor(8, 1);

lcd.print("AKTIVNI ");

}

else {

lcd.write(byte(2));

lcd.setCursor(8, 1);

lcd.print("NEAKTIVNI");

}

lcd.setCursor(1, 2);

lcd.print("OUT2=");

if ((nastav\_vystup[0] == 1) || ((soucasnyBehVystupy == 2) && kteryVystup == NULL) || (soucasnyBehVystupy == 1) || (nastav\_vystup[1] == 1) || (nastav\_vystup[2] == 1) || (nastav\_vystup[3] == 1) || (nastav\_vystup[4] == 1) || (nastav\_vystup[5] == 1) || (nastav\_vystup[6] == 1) || (nastavvystupy1 == true)) {

lcd.write(byte(1));

lcd.setCursor(8, 2);

lcd.print("AKTIVNI ");

}

else {

lcd.write(byte(2));

lcd.setCursor(8, 2);

lcd.print("NEAKTIVNI");

}

lcd.setCursor(1, 3);

lcd.print("OUT3=");

if ((nastav\_vystup[0] == 2) || ((soucasnyBehVystupy == 2) && kteryVystup == 1) || (soucasnyBehVystupy == 1) || (nastav\_vystup[1] == 2) || (nastav\_vystup[2] == 2) || (nastav\_vystup[3] == 2) || (nastav\_vystup[4] == 2) || (nastav\_vystup[5] == 2) || (nastav\_vystup[6] == 2) || (nastavvystupy2 == true)) {

lcd.write(byte(1));

lcd.setCursor(8, 3);

lcd.print("AKTIVNI ");

}

else {

lcd.write(byte(2));

lcd.setCursor(8, 3);

lcd.print("NEAKTIVNI");

}

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko) {

if (limitdisplaye != 0) lcd.backlight();

cas2 = millis() / 1000;

if (tlacitko == 'A') {

blokaceSystemu(); // zablokuje systém dokud se nezmáčkne F1

}

if (tlacitko == '^')

{

lcd.setCursor(x, y);

lcd.print(" ");

y--;

if (y == 1) {

y = 3;

}

zpozdeni(20);

}

if (tlacitko == 'v')

{

lcd.setCursor(x, y);

lcd.print(" ");

y++;

if (y == 4) {

y = 2;

}

zpozdeni(20);

}

if (tlacitko == 'C')

break;

if (tlacitko == 'E')

{

if (promenna == 2) {

if (nastavvystupy1 == false) {

nastavvystupy1 = true;

zpozdeni(20);

continue;

}

if (nastavvystupy1 == true) {

nastavvystupy1 = false;

zpozdeni(20);

continue;

}

}

if (promenna == 3) {

if (nastavvystupy2 == false) {

nastavvystupy2 = true;

zpozdeni(20);

continue;

}

if (nastavvystupy2 == true) {

nastavvystupy2 = false;

zpozdeni(20);

continue;

}

}

}

}

}

x = NULL;

y = 3;

lcd.write(" ");

return 0;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void privitani(int doba\_trvani)

{

int yPozice[] = {0, 1, 2, 3, 0} ;

String uvitani [] = {

"\*ZAVLAZOVACI SYSTEM\*",

"\*v2.0 PRESKOC = ESC\*",

"\* KOCICA FILIP ME4 \*",

"\*IP: 192.168.10.145\*"

};

int p = NULL, s = 19, w = 0;

for (p = NULL; p < 9; p++) {

lcd.setCursor(p, 0);

lcd.print(uvitani[0][p]);

lcd.setCursor(p, 1);

lcd.print(uvitani[1][p]);

lcd.setCursor(p, 2);

lcd.print(uvitani[2][p]);

lcd.setCursor(p, 3);

lcd.print(uvitani[3][p]);

lcd.setCursor(s, 0);

lcd.print(uvitani[0][s]);

lcd.setCursor(s, 1);

lcd.print(uvitani[1][s]);

lcd.setCursor(s, 2);

lcd.print(uvitani[2][s]);

lcd.setCursor(s, 3);

lcd.print(uvitani[3][s]);

delay(200);

s--;

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko == 'C') {

w = 1;

break;

}

}

if (!w) {

for (p = NULL; p < 100; p++) {

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko == 'C') {

w = 1;

break;

}

yPozice[4] = (5000 / 100) \* 2;

delay(yPozice[4]);

if ((p % 20) == 0) {

lcd.setCursor(0, yPozice[0]);

lcd.print(uvitani[0]);

lcd.setCursor(0, yPozice[1]);

lcd.print(uvitani[1]);

lcd.setCursor(0, yPozice[2]);

lcd.print(uvitani[2]); // UVÍTÁNÍ, VYPSÁNÍ IP

lcd.setCursor(0, yPozice[3]);

lcd.print(uvitani[3]);

for (int t = 0; t < 4; t++) {

yPozice[t]++;

if (yPozice[t] == 4) yPozice[t] = 0;

}

}

}

}

if (!w) {

s = 10;

for (p = 9; p > 0; p--) {

lcd.setCursor(p, 0);

lcd.print(" ");

lcd.setCursor(p, 1);

lcd.print(" ");

lcd.setCursor(p, 2);

lcd.print(" ");

lcd.setCursor(p, 3);

lcd.print(" ");

lcd.setCursor(s, 0);

lcd.print(" ");

lcd.setCursor(s, 1);

lcd.print(" ");

lcd.setCursor(s, 2);

lcd.print(" ");

lcd.setCursor(s, 3);

lcd.print(" ");

delay(200);

s++;

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko == 'C') break;

}

}

lcd.clear();

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

int Zapouzdreni\_dat::hlaseniPoruchy(bool nastavvystupy1, bool nastavvystupy2, int Binarnivstup1, int i)

{

int poc, poc2; // počitadla

int plus = 53; // Piezo bzučák

int signal = 2; // -||-

pinMode(plus, OUTPUT);

pinMode(signal, OUTPUT); //PiezoBzučák

lcd.setCursor(1, 3);

lcd.print("OUT:1");

lcd.write(byte(2));

lcd.print("2");

lcd.write(byte(2));

lcd.print("3");

lcd.write(byte(2));

lcd.setCursor(x, y);

lcd.print(" ");

lcd.setCursor(17, 3);

lcd.write(":");

lcd.setCursor(17, 2);

lcd.write(" ");

cekej01 = millis();

for (;;) {

ethernet();

for (i = NULL; i <= 6; i++) {

nastav\_vystup[i] = NULL;

}

nastavvystupy1 = false;

nastavvystupy2 = false;

soucasnyBehVystupy = NULL;

digitalWrite(rele[0], LOW);

digitalWrite(rele[1], LOW);

digitalWrite(rele[2], LOW);

digitalWrite(rele[3], HIGH);

if ((millis() > cekej01 + 1) && (millis() < cekej01 + 500)) {

if (limitdisplaye != 0) lcd.backlight();

lcd.setCursor(17, 2);

lcd.write("P=");

lcd.setCursor(19, 2);

lcd.write(byte(1));

}

if ((millis() > cekej01 + 500) && (millis() < cekej01 + 1000)) {

lcd.noBacklight();

lcd.setCursor(17, 2);

lcd.write(" ");

}

if (millis() > cekej01 + 1000) cekej01 = millis();

for (poc = NULL; poc < 20; poc++) {

digitalWrite(signal, HIGH);

delayMicroseconds(5000);

digitalWrite(signal, LOW);

delayMicroseconds(5000);

}

for (poc = NULL; poc < 8; poc++) {

digitalWrite(signal, LOW);

delayMicroseconds(25000);

}

char tlacitko = stisknutiTlacitka.getKey();

KeyState stav = stisknutiTlacitka.getState();

if ((stav == PRESSED ) || (tlacitko)) {

pozastavPoruchu = true;

lcd.setCursor(17, 2);

lcd.write("P=");

lcd.setCursor(19, 2);

lcd.write(byte(1));

break;

}

Binarnivstup1 = nactiPort(A11);

if (Binarnivstup1) break;

}

if (limitdisplaye != 0) lcd.backlight();

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::stlaceniTlacitkaLoop()

{

cas2 = millis() / 1000;

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko) {

nactiVstupy(A14, A12, A15, A13);

zhasnuti = NULL;

if (limitdisplaye != 0) lcd.backlight();

}

for (;;) {

if (tlacitko == 'A') {

blokaceSystemu(); // zablokuje systém dokud se nezmáčkne F1

}

if (tlacitko == '^')

{

if (y == NULL) {

lcd.setCursor(x, y);

lcd.write(" ");

y = 3;

x = NULL;

zpozdeni(20);

break;

}

if (y == 1) {

y = NULL;

x = NULL;

zpozdeni(20);

lcd.setCursor(0, 1);

lcd.write(" ");

lcd.setCursor(16, 1);

lcd.write(" ");

break;

}

if (y == 2) {

y = 1;

x = NULL;

zpozdeni(20);

lcd.setCursor(0, 2);

lcd.write(" ");

break;

}

if ((y == 3) && (x == 11)) {

lcd.setCursor(11, 3);

lcd.write(" ");

y = 2;

x = NULL;

zpozdeni(20);

break;

}

if ((y == 3) && (x == NULL)) {

lcd.setCursor(0, 3);

lcd.write(" ");

y = 2;

x = NULL;

zpozdeni(20);

break;

}

}

if (tlacitko == 'v')

{

if ((y == 1) && (x == NULL)) {

lcd.setCursor(0, 1);

lcd.write(" ");

x = NULL;

y = 2;

zpozdeni(20);

break;

}

if ((y == 2) && (x == NULL)) {

lcd.setCursor(0, 2);

lcd.write(" ");

x = NULL;

y = 3;

zpozdeni(20);

break;

}

if ((y == 1) && (x == 16)) {

lcd.setCursor(16, 1);

lcd.write(" ");

x = NULL;

y = 2;

zpozdeni(20);

break;

}

if (y == 3) {

lcd.setCursor(x, y);

lcd.write(" ");

x = NULL;

y = NULL;

zpozdeni(20);

break;

}

if (y == NULL) {

y = 1;

x = NULL;

zpozdeni(20);

int o = NULL;

while (o <= 18) {

lcd.setCursor(o, 0);

lcd.write(" ");

o = o + 3;

}

break;

}

}

if (tlacitko == '>')

{

if ((x == 11) && (y == 3)) {

lcd.setCursor(11, 3);

lcd.print(" ");

x = NULL;

zpozdeni(20);

break;

}

if ((x == 16) && (y == 1)) {

lcd.setCursor(16, 1);

lcd.print(" ");

x = NULL;

zpozdeni(20);

break;

}

if ((x == 0) && (y == 1)) {

lcd.setCursor(0, 1);

lcd.print(" ");

x = 16;

zpozdeni(20);

break;

}

if ((x == 18) && (y == NULL)) {

lcd.setCursor(18, 0);

lcd.print(" ");

x = NULL;

zpozdeni(20);

break;

}

if ((x == 15) && (y == NULL)) {

lcd.setCursor(15, 0);

lcd.print(" ");

x = 18;

zpozdeni(20);

break;

}

if ((x == 12) && (y == NULL)) {

lcd.setCursor(12, 0);

lcd.print(" ");

x = 15;

zpozdeni(20);

break;

}

if ((x == 9) && (y == NULL)) {

lcd.setCursor(9, 0);

lcd.print(" ");

x = 12;

zpozdeni(20);

break;

}

if ((x == 6) && (y == NULL)) {

lcd.setCursor(6, 0);

lcd.print(" ");

x = 9;

zpozdeni(20);

break;

}

if ((x == 3) && (y == NULL)) {

lcd.setCursor(3, 0);

lcd.print(" ");

x = 6;

zpozdeni(20);

break;

}

if ((x == NULL) && (y == NULL)) {

lcd.setCursor(0, 0);

lcd.print(" ");

x = 3;

zpozdeni(20);

break;

}

if ((x == NULL) && (y == 3)) {

lcd.setCursor(0, 3);

lcd.print(" ");

x = 11;

zpozdeni(20);

break;

}

}

if (tlacitko == '<')

{

if ((x == NULL) && (y == 3)) {

lcd.setCursor(0, 3);

lcd.print(" ");

x = 11;

zpozdeni(20);

break;

}

if ((x == NULL) && (y == 1)) {

lcd.setCursor(0, 1);

lcd.print(" ");

x = 16;

zpozdeni(20);

break;

}

if ((x == NULL) && (y == NULL)) {

lcd.setCursor(0, 0);

lcd.print(" ");

x = 18;

zpozdeni(20);

break;

}

if ((x == 3) && (y == NULL)) {

lcd.setCursor(3, 0);

lcd.print(" ");

x = NULL;

zpozdeni(20);

break;

}

if ((x == 6) && (y == NULL)) {

lcd.setCursor(6, 0);

lcd.print(" ");

x = 3;

zpozdeni(20);

break;

}

if ((x == 16) && (y == 1)) {

lcd.setCursor(16, 1);

lcd.print(" ");

x = 0;

zpozdeni(20);

break;

}

if ((x == 9) && (y == NULL)) {

lcd.setCursor(9, 0);

lcd.print(" ");

x = 6;

zpozdeni(20);

break;

}

if ((x == 12) && (y == NULL)) {

lcd.setCursor(12, 0);

lcd.print(" ");

x = 9;

zpozdeni(20);

break;

}

if ((x == 15) && (y == NULL)) {

lcd.setCursor(15, 0);

lcd.print(" ");

x = 12;

zpozdeni(20);

break;

}

if ((x == 18) && (y == NULL)) {

lcd.setCursor(18, 0);

lcd.print(" ");

x = 15;

zpozdeni(20);

break;

}

if ((x == 11) && (y == 3)) {

lcd.setCursor(11, 3);

lcd.print(" ");

x = NULL;

zpozdeni(20);

break;

}

}

break;

}

if (tlacitko == 'E')

{

if ((y == 2) && (x == NULL)) {

lcd.clear();

kalibrace();

}

if ((y == 3) && (x == 11)) {

lcd.clear();

nastavcas(limitdisplaye);

}

if ((y == NULL) && (x == NULL)) {

lcd.clear();

i = NULL;

cas(limitdisplaye);

}

if ((y == 1) && (x == 16)) {

lcd.clear();

rezim = vyberRezim();

}

if ((y == NULL) && (x == 3)) { // vyber v menu a skoky do fci

lcd.clear();

i = 1;

cas(limitdisplaye);

}

if ((y == NULL) && (x == 6)) {

lcd.clear();

i = 2;

cas(limitdisplaye);

}

if ((y == NULL) && (x == 9)) {

lcd.clear();

i = 3;

cas(limitdisplaye);

}

if ((y == NULL) && (x == 12)) {

lcd.clear();

i = 4;

cas(limitdisplaye);

}

if ((y == NULL) && (x == 15)) {

lcd.clear();

i = 5;

cas(limitdisplaye);

}

if ((y == NULL) && (x == 18)) {

lcd.clear();

i = 6;

cas(limitdisplaye);

}

if ((y == 1) && (x == NULL)) {

lcd.clear();

nastaveniParametru(limitdisplaye, nasthladiny , nastvlhkosti, nasthystereze);

}

if ((y == 3) && (x == NULL)) {

lcd.clear();

nastavvystupy(limitdisplaye);

}

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// Funkce s převodem z desítkové soustavy na BCD kod.

byte DECnaBCD(byte hodnota)

{

return ( (hodnota / 10 \* 16) + (hodnota % 10) );

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// Funkce s převodem z BCD kodu do desitkove soustavy.

byte BCDnaDEC(byte hodnota)

{

return ( (hodnota / 16 \* 10) + (hodnota % 16) );

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::zapisEEPROM()

{

if ((nasthladiny >= NULL) && (nasthladiny <= 100)) EEPROM.write(0 + rezimEEPROM, nasthladiny);

if ((nastvlhkosti >= NULL) && (nastvlhkosti <= 100)) EEPROM.write(1 + rezimEEPROM, nastvlhkosti);

if ((limitdisplaye >= NULL) && (limitdisplaye <= 60)) EEPROM.write(2 + rezimEEPROM, limitdisplaye); // zápis proměnných do eeprom!

if ((nedele2 >= NULL) && (nedele2 <= 9)) EEPROM.write(3 + rezimEEPROM, nedele2);

if ((soucasnyBeh >= NULL) && (soucasnyBeh <= 1)) EEPROM.write(4 + rezimEEPROM, soucasnyBeh);

if ((nasthystereze >= NULL) && (nasthystereze <= 10)) EEPROM.write(5 + rezimEEPROM, nasthystereze);

if ((kalibraceVlhkost >= NULL) && (kalibraceVlhkost <= 1023)) EEPROM.write(6 + rezimEEPROM, kalibraceVlhkost);

if ((kalibraceHladina >= NULL) && (kalibraceHladina <= 1023)) EEPROM.write(7 + rezimEEPROM, kalibraceHladina);

int c = 8 + rezimEEPROM;

for (int a = NULL; a < 7; a++) {

if ((dny[a].hodiny1 >= NULL) && (dny[a].hodiny1 <= 2)) EEPROM.write(c, dny[a].hodiny1);

c++;

if ((dny[a].hodiny2 >= NULL) && (dny[a].hodiny2 <= 9)) EEPROM.write(c, dny[a].hodiny2); // zápis struktury do eeprom!

c++;

if ((dny[a].hodinycelk >= NULL) && (dny[a].hodinycelk <= 23)) EEPROM.write(c, dny[a].hodinycelk);

c++;

if ((dny[a].minuty1 >= NULL) && (dny[a].minuty1 <= 5)) EEPROM.write(c, dny[a].minuty1);

c++;

if ((dny[a].minuty2 >= NULL) && (dny[a].minuty2 <= 9)) EEPROM.write(c, dny[a].minuty2);

c++;

if ((dny[a].minutycelk >= NULL) && (dny[a].minutycelk <= 59)) EEPROM.write(c, dny[a].minutycelk);

c++;

if ((dny[a].min1 >= NULL) && (dny[a].min1 <= 9)) EEPROM.write(c, dny[a].min1);

c++;

if ((dny[a].min10 >= NULL) && (dny[a].min10 <= 9)) EEPROM.write(c, dny[a].min10);

c++;

if ((dny[a].min1celk >= NULL) && (dny[a].min1celk <= 99)) EEPROM.write(c, dny[a].min1celk);

c++;

if ((dny[a].min2 >= NULL) && (dny[a].min2 <= 9)) EEPROM.write(c, dny[a].min2);

c++;

if (a != 6) {

if ((dny[a].min20 >= NULL) && (dny[a].min20 <= 9)) EEPROM.write(c, dny[a].min20);

}

c++;

if ((dny[a].min2celk >= NULL) && (dny[a].min2celk <= 99)) EEPROM.write(c, dny[a].min2celk);

c++;

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::prectiEEPROM()

{

nasthladiny = EEPROM.read(0 + rezimEEPROM); // čtení z EEPROM

if (nasthladiny > 100) nasthladiny = 100;

nastvlhkosti = EEPROM.read(1 + rezimEEPROM);

if (nastvlhkosti > 100) nastvlhkosti = 100;

limitdisplaye = EEPROM.read(2 + rezimEEPROM);

if (limitdisplaye > 60) limitdisplaye = 60;

nedele2 = EEPROM.read(3 + rezimEEPROM);

if (nedele2 > 9) nedele2 = NULL;

soucasnyBeh = EEPROM.read(4 + rezimEEPROM);

if (soucasnyBeh > 1) soucasnyBeh = NULL;

nasthystereze = EEPROM.read(5 + rezimEEPROM);

if (nasthystereze > 10) nasthystereze = NULL;

kalibraceVlhkost = EEPROM.read(6 + rezimEEPROM);

if (kalibraceVlhkost > 1023) kalibraceVlhkost = 1023;

kalibraceHladina = EEPROM.read(7 + rezimEEPROM);

if (kalibraceHladina > 1023) kalibraceHladina = 1023;

int c = 8 + rezimEEPROM;

for (int a = NULL; a < 7; a++) {

dny[a].hodiny1 = EEPROM.read(c);

if ((dny[a].hodiny1 < 0 ) || (dny[a].hodiny1 > 2)) dny[a].hodiny1 = NULL;

c++;

dny[a].hodiny2 = EEPROM.read(c);

if ((dny[a].hodiny2 < 0 ) || (dny[a].hodiny2 > 9)) dny[a].hodiny2 = NULL;

c++;

dny[a].hodinycelk = EEPROM.read(c);

if ((dny[a].hodinycelk < 0 ) || (dny[a].hodinycelk > 23)) {

dny[a].hodiny1 = NULL;

dny[a].hodiny2 = NULL;

dny[a].hodinycelk = NULL;

}

c++;

dny[a].minuty1 = EEPROM.read(c);

if ((dny[a].minuty1 < 0 ) || (dny[a].minuty1 > 5)) dny[a].minuty1 = NULL;

c++;

dny[a].minuty2 = EEPROM.read(c);

if ((dny[a].minuty2 < 0 ) || (dny[a].minuty2 > 9)) dny[a].minuty2 = NULL;

c++;

dny[a].minutycelk = EEPROM.read(c);

if ((dny[a].minutycelk < 0 ) || (dny[a].minutycelk > 59)) dny[a].minutycelk = NULL;

c++;

dny[a].min1 = EEPROM.read(c);

if ((dny[a].min1 < 0 ) || (dny[a].min1 > 9)) dny[a].min1 = NULL;

c++;

dny[a].min10 = EEPROM.read(c);

if ((dny[a].min10 < 0 ) || (dny[a].min10 > 9)) dny[a].min10 = NULL;

c++;

dny[a].min1celk = EEPROM.read(c);

if ((dny[a].min1celk < 0 ) || (dny[a].min1celk > 99)) dny[a].min1celk = NULL;

c++;

dny[a].min2 = EEPROM.read(c);

if ((dny[a].min2 < 0 ) || (dny[a].min2 > 9)) dny[a].min2 = NULL;

c++;

if (a != 6) {

dny[a].min20 = EEPROM.read(c);

if ((dny[a].min20 < 0 ) || (dny[a].min20 > 9)) dny[a].min20 = NULL;

}

c++;

dny[a].min2celk = EEPROM.read(c);

if ((dny[a].min2celk < 0 ) || (dny[a].min2celk > 99)) dny[a].min2celk = NULL;

c++;

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::zobrazNaDisplayi()

{

if (AN1 > 100) AN1 = 100;

if (AN2 > 100) AN2 = 100;

lcd.setCursor(1, 2);

lcd.print("MV=");

lcd.print(AN2);

lcd.print("%");

if ((AN2 - nasthystereze) < nastvlhkosti) {

lcd.write(byte(1));

}

else {

lcd.write(byte(2));

}

if ((AN2 < 100) && (AN2 > 9)) {

lcd.setCursor(8, 2);

lcd.print(" ");

}

else if (AN2 < 10) {

lcd.setCursor(7, 2);

lcd.print(" ");

}

lcd.setCursor(9, 2);

lcd.print("MH=");

lcd.print(AN1);

lcd.print("%");

if (AN1 > (nasthladiny - nasthystereze)) {

lcd.write(byte(1));

}

else {

lcd.write(byte(2));

}

if ((AN1 < 100) && (AN1 > 9)) {

lcd.setCursor(16, 2);

lcd.print(" ");

}

else if (AN1 < 10) {

lcd.setCursor(15, 2);

lcd.print(" ");

}

lcd.setCursor(1, 1);

lcd.print("NV=");

lcd.print(nastvlhkosti);

lcd.print("%");

if ((nastvlhkosti < 100) && (nastvlhkosti > 9)) {

lcd.setCursor(7, 1);

lcd.print(" ");

}

else if (nastvlhkosti < 10) {

lcd.setCursor(6, 1);

lcd.print(" ");

}

if (( x == 0 ) && ( y == 0 ))

cteniORC(&sekundy, &minuty, &hodiny, &denVTydnu, &denVMesici);

else {

lcd.setCursor(0, 0);

lcd.print(" ");

}

lcd.setCursor(9, 1);

lcd.print("NH=");

lcd.print(nasthladiny);

lcd.print("%");

if ((nasthladiny < 100) && (nasthladiny > 9)) {

lcd.setCursor(15, 1);

lcd.print(" ");

}

else if (nasthladiny < 10) {

lcd.setCursor(14, 1);

lcd.print(" ");

}

lcd.setCursor(1, 0);

lcd.write("Po");

lcd.setCursor(4, 0);

lcd.write("Ut");

lcd.setCursor(7, 0);

lcd.write("St");

lcd.setCursor(10, 0);

lcd.write("Ct");

lcd.setCursor(13, 0);

lcd.write("Pa");

lcd.setCursor(16, 0);

lcd.write("So");

lcd.setCursor(19, 0);

lcd.write("N");

lcd.setCursor(12, 3);

switch (denVTydnu) {

case 1:

lcd.print("Po,");

break;

case 2:

lcd.print("Ut,");

break;

case 3:

lcd.print("St,");

break;

case 4:

lcd.print("Ct,");

break;

case 5:

lcd.print("Pa,");

break;

case 6:

lcd.print("So,");

break;

case 7:

lcd.print("Ne,");

break;

}

lcd.setCursor(17, 1); //rezim

lcd.print("R=");

lcd.print(rezim + 1);

lcd.setCursor(15, 3);

if (hodiny < 10) {

lcd.print("0");

}

lcd.print(hodiny, DEC); //vypsani casu

lcd.setCursor(18, 3);

if (minuty < 10) {

lcd.print("0");

}

lcd.print(minuty, DEC);

Binarnivstup1 = nactiPort(A11);

if (Binarnivstup1) {

digitalWrite(rele[3], LOW); //porucha

lcd.setCursor(17, 2);

lcd.write("P=");

lcd.setCursor(19, 2);

lcd.write(byte(2));

}

else {

for (int w = 0; w < 7; w++) nastav\_vystup[w] = NULL;

nastavvystupy1 = false;

nastavvystupy2 = false;

soucasnyBehVystupy = NULL;

digitalWrite(rele[0], LOW);

digitalWrite(rele[1], LOW);

digitalWrite(rele[2], LOW);

digitalWrite(rele[3], HIGH); //porucha

lcd.setCursor(17, 2);

lcd.write("P=");

lcd.setCursor(19, 2);

lcd.write(byte(1));

}

lcd.setCursor(1, 3);

lcd.print("OUT:");

lcd.print("1");

if (((nastavvystupy1 == true) || ((soucasnyBehVystupy == 2) && ((kteryVystup == NULL) || (kteryVystup == 1))) || (soucasnyBehVystupy == 1) || (nastavvystupy2 == true)) || ((nastav\_vystup[0] == 1) || (nastav\_vystup[1] == 1) || (nastav\_vystup[2] == 1) || (nastav\_vystup[3] == 1) || (nastav\_vystup[4] == 1) || (nastav\_vystup[5] == 1) || (nastav\_vystup[6] == 1)) || ((nastav\_vystup[0] == 2) || (nastav\_vystup[1] == 2) || (nastav\_vystup[2] == 2) || (nastav\_vystup[3] == 2) || (nastav\_vystup[4] == 2) || (nastav\_vystup[5] == 2) || (nastav\_vystup[6] == 2))) {

lcd.write(byte(1));

digitalWrite(rele[0], HIGH);

nactiVstupy(A14, A12, A15, A13);

}

else {

lcd.write(byte(2));

digitalWrite(rele[0], LOW);

}

lcd.print("2");

if ((nastav\_vystup[0] == 1) || ((soucasnyBehVystupy == 2) && kteryVystup == NULL) || (soucasnyBehVystupy == 1) || (nastav\_vystup[1] == 1) || (nastav\_vystup[2] == 1) || (nastav\_vystup[3] == 1) || (nastav\_vystup[4] == 1) || (nastav\_vystup[5] == 1) || (nastav\_vystup[6] == 1) || (nastavvystupy1 == true)) {

lcd.write(byte(1));

digitalWrite(rele[1], HIGH);

}

else {

lcd.write(byte(2));

digitalWrite(rele[1], LOW);

}

lcd.print("3");

if ((nastav\_vystup[0] == 2) || ((soucasnyBehVystupy == 2) && kteryVystup == 1) || (soucasnyBehVystupy == 1) || (nastav\_vystup[1] == 2) || (nastav\_vystup[2] == 2) || (nastav\_vystup[3] == 2) || (nastav\_vystup[4] == 2) || (nastav\_vystup[5] == 2) || (nastav\_vystup[6] == 2) || (nastavvystupy2 == true)) {

lcd.write(byte(1));

digitalWrite(rele[2], HIGH);

}

else {

lcd.write(byte(2));

digitalWrite(rele[2], LOW);

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void problikavani()

{

if ((millis() > cekej01 + 1) && (millis() < cekej01 + 1500)) {

lcd.setCursor(x, y);

lcd.write(byte(3)); //blikání kurzoru

}

if ((millis() > cekej01 + 1500) && (millis() < cekej01 + 1600)) {

lcd.setCursor(x, y);

lcd.write(" "); //blikání kurzoru

}

if (millis() > cekej01 + 1600) cekej01 = millis();

if ((millis() > cekej + 1) && (millis() < cekej + 1000)) {

lcd.setCursor(17, 3);

lcd.print(":"); //blikani dvojtecky v casu ve vterinovem intervalu

}

if ((millis() > cekej + 1000) && (millis() < cekej + 2000)) {

lcd.setCursor(17, 3);

lcd.print(" "); //blikani dvojtecky v casu ve vterinovem intervalu

}

if (millis() > cekej + 2000) {

cekej = millis();

zhasnuti += 2;

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// funkce pro zápis do obvodu reálného času přes Wire.h

void zapisORC(byte sekundy, byte minuty, byte hodiny, byte denVTydnu, byte denVMesici)

{

Wire.beginTransmission(ORC\_I2C\_ADRESA);

Wire.write(0);

if ((sekundy >= NULL) && (sekundy < 60)) Wire.write(DECnaBCD(sekundy));

if ((minuty >= NULL) && (minuty < 60)) Wire.write(DECnaBCD(minuty));

if ((hodiny >= NULL) && (hodiny < 24)) Wire.write(DECnaBCD(hodiny));

if ((denVTydnu > 0) && (denVTydnu < 8)) Wire.write(DECnaBCD(denVTydnu));

if ((denVMesici >= NULL) && (denVMesici < 32)) Wire.write(DECnaBCD(denVMesici));

Wire.endTransmission();

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

// funkce pro čtení z obvodu reálného času přes Wire.h

void cteniORC(byte \* sekundy, byte \* minuty, byte \* hodiny, byte \* denVTydnu, byte \* denVMesici)

{

Wire.beginTransmission(ORC\_I2C\_ADRESA);

Wire.write(0);

Wire.endTransmission();

Wire.requestFrom(ORC\_I2C\_ADRESA, 7);

\*sekundy = BCDnaDEC(Wire.read() & 0x7f);

\*minuty = BCDnaDEC(Wire.read());

\*hodiny = BCDnaDEC(Wire.read() & 0x3f);

\*denVTydnu = BCDnaDEC(Wire.read());

\*denVMesici = BCDnaDEC(Wire.read());

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::ethernet()

{

// dostupnost klienta

EthernetClient client = server.available();

String buffer = "";

// dokud je klient připojen

while (client.connected()) {

// čti data dokud nenarazíš na znak nového řádku

if (client.available()) {

char c = client.read();

buffer = buffer + c;

if (c == '\n') {

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.println("<head>");

client.println("<meta charset='UTF-8'>"); // diakritika českeho jazyka

client.print("<a href='http://192.168.10.145?menu=14'><button style='background:black;width:100%;height:80px'><font color='gold'><marquee direction='left'><h1>Ethernetové rozhraní pro zavlažovací systém, Kočica Filip ME4</h1></marquee></font></button></a>");

client.print("<a href='http://192.168.10.145?menu=6'><button style='background:blue;width:14.285714%;height:80px'><font color='gold'><h1>PONDĚLÍ</h1></font></button></a>"); //po

client.print("<a href='http://192.168.10.145?menu=7'><button style='background:blue;width:14.285714%;height:80px'><font color='gold'><h1>ÚTERÝ</h1></font></button></a>"); //ut

client.print("<a href='http://192.168.10.145?menu=8'><button style='background:blue;width:14.285714%;height:80px'><font color='gold'><h1>STŘEDA</h1></font></button></a>"); //st

client.print("<a href='http://192.168.10.145?menu=9'><button style='background:blue;width:14.285714%;height:80px'><font color='gold'><h1>ČTVRTEK</h1></font></button></a>"); //ct

client.print("<a href='http://192.168.10.145?menu=16'><button style='background:blue;width:14.285714%;height:80px'><font color='gold'><h1>PÁTEK</h1></font></button></a>"); //pa

client.print("<a href='http://192.168.10.145?menu=11'><button style='background:blue;width:14.285714%;height:80px'><font color='gold'><h1>SOBOTA</h1></font></button></a>"); //so

client.print("<a href='http://192.168.10.145?menu=12'><button style='background:blue;width:14.285714%;height:80px'><font color='gold'><h1>NEDĚLE</h1></font></button></a>"); //ne

client.print("<a href='http://192.168.10.145?menu=2'><button style='background:purple;width:80%;height:80px'><font color='silver'><h1>Nastavení parametrů</h1></font></button></a>"); //nastav parametru

client.print("<a href='http://192.168.10.145?menu=13'><button style='background:orange;width:20%;height:80px'><font color='black'><h1>Režim + Kalibrace</h1></font></button></a>"); //režim a kalibrace

client.print("<a href='http://192.168.10.145?menu=3'><button style='background:purple;width:80%;height:80px'><font color='silver'><h1>Vstupy</h1></font></button></a>"); //vstupy

if (Binarnivstup1)

client.print("<a href='http://192.168.10.145?menu=15'><button style='background:green;width:20%;height:80px'><h1>Porucha rozepnuta</h1></button></a>"); //porucha

else

client.print("<a href='http://192.168.10.145?menu=15'><button style='background:red;width:20%;height:80px'><h1>Porucha sepnuta</h1></button></a>"); //porucha

client.print("<a href='http://192.168.10.145?menu=4'><button style='background:purple;width:60%;height:80px'><font color='silver'><h1>Výstupy</h1></font></button></a>"); //vystupy

client.print("<a href='http://192.168.10.145?menu=5'><button style='background:purple;width:40%;height:80px'><font color='silver'><h1>Aktualní čas</h1></font></button></a>"); //aktual cas na ardu

client.println("<title>\* Ethernetové rozhraní pro zavlažovací systém, Kočica Filip ME4 \*</title>");

client.println("<meta http-equiv='refresh' content='60' >"); // refresh každou minutu

client.println("</head>");

client.println("<body>");

client.println("<br />");

client.println("<h1>");

client.println("<center>");

// -----------------------------------------------------------------------------------------------------------------------------------------------------------

int pomocna\_promenna = NULL;

for (int v = NULL; v <= pocetZnaku(buffer); v++) {

for (char ch = '0'; ch <= '9' ; ch++) {

if ((buffer[v] == 'm') && (buffer[v + 1] == 'e') && (buffer[v + 2] == 'n') && (buffer[v + 3] == 'u') && (buffer[v + 4] == '=') && (buffer[v + 5] == ch)) Menu = pomocna\_promenna;

if ((buffer[v] == 'm') && (buffer[v + 1] == 'e') && (buffer[v + 2] == 'n') && (buffer[v + 3] == 'u') && (buffer[v + 4] == '=') && (buffer[v + 5] == '1') && (buffer[v + 6] == ch)) Menu = 10 + pomocna\_promenna;

pomocna\_promenna++;

}

pomocna\_promenna = NULL;

if ((buffer[v] == 'o') && (buffer[v + 1] == 'u') && (buffer[v + 2] == 't') && (buffer[v + 3] == '2') && (buffer[v + 4] == '=') && (buffer[v + 5] == '0')) nastavvystupy1 = false;

if ((buffer[v] == 'o') && (buffer[v + 1] == 'u') && (buffer[v + 2] == 't') && (buffer[v + 3] == '2') && (buffer[v + 4] == '=') && (buffer[v + 5] == '1')) nastavvystupy1 = true;

if ((buffer[v] == 'o') && (buffer[v + 1] == 'u') && (buffer[v + 2] == 't') && (buffer[v + 3] == '3') && (buffer[v + 4] == '=') && (buffer[v + 5] == '0')) nastavvystupy2 = false;

if ((buffer[v] == 'o') && (buffer[v + 1] == 'u') && (buffer[v + 2] == 't') && (buffer[v + 3] == '3') && (buffer[v + 4] == '=') && (buffer[v + 5] == '1')) nastavvystupy2 = true;

if ((buffer[v] == 'B') && (buffer[v + 1] == 'e') && (buffer[v + 2] == 'h') && (buffer[v + 3] == '=') && (buffer[v + 4] == '0')) {

soucasnyBeh = false;

zapisEEPROM();

}

}

// -----------------------------------------------------------------------------------------------------------------------------------------------------------

switch (rezim) {

case 0:

rezimEEPROM = NULL;

break;

case 1:

rezimEEPROM = 100;

break;

case 2:

rezimEEPROM = 200;

break;

}

if ((Menu == 6) || (Menu == 7) || (Menu == 8) || (Menu == 9) || (Menu == 16) || (Menu == 11) || (Menu == 12)) {

//---------------------VYPSANI PO-NE NASTAVENYCH CASU !!!!----------------------------------------------------------------

int vypisDniNaSit = NULL;

if (Menu < 13) vypisDniNaSit = Menu - 6;

else vypisDniNaSit = Menu - 12;

client.println("<input type=button onclick='history.back()' value='Zpět'>");

client.println("<br />");

client.println("<br />");

switch (vypisDniNaSit) {

case 0:

client.println("Pondělí ");

break;

case 1:

client.println("Úterý ");

break;

case 2:

client.println("Středa ");

break;

case 3:

client.println("Čtvrtek ");

break;

case 4:

client.println("Pátek ");

break;

case 5:

client.println("Sobota ");

break;

case 6:

client.println("Neděle ");

break;

}

client.print("<FORM action='http://192.168.10.145' method='GET'>");

client.println("</h1>");

client.println("<center>");

client.println(dny[vypisDniNaSit].hodiny1);

client.println(dny[vypisDniNaSit].hodiny2);

client.println(":");

client.println(dny[vypisDniNaSit].minuty1);

client.println(dny[vypisDniNaSit].minuty2);

client.println(" - ");

client.println("<input type='number' style='background:yellow;width:30px' name='q1' min='0' max='23'>");

client.println(":");

client.println("<input type='number' style='background:yellow;width:30px' name='q2' min='0' max='59'>");

client.println("<br />");

client.println("<h1>");

client.println("1.Okruh");

client.println("</h1>");

client.println(dny[vypisDniNaSit].min1);

client.println(dny[vypisDniNaSit].min10);

client.println(" - ");

client.println("<input type='number' style='background:yellow;width:30px' name='q3' min='0' max='99'>");

client.println("<br />");

client.println("<h1>");

client.println("2.Okruh");

client.println("</h1>");

client.println(dny[vypisDniNaSit].min2);

if (vypisDniNaSit == 6) client.println(nedele2);

else client.println(dny[vypisDniNaSit].min20);

client.println(" - ");

client.println("<input type='number' style='background:yellow;width:30px' name='q4' min='0' max='99'>");

client.println("<br />");

client.println("<h1>");

client.println("Současný Běh je ");

if (soucasnyBeh) {

client.print("<font color='green'>ZAPNUTÝ</font>");

} else {

client.print("<font color='red'>VYPNUTÝ</font>");

}

client.print("<FORM action='http://192.168.10.145' method='GET'>");

client.print("<P> <INPUT type='radio' name='Beh' value='1'>ON");

client.print("<P> <INPUT type='radio' name='Beh' value='0'>OFF");

if (soucasnyBeh)

client.print("<P> <INPUT type='submit' style='background:red;width:100px;height:50px'> </FORM>");

else

client.print("<P> <INPUT type='submit' style='background:lightgreen;width:100px;height:50px'> </FORM>");

for (int v = NULL; v <= pocetZnaku(buffer); v++) {

//---------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '1') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // hodiny 2

dny[vypisDniNaSit].hodiny1 = NULL;

dny[vypisDniNaSit].hodiny2 = ((int)buffer[v + 3] - 48);

dny[vypisDniNaSit].hodinycelk = ((int)buffer[v + 3] - 48);

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '1') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&') && (buffer[v + 5] == '&')) { // hodiny 1 + 2

dny[vypisDniNaSit].hodiny1 = ((int)buffer[v + 3] - 48);

dny[vypisDniNaSit].hodiny2 = ((int)buffer[v + 4] - 48);

dny[vypisDniNaSit].hodinycelk = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

//---------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '2') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // minuty 2

dny[vypisDniNaSit].minuty1 = NULL;

dny[vypisDniNaSit].minuty2 = ((int)buffer[v + 3] - 48);

dny[vypisDniNaSit].minutycelk = ((int)buffer[v + 3] - 48);

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '2') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&') && (buffer[v + 5] == '&')) { // minuty 1 + 2

dny[vypisDniNaSit].minuty1 = ((int)buffer[v + 3] - 48);

dny[vypisDniNaSit].minuty2 = ((int)buffer[v + 4] - 48);

dny[vypisDniNaSit].minutycelk = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

//---------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '3') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // okruh 10

dny[vypisDniNaSit].min1 = NULL;

dny[vypisDniNaSit].min10 = ((int)buffer[v + 3] - 48);

dny[vypisDniNaSit].min1celk = ((int)buffer[v + 3] - 48);

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '3') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&')) { // okruh 1 + 10

dny[vypisDniNaSit].min1 = ((int)buffer[v + 3] - 48);

dny[vypisDniNaSit].min10 = ((int)buffer[v + 4] - 48);

dny[vypisDniNaSit].min1celk = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

//---------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '4') && (buffer[v + 2] == '=') && ((buffer[v + 3] != '&') && (buffer[v + 3] != ' '))) { // okruh 20

dny[vypisDniNaSit].min2 = NULL;

if (vypisDniNaSit == 6) nedele2 = ((int)buffer[v + 3] - 48);

else dny[vypisDniNaSit].min20 = ((int)buffer[v + 3] - 48);

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '4') && (buffer[v + 2] == '=') && ((buffer[v + 3] != '&') && (buffer[v + 3] != ' ')) && ((buffer[v + 4] != '&') && (buffer[v + 4] != ' '))) { // okruh 2 + 20

dny[vypisDniNaSit].min2 = ((int)buffer[v + 3] - 48);

if (vypisDniNaSit == 6) nedele2 = ((int)buffer[v + 4] - 48);

else dny[vypisDniNaSit].min20 = ((int)buffer[v + 4] - 48);

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

//---------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'B') && (buffer[v + 1] == 'e') && (buffer[v + 2] == 'h') && (buffer[v + 3] == '=') && (buffer[v + 4] == '1')) {

soucasnyBeh = true;

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

}

if (!vypisMozny) {

lcd.clear();

vypisMozny = 1;

}

//---------------------VYPSANI PO-NE NASTAVENYCH CASU !!!!----------------------------------------------------------------

}

switch (Menu) {

case 2:

client.println("<input type=button onclick='history.back()' value='Zpět'>");

client.println("<br />");

client.println("<br />");

client.print("<FORM action='http://192.168.10.145' method='GET'>");

client.print("Vlhkost (max 100%)");

client.println("</h1>");

client.println("<center>");

client.print(nastvlhkosti);

client.println(" - ");

client.println("<input type='number' style='background:yellow;width:30px' name='q5' min='0' max='100'>");

client.println("<br />");

client.println("<br />");

client.println("<h1>");

client.print("Hladina (max 100%)");

client.println("</h1>");

client.print(nasthladiny);

client.println(" - ");

client.println("<input type='number' style='background:yellow;width:30px' name='q6' min='0' max='100'>");

client.println("<br />");

client.println("<br />");

client.println("<h1>");

client.print("Hystereze (max 10 %)");

client.println("</h1>");

client.print(nasthystereze);

client.println(" - ");

client.println("<input type='number' style='background:yellow;width:30px' name='q7' min='0' max='10'>");

client.println("<br />");

client.println("<br />");

client.println("<h1>");

client.print("Limit Displaye (0,1,2,3,5,10,15,30,45,60)");

client.println("</h1>");

client.print(limitdisplaye);

client.println(" - ");

client.println("<input type='number' style='background:yellow;width:38px' name='q8' min='0' max='60'>");

client.println("<br />");

client.print("<P> <INPUT type='submit' style='background:silver;width:100px;height:50px'> </FORM>");

client.println("<br />");

for (int v = NULL; v <= pocetZnaku(buffer); v++) {

//-------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '5') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // nast vlhkosti 1

nastvlhkosti = ((int)buffer[v + 3] - 48);

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '5') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&')) { // nast vlhkosti 1 + 2

nastvlhkosti = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '5') && (buffer[v + 2] == '=') && (buffer[v + 3] != ' ') && (buffer[v + 4] != ' ') && (buffer[v + 5] != ' ') && (buffer[v + 6] == '&')) { // nast vlhkosti 1 + 2 + 3

nastvlhkosti = ((100 \* ((int)buffer[v + 3] - 48)) + (10 \* ((int)buffer[v + 4] - 48)) + ((int)buffer[v + 5] - 48));

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '6') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // nast hladiny 1

nasthladiny = ((int)buffer[v + 3] - 48);

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '6') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&')) { // nast hladiny 1 + 2

nasthladiny = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '6') && (buffer[v + 2] == '=') && (buffer[v + 3] != ' ') && (buffer[v + 4] != ' ') && (buffer[v + 5] != ' ') && (buffer[v + 6] == '&')) { // nast hladiny 1 + 2 + 3

nasthladiny = ((100 \* ((int)buffer[v + 3] - 48)) + (10 \* ((int)buffer[v + 4] - 48)) + ((int)buffer[v + 5] - 48));

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '7') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // nast hystereze 1

nasthystereze = ((int)buffer[v + 3] - 48);

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '7') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&')) { // nast hystereze 1 + 2

nasthystereze = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '8') && (buffer[v + 2] == '=') && (buffer[v + 3] != ' ') && (buffer[v + 4] == ' ')) { // nast limitu displaye 1

limitdisplaye = ((int)buffer[v + 3] - 48);

if ((limitdisplaye == 0) || (limitdisplaye == 1) || (limitdisplaye == 2) || (limitdisplaye == 3) || (limitdisplaye == 5) || (limitdisplaye == 10) || (limitdisplaye == 15) || (limitdisplaye == 30) || (limitdisplaye == 45) || (limitdisplaye == 60)) {

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

else

prectiEEPROM();

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '8') && (buffer[v + 2] == '=') && (buffer[v + 3] != ' ') && (buffer[v + 4] != '&') && (buffer[v + 5] == ' ')) { // nast limitu displaye 1 + 2

limitdisplaye = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

if ((limitdisplaye == 0) || (limitdisplaye == 1) || (limitdisplaye == 2) || (limitdisplaye == 3) || (limitdisplaye == 5) || (limitdisplaye == 10) || (limitdisplaye == 15) || (limitdisplaye == 30) || (limitdisplaye == 45) || (limitdisplaye == 60)) {

zapisEEPROM();

if (vypisMozny) nastaveniNaEthernetu(2000, 0);

}

else

prectiEEPROM();

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------

}

if (!vypisMozny) {

lcd.clear();

vypisMozny = 1;

}

break;

case 3:

client.println("<input type=button onclick='history.back()' value='Zpět'>");

client.println("<br />");

client.println("<br />");

//------------------------ zobrazeni vlhkosti a hladiny a jestli je podminka splnena nebo ne -----------------------------

client.print("Nastavená Vlhkost: ");

client.print(nastvlhkosti);

client.print("%");

client.println("<br />");

client.print("Naměřená Vlhkost: ");

client.print(AN2);

client.print("%");

client.println("<br />");

if ((AN2 - nasthystereze) < nastvlhkosti) {

client.print("<font color='green'>PODMÍNKA SPLNĚNA</font>");

} else {

client.print("<font color='red'>PODMÍNKA NESPLNĚNA</font>");

}

client.println("<br />");

client.println("<br />");

client.print("Nastavená Hladina: ");

client.print(nasthladiny);

client.print("%");

client.println("<br />");

client.print("Naměřená Hladina: ");

client.print(AN1);

client.print("%");

client.println("<br />");

if (AN1 > (nasthladiny - nasthystereze)) {

client.print("<font color='green'>PODMÍNKA SPLNĚNA</font>");

} else {

client.print("<font color='red'>PODMÍNKA NESPLNĚNA</font>");

}

client.println("<br />");

client.println("<br />");

client.print("Hystereze: ");

client.print(nasthystereze);

client.print("%");

client.println("<br />");

client.print("Limit Displaye: ");

client.print(limitdisplaye);

client.print("m");

client.println("<br />");

//------------------------- zobrazeni vlhkosti a hladiny a jestli je podminka splnena nebo ne -----------------------------

break;

case 4:

client.println("<input type=button onclick='history.back()' value='Zpět'>");

client.println("<br />");

client.println("<br />");

// ------------ VYSTUPY ---------------------------------------------------------------------------------------

client.print("OUT 1 ");

if (((nastavvystupy1 == true) || ((soucasnyBehVystupy == 2) && ((kteryVystup == NULL) || (kteryVystup == 1))) || (soucasnyBehVystupy == 1) || (nastavvystupy2 == true)) || ((nastav\_vystup[0] == 1) || (nastav\_vystup[1] == 1) || (nastav\_vystup[2] == 1) || (nastav\_vystup[3] == 1) || (nastav\_vystup[4] == 1) || (nastav\_vystup[5] == 1) || (nastav\_vystup[6] == 1)) || ((nastav\_vystup[0] == 2) || (nastav\_vystup[1] == 2) || (nastav\_vystup[2] == 2) || (nastav\_vystup[3] == 2) || (nastav\_vystup[4] == 2) || (nastav\_vystup[5] == 2) || (nastav\_vystup[6] == 2)))

client.print("je <font color='green'>ZAPNUTÝ</font>");

else

client.print("je <font color='red'>VYPNUTÝ</font>");

client.println("<br />");

client.println("<br />");

client.print("OUT 2 ");

if ((nastav\_vystup[0] == 1) || ((soucasnyBehVystupy == 2) && kteryVystup == NULL) || (soucasnyBehVystupy == 1) || (nastav\_vystup[1] == 1) || (nastav\_vystup[2] == 1) || (nastav\_vystup[3] == 1) || (nastav\_vystup[4] == 1) || (nastav\_vystup[5] == 1) || (nastav\_vystup[6] == 1) || (nastavvystupy1 == true))

client.print("je <font color='green'>ZAPNUTÝ</font>");

else

client.print("je <font color='red'>VYPNUTÝ</font>");

client.println("<br />");

client.print("<FORM action='http://192.168.10.145' method='GET'>");

client.print("<P> <INPUT type='radio' name='out2' value='1'>ON");

client.print("<P> <INPUT type='radio' name='out2' value='0'>OFF");

if (nastavvystupy1)

client.print("<P> <INPUT type='submit' style='background:red;width:100px;height:50px'> </FORM>");

else

client.print("<P> <INPUT type='submit' style='background:lightgreen;width:100px;height:50px'> </FORM>");

client.print("OUT 3 ");

if ((nastav\_vystup[0] == 2) || ((soucasnyBehVystupy == 2) && kteryVystup == 1) || (soucasnyBehVystupy == 1) || (nastav\_vystup[1] == 2) || (nastav\_vystup[2] == 2) || (nastav\_vystup[3] == 2) || (nastav\_vystup[4] == 2) || (nastav\_vystup[5] == 2) || (nastav\_vystup[6] == 2) || (nastavvystupy2 == true))

client.print("je <font color='green'>ZAPNUTÝ</font>");

else

client.print("je <font color='red'>VYPNUTÝ</font>");

client.println("<br />");

client.print("<FORM action='http://192.168.10.145' method='GET'>");

client.print("<P> <INPUT type='radio' name='out3' value='1'>ON");

client.print("<P> <INPUT type='radio' name='out3' value='0'>OFF");

if (nastavvystupy2)

client.print("<P> <INPUT type='submit' style='background:red;width:100px;height:50px'> </FORM>");

else

client.print("<P> <INPUT type='submit' style='background:lightgreen;width:100px;height:50px'> </FORM>");

//--------------------------------------------------------------------------------------------------------------

break;

case 5:

//---------AKTUALNI CAS ----------------------------------------------------------------------------------------

client.println("<input type=button onclick='history.back()' value='Zpět'>");

client.println("<br />");

client.println("<br />");

cteniORC(&sekundy, &minuty, &hodiny, &denVTydnu, &denVMesici);

switch (denVTydnu) {

case 1:

client.println("Pondělí, ");

break;

case 2:

client.println("Úterý, ");

break;

case 3:

client.println("Středa, ");

break;

case 4:

client.println("Čtvrtek, ");

break;

case 5:

client.println("Pátek, ");

break;

case 6:

client.println("Sobota, ");

break;

case 7:

client.println("Neděle, ");

break;

}

client.println(hodiny);

client.println(":");

client.println(minuty);

client.println(":");

client.println(sekundy);

client.println(".");

client.println("<br />");

client.println("<h3>");

client.print("<FORM action='http://192.168.10.145' method='GET'>");

client.println("<input type='number' style='background:yellow;width:30px' name='q9' min='0' max='23'>");

client.println(" : ");

client.println("<input type='number' style='background:yellow;width:30px' name='q0' min='0' max='59'>");

// --------------------------------------------------------------------------------

client.print("<P> <INPUT type='radio' name='denVTydnu' value='1'>Pondělí");

client.print("<P> <INPUT type='radio' name='denVTydnu' value='2'>Úterý");

client.print("<P> <INPUT type='radio' name='denVTydnu' value='3'>Středa");

client.print("<P> <INPUT type='radio' name='denVTydnu' value='4'>Čtvrtek");

client.print("<P> <INPUT type='radio' name='denVTydnu' value='5'>Pátek");

client.print("<P> <INPUT type='radio' name='denVTydnu' value='6'>Sobota");

client.print("<P> <INPUT type='radio' name='denVTydnu' value='7'>Neděle");

// --------------------------------------------------------------------------------

client.print("<P> <INPUT type='submit' style='background:silver;width:100px;height:50px'> </FORM>");

for (int v = NULL; v <= pocetZnaku(buffer); v++) {

//-------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '9') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // nast hodiny 1

nastavORC[5] = ((int)buffer[v + 3] - 48);

zapisORC(00, nastavORC[6], nastavORC[5], nastavORC[7], 12);

if (vypisMozny) nastaveniNaEthernetu(2000, 1);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '9') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&')) { // nast hodiny 1 + 2

nastavORC[5] = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

zapisORC(00, nastavORC[6], nastavORC[5], nastavORC[7], 12);

if (vypisMozny) nastaveniNaEthernetu(2000, 1);

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'q') && (buffer[v + 1] == '0') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && ((buffer[v + 4] == '&') || (buffer[v + 4] == ' '))) { // nast minuty 1

nastavORC[6] = ((int)buffer[v + 3] - 48);

zapisORC(00, nastavORC[6], nastavORC[5], nastavORC[7], 12);

if (vypisMozny) nastaveniNaEthernetu(2000, 1);

}

if ((buffer[v] == 'q') && (buffer[v + 1] == '0') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&')) { // nast minuty 1 + 2

nastavORC[6] = ((10 \* ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));

zapisORC(00, nastavORC[6], nastavORC[5], nastavORC[7], 12);

if (vypisMozny) nastaveniNaEthernetu(2000, 1);

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------

if ((buffer[v] == 'd') && (buffer[v + 1] == 'n') && (buffer[v + 2] == 'u') && (buffer[v + 3] == '=') && (buffer[v + 4] != ' ')) { // nast dne

cteniORC(&sekundy, &minuty, &hodiny, &denVTydnu, &denVMesici);

nastavORC[6] = minuty;

nastavORC[5] = hodiny;

nastavORC[7] = ((int)buffer[v + 4] - 48);

zapisORC(00, nastavORC[6], nastavORC[5], nastavORC[7], 12);

if (vypisMozny) nastaveniNaEthernetu(2000, 1);

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------

}

vypisMozny = 1;

client.println("</h3>");

//--------------------------------------------------------------------------------------------------------------

break;

case 13:

client.println("<input type=button onclick='history.back()' value='Zpět'>");

client.println("<br />");

client.print("<font color='green'>Nastavení Režimu + Kalibrace</font>");

client.print("<FORM action='http://192.168.10.145' method='GET'>");

// --------------------------------------------------------------------------------

client.print("<P> <INPUT type='radio' name='rezim' value='0'>Režim NORMAL");

client.print("<P> <INPUT type='radio' name='rezim' value='1'>Režim SUCHO");

client.print("<P> <INPUT type='radio' name='rezim' value='2'>Režim VLHKO");

client.println("<br />");

client.println("<br />");

client.print("<P> <INPUT type='radio' name='kalibrace' value='0'>Kalibrovat VLHKOST");

client.print("<P> <INPUT type='radio' name='kalibrace' value='1'>Kalibrovat HLADINU");

// --------------------------------------------------------------------------------

client.print("<P> <INPUT type='submit' style='background:silver;width:100px;height:50px'> </FORM>");

for (int v = NULL; v <= pocetZnaku(buffer); v++) {

if ((buffer[v] == 'z') && (buffer[v + 1] == 'i') && (buffer[v + 2] == 'm') && (buffer[v + 3] == '=') && (buffer[v + 4] != ' ')) { // nast rezimu

rezim = ((int)buffer[v + 4] - 48);

prectiEEPROM();

nastaveniNaEthernetu(2000, 2);

}

if ((buffer[v] == 'a') && (buffer[v + 1] == 'c') && (buffer[v + 2] == 'e') && (buffer[v + 3] == '=') && (buffer[v + 4] != ' ')) { // kalibrace

int kalibrace = ((int)buffer[v + 4] - 48);

if (!kalibrace)

{

nastaveniNaEthernetu(2000, 3);

digitalWrite(A13, HIGH);

kalibraceVlhkost = analogRead(A12);

}

else

{

nastaveniNaEthernetu(2000, 4);

double dobaTrvaniPulzu = 0;

pinMode(A14, OUTPUT);

digitalWrite(A14, LOW);

delayMicroseconds(2);

digitalWrite(A14, HIGH);

delayMicroseconds(5);

digitalWrite(A14, LOW);

pinMode(A14, INPUT);

dobaTrvaniPulzu = pulseIn(A14, HIGH, 100000); // předdefinovaná fce pulseIn vrátí do proměnné dobaTrvaniPulzu výsledek měření ten pak dále převedeme.

kalibraceHladina = (dobaTrvaniPulzu / 74 / 2) \* 25; // převedeme na MILIMETRY

}

zapisEEPROM();

}

}

break;

}

client.println("</body>");

client.println("</html>");

client.stop();

}

}

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void zpozdeni(int doba)

{

bez\_znamenka long aktual;

aktual = millis();

aktual += doba;

while (millis() < aktual) {

char tlacitko = stisknutiTlacitka.getKey();

if (((!Binarnivstup1) && (!pozastavPoruchu)) || (tlacitko == 'C')) break;

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void nastaveniNaEthernetu(int doba, int ktereZobrazeni)

{

for (int w = 1; w < 3; w++) refresh[w] = true;

if (!ktereZobrazeni) {

lcd.clear();

lcd.setCursor(0, 0);

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

lcd.setCursor(7, 1);

lcd.print("ULOZENO");

lcd.setCursor(9, 2);

lcd.print("Do");

lcd.setCursor(3, 3);

lcd.print("Pameti EEPROM!");

zpozdeni(doba);

}

else if (ktereZobrazeni == 1) {

lcd.clear();

lcd.setCursor(0, 0);

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

lcd.setCursor(7, 1);

lcd.print("ULOZENO");

lcd.setCursor(9, 2);

lcd.print("Do");

lcd.setCursor(0, 3);

lcd.print("Obvodu Realneho Casu");

zpozdeni(doba);

lcd.clear();

}

else if (ktereZobrazeni == 2) {

lcd.clear();

lcd.setCursor(0, 0);

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

lcd.setCursor(3, 1);

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

lcd.setCursor(4, 2);

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

zpozdeni(doba);

lcd.clear();

}

else if (ktereZobrazeni == 3) {

lcd.clear();

lcd.setCursor(0, 1);

lcd.print("\*PROBIHA KALIBRACE!\*");

lcd.setCursor(3, 2);

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

zpozdeni(doba);

lcd.clear();

}

else if (ktereZobrazeni == 4) {

lcd.clear();

lcd.setCursor(0, 1);

lcd.print("\*PROBIHA KALIBRACE!\*");

lcd.setCursor(3, 2);

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

zpozdeni(doba);

lcd.clear();

}

vypisMozny = 0;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

int Zapouzdreni\_dat::vyberRezim()

{

x = NULL;

y = 1;

lcd.clear();

EEPROM.write(2 + rezimEEPROM, 1);

for (;;) {

ethernet();

Binarnivstup1 = nactiPort(A11);

if ((!Binarnivstup1) && (!pozastavPoruchu)) break;

if ((millis() / 1000) > (cas2 + (60 \* limitdisplaye))) lcd.noBacklight();

lcd.setCursor(0, 0);

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

lcd.setCursor(1, 1);

lcd.print("Rezim NORMAL");

lcd.setCursor(1, 2);

lcd.print("Rezim SUCHO");

lcd.setCursor(1, 3);

lcd.print("Rezim VLHKO");

lcd.setCursor(17, rezim + 1);

lcd.print("(A)");

lcd.setCursor(x, y);

lcd.write(byte(3));

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko) {

cas2 = millis() / 1000;

if (limitdisplaye != 0) lcd.backlight();

if (tlacitko == 'A') {

blokaceSystemu(); // zablokuje systém dokud se nezmáčkne F1

}

if (tlacitko == '^') {

lcd.setCursor(x, y);

lcd.print(" ");

if (y > 0) y--;

if (y == NULL) y = 3;

zpozdeni(50);

}

if (tlacitko == 'v') {

lcd.setCursor(x, y);

lcd.print(" ");

if (y < 4) y++;

if (y == 4) y = 1;

zpozdeni(50);

}

if (tlacitko == 'C') {

lcd.clear();

x = 16;

y = 1;

break;

}

if (tlacitko == 'E') {

lcd.clear();

lcd.setCursor(0, 0);

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

lcd.setCursor(3, 1);

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

lcd.setCursor(3, 2);

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

rezim = y - 1;

switch (rezim) {

case 0:

rezimEEPROM = NULL;

break;

case 1:

rezimEEPROM = 100;

break;

case 2:

rezimEEPROM = 200;

break;

}

x = 16;

y = 1;

zpozdeni(2000);

lcd.clear();

break;

}

}

}

return rezim;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::limit()

{

if (zhasnuti >= (60 \* limitdisplaye)) lcd.noBacklight();

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::kalibrace()

{

int poziceY = 1;

lcd.clear();

for (;;)

{

lcd.setCursor(0, 0);

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

lcd.setCursor(2, 1);

lcd.print("KALIBRUJ!");

lcd.setCursor(0, 2);

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

lcd.setCursor(2, 3);

lcd.print("KALIBRUJ!");

ethernet();

Binarnivstup1 = nactiPort(A11);

if ((!Binarnivstup1) && (!pozastavPoruchu)) break;

if ((millis() / 1000) > (cas2 + (60 \* limitdisplaye))) lcd.noBacklight();

lcd.setCursor(0, poziceY);

lcd.write(byte(3));

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko) {

cas2 = millis() / 1000;

if (limitdisplaye != 0) lcd.backlight();

if (tlacitko == 'A') {

blokaceSystemu(); // zablokuje systém dokud se nezmáčkne F1

}

if (tlacitko == '^') {

if (poziceY == 3) {

lcd.setCursor(0, 3);

lcd.print(" ");

poziceY = 1;

}

else if (poziceY == 1) {

lcd.setCursor(0, 1);

lcd.print(" ");

poziceY = 3;

}

zpozdeni(50);

}

if (tlacitko == 'v') {

if (poziceY == 1) {

lcd.setCursor(0, 1);

lcd.print(" ");

poziceY = 3;

}

else if (poziceY == 3) {

lcd.setCursor(0, 3);

lcd.print(" ");

poziceY = 1;

}

zpozdeni(50);

}

if (tlacitko == 'C') {

lcd.clear();

break;

}

if (tlacitko == 'E') {

lcd.clear();

lcd.setCursor(0, 1);

lcd.print("\*PROBIHA KALIBRACE!\*");

if (poziceY == 1) {

digitalWrite(A13, HIGH);

kalibraceVlhkost = analogRead(A12);

lcd.setCursor(3, 2);

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

}

if (poziceY == 3) {

lcd.setCursor(3, 2);

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

double dobaTrvaniPulzu = 0;

pinMode(A14, OUTPUT);

digitalWrite(A14, LOW);

delayMicroseconds(2);

digitalWrite(A14, HIGH);

delayMicroseconds(5);

digitalWrite(A14, LOW);

pinMode(A14, INPUT);

dobaTrvaniPulzu = pulseIn(A14, HIGH, 100000); // předdefinovaná fce pulseIn vrátí do proměnné dobaTrvaniPulzu výsledek měření ten pak dále převedeme.

kalibraceHladina = (dobaTrvaniPulzu / 74 / 2) \* 25; // převedeme na MILIMETRY

}

zapisEEPROM();

zpozdeni(2000);

break;

}

}

}

lcd.clear();

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void blokaceSystemu() {

lcd.setCursor(0, 0);

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

lcd.setCursor(0, 1);

lcd.print(" SYSTEM JE ");

lcd.setCursor(0, 2);

lcd.print(" ! BLOKOVAN ! ");

lcd.setCursor(0, 3);

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

for (int w = 0; w < 4; w++)

digitalWrite(rele[w], LOW);

while (1) { // nekonečný cyklus

char tlacitko = stisknutiTlacitka.getKey();

if (tlacitko == 'A') break;

}

lcd.clear();

for (int w = 0; w < 2; w++)

refresh[w] = true;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::nactiVstupy(int analogvstup1, int analogvstup2, int cteniVstupu1, int cteniVstupu2)

{

double dobaTrvaniPulzu = 0;

pinMode(analogvstup2, INPUT); // Analogove vstupy

pinMode(cteniVstupu2, OUTPUT); // Tyto výstupy se přepnou do log. "1" pouze při čtení vstupů,pak se nastaví zpět na log. "0".

digitalWrite(cteniVstupu2, HIGH); // na snímače přivede +5V

pinMode(analogvstup1, OUTPUT);

digitalWrite(analogvstup1, LOW);

delayMicroseconds(2);

digitalWrite(analogvstup1, HIGH);

delayMicroseconds(5);

digitalWrite(analogvstup1, LOW);

pinMode(analogvstup1, INPUT);

dobaTrvaniPulzu = pulseIn(analogvstup1, HIGH, 100000); // předdefinovaná fce pulseIn vrátí do proměnné dobaTrvaniPulzu výsledek měření ten pak dále převedeme.

AN1 = (dobaTrvaniPulzu / 74 / 2) \* 25; // převedeme na MILIMETRY

AN1 = map(AN1, 161.5, kalibraceHladina, 0, 100); // předdefinovaná fce map rozdělí 1. parametr v rozmezí od 2. parametru do 3. parametru na 0-100%.

AN2 = analogRead(analogvstup2);

AN2 = map((AN2 / 3.4), 0, kalibraceVlhkost, 0, 100);

digitalWrite(cteniVstupu2, LOW); // a zase přivede 0V

if (AN1 > 100) AN1 = 100; // pokud je vetsi nez 100% nastavi se na 100%.

else if (AN1 < 0) AN1 = 0; // pokud je mensi nez 0% nastavi se na 0%.

if (AN2 > 100) AN2 = 100;

else if (AN2 <= 5) AN2 = 0;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void Zapouzdreni\_dat::prace\_s\_vystupy()

{

cteniORC(&sekundy, &minuty, &hodiny, &denVTydnu, &denVMesici);

for (i = NULL; i < 7; i++) {

if ((nastav\_vystup[i] == 1) || (nastav\_vystup[i] == 2)) {

nastavvystupy1 = false;

nastavvystupy2 = false;

}

}

if ((AN2 > (nastvlhkosti + nasthystereze)) || ((AN1 + nasthystereze) < nasthladiny)) {

for (i = NULL; i <= 6; i++) {

if ((nastav\_vystup[i] = 1) || (nastav\_vystup[i] == 2) || (soucasnyBehVystupy == 1) || ((soucasnyBehVystupy == 2) && ((kteryVystup == NULL) || (kteryVystup == 1)))) {

if (spustVystupy) minuty\_osetreniVystupu = minuty + 1;

spustVystupy = false;

soucasnyBehVystupy = NULL;

nastav\_vystup[i] = NULL;

}

}

}

if (!spustVystupy) {

if (minuty\_osetreniVystupu > minuty)

nactiVstupy(A14, A12, A15, A13);

else if (minuty\_osetreniVystupu == minuty)

spustVystupy = true;

}

for (i = NULL; i <= 6; i++) {

if ((hodiny == dny[i].hodinycelk) && (minuty == (dny[i].minutycelk - 1)) && (denVTydnu == (i + 1))) { //nacteni vstupu minutu pred spuštěním výstupů

nactiVstupy(A14, A12, A15, A13);

}

}

if (!soucasnyBeh) {

for (i = NULL; i <= 6; i++) {

if ((hodiny == dny[i].hodinycelk) && (minuty == dny[i].minutycelk) && (nastav\_vystup[i] == NULL) && ((AN2 - nasthystereze) < nastvlhkosti) && (AN1 > (nasthladiny - nasthystereze)) && (denVTydnu == (i + 1)) && ((dny[0].min1celk > 0) || (dny[0].min2celk > 0)) && (spustVystupy)) {

nastav\_vystup[i] = 1;

if ((dny[i].min2celk > 0) && (dny[i].min1celk == NULL)) nastav\_vystup[i] = 2;

cas\_vystupu = millis() / 1000;

}

if (nastav\_vystup[i] == 1) {

if ((millis() / 1000) > (cas\_vystupu + (dny[i].min1celk \* 60))) {

cas\_vystupu = millis() / 1000;

nastav\_vystup[i] = 2;

}

}

if (nastav\_vystup[i] == 2) {

if (dny[i].min2celk > 0) {

if ((millis() / 1000) > (cas\_vystupu + (dny[i].min2celk \* 60))) {

nastav\_vystup[i] = NULL;

cas\_vystupu = NULL;

}

}

else nastav\_vystup[i] = NULL;

}

}

}

else {

for (i = NULL; i <= 6; i++) {

if ((hodiny == dny[i].hodinycelk) && (minuty == dny[i].minutycelk) && (!soucasnyBehVystupy) && ((AN2 - nasthystereze) < nastvlhkosti) && (AN1 > (nasthladiny - nasthystereze)) && (denVTydnu == (i + 1)) && ((dny[0].min1celk > 0) || (dny[0].min2celk > 0)) && (spustVystupy)) {

if (!soucasnyBehVystupy) {

cas\_vystupu = millis() / 1000;

soucasnyBehVystupy = 1;

}

identifik = i;

}

if ((soucasnyBehVystupy == 1) && (i == identifik)) {

if ((millis() / 1000) > (cas\_vystupu + oba\_vystupy)) {

soucasnyBehVystupy = 2;

cas\_vystupu = millis() / 1000;

}

}

if ((soucasnyBehVystupy == 2) && (i == identifik)) {

if ((millis() / 1000) > (cas\_vystupu + druhy\_vystup)) {

soucasnyBehVystupy = NULL;

cas\_vystupu = millis() / 1000;

}

}

if ((dny[i].min1celk \* 60) > (dny[i].min2celk \* 60) && (i == identifik)) {

druhy\_vystup = (dny[i].min1celk \* 60) - (dny[i].min2celk \* 60);

oba\_vystupy = (dny[i].min2celk \* 60);

kteryVystup = NULL;

}

else if ((dny[i].min2celk \* 60) > (dny[i].min1celk \* 60) && (i == identifik)) {

druhy\_vystup = (dny[i].min2celk \* 60) - (dny[i].min1celk \* 60);

oba\_vystupy = (dny[i].min1celk \* 60);

kteryVystup = 1;

}

else if ((dny[i].min2celk \* 60) == (dny[i].min1celk \* 60) && (i == identifik)) {

druhy\_vystup = NULL;

oba\_vystupy = (dny[i].min1celk \* 60);

kteryVystup = 2;

}

}

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

void zkontrolujEEPROM()

{

int kontrolaEEPROM;

for (int vycistiEEPROM = NULL; vycistiEEPROM < 1000; vycistiEEPROM++) {

kontrolaEEPROM = EEPROM.read(vycistiEEPROM); // čtení z EEPROM

if (kontrolaEEPROM == 255) { // Tam,kde je v EEPROM zapsano 255,to přepíše nulou

EEPROM.write(vycistiEEPROM, 0);

}

}

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

bool nactiPort(byte port) {

int vstup = NULL;

pinMode(port, INPUT);

vstup = analogRead(port);

vstup = map(vstup, 0, 1023, 0, 100);

if (vstup > 25) return 0;

else return 1;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

int pocetZnaku(String &retezec) {

int pocet = NULL;

for (; retezec.c\_str()[pocet] != '\0'; )

pocet++;

return pocet;

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

Zapouzdreni\_dat::Zapouzdreni\_dat()

{ // KONSTRUKTOR TŘÍDY

zkontrolujEEPROM();

prectiEEPROM();

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------

Zapouzdreni\_dat::~Zapouzdreni\_dat()

{ // DESTRUKTOR TŘÍDY

}

//------------------------------------------------------------------------------------------------------------------------------------------------------------------