

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

Zadání:

Vytvoř **Teploměr:**

Obecné info: - teplotní čidlo DS18B20

1) Test čidla

- obsah registru LSB zobrazit na LEDkách

2) Test LCD

- vypis textu na 2 řádky

3) Zobrazit aktuální teplotu na LCD

- 4 desetinné místa

4) Zapamatovat si min. a max. teplotu

- zobrazovat průběžně na seriovém monitoru

5) Hlavní program:

- při poklesu pod 26,5°C zapnout topení
- při překročení 28,5°C zapnout klimatizaci
- vždy vypsát příslušný text na LCD
- při překročení nebo poklesu rozsvítit příslušnou barvu na RGB
(klid = zelená, topení = červená, klimatizace = modrá)

Rozšíření:

- pomocí tlačítka přepínat na LCD mezi aktuální teplotou, minimální a maximální
- pomocí druhého tlačítka možnost smazání paměti o min a max hodnotách

Teoretický rozbor:

Použitý přípravek: Arduino Uno

Mozkem této desky je 8bitový mikrokontrolér ATmega328P o flash paměti 32 kB, paměti SRAM 2 kB a paměti EEPROM 1 kB.

Arduino UNO obsahuje 14 digitálních pinů, všechny mohou být použity jako vstup či výstup užitím funkcí pinMode(), digitalWrite() a digitalRead().

Maximální dovolený proud těmito svorkami je 20 mA, pokud proud dosáhne dvojnásobku tohoto proudu, může dojít k poškození vývojového kitu.

Dále obsahuje 6 analogových vstupů o šířce 10 bitů, ty disponují měřicím rozsahem 5 V. Toto napětí lze upravit užitím funkce analogReference() na pin AREF. Přesný krystal generuje pro celý obvod hodinový signál 16 MHz. Arduino UNO disponuje USB konektorem pro komunikaci s PC, piny umožňující pulzně šířkovou modulaci, napájecí konektor, resetovací tlačítko atp.

Teplotní čidlo DS18B20

Čidlo DS18B20 je digitální teploměr s kalibrovanou přesností $\pm 0,5\text{ }^{\circ}\text{C}$ a rozlišením až 12 bitů. Informace o teplotě se z čidla do Arduina přenáší digitálně, nedochází tak ke zkreslení naměřené hodnoty ani změně v důsledku rušení. DS18B20 využívá jednoduché komunikační rozhraní 1-Wire, které pro přenos dat potřebuje pouze jeden vodič. Teplotní rozsah je $-55 \div +125\text{ }^{\circ}\text{C}$ a rozlišovací schopnost je až $\pm 0,0625\text{ }^{\circ}\text{C}$. Teplotní čidlo zvládne také „alarm“ funkci s volitelnou horní a spodní úrovní teploty.

Teplotní čidlo DS18B20 má pro adresaci jedinečný 64-bitový kód. To spolu s vlastnostmi sběrnice 1-Wire umožňuje použít v Arduino projektu až 50 teplotních čidel pospojovaných za sebou (paralelně) a komunikovat s nimi.

Displej I2C (16x2)

Displej je vybaven převodníkem na sběrnici I2C, který je zapojen pouhými čtyřmi piny. Dvěma napájecími a dvěma datovými. Tím je zredukován počet použitých výstupů z kitu, aby jich zbylo dostatečné množství pro další periferie. Displej je napájen napětím 5 V.

Obsahuje 3 paměti:

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

Výpočty:

Odpory pro rgb: $3 \times 220\Omega$

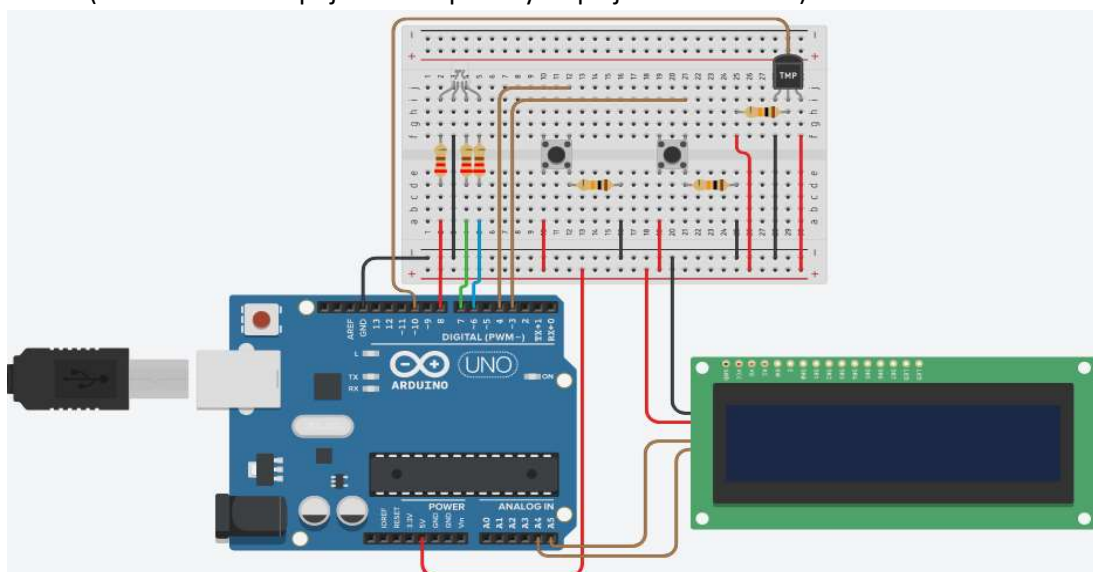
Pro teplotní čidlo se používá $4,7\text{k}\Omega$ pull-up rezistor, dle datasheetu

Pro tlačítka používám $10\text{k}\Omega$ pull-down rezistory

Pokud nemáme displej se sběrnici I2C, je nutné použít $10\text{k}\Omega$ potenciometr.

Schéma zapojení

(Ve skutečném zapojení mám použitý displej se sběrnici I2C)



Zdrojový kód:

```
#include <Wire.h> //knihovny nutné pro správnou funkci programu
#include <LiquidCrystal_I2C.h> // Library for LCD
#include <OneWire.h>
#include <DallasTemperature.h>
OneWire oneWire(10);
DallasTemperature sensors(&oneWire);

// deklarace proměnných
int y=0, x=0, stavtlac1=0, tlac1=4, stavtlac2=0, tlac2=3, cer=8, zel=7, mod=6;
float tep, min=100, max=-100;
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2); //deklarace lcd displeje

void setup() {
  Serial.begin(9600); //inicializace
  lcd.init();
  lcd.backlight();
  pinMode(tlac1, INPUT);
  pinMode(tlac2, INPUT);
}

void loop()
{
  //hlavní program ve funkcích
  cteni();
  porovnavani();
  vypis();
}

void cteni(){ //funkce sloužící pro čtení naměřených hodnot
  sensors.requestTemperatures();
  tep = sensors.getTempCByIndex(0);
}

//funkce sloužící pro obsluhu RGB LEDy
void RGB(int cer_hod, int zel_hod, int mod_hod)
{
  analogWrite(cer, cer_hod);
  analogWrite(zel, zel_hod);
  analogWrite(mod, mod_hod);
}

void mod_alarm() //funkce sloužící pro přerušování modrého alarmu
{
  digitalWrite(mod, HIGH);
  delay(300);
  digitalWrite(mod, LOW);
  delay(100);
}

void vypis() //funkce sloužící pro výpis naměřených hodnot na displej
{
  stavtlac1 = digitalRead(tlac1);
```

```

stavtlac2 = digitalRead(tlac2);

if(stavtlac1 == HIGH)
{y++; if(y>=3){y=0;}}

if(y==0){
    lcd.setCursor(0, 0);
    lcd.print("Tep: ");
    lcd.print(tep);
    lcd.print((char)223);
    lcd.print("C   ");}

if(y==1){
    lcd.setCursor(0, 0);
    lcd.print("Max tep: ");
    lcd.print(max);
    lcd.print((char)223);
    lcd.print("C   ");}

if(y==2){
    lcd.setCursor(0, 0);
    lcd.print("Min tep: ");
    lcd.print(min);
    lcd.print((char)223);
    lcd.print("C   ");}

if(stavtlac2 == HIGH)
{min=100; max=-100;}

if(x==5){
    Serial.print("\n\nAktualni teplota: ");
    Serial.print(tep);
    Serial.print("\nMaximalni teplota: ");
    Serial.print(max);
    Serial.print("\nMinimalni teplota: ");
    Serial.print(min);x=0;}
x++;
}

void porovnavani()
{
    if(tep<min){min=tep;}
    if(tep>max){max=tep;}

    lcd.setCursor(0, 1);
    if (tep>=26.5)
    { if (tep>28.5)
        {if(tep>31){mod_alarm();}}
    }
}

```

//výpis teploty na displej

//při zmáčknutí tlačítka se na displeji
//objeví výpis maximální naměřené teploty

//při opětovném zmáčknutí tlačítka se na displeji
// objeví výpis minimální naměřené teploty

//při zmáčknutí druhého tlačítka vymažeme
//max. a min. hodnoty uložené v paměti

//výpis na seriový monitor

//funkce sloužící pro porovnávání teploty
//se zadanými parametry

//porovnávání teplot
//ledka bliká modře

```

else //ledka se rozsvítí modře, výpis na lcd
  {RGB(0, 0, 255); lcd.print("Chlazení zapnuto"); }
else //ledka se rozsvítí zeleně, výpis na lcd
  {RGB(0, 255, 0); lcd.print("Teplota je OK "); }
else //ledka se rozsvítí červeně, výpis na lcd
  {RGB(255, 0, 0); lcd.print("Topení zapnuto "); }

```

Závěr:

Úkolem bylo vytvořit funkční teploměr, který bude své hodnoty ukládat do paměti a vypisovat je na lcd displej.

Pomocí RGB ledky znázorňujeme ohřev, klidový stav nebo klimatizaci.

Na lcd displeji vypisujeme, pokud je teplota nízká-zapínáme topení, pokud je teplota správná-neděje se nic a pokud je teplota vysoká, zapína se klimatizace

Úloha mi nepřípadala těžká, ale časově namáhavá.

Nejnáročnější bylo sehnat příslušné knihovny a příkazy pro zjišťování teplot z teploměru.

Dostí zrádná je i orientace v programu, pokud si v něm neuděláme pořádek, těžko se v něm vyzná.

Možnost použití vidím v domácnostech jako jednoduchý teploměr, vybavený dvěma tlačítky pro přepínání režimů obrazovky.

Po doplnění ovladačího prvku např. klávesnice a spojení s teplotním okruhem budovy by bylo možno regulovat teplotu v domácnostech. Díky klávesnici můžeme měnit teplotu, při které se zapíná topení nebo klimatizace.

Přípravek by mohl být použit i ve firmách a továrnách, díky tomu, že čidlo má pro adresaci jedinečný 64-bitový kód. To spolu s vlastnostmi sběrnice 1-Wire umožňuje použít v Arduino projektu až 50 teplotních čidel pospojovaných za sebou (paralelně) a komunikovat s nimi. Mohli bychom měřit teploty na více místech, a tak přesněji regulovat teplotu celého firemního komplexu.

