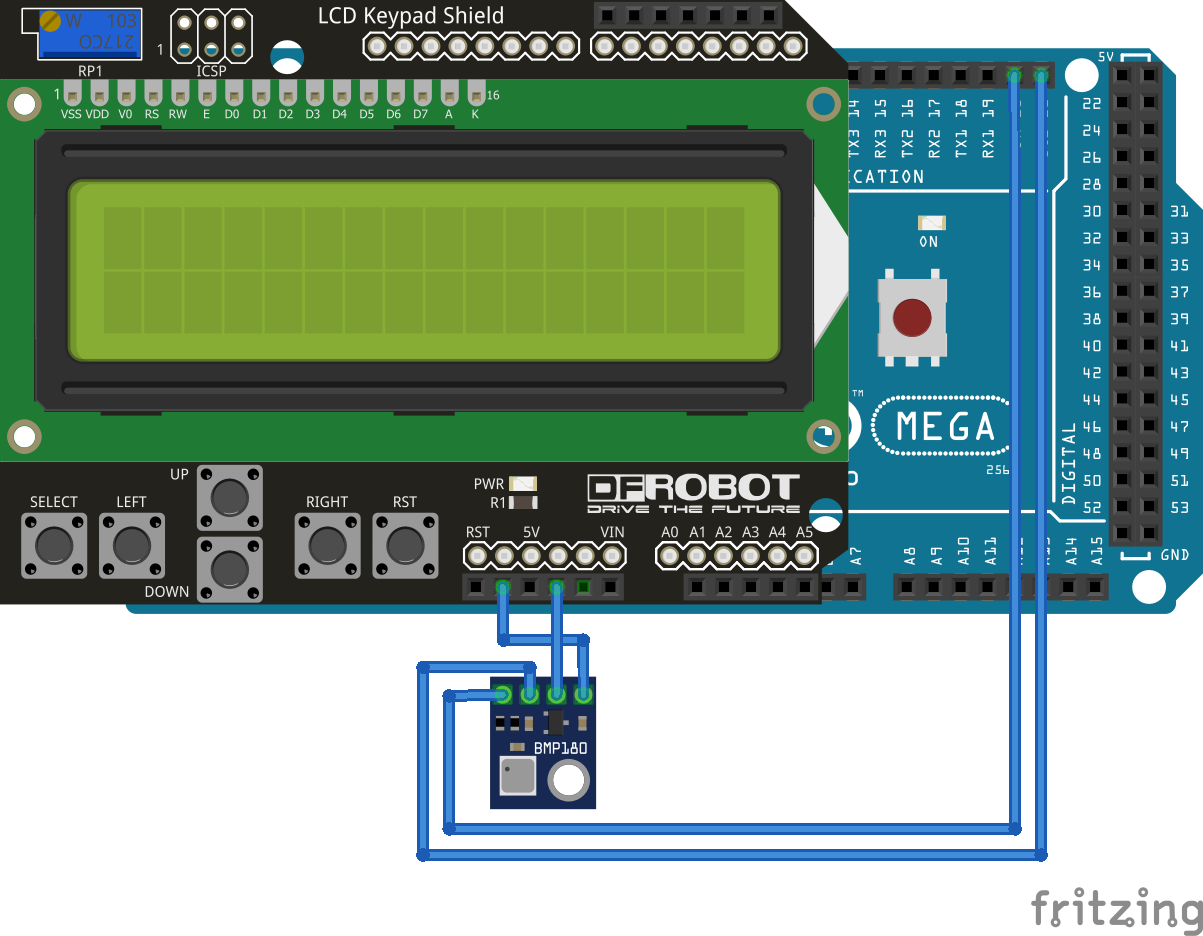
**ZOP - VAJA 7**

**1. KOSOVNICA:**

* Arduino Mega 2560
* LCD Keypad Shield
* GY-68 modul
* 4 M-F žici
* USB kabel

**2. VEZALNA SHEMA:**

****

**3. KODA:**

#include <Wire.h>

#include <**LiquidCrystal**.h>

#define ADDRESS\_SENSOR 0x77

#define tipka A0

int zasloni=1;

int tipka\_desno;

int tipka\_levo;

byte stopinja[8] = {

B00010,

B00101,

B00010,

B00000,

B00000,

B00000,

B00000,

};

const int rs = 8, en = 9, d4 = 4, d5 = 5, d6 = 6, d7 = 7;

**LiquidCrystal** lcd(rs, en, d4, d5, d6, d7);

int16\_t ac1, ac2, ac3, b1, b2, mb, mc, md;

uint16\_t ac4, ac5, ac6;

const uint8\_t oss = 3;

const uint8\_t osd = 26;

float T;

float P;

float h;

void setup() {

**Serial**.begin(9600);

Wire.begin();

lcd.begin(16, 2);

inicializacija\_senzorja();

}

void loop() {

int32\_t b5;

int32\_t p;

if(analogRead(tipka) == 0 && tipka\_desno == 0){

tipka\_desno = 1;

zasloni++;

if(zasloni == 4){

zasloni = 1;

}

}

if(analogRead(tipka) <= 500 && analogRead(tipka) >= 480 && tipka\_levo == 0){

tipka\_levo = 1;

zasloni--;

if(zasloni == 0){

zasloni = 3;

}

}

if(analogRead(tipka) == 1023){

tipka\_desno = 0;

tipka\_levo = 0;

}

b5 = temperatura();

**Serial**.print("Temperatura = ");

**Serial.**print(T);

**Serial**.println(" °C");

P = zracni\_tlak(b5);

**Serial**.print("Zračni tlak = ");

**Serial**.print(P);

**Serial**.println(" Pa");

h = nadmorska\_visina(P);

**Serial**.print("Nadmorska višina = ");

**Serial**.print(h);

**Serial**.println(" m");

**Serial**.println("");

if(zasloni == 1){

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Zracni tlak = ");

lcd.setCursor(0, 1);

lcd.print((P / 100), 0);

lcd.print(" kPa");

}

if(zasloni == 2){

lcd.clear();

lcd.createChar(0, stopinja);

lcd.setCursor(0, 0);

lcd.print("Temperatura = ");

lcd.setCursor(0, 1);

lcd.print(T, 1);

lcd.print(" ");

lcd.write(byte(0));

lcd.print("C");

}

if(zasloni == 3){

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Nadm. visina = ");

lcd.setCursor(0, 1);

lcd.print(h, 0);

lcd.print(" m");

}

delay(250);

}

//////////////////////////////////////////

//Inicializacija senzorskih spremenljivk//

//////////////////////////////////////////

void inicializacija\_senzorja()

{

ac1=read\_2\_bytes(0xAA);

ac2=read\_2\_bytes(0xAC);

ac3=read\_2\_bytes(0xAE);

ac4=read\_2\_bytes(0xB0);

ac5=read\_2\_bytes(0xB2);

ac6=read\_2\_bytes(0xB4);

b1=read\_2\_bytes(0xB6);

b2=read\_2\_bytes(0xB8);

mb=read\_2\_bytes(0xBA);

mc=read\_2\_bytes(0xBC);

md=read\_2\_bytes(0xBE);

**Serial**.println("Inicializacija senzorskih spremenljivk :");

**Serial**.print(F("AC1 = ")); **Serial**.println(ac1);

**Serial**.print(F("AC2 = ")); **Serial**.println(ac2);

**Serial**.print(F("AC3 = ")); **Serial**.println(ac3);

**Serial**.print(F("AC4 = ")); **Serial**.println(ac4);

**Serial**.print(F("AC5 = ")); **Serial**.println(ac5);

**Serial**.print(F("AC6 = ")); **Serial**.println(ac6);

**Serial**.print(F("B1 = ")); **Serial**.println(b1);

**Serial**.print(F("B2 = ")); **Serial**.println(b2);

**Serial**.print(F("MB = ")); **Serial**.println(mb);

**Serial**.print(F("MC = ")); **Serial**.println(mc);

**Serial**.print(F("MD = ")); **Serial**.println(md);

**Serial**.println("");

}

///////////////////////////////////

//Odčitavanje 2 bytov iz senzorja//

///////////////////////////////////

uint16\_t read\_2\_bytes(uint8\_t code) //Funkcija za branje 16 bitnih registrov

{

uint16\_t value;

Wire.beginTransmission(ADDRESS\_SENSOR);

Wire.write(code);

Wire.endTransmission();

Wire.requestFrom(ADDRESS\_SENSOR, 2);

if(Wire.available() >= 2)

{

value = (Wire.read() << 8) | Wire.read();

}

return value;

}

/////////////////////////////

//Kalkulacija temperature//

/////////////////////////////

int32\_t temperatura()

{

int32\_t x1, x2, b5, UT;

Wire.beginTransmission(ADDRESS\_SENSOR);

Wire.write(0xf4);

Wire.write(0x2e);

Wire.endTransmission();

delay(5);

UT = read\_2\_bytes(0xf6);

x1 = (UT - (int32\_t)ac6) \* (int32\_t)ac5 >> 15;

x2 = ((int32\_t)mc << 11) / (x1 + (int32\_t)md);

b5 = x1 + x2;

T = (b5 + 8) >> 4;

T = T / 10.0;

return b5;

}

//////////////////////////////

//Kalkulacija zračnega tlaka//

///////////////////////////////

int32\_t zracni\_tlak(int32\_t b5){

int32\_t x1, x2, x3, b3, b6, p, UP;

uint32\_t b4, b7;

Wire.beginTransmission(ADDRESS\_SENSOR);

Wire.write(0xf4);

Wire.write(0x34 + (oss << 6));

Wire.endTransmission();

delay(osd);

Wire.beginTransmission(ADDRESS\_SENSOR);

Wire.write(0xf6);

Wire.endTransmission();

Wire.requestFrom(ADDRESS\_SENSOR, 3);

if(Wire.available() >= 3)

{

UP = (((int32\_t)Wire.read() << 16) | ((int32\_t)Wire.read() << 8) | ((int32\_t)Wire.read())) >> (8 - oss);

}

b6 = b5 - 4000;

x1 = (b2 \* (b6 \* b6 >> 12)) >> 11;

x2 = ac2 \* b6 >> 11;

x3 = x1 + x2;

b3 = (((ac1 \* 4 + x3) << oss) + 2) >> 2;

x1 = ac3 \* b6 >> 13;

x2 = (b1 \* (b6 \* b6 >> 12)) >> 16;

x3 = ((x1 + x2) + 2) >> 2;

b4 = (ac4 \* (uint32\_t)(x3 + 32768)) >> 15;

b7 = ((uint32\_t)UP - b3) \* (50000 >> oss);

if(b7 < 0x80000000) {

p = (b7 \* 2) / b4;

}

else {

p = (b7 / b4) \* 2;

}

x1 = (p >> 8) \* (p >> 8);

x1 = (x1 \* 3038) >> 16;

x2 = (-7357 \* p) >> 16;

p = p + ((x1 + x2 + 3791) >> 4);

return p;

}

/////////////////////////////////

//Kalkulacija nadmorske višine//

///////////////////////////////////

float nadmorska\_visina(float P){

float h = 44330 \* (1 - pow((P / 101325),(1 / 5.2558797)));

return h;

}

**4. KOMENTAR:**

Za delovanje ali kalibracijo senzorja mi ni bilo potrebno izračunati nobene druge komponente. Tudi kalibracija ni bila potrebna, saj senzor ne meri tlaka in temperature direktno. To izračunamo s pomočjo arduina. Natančnost temperature je več kot spremenljiva, absolutna razlika z mojo domačo vremensko postajo se je gibala okoli 0,2 °C. Istega pa ne morem reči za zračni tlak. Naprave, ki bi merila zračni tlak doma nimam, a ko sem s pomočjo zračnega tlaka izračunal nadmorsko višino se je pojavila napaka od 100 do 150 m. Za to napako ne vem kaj bi bil točen razlog, možna je okvara zaradi napačnega napajanja senzorja, saj sem ga sprva napajal s 5 V, kot je pisalo na nekaterih spletnih straneh, na drugih pa je opozarjalo, da se ga striktno ne sme napajati z več kot 3,3 V, za katero sem se pozneje odločil. Zato sem naročil še dva takšna senzorja, da bom videl, kje se pojavi ta napaka. Kar se pa tiče programa, ni tako preprost. Na izbiro sem imel z kodo knjižnicami za BMP180 senzor, a sem se zaradi boljšega razumevanja in prilagajanja programa meni odločil, da bom napisal program brez knjižnice. Seveda se nisem programa spravil napisati sam, a sem našel osnovni primer in ga nadgradil. To sem moral narediti, saj moj senzor uporablja I2C povezavo, ki je meni neznana. Za preračunavanje fizikalnih količin pa mi je prav prišel datasheet za izbrani senzor in formula za nadmorsko višino.