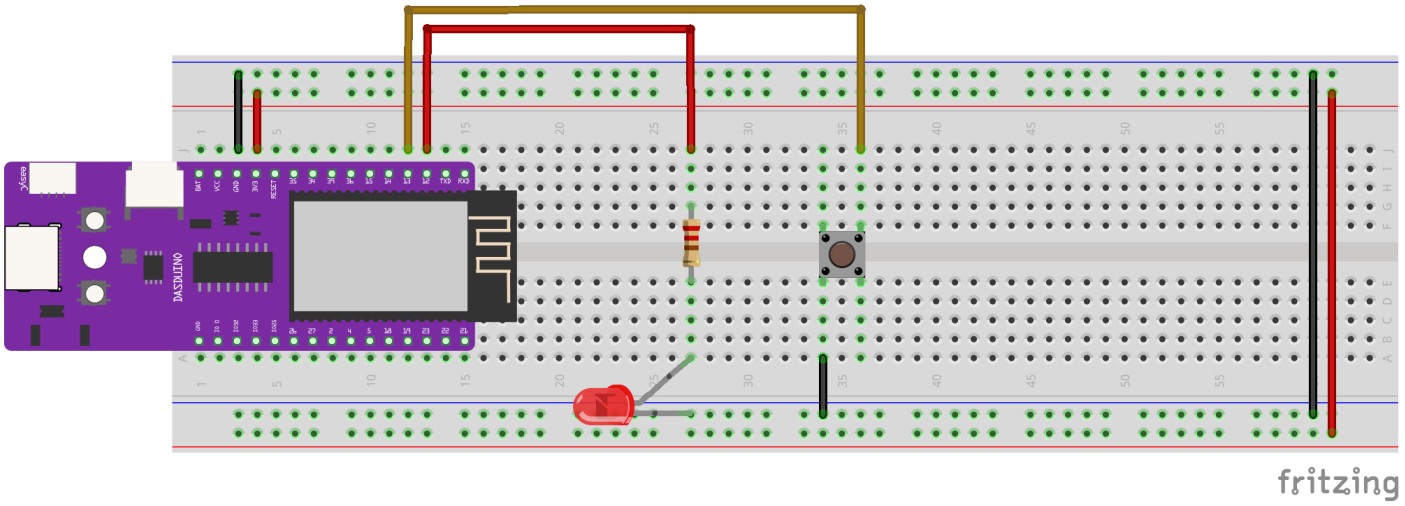
Tipkalo i svjetleća dioda

Zadatak: Na Arduino spojite jedno tipkalo i jednu svjetleću diodu. Napišite program kojim ćete upravljati svjetlećom diodom na sljedeći način – kada je tipkalo pritisnuto neka dioda bude uključena, a kada je tipkalo otpušteno neka dioda bude isključena.

Cilj zadatka je upoznati način rada Dasduino izvoda u ulaznom režimu rada. Tipkalo ćemo na Dasuino, kao i na Arduino spajati tako da jedna strana tipkala bude spojena na odabrani izvod Dasduina (u ovom primjeru odabran je izvod 12 no možete odabrati i neki drugi) a druga strana bude spojena na izvod GND. Na taj način izvod na kojem je spojeno tipkalo bit će u stanju 0 (LOW) kada je tipkalo pritisnuto odnosno u stanju 1 (HIGH) kada je tipkalo otpušteno.

Prikaz spajanja



Programski kod rješenja

```
int LedCrvena = 12;
int Tipkalo = 13;

int StanjeTipkala;

void setup() {
  pinMode(LedCrvena, OUTPUT);     //postavi izvod LedCrvena (6) kao izlazni
  pinMode(Tipkalo, INPUT_PULLUP); //postavi izvod Tipkalo (10) kao ulazni, pull-up
}

void loop() {
  StanjeTipkala = digitalRead(Tipkalo); //očitaj stanje izvoda i pohrani u StanjeTipkala
  if (StanjeTipkala == LOW) {        //ako je tipkalo pritisnuto
    digitalWrite(LedCrvena, HIGH);    //uključi svjetleću diodu
  } else {                           //inače
    digitalWrite(LedCrvena, LOW);     //isključi svjetleću diodu
  }
}
```

Na samom početku definirane su varijable LedCrvena i Tipkalo koje smo koristili da bi u programskom kodu na jednostavniji način imenovali izvode na koje su spojene pojedine komponente. Koristeći **pinMode**

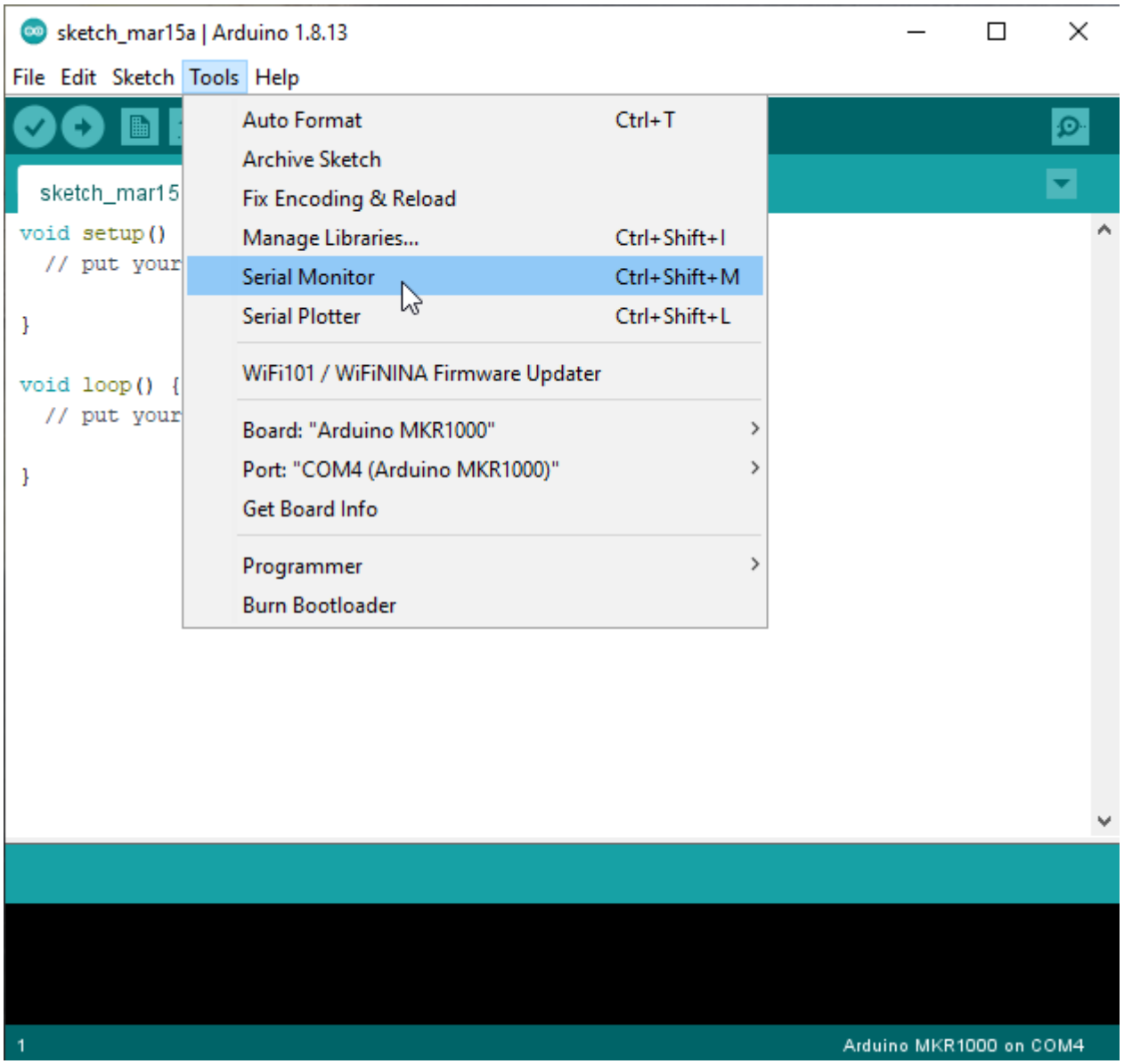
napredbu postavili smo režim rada pojedinog izvoda - za svjetleću diodu izlazni režim rada (**OUTPUT**) a za tipkalo ulazni režim rada s aktivacijom pull-up otpornika (**INPUT_PULLUP**). Naredbom **digitalRead** očitavamo stanje tipkala i pomoću **if** naredbe i **digitalWrite** naredbi uključujemo i isključujemo svjetleću diodu.

Serijska veza s računalom

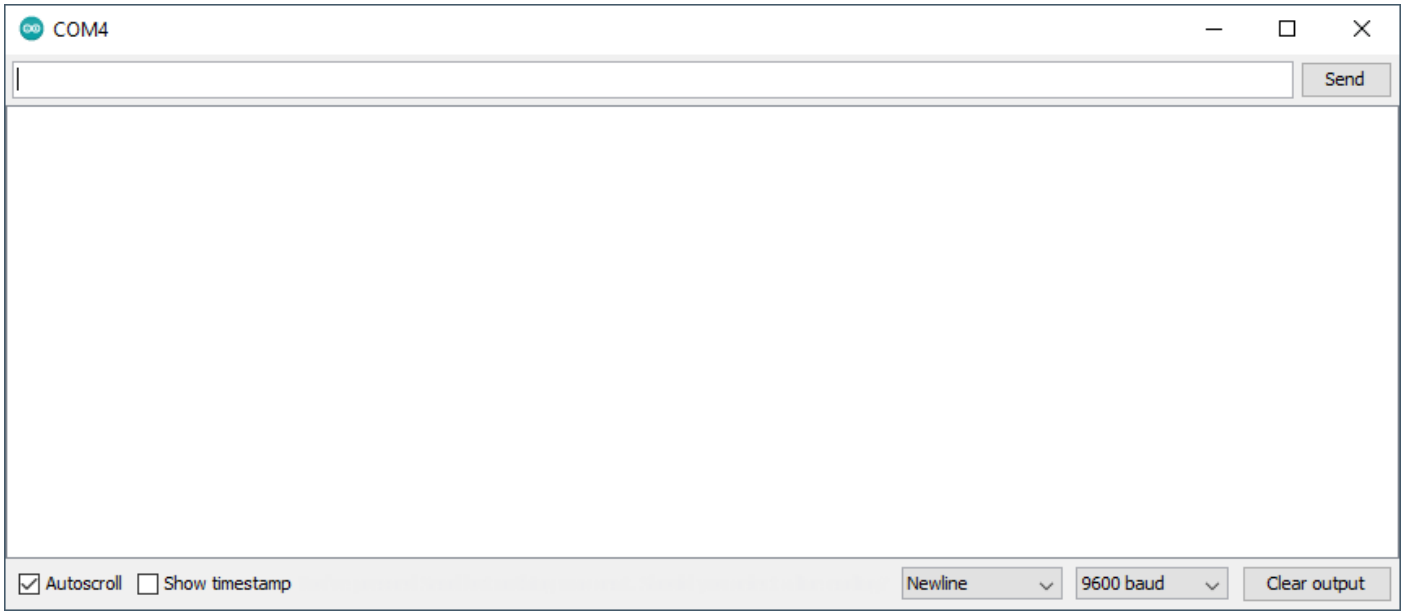
Uvod

Serial Monitor je zaseban pop-up prozor koji djeluje kao terminal tj. komunicira primanjem i slanjem serijskih podataka između računala i Arduina MKR1000. Serijski podaci mogu se obostrano slati preko USB-a,

Prilikom korištenja serial monitora bitno je da je Arduino preko USB-a priključen na računalo. Kako biste otvorili serijski monitor potrebno je odabrati Tools i zatim Serial monitor koji će se pojaviti kao pop-up prozor.



U gornjem dijelu serijskog monitora možete slati naredbe, dok donja veća površina služi za prikaz dobivenih podataka sa Arduina. U desnom donjem uglu može se mijenjati brzina slanja ili primanja podataka – kod Arduina je defaultna vrijednost od „9600 baud“. **Vi ju postavite na 115200**.



Zadatak: Nadogradite prethodni program kako bi se stanje tipkala ispisivalo serijskom vezom na računalu.

Prikaz spajanja
Shema spajanja ostaje nepromjenjena.

Programski kod rješenja

```
int LedCrvena = 12;
int Tipkalo = 13;

int StanjeTipkala;

void setup() {
  pinMode(LedCrvena, OUTPUT);     //postavi izvod LedCrvena (6) kao izlazni
  pinMode(Tipkalo, INPUT_PULLUP); //postavi izvod Tipkalo (10) kao ulazni, pull-up
  Serial.begin(115200);
}

void loop() {
  StanjeTipkala = digitalRead(Tipkalo); //očitaj stanje izvoda i pohrani u StanjeTipkala
  if (StanjeTipkala == LOW) {        //ako je tipkalo pritisnuto
    digitalWrite(LedCrvena, HIGH);    //uključi svjetleću diodu
    Serial.println("Tipkalo je pritisnuto");
  } else {                           //inače
    digitalWrite(LedCrvena, LOW);     //isključi svjetleću diodu
    Serial.println("Tipkalo je otpušteno");
  }
}
```

U **setup** dijelu programa potrebno je inicijalizirati serijsku vezu s računalom koristeći **Serial.begin** naredbu. Unutar te naredbe definira se brzina komunikacije s računalom koju smo postavili na 115200. Loop dio programa je vrslo sličan kao u prethodnom primjeru, uz dodatak naredbi **Serial.println** koji omogućavaju da se serijskim putem na računalo pošalje poruka.

easyC OLED ekran

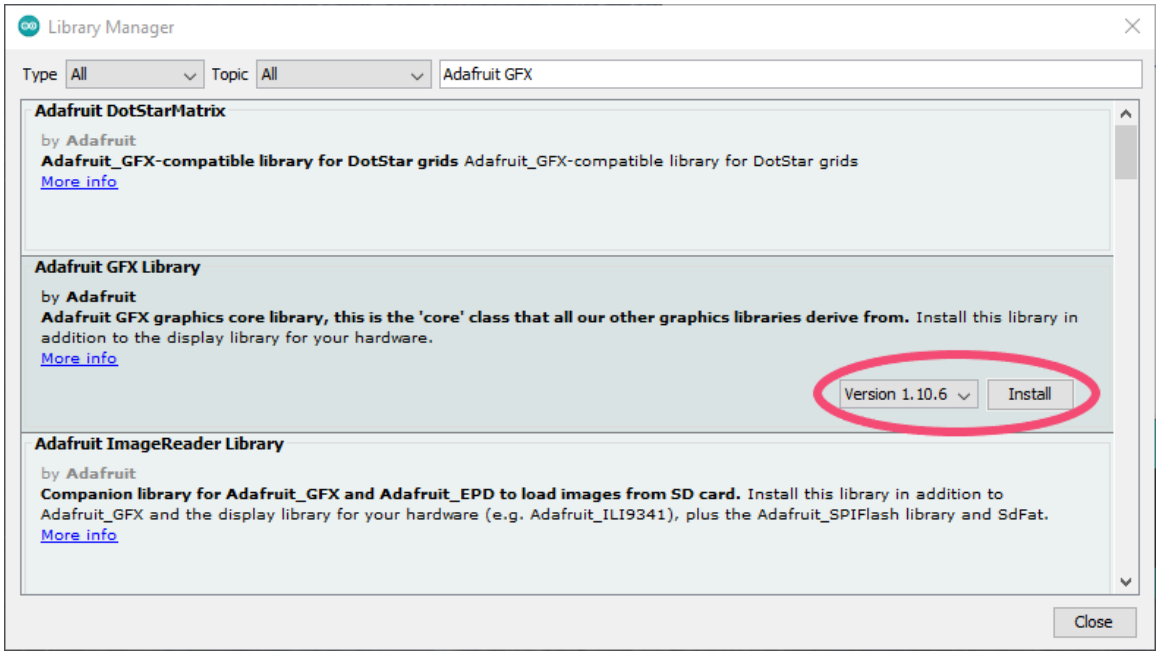
Zadatak: Spojite OLED ekran koristeći easyC priključak na Dasduino te izradite program kojim ćete na ekranu ispisati poruku “Hello, world!”.

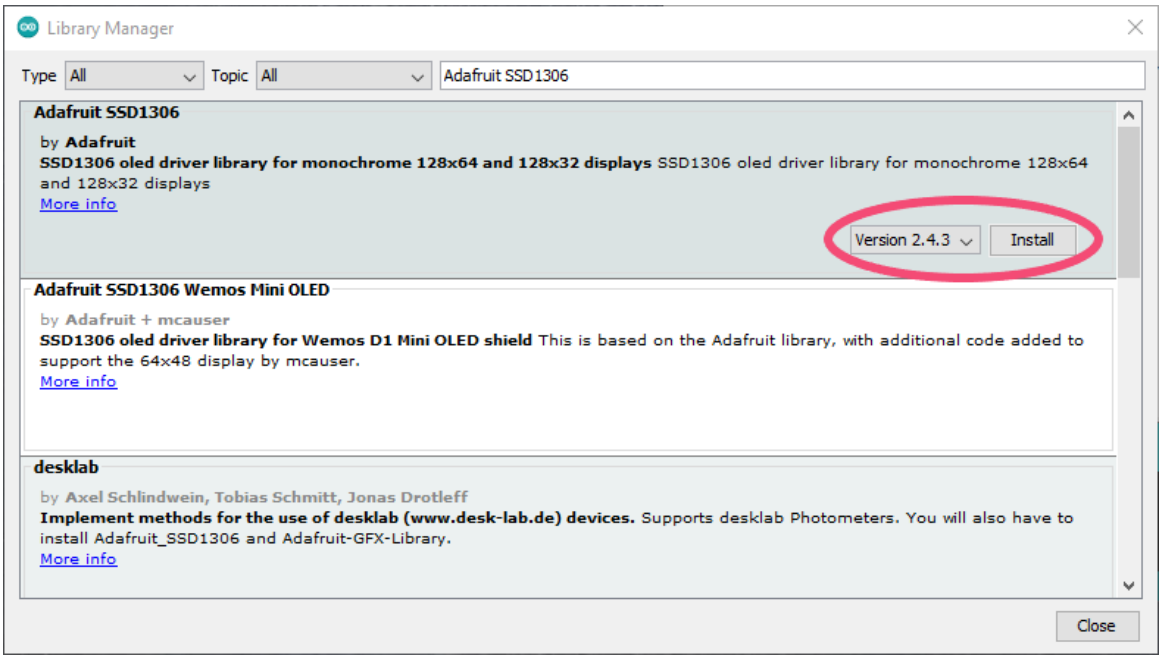
Važno: Kako bi uspješno napravili program, potrebno dodati neke knjižnice za korištenje dodatnih komponenti.

U Arduino prograsmkom okruženju otvorite Library Manager (klik na gornjem izborniku **Sketch** -> **Include Library** -> **Manage Libraries**) i instalirajte dvije biblioteke:

- Adafruit_GFX.h
- Adafruit_SSD1306.h

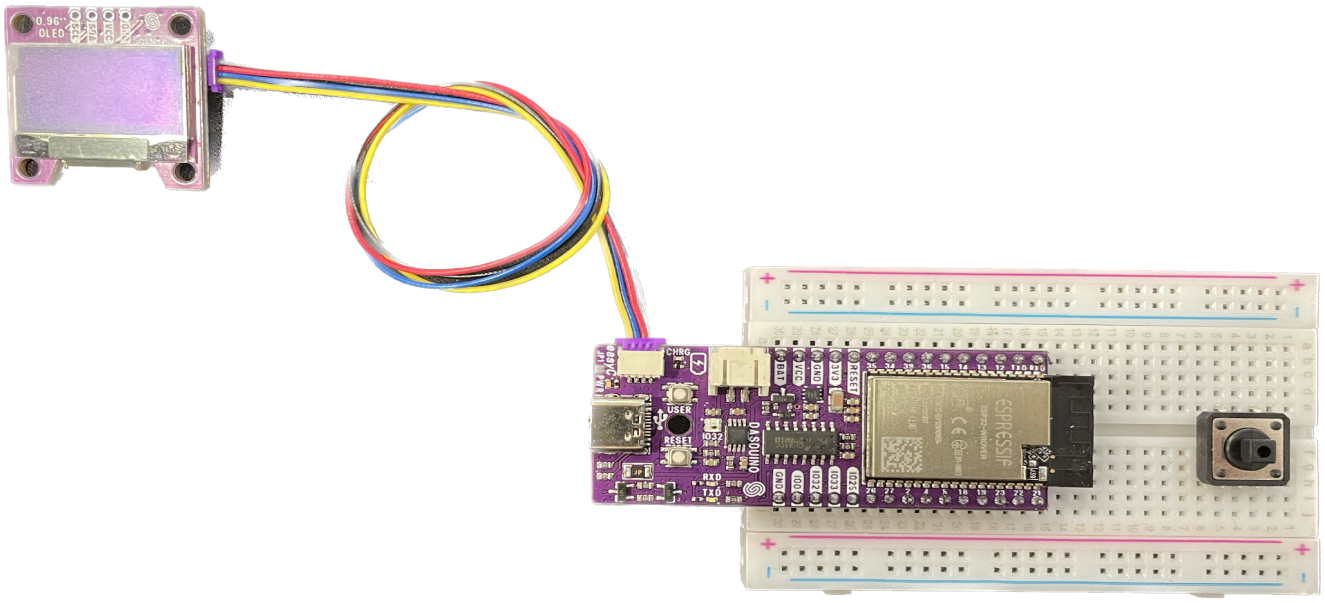
Biblioteke možete pronaći tako da upišete njihovo ime u tražilicu. Ukoliko Vas pri instalaciji bilo koje od knjižnica Arduino softver pita za instalaciju dodatnih knjižnica (dependency libraries) potvrdite instalaciju.





Napomena: Ne morate instalirati iste verzije kao na slici, slobodno instalirajte zadnje dostupne.

Prikaz spajanja



Programski kod rješenja

```
#include <Wire.h>              // omogućuje komunikaciju Arduina sa I2C komponentama
#include <Adafruit_GFX.h>      // omogućuje korištenje OLED ekrana
#include <Adafruit_SSD1306.h>  // omogućuje korištenje OLED ekrana

#define SCREEN_WIDTH 128 // sirina OLED ekrana u pikselima
#define SCREEN_HEIGHT 64 // visina OLED ekrana u pikselima

// Deklariranje SSD1306 ekrana spojenog na I2C sabirnicu (SDA i SCL izvodi)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
  Serial.begin(115200);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Adresa OLED ekrana (0x3C)
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(2000);
  display.clearDisplay();

  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 10);
  // Display static text
  display.println("Hello, world!");
  display.display();
}

void loop() {

}
```

Naredbe **include** na početku programa definiraju dodatne knjižnice koje je potrebno koristiti kako bi mogli upravljati OLED ekranom putem easyC sabirnice.

Naredbama **define** numeričkim smo vrijednostima dodijelili smo nazive i time definirali visinu i širinu OLED ekrana u pixelima.

Naredba **Adafruit_SSD1306** display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1); služi za inicijalizaciju ekrana.

Svaki put kada želite koristiti OLED ekran u svom programu morati imati sve gore navedene naredbe na samom početku, kao u ovom primjeru.

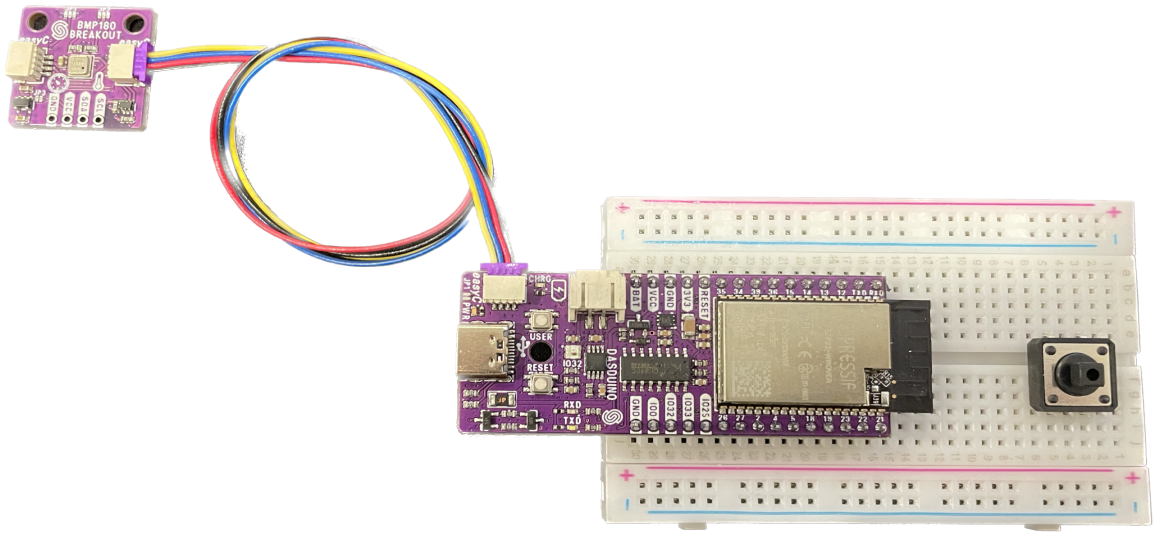
Naredbe **display.setTextSize** i **display.setTextColor** služe za postavljanje veličine i boje fonta kojom se tekst ispisuje na OLED ekran. Naredbom **display.setCursor** postavljamo kursor na željenu lokaciju na ekranu. **Display.println** zapisuje tekst na ekran na željeno mjesto a **display.display** prikazuje sve na ekranu. **Važno: bez korištenja opcije display.display tekst i slike koje smo 'ucrtali' na ekran neće se prikazati.**

Mjerenje temperature pomoću BMP180 senzora

Zadatak: Na dasduino spojite BMP180 senzor koristeći easyC priključak mikrokontrolera te izradite program kojim ćete na Serial Monitor-u prikazati tlak i temperaturu zraka.

VAŽNO:
Kako bi uspješno riješili zadatak, preuzmite biblioteku SFE_BMP180.h na svoje računalo i dodajte je u Arduino razvojno okruženje odabirom **Sketch** -> **Include Library** -> **Add .ZIP Library**…

Prikaz spajanja



Programski kod rješenja

```
#include "SFE_BMP180.h"   // omogućava korištenje BMP180 senzora
#include "Wire.h"

SFE_BMP180 tlak;

double tlakZraka, tempZraka;

void setup() {

  Serial.begin(115200);

  // provjera je li senzor povezan i radi li sve ispravno
  if(tlak.begin())  Serial.println("BMP180 uspjesno povezan.");
  else
  {
    Serial.print("BMP180 nije uspjesno povezan.");
    while(1);
  }
}

void loop() {

  // ocitavam i printam tlak zraka
  tlakZraka = ocitajTlak();
  Serial.println("Tlak zraka = " + String(tlakZraka) + "hPa");
```

```
  // ocitavam i zapisujem temp zraka
  tempZraka = ocitajTemperaturu();
  Serial.print("Temp zraka = " + String(tempZraka));
  Serial.print(char(248));
  Serial.println("C");

  // ispisuj podatke svakih 1000ms = 1sekundu
  delay(1000);
  Serial.println();

}

double ocitajTlak()
{
  char status;
  double temp, Tlak, tlak0, nadVisina;
  status = tlak.startTemperature();
  if(status != 0)
  {
    delay(status);
    status = tlak.getTemperature(temp);
    if(status != 0)
    {
      status = tlak.startPressure(3);
      if(status != 0)
      {
        delay(status);
        status = tlak.getPressure(Tlak,temp);
        if(status != 0)
        {
          return(Tlak);
        }
      }
    }
  }
}

double ocitajTemperaturu()
{
  char status;
  double temp;
  status = tlak.startTemperature();
  if(status != 0)
  {
    delay(status);
    status = tlak.getTemperature(temp);
    if(status != 0)
    {
      return(temp);
    }
  }
}
```

Naredbe **include** na početku programa definiraju dodatne knjižnice koje je potrebno koristiti kako bi mogli upravljati temperaturnim senzorom putem easyC sabirnice.

Naredba **SFE_BMP180** tlak; služi za inicijalizaciju temperaturnog senzora. Ovaj smo senzor u našem programu nazvali tlak, ali mogli smo mu dati bilo koje drugo ime.

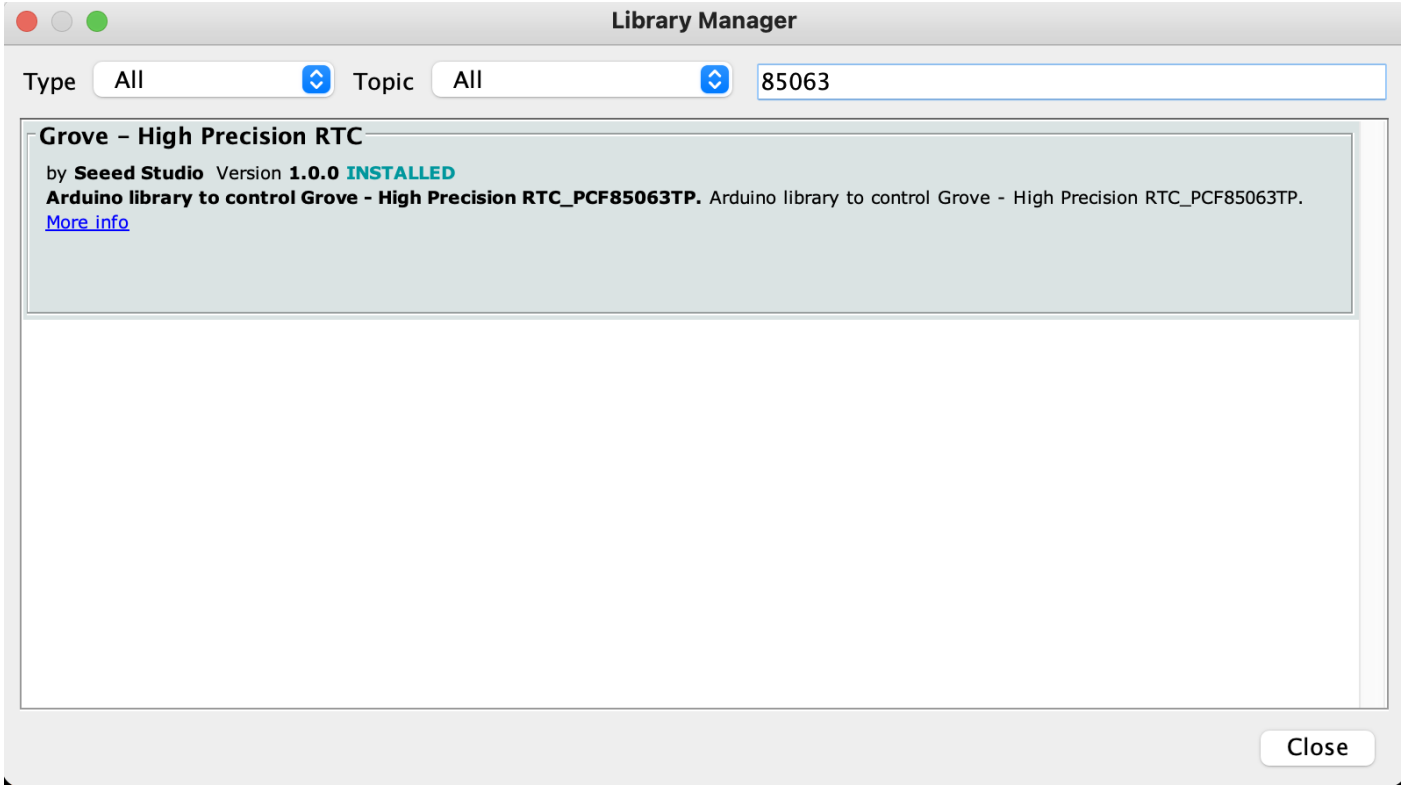
Svaki put kada želite koristiti temperaturni senzor u svom programu morati imati sve gore navedene naredbe na samom početku, kao u ovom primjeru.

Funkcije **ocitajTlak** i **ocitajTemperaturu** služe za očitavanje tlaka i temperature pomoću senzora i svaka nam vraća vrijednost očitane veličine.

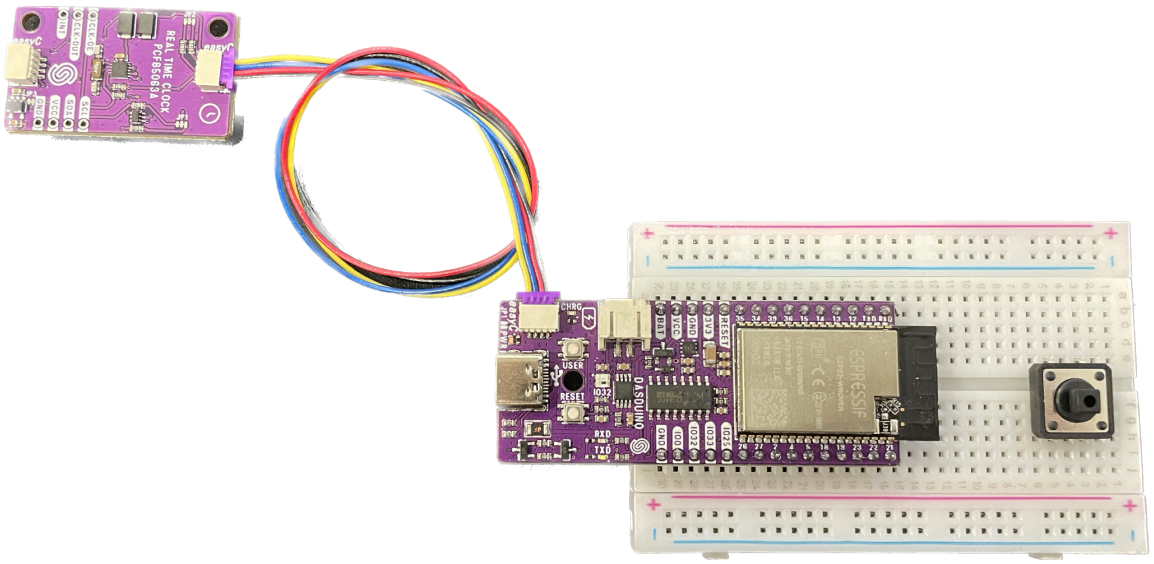
Mjerenje vremena

Zadatak: Na Dasduino spojite easyC RTC modul koji služi za mjerenje vremena (kao sat). Vrijeme i datum neka se ispisuju pomoću serijske veze na računalu.

Dodatne upute:
Kako bi sat mogao funkcionirati potrebno je dodati knjižnicu kako je prikazano na sljedećoj slici.



Prikaz spajanja



Programski kod rješenja

```
/*****************************************************************************/
//  Function: Set time and get the time from RTC chip(PCD85063TP) and display
//            it on the serial monitor.
```

```
//  Hardware: Grove - RTC v2.0
//  Arduino IDE: Arduino-1.6.6
//  Author:   lambor
//  Date:    June 14,2016
//  Version: v1.0
//  by www.seeedstudio.com
//
//  This library is free software; you can redistribute it and/or
//  modify it under the terms of the GNU Lesser General Public
//  License as published by the Free Software Foundation; either
//  version 2.1 of the License, or (at your option) any later verscion.
//
//  This library is distributed in the hope that it will be useful,
//  but WITHOUT ANY WARRANTY; without even the implied warranty of
//  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
//  Lesser General Public License for more details.
//
//  You should have received a copy of the GNU Lesser General Public
//  License along with this library; if not, write to the Free Software
//  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
//
/*****************************************************************************/
#include <Wire.h>
#include <PCF85063TP.h>

PCD85063TP sat;//define a object of PCD85063TP class
void setup()
{
  Serial.begin(115200);
  sat.begin();
}

void loop()
{
  printTime();
  delay(1000);
}
/*Function: Display time on the serial monitor*/
void printTime()
{
  sat.getTime();
  Serial.print(sat.hour, DEC);
  Serial.print(":");
  Serial.print(sat.minute, DEC);
  Serial.print(":");
  Serial.print(sat.second, DEC);
  Serial.print("  ");
  Serial.print(sat.month, DEC);
  Serial.print("/");
  Serial.print(sat.dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(sat.year+2000, DEC);
  Serial.print(" ");
  Serial.print(sat.dayOfMonth);
  Serial.print("*");
  switch (sat.dayOfWeek)// Friendly printout the weekday
  {
    case MON:
      Serial.print("MON");
```

```
      break;
    case TUE:
      Serial.print("TUE");
      break;
    case WED:
      Serial.print("WED");
      break;
    case THU:
      Serial.print("THU");
      break;
    case FRI:
      Serial.print("FRI");
      break;
    case SAT:
      Serial.print("SAT");
      break;
    case SUN:
      Serial.print("SUN");
      break;
  }
  Serial.println(" ");
}
```

Naredbe **include** na početku programa definiraju dodatne knjižnice koje je potrebno koristiti kako bi mogli upravljati satom (RTC modulom) putem easyC sabirnice.

Naredba **PCD85063TP** sat; služi za inicijalizaciju sata (RTC modula). Ovaj smo modul u našem programu nazvali sat, ali mogli smo mu dati bilo koje drugo ime.

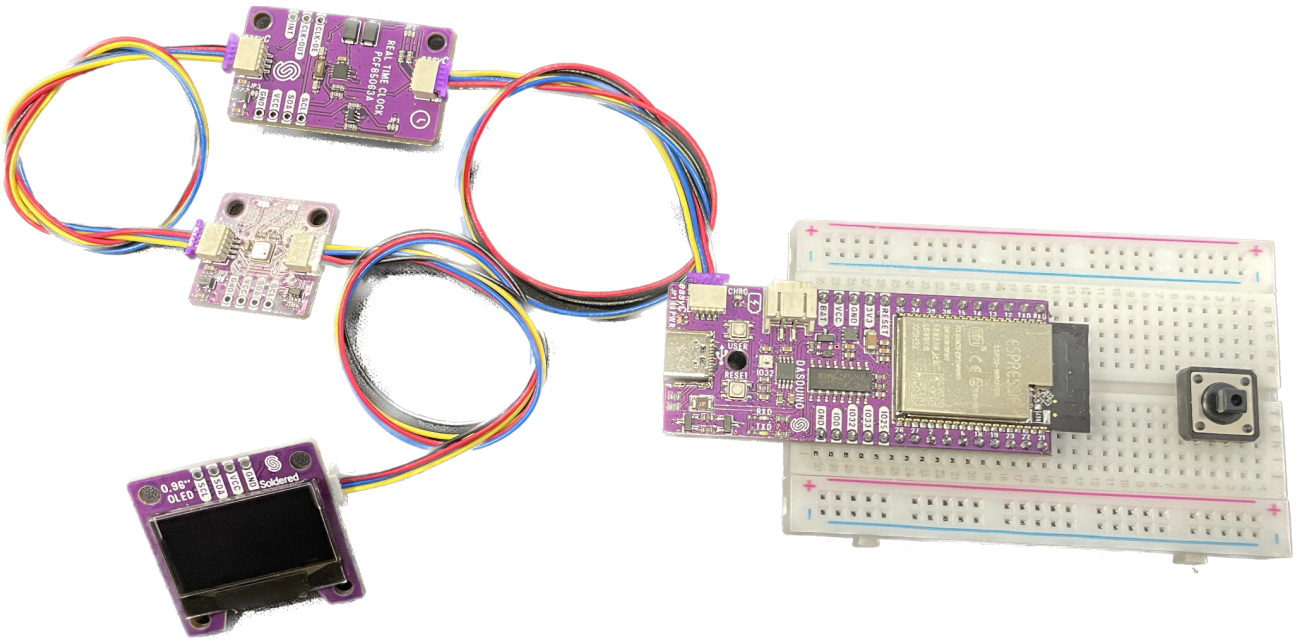
Svaki put kada želite koristiti ovaj modul u svom programu morati imati sve gore navedene naredbe na samom početku, kao u ovom primjeru.

Naredba **sat.getTime** služi za komunikaciju s RTC modulom i s njega očita točno vrijeme i datum. Naredbe **sat.hour**, **sat.minute** itd. vraćaju vrijednosti sati, minuta itd. koje onda možemo ispisati na računalu i koristiti na bilo koji drugi način.

Finalni projekt: Digitalni sat s mjerenjem temperature i tlaka zraka

Zadatak: Na Dasduino spojite easyC OLED ekran, modul za mjerenje temperature i tlaka te modul za mjerenje vremena. Koriteći se znanjima koje ste dobili u prethodnim primjerima napravite programski kod koji će omogućavati da se na OLED ekranu ispisuje vrijeme, temperatura i tlak zraka.

Prikaz spajanja



Programski kod rješenja

```
//uključivanje knjižnica potrebnih za dodatne easyC module
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <SFE_BMP180.h>
#include <PCF85063TP.h>


//definiranje veličine korištenog OLED ekrana
#define SCREEN_WIDTH 128 // sirina OLED ekrana u pikselima
#define SCREEN_HEIGHT 64 // visina OLED ekrana u pikselima

//deklariranje OLED ekrana spojenog na easyC sabirnicu
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

//deklariranje senzora za mjerenje temperature i tlaka
SFE_BMP180 senzortemp;

//deklariranje sata
PCD85063TP sat;


//inicijalizacija varijabli
int tlakZraka, tempZraka;

void setup() {
  Serial.begin(115200);                    //inicijalizacija serijske komunikacije

  //inicijalizacija sata
```

```
  sat.begin();

  //provjera je li OLED inicijaliziran
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // adresa OLED ekrana (0x3C)
    Serial.println(F("SSD1306 allocation failed"));
  }

  delay(1000);
  display.clearDisplay();                    //brisanje sadržaja oled ekrana

  //provjera je li senzor temperature uspješno povezan
  if (senzortemp.begin())  {
    Serial.println("BMP180 uspjesno povezan.");
  }
  else
  {
    Serial.print("BMP180 nije uspjesno povezan.");
  }
}

void loop() {
  // put your main code here, to run repeatedly:
  display.clearDisplay();

  tlakZraka = round(ocitajTlak());
  tempZraka = round(ocitajTemperaturu());
  sat.getTime();

  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 5);
  // Display static text
  display.print(sat.hour);
  display.print(":");
  display.print(sat.minute);
  display.print(":");
  display.println(sat.second);
  display.setCursor(0, 15);
  display.println(tlakZraka); // ispiši tlak zraka na OLED ekranu
  display.setCursor(0, 25);
  display.println(tempZraka); // ispiši temperaturu zraka na OLED ekranu
  display.display();

  delay (1000);
}

//dodatne funkcije - očitavanje tlaka i temperature

double ocitajTlak() {
  char status;
  /*
     definiramo varijable:
     temp - temperatura zraka
     Tlak - tlak zraka
     tlak0 - tlak na povrsini mora
     nadVisina - nadmorska visina
  */
  double temp, Tlak, tlak0, nadVisina;
  status = senzortemp.startTemperature();
```

```
  if (status != 0)
  {
    delay(status);
    status = senzortemp.getTemperature(temp);
    if (status != 0)
    {
      status = senzortemp.startPressure(3);
      if (status != 0)
      {
        delay(status);
        status = senzortemp.getPressure(Tlak, temp);
        if (status != 0)
        {
          return (Tlak);
        }
      }
    }
  }
}

double ocitajTemperaturu() {
  char status;
  double temp;
  status = senzortemp.startTemperature();
  if (status != 0)
  {
    delay(status);
    status = senzortemp.getTemperature(temp);
    if (status != 0)
    {
      return (temp);
    }
  }
}
```