

Meteorološka stanica temeljena na Arduino platformi

Čović, Dino

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:060811>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-03-06**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**METEOROLOŠKA STANICA TEMELJENA NA ARDUINO
PLATFORMI**

Rijeka, rujan 2022.

Dino Čović
0069088662

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**METEOROLOŠKA STANICA TEMELJENA NA ARDUINO
PLATFORMI**

Mentor: doc. dr. sc. Ivan Volarić

Rijeka, rujan 2022.

Dino Čović
0069088662

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za automatiku i elektroniku**
Predmet: **Digitalna elektronika**
Grana: **2.03.03 elektronika**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Dino Čović (0069088662)**
Studij: **Preddiplomski sveučilišni studij elektrotehnike**

Zadatak: **Meteorološka stanica temeljena na Arduino platformi / Arduino based weather station**

Opis zadatka:

Potrebno je izraditi prijenosnu meteorološku stanicu koja se sastoji od senzora temperature, tlaka i relativne vlažnosti zraka. Osim navedenih senzora, meteorološka stanica mora imati senzor kvalitete zraka, koji mjeri TVOC (eng. total volatile organic compound) i eCO2 (eng. equivalent carbon dioxide) vrijednosti. Sve navedene senzore potrebno je spojiti na Arduino razvojnu pločicu, te omogućiti prikaz izmjerenih vrijednosti na LCD ekranu.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Dino Čović

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:

Ivan Volarić

Doc. dr. sc. Ivan Volarić

Predsjednik povjerenstva za
završni ispit:

Viktor Sučić

Prof. dr. sc. Viktor Sučić

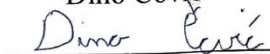
SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski sveučilišni studij elektrotehnike

IZJAVA

Sukladno „*Pravilniku o završnom radu, završnom ispitu i završetku preddiplomskih sveučilišnih studija*“, izjavljujem da sam samostalno izradio završni rad pod nazivom „*Meteorološka stanica temeljena na Arduino platformi*“ primjenom znanja stečenog tijekom preddiplomskog sveučilišnog studija elektrotehnike.

Rijeka, srpanj 2022.

Dino Čović



SADRŽAJ

1 . UVOD.....	1
2 . ARDUINO	2
2.1. Arduino Uno.....	2
2.1.1. Sklopovlje	2
2.2. Komunikacijski protokoli.....	4
2.2.1. UART.....	4
2.2.2. SPI.....	5
2.2.3. I ² C	6
2.3. Integrirano razvojno okruženje	8
2.3.1. Stvaranje i pohranjivanje Arduino programske datoteke.....	8
2.3.2. Unos programskog koda i učitavanje biblioteka	9
2.3.3. Provjera programskog koda te pohranjivanje programa unutar Arduino pločice....	11
3 . MJERNI PRETVORNICI I OSTALI UREĐAJI	12
3.1. Eksperimentalna pločica	12
3.2. GY-BMP280	13
3.3. DHT22	15
3.4. CJMCU-811.....	17
3.5. LCD 16x2 zaslon s HD44780 kontrolerom	19
3.5.1. I ² C adapter za LCD 16x2 zaslon	20
4 . IZRADA ARDUINO METEOROLOŠKE STANICE	21
4.1. Postavljanje Arduino Uno R3 pločice.....	21
4.1.1. Odabir I ² C ulaza i njihovo spajanje na eksperimentalnu pločicu	21
4.1.2. Stvaranje Arduino programske datoteke i uključivanje I ² C biblioteka	23
4.2. Povezivanje i konfiguriranje LCD 16x2 zaslona	24
4.2.1. Konfiguriranje LCD 16x2 zaslona s I ² C adapterom u Arduino razvojnom okruženju	25
4.3. Povezivanje senzora BMP280	29
4.3.1. Spajanje modula GY-BMP280 na eksperimentalnu pločicu	29
4.3.2. Učitavanje programske biblioteka i programiranje uređaja u Arduino razvojnom okruženju	30

4.4. Povezivanje senzora CCS811.....	34
4.4.1. Spajanje modula CJMCU-811 na eksperimentalnu pločicu	34
4.4.2. Učitavanje programske biblioteke i programiranje uređaja u Arduino razvojnom okruženju	36
4.5. Povezivanje senzora DHT22.....	40
4.5.1. Spajanje modula DHT22 na eksperimentalnu pločicu	40
4.5.2. Učitavanje programske biblioteke i programiranje uređaja u Arduino razvojnom okruženju	42
5 . ZAKLJUČAK.....	46
6 . LITERATURA	47

1. UVOD

Razvoj elektronike, grane elektrotehnike koja se bavi proučavanjem i upravljanjem protoka elektrona te izradom raznih elektroničkih sklopova i uređaja koji to omogućuju, započinje početkom 20. stoljeća izumom vakuumske cijevi. U sljedećih nekoliko desetljeća to će omogućiti razvoj raznih elektroničkih naprava, no važnije od svega razvoj i prvih elektroničkih računala. Međutim, razvijena računala nisu bila previše učinkovita uslijed niza nedostataka vezanih uz vakuumske cijevi. Znanstvenici su težili ka otkriću učinkovitijeg rješenja, stoga je sredinom 50-ih godina 20. stoljeća osmišljen poluvodički elektronički uređaj pod nazivom tranzistor. Tranzistor je omogućio razvoj integriranih krugova (eng. IC – *Integrated Circuit*) koji su ubrzo postali sastavni dio gotovo svih elektroničkih sklopova i uređaja pa tako i centralnih procesorskih jedinica (eng. CPU – *Central Processing Unit*). Nastaju mikroprocesorske jedinice koje sadrže jednu ili više centralnih procesorskih jedinica i obavljaju veći broj različitih funkcija u odnosu na osnovne zadatke CPU-a.

U međuvremenu razvoj elektronike bio je od velikog značaja za razna znanstvena područja. Postavljeni su temelji za ubrzani razvoj naprednih računala, digitalizacije, medicinskih uređaja, industrijske elektronike i automatizacije, mobilnih uređaja, uređaja za mjerenje fizikalnih veličina i pojava, uređaja za digitalnu obradu signala primjerice glazbe, digitalna obrada slika, a u skorije vrijeme i značajan razvoj električnih vozila i umjetne inteligencije.

Postavlja se pitanje kako je razvoj elektronike utjecao na jednu od najstarijih znanstvenih disciplina – mjeriteljstvo, metrologiju, odnosno znanost o mjerenju. Nova otkrića u elektronici uvjetovala su nastanku mjernih elektroničkih uređaja poput osciloskopa, brojača frekvencije, klasičnih multimetara, mjernih osjetila i sl.

Završni rad bavi se upravo problematikom primjene elektroničkih uređaja za mjerenje fizikalnih veličina i pojava prisutnih u Zemljinoj atmosferi. Rad opisuje primjenu mjernih osjetila (senzora) za mjerenje meteoroloških pojava i prikaz vrijednosti mjerenih fizikalnih veličina na zaslonu. Zadatke povezivanja, uspostave komunikacije, prikupljanja, analize i obrade podataka te upravljanja navedenim uređajima obavlja Arduino razvojna elektronička platforma.

2. ARDUINO

Arduino je naziv za otvorenu (eng. *open-source*) elektroničku platformu koja se sastoji od sklopovlja i programske podrške i namijenjena je za izradu elektroničkih uređaja.

Projekt Arduino nastao je 2005. godine, a do danas je istoimena tvrtka razvila više vrsta Arduino elektroničkih pločica od kojih je najpoznatija Arduino Uno. Jednostavnost, raširenost i pristupačna cijena temeljni su razlozi zbog kojih je Arduino prvi odabir za mnoge pri ostvarivanju elektroničkih projekata [1, 2].

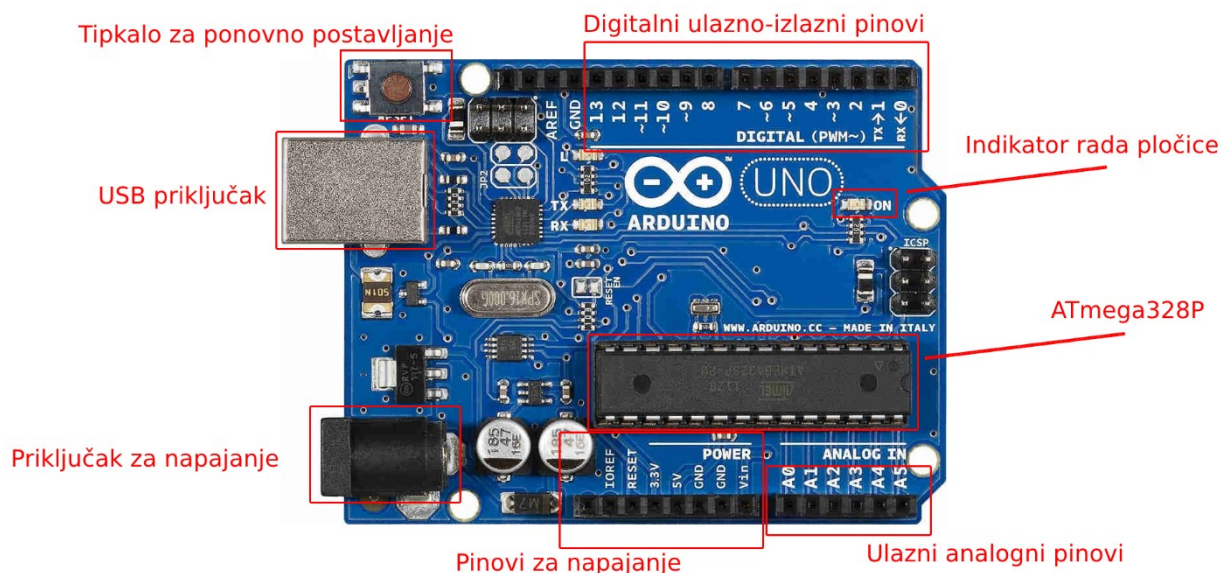
2.1. Arduino Uno

Arduino Uno najraširenija je inačica Arduina. Alternativne inačice vrijedne spomena su Arduino Nano i Arduino Mega. Glavni nedostatak Arduino Nano pločice u odnosu na Arduino Uno je manjak dijelova unutar sklopovlja, dok Arduino Mega pločica ima prošireno sklopovlje što ju čini skupljom i za potrebe izrade završnog rada nepovoljnom [3].

2.1.1. Sklopovlje

Završni rad baziran je na Arduino Uno R3 razvojnoj pločici. Arduino Uno R3 razvojna pločica temelji se na ATmega328P mikrokontroleru frekvencije do 20 MHz koji za ulogu ima pohranjivanje i izvršavanje programskog koda. ATmega328P ima mogućnost pohranjivanja 32 KB programskog koda uz 8-bitnu brzinu procesiranja informacija.

Sklopovlje se, pored već navedenog mikrokontrolera, sastoji i od ulazno-izlaznih digitalnih pinova, ulaznih analognih pinova, pinova za napajanje, USB priključka, priključka za napajanje, tipkala za ponovno postavljanje te raznih indikatora, kao što je prikazano na slici 2.1.



Slika 2.1. Sklopovlje Arduino Uno R3 elektroničke pločice [4]

Prilikom izrade elektroničkog uređaja Arduino Uno pločica povezuje se s računalom putem USB kabela (USB 2.0 tip A/B) čime se pločica spaja na 5V istosmjernog el. napona te se istovremeno uspostavlja serijska komunikacija između računala i pločice. Jednom kada se konačni programski kod pohrani u mikrokontroleru, Arduino Uno pločicu moguće je napajati vanjskim izvorom napajanja u rasponu od 7V do 12V putem USB priključka ili V_{IN} pina. Neovisno o načinu napajanja, pinovi 3.3V i 5V na izlazu daju konstantan napon odgovarajućeg iznosa. GND pinovi služe za uzemljenje.

Ukupno četrnaest digitalnih ulazno-izlaznih pinova uključeno je u sklopovlje i mogu se koristiti kao ulazi ili izlazi. Šest od ukupno četrnaest digitalnih pinova podržavaju pulsno-širinsku modulaciju PWM (eng. *pulse-width modulation*). Hoće li pojedini digitalni pin služiti kao ulazni ili izlazni definira se unutar integriranog razvojnog okruženja primjenom odgovarajućih funkcija. Određeni digitalni pinovi također imaju funkciju uspostavljanja različitih komunikacijskih protokola.

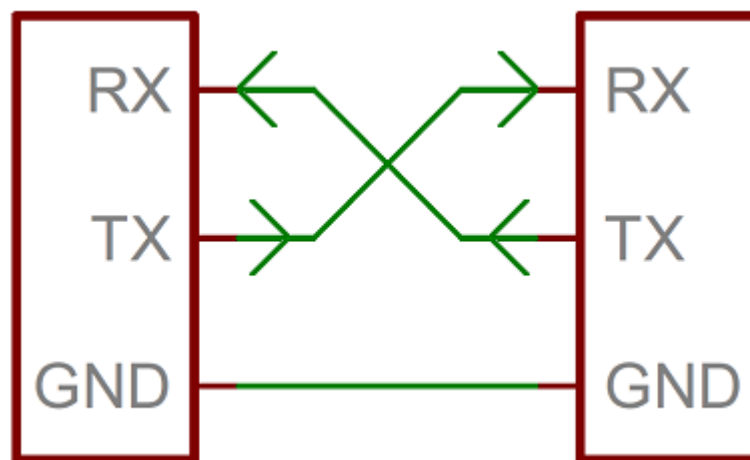
Sklopovlje sadrži i šest ulaznih analognih pinova za rad s analognim veličinama. Za razliku od digitalnih ulaza, odnosno izlaza, koji mogu poprimiti isključivo dva logička stanja, stanje niske naponske razine (logička 0) ili stanje visoke naponske razine (logička 1), putem analognih ulaza moguće je ostvariti bilo koju naponsku vrijednost u rasponu od 0V do 5V. Slično kao i kod digitalnih pinova, pojedini analogni pinovi obavljaju više funkcija prema potrebi [5].

2.2. Komunikacijski protokoli

Jedna od najosnovnijih podjela komunikacijskih sustava je na žičane i bežične. Arduino elektronička platforma temelji se na žičanoj serijskoj komunikaciji između uređaja. Uspostava bežične komunikacije pomoću Arduina zahtjeva primjenu vanjskih modula, što ne spada u zadatak završnog rada, stoga će u ovom poglavlju biti obrađena isključivo tri serijska komunikacijska protokola za koje je Arduino građen: UART, SPI, I²C. Navedeni komunikacijski protokoli razlikuju se u izvedbi, međutim svrha im zajednička: brza razmjena podataka između uređaja koji se nalaze u neposrednoj okolini [6].

2.2.1. UART

UART (eng. *Universal Asynchronous Receiver/Transmitter*) serijski je komunikacijski protokol koji omogućava izravnu razmjenu podataka između dva uređaja putem dva vodiča. Jedan vodič oba uređaja služi za slanje podataka (oznake TX), a drugi za prijem podataka (oznake RX). Blok shema UART komunikacije između dva uređaja prikazana je na slici 2.2.

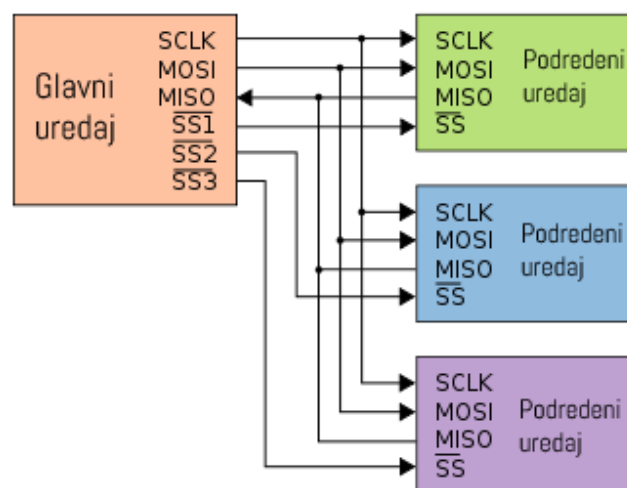


Slika 2.2. Blok shema UART komunikacije između dva uređaja [7]

Arduino Uno pločica omogućuje UART komunikaciju s vanjskim uređajima putem dva digitalna pina. Jedan služi za slanje podataka (digitalni pin 1, GPIO1), a drugi za prijem podataka (digitalni pin 0, GPIO0). Glavni nedostatak UART komunikacije je nemogućnost razmjene podataka između više od dva uređaja istovremeno. Podaci se šalju serijski u obliku tzv. podatkovnih paketa, stoga nije moguće slati više sadržaja s različitih uređaja istim vodom, jer se gubi željeno krajnje odredište pojedinih podatkovnih paketa. Rezultat toga je pogreška i gubitak podataka [8, 9].

2.2.2. SPI

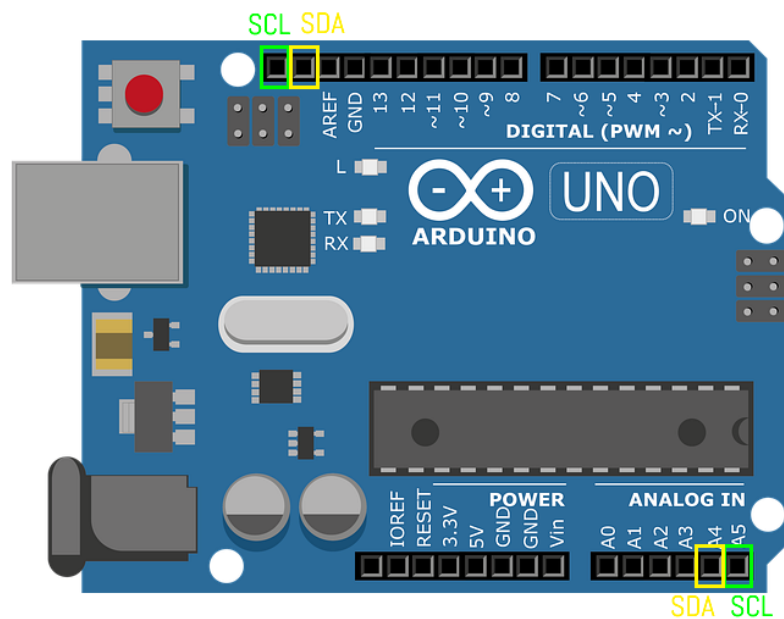
SPI (eng. *Serial Peripheral Interface*) sinkroni je serijski komunikacijski protokol koji radi na principu tzv. „master-slave“ komunikacijske strukture. Strukturu čini jedan glavni uređaj (eng. *master device*) i mnoštvo podređenih uređaja (eng. *slave device*), gdje glavni uređaj upravlja podređenima. Najveća prednost SPI komunikacije u odnosu na opisanu UART komunikaciju je mogućnost upravljanja mnoštvom uređaja istovremeno. Glavni uređaj uspostavlja komunikaciju i razmjenjuje podatke s točno odabranim podređenim uređajem, gdje se odabir uređaja vrši putem zasebnih SS (eng. *Select Slave*) vodiča. Budući da glavni uređaj zna s kojim podređenim uređajem komunicira, ne javlja se potreba za uporabom odvojenih podatkovnih vodiča, već se koriste zajednički podatkovni vodiči za slanje i prijem podataka te jedan dodatni zajednički vodič za sinkronizaciju uređaja. Sinkronizacija kod SPI komunikacije je bitna, jer osigurava jednaku brzinu razmjene podataka za sve uređaje. Podatkovni vodič za prijenos podataka iz glavnog u podređeni uređaj uobičajeno nosi oznaku MOSI (eng. *Master Out Slave In*), dok vodič za prijem podataka u glavnom uređaju nosi oznaku MISO (eng. *Master In Slave Out*). Zajednički vodič koji osigurava sinkronizaciju u većini slučajeva ima oznaku SCK ili SCLK (eng. *Serial Clock*). Na slici 2.3. prikazana je blok shema uspostavljanja SPI komunikacije između više uređaja. Digitalni pinovi 10, 11, 12 i 13 na Arduino Uno R3 pločici podržavaju uspostavu SPI komunikacije [8, 9, 10].



Slika 2.3. Blok shema uspostavljanja SPI komunikacije između više uređaja [14]

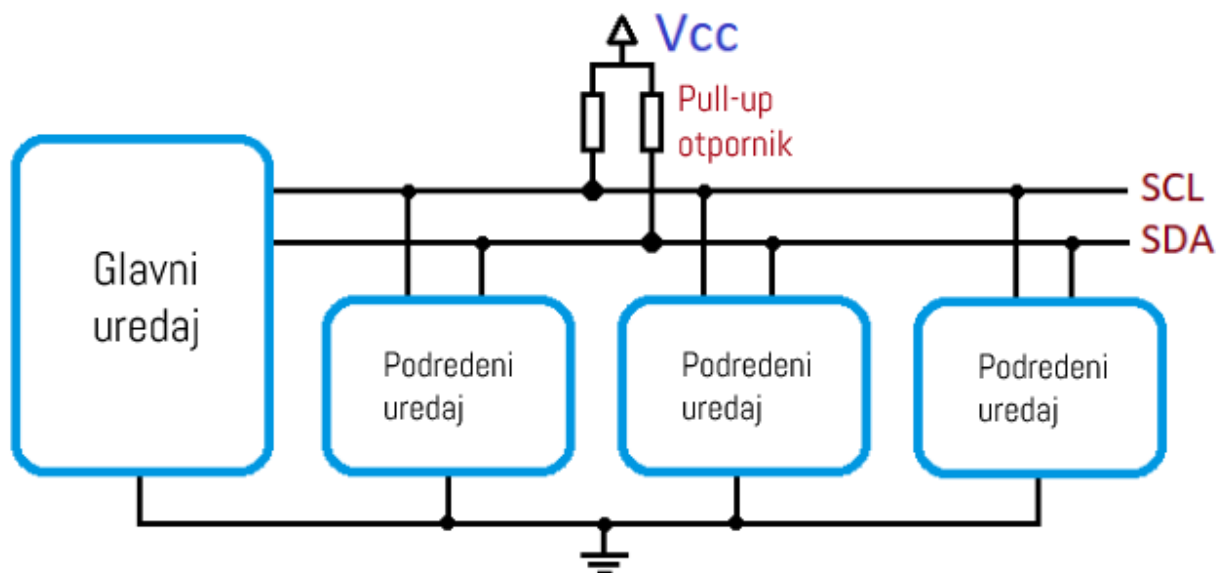
2.2.3. I²C

I²C (eng. *Inter Integrated Circuit*) sinkroni je serijski komunikacijski protokol strukture slične onoj opisanoj kod SPI komunikacijskog protokola, ali uz nekoliko unaprijeđenja. Kod I²C komunikacije također postoji „master-slave“ odnos, međutim broj glavnih uređaja nije ograničen na jedan, već je moguće povezati onoliko glavnih uređaja koliko je potrebno. Razmjena podataka vrši se u oba smjera putem zajedničkog podatkovnog vodiča koji radi kao sabirnica, oznake SDA (eng. *Serial Data*). Drugi zajednički vodič služi za sinkronizaciju uređaja putem radnog takta, oznake SCL (eng. *Serial Clock*), što ponovno ima pozitivan utjecaj na brzinu slanja i primanja podataka. Svi uređaji unutar strukture trebaju biti ispravno adresirani kako bi sabirnica mogla obavljati svoju dužnost usmjeravanja podataka na odgovarajuću lokaciju. Arduino Uno R3 pločica dolazi s ugrađenim parom SDA i SCL pinova koji su izvedeni na način da im se može pristupiti na strani analognih, odnosno digitalnih ulaza, prema slici 2.4.



Slika 2.4. Model Arduino Uno pločice s označenim pinovima za uspostavu I²C komunikacije [14]

I²C komunikacija radi na principu povratne informacije iz odabranog podređenog uređaja koju je zatražio jedan od glavnih uređaja. Drugim riječima, isključivo glavni uređaj može započeti komunikaciju s podređenim uređajem zahtjevajući razmjenu podataka. Za vrijeme trajanja komunikacije između glavnog i podređenog uređaja, preostali uređaji nalaze se u stanju mirovanja i upravo zato je moguće obavljati razmjenu podataka putem samo jedne sabirnice. SDA i SCL vodiči uređaja koji su u stanju mirovanja nalaze se u stanju visoke naponske razine (logička 1) što se ostvaruje primjenom tzv. „pull-up“ otpornika. Promjena logičkog stanja nastupa onda kada glavni uređaj zatraži uspostavu komunikacije sa sljedećim podređenim uređajem, pri čemu se njegov prethodnik dovodi u stanje mirovanja. Blok shema uspostavljanja I²C komunikacije između više uređaja prikazana je na slici 2.5. [8, 9, 12, 13].



Slika 2.5. Blok shema uspostavljanja I²C komunikacije između više uređaja [15]

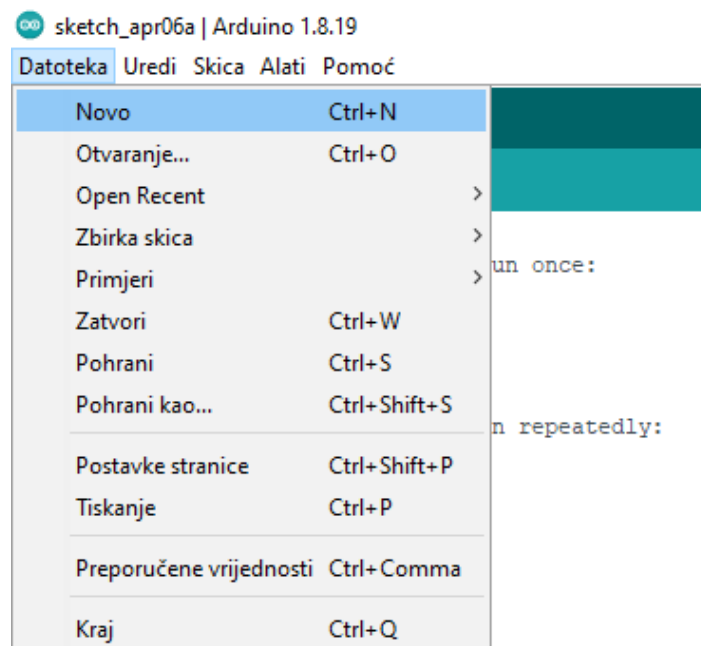
Komunikacija između uređaja korištenih u izradi završnog rada obavlja se upravo putem I²C komunikacijskog protokola. Rad ne obuhvaća primjenu UART i SPI komunikacije, no njihovo razumijevanje je ključno, budući da I²C komunikacija združuje njihove prednosti uz određena unaprijeđenja.

2.3. Integrirano razvojno okruženje

Arduino elektronička platforma uključuje programsku podršku u vidu Arduino integriranog razvojnog okruženja (eng. IDE – *Integrated Development Environment*). Arduino programski jezik temelji se na višim programskim jezicima C i C++, ali uz određene izmjene u vidu sintakse s ciljem poboljšanja korisničkog iskustva tijekom pisanja koda. Arduino IDE sadrži uređivač teksta za unos programskog koda, tekstualnu konzolu, alatnu traku s često korištenim alatima te niz raznih izbornika [16, 17].

2.3.1. Stvaranje i pohranjivanje Arduino programske datoteke

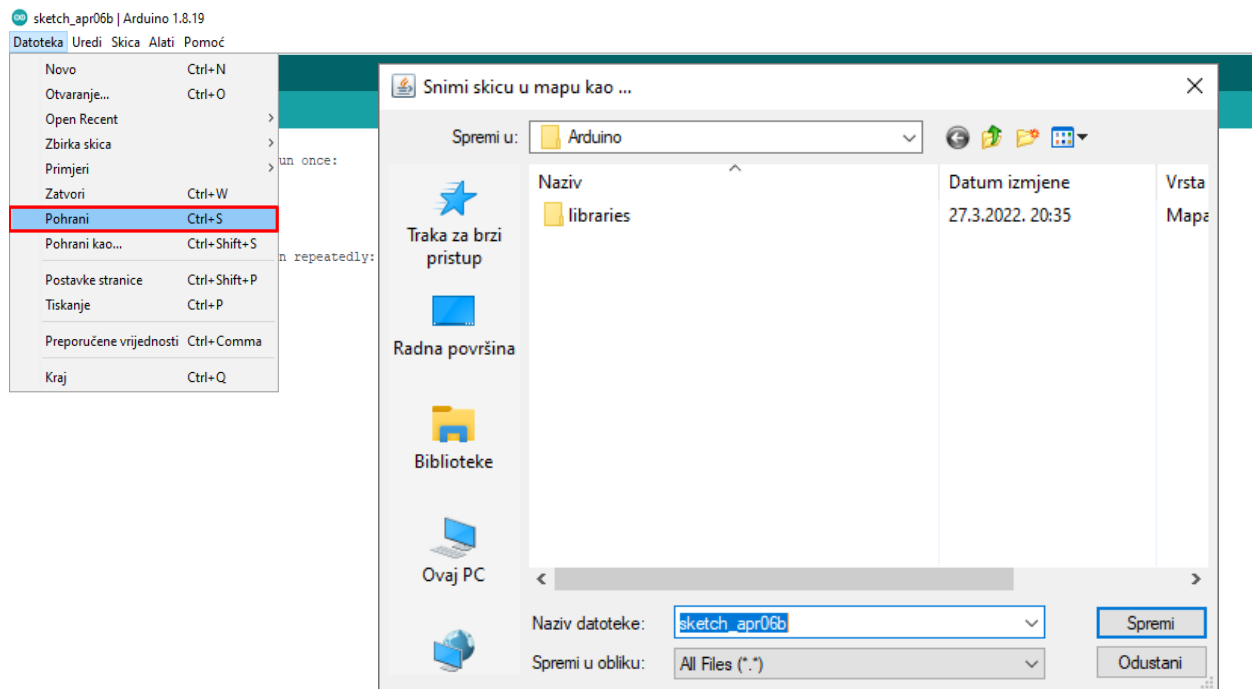
Rad u Arduino integriranom razvojnom okruženju započinje stvaranjem nove datoteke čime se otvara novi prozor u razvojnom okruženju. Nastaje nova tzv. „skica“, tj. tekstualna datoteka za pisanje programskog koda [16]. Stvaranje Arduino skice prikazano je na slici 2.6.



Slika 2.6. Stvaranje Arduino skice

Svaka Arduino skica sadrži dvije osnovne funkcije neophodne za ispravan rad programa. Prva osnovna funkcija naziva *setup* izvršava se jednom prilikom pokretanja programa i služi za postavljanje početnih uvjeta i vrijednosti raznih varijabli [18].

Druga osnovna funkcija *loop* poziva se nakon izvršenja funkcije *setup*. Unutar *loop* funkcije nalazi se programski kod koji se iznova izvršava. Radi se o beskonačnoj petlji koja se izvršava sve do trenutka isključenja Arduino pločice. Jedan izvršeni ciklus petlje predstavlja jednu iteraciju petlje. Stvorenu skicu potrebno je zatim pohraniti na računalo (slika 2.7.). Spremljene skice automatski poprimaju nastavak *.ino* [16, 19].



Slika 2.7. Spremanje Arduino skice na računalo

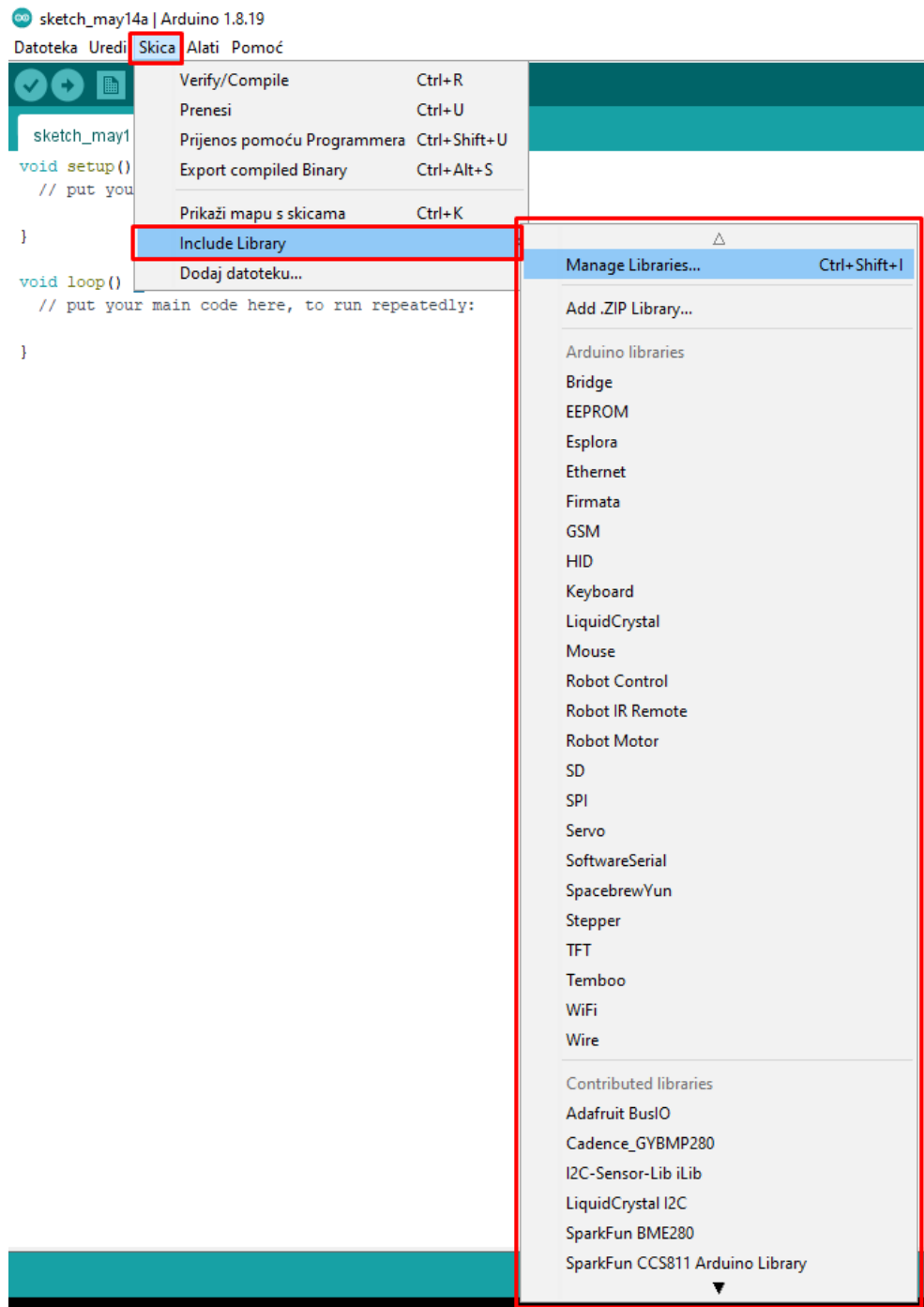
2.3.2. Unos programskog koda i učitavanje biblioteka

Osnovni preduvjet unosa programskog koda je poznavanje Arduino programskog jezika. Programski kod ima svoju strukturu i čine ga definicije, funkcije, varijable, naredbe i ostali dijelovi koji postaju dio programa. Značajnu ulogu u pisanju programskog koda imaju programske biblioteke (*eng. library*).

Biblioteka predstavlja jednu ili više programskih datoteka koje čine cjelinu, a sadrže gotov programski kod većim dijelom sastavljen od raznih funkcija. Zadatak biblioteka je olakšati izradu programa tako što korisniku pružaju već gotove alate, primjerice biblioteka koja sadrži razne matematičke funkcije i konstante. Na taj način korisnik ostaje usredotočen i izrađuje isključivo programske cjeline usko vezane uz projektni zadatak. Jedna biblioteka može se uključiti u bezbroj programskih datoteka. Postupak uključivanja biblioteka u Arduino skicu prikazan je na slici 2.8.

Općenito, programski jezici dolaze s nizom standardnih biblioteka za rad, a to isto vrijedi i za Arduino programski jezik.

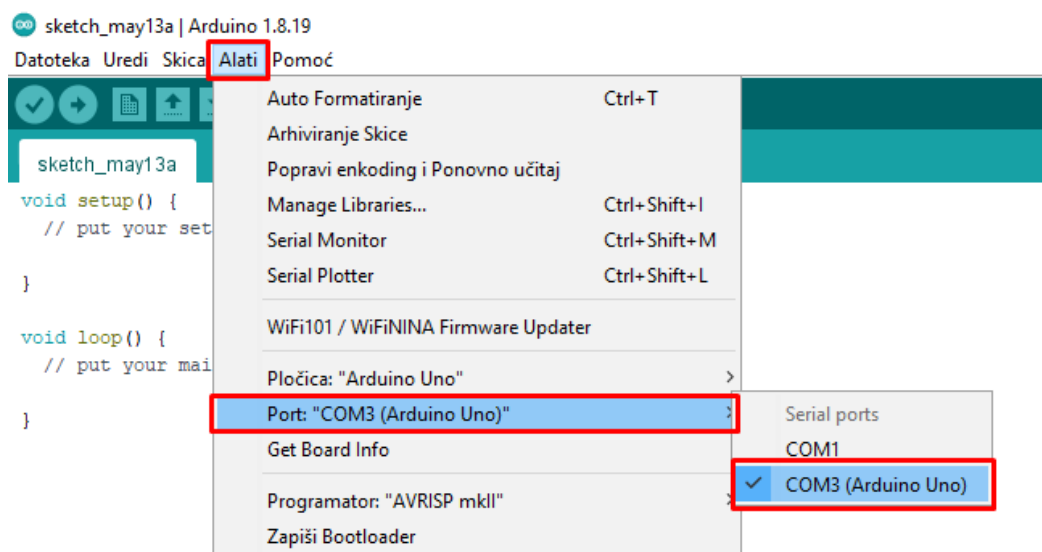
Pored standardnih postoje i dodatne biblioteke dostupne za preuzimanje na računalo. Završni rad obuhvaća primjenu standardnih Arduino biblioteka za uspostavu komunikacije između računala i Arduino pločice, razmjenu podataka, slanje povratnih informacija korisniku putem korisničkog sučelja, rad sa sensorima, prikaz vrijednosti na drugim uređajima i slično [20, 21].



Slika 2.8. Uključivanje Arduino biblioteka u skicu

2.3.3. Provjera programskog koda te pohranjivanje programa unutar Arduino pločice

Cjelokupan Arduino programski kod potrebno je provjeriti (*eng. compile*) kako bi se utvrdila njegova ispravnost. Sve eventualne pogreške trebaju se ukloniti prije nastavka. Uspješna provjera programskog koda rezultira pretvorbom istog u strojni jezik, odnosno nastaje program. Prijenos programa posljednji je korak koji podrazumijeva njegovo pohranjivanje unutar mikrokontrolera. U padajućem izborniku naziva *Alati* pod opcijom *Port* prethodno se vrši odabir Arduino pločice povezane s računalom za pohranu programa (slika 2.9.). Program je nakon prijena spreman za izvršavanje, a uređaj za rad.



Slika 2.9. Odabir odgovarajuće Arduino pločice za prijenos i pohranu programa

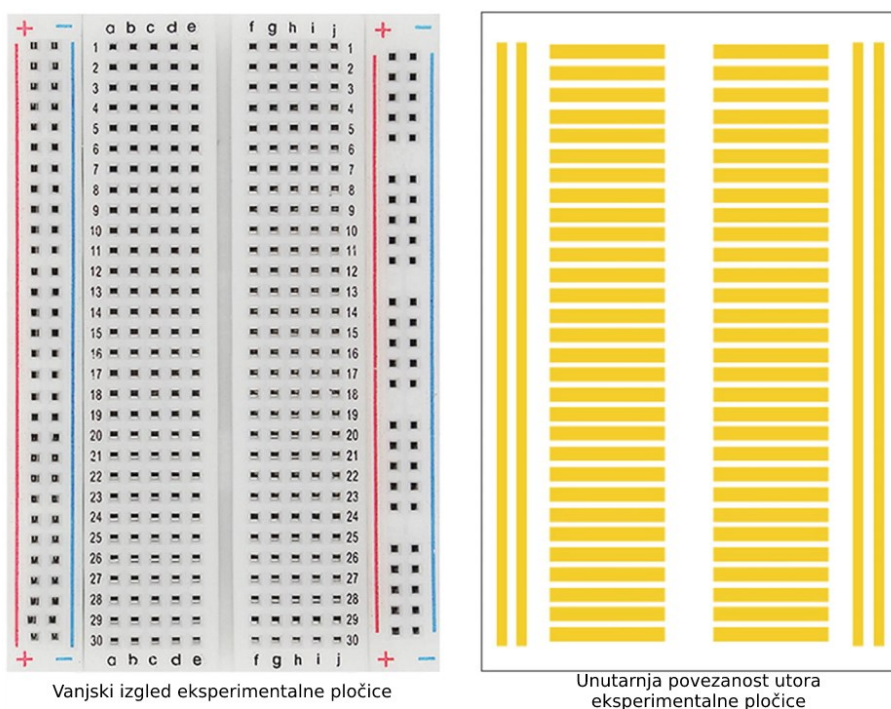
3. MJERNI PRETVORNICI I OSTALI UREĐAJI

Sklopovlje završnog rada sastoji se prije svega od Arduino Uno pločice opisane u poglavlju broj dva, ali i mjernih pretvornika te ostalih uređaja čija teorijska podloga slijedi u nastavku.

Mjerno osjetilo ili senzor je uređaj na ulazu u mjerni pretvornik koji mjeri fizikalnu veličinu, primjerice temperaturu zraka, relativnu vlažnost zraka, broj okretaja električnog stroja i sl. Mjerni pretvornik zatim pretvara vrijednosti mjerene veličine u signal pogodan za daljnju obradu. Najčešće se radi o električnom signalu. Pretvoreni signal prosljeđuje se u određeni podsustav gdje se nad njime vrše različite radnje ovisno o potrebi. Primjer jedne takve radnje bilo bi periodično prikupljanje i analiza podataka te njihov prikaz [22].

3.1. Eksperimentalna pločica

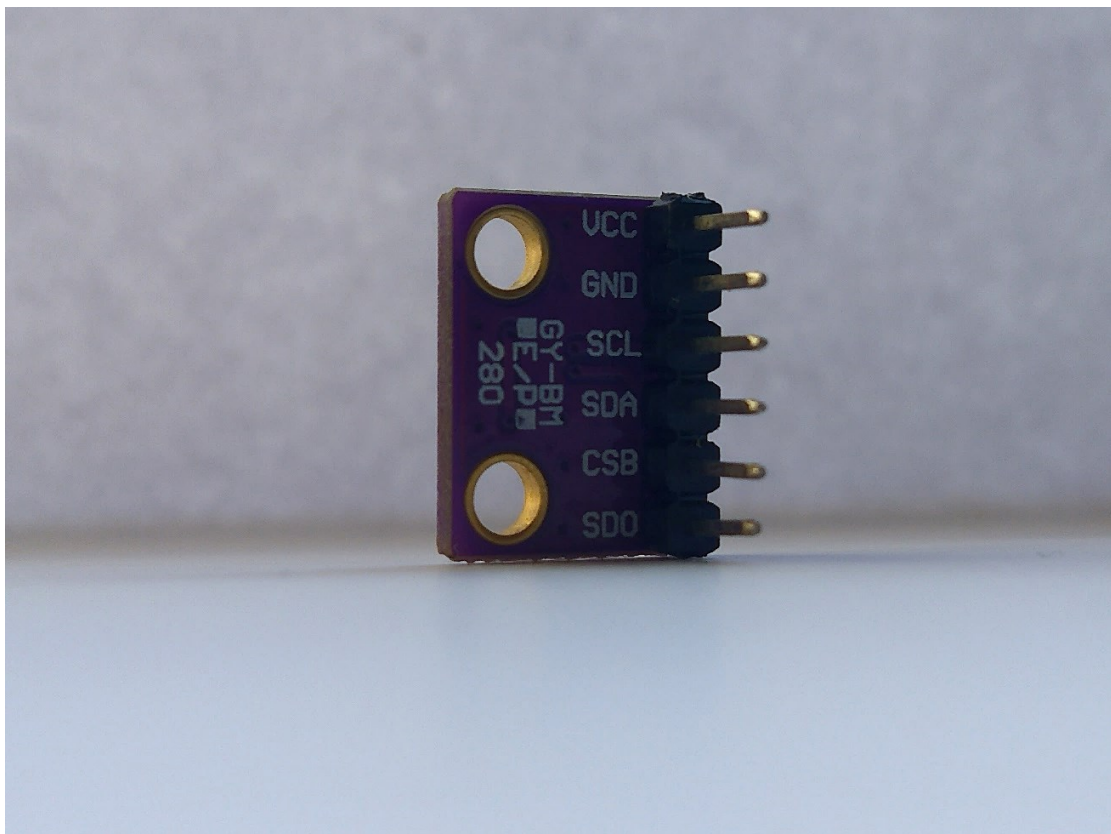
Eksperimentalna pločica (eng. *solderless breadboard*) je pločica koja služi kao pomagalo pri spajanju elektroničkih komponenti i uređaja. Sadrži mnoštvo utora za umetanje nožica različitih komponenti. Utori su u unutrašnjosti pločice međusobno povezani u odvojenim linijama pomoću metalnih (vodljivih) traka, kao što je to prikazano na slici 3.1. Kućište pločice izrađuje se od plastike. Eksperimentalne pločice razlikuju se u dimenzijama i izvedbama [23, 24].



Slika 3.1. Vanjski i unutarnji izgled eksperimentalne pločice [25]

3.2. GY-BMP280

Modul GY-BMP280 prikazan na slici 3.2. sadrži mjerno osjetilo BMP280 tvrtke Bosch koje ima sposobnost vrlo preciznog mjerenja atmosferskog tlaka s apsolutnom točnošću ± 1 hPa i temperature zraka s apsolutnom točnošću $\pm 1^\circ\text{C}$. S obzirom na visoku točnost mjerenja tlaka zraka, senzor se može koristiti i kao visinomjer s točnošću ± 1 m. Odlikuju ga pristupačna cijena, niska potrošnja energije te mogućnost uspostave I²C i SPI komunikacije s vanjskim mikrokontrolerom [26, 27, 28]. Tehničke karakteristike senzora BMP280 tvrtke Bosch navedene su u tablici 3.1.

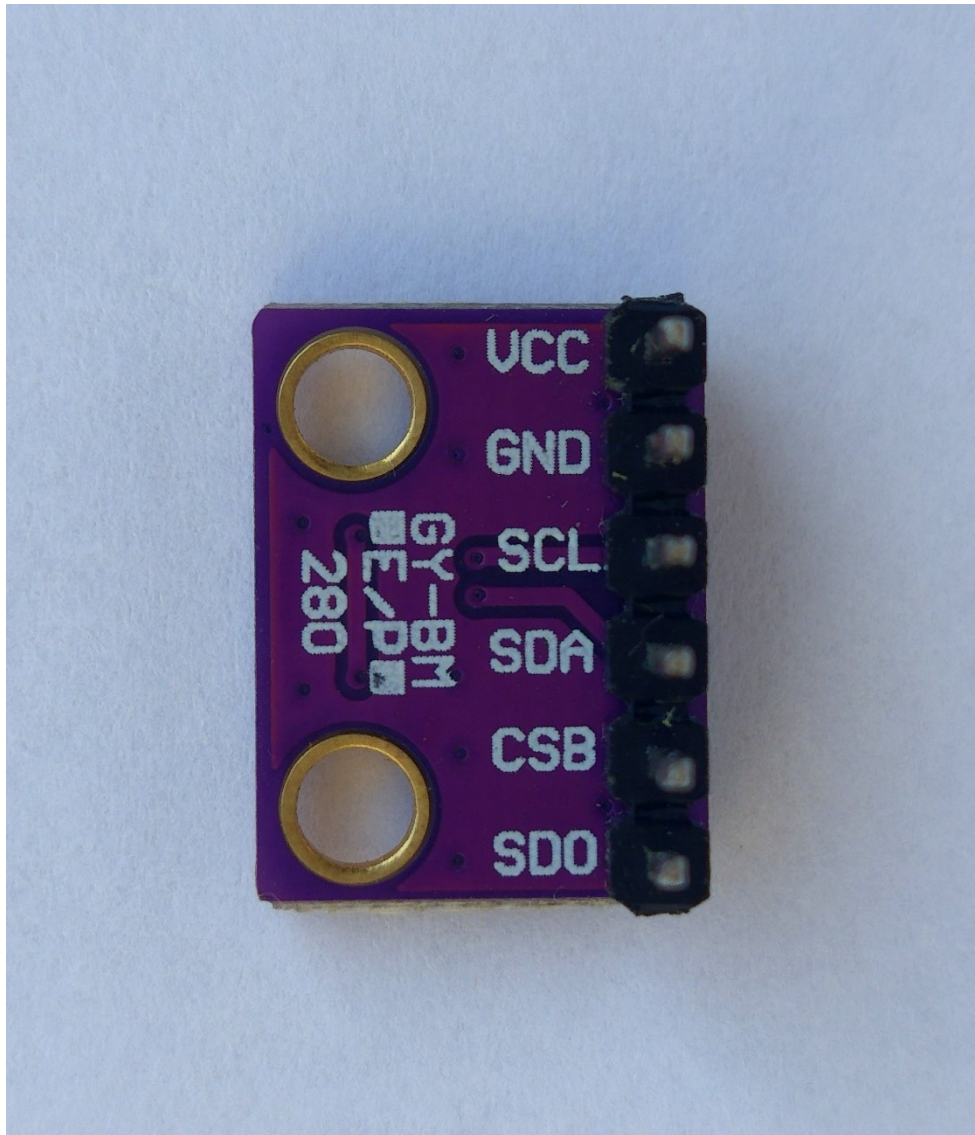


Slika 3.2. Modul GY-BMP280

Tablica 3.1. Tehničke karakteristike senzora BMP280 tvrtke Bosch

Parametar	Najmanja vrijednost	Najveća vrijednost	Mjerna jedinica
Napon napajanja	1.71	3.6	V
Vršna vrijednost struje	0.72	1.120	mA
Raspon atmosferskog tlaka	300	1100	hPa
Raspon temperature	-40	85	°C

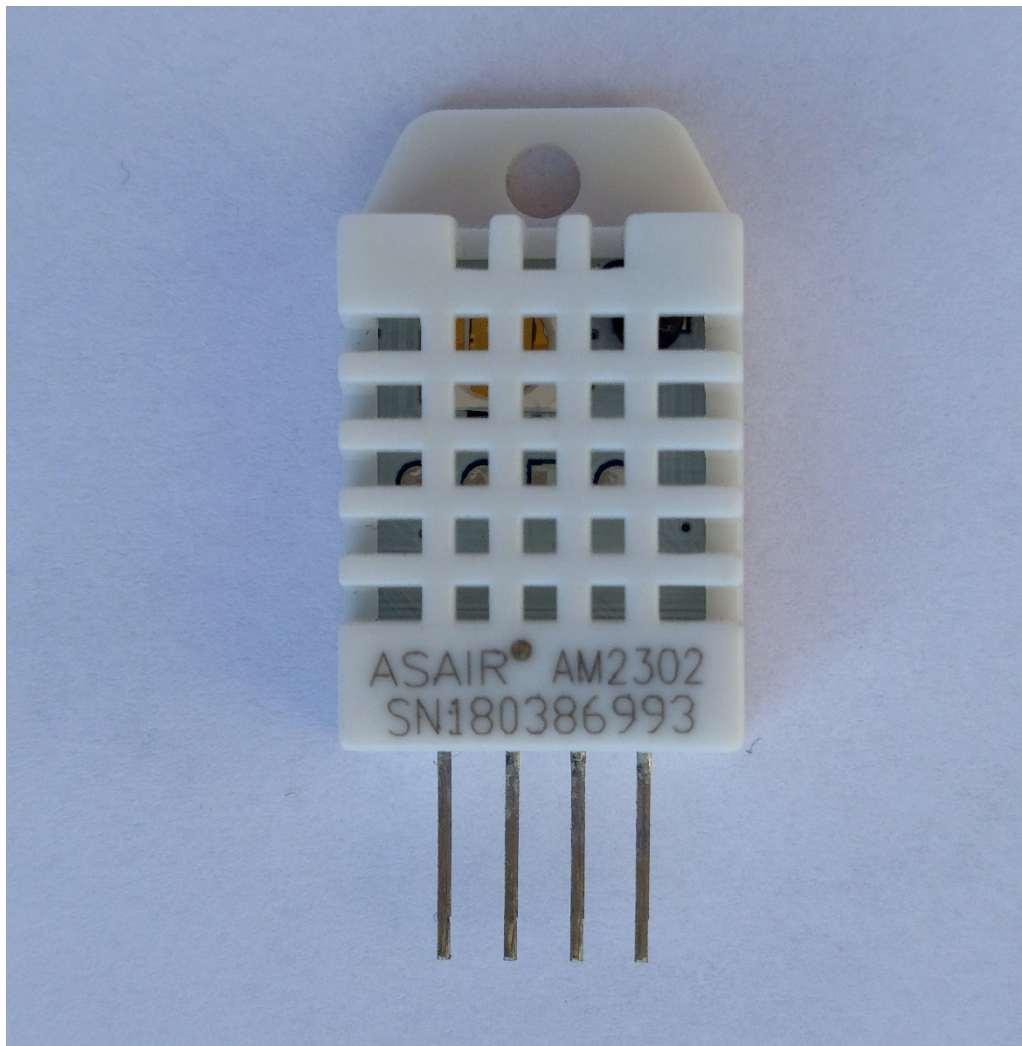
GY-BMP280 sadrži ukupno šest pinova koji služe za ostvarivanje komunikacije između senzora i mikrokontrolera. Raspored pinova modula GY-BMP280 prikazan je na slici 3.3. Pin za napajanje VCC i pin za uzemljenje GND upotrijebljuju se neovisno o komunikacijskom protokolu, dok se pinovi SCL, SDA, CSB i SDO koriste ovisno o željenom serijskom protokolu.



Slika 3.3. Raspored pinova modula GY-BMP280

3.3. DHT22

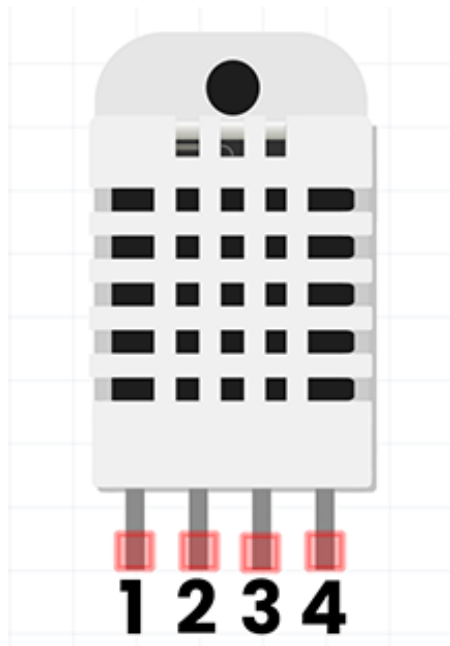
DHT22 je mjerno osjetilo temperature i relativne vlažnosti zraka koje proizvodi tvrtka Aosong Electronic. Uobičajena izvedba DHT22 senzora uključuje istoimeni modul (slika 3.4.). Radi se o pristupačnom senzoru visoke preciznosti i točnosti, a zbog jednostavnosti rukovanja odlikuje ga široka primjena. DHT22 objedinjuje kapacitivni senzor relativne vlažnosti i klasični termistor za mjerenje temperature. Glavni nedostatak senzora je kašnjenje u trajanju od dvije sekunde u očitavanju trenutnih vrijednosti mjerenja [29, 30].



Slika 3.4. Modul DHT22

Komunikacija između DHT22 i vanjskog mikrokontrolera odvija se putem samo jednog podatkovnog vodiča. Uređaj nije građen za standardne serijske komunikacijske protokole poput UART, SPI ili I²C [31, 32].

Raspored pinova modula DHT22 prikazan je na slici 3.5., dok tablica 3.2. sadrži tehničke karakteristike senzora DHT22 tvrtke Aosong Electronic.



- 1.** Pin za napajanje
- 2.** Podatkovni pin
- 3.** Pin koji se nigdje ne spaja
- 4.** Pin za uzemljenje

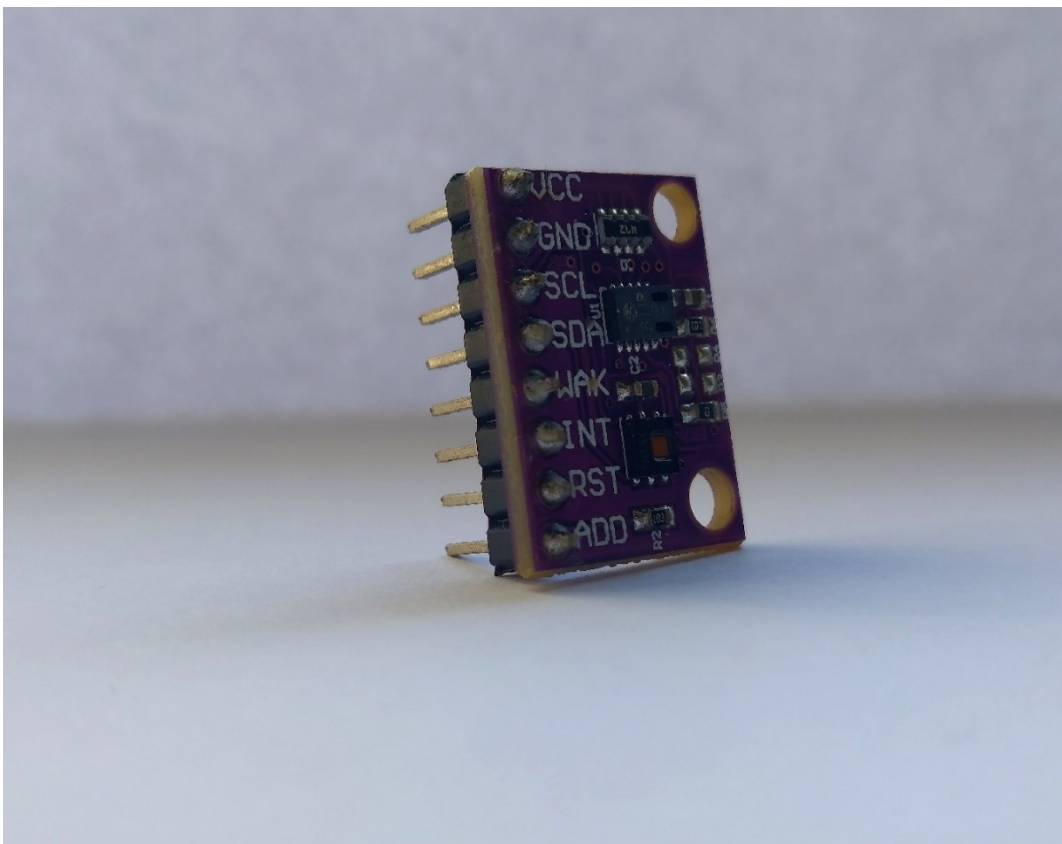
Slika 3.5. Raspored pinova modula DHT22 [33]

Tablica 3.2. Tehničke karakteristike senzora DHT22 tvrtke Aosong Electronic

Parametar	Najmanja vrijednost	Najveća vrijednost	Mjerna jedinica
Napon napajanja	3.3	6	V
Potrošnja struje	\	2.5	mA
Raspon relativne vlažnosti	0	100	%
Odstupanje relativne vlažnosti	±2	±5	%
Raspon temperature	-40	80	°C
Odstupanje temperature	\	±0.5	°C

3.4. CJMCU-811

Modul CJMCU-811 sadrži senzor kvalitete zraka CCS811 tvrtke ScioSense. Senzor je namijenjen za mjerenje kvalitete zraka unutar zatvorenog prostora, što podrazumijeva mjerenje količine hlapivih organskih spojeva (eng. TVOC - *total volatile organic compound*) u širokom rasponu, a to uključuje eCO₂ (ekvivalentni CO₂) te MOX (metalni oksidi). Hlapivi organski spojevi definirani su kao zagađivači zraka, a mogu dolaziti iz raznih izvora i javiti se kao posljedica primjerice: disanja, izgaranja, bojanja, lakiranja, varenja i općenito svih procesa koji otpuštaju štetne čestice u okolni zrak. Modul CJMCU-811 prikazan je na slici 3.6. [34, 35].



Slika 3.6. Modul CJMCU-811

Raspon mjerenja koncentracije eCO₂ iznosi od 400 do 8192 ppm (eng. *parts per million*), a za TVOC mjerenja od 0 do 1187 ppb (eng. *parts per billion*). Komunikacija s vanjskim uređajima vrši se putem I²C komunikacijskog protokola.

Modul sadrži 8 pinova, a uloga svakog od njih opisana je u nastavku. Uređaj se napaja istosmjernim naponom u iznosu od 1.8 V – 3.3 V putem VCC pina, dok se GND pin spaja na masu. SCL i SDA služe za uspostavu serijske I²C komunikacije, međutim razmjena podataka s vanjskim uređajima moguća je isključivo ako je WAK pin u stanju niske naponske razine. Kada bi WAK bio u stanju visoke naponske razine tada bi uređaj stalno bio u stanju mirovanja tj. ne bi mogao obavljati svoju funkciju. Završetkom pojedinog ciklusa mjerenja, senzor na izlazu INT pina daje povratnu informaciju što uobičajeno uključuje potvrdu o izvršenim mjerenjima. Pin RST služi za resetiranje senzora, dok pin ADD mijenja I²C adresu senzora [34]. Tehničke karakteristike senzora CCS811 tvrtke ScioSense navedene su u tablici 3.3.

Tablica 3.3. Tehničke karakteristike senzora CCS811 tvrtke ScioSense

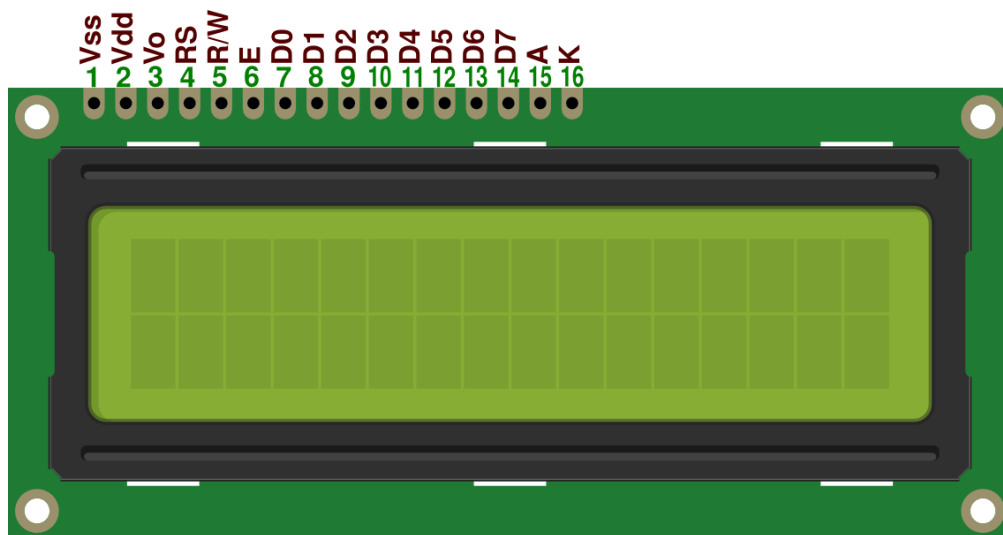
Parametar	Uvjeti	Uobičajena vrijednost	Najveća vrijednost	Mjerna jedinica
Napon napajanja	\	1.8 – 3.3	3.3	V
Struja izvora	Aktivno stanje pri 1.8 V	26	\	mA
	Stanje mirovanja pri 1.8 V	19	\	μA
Potrošnja snage	Aktivno stanje pri 1.8 V	1.2	\	mW
	Stanje mirovanja pri 1.8 V	0.034	\	mW

Sve navedeno čini ovaj uređaj vrlo jednostavnim za korištenje, uz brojne primjene u svakodnevnom životu. Praktično je rješenje za izradu uređaja koji ima zadatak kontinuiranog ispitivanja kvalitete zraka unutar manjih prostora gdje je redovita kontrola koncentracije štetnih čestica u okolnom zraku nužna zbog ljudskog zdravlja.

3.5. LCD 16x2 zaslon s HD44780 kontrolerom

U zaslonu s tekućim kristalima (eng. LCD – *Liquid Crystal Display*) slika nastaje polarizacijom i zaustavljanjem dijela svjetlosti prolaskom kroz tanki sloj tekućega kristala smješten između dviju polarizacijskih ploča, a unutrašnja orijentacija tekućih kristala i njihova propusnost mijenjaju se pomoću vanjskog električnog polja. Postoji mnoštvo različitih izvedbi LCD zaslona počevši od zaslona manjih veličina s manjim brojem redaka i stupaca, preko naprednijih zaslona većih dimenzija pa sve do onih koji omogućuju upravljanje pomoću dodira. Za potrebe izrade završnog rada dovoljan je LCD zaslon koji omogućuje prikaz ukupno 32 znakova u 2 retka, uz ograničenje od 16 znakova po retku. LCD 16x2 zaslon koristi HD44780 kontroler koji radi na principu oblikovanja standardnih skupova znakova pomoću točkastih matrica [36, 37, 38].

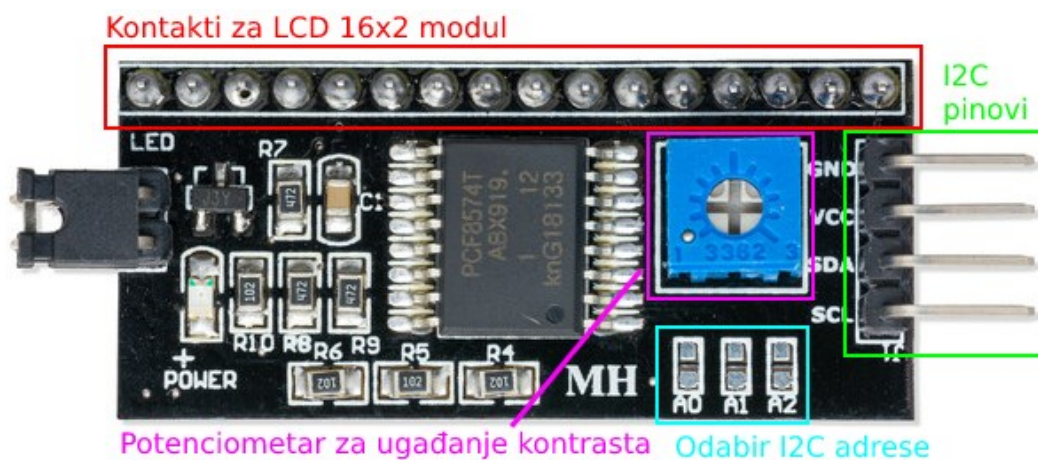
Osnovna izvedba LCD 16x2 zaslona uključuje sučelje s ukupno 16 kontakata, najčešće izvedenih pomoću pinova. Među njih spadaju pinovi za napajanje (V_{ss} , V_{dd}), pin za ugađanje kontrasta (V_o), pin za odabir registra (R_s), pin za pisanje, odnosno čitanje sa zaslona (R/W), pin radnog takta (E), niz podatkovnih pinova ($D_0 - D_7$) te pinovi za ugađanje pozadinskog svijetla (A , K), prema slici 3.7. [37, 38].



Slika 3.7. LCD 16x2 zaslon s HD44780 kontrolerom [39]

3.5.1. I²C adapter za LCD 16x2 zaslon

Komunikacija LCD 16x2 zaslona s vanjskim mikrokontrolerom putem podatkovnih pinova D₀ – D₇ nije učinkovita uslijed velikog broja pinova kojima mikrokontroler treba upravljati. Kako bi konačno ožičenje bilo što preglednije, uobičajena je praksa spajanja I²C adaptera koji omogućuje uspostavu I²C komunikacije s mikrokontrolerom. Izgled I²C adaptera prikazan je na slici 3.8. Uređaj sadrži potenciometar za ugađanje pozadinskog osvjetljenja (kontrasta) na zaslonu te preklopke koje korisniku pružaju mogućnost odabira I²C adrese. I²C adapter najčešće se ugrađuje sa stražnje strane modula radi preglednosti (slika 3.9.).



Slika 3.8. I²C adapter za LCD 16x2 zaslon [40]



Slika 3.9. I²C adapter ugrađen na pozadini LCD 16x2 zaslona

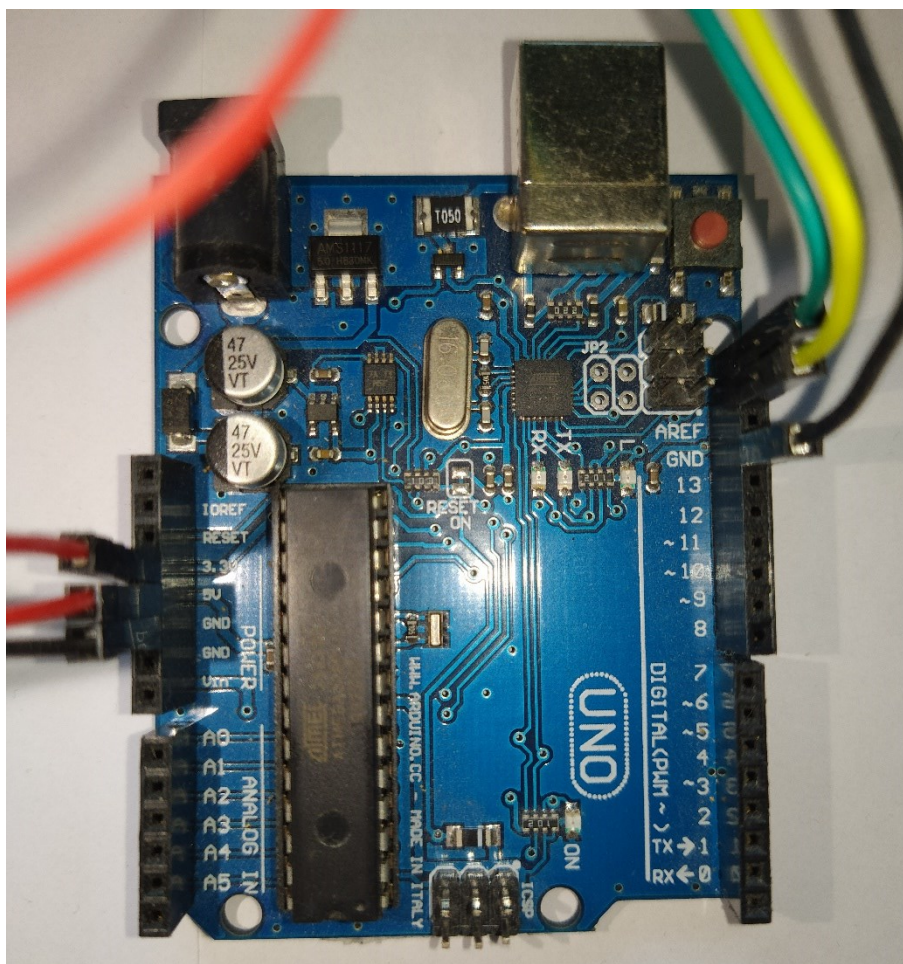
4. IZRADA ARDUINO METEOROLOŠKE STANICE

Izrada meteorološke stanice temeljene na Arduino platformi podrazumijeva njezino ostvarivanje u smislu povezivanja odgovarajućih uređaja u jedinstven sustav, kako s praktične (sklopovlje), tako i s programske strane (programska podrška).

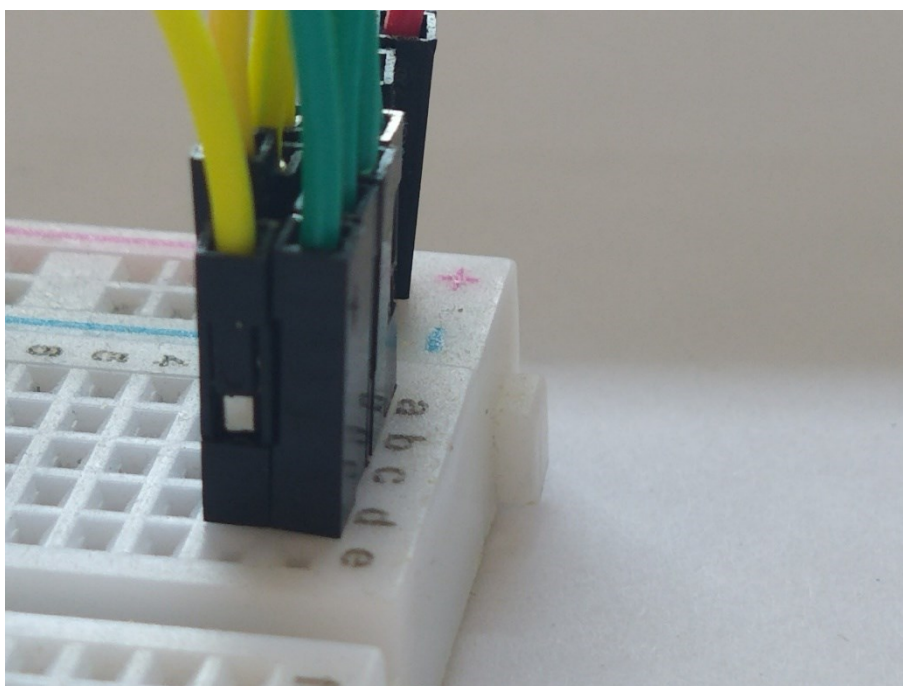
4.1. Postavljanje Arduino Uno R3 pločice

4.1.1. Odabir I²C ulaza i njihovo spajanje na eksperimentalnu pločicu

Prilikom spajanja dosta pažnje posvećeno je odabiru vodiča kako bi konačno ožičenje bilo što moguće preglednije. Za I²C sabirnice odabrani su vodiči zelene i žute boje. Zeleni vodiči predstavljaju SCL vodič, a žuti vodiči SDA vodič. Sukladno tome slijedi povezivanje I²C ulaza s eksperimentalnom pločicom kako bi se na istoj pločici omogućilo spajanje vanjskih uređaja, a samim time i njihova uspostava komunikacije s mikrokontrolerom preko izvedenih I²C sabirnica, odnosno SCL i SDA linija na pločici (slika 4.2). Na eksperimentalnu pločicu također se dovode napajanja u iznosu od 3.3V i 5V u točno predviđena mjesta, pomoću vodiča crvene boje. Vodiči crne boje predstavljaju masu. Izgled Arduino Uno R3 pločice nakon spajanja vodiča na I²C ulaze te na ulaze za napajanje i masu prikazan je na slici 4.1.



Slika 4.1. Izgled Arduino Uno R3 pločice nakon spajanja vodiča na I^2C ulaze te na ulaze za napajanje i masu



Slika 4.2. Izgled eksperimentalne pločice nakon spajanja I^2C vodiča i vodiča napajanja

4.1.2. Stvaranje Arduino programske datoteke i uključivanje I²C biblioteke

Postupak stvaranja Arduino programske datoteke te njezino pohranjivanje, kao i postupak uključivanja programskih biblioteka, opisanu su u poglavlju broj dva. U ovom poglavlju iznose se osnovne informacije o I²C programskoj biblioteci, kao i izgled programske datoteke nakon njezina učitavanja.

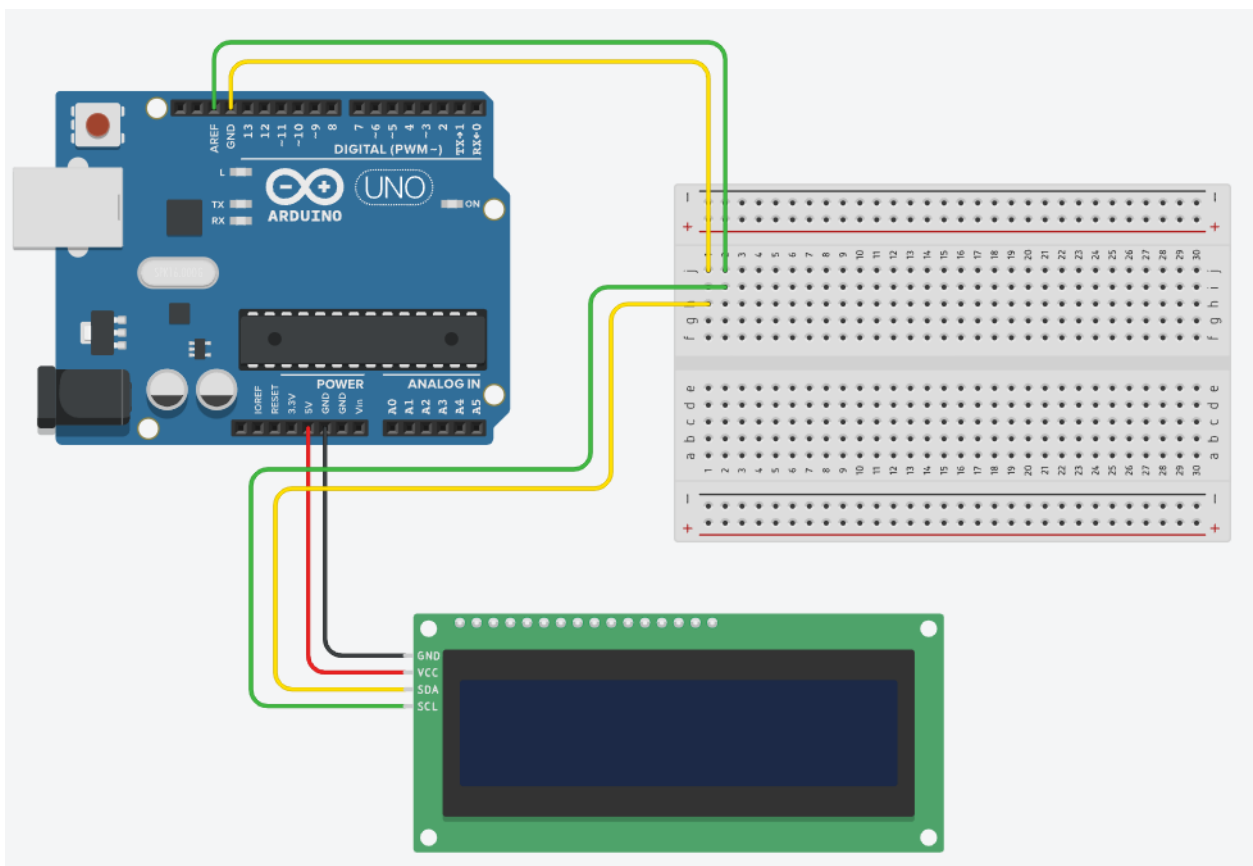
I²C programska biblioteka naziva Wire sadrži programski kod koji omogućuje I²C serijsku komunikaciju između Arduina i vanjskih uređaja koji podržavaju navedeni protokol. Biblioteka pruža niz raznih funkcija za rad, primjerice funkcije za prijenos, prijem, očitavanje podataka i sl. Na vrhu skice, iznad svih funkcija i definicija, unosi se sljedeća naredba koja osigurava ispravno učitavanje Wire biblioteke: [41]

```
#include <Wire.h>
```

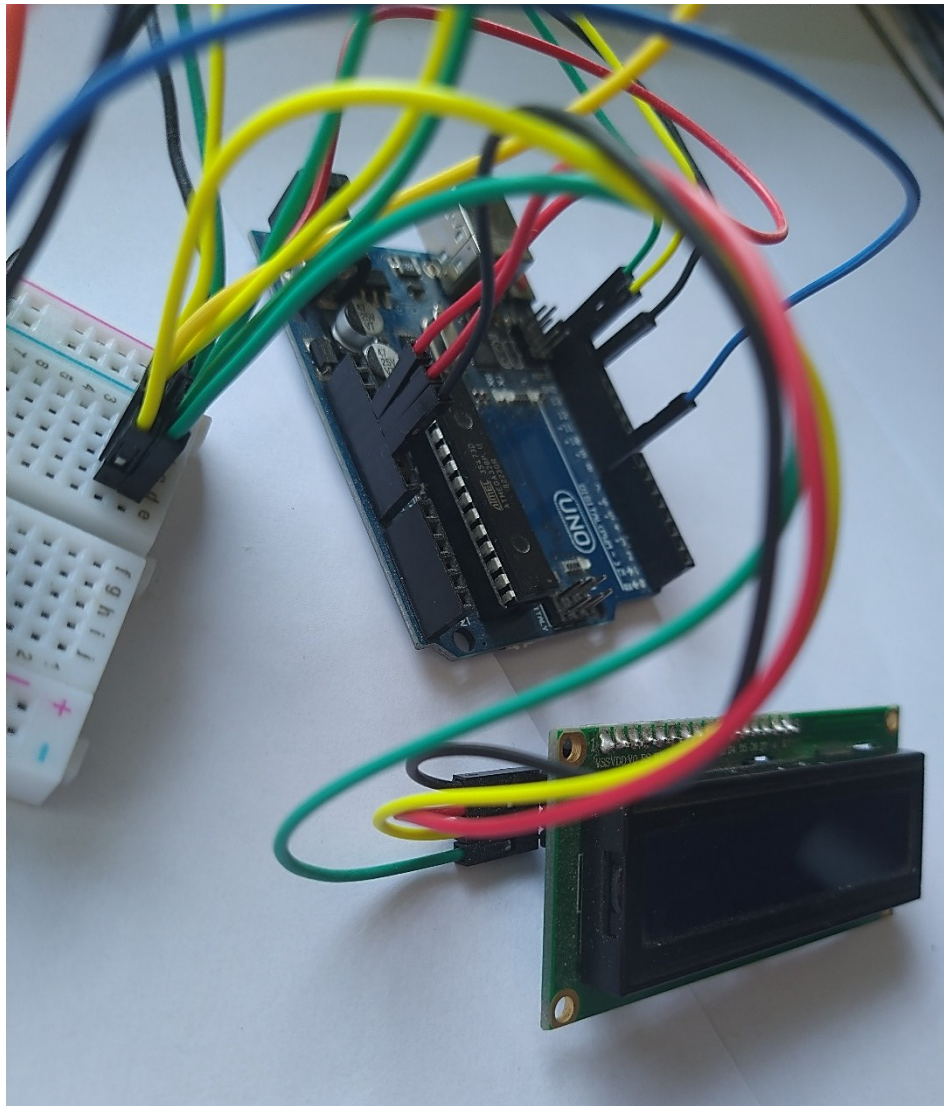
4.2. Povezivanje i konfiguriranje LCD 16x2 zaslona

Redosljed spajanja vanjskih uređaja za rad nije nužno bitan, stoga prvo slijedi konfiguracija LCD 16x2 zaslona na kojemu će se prikazivati vrijednosti mjerenih fizikalnih veličina. Na taj način moguće je provesti kratku provjeru ispravnosti rada svakog idućeg uređaja (senzora).

LCD 16x2 zaslon s I²C adapterom nije namijenjen za ugradnju na eksperimentalnu pločicu poput primjerice senzora, čipova, otpornika, kondenzatora i sl. Međutim, pinove I²C adaptera moguće je vrlo jednostavno spojiti na eksperimentalnu pločicu pomoću vodiča (slika 4.4.). Za ispravan rad zaslona potrebno je spojiti sva četiri pina I²C adaptera. Pin za napajanje VCC spaja se na 5V, GND na masu, a SCL i SDA na I²C ulaze izvedene pomoću eksperimentalne pločice (slika 4.3).



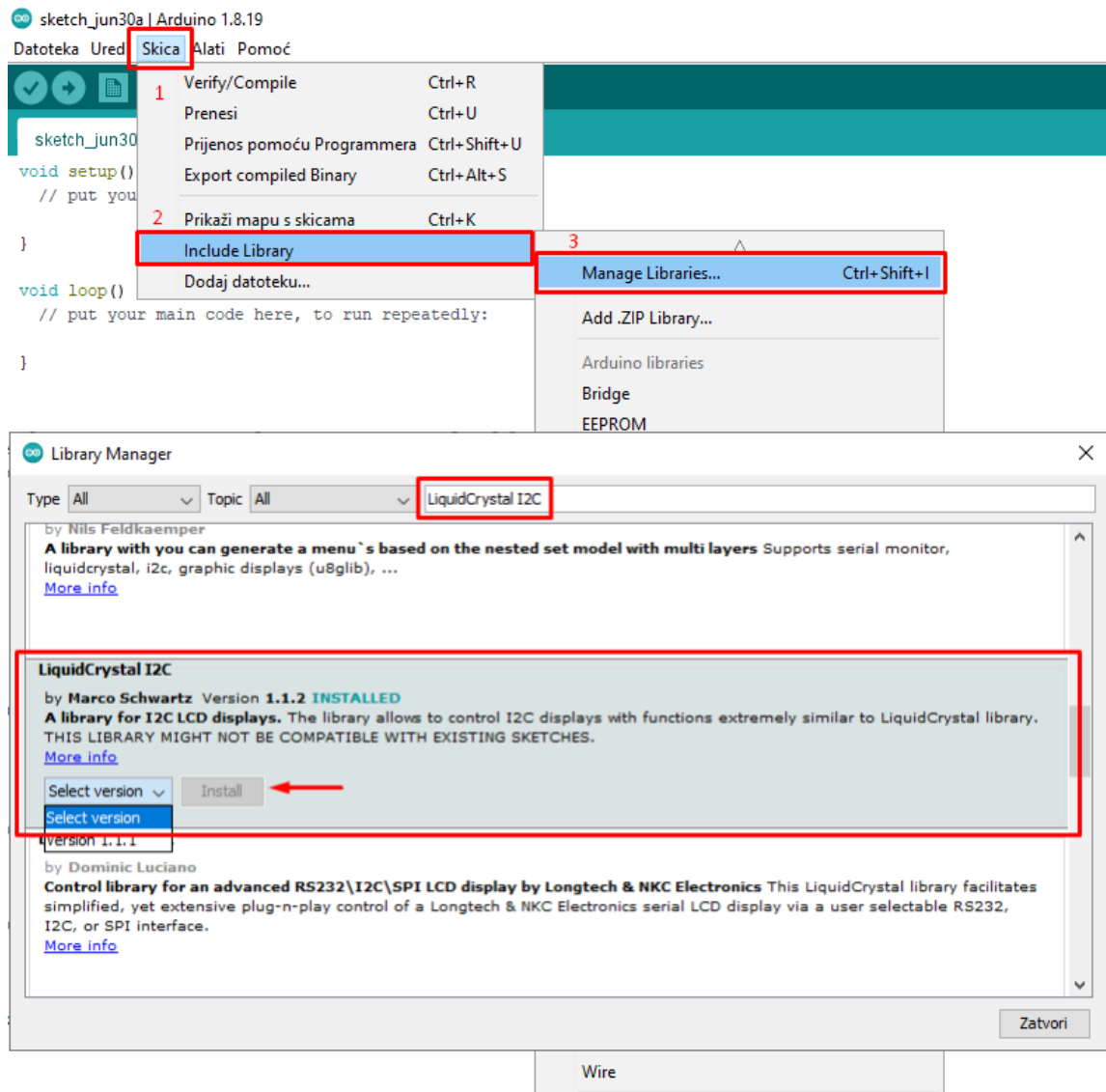
Slika 4.3. Shema spajanja LCD 16x2 zaslona s I²C adapterom izrađena u aplikaciji TinkerCad



Slika 4.4. Povezivanje LCD 16x2 zaslona s I²C adapterom s Arduinoom pomoću eksperimentalne pločice

4.2.1. Konfiguriranje LCD 16x2 zaslona s I²C adapterom u Arduino razvojnom okruženju

Kako bi uređaj obavljao svoj zadatak ispisa vrijednosti mjerenih fizikalnih veličina u sklopu Arduino meteorološke stanice, nužno je to omogućiti unutar programskog koda. Programska biblioteka naziva LiquidCrystal I2C jedna je od standardnih biblioteka unutar Arduino razvojnog okruženja. U Arduino skicu dodaje se putem upravitelja biblioteka u skladu s koracima prikazanim na slici 4.5.



Slika 4.5. Uključivanje LiquidCrystal I2C biblioteke u Arduino skicu

Naposlijetku se pri samom vrhu skice, tamo gdje se nalaze ostale naredbe za učitavanje biblioteka, unosi sljedeća naredba:

```
#include <LiquidCrystal_I2C.h>
```

U nastavku je prikazan programski kod u kojem se primjenjuju osnovne funkcije iz LiquidCrystal I2C biblioteke u svrhu provjere ispravnosti rada zaslona. Na slici 4.6. prikazan je izgled LCD 16x2 zaslona pri ispisu. Objašnjene pojedinih funkcija navedeno je u programskom kodu u obliku kratkih komentara.

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C LCD(0x27, 16, 2); // stvaranje programske instance LCD
zaslona putem I2C komunikacijskog protokola

void setup() {
  LCD.init(); // pozivanje funkcije za inicijalizaciju LCD 16x2 zaslona
  LCD.backlight(); // funkcija backlight() pali pozadinsko svijetlo na LCD
zaslonu

  LCD.setCursor(0, 0); // odabir polja i retka za ispis znakova (0, 0) ->
prvo polje u prvom retku
  LCD.print("Arduino based"); // ispis na zaslonu sukladno prethodnom
odabiru
  LCD.setCursor(0, 1); // odabir polja i retka za ispis znakova (0, 1) ->
prvo polje u drugom retku
  LCD.print("weather station"); // ispis na zaslonu sukladno prethodnom
odabiru
}

void loop() {
  // put your main code here, to run repeatedly:
}
```



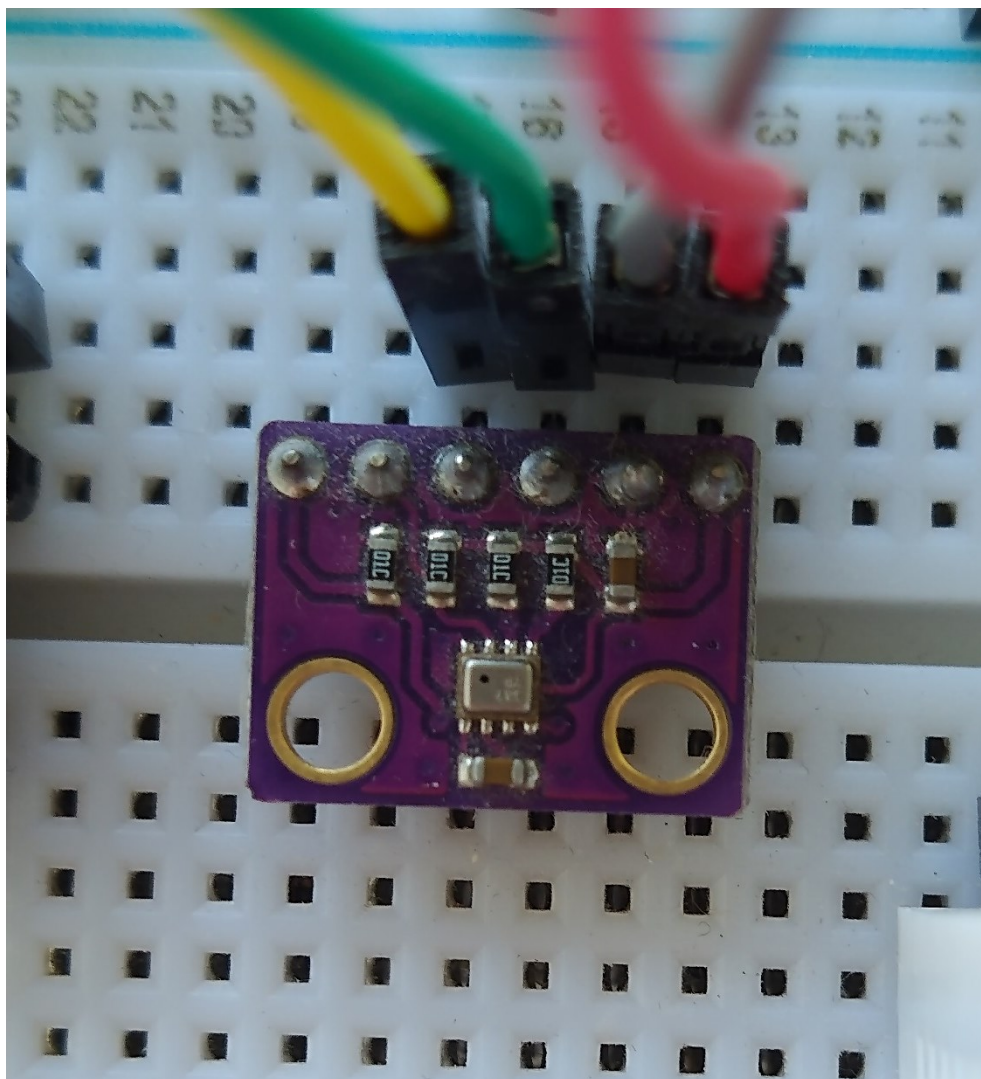
Slika 4.6. Prikaz LCD 16x2 zaslona pri ispisu u svrhu provjere ispravnosti uređaja

4.3. Povezivanje senzora BMP280

Modul GY-BMP280 moguće je smjestiti na eksperimentalnu pločicu. Uređaj zauzima vrlo malo prostora s obzirom na uobičajene veličine pločica, stoga je pogodan i za smještaj na eksperimentalne pločice manjih dimenzija.

4.3.1. Spajanje modula GY-BMP280 na eksperimentalnu pločicu

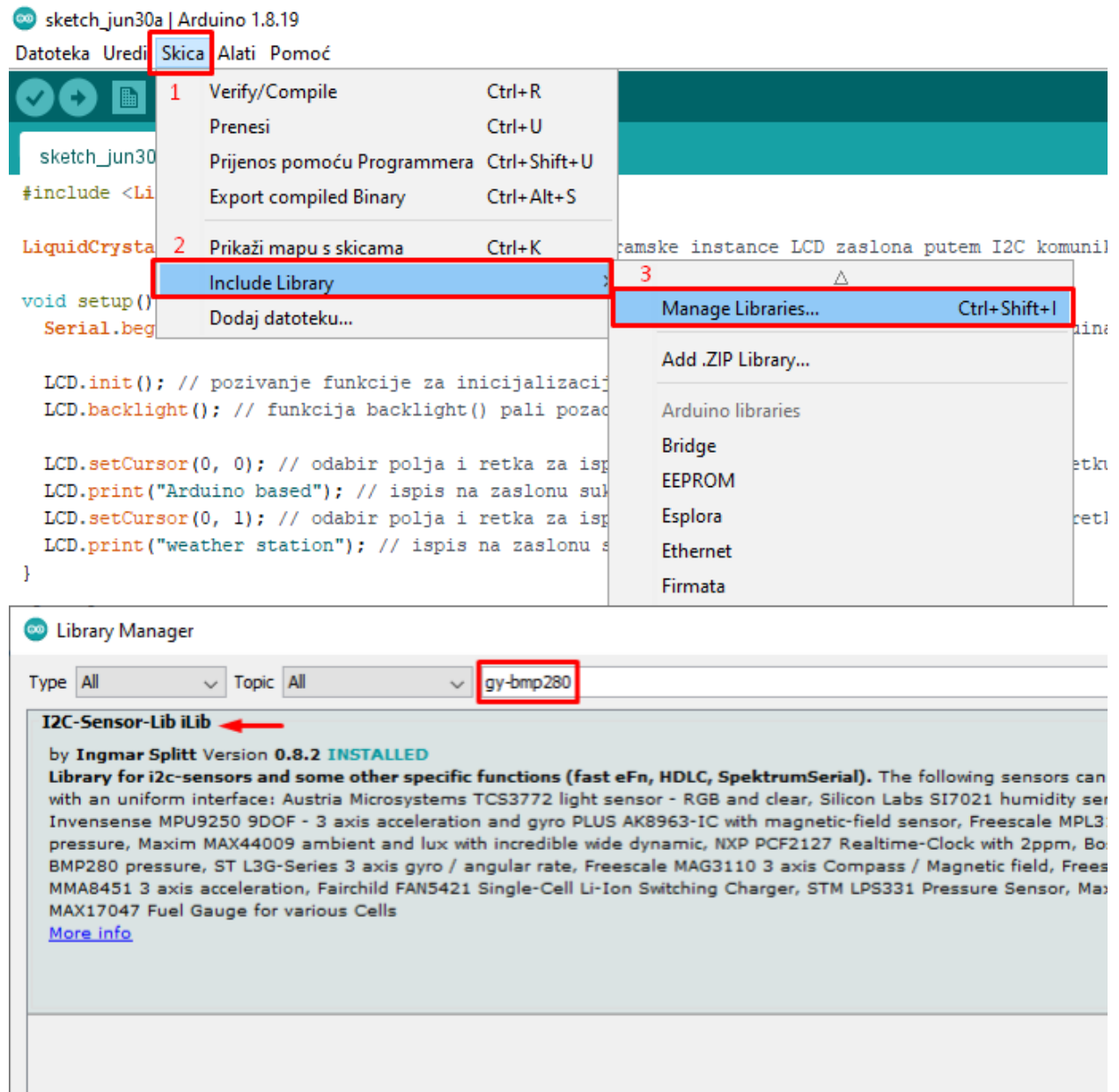
Modul ima šest pinova, od kojih se dva odnose na napajanje i masu, a preostala četiri služe za uspostavu željenog serijskog komunikacijskog protokola. Odabrani I²C pinovi (SCL, SDA) spajaju se na odgovarajuće sabirnice pomoću vodiča. Preostala dva pina (CSB, SDO) nisu spojeni. Slika 4.7. prikazuje modul GY-BMP280 spojen na eksperimentalnu pločicu.



Slika 4.7. Prikaz modula GY-BMP280 spojenog na eksperimentalnu pločicu

4.3.2. Učitavanje programske biblioteke i programiranje uređaja u Arduino razvojnom okruženju

Kako bi uređaj imao pravilnu programsku podršku potrebno je učitati ispravnu programsku biblioteku u Arduino skicu. Na slici 4.8. prikazan je postupak učitavanja odgovarajuće biblioteke u skicu.



Slika 4.8. Uključivanje I2C-Sensor-Lib biblioteke koja obuhvaća senzor BMP280 u Arduino skicu

Slijedi unos naredbe za učitavanje prethodno uključene biblioteke u skicu. Naredba se unosi pri vrhu skice iznad svih funkcija i definicija.

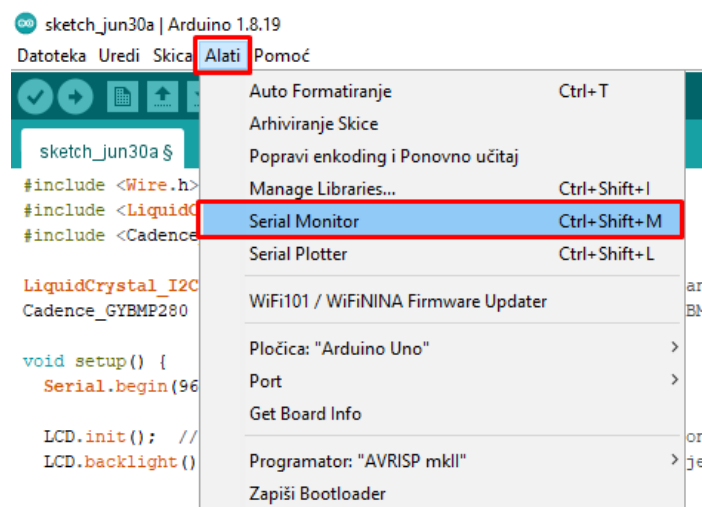
```
#include <Cadence_GYBMP280.h>
```

Komunikacija sa senzorom neće biti moguća ukoliko se ne stvori programski objekt ili instanca koja odgovara jednom senzoru. Kada bi postojao i dodatni istovrsni senzor, tada bi bilo potrebno stvoriti i dodatni programski objekt koji bi predstavljao taj senzor. Naredba koja je odgovorna za stvaranje nove instance unosi se ispod naredbi za učitavanje biblioteka i definicija, ali izvan svih funkcija.

```
Cadence_GYBMP280 GY_BMP280;
```

Senzor se inicijalizira naredbom *begin*. Pozivom funkcije *readTemperatureC* uređaj započinje mjerenje temperature zraka. Funkcija po završetku mjerenja vraća izmjerenu vrijednost u Celzijevim stupnjevima (°C). Za mjerenje atmosferskog tlaka zadužena je funkcija *readPressure* koja kao povratnu informaciju vraća izmjerenu vrijednost u paskalima (Pa). Sljedeći isječak programskog koda zadužen je za slanje povratne informacije putem korisničkog sučelja pod nazivom Serial Monitor u slučaju pogrešnog ožičenja ili neispravnog rada modula GY-BMP280. Sučelju je moguće pristupiti putem izbornika *Alati*, kao što je to prikazano na slici 4.9.

```
if(!GY_BMP280.begin()) {  
  Serial.println("[ERROR] Senzor BMP280 nije pronaden... Ispitajte  
  ispravnost senzora i spojnih vodica.");  
  while(1);  
}
```



Slika 4.9. Pristup korisničkom sučelju Serial Monitor

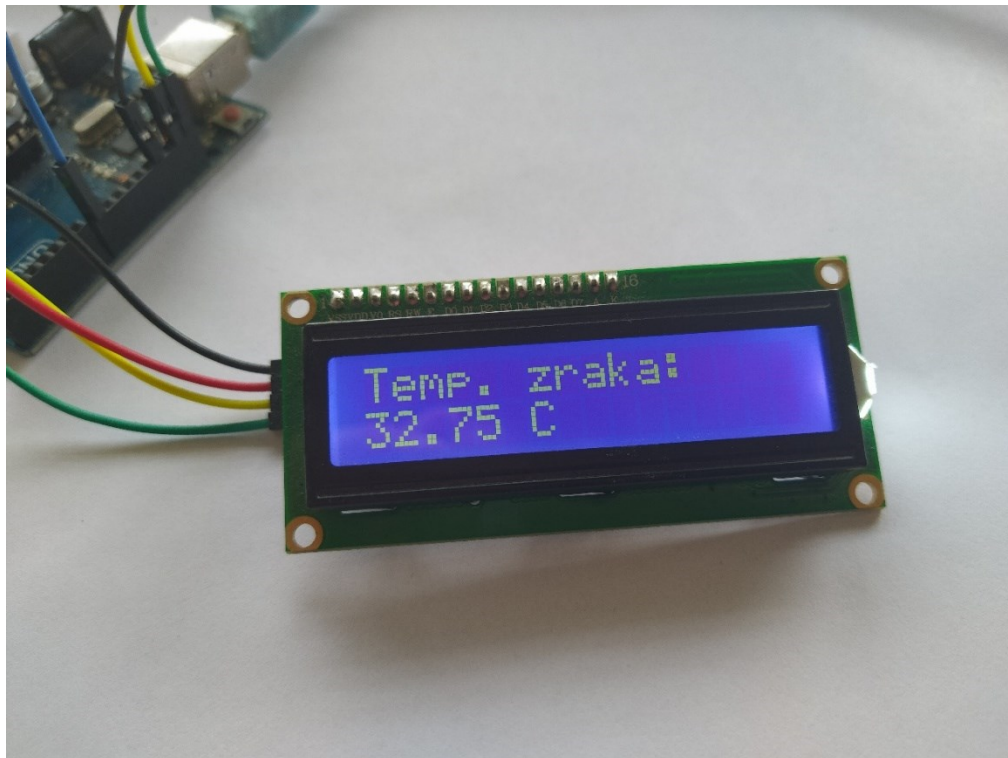
Bitno je istaknuti kako se sva mjerenja odvijaju unutar osnovne funkcije *loop* jer se programski kod svakom iteracijom iznova izvršava. U nastavku je naveden programski kod koji omogućuje mjerenje temperature zraka i atmosferskog tlaka s prikazom izmjerenih vrijednosti na LCD 16x2 zaslonu. Slika 4.10. prikazuje ispis vrijednosti temperature zraka, dok slika 4.11. prikazuje ispis vrijednosti atmosferskog tlaka.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Cadence_GYBMP280.h>

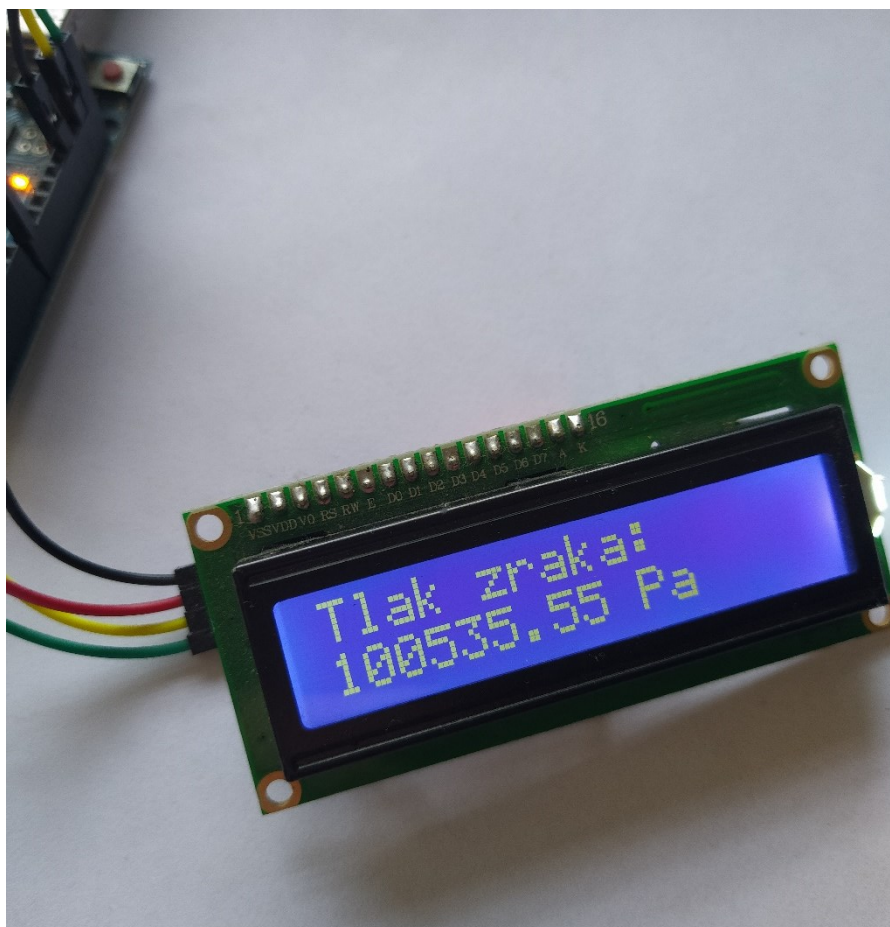
LiquidCrystal_I2C LCD(0x27, 16, 2); // stvaranje programske instance LCD
Cadence_GYBMP280 GY_BMP280; // stvaranje programske instance BMP280

void setup() {
  Serial.begin(9600);
  LCD.init(); // pozivanje funkcije za inicijalizaciju LCD zaslona
  LCD.backlight(); // funkcija "backlight()" pali pozadinsko svijetlo na
LCD zaslonu
  if(!GY_BMP280.begin()) {
    Serial.println("[ERROR] Senzor BMP280 nije pronaden... Ispitajte
ispravnost senzora i spojnih vodica.");
    while(1);
  }
  GY_BMP280.setDefaultQNH(1013.25); // prosjecni atmosferski tlak na
morskoj razini
  LCD.setCursor(0, 0);
  LCD.print("Arduino based");
  LCD.setCursor(0, 1);
  LCD.print("weather station");
  delay(5000);
  LCD.clear();
}

void loop() {
  // GY-BMP280: temperatura zraka
  LCD.setCursor(0, 0); // ispis zapocinje na nultom polju prvog retka LCD
zaslona
  LCD.print("Temp. zraka:");
  LCD.setCursor(0, 1); // ispis zapocinje na nultom polju drugog retka LCD
zaslona
  LCD.print(GY_BMP280.readTemperatureC());
  LCD.print(" C");
  delay(5000);
  LCD.clear();
  // GY-BMP280: tlak zraka
  LCD.setCursor(0, 0);
  LCD.print("Tlak zraka:");
  LCD.setCursor(0, 1);
  LCD.print(GY_BMP280.readPressure()); // funkcija "readPressure()" kao
povratnu informaciju daje relativni tlak zraka [hPa]
  LCD.print(" hPa");
  delay(5000);
  LCD.clear();
  delay(100);
}
```

Slika 4.10. Ispis vrijednosti temperature zraka na LCD 16x2 zaslonu

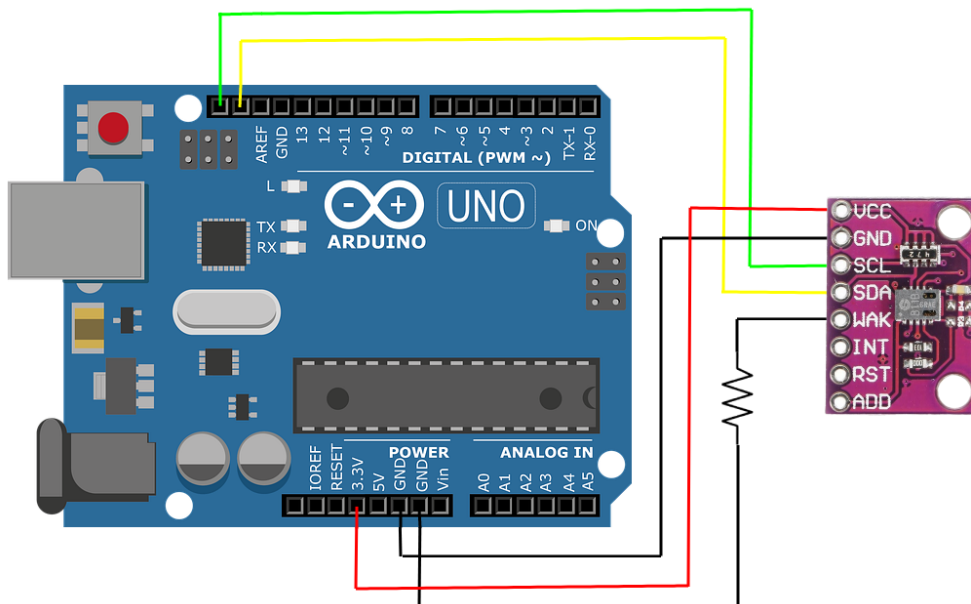


Slika 4.11. Ispis vrijednosti atmosferskog tlaka na LCD 16x2 zaslonu

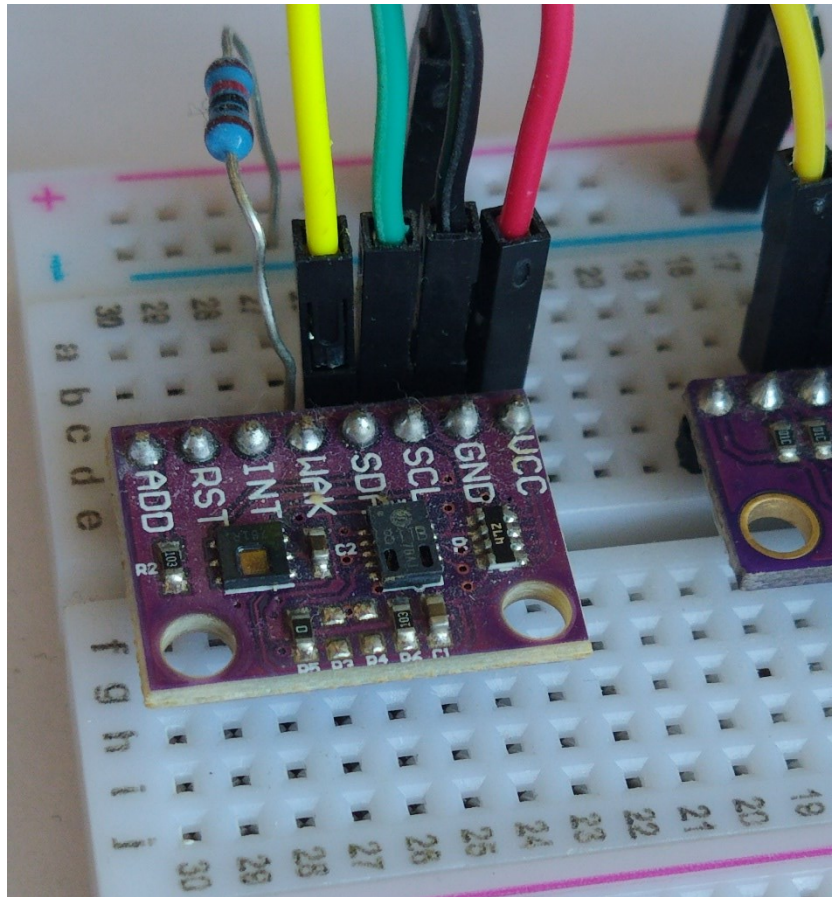
4.4. Povezivanje senzora CCS811

4.4.1. Spajanje modula CJMCU-811 na eksperimentalnu pločicu

Uređaj komunicira putem I²C protokola. VCC pin spaja se na 3.3V istosmjernog napona, GND na masu, a SCL i SDA pinovi na odgovarajuće I²C sabirnice. Mjerno osjetilo će obavljati svoj zadatak otkrivanja hlapivih organskih spojeva u okolnom zraku samo kada je WAK pin u stanju niske naponske razine, stoga se preko otpornika vrijednosti 10 kΩ spaja na masu. Shema izravnog spajanja modula CJMCU-811 prikazana je na slici 4.12., dok slika 4.13. prikazuje modul spojen na eksperimentalnu pločicu.



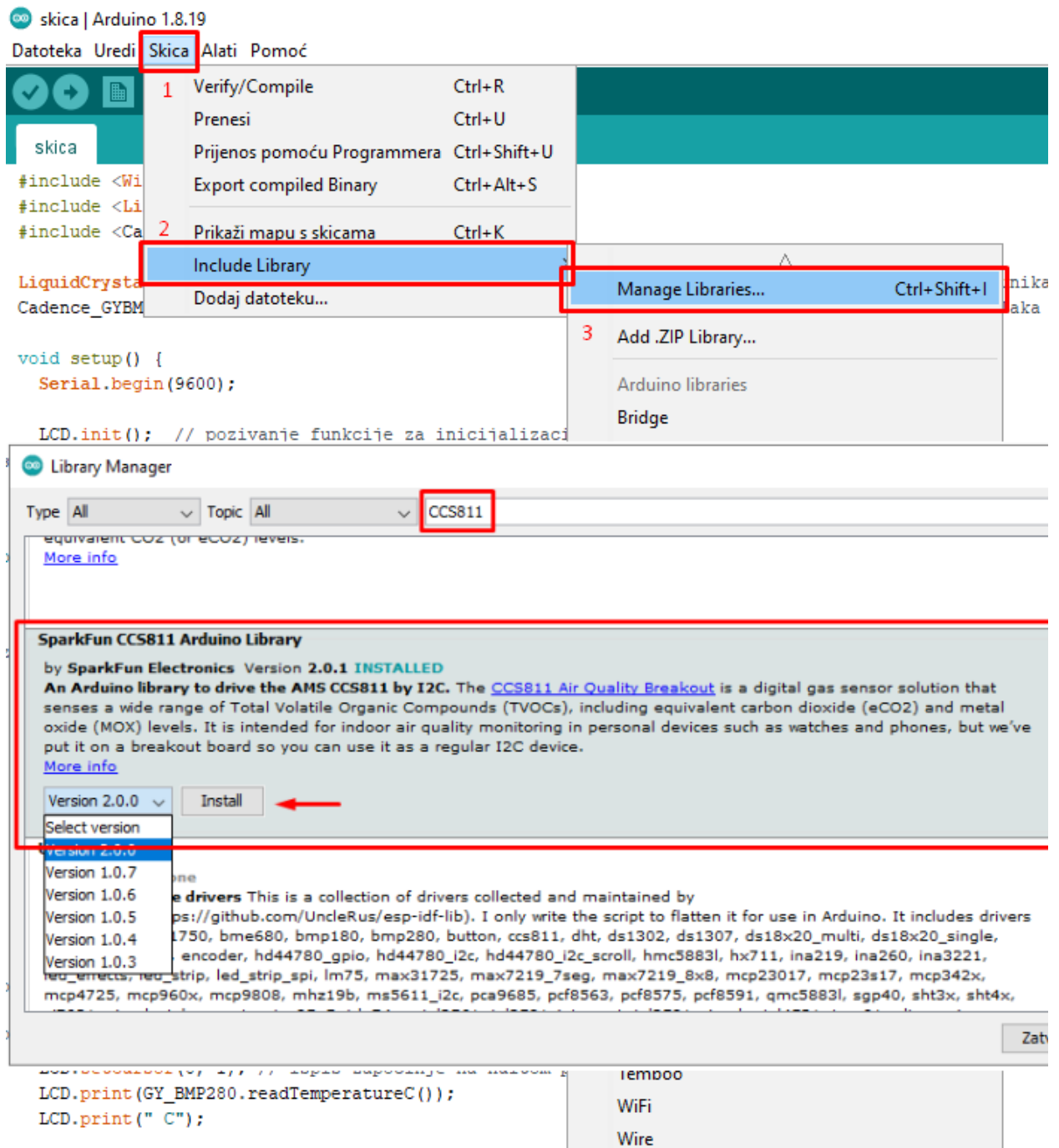
Slika 4.12. Shema izravnog spajanja modula CJMCU-811 izrađena u aplikaciji Photopea



Slika 4.13. Prikaz modula CJMCU-811 spojenog na eksperimentalnu pločicu

4.4.2. Učitavanje programske biblioteke i programiranje uređaja u Arduino razvojnom okruženju

Učitavanje odgovarajuće programske biblioteke senzora CCS811 u skicu prikazano je na slici 4.14.



Slika 4.14. Uključivanje SparkFun CCS811 biblioteke u Arduino skicu

Na vrhu Arduino skice, iznad svih funkcija i definicija, unosi se naredba za učitavanje prethodno uključene SparkFun CCS811 biblioteke:

```
#include <SparkFunCCS811.h>
```

U sljedećem koraku potrebno je stvoriti programsku instancu koja će fizički uređaj predstavljati u programskom okruženju. Naredbe se unose iznad svih funkcija, ali ispod naredbi za učitavanje programskih biblioteka:

```
#define CJMCU811_ADDR 0x5A // default I2C adresa CJMCU-811 (CCS811) senzora
CCS811 CJMCU811(CJMCU811_ADDR);
```

Unutar osnovne funkcije *setup* senzor je nužno inicijalizirati pozivom funkcije *begin*. Funkcije koje po izvršenju predstavljaju početak, odnosno kraj mjerenja, kao i funkcije zadužene za dohvat očitavanja, pozivaju se unutar osnovne funkcije *loop*. Uređaj treba kontinuirano raditi najmanje 15 minuta kako bi očitavanja bila dovoljno precizna. Pomoću funkcije *dataAvailable* moguće je provjeriti dostupnost novih podataka dobivenih mjerenjem. Funkcija se koristi zajedno s funkcijom *readAlgorithmResults* koja te podatke obrađuje. Rezultat obrade su izmjerene vrijednosti kojima se pristupa pomoću funkcija *getCO2* i *getTVOC*.

Sljedeći programski uvjet unosi se u osnovnu funkciju *setup* i služi za provjeru povezanosti CCS811 senzora:

```
if(!CJMCU811.begin()) {
  Serial.println("[ERROR] Senzor CCS811 nije pronaden... Ispitajte
  ispravnost senzora i spojnih vodica.");
  while(1);
}
```

Programski kod koji obuhvaća dohvat i ispis vrijednosti mjerenih fizikalnih veličina temperature zraka i atmosferskog tlaka te veličina kojima se mjeri kvaliteta zraka u okolnom prostoru, dan je na sljedećoj stranici. Treba napomenuti kako je programski kod za dohvat i ispis vrijednosti u slučaju senzora CCS811 smješten unutar *for* petlje, gdje se nove vrijednosti dohvaćaju i ispisuju u vremenskom razmaku od dvije sekunde. Ispis izmjerenih eCO2 i TVOC vrijednosti na LCD 16x2 zaslonu prikazan je na slici 4.15.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Cadence_GYBMP280.h>
#include <SparkFunCCS811.h>

LiquidCrystal_I2C LCD(0x27, 16, 2); // stvaranje programske instance LCD zaslona
Cadence_GYBMP280 GY_BMP280; // stvaranje programske instance BMP280 senzora
CCS811 CJMCU811(CJMCU811_ADDR); // stvaranje programske instance CCS811 senzora

void setup() {
  Serial.begin(9600);
  LCD.init(); // pozivanje funkcije za inicijalizaciju LCD zaslona
  LCD.backlight(); // funkcija "backlight()" pali pozadinsko svjetlo na LCD
  zaslonu

  if(!GY_BMP280.begin()) {
    Serial.println("[ERROR] Senzor BMP280 nije pronaden... Ispitajte ispravnost
  senzora i spojnih vodica.");
    while(1);
  }
  if(!CJMCU811.begin()) {
    Serial.println("[ERROR] Senzor CCS811 nije pronaden... Ispitajte ispravnost
  senzora i spojnih vodica.");
    while(1);
  }
  GY_BMP280.setDefaultQNH(1013.25); // prosjecni atmosferski tlak na morskoj razini
  LCD.setCursor(0, 0);
  LCD.print("Arduino based");
  LCD.setCursor(0, 1);
  LCD.print("weather station");
  delay(5000);
  LCD.clear();
}
void loop() {
  // GY-BMP280: temperatura zraka
  LCD.setCursor(0, 0); // ispis zapocinje na nultom polju prvog retka LCD zaslona
  LCD.print("Temp. zraka:");
  LCD.setCursor(0, 1); // ispis zapocinje na nultom polju drugog retka LCD zaslona
  LCD.print(GY_BMP280.readTemperatureC());
  LCD.print(" C");
  delay(5000);
  LCD.clear();
  // GY-BMP280: tlak zraka
  LCD.setCursor(0, 0);
  LCD.print("Tlak zraka:");
  LCD.setCursor(0, 1);
  LCD.print(GY_BMP280.readPressure()); // funkcija "readPressure()" kao povratnu
  informaciju daje relativni tlak zraka [hPa]
  LCD.print(" hPa");
  delay(5000);
  LCD.clear();

  for(int i = 0; i < 15; ++i) {
    if(CJMCU811.dataAvailable()) {
      CJMCU811.readAlgorithmResults();
    }
    // CJMCU-811: kvaliteta zraka
    LCD.setCursor(0, 0);
    LCD.print("eCO2: ");
    LCD.print(CJMCU811.getCO2()); // dohvat
    LCD.setCursor(0, 1);
    LCD.print("TVOC: ");
    LCD.print(CJMCU811.getTVOC());
    delay(2000);
    LCD.clear();
  }
  delay(100);
}

```



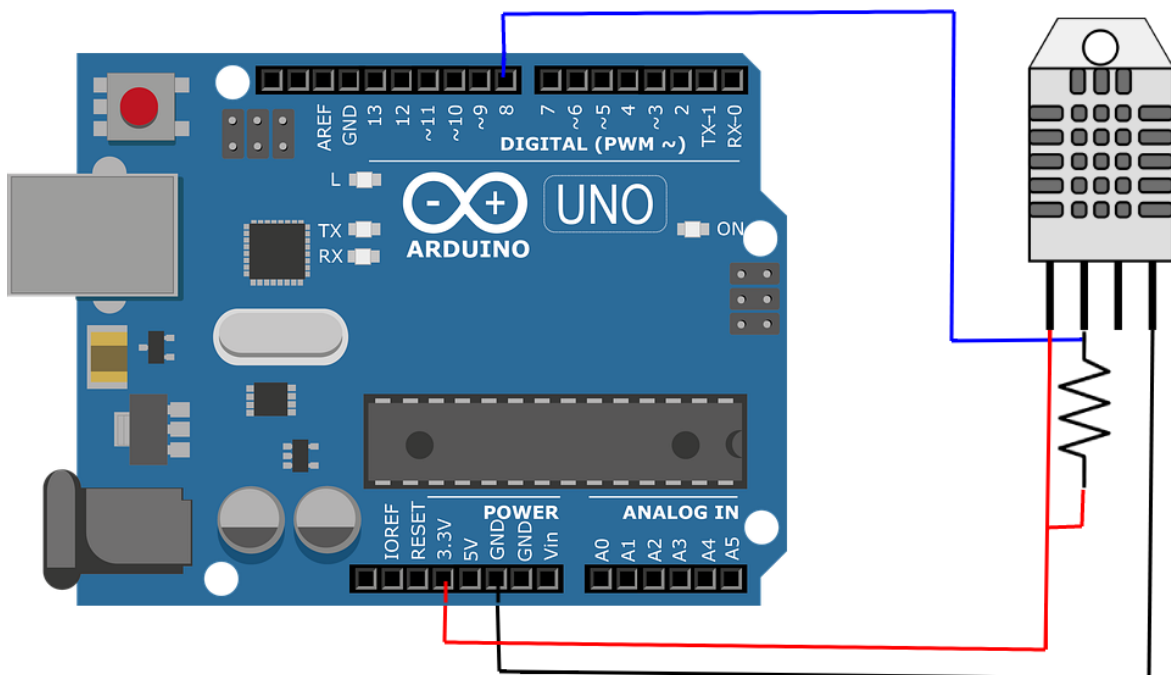
Slika 4.15. Ispis eCO₂ i TVOC vrijednosti na LCD 16x2 zaslonu 5 minuta nakon uključanja

4.5. Povezivanje senzora DHT22

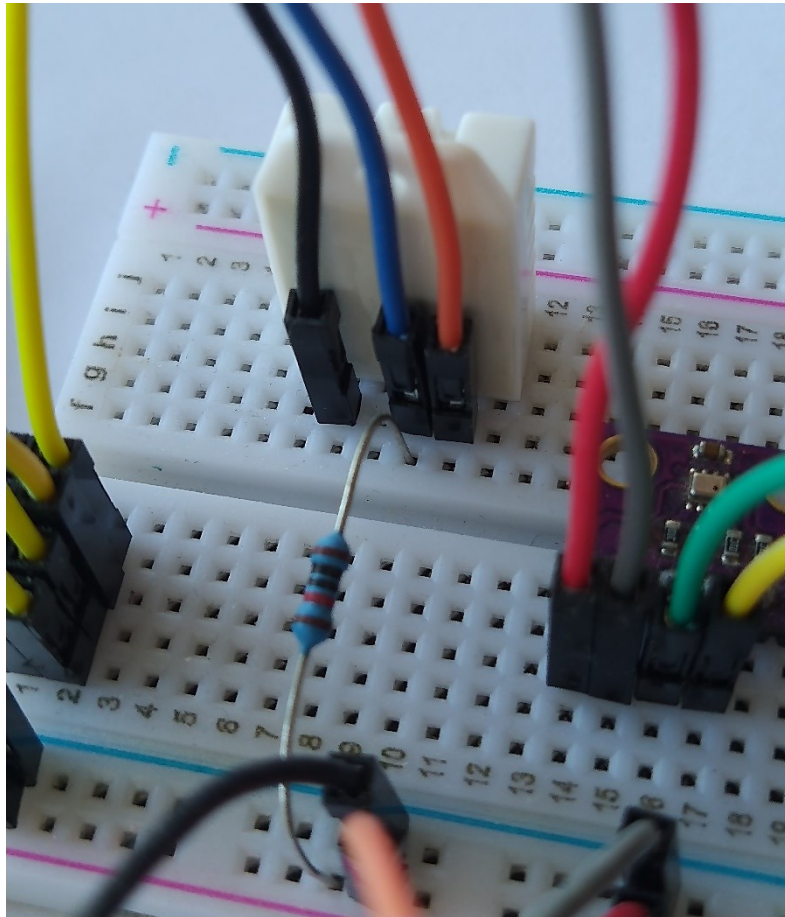
Naposljetku preostaje objasniti postupak povezivanja senzora DHT22 s Arduino pločicom. Slično kao i kod većine preostalih uređaja u sklopu meteorološke stanice, prvi korak podrazumijeva spajanje modula na eksperimentalnu pločicu, dok se u drugom koraku uređaj programira u Arduino razvojnom okruženju.

4.5.1. Spajanje modula DHT22 na eksperimentalnu pločicu

Modul DHT22 ima četiri pina što ga čini vrlo jednostavnim za spajanje na eksperimentalnu pločicu. Jedan pin nije spojen, dok se preostala tri spajaju prema shemi spajanja prikazanoj na slici 4.16. VCC pin spaja se na 3.3V ili 5V, GND na masu, a podatkovni pin spaja se na digitalni ulaz br. 8 Arduino Uno pločice preko „pull-up“ otpornika vrijednosti 10k Ω . Slika 4.17. prikazuje modul DHT22 spojen na eksperimentalnu pločicu.



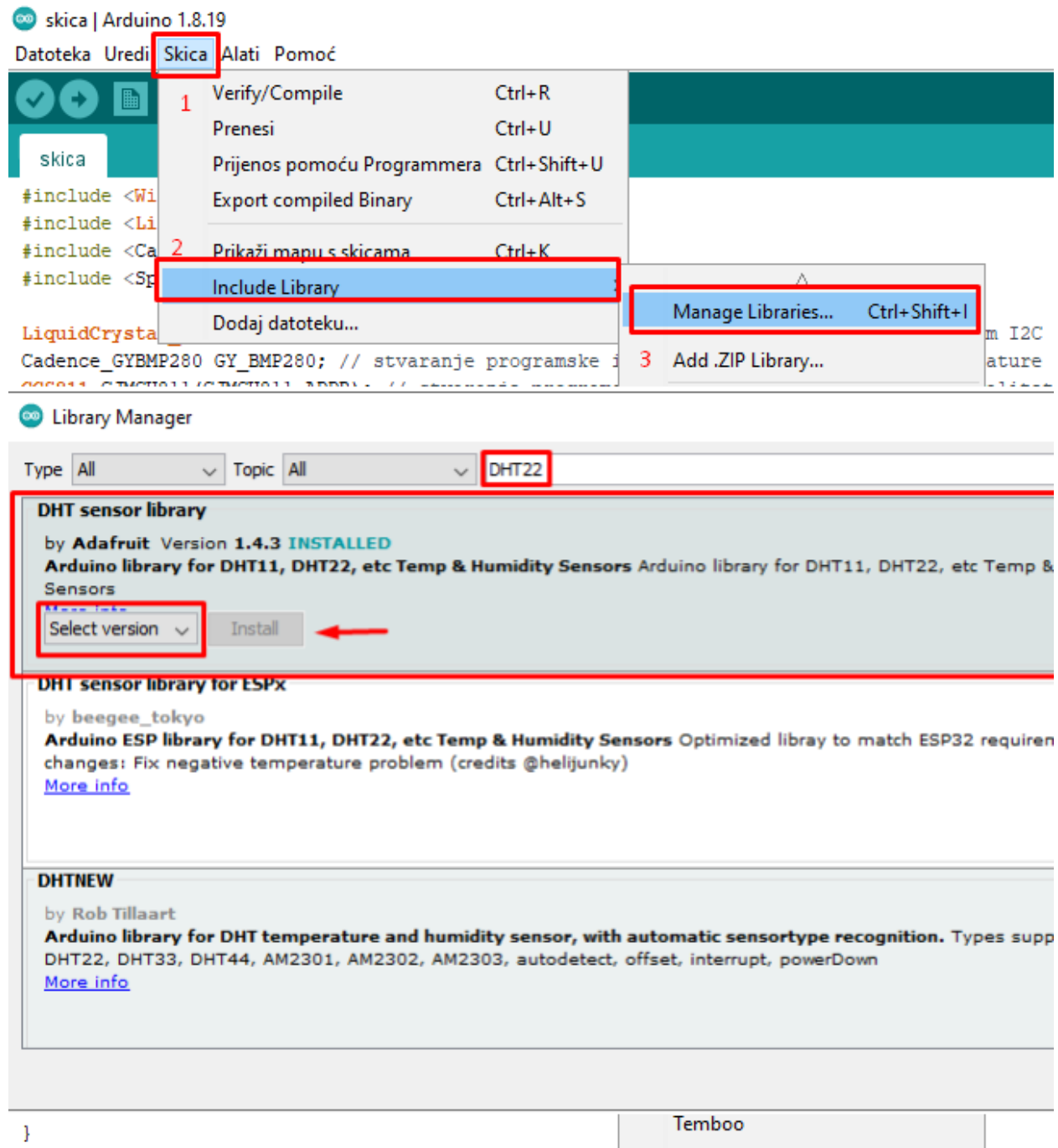
Slika 4.16. Shema izravnog spajanja modula DHT22 izrađena u aplikaciji Photopea



Slika 4.17. Prikaz modula DHT22 spojenog na eksperimentalnu pločicu

4.5.2. Učitavanje programske biblioteke i programiranje uređaja u Arduino razvojnom okruženju

Programska biblioteka senzora DHT22 dostupna za instalaciju nalazi se u upravitelju biblioteka. Postupak uključivanja biblioteke u Arduino skicu odgovara koracima prikazanim na slici 4.18.



Slika 4.18. Uključivanje Adafruit DHT biblioteke u Arduino skicu

Na vrhu skice, iznad svih funkcija i definicija, unose se sljedeće naredbe za učitavanje Adafruit DHT biblioteke:

```
#include <DHT.h>
#include <DHT_U.h>
```

Slijedi stvaranje programske instance senzora DHT22 pomoću naredbe:

```
DHT _DHT22(8, DHT22);
```

Prvi argument odnosi se na odabrani digitalni ulaz Arduino Uno pločice za uspostavu komunikacije. Drugi argument predstavlja tip DHT senzora. Naredba se unosi iznad svih funkcija, ali ispod svih definicija i naredbi za učitavanje programskih biblioteka.

Senzor je potrebno inicijalizirati unutar osnovne funkcije *setup* pomoću funkcije *begin*.

```
_DHT22.begin(); // pozivanje funkcije za inicijalizaciju DHT22 senzora
```

Budući da funkciju mjerenja temperature zraka u sklopu Arduino meteorološke stanice obavlja senzor BMP280, DHT22 će se koristiti za mjerenje isključivo relativne vlažnosti zraka. Funkcija *readHumidity* po izvršenju vraća izmjerenu postotnu vrijednost relativne vlažnosti zraka kao povratnu informaciju. Programski kod odgovoran za dohvat i ispis postotne vrijednosti relativne vlažnosti zraka na LCD 16x2 zaslonu je sljedeći:

```
// DHT22: relativna vlaznost zraka
LCD.setCursor(0, 0);
LCD.print("Rel. vlaznost");
LCD.setCursor(0, 1);
LCD.print("zraka: ");
LCD.print(_DHT22.readHumidity()); // funkcija "readHumidity()" kao povratnu
informaciju daje relativnu vlaznost zraka [%]
LCD.print("%");
delay(5000);
LCD.clear();
```

Na sljedećoj stranici naveden je cjelokupni programski kod koji je odgovoran za ispravan rad svih uređaja unutar meteorološke stanice temeljene na Arduino platformi. Komentari koji opisuju pojedinu naredbu su izostavljeni radi preglednosti.

```

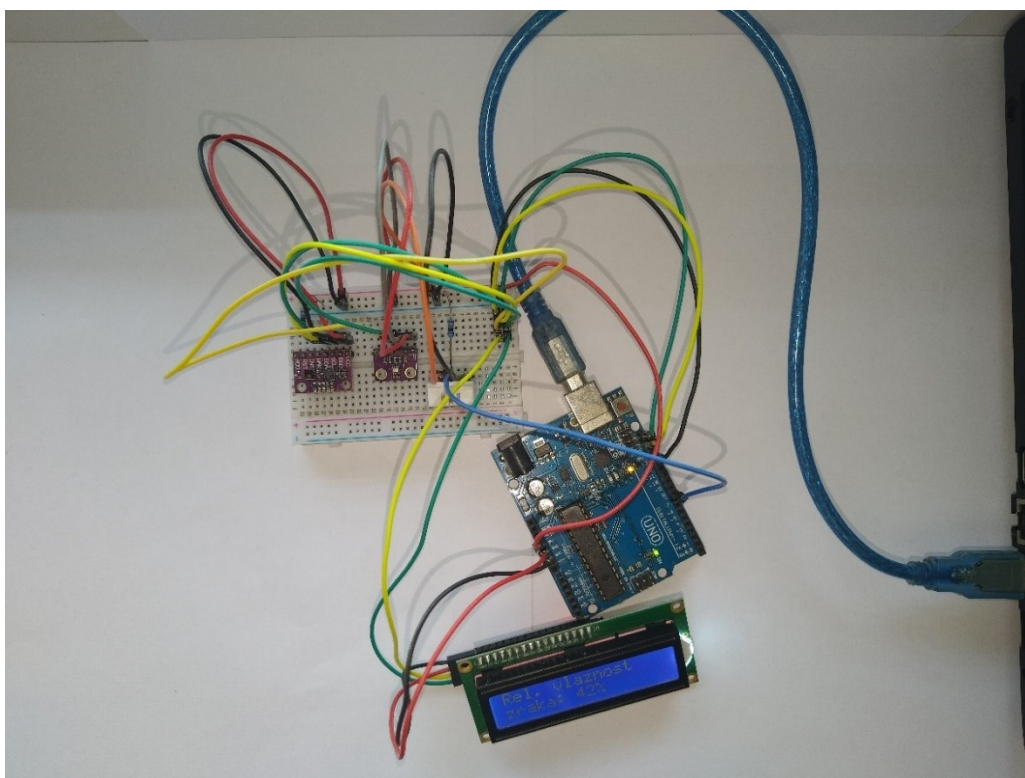
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Cadence_GYBMP280.h>
#include <DHT.h>
#include <DHT_U.h>
#include <SparkFunCCS811.h>
#define CJMCU811_ADDR 0x5A
LiquidCrystal_I2C LCD(0x27, 16, 2);
Cadence_GYBMP280 GY_BMP280;
DHT _DHT22(8, DHT22);
CCS811 CJMCU811(CJMCU811_ADDR);
void setup() {
  Wire.begin();
  Serial.begin(9600);
  LCD.init(); LCD.backlight();
  _DHT22.begin();
  if(!GY_BMP280.begin()) {
    Serial.println("[ERROR] Senzor BMP280 nije pronaden.");
    while(1);
  }
  if(!CJMCU811.begin()) {
    Serial.println("[ERROR] Senzor CCS811 nije pronaden.");
    while(1);
  }
  GY_BMP280.setDefaultQNH(1013.25);
  LCD.setCursor(0, 0);
  LCD.print("Arduino based");
  LCD.setCursor(0, 1);
  LCD.print("weather station");
  delay(5000);
  LCD.clear();
}
void loop() {
  LCD.setCursor(0, 0);
  LCD.print("Temp. zraka:");
  LCD.setCursor(0, 1);
  LCD.print(GY_BMP280.readTemperatureC());
  LCD.print(" C");
  delay(5000);
  LCD.clear();
  LCD.setCursor(0, 0);
  LCD.print("Tlak zraka:");
  LCD.setCursor(0, 1);
  LCD.print(GY_BMP280.readPressure());
  LCD.print(" Pa"); delay(5000);
  LCD.clear();
  LCD.setCursor(0, 0);
  LCD.print("Rel. vlaznost");
  LCD.setCursor(0, 1);
  LCD.print("zraka: ");
  LCD.print(_DHT22.readHumidity());
  LCD.print("%"); delay(5000);
  LCD.clear();
  for(int i = 0; i < 15; ++i) {
    if(CJMCU811.dataAvailable()) {
      CJMCU811.readAlgorithmResults();
    }
    LCD.setCursor(0, 0);
    LCD.print("eCO2: ");
    LCD.print(CJMCU811.getCO2());
    LCD.setCursor(0, 1);
    LCD.print("TVOC: ");
    LCD.print(CJMCU811.getTVOC());
    delay(2000);
    LCD.clear();
  }
  delay(100);
}

```

Na slici 4.19. prikazan je LCD 16x2 zaslon za vrijeme ispisa postotne relativne vlažnosti zraka, dok je konačan izgled rada prikazan na slici 4.20.



Slika 4.19. Ispis postotne relativne vlažnosti zraka na LCD 16x2 zaslonu



Slika 4.20. Konačan izgled meteorološke stanice temeljene na Arduino platformi

5. ZAKLJUČAK

Meteorološka stanica temeljena na Arduino platformi, izrađena u sklopu ovog završnog rada, pristupačno je i učinkovito rješenje za ispitivanje kvalitete zraka u manjim prostorima, kao primjerice u spavaćim i radnim sobama, uredima i sl. Uređaji za mjerenje u sklopu meteorološke stanice omogućuju izrazito točno i precizno mjerenje temperature zraka, atmosferskog tlaka i relativne vlažnosti uz nisku potrošnju električne energije. Ocjena kvalitete zraka u okolnom prostoru spada u jedan od zadataka meteorološke stanice. Senzor zadužen za mjerenje koncentracije štetnih čestica u zraku ima visok stupanj učinkovitosti, ali potreban mu je dulji vremenski period nakon uključanja kako bi davao relevantna očitavanja.

Jednostavnost izrade, mala potrošnja snage, raširenost i dostupnost odgovarajuće elektroničke opreme te relativno niska nabavna cijena iste, glavne su prednosti Arduino meteorološke stanice. Temeljni razlozi za odabir upravo Arduino pločice za obavljanje svih zadataka vezanih uz komunikaciju i obradu podataka jesu jednostavnost korištenja i izrazito razvijena programska podrška.

Glavni nedostaci su prilično velike dimenzije konačnog uređaja u usporedbi sa sličnim elektroničkim proizvodima te to što je za dohvat ispravnih očitavanja pojedinih senzora potrebno i do nekoliko minuta. Kod određenih uređaja javlja se i vremensko kašnjenje od nekoliko sekundi u očitavanju trenutnih vrijednosti mjerenih fizikalnih veličina.

6. LITERATURA

- [1] s interneta, <https://www.arduino.cc/en/Guide/Introduction>, 5.2.2018.
- [2] s interneta, <https://www.techopedia.com/definition/27874/arduino>
- [3] Mitchell R., „A Comparison of Popular Arduino Boards“, s interneta, <https://maker.pro/arduino/tutorial/a-comparison-of-popular-arduino-boards>, 25.10.2018.
- [4] s interneta, <https://www.maxpixel.net/static/photo/640/Digital-Arduino-Uno-Technology-Arduino-2168193.png>
- [5] Arduino, „Overview of the Arduino UNO components“, s interneta, <https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board>, 2022.
- [6] Mitchell R., „Common Communication Peripherals on the Arduino: UART, I2C and SPI“, s interneta, <https://maker.pro/arduino/tutorial/common-communication-peripherals-on-the-arduino-uart-i2c-and-spi>, 30.8.2018.
- [7] s interneta, <https://cdn.sparkfun.com/assets/2/5/c/4/5/50e1ce8bce395fb62b000000.png>
- [8] s interneta, <https://www.deviceplus.com/arduino/arduino-communication-protocols-tutorial/>, 14.10.2021.
- [9] He A., „Basic Electronics: Wired Communication Protocols in Embedded Design“, s interneta, <https://www.seeedstudio.com/blog/2019/07/03/basic-electronics-wired-communication-protocols-in-embedded-design/>, 2019.
- [10] Thangavel P., „How to use SPI (Serial Peripheral Interface) in Arduino to Communication between two Arduino boards“, s interneta, <https://circuitdigest.com/microcontroller-projects/arduino-spi-communication-tutorial>, 18.12.2018.
- [11] s interneta, https://en.wikipedia.org/wiki/Serial_Peripheral_Interface#/media/File:SPI_three_slaves.svg
- [12] Thangavel P., „How to use I2C in Arduino: Communication between two Arduino Boards“, s interneta, <https://circuitdigest.com/microcontroller-projects/arduino-i2c-tutorial-communication-between-two-arduino>, 24.12.2018.
- [13] „How to Identify the I2C Slave Addressing Standard for Developers“, s interneta, <https://www.totalphase.com/blog/2019/05/how-to-identify-the-i2c-slave-addressing-standard-for-developers/>, 21.5.2019.

- [14] s interneta, <https://www.maxpixel.net/static/photo/640/Digital-Arduino-Uno-Technology-Arduino-2168193.png>
- [15] s interneta, <https://electrosome.com/wp-content/uploads/2018/02/I2C-Interface.png>
- [16] Arduino, „Arduino Integrated Development Environment (IDE) v1“, s interneta, <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>, 2022.
- [17] s interneta, <https://coolcomponents.co.uk/blogs/news/what-language-is-an-arduino-programmed-in>, 18.11.2019.
- [18] s interneta, <https://www.arduino.cc/reference/en/language/structure/sketch/setup/>
- [19] s interneta, <https://www.arduino.cc/reference/en/language/structure/sketch/loop/>
- [20] s interneta, <https://www.arduino.cc/reference/en/libraries/>
- [21] Campbell S., „How to use Arduino libraries“, s interneta, <https://www.circuitbasics.com/how-to-use-arduino-libraries/>, 2021.
- [22] s interneta, <http://web.studenti.math.pmf.unizg.hr/~marrast/Senzori>
- [23] s interneta, <https://e-radionica.com/hr/blog/2015/08/19/eksperimentalna-plocica/>
- [24] s interneta, <https://hibp.ecse.rpi.edu/~connor/education/breadboard.pdf>
- [25] s interneta, https://media.springernature.com/original/springer-static/image/chp%3A10.1007%2F978-1-4842-3480-8_1/MediaObjects/453258_1_En_1_Fig8_HTML.jpg
- [26] s interneta, <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp280-ds001.pdf>, 2021.
- [27] s interneta, <https://components101.com/sensors/gy-bmp280-module>, 12.9.2018.
- [28] s interneta, <https://www.hotmcu.com/gybmp28033-high-precision-atmospheric-pressure-sensor-module-p-328.html>
- [29] s interneta, <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- [30] s interneta, <https://www.adafruit.com/product/385>
- [31] Issac, „DHT22 – precizni senzor temperature i vlažnosti“, s interneta, https://www.hwlibre.com/hr/dht22/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+hwlibreweb+%28Hardware+libre%29

- [32] Gajera U., „Basics of Interfacing DHT11/DHT22 Humidity and Temperature Sensor with MCU“, <http://www.ocfreaks.com/basics-interfacing-dht11-dht22-humidity-temperature-sensor-mcu/>, 6.11.2017.
- [33] s interneta, <https://cdn.pimylifeup.com/wp-content/uploads/2019/05/DHT22-Temperature-Sensor-with-numbering.png>
- [34] s interneta, <https://www.sciosense.com/wp-content/uploads/documents/SC-001232-DS-3-CCS811B-Datasheet-Revision-2.pdf>
- [35] s interneta, <https://e-radionica.com/hr/ccs811-senzor.html>
- [36] s interneta, <http://www.enciklopedija.hr/Natuknica.aspx?ID=66921>
- [37] s interneta, <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- [38] Raj. A, „16x2 LCD Module“, s interneta, <https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet>, 24.10.2015.
- [39] s interneta, <https://i.pinimg.com/originals/fd/99/7f/fd997f96f2f3015e5d661d4c26a20e85.png>
- [40] s interneta, https://e-radionica.com/media/catalog/product/cache/5/image/9df78eab33525d08d6e5fb8d27136e95/_/5/_506371-edit.jpg
- [41] s interneta, <https://www.arduino.cc/reference/en/language/functions/communication/wire/>

SAŽETAK

Završni rad obuhvaća primjenu teorijskih znanja stečenih tijekom školovanja za izradu uređaja koji ima sposobnost mjerenja fizikalnih veličina temperature zraka, atmosferskog tlaka i relativne vlažnosti zraka te veličina kojima je moguće utvrditi kvalitetu zraka u okolnom prostoru (eCO₂ i TVOC). Meteorološku stanicu čini mnoštvo elektroničkih uređaja povezanih u cjelinu. Teorijska podloga neophodna za izradu završnog rada dana je u poglavljima od 1 do 4.

Središnji uređaj meteorološke stanice je Arduino Uno R3 elektronička razvojna pločica kada je riječ o sklopovlju, odnosno Arduino programsko razvojno okruženje kada se radi o programskoj podršci. Arduino Uno pločica fizički povezuje preostale elektroničke uređaje pomoću vodiča i djeluje kao središnja točka za razmjenu podataka između uređaja. Rad iznosi detaljan opis standardnih serijskih komunikacijskih protokola koje Arduino podržava: UART, SPI i I²C.

Senzor BMP280 u sklopu modula GY-BMP280, mjerno je osjetilo temperature zraka i atmosferskog tlaka visoke preciznosti i male potrošnje električne energije. DHT22 također ima sposobnost mjerenja temperature zraka, ali mjeri i postotnu relativnu vlažnost zraka. Odlikuju ga pristupačna cijena i mala odstupanja pri mjerenju. Kvaliteta zraka ispituje se pomoću senzora CCS811 ugrađenog na modul CJMCU-811. Osjetilo ima visoku osjetljivost i širok raspon prepoznavanja hlapivih organskih spojeva (TVOC) u zraku. Vrijednosti izmjerenih fizikalnih veličina prikazuju se na LCD 16x2 zaslonu. Gotovo svi elektronički uređaji unutar meteorološke stanice komuniciraju putem I2C komunikacijskog protokola. Izuzetak je senzor DHT22 koji komunicira s Arduinom putem zasebnog podatkovnog vodiča.

Ključne riječi: Arduino Uno, I²C, LCD, BMP280, DHT22, CCS811

SUMMARY

This bachelors thesis includes the application of theoretical knowledge acquired throughout education in order to create an electronic device capable of measuring physical quantities such as air temperature, atmospheric pressure, and relative humidity, as well as quantities that can be used to determine the air quality in the surrounding area (eCO₂ and TVOC). The Arduino based weather station consists of a multitude of electronic devices. Chapters 1 to 4 contain the theoretical framework of the thesis.

The weather station's central device is the Arduino Uno R3 electronic board when referring to hardware, or Arduino integrated development environment (IDE) when it comes to software. The Arduino Uno board physically connects the rest of the electronic devices using jumper wires and acts as a pivotal point for data exchange between the devices. The thesis contains a detailed description of the standard Arduino serial communication protocols: UART, SPI and I²C.

The BMP280 sensor as part of the GY-BMP280 module measures air temperature and atmospheric pressure with high precision and low power consumption. DHT22 also has the ability to measure air temperature, as well as relative humidity. It is characterised by an affordable price and small errors during measurement. Air quality is assessed using the CCS811 sensor installed on the CJMCU-811 module. The sensor has high sensitivity and a wide range of detection of volatile organic compounds (VOC) in the air. The values of the measured physical quantities are displayed on the LCD 16x2 screen. Almost all electronic devices inside the weather station communicate via the I²C communication protocol except the DHT22 sensor, which communicates with the Arduino via a separate data line.

Keywords: Arduino Uno, I²C, LCD, BMP280, DHT22, CCS811