



Co-funded by the
Erasmus+ Programme
of the European Union



WirelessUP!

UPraising VET skills for innovation in European electrotechnical sector

Broj projekta: 2017-1-HR01-KA202-035434

WirelessUP! Toolkit

Modul 3.1: Uvođenje bežičnih tehnologija u sustave upravljanja

Arduino MKR 1000 projekt

Intelektualni rezultat 3

Veljača 2019.

Ova publikacija odražava isključivo stajalište autora publikacije i Komisija se ne može smatrati odgovornom prilikom uporabe informacija koje se u njoj nalaze.



Sadržaj

1. Uvod u Arduino MKR 1000	3
2. IOT-ova vremenska stanica – prvi dio	5
2.1. Montaža vremenske stanice – temperatura, vlaga, tlak	5
2.1.1. Priključivanje 1. korak (Slika 4):.....	5
2.1.2. Priključivanje 2. korak (Slika 5):.....	6
2.1.3. Priključivanje 3. korak (Slika 6):.....	7
2.1.4. Priključivanje 4. korak (Slika 7):.....	7
3. IOT-ova vremenska stanica – drugi dio	8
3.1. Montaža vremenske stanice – mikro čestice u zraku.....	8
3.1.1. Priključivanje 1. korak (Slika 9):.....	8
3.1.2. Priključivanje 2. korak (Slika 10):.....	9
4. Platforma Blynk.....	10
4.1. Preuzimanje aplikacije.....	10
4.2. Kreiranje projekta.....	11
4.3. Blynk kućica za uređaj (widget box)	14
4.3.1. Projekt 1 – IOT-ova vremenska stanica (prvi dio).....	14
4.3.2. Projekt 2 – IOT-ova vremenska stanica (drugi dio)	15
5. Arduino IDE program	15
5.1. Preuzmi Arduino IDE program (software)	15
5.2. Objasnjenje programa Arduino IDE.....	17
5.3. Učitavanje Arduinovog programa u mjerne uređaje i ispitivanje	18
5.3.1. Učitavanje programa u naše projekte Projekt 1 – mjerjenje TTVSU measurement.....	18
5.3.2. Učitavanje programa u projekt 2 – mjerjenje PMSmeasurement.....	22
5.3.3. Spajanje (merging) programskih kodova	23



1. Uvod u Arduino MKR 1000

MKR 1000 je jedan od mnogih razvojnih alata iz Arduinove obitelji proizvoda. Ploča obuhvaća mikro upravljač ili kontroler, modul za bežično povezivanje, LiPo bateriju za napajanje, punjenje USB-om uz pomoć USB kabela, te druge sklopovne elemente potrebne za rad mikro upravljača. Ploča MKR1000 je prikazana na slici.



Slika 1. Ploča MKR 100

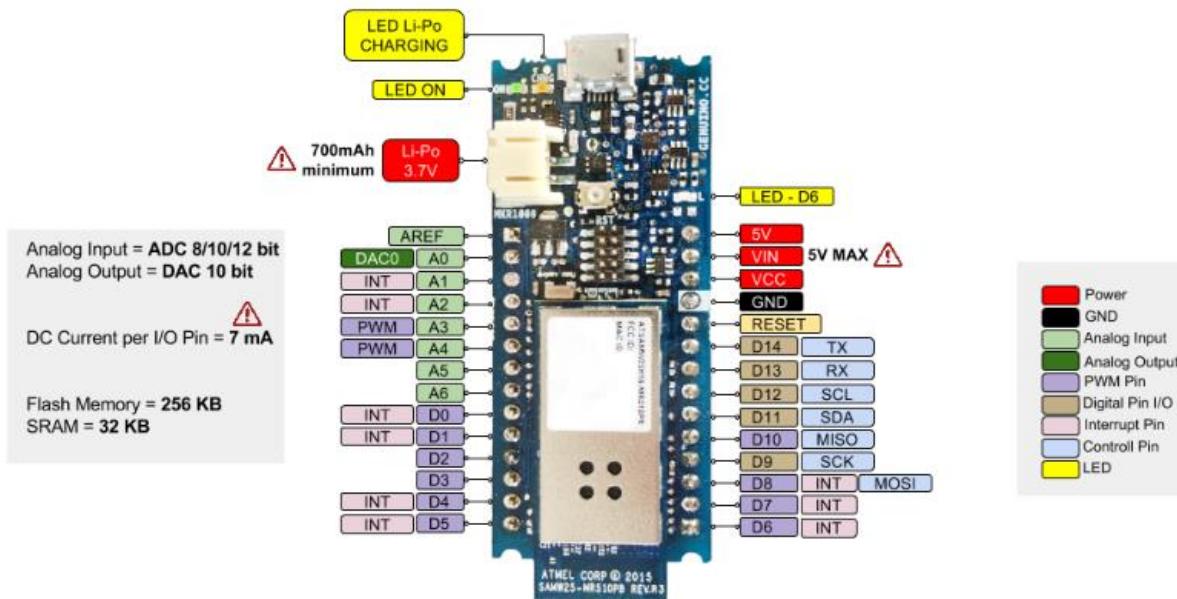
Mikro upravljač u biti možemo smatrati malim računalom. Samo se njegova svrha razlikuje od svrhe tipičnog stolnog ili prijenosnog računala. Uz pomoć programa upisanih u razvojno sučelje Arduino IDE možemo očitati status različitih davača i upravljati različitim uređajima, a oni će potom raditi na samom mikro upravljaču.

Mikro upravljač možeš zamisliti kao crnu kutiju s desetak nastavaka. Funkcije mikro upravljača se koriste za povezivanje nekih električkih elemenata s mikro upravljačem i upravljanje mikro upravljačem uz pomoć programa Arduino IDE.

Kako bi mikro upravljač znao što treba napraviti i kako upravljati elementima koje smo mu priključili, prvo moramo napisati program, a zatim ga moramo unijeti u mikro upravljač.

Ako želimo dodati još elemenata na Arduinove pločice, pokrenut ćemo mikro upravljače do kojih možemo doći preko krajnjih spojnika (konektora) (crni spojnici na rubu Arduinove ploče). Pregled izlaza mikro upravljača je dan na slici 2.

MKR1000 PINOUT



Slika 2. Nožica MKR 1000 (pinout)

Kao što vidiš, svaki nastavak (pin) ima svoju jedinstvenu oznaku ili tag. Te će se označe ili tagovi koristiti i u programima koje pišemo kako bismo upravljali Arduinovim nastavcima (pinovima). Arduinova ploča je označena slovima i brojkama. Postoje dva tipa nastavaka (pinova), digitalni izlazi (D0) i analogni izlazi (A0).

Analogni su izlazi označeni slovom A i brojkama - A0, A1, A2 sve do A6. Ti izlazi mogu mjeriti analogni napon od 0-3.3V.

Digitalni su izlazi označeni isključivo brojkama - 0, 1, 2 sve do 14, ili slovom D i brojkom. Mogu imati samo dva stanja - 0 ili 1 (NIZAK ili VISOK), ili UKLJUČEN ili ISKLJUČEN.

Analogni izlazi se mogu koristiti kao i digitalni, ali digitalni izlazi ne mogu biti analogni. Svi nastavci (pinovi) na ploči mogu raditi na dva načina – u ulaznom i u izlaznom načinu rada.

Izlazni način rada se koristi kada mikro upravljačem ili mikro kontrolerom želimo upravljati nekim komponentama koje su priključene na taj izlaz (LED lampicama, motorima, relejima itd.). Ulazni način rada se koristi kada želimo očitati stanje pojedinog izlaza ili kada priključujemo jednostavne davače ili senzore kao što su sklopke, termostati i tako dalje.

Krenimo s radom i praktičnim primjerima jer ćemo tako najbolje shvatiti na koji način radi Arduino.



2. IOT-ova vremenska stanica – prvi dio

Ovaj se projekt sastoji od davača (senzora) za jačinu svjetla, temperature, vlage, tlaka i mikro čestica u zraku (mjerjenje kvalitete zraka) na platformi MKR1000. Povrh toga, obuhvaća i kreiranje aplikacije na mobitelu za vizualizaciju izmjerениh vrijednosti. Vremensku ćemo stanicu sastaviti u nekoliko koraka.

1. Montaža uređaja za mjerjenje jačine svjetla, temperature, tlaka i vlage.
2. Montaža uređaja za mjerjenje mikro čestica u zraku (kvalitete zraka).
3. Dobivanje tokena za priključivanje na Blynk poslužitelj (server) i kreiranje aplikacije za mobitel.
4. Učitavanje programa Arduino u mjerne uređaje i ispitivanje priključenih uređaja.
5. Povratna informacija o tokenima.

2.1. Montaža vremenske stanice – temperatura, vlaga, tlak

Prvi dio IOT-ove vremenske stanice sastoji se od tri davača (senzora) za mjerjenje svjetla (davači APDS), vlage (davač DHT 11) i tlaka zraka (BMP 180) – slika 3.

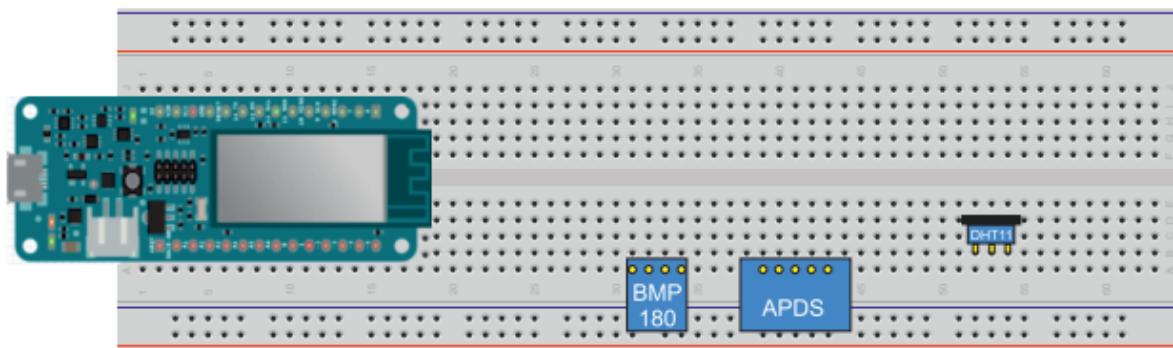
Shema na slici 4 prikazuje navedene davače montirane na probnu ploču. Žuto obojeni kontakti označavaju kontakte pojedinačnih davača koje ćemo priključiti na Arduinovu platformu MKR1000.



Slika 3. Davači za vremensku stanicu

2.1.1. Priključivanje 1. korak (Slika 4):

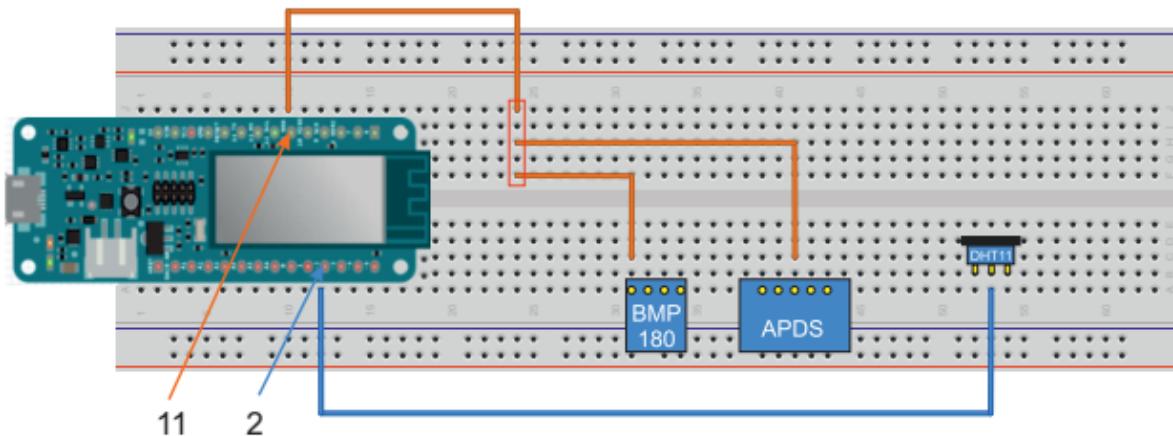
1. Umetni Arduinovu platformu MKR1000 u ispitnu ploču.
2. Umetni davač BMP180 u ispitnu ploču.
3. Umetni davač APDS u ispitnu ploču.
4. Umetni davač DHT11 u ispitnu ploču.



Slika 4. Priključivanje 1. korak

2.1.2. Priključivanje 2. korak (Slika 5):

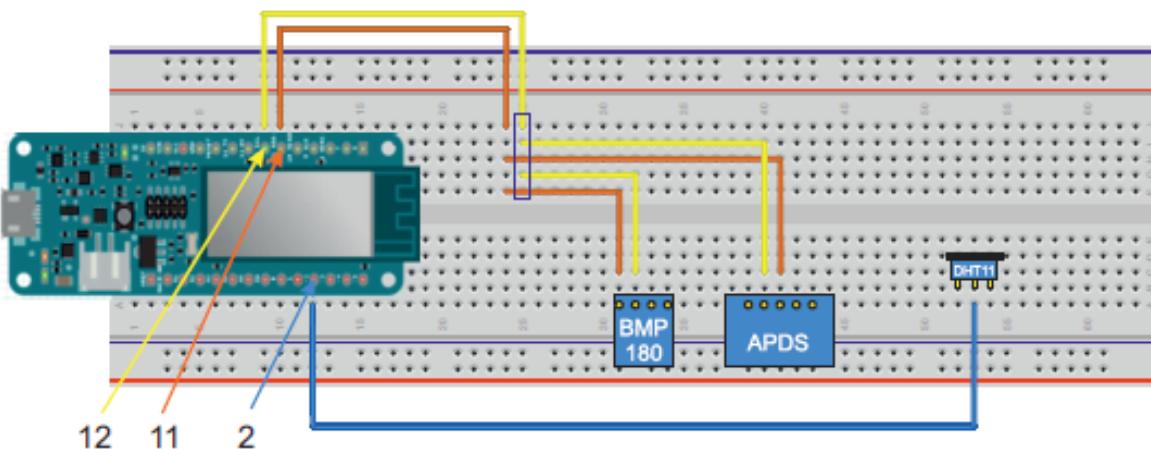
1. Priključi srednji kontakt davača DHT 11 na broj 2 na Arduinovoj ploči (plava žica na slici).
2. Priključi srednji kontakt davača APDS na jedan stupac ispitne ploče (označen narančastim pravokutnikom).
3. Priključi prvi (lijevi) kontakt davača BMP180 na isti stupac ispitne ploče (označen narančastim pravokutnikom).
4. Priključi taj stupac (označen narančastim pravokutnikom) dodatnom žicom na Arduinov nastavak digitalnog ulaza broj 11.



Slika 5. Priključivanje 2. korak – priključivanje davača na Arduinovu ploču MKR

2.1.3. Priključivanje 3. korak (Slika 6):

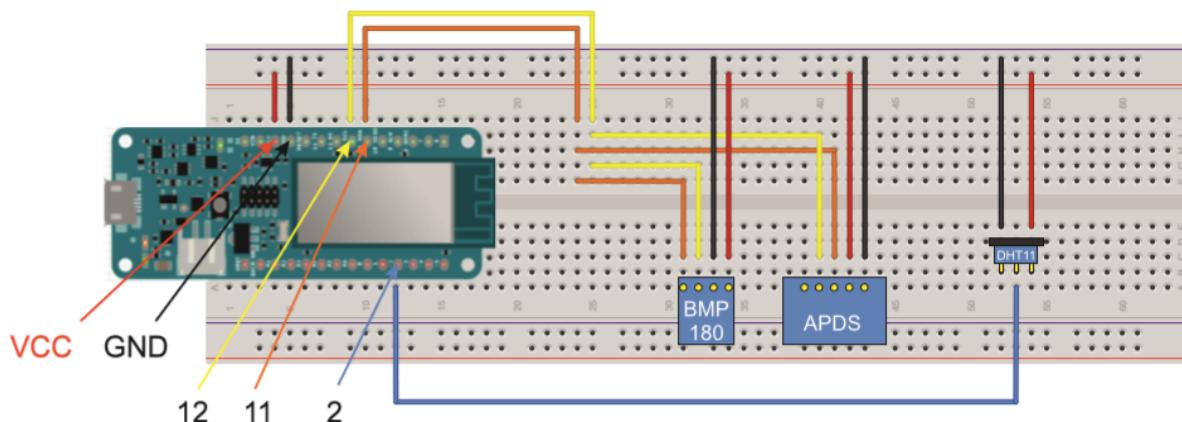
1. Priključi drugi kontakt davača APDS (žuta žica) na stupac ispitne ploče (označen plavim pravokutnikom).
2. Priključi drugi kontakt davača BMP180 (žuta žica) na stupac ispitne ploče (označen plavim pravokutnikom).
3. Priključi stupac (označen plavim pravokutnikom) dodatnom žicom na Arduinov nastavak digitalnog ulaza broj 12.



Slika 6. Priključivanje 3. korak

2.1.4. Priključivanje 4. korak (Slika 7):

1. Priključi Arduinov nastavak „VCC“ (5V) u zajednički (+) redak na ispitnoj ploči (crvena žica na slici 8).
2. Priključi Arduinov nastavak „GND“ u zajednički (-) redak na ispitnoj ploči (crna žica na slici 8).
3. Priključi treći nastavak davača BMP180 u zajednički (-) redak na ispitnoj ploči (crna žica na slici 8).
4. Priključi četvrti nastavak davača BMP180 u zajednički (+) redak na ispitnoj ploči (crvena žica na slici 8).
5. Priključi četvrti nastavak davača APDS u zajednički (+) redak na ispitnoj ploči (crvena žica na slici 8).
6. Priključi peti nastavak davača APDS u zajednički (-) redak na ispitnoj ploči (crna žica na slici 8).



Slika 7. Priključivanje 4. korak

3. IOT-ova vremenska stanica – drugi dio

Ovaj se projekt sastoji od Plantowerovog davača za mjerjenje mikro čestica u zraku (kvalitete zraka) na platformi MKR1000 i kreiranja aplikacije za mobitel za vizualizaciju vrijednosti davača. Vremensku čemo stanicu sastaviti u nekoliko koraka.

3.1. Montaža vremenske stanice – mikro čestice u zraku

Drugi dio IOT-ove vremenske stanice sastoji se od davača za mjerjenje mikro čestica u zraku (kvalitete zraka) - Slika 8.

Shema na slici 10 prikazuje navedeni davač priključen na ispitnu ploču.



Slika 8. Plantowerov davač za mjerjenje mikro čestica u zraku

3.1.1. Priklučivanje 1. korak (Slika 9):

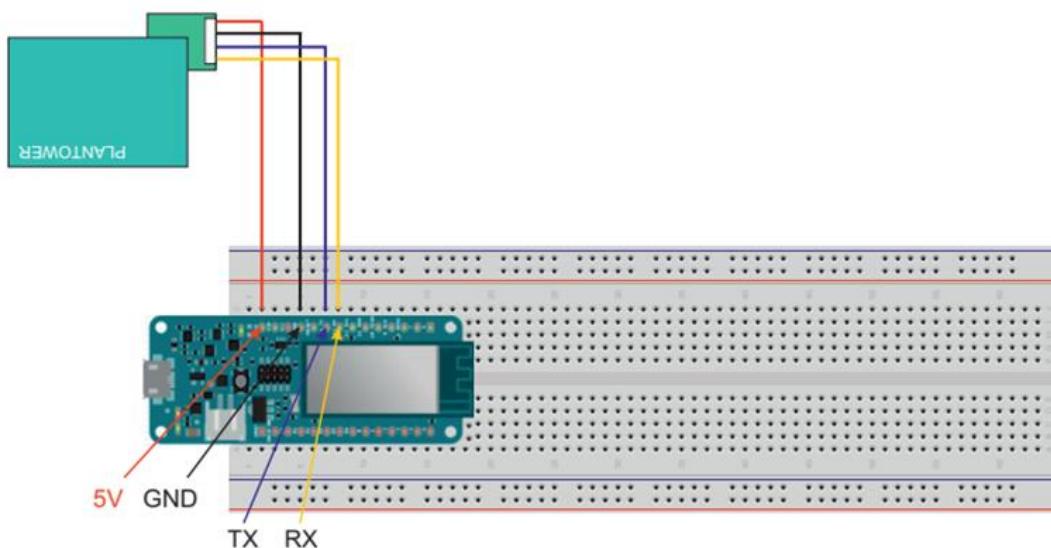
Utakni bijeli priključak u tiskanu pločicu sa strujnim krugovima i montiraj je na Plantowerov davač za mjerjenje mikro čestica u zraku. Pazi da postaviš tiskanu pločicu sa strujnim krugovima tako da sjedne u Plantowerov davač.



Slika 9. Priključivanje tiskane pločice sa strujnim krugovima na Plantowerov davač

3.1.2. Priključivanje 2. korak (Slika 10):

1. Priključi crvenu žicu Plantowerovog davača na nastavak „VCC“ (5V) Arduinove ploče MKR 1000.
1. Priključi crnu žicu Plantowerovog davača na nastavak „GND“ Arduinove ploče MKR 1000
2. Priključi plavu žicu Plantowerovog davača na digitalni nastavak 14 Arduinove ploče MKR 1000.
4. Priključi žutu žicu Plantowerovog davača na digitalni nastavak 13 Arduinove ploče MKR 1000.



Slika 10. Priključivanje Plantowerog davača na Arduinovu ploču MKR 1000

Napomena:

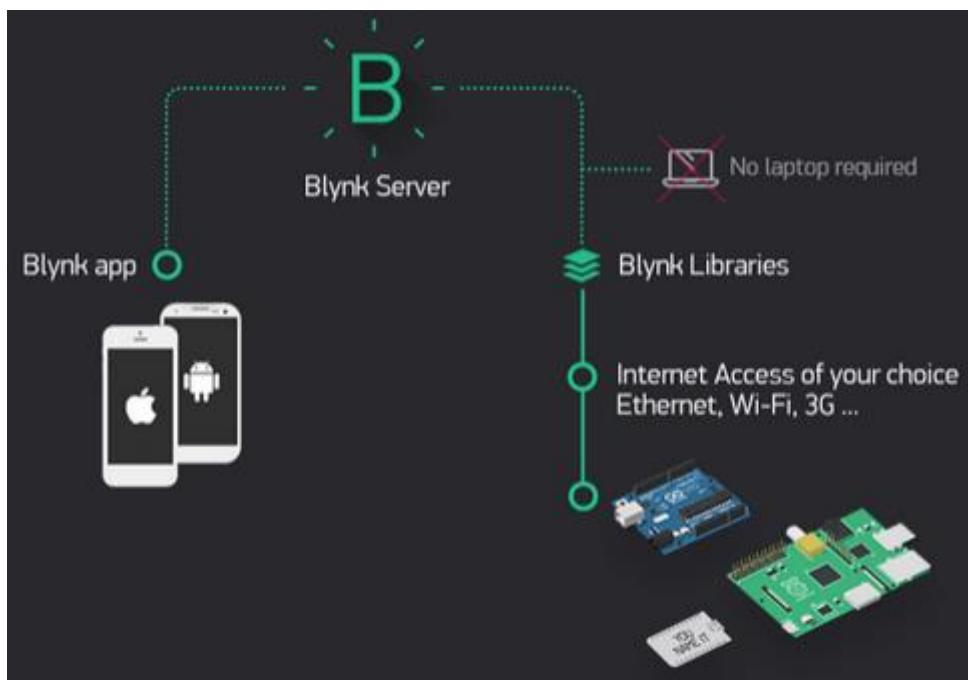
Plantowerov davač se mora proširiti muškim-muškim žicama prije priključivanja na Arduinovu ploču MKR 1000.

4. Platforma Blynk

Platforma Blynk je platforma otvorenog izvora Interneta stvari namijenjena pojednostavljinju izgradnje mrežnih aplikacija i aplikacija za mobitele za Internet stvari. Njom se može daljinski upravljati sklopoljem, prikazivati podatke davača, pohranjivati podatke, vizualizirati ih i činiti mnogo drugih stvari. Platforma Blynk se može lako priključiti na sklopnove modele sve do 400+ poput Arduina, ESP8266, ESP32, Raspberry Pi i sličnih mikro upravljačkih jedinica (MCU) i mobilnih IoT-ovih aplikacija koje se povuku i ispuste (drag-n-drop) za iOS i Android.

Na platformi se nalaze tri glavne komponente:

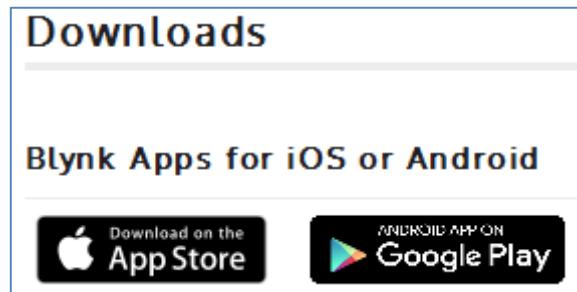
1. **Blynk aplikacija (Blynk App)** – omogućava kreiranje izvrsnog sučelja za projekte u kojima se koriste različite uređaje (widgeti).
2. **Blynk poslužitelj (Server)** – koji je odgovoran za cijelokupnu komunikaciju između pametnog telefona i sklopolja. Omogućava i rad lokalnog privatnog Blynk poslužitelja (servera) za projekte. Blynk poslužitelj može lako upravljati tisućama uređaja, a može se ugraditi i u Raspberry Pi.
3. **Blynk knjižnice (Libraries)** – omogućavaju komunikaciju s poslužiteljem (serverom) i obrađuju sve ulazne i izlazne komande.



Slika 11. Platforma Blynk

4.1. Preuzimanje aplikacije

Prije nego što počneš s prvim projektom, moraš preuzeti i instalirati aplikaciju Blynk na pametni telefon ili tablet. Aplikacija se preuzima na Google Store-u (Slika 12.) ako radiš u operativnom sustavu Android ili na App Storeu (Slika 12) ako radiš u operativnom sustavu iOS. Možeš je naći po ključnoj riječi „Blynk“.



Slika 12. Operativni sustav Blynk za pametne telefone

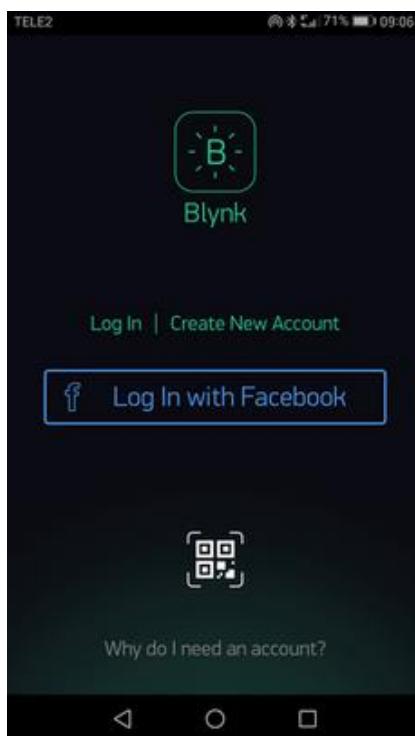
Izravne poveznice za preuzimanje su:

Android: <https://play.google.com/store/apps/details?id=cc.blynk>

IOS: <https://itunes.apple.com/us/app/blynk-control-arduino-raspberry/id808760481?ls=1&mt=8>

4.2. Kreiranje projekta

Kad završiš preuzimanje, otvori aplikaciju i registriraj se klikom na „kreiraj novi račun“ („Create New Account“) (Slika 13.). Moraš se registrirati kako bi pripremao (pripremala) projekte izravno na poslužitelju Blynk.

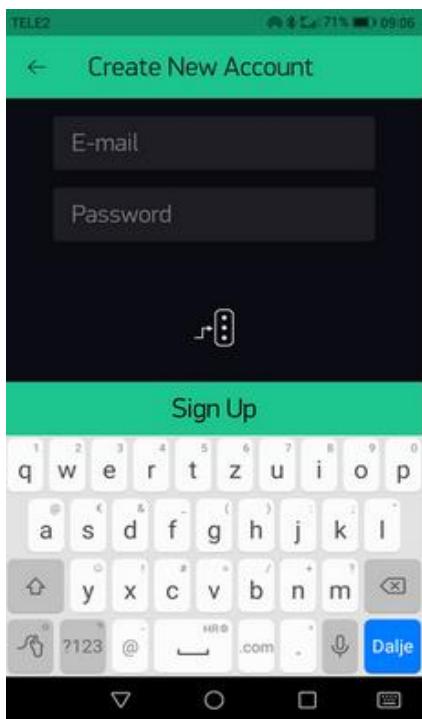


Slika 13. Aplikacija Blynk – kreiranje računa

Unesi adresu elektronske pošte (e-mail) i zaporku (password) da se registriraš.

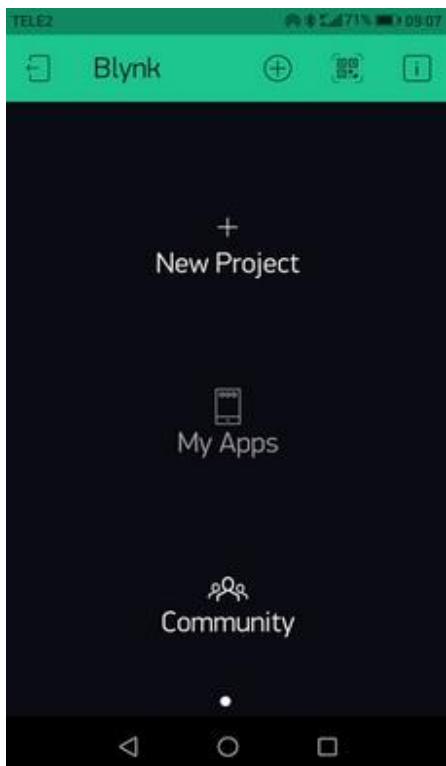


Co-funded by the
Erasmus+ Programme
of the European Union



Slika 14. Aplikacija Blink – adresa elektronske pošte i zaporka

Nakon registracije ćemo kreirati novi projekt odabirući opciju „Novi projekt“ („New Project“) (Slika 15.).



Slika 15. Aplikacija Blink – kreiranje novog projekta

Unesи осnovне podatke o projektu – odaberis ime projekta, skopovlje koje koristiš (Arduino MKR1000), tip priključivanja (WiFi) i boju sučelja – Svjetla (Slika 16.).

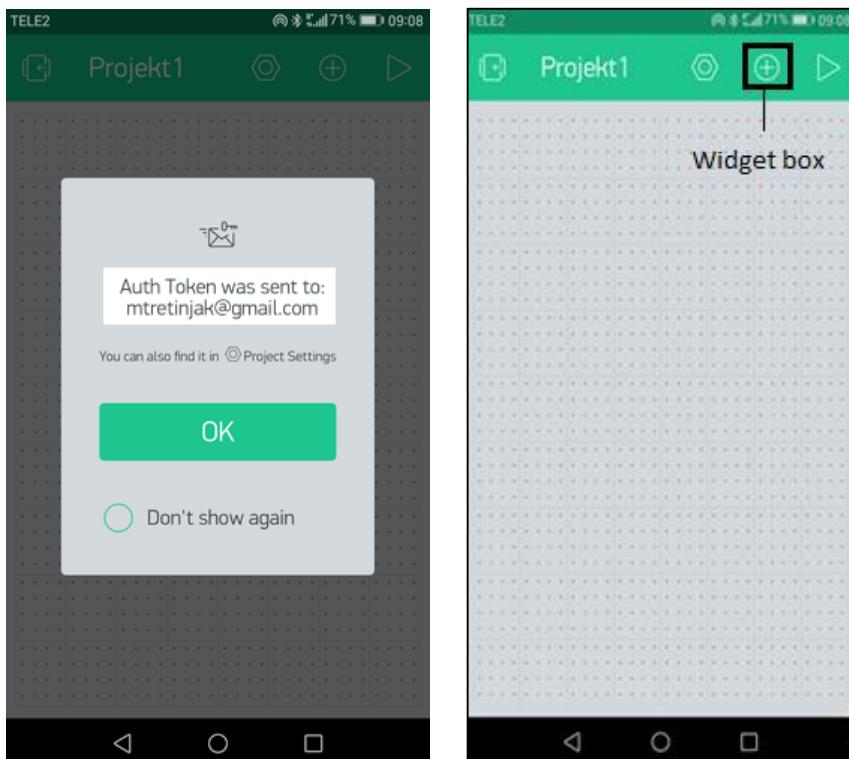


Co-funded by the
Erasmus+ Programme
of the European Union



Slika 16. Aplikacija Blynk – mogućnosti projekta

Nakon kreiranja projekta će ti se prikazati poruka koja će ti reći da si kreirao (kreirala) token za provjeru identiteta koji ti je poslan na tvoju adresu elektronske pošte (e-mail). Ovaj je token jedinstven za svaki projekt i koristi se za priključivanje sklopoljia na aplikaciju Blynk. Trebat će ti kasnije za Arduinovu programsku šifru.



Slika 17. Aplikacija Blynk – token za provjeru identiteta i kućica za uređaj (widget)



Zapiši svoj token za provjeru identiteta s elektronske poruke (e-maila) za projekt 1, a zatim i za projekt 2.

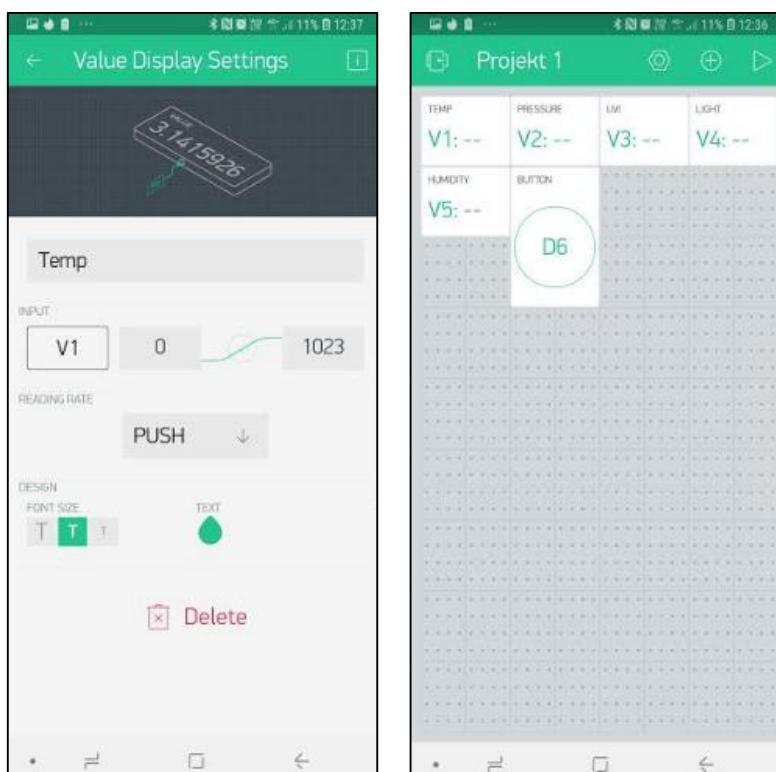
4.3. Blynk kućica za uređaj (widget box)

Dodaj uređaje (widgets) koje su ti potrebni za projekt 1 i projekt 2 u prozor aplikacije Blynk na slijedeći način.

4.3.1. Projekt 1 – IOT-ova vremenska stanica (prvi dio)

- Dodaj modul uređaja za prikazivanje vrijednosti i modul uređaja gumba projektu 1. Pod ime vrijednosti utipkaj tekst mjerne vrijednosti i postavi nastavak (pin) da pokazuje vrijednost kako ti je prikazano u slijedećem koraku.
- Ponovi 4. korak nekoliko puta da dobiješ ukupno 5 prikaza vrijednosti koji će prikazivati izmjerene vrijednosti prema popisu:
 - ➔ Prikaz vrijednosti - Temp - PIN: Virtual V1
 - ➔ Prikaz vrijednosti – Tlak (Pressure) - PIN: Virtual V2
 - ➔ Prikaz vrijednosti - UVI - PIN: Virtual V3
 - ➔ Prikaz vrijednosti – Svjetlo (Light) - PIN: Virtual V4
 - ➔ Prikaz vrijednosti – Vлага (Humidity) - PIN: Virtual V5

Dodaj modul gumba projektu koji će uključivati i isključivati LED lampicu Arduinovog digitalnog nastavka (pin) - D6. Postavi ga kao sklopku (SWITCH).



Slika 18. Aplikacija Blynk – dodavanje uređaja i postavki



4.3.2. Projekt 2 – IOT-ova vremenska stanica (drugi dio)

Dodaj drugi projekt (projekt 2) klikom na ikonu „+“ na vrhu alatne trake i ne zaboravi zapisati token za provjeru identiteta drugog projekta. Pod ime projekta unesi „projekt2“ bez razmaka. Drugi su podaci isti kao i za projekt1.

- Dodaj module prikaza vrijednosti uređaja projektu 2, isto kao i projektu 1.

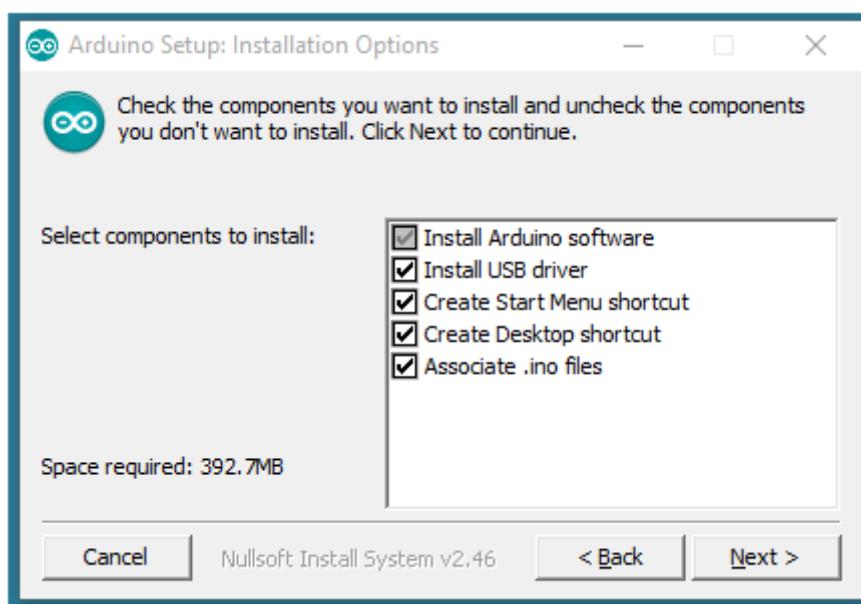
- ➔ Prikaz vrijednosti - PM 1 - PIN: Virtual V10
- ➔ Prikaz vrijednosti - PM 2.5 - PIN: Virtual V11
- ➔ Prikaz vrijednosti - PM 10 - PIN: Virtual V12

5. Arduino IDE program

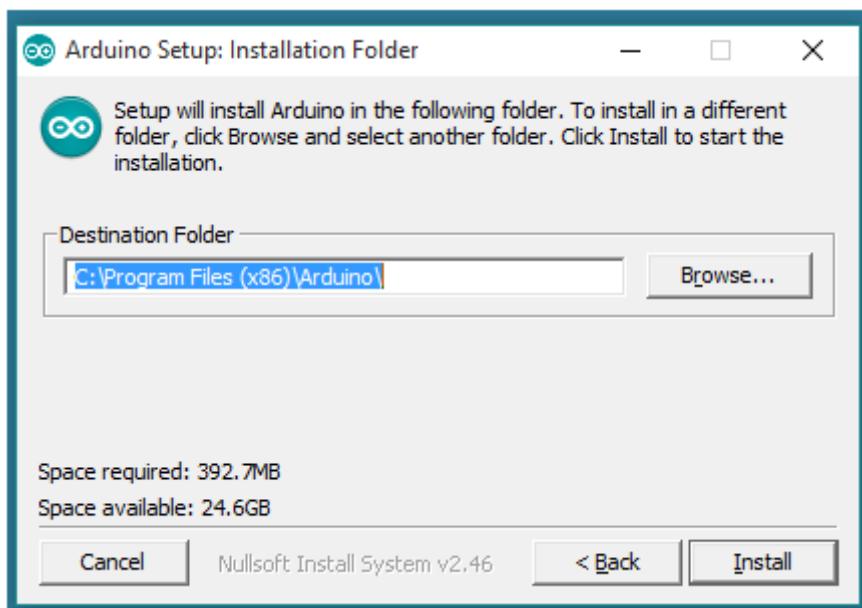
Arduino program otvorenog izvora (IDE) olakšava pisanje koda i njegovo upisivanje na ploču. Radi s Windows-ima, Mac OS X-om i Linux-om. Okružje je napisano u Javi, a temelji se na procesnim programima i drugim programima otvorenih izvora. Ovaj se program može koristiti s bilo kojom Arduinovom pločom.

5.1. Preuzmi Arduino IDE program (software)

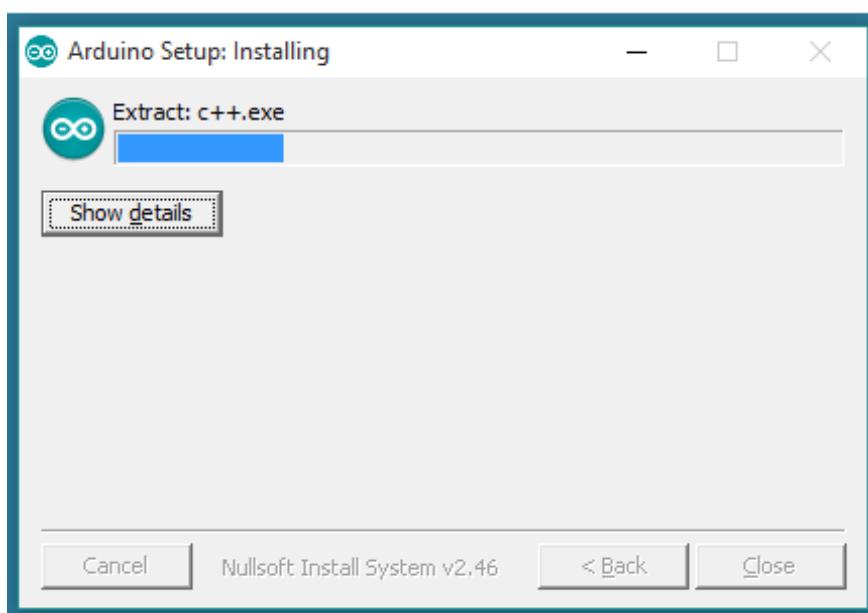
Preuzmi najnoviju verziju Arduino IDE programa sa stranice za preuzimanje. Možeš odabrati između paketa Installer (.exe) i Zip. Bolje je koristiti prvi zato što on instalira izravno sve što treba koristiti u Arduinovom programu (IDE), uključujući i pokretače (drive). Ako koristiš paket Zip, pokretače (drive) ćeš morati instalirati ručno. Zip datoteka je isto tako korisna ako želiš kreirati prijenosnu instalaciju. Kada završiš preuzimanje, nastavi s instalacijom i dopusti da se instaliraju pokretači (driveri) kada te operativni sustav na to upozori.



Odaberite komponente za instalaciju.



Odaberite instalacijski direktorij (zadrži uobičajeni (default)).

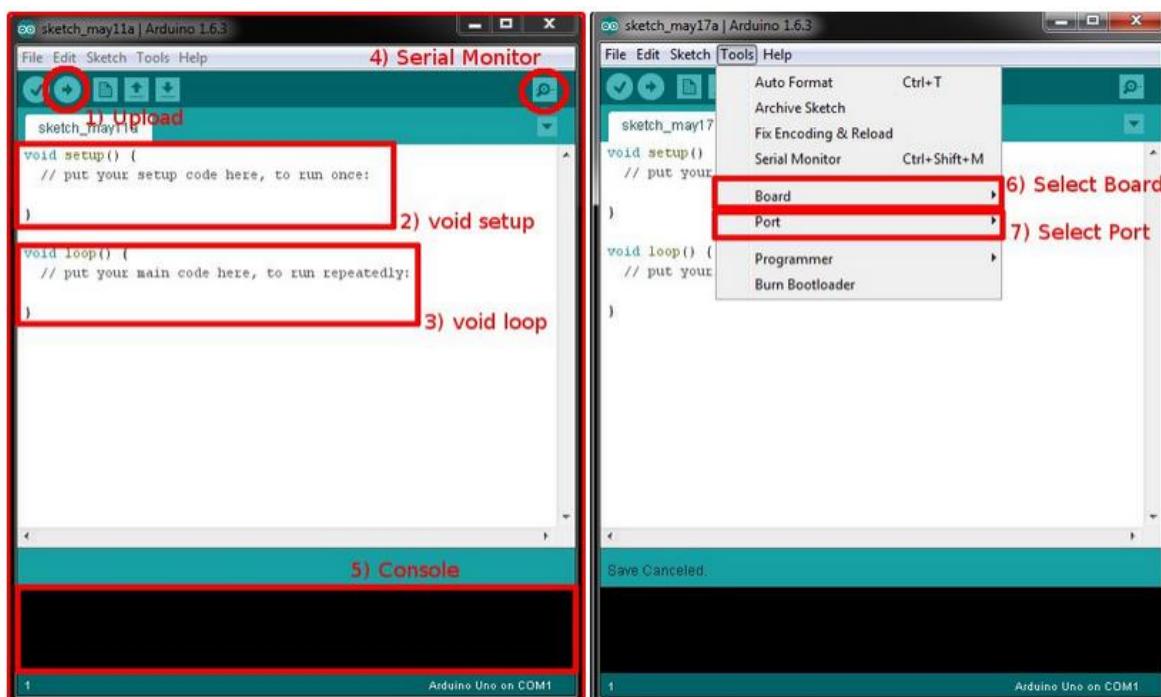


Postupak će sam izdvojiti i instalirati datoteke za odgovarajuće izvršavanje Arduinovog programa (IDE). Kada se Arduinov program (IDE) pravilno instalira, možeš se vratiti na Početnu stranicu (Getting Started Home) i odabrati svoju ploču s popisa na desnoj strani stranice.

5.2. Objasnjenje programa Arduino IDE

Nakon instalacije Arduina, otvori ga i pogledaj prethodne fotografije gdje smo ti pokazali nekoliko ključnih dijelova IDE-a (Integrated Development Environment – Integrirani razvojni okoliš).

1. U **krugu** označenom brojem 1 se nalazi gumb za učitavanje. Kada uključiš Arduino u svoje računalo, klikni taj gumb kako bi učitao svoj kod u Arduino.
2. **Void postavka ili setup** je odjeljak tvog koda koji će se pokrenuti jednom i samo jednom. Kada uključiš Arduino ili nakon resetiranja, on će pokrenuti sve kodove koje si spremio u odjeljak s twojim void postavkama.
3. **Void petlja ili loop** je odjeljak tvog koda koji će neprestano raditi. Nakon pokretanja void postavki, svi tvoji kodovi u void petlji će raditi, Arduino će otići natrag na početak odjeljka s twojom void petljom i ponovno će je pokrenuti. To se stalno ponavlja i zato se zove petlja.
4. Gumb **serijskog ekrana**. Ovim se gumbom otvara serijski ekran. Serijski ekran ti omogućava slanje komandi Arduinu i primanje poruka natrag s Arduina. Kako se to radi ćemo prodiskutirati kasnije.
5. Ovo je **Arduino konzola**. Arduino će ovdje otisnuti poruke o greški ako je došlo do problema tijekom njegovog učitavanja na ploču ili tijekom izrade koda.
6. U **meniju s alatima (tools menu)** se na samom vrhu nalazi opcija „ploča“ (Board) koju možeš odabrati. Upotrijebi je za odabir ispravnog tipa ploče. Obično je tip Arduino ploče napisan negdje na uređaju.
7. Ovdje se nalazi opcija odabira **com port** koja će ti priključiti Arduino na računalo. Ako imaš poteškoća s njegovim pronalaženjem, isključi Arduino, pogledaj popis i ponovno uključi Arduino. Novi com port je upravo taj za tvoj Arduino.



Slika 19. Arduino IDE okoliš



5.3. Učitavanje Arduinovog programa u mjerne uređaje i ispitivanje

Imamo dva programa za naše projekte, mjerjenje TTVSUmearing za projekt 1 i mjerjenje PMSmeasuring za projekt 2. Programi će se pohraniti u obliku zip datoteka i bit će dostupni preko Google Drivea.

TTVSUmearing - program za mjerjenje svjetla, temperature, tlaka i vlage.

PMSmeasuring - program za mjerjenje mikro čestica u zraku.

5.3.1. Učitavanje programa u naše projekte Projekt 1 – mjerjenje TTVSU measurement

- ➔ Pohrani program na svoje računalo.
- ➔ Otvori Arduino IDE na svom računalu.
- ➔ Otvori program za mjerjenje "TTVSUmearing".

Korak 1. – nacrt (sketch)

Nakon otvaranja nacrta ili sketcha, moramo instalirati nekoliko biblioteka (libraries). Klikni nacrt (Sketch), zatim uključi biblioteku (Include Library), a potom upravljam bibliotekama (Manage Libraries). Provjeri da li imaš sve te biblioteke. Ako ih nemaš, instaliraj ih:

- ➔ Blynk,
- ➔ WiFi101,
- ➔ Adafruit BMP085,
- ➔ SimpleDHT,
- ➔ PMS.

U Google dokumentima ćeš naći email poruku, odnosno još jednu biblioteku (library) - APDS9200. Pohrani je (.zip file) negdje na svom računalu.

Klikni na nacrt (Sketch), zatim klikni na uključi biblioteku (Include Library), pa dodaj zip biblioteku (Add zip library) i otvori datoteku APDS9200.zip file.



```
#include <Wire.h>
#include <Adafruit_BMP085.h>
#include "APDS9200.h"
#include <SimpleDHT.h>

#include <SPI.h>
#include <WiFi101.h>
#include <BlynkSimpleMKR1000.h>

int pinDHT11 = 2; //DHT11 sensor pin connected to digital pin 2

Adafruit_BMP085 bmp; //define object bmp
APDS9200 light; //define object light
SimpleDHT11 dht11; //define object dht11

BlynkTimer timer; //using Blynk timer

double templ80[6]; //variable for measuring temperature
double templ80sr = 0; //variable for measuring temperature
double tlakl80[6]; //variable for measuring pressure
double tlakl80sr = 0; //variable for measuring pressure

double uv[6]; //variable for measuring uv
double uvindexsr = 0; //variable for measuring uv
double svjetlo[6]; //variable for measuring level of light
double svjetlosr = 0; //variabile za mjerjenje level of light

double vlagall[6]; //variable for measuring humidity
double vlagallsr = 0; //variable for measuring humidity

int brojac = 0; //variable counter

char auth[] = " "; //enter the autentification token for project 1

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = " "; //enter the network name
char pass[] = " "; //enter the network password

void setup() {
    Serial.begin(9600); //serial connection initialization
    if (!bmp.begin()) { //checking of MBP sensor initialization
        Serial.println("BMP senzor nije pronadjen");
        while (1) {} //stop the program
    }
    Wire.begin(); //starting the wire communication

    Blynk.begin(auth, ssid, pass);
    timer.setInterval(10000L, myTimerEvent); //setting the values interval on 10 sec.
    brojac = 0; //setting the starting value of the counter
}

void myTimerEvent() //function active every 10 sec. using timer
{
    if (brojac < 6) {

        //----TEMPERATURE MEASUREMENT AND PRINTING THE VALUE-----TEMPERATURE MEASUREMENT AND PRINTING THE VALUE-----
        templ80[brojac] = bmp.readTemperature(); //temp180[brojac] = bmp.readTemperature();
        tlakl80[brojac] = bmp.readPressure(); //tlakl80[brojac] = bmp.readPressure();
        templ80sr = templ80sr + templ80[brojac];
        tlakl80sr = tlakl80sr + tlakl80[brojac];

        Serial.println("-----BMP180");
        Serial.print("Temperature = ");
        Serial.print(templ80[brojac]);
        Serial.println(" *C");

        Serial.print("Pressure = ");
        Serial.print(tlakl80[brojac]);
        Serial.println(" Pa");
        //-----
    }
}
```



```
-----LIGHT LEVEL MEASUREMENT AND PRINTING THE VALUE-----LIGHT LEVEL MEASUREMENT AND PRINTING THE VALUE-----
light.enableUV();
delay(500);
uv[brojac] = light.getUV();
light.enableLight();
delay(500);
svjetlo[brojac] = light.getLight();
uvindexsr = uvindexsr + uv[brojac];
svjetlosr = svjetlosr + svjetlo[brojac];
Serial.println("-----UV&LIGHT");
Serial.print("UV:");
Serial.println(uv[brojac]);
Serial.print("Svjetlo:");
Serial.println(svjetlo[brojac]);
//-----

-----HUMIDITY MEASUREMENT AND PRINTING THE VALUE-----HUMIDITY MEASUREMENT AND PRINTING THE VALUE-----
byte temperature = 0;
byte humidity = 0;
int err = SimpleDHTErrSuccess;
if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
    //Serial.print("Read DHT11 failed, err="); Serial.println(err); delay(1000);
    //return;
}

//temp11 = (double)temperature;
vlagall[brojac] = (double)humidity;
vlagallsr = vlagallsr + vlagall[brojac];
Serial.println("-----DHT11");
Serial.print(vlagall[brojac]); Serial.println(" %");
//-----

brojac++;

} else {
    -----TEMPERATURE AND PRESSURE MEASUREMENT AND PRINTING THE VALUE-----
    temp180sr = temp180sr / 6;
    tlak180sr = tlak180sr / 600;
    Blynk.virtualWrite(V1, temp180sr);
    Blynk.virtualWrite(V2, tlak180sr);

    Serial.print("Srednja temp = ");
    Serial.print(temp180sr);
    Serial.println(" *C");
    Serial.print("Srednji tlak = ");
    Serial.print(tlak180sr);
    Serial.println(" hPa");

    temp180sr = 0;
    tlak180sr = 0;
//-----
```



```
-----UV AND LIGHT LEVEL-----UV AND LIGHT LEVEL-----UV AND LIGHT LEVEL-----UV AND LIGHT LEVEL-----  
uvindexsr = uvindexsr / 6;  
svjetlosr = svjetlosr / 6;  
if ((uvindexsr >= 0) && (uvindexsr < 275)) {  
    uvindexsr = 1;  
} else if ((uvindexsr >= 275) && (uvindexsr < 525)) {  
    uvindexsr = 2;  
} else if ((uvindexsr >= 525) && (uvindexsr < 775)) {  
    uvindexsr = 3;  
} else if ((uvindexsr >= 775) && (uvindexsr < 1050)) {  
    uvindexsr = 4;  
} else if ((uvindexsr >= 1050) && (uvindexsr < 1325)) {  
    uvindexsr = 5;  
} else if ((uvindexsr >= 1325) && (uvindexsr < 1675)) {  
    uvindexsr = 6;  
} else if ((uvindexsr >= 1675) && (uvindexsr < 1960)) {  
    uvindexsr = 7;  
} else if ((uvindexsr >= 1960) && (uvindexsr < 2240)) {  
    uvindexsr = 8;  
} else if ((uvindexsr >= 2240) && (uvindexsr < 2520)) {  
    uvindexsr = 9;  
} else if ((uvindexsr >= 2520) && (uvindexsr < 2800)) {  
    uvindexsr = 10;  
} else if ((uvindexsr >= 2800) && (uvindexsr < 3080)) {  
    uvindexsr = 11;  
} else {  
    uvindexsr = 12;  
}  
  
Blynk.virtualWrite(V3, uvindexsr);  
Blynk.virtualWrite(V4, svjetlosr);  
  
Serial.print("UV index = ");  
Serial.println(uvindexsr);  
Serial.print("Svjetlost = ");  
Serial.println(svjetlosr);  
  
uvindexsr = 0;  
svjetlosr = 0;  
  
-----  
-----HUMIDITY MEASUREMENT-----HUMIDITY MEASUREMENT-----HUMIDITY MEASUREMENT-----  
vlagallsr = vlagallsr / 6;  
  
Blynk.virtualWrite(V5, vlagallsr);  
  
Serial.print("Srednja vlaga: ");  
Serial.print(vlagallsr); Serial.println(" %");  
vlagallsr = 0;  
-----  
  
brojac = 0; //setting counter to 0  
}  
  
void loop() {  
    Blynk.run(); //function for server connection and sending data  
    timer.run(); //timer function  
}
```

Slika 20. Programski kod projekta 1



Korak 2 – priključivanje Arduina (projekt 1)

Prikluči mjerač Arduino za mjerjenje svjetla, temperature, tlaka i vlage. Provjeri u Arduino programskom okružju da li si na Alati->Ploča (Tools-> Board) odabrao (odabrala) Arduino MKR1000.

Na registrirani ćeš e-mail dobiti autorizacijski token za **prvi projekt**, a ti ga moraš upisati u nacrt (sketch).

```
char auth[] = "";      // insert the blink token into the quotation marks
char ssid[] = "";     // insert the wifi network name into the quotation marks
char pass[] = "";     // insert the password for connection to the wifi newtwork into
                      the quotation marks
```

Učitaj program u Arduino ploču. Kada se učitavanje završi, otvor serijski ekran (Serial Monitor) klikom na ikonu s povećalom u desnom gornjem ugлу Arduino programskog okružja.

Na novom bi prozoru trebao imati izmjerene vrijednosti svih priključenih davača. Vrijednosti će se otiskivati svakih 10 sekundi. Provjeri da li se vrijednosti na prvom nacrtu načinjenom uz pomoć aplikacije Blynk ažuriraju (svakih 60 sekundi).

5.3.2. Učitavanje programa u projekt 2 – mjerjenje *PMSmeasurement*

Korak 1 – priključivanje Arduina (projekt 2)

Prikluči Arduino MKR 1000 za mjerjenje mikro čestica u zraku (kvalitete zraka). Provjeri u Arduino programskom okružju da li si na Alati->Ploča (Tools-> Board) odabrao Arduino MKR1000.

Na registrirani ćeš e-mail dobiti autorizacijski token za drugi projekt, a ti ga moraš upisati u nacrt (sketch).

```
char auth[] = "";      // insert the blink token into the quotation marks
char ssid[] = "";     // insert the wifi network name into the quotation marks
char pass[] = "";     // insert the password for connection to the wifi newtwork into
                      the quotation marks
```

Učitaj program u Arduino ploču. Kada se učitavanje završi, otvor serijski ekran (Serial Monitor) klikom na ikonu s povećalom u desnom gornjem ugлу Arduino programskog okružja.



```
#include "PMS.h"

#include <SPI.h>
#include <WiFi101.h>
#include <BlynkSimpleMKR1000.h>

PMS pms(Serial1);
PMS::DATA data;

double pm1sr=0;
double pm25sr=0;
double pm10sr=0;

char auth[] = "";
char ssid[] = "";
char pass[] = "";

void setup()
{
    Serial.begin(9600);
    Serial1.begin(9600);
    Blynk.begin(auth, ssid, pass);
}

void loop()
{
    if (pms.read(data))
    {

        pm1sr = data.PM_AE_UG_1_0;
        pm25sr = data.PM_AE_UG_2_5;
        pm10sr = data.PM_AE_UG_10_0;

        Serial.println("Data:");

        Serial.print("PM 1.0 (ug/m3): ");
        Serial.println(pm1sr);

        Serial.print("PM 2.5 (ug/m3): ");
        Serial.println(pm25sr);

        Serial.print("PM 10.0 (ug/m3): ");
        Serial.println(pm10sr);

        Serial.println();

        Blynk.virtualWrite(V10, pm1sr);
        Blynk.virtualWrite(V11, pm25sr);
        Blynk.virtualWrite(V12, pm10sr);
    }
}
```

Slika 21. Programski kod za projekt 2

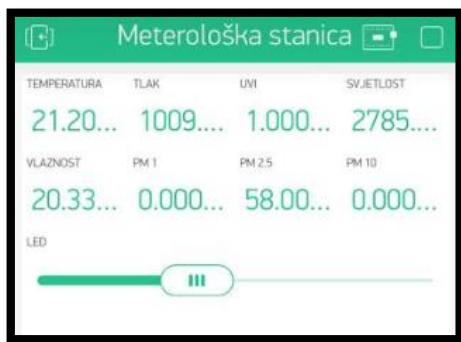
Na novom bi prozoru trebao imati izmjerene vrijednosti svih priključenih davača. Vrijednosti će se otiskivati svakih 10 sekundi. Provjeri da li se vrijednosti na prvom nacrtu (sketchu) načinjenom uz pomoć aplikacije Blynk ažuriraju (svakih 60 sekundi).

5.3.3. Spajanje (merging) programskih kodova

Povrh toga, programski se kodovi obaju projekata mogu spojiti u jedan projekt. Spajanjem programa se pohranjuje jedna platforma MKR1000 za druge projekte. Nakon spajanja svih izmjerениh vrijednosti obaju programa, oni će se prikazivati u jednoj aplikaciji Blynk i na serijskom ekranu u programu Arduino IDE (Slika 22).



Co-funded by the
Erasmus+ Programme
of the European Union



Slika 22. Vrijednosti spojene u aplikaciji Blynk