

Razvoj softvera 2020.



Semafor u Vukovaru (IoT Arduino)

Tehnička dokumentacija

Osnovna škola Nikole Andrića Vukovar

Autor: Stefan Brković, učenik 8.razreda

Mentor: Milenko Šljukić, učitelj informatike i TK

SADRŽAJ:

Uvod	3
O autoru	3
Ideja	3
Radno okruženje	6
<i>Preuzimanje Arduino IDE</i>	7
<i>Instalacija Arduino IDE</i>	8
<i>Pokretanje i podešavanje Arduino IDE</i>	9
<i>Instalacija Blynk aplikacije</i>	11
<i>Početak izrade IoT projekta u Blynk aplikaciji</i>	11
Realizacija projekta Semafor u Vukovaru	12
<i>Šema spajanja</i>	12
<i>Slika spajanja</i>	12
<i>Programski kod</i>	13
<i>Blynk aplikacija na pametnom telefonu</i>	17
Zaključak – budućnost projekta	17

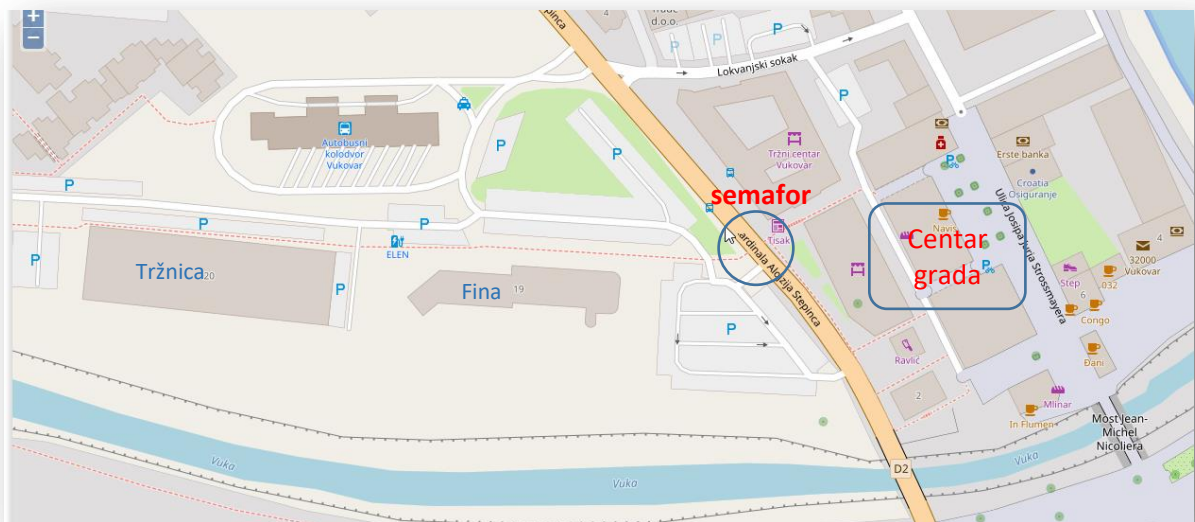
Uvod

O autoru

Zovem se Stefan Brković, učenik sam 8.razreda OŠ Nikole Andrića iz Vukovara. Od četvrtog razreda pohađam izvannastavne aktivnosti vezane za programiranje i tehničku kulturu – oblast automatike. U petom i šestom razredu sudjelovao sam na natjecanju u Infokupu kategorija algoritmi. Bio sam učesnik županijskog natjecanja. Pored informatike omiljene su mi oblasti matematika i fizika. U posljednje vrijeme uža oblast mog interesiranja je primjena programiranja u oblasti automatike.

Ideja

Ideja ovog rada zasniva se na problematici rada semafora u gradu Vukovaru gdje je vrijeme prolaska automobila 64 sekunde a pješaka 21. To je semafor na glavnoj i jedinoj cesti kroz Vukovar i svi automobili moraju ga proći. Prometni policajac na licu mjesta ili iz ureda (prateći situaciju nadzornom kamerom) mogao bi mobilnim telefonom (aplikacijom za mobilni telefon) uključivati prolaz automobila ili pješaka prema situaciji. Ovako se dešava da automobili ili pješaci bez potrebe dugo čekaju na prolaz.

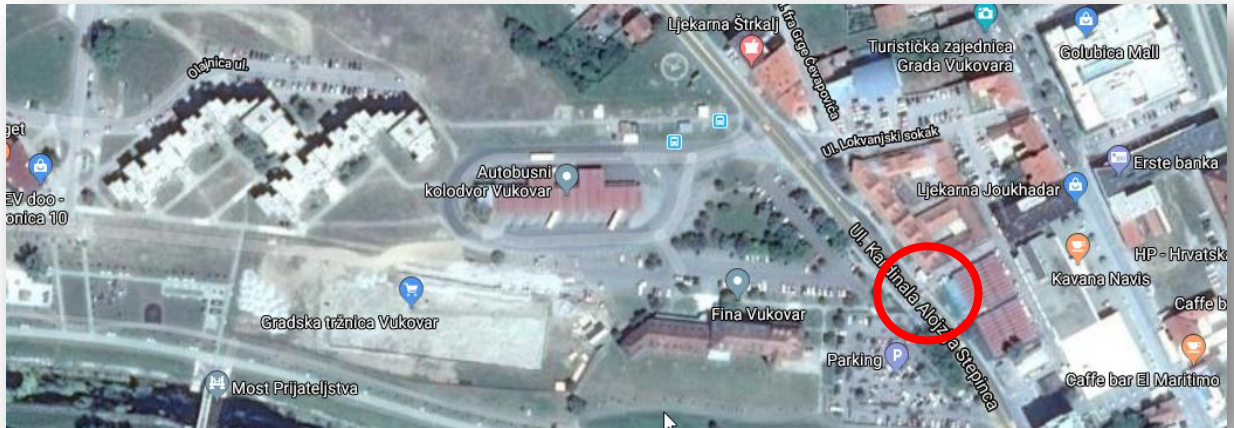


Da bi iz centra grada otišli na tržnicu, platili račune u Fini (financijskoj agenciji), autobusni kolodvor, dio grada koji se naziva Olajnica građani moraju preći prometnicu na pješačkom prijelazu (na slici oznaka **semafor**).

Nekoliko puta u toku dana (kada učenici sa autobusnog kolodvora idu prema centru grada i obratno) za 64s koliko traje zeleno svijetlo na semaforu za automobile sa obje strane pješačkog prijelaza nakupi se toliko pješaka da vrijeme od 21s za pješake nije dovoljno za prijelaz prometnice.

Ova problematika naročito dolazi do izražaja srijedom i subotom kada su tržnični dani u Vukovaru.

Ovo je satelitski snimak tog dijela grada.



Evo i slika samog semafora gdje se vidi vrijeme prolaska kako automobila (64s)



tako i pješaka (21s).



Promjena režima prometa traje 3s.



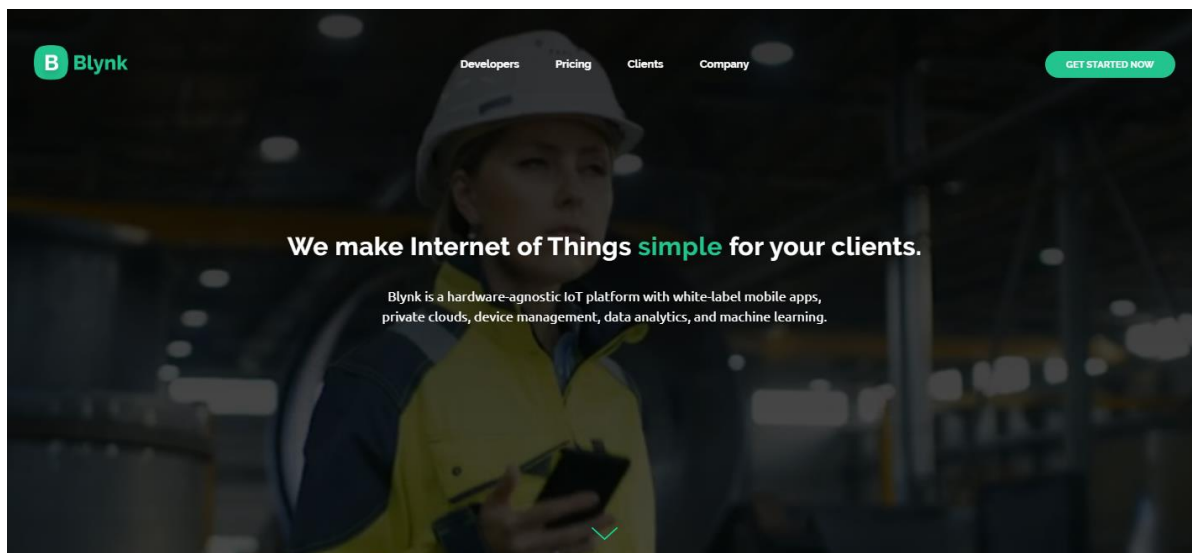
Radno okruženje

IoT - Internet of Things, na hrvatskom - Internet stvari, je tehnologija koja omogućava povezivanje tehničkih uređaja putem interneta i njihovo upravljanje.

Nešto što je do jučer bilo nezamislivo danas postaje sasvim uobičajena pojava a to je da putem pametnih telefona uključujemo grijanje u kući dok smo na putu, imamo sliku uživo dešavanja u kući, provjeravamo da li su električni uređaji isključeni i da ih uključimo/isključimo ovisno o potrebi.

Rješavanje našeg problema upravljanja radom semafora zasnovat ćemo na IoT tehnologiji tako što ćemo upravljati putem interneta i to na način da mobilnom aplikacijom upravljamo modelom semafora koji je spojen na našu Arduino platformu.

Različite su platforme koje omogućavaju interakciju Arduina i mobilnih uređaja, ali mi ćemo u našem projektu koristiti Blynk platformu. Više o platformi možete pogledati na linku: <http://blynk.io/>



Izradom korisničkog sučelja u Blynk aplikaciji na mobilnom telefonu omogućena je direktna komunikacija aplikacije i našeg Arduino hardvera putem Blynkovog cloud sustava.



Da bismo ostvarili komunikaciju između telefona i uređaja nije potrebno računalo. Arduino MKR1000 (na slici lijevo) ima u sebi integriranu mogućnost povezivanja na WiFi tako da nam nije potreban nikakav dodatni hardver.

Inače, Arduino je mikrokontroler pomoću kojeg možemo spojiti različite elektroničke komponente u jednu cjelinu te programirati ga da bi ostvarili željeni rad cijelog sklopa. Tako ćemo mi u našem projektu spojiti LED diode na Arduino MKR1000 i napisati program za rad semafora kojim ćemo upravljati (zahvaljujući Blynk-u) mobilnim telefonom.

Da bi realizirali cijeli projekt moramo proći kroz sljedeće korake:

- Preuzimanje Arduino IDE
- Instalacija Arduino IDE
- Preuzimanje Blynk aplikacije
- Izraditi prvi jednostavni IoT projekt

Preuzimanje Arduino IDE

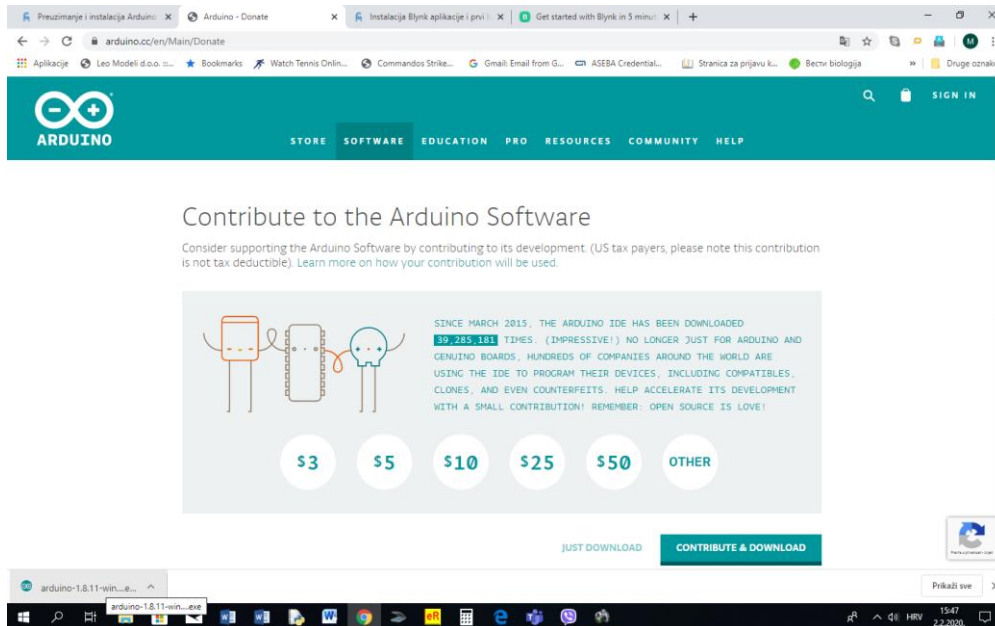
Da bi mogli započeti s radom, potrebno je na računalo instalirati softver za programiranje Arduino mikrokontrolera. Softver je otvorenog koda i potpuno je besplatan, a može se preuzeti na web stranici Arduino.cc, odnosno na sljedećem linku: <https://www.arduino.cc/en/Main/Software>



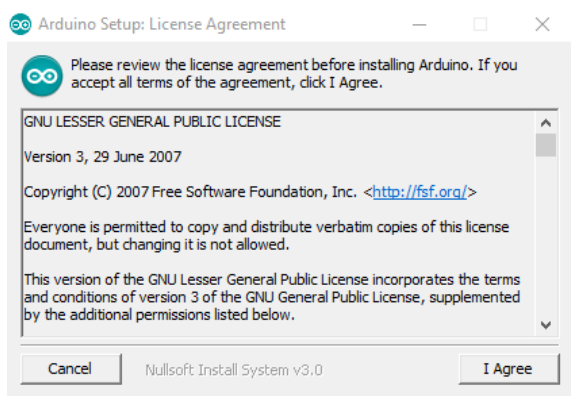
Nakon preuzimanja Arduino IDE slijedi pokretanje instalacije koja protječe vrlo jednostavno sa nekoliko klikova na gumb Next.

Instalacija Arduino IDE

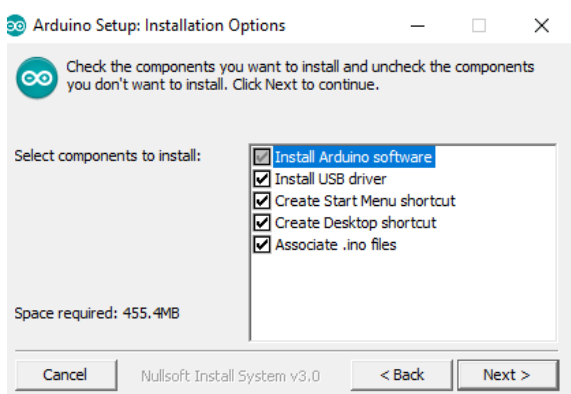
Nakon preuzimanja



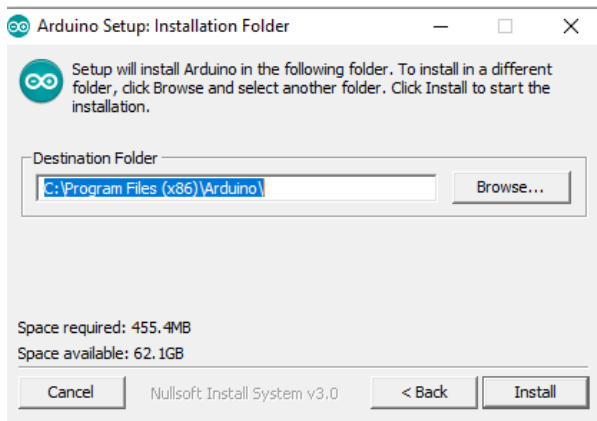
pokrećemo instalaciju:



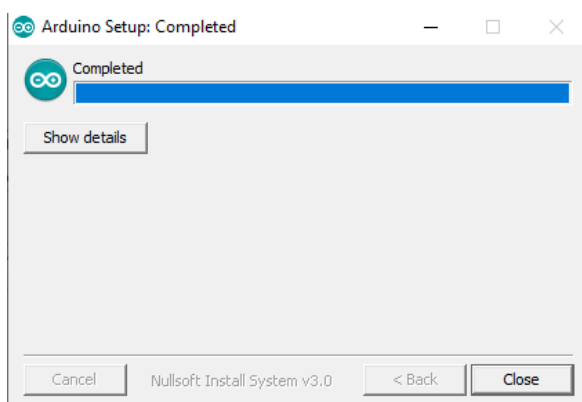
Klik na I Agree



Klik na Next

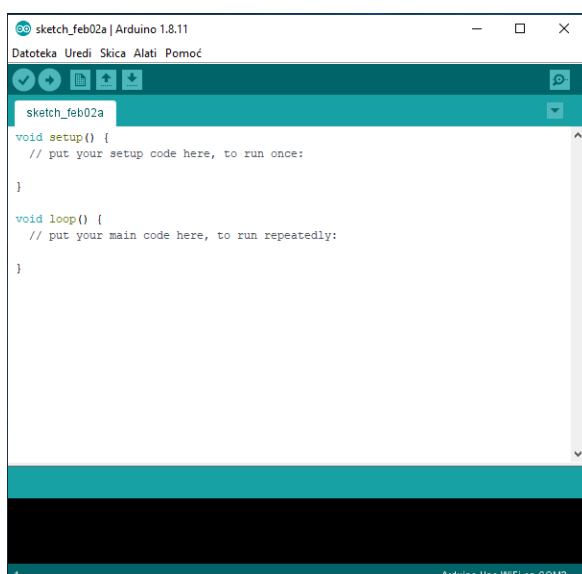


Klikom na Install postupak instalacije započinje, a klikom na Close postupak instalacije se završava.

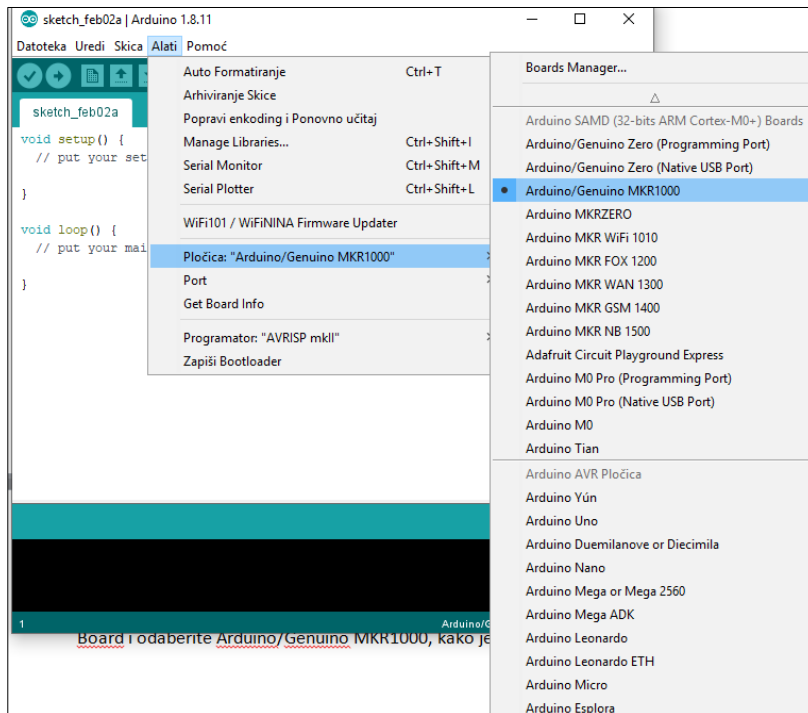


Pokretanje i podešavanje Arduino IDE

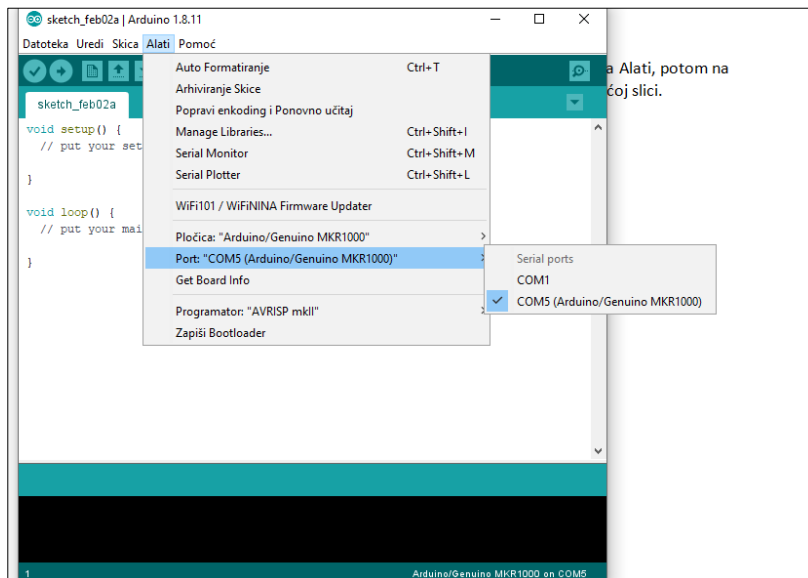
Program pokrećemo putem izbornika Start ili prečaca na radnoj površini. Nakon pokretanja dobijemo programsko okruženje kako je prikazano na sljedećoj slici.



Kako mi koristimo Arduino MKR 1000, to odmah treba podesiti. Kliknite na Alati, potom na Pločice i odaberite Arduino/Genuino MKR1000, kako je prikazano na sljedećoj slici.



Da je računalo uspješno prepoznalo priključenu Arduino pločicu, vidjet ćemo pojavljuje li se u listi priključnih uređaja. Kliknite na Alati, potom Port i pogledajte pojavljuje li se na listi COMx (Arduino/Genuino MKR1000), kao na slici u nastavku. Za kraj odaberite s liste stavku u kojoj piše Arduino/Genuino MKR1000.

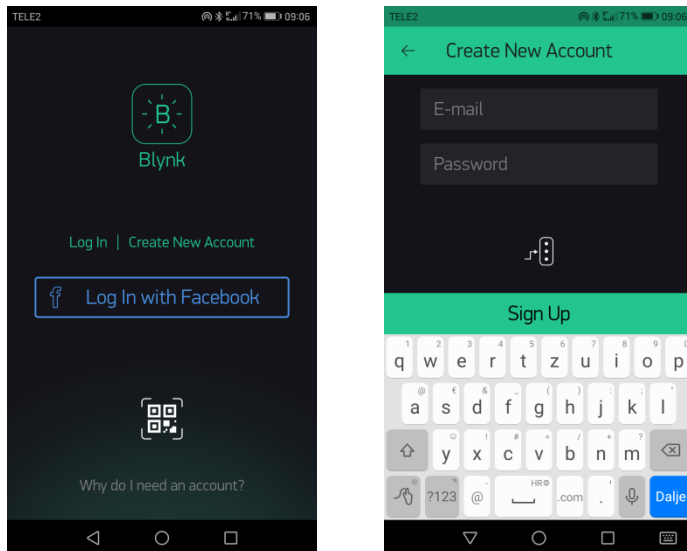


Ovu instalaciju i podešavanja za pločicu MKR1000 vršimo samo jednom i sve je spremno za pisanje aplikacije.

Instalacija Blynk aplikacije

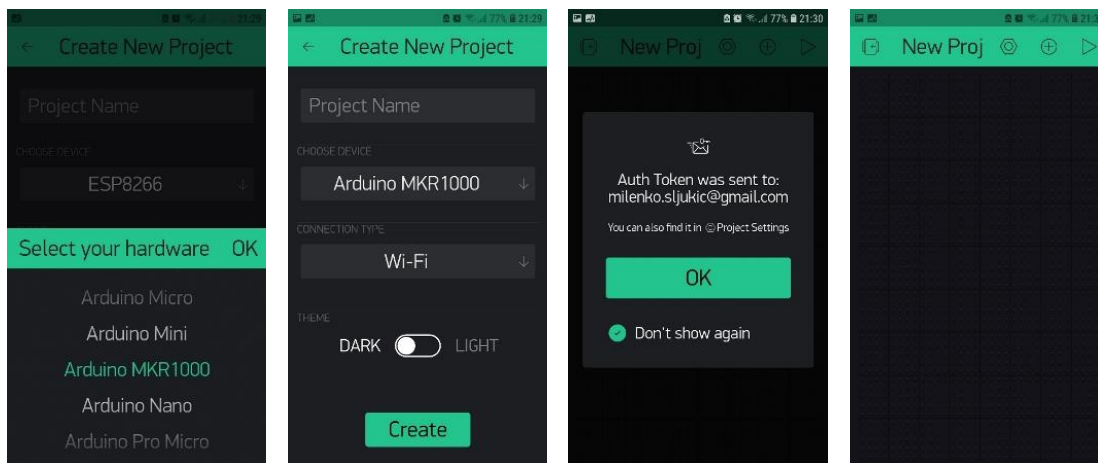
Aplikaciju Blynk možemo preuzeti unutar Trgovine Play ukoliko radite na Android operativnom sustavu odnosno preko Trgovine aplikacija (App store) ako radite na iOS operativnom sustavu. Aplikaciju preuzimamo pretragom po ključnoj riječi Blynk.

Nakon preuzimanja otvorimo aplikaciju i izvršimo registraciju klikom na Create New Account. Registracija je potrebna kako bi se naši projekti spremali direktno na Blynk serveru.



Početak izrade IoT projekta u Blynk aplikaciji

Poslije registracije kreiramo novi projekt odabirom New Project opcije.



Poslije zadavanja zahtjeva za kreiranjem novog projekta i izborom uređaja Arduino MKR1000 prikazuje se obavijest da je stvoren autentifikacijski token te da je isti poslan na e-mail. Ovaj token je jedinstven za svaki projekt i služi za povezivanje hardvera s Blynk aplikacijom. Iskoristiti ćemo ga kasnije u programskom kodu za Arduino.

Na ekranu nam se prikazuje prazna aplikacija u kojoj možemo početi stvarati našu aplikaciju za upravljanje IoT projektom.

Realizacija projekta Semafor u Vukovaru

Da bi realizirali naš projekt Semafor u Vukovaru (IoT Arduino) IoT – na Arduino MKR1000 spojimo pet svjetlećih dioda (crvena, žuta, zelena na lijevoj strani eksperimentalne pločice koje predstavljaju semafor za automobile, te crvena i zelena na desnoj strani koje predstavljaju semafor za pješake).

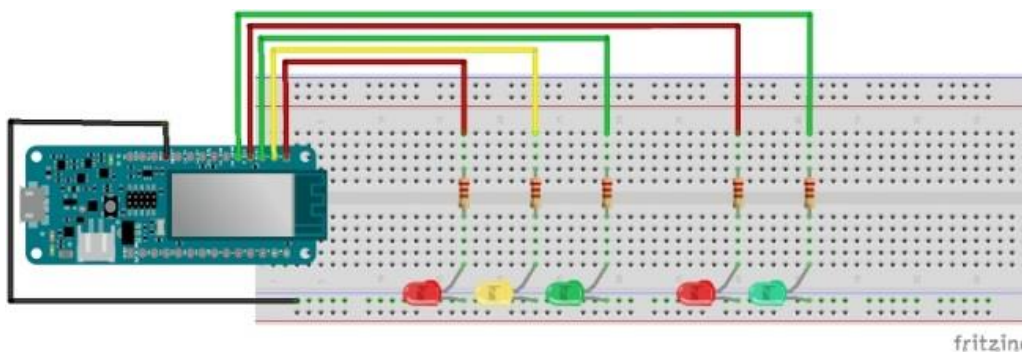
U mobilnoj aplikaciji kreirat ćemo tri gumba:

JEDNAKO - jednako je vrijeme kako za pješake tako i za automobile (u programskom kodu to će biti varijabla V1).

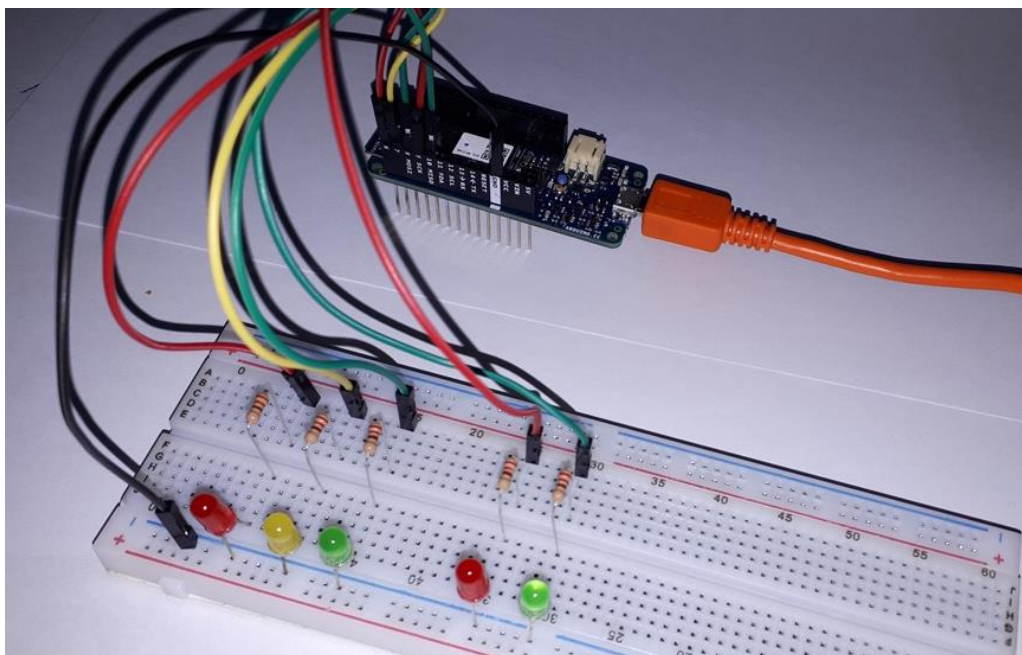
AUTOMOBILI - vrijeme za prolazak automobila kao i do sada 65s, a za pješake 23s (u programskom kodu to će biti varijabla V2).

PJESACI - vrijeme za prolazak pješaka 60s, a za automobile 30s (u programskom kodu to će biti varijabla V3).

Šema spajanja



Slika spajanja



Programski kod

```
/* Upravljanje radom semafora pomoću pametnog telefona. Ideja je
 * nastala promatranjem stvarne prometne situacije u gradu Vukovaru.
 */

//#include <Blynk.h>

/*****
 *****/

O Blynk-u:

Download latest Blynk library here:
https://github.com/blynkkk/blynk-library/releases/latest

Blynk is a platform with iOS and Android apps to control
Arduino, Raspberry Pi and the likes over the Internet.
You can easily build graphic interfaces for all your
projects by simply dragging and dropping widgets.

Downloads, docs, tutorials: http://www.blynk.cc
Sketch generator:          http://examples.blynk.cc
Blynk community:          http://community.blynk.cc
Follow us:                 http://www.fb.com/blynkapp
                           http://twitter.com/blynk\_app
Blynk library is licensed under MIT license
This example code is in public domain.
*****/

/* Na startu programa se nalaze biblioteke za WiFi i Blynk */

#define BLYNK_PRINT SerialUSB

#include <SPI.h>
#include <WiFi101.h>
#include <BlynkSimpleWiFiShield101.h>

char auth[] = "74f4b9b7a442486987c4b5e85f52259e";
//Jedinstveni token za Blynk projekt - autentifikacijski
//token. Dobije se na mail prilikom izrade dijela projekta
//na mobilnom telefonu i poslije ubaci u program.

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "SAJM"; //Naziv WiFi mreže na koju se Arduino MKR1000 povezuje
char pass[] = "57mile1959"; //Zaporka za istu mrežu

int LedCrA = 6; //definiraj LedCrA = 2 crveno svjetlo za automobile
```

```

int LedZu = 7;           //definiraj LedZu = 3 zuto svjetlo za promjenu režima prometa
int LedZeA = 8;         //definiraj LedZeA = 4 zeleno svjetlo za automobile
int LedCrP = 9;         //definiraj LedCrP = 5 crveno svjetlo za pjesake
int LedZeP = 10;        //definiraj LedZeP = 6 zelno svjetlo za pjesake

/* Imenovanje (deklaracija) logičkih varijabli čije vrijednosti (V1, V2, V3)
 * ćemo mijenjati putem izbornika u aplikaciji na pametnom telefonu.
 * Izbornik će nam omogućiti da izaberemo jedan od tri režima prometa:
 * - JEDNAKO(V1) - ujednačen vremenski iznos kako za pješake, tako i za automobile.
 * - AUTOMOBILI - vrijeme u korist automobila.
 * - PJESACI - vrijeme u korist pješaka.
 * (TelVr - telefon vrijednost (1,2 ili 3)
 */
boolean TelVr1 = 0;
boolean TelVr2 = 0;
boolean TelVr3 = 0;

//Blynk rutina za zapisivanje podataka putem virtualnog PIN-a

BLYNK_WRITE(V1)
{
  TelVr1 =! TelVr1; //Logičkim NE mijenjamo stanje varijable između 0 i 1
}

BLYNK_WRITE(V2)
{
  TelVr2 =! TelVr2;
}

BLYNK_WRITE(V3)
{
  TelVr3 =! TelVr3;
}

void setup()
{
  // Debug console
  SerialUSB.begin(9600);

  Blynk.begin(auth, ssid, pass); //Povezivanje na WiFi

  //Postavljamo led diode u izlazni način rada
  pinMode(LedCrA, OUTPUT); //postavi izvod LedCrA kao izlazni
  pinMode(LedZu, OUTPUT); //postavi izvod LedZu kao izlazni
  pinMode(LedZeA, OUTPUT); //postavi izvod LedZeA kao izlazni
  pinMode(LedCrP, OUTPUT); //postavi izvod LedCrP kao izlazni
  pinMode(LedZeP, OUTPUT); //postavi izvod LedZeP kao izlazni

```

```

//Postavljanje početnog stanja
digitalWrite(LedCrA, LOW); //isključi crvenu diodu automobili - početno stanje
digitalWrite(LedZu, LOW); //isključi žutu led diodu - početno stanje
digitalWrite(LedZeA, LOW); //isključi zelenu led diodu automobili - početno stanje
digitalWrite(LedCrP, LOW); //isključi crvenu diodu pjesaci - početno stanje
digitalWrite(LedZeP, LOW); //isključi zelenu diodu pjesaci - početno stanje
}

//Dio programa koji se stalno ponavlja

void loop(){
  Blynk.run();
  // Pomoću if naredbe aktivirat ćemo jedan od tri režima rada.
  //Kod JEDNAKOG načina rada: ovdje je
  //zeleno za pjesake 5s, zeleno za automobile 5s, promjena režima je 1.5s.
  //U stvarnosti je:
  //zeleno za pjesake 21s, zeleno za automobile 64s, promjena rezima je 3s.

  if (TelVr1 == 1) {
    digitalWrite(LedCrA, HIGH); //uključi crvenu led diodu za automobile
    digitalWrite(LedZeP, HIGH); //uključi zelenu led diodu za pješake
    delay(5000); //čekaj 5 s - svjetli LedCrA + LedZeP
    digitalWrite(LedZeP, LOW); //isključi zelenu led diodu za pješake
    digitalWrite(LedCrP, HIGH); //uključi crvenu led diodu za pješake
    digitalWrite(LedZu, HIGH); //uključi žutu led diodu
    delay(2000); //čekaj 2 s - svjetli LedCrA + LedZu + LedCrP
    digitalWrite(LedCrA, LOW); //isključi crvenu led diodu za automobile
    digitalWrite(LedZu, LOW); //isključi žutu led diodu
    digitalWrite(LedZeA, HIGH); //uključi zelenu led diodu za automobile
    delay(5000); //čekaj 5 s - svjetli LedZeA + LedCrP
    digitalWrite(LedZeA, LOW); //isključi zelenu led diodu za automobile
    digitalWrite(LedZu, HIGH); //uključi žutu led diodu
    delay(2000); //čekaj 2 s - svjetli LedZeA + LedCrP
    digitalWrite(LedZu, LOW); //isključi žutu led diodu
    digitalWrite(LedCrP, LOW); //isključi crvenu led diodu za pješake
  } else {
    digitalWrite(LedCrA, LOW); //isključi crvenu diodu - početno stanje
    digitalWrite(LedZu, LOW); //isključi žutu led diodu - početno stanje
    digitalWrite(LedZeA, LOW); //isključi zelenu led diodu - početno stanje
    digitalWrite(LedCrP, LOW); //isključi crvenu diodu pjesaci - početno stanje
    digitalWrite(LedZeP, LOW); //isključi zelenu diodu pjesaci - početno stanje
  }

  //Kod AUTOMOBILSKOG načina rada: ovdje je
  //zeleno za pjesake 5s, zeleno za automobile 10s,

```

```

//promjena rezima je 2s.
if (TelVr2 == 1) {
  digitalWrite(LedCrA, HIGH); //uključi crvenu led diodu za automobile
  digitalWrite(LedZeP, HIGH); //uključi zelenu led diodu za pješake
  delay(5000); //čekaj 5 s - svjetli LedCrA + LedZeP
  digitalWrite(LedZeP, LOW); //isključi zelenu led diodu za pješake
  digitalWrite(LedCrP, HIGH); //uključi crvenu led diodu za pješake
  digitalWrite(LedZu, HIGH); //uključi žutu led diodu
  delay(2000); //čekaj 2 s - svjetli LedCrA + LedZu + LedCrP
  digitalWrite(LedCrA, LOW); //isključi crvenu led diodu
  digitalWrite(LedZu, LOW); //isključi žutu led diodu
  digitalWrite(LedZeA, HIGH); //uključi zelenu led diodu
  delay(10000); //čekaj 10 s - svjetli LedZeA + LedCrP
  digitalWrite(LedZeA, LOW); //isključi zelenu led diodu
  digitalWrite(LedZu, HIGH); //uključi žutu led diodu
  delay(2000); //čekaj 2 s - svjetli LedZeA + LedCrP
  digitalWrite(LedZu, LOW); //isključi žutu led diodu
  digitalWrite(LedCrP, LOW); //isključi crvenu led diodu za pješake
} else {
  digitalWrite(LedCrA, LOW); //isključi crvenu diodu - početno stanje
  digitalWrite(LedZu, LOW); //isključi žutu led diodu - početno stanje
  digitalWrite(LedZeA, LOW); //isključi zelenu led diodu - početno stanje
  digitalWrite(LedCrP, LOW); //isključi crvenu diodu pjesaci - početno stanje
  digitalWrite(LedZeP, LOW); //isključi zelenu diodu pjesaci - početno stanje
}

```

```

//Kod PJESACKOG načina rada: ovdje je
//zeleno za pjesake 10s, zeleno za automobile 5s,
//promjena rezima je 2s.

```

```

if (TelVr3 == 1) {
  digitalWrite(LedCrA, HIGH); //uključi crvenu led diodu za automobile
  digitalWrite(LedZeP, HIGH); //uključi zelenu led diodu za pješake
  delay(10000); //čekaj 10 s - svjetli LedCrA + LedZeP
  digitalWrite(LedZeP, LOW); //isključi zelenu led diodu za pješake
  digitalWrite(LedCrP, HIGH); //uključi crvenu led diodu za pješake
  digitalWrite(LedZu, HIGH); //uključi žutu led diodu
  delay(2000); //čekaj 2 s - svjetli LedCrA + LedZu + LedCrP
  digitalWrite(LedCrA, LOW); //isključi crvenu led diodu
  digitalWrite(LedZu, LOW); //isključi žutu led diodu
  digitalWrite(LedZeA, HIGH); //uključi zelenu led diodu
  delay(5000); //čekaj 5 s - svjetli LedZeA + LedCrP
  digitalWrite(LedZeA, LOW); //isključi zelenu led diodu
  digitalWrite(LedZu, HIGH); //uključi žutu led diodu
  delay(2000); //čekaj 2 s - svjetli LedZeA + LedCrP
  digitalWrite(LedZu, LOW); //isključi žutu led diodu
  digitalWrite(LedCrP, LOW); //isključi crvenu led diodu za pješake
} else {

```

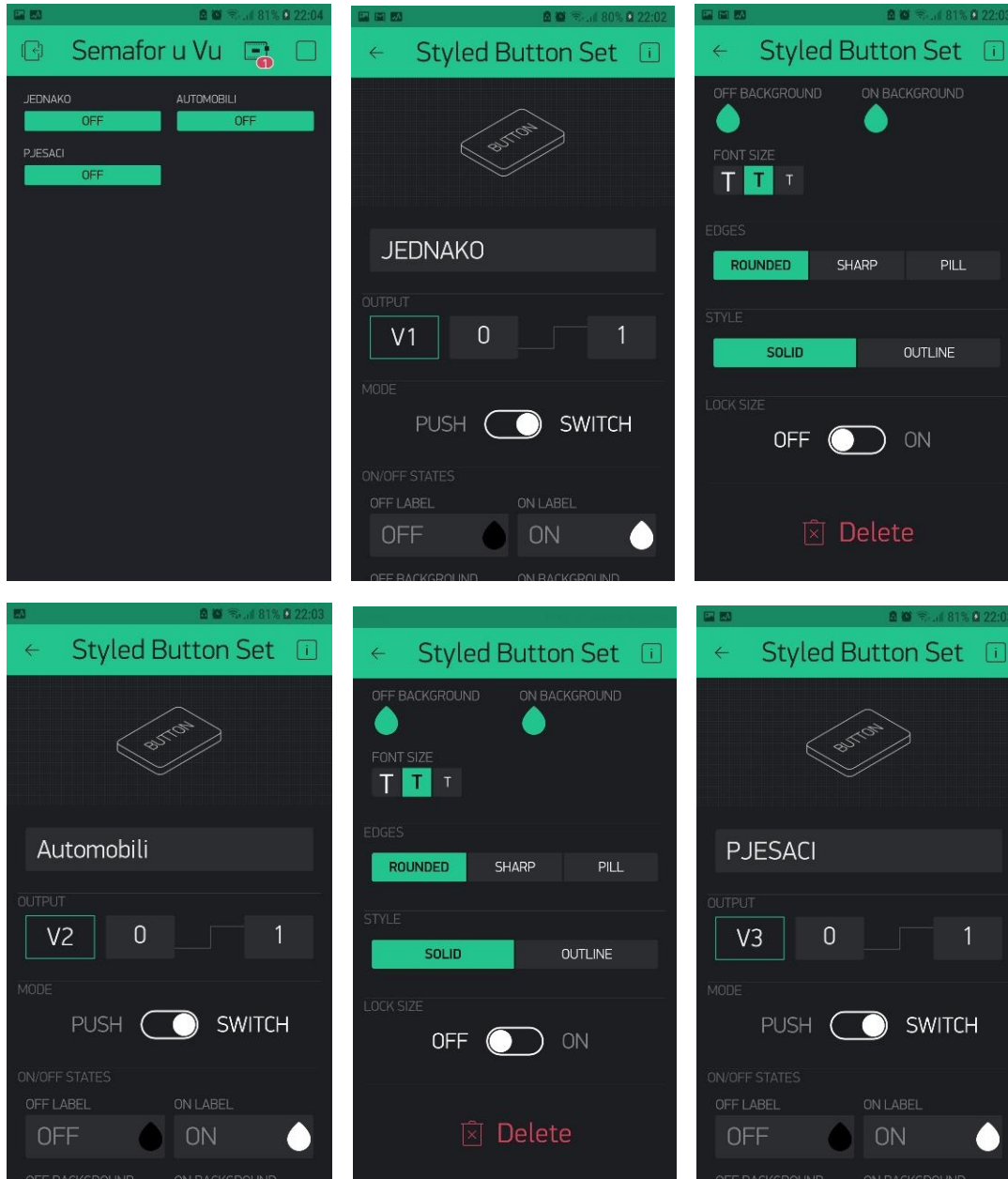


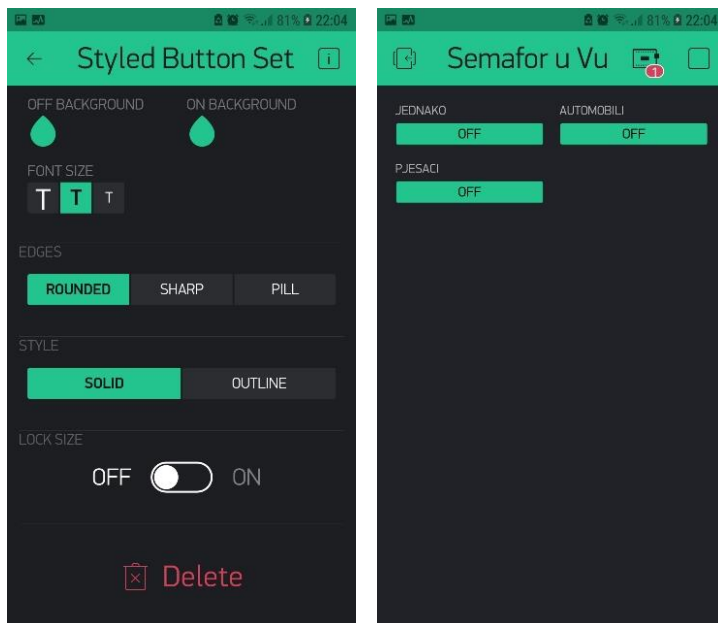
```

digitalWrite(LedCrA, LOW); //isključi crvenu diodu - početno stanje
digitalWrite(LedZu, LOW); //isključi žutu led diodu - početno stanje
digitalWrite(LedZeA, LOW); //isključi zelenu led diodu - početno stanje
digitalWrite(LedCrP, LOW); //isključi crvenu diodu pjesaci - početno stanje
digitalWrite(LedZeP, LOW); //isključi zelenu diodu pjesaci - početno stanje
}
}

```

Blynk aplikacija na pametnom telefonu





Zaključak – budućnost projekta

Rješavanje problema upravljanja radom semafora zasnovanom na IoT tehnologiji tako što ćemo upravljati putem interneta mobilnom aplikacijom sigurno daje velike osnove da se ta vrsta tehnologije može primjenjivati u budućnosti.