



IRIM - Institut za razvoj
i inovativnost mladih

Junior Engineer Academy



Svemirsko vozilo

„RASCO MARS ROVER“

Tehnički opis rada

lipanj 2021., Đurđevac

Sadržaj

1.	OPIS PROJEKTA	3
2.	PROJEKTNI TIM	4
3.	MATERIJALI KORIŠTENI ZA IZRADU	6
3.1.	Arduino MKR1000 WIFI	6
3.2.	mTurnigy nano-tech 1500mAh 3S 35~70C Li-Po	7
3.3.	MKR2 UNO adapter	8
3.4.	5V 3A UBEC.....	8
3.5.	Adafruit Motor Shield V2.....	9
3.6.	Gravity: Analog Ambient Light Sensor For Arduino	10
3.7.	DFRobot LM35 Linear Temperature Sensor	10
3.8.	Solarni panel 1.5W 12V i solarni panel 1 W 6V	11
3.9.	DC motor za zupčanicom.....	12
3.10.	Servo Motor MG995.....	12
3.11.	Mehaničke komponente	13
4.	SHEMA SPAJANJA	14
5.	ARDUINO KOD	17
6.	BLYNK APLIKACIJA ZA DALJINSKO UPRAVLJANJE SVEMIRSKIM VOZILOM	25
7.	FOTOGRAFIJE PROVEDBE PROJEKTA IZRADE SVEMIRSKOG VOZILA U OKVIRU DONACIJSKOG NATJEČAJA Junior Engineer Academy	26
8.	ODRŽAVANJE	28

1. OPIS PROJEKTA

U okviru rada u Sekciji za svemirsku tehnologiju u Strukovnoj školi Đurđevac odlučili smo napraviti model svemirskog vozila koje služi za istraživanje površine planeta Marsa. Ovaj naš model je puno jednostavniji od stvarnog modela i upravljan je daljinski putem aplikacije Blynk. Mozak vozila je mikrokontroler Arduino MKR 1000. Tijelo i noge su napravljeni od dijelova koji su ispisani na 3d pisaču Prusa I3MK3 .Za vožnju mu služe četiri istosmjerna motora a za mIjenjanje smjera kretanja četiri servo motora. Veza između mobilnog uređaja na koje je instalirana Blynk aplikacija i samog vozila ostvaruje se putem WiFi bežične veze. Napajanje elektroničkih sklopova je izvedeno putem solarnih panela na vozilu i putem LiPo baterije u samom vozilu. Vozilo je opremljeno sensorima za mjerenje temperature i nivoa okolnog svjetla .



2. PROJEKTNI TIM

Učenci:



Borna Carek

Tomislav Čukljaš

Mentori :



Ivanka Brček



Željko Brček

Informacije o školi:

Strukovna škola Đurđevac

Dr. Ivana Kranjčeva 5

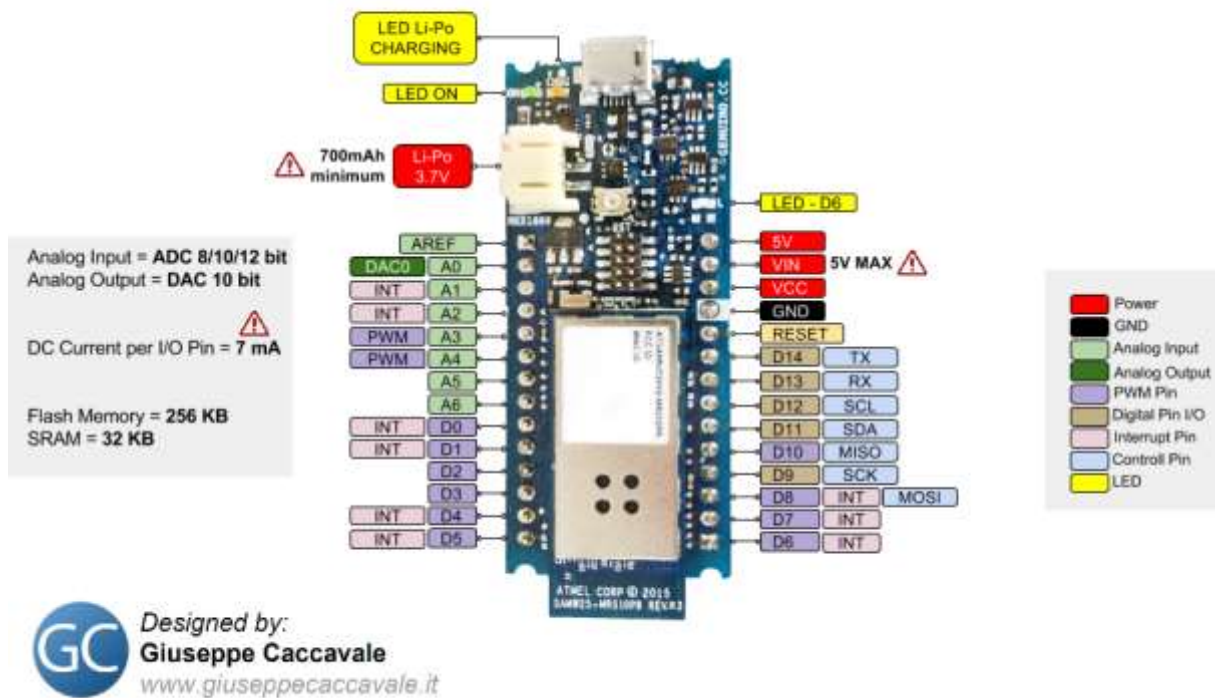
<http://ss-strukovna-djurdjevac.skole.hr/>

3. MATERIJALI KORIŠTENI ZA IZRADU

3.1. Arduino MKR1000 WIFI



MKR1000 PINOUT



Arduino razvojna ploča MKR 1000

Baziran je na Atmel ATSAMW25 SoC (System on Chip), koji je dio SmartConnect obitelji Atmel bežičnih uređaja, posebno dizajniranih za IoT projekte i uređaje.

Posebne značajke

- SAMD21 Cortex-M0 + 32bit ARM MCU male snage
- 5V napajanje putem USB / VIN-a (!!! Oprez 3.3V logička razina)
- ili
- jednostanični 3,7 V Li-Po (min 700mAh)
- 12 PWM pinova, 1 UART, 1 SPI, 1 I2C, 7 analognih ulaza, 1 analogni izlaz, 8 vanjskih prekida
- 256KB bljeskalica
- 32KB SRAM

Značajke

- Mikrokontroler SAMD21 Cortex-M0 + 32bit ARM MCU s niskom potrošnjom energije
Ugrađeno napajanje (USB / VIN) Podržane baterije od 5 V (*) Li-Po jednostruka ćelija, 3,7 V, 700mAh Minimalni radni napon u krugu 3.3V Digitalni I / O pinovi 8 PWM pinova 12 (0, 1, 2, 3, 4, 5, 6, 7, 8, 10, A3 - ili 18 -, A4 - ili 19) UART 1 SPI 1 I2C 1 analogni ulazni igle 7 (ADC 8/10/12 bit) analogni Izlazni igle 1 (DAC 10 bitni) Vanjski prekidi 8 (0, 1, 4, 5, 6, 7, 8, A1 ili 16, A2 ili 17) DC struja po I / O pinu 7 mA flash memorije 256 KB SRAM 32 KB EEPROM brzina takta 32.768 kHz (RTC), 48 MHz LED_BUILTIN 6 full-speed USB uređaj i integrirani domaćin LED_BUILTIN 6 duljina 61.5 mm širina 25 mm težina 32 gr

3.2. mTurnigy nano-tech 1500mAh 3S 35~70C Li-Po



:

- Kapacitet: 1500mAh
- Napon: 3S1P / 3 Cell / 11.1V
- Težina: 129g (uključuje žice, utičnicu i kućište)
- Dimenzije: 88mm*34mm*21mm

3.3. MKR2 UNO adapter



Ovaj adapter omogućuje napajanje vaše MKR ploče pomoću cijevnog priključka s središnjim pozitivnim polaritetom ili VIN pina na zaglavljima te korištenje sklopova prilagođenih za platformu Arduino UNO.

3.4. 5V 3A UBEC

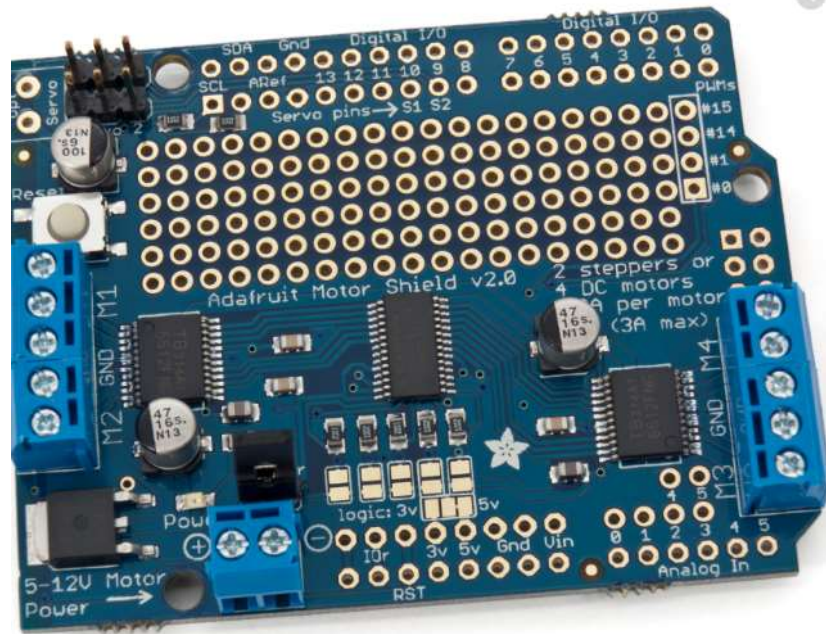


5V 3A UBEC podrška 2-6S Li-Po baterija potpuno zaštićen stabilizator napona protiv smetnji. UBEC je napredni sklopni regulator s zaštitom od strujnog preopterećenja i pregrijavanja, a maksimalna učinkovitost sustava je gotovo 90%.

Specifikacije:

- Izlaz: 5V / 3A ili 6V / 3A (odabrano preko kratkospojnika)
- Ulaz: 5,5-26V (2-6s lipo baterija ili 5-15s NiCd / NiMH baterija)
- Dimenzija: 43mm * 17mm * 7mm
- Težina: 11g
- Val: manje od 50mVp-p (@ 2A / 12V)

3.5. Adafruit Motor Shield V2



Ovaj Adafruit motor Shield za Arduino najnovija je verzija motornog štita za Arduino iz tvrtke Adafruit. Ako je ovaj štiti zadržao sposobnost pogona do 4 istosmjerna motora ili 2 koračna motora, Adafruit je dodao mnoga poboljšanja:

Ovaj komplet Adafruit Motor Shield za Arduino v2 koristi TB6612 MOSFET upravljački program s 1,2 A po kanalu i 3A vršnom strujom.

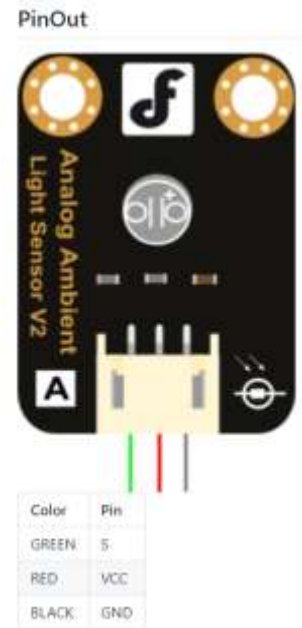
Ovaj Adafruit Motor Shield za Arduino v2 Kit ima potpuno posvećeni PWM čip upravljačkog programa koji upravlja svim motorima i kontrolama brzine umjesto da koristi Arduino PWM pinove.

Samo dva pina (SDA i SCL) potrebna su za pogon više motora, a s obzirom na to da je to I2C, na iste ige možete također povezati bilo koji drugi I2C uređaj ili štiti.

Tehničke specifikacije Adafruit Motor Shield-a za Arduino v2 Kit:

- 2 priključka za 5V "hobi" servo uređaje povezane s Arduinovim namještenim timerom visoke razlučivosti
- 4 H-mostova: TB6612 čipset pruža 1,2A po mostu (vrh 3A) s toplinskom zaštitom, internim diodama za zaštitu od povratnog udara. Može pokretati motore na 4,5 VDC do 13,5 VDC
- Do 4 dvosmjerna istosmjerna motora s pojedinačnim 8-bitnim odabirom brzine (razlučivost oko 0,5%)
- Do 2 koračna motora (jednopolarni ili bipolarni) s jednostrukom zavojnicom, dvostrukom zavojnicom, isprepletenom ili mikrostepenom
- Veliki priključci terminalnih blokova za jednostavno spajanje žica (18-26AWG) i napajanja
- Gumb za resetiranje Arduina podignut je na vrhu
- 2-pinska stezaljka i kratkospojnik zaštićeni polaritetom za spajanje vanjskog napajanja, za odvojeno napajanje logike / motora

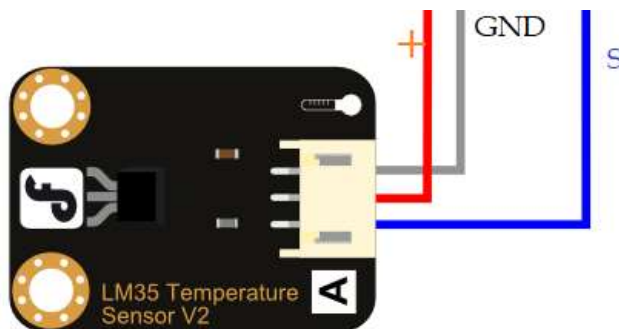
3.6. Gravity: Analog Ambient Light Sensor For Arduino



SPECIFICATION

Supply Voltage: 3.3V to 5V
Illumination range : 1 Lux to 6000 Lux
Responsive time : 15us
Interface: Analog
Size:22x30mm(0.87"x1.18")

3.7. DFRobot LM35 Linear Temperature Sensor



Specification

Sensor Chip Model: LM35

Supply Voltage: 3.3~5.0V

Sensor Chip Sensitivity: 10mV/°C

Measuring Range: 0~150°C

Precision: ±1°C

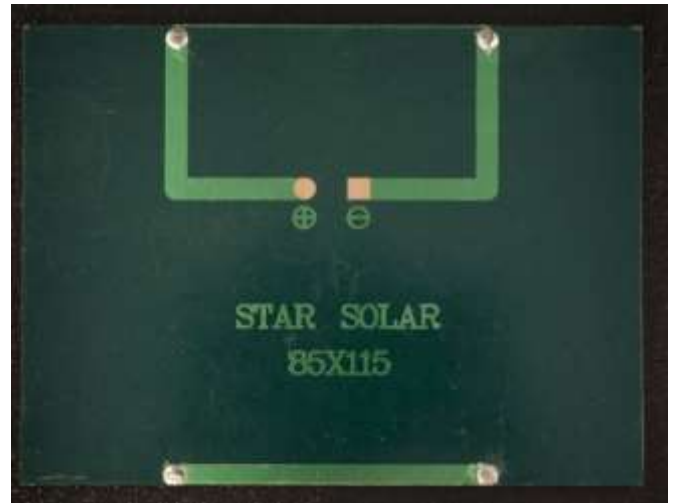
Connector Type: PH2.0-3P

Dimension: 30*22 mm

3.8. Solarni panel 1.5W 12V i solarni panel 1 W 6V

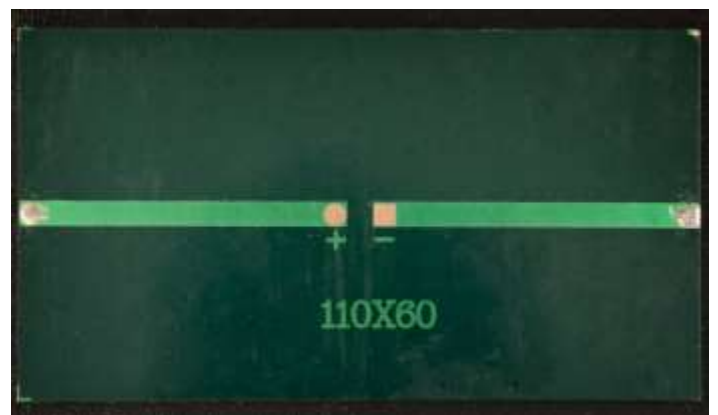


• Dimenzije: 115 x 85 x 3 mm

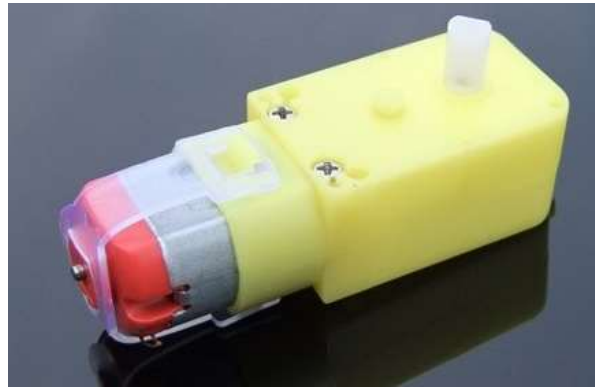


• Napon otvorenog kruga: 6.0V

• Dimenzije: 110 x 60 x 3 mm

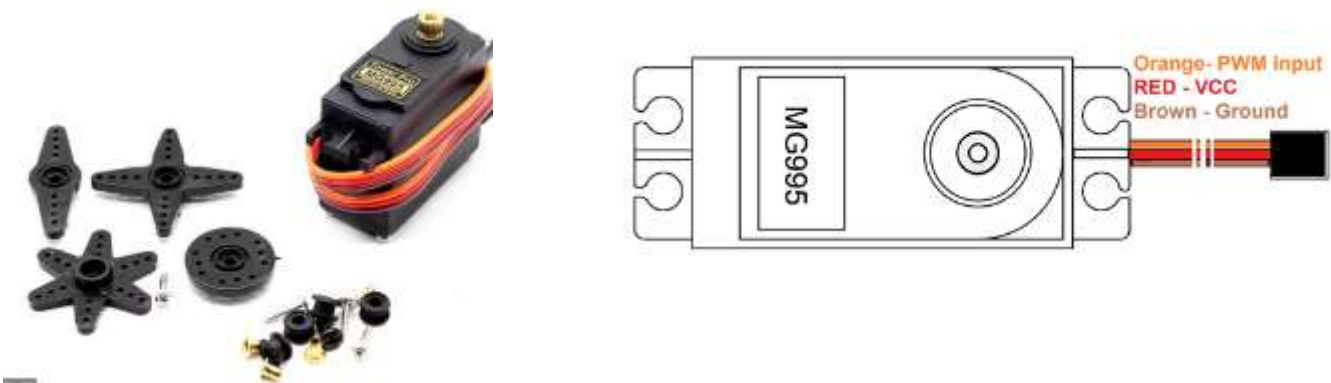


3.9. DC motor za zupčanikom



DC motor za zupčanikom ima napajanje od 3V – 6V DC. Omjer redukcije brzine vrtnje mu je 8. Maksimalni obrtni moment mu je 800k/cm. Brzina pri 6V naponu (bez tereta) je 200 okr/min . Struja (6V) (bez tereta) je 200mA. Brzina pri 3V (bez tereta) je 90 okr/min. Struja (3V) (bez tereta) je 50mA. Napajanje je od 3V – 6V.

3.10. Servo Motor MG995

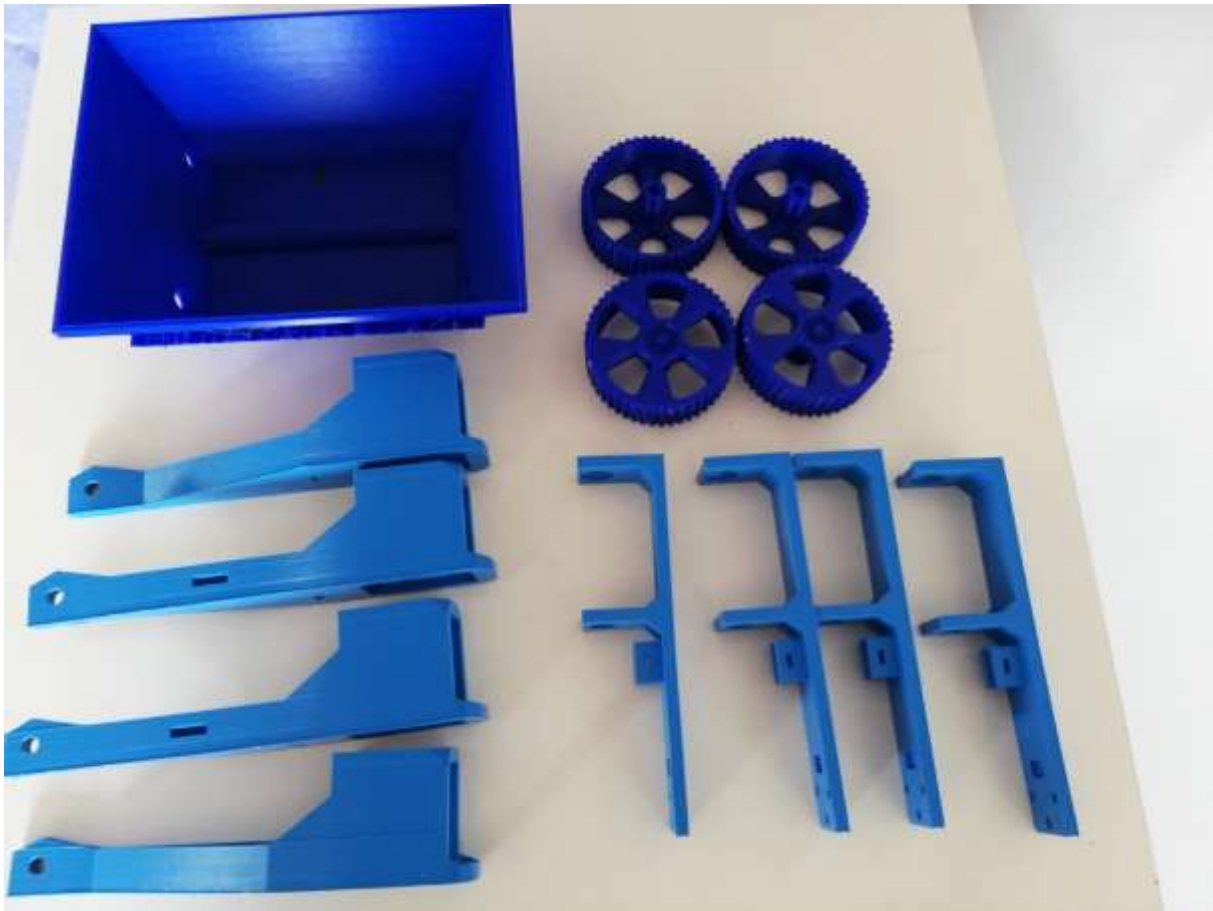


MG995 Značajke i električne karakteristike

- Servo sa metalnim zupčanicima za dulji vijek trajanja
- Stabilan dvostruki kuglični ležaj otporan na udarce
- Velika brzina rotacije za brzi odgovor
- Brzi odgovor kontrole
- Stalni zakretni moment u cijelom rasponu servo putovanja
- Izvrsna snaga držanja
- Težina: 55 g
- Dimenzija: 40,7 × 19,7 × 42,9mm
- Raspon radnog napona: 4,8 V do 7,2 V
- Moment zaustavljanja: 9,4 kg / cm (4,8 v); 11 kg / cm (6 v)
- Radna brzina: 0,2 s / 60° (4,8 V), 0,16 s / 60° (6 V)
- Stupanj rotacije: 180°
- Širina mrtvog pojasa: 5 μs
- Raspon radne temperature: 0°C do + 55°C
- Strujanje u praznom hodu: 10mA
- Punjenje radne struje bez opterećenja: 170mA
- Struja pri maksimalnom opterećenju: 1200mA

3.11. Mehaničke komponente

Kućište, kotači i noge za kretanje

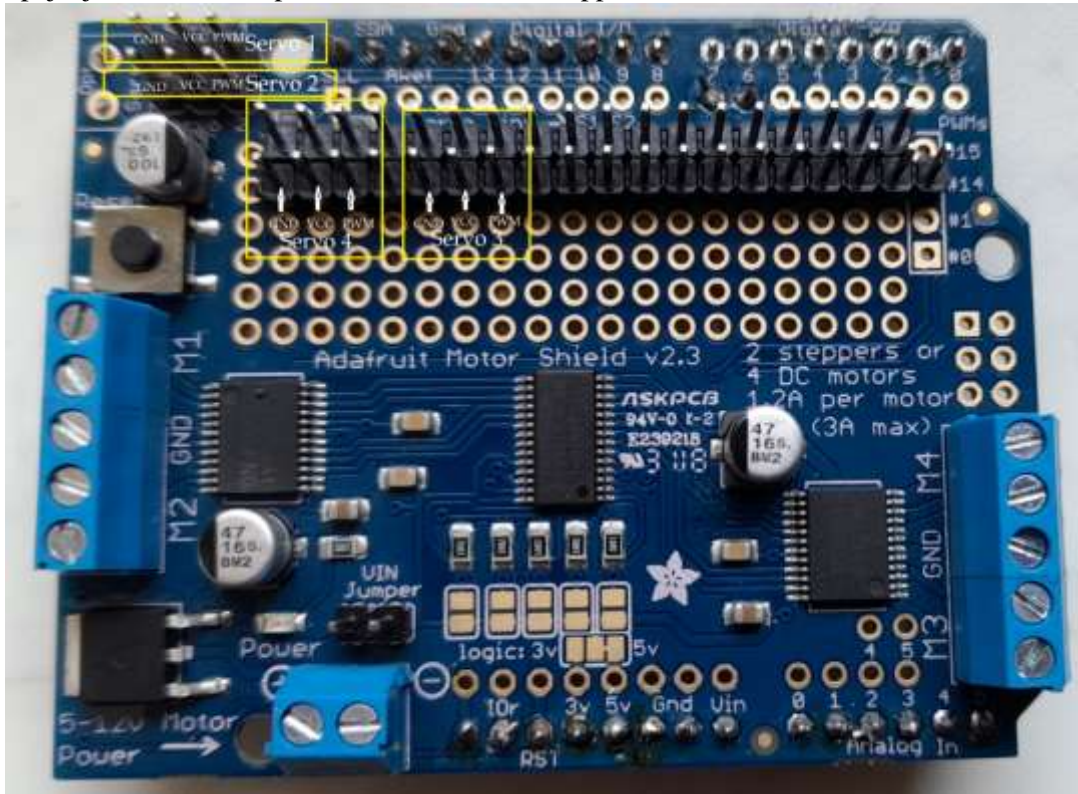


Amortizeri

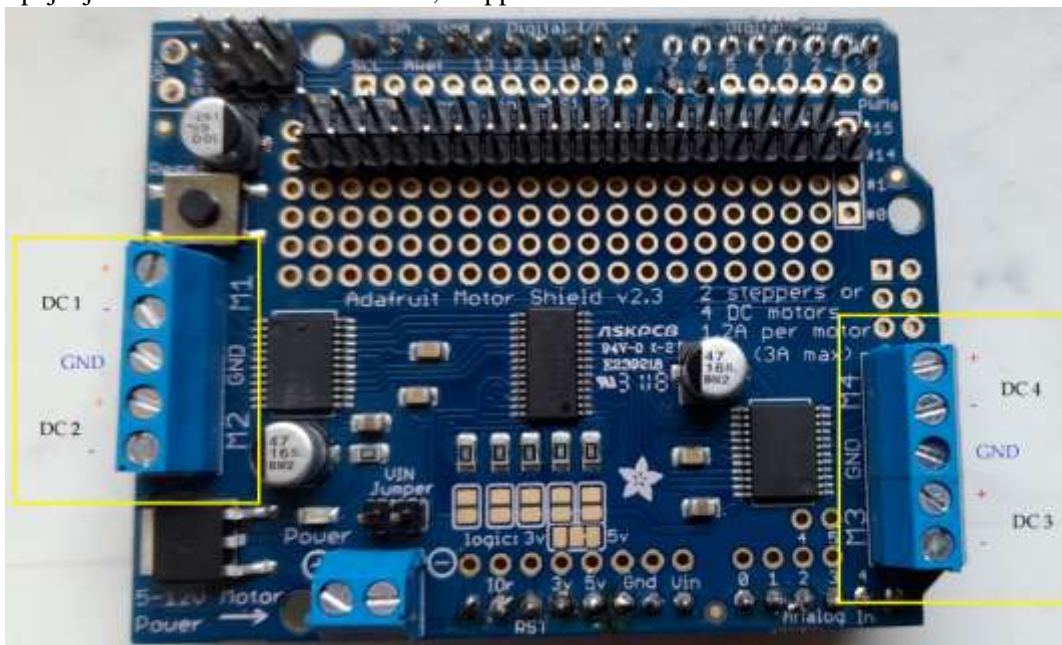


4. SHEMA SPAJANJA

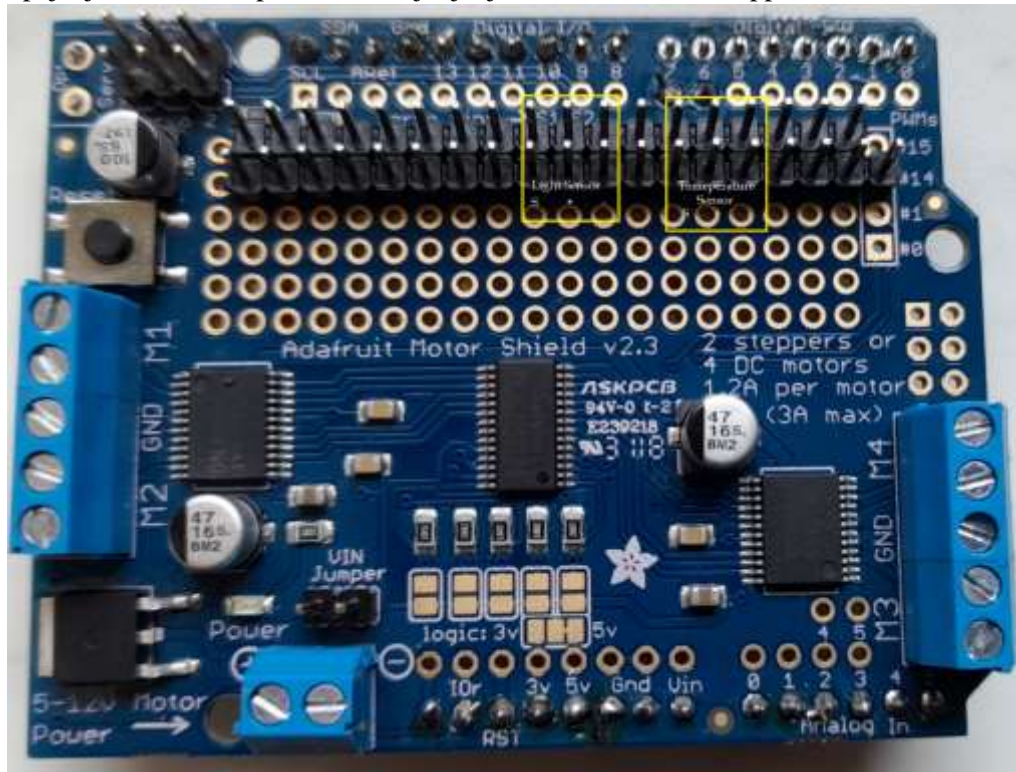
Spajanje servo motora preko Adafruit Motor, Stepper & Servo Shielda



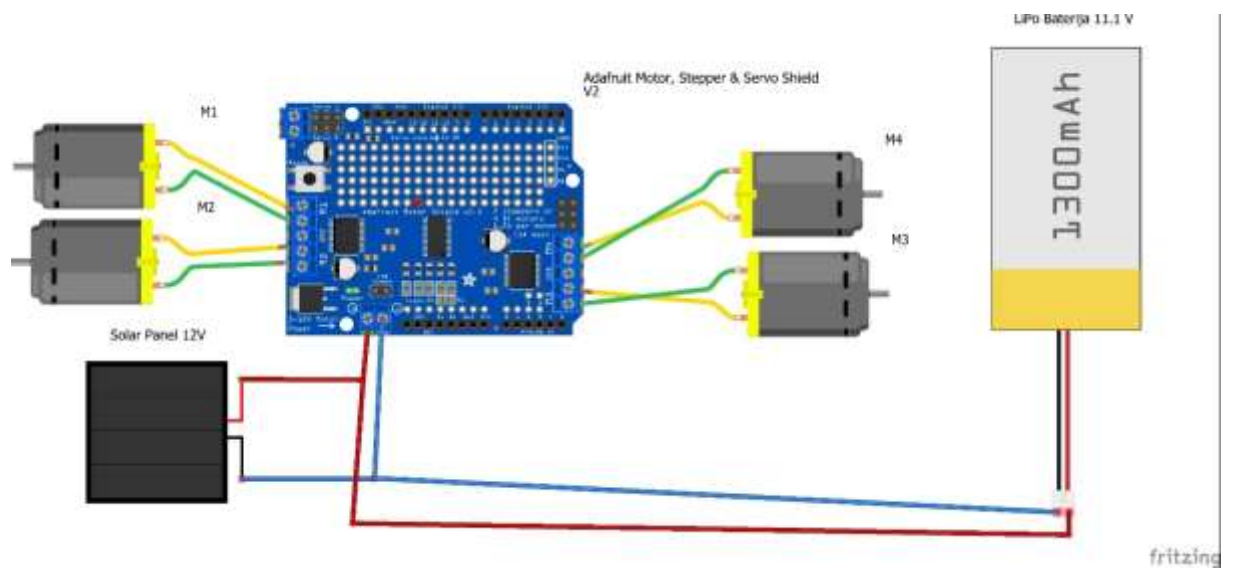
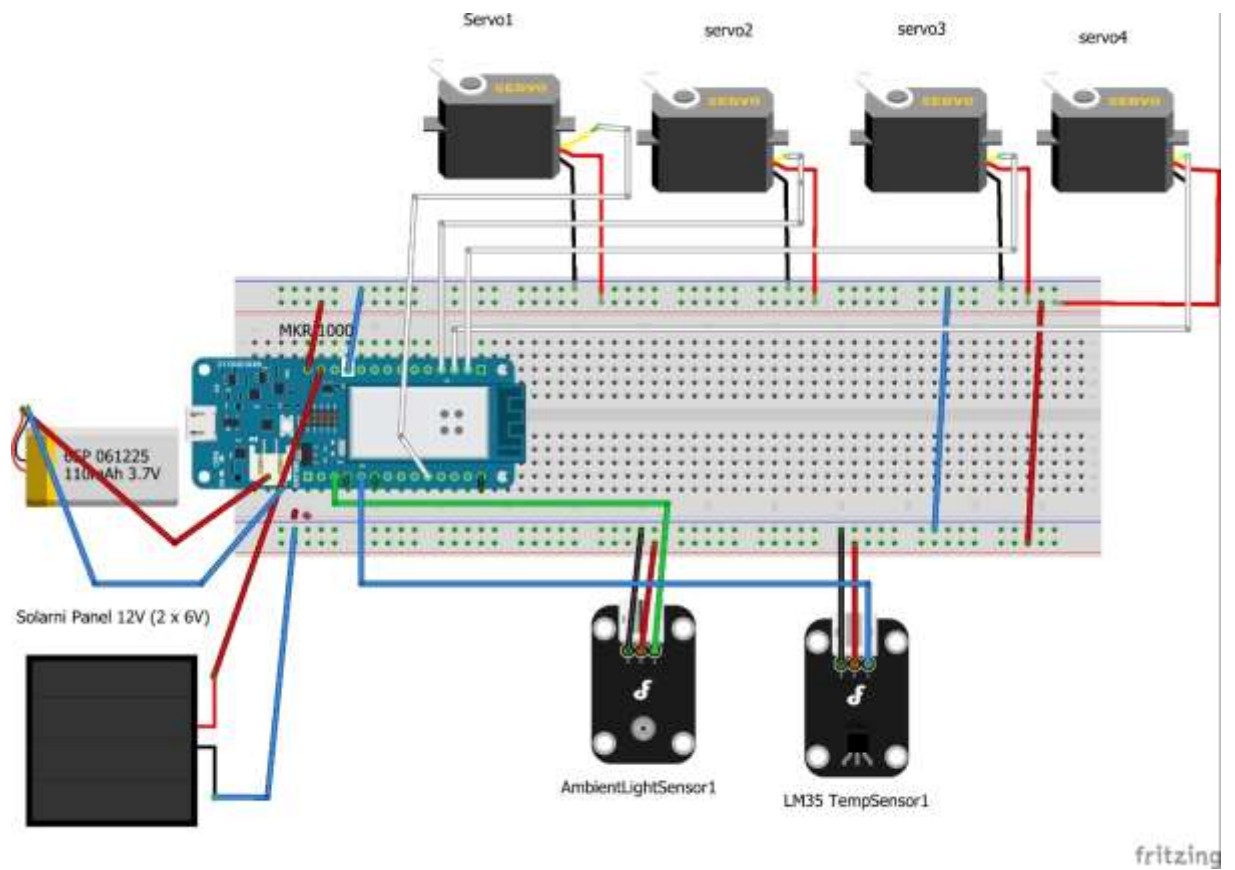
Spajanje DC motora Adafruit Motor, Stepper & Servo Shielda



Spajanje senzora temperature i osvijetljenja Adafruit Motor, Stepper & Servo Shielda



Fritzing sheme spjanja



5. ARDUINO KOD

```
//STRUKOVNA ŠKOLA ĐURĐEVAC
//SEKCIJA ZA SVEMIRSKU TEHNOLOGIJU ĐURĐEVAC
//PROJEKT "RASCO MARS ROVER"

// INICIJALIZACIJA
#define BLYNK_PRINT SerialUSB

#include <SPI.h>
#include <WiFi101.h>
#include <BlynkSimpleWiFiShield101.h>
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"
#include <Servo.h>

//podaci Blynk aplikacije
char auth[] = "112222nIFkPTvWVM-ln8FKD_ot_j1_-t"; // Autorizacijski Token za Blynk
aplikaciju "Rasco Mars Rover"
// WiFi pristupni podaci.
// xxxx
//char ssid[] = "XXXXXX";
//char pass[] = "XXXXXXXX";
// GSM router
char ssid[] = "ZZZZZZZ";
char pass[] = "zzzzzzzz";

//Adafruit Motor Shield - postavke kartice -pokretač istosmjernih motora -spajanje DC motora
Adafruit_MotorShield AFMS = Adafruit_MotorShield(); // Kriranje objekata za povezivanje
motora sa predodređenom I2C adresom
Adafruit_DCMotor *motor1 = AFMS.getMotor(1); // Prvi DC motor - priključak M1
Adafruit_DCMotor *motor2 = AFMS.getMotor(2); // Drugi DC motor - priključak M2
Adafruit_DCMotor *motor3 = AFMS.getMotor(3); // Treći DC motor - priključak M3
Adafruit_DCMotor *motor4 = AFMS.getMotor(4); // Četvrti DC motor - priključak M4

//Servo motori
Servo servo1; // kreiranje servo objekta za upravljanje servo motorom 1
Servo servo2; // kreiranje servo objekta za upravljanje servo motorom 2
Servo servo3; // kreiranje servo objekta za upravljanje servo motorom 3
Servo servo4; // kreiranje servo objekta za upravljanje servo motorom 4

int pos1 = 0; // kreiranje varijable za pohranu pozicije servo motora 1
int pos2 = 0; // kreiranje varijable za pohranu pozicije servo motora 2
int pos3 = 0; // kreiranje varijable za pohranu pozicije servo motora 3
int pos4 = 0; // kreiranje varijable za pohranu pozicije servo motora 4

int vSmjer = 1; // varijabla smjera kretanja -početna postavka - naprijed
int vZakretanje = 0; // varijabla konfiguracije-početna postavka - isključeno
```

```

int vRotiranje = 0; // varijabla okretanja-početna postavka - isključeno
int vBrzina = 0; // varijabla brzine kretanja-početna postavka - brzina 0 - zaustavljeni motori
int vReset= 0; // varijabla resetiranja-pozicije servo motora na početne pozicije, zaustavljanje dc motora
float temp = 0.0 ; //varijabla za očitavanje temperature sa DFRobot LM35 Linear Temperaturnog Sensora
float vTemp = 0.0 ; //varijabla za pohranu temperature u stupnjevima Celsijusa
int vSvjetlo = 0; // varijabla za jačinu svjetla u okolini (Lux)

##### POSTAVKE #####
void setup() {
  Serial.begin(9600);
  delay(1000);
  // Debug console
  SerialUSB.begin(9600);

  Blynk.begin(auth, ssid, pass);

  //Adafruit Motor Shield - postavke za pokretač istosmjernih motora

  AFMS.begin(); // inicijalizacija pokretača istosmjernih motora - početna frekvencija PWM signala - 1.6 KHz
  motor1->setSpeed(255); // postavke brzine vrtnje za DC motor 1 (od 0 (isključeno) do 255 (maksimalna brzina))
  motor2->setSpeed(255); // postavke brzine vrtnje za DC motor 2 (od 0 (isključeno) do 255 (maksimalna brzina))
  motor3->setSpeed(255); // postavke brzine vrtnje za DC motor 3 (od 0 (isključeno) do 255 (maksimalna brzina))
  motor4->setSpeed(255); // postavke brzine vrtnje za DC motor 4 (od 0 (isključeno) do 255 (maksimalna brzina))

  //Postavke za servo motore
  servo1.attach(1); // poveži servo objekt 1 na digitalni pin 1 Arduino mikrokontrolera
  servo2.attach(9); // poveži servo objekt 2 na digitalni pin 9 Arduino mikrokontrolera
  servo3.attach(7); // poveži servo objekt 3 na digitalni pin 7 Arduino mikrokontrolera
  servo4.attach(8); // poveži servo objekt 4 na digitalni pin 8 Arduino mikrokontrolera
}

##### LOOP petlja
void loop() {
  Blynk.run();
  temp = analogRead(A3); //Senzor temperature spojen na A3
  vSvjetlo = analogRead(A1); //Senzor osvjetljenja spojen na A1
  vTemp = ((float) temp * (5/1023.0))*100;
  Serial.print("Temp:"); //Prikaži temperaturu u serijskom monitoru
  Serial.print(vTemp);
  Serial.println(" C");
}

```

```

Serial.print("Svjetlost:"); //Prikaži osvjetljenost u serijskom monitoru
Serial.print(vSvjetlo);
Serial.println(" Lux");
Blynk.virtualWrite(6, vTemp); // pošalji temperaturu u Blynk
Blynk.virtualWrite(7, vSvjetlo); // pošalji nivo svjetlosti u Blynk
delay(500);
}

##### Podprogrami

// brzina vrtnje DC motora (brzinu kretanje vozila)
BLYNK_WRITE(V0) //Vraćanje stanja virtualnog pina V0 - slider Brzina (postavljanje brzine
kretanja vozila u Blynk aplikaciji)
{
  vBrzina = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V1 varijabli vBrzina (brzina
vrtnje DC motora)
  motor1->setSpeed(vBrzina); // postavljanje brzine vrtnje DC motora 1
  motor2->setSpeed(vBrzina); // postavljanje brzine vrtnje DC motora 1
  motor3->setSpeed(vBrzina); // postavljanje brzine vrtnje DC motora 1
  motor4->setSpeed(vBrzina); // postavljanje brzine vrtnje DC motora 1

  //***** Ispis trenutno postavljenih vrijednosti virtualnih pinova na ekranu serijskog monitora
  Serial.print("Očitanje V0 -vBrzina: ");
  Serial.print(vBrzina);
  Serial.print(" vSmjer: ");
  Serial.print(vSmjer);
  Serial.print(" vRotiranje: ");
  Serial.print(vRotiranje);
  Serial.print(" vZakretanje: ");
  Serial.println(vZakretanje);

  stanjaMotora(); // poziv funkcije stanjaMotora(), Funkcija postavlja stanja motora
}

// smjer (smjer vrtnje DC motora - smjer kretanje vozila)
BLYNK_WRITE(V1)//Vraćanje stanja virtualnog pina V1 - gumb Smjer-uključivanje promjene
smjera kretanja vozila u Blynk aplikaciji
{

  int pinValue = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V1 varijabli pinValue
  if(pinValue == 1) {
    vSmjer = 1; // smjer kretanja naprijed
  }
  if(pinValue == 0) {
    vSmjer = 0; // smjer kretanja natrag
  }
  stanjaMotora(); // poziv funkcije stanjaMotora(), Funkcija postavlja stanja motora
  // Ispis vrijednosti smjera kretanja u serijskom monitoru

```

```
Serial.print(" vSmjer: ");  
Serial.println(vSmjer);  
}
```

```
// ZAKRETANJE
```

```
BLYNK_WRITE(V2)//Vraćanje stanja virtualnog pina V2 -gumb ZAKRETANJE vozila u  
Blynk aplikaciji  
{  
  int pinValue = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V2 varijabli pinValue  
  if(pinValue == 0) {  
    vZakretanje = 0;  
    servoPosNormal(); //postavljanje normalne pozicije servo motora  
  }  
  if(pinValue == 1) {  
    vZakretanje = 1;  
    servoPosZakretanje(); //postavljanje zakretanja (za 90 stupnjeva) pozicije servo motora  
  }  
  stanjaMotora(); // poziv funkcije stanjaMotora(), Funkcija postavlja stanja motora  
  Serial.print(" vZakretanje: ");  
  Serial.println(vZakretanje);  
}
```

```
// Promjena smjera kretanja-ROTIRANJE
```

```
BLYNK_WRITE(V3)//Vraćanje stanja virtualnog pina V3 - gumb ROTIRANJE-uključivanje  
rotiranja( vrtnja u krug) vozila u Blynk aplikaciji  
{  
  int pinValue = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V2 varijabli pinValue  
  if(pinValue == 1) {  
    vRotiranje = 1;  
    servoPosRotiranje(); //postavljanje pozicije servo motora za rotiranje  
  }  
  if(pinValue == 0) {  
    if(vZakretanje == 0) {  
      vRotiranje = 0;  
      servoPosNormal(); //postavljanje normalne pozicije servo motora  
    }  
    else{  
      vRotiranje = 0;  
      servoPosZakretanje(); //postavljanje pozicije servo motora za rotiranje  
    }  
  }  
  stanjaMotora(); // poziv funkcije stanjaMotora(), Funkcija postavlja stanja motora  
  Serial.print(" vRotiranje: ");  
  Serial.println(vRotiranje);  
}
```

```
// Zaustavljanje
```

```
BLYNK_WRITE(V4) //Vraćanje stanja virtualnog pina V4 gumb STOP - trenutno zaustavljanje kretanja vozila u Blynk aplikaciji
```

```
{  
  int vStop = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V2 varijabli vStop  
  
  if(vStop == 1) {  
    motor1->run(RELEASE); // Zaustavi motor 1 (isključivanje napajanja motora)  
    motor2->run(RELEASE); // Zaustavi motor 2 (isključivanje napajanja motora)  
    motor3->run(RELEASE); // Zaustavi motor 3 (isključivanje napajanja motora)  
    motor4->run(RELEASE); // Zaustavi motor 4 (isključivanje napajanja motora)  
  }  
}
```

```
// postavljanje na početne parametre kretanja i pozicije
```

```
BLYNK_WRITE(V5) //Vraćanje stanja virtualnog pina V4 gumb STOP - trenutno zaustavljanje kretanja vozila u Blynk aplikaciji
```

```
{  
  int vReset = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V2 varijabli vStop  
  
  if(vReset == 1) {  
    motor1->run(RELEASE); // Zaustavi motor 1 (isključivanje napajanja motora)  
    motor2->run(RELEASE); // Zaustavi motor 2 (isključivanje napajanja motora)  
    motor3->run(RELEASE); // Zaustavi motor 3 (isključivanje napajanja motora)  
    motor4->run(RELEASE); // Zaustavi motor 4 (isključivanje napajanja motora)  
    vSmjer = 1; // varijabla smjera kretanja -početna postavka - naprijed  
    vZakretanje = 0; // varijabla konfiguracije-početna postavka - isključeno  
    vRotiranje = 0; // varijabla okretanja-početna postavka - isključeno  
    vBrzina = 0; // varijabla brzine kretanja-početna postavka - brzina 0 - zaustavljeni motori  
    stanjaMotora(); // poziv funkcije stanjaMotora(), Funkcija postavlja stanja motora  
    servoPosNormal(); //postavljanje normalne pozicije servo motora  
    Serial.print("Resetiranje ");  
  
  }  
}
```

```
// Funkcija postavlja stanja motora
```

```
void stanjaMotora() {  
  //(BRZINA KRETANJA 0 - ZUSTAVI MOTORE)  
  if(vBrzina == 0) {  
    motor1->run(RELEASE); // Zaustavi motor 1 (isključivanje napajanja motora)  
    motor2->run(RELEASE); // Zaustavi motor 2 (isključivanje napajanja motora)  
    motor3->run(RELEASE); // Zaustavi motor 3 (isključivanje napajanja motora)  
    motor4->run(RELEASE); // Zaustavi motor 4 (isključivanje napajanja motora)  
  }  
  //ROTACIJA  
  else if(vRotiranje == 1) {  
    if(vSmjer == 1) {  
      //NAPRIJED  
      motor1->run(FORWARD); // postavi smjere vrtnje motora 1 na Forward (naprijed)  
      motor2->run(BACKWARD); // postavi smjere vrtnje motora 2 na Backward (natrag)-  
      nasuprotni motor od 1- obrnuti smjer vrtnje
```

```

    motor3->run(BACKWARD); // postavi smjere vrtnje motora 3 na Forward (naprijed)
    motor4->run(FORWARD); // postavi smjere vrtnje motora 4 na Backward (natrag)-
nasuprotni motor od 3- obrnuti smjer vrtnje
    Serial.println("ROTACIJA-NAPRIJED");
}
if(vSmjer == 0) {
    //NATRAG
    motor1->run(BACKWARD); // postavi smjere kretanja motora 1 na Backward (natrag))
    motor2->run(FORWARD); //postavi smjere vrtnje motora 2 na Forward (naprijed)
    motor3->run(FORWARD); // postavi smjere kretanja motora 3 na Backward (natrag))
    motor4->run(BACKWARD); //postavi smjere vrtnje motora 4 na Forward (naprijed)
    Serial.println("ROTACIJA-NATRAG");
}
}
// ZAKRETANJE
else if(vZakretanje == 1) {

    if(vSmjer == 1) {
        // NAPRIJED
        motor1->run(FORWARD);
        motor2->run(FORWARD);
        motor3->run(BACKWARD);
        motor4->run(BACKWARD);
        Serial.println("ZAKRETANJE-NAPRIJED");
    }

    if(vSmjer == 0) {
        //NATRAG
        motor1->run(BACKWARD);
        motor2->run(BACKWARD);
        motor3->run(FORWARD);
        motor4->run(FORWARD);
        Serial.println("ZAKRETANJE-NATRAG");
    }
}
// NORMALNO KRETANJE
else if(vZakretanje == 0 && vRotiranje == 0) {
    if(vSmjer == 1) {
        // NAPRIJED
        motor1->run(FORWARD);
        motor2->run(BACKWARD);
        motor3->run(FORWARD);
        motor4->run(BACKWARD);
        Serial.println("NORMALNO-NAPRIJED");
    }

    if(vSmjer == 0) {
        //NATRAG
        motor1->run(BACKWARD);
        motor2->run(FORWARD);
        motor3->run(BACKWARD);
    }
}

```

```

        motor4->run(FORWARD);
        Serial.println("NORMALNO-NATRAG");
    }
}

void servoPosNormal() {
    servo1.write(90);
    servo2.write(90);
    servo3.write(90);
    servo4.write(90);
    Serial.println("SERVO NORMAL");
}

void servoPosZakretanje() {
    servo1.write(0);
    servo2.write(180);
    servo3.write(172);
    servo4.write(0);
    Serial.println("SERVO ZAKRETANJE");
}

void servoPosRotiranje() {
    servo1.write(65);
    servo2.write(115);
    servo3.write(115);
    servo4.write(65);
    Serial.println("SERVO ROTIRANJE");
}

// Pojedinačno podešavanje položaja servo motora
// Servo1
BLYNK_WRITE(V10) // M1 slider
{
    int pinValue = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V10 varijabli pinValue
    servo1.write(pinValue);
    Serial.println("SERVO1 POZICIJA"+pinValue);
}

// Servo2
BLYNK_WRITE(V11) //M2 slider
{
    int pinValue = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V11 varijabli pinValue
    servo2.write(pinValue);
    Serial.println("SERVO2 POZICIJA"+pinValue);
}

// Servo3
BLYNK_WRITE(V12) //M3 slider
{
    int pinValue = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V12 varijabli pinValue
    servo3.write(pinValue);
    Serial.println("SERVO3 POZICIJA"+pinValue);
}

```

```
}  
  
// Servo4  
BLYNK_WRITE(V13) // Slider  
{  
  int pinValue = param.asInt(); // dodjeljivanje ulazne vrijednosti s pina V13 varijabli pinValue  
  servo4.write(pinValue);  
  Serial.println("SERVO4 POZICIJA"+pinValue);  
}
```

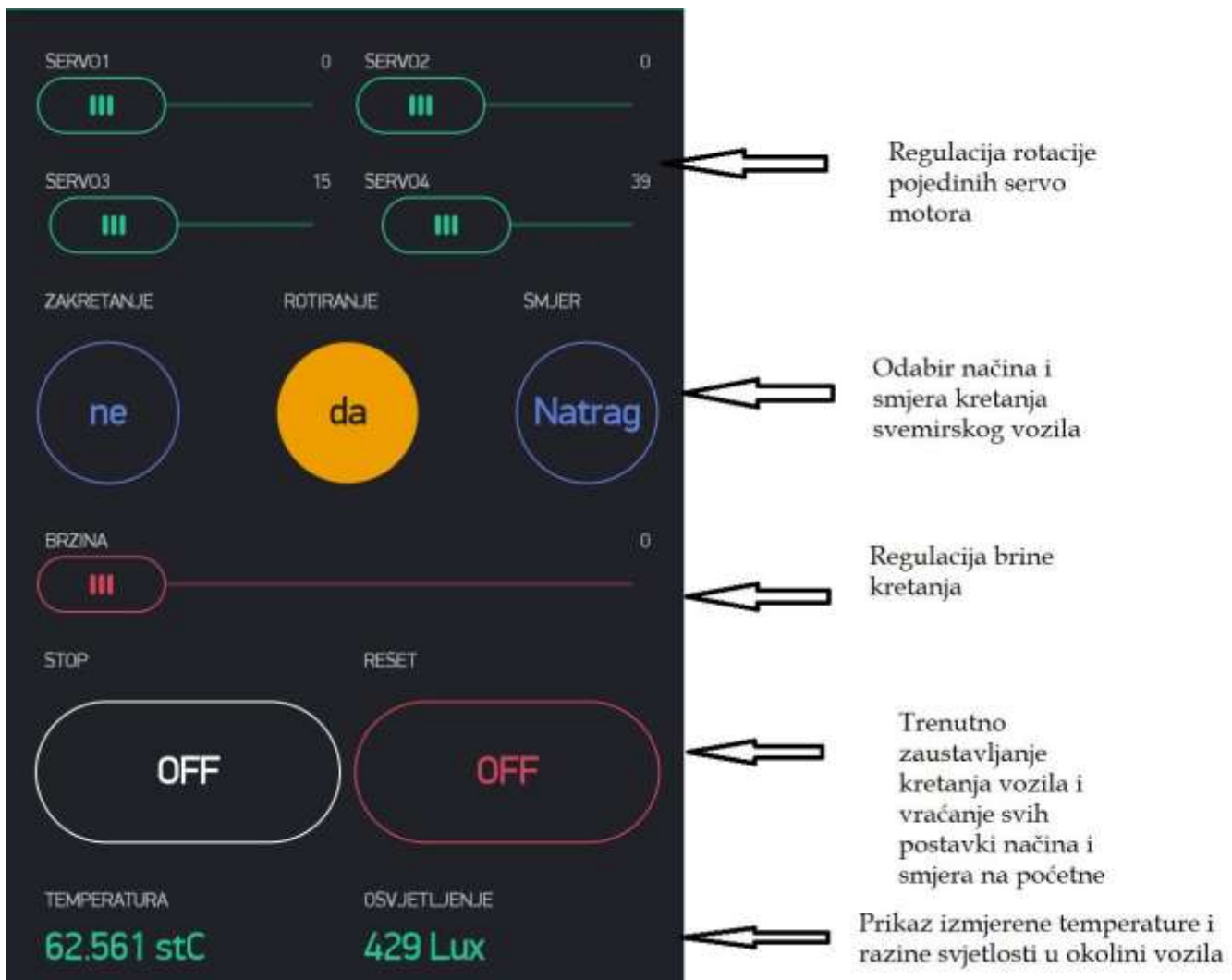

6. BLYNK APLIKACIJA ZA DALJINSKO UPRAVLJANJE SVEMIRSKIM VOZILOM

Blynk aplikacija povezana je bežičnom vezom sa svemirskim vozilom a prijenos podataka između aplikacije i Arduino mikrokontrolera MKR 1000 odvija se preko virtualnih pinova V0, V1, V2, V3, V4, V5, V6, V7, V10, V11, V12 I V13.

Popis Widgeta aplikacije:

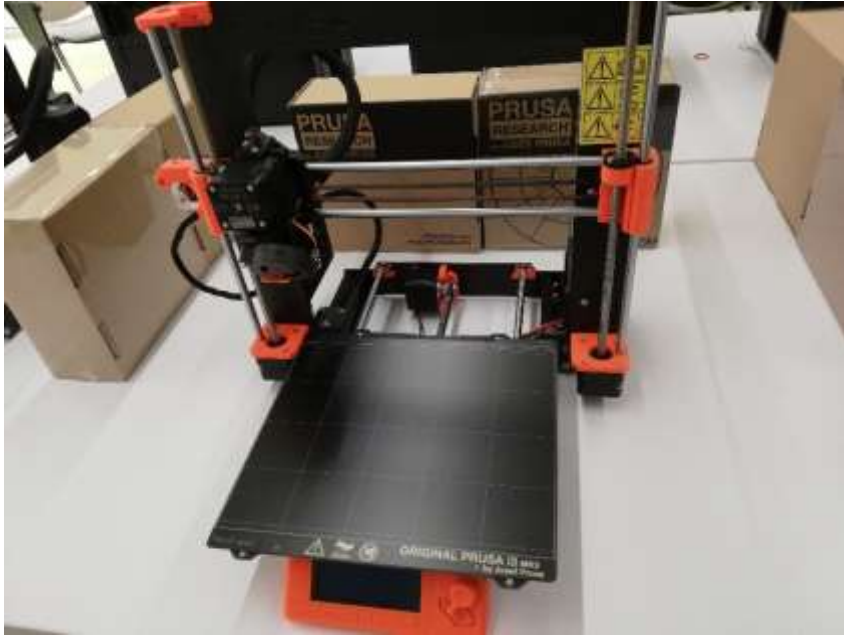
- Regulacija brzine – V0 – Slider BRZINA
- Smjer kretanja (Naprijed-Natrag) – V1- Button SMJER
- Promjera smjera kretanja za 90 stupnjeva –V2- Button ZAKRETANJE
- Promjena smjera kretanja (Pravocrtno-Rotacija) – V3- Button ROTACIJA
- Trenutno zaustavljanje kretanja-V4-Button STOP
- Trenutno zaustavljanje kretanja i vraćanje svim postavki (smjer kretanja, brzina, zakretanje, rotacija) na početne postavke – V5- Button RESET
- Promjena kuta zakretanja servo motora 1 – V10- Slider SERVO1
- Promjena kuta zakretanja servo motora 2 – V11- Slider SERVO2
- Promjena kuta zakretanja servo motora 3 – V12- Slider SERVO3
- Promjena kuta zakretanja servo motora 4 – V13- Slider SERVO4

Snimka ekrana mobilne Blynk aplikacije

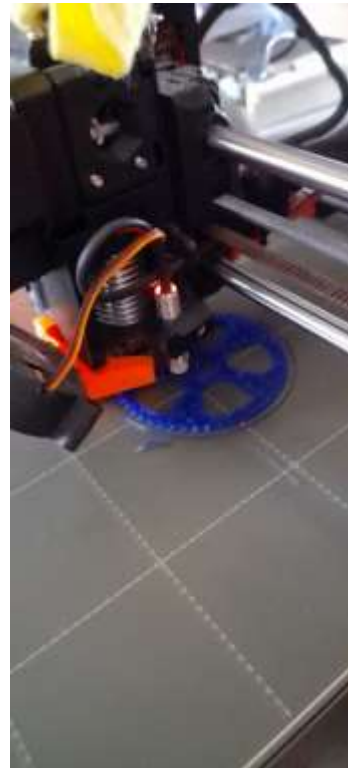
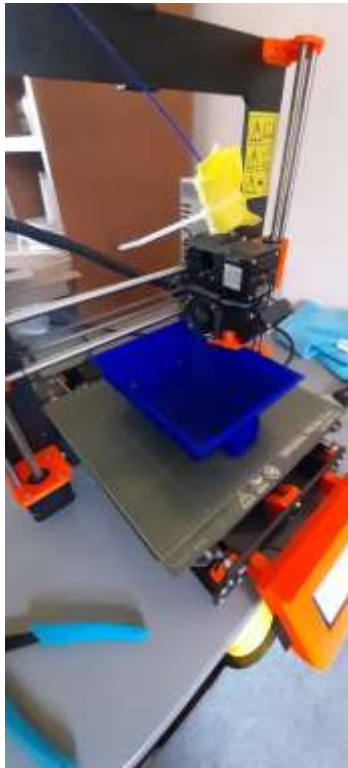


7. FOTOGRAFIJE PROVEDBE PROJEKTA IZRADE SVEMIRSKOG VOZILA U OKVIRU DONACIJSKOG NATJEČAJA *Junior Engineer Academy*

Početak projekta, preuzimanje donirane opreme i osnovna edukacija za mentora održala se je u Zagrebačkom inovacijskom centru, Avenija Dubrovnik 15, Zagrebački Velesajam, Paviljon 12 , 25.3. i 26.3. 2019.



Ispis dijelova na 3d pisaču



Sastavljanje vozila



8. ODRŽAVANJE

Sve elemente i dijelove uređaja potrebno je redovito kontrolirati i održavati. U slučaju oštećenja ili nekog njegovog dijela (kućišta, konstrukcije, podloge ili drugog elementa opreme), popravak se može izvršiti zamjenom oštećenog elementa isključivo istim elementom.