



ili



Jetson nano (autonomija)

- salje komande za motore i busilicu
- od senzora prima podatke

PROTOKOL?

BUSILICA

- elektronika
- kod
- kostur busilice pravi drugi tim

MOTORI

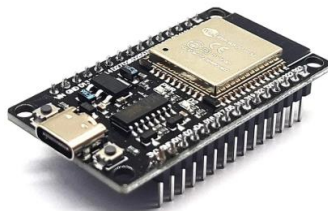
- Biramo mi drajvere i motore zajedno sa drugim timom?
- elektronika
- kod

SENZORI

- elektronika
- kod

Interrupt na high kada ce da udari u nesto, onda senzor mcu obavestava autonomiju

**AKO JE UART PROKOL
GDE BI SAMO SENZOR
MCU SLAO PODATKE**



ESP32 (za testiranje)

- Ima wifi I bluetooth
- Moze website ili aplikacija ili preko kabla za upravljanje
- salje komande za motore i busilicu
- od senzora prima podatke

PROTOKOL?

BUSILICA

- elektronika
- kod
- kostur busilice pravi drugi tim

MOTORI

- elektronika
- kod

SENZORI

- elektronika
- kod

Interrupt na high kada ce
da udari u nesto, onda
senzor mcu obavestava
autonomiju

**AKO JE UART PROKOL
GDE BI SAMO SENZOR
MCU SLAO PODATKE**

ESP32 (za testiranje)

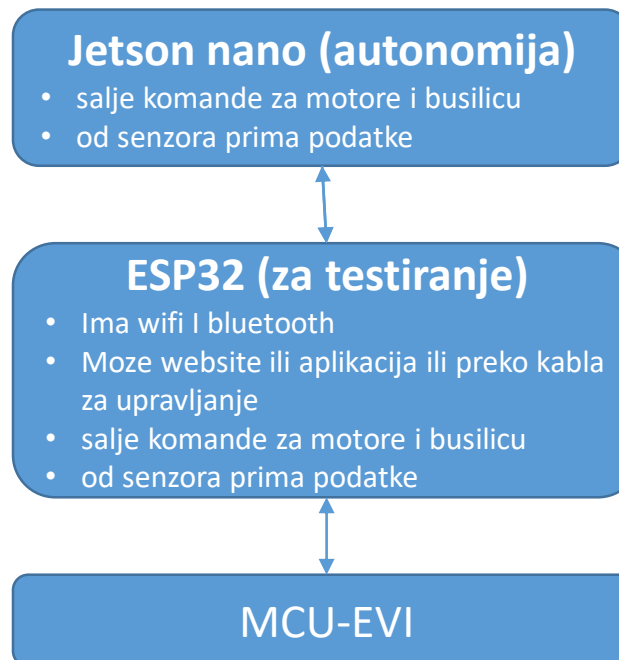
Moze da ostane i za stalno, da on bude izmedju autonomije i mcu-eva.

Imao bi onda 2 moda (koji bi se menjali preko prekidaca na njemu):

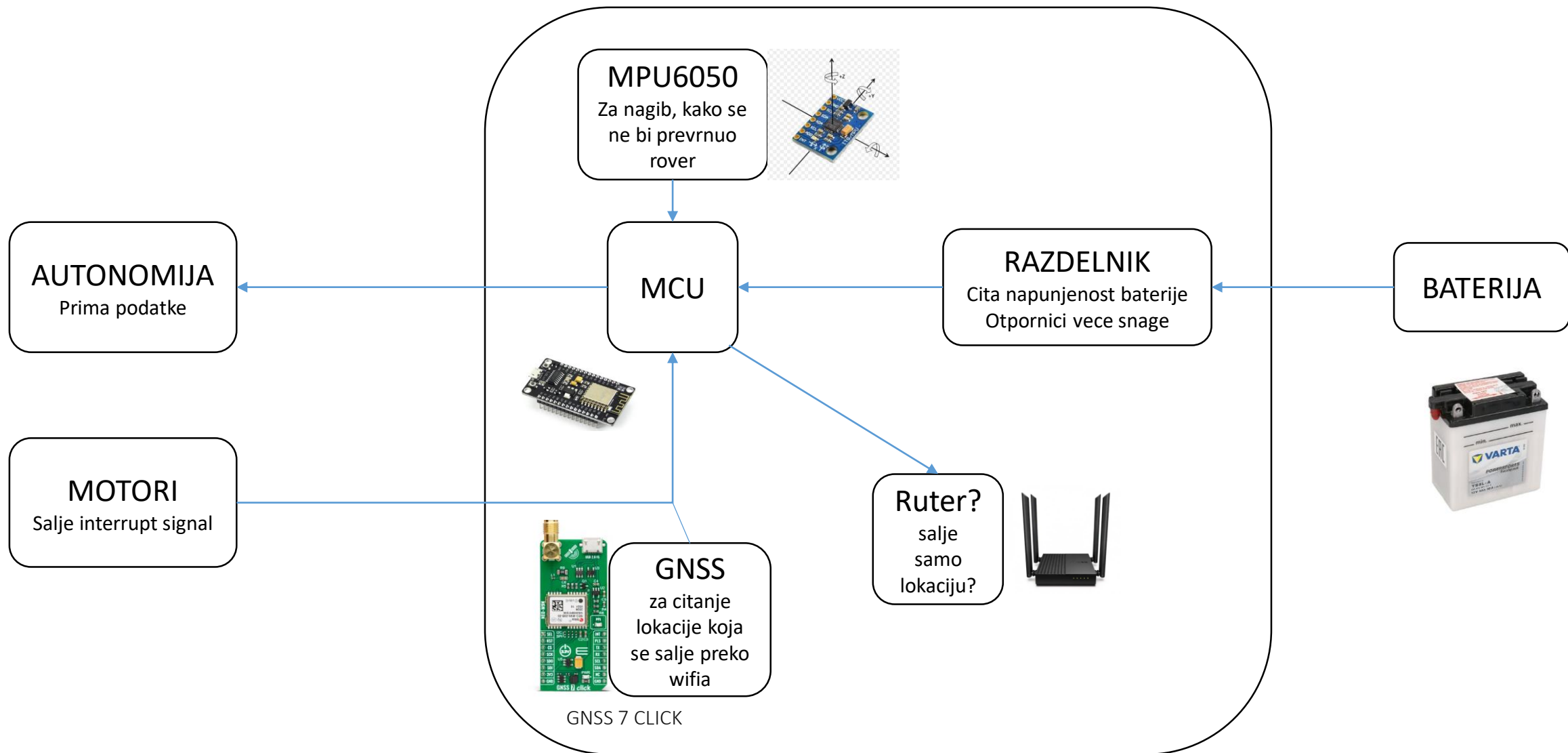
1. Radio bi kao dzojstik za upravljanje rovera (dzojstik moze na vebu, da on bude hotspot ili preko aplikacije)
2. Prosledjivao bi poruke od mcu-a ka autonomiji i obrnuto

U oba moda bi bi imao lcd na kojem bi pisao problem ako nesto ne radi (na primer autonomija ili neki mcu-u ne odgovara ili neki senzor ne radi)
Mozda bi bila i laksa implementacija nekog drugog protokola, ako necemo UART jer sa drugim mozemo da primamo poruke od vise mcu-eva

Ako sve radi onda pise na lcd-u radi sve, u kom je modu i procenat baterije

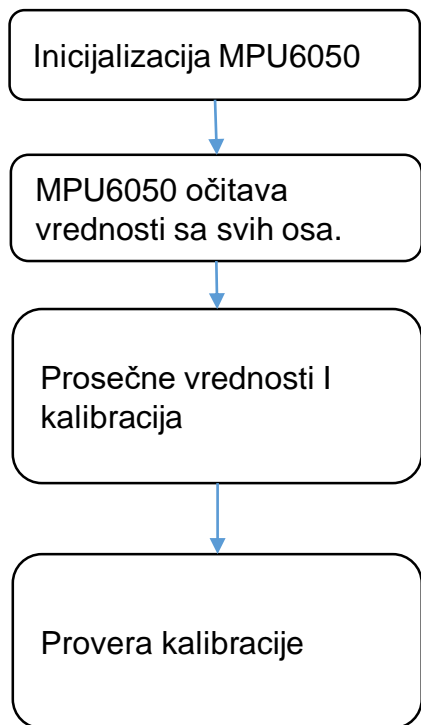


SENZORI



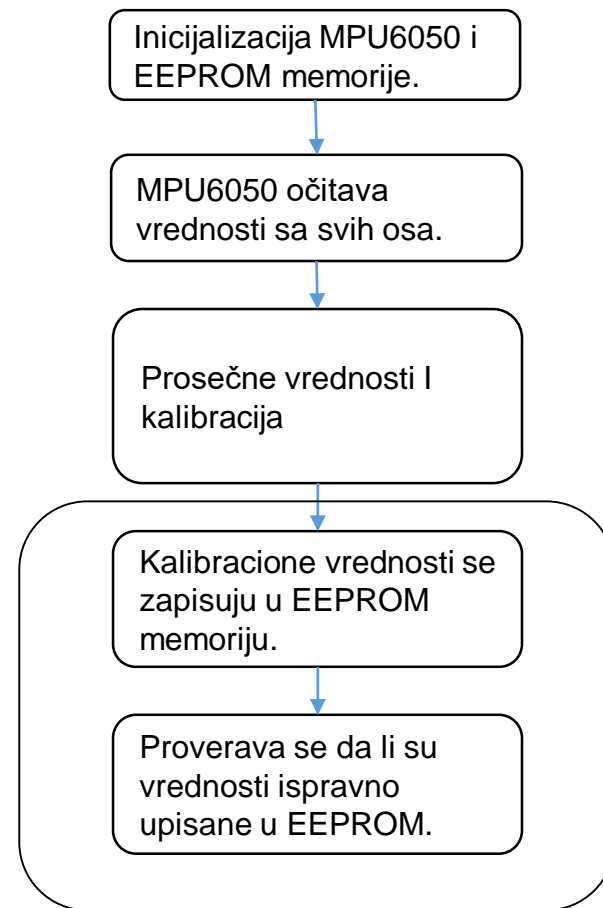
1.Kod za kalibraciju mpu6050

Sve se obradjuje u setup-u



2.Kod za kalibraciju mpu6050

(ne sme u eeprom vise od par stotina puta da se upisuju podaci, zato ne sme u loop)



Dodato cuvanje u eeprom
i provera

Setup:

1. Inicijalizacija svih modula (MPU6050, GNSS, razdelnik, MCU, ruter).
2. Ako neki senzor ne radi upaliti led za njega i poslati gresku na **ESP32**, ako nije bas potreban senzor onda nastaviti sa kodom
1. Inicijalizacija interrupt-a na HIGH za citanje sa MOTOR MCU-a (AKO JE POTREBAN)
2. Učitavanje kalibracije iz EEPROM: Vrednosti kalibracije za MPU6050.
3. Inicijalizacija watchdog timer-a

LOOP:

1. Čitanje senzora:
 - MPU6050 za nagibne podatke
 - Razdelnik meri napon baterije
 - GNSS 7 Click očitava lokaciju
2. Ako neki senzor ne odgovara while(1) ako je potreban senzor, crveni LED i salje podatke autonomiji o gresci
1. Obrada podataka: MCU obrađuje primljene podatke (filtriranje, logika, itd.).
2. Slanje podataka rutera: MCU šalje GNSS podatke preko rutera na zadatu mrežu (preko WiFi)
3. Salju se potrebni podaci AUTONOMIJI
4. Resetovanje watchdog time-a na kraju loop-a

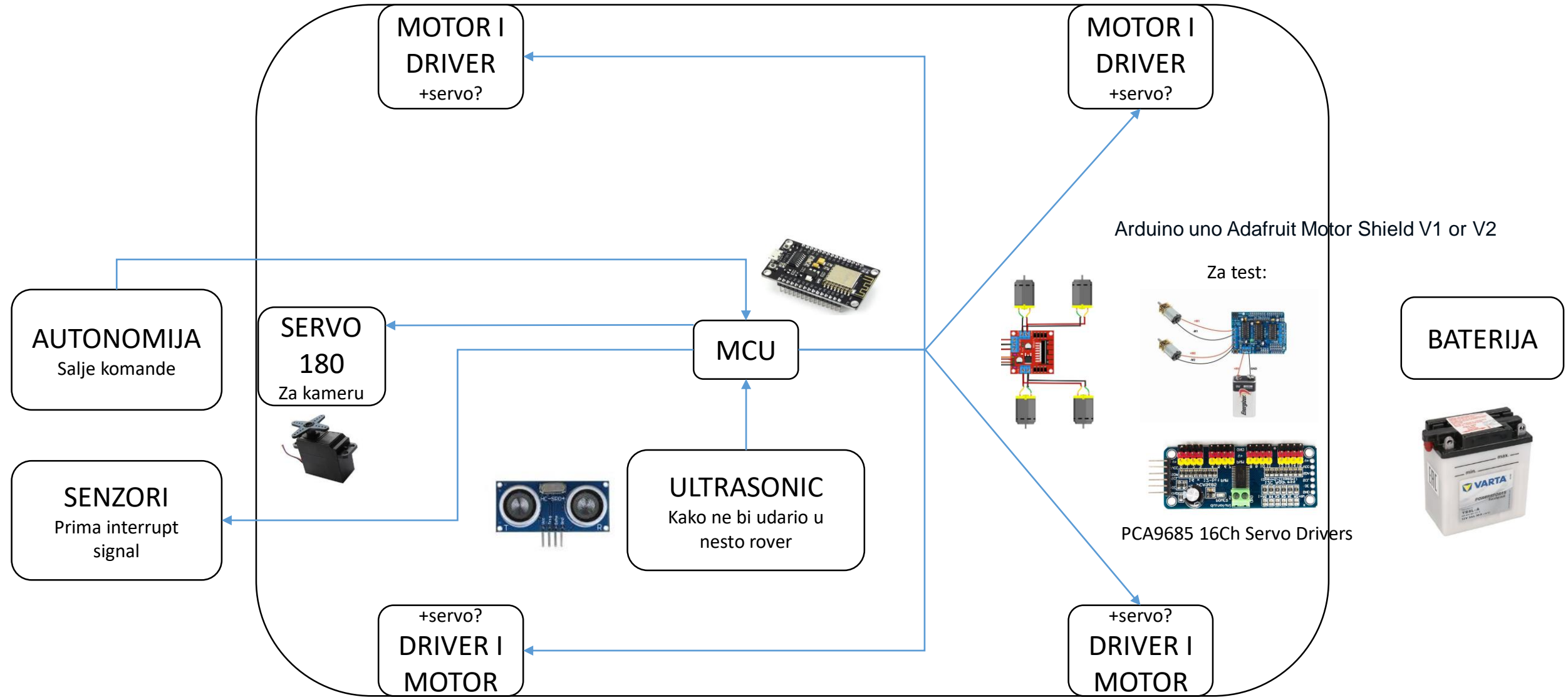
INTERRUPT (AKO JE POTREBAN):

1. Ako se pokrene onda salje podatke da ce da udari u nesto rover
2. Resetovanje watchdog timer-a
3. If (pin low) ugasi interrupt

Za kraj:

Nakon završenog koda proveriti vreme koje je potrebno da se jedna petlja završi

MOTORI



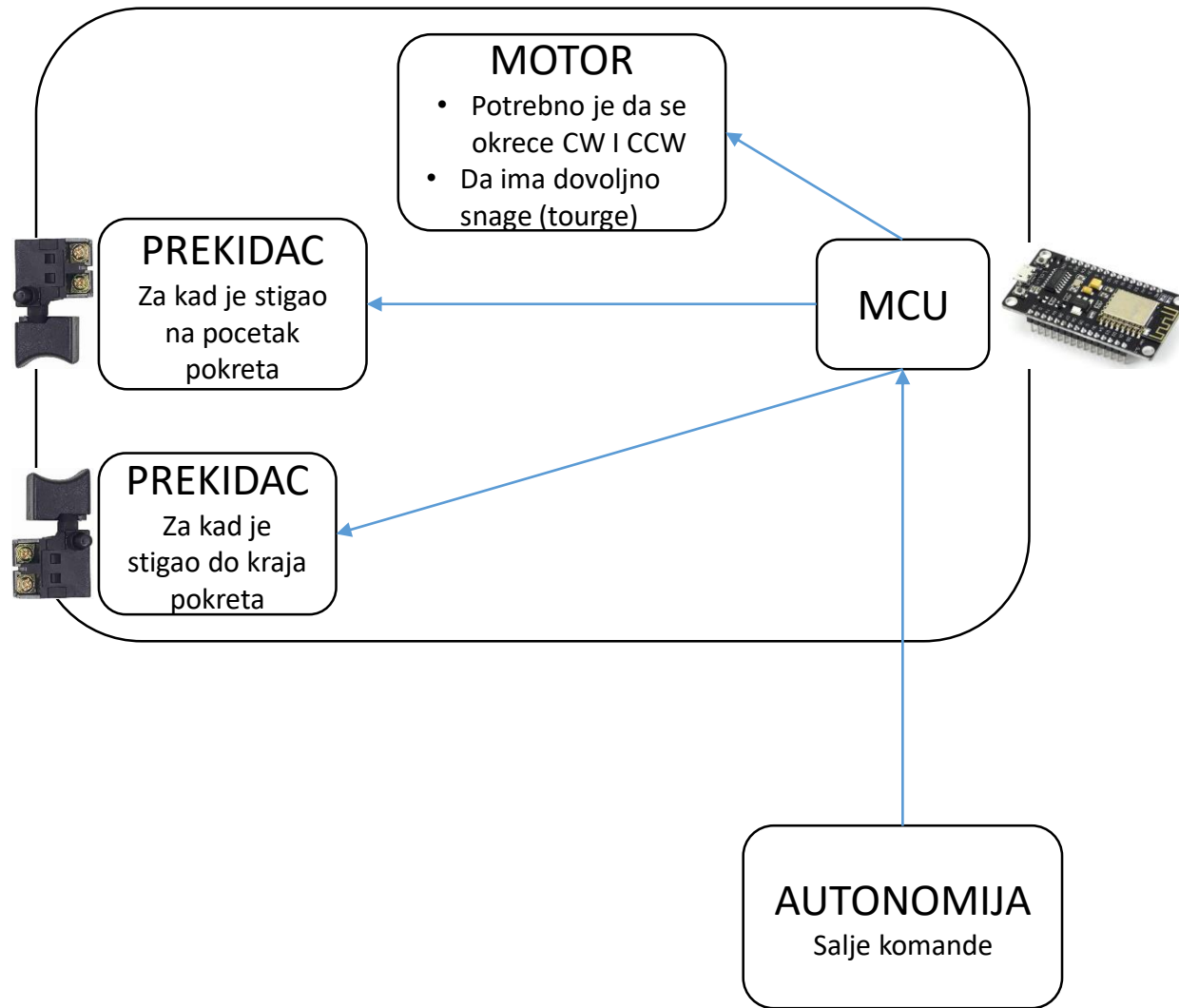
SETUP:

1. Vracanje serva na 0 poziciju
2. Inicijalizacija pinova za motore i pinova za led
3. Provera ultrasonic-a, ako cita cudne brojeve while(1) I salje poruku na ESP32 sta cita ili pin HIGH za interrupt (AKO JE POTREBAN) za SENZOR MCU, takodje ukljucuje crvenu LED da ne radi kako treba
4. Pokretanje Watchdog Timera

LOOP:

1. Citanje average od ultrasonik senzora
2. While(1) salje upozorenje ako ce da udari u nesto, cita ponovo avg, ukljucuje led da je u beskonacnoj I resetuje watchdog
3. Cekanje komande za motore
4. If(komanda razlicita od prosle):
Ukljucivanje motora I brzine sa PID po komandama koje je procitao
Pomeranje SERVA na motorima sa PID po komandama koje je procitao
5. Resetovanje watchdoga

BUSILICA



BATERIJA



SETUP:

1. Inicijalizuju se pinovi za motor, prekidače, i komunikacija sa drugim MCU-om.
2. Motor se postavlja u stanje mirovanja.
3. Inicijalizacija pinova za led
4. Inicijalizacija i pokretanje Watchdog Timera:

LOOP:

1. MCU čeka komande START,koje dolaze od "AUTONOMIJE" preko serijske komunikacije ili drugog protokola.
2. Uključuje LED ako je aktivan neki prekidač
3. (bolje bi bio interrupt umesto if-ova:)
4. If(START I PREKIDAC GORNJI AKTIVAN) ukljuci motor u stranu potrebnu za busenje
5. if (SWITCH_END) motor se prebacuje u suprotan smer.
6. Reset Watchdog Timer