



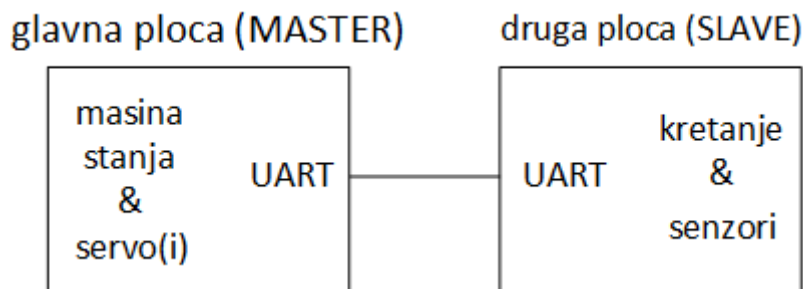
## Main state machine

### EUROBOT Main State Machine

U ovom dokumentu je potrebno prikazati osnove glavne mašine stanja.

**Ukratko:** Robot kontroliše 2 ploče. Na jednoj se vrti glavna mašina stanja i njom se kontroliše servo. Ona predstavlja MASTER, te ona inicira komunikaciju i šalje instrukcije drugoj ploči. Druga ploča omogućava kretanje i detektovanje prepreka UV senzorima i predstavlja SLAVE. Princip autonomne kontrole robota je sledeći. Glavna ploča prolazi kroz mašinu stanja (koja je napravljena prema taktici u toku meča), i sukcesivno šalje poruke kretanju kako robot treba da se pokrene. Glavna ploča takođe periodično uzima podatke sa druge ploče na osnovu kojih računa trenutnu poziciju robota na terenu i obrađuje informaciju da li senzori vide prepreku na kritičnom rastojanju.

Na slici 1 je prikazana jednostavna električna šema povezivanja mikrokontrolera i servoa.



Slika 1 – Ploče i njihova komunikacija

#### Taktika:

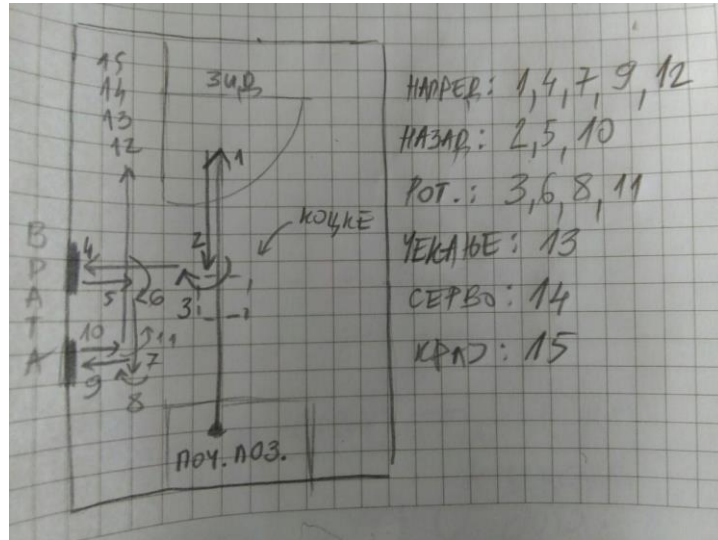
Taktika je sledeća:

- Na početku meča čeka se „povlačenje kablčića” kojim robot izlazi iz početnog stanja.
- Robot ide napred i gura zamak od peska u naznačenu oblast.
- Robot ide nazad i ravna se sa linijom daljih vrata
- Robot se okreće ka vratima, ide napred ka njima i zatvara ih
- Robot ide unazad i okreće se ka početku terena
- Robot ide napred i ravna se sa linijom bližih vrata
- Robot se okreće ka vratima i zatvara i njih
- Robot ide unazad i okreće se ka protivničkoj strani terena
- Robot ide napred do mesta gde bi mogao da smeta
- Robot čeka da istekne 90 sekundi
- Robot otvara kišobran
- Robot se resetuje

Na slici 2 je prikazan jednostavan grafik taktike.



## Main state machine

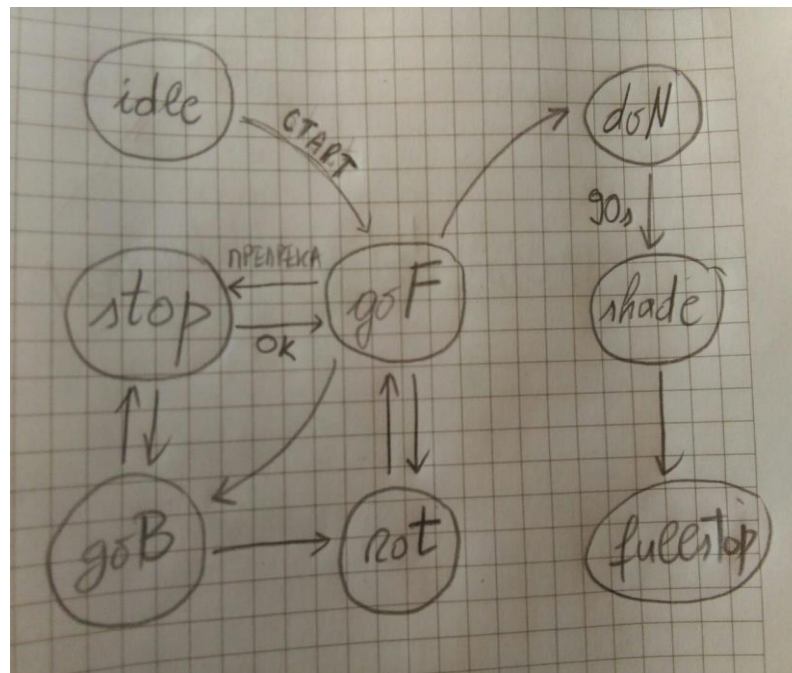


Slika 2 – Taktika

### Mašina stanja:

Glavna mašina stanja ima 8 stanja: *idle*, *goForward*, *goBackward*, *rotate*, *stop*, *doNothing*, *shade* i *fullstop*. Zamišljeno je da zahvaljujući taktici postoji neki plan u koja stanja treba doći i šta treba odraditi na terenu, dok je u slučaju odgovarajućih asinhronih događaja iste moguće procesirati i promeniti glavni tok izvršavanja.

Na slici 3 prikazana je glavna mašina stanja, a ispod nje su ukratko objašnjena stanja.



Slika 3 – Glavna mašina stanja



## Main state machine

---

**Idle:** Početno stanje. Ne radi se ništa, čeka se povlačenje kablova odnosno promena stanja na odabranom pinu mikrokontrolera što inicira prelazak u stanje *goForward* i početak meča.

**goForward:** Stanje u kom se robot kreće unapred. MASTER šalje instrukciju SLEJVU u vidu broja otkucaja enkodera koliko treba da se ide. Ta informacija se dobija na osnovu toga gde se tačno nalazimo i tačne pozicije u koju treba stići. Nakon primljenog ACK-a, čeka se da se dođe u zadatu poziciju. Ukoliko je sve u redu prelazi se na sledeće stanje po taktici. Ukoliko se stane zbog prepreke (signalizacija UV-a) prelazi se u stanje *stop* i čeka da se prepreka skloni. Vrednuju se samo prednji senzori.

**goBackward:** Stanje u kom se robot kreće unazad. Analogno stanju *goForward*.

**rotate:** Stanje u kom se robot rotira. Takođe se šalje instrukcija sa brojem otkucaja enkodera. Senzori se ne vrednuju. Po završetku kretanja, najverovatnije se prelazi u sledeće stanje po taktici (osim u slučaju popravljavanja pozicije robota).

**stop:** Stanje u koje se ulazi kada posmatrani UV senzori signaliziraju da postoji prepreka na kritičnom rastojanju od robota. Ne radi se ništa, čeka se da UV senzori prijave da se prepreka sklonila.

**doNothing:** Stanje u kom se čeka da tajmer koji odbrojava 90 sekundi promeni stanje u *shade*, što znači otvaranje kišobranja a zatim resetovanje svega. U slučaju neke greške, najbolje je zadržati se u ovom stanju. Takođe se u ovom stanju smeta protivniku.

**shade:** Stanje u kom se otvara kišobran. Ulazi se nakon što istekne 90 sekundi meča. Aktivira se servomotor povezan na istu ploču i pomera u željenu poziciju.

### Komunikacija:

Svu komunikaciju inicira MASTER, tako što ili šalje instrukcije ili traži podatke. Razlikuju se poruke koje MASTER šalje SLAVE-u i poruke koje MASTER dobija od SLAVE-a. Biće opisano šta poruke treba da sadrže, ali organizacija i raspored bitova u porukama kao i konvertovanje podataka u poruku i obratno nisu specificirani! U slučaju neuspelog slanja poruke (ako ACK signal nije stigao) još nekoliko puta se pokušava slanje svake poruke.

#### Poruke koje MASTER šalje SLAVE-u

Postoji 3 vrste poruka - zahtevanje podataka sa druge ploče (1, "gimme"), zahtevanje zaustavljanja robota (2, "stop") i slanje instrukcije za kretanje, tj. broja otkucaja enkodera (3, *string* (2 x *int*)). Po potrebi se sami sadržaji poruka mogu izmeniti.

1. vrsta poruke znači da SLAVE ploča treba da pošalje podatke o svom stanju i šalje se periodično na svakih 100ms pomoću *SysTick*-a.
2. vrsta poruke znači da je prepreka na kritičnom rastojanju i da se kretanje treba zaustaviti.
3. vrsta poruke su dva *int* podatka konvertovana u oblik poruke (*string*) koji predstavljaju otkucaje enkodera oba DC motora koje treba ostvariti.

#### Poruke koje MASTER prima od SLAVE-a:

Za sad postoji samo jedna vrsta ovih poruka i to je odgovor na 1. vrstu poruke odozgo. Dakle MASTER prima *string* o stanju ploče za kretanje i servoe, a u *stringu* se sadrže: 4 bita UV flegova (1 x *char*), stanje kretanja (1 x *char*), broj otkucaja 1. i 2. enkodera (2 x *int* unutar 8 x *char*) – ukupno 10 bajtova, što je fiksirana dužina ove poruke.

UV flegovi signaliziraju da se prepreka nalazi na kritičnom rastojanju od odgovarajućeg UV senzora a glavna ploča te flegove maskira u zavisnosti od toga u kom je stanju ili ih uopšte ne posmatra ako na primer zatvaramo vrata.



## Main state machine

Stanje kretanja je *char* koji služi kao ispomoć u kodu. Može biti: 'i' – initial, ništa se još nije desilo, 'f' – forward, kreće se unapred, 'b' – backward, kreće se unazad, 'r' – rotate, rotira se, 'w' – wait, čeka se da prođe UV opasnost, 's' – stop, zatraženo kretanje je završeno i 'x' – repaired, služi kao ispomoć u kodiranju a predstavlja stanje kada je izvršena popravka stanja nakon postizanja nedovoljno tačne pozicije na terenu.

Otkucaji 2 enkodera predstavljaju najnovije vrednosti otkucaja enkodera DC motora i koriste se za računanje nove trenutne pozicije na terenu, što onda usmerava i kontroliše dalji tok mašine stanja. Postoje formule za prebacivanje otkucaja enkodera u pređene centimetre, i prebacivanje otkucaja enkodera u poziciju na terenu. Pozicija na terenu predstavlja skup sledećih koordinata: x je koordinata x-ose, duže dimenzije terena, y je koordinata y-ose, kraće dimenzije terena i Q je ugao koji zaklapa pravac kretanja (okrenutosti prednje strane robota) sa x-osom. Potrebno je definisati koordinatni početak i početne pozicije robota kad smo levi i desni takmičar! Na osnovu trenutne pozicije na terenu mogu se u slučaju da nismo došli u željenu poziciju formirati nove instrukcije koje će tu poziciju popraviti. Dobijanje novih otkucaja enkodera na osnovu trenutne i željene pozicije vrši se preko računa sa vektorima, tačnije sabiranja vektora (pogledati kod). Takođe, na osnovu trenutne pozicije se određuje da li je potrebno vrednovati UV flegove (na primer kad zatvaramo vrata, nije).

### Kodovi i fajlovi:

Kodovi bitni za glavnu mašinu stanja nalaze se u fajlovima *main\_MainStateMachine.c* i *EUROBOT\_MainStateMachine.h*.

U prvom fajlu je *main* funkcija i glavni kod mašine stanja realizovan preko *switch-case* strukture i posebne funkcije *setNextState()* koja „u igru“ uključuje i asinhronu promene stanja iz prekida. Inače, kod je organizovan po principu „odložene obrade prekida“ pa se u prekidu samo setuju odgovarajući flegovi, a sama obrada vrši negde drugde, uz malo ali podnošljivo kašnjenje. Ako je setovan fleg *gimme* potrebno je zatražiti podatke od druge ploče. Ako je aktivan fleg *update* podaci su stigli i treba ih obraditi unutar funkcije *updatePosition()*. Inače, ako je aktivan fleg *end* prelazi se u stanje *shade* i igra se privodi kraju.

Funkcija *updatePositon()* ažurira trenutnu poziciju na terenu (*currentPosition*), postavlja bitove maske za UV flegove (*maskUV*), po potrebi inkrementira promenljivu *cnt* i po potrebi pravi nove poruke (*repairMessages[]*), potrebne ukoliko je potrebno popraviti poziciju robota na terenu (nismo stigli tačno u željeno stanje). Promenljiva *cnt* odnosi se na niz *positions[]* i niz *messages[]*. Naime, igra je zamišljena tako da se po smišljenoj taktici unapred zna u koje pozicije na terenu želimo da stignemo, samim tim i koje poruke želimo da pošaljemo (ponekad). U idealnom slučaju, samo bismo se tako i kretali. Međutim, u slučaju da se nešto nepredviđeno desi potrebno je promeniti poruku, ili sračunati nove otkucaje enkodera da dođemo u željenu poziciju (primer za to je funkcija *nextState()*). Promenljiva *cnt* stoga služi kao brojač indikator dokle se stiglo u planskom izvršavanju koda i taktike.

U fajlu *EUROBOT\_MainStateMachine.h* se nalaze sve potrebne definicije, tipovi, promenljive i funkcije koje se koriste u istom fajlu ili u fajlu *main\_MainStateMachine.c* i u fajlu *stm32f10x\_it.c*.

### Napomene:

0. Sračunati početne uslove i inicijalizovati *positions[cnt]* i *messages[cnt]*
1. Opisi funkcija, promenljivih i ostatka koda su u gorepomenutim fajlovima detaljniji
2. Funkcije tipa *tipPodatkaJedan2tipPodatkaDva* treba napisati. (Marko)
3. Razmisliti gde u kodu i u kom stanju je potrebno resetovati određene delove sistema
4. Razmisliti da li kašnjenje „odložene obrade prekida“ utiče na normalno funkcionisanje koda kao i da li negde treba zabraniti prekide