# Maketa pokretne trake

**Juranović, Vinko**

**Master's thesis / Diplomski rad**

**2017**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:200:797506

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-04-23**

*Repository / Repozitorij:*

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Diplomski studij elektrotehnike**

# MAKETA POKRETNE TRAKE

**Diplomski rad**

**Vinko Juranović**

**Osijek, 2017.**

# SADRŽAJ:


1 UVOD ..... 1
1.1 Zadatak diplomskog rada ..... 1
2 OPIS SIMULACIJE ..... 2
2.1 CoDeSys V2.3. ..... 2
2.2 Pogon i upravljanje trakom ..... 3
2.2.1 Simulacija motora ..... 3
2.2.2 Virtualni enkoder ..... 7
2.2.3 Upravljanje trakom ..... 8
2.3 Sustav nadzora proizvoda ..... 10
2.3.1 Pridruživanje inicijalnih podataka proizvodu ..... 10
2.3.2 Identifikacijska oznaka proizvoda ..... 11
2.3.3 Prijenos podataka između spremnika ..... 12
2.3.4 Računanje pomaka ..... 13
2.3.5 Osvježavanje pozicije ..... 14
2.3.6 Zauzetost područja ..... 16
2.3.7 Sinkronizacija i korekcija pozicije ..... 17
2.3.8 Promjena smjera proizvoda ..... 19
2.3.9 Udaljenost između proizvoda ..... 20
2.3.10 Odlaganje proizvoda ..... 21
3 VIZUALIZACIJA MAKETE ..... 23
4 OPIS KOMPONENATA I SUSTAVA ZA IZRADU MAKETE ..... 27
4.1 Opis makete pokretne trake ..... 27
4.2 Simatic S7-400 ..... 29
4.3 Simatic Manager ..... 30
4.4 Osnovni elementi makete ..... 31
4.5 Projektiranje i izrada sklopa za upravljanje motorima ..... 34
4.6 Kreiranje hardverske konfiguracije ..... 37
4.7 Procesiranje ulaznih signala ..... 40
4.8 Procesiranje izlaznih signala ..... 41
4.9 Prikaz makete ..... 42
5 ZAKLJUČAK ..... 43

6 LITERATURA ..... 45
7 POPIS SLIKA ..... 46
ŽIVOTOPIS ..... 48
SAŽETAK ..... 49
ABSTRACT ..... 50

# 1 UVOD

Razvoj računalnih tehnologija pronašao je svoju primjenu u raznim granama industrije pa tako i u metalnoj industriji. Kreirana su računala koja omogućuju kvalitetnije i uspješnije vođenje procesa, u kojima je vjerojatnost ljudske pogreške svedena na najnižu razinu. Automatizacija proizvodnje uključuje i automatizaciju transporta proizvoda. Jedan od načina transporta proizvoda je korištenjem trakastih transportera. U većini slučajeva uloga transportera osim prijenosa proizvoda je i praćenje proizvoda prilikom transporta. Cilj diplomskog rada je napisati program i napraviti maketu pokretne trake.

Rad se sastoji od pet poglavlja. U drugom poglavlju su opisane osnove rada u programskom okruženju *CoDeSys*. Osim upoznavanja s programskim okruženjem opisani su i blokovi korišteni za simuliranje rada makete, blokovi korišteni za praćenje proizvoda u simulaciji kao i izgled same simulacije. Treće poglavlje odnosi se na vizualizaciju simulacije makete, dok je u četvrtom poglavlju opisano teorijsko upoznavanje s programskim okruženjima *Step7* u kojem je napravljeno programsko rješenje za upravljanje maketom. Također su navedeni svi važniji problemi koji su nastali prilikom izrade makete. Rad završava petim poglavljem gdje je dan kratak osvrt na rad, odnosno zaključak rada.

## 1.1 Zadatak diplomskog rada

Zadatak diplomskog rada je izrada makete koja se sastoji od tri sekcije pokretne trake s električnim motorima i detektorima pozicije. Maketa je dodatno opremljena svjetlosnim barijerama za detekciju materijala. Osim izrade makete, rad uključuje i razvoj programa i simulaciju u alatu Step7 i CoDeSys, te vizualizaciju u alatu CoDeSys.

# 2 OPIS SIMULACIJE

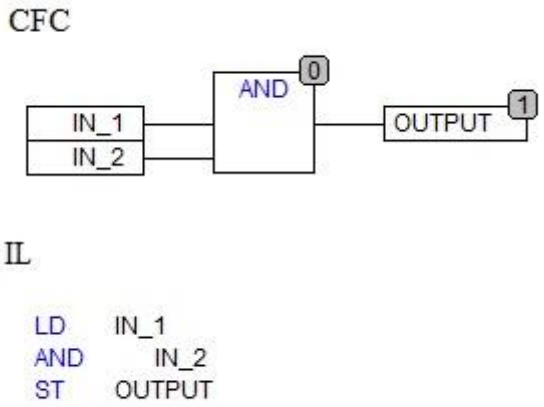
Simulacija i vizualizacija makete napravljene su u programskom okruženju CoDeSys. Osim osnovnog upoznavanja s programom CoDeSys objašnjeni su važniji dijelovi koda vezani za nadzor proizvoda i simuliranje rada makete.

## 2.1 CoDeSys V2.3.

Programsko okruženje za konfiguriranje, simuliranje i programiranje PLC-ova. *CoDeSys* kao i *Simatic Manager* omogućuje programiranje u različitim programskim jezicima. U odnosu na *Simatic Manager CoDeSys* nudi mogućnost korištenja još nekih programskih jezika kao što su:

-Lista instrukcija koja se označava kao IL (eng. *Instruction List*)

-Kontinuirani funkcijsko-blokovski dijagram koji se označava kao CFC(eng. *Countinious Function Chart*)




**Slika 2.1.** *Primjer programskih jezika CFC i IL.*

U jeziku liste instrukcija se programira slično kao u asembleru, odnosno unose se naredbe u dio prozora predviđen za programiranje. Programi u CFC-u se kreiraju slično kao u FBD-u. Osnovna razlika je u postavljanju blokova, to jest u odnosu na FBD blokovi u CFC-u mogu biti slobodni jedni u odnosu na druge. Simulacija makete je većinom napravljena u CFC programskom jeziku (slika 2.1).

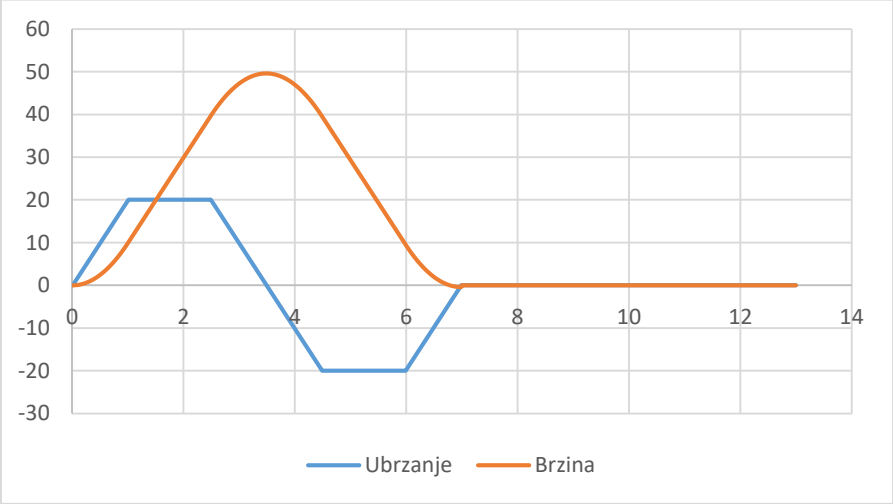
## 2.2 Pogon i upravljanje trakom

Zadatak diplomskog rada je napraviti maketu i simulaciju makete pokretne trake. Da bi simulacija što bolje prikazala stvarnu maketu bilo je potrebno napraviti i simulirati rad motora i enkodera.

### 2.2.1 Simulacija motora

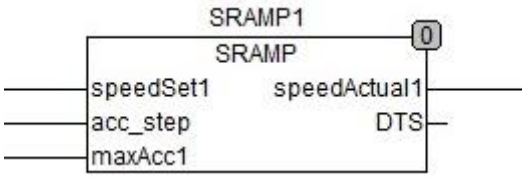
Za izradu makete se koriste istosmjerni motori bez četkica čiji rad se može simulirati takozvanom S-rampom. Rampa je dobila naziv po svom specifičnom obliku koji podsjeća na slovo S. Upotreba rampi za puštanje motora u pogon u industriji je vrlo česta kako bi se prilikom pokretanja i zaustavljanja procesa dobili lagani pokreti. Primjer toga može biti traka za punjenje boca u tvornici koja je pokretana istosmjernim motorom. Ako prilikom pokretanja motora odmah dovedemo nazivni napon, motor će pokušati što prije dostići nazivnu brzinu, ali ako napon koji dovodimo na motor ima oblik S-rampe motor će lagano ubrzavati i usporavati što je u primjeru punionice boca jako bitno kako ne bi došlo do prevrtanja i padanja proizvoda s trake. Osim toga, rampe se koriste i zbog samih motora koji su realni elementi i samim time posjeduju i odgovarajuću tromost. Treći razlog korištenja rampi su gubitci koji su znatno manji prilikom takvog puštanja motora u pogon u odnosu na slučaj kada bi se na motor trenutno doveo nazivni napon. Na taj način se smanjuje opterećenje na motor.

U simulaciji se brzina motora mijenja po S krivulji. Oblik krivulje nastaje integriranjem linearne rampe koja bi u ovom slučaju predstavljala ubrzanje kao na slici 2.2.




**Slika 2.2.** *S krivulja.*

Izgled bloka za simulaciju motora je prikazan na slici 2.3.




**Slika 2.3.** *Blok SRAMP.*

Ulazne varijable u blok su:

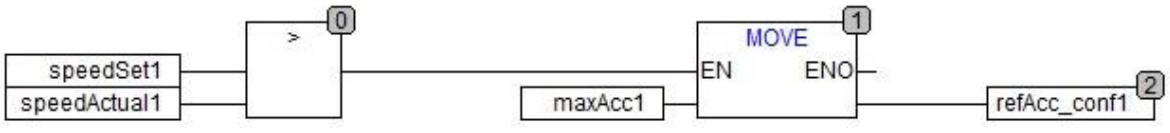
- Željena brzina
- Korak ubrzanja
- Maksimalno ubrzanje

Izlazne varijable iz bloka su:

- Trenutna brzina
- DTS (eng. *Distance to stop*)

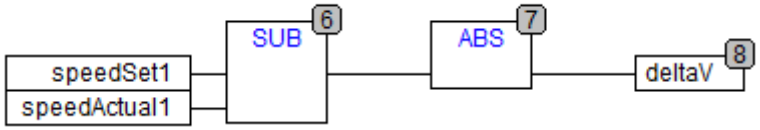
Željena brzina je ulazni parametar koji zadaje bloku koju brzinu treba dostići. Korak ubrzanja definira korisnik u ovisnosti kakav oblik S krivulje želi. Maksimalno ubrzanje je najveća razina ubrzanja do koje blok ide. Od maksimalnog ubrzanja ovisi hoće li krivulja brzine imati linearni dio ili ne, jer kada je ubrzanje konstantno brzina raste linearno u vremenu.

Trenutna brzina je vrijednost koju daje blok na izlazu. DTS (eng. *Distance to stop*) predstavlja pređeni put koji će motor prijeći kada se krene zaustavljati u određenom trenutku. Ova vrijednost kao i trenutna brzina se računa u svakom ciklusu PLC-a i govori koliki put će motor prijeći prije nego brzina padne na nulu.



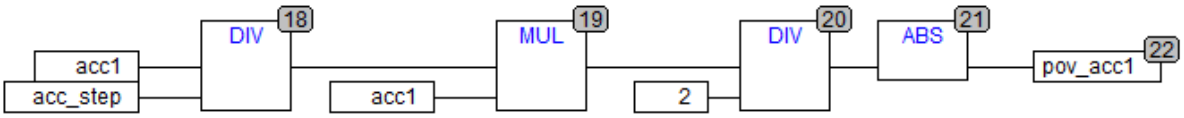

**Slika 2.4.** *Ispitivanje zadane i trenutne brzine.*

Kada se na ulaz bloka postavi brzina na koju blok treba doći, prvo se ispituje da li je zadana brzina veća ili manja od trenutne brzine. Ako je zadana brzina veća od aktualne kao na slici 2.4., vrijednost maksimalnog ubrzanja se sprema u varijablu referentnog ubrzanja.



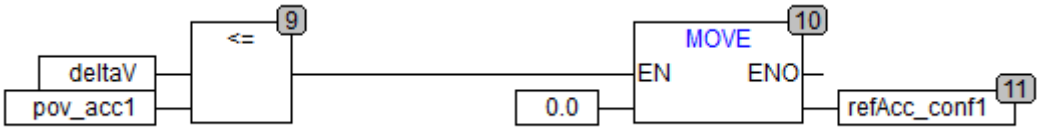

**Slika 2.5.** *Razlika između zadane i trenutne brzine.*

Kako bi krivulja mogla doći na zadanu brzinu, u svakom ciklusu PLC-a se računa preostala brzina, odnosno razlika između zadane i trenutne brzine (slika 2.5.).



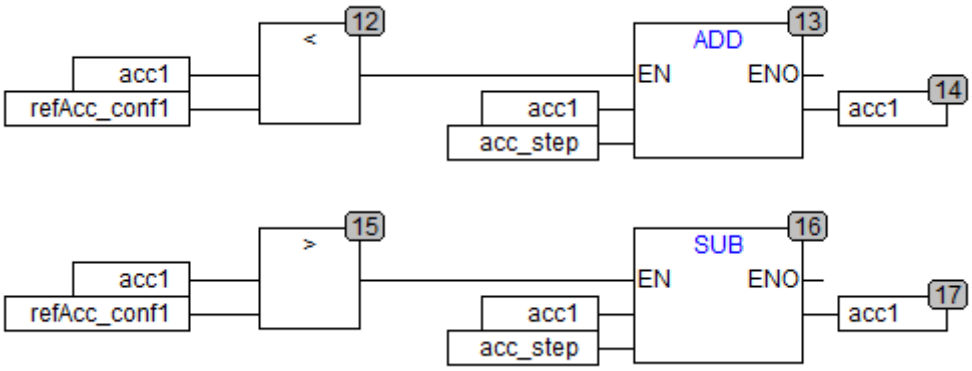

**Slika 2.6.** *Računanje površine ispod krivulje ubrzanja.*

Potom je potrebno izračunati površinu ispod krivulje ubrzanja koja predstavlja brzinu u tom trenutku (slika 2.6.).



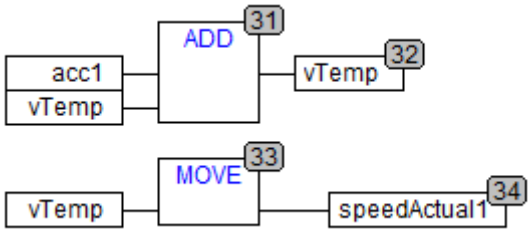

**Slika 2.7.** *Usporedba preostale brzine i površine ubrzanja.*

S obzirom na to da se brzina dobije integriranjem ubrzanja i da bi blok mogao doći na zadanu brzinu, provjerava se da li je preostala brzina manja ili jednaka od površine ispod krivulje ubrzanja. Ukoliko preostala brzina postane manja ili jednaka od površine ubrzanja, u varijablu referentnog ubrzanja se upisuje nula i ubrzanje se prema zadanom koraku spušta na nulu kako bi brzina došla na zadanu vrijednost (slika 2.7.).



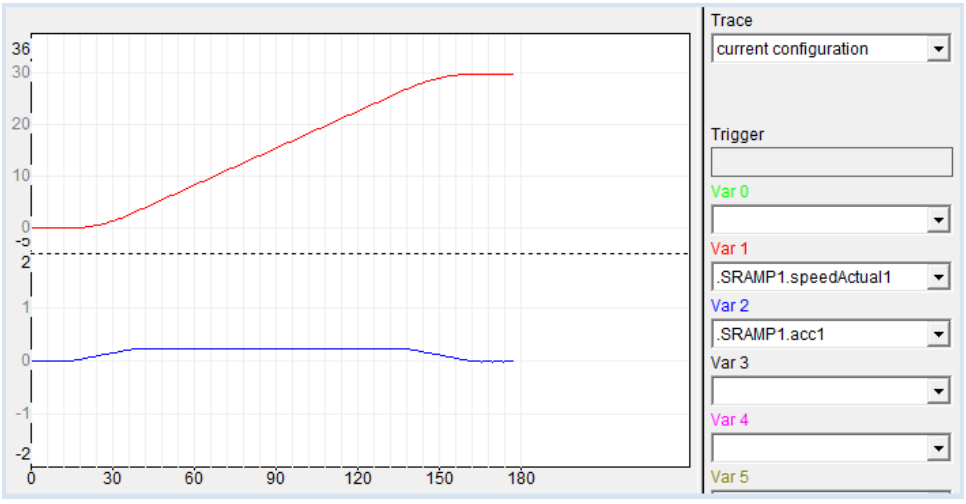

**Slika 2.8.** *Rampa ubrzavanja i usporavanja.*

Rampa ubrzanja prvo provjerava treba li povećavati ili smanjivati vrijednost ubrzanja u odnosu na referentno ubrzanje. Na slici 2.8. je prikazano kako se referentno ubrzanje mijenja ovisno od usporedbe zadane i trenutne brzine.




**Slika 2.9.** *Integriranje brzine.*

Slika 2.9. prikazuje integriranje ubrzanja i spremanje u varijablu *vTemp.* Integriranje brzine se radi u svakom ciklusu PLC-a.

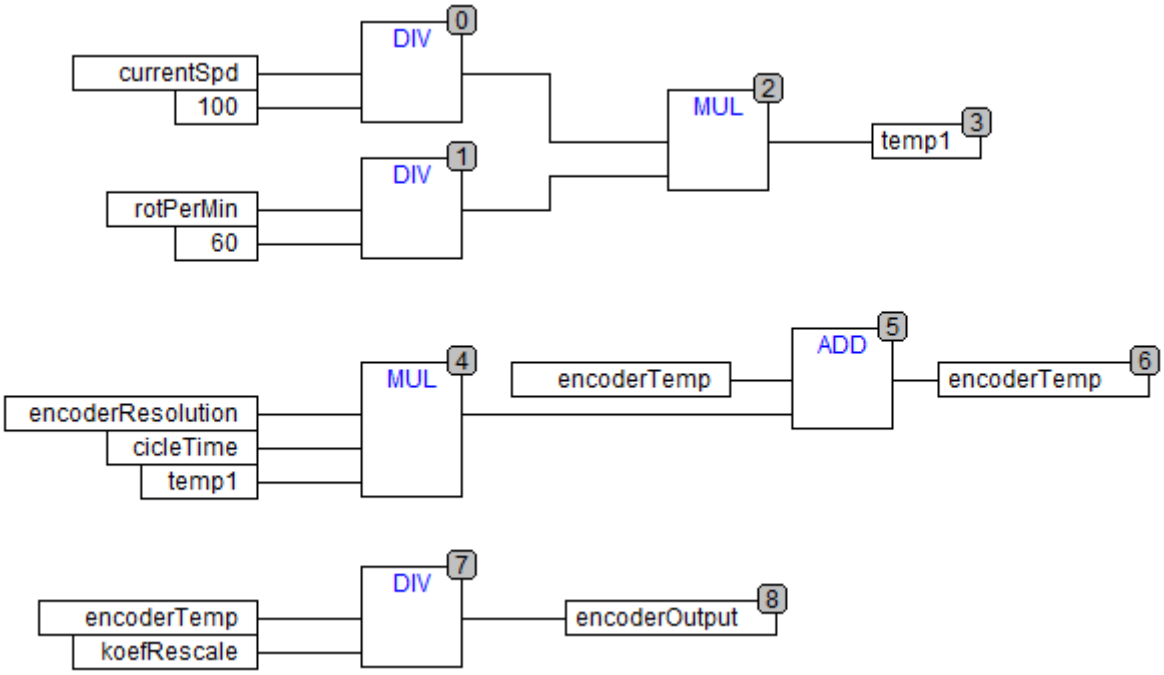



**Slika 2.10.** *Grafički prikaz brzine i akceleracije.*

Slika 2.10. je grafički prikaz brzine (crvena boja) i ubrzanja (plava boja) koji se dobiju na izlazu iz bloka *SRAMP* uz zadanu brzinu 30, te korak akceleracije 0,01 i maksimalno ubrzanje 0,25.

### 2.2.2 Virtualni enkoder

Kako bi se mogao mjeriti pomak, potrebno je simulirati rad inkrementalnog enkodera. Pretpostavlja se da je enkoder spojen na osovinu motora čija je nazivna brzina 60 *RPM*, te da je rezolucija enkodera 1024, odnosno polumjer valjka koji pokreće traku 15 *mm*.

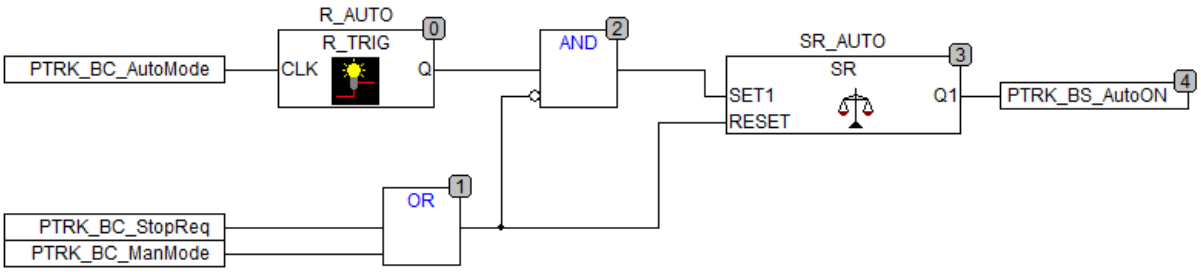



**Slika 2.11.** *Enkoder.*

Blok enkodera ima jednu ulaznu i jednu izlaznu varijablu. Ulazna varijabla je brzina koju enkoder dobije iz bloka SRAMP, a izlazna varijabla je pomak u odnosu na navedene parametre brzine, rezolucije i radijusa valjka. Zbog toga što blok brzinu dobije kao postotak, prvo je potrebno izračunati stvarnu brzinu, zatim se brzina množi s rezolucijom enkodera i vremenom ciklusa PLC-a, te se u svakom ciklusu pridodaje vrijednosti enkodera iz prethodnog ciklusa (slika 2.11.).

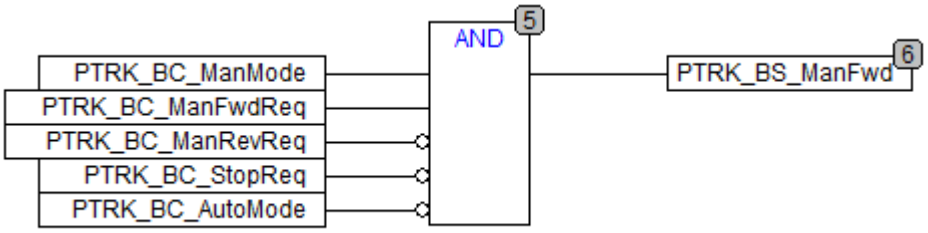
### 2.2.3 Upravljanje trakom

Za upravljanje trakom potrebno je definirati načine rada i funkciju trake. Traka ima dva načina rada: ručni i automatski. Sve naredbe za načine rada su binarnog tipa.



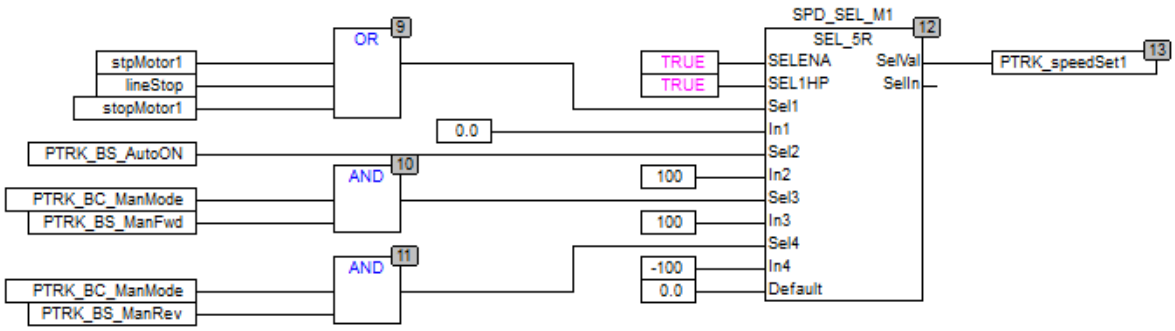

**Slika 2.12.** *Odabir automatskog upravljanja trakom.*

Automatski način rada se uključuje postavljanjem logičke jedinice u varijablu *PRTK_BC_AutoMode* uz ispunjene uvjete da traka nije zaustavljena, odnosno da nije dana naredba za zaustavljanje trake i da prethodno nije odabran ručni način upravljanja. SR bistabil se koristi za zadržavanje stanja kao i za promjenu stanja, ukoliko je korisnik odabrao u međuvremenu ručni način rada ili ako je netko dao naredbu za zaustavljanje trake (slika 2.12.).




**Slika 2.13.** *Odabir manualnog upravljanja trakom.*

Ako se odabere ručni način rada, potrebno je odrediti i smjer u kojem će se traka vrtjeti. Na slici 2.13. je prikazan primjer odabira ručnog načina upravljanja u smjeru naprijed koji podrazumijeva kretanje trake s lijeva na desno. Također, iz sigurnosnih razloga je postavljeno da će se traka pomjerati naprijed ili nazad samo dok korisnik drži pritisnuto dugme za naprijed ili nazad. Traka će se pokrenuti ako su ispunjeni sljedeći uvjeti: odabran ručni način upravljanja, odabran samo jedan smjer vrtnje i da nisu izdane naredbe za zaustavljanje trake i za automatski način rada.




**Slika 2.14.** *Odabir brzine.*

Postavljanje brzine se radi pojedinačno za svaki motor korištenjem bloka *SEL_5R* (slika 2.14.). U navedenom primjeru, blok postavlja brzinu za motor 1.

```
IF SELENA THEN
    IF SEL1 AND (SEL1HP OR SelIn=1 OR SelIn=0)THEN
        SelVal:= IN1;
        SelIn:=1;
    ELSIF SEL2 AND (SEL1HP OR SelIn=2 OR SelIn=0)THEN
        SelVal:= IN2;
        SelIn:=2;
    ELSIF SEL3 AND (SEL1HP OR SelIn=3 OR SelIn=0)THEN
        SelVal:= IN3;
        SelIn:=3;
    ELSIF SEL4 AND (SEL1HP OR SelIn=4 OR SelIn=0)THEN
        SelVal:= IN4;
        SelIn:=4;
    ELSE
        SelVal:= default;
        SelIn:=0;
    END_IF
ELSE
    SelVal:= default;
    SelIn:=0;
END_IF
```

**Slika 2.15.** *Blok SEL_5R.*

Blok *SEL_5R* služi za odabir između više ulaznih parametara i sprema vrijednost u izlaznu varijablu. Blok je zbog svoje kompleksnosti napravljen u programskom jeziku ST (slika 2.15.). Bitno je napomenuti za ovaj blok da najveći prioritet prilikom odabira ima ulaz 1 ako se ulazna varijabla SEL1HP postavi u logičku jedinicu.

## 2.3 Sustav nadzora proizvoda

Nadzor proizvoda je jako bitan element u procesu industrijske automatizacije. Omogućava dobivanje informacija i praćenje proizvoda u svakom trenutku. Jedna od prednosti sustava nadzora je pravovremeno reagiranje u slučaju kvara, nezgode ili neke druge nepredviđene situacije. Sustav nadzora je osmišljen za simulaciju makete pokretne trake i sastoji se od sljedećih elemenata: inicijalna dodjela podataka proizvodu, sinkronizacija podataka o proizvodu, osvježavanje podataka i odlaganja proizvoda.

### 2.3.1 Pridruživanje inicijalnih podataka proizvodu

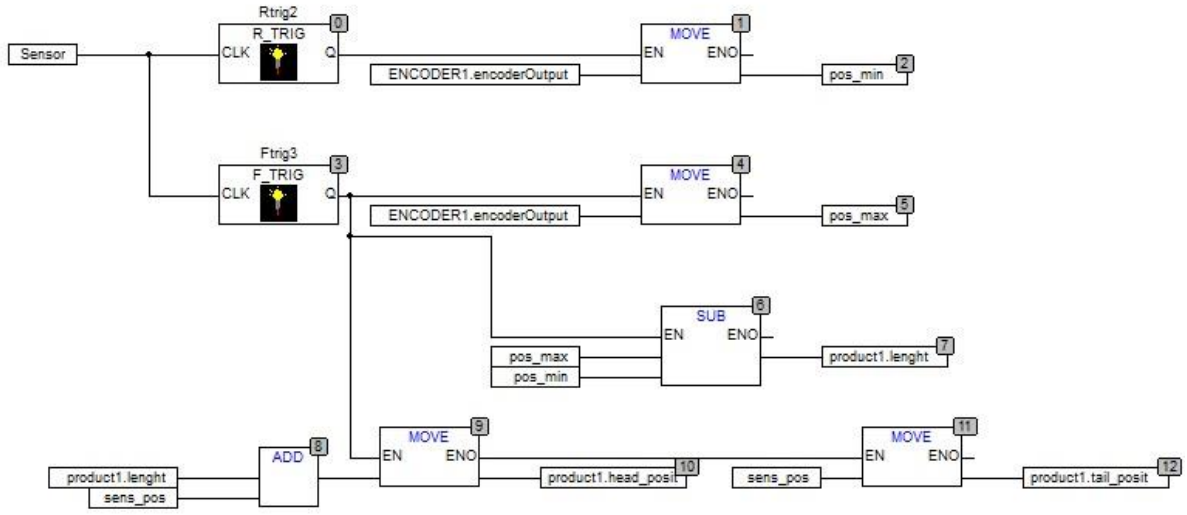
U simulaciji senzor S1 je definiran kao senzor koji će vršiti inicijalizaciju proizvoda. Senzor S1 je zamišljen kao optički prekidač s prijemnikom i predajnikom koji radi u infracrvenom dijelu spektra. Nalazi se na udaljenosti od 150 *mm* u odnosu na početak trake. Kada proizvod stigne na senzor S1 dodjeljuju mu se sljedeći podatci:

- *ID_product* – identifikacijska oznaka proizvoda
- *length* – duljina proizvoda
- *head_posit* – pozicija glave
- *tail_posit* – pozicija repa
- *curr_area* – označava područje na kojem se proizvod nalazi

```
TYPE TRK_PROD :
STRUCT
    ID_product: REAL;    (*ID of product*)
    lenght: REAL;        (*Lenght of product*)
    head_posit: REAL;    (*Head position*)
    tail_posit: REAL;    (*Tail position*)
    curr_area: REAL;     (*Current area position of product*)
END_STRUCT
END_TYPE
```

**Slika 2.16.** *Definiranje strukture podataka.*

Slika 2.16. prikazuje definiranje strukture podataka. Instance te strukture predstavljaju spremnike za podatke.

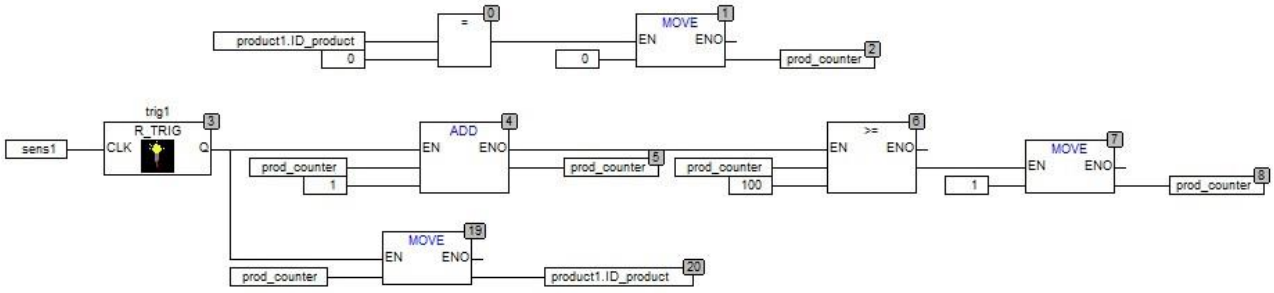



**Slika 2.17.** *Definiranje pozicije glave, repa i duljine proizvoda.*

Slika 2.17. prikazuje način dodjele pozicije glave, repa i duljine proizvoda na senzoru S1. Trigeri detektiraju rastući i padajući brid spremajući vrijednosti enkodera u tom trenutku. Na osnovu razlike između minimalne i maksimalne vrijednosti dobije se duljina proizvoda. Pozicija glave je jednaka zbroju udaljenosti na kojoj se senzor nalazi i duljini proizvoda, a pozicija repa je jednaka poziciji senzora.

### 2.3.2 Identifikacijska oznaka proizvoda

Svakom novom proizvodu koji dolazi na senzor S1 potrebno je dodijeliti identifikacijski broj ID (eng. *identification* ). Osnovni zahtjev za dodjelu ID-a je jedinstvenost, odnosno da svaki proizvod ima različit ID prema kojem se identificira i koji služi za čitanje podatka o tom proizvodu.




**Slika 2.18.** *Dodjeljivanje ID-a proizvodu.*

Prilikom dolaska novog proizvoda potrebno je prvo provjeriti da li je prethodno prošao bar jedan proizvod, odnosno da li je ID u prvom spremniku različit od nula. Ova informacija je

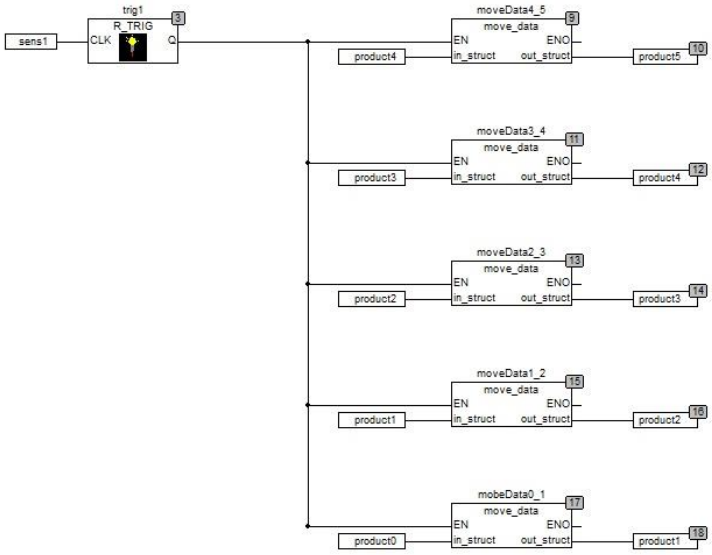
bitna zbog brojača proizvoda koji je za potrebe ove simulacije postavljen na vrijednost 100 i provjerava se njegova vrijednost. Dogovoreno je da se nakon 100 proizvoda može ponoviti identifikacijski broj proizvoda jer je već prošlo dovoljno proizvoda. Brojač proizvoda se povećava za jedan dolaskom svakog novog proizvoda, nova vrijednost brojača predstavlja ID novog proizvoda koji je stigao na traku (slika 2.18.).

### 2.3.3 Prijenos podataka između spremnika

U simulaciji se koristi šest spremnika, od kojih nulti spremnik služi za brisanje podataka i čišćenje spremnika i taj se spremnik naziva *product0*. Ostali spremnici imaju naziv od *product1* do *product5*. Podrazumijeva se da na traci neće biti više od pet proizvoda u isto vrijeme. Do prijenosa podataka između spremnika dolazi iz dva razloga:

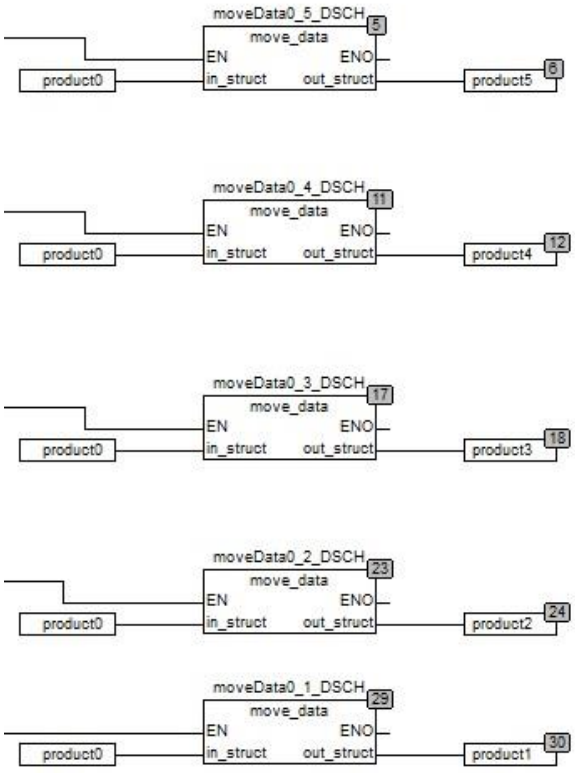
- dolazak novog proizvoda
- odlaganje proizvoda koji je stigao do kraja trake

Spremnik za proizvode se ponaša kao stog koji je poznat pod nazivom FIFO (eng. *First In First Out*). Dakle, proizvod koji je prvi ušao u spremnike prvi će i izaći iz spremnika. Ovakav način spremanja proizvoda u spremnike omogućuje lakše praćenje proizvoda.




**Slika 2.19.** *Prijenos proizvoda između spremnika.*

Prilikom dolaska novog proizvoda na senzor S1, potrebno je postojeće proizvode u spremniku prebaciti u sljedeći spremnik. Osim pomjeranja podataka potrebno je očistiti prvi spremnik, kao na slici 2.19., kako bi se novi proizvod mogao inicijalizirati i spremiti.

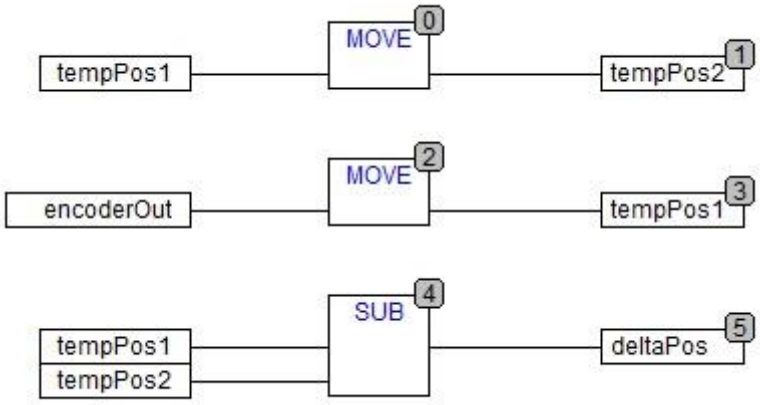



**Slika 2.20.** *Brisanje proizvoda iz spremnika.*

Kada proizvod dođe do kraja transportne trake mora se izbrisati iz spremnika. U ovom slučaju potrebno je voditi računa da je spremnik osmišljen kao FIFO stog, pa se zbog toga prvo briše zadnji proizvod u spremniku jer je taj proizvod prvi došao na traku (slika 2.20.).

### 2.3.4 Računanje pomaka

Da bi se proizvod mogao pratiti cijelo vrijeme od inicijalizacije do odlaganja s trake potrebno je računati pređeni put proizvoda u svakom ciklusu PLC-a. Osim toga uzima se u obzir na kojoj traci se nalazi proizvod, odnosno koji od tri motora pogoni traku. Za računanje pređenog puta koristi se blok koji se naziva *DELTA_POS*. Ulazni parametar bloka je vrijednost koju daje enkoder, a izlazna vrijednost je prijeđeni put koji je proizvod prešao na sekvenci na kojoj se u tom trenutku nalazi.




**Slika 2.21.** *Računanje pređenog puta po ciklusu.*

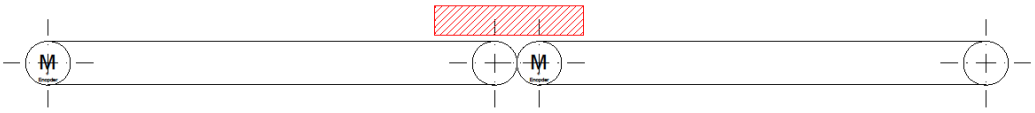
Prijeđeni put u jednom ciklusu se računa pomoću dvije privremene varijable. U svakom ciklusu PLC-a vrijednost enkodera iz prethodnog ciklusa se sprema u jednu privremenu varijablu i oduzima se od trenutne vrijednosti enkodera u trenutnom ciklusu (slika 2.21.).

### 2.3.5 Osvježavanje pozicije

Praćenje proizvoda u stvarnom vremenu se ostvaruje dodavanjem ili oduzimanjem pomaka koji je taj proizvod napravio u pojedinom ciklusu. Kod osvježavanja pozicije proizvoda dva su scenarija:

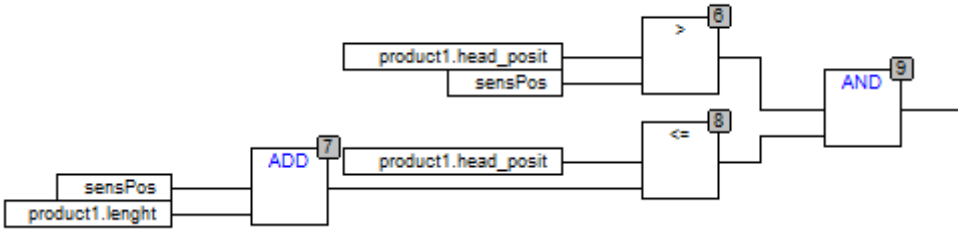
-proizvod se nalazi na dvije sekvence (slika 2.22.)

-proizvod se nalazi samo na jednoj sekvenci (slika 2.26.)



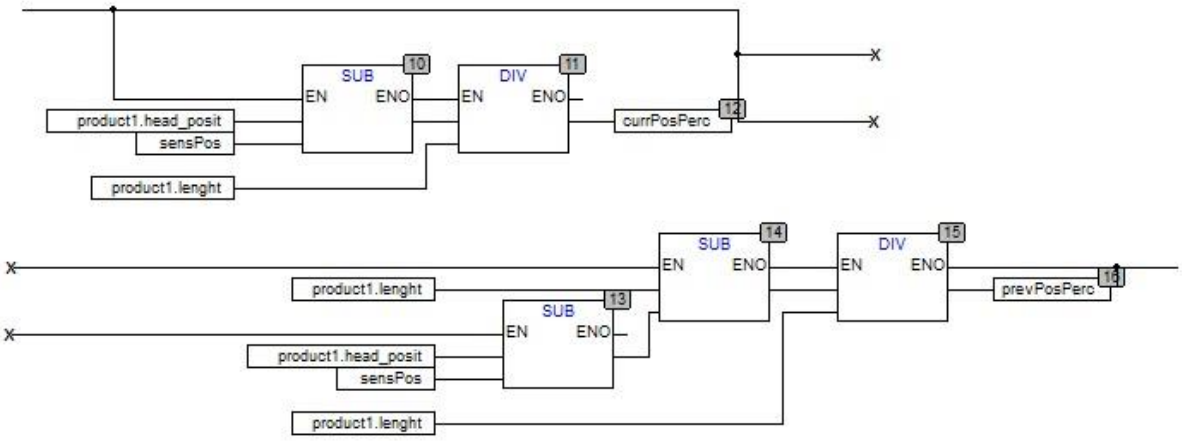
**Slika 2.22.** *Proizvod se nalazi na dvije sekvence.*

Za prvi scenarij potrebno je prvo ispitati nalazi li se proizvod na dvije trake.

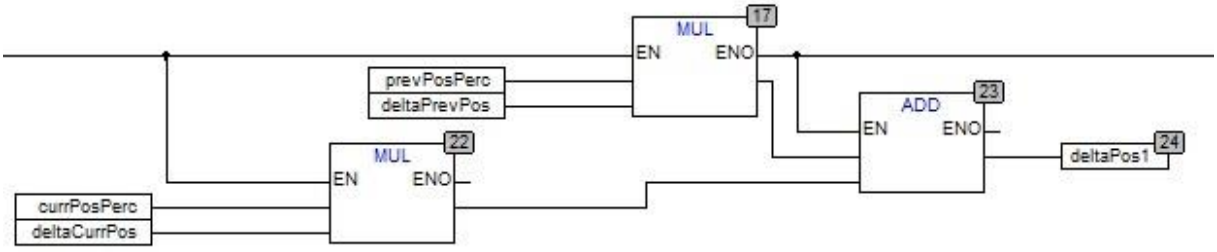



**Slika 2.23.***Uvjet da se proizvod nalazi na dvije trake.*

Na slici 2.23.vrši se ispitivanje nalazi li se proizvod u tom trenutku na dvije trake. Ispituju se uvjeti da li je pozicija glave veća od pozicije senzora, koji se nalazi na granici dvije trake i vrijedi za senzore S2 i S3. Zatim se ispituje da li je pozicija glave manja ili jednaka od zbroja pozicije senzora i duljine proizvoda. Ukoliko su ispunjena oba uvjeta izračunava se postotak proizvoda na trakama kao što je prikazano na slici 2.24.



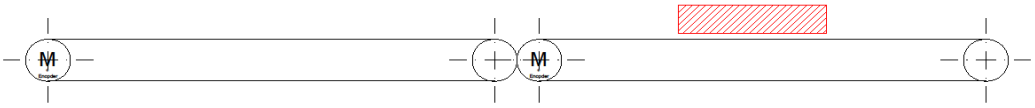

**Slika 2.24.** *Postotak materijala na pojedinom području.*




**Slika 2.25.** *Računanje ukupnog pređenog puta.*

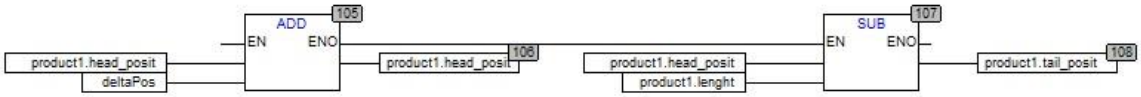
U slučaju kada se proizvod nalazi na dvije trake, pređeni put po ciklusu se računa kao zbroj pređenih puteva tih sekvenci, gdje je pređeni put pojedine sekvence pomnožen s postotkom materijala koji se u tom trenutku nalazi na toj sekvenci. Programsko rješenje je prikazano na

slici 2.24. A ukupni prijeđeni put u tom ciklusu je zbroj pojedinačnih prijeđenih puteva pomnoženih s postotkom materijala u tom ciklusu (slika 2.25.).




**Slika 2.26.** *Proizvod se nalazi samo na jednoj sekvenci.*

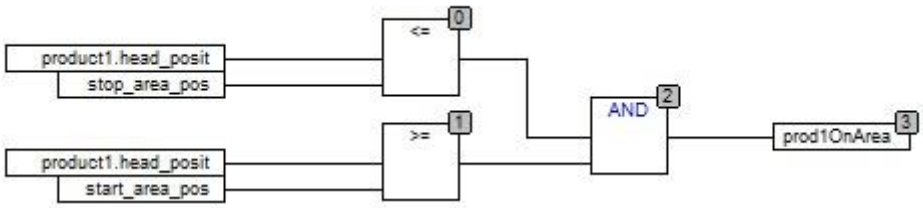
U drugom slučaju kada se proizvod nalazi na jednoj traci (slika 2.26.), za računanje se koristi samo pređeni put promatrane trake. Na kraju osvježavanje pozicije glave i repa u oba slučaja se vrši na isti način dodavanjem pređenog puta u svakom ciklusu kao što je prikazano na slici 2.27.




**Slika 2.27.** *Osvježavanje pozicije glave i repa.*

### 2.3.6 Zauzetost područja

Blok za ispitivanje zauzetosti područja ima dva ulaza, a to su početna i krajnja pozicija područja i pet izlaza koji označavaju nalazi li se proizvod unutar granica područja.

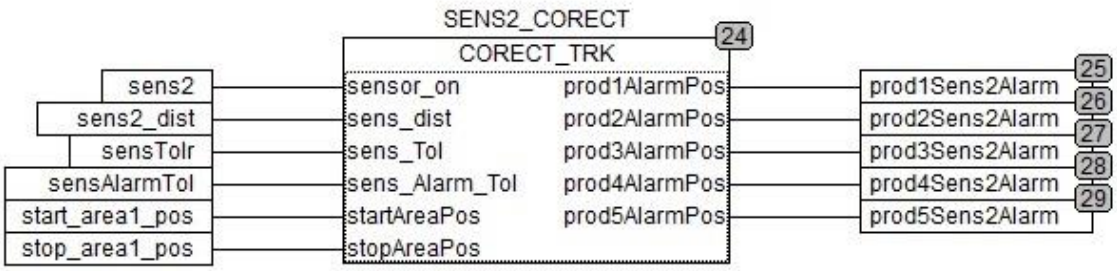



**Slika 2.28.** *Ispitivanje zauzetosti područja.*

Prilikom ispitivanja zauzetosti područja, blok provjerava poziciju glave određenog proizvoda u odnosu na ulazne varijable, te u izlaznu varijablu sprema rezultat provjere (slika 2.28.).

### 2.3.7 Sinkronizacija i korekcija pozicije

Za sinkronizaciju i korekciju pozicije se koriste foto senzori S2, S3 i S4. Korekcija pozicije se radi jer može doći do klizanja proizvoda zbog mase, izobličenja ili nasilnog zaustavljanja proizvoda iz nekog razloga. Blok za korekciju pozicije prikazan je slici 2.29.



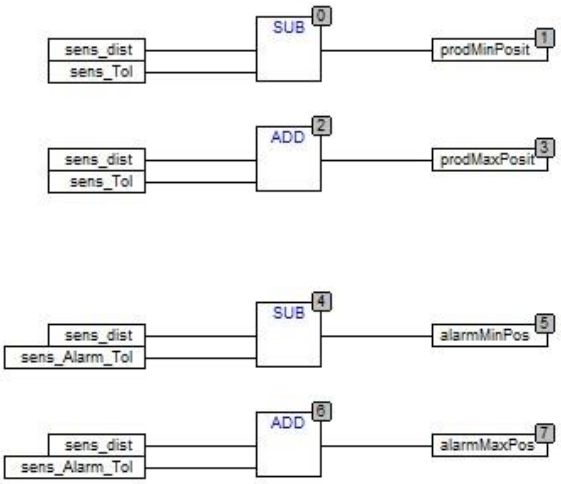

**Slika 2.29.** *Blok za korekciju pozicije proizvoda.*

Ulazne varijable bloka su:

- Stanje senzora (ON/OFF)
- Pozicija senzora
- Tolerancija senzora
- Tolerancija za uključenje alarma
- Početna pozicija područja
- Krajnja pozicija područja

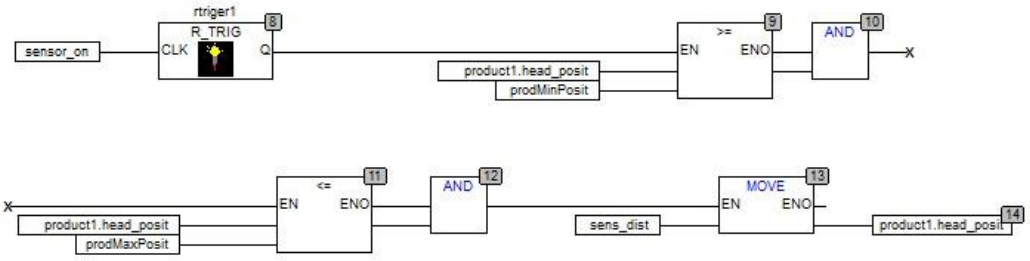
Izlazne varijable iz bloka su alarmi ukoliko se proizvod nalazi izvan granica tolerancije za alarm.

Senzor može biti u stanju nula ili jedan, odnosno detektira proizvod ili ne detektira proizvod. Pozicija senzora je udaljenost senzora od početka trake,a tolerancija senzora je vrijednost unutar koje se vrši korekcija proizvoda. Ako se proizvod pojavi na senzoru unutar tih vrijednosti tolerancije izvršit će se korekcija pozicije. Tolerancija za uključenje alarma je vrijednost unutar koje se uključuje alarm  ako se proizvod nalazi u tom intervalu vrijednosti. Početna i krajnja pozicija označavaju granice pojedinog područja.



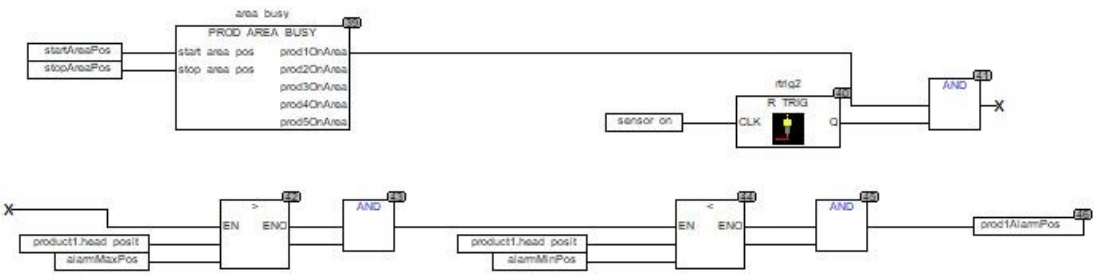

**Slika 2.30.** *Računanje tolerancija.*

Slika 2.30. prikazuje računanje minimalne i maksimalne tolerancije za korekciju pozicije proizvoda i računanje minimalne i maksimalne tolerancije alarma.




**Slika 2.31.** *Korekcija pozicije.*

Korekcija pozicije se izvršava nakon detekcije rastućeg brida na senzoru i nakon provjere uvjeta nalazi li se pozicija glave proizvoda unutar granica tolerancije. U koliko su ispunjeni uvjeti u poziciju glave se zapisuje pozicija senzora jer ako je proizvod aktivirao senzor tada se proizvod sigurno nalazi na poziciji senzora (slika 2.31.).

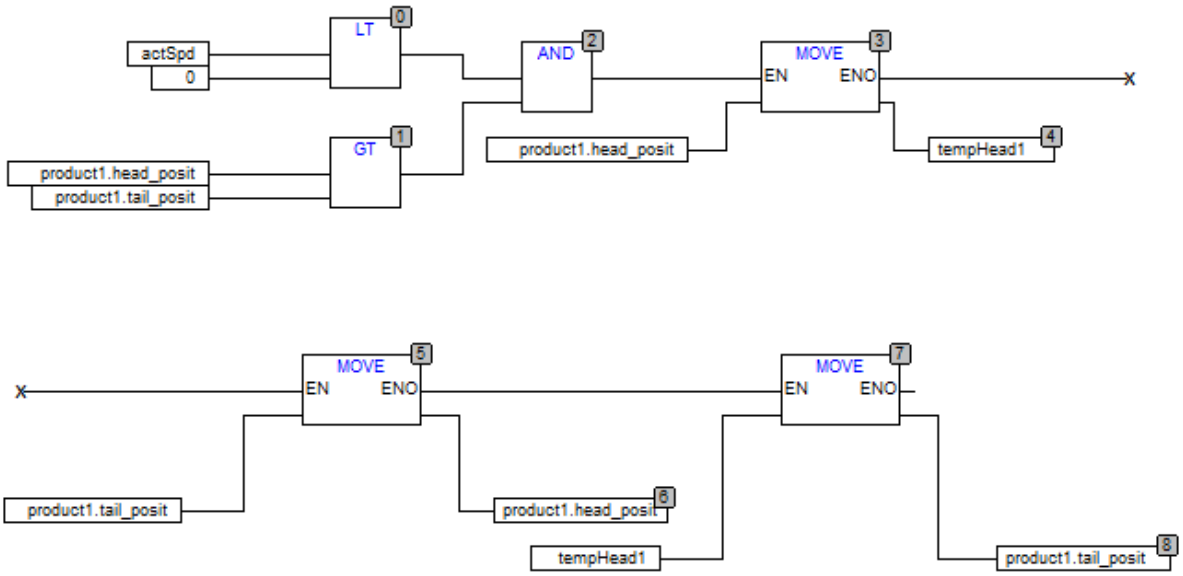



**Slika 2.32.** *Uključenje alarma za pojedini proizvod.*

Osim provjere nalazi li se pozicija glave unutar granica tolerancije za alarm vrši se i provjera nalazi li se proizvod na zadanom području. Ako su svi uvjeti ispunjeni postavlja se alarm za određeni proizvod u stanje jedan (slika 2.32.).

### 2.3.8 Promjena smjera proizvoda

U slučaju kada se odabere manualni način upravljanja trakom moguće je vrtjeti traku u oba smjera. Bitno je naglasiti da se smjer s lijeva na desno smatra smjerom vrtnje naprijed, a smjer s desna na lijevo se smatra smjerom vrtnje unazad. Zbog promjene smjera vrtnje bilo je potrebno napraviti blok koji će ispitivati na koju stranu se vrti traka i shodno tome ako traka promjeni smjer vrtnje pozicija glave postaje pozicija repa i pozicija repa postaje pozicija glave.

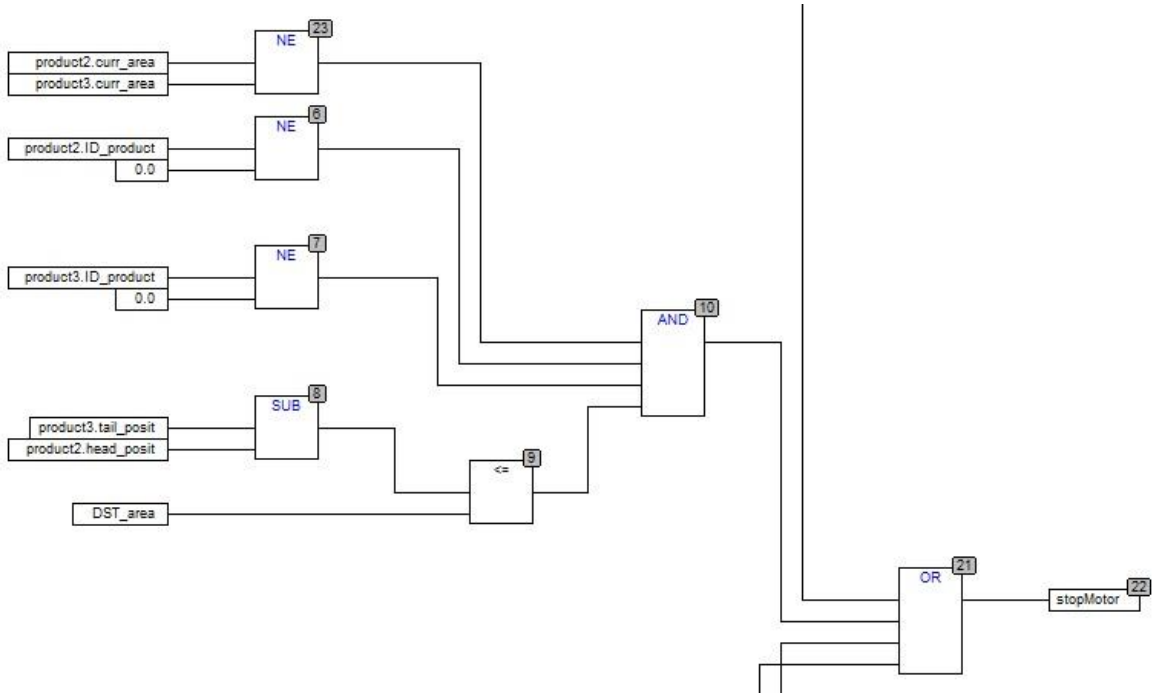



**Slika 2.33.** *Promjena pozicije glave i repa.*

U primjeru sa slike 2.33. vrši se promjena glave i repa ako je došlo do promjene smjera vrtnje, gdje je prvotni smjer bio s lijeva na desno. Do promjene će doći kada su zadovoljeni sljedeći uvjeti: brzina je manja od nula (smjer vrtnje s desna na lijevo) i kada je pozicija glave veća od pozicije repa. Promjena pozicije glave i repa je u biti promjena vrijednosti koje se nalaze u varijablama glava i rep, a promjena se vrši pomoću privremene varijable koja je u ovom slučaju *tempHead1*. Nakon ispunjavanja navedenih uvjeta vrijednost koja se nalazi u poziciji glave se sprema u privremenu varijablu, zatim se u poziciju glave sprema vrijednost koja se nalazi u varijabli pozicije repa i na kraju se u poziciju repa sprema vrijednost koja se nalazi u privremenoj varijabli.

### 2.3.9 Udaljenost između proizvoda

Sustav nadzora osim o praćenju proizvoda treba brinuti i o samim proizvodima. Stoga je potrebno u slučaju kvara ili prestanka rada jedne od traka voditi računa da se proizvod koji dolazi s prethodne trake ne sudari s proizvodom na traci koja je iz nekog razloga prestala raditi. Da bi se to izbjeglo napravljen je blok koji u svakom ciklusu provjerava kolika je udaljenost između dva susjedna bloka koja se nalaze na različitim trakama, te se ta vrijednost uspoređuje s DTS prethodne trake. Odnosno provjerava se, hoće li prethodna traka morati početi kočiti i zaustavljati se, te u kojem trenutku će se početi zaustavljati kako se proizvod koji se nalazi na toj traci ne bi sudario s proizvodom koji se nalazi na traci koja je stala.

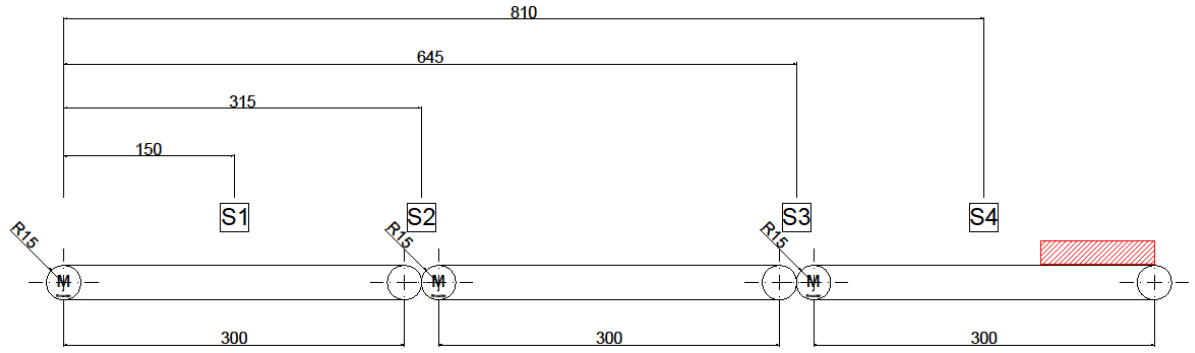



**Slika 2.34.** *Provjera udaljenosti između proizvoda.*

U primjeru sa slike 2.34. se provjerava udaljenost između proizvoda koji se nalazi u spremniku *product2* i spremniku *product3.* Prvo je potrebno provjeriti nalaze li se proizvodi na istoj traci, zatim da li je ID oba proizvoda različit od nula, odnosno postoje li proizvodi u tim spremnicima. Na kraju se od pozicije repa proizvoda u spremniku *product3* oduzima pozicija glave proizvoda u spremniku *product2* i uspoređuje se DTS trake na kojoj se nalazi proizvod spremljen u spremnik *product2.* U koliko je za bilo koja dva susjedna proizvoda ispunjen uvjet, čitava traka se zaustavlja.

### 2.3.10 Odlaganje proizvoda

Odlaganje proizvoda zamišljeno je kao zaustavljanje trake i proizvoda u trenutku kada je pozicija glave jednaka duljini trake kao na slici 2.35.

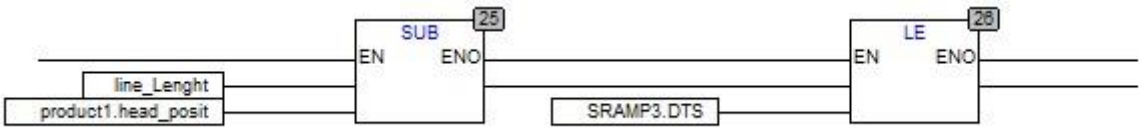



**Slika 2.35.** *Prikaz proizvoda na poziciji za odlaganje.*

Odlaganje proizvoda s trake se može podijeliti u tri djela:

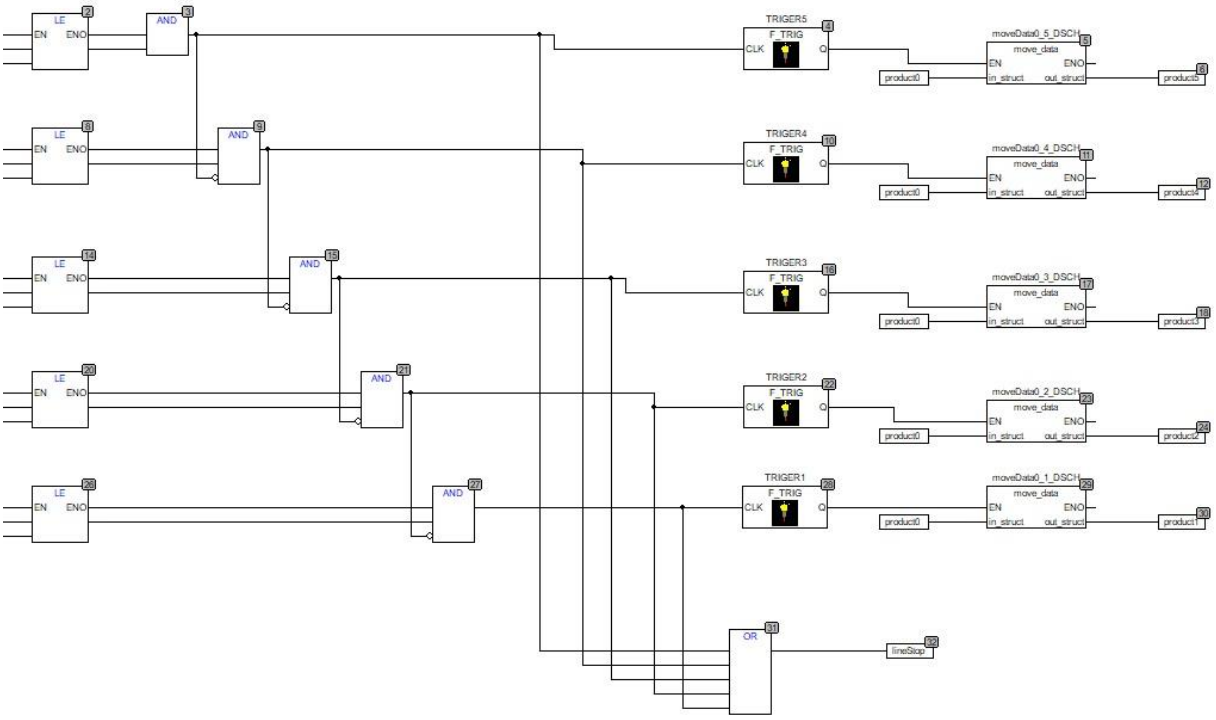
- Ispitivanje prisutnosti proizvoda na području za odlaganje
- Provjera uvjeta za zaustavljanje trake
- Zaustavljanje trake i brisanje proizvoda iz spremnika

Ispitivanje prisutnosti proizvoda na području za odlaganje se vrši pomoću bloka za provjeru zauzetosti područja.




**Slika 2.36.** *Ispitivanje uvjeta za zaustavljanje i odlaganje proizvoda.*

Potom se ispituju uvjeti za zaustavljanje trake. Uspoređuje se preostali put koji se dobije oduzimanjem pozicije glave od duljine linije s putem koji će traka prijeći ako u tom trenutku krene kočiti. Ispitivanje ovih uvjeta je prikazano na slici 2.36.




**Slika 2.37.** *Odlaganje proizvoda, zaustavljanje trake i brisanje iz spremnika.*

Do kočenja će doći u trenutku kada je preostala udaljenost između glave proizvoda i duljine linije manja ili jednaka od udaljenosti koju će traka prijeći ako u tom trenutku krene kočiti (DTS). Proizvod se briše iz spremnika kada se traka zaustavi, odnosno kada DTS bude jednak nuli (slika 2.37.).

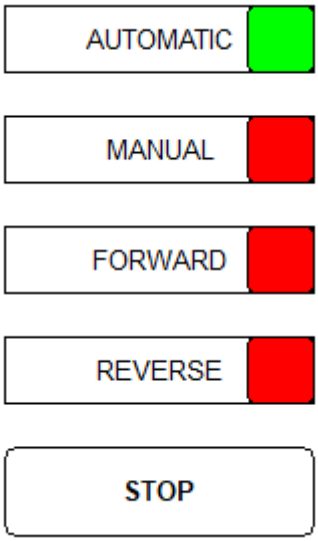
# 3 VIZUALIZACIJA MAKETE

Programsko okruženje CoDeSys osim simulacije omogućava i kreiranje vizualizacije. Za prikaz vizualizacije CoDeSys omogućava korištenje različitih geometrijskih likova: pravokutnika, pravokutnika sa zaobljenim bridovima, kružnih i eliptičnih likova, također je dozvoljeno i korištenje drugih oblika koji ne moraju biti pravilni. Osim za vizualizaciju može se koristiti i kao upravljačko sučelje koje je poznatije pod nazivom HMI (eng. *Human-Machine Interface*).

HMI makete pokretne trake sadrži sljedeće elemente:

- Kontrola trake
- Kontrola senzora
- Prikaz motora
- Model trake
- Informacije o proizvodu

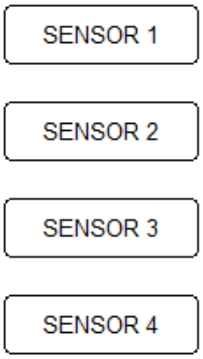
Sustav upravljanja trakom sastoji se od pet elemenata, odnosno gumba. Traka može biti u ručnom i automatskom načinu rada. U koliko odaberemo neki od načina rada pokraj navedenih gumba se nalazi indikator koji je u normalnom stanju crvene boje, te kada kliknemo na jedan od gumba indikator će promijeniti boju u zeleno. Sve naredbe za kontrolu trake su *bool* tipa.




**Slika 3.1.** *Kontrola trake.*

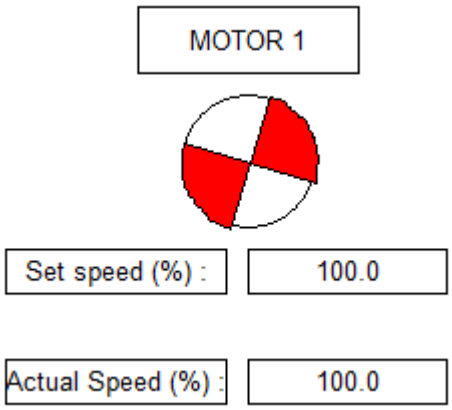
Slika 3.1. prikazuje primjer kada je uključen automatski način rada trake. Kada se želi trakom upravljati ručno prvo je potrebno uključiti ručni način rada, potom se odabire smjer vrtnje trake

naprijed ili nazad. Sustav upravljanja trakom je postavljen tako da će se traka iz sigurnosnih i praktičnih razloga kretati naprijed ili nazad samo dok korisnik drži stisnut gumb za naprijed ili nazad.



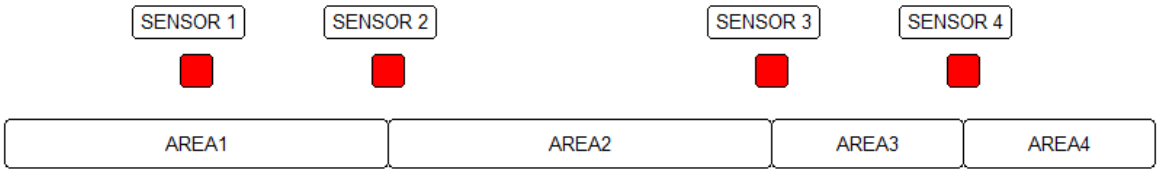

**Slika 3.2.** *Kontrola senzora.*

S obzirom na to da se radi o simulaciji, u HMI-u se nalaze gumbi koji će simulirati rad foto optičkih senzora koji se nalaze na traci. Senzori su bitni za simulaciju jer se na njima radi sinkronizacija podataka o nadzoru proizvoda (slika 3.2.).




**Slika 3.3.** *Prikaz motora.*

U HMI-u se nalaze podatci zadane i trenutne brzine za svaki motor. CoDeSys također omogućava i animacije pa je pokretanjem simulacije i uključenjem trake moguće simulirati rad motora (slika 3.3.).




**Slika 3.4.** *Model pokretne trake.*

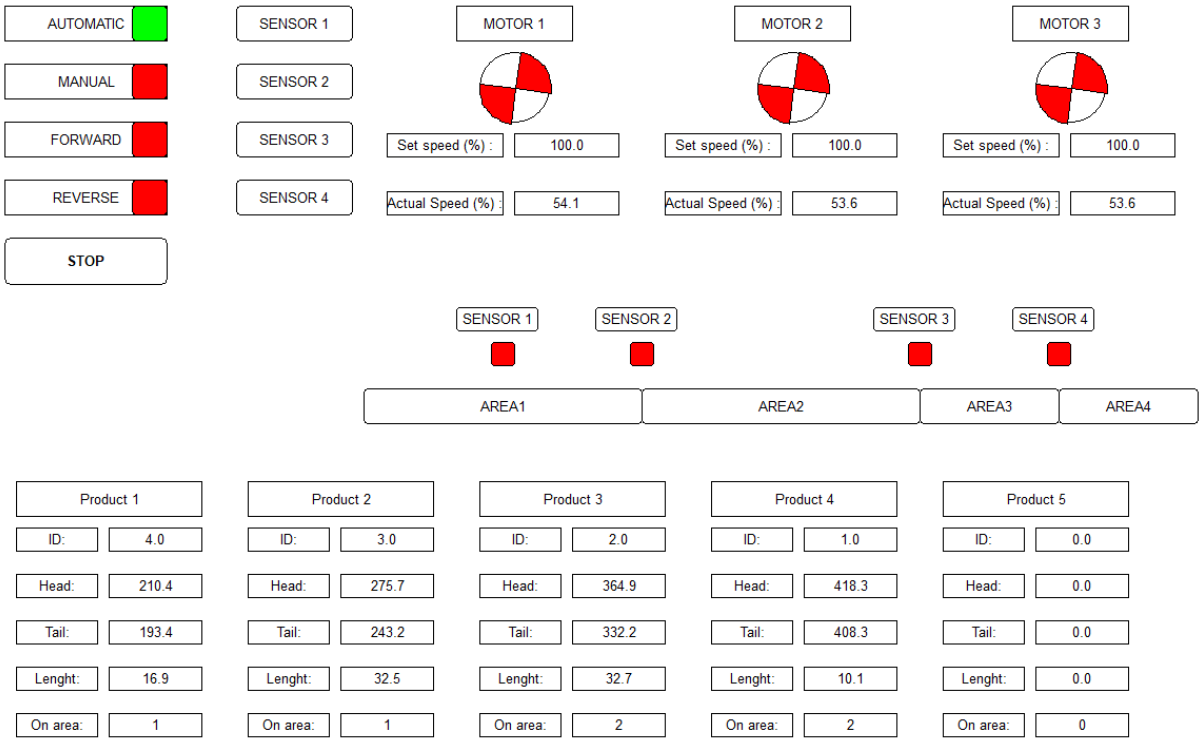
Kao što je navedeno u prethodnim poglavljima maketa se sastoji od tri motora koji pokreću tri zasebne trake. Zbog lakšeg strukturiranja i organiziranja programa traka je podijeljena u četiri područja gdje prvo i drugo područje odgovaraju prvoj i drugoj traci makete, a treća traka podijeljena je u dva djela (slika 3.4.). Područje četiri se koristi samo za odlaganje materijala.

| Product 1 | |
|---|---|
| ID: | 2.0 |
| Head: | 314.6 |
| Tail: | 284.0 |
| Lenght: | 30.6 |
| On area: | 1 |

| Product 2 | |
|---|---|
| ID: | 1.0 |
| Head: | 397.2 |
| Tail: | 378.6 |
| Lenght: | 18.7 |
| On area: | 2 |

**Slika 3.5.** *Prikaz proizvoda na traki.*

Svakom proizvodu koji dođe na traku se dodjeljuju podatci o ID-u, poziciji glave i repa, duljini i informacija na kojem području se proizvod trenutno nalazi. Sa slike je još vidljiv i primjer organiziranja spremnika kao FIFO stoga. To se može primijetiti prema ID-u proizvoda gdje je proizvod koji je došao prvi dobio ID = 1 i dolaskom novog proizvoda on se premješta u spremnik *product2*, a u prvi spremnik se spremaju podatci od proizvoda koji je došao poslije njega.




**Slika 3.6.** *HMI sučelje.*

Na HMI sučelju se nalaze komande za upravljanje trakom, gumbi za simuliranje rada senzora, animacija rada motora s prikazanom zadanom i trenutnom brzinom, prikaz programske podjele trake u četiri područja i na kraju nalaze se informacije o proizvodima.

Slika 3.6. prikazuje HMI sučelja kada je uključen automatski način rada. Na traci se nalaze četiri proizvoda od kojih su dva na području jedan, a druga dva su na području dva.

# 4 OPIS KOMPONENATA I SUSTAVA ZA IZRADU MAKETE

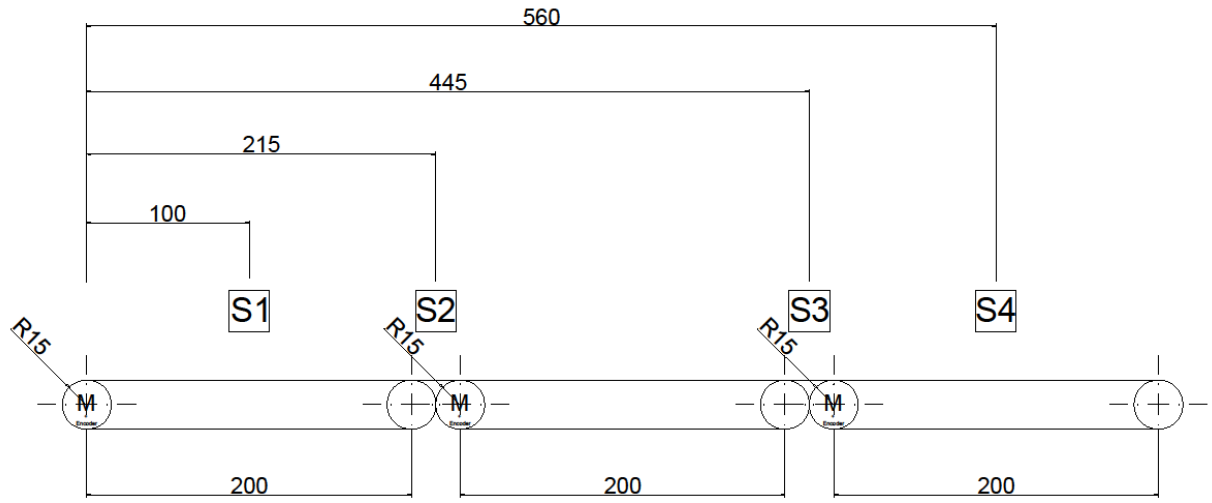
Kroz ovo poglavlje opisane su dimenzije i izgled makete, prikazano je i programsko rješenje za upravljanje maketom. Navedene su i osnovne značajke programabilnog logičkog kontrolera *Simatic S7-400*. Ukratko je opisan način korištenja programskih jezika u programskom okruženju *Simatic Manager*. Uz to nalazi se i prikaz komponenata koje je bilo potrebno osmisliti i izraditi kako bi bilo moguće realizirati maketu.

## 4.1 Opis makete pokretne trake

Potrebno je napraviti maketu pokretne trake koja se sastoji od tri zasebne sekvence, gdje svakom sekvencom upravlja DC motor s enkoderom i na odgovarajućim pozicijama se nalaze foto senzori. Trakom se upravlja preko upravljačke ploče (eng. *Industrial keyboard*).

Ukupna dužina trake je 660 mm i podijeljena je u tri grupe  kako je prikazano na slici 4.1. Dimenzije sekvenci:

- Prva sekvenca trake 0 – 215 mm
- Druga sekvenca trake 215 -445 mm
- Treća sekvenca trake 445-660 mm



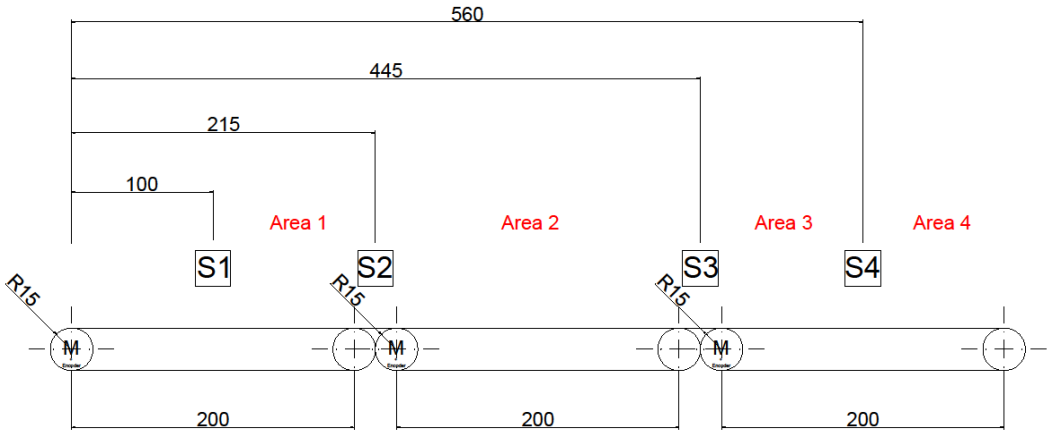

**Slika 4.1.** *Bočni prikaz makete pokretne trake.*

Za praćenje i sinkronizaciju pozicije materijala se koriste četiri fotosenzora raspoređena kao na slici 4.1.

Udaljenosti senzora u odnosu na početak trake su sljedeće:

- Senzor S1: 100 mm
- Senzor S2: 215 mm
- Senzor S3: 445 mm
- Senzor S4: 560 mm

Programski je traka podijeljena na četiri područja kao što je prikazano na slici 4.2.



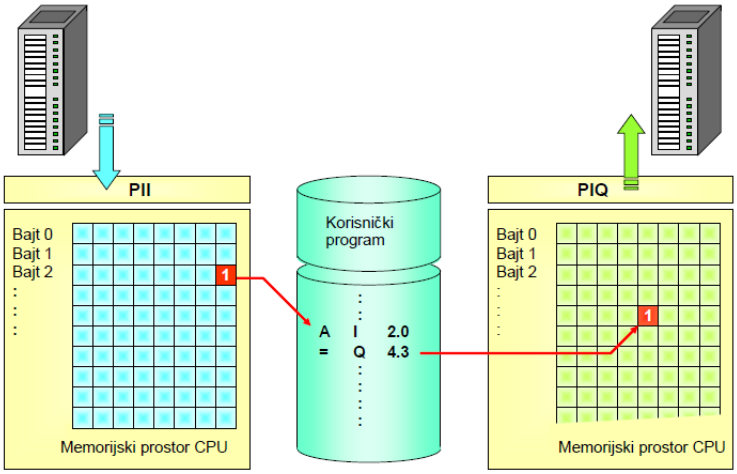

**Slika 4.2.** *Prikaz područja na maketi pokretne trake.*

Na senzoru S1 proizvodu se dodjeljuju inicijalni podatci: ID, duljina, pozicija glave, pozicija repa i područje na kojem se nalazi proizvod. Senzori S2, S3 i S4 služe za sinkronizaciju podataka o proizvodu.

Na područjima od jedan do tri se vrši praćenje i sinkronizacija podataka o proizvodu, a područje četiri osim za praćenje služi i za odlaganje proizvoda.

## 4.2 Simatic S7-400

Siemensov programabilni logički upravljač (PLC) koji se koristi za automatizaciju i upravljanje služi za direktno ili indirektno upravljanje motorima, pumpama i drugim uređajima koji se koriste u nekom procesu. Osnovne cjeline PLC-a su: ulazni i izlazni moduli, procesor i memorija. Princip rada temelji se na cikličkom čitanju ulaznih modula, obradi ulaznih podataka u programu i postavljanju odgovarajućih naredbi na izlaznim modulima.[1]




**Slika 4.3.** *Blokovski prikaz čitanja i pisanja u procesu. [1]*

U odnosu na releje i sklopke koji su se u prošlosti koristili za automatizaciju i upravljanje, PLC-ovi su višestruko jednostavniji za prilagodbu i izmjene. Zbog toga i sve većih tehnoloških zahtjeva u procesu proizvodnje upotreba PLC-ova u industriji je u porastu i kao takvi prave se još funkcionalniji i otporniji na teške uvjete rada.[1]

Prilikom izrade makete korišten je *Simatic S7-400* PLC koji se sastoji od:

- Postolja (eng. *Rack*) na koje se instaliraju moduli
- Napajanja (eng. *Power supply*)
- Procesora (eng. *CPU*)
- Komunikacijskog procesora (eng. *Communication processor*) čija je uloga komunikacija PLC-a s računalom preko ethernet kabela

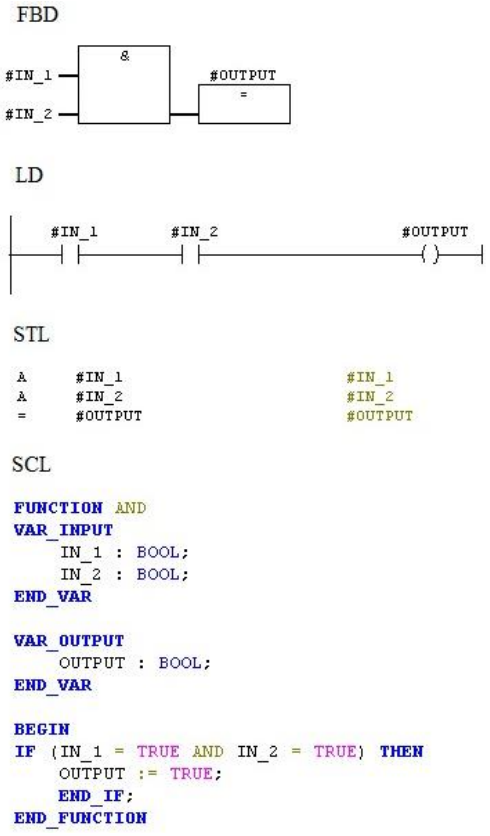
Uz navedene komponente još se koristi ET200S modul koji pomoću PROFIBUS-a komunicira s PLC-om. ET200S se koristi za dodavanje odgovarajućih kartica (AI, AO, DI, DO, Count, PS i ostalih modula).

## 4.3 Simatic Manager

*Simatic Manager* je programsko okruženje koje je razvio *Siemens*. Služi za konfiguriranje i programiranje PLC-ova. *Simatic Manager* omogućava programiranje u više programski jezika, a to su:

- Funkcijski blokovski dijagram koji se označava kao FBD (eng. *Function Block Diagram*)
- Ljestvičasti dijagram koji se označava kao LD (eng. *Ladder Diagram*)
- Instrukcijska lista koja se označava kao STL (eng. *Statement List*)

Osim navedenih programski jezika *Simatic Manager* dozvoljava i programiranje u višem programskom jeziku koji se naziva SCL (eng. *Structured Control Language*).[1]




**Slika 4.4.** *Primjer programskih jezika.*

U navedenom primjeru se vidi izvedba I logičkog bloka napisanog u različitim programskim jezicima. Blok vrši logičku I operaciju na osnovu ulaza IN_1 i IN_2, te sprema izlaz u varijablu OUTPUT. Ukoliko su i jedna i druga ulazna varijabla u stanju logičke jedinice na izlazu iz

bloka će se također pojaviti logička jedinica koja se sprema u varijablu dovedenu na izlaz logičkog bloka, u ovom primjeru je to varijabla OUTPUT.

FBD koristi blokovski prikaz, gdje se na ulaze dovode varijable, odnosno signali i na osnovu logičke ili matematičke funkcije koju blok obavlja sprema izlaz u zadanu varijablu.

LD se zasniva na drugačijem pristupu u odnosu na FBD i koristi se takozvani ljestvičasti dijagram. Bitno je napomenuti da se u LD-u program izvršava s lijeva na desno i odozgor prema dolje. Elementi se povezuju kao električna shema.

STL i SCL su programski jezici u kojima se koriste tekstualne naredbe, osim razlike u sintaksi STL i SCL se razlikuju u tome jer je SCL viši programski jezik u odnosu na STL.

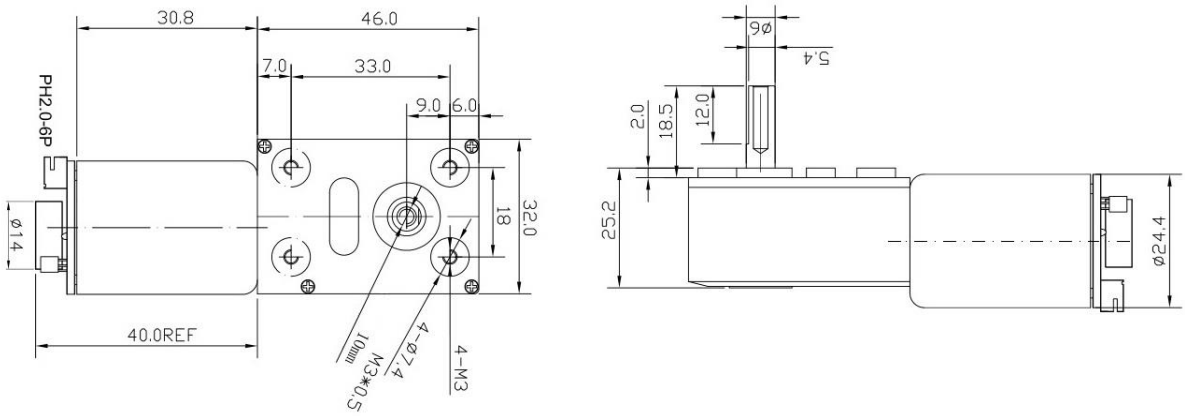
## 4.4 Osnovni elementi makete

Neki od osnovnih elemenata za izradu makete pokretne trake su:

- Motor s integriranim reduktorom i enkoderom
- Laserski senzor (infracrveni senzor s neprekinutom zrakom)
- Valjci s osovinama
- Zatezni valjci
- Postolje

Za pogon sekvenci se koriste 12 *V* istosmjerni motor bez četkica s integriranim reduktorom (eng. *gear box*) i enkoderom (slika 4.5). Maksimalna struja motora je 600 *mA* i maksimalna snaga je 2.85 *W*. Prijenosni omjer reduktora je 1:150. Reduktor na svojoj osovini daje 40 *RPM* (eng. *Rotation Per Minut*), odnosno 40 obrtaja u minuti.

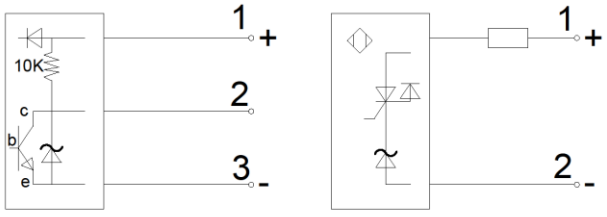
Enkoder ima dvije faze, odnosno mogućnost mjerenja pomaka i određivanja smjera u ovisnosti od redoslijeda paljenja faza. Signali na fazama se registriraju uz pomoć *hall* senzora i magneta koji se nalazi na osovini motora (slika 4.6.). Enkoder je spojen direktno na osovinu motora i ima rezoluciju 1650 PPR (eng. *Pulse Per Revolution*) za jedan okret osovine reduktora. Uzimajući u obzir omjera reduktora 1:150 dobije se da enkoder daje 11 impulsa za jedan okretaj osovine motora.




**Slika 4.5.** *Tlocrt i bokocrt istosmjernog motora s reduktorom i enkoderom. [5]*

**Slika 4.6.** *Prikaz ožičavanja motora i enkodera. [5]*

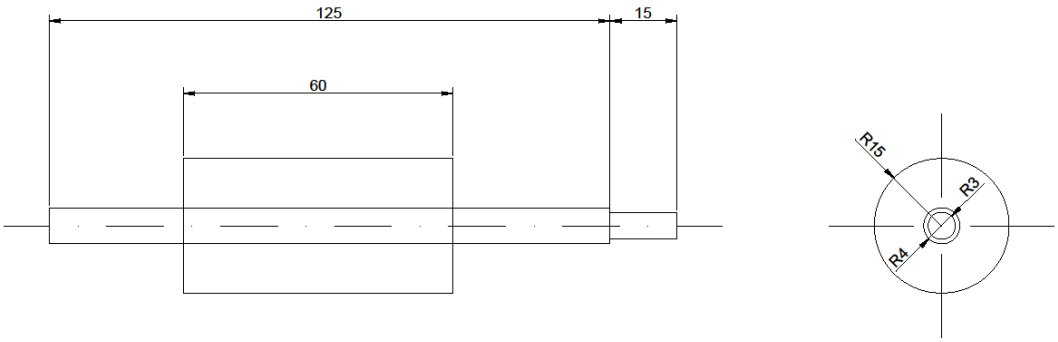
Za inicijaliziranje proizvoda i sinkronizaciju pozicije proizvoda koriste se infracrveni senzori s neprekinutim snopom. Senzor se sastoji od dva uređaja: predajnik i prijemnik. Predajnik emitira infracrvenu svjetlost, dok prijemnik detektira tu svjetlost i na upravljačkom izvodu daje logičku jedinicu. Kada je zraka prekinuta na upravljačkom izvodu će se pojavi logička nula. Izgled elemenata prikazan je na slici 4.7.




**Slika 4.7.** *Blokovski prikaz prijemnika i predajnika.*

Da bi bilo moguće realizirati pokretnu traku potrebno je napraviti odgovarajuću osovinu s fiksnim valjkom koja će biti pogonjena motorom. Bitno je napomenuti da je osovina reduktora spojena fleksibilnom vezom s osovinom na kojoj se nalazi valjak za pogon trake. Za potrebe

makete napravljene su tri osovine i tri valjka sljedećih dimenzija, kao što je prikazano na slici 4.8.

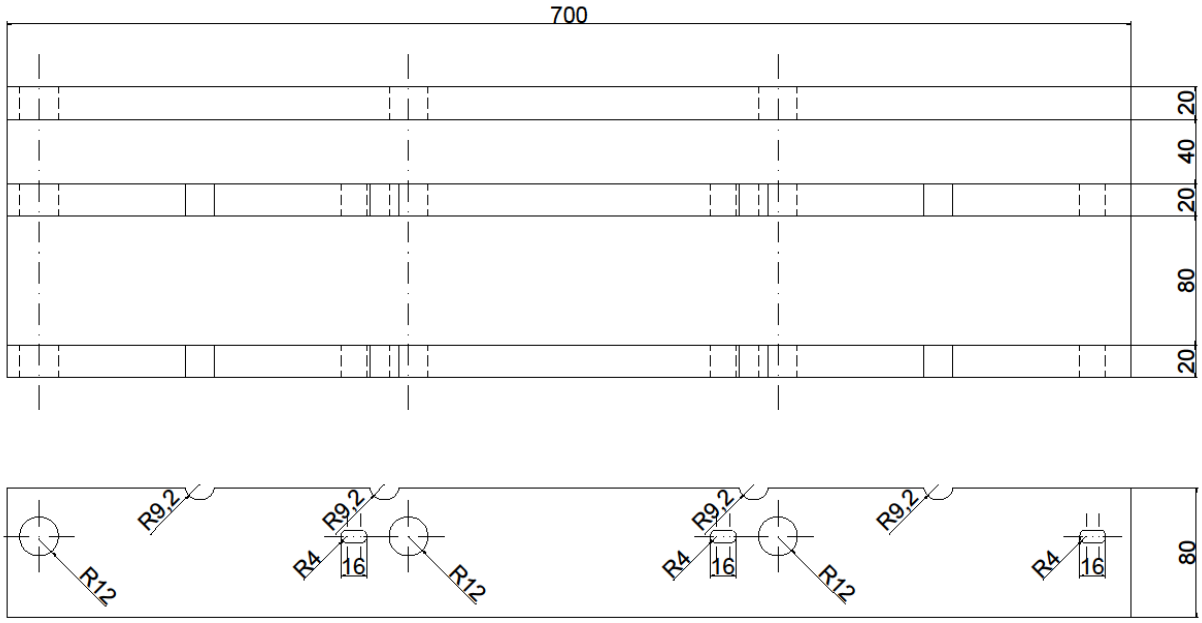



**Slika 4.8.** *Bokocrt i nacrt osovine s valjkom.*

Za dodatnu fleksibilnost, otpornost i prilagodljivost različitim masama materijala za prijenos na drugom kraju trake nalazi se valjak koji nije fiksiran na osovinu. Osim toga da bi se osigurala odgovarajuća napetost trake i izbjeglo proklizavanje između trake i pogonskog valjka druga osovina je zakačena na opruge koje osiguravaju odgovarajuću napetost trake. Samim time druga osovina nije fiksno postavljena na kućište nego se može pomjerati (slika 4.9.).

**Slika 4.9.** *Postizanje napetosti trake pomoću opruga.*

Da bi se dobila konačna maketa sve elemente je potrebno postaviti na odgovarajuća mjesta u postolju. Postolje je izrađeno od drveta zbog jednostavnije tehnološke obrade u odnosu na metal. Dimenzije postolja su prikazane na slici 4.10.

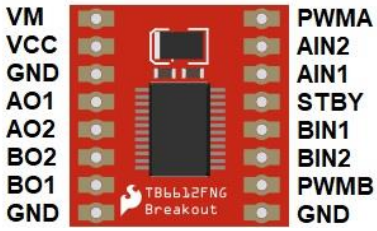



**Slika 4.10.** *Prikaz postolja za maketu pokretne trake.*

## 4.5 Projektiranje i izrada sklopa za upravljanje motorima

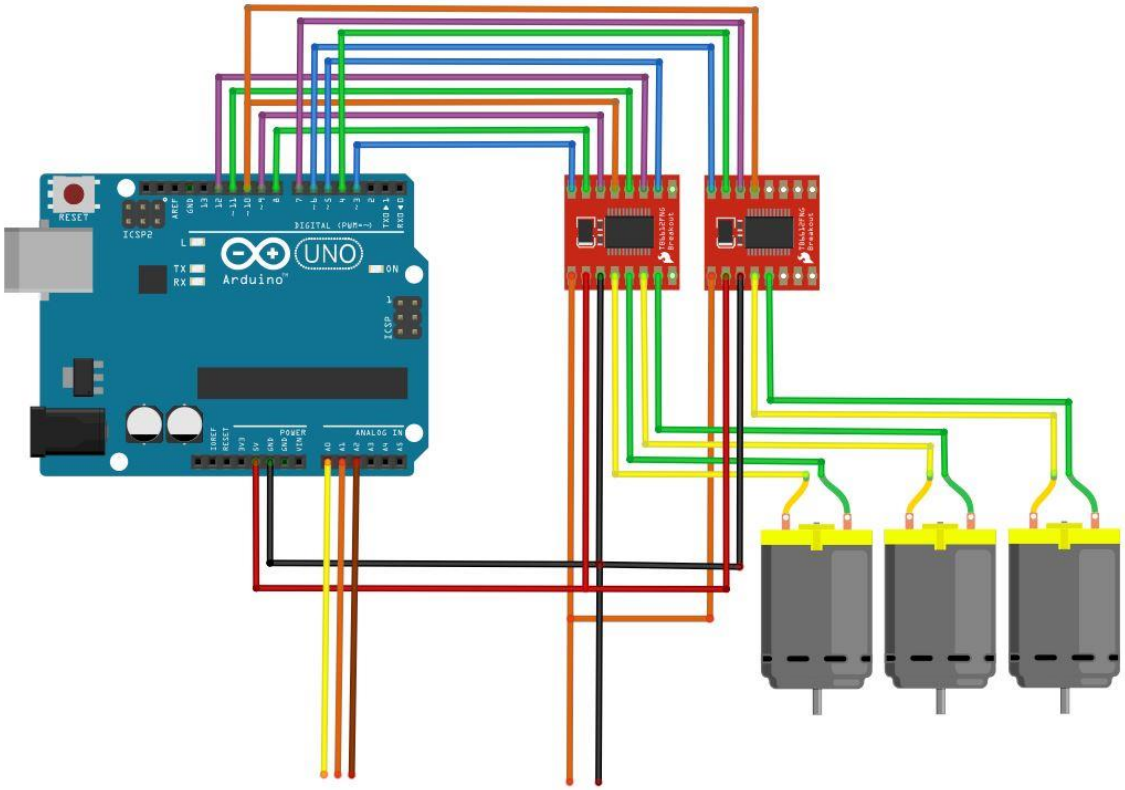
Iako se radi o motorima koji imaju malu snagu $P = 2.85\ W$, a samim time i male struje $I_{max} = 600\ mA$, analogna izlazna kartica AO (eng. *Analog output*) ne dozvoljava direktno spajanje motora na karticu zbog svoje strujne zaštite koja iznosi $I_{AO_{max}} = 75\ mA$. Zbog toga je bilo potrebno napraviti sklop koji će moći vršiti galvansko odvajanje upravljačkog (u ovom slučaju kruga koji se zatvara s karticom AO) i energetskog kruga (napajanje motora koje je u ovom slučaju ±12 *V*). Jedno od rješenja se dobije upotrebom *Arduino Uno* mikroupravljača i integrirane pločice koja omogućava upravljanje s dva DC motora ili jednim koračnim motorom, poznata je još i pod nazivom TB6612FNG kartica.

Ideja je da *Arduino* prima analogni signal od AO kartice, zatim ga pretvara u pulsno širinski modulirane PWM (eng. *Pulse-width modulation*) signale i kao takvog ga šalje na TB6612FNG karticu (slika 4.11.).




**Slika 4.11**. *Arduino sklop za upravljanje s dva istosmjerna ili jednim koračnim motorm.*

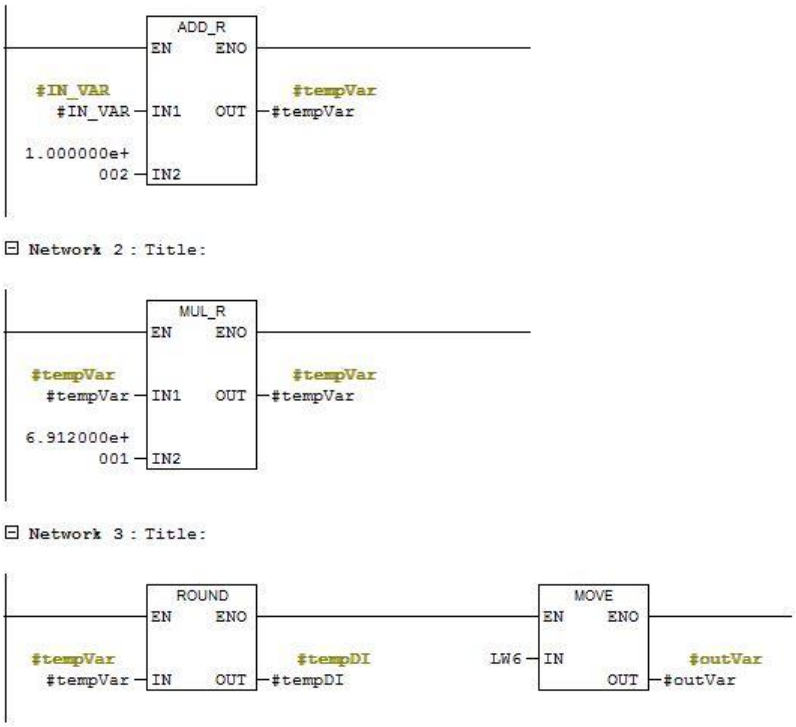
Izgled sklopa za upravljanje istosmjernim motorima koji je realiziran pomoću Arduina i dvije integrirane pločice TB6612FNG prikazan je na slici 4.12.



**Slika 4.12.** *Sklop za upravljanje motorima.*

Shema sklopa za upravljanje motorima je napravljena u programu *Fritzing Beta*. Iz sheme (slika 4.12.) je vidljivo da na analogne ulaze A0, A1 i A2 dolaze tri upravljačka signala iz PLC-a. Potrebno je uočiti da se na integriranoj pločici TB6612FNG nalaze dva napajanja (slika 4.11.) od kojih je jedno napajanje same pločice i iznosi $U_{CC} = 5\ V$, a drugo je vanjsko napajanje motora $U_M = 12\ V$.

U radu je korištena 6ES7 135-4FB01-0AB0 analogna izlazna kartica koja može dati dva analogna signala u rasponu od ± 10 *V*, te kao što je prethodno navedeno *Arduino* prima od PLC-a analogni signal i nad njima vrši PWM modulaciju. Zbog samih karakteristika analognih ulaza mikroupravljača *Arduino Uno* koji rade u rasponu od 0 V do 5 V bilo je potrebno napraviti funkcijski blok za reskaliranje napona AO kartice. Pošto raspon vrijednosti ± 10 *V* odgovara brojčanoj vrijednosti od ± 27648 i kao što je unaprijed dogovoreno brzina će biti prikazivana u postotcima, gdje 100% brzine odgovara nazivnoj brzini motora koja je 40 *RPM* blok za prilagodbu izgleda kao na slici 4.13.



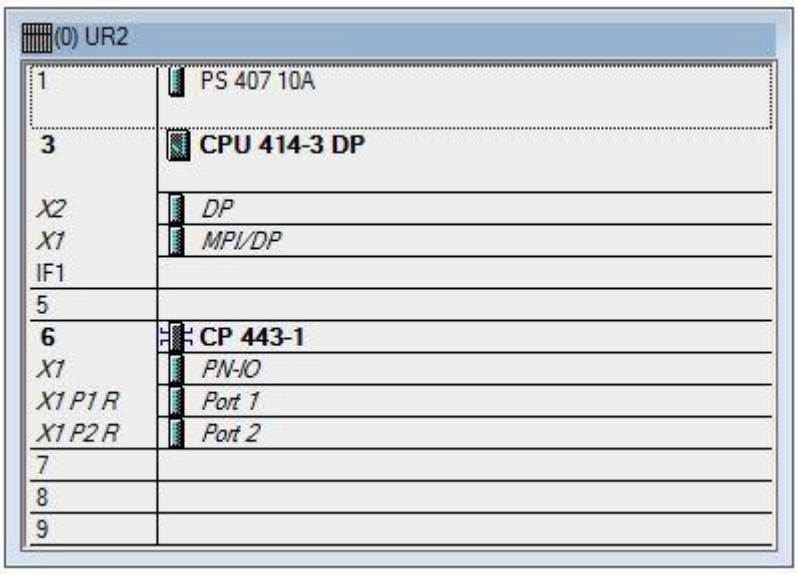

**Slika 4.13.** *Reskaliranje vrijednosti*.

Odnosno prilagodba se vrši prema jednadžbi (4-1).

$$y = 69{,}12 * (x + 100) \tag{4-1}$$

## 4.6 Kreiranje hardverske konfiguracije

Da bi se mogla uspostaviti veza između računala i PLC-a potrebno je u projektu napraviti hardversku konfiguraciju. Konfiguracija se kreira otvaranjem prozora *HW Config*, dohvaćanjem odgovarajući elemenata i slaganjem u predviđeni dio prozora *HW Config*. Elementi se dohvaćaju i slažu na odgovarajuća mjesta pomoću jedinstvenih identifikacijski brojeva otisnutih na samim elementima koji se potom unose u tražilicu prozora *HW Config* (slika 4.14.).



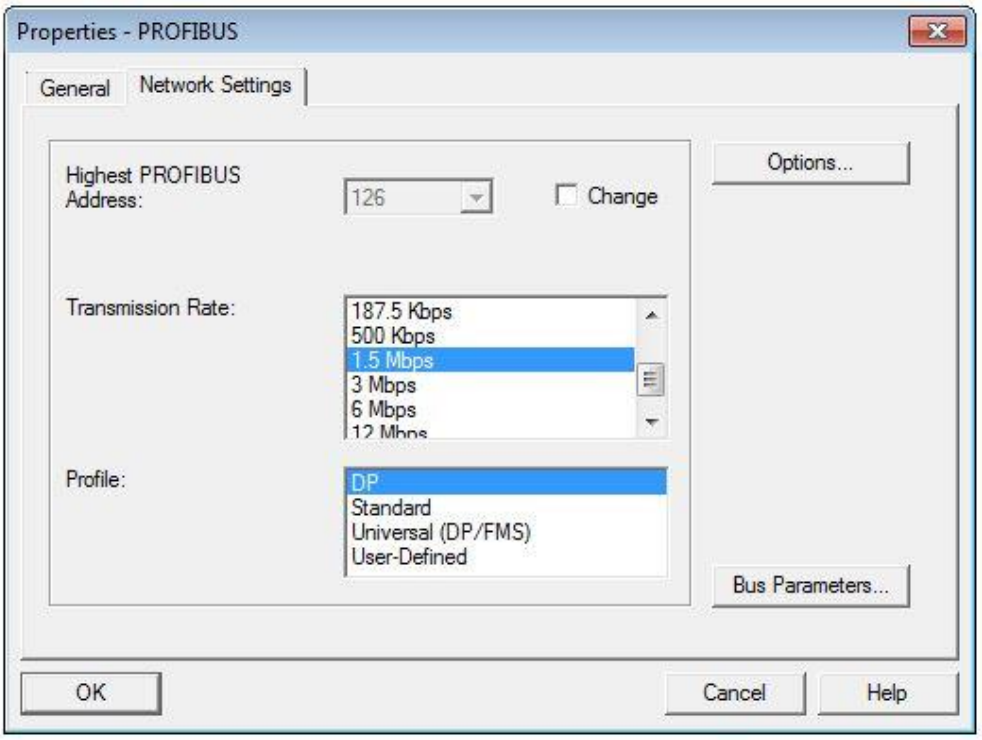

**Slika 4.14.** *Hardverska konfiguracija PLC-a.*

Na slici 4.14. je prikazana hardverska konfiguracija PLC-a koja sadrži:

- Postolje (eng. *Rack*)
- Napajanje (eng. *Power supply*)
- Procesor 414-3 DP
- Komunikacijski procesor 443-1

Na postolju se nalazi devet utora za dodavanje modula kao što je i prikazano na slici 4.14. Napajanje je spojeno na prva dva utora i osim pretvorbe izmjeničnog napona 230 *V* na 24 *V* i 5 *V* može sadržavati i jednu ili dvije baterije koje služe kao zaštita za pohranu podataka prilikom nestanka izmjeničnog napajanja. U ovom PLC-u se koristi procesor serije 414-3 s dva komunikacijska sučelja označena kao *X1* i *X2* i dodatno je opremljen je s 4 *MB* SRAM memorije. Komunikacijski procesor ima 2 ethernet porta za komunikaciju s računalom.

Da bi bilo moguće koristiti analogne i digitalne ulaze ili izlaze, te brze brojače i bilo koje druge module procesor je pomoću PROFIBUS-a spojen na ET200S modul. Brzinu prijenosa podataka PROFIBUS mrežom je moguće birati i ona se izabire iz odgovarajuće tablice ovisno o duljini komunikacijskog kabela, na primjer maksimalna brzina od 12 Mbps je moguća na udaljenosti do 100 m. Vrsta profila je standardno postavljena na DP (eng. *Decentralised Peripherals*) također je i brzina standardno postavljena na 1.5 *Mbps* (slika 4.15.).




**Slika 4.15.** *Konfiguriranje PROFIBUS komunikacije.*

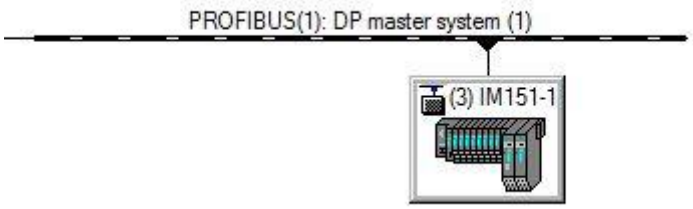
ET200S modul se može promatrati kao postolje na koje se pomoću odgovarajućeg utora spajaju različite kartice. Za izradu makete su korištene sljedeće kartice:

- Napajanje DC 24 V
- Brojač 24 V 100 kHz
- Analogne izlazne kartice
- Digitalne ulazne kartice
- Digitalne izlazne kartice

Kartica napajanja osigurava napajanje za ostale module. Brojač omogućava registriranje brzih promjena radi na napajanju 24 V i može pratiti promjene s frekvencijama manjim od 100 kHz. U radu se koristi kao brojač impulsa na enkoderu. Analogne izlazne kartice korištene u radu

daju izlazni analogni signal u rasponu ± 10 V. Digitalne ulazne kartice služe za čitanje ulaznih digitalnih signala, mogu imati dva, četiri ili osam ulaza. Digitalne izlazne kartice daju na izlazu digitalni signal i mogu imati dva, četiri ili osam izlaza.

Nakon što se u *HW Config*-u napravi konfiguracija PLC-a i odgovarajućeg PROFIBUS-a u programu se dodaje ET200S modul koji se također pronalazi po svom jedinstvenom broju (*slika 4.16.*).




**Slika 4.16.** *ET200S modul.*

| Slot | Module | Order Number | I Address | Q Address | Diagnostic address | Comment |
|---|---|---|---|---|---|---|
| 1 | PM-E DC24V | 6ES7 138-4CA01-0AA0 | 8182* | | 8182* | |
| 2 | 1 COUNT 24V C V2.0 | 6ES7 138-4DA04-0AB0 C | 526...537 | 526...531 | | |
| 3 | 2RO NO 24..230V/5A | 6ES7 132-4HB01-0AB0 | | 0.0...0.1 | | |
| 4 | 2AO U ST | 6ES7 135-4FB00-0AB0 | | 512...515 | | |
| 5 | 2AO U ST | 6ES7 135-4FB01-0AB0 | | 516...519 | | |
| 6 | 4DI DC24V ST | 6ES7 131-4BD01-0AA0 | 0.0...0.3 | | | |
| 7 | 4DI DC24V ST | 6ES7 131-4BD01-0AA0 | 1.0...1.3 | | | |
| 8 | 4DI DC24V ST | 6ES7 131-4BD01-0AA0 | 2.0...2.3 | | | |
| 9 | 4DI DC24V ST | 6ES7 131-4BD01-0AA0 | 3.0...3.3 | | | |
| 10 | 2RO NO 24..230V/5A | 6ES7 132-4HB01-0AB0 | | 1.0...1.1 | | |
| 11 | 4DO DC24V/0,5A ST | 6ES7 132-4BD02-0AA0 | | 2.0...2.3 | | |
| 12 | 4DO DC24V/0,5A ST | 6ES7 132-4BD00-0AA0 | | 3.0...3.3 | | |
| 13 | | | | | | |
| 14 | | | | | | |
| 15 | | | | | | |
| 16 | | | | | | |
| 17 | | | | | | |
| 18 | | | | | | |
| 19 | | | | | | |
| 20 | | | | | | |

**Slika 4.17.** *Konfiguracija ET200S modula.*

Slika 4.17. prikazuje kartice potrebne za rad makete. Bitno je napomenuti da se stvarni fizički redoslijed kartica mora poštovati i u samoj konfiguraciji kako PLC ne bi slao poruku o greški.

## 4.7 Procesiranje ulaznih signala

Ulazni signali se dijele na: analogne i digitalne. U izradi makete se upotrebljavaju kartice za ulazne digitalne signale. Da se olakša rad s ulaznim signalima koristi se tablica simbola (eng. *Symbol table*) gdje se digitalnim ulazima pridjeljuju simbolička imena.

| | | | | | |
|---|---|---|---|---|---|
| 12 | | S1 | I | 0.1 | BOOL |
| 13 | | S2 | I | 0.2 | BOOL |
| 14 | | S4 | I | 1.0 | BOOL |
| 15 | | S3 | I | 1.2 | BOOL |

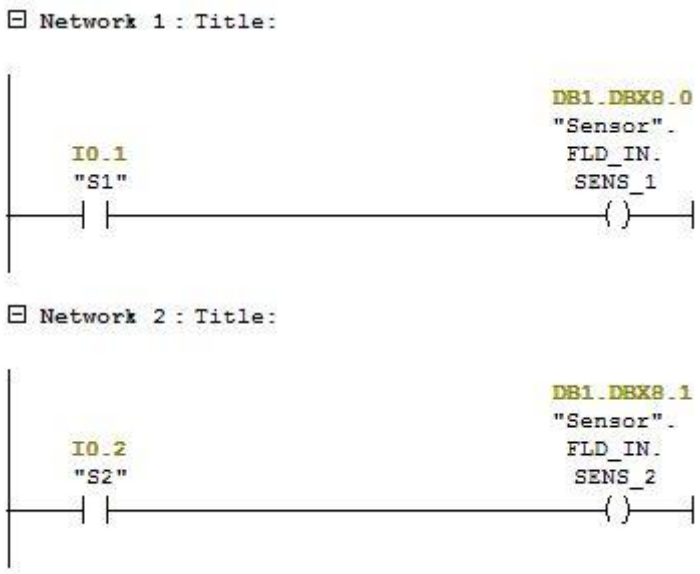
**Slika 4.18.** *Tablica simbola.*

Ulazni signali se označavaju na sljedeći način npr. I 0.0, gdje slovo označava da se radi o ulaznom signalu (eng. *Input*), prvi broj označava broj kartice, a drugi broj označava ulaz na toj kartici. Označavanje kartica i priključaka kreće od nula, pa bi za navedeni primjer to bio prvi ulaz na prvoj kartici.[1] U primjeru na slici 4.18. se može vidjeti dodjeljivanje simboličkih imena digitalnim ulazima, gdje je ulazu I 0.1 pridruženo simboličko ime S1, ulazu I0.2 je pridruženo simboličko ime S2 i tako dalje. Simbolička imena u ovom slučaju predstavljaju senzore, gdje je S1 senzor broj jedan.

Osim dodjele simboličkih imena ulaznim signalima često se ti signali spremaju u odgovarajuće varijable. Step 7 dozvoljava i korištenje adresa ulaznih signala direktno u programu (slika 4.19.), ali to se u praksi pokazalo nespretno i nepouzdano.

**Slika 4.19.** *Prikaz direktne upotrebe ulaznog signala.*

Ako bi iz bilo kojeg razloga zakazao navedeni ulaz na kartici tada bi bilo potrebno vršiti promjene na svim mjestima u programu gdje se koristi ili poziva navedeni ulaz. Stoga se vrši takozvano mapiranje. Mapiranje je pridruživanje signala unaprijed definiranim varijablama (slika 4.20.).




**Slika 4.20.** *Mapiranje ulaznih digitalnih signala.*

Na ovaj način se izbjegavaju komplikacije i prepravljanje programa ako neki ulaz prestane raditi. U tom slučaju signal se spoji na drugi ulaz, te se u bloku za mapiranje novi ulaz pridruži istoj varijabli na kojoj je bio prethodni ulaz.

## 4.8 Procesiranje izlaznih signala

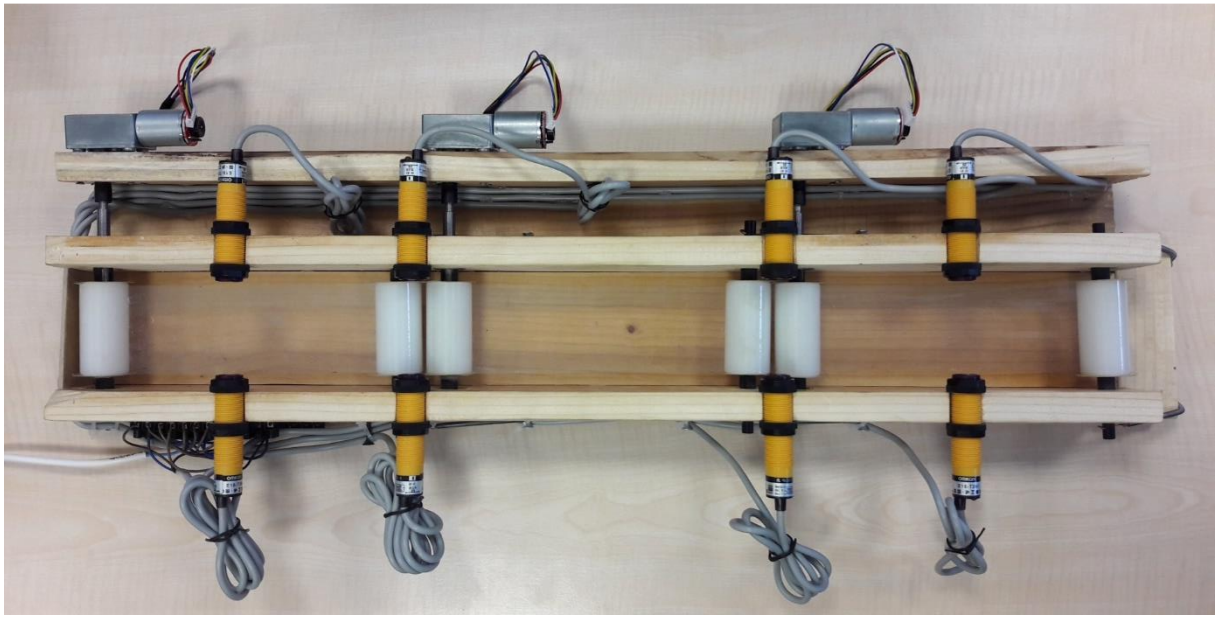
Za rad makete se koriste analogni i digitalni izlazni signali. Radi bolje strukturiranosti i preglednosti programa preporučljivo je koristiti mapiranje kao i kod ulaznih signala.

**Slika 4.21.** *Mapiranje izlaznih digitalnih signala.*

Standardom je određeno da se izlazni digitalni signali zapisuju na sljedeći način Q 2.2. Slovo Q definira da se radi o digitalnom izlaznom signalu, prvi broj predstavlja broj kartice,a drugi broj je broj izlaza na toj kartici. Na slici 4.21. varijabla *PTRK_BS_ManFWD* pomoću normalno otvorenog kontakta uključuje i isključuje digitalni izlaz Q 2.2.

## 4.9 Prikaz makete

Slika 4.22. prikazuje maketu pokretne trake koja se sastoji od tri sekvence. Postolje makete je opisano u poglavlju 4.4. i prikazano je slikom 4.10. Na postolju se nalaze tri istosmjerna motora upravljanja PLC-om i četiri infracrvena senzora. Ukupna duljina trake je 660 *mm*.



**Slika 4.22.** *Maketa pokretne trake.*

# 5 ZAKLJUČAK

U diplomskom radu je teorijski opisano programsko okruženje *CoDeSys* u kojemu je napravljena simulacija makete pokretne trake. Kako bi simulacija što realnije prezentirala maketu pokretne trake bilo je potrebno napraviti blok koji će služiti za simuliranje rada motora. Veliki broj motora u industriji se pokreće i zaustavlja pomoću odgovarajući rampi. Pokretanje i zaustavljanje motora pomoću rampe smanjuje udarna opterećenje zbog inercije, smanjuje potrošnju energije i umanjuje stres samom motoru. Zbog toga se u simulaciji brzina motora prikazuje pomoću S-krivulje. Osim motora bilo je potrebno napraviti blok za simuliranje rada inkrementalnog enkodera. Da bi se mogao simulirati rad enkodera potrebno je znati sljedeće parametre poput rezolucije enkodera, promjera valjaka i brzine motora kako bi simulacija bila što točnija.

Nadzor proizvoda je jako bitan čimbenik u procesu automatizacije u metalnoj industriji. Omogućava praćenje proizvoda tijekom transporta i pravovremeno djelovanje u slučaju kvara ili nezgode. Sustav nadzora proizvoda sastoji se od pridruživanja inicijalnih podataka svakom novom proizvodu koji dođe na traku, sinkronizacije podataka o proizvodu, računanja i osvježavanja pozicije proizvoda i odlaganja proizvoda.

Spajanjem računala na PLC moguće je promatrati što se događa na PLC-u i u kojoj fazi procesa se nalazi. Zbog toga se kreira takozvani HMI, odnosno programsko sučelje koje omogućava praćenje i upravljanje procesom preko računala. U simulaciji je napravljen HMI koji omogućava upravljanje trakom i praćenje proizvoda koji se nalaze na traci.

U drugom dijelu rada je opisana izrada makete pokretne trake. Za upravljanje maketom se koristi *Siemensov PLC S7-400* koji je industrijskim ethernet kabelom povezan na računalo. Nakon uspostave komunikacije između računala i PLC-a u programu *HW config* se dodaju svi elementi potrebni za upravljanje trakom pomoću PLC-a. PLC trakom upravlja preko ET200S modula na kojem se nalaze sve ulazne i izlazne kartice. Za komunikaciju između PLC-a i ET200S modula se koristi PROFIBUS kabel čije je parametre potrebno postaviti u programu *HW config*.

Programsko rješenje za upravljanje i nadzor trake koristi blokove koji su vrlo sličnim blokovima osmišljenim i kreiranim u simulaciji. Program za upravljanje trakom uglavnom je pisan u *ladder*-u ili *SCL* programskom jeziku. Program je napravljen u programskom okruženju *Simatic manager*.

Ukupna duljina makete je 660 *mm* i se sastoji od tri sekvence koje pokreću tri istosmjerna motora, osim toga koriste se četiri infracrvena senzora za davanje pozicije. Za izradu makete bilo je potrebno napraviti odgovarajuće postolje na koje se dodaju ostali elementi. Osim makete bilo je potrebno realizirati i sklop koji će posredovati između analogne izlazne kartice i motora. Maketa je uspješno realizirana, testirana i odobrena. Svrha makete je što vjerniji prikaz i simulacija djela proizvodne linije za prijenos materijala u metalnoj industriji.

# 6 LITERATURA


[1] https://www.scribd.com/doc/184536652/Simatic-s7-St-7pro1-Hrvatski
Datum pristupa linku 12.10.2017. u 15:20

[2] Danieli Automation, prezentacija - Level 1 basic software, CoDeSys programming basic course

[3] Training Module A1-Siemens Simatic S7

[4] http://marie-www.ee.pw.edu.pl/~purap/PLC/manuals/m07590333_00000000_1en.pdf
Datum pristupa linku 4.10.2017. u 10:00

[5] http://www.dx.com/p/40rpm-12v-encoder-motor-dc-turbo-worm-gear-motor-w-self-lock-434055#.WhwTTEpKvIV
Datum pristupa linku 27.11.2017. u 14:30-

# 7 POPIS SLIKA

**Slika 2.1.** Primjer programskih jezika CFC i IL. ..... 2
**Slika 2.2.** S krivulja. ..... 3
**Slika 2.3.** Blok SRAMP. ..... 4
**Slika 2.4.** Ispitivanje zadane i trenutne brzine. ..... 4
**Slika 2.5.** Razlika između zadane i trenutne brzine. ..... 5
**Slika 2.6.** Računanje površine ispod krivulje ubrzanja. ..... 5
**Slika 2.7.** Usporedba preostale brzine i površine ubrzanja. ..... 5
**Slika 2.8.** Rampa ubrzavanja i usporavanja. ..... 6
**Slika 2.9.** Integriranje brzine. ..... 6
**Slika 2.10.** Grafički prikaz brzine i akceleracije. ..... 6
**Slika 2.11.** Enkoder. ..... 7
**Slika 2.12.** Odabir automatskog upravljanja trakom. ..... 8
**Slika 2.13.** Odabir manualnog upravljanja trakom. ..... 8
**Slika 2.14.** Odabir brzine. ..... 9
**Slika 2.15.** Blok SEL_5R. ..... 9
**Slika 2.16.** Definiranje strukture podataka. ..... 10
**Slika 2.17.** Definiranje pozicije glave, repa i duljine proizvoda. ..... 11
**Slika 2.18.** Dodjeljivanje ID-a proizvodu. ..... 11
**Slika 2.19.** Prijenos proizvoda između spremnika. ..... 12
**Slika 2.20.** Brisanje proizvoda iz spremnika. ..... 13
**Slika 2.21.** Računanje pređenog puta po ciklusu. ..... 14
**Slika 2.22.** Proizvod se nalazi na dvije sekvence. ..... 14
**Slika 2.23.**Uvjet da se proizvod nalazi na dvije trake. ..... 15
**Slika 2.24.** Postotak materijala na pojedinom području. ..... 15
**Slika 2.25.** Računanje ukupnog pređenog puta. ..... 15
**Slika 2.26.** Proizvod se nalazi samo na jednoj sekvenci. ..... 16
**Slika 2.27.** Osvježavanje pozicije glave i repa. ..... 16
**Slika 2.28.** Ispitivanje zauzetosti područja. ..... 16
**Slika 2.29.** Blok za korekciju pozicije proizvoda. ..... 17
**Slika 2.30.** Računanje tolerancija. ..... 18
**Slika 2.31.** Korekcija pozicije. ..... 18
**Slika 2.32.** Uključenje alarma za pojedini proizvod. ..... 18

**Slika 2.33.** Promjena pozicije glave i repa. 19
**Slika 2.34.** Provjera udaljenosti između proizvoda. 20
**Slika 2.35.** Prikaz proizvoda na poziciji za odlaganje. 21
**Slika 2.36.** Ispitivanje uvjeta za zaustavljanje i odlaganje proizvoda. 21
**Slika 2.37.** Odlaganje proizvoda, zaustavljanje trake i brisanje iz spremnika. 22
**Slika 3.1.** Kontrola trake. 23
**Slika 3.2.** Kontrola senzora. 24
**Slika 3.3.** Prikaz motora. 24
**Slika 3.4.** Model pokretne trake. 25
**Slika 3.5.** Prikaz proizvoda na traki. 25
**Slika 3.6.** HMI sučelje. 26
**Slika 4.1.** Bočni prikaz makete pokretne trake. 27
**Slika 4.2.** Prikaz područja na maketi pokretne trake. 28
**Slika 4.3.** Blokovski prikaz čitanja i pisanja u procesu. [1] 29
**Slika 4.4.** Primjer programskih jezika. 30
**Slika 4.5.** Tlocrt i bokocrt istosmjernog motora s reduktorom i enkoderom. 32
**Slika 4.6.** Prikaz ožičavanja motora i enkodera. 32
**Slika 4.7.** Blokovski prikaz prijemnika i predajnika. 32
**Slika 4.8.** Bokocrt i nacrt osovine s valjkom. 33
**Slika 4.9.** Postizanje napetosti trake pomoću opruga. 33
**Slika 4.10.** Prikaz postolja za maketu pokretne trake. 34
**Slika 4.11**. Arduino sklop za upravljanje s dva istosmjerna ili jednim koračnim motorm. 34
**Slika 4.12.** Sklop za upravljanje motorima. 35
**Slika 4.13.** Reskaliranje vrijednosti. 36
**Slika 4.14.** Hardverska konfiguracija PLC-a. 37
**Slika 4.15.** Konfiguriranje PROFIBUS komunikacije. 38
**Slika 4.16.** ET200S modul. 39
**Slika 4.17.** Konfiguracija ET200S modula. 39
**Slika 4.18.** Tablica simbola. 40
**Slika 4.19.** Prikaz direktne upotrebe ulaznog signala. 40
**Slika 4.20.** Mapiranje ulaznih digitalnih signala. 41
**Slika 4.21.** Mapiranje izlaznih digitalnih signala. 41
**Slika 4.22.** Maketa pokretne trake. 42

# ŽIVOTOPIS

Vinko Juranović

Rođen u Doboju 7. prosinca 1993. godine. U Žepču, 2008. godine završava osnovnu školu „OŠ Žepče“ te upisuje srednju elektrotehničku školu u Katoličkom školskom centru „Don Bosco“. Završava srednju školu 2012. godine i upisuje preddiplomski studij elektrotehnike u Osijeku. Godine 2015. završava preddiplomski studij elektrotehnike u Osijeku te upisuje diplomski studij komunikacija i informatike na istom fakultetu.

U Osijeku, prosinac 2017.

_____

Vinko Juranović

# SAŽETAK

U ovom radu je opisana maketa pokretne trake. Rad se sastoji od dva dijela, opis simulacije pokretne trake i opis makete pokretne trake.

Prvi dio opisuje simulacija rada motora i enkodera, te sustava za praćenje proizvoda. Sustav praćenja proizvoda jako je bitan u automatizaciji pogona. Omogućava nadzor proizvoda kao i pravovremeno reagiranje u slučaju kvara ili nesreće. Simulacija je napravljena u programskom okruženju *CoDeSys*.

Osim kreiranja programa *CoDeSys* omogućuje i stvaranje sučelja za praćenje i upravljanje simulacijom, odnosno vizualizaciju simulacije.

U drugom dijelu je opisan postupak izrade makete pokretne trake i programa korištenjem *Simatic manager-a*. Maketa se sastoji od tri sekvence koje su pogonjene s tri istosmjerna motora i četiri infracrvena senzora za davanje pozicije. Za upravljanje maketom se koristi *Siemensov* PLC iz serije *S7-400*.

**Ključne riječi:** PLC, CoDeSys, Simatic Manager, pokretna traka, S-krivulja, transport, praćenje proizvoda

# ABSTRACT


**Conveyor belt model**

This thesis describes conveyor belt model. The thesis consists of two parts - description of simulation and model of conveyor belt.

First part describes motors and encoder simulations and also product tracking system. Product tracking system is very important for system automation, because it allows supervision of products and timely response in case of fault or accident. The simulation was made in *CoDeSys* programming environment.

Except for creating the software, *CoDeSys* allows making an interface for monitoring and managing the simulation, in fact visualisation of simulation.

Second part describes process of creating conveyor belt model and software using *Simatic Manager*. The model consists of three sequences which are driven by three DC motors and four infrared position signaling sensors. Siemens PLC from series S7-400 is used for model management.



**Key words:** PLC, CoDeSys, Simatic Manager, conveyor belt, S-ramp, transport, product tracking