

# Upravljanje pozicijom u otvorenoj petlji pomoću digitalnih davača pozicije te digitalnih i analognih aktuatora

---

Žagar, Benjamin

Master's thesis / Diplomski rad

2016

*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:200:751949>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-09-21**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
ELEKTROTEHNIČKI FAKULTET**

**Sveučilišni studij**

**UPRAVLJANJE POZICIJOM U OTVORENOJ PETLJI  
POMOĆU DIGITALNIH DAVAČA POZICIJE TE  
DIGITALNIH I ANALOGNIH AKTUATORA**

**Diplomski rad**

**Benjamin Žagar**

**Osijek, 2016.**

# SADRŽAJ

1. UVOD .....	1
2. UPRAVLJANJE PROCESIMA .....	3
2.1. Upravljanje s povratnom vezom .....	3
2.2. Upravljanje bez povratne veze .....	5
3. UPRAVLJANJE POZICIJOM BEZ POVRATNE VEZE U INDUSTRIJI.....	7
3.1. Industrijski postav .....	7
3.2. Programabilni logički kontroler - PLC .....	11
3.3. Digitalni davači pozicije i mehanički prekidači.....	13
3.4. AS (engl. <i>Actuator/Sensor</i> ) sučelje.....	15
3.5. Elektromotor .....	16
4. PROGRAMSKO RJEŠENJE.....	19
4.1. Programsko rješenje simulacije .....	20
4.2. Programsko rješenje vizualizacije.....	35
5. ANALIZA PROGRAMSKOG RJEŠENJA.....	43
5.1. Pokretanje vizualizacije i simulacije.....	44
5.2. Testiranje programskog rješenja .....	48
6. ZAKLJUČAK .....	53
LITERATURA.....	54
SAŽETAK .....	55
ABSTRACT.....	56
ŽIVOTOPIS.....	57



## 1. UVOD

Načelo upravljanja u osnovi nije nikakav tehnički izum. Ovo načelo utjelovljeno je u svim živim bićima i glavni je razlog održivosti života. Isto tako, ovo načelo je prisutno i u mnogim drugim netehničkim procesima kao što su biološki, ekonomski, sociološki i politički procesi [1]. Iz perspektive tehničkih procesa i sustava ovo načelo je neizostavan faktor i predstavlja temeljnu disciplinu za mnoge inženjerske znanosti. Ovaj diplomski rad bavit će se pogledom na automatsko upravljanje upravo iz perspektive tehničkih procesa i sustava.

U današnje vrijeme, automatsko upravljanje je sustavno uređena inženjerska znanost, te se može reći da živimo u doba automatizacije. Više od devedeset posto današnjih sustava ne bi moglo ispravno raditi bez primijenjenog načela automatske regulacije. Iako relativno nerazvikana disciplina, razvoj automatskog upravljanja predstavlja vrlo bitan faktor u tehnološkom napretku. Razvoj automatskog upravljanja i informacijskih tehnologija temelj su doba u kojem živimo, doba automatizacije [1].

Kao što je i ranije navedeno, automatsko upravljanje može se primijeniti na različite dinamičke sustave. Dinamički sustav predstavlja funkcijsku cjelinu za obradu i prjenos energije, materije, informacije i kapitala, gdje se ulazne veličine sustava promatraju uzrokom, a izlazne veličine sustava njegovom vremenskom posljedicom [1]. Upravo zbog toga je dinamički sustav središnja točka automatskog upravljanja. Krajnji cilj je analizirati dinamički sustav u svrhu dobivanja informacija o zakonitostima koje vladaju unutar samog procesa te na temelju tih podataka izvršiti sintezu tehničkog sustava kako bi poboljšali njegov rad. Automatsko upravljanje, kao znanstvena disciplina, daje različite načine i metode analize i sinteze dinamičkih sustava.

Jedan od takvih dinamičkih sustava je i sustav promjene pozicije. Zamislimo kola koja vuku konji. Ulaz ovog dinamičkog sustava bio bi uzrok kretanja tih kola, odnosno vuča konja. Izlaz iz sustava bio bi vremenska posljedica vuče konja, a to je promjena pozicije. Automatsko upravljanje postavlja pitanje: Kako upravljati ovim sustavom na taj način da se kola pozicioniraju približno točno tamo gdje je čovjek naumio? Najbolje rezultate postigao bi sam čovjek koji upravlja kolima. Razvojem industrije i tehnologije konje je zamjenio motor dok je čovjeka zamjenilo računalo. Razvojem automatskog upravljanja razvile su se metode kako pomoću računala upravljati motorima. Iako je ovo vrlo jednostavan sustav, upravljanje pozicijom ima široku primjenu u industriji i zbog toga je vrlo bitan. Ovaj diplomski rad bavi se upravo ovim sustavom.

Tema ovog rada opisana je u šest poglavlja. U drugom poglavlju dat će se malo veći uvid u samo upravljanje procesima te naznačiti razlika između upravljanja u zatvorenoj i otvorenoj petlji. U trećem poglavlju prikazati će se proces upravljanja pozicijom u industriji. Prikazati će se gdje se koristi, opisati će se hardware i pogon. Najveći naglasak će biti na digitalnim davačima, digitalnim i analognim aktuatorima, PLC-u (engl. *Programmable Logic Controller*) i ASi-u (engl. *Actuator/Sensor interface*). Četvrto poglavlje će predstaviti programsko rješenje. Prikazati će se kako je izvedena simulacija sustava te kojim programskim alatom. Prikazati će se i kako je ostvarena vizualizacija sustava te koji je programski alat bio potreban za izradu. Peto poglavlje predstavlja analizu programskog rješenja. Prikazati će se i komentirati kako se sustav upravljanja pozicijom ponaša za razne slučajeve i parametre. U posljednjem poglavlju dan je zaključak diplomskog rada, gdje su iznesene najvažnije činjenice vezane uz sam diplomski rad.

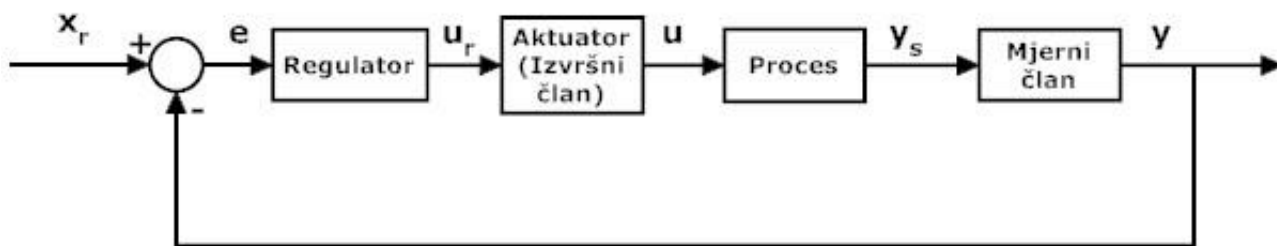
**Zadatak ovog diplomskog rada** je ostavriti simulaciju sustava upravljanja pozicijom pomoću programskog paketa Codesys. Upravljanje pozicijom vrši se u otvorenoj petlji pomoću digitalnih davača pozicije i digitalnih i analognih aktuatora. Na kraju je potrebno izraditi vizualizaciju procesa upravljanja pozicijom pomoću programskog paketa Codesys.

## **2. UPRAVLJANJE PROCESIMA**

Gledano iz inženjerske perspektive automatsko upravljanje je jedna od najbitnijih znanstvenih disciplina. Kao što je i ranije navedeno u središtu promatranja automatske regulacije je dinamički sustav. Ova znanstvena disciplina daje nam puno različitih načina kako spoznati zakonitosti nekog dinamičkog sustava, odnosno procesa, te različite načine kako da na temelju tih spoznaja projektiramo tehnički odnosno poboljšamo prirodni sustav. Iako se sam princip upravljanja procesima čini jednostavan, zbog brojnosti dinamičkih sustava koji se razlikuju po osobinama, složenosti prirodnih procesa, temeljite analize u svrhu projektiranja što boljeg sustava te različitih načina sinteze tehničkih sustava ovo je vrlo opširna i složena znanstvena disciplina u kojoj prvenstveno prednjači matematika i fizika kao osnovni alati za analizu i sintezu procesa, a tek onda programiranje i ostale tehničke discipline. Ovaj diplomski rad bavi se upravljanjem jednim relativno jednostavnim procesom, upravljanjem promjenom pozicije. Upravo zbog toga nije potrebno koristiti neke složene metode analize i sinteze procesa, koje u izvedbi koštaju, već je dovoljno koristiti upravljanje bez povratne veze. Upravljanje procesom promjene poziciju u ovom radu izvedeno je bez povratne veze i zbog toga je vrlo važno naznačiti kakvo je to upravljanje i koja je razlika u odnosu na upravljanje s povratnom vezom. O ovome može se pročitati u nastavku.

### **2.1. Upravljanje s povratnom vezom**

Kako bi se što bolje objasnio ovaj način upravljanja, ali i upravljanje bez povratne veze koje će biti objašnjeno u sljedećem poglavlju, koristi se blokovski prikaz sustava. Iz potrebe za jednostavnijim i preglednijim načinom prikaza tehničkog sustava ili regulacijskog kruga koristi se jedan od standardiziranih prikaza u kojem se sustav ili pojedini član regulacijskog kruga prikazuje blokom, jednoznačni smjer djelovanja prikazuje se smjerom strelice, a određene matematičke operacije prikazuju se simbolom. Sve ove elemente možemo vidjeti na slici 2.1. Na toj slici također možemo vidjeti i od čega se sastoji regulacijski krug. Regulator, aktuator, proces i mjerni član dio su regulacijskog kruga. Svaki ovaj član ima svoj ulaz i izlaz, te svoju funkciju u krugu.



Slika 2.1. Regulacijski krug upravljanja s povratnom vezom [1]

Proces predstavlja dinamički sustav čijim ponašanjem želimo upravljati. Aktuator predstavlja člana koji izvršava neku radnju i time daje pobudu sustavu. Suštinska uloga regulatora je da obrađuje regulacijsko odstupanje  $e$  po određenom algoritmu, djelujući upravljačkim signalom na aktuator i na kraju preko aktuatora na sam proces [1]. Mjerni član je zapravo određeni senzor koji mjeri izlazne podatke procesa. Kao što je i ranije navedeno, svaki član ima svoj ulazni i izlazni signal. Signal  $u$  predstavlja izvršnu veličinu. Ovaj signal ulaz je u proces, odnosno pobuda procesu, te izlaz aktuatora, odnosno radnja koju je aktuator izvršio. Izlaz iz procesa je signal  $y_s$  i predstavlja stvarnu vrijednost procesa. Signal  $y_s$  je vremenska posljedica pobude  $u$ . Mjerni član „mjeri“ stvarnu vrijednost procesa, odnosno pretvara određenu fizičku vrijednost u signal  $y$  koji je razumljiv regulatoru. Već je ranije spomenuto da je izlaz regulatora, signal  $u_r$ , upravljački signal koji upravlja aktuatorom. Ulaz u regulator, signal  $e$ , je veličina na kojoj se zasniva upravljanje sa povratnom vezom. Ukoliko se bolje promotri slika 2.1. može se vidjeti da je signal  $e$  dan izrazom:

$$e(t) = x_r(t) - y(t) \quad (2-1)$$

Veličina  $x_r$  se naziva referentna veličina. Kao što joj i ime kaže, ova veličina predstavlja referencu sustavu, odnosno željeno ponašanje sustava. Veličina  $y$ , koja predstavlja stvarnu vrijednost sustava, uspoređuje se sa željenom vrijednošću sustava te se dobije razlika, veličina  $e$ , koja se naziva regulacijsko odstupanje. Ukoliko se sustav ponaša onako kako želimo, prema referenci, regulacijsko odstupanje će biti jednako 0. Ukoliko postoji razlika između reference i stvarne veličine, regulator će pomoću svog algoritma obraditi tu razliku, poslati upravljački signal aktuatoru, aktuator će izvršiti radnju, dati će pobudu procesu, procesu će se promijeniti izlaz, taj izlaz će se ponovno slati na provjeru s referencom i, ukoliko ne postoji razlika, sustav se ponaša onako kako želimo. Kao što se može vidjeti, ovo je zatvoreni krug toka signala i upravo zato se

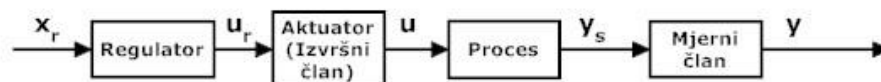


naziva upravljanje u zatvorenoj petlji. Upravljanje u zatvorenoj petlji zasniva se na povratnoj vezi kojom se izlaz procesa stalno vraća na usporedbu s referentnom veličinom.

Pošto je u središtu diplomskog rada proces upravljanja pozicijom, potrebno je napraviti analogiju kako bi se moglo pokazati koja je razlika između upravljanja s i bez povratne veze upravo kod ovog procesa. Kod upravljanja pozicijom, aktuator bi bio određena vrsta motora. Proces kojim želimo upravljati je promjena pozicije. Kada se motor pokrene, ubrza ili uspori, proces se pobudi i kao vremenska posljedica pobude, pozicija se promjeni. Mjerni član je enkoder određene vrste koji prati apsolutnu poziciju. U svakom trenutku ta apsolutna pozicija koju prati enkoder uspoređuje se s referentnom veličinom, odnosno zadanom pozicijom. Ukoliko se sustav nije pozicionirao tamo gdje referentna veličina zahtjeva, postojati će razlika između apsolutne pozicije sustava i referentne veličine. Ukoliko postoji razlika, regulator će reagirati i obraditi ju pomoću svog algoritma pritom usporavajući ili ubrzavajući motor. Na taj način ostvarena je regulacija pozicije sa povratnom vezom kod procesa upravljanja pozicijom. O upravljanju bez povratne veze može se pročitati u nastavku.

## 2.2. Upravljanje bez povratne veze

Upravljanje bez povratne veze, kao što mu i samo ime kaže, razlikuje se od upravljanja s povratnom vezom po tome što kod ovog načina upravljanja nema vraćanja izlaza procesa na stalnu usporedbu s referentnom vrijednošću. Procesom i dalje upravlja regulator preko aktuatora, ali bitna razlika je ta da vladanje procesa više neće u svakom trenutku pokušati pratiti neku referentnu vrijednost nego će se proces odvijati dok regulator svojim algoritmom ne odluči promijeniti stanje procesa pokušavajući dostići približnu vrijednost zadanu na ulazu. Ovakvo upravljanje zahtjeva i drugačije algoritme i načine sinteze regulatora jer regulator više ne radi na temelju razlike stvarne i referentne vrijednosti. Otvoreni regulacijski krug upravljanja bez povratne veze izgleda skoro identično kao i kod upravljanja s povratnom vezom samo bez povratne veze. To se može vidjeti na slici 2.2.



Slika 2.2. Regulacijska staza bez povratne veze

Pošto je u središtu diplomskog rada proces upravljanja pozicijom, potrebno je napraviti analogiju kako bi se moglo pokazati koja je razlika između upravljanja s i bez povratne veze upravo kod ovog procesa. Kao i u primjeru opisanom u prethodnom poglavlju, određena vrsta motora predstavlja aktuator za ovaj sustav. Regulator je algoritam koji upravlja motorom. Mjerni član u ovom slučaju može biti enkoder ali i digitalni davač pozicije. U ovom primjeru objasniti će se rad s digitalnim davačem pozicije. Digitalni davač pozicije je jednostavan senzor koji na izlazu daje logičku jedinicu ukoliko se nešto što pozicioniramo, na primjer vagon, nalazi iznad njega, te logičku nulu ukoliko se ništa ne nalazi iznad njega. O detaljnom principu rada ovih senzora može se pročitati u sljedećim poglavljima.

Veličina  $x_r$ , koja ulazi u regulator, predstavlja zadanu poziciju. Pozicija se može zadati npr. s kontrolnog panela i ona predstavlja poziciju na kojoj se nalazi jedan od digitalnih davača. Kada regulator primi ulaznu veličinu i obradi ju pomoću svog algoritma, šalje upravljački signal motoru koji npr. počinje ubrzavati naprijed. Vremenskom posljedicom ubrzavanja motora mijenja se i pozicija, odnosno izlaz iz procesa. Digitalni davač u svakom trenutku prati da li je vagon pozicioniran na njega. Ukoliko se vagon pozicionira na njega, te je digitalni davač pozicije u logičkoj jedinici, algoritam regulatora prepoznaje da je vagon došao na traženo mjesto te šalje upravljački signal motoru da uspori i stane. Na ovaj način vagon se pozicionirao pomoću relativne pozicije na željeno mjesto.

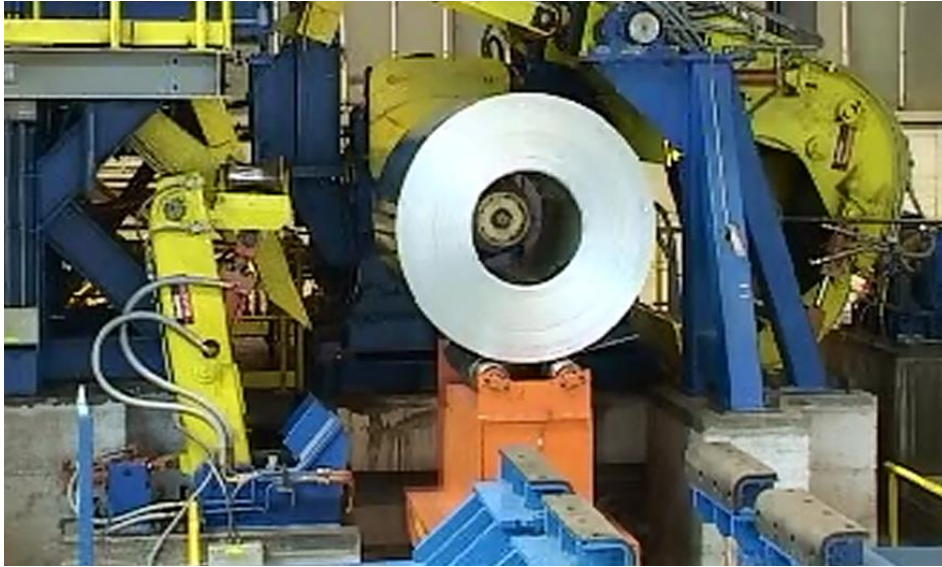
Nakon što je dan primjer za upravljanje pozicijom s i bez povratne veze, mogu se vidjeti bitne razlike između ova dva načina upravljanja. Kod upravljanja s povratnom vezom proces u svakom trenutku pokušava pratiti zadanu poziciju na temelju razlike između zadane i referentne vrijednosti. Kod upravljanja pozicijom bez povratne veze proces ne prati neku referentnu veličinu već se pomoću regulatora i digitalnog davača pokušava pozicionirati približno tamo gdje je zadano na ulazu. Programsko rješenje i simulacija u ovom diplomskom radu napravljeni su na temelju upravljanja bez povratne veze i temelje se na procesu opisanom ranije u ovom poglavlju. O opisu stvarnog industrijskog sustava upravljanja pozicijom bez povratne veze i programskom rješenju simulacije i vizualizacije ovog sustava može se pročitati u nastavku.

### **3. UPRAVLJANJE POZICIJOM BEZ POVRATNE VEZE U INDUSTRIJI**

U prošlom poglavlju dan je primjer upravljanja pozicijom s i bez povratne veze. Ovi primjeri zapravo predstavljaju teorijsku osnovu funkcioniranja jednog ovakvog sustava. Kako bi se što bolje obradila ova tema i što bolje projektiralo programsko rješenje, odnosno simulacija sustava upravljanja pozicijom, vrlo je bitno upoznati i izvedbu ovoga sustava u praksi. Kao što je i ranije navedeno, iako jednostavan, ovaj sustav je jedan od bitnijih sustava u industriji. Ukoliko se zamisli skladište nekih proizvoda koje se želi automatizirati, lako se može zaključiti kako je sustav upravljanja pozicijom neizostavan dio ove cjeline. Bilo da se radi o nekakvom sekvencijalnom preslagivanju proizvoda ili samo transportu koristiti će se upravo ovaj sustav upravljanja. No, priča kod transporta i skladištenja robe ne staje. Tu postoje još i sustavi za zavarivanje, razne hidrauličke preše, sustavi za rezanje itd. U svim ovim sustavima primjena upravljanja pozicijom je neizostavna. Upravo zbog toga što je u industriji proces promjene pozicije toliko rasporestranjen i neophodan u skoro svakom postrojenju se može pronaći sustav upravljanja pozicijom. U ovom poglavlju biti će opisan jedan proces transporta proizvoda. Prikazati će se na koji je način sve to ostvareno, biti će prikazana shema i obajašnjene funkcijske cjeline. Također, objasniti će se i način komunikacije senzora i aktuatora, te koji su to senzori i aktuatori korišteni. O svemu ovome može se pročitati u nastavku.

#### **3.1. Industrijski postav**

Kao što je i ranije navedeno, opisati će se sustav upravljanja pozicijom u svrhu transporta. Ovaj podsustav dio je velikog postrojenja za obradu metala. Kao što se može vidjeti na slici 3.1., zadaća sustava upravljanja pozicijom je transport velikih namotaja traka proizvedenih od određenog metala. Nakon što je završio prijašnji proces proizvodnje i namatanje trake, cilj ovog procesa je osigurati skladištenje traka. Na slici 3.1. može se vidjeti kako je sustav za namatanje trake završio svoj posao te se vagon za transport tereta pozicionirao ispod namotaja kako bi preuzeo teret. Nakon toga vagon transportira traku do određene pozicije, što se može vidjeti na slici 3.2. Najčešće, kada se traka pozicionira na određenu poziciju, teret preuzima dizalica i odnosi traku na skladište. Iako ovaj sustav zauzima vrlo malo mjesto u cijelom proizvodnom procesu, može se zaključiti da je neophodan.



*Slika 3.1. Sustav za transport namotaja*



*Slika 3.2. Namotaj se pozicionira na određenu poziciju*

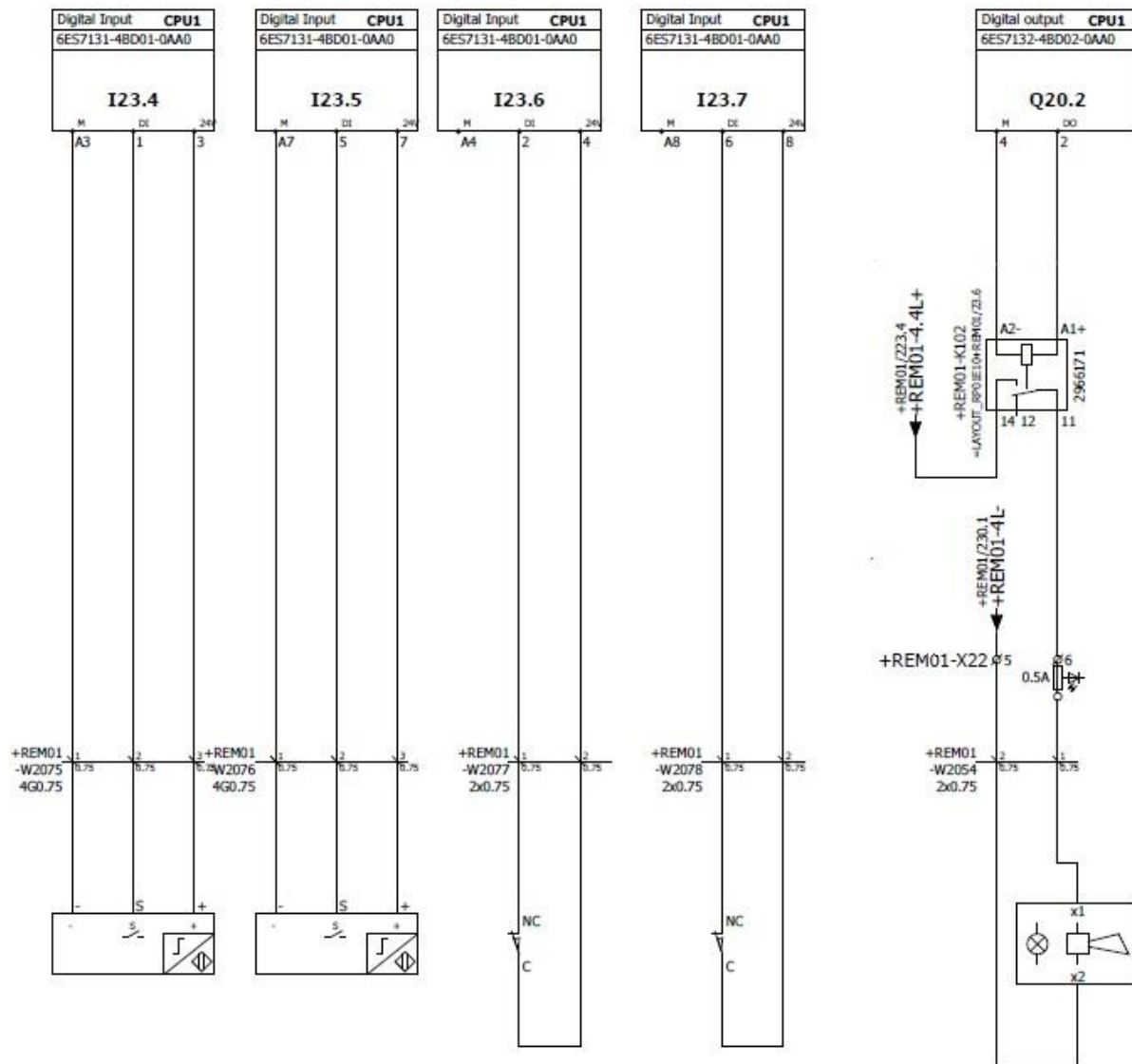
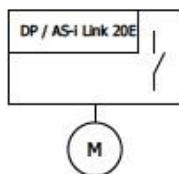
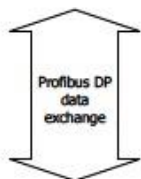
Teretna kolica (vagon) pogoni motor određene vrste i to je aktuator ovog sustava. Mjerni članovi su digitalni davači pozicije. Na slici 3.2. davači se mogu jasno vidjeti uz teretna kolica (obojeni plavom bojom). Ovaj sustav ima još i sigurnosne senzore za poziciju koji su osiguranje da vagon slučajno ne prijeđe van tračnica ili dopuštene maksimalne pozicije. U sustav su ugrađene i druge sigurnosne mjere kao što je sirena i dioda za alarm. Upravljanje i nadzor sustava vrši se iz kontrolne sobe preko kontrolnog panela. Ovo se može vidjeti na slici 3.3.



*Slika 3.3. Nadzor sustava preko kontrolnog panela*

Sada kada se zna kako ovaj sustav izgleda i koja mu je funkcija potrebno je opisati na koji način je ostvarena regulacija ovog procesa. Potrebno je opisati na koji način su spojeni aktuator i mjerni članovi s PLC-om, te koja su sučelja i komunikacijski protokoli korišteni i zašto. Najbolji način za ovo objasniti je preko električne sheme sustava koju možemo vidjeti na slici 3.4. Na slici 3.4. može se vidjeti kako je sirena i dioda za alarm spojena preko releja na PLC. Također može se vidjeti da su digitalni davači pozicije, ali i sigurnosni mehanički senzori za poziciju, spojeni direktno na digitalne ulaze PLC-a. Komunikacija i razmjena podataka između PLC-a i jedinice za upravljanje motorom izvršava se preko AS (engl. *Actuator/Sensor*) sučelja. Jedinica za upravljanje motorom dio je MCC-a (engl. *Motor control center*). O svakoj ovoj funkcijskoj cjelini sustava može se pročitati u nastavku.

ASI IFM - Profibus Motor Starter KR	
DP network ID: DP node ID: AS-i node ID: S3 CPU1	
I2301.0	Ready
I2301.1	Running
I2301.2	Not used
I2301.3	Not used
Q2301.0	Forward
Q2301.1	Backward

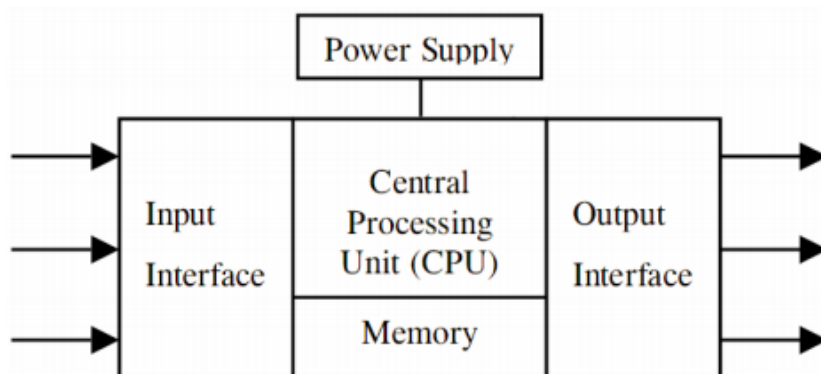


Slika 3.4. Električna shema procesa upravljanja pozicijom u industriji

### 3.2. Programabilni logički kontroler - PLC

Kao što je ranije navedeno, procesom upravlja regulator koji pomoću svoje logike šalje upravljačke signale aktuatoru uzrokujući na taj način promjenu u sustavu i omogućavajući regulaciju sustava. Kada se govori o logici samog regulatora i programiranju algoritama i programa koji omogućuju upravljanje, govori se o niskom stupnju programiranja. Programski jezik nižeg stupnja (engl. *Low level programming language*) u računarstvu označuje programski jezik koji sadrži vrlo malo ili uopće ne sadrži apstrakcije u svojim instrukcijama već su njegove instrukcije vrlo slične procesorskim instrukcijama [2]. Upravo zbog toga klasična izvedba sustava upravljanja u industriji bila bi relejna logika [3]. Relejna logika predstavlja releje i ostale komponente spojene žicama pomoću kojih se vrši neka logika. No, postoji jedan veliki problem kod ovakvih sustava upravljanja, posebno ukoliko je sustav složen. Ukoliko se dogodi pogreška u logici, promjena upravljačke funkcije ili dodavanje novih komponenti mora se izvršiti prespajanje žica [3]. Ukoliko je sustav velik i složen, što znači da će pogreške u logici, promjena upravljačkih funkcija ili dodavanje novih komponenta biti vrlo česti, tada ovaj način upravljanja postaje vrlo loš a sustav neiskoristiv. Upravo zbog toga javila se potreba da se uvede središnje računalo u sustave upravljanja. Središnje računalo za upravljanje sustavima regulacije naziva se programabilni logički kontroler (engl. *Programmable Logic Controller*) ili skraćeno PLC. Uvođenjem PLC-a u sustave upravljanja riješili su se prijašnji problemi. Ukoliko je došlo do pogreške u logici upravljanja, promjene upravljačke funkcije ili dodavanje novih komponenta jedino što se trebalo napraviti je promjena u programskom kodu [3]. Isto tako, uvelike se olakšalo dokumentiranje, dijagnostika i kopiranje aplikacija.

Prema shemi koju možemo vidjeti na slici 3.5. PLC se sastoji od nekoliko funkcijskih cjelina.



Slika 3.5. Funkcijske cjeline PLC-a [3]



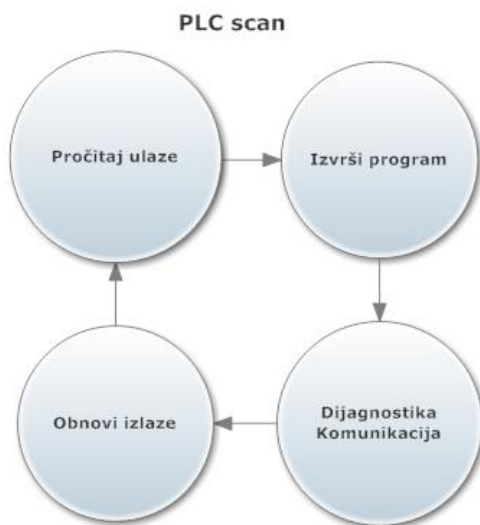
Središnji i glavni dio PLC-a je CPU (engl. *Central Processing Unit*). CPU je procesor, koji vođen zadanim programskim naredbama, izvodi osnovne radnje nad podatcima. Sljedeća funkcijska cjelina PLC-a je memorija. Najraniji PLC-i koristili su memoriju s magnetnom jezgrom i programiranje je bilo vrlo ograničeno zbog nedostatka memorije. U današnje vrijeme najčešće se koriste dvije vrste memorije EPROM (engl. *erasable programmable read-only memory*) i RAM (engl. *Random Access Memory*). EPROM je tip memorijskog čipa koji se može po potrebi micati. Ovakva vrsta memorijskog čipa može zadržati podatke i nakon što se izgubi napajanje [4]. RAM je oblik primarne računalne memorije.

Izlazi i ulazi PLC-a mogu biti analogni ili digitalni. Moduli analognih ulaza najčešće su strujni ili naponski. Strujni analogni ulazi imaju raspon od 0-20mA ili od 4-20mA. Strujni ulazi koriste se ukoliko se strujna petlja priključuje na ulaz PLC-a. Isto tako, strujni analogni ulazi koriste se ukoliko se signal mora slati na velike udaljenosti. Zbog velike udaljenosti dolazi do pada napona zbog otpora kabla. Kada se se koristi strujna petlja pad napona je zanemariv jer kroz petlju teče ista struja. Raspon od 4-20mA koristi se kako bi se moglo detektirati prekid žice, odnosno 0 mA, ali i zbog toga što se pomoću korištenja otpora od 250Ω vrlo jednostavno napravi konverzija u naponski ulaz raspona 1-5V. Naponski analogni ulazi raspona su 0-5V, 0-10V ili ±10V. Ovakvi ulazi koriste se u naponskim petljama koje se priključuju na ulaz PLC-a. Raspon koji se koristi ovisi o karakteristikama ulaznog signala. Kako bi PLC mogao obraditi signal koji dođe s analognog ulaza signal se mora diskretizirati. Kako bi se to ostvarilo koriste se A/D pretvornici koji uzorkuju ulazni signal i diskretiziraju ga. Kada PLC šalje izlazni signal na analogni izlaz potrebno ga je ponovno pretvoriti iz diskretnog u analogni. Tu se koriste D/A pretvornici. Digitalni ulazi i izlazi su bitniji za ovaj diplomski rad jer se svo upravljanje vrši upravo preko njih. Digitalni ulazi pogodni su za spajanje sklopke, osjetila ili tipkala. Upravo to su digitalni davači pomoću kojih je ostvareno upravljanje pozicijom. Digitalni izlazi koriste se za spajanje pokretača motora ili malih motora. Upravo je motor glavni aktuator procesa opisanog ovim diplomskim radom. Digitalne signale potrebno je prilagoditi tako da budu razumljivi procesorskoj jedinici. Ta prilagodba uključuje optoizolaciju i filtriranje signala. Modul digitalnog ulaza najčešće je 24V DC. Signal digitalnih ulaza i izlaza sprema se kao 8-bitni podatak. Logičke jedinice i nule interpretiraju se na sljedeći način: -30-5V je logička 0, 13-30V je logička 1. Na slici (Slika 3.5.) još možemo vidjeti kako je PLC-u potrebno napajanje kako bi radio, što je i logično.

Osim ovih osnovnih funkcijskih cjelina, PLC-u se mogu dodavati razni moduli. Ovaj modularan dizajn velika je prednost kod složenijih sustava jer se vrlo lako mogu dodavati razni moduli za



neke nove funkcije. Još jedna bitna stvar za navesti je način rada PLC-a. Ovo se može vidjeti na slici 3.6.



*Slika 3.6. Princip rada PLC-a*

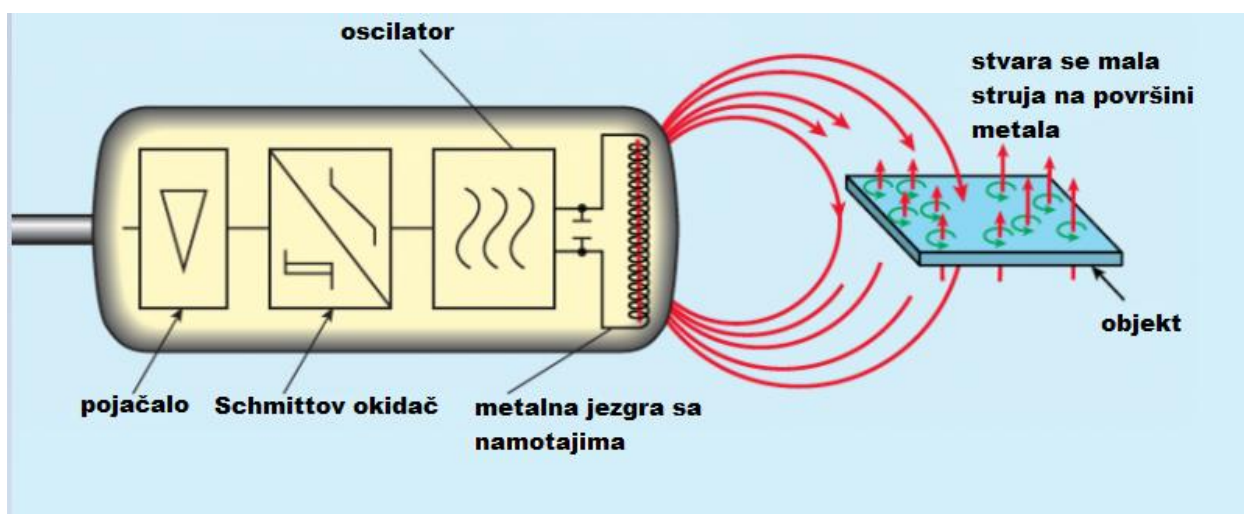
Sada kada je opisana ova središnja jedinica sustava koja prima ulaze i odašilje izlaze u raznim tipovima električkih i elektroničkih signala, te koristi te signale za praćenje i upravljanje bilo kakvim mehaničkim i električkim sustavom, može se nastaviti s opisivanjem ostalih funkcijskih cjelina sustava.

### **3.3. Digitalni davači pozicije i mehanički prekidači**

Digitalni davači pozicije i mehanički prekidači spadaju u skupinu senzora koji se nazivaju senzori neposredne blizine, a naziv su dobili po tome što im je svrha detektirati objekt koji se nalazi u njihovoj neposrednoj blizini. Senzori za detektiranje neposredne blizine objekata mogu raditi na različitim principima, a najčešće se koriste oni koji za detektiranje koriste elektromagnetizam, svijetlo, zvuk ili samo običan mehanizam sklopke. Postoji puno različitih vrsta senzora za svaki od navedenih principa rada i njihovo korištenje je specifično za različita okruženja i primjene [5]. Za primjenu u ovom procesu objasniti će se rad digitalnih davača pozicije koji rade na principu elektromagnetizma te princip rada mehaničkih prekidača.

Digitalni davači koji se zasnivaju na magnetizmu nazivaju se još i induktivni davači. Kako bi detektirali objekt nije potreban kontakt senzora i objekta koji se detektira. Ovakva vrsta senzora može detektirati samo metalne objekte, idealno napravljene od mekog čelika debljine manje od

jednog milimetra [5]. Senzori su izgrađeni od četiri glavne komponente, što se može vidjeti na slici 3.7. Te komponente su: metalna jezgra s namotajima, oscilator, Schmittov okidač i pojačivač izlaza. Oscilator proizvodi simetrično, oscilirajuće magnetno polje koje zrači iz metalne jezgre s namotajima. Kada željezni objekt uđe u to magnetsko polje inducira se mala električna struja na površini metala. Ova struja mijenja prirodnu frekvenciju magnetnog kruga te smanjuje amplitudu oscilacija. Schmittov okidač reagira na promjene u amplitudi i mijenja izlaz senzora. Kada objekt uđe u zonu detekcije izlaz senzora je *on*. Kada nema objekta u zoni detekcije izlaz senzora je *off* [5].



Slika 3.7. Shema i princip rada induktivnog davača pozicije [5]

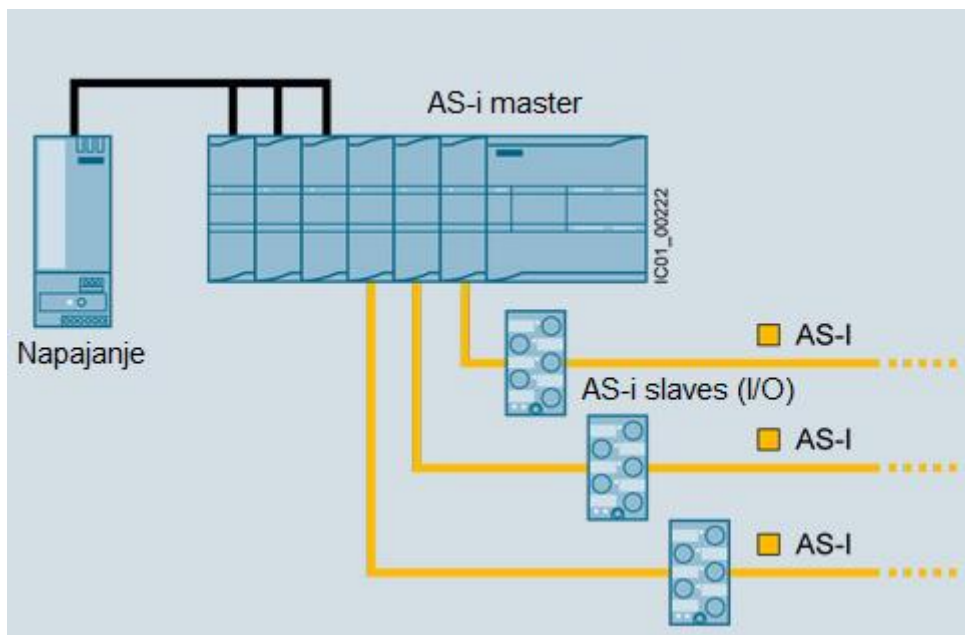
Mehanički senzor radi na vrlo jednostavnom principu. Ovakva vrsta senzora je ništa drugo nego obična sklopka koja služi za sprječavanje objekta da prijeđe određenu maksimalnu granicu. Kada objekt dođe u zonu mehaničkog senzora, on ga svojom kretanjem pritisne, otvarajući ili zatvarajući sklopku. Na taj način se može znati je li objekt došao na poziciju mehaničkog senzora ili nije.

Već je ranije spomenuto da se digitalni davači pozicije i mehaničke sklopke u sustavu upravljanja pozicijom koriste kako bi detektirali objekt na određenoj poziciji. Njihovi izlazi vraćaju se na ulaz PLC-a i njih obrađuje regulator, upravljajući na taj način pozicijom. Induktivni davači koriste se baš za pozicioniranje dok su mehaničke sklopke osiguranje da teretna kolica ne prijeđu maksimalnu poziciju. Može se zaključiti kako su digitalni davači pozicije, kao mjerni članovi sustava upravljanja pozicijom, neizostavni u ostvarenju ovakvog sustava regulacije. O ostalim cjelinama sustava upravljanja pozicijom može se pročitati u nastavku.

### 3.4. AS (engl. *Actuator/Sensor*) sučelje

Aktuator/Senzor sučelje ili skraćeno AS-i je sustav za spajanje nižeg stupnja procesa u sustavima automatske regulacije. Prije pojave AS sučelja, na nižim stupnjevima procesa, mogli su se naći pojasevi kablova. Sada je to zamjenjeno jednim električnim kablom, AS-i kablom. Uređaji koji se mogu spajati preko ovog sučelja su sljedeći: jednostavniji uređaji za upravljanje (enkoderi, tipkala i sl.), trofazni motori promjenjive brzine, svjetlosna prepreka, induktivni davači pozicije ali i neki složeniji uređaji kao što su procesni ventili. AS-i omogućuje, uz prijenos standardnih podataka, prijenos sigurnosnih podataka. To omogućuje, na primjer, da se kablovi namjenjeni za zaustavljanje stroja u nuždi mogu implementirati preko AS-i kabla [6].

AS-i se sastoji od četiri osnovne komponente. Glavna i središnja komponenta ovog sučelja je AS-i *master*. Ova komponenta najčešće služi kao *Gateway* prema višim stupnjevima industrijske mreže ili prema samom PLC-u. Druga komponenta je AS-i *slave*. Najčešće to su I/O moduli na koje se spajaju senzori i aktuatori. Sljedeća komponenta od koje se sastoji AS-i je napajanje. Napajanje se koristi kako bi se napajali AS-i *slaveovi* ali i omogućila komunikacija s AS-i *masterom*. Zadanja komponenta ovog sučelja je jedinstveni plosnati kabal žute boje [7]. Jednostavan prikaz AS sučelja može se vidjeti na slici 3.8.



Slika 3.8. Jednostavan prikaz AS sučelja

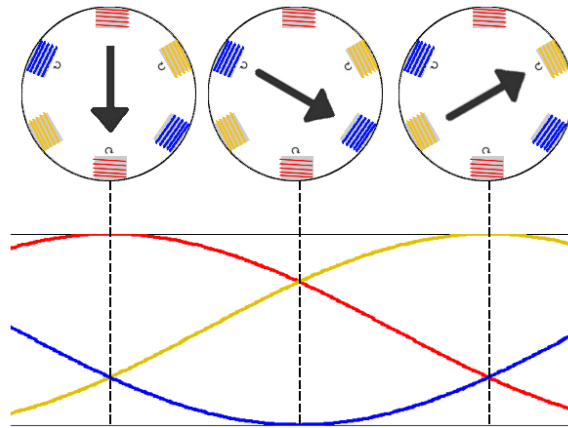
Prema shemi prikazanoj u ranijem poglavlju (Slika 3.4.) može se vidjeti kako je na AS sučelje, kod procesa upravljanja pozicijom, spojen pokretač motora, odnosno uređaj koji regulira performanse električnog motora. Također, na slici (Slika 3.4.) se može vidjeti kako je spojeno dva ulaza i dva izlaza preko AS sučelja. Ulazi su *Ready* i *Running*. *Ready* označava da je motor spreman za uporabu dok *Running* označava da je motor u pokretu. Ova dva ulaza su vrlo bitna jer će biti jedan od temelja za izradu programskog rješenja. Dva izlaza su *Backward* i *Forward*. Ova dva izlaza govore treba li se motor treba kretati naprijed ili nazad. Također su vrlo bitni u kasnijoj izradi programskog rješenja. Na slici (slika 3.4.) može se vidjeti kako je komunikacija ostvarena pomoću PROFIBUS standarda.

Nakon što je objašnjen način na koji je aktuator sustava upravljanja pozicijom spojen s ostatkom sustava ostaje objasniti još samo koji je aktuator korišten u ovom sustavu te neke njegove karakteristike. O ovome se može pročitati u nastavku.

### **3.5. Elektromotor**

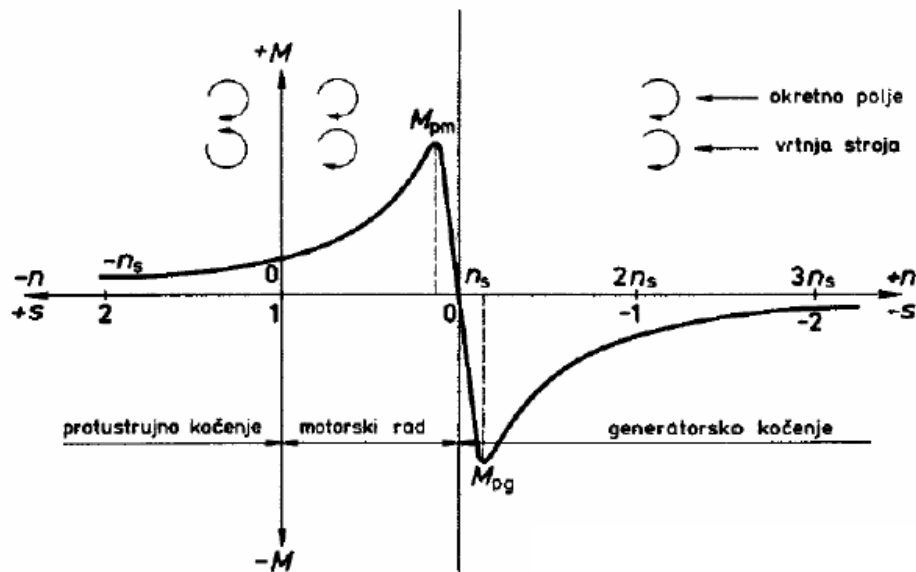
Kao što je ranije navedeno, aktuator sustava upravljanja pozicijom je određena vrsta motora. U svrhu ovog diplomskog rada opisati će se princip rada elektromotra, odnosno asinkronog motora, upravo zato što asinkroni motori, zbog svoje jednostavne konstrukcije, širokog raspona snaga, velike pouzdanosti, niske cijene te ne zahtjevnog održavanja, imaju široku primjenu u industriji. Neke od primjena asinkronih motora su u elektromotornim pogonima, dizalima, ventilatorima, crpkama itd. Razvojem tehnologije u industriji primjena asinkronih motora sve više raste. Pretežno se koriste trofazni motori.

Asinkroni motor je stroj izmjenične struje koji pretvara privedenu električnu energiju u mehaničku energiju na osovini. Ta dobivena mehanička energija predaje se radnom mehanizmu. Okretni moment motora dobiven je pomoću elektromagnetske indukcije, uzrokovane magnetskim poljem dobivenim pomoću namotaja statora. To je zapravo osnovni princip rada elektromotora. Izmjenična struja napajanja stvara magnetno polje koje rotira s oscilacijama izmjenične struje [8]. To se može vidjeti na slici 3.9.



Slika 3.9. Trofazno napajanje omogućuje rotirajuće magnetno polje u motoru [8]

Asinkroni stroj može raditi u tri područja rada. Ta tri područja rada su: motorsko područje rada, generatorsko područje rada i kočno područje rada. Na slici 3.10. može se vidjeti momentna karakteristika asinkronog stroja i moguća područja rada istog.



Slika 3.10. Momentna karakteristika asinkronog stroja [9]

$M_{pm}$  na slici (Slika 3.10) je prekretni moment motora,  $n_s$  je sinkrona brzina, a  $M_{pg}$  prekretni moment generatora. Brzina vrtnje rotora asinkronog motora različita je od brzine vrtnje okretnog polja i mijenja se s opterećenjem (zbog toga i ima naziv asinkroni stroj). Kašnjenje brzine vrtnje rotora za brzinom vrtnje okretnog polja naziva se klizanje i označuje se sa  $s$  [9].

Asinkroni motor sastoji se od dva dijela, statora i rotora. Stator je nepokretni dio i sastoji se od paketa, namota i konstrukcijskih dijelova. Na provrtu cilindričnog paketa izvedeni su utori u kojima je smješten namot. Rotor je pokretni dio i sastoji se od paketa, namota i konstrukcijskih dijelova. Rotor je oblikom cilindričan s utorima na obodu u kojima je namot. Namot rotora, ovisno o vrsti stroja, može biti izoliran (kolutni asinkroni motor) ili neizoliran i kratko spojen (kavezni asinkroni motor) [10].

Elektromotori se najčešće koriste s VFD-ovima (engl. *variable-frequency drive*) onda kada je potreban način rada s promjenjivom brzinom. VFD je tip *drivea* podesive brzine koji se koristi u elektromehaničkim sustavima kako bi se kontrolirala brzina i okretni moment motora mjenjajući ulaznu frekvenciju i napon motora.

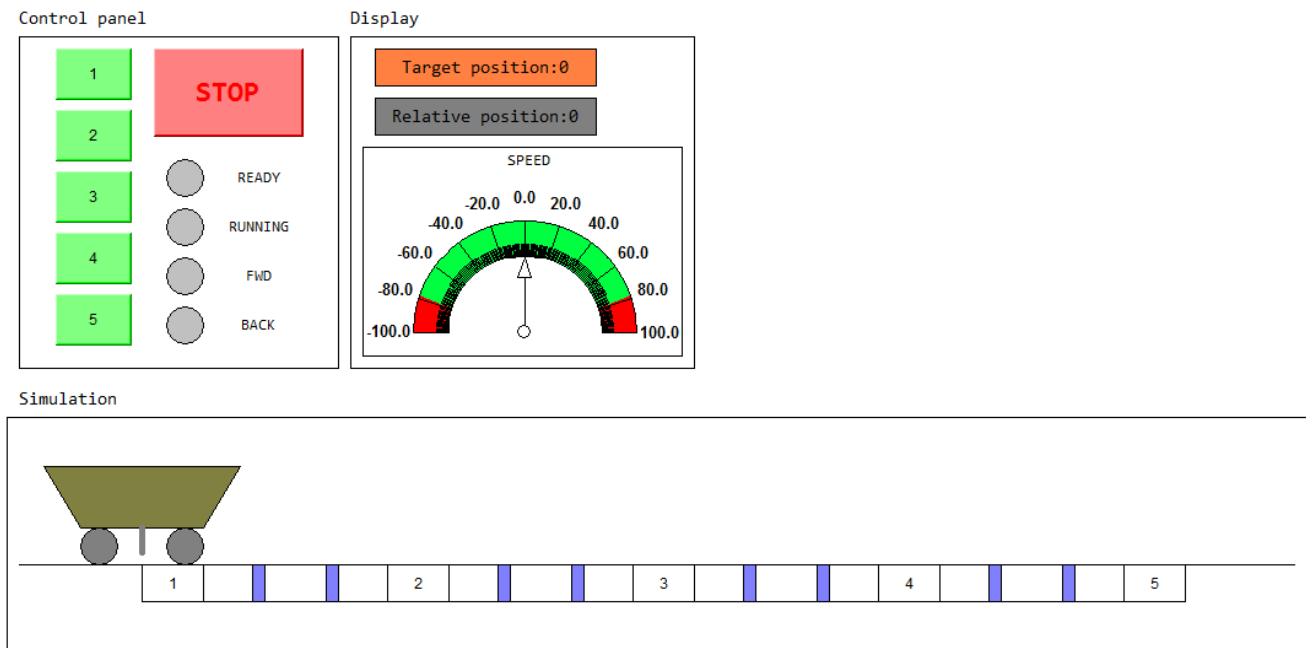
Opisom elektromotara završen je opis jednog industrijskog sustava upravljanja pozicijom. Sada se, kada je opisan jedan ovakav sustav i kada je dana teorijska podloga o svakoj funkcijskoj cjelini sustava upravljanja pozicijom, može pristupiti izradi programskog rješenja. O programskom rješenju simulacije i vizualizacije predhodno opisanog sustava može se pročitati u nastavku.

## 4. PROGRAMSKO RJEŠENJE

Zadatak ovog diplomskog rada bio je ostvariti sustav upravljanja pozicijom, točnije simulaciju i vizualizaciju jednog takvog sustava u programskom paketu Codesys. U industriji, primjena ovog sustava je raznovrsna. Upravo zato, kako bi se ovaj sustav mogao ostvariti, bilo je potrebno odabrati jednu od primjena ovakvog sustava. U ranijim poglavljima opisan je sustav transporta metalnih namotaja na skladište. Detaljno su opisane funkcijske cjeline ovog sustava i dana je dobra teorijska podloga na temelju koje se može projektrirati dobra simulacija i vizualizacija sustava. Proučavanjem i analiziranjem raznih načina kako bi se izveo jedan ovakav sustav upravljanja pozicijom odlučeno je da će izrada programskog rješenja temeljiti na sljedećem sustavu: Sustav upravljanja pozicijom koji za rad koristi pet digitalnih davača pozicije, aktuator (motor) koji pokreće teretna kolica i kontrolnu ploču preko koje se odabire željena pozicija.

Kada se na kontrolnoj ploči pritisne tipka za određenu poziciju, teretna kolica kreću prema toj poziciji s višom brzinom. Kada se teretna kolica dovoljno približe zadanoj poziciji tada kolica usporavaju na nižu brzinu koja se naziva brzina približavanja. Tom nižom brzinom kolica se približavaju zadanoj poziciji i zaustavljaju se kada ih digitalni davač pozicije detektira. Kako bi regulator prepoznao kada su se kolica dovoljno približila zadanoj poziciji (moraju usporiti na brzinu približavanja) uvedeni su dodatni induktivni davači pozicije koji se stavljaju na željenu udaljenost od zadane pozicije kako bi označili ulazak kolica u zonu približavanja i obavjestili regulator da snizi brzinu na brzinu približavanja. Dok su kolica u pokretu potrebno je onemogućiti ponovno odabiranje pozicije sve dok se kolica ne zaustave. Također, na kontrolnoj ploči se nalazi funkcija STOP koja omogućuje zaustavljanje kolica u bilo kojem trenutku.

Na slici 4.1. može se vidjeti prototip ovog sustava koji je kasnije korišten za izradu vizualizacije. Na slici (Slika 4.1.) se može vidjeti kontrolna ploča s koje se mogu odabirati pozicije ili zaustaviti kolica. Pored kontrolne ploče nalazi se prikaz bitnih parametara sustava, a to su brzina i relativna pozicija. Važno je napomenuti da je sustav ostvaren u otvorenoj petlji i u niti jednom trenutku se apsolutna pozicija ne koristi za regulaciju kolica (zato se mora pratiti relativna pozicija). Ispod prikaza parametara i kontrolne ploče, na slici 4.1., može se vidjeti simulacija promjene pozicije. Mogu se vidjeti kolica, induktivni davači pozicije (označeni brojevima) i induktivni davači za usporavanje (obojani plavom bojom).



Slika 4.1. Prototip sustava upravljanja pozicijom

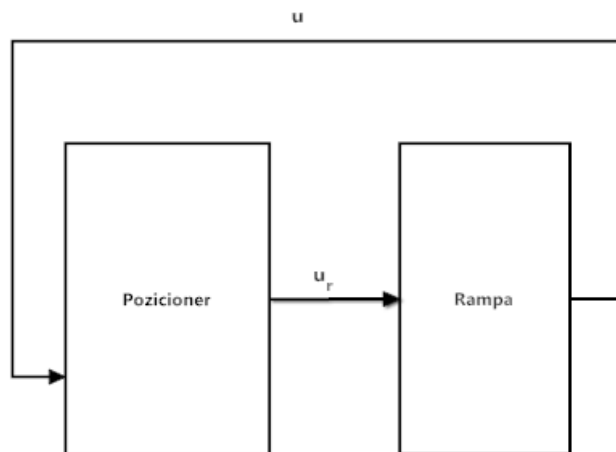
Programsko rješenje simulacije ovog sustava mora osigurati funkcionalnost vizualizaciji prikazanoj slikom 4.1. O izradi simulacije i vizualizacije za ovaj sustav može se pročitati u nastavku.

#### 4.1. Programsko rješenje simulacije

Kao što je ranije navedeno simulacija mora osigurati funkcionalnost vizualizacije. Jednostavno rečeno, kada se stisne određena pozicija, kolica tamo moraju otići. Važno je napomenuti kako je ovo simulacija i sve je odrađeno programski pa je tako uz regulator, koji je inače program koji se vrti na PLC-u, programski simuliran i rad motora, promjena pozicije i digitalni davači pozicije koji su inače dio *hardwarea*. Programsko rješenje odrađeno je uz pomoć programskog paketa Codesys koji omogućuje izradu simulacije ali i vizualizacije.

Prije nego se počne sa samim rješenjem i objašnjenjem programskog koda bitno je objasniti polazišnu točku ovog rješenja. Ranije je već napomenuto kako je upravljanje pozicijom ostvareno pomoću upravljanja bez povratne veze i temelji se na shemi prikazanoj na slici 2.2. Vrlo slično je zamišljena i izvedba programskog rješenja simulacije sustava upravljanja pozicijom. To se može vidjeti na slici 4.2.

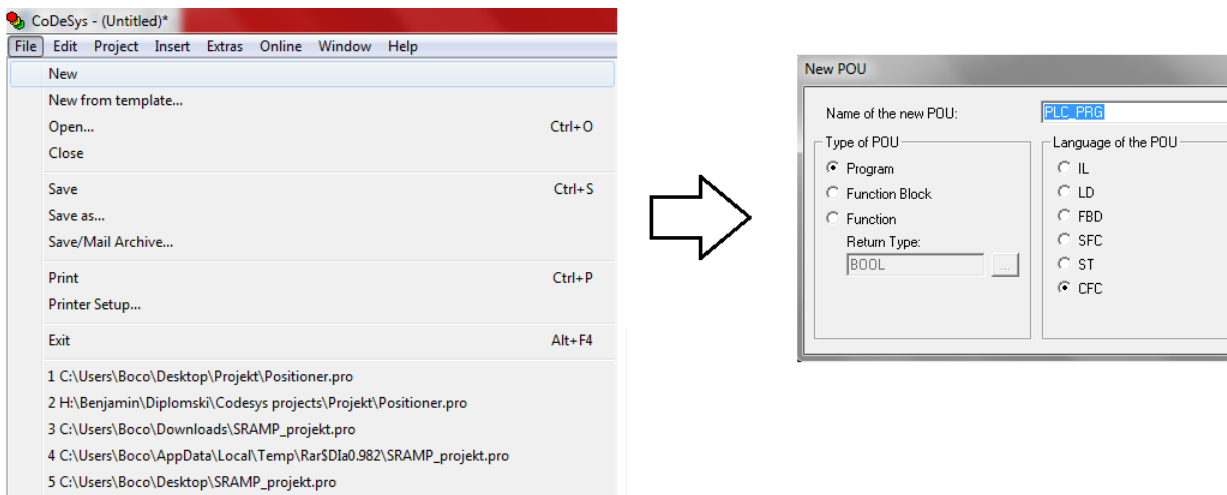




Slika 4.2. Princip rada simulacije

Kao što možemo vidjeti na slici (Slika 4.2.) korištena su samo dva funkcijska bloka kako bi se ostvarila regulacija pozicije. Blok rampa će predstavljati simulaciju rada motora. Vidimo kako je ulaz u ovaj blok upravljački signal koji šalje regulator, a izlaz iz ovog bloka upravljački signal prema procesu promjene pozicije. Funkcijski blok pozicioner zapravo sadrži proces, mjerne članove i regulator u jednom bloku. Može se zaključiti kako je ovo isti tok regulacije kao i kod sustava bez povratne veze samo je u jednom bloku (blok pozicioner) sjedinjeno više funkcijskih cjelina (regulator, proces i mjerni član), dok je aktuator u posebnom bloku (blok rampa). Vrijednost koju regulator šalje aktuatoru, upravljački signal  $u_r$ , biti će referentna brzina. Izlaz iz aktuatora, signal  $u$ , biti će stvarna brzina motora. Stvarna brzina motora biti će korištena za izračunavanje apsolutne pozicije kolica. Apsolutna pozicija kolica računa se samo u svrhu simulacije kretanja kolica, a ne u svrhu regulacije. Kako bi se moglo simulirati kretanje kolica, ali i trenutak kada kolica prelaze preko digitalnih davača, mora se u svakom trenutku znati pozicija kolica. Kada se mijenja izlaz mjernih članova (trenutak kada kolica prelaze preko jednog od njih), to detektira regulator i prema svom algoritmu šalje upravljački signal motoru da uspori, ubrza ili stane. Na ovaj način ostvarena je regulacija promjene pozicije.

Kao što je ranije navedeno, simulacija je ostvarena pomoću programskog paketa Codesys koji omogućuje izradu i vizualizacije i simulacije. Kada se otvori Codesys potrebno je ići na *File->New* kao što je prikazano na slici 4.3.

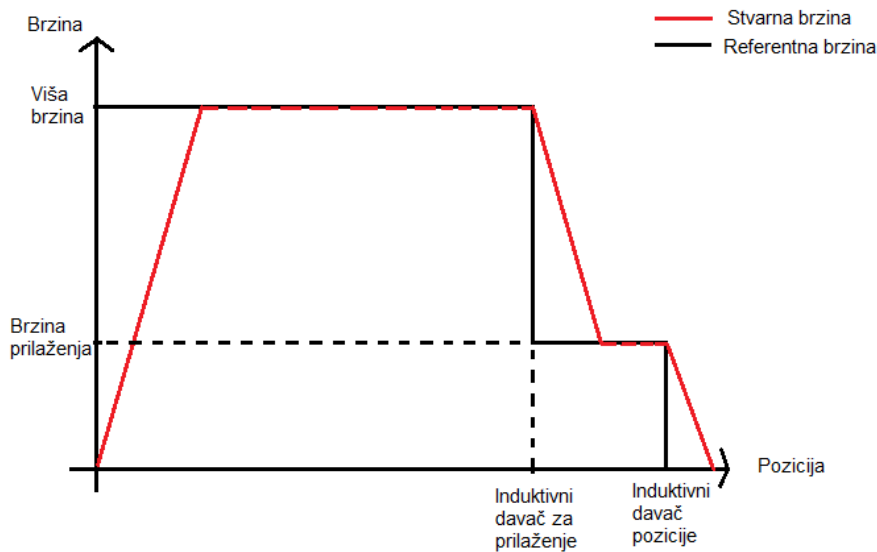


Slika 4.3. Stvaranje novog projekta o Codesys-u

Kao što se može vidjeti na slici 4.3., kada se stvori novi projekt program automatski stvara novi POU (engl. *Program Organization Unit*). Unutar POU-a se organizira program. Svaki POU se sastoji od tijela gdje se piše program u jednom od programskih jezika (podržani su IL, ST, SFC, FBD, LD ili CFC programski jezici) i dijela za deklaraciju varijabli. POU koji je predefiniran naziva se PLC\_PRG i on je glavni program koji svaki projekt mora sadržavati i unutar kojeg moraju biti pozvani svi ostali programi. POU može biti program, funkcija ili funkcijski blok. U izradi ovog projekta korištena su dva funkcijska bloka, jedan za izradu pozicionera i jedan za izradu rampe, te jedan program koji se izvršava 10ms i nazvan je INTERRUPT. Zašto je potrebno uniformno izvođenje programa u trajanju od 10ms biti će objašnjeno kasnije.

Sada kada se stvorio novi projekt i pripadajući programi i funkcijski blokovi, može se krenuti na pisanje programskog koda. Već je ranije objašnjen princip rada ove simulacije. Program će biti podijeljen u dva funkcijska bloka, pozicioner i rampa. Rampa će predstavljati simulaciju aktuatora a pozicioner će predstavljati simulaciju procesa, mjernih članova i regulatora. Prvo će se objasniti funkcijski blok rampa.

Funkcijski blok rampa, kao što je pokazano na slici 4.2., mora primiti jedan ulaz, a to je referentna brzina koju određuje regulator. Isto tako može se vidjeti kako je izlaz ovog bloka stvarna brzina motora. Kako se stvarna brzina mora ponašati u odnosu na referentnu brzinu može se vidjeti na slici 4.4.



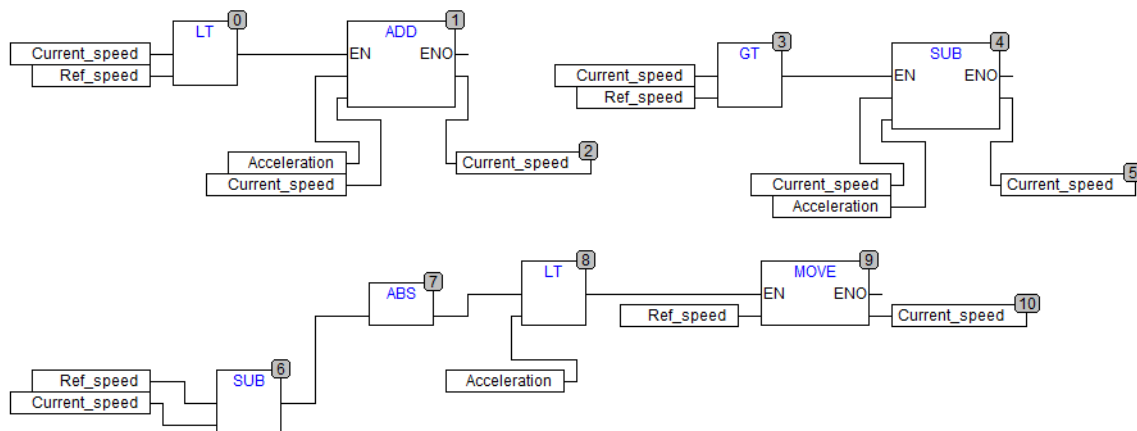
*Slika 4.4. Odnos referentne i stvarne brzine*

Na slici (Slika 4.4.) može se vidjeti kako se rampa mora ponašati. Kada na ulazu dobije vrijednost neke referentne brzine, rampa mora povećavati vrijednost stvarne brzine sve dok ne poprimi vrijednost referentne brzine. Na taj način se simulira ubrzavanje motora. Analogno tome vrijedi i za usporavanje. Još je jedan parametar, uz referentnu i stvarnu brzinu, bitan za ostvarenje ovog funkcijskog bloka. To je akceleracija. Akceleracija predstavlja ubrzanje motora. Ona određuje koliko brzo će rasti i padati stvarna brzina kako bi dosegla referentnu brzinu. Što je akceleracija veća porast stvarne brzine biti će strmiji. Na slici 4.5. možemo vidjeti kako je programski izveden ovaj funkcijski blok. Na slici 4.5. može se vidjeti kako su u dijelu za deklaraciju varijabli deklarirana tri ulaza. Ulazi u ovaj funkcijski blok su akceleracija koja se može mijenjati i zadavati po potrebi, referentna brzina koju rampa u svakom trenutku dobiva iz funkcijskog bloka pozicionera, te trenutna brzina koja se vraća s izlaza rampe nazad na ulaz rampe. Funkcijski blok rampa sadrži još jednu definiranu varijablu, a to je trenutna brzina definirana kao izlaz. Kao što je ranije navedeno ona se s izlaza rampe vraća na ulaz rampe radi provjere ali se šalje i na ulaz pozicionera.

Ukoliko se pogleda programski kod za rampu primijetiti će se da je on vrlo jednostavan. On se nalazi ispod varijabli u dijelu gdje se piše programski kod. Programski kod pisan je blokovskim prikazom koji se naziva CFC. Na slici (Slika 4.5.) mogu se vidjeti tri skupine blokova. Algoritam rampe u svakom trenutku uspoređuje trenutnu brzinu i referentnu brzinu. Ukoliko je trenutna brzina manja od referentne brzine, tada je potrebno zbrajati akceleraciju na trenutnu brzinu kako

bi trenutna brzina rasla i dosegla referentnu brzinu. Ukoliko je trenutna brzina veća od referentne brzine, tada je potrebno oduzimati akceleraciju kako bi trenutna brzina padla i opet dosegla referentnu brzinu. Na kraju, potrebno je napraviti jedno osiguranje da sustav ne oscilira. Naime, stalnim zbrajanjem akceleracije akumulira se mala pogreška. Zbog toga se trenutna brzina nikada neće izjednačiti s referentnom brzinom i rampa će stalno ubrzavati i usporavati (po malo) stvarajući oscilacije. Ovo se može dogoditi i ukoliko akceleracija nije broj koji će zbrajanjem dati neki okrugli broj. Ovo se riješilo na način da se stalno prati razlika između referentne brzine i stvarne brzine sustava. Kada ta razlika bude manja od vrijednosti akceleracije potrebno je izjednačiti trenutnu i referentnu brzinu. Kada referentna brzina i trenutna brzina imaju istu vrijednost nema više ubrzavanja ili usporavanja te nema niti oscilacija.

	VAR	VAR_INPUT	VAR_OUTPUT	VAR_IN_OUT	CONSTANT	RETAIN	INFO
	Name	Address	Type	Initial	Comment		
0001	Acceleration		REAL	0.2			
0002	Current_vel		REAL	0			
0003	Ref_speed		REAL				



Slika 4.5. Programski kod za funkcijski blok rampe

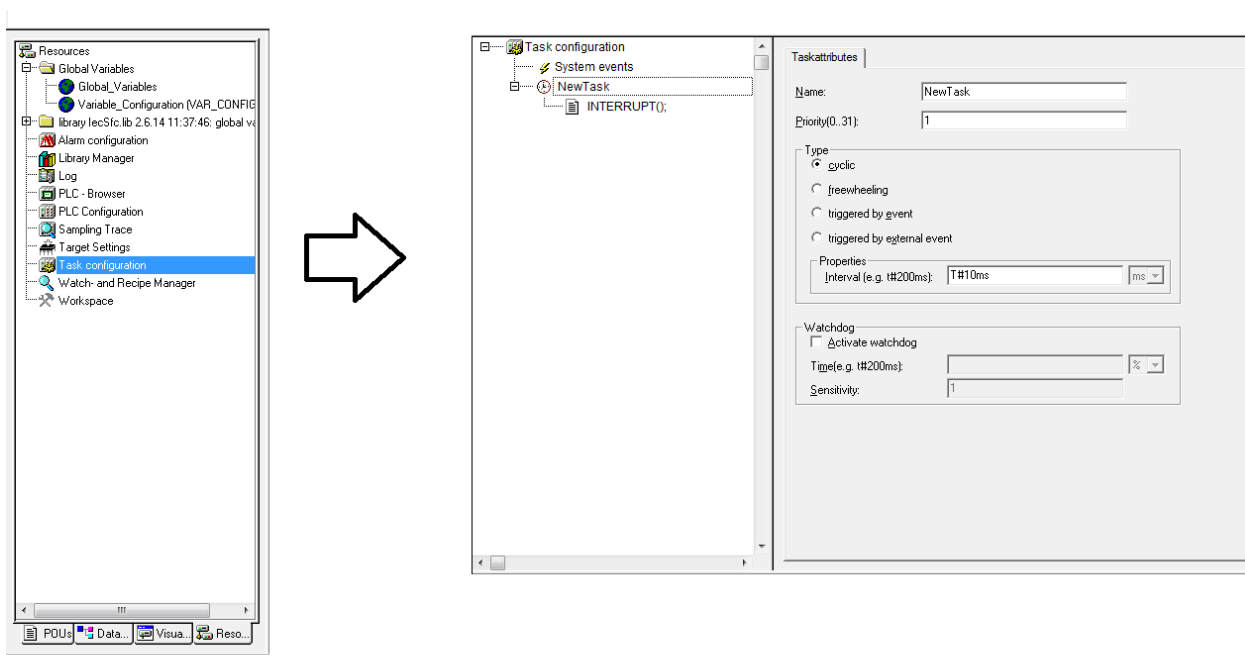
Funkcijskim blokom rampe ostvaren je aktuator sustava (motor). Može se primjetiti da kada se akceleracija zbraja motor ubrzava a kada se akceleracija oduzima tada usporava. Motorom upravlja upravljački signal regulatora. Regulator svojim algoritmom određuje referentnu brzinu i šalje ju motoru koji se ponaša prema tom upravljačkom signalu. Motor ubrzava ili usporava te šalje svoju brzinu nazad funkcijskom bloku pozicionera. Sada je potrebno opisati funkcijski blok pozicionera. O tome se može pročitati u nastavku.

Kao što je i ranije navedeno, funkcijski blok pozicionera sadrži proces, mjerne članove i regulator. Kada se govori o procesu misli se na proces promjene pozicije. Način da se simulira ovaj proces vrlo je jednostavan. Potrebno je samo u svakom trenutku izračunavati poziciju i ta pozicija će određivati gdje se u danom trenutku nalaze kolica. Pozicija teretnih kolica uvjetovana je brzinom kojom motor pokreće ta kolica. Trenutna brzina kolica je zapravo upravljački signal  $u$  koji je pobuda procesu. Kao vremenska posljedica djelovanja brzine na proces dobije se izlazni signal iz procesa  $Y_s$ .  $Y_s$  je zapravo promjenjena pozicije. Računanje pozicije određeno je sljedećom formulom:

$$\text{trenutna pozicija} = \text{trenutna brzina} * \text{vrijeme izvođenja programa} * K \quad (4-1)$$

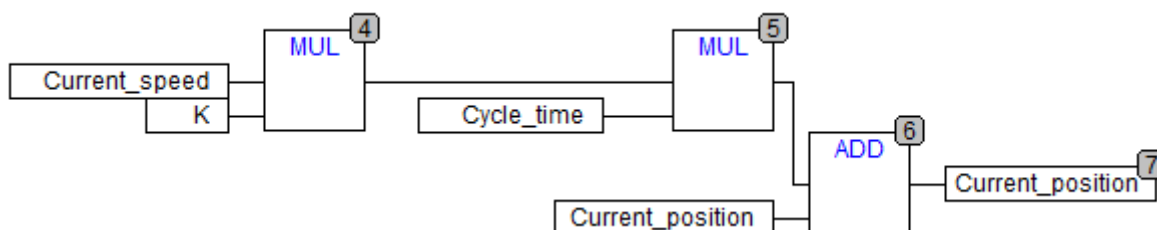
Prema formuli (4-1) može se vidjeti kako trenutna pozicija ovisi o trenutnoj brzini motora, vremenu izvođenja programa i koeficijentu  $K$  koji predstavlja zajednički koeficijent za razne utjecaje koji utječu na promjenu pozicije. Pod koeficijent  $K$ , npr., spada koeficijent mehaničkog prijenosa s osovine motora, težine, trenja i sl. Mijenjanjem koeficijenta  $K$  mogu se simulirati određeni uvjeti.

Djelovanjem brzine u vremenu mijenja se pozicija. Upravo zato, množi se trenutna brzina s vremenom izvođenja programa. Kada se govori o vremenu izvođenja programa sada se može vratiti na priču vezanu uz POU koji je nazvan INTERRUPT. Upravo ovom programu je namješteno izvođenje na točno 10ms. Važno je napomenuti kako se, ukoliko se ne namjesti vrijeme izvođenja programa, program izvodi ciklički u onom vremenu koliko mu je dovoljno da izvede sve funkcije zapisane u njemu. To znači da kako program raste ili se mijenja, mijenja se i vrijeme izvođenja programa. To znači da bi trenutna pozicija ovisila o veličini napisanog programa. Upravo zbog toga je stavljeno da se POU pod nazivom INTERRUPT izvršava točno 10ms. Sada pozicija ne ovisi o veličini programa već se izračunava uvijek jednako. Na slici 4.6. može se vidjeti na koji način se postavlja vrijeme izvođenja programa. Potrebno je odabrati *Task configuration* i dodati novi zadatak kojem se postavi interval izvođenja na 10ms. Nakon toga potrebno je dodati željeni POU u novostvoreni zadatak.



Slika 4.6. Način postavljanja vremena izvođenja programa

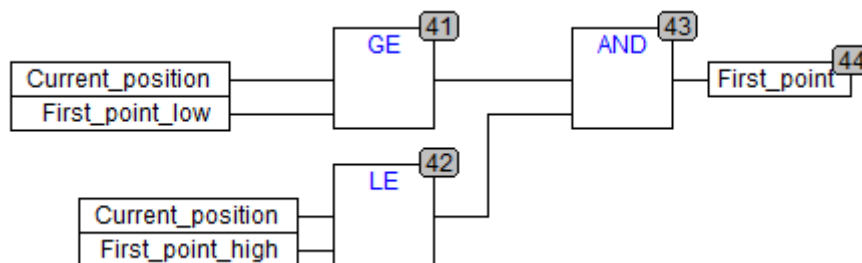
Ovo programsko rješenje računanja pozicije vrlo je jednostavno. Potrebna su dva bloka za množenje koji će množiti trenutnu brzinu, koeficijent K i vrijeme izvođenja programa. Rezultat se stalno dodaje na trenutnu poziciju. Ovaj programski dio može se vidjeti na slici 4.7.



Slika 4.7. Izračunavanje trenutne pozicije

Izračunavanjem trenutne pozicije prema prethodno opisanom načinu riješen je proces promjene pozicije. Promjenom brzine motora (aktuatora) mijenja se pozicija (izlaz iz procesa). S prethodna dva rješenja konstruirani su aktuator i proces sustava. Ostaje još konstruirati regulator i mjerne članove. O tome se može pročitati u nastavku.

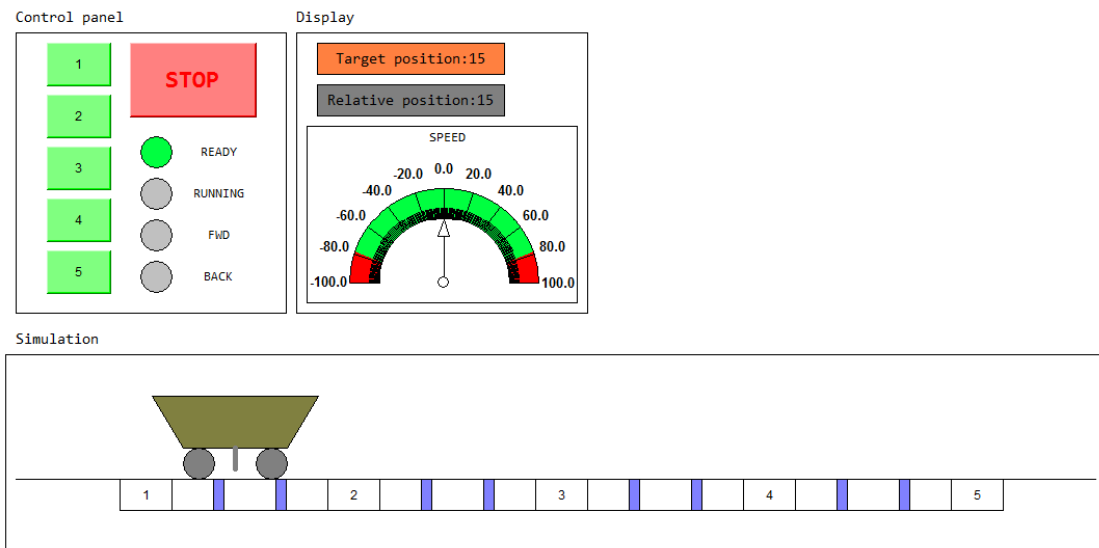
Sljedeća komponenta koja je dio funkcijskog bloka pozicionera su mjerni članovi. Ranije je već spomenuto kako mjerne članove predstavljaju induktivni davači pozicije. Potrebno je definirati pet induktivnih davača za detekciju pozicije i induktivne davače za detekciju približavanja i usporavanja na manju brzinu. Važno je napomenuti kako induktivni davač u stvarnom svijetu ima svoju širinu, odnosno zauzima određen prostor duž puta teretnih kolica. Drugim riječima nije moguće definirati induktivni davač kao jednu točku u prostoru. Potrebno je definirati njegovu širinu. U ovom programskom rješenju to je ostvareno tako da su zadane lijeva i desna točka induktivnog davača. Ukoliko se pozicija kolica trenutno nalazi između lijeve i desne točke induktivnog davača to označuje da se kolica nalaze iznad tog davača i da je potrebno detektirati prisutnost kolica. U programskom rješenju, koje se može vidjeti na slici 4.8., lijeva točka prvog davača označena je s *First\_point\_low* dok je desna strana označena s *First\_point\_high*. Na slici (slika 4.8) se također može vidjeti kako će prvi pokazivač (*First\_point*) biti u logičkoj jedinici ako je trenutna pozicija veća od *First\_point\_low* (lijeve točke) a manja od *First\_point\_high* (desne točke). Taj događaj označuje da su se kolica pozicionirala točno iznad induktivnog davača i on sada to detektira. Ovim rješenjem dana je funkcionalnost mjernim članovima. Analogno programskom rješenju sa slike 4.8. napravljeno je rješenje za svih ostalih dvanaest induktivnih davača.



Slika 4.8. Program za funkcionalnost induktivnih davača

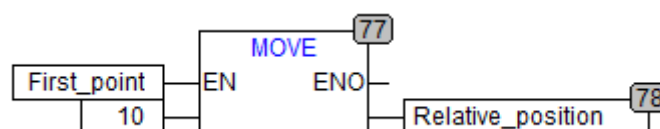
Sada kada su mjerni članovi funkcionalni potrebno je dati funkcionalnost regulatoru. Regulator je zapravo najvažniji dio ovog sustava jer bez njega upravljanje ne bi bilo moguće. O programskom rješenju regulatora može se pročitati u nastavku.

U suštini regulator bi morao pratiti signale induktivnih davača i prema njima ubrzavati ili usporavati motor. No, još je jedna vrlo bitna stvar koju regulator u ovom slučaju mora raditi, a to je praćenje relativne pozicije. Kako je ranije navedeno, u sustav je ugrađena tipka STOP pomoću koje se može zaustaviti kolica u bilo kojem trenutku. To znači da kolica mogu, u nekom trenutku, stajati na bilo kojoj poziciji između digitalnih davača pozicije. Ovu situaciju može se vidjeti na slici 4.9.



Slika 4.9. Kolica stoje između digitalnih davača pozicije

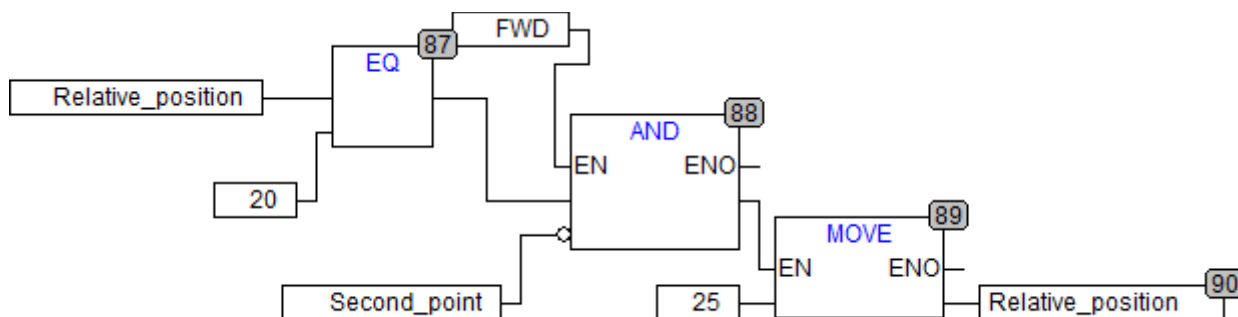
U ovoj situaciji niti jedan davač pozicije ne daje signal da se kolica nalaze nad njim. U ovom trenutku kolica ne postoje (regulator ne zna gdje su). No, ukoliko se prati relativna pozicija, tada regulator može ustvrditi da kolica jesu tu i to negdje između prvog i drugog induktivnog davača. Sada regulator zna, ukoliko se ponovno stisne jedna od tipki, gdje kolica moraju ići (naprijed ili nazad). Relativna pozicija u ovom programskom rješenju poprima vrijednost 10 ako su kolica pozicionirana iznad prvog digitalnog davača, 15 ukoliko se nalaze između prvog i drugog digitalnog davača, 20 ukoliko se nalaze iznad drugog digitalnog davača itd. Način na koji je to ostvareno može se vidjeti na slici 4.10.



Slika 4.10. Određivanje relativne pozicije kada su kolica pozicionirana na digitalnim davačima

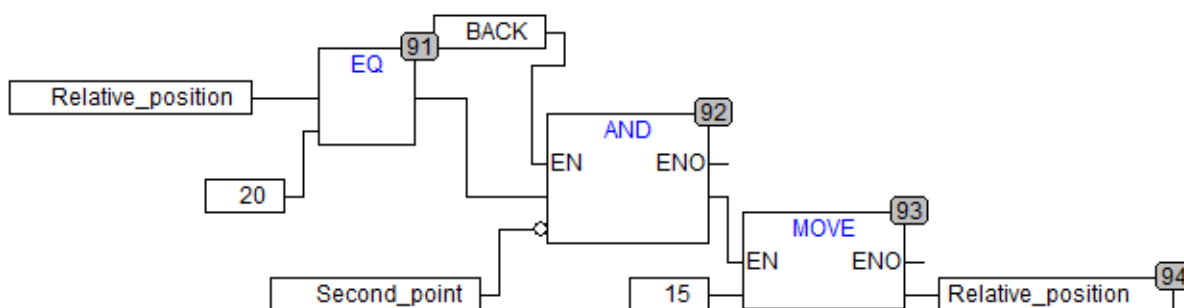


Na slici (slika 4.10) može se vidjeti da relativna pozicija poprima vrijednost 10 ukoliko je aktivan prvi digitalni davač pozicije. Analogno ovome napisan je program i za ostale pozicije. Ukoliko se teretna kolica nalaze između pozicija tada je programski kod malo drugačiji i izgleda kao na slici 4.11.



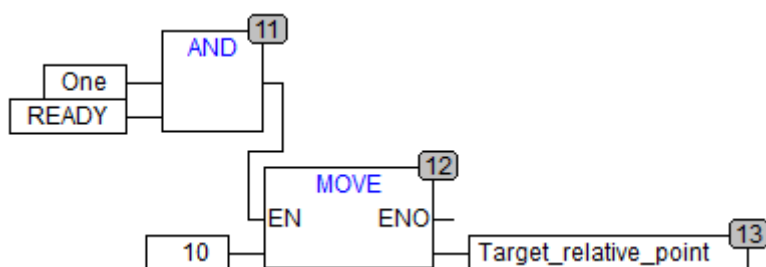
Slika 4.11. Određivanje relativne pozicije kada kolica nisu pozicionirana na digitalnim davačima (smjer unaprijed)

Na slici (Slika 4.11.) može se vidjeti kako će relativna pozicija poprimiti vrijednost 25 ukoliko se kolica kreću naprijed (FWD), ukoliko je prijašnja vrijednost relativne pozicije bila 20 i ukoliko kolica napuste poziciju drugog digitalnog davača. Analogno tome napravljeno je za sve ostale međupozicije kada je smjer kolica naprijed, ali i za sve međupozicije kada je smjer unazad. Važno je napomenuti da ukoliko su teretna kolica napustila drugi digitalni davač i ukoliko je prethodna vrijednost relativne pozicije bila 20 (isti uvjeti kao i na slici 4.11.), ali kolica sada idu unazad (BACK), vrijednost relativne pozicije mora biti jednaka 15. To se može vidjeti na slici 4.12.



Slika 4.12. Određivanje relativne pozicije kada kolica nisu pozicionirana na digitalnim davačima (smjer unazad)

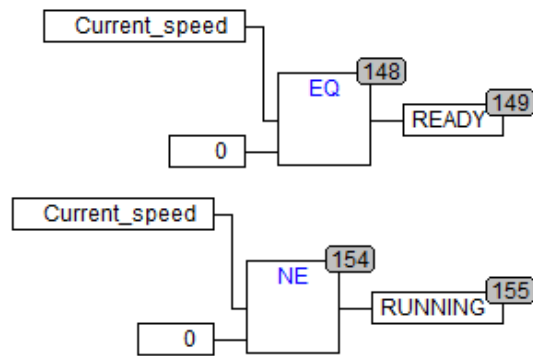
Sada kada se relativno zna gdje su kolica pozicionirana u svakom trenutku potrebno je definirati poziciju koja se može zadati s kontrolnog panela, određišanu poziciju. Odredišna pozicija označava poziciju gdje se želi pozicionirati teretna kolica. To je zapravo ulaz u sustav i vodeća vrijednost sustava. Ranije je već spomenuto da se ova vrijednost odabire preko kontrolnog panela. Ova vrijednost, kao i relativna pozicija, poprima relativne vrijednosti. Ukoliko se želi da teretna kolica dođu na prvi digitalni davač tada će vrijednost odredišne pozicije biti 10, ukoliko se želi doći na drugi digitalni davač 20 itd. Ovaj programski kod može se vidjeti na slici 4.13.



Slika 4.13. Određivanje odredišne relativne pozicije

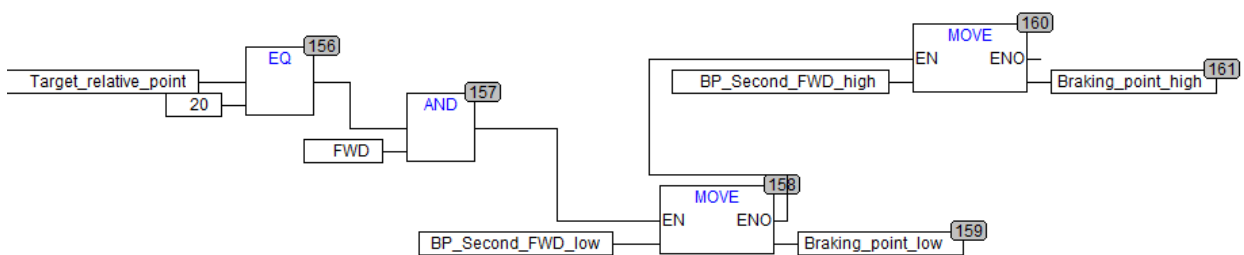
Na slici (Slika 4.13.) može se vidjeti da će odredišna reltivna pozicija biti jednaka 10 ukoliko se stisne prva tipka na kontrolnoj ploči (*One*) i ukoliko je sustav spreman za rad (*Ready*), odnosno motor ne radi (nije *Running*). Analogno ovome napisan je program za sve pozicije.

Ranije je spomenuto kako se treba onemogućiti izbor pozicije s kontrolne ploče ukoliko motor već radi. Za to nam služe kontrolne vrijednosti *Ready* i *Running*. Kada se ranije pričalo o AS sučelju spomenute su upravo ove dvije vrijednosti koje PLC prima preko AS sučelja od *drivea* motora. Ove dvije vrijednosti programski su riješene vrlo jednostavno. Ukoliko je trenutna brzina jednaka nula tada kolica stoje i *Running* je u logičkoj 1. Ukoliko je trenutna brzina različita od nula tada je *Running* u logičkoj 1. To se može vidjeti na slici 4.14.



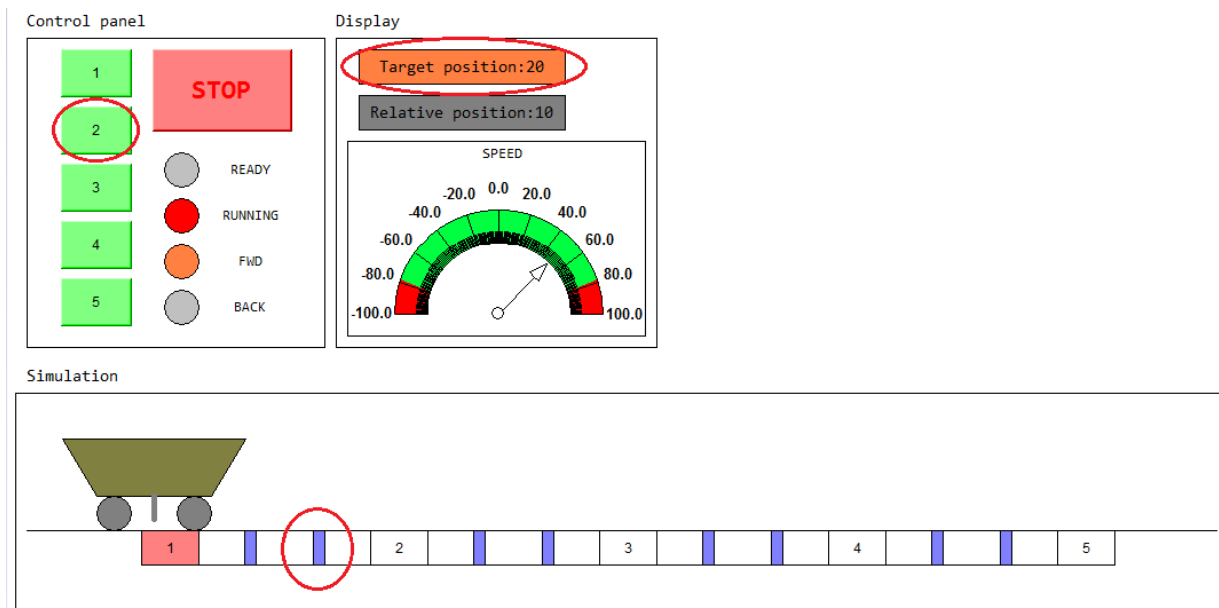
Slika 4.14. Postavljanje kontrolnih vrijednosti Ready i Running

Sada kada se prati odredišna pozicija moguće je programirati regulator da prepozna koji će od induktivnih davača namjenjenih za usporavanje (približavanje) odabrati da zakoči motor do brzine približavanja. To se može vidjeti na slici 4.15.



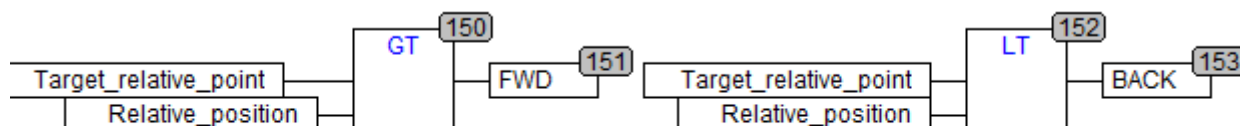
Slika 4.15. Program za odabiranje pravog induktivnog davača za kočenje

Na slici (Slika 4.15.) može se vidjeti da, ukoliko je odabrana druga pozicija (Odredišna pozicija tada ima vrijednost 20) i ukoliko kolica idu naprijed, točka kočenja (*Braking\_point*) postaje induktivni davač s lijeve strane druge pozicije. Ovo se može vidjeti i na slici 4.16. Na slici (Slika 4.16.) može se vidjeti da je odabrana druga pozicija (zaokruženo crveno na kontrolnoj ploči). Kada je odabrana druga pozicija, odredišna pozicija je poprimila vrijednost 20 (zaokruženo crveno na prikazu parametara). Ukoliko je odredišna pozicija jednaka 20 i kolica idu naprijed, digitalni davač koji će regulator odabrati je prvi lijevo od druge pozicije (zaokruženo crveno na simulaciji).



Slika 4.16. Odabiranje pravog induktivnog davača za kočenje

Kreću li se kolica naprijed ili nazad definirano je s vrijednostima pod nazivom *FWD* i *BACK*. Način na koji se određuju ove dvije vrijednosti može se vidjeti na slici 4.17.



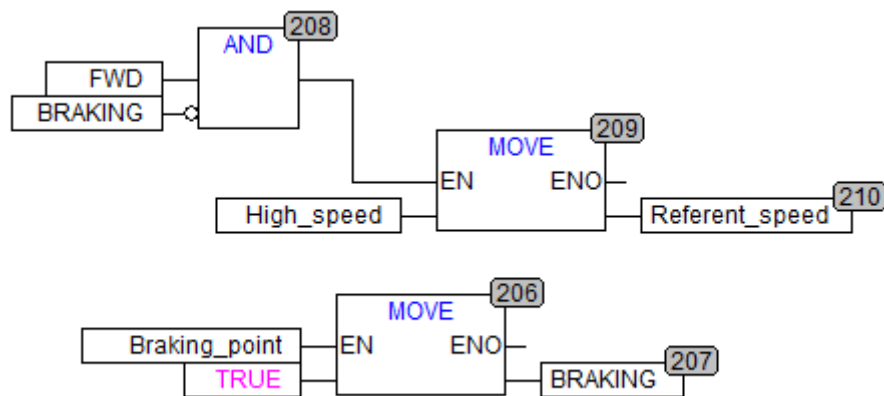
Slika 4.17. Određivanje vrijednosti za *FWD* i *BACK*

Na slici (Slika 4.17.) može se vidjeti da kolica idu naprijed (*FWD*) ukoliko je vrijednost određene pozicije veća od relativne pozicije. Kolica idu unazad (*BACK*) ukoliko je vrijednost određene pozicije manja od relativne pozicije.

Trenutnim programskim rješenjem funkcijskog bloka pozicionera definirani su svi mjerni članovi i njihove pozicije (5 digitalnih davača za poziciju i 8 digitalnih davača za usporavanje). Također, omogućeno je praćenje relativne pozicije pomoću koje regulator u svakom trenutku relativno zna gdje se nalaze kolica. Ovim programskim rješenjem omogućen je unos određene pozicije preko kontrolnog panela. Određena pozicija poprima vrijednost ovisno o tipki koja je pritisnuta. Ovisno

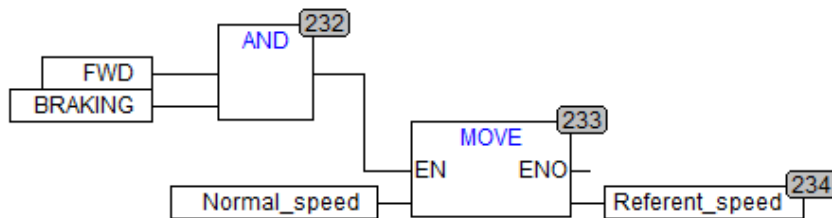
o tipki koja je pritisnuta regulator odabire i pripadajući induktivni davač za kočenje. Također je definirano, na temelju usporedbe zadane i trenutne pozicije, kretanje kolica naprijed ili nazad. Još je ostalo definirati na koji način regulator mjenja upravljačku veličinu prema aktuatoru, odnosno na koji način mjenja referentnu brzinu.

Odabiranje referentne brzine izvedeno je vrlo jednostavno. To je može vidjeti na slici 4.18.



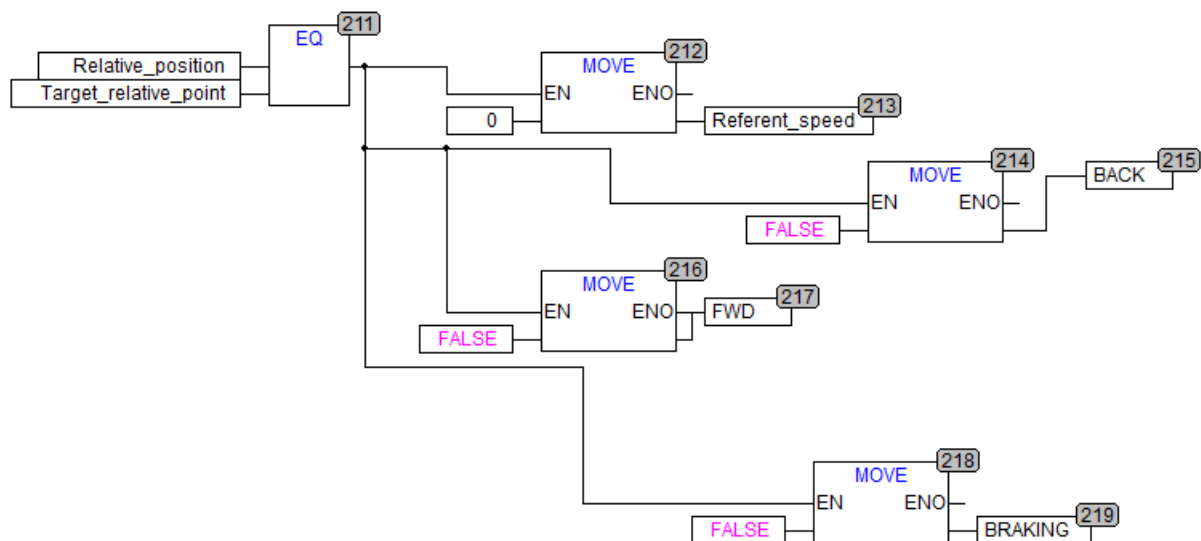
Slika 4.18. Odabiranje referentne brzine

Na slici (Slika 4.18.) se može vidjeti kako referentna brzina poprima vrijednost visoke brzine (*High\_speed*) sve dok kolica idu naprijed (*FWD*) i nisu pregazila preko induktivnog davača za kočenje odabranog od strane regulatora (*BRAKING*). Ukoliko kolica pregaze preko odabranog induktivnog davača namjenjenog za kočenje tada će referentna brzina poprimiti vrijednost brzine približavanja (*Normal\_speed*). To se može vidjeti na slici 4.19.



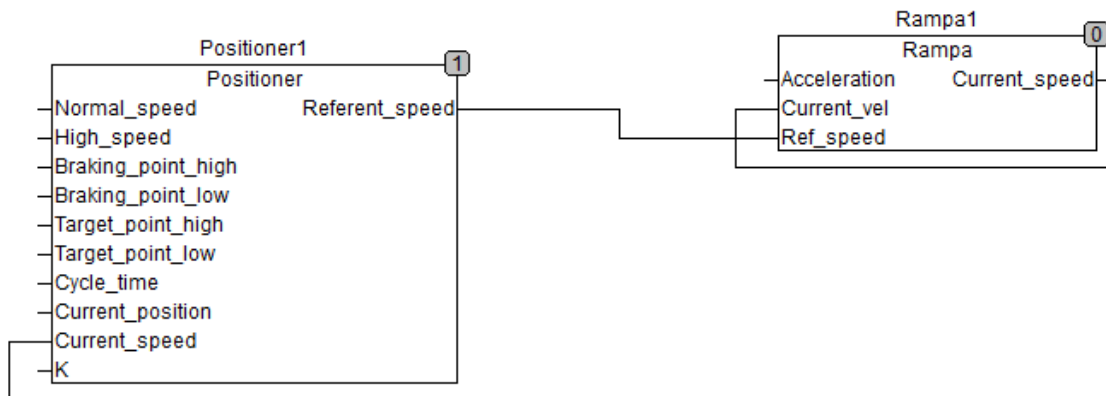
Slika 4.19. Referentna brzina poprima vrijednost brzine približavanja

Ukoliko su relativna pozicija i odredišna pozicija jednake to znači da su kolica došla na zadanu poziciju. Tada je vrijednost referentne brzine jednaka nula jer kolica moraju stati. Ovo se može vidjeti na slici 4.20.



Slika 4.20. Vrijednost referentne brzine jednaka je nula ukoliko su kolica došla na odredište

Na ovaj način ostvarena je funkcionalnost odabiranja referentne brzina koja se kao upravljački signal šalje motoru. Sada kada je ostvarena funkcionalnost funkcijskog bloka pozicionera, potrebno je još samo pozvati funkcijski blok rampe i pozicionera, te ih spojiti kako bi funkcionirali zajedno. Oba dva funkcijska bloka pozvana su unutar POU-a pod nazivom INTERRUPT koji je kasnije pozvan unutar glavnog POU-a pod nazivom PLC\_PRG. Ovo pozivanje i spajanje funkcijskih blokova može se vidjeti na slici 4.21. Na slici se može vidjeti kako je izlaz rampe (trenutna brzina) spojen na ulaz pozicionera. Također se može vidjeti kako pozicioner šalje referentnu brzinu rampi.

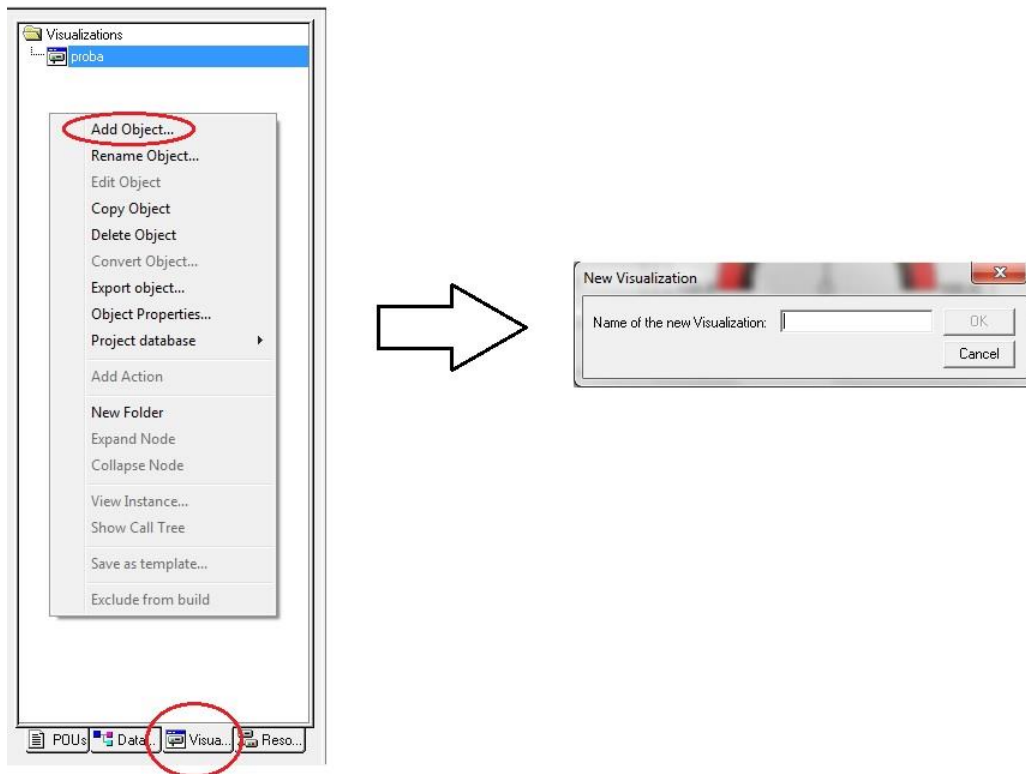


Slika 4.21. Pozivanje funkcijskih blokova u POU

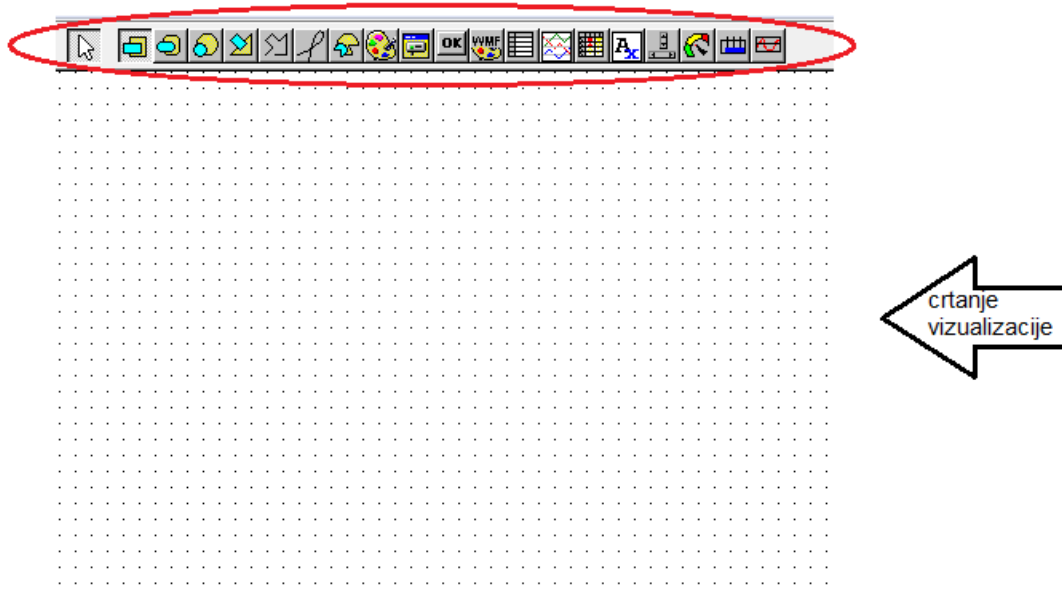
Na ovaj način opisano je programsko rješenje simulacije sustava upravljanja pozicijom. Sljedeća stvar koju je potrebno ostvariti je vizualizacija ovog sustava. O ovome se može pročitati u nastavku.

## 4.2. Programsko rješenje vizualizacije

Kada program ima svoju funkcionalnost, vizualizaciju je vrlo jednostavno napraviti. Kako bi se stvorila nova vizualizacija u Codesysu potrebno je pratiti korake kao na slici 4.22. Na toj slici može se vidjeti kako je potrebno otići na izbornik *Visualization*, stisnuti desnu tipku i dodati novi objekt. Potrebno je još nazvati vizualizaciju određenim imenom i proces stvaranja nove vizualizacije je gotov. Kada se stvori nova vizualizacija pojavi se prazna stranica na koju se mogu crtati razni objekti. Objekti koji se mogu crtati su razni oblici, poligoni, linije, tipke, prikazi itd. Ovi oblici mogu se birati iz izbornika iznad vizualizacije. To se može vidjeti na slici 4.23. Na slici 4.23. izbornik s oblicima zaokružen je crveno dok mjesto za crtanje vizualizacije označuje strjelica.



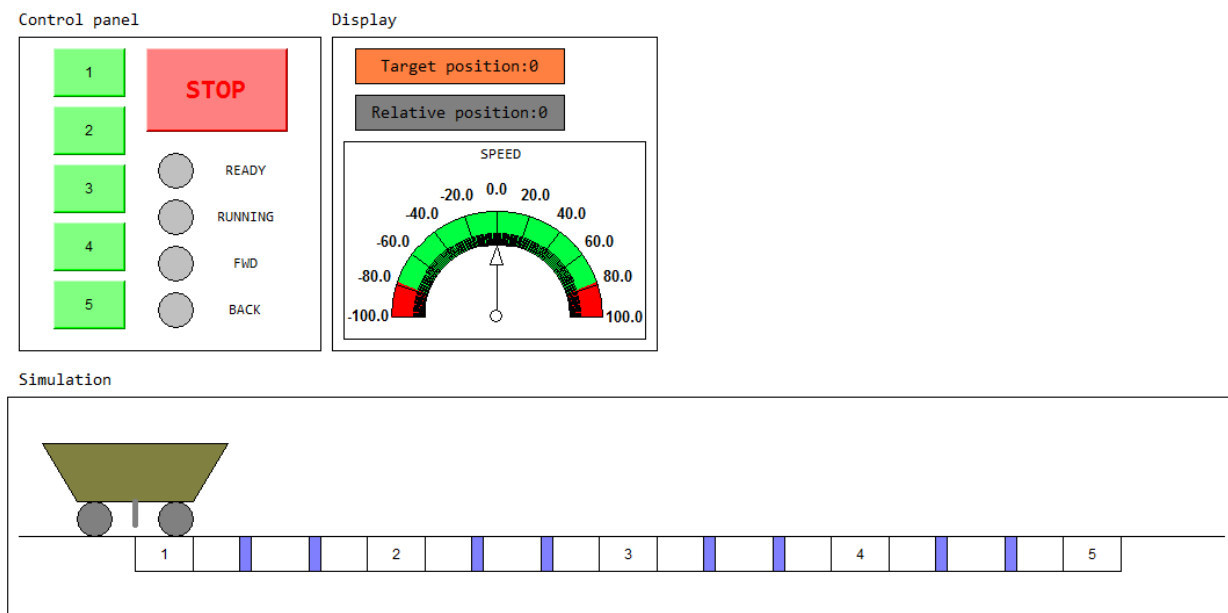
*Slika 4.22. Način dodavanja nove vizualizacije*



*Slika 4.23. Izbornik za odabir oblika i prazno mjesto za crtanje vizualizacije*



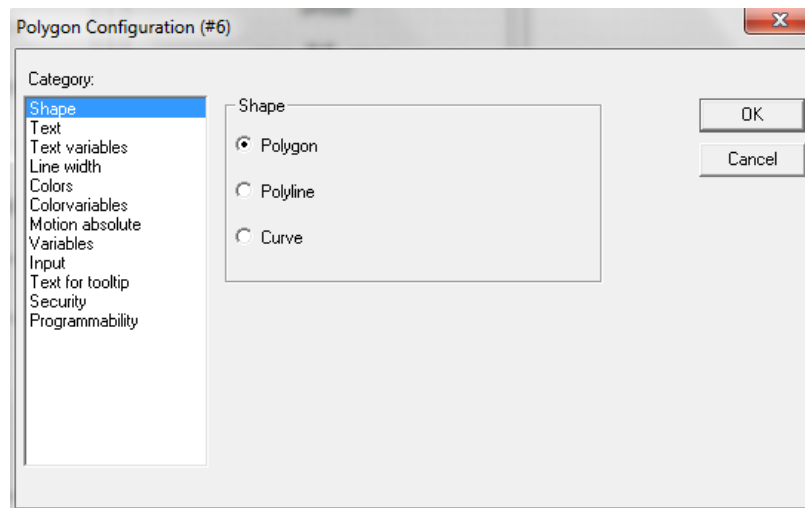
Kao što je već ranije navedeno, napravljen je prototip za izgled vizualizacije koji je na kraju i korišten za vizualizaciju. Dodavanjem jednostavnih oblika, tipki i teksta stvorena je jedinstvena vizualizacija po uzoru na prototip . Ova vizualizacija može se vidjeti na slici 4.24.



Slika 4.24. Vizualizacija korištena u završnoj verziji rješenja

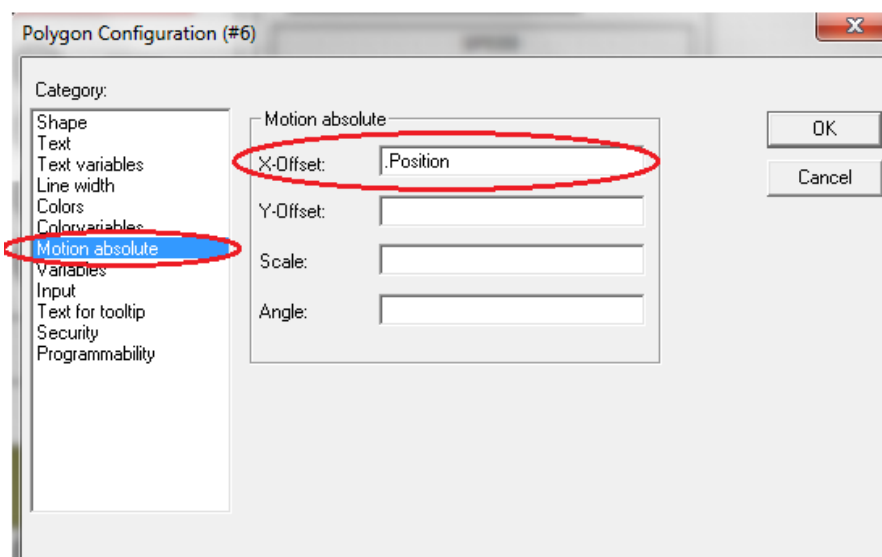
Nakon što je vizualizacija nacrtana, potrebno je dodati funkcionalnosti za pojedine objekte unutar simulacije. Teretna kolica se moraju kretati, induktivni davači moraju reagirati na prelazak teretnih kolica, na prikazu parametara se moraju prikazivati zadani parametri, a na kontrolnoj ploči mora se omogućiti funkcionalnost tipkala i signalnih lampica. Funkcionalnost objektima može se mijenjati tako da se dva puta mišem klikne na određeni objekt i na taj način pozove izbornik koji se može vidjeti na slici 4.25. Na slici 4.25. može se vidjeti kako se mogu mijenjati različiti parametri određenom objektu. Može se mijenjati oblik, tekst, boja, promjena boje na određeni događaj, apsolutna pozicija, relativna pozicija itd. Većina se ovih parametara koristilo za izradu ove vizualizacije.

Sada kada je prikazano na koji način nacrtani objekti mogu postati funkcionalni, ostaje objasniti kako je to učinjeno za svaki objekt na vizualizaciji. O tome se može pročitati u nastavku.



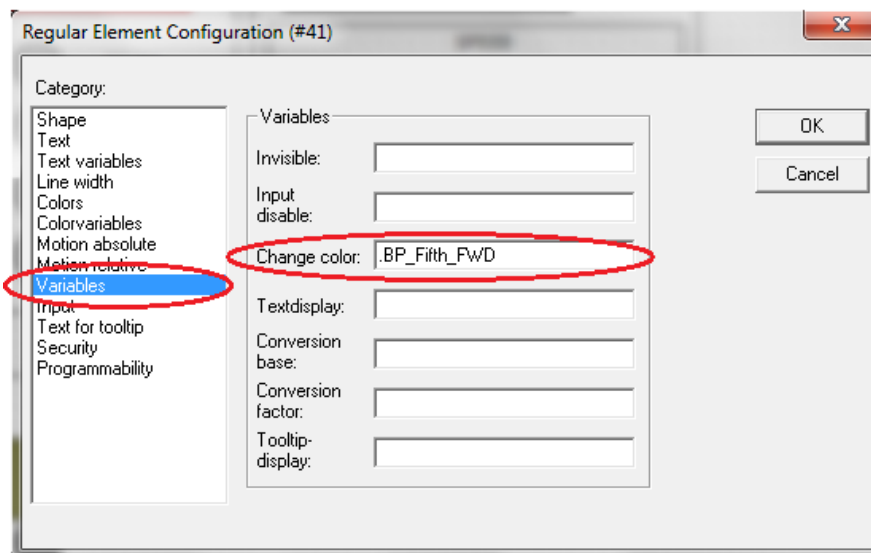
Slika 4.25. Izbornik za funkcionalnost pojedinog objekta

Teretna kolica se na vizualizaciji moraju kretati. Upravo zato je potrebno podesiti parametar pod nazivom *Motion absolute*. Podešavanje ovog parametra može se vidjeti na slici 4.26. Na slici se može vidjeti kako je u izborniku odabran *Motion absolute*. Također se može vidjeti kako se može izabrati kretanje po x ili y osi. U ovoj vizualizaciji je potrebno kretanje po x osi. Pozicija kolica na x osi mijenjati će se prema varijabli *Position* koja predstavlja apsolutnu poziciju kolica. Na ovaj način postavljena je funkcionalnost za objekt kolica.



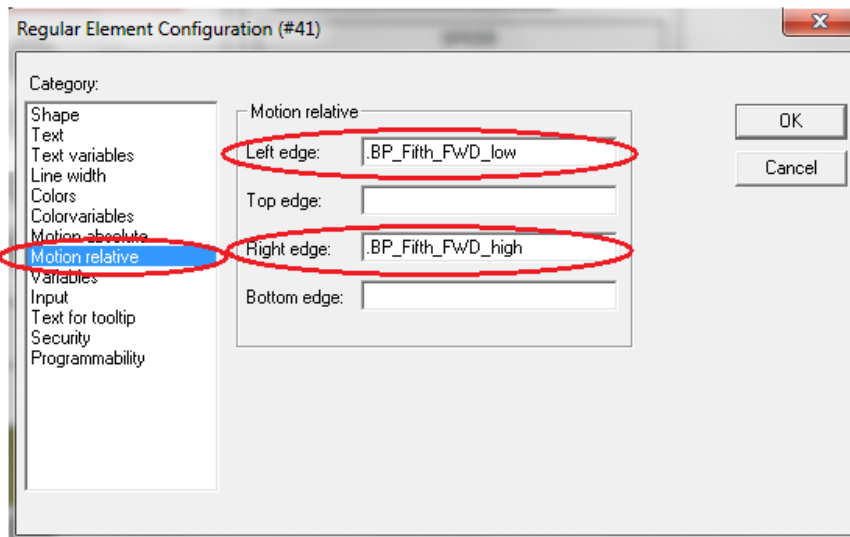
Slika 4.26. Postavljanje parametra kretanja za teretna kolica

Digitalnim davačima pozicije mora se promijeniti boja na prelazak kolica (reagiraju na prelazak kolica). Postavljanje parametra promjene boje može se vidjeti na slici 4.27. Na slici se može vidjeti kako će objekt koji predstavlja digitalni davač kočenja za petu poziciju promijeniti boju ukoliko je varijabla tog digitalnog davača u logičkoj jedinici.



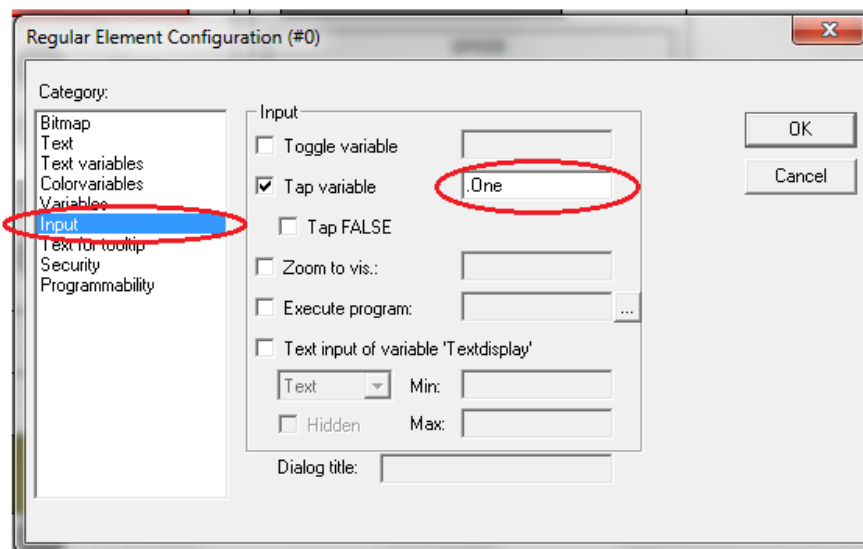
Slika 4.27. Postavljanje parametra promjene boje za induktivne davače

Digitalnim davačima pozicije također se mora namjestiti i pozicija gdje se nalaze duž puta kolica. Ovo se ostvaruje pomoću parametra *Motion relative* i može se vidjeti na slici 4.28. Na slici može se vidjeti kako je potrebno postaviti lijevi i desni rub objekta koji predstavlja induktivni davač kočenja za petu poziciju. Lijevi rub objekta poprimiti će vrijednost lijeve točke davča, dok će desni rub poprimiti vrijednost desne točke. Na ovaj način pozicioniran je digitalni davač duž puta kolica. Analogno ovom primjeru napravljeno je za svaki od preostalih dvanaest digitalnih davača pozicije. Sada, kada su digitalni davači stavljeni u funkciju, potrebno je osposobiti kontrolnu ploču i prikaz parametara. O ovome se može pročitati u nastavku.



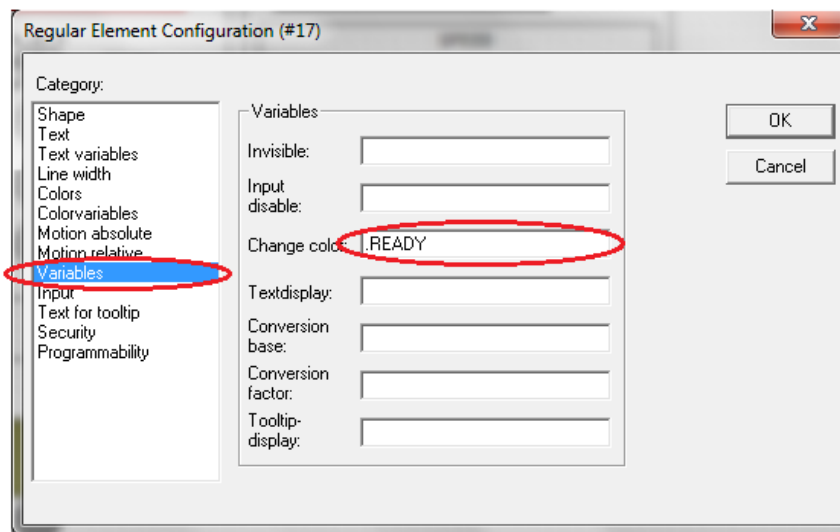
Slika 4.28. Postavljanje pozicije digitalnih davača

Na kontrolnoj ploči potrebno je dati funkcionalnost tipkama i sigurnosnim lampicama. Postavljanje parametara za tipkala može se vidjeti na slici 4.29. Na slici se može vidjeti kako je potrebno odabrati parametar *Input* i navesti varijablu kojoj će se stanje promijeniti pritiskom na tipku. U ovom slučaju prikazano je postavljanje parametara za tipku prve pozicije (*One*). Analogno ovome npravljeno je za sve ostale tipke.



Slika 4.29. Postavljanje funkcionalnosti za tipkala na kontrolno ploči

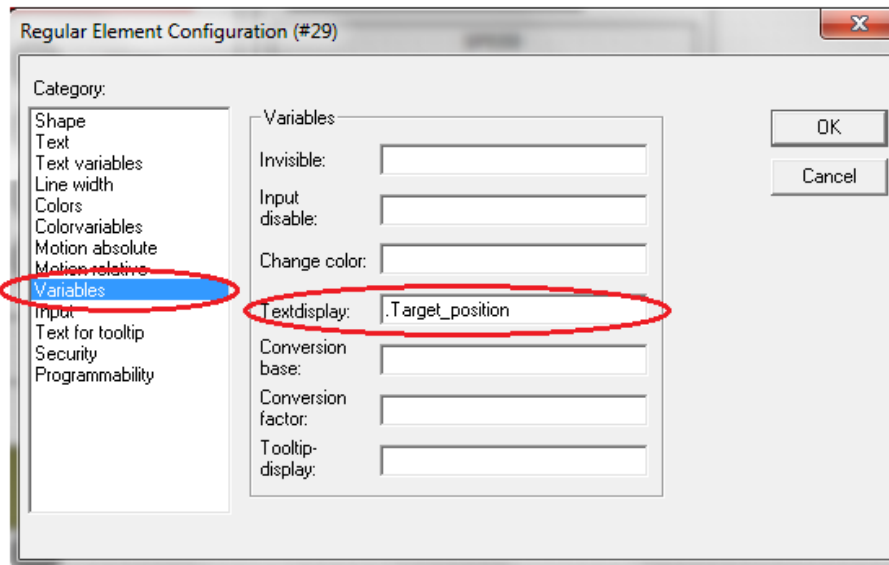
Sigurnosne lampice na kontrolnoj ploči moraju se upaliti (promijeni im se boja) ukoliko kolica idu naprijed ili nazad i stoje ili se kreću. Na slici 4.30. može se vidjeti postavljanje parametara za sigurnosnu lampicu *Ready*. Na slici se može vidjeti kako je potrebno odabrati parametar *Variables* i pod promjenu boje navesti varijablu *READY*. Sada će lampica *Ready* biti upaljena kada je motor u stanju *READY*. Analogno ovome napravljeno je za ostale lampice na kontrolnoj ploči.



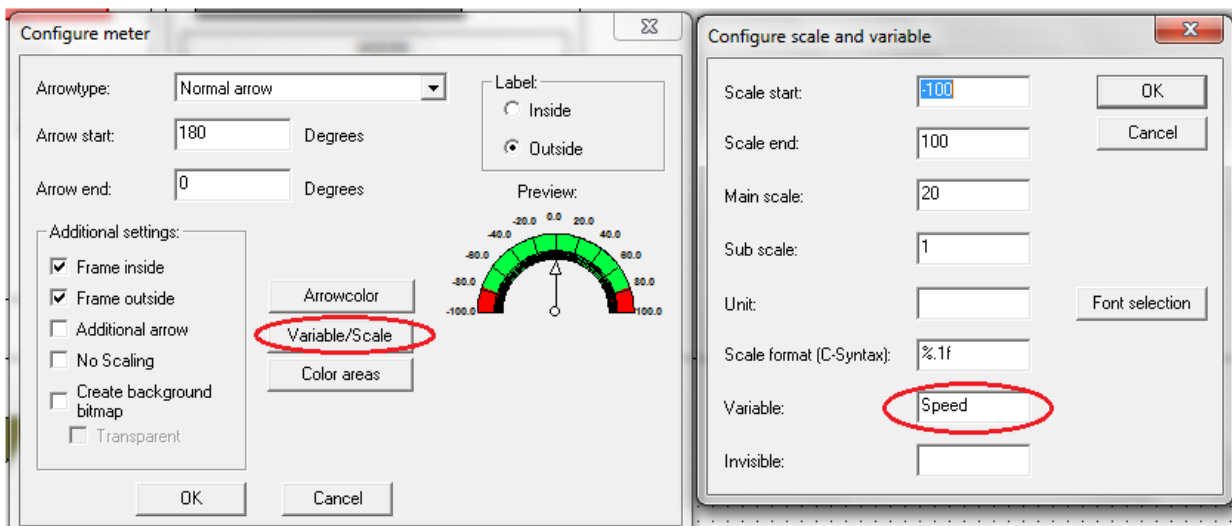
Slika 4.30. Postavljanje parametara za sigurnosnu lampicu *Ready*

Na dijelu vizualizacije za prikaz parametara može se vidjeti dva različita načina prikaza parametara. Relativna i odredišna pozicija prikazane su pomoću teksta dok je brzina prikazana pomoću brzinomjera. Postavljanje tekstualnog prikaza može se vidjeti na slici 4.31. Na slici se može vidjeti kako je potrebno odabrati parametar *Variables* i pod *Textdisplay* navesti koja varijabla se želi prikazivati. Analogno ovome napravljeno je i za ostale tekstualne prikaze.

Brzinomjer se postavlja na malo drugačiji način. To se može vidjeti na slici 4.32. Na slici se može vidjeti na koji način se postavlja brzinomjer. Potrebno je odabrati *Variable/Scale* i podesiti izgled skale i varijablu koja će se pratiti. U ovom slučaju skala ide od -100 do 100 u koracima po 20. Varijabla koja je postavljena da se prati je brzina (*Speed*).



Slika 4.31. Postavljanje tekstualnog prikaza



Slika 4.32. Postavljanje brzinomjera

Na ovaj način postavljene su sve funkcionalnosti vizualizacije. Sada je potrebno testirati programsko rješenje simulacije i vizualizacije sustava upravljanja pozicijom. Potrebno je prikazati kako se sustav ponaša za zadano rješenje i postoji li koji problem na koji treba obratiti pozornost. O analizi programskog rješenja može se pročitati u nastavku.

## 5. ANALIZA PROGRAMSKOG RJEŠENJA

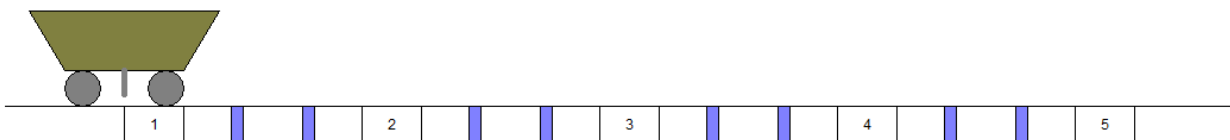
U ovom poglavlju govoriti će se o analizi programskog rješenja objašnjenog u prethodnom poglavlju. Prva stvar koja je bila potrebna kako bi rješenje bilo funkcionalno je podešavanje parametara. Nekoliko parametara je bilo potrebno podesiti i to je učinjeno eksperimentalnom metodom. Parametri koji su se trebali podesiti su akceleracija, pozicija induktivnih davača i koeficijent K. Kada se odabirala pozicija induktivnih davača pazilo se da se ti induktivni davači rasporede po praznoj stranici vizualizacije na taj način da je vizualizacija pregledna. Na slici 5.1. mogu se vidjeti parametri induktivnih davača postavljeni eksperimentalnom metodom.

0010	BP_Fifth_FWD_high		REAL	760
0011	BP_Fifth_FWD_low		REAL	750
0012	BP_Fourth_BACK_high		REAL	700
0013	BP_Fourth_BACK_low		REAL	690
0014	BP_Fourth_FWD_high		REAL	560
0015	BP_Fourth_FWD_low		REAL	550
0016	BP_Third_BACK_high		REAL	500
0017	BP_Third_BACK_low		REAL	490
0018	BP_Third_FWD_high		REAL	360
0019	BP_Third_FWD_low		REAL	350
0020	BP_Second_BACK_high		REAL	300
0021	BP_Second_BACK_low		REAL	290
0022	BP_Second_FWD_high		REAL	160
0023	BP_Second_FWD_low		REAL	150
0024	BP_First_BACK_high		REAL	100
0025	BP_First_BACK_low		REAL	90

0033	Fifth_point_high		REAL	850
0034	Fifth_point_low		REAL	800
0035	Fourth_point_low		REAL	600
0036	Fourth_point_high		REAL	650
0037	Third_point_high		REAL	450
0038	Third_point_low		REAL	400
0039	First_point_high		REAL	50
0040	First_point_low		REAL	0
0041	Second_point_high		REAL	250
0042	Second_point_low		REAL	200

Slika 5.1. Parametri induktivnih davača

Na slici 5.1. vidimo vrijednost za lijevu i desnu točku svakog od 5 glavnih induktivnih davača pozicije i 8 pomoćnih induktivnih davača pozicije. Krajnji induktivni davač pozicije (*Fifth\_point*) je smješten na poziciji od 800 jedinica dok je prvi (*First\_point*) smješten na poziciji 0. Svaki od 5 glavnih induktivnih davača dugačak je 50 jedinica, dok su pomoćni induktivni davači dužine 10 jedinica. Induktivni davači za kočenje odmaknuti su 50 jedinica od induktivnih davača pozicije. Na ovaj način osiguran je dobar raspon i pregled induktivnih davača koji se može vidjeti na slici 5.2. Na slici 5.2. se može vidjeti kako su glavni induktivni davači označeni s brojevima od 1-5, dok su induktivni davači za kočenje obojani plavom bojom. Također, može se vidjeti kako su kolica pozicionirana na poziciji 0 kao i lijeva strana prvog pokazivača. To je zbog toga što je početna pozicija kolica postavljena na 0.



Slika 5.2. Raspored induktivnih davača na vizualizaciji

Sada, kad se zna maksimalna pozicija do koje teretna kolica mogu ići i odnos pozicija induktivnih davača, potrebno je postaviti akceleraciju i koeficijent  $K$ . Postavljanje ova dva koeficijenta odrađeno je eksperimentalno. Što je koeficijent  $K$  veći to će kolica brže mjenjati poziciju a što je koeficijent  $K$  manji to će kolica ići sporije. Optimalan koeficijent  $K$  određen je nizom pokusa te je odabrana vrijednost koja osigurava dobar pregled putovanja kolica (da kolica baš ne „lete“ ili se ne „vuku“ na vizualizaciji). Na sličan način određena je vrijednost akceleracije. Kako bi se na vizualizaciji što bolje primijetilo usporavanje i ubrzavanje te razlika između više i niže brzine (brzine približavanja), potrebno je odrediti dobru vrijednost akceleracije. Ovo je također ostvareno eksperimentalnom metodom.

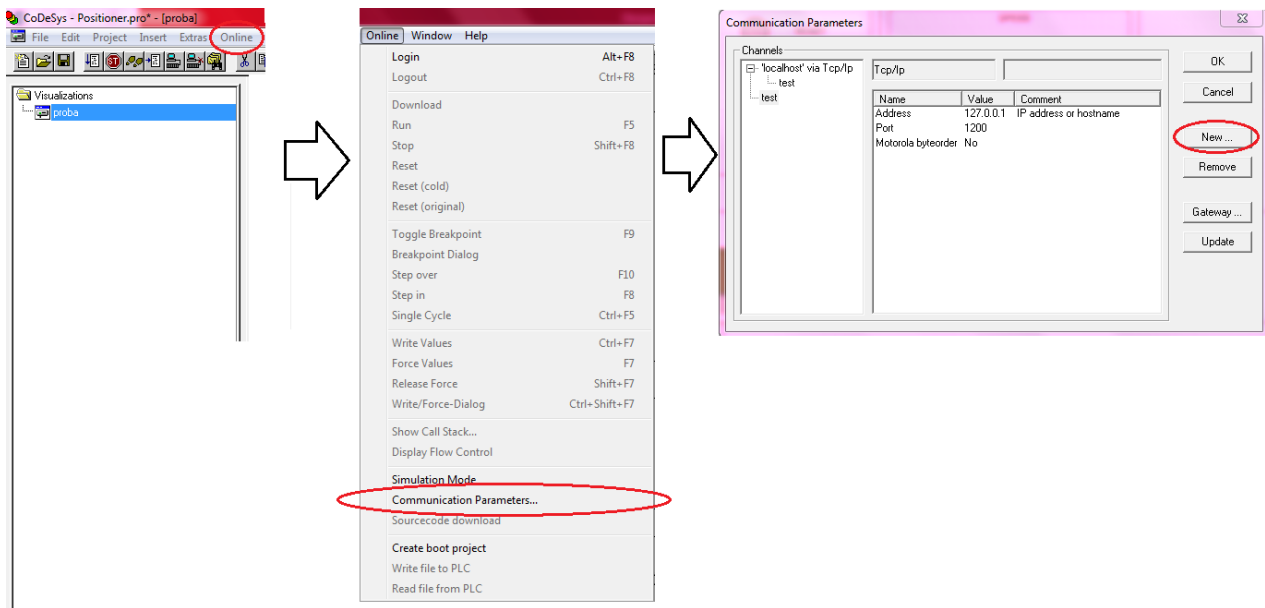
Važno je napomenuti kako je u slučaju stvarnog procesa upravljanja u industriji, ukoliko se radi vizualizacija takvog procesa, potrebno stvoriti vizualizaciju u određenom omjeru prema stvarnom procesu kako bi tehničar mogao pratiti u svakom trenutku stanje sustava i poziciju teretnih kolica. U ovom diplomskom radu ne postoji stvarni sustav već je sve simulacija. Upravo zbog toga su parametri kao što su akceleracija, koeficijent  $K$  i udaljenost među induktivnim davačima izabrani za što bolji prikaz i rad u svrhu što bolje prezentacije programskog rješenja.

Sada kada su se definirali parametri za dobar rad rješenja potrebno je pokrenuti simulaciju i vizualizaciju kako bi se prikazao rad. O ovme se može pročitati u nastavku.

## 5.1. Pokretanje vizualizacije i simulacije

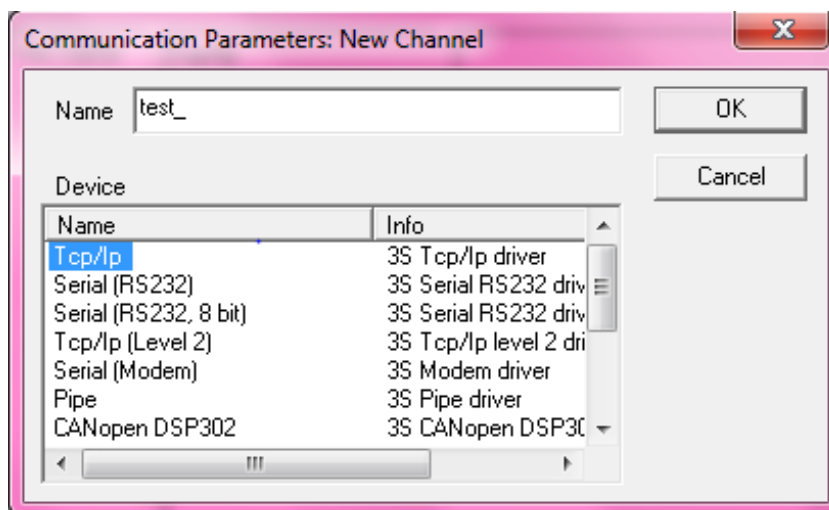
Pokretanje simulacije i vizualizacije vrši se preko softwera pod nazivom *CoDeSys SP PLCWinNt*. Ovaj software predstavlja *CoDeSys gateway* server. Prije no što se sama vizualizacija i simulacija pokrenu potrebno je podesiti nekoliko parametara. Kada se projekt završi potrebno je slijediti korake kao na slici 5.3.





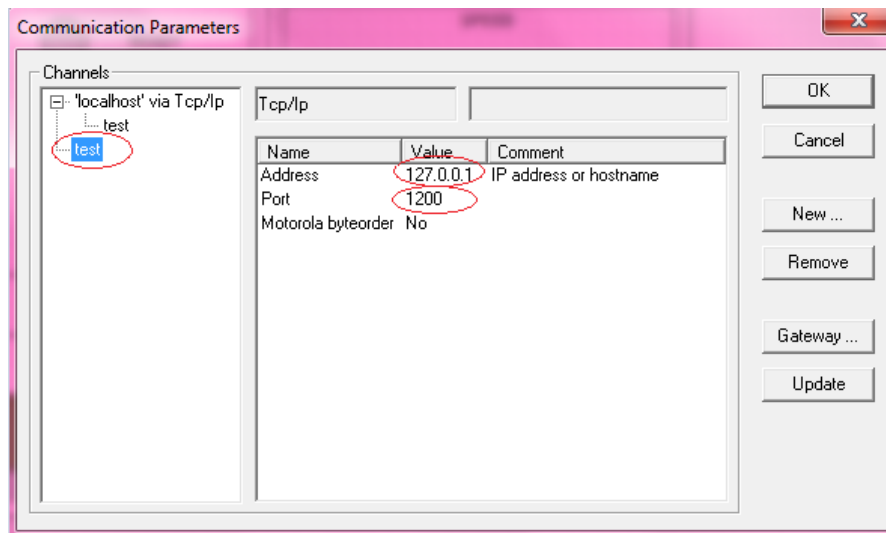
Slika 5.3. Instrukcije za postavljanje parametara

Kada se izabere dodavanje novog kanala potrebno je odabrati *Device* i ime. To se može vidjeti na slici 5.4.



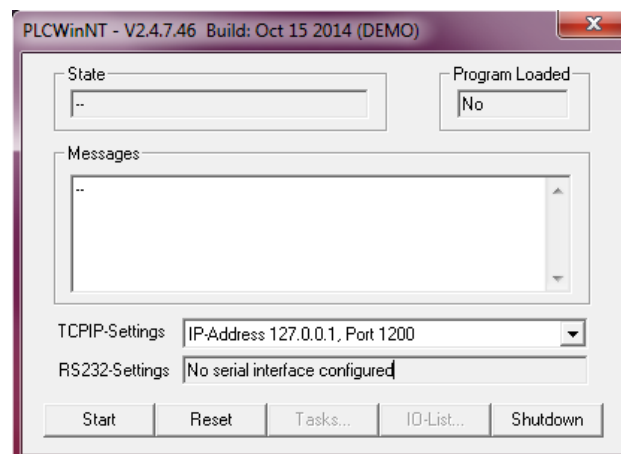
Slika 5.4. Odabiranje komunikacijskih parametara

Kada se odabere ime i *Device*, novi kanal pojavi se među kanalima. Sada je potrebno postaviti adresu i port. Ovo se može vidjeti na slici 5.4.



Slika 5.4. Postavljanje adrese i porta za određeni kanal

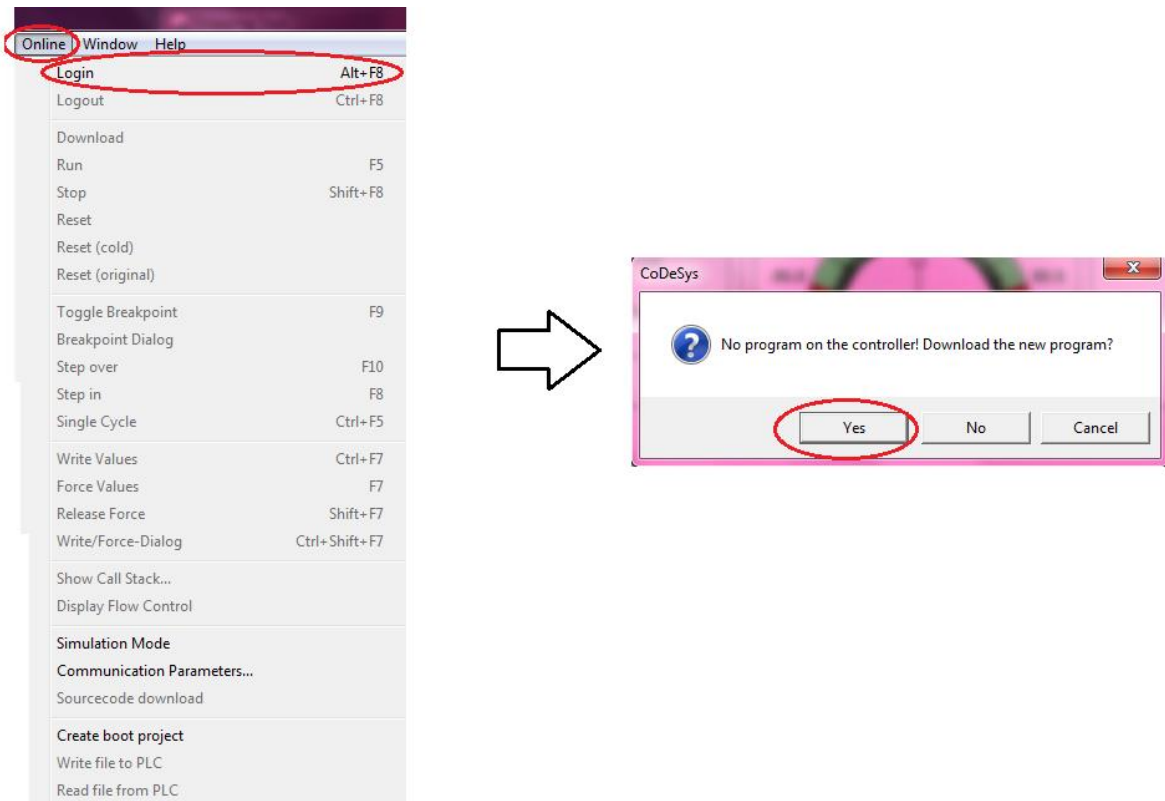
Na slici 5.4. može se vidjeti kako je postavljena adresa *local host* (može se upisati ili IP adresa ili ime), te port 1200. Sada kada je stvoren novi kanal s određenim parametrima, potrebno je pokrenuti *CoDeSys SP PLCWinNt*. Kada se program pokrene prikaže se izbornik koji se može vidjeti na slici 5.5.



Slika 5.5. izbornik *CoDeSys SP PLCWinNt*

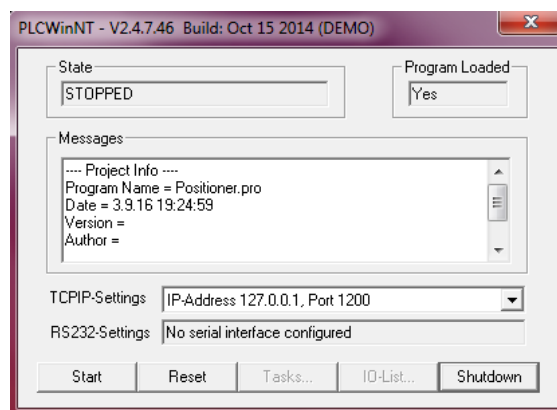
Na slici 5.5. mogu se vidjeti podešeni parametri za TCP/IP koji su isti zadanim u predhodnom koraku. Također može se vidjeti kako nema učitano programa. Sada, kada se pokrenuo *gateway*

server i kada su se namjestili svi parametri, potrebno je pokrenuti simulaciju i vizualizaciju. To se može vidjeti na slici 5.6.



*Slika 5.6. Pokretanje simulacije i vizualizacije*

Sada se na izborniku *CoDeSys SP PLCWinNt-a* može vidjeti učitani program (Slika 5.7.).

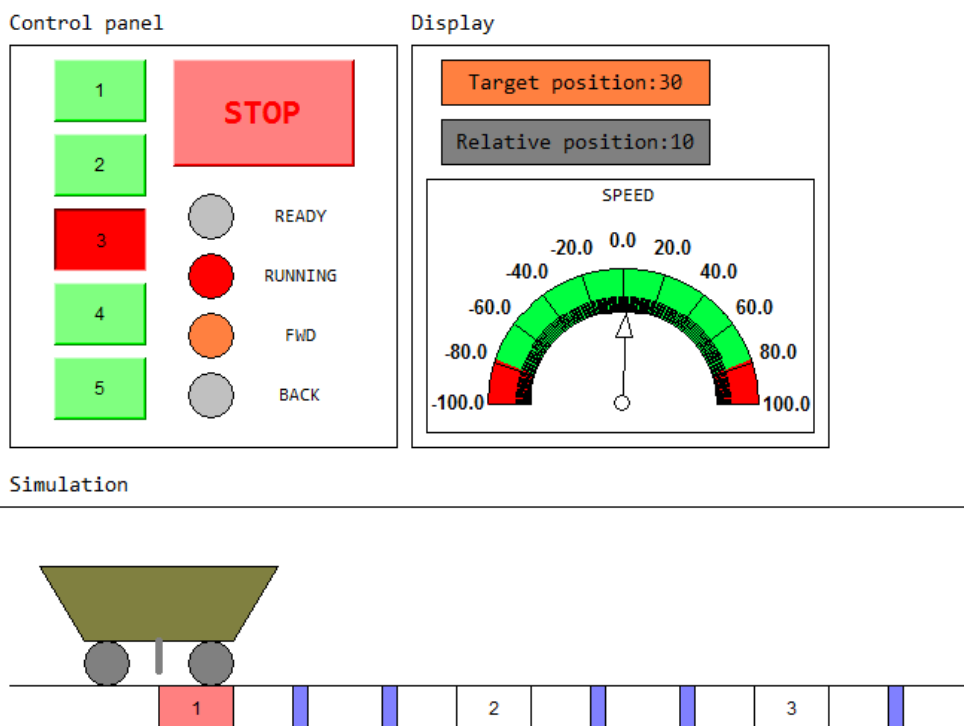


*Slika 5.7. Učitani je program*

Sada, kad su simulacija i vizualizacija pokrenute, može se testirati programsko rješenje. O tome se može pročitati u nastavku.

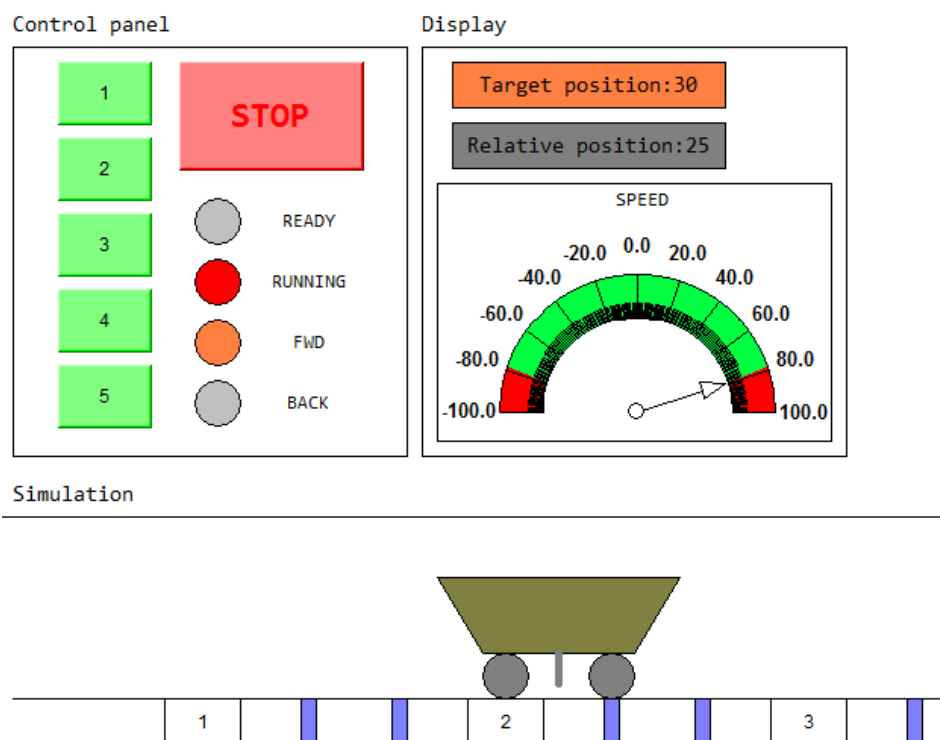
## 5.2. Testiranje programskog rješenja

U ranijim poglavljima opisano je kako izgleda vizualizacija i kako je ostvarna simulacija za ovaj proces. Sada je ostalo to sve testirati. Za testiranje ispravnog rada procesa može se odabrati nekoliko slučajeva. Prvi slučaj je slučaj u kome se odabere jedna od pozicija te kolica tamo trebaju doći. Kada se stisne jedna od tipaka s kontrolnog panela potrebno je pratiti rade li ispravno kontrolne lampice *READY*, *RUNNING*, *FWD* i *BACK*. Također, potrebno je pratiti *Display* i mijenjajunli se pravilno *Target position*, *Relative position* i *SPEED*. Na kraju je potrebno pratiti izvršavaju li traženu radnju. Prvi test napravljen je za slučaj kada se pritisne tipka za treću poziciju. Ovo se može vidjeti na slici 5.8.



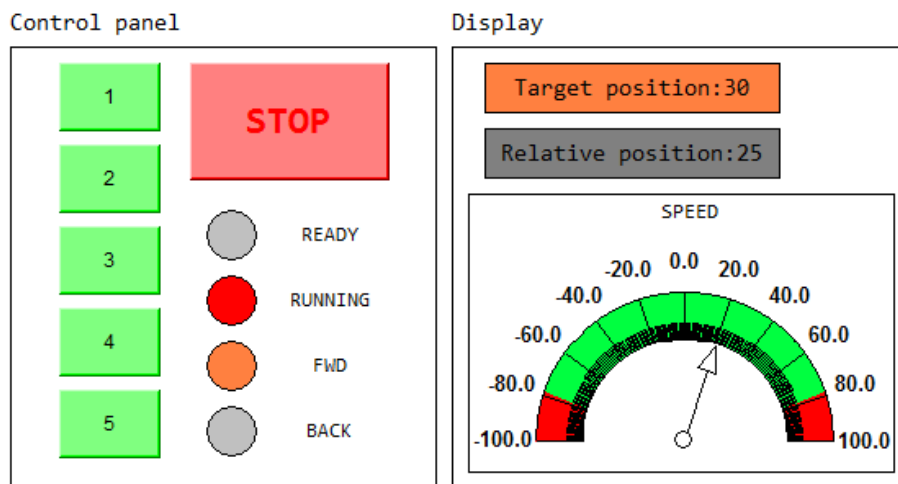
Slika 5.8. Pritisnuta je pozicija za treći induktivni davač

Odmah na početku, na slici 5.8. se može uočiti kako je stisnuta tipka za treću poziciju. Također se može vidjeti kako su upaljene kontrolne lampice *RUNNING* i *FWD* koje označavaju da se kolica kreću i da idu naprijed. Na *Display*-u se može vidjeti kako je *Target position* postavljen na 30, a *Relative position* na 10 (što je dobro jer se kolica nalaze iznad prvog digitalnog davača pozicije). Kako se kolica pomiču tako je i brzina narasla na 80, a relativna pozicija se mijenja ovisno o tome gdje se kolica nalaze. Na slici 5.9. se može vidjeti kako je brzina sada jednaka 80, a relativna pozicija jednaka je 25 zato što se kolica nalaze između drugog i trećeg digitalnog davača pozicije.

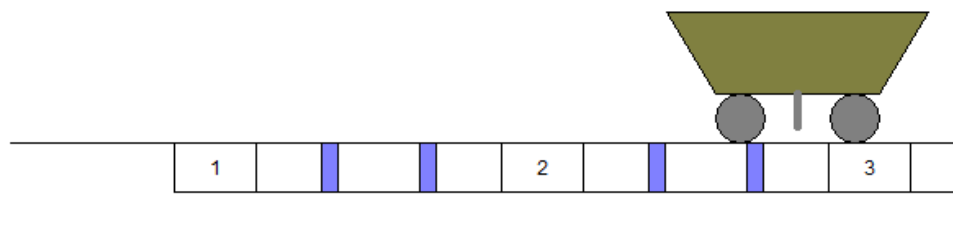


Slika 5.9. Kolica se kreću brzinom 80

Nakon što kolica pređu preko digitalnog davača za kočenje, koji se nalazi lijevo od trećeg digitalnog davača za poziciju, brzina se treba smanjiti na 20. To se može vidjeti na slici 5.10.

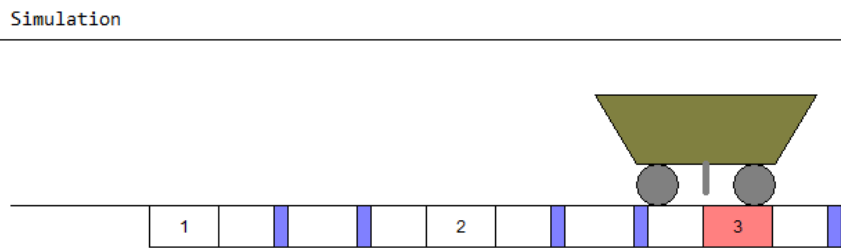
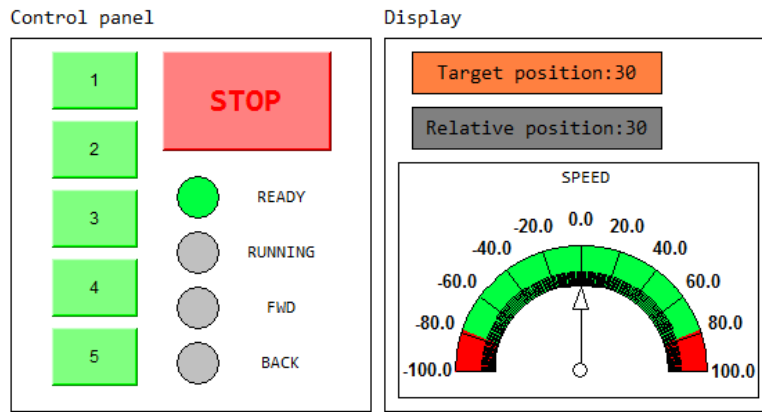


Simulation



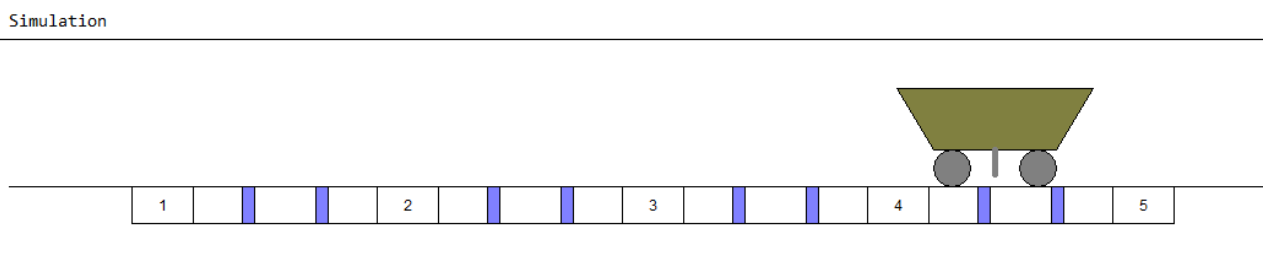
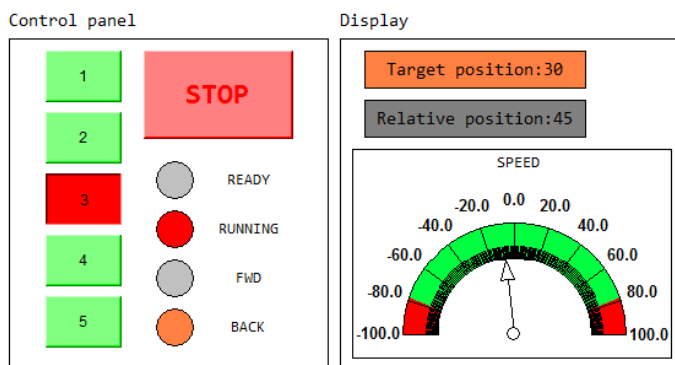
Slika 5.10. Kolica usporavaju prije treće pozicije

Nakon što kolica dođu na digitalni davač treće pozicije ona moraju stati, kontrolna lampica za *READY* se treba upaliti, a kontrolne lampice za *FWD* i *RUNNING* ugасiti. To se može vidjeti na slici 5.11. Sada se može zaključiti kako za ovaj primjer simulacija i vizualizacija rade ispravno. Nakon ovog testiranja, test je pokrenut i za sve ostale pozicije u svim mogućim kombinacijama i smjerovima. Simulacija i vizualizacija su se pokazale ispravnima za sve zadane primjere. Može se zaključiti kako su simulacija i vizualizacija dobro projektirane. No, ostaje ispitati još jedan slučaj. To je slučaj kada se kolica zaustave na tipku *STOP*. Kod ovog primjera simulacija i vizualizacija moraju proći test za situaciju kada se kolica nalaze zaustavljena negdje između bilo koja dva digitalna davača pozicije. O ovome se može pročitati u nastavku.



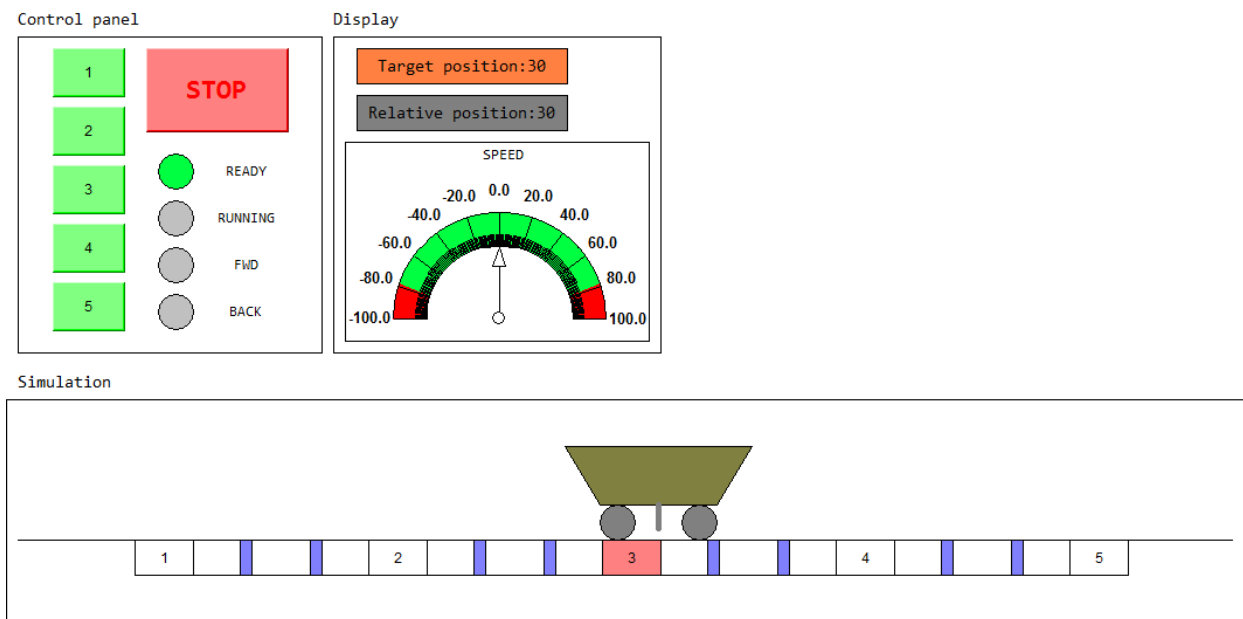
Slika 5.11. Kolica su se pozicionirala na željenu poziciju

Za testiranje *STOP* opcije prikazati će se primjer gdje će se kolica zaustaviti između četvrte i pete pozicije te će se pozvati da se vrate na treću poziciju. Ovo se može vidjeti na slici 5.12.



Slika 5.12. Kolica su zaustavljena između četvrte i pete te su pozvana na treću poziciju

Na slici 5.12. se može vidjeti da je pritisnut gumb za treću poziciju. Upalile su se lampice za *RUNNING* i *BACK*, te kolica ubrzavaju unazad prema trećoj poziciji. *Target position* je postavljen u 30 (za treću poziciju), a vrijednost *Relative position* jednaka je 45 što označava da su kolica između četvrte i pete pozicije. Kako se kolica kreću tako ubrzavaju na brzinu -80 i mijenja im se relativna pozicija. Kolica se na kraju trebaju pozicionirati na treću poziciju. To se može vidjeti na slici 5.13.



Slika 5.13. Kolica su se pozicionirala na treću poziciju

Nakon ovog slučaja, program je testiran na velikom broj slučajeva u kojima su kolica zaustavljena između dvije pozicije te zatim pozvana na neku drugu poziciju. Simulacija i vizualizacija su prošle test za svaki od tih slučajeva.

Nakon testiranja programskog rješenja, može se zaključiti kako je programsko rješenje simulacije i vizualizacije za proces promjene pozicije dobro osmišljeno te na kraju dobro i izvedeno. Ovo programsko rješenje predstavlja jedan od načina kako ostvariti upravljanje pozicijom u otvorenoj petlji pomoću digitalnih davača pozicije i digitalnih i analognih aktuatora.



## 6. ZAKLJUČAK

Može se reći kako je upravljanje procesima vrlo bitna znanstvena disciplina za doba u kojem živimo, jer današnje doba s pravom možemo nazvati doba automatizacije. U ovom radu detaljno je prikazan i opisan jedan sustav upravljanja pozicijom u industriji. Sustav upravljanja pozicijom ima veliku primjenu u industriji i u svakom postrojenju može se naći barem neki oblik ovakvog upravljanja. Neka upravljanja izvedena su s povratnom vezom jer takavi sustavi zahtjevaju veliku preciznost i osjetljivi su na smetnje koje regulator mora kompenzirati. S druge strane postoje procesi koji ne zahtjevaju veliku preciznost i skupe načine regulacije već je upravljanje bez povratne veze sasvim dovoljan alat za automatizaciju ovakvih sustava. Jedan od takvih sustava je i sustav upravljanja pozicijom opisan u ovom diplomskom radu.

Kako bi se moglo projektirati dobro rješenje vizualizacije i simulacije sustava upravljanja pozicijom potrebna je dobra teorijska podloga o tome kako taj sustav zapravo izgleda i koje su njegove funkcijske cjeline. Upravo zato su ovim radom detaljno opisana sučelja, aktuatori, regulatori i ostale komponente koje čine jedan ovakav sustav u industriji. Sustav koji je opisan u teorijskom djelu rada je sustav transporta metalnih namotaja na skladište. Iako se čini vrlo jednostavan, ovaj sustav je neophodan za funkcioniranje cijelog postrojenja.

Na kraju diplomskog rada opisan je postupak projektiranja sustava upravljanja pozicijom. Ovo je odrađeno u programskom paketu Codesys. Pri izradi ovog rješenja moralo se paziti na mnoštvo stvari i parametara kako bi krajnji rezultat što više nalikovao na stvarni sustav upravljanja pozicijom. Nakon izrade vizualizacije i simulacije sustava napravljena je analiza istih i može se utvrditi kako je programsko rješenje zadovoljavajuće.

## LITERATURA

- [1] Prof.dr.sc Nedeljko Perić, AUTOMATSKO UPRAVLJANJE predavanja, Zagreb, 2004.
- [2] [https://en.wikipedia.org/wiki/Low-level\\_programming\\_language](https://en.wikipedia.org/wiki/Low-level_programming_language)
- [3] FER, Zavod za automatiku i procesno računarstvo, Elementi automatizacije procesa - predavanja  
[https://www.fer.unizg.hr/download/repository/EAP\\_PLC\\_dio.pdf](https://www.fer.unizg.hr/download/repository/EAP_PLC_dio.pdf)
- [4] <https://en.wikipedia.org/wiki/EPROM>
- [5] Thomas A. Kinney, Proximity sensors compared article, 2001.  
<http://machinedesign.com/sensors/proximity-sensors-compared-inductive-capacitive-photoelectric-and-ultrasonic>
- [6] SIMENS, AS-i introduction and basics manual  
[https://w3.siemens.com/mcms/industrial-communication/en/support/ik-info/Documents/SYH\\_asi\\_grundlagen-76.pdf](https://w3.siemens.com/mcms/industrial-communication/en/support/ik-info/Documents/SYH_asi_grundlagen-76.pdf)
- [7] <https://en.wikipedia.org/wiki/AS-Interface>
- [8] [https://en.wikipedia.org/wiki/Induction\\_motor](https://en.wikipedia.org/wiki/Induction_motor)
- [9] Končar - Tehnički priručnik, peto izdanje, Zagreb, 1991.
- [10] Ž. Špoljarić, M. Pužar, Laboratorijske vježbe, Asinkroni motori – kratki spoj i prazni hod, ETF, Osijek, 2009.
- [11] [https://en.wikipedia.org/wiki/Variable-frequency\\_drive](https://en.wikipedia.org/wiki/Variable-frequency_drive)

## SAŽETAK

Ovaj diplomski rad opisuje jedan proces upravljanja pozicijom u industriji. U radu su dane osnovne informacije o automatskom upravljanju u otvorenoj i zatvorenoj petlji. Također, u radu je prikazan proces upravljanja pozicijom u industriji. Prikazane su osnovne hardwareske komponente koje se mogu pronaći u jednom ovakvom sustavu unutar industrijskog postrojenja, koja je uloga svake pojedine komponente te na koji način one surađuju kako bi izvršavale zadane funkcije. Opisano je programsko rješenje simulacije i vizualizacije izrađeno u programskom paketu *CoDeSys*. Vizualizacija prikazuje radna kolica, digitalne davače pozicije te kontrolne panele. Simulacija omogućuje ispitivanje razvijenog sustava upravljanja i vizualizacije. Teretna kolica mogu se pozicionirati na bilo koju poziciju korištenjem kontrolnog panela. Na samom kraju, testiranjem simulacije i vizualizacije, utvrđena je ispravnost programskog rješenja.

**Ključne riječi:** CoDeSys, upravljanje u otvorenoj petlji, digitalni davači pozicije, proces, automatska regulacija

## **ABSTRACT**

### **Position control in open loop system using digital proximity sensors and digital or analog actuators**

This thesis describes an industrial process of position control. Basic information about open and close loop control are provided. An example of position control in an industrial plant is described. Basic hardware components, which are commonly used in such a system are presented and their role in the system is described. Furthermore, it is explained how they are combined in order to solve a given task. A software solution for visualisation and simulation created in CoDeSys programming environment is presented. Visualisation includes industrial cart, inductive sensors and control panels. Cart can be positioned at any position using a control panel. Finally, simulation and visualization were tested and their functionality is confirmed.

**Key words:** CoDeSys, open loop control, inductive position sensors, process, automation

## ŽIVOTOPIS

Benjamin Žagar rođen je u Virovitici 24.11.1992. godine. U Orahovici je završio osnovnu školu Ivane Brlić Mažuranić s odličnim uspjehom. Srednju školu Stjepana Ivšića upisuje u Orahovici, smjer opća gimnazija, te ju završava s odličnim uspjehom. Nakon srednješkolskog obrazovanja upisuje Preddiplomski studij smjer Računarstvo na Elektrotehničkom fakultetu u Osijeku. Godine 2014. završava preddiplomski studij računarstva te stječe status prvostupnog inženjera računarstva. Iste godine upisuje sveučilišni diplomski studij smjer Procesno računarstvo na Elektrotehničkom fakultetu u Osijeku.

Benjamin Žagar

---

