

AVTOMATIZACIJA INDUSTRIJSKEGA PROCESA IN UPORABA SIMULATORJA NAMESTO STROJA

Automatisation of industrial processes and use of the simulator instead of the machine

Andrej Arh, dipl. inž. el.
Šolski center Kranj
andrej.arh.tsckr@gmail.com

Povzetek

V tem prispevku je predstavljena učinkovita uporaba simulatorja pri učenju programiranja industrijskih krmilnikov. Prikazane so prednosti in slabosti takega načina dela. Ker se ne potrebuje strojne opreme, lahko dijaki vadijo in se učijo tudi doma, učitelj pa lahko nazornejše prikazuje razlago preko projektorja na platno.

Ključne besede: Siemens, Simatic S7, simulacija, koračna veriga, regulacija.

Abstract

This article is introducing you the effective use of simulator for teaching of industrial controllers. It is shown the good and bad functions this kind of work. Because controller it does not need the hardware, the students can learn at home, and teacher can better show his explanation of context with projector and whiteboard.

Keywords: Siemens, Simatic S7, simulation, sequential chain, regulation.

1 Uvod

Namen tega prispevka je, da slušatelje seznanimo, da lahko s simulacijo industrijskega procesa marsikomu približamo predstavo o programiranju krmilnikov, obenem pa tudi samo razumevanje industrijskega procesa. Seveda se moramo pri tem zavedati, da simulator nikakor ne more v celoti nadomestiti fizičnega stroja glede spoznavanja tega mehatronskega sklopa. Ko spoznamo delovanje enega stroja, se lahko s simulatorjem učinkovito naučimo zahtevnejšega programiranja, ne da bi za to potrebovali paleto dragih strojev. Na ta račun lahko privarčujemo precej denarja.

2 Simulatorji

Pri računalniški simulaciji z računalniškim programom predstavimo situacijo iz resničnega sveta (https://sl.wikipedia.org/wiki/Ra%C4%8Dunalni%C5%A1ka_simulacija). V grobem je

izdelava simulatorja potekala na sledeč način: iz realnega sistema zmodeliramo matematični model, katerega nazadnje lahko sprogramiramo v simulacijski model. Sami smo aplikacijo napisali z orodjem Visual Studio Community (C#), kateri preko ActiveX kontrolnikov virtualnega krmilnika PLCSIM bere in piše vhodno-izhodno sliko krmilnika.

2.1 Ideja

Celotna ideja je bila, da lahko dijaku preko projektorja prikazujemo program in njegov učinek na stroju. Ker imamo v laboratoriju 5 različnih strojev, je zelo težko učiti osnove programiranja, če ima vsak drugačen stroj. Če pa uporabljajo simulator, imajo vsi isto učilo, zato lahko pri učenju programiranja vsi obdelujemo isti problem. Ko pa dijaki poznajo osnove, lahko vsak po svoje začnejo reševati svoj projekt.

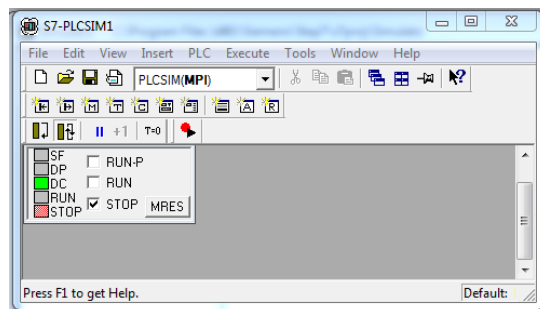
2.2 Programske zahteve

Simulator za svoje delovanje potrebuje naslednjo programsko opremo.

- Windows 7 ali novejši (XP- ne podpira .NET Framework 4.5),
- Microsoftovo ogrodje .Net Framework 4.5,
- Visual Basic Powerpack 2010,
- Simatic Manager ali TIA Portal,
- PLC Sim (Običajno se namesti s Simatic Managerjem).

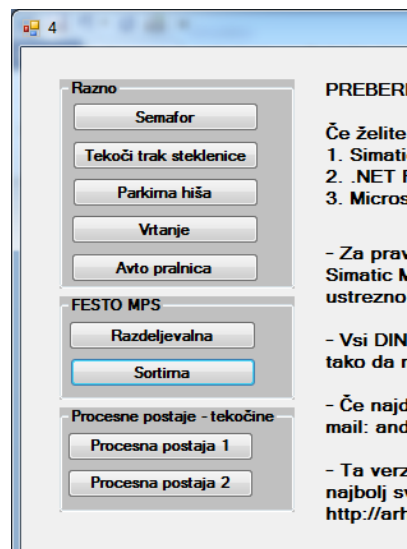
2.3 Priprava okolja za simulacijo

Ko ustvarimo projekt v okolju Simatic Manager, definiramo strojno konfiguracijo krmilnika. Preden zaženemo simulator, mora biti odprt tudi virtualni krmilnik »S7-PLCSIM1«, ker se simulator ob odpiranju simulacije poveže na ta krmilnik in prevzame kontrolo nad njim.



Slika 1: Virtualni krmilnik PLCSIM (Vir: lasten)

Ko imamo odprt virtualni krmilnik, lahko odpremo poljubno simulacijo.

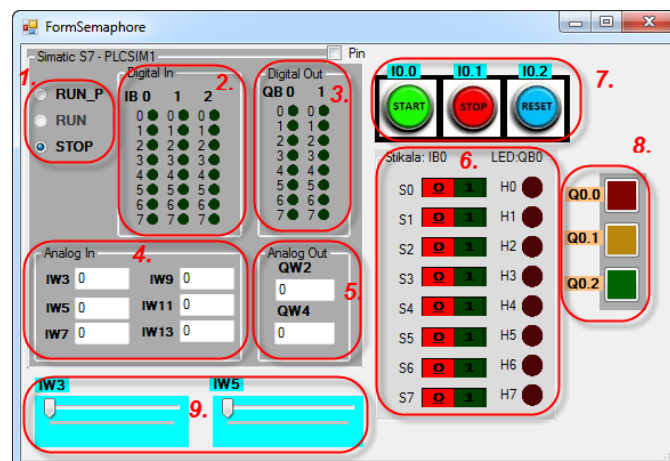


Slika 2: Izbor simulacije (Vir: lasten)

Zaenkrat so na voljo naslednje simulacije:

- semafor,
- tekoči trak s steklenicami,
- parkirna hiša,
- vrtanje obdelovanca s 3 cilindri,
- avtopralnica,
- FESTO MPS – razdeljevalna postaja,
- FESTO MPS – sortirna postaja,
- procesna postaja (reguliranje nivoja in temperature tekočine).

Če odpremo prvo simulacijo »Semafor«, ki je namenjena osnovnim logičnim funkcijam in primerjalnikom, dobimo pogled prikazan na spodnji sliki:



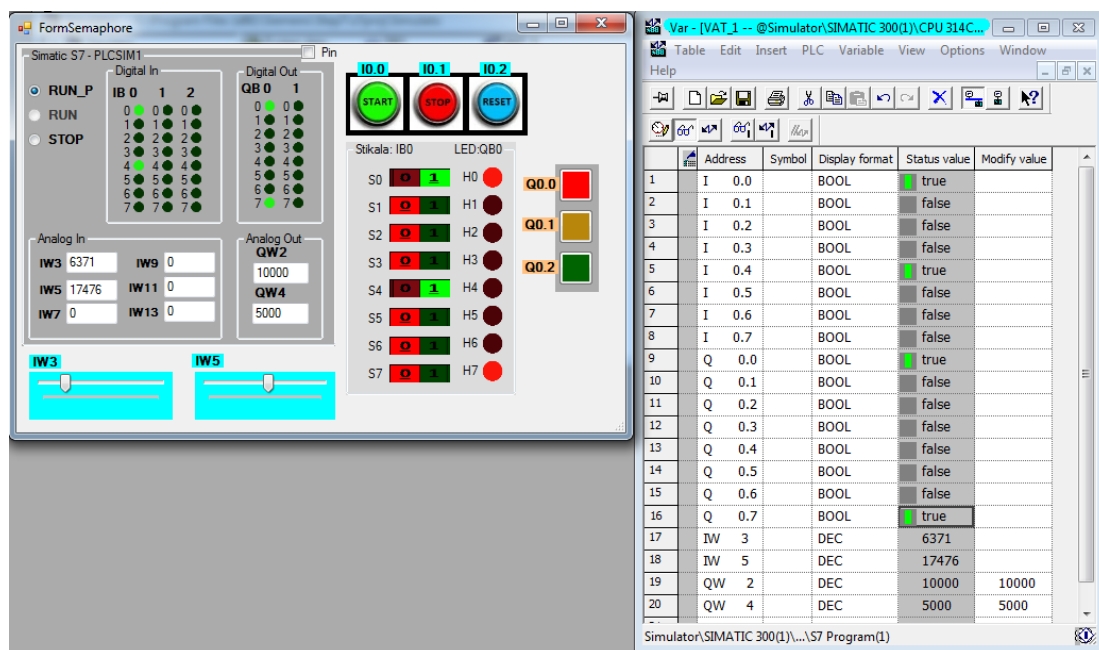
Slika 3: Sklopi osnovnega dela simulatorja (Vir: lasten)

Opis posameznih sklopov simulatorja:

1. Krmilnik zažene oz. ustavimo.
2. Prikaz stanja digitalnih vhodov krmilnika na prvih treh bajtih.
3. Prikaz stanja digitalnih izhodov krmilnika na prvih dveh bajtih.

4. Prikaz stanja na analognih vhidih krmilnika na naslovih od IW3 do IW13.
5. Prikaz stanja na analognih izhodih krmilnika na naslovih QW2 in QW4.
6. Periferni del in predstavlja 8 stikal na prvem bajtu digitalnih vhodov in 8 luči povezanih na prvi bajt digitalnih izhodov.
7. Tri tipke, ki so povezane na prve 3 bite na prvem bajtu digitalnih vhodov.
8. Semafor, ki lahko prikazuje stanje naprave.
9. Potenciometra, ki sta povezana na analogna vhoda IW3 in IW5 in sta namenjena zgolj demonstraciji.

Če virtualni krmilnik postavimo v stanje RUN_P (1. točka), lahko v VAT tabeli pregledamo stanje na vhidih in postavimo stanje na izhodih.

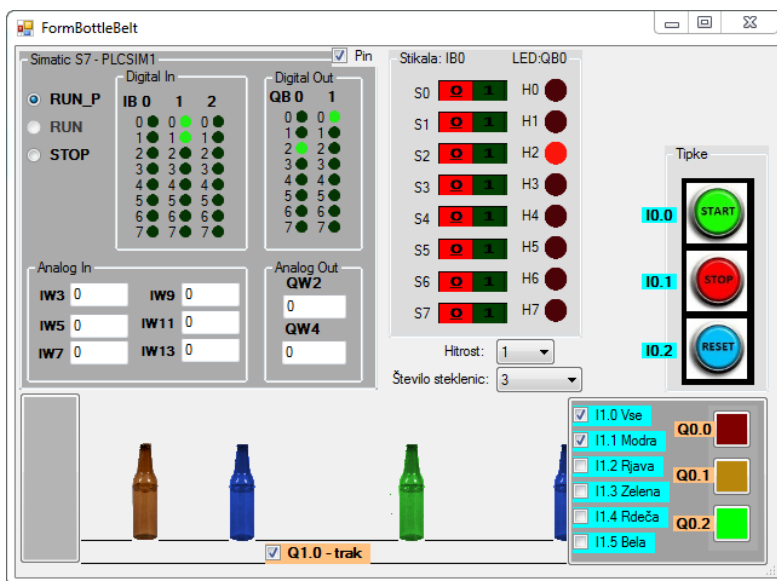


Slika 4: Prikaz povezave simulatorja in VAT tabele (Vir: lasten)

Kot primer vključimo stikala S0 in S4, v VAT tabeli opazimo spremembo stanja v vhidih. Prav tako lahko v VAT tabeli v visoko logično stanje postavimo naslova Q0.0 in Q0.7, ki prižgeta luči H0 in H7, ker sta povezani na ta naslova. Podobna zadeva je z analognimi vhodi in izhodi. Če premaknemo prvi potenciometer, ki je povezan na naslov IW3, tudi v VAT tabeli opazimo spremembo. Če so spremembe v VAT tabeli, lahko te vrednosti uporabimo v našem programu.

2.4 Tekoči trak s steklenicami

V tej simulaciji bomo konkretno predstavili vajo, ki predstavlja tekoči trak, po katerem prihajajo naključno postavljene steklenice. Cilj naloge je, da program ob pritisku na start tipko zažene tekoči trak in ko v razvrščevalnik pride 6 enakobarvnih steklenic, program ustavi tekoči trak za toliko časa, dokler operater ne zamenja ustreznega zaboja ter pritisne start tipko. Med delovanjem naj na semaforju sveti zelena luč, ko se trak ustavi, pa naj sveti rdeča luč. V Simatic managerju odpremo organizacijski blok OB1 v jeziku LAD. Na simulatorju pogledamo, katere naslove vhodov in izhodov imamo na voljo. Vhodi so označeni z modro, izhodi pa z oranžno barvo.

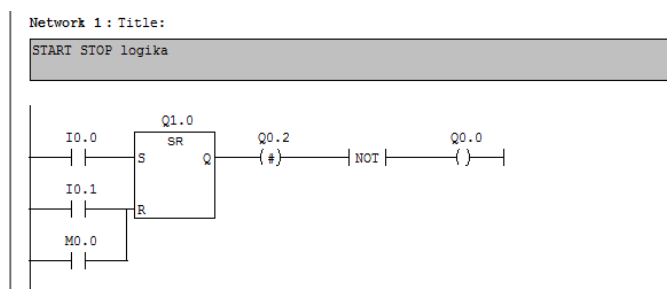


Slika 5: Simulator tekočega traku s steklenicami (Vir: lasten)

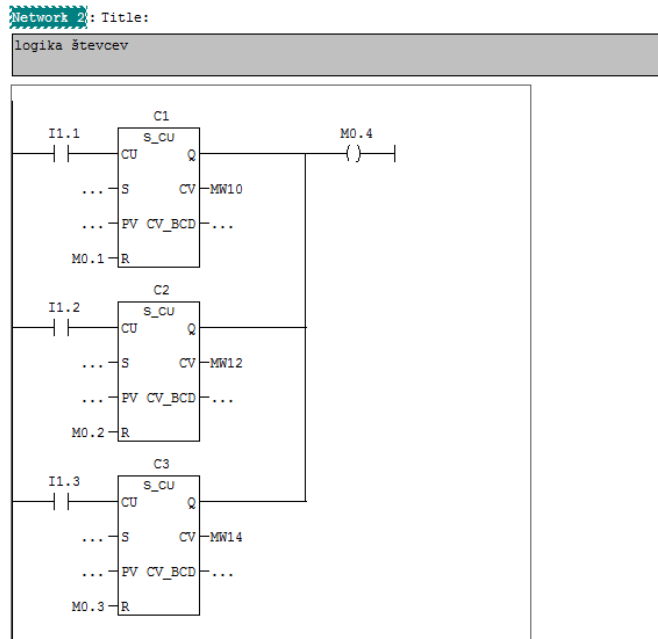
Tabela 1: Simbolna tabela

Vhod – naslov	Pomen	Izhod – naslov	Pomen
I0.0	Start tipka	Q0.0	Rdeča luč
I0.1	Stop tipka	Q0.1	Rumena luč
I0.2	Reset tipka	Q0.2	Zelena luč
I1.0	Senzor, ki zazna vse vrste barv	Q1.0	Vklop traku
I1.1	Senzor, ki zazna samo modro barvo		
I1.2	Senzor, ki zazna samo rjavo barvo		
I1.3	Senzor, ki zazna samo zeleno barvo		
I1.4	Senzor, ki zazna samo rdečo barvo		
I1.5	Senzor, ki zazna samo belo barvo		

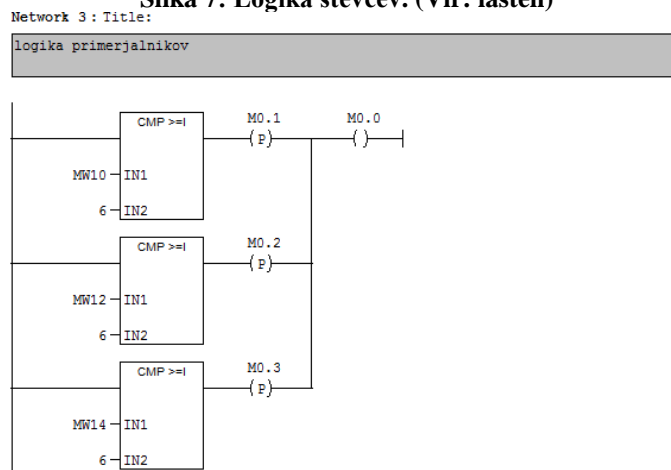
Končni prikaz programske logike napisane v LAD jeziku:



Slika 6: Vklop in izklop tekočega traku in signalizacije (Vir: lasten)



Slika 7: Logika števecv. (Vir: lasten)



Slika 8: Logika primerjalnikov (Vir: lasten)

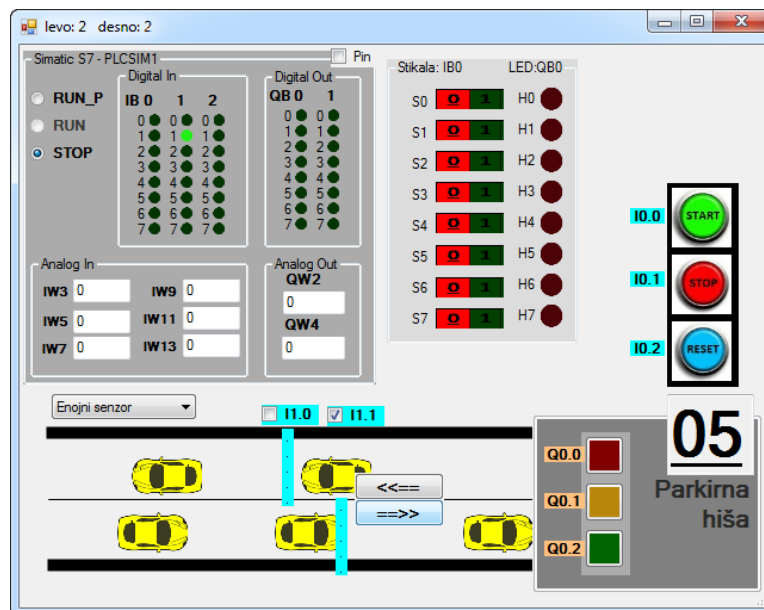
Ko smo zapisali programsko logiko, jo shranimo in z ukazom Download to module prenesemo na virtualni krmilnik. Nato v simulatorju z izbiro gumba RUN_P zaženemo virtualni krmilnik in proces se začne. S tipko start vključimo tekoči trak, medtem pa sveti zelena luč. Ko je v razvrščevalniku 6 steklenic iste barve, se mora tekoči trak s steklenicami ustaviti in čaka, dokler operater ne zamenja zaboja, kar potrdi s pritiskom na start tipko, ki zopet zažene tekoči trak in proces se nadaljuje.

Kot opazimo, za učenje programske logike ne potrebujemo drage strojne opreme, ampak samo simulator.

2.5 Parkirna hiša

Simulacija parkirne hiše je namenjena vaji s števci in primerjalniki. Gre za model preprostega parkirišča s senzorji, ki beležijo število avtomobilov v parkirni hiši. Obenem imamo tudi semafor, preko katerega lahko sporočamo kakšno je stanje v parkirni hiši in sicer: Zelena luč sporoča, da so v parkirišču še vedno prosta mesta, rdeča pa označuje, da je parkirišče popolnoma zasedeno. Na voljo imamo samo 2 vhodna senzorja, ki beležita prihode in odhode

avtomobilov s parkirišča, tipke Start, Stop in Reset, ter izhode za semafor, ki označuje zasedenost parkirišča. Fizični naslovi na krmilniku so označeni ob vsaki tipki ali senzorju ter ob luči ali aktuatorju.



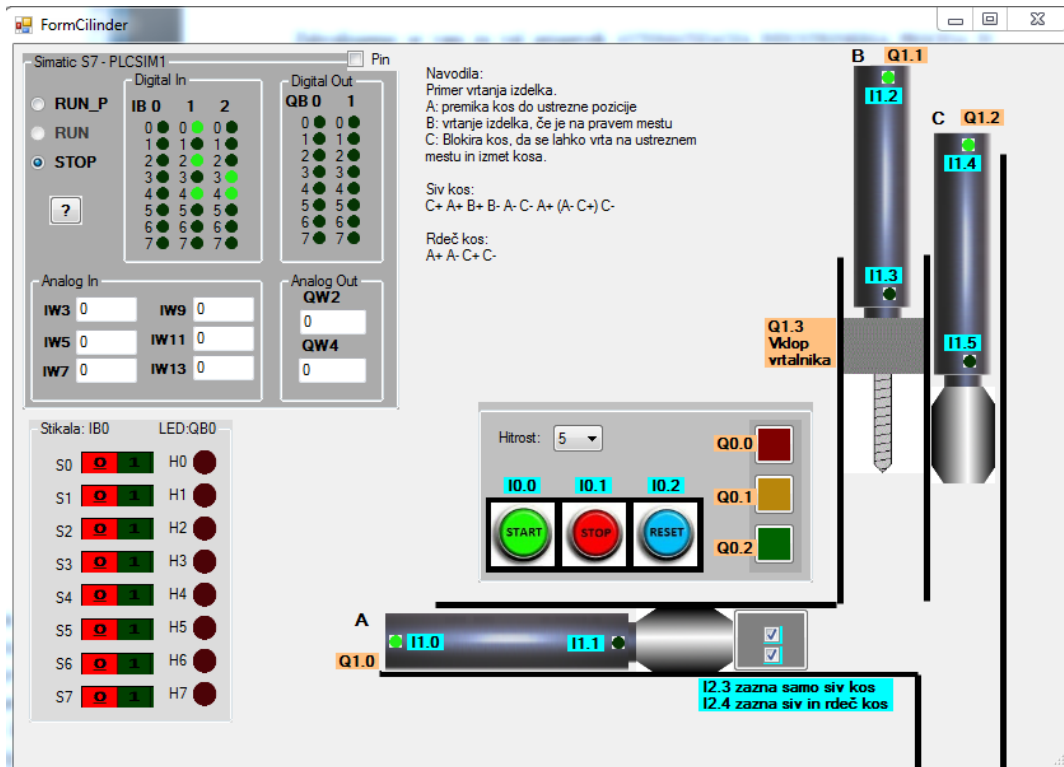
Slika 9: Simulator parkirne hiše (Vir: lasten)

Ukaz, ali naj gre avtomobil v hišo ali iz hiše, sproži uporabnik simulatorja s pritiskom na ustrezen gumb, ki se nahaja na sredini ceste med senzorji in parkirno hišo (== >> ali << ==). Pod tipkami je števec, ki prikazuje, koliko je avtomobilov v parkirni hiši in ne koliko je stanje števca v krmilniku.

2.6 Vrtanje – sekvenčno krmilje

Sekvenčna vezja vsebujejo poleg logičnih vrat tudi pomnilne elemente (flip-flope, povratne zanke itd.). Pri obravnavi sekvenčnih vezji lahko ločimo del, ki ga sestavljajo pomnilni elementi, od kombinacijskega dela vezja. Takšne strukture lahko spoznamo pri teoriji avtomatov. Kot pomnilne elemente uporabimo flip-flope in registre, kombinacijske del pa sestavimo iz logičnih vrat. So sistemi, ki vsebujejo logična vezja in spominske elemente. Izhodna funkcija ni odvisna le od trenutnega stanja vhodov, temveč tudi od stanja znotraj sistema (Weingerl, 2004).

V tem primeru gre za simulacijo vrtanja luknje s pomočjo vrtalnika in pnevmatskih cilindrov. Vaja prikazuje sklop treh cilindrov, ki se morajo gibati v točno določenem zaporedju. Vsak cilinder je opremljen s končnimi stikali, naslovi teh končnih stikal pa so zapisani na vsakem cilindru. Prav tako je vsak cilinder označen, s katerim digitalnim izhodom se ga lahko premakne v drug končni položaj. Drugi cilinder ima vpet tudi vrtalnik, katerega lahko vključimo ali izključimo z določenim digitalnim izhodom. Imamo tudi dva senzorja, katera razpoznavata barvo obdelovanca. Tipke in semafor ostajajo na istih naslovih kot v prejšnjih primerih. Zaradi lažjega sporazumevanja je vsak cilinder označen s svojo črko.



Slika 10: Simulator vrtnja (Vir: lasten)

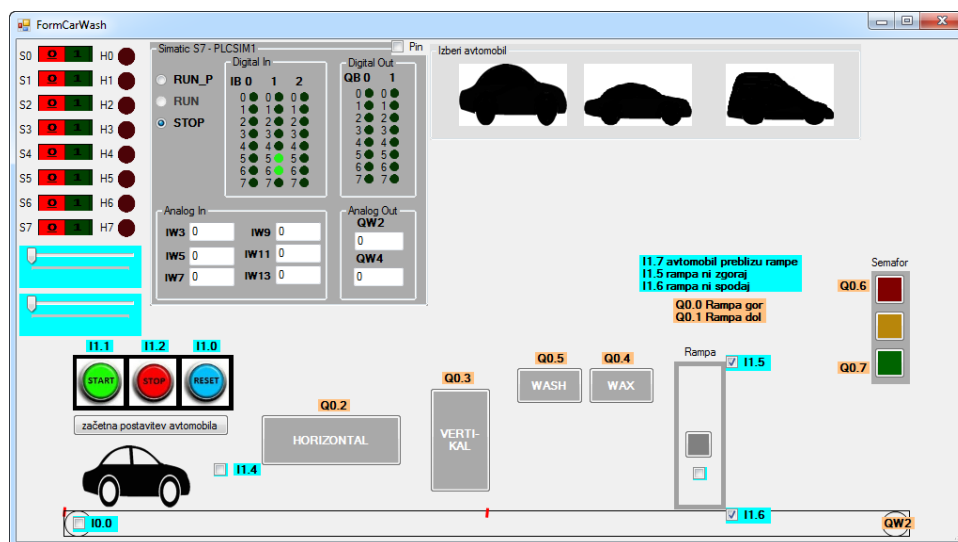
Cilj naloge je, da se v vse sive kose zvrta luknja, v rdeče pa ne. Ob pritisku na start tipko se cikel lahko začne. Ko pride obdelovanec na začetno pozicijo, preko senzorjev prepozna barvo. Če je obdelovanec rdeč, naj cilinder A potisne obdelovanec do konca, se nato umakne, cilinder C pa ga nato potisne v izmet. Če pa je obdelovanec sive barve, naj vanj zvrta luknjo. Najprej se iztegne C, ki prepreči, da bi cilinder A do konca potisnil obdelovanec. Nato se iztegne A, vendar se ne more iztegniti do končne pozicije, ker mu to preprečuje C. Ko A obmiruje na sredini, se vključi vrtnik in se začne spuščati še B, ki izvrti luknjo v obdelovanec. Nato se B umakne, umakneta se še A in C, nato se iztegne in umakne A in iztegne in umakne C. Tako je cikel končan. Med delovanjem sveti rumena luč, pred pričetkom pa zelena.

2.7 Avtopralnica

Pri simulaciji avtopralnice smo se zgledovali po fizičnem modelu »SIDEMO Car Wash«. Vsi naslovi vhodov in izhodov na simulatorju se popolnoma ujemajo s fizično napravo.



Slika 11: Fizična avtopralnica (Vir: lasten)



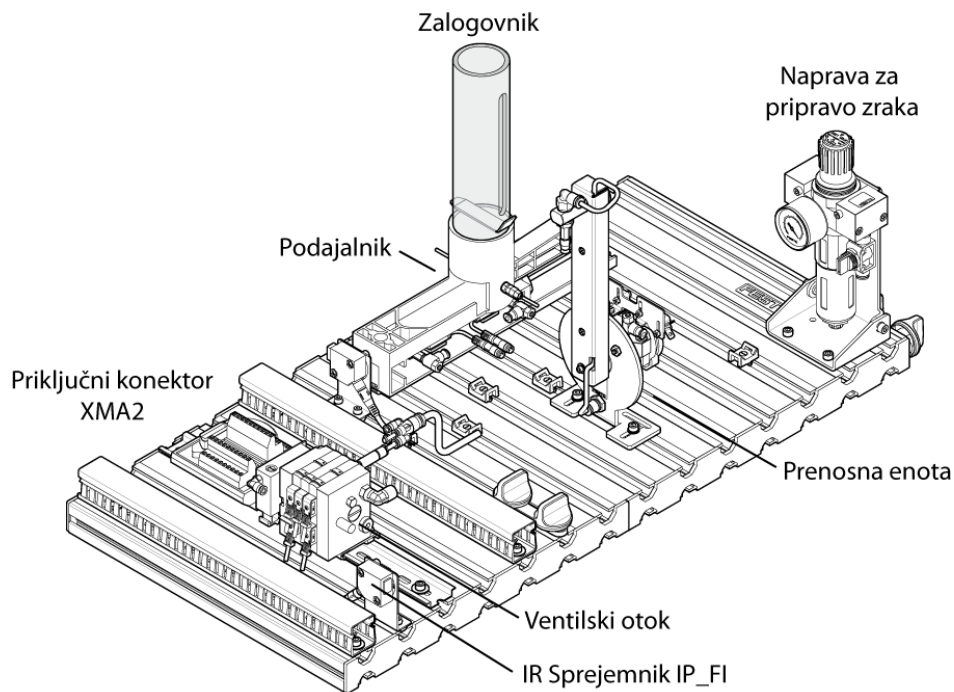
Slika 12: Simulacija avtopralnice (Vir: lasten)

Pri tem simulatorju se dijaki spoznajo z analogni in digitalni vhodi in izhodi. Ko pritisnemo Start tipko, se mora pričeti nov cikel. Vključi se tekoči trak, kateremu preko analognega izhoda QW2 določimo hitrost, in sicer od 0 do 32766, kar pomeni hitrost od 0% do 100%. Ko avto doseže senzor za pričetek, se lahko začne beležiti relativna pozicija avtomobila, preko senzorja I0.0. Tako vemo, kdaj se morata vključiti in izključiti vertikalna in horizontalna krtača. Zraven sta vključena še pranje in voskanje, ki se vključita na ustreznem mestu. Na

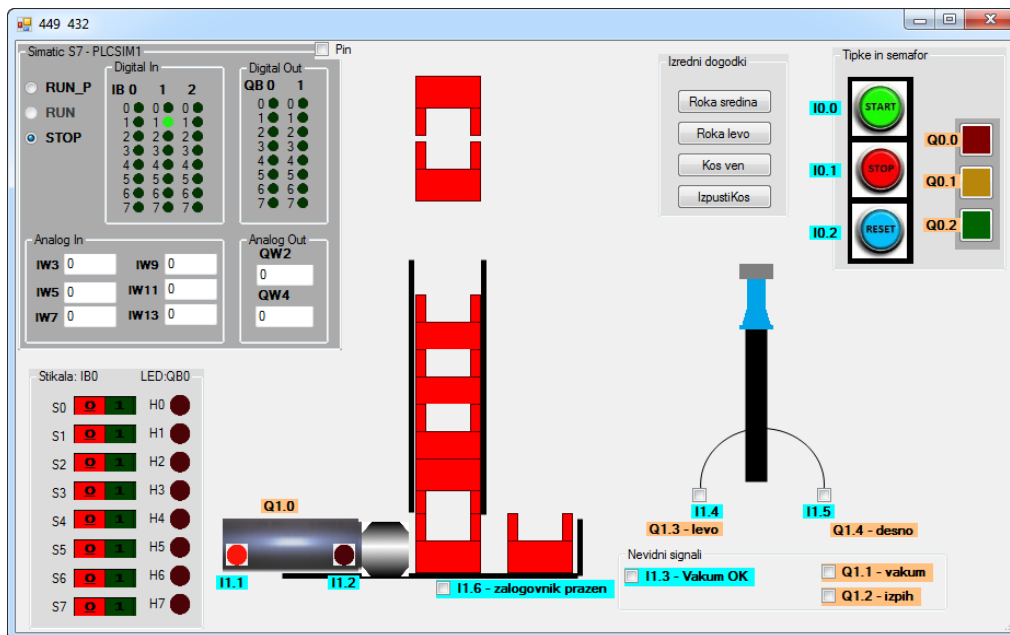
zadnjem koraku je sušilec, ki se premika vertikalno, opremljen s senzorjem, kizaznava, če je sušilec preblizu avtomobila. Sušilec ves čas sledi konturi avtomobila. Na koncu se avto ustavi in zasveti zelena luč. S kakšno hitrostjo se bo gibal avtomobil po tekočem traku, lahko preberemo tudi iz analognega vhoda IW3 ali IW5.

2.8 FESTO MPS –razdeljevalna postaja

Razdeljevalna postaja je prva iz serije FESTO MPS. Za zgled smo si vzeli fizično postajo, naslovi vhodov in izhodov pa so identični. Cilj te postaje je, da prenaša kose na naslednjo postajo. Postaja je sestavljena iz zalogovnika, podajalnika in prenosne roke. Zalogovnik je napolnjen s kosi. Ob pritisku na start tipko podajalnik potisne kos na mesto prijema. Nato se prenosna roka premakne k zalogovniku in z vklopom vakuuma prime kos, katerega prenese na drugo stran k naslednji postaji. K tej vaji so dodani tudi izredni dogodki, kot recimo na začetku je kos še na mestu prijema, kos se izgubi med prenosom, na začetku je roka že na mestu prijema, kosov ni več v zalogovniku ...



Slika 13: FESTO MPS razdeljevalna postaja. (Vir: https://wfsolutions.workforce3one.org/ws/wfsolutions/Folders/3000723343405974762/HG073-8.3_Mechatronics%20I%20Lab%20Book%20-%20Distribution%20Station.pdf)

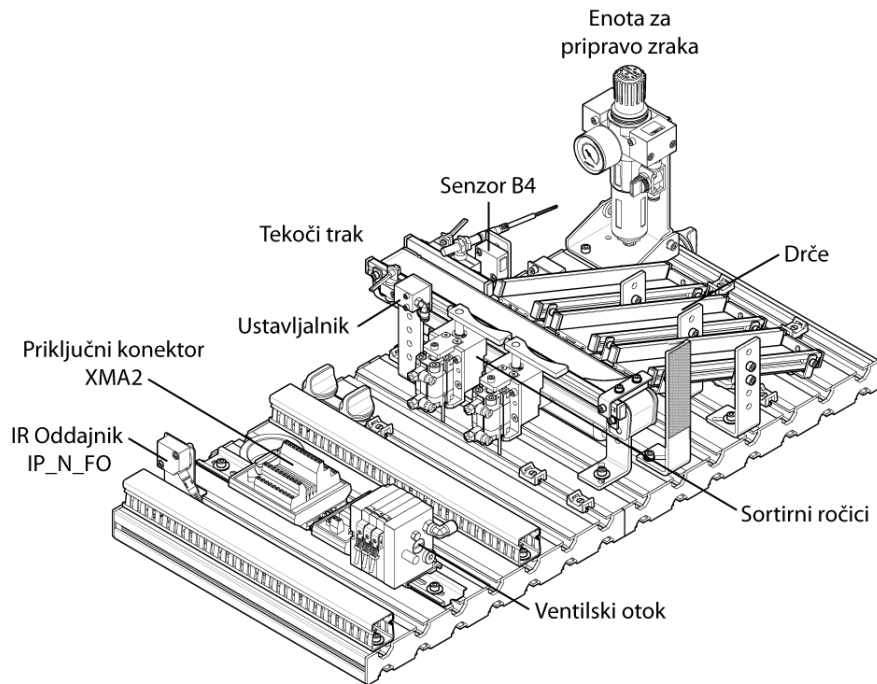


Slika 14: Simulator FESTO MPS razdeljevalna postaja (Vir: lasten)

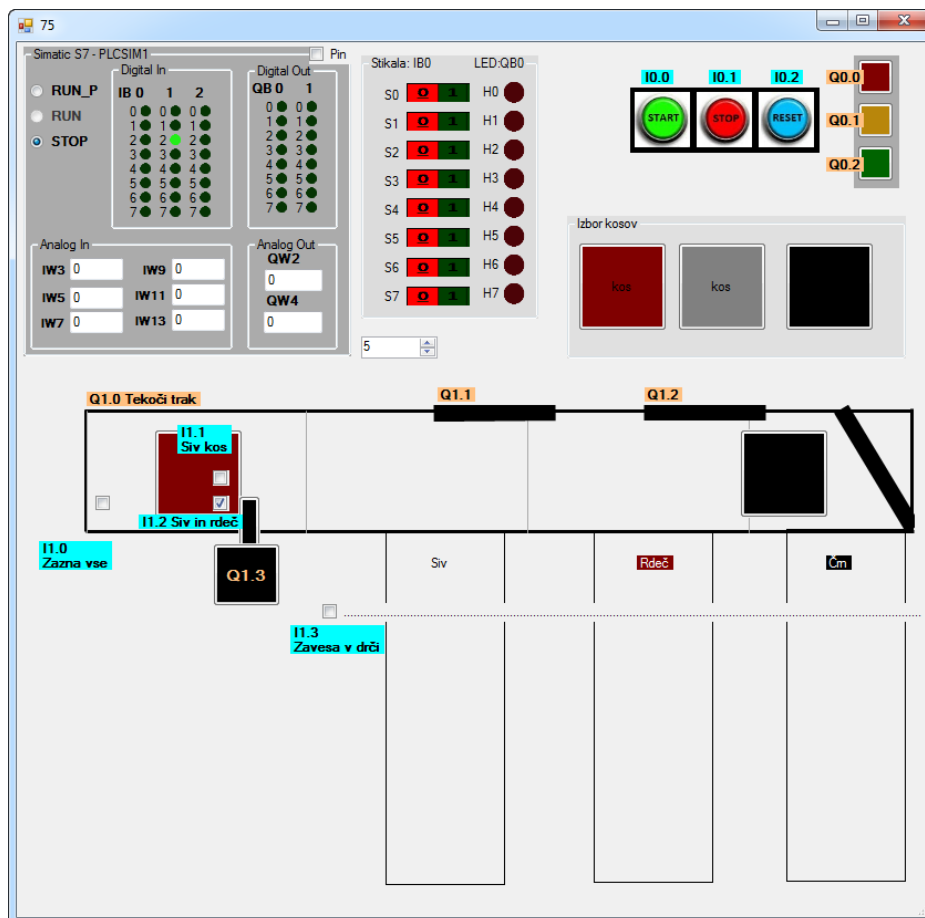
S tem morajo dijaki, ko pišejo program, dati poudarek tudi na inicializacijo programa in pa ostale izredne dogodke. Semafor in tipke so na istih naslovih kot v prejšnjih simulacijah.

2.9 FESTO MPS – sortirna postaja

Zadnja v seriji FESTO MPS je sortirna postaja, ki ima nalogo sortirati kose glede na barvo in material. Ob kliku na start tipko se mora tekoči trak izprazniti. S klikom na enega izmed kosov (rdeč, črn ali siv) se le-ta pojavi na traku. Prisotnost zazna senzor na naslovu I1.0. Vključiti se mora tekoči trak, ki pripelje kos do ustavljalnika. Ko je kos stabiliziran, preko naslednjih dveh senzorjev prepoznamo barvo kosa. V naslednjem koraku odpremo zaustavljalnik in glede na to, kakšne barve je kos, zapremo ustrezen zapah na naslovih Q1.1 ali Q1.2. Kos, ki se pelje po traku, ga nato zapah izrine na stransko drčo. Ko kos pada po drči, obenem prekine svetlobni senzor, ki zazna, da je kos v drči in cikel je tako zaključen.



Slika 15: FESTO MPS sortirna postaja. (Vir: https://wfsolutions.workforce3one.org/ws/wfsolutions/Folders/3000723343405974762/HG073-8.4_Mechatronics%20Lab%20Book%20-%20Sorting%20Station.pdf)

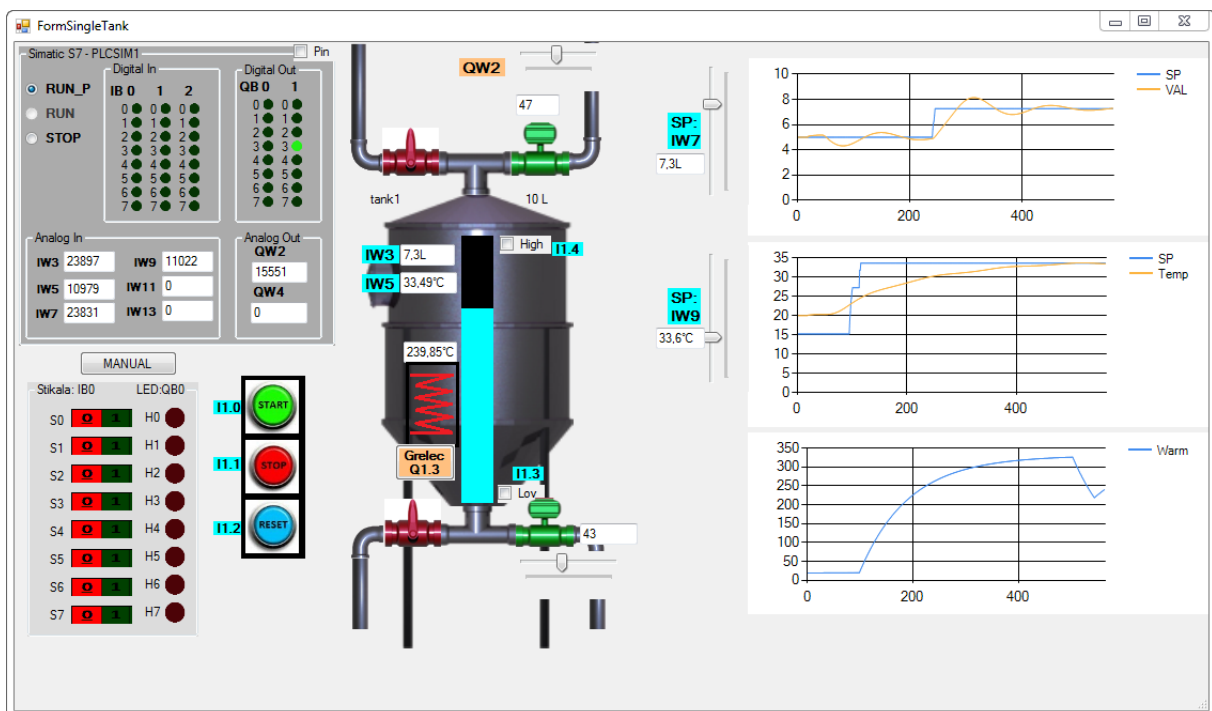


Slika 16: Simulator FESTO MPS sortirna postaja (Vir: lasten)

2.10 Procesna postaja

Regulacije parametrov v procesih so ena izmed najpomembnejših področij v tehniki nasploh, saj si naprave brez regulatorja skoraj ne moremo niti zamisliti (<http://scp.s-sceptuj.mb.edus.si/~murkos/Teorija%20in%20vaje/AVT/RegulacijaNivoja.pdf>).

Ta simulacija predstavlja rezervoar, v katerem reguliramo nivo in temperaturo tekočine, zato je namenjena regulacijam. Lahko uporabimo preprosto dvotočkovno regulacijo ali pa zahtevnejšo PID regulacijo. V tem modelu uporabljamo 2 digitalna vhoda in sicer če je nivo tekočine nad 90% ali pod 10%. Hkrati dobimo tudi preko dveh analognih vhodov nivo (IW3) in temperaturo (IW5) tekočine. Poleg tega lahko uporabimo tudi dva analogna vhoda, ki sta namenjena željeni vrednosti nivoja (IW7) in temperature (IW9), poimenovana SP ali Set point. Hkrati lahko na levi strani opazujemo grafe, ki prikazujejo Nivo tekočine in želeno vrednost, temperaturo in želeno vrednost ter samo temperaturo grelca glede na čas. Uporabljamo tudi dva izhoda, in sicer Q1.3 je namenjen vključevanju grelca in QW2, ki je namenjen krmiljenju proporcionalnega ventila, preko katerega PID uravnava dotok tekočine. Vse ostale tri ventile lahko uporabnik upravlja z miško in s tem namenoma povzroča motnjo. S tem določamo, kako naj se odzove PID regulator glede na motnjo. Na spodnji sliki je prikaz dvotočkovne regulacije temperature in PID regulacije nivoja tekočine.



Slika 17: Simulator procesne postaje (Vir: lasten)

Dobra stran simuliranja procesne avtomatizacije je v tem, da lahko preko grafov opazujemo vrednosti procesa pri različnih nastavitvah parametrov PID regulatorja.

3 Zaključek

Simulator sicer v celoti ne pokrije razumevanja tega mehatronskega sklopa, ker mora dijak še vedno najprej spoznati fizični svet, torej kako fizično ožiči naprave s krmilnikom, kako povezati pnevmatiko, priključiti motorje itd. Kar pa se tiče razlaganja programske logike in zahtevnejšega programiranja, pa

nam zelo olajša delo in iz lastnih izkušenj lahko potrdimo, da je učinek zelo pozitiven. Letos sem predlagal uporabo simulatorja tudi sodelavcu in je prav tako potrdil, da precej lažje razlaga, dijaki so bolj motivirani za delo glede na lansko leto, hitreje usvojijo snov, meni pa koristi tudi uporabniški pogled nekoga drugega, saj se s tem porodijo tudi nove ideje in kaj bi bilo potrebno še spremeniti in izboljšati. Obenem sem bil navdušen nad interesom dijakov, saj je velika večina izrazila željo, da bi se radi učili tudi doma. Simulator še nima končne podobe, ker je razvoj star eno leto in se zaenkrat malenkosti še spreminjajo glede na potrebe. Ko bo aplikacija stabilna, se bom posvetil tudi grafični podobi.

4 Viri

Avtomatizacija modelne naprave za vodenje nivoja (online). (citirano 22. 10. 2015). Dostopno na naslovu: <http://scp.s-scptuj.mb.edus.si/~murkos/Teorija%20in%20vaje/AVT/RegulacijaNivoja.pdf>.

Weingerl, P. Sekvenčna vezja (online). 2004. (citirano 22. 10. 2015). Dostopno na naslovu: <http://www.s-sers.mb.edus.si/gradiva/w3/sistemi/sekvenca.html>.

Wikipedija (online). Računalniška simulacija. (citirano 22. 10. 2015). Dostopno na naslovu: https://sl.wikipedia.org/wiki/Ra%C4%8Dunalni%C5%A1ka_simulacija.