Generator signala različitih oblika i frekvencije

Miljković, Marko

Undergraduate thesis / Završni rad

2018

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:200:987886

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni preddiplomski studij računarstva

GENERATOR SIGNALA RAZLIČITIH OBLIKA I FREKVENCIJA

Završni rad

Marko Miljković

Osijek, 2018.

Sadržaj

1. UVOD ... 1

1.1. Zadatak završnog rada ... 1

2. OPIS UPOTREBLJENIH PROGRAMA ... 2

2.1. PLC ... 2

2.2. Step 7 ... 5

2.2.1. S7-PLCSIM ... 6

2.2.2. NetToPLCsim ... 7

2.2.3. S7TrendValues ... 7

2.3. CoDeSys ... 7

3. RAZVOJ PROGRAMSKOG BLOKA ... 9

3.1. Razvoj programskog bloka u programskom alatu Step 7 ... 10

3.1.1. Organizacijski blok ... 11

3.1.2. Funkcijski blok ... 13

3.1.3. Funkcije ... 18

3.2. Razvoj programskog bloka u programskom alatu CoDeSys ... 24

3.2.1. Program ... 25

3.2.2. Funkcijski blok SIG_GEN ... 26

3.2.3. Funkcijski blokovi za proračun vrijednosti signala ... 28

4. SIMULACIJA I VIZUALIZACIJA ... 31

4.1. Simulacija i vizualizacija u programskom alatu Step 7 ... 31

4.1.1. Tablica varijabli ... 31

4.1.2. Pilasti signal ... 32

4.1.3. Sinusni signal ... 33

4.1.4. Kvadratni signal ... 34

4.1.5. Trokutasti signal ... 35

4.2. Simulacija i vizualizacija u programskom alatu CoDeSys ... 36

4.2.1. Pilasti signal ... 36

4.2.2. Sinusni signal ... 37

4.2.3. Kvadratni signal ... 38

4.2.4. Trokutasti signal ... 39

5. ZAKLJUČAK ... 41

LITERATURA ... 43

SAŽETAK ... 44

ABSTRACT ... 45

ŽIVOTOPIS ... 46

1. UVOD

Automatsko upravljanje bitna je sastavnica suvremenih tehnoloških sustava. Neke od koristi koje nam donosi jesu: poboljšana kvaliteta proizvoda, smanjenje utroška energije, smanjenje otpada, povećanje sigurnosti, smanjenje ekološkog zagađenja i sl.

U prošlosti, za zadatke automatskog upravljanja upotrebljavali su se sklopnici i releji. Takav naziv upravljanja zove se *hardwired* upravljanje. Bilo je potrebno dizajnirati dijagrame strujnog kruga, specificirati i ugraditi električne komponente, te stvoriti liste ožičenja. Električari bi tada žicama spajali komponente potrebne za izvođenje određenog zadatka. Pogreške, ili promjene funkcija zahtijevale su odspajanje i ponovno spajanje žica. To je očito iziskivalo mnogo vremena i suviše truda.

U današnje se vrijeme u tu svrhu upotrebljavaju PLC-ovi (engl. *Programmable Logic Controllers*), koji su vrsta računala. Nalaze se u komercijalnoj i industrijskoj primjeni. Mogu obavljati iste i još kompleksnije radnje od *hardwired* upravljanja. Žično spajanje između uređaja i releja izvodi se u PLC programu. Fizičko spajanje žicama vanjskih uređaja još uvijek je potrebno, ali je mnogo jednostavnije. Izmjenu funkcije i ispravljanje pogrešaka lakše je izvesti. Još neke od prednosti jesu: manja veličina, programi se odmah mogu dokumentirati, te umnožiti brže i jeftinije. Na osnovi svega navedenog vidi se zašto PLC-ovi predstavljaju veliki korak u naprijed u svijetu automatskog upravljanja.

U ovom je završnom radu najprije dan kratak uvod u PLC, te opis upotrebljenih programskih alata za razvoj programa, simulaciju i vizualizaciju funkcije generatora signala različitih oblika i frekvencija. Potom, opisani su postupci izrade programskog bloka. Na kraju, prezentirani su dobiveni rezultati, te je dan komentar autora.

1.1. Zadatak završnog rada

Zadatak je ovog završnog rada izrada programskog bloka za generiranje referentnog signala promjenjive amplitude i frekvencije. Programski blok omogućuje definiranje signala oblika pile, kvadrata, trokuta i sinusa, te u slučaju sinusnog signala definiranje faznog pomaka signala. Razvoj programa, simulacija i vizualizacija odvijaju se u programskim alatima „Step 7“ i „CoDeSys“.

2. OPIS UPOTREBLJENIH PROGRAMA

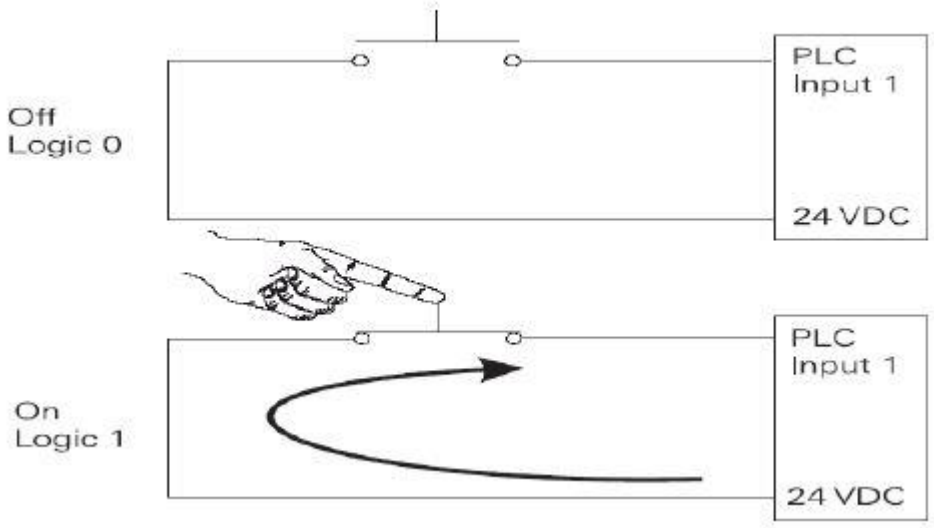
U ovom poglavlju detaljnije je objašnjen pojam PLC-a, te su opisani programski alati „Step 7“ i „CoDeSys“. Oni su upotrebljeni za razvoj, simulaciju i vizualizaciju programa za upravljanje PLC-ovima.

2.1. PLC

Kao što je rečeno u uvodu, PLC-ovi su vrsta računala. PLC nadgleda ulaze, donosi odluke na osnovi programa, te kontrolira izlaze, kako bi automatizirao proces, ili stroj. Sastoji se od ulaznih modula, procesora, te izlaznih modula. Ulaz prihvaća raznolike digitalne i analogne signale iz različitih senzora i pretvara ih u logički signal kojeg može upotrijebiti procesor. Procesor donosi odluke i izvodi upravljačke naredbe zasnovane na programskim naredbama u memoriji. Izlazni moduli pretvaraju upravljačke naredbe iz procesora, u digitalni, ili analogni signal koji služi za kontrolu različitih aktuatora. Uređaj za programiranje, najčešće osobno računalo, upotrebljava se za unos željenih instrukcija. Te instrukcije određuju što će PLC činiti za određeni ulaz. Operatorsko sučelje dozvoljava prikaz procesnih informacija i unos novih upravljačkih parametara.

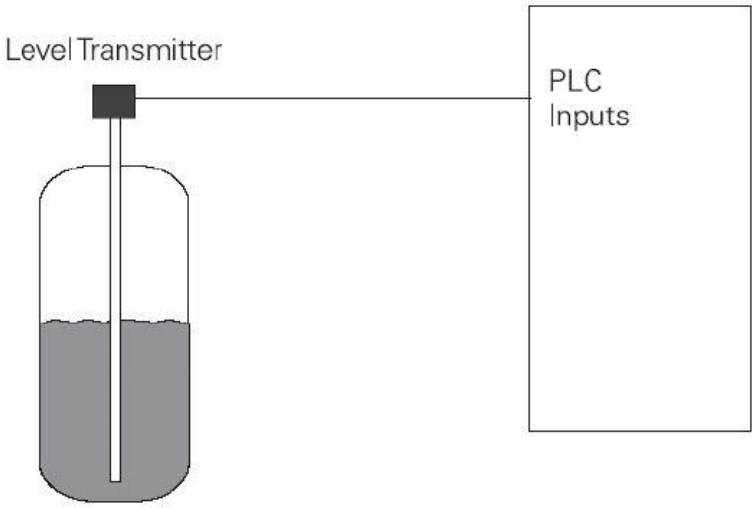
PLC-ovi *razumiju* isključivo signale koji su u pozitivnom (engl. *on*), ili negativnom (engl. *off)* stanju. Binarni sustav sastoji se od samo dva broja, to su nula i jedan. Jedinica označava da je signal prisutan, tj. da je prekidač u stanju *on*. Nula označava da signal nije prisutan, tj. prekidač je u stanju *off*.

Diskretni ulaz, koji se naziva i digitalni ulaz, ulaz je koji je ili u *on,* ili u *off* stanju. Različiti prekidači primjeri su diskretnih senzora koji su spojeni na PLC-ov diskretni ulaz. U *on* stanju, diskretni ulaz u stanju je logičke jedinice. U *off* stanju, diskretni ulaz u stanju je logičke nule. Normalno otvoreno dugme za pritisak (engl. *Normally open pushbutton,* NO) prikazano je na slici 2.1.. Jedna strana dugmeta spojena je na prvi ulaz PLC-a. Druga strana spojena je na unutrašnji električni izvor od 24 VDC (eng. *Voltage, Direct Current*). U otvorenom stanju, napon nije prisutan na ulazu. To predstavlja *off* stanje. Kada je dugme pritisnuto, 24 VDC prisutno je na ulazu. Tada je riječ o *on* stanju.



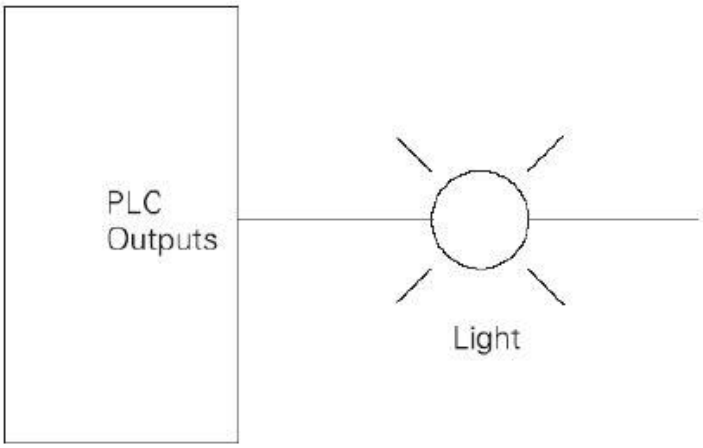
Sl. 2.1. *Normalno otvoreno dugme za pritisak [3]*

Analogni ulaz kontinuirani je, promjenjivi signal. Tipični analogni ulaz varira od 0 do 20 miliampera, ili 0 do 10 volti. Na slici 2.2. vidljiv je odašiljač razine tekućine koji nadzire razinu tekućine u spremniku. U ovisnosti o odašiljaču, signal se može povećati, ili smanjiti, kako se razina tekućine povećava, ili smanjuje.



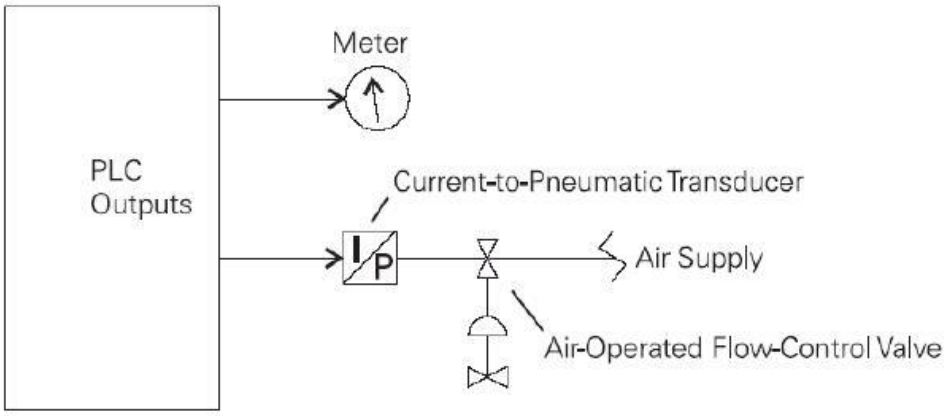
Sl. 2.2. *Odašiljač razine tekućine [3]*

Diskretni izlaz, izlaz je koji je ili u *on*, ili u *off* stanju. Naziva se još i digitalni izlaz. Zavojnice i svjetiljke primjeri su aktuatora spojenih na diskretni izlaz. Slika 2.3. prikazuje svjetiljku koja se može upaliti, ili ugasiti putem PLC izlaza na kojeg je spojena.



Sl. 2.3. *Svjetiljka spojena na diskretni izlaz PLC-a [3]*

Analogni izlaz kontinuirani je, promjenjivi signal. Izlaz može biti jednostavan poput razine od 0 do 10 VDC, koja pokreće analognu spravu za mjerenje. Primjeri izlaza analogne su sprave za mjerenje brzina, težina, temperatura. Izlazni signal može biti upotrebljen i u kompleksnijoj primjeni, poput strujno-pneumatičnog pretvarača koji upravlja ventilom upravljanim na zrak, kao što je vidljivo na slici 2.4..

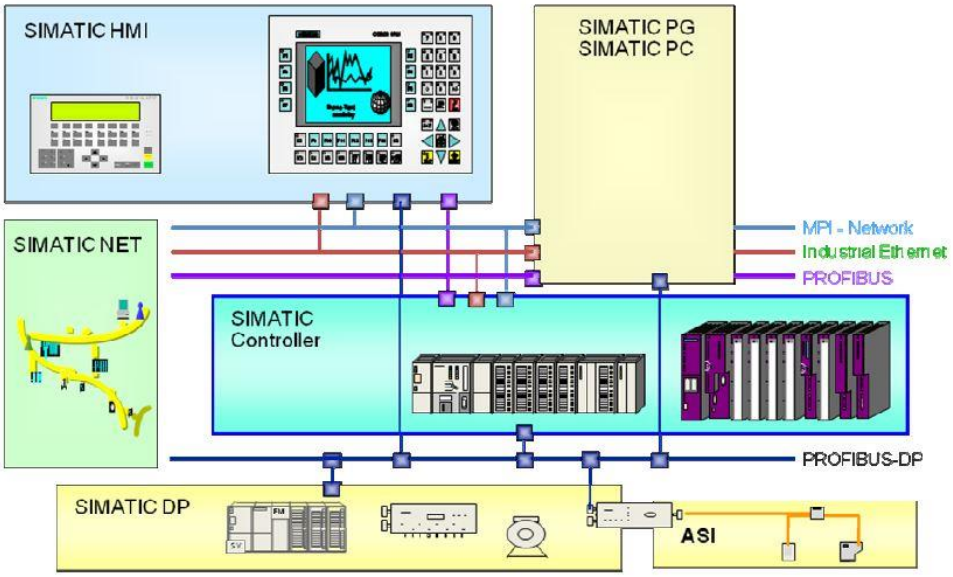


Sl. 2.4. *Analogni izlazi [3]*

2.2. Step 7

Razvoj elektronike rezultirao je velikim promjenama u automatskom upravljanju. Te promjene dovele su do proširenja, kako mogućnosti primjene automatiziranih strojeva, tako i do razvoja novih tehnologija. Uz izvor energije, upravljački elementi potrebni su za rad strojeva i procese u gotovo svim područjima proizvodnje. Potrebno je pokrenuti, nadzirati i nadgledati rad bilo kojeg stroja i procesa. U prošlosti ovi zadatci obavljali su se *hardwired* načinom upravljanja. Danas u tu svrhu rabe se PLC-ovi. Kako bi tvrtke ostale kompetitivne, nije dovoljno izolirano automatizirati samo pojedinačne procesne stanice, ili strojeve. Zahtjevi za većom fleksibilnošću i produktivnošću mogu biti ispunjeni samo kada su individualni strojevi integrirani u cijeli sustav. Tok informacija između svih komponenti bitan je za funkcioniranje cijelog sustava. Proizvodni procesi nisu više viđeni kao individualni procesi, već kao integralne komponente cijelog procesa proizvodnje. Također, cijeli proces nije više strukturiran centralno, u hijerarhijskom obliku. Sada je proces strukturiran u obliku raščlanjenih i autonomnih pojedinačnih elemenata. Potpuna integracija cijelog automatskog okruženja moguća je uz pomoć zajedničke konfiguracije i programiranja pojedinačnih sustava, zajedničkog upravljanja podatcima, te zajedničke komunikacije između svih automatskih komponenti.

„SIMATIC“ sustav upravljanja skup je koordiniranih komponenti upravljačkog procesa s jednolikom metodom konfiguracije, upravljanja podatcima, te razašiljanja podataka. „SIMATIC S7“ PLC-ovi čine osnovu sustava upravljanja. Postoje tri „SIMATIC S7“ PLC-a: S7-200, S7-300 i S7-400, koja se odabiru ovisno o željenim specifikacijama. „SIMATIC WinAC“ integracijska je platforma za upravljanje, obradu podataka, komunikaciju i tehnološke funkcije. „SIMATIC DP“ raščlanjeni U/I (ulazno/izlazni) sustav dopušta instaliranje U/I modula koji povezuju PLC sa strojem, ili procesom. „SIMATIC HMI“ (engl. *Human-Machine Interface*) nudi sve potrebne sadržaje za upravljanje i nadziranje stroja, ili procesa, od jednostavnog prikaza teksta do grafičkih operatorskih stanica. „SIMATIC NET“ povezuje sve „SIMATIC“ stanice i omogućuje komunikaciju među njima. Slika 2.5. prikazuje komponente „SIMATIC“ sustava upravljanja.



Sl. 2.5. *Komponente „SIMATIC“ sustava upravljanja [3]*

„Step 7“ programski alat kamen temeljac je potpuno integriranog sustava upravljanja. Posjeduje jednoliku konfiguraciju i programiranje, upravljanje podatcima i razašiljanje podataka. Upotrebljava se za konfiguraciju „SIMATIC“ komponenti, dodjeljivanje parametara i programiranje istih. „SIMATIC Manager“ centralni je alat za upravljanje podatcima automatskog procesa upravljanja i potrebnim programskim alatima. Svim sklopovskim i programskim zahtjevima procesa upravljanja upravlja se unutar projekta. Projekt uključuje sklopovlje, mrežu, njihovu konfiguraciju, sve programe i upravljanje podatcima za automatizaciju procesa. „SIMATIC Manager“ čuva sve podatke za automatizacijski projekt u projektnoj mapi s hijerarhijskom strukturom i sprema standardne programe i korisničke programe s mogućnošću ponovne upotrebe u biblioteke. Glavne aktivnosti koje „Step 7“ izvodi jesu: konfiguracija sklopovlja, konfiguracija komunikacijskih veza, te pisanje korisničkog programa za PLC u programskim jezicima LAD (engl. *Ladder Diagram*), FBD (engl. *Function Block Diagram*), ili STL (engl. *Statement List*) i testiranje programa *online* na PLC-u.

2.2.1. S7-PLCSIM

U programu „S7-PLCSIM“ moguće je izvršenje i testiranje „Step 7“ korisničkog programa u simuliranom PLC-u. Simulacija se izvršava na osobnom računalu. Budući da je implementacija u potpunosti ugrađena u „Step 7“ programski alat, nije potrebna konfiguracija sklopovlja.

2.2.2. NetToPLCsim

„NetToPLCsim“ program omogućava upotrebljavanje mrežne komunikacije zajedno sa „S7-PLCSIM“ programom, rabeći mrežno sučelje osobnog računala na kojem se izvršava simulacija. „NetToPLCsim“ podržava većinu funkcija koje su podržane od strane „S7-PLCSIM“ programa, poput pisanja i čitanja u podatkovnom prostoru, učitavanja programa, mrežnog pogleda programskog bloka, podrške za višestruke instance simulatora na jednom računalu i sl.

2.2.3. S7TrendValues

„S7TrendValues“ program je za praćenje rada PLC-a, ili simulatora. Nudi mogućnost nadgledanja i snimanja u stvarnom vremenu stanja svih tipova varijabli: markera, ulaza/izlaza, vremenskih brojača, brojača, baza podataka.

2.3. CoDeSys

„CoDeSys“ (engl. *Controlled Development System*) cjelokupno je razvojno okruženje za osobno računalo. Nudi jednostavan pristup IEC (engl. *International Electrotehnical Comission*) programskim jezicima na raspolaganje PLC programeru. IEC 61131-3, na kojem se zasniva „CoDeSys“, treći je dio otvorenog međunarodnog standarda IEC 61131 za programske jezike primijenjene u svrhu programiranja PLC-ova. Standard definira dva tekstualna i tri slikovna programska jezika. Tekstualni su programski jezici IL (engl. *Instruction List*) i ST (engl. *Structured Text*). Slikovni programski jezici definirani ovim standardom jesu SFC (engl. *Sequential Function Chart*), FBD (engl. *Function Block Diagram*), te LD (engl. *Ladder Diagram*). Uz sve navedene, „CoDeSys“ nudi mogućnost upotrebljavanja dodatnog slikovnog programskog jezika CFC (engl. *The Continuous Function Chart Editor*). Rabljenje uređivača i funkcija za uklanjanje grešaka temeljeno je na dokazanim razvojnim programskim okruženjima naprednih programskih jezika, poput „Visual C++“-a.

„CoDeSys“ projekt sadrži sve objekte PLC programa. Sprema se u datoteku s istim imenom. Objekti koji su sadržani unutar projekta jesu POU-ovi (engl. *Program Organization Units* – programske organizacijske jedinice), tipovi podataka, vizualizacije, resursi, te biblioteke. Proces

počinje u prvoj programskoj organizacijskoj jedinici u novom projektu, u skladu s glavnom (engl. *main*) funkcijom u „C“ programskom jeziku.

Programske su organizacijske jedinice funkcije, funkcijski blokovi i programi. Svaka programska organizacijska jedinica sastoji se od deklaracijskog dijela i tijela. Unutar tijela piše se u jednom od podržanih programskih jezika. U „CoDeSys“-u razlikuju se standardni i korisnički definirani tipovi podataka. Svaki je identifikator dodijeljen tipu podatka koji određuje koliko će se memorijskog prostora rezervirati, te koji tip vrijednosti sprema. Za vizualizaciju, što znači nadgledanje i upravljanje podatcima, nije potreban dodatni alat. Programski alat sadrži integrirani urednik za vizualizaciju, dozvoljavajući korisnicima kreiranje vizualnih objekata paralelno s razvojem programa u istom korisničkom sučelju. U kartici resursi, unutar organizatora objekata, nalaze se objekti za konfiguraciju i organizaciju projekta, te za praćenje vrijednosti varijabli. Biblioteka „standard.lib“ uvijek je dostupna. Ona sadrži sve funkcije i funkcijske blokove koji su definirani IEC61131-3 standardom kao standardne programske organizacijske jedinice za IEC sustav za programiranje. Korisnički definirane biblioteke mogu se stvoriti tako da se željeni projekt prevede i spremi u dodatnu datoteku s „.lib“ ekstenzijom. Tako stvorena datoteka može se zatim primijeniti u drugim projektima, jednako kao i standardna biblioteka.

3. RAZVOJ PROGRAMSKOG BLOKA

U ovom poglavlju prikazan je razvoj programskog bloka u programskim alatima „Step 7“ i „CoDeSys“. Zadatak je napraviti funkcijski blok naziva „SIG_GEN“ koji će imati funkciju generatora signala. Funkcijski blok, ovisno o ulazima, generira signal željenog oblika, amplitude i frekvencije, te u slučaju sinusnog signala i željenog pomaka u fazi. U tablici 3.1. vidljivi su upotrebljeni parametri.

Tab. 3.1. *Parametri funkcijskog bloka „SIG_GEN“*

Parametar	***Tip parametra***	***Tip podatka***	***Opis***
CPS	ulaz	REAL	(engl. *Cicles Per Second*), tj. frekvencija u Hertzima s gornjim limitom 0.5/*LST*
LST	ulaz	REAL	(engl. *Loop Sample Time*), tj. cikličko vrijeme izvođenja u sekundama
AMP	ulaz	REAL	(engl. *Amplitude*), tj. amplituda s kraja na kraj
PHA	ulaz	REAL	(engl. *Phase shift*), tj. fazni pomak u radijanima, upotrebljava se samo kod sinusnog signala
MODE	ulaz	INT	tip signala; 0 = pilasti signal, 1 = sinusni signal, 2 = kvadratni signal, 3 = trokutasti signal

OUT	izlaz	REAL	izlaz generatora signala frekvencije *CPS* i amplitude *AMP*

3.1. Razvoj programskog bloka u programskom alatu Step 7

U programskom alatu „Step 7“ stvoren je „OB35“ (engl. *Organization Block* – organizacijski blok) čije je cikličko vrijeme izvođenja moguće ručno definirati. „OB35“ vrši poziv FB-a (engl. *Function Block* – funkcijski blok) naziva „SIG_GEN“ sa zadanim parametrima. „SIG_GEN“ ovisno o ulaznom parametru „MODE“ vrši poziv željenog FC-a (engl. *Function* - funkcija). Postoje četiri definirana FC-a, koja pomoću matematičkih operatora izračunavaju izlaznu vrijednost jednog od četiri odabrana signala.

Upotrebljen je LAD programski jezik koji je vrlo sličan dijagramu strujnog kruga.

Na slici 3.1. vidimo otvorenu karticu „Blocks“ u „Simatic Manageru“. Ona prikazuje upotrebljene blokove u projektu. Rabljeni su sljedeći blokovi: „System Data“, „OB1“, „OB35“, „OB80“, „FB1“, „FC1“, „FC2“, „FC3“, „FC4“, „DB1“ i „VAT_“ (engl. *Variable Table*). „System Data“ sadrži listu SDB-ova (engl. *System Data Blocks*) koji predstavljaju podatkovni prostor u procesoru te sadrže postavke sustava i parametre modula. Organizacijski blokovi, funkcijski blokovi, te funkcije detaljnije su objašnjeni u nastavku, dok je tablica varijabli objašnjena u 4. poglavlju.

Object name	Symbolic name	Created in language	Size in the work me...	Type	Version (Header)
System data	---	---	---	SDB	---
OB1	Cycle Execution	LAD	38	Organization Block	0.1
OB35	Cyclic Interrupt 5	LAD	180	Organization Block	0.1
OB80	CYCL_FLT	LAD	38	Organization Block	0.1
FB1	SIG_GEN	LAD	448	Function Block	0.1
FC1	Sine wave	LAD	202	Function	0.1
FC2	Saw-tooth wave	LAD	262	Function	0.1
FC3	Triangle wave	LAD	262	Function	0.1
FC4	Square wave	LAD	228	Function	0.1
DB1		DB	58	Instance data block ...	0.0
VAT_	VAT_		---	Variable Table	0.0

Sl. 3.1. *Blokovi upotrebljeni u projektu*

3.1.1. Organizacijski blok

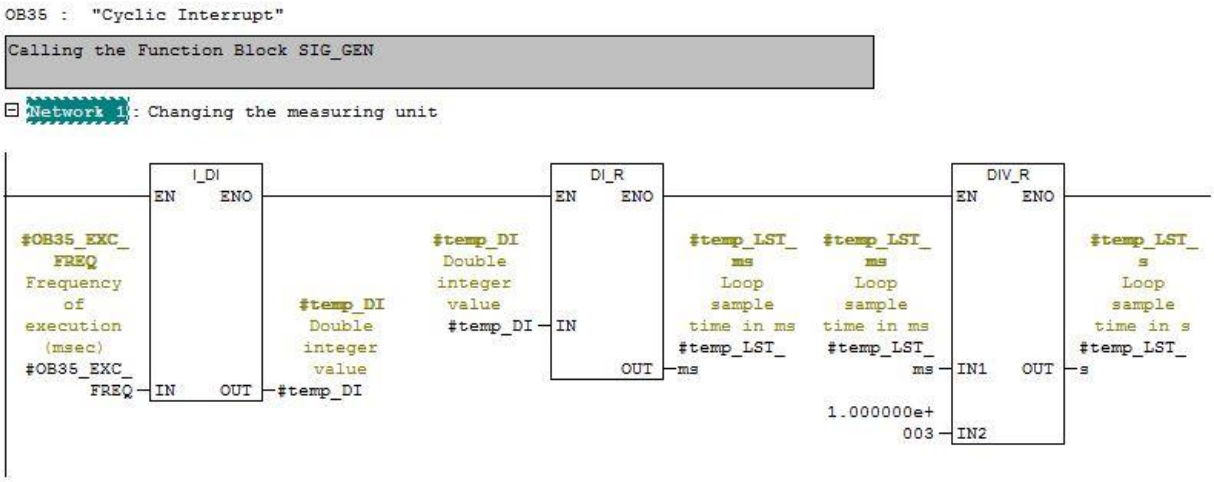
Cijeli program procesora sastoji se od operacijskog sustava i korisničkog programa. Organizacijski blokovi sučelja su između operacijskog sustava i korisničkog programa. Oni su također dio korisničkog programa, te su pozivani i obrađivani od strane operacijskog sustava kada dođe do određenog događaja. Organizacijski blokovi podijeljeni su u prioritetne razrede koji određuju slijed programskog toka. Ne mogu se pozvati iz drugih blokova, već se pozivaju od strane operacijskog sustava potaknuti određenim događajem: pokretanjem procesora, na određeno vrijeme u danu, u stalnim intervalima, kada je istekao zadani vremenski period, pri pojavi greške, kada dođe do prekida od strane sklopovlja. Organizacijski blokovi podijeljeni su u četiri skupine: organizacijski blokovi koji se izvode prilikom pokretanja programa, organizacijski blokovi koji se izvode ciklički, organizacijski blokovi koji se izvode periodički, te organizacijski blokovi koji se izvode potaknuti određenim događajem. Svi organizacijski blokovi koji se izvode periodički, te dio organizacijskih blokova koji se izvode potaknuti određenim događajem zovu se još i prekidni organizacijski blokovi. To su ime dobili jer se prekida izvođenje glavnog organizacijskog bloka, a to je „OB1“, te se izvršava jedan od prekidnih organizacijskih blokova. Drugi dio organizacijskih blokova koji se izvode potaknuti određenim događajem jesu organizacijski blokovi koji javljaju greške.

Za rješavanje zadatka „OB1“ ostaje prazan jer nije moguće ručno odrediti vrijeme potrebno za ponovno izvođenje bloka. Iz tog razloga upotrebljen je „OB35“ koji služi za izvođenje bloka u fiksnim intervalima. Zadani interval iznosi 100 milisekundi, no može se mijenjati u rasponu od 1 milisekunde do 1 minute. Na slici 3.2. vidi se deklaracijska tablica organizacijskog bloka „OB35“. Ona sadrži samo predefinirane privremene varijable, te privremene varijable „temp_DI“, „temp_LST_ms“ i „temp_LST_s“ koje služe za pretvorbu cikličkog intervala u milisekundama, u ciklički interval u sekundama.

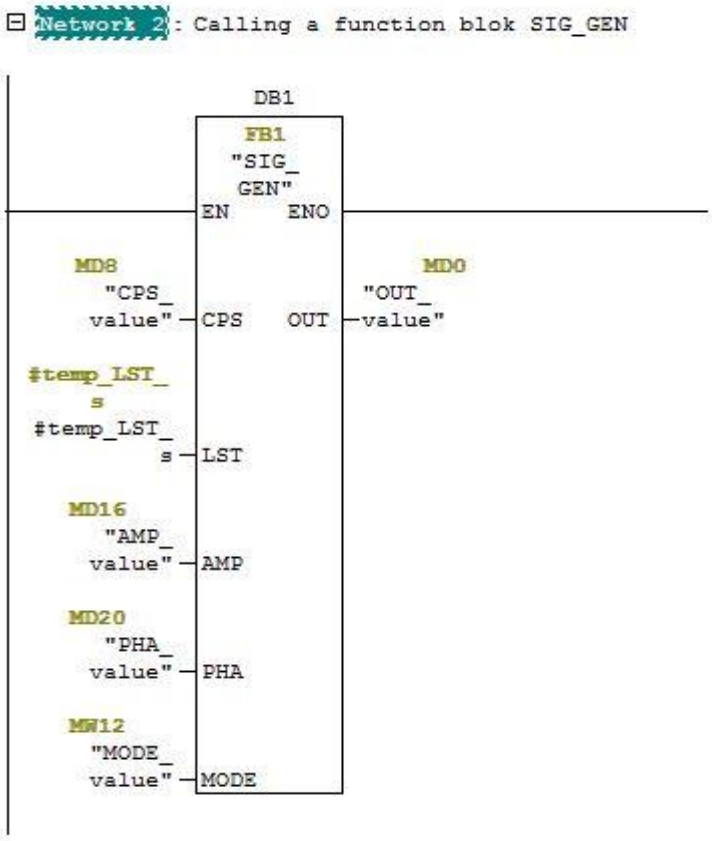
	Name	Data Type	Address	Comment
	OB35_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
	OB35_STRT_INF	Byte	1.0	16#36 (OB 35 has started)
	OB35_PRIORITY	Byte	2.0	Priority of OB Execution
	OB35_OB_NUMBR	Byte	3.0	35 (Organization block 35, OB35)
	OB35_RESERVED_1	Byte	4.0	Reserved for system
	OB35_RESERVED_2	Byte	5.0	Reserved for system
	OB35_PHASE_OFFSET	Word	6.0	Phase offset (msec)
	OB35_RESERVED_3	Int	8.0	Reserved for system
	OB35_EXC_FREQ	Int	10.0	Frequency of execution (msec)
	OB35_DATE_TIME	Date_And_Time	12.0	Date and time OB35 started
	temp_DI	DInt	20.0	Double integer value
	temp_LST_ms	Real	24.0	Loop sample time in ms
	temp_LST_s	Real	28.0	Loop sample time in s

Sl. 3.2. *Deklaracijska tablica organizacijskog bloka OB35*

Strukturu „OB35“ vidimo na slikama 3.3. i 3.4.. U ovom bloku vrši se poziv funkcijskog bloka „SIG_GEN“ i zadaju se vrijednosti parametara. Kako bi „SIG_GEN“ dobio parametre u željenom obliku, prije poziva „SIG_GEN“-a vrši se pretvorba vrijednosti predefinirane privremene varijable „OB35_EXC_FREQ“, koja predstavlja ciklički interval u milisekundama, u ciklički interval u sekundama koji se sprema u privremenu varijablu „temp_LST_s“.



Sl. 3.3. *Struktura organizacijskog bloka OB35 (pretvorba mjernih jedinica)*



Sl. 3.4. *Struktura organizacijskog bloka OB35 (poziv funkcijskog bloka „SIG_GEN“)*

Vrijednosti ostalih parametara zadaju se putem globalnih varijabli. Popis svih globalnih varijabli može se vidjeti u tablici simbola prikazanoj na slici 3.5..

S7 Program (Symbols) -- Signal_Generator_Proj\SIMATIC 400 Station\CPU 416-3 PN/DP

	Status	Symbol /	Address	Data type	Comment
1		AMP_value	MD 16	REAL	Real value of input parameter AMP
2		CPS_value	MD 8	REAL	Real value of input parameter CPS
3		CYCL_FLT	OB 80	OB 80	Cycle Time Fault
4		Cycle Execution	OB 1	OB 1	
5		Cyclic Interrupt 5	OB 35	OB 35	
6		MODE_value	MW 12	INT	Real value of input parameter MODE
7		OUT_value	MD 0	REAL	Real output signal value
8		PHA_value	MD 20	REAL	Real value of input parameter PHA
9		Saw-tooth wave	FC 2	FC 2	
10		SIG_GEN	FB 1	FB 1	
11		Sine wave	FC 1	FC 1	
12		Square wave	FC 4	FC 4	
13		Triangle wave	FC 3	FC 3	
14		var_t	MD 4	REAL	Time variable
15		VAT_	VAT 1		
16					

Sl. 3.5. *Tablica simbola*

Operacijski sustav zove „OB80“ (engl. *Time Error Organization Block*) kada god dođe do greške prilikom izvođenja nekog organizacijskog bloka, poput primjerice probijanja zadanog intervala cikličkog izvođenja. Ako „OB80“ nije programiran, procesor se zaustavi.

3.1.2. Funkcijski blok

Za razliku od funkcija, funkcijski blokovi imaju *recall* memoriju. To znači da je DB (engl. *Data Block*) pridružen funkcijskom bloku. Taj podatkovni blok naziva se *Instance Data Block*. On služi za spremanje statičkih varijabli. To su lokalne varijable koje se mogu upotrijebiti samo u funkcijskom bloku, u čijoj su deklaracijskoj tablici deklarirane. Kada se blok prestane izvršavati, vrijednosti statičkih varijabli ostanu sačuvane. Kada se pozove funkcijski blok, vrijednosti stvarnih parametara spremaju se u *Instance Data Block*. Ako se formalnom parametru ne pridijeli stvarni parametar prilikom poziva funkcijskog bloka, tada se zadnja vrijednost spremljena u *Instance Data Block*-u za taj parametar rabi prilikom izvršenja programa. Različiti se stvarni parametri mogu pridijeliti prilikom svakog poziva funkcijskog bloka. Prilikom pisanja programa unutar funkcije, morate sami tražiti prazan memorijski prostor i pridijeliti ga varijablama. Vrijednosti statičkih varijabli unutar funkcijskog bloka sačuvane su od strane Step 7 programskog alata. Još je jedna od

prednosti upotrebljavanja funkcijskog bloka ta da kod rabljenja statičkih varijabli izbjegavamo rizik zauzimanja iste memorijske lokacije više puta. Za često ponavljajuće dijelove programa upotrebljavaju se blokovi s mogućnošću dodjeljivanja parametara. Prednosti su što program mora biti stvoren samo jednom, blok je samo jednom spremljen u korisničku memoriju i možemo ga pozivati onoliko puta koliko želimo. Dodatna je prednost ta što se blok može programirati s formalnim parametrima, poput ulaza, izlaza, ulazno/izlaznih parametara, a stvarni su parametri dodijeljeni tek prilikom poziva bloka. Blokovi s mogućnošću dodjeljivanja parametara mogu biti funkcije, ili funkcijski blokovi.

Na slikama 3.6., 3.7 i 3.8. vidi se deklaracijska tablica funkcijskog bloka „SIG_GEN“. Kao ulazni formalni parametri upotrebljavaju se „CPS“, „LST“, „AMP“, „PHA“ i „MODE“. Za izlazni formalni parametar rabi se „OUT“. Privremene varijable „temp_CPS“, „temp_LST“ i“ temp_AMP“ upotrebljavaju se za spremanje vrijednosti parametara „CPS“, „LST“ i „AMP“ koje su pretvorene u odgovarajuće mjerne jedinice. Privremenu varijablu „temp_1_div_2LST“ primjenjujemo za postavljanje frekvencijskog limita.

	Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
	CPS	Real	0.0	0.000000e+000	☐	☐	Frequency in Hz
	LST	Real	4.0	0.000000e+000	☐	☐	Loop sample time in s
	AMP	Real	8.0	0.000000e+000	☐	☐	Peak-to-peak amplitude
	PHA	Real	12.0	0.000000e+000	☐	☐	Phase shift in radians
	MODE	Int	16.0	0	☐	☐	Signal type
					☐	☐	

Sl. 3.6. *Deklaracijska tablica funkcijskog bloka „SIG_GEN“ (ulazni formalni parametri)*

	Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
	OUT	Real	18.0	0.000000e+000	☐	☐	Output
					☐	☐	

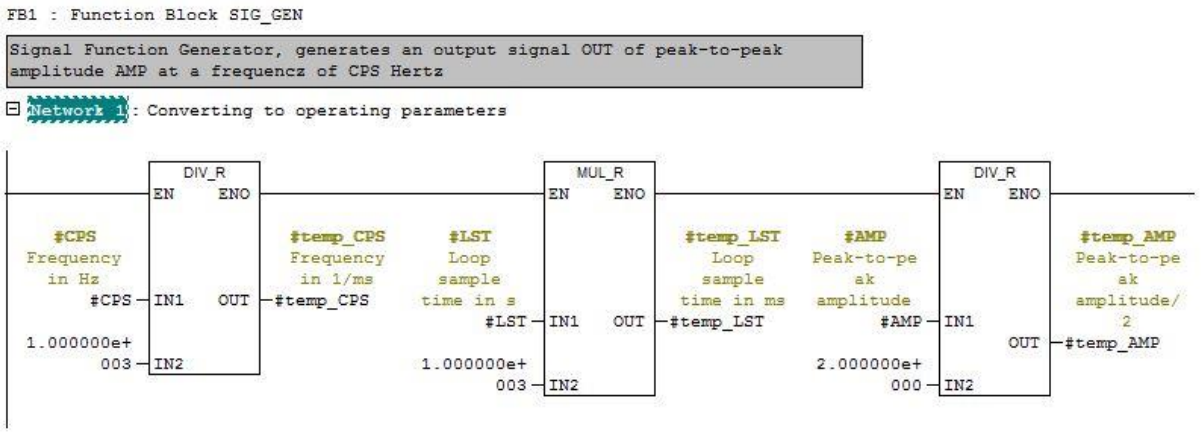
Sl. 3.7. *Deklaracijska tablica funkcijskog bloka „SIG_GEN“ (izlazni formalni parametri)*

	Name	Data Type	Address	Comment
	temp_CPS	Real	0.0	Frequency in 1/ms
	temp_LST	Real	4.0	Loop sample time in ms
	temp_AMP	Real	8.0	Peak-to-peak amplitude/2
	temp_1_...	Real	12.0	1/2*LST

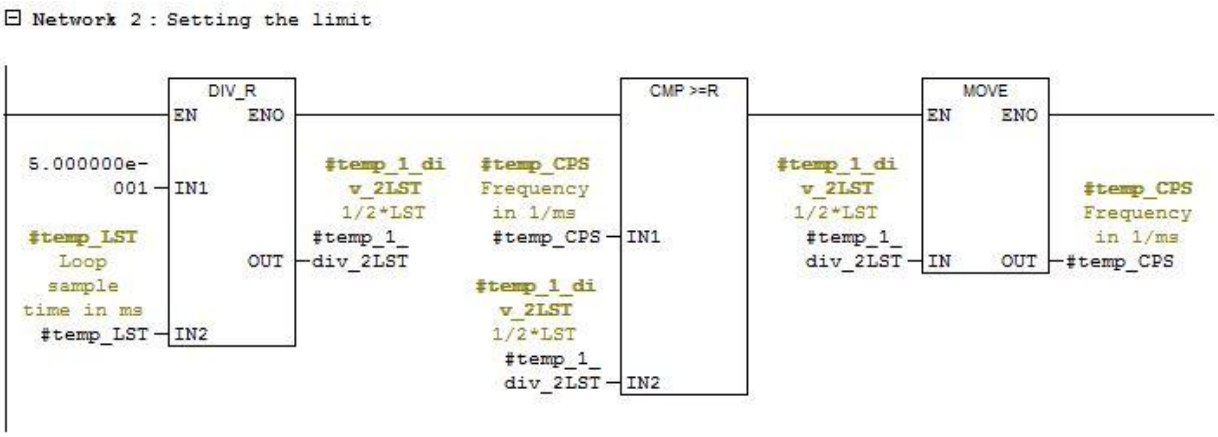
Sl. 3.8. *Deklaracijska tablica funkcijskog bloka „SIG_GEN“ (privremene varijable)*

Kao što možemo vidjeti na slikama 3.9., 3.10., 3.11., 3.12., 3.13., 3.14. i 3.15., „SIG_GEN“ najprije pretvara vrijednosti ulaznih parametara „CPS“, „LST“ i „AMP“ u odgovarajuće mjerne

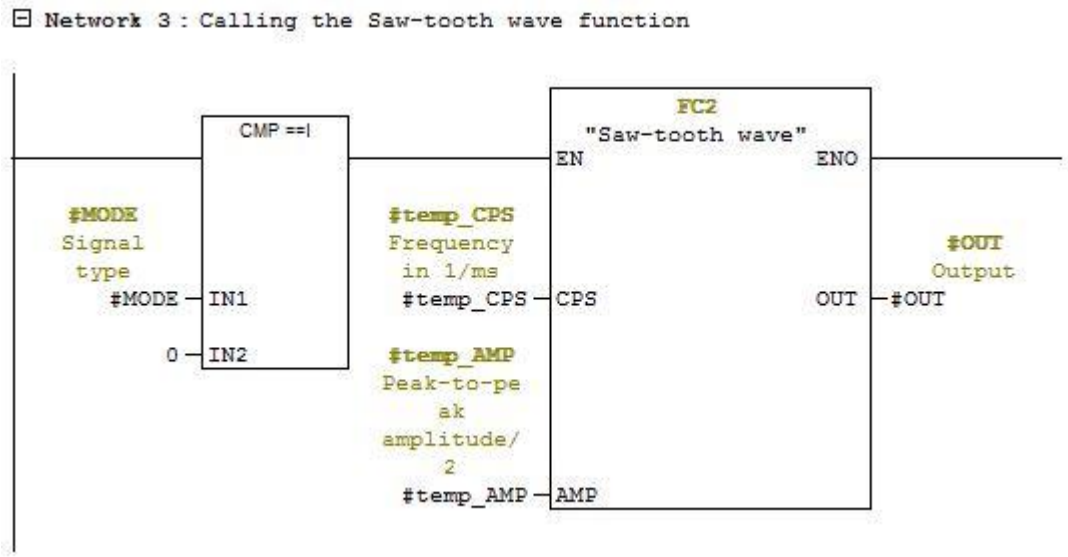
jedinice kojima se služe funkcije. Nakon toga postavlja se limit na zadanu frekvenciju „CPS“, koja ne smije biti veća od pola „LST“-a. Potom se dolazi do glavne zadaće ovog funkcijskog bloka, a to je da na temelju vrijednosti ulaznog parametra „MODE“ odredi koju će funkciju pozvati. Ako je „MODE“ jednak nuli, vrši se poziv funkcije „Saw-tooth wave“ koja računa vrijednosti potrebne za iscrtavanje pilastog signala. U slučaju da je „MODE“ jednak jedinici, poziva se funkcija „Sine wave“ za izračunavanje sinusnog signala. Kada je „MODE“ jednak dvojci poziva se funkcija „Square wave“ za izračunavanje vrijednosti kvadratnog signala. U konačnici, ako je „MODE“ jednak trojci vrši se poziv funkcije „Triangle wave“ koja računa vrijednosti potrebne za iscrtavanje trokutastog signala. Na kraju svega dodaje se vrijednost varijable „LST“, pretvorena u milisekunde, globalnoj varijabli „t“ koja predstavlja vremensku varijablu. Ta se vrijednost upotrebljava prilikom idućeg cikličkog poziva bloka.



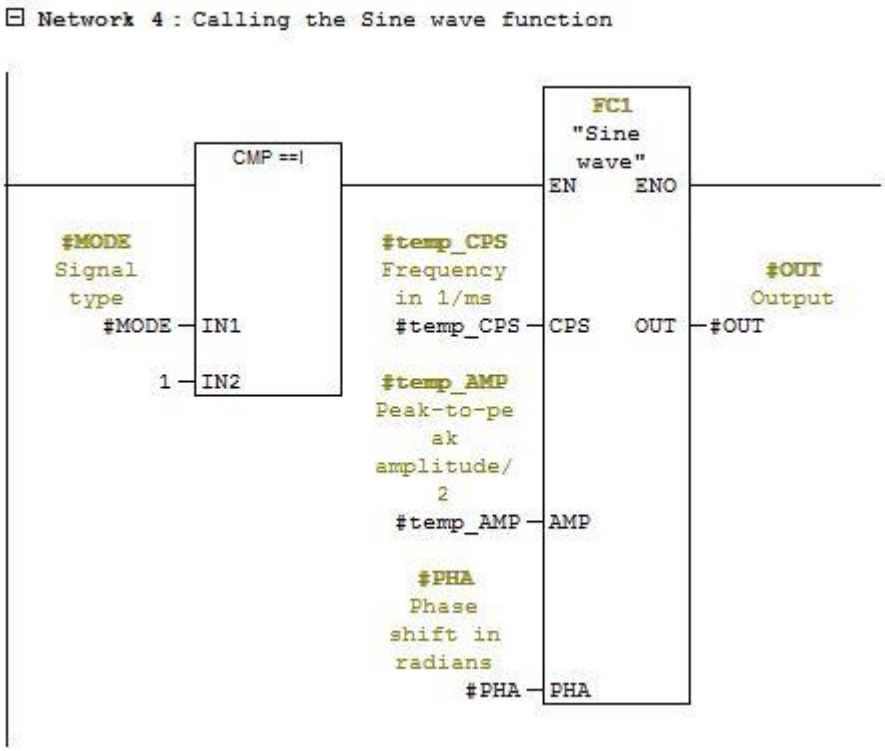
Sl. 3.9. *Funkcijski blok „SIG_GEN“ (pretvaranje mjernih jedinica)*



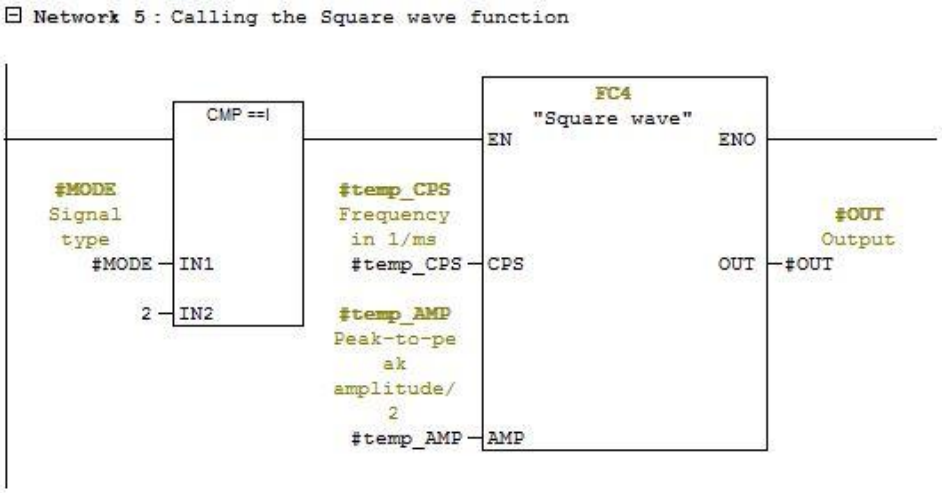
Sl. 3.10. *Funkcijski blok „SIG_GEN“ (postavljanje limita)*



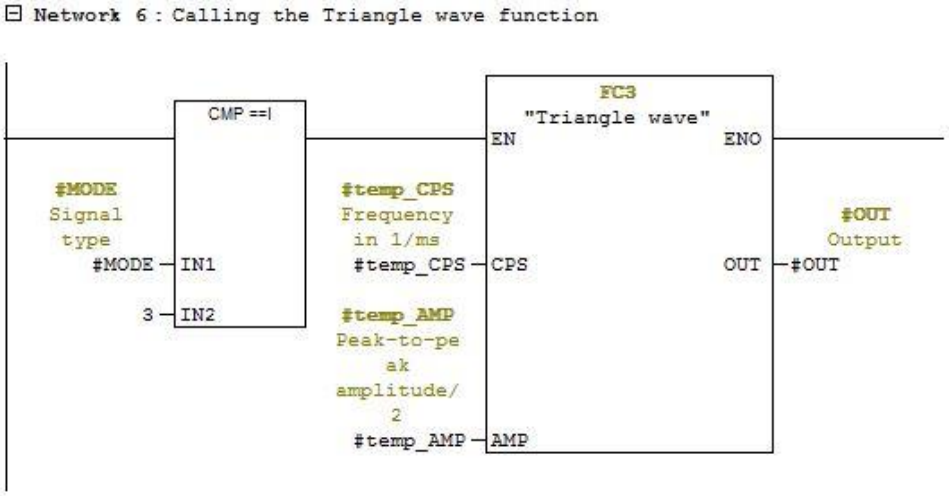
Sl. 3.11. *Funkcijski blok „SIG_GEN“ (pozivanje funkcije pilastog signala)*



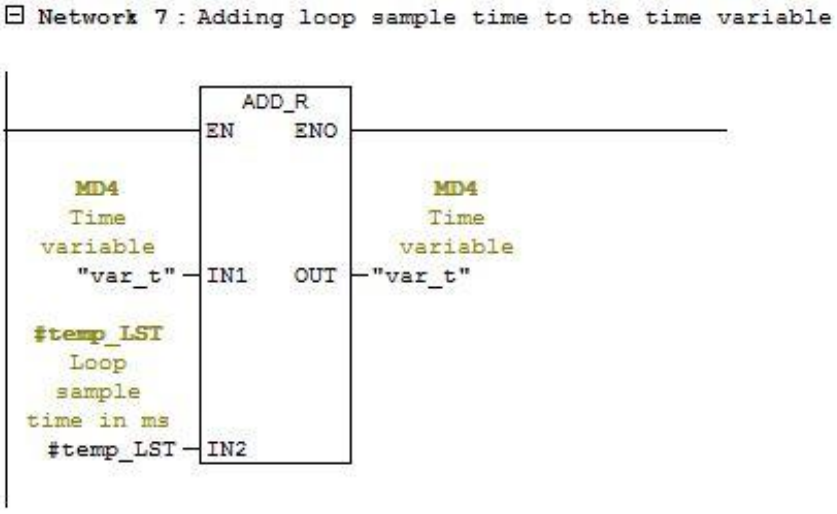
Sl. 3.12. *Funkcijski blok „SIG_GEN“ (pozivanje funkcije sinusnog signala)*



Sl. 3.13. *Funkcijski blok „SIG_GEN“ (pozivanje funkcije kvadratnog signala)*



Sl. 3.14. *Funkcijski blok „SIG_GEN“ (pozivanje funkcije trokutastog signala)*



Sl. 3.15. *Funkcijski blok „SIG_GEN“ (dodavanje „LST“-a vremenskoj varijabli)*

Na slici 3.16. vidljiv je sadržaj *Instance Data Block*-a koji je pridružen funkcijskom bloku „SIG_GEN“.

DB1 -- Signal_Generator_Proj\SIMATIC 400 Station\CPU 416-3 PN/DP

	Address	Declaration	Name	Type	Initial value	Actual valu	Comment
1	0.0	in	CPS	REAL	0.000000e...	0.000000e...	Frequency in Hz
2	4.0	in	LST	REAL	0.000000e...	0.000000e...	Loop sample time in s
3	8.0	in	AMP	REAL	0.000000e...	0.000000e...	Peak-to-peak amplitude
4	12.0	in	PHA	REAL	0.000000e...	0.000000e...	Phase shift in radians
5	16.0	in	MODE	INT	0	0	Signal type
6	18.0	out	OUT	REAL	0.000000e...	0.000000e...	Output

Sl. 3.16. *Instance Data Block*

3.1.3. Funkcije

Funkcije se primjenjuju za programiranje često ponavljajućih automatizacijskih radnji. Definiraju im se svojstva dodjeljivanjem parametara. Funkcije vraćaju povratnu vrijednost, koju nazivamo i funkcijska vrijednost, pozivajućem bloku. Povratna je vrijednost neobavezna. Kao i funkcijski blokovi, i funkcije se mogu upotrebljavati kao blokovi s mogućnošću dodjeljivanja parametara, sadržavajući formalne parametre. Za razliku od funkcijskih blokova, ne postoje statičke varijable, te funkcije ne spremaju vrijednosti. Također, ne posjeduju ni dodijeljeni podatkovni blok.

Na slikama 3.17., 3.18. i 3.19. prikazana je deklaracijska tablica funkcije „Saw-tooth wave“. Kao ulazne formalne parametre upotrebljava „CPS“ koji predstavlja frekvenciju u 1/milisekunda, i „AMP“ koji predstavlja amplitudu s kraja na kraj podijeljenu s 2 . Za izlazni formalni parametar rabi „OUT“. Privremene varijable služe za spremanje privremenih vrijednosti rezultata računskih operacija. Povratna vrijednost „RET_VAL“ nije upotrebljena.

	Name	Data Type	Comment
	CPS	Real	Frequency in 1/ms
	AMP	Real	Peak-to-peak amplitude/2

Sl. 3.17. *Deklaracijska tablica funkcije „Saw-tooth wave“ (ulazni formalni parametri)*

	Name	Data Type	Comment
	OUT	Real	Output

Sl. 3.18. *Deklaracijska tablica funkcije „Saw-tooth wave“ (izlazni formalni parametri)*

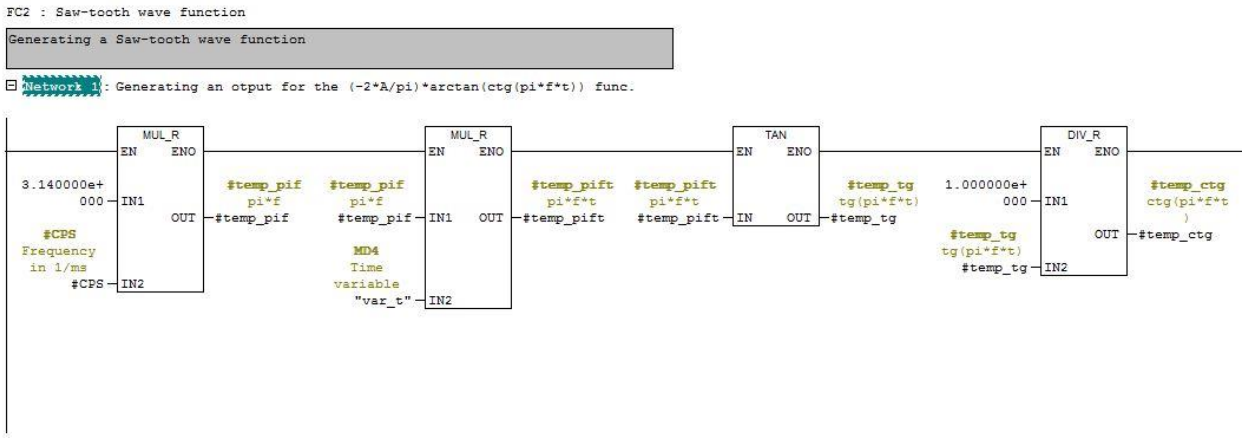
	Name	Data Type	Address	Comment
	temp_pif	Real	0.0	pi*f
	temp_pift	Real	4.0	pi*f*t
	temp_tg	Real	8.0	tg(pi*f*t)
	temp_ctg	Real	12.0	ctg(pi*f*t)
	temp_arctg	Real	16.0	arctg(ctg(pi*f*t))
	temp_ne...	Real	20.0	-2/pi
	temp_ne...	Real	24.0	-2*A/pi

Sl. 3.19. *Deklaracijska tablica funkcije „Saw-tooth wave“ (privremene varijable)*

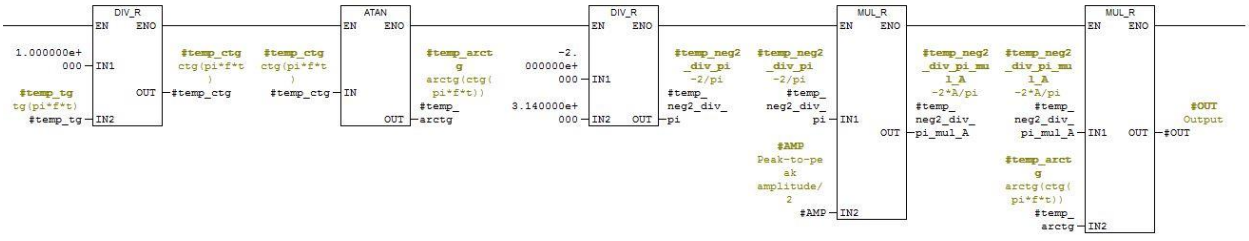
Funkcija „Saw-tooth wave“ pomoću matematičkih operacija računa vrijednost funkcije

$$OUT = \left(\frac{-2 \cdot AMP}{\pi}\right) \cdot arctg(ctg(\pi \cdot CPS \cdot t)) \qquad (3\text{-}1)$$

koja predstavlja pilasti signal. To je vidljivo na slikama 3.20. i 3.21..



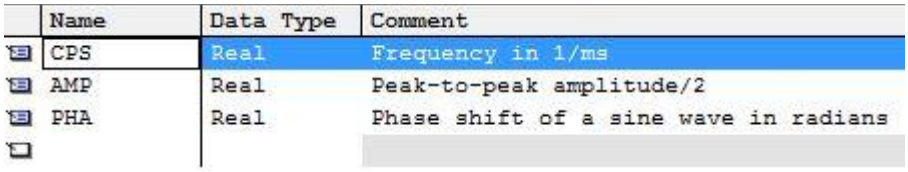
Sl. 3.20. *Funkcija „Saw-tooth wave“ (1. dio)*



Sl. 3.21. *Funkcija „Saw-tooth wave“ (2. dio)*

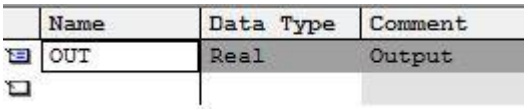
Na slikama 3.22., 3.23. i 3.24. prikazana je deklaracijska tablica funkcije „Sine wave“. Kao ulazne formalne parametre upotrebljava „CPS“, „AMP“ i „PHA“, pretvorene u odgovarajući oblik.

Za izlazni formalni parametar rabi „OUT“. Privremene varijable služe za spremanje privremenih vrijednosti rezultata računskih operacija. Povratna vrijednost „RET_VAL“ nije upotrebljena.



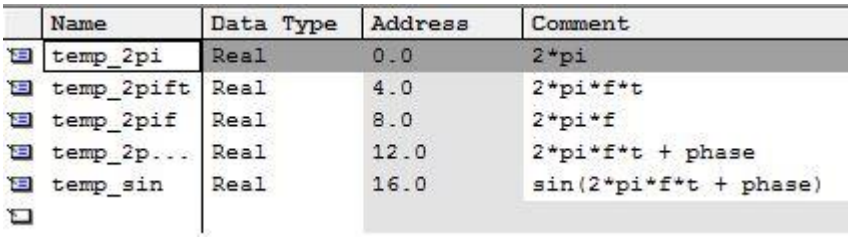
	Name	Data Type	Comment
	CPS	Real	Frequency in 1/ms
	AMP	Real	Peak-to-peak amplitude/2
	PHA	Real	Phase shift of a sine wave in radians

Sl. 3.22. *Deklaracijska tablica funkcije „Sine wave“ (ulazni formalni parametri)*



	Name	Data Type	Comment
	OUT	Real	Output

Sl. 3.23. *Deklaracijska tablica funkcije „Sine wave“ (izlazni formalni parametri)*



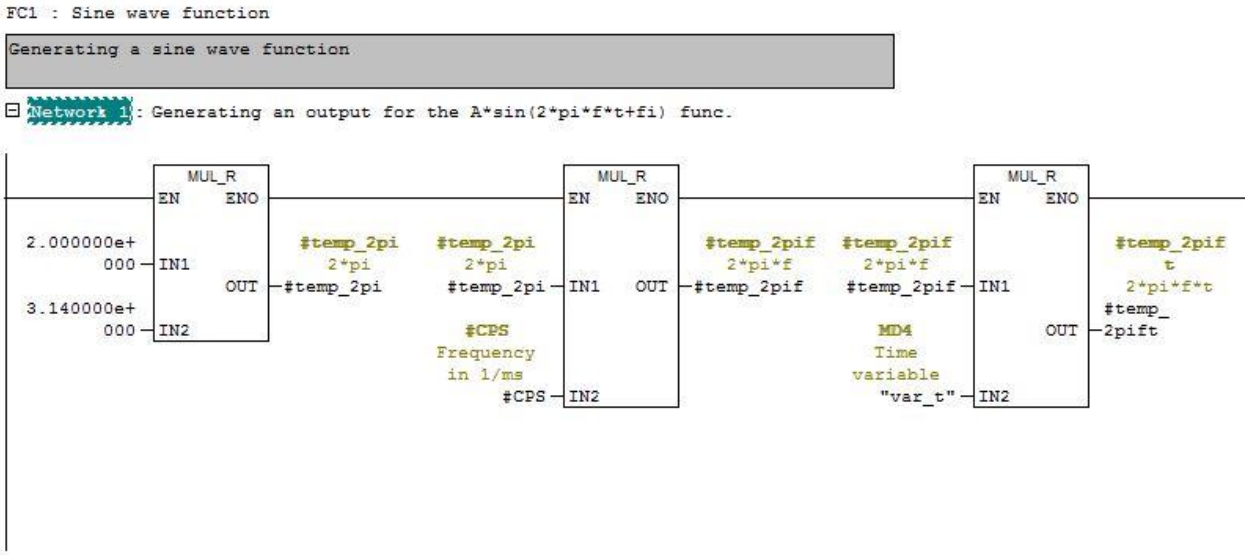
	Name	Data Type	Address	Comment
	temp_2pi	Real	0.0	2*pi
	temp_2pift	Real	4.0	2*pi*f*t
	temp_2pif	Real	8.0	2*pi*f
	temp_2p...	Real	12.0	2*pi*f*t + phase
	temp_sin	Real	16.0	sin(2*pi*f*t + phase)

Sl. 3.24. *Deklaracijska tablica funkcije „Sine wave“ (privremene varijable)*

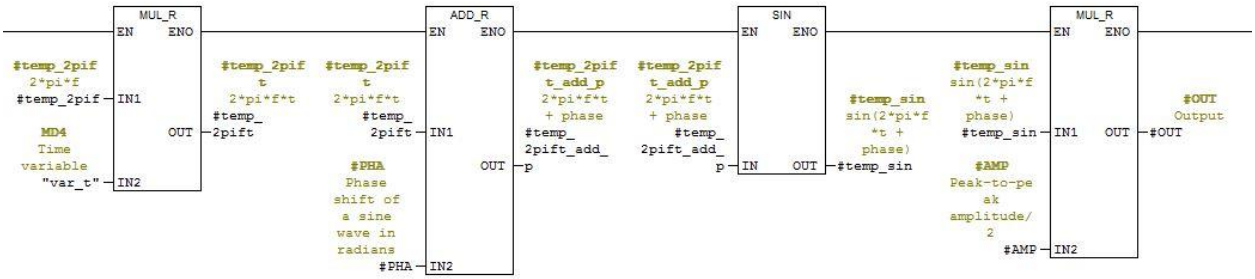
Funkcija „Sine wave“ pomoću matematičkih operacija računa vrijednost funkcije

$$OUT = AMP \cdot \sin(2 \cdot \pi \cdot CPS \cdot t + PHA) \qquad (3\text{-}2)$$

koja predstavlja sinusni signal. To je vidljivo na slikama 3.25. i 3.26..



Sl. 3.25. *Funkcija „Sine wave“ (1.dio)*



Sl. 3.26. *Funkcija „Sine wave“ (2. dio)*

Na slikama 3.27., 3.28. i 3.29. prikazana je deklaracijska tablica funkcije „Square wave“. Kao ulazne formalne parametre upotrebljava „CPS“ i „AMP“ pretvorene u odgovarajući oblik. Za izlazni formalni parametar rabi „OUT“. Privremene varijable služe za spremanje privremenih vrijednosti rezultata računskih operacija. Povratna vrijednost „RET_VAL“ nije upotrebljena.

	Name	Data Type	Comment
	CPS	Real	Frequency in 1/ms
	AMP	Real	Peak-to-peak amplitude/2

Sl. 3.27. *Deklaracijska tablica funkcije „Square wave“ (ulazni formalni parametri)*

	Name	Data Type	Comment
	OUT	Real	Output

Sl. 3.28. *Deklaracijska tablica funkcije „Square wave“ (izlazni formalni parametri)*

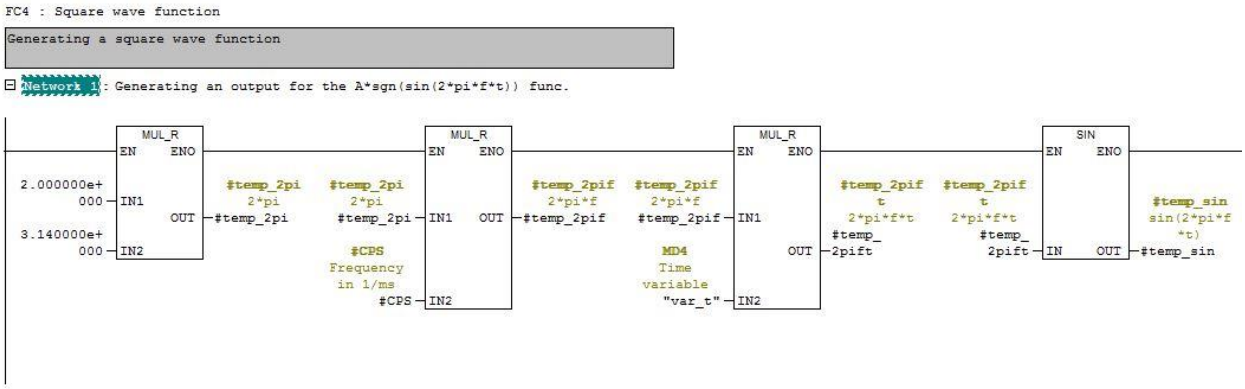
	Name	Data Type	Address	Comment
	temp_2pi	Real	0.0	2*pi
	temp_2pif	Real	4.0	2*pi*f
	temp_2pift	Real	8.0	2*pi*f*t
	temp_sin	Real	12.0	sin(2*pi*f*t)
	temp_abs	Real	16.0	| sin(2*pi*f*t) |
	temp_sgn	Real	20.0	sgn(sin(2*pi*f*t)

Sl. 3.29. *Deklaracijska tablica funkcije „Square wave“ (privremene varijable)*

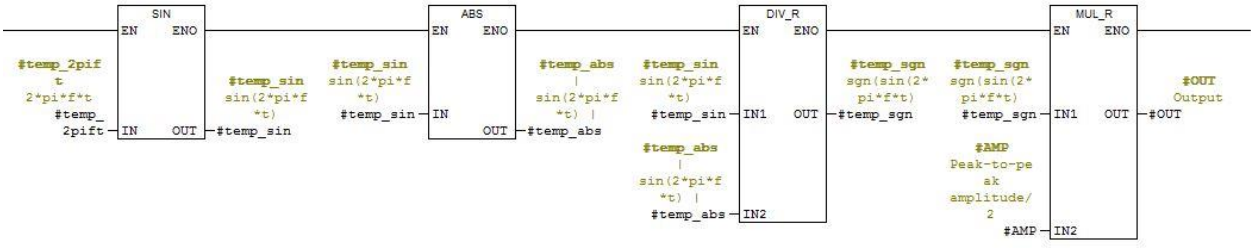
Funkcija „Square wave“ pomoću matematičkih operacija računa vrijednost funkcije

$$OUT = AMP \cdot sgn(\sin(2 \cdot \pi \cdot CPS \cdot t)) \qquad (3\text{-}3)$$

koja predstavlja kvadratni signal. To je vidljivo na slikama 3.30. i 3.31..



Sl. 3.30. *Funkcija „Square wave“ (1. dio)*



Sl. 3.31. *Funkcija „Square wave“ (2. dio)*

Na slikama 3.32., 3.33. i 3.34. prikazana je deklaracijska tablica funkcije „Triangle wave“. Kao ulazne formalne parametre upotrebljava „CPS“ i „AMP“, u odgovarajućem obliku. Za izlazni formalni parametar rabi „OUT“. Privremene varijable služe za spremanje privremenih vrijednosti rezultata računskih operacija. Povratna vrijednost „RET_VAL“ nije upotrebljena.

	Name	Data Type	Comment
	CPS	Real	Frequency in 1/ms
	AMP	Real	Peak-to-peak amplitude/2

Sl. 3.32. *Deklaracijska tablica funkcije „Triangle wave“ (ulazni formalni parametri)*

	Name	Data Type	Comment
	OUT	Real	Output

Sl. 3.33. *Deklaracijska tablica funkcije „Triangle wave“ (izlazni formalni parametri)*

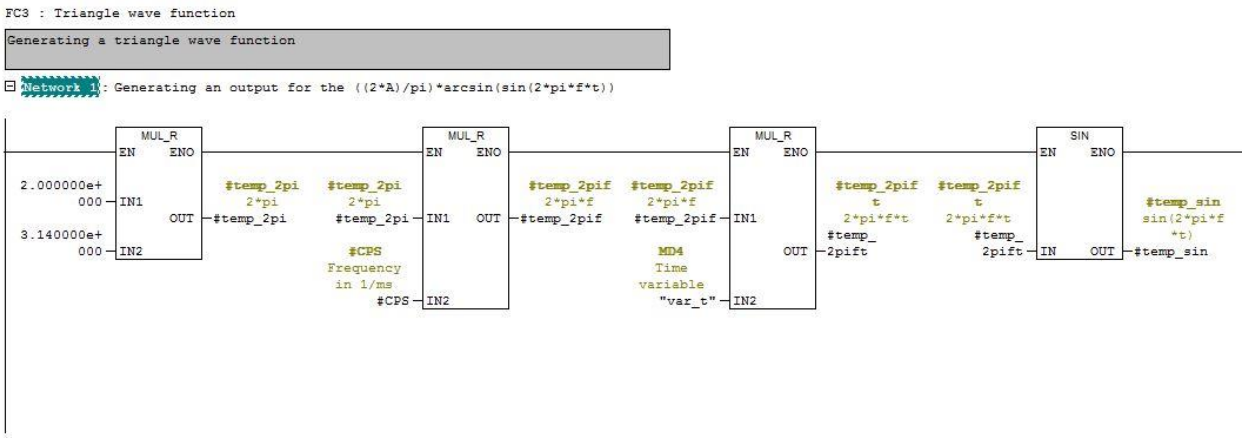
	Name	Data Type	Address	Comment
	temp_2pi	Real	0.0	2*pi
	temp_2pif	Real	4.0	2*pi*f
	temp_2pift	Real	8.0	2*pi*f*t
	temp_sin	Real	12.0	sin(2*pi*f*t)
	temp_ar...	Real	16.0	arcsin(sin(2*pi*f*t))
	temp_2_...	Real	20.0	2/pi
	temp_2_...	Real	24.0	2*A/pi

Sl. 3.34. *Deklaracijska tablica funkcije „Triangle wave“ (privremene varijable)*

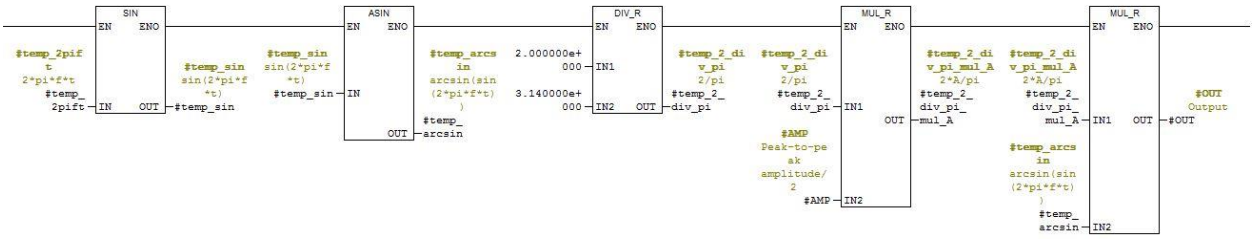
Funkcija „Triangle wave“ pomoću matematičkih operacija računa vrijednost funkcije

$$OUT = (2 \cdot \frac{AMP}{\pi}) \cdot \arcsin(\sin(2 \cdot \pi \cdot CPS \cdot t)) \qquad (3\text{-}4)$$

koja predstavlja trokutasti signal. To je vidljivo na slikama 3.35. i 3.36..



Sl. 3.35. *Funkcija „Triangle wave“ (1. dio)*



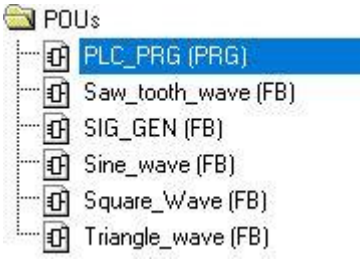
Sl. 3.36. *Funkcija „Triangle wave“ (2. dio)*

3.2. Razvoj programskog bloka u programskom alatu CoDeSys

U programskom alatu „CoDeSys“ stvoren je program „PLC_PRG“ čije je cikličko vrijeme izvođenja moguće ručno definirati. „PLC_PRG“ vrši poziv funkcijskog bloka „SIG_GEN“ sa zadanim parametrima. „SIG_GEN“ ovisno u ulaznom parametru „MODE“ vrši poziv željenog funkcijskog bloka za izračunavanje vrijednosti signala.

Primijenjen je CFC programski jezik koji je sličan FBD-u, ali umjesto mreža, radi s elementima koji slobodno stoje u prostoru.

Na slici 3.37. prikazana je otvorena kartica za programske organizacijske jedinice, unutar organizatora objekata, gdje se nalaze sve programske organizacijske jedinice koje su rabljene u projektu. Upotrebljen je program „PLC_PRG“, te funkcijski blokovi „SIG_GEN“, „Saw_tooth_wave“, „Sine_wave“, „Square_wave“ i „Triangle_wave“. U nastavku slijedi detaljnije objašnjenje navedenog.



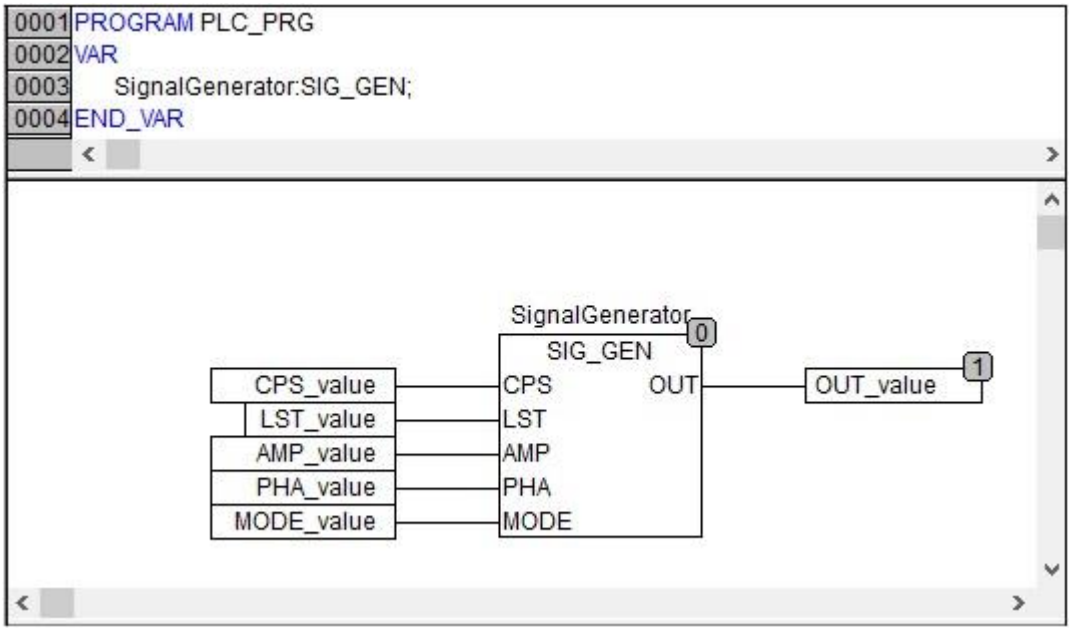
Sl. 3.37. *Programske organizacijske jedinice*

3.2.1. Program

Program je programska organizacijska jedinica koja vraća nekoliko vrijednosti tijekom izvođenja. Programi su globalno prepoznati kroz projekt. Sve su vrijednosti zadržane od zadnjeg puta kada se program izvodio. Programi mogu biti pozvani, ali ne unutar funkcije. Također ne postoje instance programa. Kada neka programska organizacijska jedinica pozove program, te ako su pri tom promijenjene vrijednosti unutar programa, tada te vrijednosti ostaju zadržane do idućeg poziva programa, čak i ako je program pozvan unutar neke druge programske organizacijske jedinice. U tome je razlika između pozivanja programa i funkcijskog bloka, gdje se mijenjaju samo vrijednosti dane instance funkcijskog bloka. Te promjene igraju ulogu samo kada je pozvana ta instanca.

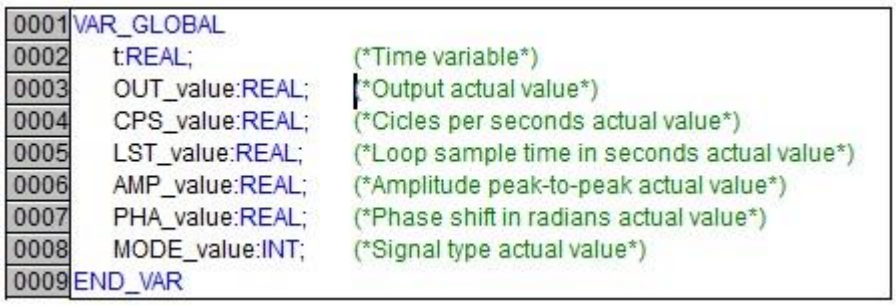
„PLC_PRG“ specijalni je predefinirani program. Svaki projekt mora ga sadržavati. On je pozivan točno jednom svaki ciklus. Prva programska organizacijska jedinica stvorena u projektu jest „PLC_PRG“.

Na slici 3.38. vidljiva je struktura „PLC_PRG“ programa za dani zadatak. U deklaracijskom uređivaču definirana je lokalna varijabla „SignalGenerator“ koja predstavlja instancu „SIG_GEN“ funkcijskog bloka. Ispod toga, nalazi se tijelo programa u kojem se on programira. „PLC_PRG“ vrši poziv funkcijskog bloka „SIG_GEN“ sa zadanim parametrima.



Sl. 3.38. *Program „PLC_PRG“*

Ulazni parametri zadani su putem globalnih varijabli „CPS_value“, „LST_value“, „AMP_value“, „PHA_value“ i „MODE_value“. Vrijednost izlaznog parametra sprema se u globalnu varijablu „OUT_value“. Popis svih globalnih varijabli upotrebljenih u projektu prikazan je na slici 3.39..



Sl. 3.39. *Globalne varijable*

3.2.2. Funkcijski blok SIG_GEN

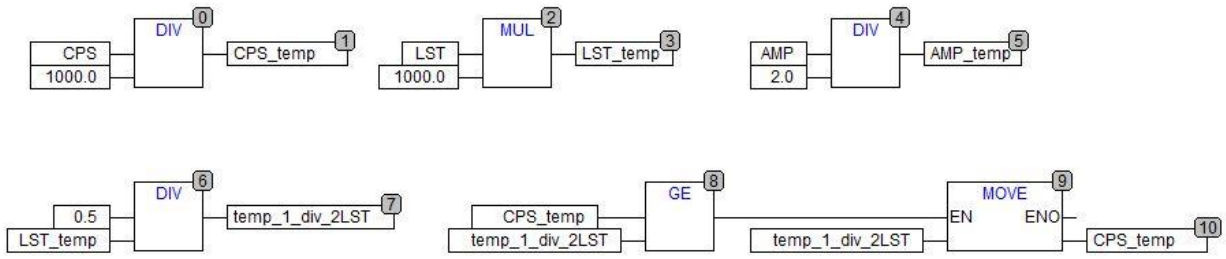
Funkcijski je blok programska organizacijska jedinica koja nudi jednu, ili više vrijednosti tijekom izvođenja. U suprotnosti funkciji, funkcijski blok nema povratne vrijednosti. Mogu se stvoriti instance funkcijskog bloka. Svaka instanca posjeduje identifikator, tj. vlastito ime, te strukturu podataka koja sadrži ulaze, izlaze, te unutarnje varijable. Instance su deklarirane lokalno, ili globalno kao varijable, gdje je ime funkcijskog bloka naznačeno kao tip identifikatora. Samo se ulaznim i izlaznim parametrima može pristupiti izvan instance funkcijskog bloka, ne i unutarnjim varijablama. Deklaracijski dijelovi funkcijskih blokova i programa mogu sadržavati deklaracije instanci. Deklaracije instanci nisu dopuštene u funkcijama. Pristup instanci funkcijskog bloka limitiran je na programsku organizacijsku jedinicu u kojoj je deklarirana, osim ako je deklarirana globalno. Ime instance funkcijskog bloka može se primijeniti kao ulaz funkcije ili funkcijskog bloka.

Na slici 3.40. vidljiv je deklaracijski uređivač funkcijskog bloka „SIG_GEN“. Ulazne varijable jesu „CPS“, „LST“, „AMP“, „PHA“ i „MODE“. Izlazna je varijabla „OUT“. Privremene varijable definirane u funkcijskom bloku jesu „CPS_temp“, „LST_temp“, „AMP_temp“ koje služe za privremeno spremanje vrijednosti ulaznih parametara, pretvorene u odgovarajuće mjerne jedinice. Privremena varijabla „temp_1_div_2LST“ služi za postavljanje frekvencijskog limita. Uz navedene privremene lokalne varijable, tu su još i „S0“, „S1“, „S2“ i „S3“ koje predstavljaju instance funkcijskih blokova „Saw_tooth_wave“, „Sine_wave“, „Square_wave“ i „Triangle wave“.

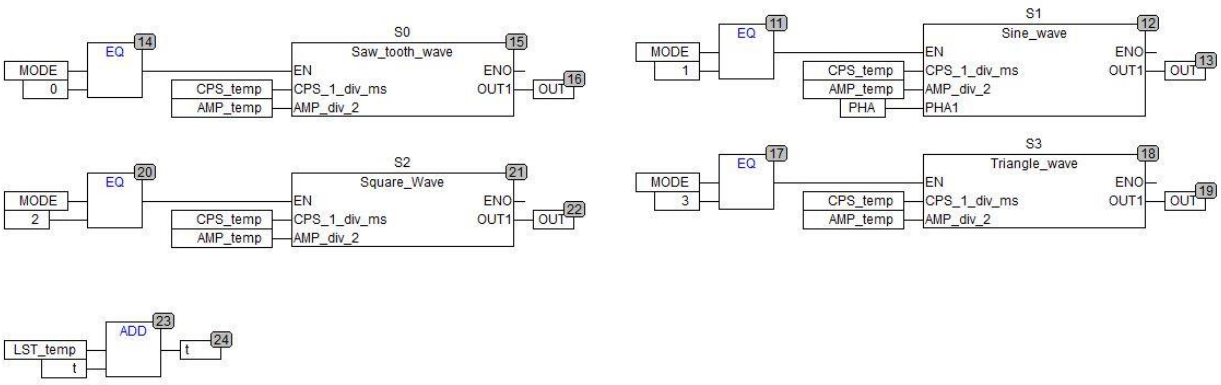
```
0001 FUNCTION_BLOCK SIG_GEN
0002 VAR_INPUT
0003     CPS:REAL;     (*Cicles per second*)
0004     LST:REAL;     (*Loop sample time in seonds*)
0005     AMP:REAL;     (*Amplitude peak-to-peak*)
0006     PHA:REAL;     (*Phase shift in radians*)
0007     MODE:INT;     (*Signal type*)
0008 END_VAR
0009 VAR_OUTPUT
0010     OUT:REAL;     (*Signal generator wave output*)
0011 END_VAR
0012 VAR
0013     CPS_temp:REAL;              (*Cicles per milisecond*)
0014     LST_temp:REAL;              (*Loop sample time in miliseconds*)
0015     AMP_temp:REAL;              (*Amplitude/2*)
0016     temp_1_div_2LST:REAL;       (*1/2*LST*)
0017     S0:Saw_tooth_wave;
0018     S1:Sine_wave;
0019     S2:Square_Wave;
0020     S3:Triangle_wave;
0021 END_VAR
0022
0023
```

Sl. 3.40. *Deklaracijski uređivač funkcijskog bloka „SIG_GEN“*

Funkcijski blok „SIG_GEN“ na sličan je način implementiran kao i u potpoglavlju 3.1.2., što je vidljivo na slikama 3.41. i 3.42..



Sl. 3.41. *Funkcijski blok „SIG_GEN“ (pretvaranje mjernih jedinica i postavljanje limita)*



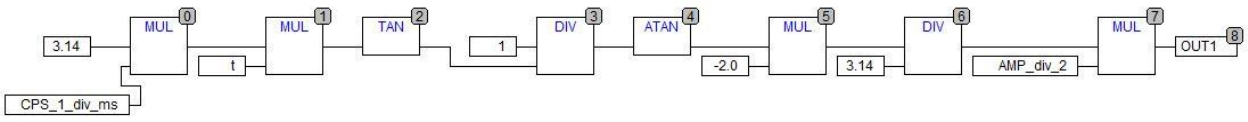
Sl. 3.42. *Funkcijski blok „SIG_GEN“ (pozivanje funkcijskih blokova i dodavanje vrijednosti „LST“-a vremenskoj varijabli „t“)*

3.2.3. Funkcijski blokovi za proračun vrijednosti signala

Funkcijski su blokovi implementirani na sličan način kao i funkcije u potpoglavlju 3.1.3., stoga je njihov opis preskočen. Na slikama 3.43., 3.44., 3.45., 3.46., 3.47., 3.48., 3.49., 3.50. prikazani su deklaracijski i programski uređivači pojedinih funkcijskih blokova.

```
0001 FUNCTION_BLOCK Saw_tooth_wave
0002 VAR_INPUT
0003     CPS_1_div_ms:REAL;    (*Cicles per second in 1/ms*)
0004     AMP_div_2:REAL;       (*Amplitude/2*)
0005 END_VAR
0006 VAR_OUTPUT
0007     OUT1:REAL;
0008 END_VAR
0009 VAR
0010 END_VAR
```

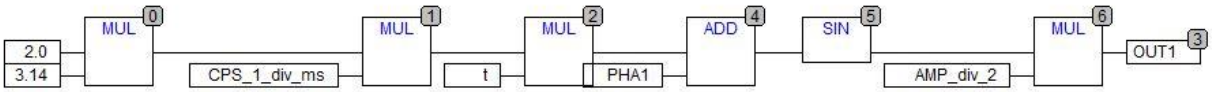
Sl. 3.43. *Deklaracijski uređivač funkcijskog bloka „Saw_tooth_wave“*



Sl. 3.44. *Funkcijski blok „Saw_tooth_wave“*

```
0001 FUNCTION_BLOCK Sine_wave
0002 VAR_INPUT
0003     CPS_1_div_ms:REAL;    (*Cicles per second in 1/ms*)
0004     AMP_div_2:REAL;       (*Amplitude/2*)
0005     PHA1:REAL;            (*Phase shift in radians*)
0006 END_VAR
0007 VAR_OUTPUT
0008     OUT1:REAL;
0009 END_VAR
0010 VAR
0011 END_VAR
```

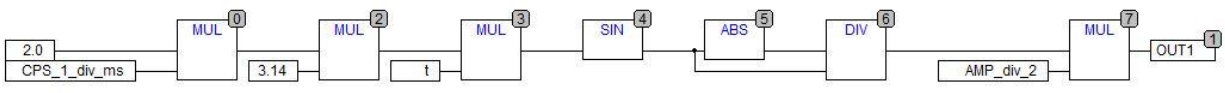
Sl. 3.45. *Deklaracijski uređivač funkcijskog bloka „Sine_wave“*



Sl. 3.46. *Funkcijski blok „Sine_wave“*

```
0001 FUNCTION_BLOCK Square_Wave
0002 VAR_INPUT
0003     CPS_1_div_ms:REAL;    (*Cicles per second in 1/ms*)
0004     AMP_div_2:REAL;       (*Amplitude/2*)
0005 END_VAR
0006 VAR_OUTPUT
0007     OUT1:REAL;
0008 END_VAR
0009 VAR
0010 END_VAR
```

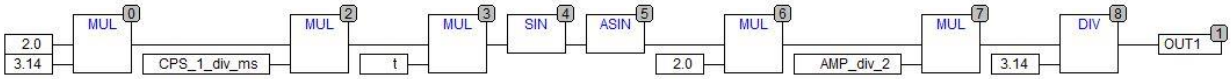
Sl. 3.47. *Deklaracijski uređivač funkcijskog bloka „Square_wave“*



Sl. 3.48. *Funkcijski blok „Square_wave“*

```
0001 FUNCTION_BLOCK Triangle_wave
0002 VAR_INPUT
0003     CPS_1_div_ms:REAL;    (*Cicles per second in 1/ms*)
0004     AMP_div_2:REAL;       (*Amplitude/2*)
0005 END_VAR
0006 VAR_OUTPUT
0007     OUT1:REAL;
0008 END_VAR
0009 VAR
0010 END_VAR
```

Sl. 3.49. *Deklaracijski uređivač funkcijskog bloka „Triangle_wave“*



Sl. 3.50. *Funkcijski blok „Triangle_wave“*

4. SIMULACIJA I VIZUALIZACIJA

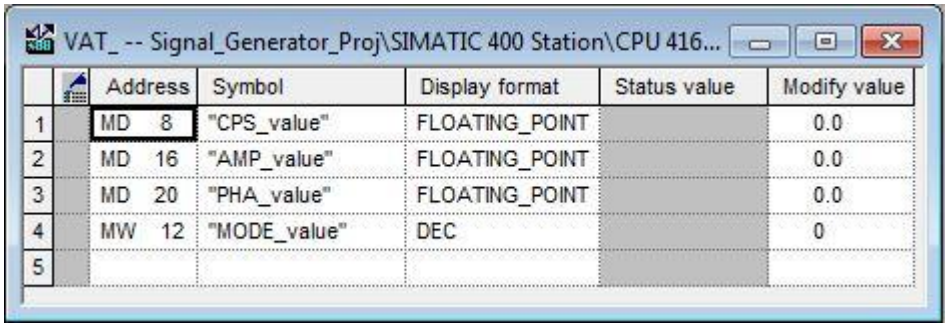
U ovom poglavlju prikazani su rezultati simulacije i vizualizacije u programskim alatima „Step 7“ i „CoDeSys“. Na temelju različitih vrijednosti parametara „CPS“, „LST“, „AMP“, „PHA“ i „MODE“ dobit će se različiti odzivi koji će potom biti analizirani.

4.1. Simulacija i vizualizacija u programskom alatu Step 7

U programskom alatu „Step 7“ unos stvarnih parametara, simulacija i vizualizacija vrše se na sljedeći način. Najprije se pokreće „S7-PLCSIM“ simulator, te se u njega skidaju programirani blokovi i podatci sustava. Potom pomoću „NetToPLCsim“ programa povezuje se „S7TrendValues“ program sa simulatorom, u svrhu praćenja vrijednosti globalne varijable „OUT_value“ u koju se ispisuju vrijednosti izlaznog parametra „OUT“. Unos željenih vrijednosti u globalne varijable „CPS_value“, „AMP_value“, „PHA_value“ i „MODE_value“ vrši se iz tablice varijabli. Vrijednost parametra „LST“ predefinirana je i iznosi 100 milisekundi.

4.1.1. Tablica varijabli

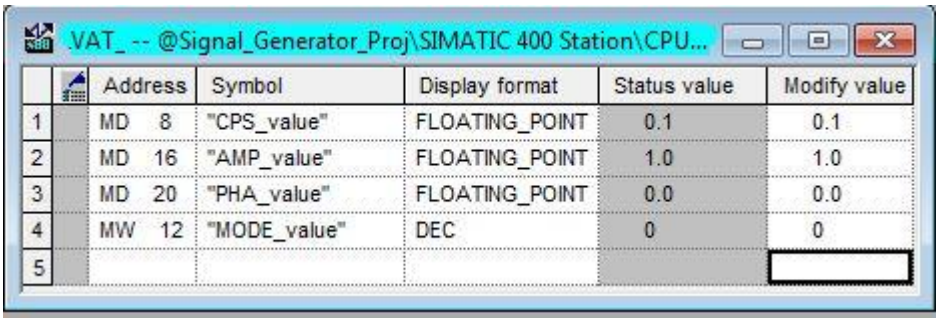
Tablica varijabli upotrebljava se za pravljenje liste svih varijabli čije je vrijednosti potrebno nadgledati ili mijenjati. Na slici 4.1. vidljiva je tablica varijabli „VAT_“ rabljena u projektu. U njoj su zapisane četiri varijable: „CPS_value“, „AMP_value“, „PHA_value“ i „MODE_value“. Tablica se sastoji od pet stupaca. Stupac „Address“ predstavlja apsolutnu adresu varijable. U stupcu „Symbol“ nalaze se simbolička imena varijabli, tj. identifikatori. Stupac „Display format“ sadrži format u kojem želimo prikazati sadržaj varijable. Stupac „Status value“ predstavlja sadržaj varijable prilikom zadnjeg ažuriranja. U stupac „Modify value“ unosi se nova vrijednost u varijablu.



Sl. 4.1. *Tablica varijabli*

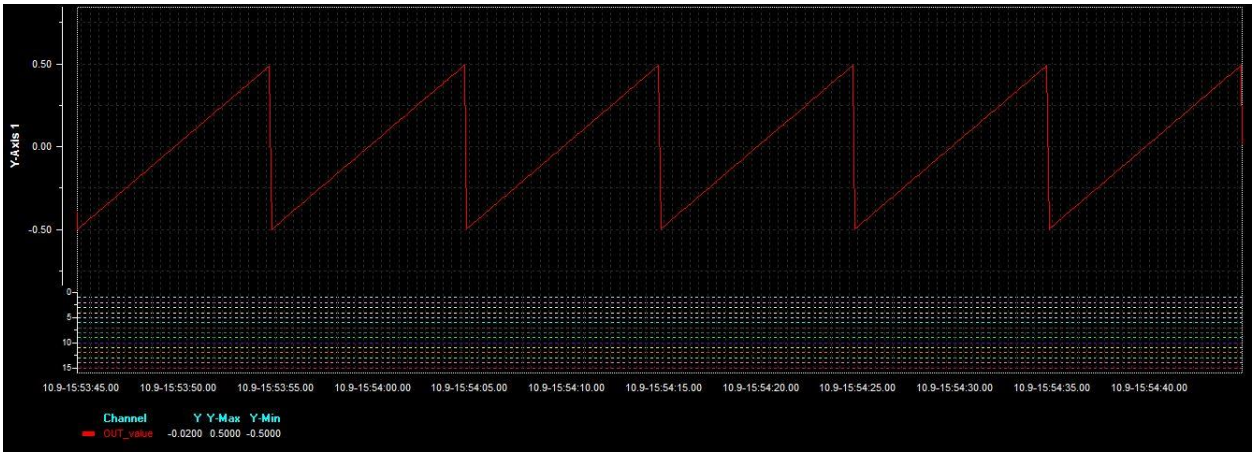
4.1.2. Pilasti signal

Kada su blokovi premješteni u PLC, te se tablica varijabli nalazi u mrežnom načinu izvođenja, tada je vidljiv trenutni sadržaj varijabli i postoji mogućnost mijenjanja njihove vrijednosti. Na slici 4.2. prikazane su nove unesene vrijednosti varijabli. U „CPS_value“ unesena je vrijednost od 0,1, vrijednost varijable „AMP_value“ iznosi 1,0, varijabla „PHA_value“ ima vrijednost 0, jednako kao i „MODE_value“.



Sl. 4.2. *Tablica varijabli s vrijednostima za pilasti signal*

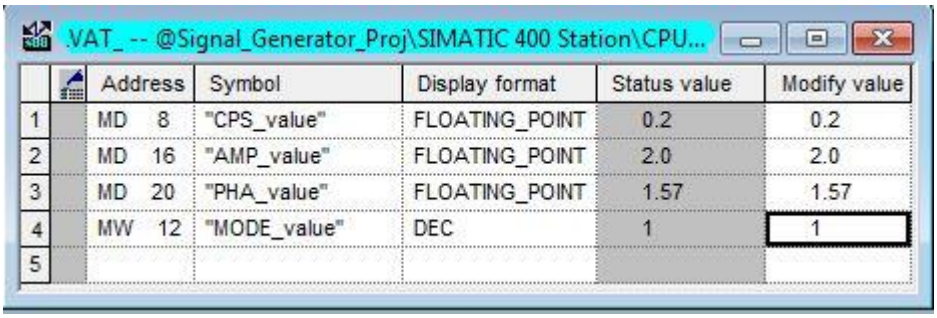
Za navedene vrijednosti, u „S7TrendValues“ iscrtava se graf koji predstavlja pilasti signal frekvencije 0,1 Hertz i amplitude s kraja na kraj u vrijednosti 1,0. Fazni pomak rabi se samo kod sinusnog signala. Graf je prikazan na slici 4.3..



Sl. 4.3. *Pilasti signal*

4.1.3. Sinusni signal

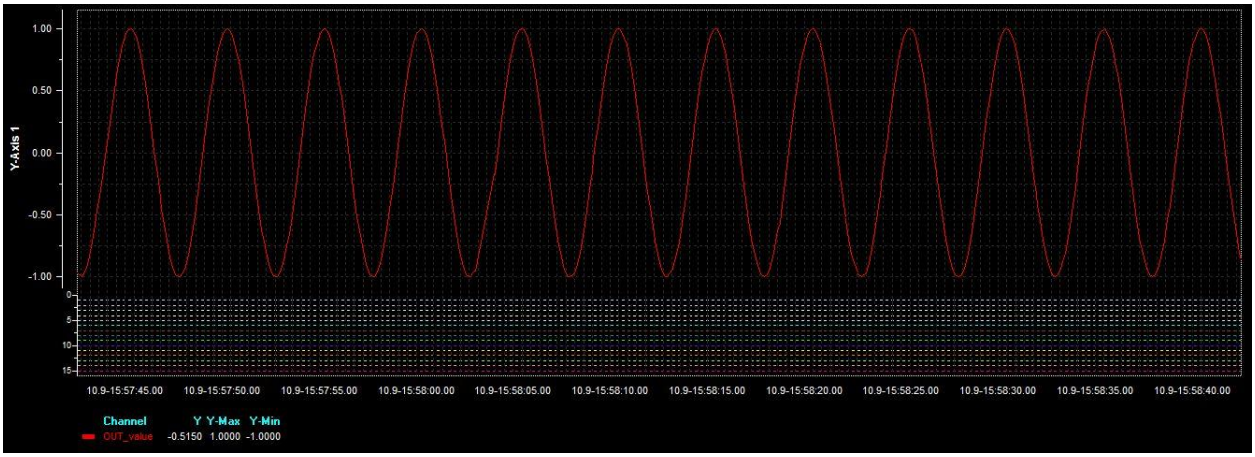
Na slici 4.4. vidljiva je tablica varijabli sa sljedećim vrijednostima: „CPS_value“ ima vrijednost 0,2, vrijednost varijable „AMP_value“ iznosi 2,0, varijabla „PHA_value“ ima vrijednost 1,57, a „MODE_value“ 1.



		Address	Symbol	Display format	Status value	Modify value
1		MD 8	"CPS_value"	FLOATING_POINT	0.2	0.2
2		MD 16	"AMP_value"	FLOATING_POINT	2.0	2.0
3		MD 20	"PHA_value"	FLOATING_POINT	1.57	1.57
4		MW 12	"MODE_value"	DEC	1	1
5						

Sl. 4.4. *Tablica varijabli s vrijednostima za sinusni signal*

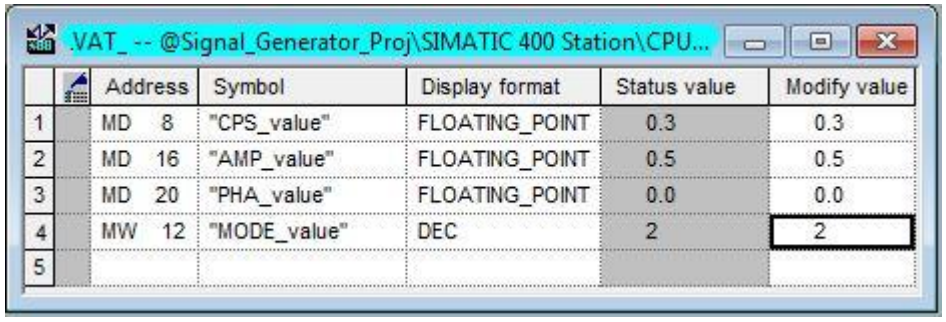
Uz navedene vrijednosti, u „S7TrendValues“ iscrtava se graf koji predstavlja sinusni signal frekvencije 0,2 Hertza, amplitude s kraja na kraj u vrijednosti 2, te pomaka u fazi $\pi/2$ radijana, što je vidljivo na slici 4.5..



Sl. 4.5. *Sinusni signal*

4.1.4. Kvadratni signal

Tablica varijabli, s vrijednostima varijabli koje iznose 0,3 za „CPS_value“, 0,5 za „AMP_value“, 0 za „PHA_value“, te 2 za „MODE_value“, nalazi se na slici 4.6..



	Address	Symbol	Display format	Status value	Modify value
1	MD 8	"CPS_value"	FLOATING_POINT	0.3	0.3
2	MD 16	"AMP_value"	FLOATING_POINT	0.5	0.5
3	MD 20	"PHA_value"	FLOATING_POINT	0.0	0.0
4	MW 12	"MODE_value"	DEC	2	2
5					

Sl. 4.6. *Tablica varijabli s vrijednostima za kvadratni signal*

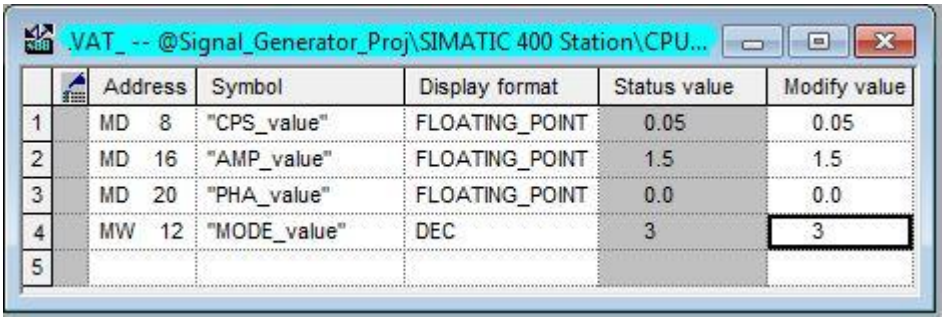
U „S7TrendValues“ za navedene vrijednosti varijabli crta se graf kvadratnog signala frekvencije 0,3 Hertza, te amplitude s kraja na kraj u iznosu od 0,5. To je vidljivo na slici 4.7..



Sl. 4.7. *Kvadratni signal*

4.1.5. Trokutasti signal

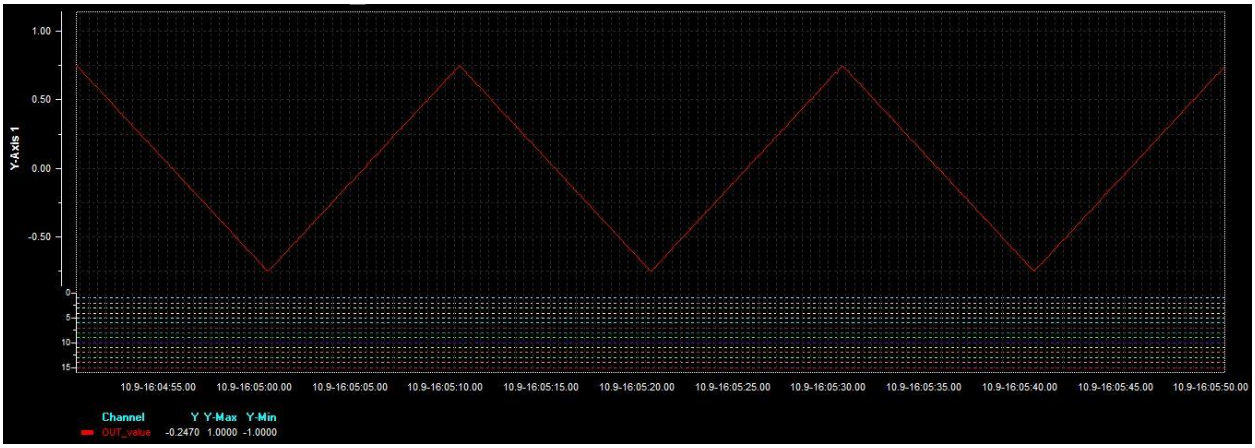
Tablica varijabli, na slici 4.8., sadrži varijable sa sljedećim vrijednostima: „CPS_value“ ima vrijednost 0,05, „AMP_value“ sadrži vrijednost 1,5, „PHA_value“ jest 0 i „MODE_value“ je 3.



	Address	Symbol	Display format	Status value	Modify value
1	MD 8	"CPS_value"	FLOATING_POINT	0.05	0.05
2	MD 16	"AMP_value"	FLOATING_POINT	1.5	1.5
3	MD 20	"PHA_value"	FLOATING_POINT	0.0	0.0
4	MW 12	"MODE_value"	DEC	3	3
5					

Sl. 4.8. *Tablica varijabli s vrijednostima za trokutasti signal*

Na slici 4.9. prikazan je graf trokutastog signala frekvencije 0,05 Hertza i amplitude s kraja na kraj vrijednosti 1,5, koji je za navedene vrijednosti iscrtan u „S7TrendValues“.



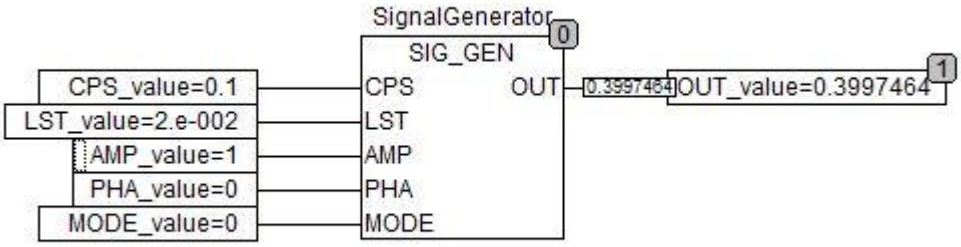
Sl. 4.9. *Trokutasti signal*

4.2. Simulacija i vizualizacija u programskom alatu CoDeSys

Unos vrijednosti parametara, simulacija i vizualizacija mogući su i u programskom alatu „CoDeSys“. Najprije u odredišnim postavkama navede se željeni PLC simulator, u ovom slučaju upotrebljen je „CODESYS SP PLCWinNT“. Nakon pokretanja simulatora, program se premjesti na simulator. Sada je moguće direktno nadzirati i mijenjati vrijednosti ulaznih parametara „CPS_value“, „LST_value“, „AMP_value“, „PHA_value“ i „MODE_value“ iz programa „PLC_PRG“. Za vizualizaciju vrijednosti globalne varijable „OUT_value“ upotrebljavaju se ugrađeni uređivači „Sampling Trace“, ili „Visualisation“. Parametar „LST“ ima predefiniranu vrijednost od 1 milisekunde.

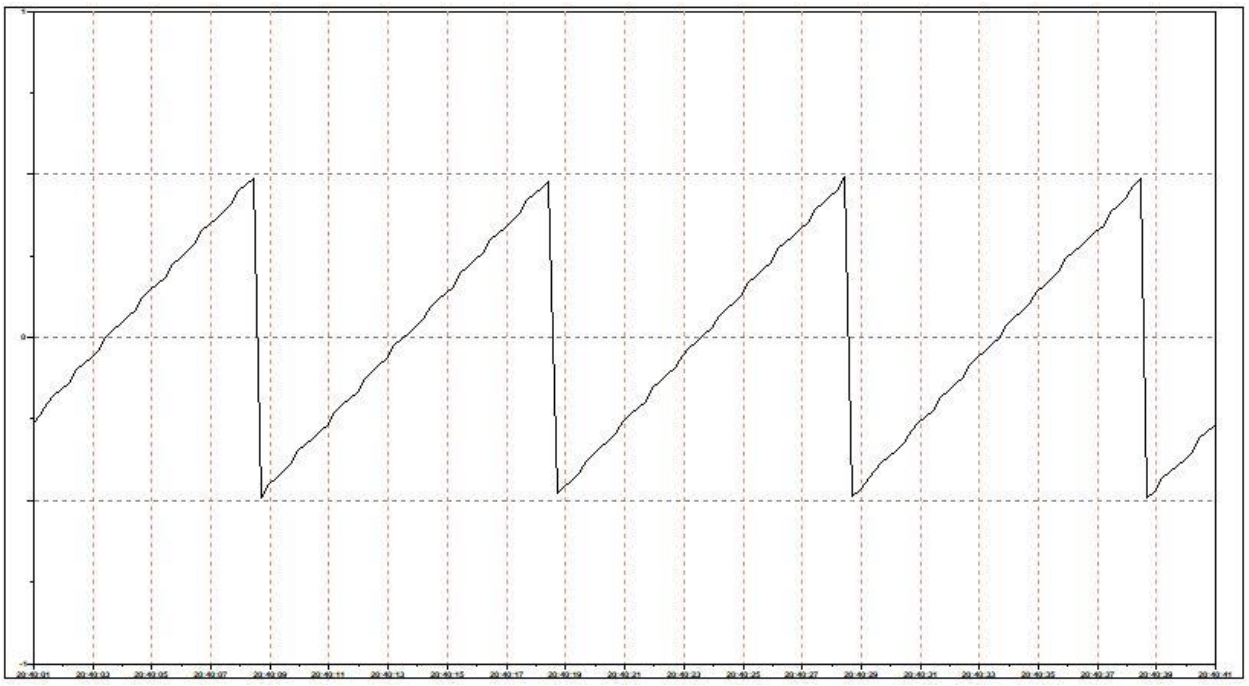
4.2.1. Pilasti signal

Kada je program premješten na PLC, te se nalazi u mrežnom načinu izvođenja, vidljiv je trenutni sadržaj varijabli i postoji mogućnost mijenjanja njihove vrijednosti. Na slici 4.10. prikazane su nove unesene vrijednosti varijabli. U „CPS_value“ unesena je vrijednost od 0,1, vrijednost varijable „AMP_value“ iznosi 1,0, varijabla „PHA_value“ ima vrijednost 0, jednako kao i „MODE_value“.



Sl. 4.10. *Stvarne vrijednosti parametara za dobivanje pilastog signala*

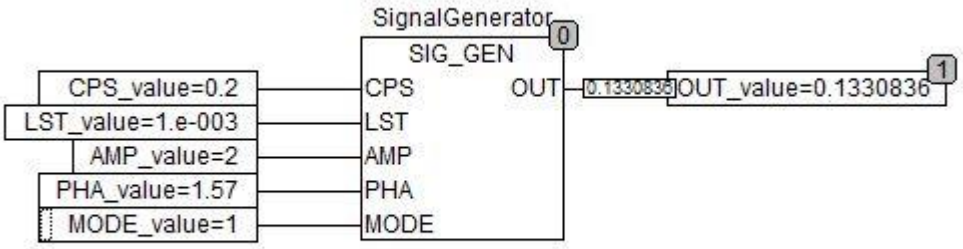
Za navedene vrijednosti, u „Visualisation“ iscrtava se graf koji predstavlja pilasti signal frekvencije 0,1 Hertz i amplitude s kraja na kraj u vrijednosti 1,0. Fazni pomak upotrebljava se samo kod sinusnog signala. Graf možemo vidjeti na slici 4.11..



Sl. 4.11. *Pilasti signal*

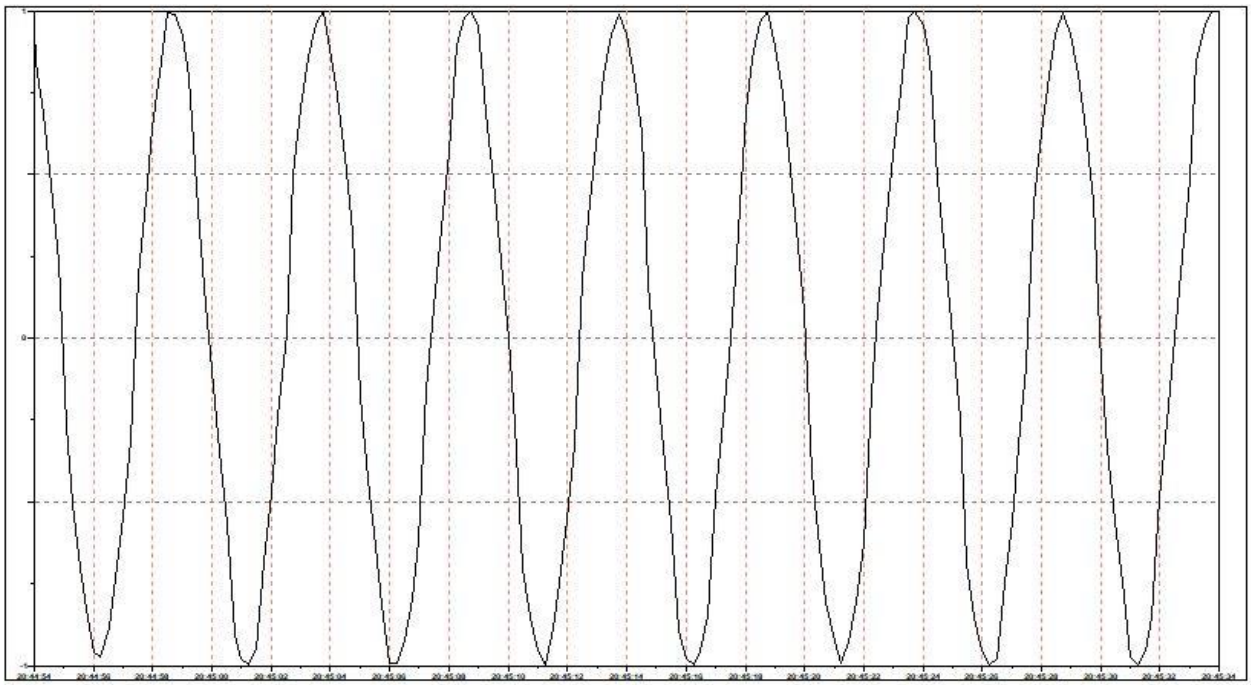
4.2.2. Sinusni signal

Na slici 4.12. prikazan je „PLC_PRG“ sa sljedećim vrijednostima: „CPS_value“ ima vrijednost 0,2, vrijednost varijable „AMP_value“ iznosi 2,0, varijabla „PHA_value“ ima vrijednost 1,57, a „MODE_value“ 1.



Sl. 4.12. *Stvarne vrijednosti parametara za dobivanje sinusnog signala*

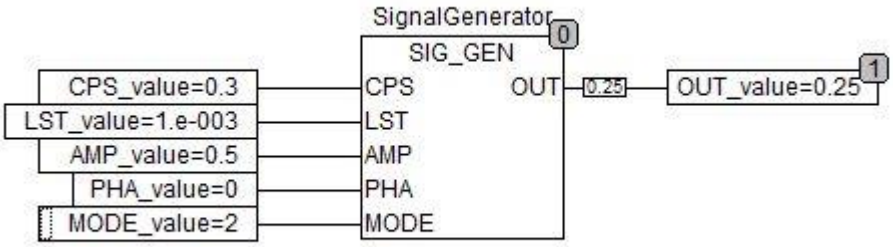
Uz navedene vrijednosti, u „Visualisation“ iscrtava se graf koji predstavlja sinusni signal frekvencije 0,2 Hertza, amplitude s kraja na kraj u vrijednosti 2, te pomaka u fazi $\pi/2$ radijana, što je vidljivo na slici 4.13..



Sl. 4.13. *Sinusni signal*

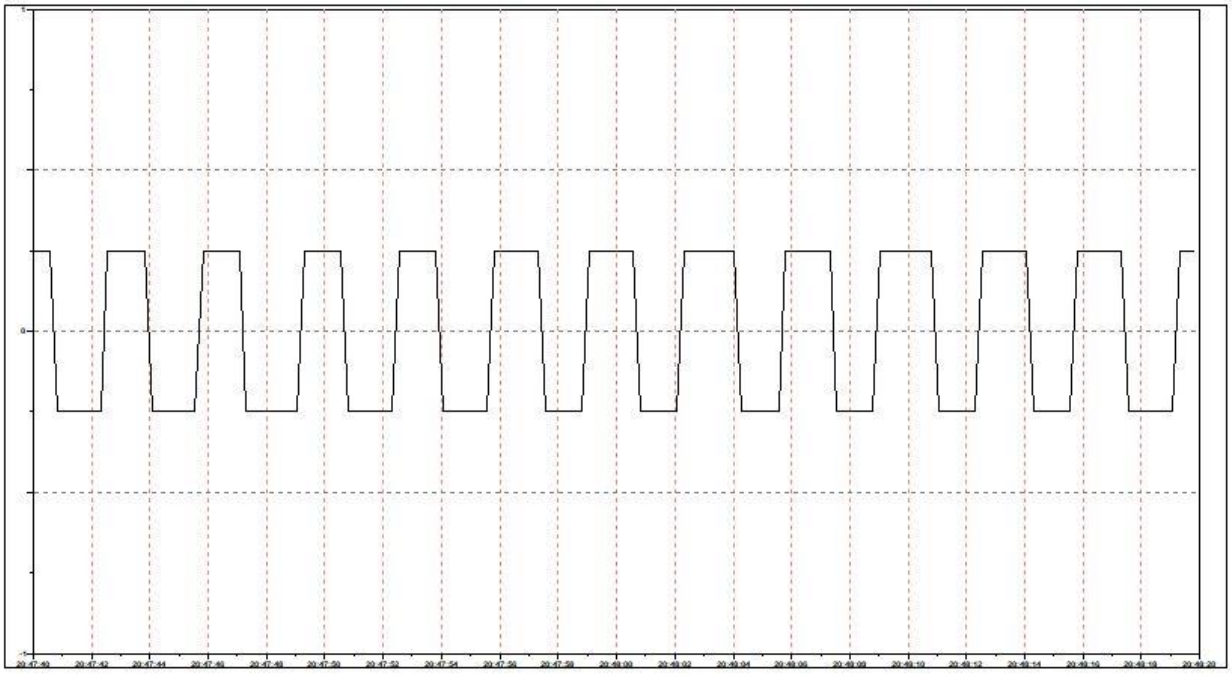
4.2.3. Kvadratni signal

PL_PRG s vrijednostima varijabli koje iznose 0,3 za „CPS_value“, 0,5 za „AMP_value“, 0 za „PHA_value“, te 2 za „MODE_value“, nalazi se na slici 4.14..



Sl. 4.14. *Stvarne vrijednosti parametara za dobivanje kvadratnog signala*

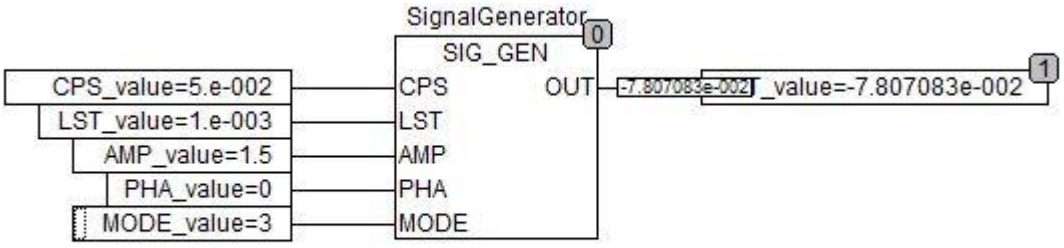
U „Visualisation“ za navedene vrijednosti varijabli crta se graf kvadratnog signala frekvencije 0,3 Hertza, te amplitude s kraja na kraj u iznosu od 0,5. To je vidljivo na slici 4.15..



Sl. 4.15. *Kvadratni signal*

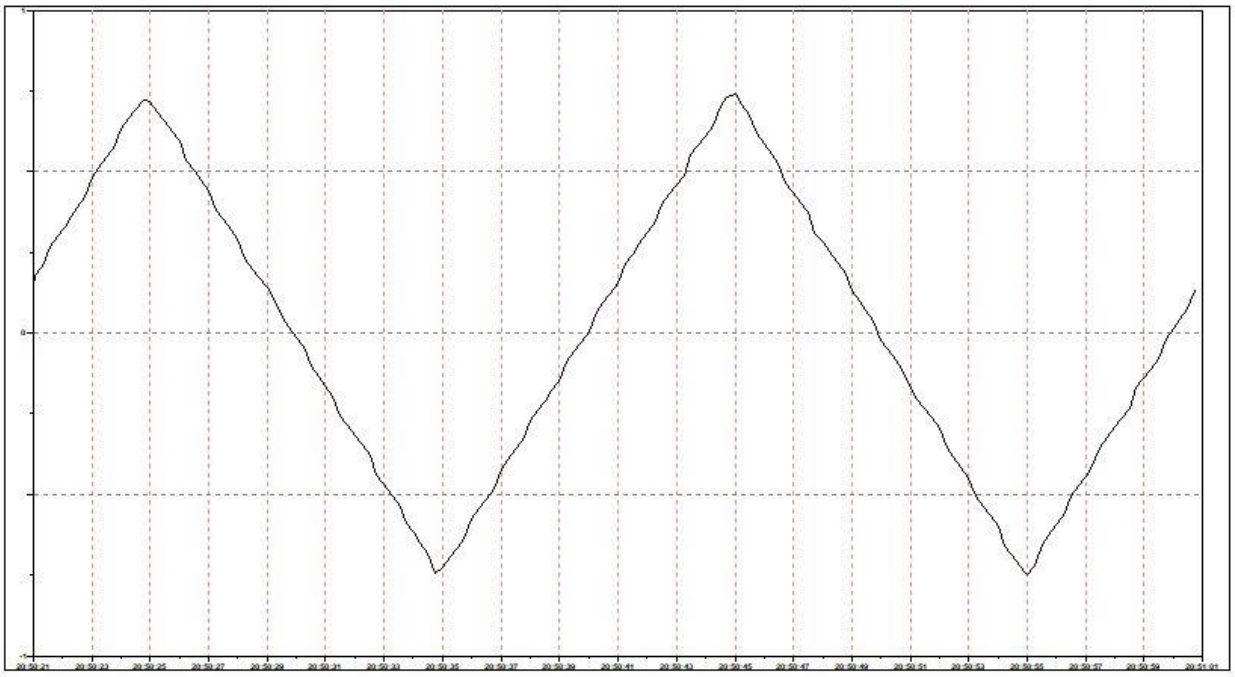
4.2.4. Trokutasti signal

„PLC_PRG“, na slici 4.16., sadrži varijable sa sljedećim vrijednostima: „CPS_value“ ima vrijednost 0,05, „AMP_value“ sadrži vrijednost 1,5, „PHA_value“ jest 0 i „MODE_value“ je 3.



Sl. 4.16. *Stvarne vrijednosti parametara za dobivanje trokutastog signala*

Na slici 4.17. prikazan je graf trokutastog signala frekvencije 0,05 Hertza i amplitude s kraja na kraj vrijednosti 1,5, koji je za navedene vrijednosti iscrtan u „Visualisation“.



Sl. 4.17. *Trokutasti signal*

5. ZAKLJUČAK

Zadatak završnog rada bio je izrada programskog bloka za generiranje referentnog signala promjenjive amplitude i frekvencije. Programski blok omogućuje definiranje signala oblika pile, sinusa, kvadrata i trokuta. Zadatak je izveden u programskim alatima „Step 7“ i „CoDeSys“.

Programski blok u programskom alatu „Step 7“ napisan je u LAD programskom jeziku. Taj programski jezik pogodan je za primjenu korisnicima koji su navikli raditi sa strujnim krugovima. Program je mogao uspješno biti napisan i u drugim programskim jezicima. Programski blok izveden je na način da se iz organizacijskog bloka „OB35“, koji omogućava cikličko izvođenje bloka u fiksnim intervalima, vrši poziv funkcijskog bloka „SIG_GEN“ sa zadanim parametrima. Na temelju parametra „MODE“, iz programskog bloka vrši se poziv jedne od četiriju definiranih funkcija za izračunavanje vrijednosti funkcije signala. Cijeli se program mogao napisati i unutar funkcijskog bloka, bez upotrebe funkcija, no ovaj je način realizacije jednostavniji za snalaženje. Programski je blok na sličan način izveden i u programskom alatu „CoDeSys“. Razlika je u tome što je u „CoDeSys“-u upotrebljen CFC programski jezik jer omogućava veći broj naredbi, tj. blokova. Jedna od tih naredbi jest „EN/ENO“ naredba koja ne postoji u ostalim programskim jezicima koje podržava „CoDeSys“, a bez koje bi izvedba zadatka bila otežana.

Za vizualizaciju rezultata dobivenih u programskom alatu „Step 7“, bilo je potrebno instalirati dodatni alat „S7TrendValues“, te „NetToPLCsim“ za povezivanje sa simulatorom. Programski alat „CoDeSys“ tu nudi ugrađene uređivače „Sampling Trace“ i „Visualisation“, bez potrebe skidanja dodatnih programa. Još jedna prednost „CoDeSys“-a u tome je što se vrijednosti varijabli u mrežnom načinu izvođenja mogu direktno unositi iz programa, dok je u „Step 7“ u tu svrhu potrebno napraviti tablicu varijabli, ili podatkovni blok.

Kada bi se, uz odgovarajuću sklopovsku konfiguraciju, program razvijen u „Step 7“, ili „CoDeSys“ programskom alatu premjestio na PLC, taj uređaj vršio bi funkciju generatora signala. Generator signala elektronički je uređaj koji ima sposobnost generiranja valnih funkcija različitih oblika, faze i amplitude. Najčešće se rabe u testiranju, rješavanju problema i popravljanju elektroničkih uređaja. Pilasti signali primjenjuju se u glazbi, te u radu s pikselima na CRT ekranima. AC (engl. *Alternationg Current*) motori, frižideri, televizori, radio uređaji i sl., upotrebljavaju sinusne signale. Kvadratni signali mogu se rabiti za predstavljanje digitalnog signala, te u elektroničkim i mikroelektroničkim krugovima kao signali za kontrolu satova i brojača. Testiranje pojačala vrši se trokutastim signalima.

Kao što se vidi iz navedenog, široke su mogućnosti primjene generatora signala. No, PLC-ovi služe za izvođenje najrazličitijih upravljačkih funkcija unutar postrojenja. U tu svrhu široko se upotrebljavaju programski alati „Step 7“ i „CoDeSys“, sa svojim prednostima i nedostacima, ali podjednako učinkovito.

LITERATURA

[1] Z. Vukić, Lj.Kuljača, Automatsko upravljanje, Kigen d.o.o., Zagreb, 2005.

[2] H. Berger, Automating with SIMATIC, Publicis Corporate Publishing, Erlangen, 2006.

[3] G. Belusic, T. Licul, V. Persic, Training center MODULE A1, Daniely-Systec d.o.o., Labin, 2007.

[4] User Manual for PLC Programming with CoDeSys 2.3, 3S – Smart Software Solutions GmBH, Kempten, 2010.

[5] CoDeSys Visualisation, Supplement to the User Manual for PLC Programming with CoDeSys 2.3, 3S – Smart Software Solutions GmBH, Kempten, 2008.

[6] Signal Generator (SGEN), (https://www.techopedia.com/definition/12336/signal-generator-sgen), pristup ostvaren 03.09.2018.

[7] What are Sawtooth Waves?, (http://www.innovateus.net/science/what-are-sawtooth-waves), pristup ostvaren 03.09.2018.

[8] Electrical Waveforms, (https://www.electronics-tutorials.ws/waveforms/waveforms.html), pristup ostvaren 03.09.2018.

[9] T. Agarwal, All Eletronics Waveform Generators with Applications, (https://www.efxkits.co.uk/different-types-of-waveform-generators-with-applications/), pristup ostvaren 03.09.2018.

SAŽETAK

Naslov: Generator signala različitih oblika i frekvencija

Zadatak završnog rada bio je izrada programskog bloka za generiranje referentnog signala promjenjive amplitude i frekvencije. Programski blok omogućuje definiranje signala oblika pile, kvadrata, trokuta i sinusa, te u slučaju sinusnog signala definiranje faznog pomaka signala. Razvoj programa, simulacija i vizualizacija odvijaju se u alatima „Step 7 i CoDeSys“. To su programski alati koji služe za programiranje PLC-ova (engl. *Programmable Logic Controllers*), računalnih uređaja koji služe za upravljanje automatskim procesima unutar postrojenja. Zadatak je u „Step 7“ izveden na način da se unutar „OB35“ (engl. *Organization Block*), koji omogućava cikličko izvođenje bloka u fiksnim intervalima, vrši poziv FB-a (engl. *Function Block*) naziva „SIG_GEN“, sa zadanim parametrima. Unutar funkcijskog bloka, na temelju ulaznog parametra „MODE“, vrši se poziv jednog od četiri FC-a (engl. *Function*). Funkcije, na temelju matematičkih operacija, izračunavaju vrijednosti pilastog, sinusnog, kvadratnog, ili trokutastog signala, te se te vrijednosti rekurzivno ispisuju u izlaznu globalnu varijablu „OUT_value“. Iscrtane vrijednosti pojedinih signala prikazane su u programu „S7TrendValues“. Pisanje programa obavljeno je u programskom jeziku LAD (engl. *Ladder Diagram*). Program je na sličan način izveden i u programskom alatu CoDeSys, s tim da se poziv funkcijskog bloka „SIG_GEN“ i predaja stvarnih parametara izvode u programu „PLC_PRG“, a umjesto funkcija, za izračunavanje vrijednosti signala također se upotrebljavaju funkcijski blokovi. Vizualizacija moguća je u ugrađenim alatima „Sampling Trace“ i „Visualisation“, dok je program napisan u programskom jeziku CFC (engl. *The Continuous Function Chart Editor*).

Ključne riječi: CoDeSys, upravljanje, PLC, signal, Step 7

ABSTRACT

Title: Generator of signals of different shapes and frequencies

The task of this bachelor thesis was to develop a program block for generating a reference signal of variable amplitude and frequency. The program block provides the possibility of defining four different signal types, namely, a saw-tooth wave, sine wave, square wave and triangle wave. A sine wave enables defining of a phase shift. Program development, simulation and visualization are done using “Step 7” and “CoDeSys” software tools. These software tools are used for programming of PLCs (Programmable Logic Controllers), computer devices that control automation processes inside of a plant. The task in “Step 7” is carried out in following way. Inside of “OB35” (Organization Block), which provides cyclic execution of the block in fixed intervals, a call of FB (Function Block), named “SIG_GEN”, is carried out with given parameters. The function block, based on the input parameter “MODE”, makes a call of one of four FCs (Function). Functions, aided by mathematical operations, calculate values of a saw-tooth wave, sine wave, square wave, or triangle wave, after which these values are recursively written in the output global variable “OUT_value”. The drawn values of individual signals are shown in “S7TrendValues” software. The program is written in LAD (Ladder Diagram) programming language. In a similar way, the program is developed using “CoDeSys” programming tool. The difference is in calling the function block “SIG_GEN” and assigning of real parameters being done using “PLC_PRG” program. The other difference is that function blocks are used for calculating signal values instead of functions. Visualization is available in the embedded tools called “Sampling Trace” and “Visualization”, while the program is written in CFC (Continuous Function Chart Editor) programming language.

Key words: CoDeSys, control, PLC, signal, Step 7

ŽIVOTOPIS

Marko Miljković rođen je 17. lipnja 1995. godine u Vinkovcima. U Vinkovcima završava osnovnu školu „Antun Gustav Matoš“, te 2009. godine upisuje Prirodoslovno – matematičku gimnaziju „Matija Antun Reljković“ u Vinkovcima. Godine 2015. ostvaruje upis na Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, smjer računarstvo.