



Upravljački algoritmi u realnom vremenu

Projektni zadatak 6 - Postrojenje za sortiranje

Vranješević Anja, RA 9/2021
Mrđen Minja, RA 2/2021
Vučenović David, RA 126/2021

Profesor: dr Željko Kanović

15. jul 2024.

Sadržaj

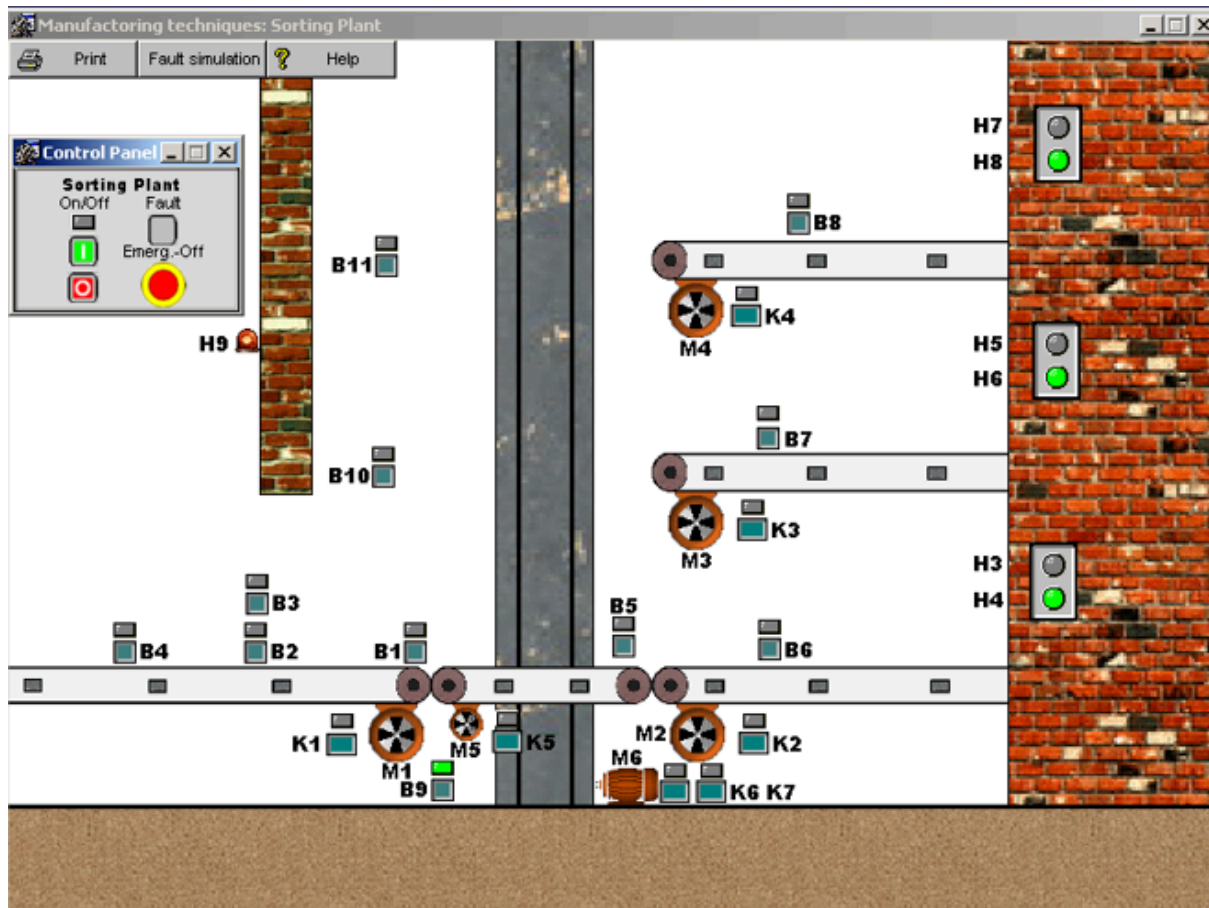
Uvod.....	1
Opis sistema (od čega se sastoji sistem).....	1
Funkcionalnost sistema.....	3
Greške (Control Faults).....	5
Implementacija.....	5
Automat stanja.....	5
Init.....	6
Wait.....	7
Stanje za upravljanje radom osnove (dovodne) trake - K1.....	7
Sledeći paket.....	8
Stanja paketa - P1, P2 i P3.....	9
Stanje za upravljanje radom trake lifta - K5.....	9
Stanja za upravljanje radom lifta - K7 i K6.....	10
Stanja za pokretanje traka za odvođenje paketa - K2, K3 i K4.....	12
Paralelizacija.....	13
Određivanje paketa.....	14
Stop i Emergency stop.....	14
Tajmeri za upravljanje motorima M2, M3 i M4.....	15
SCADA.....	16
Poteškoće prilikom izrade.....	18
Paralelizacija.....	18
Paralelizacija istovremenog rada više motora.....	18
Paralelizacija određivanja paketa i ostatka sistema.....	19
Kašnjenje.....	20
Nailazak dva ista paketa uzastopno.....	21
Tajmeri i Emergency stop.....	22
SCADA.....	23
Zaključak.....	24

Uvod

U ovom odeljku biće reči o samom sistemu, kako funkcioniše, šta je na raspolaganju od ulaza/izlaza i slično.

Opis sistema (od čega se sastoji sistem)

Izgled sistema predstavljen je na sledećoj slici:



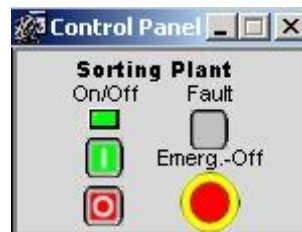
Slika 1: Izgled sistema

Sistem se sastoji od:

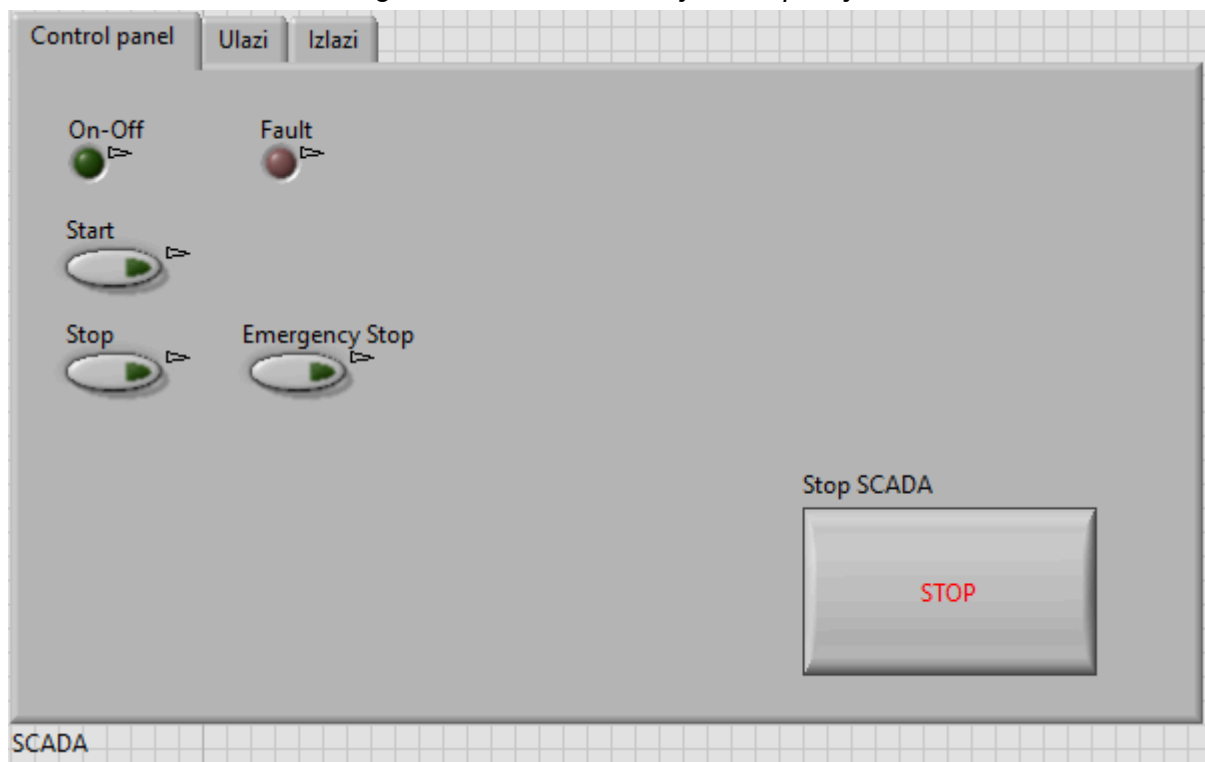
- Senzora - svaki senzor ima svoju namenu
 - Senzori **B4**, **B3** i **B2** određuju veličinu paketa
 - Senzor **B1** govori da li postoji paket koji čeka sortiranje
 - Senzori **B9**, **B10** i **B11** daju informaciju o poziciji lifta (B9 - lift je na prvom spratu, B10 - na drugom i B11 - na trećem)
 - Senzor **B5** daje informaciju o tome da li se paket nalazi na liftu
 - Senzori **B6**, **B7** i **B8** daju informaciju o tome da li se paket nalazi na nekom od spratova
- Kontrola- svaka kontrola kontroliše odgovarajući motor pokretne trake ili lifta.
 - Kontrola **K1** kontroliše motor pokretne trake **M1** kojim paketi dolaze na sortiranje
 - Kontrola **K5** kontroliše motor pokretne trake **M5** kojim paketi dolaze na lift

- Kontrole **K6** i **K7** kontrolišu rad motora **M6** kojim lift ide gore odnosno dole. Istovremeno kontrole K6 i K7 **ne mogu** biti uključene
- Kontrole **K2**, **K3** i **K4** upravljaju motorima **M2**, **M3** i **M4** koji pokreću trake za prvi, drugi i treći sprat
- Indikatori - indikatori govore da li je određena traka sprata zauzeta ili slobodna i da li je aktiviran *Emergency Off*. Takođe ovde spadaju indikatori greške (*Fault*) i indikator *On/Off*.
 - Indikatori **H3** i **H4** daju informaciju o zauzetosti trake prvog sprata. Ukoliko je indikator H3 aktivan - traka je zauzeta a ukoliko je indikator H4 aktiviran - traka je slobodna.
 - Indikatori **H5** i **H6** i indikatori **H7** i **H8** u parovima, imaju istu funkcionalnost kao i par H3 i H4 samo što su namenjene trakama za drugi odnosno za treći sprat
 - Indikator **H9** jeste alarm koji se aktivira prilikom aktiviranja *Emergency Off* tastera

Sistemom se upravlja preko tastera *Start*, *Stop* i *Emergency Off* iz Control Panel-a ili preko SCADA.




Slika 2: Izgled Control Panel-a kojim se upravlja sistem



Slika 3: SCADA u kojoj se takođe može upravljati radom sistema

Funkcionalnost sistema

Glavna funkcionalnost sistema kojim se upravlja jeste da nadolazeće pakete sortira na tri sprata. Svaki sprat odgovara jednom tipu paketa.

Sistem se pokreće na taster  (*Start*) čime se pokreće traka koja dovodi pakete u sistem za sortiranje, odnosno pokreće se motor M1 kontrolom K1. Takođe, indikator *On/Off* postaje aktivan. Kada paket dolazi on prolazi senzore B4, B3 i B2. Pomoću tih senzora određuje se veličina paketa tako što se na rasuđoj ivici B2 provere stanja B3 i B4 senzora. Kombinacije koje mogu da se dese su sledeće:

- Paket P1 (nizak i širok paket) - aktivira senzore B2 i B4



Slika 3: Izgled paketa 1

- Paket P2 (visok a uzak paket) - aktivira senzore B2 i B3



Slika 4: Izgled paketa 2

- Paket P3 (visok i širok paket) - aktivira senzore B2, B3 i B4



Slika 5: Izgled paketa 3



Kada jedan od paketa aktivira senzor B1 svojim prolaskom prvo se proverava da li je senzor B9 takođe aktivan jer ako jeste, paket može da ide na lift (u ovom slučaju se aktivira motor M5 kontrolom K5 i paket se prosleđuje na lift sve dok se ne aktivira senzor B5). Ukoliko se lift ne nalazi na prvom spratu, odnosno B9 nije aktivan, traka koja dovodi pakete u sistem za sortiranje se zaustavlja i čeka se lift, odnosno čeka se da B9 postane ponovo aktivan.

Prilikom dolaska paketa na lift, u zavisnosti od tipa paketa lift se diže ili ostaje na prvom spratu (ukoliko je u pitanju paket P1). Ukoliko je potrebno da lift ode na drugi ili na treći sprat, motor M6 se pokreće i podiže lift na odgovarajući sprat. Ako je potrebno da lift dođe na drugi sprat, motor M6 je aktivan sve dok se ne aktivira senzor B10. Isto tako, ako je potrebno da lift dođe na treći sprat, motor M6 će raditi sve dok se ne aktivira senzor B11.

Kada paket dospe na odgovarajući sprat, počinje iskrcavanje paketa sa lifta. Aktivira se motor M5 i odgovarajući motor za pokretnu traku tog sprata (M2, M3 ili M4). Kada se senzor B5 isključi, odnosno kada paket siđe sa lifta, lift počinje da ide na dole a paket koji se upravo iskrcavao sa lifta, pokretnom trakom ide dalje.

Pokretna traka koja odvozi trenutno iskrcani paket je aktivna unapred definisanim vremenom od trideset sekundi nakon čega se isključuje. Tajmer koji broji to vreme se aktivira na opadajuću ivicu senzora B6, B7 ili B8 u zavisnosti o kojem spratu je reč. Indikatori H4 (za prvi sprat), H6 (za drugi sprat) i H8 (za treći sprat) su aktivni ukoliko je odgovarajuća traka za odvođenje paketa slobodna (traka je slobodna ukoliko senzori B6 - za prvi sprat, B7 - za drugi sprat ili B8 - za treći sprat nisu aktivni). Indikatori H3 (za prvi sprat), H5 (za drugi sprat) i H7 (za treći sprat) su aktivni ukoliko je odgovarajuća traka za odvođenje paketa zauzeta.

Ovaj postupak se ponavlja sve dok postoje paketi koji dolaze u sistem za sortiranje.

Sistem se može zaustaviti tasterima  (*Stop*) i  (*Emergency Off*). Ukoliko se sistem zaustavi tasterom *Stop*, sistem će se zaustaviti tek nakon što se završi procesiranje trenutnog paketa i lift dospe na prvi sprat. Da bi sistem nastavio sa radom, potrebno je ponovo pritisnuti taster *Start*.

Ukoliko se sistem zaustavi pritiskom na taster *Emergency Off*, sistem se zaustavlja momentalno bez obzira na to gde se trenutno nalazi odnosno u kojem je stanju trenutno. Ovom prilikom, uključuje se sirena H9 i indikator greške *Fault* kao i indikatori H3, H5 i H7. Da bi se sistem ponovo pokrenuo, tj. da bi sistem nastavio sa radom potrebno je *odglaviti* taster *Emergency Off* i ponovo pritisnuti *Start*. Nakon toga, sistem nastavlja tamo gde je stao.

Greške (Control Faults)

U toku rada sistema mogu se desiti sledeće greške:

- Paket 1 je pogrešno sortiran
- Paket 2 je pogrešno sortiran
- Paket 3 je pogrešno sortiran
- Kontrole K6 i K7 su istovremeno uključene

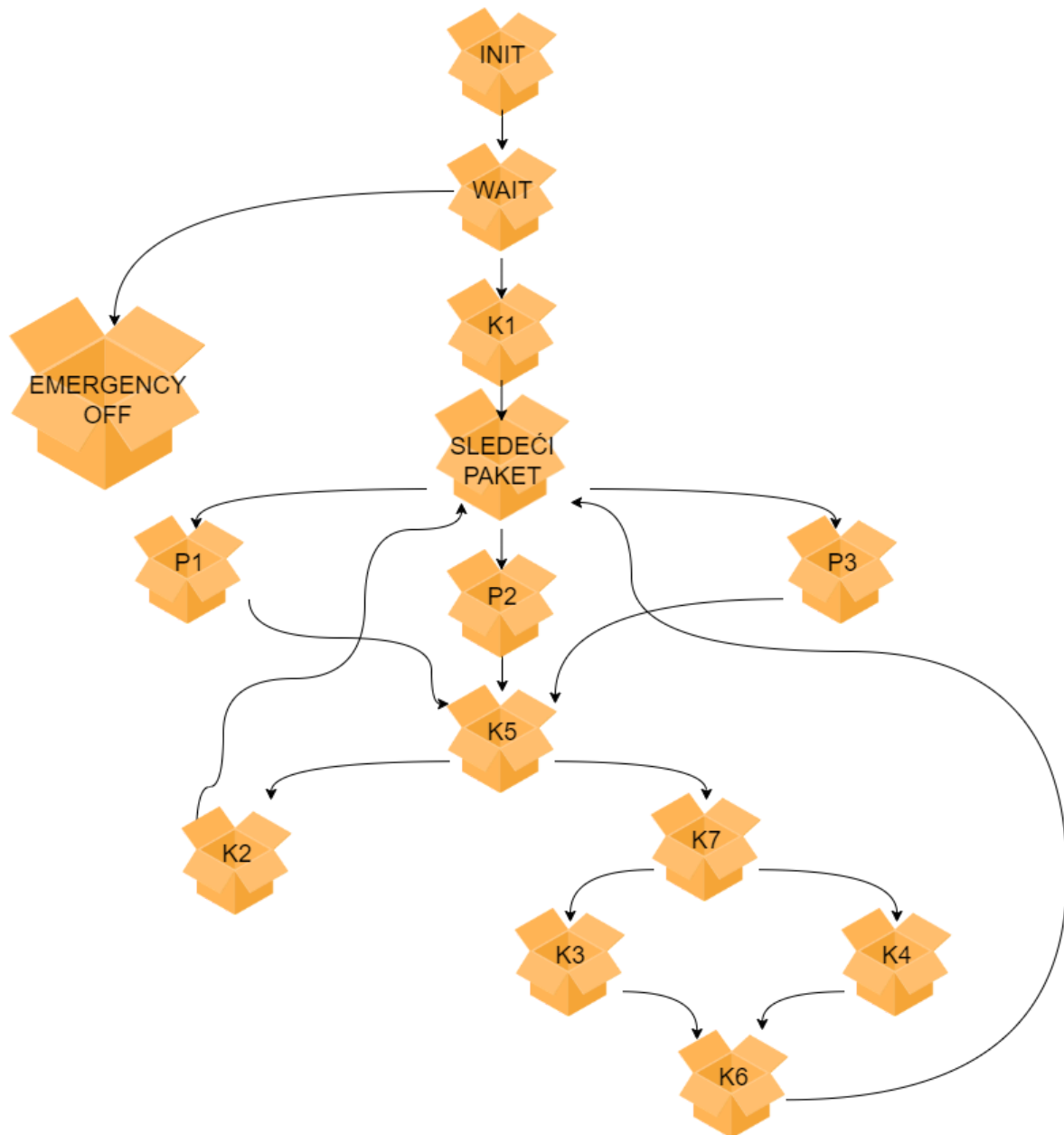
Ako se desi neki od ovih slučajeva, simulacija se automatski zaustavlja. Pri novom pokretanju se kreće iz inicijalne pozicije.

Implementacija

Implementacija je započeta određivanjem stanja onsovnog automata i svih uslova prelaska u svako stanje. Nakon toga smo uvideli da je potrebna paralelizacija pojedinih delova sistema te je iz tog razloga napravljen još jedan automat sa svojim stanjima čija je namena određivanje veličine paketa. Na posletku, ostala je implementacija stanja koji zaustavljaju sistem te implementacija tajmera koji kontrolišu zaustavljanje traka za svaki sprat. SCADA je implementirana tek na samom kraju, kada smo bili sasvim sigurni da sistem funkcioniše.

Automat stanja

Automat se sastoji iz stanja prikazanih na slici:

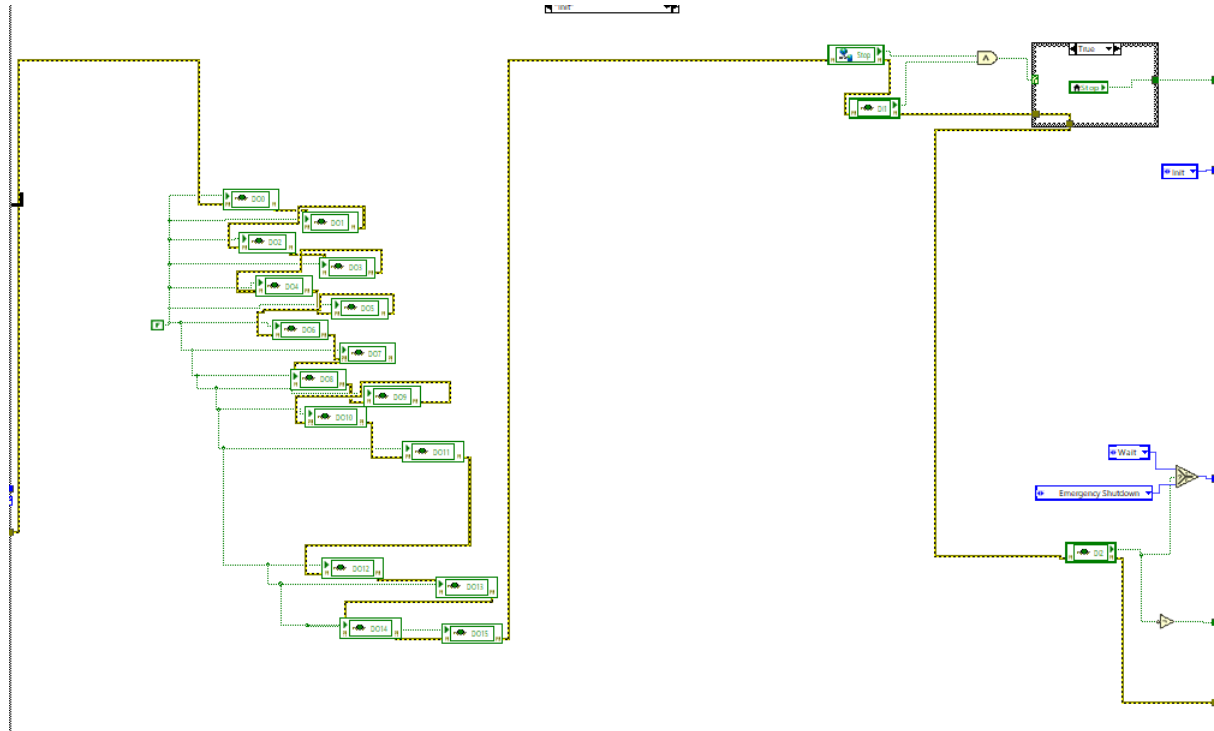


Slika 6: Prikaz stanja i prelaza u svako stanje osnovnog automata

Pre opisa svakog stanja, treba pomenuti da na slici nisu označena dve stvari. Prva je da svako stanje može da uđe ponovo u sebe osim *Init* stanja (ovo se dešava ukoliko nije ispunjen ni jedan uslov za prelazak u neko drugo stanje). Druga stvar je da se iz svakog stanja može dospeti u *Emergency Off*.

Init

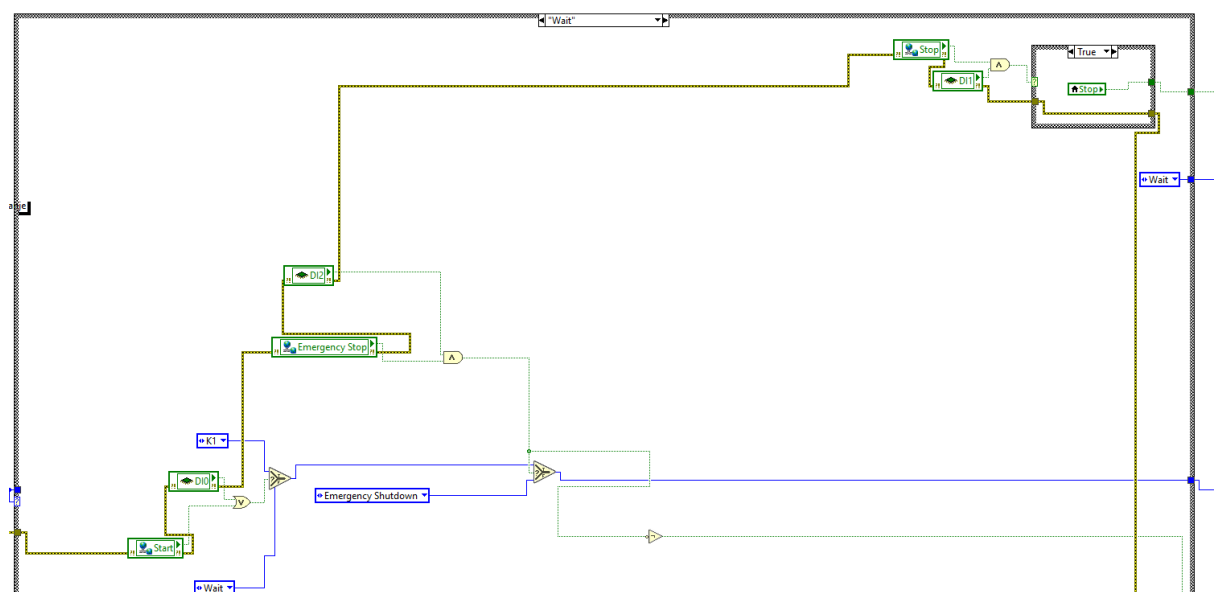
U ovom stanju se inicijalizuju sve promenljive na svoje početne vrednosti. Ovim se obezbedjuje pravilno funkcionisanje sistema u samom početku. Iz stanja *Init* se bezuslovno prelazi u sledeće stanje - *Wait*. Dakle, ovo stanje se izvrši samo jednom, prilikom prvog uključivanja sistema.



Slika 7: Izgled implementacije stanja *Init*

Wait

U stanje *Wait* se jedino može doći iz stanja *Init* i ovo je stanje koje se izvršava sve dok se ne pritisne *Start* dugme iz *Control Panel-a* ili *Scade*. Ukoliko je pritisnuto *Start* dugme sledeće stanje postaje K1.

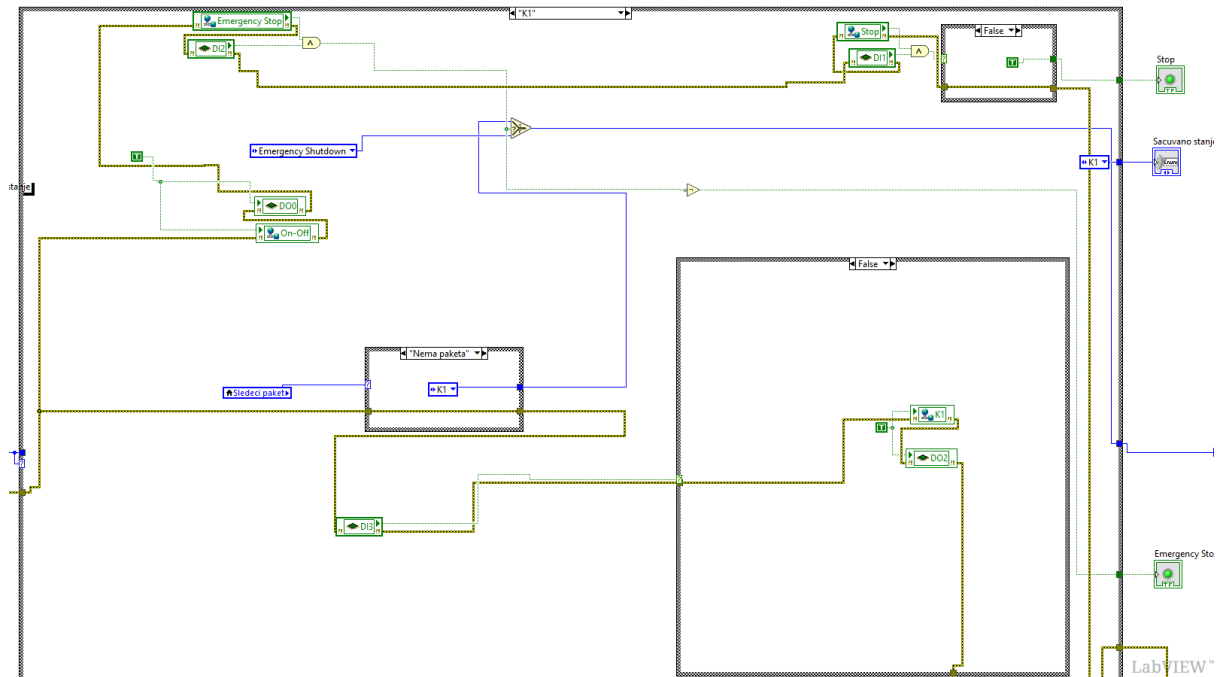


Slika 8: Izgled implemetacije stanja *Wait*

Stanje za upravljanje radom osnove (dovodne) trake - K1

U ovom stanju se pokreće motor M1 kontrolom K1 čime se pokreće osnovna traka koja dovodi pakete u sistem za sortiranje. Ova traka može biti pokrenuta ukoliko paket još nije stigao do lifta, odnosno ako je paket stigao do lifta, lift se nalazi na nivou trake koju pokreće motor M1 i lift je slobodan.

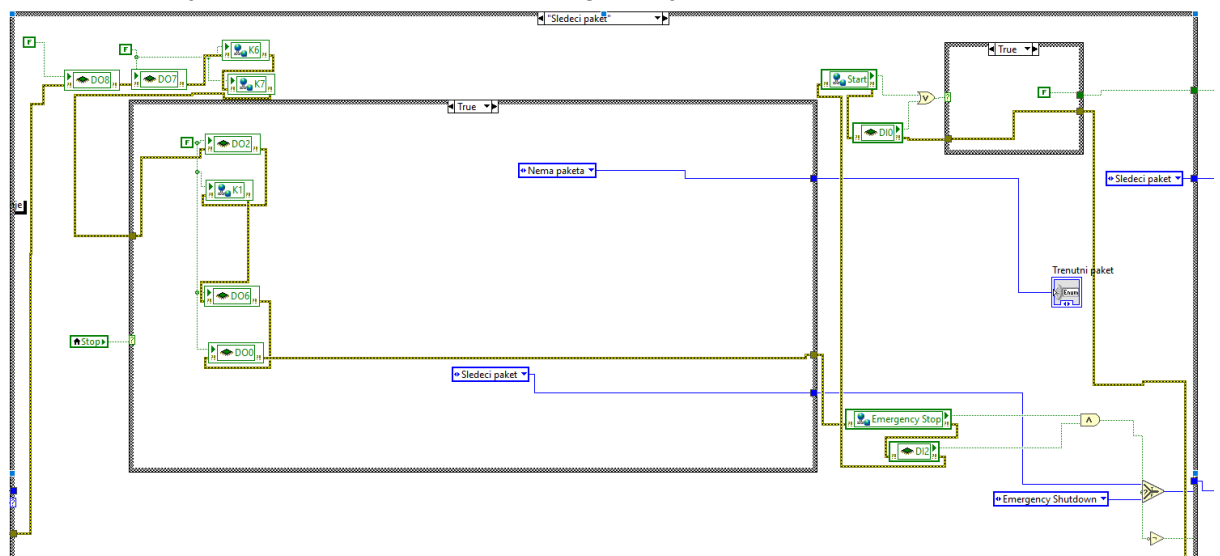
Iz ovog stanja se izlazi kada automat za određivanje veličine paketa detektuje prvi paket, nakon čega se prelazi u stanje *Sledeći paket* o kojem će biti reči u nastavku.



Slika 9: Izgled implementacije stanja K1

Sledeći paket

Stanje *Sledeći paket* "čeka" da se identifikuje paket na osnovnoj traci kako bi se iz njega prešlo u stanje P1, P2 ili P3 u zavisnosti od paketa. Ukoliko nema paketa na dovodnoj traci, stanje ostaje nepromenjeno. U ovo stanje se ponovo ulazi svaki put nakon što se jedan paket obradi, tj. kada se paket odvede na odgovarajući sprat i lift se nalazi na prvom spratu.

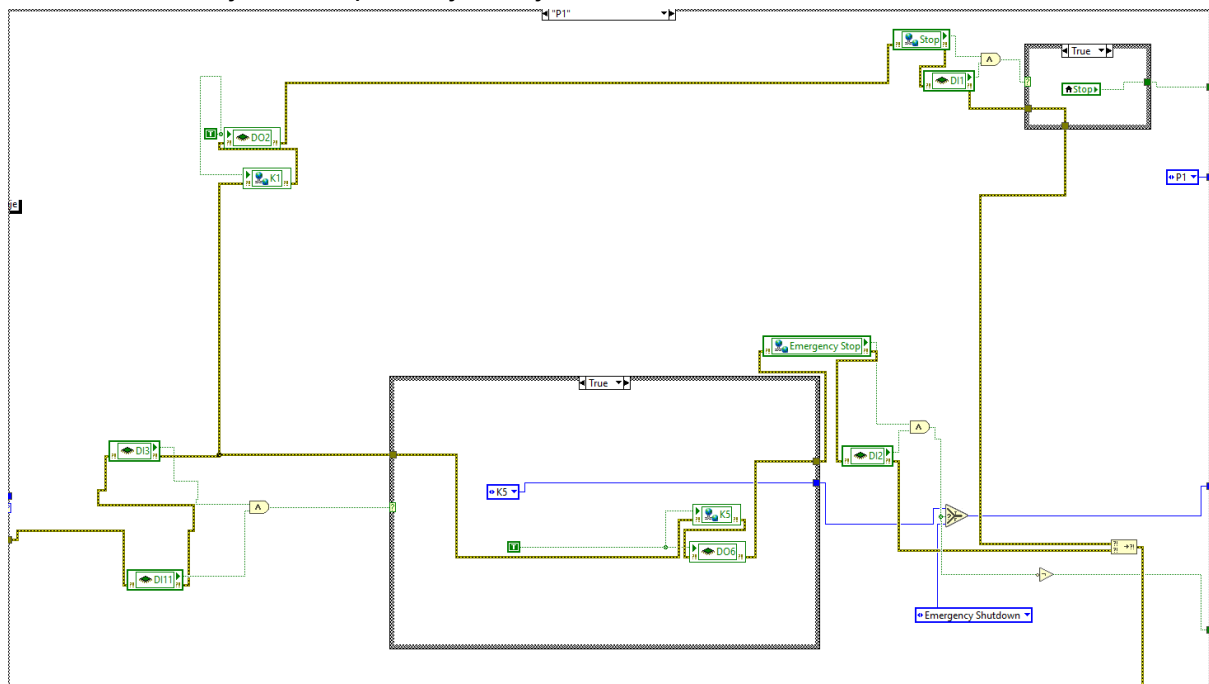


Slika 10: Izgled implementacije stanja *Sledeći paket*

Stanja paketa - P1, P2 i P3

Ova tri stanja služe da ponovo pokrenu motor M1 kako bi paket sišao sa trake, koja dovodi pakete, na traku lifta. Motor M1 se bezuslovno pokreće kada automat uđe u jedno od ova tri stanja jer u ovo stanje može da se uđe samo iz stanja *Sledeći paket*. Uslov za izlazak iz stanja *Sledeći paket*, ukoliko nije pritisnut *Emergency stop*, jeste da je lift na prvom spratu. Pored ovog uslova potrebno je i da je lift slobodan, a to je obezbeđeno s obzirom da je implementirano da dok se lift ne oslobodi automat ne prelazi u stanje *Sledeći paket*. Zbog ovakve strukture automata stanja moguće je bezuslovno pokrenuti spomenutu traku.

Iz ovog stanja se izlazi kada paket koji je u obradi dođe do kraja trake koju pokreće motor M1. Sledeće stanje za sve pakete je stanje K5.

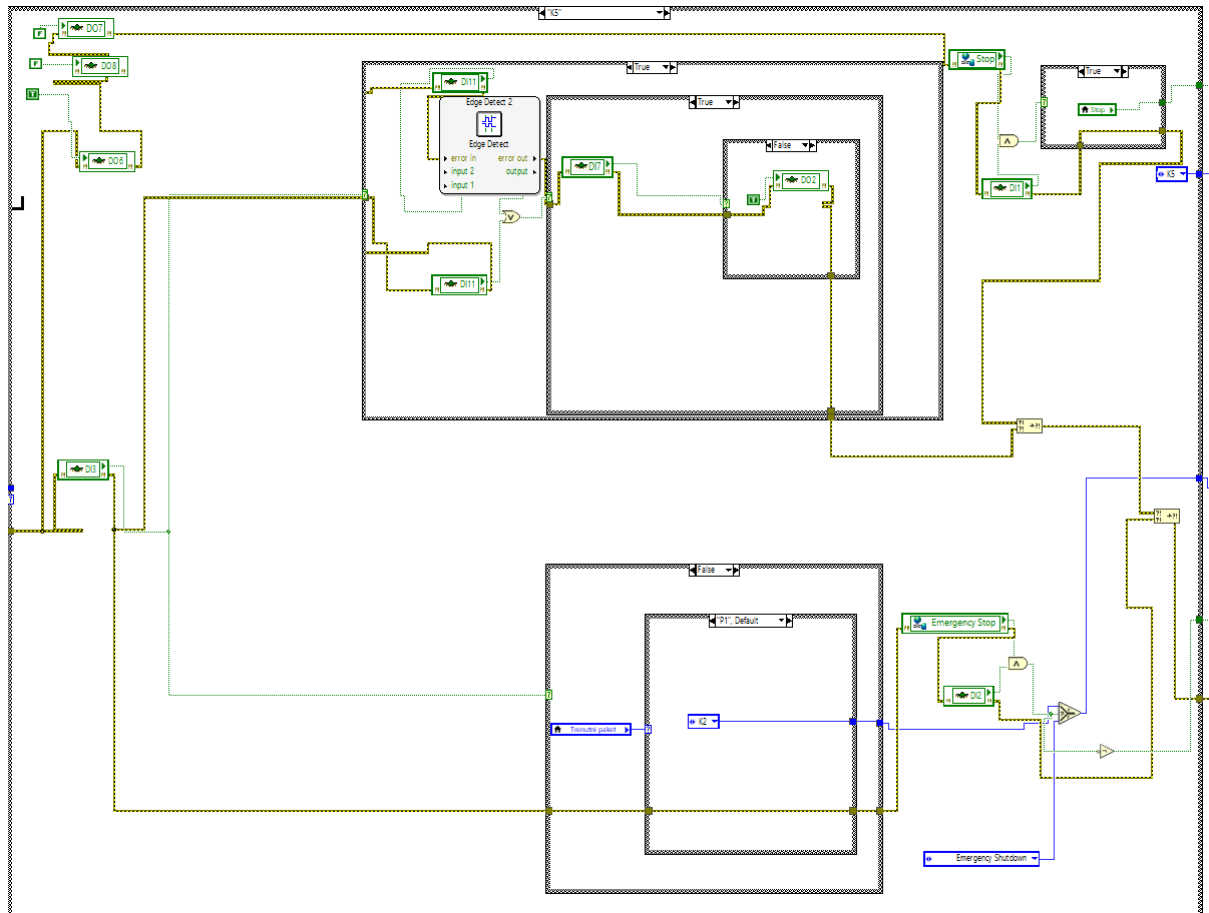


Slika 11: Prikaz stanja P1

Stanje za upravljanje radom trake lifta - K5

U ovom stanju se pokreće motor koji upravlja trakom na liftu, kao i odgovarajući motor u zavisnosti od paketa čija je obrada u toku. Kada paket siđe sa trake koju pokreće motor M1, tj. kada se senzor B1 ugasi, prelazi se u stanje K2, ako je u pitanju paket 1, odnosno stanje K7 ukoliko su u pitanju paketi 2 i 3. Ovakvom implementacijom obezbeđeno je da lift krene ka odgovarajućem spratu čim paket siđe sa trake koja dovodi pakete i time je optimizovano vreme sortiranja paketa 2 i 3.

U stanju K5 je takođe implementirana i provera da li se lift nalazi na prvom spratu. Ukoliko se nalazi proverava se da li je lift slobodan, ako jeste pokreće se traka kojom upravlja motor M1, u suprotnom traka stoji. Ovi uslovi se proveravaju samo ako na traci K1 postoji paket koji čeka da pređe na lift. Ako senzor B1 nije aktivan traka K1 se pokreće bez provere drugih uslova.

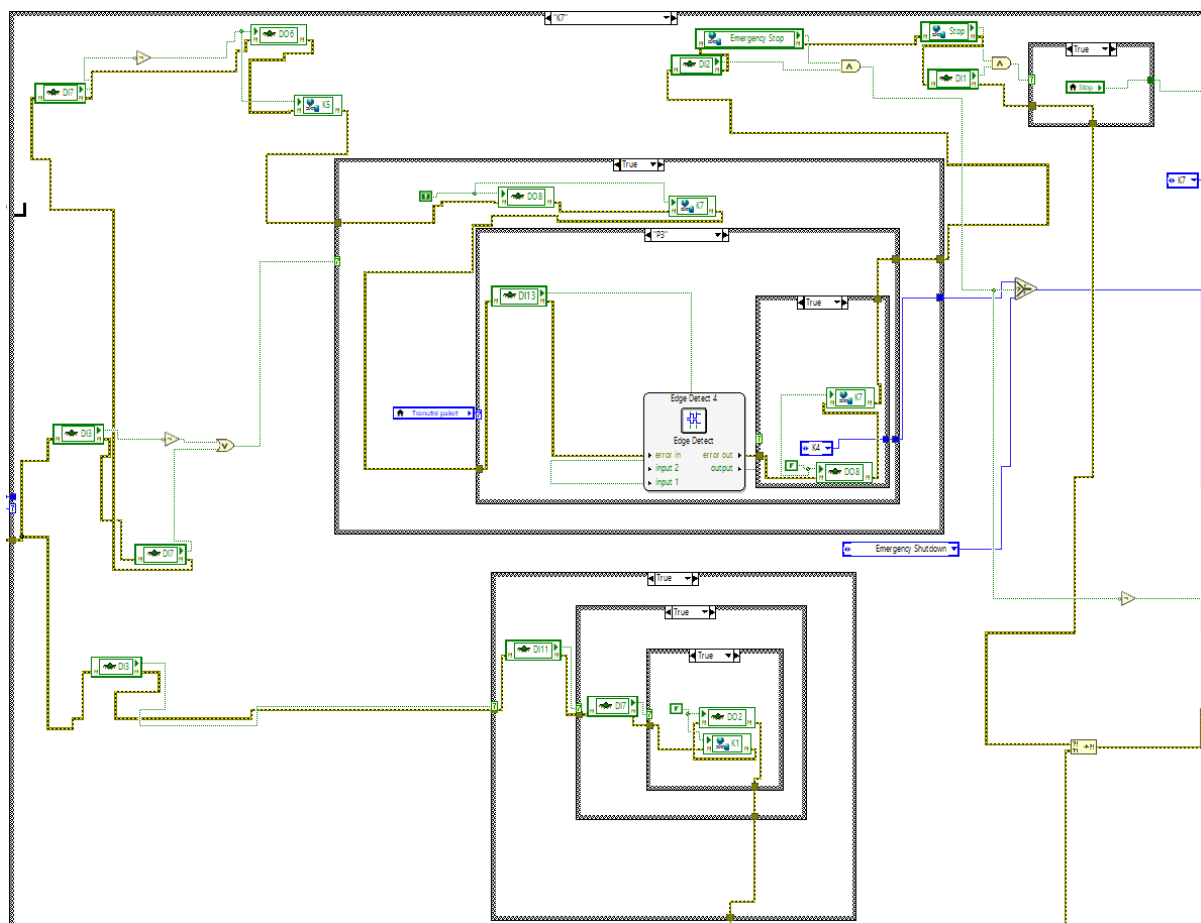


Slika 12: Prikaz stanja K5

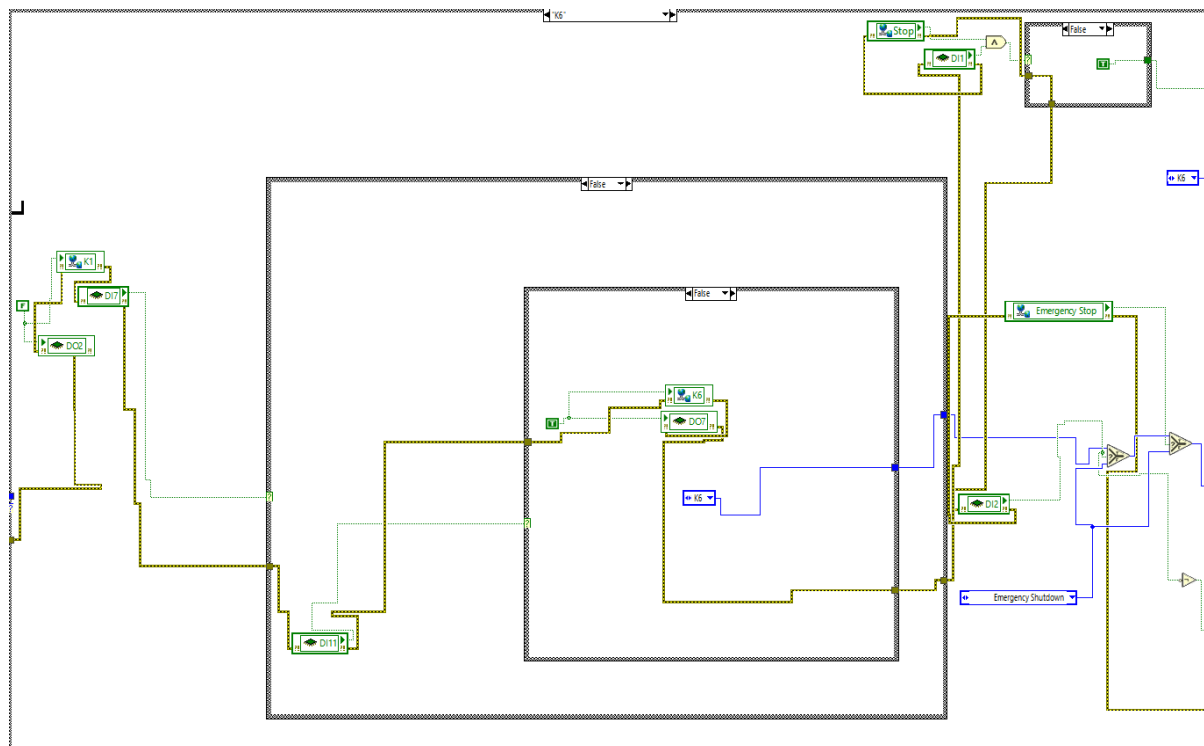
Stanja za upravljanje radom lifta - K7 i K6

U stanju K7 je implementirano da traka lifta K5 radi sve dok paket ne dodje do kraja lifta, tj. dok se senzor B5 ne aktivira. U stanje K7 mogu da uđu isključivo paketi 2 i 3. Kontrola K7 se aktivira ukoliko je paket sišao sa trake K1 ili je došao do kraja lifta. Kada paket 2 dođe do sprata 2, odnosno lift aktivira senzor B10, na kontrolu K7 se šalje false i time se zaustavlja lift. Pri istim uslovima se i izlazi iz stanja K7 i prelazi u stanje K3. Isti princip važi i za paket 3, samo se na aktivaciju senzora B11 prelazi u stanje K4. Takođe se proverava da li je sledeći paket aktivirao senzor B1, u onom trenutku kada ga aktivira zaustavlja se traka K1 ako lift nije na prvom spratu ili nije slobodan.

Stanje K6 je zaduženo za spuštanje lifta na prvi sprat. U ovo stanje se prelazi iz stanja K3 ili K4. Lift se spušta sve dok se ne aktivira senzor B9, nakon čega se prelazi u stanje *Sledeći paket*.



Slika 13: Stanje K7



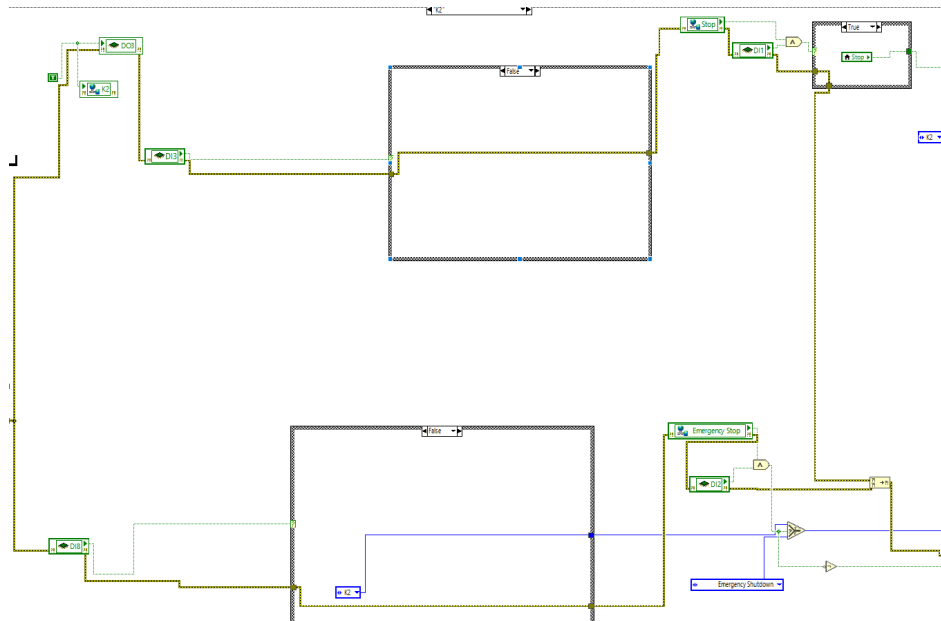
Slika 14: Stanje K6

Stanja za pokretanje traka za odvođenje paketa - K2, K3 i K4

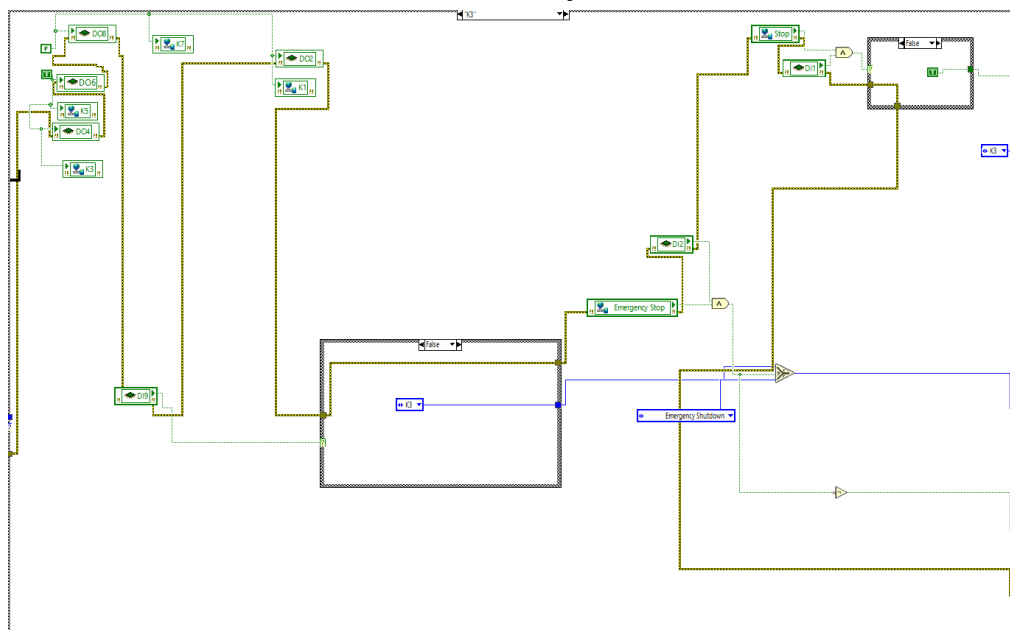
Stanja K2, K3 i K4 kontrolišu rad motora M2, M3 i M4 redom.

U stanje K2 se prelazi direktno iz stanja K5 jer nema potrebe za podizanjem lifta. Stanje K2 aktivira motor M2. Nakon što paket 1 prođe senzor B6 prelazi se u stanje *Sledeći paket*. Paralelno sa ovim, na silaznu ivicu senzora B6 aktivira se tajmer za prvi sprat koji odbrojava 30s, nakon što prođe tih 30s motor M2 se gasi, ovo će biti detaljnije objašnjeno u petlji sa tajmerima.

U stanja K3 i K4 se prelazi iz stanja K7. Ova stanja funkcionišu po istom principu kao i stanje K2. Razlikuju se po tome što se u sledeće stanje iz K3 stanja prelazi na opadajuću ivicu senzora B7, a iz K4 na opadajuću ivicu B8. Nakon ova dva stanja se prelazi u stanje K6. Tajmeri funkcionišu isto kao i za stanje K2.



Slika 15: Stanje K2



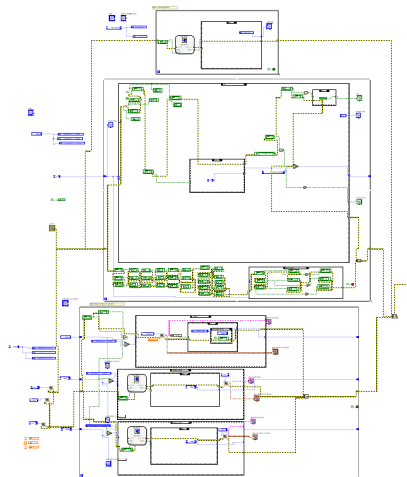
Slika 16: Stanje K3 (Slična implementacija je i za stanje K4)

Paralelizacija

S obzirom da novi paketi pristižu u sistemu i dok sistem obrađuje paket, neophodno je paralelno sa obradom paketa proveravati koji paketi stižu i sledeći su za obradu. Ovo je implementirano tako što se paralelno sa glavnom while petljom izvršava pomoćna while petlja koja sve vreme proverava da li paket stiže i koji paket je u pitanju.

Takođe je bilo potrebno implementirati da trake na spratovima rade određeno vreme, ali tako da sistem to vreme ne stoji i ne čeka, nego da dalje sortira pakete. Ovo je implementirano pomoću druge pomoćne while petlje koja paralelno sa glavnom petljom i petljom za određivanje paketa odbrojava potreban broj sekundi.

Više o ove dve pomoćne petlje opisano je u nastavku dokumentacije.

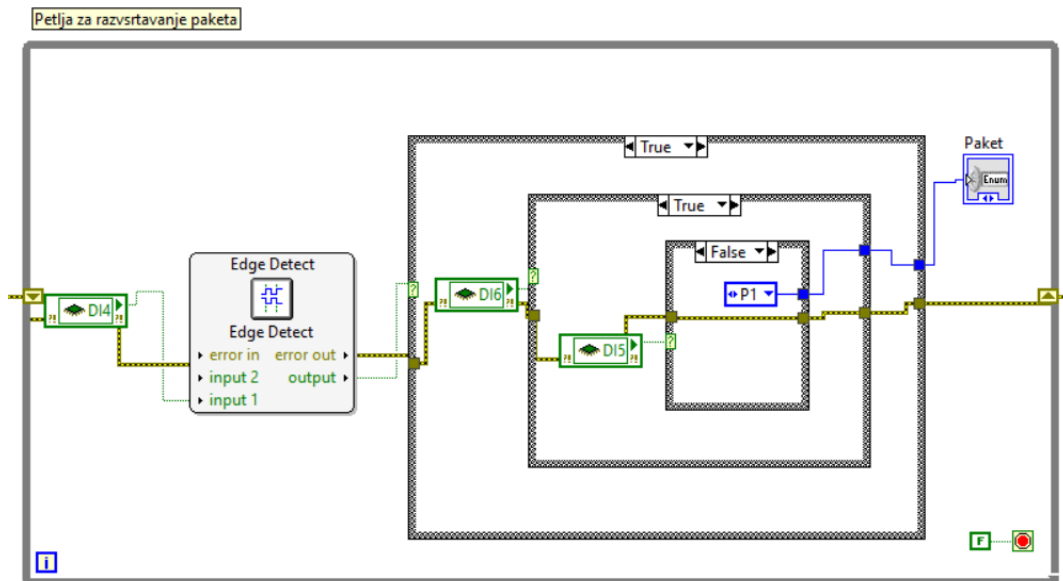


Slika 17: Tri paralelne petlje se izvršavaju istovremeno - prva za određivanje paketa, druga - glavna petlja i treća za tajmere

Određivanje paketa

Implementirana je posebna while petlja koja je zadužena da registruje pakete koji dolaze i da prepoznaje o kom paketu je reč. Za detekciju paketa zadužen je senzor B2. Na uzlaznu ivicu senzora B2 petlja zna da je paket došao i ulazi u case strukturu zaduženu za proveru o kom paketu je reč. Prvo se proverava dužina paketa tako što se proverava da li je senzor B4 istovremeno aktivan kad i senzor B2, ako jeste reč je o dugačkom paketu, a ako nije u pitanju je kraći paket, tj. paket 2. Ukoliko se ispostavi da je reč o dužem paketu proverava se visina paketa tako što se proverava da li je senzor B3 aktivan. Ako je senzor B3 aktivan u pitanju je visoki paket, tj. paket 3, a ako nije aktivan u pitanju je paket 1. Informacija o tome koji je paket naišao se čuva u promenljivoj *Paket*. Čim paket prođe kroz stanje glavne petlje *Sledeći paket*, nema potrebe da se informacija o njegovoj veličini dalje čuva, zato je dovoljna jedna promenljiva za čuvanje informacije koji je paket sledeći za obradu.

Ova petlja se izvršava paralelno sa glavnom petljom.



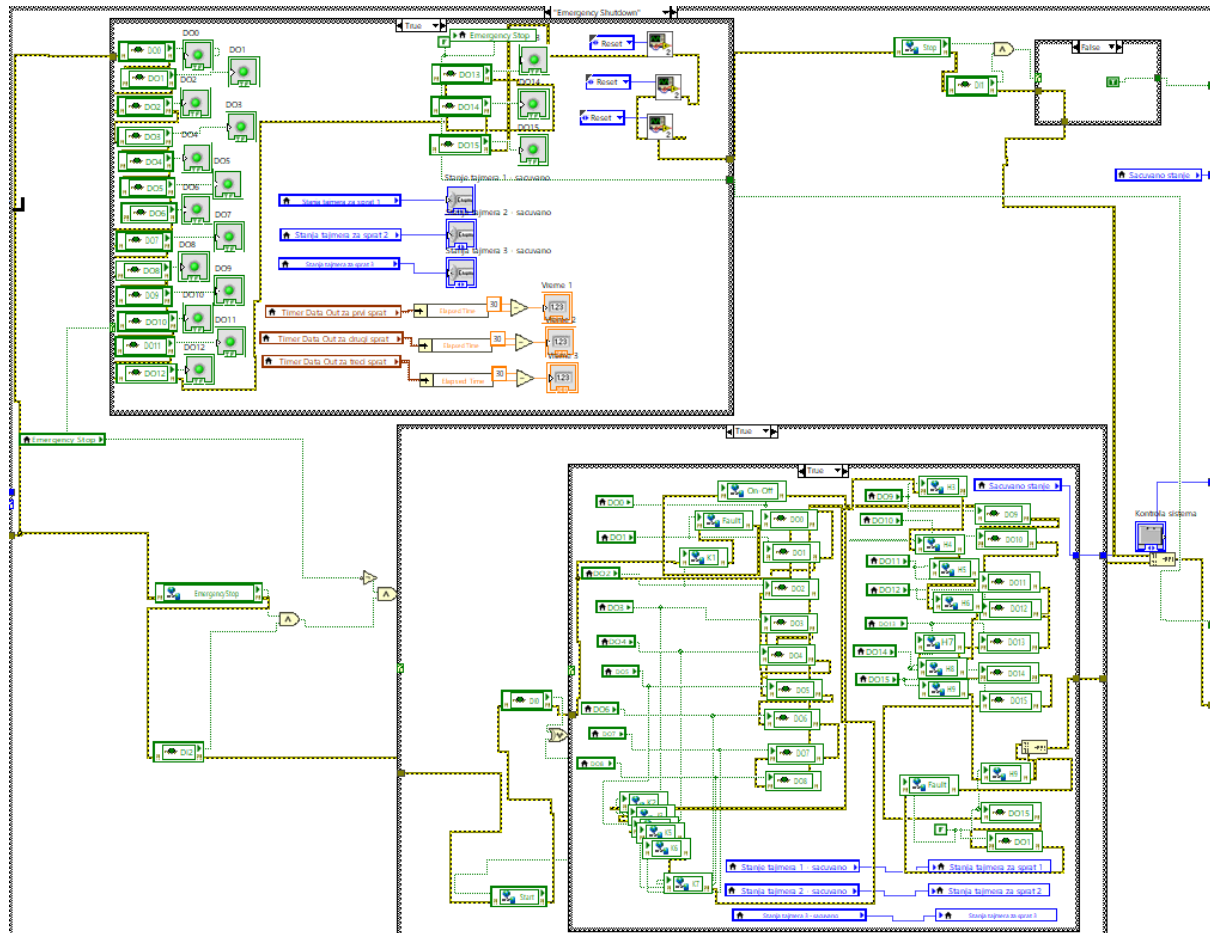
Slika 18: Prikaz petlje za određivanje paketa za sortiranje

Stop i Emergency stop

Stop i Emergency stop su posebne funkcionalnosti koje su implementirane u glavnom automatu stanja.

Stop je implementiran tako da kada se pritisne taster stop sistem prvo dovedemo u stanje *Sledeći paket* i tek ga onda zaustavimo. Nakon pritiskanja tastera za start sistem nastavlja sa radom iz stanja *Sledeći paket*. Ova funkcionalnost je implementirana tako što smo uveli promenljivu System Stop koju smo koristili kao flag. Ova promenljiva je po difoltu False, i njena vrednost može da se postavi na True u bilo kom stanju glavnog automata, ali se vrednost ovog flag-a proverava samo u stanju *Sledeći paket*, jer samo iz tog stanja može da se zaustavi sistem. Pritiskom tastera Start se flag resetuje i sistem nastavlja sa radom.

Emergency stop se može aktivirati u bilo kom stanju i kada se aktivira u trenutku se prelazi u stanje *Emergency shutdown*. Kako bi se izašlo iz ovog stanja neophodno je ponovo pritisnuti taster za Emergency stop, a zatim i taster za start. Iz *Emergency shutdown* stanja se prelazi u ono stanje u kome je pritisnut Emergency stop, ovo je implementirano tako što se u promenljivu *Sačuvano stanje* upisuje stanje u kome se sistem trenutno nalazi, samim tim u trenutku kada se uđe u *Emergency shutdown* stanje u ovu promenljivu je upisano poslednje stanje u kome je bio sistem. Kako bi se stanje u potpunosti sačuvalo, čim se uđe u *Emergency shutdown* stanje sačuvaju se vrednosti svih kontrola, kako bi iste bile prosleđene sistemu pri izlasku iz ovog stanja. Nakon što se sve kontrole sačuvaju iste se postave na False. Kada se ponovo pritisne taster za Emergency stop, a zatim i taster za start sve kontrole se vraćaju na vrednosti pre ulaska u ovo stanje i prelazi se u stanje koje se nalazi u promenljivoj *Sačuvano stanje*.

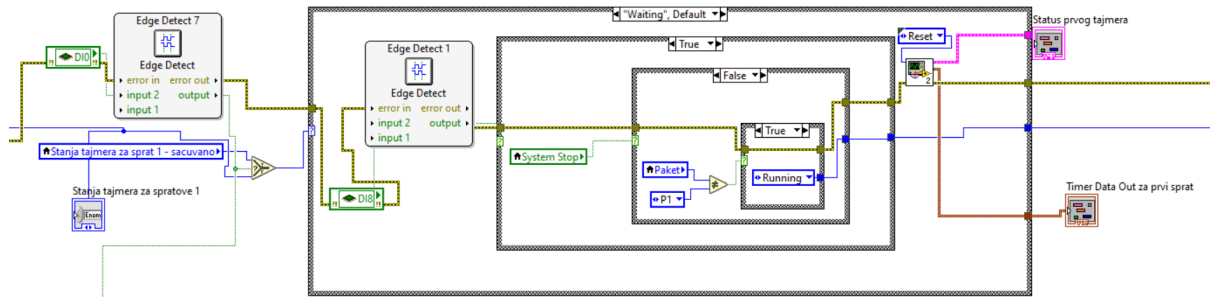


Slika 19: Emergency Stop

Tajmeri za upravljanje motorima M2, M3 i M4

Implementirana je posebna while petlja za kontrolu tajmera. Tajmeri služe da odbrojavaju 30s rada motora M2, M3 i M4. U samoj petlji su tri automata stanja, za svaki sprat poseban automat. Svaki od automata se sastoji iz dva stanja *Running* i *Waiting*. U *Running* stanju se proverava da li je tajmer odbrojao 30s, ako jeste prelazi se u stanje *Waiting*. Ako tajmer nije odbrojao 30s on u svakoj iteraciji proveru u kom stanju se nalazi glavni automat, ako se nalazi u *Emergency shutdown* stanju odgovarajuća traka se gasi i prelazi se u *Waiting*. Za svako drugo stanje glavnog automata se ostaje u *Running* stanju dok tajmer ne prebroji 30s. U svakom tajmer automatu se u stanju *Waiting* kada se ispune uslovi za prelazak u stanje *Running* prvo proveru da li je paket koji čeka na obradu isti kao i paket u obradi. Ako su paketi isti traka počinje sa radom, ali tajmer ne odbrojava vreme, tj. ne ulazi u stanje *Running*. Ovim je postignuto da se vreme odbrojava od poslednjeg paketa koji treba da je doveden na traku kojom tajmer upravlja. Ako je pritisnut stop taster i ispunjeni su uslovi za ulazak u *Running* stanje, u to stanje se ulazi bez obzira na to koji paket sledeći čeka.

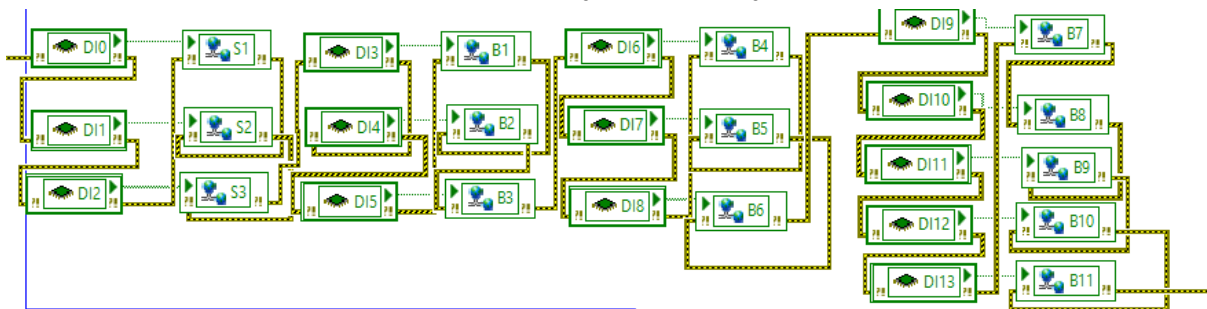
Modifikovali smo *Real time action engine* SubVI tako da kao izlaz daje i klaster *Timer Data* koji ima informaciju o tome koliko je sekundi tajmer izbrojao, tj. *Elapsed time*. Ova informacija je bila značajna pri implementaciji Emergency stop-a, nakon što se sistem ponovo pokrene posle Emergency stopa tajmeri nastavljaju da broje tamo gde su stali kada je pritisnut Emergency stop.



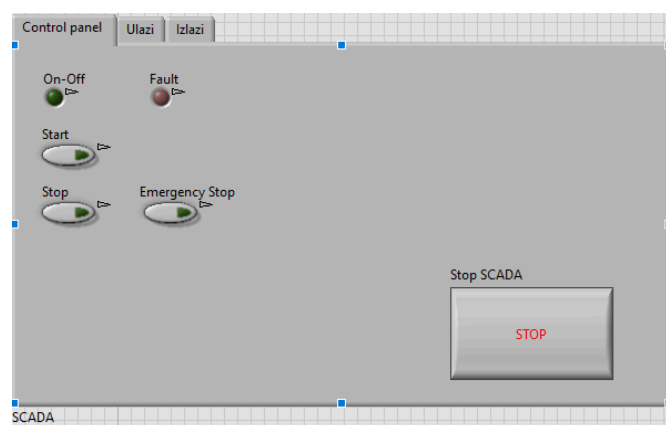
Slika 20: Dio Block Diagram-a sa prikazom rada tajmera

SCADA

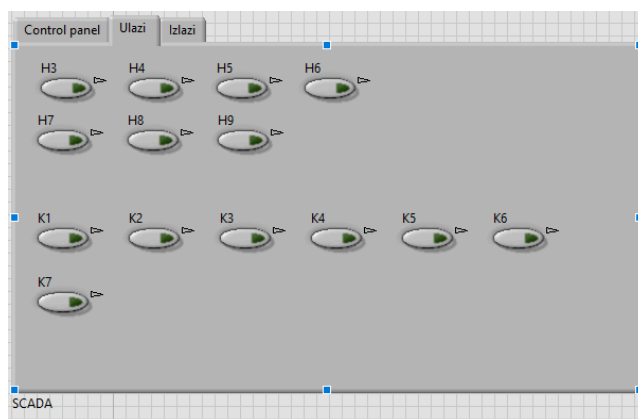
SCADA je implementirana u posebnom VI koji se nalazi na računaru. Pomoću SCADA sistema se može upravljati sistemom kao i videti stanja senzora, kontrola i sl. Funkcionalnost SCADA sistema je implementirana pomoću deljenih promenljivih tako da je moguće na nekim mestima u *Block Diagram* u videti i deljene promenljive.



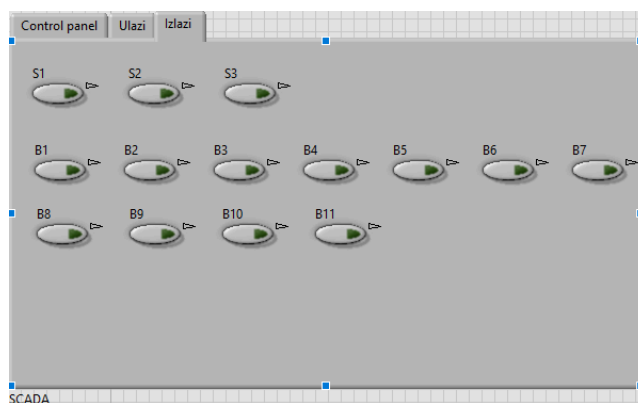
Slika 21: Deo Block Diagrama gde se mogu primetiti deljene promenljive koje se koriste radi SCADA sistema



Slika 22: Izgled SCADA sistema - Control Panel



Slika 23: Izgled SCADA sistema - Ulazi



Slika 24: Izgled SCADA sistema - Izlazi

Poteškoće prilikom izrade

Tokom izrade projekta nailazili smo na razne poteškoće o čemu će biti reči u ovom poglavlju. Projekat je u početku bio zamišljen poprilično drugačije nego što izgleda sada. Poteškoće koje smo imali su prevaziđene tako što smo ili pronašli neku zamenu u implementaciji tj. prilagodili se problemu ili smo u potpunosti rešili taj problem ali smo naveli ovde kao poteškoću, smatrajući je kao prekretnicu u razvoju projekta.

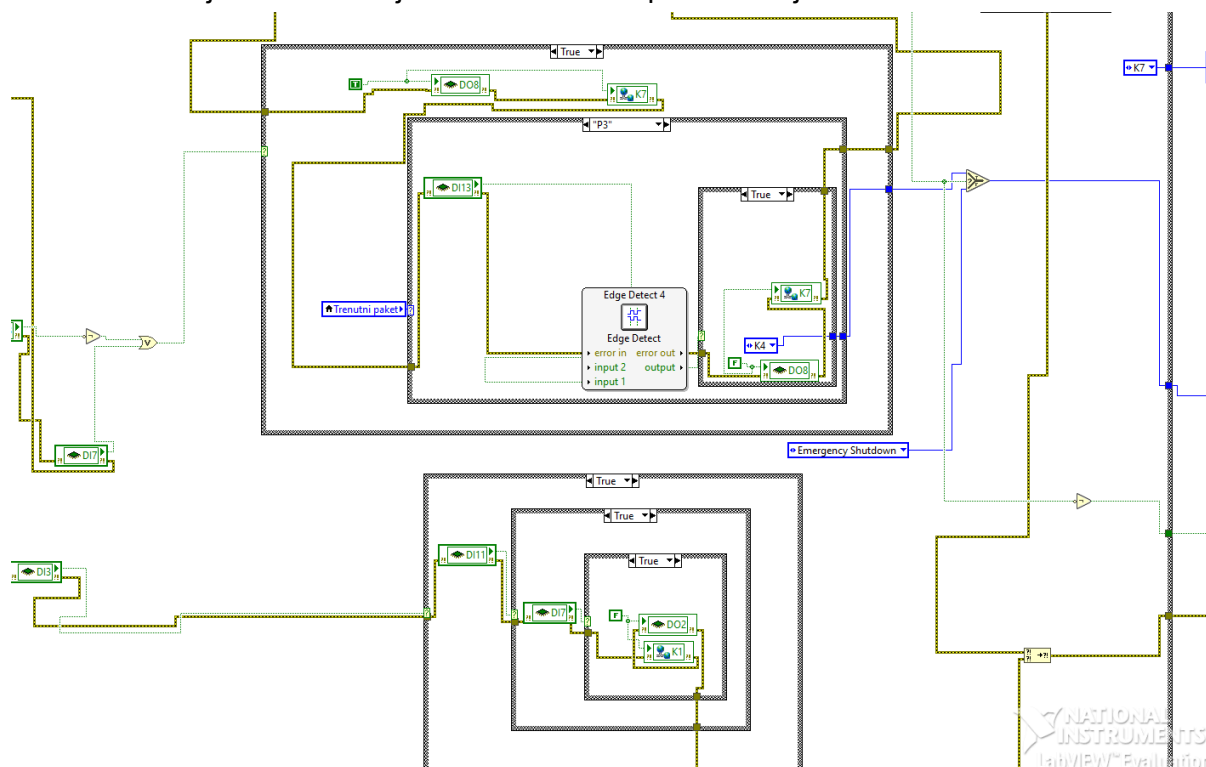
Paralelizacija

Prvi i najveći problem je bio - kako paralelizovati više stvari odjednom? Kako sortirati pakete i istovremeno gledati koji je paket sledeći? Kako upravljati sa više motora istovremeno?

Ovo nije bio problem koji se javio u toku izrade, nego je ovo bio početni problem s toga je zahtevao da odmah bude rešen kako bi se mogla nastaviti realizacija projekta.

Paralelizacija istovremenog rada više motora

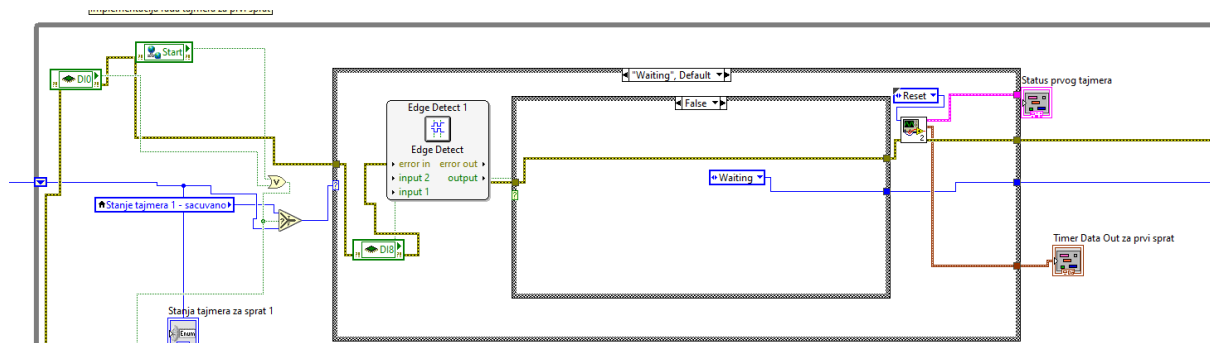
Detaljnijom analizom rada sistema shvatili smo da veliki dio istovremenog upravljanja sa više motora može biti realizovano u osnovnom automatu stanja. Npr. iako postoji stanje K1 čija je osnovna namena da pokrene motor M1 kontrolom K1, to nije jedino stanje gde se proveravaju uslovi za rad motora M1. Recimo, u stanju K7, kojim se kontroliše smer kretanja lifta na gore, takođe se proverava i da li M1 može da radi jer se može desiti da ako lift ode sa prvog sprata a motor M1 i dalje radi, da paket koji je sledeći padne sa dovodne trake. Baš u ovom stanju (K7) se proverava kada će se motor M1 isključiti. Iz ovog i sličnih razloga, pri gledanju rešenja, može se naići na međusobno slične delove koda u različitim stanjima automata ali to je urađeno isključivo tako u svrhu paralelizacije rada više motora.



Slika 25: Izgled dela Block Diagram-a - stanje K7 i istovremena provera uslova za rad motora M6 i M1

Još jedan tip problema paralelizacije koji spada pod ovu kategoriju paralelnog upravljanja sa više motora istovremeno jeste uključivanje i isključivanje rada odvodnih traka za pakete P1, P2 i P3. Iako je ovaj problem bio među posljednje rešenim problemima, o njemu pišemo ovde jer spada u ovu kategoriju problema.

Kako smo zaključili iz analize sistema da od stupanja paketa na određenu odvodnu traku i početka rada motora te trake do odlaska paketa sa te trake i zaustavljanja rada tog istog motora prođe vreme od trideset sekundi, shvatili smo da će nam biti potrebni tajmeri ali da oni neće moći biti upotrebljeni u glavnoj petlji sa osnovnim automatom stanja. S toga, za sve odvodne trake je napravljena jedna posebna petlja sa posebnim tajmerom koji broji vreme rada određene odvodne trake. Tu smo koristili poseban automat stanja sa stanjima *Waiting* i *Running*. U stanju *Waiting* se čeka paket da dođe na određenu odvodnu traku, nakon čijeg dolaska se broji trideset sekundi i po tom se zaustavlja ta odvodna traka. U stanju *Running* se proverava vreme tajmera i ukoliko je vreme isteklo, prelazi se u stanje *Waiting*. U ovom delu se takođe vodi računa o pritiskanju tastera *Emergency Off* ali o ovome će biti govora u nastavku.



Slika 26: Deo Block Diagram-a sa prikazom petlje i implementacije tajmera za prvi sprat

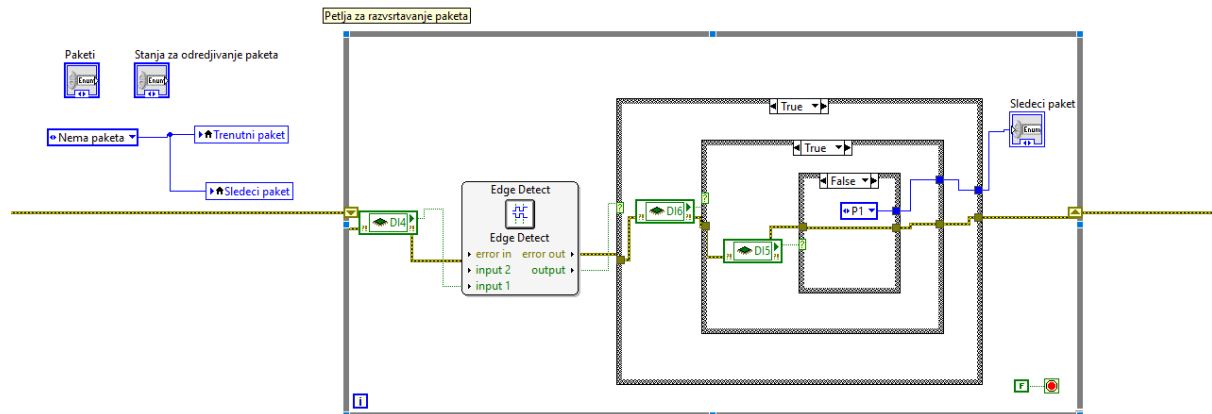
Paralelizacija određivanja paketa i ostatka sistema

Bilo bi jako zahtevno da smo određivanje paketa stavili u istu petlju koja se koristi za automat stanja s toga smo odlučili da za određivanje paketa napravimo poseban automat stanja koji će biti baziran na stanjima senzora B2, B3 i B4.



Slika 27: Dio sistema dovoljan za određivanje paketa

Kako smo napravili poseban automat stanja za određivanje tipa paketa, implementirali smo i posebnu petlju za određivanje paketa a informaciju o trenutnom paketu i paketu koji čeka sortiranje smo čuvali u posebnim promenljivama koje koristimo u glavnom automatu stanja.

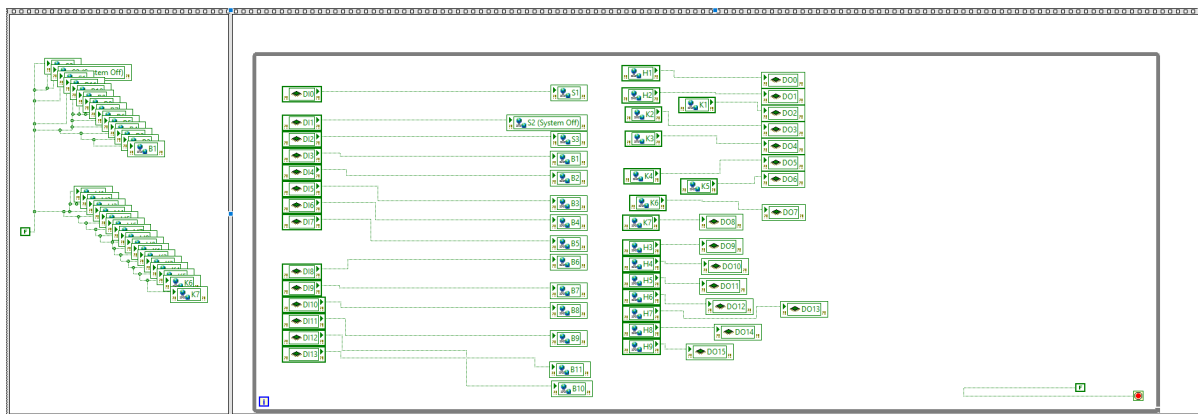


Slika 28: Izgled petlje sa automatom stanja za određivanje narednog paketa za sortiranje

Kašnjenje

Pod kašnjenjem podrazumeva se kašnjenje signala u mreži što je prouzrokovalo da se promene na mrežnim promenljivima ne uočavaju odmah kada bi i trebale već nakon izvesnog vremena koje i nije tako dugo, ali je značajno za upravljanje sistemom.

U početku je zamisao bila da se stvori višeslojna aplikacija u kojoj bi jedan sloj bio povezivanje signala sa mrežnim promenljivim, drugi - implementacija logike sistema i treći (koji bi se nalazio na samom računaru) - implementacija SCADA sistema. Međutim, kako je postojalo kašnjenje koje nismo mogli rešiti, došli smo do jedinog rešenja a to je uklanjanje sloja povezivanja i direktan pristup ulazima i izlazima sistema.

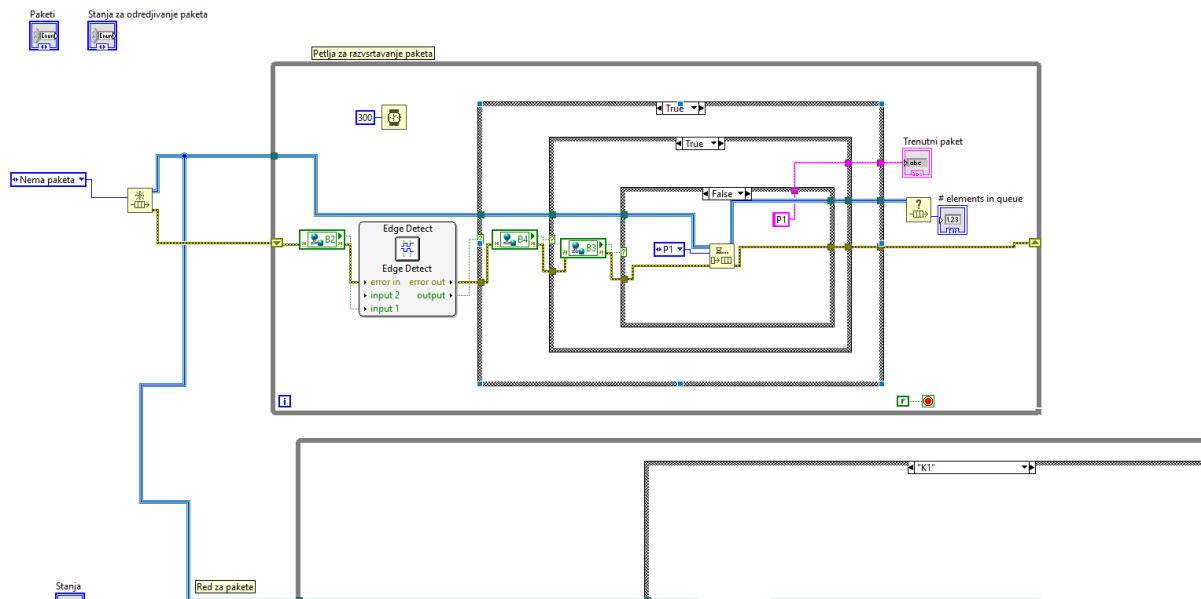


Slika 29: Izgled sloja za povezivanje koji se koristio u ranijim fazama projekta

Naravno, ispočetka nismo shvatili da je mreža problem te smo mislili da kašnjenje može biti prouzrokovano nečim drugim. Svakako, dobra pretpostavka u početku jeste bila da postoji neko kašnjenje samo nismo mogli shvatiti šta to kašnjenje izaziva. Kako nismo znali šta tačno prouzrokuje kašnjenje, posegnuli smo za raznim promenama u projektu koje su ostale i u poslednjoj verziji projekta. Problem sa mrežnim promenljivim nam je bila poslednja pretpostavka a do tada smo sumnjali u sledeće stvari:

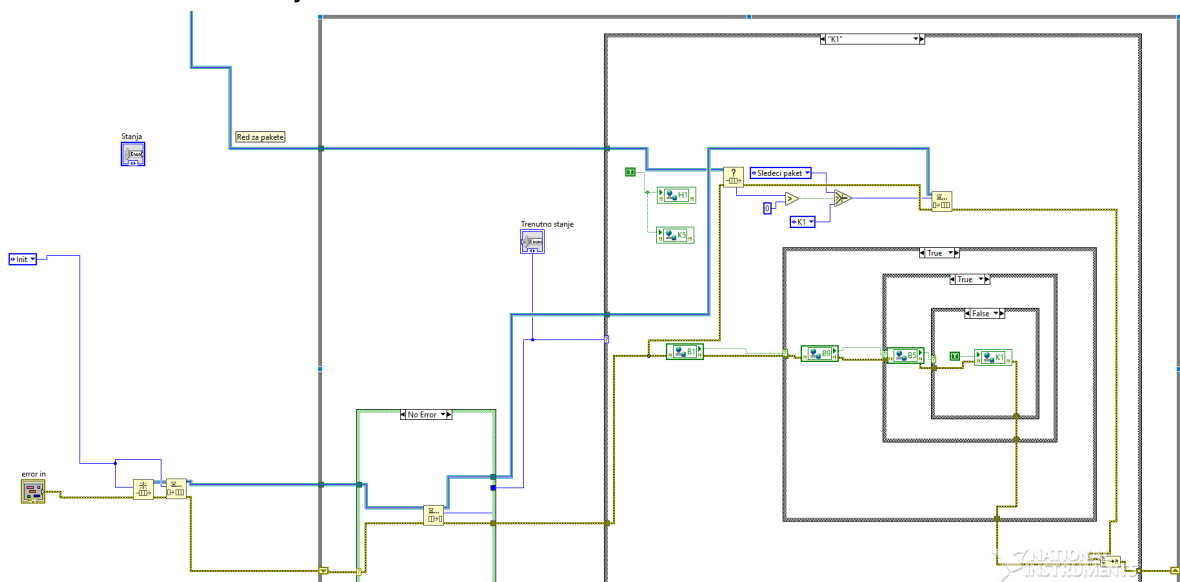
- Kašnjenje može biti prouzrokovano postojanjem mnoštva promenljivih, žica i sičnog - ovo nam je bila početna pretpostavka ali od nje smo odustali odmah u samom početku smatrajući je i ne baš adekvatnim razlogom za kašnjenje.
- Kašnjenje može biti prouzrokovano postojanjem reda za određivanje paketa - smatrali smo da red koji smo koristili za čuvanje paketa koji čekaju sortiranje može

da koči izvršavanje programa jer je već postojao jedan red koji smo koristili za osnovni automat stanja. Ova pretpostavka je rezultovala brisanjem reda za čuvanje paketa i pronalaskom drugačijeg rešenja.



Slika 30: Izgled petlje koja se koristila za određivanje paketa (implementacija sa redom)

- Kašnjenje može biti prouzrokovano redom koji se koristi za automat stanja - ovo je bila pretposljednja pretpostavka, prije nego što smo shvatili da je kašnjenje uzrokovala loša mreža. Zbog puno stanja i puno korišćenja reda, smatrali smo da ovo može kočiti izvršavanje programa, s toga smo red obrisali i koristili *Shift Register* za automat stanja.

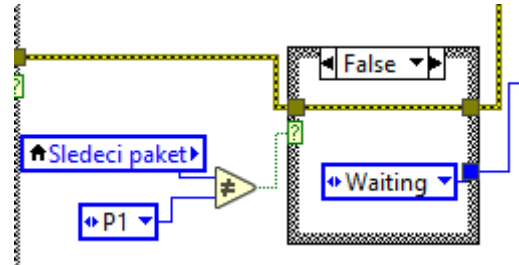


Slika 31: Osnovni automat stanja je u početku bio realizovan korišćenjem reda

Nailazak dva ista paketa uzastopno

Još i ranije je bilo reči o tome kako se tajmer koristi za određivanje zaustavljanja rada odvodnih traka. Međutim, prilikom prve realizacije tajmera, javio se problem pojavljivanja uzastopno dva ista paketa. U tom slučaju bi tajmer počeo da broji prilikom pojavljivanja

prvog paketa a prilikom pojavljivanja drugog paketa ne bi se desila nikakva akcija. Ovo je rezultovalo problemom da prilikom dolaska drugog paketa, paket stane u nekom trenutku na određenom delu odvodne trake. Kako se ovo ne bi dešavalo, implementirana je provera prilikom pokretanja tajmera a to je - da li je sledeći paket različit od trenutnog? Ako nije, tajmer se ne pokreće ako jeste - pokreće se.

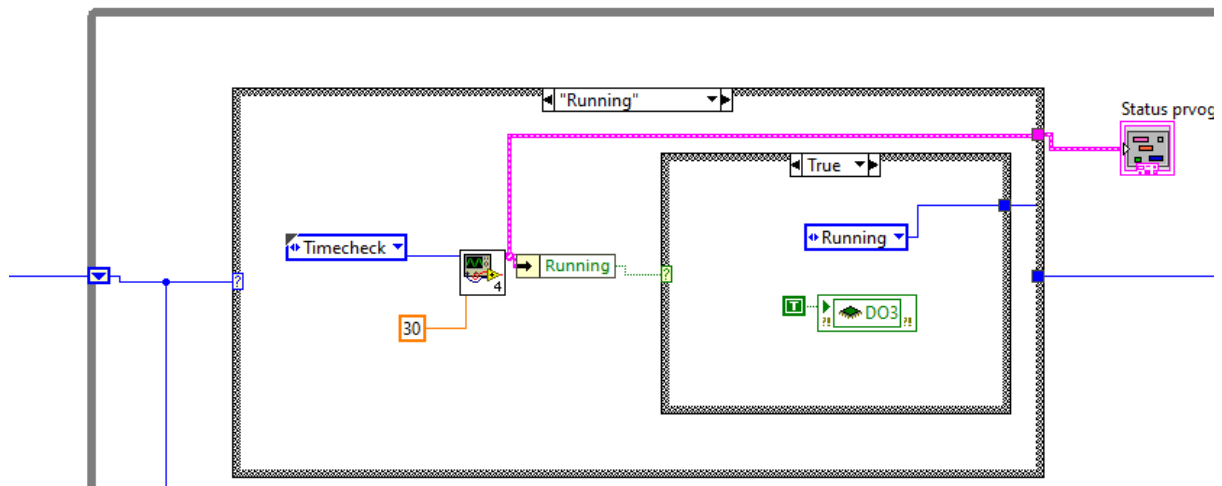


Slika 32: Dio Block Diagram-a u kojem se proverava sledeći paket

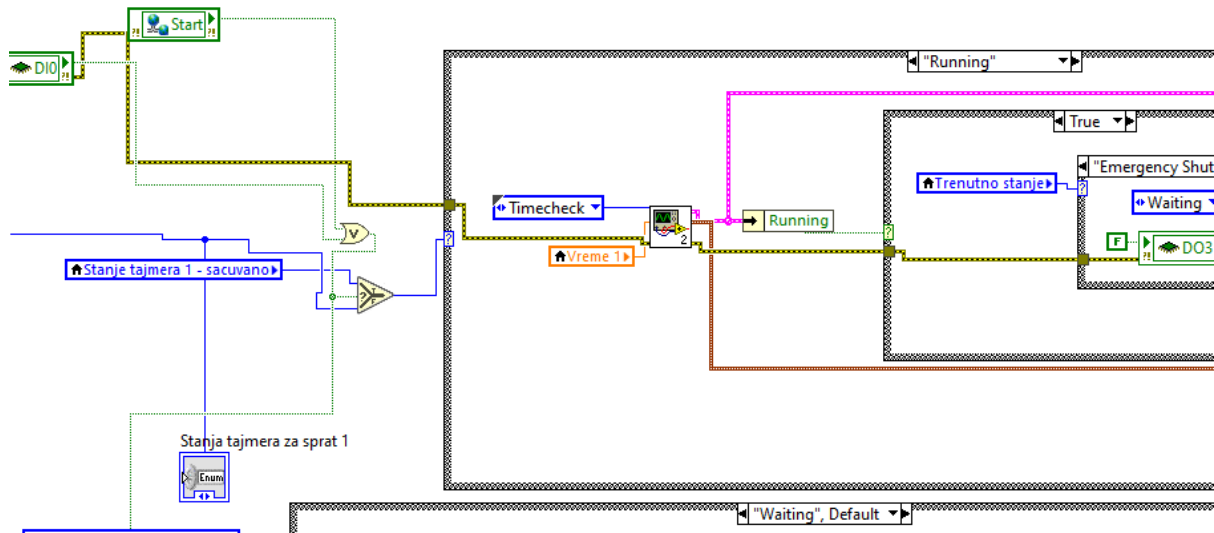
Tajmeri i Emergency stop

Nakon implementacije tajmera koji vrše kontrolu rada odvodnih traka, pojavio se problem pri zaustavljanju sistema pomoću *Emergency Off* tastera. Kada se sistem zaustavi pomoću pomenutog tastera dolazi do zaustavljanja celog sistema međutim odvodne trake rade još određeno vreme. S obzirom da je trebalo zaustaviti celi sistem (što obuhvata i odvodne trake) te takođe i zapamtiti stanje u kojem se sistem nalazio kako bi nakon ponovog pokretanja nastavio odakle je stao, morali smo izmeniti logiku svakog tajmera za odvodne trake.

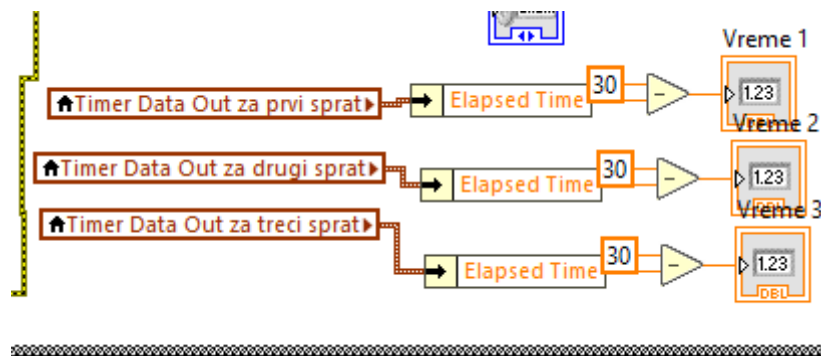
Konačno rešenje se ogleda u tome da imamo promenljive za svaki tajmer u kojima čuvamo vreme koje je potrebno odnosno koje je ostalo da tajmer odbroji. Kada se uđe u *Emergency Off* stanje, čuva se preostalo vreme i tajmeri prestaju sa radom. Kada se sistem ponovo pokrene, sačuvana vremena se prosleđuju tajmerima i tajmeri broje preostalo vreme.



Slika 33: Izgled tajmera prije korišćenja sačuvanog vremena (u ovom primeru se koristi fiksno 30 sekundi bez obzira na *Emergency Off*)



Slika 34: Izgled posljednje implementacije tajmera odbrojavanjem sačuvanog vremena (promenljiva Vreme ima vrednost 30 sekundi ukoliko nije aktiviran Emergency Off)



Slika 35: Dio Block Diagram-a u kojem se čuva vreme prilikom Emergency Off-a

SCADA

Za SCADA sistem se može reći da je jedina poteškoća za koju nismo mogli naći adekvatno rešenje osim da ostavimo sve kako jeste. SCADA sama po sebi zahteva kvalitetnu mrežu odnosno da nema ni najmanjeg kašnjenja što kod nas nije bio slučaj. Iako ima dovoljno veliko kašnjenje da poremeti normalan rad sistema, SCADA je implementirana kako bi se pokazao njen rad. Kašnjenje u prikazu stanja senzora, kontrola i indikatora je i zanemarljivo dok kašnjenje osnovnog upravljanja iz Control Panel-a ne može biti zanemarljivo, s toga se preporučuje direktna kontrola sistema odnosno kontrola sistema bez korišćenja SCADA.

Zaključak

Za kraj, smatramo da smo implementirali funkcionalan sistem koji ispunjava sve potrebne zahteve. Sistem je sposoban da raspoznaje pakete prema njihovoj veličini. Takođe je sposoban da istovremeno upravlja sa više stvari - radom lifta, određivanjem paketa, upravljanjem odvodnih traka i dovodne trake i sl.

Tokom izrade projekta, razvijajući rešenje koje će zadovoljiti potrebe problema, stekli smo iskustvo timskog rada uvažavajući različita mišljenja, te se usavršili u radu u LabView okruženju.