

Факултет инжењерских наука

Универзитета у Крагујевцу



Назив студијског програма: Машинско инжењерство

Ниво студија: Мастер академске студије

Модул: Индустријски инжењеринг

Предмет: СИМ системи

Број индекса: 411/2011

Вељко Б. Јовановић

ПРОГРАМИРАЊЕ PLC КОНТРОЛЕРА

мастер рад

Комисија за преглед и одбрану:

1. **др Миладин Стефановић, ред. проф.**

2. _____

3. _____

Датум
одбране: _____

Оцена: _____

У оквиру овог завршног рада кандидат треба да објасни примену и принцип рада PLC контролера као и основне софтверске захтеве и програмске шеме на примеру једне од радних јединица у индустријском постројењу.

Препоручена литература:

- [1] Србијанка Турајлић, Предавање, „Програмабилни логички контролери“, Београд: Електротехнички факултет, 2011.
- [2] Hugh Jack (2007). *Automating Manufacturing Systems with PLCs* преузето са http://claymore.engineer.gvsu.edu/~jackh/books/plcs/pdf/plcbook5_0.pdf (приступило 3.7.2014)
- [3] Др Золтан Јегеш, Милан Аџић, Роберт Мартон, „Управљање применом PLC уређаја“, Суботица: Виша техничка школа, 2005.
- [4] Драган М. Маринковић, „Програмабилни логички контролери – Увод у програмирање и примену“, Београд: Микрокњига, 2012.
- [5] L.A. Bryan (1997), „*Programmable Controllers, Theory and implementation second edition*“ преузето са http://eelalnx01.epn.edu.ec/bitstream/15000/9275/2/Industrial_Text_and_Video_PLC_Reference.pdf (приступило 3.7.2014)
- [6] Frank D. Petruzella (2005), „*Programmable Logic Controllers third edition*“ преузето са <http://www.msalah.com/A/Atextbook.pdf> (приступило 3.7.2014)

Резиме

У оквиру овог мастер рада дали смо увод у програмирање PLC контролера, затим смо у другом поглављу дефинисали структуру PLC система, које су компоненте PLC система, који су то улазни и излазни PLC модули, које су предности и недостаци PLC система, затим у трећем и четвртом поглављу говоримо о самом програмирању и основним карактеристикама процесорског модула. У петом поглављу смо објаснили систем за аутоматизацију у ауто индустрији за коју је коришћен PLC и дали конкретан пример система алата за монтажу петих врата.

Кључне речи: PLC (Programmable logic controller) системи, индустријска постројења, софтвер

Abstract

Within this master work, we gave an introduction to programming PLC controllers, we defined in the second chapter, the structure of the PLC system, components PLC system, which are the input and output modules, advantages and disadvantages of PLC system, then in the third and fourth chapter we detailed talk about programming and the basic characteristics of the processor module. In the fifth chapter, we explain the system for automation in car industry, which has used PLC, also we gave a concrete example of a tool for mounting the tailgate.

Key words: PLC (Programmable logic controller) systems, industrial facilities, software

Садржај

1.0 Увод	4
2.0 Структура PLC система	5
2.1 Компоненте PLC система	6
2.2 Улазни и излазни PLC модули.....	7
2.3 Дискретни U / I модули.....	8
2.4 Аналогни PLC модули	9
2.5 Предности коришћења PLC контролера	10
2.6 Недостаци коришћења PLC контролера	11
3.0 PLC Програмирање.....	12
3.1 Лествичасте (ladder) дијаграми	13
3.2 Дијаграми функцијских блокова	14
3.3 Структурални текст	15
3.4 Дијаграми секвенцијалних функција	16
3.5 Скен циклус.....	20
4.0 Основне карактеристике процесорског модула	22
4.1 Организација меморије	22
4.2 Програмске датотеке	23
4.3 Типови података.....	23
4.4 Елемент датотеке.....	25
4.5 Креирање датотеке и заузеће меморије	28
5.0 Систем за аутоматизацију у ауто индустрији за коју је коришћен PLC	30
6.0 Закључак.....	39
7.0 Литература.....	40

1.0 Увод

Употреба PLC уређаја је у савременим индустријским постројењима неизоставна ставка јер се бави регулисањем свих процеса који се дешавају на производној линији а њиховим технолошким напредком и програмирањем одређених процеса се долази до бржег процеса производње, постизања бољег квалитета рада и квалитета жељених производа.

PLC контролер је микропроцесорски уређај који користи програмабилну меморију за памћење наредби којима се захтева извођење специфичних функција, као што су логичке функције, секвенцирање, пребројавање, мерење времена и израчунавање ради управљања различитим типовима машина и процеса. Модерна контрола система и даље укључује релеје, али се ретко логички користе.

Циљ овог рада је објаснити структуру и из којих компоненти се састоји један PLC уређај, који су модули који се примењују у зависности од потребе у пракси, навести које су предности и мане као и причати о основама програмирања како теоретски тако и на конкретном примеру у производној хали а да би се то остварило рад је конципиран у следећих 5 поглавља. У уводу се укратко описује циљ и структура самог рада, затим се у другом поглављу описује структура PLC система, у трећем поглављу се говори о самом PLC програмирању, четврто поглавље се бави основним карактеристикама процесорског модула док се у петом поглављу објашњава функционисање PLC контролера на конкретном примеру у пракси у производној хали.

2.0 Структура PLC система

PLC као индустријски рачунар самим својим дизајном предвиђен је за примену у непосредном окружењу процеса са којим управља, тако да је отпоран на разне неповољне утицаје, прашина, влага, висока температура, вибрације и електромагнетне сметње, тако да се обично примењује за решавање децентрализованих управљачких задатака, на самом месту управљања, где се повезује преко улаза и излаза са уређајима као што су операторски панели, мотори, сензори, прекидачи, вентили и сличним. PLC као и сваки рачунар има оперативни систем, који свакако има много мање могућности од оперативних система опште намене, али у данашње време опште потребе за комуникацијама, може у потпуности да их подржи. Стога је могуће извести повезивање програмабилних логичких контролера (PLC-а) и евентуално централног рачунара или других рачунара, ради решавања сложенијих управљачких задатака или једноставне аквизиције података и управљања са даљине. Учинак комуникације међу PLC уређајима су тако велике да омогућавају висок степен искоришћења и координације процеса, као и велику флексибилност у реализацији управљачког процеса, тако да могућност комуникације као и флексибилност претстављају главне предности примене решења са PLC уређајима. [3]

2.1 Компоненте PLC система

PLC контролер је један од елемената аутоматизованог система, који на основу прихваћених улазних сигнала са улазних уређаја у зависности од тога како је програмиран прави излазне сигнале са којима управља излазним уређајима и обично је центар једног таквог система. Компоненте PLC система су све међусобно повезане и то су :

- централни рачунар,
- процесни рачунар,
- панели,
- прекидачи,
- сензори,
- мотори,
- вентили,
- индикатори,
- сирене,

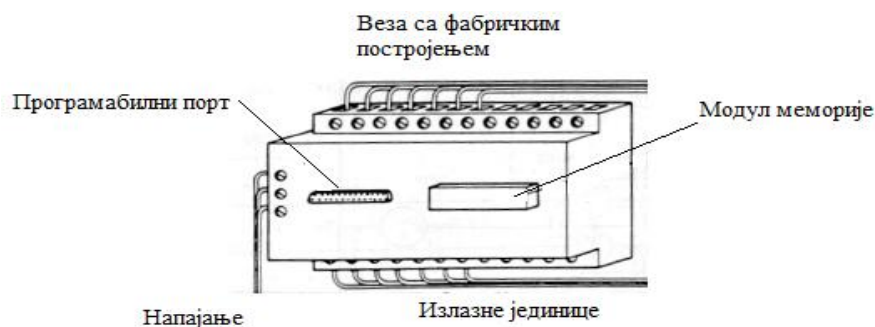
PLC контролери су уређаји који служе за управљање и контролу процеса на производној линији у индустријским постројењима и као такве њихова конструкција

може бити компактна и модуларна. Компактни PLC контролери су затворени уређаји са фиксним бројем улаза и излаза. У кућишту компактног PLC уређаја налазе се :

- извор напајања,
- централна процесорска јединица (CPU),
- улазни и излазни модул,

PLC компактни контролери су економични и служе за управљање једноставнијим системима и процесима, поседују до 16 улаза и 16 излаза.

На слици 2.1 приказана конфигурација једног компактног PLC уређаја.

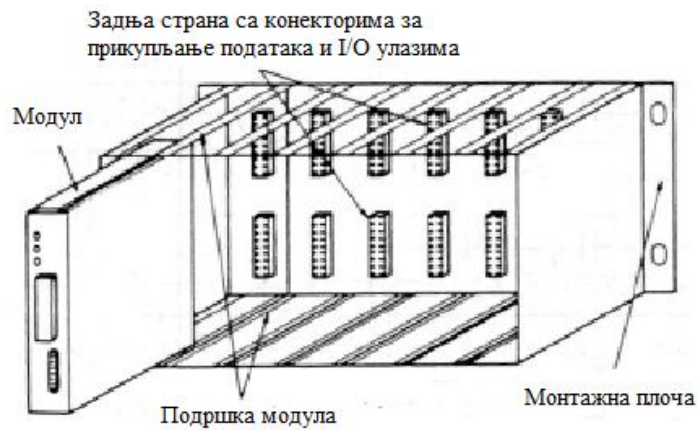


Слика 2.1 Компактни PLC [4]

Централна процесорска јединица је главна карика у функцији PLC контролера. То је обично неки од микроконтролера, пре су се користили 8-битни микроконтролери, а данас су 16-битни и 32-битни микроконтролери. CPU се брине о комуникацији, међусобној интеракцији делова контролера, извршавању програмских наредби, организацији меморије, надгледању улаза и излаза. На самом контролеру постоји неколико индентификатора у облику светлећих диода за јављање грешке.

PLC контролер се разликује од рачунарског система опште намене по томе што нема спољну меморију, као и низ стандардне улазно / излазне опреме. Поред тога, његов оперативни систем је једноставнији и пружа компаративно мање могућности од рачунара опште намене. PLC је направљен да служи за одређене послове и управљање уређајима у индустрији. Корисник PLC контролера је у највећој могућој мери ослобођен решавања различитих проблема везаних за чисто рачунарски аспект, као што су промена или додавање U/I јединица, везивање у рачунарску мрежу, размена података и слично и може да се у пуној мери концентрише на пројектовање саме апликације.

На слици 2.2 можемо видети приказ модуларног PLC уређаја као и његову конструкцију.



Слика 2.2 Модуларни PLC [3]

Сваки слот чини пар вођица дуж горње и доње стране река, које служе за механичко учвршћење модула као и конектор на задњој плочи река преко кога се модул прикључује на заједничку магистралу изведену на штампаној плочи задње стране река. Први слот је обично резервисан за модул напајања. Други слот се користи за модул логичке јединице, док се преостали слотови користе за модуле са специјалним наменама као што су меморијски модули и други. Због овако погодне конструкције могуће је лако проширење или ако постоје повећани захтеви обраде и постојећи процесорски модул више не одговара може се заменити новим, а да притом остали модули који функционишу остају у систему.

2.2 Улазни и излазни PLC модули

Модул може имати 4, 8, 12 или 16 терминала (приступних тачака). Улазни, излазни или комбиновани модули (са подједнаким или различитим бројем улазних и излазних терминала, нпр 12 улаза и 8 излаза). Код већих PLC система, улазни и излазни модули су смештени у посебне рекове који се са CPU-ом повезују помоћу одговарајућег вишежичног кабла. Сваки U / I модул има јединствену адресу.

Карактеристике U / I модула:

- опсег напона или струје,
- DC или AC,
- произвођачи PLC система нуде палете улазних и излазних модула декларисаних за различите опсеге.

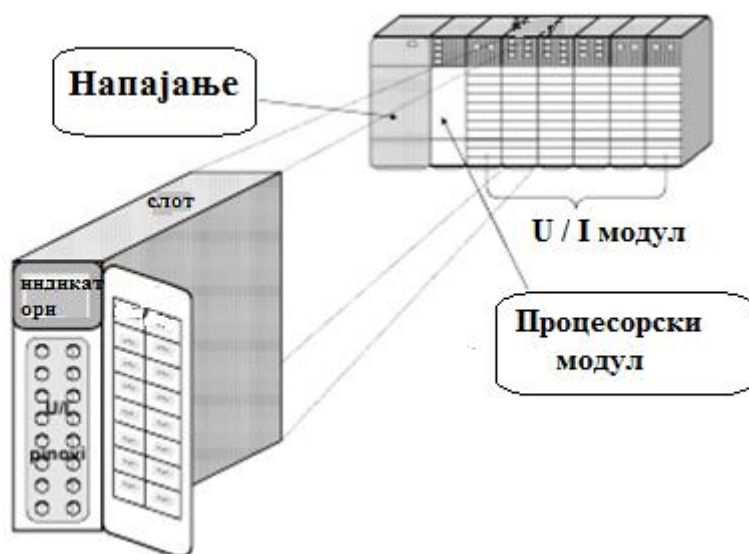
Уређаји са којима се PLC повезује су :

- улазни ON / OFF уређаји,
- улазни аналогни уређаји,
- излазни ON / OFF уређаји,
- излазни аналогни уређаји,

Дискретни U/I модули - прихватају и генеришу дискретне - дигиталне, тј ON / OFF сигнале. Аналогни U/I модули - прихватају и генеришу аналогне сигнале (Поседују уграђене A/D, односно D/A конверторе). Удаљени U/I модули-спрега са CPU није директна, већ комуницирају преносом података.

2.3 Дискретни U/I модули

Упркос чињеници да дискретни сензори и актуатори који се налазе на неком процесу или постројењу могу имати веома различите техничке карактеристике, захтев који се поставља при њиховом везивању за контролер је увек исти. Наиме, од контролера се очекује да обезбеди конверзију дигиталног (бинарног) сигнала који долази са сензора у нумеричку вредност 0 или 1 и да тај податак смести као један бит на одговарајуће место у меморији, или да очита нумеричку вредност (садржај) неког бита у меморији, да га конвертује у бинарни сигнал који се води на актуатор. Ова чињеница омогућила је пројектовање и израду типизираних U/I кола која су у стању да обрађују готово све сигнале који се срећу код индустријске мерне опреме и извршних органа. Поред тога, неколико U/I кола су груписана заједно и чине дискретни U/I модул, чија веза са контролером се остварује једноставно убацивањем у одговарајући слот на шасији.



Слика 2.3. Дигитални U/I модул. [3]

Изглед типичног U/I модула приказан је на Сл 2.3. На предњој плочи U/I модула налази се одређени број пинова (терминал поинтс) за које се везују излази са мерних инструмената, односно улази у извршне органе. Сваки пин је заправо улазна или излазна тачка одговарајућег кола за спрегу са контролером. У складу са тиме сваки пин се идентификује својим типом (улаз или излаз) и бројем који одредјује положај U/I кола у оквиру модула, и који заправо представља адресу пина. Опис пинова дат је на

унутрашњој страни врата на модулу. Поред U/I пинова, на предњој плочи модула налазе се и пинови који су интерно повезани са напајањем (DC или AC) заједничком (нултом) тачком и са земљом. Начин спрезања појединог уређаја са модулом зависи од специфичности самог уређаја, као и карактеристика модула, детаљна шема спреге везивања дата је уз сваки модул.

За време рада U/I модула, стање сваког пина се приказује на одговарајућем LED индикатору. Индикатор који је повезан са улазним пином светли ако је улазни сигнал у стању логичке јединице. Индикатор повезан са излазним пином светли ако је, као резултат обраде програма, на излазни пин постављена логичка јединица.

Постоје три типа U/I модула: улазни, излазни и комбиновани улазно / излазни модул. Они се израђују са различитим густинама пинова (4, 8, 04:32 пина по модулу) и могу се спрезати са AC, DC и TTL напонским нивоима.

Као што се види на слици 2.3 U/I модули се смештају у слотове на шасији. Максималан број модула који се може директно повезати са једним контролером, зависи од величине шасије и броја слободних слотова. Будући да сваки слот има своју адресу унутар шасије, то значи да је самим стављањем модула у слот одређена и његова адреса. Коначно, као што је већ истакнуто, и сваки пин унутар једног модула има своју адресу. У складу са тим сваки пин има своју јединствену адресу која је одређена адресом слота у који се модул поставља али адресом слота.

2.4 Аналогни PLC модули

Аналогни улазни модули прихватају аналогну информацију са аналогних сензора и конвертују у дигиталну информацију за даљу обраду у PLC контролеру. Зато се зови и A/D модули. Постоје два основна типа A/D модула: са струјним и са напонским улазом.

Аналогни модули са напонским улазом доступни су две варијанте: униполарни и биполарни. Униполарни модули приватају напон 1 поларитета, на пример од 0 до +10 V. Биполарни модули могу прихатати улазни напон оба поларитета, на пример од -10V до +10 V. [1]

Аналогни модули са струјним улазом, стандардно прихати струју у опсегу од 4 mA до 20 mA, при чему улазна струја од 4 mA представља најмању, а улазна струја од 20 mA највећу улазну вредност. (Уочимо да дозвољени опсег улазне струје не обухвата 0A. То је учињено из разлога како би се омогућила детекција прекида везе између сензора и улазног модула. Ако веза постоји, између сензора и улазног модула увек тече струја већа од 4 mA. Ако је веза у прекиду, струје нема).

Већина аналогних улазних модула се може конфигурисати од стране корисника. Конфигурацијом се подешава напонски или струјни опсег. Постоје модули и са напонским и са струјним улазима, тако да корисник може користити улаз који одговара конкретној примени.

Типично, A/D модули за индустријске примене имају 12-битну резолуцију, што значи да се пуни обим улазног напона или струје пресликава на 4096 дигиталних речи. Типично, аналогни модули имају од једног до осам независних аналогних улаза или канала. Доступни су и аналогни модули специјалне намене, као што су модули за спрегу са термопаровима. Ради се о A/D модулу који је прилагођен за прихватање излаза термопара. Излазни сигнал термопара је веома малог напона, што значи да мора бити појачан пре него што се обави A/D конверзија. Конверзија напона термопара у температуру захтева одређене компензације и корекције, што је такође функција коју обавља овај специјализован аналогни улазни модул.

Намена аналогних излазних модула је конверзија дигиталне излазне информације коју генерише PLC контролер у аналогни напон или струју која се користи за управљање неког специфичног излазног уређаја. На пример, аналогни излаз се може користити за управљање брзином ротације мотора или управљање струјом кроз грејач. Доступни су аналогни излазни модули са напонским или струјним излазом. Аналогни излазни модули генеришу сигнале чији напон или струја може бити произвољан у оквиру фиксног опсега. Аналогни излазни модули генеришу сигнале чији напон или струја може бити произвољан у оквиру фиксног опсега. Типичан опсег излазног напона је 0 до 10 V, а обим струја 4 mA до 20 mA. Поред аналогних PLC модула који су искључиво улазни или искључиво излазни, у употреби су и комбиновани аналогни U/I модули, који поседују одређени број аналогних улаза и аналогних излаза (нпр. 2 улаза и 2 излаза).

2.5 Предности коришћења PLC контролера

Проблеми на које можемо наићи приликом комбиновања више релеја у сложену електричну шему могу бити:

Поузданост система, цена релеја, жица и уградња целокупног уређаја, простор кабинета за смештање, тешко проналажење делова као и компликовано отклањање кварова. Приликом нових промена у постројењу потребан је нови распоред као и повезивање релеја, целокупан систем мора бити заустављен у току измене постројења. Додајемо да релеји имају покретне делове који су склони престанку рада и хабању из

разлога што спадају у механичке компоненте. Уочавање кварова и отказивања је теже у системима са великим бројем релеја.

Сваки уређај, па и PLC конструисани су са циљем да превазиђу многобројне проблеме засноване на електромеханичким релејима. Ови контролери су данас у широкој примени због ниских цена и поузданости. PLC контролери су ефикаснији од релејних техника из разлога што су поузданији пре свега јефтинији, лакши за коришћење и одржавање система.

Иако постоје многобројне предности коришћења PLC контролера у односу на релејну технику, главна предност ових уређаја лежи у томе што се програмирање врши помоћу одређених програмских језика чија се основа базира на лествичастим дијаграмима те не морају инжењери који користе овај уређај бити експерти за програмирање јер су једноставни, како би руководили истим. Први PLC контролери су били једноставни уређаји за on / off управљање и користили су се за замену застареле релејне технике. Међутим, такви PLC контролери нису могли обезбедити сложеније управљање, као што је управљање температуром, притиском, позицијом. У међувремену, произвођачи PLC контролера развили су и уградили у PLC контролере бројна побољшања и функционална унапређења. Савремени PLC контролери имају могућност обављање изузетно сложених задатака како што је управљање прецизним позиционирањем и управљање сложеним технолошким процесима. Такође, брзина рада PLC контролера је значајно повећана, као и лакоћа програмирања. Развијени су бројни модули специјалне намене за примене као што је радио комуникација, визија или чак препознавање говорних команди.

2.6 Недостаци PLC-а

Поред многобројних предности које поседује PLC уређај, присутни су и недостаци који не умањују квалитет функције PLC система. Развојем нових технологија развио се и концепт рада са индустријским уређајима па је прилично тешко променити начин функционисања и размишљања инжењера који су радили са старом технологијом. Такође, постоје апликације које су предвиђене само за једну функцију која се мења ретко или готово никада. PLC уређаји не подnose екстремне услове рада попут високих температура, влажности и др. Као најбитнија ставка релејног уређаја је та да постоји СТОП прекидач који зауставља рад целог система релеја које се не ресетује аутоматски, већ наставља са радом приликом укључења.

3.0 Програмирање PLC система

Први PLC уређаји су програмирани техником која се базирала на релејној логици жичаних шема. И ако елиминише потребу да се обучавају електричари, инжењери и техничари како да програмирају рачунар, ова метода је најпродуктивнија за програмирање PLC-ова данас. Постоје и друге методе за програмирање PLC-а. Једна од ранијих техника укључује одређене инструкције. Ове инструкције могу бити изведене директно из лествичастих логичких дијаграма и уметнуте у PLC уређај кроз прост програмски терминал. Међутим, ако се пође од чињенице да је PLC пројектован као наменски микрорачунарски систем за управљање и надзор рада неког процеса, са посебним оперативним системом који обезбеђује периодично понављање скен циклуса. За његово програмирање развијен је и посебан програмски језик, а за развој посебног језика била је и чињеница да је PLC почетно развијен са идејом да замени релејне системе. Успешна примена PLC-а у пракси захтевала је и да се његово програмирање прилагоди техници која је свим корисницима релејних система добро позната. Специфичност пројектовања релејних система огледа се у томе што је потребно да се утврди начин графичког представљања временске секвенце логичких операција. Класични логички дијаграми су доста корисни за приказивање релација између елемената комбинационе логике. Њихова мана је што они не пружају могућност за приказивање различитих улазно/излазних променљивих као функција времена. Са друге стране, временски дијаграми су изузетно погодни за приказивање односа појединих променљивих у току времена, али не омогућавају да се прикаже логика која условљава те односе. У циљу спајања обе врсте приказивања, за пројектовање релејних система развијени су ледер (лествичасти) дијаграми. Пројектовање PLC-ова је подразумевало да се за њих мора развити и одговарајући програмски језик заснован на ледер дијаграмима-ледер програмски језик. Овај језик погодан када се ради о програмирању мањих PLC - ова. За сложеније задатке, развијени су и други програмски језици, који се често срећу само на већим контролерима. У основи сви језици се могу поделити у четири категорије.

Ladder (лествичасти) дијаграми - Ледер логика је главна метода за програмирање која се користи за PLC. Као што већ знамо ледер логика се развила да замени релеј логику. Ледер дијаграми су намењени извођењу низа, првенствено, логичких операција, креирању и слању порука, испитивању стања контролера у мрежи, као и операцијама које би требало да изводе оператери у случају детектовања и отклањања извесних грешака.

Дијаграми функцијских блокова - намењени реализацији континуалних процеса, управљању драјверима и управљању у затвореној спрези.

Структуриран текст - намењен реализацији сложених математичких операција, обради ASCII низова и протокола, као и обради специјалних табела података.

Дијаграми секвенцијалних функција - Дијаграми секвенцијалних функција су развијени тако да програмирају веће и напредније системе. Слични су дијаграмима тока, али су притом и много моћнији. [1]

У принципу прва три језика укључују репертоар наредби којима се обављају једноставније логичке и математичке операције. Она су намењена класичном програмирању управљања у затвореној спрези, секвенцијалног управљања и обради датотека података. У том смислу ови језици се у великој мери могу алтернативно користити за реализацију појединих програмских рутина. Код неких контролера, сваки од ових језика оставља могућност кориснику и да сам креира низ наредби које постају саставни део репертоара. [2]

Дијаграми секвенцијалних функција намењени су, међутим, потпуно другачијој филозофији конципирања и програмирања управљања и надзора. Они омогућавају да се цео процес подели у низ сегмената. У сваком од сегмената се онда, уз помоћ класичних језика програмира низ операција које у њој треба да се изврше.

У последње време, за PLC-ове се развија још један сложен језик намењен управљању фазама кроз које пролази неки процес. Независно од језика којим је програм написан он се реализује када је задатак коме је он придружен активан. У принципу програми се изводе тако што се обави један скен циклус кроз све наредбе програма. Затим се извођење прекида и прелази на следећи програм, или следећи задатак.

3.1 Лествичасте (Ladder) дијаграми

PLC улази се лако представљају ледер логиком. Постоје три типа приказа инпута. Прва два су нормално отворена и затворена. Трећа функција омогућава да се инпути читају после скенирања док се у међувремену скенира ледер логика. Овим је омогућено испитавање улазних вредности више од једном у сваком циклусу. Ледер дијаграм има изглед лествица, одакле потиче и његов назив (ladder - лествице). Сваки ранг ледер дијаграма састоји се из два дела. На левој страни ранга налази се услов изражен у форми контактне (прекидачке) логике, док се на десној страни ранга налази акција која треба да се изврши уколико је услов испуњен (true) слика 3.1. [1]



Слика 3.1 Ранг ледер програма. [1]

У ледер логици постоји више врста аутпута, али они нису доступни свим PLC уређајима. Из тог разлога неки од аутпута морају бити екстерно повезани са уређајима ван PLC-а, али је могуће користити и интерне меморијске локације у PLC-у. Ледер програм се извршава у току програмског скен циклуса и то тако што се обрађује ранг по ранг у низу како су они дефинисани. У сваком рангу испитује се истинитост услова и уколико је услов истинит извршавају се одговарајуће наредбе акције. Ако се наредбом акције додељује нека вредност променљиве, онда ће се том приликом та вредност променити. Потребно је запазити, међутим, да уколико се променљива у наредби акције односи на физички излаз, вредност излаза неће бити промењена у истом тренутку времена. Излази ће бити промењени тек у излазном скен циклусу када се вредности променљивих из меморије пренесу на одговарајуће излазне модуле. Ледер програм креира се помоћу одговарајућег програмског графичког алата о коме ће касније бити више речи.

3.2 Дијаграми функцијских блокова

Бит логичке инструкције користе два основна борја 0 и 1. Ове две цифре су основа система који називамо бинарни систем. Ове две цифре називамо бинарним цифрама или битовима. У коњукцији са „И“, „ИЛИ“ и „КСОР“ функцијом заједно са аутпутима 1 се гледа као логичко ДА а 0 као логичко НЕ.

Бит логичке инструкције интерпретирају стање сигнала као 1 или 0 према Буловој логици. Овакав вид бит логичких операција добија резултат 1 или 0 и називамо га „Резултат логичке операције“ (RLO).

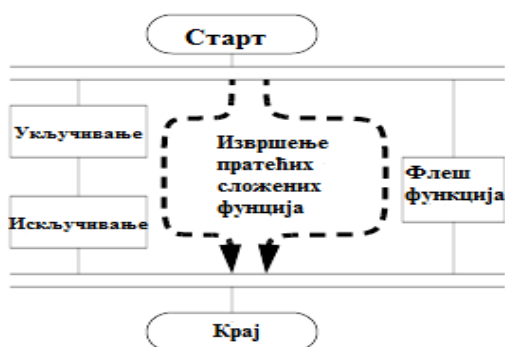
Постоје бит логичке инструкције за следеће функције AND, OR и Exclusive OR. Функцијски блок реализује функцију пресликавања улазних променљивих на излазне променљиве. Дозвољени број улаза и излаза зависи од одређене функције пресликавања. Улазне променљиве могу бити било које променљиве у програму или излази других блокова. Вредности ових променљивих се обрађују у функцијском блоку. Резултати обраде се преносе на излазне променљиве. Број различитих функција које су на располагању зависи од типа контролера. У принципу, независно од типа

контролера, сваки блок има свој посебан улаз (Enable_in) који служи за блокирање, односно деблокирање рада блока. На тај начин, функција коју обавља блок биће извршена само ако је функција којим се генерише сигнал Enable_in испуњен. Уколико су блокови повезани тако да образују затворену повратну спрегу неће бити могуће да се одреди редослед извођења операција. У циљу превазилажења овог проблема користи се посебан индикатор претпостављене расположивости података (Assume Data Available Indicator). Постављањем овог индикатора на улаз неког блока индицира се да ће он користити вредност коју је улазна променљива имала у претходном скен циклусу.

3.3 Структурални текст

Секвенцијални функцијски графикони су прилагођени за програмирање напредних система. Слични су дијаграмима тока али су моћнији. Пример са слике ради две различите ствари. Да би сте прочитали вредности са графикана почињемо од врха. Испод је двострука хоризонтална линија која прати обе стазе.

Пример који је приказан на слици 3.2 ради две различите ствари. Да би сте прочитали графикон морате почети од почетка који је обележен знаком „Старт“. Испод се налази дупла хоризонтална линија која прати две стазе. Као резултат PLC уређај ће почети да прати леву и десну страну појединачно и истовремено. Са леве стране наилазимо на две функције од којих је прва укључивање. Ова функција ће радити док не одлучи да заврши, након тога долази функција искључивања. На десној страни је „Флеш“ функција која се покреће све док не заврши рад. Ове функције делују необјашњиве, међутим свака функција као и повећање снаге биће део мале ледер логике. Ова метода је прилично другачија од дијаграма тока из разлога што не мора да прати једну траку.



Слика 3.2 Пример графика секвенцијалне функције [1]

Структурално програмирање је развијено као модернији програмски језик. Прилично је сличан језицима попут BASIC-а. Сличан пример приказан је на слици 3.3.

Он користи PLC меморијску локацију која је направљена тако да читава целобројне вредности.

```
i := 0;  
REPEAT  
i := i + 1;  
UNTIL i >= 10  
END_REPEAT;
```

Слика 3.3 Пример структуралног текста

Прва линија на програму поставља вредност 0. Следећа линија почиње петљом и завршава се тамо где се враћа кроз петљу. Следећа линија подсећа на вредност у локацији I и додаје јој вредност 1, затим враћа на исту локацију. Следећа линија проверава да ли петља треба да се заврши. Ако је ≥ 10 онда ће се петља завршити. У супротном компјутер ће се вратити на почетак програма и поновити процес. Сваки пут програм пролази кроз ову петљу и повећава вредност за један док не достигне десет.

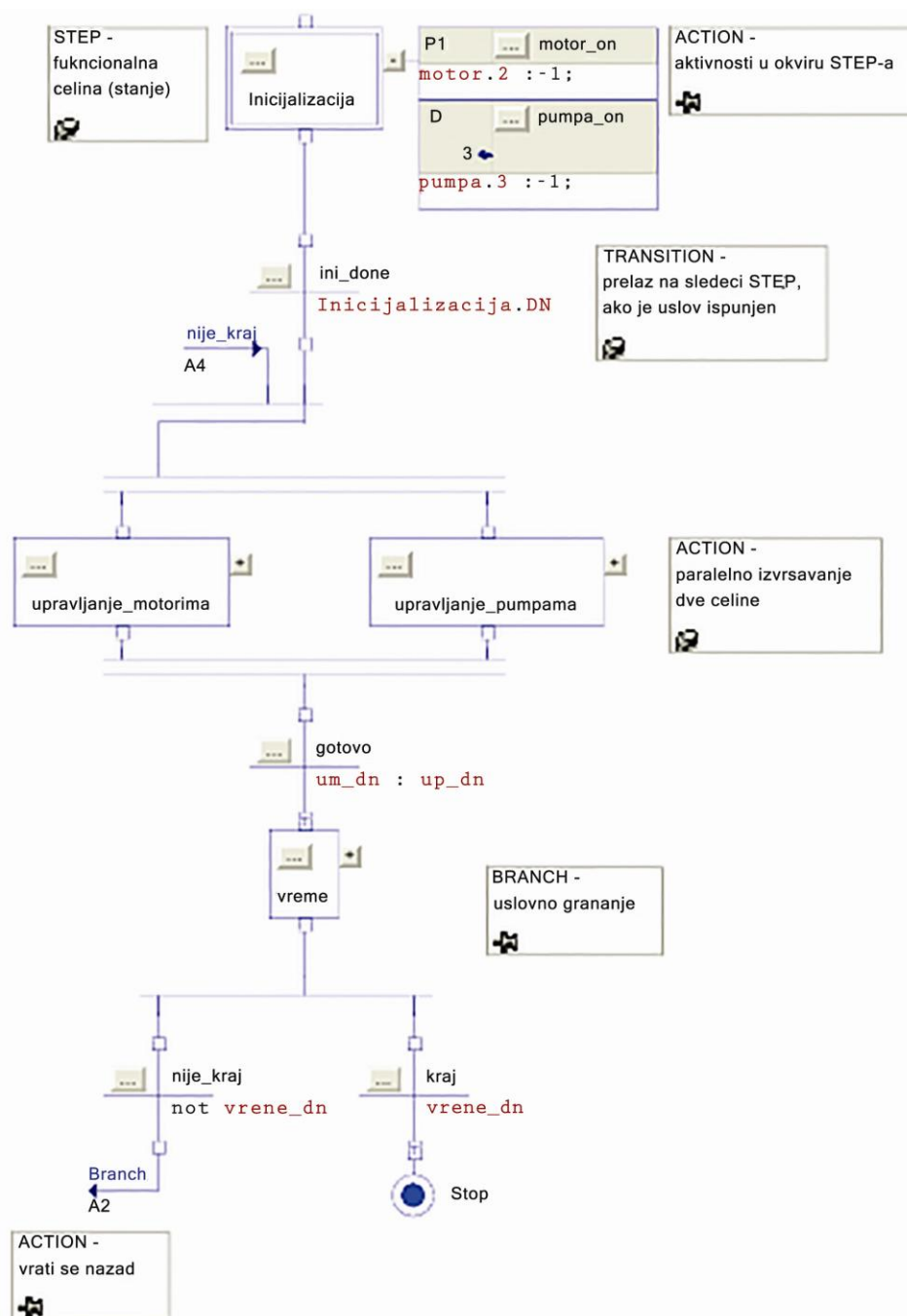
3.4 Дијаграми секвенцијалних функција

Дијаграм секвенцијалних функција (SFC) служи за графичко приказивање процесних стања. Састоји се неколико корака (STEP) у оквиру којих се обављају активности (ACTION). Транзиција у следеће стање (TRANSITION) дозвољена је када се стекну одређени услови. Можемо рећи да се SFC дијаграм битно разликује од осталих програма због тога што се у њему циклично скенирају све активности док је SFC дијаграм активан. То значи да, за разлику од других програма, код кога се циклично скенирају све линије кода, унутар SFC дијаграма постоје делови који се репетитивно скенирају и извршавају, пре него што се пређе на следеће функционалне целине. Отуда се он користи за организацију система у смислу формирања функционалних целина, као и за програмирање управљања системом кроз низ секвенцијалних корака.

Када оперативни систем, ако је SFC континуални задатак или неки други потпрограм започну извршавање SFC дијаграма, SFC програм активира одређен број стања 2 и изврши скенирање активности свих активних стања и враћа контролу оперативном систему. Ако извршавање SFC дијаграма није завршено овај програм остаје и даље активан и наставља да се извршава у складу са раније описаном мултитаскинг филозофијом.

Стање (STEP) представља једну функционалну целину која укључује активности које се одвијају у одређеном интервалу времена, у одређеној фази процеса или у одређеном подсистему. Када се у току извршавања програма дође до блока стања он

постаје активан и све придружене активности се циклично скенирају све док се не испуни услов који означава да процес треба да пређе у следећу фазу (Сл. 3.4). То заправо значи да стање, по правилу, обухвата активности које доводе до промене у систему (довођење неких делова до машинске групе, загревање процеса, додавање састојака, затварање вентила итд). Активности у оквиру стања су засебни програми који се формирају на једном од расположивих програмских језика.



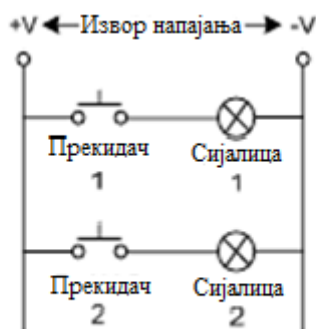
Слика 3.4 Илустрација дијаграма секвенцијалних функција

Данас можемо лако увидети да се на тржишту и у индустрији користи велики број различитих типова PLC контролера који се разликују по изгледу, величини и моћи

обrade, почевши од мањих јединица са малим и ограниченим бројем улаза и излаза до великих, модуларних јединица које се могу конфигурисати за рад са више стотина или чак хиљада улаза/излаза. Међутим, пре појаве савременијих решења PLC контролера за управљање машинама и индустријским постројењима коришћени су механички уређаји, као што су прекидачи и релеи повезани жицама. На Слици 3.5 приказано је једно веома једноставно жичано коло. Приказано коло реализује тривијалну логику : Када је прекидач затворен, сијалица светли, а када је прекидач отворен, сијалица не светли. Два пара прекидач - сијалица су независни зато што су везани паралелно и тако не утичу један на другог. Уколико сада желимо да променимо логику рада кола, тако да, на пример, укључење сијалице 2 искључи сијалицу 1, морамо, најпре, нацртати нову шему везе, а затим, раскинути постојеће везе између прекидача и сијалица и по новој шеми изнова повезати прекидаче и сијалице. Ако се након тога јави потреба да се логика рада поново промени, тако да се, на пример, сијалица 1 укључује само ако су оба прекидача затворена, док се сијалица 2 укључује било којим од два прекидача, коло се мора још једном раставити и поново повезати. Машине и постројења у индустрији поседују много сложенију структуру од овог примера.

Осим, обично, великог броја прикључених улазних и излазних уређаја, управљање оваквим системима треба да обезбеди обављање сложених операција у тачно одређеном редоследу, како и више различитих режима рада. Такав један типичан управљачки систем би захтевао на стотине релеја, повезаних километрима жице смештених у гломазним ормарима.

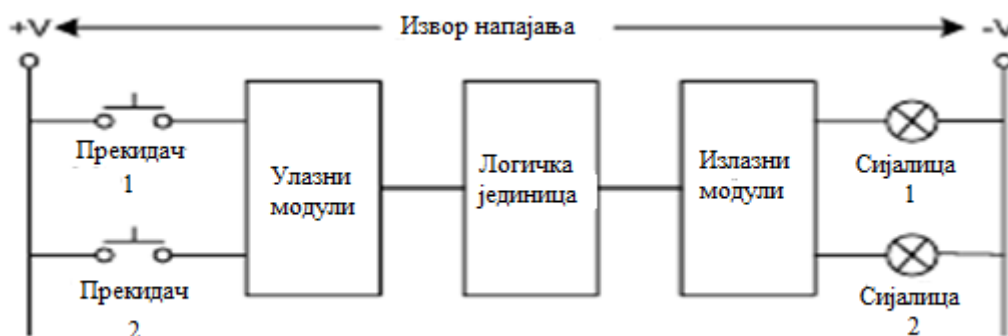
Тако да свака модификација одређених параметара у постројењу би била много компликована и дуготрајна.



Слика 3.5 Релејни систем [4]

Када су се појавили PLC контролери постало је могуће за краћи период заменити целу управљачку јединицу у одређеном сектору производње.

На Слици 3.6 је приказано како би се PLC контролер искористио у претходно разматраном примеру. Може се уочити неколико разлика у односу на првобитно решење. Прво, прекидачи нису директно повезани са сијалицама. Уместо тога, прекидачи су повезани на улазни, а сијалице на излазни модул PLC контролера. Друго, улазни и излазни модули нису директно повезани, већ посредно преко логичке јединице која сходно програму управљања укључује / искључује сијалице у зависности од текућег стања прекидача. На пример, да би се реализовала функција са Сл 3.5, логичка јединица треба бити тако програмирана да генерише побуду за сијалицу 1 за све време док је прекидач 1 затворен, иу исто време, генерише побуду за сијалицу 2 за све време док је прекидач 2 затворен. Програм по коме логичка јединица ради се уноси у меморију логичке јединице преко тастатуре или на неки други начин, где остаје запамћен све док се не унесе нови програм. Очигледно, промена логики рада система, не захтева никакво поновно повезивање и ожичавање.



Слика 3.6 Систем управљања PLC контролером [4]

Код система који користе релејну технику, струја тече од напонског извора кроз прекидач до сијалице, ако је прекидач затворен. Када је прекидач отворен, струјно коло је раскинуто, што прекида напајање сијалице. Код система који се управљају PLC контролером, струја из напонског извора тече кроз прекидач у улазни модул. У улазном модулу се детектује присуство напона и шаље сигнал мале снаге логичкој јединици. Напон на прекидачу и напонски сигнали које улазни модул шаље логичкој јединици су изоловани (галвански раздвојени). Изолација је неопходна зато што микропроцесори раде при малим напонима и струјама У примеру на слици 3.5, логичка јединица прима сигнал од улазног модула када је прекидач 1 затворен и шаље сигнал излазном модулу под условима који су одређени програмом. Конкретно, програм усмерава сигнал на излазни прикључак на који је сијалица 1 повезана увек када од улазног модула прими сигнал да је на прикључку на коме је повезан прекидач 1 детектован напон. Типично

кашњење од тренутка активирања прекидача до тренутка активирања сијалице је реда 5-10мс. Слична активност се обвља када се прекидач 2 активира, стом разликом што се сада сигнал усмерава на прикључак на који је повезана сијалица 2. У почетку, због релативно високе цене PLC контролера, њихова примена је била ограничена на релативно велике управљачке системе, код којих је број замењених релеја могао да покрије цену PLC контролера. Међутим, у новије време, цена PLC контролера је драстично пала, тако да они данас представљају економично решење за највећи број практичних примена.

3.5 Скен циклус

Програми са који користе ледер логику су базирани на релеј логици. У релејној логици сваки елемент у лествици ће се укључити што је пре могуће. Али у програму елемената може бити испитан само један по један у фиксној секвенци.



Слика 3.7. Размена података за време скен циклуса [1]

Скен циклус започиње са улазним скеном у оквиру кога PLC читава садржај улазних линија (регистара улазних модула). Очитани подаци се преносе у одређено подручје меморије-слика улаза. Затим се активира програмски скен у оквиру кога процесор извршава програмске наредбе којима су дефинисане одговарајуће аритметичко - логичке функције. Подаци (операнди) који се користе у програмским наредбама узимају се из меморије и то из подручја означеног као слика улаза (ако су операнди улазни подаци) или из подручја где се смештају интерне променљиве. Резултати обраде се смештају у посебно подручје меморије - слика излаза. Важно је истаћи да се при извршавању програмских наредби подаци не узимају директно са улазних модула, нити се резултати директно постављају на излазне модуле, већ програм размењује податке исључиво са меморијом. По завршетку програмског скена,

оперативни систем PLC контролера активира излазни скен у оквиру кога се подаци из слике излаза преносе на излазне линије (регистре излазних модула). На овај начин ствара се утисак да је PLC све операције дефинисане програмом обавио у исто време. Четврти део скен циклуса - комуникација - намењен је реализацији размене података са уређајима који су повезани са PLC-ом. Након тога, оперативни систем доводи PLC у фазу одржавања у оквиру које се ажурирају интерни тајмери и регистри, обавља управљање меморијом као и низ других послова везаних за одржавање система, о којима корисник и не мора да буде информисан. У зависности од типа уграђеног микропроцесора улазни и излазни скен циклуси извршавају се у времену реда милисекунди (од 0.25мс до 2,56 мс). Трајање програмског скена, свакако зависи од величине програма. [6]

4.0 Основне карактеристике процесорског модула

Процесорски модул садржи централну процесорску јединицу (CPU) и меморију. CPU подржава следеће типове блокова са кодом који вам омогућава да креирате ефикасну структуру за ваш кориснички програм:

- Организациони блок (ОВ) дефинише структуру програма. Неки организациони блокови имају унапред дефинисано понашање и почетне догађаје али постоје и они који се могу прилагодити вашим упутствима у креирању организационе структуре,
- функције (FC) и функцијски блокови (FB) поседују програмски код који одговара специфичним задацима или комбинацији параметара. Свака функција или функцијски блок обезбеђује сет улазних и излазних параметара за дељење података са позивајућим блоком. Функцијском блоку је придружен блок података (DB) у коме се налазе вредности на основу којих се врши егзекуција програма,
- блокови података (DB) складиште податке који могу да се користе од стране програмских блокова.

CPU (Централна процесорска јединица) има три режима рада: STOP режим, STARTUP режим и RUN режим. Статус LED диоде на предњој страни процесора нам приказује у ком је режиму процесор тренутно. У STOP режиму CPU не извршава програм, и можемо у таквом стању да преузмемо пројекат. У STARTUP режиму ОВ се извршава једном. У RUN режиму се скен циклус извршава више пута и тада не можемо да преузмемо пројекат.

CPU чита физичке инпуте непосредно пре извршења корисничког програма и складишти улазне вредности. Ово осигурава да ове вредности остану доследне током извршења инструкција корисника. CPU извршава логику корисничких инструкција и ажурира излазне вредности. После извршења корисничког програма, CPU пише резултирајуће вредности добијене извршавањем задатог програма.

4.1 Организација меморије

Процесор обезбеђује меморијске области за складиштење програма, података и конфигурације. LOAD меморија је неисцрпно складиште за кориснички програм, податке и конфигурацију. Када је пројекат преузет он се прво складишти на LOAD меморијској локацији. Ова област меморије се налази на меморијској картици или на меморији која је уграђена у CPU. На меморијској картици има више простора за складиштење него на оном који је креиран за ту сврху на CPU.

Радна меморија (RAM) је нестабилно складиште за неке елементе корисничког пројекта за време извршавања пројекта. Користи се само за побољшање перформанса система тако што CPU копира неке елементе пројекта са LOAD меморије на радну меморију.

4.2 Програмске датотеке

Програмске датотеке садрже информације о самом контролеру, главни кориснички програм и потпрограме. Свака апликација (процесорска датотека) мора да има следеће три програмске датотеке:

- **System program** - системски програм - садржи различите информације о самом систему као што су тип процесора, конфигурација U / I модула, име процесорске датотеке, лозинку и низ других релевантних података.
- **Reserved** – датотека резервисана за потребе оперативног система.
- **Main ladder program** – главни ледер програм – програм који формира сам корисник и у оквиру кога се дефинише низ операција које PLC треба да изведе.
- **Subroutine Ladder Program** – потпрограми – кориснички потпрограми који се активирају у складу са наредбама за њихово позивање које се налазе у главном програму.

Датотеке података садрже податке који се обрађују помоћу наредби ледер програма. При томе се под појмом подаци подразумевају конвертоване (нумеричке) вредности сигнала који се преко улазно / излазних модула уносе у контролер, или се из контролера преносе на излазне уређаје, као и интерне променљиве које се користе као операнди у различитим операцијама. Датотеке података организоване су у складу са типом променљивих које садрже. То заправо значи да једна датотека садржи само један тип (врсту) података. Једна процесорска датотека може да има највише 256 датотека података.

4.3 Типови података

Типови података се користе за спецификацију како величине елемената података тако и како су подаци протумачени. Свака инструкција параметара подржава најмање један тип података, а неки параметри подржавају више типова података. Формални параметар је индетификатор у инструкцији који означава локацију податка који користе ту инструкцију. Стварни параметар је локација меморије или константа која садржи податке који се користе у инструкцијама. Тип података стварних параметара који ви специфицирате мора да се слаже са типом података формалних параметара који су одређени жељеном програмском наредбом. [5]

Т.1. Типови података [2]

Тип податка	Величина (Битови)	Опсег	Пример константног уноса
Bool	1	0 до 1	TRUE, FALSE, 0, 1
Byte	8	16#00 до 16#FF	16#ABCD, 16#0001
Word	16	16#0000 до 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 до 16#FFFFFFFF	16#02468ACE
Char	8	16#00 до 16#FF	'A', 't', '@'
Sint	8	-128 до 127	123, -123
Int	16	-32,768 до 32,767	123, -123
Dint	32	-2,147,483,648 до 2,147,483,647	123, -123
USint	8	0 до 255	123
UInt	16	0 до 65,535	123
UDint	32	0 до 4,294,967,295	123
Real	32	$\pm 1,18 \times 10^{-38}$ до $\pm 3,40 \times 10^{38}$	123.456, -3.4, - 1.2E+12, 3.4E-3
LREAL	64	$\pm 2.2250738585072020 \times 10^{-308}$ до $\pm 1.7976931348623157 \times 10^{308}$	12345.123456789. -1.2E+40
Time	32	T#24d_20h_31m_23s_648ms до T#24d_20h_31m_23s_647ms сачувано као – 2,147,483648 ms до + 2,147,483648 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_4 5ms
String	Variable	0 до 254 byte-size chacters	'ABC'

Када се специфицира стварни параметар, морамо навести или ознаку (симбол) или апсолутну меморијску адресу. Ознаке повезују име симбола са типом података,

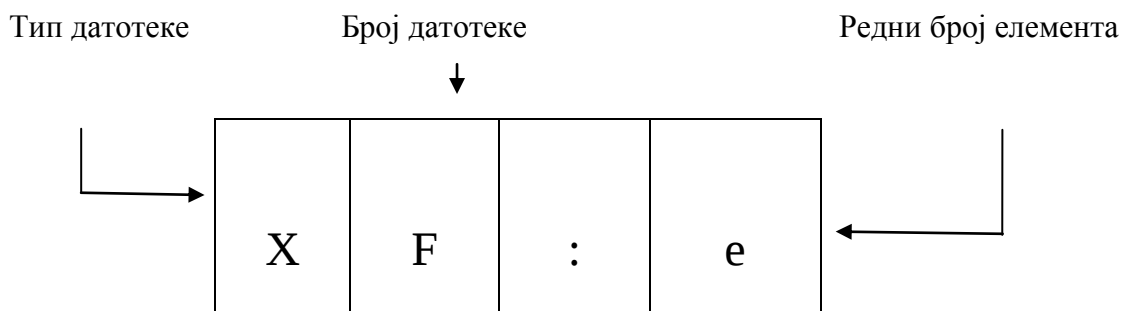
обласћу меморије, опсегом меморије, и коментаром. Ако унесемо апсолутну адресу да није повезана са ознаком, морамо да користимо одговарајућу величину која одговара одговарајућим типовима података, а стандардна ознака ће аутоматски бити креирана на уласку. Такође можете унети константну вредност за многе од улазних параметара. Табела Т.1. описује подржане елементарне типове података, укључујући примере константног уноса.

4.5 Елемент датотеке

Основна јединица датотеке је један елемент. Сваки елемент се састоји из неколико 16 - битних речи. Број речи које чине један елемент зависи од типа датотеке, односно врсте података који се у њу смештају.

Као што је већ истакнуто, подаци који су смештени у датотекама представљају операнде (променљиве) које се користе у појединим програмским наредбама. Ове променљиве се у програму позивају преко својих симболичких имена која представљају логичке адресе. При томе, адресе омогућавају да се позове не само елемент у целини, већ и његов део. То значи да се могу адресирати поједине речи у оквиру елемента или поједини битови у оквиру речи. Будући да су подаци у извесном смислу хијерархијски организовани: 1 елемент садржи неколико речи, а 1 реч 16 битова, то су и одговарајуће адресе структуриране по истом хијерахијском принципу. Поједине речи и битови у неким датотекама имају и придружене акрониме, што додатно олакшава њихово коришћење.

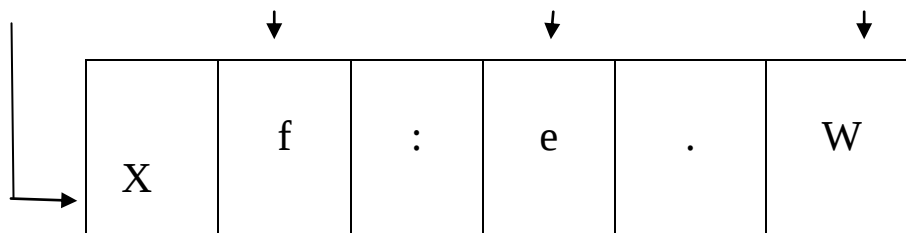
Адреса елемента - у принципу, сваки елемент у оквиру датотеке се идентификује помоћу његовог релативног положаја у односу на почетак датотеке (нулти, први, други, ... елемент). У складу са тиме адреса елемента има изглед као на слици 4.1.



Слика 4.1 Адреса елемента

Адреса речи - једна реч елемента се идентификује или помоћу релативног положаја те речи у оквиру елемента, или помоћу посебног акронима (уколико је исти дефинисан). Формат адресе једне речи приказан је на слици 4.2.

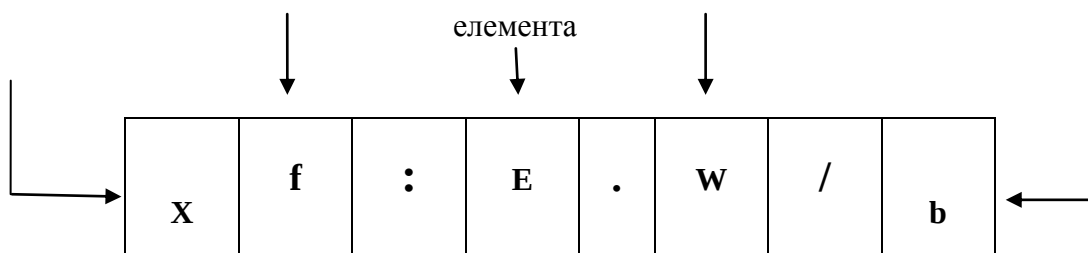
Тип датотеке Број датотеке Редни број елемента Редни број речи
(Или акроним)



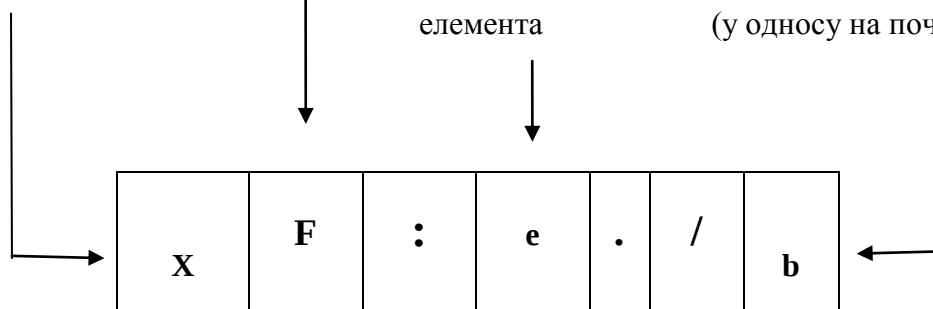
Слика 4.2. Адреса речи

Адреса бита - Један бит у оквиру речи идентификује се или преко његовог релативног положаја у оквиру те речи (нулти, први, други, ... бит бројано с десна у лево) или преко релативног положаја у односу на почетак одговарајућег елемента коме припада реч чији се бит адресира, слика 4.3.

Тип датотеке Број датотеке Редни број елемента Редни број речи Редни број бита



Тип датотеке Број датотеке Редни број елемента Редни број бита
(у односу на почетак елемента)



Слика 4.3. Адреса бита

Дужине елемената у појединим датотекама дате су у табели Т.2. Потребно је истаћи да се теоријски у свакој датотеци могу адресирати и елеменат у целини и његове речи и битови. У табели Т.2. су, међутим, наведене само оне адресе које са аспекта врсте податка и начина организације датотеке имају смисла. Подразумева се при томе да је :

- адресирани бит $0 \leq b \leq 15$
- адресирани елемент $0 \leq e \leq 255$

Може се уочити да су из табели изостављене улазне и излазне датотеке (И и О). Ово је учињено зато што оне донекле одступају од наведеног правила. Наиме, као што ће се касније видети, код ових датотека елементи могу бити дужине од 1 или две речи, што зависи од типа U / I модула. Елеменат ових датотека је заправо одређен слотом у шасији у који се модул поставља.

Т. 2. Величине елемената појединих датотека [2]

Тип података	Дужина елемента	Адресира се
B, N, A	1 реч	Бит “b” - Xf:e/b ;
		Елемент “e” – Xf:e
F	2 речи	Елемент “e” – Xf:e
T, C, R	3 речи	Бит “b” у речи 0 – Xf:e.0/b
		Реч “w” – Xf:e.w ; w = {1,2}
		Бит “b” у речи “w” – Xf:e.w/b
St	42 речи	
		“w” - Xf:e.w – $0 \leq w \leq 41$
		Елемент “e” – Xf:e

У циљу илустрације организације датотека посматрајмо бинарну датотеку – бит фајл. Максимална величина ове датотеке износи 256 елемената. Сваки елеменат је једна 16-битна реч, што значи да ова датотека може имати највише 4096 битова. У складу са изложеним начинима адресирања, један бит може бити адресиран помоћу редног броја елемента (0-255) и редног број бита у оквиру елемента (0-15) или помоћу редног броја бита у оквиру датотеке (0 - 4095), као што је то илустровано у табели Т. 2. Бит означен са X је једанаести бит у другом елементу, односно четрдесетчетврти бит у целој датотеци. Према томе његова адреса је B3: 2/11 или B3/44. Из табеле се такође види да је могуће да се адресира и цео елемент ове датотеке.

T.2. Датотека 3 – Бит [2]

Адреса елемента	Адресе појединих елемената	Data file 3 – Bit file															
		Битови															
		16	15	14	13	12	11	9	8	7	6	5	4	3	2	1	0
B3:0	B3:0/0... B3:0/15 B3/0... B3/15																
B3:1	B3:1/0... B3:1/15 B3:16... B3/31																
B3:2	B3:2/0... B3:2/15 B3/32... B3/47						X										
B3:3	B3:3/0... B3:3/15 B3/48...B3/63																
• • •	• • •																
B3:255	B3:255/0...B3:255/15 B3/4080...B3/4095																

Потребно је да се запази да ће у свакој датотеци типа бит коју корисник сам дефинише адресе бити исте, с тим што ће се уместо броја датотеке (3) ставити број који корисник придружује својој датотеци (9-255).

4.6 Креирање датотеке и заузеће меморије

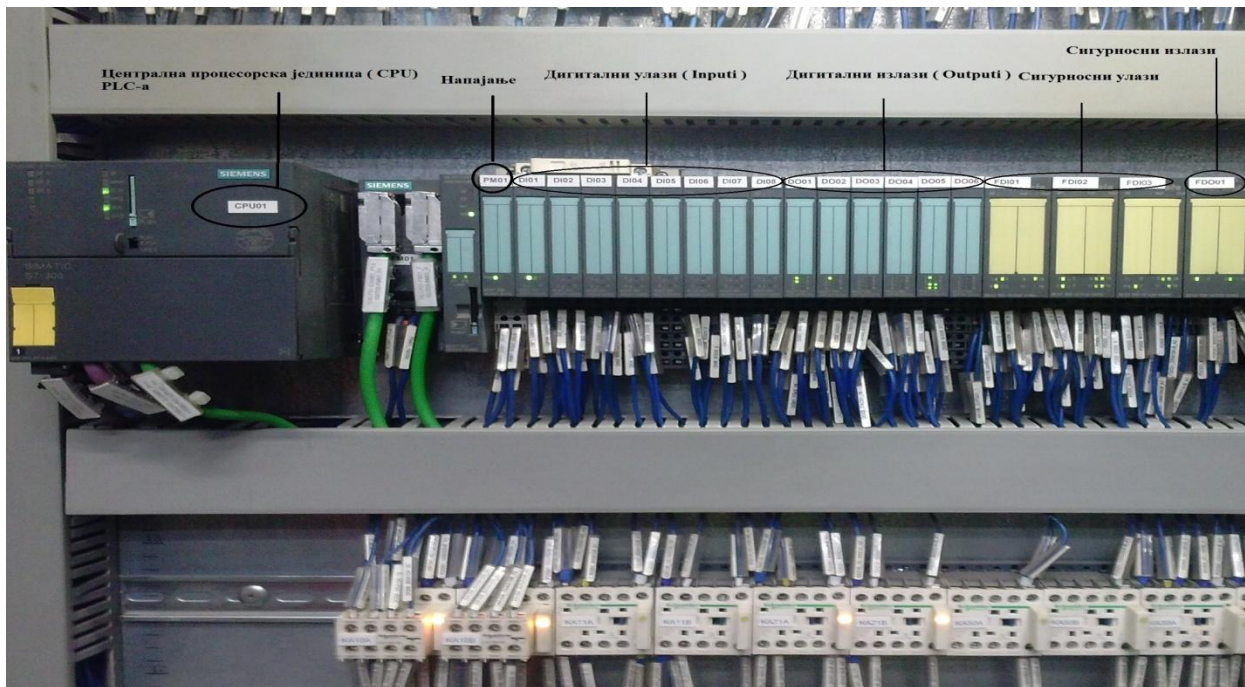
Једна датотека података заузима меморијски простор који обухвата низ суседних речи. Број речи које заузима једна датотека одређен је највећом адресом елемента те датотеке који се користи у програмским наредбама. Наиме, са изузетком датотеке статуса S, која се креира аутоматски, све остале датотеке података не постоје а приори саме по себи, већ се формирају у току креирања програма и то навођењем одговарајуће адресе у програму. Прво навођење броја датотеке иницијализује њено креирање. При томе тип датотеке који је наведен у адреси одређује број речи које се придружују једном елементу, док адреса елемента одређује низ консекутивних елемената за које се

у меморији резервише простор. Тај низ почиње од нуле, а завршава се са адресом елемента која је наведена у наредби. Ако се касније појави већа адреса елемента исте датотеке онда се претходно резервисани простор прошири тако да укључи и ту адресу.

5.0 Систем за аутоматизацију у ауто индустрији за коју је коришћен PLC

Пример система алата за монтажу петих врата

На слици 5.1. можемо видети PLC уређај марке SIEMENS S7 300 који се користи у овој ситуацији.



Слика 5.1. Приказ PLC уређаја марке SIEMENS S7 300.

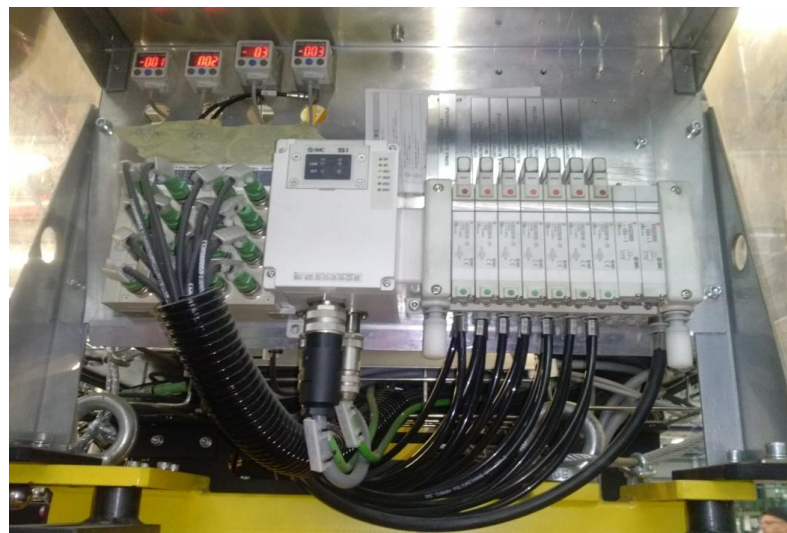
Карактеристике овог модела PLC уређаја су:

- Модуларни мали управљачки систем за најнижи ниво аутоматизације,
- обим који обухвата градацију CPU за различите нивое захтева,
- широк избор модула,
- могућност проширења модулима (до 32 модула),
- сабирнички систем уграђен у модуле,
- могућност повезивања у мрежу помоћу:
 - мултипоинт интерфејс (MPI),
 - Profibus или,
 - индустријски Ethernet.
- централна веза са рачунаром са могућношћу приступа свим модулима,
- нема ограничења на слотове,
- конфигурисање и постављање параметара помоћу алата "HWConfig".



Слика 5.2. Алат за монтажу петих врата.

На слици 5.2. можемо видети алат за монтажу петих врата где су обележени сензори који служе за потврђивање присутности радног елемента и на тај начин шаљу сигнал SMC уређају (слика 5.3.) који шаље ту информацију PLC уређају а затим добија повратну информацију (Feedback) од PLC – а на основу које отвара или затвара пнеуметике за ваздух који иде до клапни, поставља их у радни положај и хвата радни елемент (пета врата) где се даље може вршити монтажа врата на шкољку аутомобила.



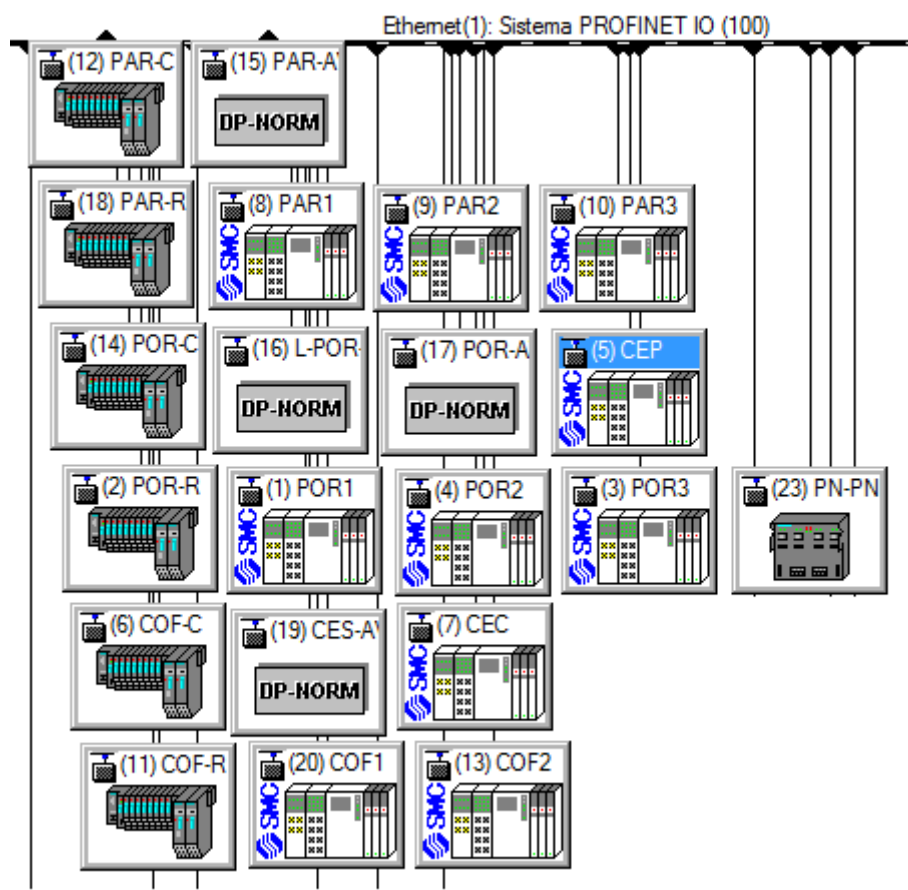
Слика 5.3. SMC уређај



Слика 5.4. Приказ алата у радном (Work) положају

На слици 5.4. можемо видети приказ алата у радном положају.

Да би се остварила цела операција потребно је да се испуне сви услови задати програмирањем PLC-а, овде на пример је потребно да се сви потребни сигнали добију од стране сензора, преко SMC уређаја и PLC-а и ако су сви резултати валидни одређена радња ће бити извршена. Хардверска конфигурација индустријских постројења врло често може бити веома сложена што резултира појавом потребе за новим технологијама које поткрепљују одређене захтеве за лакши, прегледнији, бржи и ефикаснији рад. На следећој слици ћемо представити хардверску конфигурацију горе поменутог система.



Слика 5.5. Хардверска конфигурација PLC система за алат за монтажу петих врата

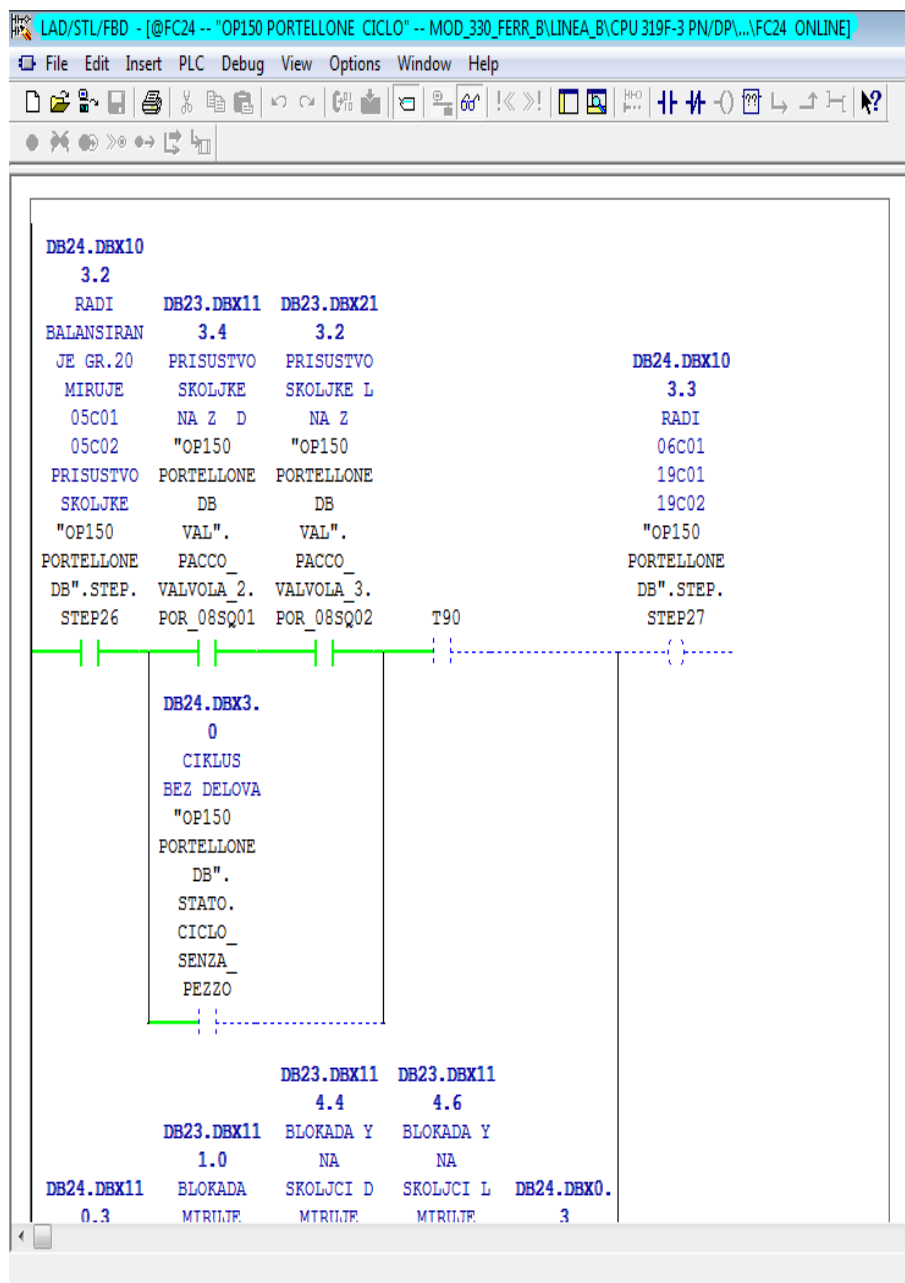
Следеће ознаке представљају PLC уређаје : PAR-C, PAR-R, POR-C, POR-R, COF-C, COF-R.

Следеће ознаке представљају SMC уређаје : PAR1, PAR2, PAR3, POR1, POR2, POR3, CEC, COF1, COF2.

Следеће ознаке представљају Екстерне уређаје : PAR-A, L-POR, POR-A, CES-A.

Ознака PN – PN представља уређај који повезује мање PLC уређаје са главним PLC уређајем.

Цео овај процес можемо најбоље видети кроз концепт једног ледер дијаграма у коме је програмиран овај циклус (Слика 5.6) у програму Siemens simatic step 7.

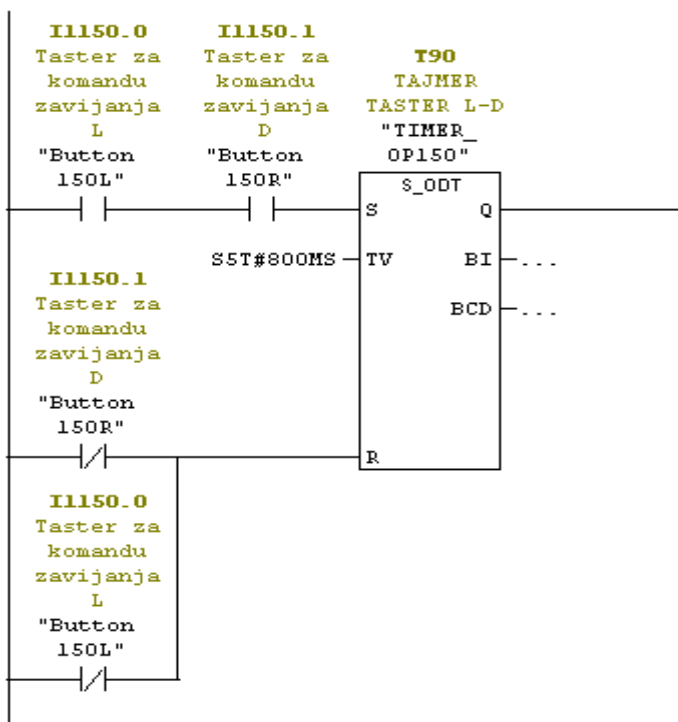


Слика 5.6. Ледер дијаграм у програму Siemens simatic step 7

На овом ледер дијаграму је представљен он лајн циклус рада алата у оном тренутку када су задовољени сви услови за остварење почетка жељене радње монтаже петих врата на шкољку аутомобила. Што значи да су добијени сви сигнали од сензора који се налазе на одређеном месту како би се установило да ли је радни елемент присутан на алату, када се ти сигнали преко SMC уређаја проследе до PLC-а у зависности како је софтвер изпрограмиран PLC шаље одређену повратну информацију SMC – у који затвара или отвара пнеуматике са ваздухом који ако се пнеуматик отвара путује до клампи и ставља их у радни положај за остваривање одређене радње у овом случају монтаже петих врата.

Network 36: TAJMER TASTER L-D

Comment:

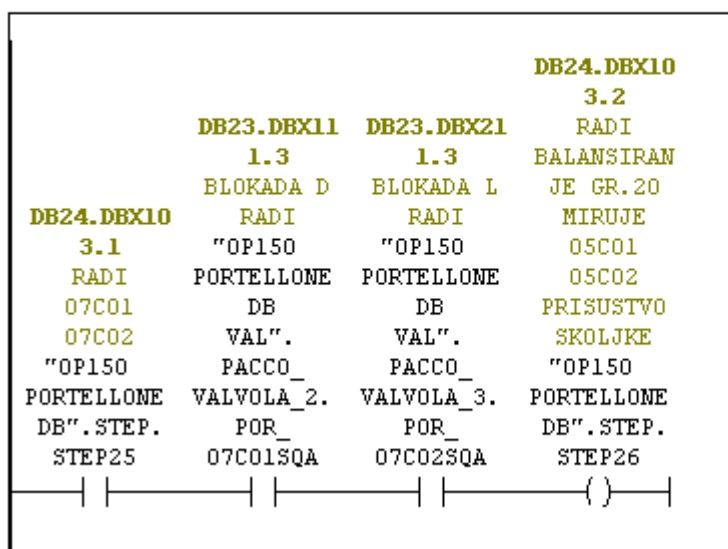


Слика 5.8 Ледер дијаграм у програму Siemens simatic step 7

Тајмер T90 се активира после 800 ms када се добију сигнали од тастера за завијање врата на левој и десној страни а ресетује се у случају да се отпусте леви или десни тастер.

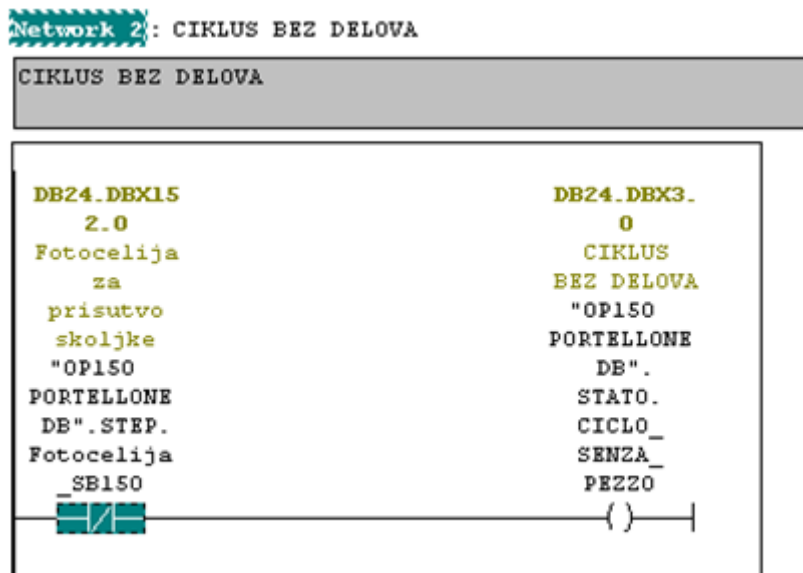
Network 32: RADI BALANSIRANJE GR.20 MIRUJE 05C01 05C02 PRISUTNA SKOLJKA

Comment:



Слика 5.9. Ледер дијаграм у програму Siemens simatic step 7

У овом сегменту можемо видети који су услови потребни да би се испунио корак 26 (Адреса DB24.DBX103.2), а то су испуњен корак 25 (DB24.DBX103.2) десна клампа у радном положају (DB23.DBX111.3), лева клампа у радном положају (DB23.DBX211.3).



Слика 5.9 Ледер дијаграм у програму Siemens simatic step 7

Адреса DB24.DBX3.0 је такозвани бајпас који служи за извршење циклуса без присуства шкољке и тај сигнал добијамо од фотоћелије.

Адреса DB24.DBX152.0 се активира у случају када нема сигнала од фотоћелије за присуство шкољке на станици и активација се добија у овом случају од нормално затвореног контакта.

Такође незаобилазни део овог система јесте и HMI панел слика 5.11.



Слика 5.11. HMI панел

Преко HMI панела се обавља главна дијагностика алата, за случај да на неким од делова алата настане квар детектоваће се у селекцији „Аларми“. Такође је могуће праћење рада циклуса као и мењање режима рада алата на пример ручни или аутоматски. У ручном режиму рада сваки део алата може засебно да се покрене.

6.0 Закључак

Системи управљања процесима у индустријским постројењима могу бити различитог облика као што су нека енергетска постројења, па све до комплекснијих машина. У раном периоду развоја сами људи су обављали управљачке задатке што је отежавало производњу и захтевало доста времена како би се реализовао рад система и ефикасније исправио квар линија у производњи али са појавом PLC уређаја сви процеси везани за отклањање кварова и за аутоматизовање процеса које је изазвано потребом за новим производним задацима је знатно убрзано.

PLC као индустријски рачунар својим дизајном предвиђен је за аутоматизовање процеса производње у индустријским постројењима. Због квалитета своје конструкције отпоран је на разне неповољне услове попут прашине, високе температуре, влаге, вибрације и електромагнетне сметње. PLC уређаји погодни су и за повезивање са централним рачунаром, или са другим рачунарима ради решавања сложенијих задатака, као и управљање процесима са даљине. Могућност комуникације међу PLC уређајима омогућава висок степен искоришћења и координације процеса и велику флексибилност у реализацији управљачких процеса, па самим тим представљају главне предности овог уређаја. PLC системи од свог настанка па све до данашњих дана са повећањем технолошког напредка су у истој линији повећавали свој значај и потребу за њима у индустрији у оном обиму да се данас ни једно савремено индустријско постројење не може замислити без PLC-а, јер су једноставни, не захтевају програмерско знање за инжењере почетнике, погодни су за различите видове модификација што је јако битно ако се ради о промени врсте производа који желимо да направимо што оставља доста простора за унапређење у будућности.

7.0 Литература

Коришћена литература:

- [1] Србијанка Турајлић, Предавање, „Програмабилни логички контролери“, Београд: Електротехнички факултет, 2011.
- [2] Hugh Jack (2007). *Automating Manufacturing Systems with PLCs* преузето са http://claymore.engineer.gvsu.edu/~jackh/books/plcs/pdf/plcbook5_0.pdf (приступило 3.7.2014)
- [3] Др Золтан Јегеш, Милан Аџић, Роберт Мартон, „Управљање применом PLC уређаја“, Суботица: Виша техничка школа, 2005.
- [4] Драган М. Маринковић, „Програмабилни логички контролери – Увод у програмирање и примену“, Београд: Микрокњига, 2012.
- [5] L.A. Bryan (1997), „*Programmable Controllers, Theory and impelementation second edition*“ преузето са http://eelalnx01.epn.edu.ec/bitstream/15000/9275/2/Industrial_Text_and_Video_PLC_Reference.pdf (приступило 3.7.2014)
- [6] Frank D. Petruzella (2005), „*Programmable Logic Controllers third edition*“ преузето са <http://www.msalah.com/A/Atextbook.pdf> (приступило 3.7.2014)