

Универзитет у Крагујевцу
Факултет инжењерских наука

Славољуб М. Радојевић
УПРАВЉАЊЕ НИВООМ ВОДЕ У ДВА
РЕЗЕРВОАРА
мастер рад

Крагујевац, 2013.

Универзитет у Крагујевцу
Факултет инжењерских наука



Назив студијског програма:

Индустријско инжењерство – Пословни информациони системи

Ниво студија: **Мастер академске студије**

Предмет: **Индустријски рачунарски системи**

Број индекса: **606/2011**

Славољуб М. Радојевић
**УПРАВЉАЊЕ НИВООМ ВОДЕ У ДВА
РЕЗЕРВОАРА**
мастер рад

Комисија за преглед и одбрану:

1. Проф. др Милан Матијевић - ментор

2. _____

3. _____

Датум
одбране: _____

Оцена: _____

Задатак мастер рада

У оквиру овог мастер рада кандидат треба да изврши управљање и надзор два одвојена дистрибуирана процеса, коришћењем два ПЛЦ-а *Siemens Simatic S7-1200*.

Задатак је да се реализује укључивање и искључивање пумпи, контрола нивоа воде у резервоарима према задатој вредности и ручна контрола вредности напона на пумпама.

Праћење промена процесних вредности из система ће бити реализовано уписом података у посебан фајл, а визуелни надзор преко ИП камере.

Препоручена литература:

1. Милан Матијевић, Горан Јакуповић, Јелена Цар, *Рачунарски подржано мерење и управљање*, универзитетски уџбеник, Машински факултет у Крагујевцу, 2008
2. Siemens AG , „*S7-1200_System_Manual_enUS*”, 2009.
3. Н. Матић, „*Увод у индустријске PLC контролере*“, Микроелектроника Београд, 2007.

Крагујевац,

ментор:

Проф. др Милан Матијевић

Резиме рада

У овом раду је изведено управљање и надзор два одвојена дистрибуирана процеса.

Оба процеса представљају управљање нивоом воде у два резервоара уз помоћ две пумпе. Два независна процеса су контролисана са два PLC контролера. Надзор и управљање се обавља преко *Touch screen Panel*-а и *PC Station*-а на рачунару. Визуелни надзор се обавља преко ИП камере.

Реализоване су функције укључивања и искључивања пумпи, контроле нивоа воде у резервоарима према задатој вредности и ручне контроле вредности напона на пумпама. Праћење промена процесних вредности из система је реализовано уписом података у посебан фајл.

Кључне речи: Програмибилни логички контролер, ПЛЦ, Спрегнути резервоари, HMI, SCADA, PID управљање, индустријски рачунарски системи, TIA портал

Abstract

This work carried out management and monitoring of two distributed separate processes.

Both processes are the management of water levels in the two tanks with two pumps. Two independent processes are controlled by two PLCs. Supervision and management is done via touch screen panel and PC Station on a computer. Visual inspection is done via IP camera.

These functions were implemented: pumps power on and off, controls of the water levels in the tanks to a given values and the values of manual control voltage at the pumps. Monitoring of process values was implemented by writing data to a file.

Key words: Programmable Logic Controller, PLC, Composite tanks, HMI, SCADA, PID control, industrial computer systems, TIA portal

Садржај

1. Увод.....	5
2. Структура управљачког система	9
2.1. Умрежавање уређаја	9
2.2. Лабораторијски модел спрегнутих резервоара	12
2.3. Појачавач снаге IPS 2M4SA/3	13
2.4. ПЛЦ	14
2.4.1. Начин рада ПЛЦ контролера	19
2.4.2. Програмирање ПЛЦ-а.....	20
2.5. Ладер дијаграм.....	22
2.5.1. Основне инструкције лествичастог дијаграма	22
2.5.2. Бит логичке инструкције	23
2.5.3. Инструкције тајмера	24
2.5.4. Инструкције бројача	25
2.5.5. Инструкције поређења.....	26
2.5.6. Математичке инструкције	27
2.5. ТПанел.....	28
2.6. ИП камера	29
2.7. Рачунар.....	31
3. ТИА Портал.....	32
3.1. Додавање уређаја.....	34
3.2. ПИД контрола.....	35
3.3. ХМИ.....	39
3.4. SCADA	40
4. Извођење задатка	43
4.1. Devices & Networks	43
4.2. PLC_01.....	46
4.3. PLC_02.....	50
4.4. HMI	52
4.5. SCADA	56
4.6. Праћење података.....	60
Литература.....	61

1. Увод

Развој система аутоматског управљања је у многоне допринео убрзаном развоју савремене технологије. Не постоји техничка дисциплина у којој аутоматско управљање није нашло примену. Теорија и техника аутоматског управљања у новије време налази примену и ван подручја чисто техничких дисциплина као на пример биологији, медицини, економији и социологији.

Циљ увођења система аутоматског управљања је обезбеђивање високе продуктивности, квалитета производа и поузданости рада постројења, квалитета живота и заштите здравља људи и животне средине, уз максималну уштеду енергије.

Развој теорије аутоматског управљања се константно среће са две основне потешкоће. Прва је то што се ова теорија чврсто ослања на достигнућа готово свих техничких дисциплина. Због тога се од инжењера аутоматике очекује да познаје не само методе аутоматског управљања, већ и потребне елементе технолошког процеса или објекта управљања који се разматра. Сталним поопштравањем техничких захтева тражи се коришћење прецизних, поузданих и уз то релативно сложених управљачких компоненти, па отуда пракса система управљања захтева ажурно праћење развоја савремене технологије.

Другу потешкоћу представља то што мада представља у суштини инжењерску дисциплину, аутоматско управљање често захтева дубље познавање извесних математичких дисциплина: теорије линеарних, нелинеарних и парцијалних диференцијалних једначина, вишу алгебру, теорију вероватноће случајних променљивих и др.

Ни једна инжењерска дисциплина није ни приближно повезана са обрадом података, синтезом и програмирањем електронских рачунара као што је то случај са аутоматским управљањем у ширем смислу. Наиме, развој теорија аутоматског управљања, система за обраду података и електронских рачунара су се често међусобно допуњавали, условљавали једно друго и учинили да заједно постану императив нашег времена.

Процес, као основна система који је потребно аутоматизовати, према дефиницији представља *“природну операцију која се прогресивно одвија кроз низ постепених промена које следе једна другу на релативно утврђен начин и воде ка одређеном резултату или исходу, или вештачка или вољно изазвана прогресивна операција која се састоји од низа управљаних акција или покрета који су систематично усмерени ка одређеном резултату или циљу”*.

Процес дакле води ка остварењу неког циља. Међутим, независно од врсте појединог процеса, чињеница је да се он никад не налази потпуно изолован од своје околине, која на њега утиче на читав низ разноврсних начина. То заправо значи да ниједан процес, сам по себи, не испуњава у потпуности постављене захтеве. Кад је у питању индустријски процес, своди се на захтев да се повећа продуктивност, смањи утрошак енергије, повећа квалитет итд.

Овако постављени захтеви могу се испунити ако се на неки начин, спољним дејством, **управљањем извршавањем процеса**.

Способност човека – оператора на процесу – да током рада уочава ефекте појединих одлука на процес и да на основу тако стеченог искуства доноси одлуку водила је ка усавршавању вештине управљања. Може се слободно рећи да је управљање у почетку била индивидуална уметност која је почивала на способности оператора да препозна услове под којима процес ради и да интуитивно одабере одговарајућу управљачку акцију.

Временом је постало јасно да квалитет низа производа зависи од тога да ли су у процесу производње одржавани константни температура, притисак, ниво, проток и слично.

При томе, када се једанпут одреди вредност одговарајућих физичких величина, задатак оператора био је сведен само на сталну проверу тих вредности и предузимање увек сличних акција којима би се те величине одржавале на задатим вредностима. Ова чињеница довела је до проналаска **регулатора** – уређаја који пореди вредност неке величине на процесу са задатом вредности и на основу тога генерише управљачку акцију којом се та вредност одржава на заданом (референтном) нивоу. Инсталирањем регулатора посао оператора је знатно поједностављен, али избор правилне референтне вредности и даље зависи од његовог искуства и субјективне оцене о променама које су неопходне за бољи рад процеса.

Развојем и усавршавањем мерне опреме постепено се појавио нови проблем.

Наиме, оператор је одједном био готово "затрпан" обиљем информација које није могао благовремено да обради. Ради помоћи оператору како у организовању, тако и у обради прикупљених информација формиране су посебни **централизовани командни центри** који су опремљени графичким панелима, контролним сијалицама и читавим низом других уређаја за приказивање информација.

Прве идеје о примени рачунара као саставних делова система за управљање процесима се јављају током 50-их година прошлог века. Развој теорије управљања довео је до формулисања низа алгоритама које је било потребно на неки начин и реализовати. У жељи да се оствари извесна флексибилност система који реализује управљање чинило се природним да се тај посао повери рачунарима. Отуда је 1954. године произведен први рачунар намењен управљању процесима који је коришћен за реализацију аутопилота и аутоматско управљање оружјем.

Компанија *Siemens & Halske* 1956. године развија први масовно произведени транзисторски дигитални рачунар – "2002". Назив *Simatic* (замишљен као spoj "*Siemens*" и "*automatic*") је две године касније регистрован у Немачком заводу за патенте као заштитни знак компаније *Siemens*, који остаје до данашњих дана као модеран и ванвременски назив и постао је *Siemens* -ов синоним за аутоматизацију.

Даљи развој рачунарске технологије и појава минирачунара а касније и микрорачунара доводи до праве револуције и у примени рачунара у системима за аутоматизацију.

Током 1969. компаније *Modicon* и *Allen - Bradley* представљају своје прве програмабилне логичке контролере – ПЛЦ-ове.

Према стандардизацији програмабилни логички контролер је дефинисан као:

“Дигитални електронски уређај који користи програмабилну меморију за памћење наредби којима се захтева извођење специфичних функција, као што су логичке функције, секвенцирање, пребројавање, мерење времена, израчунавање, у циљу управљања различитим типовима машина и процеса преко дигиталних и аналогних улазно-излазних модула.”

У основи ПЛЦ је пројектован за рад у изузетно неповољним климо-техничким условима који владају на индустријским постројењима. Он је веома поуздан, једноставан за одржавање и програмирање. ПЛЦ није замишљен као рачунар опште намене, већ као систем чији оперативни систем омогућава да се једноставно и у реалном времену обави аквизиција великог броја података, извесна, не претерано сложена обраду тих података и преношење резултата обраде на извршне органе.

Први ПЛЦ-ови компаније *Siemens - Simatic S30* настају 1973. године где је сваки контролер јединствени производи тј. сваки ПЛЦ је индивидуално конфигурисан и имплементиран у систем. Године 1976. *Siemens* производи и први сигурносни ПЛЦ отпоран на грешке *Simatic XS-S31* за управљање горионцима парних бојлера. Већ 1977. више од 30% електронских контролера уграђених у Немачкој су ПЛЦ-ови, које крајњи корисници могу самостално монтирати и инсталирати. Током 1979. долази до пробоја на тржишту ПЛЦ појављују се контролери *Simatic S5* дизајнирани за системе различитог нивоа перформанси. Од почетка 80-их па до средине 90-их *Siemens* је развијао и побољшавао генерацију *S5* контролера. На Хановерском сајму 1994. Први пут је представљена нова генерација контролера под називом *Simatic S7* која је још увек у масовној производњи.

Развој ПЛЦ-ова, као рачунарских уређаја који се спрежу директно са појединим деловима постројења, неминовно је водио ка развоју дистрибуираних управљачких система. У прво време на врху пирамиде били су минирачунари, док су ПЛЦ-ови имали улогу само аквизиције података и преношења команди. Касније су минирачунари замењени индустријским РС рачунарима, док су ПЛЦ-ови обogaћени сложенијим функцијама. Ово доводи до поновног отварања питања координације и надзора. Развој централизованих контролних соба прати развој технологије и током 60-их настају прве контролне просторије, а током 70-их настају први графички дисплеји са више боја и графичком технологијом. Година 1979. доноси и први систем *Siemens*-ов систем за контролу и управљање процесима. Даљим развојем система за аутоматизацију, крајем 70-их се појављују први *Siemens*-ови аутономни операторски *HMI (Human Machine Interface)* уређаји. Обзиром на количину података са којима су се оператори процеса сретали, отпочело се са развојем телемеријских система за обраду података, истина у потпуно измењеним технолошким условима, што доводи до настанка *SCADA* система (*Supervisory Control and Data Acquisition*). Почетком 90-их долази до експанзије уређаја за визуализацију на нивоу машина, као економичног

решења, за једноставно праћење процеса и појављују се први надзорно-управљачки уређаји на РС платформи. *Siemens*-ов SCADA систем WinCC настао је 1996. на РС платформи под *Windows* оперативним системом. Исте године *Siemens* представља нови концепт аутоматизације назван **Тотално Интегрисана Аутоматизација – TIA** (*Totally Integrated Automation*) који има за циљ превазилажење постојећих граница система:

- Између света рачунара и контролера;
- Између HMI уређаја и контролера;
- Између централизоване и дистрибуиране аутоматизације;
- Између индустријске аутоматизације и процесне аутоматизације.

Ово је постигнуто једном централном особином - интеграцијом свих компоненти аутоматизације у:

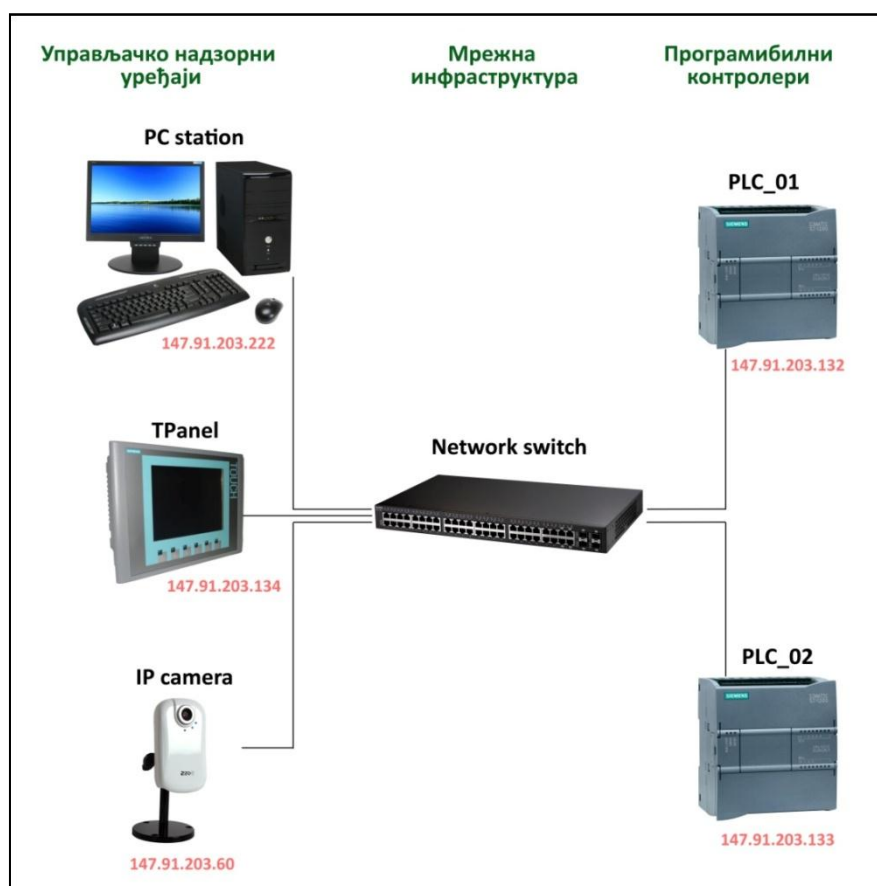
- Конфигурисању и програмирању,
- Образи података,
- Комуникацији.

Као део TIA концепта 1997. године појављује се потпуно нови систем за управљање процесима PCS 7 базиран на стандардним *Simatic* компонентама. Обзиром на све веће захтеве у визуелизацији процеса, појавом екрана осетљивих на додир (*touch panel*), панела у боји високе резолуције, *Siemens* 2004. представља софтвер за конфигурисање свих *Simatic* HMI уређаја и мањих SCADA система WinCC flexible, флексибле као наследника дотадашњег софтвера ProTool-a. WinCC flexible својим модуларним и иновативним дизајном и једноставношћу полако осваја тржиште.

2. Структура управљачког система

2.1. Умрежавање уређаја

На слици је приказана мрежа уређаја који су коришћени за извођење задатка. Сви уређаји са слике имају мрежни модул са припадајућом МАК адресом и додељеним ИП адресама из мрежног окружења. Преко УТП кабла спојени са мрежним разделником у мрежу типа звезде.



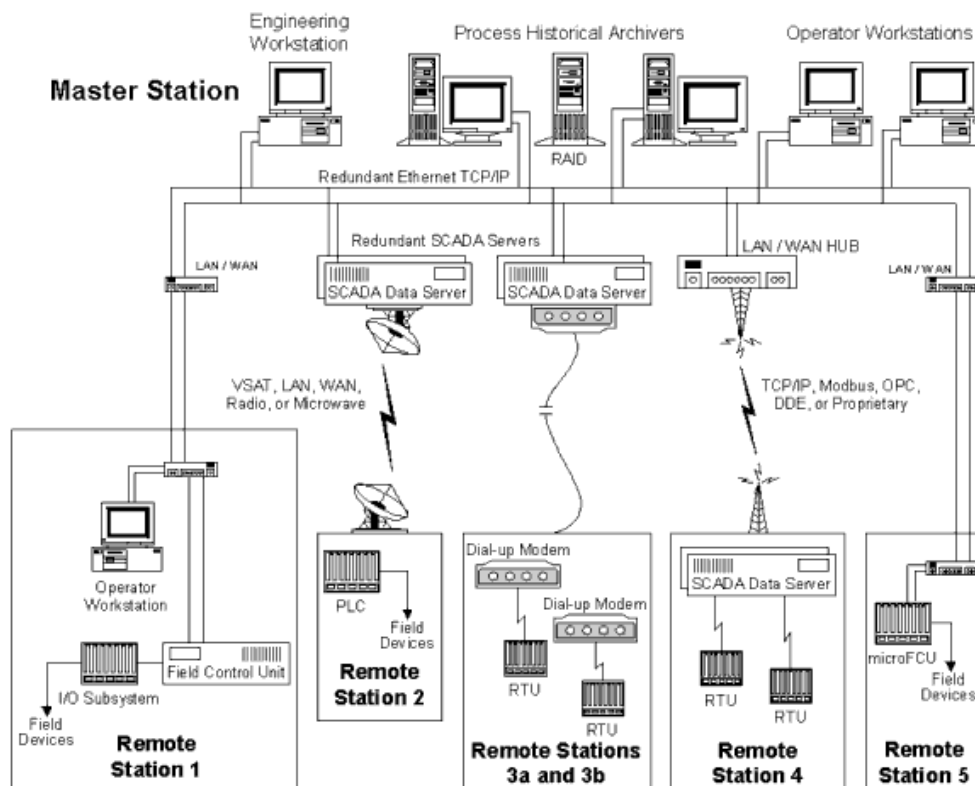
Слика 1. Мрежа уређаја

Сваком уређају је додељена ИП адреса тако да преко мреже може да се приступи, репрограмира, прати, апгрејдује. Адресе су распоређене по следећем:

- 147.91.203.60 – ИП камера
- 147.91.203.132 – ПЛЦ 1
- 147.91.203.133 – ПЛЦ 2
- 147.91.203.134 – Touch Panel KTP600
- 147.91.203.222 – PC Station

Додељивање статичких ИП адреса за ПЛЦ и ТPanel се изводи у Siemens TIA Portal-у, док се додељивање ИП адреса за РС и ИП камеру изводи преко њихових сетап интерфејса.

У овом задатку је мрежна инфраструктура представљена са једним мрежним разделником (Network Switch), као у лабораторијским условима. Међутим, мрежна инфраструктура може бити још више разграната (нпр. програмибилни контролери се могу налазити у једном граду, а управљачке конзоле у другом).

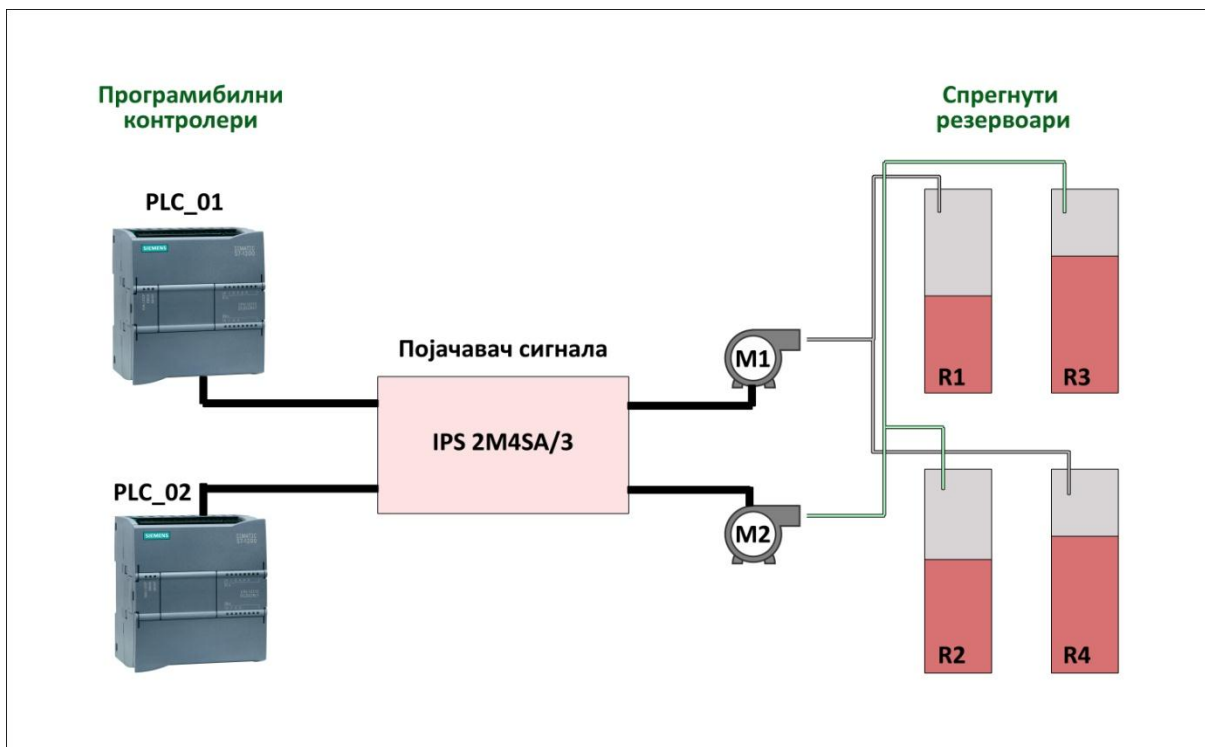


Слика 2. Пример разгранате мрежне инфраструктуре и уређаја

У оваквим разгранатим системима користе се свакојаки мрежни ресурси различитих врсти и типова преноса података (свичеви, хабови, рутери, модеми).

У основи ових уређаја је TCP/IP протокол који садржи информације о адресирању, чиме се постиже да сваки мрежни уређај (рачунар, сервер, радна станица, интерфејс рутера) који је повезан на мрежу има јединствену адресу и може се лако идентификовати, а исто тако садржи контролне информације које омогућују пакетима да буду прослеђени (рутирани) на основу познатих IP адреса.

Програмибилни контролери су преко појачавача сигнала (мултиплексера) везани за модел спрегнутих резервоара, као што је приказано на слици.



Слика 3. Веза програмилних контролера и резервоара

2.2. Лабораторијски модел спрегнутих резервоара

За управљање нивоом течности у резервоарима, коришћен је лабораторијски модел 4 спрегнута резервоара. Ради се о производу фирме “*Majk Elektronik*” из Младеновца.



Слика 4. Изглед спрегнутих резервоара

Лабораторијски модел се састоји од четири издужена резервоара унутрашњих димензија 40x40x300mm окачена изнад сабирног резервоара димензија 14x37x10cm који може да прихвати запремину од 5L течности (служи за скупљање воде). Из сабирног резервоара се вода упумпава са две независне пумпе центрифугалног типа улазног једносмерног напона у опсегу од 0V до 12V.

Сваки од резервоара на дну има отвор. Вода из горњих резервоара отиче у резервоар на доњем нивоу, а из доњих резервоара вода отиче у сабирни резервоар. На тај начин, резервоари су међусобно спрегнути, а пумпе зависно од позиције ручних вентила, пумпају воду у сва 4 резервоара са различитим протоком.

Улазни напон једносмерни пумпе је у опсегу од 0-12 V и као резултат даје опсег протока 0-20 cm³/s (улазно излазни опсег пумпе је део њене статичке

карактеристике). Унутрашње димензије резервоара су 40x40x300 mm. Давач нивоа у сваком од резервоара је давач хидростатичког притиска у резервоару који на свом излазу даје напонски сигнал у опсегу од 1-5 V. По спецификацији се ради о мерном претварачу линеарног типа коефицијента појачања 0.1 V/cm.

Можемо рећи да је улазни сигнал опсега 0-12 V, а излазни 1-5 V (што одговара опсегу нивоа течности у резервоару од 0-30 cm). Улазни или управљачки сигнал може долазити са D/A интерфејса који продукује аналогни напонски сигнал различитог опсега 0-5 V (попут NI USB 6008 или 6009 картице) или 0-10 V (попут Siemens PLC S7-1200 D/A излаза). Отуда се у лабораторији користи и посебно пројектовани уређај **IPS 2M4SA/3** чија је улога да сигнал са дигитално аналогног конвертора рачунарског подсистема опсега 0-5 V или 0-10 V конвертује у опсег 0-12 V који најбоље одговара карактеристикама пумпе.

2.3. Појачавач снаге IPS 2M4SA/3

IPS 2M4SA/3 је појачавач снаге и електрични прекидач који је посебно пројектован за потребе експерименталних истраживања у Центру за примењену аутоматику, детаљно је описан у [2].



Слика 5. Појачавач снаге IPS 2M4SA/3

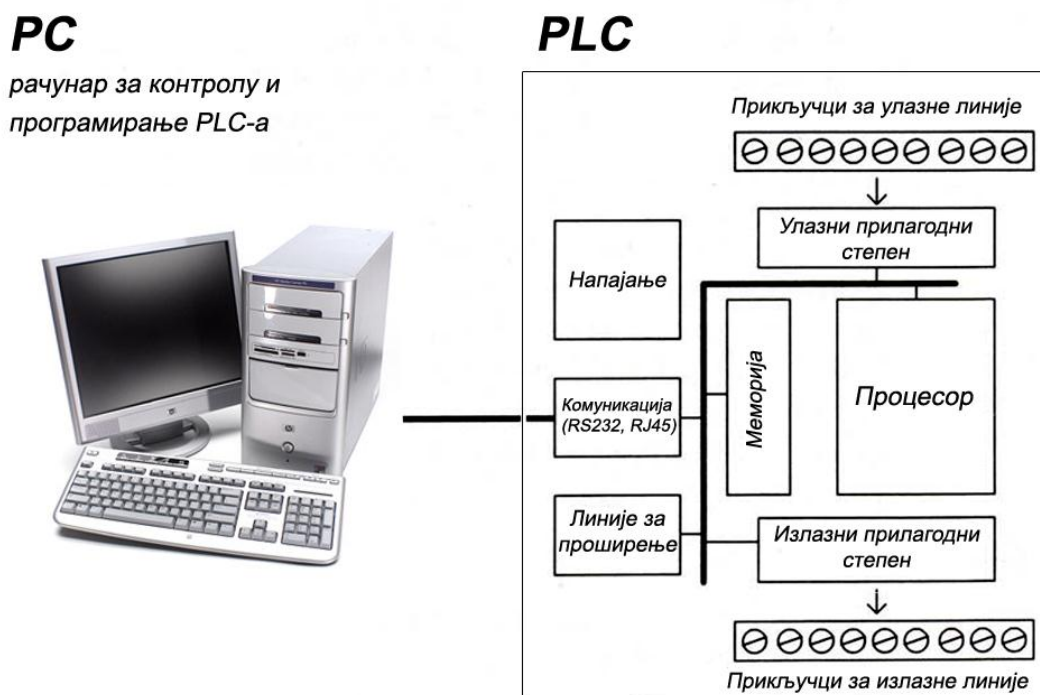
Овај уређај има могућност хардверског преласка на ручно управљање, пребацивањем положаја прекидача из аутоматског у ручни мод, и тада се аналогни излаз (ка пумпи) мења потенциометром.

2.4. ПЛЦ

PLC (Programmable Logic Controller) – програмабилни логички контролер је дигитални електронски уређај који оседује програмабилну меморију за смештање инструкција којима се реализују специфичне функције, као што су логичке операције и секвенцирање, тајминг, бројање и аритметичке операције, у циљу управљања различитим машинама и процесима путем дигиталних или аналогних улазно/излазних (I/O) модула.

Индустријски контролер (PLC) се може дефинисати и као индустријски рачунар чију су хардвер и софтвер посебно прилагођени раду у индустријским условима, а који се може лако програмирати и уграђивати у нове и постојеће индустријске системе.

Блок шема типичних компоненти од којих је начињен контролер дат је на слици.



Слика 6. Блок шема везе рачунара и ПЛЦа

Програмска јединица је обично рачунар који се користи за писање програма.

CPU (Central Processing Unit) – централна процесорска јединица је мозак контролера. То је обично неки од микроконтролера, раније су то били 8-битни, а сада су 16-битни и 32-битни микроконтролери. ЦПУ се брине о комуникацији, међусобној повезаности осталих делова контролера, извршавању програма, управљању меморијом, надгледањем улаза и постављањем излаза.

Меморија – Системска меморија користи се од стране контролера за оперативни систем. Поред оперативног система у њој се налази и кориснички програм преведен из ледер дијаграма у бинарни облик.

Корисничка меморија је подељена на блокове који имају посебне функције. Неки делови меморије користе се за чување стања улаза и излаза (имају одговарајући бит у меморији), други делови меморије користе се за чување садржаја променљивих који се користе у корисничком програму.

Напајање – Електрично напајање се користи за довођење електричне енергије до централне процесорске јединице. Већина ПЛЦ контролера ради или на 24V DC или на 220V AC. Обично се не користи за покретање спољашњих улаза и излаза.

Улаз у ПЛЦ – Служи за прихват сигнала са процеса. Мањи контролери поседују само дигиталне улазе, док већи имају и аналогне улазе. Сензори се најчешће користе као улази у контролере, али то могу бити и роботи, видео системи итд.

Излази из ПЛЦ – Служе за повезивање контролера са излазним уређајима којима се управља. Могу бити дигиталног и/или аналогног типа у зависности од самог ПЛЦ контролера.

Линије за проширење – Сваки контролер има ограничен број улазно/излазних линија. Уколико је потребно тај број се преко одређених додатних модула може повећати проширењем система преко линија за проширење.

SIMATIC S7-1200 PLC је заснован на модуларном концепту и лако се прилагођава свим једноставним и оним мало захтевнијим апликацијама. SIMATIC S7-1200 је PLC је контролер који је модуларан и компактан, разноврстан, па се самим тим савршено уклапа у мноштво разних апликација. Скалабилан и флексибилан дизајн, комуникацијски интерфејс који употпуњује највише стандарде у индустријској комуникацији, као и мноштво напредних функција чине овај контролер саставним делом комплетог и свеобухватног управљачког система.



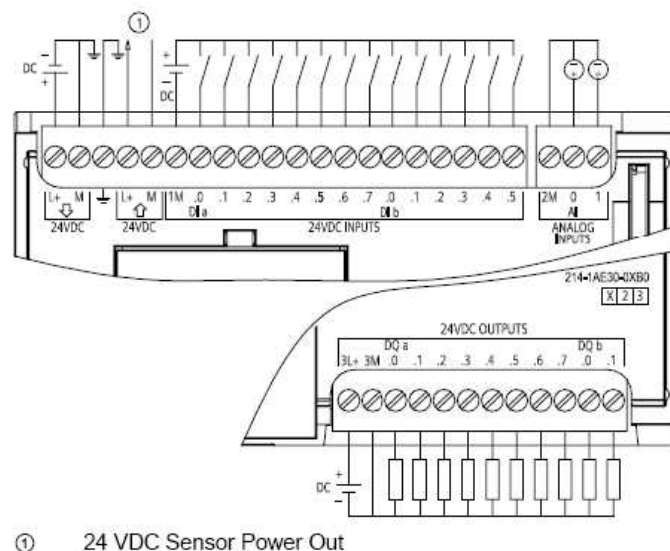
Слика 7. S7-1200 - Комуникациони модул, CPU, профинет и сигнални модули

Код новог SIMATIC-а (PLC S7-1200), SIEMENS је развио посебно добру интеграцију између PLC-а, HMI-а и софтвера. Резултат тога је флексибилност и скалабилност S7-1200 PLC-а, са којим се могу решавати и неки од најзахтевнијих процеса на машинама и производним линијама. SIMATIC S7-1200 PLC CPU јединице подржавају мноштво различитих улазно/излазних модула и комуникацијских картица. S7-1200 CPU се састоји од интегрисаног PROFINET интерфејса чиме се постиже једноставно умрежавање између PC-софтвера, контролера и HMI-а. Основне карактеристике су:

- Једноставан и ефикасан софтвер
- Скалабилан и флексибилан дизајн
- Интегрисани PROFINET интерфејс
- Једноставно умрежавање
- Комуникација са екстерним уређајима
- Интегрисане многе потраживане технологије
- 100kHz улази за бројање и мерење
- 100kHz излази за позиционирање, регулацију брзине и цикличке операције
- PLC- готови функцијски блокови за управљање кретањем
- PID инструкције у затвореној петљи
- Максимум 284 I/O

SIMATIC S7-1200 може да садржи неку од три различите централне процесорске јединице CPU 1211C, CPU 1212C и CPU 1214C. У најкомплекснијој конфигурацији коју обезбеђује процесорска јединица CPU 1214C може се надоградити 8 сигналних модула и три комуникациона модула, чиме се добија конфигурација са укупно максималних 284 дигитална улаза/излаза (I/O), као и са укупно максималним 51-ним аналогним улазом/излазом (I/O). Оно што представља новину код архитектуре SIMATIC S7-1200 је мали хардверски додатак под називом сигнал боард, помоћу кога се на једноставан и брз начин било која процесорска јединица може проширити мањим бројем дигиталних и аналогних улаза и излаза.

При имплементацији, софтвер SIMATIC STEP 7 Basic одао је нове „тајне“ контролера S7-1200. Приступ програмирању својствен серијама S7-300 и S7-400 у потпуности је задовољио потребе објектно оријентисаног кода. У односу на претходника, серију S7-200, меморија је организована по принципу дата блокова што омогућује додатну флексибилност у параметризацији склопова који се понављају.



Слика 8. Основна конфигурација улаза и излаза PLC S7-1200

На врху слике, са леве стране се могу видети шематски прикази за напајање контролера. На његове напонске пинове је потребно довести 24V једносмерне струје (20.4 до 28.8 VDC је толеранција напајања PLCа према техничкој документацији). Десно од њих се налазе пинови дигиталних улаза. Дигитални улаз по наслеђеној „PLC“ терминологији може имати вредности 0 или 1 које се уписују у бит улазног регистра. Сваки бит улазног регистра је повезан са одговарајућим физичким дигиталним улазом и ради се о томе да ли је напон од 24 VDC регистрован или није регистрован на улазу. Тачније, логичка јединица одговара стању улаза од 15 VDC на 2.5 mA – минимално, па до 30 VDC максимално, док је 24 VDC на 4 mA – номинално. Логичка нула подразумева до 5 VDC на 1 mA – максимално. Међустања не мењају вредност улазног бита. Време у коме се деси читање свих улаза (уписивање физичких стања на одговарајуће меморијске локације), извршење програмског кода, и ажурирање свих излаза је време једног скен циклуса (ово време је у принципу варијабилно). Потребно је напоменути да контролер има 14 дигиталних улаза који номинално треба да региструју напонске нивое 24V или 0V. Ових 14 улаза имају адресе од I0.0 (први улазни бит), I0.1, ..., I0.7 (осми улазни бит, или последњи улазни бит првог улазног бајта), I1.0 (први улазни бит другог улазног бајта, или девети PLC дигитални улаз)... I1.5 (шести улазни бит другог улазног бајта, или четрнаести PLC дигитални улаз). Практично, дигитални улази могу да мере дискретна стања у спољном свету на нивоу укључно/искључено, објекат јесте/није у одговарајућем положају или стању (напр. рампа за пакринг и слично). Такође, у горњем десном углу слике 7 означена су и 2 аналогна улаза које има овај PLC који могу мерити аналогне улазе опсега од 0 до 10 VDC. Вредности аналогних улаза се уписују у 16-битне регистре AI0 и AI1. Међутим, за упис аналогне вредности (0-10V) у 16-битне улазне регистре AI0 и AI1 се не користи целокупан меморијски капацитет регистра већ само део, и тако (како је већ дато у техничкој спецификацији) се уписују вредности од 0 до 27648 у те регистре а које одговарају стањима од 0-10V на аналогном улазу. Треба запазити да је присутно квантовање по нивоу (уноси се изванштан тзв. шум квантовања услед АД конверзије аналогне вредности улаза). У доњем делу слике се налазе дигитални излази којих има 10. повезани су са меморијским локацијама на адресама Q0.0...Q0.7, Q1.0 и Q1.1. Упис логичке јединице у рецимо бит Q0.0 (први бит излазног регистра) значи да се на дигиталном излазу реализује напонско стање од 24V. Практично, преко дигиталних излаза можемо

укључити или искључити електрична кола која контролишу рад предметних објеката управљања (подићи или спусти рампу, и слично). PLC има (лево од дигиталних излаза приказаних на слици) и један аналогни излаз опсега од 0V до 10V једносмерне струје, а који је адресиран са AQ0. Супротно аналогном улазу, овде уписом вредности од 0 до 27648 у регистар са адресом AQ0, на аналогном излазу се појављује напон од 0V до 10V. Горе је описана основна улазно/излазна конфигурација PLC SIMATIC S7-1200, а која може бити проширена додатним модулима. Зависно од изабране централне процесорске јединице (у овом случају CPU 1214C) различити број улазно/излазних периферија може бити придодат основној конфигурацији.

Генерално, навођењем меморијске адресе омогућен је директан приступ подацима. Да би се приступило биту у делу меморије, наводи се адреса, која садржи идентификатор меморијског места, адресу бајта и број бита. Приступање биту назива се и 'byte.bit' адресирање. У следећем примеру меморијско место и адресу бајта (I=input,3=byte 3) од адресе бита (bit 4) одваја тачка ('.').

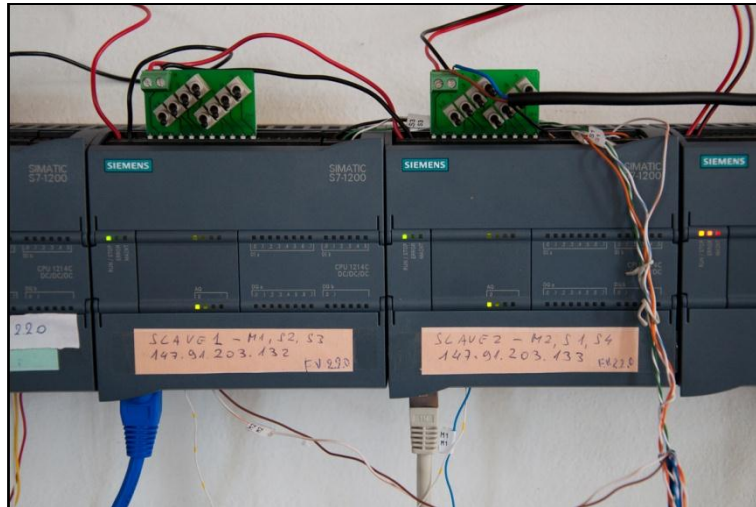
Сваком улазном/излазном уређају се додељује адреса. На овај начин се резервише место где ће бити похрањена логичка вредност која дефинише стање тог уређаја (1 када је уређај укључен, а 0 када је искључен).

На почетку сваког циклуса скенирања S7-1200 узоркује физичко место улаза и записује ове вредности у улазни регистар. Улазном регистру се приписује словна ознака 'I' и приступа са: I[адреса бајта].[адреса бита], нпр. I0.4. На крају овог циклуса скенирања, S7-1200 копира вредности смештене у излазном регистру на физичко место излаза. Излазном регистру се приписује словна ознака 'Q' и приступа са: Q[адреса бајта].[адреса бита], нпр. Q1.1.

Словне ознаке за остале меморијске локације су:

- V за променљиву меморију; складишти посредно податке операција самог програма на контролеру
- M за бит меморију; користи се као контролни релеј за складиштење посредног стања операција или других контролних информација
- T за меморију тајмера; S7-1200 обезбеђује тајмере који броје у инкрементима времена у резолуцијама од 1ms, 10ms или 100ms. Са тајмером су повезане две променљиве: тренутна вредност (16-то битна вредност која чува одбројано време) и бит тајмера (овај се бит сетује или чисти као резултат поређења тренутне вредности и унапред задате вредности)
- S за меморију бројача; S7-1200 обезбеђује три врсте бројача који броје сваки ниже-на-ниже прелазни догађај на улазу бројача: један броји само на више, други на ниже, и трећи броји и на више и на ниже. Постоје две променљиве, а аналогија је иста као и код тајмера
- HC за брзи бројач: броји брзе догађаје независно од циклуса скенирања
- AC за акумулаторе: акумулатори су читај/пиши уређаји који се могу користити као меморија
- L за локалну меморију: локална меморија је слична V меморији са разликом да је V меморија глобална (може да јој се приступи из било ког дела програма: главног програма, подпрограма или рутине прекида)
- AI за аналогне улазе: аналогне улазе (температура, напон,...) S7-1200 претвара у 16-битне дигиталне речи

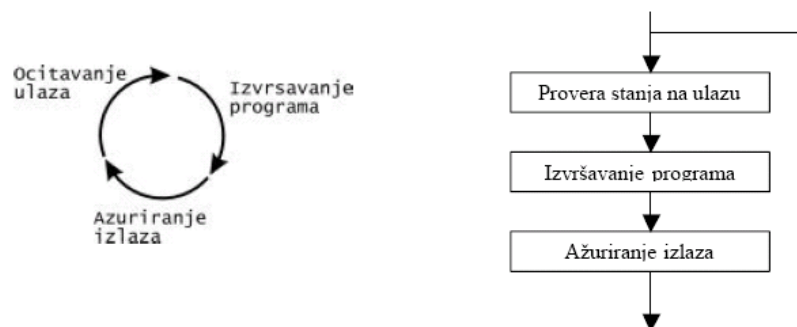
- AQ за аналогне излазе: S7-1200 претвара 16-битне дигиталне речи у излазни напон одговарајуће вредност, пропорционално дигиталној вредности
- S за меморију секвенцијалног контролног релеја (SCR); SCR или S битови се користе да организују операције машине или кораке у једнаке програмске сегменте



Слика 9. Приказ ПЛЦ-а на шини са прикљученим улазима и излазима

2.4.1. Начин рада ПЛЦ контролера

Основа рада ПЛЦ контролера заснива се на континуалном скенирању програма. Под скенирањем се подразумева пролаз кроз све услове у неком одређеном времену.



Слика 10. Скенирање улаза

Скенирање се састоји из три основна корака:

1. Први корак – Провера стања на улазу

Прво, ПЛЦ прегледа стање на сваком улазу (вредност бита на адреси сваког улазног уређаја) да би "знао" да ли је укључен (1) или искључен (0). Другим

речима, проверава да ли нпр. Сензор прикључен на први улаз проводи. После првог улаза провера други, па трећи итд. Ове податке уписује у меморију и користи их у другом кораку.

2. Други корак - *Извршавање програма*

После завршеног првог корака ПЛЦ извршава програм, инструкцију по инструкцију. Ако је на пример, програм написан да када је неки одређени улаз укључен (1) укључује се неки излаз. Пошто из првог корака ПЛЦ чува податке, на основу њих "одлучује" које ће излазе укључити, а које не. Подаци за извршавање се похрањују у меморију и користе у следећем кораку.

3. Трећи корак – *Ажурирање излаза*

ПЛЦ ажурира статус излазних уредаја, на основу вредности улаза и извршавања програма – битови на адресама излазних уредаја се сетују (додељује им се вредност 1) или ресетују (додељује им се вредност 0).

По завршетку трећег корака ПЛЦ се враћа на први корак итд. Време скенирања (*scan time*) се дефинише ако време потребно за извршавање ова три корака. Оно је променљиво, зависи од броја улаза и излаза, дужине програма.

По укључењу напајања ПЛЦ се прво иницијализује (брисање одређених меморијских области, ресетовање системских тајмера, провера улазно-излазних прикључака) и уколико нису детектоване грешке, процес почиње циклично да се извршава.

2.4.2. Програмирање ПЛЦ-а

Математичка основа за секвенцијално управљање је *Булева алгебра*. Ако логички захтеви који се постављају могу бити специфицирани у форми Боолеових једначина тада програмабилни логички контролер (ПЛЦ) постаје рачунар који решава Боолеове једначине. Према *Prema IEC (International Electrotechnical Commision)* ПЛЦ се дефинише као: "... дигитални електронски систем, пројектован да се користи у индустријском окружењу који употребљава програмабилну меморију за интерно меморисање, ускладиштење кориснички оријентисаних инструкција за имплементирање специфичних функција таквих као што су: логичке, секвенцијалне, бројачке, тајминг и аритметичке до управљачких, кроз дигиталне или аналогне улазе за управљање разних типова машина и процеса. ПЛЦ и њему придружени периферни уредаји пројектовани су тако да буду лако уградиви у индустријски систем управљања и да омогућавају laku употребу својих функција."

Најраспрострањеније програмирање ПЛЦ-а је помоћу *ladder* дијаграма, који се потом преводе у одговарајуће Боолеове једначине, које служе за извршавање задатих логичких функција. Због тога је битно подсетити се основних теорема и логике саме Боолеове алгебре.

Основне теореме Боолеове алгебре:

$$A + B = B + A$$

$$AB = BA$$

$$A + 0 = A$$

$$A \cdot 0 = 0$$

$$A + 1 = 1$$

$$A \cdot 1 = A$$

$$AA = A$$

$$A + \bar{A} = 1$$

$$A\bar{A} = 0$$

$$\overline{\overline{A}} = A$$

$$(A + B) + C = A + (B + C)$$

$$(AB)C = A(BC)$$

$$AB + AC = A(B + C)$$

$$(A + B)(A + C) = A + BC$$

$$A + AB = A$$

$$A + \bar{A}B = A + B$$

$$A(A + B) = A$$

$$\overline{AB} = \bar{A} + \bar{B}$$

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

Променљиве у Боолеовој алгебри могу имати само вредности "0" и "1". Логичке функције које садрже операције AND, OR, EXCLUSIVE OR, NOT, NAND, NOR, итд. лако се изражавају коришћењем Боолеове алгебре и помоћу њих могу бити формиране једначине које садрже Боолеове променљиве и операције. Ако прекидачко-логичке функције треба да буду оствариване ПЛЦ-ом, оне могу бити изражене у форми Булеових једначина, а дигитални рачунар унутар управљачке јединице ПЛЦ може бити употребљен да реши ове једначине у већини случајева на исти начин као и дигитални рачунар опште намене када решава алгебарске једначине. Вредности непознатих променљивих се могу одредити из вредности познатих. Размотримо пример два прекидача PR1 и PR2 чије су променљиве стања одређене са стањем (ON=1, OFF=0) прекидача који су повезани на ПЛЦ. LAMP је променљива чија ће вредност бити употребљена да одреди стање индикатора сијалице која је спојена на ПЛЦ. Ако је однос између ових променљивих одређен једначином:

$$LAMP = PR1 + PR2$$

онда се вредност непознате променљиве LAMP у било ком тренутку тачно одређује преко непознатих променљивих PR1 и PR2 сагласно OR (ИЛИ) функцији. Претходна једначина успоставља непрекидан однос између променљивих LAMP, PR1 и PR2.

Нека сада буде комплекснији случај:

$$LAMP = (LAMP + PR1) \times PR2.$$

Стање израза LAMP које је одређено претходном релацијом дато је у табели P1.

PR1	PR2	LAMP
0	0	LAMP
0	1	0
1	0	1
1	1	0

Табела 1. Приказ стања израза

Прва комбинација је најинтересантнија због тога што она показује да када су оба улаза 0 (OFF), сијалица остаје у свом претходном стању које може бити 0 или 1.

2.5. Ладер дијаграм

ЛАД едитор представља програм као графички приказ сличан електричним шемама. Ладер програм емулира ток електричне струје од извора напајања, кроз низ улазних логичких услова, који на крају резултирају стањима излаза.

Лева страна ЛАД дијаграма се напаја, затворени контакти омогућавају пролаз енергије, а отворени контакти прекидају ток. Логика је подељена у нивое - *network*, и програм се извршава један по један ниво, се лева на десно, и од врха ка дну. Такав изглед подсећа на лествице, и по томе је овај приказ и добио име. Различите инструкције су представљене графичким симболима и укључују три основне форме. Контакти представљају стања логичких улаза као што су прекидачи, тастери, или унутрашњи услови. Калем обично представља логичке излазе као што су лампе, мотори, стартери, релеји, или стања логичких излаза. Блок представља додатне инструкције, као што су тајмери, бројачи или математичке функције.

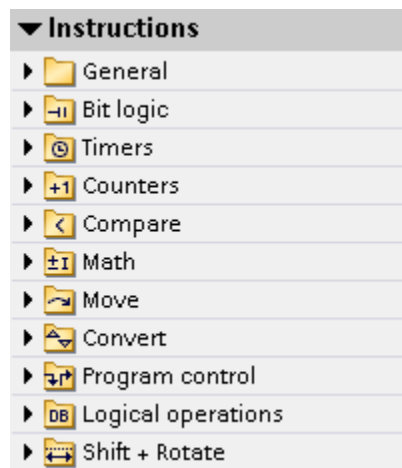


Слика 11. Пример програма у ЛАД едитору

2.5.1. Основне инструкције лествичастог дијаграма

Сет инструкција развијен у СИЕМЕНС-у за програмирање ПЛЦ-ова из фамилије S7-1200 је јако широк. Као што је већ речено инструкције су представљене графичким симболима и укључују три основне форме: контакте, калемове, и блокове. Према својој функцији могу се поделити на неколико група: генералне или основне инструкције, бит логичке инструкције, инструкције тајмера, инструкције бројача, инструкције поређења, математичке инструкције, претварачке инструкције, инструкције логичких операција, итд.

На слици је приказани основни сетови инструкција, који се, притиском на стрелицу, разлажу на појединачне иснтрукције сваког сета.



Слика 12. Основни сетови инструкција

Пре него што се пређе на конкретне инструкције, биће објашњене основне конвенције везане за програмирање S7-1200.

EN (Enable IN) - укључен улаз - представља Булов улаз за блокове у ЛАД дијаграмима. Да би блок инструкција била извршена потребано је да на улазу постоји ток струје.

ENO (Enable OUT) - укључен излаз - представља Булов излаз за блокове у ЛАД дијаграмима. Ако блок има ток струје на ЕН улазу, и блок инструкција буде извршена без грешке, ЕНО излаз прослеђује ток струје следећем елементу. Ако се открије грешка приликом извршавања блок инструкције, ток струје је прекинут у блоку који је генерисао грешку.

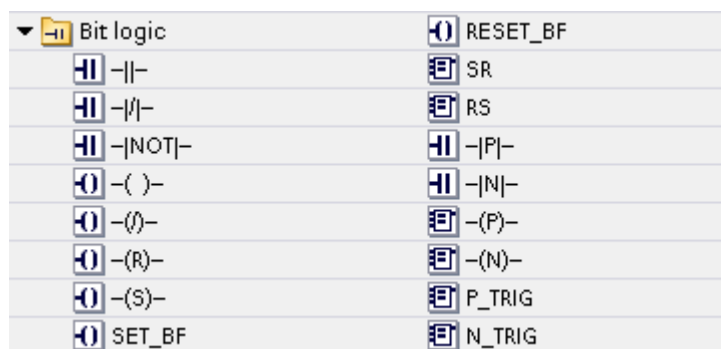
Условни/безусловни улази - у ЛАД дијаграмима, блок, или калем који је завистан од тока струје представљен је са везом на неки елемент на левој страни. Блок, или калем који је независан од тока струје представљен је директним везивањем на леву шину извора напајања.

Инструкције поређења - извршавају се без обзира на стање тока струје. Ако нема тока струје (*false*), нема ни излаза (*output is false*). Ако постоји ток струје, излаз зависи од резултата поређења. Инструкције поређења се представљају као блокови, иако се операције обављају како контакт.

2.5.2. Бит логичке инструкције

Стандардни контакти могу бити *Нормално отворени* и *Нормално затворени*, а вредности добијају из меморије, или регистра ако је тип податка I или Q. *Нормално отворени контакт* --|/-- је затворен (укључен - он) када је бит једнак јединици, а *Нормално затворени контакт* --|/|-- је затворен (укључен - он) када је бит једнак нули.

Инструкција *Негације* --|NOT|-- мењају статус тока струје улаза, тј. мењају вредности из 1 у 0, или из 0 у 1.

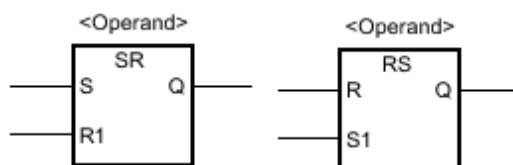


Слика 13. Бит-логичке инструкције

Контакти *Контролисање силазном ивицом операнда --|N|--* и *Контролисање узлазном ивицом операнда --|P|--* контролишу сигнал у тренутка када сигнал мења стање, у првом случају са 1 на 0, а у другом случају са 0 на 1.

Инструкција *Излазни калем --()--* се сетује појавом сигнала. Инструкције *Излазни калем сетовања --(S)--* и *Излазни калем ресетовања --(R)--* сетују (постављају, укључују) или ресетују (искључују) излаз у зависности од стања, логичка јединица или логичка нула.

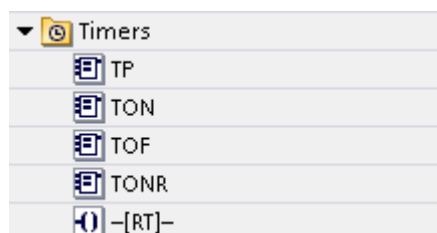
Инструкције доминантног сетовања *Set/reset flip-flop (SR)* и ресетовања *Reset/set flip-flop (RS)* су прекидачка кола са памћењем где је сетовање, односно ресетовање доминантно. Ако су и сигнал сетовања (S) и ресетовања (R1) присутни (тачни), онда је и излаз тачан, односно нетачан, респективно.



Слика 14. Set/reset и Reset/set flip-flop

2.5.3. Инструкције тајмера

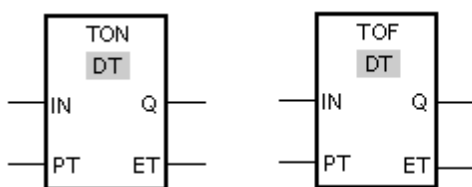
Тајмери који се најчешће користе и које ћемо укратко описати су *Тајмер са кашњењем паљења (TON-On delay)* и *Тајмер са кашњењем гашења (TOF-On delay)*.



Слика 15. Тајмери

Тајмер са кашњењем паљења (TON-On delay)

ТОН тајмер инструкција броји време када је улаз укључен (ON). Након истека задатог периода времена по паљењу улаза, пали се излаз. *TON* инструкција броји време од тренутка када се упали улаз. Када је тренутна вредност већа или једнака преподешеној вредности **PT**, бит тајмера се пали **Q**. Тренутна вредност ТОН тајмера се брише када се улаз угаси. Максимална вредност бројача тајмера је 2147483647мс.



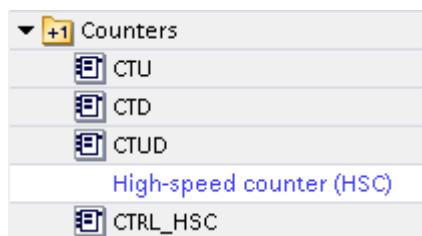
Слика 16. Тајмери ТОН и ТОФ

Тајмер са кашњењем гашења (TOF –Off Delay)

ТОФ тајмер се користи да се одложи гашење излаза за одређени период времена након што се улаз угаси. Након што се улаз упали, бит тајмера се пали, и тренутна вредност се поставља на нулу. Када се улаз угаси, тајмер броји док протекло време не достигне задату вредност. Када достигне задату вредност **PT**, бит тајмера се искључује, и тренутна вредност престаје да расте. Међутим, ако се улаз опет упали пре него што *TOF* достигне задату вредност, бит тајмера остаје укључен. Улаз, дакле, мора прећи из укљученог-у- искључено стање (on-to-off) да би *TOF* кренуо са бројањем.

2.5.4. Инструкције бројача

На слици приказани су бројачи од којих ћемо објаснити *CTU* и *CTD* бројаче који су најчешће у употреби.



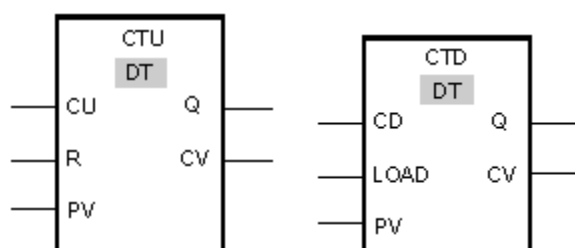
Слика 17. Бројачи

Бројач на горе (*Count up*)

Инструкција *Бројач на горе* (CTU) броји на више, од тренутне вредности бројача, сваки пут када улаз **CU** промени стање из искљученог у укључено. Када је тренутна вредност већа или једнака преподешеној вредности **PV**, излаз бројача **Q** се укључи. Бројач се ресетује када се ресет улаз **R** укључи, или када се изврши инструкција ресетовања. Када достигне максималну вредност од 32767, бројач се зауставља. Постоји још једна вредност, а то је тренутна вредност бројача **CV**.

Бројач на доле (*Count down*)

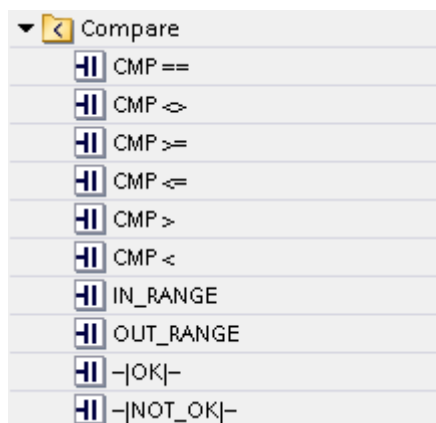
Инструкција *Бројач на доле* (CTD) броји на ниже, од тренутне вредности бројача, сваки пут када улаз **CD** промени стање из искљученог у укључено. Када је тренутна вредност **CV** једнака нули, излаз бројача **Q** се укључи. Када се укључи **LOAD** улаз, бројач ресетује **CV** и учитава преподешену вредност **PV**. Када достигне нулту вредност, бројач се зауставља и бит бројача се укључује.



Слика 18. Инструкције CTU и CTD бројача

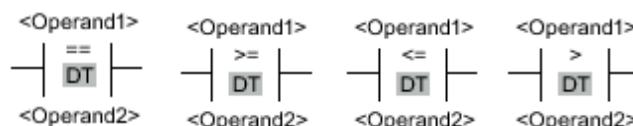
2.5.5. Инструкције поређења

Често приликом писања програма долазимо у ситуацију да неке добијене вредности треба упоредити да би се одређени процес завршио на жељени начин. Инструкције поређења нам служе баш за овај вид решавања проблема.



Слика 19. Инструкције поређења

Инструкција *Једнако* ($CMP ==$) служи да упореди вредност $\langle Operand1 \rangle$ са вредношћу $\langle Operand2 \rangle$ и у случају да је испуњен услов пропушта сигнал даље. Исто важи и за све остале инструкције. Принцип је исти. инструкција *Веће или једнако* ($CMP \geq$) упоређује да ли је прва вредност већа или једнака од друге, ако јесте пропустиће сигнал.



Слика 20. Инструкције поређења $CMP ==$, $CMP \geq$, $CMP \leq$ и $CMP >$

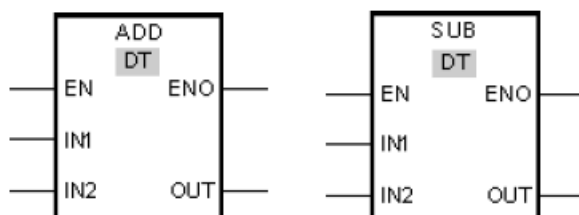
2.5.6. Математичке инструкције

Математичке инструкције користимо када желимо да манипулишемо са вредностима, било то да се односи на сабирање, одузимање, дељење или множење, и чак неке компликовањије математичке операције. Као што можемо видети на слици.



Слика 21. Математичке инструкције

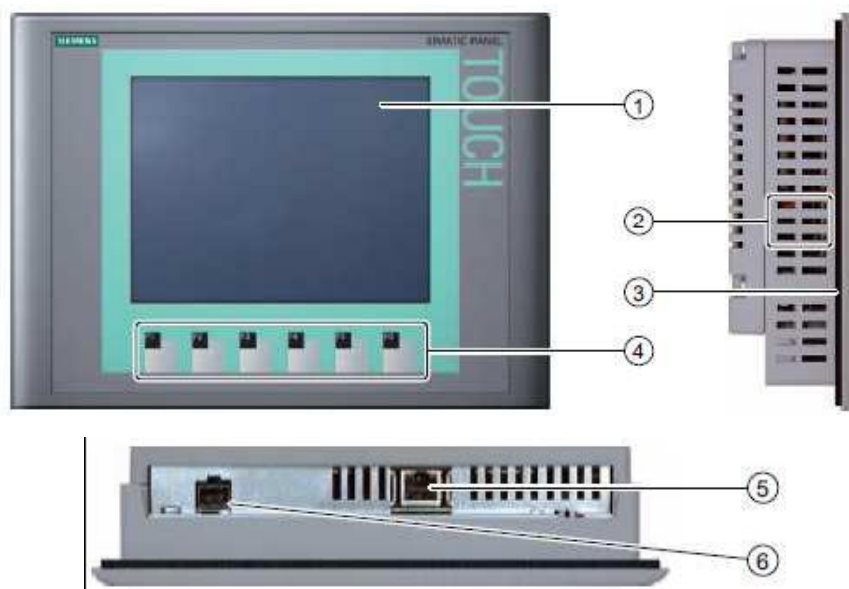
Инструкција *Сабирање* (ADD) функционише по принципу $OUT = IN1 + IN2$. Када се изврши овај услов без грешке, на излазу ENO се јавља сигнал. Инструкција *Одузимање* (SUB) функционише по принципу $OUT = IN1 - IN2$. Када се изврши овај услов на излазу ENO се јавља сигнал. Ове инструкције се извршавају само у случају појављивања сигнала на улазу EN .



Слика 22. Инструкције ADD и SUB

2.5. ТПанел

Изабрани HMI уређај јесте *HMI KTP 600 PN* који је један од стандардних панела у *SIMATIC S7* понуди. Функција HMI уређаја јесте да обезбеде комуникацију између оператера у пољу аутоматизације и процеса који се управља и надзире. Он омогућава унос одређених задатих вредности које оператер бира или уноси на основу свог експертског знања и информација које управо добија са панела. Преглед информација мора да буде што јаснији за оператера, тако да модерни HMI уређаји имају све боље могућности за визуелизацију процесних параметара и функционалности апликације надзора и управљања у целини.



Слика 23. HMI KTP600 PN - 1.Дисплеј (Touch Screen), 2.Монтажни елементи, 3.Заптивак, 4.Функцијски тастери, 5. RJ45 PROFINET конектор, 6.напонски конектор

Уређај има 5.7inch-ни дисплеј и 6 функцијских тастера. Конектор, на слици обележен бројем 5, се користи за повезивање овог уређаја у мрежу. Једноставно коришћењем конектора RJ45 овај уређај постаје део мреже и видљив је на мрежи. Важно је напоменути да је за рад овог уређаја потребно обезбедити електрично напајање од 24V једносмерне струје (конектор је на слици обележен бројем 6). Уређај подржава рад са 265 боја, у резолуцији 320x240 пиксела, а сам произвођач истиче да је предвиђен за рад са контролерима S7-1200.

2.6. ИП камера

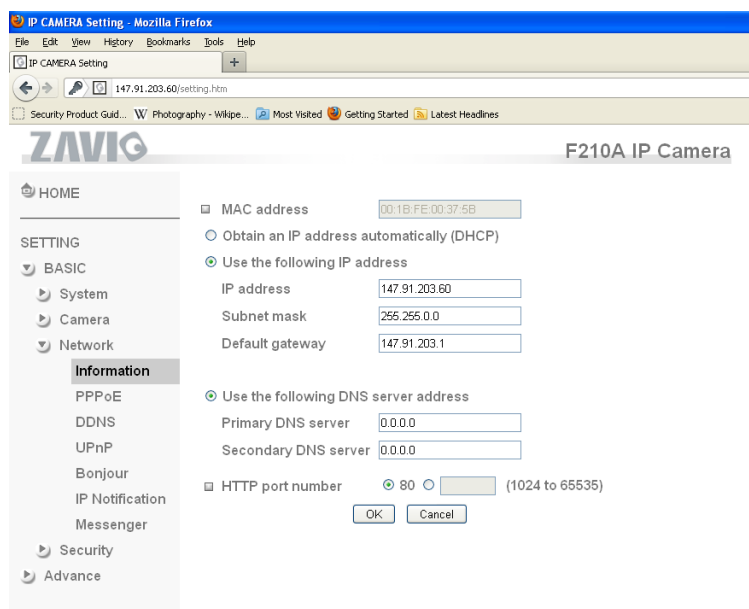
Као један од могућих независних решења за визуелни надзор процеса може се користити различите врсте камера са разноврсним системима преноса података.

ИП камера или Internet protocol camera, је тип дигиталних видео камера које се углавном користе за надзор. Ове камере, поред сензора за пријем светлосних сигнала, садрже и мрежни модул који им омогућава да шаљу и примају податке преко рачунарске мреже и интернета. Постоје различите врсте ИП камера у зависности од резолуције чипа (квалитета видеа), преноса података (cable, WiFi), преноса аудио сигнала, типа напајања. У овом случају смо користили ИП камеру Zavio F210A која се налази у Центру за примењену аутоматику.



Слика 24. ИП камера

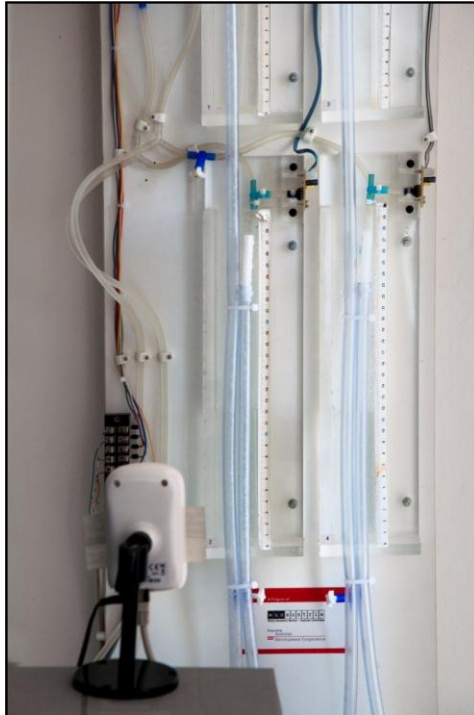
Подешавање параметара камере се може извести на два начина: коришћењем софтвера који произвођач испоручује са овим уређајима или отварањем странице за подешавање (уношењем ИП адресе камере) у претраживачу на мрежи.



Слика 25. Конфигурациони панел ИП камере

У овом случају постављена је ИП адреса 147.91.203.60 која је у истом мрежном опсегу као и сви остали управљачко надзорни уређаји.

Камера је постављена испред Система спрегнутих резервоара и са њом је омогућено даљинско надгледање процеса.



Слика 26. Положај камере

2.7. Рачунар

Рачунар који се користи за програмирање уређаја, израду програма и праћење процеса у TIA STEP 7 Professional V11 SP 2 и WinCC Profesional V11 SP2 мора задовољити следеће минималне хардверске карактеристике:

- Processor: 2.2 GHz Core 2 Duo
- Main memory: 2 GB RAM
- Hard disk: 250 GB S-ATA HDD
- Graphics: min. 1280x1024
- Screen: 15" SXGA + Display (1400 x 1050)

Да би програмер комотно радио на програмирању и праћењу процеса потребан је већи монитор, или два монитора, где би се у једном изводило програмирање, а на другом праћење процеса.

Минималне софтверске карактеристике које би требао да задовољи рачунар за инсталацију 32битне верзије су:

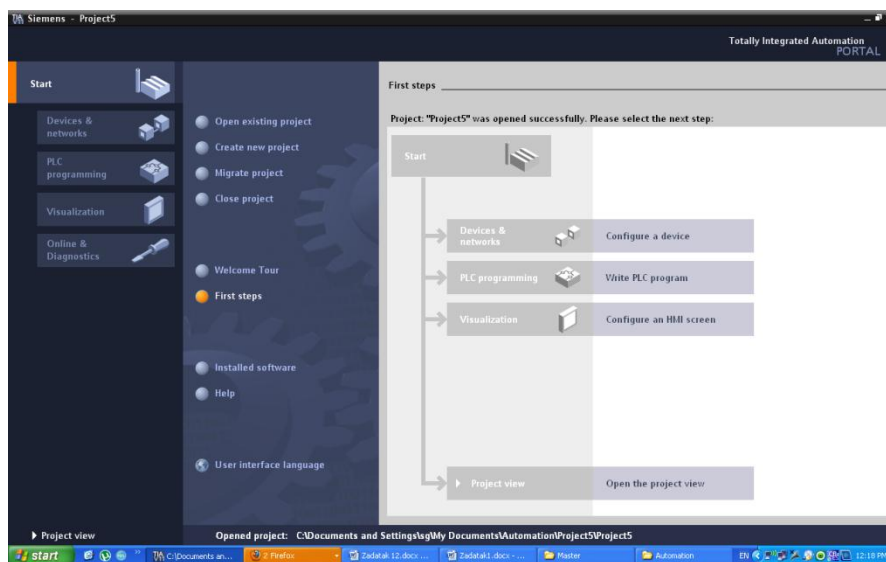
- MS Windows XP Home SP3 (само за STEP 7 Basic)
- MS Windows XP Professional SP3
- MS Windows 7 Home Premium (само за STEP 7 Basic)
- MS Windows 7 Professional, Enterprise and Ultimate
- MS Windows 2003 Server R2 Standard Edition SP2 (само за STEP 7 Professional)
- MS Windows Server 2008 Standard Edition SP2 (само за STEP 7 Professional)

3. ТИА Портал

За писање програма и контролу уређаја користићемо Сименсов софтвер *Siemens Simatic TIA Portal*, *Step7 Professional v11 SP2* и *WinCC v11 SP2* који за сада представљају једно од најефикаснијих софтверских решења за оптимално дизајнирање комплетног аутоматизованог процеса са само једног екрана.

Стартовањем Тиа портала добијамо почетни екран који нам омогућава следеће функције:

- *Start*, отварање старог и креирање новог пројекта, као и миграцију пројекта из неких старијих верзија;
- *Devices & network*, Додавање нових ПЛЦ и ХМИ уређаја, креирање мреже;
- *PLC programming*, Креирање програма за ПЛЦ;
- *Visualisation*, Креирање визуелних система праћења процеса и
- *Online & Diagnostics*, Праћење уређаја на мрежи и диагностика.



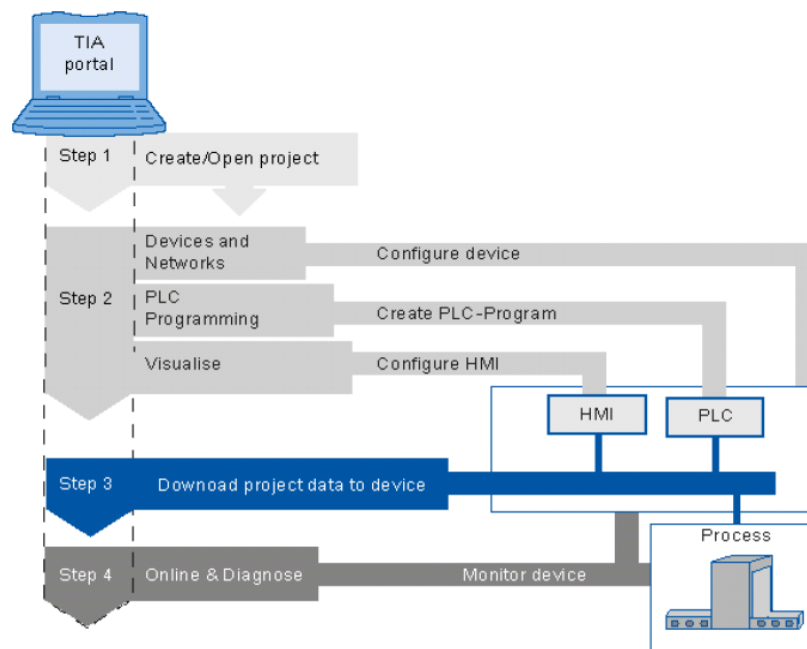
Слика 27. ТИА портал

Овај стартни екран или портал у user-friendly графичком приказу везује више софтверских целина које укључује ТИА Портал. Стартовањем *Project View*, прелазимо на сасвим другачији тип екрана у коме је могуће извести све као и на претходном, али са додатним функцијама за подешавање уређаја и креирање ладер дијаграма.

Project view екран је подељен у више целина (прозора) који се могу веома лако сужавати и ширити по потреби. Поред менија, у коме се налазе све наредбе, и иконица за брже приступање одређеним командама, овај екран садржи са леве стране *Project tree* (дрво пројекта) у коме се налазе све компоненте процеса који желимо да креирамо, тј. да контролишемо. Кликом на стрелице отварају се опције за уређаје, мреже и др. Одабиром одређене гране у главном, средишњем прозору на екрану је могуће мењање опција, програмирање, додавање нових уређаја и сл. Са десне стране екрана се налази више табулатора: *Instruction*, за постављање инструкција када се програмирају уређаји; *Testing*, за тестирање рада уређаја; *Task*, за преглед задатака; и *Libraries*, за преузимање већ снимљених и готових делова програма.

Израда пројекта у *TIA Порталу* се састоји од неколико следећих корака:

- креирање пројекта;
- израда системске конфигурације ПЛЦ уређаја;
- израда и конфигурисање мрежних веза између уређаја;
- израда управљачког програма за ПЛЦ уређаје;
- израда визуализације за ХМИ уређаје;
- учитавање израђених апликација на ПЛЦ и ХМИ уређаје;
- провера рада и отклањање грешака у апликацијама.



Слика 28. Програмирање по корацима

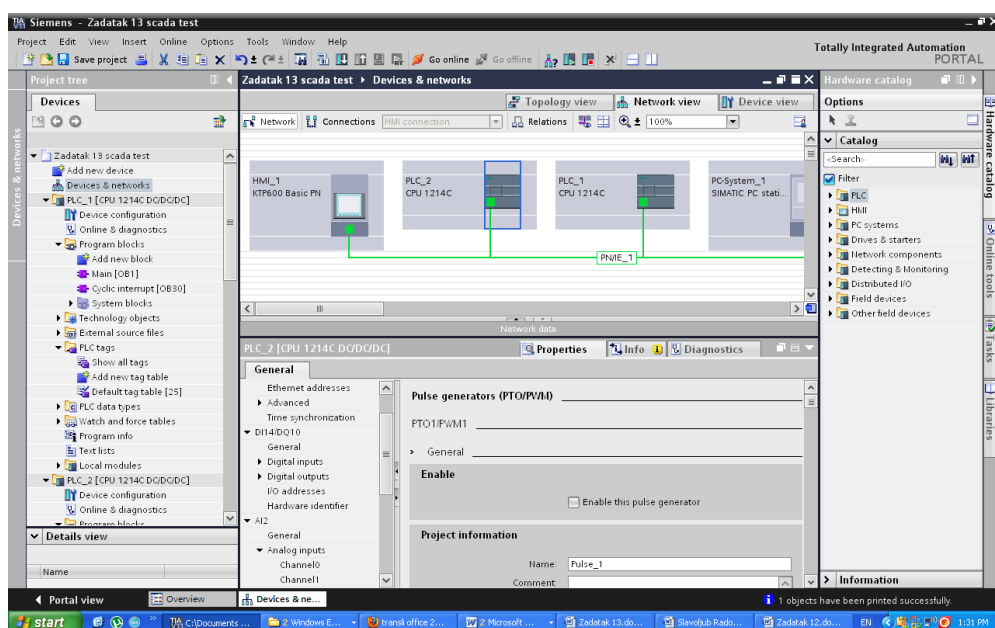
3.1. Додавање уређаја

Да би смо испрограмирали ПЛЦ потребно је додати ПЛЦ уређај, у овом случају s7-1200 са CPU 1214C DC/DC/DC, као и његову аналогну плочу, SB 1232AQ, са firmware v2.2.

Програм Степ 7 омогућава скенирање рачунарске мреже у потрази за компатибилним ПЛЦ уређајима. Да би се успешно програмирао ПЛЦ, рачунар мора да поседује ИП адресу која је из подручја ИП адреса у коме се налази ПЛЦ. У овом случају ПЛЦ је подешен на 147.91.203.132, а компјутер је подешен на 147.91.203.222.

При одабиру ПЛЦ кроз модул *modul Devices and Network* у оквиру Step 7 портала потребно га је конфигурисати на ИП адресу која је физички постављена на ПЛЦу јер по подразумеваној вредности програм је поставља на 192.168.0.1.

На сличан начин се додају и остали уређаји који се користе у процесу (ХМИ и сл.) Дакле, потребно на виртуалну шину из каталога додати уређаје и модуле од којих се састоји конфигурација ПЛЦ уређаја (CPU, CP, CM).



Слика 29. Додавање и конфигуравање уређаја

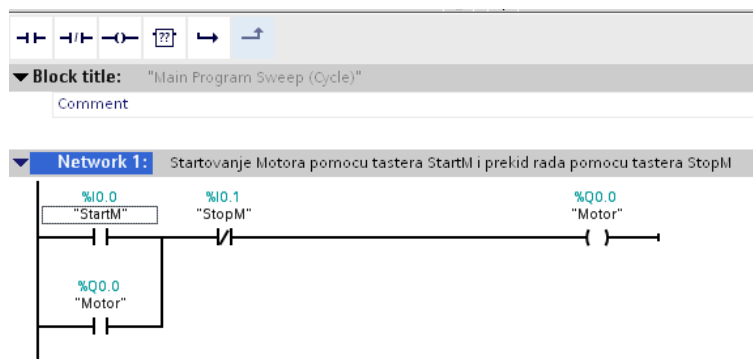
Након тога се креће у израду управљачког програма помоћу контактних дијаграма (енг. *Ladder diagram*) или функцијских блоковских дијаграма (енг. *Function Block Diagram*). Додавање нових програмских блокова се врши одабиром опције *Add new block* у делу *Devices*.

Инструкције се једноставно додају притиском на одабрану групу инструкција унутар картице *Instructions*. Најчешће коришћене инструкције је могуће поставити у алатну траку с инструкцијама изнад радне површине у којој се развија управљачки програм. Додатни преглед или подешавање својстава одабраних објеката могуће је направити унутар картице *General*. Након учитавања системске конфигурације и корисничког

програма у ПЛЦ уређај одабере се "online" поглед програма из ПЛЦ уређаја чиме је омогућено његово тестирање и отклањање грешака у логици ако је потребно.

Дакле, при постављању ПЛЦа отвара се у *Project tree* део за програмирање, тј. *Program block*, а у оквиру њега *Main* и *Cyclic interrupt*.

Креирање програма је врло једноставно јер је *Step7* програм *friendly user* оријентисан. Уз прелезак из модула *Devices and Networks* и *PLC Programming* једноставно са *drag & drop* је могуће поставити активне улазе и излазе са скице ПЛЦ на шему дијаграма.



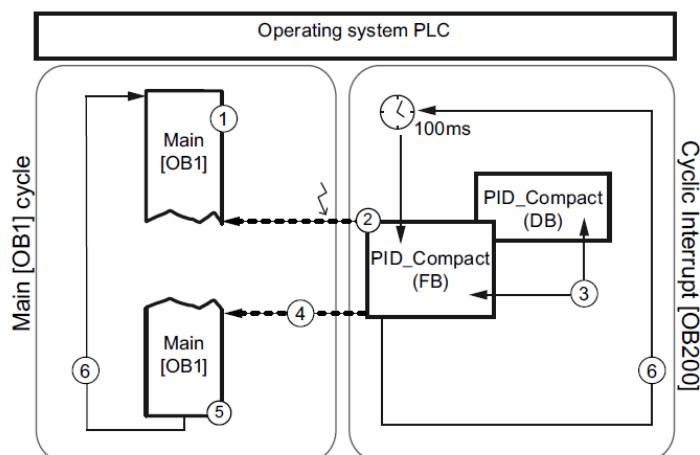
Слика 30. Изглед дијаграма (програма) у *Step7*

Дијаграм се креира одабиром симбола из библиотеке и повезивањем. Због прегледности и касније анализе пожељно је да се дају описи сваког елемента на шеми.

За извођење контроле рада пумпи за пуњење резервоара и одржавање нивоа неопходно је користити ПИД контролер.

3.2. ПИД контрола

ПИД (Пропорционални–Интегрални–Диференцијални) контролер представља технолошки објекат конструисан у оквиру ПЛЦа и користи се за регулацију излазних величина на основу улазних у цикличним прекидима.

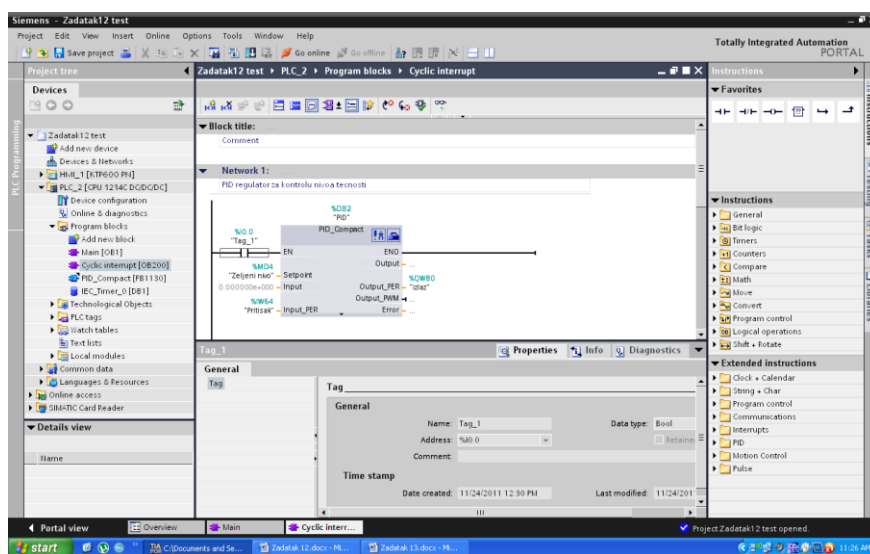


Слика 31. Извршавање програма са цикличним прекидима

1. Програм стартује са извршавањем Main [OB1].
2. Циклични прекид се окида сваких 100 ms, што прекида програм у било којој тачки и и извршава се програм цикличног прекида OB. У овом случају програм се наставља извршавањем функција у PID_Compact.
3. PID_Compact се извршава и вредности се уписују у блок PID_Compact (DB).
4. После извршавања цикличног прекида OB, Main [OB1] наставља у тачки у којој је прекинут. Вредности остају непромењене.
5. Main [OB1] се извршава до краја.
6. Програм поново стартује и извршава се.

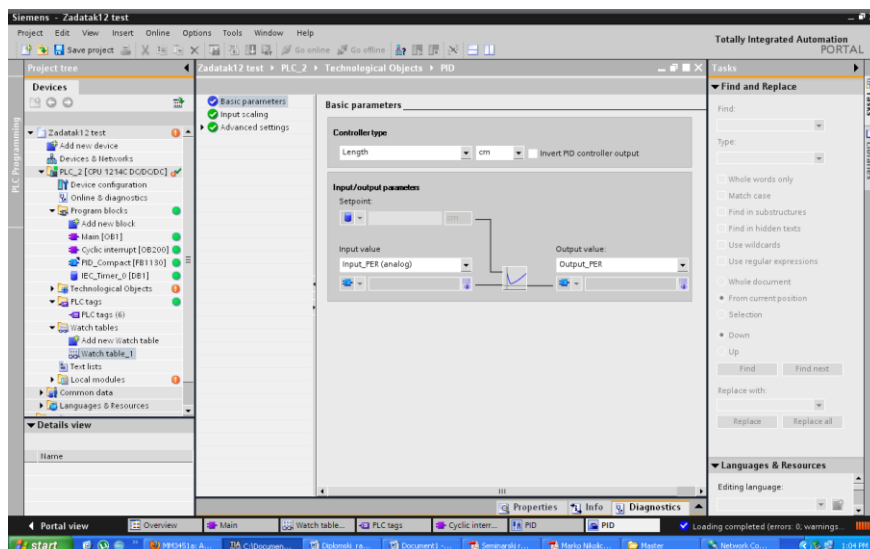
Да би могао да се контролише, потребно га је кроз програм поставити и подесити.

Кроз *Extended instructions* са десне стране се одабира PID, а затим превлачи *PID regulator* у *Cyclic interrupt* где се постављају вредности регулације.



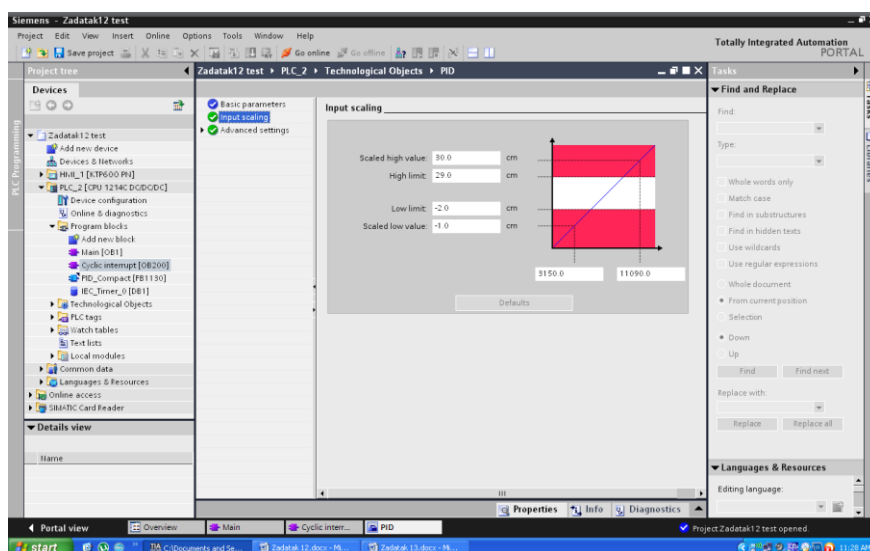
Слика 32. Конфигурисање ПИДа

Потребно је тзв. *tuning* ПИД регулације кроз конфигурацију ПИДа.



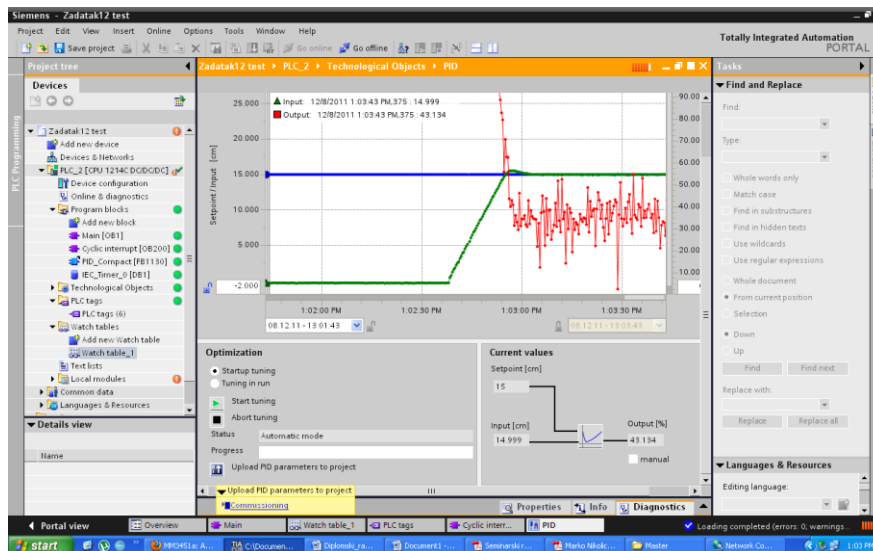
Слика 33. Тјунинг – фино подешавање ПИДа

Постављају се минималне и максималне вредности нивоа у односу на integer (word) вредности са сензора које иду на аналогне улазе.



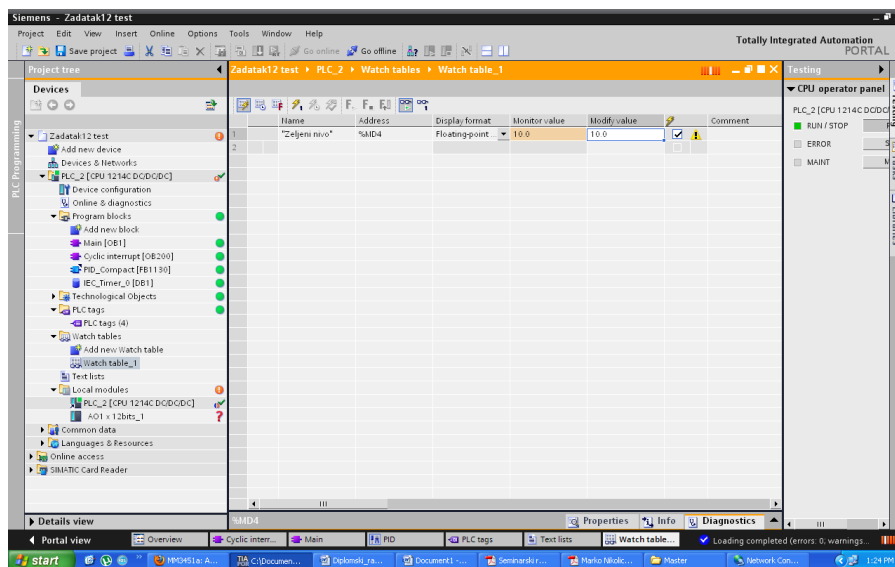
Слика 34. Постављање граничних вредности

Након постављања максималних и минималних вредности врши се тзв. tuning PIDa kroz Commisioning



Слика 35. Тјунинг

Жељена вредност нивоа је постављена кроз меморијску Zeljeni nivo на 15cm у старту али тај ниво је могуће регулисати постављањем ове променљиве у Watch table и подешавањем вредности у Modify value.

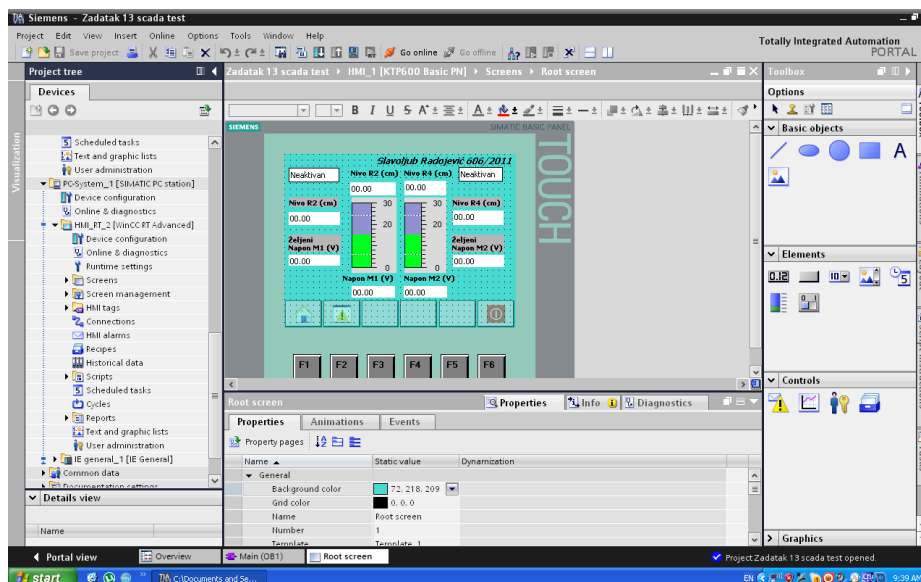


Слика 36. Постављање жељеног нивоа кроз Watch table

На Commissioning панелу је могуће касније праћење регулације нивоа преко дијаграм кроз дугме Start measurement. На дијаграму су приказане криве Setpoint, Input i Output.

3.3. ХМИ

У овом алату се израђују надзорно-управљачке апликације за ХМИ уређаје које се у правилу састоје од екрана који имају улогу графичког интерфејса, дефинисаних аларма и трендова процесних вредности. Почетни прозор се састоји од стабла пројекта, радне површине, траке с алатима, оквира с алатима, и дела за детаљан приказ и промену својстава одабраног пројекта.



Слика 37. Програмирање ХМИ уређаја

У стаблу пројекта за одабрани ХМИ уређај је могуће између осталог одабрати следеће:

- *Online & diagnostics*: добијају се информације о ХМИ уређају и ИП адреса која му је додељена.
- *Runtime settings*: мењају се поставке везане уз извршавање ХМИ апликације
- *Screens*: служи за одабир екрана од којих се састоји надзорно-управљачка апликација. Одабиром екрана у радној површини се могу додавати нови графички објекти. Промена особина тих графичких објеката се ради у делу за детаљни приказ.
- *Screen management*: служи за израду шаблона према којем се у извршавању надзорно-управљачке апликације сви основни објекти налазе на истом месту и имају иста својства.
- *HMI tags*: користи се за израду и уређивање тагова.
- *Connections*: корисити се за уређивање везе према ПЛЦ уређају.
- *HMI alarms*: служи за израду аналогних, дискретних и системских аларма, алармних класа и група.
- *Recipes*: користи се за израду рецепата тј. малих база података која се односи на неки објект у пољу с којим се управља (нпр. вентил, мотор итд.).

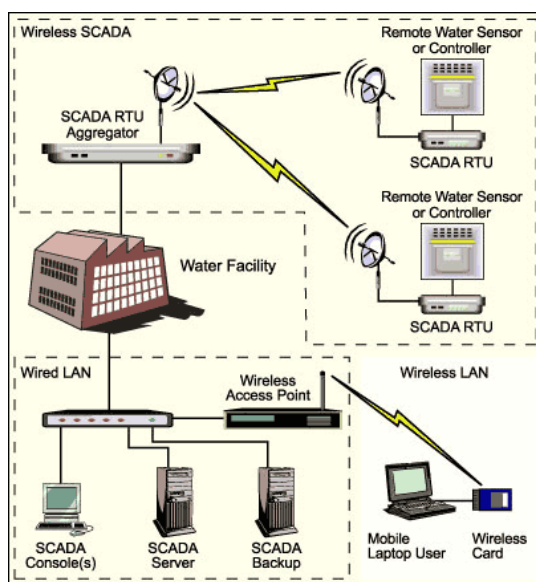
- *User administration*:- служи за уређивање листе оператера с њиховим корисничким подацима и приступним правима.

- *Text & graphic lists*: користи се за брзу проену графика на кориштеним објектима те за унос текста код апликација с више језика.

У делу с алатима за израду графичког екрана могуће је одабрати између основних објекта, елемената и контрола неких већ готових графичких елемената. Основни објекти се користе за цртање линија, квадрата, елипса, писање текста и унос готових графика израђених у неком другом програмском алату. Израђена надзорно-управљачка апликација се учитава у интерну меморију ХМИ уређаја чиме завршава рад с овим алатом.

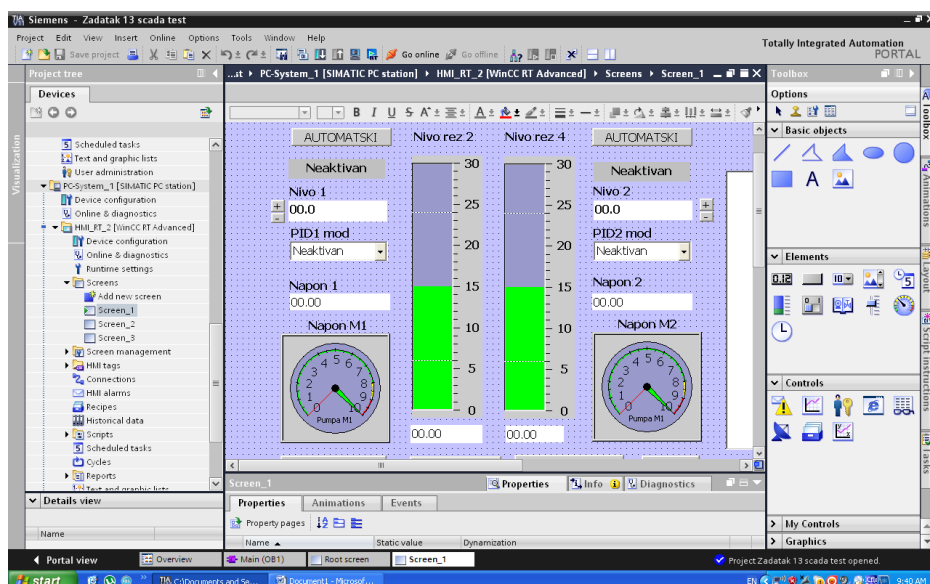
3.4. SCADA

SCADA је акроним од *Supervisory Control And Data Acquisition* (Прикупљање података, надзор, праћење и управљање) – значи систем за аквизицију података и надзорно управљање – и представља комбинацију система телеметрије и даљинског управљања, односно, састоји се од система за аквизицију података, система за пренос података до централног рачунара и система за приказ и анализу прикупљених података.



Slika 38. Primer SCADA sistema

У стварности SCADA систем чини целина од више хардверских и софтверских компоненти. Све те компоненте заједно називамо SCADA системом.



Слика 39. Приказ SCADA

SCADA се реализује постављањем новог уређаја *PC station* у *Devices&networks*. Поред тога потребно је додати и модул *WinCC RT Advanced* и комуникациони модул *IE General* из каталога.

У стаблу пројекта за одабрани *PC station* и *HMI RT* се подешавају и програмирају екрани:

- *Device configuration* – конфигурација уређаја
- *Online & diagnostics*: добијају се информације о уређају и ИП адреса која му је додељена.
- *Runtime settings*: мењају се поставке везане уз извршавање SCADA апликације
- *Screens*: служи за одабир екрана од којих се састоји надзорно-управљачка апликација. Одабиром екрана у радној површини се могу додавати нови графички објекти. Промена особина тих графичких објеката се ради у делу за детаљни приказ.
- *Screen management*: служи за израду шаблона према којем се у извршавању надзорно-управљачке апликације сви основни објекти налазе на истом месту и имају иста својства.
- *HMI tags*: користи се за израду и уређивање тагова.
- *Connections*: корисити се за уређивање везе према ПЛЦ уређају.
- *HMI alarms*: служи за израду аналогних, дискретних и системских аларма, алармних класа и група.
- *Recipes*: користи се за израду рецепата тј. малих база података која се односи на неки објект у пољу с којим се управља (нпр. вентил, мотор итд.).

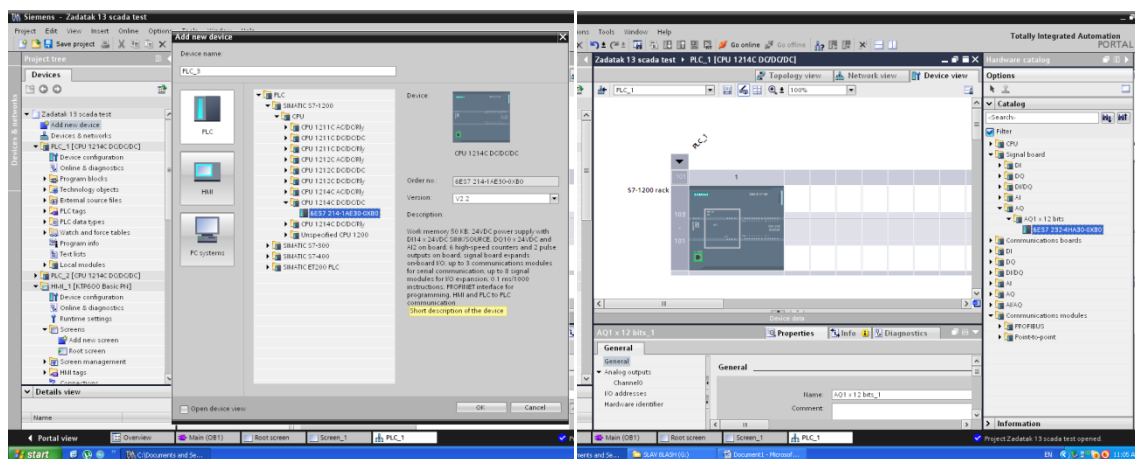
- *Historical data*: користи се за хронолошко праћење одређених тагова и снимање података у фајл за даљу обраду.
- *Scripts*: користи се за креирање посебних програмских целина (скрипти) у *Visual Basic* коду
- *Scheduled tasks*: за постављање или извршавање одређених задатака контроле у временским интервалима.
- *Cycles*: постављање временских
- *Reports*: креирање извештаја о процесима
- *User administration*:- служи за уређивање листе оператера с њиховим корисничким подацима и приступним правима.
- *Text & graphic lists*: користи се за брзу проену графика на кориштеним објектима те за унос текста код апликација с више језика.

4. Извођење задатка

4.1. Devices & Networks

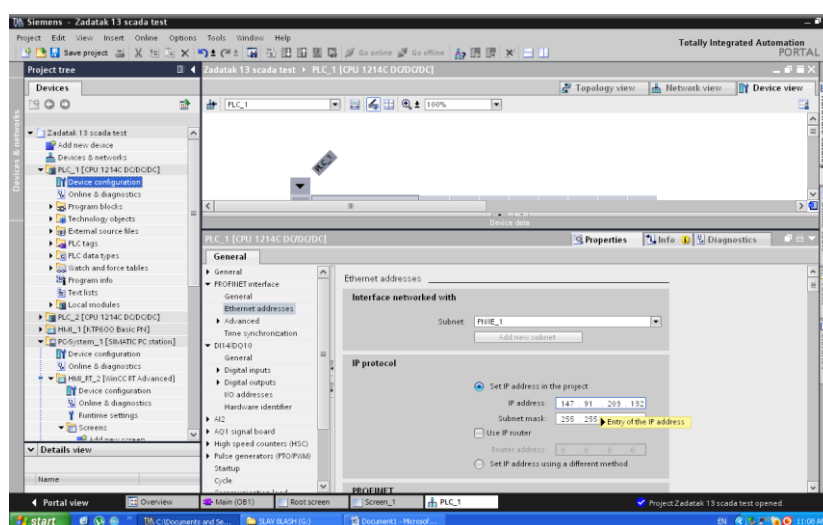
За реализацију задатка прво је потребно додати уређаје (PLC, HMI, PC_staton). Ово се изводи у *Project view-Project tree-Add new devices*:

Додајемо два PLCa S7 1200 sa CPU 1214C (6ES7 214-1AE30-0XB0) v2.2. Након тога се додаје (уграђује) сигнална плоча AQ1 x 12bit (6ES7 232-4HA30-0XB0).



Слика 40. Додавање ПЛЦа и сигналних плоча

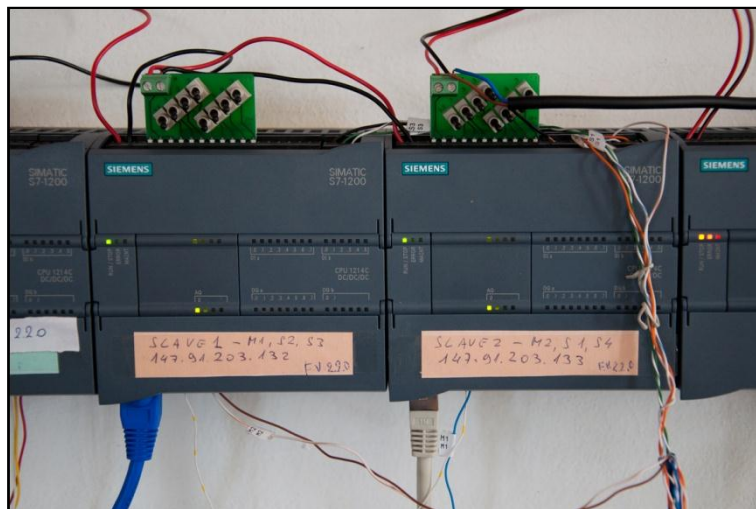
Оба ПЛЦа се кроз *Project tree-Device configuration* конфигуришу, тј. додељује им се ИП адреса. PLC_01 је постављен на 147.91.203.132 док је PLC_02 постављен на 147.91.203.133.



Слика 41. Конфигурација ИП адресе

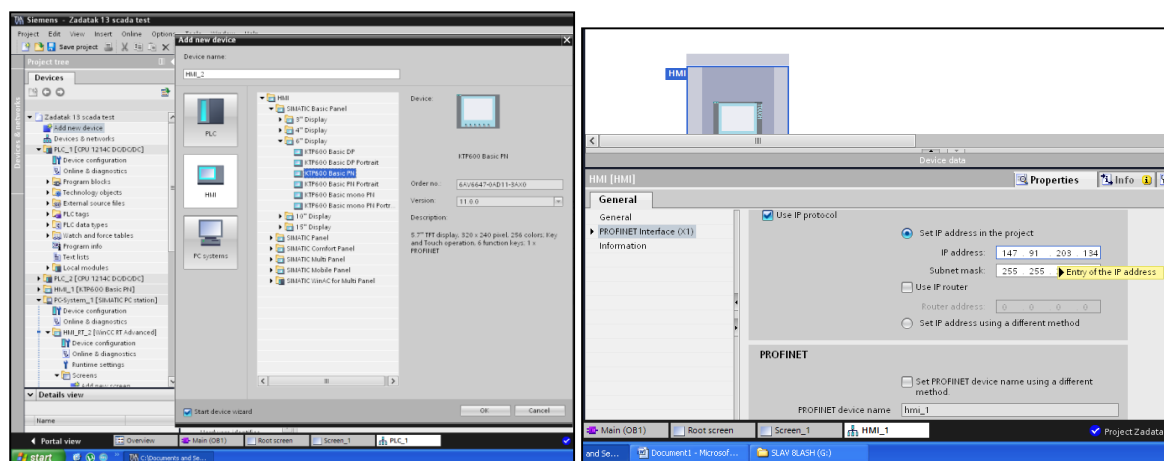
PLC_1 је на својим аналогним улазима везан преко мултиплексера за сензоре нивоа на резервоарима R2 и R3, док је PLC_02 везан за сензоре нивоа на резервоарима R1 и R4. Аналогни излаз из сигналне плоче PLC_1 је везан за пумпу M2, док је аналогни излаз

из сигналне плоче PLC_2 је везан за пумпу M1. Тиме је успостављена комуникација PLCова и посматраног процеса.



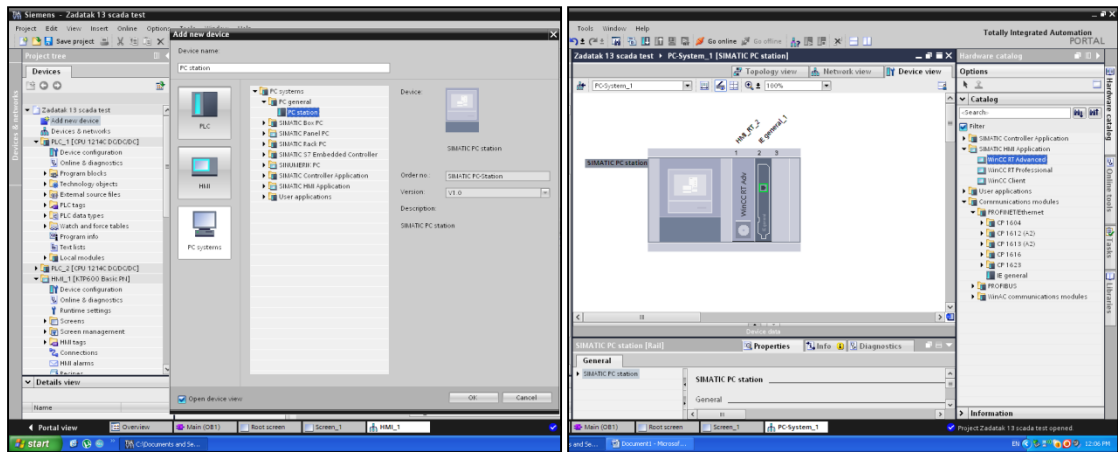
Слика 42. Приказ ПЛЦа на шини

Након тога додајемо HMI KTP600BN и кроз *Project tree-Device configuration* додељујемо ИП адресу



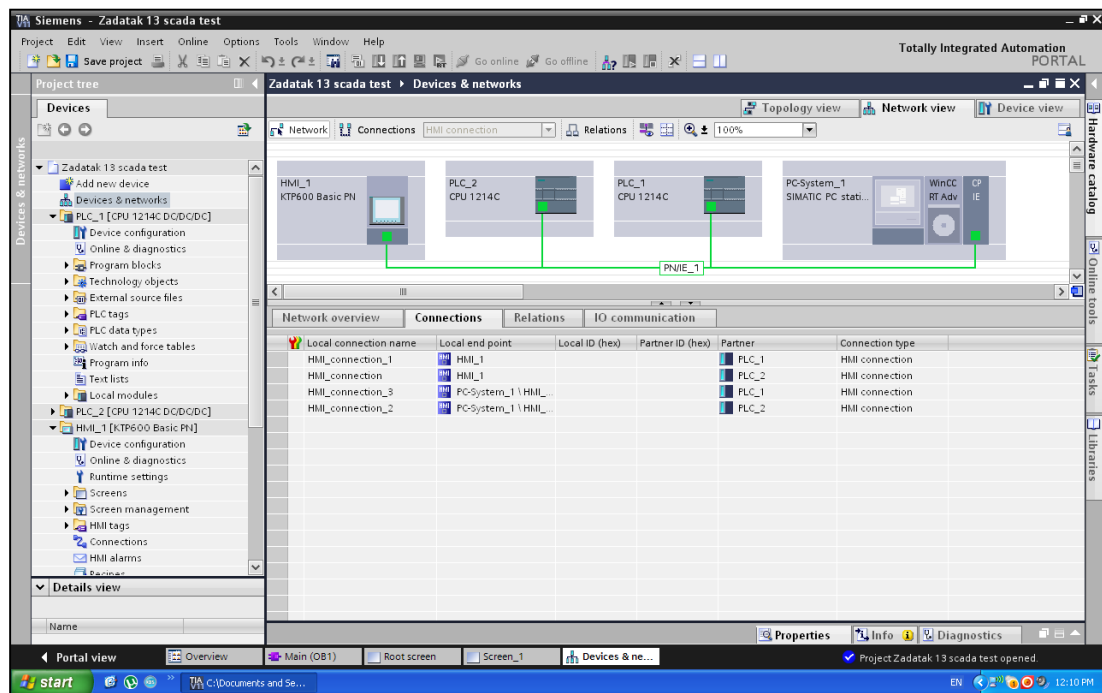
Слика 42. Подешавање ИП адреса

Након тога додајемо PC_station и кроз *Catalog* додајемо као модуле *WinCC RT Advanced* и *Communication modul – IE General*.



Слика 43. Додавање модула

Након додавања уређаја потребно је остварити етернет везу међу њима кроз Devices&Networks-Connections.

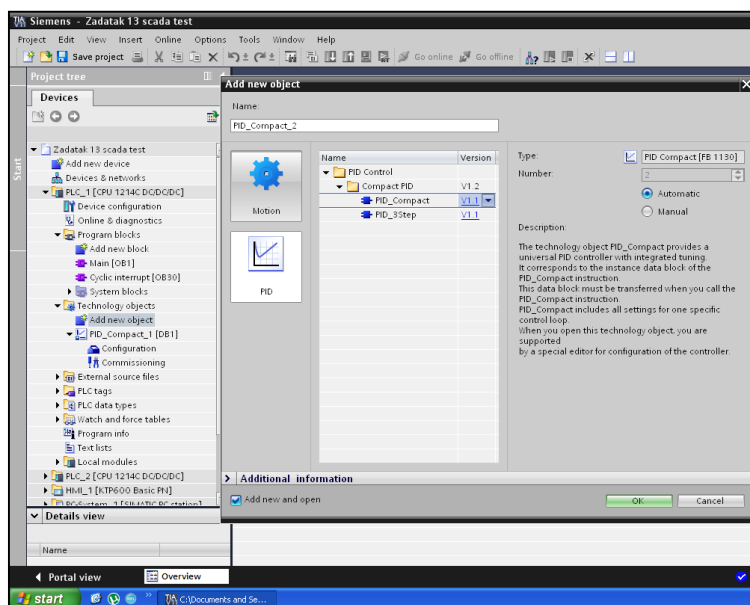


Слика 44. Повезивање уређаја

Тиме су уређаји постављени у мрежу и конфигурисани и даље је на реду програмирање.

4.2. PLC_01

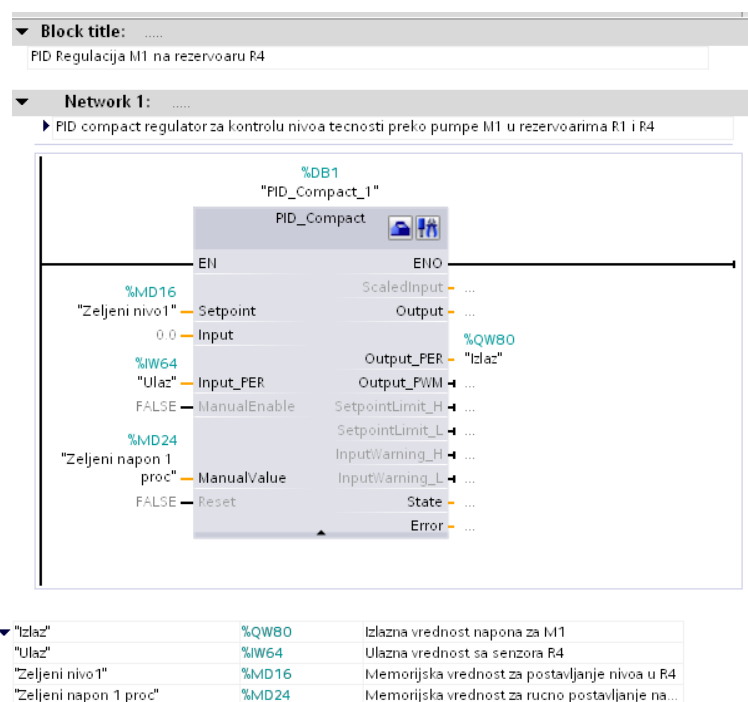
За контролу нивоа течности у резервоарима преко регулације рада пумпе потребно је у *Project tree – PLC_01 - Technology object* на *Add new object* додати *PID_compact* регулатор.



Слика 45. Додавање ПИДа

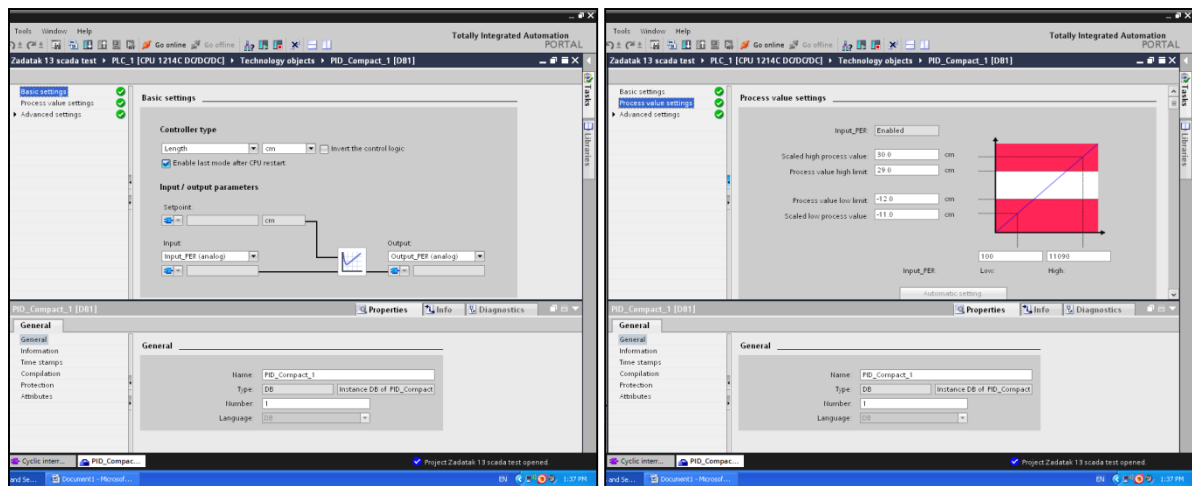
Након тога у оквиру *Project tree – PLC_01 – Program block* отвара *Cyclic interrupt [OB30]* где је постављен *PID_Compact_1* регулатор.

Са леве стране ПИД блока се налазе улазне, а са десне излазне величине (*PCL tags*).



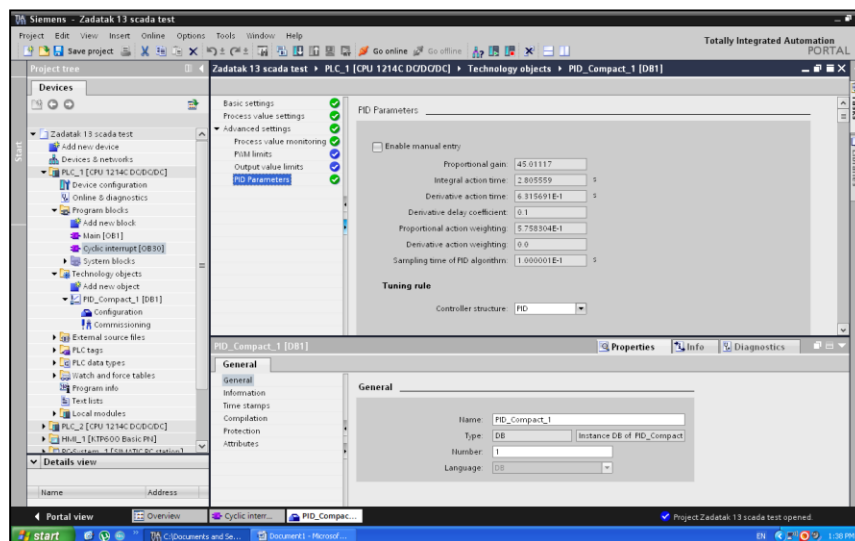
Слика 46. Приказ ПИД блока са величинама за ПЛЦ1

Да би ПИД функционисао са потребна је претходно дефинисање улазних и излазних величина и тзв. тјунинг. То се изводи кроз *Project tree-PCL_01-Technology objects-PID_Compact_1* на *Configuration* и *Commissioning*.



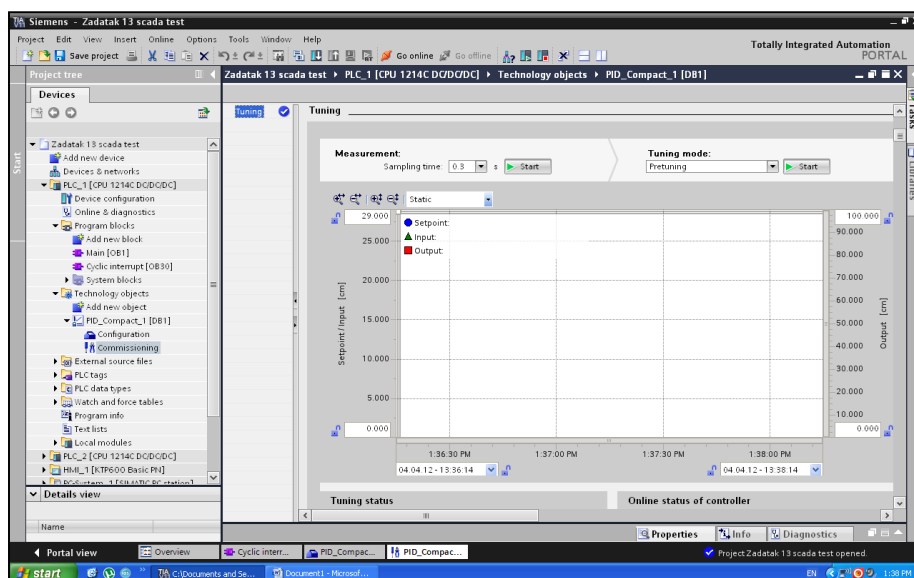
Слика 47. Configuration ПИД регулатора

На *Configuration* се дефинишу улазне величине и постављају горње и доње граничне вредности као и евентуално ручно постављање величина ПИД регулације.



Слика 48. ПИД вредности

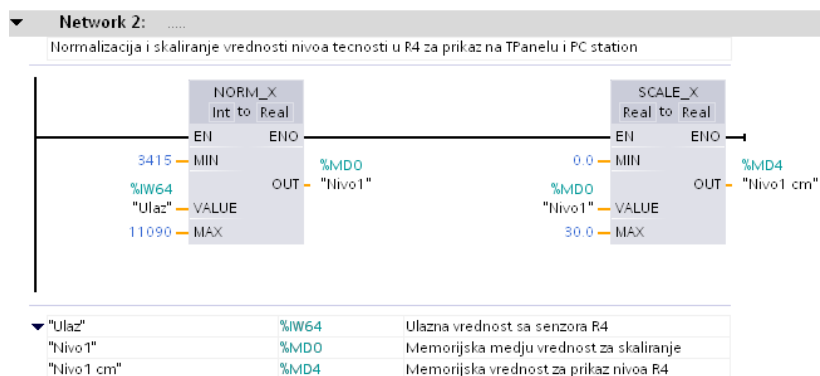
На *Commissioning* се изводи тзв. тјунинг ПИД регулатора и снимање ПИД величина.



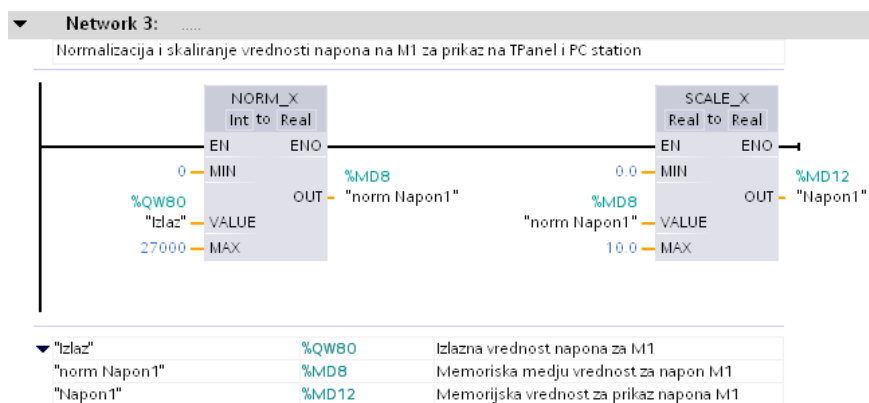
Слика 49. Тјунинг ПИД регулатора

Након тога ПИД је спреман за употребу. Променљиве које се постављају на улазу се уносе преко ХМИ и СКАДА.

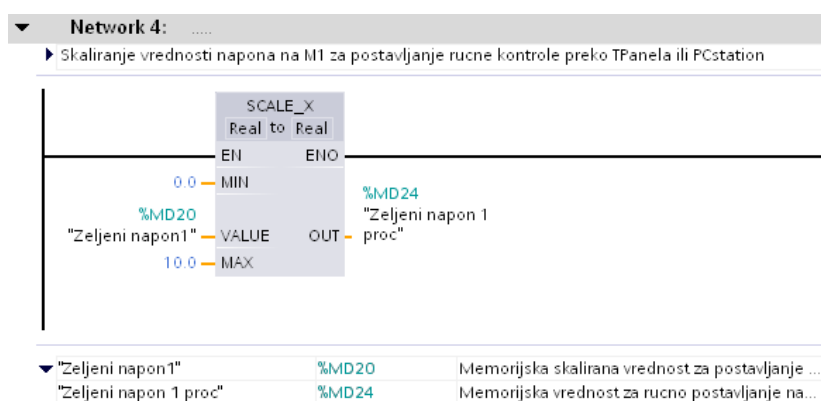
Пошто ПЛЦ комуницира са аналогним величинама процеса преко дигиталних речи (word), потребно је за приказивање и унос променљивих преко ХМИ или СКАДА, све аналогне величине превести и скалирати, што се изводи додавањем блокова *NORM_X* и *SCALE_X*.



Слика 50. Нормализација и скалирање вредности нивоа



Слика 51. Нормализација и скалирање вредности напона



Слика 52. Скалирање вредности напона на пумпи

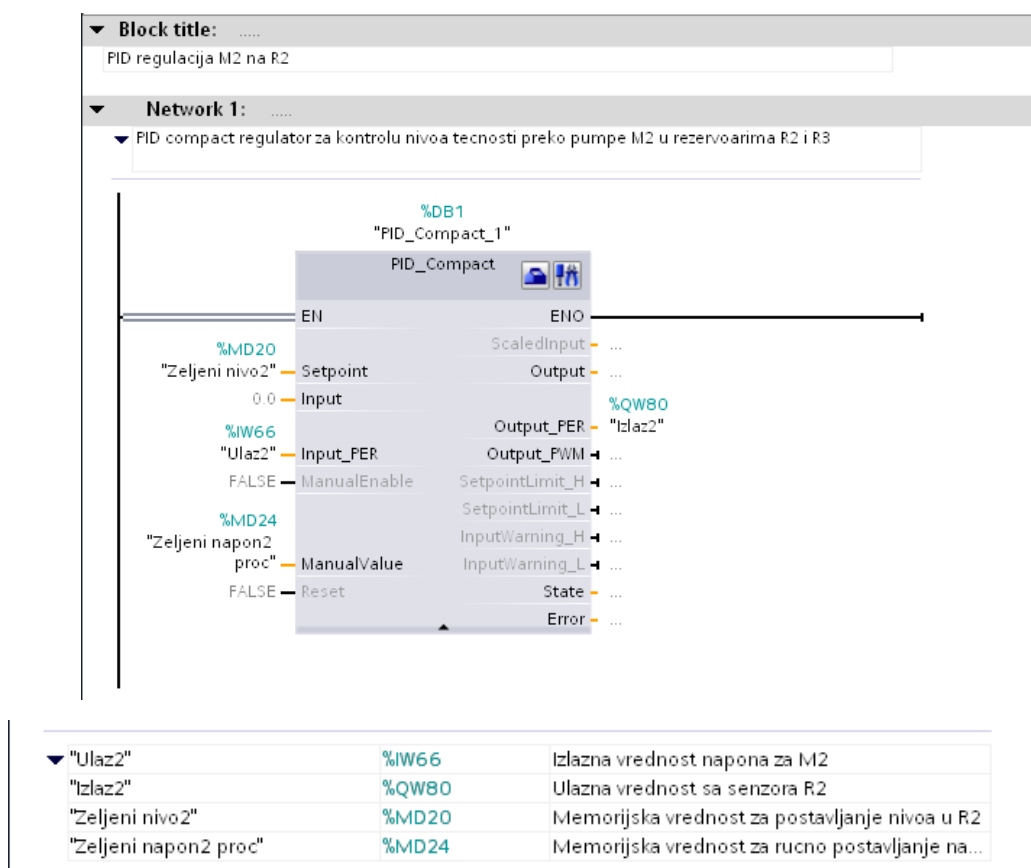
Дефинисањем улазних и излазних величина попуњава се табела тагова тј. променљивих које у зависности од намене добијају физичке меморијске адресе у ПЛЦу. Одабиром *Project tree-PLC_01-PLC tags – Default tag table* добија се приказ свих тагова који се користе у ПЛЦу. Касније се неки од ових тагови везују за улазно/излазне контроле на ХМИ и СКАДА.

Default tag table							
	Name	Data type	Address	Retain	Visibl...	Acces...	Comment
1	Izlaz	Word	%QW80	☐	☒	☒	Izlazna vrednost napona za M1
2	Ulaz	Word	%IW64	☐	☒	☒	Ulazna vrednost sa senzora P4
3	Nivo1	Real	%MD0	☐	☒	☒	Memorijska medju vrednost za skaliranje
4	Nivo1 cm	Real	%MD4	☐	☒	☒	Memorijska vrednost za prikaz nivoa P4
5	norm Napon1	Real	%MD8	☐	☒	☒	Memorijska medju vrednost za napon M1
6	Napon1	Real	%MD12	☐	☒	☒	Memorijska vrednost za prikaz napona M1
7	Zeljeni nivo1	Real	%MD16	☐	☒	☒	Memorijska vrednost za postavljanje nivoa u P4
8	Zeljeni napon1	Real	%MD20	☐	☒	☒	Memorijska skalirana vrednost za postavljanje napona na M1
9	Zeljeni napon 1 proc	Real	%MD24	☐	☒	☒	Memorijska vrednost za rucno postavljanje napona na M1
10	HMIIsakrij	Bool	%M10.1	☐	☒	☒	Pomocna promenljiva za aktiviranje/deaktiviranje kontrola
11	<Add new>			☐	☒	☒	

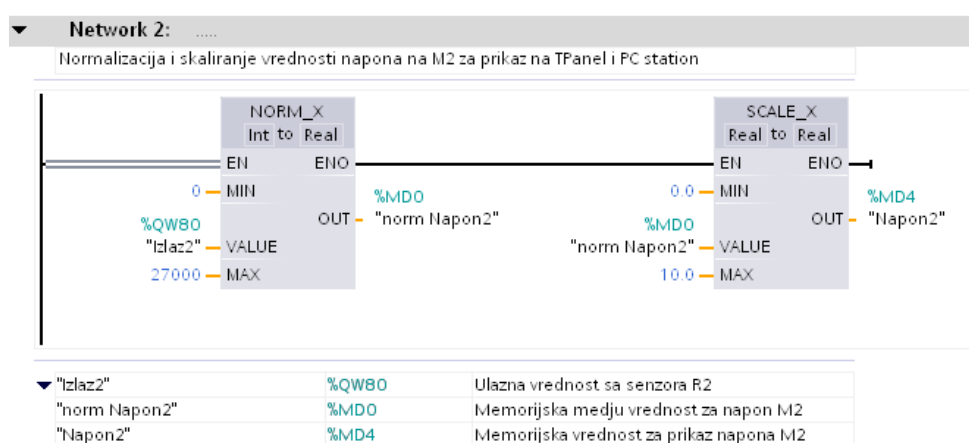
Слика 53. Табела тагова на ПЛЦ1

4.3. PLC_02

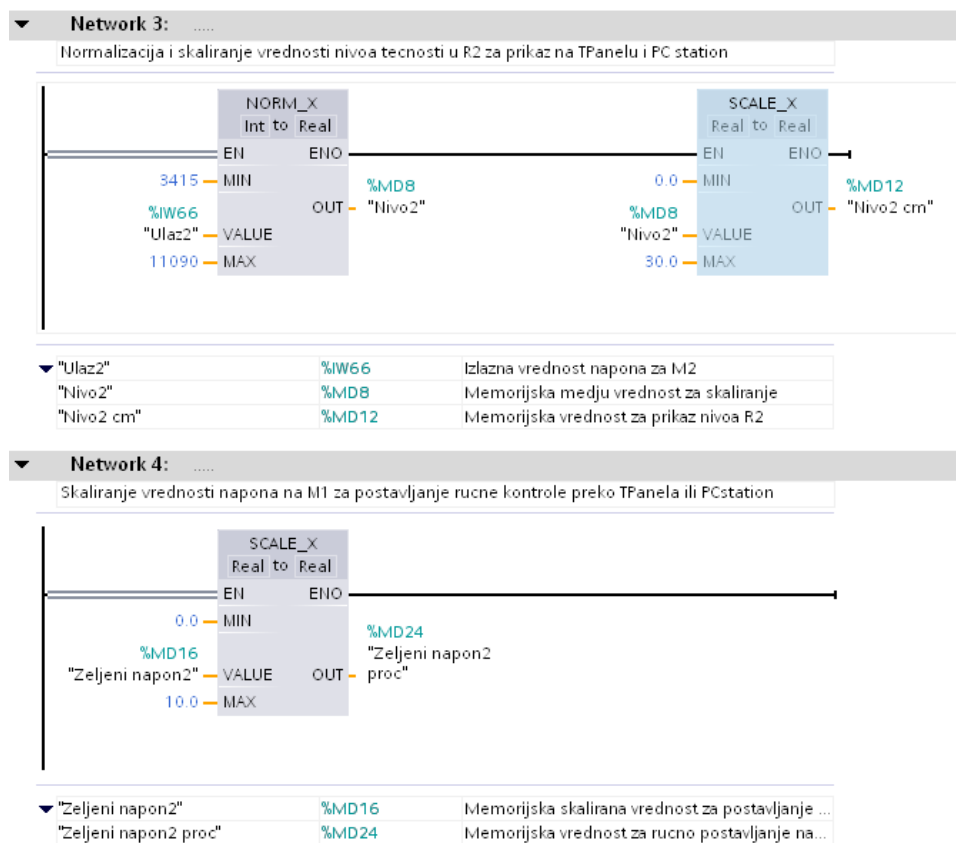
За ПЛЦ2 су везане потпуно исти начин подешавања и дефинисања улазно излазних величина (тагова), с тим што су поједине променљиве са индексом 2 да би се при креирању ХМИ и СКАДА екрана разликовале.



Слика 54. Приказ ПИД блока са величинама за ПЛЦ2



Слика 55. Нормализација и скалирање вредности напона



Слика 56. Нормализација и скалирање вредности нивоа и напона

Default tag table							
	Name	Data type	Address	Retain	Visibl...	Acces...	Comment
1	Ulaz2	Word	%IW66	☐	☒	☒	Izlazna vrednost napona za M2
2	Iblaz2	Word	%QW80	☐	☒	☒	Ulazna vrednost sa senzora R2
3	norm Napon2	Real	%MD0	☐	☒	☒	Memorijska medju vrednost za napon M2
4	Napon2	Real	%MD4	☐	☒	☒	Memorijska vrednost za prikaz napona M2
5	Nivo2	Real	%MD8	☐	☒	☒	Memorijska medju vrednost za skaliranje
6	Nivo2 cm	Real	%MD12	☐	☒	☒	Memorijska vrednost za prikaz nivoa R2
7	Zeljeni nivo2	Real	%MD20	☐	☒	☒	Memorijska vrednost za postavljanje nivoa u R2
8	Zeljeni napon2	Real	%MD16	☐	☒	☒	Memorijska skalirana vrednost za postavljanje napona na M2
9	Zeljeni napon2 proc	Real	%MD24	☐	☒	☒	Memorijska vrednost za rucno postavljanje napona na M2
10	<Add new>			☐	☒	☒	

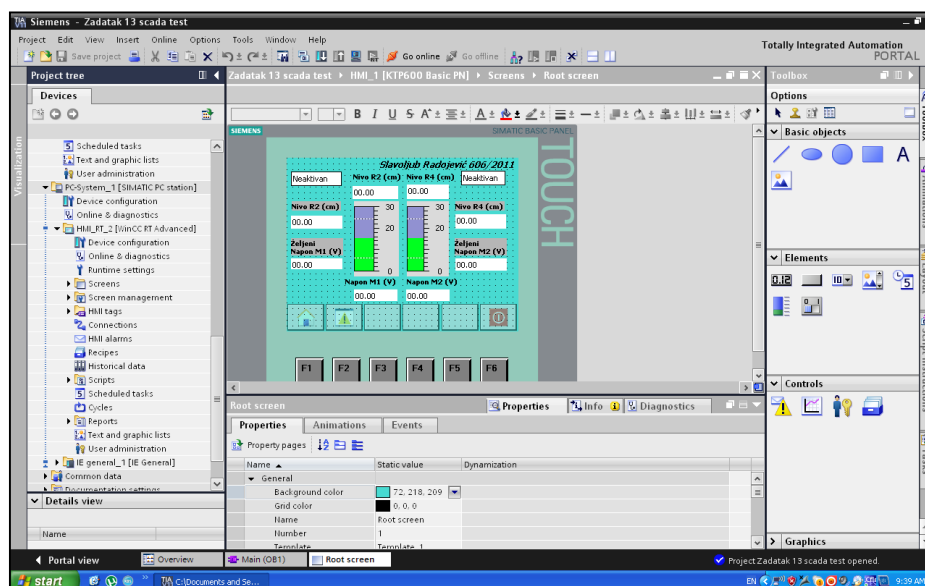
Слика 57. Табела ПЛЦ тагова за ПЛЦ1

4.4. HMI

Програмирање ХМИ, тј. визуелизација контрола и праћење величина процеса је врло једноставна. Избором и превлачењем контрола из *Toolbox-a* се исте постављају на екране.

Root screen представља први, тј. стартни екран на којем је могуће поставити чвориште за више функција које ХМИ треба да подржи. Функцију стартног екрана је могуће подешавати: десни клик на екран – *Define as root screen*.

Све контроле које се постављају на екран се налазе са десне стране у *Toolbox*-у. Ту су *Basic objects* графике (линија, круг, квадрат, лабела, слика), а затим *Elements* контроле за интеракцију са оператером: *IO field* – поље за унос вредности променљивих; *Button* – дугме за стартовање различитих функција (нови екран, ресетовање вредности...) које су дефинисане у оквиру *Event* картице; *Symbolic IO field* – Combo box за избор једне од понуђених вредности; *Graphic IO field* – Листа за избор графички приказаних опција (икона); *Date/Time field*– Поље за унос временских инстанци; *Bar* – за графичко приказивање промене променљивих; и *Switch* – Промена боолеан вредности тага.

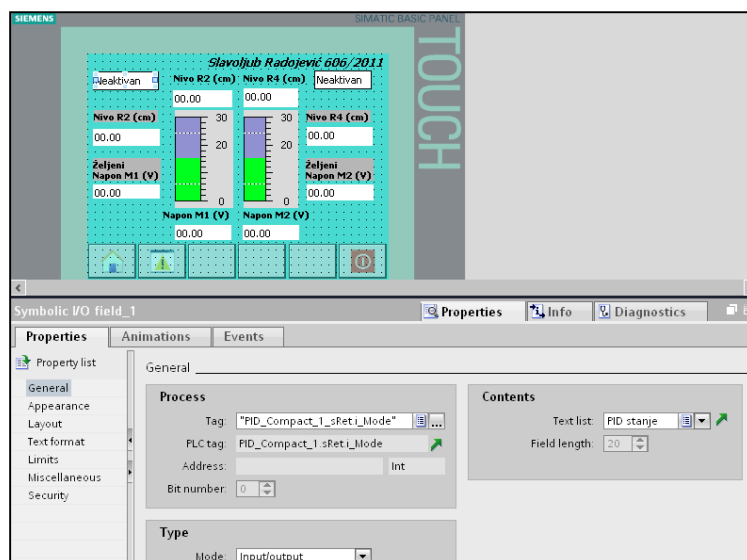


Слика 58. Стартни екран ХМИ контроле

Свакој контроли која се постави на екран потребно је доделити ПЛЦ таг и евентуално подесити опције.

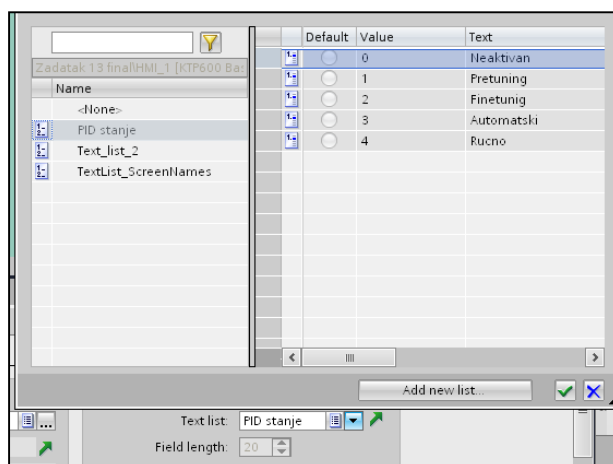
У овом задатку постављено је 12 контрола које су везане за ПЛЦ тагове ПЛЦ1 и ПЛЦ2.

Мод ПИД регулатора је је *Symbolic IO field* која се користи за дефинисање режима рада ПИДа. За ПЦЛ таг се одабира *PID_Compact_1_sRet.i_Mode* за ПЛЦ1 односно ПЛЦ2.



Слика 59. Подешавање контроле мод ПИД регулатора

У *Text list* се креира листа са вредностима 0-4 које су приказане на слици.

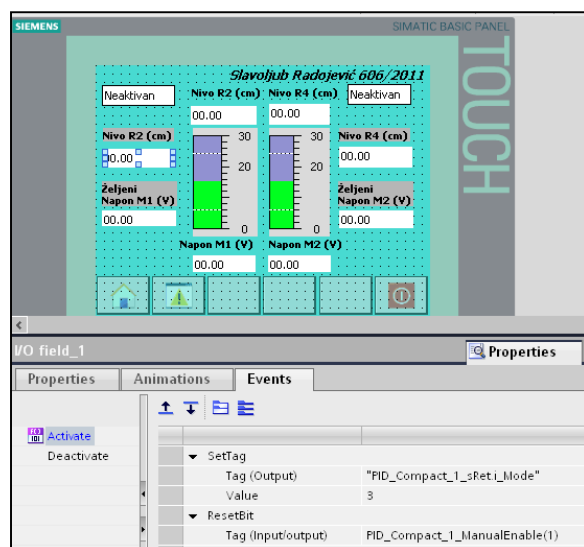


Слика 60. Листа са модовима рада ПИД регулатора

Постављање ове контроле је битно да би се ПИД, тј. процес могао искључивати.

Nivo R2 (cm) је *IO field* контрола за унос нивоа течности у резервоару Р2. Ова контрола је везана за ПЛЦ1 таг *Zeljeni nivo1*. То је променљива која представља улаз *SetPoint* на ПИД регулатору. Контрола **Nivo R4 (cm)** има иста сетовања само што је везана за ПИД регулатор на ПЛЦ2.

Овој контроли су дата и два догађаја која се извршавају након активирања контроле.



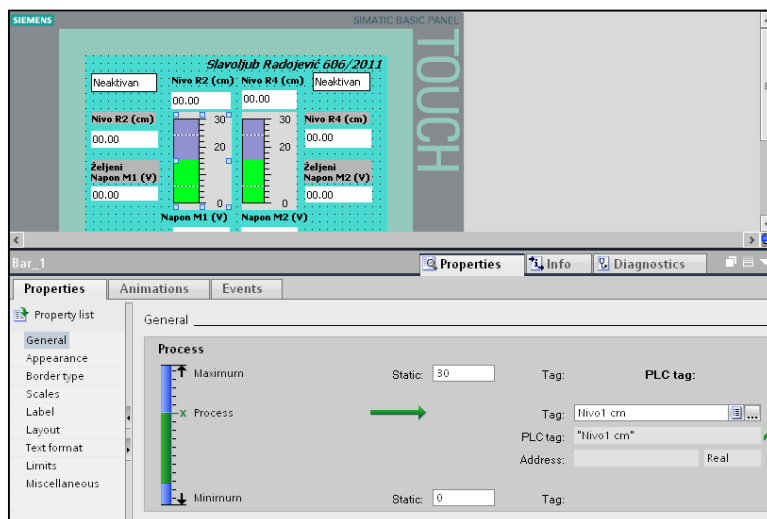
Слика 61. Постављање догађаја на контроли

То су догађаји за постављање нове вредности тагова. Први за постављање аутоматског режима рада ПИД регулатора и друга за искључење, тј. ресетовање мауелног режима.

Željeni napon M1 (V) је *IO field* контрола за постављање вредности напона на пумпи M1. Ова контрола је везана за ПЦЛ таг Željeni napon 1. Такође, су постављени догађаји за избор мануелног режима рада ПИДа и укључивање овог режима. Слично је изведено и за контролу **Željeni napon M2 (V)**.

Nivo R2 (cm) је *IO field* контрола која је одабрана да буде само *Output* која приказује тренутни ниво течности у резервоару. Ова контрола је везана за ПЦЛ таг Nivo 1 cm.

Bar контрола је такође везана за ПЦЛ таг Nivo 1 cm и омогућава оператеру графичко праћење нивоа течности. Ову контролу је потребно додатно подесити што се тиче граница, боја и сл.



Слика 62. Дефинисање Бар контроле

Napon M1 (V) je IO field контрола која је везана за ПЦД таг *Napon1* за приказивање тренутне вредности напона на пумпи M1. Слично је изведена и контрола **Napon M1 (V)**. Ове контроле су сетована као *Output*.



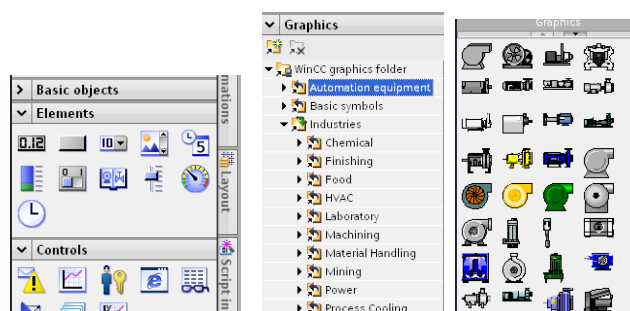
Слика 63. Коначни изглед ХМИ уређаја

4.5. SCADA

Овај виртуелни ХМИ је врло сличан по функционалности физичким ХМИ уређајима, с тим што је поред основних контрола могуће на њега поставити и још неке напредније контроле.

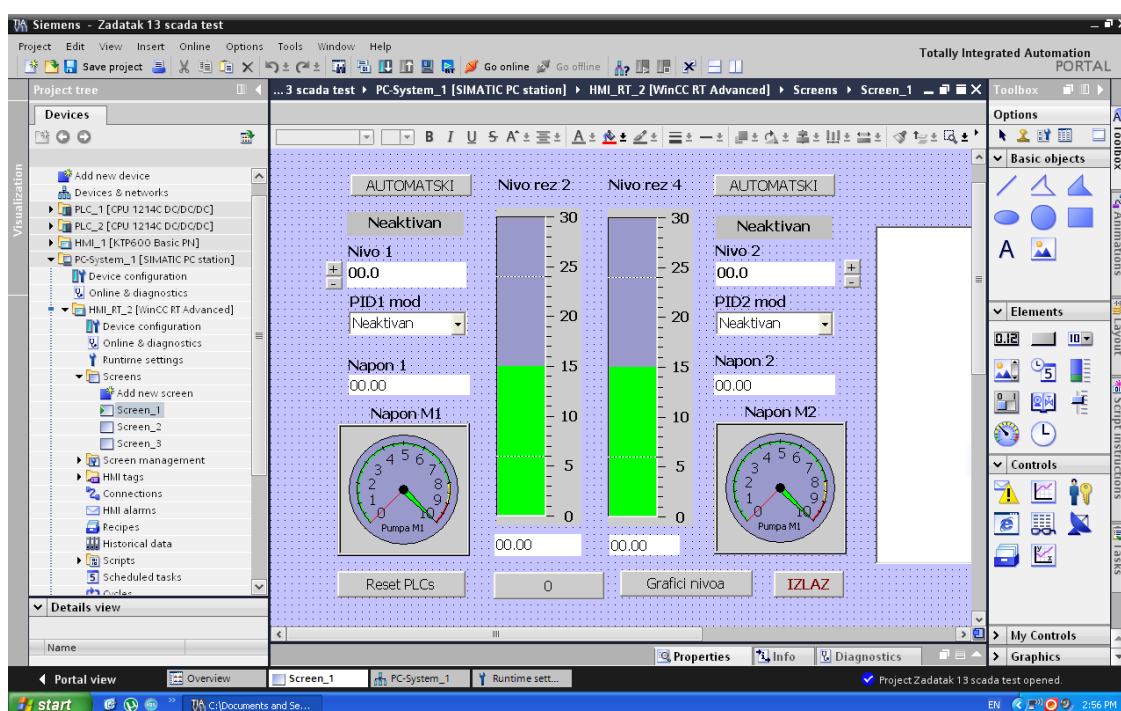
Са десне стране у Toolbox-у се поред основних објеката за цртање (линија, лабела, круг, квадрат) налазе и:

- *Elements*, са контролама *IO field*, *Symbolic IO field*, *Buton*, *Switch*, *Graphic IO field*, *Date/Time field*, *Bar*, *Symbolic Library*, *Slider*, *Gauge*, *Clock*
- *Controls*, са контролама *Alarm view*, *Trend view*, *User view*, *Html browser*, *Status Force*, *Smart client view*, *Recipe view*, *F(x) Trend view*
- *Graphic (Win CC graphic folder)* са специфичним графичким елементима категорисаним по различитим употребним особинама.



Слика 64. Контроле за постављање на SCADA

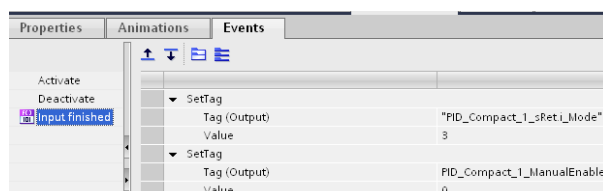
За овај задатак поставићемо контроле као на слици.



Слика 65. Изглед SCADA

Automatski/ručno је *Switch* контрола за одабир рада мода пумпе. Везана је за ПЦЈ таг PID_Compact_1_ManualEnable. Ова контрола је боолеан типа тако да омогућава сетовање 1 или 0 за овај ПЦЈ таг. Такође је изведена контрола и за ПЦЈ2.

Nivo1 (cm) је *IO field* контрола за постављање жељеног нивоа течности у резервоару. Ова контрола је везана за ПЦЈ таг Zeljeni nivo1. За ову контролу су везани и догађаји у *Event* који при завршеном уносу (*Input finished*) постављају мод ПИДа у аутоматски режим и искључују мануелни режим рада.



Слика 66. Догађаји везани за контролу

Поред ове контроле додата су још две *Button* контроле за повећање/смањење вредности ПЦЈ тага Zeljeni nivo1 за 1cm. И за ове две контроле су везани догађаји постављања у аутоматски режим и искључивања мануелног мода. Nivo2 је слична контрола која је везана за ПЦЈ2 тагове.

Napon1 (V) и **Napon2 (V)** је *IO field* контрола за постављање жељеног напона на пумпи М1 тј. М2. Као и за претходну контролу, везани су догађаји за укључивање мануелног режима и мода на ПИДу.

PID1 mod и **PID2 mod** је *Symbolic IO field* контрола већ описана на ХМИ за одабир режима рада ПИДа.

Nivo rez2 (cm) и **Nivo rez4 (cm)** је *IO field* контрола која је одабрана да буде само *Output* која приказује тренутни ниво речности у резервоару. Ова контрола је везана за ПЦЈ таг Nivo 1 cm тј. Nivo 2 cm.

Bar контрола је такође везана за ПЦЈ таг Nivo 1 cm тј. Nivo 2 cm и омогућава оператеру графичко праћење нивоа течности. Ову контролу је потребно додатно подесити што се тиче граница, боја и сл.

Режим рада је *Symbolic IO field* контрола као и **ПИД мод**, с тим што је постављена само као оутпут.

Napon M1 (V) је *IO field* контрола за приказивање тренутног напона на пумпи, постављена само као оутпут.

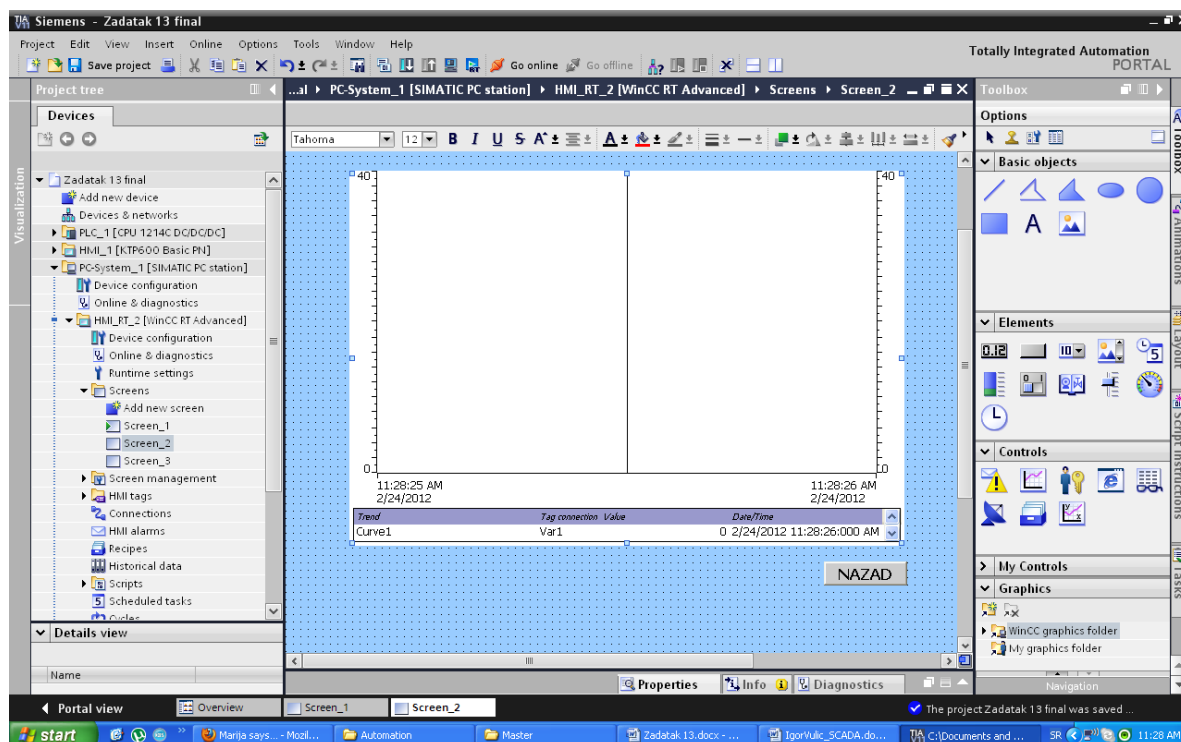
Pumpa M1 и **Pumpa M2** је *Gauge* контрола која графички приказује напон на пумпи М1, тј. М2.

Reset PLCs је *Button* контрола за коју су везани догађаји ресетовање рада ПИДа на ПЦЈ1 и ПЦЈ2 и брисања свих променљивих.

Постављена је још једна *Button* контрола за искључивање и укључивање контрола на ХМИ уређају.

Са десне стране је постављена контрола HTML browser која је подешена на стриминг сервер ИП камере, тј. на <http://147.91.203.60/main.htm> чиме се постиже да се процес визуелно прати.

График је *Буттон* контрола за стартовање екрана на коме је приказан Trend grafik.



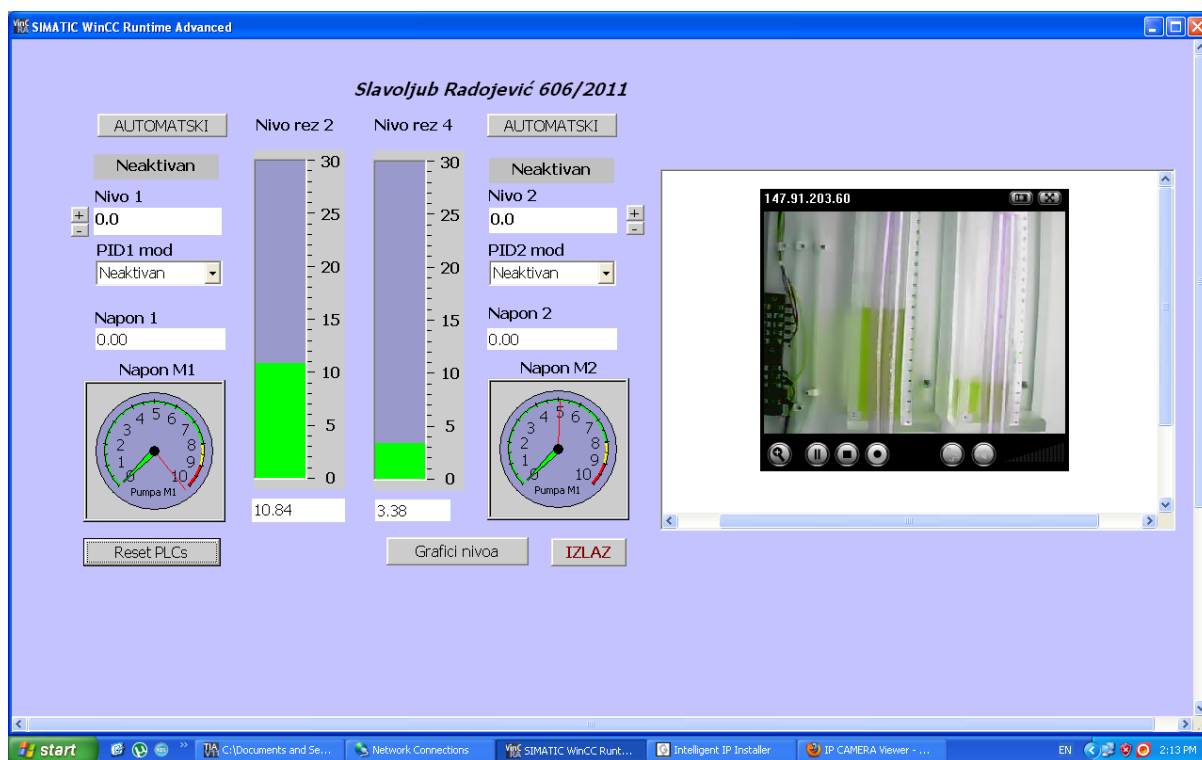
Слика 67. Постављање контроле графика на SCADA

ИЗЛАЗ је *Button* контрола за излазак из SCADA апликације.

Да би стартовали SCADA апликацију потребно је активирати *Start RunTime on PC*.



Слика 68. Изглед СКАДА на лаптопу

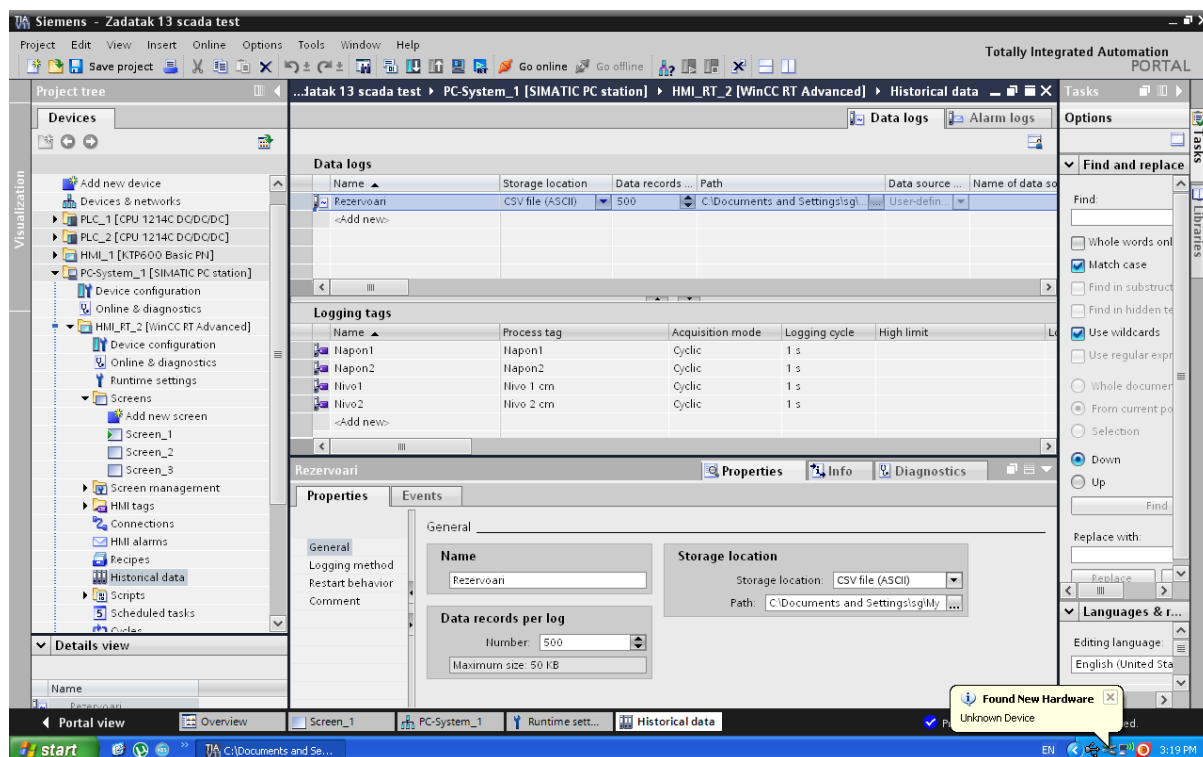


Слика 69. Коначан изглед SCADA кроз runtime

4.6. Праћење података

Праћење података је изведено преко *Project tree – PC_System – HMI_RT – Historical data*.

Потребно је креирати *Data log*, одредити тип фајла (у овом случају CSV, *Coma seperated Value*), број слогова и путању фајла у коме ће се праћени подаци записивати.



Слика 70. Праћење измерених података

У овом случају су постављени тагови *Napon1*, *Napon2*, *Nivo1* и *Nivo2*.

Литература

1. <http://www.automation.siemens.com/>
2. Siemens AG , „S7-1200_System_Manual_enUS”, 2009.
3. Siemens AG, „TIA_V11_en-US”, 2011.
4. Б. Стојановић, *Управљање кориачним мотором*, семинарски рад из предмета Пројектовање система аутоматског управљања, Машински факултет у Крагујевцу, 2002.
5. Видоје Луковић, *Програмирање и примена PLC контролера Siemens SIMATIC S7-1200*, дипломски рад, Машински факултет у Крагујевцу, 2010.
6. Иван Дачић, *Програмабилни логички контролер Siemens SIMATIC S7-200*, семинарски рад из Пројектовања САУ, Машински факултет у Крагујевцу, 2008
7. М. Равлић, Д. Лазић, М. Ристановић, М. Стефановић, В. Цвјетковић, Н. Бабајић, *„Документација техничког решења – Лабораторијски модел температурног процеса са кашњењем PT400 са управљањет и надзором преко Интернета“*, Машински факултет у Крагујевцу, 2009.
8. Марко Николић, *Опис и примена SCADA система Simatic WinCC flexible*, дипломски рад, Електротехнички Факултет Универзитета у Београду, 2007.
9. Милан Матијевић, Владимир Цвјетковић, Весна Ранковић, Миладин Стефановић, *Управљање лабораторијским процесима посредством Интернета*, Машински факултет у Крагујевцу, 2007
10. Милан Матијевић, Горан Јакуповић, Јелена Цар, *Рачунарски подржано мерење и управљање, универзитетски уџбеник*, Машински факултет у Крагујевцу, 2008
11. Н. Матић, *„Увод у индустријске PLC контролере“*, Микроелектроника Београд, 2007.