

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Simon Milavec

**Nadzorni sistem na spletnem strežniku programirljivega
logičnega krmilnika.**

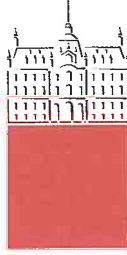
DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: izr. prof. dr. Uroš Lotrič

Ljubljana, 2012

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.



Št. naloge: 00315/2012

Datum: 03.09.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SIMON MILAVEC**


Naslov: **NADZORNI SISTEM NA SPLETNEM STREŽNIKU
PROGRAMIRLJIVEGA LOGIČNEGA KRMILNIKA
SUPERVISORY CONTROL SYSTEM IMPLEMENTED IN
PROGRAMMABLE LOGICAL CONTROLLER WEB SERVER**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

S prodorom protokola Ethernet na področje procesnega vodenja so zmogljivejši programirljivi logični krmilniki dobili spletne strežnike. K vgrajenim funkcionalnostim spletnega strežnika se lahko doda tudi uporabniške spletne strani. Raziščite možnosti, ki jih ponujajo spletni strežniki za izdelavo nadzornega sistema SCADA.

Mentor:


prof. dr. Uroš Lotrič



Dekan:


prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Simon Milavec, z vpisno številko **63080366**, sem avtor diplomskega dela z naslovom:

Nadzorni sistem na spletnem strežniku programirljivega logičnega krmilnika.

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvomizr. prof. dr. Uroša Lotriča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 24. septembra 2012

Podpis avtorja:

Na tem mestu se zahvaljujem staršem, sestri Miri, prijateljem in vsem ostalim, ki so me tekom študija spodbujali in mi stali ob strani.

Posebna zahvala gre mojemu mentorju za strokovno pomoč in napotke pri izdelavi diplomske naloge.

Kazalo

Povzetek

Abstract

1 Uvod	1
2 Opis uporabljene programske in strojne opreme	3
2.1 Programska oprema	3
2.1.1 SIMATIC STEP 7 Professional	3
2.1.2 SIMATIC WinCC flexible	4
2.1.3 Firebug	5
2.1.4 Adobe Dreamweaver	6
2.1.5 WebStorm	7
2.2 Strojna oprema	8
2.2.1 Programirljivi logični krmilnik	8
2.2.2 Linija z dvema napravama	10
3 Spletni strežnik krmilnika	11
3.1 Komunikacija s spletnim strežnikom PLK	13
3.2 Konfiguracija spletnega strežnika	14
3.3 Varnost	15
4 STANDARDNE SPLETNE STRANI	17
4.1 Dostop do standardnih spletnih strani	17
4.2 Postavitev standardnih spletnih strani	18
4.3 Prijava v sistem	19
4.4 Opis standardnih spletnih strani	19
4.4.1 Začetna stran	21
4.4.2 Identifikacija	21
4.4.3 Diagnostični medpomnilnik	22

4.4.4	Sporočila	22
4.4.5	Podatki o krmilniku.....	23
4.4.6	Komunikacije	24
4.4.7	Stanje spremenljivk.....	26
5	Uporabniško definirane spletne strani.....	27
5.1	Celoten potek izdelave uporabniško definiranih spletnih strani	28
5.2	Konfiguracija v programu STEP 7.....	29
5.3	Izdelava spletnih strani.....	30
5.3.1	HTML	30
5.3.2	CSS.....	30
5.3.3	JavaScript	31
5.4	Spletni kontrolni podatkovni blok.....	32
5.5	Uporaba ukaza WWW (SFC 99)	32
5.6	Interakcija med spletnim brskalnikom in krmilnikom.....	33
5.6.1	Prikaz spremenljivk na spletni strani	33
5.6.2	Pisanje spremenljivk iz spletne strani v krmilnik	33
5.7	Osveževanje podatkov na spletni strani	35
5.7.1	Ročno osveževanje.....	36
5.7.2	Osveževanje z HTML	36
5.7.3	Osveževanje z JavaScript.....	36
5.7.4	Osveževanje z uporabo tehnologije Ajax.....	36
5.8	Alternativne rešitve za uporabniško definirane spletne strani	39
5.8.1	Java Applets	39
5.8.2	Java Beans.....	39
5.8.3	Sm@rt Service z WinCC flexible	39
5.8.4	SIMATIC WinCC Web Navigator	39

6 Ukazi AWP (Automation Web Programming)	41
6.1 Sintaksa ukazov AWP	41
6.2 Kratek pregled ukazov AWP	42
6.3 Pravila za pisanje imen točk PLK v ukazih AWP	43
6.4 Podrobnosti in uporaba ukazov AWP	43
6.4.1 Branje spremenljivk (točk PLK)	43
6.4.2 Pisanje spremenljivk (točk PLK).....	44
6.4.3 Branje preko posebnih spremenljivk	45
6.4.4 Pisanje preko posebnih spremenljivk	46
6.4.5 Definiranje neštevnih tipov	48
6.4.6 Prirejanje vrednosti neštevnih tipov	48
6.4.7 Kreiranje dela podatkovnega bloka	49
6.4.8 Uvažanje dela podatkovnega bloka	50
7 Nadzorni sistem proizvodne linije	51
7.1 Branje spremenljivk.....	51
7.2 Vpisovanje spremenljivk	53
7.3 Prikazovanje slik	54
7.4 Pregled nadzornega sistema.....	54
7.4.1 Navigacija.....	54
7.4.2 Linija.....	55
7.4.3 Paketi	57
7.4.4 Alarmi	58
7.4.5 Recepti	59
7.4.6 Kontrola.....	60
8 Knjižnica WebSCADA	61
8.1 Izbira programskega jezika.....	61

8.2 Spletna tehnologija za izris grafičnih objektov	62
8.2.1 SVG (Scalable Vector Graphics)	62
8.2.2 HTML5 Canvas.....	63
8.2.3 Primerjava spletnih tehnologij SVG in HTML5 Canvas	64
8.3 Sestava knjižnice WebScada.....	64
8.4 Grafični gradniki	65
8.4.1 Vgradnja grafičnega gradnika v spletno stran.....	65
8.4.2 Lastnosti in izris grafičnega gradnika	66
8.4.3 Termometer	66
8.4.4 Števec	67
8.4.5 Polovični števec	67
9 Sklepne ugotovitve.....	69
Literatura	73

Povzetek

V diplomski nalogi preučujemo možnosti realizacije nadzornega sistema v spletnem strežniku programirljivega logičnega krmilnika. S prodorom protokola Ethernet na področje procesnega vodenja so zmogljivejši programirljivi logični krmilniki dobili spletne strežnike. V diplomskem delu tudi opisujemo spletni strežnik programirljivega logičnega krmilnika proizvajalca Siemens.

V prvem delu navajamo strojno in programsko opremo, ki smo jo uporabili za realizacijo nadzornega sistema. Temu sledi opis spletnega strežnika in vsega, kar nam le-ta ponuja. V nadaljevanju podrobneje pojasnujemo izdelavo uporabniško definiranih spletnih strani, s katerimi lahko ustvarimo nadzorni sistem; navajamo tudi vse, kar mora dober nadzorni sistem vsebovati in kakšne so možnosti realizacije nadzornega sistema z uporabniško definiranimi spletnimi stranmi. Temu za konec sledi opis realiziranega nadzornega sistema in knjižnice WebScada, ki vsebuje grafične gradnike za izdelavo nadzornega sistema in je bila razvita v okviru diplomskega dela.

Ključne besede:

nadzorni sistem, SCADA, spletni strežnik, programirljivi logični krmilnik.

Abstract

In this thesis, we study the feasibility of supervisory control and data acquisition (SCADA) system realisation in a web server of a programmable logic controller. With the introduction of Ethernet protocol to the area of process control, the more powerful programmable logic controllers obtained integrated web servers. The web server of a programmable logic controller, produced by Siemens, will also be described in this thesis.

Firstly, the software and the hardware equipment used for realisation of the web-based SCADA system is listed and described. It is followed by a description of a web server and its functionalities. Furthermore, the creation of user defined web pages, with which a SCADA system can be created, is explained in detail. Additional explanation is provided concerning the capabilities a good SCADA system has to offer and what are the possibilities of their realisation using user defined web pages.

Moreover, a realised SCADA system with WebScada library is described. The WebScada library, developed as a part of the thesis, contains graphic elements for formation of a SCADA system.

Keywords:

SCADA, web server, programmable logic controller.

1 Uvod

Danes živimo v dobi informatike, hitrega tehnološkega razvoja in naglice na vseh področjih. Hitrost razvoja se najbolj občuti v industriji saj se mora hitro prilagajati zahtevam trga in konkurence. Avtomatizacija proizvodnje ni več prednost pred konkurenco, ampak pogoj za njen obstanek na tržišču.

Avtomatizacija proizvodnje v osnovi pripomore k večji produktivnosti in nižanju stroškov, obenem pa prinaša večjo natančnost ter lažjo sledljivost izdelkov in kontrolo. Največja prednost avtomatizacije je zanesljivost, saj stroj dela ves delovnik popolnoma enako, česar človek ne zmore. Omogoča nam hiter odziv na zahteve trga, kar je v teh časih ključnega pomena. Posamezne procese v proizvodnji lahko avtomatiziramo s programiranimi logičnimi krmilniki (ang. Programmable Logic Controllers), v nadaljevanju krmilniki.

Pomemben del avtomatizacije proizvodnje je vodenje in nadzor avtomatiziranega ali ročnega procesa. Vmesnik za interakcijo človek-stroj (ang. Human Machine Interface) so razvili kot uporabniku prijazen prikaz delovanja sistema vodenja. Glavni namen vmesnika je učinkovito upravljanje, rokovanje in nadzor nad stroji in celotnim procesom. V začetku je nadzorni sistem sestavljala fiksno ožičena logika. Ožičenja vseh inštrumentov so bila speljana neposredno v nadzorno sobo. Operaterju je vmesnik preko različnih svetlobnih indikatorjev, števecv in drugih fiksnih elementov sporočal stanje procesa. Upravljanje procesa pa je bilo vodeno preko stikal, gumbov in drugih fiksnih elementov.

Z razvojem računalniške opreme se je nadzorni sistem preselil na osebne računalnike. Danes nam proizvajalci procesne opreme ponujajo različna programska orodja za izdelavo nadzornega sistema. Nadzorni sistem SCADA (ang. Supervisory Control And Data Acquisition) izbira informacije iz programiranih logičnih krmilnikov in jih prikaže na enostaven in pregleden način. Omogoča operaterju poseganje v proces. Pomembna funkcija nadzornega sistema je generiranje alarmov, ki pritegnejo pozornost operaterja in mu omogoča, da hitro reagira na nepričakovani dogodek. Operater preko grafičnega uporabniškega vmesnika z miško, tipkovnico ali z dotikom na zaslon upravlja in nadzira proces.

Danes živimo v dobi interneta, kjer so digitalne vsebine dostopne na vsakem koraku. Tega se zavedajo tudi proizvajalci procesne opreme. S prodorom protokola Ethernet na področje procesnega vodenja so krmilniki dobili spletne strežnike. Spletni strežniki nam omogočajo

izgradnjo nadzornega sistema, ki komunicira z procesom preko spletno temelječega uporabniškega vmesnika. Ponujajo nam nov vmesnik človek-stroj in s tam odpiranje novih možnosti za izgradnjo nadzornega sistema. Spletne strani sestavljajo grafični uporabniški vmesnik. Operater preko spletnega brskalnika komunicira in upravlja proces.

V diplomski nalogi preučujemo možnosti za realizacijo nadzornega sistema preko uporabniško definiranih spletnih strani, ki nam jih ponuja proizvajalec procesne opreme Siemens.

Implementirali smo nadzorni sistem preko uporabniški definiranih spletnih strani za nadzor modela industrijske linije. Napisali smo tudi knjižnico poimenovano WebSCADA, ki vsebuje grafične gradnike za izdelavo nadzornega sistema preko spletnih strani.

V prvem poglavju opisujemo strojno in programsko opremo, ki smo jo uporabili za realizacijo nadzornega sistema. V drugem poglavju opisujemo zmogljivost, varnost, in vse kar nam ponuja spletni strežnik krmilnika. Standardne spletne strani spletnega strežnika krmilnika so opisane v tretjem poglavju. Četrto poglavje vsebuje celoten opis izdelave uporabniško definiranih spletnih strani in njene alternative. V petem poglavju opisujemo ukaze AWP, ki se uporabljajo pri izdelavi uporabniško definiranih spletnih strani. V petem poglavju opisujemo realiziran nadzorni sistem proizvodne linije in njegovo izdelavo. Šesto poglavje opisuje knjižnico WebSCADA, njeno implementacijo, izdelavo, namen in uporabo. V zaključnem poglavju primerjamo realizacijo nadzornega sistema s programom WinCC flexible. Opisujemo tudi namen nadzornega sistema, prednosti oziroma slabosti in težave pri realizaciji.

2 Opis uporabljene programske in strojne opreme

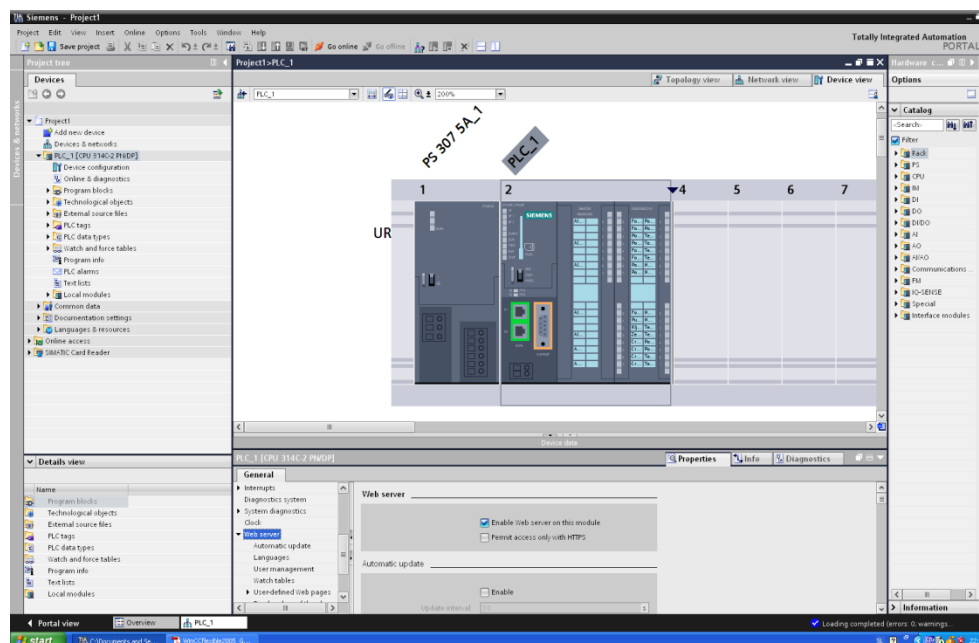
V nadaljevanju sledi opis programske in strojne opreme, ki smo jo uporabili za realizacijo nadzornega sistema SCADA.

2.1 Programska oprema

2.1.1 SIMATIC STEP 7 Professional

Simatic Step 7 Professional (slika 1) je Siemensovo razvojno okolje za programirljive logične krmilnike SIMATIC. Vsebuje programska orodja za implementacijo vseh faz projekta za avtomatizacijo [5]. Je skupek programskih orodji, ki omogočajo:

- konfiguracijo in nastavljanje parametrov strojne opreme,
- definiranje spremenljivk,
- programiranje,
- testiranje,
- simulacijo,
- opazovanje dogajanja na krmilniku,
- povezovanje spremenljivk z vhodi in izhodi,
- dokumentacijo.



Slika 1: Uporabniški vmesnik programa Step 7.

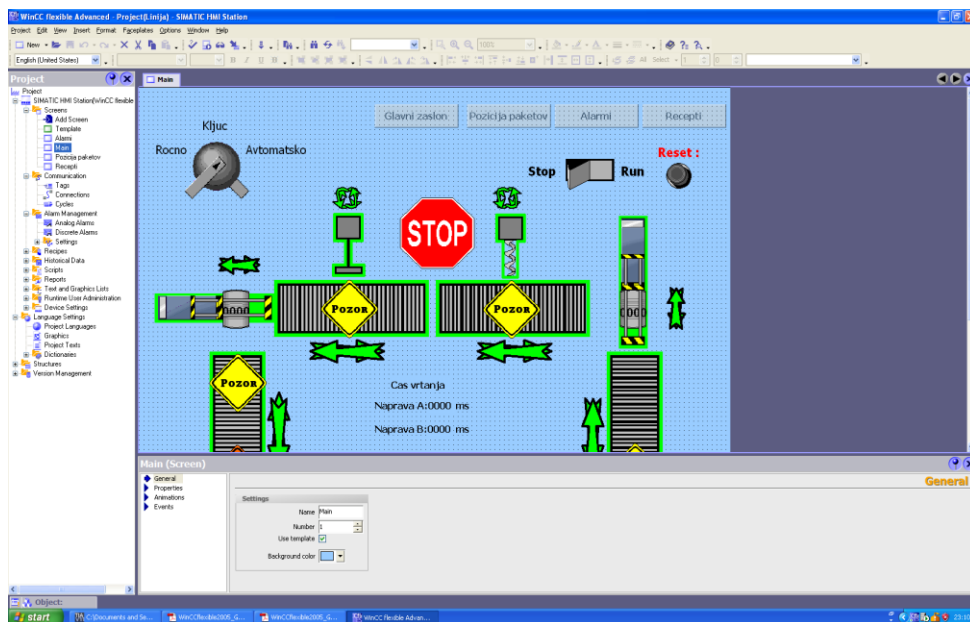
Simatic Step 7 je del okolja Siemens TIA (ang. Totally Integrated Automation) portal, ki predstavlja inženirsko razvojno okolje [9]. Vključuje vsa potrebna programska orodja za izdelavo procesnega projekta – od konfiguracije in programiranja krmilnika do izdelave nadzornega sistema.

Omogoča nam programiranje v enem od standardnih jezikov. V našem primeru smo uporabili Lestvični diagram (LAD); razvojno okolje smo uporabljali predvsem za konfiguracijo spletnega strežnika, generiranje podatkovnega bloka WWW in pošiljanje podatkov na krmilnik.

2.1.2 SIMATIC WinCC flexible

Simatic WinCC flexible (slika 2) programsko orodje pokriva celotno področje inženirstva in vizualizacije vmesnika človek – stroj [8]. Namenjen je Siemensovim krmilnikom serije SIMATIC. Knjižnice in čarovniki nam olajšajo in pospešijo izdelavo nadzornega sistema. Uporabniku prijazen vmesnik razvijalcu ponuja:

- knjižnice grafičnih objektov,
- inteligentna orodja za enostavno ustvarjanje projektov,
- podporo večjezičnosti z avtomatskim prevajanjem besedil,
- konfiguracijo za stiskanje podatkov,
- čarovnika za izdelavo prikaza alarmov in receptov,
- testiranje in simulacijo itd.



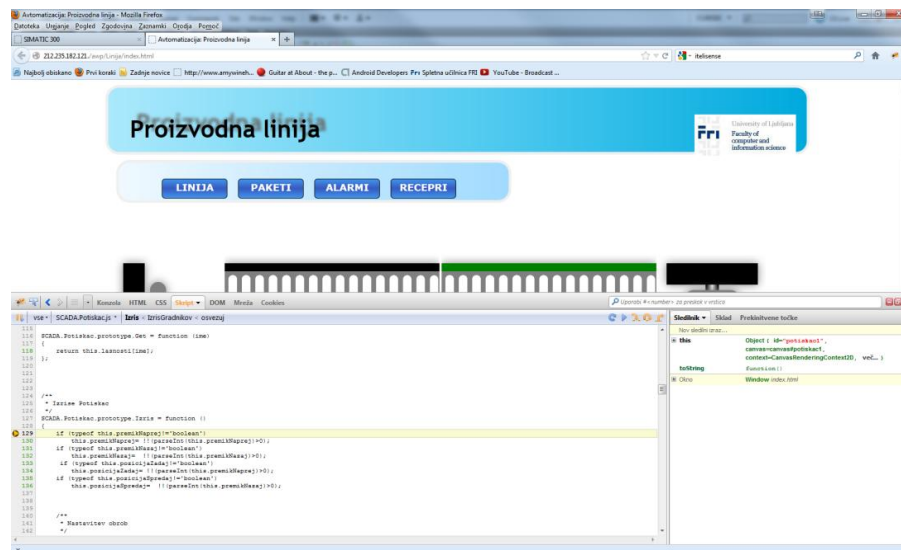
Slika 2: Izdelava nadzornega sistema s programom WinCC.

Programskega orodja WinCC v našem projektu nismo uporabili za razvoj, temveč le za primerjavo. Primerjali smo izvajanje in izdelavo nadzornega sistema s programom WinCC in z uporabniškimi spletnimi stranmi, ki so shranjene na spletnem strežniku krmilnika.

2.1.3 Firebug

Firebug je spletno razvojno orodje, namenjeno brskalniku Firefox. Ponuja nam različna orodja, ki nam lajšajo spletni razvoj [2]. V realnem času omogoča spremljanje in spreminjanje:

- predlog, ki določajo izgled spletnih strani CSS (ang. Cascade Style Sheets),
- objektnega modela dokumenta DOM (ang. Document Object Mode),
- dokumenta HTML (ang. Hyper Text Markup Language),
- zahtev XHR (ang. eXtended markup lanhuage HTTP Request) in
- skriptnega jezika JavaScript.



Slika 3: Uporaba razhroščevalnika Firebug.

Firebug vključuje zmogljiv razhroščevalnik (slika 3) za skriptni jezik JavaScript. Pri razvoju uporabniških spletnih strani nam je bil v veliko pomoč, saj omogoča:

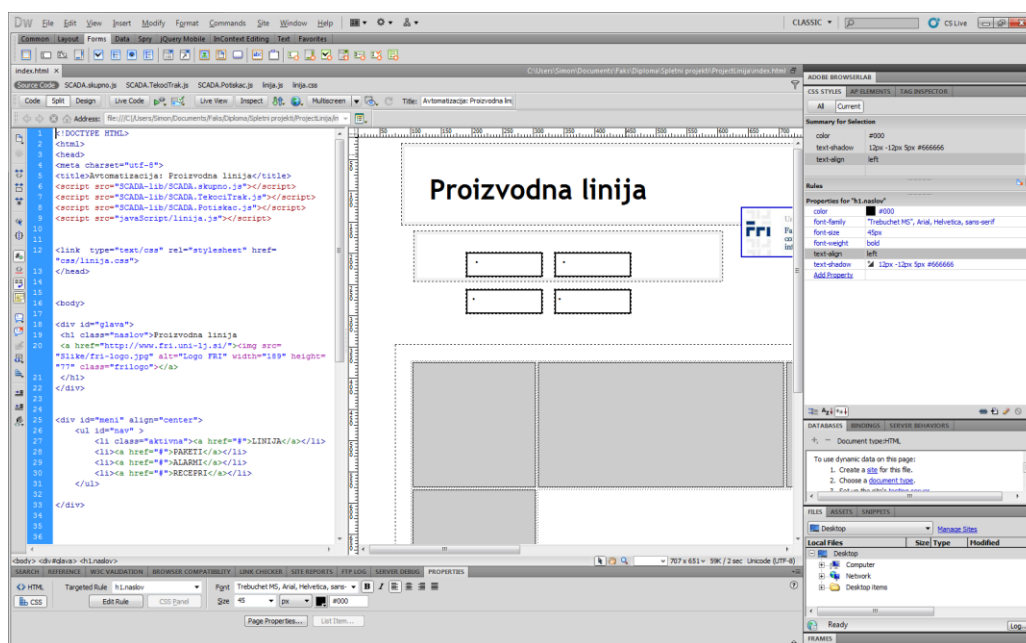
- prikaz napak,
- zaustavljanje izvajanja,
- izvajanje po korakih,
- opazovanje vrednosti spremenljivk in
- enostavno iskanje skript.

Pri razvoju in testiranju spletnih strani je nepogrešljivo programsko orodje za nadzor oziroma spremljanje aktivnosti omrežja. Orodje smo uporabili za testiranje pravilnosti prenosa iz spletnega strežnika krmilnika, prav tako pa tudi za merjenje časa od zahteve spletnega brskalnika do prenosa podatkov. Orodje omogoča:

- spremljanje časa prenosa posamezne datoteke,
- pregled glede na vrsto datoteke, ki se prenaša oziroma se je prenesla,
- pregled glave protokola za prenos hiperteksta HTTP (ang. Hypertext Transfer Protocol) in
- nadzor nad zahtevki za prenos datotek XML (ang. Extensible Markup Language).

2.1.4 Adobe Dreamweaver

Programska oprema Dreamweaver proizvajalca Adobe omogoča oblikovalcem in razvijalcem lažjo izdelavo in oblikovanje spletnih mest. Spletno stran lahko oblikujemo vizualno ali neposredno s pisanjem kode. Dreamweaver spada med skupino orodij, ki prikazuje takojšnje spremembe vizualne kode oziroma funkcionalnosti programa (ang. What You See Is What You Get - WYSIWYG) [1]. Namenjen je predvsem za razvoj v HTML in CSS okolju, ponuja pa nam dobro razvojno okolje za programske jezike, ki se izvršujejo tako na strani strežnika (PHP, JSP, ASP itd.) kot tudi na strani uporabnika (JavaScript). Dreamweaver nam je olajšal in pospešil razvoj uporabniško definiranih spletnih strani, ne le same strukture strani, temveč tudi njenega vizualnega dela.

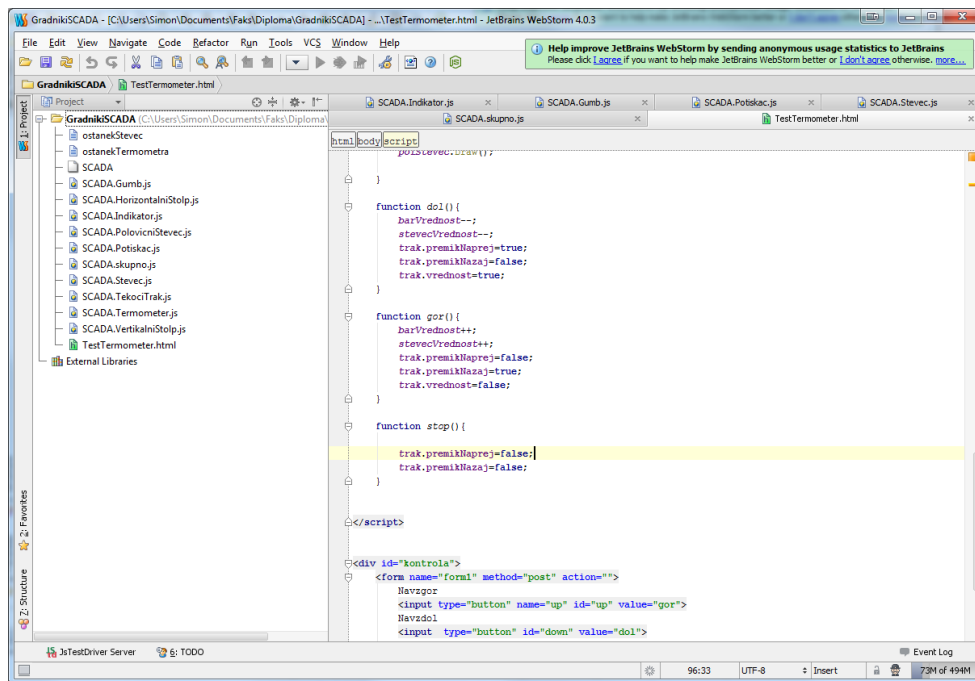


Slika 4: Uporabniški vmesnik programske opreme Adobe Dreamweaver.

Uporabniški vmesnik (slika 4) je enostavno zasnovan in uporabniku nudi dober pregled nad projektom. Uporabnik lahko izbira med več pogledi; zgolj na vizualni del projekta, pogled na kodo ali pa oboje naenkrat. Omogoča vizualno programiranje, kjer na delovno površino enostavno dodajamo gradnik, pri tem pa se v ozadju generira koda. Programska oprema je plačljiva, zato smo v namen izdelave spletnih strani uporabljali testno različico programa.

2.1.5 WebStorm

WebStorm (slika 5) je eden boljših komercialnih urejevalnikov kode za skriptni jezik JavaScript. Vgrajeno ima razvojno okolje za programski jezik JavaScript. Uporabniku nudi enostavno navigacijo skozi zbirko datotek in po izvorni kodi kreiranega projekta. Programerju olajša delo z iskanjem in prikazovanjem napak med pisanjem kode, programiranje pa nam olajša tudi z vgrajeno funkcijo nudenja namigov [3]. V primeru napake nam prav tako ponudi hitro rešitev problema. Pri samem programiranju najbolj pride do izraza funkcija samodejnega dopolnjevanja kode oziroma zaključevanja ukazov, preglednost kode pa nam izboljša z funkcijo barvanja le-te.



Slika 5: Razvojno okolje WebStorm.

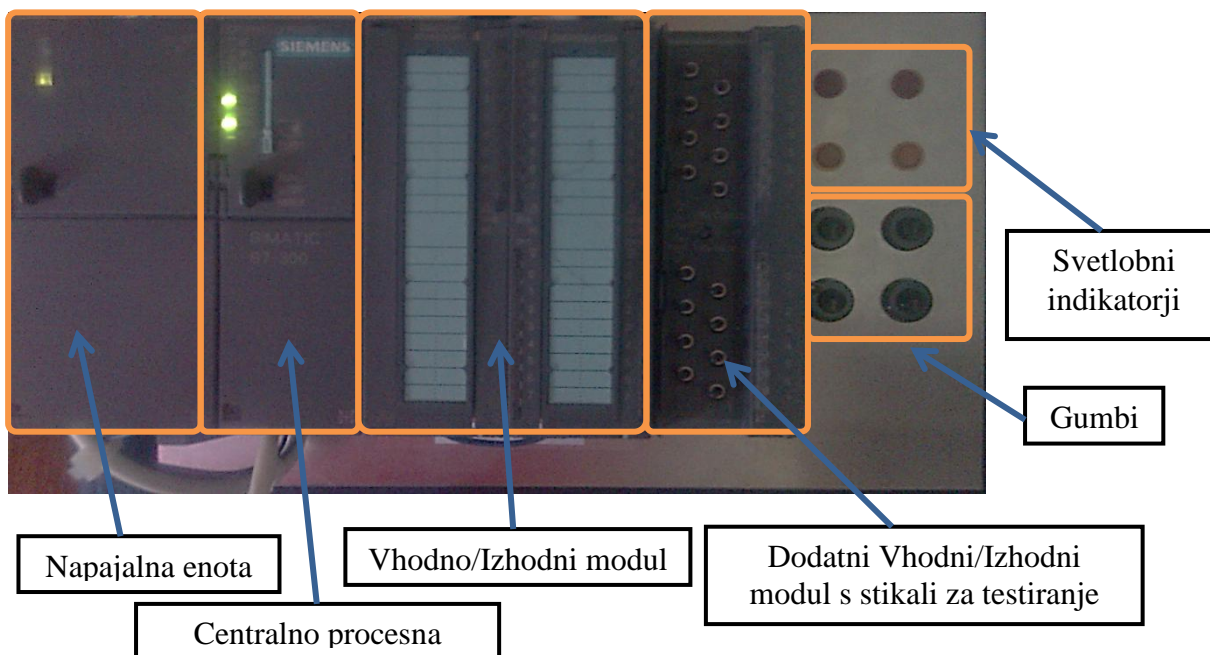
Program nam je olajšal delo pri izdelavi grafičnih gradnikov, ki smo jih razvili v namen uporabniku prijaznejšega prikaza podatkov oziroma stanja, pripomogel pa je tudi k izdelavi dinamičnega dela posamezne spletne strani (kreiranje gradnikov, preverjanje pravilnosti vnosa itd.).

2.2 Strojna oprema

2.2.1 Programirljivi logični krmilnik

V namen izdelave nadzornega sistema smo uporabljali programirljivi logični kontrolnik serije SIMATIC S7-300 proizvajalca Siemens (slika 6).

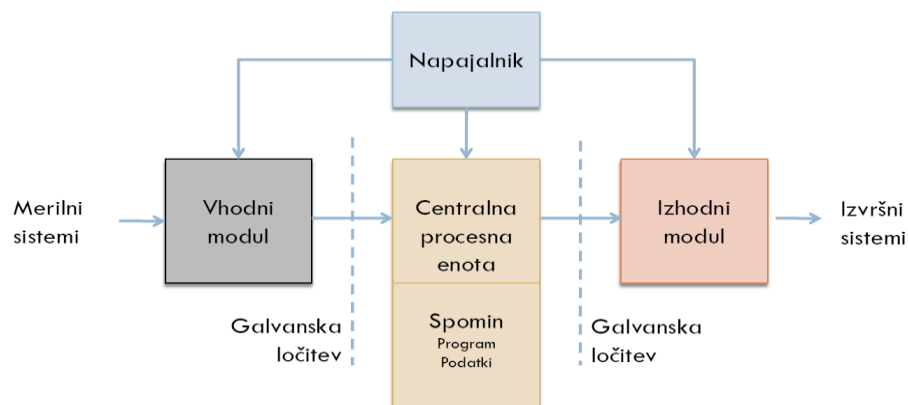
Krmilnik je digitalni računalnik namenjen avtomatizaciji elektromehaničnih procesov kot so krmiljenje strojev, proizvodne linije itd [16]. Večinoma se uporabljajo v industrijske namene, saj so bolj odporni na zunanje dejavnike (temperatura, vlaga, vibracije, prah, električne motnje itd.). Vgrajenih imajo več izhodov in vhodov, tako digitalnih kot analognih, po potrebi pa lahko vhodne in izhodne module razširimo. Velikega pomena je, da se program na krmilniku izvaja v realnem času.



Slika 6: Krmilnik Siemens SIMATIC S7-300.

Arhitektura krmilnika je podobna arhitekturi drugih računalnikom (slika 7) in je sestavljena iz naslednjih komponent [14]:

- centralno procesne enote,
- napajalne enote in
- vhodnega oziroma izhodnega modula.



Slika 7: Arhitektura krmilnika.

Novejši krmilniki proizvajalca Siemens imajo tudi vgrajeno funkcionalnost spletnega strežnika, ta faktor pa je bil tudi najpomembnejši pri izbiri centralno procesne enote krmilnika. Drugi dejavnik za izbiro je bila podpora uporabniško definiranim spletnim stranem, saj smo samo tako lahko realizirali naš nadzorni sistem. Krmilnik ima vgrajen komunikacijski standard Profinet, ki preko protokola TPC/IP omogoča komunikacijo s spletnim strežnikom krmilnika.

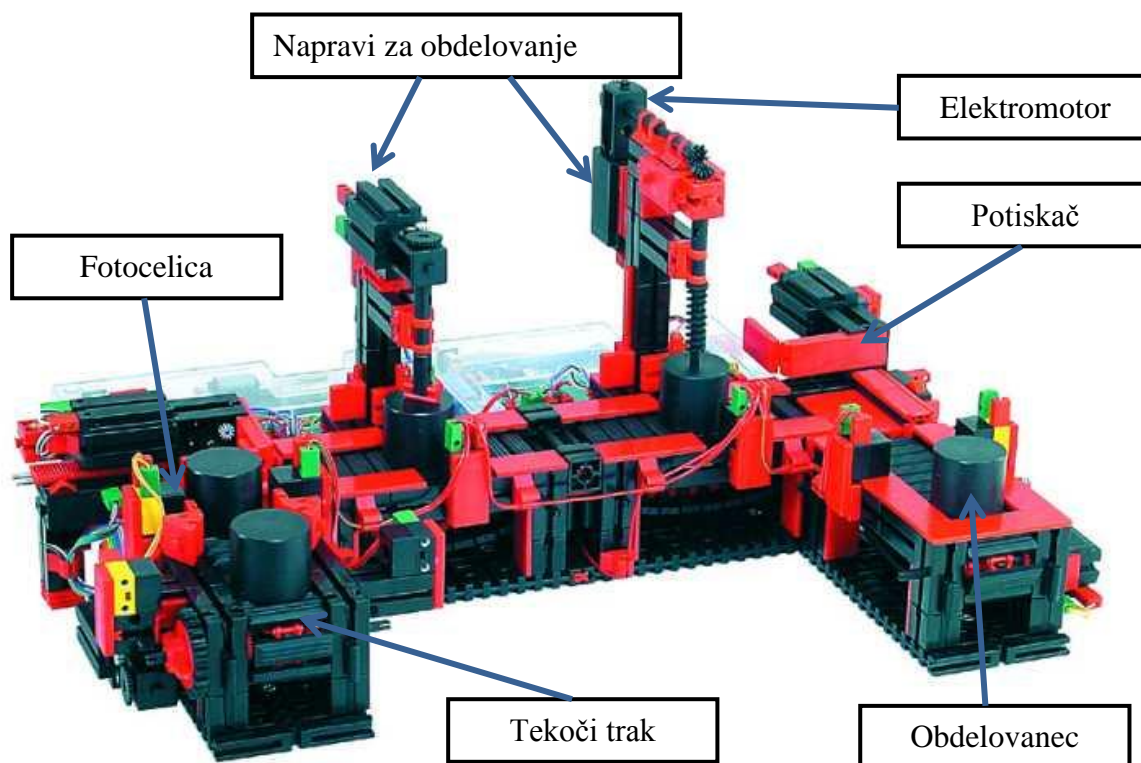
Tehnične specifikacije krmilnika, ki so pomembne za naš projekt, so opisane v spodnji tabeli (tabela 1).

Centralno procesna enota (oznaka)	314C-2PN/DP
Delovni spomin (velikost)	192 kB
Število digitalnih izhodov	16
Število digitalnih vhodov	24
TPC/IP	Ja, preko integriranega PROFINET vmesnika
Programski paket	STEP 7 verzija 5.5 ali višja
Spletni strežnik (podpora)	Ja
Število spletnih (HTTP) uporabnikov	5
Uporabniško definirane spletne strani	Ja

Tabela 1: Tehnična specifikacija krmilnika.

2.2.2 Linija z dvema napravama

Linija z dvema napravama (slika 8) je pomanjšan model industrijske linije proizvajalca Fischertechnik. Fischertechnik izdeluje didaktične pripomočke, ki so pomanjšani modeli industrijskih naprav [17]. Linijo najprej sestavljajo štirje tekoči trakovi, ki po liniji pomikajo obdelovanec. Tega potisneta na tekoči trak dva potiskača, ki imata vsak po dve končni stikali, ki določata njuni skrajni legi. Obdelovanec obdelujeta dve napravi. Tekoče trakove, potiskača in napravi za obdelovanje poganja osem ločenih elektromotorjev, linija pa je opremljena tudi s petimi fotocelicami, s katerimi določimo pozicijo obdelovanca.



Slika 8: Industrijska linija Fischertechnik.

V našem projektu smo linijo uporabili kot simulator oziroma za manipulacijo s podatki, ki smo jih nato prikazali v nadzornem sistemu. S tem smo lahko preverili, če naš nadzorni sistem služi svojemu namenu, pravilno predstavi podatke in če so ti predstavljeni v realnem času.

3 Spletni strežnik krmilnika

Spletni strežnik gosti spletne strani; je programska oprema, ki omogoča prenašanje dokumentov preko protokola HTTP do spletnega brskalnika.

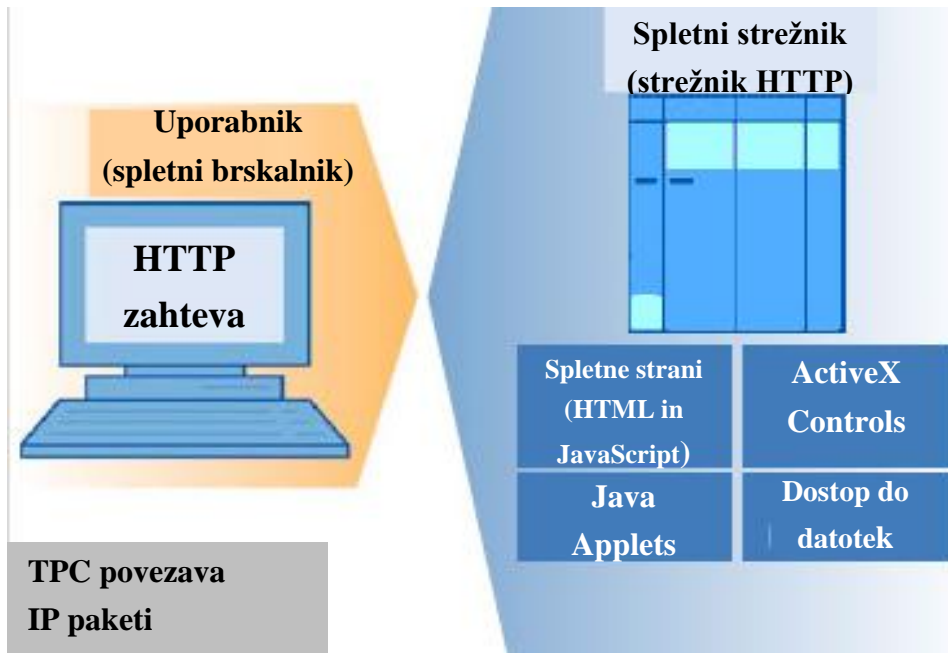
Novejši programirljivi logični kontrolniki imajo vgrajen spletni strežnik. Spletni strežnik nam preko mrežnega standarda Ethernet omogoča spremljanje in spreminjanje spremenljivk oziroma točk krmilnika. Spletni strežnik predstavlja vmesnik do krmilnika.

Spletni strežnik krmilnika proizvajalca Siemens ponuja:

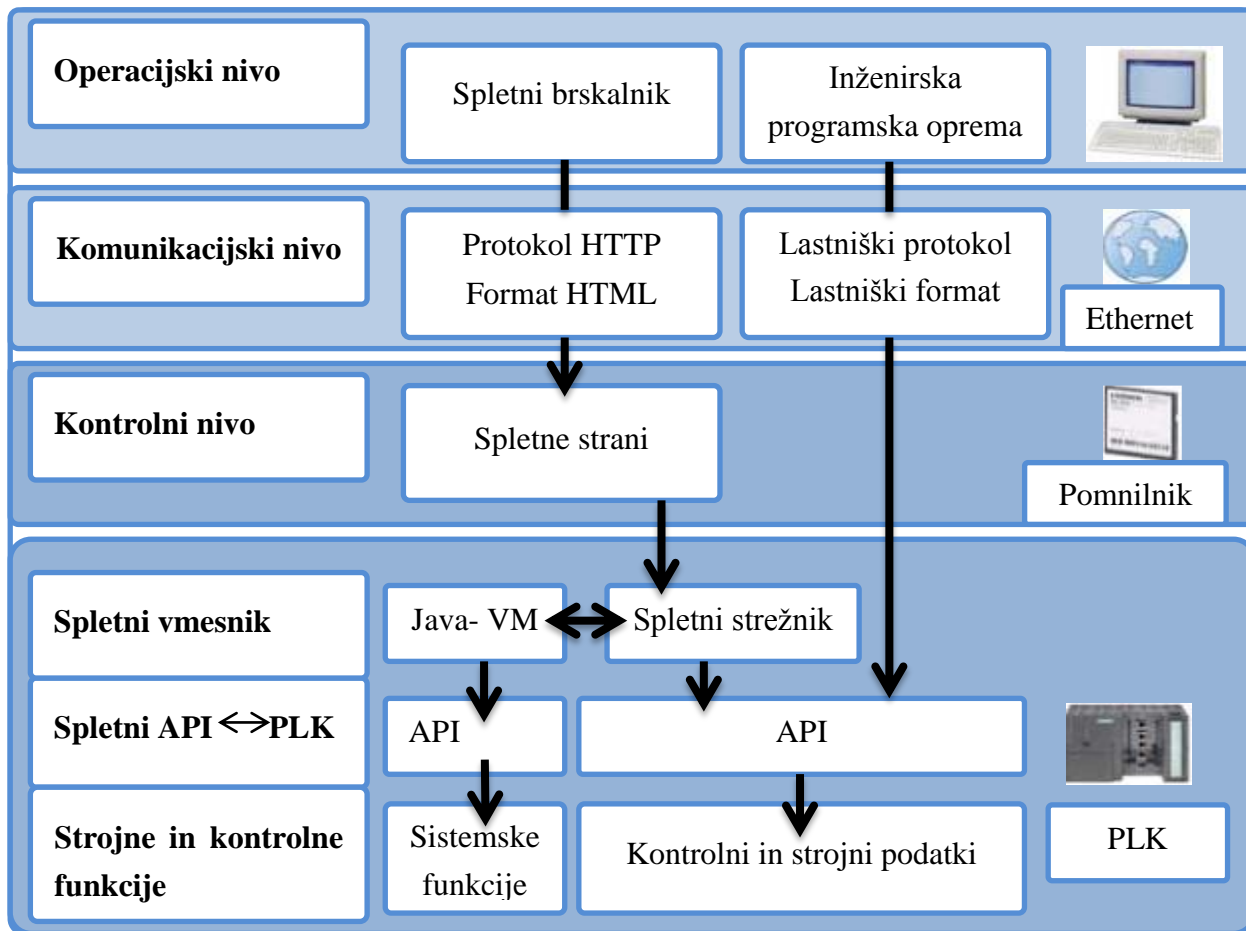
- dostop do PLK preko spletnega brskalnika,
- spremljanje spremenljivk,
- uporabniško definirane spletne strani,
- standardne spletne strani za spremljanje informacij o napravi, modulu, komunikaciji itd.,
- sposobnost priklicati diagnostične informacije.

Slika 9 prikazuje dostop do spletne strani na krmilniku. Spletni brskalnik pošlje zahtevo spletnemu strežniku. Spletni strežnik pošlje informacije in navodila za izris strani (HTML, JavaScript) spletnemu brskalniku [15]. Spletni strežnik lahko pošilja tudi dodatna navodila za izris animacije, formularjev itd. (ActiveX, Java Applets); spletni brskalnik uporabi ta navodila za kreiranje in izris spletne strani na zaslon, kar predstavlja naš vmesnik do krmilnika.

Namenska programska oprema za razvoj nadzornih sistemov SCADA omogoča direkten dostop do podatkov in sistemskih funkcij PLK-ja iz razvojnega okolja. Slabost izdelave takega nadzornega sistema sta predvsem cena programske opreme in izobraževanje, prednosti izdelave nadzornega sistema preko spletnih strani pa sta cenejša izvedba in dostopnost. Za dostop potrebujemo samo spletni brskalnik, vse potrebne datoteke za dostop pa so shranjene na spletnem strežniku PLK-ja. Slika 10 po plasteh prikazuje dva različna dostopa do podatkov na PLK-ju.



Slika 9: Spletni strežnik PLK.



Slika 10: Prikaz dveh različnih dostopov do PLK po plasteh.

3.1 Komunikacija s spletnim strežnikom PLK

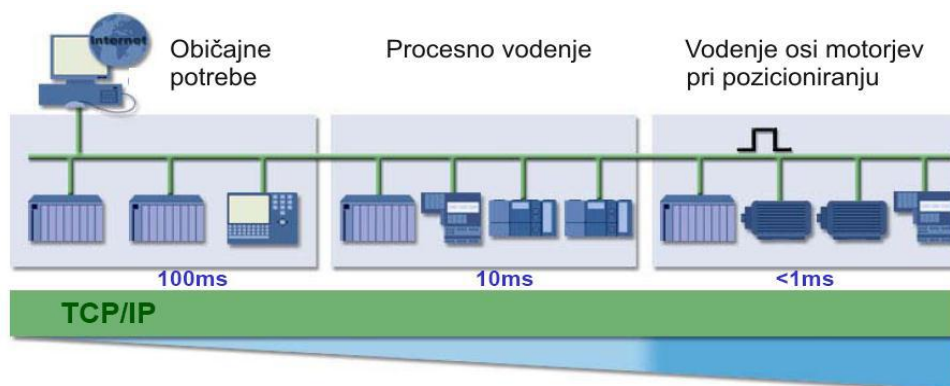
Komunikacija s spletnim strežnikom poteka preko mrežnega standarda Ethernet. Ethernet je najpopularnejši protokol za lokalna računalniška omrežja. Ta industrijski standard so prevzeli mnogi proizvajalci omrežne strojne opreme, zato se v zadnjem času pospešeno širi tudi v industrijsko okolje [10]. Več tehnik je uporabljenih za prilagoditev standarda Ethernet industrijskim zahtevam, ki med drugim zahtevajo odziv v realnem času (slika 11). Več razlogov se podaja pri uporabi standarda Ethernet v industrijske namene, ti so [20]:

- je že pisarniški standard,
- se že uporablja za visokonivojsko komunikacijo v procesni avtomatiki,
- je primeren za komunikacijo med področnimi enotami.

Prednosti Etherneta so:

- kakovostne in zanesljive povezave,
- stalna povezljivost,
- večja razdalja,
- kratki odzivni čas,
- enostavna vertikalna integracija z višjimi sistemi vodenja,
- omrežje enakovrednih računalnikov,
- boljša skupna uporabnost.

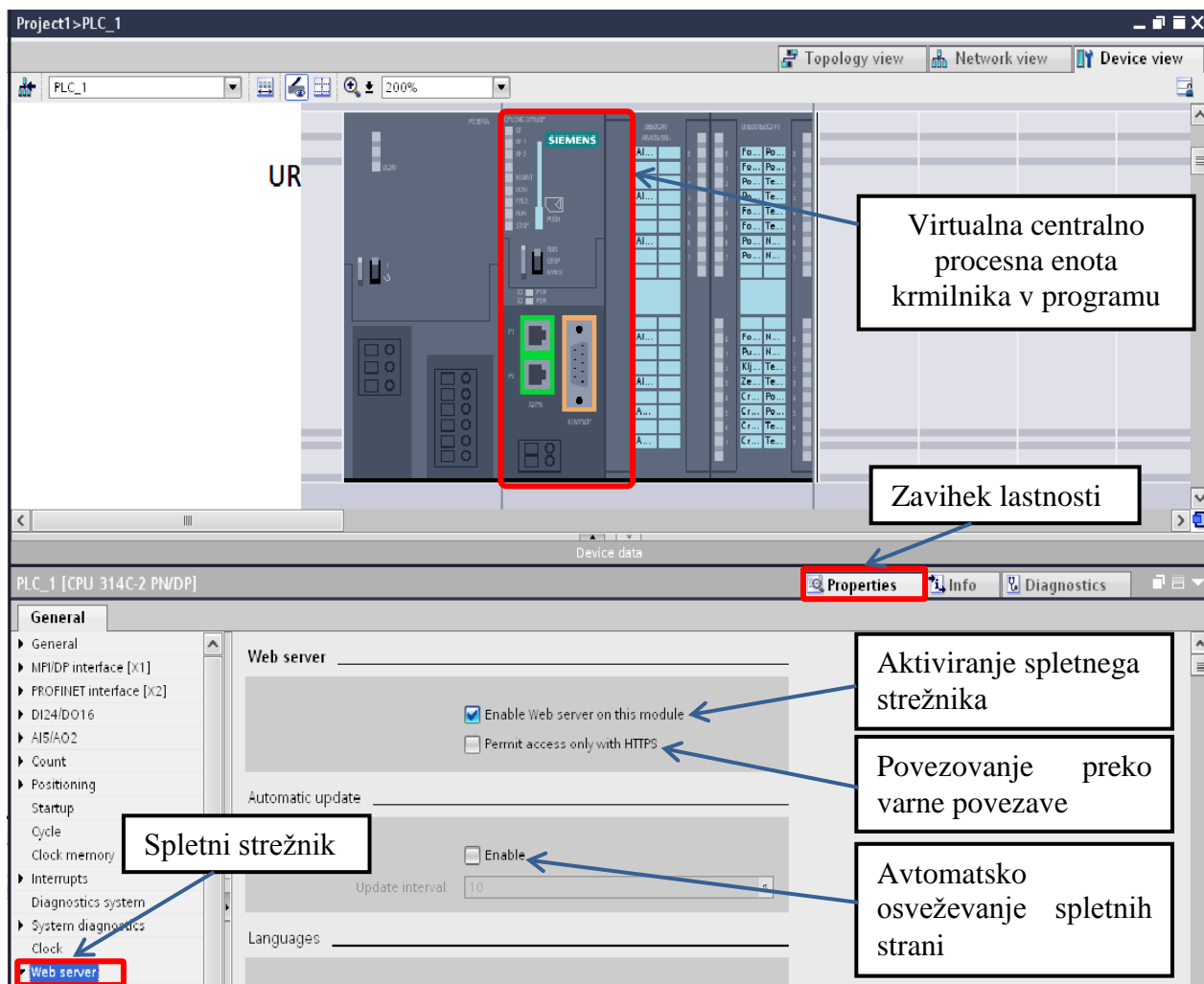
Profinet je odprt mrežni standard, osnovan na standardu Ethernet, ki za komunikacijo uporablja protokol TCP/IP (ang. Transmission Control Protocol/Internet Protocol) in standarde IT, ki ga dopolnjujejo z različnimi mehanizmi za delovanje v realnem času. Spletni strežnik na krmilnik preko vmesnika Profinet komunicira s spletnim brskalnikom preko osnovnega protokola TCP/IP.



Slika 11: Prikaz omrežnih časovnih zahtev v industriji.

3.2 Konfiguracija spletnega strežnika

Spletni strežnik na krmilniku proizvajalca Siemens omogočimo in konfiguriramo s programom STEP 7. S programom ustvarimo nov projekt, pod zavihkom "nastavitve naprave" (ang. device configuration) projektu dodamo pripadajočo strojno opremo. Nato konfiguriramo našo stojno opremo, torej pravilno nastavimo omrežne nastavitve. Program nam izriše virtualno strojno opremo, ki jo lahko interaktivno izbiramo. Izberemo centralno procesno enoto PLK, pod zavihkom "nastavitve" (ang. properties) se nam odpre nova nastavitvena stran. Na levi strani se nam izpiše tudi meni, ki ponuja vse nastavitve centralno procesne enote krmilnika; v meniju izberemo lastnost "spletni strežnik" (ang. web server). Odpre se nam nastavitvena stran (slika 12) za spletni strežnik.



Slika 12: Konfiguracija spletnega strežnika v programu STEP 7.

Nastavitvena stran spletnega strežnika nam ponuja:

- aktiviranje spletnega strežnika,
- varnostne nastavitve spletnega strežnika,
- jezikovne nastavitve,
- nalaganje uporabniško definiranih spletnih strani,
- nastavitve osveževanja standardnih spletnih strani,
- nastavitve dostopa uporabnikom,
- nastavitve spremljanja tabel,
- nastavitve prikazovanja alarmov.

Nastavitve prenesemo na krmilnik s klikom na gumb "naloži na napravo" (ang. download to device), ki se nahaja v orodni vrstici programa.

3.3 Varnost

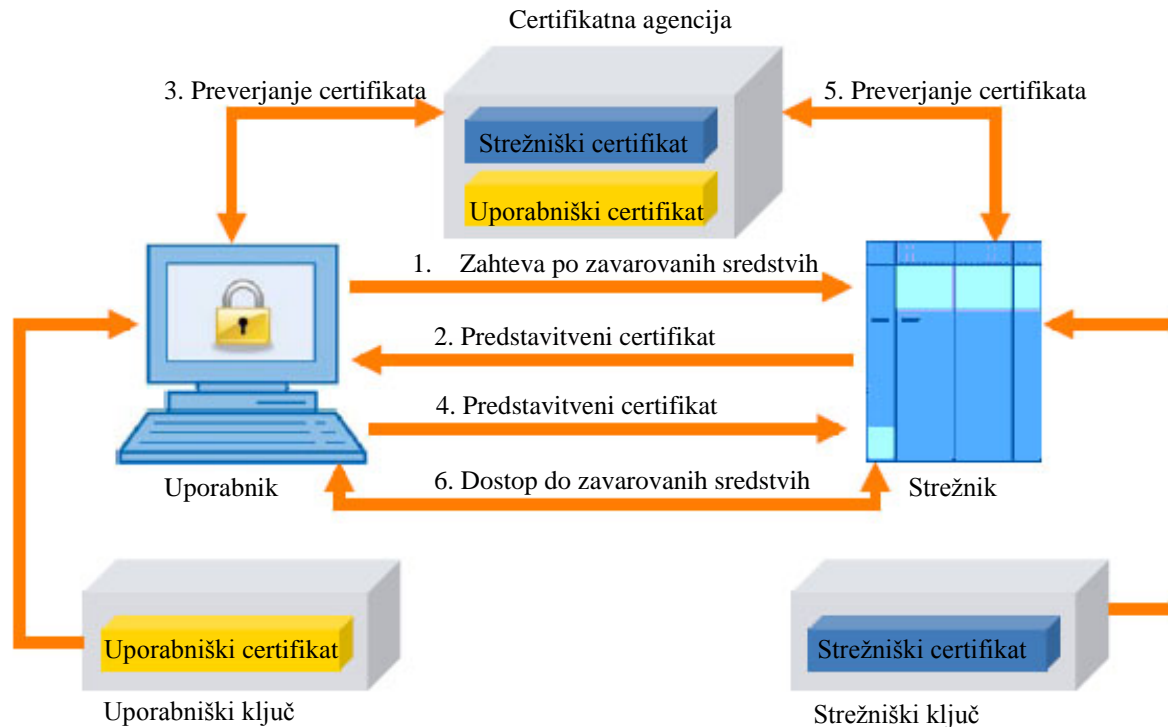
Povezovanje krmilnikov v večja omrežja odpira vprašanje o varnosti, saj je v industriji varnost prvotnega pomena. Različne varnostne mehanizme lahko uporabimo za zaščito industrijskega omrežja pred nepooblaščenim dostopom.

Zaradi možnosti vdora v sistem so vpeljani naslednji varnostni mehanizmi:

- spremenljivke krmilnika je mogoče spreminjati preko standardnih in uporabniško definiranih spletnih strani le, če je uporabnik prijavljen v sistem,
- uporaba varnostnih gesel za varovanje uporabniških računov,
- uporaba varne povezave pri dostopu iz interneta ali brezžičnega omrežja.

Spletni strežnik uporablja za varno povezavo protokol HTTPS (ang. Hyper Text Transfer Protocol Secure) [10]. Varno povezavo aktiviramo v nastavitvah spletnega strežnika v programu STEP 7. HTTPS je razširjen komunikacijski protokol za varno povezavo v računalniških omrežjih. Sestavlja ga več protokolov, na vrhu sta protokola SSL/TLS (Secure Sockets Layer/Transport Layer Security), ki s kriptiranjem zagotavljata varno povezavo preko interneta. Spletni strežnik krmilnika uporablja protokol SLL za enkripcijo podatkov, ki se prenašajo iz krmilnika.

Varna povezava (slika 13) poteka, ko spletni brskalnik pošlje zahtevo za spletno stran z varovanim prenosom. Strežnik se predstavi z javnim ključem in digitalnim potrdilom oziroma certifikatom. Ko se spletni strežnik predstavi, spletni brskalnik ustvari ključ za enkripcijo s simetričnim ključem, ga zakodira s strežnikovim javnim ključem in mu ga pošlje. Strežnik



Slika 13: Potek varne komunikacije z PLK-jem.

sprejeti ključ dekodira s privatnim ključem; spletni brskalnik se podobno kot strežnik predstavi z digitalnim potrdilom. Od tu naprej poteka povezava preko veliko hitrejših enkripcije.

Digitalno potrdilo prevzamemo na predstavitveni spletni strani krmilnika ali ga prenesemo, ko nas spletni brskalnik opozori, da stopamo v varno povezavo. Digitalno potrdilo nato uvozimo v naš spletni brskalnik; če je z digitalnim potrdilom kaj narobe, nas na to opozori brskalnik. Pri dostopu do spletnih strani na krmilniku preko varne povezave morajo biti izpolnjene naslednje zahteve:

- v PLK-ju mora biti nastavljen trenutni čas,
- v spletnem brskalniku mora biti veljavno digitalno potrdilo.

Z nadzorom dostopa lahko omejujemo oziroma nadzorujemo posamezne uporabnike. Uporabnike razvrstimo v skupine, katerim določimo dostopne pravice in dovoljenja.

Varnost v podjetju izboljšamo z navidezno ločitvijo poslovnega in industrijskega omrežja s tehnologijo navidezno zasebnega omrežja (ang. virtual private network-VPN) in preusmerjanjem vrat (ang. Port forwarding). Kontrolniki lahko izboljšajo varnost v industriji z uporabo varne povezave in uporabniške identifikacije.

4 STANDARDNE SPLETNE STRANI

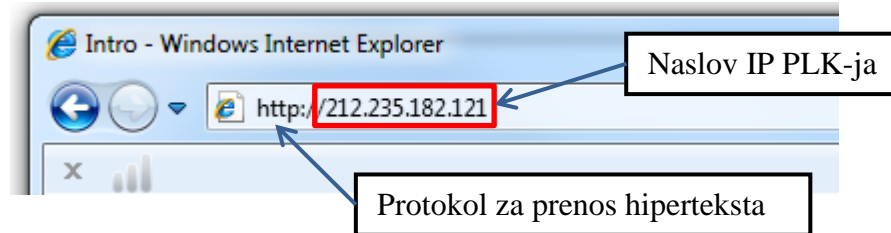
Siemensovi programirljivi logični krmilniki s podporo spletnega strežnika nam ponujajo že vgrajene standardne spletne strani z vgrajenimi osnovnimi elementi za kontrolo sistema in komunikacije. Ponujajo nam hiter in enostaven prikaz in dostop do najpomembnejših informacij o krmilniku. Integrirane so v strojno-programski opremi centralno procesne enote krmilnika [6].

4.1 Dostop do standardnih spletnih strani

Za dostop do standardnih spletnih strani krmilnika ne potrebujemo dodatnih nastavitev, le omogočiti moramo spletni strežnik na krmilniku. Do njih enostavno dostopamo preko spletnega brskalnika na osebem računalniku, mobilnih napravah, dlančnikih itd.

Za dostop do standardnih spletnih strani iz osebega računalnika moramo slediti naslednjim korakom:

- povezava krmilnika in osebega računalnika v skupno računalniško omrežje ali direktna povezava s standardnim mrežnim kablom,
- konfiguracija krmilnika in aktiviranje spletnega strežnika,
- zagon poljubnega spletnega brskalnika in vpis naslova URL (ang. Unified Resource Locator) v naslovno polje brskalnika (slika 14),
- naslov URL mora biti oblike "http://ww.xx.yy.zz", kjer "ww.xx.yy.zz" predstavlja naslov IP (internetni protokol) krmilnika.



Slika 14: Vpis URL naslova v spletnem brskalniku.

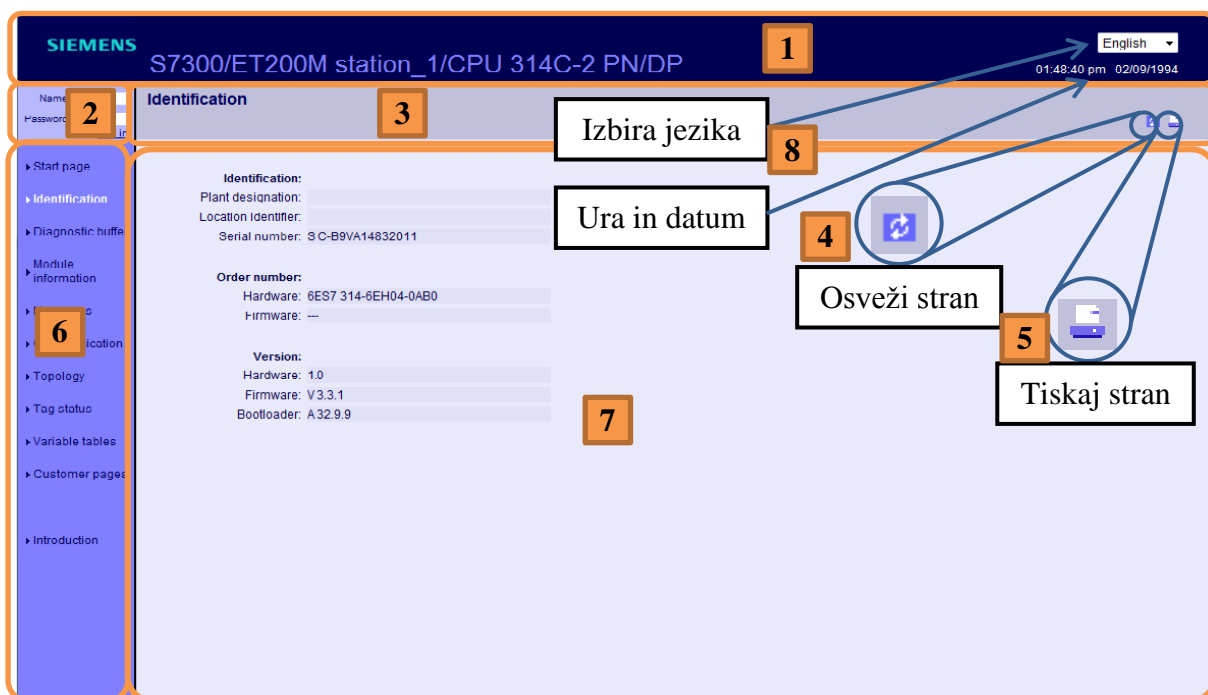
Če dostopamo do spletnega strežnika preko protokola za varen prenos HTTPS, moramo vpisati naslov URL oblike "https://ww.xx.yy.zz", kjer "ww.xx.yy.zz" predstavlja naslov IP krmilnika.

Spletni brskalniki, ki so primerni za dostop do spletnih strani krmilnika:

- Internet Explorer (verzija 6.0 ali novejša),
- Mozilla Firefox (verzija 1.5 ali novejša),
- Opera (verzija 9.0 ali novejša),
- Netscape Navigator (verzija 8.1 ali novejša).

4.2 Postavitev standardnih spletnih strani

Vsaka od standardnih spletnih strani ima enako postavitev z navigacijskimi povezavami in kontrolo strani; to prikazuje slika spodaj (slika 15).



Slika 15: Postavitev standardne spletne stran krmilnika.


Standardne spletne strani so sestavljene iz:

- glave spletnega strežnika, ki prikazuje čas in datum in nam omogoča izbiro različnih jezikov (točka 1),
- odjave in prijave v sistem (točka 2),
- standardne glave strani z imenom strani, ki je prikazana (točka 3),

- ikone "osveži stran": strani z avtomatskim osveževanjem – omogoči ali izključi avtomatsko osveževanje; za strani brez avtomatskega osveževanja – sproži osvežitev strani (točka 4),
- ikone "tiskaj": pripravi in prikaže izpisljivo različico informacij, ki jih ponuja izbrana stran (točka 5),
- navigacijskega polja za preklon med stranmi (točka 6),
- vsebine strani, ki je specifična za vsako standardno stran (točka7),
- spustnega seznamskega polja za izbiro jezika, v katerem bo spletna stran predstavljena (točka 8).

4.3 Prijava v sistem

Za pregled podatkov v standardnih spletnih straneh prijava ni potrebna. Za naprednejše funkcije, kot so spreminjanje delovanja krmilnika, pisanje vrednosti v pomnilnik in nadgrajevanje strojno programske opreme, pa je potrebna prijava z administratorskimi pravicami(slika 16).


 A screenshot of a login form. It features two input fields: the top one is labeled "Name" and the bottom one is labeled "Password". Below the "Password" field is a blue button with the text "Log in" in white. The entire form is enclosed in a light blue border.

Slika 16: Formular za prijavo v sistem.

Za prijavo kot administrator moramo slediti naslednjim korakom:

- vpis "admin" v vnosno polje "Name" (ime),
- vpis gesla v vnosno polje "Password" (geslo), če je le ta konfiguriran – v nasprotnem primeru enostavno pritisnemo tipko za vnos.

4.4 Opis standardnih spletnih strani

V nadaljevanju sledi opis posameznih standardnih spletnih strani, kaj nam omogočajo in njihova uporaba. Vse opisane spletne strani podpirajo Siemensovi programirljivi logični krmilniki serije S7-300.

Standardne spletne strani uporabniku omogočajo različne funkcije, kot so:

- spreminjanje načina delovanja centralno procesne enote,
- spremljanje in spreminjanje vrednosti PLK značk,

- spremljanje diagnostičnega medpomnilnika,
- nadgrajevanje strojno-programске opreme.

Vse, kar ponujajo standardne spletne strani krmilnika, bi lahko združili v naslednje pojme:

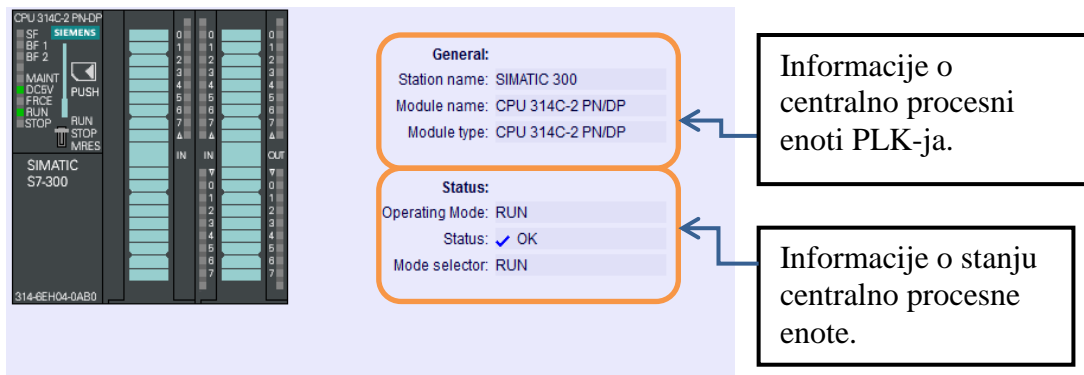
- identifikacija,
- diagnostični medpomnilnik,
- informacije o modulu,
- sporočila,
- komunikacija,
- topologija,
- spremenljivke.

Informacije	Opis
Začetna stran	Prikaz glavnih informacij o centralno procesni enoti in trenutni status krmilnika.
Informacije o PLK-ju	Izpis informacij o krmilniku kot so serijska številka, kataloška številka in številka verzije.
Vsebina diagnostičnega medpomnilnika	Prikaz vsebine diagnostičnega medpomnilnika z najnovejšimi vnosi na prvem mestu.
Status modulov	Prikaz, ali so komponente postaje v pravilnem vrstnem redu in če le-te delujejo pravilno.
Sporočila	Prikaz projektno usmerjenih tekstovnih sporočil z možnostjo filtriranja in razvrščanja.
Komunikacija	Prikaz informacij omrežnih naslovov, fizične lastnosti komunikacijskih vmesnikov in komunikacijska statistika.
Topologija	Grafični prikaz topologije omrežja.
Status spremenljivk	Prikaz statusa operandov v uporabniškem programu za opazovanje vrednosti.
Tabela spremenljivk	Prikaz tabele spremenljivk, ki je bila kreirana s programom STEP 7 s funkcijo spletnega strežnika.
Uporabniške strani	Prikaz povezav do uporabniško definiranih spletnih strani, če so le-te na voljo.

Tabela 2: Tabela standardnih spletnih strani krmilnika z opisom.

4.4.1 Začetna stran

Začetna stran (ang. start page), prikazana na sliki 17, nam prikaže glavne informacije o centralno procesni enoti PLK-ja in v kakšnem delovnem stanju se le-ta nahaja. Če smo v sistem prijavljeni kot administrator, nam spletna stran ponudi možnost spreminjanja načina delovanja CPE (izvrševanje ali prekinitev) in gumb za identifikacijo PKL-ja, ki v primeru aktiviranja povzroči utripanje svetlečih diod na krmilniku.

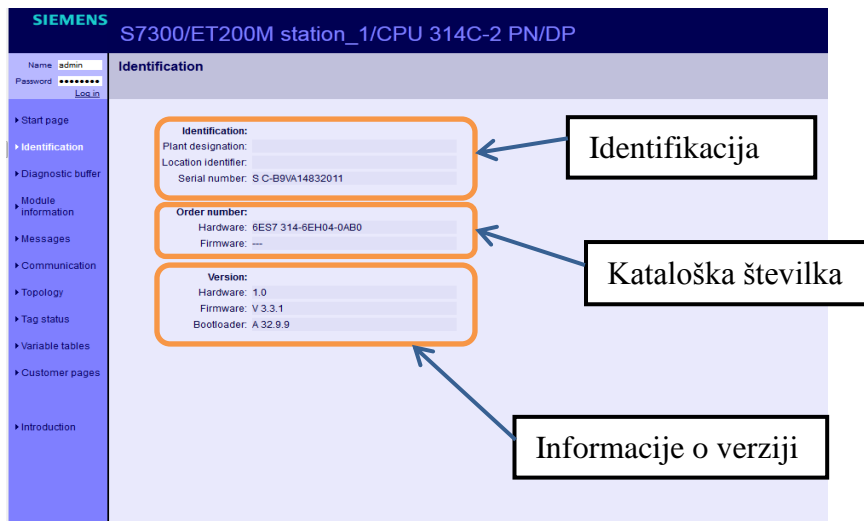


Slika 17: Prikaz informacij krmilnika.

4.4.2 Identifikacija

Identifikacijska spletna stran (ang. Identification) na sliki 18 prikazuje identifikacijske značilnosti centralno procesne enote, kot so :

- serijska številka,
- kataloška številka,
- informacije o verziji.



Slika 18: Identifikacijska spletna stran.

4.4.3 Diagnostični medpomnilnik

Spletna stran z diagnostičnimi podatki (ang. Diagnostic Buffer) na sliki 19 nam prikaže diagnostične dogodke. S spustno-seznamskim poljem lahko izbiramo razpon prikaza diagnostičnih vnosov (točka 3). Zgornji razdelek strani (točka 1) nam prikaže diagnostične vnose s časom in datumom, ko je prišlo do dogodka, in informacije o dogodku. Za podrobne informacije o dogodku izberemo posamezen vnos iz zgornjega razdelka strani, podrobni podatki pa se nam nato izpišejo v spodnjem razdelku strani (točka 2). Stran nam omogoča tudi shranjevanje in izvoz podatkov (točka 4) v formatu CSV (ang. Comma-Separated Values).

The screenshot displays the Siemens diagnostic buffer interface for station S7300/ET200M. The main table lists diagnostic events with the following data:

Number	Time	Date	Event
1	08:17:54:270 pm	02/10/1994	Event: 16# 4302
2	08:17:54:269 pm	02/10/1994	Event: 16# 1381
3	08:17:54:061 pm	02/10/1994	Event: 16# 4301
4	08:17:50:376 pm	02/10/1994	Event: 16# 4303
5	11:10:35:393 pm	02/08/1994	Event: 16# 4302
6	11:10:35:382 pm	02/08/1994	Event: 16# 1382
7	11:10:35:290 pm	02/08/1994	Event: 16# 4301
8	11:10:30:742 pm	02/08/1994	Event: 16# 4300
9	11:09:21:766 pm	02/08/1994	Event: 16# 494E
10	02:12:20:103 am	02/08/1994	Event: 16# 4302

The details view for event 3 shows the following information:

```

Details: 3
No entry in the text database: Hex values will be displayed.
Event ID: 16# 4301
OB: 16# 46
PK: 16# FF
DistID: 16# 72 C7
Additional Info 1 / 2 / 3: 16# 4303 0813 7713
incoming event
  
```

Callouts in the image identify key features: 1 (table of events), 2 (details view), 3 (dropdown menu), and 4 (save/export icon).

Slika 19: Spletna stran prikaza diagnostičnega medpomnilnika.

4.4.4 Sporočila

Spletna stran s sporočili (ang. Messages) na sliki 20 prikaže sporočila krmilnika in njihovo vsebino; sporočila se lahko uporabljajo za opozarjanje. Izpisana so v tabeli v kronološkemu vrstnemu redu in vsebujejo čas in datum generiranja sporočila. Tabela sporočil lahko filtriramo glede na zastavljene pogoje, posamezno sporočilo pa sestavljajo parametri:

- številka sporočila,
- datum,
- čas,
- tekstovna vsebina sporočila,
- status in
- potrditev.

SIEMENS CPU 317/CPU 317-2 PN/DP

English

Admin Log out

Messages

MessageNr. Filter

Message	Date	Time	Message text	Status	Acknowledg.
93	14.04.2008	08:23:24.644	PN device 5 on PN system....	incoming	
78	14.04.2008	08:23:24.796	PN device 4 on PN system....	incoming	
71	14.04.2008	08:23:24.948	PB slave 3, on PB system....	incoming	
70	14.04.2008	08:23:25.099	PB slave 1, on PB system....	incoming	not acknowledged
56	14.04.2008	08:23:25.251	PN device 3 on PN system....	incoming	not acknowledged
92	14.04.2008	08:23:25.402	PN device 2 on PN system....	incoming	not acknowledged
26	14.04.2008	08:23:25.553	PN device 1 on PN system....	incoming	

Details on message number: 93

Short description: SCALANCE-X204IRT Order number: 6GK5 204-0BA00-2BA3

Callouts: Filtriranje sporočil, Tabela sporočil, Podrobnosti sporočila

Slika 20: Prikaz sporočil na spletni strani.

Klik na posamezen parameter na vrhu tabele nam razvrsti tabelo sporočil po izbranem parametru. Podrobnosti sporočila lahko prikažemo v spodnjem razdelku strani z izbiro posameznega sporočila iz tabele.

4.4.5 Podatki o krmilniku

Spletna stran s podatki o krmilniku (ang. Module Information) na sliki 21 prikaže informacije o vseh moduli na stojalu oziroma letvi. Zgornji razdelek strani prikaže povzetek modulov, spodnji razdelek pa prikaže status in identifikacijo izbranega modula.

SIEMENS CPU 317/CPU 317-2 PN/DP

English

Admin Log out

Module information

Name Filter

CPUS317 - Ethernet(1): PROFINET-IO-System (100)

Symbol	Name	Order number	P Address	Comment
IM151-3PN-1	IM151-3PN-1	6ES7 151-3BA23-0AB0	192.168.3.152	Topology
IM151-3PN	IM151-3PN	6ES7 151-3AA20-0AB0	192.168.3.156	Topology
SCALANCE-X204IRT	SCALANCE-X204IRT	6GK5 204-0BA00-2BA3	192.168.3.167	Topology

Status Identification Statistic

Manufacturer: Siemens
 Firmware version: V6.0
 Device class: IM151-3PN
 Plant designation: AKZ (IM151-1)
 Location Identifier: OKZ (IM151-1)
 Installation date: 04.03.2007
 Description: Comment

Callouts: Filtriranje, Status, Identifikacija, Podrobnosti, Topologija, Statistika, Naslov IP

Slika 21: Spletna stran prikaza modulov

4.4.6 Komunikacije

Komunikacijska spletna stran (ang. Communicatoin) nam prikaže parametre priključenih centralno procesnih enot in statistiko komunikacije. Podrobnejše informacije se ponujajo s klikom na zavihke:

- Parameters (Parametri),
- Statistics (Statistika),
- Resources (Viri) in
- Open User Communication (odprta uporabniška komunikacija)

Zavihek "Parametri" (slika 22) nam prikaže naslov MAC, naslov IP in nastavitve IP centralno procesne enote. Prikaže tudi informacije, povezane z integriranim vmesnikom PROFINET.

The screenshot shows a web interface with a navigation bar containing 'Parameter', 'Statistics', 'Resources', and 'Open communication'. The 'Parameter' tab is active. The main content area is divided into three sections, each highlighted with an orange border:

- Network connection:**
 - MAC address: 00-1B-1B-14-A8-E1
 - Name: plcxb3144bf9
- IP parameter:**
 - IP Address: 212.235.182.121
 - Subnet mask: 255.255.255.0
 - Default router: ---
 - IP settings: IP address is set via SDB
- Physical properties:**

Port number	Link status	Settings	Mode
1	OK	automatic	100 MBit/s full-duplex
2	disconnected	automatic	10 MBit/s half-duplex

Three callout boxes on the right side of the screenshot provide additional context:

- Informacije vmesnika PROFINET.** (points to the Network connection section)
- Informacije o parametrih IP** (points to the IP parameter section)
- Informacije o fizičnem nivoju:**
 - številka vrat,
 - status povezave,
 - nastavitve,
 - način delovanja.

Slika 22: Prikaz strani zavihka Parameter.

Zavihek "Statistics" (slika 23) prikazuje statistiko poslanih oziroma prijetih podatkovnih paketov in kvaliteto prenosa podatkov.

The screenshot shows the 'Statistics' tab with the following data:

Data package since: 11:10:30 pm 02/08/1994

Total statistics

Sent data packages:
 Sent without errors: 491573
 Collision during sending attempt: 0
 Canceled due to other errors: 0

Received data packages:
 Received without errors: 256236
 Rejected due to error: 0
 Rejected due to resource bottleneck: 0

Statistics Port 1

Sent data packages:
 Sent without errors: 491573
 Collision during sending attempt: 0
 Canceled due to other errors: 0

Received data packages:
 Received without errors: 256236
 Rejected due to error: 0
 Rejected due to resource bottleneck: 0

Statistics Port 2

Sent data packages:
 Sent without errors: 0
 Collision during sending attempt: 0
 Canceled due to other errors: 0

Received data packages:
 Received without errors: 0
 Rejected due to error: 0
 Rejected due to resource bottleneck: 0

Callouts:

- Čas in datum prenosa prvega podatkovnega paketa
- Kvaliteta prenosa poslanih podatkovnih paketov.
- Kvaliteta prenosa prejetih podatkovnih paketov
- Statistični podatki o kvaliteti prenosa na oddajni strani za določeno številko vrat.
- Statistični podatki o kvaliteti prenosa na sprejemni strani za določeno številko vrat.

Slika 23: Prikaz strani zavihka Statistic.

Informacije o obremenjenosti povezav glede na število priključenih virov se izpišejo pod zavihkom "Resources" (slika 24).

The screenshot shows the 'Resources' tab with the following data:

Number of connections:
 Maximum connections: 12
 Connections not assigned: 12

Connections:	reserved	assigned
PG communication	1	0
OP communication	1	0
S7 Basic communication	0	0
S7 communication	0	0
Other communication	--	0

Callouts:

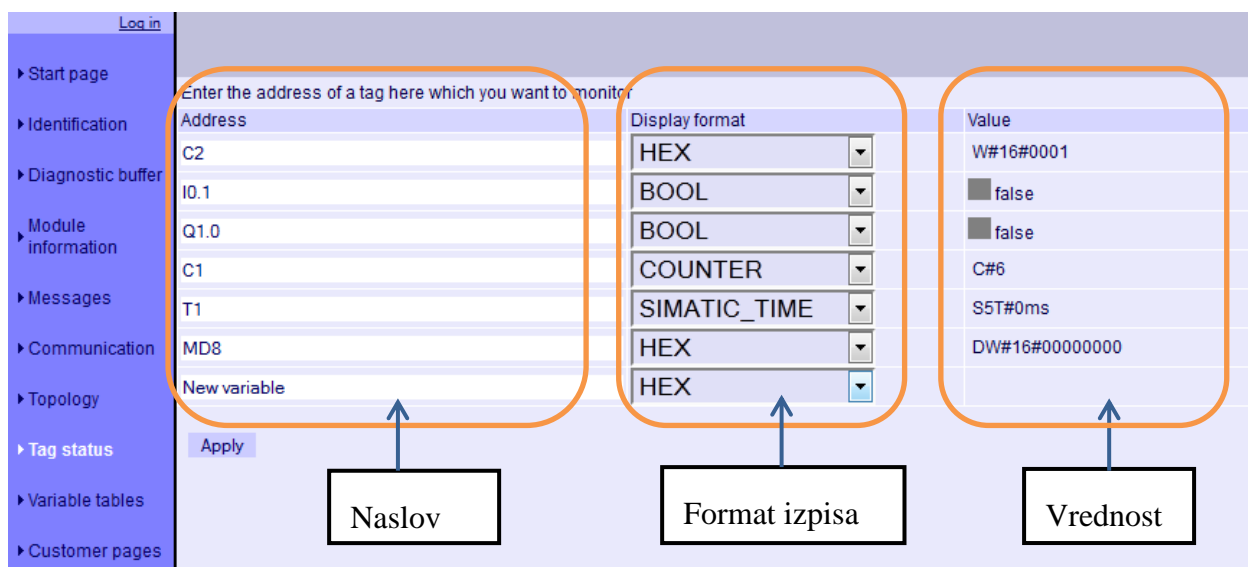
- Prikaz informacij o maksimalnem številu povezav, ki niso trenutno v uporabi.
- Informacije o številu rezerviranih in dodeljenih povezavah.

Slika 24: Prikaz strani zavihka Resources.

4.4.7 Stanje spremenljivk

Spletna stran Stanje spremenljivk (ang. Tag Status) na sliki 25 omogoča vpogled v stanje vhodov, izhodov ali pomnilnika krmilnika. V vnosno polje lahko vpišemo direktni naslov (npr. I0.0), ime oznake spremenljivke ali oznako specifičnega podatkovnega bloka. Za dostop do oznak v podatkovnih blokih napišemo ime podatkovnega bloka v oglatih oklepajih. Vsaki opazovani spremenljivki lahko tudi določimo format prikaza vrednosti; opazujemo lahko poljubno število vrednosti spremenljivk. Opazovane vrednosti se avtomatsko prikažejo in osvežujejo po prevzetih nastavitvah – ta možnost se izključi s klikom na ikono "off" (izklop) za osveževanje strani, ki se nahaja v zgornjem delu spletne strani. Ko je osveževanje onemogočeno, lahko kliknemo ikono "on" (vklop) za ponovno avtomatsko osveževanje.

Prijava v sistem kot administrator nam omogoči spreminjanje vrednosti podatkov. V vnosno polje "Modify Value" (spremeni vrednost) lahko vnesemo poljubno vrednost. Klik na gumb "Go" (pojdi) ob že vpisani vrednosti povzroči vpis vrednosti v krmilnik na prej določen naslov. Prijava kot administrator nam dovoljuje tudi vpis več vrednosti naenkrat, ki jih z pritiskom na gumb "Modify All Values" (spremeni vse vrednosti) vpišemo v pomnilnik krmilnika.

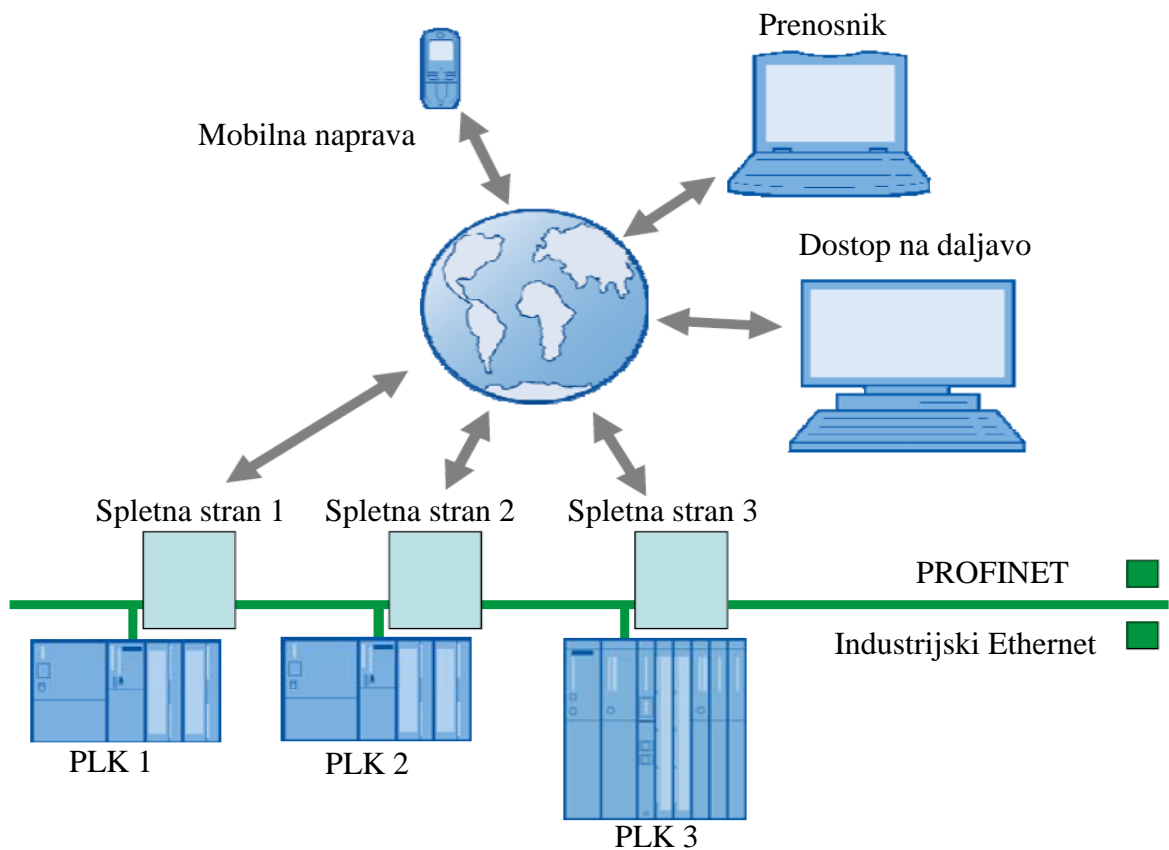


Slika 25: Spletna stran izpise stanje spremenljivk.

Spletna stran ne ohrani naših vnosov v primeru zapustitve spletne strani. Spletno stran lahko shranimo kot zaznamek, v tem primeru pa se vsi naši vnosi ohranijo; če spletne strani ne shranimo, moramo ponovno vpisati imena spremenljivk.

5 Uporabniško definirane spletne strani

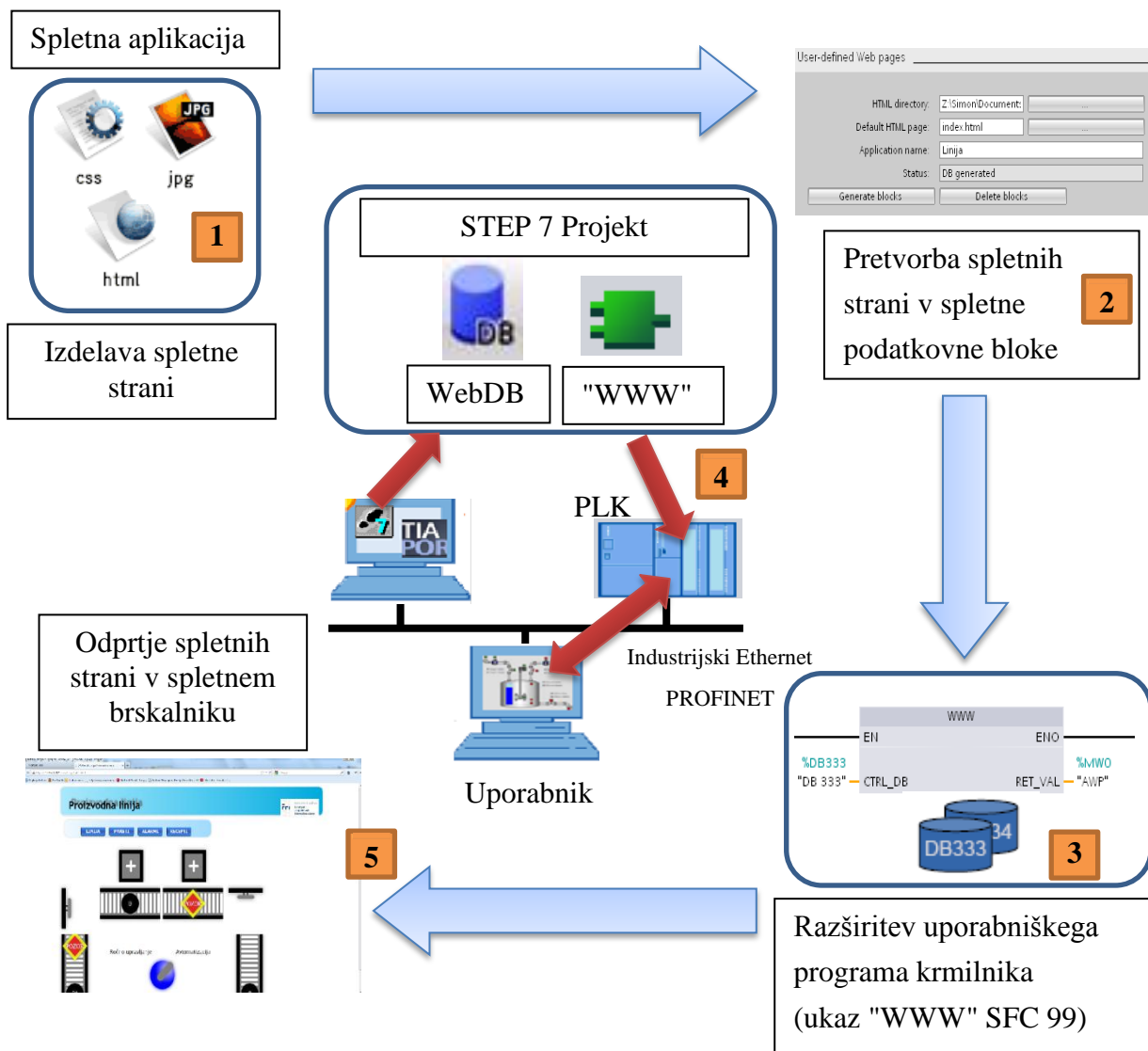
Pojem uporabniško definirana spletna stran (v nadaljevanju: spletna stran) definira poljubno oblikovano spletno stran, ki ji lahko dodamo poljubno funkcionalnost – uporabnik po svojih potrebah in željah oblikuje spletno stran in ji doda določeno funkcionalnost. Spletna stran predstavlja vmesnik za interakcijo človek – stroj [11]. Spletno stran prenesemo in shranimo na spletnem strežniku krmilnika. Uporabnik oziroma operater preko spletnega brskalnika (slika 25) dostopa do spletne strani na spletnem strežniku krmilnika in tako komunicira z napravo. Povezava do uporabniško definiranih spletnih strani se nahaja v standardnih spletnih straneh krmilnika. Spletne strani nam omogočajo realizacijo nadzornega sistema SCADA preko spletnega strežnika krmilnika. Spletna stran pride do svojega izraza, če nimamo nobenega trajnega sistema človek – stroj, vendar so diagnostične informacije in vizualizacija občasno potrebne. Oblikovalec oziroma programer ne potrebuje posebnih programskih orodij za izdelavo spletnih strani – uporabiti morata le svojo intuitivnost in zmogljivost spletnih tehnologij.



Slika 25: Dostop do spletne strani na PLK-ju

5.1 Celoten potek izdelave uporabniško definiranih spletnih strani

Celoten potek izdelave spletnih strani lahko razdelimo na več faz. Slika 26 ponazarja celoten postopek izdelave uporabniško definirane spletne strani [12]. Prva faza (točka 1) predstavlja samo izdelavo spletne strani oziroma spletnega mesta (HTML, CSS, JavaScript itd.). V drugi fazi (točka 2) s programom STEP 7 pretvorimo spletno stran v spletne podatkovne bloke (WebDB). V tretji fazi (točka 3) uporabniškemu programu krmilnika dodamo ukaz "WWW". Četrta faza (točka 4) predstavlja prenos celotnega STEP 7 projekta na krmilnik. Ko se projekt uspešno prenese na krmilnik, lahko s spletnim brskalnikom dostopamo do spletnih strani (točka 5).



Slika 26: Postopek izdelave uporabniško definiranih spletnih strani.

5.2 Konfiguracija v programu STEP 7

Pod lastnostmi centralno procesne enote krmilnika v programu STEP 7 se na levi strani prikaže meni. V meniju izberemo kategorijo "Web server" in izberemo podkategorijo "User-defined Web pages". Odpre se nam nastavitvena stran za uporabniško definirane spletne strani. Nastavitvena stran nam omogoča konfiguracijo in generiranje spletnega podatkovnega bloka (za grafični prikaz glej sliko 27).

Slika 27: Konfiguracija uporabniško definiranih spletnih strani

V nastavitvah s klikom na gumb določimo pot do datotek HTML in drugih vsebin, ki sestavljajo spletno stran (točka 1). Vnosno polje "Default HTML page" (točka 2) določa ime prevzete spletne strani oziroma domačo stran spletnega mesta. Domačo stran lahko izberemo tudi s klikom na gumb in nato določimo pot do datoteke HTML. Domača stran vsebuje povezave do drugih spletnih strani; povezave na spletne strani na spletnem strežniku krmilnika morajo biti v obliki relativnega naslova glede na izhodiščno datoteko. V vnosno polje "Application name" (točka 3) po želji vnesemo ime aplikacije – ime aplikacije predstavlja del naslova URL spletne strani. Če je podano, dobi naslov URL naslednjo obliko: "http://ww.xx.yy.zz/awp/ime aplikacije/ime spletne strani.html". Spletna stran je v krmilniku shranjena v obliki podatkovnih blokov. Klik na gumb "Generate blocks" (točka 4) povzroči generiranje podatkovnih blokov, gumb "Delete blocks" pa

izbriše podatkovne bloke. V vnosno polje "Files with dynamic content" (točka 6) vnesemo končnice datotek, ki vsebujejo dinamično vsebino – dinamične datoteke vsebujejo ukaze AWP. Številko spletnega kontrolnega podatkovnega bloka lahko spremenimo z vnosnim poljem "Web DB number" (točka 7). Začetno številko dela spletnih podatkovnih blokov lahko spremenimo z vnosnim poljem "Fragment DB start number" (točka 8). Trenutni status generiranih spletnih podatkovnih blokov se izpiše v polju "Status" (točka 5).

5.3 Izdelava spletnih strani

Za izdelavo spletnih strani ne potrebujemo dragih orodji, ki nam jih ponujajo proizvajalci procesne opreme. Postopek izdelave nam olajša in pospeši programska oprema za urejanje datotek HTML (Dreamweaver, Aptana, FrontPage), lahko pa enostavno uporabimo beležnico. Za oblikovanje spletne strani lahko uporabimo vse zmogljivosti, ki nam jih ponujajo HTML, CSS in JavaScript, poleg tega so tu še posebni ukazi (Automation Web Programming - AWP) za direktno komunikacijo s centralno procesno enoto krmilnika.

5.3.1 HTML

Jezik za označevanje nadbesedila (Hyper Text Markup Language - HTML) je označevalni jezik za prikaz spletnih strani in ostalih informacij, ki so lahko prikazane v spletnem brskalniku [18]. Zapisan je v obliki HTML elementov, ki so sestavljeni iz značk, ki jih obdajata kotna oklepaja (npr. "<h1>"), znotraj njih pa je shranjena vsebina spletne strani. Značke so običajno pisane v parih (npr. "<div>" "</div>"), med njima pa oblikovalec poljubno vstavlja HTML elemente.

HTML predstavlja osnovno zgradbo spletne strani. Vsebuje vsebino spletne strani, prav tako pa določa strukturo spletne strani in semantični pomen delov dokumenta.

HTML je osnova za izdelavo standardnih spletnih strani. V našem primeru smo uporabili HTML za razporejanj elementov oziroma grafičnih gradnikov, kot tudi za samo vsebino spletne strani. HTML nam omogoča prikazovanje vrednosti točk krmilnika s pomočjo ukazov AWP. Z uporabo HTML form pa lahko uporabnik vpisuje vrednost v točke krmilnika.

5.3.2 CSS

Kaskadne stilske podloge (v nadaljevanju: podloge, ang.: Cascading Style Sheets - CSS) so podloge, ki skrbijo za predstavitev oziroma obliko spletnih strani [19]. Z njimi definiramo stil HTML elementov v smislu pravil, kako naj se ti prikažejo na spletni strani. HTML elementom lahko določamo pozicijo, odmik, barvo, velikost, poravnanoost in druge attribute elementa. CSS podloge nam omogočajo lažje urejanje,

in dodajanje stilov, poskrbijo pa tudi za večjo preglednost dokumentov HTML, saj ločujejo vsebino in strukturo spletne strani od njene predstavitve. Pospešijo tudi izdelavo spletne strani, saj zmanjšajo ponavljanje tako, da omogočajo množici spletnih strani uporabo iste podloge.

Podlogo lahko integriramo v datoteko HTML na več različnih načinov z elementom HTML, med značkama za skriptni jezik ali v zunanji datoteki CSS. Podloge so definirane v ločeni datoteki CSS.

Primer integracije podloge v datoteko HTML:

```
<link type="text/css" rel="stylesheet" href="css/paketi.css">
```

Primer oblikovanja elementa HTML s podlogo:

```
h1.naslov {
    font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
    font-size: 45px;
    font-weight: bold;
    color: #000;
    text-shadow: 12px -12px 5px #666666;
    text-align: left; }
```

Pri izdelavi standardnih spletnih strani podloge niso nujno potrebne, olajšajo pa nam samo izdelovanje spletne strani in izboljšajo njen vizualni del. Pomagajo nam tudi pri razporejanju elementov. Odločitev za uporabo podlog je v rokah oblikovalca spletne strani, pri uporabi pa je treba paziti na podporo spletnega brskalnika; v nasprotnem primeru je spletna stran predstavljena drugače, kot smo si zamislili.

5.3.3 JavaScript

JavaScript je predmetno orientiran programski jezik, ki se uporablja za izračunavanje in rokovanje z objekti na uporabniški strani [22]. JavaScript se uporablja za ustvarjanje dinamičnih spletnih strani. S skriptnim jezikom ne omogoča direktnega branja in pisanja podatkov na spletno stran – podatke izpisujemo in vnašamo preko objektnega modela dokumenta.

Uporablja se na širokem področju:

- upravljanje z okni in okvirji,
- meniji,
- predstavitev besedila in slik,
- delo s piškotki,

- oblikovanje integriranih uporabniških vmesnikov,
- preverjanje vnesenih podatkov,
- pošiljanje podatkov,
- časovni nadzor.

JavaScript lahko vgradimo (med značkama HTML `<script>` `</script>`) ali pa vključimo v datoteko HTML. Datoteke skriptnega jezika JavaScript imajo končnico (.js).

Primer integracije skriptnega jezika JavaScript v datoteko HTML

```
<script src="javaScript/alarmi.js"></script>
```

Pri izdelavi nadzornega sistema v spletnih straneh smo uporabili vse zmogljivosti skriptnega jezika JavaScript.

5.4 Spletni kontrolni podatkovni blok

Da centralno procesna enota krmilnika razume format, v katerem so shranjene spletne strani, se te pretvorijo in shranijo v obliki podatkovnih blokov [11]. Spletne strani pretvorimo s programom STEP 7. Pri generiranju podatkovnih blokov se ustvari spletni kontrolni podatkovni blok (prevzeto: DB333) in drugi podatkovni bloki, ki vsebujejo spletno stran (prevzeto: DB334).

Kontrolni spletni podatkovni blok vsebuje:

- status in kontrolo spremenljivk spletne strani,
- status komunikacije (npr. ko spletni brskalnik pošlje zahtevo spletnemu strežniku v teku)
- informacije o napakah,
- informacije o strukturi spletne strani.

Velikost spletnih strani določa velikost programa krmilnika. Velikost programa krmilnika, podatki in konfiguracija so omejeni na velikost pomnilnika (MMC kartica) in glavnega pomnilnika v krmilniku, pri tem pa moramo paziti, da spletne strani ne vsebujejo velikih podatkov (slike itd.).

5.5 Uporaba ukaza WWW (SFC 99)

Ukaz WWW (slika 28) omogočajo dostop do uporabniško definiranih spletnih strani. Sinhronizirajo uporabniški program krmilnika z uporabniško definirano spletno stranjo [11]. Uporabniški program krmilnika mora vsaj enkrat izvršiti ukaz WWW, da omogoči dostop do uporabniško definiranih spletnih strani. Tako lahko omogočimo dostop do uporabniško definiranih spletnih strani pod posebnimi pogoji. Uporabniški program krmilnika lahko kliče



Slika 28: Ukaz WWW v lestvičnem diagramu.

ukaz WWW glede na zahteve aplikacije; za interakcijo med programom krmilnika in uporabniško definirano spletno stranjo se morajo ukaz WWW uporabiti v cikličnem delu programa.

Spletne strani so shranjene v krmilniku v obliki podatkovnih blokov. Centralno procesna enota ne ve, da so v podatkovnih blokih shranjene spletne strani, zato ukaz WWW obvesti centralno procesno enoto, kateri podatkovni blok je kontrolni spletni podatkovni blok. Vhodni parameter ukaza WWW je spletni kontrolni podatkovni blok. Izhodni parameter nam sporoča napake v numerični obliki.

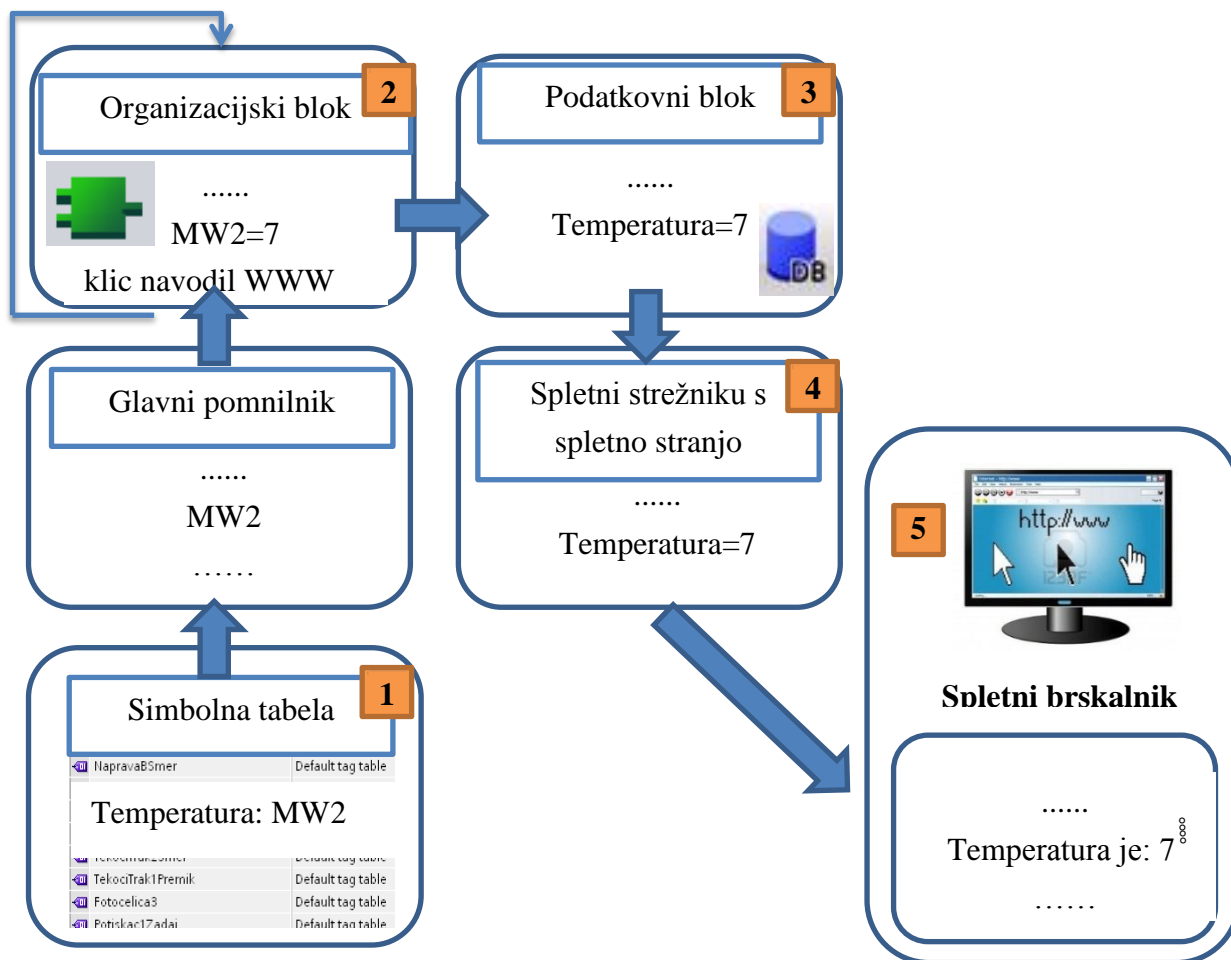
5.6 Interakcija med spletnim brskalnikom in krmilnikom

5.6.1 Prikaz spremenljivk na spletni strani

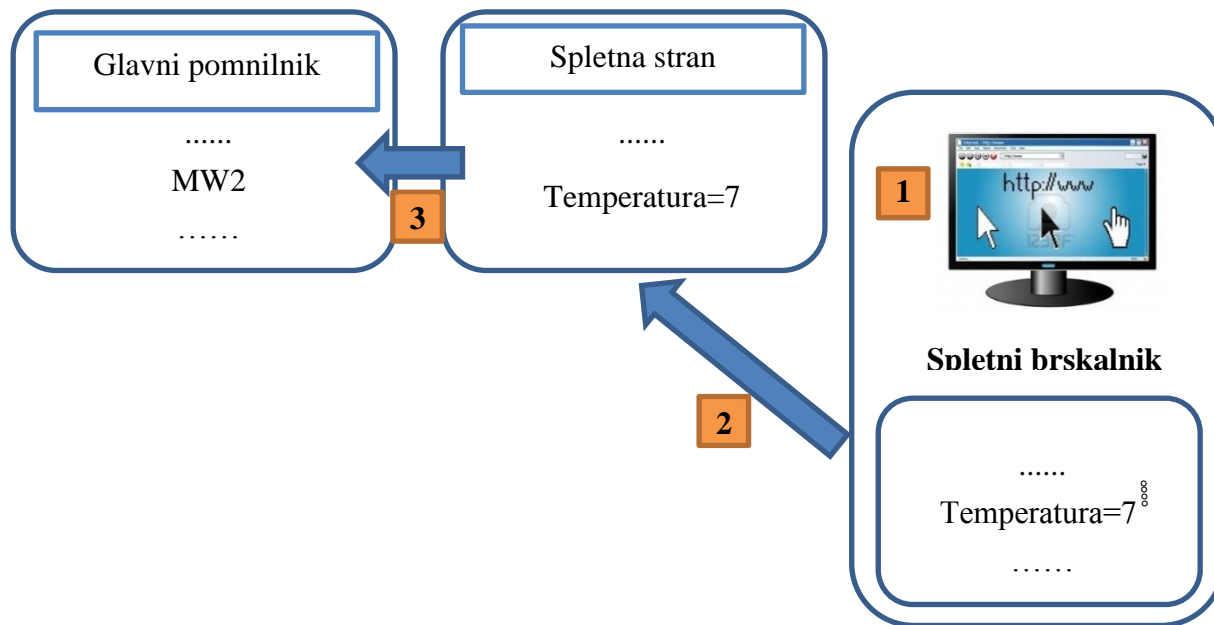
Slika 29 prikazuje interakcijo med centralno procesno enoto krmilnika in spletnim brskalnikom pri branju podatkov [11]. Spremenljivke, ki se izpisujejo na spletni strani, morajo imeti simbolično ime (točka 1). Spremenljivka je lahko izpisana na spletni strani samo s simboličnim imenom, vsi podatkovni tipi za spremenljivke so dovoljeni. V programu krmilnika se izvrši klic ukaza WWW (točka 2). Klic ukaza WWW inicializira spletni kontrolni podatkovni blok (točka 3), spletni strežnik krmilnika pa pretvori podatke s pomočjo informacij v le-tem (točka 4). Podatki se pretvorijo v spletnem brskalniku razumljiv format. Z vsako zahtevo spletnega brskalnika se spletna stran osveži (točka 5).

5.6.2 Pisanje spremenljivk iz spletne strani v krmilnik

Slika 30 prikazuje interakcijo med spletnim brskalnikom in centralno procesno enoto krmilnika pri pisanju podatkov [11]. Uporabnik preko spletnega vmesnika spremeni vrednost spremenljivke (točka 1). Spletni brskalnik pošlje zahtevo z POST metodo (točka 2), program krmilnika pa sprejme spremembo in spletna stran se osveži (točka 3). Metoda POST je ene izmed metod protokola HTTP za pošiljanje zahtev. Uporablja se, ko uporabnik želi poslati podatke spletnemu strežniku, kot del zahteve. Zahteva je lahko v obliki datoteke ali spletne forme.



Slika 29: Interakcija med krmilnikom in spletnim brskalnikom pri branju spremenljivk.



Slika 30: Interakcija med krmilnikom in spletnim brskalnikom pri branju spremenljivk.

5.7 Osveževanje podatkov na spletni strani

Osveževanje podatkov je ključnega pomena pri nadzornem sistemu. Operater sistema mora imeti pred seboj vedno sveže informacije, ki mu jih sporoča proces. Hitrost manipuliranja z procesnimi podatki je odvisna od procesa in od specifičnega parametra, ki ga želimo prikazati oziroma spremeniti. Čas med spremembo poljubnega parametra v procesu in prikazom tega na zaslonu mora biti čim bolj realen.

Čas osveževanja spletne strani je odvisen od velikosti vsebine spletne strani. Statični in dinamični (spremenljivke v krmilniku) del spletne strani morata biti osvežena.

Notranji prenosni čas med centralno procesno enoto in spletnim strežnikom krmilnika sta odvisna od števila spremenljivk, ki se prenašajo, velikost spremenljivk pa je praktično nepomembna. Hitrost prenosa lahko povečamo s komunikacijsko obremenitvijo na račun programskega ciklusa.

Tabela 3 nam prikazuje čas prenosa spremenljivk glede na število spremenljivk in nastavitve komunikacijske obremenitve.

Število spremenljivk	Komunikacijska obremenitev [%]	Čas osveževanja [s]
10	20	2.4
10	40	2.1
20	20	3.3
20	40	2.8
40	20	5.9
40	40	4.8

Tabela 3: Čas prenosa spremenljivk v krmilniku.

Hitrost prenosa spremenljivk lahko še povečamo, če vse spremenljivke, ki jih spletna stran zahteva, shranimo v isti podatkovni blok. Varna povezava (HTTPS) tudi zmanjša hitrost prenosa podatkov.

V principu je spletna stran (HTML) statična in se ne odziva na spremembe vsebine. Če se vrednost spremenljivke v krmilniku spremeni, se sprememba ne odrazi na spletni strani. Za ažurne podatke je spletno stran potrebno periodično osveževati.

Obstaja več načinov osveževanja spletne strani:

- ročno osveževanje,
- samodejno osveževanje z atributom "http-equiv" v meta podatkih datoteke HTML,
- samodejno osveževanje z JavaScript v telesu datoteke HTML,
- samodejno osveževanje z AJAX (Asynchronous JavaScript and XML).

5.7.1 Ročno osveževanje

S pritiskom na tipko "F5" na tipkovnici ali izbiro funkcije osvežitev v spletnem brskalniku ročno osvežimo spletno stran. Ta način pri izdelavi nadzornega sistema ne pride v poštev.

5.7.2 Osveževanje z HTML

HTML značka "meta" predstavlja meta podatke dokumenta HTML. Atribut "http-equiv" omogoča dostop in spreminjanje glave datoteke HTML. Z nastavitvijo vrednosti "refresh" atributu "http-equiv" omogočimo avtomatsko osveževanje strani. Atributu "content" podamo žaljen časovni interval osveževanja v sekundah.

Spodnji primer osveži spletno stran vsakih deset sekund:

```
<meta http-equiv="refresh" content="10" >
```

5.7.3 Osveževanje z JavaScript

S skriptnim jezikom JavaScript lahko periodično osvežujemo celotno spletno stran. Objekt "Window" predstavlja odprto okno v spletnem brskalniku; objekt vsebuje metodo "setTimeout", za klicanje funkcije v določenih intervalih. Funkciji podamo dva parametra, kličejo funkcijo in časovni interval v milisekundah. Objekt "Location" vsebuje informacije o trenutnem naslovu URL in vsebuje metodo "reload", ki osveži trenutno stran. Metodi "setTimeout" dodamo za parameter metodo "reload", s tem pa avtomatsko osvežujemo spletno stran.

Spodnji primer povzroči osvežitev spletne strani v pet sekundnem intervalu:

```
<script language="JavaScript">
window.setTimeout("location.reload()",5000);
</script>
```

5.7.4 Osveževanje z uporabo tehnologije Ajax

Tehnologija Ajax (Asynchronous JavaScript and XML) je skupina medsebojno povezanih spletnih tehnik, uporabljenih na uporabniški strani [21]. Uporablja se za ustvarjanje interaktivnih

spletnih aplikacij. S tehnologijo Ajax si spletne aplikacije izmenjujejo podatke asinhrono v ozadju, brez potrebe po ponovnem nalaganju celotne spletne strani.

Tehnologija Ajax nam omogoča tekoče in hitrejše spremljanje ter spreminjanje vsebine na spletni strani. Podatki se prenašajo s pomočjo objekta "XMLHttpRequest", Navkljub imenu Javascript in XML tehnologiji nista pogoj za izvajanje tehnologije Ajax. Tehnologija Ajax ni samo ena tehnologija, temveč se nanaša na naslednje tehnologije:

- HTML in CSS za oblikovanje,
- objektni model dokumenta (Document Object Model - DOM) za dinamično prikazovanje vsebine in interakcijo s podatki,
- XML in XSLT za delo s podatki,
- objekt XMLHttpRequest za asinhrono komunikacijo,
- JavaScript, ki povezuje vse te tehnologije (JavaScript služi tudi kot vmesnik med posameznimi komponentami).

Tehnologija Ajax nam omogoča osveževanje samo tistih spremenljivk, ki jih spletna stran zahteva, in ne celotne spletne strani. V primeru opisanim v nadaljevanju spletno stran osvežujemo s pomočjo tehnologije Ajax in ukazov AWP.

Ustvarimo datoteko XML (Extensible Markup Language), v katero vpišemo ukaze AWP za branje. Ukazi morajo biti napisani med značkama, katerim podamo primerno ime, na katerega se nato sklicujemo v skriptnem jeziku JavaScript. Datoteko shranimo na spletni strežnik PLK.

Primer vsebine datoteke XML s ukazi AWP za branje:

```
<alarmi>
  <alarm1>:="Alarm1":</alarm1>
  <alarm2>:="Alarm2":</alarm2>
  <alarm3>:="Alarm3":</alarm3>
  <alarm4>:="Alarm4":</alarm4>
  <alarm5>:="Alarm5":</alarm5>
</alarmi>
```

V skriptnem jeziku JavaScript pošljemo zahtevo spletnemu strežniku za prenos datoteke XML. Ustvarimo instanco objekta "XMLHttpRequest", ki vsebuje metodi "open" in "send" za pošiljanje zahteve spletnemu strežniku. Lastnosti objekta "onreadystatechange" shrani funkcijo, ki se kliče ob spremembi statusa zahteve. Z lastnostma objekta "readyState" in "readyState" preverimo, če je zahteva izpolnjena in če se je datoteka XML prenesla.

Spodnji primer prikazuje pošiljanje zahteve spletnemu strežniku za prenos datoteke XML:

```
<script>
    function naložiXML(){ //funkcija za pošiljanje zahteve spletnemu strežniku
        var xmlhttp, docXml;
        if (window.XMLHttpRequest){ //koda za IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp=new XMLHttpRequest(); }
        else { //koda za IE6, IE5
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); }
        xmlhttp.onreadystatechange=function(){
            if (xmlhttp.readyState==4 && xmlhttp.status==200){
                docXml =xmlhttp. responseXML; }
            }
        xmlhttp.open("GET","212.235.182.121./awp/Linija/xml",true);
        xmlhttp.send();
    }
</script>
```

Prenesene podatke lahko shranimo tekstovni obliki ali v formatu XML. Za shranjevanje podatkov v formatu XML uporabimo lastnost "responseXML" objekta "XMLHttpRequest".

Primer shranjevanja podatkov v formatu XML:

```
var docXml =xmlhttp. responseXML;
```

S skriptnim jezikom JavaScript in objektnim modelom dokumenta beremo podatke iz datoteke XML in jih dinamično prikazujemo na spletni strani.

Primer branja podatka iz datoteke XML in prikaz le tega na zaslonu:

```
var alarm=document.getElementById("alarmDiv");
alarm.innerHTML= docXml.getElementsByTagName("alarm1")[0].childNodes[0].nodeValue
```

5.8 Alternativne rešitve za uporabniško definirane spletne strani

Glede na zahtevo aplikacije Siemens ponuja široko paleto rešitev za dostop do podatkov v centralno procesni enoti krmilnikov serij SIMATIC S7 preko spletnih tehnologij.

5.8.1 Java Applets

Z uporabo IE CPs in Java Applets lahko enostavno ustvarimo preprosto vizualizacijo za uporabniški vmesnik. Procesne spremenljivke so integrirane preko S7 kontrol. Java uporabniški vmesnik je prikazan na standardnem spletnem brskalniku.

5.8.2 Java Beans

S7 Java Beans omogočajo ustvarjanje individualnih vtičnikov Java Applet za srednje in velike spletne aplikacije.

5.8.3 Sm@rt Service z WinCC flexible

Z programom WinCC flexible in Sm@rt Service opcijo se lahko direktno povežemo na nadzorno ploščo preko interneta.

5.8.4 SIMATIC WinCC Web Navigator

SIMATIC WinCC Web Navigator omogoča funkcionalnost upravljanja in spremljanja procesa preko intraneta ali lokalnega omrežja podjetja brez potrebe po spreminjanju projekta WinCC [7]. To pomeni, da Web Navigator ponuja enake možnosti prikaza, vnosa podatkov in opcije dostopa kot lokalna upravljalna postaja. Prikazni zaslone procesa lahko vsebujejo Visual Basic ali C skripte za dinamični prikaz.

Za spletno rešitev je Web Navigator nameščen na enouporabniškem sistemu WinCC ali strežniku. Odjemalec Web Navigator je nameščen na osebem računalniku z operacijskim sistemom Windows – tako lahko upravljamo in nadzorujemo tekoči projekt WinCC preko spletnega brskalnika MS Internet Explorer ali terminalskih storitev.

6 Ukazi AWP (Automation Web Programming)

Ukazi AWP (Automation Web Programming) omogočajo branje in pisanje podatkov v spremenljivke Siemensovih programirljivih logičnih krmilnikov preko uporabniško definiranih spletnih strani in predstavljajo ključno vez med našo spletno stranjo in spremenljivkami v krmilniku [13]. Uporabljamo jih za direktno komunikacijo z centralno procesno enoto krmilnika, običajno v povezavi z osnovnimi značkami HTML za kreiranje forme. Ukazi AWP so lahko v datoteki HTML postavljeni na poljubno mesto, a jih zaradi preglednosti kode pišemo na začetku datoteke HTML. Ukazi so namenjeni spletnemu strežniku in se na njem tudi izvedejo. S prihodom novejšje serije Siemensovih krmilnikov se je nabor ukazov AWP še povečal, tako ti omogočajo:

- branje spremenljivk,
- pisanje spremenljivk,
- branje preko posebnih spremenljivk,
- pisanje preko posebnih spremenljivk,
- definiranje neštevnih tipov,
- prirejanje vrednosti neštevnih tipov,
- kreiranje dela podatkovnega bloka in
- uvoz dela podatkovnega bloka.

Spremenljivka v centralno procesni enoti krmilnika se imenuje tudi točka PLK; ta predstavlja le eno vhodno ali izhodno vrednost v sistemu.

6.1 Sintaksa ukazov AWP

V datoteki HTML pišemo ukaze AWP med značkama `<!-- ... -->`, ki v jeziku HTML predstavljata komentar. Z izjemo ukaza za branje spremenljivk sintaksa ukazov AWP zglada takole: `<-- AWP_<ime ukaza in parametri> -->`. V skriptnih jezikih je priporočljivo pisanje ukazov AWP kot komentar; v datotekah JavaScript pišemo ukaze med znakoma za komentar `/*...*/`.

V nadaljevanju sledi opis sintakse posameznih ukazov AWP, za katero veljajo naslednji dogovori:

- postavke v oglatih oklepajih " [] " so neobvezni,
- postavke v kotnih oklepajih " < > " so obvezne,

- narekovaji so dobesedno del ukaza in morajo biti prisotni kot je navedeno,
- posebni znaki imenu točke PLK ali imena podatkovnih blokov v odvisnosti od uporabe morajo biti navedeni med dvojnima narekovajema.

6.2 Kratak pregled ukazov AWP

V nadaljevanju sledi podrobnejša predstavitev ukazov AWP in njihova uporaba v datotekah HTML. Zaradi preglednosti, pomoči razvijalcu uporabniško definiranih spletnih in hitrega vpogleda sledi povzetek ukazov AWP.

- Branje spremenljivke

`:=<ime spremenljivke>:`

- Pisanje spremenljivke

`<!-- AWP_In_Variable Name = '<ime spremenljivke 1>' [Use = '<ime spremenljivke 2>'] -->`

- Branje posebnih spremenljivke

`<!--AWP_Out_Variable Name= '<Tip> :<Ime>' [Use = '<ime spremenljivke>'] -->`

- Pisanje posebnih spremenljivk e

`<!-- AWP_In_Variable Name='<Tip>:<Name>' [Use='< ime spremenljivke >'] -->`

- Definiranje neštevnega tipa

`<!-- AWP_Enum_DefName='<Ime neštevnega tipa>' Values='<Vrednost>, <Vrednost>,...' -->`

- Navajanje neštevnega tipa

`<!-- AWP_Enum_RefName='<Ime spremenljivke>' Enum='<Neštevni tip>' -->`

- Kreiranje dela podatkovnega bloka

`<!-- AWP_Start_Fragment Name='<<Ime>' [Type=<Tip>][ID=<id>] -->`

- Uvažanje dela podatkovnega bloka

`<!-- AWP_Import_Fragment Name='<Ime>' -->`

6.3 Pravila za pisanje imen točk PLK v ukazih AWP

Ukaza za branje in pisanje spremenljivk vsebujeta atributa ime in opsijsko atribut "Use". Ime točke PLK je dodeljeno tem atributom.

Naslednja pravila veljajo za uporabo imen točk PLK v datoteki HTML:

- točka PLK mora biti obdana z dvojnimi narekovaji,
- točka PLK, uporabljena v ukazih AWP, mora biti prav tako obdana z narekovaji ali z levo poševnico pred obema narekovajema ("\" ... \""),
- če ime točke PLK vsebuje znak "\" (leva poševnica), mora biti ta znak podvojen,
- če je uporabljen absolutni naslov (vhod, izhod, pomnilnik) v ukazu AWP je obdan z enojnima narekovajema.
- imena podatkovnih blokov so obdana z dvojnima narekovajema, točka PLK pa ne.

6.4 Podrobnosti in uporaba ukazov AWP

V nadaljevanju sledi podrobnejši opis ukazov AWP s primeri uporabe v datotekah HTML.

6.4.1 Branje spremenljivk (točk PLK)

Preko uporabniško definiranih spletnih strani lahko beremo spremenljivke (točke PLK) iz centralno procesne enote krmilnika. Točka PLK mora biti določena z imenom. Izhodne spremenljivke (če gledamo iz strani krmilnika) so lahko vstavljene na poljubno mesto v datoteki HTML. Sklic na ime spremenljivke spremeni spletni strežnik z vrednostjo želene spremenljivke.

Kot parameter ukaza podamo ime točke PLK, lahko pa vpišemo naslov spremenljivke v pomnilniku, vhodno/izhodni naslov ali ime podatkovnega bloka. Pri navajanju imena spremenljivke moramo paziti na vsebovanost posebnih znakov.

Sintaksa ukaza AWP za branje spremenljivk:

`:=<ime spremenljivke>:`

Tabela 4 prikazuje različne primere imen točk PLK in ukaza AWP za branje.

Ime podatkovnega bloka	Ime točke PLK	Pomnilniški naslov	Ukaz AWP za branje
	Hitrost_motorja		:= "Hitrost_motorja":
Stevci	Motor		:= "Stevci".Motor:
		I0.0	:= I0.0:
		MW100	:= MW100:

Tabela 4: Primeri ukaza AWP za branje točk PLK.

6.4.2 Pisanje spremenljivk (točk PLK)

Preko uporabniško definirane spletne strani lahko pišemo podatke v pomnilnik centralno procesne enote krmilnika. To dosežemo z uporabo ukazov AWP, ki identificirajo spremenljivko v centralno procesni enoti, v katero želimo vpisati vrednost preko spletne strani. Spremenljivka mora biti določena kot ime točke PLK ali kot ime podatkovnega bloka. V enem ukazu lahko deklariramo več imen spremenljivk, katerih vrednosti bodo vpisane v pomnilnik krmilnika.

Točke PLK so s strani brskalnika določene bodisi v glavi HTTP (metoda piškotka ali POST) ali pa v naslovu URL (GET metoda) in jih nato spletni strežnik krmilnika vpiše v ustrezno točko PLK. Običajno se za pisanje podatkov uporablja HTTP POST ukaz.

Ukaze za pisanje se tipično uporablja v kombinaciji s spletnim obrazcem. Oblikovati je potrebno spletni obrazec z vnosnim poljem ali izbirnim seznamom. Ko uporabnik vpiše ali izbere vrednost, spletni strežnik pretvori vneseno vrednost v primerno obliko in jo zapiše v pomnilnik krmilnika.

Pravico do pisanja v krmilnik imajo samo registrirani in prijavljeni uporabniki – ukazi so spregledani, če uporabnik ni prijavljen v sistem z administratorskimi pravicami.

Sintaksa ukaza za pisanje v spremenljivke:

```
<!-- AWP_In_Variable Name = '<ime spremenljivke 1>' [Use = '<ime spremenljivke 2>'] -->
```

Kot parameter lahko ukazu podamo dve imeni točk PLK. V primeru, da podamo obe spremenljivki, se bo prva uporabljala v datoteki HTML, druga pa za zapis spremenljivke v kontrolnik. Tako dobi "originalna" spremenljivka drugo ime, ki se uporablja v datoteki HTML.

Primer uporabe z vnosnim poljem:

```
<!-- AWP_In_Variable Name=""Hitrost"" Use=""Podatki".Hitrost' -->
<form method="post">
<p>Braking: <input name=""Hitrost"" type="text" /> %</p>
</form>
```

Primer uporabe z zbirnim seznamom:

```
<!-- AWP_In_Variable Name=""Podatki".luc'-->
<form method="post">
<select name="" Podatki ". luc">
<option value=":" Podatki ". luc":> </option>
<option value=1>JA</option>
<option value=0>NE</option>
</select><input type="submit" value="Submit setting" /></form>
```

6.4.3 Branje preko posebnih spremenljivk

Spletni strežnik krmilnika ponuja možnost branja vrednosti iz centralno procesne enote preko posebnih spremenljivk, ki so shranjene v odzivni glavi protokola HTTP. Ukazi AWP za branje in pisanje posebnih spremenljivk se razlikujejo po tem, da imajo dodatni parameter.

Spletni strežnik krmilnika prebere točno PLK in jo prenese oziroma shrani v odzivno glavo protokola HTTP. Kot primer: če želimo prebrati pot do točke PLK in jo prenesti na drugo spletno stran oziroma naslov URL, uporabimo ukaz "HEADER:naslov kjer želimo točko PLK".

Sintaksa ukaza za branje posebnih spremenljivk:

```
<!--AWP_Out_Variable Name= '<Tip> :<Ime> ' [Use = '<ime spremenljivke> ' ] -->
```

Za parameter Tip lahko podamo naslednje vrednosti:

- HEADER (glava),
- COOKIE_VALUE (vrednost piškota),
- COOKIE_EXPIRES (prenehanje veljavnosti piškotka).

Parameter Ime v ukazu predstavlja imena spremenljivk v glavi protokola HTTP, ki so vnaprej določena. Ime spremenljivke predstavlja točko PLK, katere vrednost želimo prebrati. Uporaba besede " Use " v ukazu je poljubna in omogoča, da lahko po želji poimenujemo ime točke PLK.

V spodnji tabeli so prikazane spremenljivke v odzivni glavi protokola HTTP katere lahko beremo z ukazi AWP.

Tip	Ime	Opis ukaza
COOKIE_VALUE	poljubno ime	vrednost piškotka z določenim imenom
COOKIE_EXPIRES	poljubno ime	izvrševalni čas piškotka z določenim imenom v sekundah
HEADER	Status	statusna koda HTTP
HEADER	Location	pot za preusmeritev na drugo spletno stran
HEADER	Retry-After	predviden čas v katerem storitev ne bo dosegljiva
HEADER	...	vse spremenljivke glave protokola HTTP so lahko tako uporabljene

Tabela 5: Spremenljivke v odzivni glavi protokola HTTP.

Primer uporabe:

V spodnjem primeru posebna spremenljivka "HTTP:Status" dobi vrednost točke PKL z imenom "HTTP:Status". Ime točke PKL v tabeli mora biti identično spremenljivkam v glavi protokola HTTP.

```
<!-- AWP_Out_Variable Name="HEADER:Status" -->
```

V tem zgledu posebna spremenljivka "HEADER:Status" dobi vrednost točke PKL z imenom "Status".

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use="'Status'" -->
```

6.4.4 Pisanje preko posebnih spremenljivk

Spletni strežnik krmilnika ponuja možnost pisanja vrednosti v pomnilnik krmilnika preko posebnih spremenljivk, ki so shranjene v zahtevni glavi protokola HTTP. V principu so vse spremenljivke HTTP vpisane v glavo HTTP protokola .

Sintaksa ukaza za pisanje posebnih spremenljivk:

```
<!-- AWP_In_Variable Name='<Tip>:<Ime>' [Use='< ime spremenljivke >'] -->
```

Za parameter Tip lahko podamo naslednje vrednosti:

- HEADER (glava),
- COOKIE_VALUE (vrednost piškota),
- SERVER (strežnik).

Parameter Ime v ukazu predstavlja imena spremenljivk v glavi protokola HTTP, ki so vnaprej določena. Ime spremenljivke predstavlja točko PLK, katere vrednost želimo prebrati. Uporaba besede " Use " v ukazu je poljubna in omogoča, da lahko po želji poimenujemo ime točke PLK. V spodnji tabeli so prikazane spremenljivke v zahtevni glavi protokola HTTP v katere lahko pišemo z ukazi AWP.

Tip	Ime	Opis ukaza
HEADER	Accept-Language	<i>Sprejet ali prednostni jezik</i>
HEADER	Authorization	<i>Potrdilo o avtorizaciji za zahtevan vir</i>
HEADER	Accept:	<i>vrste vsebine, ki so sprejemljive</i>
HEADER	Host	<i>Gostitel in številka vrat zahtevanega vira</i>
HEADER	User-Agent	<i>Informacije o uporabniškemu posredniku (brskalniku)</i>
HEADER	<i>vse spremenljivke glave protokola HTTP so lahko tako uporabljene</i>
SERVER	current_user_id	<i>Informacije o uporabniku vpisanem v sistem</i>
SERVER	current_user_name	<i>Ime trenutnega uporabnika vpisanega v sisem</i>
SERVER	GET	<i>zahtevna metoda : GET</i>
SERVER	POST	<i>Zahtevna metoda: POST</i>
COOKIE_VALUE:	poljubno ime	<i>Vrednost piškotka s poljubnim imenom</i>

Tabela 6: Spremenljivke v zahtevni glavi protokola HTTP.

Primer uporabe:

V spodnjem primeru spletna stran vpiše vrednost iz posebne spremenljivke, shranjene v glavi protokola HTTP, v točko PLK.

```
<!-- AWP_In_Variable Name=SERVER:current_user_id' Use="moj_uporabnik"-->
```

Zgled prikazuje vpis piškotka v točko PLK "info".jezik.

```
<!-- AWP_In_Variable Name="COOKIE_VALUE:siemens" Use="info".jezik' -->
```

6.4.5 Definiranje neštevnih tipov

Numerične vrednosti, uporabljene v programu krmilnika, je mogoče pretvoriti v besedilo (in obratno) z uporabo neštevnih tipov. Namesto numeričnih vrednosti lahko uporabimo besedilo v različnih jezikih.

Sintaksa ukaza za definiranje neštevnih tipov:

```
<!-- AWP_Enum_Def Name='<Ime neštevnega tipa>' Values='<Vrednost>, <Vrednost>,... ' -->
```

Kot parameter podamo ukazu ime neštevnega tipa in vrednosti. Parameter Vrednost je sestavljen iz numerične vrednosti in besedila, ki predstavlja to vrednost. Sintaksa za parameter Vrednost je sledeča: “ <numerična vrednost>:<besedilo> ”.

Primer uporabe: primer prikazuje, kako lahko namesto numerične vrednosti za vrednost stikala uporabimo besedilo.

```
<!-- AWP_Enum_Def Name="StikaloEnum" Values='0:" Izklop", 1:"Vklop", 2:"Napaka"' -->
```

6.4.6 Prirejanje vrednosti neštevnih tipov

Spremenljivkam lahko prirejamo vrednosti neštevnega tipa; ta spremenljivka je lahko uporabljena kjerkoli v uporabniško definirani spletni strani. Nad njimi lahko izvajamo operacije branja in pisanja. Med operacijo branja spletni strežnik nadomesti numerično vrednost z ustreznim besedilom, pri operaciji pisanja pa spletni strežnik nadomesti besedilo z ustrežno numerično vrednostjo, preden jo zapiše v pomnilnik krmilnika.

Sintaksa ukaza za prirejanje vrednosti neštevnim tipom:

```
<!-- AWP_Enum_Ref Name='<Ime spremenljivke>' Enum='<Neštevni tip>' -->
```

Kot parameter podamo ukazu ime spremenljivke, ki predstavlja točko PLK. Drug parameter predstavlja ime neštevnega tipa, ki smo ga predhodno deklarirali.

Primer uporabe: primer prikazuje uporabo neštevnih tipov. Za zgled: če je vrednost točke PLK "Alarm" enaka numerični vrednosti 2, spletni strežnik izpiše besedilo "Cisterna je prazna".

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Brez alarma", 1:"Cisterna je polna",
2:"Cisterna je prazna"' -->
<!-- AWP_Enum_Ref Name="'Alarm'" Enum="AlarmEnum" -->
<p>Trenutno stanje alarma cisterne je := "Alarm":</p>
```

6.4.7 Kreiranje dela podatkovnega bloka

Del podatkovnega bloka lahko opišemo kot logične odseke uporabniško definirane spletne strani, katere centralno procesna enota krmilnika procesira individualno. Deli podatkovnega bloka običajno vsebujejo celotne spletne strani, lahko pa vsebujejo le posamezne elemente spletne strani (slike itd.).

Del podatkovnega bloka ustvarimo z ukazom AWP za kreiranje dela podatkovnega bloka – vse, kar sledi ukazu je del podatkovnega bloka, dokler ne dosežemo konca datoteke ali naslednjega ukaza za kreiranje dela podatkovnega bloka.

Sintaksa ukaza za kreiranje dela podatkovnega bloka:

```
<!-- AWP_Start_Fragment Name='<<Ime>' [Type=<Tip>][ID=<id>] -->
```

Parameter Ime predstavlja ime podatkovnega bloka. ID parameter integrira identično številko dela podatkovnega bloka; če je ne navedemo, jo navede spletni strežnik krmilnika.

Za parameter Tip lahko podamo naslednje vrednosti:

- manual
- automatic

Če ne navedemo tipa, spletni strežnik krmilnika po privzetem nastavi parameter Tip na vrednost "automatic".

Primer uporabe: ukaz, ki kreira del podatkovnega bloka za prikaz slike:

```
<!-- AWP_Start_Fragment Name='Logo_FRI' -->
<p></p>
```

6.4.8 Uvažanje dela podatkovnega bloka

V datoteki HTML ustvarimo del podatkovnega bloka, to pa lahko nato prikažemo oziroma uvozimo v poljubno uporabniško definirano spletno stran. Kot primer lahko izpostavimo problem skupka spletnih strani, ki uporabljajo enako sliko za logotip podjetja. To lahko implementiramo z definiranjem novega dela podatkovnega bloka, ki bo vseboval sliko logotipa podjetja – vsaka individualna spletna stran nato lahko uvozi del podatkovnega bloka, ki vsebuje logotip podjetja. Ukaz za uvažanje dela podatkovnega bloka je namenjen prav temu.

```
<!-- AWP_Import_Fragment Name='<Ime>' -->
```

Parameter ukaza je ime dela podatkovnega bloka, ki ga želimo uvoziti.

Primer uporabe: ukaz, ki v spletno stran uvozi del podatkovnega bloka.

```
<!-- AWP_Import_Fragment Name=' Logo_FRI ' -->
```

7 Nadzorni sistem proizvodne linije

V okviru diplomskega dela smo izdelali nadzorni sistem proizvodne linije preko uporabniško definiranih spletnih strani. Tako smo dokazali, da je realizacija nadzornega sistema preko uporabniških spletnih strani sploh mogoča. Nadzorni sistem vsebuje vse zahteve proizvodnega procesa in nadzornega sistema SCADA.

Spletne strani so oblikovane s kaskadnimi stilskimi podlogami. Za dinamično funkcionalnost uporabljamo skriptni jezik JavaScript, ki poskrbi za izris, branje podatkov, pisanje podatkov in preverjanje pravilnosti vnesenih podatkov. Spletne strani se osvežujejo s pomočjo spletne tehnologije Ajax. V krmilnik vpisujemo podatke s pomočjo obrazcev HTML, pošiljamo pa jih z metodo POST. Za izris proizvodne linije uporabimo grafične gradnike iz knjižnice WebSCADA.

Nadzorni sistem sestavlja več spletnih strani, ki so razdeljene glede na namen in vsebino.

7.1 Branje spremenljivk

Vse spremenljivke beremo s pomočjo tehnologije Ajax, z izjemo spremenljivke za stanje procesa in spremenljivk, ki se izpisujejo v vnosnih poljih spletnih obrazcev. Prebrane vrednosti spremenljivk izpisujemo v tabeli ali pa jih predstavimo s pomočjo slik in grafičnih gradnikov knjižnice WebSCADA. Za lažje implementiranje branja uporabimo standardne funkcije iz knjižnice WebSCADA. Za izpis vrednosti spremenljivk na zaslon uporabljamo skriptni jezik JavaScript v povezavi z objektnim modelom dokumenta.

Za prikaz branja spremenljivke iz krmilnika in njen izpis na zaslon v nadaljevanju opisujemo primer branja in izpisa identifikacijske številke paketa.

V datoteki HTML ustvarimo tabelo in posameznim elementom HTML dodamo identifikacijsko ime, na katero se pozneje sklicujemo.

Primer elementa HTML:

```
<td id="trakB"></td>
```

V skriptnem jeziku JavaScript z metodo "getElementById", ki ji kot atribut dodamo identifikacijsko ime elementa HTML, dostopamo do posameznega elementa v datoteki HTML.

Primer dostopa do elementa v datoteki HTML:

```
var ptrakB=document.getElementById("trakB");
```

Z metodo "setInterval" objekta Window kličemo funkcijo za osveževanje. Funkcija sprejme parametra "millisec" in "code". Parameter " millisec " predstavlja periodični čas klicanja podane funkcije v milisekundah. Parameter "code" pa predstavlja kličočo funkcijo.

Primer uporabe funkcije "setInterval":

```
window.setInterval(osvezuj,100); //interval osvezevanja 100 ms
```

Funkcija za osveževanje kliče funkcijo "NaloziXML", ki je del knjižnice WebSCADA. Funkcija prebere datoteko XML, v našem primeru iz spletnega strežnika krmilnika. Datoteke XML na spletnem strežniku vsebujejo ukaze AWP za branje, ki se ob zahtevi spremenijo v vrednosti spremenljivk.

Funkciji dodamo kot parameter:

- pot do datoteke, v katerem je shranjena datoteka XML,
- ime datoteke s končnico,
- kličočo funkcijo, ki se kliče, ko je datoteka XML prenesena.

Funkcija poskrbi, da ni potrebno vnašati tip protokola (http ali https), po katerem se bo datoteka prenesla, saj ga samodejno zazna.

Primer klica funkcije "NaloziXML", ki prenese datoteko "paketi.xml":

```
var path='/awp/Linija/xml';  
SCADA.NaloziXML(path,"paketi.xml",osvezivevIDPAketa);
```

Ko je datoteka XML, prenesena se kliče funkcija za izpis vrednosti spremenljivke. Spremenljivko preberemo iz datoteke XML in jo izpišemo na zaslon, pred tem pa jo pretvorimo v ustrezen format.

Primer branja spremenljivke iz datoteke XML in njene pretvorbe:

```
ptrakBspr=(parseInt(xmlDoc.getElementsByTagName("ptrakB")[0].childNodes[0].nodeValue))
```

Primer izpisa oziroma dodajanja vrednosti elementu HTML:

```
ptrakB.innerHTML=(ptrakBspr>0)? ptrakBspr:"";
```

7.2 Vpisovanje spremenljivk

Vse spremenljivke se prenašajo z metodo POST. Zahtevo za prenos lahko aktiviramo z elementom HTML za kreiranje forme, kateremu nastavimo vrednost tipa na "submit". Lahko pa uporabimo funkcijo "submit", ki jo kličemo v skriptnem jeziku JavaScript.

Vrednostne spremenljivke vpisujemo v vnosna polja spletnih obrazcev, pri tem pa je treba paziti na pravilnost vnesenih podatkov. S pomočjo skriptnega jezika JavaScript vnosnim poljem dodamo dinamično funkcionalnost. Vnosnim poljem lahko podamo poljubne zahteve po omejitvi vpisa. Tako nas spletni brskalnik opozori, če vneseni podatki ne ustrezajo danim zahtevam. S pomočjo dogodkov in skriptnega jezika lahko onemogočimo vnos določenih znakov.

Primer funkcije, ki se kliče ob pritisku tipke in omogoča vpisovanje samo numeričnih vrednosti:

```
SCADA.preveriVrednost=function (event){
    var kodaZnaka = (event.which) ? event.which : event.keyCode;
    if((kodaZnaka==46||kodaZnaka==8||kodaZnaka==45||kodaZnaka==47) ||(kodaZnaka >=
48 && kodaZnaka<= 57) ){
        return true;
    }
    else {
        return false;
    }
};
```

Logične spremenljivke spreminjamo s pomočjo gumbov. Pri izdelavi nadzornega sistema smo izbrali dva pristopa zaradi potreb nadzornega sistema. Funkcijo stop in zagon pošiljamo preko spletnega obrazca, kjer uporabimo lastnost nevidnosti elementa HTML, funkcijo za resetiranje pa pošiljamo preko skriptnega jezika JavaScript. Pri funkciji za resetiranje pošljemo spletnemu strežniku dve vrednosti – najprej logično vrednost "1" in nato "0", kot zahteva program na krmilniku. Pri tem smo uporabili dogodka, ki se prožita ob pritisku na gumb in ob spuščanju gumba.

7.3 Prikazovanje slik

V nadzornem sistemu smo uporabili slike samo za prikazovanje stanja procesa. Slika se dinamično spreminja glede na logično vrednost, ki predstavlja stanje procesa. Značba HTML za prikaz slik ima naslednjo obliko:

```

```

Atribut "src" predstavlja naslov URL slike. Implementirana prednost je, da naslov URL slike vsebuje ukaz AWP za branje. Spletni strežnik nadomesti ukaz AWP z ustrežno vrednostjo, tako je naslov slike odvisen od spremenljivke v krmilniku.

Ustvarili smo dve sliki z drugačnima zadnjima znakoma v imenu slike ("0" in "1"). Tako se slika dinamično prikazuje glede na logično vrednost spremenljivke.

7.4 Pregled nadzornega sistema

V nadaljevanju sledi pregled nadzornega sistema na spletnem strežniku programirljivega logičnega kontrolnika in kratek opis izdelave posamezne funkcije sistema.

7.4.1 Navigacija

Preko spletnega menija (slika 31) v zgornjem delu strani se enostavno sprehajamo po spletnem mestu. Meni omogoča navigacijo med spletnimi strani za:

- nadzor in pregled proizvodne linije,
- pregled nad paketi,
- alarmiranje,
- recepte,
- kontrolo proizvodne linije.



Slika 31: Meni nadzornega sistema.

7.4.2 Linija

Spletna stran Linija nam omogoča pregled nad celotnim procesom. Vizualno nam prikaže celotno proizvodno linijo. S pomočjo animacije operaterju prikaže stanje in dogajanje posameznega procesa proizvodne linije. Sestavljajo jo posamezni grafični gradniki iz knjižnice WebSCADA.

Proizvodna linija lahko deluje v ročnem ali avtomatskem načinu, kar nam lepo prikaže gumb (slika 32) na sredini spletne strani.

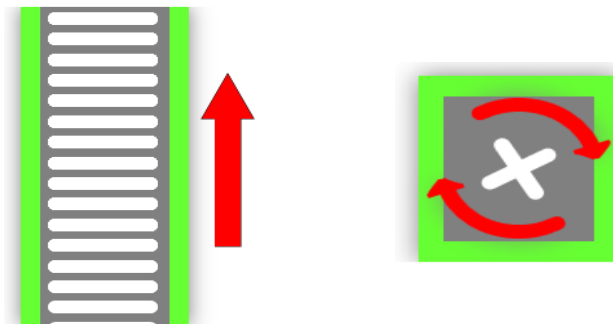
Ročno upravljanje

Avtomatizacija



Slika 32: Gumb za prikaz načina delovanja.

Ko se nahajamo v ročnem načinu, nam nadzorni sistem prikaže izbrano napravo (slika 33) z zeleno obrobo, z animacijo in puščicami pa nam prikaže smer gibanja posamezne naprave. Z animacijo prav tako prikaže stanje, v katerem se naprava nahaja.



Slika 33: Animacija na posameznih napravah.

Za lažje sledenje paketom po proizvodnji liniji se ti izrišejo na določenih lokacijah. Na paketu se izriše njegova identifikacijska številka. Slika 34 prikazuje identifikacijo in izris paketa na potiskaču.



Slika 34: Izris in identifikacija paketa na potiskaču.

Na levi strani zaslona se nahajajo kontrole za nadzor proizvodne linije. Kontrolo sestavljajo gumbi Stop, Zagon in Resetiranje. Gumb Stop zaustavi vse procese na proizvodnji linij, gumb Zagon pa jih ponovno zažene. Pri večji napaki v proizvodnem procesu je potrebno celotno proizvodno linijo ponastaviti, kar storimo z gumbom Resetiraj. Slika 35 prikazuje gumbe za



Slika 35: Gumbi za kontrolo linije.

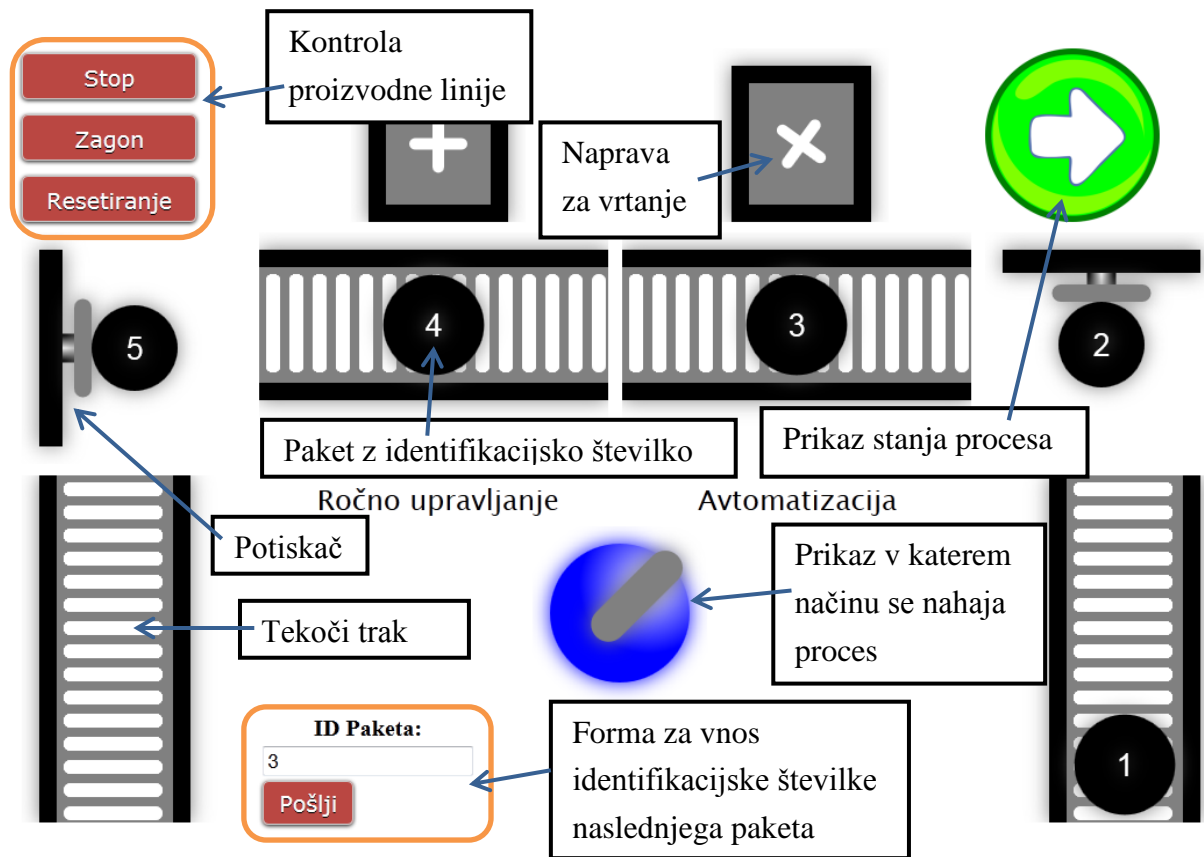
kontrolo linije.

V primeru alarma ali napake se nam na napravi izriše opozorilo (slika 36), proizvodna linija pa se ustavi.



Slika 36: Primer alarma na tekočem traku.

S sliko na desni strani spletne strani prikažemo, ali je proces v načinu izvajalna ali mirovanja. V vnosno polje ID Paketa lahko vnesemo identifikacijsko številko naslednjega paketa. Slika 37 prikazuje celotno vizualizacijo proizvodne linije.



Slika 37: Vizualizacija proizvodne linije.

7.4.3 Paketi

Spletna stran Paketi na pregleden način prikazuje, na katerem mestu se paket nahaja in njegovo identifikacijsko številko. Pozicije so prikazane v tabeli (slika 38), v kateri prvi stolpec z besedo opisuje mesto na proizvodnji liniji, drugi stolpec pa prikazuje identifikacijsko številko paketa.

POZICIJA	ID PAKETA
VHOD:	
TEKOCI TRAK-VHOD:	
POTISKAC A:	13
TEKOCI TRAK- NAPRAVA A:	12
TEKOCI TRAK-NAPRAVA B:	11
POTISKAC B:	10
IZHOD:	9

Slika 38:Tabela prikaza paketov

7.4.4 Alarmi

Spletna stran Alarmi je namenjena prikazovanju alarmov na proizvodnji liniji. Alarmi se izpišejo v tabeli (slika 39), sestavljeno iz stolpcev:

- številka alarma
- čas (čas aktiviranja alarma),
- datum (datum aktiviranja alarma),
- sporočilo (opis pozicije in vzroka alarma).

ŠT. ALARMA	ČAS	DATUM	SPOROČILO
1	19:45:31	16.09.2012	Trak A(Vhod): fotocelica nepričakovano prekinjena
2	19:46:00	16.09.2012	Prehitro nalaganje paketov
3	19:45:27	16.09.2012	Naprava A: fotocelica nepričakovano prekinjena
4	19:45:27	16.09.2012	Naprava B: fotocelica nepričakovano prekinjena
5	19:45:40	16.09.2012	Trak B(Izhod) fotocelica nepričakovano prekinjena

Slika 39: Tabela alarmov

Informacije o proženju alarmov se prenašajo s pomočjo tehnologije Ajax. Spletni strežnik vsakih sto milisekund pošlje zahtevo po datoteki XML, ki vsebuje informacije o proženju alarmov. Datoteka vsebuje logično vrednost, ki nam pove ali je prišlo do proženja alarma ter čas in datum proženja. Alarmi so označeni z identifikacijskimi številkami. Sporočila alarmov so shranjena v datoteki XML, ki je shranjena na spletnem strežniku krmilnika. Datoteka se prenese samo ob nalaganju spletne strani in se shrani v spletnem brskalniku.

Datum in čas alarma spreminja program krmilnika. Ko pride do proženja alarma, program krmilnika v spremenljivko shrani novo vrednost časa in datuma alarma. Čas in datum sta shranjena v formatu "DT" (DATE_AND_TIME). Za razumljiv izpis je format potrebno razčleniti. To storimo s skriptnim jezikom JavaScript in funkcijami za manipuliranje z znakovnimi nizi.

Ko pride do proženja alarma, s pomočjo skriptnega jezika JavaScript izpišemo v tabelo vse informacije o alarmu. Sporočilo alarma s pomočjo identifikacijske številke poiščemo v datoteki XML.

7.4.5 Recepti

Spletna stran Recepti (slika 40) nam omogoča izbiro že pripravljenega recepta ali poljuben vnos vrednosti. S pritiskom na gumb Pošlji krmilniku pošljemo recept. V našem primeru recept sestavljata časovni vrednosti vrtanja naprave. Že pripravljene recepte lahko izberemo preko spustnega seznama ali v vnosna polja vnesemo želeno vrednost. V vnosna polja lahko vnašamo le celoštevilске numerične vrednosti.

Recepti	
Izberi recept:	Plastenke ▾
Čas vrtanja naprava A[ms]:	<input type="text" value="300"/>
Čas vrtanja naprava B[ms]:	<input type="text" value="600"/>
Pošlji:	<input type="button" value="Pošlji"/>

Slika 40: Forma za vnos recepta.

Že pripravljene recepti so shranjeni v datoteki XML na spletnem strežniku krmilnika. Datoteka se prenese ob nalaganju strani. Vsebuje informacijo o imenu recepta in času vrtanja naprave "A" in "B" proizvodnje linije.


Primer recepta shranjenega v datoteki XML:

```
<recepti>
  <recept ime="Pločevinke">
    <napravaA>200</napravaA>
    <napravaB>200</napravaB>
  </recept>
</recepti>
```

Z označbo HTML "select" ustvarimo izpustni izbirni seznam. V seznam s skriptnim jezikom JavaScript vstavimo imena receptov iz datoteke XML. Ko uporabnik izbere določen recept se v vnosna polja prenesejo vrednosti, ki jih recept določa. Recepte ni mogoče ustvariti v nadzornem sistemu ampak so že vnaprej določeni. Spreminjamo, dodajamo in brišemo jih lahko samo ko načrtujemo oziroma izdelujemo nadzorni sistem.

7.4.6 Kontrola

Spletna stran Kontrola (slika 41) združuje elemente za kontrolo na eni strani. Status nam prikazuje stanje procesa, dodan pa je tudi gumb za ponastavitev alarmov. Ta spletna stran je bila dodana nadzornemu sistemu zaradi stabilnosti, saj se spletna stran, ki osvežuje v kratkem času veliko podatkov v povezavi z vnosnimi elementi, občasno obnaša nestabilno.

Kontrolna plošča	
Status:	
Ustavi linijo:	<input type="button" value="Stop"/>
Zaženi linijo:	<input type="button" value="Zagon"/>
Resetiraj linijo:	<input type="button" value="Resetiranje linije"/>
Resetiraj alarme:	<input type="button" value="Resetiranje alarmov"/>
Vnos ID paketa:	<input type="text" value="42"/> <input type="button" value="Pošlji"/>

Slika 41: Kontrolna plošča linije.

8 Knjižnica WebSCADA

Knjižnica WebSCADA je nastala v okviru diplomskega dela. V njej smo implementirali različne grafične gradnike, ki jih lahko vgradimo na spletno stran. Grafični gradniki na vizualen način prikazujejo željeno vrednost, kar kaže tudi na njihovo prednost – prijaznost uporabniku. Operater si lažje predstavlja velikost poljubnega parametra in lahko hitreje reagira na nepričakovan dogodek. Knjižnica vsebuje tudi druge funkcije, ki nam olajšajo izdelavo nadzornega sistema preko uporabniško definiranih spletnih strani.

Pri izdelavi knjižnice sta nas spletna tehnologija in sama izvedba vodili v več različnih smeri, saj je izdelava nadzornega sistema preko uporabniško definiranih spletnih strani zelo specifičen proces. Čeprav se sam nadzorni proces nahaja na spletnem strežniku in bi bil zlahka dosegljiv širši množici, namena splošne dosegljivosti nima. Nadzorni proces je v prvi vrsti namenjen operaterju, ki nadzoruje in krmili proizvodni proces, in šele nato drugim zaposlenim v podjetju.

Ker nadzorni sistem ni namenjen širši množici, smo izbrali specifično tehnologijo, ki jo podpirajo samo najnovejši spletni brskalniki. Tako bi morali v podjetju ali tovarni, če bi hoteli uporabiti našo knjižnico za izdelavo nadzornega procesa, posodobiti spletne brskalnike na vseh računalnikih, ki dostopajo do nadzornega sistema.

Pred izdelavo smo naredili pregled najnovejših spletnih tehnologij. Preučili smo možnosti, ki nam jih ponujajo za implementacijo grafičnih gradnikov in enostavnost njihove vgraditve v spletno stran. Možnosti spletnih tehnologij, med katerim smo izbirali, in razlogi za izbiro določene tehnologije, so opisani v nadaljevanju.

8.1 Izbira programskega jezika

Pri izdelavi gradnikov smo se odločali med dvema programskima jezikoma – Java in JavaScript. Gre za dva popolnoma različna koncepta programiranja in izvajanja. Java je objektno usmerjen programski jezik, medtem ko je JavaScript prav tako objektno usmerjen, a je skriptni jezik.

Vtičniki Java "Applet" (oznaka za majhno javansko aplikacijo) se izvajajo samostojno neodvisno od spletnega brskalnika, lahko pa se izvajajo tudi v spletnem brskalniku z uporabo javanskega navideznega stroja. Java je večji in bolj kompleksen programski jezik; aplikacije morajo biti prevedene v tako imenovani Java "bytecode" (vmesni ukazni jezik), preden se te izvajajo v spletnem brskalniku. Javanski vmesni jezik se izvaja v navideznem stroju. V datotekah HTML aplikacijo pokličemo, druge vloge pa dokument HTML pri javanskih programih nima.

JavaScript je skriptni jezik, ki ga pišemo v datotekah HTML. Izvaja se v spletnem brskalniku in je tako povsem odvisen od spletnega brskalnika. Ta skriptni jezika je majhen, preprost in programerju hitro razumljiv. Omogoča dobro interakcijo z objektnim modelom dokumenta DOM (ang. Document Object Model), preko katerega dostopamo in manipuliramo z elementi HTML. Ponuja večjo svobodo pri kreiranju objektov in hitrejše implementiranje dogodkov na spletni strani.

Za JavaScript smo se odločili zaradi njegove preprostosti, prilagodljivosti in dobre interakcije z objektnim modelom HTML. S podporo HTML5 nam ponuja vse za izgradnjo grafičnih objektov.

8.2 Spletna tehnologija za izris grafičnih objektov

Nove spletne tehnologije nam ponujajo bogat nabor ukazov za izris grafike v spletnem brskalniku. Za realizacijo naših grafičnih objektov smo izbirali med dvema spletnima tehnologijama: HTML5 Canvas in SVG (Scalable Vector Graphics). Obe tehnologiji nam omogočata izris bogate grafike, a sta hkrati tudi popolnoma različni.

8.2.1 SVG (Scalable Vector Graphics)

SVG (Scalable Vector Graphics) je vektorski grafični format, osnovan na XML [24]. Vsebina SVG je lahko statična, dinamična, interaktivna ali animirana. SVG grafiki lahko dodajamo stil (preko kaskadnih stilskih podlog) in dinamično obnašanje (z uporabo SVG DOM API).

Primer izrisa kroga (slika 42):

```
<!DOCTYPE html>
<html>
<body>
  <svg>
    <circle cx="100" cy="50" r="45" stroke="black" stroke-width="2" fill="blue" />
  </svg>
</body>
</html>
```



Slika 42: Krog izrisan s tehnologijo SVG

SWG nam omogoča več kot le enostavno vektorsko grafiko in animacijo – razvijemo lahko tudi interaktivno spletno stran z animiranimi dogodki, filtri, torej skoraj vse, kar si zamislimo.

8.2.2 HTML5 Canvas

Canvas (slo. platno) je element HTML5 dokumenta, ki omogoča izris grafike v spletnem brskalniku [23]. Grafiko izrisujemo sproti z skriptnim jezikom JavaScript preko vgrajenega programskega vmesnika API (ang. Application Programming Interface). Tako s programom, napisanim v skriptnem jeziku, izrisujemo grafiko. Omogoča nam izris tako 2D kot tudi 3D grafike.

Primer izrisa kvadrata (slika 43):

```
<!DOCTYPE html>
<html>
<body>
  <canvas id="kvadrat" width="300" height="150" style="border:1px solid #d3d3d3;">
    Brskalnik ne podpira HTML5 Canvas.</canvas>
  <script>
    var c=document.getElementById("kvadrat");
    var ctx=c.getContext("2d");
    ctx.fillStyle="#FF0000";
    ctx.fillRect(20,20,150,100);
  </script>
</body>
</html>
```



Slika 43: Izris kvadrata z spletno tehnologijo HTML5 Canvas.

8.2.3 Primerjava spletnih tehnologij SVG in HTML5 Canvas

V spodnji tabeli so prikazane razlike med spletnima tehnologijama SVG in HTML5 Canvas [4].

HTML5 Canvas	SVG
Odvisen od resolucije.	Neodvisen od resolucije.
Ni podpore upravljanja dogodkov.	Podpora upravljanja dogodkov.
Bolj primerna za grafiko, ki se hitro osvežuje (igre).	Bolj primerna za izris grafike na večjih površinah (zemljevidi).
Sliko lahko shranimo v formatu .png ali .jpg.	Počasen izris pri zahtevnejši grafiki.

Tabela 7: Primerjava spletnih tehnologij Canvas in SVG.

Za izris naših grafičnih gradnikov smo izbrali tehnologijo HTML5 Canvas, saj nam ta omogoča bolj fleksibilno risanje grafike, odpira pa nam tudi več možnosti za izris in animacijo.

8.3 Sestava knjižnice WebScada

Knjižnico sestavljajo različni grafični gradniki, specifični za vizualizacijo linije in splošno namenski kot na primer termometer, števec, pol števec in stolpčni diagram. Knjižnico sestavljajo tudi funkcije za prenos podatkov s tehnologijo Ajax.

Grafični gradniki knjižnice:

- gumb,
- indikator,
- naprava,
- števec,
- polovični števec,
- tekoči trak,
- termometer,
- potiskač.

8.4 Grafični gradniki

Grafični gradniki nam olajšajo in pospešijo izdelavo vizualnega dela nadzornega sistema, izboljšajo pa tudi preglednost posamezne veličine, saj jo grafično predstavijo. Vsakemu gradniku lahko po želji dodajamo ali spreminjamo prevzete lastnosti – tako grafični gradnik prilagodimo zahtevam našega nadzornega sistema. Omogočajo enostavno vgradnjo v našo spletno stran.

8.4.1 Vgradnja grafičnega gradnika v spletno stran

Vgradnja grafičnega gradnika v spletno stran je zelo enostavna. V datoteki HTML ustvarimo nov element Canvas, ki mu pripišemo atributa "height" (višina) in "width" (dolžina). Atributoma dodamo željeno vrednost dolžine in višine grafičnega gradnika. Dodamo mu standardni atribut za identifikacijo "id", kjer po želji vpišemo ime gradnika, na katerega se bomo potem sklicevali v skriptnem jeziku. Po želji lahko med značkama "canvas" napišemo tudi opozorilo, če spletni brskalnik ne podpira Canvas elementa.

Primer vgradnje grafičnega gradnika v datoteki HTML:

```
<canvas id="termometer" width="150" height="400">[Ni podpore brskalnika]</canvas>
```

V skriptnem jeziku ustvarimo nov objekt zelenega grafičnega gradnika. V konstruktorju kot prvi parameter podamo identifikacijsko ime elementa Canvas; drugi parametri v konstruktorju so odvisni od tipa gradnika, ki ga želimo vgraditi. Parametri konstruktorja običajnega grafičnega gradnika, če jih izpišemo po vrsti, so:

- identifikacija,
- minimalna vrednost,
- maksimalna vrednost,
- vrednost.

Minimalna in maksimalna vrednost predstavljata razpon prikaza vrednosti, parameter vrednost pa predstavlja trenutno vrednost.

Primer ustvarjanja novega gradnika v skriptnem jeziku JavaScript:

```
var termometer=new SCADA.Termometer("termometer",-20,80,0);
```

8.4.2 Lastnosti in izris grafičnega gradnika

Grafičnim gradnikom lahko po želji dodajamo ali spreminjamo določene lastnosti, nabor lastnosti pa je odvisen od tipa grafičnega gradnika. Lastnosti nastavljam preko funkcije "Set", kjer kot prvi parameter podamo ime lastnosti, drugi parameter pa predstavlja vrednost želene lastnosti.

Primer nastavljanja lastnosti grafičnega gradnika:

```
termometer.Set('naslov','Temperatura kotla');
termometer.Set('enote.po','c');
```

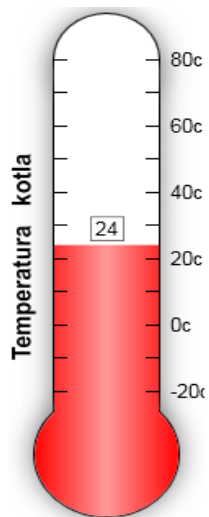
Izris grafičnega objekta izvedemo s klicem funkcije "Izris", ki izriše objekt na spletno stran. Funkcijo običajno kličemo v poljubnem časovnem intervalu, saj nam le tako grafični objekt prikazuje sveže informacije.

Primer klica funkcije "Izris":

```
termometer.Izris();
```

8.4.3 Termometer

Grafični gradnik Termometer (slika 44) nam na interaktiven način prikaže temperaturo. Lahko mu dodajamo lastnosti kot so naslov, oznake, maksimalna in minimalna temperatura in različne lastnosti, ki določajo izgled grafičnega gradnika. Stranska lestvica nam pomaga pri določitvi temperature, prav tako je temperatura izpisana s številko znotraj termometra. Lastnost limita določa, pri kateri vrednosti se bo barva termometra spremenila in tako opozorila operaterja na nepričakovan dogodek.



Slika 44: Grafični gradnik Termometer

8.4.4 Števec

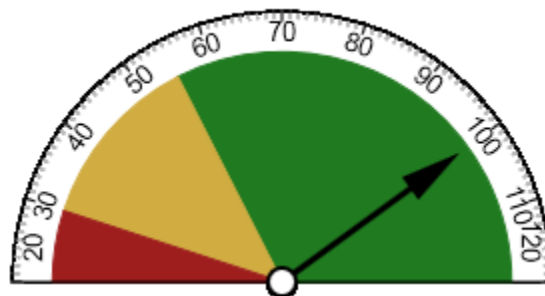
Grafični gradnik Števec (slika 45) nam prikaže veličino želenega parametra v obliki števca. Omogoča spreminjanje vizualne podobe, napisa v števcu, izpis enot, določanje natančnosti lestvice in druge lastnosti. Barvni nivo ob robu števca opozarja operaterja na mejne vrednosti in ga lahko po potrebah prilagodimo.



Slika 45: Grafični gradnik Števec

8.4.5 Polovični števec

Grafični gradnik Polovični števec (slika 46) nam s puščico pokaže veličino na lestvici v obliki polkroga. Barvni nivoji nam sporočajo kritično vrednost parametra. Vsebuje vse lastnosti grafičnega gradnika Števec.

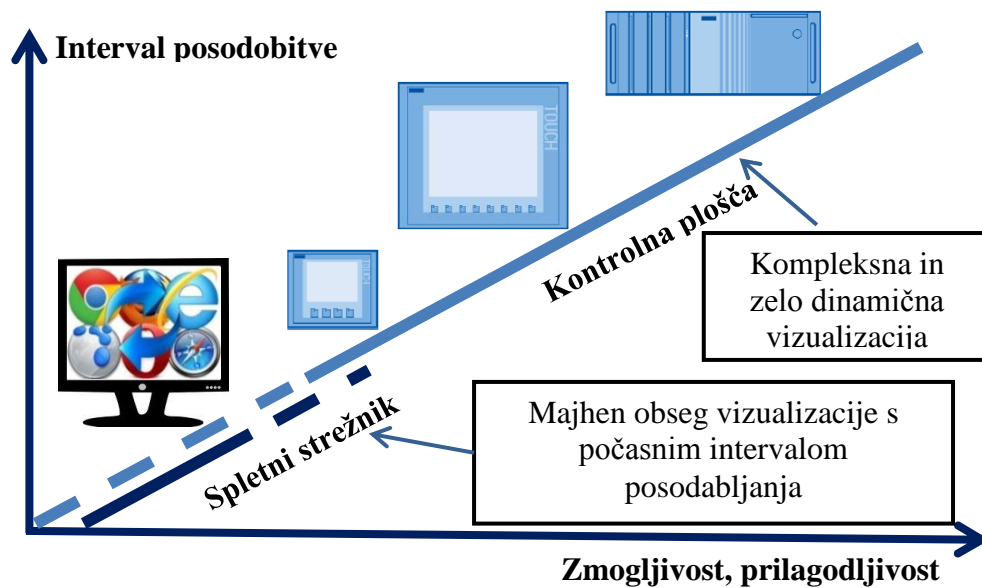


Slika 46: Polovični števec

9 Sklepne ugotovitve

Nadzorni sistem na spletnem strežniku krmilnika je primeren za majhne aplikacije z majhnim obsegom vizualizacije in zmogljivosti. Nadzorni sistem ni primeren za kritične procese, ki zahtevajo hitro reakcijo (manj kot pol sekunde), saj s počasnim odzivom ne ustreza zahtevam. Nadzorni sistem pride do svojega izraza, če nimamo nobenega namenskega sistema človek – stroj, vendar so diagnostične informacije in vizualizacija občasno potrebne. Nadzorni sistem lahko uporabljamo kot alternativo namenskemu nadzornemu sistemu, saj omogoča diagnostiko in spremljanje procesa na dolge razdalje.

Slika 47 prikazuje, kako se nadzorni sistem v spletnem strežniku glede izrisa na zaslon primerja z drugimi rešitvami.



Slika 47: Primerjava nadzornih sistemov glede na vizualizacijo in intervalom posodabljanja

Posamezna spletna stran naj ne bi vsebovala več kot štirideset spremenljivk iz kontrolnika. Če hočemo, da nadzorni sistem deluje stabilno in v realnem času, se moramo držati te omejitve. Grafika na spletni strani mora biti preprosta in dovolj razumljiva uporabniku. Pri tem moramo paziti na ločljivost zaslona. Spletno stran lahko oblikujemo tako, da se prilagaja določeni ločljivosti zaslona. Pri izdelavi moramo paziti tudi na osveževanje spremenljivk, saj prepogosto pošiljanje zahteve za prenos povzroči napako na spletnem strežniku.

Za lažjo predstavo in primerjavo dveh pristopov oziroma dveh različnih implementacij nadzornega sistema primerjamo izdelavo nadzornega sistema na spletnem strežniku krmilnika z izdelavo nadzornega sistema s programom WinCC flexible.

Za obe realizaciji potrebujemo določeno znanje, ki pa se popolnoma razlikuje. Za realizacijo nadzornega sistema na spletnem strežniku krmilnika potrebujemo znanje spletnih tehnologij in tehnologijo, ki predstavlja vmesni člen med spletno stranjo in krmilnikom. V našem primeru so predstavljali vmesni člen ukazi AWP, ki nam jih ponuja proizvajalec procesne opreme – v našem primeru Siemens. Za izdelavo nadzornega sistema s programom WinCC flexible potrebujemo znanje za rokovanje s programom.

S programom WinCC flexible enostavno kreiramo nov projekt. Čarovnik za izdelavo projekta nas popelje čez vse nastavitve projekta. Grafični vmesnik programa nam ponuja pogled na delovno površino, projekt, lastnosti in orodja. Pri implementaciji nadzornega sistema na spletnem strežniku krmilniku takih ugodnosti nimamo, a si lahko pomagamo z različnimi orodji za izdelavo spletnih strani (Dreamweaver, Frontpage).

V WinCC flexible enostavno konfiguriramo povezavo med krmilnikom in nadzornim sistemom. Ustvarimo simbolno tabelo spremenljivk in jo povežemo s spremenljivkami v kontrolniku. Na tabelo spremenljivk se sklicujemo v projektu WinCC flexible. Podatke v krmilnik pišemo in beremo preko simbolne tabele. Implementacija branja in pisanja v nadzornem sistemu na spletnem strežniku krmilnika je bolj kompleksna. Potrebno je izbrati spletno tehnologijo za prenos in jo implementirati, ob tem pa je še potrebno znanje za branje in pisanje spremenljivk kontrolnika.

V programu WinCC flexible s pomočjo čarovnika enostavno kreiramo zaslone. Zaslone so glavni elementi nadzornega sistema. Zaslone vsebujejo grafične gradnike, na primer vnosna polja, besedilo in slike. Zaslone v našem nadzornem sistemu predstavlja spletna stran. Nanj enostavno preko miške vlečemo in razporejamo grafične gradnike in jim dodajamo predpisane lastnosti. V našem primeru smo morali grafične gradnike ustvariti sami. Razporejanje elementov na spletni strani je bolj zapleten proces kot vlečenje in razporejanje elementov na zaslonu. Za dinamično funkcionalnost smo morali uporabiti svojo intuitivnost in znanje skriptnega jezika JavaScript, kar storimo v programu WinCC flexible enostavno z nastavitvami lastnosti elementa.

WinCC flexible nam ponuja funkcije za izdelavo alarmov in receptov. Alarme in recepte enostavno ustvarimo z uporabniškim vmesnikom programa. V vnosna polja želene vnašamo zahtevane podatke (ime, spremenljivke), sledimo postopku in želena funkcija je implementirana.

V nadzornem sistemu na spletnem strežniku krmilnika izdelavo implementiramo s pomočjo različnih spletnih tehnologij, kar zahteva veliko časa.

Izdelava nadzornega sistema s programom WinCC flexible je bolj enostavna in realizacija je hitrejša. Pri izdelavi smo omejeni na zmogljivost programa WinCC flexible. Izdelava nadzornega sistema na spletnem strežniku krmilnika zahteva več različnega znanja in intuitivnosti, izdelava je bolj kompleksna in zapletena, a vseeno ponuja več svobode.

Nadzorni sistem na spletnem strežniku krmilnika ima svoje slabosti in prednosti. Prednosti nadzornega sistema so prenosljivost, oddaljen dostop, cena (programska oprema), mobilne rešitve in svoboda pri realizaciji. Prednosti za izbiro nadzornega sistema tega tipa je veliko, predvsem je pomembna prenosljivost. Za dostopanje do nadzornega sistema ne potrebujemo posebne programske opreme, temveč samo spletni brskalnik. Pri izbiri igrata pomembno vlogo cena in zahteve nadzornega sistema.

Slabosti nadzornega sistema so kompleksna izdelava, omejitve glede vizualizacije, osveževanje, stabilnost in majhni sistemi. Slabost nadzornega sistema je predvsem v omejitvi kompleksnosti sistema. Nadzorni sistem je primeren za nadzor in pregled majhnih procesov.

Med implementacijo smo se srečali predvsem z neznanjem implementiranja določenih funkcij, a smo to tekom izdelave uspešno reševali. Veliko časa smo posvetili nastavitvam spletnega strežnika. Težavo vpisa uporabnika v sistem in posledično preprečevanje nadzornemu sistemu vpisovanja v krmilnik smo uspešno rešili z preizkušanjem različnih nastavitvev spletnega strežnika krmilnika. Drugih večjih težav nismo imeli.

Literatura

- [1] (2012) Adobe, "Dreamweaver". Dostopno na:
<http://www.adobe.com/products/dreamweaver.html>.
- [2] (2012) Firebug Working Group, "Firebug ". Dostopna na:
<https://getfirebug.com/>.
- [3] (2012) JetBrains, "WebStorm". Dostopno na:
<http://www.jetbrains.com/webstorm/>.
- [4] (2010) Mihai Sucan, "SVG or Canvas". Dostopno na:
<http://dev.opera.com/articles/view/svg-or-canvas-choosing-between-the-two/>
- [5] (2012) Siemens AG, *SIMATIC Controller Software*, Nürnberg, 2012. Dostopno na:
http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-industrial-software_en.pdf.
- [6] Siemens AG, *S7-1200 Programmable controller*, Nürnberg, 2012, pogl.11. Dostopno na:
<http://www.automation.siemens.com/mdm/default.aspx?DocVersionId=37249085323&Language=en-US&TopicId=>.
- [7] (2012) Siemens, "WinCC/Web Navigator". Dostopno na:.
<http://www.automation.siemens.com/mcms/human-machine-interface/en/visualization-software/scada/wincc-options/wincc-web-navigator/pages/default.aspx>.
- [8] Siemens AG, *SIMATIC WinCC flexible*, Nürnberg, 2010. Dostopno na:
http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-wincc-flexible_en.pdf.
- [9] Siemens AG, *Tia portal*, Nürnberg, 2010. Dostopno na:
<https://c4b.gss.siemens.com/resources/images/articles/e20001-a390-p230-x-7600.pdf>.
- [10] Siemens AG, *Web server*, Nürnberg, 2011. Dostopno na:
<https://c4b.gss.siemens.com/resources/images/articles/e20001-a390-p230-x-7600.pdf>.
- [11] Siemens AG, *Creating and Using Own Web Pages for S7-1200*, Nürnberg, 2012. Dostopno na:

http://cache.automation.siemens.com/dnl/zA/zAxNDQ3AAAA_58862931_Tools/58862931_S7-1200_Webserver_DOKU_v10_en.pdf.

[12] Siemens AG, *Visualization with User-defined Web Pages*, Nürnberg, 2010. Dostopno na: http://cache.automation.siemens.com/dnl/TM/TM0ODIzMQAA_44212999_Tools/44212999_AWP_PN_CPU_V10_e.pdf.

[13] Siemens AG, *S7-300 CPU 31xC and CPU 31x: Technical specifications*, Nürnberg, 2011, pogl. 3.7 . Dostopno na: http://www.automation.siemens.com/doconweb/pdf/SINUMERIK_SINAMICS_04_2010_E/S7_BA31A.pdf?p=1.

[14] Uroš Lotrič, *PLK strojna oprema*, 2010. Dostopno na: <http://ucilnica.fri.uni-lj.si/course/view.php?id=53>.

[15] Zuri Evans, *Web Server Technology in Automation*, Nocross, 2011. Dostopno na: http://w3.usa.siemens.com/us/internetdms/dt/DrivesSolutions/DrivesSolutions/Docs3_MotionControllers/MC-SIMOTION_Web_Server-Presentation.pdf

[16] (2012) Wikipedia, "Programmable logic controller". Dostopno na: http://en.wikipedia.org/wiki/Programmable_logic_controller.

[17] (2012) Wikipedia, "Fischertechnik". Dostopno na: <http://en.wikipedia.org/wiki/Fischertechnik>.

[18] (2012) Wikipedia, "HTML". Dostopno na: <http://en.wikipedia.org/wiki/HTML>.

[19] (2012) Wikipedia, "CSS". Dostopno na: http://en.wikipedia.org/wiki/Cascading_Style_Sheets.

[20] (2012) Wikipedia, "Industrial Ethernet". Dostopno na: http://en.wikipedia.org/wiki/Industrial_Ethernet.

[21] (2012) Wikipedia, "Ajax". Dostopno na: http://en.wikipedia.org/wiki/Ajax_%28programming%29.

[22] (2012) Wikipedia, "JavaScript". Dostopno na: <http://en.wikipedia.org/wiki/JavaScript>.

[23] (2012) W3Schools, "HTML5 Canvas". Dostopno na:
http://www.w3schools.com/html/html5_canvas.asp.

[24] (2012) W3Schools, "Inline SVG". Dostopno na:
http://www.w3schools.com/html/html5_svg.asp.