Sustav automatskog upravljanja garažnim vratima s mogućnošću upravljanja i nadzora putem Interneta

Hajduković, Matija

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

SUSTAV AUTOMATSKOG UPRAVLJANJA GARAŽNIM VRATIMA S MOGUĆNOŠĆU UPRAVLJANJA I NADZORA PUTEM INTERNETA

Diplomski rad

Matija Hajduković

Osijek, 2022.

SADRŽAJ

1. UVOD

U današnjem svijetu, u kojem rastu zahtjevi za povezivošću, pa tamo gdje to do sada bilo nezamislivo ili teško izvedivo, potrebno je postaviti odgovarajuću sigurnosnu zaštitu za sustave s udaljenim upravljanjem. Korisnici očekuju sustave koji omogućuju upravljanje izvršnim elementima neovisno o udaljenosti i uvjetima u kojima se nalaze. Takvi sustavi, koji za svoje usluge koriste javnu internetsku mrežu, uvode nove ranjivosti, koje mogu biti korištene za krađu podataka, krađu imovine ili onemogućavanje ispravnog rada sustava. Iz prethodno navedenih razloga potrebno je voditi računa o kibernetičkoj sigurnosti. Kibernetička sigurnost postaje sastavni dio modernih sustava nadzora i upravljanja.

Unutar ovog rada realiziran je sustav automatskog upravljanja garažnim vratima s pripadajućim sučeljem čovjek-stroj, koji ima mogućnost udaljenog (engl. *remote*) upravljanja kao i ručnog upravljanja pored samih garažnih vrata. Sučelje čovjek-stroj implementirano je na pripadajućem web serveru PLC uređaja te je izrađeno u obliku web stranice. Komunikacija između opratera (korisnika) i PLC uređaja ostvarena je putem javne Internetske mreže. Nakon osmišljavanja i realizacije sustava posebno je obrađen aspekt kibernetičke sigurnosti i izvršen je *brute-force* napad kojim se pokušala zadobiti kontrola nad sustavom.

U drugom poglavlju razmatra se oprema koja je korištena, a u trećem poglavlju objašnjeno je na koji su način upravljački sustav i sučelje čovjek-stroj implementirani. Unutar četvrtog poglavlja objašnjava se način na koji se sustav testira i pušta u pogon. Ovo poglavlje osvrće se na korisnike koji imaju pristup kao i načine na koji je sustav testiran. U posljednjem, petom, poglavlju prikazane su neke od aktualnih sigurnosnih ranjivosti, koje su bile aktualne. Objašnjava se njihov mehanizam, razmjer potencijalne štete te moguće protumjere. Također je unutar tog poglavlja izveden penetracijski tekst koristeći *hydra* program kako bi se pokušala zadobiti kontrola nad sustavom.

1.1. Zadatak diplomskog rada

Potrebno je osmisliti i realizirati jednostavan sustav automatskog upravljanja i pripadajuće sučelje čovjek-stroj koristeći PLC uređaj iz serije Siemens S7-1500. Kao objekt upravljanja koriste se kućna garažna vrata. Na primjeru upravljanja izvršnim elementom garažnih vrata, a to je

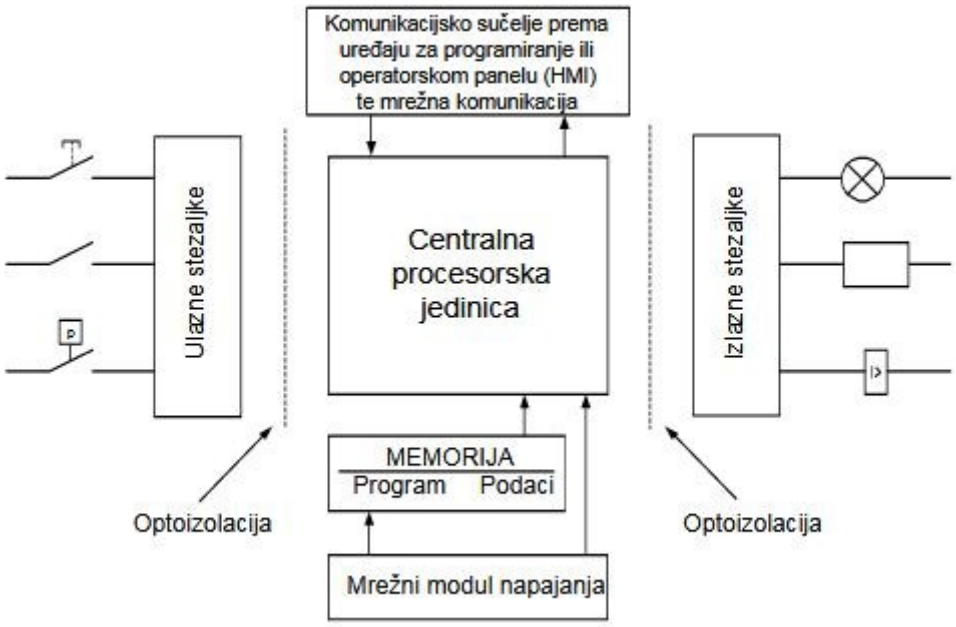
elektromotor s dva smjera i izravnim uklopom, pokazuju se osnovna načela programiranja izvršnih, zaštitnih i blokadnih funkcija upravljanja aktuatorima. Programsko rješenje mora obuhvaćati više programskih jezika prema IEC 61131-3 Osim izravnog upravljanja pomoću tipkala, treba realizirati nadzor i upravljanje garažnim vratima pomoću ugrađenog Web poslužitelja, na kojeg se korisnik može spojiti mobilnim telefonom. Posebnu pažnju potrebno je posvetiti kibernetičkoj sigurnosti ovakvog rješenja.

2. GARAŽNA VRATA I OPREMA ZA REALIZACIJU SUSTAVA UPRAVLJANJA

Sustav garažnih vrata sastoji se od programabilnog logičkog kontrolera (engl. *Programmable Logic Controller,* PLC), na kojem postoji ugrađeni web poslužitelj pomoću kojeg je moguće upravljati sustavom. Izvršni element garažnih vrata je elektromotora s dva smjera s izravnim uklopom. Sučelje čovjek-stroj implementirano je kao web stranica i postavljeno na web server, koji se nalazi na samom PLC uređaju. U ovom poglavlju opisana je potrebna oprema i korištene tehnologije za ovaj sustav.

2.1. Programabilni logički kontroler

Programabilni logički kontroleri predstavljaju ugrađeno industrijsko računalo, koji sadrže automatizacijske funkcije spremljene u obliku programa u samoj memoriji upravljača. Uvedeni su kao zamjena za upravljačke ormare s relejima i sklopnicama, koji su bili spojeni žicama. Njihove osnovne prednosti su fleksibilnost, male dimenzije i lakoća uporabe. Struktura samog PLC-a može se podijeliti na 4 dijela, koja su prikazana na slici 2.1.



Slika 2.1. *Struktura programabilnog logičkog kontrolera [2].*

Na ulazni se dio kontrolera pomoću stezaljki dovode ulazni signali iz procesa kojim se upravlja. Signale koje PLC prihvaća na svoje stezaljke dijele se na analogne i digitalne signale. Pri tome se kod digitalnog signala razlikuju dva stanja, visoko i nisko stanje. Visoko stanje podrazumijeva naponsku razinu od 14-30 VDC, a nisko 0-5 VDC. Analogna informacija ima drugačiju podjelu, na strujni ili naponski signal. Strujni analogni signal može biti ili 0-20 mA ili 4-20 mA dok naponski može biti 0-10 VDC ili -10 - +10 VDC. Svaki analogni signal može imati različitu rezoluciju te također postoje kartice koje mjere otpor.

Izlazni dio, kao i ulazni, sadrži stezaljke na koje se spajaju uređaji kojima kontroler šalje digitalnu ili analognu informaciju te na taj način upravlja procesom. Te su stezaljke optoizolirane od procesorske jedinice kako bi se galvanski odvojili krugovi.

Centralna procesorska jedinica (engl. *Central Processing Unit,* CPU) zadužena je za kontrolu i nadziranje svih operacija unutar PLC-a. Procesorska jedinica čita stanja svih ulaza samog uređaja (analognih i digitalnih), obrađuje ih prema programu koji je napisan, te upravlja izlazima sukladno tomu. Povezivanje sa samom procesorskom jedinicom, a time i kontrolerom, radi se pomoću Profibus/MPI sučelja ili Etherneta.

Memorijski blok dijeli se na 4 segmenta [30]:

- Radna memorija koda (engl. *Code work memory*) – Sadrži funkcijske blokove, organizacijske blokove i funkcije,
- Radna memorija podataka (engl. *Data work memory*) – sadrži globalne podatkovne blokove, instance podatkovnih blokova i sl.,
- engl. *Retentive memory* – Sadrži dijelove globalnih podatkovnih blokova, instance podatkovnih blokova, brojače vremena i sl.,
- Dodatna memorija (engl. *Additioanl memory areas*) – Brojači vremena, brojači, privremeni lokalni podaci i sl.

S7-1500

Za sustav automatskog upravljanja i pripadajućeg sučelja čovjek-stroj koristi se PLC uređaj iz serije Siemens S7-1500, točnije S7-1513-1 PN, prikazan na slici 2.2. Taj uređaj odabran je ponajprije zbog ugrađenog web servera, koji je moguće koristiti čak i kada je PLC zaustavljen.

Slika 2.2. *S7-1513-1 PN programabilni logički kontroler [3].*

Sam kontroler još nudi dodatne funkcije poput [4]:

- Komunikacija putem Ethernet/PROFINET protokola
- HMI (engl. *Human Machine Interface*) komunikacija
- Komunikacija putem OPC UA (engl. *Open Platform Communication Unified Arhitecture*) protokola

Navedeni PLC može se proširiti i s dodatnim periferijskim modulima. Bitna je stavka za taj sustav integrirani web server, koji omogućuje nadziranje i administraciju CPU-a od strane autoriziranih korisnika. Samim time omogućena je monitoring, dijagnosticiranje PLC-a i modifikacije varijabli bez korištenja programa kao što su TIA Portal ili bez mijenjanja stanja ulaza kontrolera pomoću tipkala i sl., već je samo potreban web preglednik (Mozilla Firefox, Google Chrome, Mobile Safari, Android Browser, …). Web server sadrži dijagnostički međuspremnik,

korisnički definiranu web stranicu (koja se u ovom slučaju koristi kao sučelje čovjek-stroj), identifikacijske podatke o PLC-u i slično. Početnu stranu web servera moguće je vidjeti na slici 2.3.

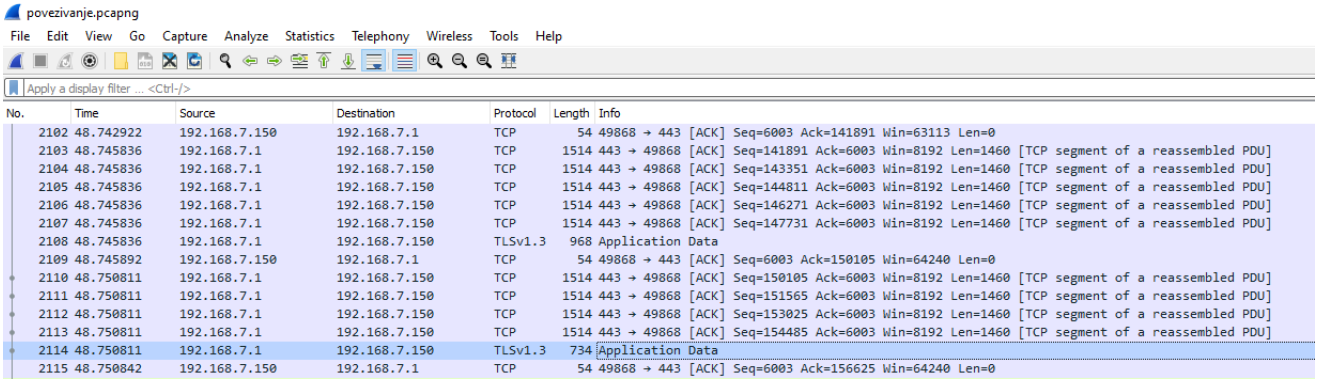
Slika 2.3. *Prikaz početne stranice web servera.*

Sam web server u sebi već ima ugrađene neke sigurnosne funkcije, kao što su [5]:

- Pristup preko HTTPS (engl. *Hypertext Transfer Protocol Secure*) protokola koristeći CA potpisan certifikat
- Mogućnost konfiguracije autorizacije korisnika putem korisničkih lista
- Mogućnost zabrane pristupa pojedinom korisniku određenim sučeljima

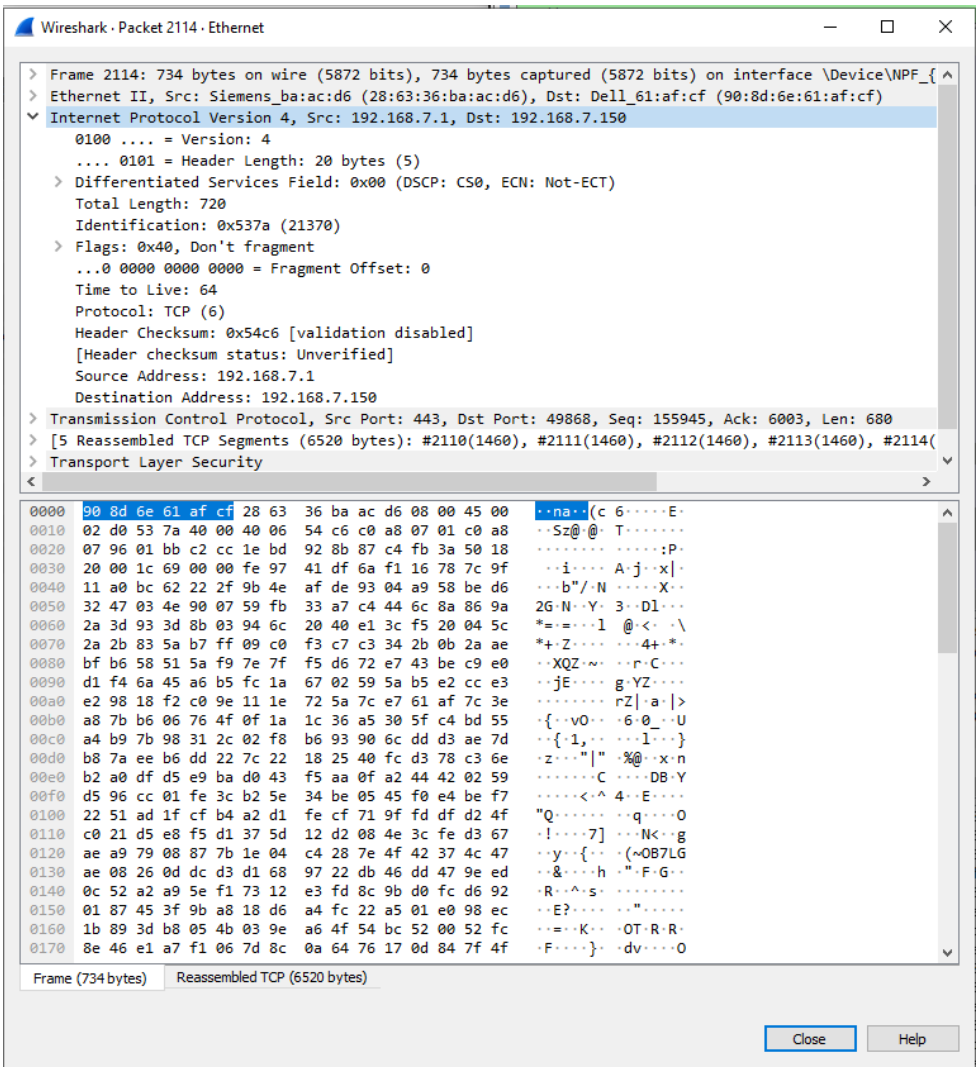
Ovisno o web pregledniku različiti broj komunikacijskih veza moguć je između CPU-a i samog web servera. Što je više veza omogućeno to će se više korisnika moći spojiti, ali jedan uređaj može biti prijavljen na samo jedan korisnički račun. Ako više nema slobodnih veza, problemi u prikazu ili funkcionalnosti mogu se pojaviti budući da web server počinje odbijati sve ostale veze koje se pokušavaju spojiti.

Komunikacija između web servera i korisničke aplikacije (web preglednika) odvija se većinom isključivo putem TCP (engl. *Transmission Control Protocol*) protokola. To je moguće i dokazati korištenjem Wireshark programskog alata koji može snimiti promet koji se šalje kroz mrežu. Prikaz prometa između sučelja čovjek-stroj i upravljačkog kôda moguće je vidjeti na slici 2.4.



Slika 2.4. *Wireshark prikaz komunikacije između sučelja i web servera.*

192.168.7.1 predstavlja IP adresu PLC uređaja, odnosno web servera, a 192.168.7.150 je IP adresa računala s kojeg se pristupa web serveru. Vidi se kako oni međusobno razmjenjuju pakete, a ako se detaljnije prouči jedan od njih, može se vidjeti da je zapravo riječ o TCP/IP (engl. *Transmission Control Protocol/Internet Protocol*) paketima. Dodatno se može primijetiti TLS (engl. *Transport Layer Security*) protokol, koji predstavlja kriptografski protokol koji pruža sigurnost prilikom komunikacije putem interneta. Najvidljivija upotreba tog protokola može se vidjeti kod osiguravanja HTTPS protokola. Na slici 2.5. vidi se prikaz 2114 paketa unutar kojeg se vidi IP protokol (engl. *Internet Protocol*) koji se koristi za logičko adresiranje te već spomenute TLS i TCP protokole.



Slika 2.5. *Paket 2114 unutar Wireshark programa.*

Transportni protokoli nikad nisu meta napada, već ako su nesigurni, mogu poslužiti za realizaciju napada. TCP protokol se koristi za slanje podataka koji pokušavaju onesposobiti sustav ili omogućiti neautorizirani pristup sustavu. U ovom sustavu najveću ranjivost predstavlja web poslužitelj, koje je potrebno zaštiti da napadač ne bi dobio administrativne ovlasti i time ugrozio cijeli sustav.

2.2. Funkcija sustava garažnih vrata

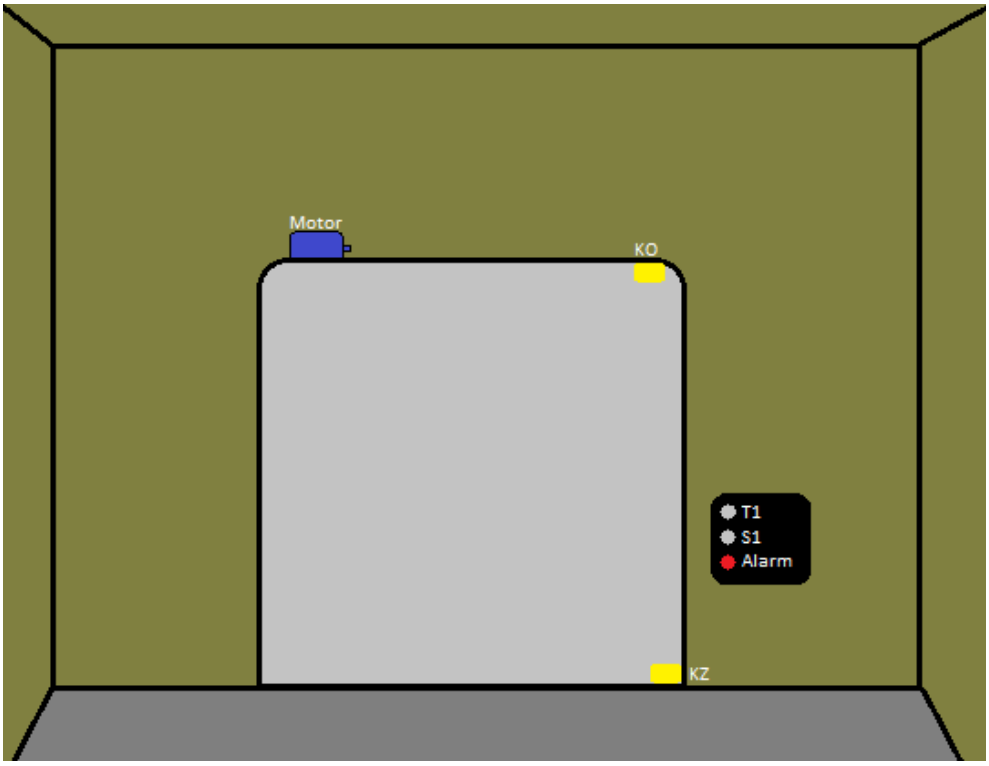
Vrata su u normalnom radu otvorena ili zatvorena. Mogu ostati u među položaju samo ako se pojavi smetnja, odnosno alarmantno stanje. Prilikom uključenja PLC-a, ako garažna vrata nisu

zatvorena, treba ih zatvoriti. Pojava bilo koje smetnje na vratima njih i zaustavlja, odnosno blokira upravljanje vratima sve dok se smetnja ne otkloni i potvrdi. Ako su vrata otvorena dulje od 2 minute, automatski se zatvaraju. Samim garažnim vratima moguće je upravljati iz garaže ili s ulice.

Sustav se sastoji od sljedećih ulaznih signala:

- Multifunkcionalno tipkalo T1 u garaži za otvaranje, zatvaranje vrata i potvrdu smetnje,
- Multifunkcionalno tipkalo T2 s ulične strane za otvaranje, zatvaranje vrata i potvrdu smetnje,
- Grebenasta sklopka S1 (0-1) u garaži za dozvolu otvaranja vrata s ulične strane,
- Krajnji kontakt KO za signalizaciju položaja „vrata otvorena”,
- Krajnji kontakt KZ za signalizaciju položaja „vrata zatvorena”,
- Signal s sklopnika K1 „vrata u otvaranju”,
- Signal s sklopnika K2 „vrata u zatvaranju”,
- Signal s motorne zaštitne sklopke MZS „ispad”.

Upravljanje vratima omogućeno je s dvije pozicije. Unutarnji panel sastoji se od tipkala T1 i grebenaste sklopke S1. Pritiskom na tipku T1 vrata se zatvaraju ili otvaraju ovisno o tome jesu li zatvorena ili otvorena. Ako su vrata u međupoložaju pritiskom na tipkalo, vrata se nastavljaju kretati u smjeru kretanja prije zaustavljanja. Na slici 2.6. vidi se prikaz sustava garažnih vrata iz garaže.



Slika 2.6. *Prikaz sustava garažnih vrata iz garaže.*

Vanjski panel koji se nalazi na uličnoj strani sadrži tipkalo T2, koje ima istu funkciju kao i unutrašnje tipkalo T1. Jedina razlika je ta što to tipkalo ima funkciju samo u slučaju kada je grebenasta sklopka S1 u položaju 1, tj. kada je omogućeno upravljanje s ulične strane. Na slici 2.7. može se vidjeti prikaz sustava garažnih vrata s ulične strane.

Slika 2.7. *Prikaz sustava garažnih vrata s ulice.*

Pri pojavi smetnje, odnosno alarma, vrata se odmah zaustavljaju i dojavljuje se alarm isprekidanim impulsom od 0.5 Hz. Pritiskom na tipkalo T1 ili T2 potvrđuje se smetnja i dojava alarma prestaje. Smetnje koje odmah zaustavljaju vrata su:

1. Ispad motorne zaštitne sklopke (MZS),
2. Oba položaja vrata aktivna (KO i KZ),
3. Vrata nakon zadane komande nisu došla u željeni položaj u vremenu od 30 sekundi,
4. 5 sekundi nakon komande za otvaranje ili zatvaranje nije stigla odgovarajuća potvrda smjera (K1/K2).

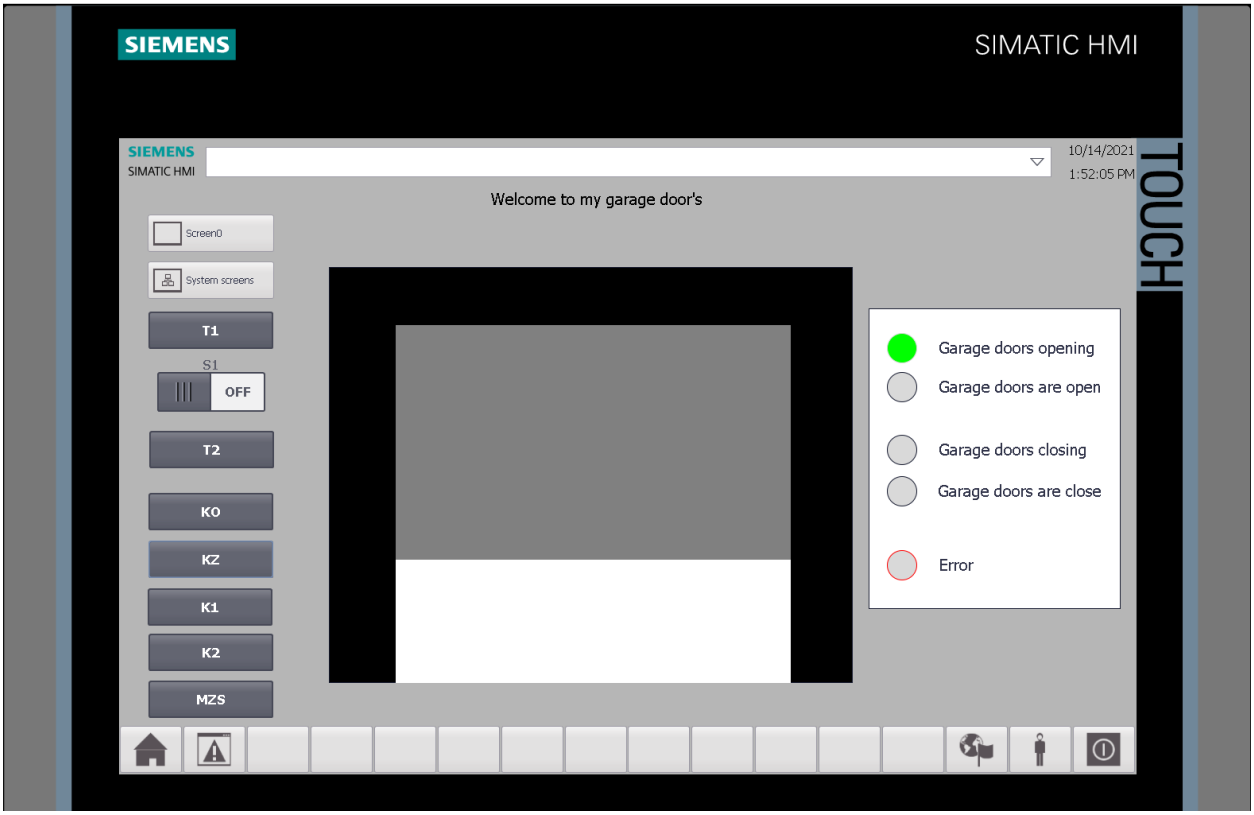
2.3. Sučelje čovjek-stroj

Sučelje čovjek-stroj (engl. *Human Machine Interface*, HMI) sastoji se od hardvera i softvera, koji omogućuju pretvorbu korisničkih ulaza i podataka koji su pogodni za obradu od strane CPU-a i/ili uvid operatera u stanje procesa. Iako se termin 'HMI' može upotrijebiti za svaki interaktivni

računalni sustav koji omogućava korisniku/operateru interakciju s nekakvim uređajem, sam pojam HMI koristi se ako je riječ o industrijskom procesu. Primjer HMI se vidi na slici 2.8.

Kao što je već napomenuto, HMI u industrijskom okruženju omogućuje [7]:

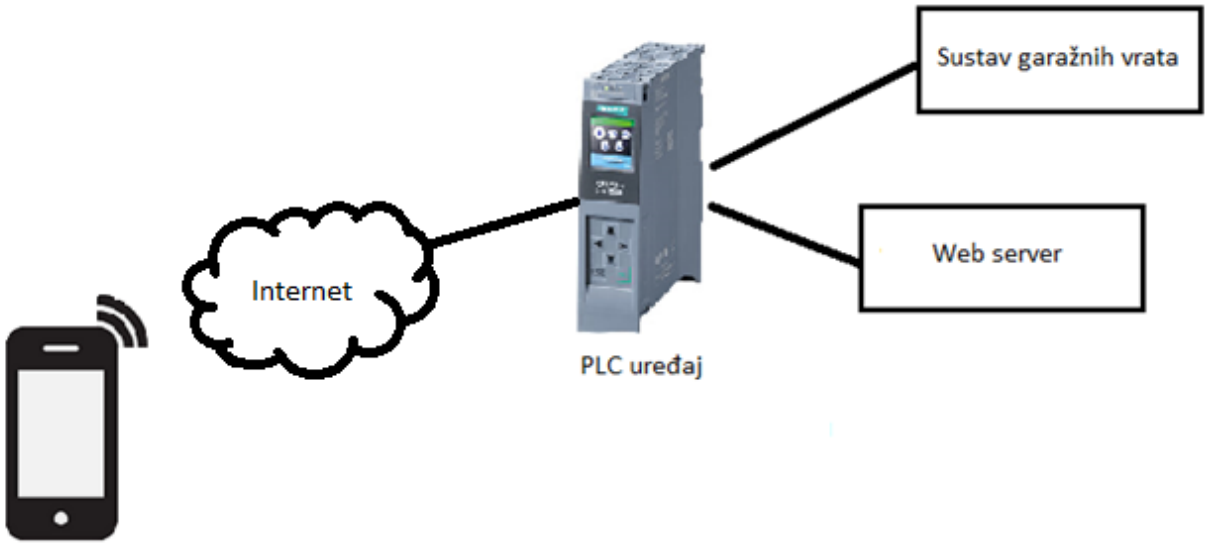
- Vizualizaciju podataka,
- Praćenje trendova, alarma, upozorenja te procesnih i sistemskih događaja,
- Nadgledanje indikatora ključnih učinka (engl. *Key Performance Indicator*, KPI),
- Nadgledanje ulaza i izlaza PLC-a.



***Slika 2.8.**: Prikaz sučelja čovjek-stroj u TIA Portal v17 programu.*

Napretkom tehnologije došlo je i do napretka u sučeljima, od HMI-a visokih performansi, koji osiguravaju brzu i efektivnu interakciju operatera preko mobilnih uređaja, i sučelja osjetljivih na dodir do *Edge-of-Network* i HMI koji šalju podatke na oblak (engl. *Cloud*).

U ovom radu sučelje čovjek-stroj, umjesto u obliku odvojenog uređaja, implementirano je u obliku ugrađenog web servera, koji je sastavni dio PLC uređaja. Spojni put između sustava upravljanja i HMI-a je Internet i prikazan je na slici 2.9.



Slika 2.9. *Prikaz spojnog puta između sučelja čovjek-stroj i samog PLC uređaja.*

Samom sučelju moguće je pristupiti putem bilo kojeg uređaja koje ima omogućen pristup internetu putem web preglednika, kao što su mobilni uređaj, tablet, laptop i sl. Iako su na samom PLC uređaju konfigurirani različiti korisnici koji imaju različita prava pristupa, svatko može pristupiti početnoj stranici. Budući da se PLC uređaju, to jest njegovom web serveru, može pristupiti iz javne mreže, te s obzirom na to da se putem tog web sučelja može upravljati procesom ili rekonfigurirati pojedine komponente, takav je sustav kibernetički ranjiv te ga je potrebno štititi različitim tehničkim i/ili administrativnim mjerama.

3. REALIZACIJA SUSTAVA UPRAVLJANJA GARAŽNIM VRATIMA

Softver za PLC programiran je TIA Portal razvojnim okruženjem pomoću SCL i LAD programskog jezika prema IEC 61131-3 standardu dok je za sučelje čovjek-stroj korišten Visual Studio Code. Unutar ovog poglavlja objašnjeni su specifikacije sustava, programsko okružje i programski jezici koji su korišteni te softverski kôd.

3.1. Izrada programa za PLC

Upravljački dio sustava izrađen je pomoću TIA Portal razvojnog okruženja koristeći SCL i LAD programski jezik.

3.1.1. TIA Portal

Za izradu softvera za PLC uređaj korišten je TIA Portal (engl. *Totally Integrated Automation Portal*). TIA portal je softverska aplikacija, koja integrira različite SIMATIC proizvode u jednu aplikaciju. Omogućava lako rukovanje podacima, lak prikaz podataka, izradu sučelja čovjek-stroj i sl. Za razliku od ostalih programa PLC upravljački program i HMI sučelje razvijaju se unutar istog alata, što pruža različite mogućnosti integracije te ubrzava proces izrade sustava. Program korisniku pruža vrlo jednostavno sučelje za upravljanje upravljanje procesima te izradu korisničkog sučelja [8].

Prilikom programiranja PLC-a unutar TIA Portala korišteni su tzv. blokovi [8]:

- Podatkovni blok (engl. *Data block,* DB) – koristi se za spremanje vrijednosti ili znakovnih nizova. Predstavljaju globalne blokove kojima se može pristupiti iz svakog koda.
- Funkcijski blok (engl. *Function block,* FB) – predstavlja blok sa statičnim podacima kojemu je pridružen podatkovni blok. Parametrima ovog bloka može se pristupiti u svakom trenutku i u bilo kojoj točki korisničkog programa.
- Organizacijski blok (engl. *Organization block*, OB) – predstavlja sučelje između centralne procesorske jedinice i korisničkog programa.
- Funkcija (engl. *Function*, FC) – predstavlja blok bez memorije koji sadrži program koji se izvršava prilikom poziva same funkcije.

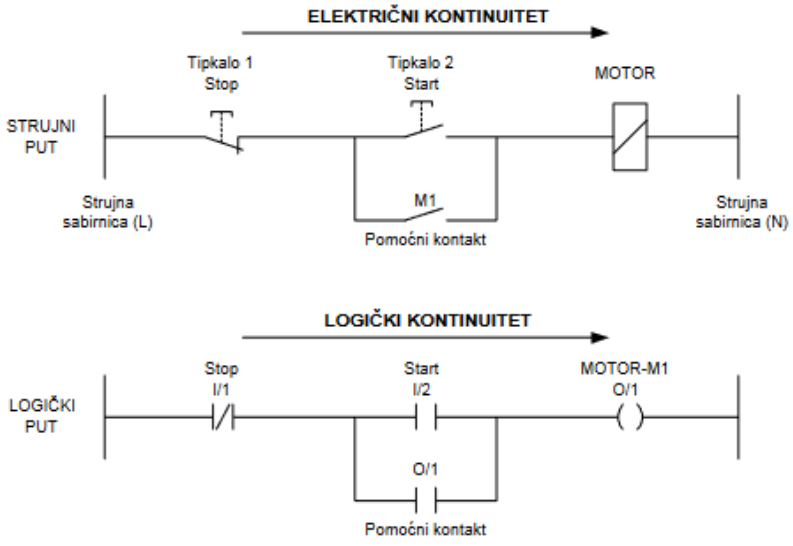
Postoje nekoliko najpopularnijih programskih jezika za programiranje samog PLC-a, a to su:

- Ljestvičasti dijagram (engl. *Ladder Diagram*, LAD),
- Strukturirana lista (engl. *Statement list*, STL),
- Funkcijski blok dijagram (engl. *Function block Diagram*, FBD),
- Strukturirani kontrolni tekst (engl. *Structured Controle Language*, SCL).

3.1.2. Ljestvičasti dijagram

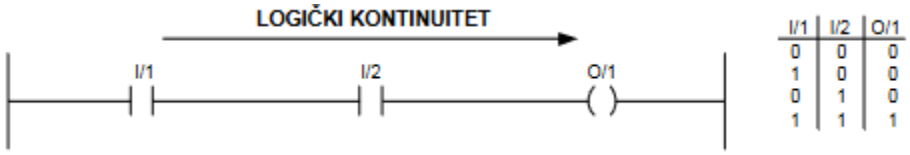
Ljestvičasti dijagram (engl. *Ladder diagram*) programski je jezik koji je nastao na bazi strujnih shema kojima se zapravo prikazuje protok struje u strujnom krugu i predstavlja jedan od najkorištenijih načina programiranja. Razlog tomu leži u jednostavnosti zbog koje je mnogi tehničari i inženjeri mogu razumjeti. Ovaj programski jezik može se razumjeti kao da se s lijeve strane nalazi pozitivan pol, a s desne negativan pol strujne petlje [9].

Shema strujnog kruga, prikazana na slici 3.1., prikazuje stanje kontakata, jesu li pritisnuti ili ne. Ljestvičasti dijagram ispituje je li nekakva naredba istinita (logička jedinica '1') ili nije (logička nula '0') te na temelju toga pali ili gasi određeni izlaz; na slici 3.1. to je predstavljeno motorom.



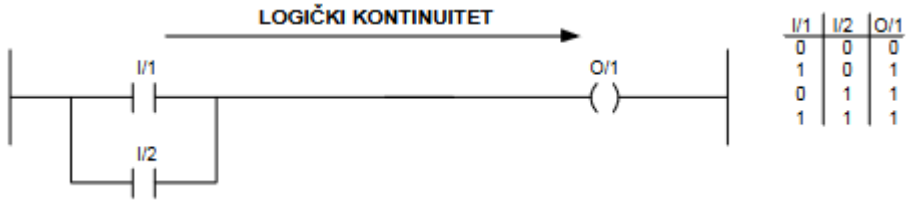
Slika 3.1. *Usporedba strujnog i logičkog puta [9].*

Programiranje ljestvičastim dijagramom izvodi se pomoću tri osnovne logičke naredbe, 'I', 'ILI' i 'NE'. Na slici 3.2. vidi se prikaz logičke 'I' operacije gdje se vidi kako ulazi *I/1* i *I/2* moraju biti pritisnuti kako bi se izlaz O/1 postavio u logičko stanje '1'.



Slika 3.2. *Logička I operacija [9].*

Na slici 3.3. vidi se prikaz logičke 'ILI' operacije gdje ili ulazi *I/1* ili *I/2* moraju biti u stanju logičke jedinice kako bi se izlaz *O/1* postavio u logičku jedinicu.



Slika 3.3. *Logička ILI operacije [9].*

Kombinacijom tih dvaju logičkih operacija izvode se ostali logički sklopovi kao što su isključivo 'ILI' ('XILI'), logički sklop 'NI' ili nekakvi složeniji Boolovi izrazi.

3.1.3. Strukturirani kontrolni jezik

Strukturirani kontrolni jezik (SCL) predstavlja programski jezik visoke razine, koji je sličan PASCAL-u. SCL je uveo funkcionalnosti koje do sad nisu bile moguće programiranjem pomoću ljestvičastog dijagrama, kao što su FOR i WHILE petlje i CASE izrazi. Ako se ukaže potreba za korištenjem polja podataka (engl. *array*), SCL korištenjem FOR petlji i IF izraza može puno brže i efikasnije pretražiti potrebno polje i odraditi određenu zadaću [10]. Primjer SCL koda može se pogledati na slici 3.4.

```
// Re-Initialize static variables
IF #xRestart THEN
    #sValues[0] := #sValues[1] := #xCurrIntegral := #OUT := 0;
END_IF;
```

Slika 3.4. *Primjer strukturiranog kontrolnog jezika.*

3.1.4. Programsko rješenje sustava upravljanja

Programski kôd podijeljen je na 6 dijelova:

1. Automatsko zatvaranje garažnih vrata,
2. Davanje dozvole za otvaranje/zatvaranje vrata,
3. Događaja pritiska tipkala,
4. Pamćenje posljednjeg smjera,
5. Otvaranje/Zatvaranje garažnih vrata,
6. Dojava alarma.

Automatsko zatvaranje garažnih vrata

Ako su vrata otvorena duže od dvije minute, potrebno ih je automatski zatvoriti. To se ostvaruje korištenjem ugrađenih brojača i KO-a. Na „ulaz" brojača, na slici 3.5. *#IEC_Timer_2min_alarm*, dovodi se KO signal, koji, ako je u logičkom stanju 1, pali brojač koji nakon dvije minute postavlja varijablu *#xAuto_close* u logičku jedinicu. Ako se ona postavi u logičku jedinicu, postavlja se zastavica *#sCls_gate*, koja započinje zatvaranje vrata.

```
REGION _AUTO_CLOSE
    // If gate is opened more than 2 min close it
    #IEC_Timer_2min_alarm(IN := #KO,
                          PT := #timer_2min,
                          Q => #xAuto_close,
                          ET => #xT);


    IF #xAuto_close THEN
        #sCls_gate := TRUE;
    END_IF;
END_REGION
```

Slika 3.5. *Kôd za automatsko zatvaranje garažnih vrata.*

Davanje dozvole za otvaranje/zatvaranje vrata

Sustav je isprogramiran da se za otvaranje/zatvaranje garažnih vrata prvo mora dati dozvola te se tek onda može krenuti u otvaranje/zatvaranje. Dozvola za otvaranje vrata izdaje se tako da se radi logička 'I' operacija između varijable *#KZ*, koja govori jesu li vrata potpuno zatvorena, i svih alarmnih stanja među kojima je napravljena logička 'ILI' operacija te potom logička operacija 'NE'. Ako je samo jedan od alarmnih stanja u logičkom stanju 1, konačni negirani rezultat je logička 0, što u krajnjem slučaju rezultira da varijabla *#xPerm_opn* ima vrijednost logičke 0, tj. da nema dozvole za otvaranje vrata. Na ovaj način, prikazan na slici 3.6., na efikasan se i vrlo čitljiv način omogućava otvaranje vrata samo ako su zatvorena i ako trenutno nema nikakvih aktivnih alarma.

```
REGION _PERMISSIONS
    // Permissions open the door
    IF #KZ AND NOT (#sAlarm_MZS OR #sAlarm_K1K2 OR #sAlarm_KOKZ OR #sAlarm_stuck) THEN
        #xPerm_opn := TRUE;
    ELSE
        #xPerm_opn := FALSE;
    END_IF;

    // Permission to close the door
    IF #KO AND NOT (#sAlarm_MZS OR #sAlarm_K1K2 OR #sAlarm_KOKZ OR #sAlarm_stuck) THEN
        #xPerm_cls := TRUE;
    ELSE
        #xPerm_cls := FALSE;
    END_IF;
END_REGION
```

Slika 3.6. *Kôd za dozvolu otvaranja/zatvaranja garažnih vrata.*

Događaj pritiska tipkala

Postoje dva tipkala koja se mogu pritisnuti. Tipkalo T1 s unutrašnje strane i tipkalo T2 s vanjske strane, za koje još mora i grebenasta sklopka S1 biti u položaju 1. Nakon što je dodijeljena dozvola pritiskom na tipku T1 ili T2, moguće je otvoriti zatvorena vrata ili zatvoriti otvorena vrata. Također pritiskom na tipku T1 ili T2, ako je on uključen, alarm se potvrđuje i gasi te ponovnim pritiskom na tipku vrata nastavljaju s ponašanjem koje je bilo prije podizanja alarma. Dodatno se uvode dvije nove varijable, *#sOpening* i *#sClosing*, koje se koriste pri signalizaciji određenih alarma.

Varijabla *#M_BIT* uvodi se kako bi se softverski simuliralo tipkalo. Naime, prilikom fizičke implementacije, umjesto tipkala, tehničar može postaviti prekidač, koji može ostati u jednom stanju ili pak korisnik može držati tipkalo. Na taj se način osigurava da se gleda samo rastući brid signala koji dolazi s tipkala T1. Na analogan način napravljena je detekcija ruba za tipkalo T2, samo je korištena varijabla *#M_BIT2*. Kôd za otvaranje/zatvaranje vrata prikazan je na slici 3.7.

```
REGION _PUSH_BUTTON_EVENT
    // T1 or T2 is pushed
    IF (#T1 AND NOT (#M_BIT)) XOR (#S1 AND (#T2 AND NOT (#M_BIT2))) THEN

        //open
        IF #xPerm_opn THEN
            #sOpn_gate := TRUE;
        END_IF;

        //close
        IF #xPerm_cls THEN
            #sCls_gate := TRUE;
        END_IF;

        //resume opening/closing door after error
        IF (#sAlarm_MZS OR #sAlarm_K1K2 OR #sAlarm_KOKZ OR #sAlarm_stuck) = FALSE AND #KO = FALSE AND #KZ = FALSE THEN
            IF #sGare_dir THEN
                #sOpn_gate := TRUE;
            ELSE
                #sCls_gate := TRUE;
            END_IF;
        END_IF;

        // alarm shut down
        IF #sAlarm_MZS OR #sAlarm_K1K2 OR #sAlarm_KOKZ OR #sAlarm_stuck THEN
            #sAlarm_MZS := FALSE;
            #sAlarm_K1K2 := FALSE;
            #sAlarm_KOKZ := FALSE;
            #sAlarm_stuck := FALSE;
        END_IF;

        // set temp variables
        #sOpening := #sOpn_gate;
        #sClosing := #sCls_gate;

    END_IF;

    //set temp bit for edge detecting
    #M_BIT := #T1;
    #M_BIT2 := #T2;
END_REGION
```

Slika 3.7. *Kôd za obradu pritiska tipkala.*

Pamćenje posljednjeg smjera

Pamćenje smjera radi se provjerom varijable *#sCls_gate* ili *#Opn_gate*. Ako je *#Opn_gate* postavljena u logičku jedinicu, varijabla *#sGate_dir* postavlja se u *true*, što znači da je zadnji smjer

garažnih vrata bio otvaranje. U suprotnom se postavlja u *false*, što znači da je zadnji smjer garažnih vrata bio zatvaranje. Na slici 3.8. se vidi kôd zadužen za pamćenje posljednje smjera.

```
REGION _REMEMBER_LAST_DIRRECTION
    // Remember the last direction
    IF #sOpn_gate THEN
        #sGare_dir := TRUE; // last dirrection of the gate was 'open'
    ELSIF #sCls_gate THEN
        #sGare_dir := FALSE; // last dirrection of the gate was 'close'
    END_IF;
END_REGION
```

Slika 3.8. *Kôd za pamćenje posljednjeg smjera.*

Otvaranje/Zatvaranje garažnih vrata

Samo otvaranje ili zatvaranje vrata je vrlo jednostavan proces u kojem se koriste već spomenute varijable, koje su postavljene prilikom obrade događaja pritiska tipkala. Vrata završavaju s otvaranjem/zatvaranjem ako se varijable *#KO* odnosno *#KZ* postave u logičku jedinicu kao što je i prikazano na slici 3.9.

```
REGION _OPEN/CLOSE_GATE
    // Start opening
    #OTV := #sOpn_gate;

    // Start Closing
    #ZAT := #sCls_gate;


    // Stop the opening
    IF #KO THEN
        #sOpn_gate := FALSE;
        #sOpening := #sOpn_gate;
    END_IF;

    // Stop the closing
    IF #KZ THEN
        #sCls_gate := FALSE;
        #sClosing := #sCls_gate;
    END_IF;
END_REGION
```

Slika 3.9. *Kôd za otvaranja/zatvaranje garažnih vrata.*

Dojava alarma

Budući da postoje 4 različita alarma, postoje i 4 različita dijela kôda, koji obrađuju svaki alarm posebno. Za svaki postoji posebna zastavica, a budući da se program ciklički izvršava, i ako postoji samo jedna, pamti se ona koja je posljednja. *#sOpening* i *#sClosing* ovdje se koriste kao privremene varijable kako bi se u trenutku pojave alarma zaustavilo otvaranje ili zatvaranje garažnih vrata. Budući da za dva alarma postoji vremenski uvjet, postoje i dva brojača, dok se za 'ispad motorne zaštitne sklopke' i 'oba položaja vrata aktivna' može poslužiti IF uvjetom. Programski kôd za dojavu alarma prikazan je na slici 3.10.

```
(*
  alarm_MZS -> MZS error
  alarm_K0KZ -> Both KZ and K0 are active
  alarm_stuck -> If door didnt close or open within 30sec
  alarm_K1K2 -> Direction confirmation(K1 or K2) did not come within 5 sec of opening/closing gate
 *)

//Send alarm signal
IF #sAlarm_MZS OR #sAlarm_K1K2 OR #sAlarm_K0KZ OR #sAlarm_stuck THEN
    #sOpn_gate := FALSE;
    #sCls_gate := FALSE;
    #ALM := "Clock_0.5Hz";
ELSE
    #ALM := FALSE;
END_IF;

//Direction confirmation did not come within 5 sec
#IEC_Timer_5s_alarm(IN := ((#sOpening AND NOT #K1) OR (#sClosing AND NOT #K2)),
                    PT := #timer_5s,
                    Q => #sAlarm_K1K2,
                    ET => #xT);


//If door didnt close or open within 30sec
#IEC_Timer_30s_alarm(IN := #sOpening OR #sClosing,
                     PT := #timer_30s,
                     Q => #sAlarm_stuck,
                     ET => #xT);


//MZS error
IF #MZS THEN
    #sAlarm_MZS := TRUE;
END_IF;

//Both KZ and K0 are active
IF #KZ AND #K0 THEN
    #sAlarm_K0KZ := TRUE;
END_IF;
```

Slika 3.10. *Kôd za dojavu alarma.*

Prikaz svih korištenih varijabli, kao i njihov kratki opis, prikazan je na slici 3.11

Block_2_DB

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	Input									
2	T1	Bool	false		☑	☑	☑			Inside push button
3	T2	Bool	false		☑	☑	☑			Outside push button
4	S1	Bool	false		☑	☑	☑			Inside comb switch
5	KO	Bool	false		☑	☑	☑			Signal "doors are open"
6	KZ	Bool	false		☑	☑	☑			Signal "doors are closed"
7	K1	Bool	false		☑	☑	☑			Doors are opening
8	K2	Bool	false		☑	☑	☑			Doors are closing
9	MZS	Bool	false		☑	☑	☑			Motor protection switch
10	Output									
11	OTV	Bool	false		☑	☑	☑			Command to open the door
12	ZAT	Bool	false		☑	☑	☑			Command to close the door
13	ALM	Bool	false		☑	☑	☑			Alarm
14	InOut									
15	M_BIT2	Bool	false							temp bit for T2
16	M_BIT	Bool	false							temp bit for T1
17	Static									
18	sAlarm_MZS	Bool	false							Flag for MZS alarm
19	sAlarm_KOKZ	Bool	false							Flag for KO and KZ alarm
20	sAlarm_stuck	Bool	false							Flag for if gate didnt close/open
21	sAlarm_K1K2	Bool	false							Flag for K1 or K2 alarm
22	sCls_gate	Bool	false		☑		☑			Flag to close the door
23	sOpn_gate	Bool	false		☑		☑			Flag to open the door
24	sGate_dir	Bool	false							Remember the last dirrection of the gate
25	sOpening	Bool	false							temp var for opening gate
26	sClosing	Bool	false							temp var for closing gate
27	IEC_Timer_30s_alarm	TON_TIME			☑	☑	☑	☑		Timer for alarm_stuck
28	IEC_Timer_5s_alarm	TON_TIME			☑	☑	☑	☑		Timer for alarm_K1K2
29	IEC_Timer_2min_alarm	TON_TIME			☑	☑	☑	☑		Timer when door's are open

Slika 3.11. *Popis svih varijabli unutar podatkovnog bloka.*

3.2. Izrada sučelja čovjek-stroj

Sučelje čovjek-stroj zamišljeno je kao web stranica na kojoj se nalazi tipka za otvaranje ili zatvaranje garažnih vrata, vizualni prikaz stanja vrata i signalizacija alarmantnog stanja. Sučelje je izrađeno korištenjem Visual Studio Code razvojnog okruženja koristeći više programskih jezika za izradu web stranica.

3.2.1. Visual Studio Code

Visual Studio Code je besplatni editor programskog koda, koji je razvio Microsoft. Služi za razvoj web stranica, mobilnih programskih izvedbi, razvoj računalnih igara i sl. Nudi mogućnost otklanjanja pogrešaka (engl. *debugging*), refaktoriranje koda, inteligentno dovršavanje koda (engl. *IntelliSense*), korištenje git naredbi za verziranje iz samog editora, itd. Podržava različite programske jezike kao što su: Python, C3, C/C++, JavaScript, TypeScript, HTML/CSS, Java, PHP i mnogi drugi [11].

3.2.2. HTML

HTML (engl. *Hyper Text Markup Language*) predstavlja opisni jezik koji se koristi za izradu web stranica. Koristi se kako bi se opisao sadržaj i struktura web stranice, tj. kako bi sam preglednik (engl. *browser*) znao kako tu stranicu treba prikazati. HTML kao takav ne spada u niti jedan programski jezik jer se njime ne može izvršiti nikakva zadaća, već se koristi isključivo za određivanje strukture sadržaja i funkcije nekog HTML dokumenta [12]. Tablica 3.1. sadrži prikaz nekolicine HTML elemenata koji su korišteni u ovom radu.

Tablica 3.1. *Popis korištenih HTML elemenata [12].*

Element	Objašnjenje
`<div> … </div>`	Služi za grupiranje sadržaja.
`<p> … </p>`	Definira odlomak unutar HTML dokumenta.
`<h1> … </h1>` do `<h6> … </h6>`	Definira naslov . *h1* predstavlja prvu razinu, a *h6* posljednju.
`<img src=”put_do_slike.png”/>`	Služi za umetanje slike u HTML dokument.
`<form method=”” action=””> … </form>`	Definira web obrasac. Unutar method atributa se navodi *get*, koji šalje podatke putem URL-a (engl. *Uniform Resource Locator*), ili *post* koji šalje podatke unutar HTTP zahtjeva. *action* označava putanju do stranice koja obrađuje podatke.
`<input type=”” value=””>`	Definira dugme unutar obrasca pomoću kojeg se šalju podaci uneseni u obrazac.

Važna napomena je ta da se uz svaki HTML element može dodijeliti identifikacijska oznaka (engl. *Identification*, ID) ili oznaka klase (engl. *Class*). Identifikacijska oznaka je jedinstvena, dok klasa može označavati skupinu elemenata. Pogledom na sliku … vidi se primjer definiranja `div`, `p` i `img` elementa, svaki sa svojom identifikacijskom oznakom.

3.2.3. CSS

CSS (engl. *Cascading Style Sheet*) je jezik koji služi za oblikovanje web stranice. CSS je jezik koji se uvijek veže uz HTML. Ideja je razdvajanje prezentacijskog kôda u zasebnu datoteku te njegovo definiranje uz pomoć jednostavnih pravila. CSS se uobičajeno piše odvojeno od HTML kôda zbog preglednosti, ali se povezuju na način da se doda HTML element `<link>`. Sintaksa se

sastoji od selektora koji definira na koji se element, ili skupinu elemenata, utječe. Nakon selektora slijede uglate zagrade unutar kojih se upisuju deklaracije odvojene točka-zarezom, gdje svaka deklaracija ima svoje ime i pripadajuću vrijednost. UZ HTML oznake kao selektore mogu se imati klase i identifikacijske oznake, koje su definirane u HTML datoteci [14].

3.2.4. JS

JavaScript (JS) je jednostavan programski jezik više razine, koji je najviše namijenjen za razvoj interaktivnih web stranica, ali se može koristiti za izradu internetskih aplikacija, igara i još mnogo toga. JS kao takav je skriptni programski jezik jer se čitavi kôd prevodi u strojni jezik pomoću interpretera. JavaScript podržavaju svi moderni Internet preglednici te predstavlja jedan od najčešće korištenih jezika u izradi internetskog sadržaja [15].

Preporuča se pisati JavaScript kôd prema ASCII (engl. *American Standard Code for Information Interchange*) standardu iako je moguće pisati i prema UNICODE standardu. JavaScript razlikuje velika i mala slova što znači da se mora pripaziti prilikom pisanja ključnih riječi, varijabli ili funkcija kako bi se izbjegle pogreške. Kraj naredbe označava se isto kao u C programskom jeziku, a to je znak točka-zarez (;). Iako to nije obavezan znak, što znači da ga programer može, ali i ne mora staviti, JavaScript ubacuje točku-zarez na kraj svakog reda [14].

JavaScript skripta se može umetnuti bilo gdje na stranici, iako ju je najbolje umetnuti na samo dno dokumenta, prije zatvaranja `<\body>` tag-a. Postoje 4 načina na koje se skripta može uključiti u HTML dokument [14]:

- Korištenjem `<script> … </script>` oznaka,
- Uključivanjem vanjske datoteke u kojoj se kôd nalazi,
- Preko obrade događaja navedenog kao HTML atribut,
- Pisanjem kôda unutar URL-a.

3.2.5. Programsko rješenje sučelja čovjek-stroj

Sučelje čovjek-stroj realizirano je u obliku web stranice prikazane na slici 3.12. Na sučelju se nalazi tipka za otvaranje/zatvaranje garažnih vrata, signalizacije alarmnog stanja i slike koja pokazuje u kojem se stanju garažna vrata trenutno nalaze.

Slika 3.12. *Web sučelje čovjek-stroj.*

Prije svega potrebno je povezati varijable koje se nalaze na PLC uređaju s varijablama koje su korištene na samom sučelju. To se radi pomoću AWP (engl. *Automation Web Programming*) naredbi koje omogućuju definiranje sučelja između web stranice i podataka koji se nalaze na centralnoj procesorskoj jedinici samog PLC uređaja. AWP naredbe, prikazane u tablici 3.2., implementiraju se pomoću komentara unutar HTML datoteke, prikazanih na slici 3.13., a način korištenja je prikazan na slici 3.14. Pomoću njih je moguće [5]:

- Čitati PLC oznake,
- Pisati u PLC oznake,
- Čitati specijalne PLC oznake,
- Pisati u specijalne PLC oznake,
- Definirati tip podataka *enum,*
- Dodijeliti oznake tipovima podataka *enum,*
- Definirati fragmente podatkovnih blokova,
- Uvesti fragmente,
- Pristupanje vrijednostima polja (engl. *array*),
- Pristup vrijednostima PLC oznaka koji su STRUCT tipa podatka.

Tablica 3.2. *Popis AWP naredbi [5].*

Funkcija	**Značenje**
Čitanje PLC oznaka	`:=<Varname>:`
Pisanje u PLC oznake	`<!-- AWP_In_Variable Name='<Varname1>' -->`
Čitanje specijalne PLC oznake	`<!-- AWP_Out_Variable Name='<Typ>:<Name>' -->`
Pisanje u specijalne PLC oznake	`<!-- AWP_In_Variable Name='<Typ>:<Name>' -->`
Definiranje tipa podatka *enum*	`<!-- AWP_Enum_Ref Name='<Varname>' Enum='<Name Enum-Typ>' -->`
Dodjeljivanje tipa podatka *enum* oznakama	`<!-- AWP_Start_Fragment Name='<Name>' [Type=<Typ>] [ID=<Id>] -->`
Definiranje podatkovnog bloka fragmenata	`<!-- AWP_Import_Fragment Name='<Name>' -->`
Uvoz podatkovnog bloka fragmenata	`<!-- AWP_In_Variable Name='<Typ>:<Name>' -->`
Pristup vrijednostima polja (engl. a*rray)*	`<!-- AWP_Start_Array Name='"<DB name>".<array name>' --> ... <!-- AWP_End_Array -->`
Pristup vrijednostima PLC oznaka koje su STRUCT tipa podatka	`<!-- AWP_Start_Struct Name='"<DB name>".<struct name>' --> ... <!-- AWP_End_Struct -->`

```
<!-- AWP_In_Variable Name='"Block_2_DB".T1' -->
<!-- AWP_Out_Variable Name='"Block_2_DB".KZ' -->
<!-- AWP_Out_Variable Name='"Block_2_DB".KO' -->

<!-- AWP_Out_Variable Name='"Block_2_DB".sAlarm_MZS' -->
<!-- AWP_Out_Variable Name='"Block_2_DB".sAlarm_KOKZ' -->
<!-- AWP_Out_Variable Name='"Block_2_DB".sAlarm_stuck' -->
<!-- AWP_Out_Variable Name='"Block_2_DB".sAlarm_K1K2' -->
```

Slika 3.13. *Pisanje u PLC oznake pomoću AWP naredbi.*

```
var KZ =: = "Block_2_DB".KZ: ;
var KO =: = "Block_2_DB".KO: ;
var alarm =: = "Block_2_DB".sAlarm_K1K2: ||: = "Block_2_DB".sAlarm_MZS: ||: = "Block_2_DB".sAlarm_KOKZ: ||: = "Block_2_DB".sAlarm_stuck: ;
```

Slika 3.14. *Čitanje PLC oznaka pomoću AWP naredbi.*

Za slanje podataka prema PLC, odnosno ažuriranje oznaka, koristi se forma. Mogu se koristiti HTTP POST ili GET metode, no ako se osvježi stranica, podaci se ponovno šalju i to može prouzročiti nepoželjno ponašanje sustava u vidu dojave nepostojećeg alarma. Iz navedenih razloga koristi se *.submit()* metoda.

```
<form method="POST" id="formT1">
    <input type="button" value="T1" id="btnOpen" onclick="submitForm()" />
    <input type="hidden" name='"Block_2_DB".T1' value="1" />
</form>
```

Slika 3.15. *Forma za slanje podataka.*

```
//submitting form
function submitForm() {
    var formOpn = document.getElementById("formT1");
    formOpn.submit();
}
```

Slika 3.16. *Funkcija za slanje forme.*

Pogledom na sliku 3.15. vidi se jedna forma s dva *input* elementa, *button input* i jedan *input* element koji je skriven. Klikom na tipku odlazi se do funkcije *submitForm()*, prikazana na slici

3.16., koja šalje formu serveru, odnosno šalje logičku jedinicu PLC-u, koji će to interpretirati kao pritisak tipke T1.

Sljedeća stavka web stranice je način signalizacije alarma. Postavljen je jedan *div* element koji, ako dođe do alarma, počinje blinkati crveno. To je napravljeno pomoću CSS *animation* svojstva u kombinaciji s *@keyframes* pravilom prikazanima na slici 3.17.

```
.blink_me {
    animation: blinker 1s linear infinite;
}

@keyframes blinker {
    50% {
        opacity: 0;
    }
}
```

Slika 3.17. *CSS kôd za animaciju signalizacije alarma.*

Prilikom učitavanja stranice pročitaju se sve PLC alarmne oznake (*sAlarm_K1K2, sAlarm_MZS, sAlarm_KOKZ, sAlarm_stuck*) te se napravi logička ILI operacija kako bi se utvrdilo je li trenutno stanje alarmantno ili nije. Nakon toga provjerava se ako je varijabla *alarm* u logičnoj jedinici (*true*) *div* elementu, u kojem slučaju dodaje se klasa „*blink_me*“ zbog koje on počinje blinkati. Na slici 3.18. vidi se programski kôd koji je zadužen za provjeru alarmantnog stanja.

```
if (alarm == true) {
    alarmDiv.style.background = "red";
    alarmDiv.classList.add("blink_me");
} else if (alarm == false) {
    alarmDiv.style.background = "gray";
    alarmDiv.classList.remove("blink_me");
}
```

Slika 3.18. *JS kôd signalizaciju alarmantnog stanja.*

Na sučelju postoji još i slika koja prikazuje u kojem se stanju garažna vrata trenutno nalaze. Postoje tri slike, za otvaranje, zatvaranje i međupoložaj. Odabir koja se slika prikazuje ovisi o stanju *KZ* i *KO* varijabli, koje se isto kao i varijabla *alarm* pročitaju s podatkovnog bloka PLC-a prilikom učitavanja stranice. Slika 3.19. prikazuje kôd koji mijenja slike na sučelju čovjek-stroj.

```
if (KZ == 1) {
    doorStatus.innerHTML = "Garage door: closed"
    imgClosedDoor.style.display = 'block';
} else if (KO == 1) {
    doorStatus.innerHTML = "Garage door: open"
    imgOpenDoor.style.display = 'block';
} else if (KZ || KO == 0) {
    doorStatus.innerHTML = "Garage door: opening/closing"
    imgPartiallyOpenDoor.style.display = 'block';
}
```

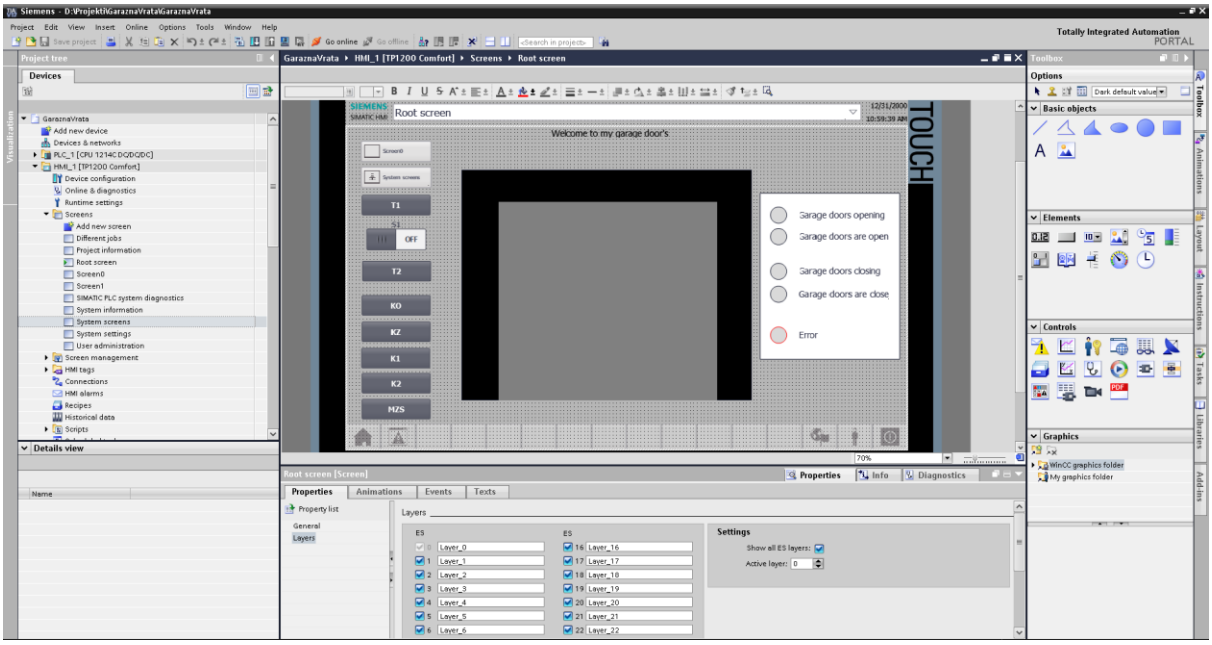
Slika 3.19. *JS kôd za prikaz stanja garažnih vrata.*

4. PUŠTANJE U POGON I TESTIRANJE UPRAVLJAČKOG PROGRAMA

Nakon završenog softverskog dijela sustava potrebno ga je testirati. Provedeno je ručno testiranje svakog dijela sustava. Za upravljački kôd izrađeno je HMI sučelje unutar TIA Portala koji se koristi kako bi se dobio vizualni prikaz varijabli. Potom je sučelje čovjek-stroj postavljeno na web server, napravljena je konfiguracija korisnika te je testirano samo sučelje prateći varijable programskog bloka unutar TIA Portala.

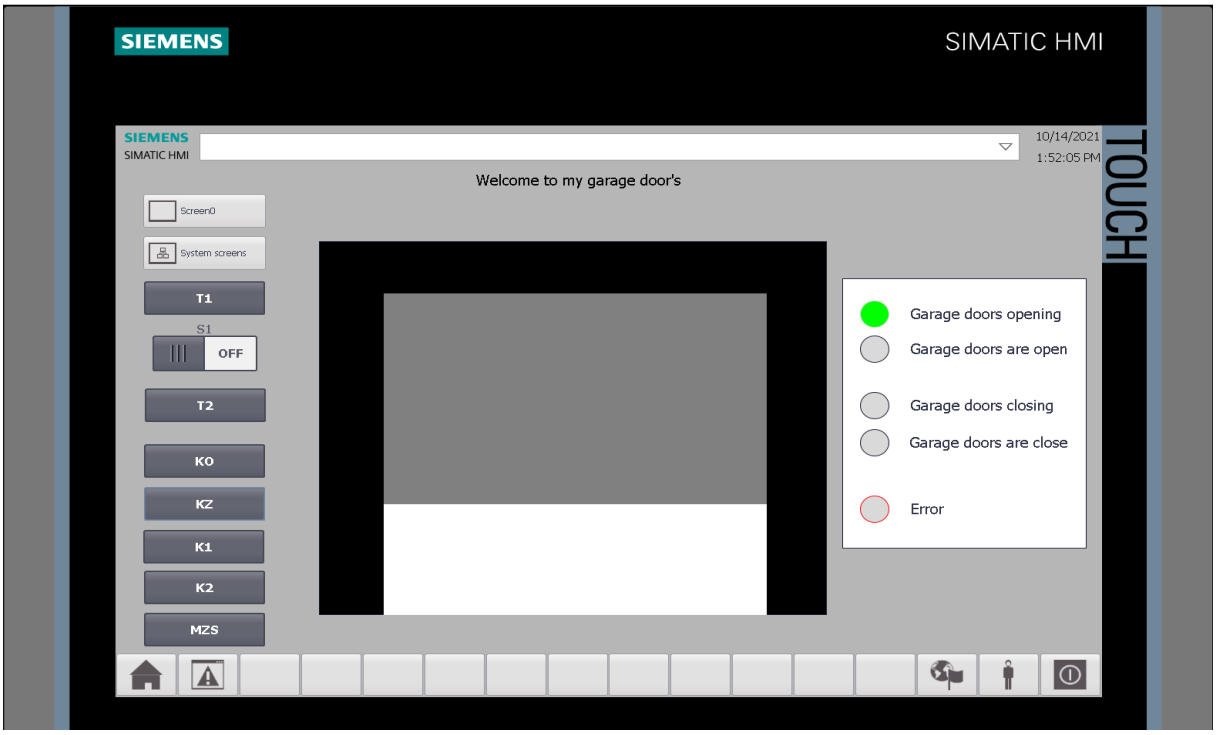
4.1. Testiranje upravljačkog kôda

Za potrebe testiranja upravljačkog kôda izrađen je HMI unutar programa TIA Portal koristeći TP1200 Comfort panel, prikazan na slici 4.1. Na njemu se nalaze svi ulazni signali, signalizacija za zatvaranje/otvaranje vrata i signalizacija alarmantnog stanja. Tim panelom moguće je simulirati ulazne signale te na taj način vidjeti na koji će način PLC obraditi ulazni signal. Tako je omogućeno lakše i jednostavnije testiranje samog kôda. Svaka tipka na panelu nosi naziv ulaznog signala kojeg predstavlja, a na sredini se nalazi grafički prikaz stanja vrata.

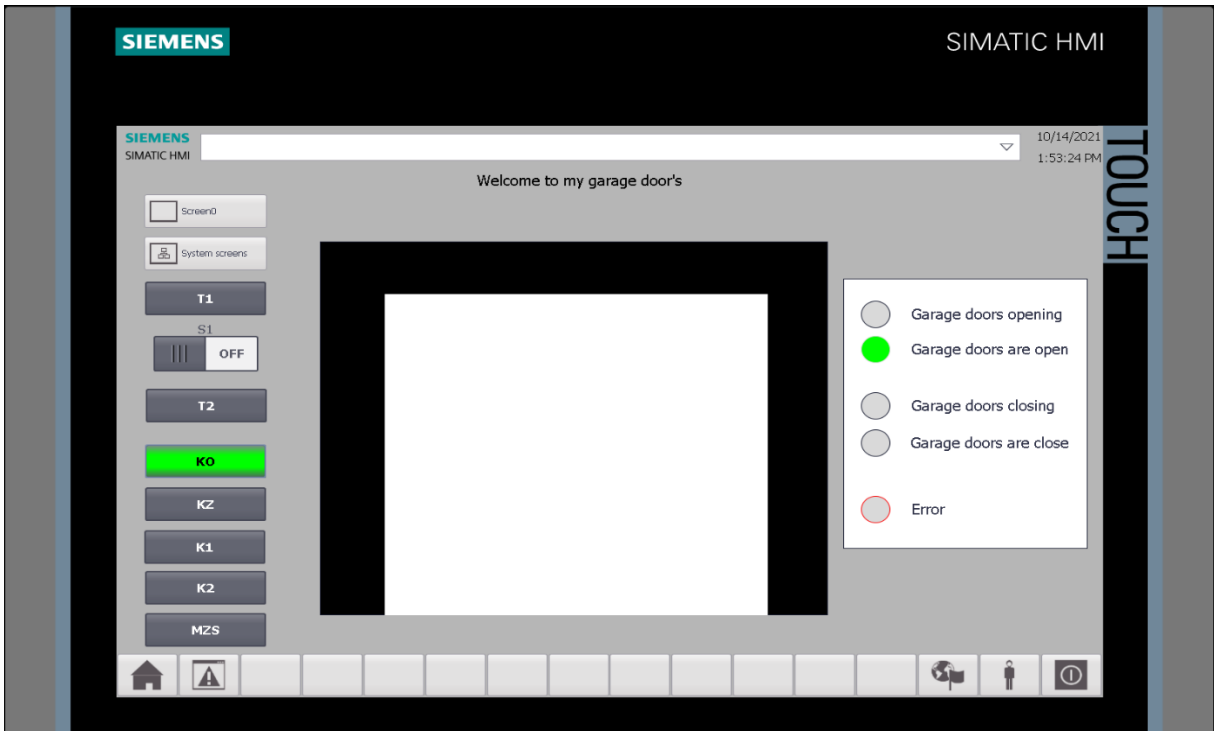


Slika 4.1.: *Prikaz HMI panela unutar TIA Portala.*

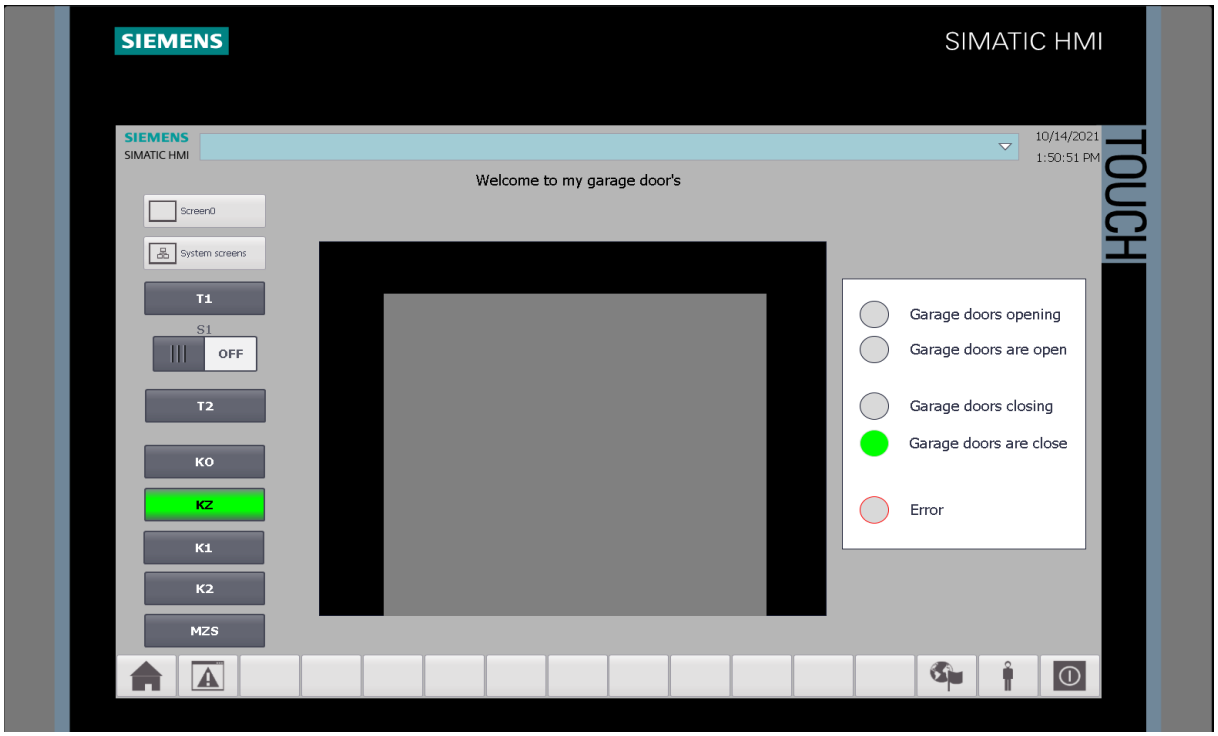
Klikom na tipku *T1* ili *T2*, ako je uključena tipka *S1*, vrata kreću u otvaranje ili zatvaranje što se i signalizira na desnoj strani panela zelenom bojom pored odgovarajuće poruke. Zatvaranje vrata omogućeno je na identičan način samo će se signalizirati druga poruka na desnoj strani panela. Slika 4.2. prikazuje HMI panel koji prikazuje garažna vrata u stanju otvaranja, na slici 4.3. moguće je vidjeti izgled panela ako su vrata otvorena dok je na slici 4.4 prikaz panela ako postoji alarm.



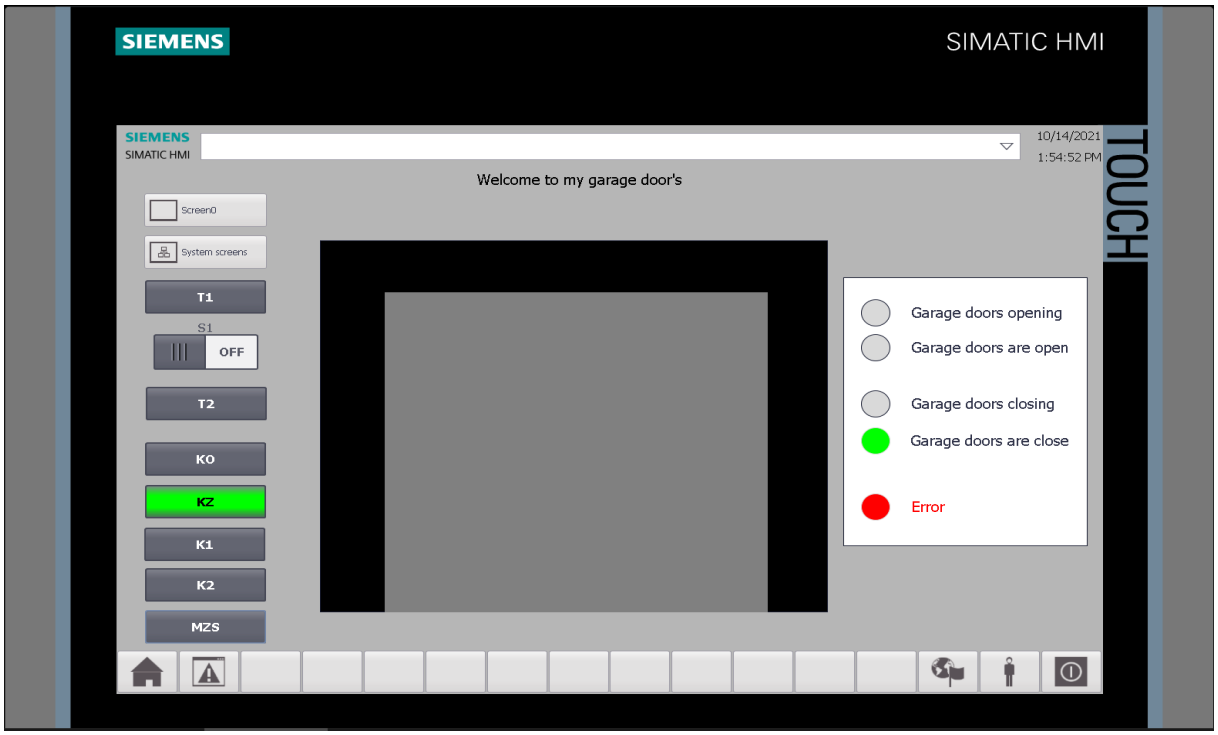
Slika 4.2. *Prikaz HMI panela prilikom otvaranja garažnih vrata.*



Slika 4.3. *Prikaz HMI panela prilikom otvorenih garažnih vrata.*



Slika 4.4. *Prikaz HMI panela prilikom zatvorenih garažnih vrata.*

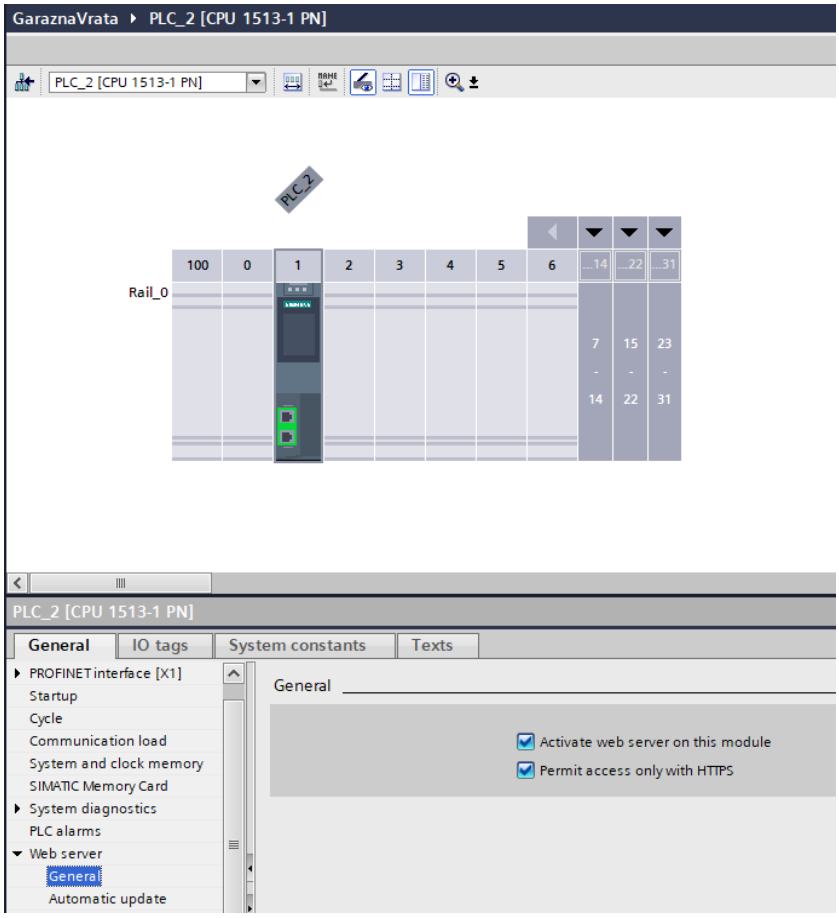


Slika 4.5. *Prikaz HMI panela prilikom alarmantnog stanja.*

Nedostatak takvog testiranja kôda je što se pravovremeno moraju upaliti određeni signali, npr. unutar 5 sekundi od početka zatvaranja/otvaranja vrata potrebno je kliknuti na tipku *K1* ili *K2* kako se alarm ne bi oglasio. Potrebna je velika koncentracija osobe koja vrši testiranje kako bi se zadavali signali iz stvarnog svijeta koji će na vjeran način oponašati željeno ponašanje sustava.

4.2. Konfiguriranje web servera i korisnika

Prije svega potrebno je konfigurirati web server na S7-1513 PN PLC-u. Potrebno je otići na PLC uređaj unutar TIA Portal programa te pod opcijom *Web server* aktivirati web server na tom modulu [6], kao što je prikazano na slici 4.6.



Slika 4.6. *Konfiguracija web servera.*

Nakon toga potrebno je dodijeliti korisnike, kao što je prikazano na slici 4.7. To je moguće unutar opcije *User management* gdje je svakom pojedinom korisniku potrebno dodijeliti korisničko ime, lozinku i prava pristupa. Konkretno unutar ovog sustava konfigurirana su tri korisnika, svaki s različitim pravima pristupa:

- Administrator

  Korisničko ime: admin

  Lozinka: Admin1234

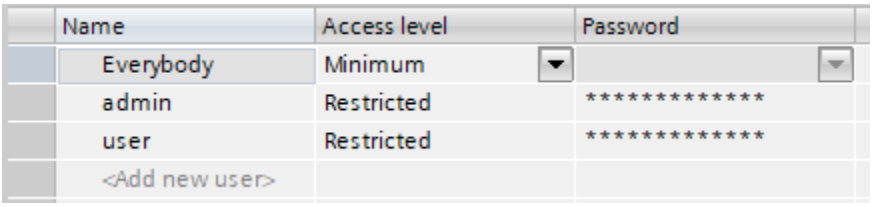
- Korisnik

  Korisničko ime: user

  Lozinka: 31000

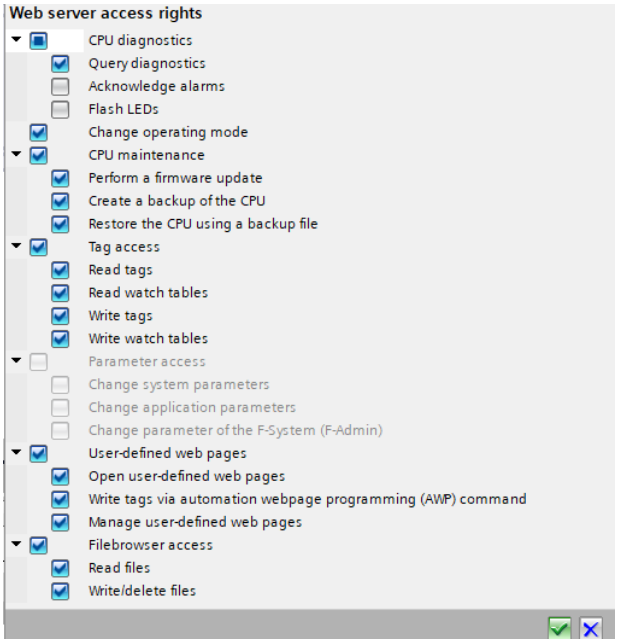
- Svatko (engl. *Everybody*)

Nije potrebna prijava s korisničkim imenom i lozinkom



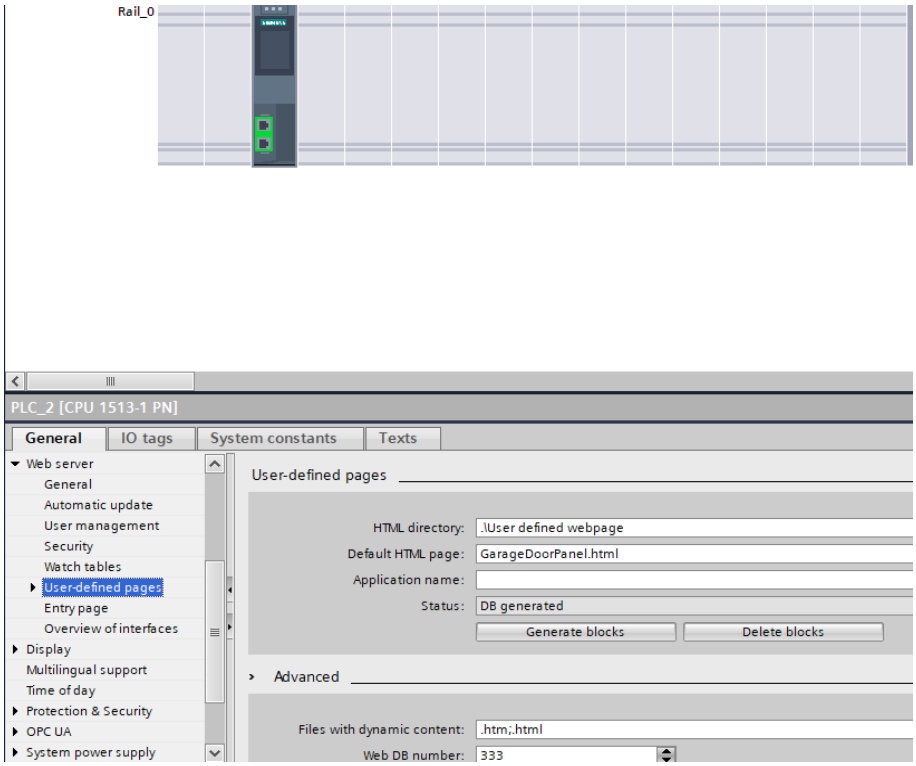
Slika 4.7. *Konfigurirani korisnici.*

Najveća prava pristupa ima administrator, koji je u pravilu zadužen za ažuriranje sustava, održavanje PLC-a i sl. Njegova prava pristupa mogu se vidjeti na slici 4.8.

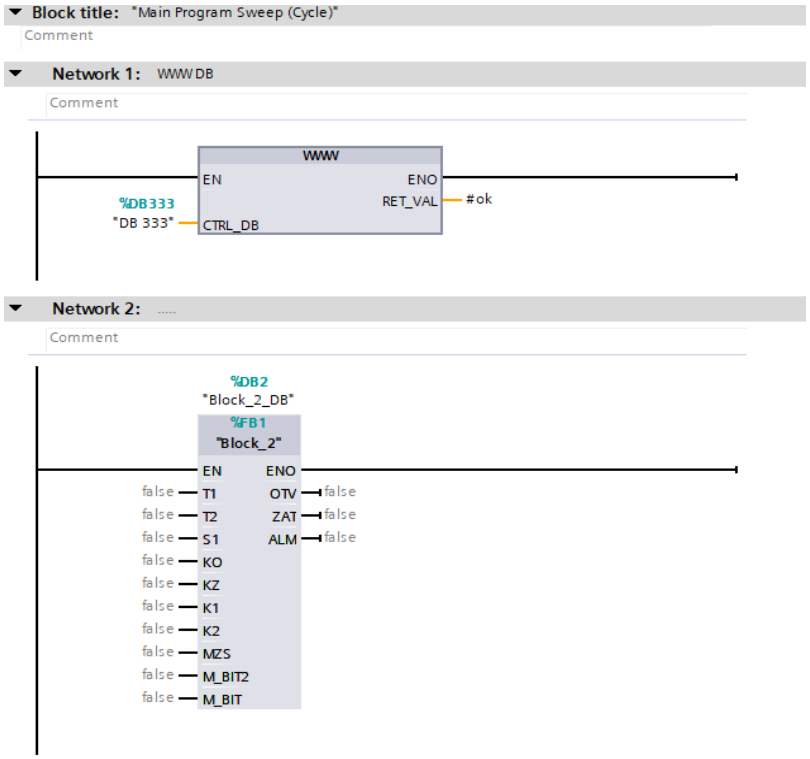


Slika 4.8. *Administratorska prava pristupa.*

Kako bi se web sučelje moglo testirati, potrebno ga je postaviti na web server PLC-a. To je moguće napraviti u TIA Portalu odlaskom na opciju *User-defined pages*, gdje je potrebno odabrati putanju do web sučelja kao i samo web sučelje te kliknuti na tipku *Generate blocks* kao što je prikazano na slici 4.10. Nakon toga generira se podatkovni blok (DB 333), koji sadrži sve varijable potrebe za komunikaciju s web sučeljem. Taj blok potrebno je postaviti na ulaz WWW podatkovnog bloka, koji opisuje web sučelje, unutar glavnog organizacijskog bloka *Main[OB1]*. WWW podatkovni blok prikazan je na slici 4.11.



Slika 4.10. *Postavljanje web sučelja na web server.*

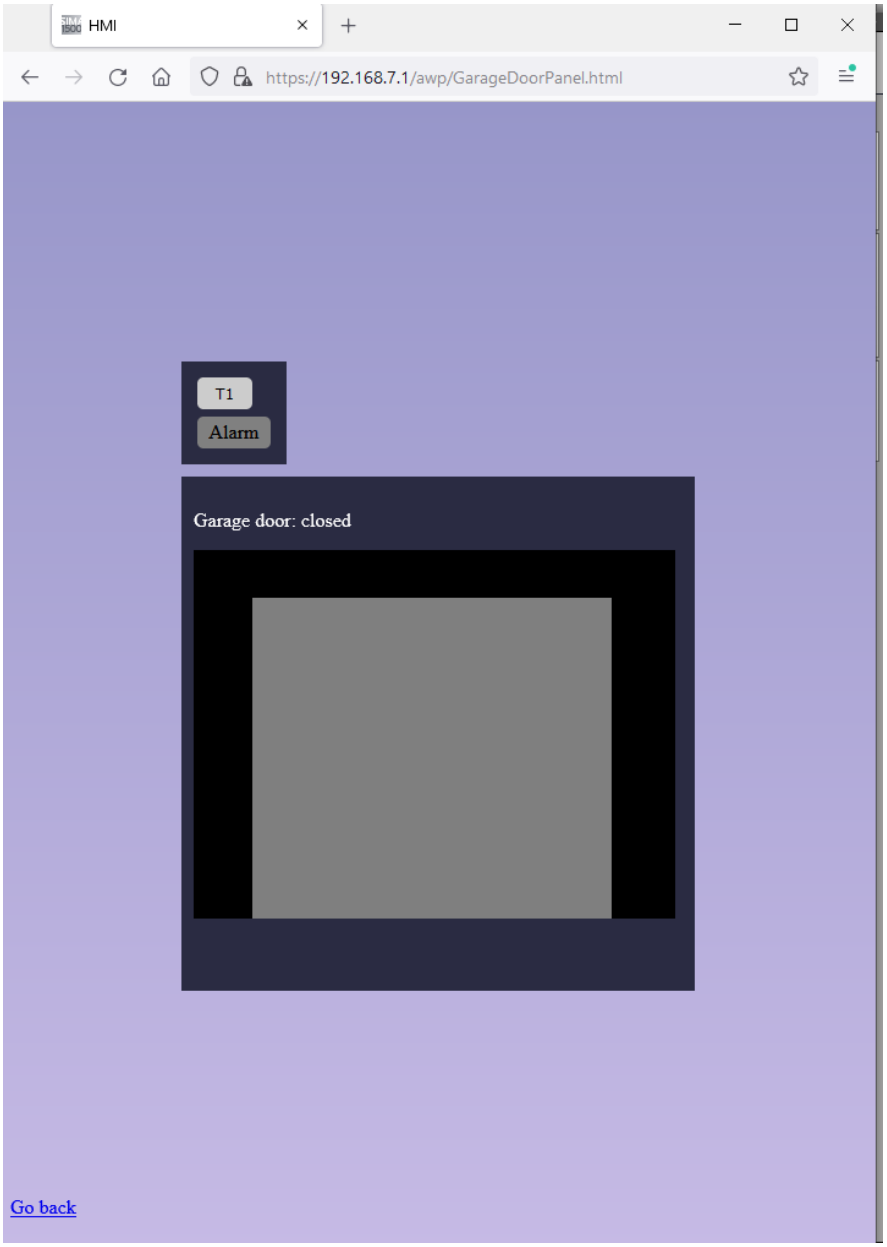


Slika 4.11. *Main[OB1] nakon postavljenog WWW bloka.*

Sučelje je sada moguće otvoriti odlaskom na početnu stranicu web servera. U preglednik je potrebno upisati IP adresu web servera, u ovom slučaju to je 192.168.7.1, kao što je prikazano na slici 4.12.

Slika 4.12. *Početna stranica web servera.*

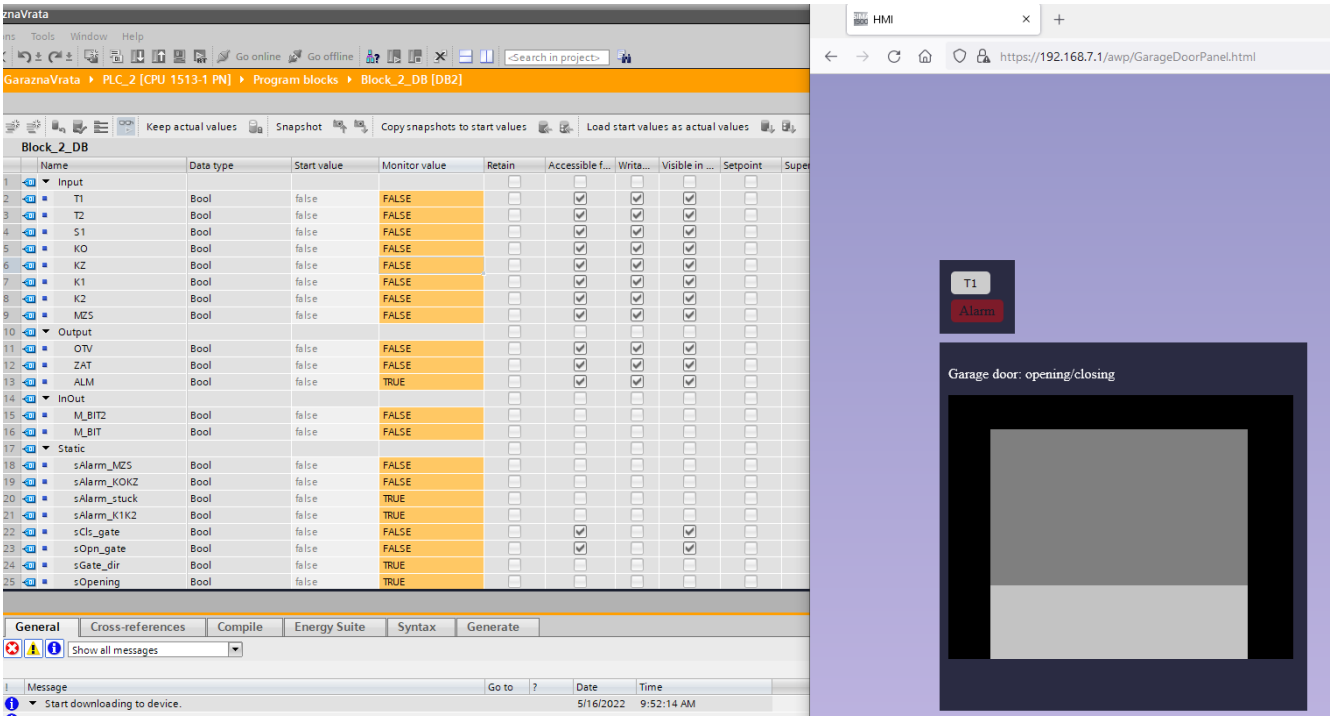
Nakon klika na tipku *ENTER* moguće je prijaviti se nekim od korisničkih računa te klikom na tipku *User-defined pages* pokrenuti web sučelje, koje je prikazano na slici 4.13.



Slika 4.13. *Web sučelje čovjek-stroj.*

4.3. Testiranje web sučelja

Klikom na tipku *T1* vrata prelaze u otvaranje ili zatvaranje, što je na web sučelju prikazano odgovarajućim tekstom i slikom. Ako se pojavi alarm, to će se i signalizirati, kao što je prikazano na slici 4.14. gdje se vidi signalizacija alarma unutar TIA Portala i na samome web sučelju.



Slika 4.14.: *Signalizacije alarma na web sučelje čovjek-stroj.*

5. KIBERNETIČKA SIGURNOST SUSTAVA

Kibernetička sigurnost sustava prvo je provjerena pomoću *Siemens Security Advisories*-a [18], kojeg je moguće preuzeti sa službenih internetskih stranica proizvođačke opreme (Siemens). U sklopu toga nalaze se svi otkriveni i riješeni sigurnosni problemi koji su vezani za sve sigurnosne ranjivosti na svim Siemens proizvodima. U okviru ovog rada obrađene su i predložene mjere zaštite za one rizike koji su povezani s ugrađenim web poslužiteljem PLC-ova iz serije S7-1500 i web server.

Svaka ranjivost rangirana je pomoću CVSS 3.1 verzije (engl. *Common Vulnerability Scoring System version 3.1*), koja predstavlja jednu od najraširenijih normi za rangiranje težine ranjivosti. Rezultat CVSS ocjenjivanje je brojevima od 0 do 10, koji označavaju ozbiljnost ranjivosti, a prikazani su u tablici 5.1.

Tablica 5.1. *Kvalitativna procjena rizika [17].*

Procjena	**CVSS rezultat**
Nema rizika	0.0
Niski rizik	0.1 – 3.9
Srednji rizik	4.0 – 6.9
Visoki rizik	7.0 – 8.9
Kritičan rizik	9.0 – 10.0

5.1. SQL ubrizgavanje

Ranjivost koja nosi oznaku SSA-604937 [20] odnosi se na više sigurnosnih ranjivosti, koje su vezane za bilo koji web server, a primjer je uzet za MES sustav (Sustav izvršenja proizvodnje) koji predstavlja softverski sustav za upravljanje radnim procesima unutar industrijskih postrojenja [29]. Jedna od njih, koja nosi oznaku CWE-89, odnosi se na neispravnu neutralizaciju specijalnih oznaka u SQL naredbi. Nosi CVSS rang od 8.1 i visoku vjerojatnost iskorištavanja.

Ako softver prilikom korisničkog unosa podataka ne odstrani specijalne oznake, npr. SQL sintakse ili jednostruke navodnike, napadaču omogućava konstrukciju SQL upita, koji mogu prouzročiti manipulaciju podacima unutar baze podataka. SQL ubrizgavanje (engl. *SQL injection*)

postalo je uobičajeni problem kod web stranica koje koriste baze podataka. Tu je ranjivost vrlo lako uočiti i počiva na činjenici da SQL jezik (engl. *Structured Query Language*) ne razlikuje kontrolnu i podatkovnu ravninu.

Kôd prikazan na slici 5.1. napisan je u C# jeziku koji dinamički stvara i izvršava SQL upit koji pretražuje bazu podataka dok ne nađe podatak koji pripada prijavljenom korisniku i koji nosi specifičan naziv (u kôdu označeno s *itemname*).

Example Language: **C#** (bad code)

```
...
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE owner = '" + userName + "' AND itemname = '" + ItemName.Text
+ "'";
sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
sda.Fill(dt);
...
```

Slika 5.1. *Prikaz kôda koji omogućava SQL ubrizgavanje [21].*

Ako korisnik, npr. s korisničkim imenom wiley, unese znakovni niz prikazan na slici 5.2., SQL upit dobiva oblik prikazan na slici 5.3. što se zapravo logički svodi na upit prikazan na slici 5.4.

```
name' OR 'a'='a
```

Slika 5.2. *Prikaz zlonamjernog kôda [21].*

```
SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name' OR 'a'='a';
```

Slika 5.3. *Prikaz regulirajućeg SQL upita [21].*

```
SELECT * FROM items;
```

Slika 5.4. *Prikaz logičkog ekvivalenta napadačkog SQL upita [21].*

Može se vidjeti kako SQL upit na slici 5.4. vraća sve podatke iz baze podataka, neovisno o tome koji korisnik „posjeduje“ taj podatak. Taj se problem može riješiti tako da se korisnikov unos

(u kôdu označeno s *itemname*) filtrira izbacivanjem svih nedozvoljeni znakova, kao što su crtice, navodnici, razmaci i slično.

Siemens preporučuje korištenje vatrozida (engl. *firewall*), koji filtrira podatkovni promet koji sadrži SQL ubrizgavanje [20].

5.2. Nepravilna kontrola pristupa

Sljedeća u nizu ranjivosti koja se može povezati s S7-1500 web serverom je ona pod oznakom SSA-350757 [22], unutar koje napadač dobiva neovlašteni pristup uređajima koji su konfigurirani unutar TIA Portala zbog pogrešne konfiguracije korisnika (napadač mora imati pristup web serveru). Ranjivost se klasificira pod oznakom CWE-284 i nosi CVSS rang od 6.4.

Kontrola pristupa uključuje nekoliko vrsti zaštite [23]:

- Provjeru autentičnosti (provjera je li korisnik zaista onaj koji tvrdi da jest),
- Autorizacija (provjera ima li taj korisnik doista pristup tom resursu/funkcionalnosti),
- Odgovornost (praćenje aktivnosti koje su napravljene).

Ako jedan od navedenih mehanizama nije pravilno primijenjen ili na bilo koji drugi način popusti, napadač može to iskoristiti kako bi dobio ovlašteni pristup informacijama, funkcijama i sl .

Postoje dva jasna tipa ponašanja koja mogu dovesti do nepravilne kontrole pristupa [23]:

- Specifikacija - Neispravna specifikacija prava korisnika koju provodi administrator ili sam program.
- Provedba - Mehanizam/program sadrži pogrešku koja sprječava ispravnu provedbu točno specificirane kontrole pristupa.

Na primjeru prikazanom na slici 5.5., koji je pisan PHP programskim jezikom, vidi se jedan primjer na koji napadač može zadobiti pristup neovlaštenim informacijama.

Example Language: **PHP** *(bad code)*

```
function runEmployeeQuery($dbName, $name){
  mysql_select_db($dbName,$globalDbHandle) or die("Could not open Database".$dbName);
  //Use a prepared statement to avoid CWE-89
  $preparedStatement = $globalDbHandle->prepare('SELECT * FROM employees WHERE name = :name');
  $preparedStatement->execute(array(':name' => $name));
  return $preparedStatement->fetchAll();
}
/.../

$employeeRecord = runEmployeeQuery('EmployeeDB',$_GET['EmployeeName']);
```

Slika 5.5. *Prikaz kôda koji omogućava neispravnu autorizaciju [24].*

Kôd dohvaća sve podatke iz baze podataka koji pripadaju korisniku. Pomoću *prepare* naredbe izbjegava se SQL ubrizgavanje, ali nigdje se ne provjerava može li doista taj korisnik dohvatiti podatke iz baze podataka. Upravo taj aspekt predstavlja sigurnosni rizik budući da napadač može potencijalno dohvatiti podatke na koja nema pravo.

Siemens predlaže ažuriranje TIA Portala na noviju verziju, dok za TIA Portal V15 još ne postoji ni jedno rješenje. Nadalje predlaže da se prilikom svake promjene, koja je vezana za autentifikaciju korisnika, pokuša direktno pristupiti web serveru pomoću neovlaštenog pristupa (korisnički račun *Everybody*). Ako je to moguće, potrebno je napraviti ponovnu konfiguraciju koristeći ažuriranu verziju TIA Portala. Ako novija verzija nije dostupna, ažuriranje korisnika i njihovih prava pristupa neće biti od pomoći. Potrebno je obrisati PLC iz TIA Portala te ga ponovno konfigurirati s novim projektom [23].

5.3. Čitanje izvan granica

Ranjivost pod oznakom SSA-480230 [25] može dozvoliti neautoriziranom napadaču, koji ima pristup web serveru, mogućnost DoS napada (engl. *Denial of Service attack*). DoS napad potencijalno može dovesti do ponovnog pokretanja web servera, čime bi on postao nedostupan korisniku. Za uspješno obavljanje takvog napada nije potreban privilegirani pristup niti ikakva interakcija s korisnikom. Takva nesigurnost nosi 7.5 CVSS rang.

Sljedeći primjer, prikazan na slici 5.6., temelji se na čitanju podataka izvan ili prije početka međuspremnika (engl. *buffer*). To omogućava čitanje nedozvoljenih i/ili osjetljivih podataka ili prestanak rada sustava.

Example Language: **C** (bad code)

```
int getValueFromArray(int *array, int len, int index) {

  int value;

  // check that the array index is less than the maximum

  // length of the array
  if (index < len) {

    // get the value at the specified index of the array
    value = array[index];
  }
  // if array index is invalid then output error message

  // and return value indicating error
  else {
    printf("Value is: %d\n", array[index]);
    value = -1;
  }

  return value;
}
```

Slika 5.6. *Prikaz lošeg kôda koji omogućava čitanje izvan granica [26].*

Metoda *getValueFromArray()* dohvaća podatak iz polja na specifičnom mjestu koji se daje kao parametar metodi (*index*). Nedostatak te metode je taj što samo provjerava je li dobiveni parametar (*index*) veći od duljine niza (*len*), ali ne provjerava minimalnu vrijednost. To omogućava unos negativne vrijednosti koja biva prihvaćena i rezultira čitanjem izvan granica (engl. *out of bounds read*), što može dovesti do čitanja nedozvoljenih podataka. Potrebno je modificirati kôd na način koji je prikazan na slici 5.7.

Example Language: **C**

```
...

// check that the array index is within the correct

// range of values for the array
if (index >= 0 && index < len) {

...
```

Slika 5.7. *Ispravno napisan kôd koji onemogućava čitanje izvan granica [26].*

Siemens predlaže sljedeće generalne sigurnosne korake kako bi se smanjili rizici koji prolaze iz ove ranjivosti[25]:

- Ažurirati verziju softvera na V2.6.1 (ako je riječ o S7-1500 CPU),
- Ograničiti pristup integriranom web serveru,
- Deaktivirati web server ako nije potreban,
- Implementirati strategije i mjere za ublažavanje koje su opisane u *Operational Guidelines for Industrial Security* [15], implementaciju vatrozida i VPN mreža (engl. *Virtual Private Network*), korištenje samo komunikacijskih protokola koji su bazirani na TLS (engl. *Transport Layer Security*) protokolu, autentifikaciju uređaja putem lozinki i certifikata itd.

5.4. Neispravna autorizacija

Ranjivost pod oznakom SSA-865327 [27] nadovezuje se na nepravilnu kontrolu pristupa, koja je već obrađena u ovom radu. Nosi CVSS rang od 5.3 i omogućuje neautoriziranom napadaču da pročita vrijednosti PLC oznaka.

Iako softver može provoditi provjeru autentičnosti, to ne znači da on to radi ispravno te napadač svejedno može zadobiti pristup informacijama na koje nema pravo. To može prouzročiti različite sigurnosne rizike kao što su DoS uvjeti ili napadi, izvršavanje proizvoljnog kôda, curenje informacija i sl.

Kao i kod svih sigurnosnih rizika, u slučaju eksploatacije ove ranjivosti mogu biti ugrožene sljedeće vitalne karakteristike sustava [28]:

- Povjerljivost (engl. *Confidentiality*) - napadač može pročitati osjetljive informacije čitajući izravno iz baze podataka ili koristeći funkcije kojima ima pristup kako bi pročitao podatke.
- Integritet (engl. *Integrity*) – napadač može modificirati podatke pišući direktno u bazu podataka ili koristeći funkcije kojima ima pristup kako bi pročitao podatke.
- Kontrola pristupa (engl. *Access Control*) - napadač može zadobiti privilegirani pristup modificirajući ili čitajući podatke iz baze podataka ili koristeći funkcije kojima ima pristup kako bi zadobio privilegirani pristup.

Primjer na slici 5.8. prikazuje na koji je način moguće iskoristiti neispravnu autorizaciju. Provjera ovlasti korisnika se vrši pomoću *role* varijable. Programer očekuje da će se kolačić (engl. *cookie*) postaviti samo kada se funkcija *getRole()* izvrši. Čak je implementiran i kolačić s vremenskim istekom nakon dva sata.

Example Language: **PHP** *(bad code)*

```
$role = $_COOKIES['role'];
if (!$role) {
  $role = getRole('user');
  if ($role) {
    // save the cookie to send out in future responses
    setcookie("role", $role, time()+60*60*2);
  }
  else{
    ShowLoginScreen();
    die("\n");
  }
}
if ($role == 'Reader') {
  DisplayMedicalHistory($_POST['patient_ID']);
}
else{
  die("You are not Authorized to view this record\n");
}
```

Slika 5.8. *Prikaz kôda koji neispravno vrši autorizaciju [28].*

Problem je što ništa ne zaustavlja napadača da *role* kolačić postavi na vrijednosti *Reader* što bi značilo da se funkcija *getRole()* nikada neće ni izvršiti, što znači da je neautorizirani korisnik upravo dobio pravo pristup informacijama.

Rješenje tog primjera je ili ne koristiti kolačiće ili ih svaki puta postaviti pomoću funkcije *getRole()*. Nadalje moguće je poboljšati sigurnost koristeći osnovne (engl. *basic*) ili *digest* autentifikacije metode za PHP programski jezik, koje se nalaze unutar *$_SERVER* polja podataka.

Jedini prijedlog Siemensa za S7-1500 PLC uređaje je ažuriranje na verziju V2.9.2 ili noviju, kao i osnovne sigurnosne mjere koje su već navedene u potpoglavlju 5.3 [27].

5.5. Hydra

Kibernetička sigurnost implementiranog sustava dodatno je provjerena pomoću Hydra alata. Hydra je brzi i fleksibilni alat za penetracijsko testiranje koji *brute-force*-anjem korisničkog imena

i/ili lozinke pokušava doći do pristupa određenoj web stranici. Hydra podržava protokole kao što su AFP (engl. *Apple Filing Protocol),* HTTP-FORM-GET, HTTP-FET, HTTP-FORM-POST, HTTP-PROXY i mnoge druge [15].

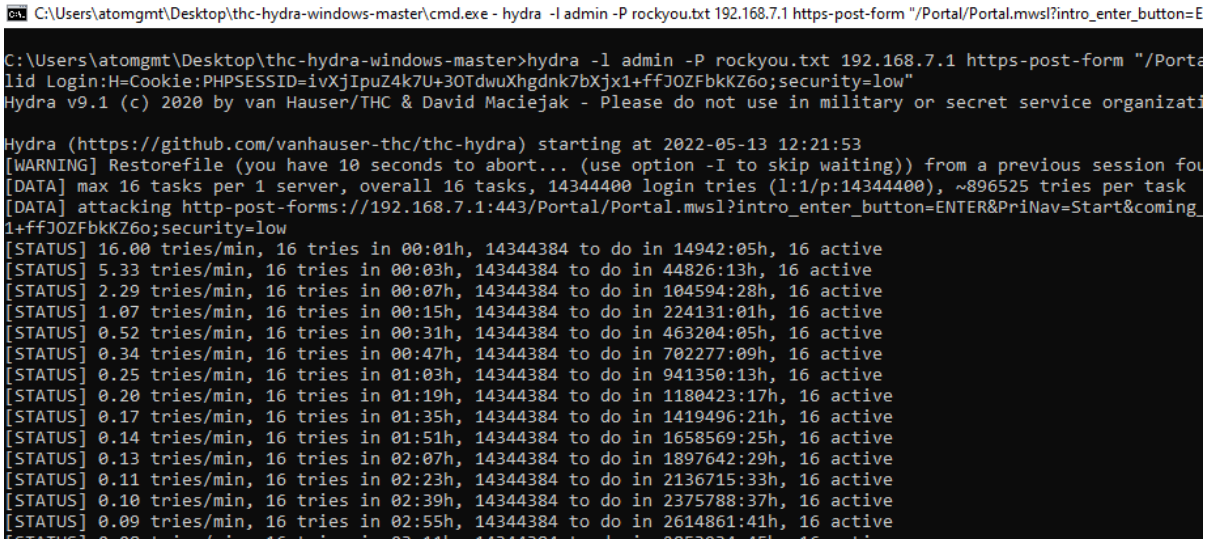
Da bi Hydra mogla pokušati dobiti pristup web stranici, potrebno je koristiti listu koja sadrži popis korisničkih imena i lozniki. U tom slučaju korištena je *rockyou.txt* datoteka [16] koja sadrži oko 32 milijuna lozinki. Ta datoteka dolazi od organizacije koja nosi isti naziv, iz koje je nepoznata grupa hakera (engl. *hackers*) 2009. godine uspjela ukrasti veliki popis svih korisničkih imena i lozinki.

Hydra se pokreće koristeći terminal unutar kojeg je potrebno definirati parametre poput [15]:

- Korisničkog imena i lozinke, ako su poznati, u suprotnom potrebno je koristiti .txt datoteku, unutar koje se nalazi popis korisničkih imena i lozinki, koja se koristi prilikom napada,
- IP adrese koja se napada,
- Tip protokola koji se koristi.

Osim toga moguće je i definirati ostale stvari kao npr. željeni simultani broj priključaka koji postoje na IP adresi (prema zadanim postavkama je 16), vrijeme čekanja na odgovor (engl. *response*) i sl.

Naredba koja je korištena u ovom primjeru je: *hydra -l user -P rockyou.txt 192.168.7.1 https-post-form "/Portal/Portal.mwsl?intro_enter_button=ENTER&PriNav=Start&coming_from_intro=true:Redirection=&Login=user&Password=^PASS^:F=Invalid Login:H=Cookie:PHPSESSID=5knVK1zc5UXxvfdcFgNSeNtJF+hM633RuXvSLbmQcbxZ;security=low"* i prikazana je na slici 5.9.



Slika 5.9. *Prikaz hydra brute force-anja.*

Objašnjenje Hydra naredbe [15]:

- *-l user* govori Hydri da je korisničko ime s kojim se treba ulogirati *user,*
- *-P rockyou.txt* govori da Hydra treba koristiti popis lozinki koje se nalaze u datoteci rockyou.txt,
- 192.168.7.1. predstavlja IP adresu koju se napada,
- *https-post-form* predstavlja protokol koji se napada,
- *"/Portal/Portal.mwsl?intro_enter_button=ENTER&PriNav=Start&coming_from_intro=true* predstavlja putanju do web stranice koju treba napasti, odnosno gdje je se treba prijavljivati,
- *F=Invalid Login* ovo predstavlja poruku koju će web stranice prikazati ako podaci za prijavu nisu ispravni; na taj način Hydra zna jesu li korisničko ime i lozinka ispravni,
- *H=Cookie:PHPSESSID=5knVK1zc5UXxvfdcFgNSeNtJF+hM633RuXvSLbmQcbxZ;security=low"* predstavlja kolačić (engl. *cookie*) koji se koristi na web stranici.

Broj pokušaja u minuti (engl. *tries/min*), prikazan na slici 5.9., postepeno se smanjuje i treba sve više i više vremena. Nakon otprilike 50 sati pada na nula što znači da sam web server postepeno smanjuje broj pokušaja koje Hydra može izvesti. Time se zaključuje da server ima *brute force* zaštitu koja usporava napad i značajno ograničava mogućnosti napadača. Teoretski se može

dogoditi da se pogodi korisničko ime i lozinka unutar prvih 100 pokušaja što znači da je i dalje moguće pristupiti web serveru, ali to ne mora nužno biti slučaj.

Također je moguće zaustaviti Hydru te naredbom *hydra -R* nastaviti izvođenje od posljednje kombinacije, budući da Hydra pamti gdje je stala. Na taj se način ponovno kreće s većom brzinom, koja se opet postepeno smanjuje, ali postavlja se pitanje koliko puta je potrebno to napraviti napraviti i je li isplativo.

ZAKLJUČAK

Prodor komercijalnih IT tehnologija u svijet automatskog upravljanja te svepristutan širokopojasni pristup internetu rezultiraju pritiskom na povezivost industrijskih računalnih sustava. Zbog toga je potrebno postaviti odgovarajuću zaštitu za takve sustave.

Ovim radom izrađen je jednostavni sustav automatskog upravljanja sa sučeljem čovjek-stroj s posebnim pogledom na kibernetičku sigurnosti. Programsko rješenje pisano je prema normi IEC 61131-3 koja nalaže korištenje više programskih jezika. Upravljački kod pisan je SCL i LAD programskim jezicima. Izrađeno sučelje čovjek-stroj, koje se sastoji od web stranice, omogućuje jednostavno povezivanje i upravljanje sustavom. Nadogradnja je moguća u vidu veće responzivnosti sučelja kako bi se ono prilagodilo različitim rezolucijama uređaja. Također je dodatno moguće raščlaniti alarme kako bi korisnik mogao imati detaljniji uvid u to koji alarm je trenutno aktivan.

U sklopu rada odrađen je penetracijski test alatom Hydra koji se pokazao neuspješnim, odnosno pokazalo se da je sustav otporan na *brute-force* napad. Sustav prepoznaje veliku količinu upita koji se šalju na server te ih počinje smanjivati i time onemogućavati napadača. Napadač može, nakon što broj upita po sekundi padne na nulu, pokušati ponoviti napad s preostalim imenima i zaporkama, no to vremenski nije isplativo i potrebna je veća angažiranost napadača.

Siemens redovito izdaje zakrpe za svoje sustave koji se nalaze na različitima platformama i time pokazuje brigu i ujedno ublažuje sigurnosne rizike. Također pokušava obrazovati svoje klijente te time drastično smanjiti sigurnosni rizik kojem se sustav izlaže.

Zadatak diplomskog rada je uspješno obavljen. Pokazala su se osnovna načela programiranja izvršnih, zaštitnih i blokadnih funkcija te je također je relizirano upravljanje garažnim vratima putem web panela za dvije vrste korisnika. Korisnici se mogu spojiti putem računala ili putem pametnog telefona nakon što se povežu na istu mrežu kao i PLC. Nije bilo moguće pokriti sve sigurnosne napade na ovaj sustav jer je moguće provesti vrlo veliki broj napada na njega. Jedan od takvih napada je napad na specifični priključak. Moguće je prvo skenirati priključke (engl. *ports*) alatom kao što je Nmap koji daje prikaz portova sustava kao i informacije kao što je broj

priključaka (engl. *port number*), protokol koji se na njemu koristi, je li on otvoren ili ne i sl. Nakon toga je moguće alatom Hydra izvršiti napad na taj specifični port.

LITERATURA

1) P. Biswanath, Industrial Electronics and Control, Chapter 8: Programmable Logic Controller - http://davidlu.net/Topic_8_PLC.pdf?G=736 (1.4.2022.)

2) dip.ing Goran Malčić, Programirljivi logički kontroleri, Tehničko veleučilište u Zagrebu,elektrotehnički odjel - https://kupdf.net/download/plc-skripta-tvz_5b08a8eae2b6f5ff70117aa9_pdf (1.4.2022.)

3) Siemens mall industry catalog, S7-1500, CPU 1513-1 PN - https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7513-1AL02-0AB0 (28.6.2022.)

4) Simatic manual, S7-1500 CPU 1513-1 PN - https://cache.industry.siemens.com/dl/files/842/109752842/att_936139/v1/s71500_cpu1513_1_pn_dtc_manual_en-US_en-US.pdf (28.6.2022.)

5) SIMATIC S7-1500, SIMATIC Drive Controller, ET 200SP, ET 200pro Web server - https://cache.industry.siemens.com/dl/files/560/59193560/att_898124/v1/s71500_webserver_function_manual_en-US_en-US.pdf?download=true (28.6.2022.)

6) Creating and using user-defined web pages on S7-1200 / S7-1500 - https://support.industry.siemens.com/cs/document/68011496/creating-and-using-user-defined-web-pages-on-s7-1200-s7-1500?dti=0&lc=en-WW (19.4.2022.)

7) Inductive automation, What is HMI? - https://www.inductiveautomation.com/resources/article/what-is-hmi (19.4.2022.)

8) Siemens SIMATIC TIA Portal STEP 7 Basic V10.5 - https://cache.industry.siemens.com/dl/files/542/40263542/att_829827/v1/GS_STEP7Bas105enUS.pdf (4.4.2022.)

9) Goran Malčić dip.ing, Programirljivi logički kontroleri, Tehničko veleučilište u Zagrebu, Elektrotehnički odjel- https://nastava.tvz.hr/gmalcic/PLC_skripta_TVZ.pdf (28.6.2022.)

10) Berno David, Efficient SCL Development in TIA Portal V14 - https://www.dmcinfo.com/latest-thinking/blog/id/9540/efficient-scl-development-in-tia-portal-v14 (11.4.2022.)

11) Visual Studio Code - https://code.visualstudio.com/ (4.4.2022.)

12) Tečajevi srca, Uvod u HTML C201, priručnik za polaznike 2016 Srce - https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c201_polaznik.pdf (4.4.2022.)

13) Tečajevi srca, Uvod u HTML C201, priručnik za polaznike 2014 Srce - https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c201_polaznik.pdf (4.4.2022.)

14) Tečajevi srca, Osnove JavaScripta C501, priručnik za polaznike 2015 Srce - https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c201_polaznik.pdf (5.4.2022.)

15) hydra | Kali Linux Tools - https://www.kali.org/tools/hydra/#hydra (13.5.2022.)

16) „rockyou.txt“ datoteka - https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt (23.5.2022.)

17) Common Vulnerability Scoring System version 3.1: Specification Document - https://www.first.org/cvss/specification-document (3.6.2022.)

18) Siemens Security Advisories - https://new.siemens.com/global/en/products/services/cert.html (3.6.2022.)

19) Operational Guidlines for Industrial Securtiy, Siemens 2020, Version 2.1 - https://cert-portal.siemens.com/operational-guidelines-industrial-security.pdf (4.6.2022)

20) SSA-604937: Multiple Web Server Vulnerabilities in Opcenter Execution Core - https://cert-portal.siemens.com/productcert/html/ssa-604937.html (4.6.2022.)

21) CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') - https://cwe.mitre.org/data/definitions/89.html (4.6.2022.)

22) SSA-350757: Improper Access Control Vulnerability in TIA Portal Affecting S7-1200 and S7-1500 CPUs Web Server (Incl. Related ET200 CPUs and SIPLUS variants) - https://cert-portal.siemens.com/productcert/html/ssa-350757.html (8.6.2022.)

23) CWE-284: Improper Access Control - https://cwe.mitre.org/data/definitions/284.html (8.6.2022.)

24) CWE-285: Improper Authorization - https://cwe.mitre.org/data/definitions/285.html (8.6.2022.)

25) SSA-480230: Denial of service in Webserver of Industrial Products - https://cert-portal.siemens.com/productcert/html/ssa-480230.html#mitigations (4.6.2022.)

26) CWE-125: Out-of-bounds Read - https://cwe.mitre.org/data/definitions/125.html (4.6.2022.)

27) SSA-865327: Incorrect Authorization Vulnerability in Industrial Products - https://cert-portal.siemens.com/productcert/html/ssa-865327.html (9.6.2022.)

28) CWE-863: Incorrect Authorization - https://cwe.mitre.org/data/definitions/863.html (9.6.2022.)

29) Što je sustav izvršne proizvodnje (mes)? - https://hr.theastrologypage.com/manufacturing-execution-system (5.7.2022.)

30) S7-1500 Structure and Use of the CPU Memory, Function manual, Siemens - https://cache.industry.siemens.com/dl/files/101/59193101/att_80162/v1/s71500_structure_and_use_of_the_PLC_memory_function_manual_en-US_en-US.pdf (5.7.2022.)

SAŽETAK

Ovaj rad bazira se na izradi sustava automatskog upravljanja garažnim vratima s pripadajućim sučeljem čovjek-stroj pomoću PLC uređaja. Za upravljački dio sustava korišten je PLC S7-1513-1 PN i TIA Portal, a upravljački kôd napisan je pomoću LAD i SCL programskih jezika. Također je pomoću Visual Studio Code-a izrađeno sučelje čovjek-stroj u obliku web stranice s kojeg je omogućeno upravljanje sustava te je postavljeno na web server PLC-a. Sučelje je izrađeno koristeći HTML, CSS, JS programske jezike. Upravljački kôd testiran je pomoću HMI panela, koji je izrađen unutar TIA Portala, a sučelje je testirano prateći varijable unutar podatkovnog bloka TIA Portala. Nakon ručnog testiranja obrađene posebno je obrađen aspekt kibernetičkih sigurnosti sustava. pomoću *Siemens Security Advisories*-a(preporuke vezane uz KS) te je izvršen *brute-force* napadna ugrađeni web server PLC-a programskim alatom Hydra.

Ključne riječi: PLC, S7-1513-1 PN, TIA Portal, LAD, SCL, Visual Studio Code, HTML, CSS, JS, HMI, Siemens Security Advisories, Hydra

ABSTRACT

Title: Automatic Garage Door Control System with ability to control and manage over internet

This paper is based on making an automatic garage door system with a human-machine interface. Automatic control system was made using a PLC S7-1513-1 PN and TIA Portal and the code was written in LAD and SCL programming languages. Also, a human-machine interface was made using Visual Studio Code that enables controlling of the system and is uploaded to the PLC's web server. Human-machine interface was programmed using HTML, CSS and JS programming languages. Automatic control system was manually tested with an HMI panel that was made using TIA Portal and the human-machine interface was tested by watching variable response inside data block within TIA Portal. After testing, some cybersecurity aspects of the system were dealt with using Siemens Security Advisories and a brute force attack was done using Hydra tool.

Key words: PLC, S7-1513-1 PN, TIA Portal, LAD, SCL, Visual Studio Code, HTML, CSS, JS, HMI, Siemens Security Advisories, Hydra

ŽIVOTOPIS

Matija Hajduković rođen je 15. lipnja 1998. u Osijeku gdje 2012. godine završava Osnovnu školu Frana Krste Frankopana. Kasnije te iste godine upisuje Elektrotehničku i prometnu školu u Osijeku koju nakon 4 godine završava te stječe zvanje tehničara za elektroniku. Nakon završetka srednje škole upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, smjer elektrotehnika, a na 2. godini studija opredjeljuje se za smjer Komunikacije i informatika. Godine 2020. steče zvanje sveučilišnog prvostupnika (baccalaureus) inženjera elektrotehnike i informacijskih tehnologija te upisuje diplomski sveučilišni studij, smjer Mrežne tehnologije, na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.