

Univerza v Ljubljani  
Fakulteta za elektrotehniko

Rok Terkaj

# **Izvedba nadzornega sistema na potniški ladji**

Diplomsko delo visokošolskega strokovnega študija

Mentor: prof. dr. Gašper Mušič

Ljubljana, 2017



## Zahvala

Posebna zahvala gre mojemu mentorju prof. dr. Gašperju Mušiču za vse nasvete in vodila pri moji diplomski nalogi.

Zahvalil bi se tudi sodelavcem z oddelka PE MI, s katerimi smo sodelovali v podjetju INEA, kjer sem delal v tem letu, in Blažu Tavčarju, ki mi je omogočil delo na tem projektu. Močno so mi izboljšali znanje o nadzornih sistemih in mi v določenih situacijah tudi pomagali s svojimi nasveti in izkušnjami.

Zahvaliti se želim tudi družini in prijateljem, ki so me v času študija na Fakulteti za elektrotehniko vzpodbujali in verjeli vame.



# Vsebina

<b>1</b>	<b>Uvod</b>	<b>5</b>
<b>2</b>	<b>Opis programske in strojne opreme</b>	<b>7</b>
2.1	Programsko okolje Intouch 2014 . . . . .	7
2.1.1	Razvojno okolje Window Maker . . . . .	8
2.1.2	Izvajalsko okolje Window Viewer . . . . .	10
2.2	System Management Console . . . . .	11
2.2.1	Komunikacijski gonilnik FSGateway . . . . .	13
2.2.2	Povezave s krmilniki . . . . .	14
2.2.3	Zavihek SMC - Device Groups . . . . .	16
2.3	Uporabljeni krmilniki . . . . .	17
2.3.1	Siemens S7-300 . . . . .	18
2.3.2	Siemens S7-1500 . . . . .	18
<b>3</b>	<b>Izvedba nadzornega sistema</b>	<b>21</b>
3.1	Delo s signali . . . . .	22
3.2	Animiranje . . . . .	25
3.3	Urniki in zmanjšana poraba energije . . . . .	28
3.3.1	Pregled tabele . . . . .	29

3.3.2	Delovanje urnikov . . . . .	31
3.4	Poraba goriva in poročila . . . . .	32
3.4.1	Vrste goriva in GT . . . . .	32
3.4.2	Merjenje porabe goriva . . . . .	34
3.4.3	Poročila porabe goriva in ladijska poročila . . . . .	35
3.5	Varnost . . . . .	39
3.5.1	Varovanje operacijskega sistema . . . . .	39
3.5.2	Vrste uporabnikov in varovanje aplikacije . . . . .	39
3.6	Časovni poteki signalov in alarmiranje . . . . .	41
3.6.1	Časovni poteki signalov . . . . .	41
3.6.2	Alarmi in alarmna konzola . . . . .	43
3.7	Diagnostika na nadzornem sistemu . . . . .	44
<b>4</b>	<b>Sistem Silverstream</b>	<b>47</b>
4.1	Ventili in kompresorji sistema Silverstream . . . . .	48
4.1.1	Alarmiranje sistema Silverstream . . . . .	52
4.1.2	Modul PCM . . . . .	52
<b>5</b>	<b>Zaključek</b>	<b>55</b>

## Seznam slik

2.1	Prikaz osnovnega okna Window Makerja . . . . .	9
2.2	Prikaz osnovnega okna Window Viewerja . . . . .	11
2.3	Prikaz orodne vrstice SMC . . . . .	12
2.4	Prikaz različnih krmilnikov v SMC . . . . .	15
2.5	Zavihek Diagnostics . . . . .	17
2.6	Siemens PLC S7 - 1500, skupaj z modulom . . . . .	19
3.1	Dostopno ime v Intouchu . . . . .	23
3.2	Primer datoteke, namenjene uvozu signalov . . . . .	25
3.3	Primer barvne animacije . . . . .	26
3.4	Primer animacije izpisa . . . . .	27
3.5	Primer animacije vidnosti s statičnimi barvami . . . . .	28
3.6	Tabelni pregled urnikov . . . . .	29
3.7	Izbira časovnega delovanja urnikov . . . . .	30
3.8	Tabela v Microsoft Access . . . . .	31
3.9	Zaslon namenjen prikazovanju porabe goriva . . . . .	35
3.10	Zaslon za tvorjenje poročil porabe goriv . . . . .	37
3.11	Tvorjenje skripta za poročila . . . . .	38

3.12	Obstoječi uporabniki sistema SCADA . . . . .	40
3.13	Primer okna za prikaz časovnega poteka . . . . .	42
3.14	Zaslon z diagnostiko . . . . .	45
4.1	Delovanje sistema Silverstream . . . . .	48
4.2	Sistem Silverstream v nadzornem sistemu . . . . .	50
4.3	Prikaz zaslona kompresorja . . . . .	51
4.4	Nadzorni modul PCM . . . . .	54



## Seznam uporabljenih simbolov

Kratica	Angleško ime	Slovensko ime
PLC	Programmable logic controller	Programirljivi logični krmilnik
SCADA	Supervisory control and data acquisition	Nadzorni sistem
OPC	Object Linking and Embedding for Process Control	Povezovanje in vključevanje objektov za vodenje procesov
HMI	Human Machine Interface	Vmesnik človek-stroj
HFO	Heavy Fuel Oil	Težko dizelsko gorivo
MGO	Marine Gas Oil	Lažje dizelsko gorivo
GT	Gas Turbine	Turbina za gorivo
DG	Diesel generator	Dizelski generator
PCM	Pleiger Control Module	
PID	Proportional–Integral–Derivative controller	Proporcionalno-integrirno-diferencirni regulator
SMC	System Management Console	Konzola za upravljanje s sistemom
SQL	Structured Query Language	Strukturirani povpraševalni jezik
RTU	Remote terminal unit	Enota oddaljenega terminala
HVAC	Heating, ventilation and air conditioning	Gretje, prezračevanje in klimati
TCP	Transmission Control Protocol	Protokol za nadzor prenosa
IP	Internet Protocol	Internetni Protokol



## Povzetek

Sistem SCADA (Supervisory control and data acquisition) je nadzorni sistem, katerega arhitektura vsebuje računalnike, omrežne podatkovne komunikacije in grafične uporabniške vmesnike za nadzorno vodenje procesov. Moje delo v podjetju INEA je temeljilo predvsem na razvoju nadzornih sistemov za kemijsko, gradbeno in kasneje za pomorsko industrijo, kar je privedlo do teme za diplomsko delo.

Diplomsko delo sestoji iz dveh delov: opisa programske in strojne opreme in izvedbe nadzornega sistema na potniški ladji.

V prvem delu se bom osredotočil na uporabljeno programsko in strojno opremo skupaj s protokoli in krmilniki, ki so bili pri tem projektu uporabljeni. V drugem delu bo predstavljeno animiranje elementov v aplikaciji, urniki, merjenje porabe goriv in poročila ter sistem Silverstream. Tako bo bralcu najprej predstavljeno, kaj to nadzorni sistem sploh je, nato pa nekaj zanimivih stvari, s katerimi sem se srečeval pri svojem vsakodnevnem delu.

**Ključne besede:** SCADA, PLK, potniška ladja, OPC, sistem Silverstream



## Abstract

System SCADA (Supervisory Control and Data Acquisition) is a control system whose architecture includes computers, network data communications and graphical user interfaces for process control and supervision. My work at INEA was based primarily on the development of control systems for the chemical, construction and later for the marine industry, which also led to this topic.

My final work consists of two main parts: a description of the software and hardware and the implementation of the supervisory control system on a cruise ship. In the first part, I will concentrate on the software and hardware used, together with the protocols and PLCs that were used in this project.

The main part consists of the chapters that I work with in the project. Animation of the elements in the application, schedules, fuel consumption and reports, and the Silverstream system will be presented. Thus, the reader will first be introduced to what the supervisory control system is at all, and then some interesting things that I worked with everyday.

**Key words:** SCADA, PLC, Cruise ship, OPC, system Silverstream



# 1 Uvod

Moje delo v podjetju INEA je temeljilo predvsem na razvoju nadzornih sistemov za kemijsko, gradbeno in kasneje za pomorsko industrijo. Večino svojega časa sem posvetil razvoju nadzornega sistema za potniške ladje. Srečeval sem se s klimati, gorivi, s sistemom Silverstream itd.

Na tako velikem objektu, kot je potniška ladja za križarjenje, kjer je poleg posadke tudi zelo veliko potnikov, je zelo pomembno, da njihovo bivanje poteka brez zapletov in v lagodju, saj traja potovanje tudi po več dni. Potniško ladjo si lahko predstavljamo kot luksuzen hotel na vodi. Nadzorni sistem je torej zato zelo pomemben, saj operaterji z njim uravnavajo klimatske sisteme in s tem temperaturo, spremljajo alarme, porabo posameznega goriva ... Predstavljajte si, da temperatura v bivalnih prostorih, jedilnicah in v ostalih prostorih, kjer se zadržujejo potniki, ne bi bila ustrezna glede na klimo zunaj? To bi povzročilo pravo jezo in ogorčenje potnikov. Kaj bi se zgodilo, če vodilni ne bi vedeli, koliko goriva porabi ladja dnevno, mesečno, letno? Vse to boste izvedeli v mojem diplomskem delu.

Večina ljudi si pod pojmom nadzorni sistem predstavlja samo izrisane grafike, njihove animirane sestavne dele in izpisne podatke poleg elementov grafik. Na vsaki grafiki in vsakem prikazovalnem polju, torej na njeni animaciji, se skriva neko zelo pomembno ozadje, saj se tukaj nahajajo podatki o sistemu, katerega nadzorujemo.

V nadaljevanju bom bolj podrobno opisal programsko in strojno opremo, s katero sem se srečeval pri svojem vsakodnevnem delu. S tem mislim na razvojno

okolje, v katerem sem razvijal nadzorni sistem, programsko opremo, ki sem jo pri svojem delu potreboval – krmilnike in strežnik, ter tudi komunikacijo med krmilniki in nadzornim sistemom s pripadajočim komunikacijskim gonilnikom.

V zadnjem delu sledi opis nekaterih stvari, ki se mi zdijo pomembne za delovanje nadzornega sistema oz. za operaterje na ladji. Opisal in predstavil bom animacije in njihov pomen uporabniku nadzornega sistema, uvoz velike količine spremenljivk, predstavil sistem Silverstream (t. i. sistem vodnih mehurčkov), goriva, poročila, alarmiranje, konzolo za spremljanje časovnih potekov signalov in urnike. Tako bo lahko bralec dobil malo boljšo predstavo o tem, kaj nadzorni sistem sploh je in zakaj je za uporabnika tako pomemben.



## 2 Opis programske in strojne opreme

Sistem SCADA je nadzorni sistem, katerega arhitektura vsebuje računalnike, omrežne podatkovne komunikacije in grafične uporabniške vmesnike za nadzorno vodenje procesov. Lahko uporablja tudi druge naprave, večinoma PLC-je. Koncept SCADA je bil sprva namenjen kot neko univerzalno sredstvo za dostop do različnih krmilnih modulov, ki omogočajo dostop preko standardnih komunikacijskih protokolov. So najbolj pogosto uporabljeni tipi industrijskih nadzornih sistemov za vodenje [2].

### 2.1 Programsko okolje Intouch 2014

Pri svojem vsakodnevnem delu, sem uporabljal programsko okolje Wonderware Intouch 2014. Intouch HMI je orodje za vodenje in vizualizacijo procesov. Vmesnik človek-stroj je t. i. okno operaterja nadzornega sistema. Grafično predstavi podatke o obratu operativnemu osebju v obliki raznih diagramov, ki predstavljajo shematično predstavitev obrata, ki ga nadzorujemo. HMI je povezan z nadzornim računalnikom SCADA z nalogo zagotavljanja podatkov za animiranje elementov, prikazov alarmov in grafov za časovno spremljanje signalov. V mnogih instalacijah je HMI grafični uporabniški vmesnik za operaterja, ki zbira vse podatke iz zunanjih naprav, ustvarja poročila, zbira alarmne podatke in pošilja obvestila [2].

Programsko okolje je razdeljeno na dva dela, na razvojni in del namenjen izvajanju aplikacije. Njegove storitve so:

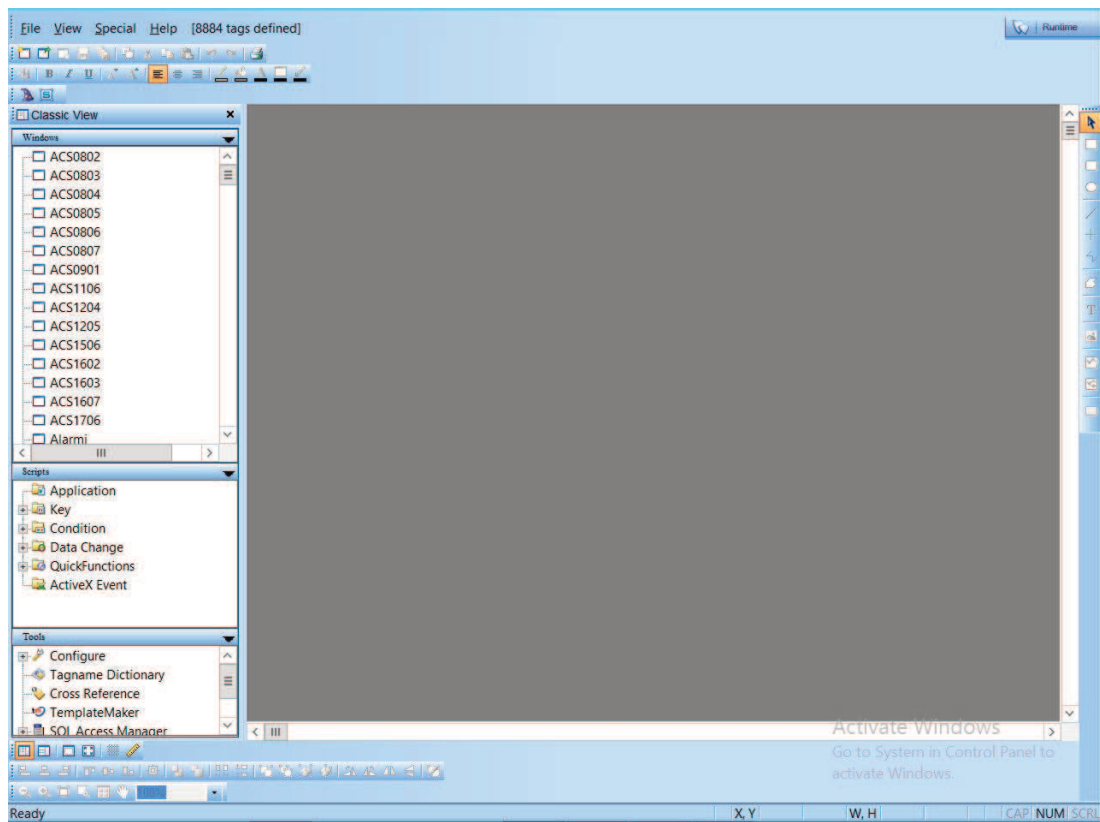
- upravljanje z licencami,
- upravljanje z verzijami,
- sistemska diagnostika in administracija,
- prikazovanje podatkov,
- prikazovanje dogodkov in skriptiranje,
- prikazovanje alarmov in zgodovine,
- internacionalizacija [1].

### 2.1.1 Razvojno okolje Window Maker

Kot že ime pove, Window Maker služi za razvoj aplikacije, je razvojno okolje aplikacije. Namenjen je kreiranju objektno naravnanih grafik in animiranju letih. Torej tukaj lahko razvijalec v celoti in brez težav razvije svojo aplikacijo, ki se lahko z lahkoto poveže na industrijske vhodno-izhodne (V/I) sisteme. Poznamo tri vrste verzij aplikacije: Stand Alone, Modern in Legacy. Verzija aplikacije, ki sem jo uporabljal, je bila Stand Alone, kar pomeni, da se lahko uporabi le v Intouchu in da v njej niso dostopne ArchestraA grafike.

Pred opisom programa je potrebno omeniti še, da je okolje konfigurabilno, torej si ga lahko uporabnik preoblikuje po svojih željah in potrebah. To pomeni, da si lahko na sliki 2.1 prikazane orodne vrstice premakne na željen položaj znotraj razvojnega okolja. Slika 2.1 prikazuje aplikacijski raziskovalnik, ki omogoča hierarhični grafični pogled na elemente aplikacije. Ta pogled razvijalcu omogoča hiter dostop do oken, skript, menijev in ostalih sestavnih delov, ki smo jih vključili v aplikacijo. Omogoča tudi shranjevanje, odpiranje oken in preimenovanje aplikacije [1].

V svoji aplikaciji imam vključen klasičen pogled. Obstajata sicer dve vrsti pogleda, razlika med klasičnim in projektnim pogledom je le ta, da zadnji



Slika 2.1: Prikaz osnovnega okna Window Makerja

omogoča razdelitev delov projekta v projektne mape. Tako si lahko razdelimo projekt na več različnih delov in na tak način lažje vodimo evidenco nad svojim delom. Leva stran okolja je sestavljena iz treh delov in kot sem že omenil, predstavlja hierarhičen grafičen pogled na aplikacijo. Prikazuje, katere elemente smo skonfigurirali in omogoča lahek dostop do njih:

- Seznam in urejanje vseh vsebovanih projektnih oken ter dodajanje novih. Tukaj programsko okolje daje možnost izbire velikosti koordinat okna in tip okna.
- Nabor različnih vrst skriptov, ki jih lahko dodajamo in urejamo. Poznamo skripte, ki se spreminjajo glede na željen pogoj, na spremembo podatka in funkcije, ki jih lahko kličemo znotraj skriptov.

- Nabor orodij za konfiguracijo projektnih delov, komunikacije, dodajanje novih in spreminjanje obstoječih signalov itd.

Na desni strani slike 2.1 lahko opazimo orodja za risanje, s katerimi načrtujemo elemente željenega zaslona in s katerimi zastavljamo svoj projekt. Lahko jih uporabimo za risanje enostavnih ali kompleksnih objektov. Ti elementi so edini, ki so v tem projektu omogočeni in katere sem vseskozi uporabljal. Na spodnji strani zaslona opazimo ikone, ki so združene glede na svoje skupne funkcionalnosti in tvorijo urejevalno vrstico. Govora je glede na različne načine poravnav, pozicij prikazovanja elementov, rotiranju itd. Nad njo stoji tudi vrstica z elementi za približevanje. Desno spodaj imamo statusno vrstico, ki sporoča pozicijo objekta na zaslonu, velikost objekta ter omogoča spreminjanje koordinat pozicije in velikosti objektov [1].

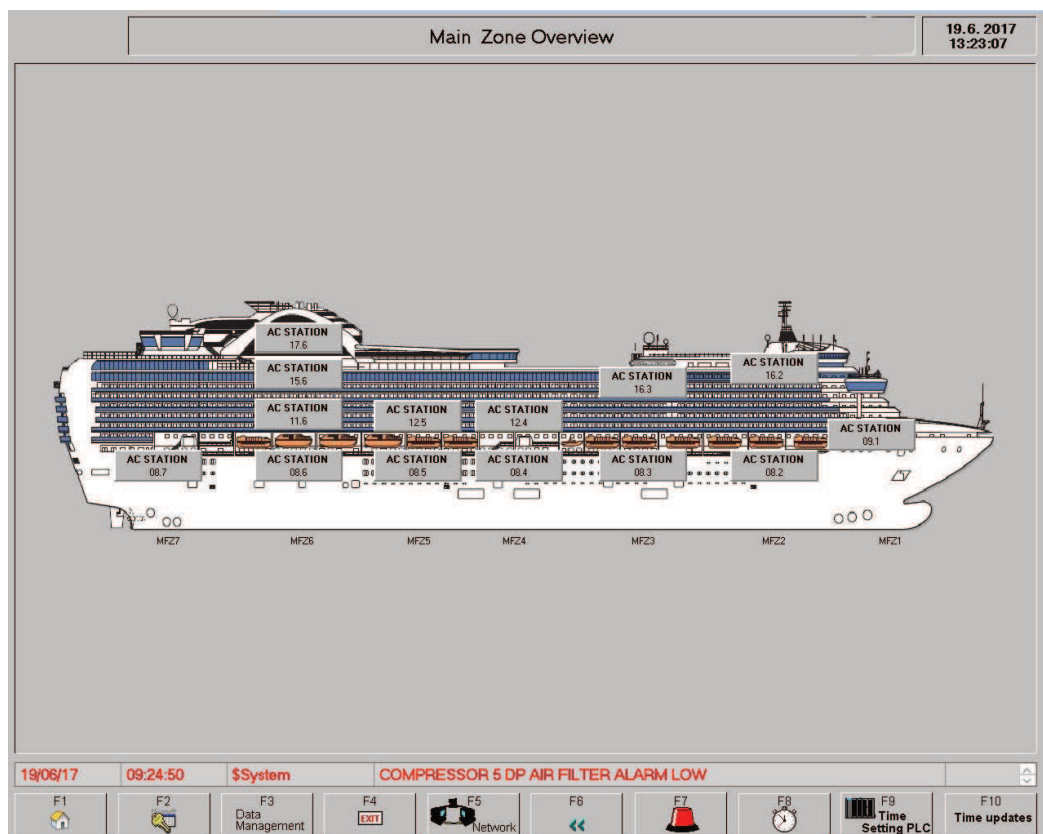
### 2.1.2 Izvajalsko okolje Window Viewer

Window Viewer je programsko okolje, ki je namenjeno izvajanju grafik, ki so bile predhodno kreirane v Window Makerju s strani razvijalca aplikacije. Window Viewer izvaja t. i. Quick skripte, historizacijo podatkov, tvorjenje poročil, alarmiranje itd. Nadzorni sistem je povezan s PLK-jem in prikazuje veličine, ki so namenjene prikazovanju na nadzornem sistemu. Če razvijalec omogoči, lahko tudi pišemo veličine v krmilnik [1].

Izvajalsko okolje je zgrajeno iz:

- Statusne vrstice na vrhu zaslona z imenom akutalnega okna, logotipom stranke in digitalnim prikazom datuma in ure.
- Osnovnega okna prikaza, ki se prikaže ob aktivaciji Window Viewerja. Na sliki 2.2 lahko opazimo celotni pregled ladje, ki služi kot osnova projekta.
- Navigacije, ki vsebuje vse gumbe za prehode med zaslone, alarmno kontrolo ter upravljanjem z urniki. Ker vemo, da ladja pluje po različnih delih sveta,

ki se nahajajo v različnih časovnih pasovih, je zelo pomembno, da nadzorni sistem vsebuje tudi gumb za časovne nastavitve.



Slika 2.2: Prikaz osnovnega okna Window Viewerja

## 2.2 System Management Console

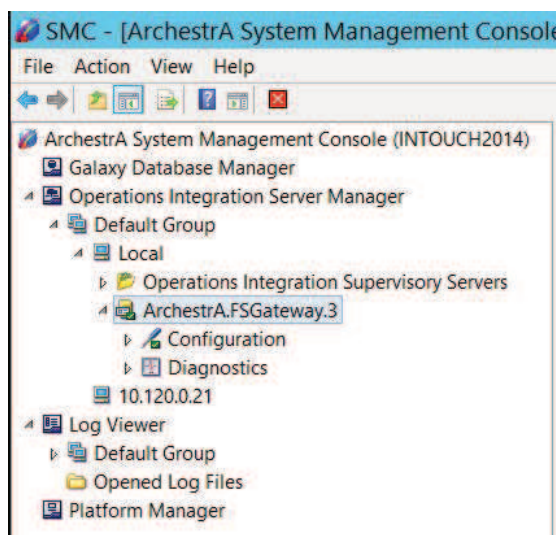
SMC (ArchestrA System Management Console) je Wonderwarejeva konzola, ki se uporablja za pregled kakršnih koli napak med izvajanjem aplikacije in tudi za komunikacijo med nadzornim sistemom in pripadajočimi krmilniki. V njej lahko skonfiguriramo različne komunikacijske gonilnike, ki skrbijo za različne tipe komunikacij skupaj z njihovimi protokoli. Potrebno se je zavedati, da imamo lahko v že omenjeni konzoli uporabljenih več različnih tipov gonilnikov, ki lahko

skrbijo za komunikacijo s krmilniki različnih proizvajalcev. Paziti je potrebno le na to, da zaradi tega ne prihaja do motenj v komunikaciji in posledično časovnih zamikov ali napak.

Dva najbolj pomembna sestavna dela konzole SMC sta:

- Operations Integration Server Manager (OI Server Manager) in
- Log Viewer

V zavihku OI Server Manager najdemo pripadajoči komunikacijski gonilnik, s katerim poteka komunikacija med strežnikom OPC in nadzornim sistemom, kar bo opisano v razdelku 2.2.1. Naslednja zelo uporabna stvar SMC-ja je razdelek Log Viewer. Njegova naloga je, da javi kakršno koli napako, ki se je zgodila v komunikaciji med nadzornim sistemom in strežnikom OPC ali pri izvajanju aplikacije. Na tak način jo lahko hitro izsledimo in odpravimo. Javi tudi čas pojava le-te ter barvo njenega kritičnega stanja – siva, rumena in rdeča barva. Nudi možnost izvoza v datoteko, kar se včasih izkaže za zelo uporabno stvar, saj ta datoteka prihrani čas pri morebitnem iskanju napake.



Slika 2.3: Prikaz orodne vrstice SMC

### 2.2.1 Komunikacijski gonilnik FSGateway

FSGateway je gonilnik, ki je uporabljen za komunikacijo med strežnikom OPC in nadzornim sistemom, preko protokola SuiteLink. Protokol SuiteLink temelji na povezavi TPC/IP in deluje kot storitev za Windows. Najpogosteje ga najdemo v industrijski uporabi, kjer so pomembni časovna in druga obsežna diagnostika [3].

Lastnosti protokola SuiteLink:

- Kakovost časa (VTQ) postavlja časovni žig in kazalnik kakovosti za vse vrednosti podatkov.
- Obsežna diagnostika vključno z nalaganjem strežnika, računanjem porabe virov in omrežnega prometa. Dostopna prek operacijskega sistema Microsoft Windows NT.
- Med dvema aplikacijama, ki sta na enem vozlišču, ali med aplikacijami na več vozliščih je mogoče ohraniti visok obseg podatkov [3].

Gonilnik je potrebno namestiti lokalno, torej na računalniku oz. strežniku, na katerem se nahaja nadzorni sistem v točno določeni mapi. Nahaja se pod osnovno skupino, ki nosi ime DAServer Manager. Lokacija gonilnika v SMC-ju je pod zavihkom OI Server Manager, in sicer v skupini ArchestrA.FSGateway.3, kar nazorno prikazuje slika 2.3. Gonilnik ima hierarhično drevesno strukturo skupaj s konfiguracijsko detajlno strukturo. Potrebno je nastaviti globalne parametre gonilnika, ki se nahajajo pod zavihkom namenjenim za konfiguracijo. Tukaj se nastavi časi osveževanja, posodobitev in transakcij podatkov.

Naslednja zelo pomembna stvar za delovanje komunikacijskega gonilnika je pravilna izbira povezave. Iz zgornjega zapisa lahko opazimo, da je za tip povezave izbrana OPC-povezava. Za pravilno delujočo komunikacijo je potrebno nastaviti tudi ustrezne parametre strežnika. Za komunikacijo je uporabljen strežnik OPC in tukaj je zelo pomembno, da v parametrih zapišemo njegov IP-naslov (strežnik

se nahaja na IP-naslovu 10.120.0.10) in ime aktualnega strežnika (OPCSimatic.NET.1). Le tako bosta strežnik OPC in nadzorni sistem komunicirala brez težav preko izbranega komunikacijskega gonilnika.

### **2.2.2 Povezave s krmilniki**

V razdelku 2.2.1 je na kratko opisano, kako poteka komunikacija med strežnikom OPC in nadzornim sistemom. Virtualni okolji, na katerih se nahajata strežnik OPC in programsko okolje Intouch 2014, sta nameščeni v posebnem računalniku, ki nosi ime Glavni strežnik. Namenjen je delovanju v industrijskem okolju, zato ni občutljiv na tresljaje in je obdan s kovinskim ogradjem. Ima več diskov, z namenom večkratnega shranjevanja podatkov z razlogom, če kakšen od njih odpove. Vsak naslednji disk je namenjen rezervi prejšnjemu. Na strežnik je priključen HP-jev Thin Client, ki služi prikazovanju podatkov na zaslon. Tudi on je prav tako uporabljen zaradi njegove prilagodljivosti na industrijska okolja. Iz zadnje strani strežnika je vhod za omrežni kabel Profibus, ki služi za povezavo s krmilniki.

Virtualno okolje s strežnikom OPC, mora za pravilno komunikacijo z nadzornim sistemom vsebovati:

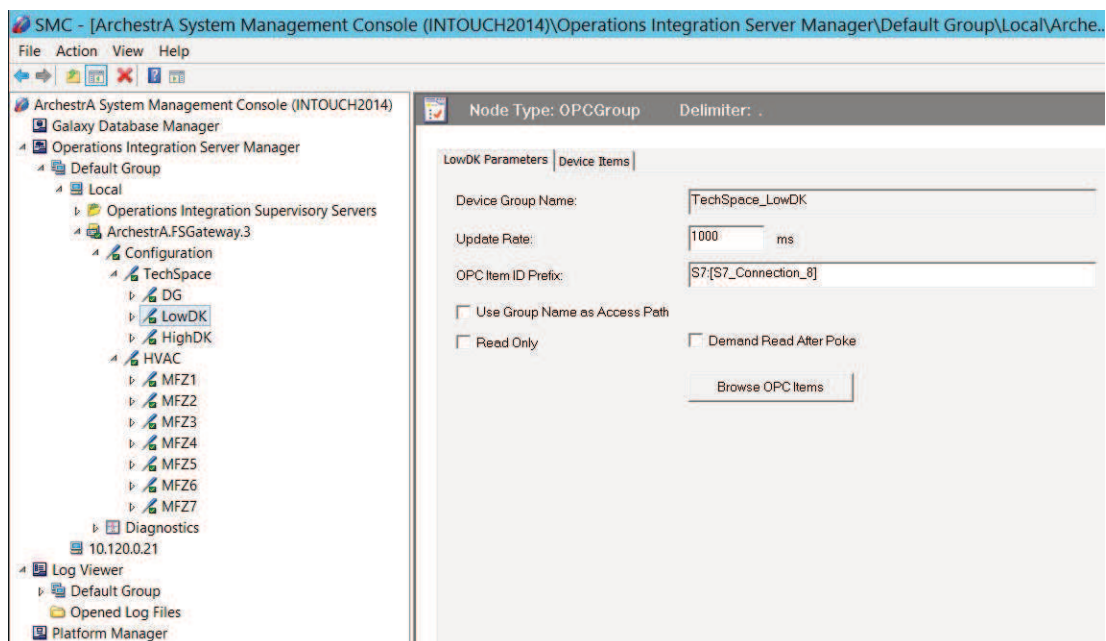
- Profibus kartico: Odgovorna je za komunikacijo med krmilniki, ki so med seboj povezani z omrežnim kablom Profibus in strežnikom OPC.
- Ethernet kartico: Odgovorna je za komunikacijo med krmilniki, ki so med seboj povezani z omrežnim kablom Ethernet.
- Kartico za delovanje strežnika OPC.

Za uspešno in zanesljivo komunikacijo med strežnikom in nadzornim sistemom pa seveda potrebujemo gonilnik, opisan v razdelku 2.2.1. Na ladji je več krmilnikov, katerih opis sledi v nadaljevanju podpoglavja, ki imajo svoj OPC ID Prefix, ki predstavlja njihovo identifikacijo.



Krmilniki so razdeljeni v dve skupini:

- Krmilniki, ki so odgovorni za klimate. To so krmilniki vrste S7-300, njihovo število pa je enako številu con na ladji.
- Krmilniki, ki so odgovorni za naprave in njihovo delovanje v strojnici. Zanimi so poleg že omenjenih S7-300 tudi vrste S7-1500.



Slika 2.4: Prikaz različnih krmilnikov v SMC

Slika 2.4 prikazuje razdelitev pripadajočega gonilnika na dve povezavi OPC:

- TechSpace, ki predstavlja nivo strojnice in vsebuje 3 krmilnike.
- HVAC, ki predstavlja nivoje ladijskih con in vsebuje 7 krmilnikov.

Ker ima vsak od nivojev strojnice in vsak od ladijskih con svoj krmilnik, je zelo pomembno iz vidika pravilno delujoče in ponastavljene komunikacije, da ima vsak od njih svojo povezavo OPC in svoj identifikacijski naslov. Identifikacija krmilnika je določena s strani PLK-jev, kjer programerji PLK-jev nastavijo

ustrezno ime in krmilnike povežejo med sabo. Za razvijalce nadzornega sistema je zelo pomembno, da se imeni med seboj ujemata zaradi preprostega razloga: pod enim naslovom potujejo vsi signali, ki so obstoječi na tistem krmilniku.

Končna datoteka s povezavami krmilnikov se skreira v mapi od komunikacijskega gonilnika FSGateway. Lahko jo skreiramo ročno, vendar je za nadaljnjo konfiguracijo povezav zelo dobrodošlo urejanje datoteke v Notepad++ v obliki .xml. Tako razvijalec pridobi na času in zmanjša pogostost nastopa neželenih napak. Datoteko se lahko pridobi tudi z izvozom datoteke iz konzole SMC.

### **2.2.3 Zavihek SMC - Device Groups**

Konzola SMC vsebuje tudi zavihek z diagnostiko, ki je prikazan na sliki 2.5. Ta zavihek razvijalcu pove vse o obstoječih signalih v aplikaciji: koliko je aktivnih signalov, signalov v napaki, kakšen je njihov status delovanja, njihova aktualna vrednost itd.

Zavihek z diagnostiko je zelo pomemben tudi za uporabnika aplikacije, saj lahko takoj opazi, v kakšnem stanju so signali (ali so v napaki oz. v pravilnem delovanju) in v tem primeru tudi primerno postopa. Omogočeno mu je deaktivirati in ponovno aktivirati cel gonilnik ali pa le eno od skupin OPC-ja, in sicer tisto, na kateri se nahaja nedelujoči krmilnik in tako lahko takoj na objektu razreši težave, ki jih je povzročila nedelujoča povezava.

Prav zaradi lažjega sledenja signalov je zavihek narejen za vsak krmilnik posebej. Program vsebuje zelo veliko signalov, ki bi jim bilo skoraj nemogoče slediti, brez ustrezne razporeditve diagnostike. Slednja je odvisna od razvijalca, vendar potrebno se je zavedati, da lahko to diagnostiko uporabljajo tudi ladijski operaterji. Slika 2.5 prikazuje pogled na zavihek Device Groups in diagnostiko konzole SMC. Kot vidimo iz slike, so signali obarvani rdeče, kar pomeni, da niso aktivni oz. so nedelujoči. Ker opazimo, da je njihov status R/W, bi v tem primeru postopali na tak način, da bi ponastavili obstoječe skupine in v Log Viewerju opazovali status rezultata.

Name	R/W Status	Value	Time	Qualit...	MsgID	Loc
S7:\$7_Connection_10 DB1111.W1152	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1111.W1162	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1111.W1126	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1940	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1128	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1152	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1780	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1968	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1668	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1460	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1948	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1144	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1904	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1944	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1972	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1148	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1976	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1448	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1452	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1140	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1988	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1924	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1472	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1996	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1348	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1920	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1676	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1456	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1900	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1464	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1136	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1896	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1928	R/W		11:54:19	0008		Tec
S7:\$7_Connection_10 DB1002.REAL1952	R/W		11:54:19	0008		Tec

Slika 2.5: Zavihek Diagnostics

## 2.3 Uporabljeni krmilniki

Na tako velikem objektu, kot je potniška ladja, se nahaja več med seboj povezanih krmilnikov, katerih povezavo predstavlja omrežni kabel Profibus.

PLK je industrijski digitalni računalnik, ki je prilagojen za vodenje procesov. Zagotavlja visoko mero zanesljivosti, enostavnosti programiranja in odkrivanja napak procesa. PLK-ji se lahko gibljejo od majhnih naprav z desetimi vhodi-izhodi (V/I), v integriranem ohišju s procesorjem do velikih regalnih modularnih naprav s številom nekaj tisoč vhodov-izhodov (V/I). Lahko so zasnovani za večkratno ureditev digitalnih in analognih vhodov-izhodov (V/I), razširjenih temperaturnih območij, odpornosti proti električnemu hrupu in za odpornost na vibracije in udarce. Programi za nadzor delovanja naprave so običajno shranjeni v centralni procesni enoti. Funkcionalnost PLK-jev se je skozi leta razvila, da vključuje sekvenčno relejsko vodenje, vodenje procesov ... Ravnanje s podatki,

shranjevanje, procesna moč in komunikacijske zmogljivosti nekaterih sodobnih PLK-jev so približno enake namiznim računalnikom [4]. Vsak krmilnik vsebuje svoje bloke, ki opravljajo točno določene naloge – vsak krmilnik takorekoč skrbi za svoj del ladje. Tudi za potrebe testiranja smo v podjetju INEA uporabili vsaj dva različna krmilnika: S7-300 in S7-1500.

### 2.3.1 Siemens S7-300

Siemensovi krmilniki S7-300 so še vedno vodilni na svetu zaradi svojega ponujanja maksimalne avtomatizacije za majhno ceno. So zelo preprosti za programiranje in operiranje, prihranijo veliko prostora in so zelo zmogljivi.

Uporablja se jih v:

- avtomobilski industriji,
- procesni tehniki,
- za PID-regulacije,
- mehanski industriji,
- plastični tehniki,
- živilski industriji.

Če naloga avtomatizacije zahteva več kot 8 modulov, je mogoče krmilnik razširiti s pomočjo razširitvene ploščice do 32 modulov. Na njega je možno priklopiti do 125 postaj. Komunikacija poteka preko priključka Ethernet ali priključka Profibus [5].

### 2.3.2 Siemens S7-1500

Krmilnik S7-1500 (slika 2.6) je zelo napreden krmilnik, ki se uporablja v celotni avtomatizaciji industrije do naprav z visoko stopnjo zmogljivosti.

Vsak PLC vsebuje:

- centralno procesno enoto,
- en ali dva vira napajanja,
- vhodno-izhodne (V/I) module,
- zaslonček,
- komunikacijske module,
- požarni zid.

Tudi tukaj poteka komunikacija preko priključka Profibus ali priključka Ethernet. Ima tri priključke Ethernet in en priključek Profibus [6].



Slika 2.6: Siemens PLC S7 - 1500, skupaj z modulom



### 3 Izvedba nadzornega sistema

V tem poglavju se bom osredotočil predvsem na sistem SCADA, torej na programsko okolje Intouch in opisal delo v njem. Predstavljene bodo stvari, ki so potrebne za razumevanje razvojnega okolja, nekateri pomembni sestavni deli tega projekta ter sistem, s katerim se je podjetje INEA srečalo prvič do sedaj.

Sistemi za nadzor in vodenje so sestavni del vsakega vsaj delno avtomatiziranega procesa. Sisteme za nadzor in zbiranje podatkov se lahko razvija na obstoječi ali na povsem novi platformi. Razvijalci se ukvarjajo s t. i. psihologijo grafike, ki skrbi, da so aktualne informacije podane na ustrezen način ter ob pravem času. Na potniški ladji se nahaja več nadzornih sistemov, saj za delovanje naprav in nadzor nad njihovim delovanjem skrbi več podjetij. Vsako podjetje skrbi za točno določen del ladje oz. njenih sestavnih delov.

Funkcije nadzornega sistema na ladji so naslednje:

- upravljanje z urniki, ki skrbijo za klimatske sisteme (izbira časovnega obdobja delovanja, izbira željenih prostorov delovanja),
- pregled nad porabo goriv za posamezen dizelski generator (DG) in turbino za gorivo (GT), ter tvorjenje poročil za čas križarjenja,
- pregled nad aktualnimi alarmi ter možnost potrditve le-teh. Za lažji pregled, je operaterjem omogočena uporaba filtra po željenih ladijskih prostorih,
- pregled nad kompresorji in ventili sistema Silverstream in upravljanje s tem sistemom,

- pregled nad delovanjem frekvenčnikov v ventilacijskih sobah in spremljanje delovanja na konzoli za časovno spremljanje signalov,
- pregled nad porabo energije za različna časovna obdobja,
- pregled nad delovanjem črpalk (Chillerji), ki črpajo morsko vodo (moč delovanja, temperatura morske vode, količina porabljene energije ...).

### 3.1 Delo s signali

Tako velik projekt, kot je ta, vsebuje veliko količino signalov različnega tipa. Signali so, kot sem že omenil, razporejeni v različne bloke po različnih krmilnikih. Prava izguba časa bi bila, da bi vsakega od njih vnašali v programsko okolje ročno, saj govorimo o nekaj tisočih signalih. Vsak od njih vsebuje tudi različne lastnosti in vrednosti skaliranja. Pomembno je, da si znamo olajšati svoje delo in tako časovno ne zaostajamo za rokom projekta in si ne privoščimo večjih napak. Program Intouch 2014 ima opcijo uvoza oz. izvoza spremenljivk. V Intouch Managerju, ki služi kot pregled nad obstoječimi aplikacijami, sta temu namenjeni dve ikoni, in sicer DBLoad, ki služi uvozu signalov, in DBDump, katerega naloga je izvoz signalov iz aktualnega projekta. Prav zaradi tega se vnaprej pripravijo posebne datoteke, ki nosijo skupno ime – datoteke namenjene uvozu signalov. Vsaka datoteka za uvoz mora biti tvorjena pravilno, sicer programsko okolje Intouch ne pusti uvoza in javi napako.

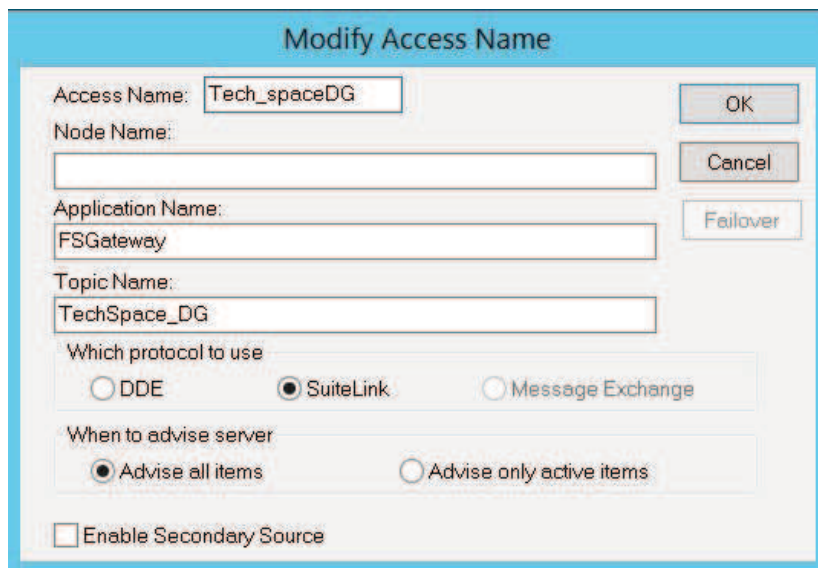
Stvari sem se lotil na sledeč način: moja naloga je bila, da zajamem vse tipe spremenljivk, torej spremenljivke tipa Bool, Integer in Real. Za vsakega od njih skrbi orodna vrstica, pod katero se nahajajo signali istega tipa. Uporabljal sem program PLK in v njem poiskal ustrezne signale v krmilnikovih blokih, poiskal in seštel začetne vrednosti blokovnih struktur in njihovih elementov ter izpisal ime signala skupaj z njegovim naslovom. Tvoril sem Excel datoteko in v njej t. i. Alias tabelo, v kateri se nahajajo oznaka signala v PLK in v sistemu SCADA in njegov naslov. Alias tabela je tabela, ki služi popisu signalov. Običajno se



držimo pravila, da se oznaki signala v PLK in sistemu SCADA ujemata zaradi lažjega sledenja. Čeprav za delovanje programa in za njegovo funkcionalnost ime signala ni pomembno, temveč je pomemben pravilen naslov v krmilniku.

Že v razdelku 2.2.2 sem omenil, da je krmilnik prepoznaven po svojem unikatnem naslovu. Vsak krmilnik ima svojo identifikacijo, po kateri ga ločimo od ostalih. Identifikacijski naslovi so po svoji obliki sledeči in potekajo od S7\_Connection\_1 do S7\_Connection\_11. Predpona S7\_Connection pomeni, da gre za Siemensovo povezavo, zadnja številka pa predstavlja številko krmilnika v omrežju, kateremu pripada ta povezava.

Zelo pomembno za pravilno delovanje sistema kot celote je tvorjenje dostopnih imen v razvojnem okolju. V sistemu SCADA pripada vsakemu krmilniku tudi dostopno ime. Slika 3.1 prikazuje primer dostopnega imena (Access Name), skupaj s pripadajočim protokolom. Ta dostopna imena lahko vidimo tudi v konzoli SMC, vsakemu dostopnemu imenu pripada tudi ime povezave.



Slika 3.1: Dostopno ime v Intouchu

Dostopno ime služi namenu dostopanja do nadzornega sistema skupaj z imenom povezave. Dostopna imena so bila tvorjena na podlagi dokumenta oz. spe-

cifikacij, ki smo jih predhodno dobili od določenega podjetja. Imena vsebujejo za krmilnike po ladijskih conah predpono HVAC, za tiste, ki se nahajajo v strojnici, pa predpono TechSpace. Kot sem že omenil na začetku podpoglavja, je v programu zelo veliko krmilniških blokov, ki so pod enakimi imeni razporejeni po različnih krmilnikih. Z uporabo dostopnih imen strežnik OPC točno ve, katere signale mora posredovati na določen zaslon nadzornega sistema.

Datoteke za uvoz so bile zaradi lažjega spremljanja in kasnejše podpore tvorjene za vsak krmilnik posebej. V svojem imenu morajo vsebovati končnico .CSV, saj Intouch 2014 za uvoz podpira to vrsto datoteke. Vsakemu zaslonu oz. tisti skupini zaslonov, ki se je na določen krmilnik navezovala, je pripadala svoja datoteka za uvoz. Pod orodno vrstico, ki sporoča tip spremenljivke, vnesemo signale, skupaj z njihovim naslovom. S pomočjo tega bo strežnik vedel, na katerem krmilniku se nahajajo signali, ki se prikazujejo. Pomembno je tudi, da v tej datoteki definiramo minimalno in maksimalno vrednost izpisa ter vrednost skaliranja za spremenljivke tipa Real, če je to potrebno. Le tako se bodo na zaslonu izpisovale ustrezne željene vrednosti, ki jih pridobi nadzorni sistem iz določenega krmilnika. Datoteke namenjene uvozu pri projektih vzamejo največ časa, vendar pa so zelo uporabne za lažje nadaljnje delo, ker lahko vse popravke opravimo v tej datoteki.

Na sliki 3.2 je prikazan primer ene od datotek za uvoz. Tukaj vidimo vrstice, ki ponazarjajo tip spremenljivk (IODisc in IOInt), imena signalov v sistemu SCADA, komentar, ki olajša sledenje signalom, njihove enote, minimalno in maksimalno vrednost itd.

```

:mode=update
:IODisc;Group;Comment;Logged;EventLogged;EventLoggingPriority;RetentiveValue;InitialDisc;OffMsg;OnMsg;AlarmState;AlarmP
"Eng6_Eng_run_status";"TechSpace";"Engine 1 Run Status";No;No;0;No;Off;"FALSE";"";None;1;Direct;"Tech_spaceDG";No;"S7:[8
"Eng1_Eng_run_status";"TechSpace";"Engine 1 Run Status";No;No;0;No;Off;"FALSE";"";None;1;Direct;"Tech_spaceDG";No;"S7:[8
"Eng5_Eng_run_status";"TechSpace";"Engine 1 Run Status";No;No;0;No;Off;"FALSE";"";None;1;Direct;"Tech_spaceDG";No;"S7:[8
"Eng4_Eng_run_status";"TechSpace";"Engine 1 Run Status";No;No;0;No;Off;"FALSE";"";None;1;Direct;"Tech_spaceDG";No;"S7:[8
"Eng3_Eng_run_status";"TechSpace";"Engine 1 Run Status";No;No;0;No;Off;"FALSE";"";None;1;Direct;"Tech_spaceDG";No;"S7:[8
"Eng2_Eng_run_status";"TechSpace";"Engine 1 Run Status";No;No;0;No;Off;"FALSE";"";None;1;Direct;"Tech_spaceDG";No;"S7:[8
"CRUISE_SBB_PORT";"TechSpace";"";No;No;0;No;Off;"FALSE";"TRUE";None;1;Direct;"Tech_spaceDG";No;"S7:[87_Connection_10]DB2
"CRUISE_FAOP_OUT";"$System";"";No;No;0;No;Off;"FALSE";"TRUE";None;1;Direct;"Tech_spaceDG";No;"S7:[87_Connection_10]DB2100
"FINISH_CRUISE";"$System";"";No;No;0;No;Off;"FALSE";"TRUE";None;1;Direct;"Tech_spaceDG";No;"S7:[87_Connection_10]DB2100
"FINISH_SEGMENT";"$System";"";No;No;0;No;Off;"FALSE";"TRUE";None;1;Direct;"Tech_spaceDG";No;"S7:[87_Connection_10]DB2100
"CRUISE_FWE_IN";"$System";"";No;No;0;No;Off;"FALSE";"TRUE";None;1;Direct;"Tech_spaceDG";No;"S7:[87_Connection_10]DB2100
"CRUISE_SBB_PORT";"TechSpace";"";No;No;0;No;Off;"FALSE";"TRUE";None;1;Direct;"Tech_spaceDG";No;"S7:[87_Connection_10]DB2
"CRUISE_SBB_SEA";"TechSpace";"";No;No;0;No;Off;"FALSE";"TRUE";None;1;Direct;"Tech_spaceDG";No;"S7:[87_Connection_10]DB2
"START_SEGMENT";"$System";"";No;No;0;No;Off;"FALSE";"TRUE";None;1;Direct;"Tech_spaceDG";No;"S7:[87_Connection_10]DB2100
:MemoryInt;Group;Comment;Logged;EventLogged;EventLoggingPriority;RetentiveValue;RetentiveAlarmParameters;AlarmValueDeadba
:IOInt;Group;Comment;Logged;EventLogged;EventLoggingPriority;RetentiveValue;RetentiveAlarmParameters;AlarmValueDeadband;
"Eng4_Fuel_oil_cons";"TechSpace";"Engine 1 Fuel Oil Consumption Real Time";No;No;0;No;No;0;0;"g/kWh";0;0;10000;0;0;Off;0
"Eng4_Energy_total_KWh";"TechSpace";"Energy Meter Energy total";No;No;0;No;No;0;0;"kWh";0;0;1000000000;0;0;Off;0;1;Off;0
"Eng6_Line_volt_V12";"TechSpace";"Energy Meter Line Voltage V1";No;No;0;No;No;0;0;"V";0;0;10000;0;0;Off;0;1;Off;0;1;Off;0
"Eng3_Diff_mass_total_kg";"TechSpace";"Flow Meter Return Differential total";No;No;0;No;No;0;0;"kg";0;0;1000000000;0;0;0
"Eng3_Diff_mass_flow_return";"TechSpace";"Flow Meter Return Differential Flow";No;No;0;No;No;0;0;"kg/h";0;0;1000000000;0;0
"Eng1_One_hour_aver_SF0C";"TechSpace";"Engine 1 Specific Fuel Oil Consumption 1 hour aver";No;No;0;No;No;0;0;"g/kWh";0;0
"Eng2_Sys_power_factor_Pfsum";"TechSpace";"Energy Meter Power factor";No;No;0;No;No;0;0;"";0;0;1000;0;0;Off;0;1;Off;0;1
"Eng6_Frequency";"TechSpace";"Energy Meter Frequency";No;No;0;No;No;0;0;"Hz";0;0;1000;0;0;Off;0;1;Off;0;1;Off;0;1;Off;0
"Eng2_Density_return";"TechSpace";"Flow Meter Return Density";No;No;0;No;No;0;0;"kg/m3";0;0;10000;0;0;Off;0;1;Off;0;1;Off;0
"Eng2_Temperature_return";"TechSpace";"Flow Meter Return Temperature";No;No;0;No;No;0;0;"°C";0;0;10000;0;0;Off;0;1;Off;0;1;Off;0
"Eng4_Mass_total_kg_return";"TechSpace";"Flow Meter Return Flow Total";No;No;0;No;No;0;0;"kg";0;0;1000000000;0;0;Off;0;1;Off;0
"Eng5_Spec_fuel_oil_cons";"TechSpace";"Engine 1 Specific Fuel Oil Consumption Real Time";No;No;0;No;No;0;0;"g/kWh";0;0;1
"Eng1_Volume_flow_rate";"TechSpace";"Flow Meter Inlet Fuel Flow";No;No;0;No;No;0;0;"m3/h";0;0;10000;0;0;Off;0;1;Off;0;1;Off;0
"Eng4_Aver_line_volt_V1avg";"TechSpace";"Energy Meter Average Voltage";No;No;0;No;No;0;0;"V";0;0;10000;0;0;Off;0;1;Off;0
"Eng3_Sys_power_factor_Pfsum";"TechSpace";"Energy Meter Power factor";No;No;0;No;No;0;0;"";0;0;1000;0;0;Off;0;1;Off;0;1;Off;0
"Eng5_Aver_curr_Iavg";"TechSpace";"Energy Meter Average Current";No;No;0;No;No;0;0;"A";0;0;10000;0;0;Off;0;1;Off;0;1;Off;0
"Eng2_Line_volt_V23";"TechSpace";"Energy Meter Line Voltage V2";No;No;0;No;No;0;0;"V";0;0;10000;0;0;Off;0;1;Off;0;1;Off;0
"Eng1_Temperature_inlet";"TechSpace";"Flow Meter Inlet Temperature";No;No;0;No;No;0;0;"°C";0;0;10000;0;0;Off;0;1;Off;0;1;Off;0
"Eng3_Aver_line_volt_V1avg";"TechSpace";"Energy Meter Average Voltage";No;No;0;No;No;0;0;"V";0;0;10000;0;0;Off;0;1;Off;0
"CRUISE_SEGMENT";"TechSpace";"Cruise segment number";No;No;0;No;No;0;0;"";0;0;100;0;0;Off;0;1;Off;0;1;Off;0;1;Off;0
"CRUISE_PHASE";"TechSpace";"Cruise segment phase";No;No;0;No;No;0;0;"";0;0;4;0;0;Off;0;1;Off;0;1;Off;0;1;Off;0;1;Off;0
"Amb_temp_eng_room_2";"TechSpace";"";No;No;0;No;No;0;0;"°C";0;0;100;0;0;Off;0;1;Off;0;1;Off;0;1;Off;0;1;Off;0;1;Off;0;1

```

Slika 3.2: Primer datoteke, namenjene uvozu signalov

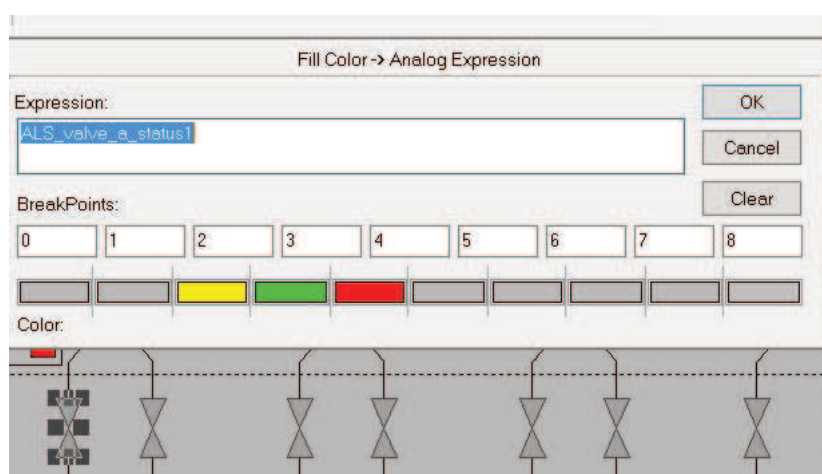
## 3.2 Animiranje

To poglavje se mi zdi zelo pomembno, zato ga bom opisal malo bolj podrobno. Vsak človek, ki se prvič sreča z nadzornim sistemom, opazi na zaslonih različno obarvane elemente, utripajoče lučke, izpisna polja itd. Vse te stvari so posledica animiranja signalov, ki smo jih predhodno s pomočjo datotek uvozili v programsko okolje. Za razumevanje naslednjih poglavij želim razložiti princip animiranja in prikazati nekaj primerov.

Za animiranje so pomembni obstoječi signali v aplikaciji. Potek uvažanja signalov zasledimo v podpoglavju 3.1. Torej če povzamem, pred animiranjem mora aplikacija vsebovati vse signale, ki jih za nadaljnje delo potrebujemo.

- Barvna animacija in njen primer: S paleta standardnih načrtovalskih elementov, ki se nahajajo na desni strani zaslona na sliki 2.1, izrišemo željen

element. Elemente lahko med sabo združimo ali ustvarimo iz njega simbol, če je želja razvijalca aplikacije uporabiti isti element z istimi animacijami na večih mestih v projektu. To sem v nekaterih delih projekta storil tudi sam, ker je veliko elementov enakih in vsebujejo enake animacije barvanja. Dvojni klik na željen element ponuja različne možnosti animiranja. Za animacijo barvanja uporabimo eno od možnosti pri zavihku Fill Color, glede na tip signala.



Slika 3.3: Primer barvne animacije

Okvirčku po imenu Expression na sliki 3.3 pripada željen signal. V spodnjem delu okna lahko uporabimo različne animacije glede na trenutno vrednost signala. Signal je v tem primeru tipa Integer, zato imamo štiri različne vrednosti barvanja, ki se izvedejo glede na prebrano vrednost iz krmilnika.

- Primer večkratnih animacij barvanja: Intouch ima opcijo združitve večih animacij, v tem primeru barvanja. Postopek je tak, da načrtamo več elementov, jim dodamo statične barve in nato vsakemu od njih priredimo spodaj opisano animacijo vidnosti – Visibility. Tako se ob sprožitvi določenega bita na krmilniku prikaže element, kateremu pripada ta določen signal. Naenkrat se lahko prikaže le en od načrtanih elementov.

- Animacija izpisa in njen primer:

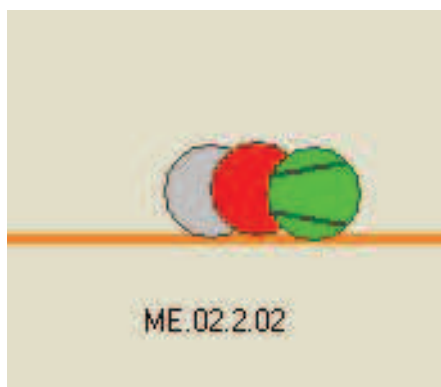
V sistemu SCADA se poleg barvnih animacij najbolj pogosto uporabljajo tudi animacije izpisa. Poznamo dve vrsti teh animacij: izpis celih števil in izpis števil z decimalno vejico. S tem je pogojena tudi animacija izpisa. Kot vidimo na sliki 3.4, je simbol zgrajen iz pravokotnika in štirih znakov izpisa, ki so ponazorjeni z znakom lojtre. V primeru realnega števila lahko znake izpisa ločimo s piko, kar pomeni, da se izpiše decimalno število. Tudi v tem primeru je to izpisno polje uporabljeno kot simbol, ki je zaradi istih lastnosti uporabljen na večih mestih v programu. Tudi tukaj je aktualen dvoklik na simbol, kjer izberemo ustrezno animacijo glede na tip željenega prikazovanega signala in ustrezen signal.



Slika 3.4: Primer animacije izpisa

- Animacija vidnosti in njen primer:

Animacije vidnosti se uporabljajo predvsem v primerih različnih statusov obratovanja sistema: ročno, avtomatsko, daljinsko, lokalno, za stanja alarmov ter za prikaz statičnih barv elementov, kar je prikazano na sliki 3.5. Princip je tak, da se ob postavitvi določenega bita v krmilniku na zaslonu prikaže le eden od statusov. Animacija se imenuje Visibility, kateri poleg imena signala dodamo še pogoj, ki predstavlja pojav željenega elementa. To pomeni, da se animacija lahko pojavi le v primeru, ko je pogoj izpolnjen.



Slika 3.5: Primer animacije vidnosti s statičnimi barvami

- Primer animacije z akcijsko skripto:

V veliko primerih razvijalcu aplikacije ni dovolj uporaba ene spremenljivke za animacijo, saj mora pri njej upoštevati tudi določene pogoje (kdaj se naj spremenljivka izvede itd.). V tem primeru program Intouch omogoča možnost implementacije akcijskega skripta. Akcijski skript se nahaja med standardnimi animacijami določenega objekta. Tukaj lahko sestavljamo pogoje, priredimo začetne vrednosti spremenljivk ... Za implementacijo akcijskih skriptov se uporablja programski jezik C.

### 3.3 Urniki in zmanjšana poraba energije

Urniki so odgovorni za klimatske sisteme. Klimatski sistem deluje na principu dobave ohlajene morske vode, ki skrbi za hlajenje po ladji. Voda se ohladi v štirih HVAC-enotah, ki jim pravimo Chillerji. Obstajajo torej štiri ločene hladilne klimatske naprave in vsaka ima svojo lastno hladilno črpalko za morsko vodo. Vsaka črpalka se izprazni v lastno hladilno enoto za klimatizacijo, nato morsko vodo potiska preko enega od dveh čezmorskih izpustnih priključkov. Ti nato krožijo vodo



skozi hladilne naprave in nato v omare za klimatizacijo, ki se nahajajo v bivalnih prostorih. V kabinah in ostalih prostorih so zagotovljene kontrole, da se doseže željena temperatura. Lokalni temperaturni regulator nastavi hitrost ventilatorja, zunanja temperatura pa določa hladilno obremenitev klimatske naprave.

### 3.3.1 Pregled tabele

The screenshot shows a software interface titled "Daily schedule" with buttons for "Edit", "New", and "Delete". Below is a table with columns for ID, Item, Description, Enable, and 24 hourly slots (u0000 to u2300). The table lists various ship areas and their scheduled energy consumption status (1 for active, 0 for inactive).

ID	Item	Description	Enable	u0000	u0015	u0030	u0045	u0100	u0115	u0130	u0145	u0200	u0215	u0230	u0245	u0300	u0315	u0330	u0345	u0400	u0415	u0430
74	AP09101ES	Dk6 FZ1 - Crew Bar, Gymnasium	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
75	AS09101ES	Dk6 FZ1 - Workshops	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
76	AT09101ES	Dk3 - 15 FZ1 - Stairs & Lift Machinery	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
77	AP08201ES	Dk6-7 FZ2 - Main Theatre PS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
78	AP08203ES	Dk6 FZ2 - Computer Rm, Offices & Conference Rm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	AP08204ES	Dk6-7 FZ2 - Main Theatre SB	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
80	AP08205ES	Dk5-7 FZ2 - Back Stage Electrical locker	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
81	AP17201ES	Dk15-17 FZ2 - Thermal Suite, Massage Rms, Gym Iron	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
82	AP17202ES	Dk15-16 FZ2 Gymnasium & Beauty Saloon	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
83	AP08301ES	Dk6 FZ3 - Casino PS & Smoking area	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	AP08302ES	Dk5-6 FZ3 Casino SB, Offices, Art Gallery - DK5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	AP08303ES	Dk7 FZ3 Library, Wheelhouse, Shop & Dk16 Wedding	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
86	AT17301ES	Dk2-16 FZ3 Stairs & Lift Machinery Dk17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
87	AP08401ES	Dk5 FZ4 - Bar, Vines, Internet & Foyer - DK5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
88	AP08402ES	Dk6-7 Shops SB, Foyer - DK6,7, lounge Bar Dk7 PS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
89	AS08401ES	Dk2-4 FZ4 - Valet Rm, Embarkation Dk4, stairs dk3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
90	AT08401ES	Dk2-15 FZ4 - Main Stairs, Lift Machinery Dk16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
91	AP14401ES	Dk8-15 - Lift Entrance & Pizza oven Exhaust	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
92	AP08501ES	Dk7 FZ5 - Explorer Lounge & Arcade	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
93	AP08502ES	Dk5-6 FZ5 - Lower / Upper Restaurant PORTside	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
94	AP08503ES	Dk5-6 FZ5 - Lower / Upper Restaurant STBside	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
95	AG08601ES	Dk6 FZ6 Galley Scullery & Fwd Pastry	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
96	AG08602ES	Dk6 FZ6 Main Hot Galley PORT, Dish Wash SB	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
97	AG08603ES	Dk6 FZ6 Main Hot Galley STBD, Wine Station SB	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
98	AG08604ES	Dk6 FZ6 Main Hot Galley STBD, Garde Manger & Dish	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Slika 3.6: Tabelni pregled urnikov

Tabela na sliki 3.6 je zgrajena iz:

- identifikacijske oznake sobe in njihovega opisa (npr. dk8 FZ1 – Crew Bar & Gymnasium),
- statusa aktivacije urnika,
- časovnega poteka izvajanja urnikov. Urniki se izvajajo na vsakih 15, 30 ali 45 minut, kar je odvisno od položaja ladje (ali je na morju/v pristanišču). Urnik je v fazi izvajanja, ko se v tabeli ob izbranem prostoru ter časovnem obdobju izpiše število 1.

Napis na sliki 3.6 "Daily schedule" omogoča izbiro delovanja urnikov za različna časovna obdobja in glede na položaj ladje, ali se nahaja na morju ali

v pristanišču. Urniki se največkrat izvajajo ponoči, ker lahko posadka najbolj prihrani pri porabi energije. To je od 23.00 do 8.00 ure zjutraj, kar prikazuje slika 3.7. Takrat seveda večina potnikov spi in ne opazijo nikakršne moteče spremembe. Urnik je v fazi delovanja, ko je kvadratega za izbiro časovnega obdobja obarvan modro. Zgornja kontrola omogoča tudi dodajanje novih prostorov oz. brisanje starih. Za to se uporablja Microsoftova tabela, ki bo predstavljena v nadaljevanju. Če hočemo, da se urnik izvaja avtomatsko za določene čase, je obvezen stisk gumba za potrditev izbire. Le tako shranimo željene spremembe, za katere želimo, da postanejo aktivne. Program omogoča tudi ročno izvajanje urnikov. Izvajanje urnikov je sicer pogojeno z nivojem varnosti, vendar se ob ustreznem nivoju lahko odločimo za določen prostor na potniški ladji in na željenem zaslonu vključimo način varčevanja z energijo.

Konzola je narejena na tak način, da operaterju omogoča tudi izbiro ostalih prostorov po različnih conah ladje. Tako je uporabniku olajšano delo, saj ne izgublja časa z vračanjem na osnovno okno in z iskanjem novega, željenega prostora, kateremu bi priredili način varčevanja z energijo.

ID	87	
Item	AP08401ES	→
Description	Dk5 FZ4 - Bar, Vines, Internet & Foyer - DK6	
Function	X Enable	
Sea day		
0:00 1:00 2:00 3:00 4:00 5:00 6:00 7:00 8:00 9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00 24:00		
[Blue bar from 0:00 to 8:00] [Yellow bar from 8:00 to 24:00]		
Dock day		
0:00 1:00 2:00 3:00 4:00 5:00 6:00 7:00 8:00 9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00 24:00		
[Blue bar from 0:00 to 8:00] [Yellow bar from 8:00 to 24:00]		
		Confirm Cancel

Slika 3.7: Izbira časovnega delovanja urnikov

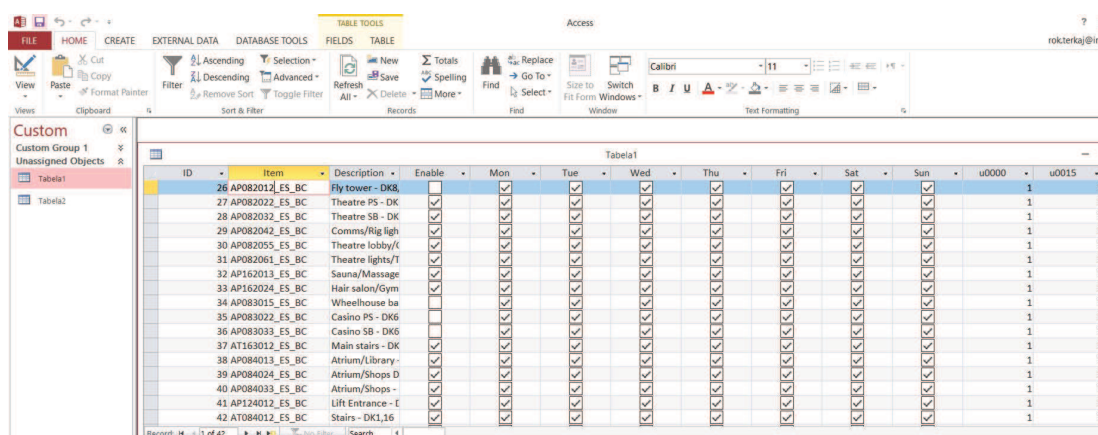


### 3.3.2 Delovanje urnikov

Delovanje urnikov je pogojeno z .mdb datoteko, ki bo predstavljena pod sliko 3.8. Za potrebe komunikacije s krmilniki je bila potrebna uporaba ustreznega gonilnika, zato tudi pri tukajšnji komunikaciji potrebujemo gonilnik za dostop do Microsoftove podatkovne baze. Potrebujemo ga zaradi komunikacije z bazo.

Sistem Microsoft Access je programsko orodje za upravljanje z bazami podatkov. Vsebuje grafični uporabniški vmesnik in orodja za razvoj programske opreme. Vključen je v Microsoft Office, in sicer ob njegovi inštalaciji. Podatke shrani v svoji lastni obliki, lahko tudi uvozi ali pa se celo poveže neposredno s podatki, shranjenimi v drugih aplikacijah in bazah. Podatki se tudi samodejno shranjujejo, in sicer v zbirko podatkov strežnika SQL [7].

Gonilnik je potrebno dodati v programsko orodje ODBC Data Source pod njegovim zavirkom System DSN, ki se nahaja na strežniku, na virtualnem okolju z nadzornim sistemom. Vmesnik ODBC je vmesnik programskega jezika C, ki omogoča dostop različnim aplikacijam do podatkov različnih sistemov za upravljanje z bazami podatkov. Je vmesnik visoke zmogljivosti [8]. Tukaj izberemo ime baze s podatki, obvezna je tudi izbira prave poti, ki kaže na izbrano bazo oz. njeno datoteko: Urniki.mdb.



ID	Item	Description	Enable	Mon	Tue	Wed	Thu	Fri	Sat	Sun	u0000	u0015
26	AP082012_ES_BC	Fly tower - DK8									1	1
27	AP082022_ES_BC	Theatre PS - DK									1	1
28	AP082032_ES_BC	Theatre SB - DK									1	1
29	AP082042_ES_BC	Comms/Rig high									1	1
30	AP082055_ES_BC	Theatre lobby/t									1	1
31	AP082061_ES_BC	Theatre lights/t									1	1
32	AP162013_ES_BC	Sauna/Massage									1	1
33	AP162024_ES_BC	Hair salon/Gym									1	1
34	AP083015_ES_BC	Wheelhouse ba									1	1
35	AP083022_ES_BC	Casino PS - DK6									1	1
36	AP083033_ES_BC	Casino SB - DK6									1	1
37	AT163012_ES_BC	Main stairs - DK									1	1
38	AP084013_ES_BC	Atrium/Library									1	1
39	AP084024_ES_BC	Atrium/Shops D									1	1
40	AP084033_ES_BC	Atrium/Shops									1	1
41	AP124012_ES_BC	Lift Entrance - E									1	1
42	AT084012_ES_BC	Stairs - DK1,16									1	1

Slika 3.8: Tabela v Microsoft Access

Tabela na sliki 3.8 je zgrajena na enak način kot v sistemu SCADA. Sestavljena je iz vrstic in stolpcev, ki predstavljajo identifikacije klimatov, spremenljivke in časovna območja. Če želimo dodati nov klimat oz. nov sistem, dodamo novo vrstico in jo izpolnimo z ustreznimi željenimi podatki. Poleg tega je potrebno še v nadzornem sistemu dodati novo spremenljivko, saj ta ostane v tej datoteki in napolni tabelo ob zagonu sistema SCADA.

Za pravilno delovanje urnikov moramo poleg tabele tudi v nadzornem sistemu nastaviti elemente, ki skrbijo za medsebojno povezavo. V tabeli na sliki 3.6 moramo obvezno vnesti ime baze, da sta datoteka Urniki.mdb in tabela na sliki med seboj povezani. Tukaj so nastavljeni tudi: časovna obdobja, dnevi v tednu, imena spremenljivk in njihov opis. Ob kliku na posamezen gumb zavihka "Daily schedule" se v ozadju izvajajo skripti s stavki SQLSelect (povedo nam, na katere elemente datoteke naj se tabela poveže) in stavki SQLConnect (povedo nam, s katero od dveh podanih tabel naj nadzorni sistem komunicira). S stavki jezika SQL sporočimo, iz katere tabele v zgoraj omenjeni datoteki .mdb se naj berejo podatki in seveda kateri. Vključeni sta dve tabeli, in sicer eno za časovno in eno za dnevno izvajanje urnikov. Za obe tabeli sta torej namenjena dva skripta, ker se v obeh nahajajo in izvajajo različni tipi podatkov. Vsak gumb ima torej svoj akcijski skript z določeno funkcijo oz. nalogo.

## 3.4 Poraba goriva in poročila

### 3.4.1 Vrste goriva in GT

- Heavy Fuel Oil (HFO) je težko gorivo, ki je dovedeno do dizelskih generatorjev pod pritiskom električnih črpalk. Uporablja se na odprtem morju, torej v območju oddaljenem od pristanišč. Ladja ima štiri dizelske generatorje in dva ločena sistema za gorivo, ki sta identična. En sistem je odgovoren za generatorja 1 in 3, drugi sistem pa je odgovoren za generatorja 2 in 4.

- Marine Gas Oil (MGO) je lažje dizelsko gorivo, ki se uporablja v bližini pristanišč oz obale. To pove, da je bolj ekološko od HFO-ja. Za dovajanje MGO-ja velja ista zadeva kot pri HFO-ju. Za razliko od HFO-ja, kjer sta na ladji dva rezervoarja za gorivo je za MGO le eden.
- GT: Predstavlja turbino za gorivo, ki za svoje delovanje ves čas uporablja gorivo MGO. Njena naloga je, da dovaja vroče gorivo k močnostni turbini. GT na ladji uporablja dva rezervoarja za gorivo. Tudi tukaj priteka gorivo h generatorju s pomočjo električnih črpalk.

Za izbiro vrste goriva je odgovoren trosmerni ventil. Torej gorivo ena od dveh črpalk črpa iz rezervoarja in ga potiska preko ventilov, katerih naloga je skrb za regulacijo temperature, pritiska in vlažnosti, do generatorja po ceveh namenjenih za gorivo. Električni črpalke sta sicer dve, vendar naenkrat deluje le ena od teh, druga je v stanju pripravljenosti v primeru nedelovanja prve. Gorivo se nato segreje do ustrezne temperature, ki je primerna za delovanje sistema, za kar so odgovorni njegovi grelniki. Na ceveh imamo tudi merilnike pretoka, ki skrbijo za zbiranje podatkov (opis sledi v naslednjem razdelku). Pred generatorji se nahajajo tudi ventili za regulacijo tlaka, katerih naloga je, da gorivo potuje pod konstantnim tlakom. Omembe vreden je tudi sistem za dovajanje goriva, ki se tako kot ventili nahaja pred dizelskimi generatorji (DG-ji) in skrbi za časovno kontrolirano prepuščanje goriva. Poleg sistema za dovajanje goriva imamo tudi sistem za dovajanje vode. Pomemben je za ohranjanje primerne temperature goriva pred uporabo. Za dovajanje vode so odgovorne črpalke z imenom Chillerji. Dovajajo seveda morsko vodo. Dizelski generatorji so odgovorni za napajanje električnih generatorjev, ki proizvajajo elektriko za motorje, strojnico in hotelske storitve po ladji.

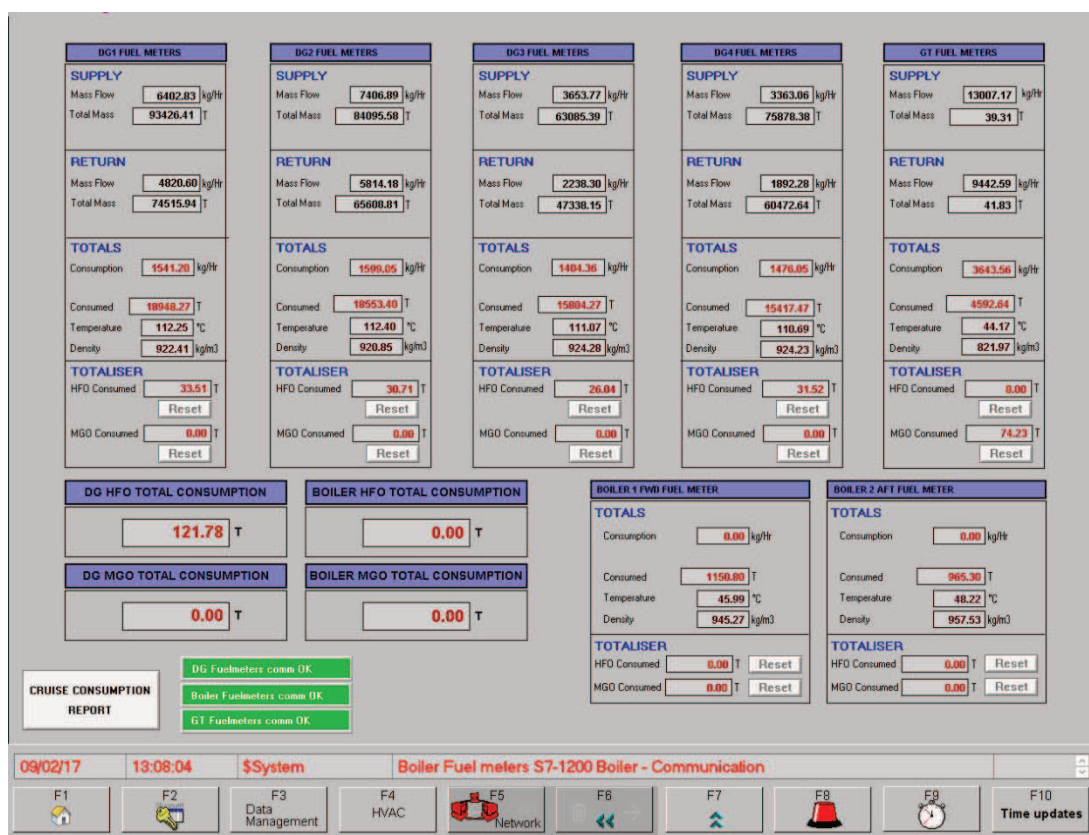
### 3.4.2 Merjenje porabe goriva

Na zaslonu Fuel Meter Data se beležijo podatki o porabi goriva, njegovi temperaturi, gostoti itd., za katere skrbijo merilniki, omenjeni v razdelku 3.4.1. Zaslon je zgrajen iz štirih DG-jev, ki predstavljajo dizelske generatorje in ene turbine za gorivo (GT).

Na zaslonu opazimo izpisna polja, na katerih se prikazujejo podatki iz krmilnika za vsak dizelski generator posebej.

- Supply: Zgornji del slike 3.9, razdelek imenovan Supply predstavlja količino goriva, ki je bila dovedena v dizelski generator (DG). Mass Flow predstavlja količino trenutno dovajanega goriva, Total Mass pa skupno količino dovedenega goriva.
- Return: Prikazuje količino neizgorenega goriva v DG. Ponovno imamo podatke za že zgoraj omenjena Mass Flow (v tem primeru Return Flow in Total Mass).
- Totals: Podatki se v krmilniku preračunavajo iz dveh zavihkov. Imamo 4 izpisna polja, ki predstavljajo: Consumption predstavlja trenutno porabo goriva, Consumed pomeni količino že porabljenega goriva, Density vsebuje podatek o gostoti goriva in Temperature, ki vsebuje podatek o temperaturi goriva.
- Totaliser: Vsebuje podatke o količini že porabljenega goriva HFO ali MGO.

Na spodnji strani zaslona imamo števec, ki prikazujejo skupno količino porabljenih goriv. Delovanje tega zaslona je pogojeno s statusi komunikacije z DG-ji, Boilerji in GT-jem in je osnova za tvorjenje poročil. Statusi uspešne povezave so vidni na sliki 3.9, ki so obarvani zeleno in so animirani na animacijo vidnosti. Med delovanjem sistema SCADA lahko tako administrator kot operater ponastavi porabo goriva s klikom na gumb za ponastavitev podatkov. Več o varnosti sledi v podpoglavju 3.5.



Slika 3.9: Zaslon namenjen prikazovanju porabe goriva

### 3.4.3 Poročila porabe goriva in ladijska poročila

Poročila so zelo pomemben del ladijskega nadzornega sistema, ki so namenjena glavnemu inženirju, odgovornemu za ventilacijo in goriva, ter tudi ljudem na vrhovnih položajih ladijskih združb. Namenjena so pregledu nad porabo vrste posameznega goriva. Tako se lahko vodi evidenco nad količino porabljenega MGO-ja in HFO-ja. Ladja namreč porabi več kot tisoč litrov goriva v eni uri, torej si predstavljate, kaj bi se zgodilo, če na koncu dneva, meseca, leta ne bi imeli zabeležene celotne vsote porabljenega goriva iz vidika rezervoarjev za gorivo.

Tvorjenju in definiciji poročil služi zaslon Cruise Consumption Report, ki je prikazan na sliki 3.10. Sestavljen je iz izpisnih polj, ki prikazujejo trenutno porabo goriva za vsak dizelski generator (DG) in turbino za gorivo (GT). Pomemben

podatek predstavlja vpisno polje HFO/MGO Treshold Temperature, glede na vrednost tega podatka sistem ve, ali gre za gorivo HFO ali MGO. Če vpisana temperatura znaša nad 50 °C sistem ve, da gre za štetje goriva HFO, v nasprotnem primeru uporablja gorivo MGO. Za pravilno delovanje in tvorjenje poročil, mora zaslon vsebovati podatke za temperaturo in podatke za Incinerator gas ter Fuel.

Na desni strani zaslona na sliki 3.10 imamo prikazano skupno porabo po odsekih, ki se imenuje Boiler + DG Total. Leva stran zaslona je sestavljena iz štirih gumbov, ki predstavljajo:

- SBB-PORT: Faza se prične beležiti, ko je ladja zasidrana in čaka na izkrcaje oz. vkrcanje. Takrat ne delujejo vsi motorji, zato se poraba ne beleži za vse DG-je in GT.
- FAOP-OUT: Beleženje se prične ob ladjinem izplutju. Tukaj se podatki beležijo za vse štiri dizelske generatorje in oba boilerja, porabe so tudi večje.
- SBB-SEA: Predstavlja fazo beleženje, ko je ladja na morju oz. glavni del križarjenja. Tudi tukaj velja isto kot v prejšnji fazi, porabe so dosti večje, kot v predhodnih korakih.
- FWE-IN: Predstavlja zadnjo fazo in sicer fazo vračanja ladje na mesto priveza. Tukaj so porabe manjše kot v prejšnjem koraku, saj se za pot nazaj v bližini pristanišč uporablja gorivo MGO, tako se tudi ladja pripravlja na fazo priveza in posledično ne delujejo vsi DG-ji.

Zgoraj je bil predstavljen kratek opis zaslona, sledi predstavitev tvorjenja poročil. Zgornje štiri faze predstavljajo tudi že omenjeni štirje zaslonski gumbi, ki so prikazani na levi strani slike 3.10. Ob kliku na prvi gumb sistem nudi tvorjenje imena dokumenta in možnost pričetka izvajanja prve faze. Slednja traja tako dolgo skupaj z beleženjem podatkov dokler uporabnik ne klikne na naslednji gumb in potrdi zaključitev tekoče faze. Sistem je narejen na tak način, da operaterju

Cruise No.  Segment No.  HFO/MGO Threshold Temperature  °C

Filename

**SHIP STATUS**

SBB - PORT ☐ FAOP - OUT ☒ SBB - SEA ☐ FWE - IN ☐

	DG 1	DG 2	DG 3	DG 4	GT	BOILER 1	BOILER 2	BOILER + DG TOTAL		FUEL LEFT	INCINERATOR	GAS LEFT	
HFO	0.02	0.02	0.02	0.02	0.00	0.01	0.00	0.08	T	HFO	9999.00	0.00	1600.00
MGO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	T	MGO	9999.00		
HFO	34.30	31.81	26.09	31.56	0.00	0.24	0.33	124.34	T	HFO	0.00	0.00	0.00
MGO	0.01	0.01	0.00	0.00	81.65	6.61	0.00	6.64	T	MGO	0.00		
HFO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	T	HFO	0.00	0.00	0.00
MGO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	T	MGO	0.00		
HFO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	T	HFO	0.00	0.00	0.00
MGO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	T	MGO	0.00		
HFO	0.07	0.07	0.05	0.04	0.00	0.01	0.00	0.24	T				
MGO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	T				

**TOTALS >**

HFO - CRUISE TOTAL FUEL  T

MGO - CRUISE TOTAL FUEL  T

CRUISE TOTAL GAS  T

Ensure Incinerator Values Are Entered Before Entering Ship Status

SBB - PORT - Stand By Below in Port  
FAOP - OUT - Full Away On Passage  
SBB - SEA - Stand By Below at Sea  
FWE - IN - Finish With Engines In Port  
ROB - Remaining On Board (Fuel)  
All Values in Metric Tonnes T

**KEY**

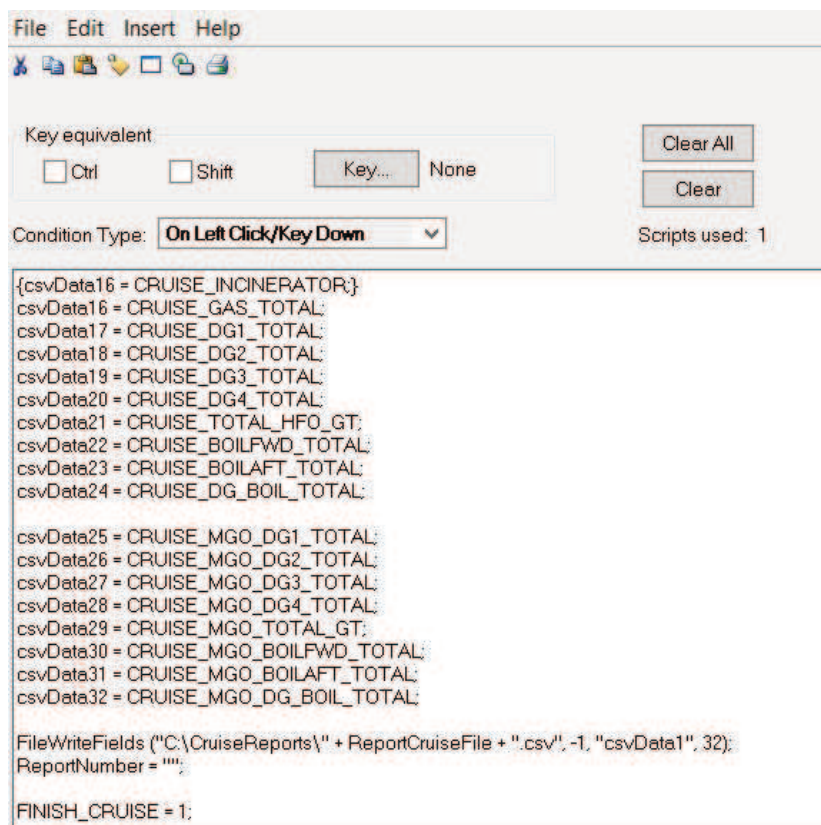
9999.99 Fuel Oil  
9999.99 Gas Oil  
9999.99 Incinerator (Manual Entry)  
● Phase Active

Slika 3.10: Zaslona za tvorjenje poročil porabe goriv

omogoča izbiro gumbov po točno določenem zaporedju, kot je prikazano na sliki 3.10. Ob kliku na ustrezno fazo križarjenja se torej pojavi okno s potrditvijo le-te in le tako sistem nadaljuje beleženje podatkov. Na koncu beleženja imamo na izbiro gumb FINISH SEGMENT, ki skrbi bodisi za zaključitev ene faze križarjenja ali celotnega križarjenja. Če izberemo možnost FINISH SEGMENT se zdajšnji zaključi in začne s štetjem novi. Podatki prejšnjih ostanejo na zaslonu, novi se le prištevajo. Če izberemo možnost FINISH CRUISE, se vsi podatki na določenih fazah zaključijo, vpišejo v dokument, shranijo in pobrišejo. Zaslona je prazen in pripravljen na ponovno uporabo.

Poročila so zgrajena tako, da se dejanske vrednosti signalov na zaslonu vpišejo v spremenljivke tipa Memory, ki se vpisujejo v .CSV dokument. Vsak od štirih gumbov vsebuje skript, ki opravlja nalogo zapisovanja podatkov skupaj z imenom faz križarjenja v vrstice .CSV dokumenta. Dokument je sestavljen iz orodne vrstice, v kateri se nahajajo imena podatkov, pod njimi pa so imena faz

križarjenja z realnimi vrednostmi podatkov.



Slika 3.11: Tvorjenje skripta za poročila

Slika 3.11 prikazuje skript za tvorjenje poročila v eni že omenjenih faz. Na desni strani imamo vidne spremenljivke tipa Memory, katerim so prirejene dejanske spremenljivke, ki predstavljajo signale. V spodnjem delu je določena pot shranjevanja ter ime dokumenta.

Ladijska poročila vsebujejo podatke o gorivih, porabi energije (dnevno, mesečno, letno), temperaturah v sobah, moči, hitrosti in navoru DG-jev. Zgrajena so na isti način kot poročila o porabi goriv. Razlika med njima je ta, da se ta izvajajo vsako uro, kar dosežemo s pisanjem v skript (QuickFunctions), ki se izvaja ves čas izvajanja aplikacije. Ladijska poročila so obsežnejša in nudijo dovolj informacij za temeljito spremljanje celotnega ladijskega sistema.



## 3.5 Varnost

Z uporabo varnosti v aplikaciji lahko dosežemo kontrolo nad specifičnimi funkcijami, ki jih lahko izvaja operater. To dosežemo s povezavo funkcij na signale.

Varnost in funkcionalnost sta pogojeni s prijavo uporabnika v aplikacijo, in sicer z vnosom njegovega uporabniškega imena in gesla. Naloga razvijalca aplikacije je, da skonfigurira uporabniško ime, geslo in nivo dostopa posameznega uporabnika v razvojnem okolju. Pomen zavarovanja Intouch aplikacije je različen za uporabnike: razvijalci aplikacij poskušajo zavarovati svoj operacijski sistem, akcije operaterjev, lahko pa seveda funkcije cele aplikacije. Razvijalec lahko na podlagi varnosti tudi sledi akcijam operaterja [1].

### 3.5.1 Varovanje operacijskega sistema

Veliko operaterjev mora za zaslonom, na katerem se nahaja nadzorni sistem, preživeti veliko časa. Zaželeno je, da imajo dostop izključno do te aplikacije. V Window Makerju lahko onemogočimo določene tipke, s katerimi lahko operater pride v sistem Windows: ALT, TAB in tipka Windows. Poleg teh tipk je onemogočen tudi upravitelj opravil, tako da ima operater zagotovo dostop le do okolja, namenjenega izvajanju aplikacij [1].

### 3.5.2 Vrste uporabnikov in varovanje aplikacije

Uporabnike razvijalec aplikacije skonfigurira v razvojnem okolju v že večkrat omenjenem Window Makerju. Uporabnike lahko skreiramo v zavihku zgornje orodne vrstice Special pod opcijo Security. Aplikacija vsebuje šest vrst uporabnikov (slika 3.12), vendar načeloma sta v uporabi le dve: Administrator in Consumption, ki predstavlja operaterja.

Administrator ima vse pravice in nivo varnosti 9999. Administratorski uporabnik je namenjen predvsem glavnemu inženirju odgovornemu za ventilacijo in

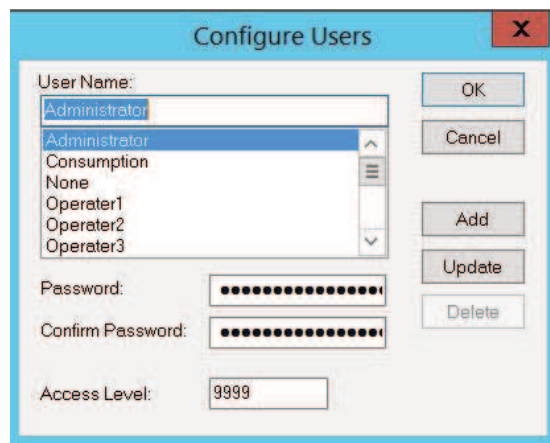
goriva. Omogočeno jim je:

- upravljanje z urniki, in sicer njihova sprožitev ter spreminjanje lastnosti,
- izhod iz aplikacije,
- ponastavljanje frekvenčnikov,
- ponastavljanje števecv porabe goriv,
- tvorjenje poročil.

Operater ima omejene pravice in nivo varnosti 5000. Ta uporabnik je namenjen predvsem operaterjem in ima omejene pravice:

Večinoma lahko le spremlja procese, ki se izvajajo na objektu. Nima dovoljenja za

- spreminjanje in pregled nad urniki,
- ponastavljanje števecv porabe energije.



Slika 3.12: Obstoječi uporabniki sistema SCADA

Seveda omejitev dostopnega nivoja ne pomaga kaj dosti, če želimo onemogočiti klike na določene gumbe aplikacije. Naloga vsakega razvijalca je, da to stori tudi na animaciji gumba oz. na animiranem signalu.

Uporabnik se v aplikacijo prijavi s klikom na gumb za prijavo. Odpre se mu okno, kjer mora vnesti svoje uporabniško ime in geslo – odvisno, ali gre za administratorskega ali operatorskega uporabnika. Gumb za odjavo mora biti na istem oknu, tam se tudi uporabnik odjavi ob končani uporabi aplikacije. V primeru napačnega uporabniškega imena oz. gesla nas aplikacija opozori. V primeru neprijave v aplikacijo, lahko na njej dogajanje le spremljamo brez akcij. Izhod iz aplikacije je onemogočen, zaradi tvorjenja ladijskih poročil, ker je njihovo izvajanje pogojeno s skriptom.

## 3.6 Časovni poteki signalov in alarmiranje

### 3.6.1 Časovni poteki signalov

Časovni poteki signalov so za uporabnike aplikacij zelo pomembni. Na njih se prikazujejo vrednosti izbranih signalov v odvisnosti od časa. Njihovi sestavni deli so tudi drsna polja, s kateri se lahko postavimo na točno določen čas, v katerem želimo opazovati izbran signal.

Za svoje delo sem uporabil standardni element iz orodne vrstice za načrtovanje elementov: Historical Trend. Izbral sem ga iz preprostega razloga: ker beleži tudi zgodovino oz. shranjuje vrednosti signalov v datoteke, ki se beležijo v mapo imenovano HstData. Operater ima s tem dano opcijo, da pregleda signale za željena časovna obdobja v preteklosti. Ker je kontrola ena in v projektu uporabljena večkrat, sem za svoje delo uporabil spremenljivke tipa Memory, katerim sem nato priredil dejanske signale. Vse to se zgodi v akcijskem skriptu, ki se izvede ob kliku na ustrezen gumb.

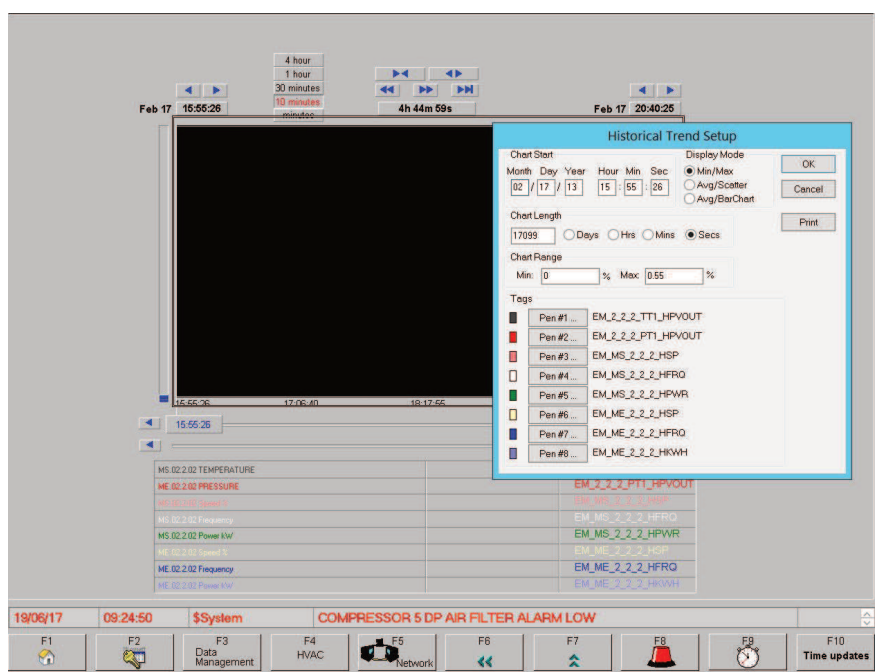
Na sliki 3.13 pa lahko opazimo sestavo okna za spremljanje signalov. Sestavljajo ga naslednji elementi:

- Drsna polja, s katerimi se pomikamo po določenih časovnih območjih. Čas lahko izberemo avtomatsko s klikom na gumb ali pa do njega pridemo ročno

z vpisom željene vrednosti.

- Spodnjo legendo, s katero izberemo željen opazovan signal. Na levi strani najdemo opis, na desni pa definicijo signala.
- Oknom prikaza časovnega poteka signalov: Tukaj lahko izberemo željeno časovno obdobje prikazovanja, zamenjamo željene opazovane signale, izberemo način prikaza ali spremenimo dolžino okna prikaza.

Signali se na kontroli izrisujejo le pod enim pogojem: Ob definiciji željenega signala je obvezna izbira opcije imenovane Log Data, s tem omogočimo shranjevanje signalov v datoteke. Na kontroli za časovno spremljanje signalov se večinoma prikazujejo signali, povezani s temperaturami, tlaki, močmi in frekvencami. Iz kontrolnega okna lahko takoj opazimo, če je opazovan signal v napačnem območju ali če kaj ne deluje v skladu s pričakovanji. Tako nas lahko operaterji tudi kontaktirajo za pomoč pri odpravljanju težav.



Slika 3.13: Primer okna za prikaz časovnega poteka

### 3.6.2 Alarmi in alarmna konzola

Naslednja zelo pomembna stvar, ki je osnovni gradnik vsakega nadzornega sistema, so alarmi. Predstavljajte si sistem brez alarmov: kako bi uporabnik vedel, kdaj je prišlo do alarma, njegovo lokacijo in kje ga odpraviti, kaj bi se lahko zgodilo z ladjo. Za vse to poskrbijo alarmi in alarmna konzola, ki vsebuje vse potrebne informacije o kraju, času in vzroku nastanka alarma.

Za prikaz alarmov je uporabljena osnovna alarmna konzola, ki sestoji iz:

- datuma, ki predstavlja datumski čas nastopa alarma,
- časa, ki predstavlja čas nastopa alarma,
- imena, ki ga definiramo ob definiciji spremenljivke,
- imena skupine, ki sporoča, kateri skupini določen alarm pripada,
- opisa, ki je namenjen opisu aktualnega alarma.

V konzoli določimo format izpisa datuma in časa, barvo aktualnih in potrjenih alarmov, kdaj naj se alarmi pojavijo (tako ob zagonu aplikacije) in zelo pomembno stvar, ki se imenuje Query. Slednji sporoča, iz katere definirane skupine naj se aktualni alarmi berejo ter s tem tudi beležijo na kontroli.

Signal, ki predstavlja alarm, definiramo v datoteki za uvoz. Alarmi so seveda tipa Bool, pazljivi moramo biti tudi na to, da alarmu določimo komentar, saj le tako uporabnik ve, za kakšen alarm gre in kje se nahaja, prioriteto pojava v alarmni konzoli ter skupino, kateri pripada. Na desni strani alarmne konzole imamo aktualen filter. Tukaj lahko s klikom na določen element filtra, ki predstavlja točno določeno cono, pridobimo aktualne alarme na njej in jih potrjujemo. Vse akcije operaterjev oz. alarmov se shranjujejo v zgodovino alarmov. Na tem mestu lahko uporabniki preverijo, če kakšen alarm nastopa pogosteje, ter si tudi ogledajo alarme po željenih časovnih obdobjih.

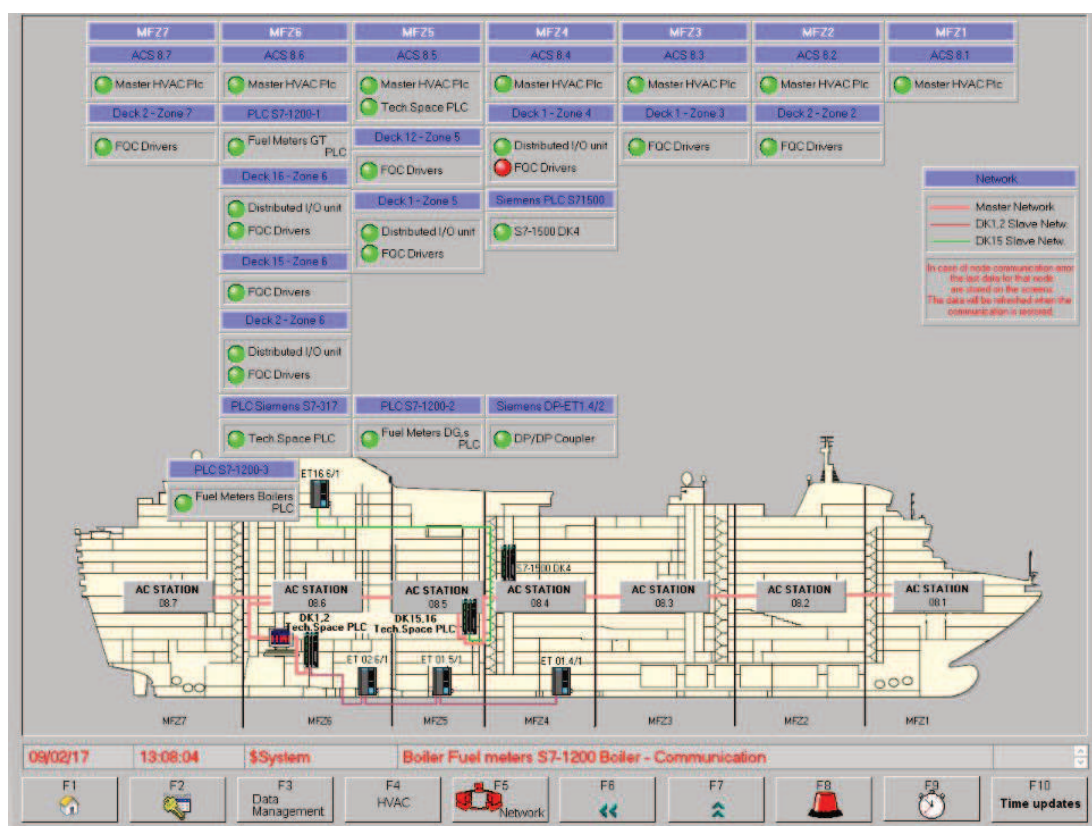
### 3.7 Diagnostika na nadzornem sistemu

V razdelku 2.2.3 je bilo govora o diagnostiki, ki se nahaja v konzoli SMC. Poznamo pa še eno vrsto diagnostike, ki se nahaja v nadzornem sistemu in daje uporabniku hiter in zanesljiv pogled nad delovanjem celotnega sistema.

Zaslon z diagnostiko sestoji iz celotne mreže na ladji – med seboj povezanih krmilnikov in frekvenčnikov. Slednji so med sabo povezani preko omrežja gospodarja in sužnja. Mreža prvega je označena z odebeljeno rdečkasto črto in predstavlja vse krmilnike ter strežnik, mreža drugega pa z zeleno in vijolično črto (glej sliko 3.13).

Na tem zaslonu prikazujemo statuse povezav. Če je povezava aktivna, lučke svetijo zeleno, napaka izgine. Ob slabem statusu napaka utripa, lučke seveda svetijo rdeče. Signali za statuse povezav so naslovljeni na strežnik OPC, preko katerega poteka vsa komunikacija. Če pride do prekinitve povezave, ostanejo na zaslonih zadnji aktualni podatki, ki se osvežijo, ko postane povezava ponovno aktivna. Ta zaslon je za uporabnika zelo pomemben, saj lahko takoj iz njegovih statusov opazi, kje tiči napaka in to napako poskuša odpraviti. Poleg utripajočih se statusov krmilnikov lahko uporabnik ob kliku na željen pravokotnik na zgornjem delu zaslona preveri tudi status obstoječih frekvenčnikov. Krmilnik lahko ponovno zažene ročno na lokaciji ali pa ponovno zažene skupino, ki pripada temu krmilniku v konzoli SMC.

Na sliki 3.14 lahko opazimo aktivno povezavo, saj zaslon ne vsebuje nikakršnih opozoril. Edino opozorilo dobimo na 4. coni, kjer je nedelujoč frekvenčnik, ki je obarvan z rdečo barvo. Ponazorjeno imamo omrežje gospodarja in sužnja s pripadajočo legendo.



Slika 3.14: Zaslon z diagnostiko





## 4 Sistem Silverstream

Sistem Silverstream predstavlja sistem vodnih mehurčkov (za lažjo predstavo glej sliko 4.1). Je nova, t. i. tehnologija "podmazovanja" z zrakom.

Podjetje Silverstream Technologies je razvilo edinstven način distribucije zraka pod plovilom: posebne enote za izpust zraka so ključni del za učinkovito delovanje sistema, ki se imenujejo zračne reže. Enote so izdelane iz močno odpornega jekla, ki kljubuje morskim pogojem [9]. Poskusi, ki jih je preverila skupina ladij združbe Lloyds, kažejo na to, da prihaja do 4.0 % prihranka energije pri plovilih pri obremenjenih razmerah. Testiranje in zbiranje podatkov je potekalo v optimalnih in obremenjenih zračnih tokovih. Ugotovitve so bile naslednje: popolnoma optimiziran sistem ima možnost stalnega zagotavljanja več kot 5.0 % prihrankov pri učinkovitosti in pri porabi energije. Želja podjetja Silverstream je, da doseže za potniške ladje njihov sistem pri polni učinkovitosti 10.0 % prihrankov pri porabljeni energiji [10].

Vse to se pozna pri porabi goriva in onesnaževanju okolja. Kot sem že omenil, ladja porabi več kot tisoč litrov goriva v eni uri. Naloga sistema vodnih mehurčkov je ta, da ladjo malo dvigne, kar daje občutek, da lebdi v vodi. Tako je upor med vodo in ladijskim trupom veliko manjši in seveda pri manjšem uporju je veliko manj zunanjih dejavnikov, ki povečujejo porabo goriva. Iz ekonomskega vidika je to za lastnike dobrodošla zadeva, saj tako v enem mesecu, letu ladja porabi občutno manj goriva kot pred prehodom na to tehnologijo in posledično zmanjša vsoto denarnih sredstev. Dobra stvar že omenjene tehnologije je ta, da jo lahko namestijo tako na nove kot stare ladje. Seveda je za stare ladje obvezen obisk

suhega priveza. Silverstream je tehnologija, ki se jo poslužujejo vse večje ladijske korporacije vključno s to, za katero dela podjetje INEA.



Slika 4.1: Delovanje sistema Silverstream

## 4.1 Ventili in kompresorji sistema Silverstream

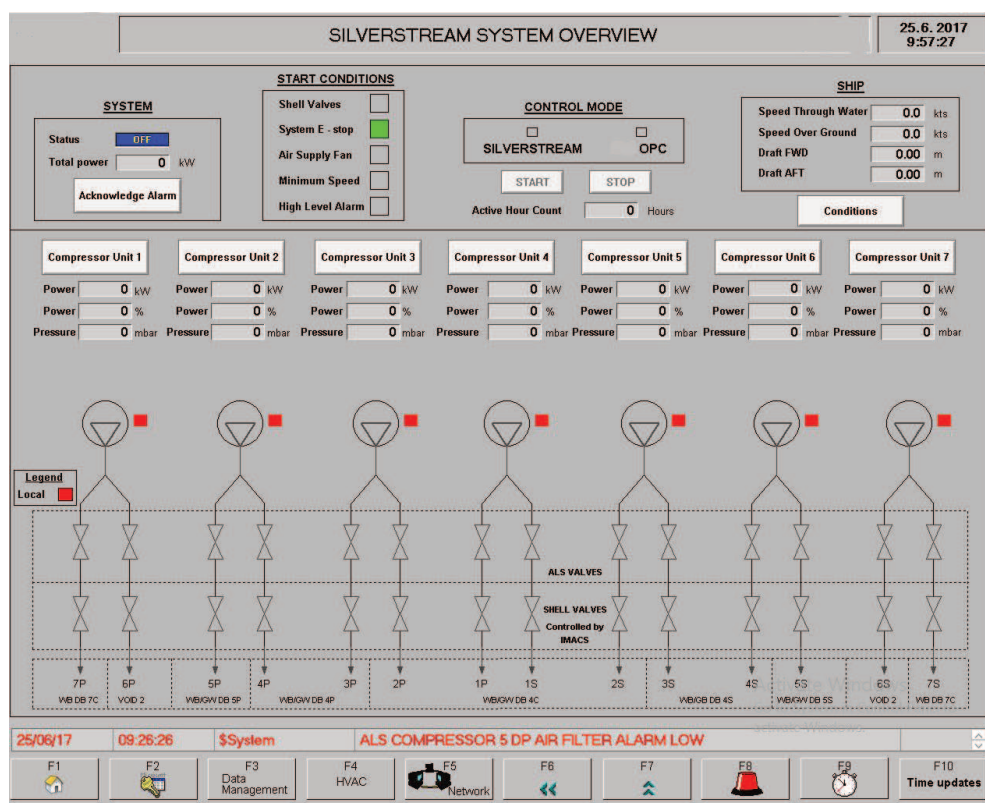
Sistem Silverstream je zgrajen iz 28 ventilov (14 ALS in 14 Shell) ter 7 kompresorjev, na katere so ti ventili povezani. Za delovanje ventilov ALS skrbi sistem, ki ga je razvilo podjetje INEA, za delovanje ventilov SHELL pa ladijski. Delovanje ventilov in kompresorjev je pogojeno z delovanjem ventilacijske sobe skupaj z dvema frekvenčnikoma, ki služita kot asinhronska motorja in skrbita za regulacijo zraka in tlaka.

Za regulacijo je namenjen PID-regulator, katerega člene lahko operater nastavi ročno ali izbere avtomatsko delovanje regulatorja, kjer so členi že zapisani v bloku krmilnika. Ročno lahko določimo tudi temperaturni in tlačni koeficient, ki sta pomembna za ustrezno delovanje PID-ja. Če kompresorji ne delujejo, se regulira hitrost glede na temperaturo in tlak. Če kompresorji delujejo, naraste

temperaturni frekvenčnik na 100 %, tlačni pa pade na njegov minimum, in sicer 50 %. Na zaslonu se beležijo podatki o delovanju frekvenčnikov: hitrosti in frekvenci delovanja, o trenutni moči in toku, temperaturi in statusu delovanja. Upravlja se ju lahko lokalno ali daljinsko. Delovanje sobe je omogočeno spremljati na pripadajoči kontroli za časovno spremljanje signalov. Slednja prikazuje odvisnost temperature, pritiska, hitrosti ter frekvence od časa. Ventilacijski sobi pripadajo tudi števeci porabe energije. Na zaslonu za beleženje porabe, se beležijo podatki za urno, dnevno, tedensko in letno porabo energije zgoraj omenjene sobe, ter podatki porabe obeh frekvenčnikov. Določen nivo varnosti tudi omogoča ponastavljanje števecov porabe energije za izbran frekvenčnik.

Kompresorji sistema Silverstream so odgovorni za poganjanje zraka preko cevi iz ladijskega trupa in za regulacijo tlaka. Zrak iz kompresorjev potuje po ceveh, ki so namenjene za izpust zraka. Tako se ustvarijo majhni zračni mehurčki, ki potujejo skozi zračne reže, katerih naloga je, da omogočajo plovbo z manjšim uporom. Mehurčki potujejo skozi celotno zunanjo površino plovila. Zelo velika sprememba, ki jo lahko opazimo, je ta, da se ob delovanju sistema Silverstream poveča hitrost plovila. Zanimivost: kompresorji za svoje delovanje ne potrebujejo goriva, temveč elektriko. Ventili in kompresorji so glede na njihov status animirani na sistemu SCADA. Delovanje ventilov je pogojeno s tremi statusi: odprt, zaprt in v alarmu. Glede na njihov status se na zaslonu prikaže tudi ustrezna barvna animacija. Animiranje delovanja motorjev kompresorjev je enako kot pri delovanju ventilov.

Zgornji del zaslona na sliki 4.2 uporabniku sporoča osnovne podatke o sistemu in ladji, na kateri je lociran. Sistemski podatki operaterjem sporočajo status delovanja – ali sistem deluje, ali je ugasnjen, celotno moč delovanja kompresorjev, uporabniku pa nudijo tudi potrjevanje vseh aktualnih alarmov. Zelo pomembna stvar je tudi zagon sistema ALS iz nadzornega sistema. Na voljo sta dva systemska uporabnika: Silverstreamov ter sistemski uporabnik, ki ga nadzira podjetje INEA. Gumba sta narejena na tak način, da če ima kontrolo nad sistemom eden, drugi do njega ne more dostopati. To sporoča tudi status nad določenim uporabnikom



Slika 4.2: Sistem Silverstream v nadzornem sistemu

sistema. Če povzamem vse skupaj: sistem Silverstream je kontroliran s strani operaterjev. Active Hour Count sporoča, koliko časa je sistem že v dejanski uporabi oz. se izvaja. Podatki o ladji so za operaterje sistema tudi zelo pomembni, saj mu sporočajo hitrost plovbe in kako globoko je potopljen zadnji oz. prvi del ladje.

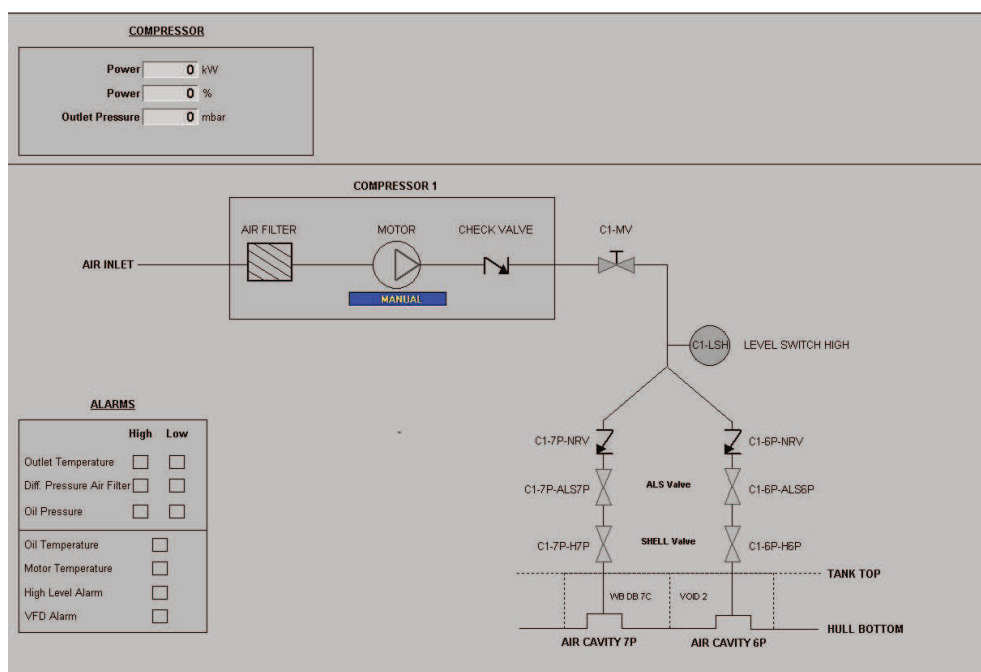
Za učinkovito delovanje sistema Silverstream, je odgovoren spodnji del zaslona, ki vsebuje podatke o kompresorjih in pripadajočih ventilih. Za vsak kompresor se izpisujejo podatki za moč delovanja in tlak, pod katerim deluje. Kot sem že omenil, se na tem sistemskem pogledu prikazujejo statusi delovanja kompresorjev in ventilov.

Nekaj informacij o delovanju ventilov:

Ker je strežnik OPC glavni za delovanje ventilov, se želimo izogniti nekomu,

ki zapira ventile ALS direktno iz OPC-ja. Prav zaradi tega je blokirano krmiljenje, ko delujejo kompresorji. Iz že omenjenega ukaza za Start/Stop sistema, pošlje HMI ukaz na krmilnik, ki mora za uspešno delovanje ukaza poslati nazaj dovoljenje za izvedbo le-tega. Če se komunikacija prek serijske linije do OPC-ja izgubi, sistem podjetja INEA ustavi kompresorje, ventile pa zapira dodatna logika na PLK-ju, ki se imenuje funkcija nadzora. Če ventili SHELL ne delujejo in so zaprti, tudi ventili ALS ne delujejo, vendar za pravilno delovanje je potrebno izključiti sistem ALS pred sistemom SHELL. Ventili imajo zelo pomembno nalogo, in sicer da ne prepuščajo morske vode skozi zračne reže, ko je sistem Silverstream v delovanju.

Na zaslonu imamo tudi gumb, s katerim prikažemo posamezen kompresor. Na njem poleg že omenjenih podatkov za moči ter tlak, opazimo tudi alarme. Poleg alarmov v alarmni konzoli lahko takoj opazimo status napak delovanja kompresorja na njegovem oknu. Kompresor je zgrajen iz zračnega filtra, motorja ter ventila, povezan pa je z že omenjenimi ventili ALS in SHELL.



Slika 4.3: Prikaz zaslona kompresorja

#### 4.1.1 Alarmiranje sistema Silverstream

Alarmi sistema Silverstream se poleg informacij v alarmni konzoli nahajajo tudi na zaslonu kompresorja, kar je zelo dobrodošla stvar za operaterje na ladji. Na sliki 4.3 opazimo na spodnji levi strani zaslona tabelo statusov alarmov. Normalno delovanje je predstavljeno s sivo barvo, stanje napake pa je izraženo z rdečo barvo.

Seznam uporabljenih alarmov:

- Outlet Temperature Alarm – Alarm High, Alarm Low (nastopi ob previsoki zunanji temperaturi),
- Differential Pressure Alarm – Alarm High, Alarm Low (nastopi ob previsokem tlaku),
- Oil Pressure Alarm – Alarm High, Alarm Low (nastopi ob previsokem tlaku goriva),
- Oil Temperature: nastopi ob previsoki temperaturi goriva,
- Motor Temperature: nastopi ob preobremenjenosti motorja,
- High Level Alarm: nastopi ob nedelovanju kompresorjev,
- VFD Alarm: predstavlja napako frekvenčnika.

Vsi alarmi sistema Silverstream so zaradi pomembnosti označeni s prioriteto 1, ki je najvišja prioriteta pojava alarma in se prikažejo takoj ob kliku na glavno okno alarmne konzole pred vsemi drugimi alarmi.

#### 4.1.2 Modul PCM

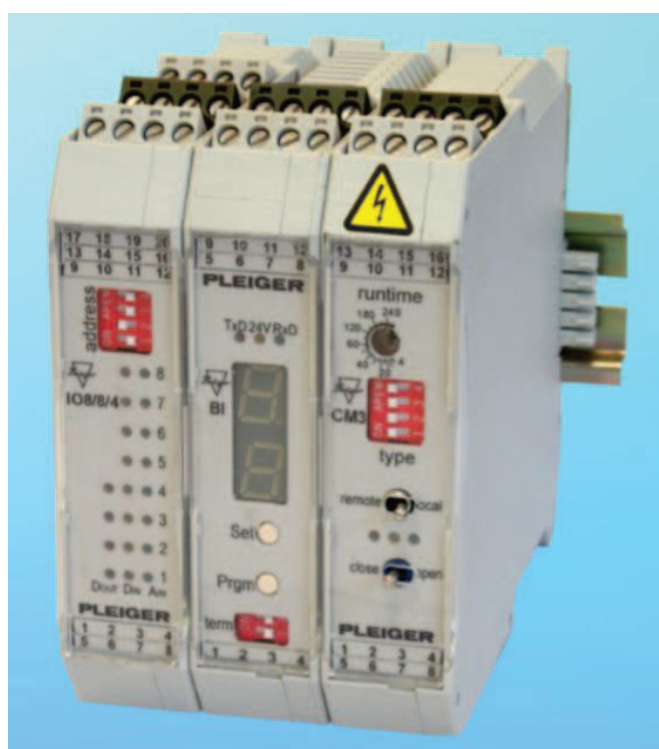
Za povezavo med kompresorji in ventili je potreben modul PCM, kar predstavlja Pleiger Control Module. Prikazan je na sliki 4.4. Odgovoren je za elektrohidravlične rotacijske pogone. Lahko se ga upravlja ročno prek digitalnih vhodov

in izhodov ali iz nadzornega sistema višje – daljinsko. Pošilja podatke o zaprtju in odprtju ventilov, statusih delovanja ventilov in kod napak. Povezava poteka preko protokola RTU [11].

Protokol RTU:

- serijski (RS-232 ali RS-458) protokol, izhaja iz arhitekture gospodar/suženj,
- je zanesljiv in enostaven za uporabo,
- pogosto se uporablja v sistemih za upravljanje zgradb, industrijski avtomatizaciji,
- zanj značilna 16-bitna sporočila.

Nadzorni modul CM3 nadzira in krmili pogonske ventile. Za vsak pogon je potreben en kontrolni modul. Njegov krmilni modul ima več elementov delovanja in prikaza, ki omogočata ročno upravljanje aktuatorja na mestu. Ima dva pomembna gumba: preklop med lokalnim in daljinskim načinom delovanje, ter preklop med odprtjem in zaprtjem ventilov. Beseda "Local" pomeni, da lahko s tem modulom upravljamo lokalno, "Remote" pa daljinsko. Daljinsko se ga upravlja prek njegovega ožičenja. Preklop med odprtim in zaprtim je mogoč le v lokalnem načinu, kjer lahko upravljamo z ventili. Na spodnji strani nadzornega modula PCM imamo še svetleče diode (LED), ki ponazarjajo status delovanja ventilov: Rdeča barva pomeni, da je ventil v alarmem stanju, zelena predstavlja način delovanja in rumena pomeni, da ventil ne deluje in čaka na ukaz [12].



Slika 4.4: Nadzorni modul PCM



## 5 Zaključek

V diplomskem delu je predstavljena uporabljena programska in strojna oprema skupaj s protokoli za izveden nadzorni sistem. V glavnem delu je bilo predstavljeno animiranje in tista poglavja, ki so pomembna za delovanje nadzornega sistema. Testiranje nadzornega sistema je potekalo v podjetju INEA, kjer smo pripravili ladijski strežnik, ki smo ga kasneje poslali na lokacijo. Testiranje in priprava aplikacije sta torej potekala v podjetju INEA, glavne stvari zagona pa na ladji, kjer je potekal zagon v realnih pogojih in razmerah.

Sistem Silverstream je najpomembnejši del tega nadzornega sistema. Izdelava in testiranje tega projekta sta nam vzela malo več časa, saj smo se prvič resno srečali s takim problemom. Sodeloval sem s sodelavci z oddelka PLK, s katerimi smo se usklajevali glede ustreznosti dela in pomoči. Sistem skupaj z moduli je zagnan na potniški ladji, sedaj čakamo na rezultate uspešnosti tega. Kot sem omenil v poglavju 4, so pričakovanja do 5.0 % prihranka pri porabi energije.

Z nadzornim sistemom podjetja INEA so uporabniki na potniški ladji zelo zadovoljni. Predvsem jih veseli uspešnost in rezultati delovanja urnikov, pregleda porabe goriv ter poročila in izračuni ostalih ladijskih porab. Pri pregledu mesečnih stroškov in porab so jim v veliko pomoč poročila o porabi goriv in celotna ladijska poročila. Tukaj bi mogoče spremenil le eno stvar, in sicer, da bi se poročila beležila v bazo, torej uporabil bi jezik SQL. Vendar to ni odvisno od nas, to je odvisno od želja in ponudbe naročnika. Moje želja je tudi, da se kasneje uporabi prehod iz aplikacije Stand Alone na Modern (Published), kjer so aktualne ArchestrA grafike. Tukaj so simboli vizualno narejeni na ustreznejši

način, tudi izrisane grafike izgledajo vizualno primernejše kot v aplikaciji Stand Alone.

Nadaljnje delo, glede na nove želje naročnika se bodo implementirale na obstoječo aplikacijo, saj je načrtana na ustrezen način, razen v primeru prehoda na aplikacijo druge verzije.

## Literatura

- [1] Invensys: *Wonderware Intouch training manual*, 2009
- [2] Wikipedia, Supervisory control and data acquisition. Dosegljivo: <https://en.wikipedia.org/wiki/SCADA>. [Dostopano: 10. 6. 2017].
- [3] Logic Control: Wonderware ArchestrA Protocols User's Guide, SuiteLink. Dosegljivo: <https://www.logic-control.com/datasheets/1/InTouch/Protocol.pdf> [Dostopano: 12. 6. 2017].
- [4] Wikipedia, Programmable logic controller. Dosegljivo: [https://en.wikipedia.org/wiki/Programmable\\_logic\\_controller](https://en.wikipedia.org/wiki/Programmable_logic_controller). [Dostopano: 16. 6. 2017].
- [5] SIEMENS, SIMATIC S7-300. Dosegljivo: <http://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-300/pages/default.aspx>. [Dostopano: 16. 6. 2017].
- [6] SIEMENS, SIMATIC S7-1500. Dosegljivo: <http://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-1500/pages/default.aspx>. [Dostopano: 16. 6. 2017].
- [7] What is Microsoft Access?. Dosegljivo: <http://www.simply-access.com/What-Is-Microsoft-Access.html>. [Dostopano: 22. 6. 2017].
- [8] OpenLink Documentation: Using ODBC 32 - bit on 64-bit Windows. Dosegljivo: <http://wikis.openlinksw.com/UdaWikiWeb/Win32vs64OdbcAdmin> [Dostopano: 23. 6. 2017].

- [9] Silverstream Technologies, The Technology. Dosegljivo: <http://www.silverstream-tech.com/the-technology/>. [Dostopano: 1. 7. 2017].
- [10] Shell Global. (4.2.2015, London). SILVERSTREAM AIRLUBRICATION TECHNOLOGY DELIVERS SIGNIFICANT ENERGY SAVINGS. Dosegljivo: <http://www.shell.com/business-customers/trading-and-supply/trading/news-and-media-releases/silverstream-air-lubrication-technology.html>. [Dostopano: 1. 7. 2017].
- [11] RTA: Modbus RTU. Dosegljivo: <http://www.rtaautomation.com/technologies/modbus-rtu/>. [Dostopano: 5. 7. 2017].
- [12] PCM Manual: PCM - Pleiger Control Modules. Dosegljivo: <http://www.pleiger-elektronik.de/en/downloads-en/category/10-ship-electronics?download=26:pcm-manual-cm3-io-8-8-4-rk16-8-bi> [Dostopano: 5. 7. 2017].