

Burn-in för heavy duty LED-lampor

Pauli Ahonen

Examensarbete för ingenjörsexamen (YH)

Utbildningsprogrammet för el- och automationsteknik

Vasa 2018



EXAMENSARBETE

Författare: Ahonen Pauli
Utbildning och ort: Elektroteknik Vasa
Inriktningsalternativ: Automationsteknik
Handledare: Stefan Emet

Titel: *Burn-in för Heavy Duty LED-lampor*

Datum 19.06.2018

Sidantal 33

Abstrakt

Detta examensarbete handlar om burn-in-tiden för heavy duty LED-lampor. Dokumentet innehåller information om hur man kan förkorta burn-in-tiden för lamporna. Dokumentet beskriver också dess funktioner, struktur/topologi, projektering och programmering i Easy Builder Pro och LOGO! Soft Comfort.

Burn-in innebär att man simulerar de påfrestningar som ett elektroniksystem utsätts för under sin livstid. Det sker vanligen med temperaturväxling, med till och frånslag eller med vibrationer. Genom burn-in kan man upptäcka kretskort som har inbyggda svagheter.

Testsystemet förverkligades med en PLC från Siemens, signalförbättrings apparat från STATUS och en display från WEINTEC samt en shunt.

Resultatet av examensarbetet blev en lösning på hur man kan gå till väga för att förkorta burn-in-tiden för LED-lampor.

Språk: Svenska

Nyckelord: Burn-in, LED, PLC, testsystem

OPINNÄYTETYÖ

Tekijä: Ahonen Pauli
Koulutusohjelma ja paikkakunta: Sähkötekniikka, Vaasa
Suuntautumisvaihtoehto: Automaatiotekniikka
Ohjaaja: Stefan Emet

Nimike: Raskaiden LED-valojen Burn-in

Päivämäärä 19.06.2018

Sivumäärä 33

Tiivistelmä

Tämä opinnäytetyö käsittelee raskaiden LED-valojen burn-in testausta. Työssä on tietoa siitä, kuinka voidaan lyhentää LED-lamppujen burn-in testausaikaa. Työssä kuvataan myös ominaisuuksia, rakennetta/topologiaa, suunnittelua ja ohjelmointia Easy Builder Pro- ja LOGO! Soft Comfortissa.

Burn-in tarkoittaa sitä, että simuloidaan stressiä, johon elektroninen järjestelmä altistuu sen käyttöiän aikana. Se tapahtuu yleensä lämpötilan muutoksella, päälle- ja poiskytkennällä tai värinöillä. Burn-inin kautta voi havaita piirilevyn sisäänrakennetut heikkoudet.

Testijärjestelmä toteutettiin Siemens PLC:llä, STATUS-signaalinmoukkaimella, WEINTEC-näytöllä ja shuntilla.

Opinnäytetyön tulos on ratkaisu siihen, miten voidaan lyhentää LED-valojen burn-in-testauksen aikaa.

Kieli: Ruotsi

Avainsanat: Burn-in, LED, PLC, testsystem

BACHELOR'S THESIS

Author: Ahonen Pauli
Degree Programme: Electrical Engineering
Specialization: Automation
Supervisors: Stefan Emet

Title: Burn-in for Heavy Duty LED-Lights

Date 19.06.2018

Number of pages 33

Abstract

This thesis project is about burn-in of heavy duty LED-lights. The document contains information on how to shorten the burn-in-time of the lights. The document also describes its features, structure/topology, design and programming in Easy Builder Pro and LOGO! Soft Comfort.

Burn-in means simulating the stresses that an electronic system is exposed to during its lifetime. It usually happens with temperature changes, on and off switching, or with vibrations. Through burn-in you can detect circuit boards that have built-in weaknesses.

The test system was implemented with a Siemens PLC, signal conditioner from STATUS and a display from WEINTEC and a shunt.

The result of the thesis was a solution on how to shorten the burn-in-time of the LED-lights.

Language: Swedish

Key words: Burn-in, LED, PLC, testsystem

Innehållsförteckning

1	Inledning.....	1
1.1	Mål och syfte.....	1
1.2	Uppdragsgivare.....	1
2	Teori.....	2
2.1	Burn-in.....	2
2.2	Kopplingen.....	4
2.3	PLC.....	5
2.4	PLC-kommunikation.....	10
2.5	HMI.....	14
2.6	Signalkonverter / förbättrare.....	15
3	Projektets genomförande.....	16
3.1	LOGO! Soft Comfort.....	16
3.2	EasyBuilder Pro.....	19
4	Resultat.....	27
5	Diskussion.....	29
6	Källförteckning.....	33

Figurförteckning

Figur 1. Bathtub Curve 1.	2
Figur 2. Bathtub Curve 2.	3
Figur 3. Kopplingsschema.	4
Figur 4. LOGO! Soft Comfort huvudkopplingen.	16
Figur 5. LOGO! Soft Comfort alarmkopplingen.	17
Figur 6. LOGO! Soft Comfort ethernet connections.	17
Figur 7. LOGO! Soft Comfort module adress och server connection..	18
Figur 8. LOGO! Soft Comfort ner laddning till PLC:n.....	19
Figur 9. EasyBuilder Pro systemparametrar.....	19
Figur 10. EasyBuilder Pro systemparametrar (settings).	20
Figur 11. EasyBuilder Pro event (alarm) log.	21
Figur 12. EasyBuilder Pro event (alarm) log (message).	21
Figur 13. EasyBuilder Pro numeric object och event display.....	22
Figur 14. EasyBuilder Pro numeric object's properties.	22
Figur 15. EasyBuilder Pro event display properties.....	23
Figur 16. Easybuilder Pro event display properties (sort).....	23
Figur 17. EasyBuilder Pro compile.....	24
Figur 18. EasyBuilder Pro download.	24
Figur 19. HMI test 1.....	25
Figur 20. HMI test 2.....	26
Figur 21. Värmekamera.	28
Figur 22. Shunt.	30
Figur 23. PLC.	31
Figur 24. SEM1600B.	32
Figur 25. TP-LINK.....	33

Tabellförteckning

Tabell 1. Fördelningstabell med 50 W LED-lampor.....	27
Tabell 2. Kyltidstabell med 10 st 50 W LED-lampor.	27
Tabell 3. Tabell av värden över shunten med olika 50 W LED-lampor.	28
Tabell 4. Tabell av värden i LOGO! Soft Comfort och utgången på SEM1600B.	29

Ordförklaring

ALU	Aritmetisk logisk enhet.
Bit	Grundenhet för information.
Buss	Ett system av gemensamma ledningar som förbinder digitala moduler.
Byte	Digital enhet bestående av 8 bitar.
COM	Component Object Model.
CPU	Central Processing Unit.
Double word	Digital enhet bestående av 32 bitar.
EPROM	Erasable programmable read-only memory.
Longword	Digital enhet bestående av 32 bitar.
PC	Personal Computer.
PLC	Programmable Logic Controller.
RAM	Random Access Memory.
Relä	En reglerteknisk elektrisk apparatur, där till- och frånslag av en låg styrsänning kan styra slutande och öppnande av en annan elektrisk krets, där både sänning och ström kan vara betydligt större.
ROM	Read-only Memory.
SCADA	Supervisory Control And Data Acquisition.
USB	Universal Serial Bus.
WLAN	Wireless Local Area Network.
Word	Digital enhet bestående av 16 bitar.
LED	Light Emitting Diode.

HMI	Human-Machine Interface.
DPPM	Defective Parts Per Million.
VDC	Volt Direct Current.
LCD	Liquid Crystal Display.
TSAP	Transport Services Access Point.

1 Inledning

Detta examensarbete innehåller information om hur man kan förkorta burn-in-tiden för heavy duty LED-lampor. Burn-in innebär att man simulerar de påfrestningar som ett elektroniksystem utsätts för under sin livstid. Det sker vanligen med temperaturväxling, med till och frånslag eller med vibrationer. Genom burn-in kan man upptäcka kretskort som har inbyggda svagheter. Alternativet för testsystemet som jag valde var att använda en PLC som mäter spänningen över en shunt för en låda med 10 st LED-lampor, som sedan alarmerar när ett fel uppstår.

1.1 Mål och syfte

Målet med projektet var att få en fungerande lösning på hur man kan förkorta burn-in-tiden som lämpar sig med existerande testning av LED-lamporna.

Syftet med projektet var att man ville förkorta burn-in-tiden på 12 h och att förbättra existerande testning.

1.2 Uppdragsgivare

Uppdragsgivaren till projektet är NORDIC LIGHTS – Herrmans Oy Ab från Jakobstad. NORDIC LIGHTS framgång är baserad på produkternas hållbarhet. NORDIC LIGHTS arbetsbelysningar används över hela världen i de svåraste och mest krävande arbetsmiljöerna. Sikten som erbjuds av arbetsstrålkastarnas högkvalitativa optik stöder maskinerna att upprätthålla en hög arbetseffektivitet under mörka arbetstimmar, samtidigt som arbetslamporna själva behöver väldigt lite underhåll. Genom att använda produkter från NORDIC LIGHTS säkerställer man att varken mörker, vibrationer, stötar, damm eller fukt påverkar sikten.

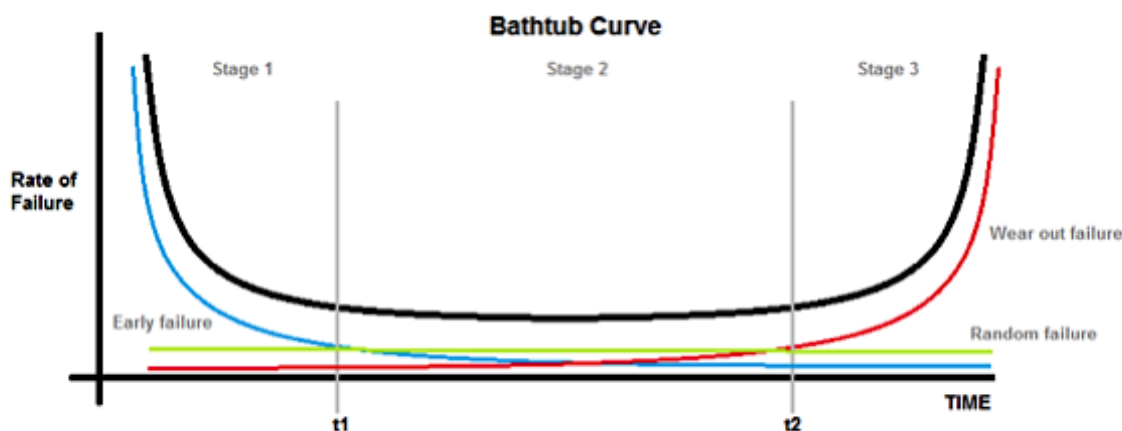
2 Teori

I teorin kommer det att lyftas upp de delar ur processen som behövs för att få en bra förståelse om hur burn-in och kopplingen fungerar. Det kommer även att tas upp information om PLC, HMI och signal converter samt om programmen LOGO! Soft Comfort, EasyBuilder Pro.

2.1 Burn-in

Test med burn-in är en process där vi upptäcker tidiga fel i komponenter, vilket ökar komponentens tillförlitlighet. I halvledarvärlden betyder det att vi tar oss närmare noll DPPM. Vid burn-in utövas komponenten under extrema driftsförhållanden (förhöjda temperaturer och spänningar). Detta lägger stress på komponenterna som testas och eliminerar dom "svaga" produkterna som ska levereras till kunderna.

Med burn-in-testningen försöker man få bort misslyckade produkter från steg 1 i "badkar" - kurvan för tillförlitlighet av elektronikutrustning, vilket ger felfrekvens mot tidpunkten för elektroniska komponenter.



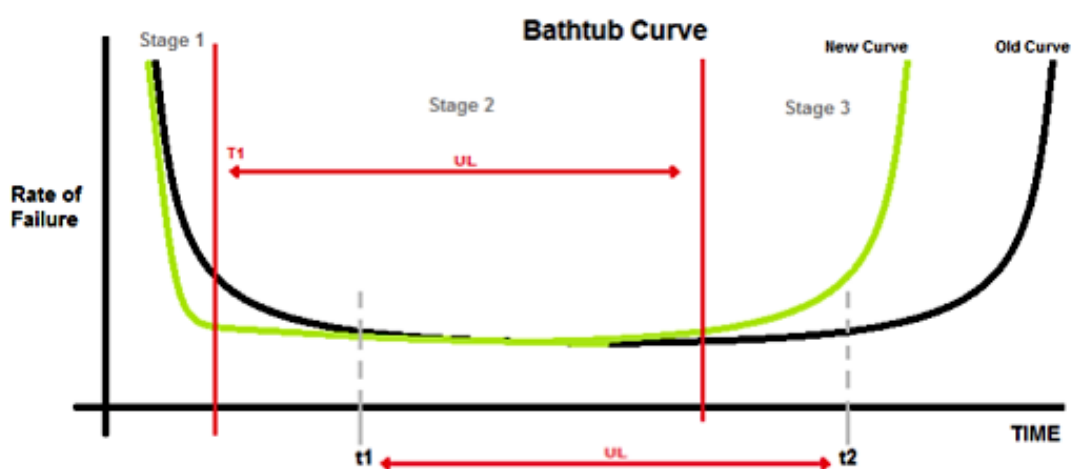
Figur 1. Bathhtub Curve 1.

Denna kurva har tre steg:

- Steg 1: Tidigt livstid - Det här är perioden där tidiga fel uppträder i en komponent. Dessa beror på brist på kontroll i tillverkningsprocesser på molekylär nivå. Under denna period misslyckas komponenterna med en hög frekvens men denna frekvens minskar med tiden. (Den blåa kurvan visar felaktighet på grund av tidigt misslyckande)

- Steg 2: Användbart livstid - Det här är den period då felaktigheten är nästan konstant och på grund av slumpmässigt förekommande fel. (visas med grön kurva)
- Steg 3: Livstidens slut - Period som markeras av ökad felfrekvens på grund av åldring av komponenter. Den här perioden markerar slutet på livslängden på en enhet. Dessa misslyckanden beror på att kritiska vägar i enheten blir slitna. (Den röda kurvan visar felaktighet på grund av åldring).

Genomförandet av burn-in-testningen minskar den totala livslängden för en enhet, men det har ingen inverkan på användbar livslängd (Steg 2) på en enhet.



Figur 2. Bathhtub Curve 2.

Fel som upptäcks vid burn-in beror på brister i tillverkning och förpackningsprocesser, som blir allt vanligare med ökande kretskomplexitet och aggressiv tekniskskalning. Traditionell fastmontering testar inte dessa typer av fel eftersom de kan vara latent och måste stressas för att manifesteras sig som "fel" (vid burn-in).

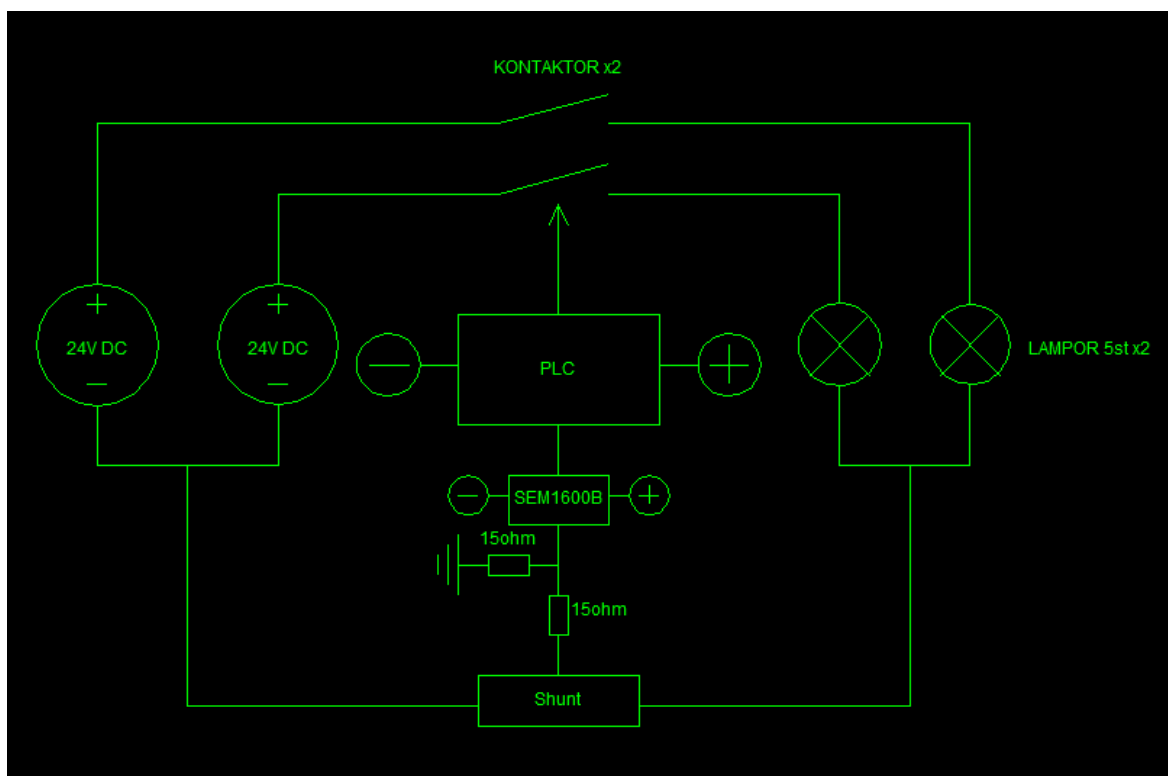
Felen som upptäcks vid burn-in-testningen kan identifieras som dielektriska fel, ledarfel, metalliseringsfel, elektromigration etc. Dessa fel är latent och slumpmässigt manifesteras under enhetens livscykel. Vid burn-in-testningen stressar man enheten och accelererar dessa latent fel som kan uppenbara sig som fel. (Mayank&Sandeep, 2014)

2.2 Kopplingen

Kopplingen är uppbyggd med 2 st 24 VDC strömkällor, 2 st kontaktorer, 10 st LED-lampor (50 W/st), shunt (25 A 60 mV), 2 st 15 ohms resistorer, signal converter (SEM1600B), PLC (SIEMENS LOGO! Model-0BA8), HMI (WEINTEC Model-MT8073iE) samt en hub.

Jag hamnade att sänka signalen från shunten för att signalen överskred gränsen som SEM1600B kunde hantera, signalen som var ungefär 53 mV vid full belastning sänktes med hälften med hjälp av 2 st 15 ohms resistorer vilket betyder att vi får en signal på 26,5 mV in till SEM1600B.

SEM1600B är menad för ansträngningsmätare och lastcellsensorer vilket inte passade så bra i detta fall men man kan undvika detta med att använda en annan sorts signal converter t.ex. SEM1600VI som är menad för ström eller spänningsprocess signaler.



Figur 3. Kopplingsschema.

2.3 PLC

En programmerbar logisk kontroll (PLC) är en industriell solid state-dator som övervakar ingångar och utgångar och gör logiska baserade beslut för automatiserade processer eller maskiner. PLC:n introducerades i slutet av 1960-talet av uppfinnaren Richard Morley för att tillhandahålla samma funktioner som relälogiksystem. Reläsystemen vid den tidpunkten brukade krångla och skapa förseningar. Tekniker fick då felsöka en hel vägg av reläer för att åtgärda problemet.

PLC:na är robusta och klarar av hårda förhållanden som t.ex. värme, kyla, damm och extrem fukt. Deras programmeringsspråk är lätt att förstå, så de kan programmeras utan svårigheter. PLC:na är modulära så de kan anslutas till olika system. Reläer under belastning kan orsaka oönskad överslag mellan kontakterna. Överslag genererar höga temperaturer som försämrar kontakterna i reläerna. Kontakterna kan också svetsas fast vilket resulterar i enhetsfel. Genom att byta ut reläer med PLC förhindrar överhettning av kontakter.

Fördelen med PLC är att en enda krets med en kompakt konstruktion kan ersätta över hundra reläer. PLC:n är programmerbar och inte hårdkodad så att dess funktion kan ändras. PLC:n kan dock vara långsammare än hårdkodad relälogik. PLC:n klarar också av svåra förhållanden som t.ex. damm, fukt, värme, kyla, elektriskt brus och elektromagnetisk störning. PLC:n är också lättare att felsöka för den har inbyggda diagnos- och överstyrningsfunktioner som gör att användarna enkelt kan spåra och korrigera programvaru- och hårdvaruproblem. Kontrollprogrammet kan ses i realtid när det körs för att hitta och åtgärda problem.

Nackdelen med PLC är de inte fungerar bra vid hantering av komplexa data. När man hanterar data som kräver C++ eller Visual Basic används ofta datorer. PLC:n kan inte heller visa data så det krävs ofta externa skärmar. (Gonzalez, 2015)

Alla PLC:n har samma grundläggande komponenter. Dessa komponenter arbetar tillsammans för att få in information till PLC:n från fältet samt utvärdera den informationen och skicka informationen tillbaka till olika områden. Om en av dessa huvudkomponenter fattas så kan inte PLC:n fungera korrekt.

Grundkomponenterna inkluderar en strömförsörjning, centralbehandlingsenhet (CPU eller processor), samprocessormoduler, ingångs- och utgångsmoduler (I/O) och kringutrustning. (Odesie)

Liksom andra datoriserade enheter finns det en central processenhet (CPU) i en PLC. CPU:n, vilken är "hjärnan" i en PLC, gör följande operationer:

- Uppdatering av ingångar och utgångar. Med denna funktion kan en PLC läsa statusen på sina ingångsterminaler och aktivera eller avaktivera sina utgångsterminaler.
- Utföra logiska och aritmetiska operationer. En CPU utför alla matematiska och logiska operationer involverade i en PLC.
- Kommunicera med minnet. PLC:ns program och data lagras i minnet. När en PLC fungerar kan dess CPU läsa eller ändra innehållet i minnesplatserna.
- Skanna applikationsprogram. Ett applikationsprogram, som kallas för ladder logikprogram, är en uppsättning av instruktioner skrivna av en PLC programmerare. Skanningsfunktionen tillåter PLC:n att exekvera applikationsprogrammet som anges av programmeraren.
- Kommunikerar med en programmeringsterminal. CPU:n överför program och data mellan sig själv och programmeringsterminalen. PLC:ns CPU styrs av operativsystemprogramvaran. Operativsystemprogramvaran är en grupp av övervakningsprogram som laddas och lagras permanent i PLC:ns minne av PLC tillverkaren.

Minnet är en komponent som lagrar information, program och data i en PLC. Processen som sätter ny information till minnesplatsen heter *writing*. Processen som hämtar information från en minnesplats heter *reading*.

De vanliga typerna av minne som används i en PLC är ROM-minne (Read Only Memory) och RAM-minne (Random Access Memory).

En ROM-plats kan läsas men inte skrivas. ROM används för att lagra program och data som inte bör ändras t.ex. PLC:ns operativa program lagras i ROM.

En RAM-plats kan läsas eller skrivas. Det betyder att informationen som lagras i en RAM-plats kan hämtas och / eller ändras. Ladder logikprogram lagras i RAM. När en ny ladder logikprogram laddas till PLC-minnet så överskrivs och raderas det gamla programmet.

RAM-minnet behöver ett eget batteri så att den kan hålla information inuti minnet när det inte finns utomstående strömförsörjning.

ROM-typen kan endast programmeras en gång efter det kan man bara avläsa minnena. ROM-minnet behöver inte ha ett eget batteri för att hålla kvar informationen.

Minneskapaciteten hos PLC varierar. Minneskapaciteter uttrycks ofta i termer av *kilo-bytes* (KB). En byte är en grupp av 8 bitar. En bit är en minnesplats som kan lagra ett binärt nummer som har värdet av antingen 1 eller 0. (Binära tal adresseras i modul 2). 1 KB minne betyder att det finns 1024 byte RAM. 16 KB minne betyder att det finns $16 \times 1024 = 16384$ byte RAM. (PDH)

PLC:n får signaler från olika givare och styr maskiner med hjälp av ingångs- och utgångsmodulerna. Modulerna förkortas ofta till I/O-moduler och en I/O-lista består av dessa moduler.

Sensorer som ger en utspänning på 24 V om sensorn är påverkad och 0 V när den är opåverkad kopplas ihop med digitala ingångar.

Analoga ingångar känner av 0–10 V, 2–10 V och 4–20 mA. De analoga signalernas värden måste omvandlas till binär form som görs i ett register i CPU:n.

Interruptingångar används när man vill att styrförloppet skall sakta ner så att funktionerna ska hinna utföras. Detta görs genom att man tvingar PLC:n att läsa ett interruptprogram. När interruptprogrammet lästs så fortsätter PLC:n att läsa huvudprogrammet därifrån det blev. PLC:ns interrupt-funktion kopplas till specialmodulerna.

Analoga utgångar används när man vill styra varvtalet på t.ex. maskiner och fläktar genom att styra spänningen. (Bergström, 2016)

PLC:ns mikroprocessors viktigaste komponenter är ALU, register och styrenhet.

ALU:ns uppgift är att verkställa logiska och aritmetriska operationer, som t.ex. addition och subtraktion, på värden i registret.

Mikroprocessors styrenhet är uppbyggd av många räknare och logiska grindar som är styrda av klocksignaler. Styrenhetens uppgift är att styra komponenterna i mikroprocessorn så att operationerna kommer i rätt följd.

Registret är en word, longword eller byte av minnet som är en komponent av mikroprocessorn. Detta minne är inkorporerad inuti mikroprocessorn, till skillnad från det

allmänna minnet i PLC:n. Register används för tillfällig lagring av data och adresser inuti CPU:n. (Ahlvik, 2012).

PLC:ns alla in- och utgångar har unika adresser för att kunna igenkännas. I standarden för programspråk IEC-61131-3 rekommenderas en metod till direkt representation av information som förknippas med ingångar, utgångar och minne till en PLC. Metoden baserar sig på att minnet i PLC:n är ordnat i tre områden, ingångsminne, utgångsminne internt minne. Man kallar ingångsminnena I, utgångsminnena Q och de interna minnena M. Man kan direkt referera allt minnesutrymme på följande sätt:

%(första bokstaven som specificerar minnesområdet, är det ett ingångsminne (I), utgångsminne (Q) eller internt minne (M)) (andra bokstavskoden specificerar hur minnet är ordnat, är det bit (X), byte (B), word (W), double word (D) eller longword (L)) (numeriskt fält som används för att ange minnespositionen).

%-tecknet indikerar en direkt refererad variabel. Man antar att man använder sig av bit (X) om man inte angett den andra bokstavskoden. Om det finns många in- och utgångar kan man namnge dem som t.ex. pump 1, startknapp, etc., då kommer man ihåg in- och utgångarnas funktion lättare.

Programmerbara logiska styrenheter använder en mängd olika programmeringsspråk för PLC-programmering. Dessa inkluderar:

- IEC 61131-3
- Sekvensiellt funktionsschema – Sequential function chart (SFC)
- Funktionsblockschema – Function block diagram (FBD)
- Ladder diagram (LD)
- Strukturerad text – Structured text (ST)
- Instruktionslista – Instruction list (IL)
- Relä ladder logik – Relay ladder logic (RLL)
- Flödesschema – Flow Chart
- C

- BASIC

Programmeringsmiljön IEC 61131-3 ger stöd för fem språk som anges av den globala standarden: Sekventiell funktionsdiagram, funktionsblockdiagram, ladder diagram, strukturerad text och instruktionslista. Detta möjliggör kompatibilitet mellan flera leverantörer och flerspråkig programmering.

SFC är ett grafiskt språk som ger koordinering av programsekvenser, stödjer alternativa sekvensval och parallella sekvenser.

FBD använder ett brett funktionsbibliotek för att bygga komplexa procedurer i ett grafiskt format. Standardmatematiska och logiska funktioner kan koordineras med anpassningsbara kommunikations- och gränssnittsfunktioner.

LD är ett grafiskt språk för diskret styrning och sammankopplingslogik. Det är helt kompatibelt med FBD för diskret funktionskontroll.

ST är ett textspråk som används för komplexa matematiska procedurer och beräkningar som är mindre lämpade för grafiska språk.

IL är ett språk på låg nivå som liknar monteringskoden. Den används i relativt enkla logiska instruktioner.

Relay Ladder Logic (RLL), eller ladder diagram, är det primära programmeringsspråket för programmerbara logiska kontrollers (PLC). Ladder logikprogrammering är en grafisk representation av programmet utformat för att se ut som relä logik.

Flödesschema är ett grafiskt språk som beskriver sekventiella operationer i en styrsekvens eller -applikation. Det används för att bygga modulära samt återanvändbara funktionsbibliotek.

C är ett programmeringsspråk på hög nivå som är lämpligt för att hantera de mest komplexa uppgifterna för beräkning, sekvensering och dataloggning.

BASIC är ett språk på hög nivå som används för att hantera matematisk, sekventiell, gränssnittsfunktioner samt datainsamling. (Globalspec)

Programmeringsspråk väljs beroende på vilken sorts bakgrund man har (utbildning, arbete, etc.) samt vad man programmerar (simpel PLC-program eller avancerade program).

2.4 PLC-kommunikation

Extensionsmoduler

PLC:ns I / O-antal kan ökas med vissa tilläggsmoduler genom systemförlängning och förlängningslinjer. Varje modul kan innehålla förlängning med både ingångs- och utgångsledningar.

Extensionsmoduler kan ha ingångar och utgångar av olika typer än de på PLC-styrenheten. När det finns många I / O placerade på långa avstånd från PLC, är en ekonomisk lösning att använda I / O-moduler och använda kablar för att ansluta dessa över de långa avstånden till PLC.

Fjärr PLC

I vissa situationer kan ett antal PLC kopplas samman med en master PLC-enhet som skickar och tar emot I / O-data från de andra enheterna.

Kablar

Man använder tvinnad parkabel som ofta är skärmade. Koaxialkabel möjliggör överföring av högre datahastigheter och behöver inte skärmning.

Fördelen med fiberoptisk kabel är att den har skydd mot elektriska störningar samt är kablens liten och flexibel.

Parallell kommunikation

Parallell kommunikation är när alla ingående bitar av ett ord överförs samtidigt längs parallella kablar. Detta gör att data kan överföras över korta avstånd med höga hastigheter. Detta kan användas vid anslutning av laboratorieinstrument till systemet.

Standardgränssnittet som används mest för parallellkommunikation är IEEE-488, betecknad som GPIB (General Purpose Instrument Bus).

Parallell datakommunikation kan äga rum mellan lyssnare, talare och kontroller. Det finns 24 linjer: 8 data (dubbelriktad), 5 status och kontroll, 3 handskakningar och 8 jordlinjer.

Seriekommunikation

Seriekommunikation är när data överförs en bit åt gången. Ett dataord måste separeras i dess ingående bitar för överföring och sedan återmonteras i ordet när de tas emot. Seriekommunikation används för att överföra data över långa avstånd. Detta kan användas för anslutningen mellan en dator och en PLC.

Serienormer

RS-232-kommunikation är den mest populära metoden mellan PLC och extern enhetskommunikation. RS-232 ingår i ett kommunikationsgränssnitt under SCADA-applikationer. Andra standarder som RS-422 och RS-423 liknar RS-232 även om de tillåter högre överföringshastigheter och längre kablar.

Det finns 2 typer av RS-232-enheter:

- DTE (Data Terminal Equipment) - Ett vanligt exempel är en dator.
- DCE (Data Communications Equipment) - Ett vanligt exempel är ett modem.

PLC kan vara antingen en DTE- eller DCE-enhet.

ASCII

ASCII är en teckenkodning som används för att representera bokstäver och andra tecken i datorer (varje bokstav / tal är översatt till ett eller nollor). Det är en 7-bitars kod, så man kan översätta 128 tecken (2^7 är 128).

Protokoll

Det är nödvändigt att utöva kontroll över flödet av data mellan två enheter, vad som utgör meddelandet och hur kommunikationen ska initieras och avslutas definieras. Detta kallas protokoll.

En enhet måste ange den andra för att starta eller sluta skicka data.

Att ansluta flera enheter kan orsaka problem på grund av kompatibilitetsproblem.

För att underlätta kommunikationen mellan olika enheter utvecklade International Standard Organization (ISO) 1979 en modell som ska användas för standardisering för Open System Interconnection (OSI). (PLC Manual)

PLC:n kan kommunicera med en PC via PC:ns USB-port. För att kunna göra detta behövs en omvandlings kabel t.ex. SSW5/USB. Om PC:n inte har seriegränssnitt kan kabeln kopplas till PLC:n via en virtuell COM-port.

USB 1.0

- Två hastigheter, 1,5 Mbit / s (låg bandbredd) och 12 Mbit / s (full bandbredd)
- Grundläggande implementering av enhetsklasser

USB 2.0

- Maximal signalhastighet på 480 Mbit / s
- Minikontakt
- Elektriska gränssnittet förbättrat
- Batteriladdningsspecifikation för personer som vill använda sin USB-port som laddstation
- On-The- Go, vilket innebär att en USB-enhet skulle kunna prata direkt till en annan USB-enhet utan att använda root hub

USB 3.0

- 5 Gbit / s dataöverföringshastighet (Superhastighet)
- Minskad strömförbrukning
- Enhetsinitierad kommunikation (I 1.0 och 2.0 kan en enhet bara kommunicera när den talas till av värden)

Dock är RS-232 mer tillförlitlig än USB och därför används RS-232 oftare. (McMahon, 2016), (RTA)

Ethernet är det vanliga sättet att ansluta datorer i ett nätverk. Detta ger ett enkelt gränssnitt och man kan ansluta flera olika enheter t.ex. datorer, routrar och switchar. Med en router och några Ethernet-kablar kan man skapa ett LAN, vilket gör att alla anslutna enheter kan kommunicera med varandra.

Ethernet har flera standarder som alla använder samma gränssnitt. Dessa inkluderar:

- 10BASE-T - stöder upp till 10 Mbps
- 100BASE-T - stöder upp till 100 Mbps
- 1000BASE-T (kallas även "Gigabit Ethernet") - stöder upp till 1000 Mbps

De flesta Ethernet-enheter är bakåtkompatibla med lägre Ethernet-kablar och enheter med låg hastighet. Förbindelsen kommer bara att vara lika snabb som den lägsta gemensamma nämnaren. Om du till exempel kopplar en dator med en 10BASE-T NIC till ett 100BASE-T-nätverk, kan datorn bara skicka och ta emot data vid 10 Mbps. Om du har en Gigabit Ethernet-router och ansluter enheter till den med 100BASE-T-kablar, blir den maximala dataöverföringshastigheten 100 Mbps.

Medan Ethernet fortfarande är standard för trådbundet nätverk, har det ersatts på många områden med trådlösa nätverk. Med Wi-Fi kan du ansluta din bärbara dator eller smartphone till ett nätverk utan att kopplas till väggen med en kabel. 802.11ac Wi-Fi-standarder ger även snabbare maximala dataöverföringshastigheter än Gigabit Ethernet. Kabelanslutna anslutningar är dock mindre benägna att störa och är säkrare än de trådlösa. Därför använder många företag och organisationer fortfarande Ethernet.

Ethernet är också känt genom sitt tekniska namn, "IEEE 802.3." (TechTerms, 2014)

System som använder sig av LAN använder ett enda nätverk medan Ethernet är flera nätverk som är ihopkopplade. Man har en större räckvidd om man använder Ethernet men överföringshastigheten är långsammare för att den måste sändas över längre distanser.

Wi-Fi, känt som WLAN, är ett trådlöst nätverk. För att kunna använda WLAN måste man ha en trådlös router som kopplas till LAN eller till telefonnätverket. Nuförtiden behövs bara ett SIM-kort för att kommunicera med telefonnätverket. Enheten måste ha en antenn som kan ta emot WLAN. Fördelen är att man inte behöver dra kablar mellan routern och enheterna men överföringshastigheten är inte lika snabb som LAN.

2.5 HMI

HMI är akronym för Human Machine Interface, vilket helt enkelt är ett gränssnitt mellan en användare och en maskin. HMI är en term som är specifik för tillverknings- och processtyrningssystem. En HMI ger en visuell representation av statusen för ett styrsystem med realtidsdatainsamling.

HMI kan också fungera som en centraliserad styrenhet för tillverkningslinjer, eftersom de vanligtvis är utrustade med icke-flyktigt minne som innehåller datauppskrifter, kan utföra händelsesloggning och händelseutlösning och kan tillhandahålla video-flöde för statusövervakning. För att en produktionslinje ska integreras med en HMI måste den först anslutas till en programmerbar logisk kontroll (PLC) eller rörelsekontrollenhet. Det är PLC:n eller rörelsekontrollen som tar informationen från maskinsensorerna och omvandlar den till boolesk algebra för att kommunicera med HMI.

En HMI används för tre primära roller: ersättning av knapparna, datahantering och systemövervakning.

- Tryckknapps utbytesfunktionen ersätter På / Av-knappar, brytare eller någon mekanisk enhet som utför en kontrollfunktion med lysdioder. Avlägsnandet av dessa mekaniska anordningar är möjligt eftersom HMI kan ge en visuell representation av alla dessa enheter på LCD-skärmen samtidigt som de utför alla funktioner.
- Datahantering används i applikationer som kräver kontinuerlig återkoppling och övervakning. HMI som fungerar som datahanterare är utrustade med stora minneskapaciteter.
- Den sista av de tre HMI-typerna kallas systemövervakaren eftersom den fungerar med SCADA (Supervisory Control And Data Acquisition) och MES (Manufacturing Execution System) för att visa information. Dessa är centraliserade eller distribuerade kontrollsystem som övervakar och kontrollerar hela platser eller komplex av stora delsystem som sprids ut över stora områden. En HMI är vanligtvis kopplad till SCADA-systemets databaser och program för att tillhandahålla realtid, diagnostisk data och hanteringsinformation.

Den största fördelen med en HMI är dess användarvänlighet och förmåga att personifiera sitt grafiska gränssnitt. En HMI kan fullt ut stödja komplexa applikationer med flera skärmar och flera rutiner som körs samtidigt. Varje HMI kommer med olika funktioner; vissa kan

spela ljud, spela upp video eller till och med ha fjärråtkomstkontroll. Valet av en HMI bör optimeras för en kunds specifika tillämpning med hänsyn till den typiska användarens förmåga, liksom miljöförhållanden som ljud, ljus, damm, syn etc. (Electromate)

Grundprinciperna för presentation av information i användargränssnittet är att informationen ska vara synlig och lätt tillgänglig för användaren den ska också vara lättläst och logiskt placerad. Man ska också kunna sammankoppla eller urskilja informationen.

Systemets användare bör ha tillgång till dialoger och annan relevant information om systemet. Det ska även vara möjligt att snabbt få tillgång till den information som behövs. Det är viktigt att användaren lätt kan hitta information om en specifik uppgift eftersom dålig information kan leda till misstag eller onödig stress. Det bör finnas en homogen logik mellan programsidor, både på utförandeplanet och hur informationen presenteras. En sådan sammanhållning gör systemet mera förutsägbart och trevligare att använda. Systemet ska innehålla lämpliga symboler och meddelanden så att användaren håller sig medveten om vad som händer i systemet. Användaren ska inte behöva memorera information utan informationen ska anpassas till varje interaktion mellan systemet och användaren. Användaren ska känna att hen har hela tiden full kontroll över systemet. Systemet ska ge respons på utförd aktivitet så att användaren är medveten om situationen. (Ahlvik, 2012).

2.6 Signalkonverter / förbättrare

En signalkonverter är en enhet som omvandlar signaler från sensorer till industriella strömsignaler, omvandlar analoga ingångssignaler till analoga utsignaler, normaliserar signaler eller isolerar signaler. De elektroniska signalerna som detekteras av en detektor har olika karaktär på grund av skillnader i data som mäts (t.ex. temperaturer, tryck och spänningar). Detta gör det svårt att mata in data direkt i styrkretsen. Därför är det nödvändigt att omvandla eller normalisera data till en signalform som är lätt att använda. (Omron)

Signalförbättrare tillhandahåller filtrering, förstärkning, omvandling och / eller andra processer som krävs för att göra sensorutgångar lämpliga för avläsning av PLC, datorer m.m. Signalförbättrare används främst för datainsamling, där sensorsignaler måste normaliseras och filtreras till nivåer som är lämpliga för analog-till-digital konvertering. Den digitala signalen är då tillgänglig för att analyseras eller tolkas av en datoriserad enhet. (Globalspec)

3 Projektets genomförande

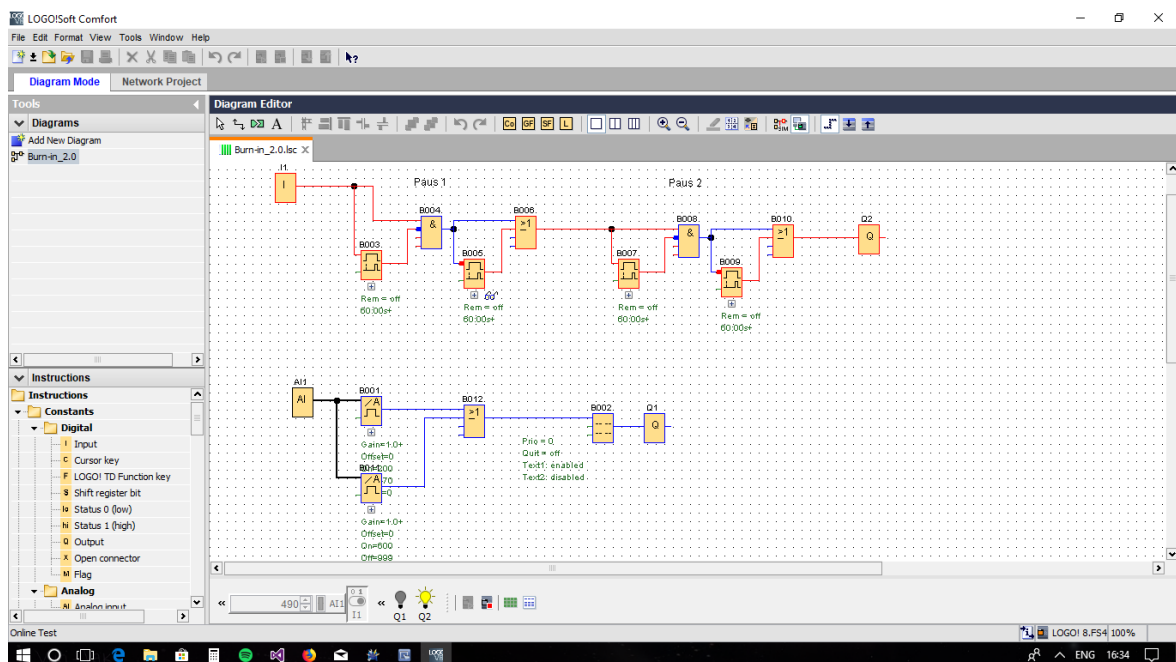
Genomförande kapitlet innehåller information om hur simuleringen är uppbyggd.

3.1 LOGO! Soft Comfort

LOGO! Soft Comfort är ett program som programmerar och ställer in en PLC. LOGO! Soft Comfort är ett enkelt och smidigt program att använda.

Huvudkopplingen är uppbyggd med 4 st On-Delay, 2 st AND, 2 st OR funktioner samt en digital input och output.

Pausen fungerar så att när den första On-Delayen får en insignal börjar den räkna upp till ett visst antal innan den aktiveras och då bryts signalen till Q2 och då börjar den andra On-Delayen räkna upp tills den aktiveras som då aktiverar Q2 igen, Paus1 och Paus2 är identiska.

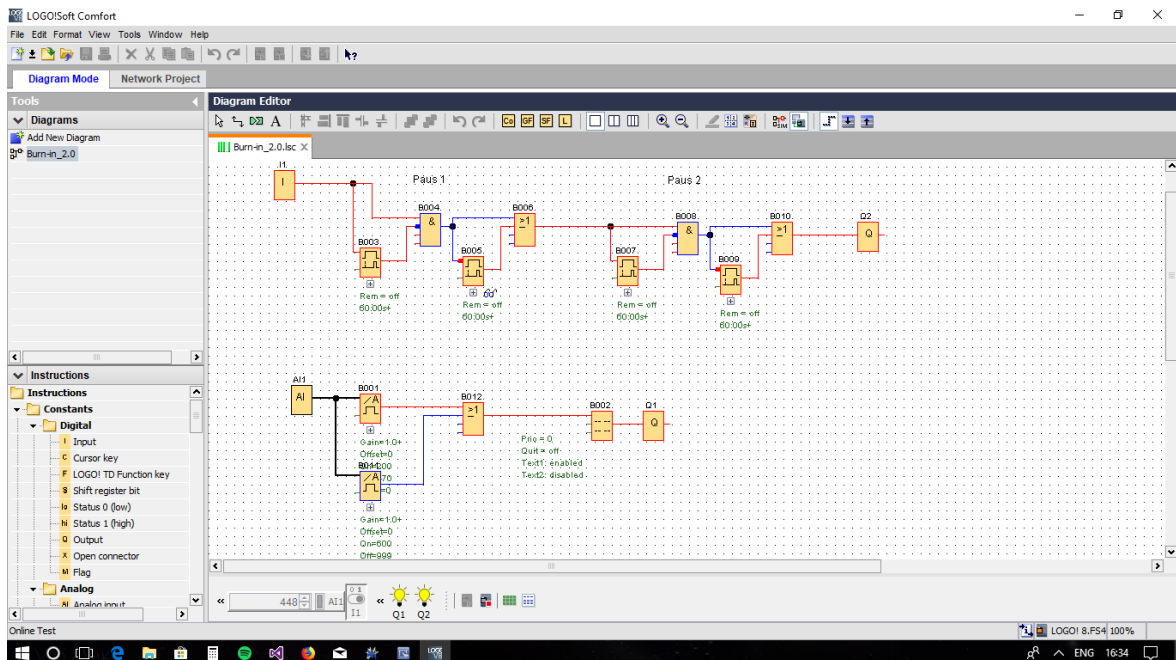


Figur 4. LOGO! Soft Comfort huvudkopplingen.

Alarmkopplingen är uppbyggd med 2 st Analog threshold triggers, 1 st OR funktion, 1 st Message-text samt en Analog input och Output.

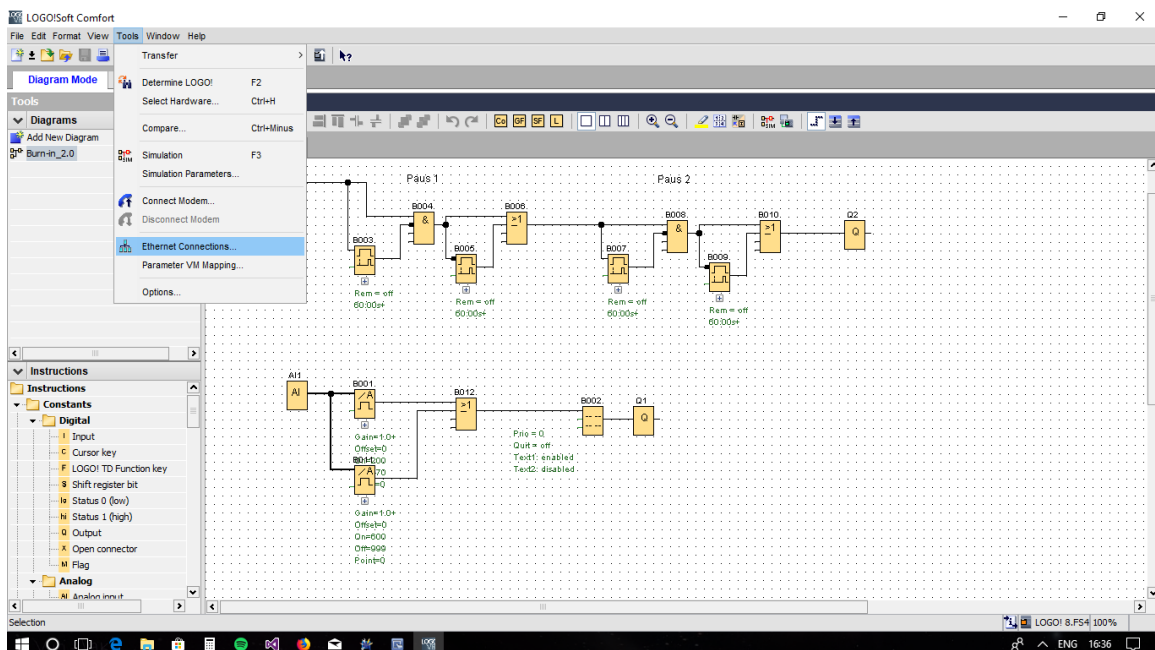
Den första Analog threshold triggern mäter det analoga värdet som kommer in till PLC:n och aktiveras om värdet sjunker under en viss gräns. Den andra Analoga threshold triggern

fungerar på samma sätt men aktiveras om värdet blir högre än ett visst värde. Message-text skriver ut på PLC-skärmen ett visst ord eller en mening när en av Analog threshold trigger aktiveras.



Figur 5. LOGO! Soft Comfort alarmkopplingen.

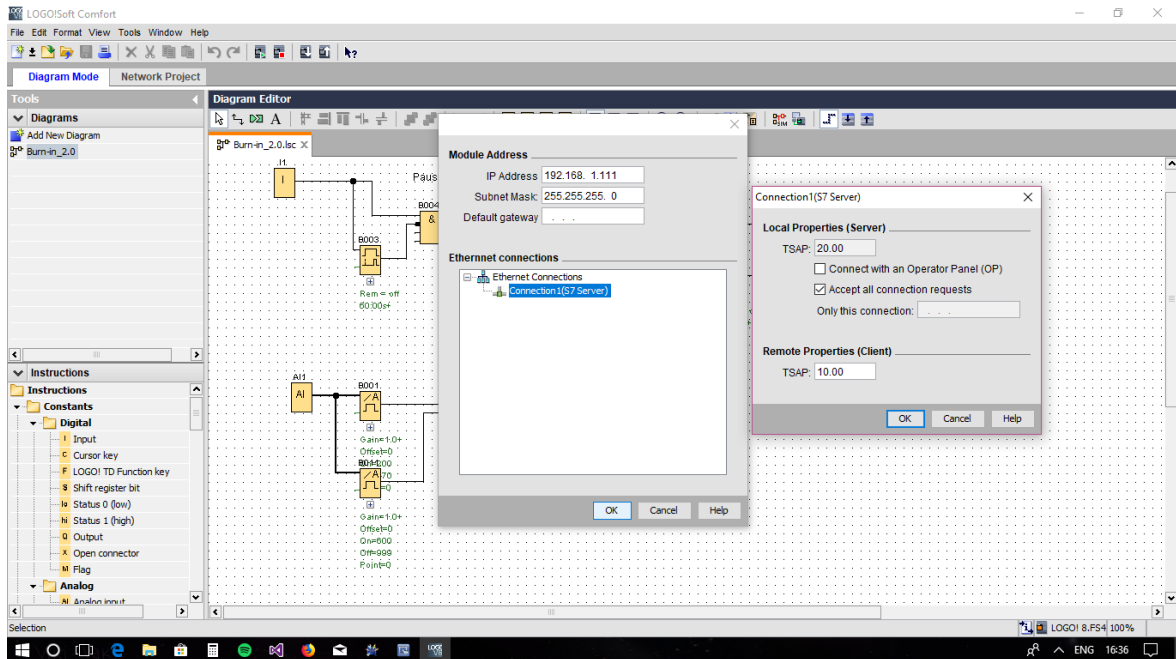
För att få skärmen att fungera med PLC:n så måste man gå till Ethernet Connections, som finns under Tools.



Figur 6. LOGO! Soft Comfort ethernet connections.

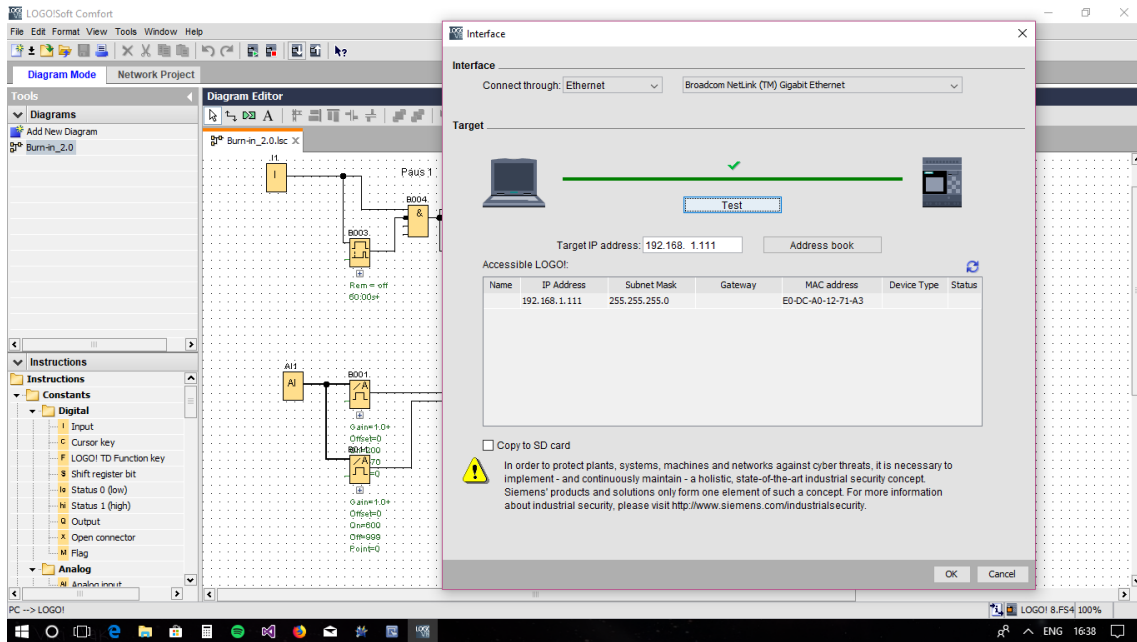
Högerklicka på Ethernet Connections och välj server Connection och kryssa i Accept all connection requests och lägg Local TSAP till 20.00 och Remote TSAP till 10.00.

Alla enheters IP Adresser måste vara på samma nivå med andra ord så måste alla börja men 192.168.1.XXX och alla enheter måste också ha samma Subnet Mask vilket är 255.255.255.0.



Figur 7. LOGO! Soft Comfort module address and server connection..

Efter att programmet är uppbyggt och inställningarna är gjorda, kan man ladda ner programmet till PLC:n.

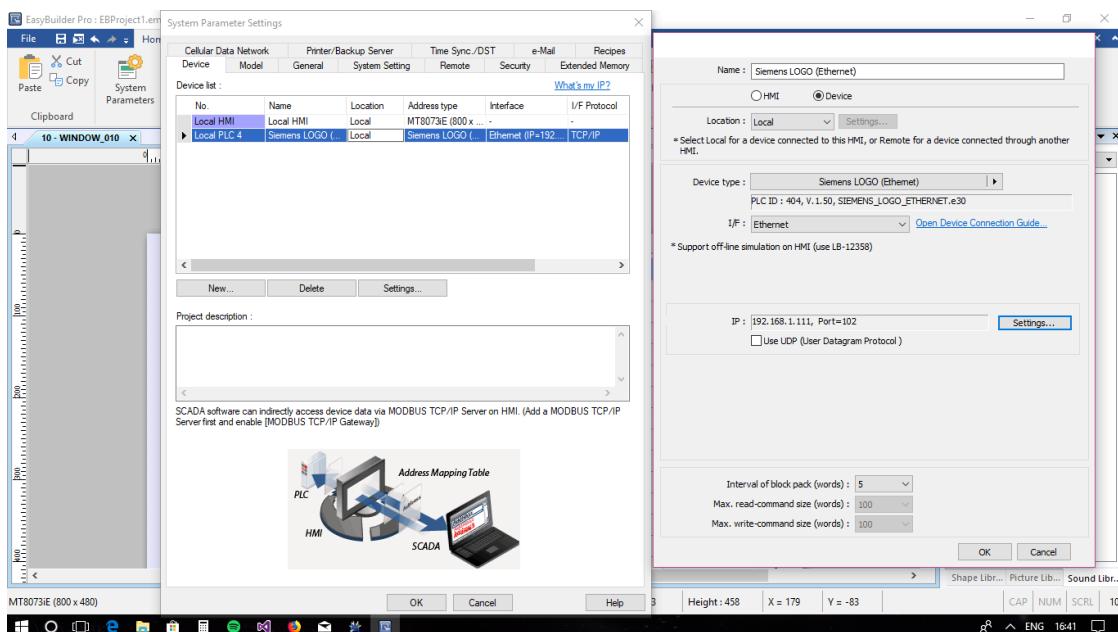


Figur 8. LOGO! Soft Comfort ner laddning till PLC:n.

3.2 EasyBuilder Pro

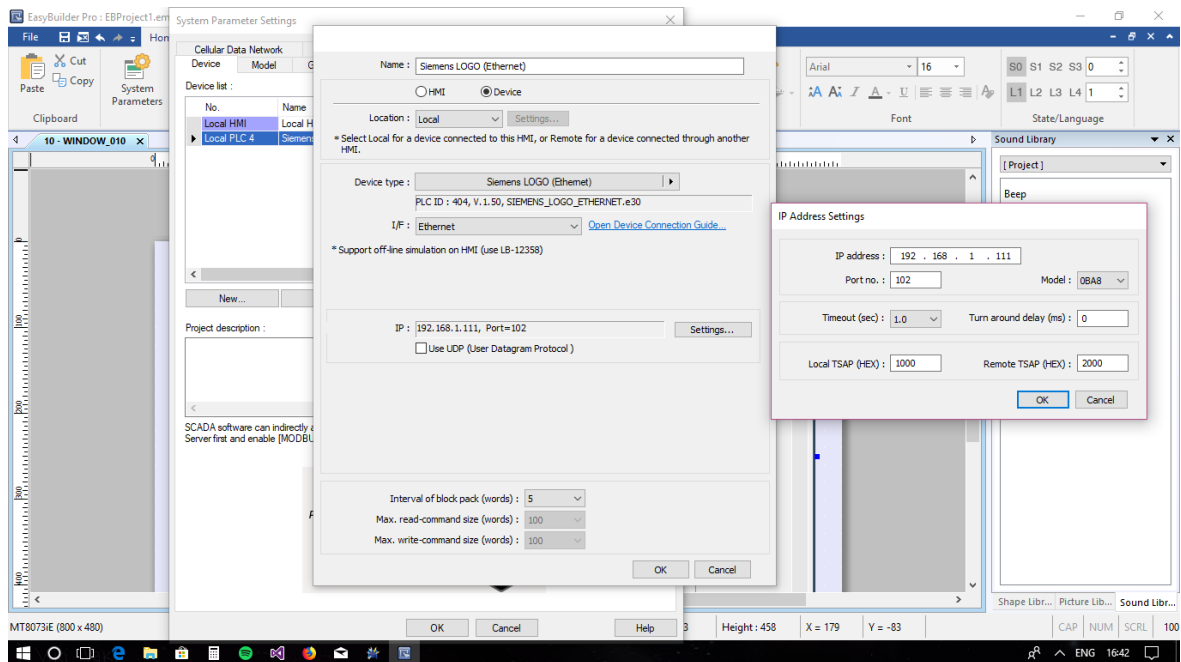
EasyBuilder Pro är ett program som man kan programmera samt skapa en layout till skärmen. EasyBuilder Pro är ett enkelt och smidigt program att använda.

Under systemparametrar ska man lägga till den PLC man har. Tryck på New... och välj rätt Device type och vilken typ av I/F. I detta fall använde jag Ethernet.



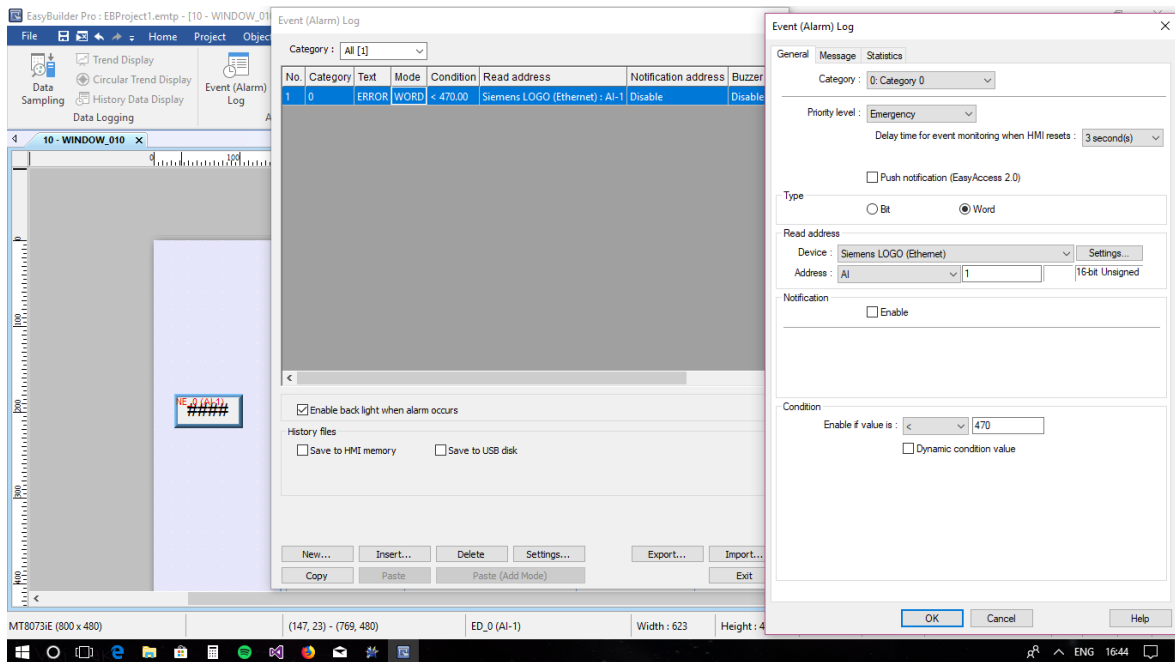
Figur 9. EasyBuilder Pro systemparametrar.

Under Settings... ska man kolla att IP adressen, modellen och portnumret är rätt, porten är 102 som default. Local TSAP ska vara 1000 och Remote TSAP ska vara 2000.



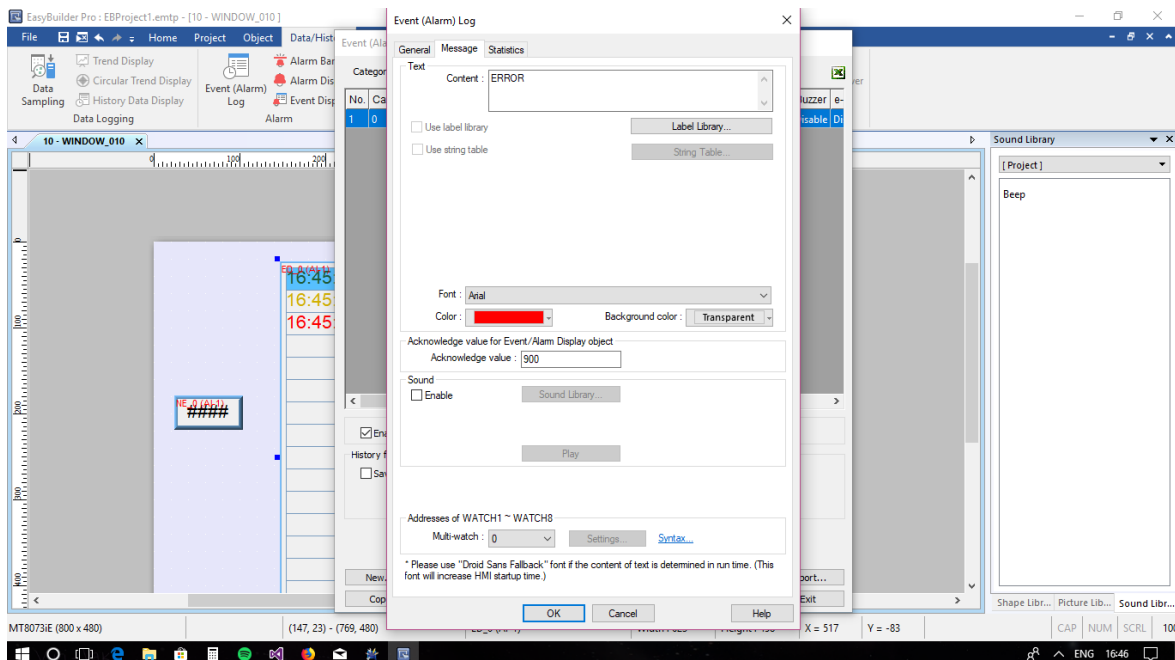
Figur 10. EasyBuilder Pro systemparametrar (settings).

Under Event (Alarm) Log ska vi lägga till alarmen. Tryck på New... och välj rätt Device och Adress. Under Condition kan vi bestämma när alarmet ska aktiveras, i detta fall har jag att alarmet ska aktiveras om värdet sjunker under 470. Man kan ha flera olika alarm för olika nivåer, t.ex. om man ställer in Alarm1=470 och Alarm2=430 så skulle Alarm1 betyda att 1 st lampor har gått sönder och Alarm2 skulle betyda att 2 st lampor har gått sönder.



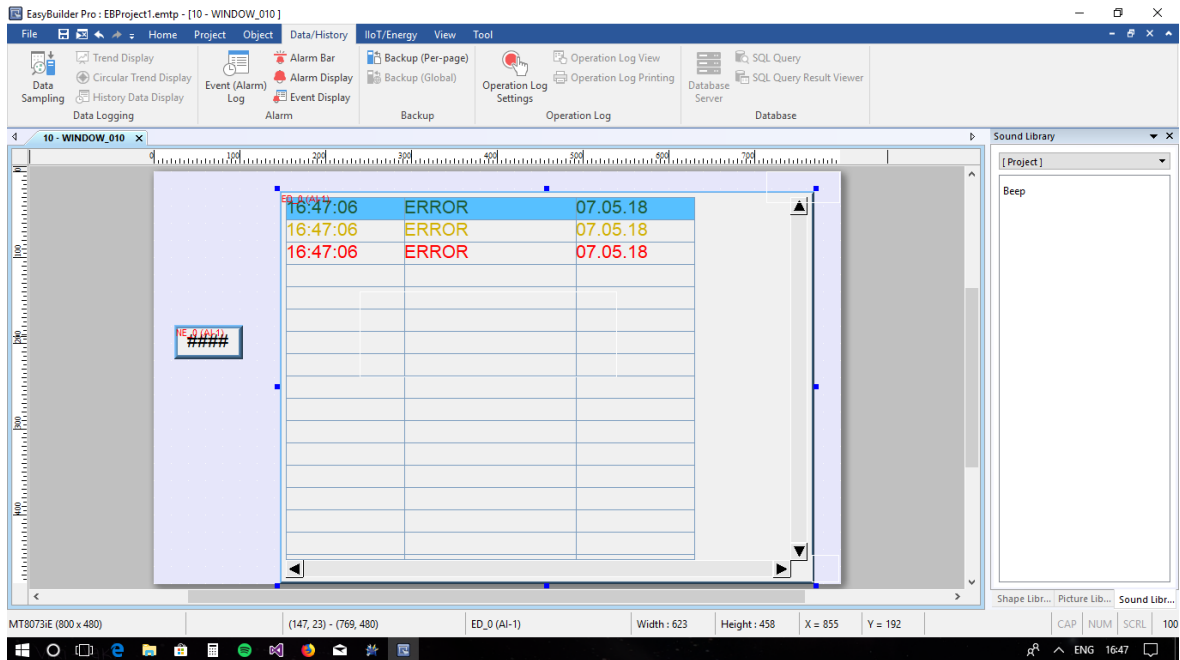
Figur 11. EasyBuilder Pro event (alarm) log.

Under Message-fliken kan man namnge alarmet. Man kan också ställa in ett Acknowledge value om man vill.



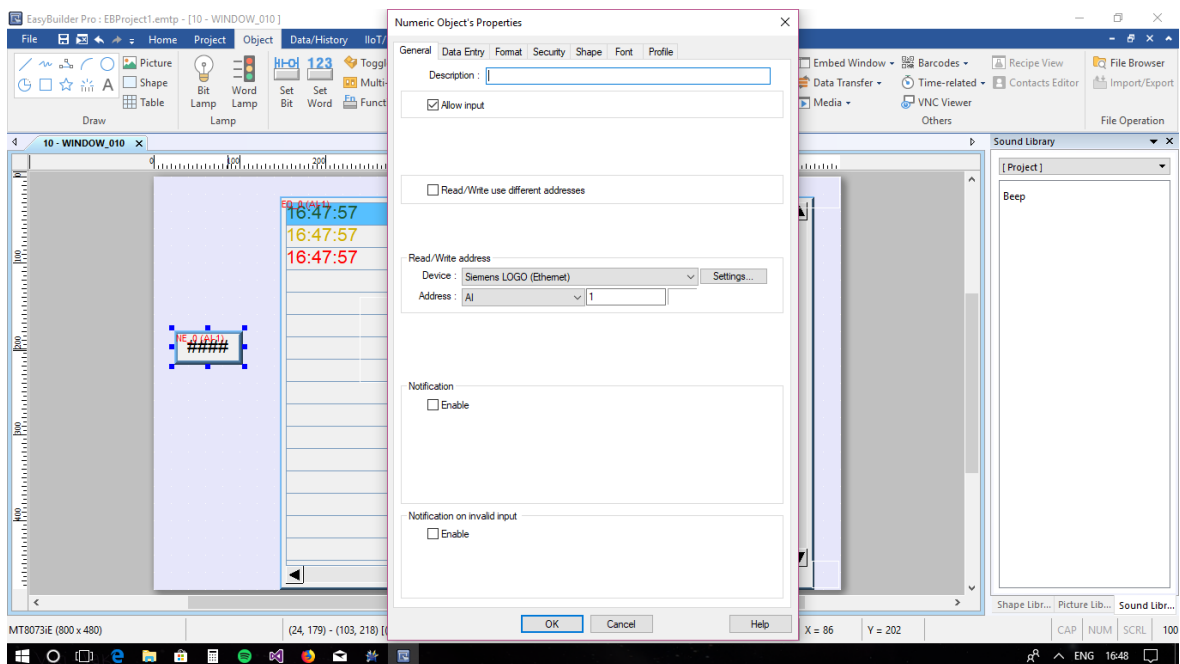
Figur 12. EasyBuilder Pro event (alarm) log (message).

Som layout till skärmen har jag ett Numeric Object och en Event Display.



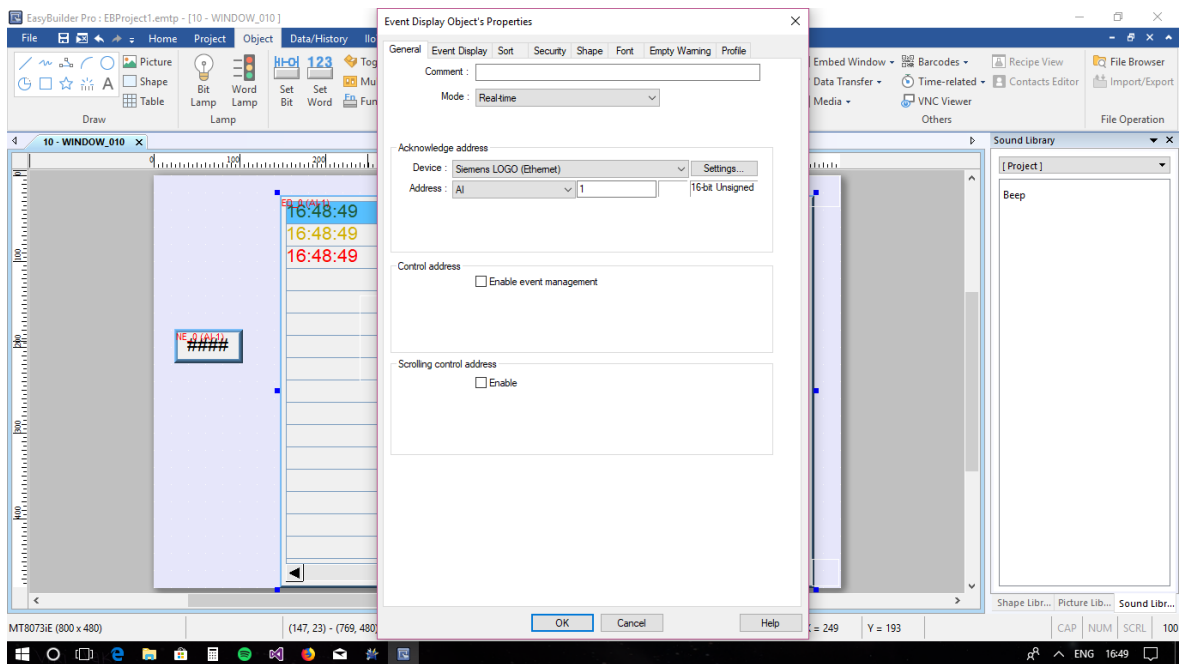
Figur 13. EasyBuilder Pro numeric object och event display.

Inställningarna för Numeric Object så ska man välja rätt PLC och adress.



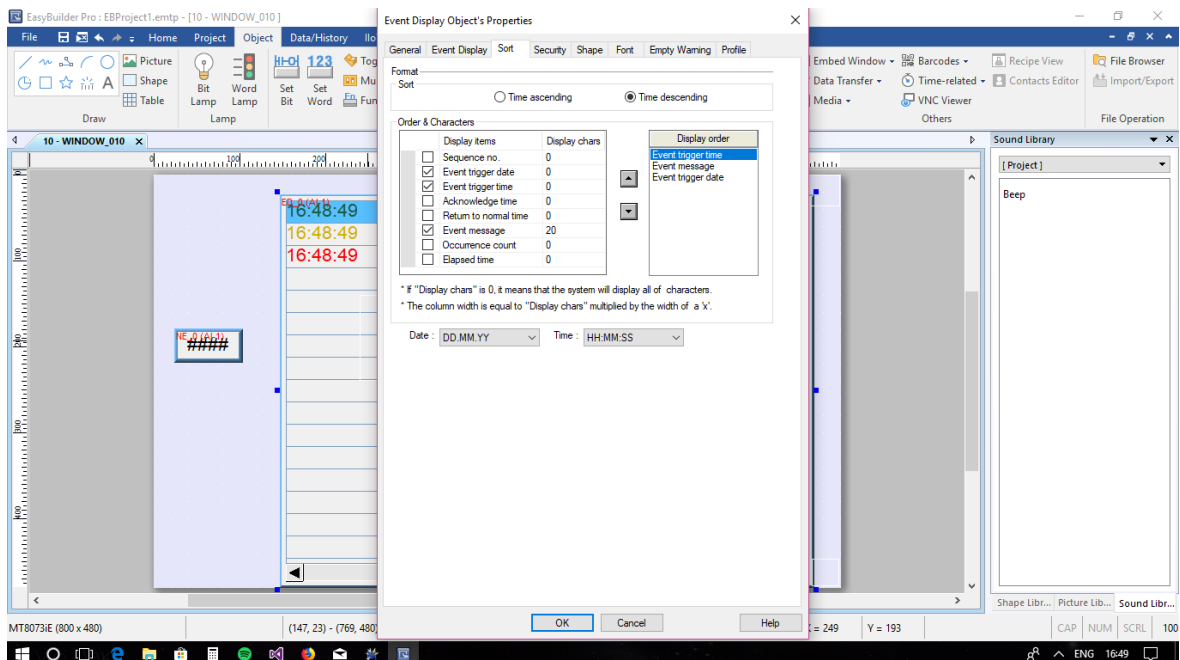
Figur 14. EasyBuilder Pro numeric object's properties.

Inställningarna för Event Displayn ska man välja rätt PLC och Adress om man vill att alarmen ska bli kvitterade.



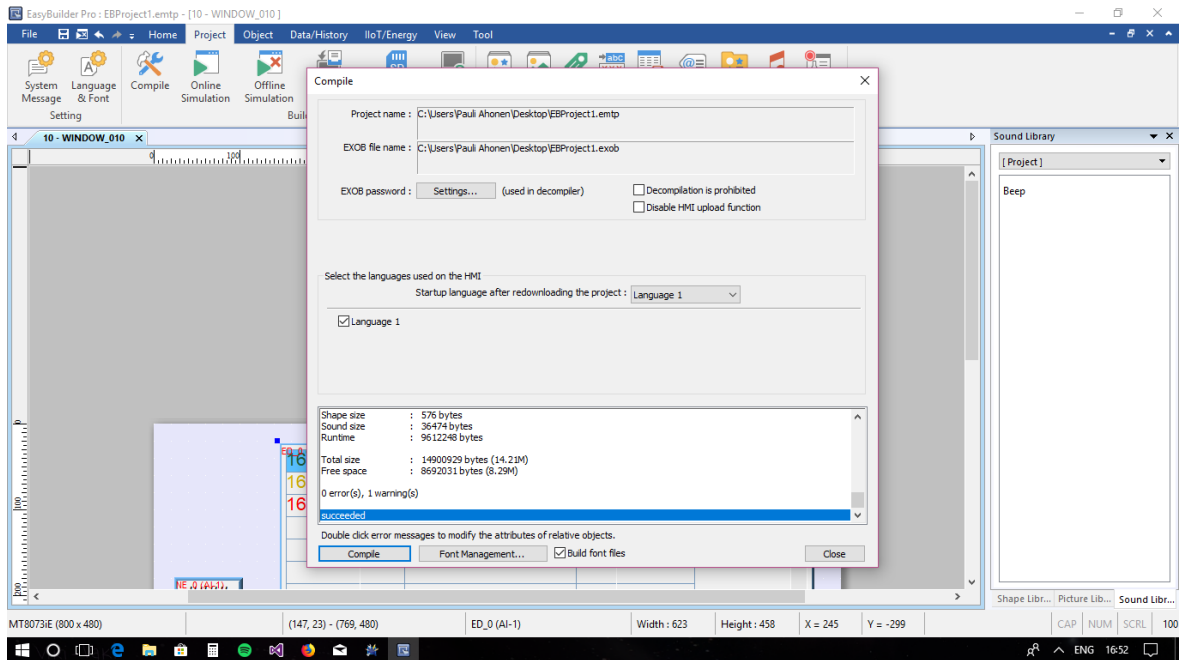
Figur 15. EasyBuilder Pro event display properties.

Under Sort fliken kan vi bestämma vad som ska visas när alarmet aktiveras. Man kan också bestämma vilken ordning man vill ha på Date och Time.



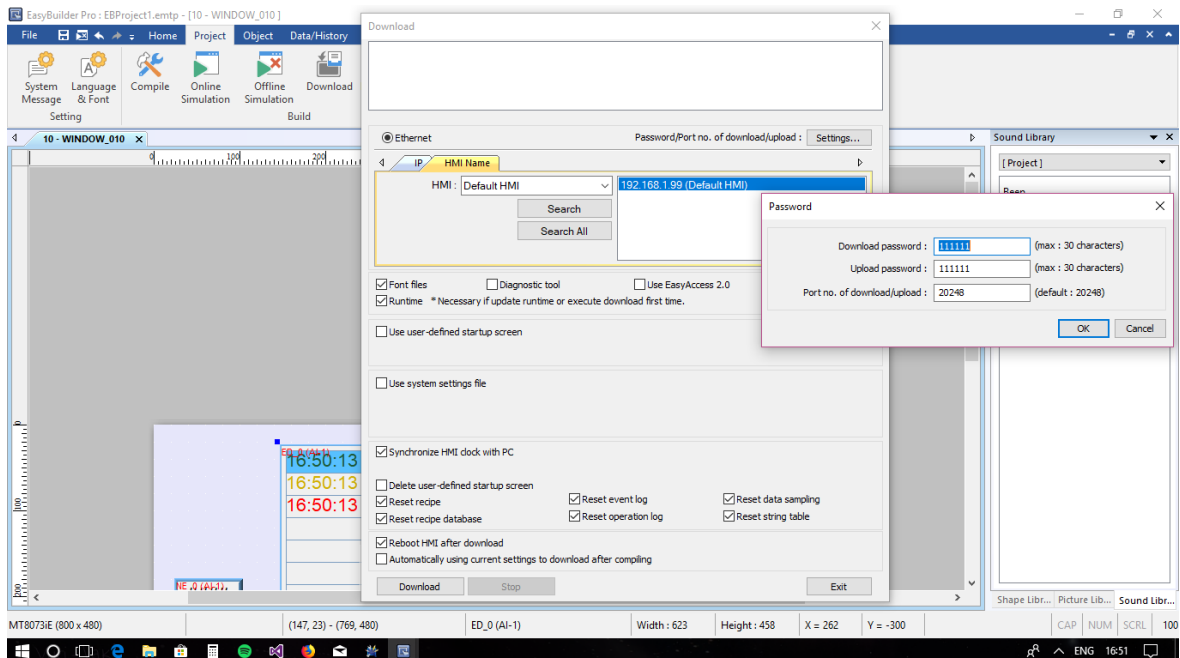
Figur 16. Easybuilder Pro event display properties (sort).

Innan man laddar ner programmet till skärmen ska man gå till Compile, trycka på Compile och kontrollera att det står succeeded.



Figur 17. EasyBuilder Pro compile.

Efter Compile så kan man ladda ner programmet till skärmen, trycka på Download och välja rätt skärm under HMI Name. Sedan ska man gå till settings... för att kolla att lösenordet och portnumret är rätt. Som default är lösenordet 111111 och portnumret 20248.



Figur 18. EasyBuilder Pro download.

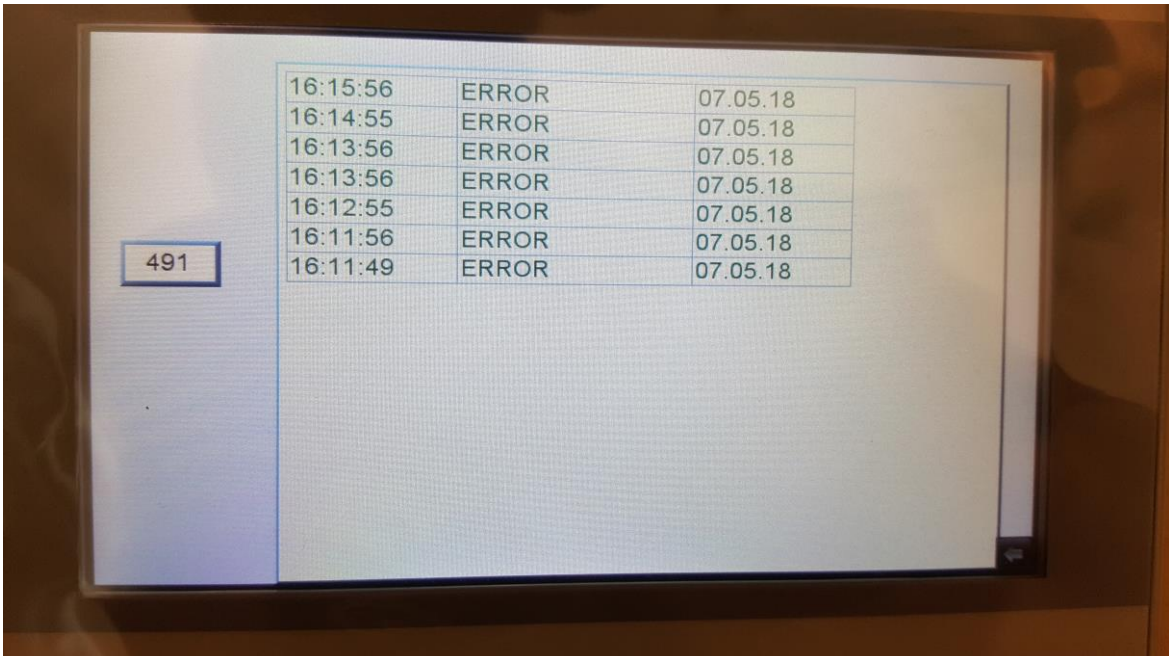
Programmet fungerar så att när programmet startar så lyser lamporna i 1 min före första pausen startar och pausen är 1 min lång. Efter pausen lyser lamporna i 1 min före andra pausen startar. Andra pausen är också 1 min lång och efter andra pausen så lyser lamporna tills man stoppar programmet.

Om man kollar på alarman så ser man att det kan komma 2 st av samma alarm.

TEST 1

Om vi tittar på alarman så börjar det med:

- 16:11:49 = Skärmen startar upp
- 16:11:56 = Programmet startar
- 16:12:55 = Paus 1
- 16:13:56 = Lamporna lyser igen (x2)
- 16:14:55 = Paus 2
- 16:15:56 = Lamporna lyser igen



The image shows a screenshot of an HMI test interface. On the left side, there is a button labeled '491'. On the right side, there is a table with three columns: timestamp, error type, and date. The table contains seven rows of data, all with 'ERROR' as the error type and '07.05.18' as the date. The timestamps are: 16:15:56, 16:14:55, 16:13:56, 16:13:56, 16:12:55, 16:11:56, and 16:11:49.

16:15:56	ERROR	07.05.18
16:14:55	ERROR	07.05.18
16:13:56	ERROR	07.05.18
16:13:56	ERROR	07.05.18
16:12:55	ERROR	07.05.18
16:11:56	ERROR	07.05.18
16:11:49	ERROR	07.05.18

Figur 19. HMI test 1.

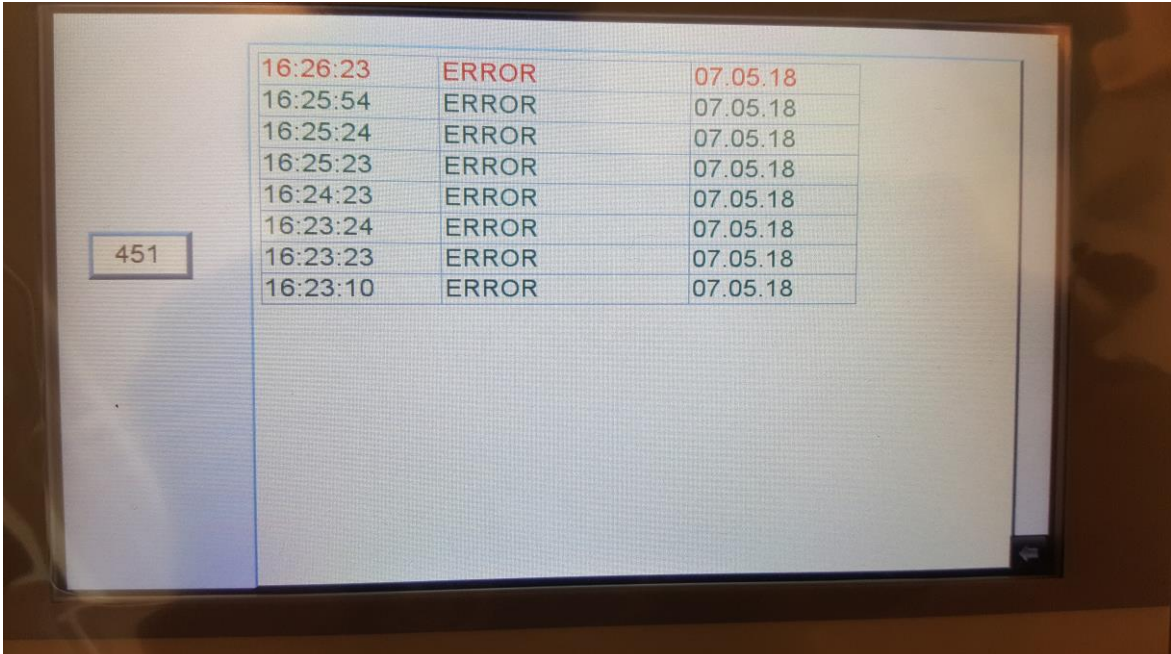
TEST 2

Här drog jag ut en lampa mellan Paus 1 och Paus 2.

Om vi tittar på alarmen så börjar det med:

- 16:23:10 = Skärmen startar
- 16:23:23 = Programmet startar (x2)
- 16:24:23 = Paus 1
- 16:25:23 = Lamporna lyser igen (x2)
- 16:25:54 = FAIL
- 16:26:23 = Paus 2

Efter Paus 2 kommer det inte flera alarm för att värdet inte stiger över 470.



16:26:23	ERROR	07.05.18
16:25:54	ERROR	07.05.18
16:25:24	ERROR	07.05.18
16:25:23	ERROR	07.05.18
16:24:23	ERROR	07.05.18
16:23:24	ERROR	07.05.18
16:23:23	ERROR	07.05.18
16:23:10	ERROR	07.05.18

Figur 20. HMI test 2.

4 Resultat

Resultatet blev en fungerande lösning på hur man kan få fram tiden när lamporna går sönder i anläggningen. När man vet tiden då lamporna går sönder så kan man optimera testningstiden.

Tabell 1. Fördelningstabell med 50 W LED-lampor (millivolt är mätt över shunten och ampere är från strömkällorna).

Tabell 1. Fördelningstabell med 50 W LED-lampor.

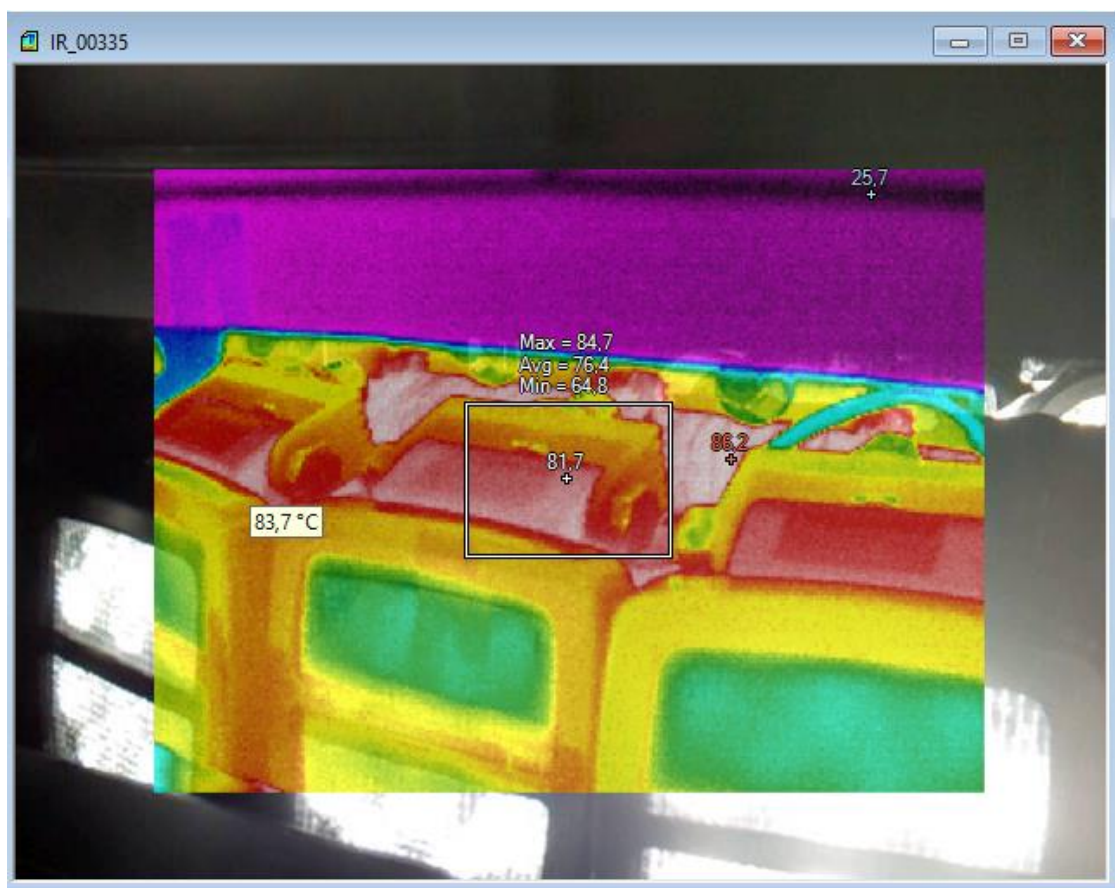
50W Lampor	Kall	Varm	Kabel 1	Kabel 2	Totalt
st	mV	mV	Ampere (A)		A
1	5,1	5,04	0	2,1	2,1
2	10,25	10	2,1	2,1	4,2
3	15,45	15,1	2,1	4,2	6,3
4	20,66	20,27	4,2	4,2	8,4
5	26	25,55	4,2	6,4	10,6
6	31,3	30,7	6,4	6,4	12,8
7	36,6	36,1	6,4	8,6	15
8	42,9	47,1	8,6	8,6	17,2
9	47,9	47,1	8,6	11	19,6
10	53,8	52,4	10,8	11	21,8
			Medelvärde per lampa		2,18

Tabell 2. Kyltidstabell med 10 st 50W LED-lampor. Som vi ser i tabellen så hinner inte lamporna kylas ner till rumstemperatur under pausen.

Tabell 2. Kyltidstabell med 10 st 50 W LED-lampor.

BURN-IN	5 Korgar på varandra		2 Korgar på varandra
50W Lampor	Mitten	Ovanpå	Sidan om test området
MaxTemp	87 °C	85 °C	85 °C
MinTemp	65 °C	60 °C	60 °C
Kyltid			
30min			
MaxTemp	63 °C	57 °C	51 °C
MinTemp	53 °C	47 °C	38 °C
60min			
MaxTemp	48 °C	42 °C	36 °C
MinTemp	40 °C	36 °C	29 °C
Uppvärmning ca. 0,1 °C/s			

Temperaturerna är mätta med en värmekamera.



Figur 21. Värmekamera.

Tabell 3. Värdet över shunten med olika 50 W LED-lampor. Värdet sjunker med ca. 0,03 mV när lampan blir varm.

Tabell 3. Tabell av värden över shunten med olika 50 W LED-lampor.

50W Lampor	Kall (mV)	Varm (mV)
Lampa 1	5,07	5,04
Lampa 2	5,33	5,3
Lampa 3	5,1	5,09
Lampa 4	5,15	5,11
Lampa 5	5,08	5,06
Lampa 6	5,06	5,03
Lampa 7	5,11	5,08
Lampa 8	5,12	5,09
Lampa 9	5,17	5,14
Lampa 10	5,1	5,07

Tabell 4. Tabell av värden i programmet LOGO! Soft Comfort och från utgången på SEM1600B.

Tabell 4. Tabell av värden i LOGO! Soft Comfort och utgången på SEM1600B.

50W Lampor		
st	Värdet i programmet	Värdet från SEM1600B
0	90	1,6V
1	124	2,3V
2	166	2,97V
3	207	3,65V
4	248	4,3V
5	290	5V
6	327	5,7V
7	367	6,4V
8	405	7V
9	444	7,7V
10	485	8,37V

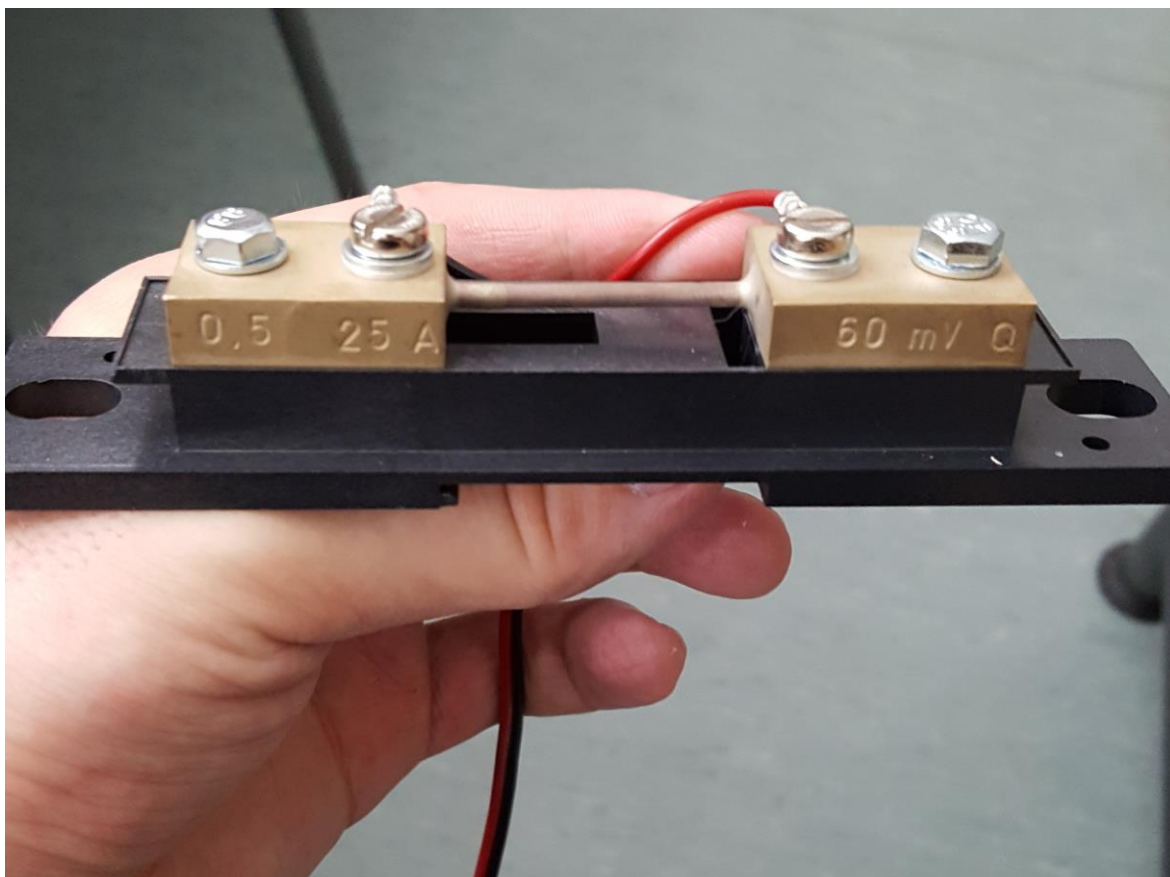
5 Diskussion

Det här arbetet har varit intressant och lärorikt men även utmanande. Det som var mest utmanande var att få allt att fungera som det ska t.ex. vissa skärmar fungerar inte med alla PLC:n och att veta vilken sorts signalförbättrare man ska använda samt att få alla inställningar rätt. PLC-programmeringen var enkelt att bygga upp men alarmsystemet borde dock förbättras.

Förbättringsförslag

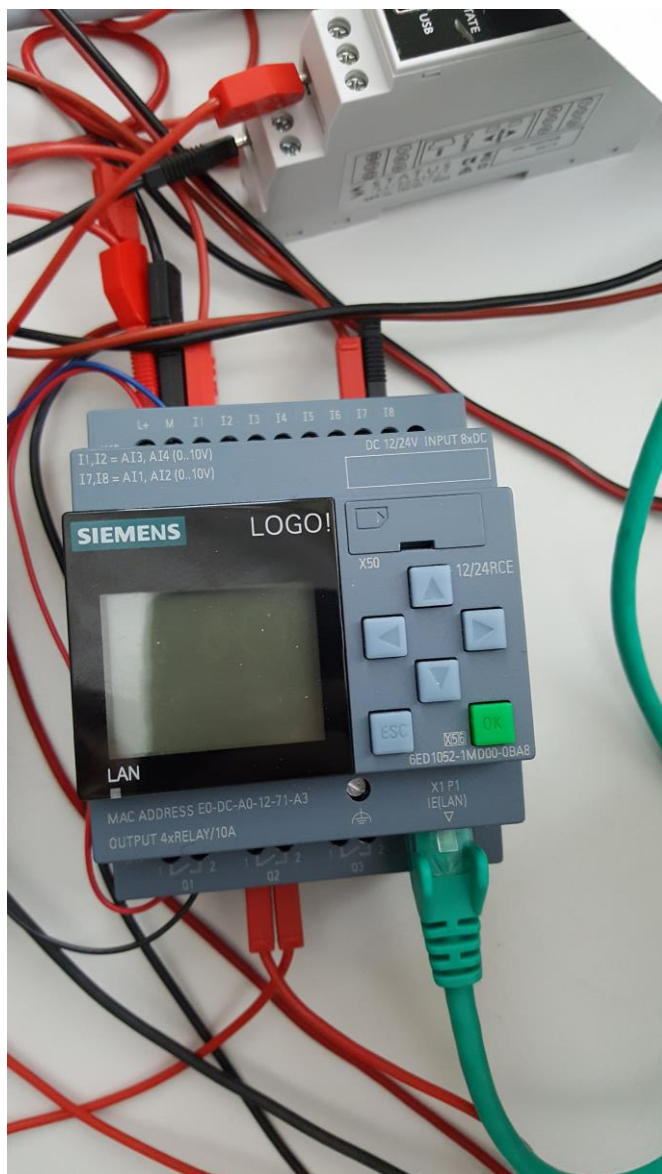
- Fläkt för varje låda för snabbare kyltid
- Alarmsystemet borde förbättras så att inget alarm aktiveras vid pauserna.

I figuren nedan ses shunten som jag använde i projektet.



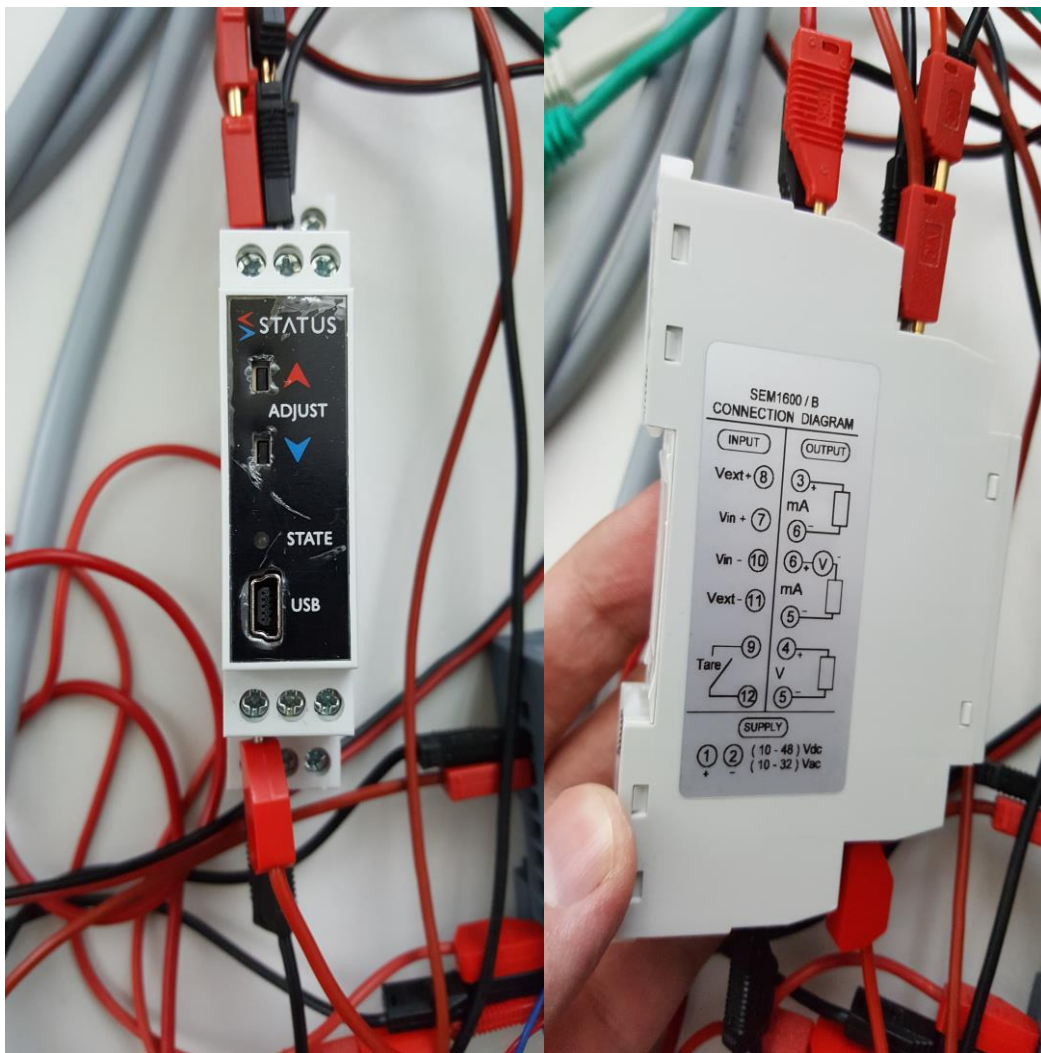
Figur 22. Shunt.

I figuren nedan ses PLC:n som jag använde i projektet.



Figur 23. PLC.

I figuren nedan ses signalförbättraren (SEM1600B) som jag använde i projektet.



Figur 24. SEM1600B.

I figuren nedan ses en hubb som jag använde i projektet.



Figur 25. TP-LINK.

6 Källförteckning

Ahlvik, Marcus (2012). Utveckling av testsystem för säkerhetsautomationssystem i kraftvärmeverk. Examensarbete för ingenjörsexamen. Yrkeshögskolan Novia, Vasa. [Online]

https://www.theseus.fi/bitstream/handle/10024/43447/Ahlvik_Marcus.pdf?sequence=1

(Hämtat 31.5.2018)

Bergström, Jens (2016). Vidareutveckling av sorteringsprocess. Examensarbete för ingenjörsexamen. Yrkeshögskolan Novia, Vasa. [Online]

<https://www.theseus.fi/bitstream/handle/10024/123001/Jens%20Bergstrmexamensarbete.pdf?sequence=1&isAllowed=y> (Hämtat 31.5.2018)

Electromate. HMI's & operator displays. [Online]

<https://www.electromate.com/hmis-operator-displays.html> (Hämtat 2.6.2018)

Globalspec. Programmable Logic Controllers (PLC) Information. [Online]

https://www.globalspec.com/learnmore/industrial_computers_embedded_computer_compo_nents/industrial_computing/programmable_logic_controllers_plcs (Hämtat 7.9.2018)

Globalspec. Signal conditioners information. [Online]

https://www.globalspec.com/learnmore/data_acquisition_signal_conditioning/signal_c_onditioning/signal_conditioners (Hämtat 2.6.2018)

Gonzalez, Carlos (2015). Engineering Essentials: What is a Programmable Logic Controller? [Online]

<https://www.machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller>

Mayank Parasrampur & Sandeep Jain (2014). Burn-in 101. [Online]

<https://www.edn.com/design/integrated-circuit-design/4435976/Burn-in-101> (Hämtat 15.6.2018)

McMahon, Russell (2016). What are the differences between USB and RS-232. [Online]

<http://electronics.stackexchange.com/questions/34549/what-are-the-differences-betweenusb-and-rs232> (Hämtat 1.6.2018)

Odesie. Programmable Logic Controller (PLC). [Online]

<https://www.myodesie.com/wiki/index/returnEntry/id/2962> (hämtat 9.9.2018)

Omron. Signal converters. [Online]

<https://www.ia.omron.com/support/guide/63/introduction.html> (Hämtat 2.6.2018)

PDH. Introduction to programmable Logic Controllers – Part 1. [Online]

<https://pdhonline.com/courses/e116/PLC-module1.pdf> (Hämtat 9.9.2018)

PLC Manual. PLC Manual. [Online]

<http://www.plcmanual.com> (Hämtat 8.9.2018)

RTA. USB. [Online]

<https://www.rtaautomation.com/technologies/usb/> (Hämtat 10.9.2018)

TechTerms. Ethernet. (2014) [Online]

<https://techterms.com/definition/ethernet> (Hämtat 10.9.2018)