



CHALMERS



# Styrssystem för en ny pump

Samt analys av förutsättningar för liknande projekt

Examensarbete inom högskoleingenjörsprogrammet Elektroteknik

Lukas Persson

INSTITUTIONEN FÖR ELEKTROTEKNIK

---

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2026

[www.chalmers.se](http://www.chalmers.se)

Styrsystem för en ny pump

Samt analys av förutsättningar för liknande projekt

Lukas Persson

© Lukas Persson, 2026.

Institutionen för Elektroteknik

Chalmers Tekniska Högskola

SE-412 96 Göteborg

I samarbete med SGS Studentbostäder

Sverige

Erkännanden, dedikationer och liknande personliga uttalanden återspeglar författarens egna åsikter.

Control system for control and monitoring of a new pump

With an analysis of prerequisites for similar projects

Lukas Persson

Department of Electrical Engineering

Chalmers University of Technology

### Sammanfattning:

Målet med detta projekt är att förbättra kunskapen och föreståelsen inom SGS Studentbostäder gällande den fastighetsautomation som används i deras byggnader. Projektet fokuserar på implementering av Modbus-kommunikation mellan en ny cirkulationspump och ett existerande styrsystem som ansvarar för byggnadens uppvärmning och ventilation. Det tekniska arbetet avser endast konfigurering av mjukvaran för styrsystemet och ett externt övervakningsprogram, inte den fysiska installationen av hårdvara.

Det primära syftet med etablering av Modbus-kommunikation mellan pumpen och styrsystemet var att övervaka väsentliga prestandarelaterade parametrar hos pumpen via ett externt program kopplat till styrsystemet. En undersökning gjordes också gällande styrning av pumpens driftläge via Modbus med detta externa program.

Data som skickades från pumpen möjliggjorde realtidsövervakning av pumpens prestanda i det externa övervakningsprogrammet. Styrning av pumpens driftlägen via det externa övervakningsprogrammet implementerades ej fullständigt på grund av en konflikt mellan Modbus och pumpens digitala ut- och ingångar som användes för start och stopp av drift samt andra praktiska problem.

Baserat på det tekniska arbetet gjordes en analys gällande styrsystemets funktionalitet och begränsningar. Som en del av analysen ingår också en utredning av SGS organisatoriska struktur och hur organisationen kan arbeta med och förhålla sig till fastighetsautomation i framtiden.

## Abstract:

This project aims to improve the knowledge and understanding within SGS studentbostäder regarding the automation and control systems used in their buildings. The project focused on implementing Modbus communication between a new circulation pump and an existing control system responsible for heating and ventilation in a residential building. The technical work only concerned configuring the software of the control system and an external monitoring program, not the physical installation of any hardware.

In enabling Modbus communication between the pump and the rest of the control system the primary goal was to monitor the performance of the pump via the external monitoring program. An investigation was also made to determine if it was possible to control the pump externally using the external monitoring program.

The data sent via Modbus from circulation pump to the external monitoring program contained performance data that enabled real-time monitoring of the performance of the pump. Writing to the internal memory of the pump from the external monitoring program was not fully implemented due to a conflict between Modbus and the digital in/outputs used to start and stop the pump.

Based on the technical work done, an analysis was done regarding the functionality and limitations of the control system. The analyses also examines the organisational structure of SGS to determine the need and feasibility to work internally with automation and control systems, or if they should continue outsourcing the installation and maintenance of these systems to external parties.

The report is written in Swedish.

**Keywords:** Modbus, control system, circulation pump.

# Innehållsförteckning

1 Inledning.....	1
1.1 Bakgrund .....	1
1.2 Syfte .....	1
1.3 Avgränsningar.....	1
1.4 Preciserar frågeställningen.....	2
2 Teknisk bakgrund .....	3
2.1 Fältbussprotokoll och Modbus.....	3
2.2 UltraBase20 – Hårdvara.....	6
2.3 UltraBase20 - Mjukvara.....	8
2.4 Web Port.....	9
2.5 MAGNA3 och CIM-200 .....	12
3 Systembeskrivning .....	13
4 Metod .....	14
5 Inläsning av data från MAGNA3 .....	15
5.1 Upprätta Modbus-kommunikation .....	15
5.2 Konfigurering av slav-registret.....	16
5.3 Konfigurering av Web Port taggar och objekt .....	17
6 Skriva data till MAGNA3 .....	19
6.1 Skicka taggvärden från Web Port.....	19
6.2 Bitvis skrivning till adressregister via Modbus .....	21
7 Lösning via användning av alternativt styrsystem .....	23
8 Resultat.....	26
9 Slutsats och diskussion .....	28
Referenser .....	29

# 1 Inledning

Det finns flera sätt att värma upp byggnader exempelvis genom att använda oljepannor, bergvärme eller fjärrvärme. I större tätorter med välutvecklad infrastruktur är fjärrvärme vanligt med stora centrala värmeverk som distribuerar varmvatten eller ånga via fjärrvärmerör i marken [1]. I byggnaderna används en värmeväxlare som överför värmen från fjärrvärmerören till byggnaden utan någon direkt överföring av vatten. I bostadens slutna rörsystem pumpas sedan det uppvärmda vattnet runt till radiatorer med hjälp av en cirkulationspump.

## 1.1 Bakgrund

SGS Studentbostäder (SGS) äger och förvaltar ett hundratal flerbostadshus med totalt cirka 6600 studentbostäder. Byggnadernas tekniska system för uppvärmning via fjärrvärme och ventilation styrs och regleras i huvudsak via datorer. SGS organisation inkluderar personal för teknisk drift och förvaltning av dessa datoriserade styr- och övervakningssystem. Personalens möjligheter att lägga till, ta bort och anpassa systemen begränsas av de användargränssnitt som installerats och som inte ger full tillgång till styr- och övervakningssystemens programvara. Om utrustning ska ändras eller om underliggande programkod ska modifieras krävs det att en extern aktör anlitas.

## 1.2 Syfte

Syftet med detta examensarbete är att via modifiering av programvaran som styr och övervakar en nyinstallerad cirkulationspump ge personal hos SGS bättre förståelse för hur fastighetens styr- och övervakningssystem fungerar samt ge underlag för framtida beslut inom organisationen gällande fastighetsautomation.

## 1.3 Avgränsningar

- Arbetet är begränsat till en byggnad på Vasagatan 12 och fokuserar främst på cirkulationspumpen MAGNA3.
- Arbetet avser endast modifikation av mjukvaran i relevanta styrsystem och programvara, ej den fysiska installationen av hårdvara.

## 1.4 Precisera frågeställningen

- Hur fungerar det fastighetsautomationssystem som används av SGS på Vasagatan 12 i dagsläget?
- Vilka begränsningar existerar kring modifikation av systemet?
- Skulle växling från en DUC till en PLC förenkla systemet eller möjliggöra ytterligare funktionalitet?
- Hur ser SGS organisatoriska struktur ut angående kompetens och kunskap kring fastighetsautomation?
- Vilka organisatoriska ändringar skulle behövas inom SGS för att en större del av fastighetsautomationen skulle kunna hanteras internt?

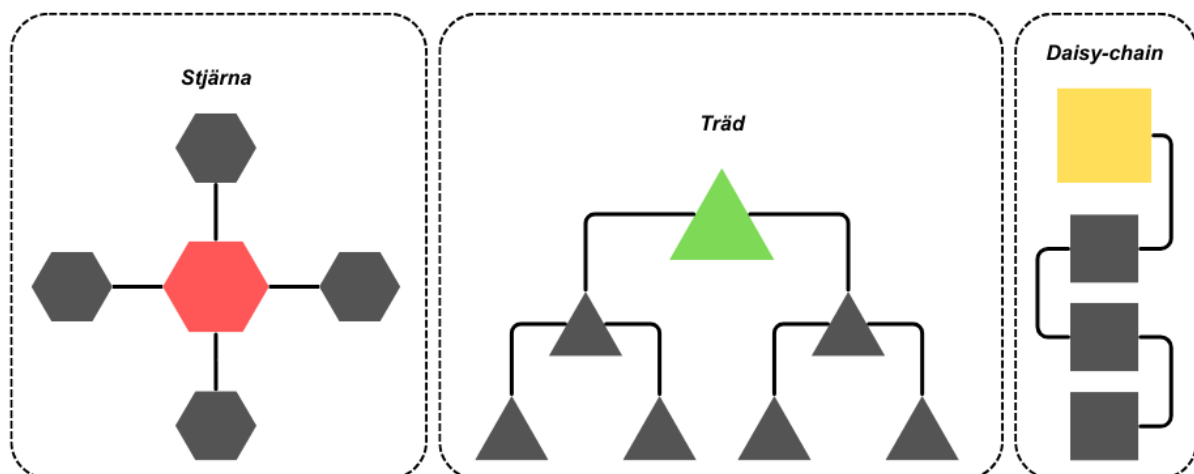
## 2 Teknisk bakgrund

Detta kapitel går igenom den tekniska bakgrunden för den hård- och mjukvara som används i detta projekt.

### 2.1 Fältbussprotokoll och Modbus

"*Fältbussprotokoll*" eller bara "*Fältbuss*" är ett samlingsbegrepp som refererar till en uppsättning av digitala industriella nätverksprotokoll [2]. Digitala industriella nätverksprotokoll är kommunikationsprotokoll utvecklade för att möjliggöra övervakning och styrning av flera separata ställdon och enheter samtidigt genom att koppla ihop dessa med en centralstyrenhet. Kommunikationen mellan enheter sker antingen via en vanlig serieportstandard som RS485 eller via ett Ethernetprotokoll.

Att fältbussprotokoll använder digital kommunikation möjliggör också för användning av vanliga nätverkstopologier som stjärna, träd och daisy-chain (se figur 1) [3]. Dessa topologier minskar komplexiteten av kablaget avsevärt då en enda styrenhet kan användas för att styra och kommunicera med flera separata enheter samtidigt. Ett av de vanligaste fältbussprotokollen är Modbus som finns i två olika versioner, Modbus RTU för kommunikation via serieport och Modbus TCP/IP för kommunikation via Ethernetport.

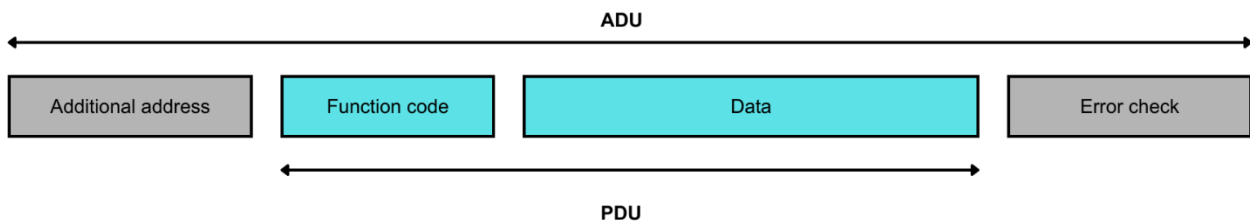


Figur 1: Diagram som visar tre vanliga nätverkstopologier; Stjärna, Träd och Daisy-chain.

Modbus följer en master/slave modell som baseras på en master som initierar en "*transaktion*" med en eller flera slavar som utför en handling baserat på transaktionens funktionskod och skickar tillbaka ett svar till mastern [4]. Modbus-organisationen har officiellt ändrat terminologin från master/slave till klient/server för Modbusprotokollet. Denna rapport kommer dock att för tydlighetens skull följa master/slave terminologin då denna terminologi används flitigt av annan mjukvara i detta projekt.

Modbusprotokollet definierar hur meddelandestrukturen för en transaktion mellan slav och master ska se ut uppdelat mellan en "*protocol data unit*" (PDU) och en "*application data unit*" (ADU) [4].

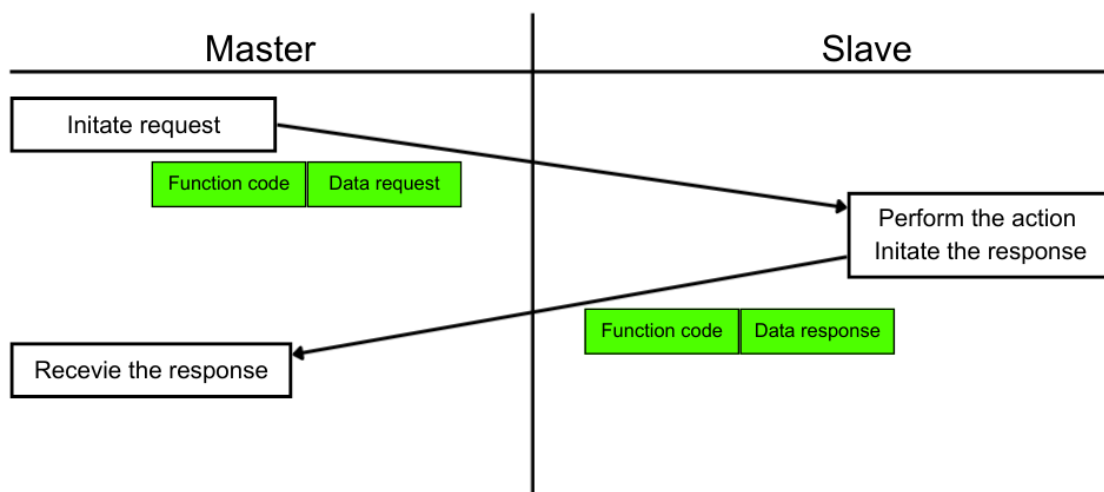
En PDU innehåller funktionskoden följt av ett datafält medan en ADU innehåller hela PDU:n och eventuellt ytterligare nätverk eller bussspecifika adresser samt felkontrollsbitar (se figur 2 nedan).



Figur 2: Diagram som visar meddelandestrukturen för Modbus.

Funktionskoden i en PDU är 1 byte lång med giltiga värden på 1-127 då 128-255 är reserverade för undantagshantering och "0" ej används [4]. Datafältet i en PDU innehåller ytterligare information som kan användas av slaven för att utföra det som specificeras av funktionskoden, ofta är detta adressen för det registret som ska behandlas och eventuellt skrivdata från mastern. Datafältet kan också lämnas tomt om funktionskoden ej kräver ytterligare information.

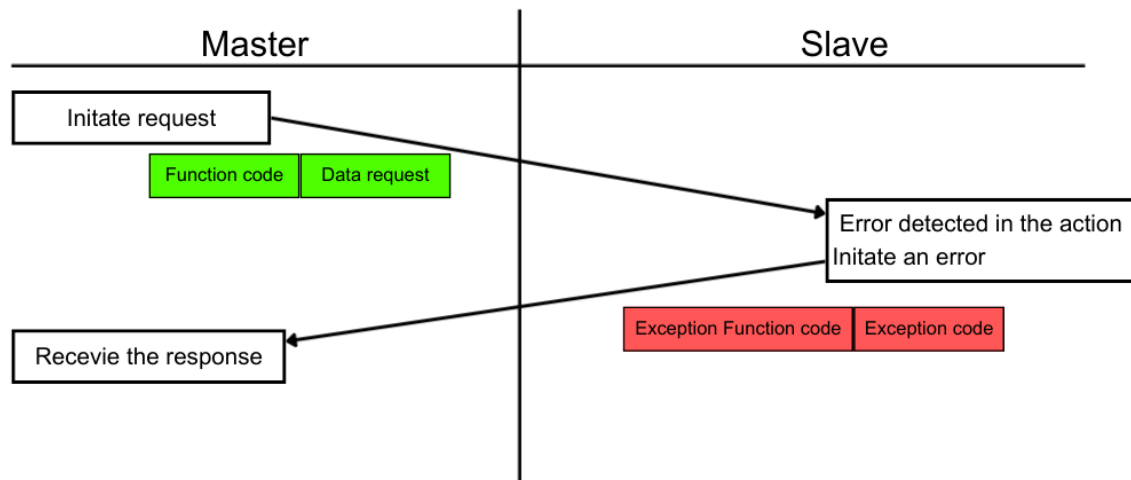
När en transaktion har initierats av en master och bearbetats av slaven utan fel kommer mastern att få ett svar från slaven innehållande den information som begärdes i datafältet med en tillhörande funktionskod som är den samma som skickades av mastern till slaven (Se figur 3).



Figur 3: Diagram som visar hur en transaktion utan fel mellan master och slav utförs med Modbus.

Om fel har uppstått så kommer datafältet innehålla en undantagskod som indikerar vad som gick fel och hur mastern bör hantera felet och funktionskoden kommer vara

den samma fast den mest signifikanta biten satt till ett för att indikera att fel uppstod (se figur 4).



Figur 4: Diagram som visar hur en transaktion då fel uppstår mellan master och slave utförs med Modbus

Modbus kategoriserar data som ska kommuniceras som fyra distinkta typer [4]:

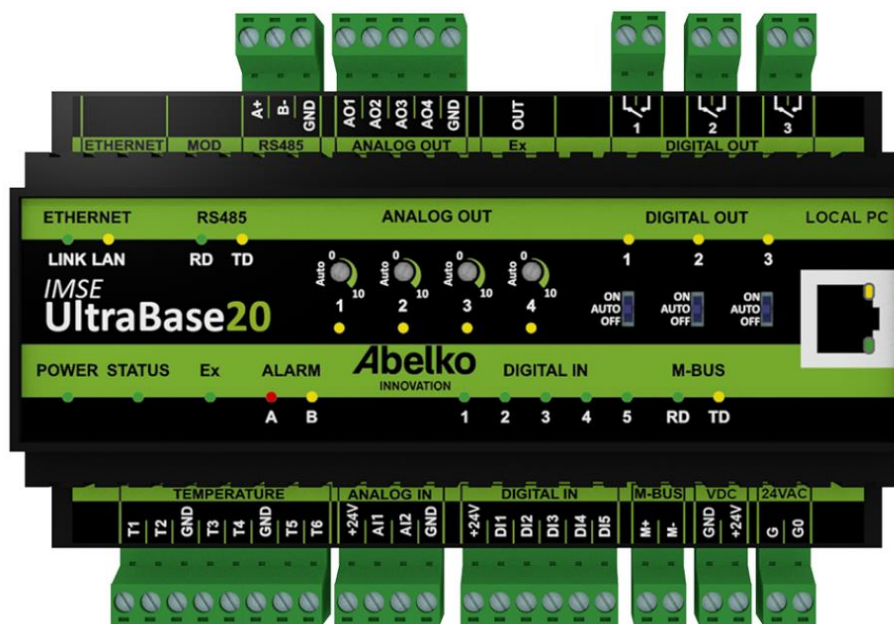
- Discrete input, 1-bit, endast läsbart.
- Coils, 1-bit, läs- och skrivbart.
- Input Registers, 16-bit word, endast läsbart.
- Holding Registers, 16-bit word, läs- och skrivbart.

Enheter konfigurerade för Modbus-kommunikation tillhandahåller sedan register i minne vars innehåll motsvarar till någon av de fyra datatyperna ovan. När ett meddelande skickas är det funktionskoden som specificerar vilket register, och då också vilken sorts datatyp, enheten ska bearbeta data i.

## 2.2 UltraBase20 – Hårdvara

UltraBase20 är en datorundercentral (DUC) tillverkad av Abelko Innovation. UltraBase20 har många likheter med en vanlig PLC men är annorlunda i och med att den är designad för fastighetsautomation och inte besitter samma bredd av funktionalitet som en typisk PLC [5]. UltraBase20:s syfte är att styra och övervaka fastigheters eller andra anläggningars olika givare, ställdon och andra kopplade enheter [6]. UltraBase20 är designad att endast fungera med ett tillhörande webbaserat mjukvaruprogram och programmeringsspråk, dessa också utvecklade av Abelko Innovation.

Den fysiska enheten är utrustad med en rad plintar på över- och undersidan för in- och utmatning av signaler från externa enheter. Ett antal lysdioder som indikerar funktionalitet eller status för enheten samt dess kontakter finns också på enhetens framsida (se figur 5). Det finns även möjlighet till att manuell styra "ANALOG OUT" signalerna via att vrida vreden mitt på enheten samt "DIGITAL OUT" signalerna via knapparna markerade med "ON", "AUTO" och "OFF". Alla kontakter hos enheten beskrivs i tabell 1 nedan.



Figur 5: Illustration av UltraBase20 [6]. Återgiven med tillstånd.

Tabell 1: Namnen på kontakter hos UltraBase20 och deras respektive beskrivningar.

<i>Namn</i>	<i>Beskrivning</i>
<i>ETHERNET</i>	Ethernetkontakt, för uppkoppling till externt nätverk.
<i>RS485</i>	RS485 porten, till för fältbuskommunikation.
<i>ANALOG OUT</i>	Analoga utgångar för 0-10 V med en upplösning på 10 bitar.
<i>Ex OUT</i>	Används för att koppla expansionsmoduler till enheten.
<i>DIGITAL OUT</i>	Digitala reläutgångar, Brytförmåga 250 VAC/2 A eller 30 VDC/2 A vid resistiv last.
<i>TEMPERATURE</i>	Resistansingångar för att mäta temperatur från temperaturgivare.
<i>ANALOG IN</i>	Analoga ingångar som kan användas till att mäta spänning mellan 0 och 10 V eller ström mellan 0-20 mA.
<i>DIGITAL IN</i>	Digitala ingångar, ingångsresistans 10 k $\Omega$ , Omslagsnivå inställbart mellan 4 och 12 V.
<i>M-BUS</i>	En Meter-Bus mini master-port.
<i>VDC/24VAC</i>	Strömförsörjning till enheten med antingen 24 V likspänning eller växelspanning.
<i>LOCAL PC</i>	Ethernetkontakt som kan användas till att koppla en dator direkt till enheten för att konfigurera den via dess mjukvara, ifall enheten ej är kopplad till ett externt nätverk.

## 2.3 UltraBase20 - Mjukvara

Mjukvaran till UltraBase20 är webbaserat och nås via att ansluta till enhetens IP-adress via en webbläsare eller via direkt anslutning till en dator med Ethernetkabel vid porten märkt "LOCAL PC" på enhetens framsida [6]. Gränssnittet är uppdelat i 7 flikar enligt tabell 2 nedan.

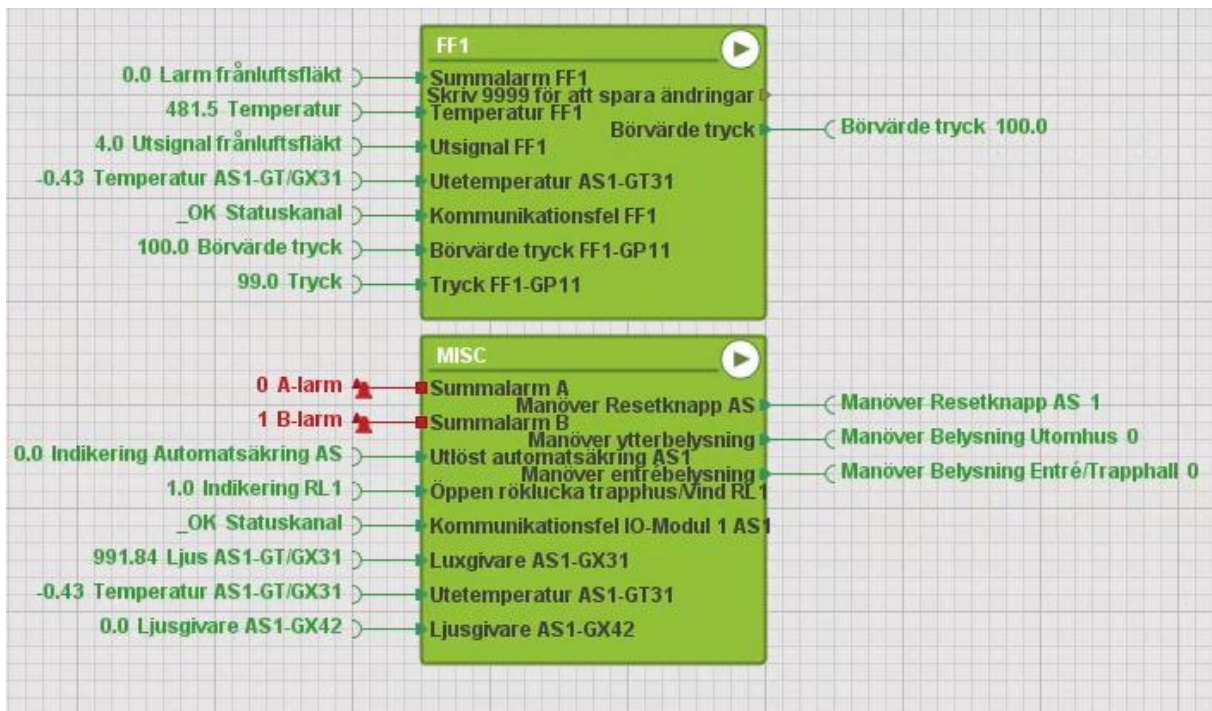
Tabell 2: Namn på flikar i gränssnittet för UltraBase20 och deras respektive beskrivningar.

<i>Namn</i>	<i>Beskrivning</i>
<i>Hemfliken</i>	Kan konfigureras till att visa översiktsbilder av systemet, viktiga larm, viktiga data-värden m.m.
<i>Larmfliken</i>	Listar alla larm, deras status, historik och grupperingar.
<i>Datafliken</i>	Kan konfigureras till att visa loggad data från olika givare.
<i>Kommunikationsfliken</i>	Konfigurering av nätverksinställningar för Ultrabase20 och fältbuss kommunikation mellan inkopplade enheter.
<i>Konfigureringsfliken</i>	Åtkomst till det grafiska programmeringsprogrammet, skapandet av översiktsbilder, konfigurering av UltraBase20 in- & utgångar, konfigurering av externa enheter för fältbusskommunikation, m.m.
<i>Systemfliken</i>	Inställningar för bl.a. backups, användare och uppdateringar
<i>Hjälpfliken</i>	Manualer skapade av Abelko samt kundsupport.

All mjukvara i UltraBase20 är skrivet i ett textbaserat programmeringsspråk, UltraScript, vars syntax är baserat på andra språk som Pascal, Modula2 och Structured text [7]. Unikt för UltraScript jämfört med andra vanliga programmeringsspråk är att det ej går att skapa for-loops eller while-loops samt indexerade matriser. Detta då dessa är "farliga" programmeringskonstruktioner som kan ledda till att programmet fastnar i en oändlig loop eller att matriser överindexeras vilket skaparna av UltraScript ha velat undvika då språket är skapat speciellt för industriella styrsystem där programmeringsfel kan få stora fysiska konsekvenser. UltraScript exekverar alla applikationer och skript en gång per sekund vilket innebär att ingen main-loop eller end behövs.

Programmering av styrsystemets logik kan göras antingen i en grafisk programmeringsvy som sedan översätts till UltraScript vid kompilering eller direkt i skriftlig form med UltraScript [7]. Programmeringen är hierarkisk uppdelad i olika applikationer med huvudapplikationer högst upp som kan innehålla under-applikationer, scripts, kanaler, larm etc (se figur 6). Endast huvudapplikationer kan

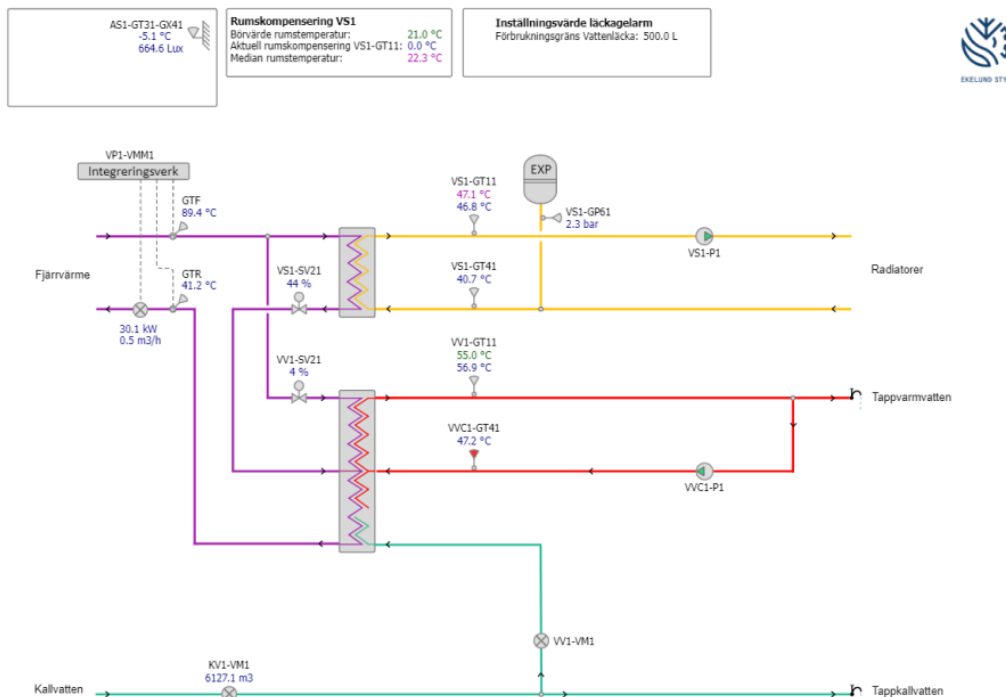
startas och stoppas självständigt med kravet att de innehåller ett syntaktiskt korrekt script. En applikations in- och utgångar kan också kopplas till andra applikationer.



Figur 6: Två olika huvudapplikationer (FF1 och MISC) med ingångar som går in i vänster, utgångar kommer ut till höger, grön färg på blocket indikerar att applikationen körs.

## 2.4 Web Port

Web Port är ett webbaserat program avsett för integration av olika styr- och övervakningsenheter till ett gemensamt användargränssnitt [8]. I Web Ports gränssnitt kan data från styrsystemen loggas, larm kvitteras och parametrar kopplade till styrsystemet kan ändras. Web Port kopplas ihop med olika styrsystem via en gemensam tagglista där alla värden, larm och annan information som skickas mellan Web Port och styrsystemet finns upplagda. All kommunikation mellan styrsystem och Web Port sker via Ethernet. Dessa taggar kan sedan kopplas i Web Port till olika dynamiska objekt som används för att skapa en schematisk bild av styrsystemet och med möjlighet för styrning av styrsystemets olika enheter samt loggning av data och larm som uppdateras i realtid (se figur 7).



Figur 7: Översiktvy av värmesystemet på Vasagatan 12 i Web Port.

Alla taggar i Web Port måste följa en specifik namngivningsstruktur uppdelat i ett prefix och ett suffix som används av Web Port programmet för att automatiskt koppla taggar till dynamiska objekt och andra funktioner som larm och trenddiagram [8]. Prefixet anger vilken enhet taggen är kopplat till, suffixet anger funktionen på taggen.

Exempelvis taggen *VS1\_GT41\_PV*, här är *VS1\_GT41* prefix och anger att systemet/applikationen är VS1 och GT41 anger vilken givare/enhet det är. PV (Present Value) är suffixen och indikerar att den visar det aktuella värdet hos givaren.

Alla taggar definieras i tagglistor som är uppdelade i en rad kolumner som bestämmer taggens funktion, se tabell 3 och figur 8 nedan.

Namn	IO-Enhet	Adress	Datotyp	Rå-min	Rå-max	Vy-min	Vy-max	Enhet	Format	Beskrivning	Typ	Värde
AS1_GT31TEST_PV	AS1TEST	3.00001	INT	-32000	32000	-3200	3200	°C	0.0	Utetemperatur AS1-GT31		-2.9 °C
AS1_GT31_AL1	AS1	2.00852	DIGITAL	0	0	0	0			Givarfel AS1-GT31-GX41		0
AS1_GT31_PV	AS1	3.00001	INT	-32000	32000	-3200	3200	°C	0.0	Utetemperatur AS1-GT31		-2.9 °C
AS1_GT31_PV1	AS1	3.00920	INT	-32000	32000	-3200	3200	Lux	0.0	Ljuskivare AS1-GX41		927.1 Lux
AS1_GX42_PV	AS1	3.00934	INT	-32000	32000	-3200	3200	Lux	0.0	Ljuskivare AS1-GX42		0.0 Lux
FF1_AL1	AS1	2.00351	DIGITAL	0	0	0	0			Summalarm FF1		0
FF1_GP11_AD	AS1	3.00432	INT	-32000	32000	-3200	3200	Pa	0.0	Max regleravvikelse FF1-GP11		(5.0 Pa)
FF1_GP11_ADL	AS1	3.00433	INT	-32000	32000	-3200	3200	m	0.0	Larmfördröjning regleravvikelse FF1-GP11		(30.0 m)
FF1_GP11_AL	AS1	2.00352	DIGITAL	0	0	0	0			Regleravvikelse FF1-GP11		0
FF1_GP11_AL1	AS1	2.00350	DIGITAL	0	0	0	0			Kommunikationsfel FF1-GP11		0

Figur 8: Inuti en tagglista i Web Port, samtliga inställningar för varje tagg uppdelade i kolumner.

Tabell 3: Lista på taggars inställningar och deras beskrivningar i Web Port.

Inställning	Beskrivning
<b>Namn</b>	Namnet på taggen, är uppdelat enligt en taggstandard med prefix och suffix.
<b>IO-Enhet</b>	Anger vilken IO-enhet taggen är kopplad till.
<b>Adress</b>	Anger adress och funktionskod för taggen, bör vara samma som i slavregistret hos Abelko enheten.
<b>Datotyp</b>	Taggens datatyp.
<b>Rå min/max</b>	Minsta och största värdet som kan läsas från IO-enheten.
<b>Vy min/max</b>	Minsta och största värdet som kan visas i Web Port.
<b>Enhet</b>	Enheten för värdet på taggen, tex Rpm eller %.
<b>Format</b>	Anger taggens formatering, tex antal decimaler.
<b>Beskrivning</b>	Taggens beskrivning, används automatiskt i dialoger, larmtext m.m.
<b>Typ</b>	Indikerar om taggen är ett larm eller om värdet lagras för ett trenddiagram.
<b>Värde</b>	Värdet på taggen, kan klicka för att temporärt aktivera automatisk uppdatering av värdet.

Vy min/max tillsammans med Rå min/max används även för att skala värdet från taggen, exempelvis Rå min/max = 0/10000 och Vy min/max = 0/1000 leder till att taggens värde skalas ner med en tiopotens.

De dynamiska objekt som används för att skapa översiktbilder i Web Port är definierade i en JSON skriptfil, "lib.conf", som finns i ett symbolbibliotek [8]. Inuti

denna JSON-fil finns alla funktioner för ett objekt, tillhörande figur, tillstånd etc. Utöver objekt definieras även de suffix som kan användas av taggar i denna skriptfil.

## 2.5 MAGNA3 och CIM-200

MAGNA3 är en cirkulationspump tillverkad av Grundfos som är konstruerad för att cirkulera vätska i olika typer av värmesystem [9]. Den har olika funktioner såsom olika driftlägen, en del med tillhörande algoritmer för att optimera livslängd, energikonsumtion och flöde, samt möjlighet att logga prestandarelaterade datavärden som flöde, tryck och varvtal. Utöver dessa finns säkerhetsfunktioner inbyggda som larmar och/eller stoppar pumpen om större fel inträffar. Via en terminal på pumpens framsida kan all funktionalitet konfigureras direkt på plats om extern styrning ej är möjlig eller nödvändig (se figur 9).



Figur 9: Cirkulationspumpen MAGNA3:s framsida med dess terminal bestående av en skärm och navigationsknappar, installerad på Vasagatan 12.

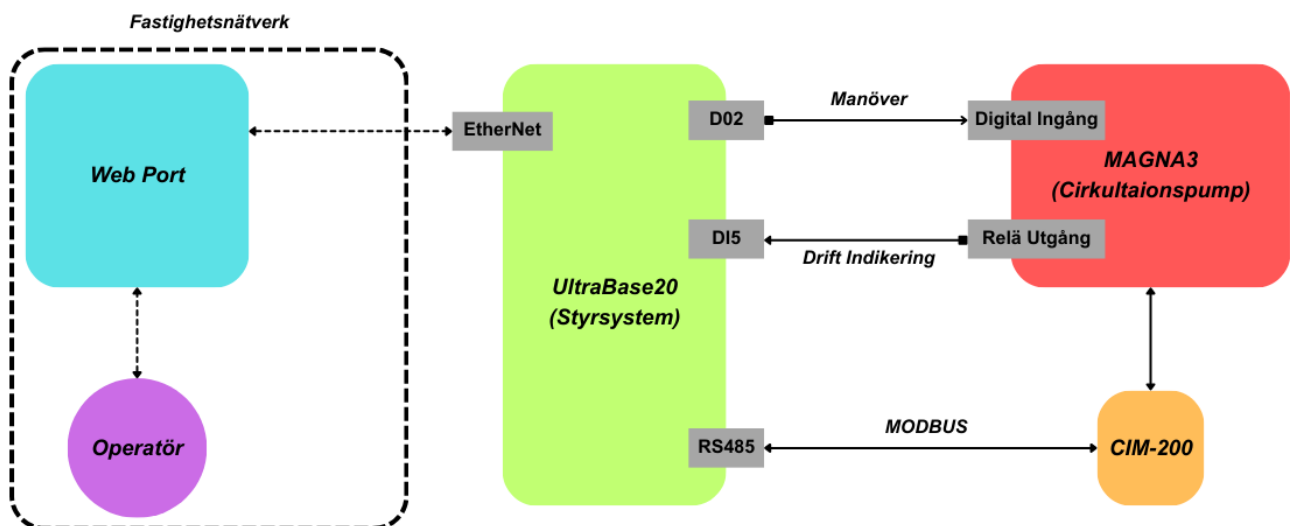
Längst upp på pumpens framsida finns även en indikeringslampa, "Grundfos Eye", som indikerar driftstatus för pumpen via olika sekvenser av blinkningar i olika färger. Start/stopp av pumpen kan göras direkt via terminalen eller externt, antingen via en digital in- och utgång hos pumpen eller, om en extra separat CIM-modul installerats, kan styrning av start/stop samt läsning av mätvärden göras via ett fältbussprotokoll istället [9]. För kommunikation via fältbussprotokollet Modbus är det en CIM-200 modul som krävs.

### 3 Systembeskrivning

Styrsystemet som använts i detta projekt är en UltraBase20 med tillhörande mjukvara. För att göra styrning och övervakning av systemet mer intuitivt och användarvänligt används även det externa programmet Web Port som kommunicerar med styrsystemet UltraBase20 över ett internt fastighetsnätverk.

I UltraBase20:s grafiska programmeringsmodul finns all styrlogik för systemet, uppdelat i olika huvudapplikationer med in- och utgångar kopplade till de fysiska plintarna på UltraBase20:n. Den fysiska kopplingen mellan plintarna och applikationernas in/utgångar görs möjlig via konfigurering av styrsystemets mjukvara, specifikt under fliken "In- & utgångar" och "externa enheter".

UltraBase20 är kopplad till Cirkulationspumpen MAGNA3 via digitala in- och utgångar för start och stopp av drift samt driftindikering av pumpen. En CIM-200 modul är också kopplad externt till MAGNA3 som möjliggör Modbus-kommunikation via RS485 porten på styrsystemet (se figur 10). UltraBase20 kopplas ihop med Web Port via ett Modbus-slavregister hos styrsystemet som laddas upp som en tagglista i Web Port. Slavregistret och tagglistan innehåller en identisk uppsättning av adresser och funktionskoder för den data som kommuniceras via Ethernet över det interna fastighetsnätverket. I Web Port kan sedan delar av styrsystemet styras och övervakas via ett användarvänligt gränssnitt som inkluderar en översikt av hela värme- och ventilationssystemet.



Figur 10: Diagram som visar hur MAGNA3 är kopplat till det externa styrsystemet UltraBase20 och Web Port.

## 4 Metod

Funktionalitet och relevanta läs -och skrivbara data-värden via tillverkarens Modbus-dokumentation för MAGNA3 [10] identifierades och en kravspecifikation utformades och godkändes sedan av SGS.

De tekniska kraven var:

- Läs in data från adresserna specificerade i tabell 4 nedan från cirkulationspumpen, MAGNA3, och visa denna data i översiktssyn för värmesystemet i Web Port.
- Undersök möjligheten att ändra cirkulationspumpens driftläge från Web Port via Modbus.

Tabell 4: Läsbara Modbus adresser hos MAGNA3.

Modbus Adress	Registernamn	Beskrivning
00203	ControlMode	Indikerar aktuellt driftläge.
00204	OperationMode	Indikerar aktuellt operand läge.
00301	Head	Indikerar aktuellt tryck i systemet.
00302	VolumeFlow	Indikerar aktuellt flöde i systemet.
00303	RelativePerformance	Indikerar aktuell prestanda relativt maximalt möjligt.
00304	Speed	Indikerar aktuellt varvtal.

Baserat på det tekniska arbetet utformades en rapport, avsedd för att nyttjas inom SGS, som redogör hur den tekniska implementeringen utfördes och hur mjukvaran till styrsystemet UltraBase20 och Web Port generellt fungerar. Till denna rapport gjordes också en organisatorisk analys av SGS vars ändamål var att undersöka hur SGS skulle kunna jobba mer med fastighetsautomation i framtiden.

Den tekniska delen av arbetet som inkluderade modifikation och konfigurering av UltraBase20 och Web Port utfördes via fjärranslutning till SGS interna fastighetsnätverk. Under arbetets gång hölls regelbunden kontakt med personal från Ekelunds Styr som hade relevant expertis gällande de installerade styrsystemets hård -och mjukvara. Detta då Ekelunds Styr är den styrentreprenad som har installerat all hård- och mjukvara som utgör styrsystemet i den bygganden som detta projekt utgår ifrån. I projektet användes också programmet Codesys för att visa hur en alternativ lösning skulle kunna implementerats i en PLC som inte har av samma begränsningar som UltraBase20.

Den skriftliga SGS rapporten som innehöll hur det tekniska arbetet utfördes, hur de olika involverade systemen fungerar och den organisatoriska analysen av SGS presenterades och överlämnades vid projektets slut till personal på SGS.

## 5 Inläsning av data från MAGNA3

För att övervaka de parametrar som listas i kravspecifikationen måste ett antal adresser i MAGNA3:s minne läsas och överföras via Modbus till styrsystemet UltraBase20.

### 5.1 Upprätta Modbus-kommunikation

För att etablera Modbus-kommunikation mellan MAGNA3 och UltraBase20 så skapades en ny extern enhet i UltraBase20:s programvara under *konfigureringsfliken*. Namn på den nya externa enheten blev *Grundfors* som gavs tekniska kommunikationsparametrar enligt etablerad praxis från Ekelunds Styr:

- Baudrate: 9600
- Antal bitar: 8
- Paritet ingen
- Stoppbitar: 2
- Timeout [ms]: 1000
- Teckentimeout: 4
- Meddelandefördröjning [ms]: 4.167

De Modbus-adresser som identifierades i kravspecifikationen som relevanta övervakningsparametrar hos MAGNA3 lades till hos den externa enheten *Grundfors* enligt figur 11 nedan:

**Redigera extern enhet**

Namn: Grundfors Adress: 11

Baudrate: 9600 Antal bitar: 8 Paritet: Ingen Stoppbitar: 2 Timeout [ms]: 1000 Teckentimeout: 4

Meddelandefördröjning [ms]: 4.167

Register	Namn	FC	RW	Typ	Skala	Uppdateringstid	Värde	Enhet
203	ControlMode	3	R	Int 16	1	10 s	128.0	X
204	OperationMode	3	R	Int 16	1	10 s	0.0	X
301	Head	3	R	Int 16	100	5 s	1.93	bar X
302	VolumeFlow	3	R	Int 16	10	5 s	3.8	m3/h X
303	RelativePerforman	3	R	Int 16	100	5 s	69.7	% X
304	Speed	3	R	Int 16	1	5 s	2105.0	rpm X

Lägg till

Ladda upp: Bläddra... Exportera: JSON CSV

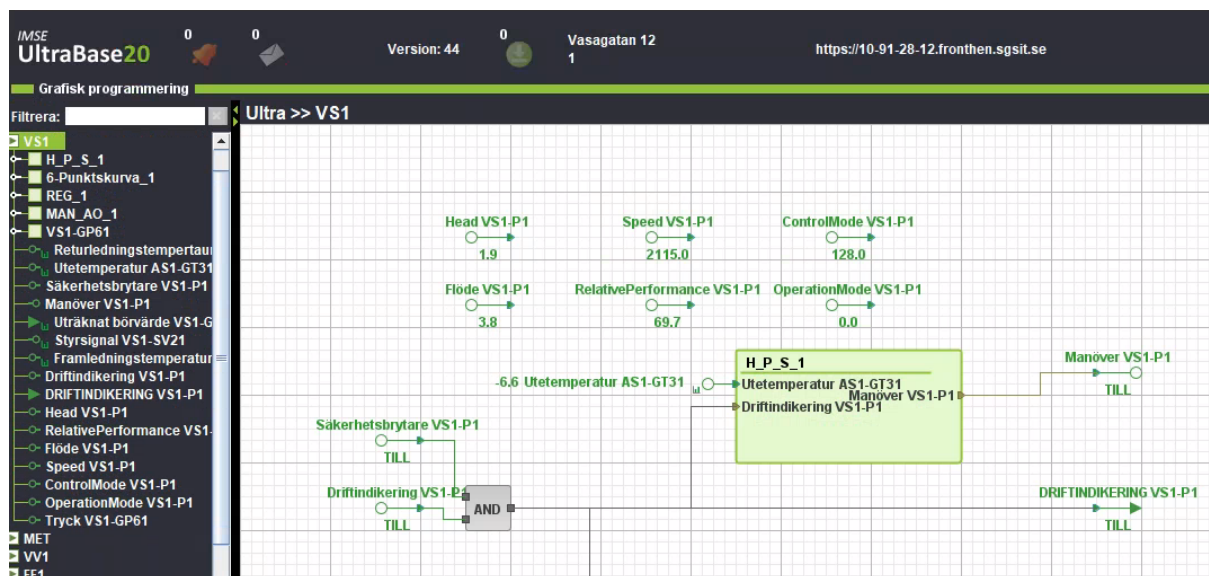
Stäng Spara

Figur 11: Kommunikationsinställningar och lista på de register som avläses från MAGNA3.

Parametrarnas registernummer, namn, funktionskoder, skalning, typ, och enhet valdes i enlighet med MAGNA3:s Modbus konfigureringsmanual [10]. Uppdateringstiderna gavs värden utifrån rekommendationer från Ekelunds Styr.

## 5.2 Konfigurering av slav-registret

För att kunna skicka de nya parametrar vidare via ett slav-register krävdes det att samtliga paramaterar lades till som kanaler i huvudapplikationen för värmesystemet i den grafiska programmeringsvyn. Detta gjordes genom att lägga till parametrarna som *kanaler* inuti huvudapplikationen för värmesystemet VS1 genom att dra in dem från den vänstra menyn i gränssnittet till den rutade arbetsytan. Då dessa värden bara skulle läsas och inte användas för ytterligare logik inuti applikationen kopplades de inte vidare till det övriga programmet (se figur 12).



Figur 12: Den grafiska programmeringsvyn med de nya datavärdena från MAGNA3 utsatta.

Då UltraBase20:n alltid låg i aktiv drift var det nödvändigt att först avaktivera *Modbus slav-registret* för att tillåta ändring av dess innehåll. Väl avaktiverat lades de nya parametrarna till längst ner i registerlistan som tillhörde VS1 applikationen (se figur 13).

Registernummer	Resurstyp	Namn	Applikation	Typ	Skala	Läsa/Skriva
00042	Kanal.Värde	Flöde VS1-P1	VS1	Int 16	10	Läsa/Skriva
00043	Kanal.Värde	Head VS1-P1	VS1	Int 16	10	Läsa/Skriva
00044	Kanal.Värde	Speed VS1-P1	VS1	Int 16	10	Läsa/Skriva
00045	Kanal.Värde	RelativePerformance VS1-P1	VS1	Int 16	10	Läsa/Skriva
00046	Kanal.Värde	OperationMode VS1-P1	VS1	Int 16	1	Läsa/Skriva
00047	Kanal.Värde	ControlMode VS1-P1	VS1	Int 16	1	Läsa/Skriva

Figur 13: UltraBase20:s slavregistret innehållande datavärden från MAGNA3.

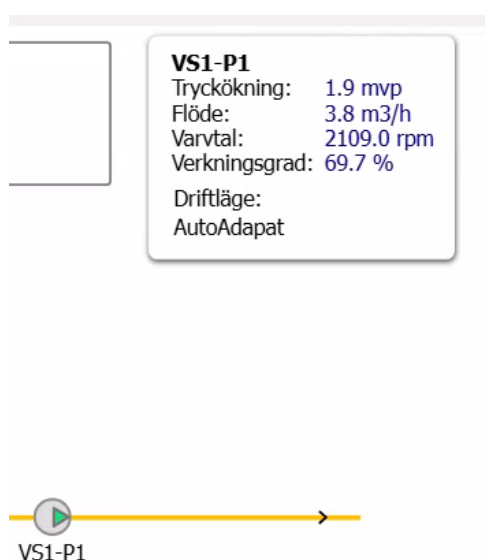
## 5.3 Konfigurering av Web Port taggar och objekt

I Web Port programmet redigerades tagglistan via tillägg av de nya parametrarna som gavs samma adress, enhet och datatyp som skrevs i slav-registret men med nya tagg-namn, eftersom taggnamnen måste följa Web Ports namngivningsstruktur gällande suffix och prefix. Då ett dynamiskt objekt och taggar för driftindikering hos pumpen, VS1-P1, redan fanns och SGS önskade visa de nya parametrarna separat från driftindikeringssymbolen så gavs de nya taggarna ett annat prefix än VS1\_P1 så att de inte automatiskt blev länkade till detta objekt (se figur 14).

Namn	IO-Enhet	Adress	Datatyp	Rå-min	Rå-max	Vy-min	Vy-max	Enhet	Format	Beskrivning	Typ	Värde
XVS1_P1_PV1	AS1	3:00043	INT	0	10000	0	1000	mvp	0.0	Tryckökning		1.9 mvp
XVS1_P1_PV2	AS1	3:00042	INT	0	10000	0	1000	m3/h	0.0	Flöde		3.7 m3/h
XVS1_P1_PV4	AS1	3:00045	INT	0	10000	0	1000	%	0.0	Verkningsgrad		69.7 %
XVS1_P1_PV5	AS1	3:00044	INT	0	10000	0	1000	rpm	0.0	Varvtal		2120.0 rpm
YVS1_P1_CMD	AS1	3:00047	INT	0	0	0	0		0	Aktuellt kontrolläge		128

Figur 14: Nya taggar i Web Port med nya prefix XVS1 och YVS1 men med samma adress, funktionskod och enhet som i slavregistret för UltraBase20.

Ett dynamiskt objekt kallat *Value* lades till i översikt bilden för värmesystemet (Se figur 15). Detta nya "Value" objekt redigerades med ett skript så att taggarna listades i rätt ordning, i detta fall listas PV1 först följt av PV2, PV5 och PV4 (Se figur 16).



Figur 15: En del av översiktsvyn av värmesystemet med de nya parametrarna separat från driftindikeringssymbolen för pumpen VS1-P1.



Figur 16: Inställningsfönstret för "Value" objektet med skript tillagt under "Eget värde".

Då Läsning av MAGNA3:s driftläge ger ett heltal som motsvarar det aktuella driftläget, blev det nödvändigt att översätta detta heltal till text via ett skript. För att åstadkomma detta gavs denna parameter sitt eget prefix, *YVS1\_P1*, skilt från de andra nya taggarna som använde *XVS1\_P1*, eftersom Web Port inte stödjer flera olika skript för samma objekt samtidigt. Översättningen sker genom att associera ett heltal med en textsträng. Även här användes ett "Value" objekt med tillhörande skript (Se figur 17).

**Beskrivning- och tagginställningar**

Visa beskrivningar  Visa värden

Värde- och beskrivningsposition  
Samma som textposition

Justering värde  
Vänster

Förskjutning horisontal  
-25.3308

Förskjutning vertikal  
219.708

Storlek på text  
12

Fet text

Beskrivningsfärg  
rgb(0, 0, 0)

Bakgrundsfärg  
rgba(0, 0, 0, 0)

Egen stil

Eget värde  
list#\_cmd = 128 : \_cmd = AutoAdapat; \_cmd=4 : \_cmd =  
Konstant tryck; \_cmd=7 : \_cmd = Konstant flöde;

Figur 17: Inställningsfönster för "Value" objektet med skript tillagt under "Eget värde".

## 6 Skriva data till MAGNA3

För att en operatör i Web Port ska kunna styra cirkulationspumpen MAGNA3 krävs att data först skickas från Web Port till UltraBase20 och sedan från UltraBase20 till cirkulationspumpen.

### 6.1 Skicka taggvärden från Web Port

För att skicka data från Web Port till styrsystemet UltraBase20 och vidare till cirkulationspumpen MAGNA3 redigerades först definitionen för en oanvänd standardfunktion "F15" i JSON filen lib.conf i Web Ports symbolbibliotek (se figur 18).

```
472     "F15": {
473         "Description": "Kontrolläge",
474         "Type": "Dropdown",
475         "Action": "AutoAdapt:CMD=128|ConstantFlow:CMD=7|ConstantPressure:CMD=4",
476         "Level": "Driftläge",
477         "Access": 3
478     }
```

Figur 18: JSON skriptfilen lib.conf med definition av funktionen F15.

Funktioner i Web Port följer alla en grundläggande syntaktisk JSON struktur som kan ses i funktionsdefinitionen för "F15":

- "F15": Funktionens id, måste vara unikt från övriga funktioner i JSON filen.
- "Description": Beskrivning av funktionen, visas intill inställningsrutan.
- "Type": Definierar funktionens typ som kan vara en av nio förbestämda sorter, exempelvis *Dropdown* eller *Setpoint* för flervalstypen respektive inskrivning av ett börvärde.
- "Action": Kodning av funktionens beteende.
- "Level": Olika funktioner beroende på funktionens typ. För en dropdown-menyn bestämmer *Level* den text som står i funktions fliken.
- "Access": Access nivån för funktionen.

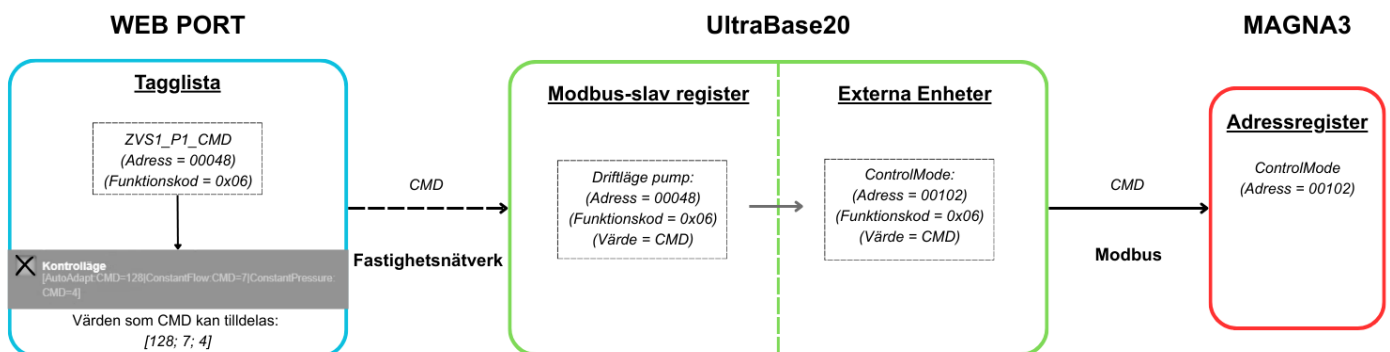
Den nya funktionen "F15" definieras som en dropdown-menyn med tre val, "AutoAdapt", "ConstantFlow" och "ConstantPressure". En tagg med suffixet *CMD* som är kopplat till ett objekt där denna funktion är aktiverad kommer anta det värde som menyvalet associeras med i JSON filen. Exempelvis val av AutoAdapt kommer sätta *CMD* taggens värde till 128. De valbara värden taggen kan anta i detta fall överensstämmer med de värden som MAGNA3 tolkar som olika driftlägen enligt dess Modbus-dokumentation (se figur 19 nedan).

00102	ControlMode	<p>Control bit that sets the control mode enumeration          Note that some modes are not supported by all E-pumps.</p> <p>0: Constant speed          1: Constant frequency          3: Constant head          4: Constant pressure          5: Constant differential pressure          6: Proportional pressure          7: Constant flow          8: Constant temperature          9: Constant differential temperature          10: Constant level          128: AUTO<sub>ADAPT</sub>          129: FLOW<sub>ADAPT</sub> (set FLOW<sub>LIMIT</sub> in register 00106)          130: Closed-loop sensor          131: AUTO<sub>ADAPT</sub>(CP)</p> <p>You can read the actual control mode from register 00203.</p>
-------	-------------	---

Figur 19: Modbus-dokumentation för MAGNA3 som visar adress 00102 med listade driftlägen till höger med sina respektive associerade heltal (0-131).

För att säkerställa att skrivning till MAGNA3 från en tagg i Web Port var tekniskt möjligt skapades en ny tagg med ett unikt prefix (ZVS1\_P1) och suffixet CMD. Taggen gavs funktionskoden 0x06 och ett objekt kopplat till taggen via ett identiskt prefix skapades i översiktsvyn för värmesystemet. Ovannämnda tagg-funktion "Kontrolläge" aktiverades för detta objekt så att taggen antar ett heltalsvärde som tolkas av MAGNA3 som ett driftläge.

För att skicka detta värde till UltraBase20:n redigerades dess Modbus slav-register med tillägg av en ny parameter med samma namn, adress och funktionskod som ovannämnda tagg i Web Port. Den externa enheten "Grundfors" redigerades sedan med tillägg av ett nytt register 00102 med funktionskod 0x06. Figur 20 visar ett diagram över denna process.



Figur 20: Diagram som visar hur värdet på en tagg i Web Port skickas till UltraBase20 och slutligen skrivs till ett register hos cirkulationspumpen MAGNA3.

## 6.2 Bitvis skrivning till adressregister via Modbus

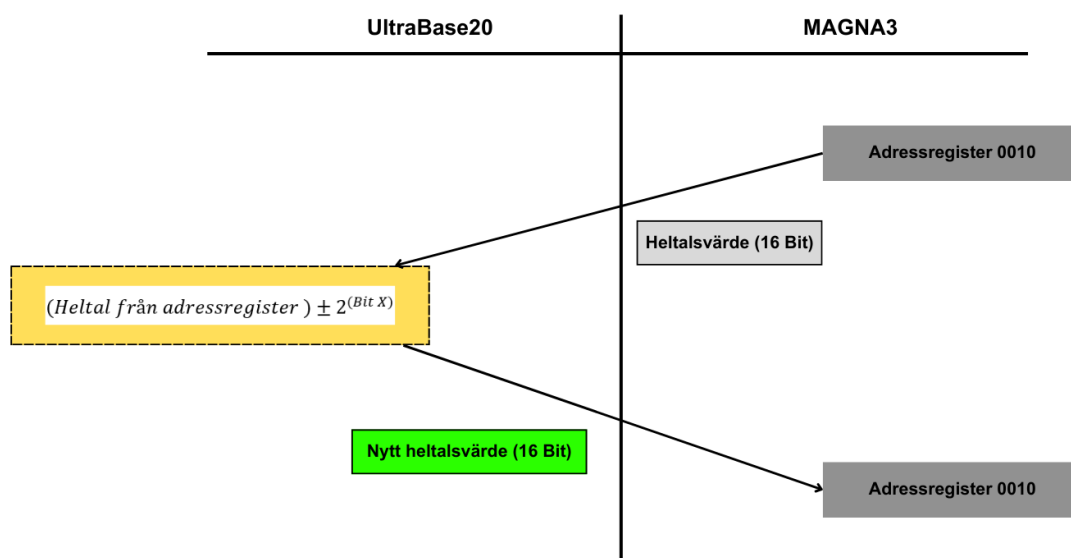
För att styra all funktionalitet hos cirkulationspumpen via Modbus krävs i vissa fall att man skriver en etta eller nolla till en specifik bit hos ett adressregister i pumpens minne. Exempelvis för start/stopp krävs det att bit 1 hos adressregistret 00101 ett/nollställs. UltraBase20 har dock bara möjlighet att skriva 16 eller 32 bitars *Word* eller *Integer* till en extern enhet. Vid ändring av en ensam bit i ett adressregister på 16 bitar krävs då att ett decimalt heltal skickas till adressen som motsvarar en korrekt uppsättning av ett/nollställda bitar i registret för adressen. För att inte ändra andra bitar av misstag krävs följande:

- Det relevanta adressregistrets heltalsvärde läses av.
- Addera/subtrahera det heltalsvärde som motsvarar bitändring för relevanta bitar till det värde som lästes in från adressregistret.
- Skicka detta heltalsvärde till adressregistret hos den externa enheten via Modbus.

Den generella realisering av att ta fram heltalsvärdet som motsvarar ett/nollställning av en bit visas i ekvation 1:

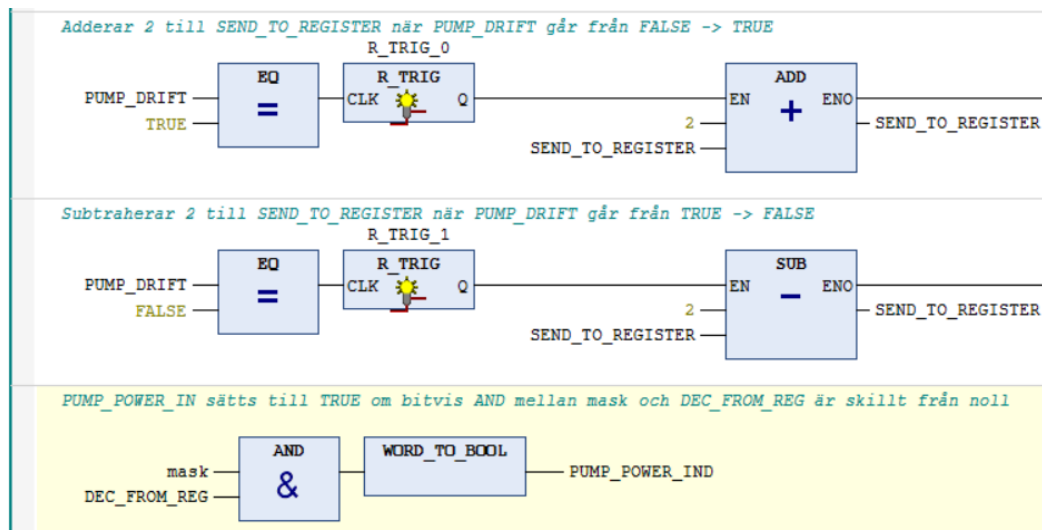
$$(Heltal \text{ från adressregister} ) \pm 2^{(Bit X)} \quad (1)$$

I praktiken för start/stopp av MAGNA3 är det dock enklast att använda konstanten 2 som motsvarar ett/nollställning av bit 1 som startar/stoppar pumpen. Flödesschema över den generella processen syns i figur 21 nedan.



Figur 21: Diagram över en generell metod för ett/nollställning av godtycklig bit X hos register 0010.

I programmet Codesys implementerades den praktiska lösningen av läsning och skrivning till en individuell bit i ett register med FBD-kod (se figur 22).



Figur 22: FBD-kod i Codesys för bitvis skrivning och läsning av ett 16-bitars heltal.

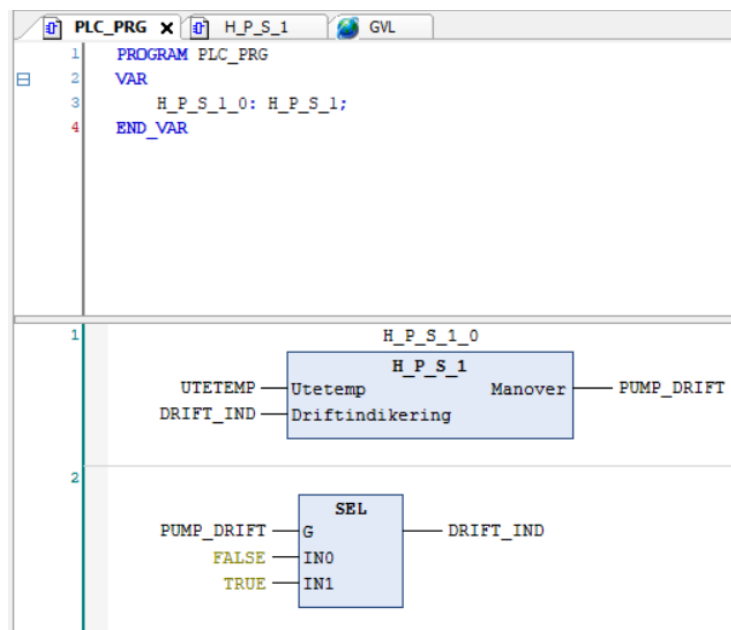
*PUMP\_DRIFT* är en boolesk variabel som indikerar om pumpen ska sättas i drift eller ej baserat på övrig logik hos programmet. Baserat på tillståndet hos *PUMP\_DRIFT* aktiveras ett ADD/SUB-block som adderar eller subtraherar ett heltal som motsvarar korrekt bitändring för drift hos pumpen, bit 1 i detta fall.

*DEC\_FROM\_REG* innehåller ett heltal från ett av pumpens register där varje bit indikerar status för pumpens olika funktioner. En mask används för att läsa av statusen på bit 9 i detta register via bitvis AND då denna bit indikerar om pumpen är i drift eller ej. *PUMP\_POWER\_IND* sätts till sant då bit 9 hos *DEC\_FROM\_REG* är lika med ett och vice versa.

## 7 Lösning via användning av alternativt styrsystem

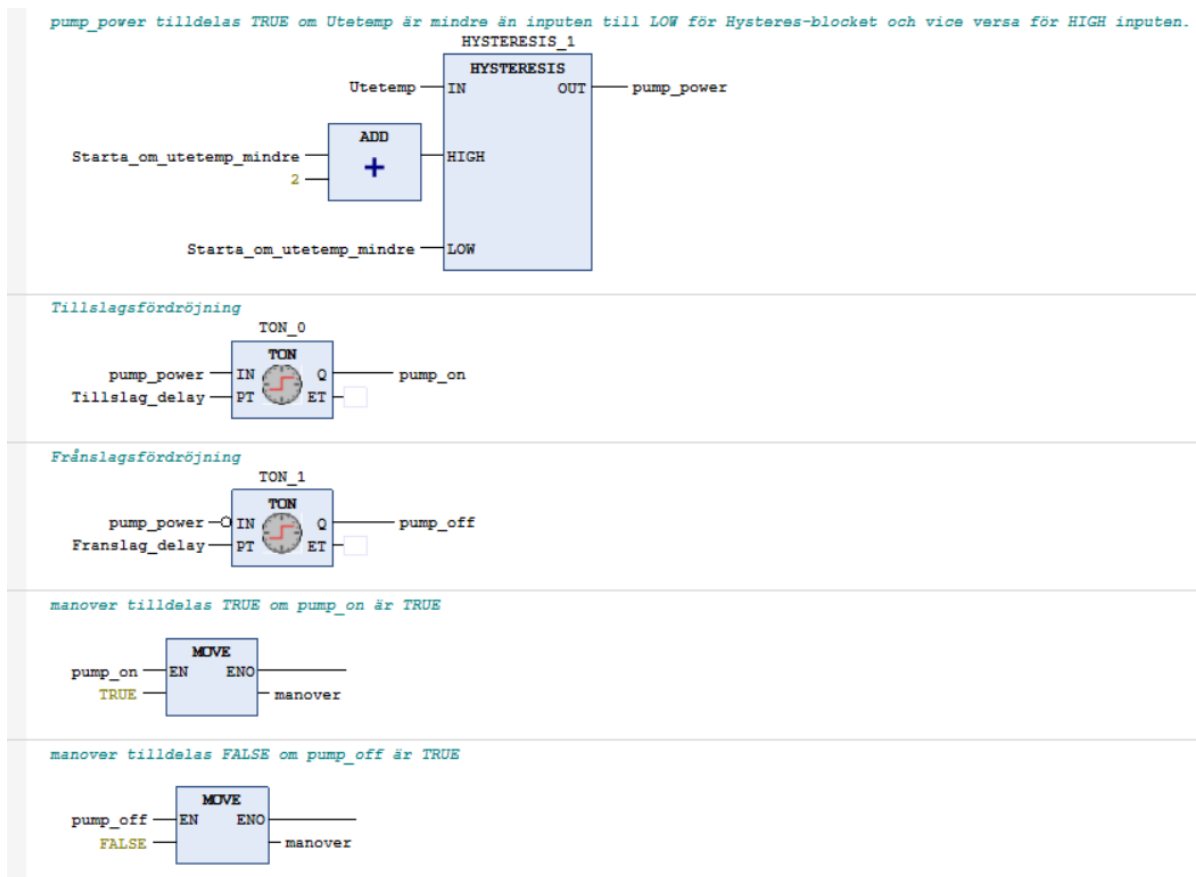
Ett problem som uppstod med UltraBase20 var att vid skrivning via Modbus till externa enheter så finns det ingen möjlighet för UltraBase20 att kontrollera om adressen som skrivs till redan innehåller exakt samma data som skickas. Detta leder till att data skickas och minnet i den externa enheten skrivs över i onödan vid varje skrivcykel. Då detta problem är en konsekvens av hur UltraBase20 är designat och byggt finns det inga lösningar som inte involverar en komplicerad ändring av skriptet som styr själva UltraBase20:n. Som ett alternativ till skriptändring i UltraBase20:n har logiken som styr cirkulationspumpen programmerats in i ett alternativt styrsystemsprogram Codesys. Syftet med detta var att visa hur en alternativ lösning skulle kunna implementeras i en Codesys-kompatibel PLC som inte har samma begränsningar som en UltraBase20.

Endast den delen av styrlogiken som används för styrning av cirkulationspumpen MAGNA3 har realiserats i Codesys med språket "Function Block Diagram" (FBD). Alla signaler som i verkligheten kommer från olika externa enheter kopplade till UltraBase20:s olika fysiska ingångar har simulerats med globala variabler i Codesys.



Figur 23: Skärmdump som visar top-nivån på PLC programmet i Codesys.

I figur 23 ovan finns två block; H\_P\_S\_1 som innehåller start/stopp logiken och SEL som är ett *selector*-block som används för att simulera driftindikeringens signalen som läses från pumpen via Modbus.



Figur 24: Logiken inuti H\_P\_S\_1; baserat på värdet hos "Utetemp" så startar eller stoppar pumpen efter en inställbar fördröjning.

I figur 24 ovan syns logiken inuti blocket H\_P\_S\_1, funktionaliteten är densamma som för applikationen med samma namn i UltraBase20 skapad av Ekelunds Styr, dock har vissa delar av logiken anpassats för att fungera i Codesys. Funktionaliteten beskrivs nedan:

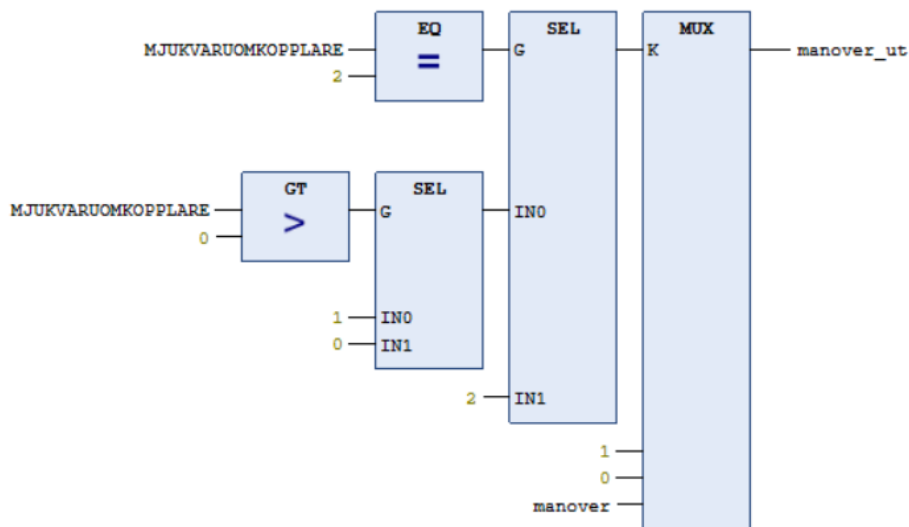
När variabeln *Utetemp* är mindre än det som skrivs till *LOW* ingången hos hysteres-blocket startas en timer. Om variabeln *Utetemp* aldrig överskrider värdet som skrivs till *LOW* ingången under denna tidsfördröjning startas pumpen.

När variabeln *Utetemp* är större än det som skrivs till *HIGH* ingången hos hysteres-blocket startas en timer. Om variabeln *Utetemp* aldrig underskrider värdet som skrivs till *LOW* ingången under denna tidsfördröjning stoppas pumpen.

Hur lång tidsfördröjningen är bestäms av variablerna *Tillslag\_delay* vid start och *Franslag\_delay* vid stopp.

Variablerna *Tillslag\_delay*, *Franslag\_delay* och *Starta\_om\_utetemp\_mindre* är i verkligheten parametrar som tas in från Web Port och *Utetemp* är en temperaturgivare inkopplad till en fysisk ingång på UltraBase20. I Codesys har dessa simulerats som globala variabler och tillskrivits värden för hand i syfte att kontrollera korrekt funktionalitet.

Om MJUKVARUOMKOPPLARE sätts till 1 eller 0 blir utsignalen från H\_P\_S\_1 blocket densamma, reglering baserat på Utetemp följs bara om MUKVARUOMKOPPLARE == 2



Figur 25: Logik för den mjukvaruomkopplaren skapad av Ekelunds Styr som finns i UltraBase20 skriven med FBD kod i Codesys, även denna finns inuti H\_P\_S\_1-blocket.

Figur 25 ovan visar hur mjukvaruomkopplaren för handkörning av pumpen via Web Port byggts upp i Codesys med selector och multiplexer block inuti H\_P\_S\_1-blocket. Logiken är densamma som den Ekelunds Styr byggt upp i UltraBase20. Baserat på värdet för MJUKVARUOMKOPPLARE sätts pumpen i tre olika lägen:

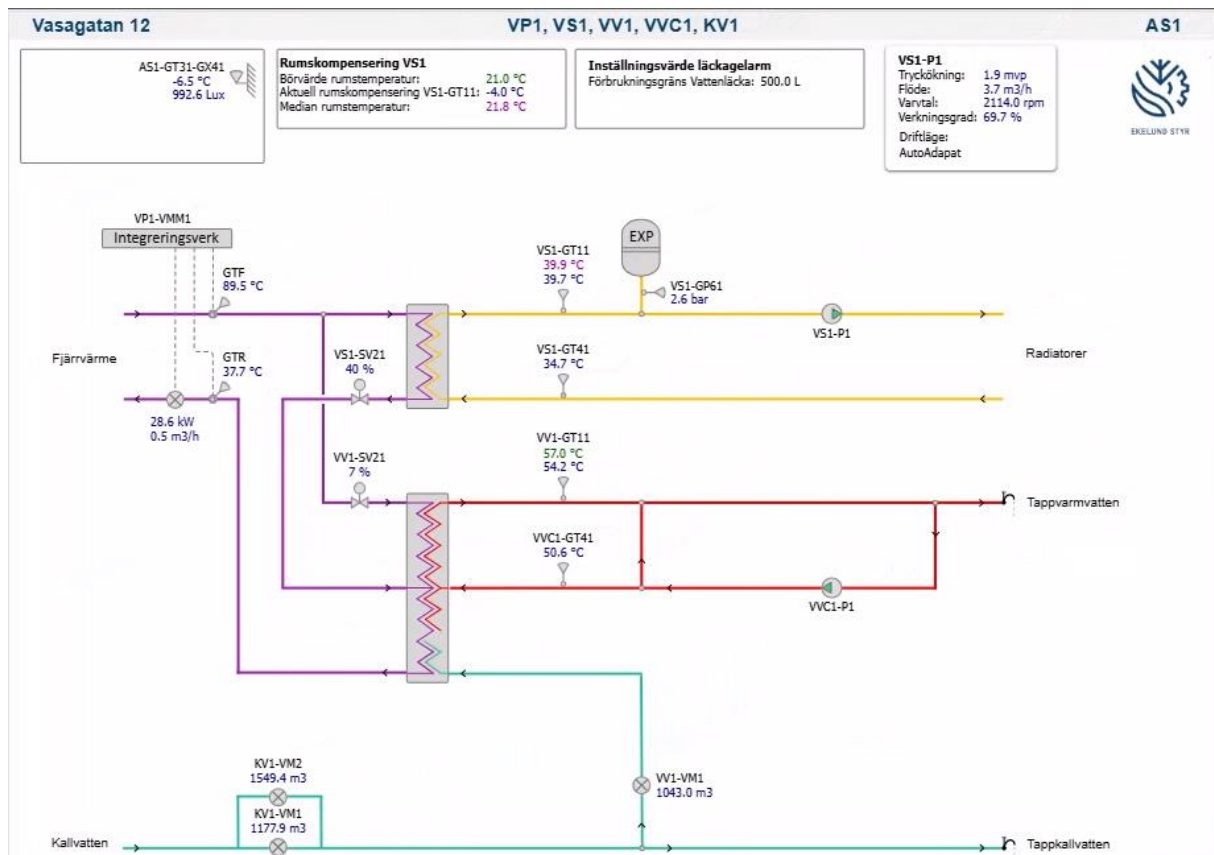
**MJUKVARUOMKOPPLARE = 2;** "Auto", startar eller stoppar pumpen baserat på logiken som syns i figur 21.

**MJUKVARUOMKOPPLARE = 1;** "Till" pumpen startas oavsett övrig logik.

**MJUKVARUOMKOPPLARE = 0;** "Från" pumpen stoppas oavsett övrig logik.

## 8 Resultat

Övervakning i Web Port av alla parametrar från cirkulationspumpen MAGNA3 är tillagda i det övre högra hörnet på översiktsvyn för värmesystemet (Se figur 26).



Figur 26: Den fullständiga översiktsvyn av värmesystemet med de nya övervakningsparameterna tillagda i övre högra hörnet.

Ingen fullständig styrning av MAGNA3 via Web Port implementerades och implementeringen som beskrevs i kapitel 6 togs bort på grund av följande konflikter och problem som identifierades tillsammans med SGS och Ekelunds Styr:

All logik som redan var gjord av Ekelunds Styr i UltraBase20 för cirkulationspumpens start och stopp var byggd kring kommunikation via digitala in- och utgångar mellan pump och styrsystem, inte Modbus. Om data skrevs till någon av pumpens adressregister via Modbus överskred denna kommunikation alla signaler hos pumpens digitala in- och utgångarna. Detta innebär att styrlogiken för pumpens start och stopp skulle behöva ändras så att alla signaler mellan styrsystemet och pumpen skedde via Modbus. Problemet med att ändra stor del av styrlogiken i UltraBase20 är att all programmering skulle vara tvungen att göras i den aktiva UltraBase20-enheten på Vasagatan 12 då ingen extra UltraBase20-enhet eller liknande fanns att tillhandahålla som testplattform från SGS för detta projekt. Risken för potentiella fel vid omprogrammering av start och stopp logiken ansågs för stora vilket innebär att ingen teknisk implementering gjordes utöver ett demonstrativt exempel som

användes för att informera den teoretiska beskrivningen av processen i rapporten till SGS.

Ett ytterligare problem är mer fundamentalt till styrsystemet UltraBase20 gällande skrivning till externa enheter. När man väljer uppdateringstid för en signal hos en extern enhet innebär detta att styrsystemet kommer att skicka en signal via Modbus till den externa enheten *oavsett* om registret den skriver till redan innehåller exakt samma data som skickas. Detta leder till att en kompromiss måste göras gällande styrsystemets responsivitet och livslängd på flashminnet hos den externa enheten. För fullständig implementering av styrning för pumpen via Modbus skulle även fler adressregister än bara *ControlMode* behöva skrivas till vilket skulle belasta CIM-200 enheten och MAGNA3 ännu mer.

För att visa en möjlig lösning till problemet med onödig skrivning via Modbus implementerades den start/stop logik som används för cirkulationspumpen i ett separat styrsystemsprogram, Codesys, som är kompatibelt med en rad olika PLC:s som inte har dessa inbyggda begränsningar. Endast logiken för cirkulationspumpen implementerades i Codesys med språket Function Block Diagram (FBD), verifiering av korrekt funktionalitet gjordes via mjukvarusimulering i Codesys-programmet. Denna lösning implementerades ej i någon av SGS byggnader utan skapades endast för att demonstrera hur samma logik skulle kunna programmeras i Codesys för en PLC som inte har samma begränsningar som UltraBase20.

En skriftlig rapport för SGS innehållande dokumentation som redogör för den tekniska implementeringen gjord i detta projekt, hur styrsystemet vid Vasagatan 12 fungerar samt en organisatorisk analys av SGS och slutsats angående arbete med fastighetsautomation inom organisationen skrevs och redovisades för personal på SGS. SGS-rapportens slutsats angående arbete med fastighetsautomation inom organisationen baserades på diskussioner med handledare på SGS, personal från Ekelunds Styr och styrtekniker på SGS.

Den interna SGS rapportens slutsats är att nuvarande personal som jobbar med fastigheterna på SGS saknar kunskap och kompetens gällande programmering av styrsystem. Även om rätt kompetens fanns hos styrteknikerna så blir deras arbetsbörda för stor av ytterligare ansvarsområden. SGS har också byggnader med andra styrsystem än UltraBase20, vilket skulle innebära att behovet av externa entreprenader inte upphör även om kompetens fanns för just UltraBase20. Anställning av ny personal eller vidareutbildning av existerande personal anses inte heller ekonomiskt eller organisatoriskt möjligt i dagsläget. SGS rapporten har dock gett SGS bättre förståelse kring styrsystemens funktion och deras begränsningar. Rapporten kan även komma att informera framtida beslut och projekt gällande fastighetsautomation inom SGS.

## 9 Slutsats och diskussion

Projektets syfte att öka SGS kunskap och förståelse kring styrsystem och fastighetsautomation anses uppfyllt trots att försöket med fullständig implementering gällande skrivning till pumpen inte uppnåddes. Projektets största framsteg har varit att påvisa för SGS möjligheten av att läsa en extern enhets minne med Modbus för att i realtid se dess tillstånd i ett externt program. Åtkomst till mer data från externa enheter skulle potentiellt kunna leda till bättre övervakning inom SGS av Modbus-kompatibla enheter i framtiden. Potentiella fel skulle då kunna upptäckas och åtgärdas snabbare innan de hinner leda till större skada av byggnaden. Om fel skulle kunna diagnosticeras på distans i Web Port skulle detta också leda till att driftteknikernas arbetsbörda och stress skulle minska då dem inte behöver undersöka lika många problem på plats.

Kommunikationskonflikten mellan Modbus och de digitala in- och utgångarna skulle förmodligen kunnat lösas om en separat UltraBase20-enhet funnits tillgänglig för detta projekt. I så fall hade start/stopp-logiken för pumpen kunnat omprogrammeras till att använda Modbus utan risk för att potentiella fel skulle påverkat byggnaden. Då arbetet utfördes i januari och februari 2026 skulle den potentiella skadan av ett fel med pumpens start/stopp logik vara extra stor då pumpen krävs för att hålla temperaturen uppe i byggnadens bostäder.

Kompromissen mellan livslängd och responsivitet för pumpen skulle dock finnas kvar och förmodligen leda till att SGS väljer att inte använda Modbus för mer än till övervakning av en del av pumpens prestandaparametrar. Grundfos specificerar inte typen eller storleken på minnet som används i MAGNA3 eller CIM-200. Således blir påverkan av den onödiga skrivningen på enheternas livslängd svår att fastställa. Dock skulle risken av en minskad livslängd på pumpen eller CIM-200 enheten varken vara ekonomiskt eller ekologiskt acceptabelt för SGS som både har en budget och riktlinjer gällande miljömål att uppfylla. Att SGS inte går vidare med styrning via Modbus görs ännu mer troligt då den funktionalitet som möjliggörs går att åstadkomma med befintlig personal som kan konfigurera pumpen på plats via dess inbyggda terminal.

Den alternativa lösningen som demonstrerades med Codesys ansågs ej aktuell som ersättning för redan installerade styrsystem för SGS då detta skulle innebära att helt byta styrsystem och använda ny mjukvara för relativt små fördelar. Dock fanns intresse inom SGS att minska antalet olika sorters styrsystem som används i deras byggnader. Detta kommer troligen åstadkommas genom att SGS identifierar det styrsystem de föredrar och sedan ställer användningen av det styrsystemet som ett krav för alla entreprenader som anlitas för framtida byggen eller renoveringar.

## Referenser

- [1] Energimyndigheten, "Fjärrvärme," 2022. [Online]. Tillgänglig: <https://www.energimyndigheten.se/effektiv-energianvandning/guider/husguiden-for-dig-som-vill-energieffektivisera-ditt-hus/se-over-husets-uppvarmningssystem/fjarrvarme/>. [Använd 12 Mars 2026].
- [2] E. Knapp, *Industrial Network Security*, USA: Syngress, 2011.
- [3] W. Buchanan, *Computer Buses*, USA: CRC, 2000.
- [4] *Modbus Application Protocol Specification V1.1b3*, Boxborough, USA: Modbus Organization, 2012. [Online]. Tillgänglig: <https://www.modbus.org/file/secure/modbusprotocollspecification.pdf>. [Använd 16 Februari 2026].
- [5] Abelko Innovation, "Fastighetsautomation," u.d. [Online]. Tillgänglig: <https://abelko.se/fastighetsautomation/>. [Använd 12 Mars 2026].
- [6] *IMSE Ultra Konfigureringsmanual*, Luleå, Sverige: Abelko Innovation AB, u.d. [Online]. Tillgänglig: [https://media.abelko.se/Manualer/Ultrafamiljen/UBxx/config\\_manual\\_sv.pdf](https://media.abelko.se/Manualer/Ultrafamiljen/UBxx/config_manual_sv.pdf). [Använd 16 Februari 2026].
- [7] *IMSE Ultra Script Language Reference manual*, Luleå, Sverige: Abelko Innovation, u.d. [Online]. Tillgänglig: [https://media.abelko.se/Manualer/Ultrafamiljen/UBxx/reference\\_en.pdf](https://media.abelko.se/Manualer/Ultrafamiljen/UBxx/reference_en.pdf). [Använd 16 Februari 2026].
- [8] Kiona, "Kiona Kundsupport," 2026 [Online]. Tillgänglig: <https://iwmac.zendesk.com/hc/sv/articles/7127736793116-Taggstandard>. [Använd 16 Februari 2026].
- [9] *MAGNA3 Cirkulationspumpar 50 Hz*, Mölndal, Sverige: Grundfos, u.d. [Online]. Tillgänglig: <https://api.grundfos.com/literature/Grundfosliterature-4609606.pdf>, [Använd 20 Januari 2026].
- [10] *Modbus for Grundfos pumps*, Mölndal, Sverige: Grundfos, 2025. [Online]. Tillgänglig: <https://api.grundfos.com/literature/Grundfosliterature-6012947.pdf>, [Använd 18 December 2025].



**CHALMERS**