

# Elnätskommunikation med gasdetektor



**LUNDS  
UNIVERSITET**

Lunds Tekniska Högskola

**LTH Ingenjörshögskolan vid Campus Helsingborg  
Datateknik**

Examensarbete:  
Christer Nilsson  
Daniel Tretinjak



© Copyright Christer Nilsson, Daniel Tretinjak

LTH Ingenjörshögskolan vid Campus Helsingborg  
Lunds universitet  
Box 882  
251 08 Helsingborg

LTH School of Engineering  
Lund University  
Box 882  
SE-251 08 Helsingborg  
Sweden

Tryckt i Sverige  
Media-Tryck  
Biblioteksdirektionen  
Lunds universitet  
Lund 2010

## Sammanfattning

I denna rapport kommer vi att redogöra för hur vi gick till väga för att skapa en första prototyp till en plattform för kommunikation mellan detektorer och en central. Detektorerna i detta fall är gasdetektorer för kylmedium.

Kommunikationen mellan dessa gasdetektorer och centralen kan ske på många olika sätt, men begränsas helt av miljön de installeras i. I en vanlig byggnad kan det ske på flertal olika vis. Ethernet, WLAN och Bluetooth är tekniker som finns tillgängliga. När det kommer till fartygsmiljö är all radiobunden kommunikation i princip omöjlig p.g.a. stålväggar. Dessa väggar bildar en Faradays bur, vilket försvårar radiokommunikation. Samtidigt är det högst olämpligt att borra nya hål för sladdar till alla detektorerna, då det finns krav på vattentätighet mellan skotten. För att möjliggöra kommunikation på ett praktiskt och bra sätt, valde vi att låta kommunikationen ske med hjälp av Power Line Communication (PLC).

Det finns leverantörer som har färdiga modem för PLC och vi har haft med en del av dessa i vår undersökning av lämpliga lösningar. För att komma PLC-tekniken riktigt nära och upptäcka dess egenheter och att samtidigt komma ett steg närmare den slutliga lösningen, valde vi att bygga ett eget modem för kommunikation. Dessutom hade ingen av de färdiga modemerna som vi tittat på någon elektronik för matningsspänning. Med hjälp av applikationsexempel från modemkretsleverantören kunde vi utveckla ett prototypkort med de funktioner som vi ansåg skulle behövas.

Stegvis utvecklades systemet tills det fungerade som förväntat. Meddelande skickades från en PC via ett länkkort till ett detektorkort som svarar med avlästa värden från en gasgivare.

Nyckelord: Powerline communication, gasdetektor

## **Abstract**

Communications where the location and environment limits the options is an issue for Samon AB. Some of their gasdetectors work in a marine environment where they are surrounded by steel walls. These steel walls act like a Faraday cage and make radio communication ineffective. Other gasdetectors are retrofitted in hotels when the demand for air-conditioning increases. With retrofitting comes the issue with new cables. Not only a cable for the communications but also for the powersupply.

Our solution involves Powerline Communications (PLC) and resolves the issues in both cases. By creating a platform for communication and powersupply in one, and by using existing wiring, no new cables are needed. This makes installation easy and reduces the cost of new wiring.

Keywords: Powerline communication, gasdetectors

## Förord

Denna rapport är resultatet av ett examensarbete som Christer Nilsson och Daniel Tretnjak utförde på vårterminen 2010. Examensarbetet är det sista momentet i utbildningen till högskoleingenjör i datateknik vid Lunds Tekniska Högskola Campus Helsingborg. Arbetet utfördes i samarbete med Samon AB.

Vi skulle vilja tacka följande personer för all hjälp och råd vi fått på vägen:

Christian Nyberg, Examinator LTH

Jonas Sjöström, Dzenkis Junuzovic, SAMON AB

Ulf Johansson, Elsäkerhetsverket

Svante Skeppstedt, Svensk Elstandard

Thomas Pehrsson, Robert Tretnjak, E.ON

Mats Anderzén, Transportstyrelsen Sjöfartsavdelningen

Pär Löfgren, Bo Liljedahl, Mikrodidakt AB

Kurskamraterna på LTH

# Innehållsförteckning

<b>1 Nulägesbeskrivning .....</b>	<b>2</b>
<b>2 Arbetets avgränsningar .....</b>	<b>2</b>
<b>3 Översikt av PLC .....</b>	<b>3</b>
<b>3.1 Modulationstekniker .....</b>	<b>3</b>
<b>3.2 Frekvenser .....</b>	<b>3</b>
<b>3.3 Protokoll.....</b>	<b>4</b>
3.3.1 CENELEC .....	4
3.3.2 X10 .....	4
3.3.3 KNX bus.....	4
3.3.4 LonWorks.....	4
3.3.5 BACnet .....	5
<b>3.4 Användningsområde .....</b>	<b>5</b>
<b>3.5 Standarder .....</b>	<b>5</b>
<b>4 Systemlösning .....</b>	<b>6</b>
<b>4.1 Hårdvara.....</b>	<b>7</b>
4.1.1 Modemkrets .....	7
4.1.2 MCU.....	7
4.1.3 Detektorkort och länkkort .....	8
4.1.3.1 Mönsterkort.....	8
4.1.3.2 Kretskort .....	8
4.1.3.3 Bandpassfilter.....	9
<b>4.2 Mjukvara.....</b>	<b>9</b>
4.2.1 Masterprogrammet.....	9
4.2.1.1 Utvecklingsmiljö.....	10
4.2.1.2 Programmets uppbyggnad.....	10
4.2.2 Länkkortet .....	11
4.2.2.1 Utvecklingsmiljö.....	11
4.2.2.2 Periferienheter i MCU .....	11
4.2.2.3 Programmets uppbyggnad.....	12
4.2.3 Detektorkortet .....	13
4.2.3.1 Utvecklingsmiljö.....	13
4.2.3.2 Periferienheter i MCU .....	13
4.2.3.3 Programmets uppbyggnad.....	14
<b>4.3 Kommunikationsprotokoll .....</b>	<b>16</b>
4.3.1 Modbus .....	16
<b>5 Test .....</b>	<b>19</b>
<b>5.1 Sal C428, liten datorsal .....</b>	<b>19</b>
<b>5.2 Sal C451, stor datorsal .....</b>	<b>19</b>
<b>5.3 Sal C421, laborationssal .....</b>	<b>19</b>

5.4 Sal C422, föreläsningssal.....	19
5.5 1-fas testet.....	20
5.6 3-fas testet.....	21
6 Vidareutveckling.....	22
7 Slutsats .....	23
8 Referenser .....	24
9 Bilagor.....	25
9.1 Förkortningar .....	25
9.2 Kopplingsschema .....	26
9.3 Komponentplacering .....	29
9.4 Komponentgruppering .....	30
9.5 Plan 4 i C-huset.....	31



## **Inledning**

Vårt syfte med detta examensarbete är att undersöka möjligheterna till kommunikation med Samons gasdetektorer via Powerline Communication (PLC), på ett sätt som minimerar ny kabeldragning och klarar varierande miljöer.

Målsättningen är att kunna uppvisa en fungerande prototyp av ett system för Powerline Communication, där det ingår ett masterprogram, ett länkkort och ett eller flera slavkort. Eftersom specifika krav på kommunikationens kvalitet inte har satts har vi jobbat efter worst-case scenario vilket innebär mycket längre meddelande, fler meddelande i följd till slavkort än troligt behov samt en meddelandefrekvens ut från länkkortet, som är högt över kundens förväntan.

I rapporten avses förkortningen PLC som Powerline Communication, om inget annat anges.

## 1 Nulägesbeskrivning

Med en allt större fokus på växthusgaser och deras negativa påverkan på vår miljö, installeras gasdetektorer på platser där det kan ske läckage. I denna rapport ska vi fokusera på en lösning för kommunikation med gasdetektorer i miljöer där kabeldragning är kostsam eller olämplig. Många kylsystem för luftkonditioneringar innehåller växthusgasklassat kylmedium. Ett läckage av kylmedium innebär sämre prestanda på kylsystemet, miljöförstöring i form av växthusgas och en möjlig risk för personer som utsätts för gasen. Med hjälp av vår lösning ska ett läckage upptäckas genom att ett uppmätt värde från en gasdetektor överskrider inställda gränsvärden.

Samon AB är ett Vellingebaserat företag, som grundades 1990. Affärsidén är att utveckla, tillverka och marknadsföra gasvarnare för bland annat kyl & frysanläggningar. Bland deras kunder hittar vi Carlsberg, Dafgård, US Navy, Svenska Marinen, Silja Line och Astra Zeneca.

Idag arbetar Samons gasdetektorer helt fristående. De övervakar sin omgivning och har en reläutgång som styr exempelvis en ventil. Ventilen kan, vid ett larm, stänga av tillförseln av det ämne som detektorn övervakar och ge ett larm. I nuläget används 3 larmnivåer på detektorerna vars gränser kalibreras för hand, genom justering av vridpotentiometrar. Dessa larmnivåer aktiverar reläer som i sin tur styr andra enheter som ventiler och larmsignaler. Servicetekniker får med bestämda mellanrum kontrollera och kalibrera larmnivåerna, vilket kan innebära mycket jobb om det finns många detektorer i en fastighet.

Vid våra möten med kontaktpersonerna på Samon AB fick vi informationen att de inte kände till något system för PLC med gasdetektorer. Vid sökning efter möjliga lösningar på Internet kunde vi bekräfta att utbudet var praktiskt taget obefintligt. Båda parter var intresserade av ett undersökande arbete inom området och därmed avgränsades arbetet till enbart PLC. Att undersöka och jämföra andra kommunikationssätt hade varit alltför tidskrävande.

## 2 Arbetets avgränsningar

Vi avgränsar oss från att inte gå in djupare i tolkningar av klassningssällskapens regler och föreskrifter gällande PLC-utrustning på fartyg. Detta anser vi vara för tidskrävande för vårt examensarbete.

Vi följer inte Modbus standarden till fullo, utan väljer att begränsa oss till ett Modbus liknande protokoll. Funktionskoderna, formen på meddelanden, svar

och CRC är enligt standard men vi avviker i att vi inte använder oss av paritetskontroll. Detta hann vi inte med att implementera.

### **3 Översikt av PLC**

Powerline communication kan förekomma på olika spänningsnivåer i elnätet. Normalt anses en transformator fungera som en barriär mellan två spänningsnivåer där kommunikationen inte kan ta sig igenom. Om man behöver kommunicera mellan två spänningsnivåer får man koppla in en brygga. [6]

#### **3.1 Modulationstekniker**

Den PLC som denna rapport tar upp bygger på att man överlagrar en modulerad bärvåg på elnätets kablar, mellan fas och nolla. Den överlagrade signalen kan variera i bandbredd beroende på hur mycket information som skall överföras. Man delar upp PLC i två grupper, narrowband för de applikationer som kräver att en mindre mängd data överförs och broadband för de applikationer som kräver att en större mängd överförs [7]. Denna rapport handlar om det som kallas för narrowband PLC.

De två vanligaste grundprinciperna för PLC är FSK (Frequency Shift Keying) och PSK (Phase Shift Keying). Dessa grundprinciper existerar i vidareutvecklade varianter och heter då t.ex. BPSK (Binary Phase Shift Keying), S-FSK (Spread Frequency Shift Keying), B-FSK (Binary Frequency Shift Keying), QPSK (Quadrature Phase Shift Keying), 8PSK och narrowband OFDM (Orthogonal Frequency Division Multiplexing)[11,12,13]

#### **3.2 Frekvenser**

Frekvensområdet för narrowband-applikationer ligger mellan 3 och 148,5kHz. Detta frekvensområde är i sin tur uppdelat på fyra band, A till D, enligt den europeiska standarden CENELEC 50065-1 [2].

Band A är frekvenser mellan 3-95kHz och är avsett för eldistributörerna och deras licensinnehavare. Band B är frekvenser mellan 95-125kHz.

Band C är frekvenser mellan 125-140kHz.

Band D är frekvenser mellan 140-148kHz.

Band C är det enda band där standarden kräver ett Access-protokoll.

Frekvensområdet för broadband-applikationer ligger mellan 1 och 30 MHz.

### 3.3 Protokoll

Till att börja med kan man dela upp protokollen i de som reglerar de fysiska lagren och de som reglerar lagren med mjukvaran. De ovan nämnda moduleringssteknikerna ligger i det fysiska lagret och regleras alla av sitt eget protokoll. Protokollen för mjukvaran finns i många former och riktar sig till olika användningsområden. En del riktar sig mot det egna hemmet för styrning av t.ex. belysningen. Andra riktar sig mot den kommersiella marknaden för att kontrollera klimatanläggningar i kontorsbyggnader. Denna rapport behandlar bara de protokoll som riktar sig till narrowband-applikationer. Nedan följer en beskrivning av några av de vanligaste protokollen. Modbus protokollet behandlar vi i avsnitt 4.3.1.

#### 3.3.1 CENELEC

CENELEC är egentligen en standard men innehåller även ett accessprotokoll som reglerar hur enheterna hanterar tillgången till mediet. Det vill säga det ser till att enheterna inte ”pratar i mun” på varandra. Detta protokoll måste följas om enheten kommunicerar på C-bandet. [2]

#### 3.3.2 X10

X10 har funnits sen 70-talet och är framtaget för att styra hemmets diverse elektriska utrustningar. Det är ett enklare protokoll som bl.a. innehåller kommando för att tända och släcka lampor. Det finns en uppsjö av tillverkare och produkter på marknaden som använder detta protokoll. [8]

#### 3.3.3 KNX bus

KNXbus är utvecklat i Europa och är ett protokoll som har sitt ursprung i EIB (European Installation Bus), EHS (European Home Systems) och BatiBus och har därigenom funnits i över 15 år. Detta protokoll gör det möjligt att styra all data mellan alla möjliga komponenter i en fastighet. Protokollet är tänkt att användas tillsammans med flera medier nämligen twisted pair, radiovågor, elnätet och IP/Ethernet. Bakom KNXbus står KNX Association [9] som är en sammanslutning mellan företag inom fastighetsautomation. Tanken är att tillhandahålla en standard som gör att alla som följer detta protokoll kan ta fram enheter som kan kombineras med varandra. Det finns installationer i Europa, Asien, Nord- och Sydamerika. KNX är erkänd som standard genom ISO/IEC 14543-3, CENELEC EN 50090 och CEN EN 13321-1 samt GB/Z 20965.

#### 3.3.4 LonWorks

LonWorks har utvecklats i USA av Echelon [11] sen 1980. Tillsammans med LonMark bildar det en systemlösning och är precis som KNX anpassat för att

styra all data mellan diverse enheter i en fastighet. KNX och LonMark är konkurrenter och förekommer på samma marknader. Givetvis kan enheter med olika tillverkare kombineras med varandra och enheterna kan kommunicera på olika medier såsom twisted pair, elnätet och IP/Ethernet. LonMark har mer än 700 medlemsorganisationer och LonWorks har blivit erkänd som en global standard genom ISO/IEC 14908.

### 3.3.5 BACnet

BACnet [10] påbörjades 1987 och är också utvecklat i USA. BACnet står för Building Automation and Control Networks och är en öppet protokoll vilket betyder att vem som helst kan bidra. Vem som helst får också använda protokollet, den enda "haken" är att ASHRAE, som var de som tog fram protokollet, har Copyright på standarden och säljer själva dokumentet. För intäkterna håller ASHRAE i utvecklingen och underhåller dokumentet. Protokollet utnyttjar fyra av de sju lagren i OSI-modellen, nämligen det fysiska lagret, datalänklagret, nätverkslagret och applikationslagret. Protokollet specificerar dock bara Nätverkslagret och Applikationslagret och kan därför kombineras med önskat Fysiska lager och/eller Datalänklager som kan vara Ethernet, EIA-232, EIA-485, ARCNET och LonTalk. BACnet har blivit erkänd som en standard globalt.

## 3.4 Användningsområde

Idag kan man använda PLC till en rad olika applikationer både i det egna hemmet och i kommersiella byggnader. Ett vanligt område är applikationer inom AMR (Automatic Meter Reading) och är det som Eldistributörerna använder för att avläsa kundernas elmätare. Ett annat stort område som har funnits länge är applikationer för att bland annat styra hemmets belysning (exempelvis med X10).

I kommersiella byggnader används PLC för att styra och känna av temperatur, ventilation, ljus, larm, hissar, brandlarm och solskydd. Dessa exempel klassas som narrowband applikationer. Som exempel på broadband-applikationer kan nämnas tillgång till Internet, telefon och TV. [7, 9, 11, 12]

## 3.5 Standarder

Det finns många standarder inom PLC och de reglerar kommunikationen på olika nivåer i de olika lagren. Det finns standarder som gäller globalt och de som gäller t.ex. i en världsdel. För att kommunicera med hjälp av PLC i Europa måste man följa CENELEC 50065-1 och för att göra samma sak i USA måste man följa FCC reglemente. Dessa standarder reglerar hur man får

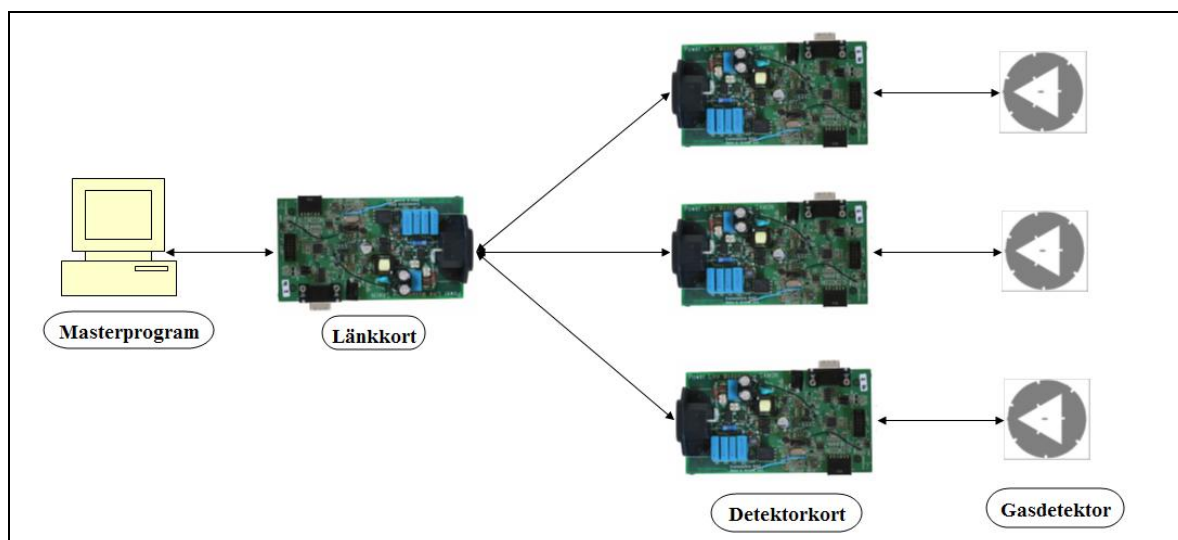
applicera en signal ovanpå matningsspänningen, vilka frekvenser, nivåer och så vidare.

De olika modulerings teknikerna följer sina respektive protokoll och dessa har i sin tur blivit erkända som standarder. Exempelvis följer S-FSK en standard som heter IEC 61334-5-1. Ovanpå det fysiska lagret, exempelvis S-FSK, ligger ytterligare protokoll i olika lager beroende på vad man valt för system. Det kan t.ex. vara KNX som i sin tur har blivit en internationell standard som heter ISO/IEC 14543-3.

## 4 Systemlösning

Efter att vår tidplan godkännts av examinator och handledarna på Samon AB, bestämde vi oss för att jobba intensivt med faktainsamlingen. Första delen av faktainsamlingen var att undersöka vilka regler och föreskrifter vi skulle behöva känna till. Det visade sig att det finns en del standarder och forskningsrapporter [1,2] som behandlar PLC och hur mycket påverkan det får ha på elnätet men även vad som kan påverka kommunikationen. En annan del av faktainsamlingen var val av komponenter. Detta är en viktig del av projektet och skulle kunna ha avsevärd inverkan på projektets resultat. De stora valen var PLC modemkrets och MCU. Här fanns en del olika alternativ till hårdvarulösningar och hade tiden funnits, så hade vi kunnat göra en noggrannare utvärdering av de olika produkterna. Under avsnitt 4.1 går vi in lite närmare på hur vi gjorde våra val av komponenter.

Vår systemlösning innehåller ett länkkort och minst ett detektorkort (se figur 1). Länkkortets uppgift är att vidarebefordra meddelanden till och från masterprogrammet och de olika detektorkorten. Alla våra kort är bestyckade och uppbyggda på samma sätt, så det är endast programkoden som skiljer sig mellan de olika typerna. Skillnaden mellan korten är att länkkortet bara tar emot meddelande och skickar dem vidare utan att avkoda dem på något sätt. Detektorkorten däremot tar emot meddelandena och avkodar dem samt returnerar ett svar. Dessutom konverterar detektorkorten de analoga signalerna från det anslutna givarkortet till digitala värden.



Figur 1: Hela kedjan i systemlösningen

## 4.1 Hårdvara

Att välja hårdvara var en fas i arbetet som kom tidigt. Efter att så mycket fakta som möjligt samlats in kunde beslut tas.

### 4.1.1 Modemkrets

När det gäller modemkretsen hade vi hittat tre olika tillverkare: STMicroelectronics (ST), Echelon och Archnet. Archnets produkter valdes bort då dessa kom i färdiga moduler och gav oss begränsade möjligheter till att utveckla ett mer, efter våra behov, anpassat kort. Valet stod och vägde mellan Echelons och ST:s kretsar, men eftersom i Echelons lösning ingår deras egna inbyggda MCU, begränsade detta vårt val av MCU. Därför bestämde vi oss för ST7540 modemkrets från ST.

### 4.1.2 MCU

När det kom till val av processor så hade vi två alternativ. Första alternativet var en ATmega128 från Atmel och andra alternativet var en C8051F350 från Silicon Labs. Den avgörande skillnaden mellan dessa två MCU var deras ADC. ATmegan hade en 10-bitars SAR medan C8051 hade en 24-bitars Delta Sigma. Skillnaden i upplösning ansåg vara vi var en nödvändighet då vi skulle mäta styrkan på den mottagna signalen. Denna signals styrka kunde variera från cirka  $250\mu\text{V}$  till flera Volt, beroende på dämpningen i ledningarna. På grund av detta breda spänningsintervall var en 24-bitars ADC mer lämplig. Att kunna mäta signalstyrkan ansågs endast som ett hjälpmedel och skulle inte

vara en nödvändighet att implementera på slutprodukten. På slutprodukten kan man använda en billigare MCU beroende på kundernas behov.

### 4.1.3 Detektorkort och länkkort

Här följer en redogörelse för kortens bestyckning. Vi tog höjd för en del funktioner vi inte hann med att implementera, men vi ansåg att det är bättre att lägga till lite för mycket än att behöva komplettera.

#### 4.1.3.1 Mönsterkort

Ett kopplingsschema (se avsnitt 9.2) för ett mönsterkort ritades upp med STMicroelectronics applikationsexempel [5] från deras utvärderingskort för ST7540 till grund. Utöver deras lösning har vi lagt till följande funktioner:

- Möjlighet att koppla in oss på 3-fas uttag
- nollgenomgångsdetektor
- En krets för mätning av signalstyrkan på ingående signal
- Förberett med kontakt för en I2C teckendisplay.

Anledningen till att vi förberedde tillgång till alla tre faserna var för att vi visste att fasöverhörningen var dålig. Fasöverhörningen kan göra att signalen tappar uppemot 25dB [14]. Med möjligheten att koppla in oss på alla 3 faserna behöver vi inte förlita oss på att signalen ska sprida sig mellan faserna via 3-fas-utrustningar eller att de olika fasernas kablar ligger dragna intill varandra.

Nollgenomgångsdetektorn är en krets som känner av när 230V spänningens fas närmar sig nollgenomgången. I detta område är det som tystast och man kan kommunicera med mindre störningar. Denna typ av kommunikation ger betydligt lägre hastighet och används av t.ex X10 protokollet.

I våra prototyper skulle det finnas möjlighet att mäta signalstyrkan på inkommande signal och på detta sätt utvärdera möjligheterna till bra kommunikation i olika eluttag. I detta fall skulle vår signalstyrkemätningsskrets komma till användning och förse oss med information.

Att få ut signalstyrka och annan information direkt vid detektor/länkkort och inte bara i PC, hade underlättat vid testning. Detta skulle vår display lösa där all nödvändig information skulle kunna visas.

#### 4.1.3.2 Kretskort

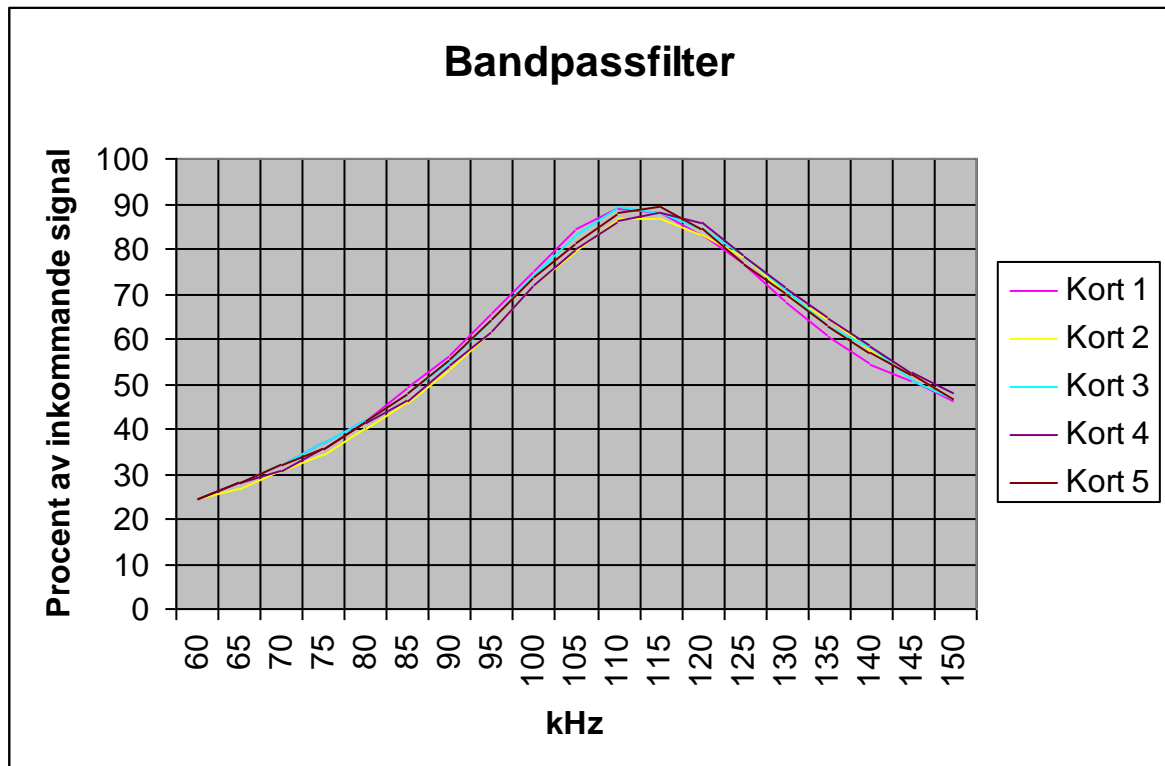
Med hjälp av kopplingsschemat på mönsterkortet designades en kretskortslayout (se avsnitt 9.3). Denna layout skickades till Malmö Mönsterkort AB, som tillverkade dessa åt oss. På varje kretskort skulle det



monteras 160 komponenter och vi hade komponenter till 5 kretskort. Montering skedde manuellt på Mikrodidakt AB och efter uppmätning och omlödning av ett par komponenter kunde vi fastställa att korten fungerade som förväntat.

#### 4.1.3.3 Bandpassfilter

En uppmätning av vårt bandpassfilter på inkommande signal gav följande diagram.



Figur 2: Bandpassfiltrets filterverkan

Uppmätningen visar att alla kort ligger rätt med sina filter på inkommande signal eftersom vi valt att använda oss av 110 kHz som bärvågsfrekvens. Bra filtrering är en förutsättning för lyckad kommunikation.

## 4.2 Mjukvara

Mjukvaran i våra kort utgjorde största delen av utvecklingstiden. Många tester gjordes med hjälp av en logikanalysator som gav oss ovärderlig information genom att förvissa oss om att signalerna på SPI-bussen hade rätt timing.

### 4.2.1 Masterprogrammet

Denna mjukvara är utvecklad för att kunna köras på en PC med .NET Framework 3.5 installerat. Syftet med programmet är att agera master i ett nätverk med flera slavar som kommunicerar enligt Modbusprotokollet.

#### 4.2.1.1 Utvecklingsmiljö

Programmet är framtaget i Visual Studio 2008 som ett ”fönster”-program i C#. För att kunna kommunicera på PC COM-port har klassen SerialPort använts som tillhör namespace System.IO.Ports.

#### 4.2.1.2 Programmets uppbyggnad

Som grundfunktion finns kommunikationen med PC COM-port. Detta gör att programmet kan skicka och ta emot meddelande med de anslutna detektorkorten. Metoden som läser tecken på serieporten och som avkodar ett meddelande körs på en egen tråd. Alla andra metoder körs på huvudtråden.

När programmet startas tar det reda på vilken eller vilka COM-portar som finns på PC och den första kommer att vara vald i rutan längst ner. Om ingen COM-port detekteras talar programmet om detta och avslutas. Man kan välja andra värden på baudrate än 4800 men eftersom länkkortet och detektorkorten är konfigurerade för 4800 Baud så är det meningslöst.

Programmet startar med att fylla på sändbufferten med 6 st meddelande som har var sin modbus-adress, nämligen adress 1 – 5 samt adress 255. Alla dessa meddelanden frågar efter configurationsordet. När man trycker på startknappen börjar programmet sända dessa meddelanden i tur och ordning från sändbufferten. När alla sänts, börjar sändningen om från början igen. Efterhand som eventuella detektorkort svarar på meddelanden, som motsvarar deras adress, kommer data i svaren att presenteras i var sin flik i GUI. Varje detektorkort som svarar tilldelas en flik i den ordning de svarar. När ett detektorkort svarat bygger programmet på med ytterligare frågor (meddelanden) för detta detektorkort och lägger dem i sändbufferten. Programmet fortsätter att sända alla frågor om och om igen och på detta sätt kommer till slut alla fält i alla modemflikar att visa all data från respektive detektorkort.

I GUI finns också en flik som heter System och här visas statistik på antalet sända meddelanden (requests) och på antalet mottagna meddelanden (responses) för respektive detektorkort som svarat. Varje detektorkort har också motsvarande statistik på antalet mottagna och sända meddelanden som också visas här. Dessutom finns en flik som heter Control och här kan man skapa meddelande som konfigurerar om detektorkorten. Man kan även plocka bort ett detektorkort och då nollställs innehållet i alla fälten för det kortet, både i modemfliken och i systemfliken. Samtidigt plockas samtliga frågor för detta detektorkort bort ur sändbufferten utom det som skapades när programmet startade.

Det kan vara värt att nämna att en svårighet med klassen SerialPort handlar om att läsa av inkommande tecken. Hur många tecken som klassen tagit emot från hårdvaran varierar från gång till gång. Det är inte så att om ett inkommande meddelande innehåller, som i vårt fall, 20 tecken (bytes) att klassens buffert tagit emot just 20 tecken. Det kan delas upp på t.ex. två omgångar med olika många tecken.

#### 4.2.2 Länkkortet

Detta kretskort tar emot meddelanden på en serieport och skicka dem vidare ut på 230-voltsnätet. Det tar också emot meddelanden på 230-voltsnätet och skickar dem vidare på serieporten.

##### 4.2.2.1 Utvecklingsmiljö

Eftersom MCU kommer från Silicon Labs har också programmet tagits fram i deras utvecklingsmiljö. Till denna miljö, där man kan editera, kompilera, ladda ner och debugga sin kod, följer Keils kompilator och länkare med. Denna gratisversion av Keils kompilator för C8051 har en begränsning i hur mycket kod den kan hantera, vilket är 0x800H bytes. Om koden är större än 0x800H, måste man använda en licensversion. Det går också bra att använda andra fabrikat än Keil.

##### 4.2.2.2 Periferienheter i MCU

Serieporten: Här används UART-interfacet som är inställt på 8 databitar, ingen paritet och 1 stopbit. För att driva baudrategeneratoren används Timer1 som är inställt på att generera en baudrate på 4800. Koden är förberedd för att lätt kunna ändra baudrate och tanken är att man via ett meddelande skall kunna göra detta. Timer3 är inställd för att mäta tiden mellan två meddelande, Inter Frame Delay. Koden är förberedd för att kunna hantera Inter Frame Delays för baudrates mellan 1200 och 4800. För att kunna ta hand om inkommande och utgående tecken på UART-interfacet används ett interrupt. Till Timer3 är också ett interrupt kopplat så att ett slut på ett meddelande kan detekteras.

230-voltsnätet: Detta sköts av modemkretsen som kommunicerar med MCU via ett SPI-interface samt två portpinnar. På SPI-interfacet är modemkretsen master och MCU slav men MCU styr i vilken riktning kommunikationen går samt om det ska gälla kommunikation av data eller register med hjälp av portpinnarna. Till SPI-interfacet är ett interrupt kopplat för att hantera inkommande och utgående tecken.

#### 4.2.2.3 Programmetts uppbyggnad

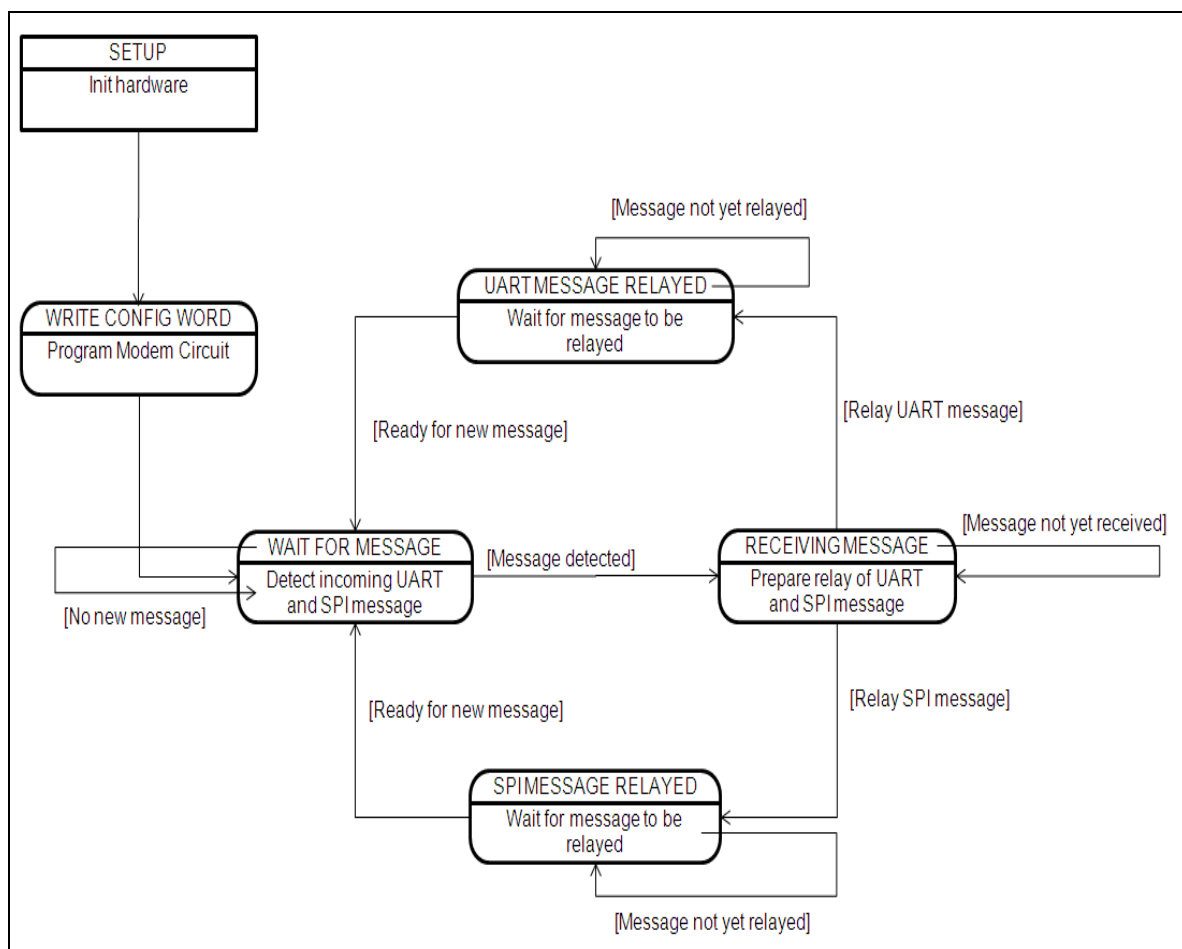
Programmet jobbar som en tillståndsmaskin (se figur 3) och har följande tillstånd: WRITE CONFIG WORD, WAIT FOR MESSAGE, RECEIVING MESSAGE, UART MESSAGE REPLAYED och SPI MESSAGE REPLAYED.

Vid uppstart initierar programmet alla MCU periferienheter och går till tillståndet WRITE CONFIG WORD där skrivning av konfigurationsordet till modem-kretsen sker. Konfigurationsordet har ett förutbestämt värde som inte går att ändra. Tanken är att detta ska gå att göra genom kommando via meddelande men är inte implementerat ännu.

Därefter hamnar programmet i tillståndet WAIT FOR MESSAGE vilket är utgångsläget. Här står programmet och väntar på att ett meddelande skall komma, antingen från serieporten eller från modemmet. Detektering av inkommande meddelande på serieporten är interrupt-styrt medan detektering av inkommande meddelande på SPI-interfacet är pollat. Meningen är att oavsett varifrån meddelandet kommer så skall inga fler meddelande tas emot förrän programmet återvänt till detta tillstånd. På denna punkt behöver programmet utvecklas vidare, t.ex. behöver UARTen stängas av när ett meddelande kommit in och startas när nya meddelande kan tas emot igen.

När ett meddelande kommer in, antingen på serieporten eller på modemmet, hoppar programmet till nästa tillstånd RECEIVING MESSAGE och väntar på att meddelandets slut skall detekteras. När meddelandet kommit in görs en beredning för att skicka meddelandet vidare som ser olika ut beroende på om meddelandet kommit in på serieporten eller modemmet.

Därefter hoppar programmet vidare till något av tillstånden UART MESSAGE RELAYED eller SPI MESSAGE RELAYED. Här väntar programmet på att meddelandet skall detekteras som färdigskickat. Till slut hoppar programmet tillbaka till utgångsläget och processen börjar om.



Figur 3: Statemaskin för länkkortet

#### 4.2.3 Detektorkortet

Detta kretskort avläser 4 analoga signaler från ett anslutet givarkort och sparar de digitaliserade värdena i ett EEPROM. Dessutom tar det emot ett meddelande på 230-voltsnätet, avkodar meddelandet och läser eller skriver begärd information från/till EEPROM.

##### 4.2.3.1 Utvecklingsmiljö

Se 4.2.2.1

##### 4.2.3.2 Periferienheter i MCU

Analog avläsning: Här används ADC-interfacet och dess MUX för att mäta från flera källor. Den är konfigurerad för att kunna göra 10 mätningar per sekund. Detta är den långsammaste hastigheten men ger den högsta upplösningen. Avläsningen sker med hjälp av pollning, det vill säga efter att en avläsning startats väntar programmet till avläsningen är klar.

230-voltsnätet: Detta sköts av modemkretsen som kommunicerar med MCU via ett SPI-interface samt två portpinnar. På SPI-interfacet är modemkretsen master och MCU slav men MCU styr i vilken riktning kommunikationen går samt om det ska gälla kommunikation av data eller register med hjälp av portpinnarna. Till SPI-interfacet är ett interrupt kopplat för att hantera inkommande och utgående tecken. Dessutom används ett externt interrupt för att detektera inkommande meddelanden till skillnad från länkkortet som använder ett pollat system för denna uppgift. Skälet att använda ett externt interrupt är att detta kort sköter uppgifter som tar lite längre tid och gör att det inte skulle hinna med att starta SPI-interfacet i tid annars.

Läsa/Skriva till EEPROM: För att kommunicera med EEPROM används ett interface som kallas Smbus och är kompatibelt med I2C. Till detta är ett interrupt kopplat för att hantera inkommande och utgående tecken. Dessutom är två Timerinterrupt igång för att dels generera I2C-klockan och dels för att detektera om en slav håller I2C-klockan låg i 25ms.

#### *4.2.3.3 Programmets uppbyggnad*

Programmet jobbar som en tillståndsmaskin (se figur 4) och har följande tillstånd: INIT, WRITE CONFIG WORD, WAIT FOR MESSAGE, DECODE MESSAGE och MESSAGE REPLIED.

Vid uppstart börjar programmet med att initiera alla MCU periferienheter och går sen till tillståndet INIT, där detektering av läge (mode) sker. Om man satt en bygel på kortet mellan jord och testpunkten E8 (Watchdog) detekteras detta vid uppstart och i tillståndet INIT sätts i så fall kortets adress till 255. Dessutom skrivs det förutbestämda konfigurationsordet och ett standardvärde för samtliga kalibreringsfaktorer samt en standardtext för kortets ID till EEPROM. Detta används för att kunna starta från en default-inställning. Om ingen bygel detekteras läses istället kortets adress samt konfigurationsordet från EEPROM.

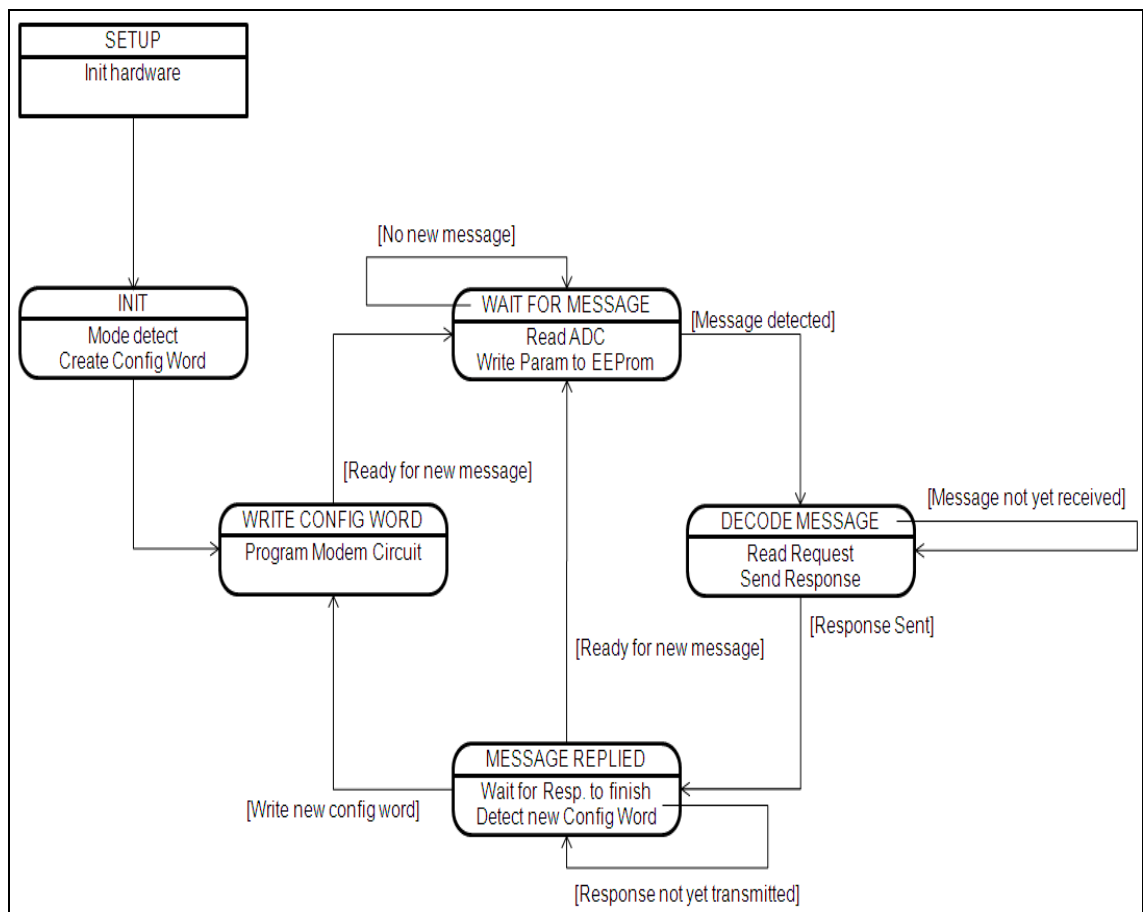
Därefter hoppar programmet till tillståndet WRITE CONFIG WORD där skrivning av konfigurationsordet till modemkretsen sker. I detta program går det att ändra konfigurationsordet genom kommando via meddelande, men i praktiken kommer det inte att göras eftersom alla program i hela kedjan är beroende av att samma konfiguration används.

Sen hoppar programmet till det tillståndet som är utgångsläget nämligen WAIT FOR MESSAGE. Här står programmet och väntar på att ett meddelande skall komma in. Under tiden som det väntar görs avläsning av de analoga signalerna från det anslutna givarkortet samt skrivning av statistik till EEPROM. Detektering av ett inkommande meddelande sker med hjälp av ett

externt interrupt. När ett meddelande tagits emot accepteras inga nya meddelande förrän senaste meddelande blivit behandlat.

När ett inkommande meddelande detekteras skiftas tillståndet till DECODE MESSAGE. Här väntar programmet tills hela meddelandet kommit in och sen avkodas och kontrolleras meddelandet. Ett svar skapas och data läses eller skrivs till EEPROM beroende på vad meddelandet innehöll för kommando.

När initieringen av svaret är klart går programmet till nästa tillstånd MESSAGE REPLIED och väntar på att meddelandet skall detekteras som färdigskickat. Om meddelandet har indikerats som att konfigurationsordet behöver skrivas till modemmet kommer nästa tillstånd att bli WRITE CONFIG WORD och när det är gjort kommer programmet tillbaka till utgångsläget WAIT FOR MESSAGE och processen börjar om. Om meddelandet inte indikerats som att konfigurationsordet behöver skrivas hoppar programmet direkt till WAIT FOR MESSAGE.



Figur 4: Statemaskin för detektorkortet

## 4.3 Kommunikationsprotokoll

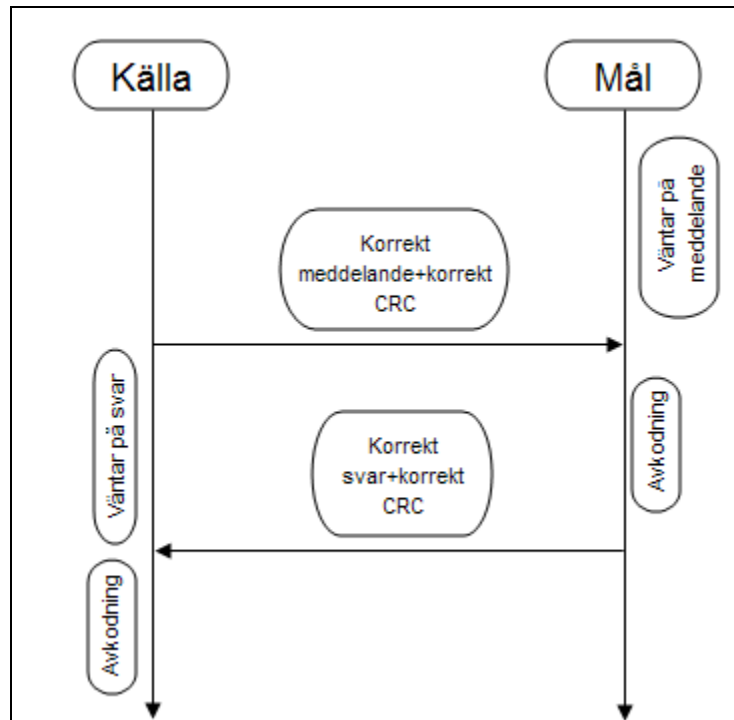
Ett kommunikationsprotokoll skulle bestämmas och X10 föreslogs. Efter lite efterforskningar om X10, som är en standard för hemautomation, visade det sig att detta protokoll inte skulle täcka våra behov. Vi behövde ett protokoll som mer var riktat mot industri och hade möjlighet att kommunicera med både PC och en PLC (Programmable Logic Controller). Modbus var ett protokoll som passade in på våra behov. Det är väldokumenterat och är mer eller mindre en standard i automationssammanhang. Själva Modbus protokollet har tre varianter att välja mellan. ASCII, RTU och TCP är de tre varianterna, där ASCII och RTU var de aktuella för oss eftersom vi skulle ha kommunikation över serieport. Eftersom Modbus standarden säger att RTU är default varianten fann vi ingen anledning att implementera den långsammare ASCII varianten.

### 4.3.1 Modbus

I Modbusstandarden [3] fanns en hel del fördefinierade funktioner men även stora möjligheter till att definiera egna vid behov. I vår prototyplösning behövdes tre olika funktioner där alla tre var fördefinierade av standarden. De tre funktionerna var: Read Holding Registers (funktionskod: 03), Read Input Registers (funktionskod: 04) och Write Multiple Registers (funktionskod: 16). Dessa funktioner används till att skriva och läsa i vårt EEPROM på kretskorten. Read Holding Registers läser de olika inställningarna som modemkortet har, medan Read Input Registers läser av våra mätvärden från externa källor, i detta fall Samons gasdetektorkort. Write Multiple Registers används för att skriva in nya inställningar i registret på våra detektorkort. Dessa inställningar läses in och inställningar ändras efter behov.

Dämpningar och störningar från annan inkopplad utrustning är en signifikant källa till förvrängda meddelanden. För att kontrollera att meddelandens integritet har bibehållits från källa till mål, har Modbusprotokollet en felkontroll som standard. I vårt fall använde vi oss av CRC eftersom det är standard för Modbus RTU





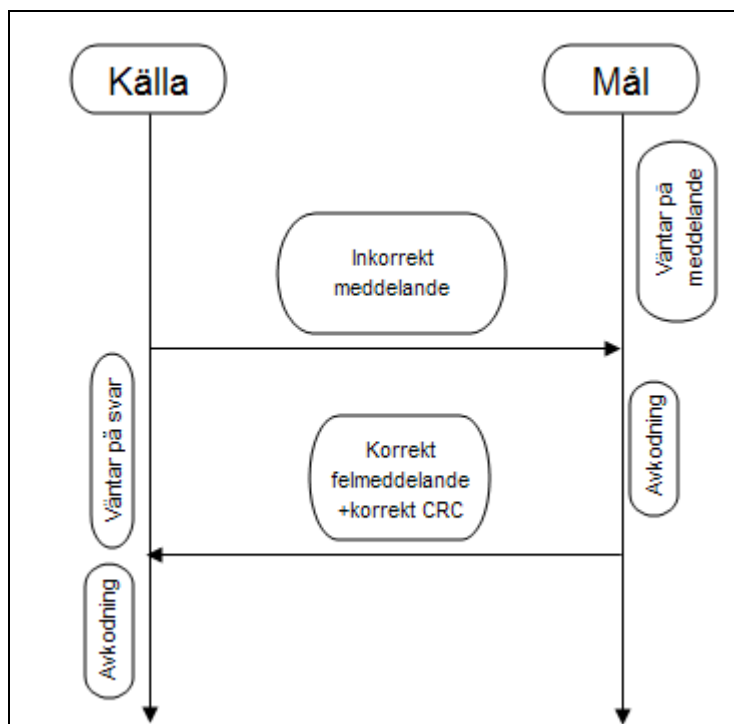
Figur 5: Ett korrekt meddelande och svar

I figur 5 visas hur en korrekt kommunikation mellan källa och mål sker. Källan sänder ett meddelande med korrekt mottagaradress. När målet känner av att ett meddelande har dess adress påbörjar den avkodningen. Meddelandets olika datafält är beroende av vilken funktionskod som finns i meddelandet. En översikt av de funktioner vi har valt att använda finns i figur 6.

Funktionskod 16 meddelande						
Adress	Funktion	Startadress	Antal Register	Antal bytes	N * Bytes data	CRC
Funktionskod 16 svar						
Adress	Funktion	Startadress	Antal Register	CRC		
Funktionskod 3 och 4 meddelande						
Adress	Funktion	Startadress	Antal Register	CRC		
Funktionskod 3 och 4 svar						
Adress	Funktion	Antal bytes	N * Bytes data	CRC		
Felmeddelande						
Adress	Funktion + 0x80	Felkod	CRC			

Figur 6: Översikt av våra meddelanden och svar

En förutsättning för att avkodning ska påbörjas är att adressen måste stämma överens med målets bestämda adress. Alla adresser utom den egna ignoreras och inga svar sänds. Vid rätt adress kontrolleras först att funktionskoden är en kod som stöds av målet. Om den stämmer fortsätter kontroll av fältet ”Antal Register”. Detta fält måste ligga inom ett visst intervall för att godkännas. Intervallet i vårt fall ligger mellan 1 och 4. Nästa kontroll är av fältet ”Startadress”, även här gäller det att adressvärdet ligger inom rätt intervall, i vårt fall 0x0000 till 0x003E. Skulle någon av dessa kontroller gå fel avbryts vidare avkodning och ett felmeddelande genereras (se figur 4). Felmeddelandet returnerar adressen och funktionen men adderat med 0x80. En felkod beskriver vilken operation i avkodningen som gått fel. Den sista kontrollen som görs på ett meddelande är CRC. Om den stämmer behandlas meddelandet och ett svar genereras och skickas. Skulle CRC vara fel ignoreras meddelandet då det är korrupt och inte tillförlitligt.



Figur 7: Ett inkorrekt meddelande och ett korrekt felsvar

För att lättare förstå principerna runt kommunikation med Modbus använde vi oss av ett program som heter Modbus Test Pro [4]. Detta program användes som simulator för vårt masterprogram, som ännu inte var framtaget. Modbus Test Pro skickar Modbusmeddelanden till serieporten på datorn och vidare in till serieporten på vårt kretskort. Vid ett svar på en förfrågan så kontrolleras CRC i Modbus Test Pro som test av meddelandets integritet.

## **5 Test**

För att undersöka effekterna av fasöverhörning och dämpning utfördes ett funktionstest av systemet. Fyra detektorkort placeras ut på fyra olika platser i fyra olika miljöer (se avsnitt 9.5). Kommunikation upprättades i 60-90 minuter och statistiken sparades för jämförelse. Testerna utfördes först med endast kommunikation på en fas dvs. länkkortet var endast kopplad på en av faserna. Därefter gjordes testet om med länkkortet inkopplat på alla tre faser via ett 3-fasuttag. Modemnummer är inte konsekventa mellan testerna då de får sina nummer i den följd de svarar.

### **5.1 Sal C428, liten datorsal**

Denna sal fungerade som central i kommunikationen. Här befann sig länkkortet och ett av detektorkorten. Salen innehöll vid tidpunkten för testet 6 stationära datorer som var igång, 6 skärmar av äldre modell varav 2 igång, en skrivare och två bärbara datorer.

### **5.2 Sal C451, stor datorsal**

I denna sal fanns 25 stationära datorer igång och lika många skärmar. I salen finns även två skrivare. Antal skärmar igång varierade under testet då dessa salar används av studenter.

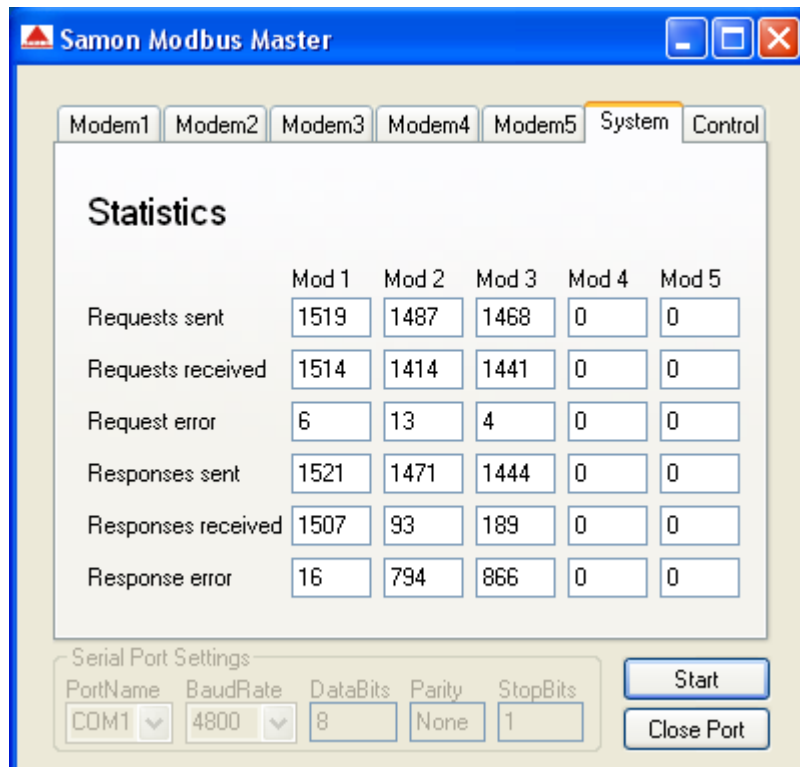
### **5.3 Sal C421, laborationssal**

Laborationssalen innehåller en mängd utrustning som inte var igång under testerna. 8 stationära datorer var igång, 8 skärmar var inkopplade men antalet igång varierade. Utöver detta fanns det 8 oscilloskop, 8 tongeneratorer, 8 multimetrar och 8 spänningskällor inkopplade men ej igång.

### **5.4 Sal C422, föreläsningssal**

Inga stationära datorer eller skärmar finns inkopplade i denna sal. En eller två bärbara datorer, tillhörande studenter, var inkopplade under testerna.

## 5.5 1-fas testet



Figur 8: Statistik över 1-fas testet

I 1-fas testet fick vi kontakt med 3 av de 4 modemerna (se figur 8).

Radrubrikerna betyder följande:

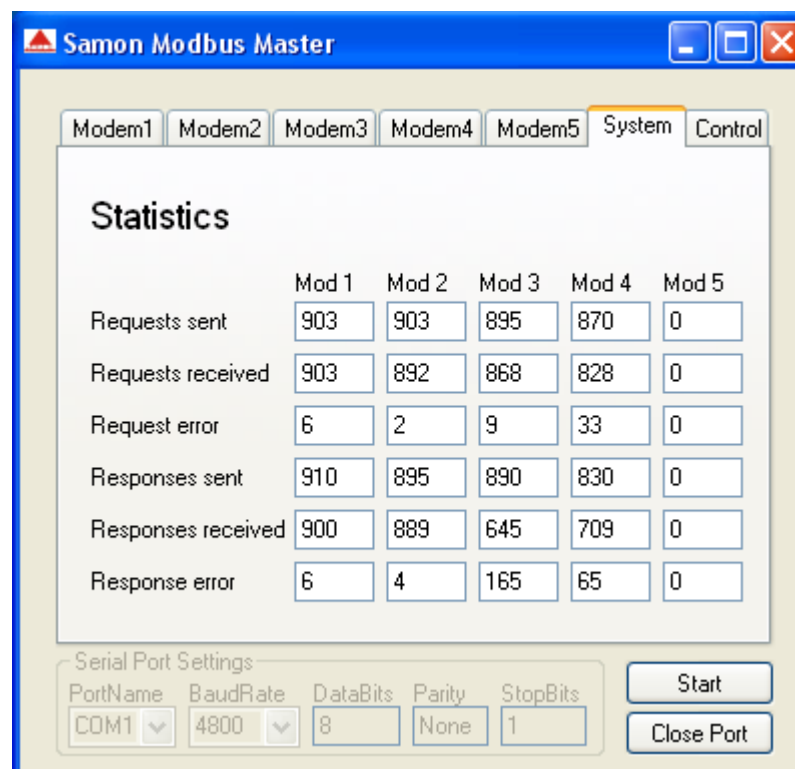
- Requests sent – antalet meddelande(frågor) som skickats från masterprogrammet
- Requests received – antalet meddelande(frågor) som detektorkortet tagit emot
- Request error – antalet felaktiga meddelande(frågor) som detektorkortet tagit emot
- Responses sent – antalet meddelande(svar) som skickats från detektorkortet
- Responses received – antalet meddelande(svar) som masterprogrammet tagit emot
- Response error – antalet felaktiga meddelande(svar) som masterprogrammet tagit emot

Summan av antalet mottagna meddelande och antalet felaktiga meddelande borde vara lika med antalet sända meddelande. Detta gäller i båda riktningarna, d.v.s. både för frågor och svar. Om antalet skickade och mottagna+felaktiga meddelande inte är lika, beror det dels på att uppdateringen av fälten inte sker i rätt ordning och dels på att meddelanden helt enkelt inte detekterats alls. Man får tänka på att masterprogrammet ”vet” hur många meddelande som skickats vilket detektorkortet inte gör. Detta gör

att statistiken som sköts av detektorkortet ser bättre ut än statistiken som sköts av masterprogrammet. Med andra ord, om ett meddelande helt missas av detektorkortet syns detta inte statistiken som sköts av detektorkortet men kommer att synas i statistiken som sköts av masterprogrammet. Dessutom kommer meddelande som registreras som felaktiga i detektorkortet också att registreras som felaktiga i masterprogrammet.

Modem 1 var detektorkortet i C428, alltså i samma rum som länkkortet. Detta modem fungerade bäst och det kan man se genom att antalet mottagna meddelande är högt och antalet fel är lågt. Dessutom gäller detta i båda riktningarna. Anledningen är troligen att avståndet är kort och därför dämpas signalen inte så mycket samt att signalen inte utsätts för några störningar. Modem 2 var detektorkortet i C421 och modem 3 var detektorkortet i C422. Båda fungerade betydligt sämre än modem 1 med modem 2 som sämsta kort eftersom antalet mottagna meddelande är lägst. Det man kan se är att kommunikationen fungerar sämst i riktning från detektorkorten tillbaka till länkkortet vilket är märkligt men kanske kan förklaras med att länkkortet eventuellt var bäst på att sända av alla kort. Det fjärde detektorkortet svarade överhuvudtaget inte alls och därför är siffrorna i kolumnen för modem 4 alla lika med 0. Anledningen är att detta detektorkortet var anslutet på en annan fas än länkkortet och då fungerar inte kommunikationen alls.

## 5.6 3-fas testet



Figur 9: Statistik för 3-fas test

I 3-fas testet fick vi kontakt med alla inkopplade modem (se figur 9). Modem 1 var detektorkortet i C428 vilket var samma som i 1-fas testet. Modem 2 var detektorkortet i C422 vilket var modem 3 i 1-fas testet. Modem 4 var detektorkortet i C421 vilket var modem 2 i 1-fas testet. I detta test fungerade modem 2 bäst men modem 1 är i princip lika bra. Modem 4 kommer inte långt efter heller. Överhuvudtaget fungerade dessa modem markant bättre än i 1-fas testet och den enda förklaringen vi kan komma på är att detta test råkade utföras när den allmänna störningsnivån var svagare. Vi har tidigare kunnat se, vid ett flertal tillfällen, att kommunikationskvaliteten varierat rent allmänt från tidpunkt till tidpunkt.

Den största skillnaden är att i detta test fungerade kommunikationen med detektorkortet i C451, modem 3. Resultatet för modem 3 är mycket bättre än för modem 2 och 3 i 1-fas testet och det beror troligen på, som nämnts tidigare att vid tillfället för 1-fas testet var den allmänna kommunikationskvaliteten sämre. Dock kan man se att modem 3 fungerade sämst av alla modem i detta test. Vår tolkning av detta är att miljön i den salen är ganska svår för PLC på grund av alla datorer som är inkopplade där med sina nätfilter som dämpar signalen. Fortfarande skulle kommunikationen med vilket modem som helst i detta test vara tillräckligt bra för våra behov.

## **6 Vidareutveckling**

Vårt arbete har lagt en grund för fortsatt utveckling och testning av systemet. Vi har under arbetets gång fått nya idéer och förslag på tester men som inte kunnat implementeras eller utföras p.g.a. tidsbrist. De första uppslagen vi fick för att utveckla arbetet var att testa de olika modemkretsarna från leverantörerna. Kretsen vi använde var den enklaste modellen från ST, det finns mer avancerade kretsar tillgängliga. Vi har även förhört oss om att i skrivande stund finns det nyare modemkretsar på väg ut på marknaden. Även andra leverantörer jobbar vidare med utveckling av PLC modem vilket gör att framtiden inom PLC är högst intressant.

En utveckling av kretskortet där det är förberett att byta frekvens lätt är ett steg mot ökad flexibilitet. Det hade varit bra att kunna utforska i vilka frekvenser det finns mest störningar och på detta sätt kunna hitta eventuella störkällor. Det skulle ge slutkunden möjligheter att antingen eliminera störkällor eller byta frekvens.

Ett test vi önskat att vi haft tid till är att minska längden på våra meddelanden. Istället för att skicka meddelanden som är långa eller ger långa svar tillbaka, minimera längden av dessa. Vi misstänker att felprocenten av felaktiga

meddelanden hade förbättrats, då vi tror att långa meddelanden löper större risk för att bli störda eller på andra sätt korrupta.

I protokollet hade nästa steg varit att implementera Modbus TCP vilket används mer och mer som standard. Detta underlättar även för slutkunden av produkten eftersom i princip alla datorer och PLC (Programmable Logic Controller) har en ethernetport medan serieporten har bortprioriterats på många nyare datorer.

Problemet med två PLC system kopplade på samma ledning, använder samma frekvensband och utan CSMA är något som bör utforskas. Effekterna av två system som kolliderar bör studeras och lämpliga åtgärder föreslås.

I marina sammanhang måste man undersöka klassningssällskapens regler och direktiv för PLC. Vad vi har fått reda på så finns det inga i dagsläget gällande standarder för PLC på fartyg. Detta är något klassningssällskapen själva beslutar om. En studie om standarder under utveckling och vad dessa innefattar bör göras om vidareutveckling ska ske.

En lösning för hur kommunikation förbi undercentraler kan ske hade varit intressant. Vårt system kan bara kommunicera på ena sidan om centralen, då den är galvaniskt åtskild från huvudcentralen.

## **7 Slutsats**

Att ta fram en lösning för PLC är inga större problem och detta lyckades vi också med. Problemet handlar mer om vilken typ av lösning man skall välja eftersom det finns så mycket att välja på. De flesta tillverkare av transceiver/modemkretsar har hållit på med PLC i många år och levererat miljontals kretsar till marknaden om man skall tro vad dom själva säger. Vad som är svårt läsa sig till är hur bra det fungerar i praktiken, vilka avstånd klarar tekniken av att kommunicera på. Med vår lösning kommer Samon AB att kunna installera en provanläggning och på så sätt lära sig mer om vad som behöver tas fram.

Vår slutsats är att kommunikationsmässigt ser vi att det fungerar bra när alla tre faser används och man inte förlitar sig på fasöverhörningen. För att få full täckning i större installationer med många undercentraler måste man ta reda på hur elnätet i en aktuell byggnad ser ut. Man behöver veta vilka spänningsnivåer som existerar i byggnaden eftersom varje transformator kommer troligen att behöva en brygga.

Moduleringstekniken bör väljas med tanke på hur många enheter som är kopplade. Ju fler enheter det finns på samma frekvens, desto högre krav på överföringskapacitet. Detta kombinerat med CSMA/CD, för att kontrollera mediets tillgänglighet före sändning kan med fördel väljas även om, som i vårt fall, datamängden per enhet är liten. Enda anledningen att välja ett protokoll typ KNX eller LonWorks är ifall man vill kunna styra/läsa av enheter som sköter olika uppgifter i samma byggnad från en gemensam central. Om man är först med att installera en funktion som utnyttjar PLC i en byggnad kan man välja protokoll fritt, under förutsättning att det finns kapacitet, minnesutrymme och CPU-kraft att senare implementera ett standardiserat protokoll.

## 8 Referenser

1. Ulrich Stein, Jan Carlsson, Jan Welinder, 2006. "Signalöverföring på lågspänningsnätet", Elforsk rapport 06:22.
2. Svensk Standard, 2001. "Signalöverföring i lågspänningsinstallationer i frekvensområdet 3 kHz till 148,5 kHz – Del 1: Allmänna fordringar, frekvensband och elektromagnetiska störningar", SS-EN 50065-1
3. Modbus Organization, 2006. "Modbus application protocol specification V1.1b", <http://modbus-ida.org/>
4. ModBus Test Pro v2.3.2, Tim Kirk, Rogue Engineering Inc. <http://rogue-engr.com>
5. AN2451 Application note rev.3 , January 2010, STMicroelectronics
6. Luis F. Montoya, "Power Line Communications Performance Overview of the Physical Layer of Available protocols". <http://masters.donntu.edu.ua/2006/kita/avramenko/library/lib9a.pdf>
7. PLC Forum, [http://www.plcforum.org/frame\\_plc.html](http://www.plcforum.org/frame_plc.html), 2010-06-13
8. X10.com, <http://www.x10.com/homepage.htm>, 2010-06-13
9. KNX Association, <http://www.knx.org/>, 2010-06-13
10. BACnet Website, <http://www.bacnet.org/>, 2010-06-13
11. Echelon Corporation, <http://www.echelon.com/>, 2010-06-13
12. STMicroelectronics, <http://www.st.com/stonline/>, 2010-06-13
13. Cypress Semiconductors, <http://www.cypress.com/>, 2010-06-13
14. Tobias Björkman, "Elnätskommunikation för mätaravläsning", 2009:123CIV, <http://epubl.luth.se/1402-1617/2009/123/LTU-EX-09123-SE.pdf>, 2010-03-30



## **9 Bilagor**

### **9.1 Förkortningar**

ADC - Analog-to-Digital Converter

AMR - Automatic Meter Reading

ASCII - American Standard Code for Information Interchange

CRC - Cyclic Redundancy Check

CSMA - Carrier Sence Multiple Access

CSMA/CD – Carrier Sence Multiple Access / Collision Detect

EEPROM - Electrically Erasable Programmable Read-Only Memory

FCC – Federal Communications Commission

MCU - Micro Controller Unit

PC - Personal Computer

PLC - Powerline communication alt. Programmable Logical Central

RTU - Remote Terminal Unit

SAR - Successive Approximation

SMbus - System Management Bus

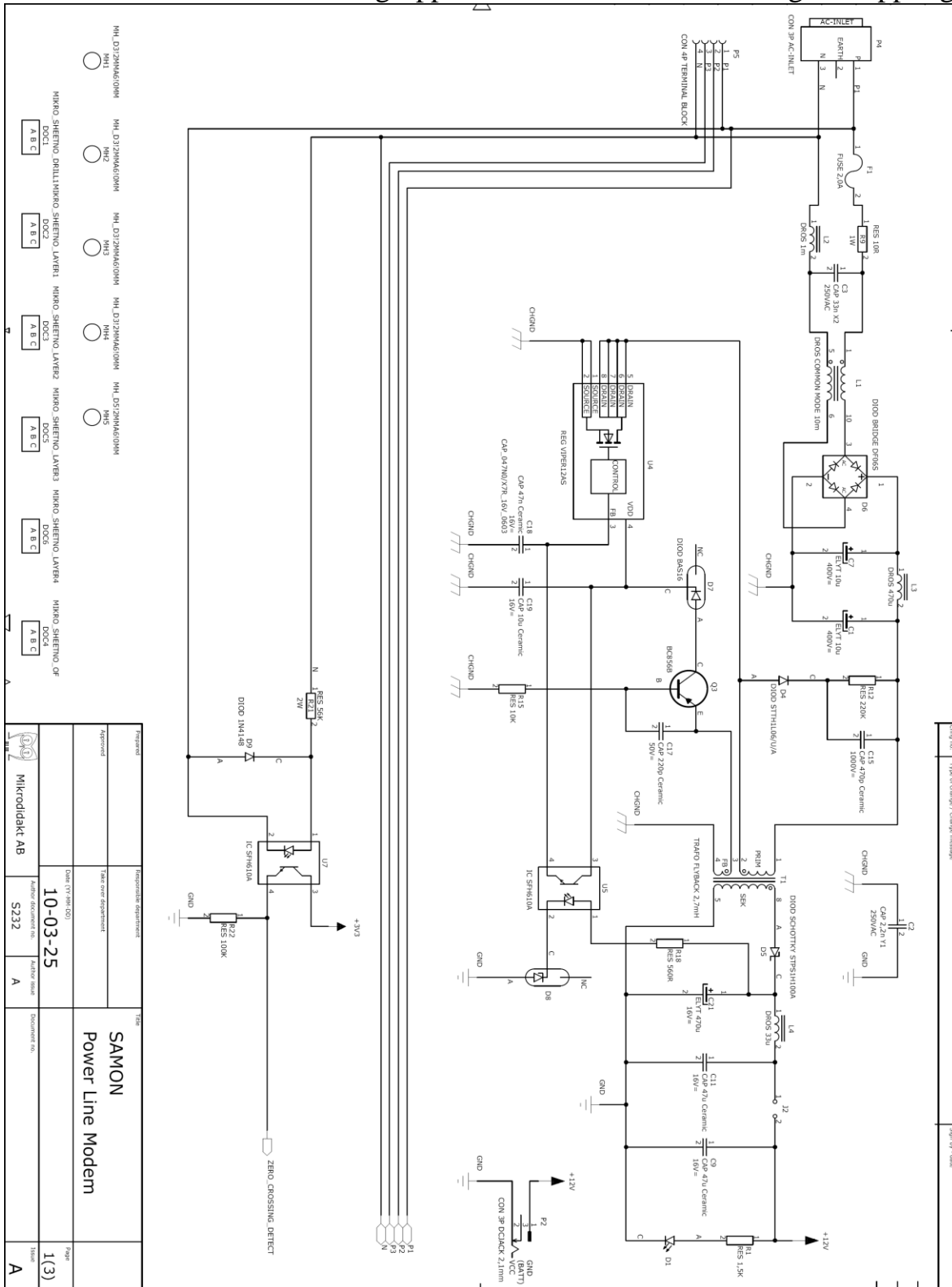
SPI - Serial Peripheral Interface

TCP - Transmission Control Protocol

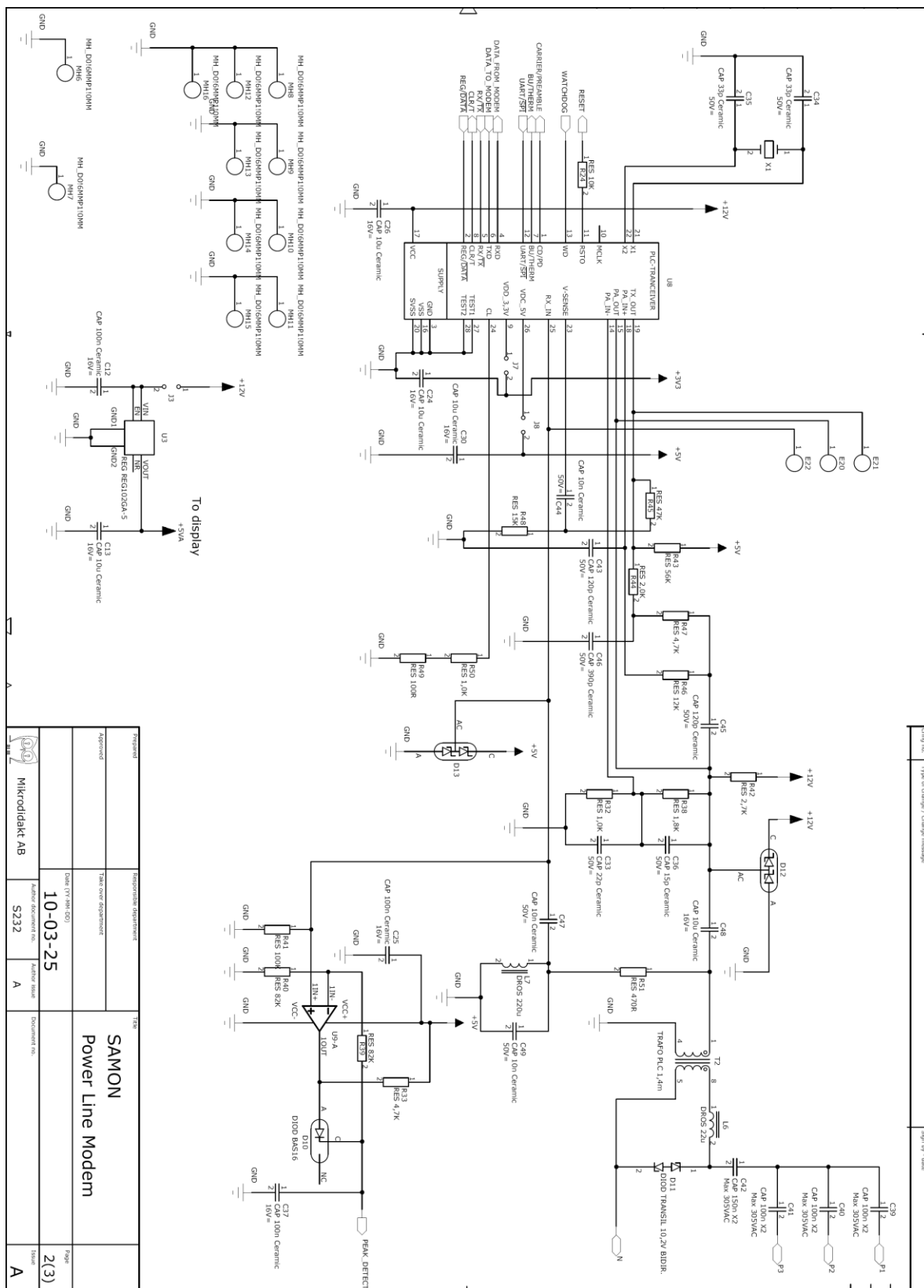
UART - Universal Asynchronous Receiver/Transmitter

## 9.2 Kopplingschema

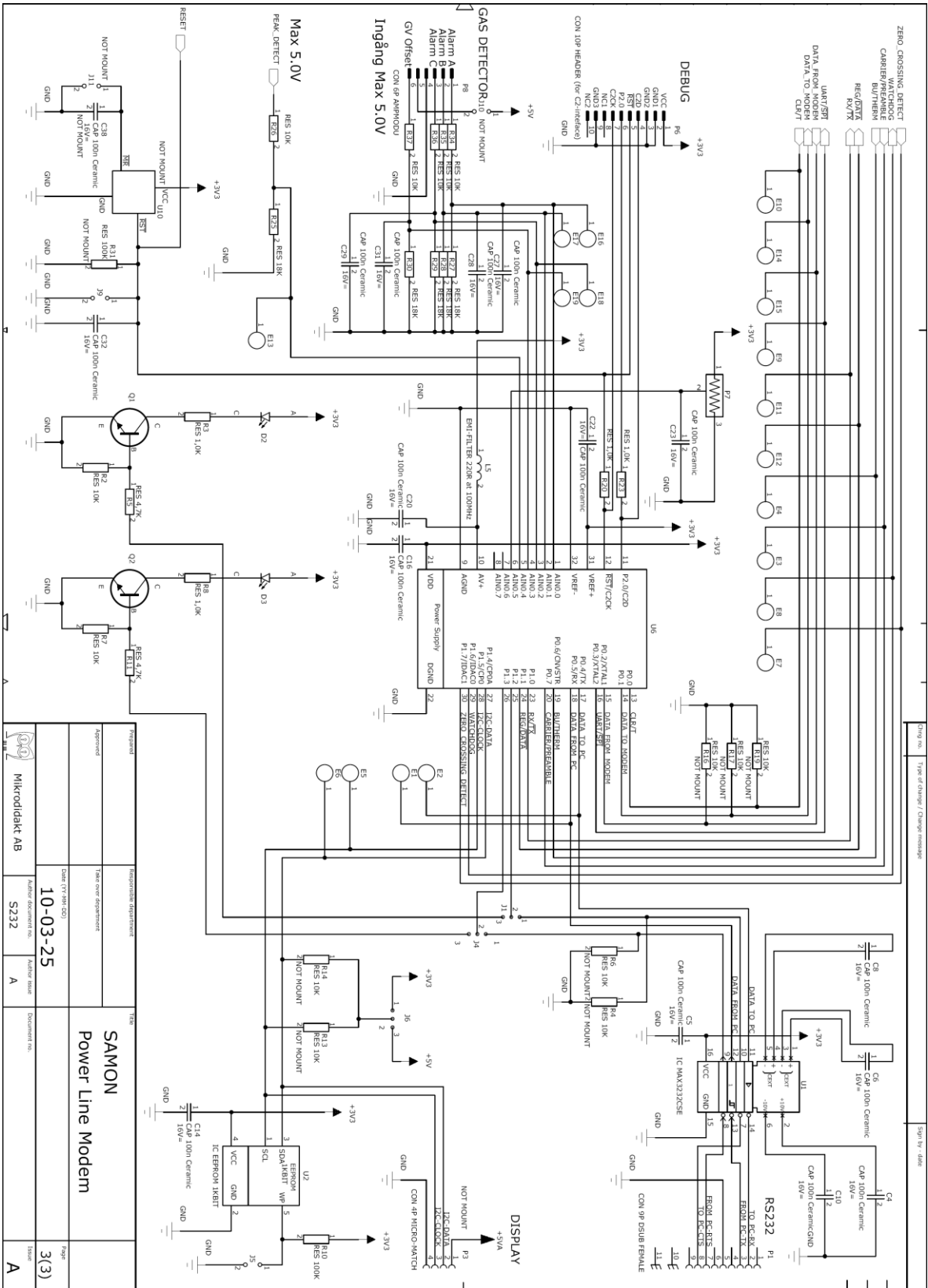
Kopplingschemat är uppdelat i tre delar och lika många figurer. Varje figur behandlar en funktion eller en grupp av funktioner som har en logisk koppling.



Figur 10: Denna del visar inkopplingen mot 230-voltsnätet och switchregulatorn som skapar matningsspänningen på 12V. Här finns också nollgenomgångsdetektorn.

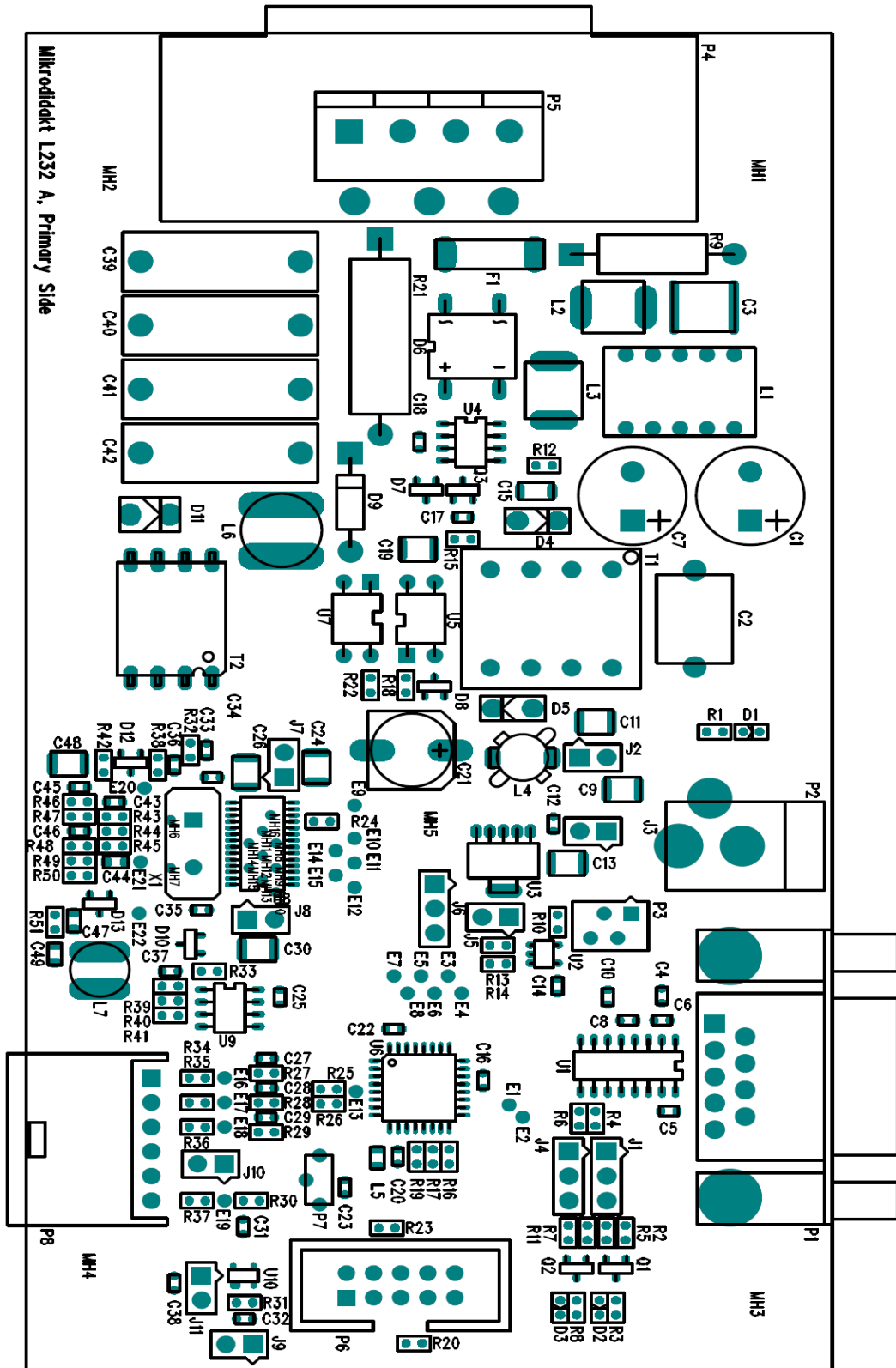


Figur 11: Här finns modemkretsen med sina filter för inkommande och utgående signaler samt den inbyggda förstärkaren. Här finns också spänningsregulatorerna för både 5V och 3,3V som finns inbyggda i modemet. Dessutom finns här OP-förstärkaren som mäter toppvärdet på den mottagna signalen.



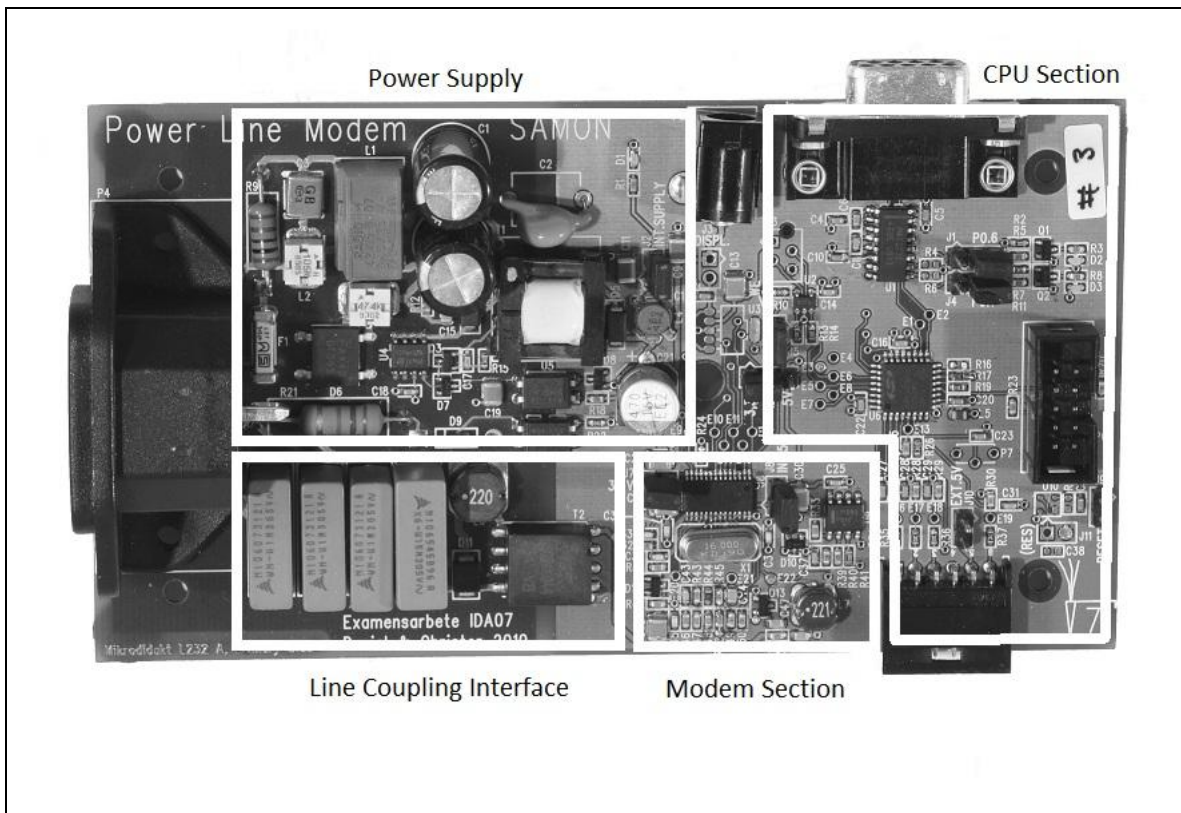
Figur 12: Här visas CPU med RSR232-interfacet, anslutningen till gasetektorns kort, anslutningen till debugutrustningen och EEPROM.

### 9.3 Komponentplacering



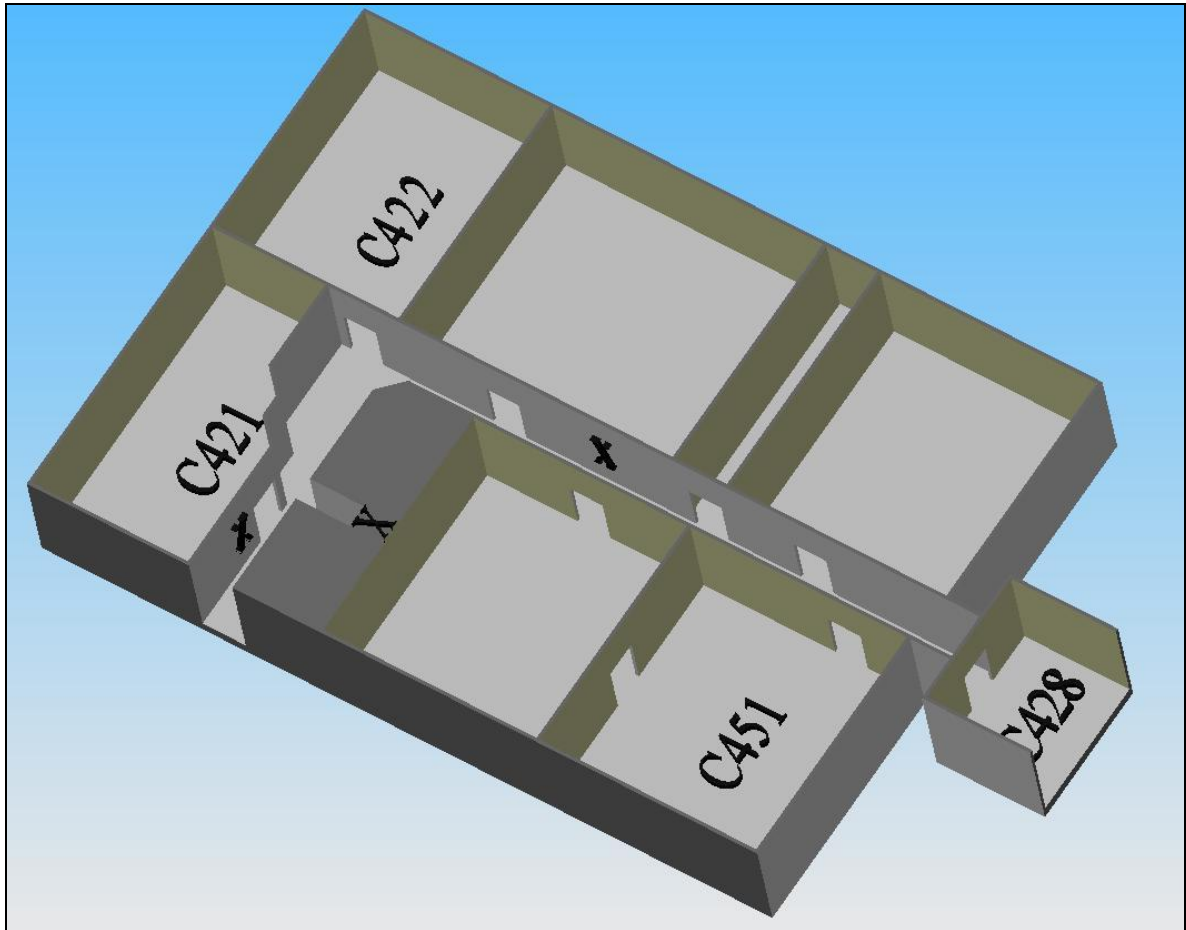
Figur 13: Layout över komponentplacering med lödpunkter för både hål- och ytmonterade komponenter markerade.

## 9.4 Komponentgruppering



Figur 14: En översikt av ett monterat kretskort med alla komponenter. De olika grupperingarna har markerats och funktionen för gruppen har angetts.

## 9.5 Plan 4 i C-huset



Figur 15: En ej skalaenlig skiss av plan 4 i C-huset. De rum som tester utförts i har fått sina nummer inskrivna. Undercentralerna är markerade med X. Ungefärligt avstånd från vägg motsatt dörren i C428 till vägg motsatt dörr i C421 är 60meter.