

Undersökning och framtagning av centralenhet för ONE Demand

DARDAN BEHRAMI

HARIS HADZIMESINOVIC

BACHELOR'S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY





Undersökning och framtagning av centralenhet
för ONE Demand
av

Dardan Behrami
Haris Hadzimesinovic

Abstract

This thesis was done in collaboration with One Nordic AB. The purpose was to examine the system One Demand, a system used for sampling measurement data, to begin the development of a new version of the system. The new system is comprised of two parts, a central unit and a graphical user interface.

The first step was to find a single board computer, on which the computer program is to be developed, to replace the old central unit. The result was a Raspberry Pi 3 model B which through its digital inputs is able to read pulses from energy meters. Depending on what energy meter is being used and its settings, each pulse is equal to a certain amount of energy in kWh. The GUI, which is written in Python, can count these pulses, allowing a user to set what each pulse amounts to for each meter. Sampling is done, for each meter separately, both by visualizing the energy with a graph and by logging. The graph draws the measured energy in a one-hour period. After an hour has passed the measured energy is first logged on a file before being reset to zero. The graph itself is visualized on a period of two hours and a screenshot is taken before the graphing begins for the next two-hour period.

Finally, a technical specification was written. This document specifies the technical data of the central unit and contains user manuals for the central unit and the GUI.

Keywords: Single board computer, GUI, Raspberry Pi, power consumption, kWh.

Sammanfattning

Detta examensarbete gjordes i samarbete med ONE Nordic AB. Syftet var att undersöka systemet ONE Demand, ett system som används för att samla in mätvärden, för att börja utveckla en ny version av systemet. Systemet består av två delar, en centralenhet och ett datorprogram.

Det första steget var att hitta en enkortsdator vars en GUI skulle programmeras på och ersätta befintlig centralenhet. Resultatet blev att en Raspberry Pi 3 model B köptes in. Genom sina digitala ingångar kan RPi 3 läsa av pulser från energimätaren. Beroende på vilken energimätare som används och dess inställningar är varje puls lika med en viss mängd energi i kWh. Användargränssnittet, som är skrivet i Python, visar räkningen av dessa pulser vilket möjliggör det för användaren att se hur stor förbrukning varje energimätare mäter upp. Förbrukningen visas även upp i en graf med perioden 1 timme. Efter varje timme loggas förbrukningen i en extern fil.

Slutligen skrevs en teknisk specifikation. Detta dokument anger enkortsdatorns tekniska data och innehåller manualer för installation av RPi 3 och GUI.

Nyckelord: Enkortsdator, GUI, Raspberry Pi, energiförbrukning, kWh

Förord

Vi skulle vilja tacka ONE Nordic AB för möjligheten att få utföra ett examensarbete i samarbete med dem. Ett speciellt tack till vår handledare på ONE David Wadst. Vi skulle vilja tacka Jasir Sharif från ONE för hans stöd genom hela processen. Vi skulle vilja tacka vår handledare på LTH, Mats Lilja. Hans entusiasm och glädje har smittats av under hela processen. Avslutningsvis vill vi tacka Christian Nyberg för hans tydliga direktiv gällande examensarbetet.

Innehåll

1	Inledning	7
1.1	Bakgrund	7
1.1.1	ONE Nordic AB	7
1.1.2	ONE Demand	7
1.2	Syfte	9
1.3	Målformulering	9
1.4	Problemformulering	10
1.5	Motivering av examensarbetet	10
1.6	Avgränsningar	10
2	Teknisk bakgrund	11
2.1	CPU	11
2.2	ED3000	12
2.3	ONE Demand	13
2.4	Enkortsdator	17
2.4.1	Firefly-RK3288	17
2.4.2	Raspberry Pi 3 model B	17
2.4.3	BeagleBoard	17
2.4.4	HummingBoard Base	18
2.4.5	Jämförelse	18
2.5	Programmeringsspråk	19
2.5.1	Forth	19
2.5.2	Java	19
2.5.3	Python	19
2.6	Specifikationsdokument	20
2.7	GPL och LGPL	20
2.8	CE-märkning	21
2.9	EMC-direktivet	22
2.10	S0-gränssnitt	22
2.11	IEC 60529	22
3	Metod	23
3.1	Fas: Förstudie	23
3.2	Fas: Undersökning	23
3.3	Fas: Utformning av kravlista	23
3.4	Fas: Val av enkortsdator	23
3.5	Fas: Test 1	24
3.6	Fas: Val av programmeringsspråk	24
3.7	Fas: Programmering av GUI	24
3.8	Fas: Test 2	24
3.9	Fas: Test 3	24
3.10	Fas: Vad för tester ska utföras för CE-märkning?	24
3.11	Fas: Slutsats	25
3.12	Källkritik	25

4	Analys	27
4.1	Utformning av kravlista	27
4.2	Val av enkorts dator	27
4.3	Val av programmeringsspråk	28
5	Test	29
5.1	Test 1	29
5.2	Test 2	31
5.3	Test 3	32
6	Resultat	33
6.1	Varför uppfattas ED3000 som långsam?	33
6.2	Vilka krav måste enkorts datorn uppfylla?	33
6.3	Raspberry Pi 3 model B	34
6.4	Test 1 resultat	35
6.5	Vilka krav ska en GUI uppfylla?	36
6.6	Val av programmeringsspråk för GUI	36
6.6.1	Utvecklad GUI	37
6.7	Test 2 resultat	37
6.8	Test 3 resultat	37
6.9	Statisk IP-adress för RPi 3	39
6.10	Expansionsmöjligheter för RPi 3	40
6.11	Utseende för specifikationsdokument	40
6.12	Test för CE-märkning	41
6.12.1	Kapsling	41
6.12.2	Strömadapter	41
6.13	Färdig produkt	41
7	Slutsats	45
7.1	Samhällsnytta	46
8	Framtida arbeten	47
8.1	Inköp av kretskort	47
8.2	Inköp av kapsling	47
8.3	Genomförande av tester för EMC	47
8.4	Vidareutveckling av GUI	47
9	Referenser	48

1 Inledning

I detta kapitel ges en kort presentation av företaget på vars uppdrag examensarbetet genomfördes. Här beskrivs ONE Demand som är produkten som ska undersökas och utvecklas. Syftet, målformuleringen och problemformuleringen redogörs i detta kapitel.

1.1 Bakgrund

Examensarbetet skrivs och utförs i uppdrag från ONE Nordic. Det är produkten ONE Demand som ska undersökas.

1.1.1 ONE Nordic AB

ONE Nordic AB, fortsättningsvis benämnt ONE, är ett ledande företag inom teknisk konsultering, entreprenader och service inom energisektorn. Företaget har ca 1100 anställda och omsätter ca 2,3 miljarder svenska kronor per år. Huvudkontoret ligger i Malmö. ONE erbjuder ett stort antal olika produkter och tjänster. En av produkterna som ONE erbjuder sina kunder är ONE Demand. [1]

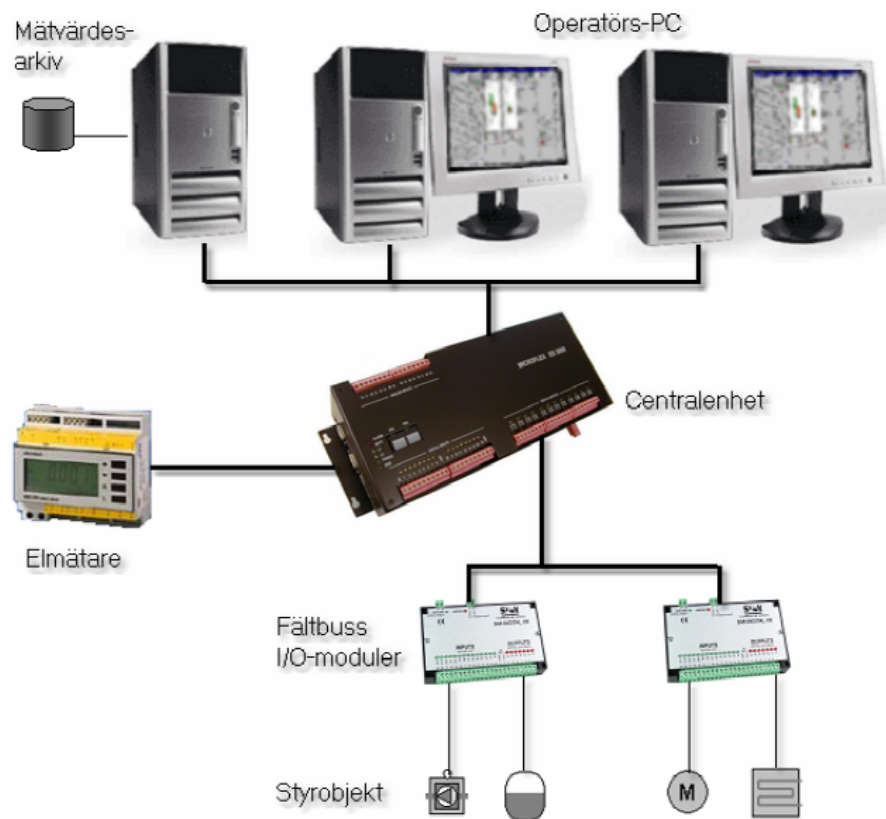
1.1.2 ONE Demand

ONE Demand utvecklades för ca 15 år sedan. Produkten utvecklades till ett helautomatiskt system som används för mätvärdesinsamling och effektstyrning. Systemet installeras ute på kundens anläggning i syfte att få in mätvärden från maskinerna, energimätare och andra sorters apparater i anläggningen. Dessa mätvärden kan användas i syfte att se om energiförbrukningen behöver styras ner. Energiförbrukningen behövs styras ner då den är på väg att överskrida den angivna tillåtna förbrukningen, styrgränsen. Det som installeras är en centralenhet, sensorer och fältbussar. Ett datorprogram installeras på kundens dator.

Centralenheten som installeras heter ED3000. Den är uppbyggt av diskreta komponenter, exempel på komponenter är en processor, RAM, ROM, både digitala och analoga in- och utgångar och lysdioder. Enheten är inte utvecklad för att ha något datorprogram installerat i sig men den kör alla funktioner som ställs in på ONE Demand.

Fältbussarna är kopplade mellan styrobjecten och centralenheten, styrobjecten utgör det kunden har angett som ska styras ner vid för hög förbrukning. Fältbussarna fungerar som en förlängning av centralenheten och utgörs av SI-OX, M-bus, modbus och dupline. Andra sorter av fältbussar kan förekomma.

Programmet som installeras på kundens dator heter ONE Demand, när programmet först utvecklades hette det EnergiDirigent. Detta program gör det



Figur 1: Översiktsbild för ONE Demand

möjligt för kunden att samla in mätvärden och styra effektuttaget. I programmet kan kunden eller anställda på ONE ange hur effektstyrningen ska ske. Kunden anger en styrgräns som anger hur mycket energiproduktionen får förbruka. Kunden kan se mätvärden i realtid men även mätvärden längre bak i tiden.[2]

Kommunikationen mellan centralenheten och datorn kan ske seriellt via RS232 eller TCP/IP via RJ45. Centralenheten kommunicerar med mätare eller fältbussar via framdragen kommunikationsslinga eller via TCP/IP nätverk. Mätvärdena kan sparas på centralenheten i upp till en månad men det finns även en möjlighet att ställa in så att mätvärdena lagras på en server, dator eller databas. Det är något som kunden själv bestämmer. Figur 1 visar en översiktsbild för hur en installation av ONE Demand kan se ut.

ONE Demand uppfattas som långsamt och föråldrat av anställda på ONE. De vill att systemets prestanda ska höjas samt att det ska vara möjligt att bygga ut centralenheten i framtiden.

1.2 Syfte

Syftet med detta examensarbete är att undersöka ett alternativ till centralenheten med prioritering på att det ska vara en enkortsdator som kan byggas ut för att uppfylla samma krav som den befintliga centralenheten.

1.3 Målformulering

Målet med detta examensarbete är att hitta en enkortsdator som kan ersätta den befintliga centralenheten med möjlighet till att programmet, ONE Demand, ska kunna installeras och köras på den. Efter att ha hittat lämplig enkortsdator med rätt expansionskort ska detta/dessa köpas in i syfte att kunna bygga en prototyp. Eftersom datorprogrammet är utvecklat specifikt för den befintliga centralenheten måste koden för funktionerna skrivas på så sätt att den är kompatibel med prototypen, en enkel GUI för detta kommer att skapas. Nuvarande programmeringsspråk är Java och Forth. Ett test på prototypen kommer att göras. Prototypen ska vara möjlig att bygga ut med expansionskort. Prototypen ska även uppfylla kraven för en CE-märkning.

Avvägningar mellan prestanda och kostnad ska undersökas innan valet av ny enkortsdator görs. En undersökning kommer göras för att se om en ny enkortsdator skulle vara bättre än den gamla centralenheten. Under arbetets gång ska en kravlista för den nya centralenheten tas fram.

Vid val av ny enkortsdator, som blir hjärtat i den nya centralenheten, skall även ett specifikationsdokument tas fram. Syftet med specifikationsdokumentet är att ONE ska förstå hur den nya centralenheten är uppbyggd och hur den fungerar.

1.4 Problemformulering

Examensarbetet ska svara på följande frågor:

1. Vilka krav måste prototypen uppfylla?
2. Vilka expansionsmöjligheter finns det för enkortsdatorer?
3. Hur ska koden skrivas för att bli kompatibel med prototypen?
4. Vad för test ska utföras på prototypen för att se att den fungerar?
5. Hur ska specifikationsdokumentet se ut?

1.5 Motivering av examensarbetet

Bland företagen som vi var i kontakt med kändes just examensarbetet hos ONE Nordic som mest intressant och även mest relevant för en elektroteknisk utbildning. ONE ser på detta examensarbete som en startpunkt för deras egna projekt, så som vi har förstått det så är detta något de velat börja med men inte haft resurserna till det än. ONE vill ha det presenterat så att någon annan ska kunna ta över efter oss. Ett antal av ONE:s kunder har frågat om framtida uppdateringar av ONE Demand, men det är inte något kunderna har begärt ska göras.

1.6 Avgränsningar

Examensarbetet är avgränsat gällande att en färdig enkortsdator med lämpliga expansionskort ska väljas och inte att vi själva ska bygga ett eget kretskort. Vi kommer bara att skriva kod som rör testet eftersom ONE Demand är ett stort och komplext datorprogram. Vi avgränsar oss från effektstyrning och fokuserar endast på mätvärdesinsamling från 4 st energimätare.

2 Teknisk bakgrund

I detta kapitel redogörs all teknisk bakgrund som använts för att nå ett resultat.

2.1 CPU

Processorn, även känd som CPU (central processing unit), utgör hjärnan i ett kretskort eller i en dator. Det är processorn som exekverar programmen som körs. Det är många faktorer som avgör hur snabbt en processor kan exekvera ett program. Komponenten exekverar allt i cykler i den hastighet som angetts av producenten. Denna hastighet anges i enheten hertz (Hz) som utgör en cykel per sekund och benämns som klockfrekvens. En processors snabbhet utgörs inte bara av klockfrekvens utan snabbheten beror också på databussens bredd, latens i minnet och cache strukturen. Det finns även processorer med flera kärnor. En processor med dubbla kärnor anses vara snabbare än en processor med en kärna, givet att de har samma klockfrekvens.[16]

En processor består av fyra delar; ALU, styrenhet och cacheminne. ALU (arithmetic logic unit) utför logiska operationer som AND, OR, XOR osv. Styrenheten styr alla av processorns operationer. Processorns viktigaste funktioner är fetch, decode och execute. Fetch utgör funktionen att ta till sig en instruktion från ett minne. Instruktionen som processorn nu tagit emot ska avkodas, det görs av funktionen decode. Instruktionen omvandlas till signaler. Efter dessa två funktioner ska instruktionen exekveras. Detta görs av funktionen execute. Beroende på hur processorns arkitektur är uppbyggt så sker detta på olika sätt, den kan utföras direkt eller utföras enligt ett visst antal steg.[16]

2.2 ED3000

Centralenheten som används för ONE Demand heter ED3000 och installeras ute hos kundens anläggning. Centralenheten används för mätvärdesinsamling samt effektstyrning. Via ett datorprogram som installeras hos kunden är det möjligt att ansluta sig till centralenheten. ED3000 har använts i femton år och är uppbyggt av diskreta komponenter. ED3000 har en processor från Motorola, modellen är MC68EN302PV20 20 MHz. Processorns klockfrekvens är 20 MHz. Den har RAM på 16 MB. Centralenheten är uppbyggt med 3 stycken COM portar för seriell uppkoppling mot fältbuss eller dator, Ethernet port för att anslutas till nätverket, 16 st digitala ingångar och 12 st digitala utgångar. Över varje in- och utgång finns det lysdioder som visar om in- eller utgången är aktivt. Figur 2 visar en utvald del av specifikationen för ED3000, se bilaga 1 för en fullständig specifikation för ED3000.[3]

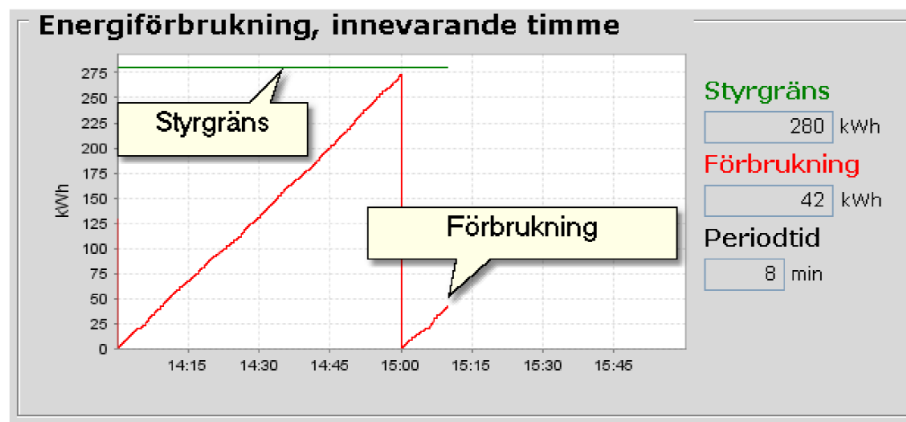
CPU	Motorola MC68EN302PV20 20 MHz
RAM	16 MB
COM -portar	3 st
Ethernet port	1 st
Digitala ingångar	16 st
Digitala utgångar	8 st
Livslängd	Minst 10 år vid normaldrift

Figur 2: ED3000 specifikation

2.3 ONE Demand

Datorprogrammet som installeras på kundens dator heter ONE Demand. Det är skrivet i Java och FORTH. Java är ett objektorienterat programmeringsspråk och FORTH är ett stackbaserat programmeringsspråk. GUI:n (graphical user interface) är programmerat i Java och alla funktionerna är programmerade i FORTH. Java utvecklades år 1995 James Gosling och företaget Sun Microsystems, företaget ägs nu av Oracle. FORTH utvecklades av Charles Moore år 1970. Programmet kräver att datorn på vilken den ska installeras ska ha en processor som har en klockfrekvens på minst 750 MHz och ha operativsystemet Windows, minst Windows 98.[5]

Programmet samlar in mätvärden från ONE:s kunders produktion. Det är främst värden som visar hur mycket energi en produktion förbrukar, men kan även vara värden som inte har något med själva produktionen att göra. Samplingen av dessa värden sker en gång per minut. Programmet kan med hjälp av dessa värden styra energiförbrukningen. Den gör det genom att tolka ett invärde och skriva ut det i en graf. I grafen visas en styrgräns upp som angetts av ONE:s kund. Styrgränsen anger hur mycket kunden vill att produktionen ska förbruka. Denna graf syns på programmets första sida. Bredvid grafen står styrgränsen angivet, den totala förbrukade energin under nuvarande period och hur långt in på perioden som systemet mätt, se figur 3. Effektstyrningen sker periodvis och normalt är perioden en timme, men det är kunden som bestämmer detta. Det är programmet som ser över förbrukningen. Skulle samlingen av värdena visa att förbrukningen kommer att överskrida styrgränsen så kommer den att styras ner av programmet. Skulle samlingen av värdena visa att förbrukningen inte kommer att överskrida styrgränsen så gör programmet inget. Vid nedstyrning har ONE:s kund angett vilka objekt som ska styras ner, kunden har även angett prioriteringar som bestämmer vilka objekt som främst ska styras ner och vilka objekt som inte får styras ner.



Figur 3: Förstasidan

Genom att gå in på sidan 'Kodsidor' kan en operatör se om lysdioderna på ED3000 är på, se figur 4, förutsatt att man har loggat in med rätt inloggning. Det är även på sidan 'Kodsidor' som objekt läggs till och funktioner kan implementeras. Programmets funktioner är uppbyggda av block. När en operatör med rätt behörighet vill implementera en ny funktion måste operatören gå offline först. Operatören skapar en ny kodsida, sedan får man välja ett block som redan är implementerat i programmet. Det finns ett stort antal block och med hjälp av dessa block kan man programmera räknare, pulsgenerator, med mera, men även vad som ska ske när ett invärde mottas. Varje kodsida rymmer åtta block, se figur 5. Figur 5 visar 3 block, de 2 blocken till vänster är en logisk operation. Det övre blocket är en AND-grind med en konstant etta i ingång i2 och den digitala utgången nr 6 i den andra ingången i2. När AND-grinden är aktiv etta ska den digitala ingången nr 4 på centralenheten vara igång, detta visas på blocket under AND-grinden. Blocket till höger om AND-grinden är en pulsgenerator där utgång 8 sätter igång generatören och utgången är utgång 7. Under rubriken 'Pulse shape' anges hur lång pulsen ska vara aktiv och perioden på pulsen. Block läggs till genom att trycka på knappen 'Välj objekt och gå till den sida där objektet finns'. Det finns flera objekt, några av dem är logiska grindar, pulsgenerator och räknare.

Stora Enso Grums - ED3000 - ED3000

Arkiv

Beskrivning

Kodsidor

ED 3000

☒ ED3000

SIOX
I/O ADAM
ELMÄTARE
EFFEKTSTYRNING
LOGIK
LOGG
LARM
STYRGRÄNS
RAPPORT

MF-DIDO
MF-DIDO

ED3000 I/O

Push buttons

p1 Push button 1
p2 Push button 2

Input signals

i1 Digital input 1
i2 Digital input 2
i3 Digital input 3
i4 Digital input 4
i5 Digital input 5
i6 Digital input 6
i7 Digital input 7
i8 Digital input 8
i9 Digital input 9
ia Digital input 10
ib Digital input 11
ic Digital input 12
id Digital input 13
ie Digital input 14
if Digital input 15
ig Digital input 16

Pulse counters

c1 0
c2 0
c3 0
c4 0
c5 0
c6 0
c7 0
c8 0
c9 0
ca 0
cb 0
cc 0
cd 0
ce 0
cf 0
cg 0

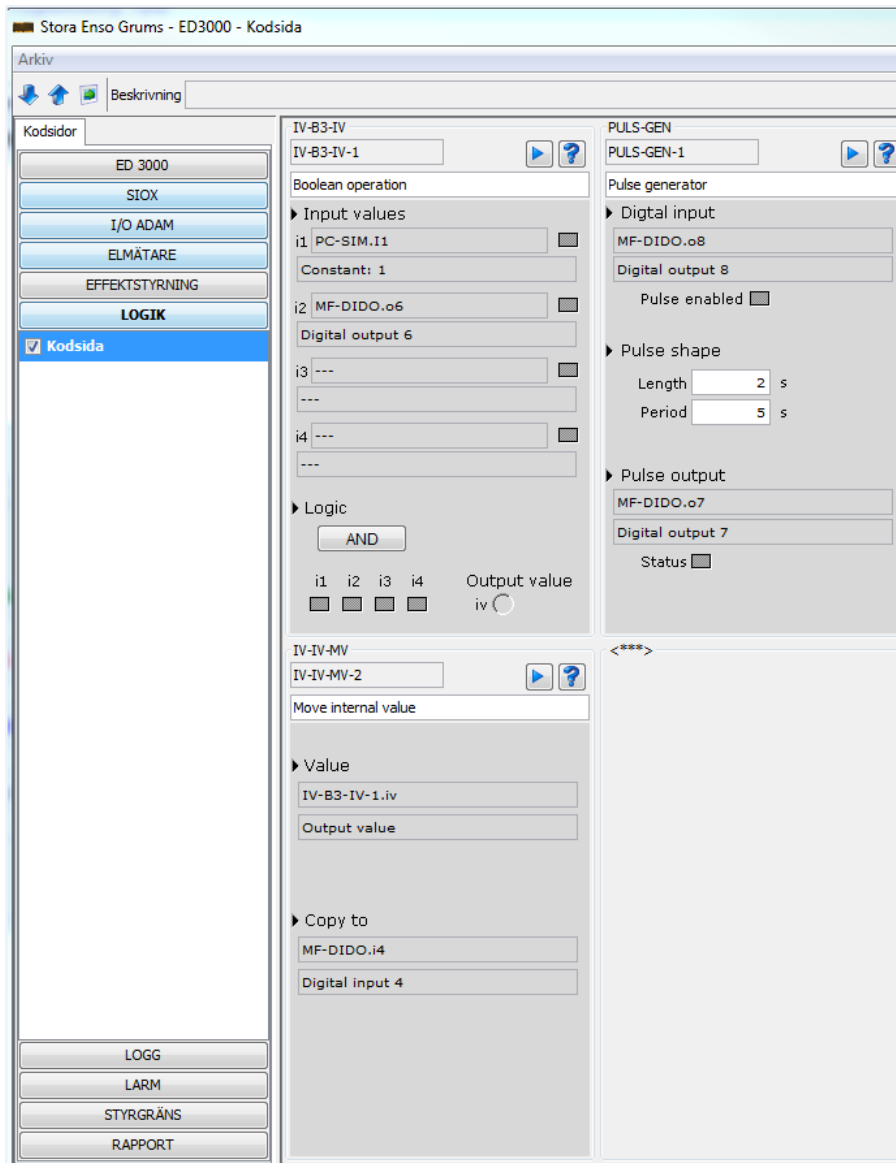
Debounce time

d1 0 ms
d2 0 ms
d3 0 ms
d4 0 ms
d5 0 ms
d6 0 ms
d7 0 ms
d8 0 ms
d9 0 ms
da 0 ms
db 0 ms
dc 0 ms
dd 0 ms
de 0 ms
df 0 ms
dg 0 ms

Output signals

o1 SYSTEM WATCHDOG
System watchdog ON
o2 EFFEKTSTYRNING WATCHDOG
Power control watchdog ON
o3 A-LARM
o4 B-LARM
o5 Digital output 5
o6 Digital output 6
o7 Digital output 7
o8 Digital output 8
o9 Digital output 9
oa Digital output 10
ob Digital output 11
oc Digital output 12

Figur 4: Gränssnitt för ED3000



Figur 5: Blockschema

2.4 Enkortsdator

En enkortsdator är en hel dator byggt på ett kretskort. Kretskortet innehåller en CPU, minne, in- och utgångar, grafikkort, allt som krävs för att utgöra en funktionell dator. På grund av att enkortsdatorer har lättillgängliga in- och utgångar är det enkelt att bygga ut ens enkortsdator med så kallade expansionskort, i syfte att kunna anpassa enkortsdatorn så att den uppfyller användarens behov. Marknaden för enkortsdatorer är stor, enkortsdatorer kan användas i hemmet men även i stora industrier. Det finns flera producenter av enkortsdatorer t.ex. Raspberry Pi Foundation, Firefly Team, Texas Instruments, SolidRun med flera. Enkortsdatorerna som beskrivs i avsnitten 2.4.1-2.4.4 kan användas till utbildning och egna projekt, till exempel fjärrstyrning av belysning i hemmet. Detta görs via användning och programmering av en enkortsdators in- och utgångar, så kallad GPIO(General-purpose input/output).

2.4.1 Firefly-RK3288

Firefly-RK3288 är en enkortsdator utvecklad av företaget Firefly. Den första hårdvaruversionen är "2014.08.09", efter feedback från användare kom den andra och slutgiltiga hårdvaruversionen "2014.09.30".[4] Det finns även en annan variant av enkortsdatorn, så kallad Firefly-RK3288-Reload. Kärnan utgörs av en processor från ARM och har en klockfrekvens på 1,8 GHz. Kortet har 2GB RAM-minne och kan köras med operativsystemet Ubuntu. Firefly-RK3288 har en ethernet port, 3 USB uttag, 6 st GPIO och den har ett inbyggt chip som gör det möjligt att koppla upp sig till WiFi.[6] Programmering av enkortsdatorn görs med till exempel C eller Python.

2.4.2 Raspberry Pi 3 model B

Raspberry Pi 3 model B(RPi 3) som lanserades februari 2017 är efterträdaren till Raspberry Pi 2 och har en processor från företaget ARM med en klockfrekvens på 1,2 GHz. På kortet finns det en Ethernet port, 4 USB uttag, 26 st General purpose input output (GPIO) och 1 GB RAM. RPi 3 är kompatibelt med flera operativsystem som Raspbian, NOOBS, Ubuntu och Windows 10 IoT core. RPi har även inbyggda chip som gör det möjligt att koppla upp sig till WiFi och Bluetooth. [7] Programmering av en RPi kan göras med bland annat programmeringsspråken Java, C/C++ eller Python.

2.4.3 BeagleBoard

BeagleBoard, version 3, är utvecklad av Texas Instruments och har en processor med klockfrekvens 1 GHz. Enkortsdatorn lanserades april 2013. Kortet har 512 MB RAM och är kompatibelt med operativsystem Ubuntu och Debian. BeagleBoard har 2 USB uttag, 26 st GPIO och kan också koppla upp sig till WiFi.[8]

Programmering av en BeagleBoard kan göras med ett programmeringsspråk, till exempel Java, C/C++ eller Python.

2.4.4 HummingBoard Base

Företaget Solid Run har utvecklat en enkortsdator vid namn HummingBoard Base. Den lanserades i juni 2014. Enkortsdatorn lanserades i fyra olika versioner, Base, Pro, Gate och Edge. Processorn har en klockfrekvens på 1GHz och kortet har 2 USB uttag. Kortet har 26 st GPIO och en Ethernet port.[9] HummingBoard använder sig av operativsystemet Linux. Programmering av en Hummingboard kan göras med ett programmeringsspråk, till exempel C eller Python.

2.4.5 Jämförelse

Enkortsdatorerna som det skrivs om i föregående kapitel är väldigt lika, de kan alla kopplas upp till en egen skärm som en vanlig PC och måste alla kopplas till en strömadapter. Det finns små skillnader bland dessa enkortsdatorer och de visas upp i figur 6. Det finns även skillnader gällande mjukvaran i dessa enkortsdatorer och detta kommer vara av stor betydelse vid val av enkortsdator. Innan beslutet tas om vilken enkortsdator som väljs ut ska även en undersökning göras om vilka expansionsmöjligheter som finns.

Företag	Modell	Antal kärnor i CPU:n	Klockfrekvens på CPU (GHz)	RAM	Pris (kr)
Firefly Team	Firefly-RK3288	4	1,8	2 GB	Ca 1290
Raspberry Pi Foundation	Raspberry Pi 3 model B	4	1,2	1 GB	Ca 400
Texas Instruments	BeagleBoard	1	1	512 MB	Ca 700
Solid Run	HummingBoard Base	2	1	1GB	Ca 740

Figur 6: Enkortsdatorer

2.5 Programmeringsspråk

I detta kapitel kommer det att skrivas om programmeringsspråk som används i ONE Demand, men även om andra programmeringsspråk som kan byta ut det språk som används i nuläget för ONE Demand.

2.5.1 Forth

Funktionerna i ONE Demand är skrivna i Forth. Forth är ett imperativt, stackorienterat och procedurellt språk utan ett typsystem. Språket kan användas både som ett högnivå- och som ett lågnivåspråk. Forth används huvudsakligen inom inbyggda system och i firmware. Utvecklingsmiljöer som kan användas för Forth är till exempel Win32Forth, GForth och SwifhForth. Språket utvecklades av Charles Moore år 1970. [10] Se Bilaga 2 för exempel på Forthkod.

2.5.2 Java

Java är ett strukturerat och objektorienterat programmeringsspråk som använder, bland annat, statisk och stark typning. Språket skapades av James Gosling och utvecklades av Sun Microsystems (numera ägd av Oracle Corporation). Java används inom programvaruutveckling för Android och inom utveckling av webbapplikationer. Populära utvecklingsmiljöer för Java är bland annat NetBeans, Eclipse och Android Studio. JavaFX är en mängd paket som innehåller flera bibliotek som gör det möjligt för en utvecklare att implementera grafer, mediaspelare och användargränssnitt.[13] Se Bilaga 3 för kodexempel på Java med JavaFX.

2.5.3 Python

Python är ett högnivåspråk med stöd för flera olika programmeringsparadigm, till exempel objektorienterad och funktionell programmering. Språket är en öppen programvara som utvecklades av den ideella föreningen Python Software Foundation. Python har enkla syntaxer och stöder moduler och paket, den använder dynamisk och stark typning. Det som gör Python-kod enkel att läsa är att den varken använder semikolon för att avsluta rader eller klamrar för att visa var ett kod-block börjar och slutar, så som i till exempel Java. Den använder sig istället av indragningar för att visa var ett kod-block börjar och slutar. Python kan användas för en mängd olika saker, till exempel för programvaru- eller webbutveckling. Bland populära utvecklingsmiljöer för Python finns bland annat PyDev, PyCharm och Wing IDE. Många bibliotek finns tillgängliga för Python. Det finns tre olika paket som kan användas för att skapa ett grafiskt användargränssnitt. Tkinter är standard paketet för GUI-utveckling med Python och kan implementera enkla funktioner som knappar, textrutor och eti-

kett.[14][17] De andra två paket använder sig av Qt. Se Bilaga 4 för kodexempel på Python med Tkinter.

Qt är ett applikationsramverk som kan användas för utveckling av grafiska program. Den utvecklas av Qt Project och finns för en mängd olika programmeringsspråk. PyQt och Pyside är två olika Python-bibliotek som kan användas för att binda samman programmeringsspråket Python med applikationsramverket Qt för utveckling av grafiska program. PyQt är utvecklad av Riverbank Computing Limited medans PySide hanteras av en grupp från själva Qt Project. PyQt stödjer både Qt 4 och Qt 5, till skillnad från PySide som för tillfället endast stödjer Qt 4. PySide för Qt 5 är under utveckling. Eftersom att både PyQt och PySide använder Qt är skillnaderna i hur man skriver kod med dem minimala. För att använda PyQt eller PySide krävs det att dessa paket installeras på datorn, till skillnad från Tkinter som kommer förinstallerat med Python. [18][19] Se Bilaga 5 för kodexempel på PyQt respektive PySide med Python.

2.6 Specifikationsdokument

Det finns olika sorters specifikationsdokument, det finns dokument som beskriver en hårdvara och det finns dokument som beskriver mjukvara. Specifikationsdokument som beskriver hårdvara listar hårdvarans attribut. Det kan vara hur mycket RAM minne en hårdvara har, vilken sorts processor den har och hur många in-och utgångar hårdvaran besitter. Ett specifikationsdokument som beskriver mjukvara innehåller information som vilket programmeringsspråk som använts, hur funktioner fungerar och hur man navigerar sig genom programmet.

2.7 GPL och LGPL

GPL(GNU General Public License) och LGPL(GNU Lesser General Public License) är två licenstyper för programvara. De båda licenserna indikerar att programvaran är fri för användning. Programvara som är under LGPL eller GPL licens får användas avgiftsfritt. Skillnaderna mellan LGPL och GPL är att vid användning av en programvara under GPL licensen finns krav på att källkoden för den produkt som användaren har utvecklat med hjälp av programvaran måste finnas tillgänglig, programvara under LGPL licensen har inte liknande krav. [20][21]

2.8 CE-märkning

En tillverkare försäkrar konsumenten om att en produkt uppfyller alla EU-direktiv genom att märka produkten med CE, se figur 7. CE-märkningen gäller i EES-området, det vill säga EU, Island, Lichtenstein och Norge. Produkter som säljs i ESS-området måste ha en CE-märkning även om de blivit producerade utanför ESS-området. Det är tillverkaren som ansvarar för att produkten uppfyller kraven för märkningen. [11]

En produkt CE-märks efter ett visst antal steg. Det första steget är att se vilka krav som gäller för just den produkten som produceras. Det finns olika krav för olika typer av produkter. Det finns ett antal tester som måste utföras på produkten. Dessa tester utförs av tillverkaren men i vissa fall måste produkten också testas av särskilda organ. Efter testerna tas en teknisk dokumentation fram som visar att de tekniska kraven för produkten är uppfyllda. Märket kan nu fästas på produkten. Märket ska vara synligt, lättläsligt och inte möjligt att suddas bort, om produkten testats av ett organ så ska organets ID-nummer också finnas på produkten.



Figur 7: CE-märket [22]

2.9 EMC-direktivet

EMC-direktivet är ett direktiv som EU tagit fram gällande elektromagnetiska kompatibiliteten. Syftet med direktivet är att alla elektriska anordningar ska uppfylla krav som gör så att de inte påverkar andra elektriska anordningar. Direktivet gör så att produktens elektromagnetiska strålning begränsas så att den inte stör radio, telekommunikation eller andra anordningar. Direktivet gör även så att produkten inte låter sig störas av andra produkter. [12]

2.10 S0-gränssnitt

S0-gränssnittet hänvisar till Europastandarden EN 62053-31. Standarden gäller för passiva, tvåtråds, externt drivna pulsgeneratorer som används i energimätare. Standarden fastställer två klasser, klass A och klass B. Klass A är för pulser som sänds över långa avstånd och klass B är för korta avstånd, avstånden är inte definierade i standarden. Tabell 1 visar hur stor spänning energimätaren kan matas med, den visar även hur pulserna ska tolkas. Pulsen tolkas vara aktivt hög då strömmen är större än minimum gränsen och är låg då strömmen är under max gränsen för OFF-state. Enligt standarden ska pulslängden vara $t_{on} \geq 30ms$ och tiden mellan punkterna vara $t_{off} \geq 30ms$. Pulsspänningen beror på matningens spänning.[15]

Tabell 1: Klass A och Klass B

Parametrar	Klass A	Klass B
Max volt (V dc)	27	15
Max ström ON-state (mA)	27	15
Min ström ON-state (mA)	10	2
Max ström OFF-state (mA)	2	0,15

2.11 IEC 60529

Standarden IEC 60529 anger graden av skydd mot ingång av solida objekt och vätskor genom kapslingar. Skyddet anges med en IP kod som består av 2 siffror. Första siffran i IP koden anger skyddet mot intrång av solida objekt där 0 är inget skydd alls och 6 är skydd mot all sorts intrång. Den andra siffran i IP koden anger skydd mot intrång av vätskört, där siffran 0 är inget skydd alls och siffran 8 anger att kapslingen är vattentät.[23]

3 Metod

I det här kapitlet redogörs vilka steg som togs för att få fram ett resultat. Arbetsflödet kan ses på figur 8. Rutorna i figur 8 visar faserna som arbetet var uppdelat i. Författarna till examensarbetet har jobbat nära varandra under varje fas för att kunna diskutera snabbt och enkelt om varje beslut. Det har fungerat väldigt bra.

3.1 Fas: Förstudie

Det första som gjordes var att läsa om och förstå själva produkten ONE Demand. Flera manualer om produkten lästes, både om mjukvaran och hårdvaran. Manualerna lästes för att få en förståelse om vad systemet gör men framförallt för att redogöra vad som krävs av hårdvaran. Dessa manualer visade varför hårdvaran uppfattas som långsam, mer om detta i avsnitt 6. Manualerna gav även en klar uppfattning om vad för krav det ska vara på en enkortsdator för att kunna utföra samma funktioner som ED3000.

3.2 Fas: Undersökning

En undersökning gjordes för att få en djup insyn över vad för enkortsdatorer som fanns tillgängliga på marknaden. Undersökningen fokuserade på enkortsdatorer från ledande företag inom branschen. Det gjordes även en undersökning kring vilka programmeringsspråk som var bäst lämpade för respektive enkortsdator.

3.3 Fas: Utformning av kravlista

Efter faserna Förstudie och Undersökning fanns det nu tillräckligt med kunskap för att skapa en kravlista för enkortsdatoren. Kravlistan togs fram i samarbete med anställda på ONE. Kravlistan behövdes till nästa fas som är 'Val av enkortsdator'.

3.4 Fas: Val av enkortsdator

En enkortsdator köptes in efter att kravlistan utformats. Enkortsdatoren som valdes uppfyllde kraven som listats upp i kravlistan, fler anledningar till varför just denna enkortsdator valdes står skriver i avsnitt 4.2. Enkortsdatoren köptes från en butik som låg lokalt. Efter att ha köpt in enkortsdatoren börjades utformningen av Test 1.

3.5 Fas: Test 1

Efter att enkortsdatorn köptes in gjordes Test 1. Det är ett test som visar om enkortsdatorn kan läsa in värden från en energimätare. Mer om Test 1 står skrivet i avsnitt 5.1. Efter att Test 1 utfördes påbörjades fas 'Val av programmeringsspråk'.

3.6 Fas: Val av programmeringsspråk

Val av programmeringsspråket baserar sig på resultaten från fas 'Undersökning'. Ett programmeringsspråk valdes som var kompatibelt med enkortsdatorn som valdes i fas 'Val av enkortsdator'. Efter att programmeringsspråket valts ut kunde nu en GUI programmeras.

3.7 Fas: Programmering av GUI

När programmeringsspråket valdes ut programmerades en GUI. GUI:n programmerades på en annan dator än enkortsdatorn och fördes sedan över till enkortsdatorn. Detta eftersom att det var enklare att programmera på en bärbar dator än på enkortsdatorn. Utseendet på GUI:n bestämdes tillsammans med anställda på ONE. GUI:n som programmerades användes i Test 2.

3.8 Fas: Test 2

Test 2 är ett test som visar hur GUI:n som programmerades i föregående fas fungerar. Testet höll på under en hel natt. Mer om Test 2 är skrivet i avsnitt 5.2

3.9 Fas: Test 3

Test 3 utfördes efter Test 2. Test 3 är ett test som visar hur enkortsdatorn reagerar vid plötsligt strömavbrott. Mer om Test 3 skrivs i avsnitt 5.3.

3.10 Fas: Vad för tester ska utföras för CE-märkning?

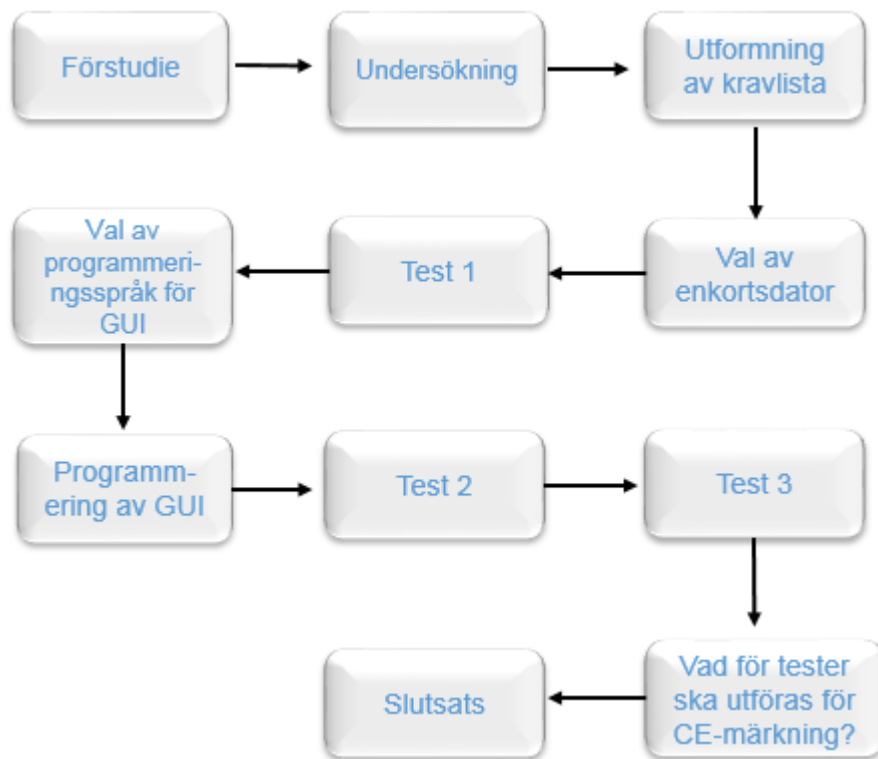
Efter att ha utfört alla tester gjordes en lista över vilka tester som skulle utföras på produkten så att den skulle godkännas för en CE-märkning. Inga tester för CE-märkning utfördes eftersom att produkten inte är klar än. Mer om testerna för CE-märkning är skrivet i avsnitt 6.12.

3.11 Fas: Slutsats

Den sista fasen är slutsatsen. Slutsatsen gjordes med hjälp av alla de föregående faserna. All undersökning sammanställdes i syfte att kunna få fram en slutsats. Slutsatsen är skriven i avsnitt 7.

3.12 Källkritik

Källorna till detta examensarbete har bestått av manualer och internet. Informationen om ONE Demand har inhämtats från manualer men även från anställda på ONE som förklarat systemet. Källorna [2][3][5] är manualer som ONE sammanställt och anses vara pålitliga eftersom det är ONE som skrivit dem om sin egen produkt. Källan [1] är tagen från ONE:s officiella hemsida och den anses också vara pålitlig för att de skriver om sig själva. Källorna [4], [6]-[10], [13][14] och [17]-[19] är tagna från företags hemsida som ansvarar för sina egna produkter. Källan [16] är tagen etablerad hemsida som skriver om tekniska prylar. Källorna [11][12] är från Europeiska Unionens officiella hemsida och [22] är en bild som är tagen från samma hemsida. [15][23] är en europeiska standarder som köpts in vilket gör de pålitliga. Källorna [20][21] är tagna från de som har utformat licenserna GNU GPL och LGPL och är på så sätt pålitliga.



Figur 8: Flödesschema

4 Analys

I det här kapitlet skrivs det om hur resonemanget kring examensarbetet gick till.

4.1 Utformning av kravlista

Först gjordes det en undersökning om ED3000 i syfte om att förstå vad det är den gör. Enkortsdatoren som valdes ska uppfylla samma funktionalitet som ED3000 gör. Innan valet av enkortsdator gjordes så undersöktes det först vilka krav som den skulle uppfylla. När kraven togs fram så togs hänsyn till det faktum att programmet ska vara installerat på enkortsdatoren, så vissa moduler som ED3000 har är inte nödvändiga på enkortsdatoren. ED3000 behöver en COM port i syfte att kunna kommunicera med en dator seriellt, en enkortsdator behöver inte detta. Enkortsdatoren som väljs behöver ha en processor som har en så hög klockfrekvens som möjligt, men hur hög den bör vara är inte definierat. Det finns inte heller något som säger hur mycket minne enkortsdatoren bör ha, det ska vara tillräckligt för att kunna spara flera filer. Se avsnitt 6.2 för kravlistan.

4.2 Val av enkortsdator

När valet av enkortsdatoren gjordes så sammanställdes först all undersökning som gjordes om diverse enkortsdatorer och de jämfördes mot varandra. Enkortsdatoren som valdes var tvungen att uppfylla alla krav som sammanställts i kravlistan. Kravlistan hittas i avsnitt 6.2. Förutom att enkortsdatoren ska uppfylla alla krav i kravlistan ska den även vara lättillgänglig för inköp, d.v.s. att den ska kunna köpas in från en lokal butik. Detta eftersom att det inte ska vara nödvändigt att vänta på långa leveranstider. Det ska inte heller finnas någon kostnad för licens på enkortsdatoren, eftersom ONE vill slippa betala någon avgift. Det ska även vara lätt att hitta fakta om enkortsdatoren, hur man använder och programmerar den samt ett stort internetforum. Detta för att författarna till examensarbetet och anställda på ONE inte har någon tidigare erfarenhet av enkortsdatorer.

Efter en jämförelse av olika enkortsdatorer, i form av en tabell (se figur 6), valdes enkortsdatoren Raspberry Pi 3. Figur 6 visar att RPi 3 är billigare än alla andra, figuren visar att RPi:s processor inte har högst klockfrekvens, men den är med sin klockfrekvens på 1,2 GHz fortfarande mycket högre än ED3000:s klockfrekvens på 20 MHz. Det lades inte så stor fokus på klockfrekvensen eftersom att alla processorer i undersökningen hade mycket högre klockfrekvens än den som är på ED3000. Det som verkligen talade för RPi 3 var att det finns många expansionsmöjligheter för den, det finns ett stort internetforum där andra RPi-användare gärna hjälps åt och att av alla enkortsdatorer som var med i undersökningen så var det bara RPi 3 som fanns tillgänglig i en lokal butik.

Mer om RPi 3 står skriver i avsnitt 6.3 och expansionsmöjligheterna för en RPi 3 är skrivet i avsnitt 6.10.

4.3 Val av programmeringsspråk

Vid val av vilket programmeringsspråk som skulle användas så gjordes det först en undersökning kring vilka programmeringsspråk som är kompatibla med enkortsdatorn, samt vilka programmeringsspråk som rekommenderas för enkortsdatorn. Detta gjordes eftersom att vissa enkortsdatorer inte är kompatibla med en del programmeringsspråk. För att välja ett lämpligt programmeringsspråk gjordes även en undersökning om vilka applikationsramverk för GUI-utveckling som finns tillgängliga för respektive programmeringsspråk. Efter en jämförelse av flera programmeringsspråk stod valet mellan Java och Python, detta på grund av att de båda kan användas för att skapa en GUI som uppfyller kraven (se avsnitt 6.5). Python valdes över Java på grund av att den har bättre plattformar för GUI-utveckling för en Raspberry Pi. För Python var det tre olika Python-paket som tittades på för möjligheten att skapa en GUI, dessa är Tkinter, PyQt och PySide. Vid jämförelse av de tre olika paketen var det i slutändan Tkinter som valdes. Alla tre kan uppfylla kraven, men med PyQt tillkommer i det här fallet en kostnad. En utförlig motivering till varför Python med Tkinter valdes över Java är skrivet i avsnitt 6.6.

5 Test

I det här kapitlet kommer det skrivas om vad för tester som utfördes. Det utfördes 3 stycken tester. Det första testet gjordes för att se om RPi 3 kan ta emot pulser från en energimätare. Detta är själva poängen med examensarbetet, att en enkortsdator ska kunna ta emot pulser från en energimätare. Efter första testet utfördes Test 2. Detta är ett test där pulserna som mottas från en energimätare kan tolkas och visas upp i en GUI. Detta är viktigt för att ONE:s kunder ska kunna se förbrukningen i en GUI och i en extern fil. Det sista testet är ett strömavbrotts-test. Eftersom RPi 3 ska köras utan uppehåll är det viktigt att veta hur den reagerar på plötsligt strömavbrott.

5.1 Test 1

Första testet som gjordes med Raspberry Pi 3 benämns som test 1. Material som användes var:

- Raspberry Pi 3
- Energimätare DIZ Tillquist
- Kopplingsdäck
- Sladdar
- Resistorer
- Oscilloskop

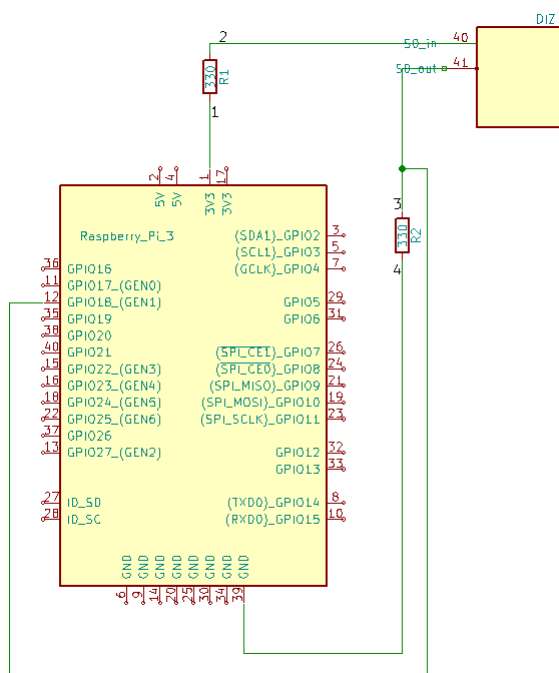
Syfte

Syftet med detta test var att koppla ihop RPi 3 med en energimätare och räkna pulser, detta för att se om RPi 3 kan utföra samma funktioner som ED3000.

Tillvägagångssätt

Det första som gjordes var ett kretsschema för hur allt skulle kopplas ihop, se figur 9. Eftersom att pulsen måste ha en ström på minst 2mA så valdes en resistor med hjälp av Ohms lag. RPi 3 och energimätaren kopplades upp enligt figur 9 på ett kopplingsdäck. Energimätaren som används är från Tellquist och är av modell DIZ, mätaren har in- och utgång som följer S0-gränssnittet. Spänningen över R1 kunde mätas upp men den beräknades först med spänningsdelningsformeln. Spänningen över R1 mättes upp med multimeter mellan punkterna 1 och 2 i kretsschemat, spänningen över R2 mättes upp med multimeter över punkterna 3 och 4 i kretsschemat. Ett program kodades med språket Python och kördes. Resultatet skrevs ut på RPi 3s terminal. Programmet var skrivet så att varje

puls skulle tas emot och skalas eftersom en puls på energimätaren utgör 0,001 kWh. Det som skrevs ut var pulsräkningen och värdet efter skalning. Resultatet för test 1 beskrivs i avsnitt 6.4.



Figur 9: Kretsschema för test 1 och test 2

5.2 Test 2

Test 2 utfördes efter test 1. Detta är ett test där pulserna som läses in från energimätaren visas upp i en GUI.

Syfte

Syftet med test 2 var att se om alla värden från energimätaren kan presenteras i en GUI

Tillvägagångssätt

Det första som gjordes var ett kretschema för hur allt skulle kopplas in. Figur 9 visar kretsschemat. Material som användes var:

- Raspberry Pi 3
- Energimätare som följer S0-gränssnittet
- Kopplingsdäck
- Sladdar
- Resistorer

Ett programmeringsspråk för GUI valdes. En GUI kodades där förbrukningen visades i kWh och det visades upp i en graf. Skalningen för alla pulser från energimätaren ställdes in i programmet. Programmet kördes över natten. Resultat och slutsats för test 2 skrivs om i avsnitt 6.7.

5.3 Test 3

Test 3 är ett test som utförs för att se vad som händer med RPi 3 vid plötsligt strömavbrott.

Syfte

Syftet med test 3 var att se hur RPi 3 reagerar vid plötsligt strömavbrott.

Tillvägagångssätt

Ett plötsligt strömavbrott simuleras enklast genom att dra ut strömadaptern från vägguttaget. Detta gjordes och efter 1 minut kopplades den in igen. Resultat för test 3 är skrivet i avsnitt 6.8.

6 Resultat

I det här kapitlet presenteras alla resultat. Resultaten är i kronologisk ordning.

6.1 Varför uppfattas ED3000 som långsam?

Efter en undersökning av hur ED3000 är uppbyggt så undersöktes processorn. Det är en processor från Motorola med klockfrekvens på 20 MHz. Detta är en relativt låg klockfrekvens då de flesta av dagens processorer ligger på över 1 GHz. Processorn har dessutom bara en kärna. Detta kan vara en av anledningarna till varför ED3000 uppfattas som långsam. Eftersom det är ett stort program som körs så krävs det en processor med högre klockfrekvens.

6.2 Vilka krav måste enkortsdatorn uppfylla?

Innan en enkortsdator valdes var det nödvändigt att kartlägga vilka funktioner som är essentiella för att kunna utföra samma saker som ED3000 gör. Förutom att kartlägga vilka funktioner ED3000 har, var det också tvunget att se till så att enkortsdatorn kan ha datorprogram installerat i sig och ha möjligheter till att bygga ut enkortsdatorn. Kraven som tagits fram är för en enkortsdator som ska kunna ersätta ED3000. Kraven på enkortsdatorn är:

- Minst 16 st digitala ingångar.
- Minst 8 digitala utgångar.
- Ska kunna köras utan uppehåll i flera år.
- Ska kunna ha ett datorprogram installerat i sig.
- Kunna koppla upp sig mot internet.
- Måste ha möjlighet för seriell kommunikation med fältbuss.
- Ska kunna lagra värden i sig och i en extern databas.
- Enheten måste kunna ha en statisk IP-adress.
- Det ska vara möjligt att komma åt enkortsdatorn från en annan dator, fjärrstyrning.
- Enheten ska kunna återhämta sig vid plötsligt strömbrott. Programmet ska startas av sig själv och data får inte försvinna vid strömbrott.

6.3 Raspberry Pi 3 model B

Efter en noggrann undersökning var valet kring vilken enkortsdator som skulle väljas ganska enkel. Valet föll på RPi 3 eftersom den uppfyller alla krav som listades på kravlistan. RPi 3 valdes inte endast för att den uppfyller kraven utan även för att Raspberry Pi Foundation inte kräver någon avgift för licens, mot förmodan att man inte säljer produkten som en RPi 3. RPi 3 valdes dessutom för att vid jämförelsen av enkortsdatorerna, se figur 6, var den billigast och processorn hade relativt hög klockfrekvens. Fokus las inte på klockfrekvensen eftersom alla processorer i undersökningen hade högre klockfrekvens än den på ED3000. Förutom det faktum att RPi 3 är billigare än andra enkortsdatorer så är den även lättillgänglig, de flesta lokala teknikbutikerna säljer RPi 3, vid val av en annan enkortsdator var man tvungen att beställa och betala frakt samt vänta på leverans. En annan anledning till att RPi 3 valdes var att det finns många expansionskort tillgängliga för den. Mer om expansionskort är skrivet i avsnitt 6.10.

Raspberry Pi 3 model B är den tredje generationen i enkortsdatorerna Raspberry Pi. RPi 3 har följande hårdvaruspecifikation:[7]

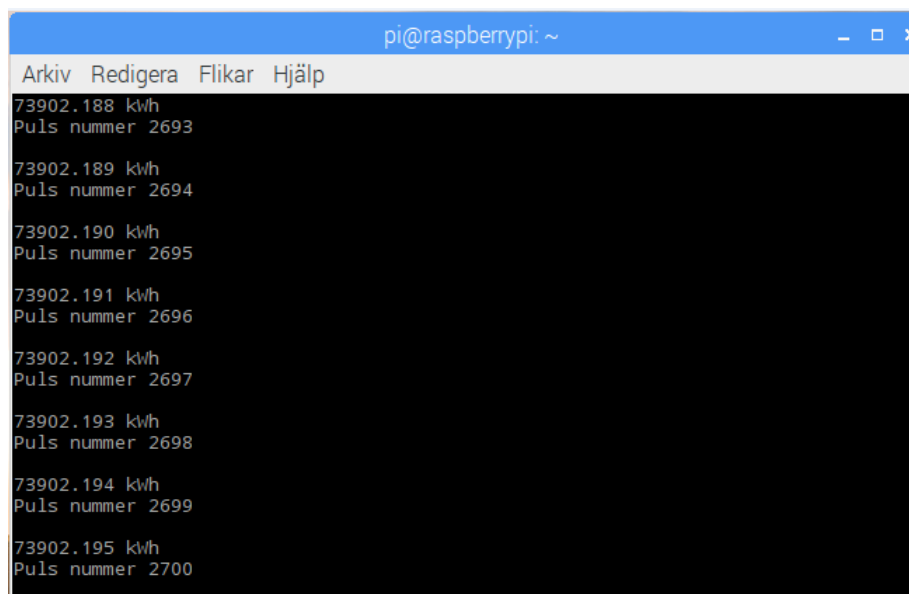
- 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- 1GB RAM
- 4 USB portar
- HDMI port
- Ethernet port
- 3.5mm ljuduttag
- Plats för microSD
- VideoCore IV 3D GPU

6.4 Test 1 resultat

Test 1 skulle visa att det går att räkna pulser från en energimätare med hjälp av RPi 3. Programmeringsspråket som användes var Python eftersom det har ett eget bibliotek som gör det enklare att använda sig av RPi 3 GPIO. Testet kördes i 3 timmar för att se om terminalen skulle visa samma värde som energimätaren, vilken den gjorde. Figur 10 visar värdena som terminalen visar. Tabell 2 visar att terminalen visar samma värde som energimätaren.

Tabell 2: Visar timvis under 3 timmar pulstalet, värdet på terminalen och värdet på energimätaren

Efter x antal timmar (h)	Pulstal	Värde på terminal (kWh)	Värde på DIZ (kWh)
0	0	73899,495	73899,495
1	900	73900,395	73900,395
2	1800	73901,295	73901,295
3	2700	73902,195	73902,195



```
pi@raspberrypi: ~
Arkiv Redigera Flikar Hjälp
73902.188 kWh
Puls nummer 2693
73902.189 kWh
Puls nummer 2694
73902.190 kWh
Puls nummer 2695
73902.191 kWh
Puls nummer 2696
73902.192 kWh
Puls nummer 2697
73902.193 kWh
Puls nummer 2698
73902.194 kWh
Puls nummer 2699
73902.195 kWh
Puls nummer 2700
```

Figur 10: Visar värdet för test 1

6.5 Vilka krav ska en GUI uppfylla?

Vid skapandet av en GUI var det nödvändigt att ta hänsyn till följande krav:

- Möjlighet att lägga till fyra mätare
- Ska kunna visa upp mätdata för varje mätare, både i form av pulser och om möjligt i en graf
- Grafen för en mätare ska visa data för pågående timme
- Det ska vara möjligt att spara mätvärden externt, i fil eller som bild(skärmdump)

6.6 Val av programmeringsspråk för GUI

En GUI programmerades i syfte att visa upp förbrukningen för användaren. Först skulle ett programmeringsspråk väljas ut för användning. Raspberry Pi Foundation har inget officiellt programmeringsspråk men rekommenderar användning av Python. En RPi 3 är förinstallerad med både version 2 och 3 av Python. Det tillkommer även ett Python-bibliotek, RPi.GPIO, som kan användas för att styra in- och utgångarna. JavaFX är mycket utvecklat men när det används på ARM-baserade enheter måste en äldre version användas, eftersom att när JDK 8u33 för ARM släpptes ut bestämde Oracle sig för att inte längre uppdatera JavaFX för ARM-baserade enheter. Det är även möjligt att använda sig av Java för GUI-utveckling tillsammans med Pi4J, ett bibliotek för Java, för att styra in- och utgångar på en RPi. Programmeringsspråket som valdes var Python, inte bara för att Raspberry Pi Foundation rekommenderar det men även för att det har ett stort internetforum.

Vid val av en plattform för GUI-utveckling med Python fanns det tre olika möjligheter att välja mellan. Den första är Tkinter som är standardbiblioteket för GUI-utveckling med Python. Tkinter har en väldigt enkel design och ett fåtal funktioner. Det andra och tredje valet är PyQt respektive PySide. Både PyQt och PySide använder applikationsramverket Qt och har, i jämförelse med Tkinter, fler funktioner för att skapa och designa en GUI. PySide har för tillfället endast stöd för Qt-version 4 medan PyQt stödjer både version 4 och 5 av Qt. Nackdelen med PyQt är att den går under en GPL-licens vilket medför att det krävs en kommersiell-licens vid försäljning av proprietär programvara som utvecklats med PyQt. PySide, däremot, är under en LGPL-licens och kan användas fritt för utveckling av proprietär programvara.

Trots fördelarna med Qt valdes Tkinter. En RPi är skapad med fokus på enkelhet vilket innebär att det utseendemässigt finns få skillnader mellan Qt och Tkinter. De stora skillnaderna är mellan funktionerna men det finns många andra bibliotek som kan importeras för att använda funktioner som inte kommer med Tkinter. GUI:n är endast till för att utan avbrott kunna visualisera mätvärden från dem olika mätarna och spara dessa värden. På grund av detta

finns det i det här sammanhanget ingen anledning till att använda sig av Qt, andra bibliotek ger den funktionaliteten som behövs.

6.6.1 Utvecklad GUI

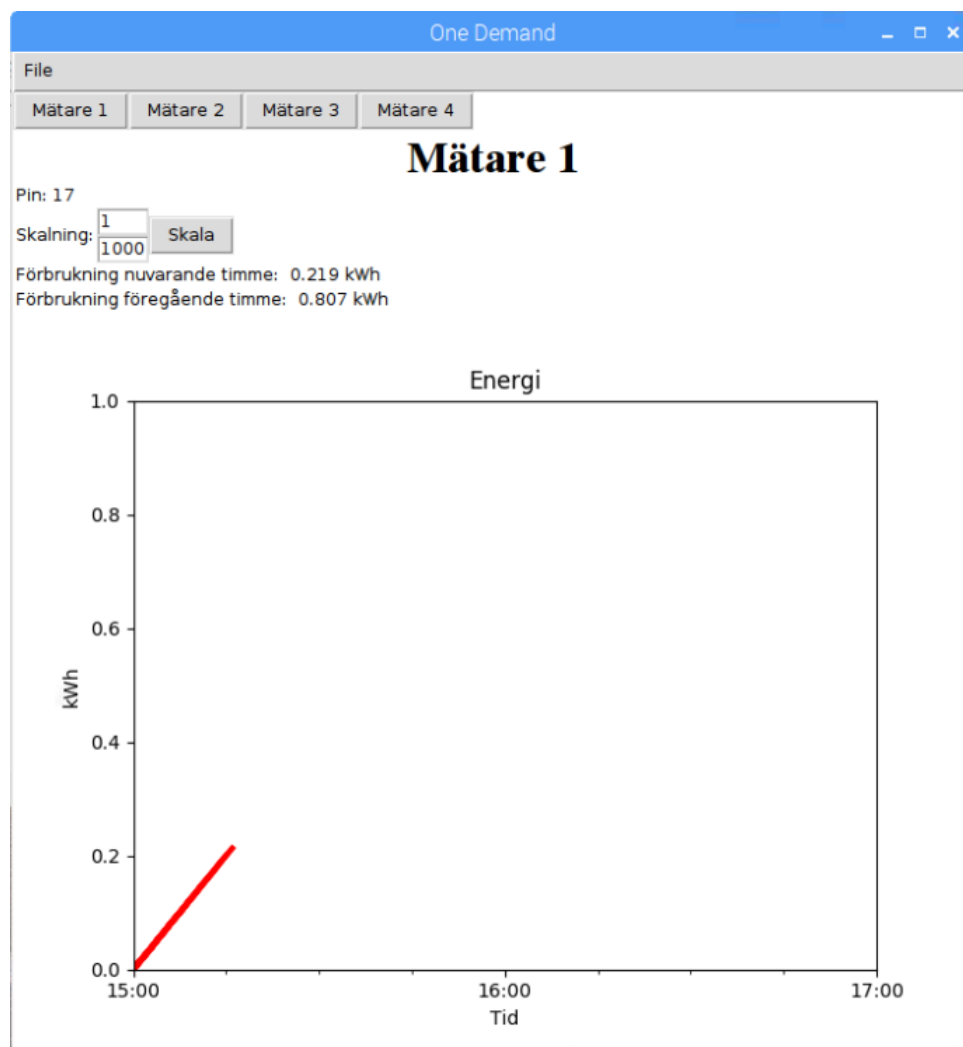
GUI:n som utvecklades innehåller 4 sidor, en sida för respektive mätare. Figur 11 visar 'Mätarsida 1'. Varje sida innehåller information om vilken ingång på RPi 3 som används. Varje sida visar förbrukningen för både nuvarande och föregående timme. Beslutet om hur GUI:n skulle se ut togs tillsammans med anställda på ONE. Grafen som programmerades till i den nya GUI:n är inspirerad av grafen som visas i figur 3. Det finns en möjlighet för användaren att ställa in hur stor förbrukningen är per puls. Längst ner på var sida är en graf som visar förbrukningen inom perioden av en timme. Grafen är utvecklad så att två timmars förbrukning kan visas upp samtidigt. Förbrukningen för var timme loggas på en extern fil och en skärmdump av grafen sparas även vid slutet av varje två timmars period. Även beslutet om hur mätvärdena ska presenteras i den externa filen togs tillsammans med anställda på ONE.

6.7 Test 2 resultat

Efter att programmet varit igång över natten visade grafen rätt förbrukning. Figur 11 visar hur GUI:n såg ut efter att ha varit igång hela natten. På figuren kan man se förbrukningen för nuvarande timme under rubriken 'Förbrukning nuvarande timme'. 'Förbrukning föregående timme' visar hur stor förbrukningen var under föregående timme. Grafen plottar förbrukningen och börjar på noll vid start av ny timme. Varje timme loggas den totala förbrukningen per timme i en extern fil. Figur 12 visar värdena som loggades per timme.

6.8 Test 3 resultat

Ett plötsligt strömbavbrott simulerades genom att dra ut strömadaptern från vägguttaget. När strömadaptern kopplades in igen i vägguttaget så startades RPi 3. Detta var det önskade resultatet.



Figur 11: GUI visar förbrukningen

```
2017-05-18 01:00
kWh: 0.827

2017-05-18 02:00
kWh: 0.837

2017-05-18 03:00
kWh: 0.84
```

Figur 12: Loggning per timme i fil

6.9 Statisk IP-adress för RPi 3

En dynamisk IP-adress kan vid omstart byta en dators IP-adress, därför behöver en RPi 3 ha en statisk IP-adress eftersom det ska vara möjligt att koppla upp sig mot den. En statisk IP-adress ställs in genom att följa följande steg:

1. Första steget är att ta reda på enhetens gateway adress. Skriv in 'netstat -nr' på terminalen. Figur 13 visar hur det ser ut på terminalen.

```
pi@raspberrypi:~ $ netstat -nr
Kernel IP routing table
Destination      Gateway          Genmask
0.0.0.0          10.109.150.1    0.0.0.0
10.109.150.0     0.0.0.0         255.255.255.0
```

Figur 13: Terminal visar hur man hittar sin gateway adress

2. Ta reda på enhetens DNS adress. Detta görs genom att skriva in 'cat /etc/resolv.conf' på terminalen. Figur 14 visar hur det ser ut på terminalen.

```
pi@raspberrypi:~ $ cat /etc/resolv.conf
# Generated by resolvconf
nameserver 10.109.134.10
pi@raspberrypi:~ $
```

Figur 14: Terminal visar hur man hittar sin DNS adress

3. Nu ska dhcpd.conf filen redigeras. Detta görs genom att skriva in 'sudo nano/etc/dhcpd.conf' på terminalen.
4. Längst ner på dhcpd.conf filen skriver du in följande:
`interface eth0 // Detta skriver du ifall du vill ha statisk IP-adress vid`

uppkoppling via Ethernet

```
static ip_address=100.12.0.1/24 // här skriver du din önskade IP-adress.  
Ha kvar "/24"  
static routers=324.109.0.1 // häyr skriver du din gateway adress  
static domain_name_servers=324.109.0.2 // här skriver du din DNS adress
```

```
interface wlan0 // Detta skriver du ifall du vill ha statisk IP-adress vid  
uppkoppling via WiFi
```

```
static ip_address=100.12.0.340/24 // här skriver du din önskade IP-adress.  
Ha kvar "/24"  
static routers=132.234.0.3 // här skriver du din gateway adress  
static domain_name_servers=342.567.0.12 // här skriver du din DNS adress
```

5. Tryck på knapparna ctrl + x för att lämna filen och spara genom att trycka på 'J' och sen 'Enter'.
6. Sista steget är att skriva in 'sudo reboot' på terminalen. Enheten kommer att startas om och kommer att ha den IP-adressen du angett.

6.10 Expansionsmöjligheter för RPi 3

I princip finns det oändligt många expansionsmöjligheter för RPi 3. Det går att skapa egna kretskort som kan fästas på RPi:s in- och utgångar, det finns även redan färdigt byggda kretskort som går att köpa in. Eftersom RPi 3 har färdigmonterade hål i sitt kretskort, går det att sätta på kretskort som inte ansluts till RPi 3:s in- och utgångar med hjälp av avståndsskruvar. Exempel på färdiga expansionskort är realtime clock module, kamera och olika displayer. Expansionskort som är relevanta till detta examensarbete är det skrivet om i avsnitt 6.13.

6.11 Utseende för specifikationsdokument

Ett specifikationsdokument måste skrivas tydligt. Dokumentet måste ha en rubrik som anger vilken produkt dokumentet tillhör till. Om det är en teknisk specifikation skall komponenter vara listade. En specifikation kan även ha ett kretsschema för hur hårdvaran ser ut. Det bör finnas en tabell som informerar om hur stor matningen till hårdvaran ska vara, hur mycket energi hårdvaran förbrukar, hur många in- och utgångar hårdvaran har och alla storheter bör skrivas med korrekt enhet.

Ett specifikationsdokument som beskriver mjukvara bör skrivas som en manual. Dokumentet kan börja med att informera vilket sorts programmeringsspråk som

använts samt hur en användare kan navigera sig genom programmet. Dokumentet beskriver vad en användare kan förvänta sig av mjukvaran.

6.12 Test för CE-märkning

Det finns ett antal tester som måste utföras för en produkt som behöver CE-märkas. För denna sortens produkt räcker det med att den klarar av EMC-direktivet. Testerna som utförs för EMC för en sådan här produkt är:

- Radio interferens.
- Immunitet mot transienter.
- Immunitet mot dämpade oscillerade vågformer.

6.12.1 Kapsling

Efter en diskussion med anställda på ONE kom man fram till att kapslingen skall vara klassad IP 21 enligt IEC 60529. Kapslingen skall även vara gjord utav plåt. Materialet valdes till plåt eftersom att det skulle klara av testet mot radio interferens. Dimensionerna för kapslingen får bestämmas efter att expansionskorten köpts in.

6.12.2 Strömadapter

Eftersom RPi 3 inte försörjs av ett batteri måste den alltid vara inkopplad till en strömadapter. Enligt Raspberry Pi Foundation är det rekommenderat att strömadaptern ska kunna mata 2,5 A till RPi 3. En adapter som är CE-märkt ska kunna klara av EMC-testerna.

6.13 Färdig produkt

För att hårdvaran ska utvecklas till en färdig produkt behöver det läggas till kretskort till RPi 3 och kapsling. Ett kretskort med skruvplintar behövs som kopplas på RPi:s in- och utgångar, detta i syfte att kunna använda tjockare sladdar. Ett annat kretskort behövs som innehåller skruvplintar och resistorer, de ska vara kopplade så att RPi:s och energimätarens in- och utgångar kan kopplas enligt kretsschemat som användes i test 1, se figur 9, detta kretsschema skall skruvas fast på det kretsschemat som är inkopplad på RPi 3:s in- och utgångar med avståndsskruvar. RPi 3 och de övriga kretskorten förseglas med en kapsling. På kapslingen bör in- och utgångarna vara numrerade.

Bilaga 6 visar hur ett kretskort kan se ut då alla in- och utgångar kopplas till en skruvterminal. Bilaga 7 visar samma kretskort som Bilaga 6 fast i 3D. Detta

kretskort kopplas på RPi 3 in- och utgångar och skruvas fast med avståndsskruv för bättre stabilitet. Tabell 4 visar vilken skruvterminal som går till respektive in- och utgång på RPi 3. Bilaga 10 visar hur en inkoppling av 4 st energimätare kan se ut. Bilaga 8 visar hur kretskortet för spänningsdelningen ser ut. Kortet är designat för endast inkoppling av 4 energimätare. Bilaga 9 visar samma kretskort som Bilaga 8 fast i 3D. Detta kretskort sätts enbart fast med avståndsskruvar. När man kopplar in energimätaren till RPi 3 ska man koppla så att en energimätare är på en sida av kretskortet på Bilaga 8. Tabell 3 visar funktionen av varje in- och utgång. Skruvterminalerna J1 och J2 är reserverade för inmatning med 3,3V och skruvterminalerna är reserverade för inmatning med 5V.

Tabell 3: Specifikation för in- och utgångar för kretskort från figur 8.4

Pin nummer	Befattning
J1 och J2	
1	Ingång för 3,3V
2	Utgång till S0-gränssnitt
3	Ingång från S0-gränssnitt
4	Utgång till RPi 3
5	Utgång till jord på RPi 3
J3 och J4	
1	Ingång för 5V
2	Utgång till S0-gränssnitt
3	Ingång från S0-gränssnitt
4	Utgång till RPi 3
5	Utgång till jord på RPi 3

Tabell 4: Visar vilken skruvterminal går till respektive in- och utgång på RPi 3

Skruvterminal J3	Pin på RPi3
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
Skruvterminal J1	
1	13
2	14
3	15
4	16
5	17
6	18
Skruvterminal J2	
1	19
2	20
3	21
4	22
5	23
6	24
7	25
8	26
9	27
10	28
Skruvterminal J4	
1	29
2	30
3	31
4	32
5	33
6	34
7	35
8	36
9	37
10	38
11	39
12	40

7 Slutsats

Syftet med examensarbetet var att hitta en enkorts dator som ska kunna byta ut nuvarande centralenhet. Det kan konstateras att detta är möjligt med en RPi 3. Examensarbetet visar att det är möjligt att samla in mätvärden från flera energimätare med en RPi 3. De huvudsakliga slutsatserna för detta examensarbete kommer att skrivas i detta kapitel. Slutsatserna är skrivna här nedan:

Vilka krav måste prototypen uppfylla?

- Kraven på prototypen är att den ska ha samma funktionalitet som ED3000. Prototypen ska kunna ha datorprogram installerat i sig. Den skall kunna kopplas till 4 st energimätare. En mer utförlig kravlista finns under rubriken 'Vilka krav måste enkorts datorn uppfylla?'

Vilka expansionsmöjligheter finns det för enkorts datorer?

- Expansionsmöjligheterna för en RPi 3 är i princip oändliga. Det finns redan färdiga expansionskort, t.ex olika sorters sensorer, kamera, högtalare etc. Det finns även möjlighet till att designa egna kretskort och sedan koppla på det på RPi 3s in- och utgångar. Expansionskortet behöver inte specifikt kopplas på RPi 3s in- och utgångar, det kan även sättas fast med avståndsskruvar.

Hur ska koden skrivas för att bli kompatibel med prototypen?

- Programmeringsspråket som användes var Python. Koden skrevs och modifierades på så sett så att den blev liten och effektiv eftersom det skulle ta så lite plats som möjligt. Python valdes eftersom det är ett programmeringsspråk med välutvecklade bibliotek just för RPi 3.

Vad för test ska utföras på prototypen för att se att den fungerar?

- Tester som utfördes för att se om prototypen fungerade enligt önskat var test 1, test 2 och test 3. Test 1 var ett test som skulle visa att RPi 3 kan ta emot pulser från en energimätare. Test 2 var ett test som skulle visa att pulser kan tolkas och visas upp snyggt i en GUI. Det visade sig att pulser kan tolkas och skrivas ut snyggt i en GUI. Test 3 var en simulering av ett strömavbrott. RPi 3 startade direkt efter att strömmen slagits på igen.

Hur ska specifikationsdokumentet se ut?

- Specifikationsdokumentet ska innehålla information om prototypen, så som teknisk data för hårdvaran och användarmanualer för både hårdvaran och det grafiska användargränssnittet. Detta gjordes och gavs till One Nordic.

7.1 Samhällsnytta

Eftersom detta är en produkt som mäter energiförbrukningen har denna produkt en direkt koppling till önskan att bli mer medveten om sin energiförbrukning. Vetskapen om sin energiförbrukning kan leda till ett försök att göra den mer effektivare i syfte att vara mer skonsam mot miljön men framförallt att vara mer effektiv i förbrukningen av energin. Eftersom produkten loggar energiförbrukningen för varje timme blir det enklare att se hur förbrukningen är var timme, även när anställda inte är på plats.

8 Framtida arbeten

I detta kapitel kommer det att skrivas om vad författarna av detta examensarbete tycker ska göras i framtiden.

8.1 Inköp av kretskort

Kretskorten som har designats, eller liknande, bör köpas in och fästas på RPi 3. Detta för att kunna byta ut kopplingsdäcken mot ett kretskort och så att det ser mer ut som en produkt som kan lanseras. Prototypen kan även göras kompatibel med fältbussar.

8.2 Inköp av kapsling

När kretskorten köpts in och fästs fast på RPi 3 bör kapslingen köpas in. Kapslingen bör vara gjord av plåt och vara rätt numrerad, vilket bör skrivas på kapslingen. Kapslingen kommer att behöva vara måttbeställd så att den passar prototypen.

8.3 Genomförande av tester för EMC

Tester för EMC kan genomföras först efter att kretskort och kapslingen köpts in. Dessa tester ska genomföras innan lanseringen av produkten.

8.4 Vidareutveckling av GUI

Eftersom examensarbetet avgränsar mot effektstyrning kan nästa steg vara att implementera detta. Andra funktioner kan även implementeras som en menyrad, finare design etc. Koden för GUI är öppen för förändringar när den lämnades över till ONE Nordic AB.

9 Referenser

- [1] ONE Nordic AB- Om oss <http://www.one-nordic.se/Om-One/> (hämtad 2017-03-15)
- [2] E.ON Försäljning Sverige AB- E.ON EnergiDirigent v1.4- Operatörsmanual (2007)
- [3] ONE Nordic AB- ED3000 TEKNISKA DATA
- [4] Firefly-Rk3288 - <http://wiki.t-firefly.com/index.php/Firefly-RK3288/Version.Introduction/en> (hämtad 2017-05-31)
- [5] E.ON Försäljning Sverige AB - E.ON EnergiDirigent v1.4 - Administratörsmanual (2007)
- [6] Firefly- Firefly-RK3288 Specifications http://en.t-firefly.com/en/firenow/firefly_rk3288/specifications/ (hämtad 2017-03-21)
- [7] Raspberry Pi Foundation - Raspberry Pi 3 model B <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (hämtad 2017-03-21)
- [8] BeagleBoard - <https://beagleboard.org/> (hämtad 2017-03-21)
- [9] Solid Run-HummingBoard Specifications <https://www.solid-run.com/freescale-imx6-family/hummingboard/hummingboard-specifications/> (hämtad 2017-03-21)
- [10] Forth - <https://www.forth.com/resources/forth-programming-language/> (hämtad 2017-05-07)
- [11] Your Europe- CE-marking <http://europa.eu/youreurope/business/product/ce-mark/> (hämtad 2017-04-15)
- [12] European Commission-EMC Directive- <http://ec.europa.eu/growth/sectors/electrical-engineering/emc-directive> (hämtad 2017-04-11)
- [13] Oracle What is JavaFX?- <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm> (hämtad 2017-04-23)
- [14] Python What is Python?- <https://www.python.org/doc/essays/blurb/> (hämtad 2017-04-23)
- [15] Europastandard EN-62053-31 (inköp)
- [16] Computer hope CPU <http://www.computerhope.com/jargon/c/cpu.htm> (hämtad 2017-04-15)
- [17] Python Tkinter - <https://wiki.python.org/moin/TkInter> (hämtad 2017-05-02)
- [18] Python PyQt - <https://wiki.python.org/moin/PyQt> (hämtad 2017-05-02)

- [19] Python PySide - <http://wiki.qt.io/PySide> (hämtad 2017-05-02)
- [20] GNU GPL - <https://www.gnu.org/licenses/gpl-3.0.en.html> (hämtad 2017-04-17)
- [21] GNU LGPL- <https://www.gnu.org/licenses/lgpl-3.0.en.html> (hämtad 2017-04-17)
- [22] CE-mark <https://ec.europa.eu/growth/sites/growth/files/ce-mark.png>
(Bild hämtad 2017-04-04)
- [23] Standard IEC 60529 (inköp)

Bilaga Referenser

- [1] ONE Nordic AB- ED3000 TEKNISKA DATA

Ordlista

GUI	Graphical user interface
CPU	Central processing unit
ALU	Arithmetic logic unit
kWh	Kilowatttime
GPL	GNU General Public License
LPGL	GNU Lesser General Public License
EMC	Electromagnetic compatibility

Bilaga 1

ED 3000 TEKNISKA DATA

PRELIMINÄR 1999-10-01

Allmänt:

Spänningsmatning: 24 VAC +/- 20 % alt.

24 VDC +/- 20%.

Kontakt: 1 st. 2-polig fränskiljbar skruvplint, 5,08 mm delning.

Effektförbrukning: Max 15 VA.

Internt genererad matning: 24 VDC / 200 mA för digitala ingångar.

18 VDC / 100 mA för SIOX

Indikeringar: 5 st. lysdioder för status på spänningarna.

24V (200 mA), 15V (100 mA),

5V (1 A), 18V (100 mA) resp. -8V (100 mA).

Kapsling: Lackerad plåtlåda.

Kapslingsklass: IP 20 enligt EN 60 529/IEC 529.

EMC Standard Immunitet: Minst enligt EN-50082-1, (1997) , Generic immunity standard.
Part 1. Residential, commercial and light industry

EMC Standard Emission: Minst enligt EN-50081-1, (1992) , Generic emission standard.
Part 1. Residential, commercial and light industry

Omgivningstemperatur: 0 C - +40 C vid drift.

-20 C - +70 C vid lagring.

Luftfuktighet: Max 90 % icke-kondenserande.

Yttermått (inkl fästöron): 349*137*89 mm.

Vikt: ca: 2,5 kg.

Centralkort (undre kort):

CPU Motorola MC68EN302PV20 20 Mhz

Integrerad multiprotokoll processor med Ethernet.

Realtidsklocka Batterimatad, Digitalt justerbar.

År, månad, dag, timme, minut, sekund, veckodag. Automatisk skottårs korrigering.

Programminne RAM 2 Mbyte batterimatat CMOS RAM utbyggbart till
16 Mbyte fördelat på 2 st. 72 pinnars SIMM moduler.

Programminne ROM 512 kbyte utbyggbart till 2 Mbyte OTP eller
FLASH EPROM fördelat på 2 st. 32 pinnars lågprofil PLCC kapslar.

Programminne EEPROM 1 kbyte främst för systemparametrar

Batteri Laddningsbart litiumbatteri 3V, 50 mAh
alt. 2 * R03 1,5V.

Laddningsström Li: Max 1 mA, Trickle charge.

Laddningstid Li: ca 50 timmar till fulladdat.

Backupstid: Typ 1 år vid fulladdat batteri (1Mbyte RAM).

Livslängd Li: Min 1000 cykler 10 %, min 10 år vid normaldrift.

Seriekanal 1 och 2 2 st. RS-232-D/E, DTE, 1.200-105.200 bps.

Max kabellängd enligt EIA-RS-232-D: 15m.

Kontakt: DE9P, 9-polig DSUB hane.

1 DCD

2 RXD

3 TXD

4 DTR

5 GND

6 DSR

7 RTS

8 CTS

Skydd: EMI-filter för störningsdämpning av transienter.

Seriekanal 3 (Alt 1) Transfortoriserat RS-485 interface,
1.200-105.200 bps.

Max kabellängd enligt EIA-RS-485: 1200m (100.000 bps, R < 80/1000m).

Rekommenderad kabel: UTP 4*2 Cat 3 eller bättre.

Kontakt: RJ45 8/8.

1 15 VDC +, 100 mA, för handterminal

2 15 VDC +

3 A (+)
4 B (-)
5 ISO GND
6 ISO GND
7 15 VDC -, för handterminal
8 15 VDC -, för handterminal

Bygglingsbart termineringsmotstånd 120 .
Skydd: Transformator optokopplare.
PTC-motstånd som kortslutningsskydd.
Zenerdioder för transienter.
Isolationsspänning: Typ 1600 VRMS .

Seriekanal 3 (Alt 2) RS-232-D/E, DTE, 1.200-105.200 bps.
(övre kortet) Max kabellängd enligt EIA-RS-232-D: 15m.
Kontakt: DE9P, 9-polig DSUB hane.

2 RXD
3 TXD
4 DTR
5 GND

Skydd: EMI-filter för störningsdämpning av transienter.

Seriekanal 3 (Alt 3) SIOX buss 300-19.200 bps
(övre kortet) Optoisolerad strömslinga.
Kortslutningsström: 100 mA
Max kabellängd: 100m
Kontakt: 2 polig fränskiljbar skruvplint. 5.08 mm delning.
Märkning: SIOX -, SIOX +

Ethernet Enligt IEEE 802.3, 10BASE-T (10 Mhz Twisted pair).
Max kabellängd per segment: 100m.
Rekommenderad kabel: UTP 4*2 Cat 3 eller bättre.
Kontakt: I (rak) för anslutning till HUB, Växel etc.
Kontakt: X (korsad) för anslutning till PC etc.
Kontakt: I (rak): RJ45 8/8.

1 TD+
2 TD-
3 RD+
6 RD-

Kontakt: X (korsad): RJ45 8/8
1 RD+
2 RD-

3 TD+
6 TD-
6 st lysdioder på CPU-kortet för indikering av portens status.
TX (Sändning), RX (Mottagning), CL (Kollision),
LIL (Bra länk), PLR (Polaritetsfel), JAB (Störning).
Skydd: Transformator, Lågpas filter.
Isolationsspänning ingång/utgång: Typ 2000 VRMS .
Isolationsspänning TD/RD: Typ 500 VRMS .

PCMCIA (Option) PCMCIA Ver. 2.1 / JEIDA ver. 4.1
2 st av typ I och II alt. 1 st av typ III.

I/O-kort (övre kortet):

Analoga Ingångar
Antal: Inga

Analoga Utgångar
Antal: Inga

Digitala Ingångar
Antal: 16 st, positiv logik, 24 VDC.
Omslagspunkt: ~ 8 V.
Statusindikering: 16 st. gula lysdioder placerade i fronten.
Filter: Första ordningens lågpasfilter. Tidskonstant 1 ms
Samplingstid: 10 ms.
Kontaktdon: 2 st. 10-poliga fränskiljbar skruvplint, 5,08 mm delning.
Märkning: Screentryckt eller folie
0V,
1 (AKTIV ENERGI),
2 (SYNKPULS),
3 (REAKTIV ENERGI),
4, 5, 6, 7, 8, +24V
0V, 9, 10, 11, 12, 13, 14, 15, 16, +24V
Skydd: Zenerdioder som överspänningsskydd.
Lågpasfilter.

Tryckknappar
Antal: 2 st, Samplingstid 10 ms.
Märkning: (TK1, TK2)

Digitala Utgångar
Antal: 12 st, slutande reläer.
Statusindikering: 4 st. röda samt 8 st gula lysdioder placerade i fronten.
Samplingstid: 10 ms.

Kontaktidon: 3 st. 8-poliga frånskiljbar skruvplint, 5,08 mm delning.
Märkning: 1 (VAKTHUND SYSTEM),
2 (VAKTHUND PROGRAM),
3 (A-LARM),
4 (B-LARM),
5, 6, 7, 8, 9, 10, 11, 12
Brytförmåga: AC: 10A, 240 VAC, 2000 VA.
DC: 3A vid 30 V; 0,6A vid 60 V; 0,15A vid 220 V.
Livslängd: Min. 10*10⁶ cykler.
Skydd: Galvanisk separation genom relät.
Isolationsspänning: 1000 VRMS över kontakter,
5000 VRMS mellan kontakt och spole.
Metalloxidvaristor 250 VAC som transientskydd och gnistsläckare.

Lysdioder

Antal: 3 st placerade i fronten.
Märkning: ALARM, röd, programmerbar.
POWER, grön, indikerar 5V internspänning.
RUN, grön, indikerar med blink (ca. 0,6 Hz) att
systemet fungerar (systemprogrammen exekverar).

Bilaga 2

Exempel på Forthkod:

```

: STAR ( -- ) \ Print a single star
42 EMIT ; \ 42 is the ASCII code for *

: STARS ( n -- ) \ Print n stars
0 DO STAR LOOP ; \ Loop n times (0 up to n-1) and execute STAR

: SQUARE ( n -- ) \ Print an n-line square of stars
DUP 0 DO \ Loop n times, keeping (DUP-licating) n on the stack
DUP STARS CR \ Each time, print n stars then print CR
LOOP DROP ; \ After loop is done, drop the n from the stack

: TRIANGLE ( n -- ) \ Print an n-line triangle
1 + 1 DO \ Loop n times from 1 to n (instead of 0 to n-1)
I STARS CR \ This time use the inner loop index I
LOOP ;

: TOWER ( n -- ) \ Print a "tower" with an base of size n
DUP \ DUP-licate n (since it is used twice below)
1 - TRIANGLE \ Print a triangle 1 size smaller than n
SQUARE ; \ Print a square base of size n

```

```
\ Sample test session:  
CR 7 STARS \ user types this line, output follows  
***** ok  
  
CR 3 TRIANGLE \ user types this line, output follows  
*  
**  
***  
ok  
  
CR 6 TOWER \ user types this line, output follows  
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
ok  
\ End of sample code and test session.
```

Bilaga 3

Exempel Java- och JavaFXkod:

```
package helloworld;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

public class HelloWorld extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Hello World!");
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

        StackPane root = new StackPane();
        root.getChildren().add(btn);
        primaryStage.setScene(new Scene(root, 300, 250));
        primaryStage.show();
    }
}
```

Bilaga 4

Exempel på Python- och Tkinterkod:

```
from tkinter import *

class Application(Frame):
    def say_hi(self):
        print("hi there, everyone!")

    def createWidgets(self):
        self.QUIT = Button(self)
        self.QUIT["text"] = "QUIT"
        self.QUIT["fg"] = "red"
        self.QUIT["command"] = self.quit

        self.QUIT.pack({"side": "left"})

        self.hi_there = Button(self)
        self.hi_there["text"] = "Hello",
        self.hi_there["command"] = self.say_hi

        self.hi_there.pack({"side": "left"})

    def __init__(self, master=None):
        Frame.__init__(self, master)
        self.pack()
        self.createWidgets()

root = Tk()
app = Application(master=root)
app.mainloop()
root.destroy()
```

Bilaga 5

Exempel på kod för PyQt respektive PySide:

```
#!/usr/bin/env python
```

```
import sys
from PyQt4 import Qt

a = Qt.QApplication(sys.argv)
hello = Qt.QLabel("Hello, World")
hello.show()
a.exec_()
```

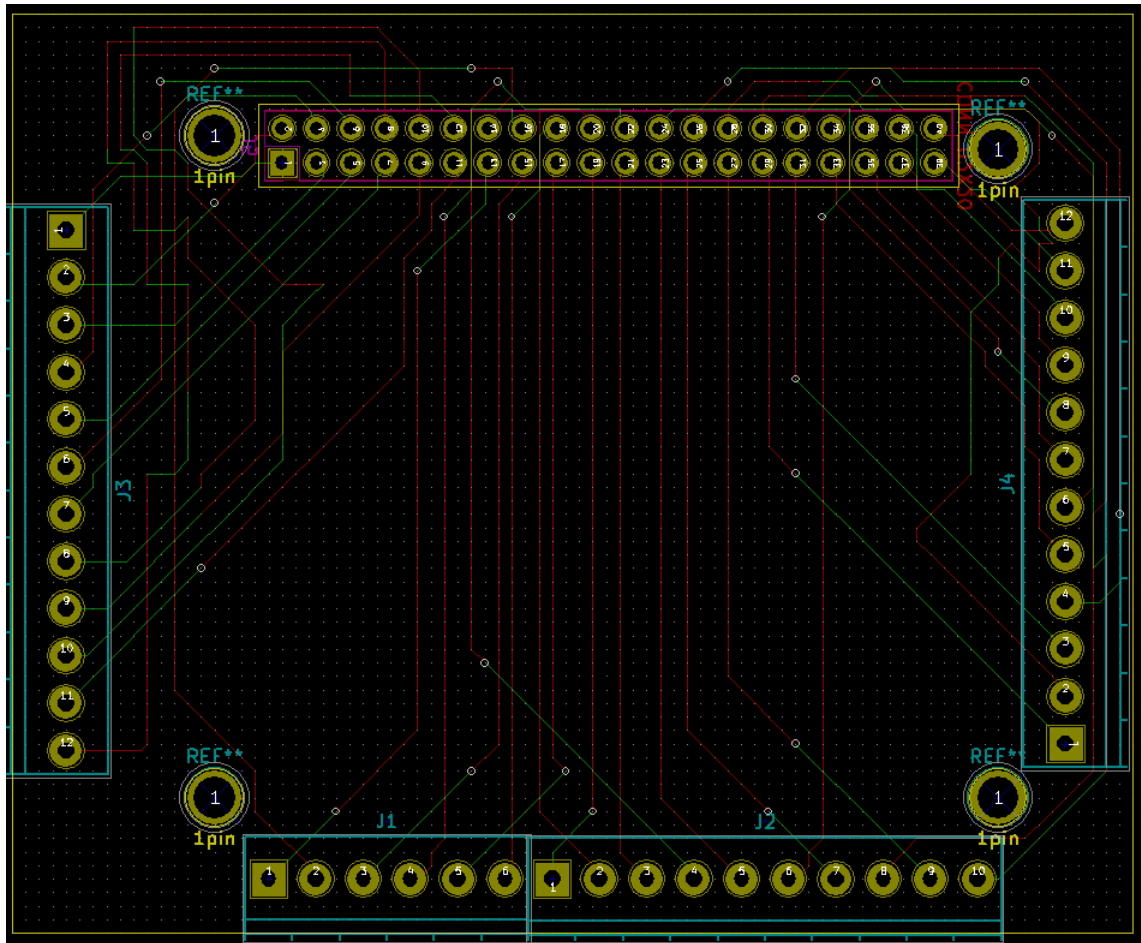
```
#!/usr/bin/python
```

```
# Import PySide classes
import sys
from PySide.QtCore import *
from PySide.QtGui import *

# Create a Qt application
app = QApplication(sys.argv)
# Create a Label and show it
label = QLabel("Hello World")
label.show()
# Enter Qt application main loop
app.exec_()
sys.exit()
```

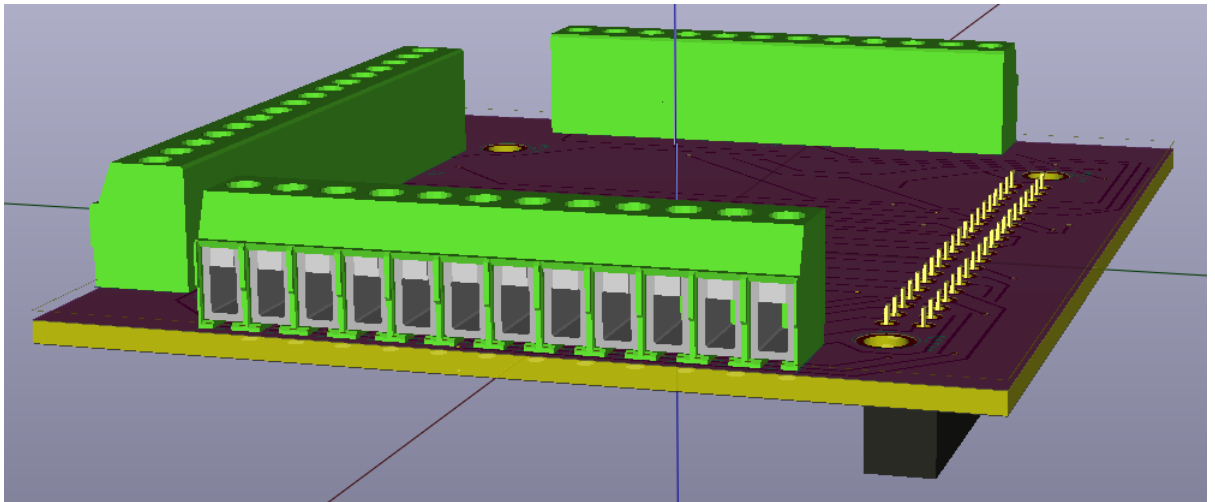
Bilaga 6

Kretskort för att göra alla in- och utgångar till skruvterminaler.



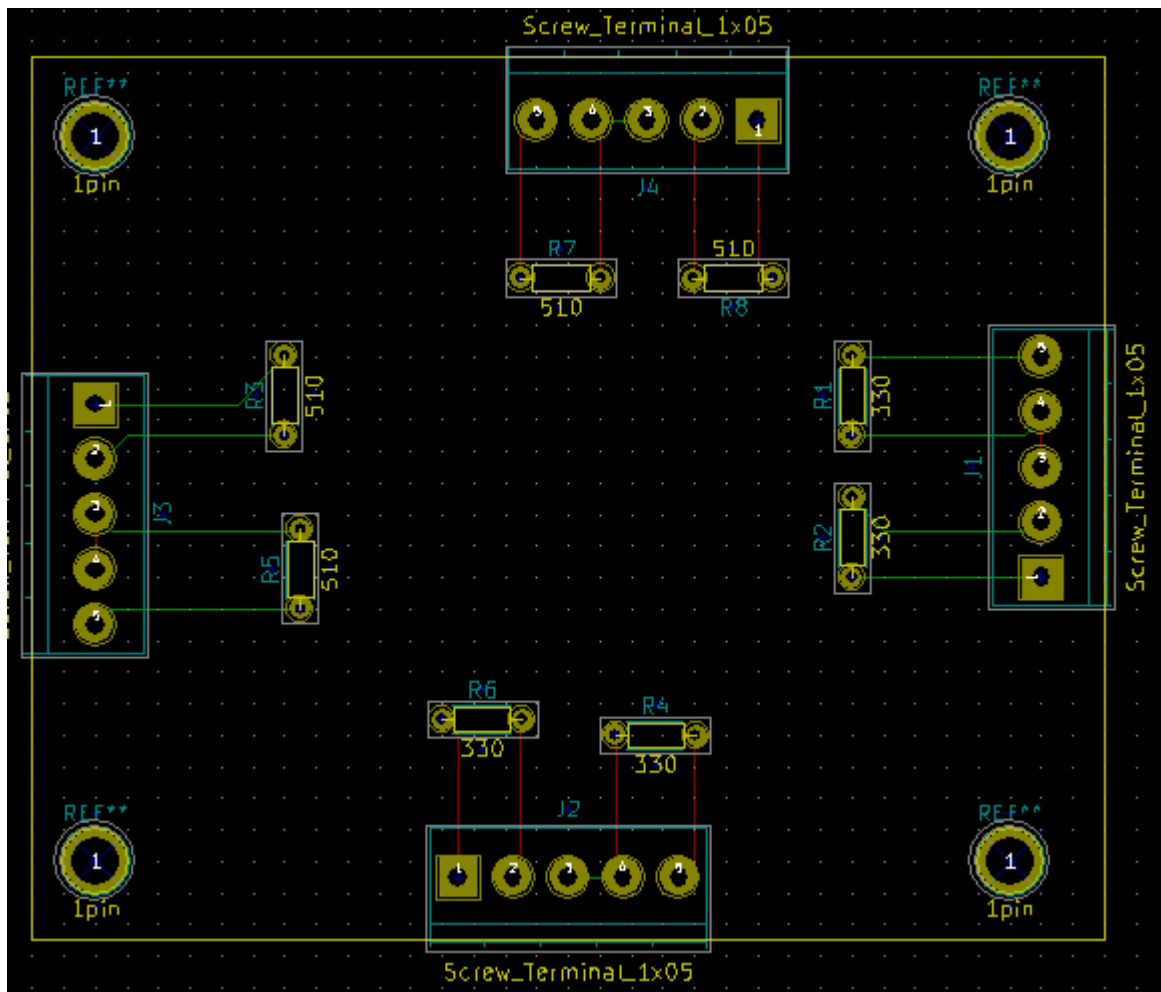
Bilaga 7

Bild som visar kretskortet på Bilaga 6 i 3D



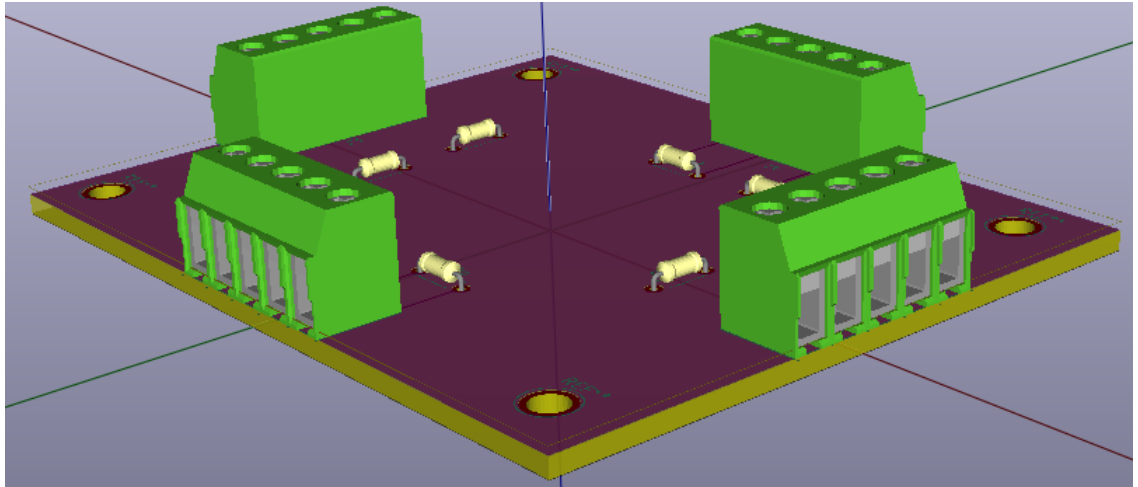
Bilaga 8

Kretskort med spänningsdelning för 4 energimätare.



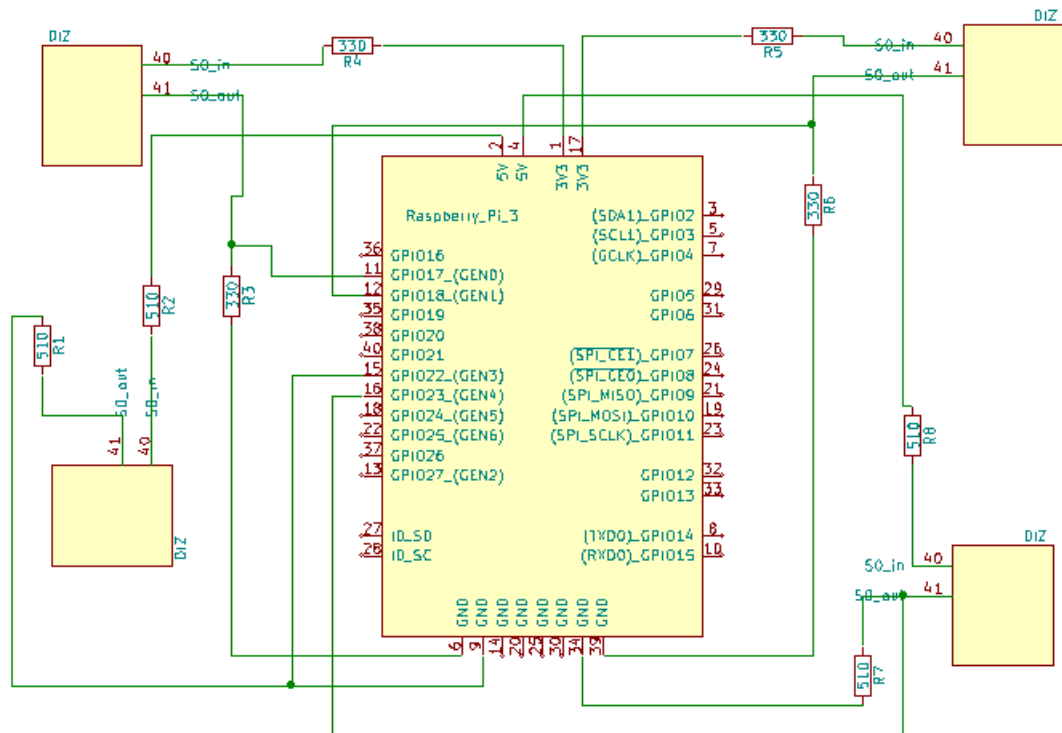
Bilaga 9

Bild som visar kretskortet på figur 8.4 i 3D



Bilaga 10

Kretsschema för RPi 3 kopplad med 4 st energimätare





LUND
UNIVERSITY

Series of Bachelor's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2017-587
<http://www.eit.lth.se>