



CHALMERS



Utveckling av Testbänk för PLC01

Mjuk- och hårdvarudesign för simpel testtrigg

Zacharias Faleberg Nilsson
August Berg

INSTITUTIONEN FÖR DATA OCH INFORMATIONSTEKNIK

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2021
www.chalmers.se

EXAMENSARBETE 2021

Utveckling av Testbänk för PLC01

Mjuk- och hårdvarudesign för simpel testtrigg

Zacharias Faleberg Nilsson August Berg



CHALMERS

Institutionen för Data och Informationsteknik

Avdelning för Datorteknik

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2021

Utveckling av Testbänk för PLC01
Mjuk- och hårdvarudesign för simpel testtrigg
Zacharias Faleberg Nilsson
August Berg

© Zacharias Faleberg Nilsson, 2021.
© August Berg, 2021.

Handledare: Arne Linde, Institutionen för Data och Informationsteknik
Examinator: Jonas Duregård, Institutionen för Data och Informationsteknik

Examensarbete 2021
Institutionen för Data och Informationsteknik
Avdelning för Datorteknik
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Omslag: PLC01.

Typsatt i L^AT_EX
Göteborg, Sverige 2021

Utveckling av testbänk för PLC01
Mjuk- och hårdvarudesign för simpel testrigg
Zacharias Faleberg Nilsson
August Berg
Institutionen för Data och Informationsteknik
Chalmers Tekniska Högskola

Sammanfattning

Att brygga den omoderna styr- och reglerteknik världen med den moderna utvecklingen av mjuk- och hårdvara har sina fördelar, PLC01 är ett försök till detta. En metod för enklare produktion och utveckling av PLC01 är att tillverka en testbänk som verifierar nyproduktionen av dessa enheter. För en smidig verifiering krävs en automatiserad testbänk. Testbänken kommer utvecklas med skriptbar mjukvara och skalbar hårdvara. Att spegla analoga ingångar mot utgångar mellan nyproducerad PLC01 och testbänken utgör ett test utav spänningsnivåer. För temperaturmätning mäts yttre resistansen mellan två portar. För testning av motstånd mellan portar undersöktes flera metoder för att simulera en resistansändring.

Abstract

To bridge the outdated control and regulation technology world with the modern software development has its advantages, PLC01 is an attempt at this. One method to simplify the production and development of PLC01 is to manufacture a test bench that verifies the new production of these units. For a smooth verification, an automated test bench is required. The testbench will be developed with scripted software and scaleable hardware. Mirroring analog inputs and outputs between newly produced PLC01 and the test bench is a test of voltage levels. For temperature measurement, the external resistance between two ports is measured. To test the resistance, several methods were examined to simulate a resistance change.

Förord

Vi vill tacka vår kontaktperson på Teracloud AB David Brinnen för möjligheten att utföra detta examensarbete och för handledning och stöd under projektets gång. Sedan vill vi även tacka vår handledare Arne Linde för stöd och idéskapande inom projektet och rapporten. Vidare vill vi också tacka David Albady, Alexey Shilenkov och Rafael Martins på Teracloud AB för genomgång och hjälp med PLC01.

Göteborg, juni 2021

Innehåll

Figurer	xi
Tabeller	xiii
1 Introduktion	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Avgränsningar	2
1.4 Frågeställning	2
2 Teoretisk bakgrund	3
2.1 Programmable Logic Controller (PLC)	3
2.2 Ratiometrisk mätning	3
2.3 RS232	4
2.4 Temperaturgivare	4
2.5 Secure Shell (SSH) och SFTP	5
2.6 Programmeringsspråket Python	5
2.6.1 Subprocess	5
2.7 CanOpen	5
2.8 I2C (I^2C)	6
2.9 Digital potentiometer	6
2.10 Arduino UNO	6
3 Metod	9
3.1 Tidsplan och planeringsrapport	9
3.2 Konstruktion	9
3.3 Konfiguration	10
3.4 Automatiserad testning	10
3.5 Temperaturmätning	14
3.5.1 Metod 1: Simulerad yttre resistans	14
3.5.2 Metod 2: Stickprov	14
3.5.3 Metod 3: Digital potentiometer	15
3.6 Verifiering av testbänkens funktion	16
4 Resultat	19
4.1 Analog spänningsmätning	19
4.2 Temperaturmätning	19

4.2.1	Metod 1	19
4.2.2	Metod 2	20
4.2.3	Metod 3	21
4.3	Uppbyggnad	21
5	Diskussion	23
5.1	Mätning av temperatur/resistans	23
5.2	Utvecklingsmöjligheter	24
5.2.1	Inläsning av portar genom config.cfg-filen	24
5.2.2	Strömmätning	24
5.2.3	PLC01 som styrenhet	24
6	Slutsats	25
	Bibliography	27

Figurer

2.1	Resitansändring för PT-1000	4
3.1	PLC01s I/O portar visualiserat från sidan.	9
3.2	Fullständig krets.	10
3.3	Flödesschema för test av portar.	11
3.4	Körning av test där input inte kunde verifieras.	12
3.5	Testfil där testresultat skrivs ut i.	13
3.6	Körning av test sett från PLC01.	13
3.7	Körning av test sett från tesbänk.	13
3.8	Koppling vid resistansmätning metod 1.	14
3.9	Koppling vid resistansmätning metod 2.	15
3.10	Ett exempel hur testet ser ut under körning.	16
3.11	övervakning av portar på PLC01.	17
4.1	Metod 1: Simulering av resistans.	22
4.2	Metod 2: Stickprov.	22
4.3	Metod 3: Digital potentiometer	22

Tabeller

3.1	Resistansförändring beroende på seriell data överföring.	15
4.1	Mätvärden för Metod 1: Simulerad yttre resistans.	20
4.2	Mätvärden för Metod 2: Stickprov.	21
4.3	Resistansförändring beroende på seriell data överföring.	21

1

Introduktion

1.1 Bakgrund

Styr- och reglerteknik är en central del i den industriella världen idag. För att kunna styra robotar eller reglera värden med enkel logik krävs en PLC som är själva hjärnan i processen. Vissa hävdar att den nuvarande PLC:n inte kan mäta sig med den moderna programutvecklingen. Teracloud AB har som vision att brygga den omoderna styr- och reglertekniksvärlden till den modern mjukvara och hårdvara med utvecklingen av en ny typ av PLC. Teracloud AB tillsammans med systerbolaget EnergyMachines ApS utvecklar en ny typ av PLC som de kallar PLC01 som bland annat kommer att användas i EnergyMachines värmepumpar. Projekt påbörjades 2017 med den första generationen av produkten och företaget utvecklar just nu generation tre av PLC01. I deras utveckling av produkten behövs en testbänk för enklare felsökningen och verifiering av produkten och kunna testa flertal enheter under kort tid. Denna testbänk är vår uppgift att utveckla. Testbänken ska verifiera bland annat I/O-portar och resistorbaserade sensorer. För dessa sensorer behövs det utvecklas en hårdvara, eftersom ett test inte kan genomföras genom att spegla portar på liknande sätt som vid test av de analoga portarna. Material för konstruktionen av hårdvaran kommer företaget att tillhandahålla. Signalernas karaktäristik är både inkommande och utgående vilka är definierade i mjukvaran. En enhet kan konfigureras att insignalerna motsvarar en andra enhet utsignaler. På så sätt kan två enheter testas samtidigt på ett simpelt och billigt tillvägagångssätt. Denna lösning för testning av hårdvara och mjukvara möjliggör test i mindre tekniskt utrustade miljöer utan att använda en testbänk med verktyg. I dagsläget kontrolleras varje del i PLC:n var för sig, detta är en tidskrävande process. Uppgiften är att utveckla mjuk- och hårdvara för att effektivisera dessa steg.

1.2 Syfte

Syftet med projektet är att finna en simpel och till stor del automatiserad lösning till att verifiera hårdvarans funktionalitet genom att testa att I/O-portarna ger önskade analoga och digitala värden. Testbänken som utvecklas är tänkt att vara en skalbar grund som kan utvecklas vidare och användas inom företaget för att verifiera och felsöka nytillverkade enheter i takt med att PLC01 utvecklas.

1.3 Avgränsningar

PLC01:s funktioner kommer inte att testas genom att styra någon maskin, anläggning eller annan process. Eftersom två stycken PLC01 är vad som finns tillgängligt under projektet så kommer den ena PLC:n speglas mot den andra och användas för att mäta att rätt utsignaler etc. genereras av den PLC01 som testas. Flera PLC01 kan inte testas, då endast två stycken enheter fanns till hands under projektet. Testbänken konstruerades för att kunna användas på samtliga typer av PLC01. Eftersom PLC01 i skrivande stund är i utvecklingsstadiet, kunde vissa funktioner som beräknas att vara tillgängliga i slutprodukten, men som ännu inte var implementerade i den testade versionen, aldrig testas. T.ex. är strömstyrkan hos I/O-portar en intressant del att testa, men strömstyrka kan varken ändras eller mätas i nuvarande version utav PLC01 och därför implementerades aldrig test utav strömstyrka i testbänken.

1.4 Frågeställning

- Hur kan en krets konstrueras för simulering av en temperaturgivare?
- Kan en helt automatisk testbänk utvecklas för PLC01?
- Hur ser utvecklingsmöjligheterna ut för framtida studier?

2

Teoretisk bakgrund

2.1 Programmable Logic Controller (PLC)

Ett programmerbart styrsystem förkortat PLC från engelskans programmable logic controller är i industri- och automationssammanhang vanligt förekommande. En PLC använder sig utav signaler i ingående och utgående riktning (förkortning I/O) och logik för att kunna ge styrsignaler. Dessa styrsignaler är inom industri oftast monteringsband, robotar eller annan typ av mekanik i en större process. En PLC är en enklare typ utav dator. Följande är de delar som krävs för en PLC [1] :

- Processor - läser av ingångar och utför PLC-programmet som därav ger utsignaler.
- Minne - oftast används ett icke-flyktigt minne (ett minne som behåller all data även när spänningen stängts av).
- Ingångar - tar in elektroniska signaler och skickar informationen till processorn. Dessa elektroniska signaler kan vara resultat av gränslägesbrytare eller temperaturgivare.
- Utgångar - ger ifrån sig elektroniska signaler efter kommando från processorn. Dessa elektroniska signaler kan dra en kontaktor som driver en motor.

Ingångar och utgångar kan tolkas som digitala portar beroende på vilken spänning som används. En spänning mellan 0 till 0.8 volt tolkas som låg (0:a). Om spänningen mäts mellan 2 och 5 volt antas detta som hög (1:a). En PLC01 använder sig inte utav digitala portar utan analoga. Detta betyder att inte bara hög eller låg nivå kan antas utan vilken spänning som helst mellan 0 och 5 volt. I de fall då PLC01:s portar behöver användas som digitala kommer de standarder som beskrevs ovan användas. [2].

2.2 Ratiometrisk mätning

Ratiometrisk mätning inom elektroteknik syftar på hur en utgång är direkt proportionell mot dess ingång. Det betyder att en ratiometrisk mätning hittar ett förhållande i form av procent mellan två värden. Detta används bland annat i en multimeter för att kunna mäta den yttre resistansen där förhållandet jämförs med det inre motståndet. För att kunna utföra mätningen så läggs en känd spänning mellan två portar på multimetern där resistorn som ska mätas placeras. Multimetern mäter sedan hur stor spänning som har lagts över det inre motståndet, dess yttre resistans kan sedan beräknas [3].

2.3 RS232

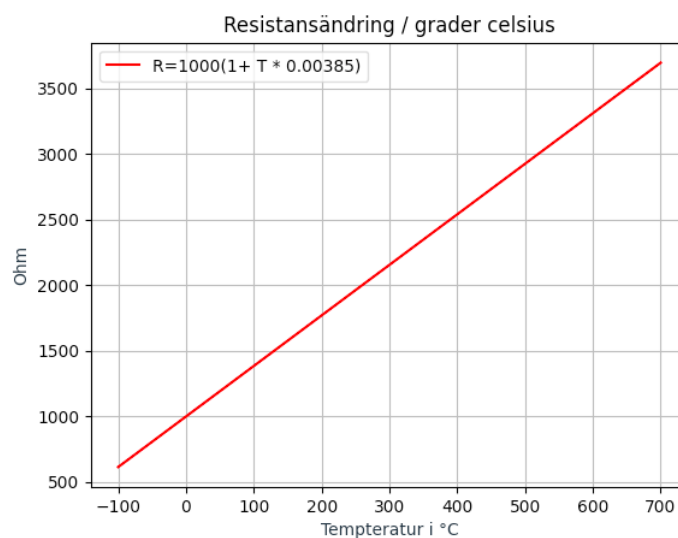
RS232, där RS står för engelskans recommended standard är en standard för seriell kommunikation. Binär data överförs med RS232 bitvisa serier, där start- och stopp-bitar används för att signalera när dataorden börjar och slutar. Längden för ett dataord kan väljas mellan 5 och 8 bitar, exkluderat start- och stoppbitar [4]. I projektet användes seriell dataöverföring mellan dator och PLC. Anslutningen kopplas mellan en USB-port från datorn till en USB/RS232-converter integrerad i PLC-enheten. Data överfördes med en hastighet på 115.2 kBd [5].

2.4 Temperaturgivare

Vanligt använda givare för att mätning av temperatur är så kallade motståndstermometrar, till exempel PT-1000 eller NTC-motstånd. Gemensamt för dessa givare är att deras resistansvärde varierar med ändring av dess temperatur. PT-1000 är en typ av givare som är tillverkad av metallen platina och har ett resistansvärde på 1000Ω vid 0°C . Vid en ökning av temperatur ökar givarens resistans linjärt enligt ekvation (2.1). Denna linjära graf kan ses nedan (se Figur 2.1). NTC-motstånd är till skillnad från PT-1000 tillverkade av halvledarmaterial och dess resistans minskar med en ökning av temperatur. Resistansminskningen hos NTC-motstånd sker olinjärt och beräknas med ekvation (2.2) där T är temperaturen och α är temperaturkoefficienten som anges i kelvin [6].

$$R = 1000(1 + T \cdot 3.85 \cdot 10^{-3})\Omega \quad (2.1)$$

$$R_T = R_0 \cdot e^{\alpha/T}\Omega \quad (2.2)$$



Figur 2.1: Resistansändring för PT-1000

2.5 Secure Shell (SSH) och SFTP

SSH är ett TCP/IP protokoll för kommunikation över datanätverk. Secure Shell användas för att ansluta till andra enheter via internet, eller på lokala nätverk. I detta projekt används SSH för att kunna ansluta och modifiera en PLC. När själva SSH anslutningen är etablerad kan PLC:ns innehåll ses från datorn som skapade SSH anslutningen (mappar, filer etc.) [7]. Innehållet kan även modifieras genom SSH anslutning, det betyder t.ex. att text filer kan skrivas i, raderas eller skapas. SFTP är i grund protokollet FTP vilket kopplas in i SSH protokollet för att tillhandahålla en krypterad länk. Under SSH anslutning kan SFTP (SSH File Transfer Protocol) utföras. SFTP används för att hantera och skicka filer över en den krypterade SSH anslutning. [8].

2.6 Programmeringsspråket Python

Python kommer användas främst för att läsa och skriva värden till I/O-portarna i form av spänningsnivåer. Det kommer också användas för loggning av resultat, verifiering utav portar, samt start och stop för testkörning. Python är i detta sammanhang ett gynnsamt språk då operationer i detta projekt inte behöver ta hänsyn till tidseffektivitet, i sådant fall skulle ett språk som C eller C++ användas. Python uppdateras och ändras precis som många andra programmeringsspråk med tiden till nyare versioner, Python3 som är den nyaste versionen är den som kommer att användas i detta projekt. Python har valts på grund av dess enkelhet där det finns mängder inbyggda bibliotek och moduler för olika tillämpningar [9].

2.6.1 Subprocess

Subprocess är en inbyggd python3 modul som kan användas för att köra flera processer samtidigt. Modulen kan också användas för att skriva och exekvera shellkommands. Vilket användas i detta projekt för att kunna läsa och skriva till I/O-portar.

2.7 CanOpen

CanOpen är ett kommunikationsprotokoll som bland annat kan förse en användare med information om en enhets I/O-portar. Protokollet behövs för att kunna skriva och läsa från enhetens portar. De olika läs/skriv funktionerna är definierade i enhetens config-fil. I denna fil kombineras en eller flera pinnar med olika konfigurationer för att skapa en grupp. Detta betyder att en grupp kommer ha två typer av information, vilken eller vilka pinnar som är aktuella och vilken egenskap som ska användas. Tillsammans utger detta ett fungerande system där en eller flera pinnar definieras med en egenskap. Dessa grupper kan sedan läsas av i terminalen. Följande egenskaper kan göras för nämnda grupper:

- Ingång för analog volt.
- Ingång för analog strömskillnad.
- Ingång för resistor.

- Ingång för långsam resistor.
- Ingång för låg resistor exakthet.
- Utgång analog volt.

2.8 I²C (I^2C)

I²C ett protokoll för seriell kommunikation som fungerar med tvåtrådskommunikation, dessa två trådar överför följande information mellan en styrenhet och en eller flera komponenter:

- SDA (Serial Data)
- SCL (Serial Clock)

Kommunikationen sker genom att data skickas till och från styrenheten via en kabel (SDA) kopplad till en eller flera komponenter där ut- och inmatning av bitar synkroniseras med den seriella klockan (SCL). För att starta dataöverföring genereras ett tillstånd där SDA går från hög (1) till låg (0) medans SCL fortfarande är hög. Därefter skickas en serie av bitar som innehåller adressen till den komponent som styrenheten vill kommunicera med följt av en ensam bit (R/W-bit) som specificerar om styrenheten vill skicka data till (logiskt låg signal) eller ta emot (logiskt hög signal) från komponenten. Mottagaren skickar sedan tillbaka en bit (ACK-bit) som bekräftar att anslutningen varit lyckad eller inte. Dataord bestående av 8 bitar skickas sedan till eller från slaven beroende på R/W-biten, efter varje dataord skickas en ACK-bit tillbaka från mottagaren för att bekräfta överföring. Överföringen avslutas genom att ett tillstånd genereras där SDA går från låg till hög medans SCL är hög [10].

2.9 Digital potentiometer

En vanlig axelpotentiometer fungerar som ett motstånd varierbart motstånd. Med en rörlig arm som kan flyttas för hand över en glidbana gjord av exempelvis ett kolskikt kan resistansen mellan två kopplingspunkter eller pinnar ändras [11]. En digital potentiometer fungerar på liknande sett, men hos en sådan ändras resistansen genom en digital styrning. Vanliga kommunikationsprotokoll för styrning av en digital potentiometer är SPI eller I²C [12]. I detta projekt användes en digital potentiometer med en maximal resistans på 10k Ω för simulering av en resistansändring vid metod 3. Den potentiometern styrdes genom I²C (se 2.8) och kunde då genom en koppling av två kopplingskablar ändra resistansen till 255 olika värden mellan ca 0 och 10k Ω .

2.10 Arduino UNO

Arduino UNO är en microcontroller som kan användas till olika elektronikprojekt, den programmeras med en variant av programmeringsspråket C++. Några av komponenterna som finns på en Arduino är analoga och digitala pinnar, LEDs och pinnar för seriell dataöverföring [13]. En Arduino UNO användes i detta projekt på grund

av sin kompatibilitet med I2C som en styrenhet till den digitala potentiometer (se 2.9) vilken användes vid resistanssimuleringen.

3

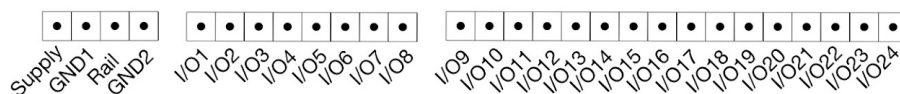
Metod

3.1 Tidsplan och planeringsrapport

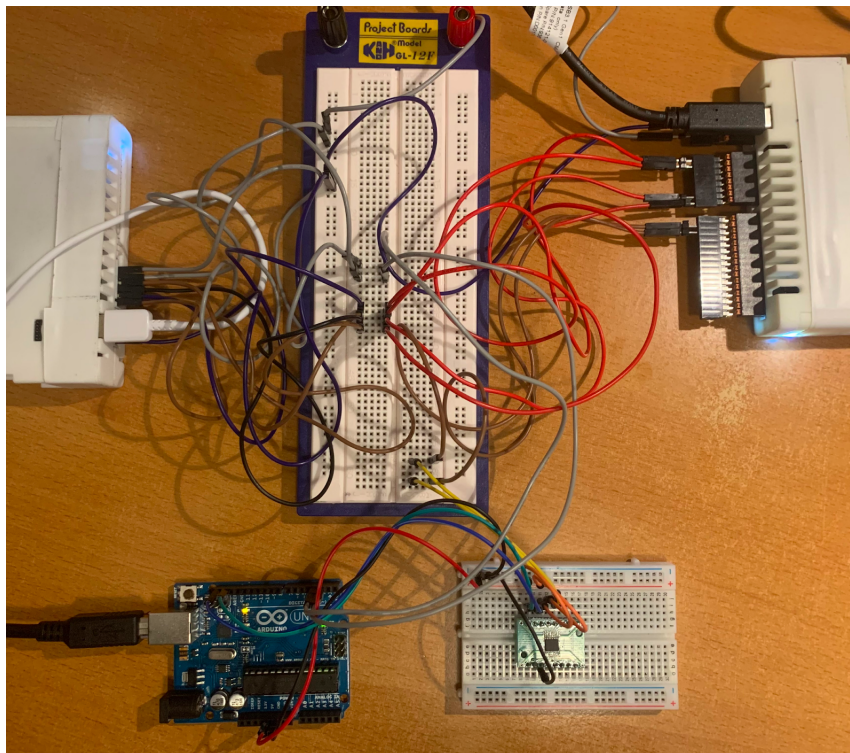
Som inledning till projektet estimerades hur lång tid olika delar av arbetet skulle ta att genomföra. Genom detta togs en planeringsrapport med en tidsplanering för under hur lång tid och i vilken ordning de olika delarna bör genomföras. Flera delmål sattes upp för att kunna kontrollera hur väl projektet låg tidsmässigt. Tidsplaneringen bestämdes veckovis och delar av projektet beräknades att utföras parallellt.

3.2 Konstruktion

Testanläggningen konstruerades genom att använda två PLC01-enheter. I/O portarna på den ena enheten kopplades med färgkodade kopplingskablar till respektive I/O portar på den andra enheten. Samtliga I/O-portar visualiseras nedan (se Figur 3.1). Förutom I/O-portar kopplades även GND1-porten på respektive enhet till varandra med kopplingskablar för gemensam jordning. Även supply-portarna på de båda enheterna kopplades ihop. Detta minskar risken för oönskade avbrott i strömförsörjning om den vanliga strömförsörjningen skulle sluta fungera hos någon av PLC-enheterna. Den enheten med avbrott i sin vanliga strömförsörjning får då strömförsörjning från den speglade enheten. Samtliga I/O-kopplingar mellan enheterna och den gemensamma jordningen kopplades även till en kopplingsplatta för att underlätta felsökning med en multimeter (se figur 3.2). Varje enhet är uppdelad i en primär och en sekundär sida, där varje sida behöver egen strömförsörjning. De två enheterna som bildar testbänken kommer att behöva totalt fyra kablar för att förse samtliga sidor med ström, dessa kan kopplas in i fyra stycken USB-portar för att få 5V matningsspänning till enheterna. Dessa kablar kommer även att kunna användas för seriell anslutning via USB-RS232-converter inbyggt i PLC01.



Figur 3.1: PLC01s I/O portar visualiserat från sidan.



Figur 3.2: Fullständig krets.

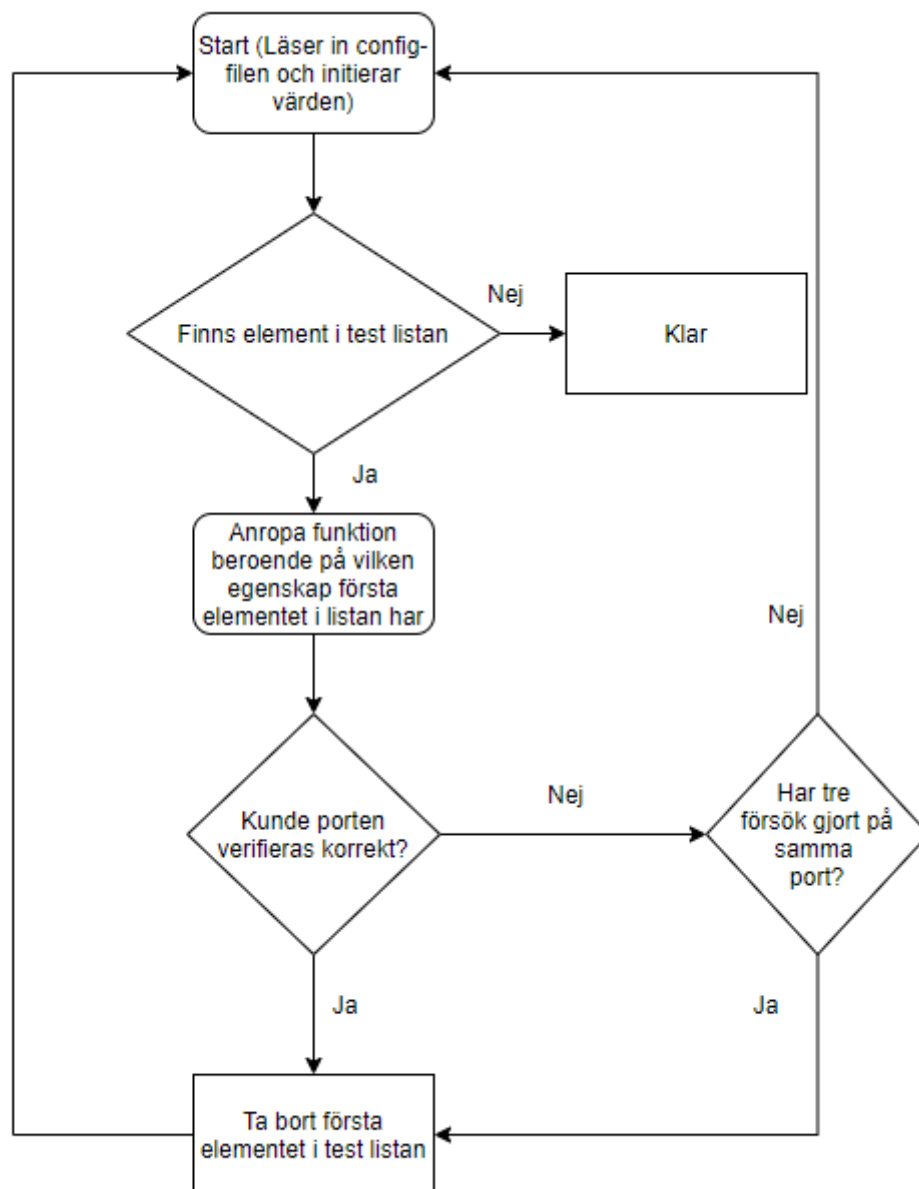
3.3 Konfiguration

Konfigurationen för PLC01 gjordes via canOpen (se 2.7) i en konfigurationsfil i enheten. I konfigurationsfilen väljs bl.a. vilka I/O-portar som ska ingå i vilka grupper, samt vilken mode varje grupp ska ha (t.ex. Input analog volt, output analog volt eller resistansmätning). Olika konfigurationer testades för att säkerställa att skripten kan läsa in och anpassa sig efter specificerade konfigurationer.

3.4 Automatiserad testning

Skript för funktioner i testbänkens mjukvara är skrivna i programmeringsspråket Python 3. Skripten skrevs huvudsakligen i IDE:t PyCharm. Eftersom PLC01 styrs från ett linuxbaserat operativsystemet, användes i skripten python-modulen subprocess, för att exekvera shell-kommandon. Testbänkens mjukvarudel delades huvudsakligen upp i två delar, ett skript för ena PLC01 som utför testet, ett skript som laddas in till ena PLC01 som ska testas och ett skript för den Arduino UNO, som användes som styrenhet för digitala potentiometern. Innan skripten skrevs, konstruerades de önskade processerna i grafiskt i form av flödesscheman nedan (se Figur 3.3). Genom att följa flödesscheman skrevs sedan skripten med python-kod.

Kommunikationen sker mellan PLC01-enheterna genom en koppling av två portar på varje enhet. Vilka portar som ska sköta kommunikationen kan väljas på båda enheterna vid start, men test kommer inte utföras på de valda kommunikationsportar (com-portar). Två extra portar hos den PLC01-enhet som fungerar som testbänk



Figur 3.3: Flödesschema för test av portar.

behövs för att styra arduino. Dessa portar, som också väljs av användaren vid start, kan därför inte spegla portarna hos den PLC01 som testas. Vill man testa samtliga portar behöver skripten köras två gånger med olika kommunikationsportar valda. När kommunikationsportar är valda försöker båda sidor kommunicera med varandra genom handskakning. Handskakningen sker innan varje test och fungerar på följande sätt:

1. Testbänken lägger digital 1:a på com-port-1 och inväntar detsamma på com-port-2.
2. PLC01 inväntar digital 1a: på com-port-1, när detta läses av läggs en 1:a på com-port-2.

3. Testbänken läser av aktiv 1:a på com-port-2 och sätter 0:a på com-port-1.
4. PLC01 noterar att en 0:a ligger på com-port-1 ingång och sätter 0:a på com-port-2.

Felmarginalen sattes till 2% över och under förväntat värde vid mätning av analog spänning och resistans. Vid en felregistrering hos någon av portarna gör testbänken en ny mätning. Om tre felmätningar gjorts efter varandra, utan godkänt resultat vid test av samma port, meddelas att testet inte godkänts och programmet går vidare till test av nästa port (se Figur 3.4). Resultatet av ett utfört test sparas i en textfil på den PLC01 som används som testbänk (se Figur 3.5). Längre ner kan ett fullt test ses från båda sidor (se Figur 3.6 för PLC01 och Figur 3.7 för Testbänk).

```
root@plc-0201138d6b89:~# python3 main.py
Following groups are active: [0, 1, 2, 3, 4]
Following modes are active: [1, 6, 6, 1, 3]
Group for SENDING confirmation: 1
Group for RECEIVING confirmation: 0
[2, 3, 4]
[6, 1, 3]
[001]=0xffff,
[001]=0x0000,
[001]=0xffff,
[001]=0x0000,
[002]=0xffff,
Output confirmed
[001]=0xffff,
[001]=0x0000,
[003]=0.342197,

Input not confirmed, retrying..
[003]=0.346374,

Input not confirmed, retrying..
[003]=0.356224,

Input not confirmed, retrying..
Test timeout for group: 3
Continuing test
[001]=0xffff,
[001]=0x0000,
Reads resistance on group: 4
35.91
[001]=0xffff,
[001]=0x0000,
2460.0
[001]=0xffff,
[001]=0x0000,
4906.0
[001]=0xffff,
[001]=0x0000,
9822.0
[001]=0xffff,
[001]=0x0000,
Resistance measured

Exiting
root@plc-0201138d6b89:~#
```

Figur 3.4: Körning av test där input inte kunde verifieras.

```

root@plc-0201138d6b89:~# cat output.txt
Measured analog input volt: 4.029 V
Sent analog output volt confirmed
Measured resistance 36 Ohm confirmed
Measured resistance 2,5k Ohm confirmed
Measured resistance 5k Ohm confirmed
Measured resistance 10k Ohm confirmed
root@plc-0201138d6b89:~#

```

Figur 3.5: Testfil där testresultat skrivs ut i.

```

root@plc-0201138d6b89:~# python3 main.py
Following groups are active: [0, 1, 2, 3, 4]
Following modes are active: [1, 6, 6, 1, 3]
Group for SENDING confirmation: 1
Group for RECEIVING confirmation: 0
[2, 3, 4]
[6, 1, 3]
[001]=0xffff,
[001]=0x0000,
[001]=0xffff,
[001]=0x0000,
[002]=0xffff,
Output confirmed
[001]=0xffff,
[001]=0x0000,
[001]=0xffff,
[001]=0x0000,
[003]=4.086403,
Input confirmed
Reads resistance on group: 4
35.81
[001]=0xffff,
[001]=0x0000,
2463.0
[001]=0xffff,
[001]=0x0000,
4939.0
[001]=0xffff,
[001]=0x0000,
9809.0
[001]=0xffff,
[001]=0x0000,
Resistance measured

Exiting
root@plc-0201138d6b89:~#

```

Figur 3.6: Körning av test sett från PLC01.

```

root@plc-0201138d6b89:~# python3 test.py
Following groups are active: [0, 1, 2, 3, 4, 5]
Following modes are active: [6, 1, 1, 6, 6, 6]
Group for SENDING confirmation: 0
Group for RECEIVING confirmation: 1
Group for setting pin_1: 4
Group for setting pin_2: 5
[2, 3, 4]
[1, 6, 3]
[000]=0xffff,
[000]=0x0000,
[000]=0xffff,
[000]=0x0000,
[000]=0xffff,
[000]=0x0000,
4.029
Input confirmed
[000]=0xffff,
[000]=0x0000,
[003]=0xffff,
Output confirmed
[004]=0xffff,
[004]=0x0000,
[005]=0xffff,
[004]=0xffff,
[004]=0x0000,
[005]=0x0000,
Resistance measured

Exiting
root@plc-0201138d6b89:~#

```

Figur 3.7: Körning av test sett från testbänk.

3.5 Temperaturmätning

Temperaturmätningen hos PLC01 görs med en resistansbaserad mätning mellan två I/O-portar. Utmätt resistansvärde kan sedan omvandlas till temperatur med matematiska formler. Testen genomfördes därför av testbänken genom att kontrollera att rätt resistansvärde mättes ut av PLC01. Två olika metoder för simulering av temperaturgivare och resistansmätning undersöktes.

3.5.1 Metod 1: Simulerad yttre resistans

Vid test av den första metoden kopplades de två PLC01-enheterna ihop (se Figur 3.8) Här följer de olika portarnas konfigurationer:

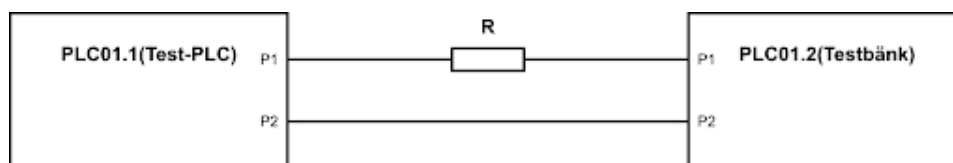
PLC01.1:

- P1: Resistans ingång (par med P2)
- P2: Resistans ingång (par med P1)

PLC01.2

- P1: Analog volt ingång.
- P2: Analog volt utgång.

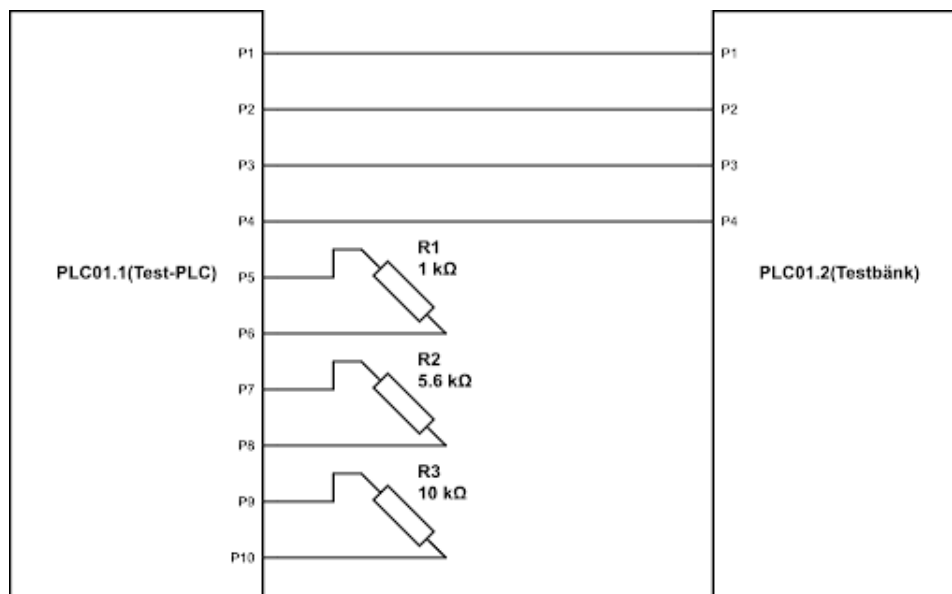
Med denna koppling kan testbänkens spänning läsas av på P1 och sedan skicka ut samma spänning på P2. Om test-PLC ser korrekt R kan då en annan R bli simulerad genom att öka eller sänka spänningen på P2. Detta antas fungera eftersom resistansmätningen använder sig utav ratiometrisk mätning. Spänningen som kommer in på test-PLC:ns P2 kommer tolkas som den spänning den själv lagt ut över sina två portar. I själva verket kommer denna spänning från testbänken. När testbänken höjer eller sänker spänningen kommer test-PLC:n att tolka det som att det yttre motståndet (resistansen) har förändrats på grund utav att den yttre spänningen har ökats eller sänks.



Figur 3.8: Koppling vid resistansmätning metod 1.

3.5.2 Metod 2: Stickprov

Stickprovsmätning är ett säkrare sätt att verifiera resistansmätning i jämförelse med metod 1. I detta fall användes inga yttre spänningar för att manipulera PLC01:s mjukvara. Här kopplas tre styckena olika resistorer upp emot sex olika portar som är konfigurerade för resistansmätning fördelade parvis. Mätning görs och läses av för verifiering. Kopplingen för denna metod illustreras nedan (se Figur 3.9).



Figur 3.9: Koppling vid resistansmätning metod 2.

3.5.3 Metod 3: Digital potentiometer

För att testa resistansmätningen hos PLC01 användes en digital potentiometer (AD5241). Resistansen hos den digitala potentiometer kan genom I2C ändras mellan ca 0 till 10 kOhm. Som styrenhet programmerades en Arduino UNO i C++. Skriptet undersökte kombinationer hos två av arduinons digitala in-portar och skickade beroende på kombination av hög och låg signal på portarna ut data på två andra seriella ut-portar. De digitala portarna kopplades till den ena PLC01-enheten (testbänken) och de seriella ut-portarna kopplades till digitala potentiometern för att kunna styra dess resistansvärde. På så sätt styrdes den digitala potentiometern av arduinon, som i sin tur styrdes av PLC01. Kombinationer av hög och låg signal på de två in-portarna hos arduino kan ses i Tabell 3.1 nedan, vilken data som skickas till digitala potentiometern samt det förväntade resistansvärdet hos potentiometern.

I/O-2	I/O-3	Seriell Data (Hexadecimal)	Resistansvärde (Ω)
0	0	0x00	36
1	0	0x40	2 500
0	1	0x80	5 000
1	1	0xFF	10 000

Tabell 3.1: Resistansförändring beroende på seriell data överföring.

Hos den enhet av PLC01 som testades kopplades två portar till resistansmätning mot två pinar hos digital potentiometern (B1, W1). Det är mellan dessa två portar resistansen förväntas ändras enligt Tabell 3.1. Genom denna koppling, samt de tre skripten, kan testbänken ändra resistansen, vänta på en bekräftelse från den PLC01 som testas att rätt resistans mäts ut, varpå testbänken kan ändra resistansen ännu

en gång. I detta projekt testades fyra olika värden på resistans, där det lägsta samt högsta värdet som kan väljas på den digital potentiometern var inkluderade. Beroende på vilken seriell data man väljer att skicka till potentiometern kan 255 olika resistansvärden sättas, på så sätt kan olika värden väljas för att simulera flera olika resistansbaserade temperaturgivare.

3.6 Verifiering av testbänkens funktion

Vid verifieringen testades att skripten på båda enheter kördes från början till slut utan anmärkningar. Flera tester gjordes där olika värden valdes för att finna buggar eller andra fel i koden. I testet nedan valdes grupp 7 och 6 som kommunikationsgrupper. Efter detta sker en handskakning så de båda skripten körs parallellt och verifierar varandra. Denna handskakning slutförs innan varje mätning av analog spänning (in/ut). Som kan ses nedan (se Figur 3.10) nekades mätningen av analog spänning in två gånger innan rätta värden mättes. Efter detta utfördes mätning av analog spänning ut. Avslutningsvis mättes tre olika resistanser som var inkopplade på sex olika portar parvis (se Figur 3.10).

```
root@plc-0201138d6b89:~# python3 main.py
Following groups are active: [0, 1, 3, 4, 5, 6, 7]
Following modes are active: [1, 6, 3, 3, 3, 1, 6]
Group for SENDING confirmation: 7
Group for RECEIVING confirmation: 6
[0, 1, 3, 4, 5]
[1, 6, 3, 3, 3]
[007]=0xffff,
[007]=0x0000,
[007]=0xffff,
[007]=0x0000,
[000]=0.019462,

Input not confirmed, retrying..
[007]=0xffff,
[007]=0x0000,
[000]=0.019297,

Input not confirmed, retrying..
[007]=0xffff,
[007]=0x0000,
[007]=0xffff,
[007]=0x0000,
[000]=4.084457,

Input confirmed
[007]=0xffff,
[007]=0x0000,
[001]=0xffff,
Output confirmed
Reads resistance on group: 3
10021.2
Resistance measured

Reads resistance on group: 4
5594.50
Resistance measured

Reads resistance on group: 5
1000.53
Resistance measured

Exiting
root@plc-0201138d6b89:~#
```

Figur 3.10: Ett exempel hur testet ser ut under körning.

Testbänken verifierades även genom att medvetna fel implementerades i koden, t.ex. Att spänningen skrevs till maxvärde på en port när testbänken förväntade sig en spänning runt 0V. På så sätt kunde det noteras att testbänken skrev ut en anmärkning på att rätt värde inte mätt. Förändring av värden i spänning, resistans eller ström beroende på MOU kan övervakas i realtid på samtliga portar på PLC01, hur denna övervakning ser ut visas nedan (se Figur 3.11). Detta är användbart vid manuell verifiering av testbänkens funktion.

```
2021-03-06 13:32:29,117
[000]= 4.089050, [001]= 4.089050, [002]= 0.652276, [003]= 10022.796875, [004]= 5594.851562, [005]= 999.200195, [006]= 0.440582, [007]= 0.025630,
[008]= 2.244518, [009]= 0.000692, [010]= 1.910440, [011]= 0.032648, [012]= 4.065376, [013]= 0.000000, [014]= 0.000000, [015]= inf,
[016]= inf, [017]= inf, [018]= inf, [019]= inf, [020]= inf, [021]= inf, [022]= inf, [023]= inf,
[024]= inf, [025]= 0.000000, [026]= 0.000000, [027]= 0.000000, [028]= 0.000000, [029]= 0.000000, [030]= 0.000000,
[031]= 0.000000,
[032]= 0.000000, [033]= 0.000000, [034]= 0.000000, [035]= 0.000000, [036]= 0.000000, [037]= 0.000000, [038]= 0.000000,
[039]= 0.000000,
[040]= 0.000000, [041]= 0.000000, [042]= 0.000000, [043]= 0.000000, [044]= 0.000000, [045]= 0.000000, [046]= 0.000000,
[047]= 0.000000,
[048]= 0.000000, [049]= 0.000000, [050]= 0.000000, [051]= 0.000000, [052]= 0.000000, [053]= 0.000000, [054]= 0.000000,
[055]= 0.000000,
```

Figur 3.11: övervakning av portar på PLC01.

4

Resultat

4.1 Analog spänningsmätning

Vid verifiering av spänning ut och in speglades testbänks PLC01:s portar gentemot den PLC01 som skall testas. Konfigurationen av I/O-portarna gjordes i config.cfg-filen. Test-PLC:n la 4V på dess utgång, detta kunde läsas av och verifieras på testbänken. Vid verifiering av test-PLC:ns ingång lades 4V på testbänken utgång. Detta lästes av som 4V på test-PLC:n. Kommunikationsflagga skickades som möjliggjorde fortsättning av skriptet. Här gjordes flera tester som kunde verifiera att denna metod gav goda resultat (se Figur 3.6 och 3.7).

4.2 Temperaturmätning

För verifiering av temperaturmätning/resistansmätning testades tre olika metoder. Temperaturmätningen genomfördes med goda resultat där både den andra och tredje metoden fungerade bra. Metod 1 kunde inte göra en korrekt verifiering utav resistansen.

4.2.1 Metod 1

Denna metod använde två portar på vadera PLC01 som var kopplade mot varandra med en resistans R mellan det första port paret (se Figur 4.1). Under testning av metod 1 (3.7.1) resulterade manuella ändringen av spänningen i stora fluktuationer och inkonsekventa mätvärden av resistansen. Anledning att resultatet av denna metod avvek från det önskade resultatet beror på hur resistansmätningen sker i PLC01. Mätningen sker ratiometriskt där förhållandet mellan spänningen över den resistans som ska mätas räknas ut jämfört med en intern resistans i PLC01. Spänningen över den interna resistorn är inte bestämd, spänningen över den interna och den önskat mätta externa resistansen bestäms av en reglerade strömkälla. Ett korrekt mätvärde kan därför endast erhållas, om samma ström går genom både den interna och externa resistansen utan något läckage. PLC01 har möjlighet att mäta resistans i ett manuellt och ett automatiskt läge. När mätningen sker automatiskt, väljer PLC:n själv utifrån en algoritm strömmen och man kan då inte veta i förväg vilka spänningar som kommer att användas i ett försök att simulera en ändring av resistans. Sker mätningen istället genom det manuella läget kan istället strömmen väljas av användaren, vid spänningsändring vid den externa resistansen i ett försök att simulera en resistansändring kommer inte längre strömmen genom den interna och den externa

resistorn vara överensstämmande. Detta resulterar i stor osäkerhet i resistansmätningen. I tabellen nedan kan mätvärden ses (se Tabell 4.1). De värden som ses under hexadecimal var de värden som matades in i CanOpen för att kunna lägga/ändra spänning på pinnar, talen kunde skrivas från 0000 upp till ffff. Ur tabellen kan det konstateras att denna koppling ej gav bra resultat. Man kan se hur spänningsnivåerna skiftar drastiskt beroende på vilka hexadecimala tal som skrevs in. Dessa spänningsnivåer förväntas vara mer linjära gentemot de hexadecimala talen. Anledningen till det beteende vi kan se i tabellen nedan är pågrund utav att en spänningsutgång har kopplats till en resistansmätningssingång. Resistansmätningssingången bidrar själv till en spänning på dessa port, eftersom spänning är en potential och är lika överallt där ingen komponent sitter, betyder detta att spänningen också påverkar utgången och därför får vi värden som inte förväntas. De motstånd som mättes upp motsade vår teori om att när spänningen ökar externt kommer PLC01 tolka detta som att resistansen ökas och därför visa ett högre motstånd som svar. Som kan ses i tabellen visas det tvärtom. När spänningen ökar mellan 0.15 och 1.60 så sänks spänningen. Runt 1.80 och över mättes den yttre resistansen att gå mot oändligheten.

Hexadecimal	Spänning (V)	Uppmätt Resistans (Ω)
1600	0.15	1850
1700	0.25	1600
1800	0.40	1400
1900	0.50	1250
2000	1.60	200
2100	1.80	400
2200	1.80	∞
2300	1.90	∞

Tabell 4.1: Mätvärden för Metod 1: Simulerad yttre resistans.

4.2.2 Metod 2

I detta fall användes tre olika resistorer som sattes mellan sex olika portar i en PLC01 (se Figur 4.2). För testning av temperatur mätning användes tre styckena olika resistorer (1k, 5.6k och 10k). Dessa testades på sex olika portar som konfigurerades parvis för resistansmätning. I terminal kunde det läsas följande värden: 1000.53, 5594.50, 10021.2. Dess värden är godkända med avseende på den felmarginalen vi tillåter på 2%. Denna metod var den simplaste där ingen hårdvara behövde utvecklas. Den enda mjukvaran som behövdes var de olika konfigurationerna som gjordes i config.cfg-filen. Det testades även andra varianter av denna metod där fler resistorer användes mellan samma portar. Ett test där två seriekopplade resistorer mättes och ett test där två parallellkopplade resistorer mättes. Även dessa mindre tester gav godtyckliga resultat enligt förväntade värden som räknades ut för hand. Nedan kan de uppmätta värdena ses i en tabell (se Tabell 4.2). Ur tabellen ser man att den uppmätta resistansen stämmer väl överens med den resistorstorlek som kopplades upp mellan de olika portarna.

Resistorstorlek (Ω)	Uppmätt Resistans (Ω)
1k	1000.53
5.6k	5594.50
10k	10021.20
1k + 1.7k	2702.30
1700 // 1700	849.75

Tabell 4.2: Mätvärden för Metod 2: Stickprov.

4.2.3 Metod 3

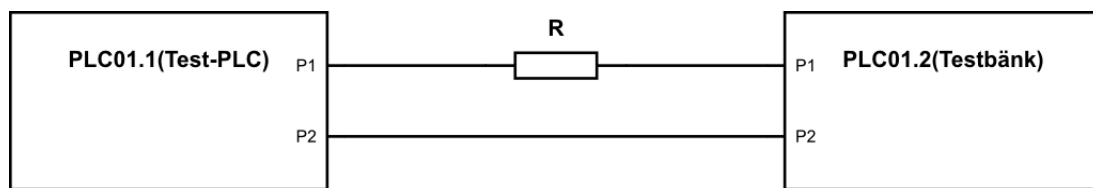
Även i denna metod kunde olika resistansvärden läsas av och godkännas med en felmarginal på 2%. Denna metod skulle kunna jämföras med stickprovstestning då specifika värden väljs ut och mäts. Möjligheterna denna metod medför är däremot större. Med en testning där resistansen ändras digitalt tillåts ett större spektrum av tester. Denna metod skulle klart väljas som den bästa metod när det kommer till skalbarhet och automation. Testbänken kunde genom en tvåsiffrig binär sträng styra Arduino UNO, den binära koden skickades med två kablar som kunde tolkas som 00, 01, 10 och 11. När den binära koden lästs av används den i Arduino UNOs skript. Detta skriptet fungerade som en översättning för I2C kommunikation med den digitala potentiometern. Hela detta flödet skedde med goda resultat. Den digitala potentiometern antog mycket precisa värden och läsningen som test-PLC:n gjorde utfördes utan problem. Kopplingsschema och fullständig krets kan ses nedan (se Figur 4.3). Mätvärden kan ses nedan i tabellen (se Tabell 4.3). Den binära sekvensen är den som skickas från testbänken till Arduino UNO. Efter detta skickas den seriella datan till den digitala potentiometern som tolkar och översätter denna data till ett resistansvärde. Som det kan ses ur tabellen stämmer den uppmätta resistansen med det förväntade resistansvärdet med en godtycklig marginal.

Binär sekvens	Seriell Data	Resistansvärde (Ω)	Uppmätt Resistans (Ω)
00	0x00	36	35.81
10	0x40	2.5k	2463.0
01	0x80	5k	4939.0
11	0xFF	10k	9809.0

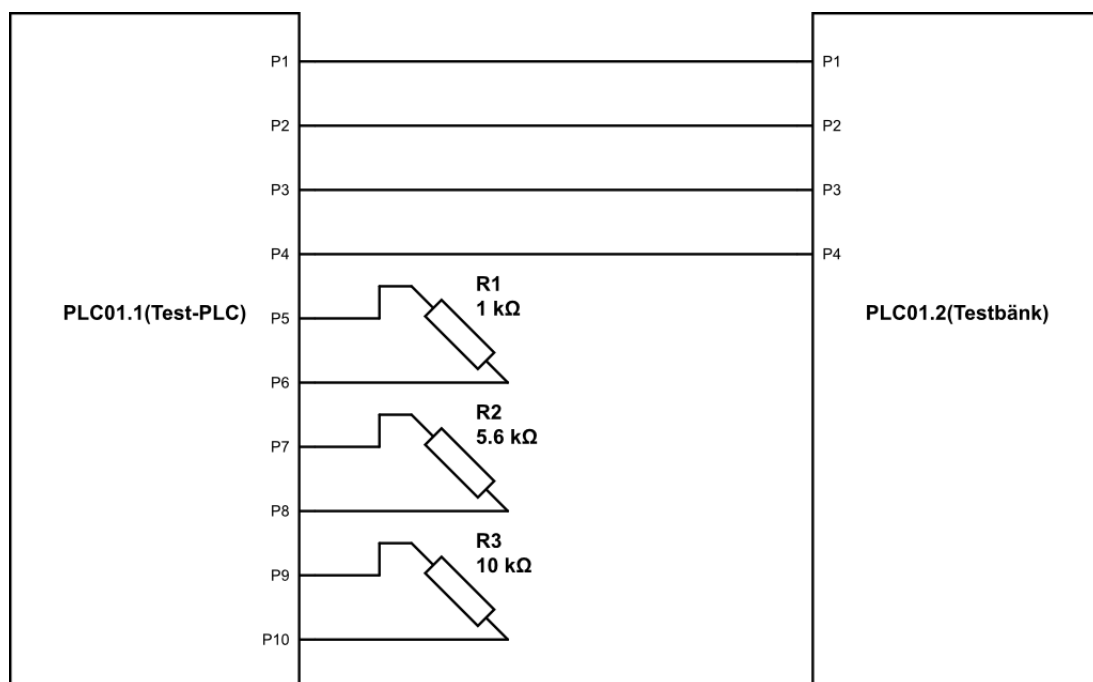
Tabell 4.3: Resistansförändring beroende på seriell data överföring.

4.3 Uppbyggnad

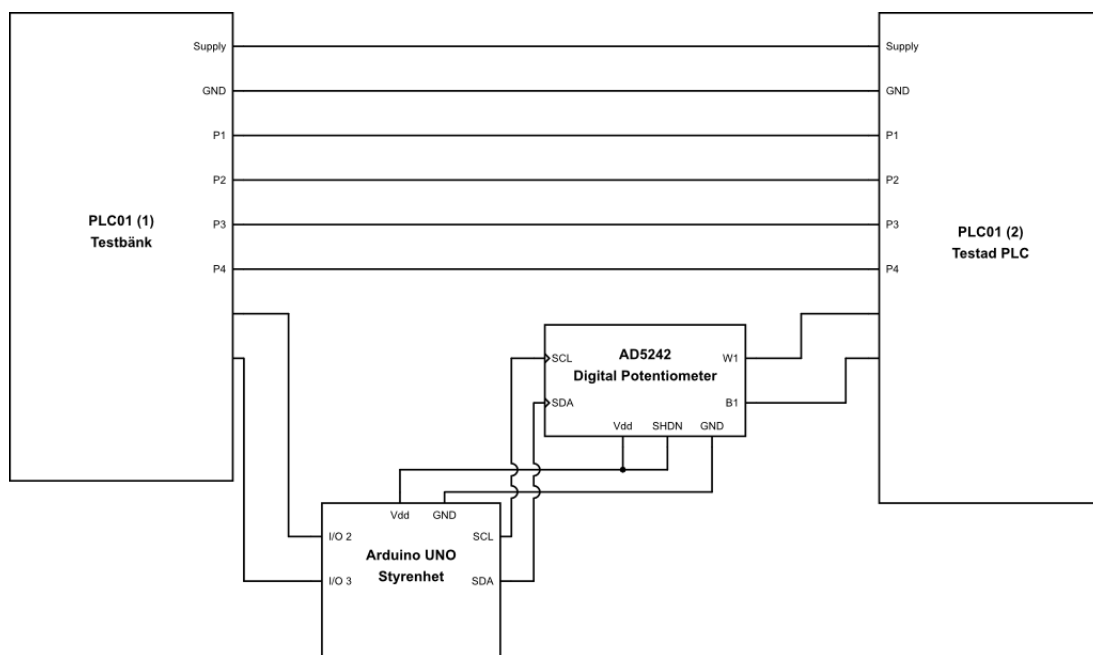
Nedan kan de olika uppbyggnaderna ses.



Figur 4.1: Metod 1: Simulering av resistans.



Figur 4.2: Metod 2: Stickprov.



Figur 4.3: Metod 3: Digital potentiometer

5

Diskussion

Syftet till projektet var att skapa en skalbar grund för en testbänk av PLC01. Metoder för att testa spänning och resistans som tagits fram och implementerats i en lättanvänd testbänk. Testet av spänningsmätning är fullt skalbar eftersom det bygger på en spegling av portar hos de båda enheterna. Genom att koppla flera speglade I/O-portar och konfigurera dessa portar till analog spänningsmätning kan även dessa testas med testbänken. Den valda metoden för resistansmätning gör att även detta test är skalbart. Den potentiometer som valdes, AD5242, har två kanaler. Detta gör det möjligt att utföra test på två par portar hos PLC01 kopplade till samma digitala potentiometer. Potentiometern har möjlighet att anta fyra olika adresser, vilket gör det möjligt att koppla fyra stycken AD5242 till samma I2C-buss. Det gör att resistansmätningen enkelt kan skalas till att testas på flera portar. Genom mindre ändringar i testprogrammet kan även olika nivåer på resistans och spänning samt precision väljas. Det faktum att testbänken använder sig av konfigurationsfilen för att bestämma vilka mätningar som ska utföras på varje grupp av portar underlättar för att lägga till en funktion för mätning av ström till befintlig mjukvara i testbänken.

5.1 Mätning av temperatur/resistans

Testkörning av metod 1 resultera i stora fluktuationer. Anledningen till detta var att förspänningsströmmen och resistans omfånget var satt till automatiskt. Precis som en multimeter hittar PLC01 en lämplig ström och inre resistans gentemot det yttre motståndet. Problemet som uppstod var att när en annan spänning blev injicerat till en mätport ändrades dess förspänningsström och inre motstånd. I det fall då förspänningsström samt resistansomfånget är satt till konstanta värden skulle dess mätvärde variera kraftigt. Även här uppstår problemet att den interna spänningen skulle vara ett direkt resultat av strömmen som går genom den inre ratiometriska referensen. Detta betyder att den interna spänningen inte har någon gemensam nämnare med den yttre, den inre förspänningsströmmen är inte samma som den yttre. I teori skulle en sådan lösning fungera, men i praktiken kommer strömläckage och annat påverka kretsen. Resultatet av detta är felaktigheten i förspänningsströmmen skulle översättas 1:1 felaktighetsmätning utav resistansen. Stickprovsmätning fungerar egentligen på samma sätt som en digital potentiometer där resistansen ändras. Skillnaden på dessa metoder är att den digitala potentiometer erbjuder större möjligheter när det kommer till uppskalning och utbyggnad. Innan någon metod väljs ska det påpekas att det finns stora ekonomiska skillnader i de olika fallen. En

digital potentiometer tillsammans med en Arduino är många gånger dyrare än det simplare stickprovet.

5.2 Utvecklingsmöjligheter

Under projektets gång kunde flera utvecklingsmöjligheter hittas. Dessa utvecklingsmöjligheter är sådana som skulle kunna underlätta för just detta projekt. Andra utvecklingsmöjligheter, som inte gav någon direkt hjälp till uppbyggnad av denna testbänk, tas inte upp nedan.

5.2.1 Inläsning av portar genom config.cfg-filen

Vid inläsning av konfigurationsfilen hittar testprogrammet vilka grupper som är aktiva och vad de har för inställningar. För ett tydligare test skulle också de olika pinnarna som grupperna är kopplade till, skrivas ut på terminalen. Detta underlättar om någon felmätning skulle göras, då den specifika pinnen som felet uppstår på skrivs ut direkt. Nuvarande version skriver endast ut gruppen som felet uppstår på. Denna utveckling hade underlättat just detta arbetet.

5.2.2 Strömmätning

Strömmätning är en funktion som finns möjlig i konfigurationsfilen. Däremot är inte hårdvaran för PLC01 tillräckligt utvecklad än för att kunna göra sådana mätningar. Då dessa funktioner är färdiga att användas kan även dessa implementeras i testbänken för att automatiskt kunna verifiera att den mätta strömmen är korrekt. Precis som i resistansmätning finns det flera olika lösningar till detta. Kanske en av de enklaste är att använda sig utav denna redan befintliga digitala potentiometern och förändra ett motstånd för att sedan se hur spänning och ström påverkas. När dessa variabler är konstaterade kan ström verifieras korrekt.

5.2.3 PLC01 som styrenhet

För den digitala potentiometern behöves en styrenhet som skickar styrsignaler i form av tvåtrådskommunikation. Detta kan förnuvarande inte en PLC01 göra, därav användningen utan Arduino UNO i projektet, men utvecklingen av detta är under arbete. Detta skulle möjliggöra en enklare typ av testtrigg då en extern styrenhet (Arduino, Raspberry Pi eller liknande) kan plockas bort helt. Istället skulle två portar (t.ex. com-portar) direkt kopplas till den digitala potentiometern utan behovet av en mellan hand.

6

Slutsats

En automatiserad testbänk för två PLC01 enheter kunde implementeras med goda resultat. För fullt automatiserad resistansmätning rekommenderas en digital potentiometer. Den digitala potentiometern tillåter förändring i motstånd som sedan kan läsas och verifieras av testbänken. För styrning av en digital potentiometer behövs någon typ utav styrenhet (Arduino, Raspberry Pi eller liknande). Även då detta höjer priset för testtriggen, så utökar det möjligheterna drastiskt. I det fall en billigare och simplare testtrigg önskas så rekommenderas att använda sig utav metod 2. Denna metod använder stickprov som test utav resistansmätning, där olika resistorer sätts mellan port-par och mäter deras motstånd. Python är ett smidigt programmeringsspråk för denna typ utav utveckling, då resultat enkelt kan läsas ut och jämföras mot en specifik felmarginal. Precisionen för spänningar bör arbeta med 0.5%, detta kan möjligtvis sänkas i framtiden om utvecklingen av PLC01 fortsätter. Den utvecklade testtriggen är fullt skalbar och kan utökas om hårdvara finns till hands. Detta gick att bevisa utifrån en mindre testtrigg som utvecklats (se Figur 4.4). Mjukvarans skalbarhet är lika god som hårdvarans. Mjukvaran består till stor del av funktioner som kan köras flera gånger om beroende på hur många portar som ska verifieras.

För framtida studier är utveckling av PLC01 i arbete och kommer medföra flera möjligheter. Vissa av dessa är strömmätning och tvåtrådskommunikation. Då en strömmätning kan göras mellan två portar skulle detta kunna adapteras i testtriggen för automatisk verifiering. I det fall PLC01 uppnår tvåtrådskommunikation kan, den annars vitala, styrenheten plockas bort då en PLC01 hade använts som styrenhet.

Litteraturförteckning

- [1] Jay F. Hooper. INTRODUCTION TO PLCs, Second Edition. Carolina Academic Press; 2nd edition (September 30, 2006) ISBN: 1594603316
- [2] Tony R. Kuphaldt. Logic Signal Voltage Levels. URL: <https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/>. (accessed: 30.05.2021).
- [3] Arun T Vemuri och Matthew Sullivan. Ratiometric measurements in the context of LVDT-sensor signal conditioning. URL: <https://www.ti.com/lit/an/slyt680/slyt680.pdf?ts=1622112759527>. (accessed: 30.05.2021).
- [4] Arc Electronics. RS232 Data Interface a Tutorial on Data Interface and cables. URL: <https://www.arcelect.com/rs232.htm/>. (accessed: 06.08.2021).
- [5] Lammert Bies. RS232 Specifications and standard. URL: <https://www.lammertbies.nl/comm/info/rs-232-specs>. (accessed: 30.05.2021).
- [6] Lars Bengtsson. ELEKTRISKA M ATSYSTEM OCH M ATMETODER. StudentlitteraturAB, 2012. ISBN: 9789144080680.
- [7] Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes. SSH, THE SECURE SHELL: THE DEFINITIVE GUIDE: THE DEFENITIVE GUIDE SECOND EDITION. O'Reilly Media; Second edition (May 31, 2005) ISBN: 0596008953
- [8] Linuxize. How to Use SFTP Command to Transfer Files. URL: <https://linuxize.com/post/how-to-use-linux-sftp-command-to-transfer-files/>. (accessed: 30.05.2021).
- [9] Al Sweigart. AUTOMATE THE BORING STUFF WITH PYTHON: PRACTICAL PROGRAMMING FOR TOTAL BEGINNERS. No Starch Press; 1st edition (May 1, 2015) ISBN: 1593275994
- [10] Scott Campbell. BASICS OF THE I2C COMMUNICATION PROTOCOL. URL: <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>. (accessed:30.05.2021).
- [11] Lars Nordlund och Ingmar Wiklund. GRUNDLÄGGANDE ELEKTRONIK, andra upplagan. Liber, 2012.
- [12] EEPower. Digital Potentiometer. URL: <https://eepower.com/resistor-guide/resistor-types/digital-potentiometer/>. (accessed: 20.09.2021).
- [13] SPARKFUN. What is an Arduino?. URL: <https://learn.sparkfun.com/tutorials/what-is-an-arduino>. (accessed: 20.09.2021).

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS