



CHALMERS

Utveckling av ett automatiserat testverktyg för signaltester

Examensarbete inom högskoleingenjörsprogrammet Mekanik

WILLIAM MUNTHÉR

ENDRIT REXHAJ

INSTITUTIONEN FÖR ELEKTROTEKNIK

CHALMERS TEKNISKA HÖGSKOLA

Göteborg 2026

www.chalmers.se

EXAMENSARBETE I MEKATRONIK

Utveckling av ett automatiserat testverktyg för signaltester

WILLIAM MUNTHE
ENDRIT REXHAJ



CHALMERS

Institutionen för elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2026

Utveckling av ett automatiserat testverktyg för signaltester
WILLIAM MUNTHE
ENDRIT REXHAJ

© WILLIAM MUNTHE, ENDRIT REXHAJ. 2026.

Handledare: Sebastian Rundberg, Init Sweden AB
Examinator: Veronica Olesen, Institutionen för elektroteknik

Examensarbete 2026
Institutionen för elektroteknik
Chalmers tekniska högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Utveckling av ett automatiserat testverktyg för signaltester

WILLIAM MUNTHER

ENDRIT REXHAJ

Institutionen för elektroteknik

Chalmers tekniska högskola

Förord

Detta examensarbete för Mekanikprogrammet, på Chalmers tekniska högskola genomfördes i samarbete med företaget Init Sweden AB på plats i Göteborg.

Vi vill rikta ett stort tack till all personal som har varit till hjälp här på Init och ett speciellt tack till Sebastian Rundberg som ställde upp till att vara vår handledare under projektets gång. Under detta halvår har alla varit till stor hjälp i att svara på frågor samt dela all sin kunskap inom detta arbetsområde. Utan all denna hjälp hade det inte varit möjligt för oss att komma fram till de resultat vi gjorde.

Vi vill även ge ett stort tack till Veronica Olesen för hennes insats som vår examinator samt för all värdefull feedback på rapportens innehåll och struktur.

Sammanfattning

Denna studie behandlar automatisering av signaltester vid driftsättning av industriella PLC-baserade styrsystem. Normalt utförs dessa styrtester helt manuellt i samarbete mellan systemutvecklare och elektriker, vilket är mycket tidskrävande och kan öka risken för mänskliga fel. Behovet av ett mer effektivt tillvägagångssätt identifierades av teknikkonsultbolaget Init, som dagligen arbetar med digital automatisering inom infrastruktur och industri.

Syftet med detta projekt var att utveckla ett automatiserat testverktyg för verifiering av testsignaler vid driftsättning av PLC-baserade industriella styrsystem. Verktyget skulle kontrollera att rätt fysiska insignaler är korrekt kopplade till avsedda ingångar hos PLC:n enligt I/O-lista och kabelförteckning, med målet att minska mängden mänskliga fel och reducera manuellt arbete vid driftsättning.

Arbetet genomfördes genom att utveckla ett verktyg i Python med Modbus TCP som kommunikationsprotokoll. Tester gjordes mot en fysisk Schneider Electric M241 PLC. Konfigurationen av PLC:n gjordes i Machine Expert och en kravspecifikation upprättades som en grund för själva utvecklingen.

Resultaten visar att ett automatiserat testverktyg av just denna typ är fullt möjligt att uppnå. Verktyget läser in testfall från en Excel-fil, kommunicerar med PLC:n via Modbus TCP och skriver sedan tillbaka resultaten, datum och status i samma fil. Samtliga krav i kravspecifikationen lyckades uppfyllas.

Slutsatsen är att det framtagna verktyget lyckades förenkla en tidigare manuell och tidskrävande process till ett smidigare arbetsflöde.

Abstract

This project addresses the automation of signal testing during commissioning of industrial PLC-based control systems. Ordinarily, these tests are performed completely manually in cooperation between electricians and systems developers, which is a process that is time consuming and prone to human error. The need for a much more efficient approach was identified by the engineering consultancy Init, which works daily with digital automation in industry and infrastructure.

The aim of this project was to develop an automated test tool for verification of test signals during commissioning of PLC-based industrial control systems. The tool was designed to verify that the correct physical input signals are properly connected to the PLC's intended inputs, according to the cable schedule and I/O-list with the goal of reducing human error greatly and minimizing manual effort during commissioning.

The project was carried out by developing a tool in Python using Modbus TCP as the communication protocol towards a physical Schneider Electric M241 PLC. Machine Expert was used to configure the PLC, and a requirement specification was established as the foundation for development.

The results demonstrate that an automated test tool of this type is fully achievable. The tool reads test cases from an Excel file, communicates with the PLC via Modbus TCP, and writes back the results, date, and status all to the same file. All requirements defined in the requirements specification were successfully accomplished.

The conclusion is that the developed tool successfully simplified a previously manual and time consuming process into a smoother workflow.

Beteckningar

Nedan följer en lista på de beteckningar som använts under rapporten:

PLC	Programmable Logic Controller.
HMI	Human Machine Interface, ett gränssnitt för interaktion mellan människa och maskin.
SCADA	Supervisory Control and Data Acquisition, ett system för övervakning och styrning för industriella processer.
I/O	Input/Output, står för ingångar och utgångar.
Modbus TCP	Modbus Transmission Control Protocol, ett industriellt kommunikationsprotokoll som använder Ethernet.
SAT	Site Acceptance Test, ett testförande inom automation där funktionen hos ett system verifieras på plats efter installation.

Innehåll

Beteckningar	ix
Figurer	xii
1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Precisering av frågeställning	2
1.4 Avgränsningar	2
2 Teknisk bakgrund	3
2.1 Python	3
2.2 Industriella kommunikationsprotokoll	3
2.2.1 Modbus TCP	4
2.3 Schneider Machine Expert	4
2.4 Tester	4
2.4.1 Site Acceptance Test	5
2.4.2 Signaltester	5
2.4.3 Automatiserade tester	5
3 Metod	6
3.1 Testmiljö och hårdvaruuppsättning	6
3.2 Kommunikationsprotokoll	7
3.3 Utvecklingsmiljö och programmeringsspråk	7
4 Resultat	9
4.1 Testprocessens användarflöde	9
4.2 Systembeskrivning	10
4.3 Inläsning av testfall	11
4.4 Testlogik och utläsning av testfall	12
4.5 Excel-filen	14
5 Diskussion	15
5.1 Begränsningar	16
5.2 Samhällsaspekter och hållbarhet	16
5.3 Fortsatt utveckling av testverktyg	17

Bibliography

19

Figurer

3.1	Hårdvaruuppsättning	6
4.1	Flödesschema av användarinteraktion	9
4.2	Flödesschema av testverktyget	10
4.3	Flödesschema av Excel-inläsning	11
4.4	Flödesschema av testlogik	12
4.5	Flödesschema av Excel-utläsning	13
4.6	Excel innan test	14
4.7	Excel efter genomfört test	14

1

Inledning

I samband med en ökad automatisering av infrastruktur, tillverkningsindustri samt energisektor ökar behovet av automationssystem. I ett större automationssystem kan ett flertal signaler hanteras parallellt, där ett individuellt system kan hantera allt från ett tiotal signaler till mer komplexa system som kan hantera tusentals signaler. Samtidigt som processerna automatiseras utförs tester vid driftsättning manuellt. Automatiserade tester underlättar det monotona arbete som annars behövs göras helt manuellt.

1.1 Bakgrund

Ett av de områden teknikkonsultbolaget Init jobbar med i sin dagliga verksamhet är digital automatisering inom industrin och infrastruktur. Lösningar till industriella styrsystem utvecklas och programmeras med PLC-system (Programmable Logic Controller) för att uppnå de krav kunden har.

Vid driftsättning eller ändringar av dessa styrsystem krävs det omfattande tester av signaler för att säkerställa att systemet fungerar enligt kravspecifikation. Signaltesterna utförs manuellt i samarbete med elektriker samt systemutvecklare. Manuellt utförande av tester är mycket tidskrävande eftersom de testade systemen kan innehålla stora mängder signaler. Det kan därmed potentiellt leda till mänskliga fel och resultera i systemfel. Följaktligen har Init identifierat ett behov av att undersöka möjligheterna för att automatisera och förenkla testandet av signaler till PLC-system. Vilket kan leda till omfattande besparingar i tid och mänskliga fel.

1.2 Syfte

Syftet med detta examensarbete är att utveckla ett automatiserat testverktyg för verifiering av insignaler vid driftsättning av PLC-baserade industriella styrsystem. Verktygets syfte är att kontrollera ifall rätt fysisk insignal är korrekt kopplad till den avsedda ingången i PLC:n, enligt I/O-lista och kabelförteckning. Verktyget har som mål att leda till minskade mänskliga fel och reducerat manuellt arbete vid driftsättning.

1.3 Precisering av frågeställning

För att utveckla ett automatiserat testverktyg för verifiering av signaltester på en PLC:s ingångar har följande kravlista framställts:

- Testverktyget är förväntat att kunna ha ett semi-automatiskt utförande, testerna ska alltså enkelt kunna initieras genom ett enkelt knapptryck från användaren.
- Testverktyget ska uppfylla kravet att kunna läsa in testdata från Excel, vilket innebär att värden från en Excel-fil ska kunna importeras, behandlas och användas i testprocessen.
- Testverktyget ska kunna lagra resultat i samma Excel-fil som tidigare nämnts, för inläsning av testdata, vilket innebär att testresultat ska kunna sparas på ett organiserat sätt.
- Testverktyget ska kunna utföra en automatisk utskrivning av datum i Excel-filen på de tester som godkänds.
- Systemet ska kunna integreras med fysiska PLC-enheter utan användning av någon form av simulation. Alltså sker kommunikationen direkt med fysisk hårdvara vilket ställer krav på en tillförlitlig och stabil dataöverföring.

1.4 Avgränsningar

Arbetet avgränsats till att främst använda PLC-modellen M241 och håller sig till programvaran EcoStruxure Machine Expert vilket är Schneiders PLC utvecklingsmiljö som i sin tur är baserat på CODESYS. Dessutom kommer programvaran Pycharm användas vilket är en integrerad utvecklingsmiljö för programmering i Python.

Arbetet har även avgränsas till att endast verifiera kopplingarna och signalerna till PLC:ns digitala ingångar och inte inkludera testning av själva programvaran. Ännu en avgränsning är att fokuset inte ligger på att utveckla signal-tester på en industriell skala utan att ta fram och demonstrera testerna för en automatiserad verifiering. Till sist kommer arbetet inte att omfatta områden inom HMI samt SCADA-relaterade system.

2

Teknisk bakgrund

För att kunna förstå de verktyg, teknologier och grundläggande begrepp som används i arbetet ges här en översikt av väsentliga begrepp och system som ligger till grund för det praktiska genomförandet.

2.1 Python

Python är ett programmeringsspråk med öppen källkod, vilket innebär att källkoden är fritt tillgänglig för alla att läsa och använda [1]. Det är ett mycket vanligt språk för nybörjare och lätt att använda eftersom att syntaxen är enkel och lättläst. Dessutom får användaren tydliga felmeddelanden om vad som är fel, vilket underlättar felsökning av kod.

Python är även ett objektorienterat språk som innebär att man bygger sitt program i objekt [1]. Ett objekt är något som samlar både funktioner och data. Objekt representerar ofta något verkligt som till exempel en fil eller sensor.

I Python finns det flera olika utvecklingsmiljöer, ett stort standardbibliotek, och flera tredjepartsbibliotek som alltså är bibliotek som inte ingår i Pythons programmeringsspråk utan är skapat av andra organisationer eller utvecklare [2]. Dessa används inom många olika områden, till exempel programmering av mikrokontroller, maskininlärning (verktyg inom AI), textbehandling och mycket mer.

2.2 Industriella kommunikationsprotokoll

Industriella kommunikationsprotokoll är de regler som styr hur data struktureras, skickas och hur olika enheter adresseras mellan maskiner [3]. Utan gemensamma protokoll skulle enheter från olika tillverkare inte kunna byta information med varandra. Industriell kommunikation möjliggör flexibilitet och effektivitet inom industriella tillämpningar. Via realtidsöverföring av data kan industriella processer starkt förbättras samt kan mängden uppehåll minska och vid förändrade behov kan maskiner snabbt anpassas [3].

Fältbussar är ett samlingsbegrepp för protokoll som används för kommunikation nära sensorer och aktuatorer, vilket omfattar både modernare Ethernet-baserade lösningar och andra seriella standarder [4]. Olika applikationer ställer olika krav på

kommunikationen. För elkraftsystem är det viktigt att tidsstämplingar på händelser i systemet är korrekta för att kunna följa ett händelseförlopp. Andra system kan prioritera andra aspekter beroende på ändamål.

2.2.1 Modbus TCP

Modbus är ett industriellt kommunikationsprotokoll [5]. Protokollet bygger på klient-server-arkitektur, detta betyder alltså att en enhet skickar förfrågningar och en annan enhet svarar med den efterfrågade datan [5]. Modbus är plattformsoberoende då den inte är begränsad till någon specifik hårdvara eller tillverkare [5].

Modbus används för att läsa och skriva data i form av register, detta gör det möjligt för överföring av exempelvis mätvärden och styr signaler inom enheter som till exempel en PLC eller sensorer [5]. Det finns olika register beroende på ändamål. En typ av register som används för att skriva och läsa data är holding register. Holding register är ett 16-bitars skriv/läs register som finns i Modbus protokollet. Det används för att kunna skriva och läsa data mellan Modbus klienten och servern.

Modbus TCP är en Ethernet-baserad variant av Modbus som möjliggör kommunikation över TCP/IP-nätverk via port 502, vilket gör den passande för moderna industriella system [5].

2.3 Schneider Machine Expert

Schneider Machine Expert, också kallat EcoStruxure Machine Expert, är en integrerad utvecklingsmiljö från företaget Schneider Electric som används för programmering samt konfigurering av PLC:er [6]. Programvaran stöder IEC 61131-3 vilket är en internationell standard som definierar programmeringsspråk och programmeringsmetoder åt PLC:er, och möjliggör utveckling av styrlogik med hjälp av etablerade PLC-språk som Ladder diagram, FBD (function block diagram), och ST (structured text) [6].

2.4 Tester

Testning är en väsentlig del av systemutveckling och syftar till att kontrollera att ett system uppfyller krav som ställs [7]. Beroende på vad som testas kan tester delas in i olika nivåer. Från enhetstester som verifierar enskilda komponenter, till integrationstester som kontrollerar att samspelet mellan olika komponenter fungerar korrekt, och till slut systemtester som verifierar hela systemet som en helhet [7].

Inom industriella och tekniska system är testning en väsentlig del av verifiering och valideringsprocessen, här är syftet att säkerställa att systemets beteende överensstämmer med de förväntade värdena [8]. Automatiserade tester är extra gynnsamma i system som kräver hög tillförlitlighet eller en större mängd upprepade tester, eftersom de möjliggör en systematisk och strukturerad kontroll av funktionalitet.

2.4.1 Site Acceptance Test

Site Acceptance Test, förkortat SAT, är ett acceptanstest som utförs på plats efter att ett system har installerats i sin aktuella driftsmiljö. Syftet med denna process är att verifiera att systemet uppfyller de krav som ställts av beställaren och att det fungerar korrekt under verkliga förhållanden innan det tas i drift [9].

2.4.2 Signaltester

Signaltester syftar till att verifiera att kommunikationen mellan systemets komponenter fungerar korrekt och är en del av SAT [9]. Detta innebär att kontrollera att de förväntade signalerna tas emot och skickas med rätt värden, vilket säkerställer att systemet reagerar korrekt på olika insignaler.

2.4.3 Automatiserade tester

Automatiserade tester är något som används för att verifiera att ett visst system fungerar enligt specifika krav och detta testas då med hjälp av programkod istället för manuellt [10]. Testerna bygger på jämförelser mellan förväntade och de faktiska värdena. Detta möjliggör repeterbar och effektiv testning som bidrar till en minskad risk för mänskliga fel och en förbättrad kvalitet i testprocessen.

3

Metod

I detta avsnitt kommer metoden av arbetet att förklaras. Det kommer redogöras för de olika delar som ingår i testverktyget och vilken betydelse de har för att testverktyget ska fungera enligt plan.

3.1 Testmiljö och hårdvaruuppsättning

Den testmiljö som satts upp vid arbetets start, se figur 3.1, omfattas av en PLC av fabrikatet Schneider Electric M241 vilken tillhandahålls av Init för att testa det framtagna verktyget. Den matas med 24 V DC av ett nätaggregat från Siemens SITOP PSU100L som gör om 230 V AC till 24 V DC.

För att simulera inkoppling av kablar så förses PLC:n med antingen 24 V eller 0 V på samtliga ingångar. Detta tillåter testverktyget att verifiera ifall rätt ingång sätts på eller av, vilket efterliknar hur kablar kopplas in i PLC:n vid verkliga tester.

En Ethernet-switch från TP-link används för att möjliggöra uppkoppling till flera enheter samtidigt. Det går således att hantera flera nödvändiga programvaror parallellt i utvecklingsfasen av testverktyget. För körning av testerna användes en Windows-baserad PC under arbetet.



Figur 3.1: Hårdvaruuppsättning

3.2 Kommunikationsprotokoll

I detta projekt valdes Modbus TCP som kommunikationsprotokoll eftersom protokollet tillför god tillgänglighet samt enkelhet som stöd i testmiljön. Protokollet ger ett gott stöd inom industriell automation och stöds direkt via den använda PLC-plattformen. Det direkta stödet av Modbus TCP möjliggjorde en god kommunikation utan att vara beroende utav speciallösningar eller någon form av extern hårdvara.

Modbus TCP är plattformsoberoende, vilket betyder att kommunikation mellan PLC och externa testverktyg kan etableras totalt oberoende av andra faktorer såsom leverantörer eller operativsystem. Detta var då särskilt fördelaktigt i detta projekt där testerna genomfördes från en PC-baserad testmiljö.

En ytterligare anledning till att Modbus TCP valdes var den enkla konfigurationen. I praktiken kräver endast kommunikationen en Ethernet-anslutning och en IP-adress, vilket gjorde det möjligt att snabbt kunna sätta upp och ändra testmiljön. Detta förenklade då felsökning och testning av kommunikationskopplingar och ingångar, vilket låg i linje med arbetets mål.

3.3 Utvecklingsmiljö och programmeringsspråk

Konfiguration av PLC:n gjordes i Schneider Electrics programvara Machine Expert. Först sattes kommunikation via Modbus TCP på för PLC:n. Därefter behövdes adresserna till de digitala ingångarna skrivas över till PLC:ns minnesregister så att verktyget kan läsa av ingångarnas tillstånd via Modbus TCP.

Det programmeringsspråk som valdes för att utveckla testverktyget var Python. En anledning till detta var att Python är öppen källkod och gratis att använda utan licenser. En annan faktor till att testverktyget utvecklades i Python var att programmet körs via ett kommandofönster vilket tillåter användaren att få kontroll över de automatiserade testerna samt få feedback på hur det går. Kommandofönstret ger därmed verktyget ett enkelt användargränssnitt.

Ett ytterligare skäl till att Python används är det breda urval av bibliotek som finns att använda. Nedan nämns de bibliotek som används i projektet:

- **Openpyxl** används vid utvecklandet av testverktyget för att kunna läsa av vilka testfall som ska göras och därefter skriva resultatet i samma Excel-fil efter genomfört test.
- **PyModbusTCP** används för att möjliggöra kommunikation mellan PLC och testverktyget via Modbus TCP. Biblioteket tillåter inläsning av de digitala ingångarnas tillstånd som skrevs över till minnesregistret.

3. Metod

- **Datetime** används för att få datum på när en ingång godkänns av testverktyget i Excel-filen.

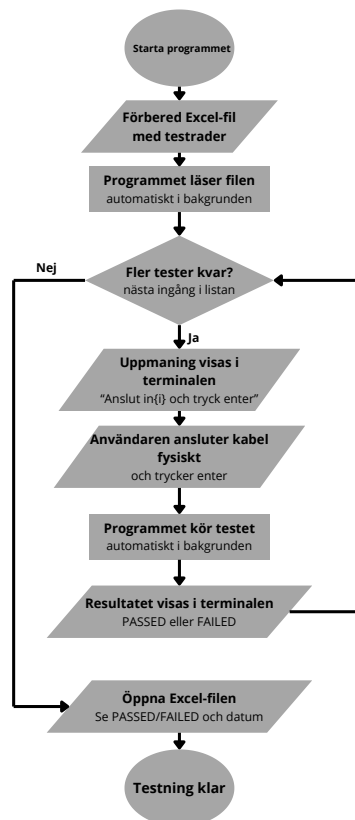
4

Resultat

I det här avsnittet kommer resultatet av det automatiserade testverktyget att presenteras. Först kommer ett flödesschema som förklarar användarinteraktionen med verktyget. Sedan kommer fyra flödesscheman som förklarar uppbyggnaden av testverktygets kod att redogöras.

4.1 Testprocessens användarflöde

Flödesschemat i figur 4.1 illustrerar testprocessens användarinteraktion. Processen startar med att användaren förberett en Excel-fil innan körning av programmet. Varje rad i filen motsvarar ett testfall med en ingång. Programmet läser automatiskt in filen i bakgrunden. Så länge det finns testfall kvar i listan uppmanas användaren i terminalen att förbereda nästa aktuella signal.

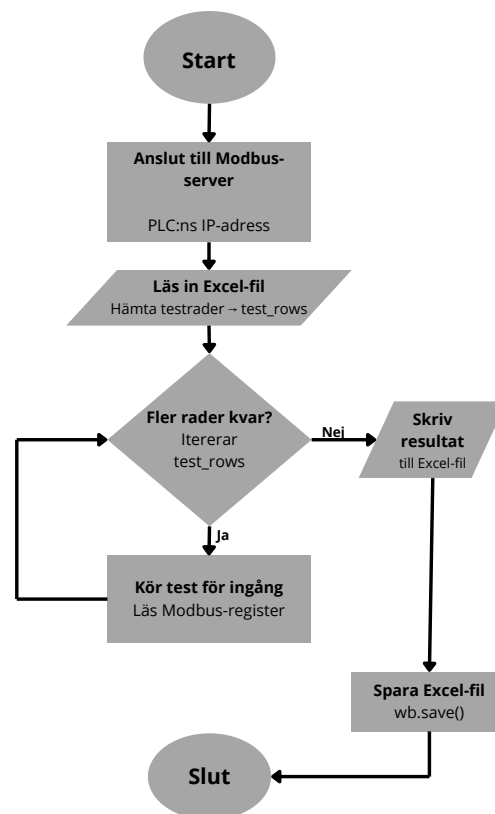


Figur 4.1: Flödesschema av användarinteraktion

För varje ingång i listan ansluter användaren in en fysisk kabel och verifierar anslutningen via att trycka enter. Därefter körs testet automatiskt av programmet och visar resultaten som antingen "PASSED" eller "FAILED". När samtliga tester har utförts sparas resultaten i Excel-filen.

4.2 Systembeskrivning

Koden för testverktyget delas upp i tre huvuddelar: inläsning av testfall, testförfarande och protokollföreläggning av resultat. Samtliga delar behövs för kunna uppnå de krav som sattes i början av arbetet på testverktyget. Figur 4.2 nedan visar ett flödesschema av hela programmet.



Figur 4.2: Flödesschema av testverktyget

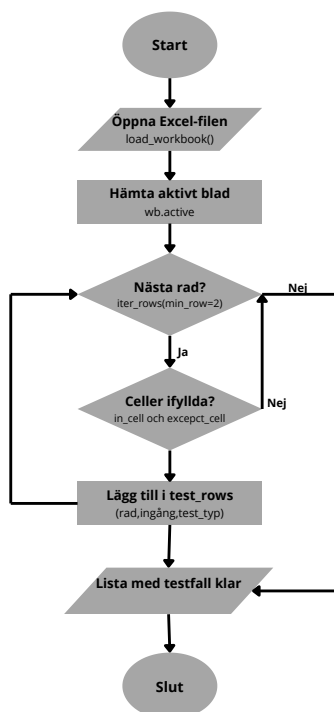
När programmet startas ansluter den till PLC:n via Modbus TCP på PLC:ns statiska IP-adress och den port som användaren skriver in i koden innan start. Sedan läses Excel-filen med testfallen. Varje rad i filen innehåller information om vilken ingång som testas samt ifall den förväntas vara av eller på.

För varje testrad så läser programmet Modbus holding register som innehåller statusen på ingången och jämför med det förväntade värdet ifall de överensstämmer. När alla testfallen genomförts skrivs dessa till Excel-filen med "PASSED" eller "FAILED", samt datum vid godkänt av testad ingång.

4.3 Inläsning av testfall

Figur 4.3 nedan visar det framtagna flödesschemat för inläsning samt bearbetning av data från Excel-filen. Syftet med denna process är att extrahera väsentliga rader och lagra dem i en lista för vidare användning. Processen inleds först med att Excel-filen öppnas och därefter hämtas det aktiva databladet, därpå går algoritmen genom bladet radvis med start från andra raden.

För varje rad som programmet går igenom i bladet så plockas två specifika celler ut, en för ingångens namn och en annan med det förväntade värdet. Varje rad kontrolleras av programmet att de innehåller något.

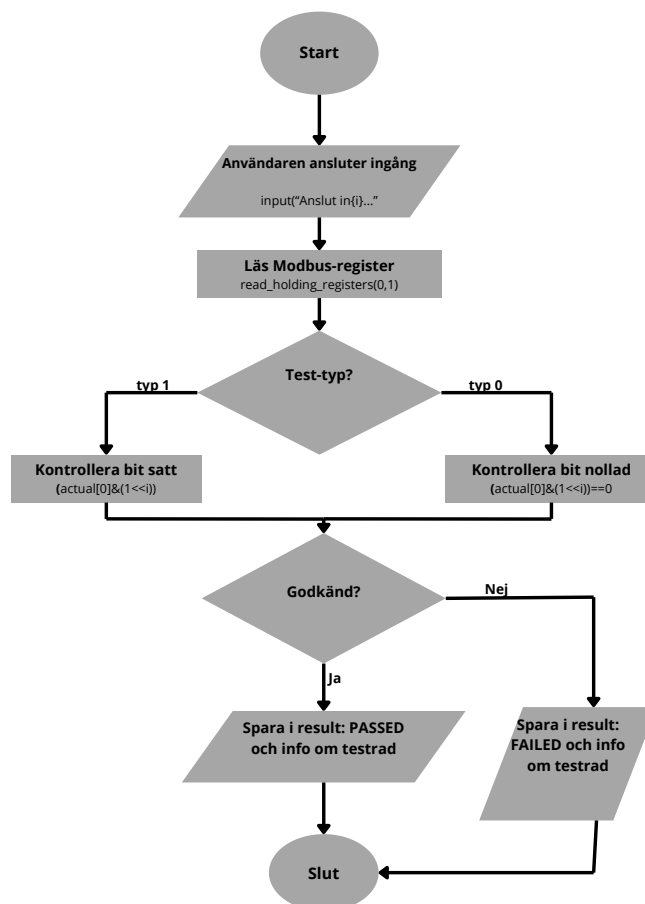


Figur 4.3: Flödesschema av Excel-inläsning

När en rad uppfyller kriterierna läggs den in i listan för det som ska testas tillsammans med ingångens namn och vilken testtyp det är. Om villkoren inte uppfylls fortsätter iterationen till nästa rad utan att det sker någon lagring. Processen upprepas tills alla rader har behandlats. Till slut är en lista klar för användning, vilken innehåller samtliga rader från Excel-bladet som ska testas.

4.4 Testlogik och utläsning av testfall

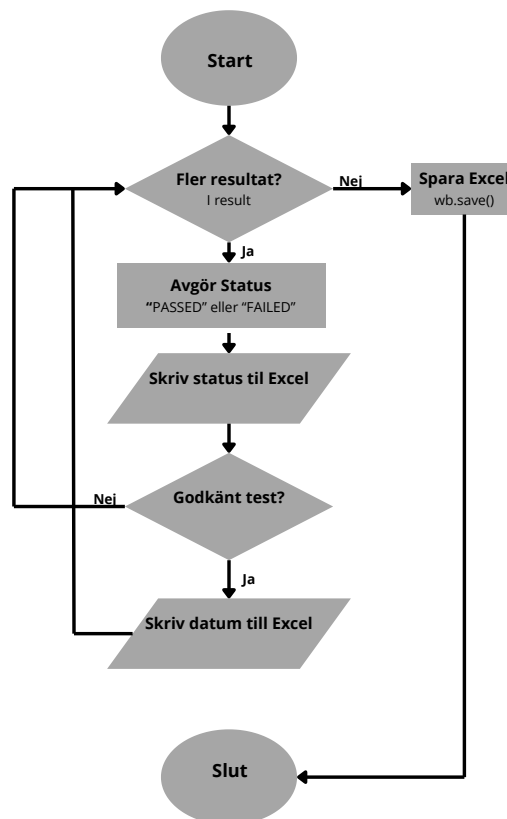
Flödesschemat i figur 4.4 visar den logik som ligger till grund för verifieringen av PLC:ns digitala ingångar som läses via Modbus-kommunikation. Denna process sker för varje enskild ingång som ska testas. Processen inleds med att användaren instrueras att manuellt ansluta den aktuella ingången. Med hjälp av en Modbusfunktion läser programmet det unsigned 16 bitars tal som motsvarar ingångarna på PLC:n. Baserat på det förväntade värdet i testfallet avgör programmet vilken typ av bitkontroll som ska utföras. Antingen kontroll av en bit som är satt, alltså typ "1", eller av, alltså typ "0".



Figur 4.4: Flödesschema av testlogik

Beroende på det testvärde som är förväntat kontrolleras antingen att motsvarande bit är satt eller nollad. Detta genomförs genom att programmet använder sig av en bitvis AND-operation mellan registervärdet och en bitmask. Därefter sparas resultatet som "PASSED" ifall rätt ingång är satt eller "FAILED" vid avvikelse.

I figuren 4.5 nedan visas processen för testresultatets bearbetning och export till Excel. Listan innehåller element med de olika testfallens resultat.



Figur 4.5: Flödesschema av Excel-utläsning

För varje varv kontrolleras ett element i listan och avgör testets status, vilket visas som antingen "PASSED" eller "FAILED". Därefter skrivs denna status in till Excel-filen i den angivna kolumnen för status. Därefter kontrolleras det om testet är godkänt. Ifall villkoret uppfylls så skrivs datumet vid testtillfället till rätt cell i Excel-bladet. När samtliga testfall har bearbetats sparas Excel-filen och processen avslutas.

4.5 Excel-filen

Innan användaren kan starta igång programmet måste Excel-filen med nödvändig information skrivas in för de testfall som ska utföras. Hur detta kan se ut visas i figur 4.6 nedan.

	A	B	C	D	E	F	G	H
1	funktion	SCADA(variabelnamn)	ingång	förväntat värde	Pass/fail	datum	sign	anmärkning
2	UTLÖST TC01/24DC UPS5206.KA1	GE_5000_5206_UPS5206_TC01_SA_AL	in7		1			
3	UTLÖST TC02/24DC UPS5206.KA1	GE_5000_5206_UPS5206_TC02_SA_AL	in1		0			
4	SÄKRING 24VDC UPS5206.KA1	GE_5000_5206_99_UPS5206_FC01_UTL_SA	in0		1			
5	UPS1 SUMMALARM	GE_5000_5206_99_UPS1_SUM_AL	in5		0			

Figur 4.6: Excel innan test

I kolumn A så skrivs den funktion som ingången har i systemet och i kolumn B skrivs tillhörande variabelnamnet i SCADA-systemet. Dessa två kolumner läses inte in av programmet utan finns med för att användaren ska få en tydlig dokumentation på vad som testas. I kolumn C skrivs vilken ingång som ska testas i det formatet som visas i figur 4.6. Därefter skriver användaren in i kolumn D ifall ingången antingen förväntas vara "1" för på och "0" för av vid testtillfället. Kolumn E och F ges efter kört program. Då ges status på ifall respektive ingång stämde överens med det förväntade värdet och vid "PASSED" så fås även datum på när testet genomfördes.

Efter att testverktyget körts så fylls varje rad med "PASSED" eller "FAILED" beroende på ifall det förväntade värdet motsvarade det som lästes av programmet, se figur 4.7. Vid godkänd ingång skrivs dessutom datum vid testtillfället. Kolumn G och H lämnas tomma för kommentering av testerna efter de genomförts.

	A	B	C	D	E	F	G	H
1	funktion	SCADA(variabelnamn)	ingång	förväntat värde	Pass/fail	datum	sign	anmärkning
2	UTLÖST TC01/24DC UPS5206.KA1	GE_5000_5206_99_UPS5206_TC01_SA_AL	in7		1 PASSED	2026-04-21		
3	UTLÖST TC02/24DC UPS5206.KA1	GE_5000_5206_99_UPS5206_TC02_SA_AL	in1		0 FAILED			
4	SÄKRING 24VDC UPS5206.KA1	GE_5000_5206_99_UPS5206_FC01_UTL_SA	in0		1 FAILED			
5	UPS1 SUMMALARM	GE_5000_5206_99_UPS1_SUM_AL	in5		0 PASSED	2026-04-21		

Figur 4.7: Excel efter genomfört test

5

Diskussion

Under examensarbetet utvecklades ett automatiskt testverktyg för att kunna förenkla verifiering av signaler till PLC-system. Arbetet inleddes först med att en kravspecifikation framställdes som gav ett tydligt underlag för utvecklingsprocessen i projektet. Verktøget utvecklades genom att använda programmeringsspråket Python tillsammans med Machine Expert, och Modbus TCP användes som kommunikationsprotokoll.

Syftet var att undersöka ifall ett automatiserat testverktyg för verifiering av signaler via ett PLC-baserat styrsystem gick att genomföra och resultatet visar att detta är fullt möjligt. Verktøget läser in en Excel-fil med flera testfall, kör testerna mot en PLC, sen skriver tillbaka resultaten i samma Excel-fil när testet är avslutat. Användaren styr genom att tabba sig vidare mellan testerna, vilket gör att systemet fungerar semiautomatiskt istället för som tidigare helt manuellt. Detta gör att både syftet och frågeställningens krav kan anses uppnådda.

Testverktyget har dock vissa begränsningar som lämnar utrymme för vidare utveckling. Verktøget är begränsat till kommunikation endast via Modbus TCP och stödjer därmed inte andra protokoll. Vidare läser verktøget endast ett register, vilket innebär att ett PLC-system med större antal ingångar skulle kräva eventuellt flera anpassningar. Dessutom måste digitala ingångar skrivas över till minnesregistret för att verktøget skulle kunna läsa dem, vilket begränsar kompatibiliteten med andra typer av PLC-modeller.

Efter att testverktyget utvecklades gjordes en rad olika tester med handledare där det validerades att det framtagna verktøget förenklade den tidigare testprocessen. Det är enkelt att sätta upp och använda vilket är ett avgörande attribut för praktiskt användande i fält. Testning direkt mot en fysisk PLC var dessutom mycket värdefullt då det möjliggjorde verifiering i en verklig miljö samt en snabb identifiering av eventuella kommunikationsfel.

5.1 Begränsningar

Trots att arbetet lyckats uppnå de förväntningar som ställdes vid arbetets start finns det vissa begränsningar som bör redovisas. Den första begränsningen hos testverktyget är antalet ingångar som kan testas. I detta arbete har endast testverktyget utvecklats till att läsa ett register via Modbus TCP, då PLC:n som verktyget testades mot under arbetet endast hade 14 ingångar. Testverktyget kan därmed inte vid mer omfattande system användas som planerat. En annan begränsning som finns är att verktyget endast kan kommunicera via Modbus TCP. Detta leder till att PLC-system som saknar stöd för Modbus inte kan testas med verktyget.

Därtill finns det en begränsning vid konfiguration av PLC:n i Machine Expert. Det förväntas att PLC:ns ingångar skrivits över till minnesregister för att testverktyget ska kunna läsa statusen på ingångarna. Detta kan se olika ut beroende på PLC-fabrikat och därmed begränsa verktygets möjlighet till att vara fabriksberoende.

5.2 Samhällsaspekter och hållbarhet

Automatiserade tester bidrar generellt till en förbättrad arbetsmiljö genom att minska den mentala belastning som repetitiva och manuella testprocesser medför. Genom att automatisera upprepade moment kommer också tid och mental kapacitet att frigöras för systemutvecklare och elektriker.

Ur ett samhällsperspektiv är det även viktigt att reflektera kring vilka konsekvenser automatisering kan medföra. I takt med att flera arbetsmoment automatiseras medförs det en risk att behovet av manuell arbetskraft minskar, vilket kan leda till negativ påverkan på arbetsmöjligheter.

Ur ett säkerhetsbaserat perspektiv bidrar en väl strukturerad driftsättningsprocess till att minimera risken för felkonfigurationer. Felaktigt inkopplade ingångar kan utgöra en risk för hårdvara och person, då elektriska fel kan orsaka kortslutning eller överspänning.

5.3 Fortsatt utveckling av testverktyg

Samtliga begränsningar som nämns ovan kan användas som utgångspunkt vid fortsatt utveckling av testverktyget. Det finns ett par rekommenderade områden hos testverktyget som hade gjort det mer effektivt. En idé hade varit att försöka utveckla testverktyget till att kunna klara av tester med ett stort antal ingångar. Detta hade kunnat uppnås genom att skriva de specifika Modbus-adresserna till ingångarna i Excel-filen och att testverktyget hade lyckats testa dessa. Detta skulle därmed ersätta den formatering som gjordes i detta arbetet, vilket var att referera till ingångsnumren på PLC:n. På så sätt hade det möjliggjort att verktyget blir mer effektivt då tester av PLC:er med ett stort antal ingångar hade kunnat testats.

En annan idé för vidareutveckling av verktyget hade varit att utöka stödet för andra kommunikationsprotokoll, för att göra verktyget mer generellt och stödja PLC:er som inte använder sig av Modbus TCP.

Litteraturförteckning

- [1] Python Software Foundation, “What is Python? Executive Summary” [Online]. Tillgänglig: <https://www.python.org/doc/essays/blurb/>. [Hämtad: 25-Mar-2026].
- [2] R. Mastrodomenico, “Introduction,” in *The Python Book*. Hoboken, NJ, USA: John Wiley & Sons, 2022. [Online]. Tillgänglig: <https://onlinelibrary.wiley.com/doi/10.1002/9781119573364.ch1>. [Hämtad: 25-Mar-2026].
- [3] Robustel, “What is Industrial Communication?,” [Online]. Tillgänglig: <https://robustel.com/what-is-industrial-communication/>. [Hämtad: 03-Apr-2026]
- [4] Hilscher, “Fieldbus.” [Online]. Tillgänglig: <https://www.hilscher.com/service-support/glossary/fieldbus>. [Hämtad: 03-Apr-2026].
- [5] Modbus Organization, “Modbus Specifications.” [Online]. Tillgänglig: <https://www.modbus.org/modbus-specifications>. [Hämtad: 03-Apr-2026].
- [6] Schneider Electric, “EcoStruxure Machine Expert - Programming Documentation.” [Online]. Tillgänglig: https://product-help.se.com/ESME/2.5/VLP_Software/VLP_Programming/en-US/SoMProg. [Hämtad: 07-Apr-2026].
- [7] M.A. Umar “A Study of Software Testing Categories, Levels, Techniques, and Types,”TechRxiv, 2020. [Online]. Tillgänglig: <https://www.techrxiv.org/doi/full/10.36227/techrxiv.12578714.v2> [Hämtad: 19-Maj-2026].
- [8] IBM, “Software Testing.” [Online]. Tillgänglig: <https://www.ibm.com/think/topics/software-testing>. [Hämtad: 08-Apr-2026].
- [9] S3 Process, “Site Acceptance Testing Explained: Documentation & Examples.” S3 Process. [Online]. Tillgänglig: <https://www.s3process.co>.

uk/site-acceptance-testing-explained-documentation-examples/
[Hämtad: 19-Maj-2026].

- [10] Wikipedia contributors, “Test automation.” Wikipedia The Free Encyclopedia. [Online]. Tillgänglig: https://en.wikipedia.org/wiki/Test_automation. [Hämtad: 08-Apr-2026].

INSTITUTIONEN FÖR ELEKTROTEKNIK
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige
www.chalmers.se



CHALMERS