PLC - introduktion

Allmänt

PLC är engelska och betyder Programmable Logical Controller. Detta är en speciell typ av dator / processor / CPU som är anpassad för styrning i industriella miljöer. Före 1980-talet kallade man PLC för PC. I och med persondatorns genombrott på 1980-talet fick man ge vika för begreppet PC och i stället använda PLC för att inte bland ihop dessa.

Som alla datorer så måste också ett PLC-system programmeras, men programmeringsspråket i ett PLC-system är speciellt anpassat för de människor som tidigare arbetat med relästyrningar.

PLC-systemet har använts i industriella styrningar sedan början av 1970-talet, så tekniken är väl beprövad. Fördelarna gentemot traditionell relästyrning upptäcktes tidigt. Installationskostnaderna minskar betydligt jämfört med denna äldre reläteknik, för att inte tala om kostnaderna som uppkommer när anläggningar ska uppgraderas. Det är i detta läget man ser de stora fördelarna med PLC-systemet. Att göra konstruktions- och programmeringsändringar är betydligt enklare då ett PLC-system styr, än då man använder reläteknik. Dessutom har PLC-systemet funktioner som hjälper dig mycket vid felsökning i maskiner. Du kan enkelt få reda på hur olika signaler förändras och du kan också enkelt prova dig fram genom att påverka signaler. Utöver detta finns unika funktioner för kommunikation både mellan tekniska system och med människa.

Fördelar med PLC-system

Människan har i generationer arbetat för att förbättra sina livsvillkor. Man har uppfunnit verktyg och metoder som har gjort livet lättare att leva. Sedan industrialismens intåg i slutet av 1800-talet har det också handlat om att kunna tillverka så många produkter som möjligt till så låg kostnad som möjligt. På senare år har man dessutom utvecklat produktionen så att det blir låg belastning på miljön. Detta arbete, i varje fall i den svenska industrin, har varit mycket framgångsrikt. I princip kan man säga att det finns tre orsaker till att människan gör förändringar i sitt arbete eller i sin vardag:
– Förenkla och underlätta arbetet
– Bättre produktionsekonomi
– Få en bättre miljö

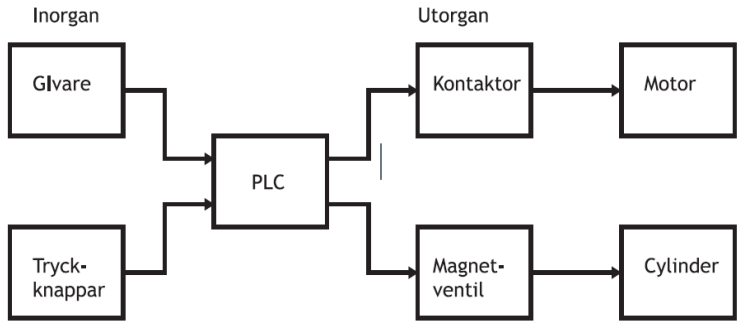
Det som betytt mest för den industriella utvecklingen är elektroniken. Med hjälp av denna har man kunnat ta fram komponenter och apparater som i stor utsträckning gjort industriell tillverkning effektivare och bättre. En annan sak som har betytt oerhört mycket är naturligtvis datorn. I industriell tillverkning sker processerna på olika sätt som exempelvis:
– Blandning av olika vätskor eller torrvaror
– Styra kemiska processer, t.ex. framställning av saltsyra
– Mekanisk bearbetning, t.ex. svarvning, kapning
– Transport av materiel, t.ex. på ett transportband
– Torkning av materiel, t.ex. torkning av papper
– Sortering av olika varor efter vilken typ av varor, eller i vissa fall sortera bort felaktiga varor
– Förpackning av varor, t.ex. i säckar, kartonger eller i burkar
– mm.

Delar i ett styrsystem

För att göra alla dessa saker behövs lite olika maskiner och utrustningar. Det behövs t.ex. transportband, blandningskärl, svarvar, torkutrustning, förpackningsmaskiner etc. Och för att dessa maskiner och utrustningar ska kunna utföra sitt arbete på ett riktigt sätt så behöver de styras. Det är här automationen kommer in. Ordet automation betyder en att en process fortgår utan mänsklig

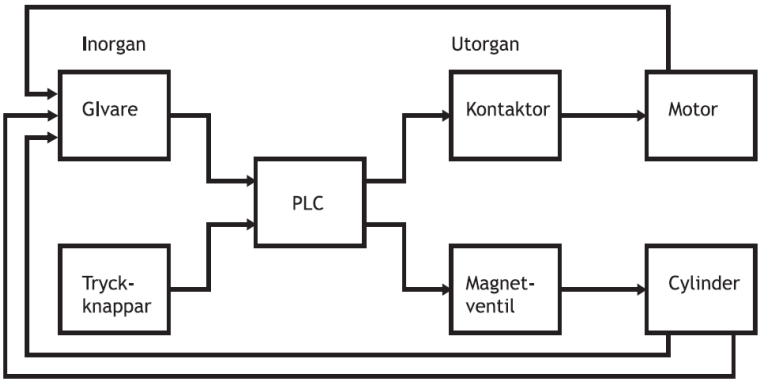
inverkan och står för den teknik och de metoder som används för att göra industriell tillverkning mer effektiv och mer miljövänlig. Här följer några exempel på komponenter som krävs i industriell automation:
– Givare
– Pneumatiska cylindrar
– Magnetventiler
– PLC, Programmable Logical Controller, en industriell dator som styr tillverkningen
– Motorer
– Kontaktorer, används för att styra motorerna
– mm

PLC-systemet är den centrala delen i en industriell styrning. PLC-programmet som man programmerat styr alla enheter så att maskinen ska fungera som konstruktören tänkt sig. Med hjälp av ett PLC-system får man ett kraftfullt verktyg för att underlätta arbetet att konstruera, ta i drift och underhålla en industriell styrning.



Delarna i en industriell styrning

I bilden ovan så styr PLC-systemet både en motor och en cylinder via kontaktor och magnetventil. För att få en säker styrning bör man naturligtvis låta PLC-systemet kontrollera om motorn och cylindern utför sitt arbete på rätt sätt. Detta gör man genom att montera givare på cylinder och motor och koppla dessa till PLC-systemet. Man får då ett återkopplat system.

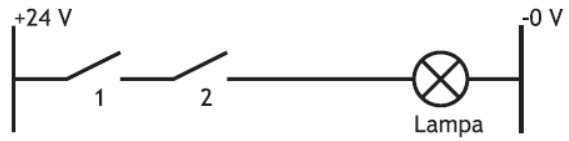


Återkoppling av givare till PLC-systemet

Ett PLC-system styr sina styrdon, ventiler, kontaktorer osv, genom att ge signal till dessa med sina utgångar. På dessa utgångar sitter det normalt reläer eller transistorer monterade (sitter inne i PLC-systemet) och när PLC-systemet sluter dessa skickas en spänning till ventil eller kontaktor. Till PLC-systemets ingångar kopplas givare och tryckknappar. Med hjälp av statusen på ingångarna har PLC-systemet kontroll över vad som händer i anläggningen. När exempelvis en tryckknapp påverkas sänds en spänning till ingången. Det som bestämmer när respektive utgång ska vara till eller från är PLC-programmet.

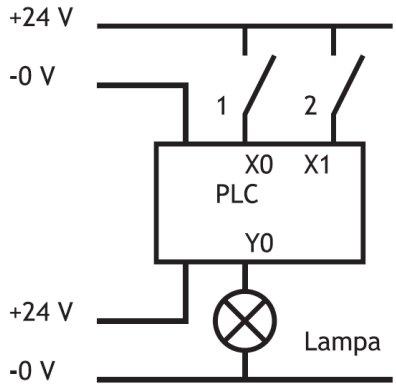
Programmering av PLC-system

För att få de olika styrdonen, cylindrar och motorer, att agera i rätt ögonblick måste det finnas överordnad logik som ser till detta. I moderna anläggningar finns denna logik oftast i PLC-systemet. Genom att bygga samman ett antal logiska funktioner kan man se till att styrdonen agerar på ett korrekt sätt. När man programmerar ett PLC-system så är det egentligen detta man gör -man bygger samman olika logiska funktioner.



Ett enkelt styrsystem

Vi tänker oss att man kopplar ihop två stycken elkopplare i serie till en lampa. Kretsen matas av plus och minus. Den logiska regeln för kretsen att lampan ska vara tänd lyder så här:
Om elkopplare 1 är påverkad och elkopplare 2 är påverkad ska lampan vara tänd.
Detta är den regeln för detta styrsystem. Samma funktion kan enkelt uppnås med hjälp av ett PLC-system. För detta krävs att elkopplarna kopplas till ingångarna på PLC-systemet och lampan till en av utgångarna.



PLC-systemet styr lampan

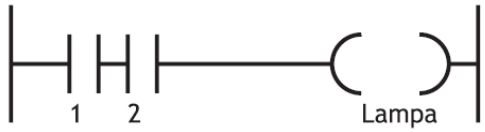
Till ingång X0 har vi kopplat elkopplare 1 och till ingång X1 har vi kopplat elkopplare 2. När respektive elkopplare blir påverkad flyter en ström in på ingången och detta känner PLC-systemet av. Till utgång Y0 har vi kopplat lampan, vilket innebär att när PLC-systemet aktiverar utgång Y0 så tänds lampan.
För att PLC-systemet ska utföra avsedd funktion måste vi skriva ett PLC-program, och mata in detta i PLC-systemet. Detta program ska tända lampan om både elkopplare 1 och elkopplare 2 påverkas.

Inmatning av program

Inmatningen av PLC-programmet görs antingen med en programmeringsenhet (handdosa) eller med hjälp av ett datorprogram, ett s k programmeringsverktyg.
Programmeringsverktyget är ett mycket effektivt hjälpmedel för dig när du programmerar, tar i drift eller felsöker i maskiner som styrs av PLC-system. Det finns ett antal olika programmeringsspråk att välja mellan. Dessa är:
– Instruktionslista (textbaserad)
– Ladder (liknar reläschema)
– FBD (function block diagram)
– ST eller SCL (högnivå)
– SFC (sekvensstyrning)

Beroende på vad man ska programmera väljer man lämpligt anpassat språk.

Här nedan är programmet skrivet i ladder.



Ladder (reläschema) som beskriver styrning av lampan

Som man kan se har detta schema i stort sett samma utseende som elschemat vi ritade för elkopplarna. Reläschema kommer ursprungligen från USA, och används där som standard för elritningar. Elschema har kommit att bli standard när det gäller beskrivning av PLC-systemets styrfunktioner och används i alla PLC-fabrikat.

Genom att sedan mata in detta Reläschema i ett programmeringsverktyg, och föra över programmet till PLC-systemet, och sedan starta PLC-systemet, så har man utfört allt som behövs. Stegen är alltså dessa:

1. Gör elkonstruktionen, dvs. beskriv vilka elkopplare, kontaktorer, reläer etc som ska kopplas till PLC-systemet. Bestäm också till vilka in- och utgångar dessa ska kopplas.
2. Skriv programmet med hjälp av ett programmeringsverktyg. Skriv programmet i något av de fem språken.
3. För över PLC-programmet från datorn till PLC-systemet.
4. Börja med att testa inkoppling, dvs prova så att alla elkopplare, kontaktorer etc är kopplade till rätt in- eller utgång. Denna test kan du enkelt göra genom att använda programmeringsverktyget.
5. Testkör programmet. I denna fas har du stor hjälp av de funktioner i programmeringsverktyget som låter dig studera statusen på alla in- och utgångar. Att hitta fel i anläggningen blir betydligt smidigare om du använder datorprogram.
6. Om funktionen inte blev helt korrekt gör du ändringar i programmet, för över och testkör det nya programmet. Med hjälp av programmeringsverktyget gör du dina ändringar i funktionen och kan sedan prova igen.
7. Dokumentationen över den styrda maskinen ska sedan innehålla både de vanliga elritningarna, men också den dokumentation som du erhåller från programmeringsverktyget och de andra programmeringsverktygen. En av fördelarna med programmeringsverktyget är just att du på ett enkelt sätt får ut dokumentationen.

Jämförelse med ett PLC system och ett styrsystem med reläer

När man konstruerar ett program finns ett antal olika logiska funktioner att använda. Med en AND-funktion seriekopplar man och med en OR-funktion parallellkopplar man. Genom att kombinera och bygga samman ett antal logiska funktioner kan man se till att styrdonen agerar på ett korrekt sätt.

När man programmerar ett PLC-system så är det egentligen detta man gör, man bygger samman olika funktioner. De vanligaste funktionerna är:
– AND-funktion (OCH - seriekoppling)
– OR-funktion (ELLER - parallellkoppling)
– NOT-funktionen (INTE - normalt sluten kontakt)
– Fördröjning (tidrelä)

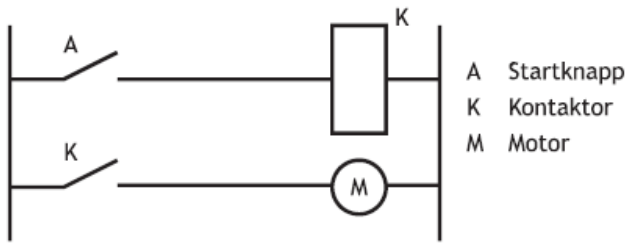
Genom att kombinera olika funktioner kan man konstruera mycket avancerade program till PLC-systemet.

Exempel på funktioner

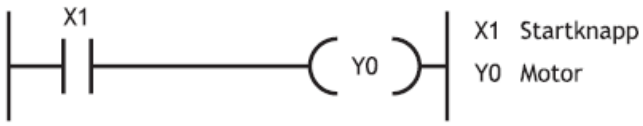
Här följer några exempel hur man realiserar en elektrisk funktion till ett program.

Exempel 1

En motor ska startas om startknappen A påverkas. Startknappen kopplas till ingång X1 och kontaktorn kopplas till utgång Y0.

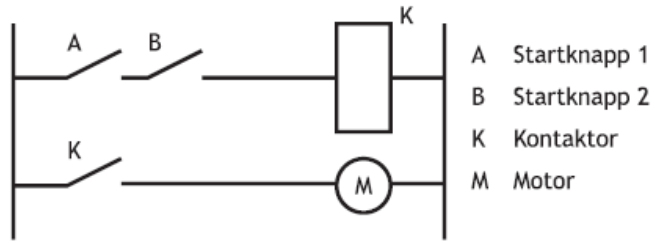


Elschema

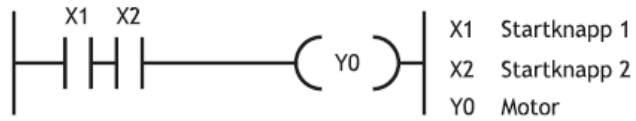


Ladder (reläschema)

Exempel 2
En motor har försetts med två startknappar. Motorn ska startas om bägge startknapparna påverkas. Startknapp A kopplas till ingång X1 och Startknapp B kopplas till ingång X2. Kontaktorn kopplas till utgång Y0.

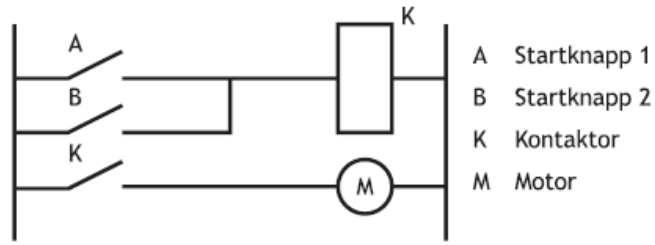


Elschema

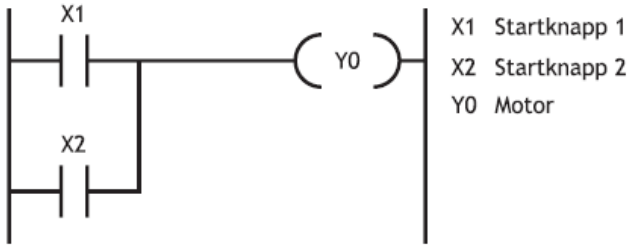


Ladder (reläschema)

Exempel 3
Om vill ändra funktionen för de två startknapparna så att man ska kunna starta motorn från någon av de två gör man en enkel ändring i programmet. Man behöver alltså inte koppla om några trådar. Startknapp A kopplas till ingång X1 och Startknapp B kopplas till ingång X2. Kontaktorn kopplas till utgång Y0.



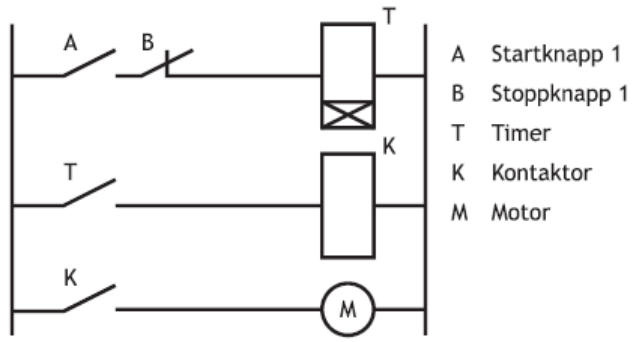
Elschema



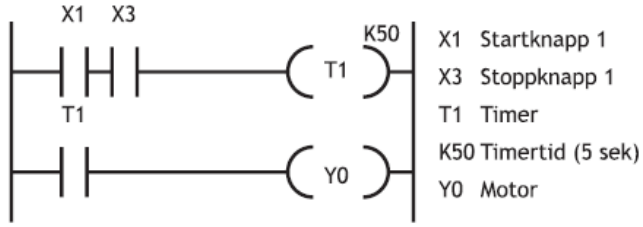
Ladder (reläschema)

Exempel 4

En motor ska starta 5 sekunder efter det att startknappen påverkats. Startknappen A kopplas till ingång X1 och stoppknappen B till ingång X3, kontaktorn kopplas till utgång Y0.



Elschema



Ladder (reläschema)

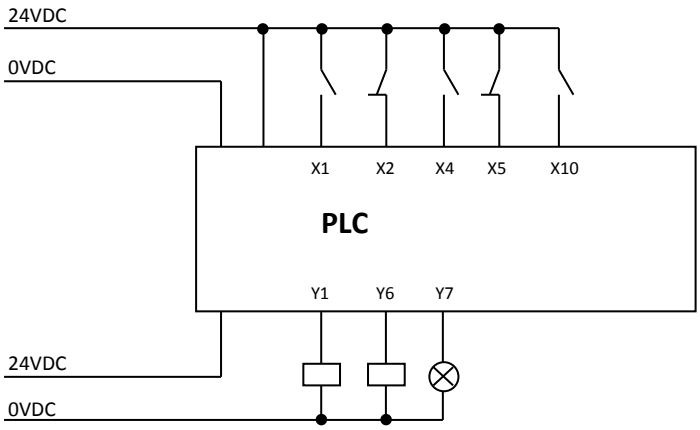
PLC arbetssätt

Processorn i PLC´t kan endast utföra en instruktion i taget. Allt eftersom den bearbetar instruktioner och data, sparar den aktuella tillstånd som senare, då alla instruktioner är lästa, kan exekveras.

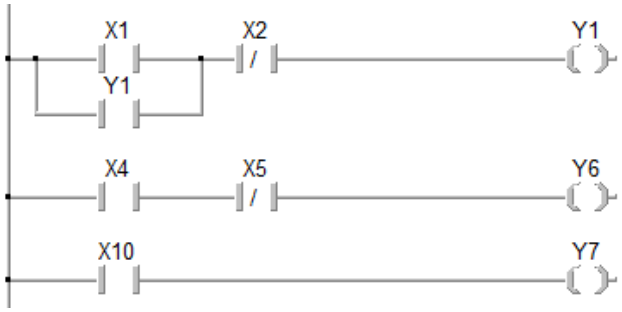
Till ett PLC ansluts sensorer och givare till ingångar. Dessa ”läser” aktuell status t.ex. om en elkopplare är till eller från, att en cylinder är i sitt hemmaläge etc.

Till utgångar ansluts don som ska styras som t.ex. kontaktor, magnetventil, lampor etc.

PLC



Anslutningar



Ladder i program

Processorns (CPU) programbearbetning

När PLC sätts i RUN-läge startar övervakningstiden för programcykeltiden, den tid det tar för CPU att exekvera alla instruktioner i programmet. I vissa PLC kan man ställa in den maximalt tillåtna tid för detta. Skulle CPU ta längre tid än den tillåtna maximala kommer PLC normalt gå i STOPP-läge.

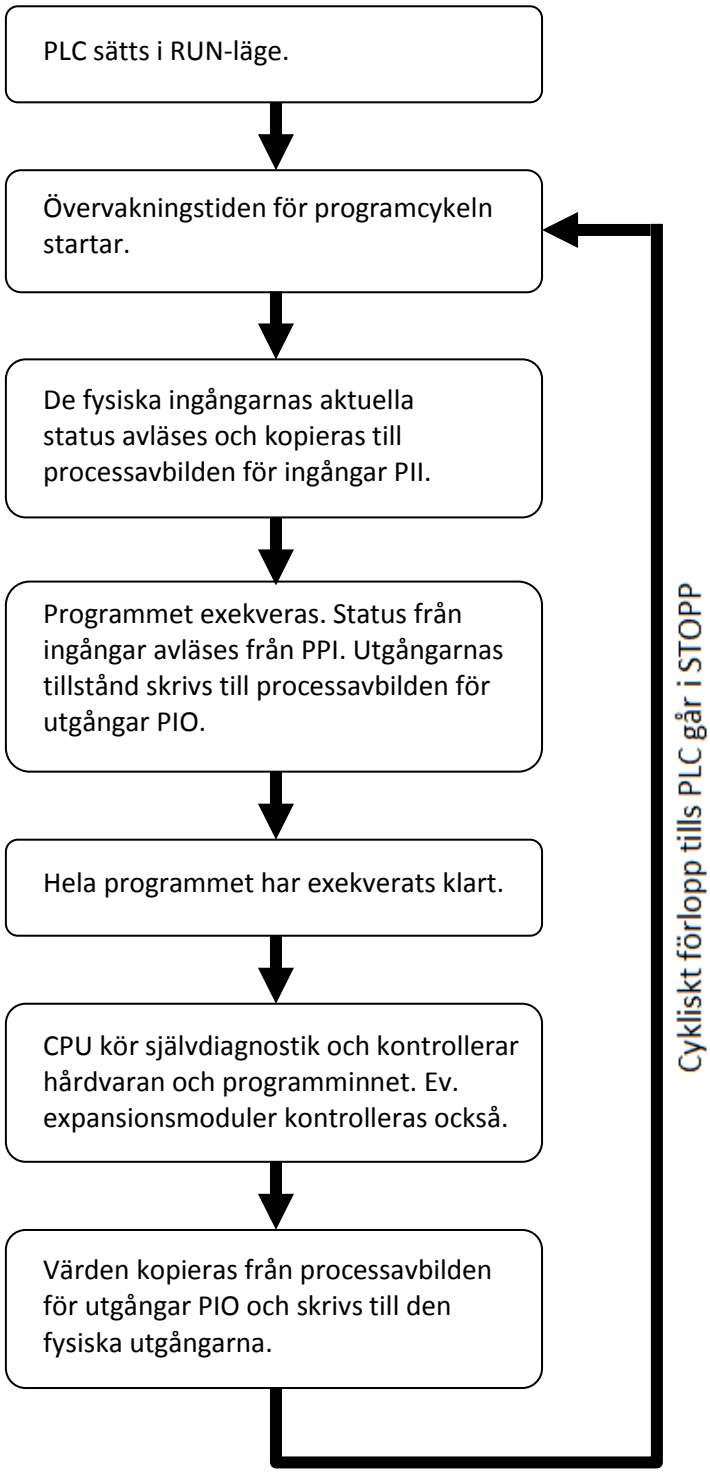
Därefter avläser PLC statusen på alla digitala ingångar och kopierar aktuell status till ett temporärt minne kallat process image for input PII.

Sedan börjar CPU att exekvera programmet. När instruktioner avläses bildas ett s.k. RLO result of logic operation. Beroende på om RLO ger en 0:a eller 1:a utförs det som kommer efter. Exempel: I programexemplet ovan i andra uttrycket står det att X4 ska vara 1 och X5 0 för att Y6 ska tilldelas en 1:a (TILL). Då ger RLO en 1:a och skickar en 1:a till adressen Y6 i processavbilden för utgångar PIO. Så här arbetar PLC genom alla instruktioner.

CPU kör sedan en självdiagnostik och kontrollerar hårdvaran i systemet samt expansionsmoduler om det finns. CPU kontrollerar också programminnet. Skulle fel indikeras går vanligtvis CPU i stopp-läge.

När alla instruktioner i programmet är exekverade kopieras aktuellt värde för alla digitala utgångar från PIO till de fysiska utgångarna.

Detta innebär att det är en viss eftersläpning från det att ingångarna avläses tills det att aktuell utgång aktiveras. Är det ett mindre program går detta snabbare än om programmet och stort. Det är programcykeltiden som avgör hur snabbt detta sker. Normalt handlar det om några µsek till msek beroende på CPU-hastighet.

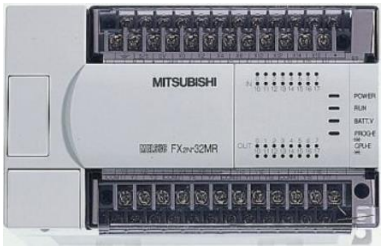


Exempel

Antag att ingångarna vid ett tillfälle har följande status:

X1=0
X2=0
X4=1
X5=0
X10=1

Ingångar

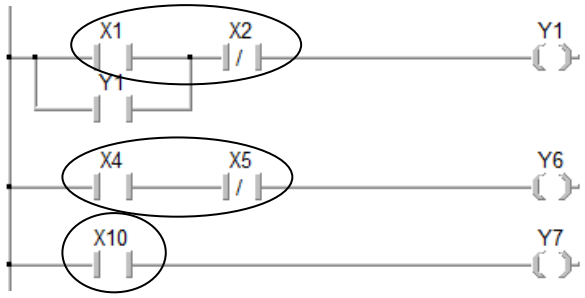


Utgångar

Det som kommer att ske är att aktuell status på alla ingångar skrivs till PII.

X1=0
X2=0
X4=1
X5=0
X10=1

PII (Process image for input)
X1=0
X2=0
X4=1
X5=0
X10=1



När programmet exekveras kommer information om de digitala ingångarnas status att läsas från PII.

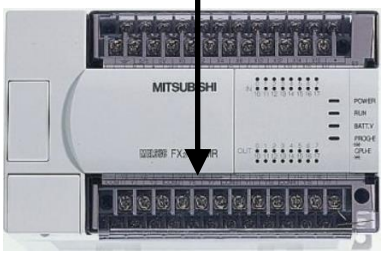
I första uttrycket kommer Y1 att tilldelas en 0:a. Detta skrivs in i PIO.

I andra uttrycket kommer Y6 att tilldelas en 1:a. Detta skrivs till PIO.

I tredje uttrycket kommer Y7 att tilldelas en 1:a. Detta skrivs till PIO.

PIO (Process image for output)
Y0=0
Y6=1
Y7=1

Ingångar



Utgångar

När alla instruktioner d.v.s. hela programmet är exekverat kommer statusen i PIO att kopieras till de fysiska utgångarna på PLC.

Därefter sker samma sak om igen d.v.s. ingångarnas status Kopieras till PII och programmet exekveras igen.

Tiden det tar för CPU att exekvera hela programmet kallas För programcykeltid. Den är ofta endast någon µsek till msek.