

GRAF CET SYSTEM

GRAF CET är ett standardiserat system som beskriver sekvensförlopp. Systemet är standardiserat inom IEC. GRAF CET översatt till svenska = grafisk beskrivning av styrning steg- övergång. GRAF CET S´ s 3 huvudkriterier är följande:

- Både den eller de som konstruerar processen och de som utformar styrsystemet måste ha ett gemensamt språk för att kunna beskriva önskade funktioner.
- Språket måste vara enkelt och entydigt.
- Språket måste vara oberoende av typ av styrsystem.

Som synes är det ett mycket flexibelt system att använda, man binder sig inte vid ett speciellt styrsystem utan har möjlighet att använda t ex. pneumatiska styrsystem, reläer, eller PLC system.

Tekniken bygger på "steg för steg" modellen. Med detta menas att endast ett steg kan vara påverkat åt gången. För att åstadkomma detta, ska efterföljande steg nollställa föregående. Tack vare denna konstruktion riskerar programmet aldrig att hoppa över något steg. Därför erbjuder GRAF CET en större säkerhet än andra metoder.

Kravet för att stegen ska växla till nästa o s v. är att villkoren mellan stegen är uppfyllda. Mitsubishi har utvecklat ett program kallat MEGRAF, detta översätter ett program till GRAF CET.

GRAF CET ENKLARE BESKRIVNING

För att enkelt beskriva GRAF CET tänker vi oss att vi ska beskriva arbetsgången då man lagar en punktering på ett däck.

Som alla känner till gör man på följande sätt::

Aha... punktering.

Däcket ska avmonteras.

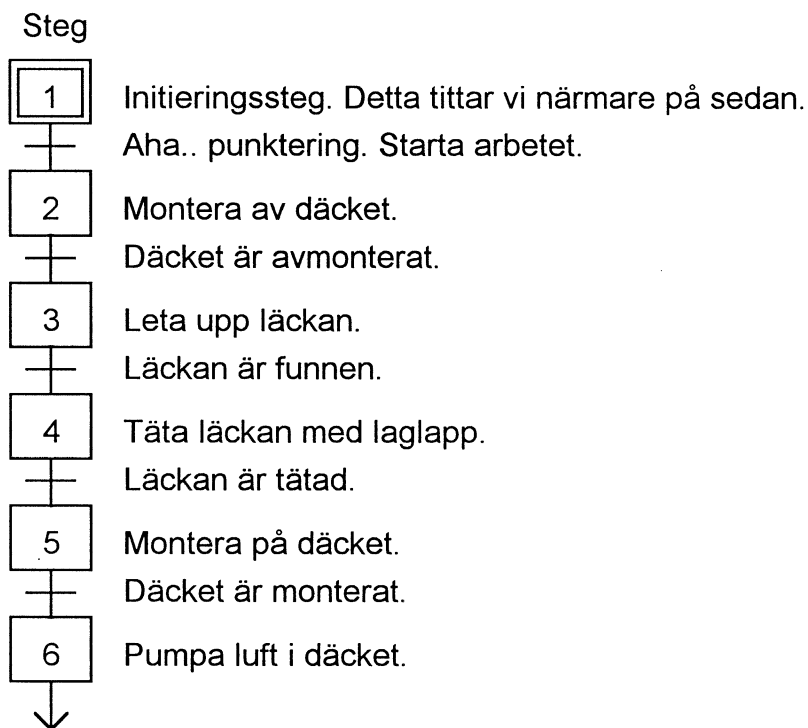
Hålet ska letas upp.

Hålet ska tätas.

Däcket ska monteras på.

Däcket ska pumpas.

Nu ska vi presentera denna arbetsgång i GRAFCET. Det gäller att rätt ordning sker, man kan ju inte klistra på laglappen innan man har monterat av däcket.



Förklaringar:



Initieringssteg. Detta steg ska "sparka igång" programcykeln.



Steg för händelser som ska utföras t ex. reläer, minnen.



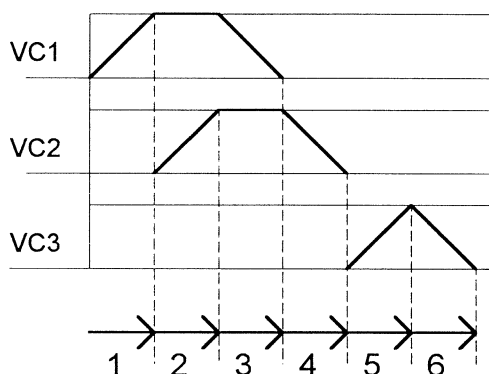
Villkor, slutande funktion t ex. givare, sensorer, tryckknappar.



Villkor, brytande funktion t ex. givare, sensorer, tryckknappar.

GRAFSET EXEMPEL MED CYLINDRAR

Det tidigare exemplet vi har jobbat med ska vi nu översätta till GRAFCET. Vi börjar med att rita upp följdidiagrammet och tar reda på hur många steg / händelser vi behöver.



Det blir totalt 6 steg.

Följande ska hända i de olika stegen:

- Steg 1 VC1 ska gå ut.
- Steg 2 VC1 ska stanna kvar ute. VC2 ska gå ut.
- Steg 3 VC1 ska gå in. VC2 ska stanna kvar ute.
- Steg 4 VC2 ska gå in.
- Steg 5 VC3 ska gå ut.
- Steg 6 VC3 ska gå in.

Nu ska vi använda adresser som PLC:t känner igen. Vi använder minnena M0 - M5 för de 6 olika händelserna.

Vi anger stegen dessa adresser:

Steg	Adress
Steg1	M1
Steg2	M2
Steg3	M3
Steg4	M4
Steg5	M5
Steg6	M0

Som du ser har vi "flyttat" fram alla steg ett snäpp. Detta återkommer vi till längre fram.

Så här kommer de olika stegen att se ut.



Som vi ser ovan är vårt första steg M0 som gör att VC3 går in. Detta medför att själva programcykeln startar i steg M1. M0 har blivit ett initieringssteg. Om vi ska starta programcykeln med tryckknapp A ska denna placeras som ett villkor innan steg M1.

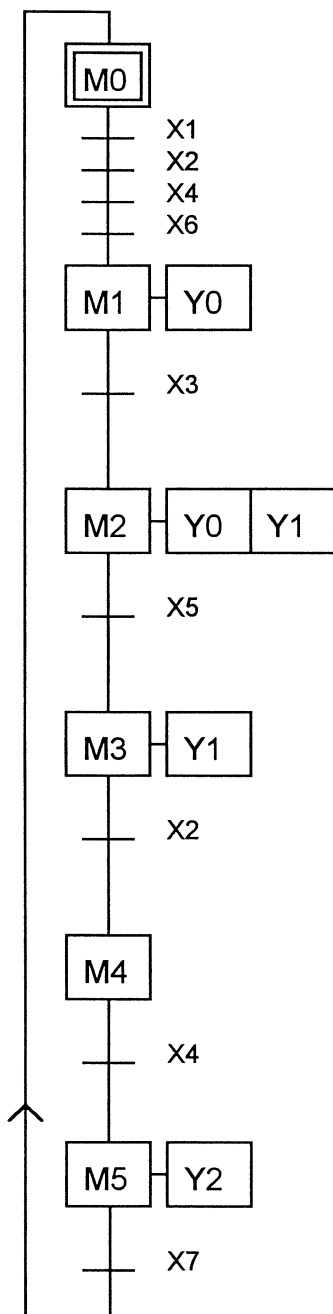
GRAFDET FÖRKLARING AV STEG / VILLKOR

Nu är det dags att placera in alla villkor för programcykeln.

Studera följdidiagrammet när du tittar på de olika stegen.

Vi börjar med villkoren mellan M0 och M1. För att programcykeln ska starta fordras att tryckknapp A (X1) trycks in. För att kontrollera att alla cylindrar står i sitt "hemmaläga" lägger vi in de 3 signalerna C1B (X2), C2B (X4) och C3B (X6). Vi ska alltså lägga in 4 signaler eller villkor mellan steg M0 och M1.

När cylinder VC1 har gått ut helt påverkas C!F (X3) då ska detta villkor läggas mellan M1 och M2 o s v. Studera stegskvensen .



Signalerna X2 (C1B), X4 (C2B), och X6 (C3B) indikerar att alla 3 cylindrarna står i sina hemmalägen. När X1 (tryckknapp A) trycks in ska programmet starta.

Nu "ettställes" steg M1 och steg M0 ska "nollställas". Då steg M1 är "ettställt" ska Y0 (VC1) gå ut.

Då Y0 (VC1) har gått ut helt påverkas X3 (C1F) och villkoret mellan steg M1 och M2 är uppfyllt.

Nu "ettställes" steg M2 och steg M1 ska "nollställas". Då steg M2 är "ettställt" ska Y0 (VC1) stanna kvar ute och Y1 (VC2) ska gå ut.

Då Y1 (VC2) har gått ut helt påverkas X5 (C2F) och villkoret mellan steg M2 och M3 är uppfyllt.

Nu "ettställes" steg M3 och steg M2 ska "nollställas". Då steg M3 är "ettställt" ska Y1 (VC2) stanna kvar ute och Y0 (VC1) ska gå in.

Då Y0 (VC1) har gått in helt påverkas X2 (C1B) och villkoret mellan steg M3 och M4 är uppfyllt.

Nu "ettställes" steg M4 och steg M3 ska "nollställas". Då steg M4 är "ettställt" ska Y1 (VC2) ska gå in.

Då Y1 (VC2) har gått in helt påverkas X4 (C2B) och villkoret mellan steg M4 och M5 är uppfyllt.

Nu "ettställes" steg M5 och steg M4 ska "nollställas". Då steg M5 är "ettställt" ska Y2 (VC3) gå ut.

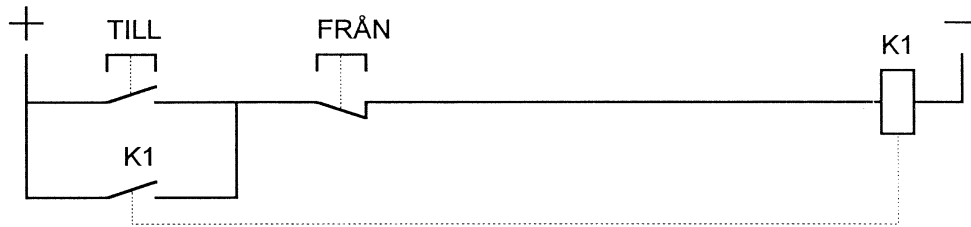
Då Y2 (VC3) har gått ut helt påverkas X7 (C3F) och villkoret mellan steg M5 och M0 är uppfyllt.

Nu måste vi "knyta" ihop det sista steget M5 med det första M0 för att vi ska kunna köra programmet flera gånger. Som vi kan se så kommer steg M0 att "nollställa" Y2 (VC3) som gör att X6 (C3B) får signal.

Nu står alla 3 cylindrarna i sitt "hemmaläge" och programmet inväntar signal på X1 (tryckknapp A), då ska ytterligare en programcykel utföras.

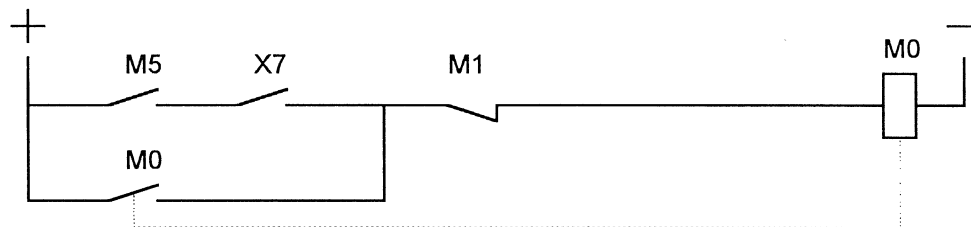
GRAF CET STEG ⇒ STEG ÖVERGÅNG

GRAF CET systemet jobbar med s k. hållkretsar även kallat minneskretsar och självhållningskretsar. För att förklara detta använder vi reläteknik. Se nedan.



För att "ettställa" relä K1 krävs att TILL trycks ned och att FRÅN är opåverkad. När TILL trycks ned "ettställs" relä K1. Då TILL släpps får relä K1 självhållning via hjälpkontakt på K1. Då FRÅN trycks ned "nollställs" relä K1. Detta är en s k. hållkrets / minneskrets.

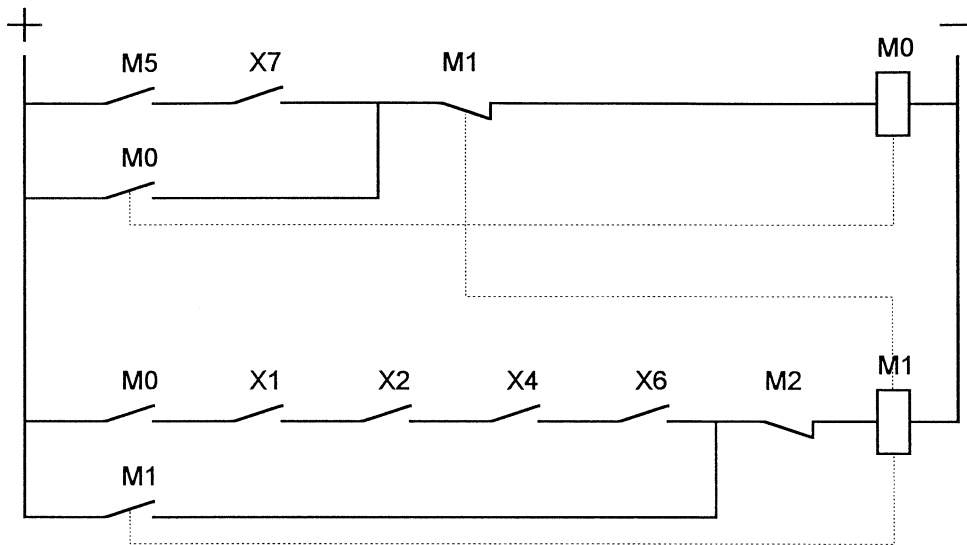
Nu ska vi översätta kretsen ovan till GRAF CET systemet. Vi visar det med steg M0 och M1. De övriga stegen jobbar på liknande sätt.



TILL knappen är utbytt mot X7. Vi har lagt till M5, detta för att indikera att föregående steg har "ettställts". Hjälpkontakten M0 ser till att hålla kvar M0 för självhållning. Steget efter M0 är M1, och som vi ser bryter M1 steget M0. Här fungerar M1 som FRÅN knapp och detta sker då M1 "ettställs". Summerar vi detta så krävs följande för att steg M0 ska "ettställas":

- Steg M5 ska vara "ettställt".
- X7 (C3F) ska ha signal.
- Steg M1 ska vara "nollställt".

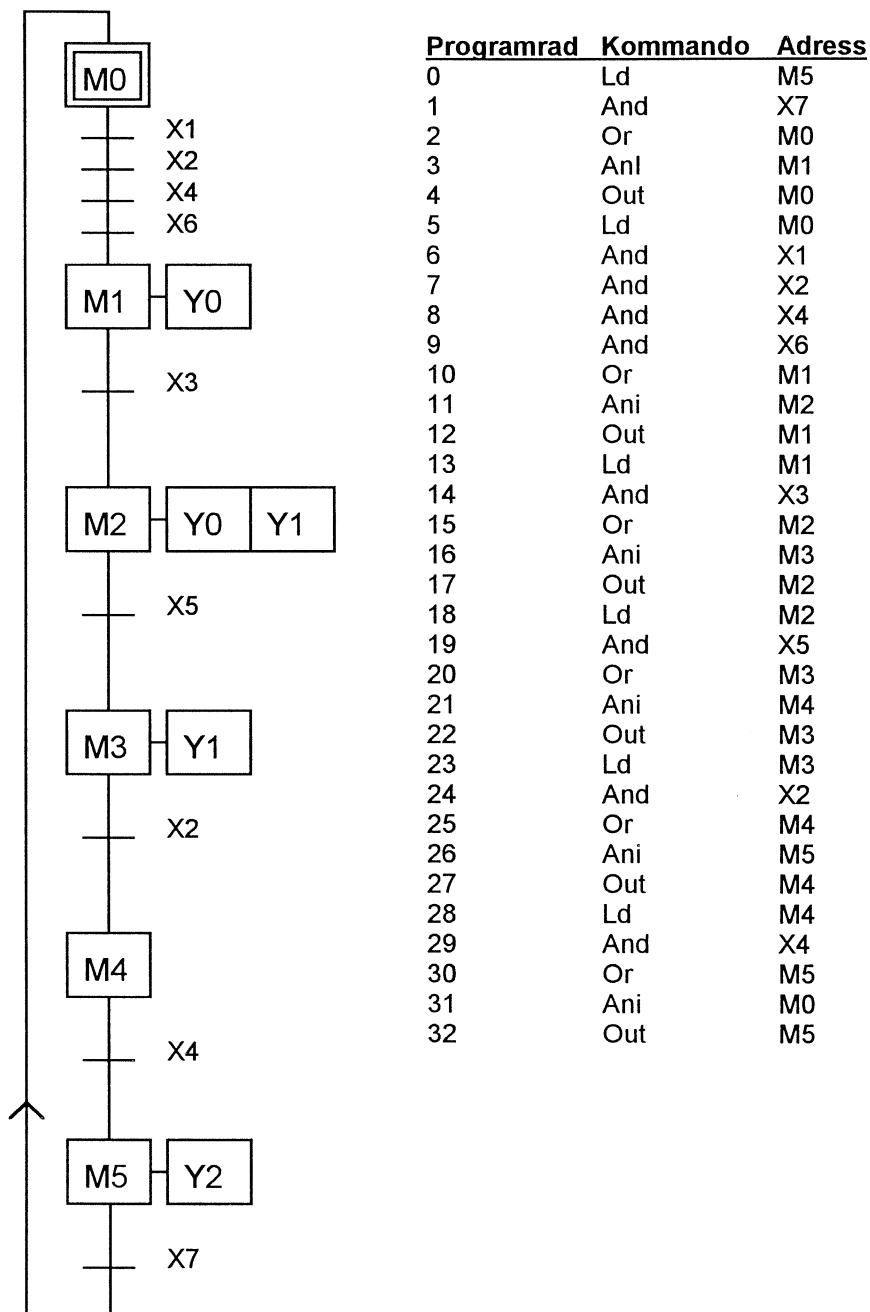
Nu ska vi se på nästa steg i programmet.



Då steg M0 är "ettställt" är första villkoret uppfyllt. Då X2 (C1B), X4 (C2B) och X6 (C3B) står i sina hemmalägen är dessa villkor också uppfyllda. Nu krävs endast att X1 (tryckknapp A) trycks in så är samtliga villkor mellan steg M0 och M1 uppfyllda, då "ettställs" M1 och "nollställer" M0. M1 får självhållning genom hjälpkontakt på M1. Så här arbetar programmet igenom alla stegen.

GRAFSET PROGRAMSKRIVNING

Nu ska vi skriva själva programmet till PLC:t.



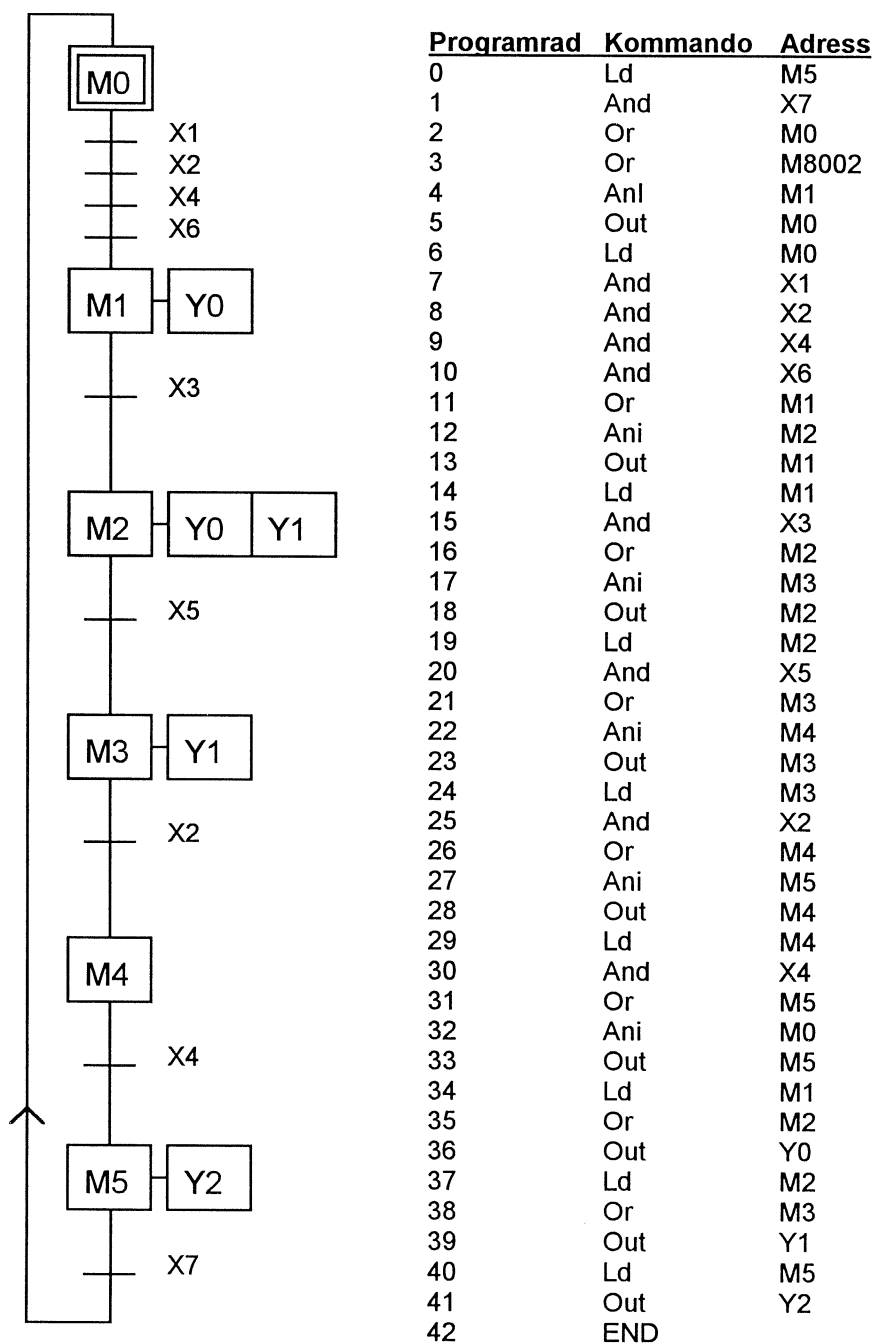
Programraderna behöver man inte mata in, dessa ordnar PLC't själv.

GRAFSET INITIERINGSPULS

Om man studerar programmet ovan närmare finner man att programmet aldrig kommer igång. För att det ska kunna starta måste steg M0 vara "ettställt". Detta kan man ordna med en specialminnescell i PLC't och den heter M8002. Denna puls varar endast ca. 0,8µs då spänningen till PLC't slås på. Det räcker för att "ettställa" steg M0. Vi **lägger till Or M8002** mellan programrad 2 och 3.

GRAFSET UTGÅNGSSIGNALER

Nu återstår det att skriva in programmet för utgångarna Y0, Y1 och Y2. Vi studerar sekvensen och lägger in utgångarna enligt de olika stegen. Observera att specialminnescellen M8002 är inlagd nedan, altså börjar vi på programrad 34. Se nedan.



Observera att på sista programraden måste kommandot END skrivas in.

Mata in programmet och för över det till PLC't och provkör. Prova också att läsa driftstatus på stegen M0 -M5 så ser man tydligt steg för steg gången.