

| Dokument | Förklaring | Dat. | Revision |
|----------------|--------------------------------|--------|----------|
| KI-221-003-003 | Kom igång med trukturerad Text | 080402 | 1.0 |

Twincat: PLC Control

Kom igång med Strukturerad Text (ST) programmering

1. Kod exempel.

- a. Exempelen som demonstreras visas i *kursiv* text.
- b. Nyckelord visas i **Blå** text
- c. Kommentarer för kod visas med **grön** text.

2. Säkerhet och krav.

- a. Ansvarig person skall se till att applikationen eller användandet av de beskrivna produkterna följer de brukliga kraven för säkerhet, inkluderat relevanta lagar, förordningar, riktlinjer och standarder.
- b. Denna dokumentation är endast avsedd att användas av utbildade specialister i Automatisering vilka är familjära med applicerbara nationella standarder.

Bakgrund och generell beskrivning av Strukturerad text (ST)

Programmeringsspråket strukturerad text består av instruktioner som påminner mycket om högnivåspråket tex PASCAL eller C. ST programmering är en del av IEC 61131 standarden.

Ett litet program för att räkna upp en variabel till 10 kan se ut såhär.

```
IF iSumma <= 9 THEN  
    iSumma:=iSumma + 1;  
END_IF
```

Anrops prioritet

I ST finns det olika operander man kan använda för sina villkor. I tabellen nedan visas de i prioritetsordning.

| Operand | Symbol | Prioritet |
|-----------------|---------------------------|------------------|
| Inom paranteser | (uttryck) | Högsta prioritet |
| Funktionsanrop | Funktionsnamn(parametrar) | |
| Exponent | EXPT | |
| Negering | NOT | |
| Multiplikation | * | |
| Division | / | |
| Modulo | MOD | |
| Summera | + | |
| Subtrahera | - | |
| Jämförelser | <,>,<=,>= | |
| Lika med | = | |
| Ej lika med | <> | |
| Boolean AND | AND | |
| Boolean XOR | XOR | |
| Boolean OR | OR | Lägsta prioritet |

Instruktioner.

I ST finns ett antal instruktioner; tex IF THEN, WHILE DO, CASE OF. Dessa finns listade i tabellen nedan med exempel.

| Instruktion | Exempel |
|-----------------------|---|
| Tilldelning | A:=B; CV:=CV+1; C:=SIN(X); |
| Anropa funktionsblock | MyTimer(IN:=TRUE, PT:=T#1s); A:=MyTimer.Q; |
| RETURN | RETURN |
| IF | IF D<0.0 THEN C:=A; ELSEIF D=0.0 THEN C:=B; ELSE C:=D END_IF |
| CASE | CASE i OF 1: BOOL1:=TRUE; 2: BOOL2:=TRUE; ELSE BOOL1:=FALSE; BOOL2:=FALSE; END_CASE |
| FOR | FOR I:=1 TO 100 BY 2 DO IF ARR[i]=70 THEN J:=I; EXIT END_IF END_FOR |
| WHILE | WHILE J<= 100 AND ARR[J] <> 70 DO J:=J+2; END_WHILE |
| REPEAT | REPEAT J:=J+2; UNTIL J= 101 OR ARR[J]= 70 END_REPEAT |
| EXIT | EXIT; |
| Tom instruktion | ; |

Lite mer om tilldelningsoperanden (:=).

När man ska tilldela en variabel ett värde använder man operanden tilldelning (:=). På den vänstra sidan av tilldelningsoperanden har man en variabel som får värdet av uttrycket på höger sida.

Exempel:

```
Var1 := Var2 * 10;
```

Efter exekvering av denna rad kommer Var1 ha ett värde som är tio gånger så stort som Var2.

Lite mer om anrop av funktionsblock (FB)

I strukturerad text anropar man funktionsblock genom att skriva namnet på funktionsblockets instansnamn och tilldela funktionsblockets in parametrar värden inom parentes. I följande exempel anropas en timer, myTimer, med värden för in parametrarna IN och PT.

Den returnerade variabeln Q tilldelas variabeln A.

Den returnerade variabeln adresseras genom att man använder namnet på funktionsblockets instans följt av en punkt och variabelns namn.

```
myTimer(IN:=TRUE , PT:=T#1s , Q=> , ET=> );  
A:=myTimer.Q;
```

T#1s eller **T#1000ms** är så man skriver en sekund i TwinCAT PLC Control.

Exekvering med exempel

Skillnaden mellan strukturerad text och liknande högnivåspråk är att programmet skrivet i strukturerad text kommer exekveras cykliskt hela tiden uppifrån och ner till skillnad från högnivåspråket som mer eller mindre kommer att exekveras en gång per händelse uppifrån och ner. Detta ställer lite andra krav på tankegången och programmeringen.

Automation AB

Följande exempel startar en timer myTimer, när variabeln bStart går hög och bDone är låg. När timern har gått ut sätts variabeln bDone.

```
(*Timern startar med variabeln bStart*)  
myTimer(IN:=bStart , PT:=T#1s , Q=> , ET=> );
```

```
(*Om timerns tid löpt ut så sätt bDone till "ett"*)  
IF myTimer.Q THEN  
    bDone:=TRUE;  
END_IF
```

Detta exempel fungerar bra, fast bara en gång!
Problematiken är att bDone aldrig blir låg. Det vill vi ju naturligtvis uppnå i detta exempel.

Lösningen på detta är att använda t.ex. en ELSE sats.

```
(*Timern startar med variabeln bStart*)  
myTimer(IN:=bStart , PT:=T#1s , Q=> , ET=> );  
  
(*Om timerns tid löpt ut så sätt bDone till "ett" annars till "noll"*)  
IF myTimer.Q THEN  
    bDone:=TRUE;  
ELSE  
    bDone:=FALSE;  
END_IF
```

Övrigt.

I strukturerad text så är det viktigt att skapa sina instansnamn av text funktionsblock innan man använder dessa i sin kod. Gör man detta innan kodningen börjar så underlättar det en hel del vid programmeringen och misstag undviks.

Några exempel på instansiering av FB och deklaration av variabler.

PROGRAM MAIN

(*Deklarationsdelen för variabler i PLC Control*)

VAR

```
fbTimer:TON; (*Instansnamn för en timer*)
fbCounter:CTU; (*Instansnamn för en räknare*)
fbTrig_R:R_TRIG; (*Instansnamn för flank trigger*)
fbAx1_Mc_Power:MC_Power; (*Instansnamn för et fb till motioncontrol*)
bTest:BOOL; (*Variabel av typen Bool*)
iValue:INT; (*Variabel av typen Integer*)
```

END_VAR