Remanent minneshantering – TC2

Alla Beckhoff-system har möjlighet att lagra data vid spänningsbortfall. När spänningen bryts nollställs data som inte har lagrats och i vissa applikationer behöver en del variabler bibehålla sitt värde när systemet startas upp igen, i andra applikationer skall systemet starta upp med samma förinställda grundvärden varje gång. Minneslagring kan ske på olika sätt, det beror på vilken typ av data som skall sparas, hur ofta ändringar skall lagras, mängden data samt vilken typ av hårdvara som skall användas. Detta dokument beskriver endast minneslagring med remanent data i CX och PC-system. Det finns också en överskådlig presentation av UPS hantering beskriven i slutet av dokumentet.

Teknisk bakgrund

Dokumentet skall användas som hjälp för att använda retain och persistent minnet i Beckhoff-systemen, beskrivningen är endas överskådlig och ej fullständig. De Beckhoff-system som har använts som exempel i detta dokument är en CX1020, CX5020 och en CX8090. Bibliotek som behöver installeras för de respektive systemen är:

CX	Bibliotek	Funktionsblock	TwinCAT version
CX1020	TcUtilities.lib	FB_WritePersistentData	TwinCAT v2.10 eller högre
CX5020	TcSUPS.lib	FB_S_UPS	TwinCAT v2.11 build 2016 eller högre (R2)
CX8090	TcSystemCX80xx.lib	FB_S_UPS_CX80xx	TwinCAT v2.11 build 2220 eller högre (R3)

Hanteringen av remanent data i CX10x0, CX50x0 och CX80x0 är i grunden densamma men CX50x0 och CX80x0 har inbyggd UPS med möjlighet att skriva persistent data vid spänningsbortfall. Exemplet för CX5020 samt CX8090 bygger på funktionsblock för att skriva persistent data med hjälp av den integrerade 1-sekund UPS:en vid spänningsbortfall eller omstart. I exemplet för CX1020 finns ingen UPS integrerad, skrivning av persistent data hanteras händelsestyrt med anrop av funktionsblock i PLC koden.

1 Remanent data

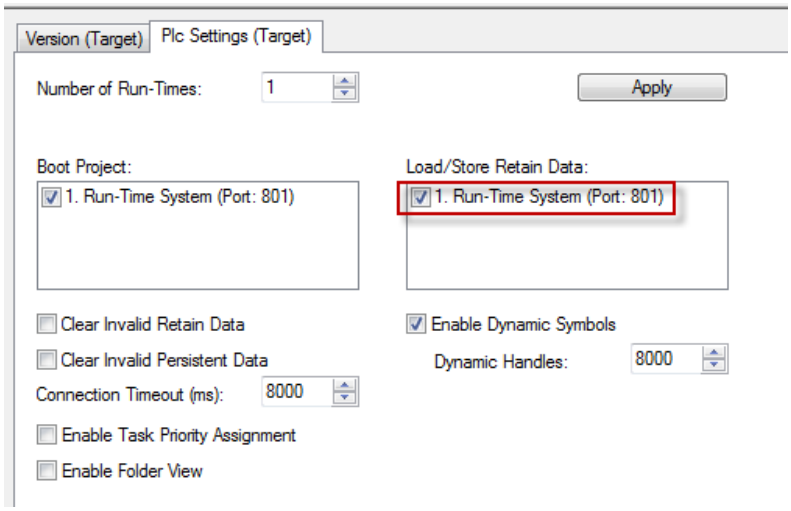
Remanent data är data som bibehåller sin information efter spänningsbortfall. Exempel på lagringsmedium med remenenta egenskaper är hårddiskar, Novram och flashminnen. *Retain* data och *Persistent* data är exempel på *Remanent* data, detta kan skrivas både cykliskt och acykliskt. En UPS är att rekommendera vid användning av retain och persistent data eftersom det är viktigt att kunna stänga ner PLC:n kontrollerat, bryts spänningen vid skrivning till fil kan data annars bli korrupt.
Remanent data på Beckhoff CX-system uppnås med hjälp av Novram eller genom att deklarera variabler som *Var Retain* eller *Var Persistent*. Det finns också funktionsblock för att kontrollera skrivning av persistent data. Vid kontrollerad nerstängning av TwinCAT skrivs variabler som är deklarerade som *Var Retain* eller *Var Persistent* till respektive fil under Boot mappen på målsystemet.

1.1 Persistent data

Persistent data är data som ändras i PLC programmet under exekveringen och bibehåller sitt aktuella värde efter en avstängning eller spänningsbortfall av anläggningen. Behåller även data efter förändringar i den befintliga koden t.ex. vid onlineändringar eller när en ny projekt nerladdning görs. Värdena behålls tills de blir överskrivna av PLC kod eller en *Reset All* görs i TwinCAT PLC Control.
Notera! Om en persistent deklarerad variabel är en del av en instans (funktionsblock eller datastrukt) kommer hela instansen att lagras persistent.

1.2 Retain data

Retain data är data som behålls efter ett spänningsbortfall, onlineändringar och återstart av TwinCAT men kommer att nollställas av ny nerladdning av PLC-projekt eller vid en *Reset All* i TwinCAT PLC Control.
För att retain data skall kunna användas måste *Load/Store Retain Data* vara ikryssad i System Managern – PLC Configuration under fliken *PLC Settings (Target)*.



2 Översikt på enheter med Remanent minne

Tabellen visar de olika interna minnen som finns tillgängliga för olika CX-modeller.

CX	Novram	Flash Minne (Remanent data)
CX9000	Basic CPU med inbyggd spänningsaggregat och Novram*	16 Mbyte (internt, går inte att utöka)
CX9001	Basic CPU med inbyggd spänningsaggregat och Novram*	32 Mbyte (internt, går inte att utöka)
CX9010	Basic CPU med inbyggd spänningsaggregat och Novram*	32 Mbyte (internt, går inte att utöka)
CX9020	Basic CPU med inbyggd spänningsaggregat och Novram*	512 Mbyte (microSD, går att utöka)
CX1010	Novram integrerat i separat spänningsaggregat CX1100-000x	64 Mbyte (CF kort)
CX1020	Novram integrerat i separat spänningsaggregat CX1100-000x	64 Mbyte (CF kort)
CX1030	Novram integrerat i separat spänningsaggregat CX1100-001x	64 Mbyte (CF kort)
CX5010/CX5020		Integrerad 1-Sekund UPS, 1 Mbyte på CF kort
CX80x0		Integrerad 1-Sekund UPS, 1 Mbyte på microSD kort
PC/IPC		HDD, SSD, CF kort, CFast kort

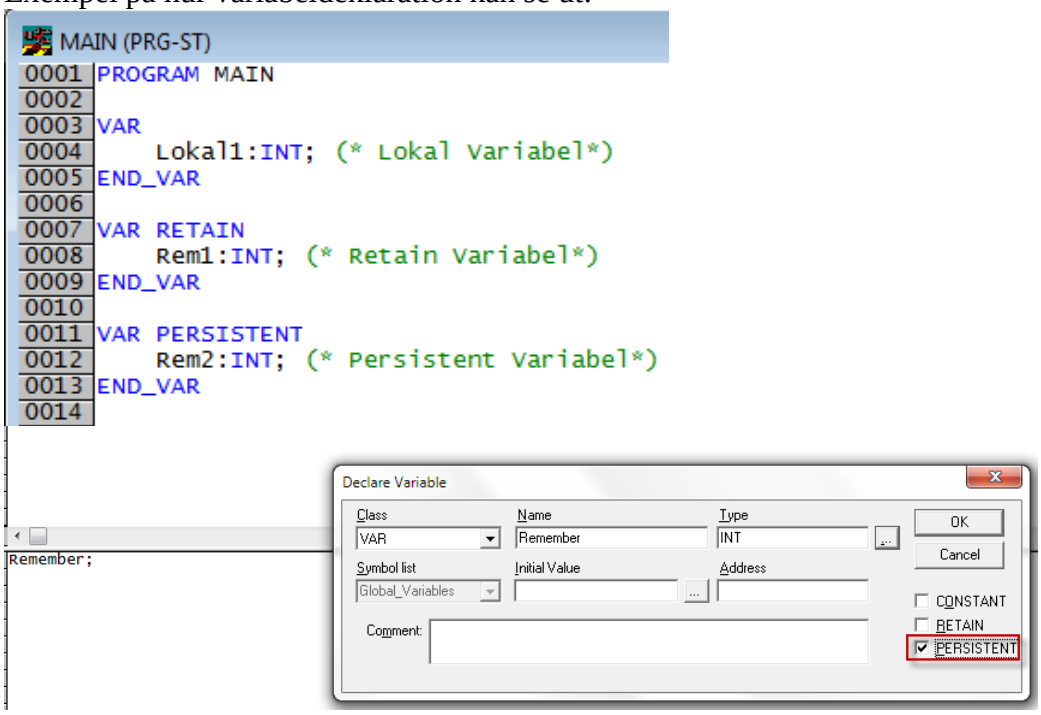
* Novram finns inbyggt i vissa system, en form av remanent minne. Hanteras i ett separat dokument.

3 Hantering av Retain och Persistent data i Beckhoff system

Vid uppstart av CX-system, Embedded PC och PC- system läses filer från Boot mappen i flashminnet, där finns information om program och parametrar (data-backup). Boot mappen ligger under c:\TwinCAT\Boot för system med Windows Embedded Standard/XP operativ och för system med Windows CE/Compact 7 operativ ligger Boot mappen under \Hard Disk\TwinCAT\Boot.
BC-och BX-systemen fungerar annorlunda och har en intern minnesarea specificerad för remanent data och hanteringen är helt annorlunda mot CX-systemen. Där är det adresseringen av variabeln som avgör om den är remanent eller inte, detta förklaras inte i detta dokument.

CX50x0 och CX8090 har inget NOVRAM, använd den inbyggda 1-sekund UPS för att säkert lagra persistent data på Compact Flash/microSD kort, retain data supportas inte av den inbyggda UPS:en. Upp till 1 MB data kan lagras.

Exempel på hur variabeldeklaration kan se ut:



När TwinCAT stängs ner kontrollerat kommer retain och persistent data att skrivas i sina respektive filer på hårddisken, *TCPLC_R_x.wbp* och *TCPLC_T_x.wbp*. Grundinställt hamnar dessa två filer under Boot mappen. Bryts spänningen utan att en korrekt nedstängning av systemet kommer ingen retain eller persistent fil att hinna skapas och vid uppstart finns ingen fil att läsa. När systemet startar upp på nytt kommer dessa filer att läsas in och wbp filen byter samtidigt namn till wb~ och blir en back-up fil. En ny wbp fil skapas inte förrän en ny skrivning sker. Det kan därför bli problem om ett system startas om två gånger eftersom det inte finns någon wpa fil att läsa andra gången om inte en sparning har gjorts mellan avstängningarna (och en ny fil har hunnit skapas).
På Windows Embedded Operativ finns möjlighet att aktivera skrivskydd, EWF och FBWF. Kontrollera att Boot mappen inte är skrivskyddad, i det fallet går det inte att skapa retain eller persistent fil.

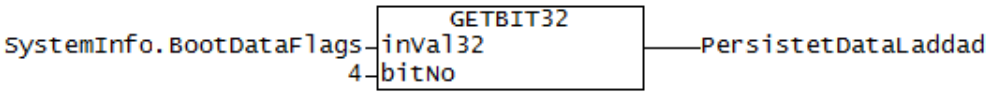
Tabell: Backup-filer remanent data

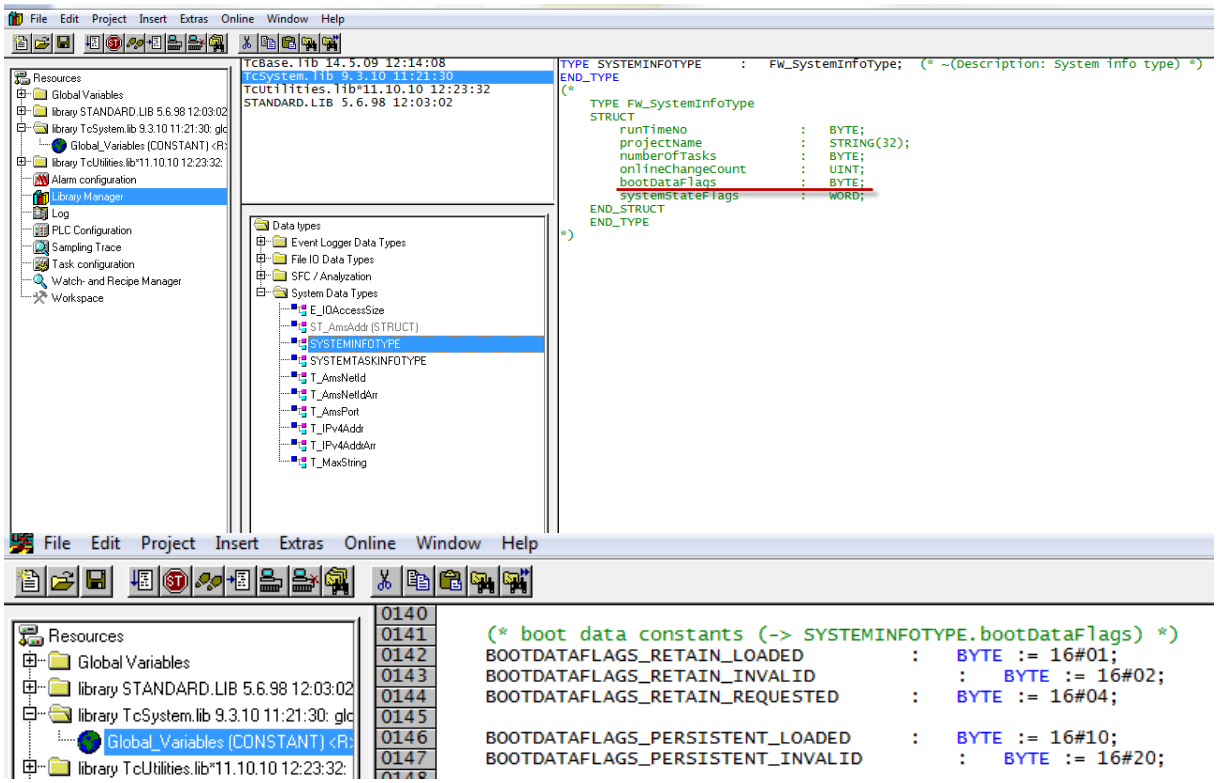
Fil namn	Beskrivning
TCPLC_P_x.wbp	Boot projekt (x = nummer på run-time systemet)
TCPLC_R_x.wbp	RETAIN variabler (x = nummer på run-time systemet)
TCPLC_T_x.wbp	PERSISTENT variabler (x = nummer på run-time systemet)
TCPLC_R_x.wb~	Backup kopia på RETAIN variabler (x = nummer på run-time systemet)
TCPLC_T_x.wb~	Backup kopia på PERSISTENT variabler (x = nummer på run-time systemet)

Fördelen med att skriva den persistenta datan till fil med funktionsblock är att kunna på ett kontrollerat sätt spara data och på så sätt vara säker på att det finns en wpb fil. Rekommendationen är att använda sig av funktionsblocket (*FB_WritePersistentData*) som skapar wpb fil, och efter uppstart med en korrekt inläsning skriva tillbaka filen. Vid uppläsning av wpb fil skapas en wb~ fil. Då finns alltid har en korrekt skriven backupfil

som kan användas vid ett spänningsbortfall som gör att systemet ej hinner skriva ned den persistenta filen korrekt.

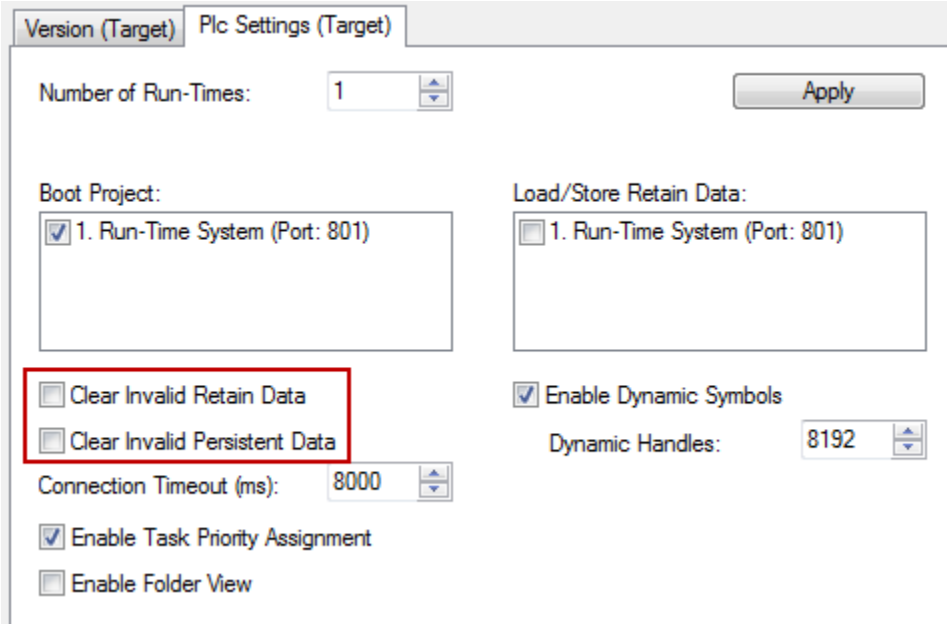
För att kunna kontrollera om persistent och retain data är korrekt läst från fil finns en systemvariabel, *bootDataFlags* (1 byte stor) i biblioteket TcSystem.lib. Denna byte indikerar status på boot data efter laddande av den. De övre fyra bitarna (b4-b7) talar om status för den persistenta datan och de under fyra bitarna (b0-b3) indikerar status på retaindatan.
Exempelvis går det att kontrollera att den persistenta datan är korrekt uppladdad med nedan kod (bit 4 i *bootDataFlags*):



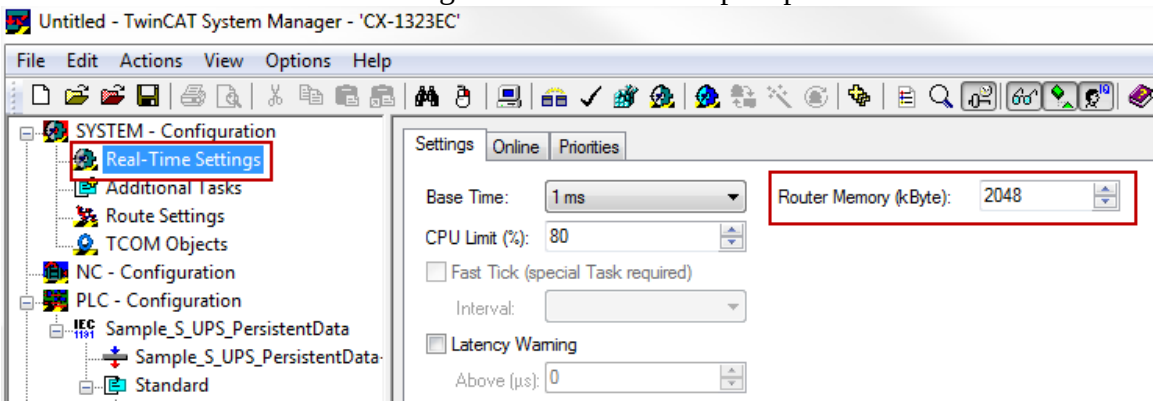


Bit nummer	Beskrivning
0	RETAIN variabler: LOADED (utan fel)
1	RETAIN variabler: INVALID (back-up kopian blev laddad, eftersom ingen giltig data fanns tillgänglig)
2	RETAIN variabler: REQUESTED (RETAIN variabler ska laddas, en inställning i TwinCAT System Control)
3	Reserverad
4	PERSISTENT variabler: LOADED (utan fel)
5	PERSISTENT variabler: INVALID (back-up kopian blev laddad, eftersom ingen giltig data fanns tillgänglig)
6	Reserverad
7	Reserverad

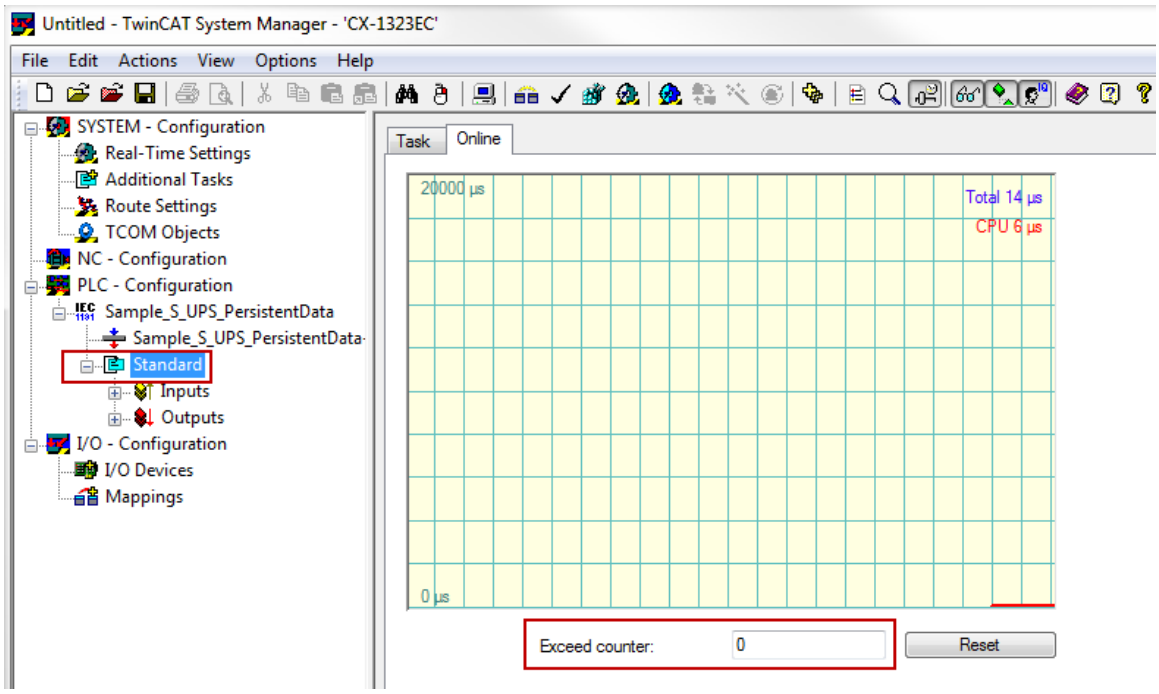
Om det inte går att skriva till retain eller persistent filen då TwinCAT stängs ner eller om filen blir korrupt av annan anledning, kommer backupfilen att laddas vid uppstart av systemet och bit 1 sätts i *bootDataFlags* för retain respektive bit 5 för persistent. Vill man inte att denna backup fil skall laddas kan man i System Managern välja bort detta genom att kryssa för *Clear Invalid Retain Data* eller *Clear Invalid Persistent Data*.
Väljer man detta alternativ kommer retain och persistenta variabler att starta med initieringsvärdet eller värdet 0 om ingen retain/persistent fil finns att ladda, dvs den laddar ingen backupfil.



Notera!
”Load/Store Retain Data” i bilden ovan skall endast kryssas i om retain data skall användas, den har inget med persistent data att göra.
Routerminnet måste vara tillräckligt stort för att kunna spara persistent data.



Normalt behöver inte detta ökas men om Exceed countern i Tasken ökar när persistent data skrivs skulle detta kunna bero på att Routerminnet behöver utökas.



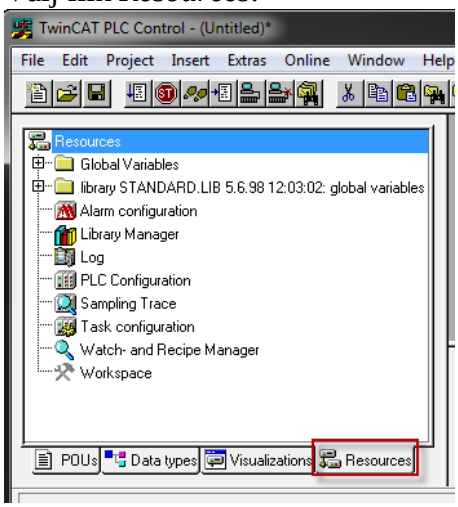
4 Exempel 1, Persistent data med funktionsblock för en CX1020.

För att använda persistent data till CX och PC så rekommenderas att använda funktionsblock som finns i bibliotek *TcUtilities.Lib*. Detta följer med vid installation av TwinCAT men behöver länkas in i mjukvaran.
Funktionsblocket *FB_WritePersistentData* kan nyttjas för att vid önskat tillfälle skriva ner den persistenta datan till lagringsmediet. Notera att det endast är persistent data, inte retain data som skrivs med detta funktionsblock.

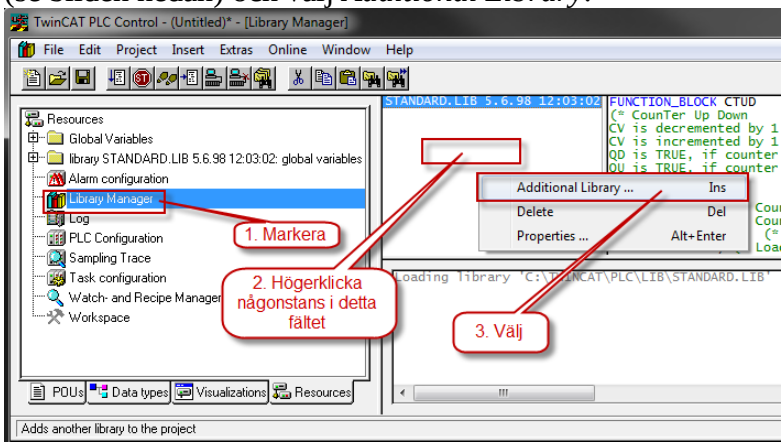
Val av CPU	PLC bibliotek att inkludera	TwinCAT version
PC eller CX (x86)	TcUtilities.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib inkluderas per automatik)	TwinCAT v2.9.0 Build >= 959
CX med Intel (x86 processor) CX med ARM processor		TwinCAT v2.10.0 Build >= 1301

4.1 Lägg till bibliotek i PLC Control

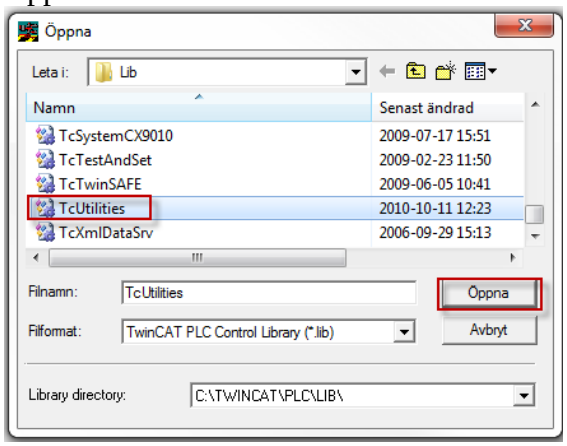
Börja med att addera biblioteket i TwinCAT - PLC Control. Skapa ett nytt projekt gå in i menyn File – New.
Välj flik *Resources*.



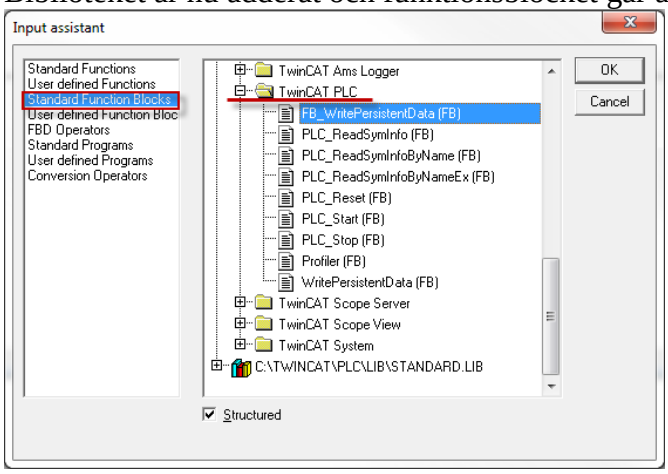
Markera *Library Manager*, dubbelklicka på den, högerklicka någonstans i det vita fältet (se bilden nedan) och välj *Additional Library*.



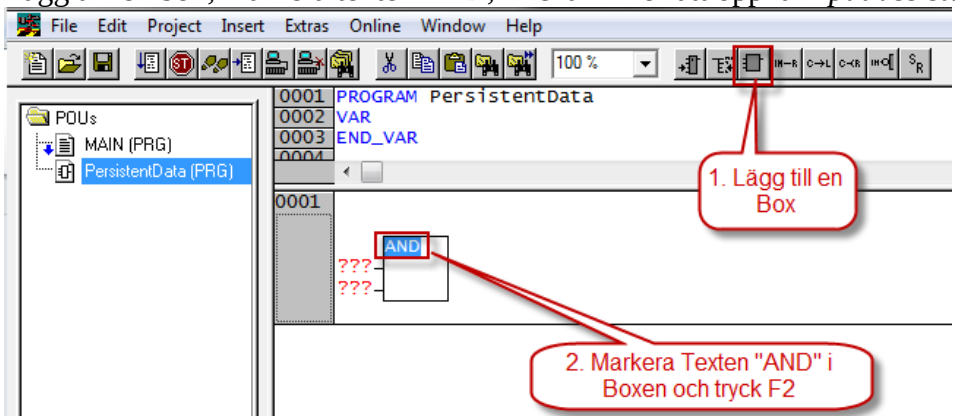
Öppna biblioteket som heter *TcUtilities*.



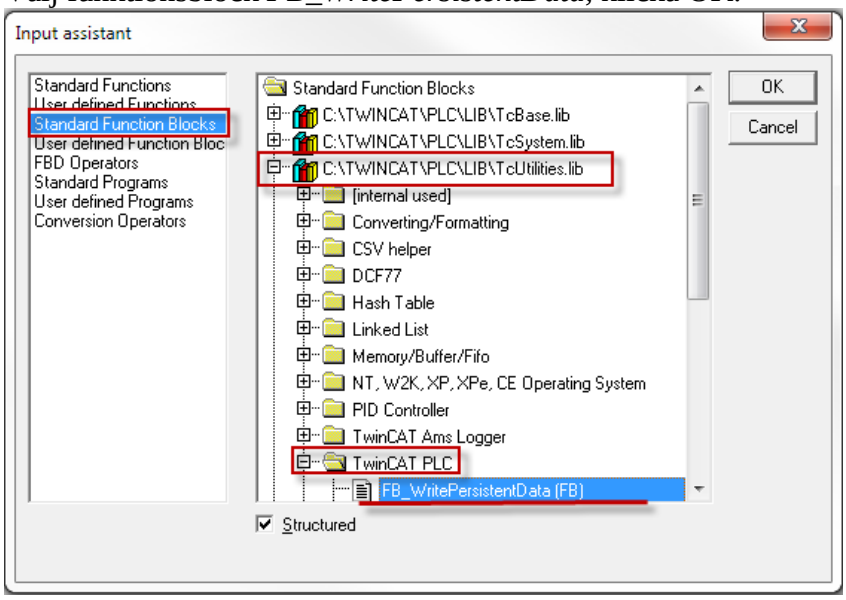
Biblioteket är nu adderat och funktionsblocket går att använda i projektet.



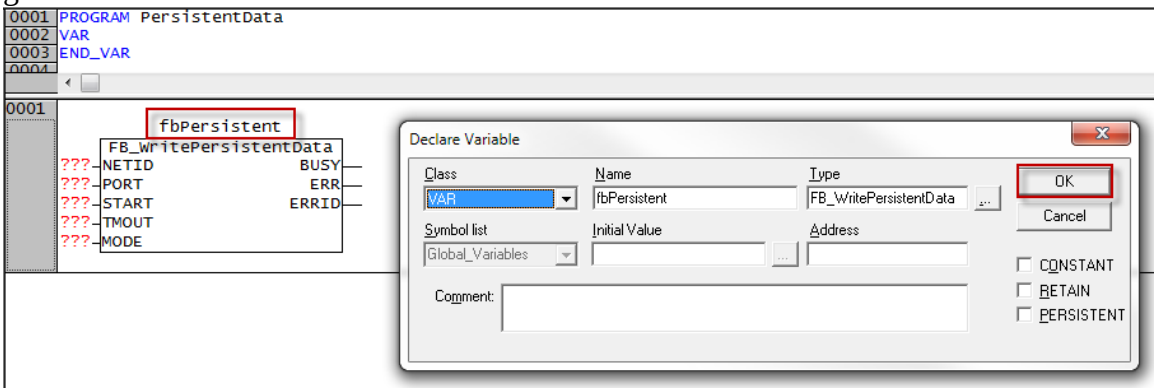
I programexemplet är *Main* programmet skapat i Strukturerad Text och programdelen med persistent data funktionsblock är skapat i Funktions Blocks Diagram editorn. Lägg till en box, markera texten *AND*, klicka F2 för att öppna *Input assistant*.



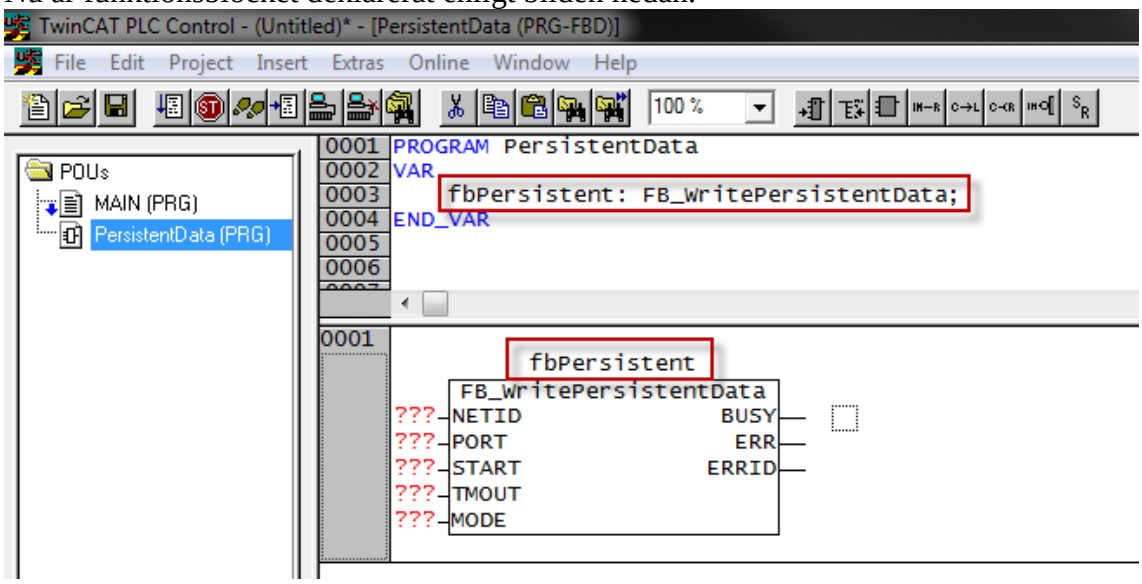
Välj funktionsblock *FB_WritePersistentData*, klicka OK.



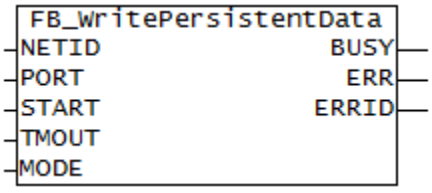
Funktionsblocket skall deklareras med ett instansnamn, i exemplet nedan heter den *fbPersistent*. Skriv namnet ovan funktionsblocket och tryck på Enter, deklarera variabeln genom att klicka OK.



Nu är funktionsblocket deklarerat enligt bilden nedan.



4.2 Beskrivning av funktionsblocket *FB_WritePersistentData*



Beskrivning på ingångspinnar till funktionsblocket *FB_WritePersistentData*

Var Input	Typ	Beskrivning	
NETID	T_AmsNetId	AmsNetId adress, på lokal PC skriv en tom sträng ’’	
PORT	UINT	ADS portnummer på den PLC runtime persistent data skall skrivas till	
START	BOOL	Positiv flank på denna bit aktiverar skrivning	
TMOUT	Time	Ange längden på den timeout som inte får överskridas under exekveringen av ADS kommandon (ADSWRTCTRL)	
MODE	E_PersistentMode	Persistent data mode	
		SPDM_2PASS := 0 *1)	All persistentdata är från samma PLC cykel (skrivs ner under ett scanvarv). Kan när skrivning sker påverka så att PLC cykel tid överskrids
		SPDM_VAR_BOOST := 1	Data i varje persistentvariabel är från samma PLC cykel (skrivning delas upp under flera scanvarv). Överskrider endast PLC cykeltiden ifall skrivning av den största persistenta variabeln tar längre tid att skriva än vad PLC cykeltiden är. (Rekommenderas ej med 1-sekunds UPS:en)

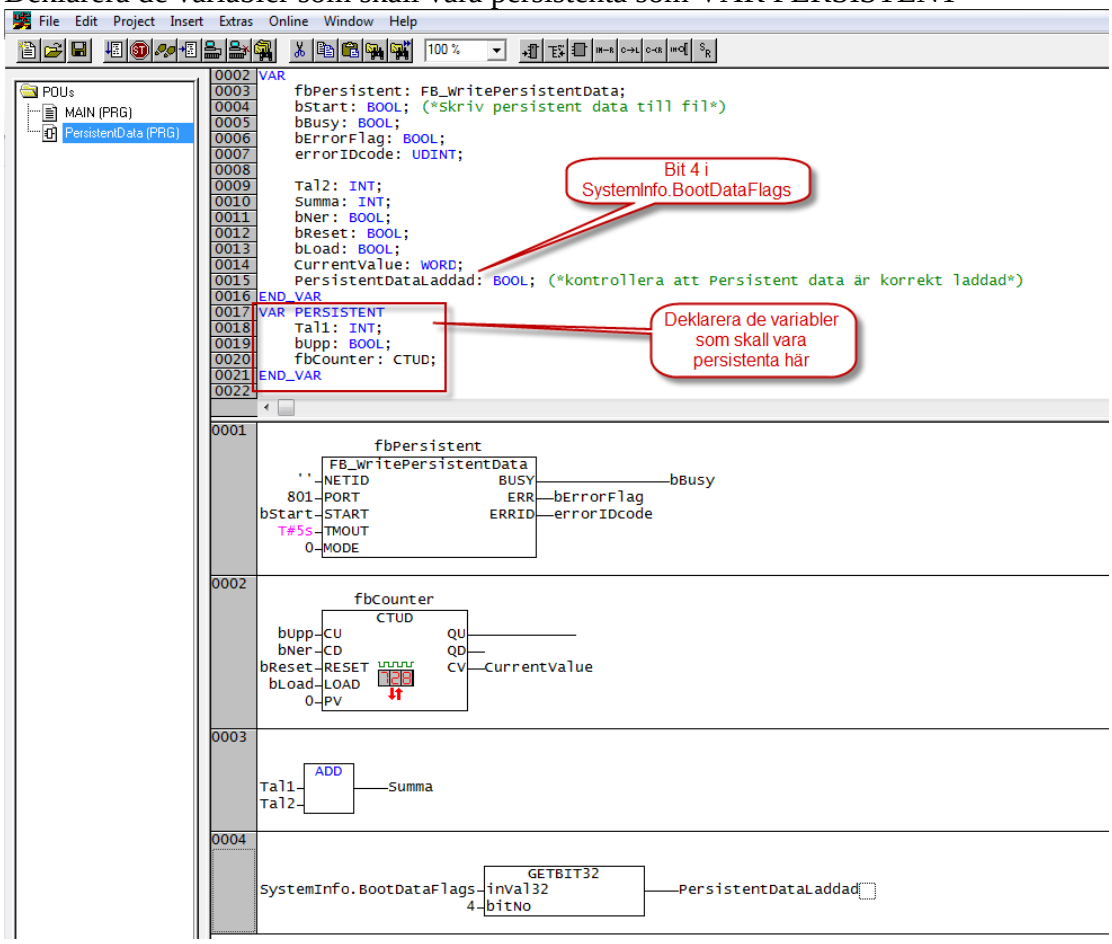
*1) Rekommenderas i normalfallet, är det mycket persistent data så kan det dock vara bättre att dela upp skrivningen med mode SPDM_VAR_BOOST. Kontrollera om Exceed counter räknar upp i System Manager, byt då till SPDM_VAR_BOOST. Med 1-sekund UPS bör endast grundinställd parameter SPDM_2PASS användas, detta för att undvika att kondensatorn hinner ladda ur innan all data hunnit skrivas vilket skulle kunna bli fallet om det är mycket data och alternativet SPDM_VAR_BOOST valts.

Beskrivning på utgångspinnar till funktionsblocket *FB_WritePersistentData*

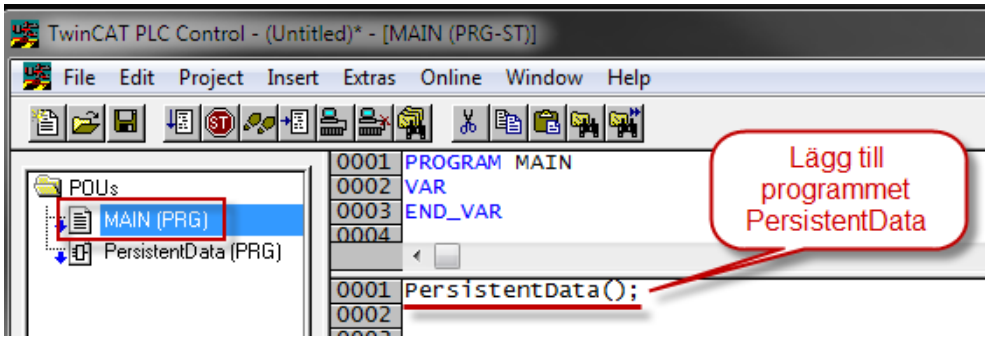
Var Output	Typ	Beskrivning
BUSY	BOOL	Denna bit är satt så länge funktionsblocket exekveras, så länge skrivning/läsning pågår
ERR	BOOL	Fel flagga, indikerar ADS error
ERRID	UDINT	Om ERR är satt kan ADS felkoden utläsas i denna "ADS Return Codes”

4.3 Exempelkod i TwinCAT PLC Control för CX1020

Deklarera de variabler som skall vara persistenta som VAR PERSISTENT



När PLC koden är klar skall programdelen *PersistentData* läggas till i *Main* programmet, se bild nedan:



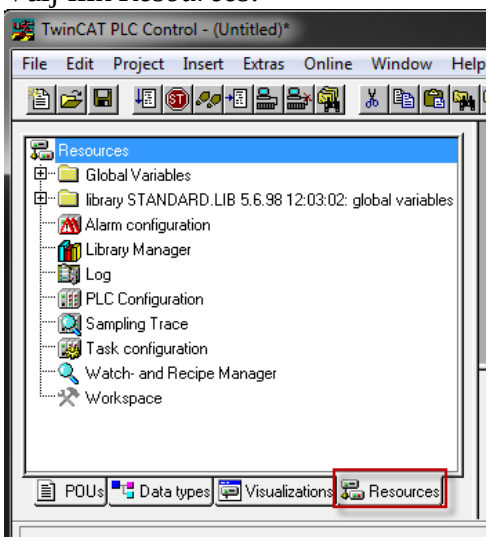
Persistent data kommer att skrivas till fil i Boot mappen på positiv flank när bStart i exemplet ovan går till.

5 Exempel 2, Persistent data och UPS med funktionsblock för en CX5020.

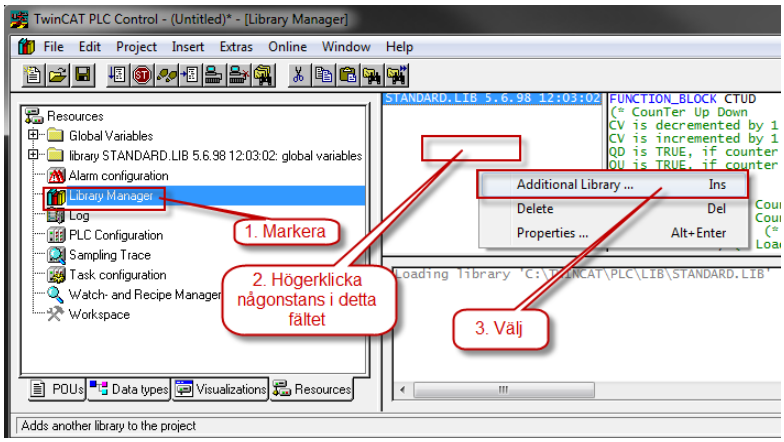
CX50x0 har inbyggd 1-sekund UPS med kapacitet att lagra 1 MB persistent data och göra en QuickShutdown av operativet vid spänningsbortfall. Det går att integrera 1-sekund UPS även till CP62xx, CP77xx och C6915 som tillval. Rekommendationen är att använda funktionsblocket *FB_S_UPS* som finns i biblioteket *TcSUPS.*
Detta bibliotek följer med vid installation av TwinCAT 2.11 (build 2021 och högre) men behöver länkas in i mjukvaran.
Funktionsblocket *FB_WritePersistentData* kan nyttjas för att vid önskat tillfälle skriva ner den Persistenta datan till minnet. Notera att det endast är persistent data, inte retain data som skrivs med detta funktionsblock. Det går att kombinera funktionsblocket *FB_S_UPS* och funktionsblocket *FB_WritePersistentData* som hanterades i exempel 1 ovan, funktionsblocken skall ha unika instansnamn.

5.1 Lägg till bibliotek i PLC Control

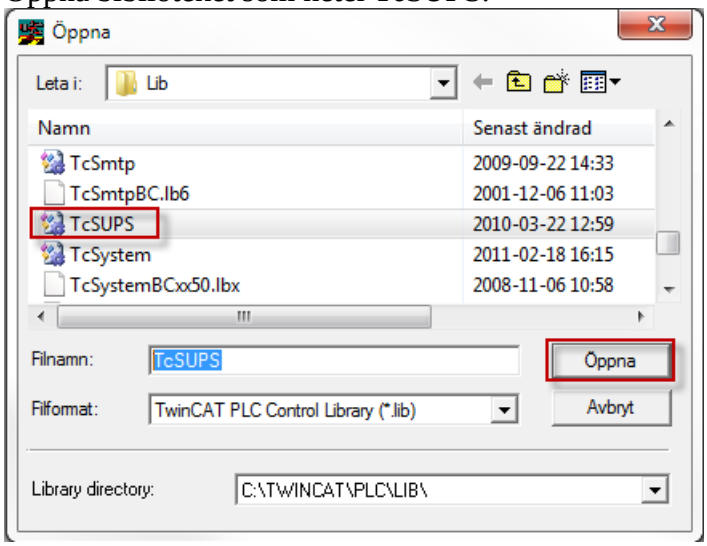
Börja med att addera biblioteket i TwinCAT - PLC Control. Skapa ett nytt projekt gå in i menyn File – New.
Välj flik *Resources*.



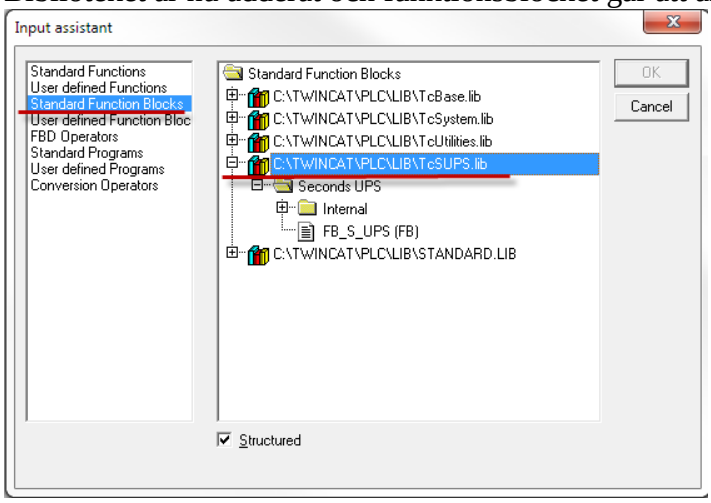
Markera *Library Manager*, dubbelklicka på den, högerklicka någonstans i det vita fältet (se bilden nedan) och välj *Additional Library*.



Öppna biblioteket som heter *TcSUPS*.

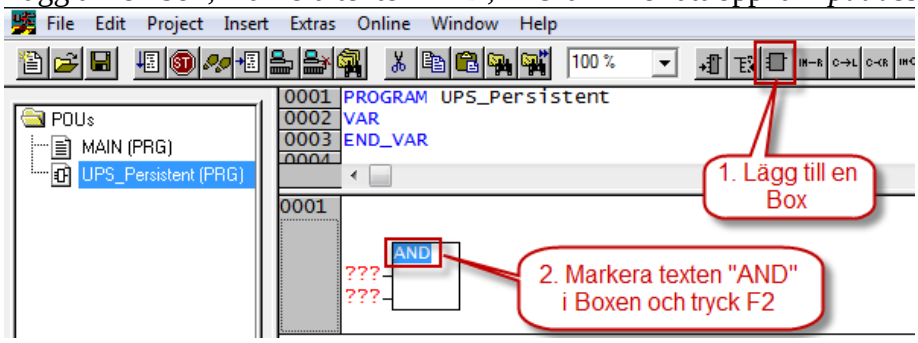


Biblioteket är nu adderat och funktionsblocket går att använda i projektet.

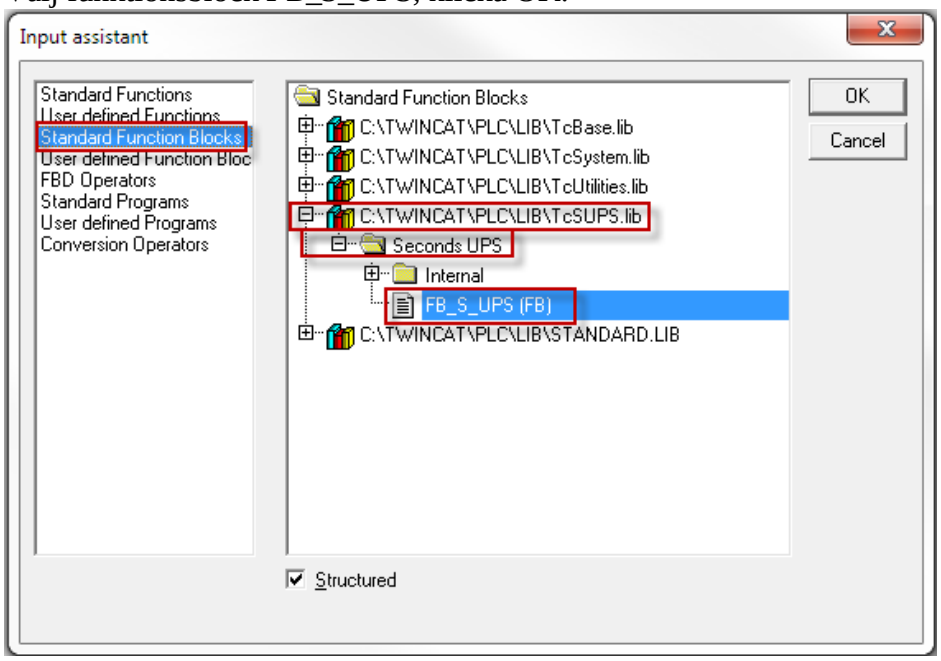


I programexemplet är *Main* programmet skapat i Strukturerad Text och programdelen med Persistent data funktionsblock är skapat i Funktions Blocks Diagram editorn.

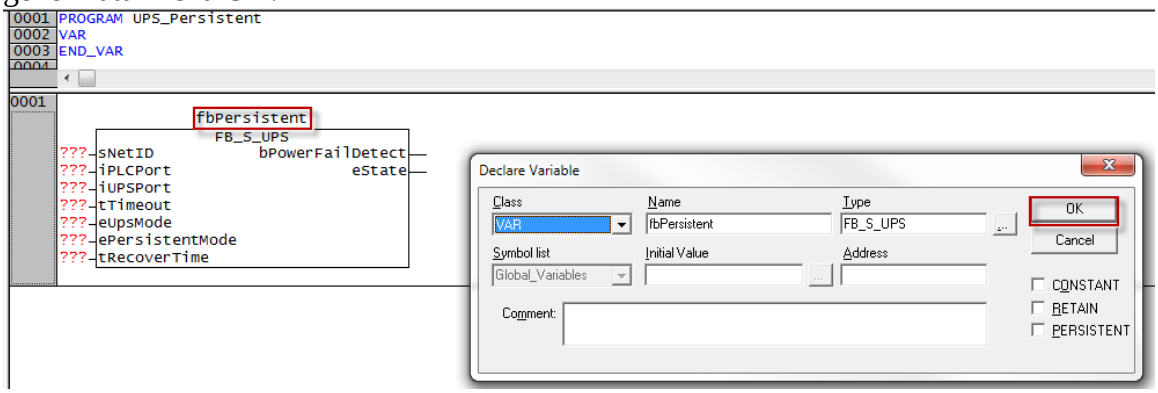
Lägg till en box, markera texten *AND*, klicka F2 för att öppna *Input assistant*.



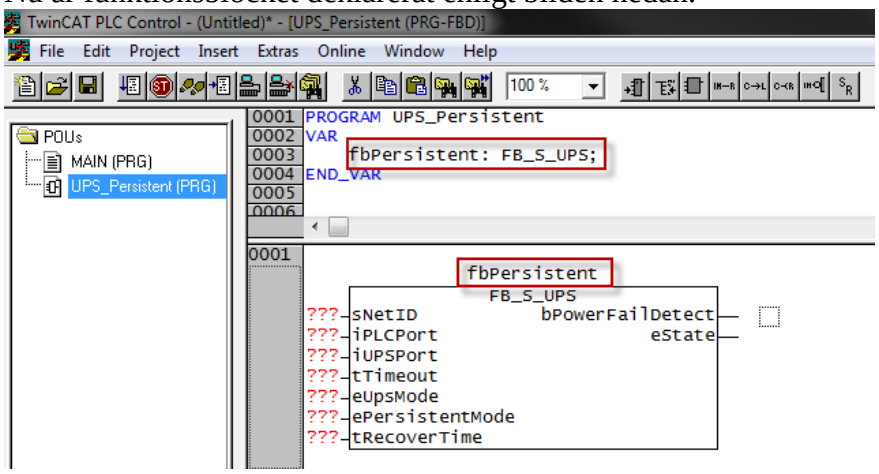
Välj funktionsblock *FB_S_UPS*, klicka OK.



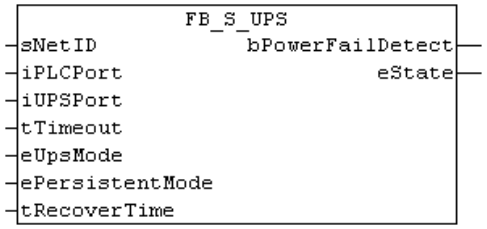
Funktionsblocket skall deklareras med ett instansnamn, i exemplet nedan heter den *fbPersistent*. Skriv namnet ovan funktionsblocket och tryck på Enter, deklarera variabeln genom att klicka OK.



Nu är funktionsblocket deklarerat enligt bilden nedan.



5.2 Beskrivning av funktionsblocket *FB_S_UPS*



Beskrivning på ingångspinnar till funktionsblocket *FB_S_UPS*

Var Input	Typ	Beskrivning	
sNetID	T_AmsNetId	AmsNetId adress, på lokal PC skriv en tom sträng ’’	
iPLCPort	UINT	ADS portnummer på den PLC runtime persistent data skall skrivas till	
iUPSPort	UINT	16#4A8 port för att läsa Power state på UPS	
tTimeout	Time	ADS Timeout	
eUpsMode	E_PersistentMode	eSUPS_WrPersistData_Shutdown (grundinställning):	Skriver persistent data och därefter genomförs en QuickShutdown
		eSUPS_WrPersistData_NoShutdown	Endast skrivning av persistent data (ingen QuickShutdown)
		eSUPS_ImmediateShutdown	Endast QuickShutdown (ingen skrivning av persistent data)
		eSUPS_CheckPowerStatus	Endast status kontroll (ingen skrivning av persistent data eller QuickShutdown)
ePersistentMode	E_PersistentMode	Persistent data mode	
		SPDM_2PASS := 0 *1)	All persistent data är från samma PLC cykel (skrivs ner under ett scanvarv). Kan när skrivning sker påverka så att PLC cykel tid överskrids
		SPDM_VAR_BOOST := 1	Data i varje persistent variabel är från samma PLC cykel (skrivning delas upp under flera scanvarv). Rekommenderas ej med 1-sekunds UPS:en
tRecoverTime	Time	Tid för att återhämta sig från korta strömavbrott i läge eSUPS_WrPersistData_NoShutdown / eSUPS_CheckPowerStatus	

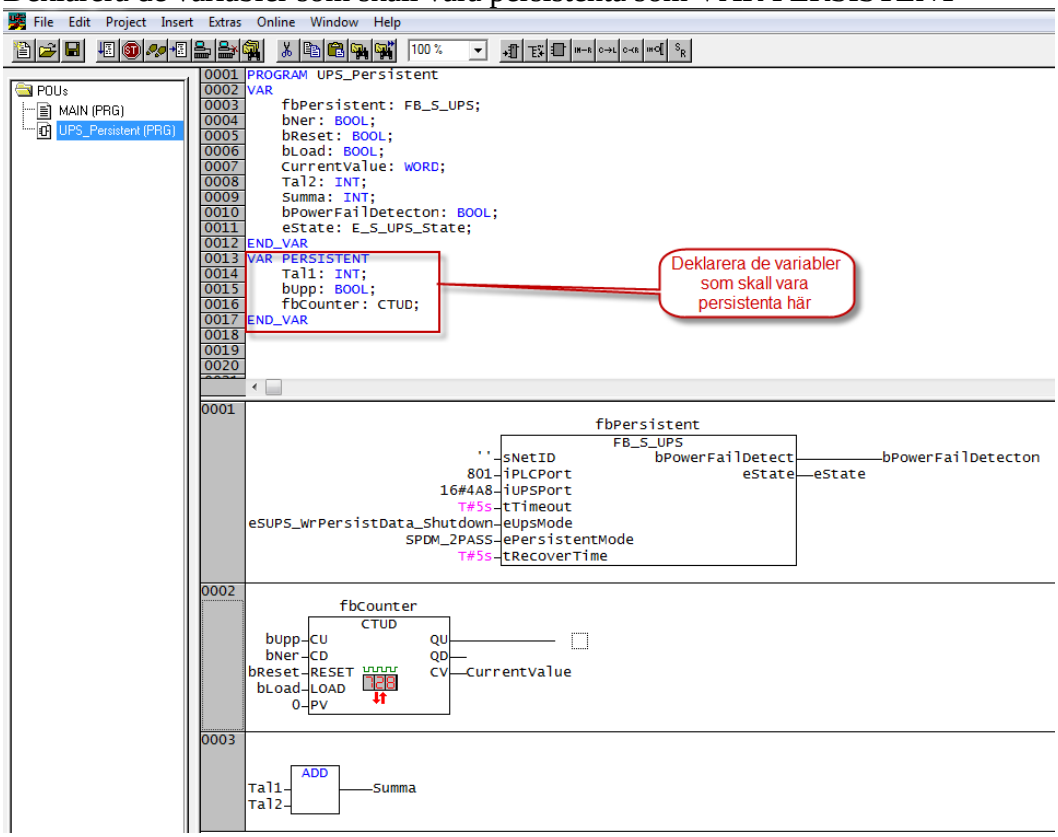
*1) Rekommenderas i normalfallet. Med 1-sekund UPS bör endast grundinställd parameter SPDM_2PASS användas, detta för att undvika att kondensatorn hinner ladda ur innan all data hunnit skrivas vilket skulle kunna bli fallet om det är mycket data och alternativet SPDM_VAR_BOOST valts.

Beskrivning på utgångspinnar till funktionsblocket *FB_S_UPS*

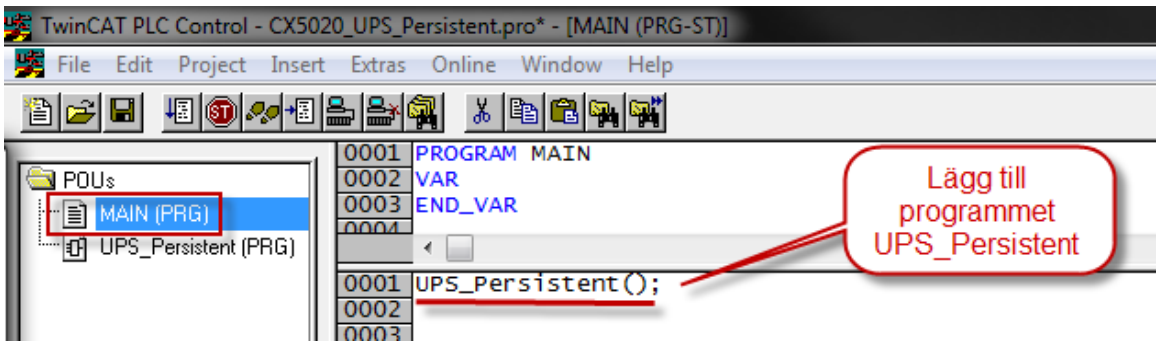
Var Output	Typ	Beskrivning
bPowerFailDetect	BOOL	Denna bit är till då spänningsbortfall detekteras
eState	E_S_UPS_State	eSUPS_PowerOK: Spänning är OK
		eSUPS_PowerFailure: Spänningsbortfall detekterats (endast till ett scanvarv)
		eSUPS_WritePersistentData: Skriving av persistent data är aktiv
		eSUPS_QuickShutdown: QuickShutdown är aktiv
		eSUPS_WaitForRecover: väntar i 10s innan PowerOK går till igen (gäller endast då mode valts utan Shutdown)
		eSUPS_WaitForPowerOFF: Väntar på att stänga av PC med UPS

5.3 Exempelkod i TwinCAT PLC Control för CX5020

Deklarera de variabler som skall vara persistenta som VAR PERSISTENT



När PLC koden är klar skall programdelen *UPS_Persistent* läggas till i *Main* programmet, se bild nedan:

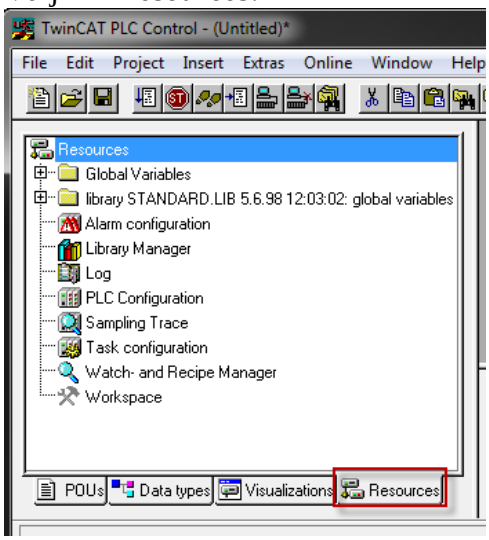


6 Exempel 3, Persistent data och UPS med funktionsblock för en CX8090.

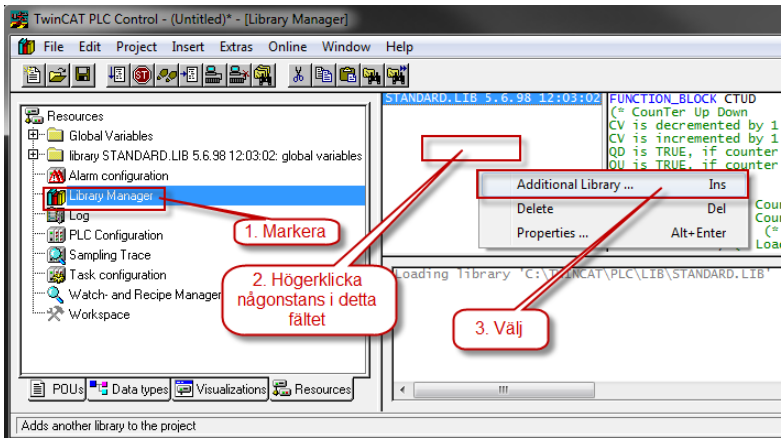
CX80x0 har inbyggd 1-sekund UPS med kapacitet att lagra 1 MB persistent data och göra en QuickShutdown av operativet vid spänningsbortfall. Rekommendationen är att använda funktionsblock som finns i biblioteket *TcSystemCX80xx*.
Detta bibliotek finns med vid installation av TwinCAT 2.11 R3 men behöver länkas in i mjukvaran.
Funktionsblocket *FB_WritePersistentData* kan nyttjas för att vid önskat tillfälle skriva ner den persistenta datan till minnet. Notera att det endast är Persistent data, inte Retain data som skrivs med detta funktionsblock. Det går att kombinera funktionsblocket *FB_S_UPS_CX80xx* och funktionsblocket *FB_WritePersistentData* som hanterades i exempel 1 ovan, funktionsblocken skall ha unika instansnamn.

6.1 Lägg till bibliotek i PLC Control

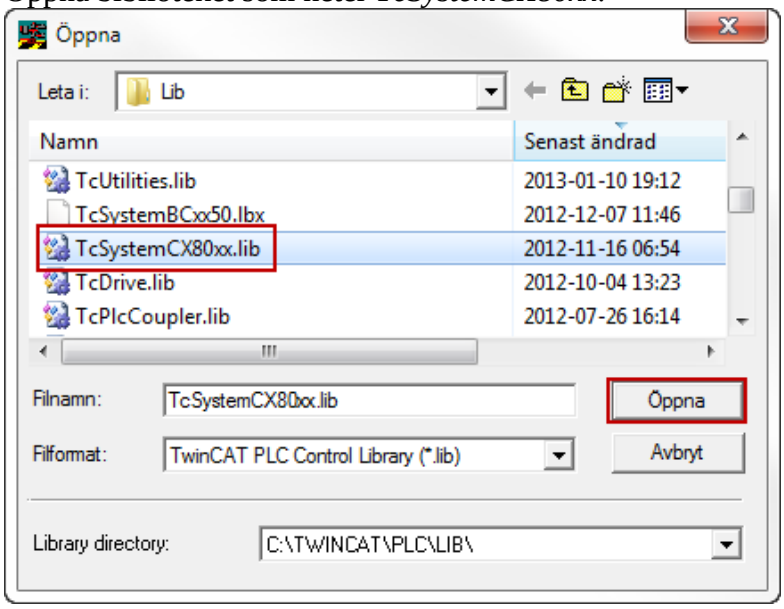
Börja med att addera biblioteket i TwinCAT - PLC Control. Skapa ett nytt projekt gå in i menyn File – New.
Välj flik *Resources*.



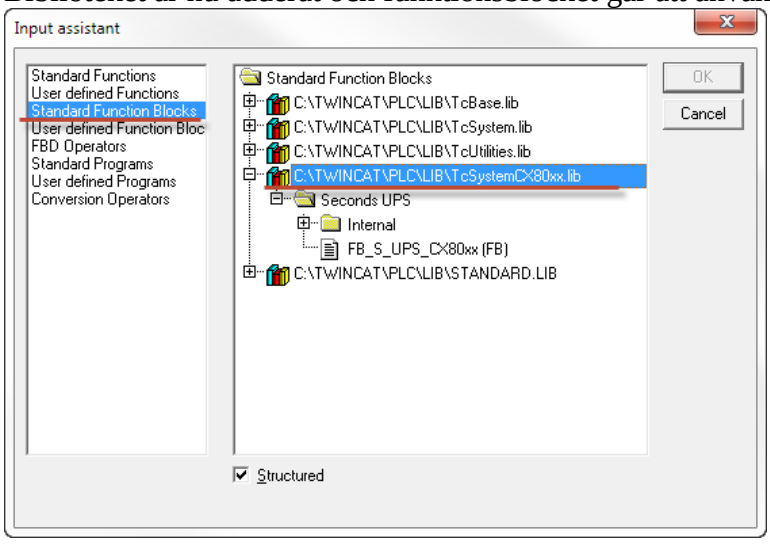
Markera *Library Manager*, dubbelklicka på den, högerklicka någonstans i det vita fältet (se bilden nedan) och välj *Additional Library*.



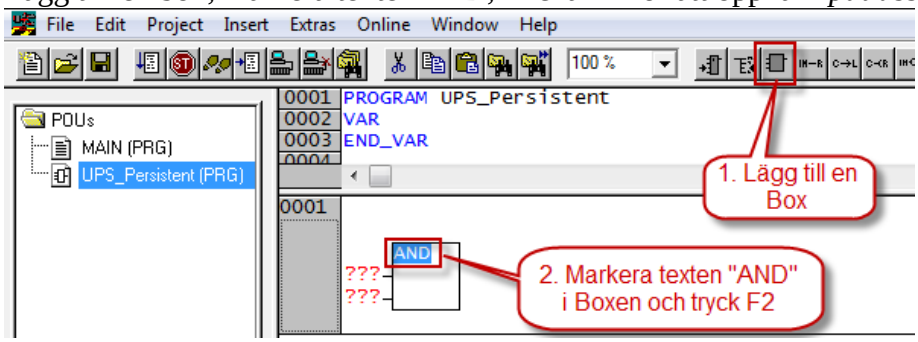
Öppna biblioteket som heter *TcSystemCX80xx*.



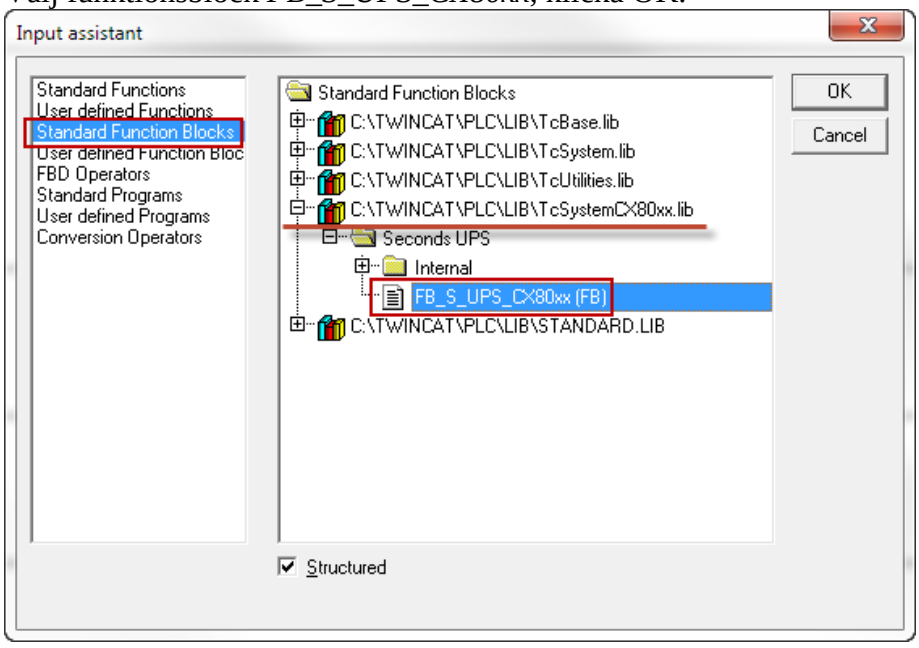
Biblioteket är nu adderat och funktionsblocket går att använda i projektet.



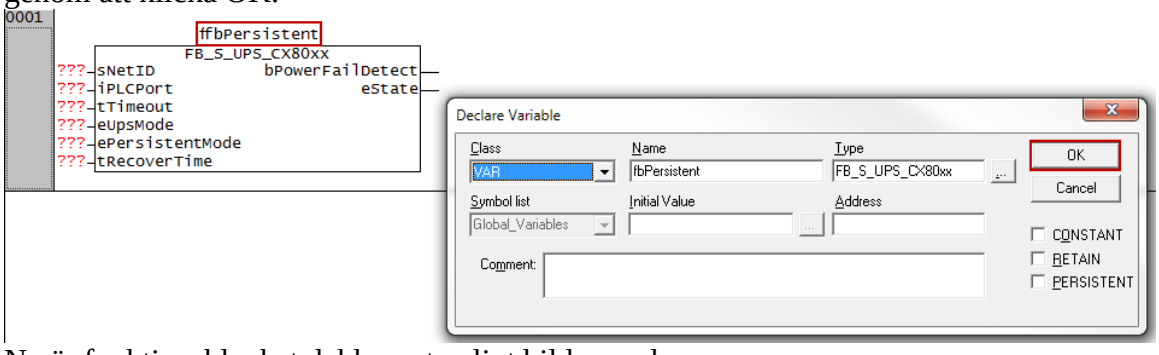
I programexemplet är *Main* programmet skapat i Strukturerad Text och programdelen med Persistent data funktionsblock är skapat i Funktions Blocks Diagram editorn. Lägg till en box, markera texten *AND*, klicka F2 för att öppna *Input assistant*.



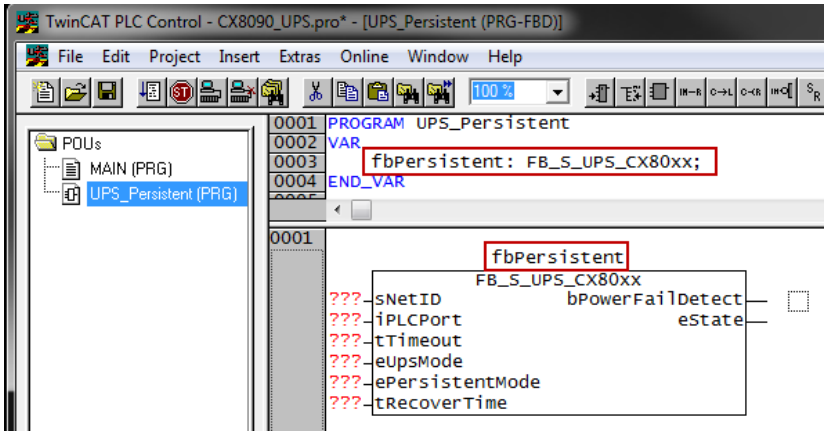
Välj funktionsblock *FB_S_UPS_CX80xx*, klicka OK.



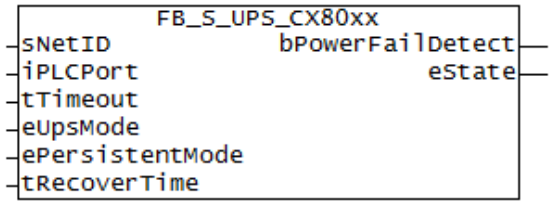
Funktionsblocket skall deklareras med ett instansnamn, i exemplet nedan heter den *fbPersistent*. Skriv namnet ovan funktionsblocket och tryck på Enter, deklarera variabeln genom att klicka OK.



Nu är funktionsblocket deklarerat enligt bilden nedan.



6.2 Beskrivning av funktionsblocket *FB_S_UPS_CX80xx*



Beskrivning på ingångspinnar till funktionsblocket *FB_S_UPS_CX80xx*

Var Input	Typ	Beskrivning	
sNetID	T_AmsNetId	AmsNetId adress, på lokal PC skriv en tom sträng ’’	
iPLCPort	UINT	ADS portnummer på den PLC runtime persistent data skall skrivas till	
tTimeout	Time	ADS Timeout	
eUpsMode	E_PersistentMode	eSUPS_WrPersistData_Shutdown (grundinställning):	Skriver persistent data och därefter genomförs en QuickShutdown
		eSUPS_WrPersistData_NoShutdown	Endast skrivning av persistent data (ingen QuickShutdown)
		eSUPS_ImmediateShutdown	Endast QuickShutdown (ingen skrivning av persistent data)
		eSUPS_CheckPowerStatus	Endast status kontroll (ingen skrivning av persistent data eller QuickShutdown)
ePersistentMode	E_PersistentMode	Persistent data mode	
		SPDM_2PASS := 0 *1)	All persistent data är från samma PLC cykel (skrivs ner under ett scanvarv). Kan när skrivning sker påverka så att PLC cykel tid överskrids
		SPDM_VAR_BOOST := 1	Data i varje persistent variabel är från samma PLC cykel (skrivning delas upp under flera scanvarv). Rekommenderas ej med 1-sekunds UPS:en
tRecoverTime	Time	Tid för att återhämta sig från korta strömavbrott i läge eSUPS_WrPersistData_NoShutdown / eSUPS_CheckPowerStatus	

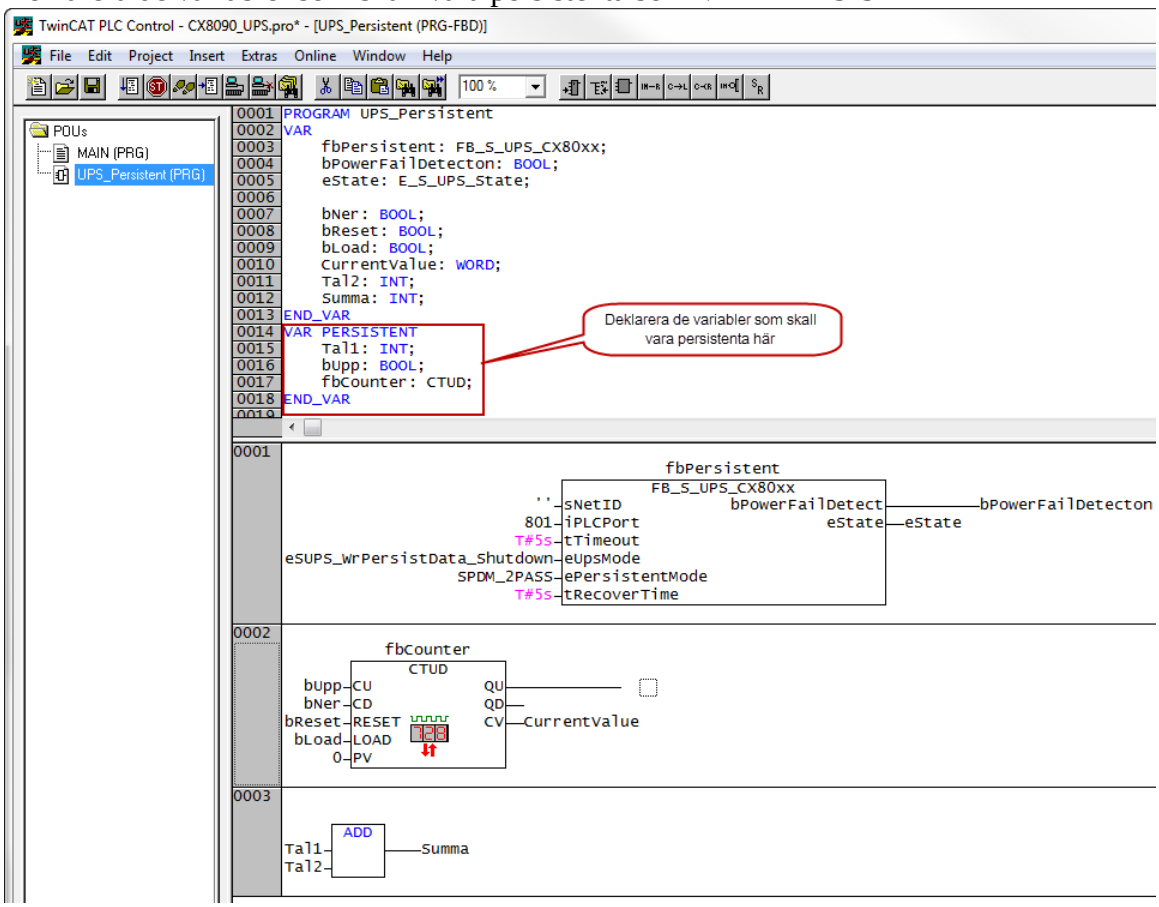
*1) Rekommenderas i normalfallet. Med 1-sekund UPS bör endast grundinställd parameter SPDM_2PASS användas, detta för att undvika att kondensatorn hinner ladda ur innan all data hunnit skrivas vilket skulle kunna bli fallet om det är mycket data och alternativet SPDM_VAR_BOOST valts.

Beskrivning på utgångspinnar till funktionsblocket *FB_S_UPS_CX80xx*

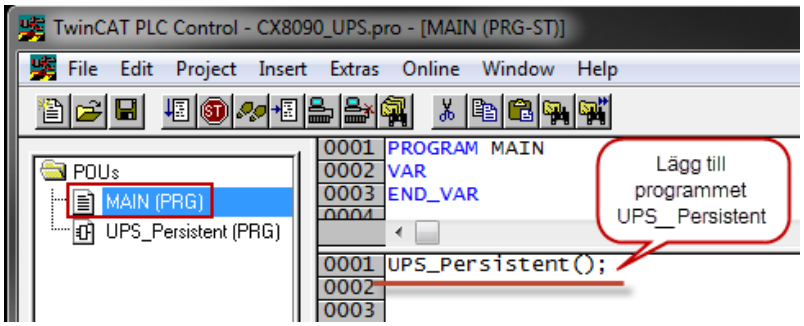
Var Output	Typ	Beskrivning
bPowerFailDetect	BOOL	Denna bit är till då spänningsbortfall detekteras
eState	E_S_UPS_State	eSUPS_PowerOK: Spänning är OK
		eSUPS_PowerFailure: Spänningsbortfall detekterats (endast till ett scanvarv)
		eSUPS_WritePersistentData: Skriving av persistent data är aktiv
		eSUPS_QuickShutdown: QuickShutdown är aktiv
		eSUPS_WaitForRecover: väntar i 10s innan PowerOK går till igen (gäller endast då mode valts utan Shutdown)
		eSUPS_WaitForPowerOFF: Väntar på att stänga av PC med UPS

6.3 Exempelkod i TwinCAT PLC Control för CX8090

Deklarera de variabler som skall vara persistenta som VAR PERSISTENT



När PLC koden är klar skall programdelen *UPS_Persistent* läggas till i *Main* programmet, se bild nedan:



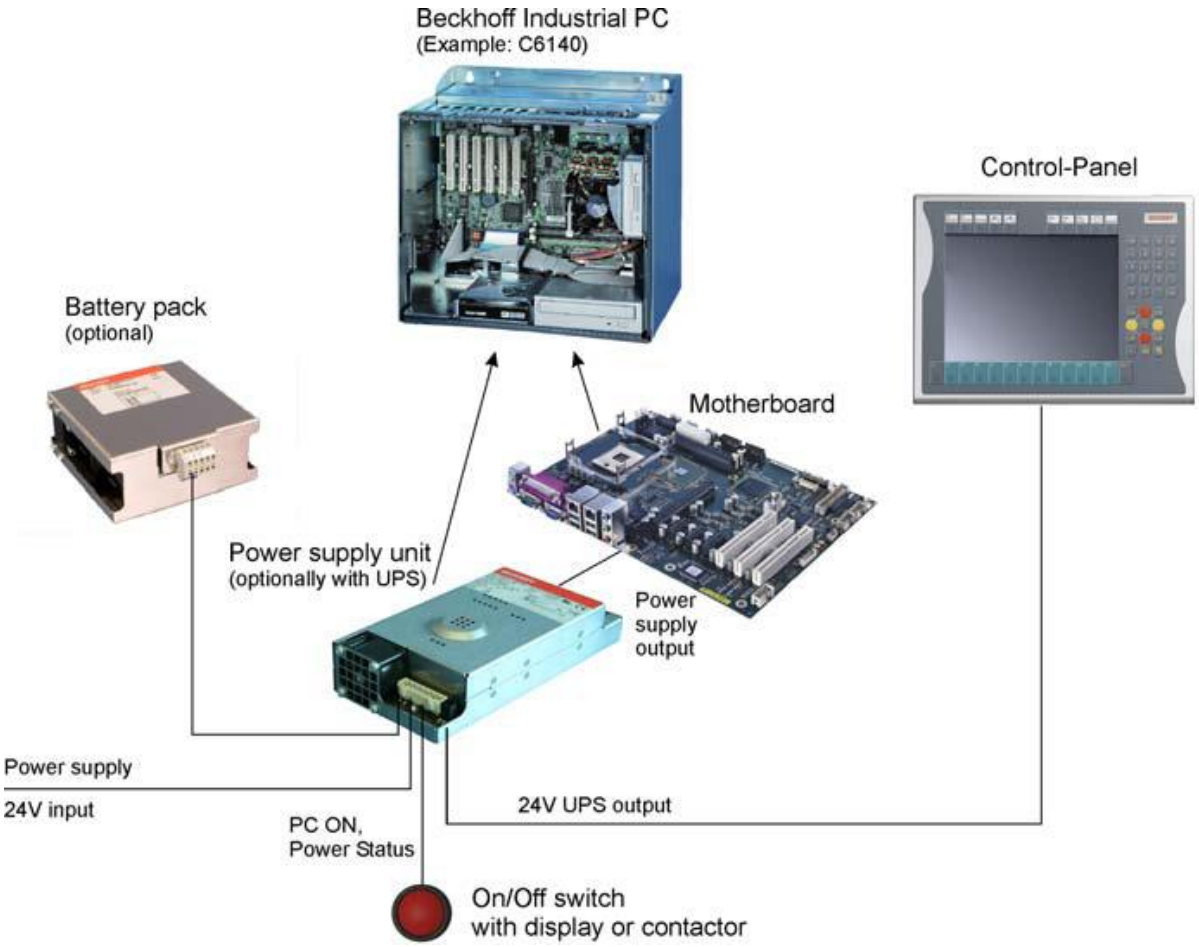
UPS

Används ingen UPS och spänningen till systemet bryts kommer TwinCAT inte att stängas ner korrekt och ingen skrivning sker av Retain eller Persistent data. Vid återstart kommer variablerna att initieras med senast sparad data (backupdata) som kan vara gammal, från senaste korrekta TwinCAT nedstängning. Med UPS kommer informationen om ett spänningsbortfall att ges till TwinCAT System Service som stänger ner TwinCAT kontrollerat och skriver ner Persistent och Retain data till respektive fil. Efter lagring av data kommer TwinCAT System Service att stänga ner Windows operativsystem.
Det finns olika UPS som kan användas till Beckhoff systemen:

- Standard UPS (C9900-U330 + (C9900-U209 eller C9900-P209))
  - 24V UPS med batteripaket till Beckhoff Industri PC och Control Paneler
- CX1190-UPS (CX1100-09x0)
  - För CX1010, CX1020 och CX1030
- CX2100-UPS (CX2100-0904)
  - För CX2020, CX2030 och CX2040
- 1 sekund UPS (gör en Quickshutdown)
  - Finns inbyggt i CX50x0 och CX80x0
  - Tillval (C9900-U211) till Control Panel CP62xx, CP77xx med Intel Atom moderkort
  - Tillval (C9900-U212) till Control Panel CP26xx, CP66xx med ARM processor
  - Tillval till Industri PC C6915

Models with 1 second UPS	
CX5000	x86 Embedded PC  1 second UPS inclusive
CX8000	Embedded PC  1 second UPS inclusive
CP62xx	„Economy“ built-in Panel PC  1 second UPS optional
CP77xx	CP77xx-0030 | Panel PC IP 65  1 second UPS optional
C6915	Control cabinet Industrial PC  1 second UPS optional
CP26xx	CP26xx-0000 | Panel PC with ARM Cortex  1 second UPS optional
CP66xx	„Economy“ built-in Panel PC  1 second UPS optional

Exempel med UPS



Fel i och förbättringar av detta dokument meddelas till support@beckhoff.se.