

Minneslagring med Novram

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 data skall lagras, mängden data samt vilken typ av hårdvara som skall användas. Detta dokument beskriver endast minneslagring med Novram som kan användas både med och utan funktionsblock.

Teknisk bakgrund

Dokumentet skall användas som hjälp för att använda Novram-minnet i Beckhoff-systemen, beskrivningen är endast överskådlig och ej fullständig. Det Beckhoff-system som har använts till exemplen i detta dokument är en CX9010 med inbyggt Novram. Biblioteket som behöver installeras i TwinCAT är *TcIoFunctions.Lib*. Notera att detta gäller endast i exemplet då funktionsblock används.

1 Novram – Icke flyktigt minne

Novram-minnet användas för backup av t.ex. driftdata från PLC:n. Novram är ett minne som kan lagra data utan begränsat antal skrivningar. När PLC koden exekveras görs skrivningar cykliskt till arbetsminnet, RAM minnet. RAM minnet bibehåller ingen data då spänningen bryts till systemet. För att lagra driftdata eller annan data får denna kopieras från RAM-minnet till Novram och vid spänningtillslag laddas datan på nytt tillbaka till RAM minnet. Läsning och skrivning kan kontrolleras av PLC programmet med hjälp av funktionsblock eller genom att länka variabler i System Managern. Används funktionsblock kan denna triggas av en digital signal eller läggas i en egen task med cykliskt intervall. Skrivning till Novram tar CPU-kraft och är det mycket data som skall sparas bör detta delas upp så inte all data skrivs på en gång, detta går att göra med flera funktionsblock. Planera även hur ofta skrivning ska ske till Novram.

Notera att PLC-programmet inte lagras i Novram, det sparas genom skicka ner ett separat boot-projekt till PLC:n.

2 Översikt på enheter med Novram-minne

Tabellen nedan visar de olika interna minnen som finns tillgängliga för olika CX-modeller samt även vilka extra minneskort som finns till Beckhoffs industri PC.

Beckhoff-system	Beskrivning	Novram storlek
CX		
CX9000	CPU med inbyggd spänningsaggregat och Novram	128 kbyte
CX9001	CPU med inbyggd spänningsaggregat och Novram	128 kbyte
CX9010	CPU med inbyggd spänningsaggregat och Novram	128 kbyte
CX1010	Novram integrerat i separat sp-aggregat CX1100-000x	8 kbyte
CX1020	Novram integrerat i separat sp-aggregat CX1100-000x	8 kbyte
CX1030	Novram integrerat i separat sp-aggregat CX1100-001x	8 kbyte
CX5010/CX5020		Inget Novram Integrerad Seconds UPS, 1 Mbyte på CF kort
PC		
PC Fältbuss kort PCI	FC3101-0002 Profibus DP	32 kbyte
	FC3102-0002 Profibus DP	
	FC5101-0002 CANopen	
	FC5102-0002 CANopen	
	FC5201-0002 DeviceNet	
	FC5202-0002 DeviceNet	
PC fältbuss kort Mini PCI IP 20 till C692x, CP62xx, CP66xx. Mini PCI plug-in kort kan bara beställas tillsammans med en Beckhoff PC, eftermonteras inte.	FC3151-0002 Profibus DP	128 kbyte
	FC5151-0002 CANopen	
	FC5251-0002 DeviceNet	
	FC7551-0002 Sercos	
Panel PC		
CP67xx-0020, CP77xx-0020	C9900-R229	256 kbyte
Tillbehör till Panel PC series CP62xx, CP64xx, CP65xx, CP67xx, CP71xx, CP72xx, CP77xx Control cabinet Industri PC. Control Panel, Mini PCI plug-in kort Novram. Går att eftermontera.	C9900-R230	128 kbyte
	C9900-R231	256 kbyte
	C9900-R232	512 kbyte

PC fältbuss kort Mini PCI IP 65 till CP72xx. Mini PCI plug-in kort kan bara beställas tillsammans med en Beckhoff PC, eftermonteras inte.	FC3151-0012 Profibus DP	128 kbyte
	FC5151-0012 CANopen	
	FC5251-0012 DeviceNet	
	FC7551-0012 Sercos	
Terminal		
EL6080	EtherCAT terminal med Novram	128 kbyte

1 kbyte = 1024 byte

Exempel på lagningskapacitet:

32 kbyte = 32768 byte => 16384 word (eller 16-bitars tal)

Fördelar med Novram:

- Det är enkelt att använda
- Det finns färdiga funktionsblock
- Ej batterimatat minne

Nackdelar:

- Begränsad storlek på minnet.
- Datan kan inte editeras utanför PLC programmet, nackdel vid recepthantering.
- CPU belastningen blir hög om mycket data skrivs till Novram. (I det fallet är det bättre att använda funktionsblock och dela upp belastningen på flera skrivningar).

Följande avsnitt visar två exempel på hur Novram kan nyttjas, det första om funktionsblock används och det andra om länkningen görs direkt i System Managern utan funktionsblock.

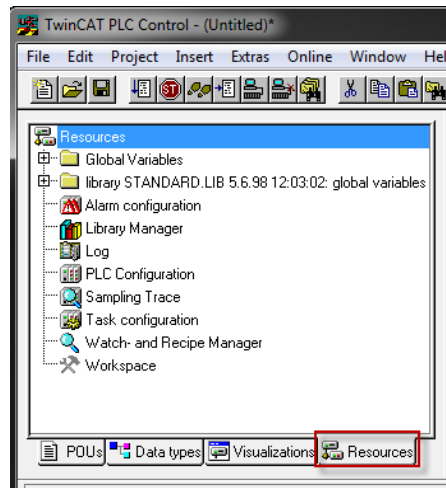
3 Exempel 1, med Novram och funktionsblock

Biblioteket med funktionsblock för hantering av Novram (*TcIoFunctions.Lib*) följer med vid installation av TwinCAT men behöver länkas in i mjukvaran. Notera att valet av funktionsblock styrs av vilken CPU systemet använder.

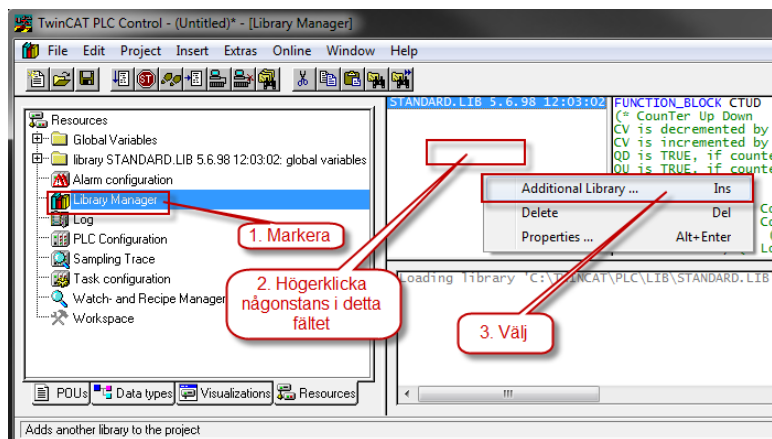
Val av CPU	Använd funktionsblock	TwinCAT version
FCxxx kort CX med Intel (x86 processor) EJ för CX med ARM processor.	FB_NovRAMReadWrite	TwinCAT v2.8.0 Build > 722
FCxxx kort CX med Intel (x86 processor) CX med ARM processor	FB_NovRAMReadWriteEx	TwinCAT v2.10.0 Build > 1231

Programexemplet som bifogas är skapat för en CX9010 med ARM processor, dvs funktionsblocket *FB_NovRAMReadWriteEx* skall användas.

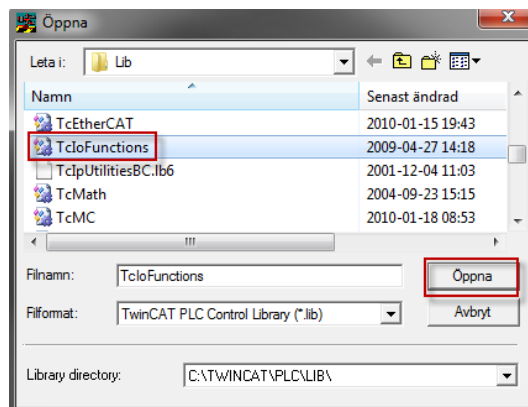
Starta TwinCAT PLC Control, skapa ett nytt projekt gå in i menyn File – New. Addera in biblioteket *TcIoFunctions.Lib* genom att välja flik *Resources*.



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

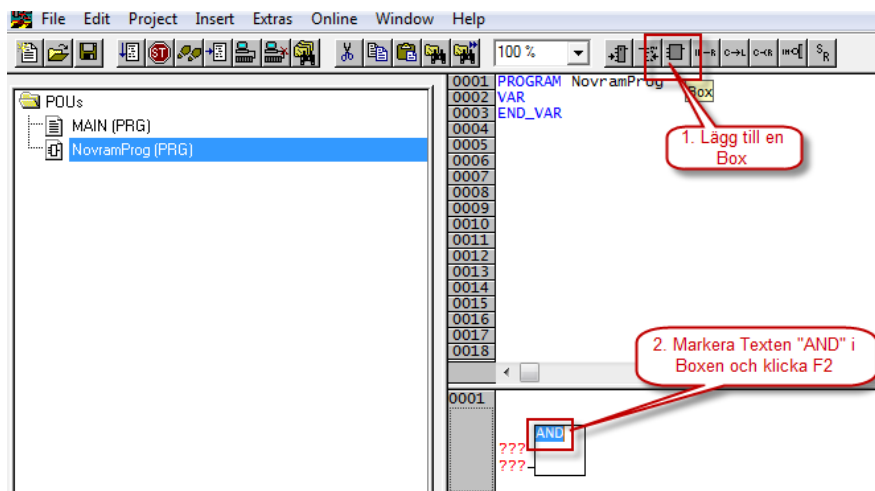


Öppna biblioteket som heter *TcIoFunctions*.

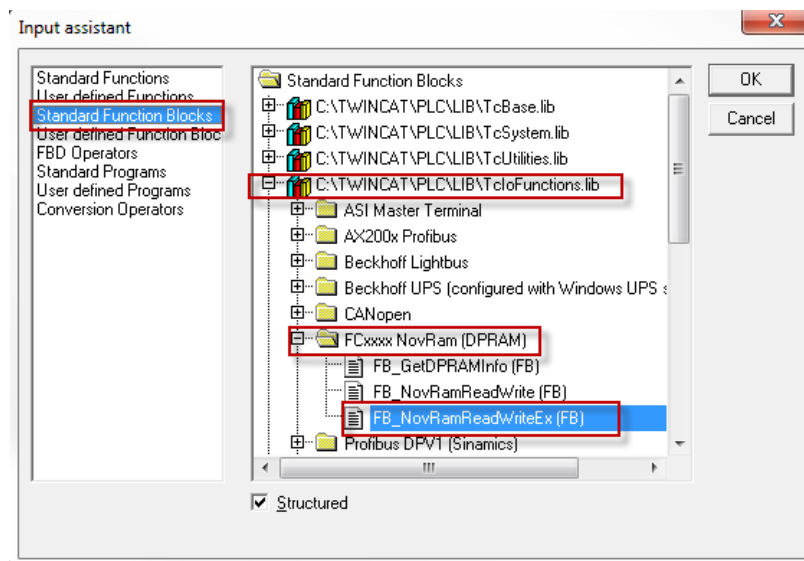


Biblioteket är nu adderat och funktionsblocken går att använda i projektet.

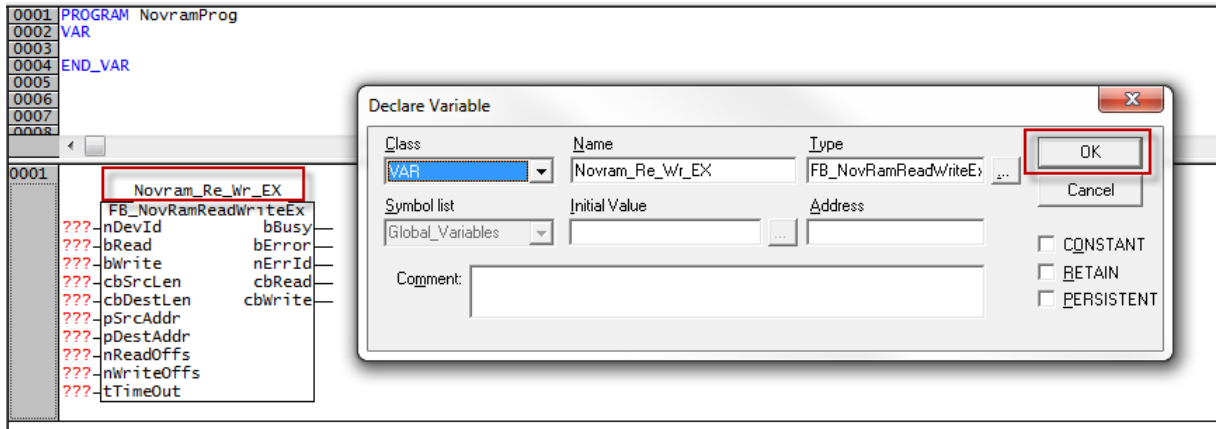
I programexemplet är *Main* programmet skapat i Strukturerad Text och programdelen med Novram funktionsblock är skapat i Funktions Blocks Diagram editorn. Lägg till en box, markera texten *AND*, klicka F2 för att öppna *Input assistant*.



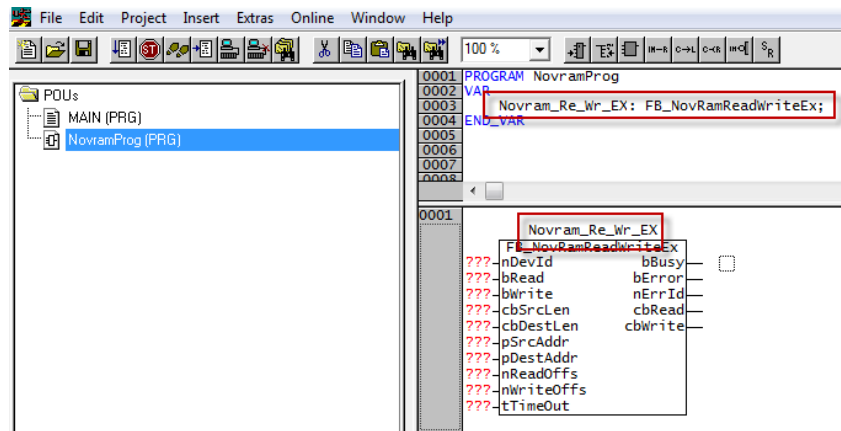
Välj funktionsblock *FB_NovRamReadWriteEx*, klicka OK.



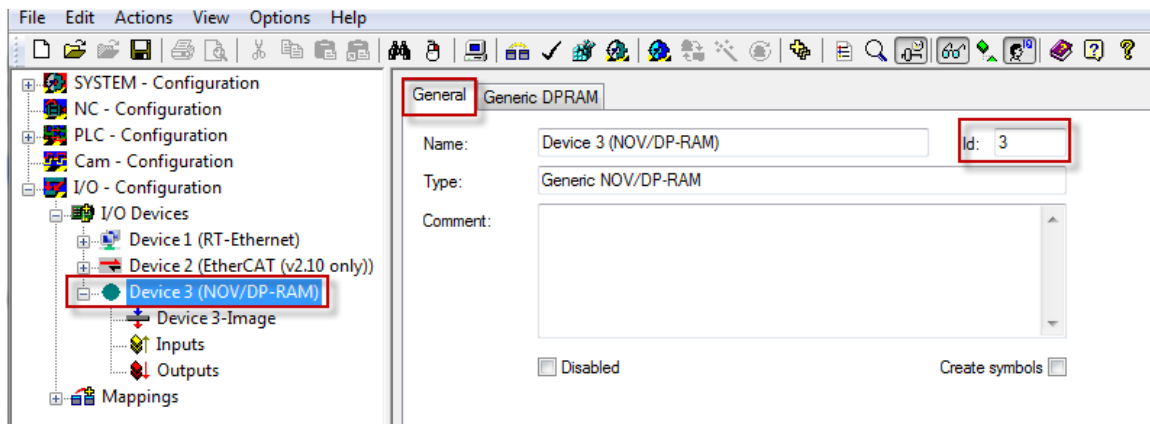
Funktionsblocket skall deklaras med ett instansnamn, i exemplet nedan heter den *Novram_Re_Wr_EX*. Skriv namnet ovan funktionsblocket och tryck på Enter, deklarerar variabeln genom att klicka OK.



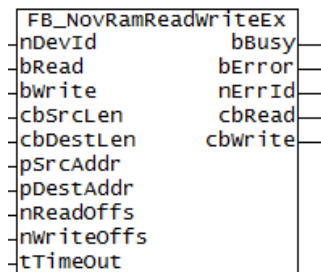
Nu är funktionsblocket deklarerat enligt bilden nedan.



Nästa steg är att starta System Managern om detta inte är gjort. Skanna upp systemet (i detta exempel en CX9010) med hjälp av System Manager. Markera *NOV/DP-RAM* välj fliken *General* och här under *Id* utläses *device id*, i detta exempel 3. Detta nummer skall anges på ingångspinnen *nDevId* till funktionsblocket i TwinCAT PLC Control projektet.



3.1 Beskrivning av funktionsblocket *FB_NovRamReadWriteEx*



Beskrivning på ingångspinnar till funktionsblocket *FB_NovRamReadWriteEx*:

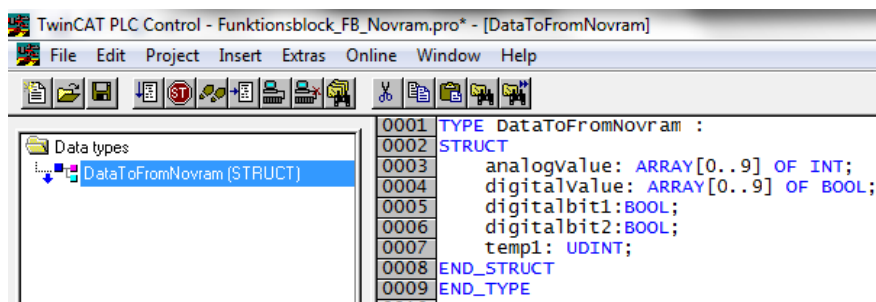
Var Input	Typ	Beskrivning
nDevId	UDINT	Novram device ID , utläses i System Managern
bRead	BOOL	Positiv flank på denna bit aktiverar läsning från Novram
bWrite	BOOL	Positiv flank på denna bit aktiverar skrivning till Novram
cbSrcLen	UDINT	Längd på data (byte) som skall skrivas till Novram
cbDestLen	UDINT	Längd på data (byte) som skall läsas från Novram
pSrcAddr	UDINT	Adresspekare som pekar på en data buffert som innehåller data som skall skrivas till Novram. Adresspekaren fås genom ADR() operator.
pDestAddr	UDINT	Adresspekare som pekar på en data buffert där data som skall läsas från Novram skall placeras. Adresspekaren fås genom ADR() operator.
nReadOffs	UDINT	Läs offset på minnesarean i Novram minnet
nWriteOffs	UDINT	Skriv offset på minnesarean i Novram minnet
tTimeOut	Time	Ange längden på den timeout som inte får överskridas under exekveringen av funktionsblocket.

Beskrivning på utgångspinnar till funktionsblocket *FB_NovRamReadWriteEx*:

Var Output	Typ	Beskrivning
bBusy	BOOL	Denna bit är satt så länge funktionsblocket exekveras, så länge skrivning/läsning pågår
bError	BOOL	Felflagga
nErrId	UDINT	Om bError är satt kan ADS felkoden utläsas i denna (" ADS Return Codes ").
cbRead	UDINT	Antalet korrekt lästa byte
cbWrite	UDINT	Antalet korrekt skrivna byte

Gå tillbaka till PLC Control och programkoden. För att smidigt samla all data som skall till och från Novram-minnet går det att lägga allt i en datastruktur, då behöver endast ett funktionsblock användas. En datastruktur i fortsättningen kallad *strukt* är en sammansättning av olika datatyper och hanteras som en variabel. Det går att lägga upp data som enskilda variabler men då kan det behövas fler funktionsblock om inte adressarean ligger i följd.

Gå in i fliken *Data types*, markera *Data types* mappen, högerklicka och välj *Add Objects*. Ge din *strukt* ett namn, i exemplet nedan heter den *DataToFromNovram*. Den kan innehålla olika datatyper, se bilden i exemplet nedan:



Nästa steg är att skapa en instans av strukten, ge den ett namn så den kan användas till funktionsblocket. Deklarera också de nödvändiga variablerna för att kunna skriva och läsa till och från Novram, se exemplet nedan.

```

0001 PROGRAM NovramProg
0002 VAR
0003   Novram_Re_wr_EX: FB_NovRamReadwriteEx; (*Instance name for the functionblock FB_NovRamReadwriteEx*)
0004   (*-----*)
0005   bNovramRead: BOOL; (*Rising edge at this input, and cbDestLen data are copied from the
0006   (*Rising edge at this input, and cbSrcLen data are copied from the
0007   (*The address offset in the NOV-RAM from which data are to be read,
0008   (*The address offset in the NOV-RAM from which data are to be written,
0009   (*This output is set when the function block is activated, and remains
0010   (*This output is set if an error occurs during execution*)
0011   (*If the bError output is set, this parameter supplies the ADS error
0012   (*Number of successfully read data bytes.*)
0013   (*Number of successfully written data bytes.*)
0014   (*-----*)
0015   (*-----*)
0016   bBusyFlag: BOOL; (*This output is set when the function block is activated, and remains
0017   (*This output is set if an error occurs during execution*)
0018   (*If the bError output is set, this parameter supplies the ADS error
0019   (*Number of successfully read data bytes.*)
0020   (*Number of successfully written data bytes.*)
0021   (*-----*)
0022   testStruct:DataToFromNovram; (*This struct will contain all the data that will be written and
0023   (*This struct will contain all the data that will be written and
0024   (*-----*)
0025   bReadManualFromNovram: BOOL;
0026   END_VAR
0027
0028
0001
SystemTaskInfoArr[1].firstCycle OR bReadManualFromNovram
0002
Novram_Re_wr_EX
1-nDevId
bNovramRead-bRead
bNovramWrite-bWrite
SIZEOF(testStruct)-cbSrcLen
SIZEOF(testStruct)-cbDestLen
ADR(testStruct)-psrcAddr
ADR(testStruct)-pDestAddr
readoffs-nReadOffs
writeoffs-nWriteOffs
t#5s-tTimeout
bBusyFlag-bBusyFlag
bErrorFlag-bErrorFlag
nErrorID-nErrorID
cbRead-cbNumberReadData
cbWrite-cbNumberWriteData

```

SIZEOF som används i exemplet ovan räknar automatiskt ut längden på strukten. Alternativet är att ange en fast längd på data som skall skickas, respektive läsas. *ADR* används för att tala om var i minnet denna strukt är placerad (startadressen på datan), var funktionsblocket ska börja läsa respektive skriva.

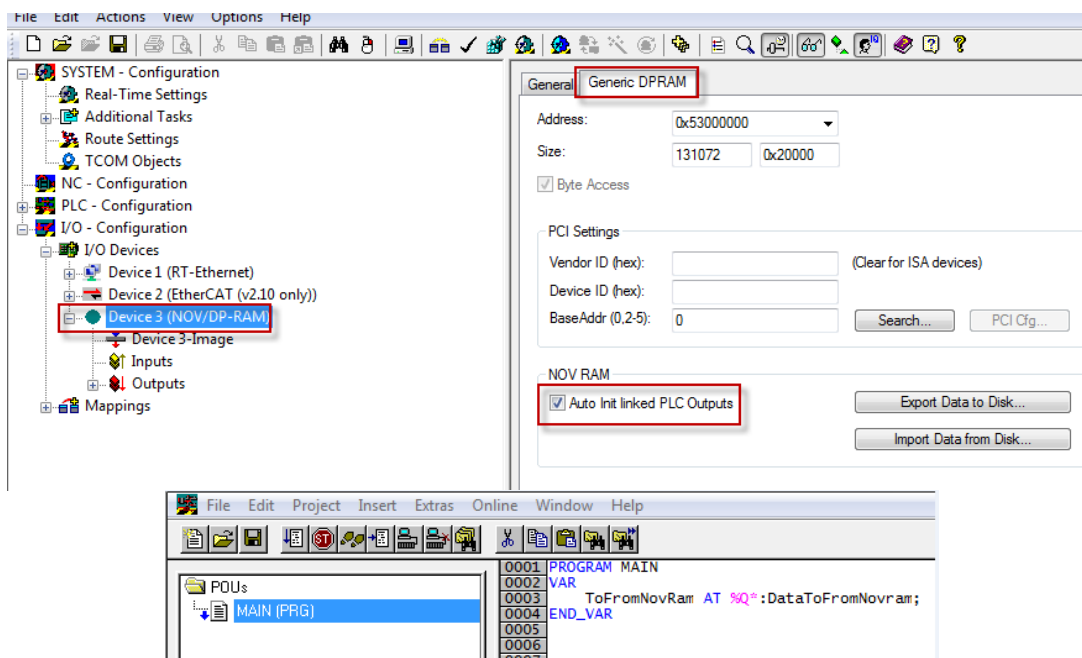
Om data ska skrivas cykliskt till Novram går det att göra om man lägger till en TON eller en Task med önskat intervall. Det går också att skriva händelsestyrt, dvs vid positiv flank på *bNovramWrite* pinne genereras en skrivning. Återläsning från Novram sker inte automatiskt när funktionsblock används utan måste triggas av ett villkor. På positiv flank av *bRead* kommer den sparade datan att kopieras tillbaka till strukten. För att läsa från Novram vid uppstart måste kod skapas, i exemplet har system biten *firstCycle* nyttjats för att läsning ska ske från Novram vid uppstart. Som exemplet är upplagt sker läsning och skrivning till samma strukt, detta går att separera och ha på olika adressareor om så önskas.

Notera att varje bit (BOOL) som läggs i en strukt kommer att reservera 1 byte i minnet och inte en bit.

4 Exempel 2, knytning till Novram görs med System Managern utan funktionsblock

Öppna System Managern och scanna av systemet, i exemplet nedan är en CX9010 använd.

Novram minnet hittades som Device 3. Markera Novram minnet och gå in under fliken *Generic DPRAM*.



Kryssa i *Auto Init linked PLC Outputs* om även läsning ska kunna ske tillbaka från Novram minnet, i annat fall sker bara skrivning till Novram. Inga värden kommer att återknytas vid t.ex. ett spänningsbortfall om rutan inte kryssar i.

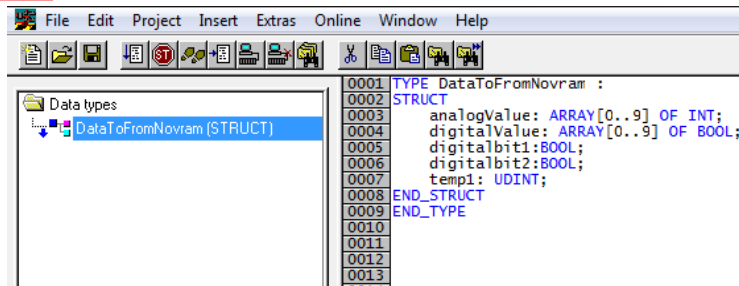
Ändra inte *Address* eller *Size* under fliken, detta scannas automatiskt upp av System Managern.

Export Data to Disk: Skriver det aktuella innehållet i Novram till en XML baserad fil på hårddisken.

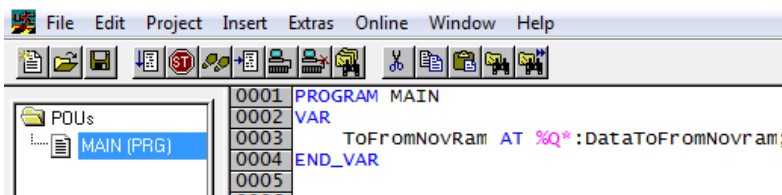
Import Data from Disk: Tidigare skapad Novram XML fil kan importeras och skrivs ner till Novram. Kan t.ex. nyttjas då flera styrsystem ser likadana ut och skall ha samma startup paramterar.

Export/Import till XML fil går endast att göra i Config mode.

Skapa en strukt som skall knytas till Novram (se exempel 1, sid 7 hur en strukt skapas). Bilder nedan från PLC Control projektet:

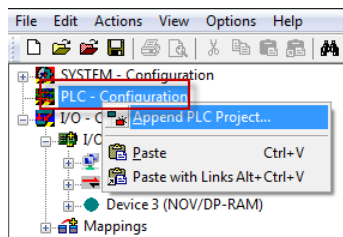


Adressera strukten AT %Q* för att kunna knyta variablerna i System Managern.

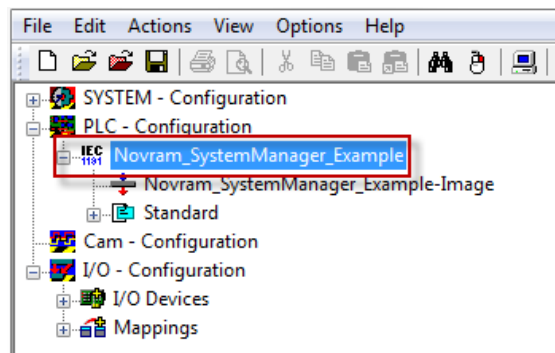


Gå tillbaka till System Managern:

Markera *PLC Configuration* och högerklicka med musen, välj *Append PLC project* och peka ut korrekt *.tpy fil.

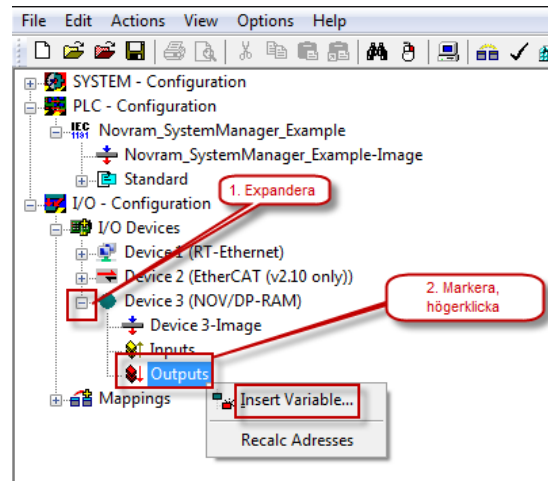


Denna variabelfil (*.tpy) skapas i projektmappen efter en *rebuild all* i PLC Control-projektet.

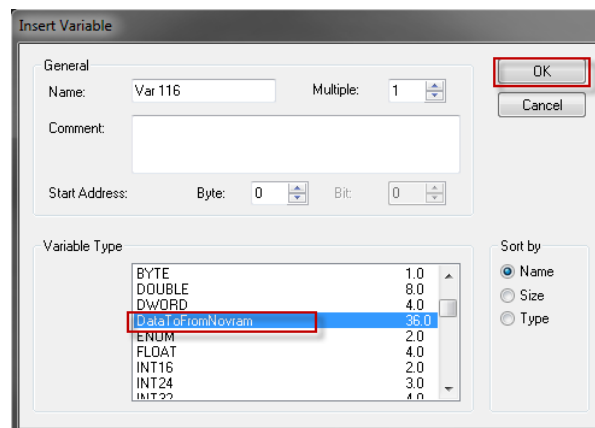


Nu ligger strukten i System Managern och skall knytas mot Novram-minnet.

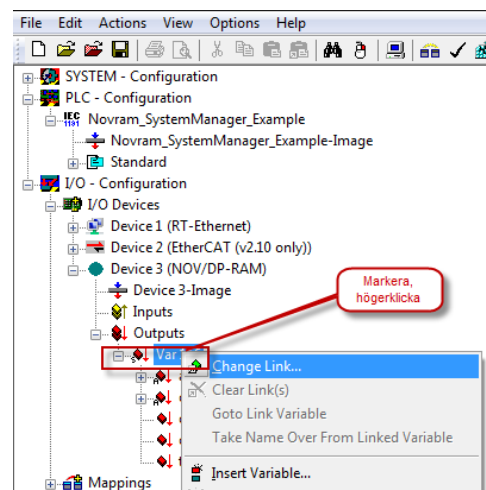
Expandera Novram-minne, markera *Outputs*, högerklicka och välj *Insert Variables*.



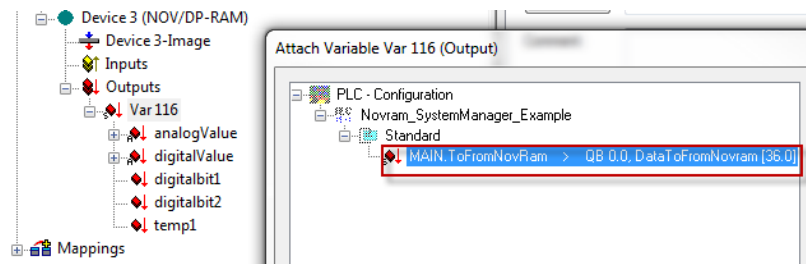
Välj in Strukten som skapades, i exemplet heter den *DataToFromNovram*.



Strukten skall länkas till PLC projektet, detta görs genom att markera den, högerklicka och välj *Change Link*.



Välj motsvarande strukt som skapades i PLC-programmet och tryck OK.



Notera att om fler variabler läggs till strukten i PLC koden efter att länkningen i System Managern är gjord måste en ny länkning göras i System Managern.

Nu är länkningen klar och värdena skrivs cykliskt ner till Novram minnet och efter projektnerladdning eller spänningsbortfall kommer de senaste skrivna värdena i Novram att kopieras tillbaka till PLC projektet med hjälp av strukten *ToFromNovRam*.

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

This publication contains statements about the suitability of our products for certain areas of application. These statements are based on typical features of our products. The examples shown in this publication are for demonstration purposes only. The information provided herein should not be regarded as specific operation characteristics. It is incumbent on the customer to check and decide whether a product is suit-able for use in a particular application. We do not give any warranty that the source code which is made available with this publication is complete or accurate. This publication may be changed at any time with-out prior notice. No liability is assumed for errors and/or omissions. Our products are described in detail in our data sheets and documentations. Product-specific warnings and cautions must be observed. For the latest version of our data sheets and documentations please visit our website (www.beckhoff.com).

© Beckhoff Automation GmbH, September 2009

The reproduction, distribution and utilisation of this document as well as the communication of its contents to others without express authorisation is prohibited. Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.