NMPS

Användarhandledning

Revision: 97 11 04 Claes Sjöfors
Version: V4.0-1

SSAB Oxelösund

Inledning

Detta är en preliminär objektsbeskrivning tillkommen i all hast...
NMps innehåller att antal objekt för att hantera dataobjekt i plceditorn. Med dataobjekt menas här företrädesvis objekt som definierats som en användarklass men kan i princip vara vilken typ av objekt som helst.

Objekt

NMps omfattar ett antal objekt som ligger i klassvolymen NMPS. En del objekt skapar man i konfiguratören, där man hämtar dem under AllClasses-NMPS i paletten. Andra objekt skapar man i plc-editorn, där del ligger under fliken NMPS i plceditorn palett.

NMpsCell

Funktion

Ett NMpsCell objekt är en förvaringscell för data objekt.

Cellen innehåller ett antal positioner för att lagra referenser till dataobjekten. Positionerna är indexerade från ett och uppåt. Varje position innehåller en pekare till dataobjektet, objid för dataobjektet, två Boolean som markerar om bakkant resp framkant befinner sig i cellen, samt slutligen dlid.

Första dataobjektet som läggs in i en cell hamnar alltid på index 1. När flera objekt matas in läggs det senaste objektet på index 1 och de övriga objekten förskjuts ett index.

Dataobjekten kan läggas in på två sätt i cellobjektet

- mha ett trp objekt (NMpsTrp, NMpsTrpRR eller NMpsTrpFF) som lägger in ett dataobjekt först eller sist i cellen.
- mha externfunktionen kan ett dataobjekt läggas in på valfri plats i cellen från ett applikationsprogram (eller från rtt).

Se appendix A hur man vid simulering kan skapa och lägga in ett objekt i cellen mha en rtt kommandofil.

Underfönster

Ett underfönster till en NMpsCell används om samma operationer ska utföras för samtliga dataobjekt i cellen. Koden i underfönstret kommer under ett plc scan, först att exekveras en gång för att kunna initiera variabler i
fönstret, och sedan en gång för varje dataobjekt i cellen. Dataobjekten refereras med CurrentData. Om man vill separera dataobjeken kan man använda objektet CurrentIndex som anger vilket index i cellobjektet som aktuellt dataobjekt ligger på. Under initierings-exekveringen är CurrentData och CurrentIndex lika med noll.

Eftersom fönstret exekveras flera gånger i samma scan får det ej innehålla objekt med tillståndsattribut. Följande
är tillåtna And, Or, ...

Attribut

In

Koppling till ett NMpsTrp objekt.

Out

Koppling till ett NMpsTrp objekt.

MaxSize

Maximalt antal dataobjekt som ska rymmas i cellen (maxvärde 30).

Function

Speciella funktioner hos cellobjektet.
Function är en bitmask och olika funktioner kan kombineras.

Function	Beskrivning
4	Om ett objekt tas bort ur cellen med externfunktionen (ExternOpType = 2, 3 el 4) tas dataobjektet även bort ur rtdb.
8	När cellen återställs genom att resetobjektet, sätts tas dataobjektet även bort ur rtdb.

LastIndex

Antal dataobjekt som cellen för närvarande innehåller.

CellFull

Markerar att cellen är full (dvs aktuellt antal dataobjekt är lika med MaxSize).

FrontNew

FrontNew är true under en scantid om ett dataobjekt transporterats in i cellen genom In, eller med extern operation 1.

RearNew

RearNew är true under en scantid om ett dataobjekt transporterats in i cellen genom Out.

ExternObjId

Objid för ett dataobjekt som ska läggas in eller tas bort från cellen från ett externt program.

ExternOpType

Typ av extern operation:

ExternOpType	Beskrivning
0	Lägger in ExternObjId på index 1 i cellen.
1	Lägger in ExternObjId på index angivits i ExternIndex i cellen.
2	Tar bort dataobjekt på index 1 i cellen.
3	Tar bort dataobjekt på index angett i ExternIndex i cellen.
4	Tar bort dataobjekt med objid som angivits med ExternObjId från cellen.
8	Flytta dataobjekt med objid som angivits med ExternObjid framåt i cellen.
9	Flytta dataobjekt med objid som angivits med ExternObjid bakåt i cellen.
10	Tar bort dataobjekt med objid som angivits med ExternObjId från cellen. Dataobjektet tas inte bort ur databasen även om Function för cellen är 4.

ExternFlag

Markerar att en externfunktion ska utföras. Externflag nollställs av cellobjektet när operationen är utförd.
Externflag ska sättas sist av externt pgm. Om det finns risk för kollisioner mellan externa program bör man kontrollera att Exterflag inte är satt.

ExternIndex

Index i cellen där objekt ska läggas in eller tas bort (vid ExternOpType 1 och 3).

ExternStatus

Resultat av senaste externoperation.

DataLast_Pointer

Innehåller referens till dataobjekt med högsta index, dvs som närmast till att transporteras ut genom Out.

DataLast_ObjId

Objid för dataobjekt med högsta index i cellen.

DataLast_Front

True om dataobjektet med högsta index i cellen har sin framända inne i cellen.

DataLast_Back

True om dataobjektet med högsta index i cellen har sin bakända inne i cellen.

DataX_Pointer

Referens till dataobjekt på index X i cellen.

DataX_ObjId

Objid för dataobjekt på index X.

DataX_Front

True om dataobjektet på index X har sin framända inne i cellen.

DataX_Back

True om dataobjektet på index X har sin bakända inne i cellen.

NMpsOutCell

Funktion

NMpsOutCell objekt fungerar som slutstation i en NMps sekvens. Ett objekt som transporteras in i ett NMpsOutCell objekt tas bort ur sekvensen. NMpsOutCell objektet hanterar inte fram och bak-kant, dvs NMpsTrp objektet som kopplas till In attributet ska ha Function 1 eller 2.

Attribut

In

Koppling till ett NMpsTrp objekt.

Function

Speciella funktioner hos cellobjektet.

Function	Beskrivning
0	NMPS tar bort sin direktlänkning av dataobjektet, men objektet tas ej bort ur databasen.
4	Objektet tas bort ur databasen.

NMpsStoreCell

Funktion

NMpsStoreCell fungerar i stort som en NMpsCell. Skillnaden ligger i att NMpsStoreCell innehåller ett utvals attribut för varje dataobjekt. Vid en transport kommer det senast utvalda objektet att transporteras. Är inte något dataobjekt utvalt flyttas det äldsta objektet, som i NMpsCell.

Ett dataobjekt väljs ut genom att Select attributet sätts på det index där objektet ligger. Detta kan göras från
applikations program eller från en operatörsbild. Man kan även välja ut ett dataobjekt med externfunktionen
genom att ange objid för dataobjektet.

Man kan begränsa antalen utvalda objekt till högst ett, eller till exakt ett mha attributet Function.

Attribut

In

Koppling till ett NMpsTrp objekt.

Out

Koppling till ett NMpsTrp objekt.

MaxSize

Maximalt antal dataobjekt som ska rymmas i cellen (maxvärde 30).

Function

Speciella funktioner hos cellobjektet.
Function är en bitmask och olika funktioner kan kombineras.

Function	Beskrivning
0	Inga begränsningar på antal utvalda objekt.
1	Högst ett objekt är utvalt. Om det finns ett utvalt objekt, och ett nytt väljs ut kommer det tidigare utvalda att återställas.
2	Exakt ett objekt är utvalt (om det finns objekt i cellen). Om det finns ett utvalt objekt, och ett nytt väljs ut kommer det tidigare utvalda att återställas. Om det utvalda objektet transporteras ut ur cellen, eller om utval tas bort på

	annat sätt, kommer det objekt med högsta index att väljas ut.
4	Om ett objekt tas bort ur cellen med externfunktionen (ExternOpType = 2, 3 el 4) tas dataobjektet även bort ur rtdb. Detta funktion bör ej användas i de fall ett dataobjekt kan befinna sig i flera celler samtidigt (Function 0 hos transportobjektet).
8	När cellen återställs genom att resetobjektet sätts, tas dataobjektet som finns i cellen även bort ur databasen.

LastIndex

Antal dataobjekt som cellen för närvarande innehåller.

CellFull

Markerar att cellen är full (dvs aktuellt antal dataobjekt är lika med MaxSize).

FrontNew

FrontNew är true under en scantid om ett dataobjekt transporterats in i cellen genom In, eller med extern operation 1.

RearNew

RearNew är true under en scantid om ett dataobjekt transporterats in i cellen genom Out.

ExternObjId

Objid för ett dataobjekt som ska läggas in eller tas bort från cellen från ett externt program.

ExternOpType

Typ av extern operation

ExternOpType	
0	Lägger in ExternObjId på index 1 i cellen.
1	Lägger in ExternObjId på index angett i ExternIndex i cellen.
2	Tar bort dataobjekt på index 1 i cellen.
3	Tar bort dataobjekt på index angett i ExternIndex i cellen.
4	Tar bort dataobjekt med objid som angetts med ExternObjId från cellen.
5	Väljer ut dataobjekt med objid angett i ExternObjId.
6	Lägger in ExternObjId på index 1 i cellen och väljer ut det.
7	Tar bort utval på objekt angivet i ExternObjId.
8	Flytta dataobjekt med objid som angivits med ExternObjid framåt i cellen.
9	Flytta dataobjekt med objid som angivits med ExternObjid bakåt i cellen.
10	Tar bort dataobjekt med objid som angivits med ExternObjId från cellen. Dataobjektet tas inte bort ur databasen även om Function för cellen är 4.

ExternFlag

Markerar att en externfunktion ska utföras. Externflag nollställs av cellobjektet när operationen är utförd.

Externflag ska sättas sist av externt pgm. Om det finns risk för kollisioner mellan externa program bör man kontrollera att Exterflag inte är satt.

ExternIndex

Index i cellen där objekt ska läggas in eller tas bort (vid ExternOpType 1 och 3).

ExternStatus

Resultat av senaste externoperation.

DataLast_Pointer

Innehåller referens till dataobjekt med högsta index, dvs objektet som ligger närmast till för att transporteras ut genom Out.

DataLast_ObjId

Objid för dataobjekt med högsta index i cellen.

DataLast_Front

True om dataobjektet med högsta index i cellen har sin framända inne i cellen.

DataLast_Back

True om dataobjektet med högsta index i cellen har sin bakända inne i cellen.

DataLast_Select

True om dataobjektet är utvalt.

DataX_Pointer

Referens till dataobjekt på index X i cellen.

DataX_ObjId

Objid för dataobjekt på index X.

DataX_Front

True om dataobjektet på index X har sin framända inne i cellen.

DataX_Back

True om dataobjektet på index X har sin bakända inne i cellen.

DataX_Select

True om dataobjektet är utvalt. Kan sättas och återställas från applikationsprogram och operatörsbilder.

NMpsCellMir

Funktion

NMpsCellMir speglar innehållet i en eller att antal andra celler. De speglade cellerna kan ligga på samma nod eller på en annan nod. Även dataobjekt och valda delar av innehållet i dataobjekt kan speglas. Se beskrivning av objektet NMpsMirrorConfig för konfigurering av NMpsMirror funktionen.

Attribut

CellObjects

De cellobjekt som ska speglas. Cellobjekten kan ligga på samma eller på en annan nod.

NumberOfCellObj

Antal cellobjekt angivna i attributet 'CellObjects'.

DataCollect

Anger att konverterade dataobjekt vars ursprungliga objekt har försvunnit ut ur de speglade cellerna, ska samlas upp i en cell.

CollectCell

Cell som borttagna konverterade dataobjekt samlas upp i.

DataObjConv

Anger att dataobjekt i original cellerna ska speglas.

ConvConfig

Konfigurationsobjekt som specificerar spegling och konvertering av dataceller.

ReleaseTime

Tid i sekunder som ett försvunnet objekt ska hållas kvar av speglingsfunktionen. Vid spegling mellan noder kan ett dataobjekt en kort stund försvinna t ex vid en förflyttning mellan två celler. Genom att det konverterade objektet sparas en viss tid återknyts kontakten med detta när originalobjektet dyker upp igen.

MaxSize

Maximalt antal dataobjekt som ska rymmas i cellen (maxvärde 30).

Function

Ej implementerat.

LastIndex

Antal dataobjekt som cellen för närvarande innehåller.

CellFull

Markerar att cellen är full (dvs aktuellt antal dataobjekt är lika med MaxSize).

DataLast_Pointer

Innehåller referens till dataobjekt med högsta index, dvs som närmast till att transporteras ut genom Out.

DataLast_ObjId

Objid för dataobjekt med högsta index i cellen.

DataX_Pointer

Referens till dataobjekt på index X i cellen.

DataX_ObjId

Objid för dataobjekt på index X.

NMpsTrp

Funktion

Transporterar ett dataobjekt mellan två celler.
Transporten kan ske på olika sätt

- Vid en trigg transporteras hela objektet mellan cellerna.
- Ett objekt flyttas successivt mellan cellerna med framkant först och bakkant sedan (eller vice versa).
- Vid en trigg flyttas samtliga dataobjekt från en cell till nästa cell
- Vid en trigg flyttas samtliga dataobjekt till nästa cell under förutsättning att destinationscellen är tom.

Attribut

In

Kopplas till Out attributet på ett cellobjekt.

Out

Kopplas till In attributet på ett cellobjekt.

Function

Anger vilken typ av transport som transportobjektet ska utföra.

Function	Beskrivning
0	Ett objekts flytta successivt mellan cellerna med framkant först och bakkant sedan (eller vice versa). Vid en positiv flank på TriggForwFront flyttas framkanten på dataobjektet från In till Out. Vid en positiv flank på TriggForwBack flyttas bakkanten från In till Out. Vid en positiv flank på TriggReverseFront flyttas framkanten från Out till In. Vid en positiv flank på TriggReverseBack flyttas bakkanten från Out till

	In.
1	Ett objekt flyttas mellan cellerna. Endast Trigg...Front ingångarna ska användas för att trigga en transport. Vid en positiv flank på TriggFrowFront flyttas objektet från In till Out. Vid en positiv flank på TriggReverseFront flyttas objektet från Out till In.
2	Samtliga objekt i sändcellen flyttas. Vid en positiv flank på TriggForwFront flyttas samtliga dataobjekt i cellen kopplad till In, till cellen kopplad till Out. Vid en positiv flank på TriggReverseFront flyttas samtliga dataobjekt i cellen kopplad till Out, till cellen kopplad till In.
4	Samma funktion som 2 med undantaget att flyttningen endast sker om mottagarcellen är tom.
8	Återställer trigg-ingångar (om dessa ej är kopplade).
16	Trigga på nivå istället för positiv flank.

TriggForwFront

Flyttar ett dataobjekt från In till Out vid positiv flank.
Flyttar samtliga objekt, ett objekt, eller endast framkanten av ett objekt beroende på Function.

TriggReverseFront

Flyttar ett dataobjekt från Out till In vid positiv flank.
Flyttar samtliga objekt, ett objekt, eller endast framkanten av ett objekt beroende på Function.

TriggForwBack

Flyttar bakkanten av ett dataobjekt från In till Out vid positiv flank. Kräver att Function = 0.

TriggReverseBack

Flyttar bakkanten av ett dataobjekt från Out till In vid positiv flank.

Status

Är true om en förflyttning misslyckas beroende på någon av följande faktorer.

- Det finns inget dataobjekt i sändcellen
- Mottagarcellen är full.
- En framkants trigg kommer när en bakkantstrigg borde komma.
- En bakkantstrigg kommer när en framkantstrigg borde komma.

NMpsTrpRR

Funktion

Förflyttar dataobjekt från en utgång på ett cellobjekt till en utgång på ett annat cellobjekt. Funktionen är i övrigt identisk med ett NMpsTrp objekt Function = 0 bör ej användas (dvs Trp objektet ska hantera fram och bakkant) Om den används kommer objektet att vändas av trp objektet eftersom framkanten måste ligga före bakkanten i cell objekten. Dvs framkanten kommer att befinna sig samtidigt i båda cellerna...

Attribut

Se NMpsTrp.

NMpsTrpFF

Funktion

Förflyttar dataobjekt från en ingång på ett cellobjekt till en ingång på ett annat cellobjekt. Funktionen är i övrigt identisk med ett NMpsTrp objekt Function = 0 bör ej användas (dvs Trp objektet ska hantera fram och bakkant) Om den används kommer objektet att vändas av trp objektet eftersom framkanten måste ligga före bakkanten i cell objekten. Dvs bakkanten kommer att befinna sig samtidigt i båda cellerna...

Attribut

Se NMpsTrp.

GetData

Funktion

Hämtar upp referens till ett dataobjekt.
Dynamiska objekt hanteras ej av GetData.

Attribut

Out

Kopplas till ett en dataingång på tex DataArithm, DataCopy, DataReset.

DataObjid

Dataobjekt som ska refereras.

DataArithm

Funktion

DataArithm objektet gör det möjligt att komma åt innehållet i dataobjekt, analoga och digital signaler och parametrar i c-kod.
Objektet innehåller

- fyra dataingångar.
- sex digitala ingångar.
- sex analoga ingångar.
- fyra datautgångar.
- sex digitala utgångar.
- sex analoga utgångar.

Funktionen för digitala och analoga signaler är i densamma som för CArithm, men DataArithm kan dessutom hantera dataobjekts referenser.
Ett dataobjekt kan kopplas till en dataingång (antingen från en GetData eller från ett NMpsCell objekt). I koden hanteras den som en c-struct.

Attribut

DataIn1-DataIn4

Dataingångar som kopplas till en datareferens utgång, dvs GetData, NMpsCell eller DataArithm.
Ingångarna refereras i koden som Da1, Da2, Da3 resp Da4.

AIn1-AIn6

Analoga ingångar. Ingångarna refereras A1 - A6 i c-koden.

DIn1-DIn6

Digitala ingångar. Ingångarna refereras d1 - d6 i c-koden.

OutData1-OutData4

Datautgångar. Utgångarna refereras ODa1 - ODa4 i c-koden, och kan tilldelas värdet av en dataingång.

OutA1-OutA6

Analoga utgångar. Refereras OA1- OA6 i koden.

OutD1-OutD6

Digitala utgångar. Refereras od1- od6 i koden.

Code

C-kod.

Syntaxregler

Digitala och analoga in och utgångar hanteras på samma sätt som i ett CArithm objekt. Man kan ge ett aliasnamn till en in- eller utgång mha 'aliasdef' (för dataingångar invändes 'classdef').

Exempel

```
aliasdef d1 StartaMotor, od1 StoppaMotor;
aliasdef A1 Temperatur;

if ( Temperatur > 200)
  StoppaMotor = 1;
```

Dataingångar

En Dataingång som används i koden måste klass definieras med en classdef sats. Classdef satsen innehåller ordet classdef, dataingång och klassen på det dataobjekt som ska hanteras.
Man kan även ge ett aliasnamn i classdef deklarationen.

Exempel

```
classdef Da1 plate;
classdef Da2 plate FrämstaPlåt;
classdef Da3 vdata Värningsdata, Da4 data SistaPlåt;
```

Dataingångarna hanteras som en pekare på angiven klass.
Antag att klassen data innehåller attributet Width.
Attributet i dataobjektet refereras då med Da1->Width.

Exempel

```
if ( d1)
  Da1->Width = 55.;
else
  Da1->Width = 66.;

 if  ( FrämstaPlåt->Temperatur > SistaPlåt->Temperatur)
  od1 = 1;
```

Om dataingången är kopplad till ett NMpsCell objekt är datapekaren noll om refererad plats i cellen inte innehåller något dataobjekt. I detta fall måste man först testa om pekaren är noll innan den används. Plcprogrammet kommer annars att drabbas av access violation...

Exempel

```
if ( Da1 != 0)
{
  if ( d1)
    Da1->Width = 55.
  else
    Da1->Width = 66.
}
```

Om dataingången är kopplad till ett cellobjekt kan man komma åt informationen om fram och bakkant genom Da1Front, Da1Back, Da2Front, ...

Datautgångar

En datautgång kan tilldelas värdet av en dataingång

Exempel

```
if ( d1)
  ODa1 = Da1;
else
  ODa1 = Da2;
```

DataCopy

Funktion

Kopierar innehållet i ett dataobjekt till ett annat dataobjekt.
För närvarande måste ett av objekten vara refererat med en GetData.

Attribut

DataFrom

Kopplas till en datareferens utgång.

DataTo

Kopplas till en datareferens utgång.

Condition

Om condition är true kommer kopieringen att ske, annars inte.

DataReset

Funktion

Nollställer innehållet i ett dataobjekt, dvs sätter samtliga attribut till binärt noll.
För närvarande måste objektet refereras med en GetData.

Attribut

Data

Kopplas till en GetData.

Condition

Om condition är true kommer nollställningen att ske, annars inte.

CurrentData

Funktion

Objektet används i ett NMpsCell underfönster.
Underfönstret kommer att exekveras för varje dataobjekt som befinner sig i cellen och referenser till dataobjekten i cellen sker med ett CurrentData objekt.

Attribut

dataCurrent_Pointer

Kopplas till en datareferens ingång.

CurrentIndex

Funktion

Objektet används i ett NMpsCell underfönster.
Objektet hämtar upp index för det dataobjekt som exekveras just nu.

Attribut

dataCurrent_Index

Innehåller index för det dataobjekt som exekveras.

DataCollect

Funktion

Samlar i hop ett antal dataobjekt (max 24 st), och placerar pekare till objekten i en array. Detta gör det möjligt att komma åt dataobjekten i en DataArithm.
Normalt kan endast 4 dataobjekt hanteras i en DataArithm, men genom att utnyttja DataCollect kan upp till 96 st hanteras. Dessutom är det ofta bekvämt att ha dataobjekten ordnade i en array för att kunna använda loopar i c-koden.

I MaxIndex ska antalet dataingångar som används i DataCollect objektet anges. Samtliga dataingångar mellan 1 och MaxIndex måste vara kopplade.

Attribut

MaxIndex

Antal kopplade ingångar. Måste datasättas i attributeeditorn (eller i runtiime).

Da1 - Da24

Dataingångar som kopplas till GetData, NMpsCell objekt el dyl.

DataP

Array med 24 element som innehåller pekare till de objekt som är kopplade till DataCollekt objektet.

DataOut

Datautgång som kan kopplas till ett DataArithm objekt (eller annan dataingång).

Exempel

Här följer ett exempel på koden i en DataArithm som har DataCollect objektet kopplat till Da1 ingången. Ett antal objekt av klassen Mtrl är kopplade till DataCollect objektet.

```
classdef Da1 DataCollect
int i;

OA1 = 0;
for ( i = 0; i < Da1->MaxIndex; i++)
  if ( ((pwr_sClass_Mtrl *)(Da1->DataP[i]))->Length > 10.0)
    OA1++;
```

DpCollect

Funktion

Samlar i hop ett antal attribut av typen Boolean (max 24 st), och placerar värdet i en vektor. Detta gör det möjligt att komma åt värdet i en DataArithm. Normalt kan endast 6 digital attribut hanteras i en DataArithm, men genom att utnyttja DpCollect kan upp till 96 st hanteras.

Dessutom är det ofta bekvämt att ha värdena ordnade i en vektor för att kunna använda loopar i c-koden.

I MaxIndex ska antalet ingångar som används i DpCollect objektet anges. Ingångarna behöver ej vara kopplade utan kan datasättas i attributeditorn eller i runtime.

Attribut

MaxIndex

Antal kopplade ingångar. Måste datasättas i attributeditorn (eller i runtime).

Dp1 - Dp24

Ingångar av typen Boolean.

Dp

Array med 24 element som innehåller värdet av kopplade attribut.

DataOut

Datautgång som kan kopplas till ett DataArithm objekt (eller annan dataingång).

Exempel

Här följer ett exempel på koden i en DataArithm som har DpCollect objektet kopplat till Da1 ingången.

```
classdef Da1 DpCollect;
int i;

OA1 = 0;
for ( i = 0; i < Da1->MaxIndex; i++)
  if ( Da1->Dp[i])
    OA1++;
```

DpDistribute

Funktion

Distribuerar en vektor med boolean till ett antal attribut av typen Boolean (max 24 st). Vektorn deklareras i en DataArithm, och adressen till vektorn läggs ut på en datautgång, till vilken dataingången på DpDistribute-objektet kopplas. Elementen i vektorn läggs ut på dp-utgångarna i DpDistribute-objektet. Normalt kan endast 6 digital attribut hanteras i en DataArithm, men genom att utnyttja DpDistribute kan upp till 96 st hanteras.

I MaxIndex ska antalet utgångar som används i DpDistribute objektet anges.

Vektorn ska vara en static deklarerad array av pwr_tBoolean med antal element större eller lika med MaxIndex. I fösta hand ska typen nmps_sDpDistr användas, en array av pwr_tBoolean med 24 element som finns deklarerad i ssab_inc:rs_plc_macro_nmps.h.

Attribut

MaxIndex

Antal använda utgångar. Måste datasättas i attributeditorn (eller i runtiime).

Dp1 - Dp24

Utgångar av typen Boolean.

DataIn

Dataingång som kopplas till en datautgång på ett DataArithm objekt. I koden i dataartithm-objektet måste datautgången tilldelas adressen till en vektor.

Exempel

Här följer ett exempel på koden i en DataArithm som har DpDistribute objektet kopplat till ODa1 utgången.

```
static nmps_sDpDistr vect;
int i;
```

```
ODa1 = vect;
for ( i = 0; i < 24; i++)
{
  if (d1)
    vect[i] = 1;
  else
    vect[i] = 0;
}
```

ApCollect

Funktion

Samlar i hop ett antal analoga attribute (max 24 st), och placerar värdet i en vektor. Detta gör det möjligt att komma åt värdet i en DataArithm. Normalt kan endast 6 analoga attribut hanteras i en DataArithm, men genom att utnyttja DpCollect kan upp till 96 st hanteras.

Dessutom är det ofta bekvämt att ha värdena ordnade i en vektor för att kunna använda loopar i c-koden.

I MaxIndex ska antalet ingångar som används i ApCollect objektet anges. Ingångarna behöver ej vara kopplade utan kan datasättas i attributeditorn eller i runtime.

Attribut

MaxIndex

Antal kopplade ingångar. Måste datasättas i attributeeditorn (eller i runtiime).

Ap1 - Ap24

Analoga ingångar.

Ap

Array med 24 element som innehåller värdet av kopplade attribut.

DataOut

Datautgång som kan kopplas till ett DataArithm objekt (eller annan dataingång).

Exempel

Här följer ett exempel på koden i en DataArithm som har ApCollect objektet kopplat till Da1 ingången.

```
classdef Da1 ApCollect
int i;

OA1 = 0;
for ( i = 0; i < Da1->MaxIndex; i++)
  if ( Da1->Ap[i] > 10.0)
    OA1++;
```

ApDistribute

Funktion

Distribuerar en vektor med float till ett antal attribut av typen Float (max 24 st). Vektorn deklareras i en DataArithm, och adressen till vekorn läggs ut på en datautgång, till vilken dataingången på ApDistribute-objektet kopplas. Elementen i vektorn läggs ut på Ap-utgångarna i ApDistribute-objektet. Normalt kan endast 6 float-attribut hanteras i en DataArithm, men genom att utnyttja ApDistribute kan upp till 96 st hanteras.

I MaxIndex ska antalet utgångar som används i ApDistribute objektet anges.

Vektorn ska vara en static deklarerad array av pwr_tFloat32 med antal element större eller lika med MaxIndex. I fösta hand ska typen nmps_sApDistr användas, en array av pwr_tFloat32 med 24 element som finns deklarerad i ssab_inc:rs_plc_macro_nmps.h.

Attribut

MaxIndex

Antal använda utgångar. Måste datasättas i attributeditorn (eller i runtime).

Ap1 - Ap24

Utgångar av typen float.

DataIn

Dataingång som kopplas till en datautgång på ett DataArithm objekt. I koden i dataartithm-objektet måste datautgången tilldelas adressen till en vektor.

Exempel

Här följer ett exempel på koden i en DataArithm som har ApDistribute objektet kopplat till ODa1 utgången.

```
static nmps_sApDistr vect;
```

```
int i;

ODa1 = vect;
for ( i = 0; i < 24; i++)
    vect[i] = 0.1 * i;
```

DataSelect

Funktion

Väljer en dataingång beroende på index.
Upp till sexton dataobject kan kopplas till SelectData, och en av dessa läggs ut på utgången.
Dataobjekten kan kopplas genom en GetData eller från en NMpsCell el dyl.

Om Index < 1 väljs Da1, om Index > MaxIndex väljs den MaxIndex'te ingången.

I MaxIndex ska antalet dataingångar anges som SelectData ska välja mellan.

Samtliga dataingångar mellan 1 och MaxIndex måste vara kopplade.

Attribut

Index

Styr vilken dataingång som slussas över till utgången. Om Index = 1 väljs den första dataingången etc.

MaxIndex

Antalet dataingångar som DataSelect objektet väljer mellan.

Da1 - Da16

Dataingångar som kopplas till GetData, NMpsCell objekt el dyl.

Out

Datautgång som innehåller referens till utvalt data.

CLoop

Funktion

CLoop objektet innehåller ett underfönster som exekveras ett antal gånger under ett plcpgm-exekveringsvarv. Antalet gånger exekveringen sker bestäms av attributen

StartIndex, StopIndex och Increment. Aktuellt loop index kan hämtas upp i underfönstret med ett CurrentIndex objekt.

Attribut

StartIndex

Index vid vilket loopen startar.

StopIndex

Loopen fortsätter tills loop index är mindre eller lika med StopIndex.

Increment

Värde som adderas till loop index vid varje varv.

CurrentIndex

Innehåller loopindex under loopen. CurrentIndex kan nås från underfönstret med objektet CurrentIndex.

Func

Funktion

Func gör det möjligt att använda samma kod på flera olika ställen i ett system. Om originalkoden ändras distribueras ändringen till respektive instans vid nästa kompilering av denna.

Func objektet pekar på ett PlcPgm objekt under en $LibHier. I detta PlcPgm ligger orginalkoden. Vid kompilering skapas en kopia av originalkoden som ett underfönster till Func objektet. Till ett original PlcPgm kan knytas ett godtyckligt antal Func objekt. In- och utgångar definieras i originalkoden genom FuncInput och FuncOutput.

Originalkoden ska inte kompileras. Om en ändring görs i originalkoden ska samtliga fönster som innehåller ett Func objekt som refererar till denna kod kompileras om.

Func objektet innehåller 8 analoga och 8 digital in- och utgångar. Om detta antal inte räcker till kan man utöka dem med ett FuncExtend objekt (och fler FuncInput och FuncOutput objekt i originalkoden).

När man har skapat ett Func objekt och kopplat detta till ett PlcPgm kan man efter kompilering öppna Func-objektets underfönster. View och Trace mod är tillåten i detta fönster, editeringar ska göras i original-fönstret. Referenser till objekt i underfönstret kan endast göras via objektsnamnet, ej via objekts identiteten. Detta beror på att originalfönstret kopieras vid varje kompilering och objekten byts ut. Referenser från prooper, appilkationsprogram, rtt fungerar, dock ej t ex GetAp, GetDp, StoAp och StoDp i plceditorn. Kompilering av underfönstret sker automatiskt i samband med kompilering av Func objektet. Underfönstret ska inte kompileras separat.

Om man vill ta bort ett Func objekt som har kompilerats och fått ett underfönster måste man göra på ett lite speciellt sätt. Ta eventuellt en backup. Gå in i plceditorn och ta bort alla kopplingar till Func-objektet. Gör save och gå ur plceditom. Ta upp konfiguratören och leta upp Func-objektet där. Gör delete på objektet och det underliggande trädet.

Attribut

Condition

Exekveringen av koden kan styras av Condition. Om Condition är 1 kommer koden att exekveras, annars inte.

A1-A8

Analoga ingångar. Ingångarna kan kopplas eller datasättas från attributeditorn.

d1-d8

Digitala ingångar. Ingångarna kan kopplas eller datasättas från attributeditorn.

OA1-OA8

Analoga utgångar.

od1-od8

Digitala utgångar.

Function

Namn på det PlcPgm objekt under vilket originalkoden ligger. Sista segmentet av namnet visas grafiskt i objektet.

FuncExtend

Funktion

Om antal in- eller utgångar i ett Func objekt ej räcker till kan man utöka dessa med ett eller flera FuncExtend objekt. FuncExtend objektet kopplas till Func objektet genom att Func objektets namn anges i attributet 'FuncObject'. Till varje FuncExtend ska höra ett FuncInput/FuncOutput-par i originalfönstret. Genom att ange samma index i alla tre objekten knyter dem till varandra. Man kan utelämna FuncInput eller FuncOutput objekt om det endast behövs fler utgångar resp ingångar. Det första FuncExtend objektet bör ha index 1, det andra index 2, osv. Index 0 är reserverat för Func objektet.

Attribut

A1-A8

Analoga ingångar. Ingångarna kan kopplas eller datasättas från attributeditorn.

d1-d8

Digitala ingångar. Ingångarna kan kopplas eller datasättas från attributeditorn.

OA1-OA8

Analoga utgångar.

od1-od8

Digitala utgångar.

FuncObject

Namn på det Func objekt som ska utökas.

Index

Index som anger vilket FuncInput/FuncOuput-par FuncExtend-objektet ska kopplas till. Det första FuncExtend objektet ska ha index 1, det andra index 2 osv.

FuncInput

Funktion

Mha FuncInput objektet anger man vilka kopplingar i ett Func fönster som ska hämtas från Func objektets ingångar (eller ett FuncExtend-objekts ingångar).

Attribut

A1-A8

Analoga utgångar från FuncInput objektet som kopplas till analoga ingångar i ett Func eller FuncExtend objekt.

d1-d8

Digitala utgångar från FuncInput objektet som kopplas till digitala ingångar i ett Func eller FuncExtend objekt.

Index

Index som ska anger vilket Func eller FuncExtend objekt ingångarna ska kopplas till. Index 0 innebär att de kommer att kopplas till Func objektet, index > 0 att de kommer att kopplas till ett FuncExtend-objekt med samma index.

FuncOutput

Funktion

Mha FuncOutpu objektet anger man vilka kopplingar i ett Func fönster som ska läggas i Func objektets utgångar (eller ett FuncExtend-objekts utgångar).

Attribut

OA1-OA8

Analoga ingångar från FuncOutput objektet som kopplas till analoga utgångar i ett Func eller FuncExtend objekt..

od1-od8

Digitala ingångar från FuncInput objektet som kopplas till digitala utgångar i ett Func eller FuncExtend objekt.

Index

Index som ska anger vilket Func eller FuncExt end objekt utgångarna ska kopplas till. Index 0 innebär att de kommer att kopplas till Func objektet, index > 0 att de kommer att kopplas till ett FuncExtend-objekt med samma index.

NMpsMirrorConfig

Funktion

Konfigurering av NMpsMirror funktionen, dvs spegling av celler och data objekt.

Attribut

ScanTime

Cykeltid för rs_nmps_mirror.

MirrorCellCount

Visar antal funna NMpsCellMir objekt.

MirrorCellInitCount

Visar antal initierade NMpsCellMir objekt.

OrigCellCount

Visar antal funna cellobjekt som ska speglas.

RemoteOrigCellCount

Visar antal funna cellobjekt som ska speglas från en annan nod.

DataObjectCount

Visar antal dataobjekt som NmpsMirror funktionen för närvarande hanterar.

ConvertDataCount

Visar antal dataobjekt som konverteras.

RemoteDataCount

Visar antal dataobjekt som speglas från en annan nod.

RemoteDataDownCount

Visar antal dataobjekt som speglas från en nod som man fn ej har kontakt med..

PendingRemoveCount

Visar antal dataobjekt som är borttagen ur cellen men hålls kvar pga att ReleaseTime i NMpsCellMir objektet ej ännu har uppnåtts.

RemoveCount

Visar antal dataobjekt som har tagits bort ur speglings funktionen.

CreateDataCount

Visar antal dataobjekt som har skapats i rtdb av speglingsfunktionen.

DeleteDataCount

Visar antal dataobjekt som har tagits bort ur rtdb av speglingsfunktionen.

ReconnectDataCount

Visar antal objekt som speglingsfunktionen har återknutit kontakten med.

LoopCount

Visar antal varv speglings jobbet har gått.

Initialize

Detta attribut kan sättas i runtim och medför ett en ny initiering av speglingsfunktionen kommer att genomföras. Detta kan behövas t ex vid mjuk omstart med ändringar i speglings funktionen.

NMpsConvConfig

Funktion

Konfigurering av spegling och konvertering av dataobjekt..

Attribut

InDataClass

Klass på dataobjekt i de celler från vilken dataobjekt ska speglas.

OutDataClass

Klass på de objekt som skapas av speglings funktionen.

OutDataParent

Objekt under vilket skapade dataobjekt kommer att placeras.

CopyOffset

Start offset i det ursprungliga dataobjektet på den dataarea som ska kopieras till det nya objektet.

PasteOffset

Offset i det nya objektet där den kopierade dataarean ska placeras.

CopySize

Storlek i bytes på den dataarea som ska kopieras.

UpdateTime

Ej implementerat

RemTransSend

Funktion

Objektet markerar ett objekt av typ RemTrans för sändning. I ett underfönster kan läggas kod för ifyllnad av sändbuffer. Objektet exekverar först underfönstret och sätter sedan DataValid i RemTrans objektet. Statusflaggor från RemTrans-objektet läggs ut på utgångar i RemTransSend objektet.

Attribut

RemTrans

Dataingång som kopplas till en GetData som pekar på RemTrans objektet.

Send

Sändning av data sker på positiv flank. Underfönstret exekverar och DataValid sätts i RemTrans objektet.

Occupied

Markerar att RemTrans objektet inte kan hantera fler sändningar för tillfället, beroende på att den senaste transen ännu inte har sänts, eller att om transarna buffras, samtliga buffrar är upptagna.

Buffer

Markerar att det finns buffrade transar.

Error

Senaste sändningen resulterade i felstatus.

RemTransRcv

Funktion

Objektet övervakar ett objekt av typ RemTrans för mottagning av transar. I ett underfönster kan läggas kod för uppackning av mottagningsbuffer. När en trans har mottagits exekveras underfönstret och markerar med utgången Recieved att en trans har mottagits.

Attribut

RemTrans

Dataingång som kopplas till en GetData som pekar på RemTrans objektet.

Received

Data har mottagits. DataValid i RemTrans objektet var satt och underfönstret har exekverats. Received är sann i en cykel.

NMpsBackupConfig

Funktion

Konfigurering av backup av celler och dataobjekt.
Backup av NMps sker på cell objekt med backup-biten satt i Function attributet, samt på de data objekt som ligger i dessa celler. Av dataobjekten sker en cyklisk backup, av cellobjekten en cyklisk och en händelsestyrd backup.

Följande saker krävs för att få backup på NMps

- Ett NMpsBackupConfig objekt ska finnas på noden.
- Programmet ssab_exe:rs_nmps_bck.exe läggs in iett $Appl objekt (VMS) eller ssabb_root:<vax_eln.exe>rs_nmps_bck.exe_eln läggs in i systemet mha EBUILD (ELN).
- Backup biten sätts i Function-attributet i de cellobjekt som ska backas upp.

Backupen lagras i två olika filer med extention .bck1 resp .bck2. Backupjobbet alternerar mellan dessa två filer för att säkerställa att det alltid finns en läsbar fil.

Attribut

FullCycleTime

Tid i sekunder för tiden mellan två fullständiga backuper.

IncrementCycleTime

Scantid i sekunder för den händelsestyrda backupen.

BackupFile

Filnamn för backupfilen. Extention ska ej finnas med i filnamnet (filerna får extention.bck1 och .bck2). Ska backupfilen finnas på en disk på en annan node ska nodnummer eller nodnamn anges.

BackupOn

1 om backupen ska i drift.

NoRead

Denna sätts till 1 för att undvika inläsning av backupfilen vid systemstart (kan sättas i pwrp_alias.dat).

ReloadFailure

Sätt till 1 om inläsningen av backupen vid systemstart skulle misslyckas.

ForceFullBackup

Om denna sätts till 1 kommer en fullständig backup att ske. ForceFullBackup återställs av rs_nmps_bck jobbet.

DataRequest

Funktion

DataRequest sänder en förfrågan med en nyckel och tar emot en data trans.
Förfrågan sker genom att en nyckel läggs in i objektets Key- eller KeyStr-attribut. Man kan välja på heltalsnyckel eller ascii nyckel. Triggning av förfrågan kan väljas att ske vid följande tillfällen

- värdet i nyckelattributet har ändrats
- trigg-ingången är sätt
- OpTrigg-attributet sätts.

Om svaret inte har kommit inom timeout-tiden, meddelas operatören.
Data i svaret kan hanteras på olika sätt beroende på Function-attributet.

- Data kopieras via DataConvert funktionen till ett statiskt objekt (dvs ett objekt skapat i utvecklingsmiljön). Data i objektet kan visas upp i bilder, och validering i plc-och applikations-program kan utföras.
- Ett dynamisk objekt skapas med samma namn som nyckeln, och data kopieras till objektet via DataConvert funktionen. Objektet läggs in i en NMpsCell.
- Data kopieras till ett statiskt objekt, och väntar på acceptans från operatör och/eller applikationen. När data accepterats skapas ett dynamisk objekt som läggs in i en NMpsCell.

Acceptansen i tredje alternativet kan ske på tre olika
sätt

- Operatören accepterar data genom att sätta OpAccept attributet.
- Applikationen accepterar data genom att sätta Accept-ingången.
- Både operatören och applikationen måste ha accepterat data innan acceptansen är komplett.

Om data ej accepteras sker återställning genom att Reset-ingången sätts eller genom att OpReset-attributet sätts.

Förfrågan innehåller nyckel (string40) och ett statusord (int32).

Förfrågan och mottagning sker via RemTrans-objekt.

Attribut

Out

Utgången kan kopplas till en NMpsCell. Kopplingen är enbart grafisk, den verkliga kopplingen sker genom CellObject-attributet. Kopplingen ska göras med en AnalogFeedback-koppling.

Trigg

Triggar förfrågan på positiv flank. Function ska inte ha Change-biten satt för att triggning ska ske med Trigg-ingången.

Accept

Applikationen accepterar data i visnings-objektet och skapar ett dynamiskt objekt som läggs i en cell. Accept och Cell bitarna i Function ska vara satt. Om AcceptBoth biten är satt måste dessutom operatören ha accepterat i OpAccept innan data accepteras.

Reset

Väntan på acceptans återställs på positiv flank.

Error

Utgång som indikerar att ett fel har registrerats. Error-utgången tas ner efter ett scan.

Key

Nyckel för data som efterfrågas, om nyckeln är ett heltal. IntKey-biten i Function-attributet ska vara satt.

KeyStr

Nyckel för data som efterfrågas, om nyckeln är en sträng. IntKey-bitet i Function-attributet ska inte vara satt.

Function

Anger vilken funktion som ska användas. En bitmask.

Bit	Namn	Funktion
1	DisplayObject	Mottaget data kopieras till ett statiskt objekt, angivet i attributet DisplayObjekt.
2	Accept	Innan ett dynamisk objekt skapas väntas på accept från operatör och/eller applikation.
4	CellInsert	Ett dynamisk objekt, namngivet efter nyckeln, skapas och läggs in i en cell.
8	Change	Triggning av förfrågan sker genom att värdet i Key, eller KeyStr

		ändras. Triggning sker ej om värdet är 0 eller en NULL-string. Om biten ej är satt sker triggning med Trigg-ingången eller OpTrigg attributet.
16	AcceptBoth	För att data ska accepteras, måste både Accept-ingången och OpAccept-attributet ha satts.
32	DisplayReset	Vid återställning nollställs visningsobjektet.
64	IntKey	Nyckeln är ett heltal som läggs i attributet Key. Om biten ej är satt är nyckeln en sträng i KeyStr.

TimeoutTime

Timeout tid för en förfrågan. Om svar ej har erhållits inom tiden skickas ett meddelande till operatören. Texten hämtas från AlarmText[0], om denna är ifylld.

AlarmText

Meddelande som skickas tilll operatören när något går snett. Meddelandet sänds som B-larm.

AlarmText[0] Meddelande vid timeout.
AlarmText[1] Cellen som det skapade objektet ska läggas i är full.
AlarmText[2] Acceptans-begäran detekterad utan att objektet väntar på acceptans.
AlarmText[3] Fatalt fel, ytterligare info finns på konsolloggen.
AlarmText[4] Det dynamiska objektet som ska skapas finns redan.
AlarmText[5] Reserv.

ReturnStatus

Om status-ordet i mottagar-transen matchar ett värdet i ReturStatus, sänds ett meddelande till operatören. Texten hämtas från ReturnStatusText (med samma index).

ReturnStatusText

Larmtext som skickas vid olika ReturStatus.

DisplayObject

Visningsobjekt. Data i transen kopieras till objektet vi DataConvert funktionen.
Vid acceptans kopieras det vidare till ett dynamiskt objekt, om så önskas.

SendRemTrans

RemTrans-objekt som används för att sända förfrågan.

RcvRemTrans

RemTrans-objekt som används för att ta emot data.

CellObject

NMpsCell-objekt som det skapade objektet ska läggas in i.

DataClass

Klass för objekt som skapas.

DataParent

Plats i hierarkin under vilken skapade objekt ska läggas.

ToConvdef...

FromConvDef...

Se DataCnv objektet.

OpTrigg

Triggning från operatören. OpTrigg återställs automatiskt efter att triggningen registrerats.

OpAccept

Acceptans från operatören. OpAccept återställs automatisk efter att acceptansen registrerats.

OpReset

Reset från operatören. OpReset återställs automatiskt efter att OpReset har registrerats.

DataRcv

Funktion

DataRcv tar emot en trans.

Data i svaret kan hanteras på olika sätt beroende på Function-attributet.

- Data kopieras via DataConvert funktionen till ett statiskt objekt (dvs ett objekt skapat i utvecklingsmiljön). Data i objektet kan visas upp i bilder, och validering i plc-och applikations-program kan utföras.
- Ett dynamisk objekt skapas med samma namn som nyckeln, och data kopieras till objektet via DataConvert funktionen. Objektet läggs in i en NMpsCell.
- Data kopieras till ett statiskt objekt, och väntar på acceptans från operatör och/eller applikationen. När data accepterats skapas ett dynamisk objekt som läggs in i en NMpsCell.

Acceptansen i tredje alternativet kan ske på tre olika sätt

- Operatören accepterar data genom att sätta OpAccept attributet.
- Applikationen accepterar data genom att sätta Accept-ingången.
- Både operatören och applikationen måste ha accepterat data innan acceptansen är komplett.

Om data ej accepteras sker återställning genom att Reset-ingången sätts, eller genom att OpReset-attributet sätts.

Kvittens kan skickas till sändaren.
Förfrågan och mottagning sker via RemTrans-objekt.

Attribut

Out

Utgången kan kopplas till en NMpsCell. Kopplingen är enbart grafisk, den verkligakopplingen sker genom CellObject-attributet. Kopplingen ska göras med en AnalogFeedback-koppling.

Accept

Applikationen accepterar data i visnings-objektet och skapar ett dynamiskt objekt som läggs i en cell. Accept och Cell bitarna i Function ska vara satt. Om AcceptBoth biten är satt måste dessutom operatören ha accepterat i OpAccept innan acceptansen är komplett.

Reset

Väntan på acceptans återställs på positiv flank.

Error

Utgång som indikerar att ett fel har registrerats.
Error-utgången tas ner efter ett scan.

Key

Nyckel för data som efterfrågas. Key används om nyckeln är ett heltal. IntKey-biten i Function-attributet ska vara satt.

KeyStr

Nyckel för data som efterfrågas. KeyStr används om nyckeln är en sträng. IntKey-biten i Function-attributet ska inte vara satt.

Function

Anger vilken funktion som ska användas. En bitmask.

Bit	Namn	Funktion
1	DisplayObject	Mottaget data kopieras till ett statiskt objekt, angivet i attributet DisplayObjekt.
2	Accept	Innan ett dynamis k objekt skapas väntas på accept från operatör och/eller applikation.
4	CellInsert	Ett dynamisk objekt, namngivet efter nyckeln, skapas och läggs in i en cell.
8	Ack	Kvittens skickas till sändaren. Kvittensen skickas med remtrans-objektet angivet i SendRemTrans.
16	AcceptBoth	För att data ska accepteras, måste både Accept-ingången och OpAccept-attributet ha satts.
32	DisplayReset	Vid återställning nollställs visningsobjektet.
64	Header	Mottagen trans antas innehålla en header med nyckel (string40) och status (int32). Om motsvarande bit sätts i sändande DataSend-objekts Function-attribut kommer en header att läggas i transen. En header erfordras om ett dynamiskt objekt ska skapas.
128	AckOther	Mottagningen av data fungerar samtidigt som en kvittens för en tidigare sändning från ett DataSend-objekt. Den nyckel som DataSend-objektet skickar måste finnas i den mottagna transen.

AlarmText

Meddelande som skickas tilll operatören när något går snett. Meddelandet sänds som B-larm.

AlarmText[0] Reserv.
AlarmText[1] Cellen som det skapade objektet ska läggas i är full.
AlarmText[2] Acceptans-begäran detekterad utan att objektet väntar på acceptans.
AlarmText[3] Fatalt fel, ytterligare info finns på konsolloggen.
AlarmText[4] Det dynamiska objekt som ska skapas finns redan i rtdb.
AlarmText[5] Reserv.

DisplayObject

Visningsobjekt. Data i transen kopieras till objektet vi DataConvert funktionen.
Vid acceptans kopieras det vidare till ett dynamiskt objekt, om så önskas.

SendRemTrans

RemTrans-objekt som används för att sända kvittens.

RcvRemTrans

RemTrans-objekt som används för att ta emot data.

CellObject

NMpsCell-objekt som det skapade objektet ska läggas in i.

DataClass

Klass för objekt som skapas.

DataParent

Plats i hierarkin under vilken skapade objekt ska läggas.

ToConvdef...

FromConvDef...

Se DataCnv objektet.

OpAccept

Acceptans från operatören. OpAccept återställs automatisk
efter det acceptansen registrerats.

OpReset

Reset från operatören. OpReeset återställs automatiskt
efter att OpReset har registrerats.

DataSend

Funktion

DataSend sänder en trans och väntar ev på kvittens.
Om kvittensen inte har kommit inom timeout-tiden, meddelas operatören.

Sändning och mottagning av kvittens sker via RemTrans-objekt.

Transen kan innehålla en header med bl a objektsnamn, om t ex motsvarande objekt ska skapas i mottagaränden.

Attribut

DataIn

Ingång som kopplas till en data utgång. Objektet som ingången pekar på kommer att konverteras och läggas i transen. En header som innehåller bl a objektsnamn kan läggas i transen om t ex motsvarande objekt ska skapas i mottagarnoden.

Trigg

Triggar sändning på positiv flank.

DataSent

Markerar att data har sänts. Om kvittens har begärts sätts DataSent när kvittensen har anlänt. DataSent återställs efter ett scan.

Error

Utgång som indikerar att ett fel har registrerats.
Error-utgången tas ner efter ett scan.

Function

Anger vilken funktion som ska användas. En bitmask.

Bit	Namn	Funktion
8	Ack	En kvittens förväntas komma genom RcvRemTrans objektet inom Timout tiden. DataSent flaggan sätts inte förrän kvittensen har anländ.
64	Header	En header läggs först i transen innehållande objektsnamn (string40) och status (int32).
128	AckOther	En kvittens förväntas komma till ett godtyckligt DataRcv-objekt, dvs kvittensen kan innehålla data. Function för DataRcv-objektet ska också vara AckOther. DataSent flaggan sätts vid sändningen och ej när kvittensen kommer.

TimeoutTime

Timeout tid för en sändning. Om kvittens ej har erhållits inom tiden skickas ett meddelande till operatören (om AlarmText[0] är ifylld). Kvittens biten i Function attributet måste vara satt.

AlarmText

Meddelande som skickas till operatören när något går snett. Meddelandet sänds som B larm.
AlarmText[0] Meddelande vid timeout.
AlarmText[1] DataIn pekar ej på något dataobjekt. Triggning kan ej utföras.
AlarmText[2] Reserv.
AlarmText[3] Fatalt fel, ytterligare info finns på konsolloggen.
AlarmText[4] Reserv.
AlarmText[5] Reserv.

ReturnStatus

Om status-ordet i kvittens-transen matchar ett värdet i ReturStatus sänds ett meddelande till operatören med texten i ReturnStatusText med samma index.

ReturnStatusText

Larmtext som skickas vid olika ReturStatus.

SendRemTrans

RemTrans-objekt som används för att sända data.

RcvRemTrans

RemTrans-objekt som används för att ta emot kvittens

ToConvdef...

FromConvDef...

Se DataCnv objektet.

DataCnv

Funktion

DataCnv konverterar data i ett objekt och lägger resultatet i ett annat objekt.
In- och ut-objekt anges med grafiska kopplingar.

Konverteringen definieras med två konverterings definitioner. Dessa anger hur data i in- resp ut-objektet är strukturerat.

Konverteringen sker genom att attribut i de båda konverteringsdefinitionerna paras ihop om de har samma namn. Om ett attribut har samma typ i båda definitionerna kommer det att resultera i en memcpy, annars sker en konvertering.

Datastrukturen definieras i objekt av klasses ConvDef. I detta objekt specificerar man typ, namn och eventuellt storlek och format för elementen i datastrukturen. En rad i ConvDef objektet innehåller två argument: typ och namn, samt i vissa fall ett tredje: storlek, format el dyl.

Typ	Beskrivning	Kommentar
Char	8-bitars Character	
Boolean	Boolean	
Int8	8-bitars int	
Uint8	8-bitars unsigned int	
Int16	16-bitars int	
UInt16	16-bitars unsigned int	
Int32	32-bitars int	
UInt32	32-bitars unsigned int	
Float32	32-bitars float	Om konvertering till String eller Ascii anges format i tredje argumenet (c-sytax, t ex ”%8.2f”).
Float64	64-bitars float	
String	Null-terminerad sträng	Elementets storlek ska specifieras i tredje argumentet (antal byte).
Ascii	Sträng utfylld med space	Elementets storlek ska specifieras i tredje argumentet (antal byte).
Binary	Binärt eller ospecificerat element	Elementets storlek ska specifieras i tredje argumentet (antal byte).
Unknown	Okänt element	Elementets storlek ska specifieras i tredje argumentet (antal byte).

Implementerade konverteringar

Från	Till
Char	String
Int16	Int32
Int32	Int16 Float32 String
Float32	Int32 String Ascii
String	Int32 Float32
Ascii	Int32 Float23

Attribut

DataFrom

Ingång som kopplas till en data utgång. Data i objektet
som ingången pekar på kommer att konverteras och läggas i objektet som DataTo pekar på.

DataTo

Ingång som kopplas till en data utgång. Konverterad data
läggs i objektet som ingången pekar på.

Condition

När Condition är sann sker konvertering. Om Condition ej
kopplas sker alltid konvertering.

ToConvdefType

Anger vilken typ av konverterings-definition som används.

ConvDefType	Namn	Funktion
0	Object	Konverteringen definieras av ett objekt av klassen ConvDef. Attributet ToConvDef innehåller ConvDef objektet.
1	File	Konverteringen definieras av en fil.
2	Class	Konverteringen definieras av en klass.

ToConvdef

Objekt av klassen ConvDef eller en klass som specificerar datastrukturen i DataFrom
objektet.

ToConvdefFile

Filspecifiktation som på en fil som specificerar datastrukturen i DataFrom objektet.

FromConvdefType

Se ToConvdefType. Motsvarande funktion för DataFrom objektet.

FromConvdef

Se ToConvdef. Motsvarande funktion för DataFrom objektet.

FromConvdefFile

Se ToConvdefFile. Motsvarande funktion för DataFrom objektet.

CellDisp

Funktion

CellDisp används för att visa innehållet i de data-objekt som ligger i NMpsCell-objekt i operatörsbilder. Det ger också möjlighet åt operatören att ta bort data-objekt, flytta data-objekt bakåt eller framåt i en cell, välja ut ett dataobjekt för att visa mer information om objektet.

CellDisp-objektet gör det möjligt att visa en tabell över data-objekten i en cell, med objektsnamn och ett antal utvalda attribut. CellDisp-objektet innehåller ett antal vektorer dit valda attribut i data-objekten i cellerna kopieras. Vektorerna kan sedan visas i en operatörsbild.

Det finns även en utvalsfunktion med vilken ett objekt kan väljas ut. Det utvalda objektet kopieras till ett visningsobjekt, dvs ett permanent objekt av samma klass som de dataobjekt som finns i cellen. Om man i en operatörsbild visar innehållet i visningsobjektet kommer detta alltid att spegla det utvalda objektet.

Om man har flera tabeller i samma bild, men vill ha en gemensam utvalsfunktion, kopplar man samman ett antal CellDisp-objekt mha ett DispLink-objekt. DispLink-objektet ser till att högst ett objekt är utvalt samtidigt. I DispLink-objektet finns dessutom funktioner för att ta bort det utvalda objektet, flytta det utvalda objektet ett steg framåt eller bakåt i cellen.

Utvalet sker antingen genom att varje tabellrad förses med en utvals -tryckknapp, eller med SelectNext/SelectPrevious funktion som finns i DispLink objeket. Genom SelectNext/SelectPrevious kan utvalet ske mha funktionstangenter i de fall detta är att föredra framför mus-klickningar.

En tabell kan visa objekten i upp till 10 celler. I CellDisp-objektet finns det plats för 5 attribut av typen float, 5 integer och 5 boolean. Dessutom finns en vektor som innehåller objid för data-objekten.

Max antal dataobjekt som kan visas i samma tabell är 60.
Samma CellDisp kan hämta in dataobjekt från upp till 10 cell-objekt.
Cell objektet kan vara av klassen NMpsCell eller NMpsStoreCell.
Ett DispLink-objekt kan länka ihop ett obegränsat antal CellDisp-objekt.
Utvalsfunktionen är tv ej kopplad till urvalsfunktionen i NMpsStoreCell-objeket.

Attribut

Cell1-Cell10

Kopplas till GetData-objekt som refererar objekt av klassen NMpsCell eller NMpsStoreCell.

Link

Kopplas till utgången på ett DispLink-objekt. Används för att länka urvalsfunktionen med andra CellDisp-objekt, eller för att utnyttja funktioner för att ta bort objekt ur cellen, flytta objekt framåt eller bakåt i cellen, välja ut nästa eller föregående.

DisplayObject

Kopplas till GetData-objekt som refererar ett visnings-objekt av samma klass som de dataobjekt som hanteras i cellerna. DisplayObject gör det möjligt att visa upp mer detaljerad information om det utvalda objektet genom att innehållet i utvalt objekt kommer att kopieras till visnings-objektet.

DataClass

Klassen på de dataobjekt som hanteras i cellerna.

Function

Function	Beskrivning
0	Normal funktion.
1	Data-objekten läggs i omvänd ordning i CellDisp-objektet.

Number

Om utvals-funktionen ska vara gemensam med andra DispCell-objekt måste det finnas en ordning mellan objekten, som bestämmer i vilken ordning utvalet hoppar mellan olika DispCell-tabeller. DispCell-objekten som är kopplade till ett gemensamt DispLink-objekt numreras från ett och uppåt. Numret för varje DispLink-objekt som tillhör en länk måste vara unikt, och det får inte finnas något hopp i nummerserien.

MaxSize

Storlek på den tabell som visas upp (antal rader). MaxSize kan inte vara större än 60.

SelDirection

SelDirection ändrar utvals-riktningen, dvs man får omvänd funktion för funktionerna SelectNext /SelectPrevious och MoveForward/MoveBackward. Detta kan vara användbart om man har länkat ihop flera CellDisp-objekt, och en del tabeller visar dataobjekten i rättvänd ordning, andra i omvänd ordning. Betydelsen av ‘Next’ och ‘Previous’ är då inte entydig. SelDirection 0 innebär att Next och Forward syftar mot högre index i cellen, SelDirection 1 att Next och Forward syftar mod lägre index.

FloatAttr

Vektor pwr_tString32 med 5 element.
Här anges namnet på de attribut dataobjekten av typen float som ska visas i tabellen. Max 5 attribut kan anges. Attributet i FloatAttr[0] kommer att kopieras till F1, FloatAttr[1] till F2 osv.

BooleanAttr

Vektor pwr_tString32 med 5 element.
Här anges namnet på del attribut i dataobjekten av typen boolean som ska visas i tabellen. Max 5 attribut kan anges. Attributet i BooleanAttr[0] kommer att kopieras till B1, BooleanAttr[1] kommer att kopieras till B2 osv.

IntAttr

Vektor pwr_tString32 med 5 element
Här anges namnet på del attribut i dataobjekten av typen integer som ska visas i tabellen. Max 5 attribut kan anges. Attributet i IntAttr[0] kommer att kopieras till I1, IntAttr[1] kommer att kopieras till I2 osv.

Select

Vektor pwr_tBoolean med 60 element.
Ett objekt på rad i väljs ut genom att Select[i] sätts.
Select[n] kopplas i operatörsbilden till en tryckknapp för rad n i tabellen.
OBS! pwr_SetDig kan ej användas i gms eftersom den ej hanterar vektorer, använd pwr_SetValue istället.

Objid

Vektor pwr_tObjid med 60 element.
Innehåller objid för dataobjekt i cellen.

F1-F5

Vektorer pwr_tFloat32 med 60 element.
Innehåller värde i attributen angivet i FloatAtt för respektive dataobjekt i cellen.

B1-B5

Vektorer pwr_tBoolean32 med 60 element.
Innehåller värde i attributen angivet i BooleanAtt för respektive dataobjekt i cellen.

I1-I5

Vektorer pwr_tInt32 med 60 element.
Innehåller värde i attributen angivet i IntAtt för respektive dataobjekt i cellen.

DispLink

Funktion

Ett DispLink-objekt kopplas till ett eller flera CellDisp för att samordna utvalsfunktionen i CellDisp-objeketen, samt för att hantera bortagning och omflyttning av objeket i cellerna.

Attribut

MoveForward

Flyttar utvalt objekt ett steg framåt (till närmast högre index) i cellen. Förflyttning kan endast ske inom en cell.

MoveBackward

Flyttar utvalt objekt ett steg bakåt (till närmast lägre index) i cellen. Förflyttning kan endast ske inom en cell.

Remove

Ta bort utvalt objekt.
Objektet tas bort ur cellen mha cellens extern funktion. Eventuellt (beroende på Function i cellen) tas även objektet bort ur databasen.

SelectNext

Väljer ut nästa objekt, dvs objektet som ligger på närmast högre index relativt utvalt objekt. Om det inte finns något objekt på högre index i cellen, väljs objektet på index 0 i cellen med närmast högre Number.
Om det inte finns en cell med högre Number väljs objektet på index 0 i cellen med Number 1.

SelectPrevious

Väljer ut föregående objekt, dvs objektet som ligger på närmast lägre index relativt utvalt objekt. Om det inte finns något objekt på lägre index i cellen, väljs objektet med högsta index i cellen med närmast lägre Number.
Om det inte finns en cell med lägre Number väljs objektet med högsta index i cellen med högst Number.

SelectObjid

Innehåller objid för utvalt objekt. Härifrån kan utvalt objekt hämtas från t ex ett applikationsprogram.

Applikationsgränssnitt

Till NMps finns ett antal funktioner för att hantera celler och dataobjekt från ett c-program.

nmpsappl_mirror

nmpsappl_mirror speglar innehållet i en eller flera celler till ett applikationsprogram. Funktionen hanterar direktlänkning av celler och dataobjekt och returnerar ej lista på dataobjekt till applikationen, tillsammans med information om bakkant, framkant, utval, vilka objekt som är nya eller har försvunnit, vilken cell ett objekt tillhör. Applikationen förses även med pekare till dataobjekten.

Man initierar en spegling genom att anropar rutinen nmpsappl_MirrorInit. Som argument skickar man med en lista på de celler som ska speglas. Sedan anropas nmpsappl_Mirror cykliskt för att få information om vilka dataobjekt som ligger i cellerna. Alla dataobjekt som finns i den angivna cellerna bakas ihop i en vektor, och ordningen bestämms av den ordning man angivit cellerna till nmpsappl_MirrorInit.

Flera speglingar (max 32 st) kan hanteras i samma program, och varje spegling omfattar maximalt 32 celler. Samma cell kan tillhöra flera speglingar. Man kan t ex spegla samtliga celler i systemet i en array, samtidigt som man speglar enstaka celler, eller urval av celler i andra arrayer.

nmpsappl_MirrorInit

Anrop

```
pwr_tStatus  nmpsappl_MirrorInit(
                  pwr_tString          *cell_array,
                  unsigned long        options,
                  nmpsappl_t_ctx       *ctx)
```

Funktion

Med nmpsappl_MirrorInit initierar man en spegling genom att ange de celler som ska speglas. Man för till baka en context som används för att skilja olika speglingar åt. Efter initieringen anropas nmpsappl_Mirror cykliskt för att få information om cellinnehållet.

Argument

cellarray

Typ

Pekare på en vektor pwr_tString80[]

Funktion

En vektor som innehåller namnet på de celler som ska speglas.
De speglade dataobjekten ordnas i den ordning cellerna anges i vektorn.
En NULL-string markerar sista cellnamnet.

Options

Typ

unsigned int

Funktion

Options är en bitmask. Bitarna anger olika funktioner för nmpsappl_mirror.

Option	Beskrivning
nmpsappl_mOption_Remove	Dataobjekt som har försvunnit från de angivna cellerna sedan senaste anropet läggs med i dataobjektlistan med remove-flaggan satt.
nmpsappl_mOption_NamePath	Namnfältet i den datastruktur som returneras för ett dataobjekt på dataobjektet innehåller hierarkinamnet.

ctx

Typ

nmpsappl_t_ctx

Funktion

En kontext-pekare som skickas med till nmpsappl_mirror rutinen.

nmpsappl_Mirror

Anrop

```
pwr_tStatus nmpsappl_Mirror(
                  nmpsappl_t_ctx       applctx,
                  int                  *data_count,
                  nmpsappl_t_datainfo  **datainfo)
```

Funktion

nmpsappl_Mirror anropas cykliskt för att uppdatera speglingen. Rutinen fyller i en lista på samtliga dataobjekt som finns i cellerna. Listan är av typen nmpsappl_t_datainfo som beskrivs av följande struct

```
typedef struct {
        pwr_tObjid                      objid;
        pwr_tString80                   name;
        pwr_tAddress                    object_ptr;
        pwr_tBoolean                    select;
        pwr_tBoolean                    front;
        pwr_tBoolean                    back;
        pwr_tBoolean                    new;
        pwr_tBoolean                    removed;
        unsigned long                   cell_mask;
        } nmpsappl_t_datainfo;
```

Element	Beskrivning
objid	dataobjektets objid.
name	dataobjektet namn (sista namnledet).
object_ptr	pekare till dataobjektet.
select	select-attributet för dataobjektet i cell-objektet. Om ett dataobjekt ligger i flera celler, sätts select om objektet är utvalt i minst en av cellerna.
front	front-attributet för dataobjektet i cell-objektet. Om ett dataobjekt ligger i flera celler, sätts front om front-flaggen är satt i en av cellerna.
back	back-attributet för dataobjektet i cell-objektet. Om ett dataobjekt ligger i flera celler, sätts back om back-flaggen är satt i en av cellerna.
new	markerar att ett dataobjekt är nytt sedan senaste speglingen.
Removed	markerar att dataobjektet har försvunnit sedan senaste speglingen. Kräver att nmpsappl_mOption_Remove har angetts i options
cell_mask	anger vilken eller vilka celler dataobjektet befinner sig i. cell_mask är en bitmask där första biten anger första cellen (dvs den som angivits först i listan till nmpsappl_mirror_init), osv.

.

Argument

ctx

Typ

nmpsappl_t_applctx

Funktion

En kontext-pekare som erhölls vid anrop till nmpsappl_mirror_init.

data_count

Typ

pekare på en int

Funktion

Antal dataobjekt i dataobjekts-listan

datainfo

Typ

pekare till en vektor av typen nmpsappl_t_datainfo.

Funktion

Lista på dataobjekten i cellerna.

Exempel

```
#include "pwr_inc:pwr.h"
#include "ssab_inc:rs_nmps_appl.h"
#include "pwrp_inc:pwr_cvolvkvendclasses.h"

main()
{
        nmpsappl_t_ctx       ctx;
        int                  i;
        int                  plate_count;
        nmpsappl_t_datainfo  *plate_info;
        pwr_tStatus          sts;
        pwr_sClass_Plate     *plate_ptr;
        pwr_tString80        cellnames[] = {
                        "VKV-END-PF-Plåtföljning-W-Svb15",
```

```
                        "VKV-END-PF-Plåtföljning-W-R30A",
                        ""};

        /* Init pams and gdh */
        sts = gdh_Init( 0, NULL);
        if (EVEN(sts)) LogAndExit( sts);

        /* Init the mirroring*/
        sts = nmpsappl_MirrorInit( cellnames,
                nmpsappl_mOption_Remove, &ctx);
        if (EVEN(sts)) exit(sts);

        for (;;)
        {
          sts = nmpsappl_Mirror( ctx, &plate_count,
                    &plate_info);

          for ( i = 0; i < plate_count; i++)
          {
            plate_ptr = plate_info[i].object_ptr;
            printf( "%20s %1d %1d %1d %1d %1d %4d %5.1f\n",
                        plate_info[i].name,
                        plate_info[i].front,
                        plate_info[i].back,
                        plate_info[i].select,
                        plate_info[i].new,
                        plate_info[i].removed,
                        plate_info[i].cell_mask,
                        plate_ptr->Tjocklek);
          }
          sleep( 1);
        }
}
```

nmpsappl_RemoveData

Anrop

```
pwr_tStatus nmpsappl_RemoveData(
                  nmpsappl_t_ctx        ctx,
                  pwr_tObjid            objid)
```

Funktion

nmpsappl_DataRemove tar bort ett dataobjekt ur de celler som speglas.

Argument

ctx

Typ

nmpsappl_t_applctx

Funktion

Kontext-pekare som erhölls vid anrop till nmpsappl_mirror_init.

Objid

Typ

pwr_tObjid

Funktion

Objid för dataobjekt som ska tas bort.

nmpsappl_RemoveAndDeleteData

Anrop

```
pwr_tStatus nmpsappl_RemoveAndDeleteData(
                  nmpsappl_t_ctx       ctx,
                  pwr_tObjid           objid)
```

Funktion

nmpsappl_DataRemove tar bort ett dataobjekt ur de celler som speglas. Dessutom tas objektet bort ur databasen.

Argument

ctx

Typ

nmpsappl_t_applctx

Funktion

Kontext-pekare som erhölls vid anrop till nmpsappl_mirror_init.

Objid

Typ

pwr_tObjid

Funktion

Objid för dataobjekt som ska tas bort.

Appendix

Appendix A

Exempel på rtt kommandofil för att skapa ett dataobjekt och lägga in det in en cell.

```
!
! Rtt kommandofil för att skapa ett plåtobjekt
! och lägga in det in FrånTRP-cellen.
!
! Skapa ett löpnummer för plåtobjektet
!
if defined lopnum
  define lopnum ++
else
  define lopnum '#SIM-END-PF-LöpnrRäknare.ActualValue'
  define lopnum ++
endif
set par/name=SIM-END-PF-LöpnrRäknare.ActualValue/value='lopnum'!
!
define pname ”VKV-END-PF-Plåtar-”'lopnum'
define cname ”VKV-END-PF-PlåtFöljning-W-FrånTRP”
!
! Skapa och initiera plåt-objektet
!
create object/class=VKVEND_Plat/name='pname'
set par/name='pname'.Lopnr/val='lopnum'
set par/name='pname'.Langd/val=12
set par/name='pname'.Bredd/val=2.2
set par/name='pname'.Tjocklek/val=5
define offsetr30 '#VKV-END-R31-OffsetR30.ActualValue' + 2
set par/name='pname'.Koord_R30/val='offsetr30'
set par/name='pname'.Koord_xf_Svb/val=13
set par/name='pname'.Koord_xB_Svb/val=1
set par/name='pname'.Koord_f_R31/val=13
set par/name='pname'.Koord_b_R31/val=1
set par/name='pname'.HogNr/val=289
!
! Lägg in plåt-objektet i cellen
!
set par/name='cname'.ExternObjId/val='cname'
set par/name='cname'.ExternOpType/val=0
set par/name='cname'.ExternFlag/val=1
```