

I/O-HANTERING MED PROFIBUS I PROVIEW

Revision 2.0	2001-10-17
Revision 2.1	2003-06-06

1 Inledning

Proview 3.0 under operativsystemet LynxOS stödjer Profibus DP. Profibus-kommunikationen hanteras av ett masterkort på PCI-bussen. Proviews I/O-hantering jobbar sedan via en driver mot hårdvaran. Detta dokument beskriver hur Profibus DP-slavar konfigureras och används i Proview, vilka objekt som skall användas och hur attribut i dessa skall ställas in.

Den nya I/O-hanteringen i 3.0 ger bättre förutsättningar att införa nya bussar och korttyper än tidigare och profibushanteringen kommer att helt följa den nya I/O-modellen. Hur denna är strukturerad beskrivs separat.

2 Konfigurering i anläggningshierarkin

Alla data till/från profibusslavar betraktas som I/O. Detta betyder att vanliga Di-, Do-, Ai-, och Ao-objekt används i anläggningshierarkin.

För slavar med I/O-moduler är detta helt naturligt. Kanaler och signaler konfigureras med samma objekt och enligt samma principer som traditionellt I/O.

Det kan möjligtvis tyckas lite konstigt för slavar där datat som utbyts är av parametertyp. Ett exempel på detta är frekvensomriktare, där dataarean kan vara bitorienterade statusord eller parametrar (heltal eller flyttal). Tanken är då att dessa konverteras av I/O-hanteringen till digitala eller analoga signaler enligt en förutbestämd mappning.

Fördelen med detta är att befintliga objekt kan användas fullt ut för alla signaler och parametrar från profibusslavar. Inga specialobjekt behöver tillverkas för användning i anläggninghierarkin.

3 Konfigurering i nodhierarkin

Profibusslingan konfigureras i fyra nivåer:

Profibus master

Konfigurering av masterkortet och bussparametrar, ex: hastighet, max antal slavar. En objekttyp per typ av masterkort. Än så länge används bara Softings Profiboard och objektet heter Pb_Profiboard.

Masterobjekt namnas **Px**, där x är numret på profibusslingan, ex **P1**.

DP-Slav

Ett konfigureringsobjekt för varje slav i slingan. I Proview v3.0 finns en objekttyp per möjlig slav med förinställda attribut. Från v3.3 ska alla DP-slavar konfigureras med samma objekt, Pb_DP_Slave och laddas med parametrar från GSD-fil.

I slavobjektet konfigureras data som är slavspecifik, ex. slavadress, konfigureringsdata och felhanteringsdata. I v3.0 finns slavobjekt för Siemens ET200M, ABB's frekvensomriktare ACS600 och Eurotherms utrustning för brännarstyrning. Dessa objekt heter Pb_ET200M, Pb_NPBA12 och Pb_Euro2500. Dessa slavobjekten ser mycket lika ut och skiljer sig åt endast i de förinställda parametrarna.

Slavobjekt namnas **Sx**, där x är slavnumret inom slingan, ex **P1-S2**.

Observera att första slaven ska ha adress 2 eftersom adress 1 är reserverad för felsökningsutrustning.

Modul

Om slaven kan hantera flera moduler, som ofta är fallet med I/O-slavar, tex Siemens ET200M, konfigureras varje modul i ett objekt. Även övriga typer av slavar konfigureras med moduler för gruppering av de olika parametertyperna. Moduler finns för konfigurering av Di (Pb_Di), Do (Pb_Do), Ai (Pb_Ai) och Ao (Pb_Ao). Dessa är "generiska" vad gäller antalet kanaler, som sätts i attributet NumberOfChannels.

Modulobjekt i modulatorierade slavar namnas **SxMy**, där x är slavnumret och y är modulnumret inom slaven. Moduler numreras löpande inom slaven oavsett typ, ex **P1-S2-S2M4**. (För Siemens I/O ET200M startar modulnumreringen på 4 i och med att själva slavmodulen tar 3 slots i anspråk.) Syftet med att ha slavdata även i modulnamnet är att få ett unikt kanalnamn redan efter två hierarkiled bakifrån vilket är en fördel programmeringsmässigt.

För slavar med parameterdata delas data in i fiktiva moduler som namnas efter datats betydelse eller enligt beskrivning av aktuell slav. Exempelvis **CW** - Control Word, modul med "digitala ingångar", **PDO** - Process Data Output,

modul med ”analoga utgångar”.

Observera vikten av att konfigurera modulerna i samma ordning som de sitter i verkligheten. Detta är nämligen enda sättet för I/O-hanteringen att få rätt ordning i slavens dataarea.

Kanal

Ett objekt för varje signal/parameter i modulen. Kanaler är av de traditionella typerna ChanDi, ChanDo, ChanAi (ChanAit) och ChanAo och kopplas på normalt sätt till Di-, Do-, Ai- eller Ao-objekt på anläggningssidan.

Kanaler numreras enligt vedertaget sätt för respektive modultyp. Exempelvis numreras kanaler på en ET200-modul oktalt och denna numrering finns i klartext på modulen ex. **P1-S2-S2M4-01**.

Numrering eller namnsättning av parameterdata görs på lämpligaste sätt utifrån beskrivningen av respektive slav.

Observera attributet Number i kanalobjekten som fungerar på samma sätt som för QBUS-I/O, dvs första signalen ska ha Number=0, andra signalen Number=1 osv.

Exempel

Exempel på nodhierarki med en Siemens ET200M-slav och en ABB-frekvensomriktare:

```
Noder $NodeHier
  HQLBL1 $Node
    P1 Pb_Profiboard
      S2 Pb_ET200M (Pb_DP_Slave från v3.3)
        S2M1 Pb_DI
          00 ChanDi
          01 ChanDi
          02 ChanDi
          03 ChanDi
          04 ChanDi
          05 ChanDi
          06 ChanDi
          07 ChanDi
          10 ChanDi
          ...
          17 ChanDi
          20 ChanDi
          ...
          27 ChanDi
          30 ChanDi
          ...
          37 ChanDi
        S2M2 Pb_Do
          00 ChanDo
          ...
```

```
37  ChanDo

S3  Pb_NPBA12 (Pb_DP_Slave från v3.3)
SW  Pb_DI
    00  ChanDi
    ...
    15  ChanDi
CW  Pb_DO
    00  ChanDo
    ...
    15  ChanDo
PDI  Pb_AI
    Act  ChanAi
    ...
    ...  ChanAi
PDO  Pb_AO
    Ref  ChanAo
    ...
    ...  ChanAo
```

Här har vi konfigurerat två I/O-moduler i en ET200M, en Di-modul med 32 kanaler och en Do-modul med 32 kanaler. Dessutom finns en frekvensomriktare från ABB (med profibusinterface NPBA-12).

Det finns scripts för enklare konfigurering av objekt i nodhierarkien liknande de som finns för QBUS-I/O.

4 Objektklasser för Profibus

Ett antal nya objektklasser behövs för att kunna konfigurera en profibusslinga i nodhierarkin. Objekten blir unika för varje typ av master, slav respektive modul.

4.1 Pb_Profiboard

Objekt som konfigurerar masterkortet av typen Profiboard från Softing. Definierar parametrar som rör hela bussen.

Attribut

Process

Process som hanterar slingan (1=PLC).
Default = 1.

ThreadObject

PLC-tråd som hanterar slingan (Om Process=1)
Inget defaultvärde.

Status

Indikerar status på profibusslingan. Sköts av I/O-hanteringen.
0 = Ej initierad, 1 = Mode stopped, 2 = Mode clear, 3 = Mode operating

DisableBus

Tänkt att stoppa bussen på positiv flank. Är ännu ej implementerat.

NumberSlaves

Antal konfigurerade DP-slavar i slingan. Beräknas av I/O-hanteringens vid initiering.

BusNumber

Ordningsnummer på masterkortet. Flera masterkort kan användas inom samma system.

Default = 1

MaxNumberSlaves

Max antal slavar som får finnas i Pb-slingan. 0-127.

Default = 48.

MaxSlaveOutputLen

Max storlek på slavs utdata.

Default = 128

MaxSlaveInputLen

Max storlek på slavs indata.

Default = 128

Baudrate

Busshastighet i KBaud. Möjliga värden är 500, 1500, 3000, 6000, 12000.

Default = 1500

Övriga attribut gäller inställningar av bussen. Dessa parametrar skall ändras om man ändrar BaudRate. Rekommenderade värden på dessa finns i nedanstående tabell.

Par/Baudrate	500 kBaud	1,5 MBaud	3 MBaud	6 MBaud	12 MBaud
Tsl	200	300	400	600	1000
MinTsdr	11	11	11	11	11
MaxTsdr	100	150	250	450	800
Tset	1	1	4	8	16
Tqui	0	0	3	6	9
G	1	10	10	10	10
Hsa	126	126	126	126	126
MaxRetryLimit	1	1	2	3	4

Parametrar som inte nämns i tabellen skall **inte** ändras från sina defaultvärden.

4.2 Objektklasser för slavar

Proview v3.0

Varje slav har sin egen objektklass men attributen är desamma i alla klasserna. Det som skiljer är defaultvärden på en del attribut. I dagsläget finns objektklasser för följande slavar:

Siemens ET200M distribuerat I/O.	Pb_ET200M
Pb interface NPBA-12 till ABB's frekvensomriktare ACS600.	Pb_NPBA12
Pb interface till Eurotherms reglerutrustning Euro2500.	Pb_Euro2500

Attribut i slavobjekten i Proview 3.0

Process

Process som hanterar slaven . (1=PLC)
Default = 1

ThreadObject

PLC-tråd som hanterar slaven (om Process=1)
Inget defaultvärde.

SlaveAddress

Slavens adress i Pb-slingan. Observera att slavadress 1 är reserverad för felsökningsutrustning och att första användbara slavadress därför är 2.
Inget defaultvärde.

Status

Status för slavenheten. Sköts av I/O-hanteringen. Kan användas för att generera kommunikationslarm.

0 - Slaven är ej initierad. Detta betyder för det mesta att slaven är felaktigt parametrerad.

1 - Slaven är initierad, men kontakt är ej etablerad.

2 - Slaven är initierad och kontakt är etablerad.

DisableSlave

Stoppar I/O-kopieringen mellan slavens dataarea hos mastern och Proview.

StallAction

Åtgärd vid tappad kontakt och/eller då maximalt antal fel uppnåts.

0 - Ingen åtgärd.

1 - Samtliga ingångar sätts till 0.

2 - Enligt 1 plus åtgärd enligt Emergency break i nodobjektet.

ErrorCount

Felräknare. Sköts av I/O-hanteringen.

ErrorHardLimit

Gräns för ErrorCount då åtgärd enligt StallAction vidtages.

WdFact1, WdFact2

Watchdogfaktorer för slaven. $10 * WdFact1 * WdFact2$ ger den timeout i ms som slaven använder för att detektera avbrott i kommunikationen. Exempelvis sätts WdFact1 till 100 och WdFact2 till det antal sekunder som motsvarar önskad tid.

GroupIdent

Group ident för slaven.

PNOIdent

Slavens PNO identitet. Är förinställt för respektive objektklass och kan inte ändras.

PrmUserDataLen

Längd på prm user data. Är förinställt för respektive objektklass och kan inte ändras.

PrmUserData[]

Slavens parameterdata. Är förinställt för respektive objektklass och kan inte ändras.

AutoConfigure

Varje slav kan konfigureras på två sätt. Antingen frågar man slaven över bussen vid uppstart hur hans konfigurering ser ut (AutoConfigure = 1) eller så laddar man en inställd konfigurering från Proview (AutoConfigure = 0). Vid autokonfigurering kontrollerar I/O-hanteringen att slavens konfigurering stämmer överens med vad som är konfigurerat i Proview. Laddar man däremot konfigureringen ”manuellt” görs en automatisk kontroll av mastern att den stämmer med slavens konfigurering.

Autokonfigurering gör att man slipper skriva in konfigureringsdata i slavobjektet men tar en viss tid i anspråk vid systemstart.

ConfigDataLen

Längd i byte på ConfigData. Används endast om AutoConfigure = 0.

ConfigData[]

Konfigureringsdata som ska laddas till masterkortet vid initiering. Används endast om AutoConfigure = 0.

BytesOfInput

Antal byte indata. Beräknas av I/O-hanteringens vid initiering av slaven.

BytesOfOutput

Antal byte utdata. Beräknas av I/O-hanteringens vid initiering av slaven.

OffsetInputs

Offset för indata i dataarean på masterkortet. Beräknas av I/O-hanteringens vid initiering av slaven.

OffsetOutputs

Offset för utdata i dataarean på masterkortet. Beräknas av I/O-hanteringens vid initiering av slaven.

NumberModules

Antal moduler under slaven. Beräknas av I/O-hanteringens vid initiering av slaven.

Inputs

Dataarea för indata. Här lägger I/O-hanteringens hela slavens indataarea vid varje läsning från masterkortet. Härifrån fördelas sedan datat ut på moduler och signaler. Denna area kan alltså användas för att se det data som slaven skickar till mastern ograverat.

Outputs

Dataarea för utdata. Här lägger I/O-hanteringens hela slavens utdataarea vid varje skrivning till masterkortet. Datat plockas ihop från modulerna innan hela arean skrivs i en operation till masterkortet.

Diag

Dataarea för diagnostik.

Proview v3.3 och senare

Från version 3.3 finns endast ett slavobjekt som ska användas för alla DP-slavar, **Pb_DP_Slave**. Objektet kan konfigureras direkt från en GSD-fil med ett proview-script. En GSD-fil är en standardiserad textfil som beskriver hur slaven ska parametreras. En sådan ska finnas till varje DP-slav och tillhandahålls av tillverkaren.

Skriv in sökvägen till slavens GSD-fil i attributet GSDfile, ex.
pwrp_root:[common.db]siem801d.gsd. Ladda sedan parametrar från GSD-filen genom att gå in i **Utilities**, ta fram Command Window och i editeringsrutan skriva följande:

i version 3.3

@ssab_exe:pb_slave_config *objektnamn*

i version 3.4 och framåt

@"\$pwr_exe/pb_slave_config" *objektnamn*

där *objektnamn* är hela slavobjektets namn, ex. Noder-RHYM4F-P1-S2

De attribut som därefter behöver konfigureras för hand i slavobjektet är ThreadObject och SlaveAddress.

GSD-filen kan behöva modifieras innan laddning för att passa slavens konfigureringsverktyget i det styrsystem man arbetar med. I framtiden kommer sådant verktyg att finnas även i Proview.

Attribut i Pb_DP_Slave i Proview 3.3 och framåt*Description*

Beskrivning (endast för information).

GSDfile

GSD-fil för parametrering. Se ovan.

Status

Status för slavenheten. Sköts av I/O-hanteringen. Kan användas för att generera kommunikationslarm.

0 - Slaven är ej initierad. Detta betyder för det mesta att slaven är felaktigt parametrerad.

1 - Slaven är initierad, men kontakt är ej etablerad.

2 - Slaven är initierad och kontakt är etablerad.

Process

Process som hanterar slaven . (1=PLC)

Default = 1

ThreadObject

PLC-tråd som hanterar slaven (om Process=1)

Inget defaultvärde.

2003-06-06

SlaveAddress

Slavens adress i Pb-slingan. Observera att slavadress 1 är reserverad för felsökningsutrustning och att första användbara slavadress därför är 2. Inget defaultvärde.

ByteOrdering

Bitmap som beskriver byteordning och talrepresentation hos slaven.

Bit nr	Betydelse vid 0:a	Betydelse vid 1:a
0	Little Endian	Big Endian
1	Digitala moduler byteorienterade (ex. DI-signalmodul i ET200M)	Digitala moduler wordorienterade (ex. statusord i NPBA12)
2	Dataord (16 bitar) är signed.	Dataord (16 bitar) är unsigned.

Little endian används i VAX-, Alpha- och Intelbaserade system. Motorolabaserade system och många PLC-system, bl.a. Simatic S5 och S7 använder Big endian. Konvertering sköts automatiskt av I/O-hantering.

StallAction

Åtgärd vid tappad kontakt och/eller då maximalt antal fel uppnåtts.

0 - Ingen åtgärd.

1 - Samtliga ingångar sätts till 0.

2 - Enligt 1 plus åtgärd enligt Emergency break i nodobjektet.

DisableSlave

Stoppas I/O-kopieringen mellan slavens dataarea hos mastern och Proview.

ErrorCount

Felräknare. Sköts av I/O-hantering.

ErrorSoftLimit

Används ej.

ErrorHardLimit

Gräns för ErrorCount då åtgärd enligt StallAction vidtages.

WdFact1, WdFact2

Watchdogfaktorer för slaven. $10 * WdFact1 * WdFact2$ ger den timeout i ms som slaven använder för att detektera avbrott i kommunikationen. Exempelvis sätts WdFact1 till 100 och WdFact2 till det antal sekunder som motsvarar önskad tid.

AutoConfigure

Ska inte användas i v3.3.

2003-06-06

*VendorName**Model Name**Revision**HardwareRelease**SoftwareRelease**PNOIdent**GroupIdent**PrmUserDataLen**PrmUserData[]**ConfigDataLen**ConfigData[]**SlaveUserDataLen**SlaveUserData[]*

Samtliga dessa attribut laddas från GSD-fil enligt beskrivningen ovan.

BytesOfInput

Antal byte indata. Beräknas av I/O-hanteringens vid initiering av slaven.

BytesOfOutput

Antal byte utdata. Beräknas av I/O-hanteringens vid initiering av slaven.

OffsetInputs

Offset för indata i dataarean på masterkortet. Beräknas av I/O-hanteringens vid initiering av slaven.

OffsetOutputs

Offset för utdata i dataarean på masterkortet. Beräknas av I/O-hanteringens vid initiering av slaven.

NumberModules

Antal moduler under slaven. Beräknas av I/O-hanteringens vid initiering av slaven.

Inputs

Dataarea för indata. Här lägger I/O-hanteringens hela slavens indataarea vid varje läsning från masterkortet. Härifrån fördelas sedan datat ut på moduler och signaler. Denna area kan alltså användas för att se det data som slaven skickar till mastern ograverat.

Outputs

Dataarea för utdata. Här lägger I/O-hanteringens hela slavens utdataarea vid varje skrivning till masterkortet. Datat plockas ihop från modulerna innan hela arean skrivs i en operation till masterkortet.

Diag

2003-06-06

Dataarea för diagnostik.

4.3 Pb_Di

Objekt som konfigurerar en Pb-modul med Digigala ingångar eller en uppsättning ”digitala” inparametrar.

Attribut

Process

Process som hanterar modulen (1=PLC).

ThreadObject

PLC-tråd som hanterar modulen (om Process=1).

NumberOfChannels

Antal kanaler på modulen alternativt antal digitala inparametrar i slaven. Kan ligga i intervallet 1-32. Observera att alla kanaler inte behöver vara konfigurerade eller kopplade. Default 32.

Status

Modulens status. Sköts av I/O-hanteringen.

0 = ej initierad

1 = initierad

ConvMask1

16 bitars heltal som utgör en bitmask som beskriver vilka kanaler som ska konverteras. Bit 0 motsvarar kanal 0 osv, och en etta i respektive position betyder att motsvarande kanal skall konverteras. ConvMask1 gäller för kanalerna 0-15. Ska normalt vara 65535 för konvertering av alla kanaler. (Konvertering i I/O-hanteringen för Profibus innebär kopiering från slavens dataarea till signalens ActualValue).

ConvMask2

Som ConvMask1 fast för kanal 16-31.

InvMask1

Mask som beskriver invertering av kanaler enligt samma princip som ConvMask1. Ska normalt vara 0.

InvMask2

Som InvMask1 fast för kanal 16-31.

BytesOfInput

2003-06-06

Antal byte indata för modulen. Beräknas av I/O-hanteringens vid initiering av modulen.

BytesOfOutput

Antal byte med utdata för modulen. Beräknas av I/O-hanteringens vid initiering av modulen.

OffsetInputs

Offset i slavens dataarea för modulens indata. Beräknas av I/O-hanteringens vid initiering av modulen.

OffsetOutputs

Offset i slavens dataarea för modulens utdata. Beräknas av I/O-hanteringens vid initiering av modulen.

4.4 Pb_Do

Objekt som konfigurerar en Pb-modul med Digigala utgångar eller en uppsättning ”digitala” utparametrar.

Attribut*Process*

Process som hanterar modulen (1=PLC).

ThreadObject

PLC-tråd som hanterar modulen (om Process=1).

NumberOfChannels

Antal kanaler på modulen alternativt antal digitala utparametrar i slaven. Kan ligga i intervallet 1-32. Observera att alla kanaler inte behöver vara konfigurerade eller kopplade. Default 32.

Status

Modulens status. Sköts av I/O-hanteringens.

0 = ej initierad

1 = initierad

TestMask1

16 bitars heltal som utgör en bitmask som beskriver vilka kanaler som ska skrivas från TestValue i stället för från den kopplade signalen. Bit 0 motsvarar kanal 0. TestMask1 gäller för kanalerna 0-15. Ska normalt vara 0.

TestMask2

Som TestMask1 fast för kanal 16-31.

2003-06-06

InvMask1

Mask som beskriver invertering av kanaler enligt samma princip som TestMask1. Ska normalt vara 0.

InvMask2

Som InvMask1 fast för kanal 16-31.

TestValue1

16 bitars heltal som utgör en bitmask som beskriver värde som ska skrivas till de kanaler som satts i testläge enligt TestMask1. Bit 0 motsvarar kanal 0. TestValue1 gäller för kanalerna 0-15. Ska normalt vara 0.

TestValue2

Som TestValue1 fast för kanal 16-31.

FixedOutValue1

16 bitars heltal som utgör en bitmask som skrivs till utgångarna när I/O-hantering avslutas eller stallas. FixedOutValue1 gäller för kanalerna 0-15.

FixedOutValue2

Som FixedOutValue2 fast för kanal 16-31.

BytesOfInput

Antal byte indata för modulen. Beräknas av I/O-hantering vid initiering av modulen.

BytesOfOutput

Antal byte med utdata för modulen. Beräknas av I/O-hantering vid initiering av modulen.

OffsetInputs

Offset i slavens dataarea för modulens indata. Beräknas av I/O-hantering vid initiering av modulen.

OffsetOutputs

Offset i slavens dataarea för modulens utdata. Beräknas av I/O-hantering vid initiering av modulen.

4.5 Pb_Ai

Objekt som konfigurerar en Pb-modul med Analoga ingångar eller en uppsättning "analog" inparametrar. Det förutsätts att råvärdet är i heltalsformat, 2-komplement eller unsigned. För övrigt hanteras värdet enligt normala konverteringsmetoder.

Attribut

2003-06-06

Process

Process som hanterar modulen (1=PLC).

ThreadObject

PLC-tråd som hanterar modulen (om Process=1).

NumberOfChannels

Antal kanaler på modulen alternativt antal analoga inparametrar i slaven. Kan ligga i intervallet 1-32. Observera att alla kanaler inte behöver vara konfigurerade eller kopplade. Default 8.

BytesPerChannel

Antal byte per kanal/parameter i slavens dataarea. Default 2.

ByteSwap

= 1 om råvärdet ska bytevändas innan konvertering. Default 1.

Status

Modulens status. Sköts av I/O-hanteringen.

0 = ej initierad

1 = initierad

BytesOfInput

Antal byte indata för modulen. Beräknas av I/O-hanteringen vid initiering av modulen.

BytesOfOutput

Antal byte med utdata för modulen. Beräknas av I/O-hanteringen vid initiering av modulen.

OffsetInputs

Offset i slavens dataarea för modulens indata. Beräknas av I/O-hanteringen vid initiering av modulen.

OffsetOutputs

Offset i slavens dataarea för modulens utdata. Beräknas av I/O-hanteringen vid initiering av modulen.

4.6 Pb_Ao

Objekt som konfigurerar en Pb-modul med Analoga utgångar eller en uppsättning ”analog” utparametrar. Det förutsätts att råvärdet är i heltalsformat, 2-komplement. För övrigt hanteras värdet enligt normala konverteringsmetoder.

Attribut*Process*

Process som hanterar modulen (1=PLC).

ThreadObject

PLC-tråd som hanterar modulen (om Process=1).

NumberOfChannels

Antal kanaler på modulen alternativt antal analoga utparametrar i slaven. Kan ligga i intervallet 1-32. Observera att alla kanaler inte behöver vara konfigurerade eller kopplade. Default 4.

BytesPerChannel

Antal byte per kanal/parameter i slavens dataarea. Default 2.

ByteSwap

= 1 om råvärdet ska bytevändas innan utställning. Default 1.

Status

Modulens status. Sköts av I/O-hanteringen.

0 = ej initierad

1 = initierad

BytesOfInput

Antal byte indata för modulen. Beräknas av I/O-hanteringen vid initiering av modulen.

BytesOfOutput

Antal byte med utdata för modulen. Beräknas av I/O-hanteringen vid initiering av modulen.

OffsetInputs

Offset i slavens dataarea för modulens indata. Beräknas av I/O-hanteringen vid initiering av modulen.

OffsetOutputs

Offset i slavens dataarea för modulens utdata. Beräknas av I/O-hanteringen vid initiering av modulen.

5 Konfigurering via EXCEL och textfiler

Profibus-I/O kan konfigureras från en beskrivningsfil tillverkad med EXCEL och exporterad till textformat på samma sätt som vanligt I/O. Exempel på

2003-06-06

EXCEL-fil med förklarande rubriker finns på intranätet (förmodligen strax intill den länk som tog dig hit).