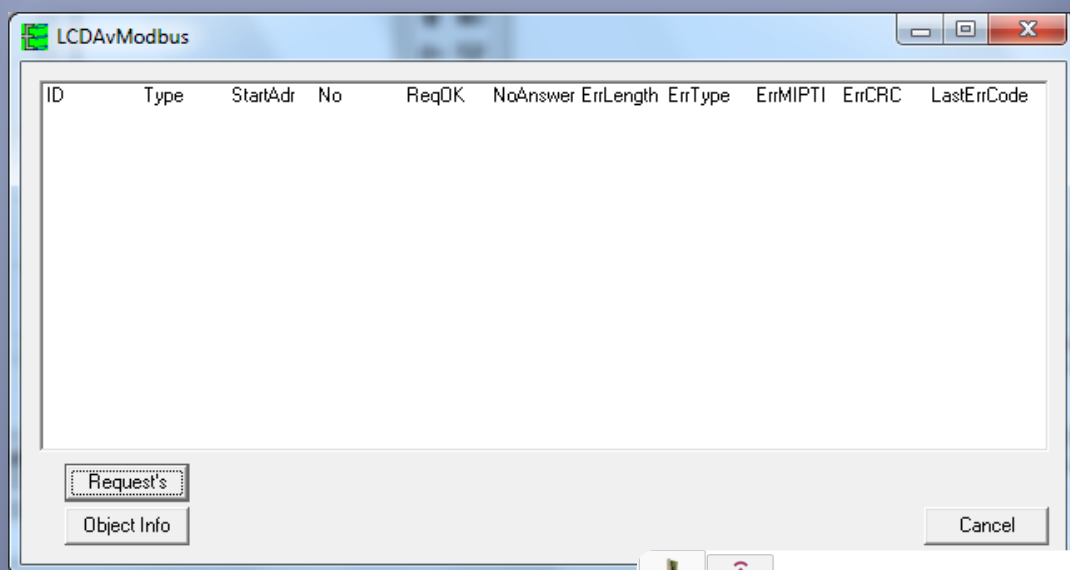




UCM		Intern					
Typ	ID	Inaktiverad	Namn	Enhet ID	Modulbärare	Modul	Plint
	94		CP01 CIRCULATIONS PUMP	002	001	A	5
	96		TF01 TILLUFTSFLÄKT	002	001	A	6
	97		FF01 FRÄNLUFTSFLÄKT	002	001	A	1
	98		ST23A-B SPJÄLLSTÄLLDON	002	001	A	2
	99		ST24A-C SPJÄLLSTÄLLDON	002	001	A	7
	104		ST23A-B SPJÄLLSTÄLLDON	002	001	D	2
	105		SO01 SERVICEOMKOPPLARE	002	001	D	5
	106		ST24A-C SPJÄLLSTÄLLJÖ	002	001	D	8

Modbus för Avalon och Atlantis

Användarhandbok



ANSVARSBEGRÄNSNING

All information i denna handbok har kontrollerats noggrant och bedöms vara korrekt. Emellertid lämnar Larmia Control AB inga garantier vad gäller manualens innehåll. Användare av denna manual ombeds rapportera felaktigheter, tvetydigheter eller oklarheter till Larmia Control AB, för eventuella korrigeringar i framtida utgåvor. Informationen i denna handbok kan ändras utan föregående meddelanden.

Mjukvaran som beskrivs i handboken levereras under licens från Larmia Control AB och får endast användas eller kopieras enligt licensvillkoren. Ingen del av denna bok får återges eller överföras i någon form eller på något sätt, elektroniskt eller mekaniskt, för något som helst ändamål utan uttryckligt skriftligt medgivande från Larmia Control AB.

COPYRIGHT

© Larmia Control AB. Med ensamrätt.

VARUMÄRKEN

MS-DOS, Windows, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista och Windows 7 är registrerade varumärken som tillhör Microsoft Corporation.

Andra produktnamn som förekommer i denna bok används enbart i identifieringssyfte och kan vara ägarens registrerade varumärken.

DISCLAIMER

The information in this manual has been carefully checked and is believed to be correct. Larmia Control AB however, makes no warranties as regards the contents of this manual and users are requested to report errors, discrepancies or ambiguities to Larmia Control AB, so that corrections may be made in future editions. The information in this handbook is subject to change without prior notification.

The software described in this book is supplied under licence by Larmia Control AB and may be used or copied only in accordance with the terms of the licence. No part of this book may be reproduced or transmitted in any form, in any fashion, electronically or mechanically, without the express, written permission of Larmia Control AB.

COPYRIGHT

© Larmia Control AB. All rights reserved.

TRADEMARKS

MS-DOS, Windows, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista and Windows 7 are registered trademarks of Microsoft Corporation.

Some product names mentioned in this book are used for identification purposes only and may be the registered trademarks of their respective companies.

Innehåll

Översikt	1
JBus	1
Atlantis.....	2
Editering	2
Automatisk styrning av analoga och digitala utgångar.	2
Kommunikationskontroll.....	3
Avalon.....	4
Editering	4
Automatisk styrning av analoga och digitala utgångar	4
Kommunikationskontroll.....	5
Datatyper.....	6
Funktionskoder.....	7
Skalning av värden.....	9
Analoga värden.....	9
Digitala värden	9
Installation av Modbus till Atlantis.....	10
Konfiguration.....	11
Ini-fil.....	11
Exempel på ini-filer.....	12
Kommunikation via RS485.....	12
Kommunikation via TCP	12
Drivrutin	13
Driftsättning	14
Felkoder.....	16
Anpassningar	17
Enheter som saknar vissa register.....	17
Enheter som inte kan skicka tillräckligt mycket data i meddelandet.....	17
Hantering av 32-bitars tal.....	17
Skriva bitvärden till Holding Register	17
Modem	19

Översikt

Modbus hanterar fyra olika register:

- Input Register
- Holding Register
- Input Status
- Coil

Olika objekttyper i Larmias system läser som standard från olika register:

Objekttyp	Register
Analog in	Input Register
Puls	Input Register
Analog ut	Holding Register
Manöver	Coil
Indikering	Input Status
Larm	Input Status

Ibland förekommer det att enheter t.ex. inte implementerar Input Register utan endast använder Holding Register, vilket medför att Analog in-objekt inte fungerar. Detta kan dock enkelt lösas via inställningar i drivare.

Se avsnittet [Anpassningar](#) för mer information.

JBus

JBus är ett protokoll som baseras på Modbus, och de två protokollen är i stort sett identiska. Den egentliga skillnaden rör registeradresseringen; Modbus-register börjar på adress 0 medan JBus-register börjar på adress 1.

Larmias Modbus-drivare hanterar både Modbus och Jbus.

| **OBSERVERA** skillnaderna i adressering när du editerar objekten.

Editering

Använd adresstypen **MODBUS** i ED10.

Analogue in
N24

Namn
Analog in

Tag-namn

Minimum Maximum Filter (s) Decimals Analog-enhet
-50 150 1 1 °C

☐ Inaktiverad
☒ Loggning ☐ Skalning

Adresstyp Slinga Enhet Adress Datatyp
MODBUS 1 1 1

☐ Används av följande objekt...

För denna adresstyp finns fyra fält: **Slinga**, **Enhet**, **Adress** och **Datatyp**.

Fält	Beskrivning
Slinga	Nummer på slingan
Enhet	Enhetsnummer
Adress	Modbus-adress (registeradress)
Datatyp	Datatyp. Utlämnas om värdet är ett 16-bitars heltal eller bitvärde. Se Datatyper för mer information.

OBSERVERA att den första Modbus-adressen är 0. Vissa adresslistor börjar på 1 och man ska då ta adressen enligt listan -1.

Automatisk styrning av analoga och digitala utgångar.

Om **DHC-Styrning** markeras i editeringen tar PC:n över styrningen av objektet. I **Auto**-fältet kan man då ange ett värde eller en koppling som styr objektet.

Kommunikationskontroll

Tidsgränsen för kommunikationsfel är som standard 120 sekunder men kan enkelt ändras till valfritt värde. För att ändra tidsgränsen till t.ex. 60 sekunder ska raden `COMERR_DELAY=60` läggas till i filen **Larmia.edt\modbusX.ini**, där X är numret på slingan.

För att kontrollera om det uppstår kommunikationsfel mot en specifik Modbus-enhet kan man skapa ett Larmobjekt med adresstypen **COM-Control**.

The screenshot shows the configuration window for a Modbus communication error alarm. At the top, there is a red circular icon with a left-pointing arrow. The window title is "Larm N25". The form contains the following fields and options:

- Namn:** Modbus Komfel Slinga 1, Enhet 1
- Tag-namn:** (empty)
- Larmtext:** KOMFEL
- NO/NC:** NO (dropdown)
- Larmklass:** B (dropdown)
- Larmfördröjning (s):** 1
- ☐ Inaktiverad
- ☐ Kvitteringsblockering
- ☐ Inga händelser
- ☐ Skalning
- Koppling:** (empty)
- Larmblockering:** (empty)
- Adresstyp:** COM-Control (dropdown)
- Drivare:** Modbus Larm (dropdown)
- Slinga:** 01
- Enhet:** 001
- ☐ Används av följande objekt...

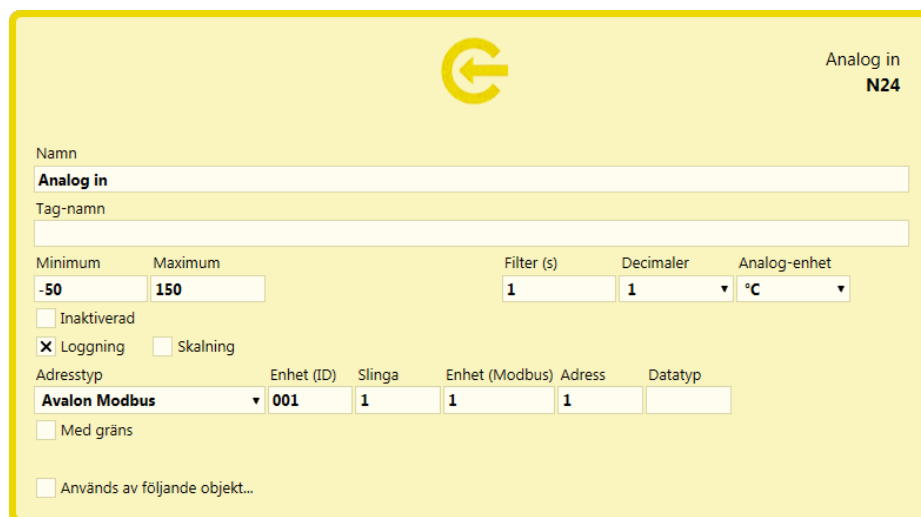
För denna adresstyp finns tre fält: **Drivare**, **Slinga** och **Enhet**.

Fält	Beskrivning
Drivare	Anger vilken drivare som ska kontrolleras. I detta fall anges Modbus Larm .
Slinga	Nummer på slingan
Enhet	Enhetsnummer

Avalon

Editering

Använd adresstypen **Avalon Modbus** i ED10.



För denna adresstyp finns fem fält: **Enhet (ID)**, **Slinga**, **Enhet (Modbus)**, **Adress** och **Datatyp**.

Fält	Beskrivning
Enhet (ID)	PLC-ID't för Avalon
Slinga	Nummer på slingan
Enhet (Modbus)	Enhetsnummer
Adress	Modbus-adress (registeradress)
Datatyp	Datatyp. Utelämnas om värdet är ett 16-bitars heltal eller bitvärde. Se Datatyper för mer information.

OBSERVERA att den första Modbus-adressen är 0. Vissa adresslistor börjar på 1 och man ska då ta adressen enligt listan -1.

Automatisk styrning av analoga och digitala utgångar

Om ett värde eller en koppling har angetts i **Auto**-fältet för en analog- eller digital utgång så tar Avalon över styrningen av detta objekt.

Om **Auto**-fältet lämnas tomt bestäms värdet av Modbus-enheten.

Kommunikationskontroll

I Avalon kontrolleras kommunikationen genom att Avalon förväntar sig ett nytt värde från samtliga in- och utgångar inom 10 sekunder. Om ett objekt inte får något värde inom denna tid genereras ett kommunikationsfel för detta objekt samt att systemfunktionen MKOM går TILL.

Tidsgränsen för kommunikationsfel är änderingsbar. För att ändra tidsgränsen till t.ex. 20 sekunder ska raden `COMERRDELAY=20` läggas till i filen **Larmia.prg\Avalon.ini**.

För att kontrollera om det uppstår kommunikationsfel mot en specifik Modbus-enhet kan man skapa ett Larmobjekt i Avalon med adresstypen **Avalon COM-Control**.

The screenshot shows the configuration window for a Modbus communication fault alarm. The 'Name' field is 'Modbus Komfel Slinga 1, Enhet 1'. The 'Alarmtext' is 'KOMFEL'. The 'NO/NC' is set to 'NO', 'Alarmklass' is 'B', and 'Larmfördröjning (s)' is '1'. There are checkboxes for 'Inaktiverad', 'Kvitteringsblockering', 'Inga händelser', and 'Skalning'. The 'Koppling' section shows 'Larmblockering'. The 'Adresstyp' is 'Avalon COM-Control', 'Enhet (ID)' is '001', and 'Adress' is 'MOD1:1'. There is a checkbox for 'Används av följande objekt...'.

För denna adresstyp finns två fält: **Enhet (ID)** och **Adress**.

Fält	Beskrivning
Enhet (ID)	PLC-ID't för Avalon
Adress	Adressträng för KOM-kontrollen. I detta fall anges adressen med formatet <code>MODs:y:</code> där <code>s</code> är slingnumret och <code>y</code> är Modbus-enhetsnumret.

Följande exempel kontrollerar enhet 1 på slinga 1, anslutet till Avalon 001:

```
Enhet (ID) : 001
Adress : MOD1:1:
```

OBSERVERA det avslutande kolon-tecknet (:) i adressträngen.

Datatyper

Datatyp	Beskrivning
BYTE	8 bitar med tecken
UBYTE	8 bitar utan tecken
INT16	16 bitars heltal
UINT16	16 bitars heltal utan tecken
INT32B	32 bitars heltal (Big-Endian, Modbus-standard)
INT32	32 bitars heltal (Little-Endian)
FLOATB	Flyttal (Big-Endian, Modbus-standard)
FLOAT	Flyttal (Little-Endian)
REGBITx	Läser bit x i ett 16-bitars heltal. x är ett tal mellan 0-15. Denna datatyp kan endast användas för Indikeringar och Larm. <i>För Analog ut, se Anpassningar för mer information.</i>
BIN8	Används om ett Analog ut-objekt skall skriva värde till ett digitalt register (Force Multiple Coil).

Funktionskoder

I Modbus-standarden finns fyra olika register med olika funktionskoder:

Register	Typ	Funktionskod
Coil Status	Binära (R/W)	1
Input Status	Binära (R)	2
Holding Register	Analoga (R/W)	3
Input Register	Analoga (R)	4

De olika objekttyperna kopplas med automatik till specifika register:

Objekttyp	Register
Manöver	Coil Status
Larm och Indikering	Input Status
Analog Ut	Holding Register
Analog In och Pulsobjekt	Input Register

Ibland förekommer det att enheter t.ex. inte implementerar Input Register utan endast använder Holding Register, vilket medför att Analog in-objekt inte fungerar. Man kan då för varje objekt ange vilken funktionskod som skall användas.

I fältet **Datatyp** anges vilken funktionskod som ska användas för läsning genom att ange **FCR="Funk. kod"**, eller för skrivning genom att ange **FCW="Funk. kod"**.

Slinga	Enhet (Modbus)	Adress	Datatyp
1	1	1	FCR=3

Giltiga funktionskoder för FCR:

Objekttyp	Defaultvärde	Alternativa funktionskoder
Manöver	1	2
Larm och Indikering	2	1
Analog Ut	3	4
Analog In och Pulsobjekt	4	3

Giltiga funktionskoder för FCW:

Objekttyp	Defaultvärde	Alternativa funktionskoder
Analog Ut	6	16

Ett alternativ till detta är att göra en global inställning i filen **ModbusX.ini** genom att ange funktionskoden i parametrarna `READ_COIL_CODE`, `READ_DISCRET_INPUT_CODE`, `READ_HOLDING_CODE` eller `READ_INPUT_REGISTER_CODE`.

Skalning av värden

Analoga värden

I fältet Skalningsuttryck anges hur värdet från en modbusenhet ska skalas:

Ex.

$X/100$ – Dividerar det modbus värdet med 100.

$X*50$ – Multiplicerar det modbus värdet med 50

OBS !!

Vid skalning skall man tänka på hur värdet skall skalas då det läses in. Detta gäller både analog in och analog ut. Om objektet är en analog ut kan värdet skickas ut till ett modbusregister. I detta fall kommer skalningen automatiskt inverteras (Detta sker i drivaren)

Digitala värden

Om ingångsvärdet inte är digitalt kan man använda ett skalningsuttryck för att definiera när objektet anses vara **TILL** resp. **FRÅN**. I fältet Skalningsuttryck anges definitionen:

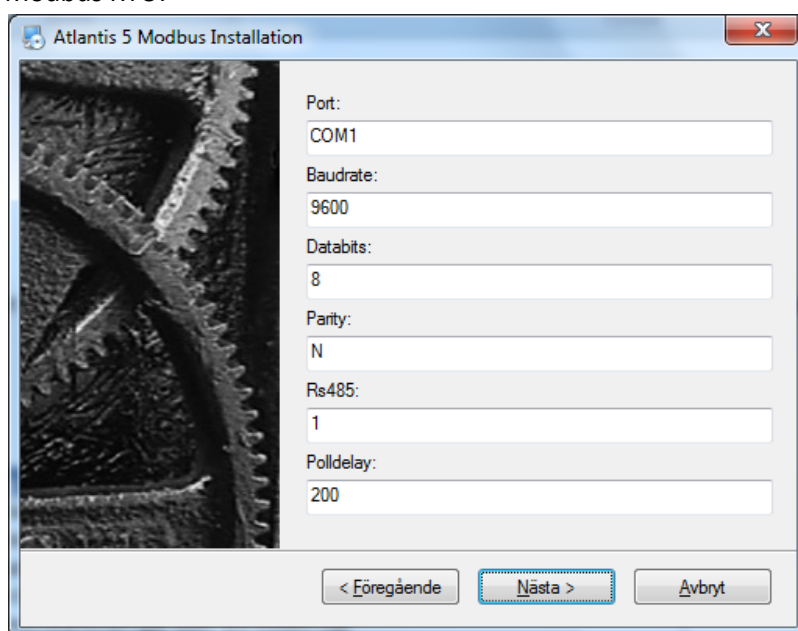
$X > 8$ – Om ingångsvärdet är större än 8 är objektet **TILL**, annars **FRÅN**.

Installation av Modbus till Atlantis

Starta installationsprogrammet och välj om du vill kommunicera via **Modbus RTU (RS485)** eller **Modbus TCP (TCP/IP)**.

Beroende på vilken typ av Modbus som väljs kan man redan under installationsförloppet ange de vanligaste inställningarna.

Modbus RTU:

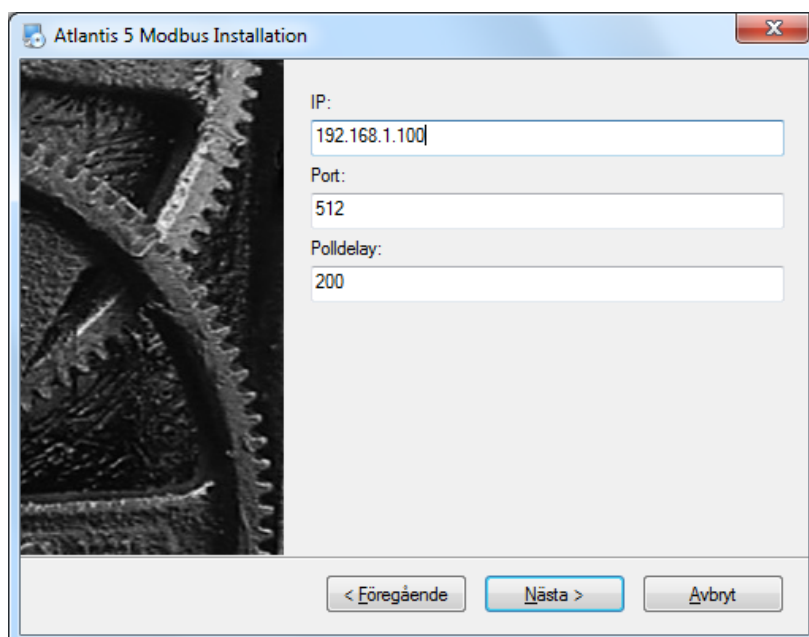


The screenshot shows the 'Atlantis 5 Modbus Installation' window. On the left is a background image of interlocking gears. On the right, there are input fields for the following settings:

- Port: COM1
- Baudrate: 9600
- Databits: 8
- Parity: N
- Rs485: 1
- Polldelay: 200

At the bottom, there are three buttons: '< Föregående', 'Nästa >' (which is highlighted with a blue border), and 'Avbryt'.

Modbus TCP:



The screenshot shows the 'Atlantis 5 Modbus Installation' window. On the left is a background image of interlocking gears. On the right, there are input fields for the following settings:

- IP: 192.168.1.100
- Port: 512
- Polldelay: 200

At the bottom, there are three buttons: '< Föregående', 'Nästa >' (which is highlighted with a blue border), and 'Avbryt'.

Konfiguration

Ini-fil

För varje slinga (port) finns en ini-fil. Filen heter **ModbusX.ini**, där **X** är numret på slingan.

I Avalon ligger filen i mappen **Larmia.prg** och i Atlantis ligger den i mappen **Larmia.edt**.

I filen kan följande inställningar göras:

Namn	Förklaring	Std. värde
PORT	Kommunikationsport eller IP-port. Ex: COM1 eller 192.168.1.10/502	
BAUDRATE	Kommunikationshastighet.	9600
DATABITS	Antal databitar.	8
PARITY	Partiet. N , E eller O .	N
RS485	Anger om kommunikationen går över ett RS485-nät. 1 om kommunikation går över RS485, annars 0 .	1
MODBUSTCP	Anger om kommunikationen går över nätverk (Modbus TCP). 1 om kommunikationen går över nätverk, annars 0 .	0
POLLDELAY	Antal millisekunder mellan varje fråga. Halva tiden före och halva tiden efter frågan.	200
READ_COIL_CODE	Anger vilken funktionskod som skall användas av Manövrar.	1
READ_DISCRET_INPUT_CODE	Anger vilken funktionskod som skall användas av Indikeringar och Larm.	2
READ_HOLDING_CODE	Anger vilken funktionskod som skall användas av Analoga utgångar	3
READ_INPUT_REGISTER_CODE	Anger vilken funktionskod som skall användas av Analoga ingångar.	4
MAX_NO_REGISTER_READ	Anger det maximala antal register som kan avfrågas i en fråga.	32
MAX_NO_DISCRETE_READ	Anger det maximala antal punkter som kan avfrågas i en fråga.	64

EDITHEX	Anger om enhets-ID'n och adresser har skrivits hexadecimalt i editeringen. 1 om hexadecimalt har använts, annars 0 (decimalt).	0
TIMEOUTDELAY	Antal millisekunder som enheten har på sig att svara	1000
COMERR_DELAY	Antal sekunder som en Modbus-enhet har på sig att svara innan ett kommunikationslarm genereras. <i>Gäller endast Atlantis.</i>	120
DELAYRTS	Fördröjning (i millisekunder) efter det att ett meddelande har skickats tills RTS-signalen sätts låg.	15
RESTART_AT_IPFAILURE	Anger om drivrutinen ska starta om eller etablera ny kontakt om den tappar kontakt med IP-porten. 1 om drivrutinen ska starta om, 0 om ny kontakt ska upprättas.	0
WRITECOM	Anger om kommunikationen ska loggas till filen Larmia.dok\LCMSG_MODBUS_COM.TXT . 1 för att logga kommunikation, annars 0 .	0

Om en parameter inte har angetts i **Ini**-filen kommer drivaren internt att använda standardvärdet.

Exempel på ini-filer

Kommunikation via RS485

```
PORT=COM1
BAUDRATE=9600
DATABITS=8
PARITY=N
RS485=1
```

Kommunikation via TCP

```
PORT=192.168.3.10/502
MODBUSTCP=1
```

```
READ_INPUT_REGISTER_CODE=3
MAX_NO_REGISTER_READ=8
```

Drivrutin

För att starta drivrutinen manuellt måste man skapa en genväg till filen och som argument ange vilken slinga den skall köra mot.

Filen heter **LCDModbusV2.exe** i Atlantis och **LCDAvModbus.exe** i Avalon, och i båda fallen ligger den i **Larmia.prg**-mappen

OBSERVERA att drivrutinen normalt bör köras som tjänst. I detta fall måste startkommandot läggas in i **Larmia.prg\ProcessInfo.ini**.

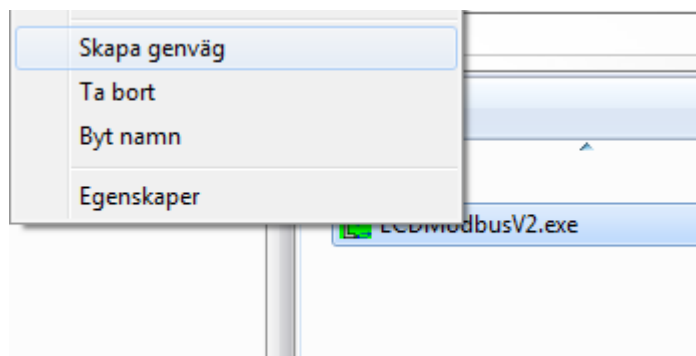
Driftsättning

Vid driftsättning av Modbus-kommunikation bör man först börja med att testa kommunikationen med något fristående program. Ett enkelt men fullt funktionsdugligt program är **modscan32** som även finns i en gratis utvärderingsversion (<http://www.win-tech.com>).

Med detta program kan kontrollera att kommunikation med enheten fungerar samt att enheten svarar på alla adresser.

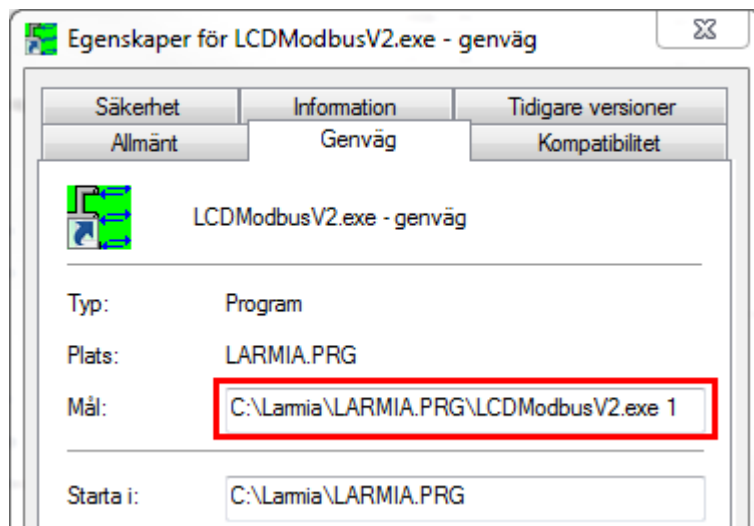
När man har bekräftat att kommunikationen med enheten/enheterna fungerar som förväntat går man vidare genom att starta Modbus-drivaren manuellt.

Skapa först en **Genväg** till drivaren genom att högerklicka på filen och välja **Skapa genväg**.



Högerklicka sedan på den skapade **Genvägen** och välj **Egenskaper**.
Välj fliken **Genväg**.

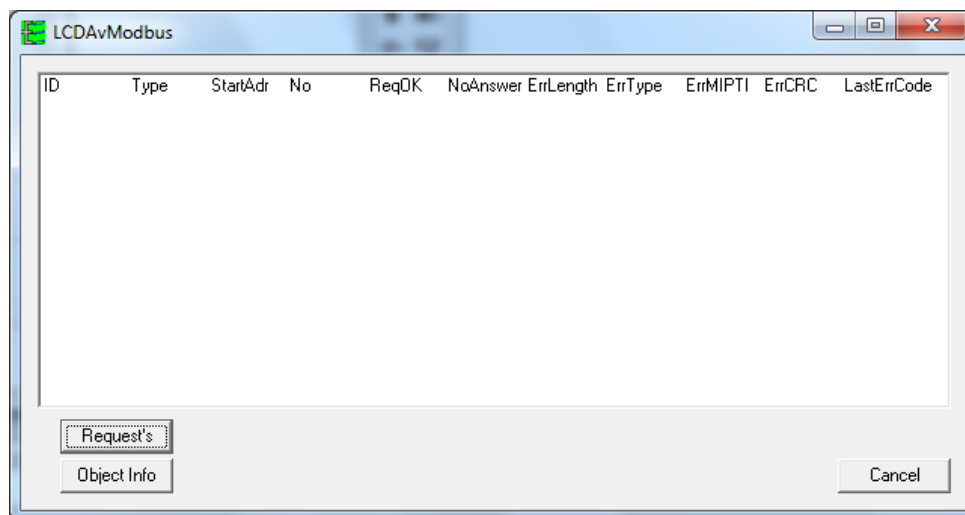
I **Mål**-fältet lägger du sedan till numret på den slinga som ska köras.



OBSERVERA att det ska vara ett mellanslag mellan programnamnet och siffran.

OBSERVERA att det måste finnas en motsvarande **ModbusX.ini**-fil där X'et motsvarar slingnumret.

När man startar Modbusdrivaren manuellt visas en dialog där man kan se alla de frågor som drivrutinen skickar till enheterna.



Dialogen visar en mängd data och information om meddelandena i tabellform.

Namn	Förklaring
ID	Enhetens Modbus-nummer.
Type	DI - Digital ingång (Status Register). DO - Digital utgång (Coil). AI - Analog ingång (Analog Register). AO - Analog utgång (Holding Register).
StartAdr	Frågans startadress.
No	Antal värden som läses i denna fråga.
ReqOK	Antal korrekta svar på frågan sedan drivaren startades.
NoAnswer	Antal frågor utan svar sedan drivaren startades.
ErrLength	Antal gånger som längden på svaret (antalet bytes) inte stämde överens med det förväntade.
ErrType	Antal gånger som enheten förväntade sig en annan typ av fråga på denna adress.
ErrMIPTI	Antal gånger som enheten har svarat med en felaktig "Transaction Identifier". <i>Endast vid kommunikation via Modbus TCP.</i>
ErrCRC	Antal gånger som svaret har haft fel checksumma.
LastErrCode	Den senaste felkoden som inkom som svar. <i>Se Felkoder för mer information.</i>

Klicka på **Request's** för att uppdatera tabellen.

Drivaren kan också visa inlästa värden, och information om dessa, genom att klicka på **Object Info**. Även här uppdateras tabellen för varje knapptryck.

Felkoder

Felkod	Beskrivning
01	Ogiltig typ av fråga. Kan t.ex. betyda att det inte finns några Holding Register på angiven adress.
02	Ogiltig startadress.
03	Ogiltigt antal register som skall hämtas i samma fråga.

Anpassningar

Enheter som saknar vissa register

Ibland förekommer det att enheter inte implementerar Input Register utan endast använder Holding Register. Detta gör att Analog in-objekt inte fungerar.

Lös detta genom att ange `READ_INPUT_REGISTER_CODE=3` i korrekt Ini-fil. Då kommer Analog in-objekt att fråga på Holding Register.

Enheter som inte kan skicka tillräckligt mycket data i meddelandet

Som standard är det maximala antal register som skickas i ett meddelande 32 stycken, men vissa enheter klarar inte att skicka så stora meddelanden.

Lös detta genom att ange ett värde lägre än 32 i parametern `MAX_NO_REGISTER_READ` i korrekt Ini-fil.

Hantering av 32-bitars tal

Ibland förekommer det att enheter inte följer Modbus-standardens gällande hanteringen av 32-bitars heltal, byte:arna kastas om.

Standarden säger att s.k. Big-Endian skall användas. I detta fall anges `INT32B` i fältet **Datotyp**.

Om enheten istället använder s.k. Little-Endian skall `INT32` anges i fältet **Datotyp**.

Det samma gäller för flyttal; använd `FLOATB` resp. `FLOAT`.

Skriva bitvärden till Holding Register

Ibland förekommer det att man vill skriva ett värde från t.ex. ett Manöver-objekt till en viss bit i ett Holding Register.

Lös detta genom att skapa ett Villkor som skapar värdet till registret.

Villkorsingång	Objektkoppling
A	Manöver som skall skrivas till första biten i registret
B	Manöver som skall skrivas till andra biten i registret
C	Manöver som skall skrivas till tredje biten i registret
D	Manöver som skall skrivas till fjärde biten i registret

Utgång: $A+B*2+C*4+D*8$

Detta värde kopplas till ett Analog ut-objekt som har adressen till registret.

Modem

Vid anslutning av enheter som kommunicerar enl. Modbus RTU (2-tråd RS485) används t.ex. MDW-45. Modemet ansluts sedan mot ledig COM-port. Alternativt används en IP-port, t.ex. EDW-100. Denna kan konfigureras för Modbus vilket gör att Modbus TCP kan användas i stället.