

Styrssystem

Unity Pro M340

Kom-igång med
Seriell Modbus kommunikation

2013-11-12



INNEHÅLLSFÖRTECKNING

1	MODBUS I UNITY PRO	3
1.1	READ_VAR OCH WRITE_VAR	3
2	ANPASSADE FUNKTIONSBLOCK	3
2.1	FUNKTIONSBLOCK	3
2.2	TILLÅTA DYNAMIC ARRAYS	4
2.3	SERIEPORTS INSTÄLLNINGAR	5
2.4	MODBUS_READ_REQUEST_M340	6
2.5	MODBUS_WRITE_REQUEST_M340	7
2.6	FELRAPPORTERING	8
2.7	EXEMPEL PÅ ANVÄNDNING	9
3	ALLMÄNT	11

Schneider Electric Sverige AB

Box 1009, Eskilstunavägen 7
SE-611 29 Nyköping, Sweden
Tel: +46 (0)155-26 54 00
Fax: +46 (0)155-21 42 86

Org.nr 556259-3532
www.schneider-electric.se
info@se.schneider-electric.com

1 MODBUS I UNITY PRO

1.1 READ_VAR OCH WRITE_VAR

I UnityPro använder man funktionsblocken READ_VAR och WRITE_VAR för Modbus kommunikation. Men READ_VAR och WRITE_VAR är generella block för olika typer av kommunikation. Inte bara för Modbus. Detta gör de svåra att förstå och att börja använda.

Vill man prova dessa standardblock så finns de beskrivet i standardhjälpen för UnityPro och tas inte upp i detta dokument.

I nästa kapitel så beskrivs Modbus kommunikation med två anpassade funktionsblock som gör det lättare att komma igång och programmera Modbus i UnityPro mot M340 styrsystemet.


2 ANPASSADE FUNKTIONSBLOCK

Med detta dokument ska 2st funktionsblock medfölja. Dom ska du importera in i ditt UnityPro projekt. Dessa funktionsblock är baserade på READ_VAR och WRITE_VAR med skillnaden att de är anpassade bara för att läsa och skriva Modbus request på ett enkelt sätt.

Följande Modbus funktioner stöds av M340:

<u>Funktionsnamn</u>	<u>Funktionskod</u>
Read Coils	1
Read Discrete Inputs	2
Read Holding Registers	3
Read Input Register	4
Write Multiple Coils	15
Write Multiple Registers	16

2.1 FUNKTIONSBLOCK

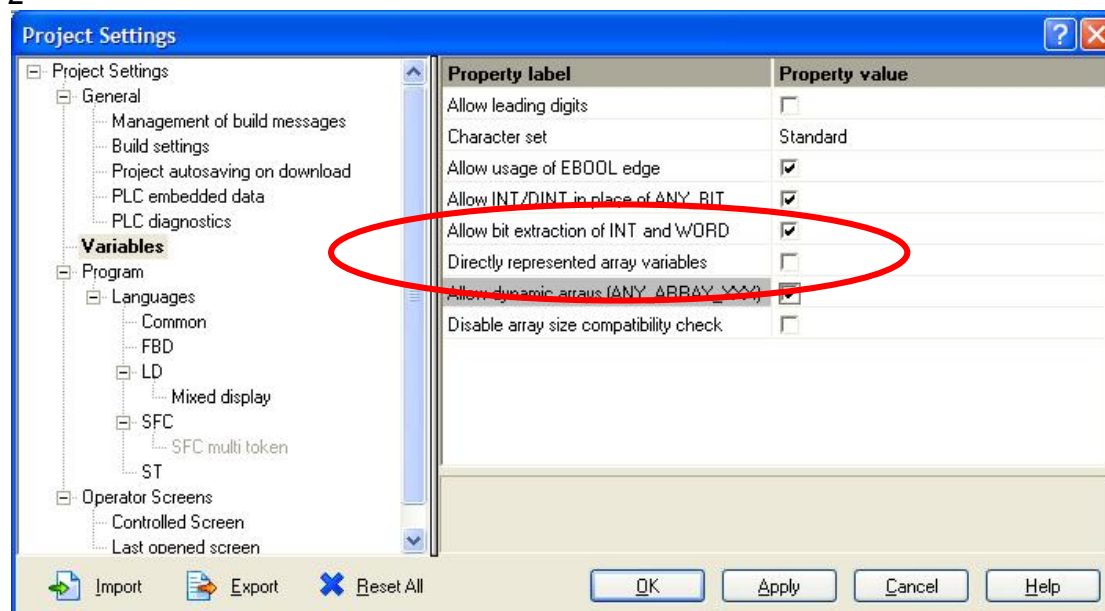
1. Funktionsblocken ska följa med detta dokument som en zip-fil med namnet "Modbus DFB M340.zip".
2. Packa upp zip-filen på din dator och kom ihåg vart du lägger filerna.
3. I Unity Pro's "Project Browser" högerklickar du på   **Derived FB Types** och väljer Import.
4. Leta nu upp funktionsblocken som du sparade undan ifrån zip-filen och importera dem in under mappen "Derived FB Types".

Nu kan du börja använda dem i ditt projekt.

2.2 TILLÅTA DYNAMIC ARRAYS

För att funktionsblocken ska fungera så måste man tillåta dynamic arrays i sitt projekt. Detta görs ifrån menyn Tools->Project Settings:

2

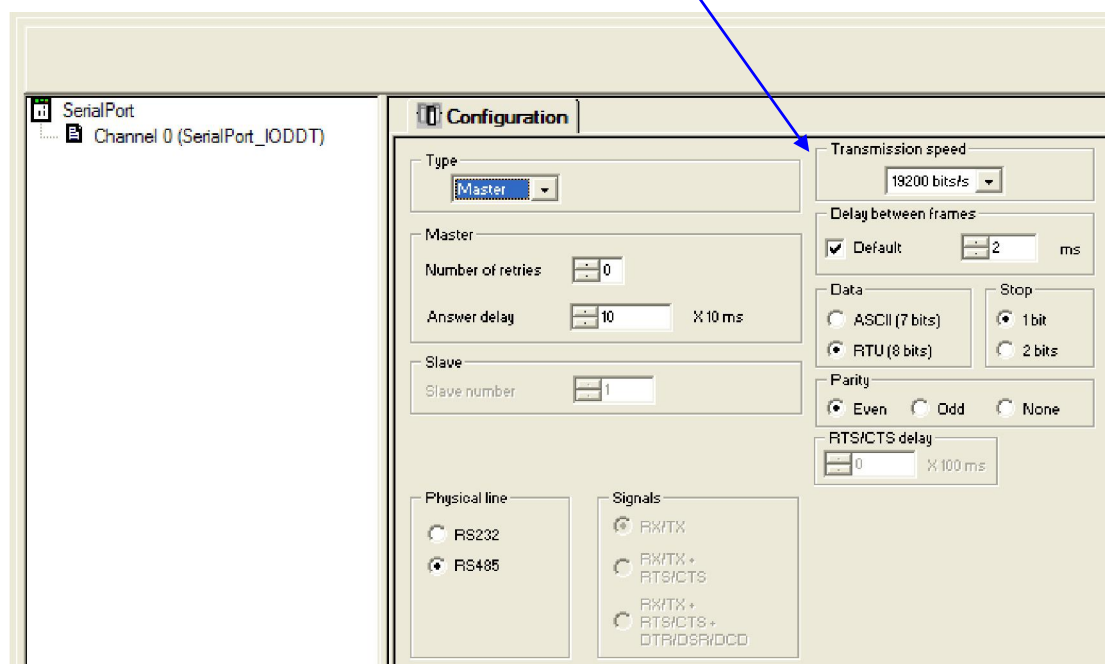


2.3 SERIEPORTS INSTÄLLNINGAR

Serieporten måste konfigureras för Modbus master i UnityPro. 'Type' ska vara Master och 'Physical line' ska vara RS485. Transmission speed, Data, Stop, och Parity är lättast att ställa in till det värde som de flesta slavarna har som standardinställning i eran installation. 'Delay between frames' ska ha sitt standardvärde 2ms.

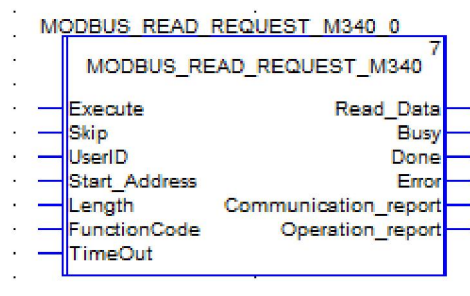
Parametern 'Answer delay' är max tiden PLC't väntar på svar ifrån en Modbus slav. Parametern Number of retries är hur många efterföljande försök PLC't max ska få prova att skicka om meddelandet till slaven som inte svarar innan den avbryter med felkod.

ATV omriktarna som [exempel](#) har följande standard inställningar för Modbus.



2.4 MODBUS_READ_REQUEST_M340

MODBUS_READ_REQUEST_M340 är till för att läsa ett eller flera objekts värde(n).

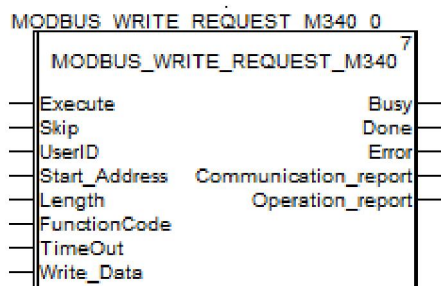


Beskrivning av funktionsblock

Pinne In	Data typ	Beskrivning
Execute	BOOL	Stigande flank: Läser valt/valda objekt en gång. Fallande flank: Nollställer alla ut-pinnar förutom Read_Data värdet.
Skip	BOOL	Hoppar över hela läs processen och sätter "Done" utgången direkt när "Execute" ingången blir hög.
UserID	INT	Slavnumret på Modbus produkten.
Start_Address	DINT	Modbus adressen där läsningen ska starta ifrån.
Length	INT	Hur många ord (16-bits) som ska läsas inkl. start adress.
FunctionCode	INT	Modbus funktionkod. Default = 3
TimeOut	INT	Max väntetid på svar ifrån serieportshanteringen. Upplösning: Värde x 100ms. Värde 10 = 1sek OBS! Detta värde måste vara högre än parametrarna "Number of retries" x "Answer delay" som man hittar under serieportens inställningarna. Se kapitel Serieports inställningar för mer info.
Pinne Ut	Data typ	Beskrivning
Read_Data	ARRAY OF INT	Denna array storlek måste vara minst lika stor som ingångsparametern "Length". Det går bra att ha den större är "Length" värdet. Standard är ARRAY[1..10] OF INT.
Busy	BOOL	Läsning pågår, ej klar.
Done	BOOL	Läsningen är utförd med eller utan fel. Se ERROR om fel har uppstått eller ej under läsningen.
Error	INT	Läsning klar med fel. Communication_report och Operation_report presenterar utförligare felrapportering.
Communication_report	INT	Se separat beskrivning under kapitel Felrapportering för med info.
Operation_Report		Se separat beskrivning under kapitel Felrapportering för med info.

2.5 MODBUS_WRITE_REQUEST_M340

MODBUS_WRITE_REQUEST_M340 är till för att skriva ett eller flera objekts värde(n).



Beskrivning av funktionsblock

Pinne In	Data typ	Beskrivning
EXECUTE	BOOL	Stigande flank: Skriver valt/valda objekt en gång. Fallande flank: Nollställer alla ut-pinnar förutom Write_Data värdet.
Skip	BOOL	Hoppar över hela skriv processen och sätter "Done" utgången direkt när "Execute" ingången blir hög.
UserID	INT	Slavnumret på Modbus produkten.
Start_Address	DINT	Modbus adressen där läsningen ska starta ifrån.
Length	INT	Hur många ord (16-bits) som ska skrivas inkl. start adress.
FunctionCode	INT	Modbus funktionkod. Default = 16
TimeOut	INT	Max väntetid på svar ifrån serieportshanteringen. Upplösning: Värde x 100ms. Värde 10 = 1sek OBS! Detta värde måste vara högre än parametrarna "Number of retries" x "Answer delay" som man hittar under serieportens inställningarna. Se kapitel Serieports inställningar för mer info.
Write_Data	ARRAY OF INT	Denna array storlek måste vara minst lika stor som ingångsparametern "Length". Det går bra att ha den större är "Length" värdet. Standard är ARRAY[1..10] OF INT.
Pinne Ut	Data typ	Beskrivning
Busy	BOOL	Läsning pågår, ej klar.
Done	BOOL	Läsningen är utförd med eller utan fel. Se ERROR om fel har uppstått eller ej under läsningen.
Error	INT	Läsning klar med fel. Communication_report och Operation_report presenterar utförligare felrapportering.
Communication_report	INT	Se separat beskrivning under kapitel Felrapportering för med info.
Operation_Report		Se separat beskrivning under kapitel Felrapportering för med info.

2.6 FELRAPPORTERING

Följande felrapportering dyker upp om Modbus läsning eller skrivning inte gick igenom. Fel koderna rapporteras ifrån READ_VAR och WRITE_VAR standardblocken som används som grund för dom anpassade funktionsblocken.

Mer om dessa fel koder hitta du i hjälpen för Unity Pro.

Communication_report och Operation_report meddelanden och dess betydelse.

Communication_report	0x00 = No Errors 0x01 = Time Out 0x02 = User cancelled 0x03 = Incorrect address format 0x04 = Incorrect destination address 0x06 = Incorrect Com Fb parameters 0x07 = Generic transmission interruption 0x09 = Buffer received too small 0x0B = No system resources 0xFF = Network exchange detected error
Operation_report	If Communication_report = 0x00 0x00 = No Errors 0x01: Cannot be treated 0x02: Incorrect response If Communication_report = 0xFF 0x07: Generic exchange detected error 0x0B: The destination device has no more resources. 0x0D: The device cannot be reached. 0x2B: SDO exchange detected error

2.7 EXEMPEL PÅ ANVÄNDNING

Vi vill läsa och skriva objekt för att styra en ATV omriktare på Modbus.
Först måste vi titta i dokumentationen för att hitta vad objekten har för Modbus adresser.

Vi vill läsa följande objekt:

Modbus address	CANopen address	Code	Read/Write	Name/Description/Possible values
3201	2002 / 2	ETA	R	Status word bit 0: Ready to switch on bit 1: Switched on bit 2: Operation enabled bit 3 = 0: No detected fault bit 3 = 1: Malfunction, detected fault (FAI) bit 4: Voltage disabled (still equals 0) bit 5: Quick stop bit 6: Switch on disabled bit 7 = 0: No alarm bit 7 = 1: Alarm present bit 8: Reserved bit 9 = 0: Forced local mode in progress (FLO) bit 9 = 1: No forced local mode bit 10 = 0: Reference not reached (transient state) bit 10 = 1: Reference reached (steady state) bit 11 = 0: LFRD reference normal bit 11 = 1: LFRD reference exceeded (< LSP or > HSP) Note: LFRD is expressed in rpm, LSP and HSP in Hz bits 12 and 13: Reserved bit 14 = 0: No stop imposed by STOP key on built-in keypad or on the remote display terminal bit 14 = 1: Stop imposed by STOP key on built-in keypad or on the remote display terminal bit 15 = 0: Forward rotation (output frequency) bit 15 = 1: Reverse rotation (output frequency)
3202	2002 / 3	rFr	R	Output frequency applied to the motor (signed value) Unit: • 1 = 0.1 Hz if bit 9 of CMI (page 20) = 0 • 1 ≈ 0.018 Hz (resolution 32767 points = 600 Hz) if bit 9 of CMI = 1
3203	2002 / 4	FrH	R	Frequency reference before ramp (absolute value) Unit: 0.1 Hz
3204	2002 / 5	LCr	R	Current in the motor Unit: 0.1 A
3205	2002 / 6	Otr	R	Motor torque Unit: 1% 100% = Nominal motor torque, calculated using the configuration parameters.

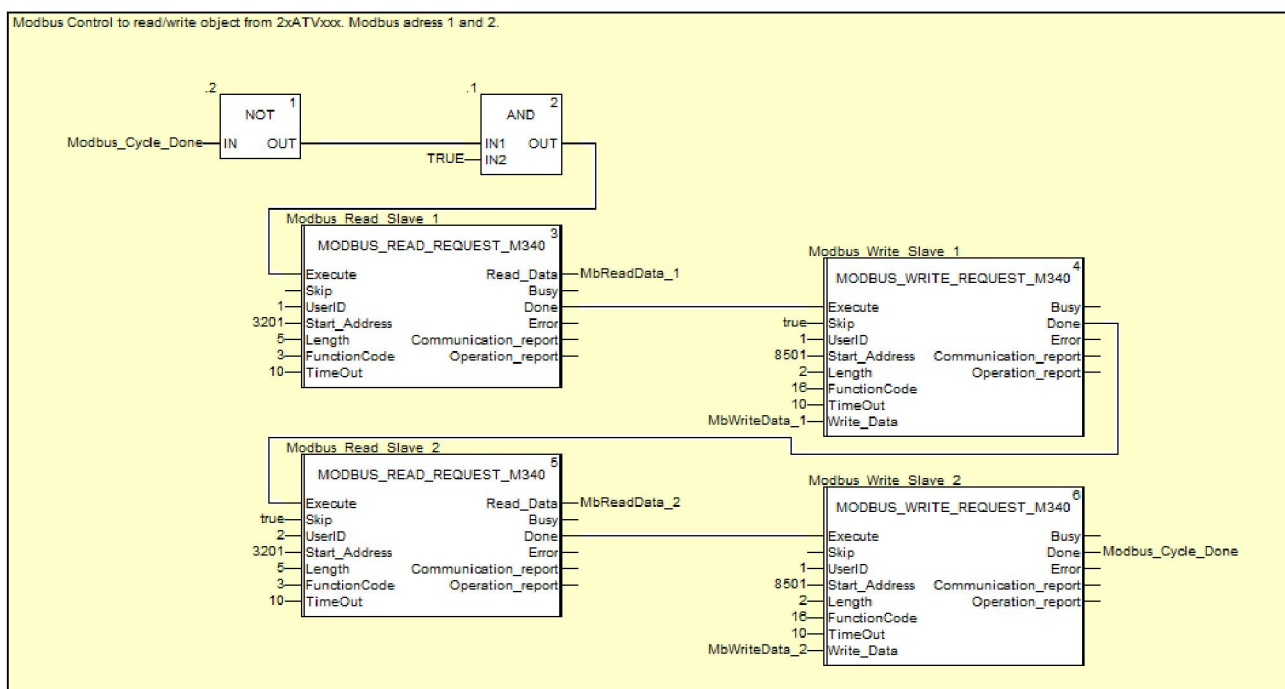
Vi vill skriva följande objekt:

Modbus address	CANopen address	Code	Read/Write	Name/Description/Possible values
8501	2037 / 2	CMD	R/W	Control word bit 0: "Switch on": active at 1 bit 1: "Disable Voltage": active at 0 bit 2: "Quick Stop": active at 0 bit 3: "Enable Operation": active at 1 bits 4 to 6: Reserved: set to 0 bit 7: Fault state reset: active on rising edge 0 -> 1 bits 8 to 10: Reserved: set to 0 For "Access level" LAC (page 35) = L1 or L2: bit 11 = 0: Forward direction command bit 11 = 1: Reverse direction command bit 12 = 0: No action bit 12 = 1: Stop command depending on the Stt "Stop type" parameter bit 13 = 0: No action bit 13 = 1: Injection stop command bit 14 = 0: No action bit 14 = 1: Fast stop command bit 15: Reserved: set to 0
8502	2037 / 3	LFr	R/W	Frequency reference via the bus (signed value) Unit: • 1 = 0.1 Hz if bit 9 of CMI (page 20) = 0 • 1 ≈ 0.018 Hz (resolution 32767 points = 600 Hz) if bit 9 of CMI = 1

Här följer ett enkelt exempel med FBD programmering att läsa och skriva dessa objekt med medföljande funktionsblocken.

Läsning och skrivning görs först mot Modbus slav 1 och fortsätter därefter med slav 2. Genom att knyta samman Done utgången med Execute ingången på nästföljande funktionsblock blir programmeringen mycket enkel och överskådlig.

Ska maskinen innehålla någon slav som tillfälligt inte ska vara tillslagen så sätter man bara ingången Skip på den slavens funktionsblock för att enkelt koppla bort den. Då slipper man problem med att PLC't blir "hängande" och vänta på svar ifrån just den Modbus slaven.



Variablerna MbReadData_x och MbWriteData_x är av typen ARRAY[1..10] OF INT och innehåller följande värden när programmet körs.

MbReadData_x:

- [1] - 3201 Statusword (ETA)
- [2] - 3202 Actual Velocity (0.1 Hz)
- [3] - 3203 Actual Reference Velocity (0.1 Hz)
- [4] - 3204 Actual Motor Current (0.1 A)
- [5] - 3205 Actual Motor Torque (0.1%)
- [6] – [10] förblir tom

MbWriteData_x array:

- [1] - 8501 Controlword (CMD)
- [2] - 8502 Target Velocity (0.1 Hz)
- [3] - [10] förblir tom

3 ALLMÄNT

Schneider Electric tar inte ansvar för vare sig direkt eller indirekt för eventuella felaktigheter i supportdokumentet. Schneider Electric förbehåller sig rätten att utan förvarning utföra förändringar i supportdokumentet. Det är upp till användaren att utföra erforderliga prov i syfte att verifiera funktionen i aktuell applikation. I och med användande av supportdokumentet accepterar användaren ovanstående.