



Flexit GO

NO**VEILEDNING**

Ekstern Modbus-kommunikasjon

SV**ANVISNING**

Extern Modbus-kommunikation

EN**INSTRUCITON**

External Modbus communication

NO Innhold

1.	Koble til enheten	4
1.1.	NORDIC-serien	4
1.2.	ECONORDIC-serien.....	4
2.	Kommunikasjonsparametre	4
2.1.	Mapping av applikasjonsparametre.....	4
2.2.	Lese/skrivetilgang til Modbus-slaveobjekt	4
2.1.1.	Les holdingregistre, FC=03.....	5
2.1.2.	Skriv til enkelt register, FC=06	5
2.1.3.	Skriv til flere registre, FC=16	5
2.3.	Mapping av datatyper	7

SV Innehåll

1.	Ansluta till enheten	8
1.1.	NORDIC-serien	8
1.2.	ECONORDIC-serien.....	8
2.	Kommunikationsparametrar	8
2.1.	Mapping av programparametrar	8
2.2.	Läs-/skrivåtkomst till Modbus-slavobjekt	8
2.1.1.	Läsa hållregister, FC=03	9
2.1.2.	Skriva enstaka register, FC=06	9
2.1.3.	Skrivning till flera register, FC=16	9
2.3.	Mapping av datatyper.....	11

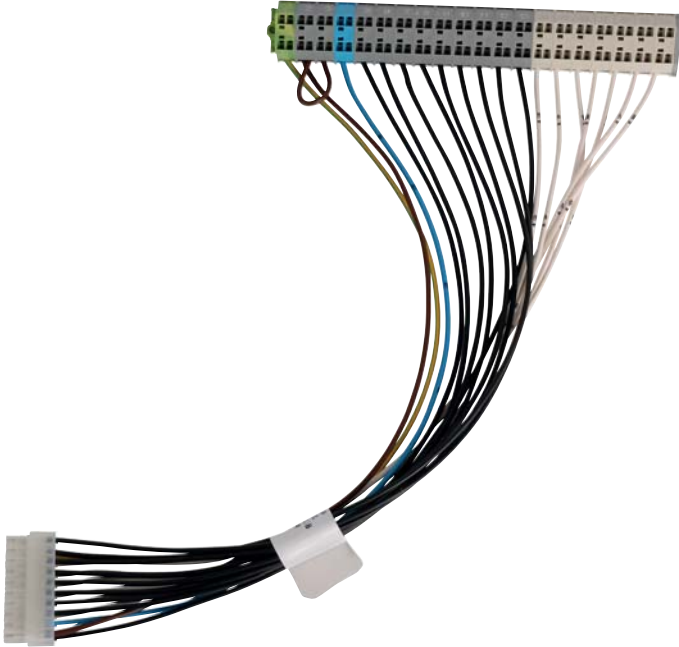
EN Contents

1.	Connecting to the unit	12
1.1.	NORDIC series.....	12
1.2.	ECONORDIC series	12
2.	Communication parameters.....	12
2.1.	Mapping of application parameters.....	12
2.2.	Read/write access to Modbus slave object	12
2.1.1.	Read Holding Registers, FC=03.....	13
2.1.2.	Write Single Register, FC=06	13
2.1.3.	Write Multiple Register, FC=16	13
2.3.	Mapping of Data types.....	15

1. Koble til enheten

1.1. NORDIC-serien

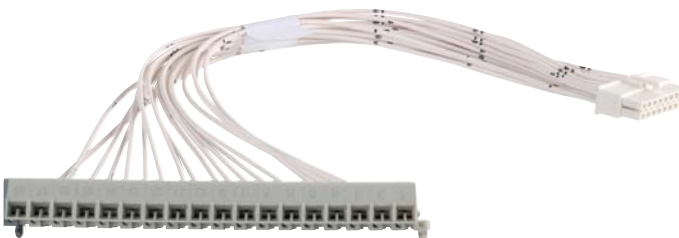
Bruk ledningssettet, art.nr. 116405, for å koble Modbus til produktet.



Terminalnr.	Signal
16	D+
17	D-
18	Ref (G0)

1.2. ECONORDIC-serien

Bruk ledningssettet, art.nr. 117050, for å koble Modbus produktet.



Terminalnr.	Signal
2	D+
3	D-
4	Ref (G0)

Lav stigehastighet 1/8 enhetbelastning sender/mottaker
Kabellengde: Maks. 30 m

2. Kommunikasjonsparametre

Parameter	Standardverdi	Område
Adresse	1	1-247
Baudrate	9600	9600 / 19200
Databits	8	-
Paritet	Even	None / Even / Odd
Stoppbits	1	1 / 2

2.1. Mapping av applikasjonsparametre

Generisk Modbusgrensesnitt for å gi tilgang til produktet ved kjøring med systemstatus 'I drift'

2.2. Lese/skrivetilgang til Modbus-slaveobjektet

Regulatoren støtter følgende funksjonskode (FC) fra Modbus-masteren:

Les inngangsregistre, FC=04

Forespørsel

Funksjonskode	1 byte	0x04
Startadresse	2 byte	0x0000 til 0xFFFF
Antall inngangsregistre	2 byte	0x0001 til 0x007D

Svar

Funksjonskode	1 byte	0x04
Antall byte	1 byte	2 x N*
Inngangsregistre	N* x 2 byte	

N* = Antall inngangsregistre

Feil

Feilkode	1 byte	0x84
Unntakskode	1 byte	01 eller 02 eller 03 or 04

Les inngangsregistre(FC 04)

- En blokk sammenhengende inngangsregistre leses på rad: Antall inngangsregistre N = 1...125
- Registerdata i svarmelding er pakket som to byte per register. Den første inneholder de høyeste bitene og den andre de laveste bitene i registeret.
- I forespørselsmeldingen ble de to byte i det første registeret (startadresse) sendt først.

2.1.1. Les holdingregistre, FC=03

Forespørsel

Funksjonskode	1 byte	0x03
Startadresse	2 byte	0x0000 til 0xFFFF
Antall inngangsregistre	2 byte	1 til 125 (0x7D)

Svar

Funksjonskode	1 byte	0x03
Antall byte	1 byte	2 x N*
Inngangsregistre	N* x 2 byte	

N* = Antall inngangsregistre

Feil

Feilkode	1 byte	0x83
Unntakskode	1 byte	01 eller 02 eller 03 eller 04

Les holdingregistre(FC 03)

- En blokk av sammenhengende inngangsregistre leses på rad: Antall holdingregistre N = 1...125
- Registerdata i svarmelding er pakket som to byte per register. Den første inneholder de høyeste bitene og den andre de laveste bitene i registeret.
- I forespørselsmeldingen ble de to byte i det første registeret (startadresse) sendt først.

2.1.2. Skriv til enkelt register, FC=06

Forespørsel

Funksjonskode	1 byte	0x06
Registeradresse	2 bytes	0x0000 til 0xFFFF
Registerverdi	2 bytes	0x0001 til 0xFFFF

Svar

Funksjonskode	1 byte	0x06
Registeradresse	2 bytes	0x0000 til 0xFFFF
Registerverdi	2 bytes	0x0001 til 0xFFFF

Feil

Feilkode	1 byte	0x86
Unntakskode	1 byte	01 eller 02 eller 03 eller 04

Skriv til enkelt register (FC 06)

2.1.3. Skriv til flere registre, FC=16

Forespørsel

Funksjonskode	1 byte	0x10
Startadresse	2 byte	0x0000 til 0xFFFF
Antall registre	2 byte	0x0001 til 0x007B
Antall byte	1 byte	2 x N*
Registerverdi	N* x 2 byte	verdi

N* = Antall inngangsregistre

Svar

Funksjonskode	1 byte	0x10
Startadresse	2 byte	0x0000 til 0xFFFF
Antall registre	2 byte	1 til 123 (0x7B)

Feil

Feilkode	1 byte	0x90
Unntakskode	1 byte	01 eller 02 eller 03 eller 04

Skriv til flere registre (FC 16)

- En blokk av sammenhengende registre skrives på rad
Antall registre N = 1...123
- De forespurte skrevne verdier spesifiseres i datafeltet for forespørsel. Data pakkes som to byte per register, dvs. antall byte := 2 x N
- I forespørselsmeldingen ble de to byte i det første registeret (startadresse) sendt først.

Skriveforespørselen inneholder

- Adressen til MODBUS-slaven
- Funksjonskode (FC)
- Startadressen
- Og avhengig av FC, ytterligere kontrollinformasjon for å definere nummerdata som skal skrives (N av elementer, antall byte)
- Data som skal skrives

Et positivt svar som inneholder:

- svarets FC = forespørselens FC
- svarets startadresse = forespørselens startadresse
- og avhengig av FC
- skrevne data: svardata skal leses tilbake fra BA-objektet for å indikere min. / maks. trunkering=> ikke enkelt ekko av forespørselen!
- eller informasjon for å indikere antall skrevne dataelement

Et unntakssvar, hvis serveren ikke kan behandle skriveforespørselen. Målet er å gi kunden relevant informasjon om feilen som ble oppdaget ved behandlingen.

- unntakets FC = forespørselens FC+ 0x80
- en unntakskode oppgis or å angi årsaken til feilen

Kode	Navn	Betydning
01	UGYLDIG FUNKSJON	<ul style="list-style-type: none"> • Funksjonskoden som mottas i forespørselen støttes ikke av serveren. • Eller serveren kan for tiden ikke behandle forespørselen, for eksempel fordi den ikke er konfigurert
02	UGYLDIG DATAADRESSE	<ul style="list-style-type: none"> • Registeradressen/startadressen som mottas i forespørselen er ikke en tillatt adresse for serveren. • Eller kombinasjonen startadresse og antall registre er ugyldig. f.eks. for en regulator med 100 registre, kaller PDU det første registeret 0, og det siste 99. Dersom det sendes en forespørsel med startadresse 96 og med antall registre 4, så vil denne forespørselen kjøre (i det minste hva adresse gjelder) på registrene 96, 97, 98 og 99. Dersom det sendes en forespørsel med startadresse 96 og med antall registre 5, så vil forespørselen mislykkes og du får unntakskoden 0x02 "Ugyldig dataadresse", siden den forsøker å kjøre på registrene 96, 97, 98, 99 og 100, og det ikke finnes noe register med adresse 100. • Ved masselesing/skriving dersom det er en adresseluke mellom dataelementene i det gitte adresseområdet.
03	UGYLDIG DATAVERDI	<ul style="list-style-type: none"> • En datatype som inngår skrive-dataen er ikke en tillatt verdi for serveren. Angir en feil i strukturen i det gjenværende av en kompleks forespørsel, som at implisert lengde er feil (datatype samsvarer ikke). • Men, det betyr IKKE uttrykkelig at et dataelement som sendes for lagring i et register har en verdi utenfor tillatt min/maks-område, siden Modbus-protokollen ikke kjenner til betydningen til en gitt verdi i et gitt register.
04	FEILMELDING SERVERENHET	<ul style="list-style-type: none"> • En uopprettelig feil oppstod da serveren forsøkte å utføre den forespurte handlingen, f.eks. <ul style="list-style-type: none"> - Master prøver å skrive skrivebeskyttet element - tilgang til BA-objekttype som ikke støttes, mapping er ikke mulig - Master prøver å skrive datafragmenter, ugyldig data - osv...

2.3. Mapping av datatyper

1. "Binary value" med ett register:

Hver aktuell binær verdi vil oppta ett register

Datatype	Byterekkefølge
Binary	00 aa, med aa = 0x00 eller 0x01

2. "Multistate value" med ett register:

Hver aktuell flerstatusverdi vil oppta ett register

Datatype	Byterekkefølge
Multistate (8 bit)	00 aa, med 1:1 BACnet-koding
Multistate (16 bit)	bb aa, med 1:1 BACnet-koding

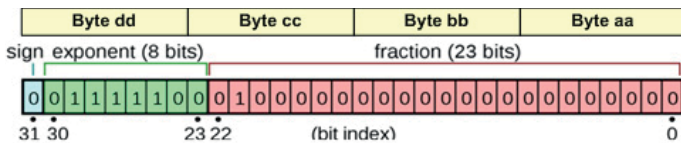
3. "Unsigned integer" med to registre:

Datatype	Byterekkefølge
Uint (32 bit)	dd cc bb aa
	dd cc Registeradresse x
	bb aa Registeradresse x+1

4. "Real value" med to registre:

Datatype	Byterekkefølge
Real (32 bit)	dd cc bb aa
	dd cc Registeradresse x
	bb aa Registeradresse x+1

med:

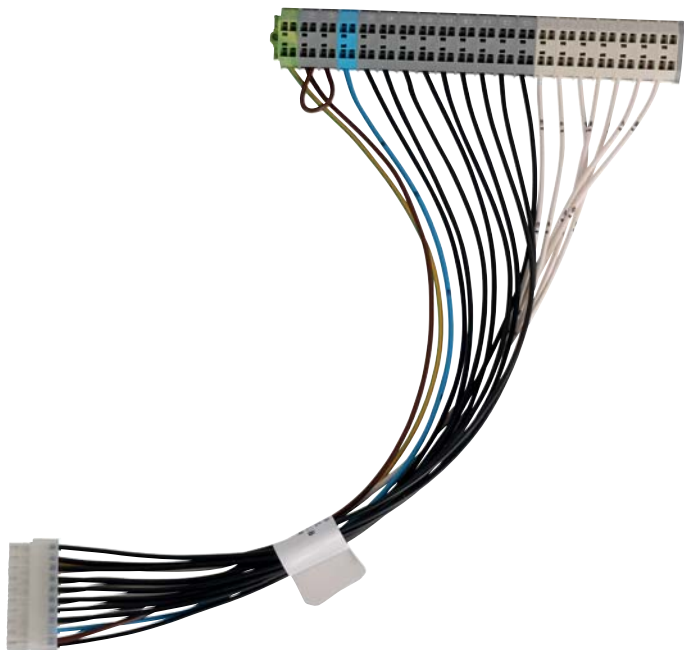


IEEE-754 32-biters koding av "Float"

1. Ansluta till enheten

1.1. NORDIC-serien

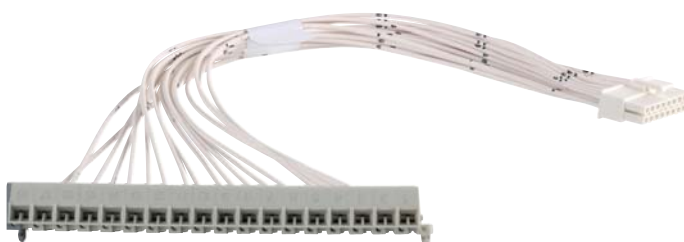
Använd kablage, artikelnr 116405 för att ansluta till produkten.



Plintnr.	Signal
16	D+
17	D-
18	Ref (G0)

1.2. ECONORDIC-serien

Använd kablage, artikelnr 117050, för att ansluta till produkten.



Plintnr.	Signal
2	D+
3	D-
4	Ref (G0)

Låg stighastighet 1/8 enhetslast sändare/mottagare
Kabellängd: Max 30 m.

2. Kommunikationsparametrar

Parameter	Standardvärde	Område
Adress	1	1-247
Överföringshastighet	9600	9600 / 19200
Databitar	8	-
Paritet	Even	Ingen/Jämn/Udda
Stoppbitar	1	1 / 2

2.1. Mappning av programparametrar

Generiskt Modbusgränssnitt för åtkomst till produkten vid körning med systemstatus "I drift"

2.2. Läs-/skrivåtkomst till Modbus-slavobjekt

Styrenheten stöder sådan funktionskod (FC) från Modbus-master:

Läsa indataregister, FC=04

Begäran

Funktionskod	1 byte	0x04
Startadress	2 byte	0x0000 till 0xFFFF
Antal indataregister	2 byte	0x0001 till 0x007D

Svar

Funktionskod	1 byte	0x04
Antallbyte	1 byte	2 x N*
Indataregister	N* x 2 byte	

N* = antal indataregister

Fel

Felkod	1 byte	0x84
Undantagskod	1 byte	01 eller 02 eller 03 or 04

Läsa indataregister (FC 04)

- Ett block av angränsande indataregister läses i ett steg: Indataregisters kvantitet N = 1...125
- Registerdata i svarsmeddelandet är packade som två byte per register. Den första innehåller de högsta bitarna och den andra innehåller de lägsta bitarna i registret.
- I meddelandet med begäran skickas dessa två byte i det första registret (startadress) först.

2.1.1. Läs hållregister, FC=03

Begäran

Funktionskod	1 byte	0x03
Startadress	2 byte	0x0000 till 0xFFFF
Antal register	2 byte	1 till 125 (0x7D)

Svar

Funktionskod	1 byte	0x03
Antal byte	1 byte	2 x N*
Registervärde	N* x 2 byte	

N* = antal indataregister

Fel

Felkod	1 byte	0x83
Undantagskod	1 byte	01 eller 02 eller 03 eller 04

Läs hållregister (FC 03)

- Ett block av angränsande hållregister läses i ett steg: Hållregisters kvantitet N = 1...125
- Registerdata i svarsmeddelandet är packade som två byte per register. Den första innehåller de högsta bitarna och den andra innehåller de lägsta bitarna i registret.
- I meddelandet med begäran skickas dessa två byte i det första registret (startadress) först.

2.1.2. Skriva enstaka register, FC=06

Begäran

Funktionskod	1 byte	0x06
Registeradress	2 byte	0x0000 till 0xFFFF
Registervärde	2 byte	0x0001 till 0xFFFF

Svar

Funktionskod	1 byte	0x06
Registeradress	2 byte	0x0000 till 0xFFFF
Registervärde	2 byte	0x0001 till 0xFFFF

Fel

Felkod	1 byte	0x86
Undantagskod	1 byte	01 eller 02 eller 03 eller 04

Skriva enstaka register (FC 06)

2.1.3. Skrivning till flera register, FC=16

Begäran

Funktionskod	1 byte	0x10
Startadress	2 byte	0x0000 till 0xFFFF
Antal register	2 byte	0x0001 till 0x007B
Antal byte	1 byte	2 x N*
Registervärde	N* x 2 byte	värde

N* = antal indataregister

Svar

Funktionskod	1 byte	0x10
Startadress	2 byte	0x0000 till 0xFFFF
Antal register	2 byte	1 till 123 (0x7B)

Fel

Felkod	1 byte	0x90
Undantagskod	1 byte	01 eller 02 eller 03 eller 04

Skriva till flera register (FC 16)

- Ett block av angränsande register skrivs i ett steg antal register N = 1...123
- De skrivna värdena specificeras i datafältet för begäran. Data packas som två byte per register, dvs. antal byte = 2 x N
- I meddelandet med begäran skickas dessa två byte i det första registret (startadress) först.

Skrivbegäran innehåller

- MODBUS-slave adressen
- Funktionskoden (FC)
- Startadressen
- Beroende på FC ytterligare kontrollinformation för att definiera nummerdata som ska skrivas (N av element, antal byte)
- Data som ska skrivas

Ett positivt svar som innehåller:

- svarets FC = begärens FC
- svarets startadress = begärens startadress
- och beroende på FC
- skrivna data: svarsdata ska läsas tillbaka från BA-objektet för att ange t.ex. min./max. trunkering => inget enkelt eko av begäran!
- eller information för att ange antalet skrivna dataelement

Ett undantagssvar om servern inte kan behandla skrivbegäran. Målet är att ge kunden relevant information om felet som upptäcks under behandlingen:

- undantagets FC = begärens FC + 0x80
- en undantagskod tillhandahålls för att ange orsaken till felet

Kod	Namn	Betydelse
01	OGILTIG FUNKTION	<ul style="list-style-type: none"> • Den funktionskod som mottagits i begäran stöds inte av servern. • Eller servern kan för närvarande inte behandla begäran, till exempel eftersom den inte är konfigurerad.
02	OGILTIG DATAADDRESS	<ul style="list-style-type: none"> • Den registeradress/startadress som mottas i begäran är inte en tillåten adress för servern. • Eller kombinationen av startadress och registers kvantitet är ogiltig. T.ex. för en styrenhet med 100 register kallar PDU det första registret 0 och det sista 99. Om en begäran skickas med en startadress på 96 och antalet register = 4, så kommer denna begäran att arbeta (åtminstone vad gäller adress) på register 96, 97, 98, 99. Om en begäran skickas med en startadress på 96 och antalet register = 5, så kommer denna begäran att misslyckas med undantagskod 0x02 "Ogiltig dataadress" eftersom den försöker arbeta med register 96, 97, 98, 99 och 100, och det finns inte något register med adress 100. • Vid massläsning/skrivning finns det en adresslucka mellan dataobjekt i det givna adressintervallet.
03	OGILTIGT DATAVÄRDE	<ul style="list-style-type: none"> • En datatyp som ingår i skrivdata är inte ett tillåtet värde för servern. Detta anger ett fel i strukturen på återstoden av en komplex begäran, så att den underförstådda datalängden är felaktig (datatyp som inte matchar). • Det betyder dock INTE uttryckligen att ett dataobjekt som skickas för lagring i ett register har ett värde utanför det tillåtna min./max. intervallet, eftersom MODBUS-protokollet inte är medvetet om betydelsen av ett visst värde i ett visst register.
04	SERVERENHETSFEJ	<ul style="list-style-type: none"> • Ett oåterkalleligt fel inträffade när servern försökte vidta de begärda åtgärderna, t.ex. <ul style="list-style-type: none"> - Master försöker skriva skrivskyddat dataobjekt - Åtkomst till BA-objektstyp som inte stöds, mappning är inte möjlig - Master försöker skriva datafragment, ogiltiga data - etc.

2.3. Mappning av datatyper

1. "Binary value" med ett register:

Varje aktuellt BA-binärvärde upptar ett register

Datatype	Byteordning
Binary	00 aa, med aa = 0x00 eller 0x01

2. "Multistate value" med ett register:

Varje aktuellt BA-flerlägesvärde upptar ett register

Datatype	Byteordning
Multistate (8 bit)	00 aa, med 1:1 BACnet-kodning
Multistate (16 bit)	bb aa, med 1:1 BACnet-kodning

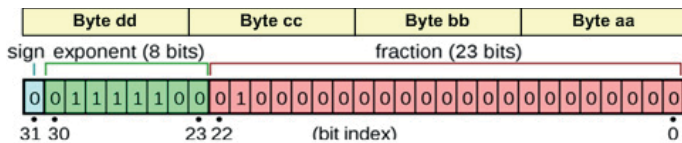
3. "Unsigned integer" med två register:

Datatype	Byteordning
Uint (32 bit)	dd cc bb aa
	dd cc Registeradress x
	bb aa Registeradress x+1

4. "Real value" med två register:

Datatype	Byteordning
Real (32 bit)	dd cc bb aa
	dd cc Registeradress x
	bb aa Registeradress x+1

med:

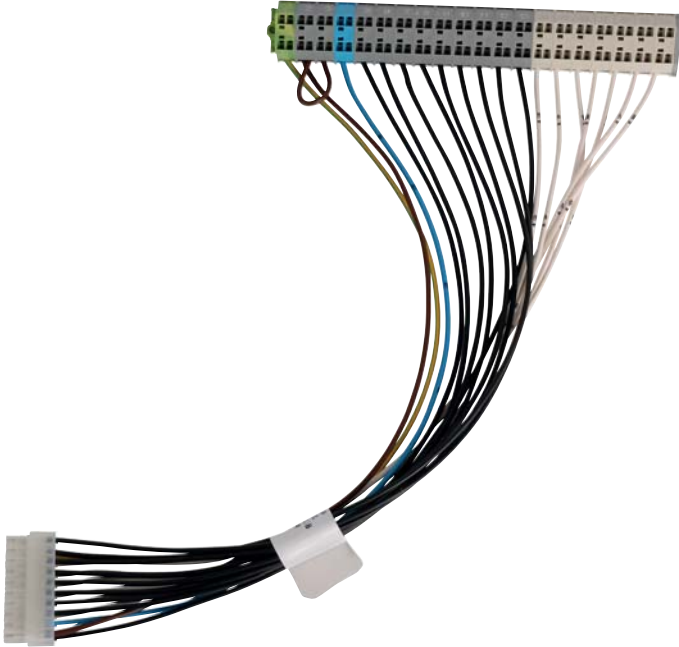


IEEE-754 32-bitars kodning av "Float"

1. Connecting to the unit

1.1. NORDIC series

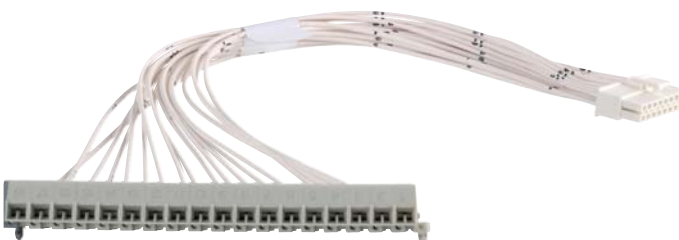
Use cable harness, part no. 116405 in order to connect Modbus slave to the product.



Terminal no.	Signal
16	D+
17	D-
18	Ref (G0)

1.2. ECONORDIC series

Use cable harness, part no. 117050 in order to connect Modbus slave to the product.



Terminal no.	Signal
2	D+
3	D-
4	Ref (G0)

Low slew rate 1/8 unit load transceiver
Cable length: 30 m max

2. Communication parameters

Parameter	Default value	Range
Address	1	1-247
Baudrate	9600	9600 / 19200
Databits	8	-
Parity	Even	None / Even / Odd
Stop bits	1	1 / 2

2.1. Mapping of application parameters

Generic MODBUS adapter with access to the product at runtime, with System-Status 'Operational'

2.2. Read/write access to Modbus slave object

The controller supports such function code (FC) from Modbus Master:

Read Input Registers, FC=04

Request

Function code	1 byte	0x04
Starting address	2 bytes	0x0000 to 0xFFFF
Quantity of Input Registers	2 bytes	0x0001 to 0x007D

Response

Function code	1 byte	0x04
Byte count	1 byte	2 x N*
Input Registers	N* x 2 bytes	

N* = Quantity of Input Registers

Error

Error code	1 byte	0x84
Exception code	1 byte	01 or 02 or 03 or 04

Read Input Registers (FC 04)

- A block of contiguous Input Registers is read in one step: Quantity of Input Registers N = 1...125
- The Register data in the response message are packed as two bytes per Register. The first byte contains the high order bits and the second contains the low order bits of the Register.
- In the request message the two bytes of the first Register (Starting Addr) are sent first.

2.1.1. Read Holding Registers, FC=03

Request

Function code	1 byte	0x03
Starting address	2 bytes	0x0000 to 0xFFFF
Quantity of Registers	2 bytes	1 to 125 (0x7D)

Response

Function code	1 byte	0x03
Byte count	1 byte	2 x N*
Register value	N* x 2 bytes	

N* = Quantity of Input Registers

Error

Error code	1 byte	0x83
Exception code	1 byte	01 or 02 or 03 or 04

Read Holding Registers (FC 03)

- A block of contiguous Holding Registers is read in one step: Quantity of Holding Registers N = 1...125
- The Register data in the response message are packed as two bytes per Register. The first byte contains the high order bits and the second contains the low order bits of the Register.
- In the request message the two bytes of the first Register (Starting Addr) are sent first.

2.1.2. Write Single Register, FC=06

Request

Function code	1 byte	0x06
Register address	2 bytes	0x0000 to 0xFFFF
Register value	2 bytes	0x0001 to 0xFFFF

Response

Function code	1 byte	0x06
Register address	2 bytes	0x0000 to 0xFFFF
Register value	2 bytes	0x0001 to 0xFFFF

Error

Error code	1 byte	0x86
Exception code	1 byte	01 or 02 or 03 or 04

Write Single Register (FC 06)

2.1.3. Write Multiple Register, FC=16

Request

Function code	1 byte	0x10
Starting address	2 bytes	0x0000 to 0xFFFF
Quantity of registers	2 bytes	0x0001 to 0x007B
Byte count	1 byte	2 x N*
Registers value	N* x 2 bytes	value

N* = Quantity of Input Registers

Response

Function code	1 byte	0x10
Starting address	2 bytes	0x0000 to 0xFFFF
Quantity of registers	2 bytes	1 to 123 (0x7B)

Error

Error code	1 byte	0x90
Exception code	1 byte	01 or 02 or 03 or 04

Write Multiple Registers (FC 16)

- A block of contiguous Registers is written in one step
- Quantity of Registers N = 1...123
- The requested written values are specified in the request data field. Data is packed as two bytes per register, i.e. Byte Count := 2 x N
- In the request message the two bytes of the first Register (Starting Addr) are sent first.

The Write-request contains the

- Address of the MODBUS Slave
- Function Code (FC)
- Start Address
- And depending on the FC additional control information to define the number data to be written (N Of Elements, ByteCount)
- The data to be written

A **positive response** containing:

- the response FC = the request FC
- the response Start Address = the request Start Address
- and depending on the FC
- the written data: response data shall be read back from the BA-Object to indicate e.g. min. / max. truncation => no simple echo of the request!
- or information to indicate the number written of data elements

An **exception response**, if the server is not able to process the write request. The objective is to provide to the client relevant information concerning the error detected during the processing:

- the exception FC = the request FC + 0x80
- an exception code is provided to indicate the reason of the error

Code	Name	Meaning
01	ILLEGAL FUNCTION	<ul style="list-style-type: none"> • The Function Code received in the request is not supported by the Server. • Or the Server is currently not able to process the request, for example because it is un-configured
02	ILLEGAL DATA ADDRESS	<ul style="list-style-type: none"> • The Register Address/Starting Address received in the request is not an allowable address for the Server. • Or the combination of Starting Address and Quantity of Registers is invalid. e.g. for a controller with 100 Registers, the PDU addresses the first Register as 0, and the last one as 99. If a request is submitted with a Starting Address of 96 and a Quantity of Registers of 4, then this request will successfully operate (address-wise at least) on registers 96, 97, 98, 99. If a request is submitted with a Starting Address of 96 and a Quantity of Registers of 5, then this request will fail with Exception Code 0x02 "Illegal Data Address" since it attempts to operate on registers 96, 97, 98, 99 and 100, and there is no Register with address 100. • In case of bulk reading/writing if there is address gap between data items in the given address range
03	ILLEGAL DATA VALUE	<ul style="list-style-type: none"> • A data type contained in the write data is not an allowable value for the Server. This indicates a fault in the structure of the remainder of a complex request, such as that the implied data length is incorrect (data type mismatch). • However, it specifically does NOT mean that a data item submitted for storage in a Register has a value outside the allowed min / max range, since the MODBUS protocol is unaware of the significance of any particular value of any particular Register.
04	SERVER DEVICE FAILURE	<ul style="list-style-type: none"> • An unrecoverable error occurred while the server was attempting to perform the requested action, e.g. <ul style="list-style-type: none"> - Master tries to write read-only data item - access to unsupported BA-Object type, mapping is not possible - Master tries to write data fragments, illegal data - etc...

2.3. Mapping of Data types

1. Binary value encoding with one Register:

Each binary BA-Present-Value will occupy one Register

Data type	Byte order
Binary	00 aa, with aa = 0x00 or 0x01

2. Multistate value encoding with one Register:

Each multistate BA-Present-Value will occupy one Register

Data type	Byte order
Multistate (8 bit)	00 aa, with 1:1 BACnet encoding
Multistate (16 bit)	bb aa, with 1:1 BACnet encoding

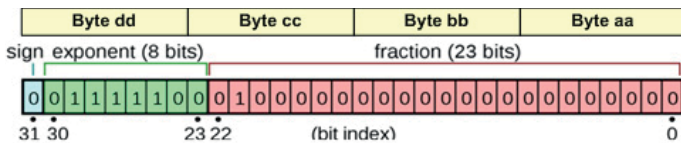
3. Unsigned integer value encoding with two Registers:

Data type	Byte order
Uint (32 bit)	dd cc bb aa
	dd cc Register Address x
	bb aa Register Address x+1

4. Real value encoding with two Registers:

Data type	Byte order
Real (32 bit)	dd cc bb aa
	dd cc Register Address x
	bb aa Register Address x+1

with:



IEEE-754 32-bit float encoding



Flexit AS, Televeien 15, N-1870 Ørje
www.flexit.no