

**TABLE OF CONTENTS**

Introduction .....	1
Ethernet Connections .....	2
Hardware .....	2
Software .....	2
Modbus Controllers .....	4
Protocol .....	4
Message Structure .....	4
Device Address Field .....	4
Function Code Field .....	5
Data Block Field .....	5
Error Check Field.....	5
Serial Transmission Details.....	5
Message Framing and Timing Considerations.....	5
Error Handling and Exception Responses .....	6
Detailed Message Query and Response .....	6
Data Formats.....	9
CRC Error Check.....	10
DECS-2100 Modbus Register Space.....	11
Data Map .....	11
Data Software Blocks .....	12
Inputs/Control.....	12
Outputs/Data .....	12
Help Files.....	15
Single-Channel versus Multi-Channel Systems .....	16
Making and Saving Changes.....	17

**INTRODUCTION**

Ethernet communication reduces the amount of hardwiring needed for alarm and status signals while increasing the amount of information available. This information may be used to interface with external distributed control systems such as SCADA, etc.

Each DECS-2100 system is equipped with a default HMI on the logic cubicle door. The default (local HMI uses one of two available Ethernet ports on the ECM-2 module. Each DECS-2100 system also contains one additional Ethernet port that may be used for a remote HMI, as a connection to another plant control system, or to an Ethernet switch so that multiple devices can gather data.

Basler Electric recommends that the DECS-2100 be placed on a dedicated network. Placing it on an “open network” may cause unwanted communication errors.

Publication <b>9411300996</b>	Revision <b>A</b>	<b><i>Instructions</i></b>	Date <b>08/14</b>	Copyright <b>2014</b>
----------------------------------	----------------------	----------------------------	----------------------	--------------------------

## ETHERNET CONNECTIONS

### Hardware

Each ECM-2 module is equipped with two Ethernet ports which use an RJ-45 connector. Port 2 is typically used for communication with the local HMI on the door of the DECS-2100 enclosure. Each ECM-2 (on per channel) would be linked from Ethernet port 2 to an Ethernet switch also linked to the local HMI.

The ECM-2 supports 10Base-T and 100Base-T wiring standards.

The user could connect their plant SCADA system or other Ethernet master to the Ethernet switch providing the signal to the HMI. However, the same IP addresses would need to be used for the ECM2 modules. Alternately, the user could connect the plant SCADA to Ethernet port 1—either directly or through a separate Ethernet switch. Due to communication issues, Basler Electric prefers the use of Ethernet port 1 for communication with external plant devices and the use of Ethernet port 2 for the local HMI only.

For serial Modbus™ communications, three RS-485 ports are located to the left of the two Ethernet ports shown in Figure 1.

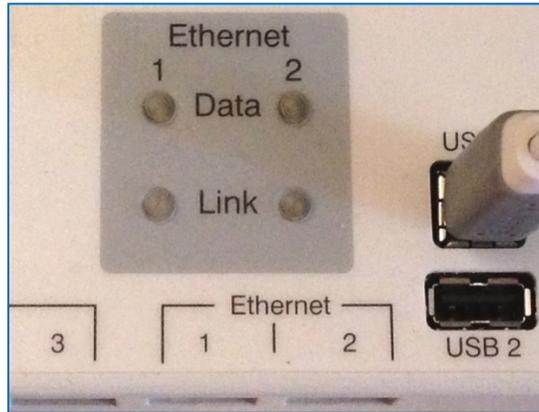


Figure 1. ECM-2 Ethernet Ports

### Software

BESTCOMS™ Pro software provides the ability to set Ethernet addressing on the two ECM-2 Ethernet ports. Each port gathers data from the same data maps so the information is identical.

Once the user is connected to the controller via BESTCOMS™ Pro, the Ethernet ports can be configured in the Settings Explorer (General Settings, Controller Settings, Ethernet Settings). Ethernet 1 and Ethernet 2 are set independently. Once the desired port is selected, as in the right side of Figure 2, the addressing may be entered as illustrated in Figure 3.

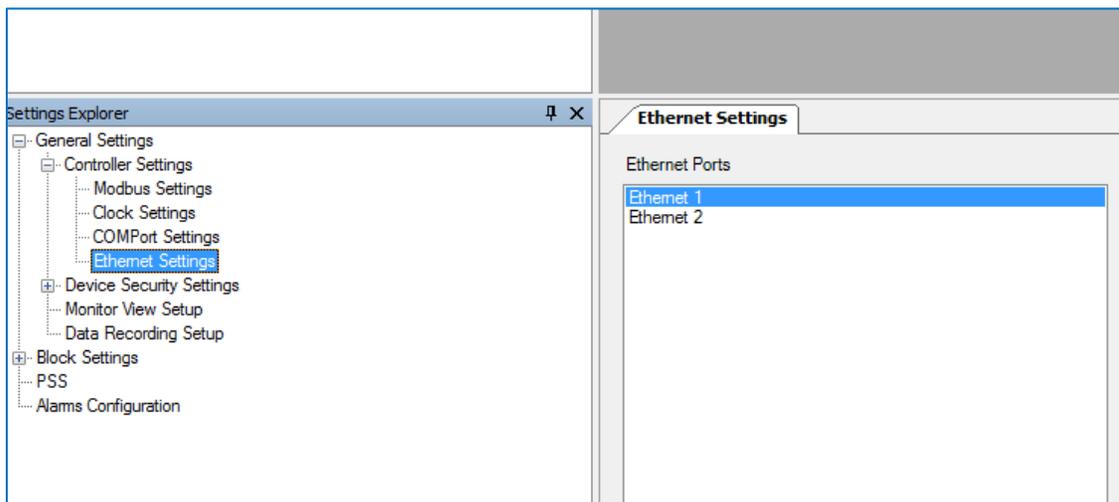
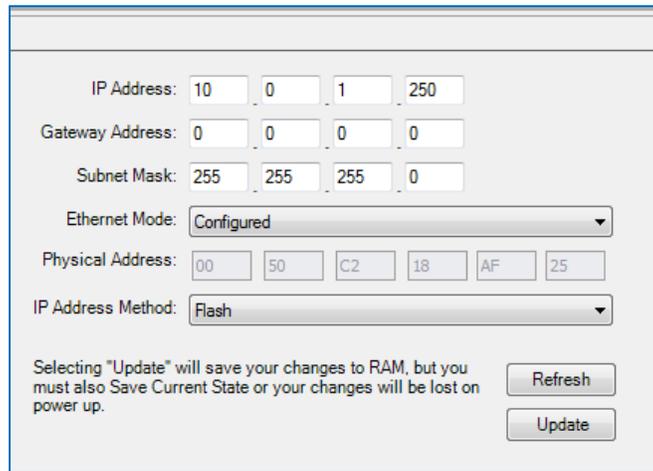


Figure 2. Ethernet Port Selection



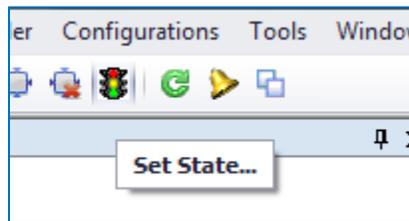
IP Address: 10 . 0 . 1 . 250  
 Gateway Address: 0 . 0 . 0 . 0  
 Subnet Mask: 255 . 255 . 255 . 0  
 Ethernet Mode: Configured  
 Physical Address: 00 . 50 . C2 . 18 . AF . 25  
 IP Address Method: Flash

Selecting "Update" will save your changes to RAM, but you must also Save Current State or your changes will be lost on power up.

Refresh  
Update

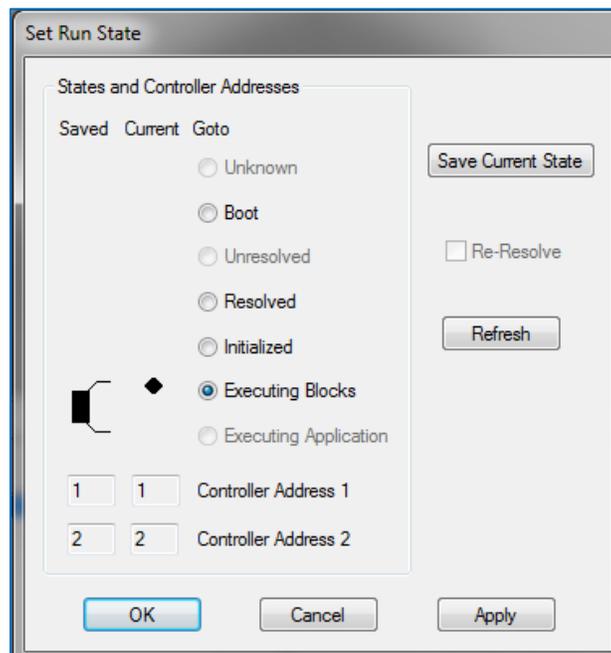
**Figure 3. Ethernet Port Addressing**

After entering an address, click the Update button. This enters the address into RAM of the ECM-2 processor. To permanently save or update the address into flash memory, use the traffic light icon on the BESTCOMS™ Pro menu bar (Figure 4).



**Figure 4. Permanently Saving the Ethernet Address**

Clicking the traffic light icon opens the Set Run State window (Figure 5). Click the Save Current State button to save the Ethernet address into flash memory. Then, click the OK button to close the Set Run State window. This process is safe to perform while the unit is online.



Set Run State

States and Controller Addresses

Saved	Current	Goto
<input type="radio"/>	<input type="radio"/>	Unknown
<input type="radio"/>	<input type="radio"/>	Boot
<input type="radio"/>	<input type="radio"/>	Unresolved
<input type="radio"/>	<input type="radio"/>	Resolved
<input type="radio"/>	<input type="radio"/>	Initialized
<input type="radio"/>	<input checked="" type="radio"/>	Executing Blocks
<input type="radio"/>	<input type="radio"/>	Executing Application

Controller Address 1: 1 . 1  
 Controller Address 2: 2 . 2

Save Current State  
 Re-Resolve  
 Refresh

OK Cancel Apply

**Figure 5. Set Run State Window**

For a system with multiple channels, this process is repeated for each channel with each channel having a unique IP address.

### Modbus Controllers

It is important to note that each ECM-2 (IP address) contains two Modbus controllers. Data maps in the odd-numbered controllers are configurable and may cover most of the DECS-2100 system. Data maps in the even-numbered controllers are not configurable and cover only bridge and bridge link data. In order to poll data from the even-numbered controllers, the Ethernet system used must be able to address two Modbus controllers on the same IP address. Modbus controller settings are illustrated in Figure 6. Typical channel and controller numbers are shown in Table 1.

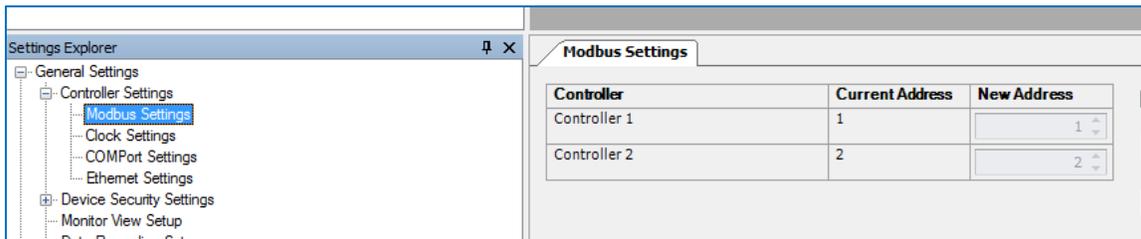


Figure 6. Modbus Controller Settings

Table 1. Typical Modbus Controller and Channel Assignments

Channel	Controller 1	Controller 2
Main	1	3
Redundant	3	4
Supervisory	5	6

Once the addresses are assigned and saved, a ping test should be successful.

### **Protocol**

Modbus communication employs a master-slave technique in which only the master can initiate a transaction. This transaction is called a query. When appropriate, a slave (DECS-2100) responds to a query. When a Modbus master communicates with a slave, information is provided or requested by the master.

When a slave receives a query, the slave responds by supplying the requested data to the master or performing the requested action. A slave device never initiates communication and will always respond to a query unless certain error conditions exist. The DECS-2100 communicates over Modbus only as a slave device.

A master can only query slave devices individually. If a query requests an action that the slave is unable to execute, the slave responds with a message containing an exception response code defining the error detected.

### Message Structure

Master-initiated queries and DECS-2100 (slave) responses share the same message structure. Each message is comprised of four message fields:

- Device address (1 byte)
- Function code (1 byte)
- Data block (n bytes)
- Error check field (2 bytes)

### Device Address Field

The device address field contains the unique Modbus address of the slave being queried. The addressed slave repeats the address in the device address field of the response message. This field is one byte.

Modbus protocol limits a device address from 1 to 247. The address is user-selectable at installation and can be altered during real-time operation.

### Function Code Field

The function code field in the query message defines the action to be taken by the addressed slave. This field is echoed in the response message and is altered by setting the most significant bit (MSB) of the field to 1 if the response is in error. This field is one byte.

The DECS-2100 maps all registers into the Modcon 984 holding register address space (4XXXX) and supports the following function codes:

- Read coils (function code 1)
- Read output registers (function code 3)
- Write/preset single coil (function code 5)
- Preset multiple registers (function code 16) and
- Loopback diagnostic test (function code 8) with diagnostic sub-functions:
  - Return query data (diagnostic code 0)
  - Restart comm. Option (diagnostic code 1) and
  - Force slave to listen only mode (LOM, diagnostic code 4)

DECS-2100 Modbus performs all of the above functions when a Modbus message contains a unique address over the range of 1 to 247. The DECS-400 also recognizes a broadcast (group) address of 0. Only functions 16 and 8 are recognized as valid for broadcast. The DECS-400 does not send a response message for a broadcast query.

In listen-only mode (LOM), received data is monitored but no responses are transmitted. The only query recognized and processed while in LOM is a maintenance restart command (function code 8, diagnostic code 1).

### Data Block Field

The query data block contains additional information needed by the slave to perform the requested function. The response data block contains data collected by the slave for the queried function. An error response will substitute an exception response code for the data block. The length of this field varies with each query. See *Register Definitions* for interpretation of register data.

### Error Check Field

The error check field provides a method for the slave to validate the integrity of the query message content and allows the master to confirm the validity of response message content. This field is 2 bytes.

### Serial Transmission Details

Communication settings for the DECS-2100's three RS-485 ports are listed in Table 2. The baud rate, data size, parity, stop bits and Modbus slave address apply only to the RS-485 interface and are not used when the Modbus/TCP mode is enabled.

**Table 2. RS-485 Port Settings**

Setting	Programmable	Default Value	Value Range
Baud	Yes	Port 1: 115,200 Port 2: 19,200 Port 3: 19,200	9600, 19200, 38400, 57600, 115200
Data size (in bits)	No	8	7, 8
Parity	Yes	None	N (none), O (odd), E (even)
Stop bits	Yes	1	1, 2
Modbus Slave Address	Yes	247	0 for broadcast, 1 to 247 for slave

### Message Framing and Timing Considerations

When receiving a message, the DECS-2100 requires an inter-byte latency of 3.5 character times before considering the message complete.

Once a valid query is received, the DECS-2100 waits for a period as specified in the Modbus Response Delay Time Register (48108) before responding. This register contains a value from 0 to 200 milliseconds. The default value is 10 milliseconds. The remote delay time parameter may be set to 0 in order to minimize response latency.

Table 3 lists the response message transmission times and 3.5 character times for the maximum response message length (225 characters) to a read query for 125 point and various baud rates.

**Table 3. Response Message Transmission Times**

Baud	1 Character Time	3.5 Characters Time	Maximum Read Register Response Message (255 Characters) Transmission Time
9600	1.0417 ms	3.645 ms	265.6335 ms
19200	0.52083 ms	1.823 ms	132.812 ms
38400	0.26042 ms	0.9115 ms	66.406 ms
57600	0.1736 ms	0.6076 ms	44.271 ms
115200	0.0868 ms	0.3038 ms	22.135 ms

***Error Handling and Exception Responses***

A received query is ignored when it contains a nonexistent device address, a framing error, or CRC error. In these cases no response is transmitted. Queries addressed to a DECS-2100 with an unsupported function code, unsupported register reference, or illegal data block values results in an error response message containing an exception response code.

Each error response message consists of a slave (DECS-2100) address, function code with the high-order bit set, error code, and error check (CRC) field.

Supported DECS-2100 exception response codes are listed in Table 4.

**Table 4. DECS-2100 Exception Response Error Codes**

Code	Name	Description
01	Illegal Function	The query function/sub-function code is unsupported; query read of more than 125 registers; query “preset multiple registers” of more than 100 registers
02	Illegal Data Address	A register referenced in the data block does not support queried read/write. For function codes 3 and 16: <ol style="list-style-type: none"> <li>Starting register address is mapped to DECS-300 Modbus address space but is not referenced to the highest order 16 bits of the assigned application data (see <i>Data Formats</i>).</li> <li>The number of registers is too small to hold entire value of all data (variables) assigned to those registers (see <i>Data Formats</i>).</li> </ol>
03	Illegal Data Value	A preset register data block contains an incorrect number of bytes or one or more data values out of range.

***Detailed Message Query and Response***

The following paragraphs describe supported DECS-2100 message queries and responses.

**Read Holding Registers**

**Query**

This query message requests a register or block of registers to be read. The data block contains the starting register address and the quantity of registers to be read. A register address of N will read holding register N+1. Table 5 illustrates the holding register query format.

**Table 5. Holding Register Query Format**

Device address	Function code = 03	Starting address high	Starting address low	Number of registers high	Number of registers low	CRC low	CRC high
----------------	--------------------	-----------------------	----------------------	--------------------------	-------------------------	---------	----------

The number of registers cannot exceed 125 without causing an error response with the exception code for an illegal function.

**Response**

The response message contains the data queried. The data block contains the block length in bytes followed by the data for each requested register. For each requested register, there is one Data Hi and one Data Lo. Attempting to read an unused register or a register which does not support a read returns a zero (0). If the query is a broadcast (device address = 0), no response message is returned.

Maximum response message length obtained for query of 125 registers is  $5 + (125 \times 2) = 255$  bytes. Table 6 illustrates the holding register response format.

**Table 6. Holding Register Response Format**

Device address	Function code = 03	Byte count	Data high	Data low	For each requested register	Data high	Data low	CRC low	CRC high
		250 max.	First queried register high	First queried register low	Data high and data low	Last queried register high	Last queried register low		

**Preset Multiple Registers**

A preset, multiple-registers query could address multiple registers in one slave or multiple slaves. If the query is broadcast (device address = 0), no response is required.

**Query**

A preset multiple registers query requests a register or block of registers to be written. The data block contains the starting address and the quantity of registers to be written, followed by the data block byte count and data. The DECS-2100 performs the write when the device address matches the DECS-400 remote address or when the device address is 0. A device address is 0 for broadcast query.

A register address of N will write Holding Register N+1.

All Modbus generic data formats can be loaded by this function (see *Data Format*).

No data will be written if any of the following exceptions occur:

- Queries to write to read-only or unsupported registers result in an error response with an exception code of Illegal Data Address
- Queries attempting to write more than 100 registers cause an error response with an exception code of Illegal Function
- An incorrect byte count results in an error response with an exception code of Illegal Function
- A query to write an illegal value (out of range) to a register results in an error response with an exception code of Illegal Data Value
- Query Starting Register address is mapped to DECS-2100 Modbus address space, but is not referenced to the lower order 16 bits of the assigned application data. See *Data Formats*.
- The number of query registers is too small to hold entire value of all data (variables) assigned to those registers. See *Data Formats*.

The query message format is:

Device Address  
 Function Code = 10 (hex)  
 Starting Address High  
 Starting Address Low  
 Number of Registers High (total number of registers to be loaded)  
 Number of Registers Low

Byte Count (total number of registers to be loaded times 2)  
 Data High  
 Data Low  
 Data High  
 Data Low  
 CRC Error Check (Lo, Hi)

Note: maximum length of Preset Multiple Registers Query is  $9+(100 \times 2)=209$  bytes.

**Response**

The response message echoes the starting address and the number of registers. There is no response message when the query is a broadcast (device address of 0). The multiple register response format is shown in Table 7.

**Table 7. Multiple Register Response Format**

Device address	Function code = 10 (hex)	Starting address high	Starting address low	Number of registers high	Number of registers low	CRC low	CRC high
----------------	--------------------------	-----------------------	----------------------	--------------------------	-------------------------	---------	----------

**Preset Single Register (Write Single Holding Register)**

A Preset Single Register query message requests a single register to be written. The DECS-2100 performs the write when the device address matches the DECS-2100 remote address.

**Query**

Device Address  
 Function Code = 06 (hex)  
 Address Hi  
 Address Lo  
 Data Hi  
 Data Lo  
 CRC Hi error check  
 CRC Lo error check

**Response**

During a normal response, the response message echoes the query message after the register has been altered.

In a response to an error, data will cease to be written if any of the following exceptions occur:

- Queries to write to the read-only registers result in an error response with an exception code of Illegal Data Address
- A query to write an illegal value (out of range) to a register results in an error response with an exception code of Illegal Data Value

There are several instances of registers that are grouped together to collectively represent a single numerical DECS-2100 value. Examples include floating-point data and 32-bit integer data. A query to write a subset of such a register group will result in an error response with an exception code of Illegal Data Address.

**Loopback Diagnostic Test (FC=8) with Diagnostic Sub Function, Return Query Data**

This query contains data to be returned (looped back) in the response. The response and query message should be identical. If the query is broadcast (device address = 0), no response message is returned. Table 8 illustrates the format for this type of query.

**Table 8. Loopback Diagnostic Test, Return Query Data Format**

Device address	Function code = 08 (hex)	Sub-function high 00	Sub-function low 00	Data high XX (don't care)	Data low XX (don't care)	CRC low	CRC high
----------------	--------------------------	----------------------	---------------------	---------------------------	--------------------------	---------	----------

**Loopback Diagnostic Test with Diagnostic Sub-Function, Restart Communication Option**

This query causes the DECS-2100 remote communication function to restart, terminating an active listen-only mode of operation. Primary relay operations are not affected. Only the remote communication function is affected. If the query is a broadcast (device address of 0), no response message is returned.

If the DECS-2100 receives this query while in the listen-only mode (LOM), no response message is generated. Otherwise, a response message identical to the query message (see Table 9) is transmitted prior to the communication restart.

**Table 9. Loopback Diagnostic Test, Restart Communication Query Format**

Device address	Function code = 08 (hex)	Sub-function high 00	Sub-function low 00	Data high XX (don't care)	Data low XX (don't care)	CRC low	CRC high
----------------	--------------------------	----------------------	---------------------	---------------------------	--------------------------	---------	----------

**Loopback Diagnostic Test with Diagnostic Sub-Function, Force Slave into Listen-Only Mode**

This query forces the addressed DECS-2100 into the listen-only mode (LOM) for Modbus communication, isolating it from other devices on the network.

While in LOM, received data is monitored but no responses are transmitted. The only query that will be recognized and processed is the maintenance restart command (function code 8, diagnostic code 1).

When the DECS-2100 receives the restart communication query, LOM is terminated.

The LOM query data format is illustrated in Table 10.

**Table 10. Loopback Diagnostic Test, LOM Query Format**

Device address	Function code = 08 (hex)	Sub-function high 00	Sub-function low 00	Data high XX (don't care)	Data low XX (don't care)	CRC low	CRC high
----------------	--------------------------	----------------------	---------------------	---------------------------	--------------------------	---------	----------

Data Formats

DECS-2100 data does not require conversion for transmission over a Modbus network.

Modbus registers hold original DECS-2100 data of the generic (built-in) data types listed in Table 11.

**Table 11. Built-In DECS-2100 Modbus Register Data Types**

Generic Data Types	Modbus Controller	Corresponding Built-In Data Type (Storage Format)	Data Range	Data Size in Bytes	Total Number of Modbus Holding Registers
C1	1–6	LOGIC_32: Discrete/Binary Point	0 to 1 (False, True)	N/A	
I16	2, 4, 6	INT16: Signed Short Integer	–32,768 to 32,767	2	1
R32_23	1, 3, 5	FLOAT32: Floating Point Number	Approximately $8.43 \times 10^{-37}$ to $3.38 \times 10^{38}$	4	2

It should be noted that an ASCII string is not a DECS-2100 generic data type. An ASCII string will be considered as a sequence of “(string length + 1)” data of I8 type and for its transmission via a Modbus network, “(string length + 1)” holding registers are needed.

DECS-2100 data is copied to assigned holding registers [HR] by the rules presented in the following paragraphs.

**Generic Types I16**

Data of type INT16 is saved in one holding register. The high data byte is copied to the high HR byte and the low data byte is copied to the low HR byte.

For example, assume that DECS-2100 or INT16 type data whose value is 0xF067 is mapped to HR 47003. Data is copied to HR 47003 as shown in Table 12.

**Table 12. Generic Type INT16 Data Format**

HR 47002 Low Byte	HR 47003 High Byte	HR 47003 Low Byte	HR 47004 High Byte
...	0xF0	0x67	...

**Floating Point (R32\_32) Data Format**

The specific floating point format matches the floating point format used for the Modicon 984-8 family of programmable controllers.

Its representation, in bit format, is:



“S” is the sign bit for the floating point value: 1 if negative and 0 if positive. The “E” field is the two’s compliment exponent biased by 127 decimal. The “M” field is the 23-bit normalized mantissa. The most-significant bit of the mantissa is always assumed to be 1 and is not explicitly stored, yielding an effective precision of 24 bits.

The value of the floating-point number is obtained by multiplying the binary mantissa times two raised to the power of the unbiased exponent. The assumed bit of the binary mantissa has the value of 1.0, with the remaining 23 bits providing a fractional value. Table 13 shows the floating-point format.

**Table 13. Floating Point Data Format**

Sign	2’s Complement of (Exponent + 127)	Mantissa
1 bit	8 bits	23 bits

The floating point format allows a maximum value of  $3.38 \times 10^{38}$ .

Note that bytes 0 and 1 of the floating point value are stored in the lower numbered register, and bytes 2 and 3 are contained in the higher numbered register.

For example, number 123 in floating-point format is mapped to two holding registers (such as 45005 and 45006) as shown in Table 14.

**Table 14. Floating Point Value Example**

Register	45005	45006
Hexadecimal	0000	42F6
Binary	0000 0000 0000 0000	0100 0010 1111 0110

**Note**

For DECS-2100 Modbus, two consecutive holding registers that are mapped to any of the four-byte, generic data types, are considered to be linked together as an atomic, indivisible unit of information that can be read or written by a Modbus message only as one entity. That is, one cannot be read or written without the other.

CRC Error Check

This field contains a two-byte CRC value for transmission error detection. The master first calculates the CRC and appends it to the query message. The DECS-2100 recalculates the CRC value for the received query and performs a comparison to the query CRC value to determine if a transmission error has occurred. If so, no response message is generated. If no transmission error has occurred, the slave calculates a new CRC value for the response message and appends it to the message for transmission.

The CRC calculation is performed using all bytes of the device address, function code and data block fields. A 16-bit CRC register is initialized to all 1's. Then, each eight-bit byte of the message is used in the following algorithm.

First, exclusive-OR the message byte with the low-order byte of the CRC register. The result, stored in the CRC register, will then be right-shifted eight times. The CRC register MSB is zero filled with each shift. After each shift, the CRC register LSB is examined. If the LSB is a 1, the CRC register is then exclusive-ORed with the fixed polynomial value A001 (hex) prior to the next shift. Once all bytes of the message have undergone the above algorithm, the CRC register will contain the message CRC value to be placed in the error check field.

DECS-2100 Modbus Register Space

Modbus address space from 40000 to 49999 refers to function codes 3, 6, and 16. The DECS-2100 uses address space from 40001 to 44999 (4,999 registers). This address space is divided into 14 areas, referred to as information categories.

**DATA MAP**

The following data map information pertains to controllers 1, 3, or 5 in the ECM-2. See Figure 7 for reference. For even-numbered controllers (2, 4, or 6), contact Basler Electric.

Name	Modbus Address
Registers	
PMIN	
01	40001
02	40003
03	40005
04	40007
05	40009
06	40011

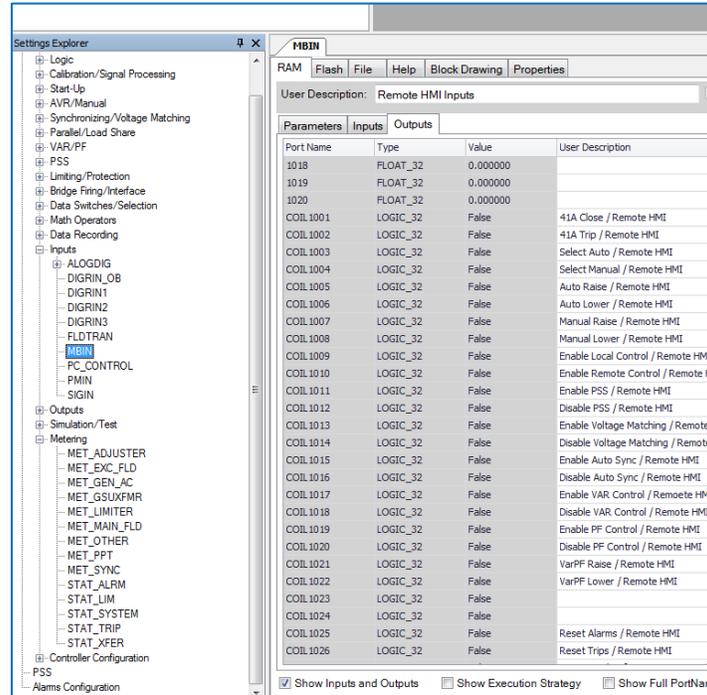
Figure 7. Data Map Information

The entire ECM-2 Modbus map is available in BESTCOMS™ Pro for reference, viewing, or printing. In the View Explorer under the Reports tree there is a selection for Modbus map. The Print Preview button offers the option to print or export to a spreadsheet or PDF file.

## DATA SOFTWARE BLOCKS

### Inputs/Control

The excitation system can be controlled with commands over the Ethernet link using remote input coils already in place. Input coil addresses for control are found in the MBIN block (Figure 8). Coils 1001 through 1064 are available. Many are preprogrammed for typical inputs but any may be changed for alternate functions. If alternate functions are desired, additional configuration changes may be required in logic blocks.



Port Name	Type	Value	User Description
1018	FLOAT_32	0.000000	
1019	FLOAT_32	0.000000	
1020	FLOAT_32	0.000000	
COIL.1001	LOGIC_32	False	41A Close / Remote HMI
COIL.1002	LOGIC_32	False	41A Trip / Remote HMI
COIL.1003	LOGIC_32	False	Select Auto / Remote HMI
COIL.1004	LOGIC_32	False	Select Manual / Remote HMI
COIL.1005	LOGIC_32	False	Auto Raise / Remote HMI
COIL.1006	LOGIC_32	False	Auto Lower / Remote HMI
COIL.1007	LOGIC_32	False	Manual Raise / Remote HMI
COIL.1008	LOGIC_32	False	Manual Lower / Remote HMI
COIL.1009	LOGIC_32	False	Enable Local Control / Remote HMI
COIL.1010	LOGIC_32	False	Enable Remote Control / Remote HMI
COIL.1011	LOGIC_32	False	Enable PSS / Remote HMI
COIL.1012	LOGIC_32	False	Disable PSS / Remote HMI
COIL.1013	LOGIC_32	False	Enable Voltage Matching / Remote HMI
COIL.1014	LOGIC_32	False	Disable Voltage Matching / Remote HMI
COIL.1015	LOGIC_32	False	Enable Auto Sync / Remote HMI
COIL.1016	LOGIC_32	False	Disable Auto Sync / Remote HMI
COIL.1017	LOGIC_32	False	Enable VAR Control / Remote HMI
COIL.1018	LOGIC_32	False	Disable VAR Control / Remote HMI
COIL.1019	LOGIC_32	False	Enable PF Control / Remote HMI
COIL.1020	LOGIC_32	False	Disable PF Control / Remote HMI
COIL.1021	LOGIC_32	False	VarPF Raise / Remote HMI
COIL.1022	LOGIC_32	False	VarPF Lower / Remote HMI
COIL.1023	LOGIC_32	False	
COIL.1024	LOGIC_32	False	
COIL.1025	LOGIC_32	False	Reset Alarms / Remote HMI
COIL.1026	LOGIC_32	False	Reset Trips / Remote HMI

Figure 8. MBIN Block

Input registers are also available in MBIN for setting inputs such as adjuster positions or presets. However, none are typically preprogrammed.

### Outputs/Data

Output data coils and registers are available in several BESTCOMS™ *Pro* output blocks.

A list of 14 blocks can be found in the Settings Explorer under Block Settings, Metering. These 14 blocks contain most, but not all, of the available coils and registers for reading through the Ethernet link. Blocks that begin with “MET\_” contain output registers and the blocks that begin with “STAT\_” contain output coils.

To view the assignments, click on the MET or STAT block desired and then check “Show Inputs and Outputs”. Open the Input tab to reveal the assignments. Figures 9 and 10 illustrate the output assignments and selections required to view them.

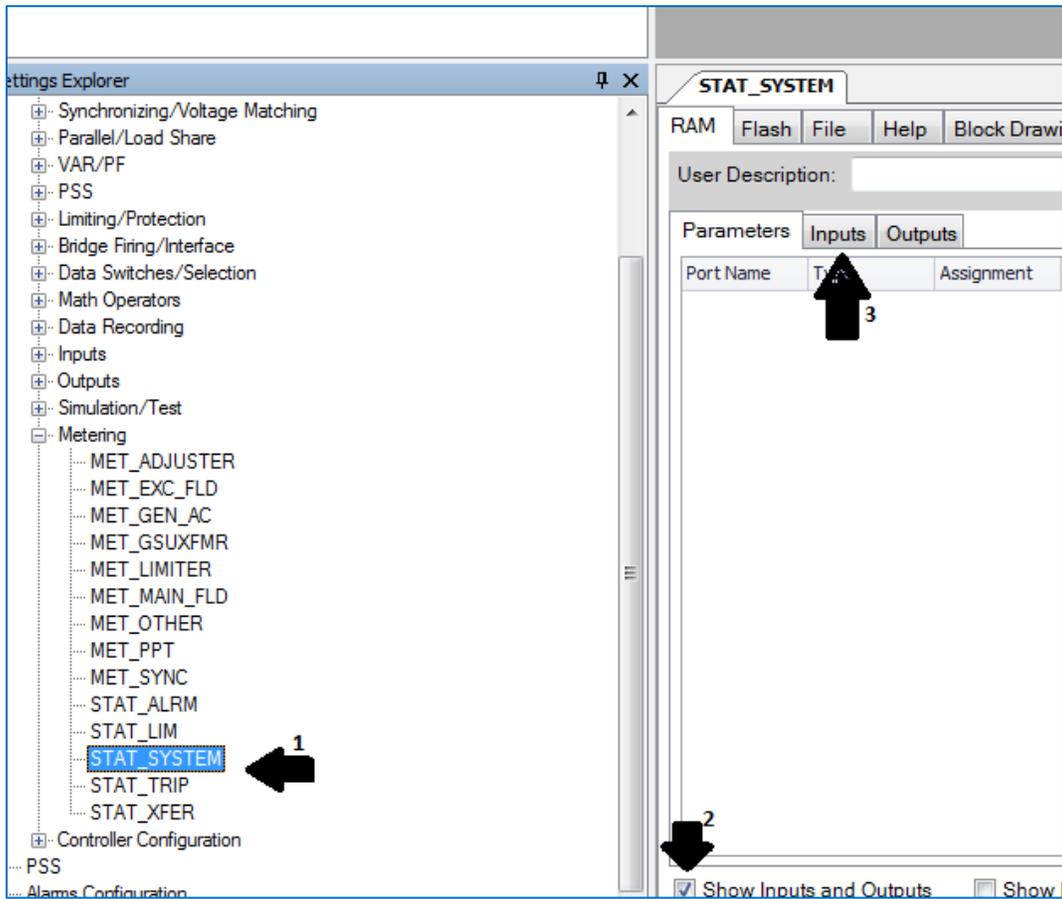


Figure 9. Output Assignments Example, 1 of 2

Modbus Settings		STAT_ALARM	
RAM	Flash	File	Help
Block Drawing			
Properties			
User Description:			
Parameters			
Inputs			
Outputs			
Port Name	Type	Assignment	Value
41A_FLOPN	LOGIC_32	BRKRCNTL00_F2	False
FLD_FLSH	LOGIC_32	FLDFLSH_F1	False
LOS	LOGIC_32	LOS200_F6	False
REGPTFL	LOGIC_32	AND58_OUT	False
METPTFL	LOGIC_32	AND59_OUT	False
VOLT_UNBAL	LOGIC_32	AND10_OUT	False
CUR_UNBAL	LOGIC_32	AND11_OUT	False
PSSCLP	LOGIC_32	OR39_OUT	False
PSAC	LOGIC_32	OR61_OUT	True
PSDC	LOGIC_32	DIGRIN1_C601	False
MNFLDGND	LOGIC_32	FLDGND_LOW	False
EXFLDGND	LOGIC_32	SIGIN_LOGICFALSE	False
PPT_TEMP	LOGIC_32	DIGRIN1_C606	False
PPT_HITEMP	LOGIC_32	DIGRIN1_C605	False
BRDGALM	LOGIC_32	AND19_OUT	False
BRDGTEMP	LOGIC_32	BCFG_COOLING_LOSS	False
LBF	LOGIC_32	LATCH11_OUT	False
USER 1	LOGIC_32	LATCH49_OUT	False
USER 2	LOGIC_32	AND20_OUT	False
USER 3	LOGIC_32	SIGIN_LOGICFALSE	False

Figure 10. Output Assignments Example, 2 of 2

Finding an input with an assignment such as SIGIN\_LOGICFALSE in a coil or SIGIN\_ZERO in a register would signify that the address is not be used in the configuration and thus it is a “spare”. So, if desired, the user may make an assignment for their use without disturbing the HMI.

In addition to the 14 output blocks available under Metering, there is also a block named PMOUT (Figure 11). It is found under Block Settings, Outputs, PMOUT. PMOUT contains 50 registers and 64 coils available and not typically assigned by Basler Electric. The user typically has access to all of the registers and coils in PMOUT without regard to the local HMI.

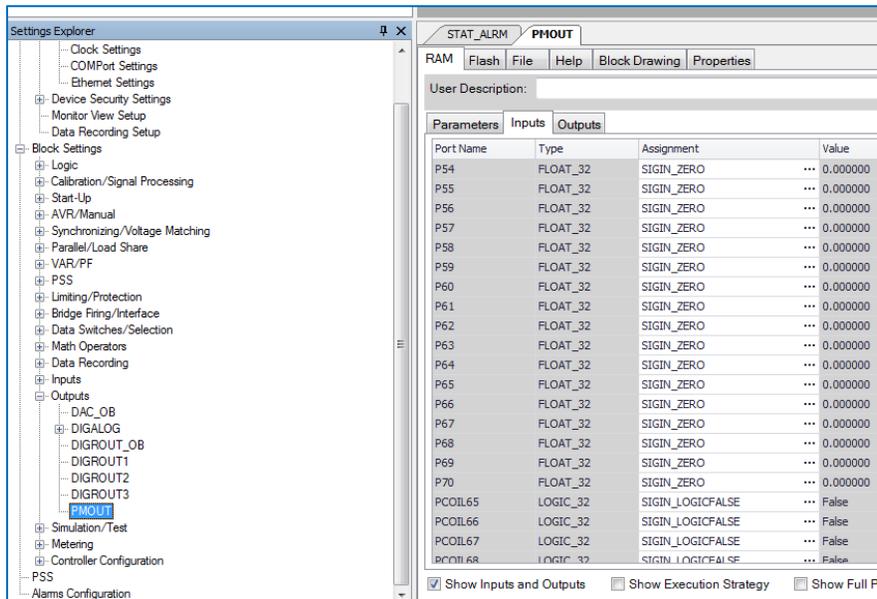


Figure 11. PMOUT Block

## HELP FILES

Coil and register numbers are shown in the individual block help file. With the block open in the Settings Explorer window, click the Help tab. Each variable is listed along with the Modbus address, data type, and scale. Most registers are in per-unit (PU) scale. An example of coil number information is illustrated in Figure 12 and an example of register number information is illustrated in Figure 13.

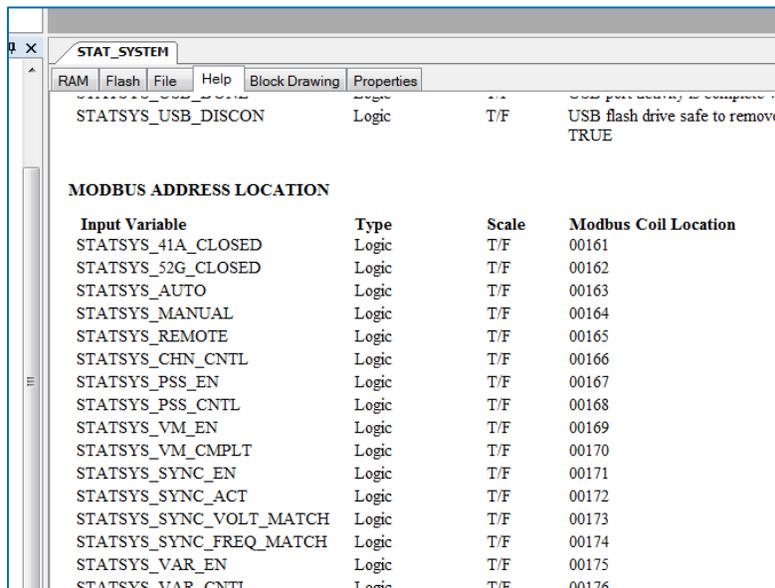


Figure 12. Coil Number Information Example

The DECS-2100 calculates most analog values in per-unit or percentage of full load. An example of PU scaling: if an 18 kV generator has a terminal voltage of 17,500 Vac, the PU value is .9722. To obtain the actual value, multiply the PU value by the rated value.

STAT_SYSTEM		MET_ADJUSTER	
RAM	Flash	File	Help
Block Drawing		Properties	
METADJ_TRKERR	Floating Point	PU	Manual and Auto control modes Difference between operating point and Redundant control channels
<b>MODBUS ADDRESS LOCATION</b>			
Input Variable	Type	Scale	Modbus Register Location
METADJ_ACREF	Floating Point	PU	40205 & 40206
METADJ_ACPOS	Floating Point	PU	40207 & 40208
METADJ_ACERR	Floating Point	PU	40209 & 40210
METADJ_MANREF	Floating Point	PU	40211 & 40212
METADJ_MANPOS	Floating Point	PU	40213 & 40214
METADJ_MANERR	Floating Point	PU	40215 & 40216
METADJ_VARREF	Floating Point	PU	40217 & 40218
METADJ_VARPOS	Floating Point	PU	40219 & 40220
METADJ_VARERR	Floating Point	PU	40221 & 40222
METADJ_PFREF	Floating Point	PU	40223 & 40224
METADJ_PFPOS	Floating Point	PU	40225 & 40226
METADJ_PFERR	Floating Point	PU	40227 & 40228
METADJ_AMBAL	Floating Point	PU	40229 & 40230
METADJ_TRKERR	Floating Point	PU	40231 & 40232

Figure 13. Register Number Information Example

## SINGLE-CHANNEL VERSUS MULTI-CHANNEL SYSTEMS

In a two- or three-channel system, each channel may be scanned (queried for data) separately with data from each channel having dedicated memory locations in the master device. Logic in the polling device (the master) would then use the data received from the controlling channel for alarm and status outputs.

If command signals are used, any command (start, stop, raise, lower, etc.) should be broadcast to all channels.

Assuming a healthy system, all Modbus output coils in each channel should agree except coil 0166. Coil 0166 is the bit the master uses to discern which channel is in control. Table 15 serves as a truth table which may be used to determine channel control.

Table 15. Controlling Channel Truth Table

Condition	Coil 0166 Main Channel	Coil 0166 Redundant Channel
Alarm condition	0	0
Main channel in control	0	1
Redundant channel in control	1	0
Error condition	1	1

The 0 condition for both channels is noted as an alarm condition in Table 15 because if either channel loses power or control fuse ANFU opens, the 0 condition will exist for both channels.

An alternate method of determining which channel is in control is available through a hardwired signal obtained from the KT relay. Spare KT relay contacts are provided. If the KT relay is not energized, the main channel is in control. Conversely, when the KT relay is energized, the redundant channel is in control.

The third channel, if provided, is never in control of the regulator so coil 0166 carries no meaning in that channel. The third channel is only used as a voter in the two-of-three voting actions such as open, close, transfer, and trip.

## MAKING AND SAVING CHANGES

---

When assignment changes are made, the changes need to be saved into flash memory and possibly onto the user's PC. If the only changes made are to the Ethernet addressing, then saving is not necessary.

Changes made to assignments are saved in RAM. To save the changes to flash memory and the user's PC, select Configurations on the menu bar. Under Configurations select "Copy RAM to Flash" (Figure 14). This will copy the file, including changes made to the ECM-2 flash memory. Remember that any changes made in one channel should be made in all channels. If the user wishes to keep a copy of the configuration on the PC, "Copy RAM to File" will copy the configuration file to the PC. Use the menu "File/Save As" command to save the file to the desired location on the PC.

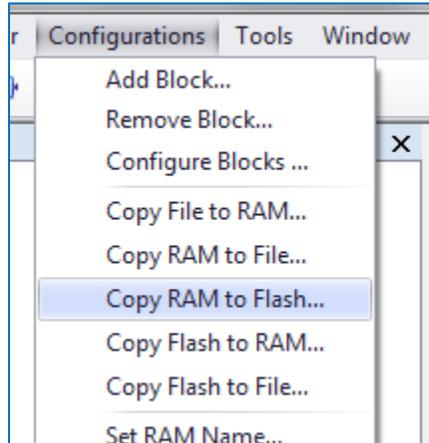


Figure 14. "Copy RAM to Flash" Menu Selection



Publication <b>9411300996</b>	Revision <b>A</b>	<b><i>Instructions</i></b>	Date <b>08/14</b>	Page <b>18 of 18</b>
----------------------------------	----------------------	----------------------------	----------------------	-------------------------