

# Communication between Siemens S7-1200 and Vaisala HMP7 probe via Modbus RTU

2021-10-04

This guide is meant to be used to help you communicate with an HMP7 probe with a Siemens S7-1200 PLC using Simatic Step 7 v16 software.

In this guide we show you how to wire an HMP7 probe to the communication board (CB 1241 RS485) on a Siemens S7-1200 PLC, communicate between the HMP7 and the S7-1200 via Modbus RTU and convert the holding registers into human readable floating-point values. We use the Siemens S7-1200 as a Modbus master to read the relative humidity and temperature readings from the HMP7 as the Modbus slave. Some tables and diagrams are provided to use as a guide, but it is highly recommended to reference your own instruments' manual. It is assumed that you have at least some basic experience with ladder logic, the Simatic Step 7 software and connecting the Siemens S7-1200 to a PC.

# 1 Wiring

In this section we connect a single HMP7 probe to the communication board (CB 1241 RS485) on a Siemens S7-1200 PLC. We use a cable from Vaisala for the physical connection. Make sure your PLC is detached from a power source before you attempt any wiring.

- 1. If you are using a cable from Vaisala, the following wire colors signify the function:
  - Brown: Power supply
  - White: RS-485 –
  - Blue: Power GND and RS-485 common
  - Black: RS-485 +
- 2. If not, reference the diagram below.



Figure 4 M12 5-pin A-coded male connector pinout

Pin #	Function	Notes	Wire colors in Vaisala cables
1	Power supply	Operating voltage:	Brown
		<ul> <li>HMP7: 18 30 V DC</li> <li>Other models: 15 30 V DC</li> </ul>	
		Current consumption: 10 mA typical, 500 mA max.	
2	RS-485 -		White
3	Power GND and RS-485 common		Blue
4	RS-485 +		Black
5	Not connected		Gray

(HMPx Modbus wiring: HMP Series User Guide p.22)



- Probe
   RS-485 host

   Pin #4 0
   RS-485 host

   RS-485 +
   RS-485 +

   Pin #2 0
   RS-485 

   RS-485 RS-485 common

   Pin #5 0
   Pin #1 0

   Pin #3 0

   Power GND RS-485 common
- 3. See the diagram below for the wiring.





Recommended maximum length of the RS-485 line is 30 m (98 ft).

### (HMPx Modbus wiring: HMP Series User Guide p.23)



① Connect M to the cable shield

- ② A = TxD/RxD (Green wire / Pin 8)
- 3 B = TxD/RxD + (Red wire / Pin 3)
  - 4. Attach T/RA to the white cable (pin 2) and T/RB to the black cable (pin 4) in the Vaisala cable. Connect the Brown cable (pin 1) to DC power +, and the blue cable (pin 3) to ground (M) and DC power -



# 2 Configuring the port on the communication module for Modbus RTU

In this section we will configure the communication module for Modbus RTU using the function block MB\_COMM\_LOAD in TIA16 with the Simatic Step 7 Basic software.

- 1. Drag an empty box to an empty network and type in MB\_COMM\_LOAD or select it from the instruction window on the right-hand side of the screen.
- 2. The Call options window opens where you can change the name of the block and confirm it.

Vetwork 1:	
Comment	
%DB1       *MB_COMM_ LOAD_DB*       False       EN        EN       EN       EN       EN       EN       EN       EN       EN       EN       EN       EN       EN       EN       EN       EN       EN	Call options Data block Name MB_COMM_LOAD_DB_1  Single instance  If you call the function block as a single instance, the function block saves its data in its own instance data block.
Network 2:	
Comment	
1	
	more
	OK Cancel

3. Create a new Data block to store the variables for MB\_COMM\_LOAD

Add new block				×
Name:				
MB_CONFIG				
	Type:	Global DB		
<b></b>				
-OB	Language:	DB		
Organization block	Number:	2		
		🔘 Manual		
		<ul> <li>Automatic</li> </ul>		
	Description:			
гв	Data blocks (DBs	) save program data.		
Function block				
Function				
БВ				
Data block				
	more			
> Additional inform	ation			
🛃 Add new and open			ОК	Cancel

4. Uncheck the "Optimized block access" box under attributes in properties





- 5. Add the following variables with the data types in the data block for the MB\_COMM\_LOAD function block:
  - BAUD, UDInt
  - PARITY, UInt
  - DONE, Bool
  - ERROR, Bool
  - STATUS, Word

	PORT_CONFIG										
		Name		Data type	Offset	Start value	Retain	Accessible f	Writa	Visible in	Setpoint
1	-	<ul> <li>Static</li> </ul>									
2	-	<ul> <li>BAUD</li> </ul>		UDInt		19200		<b></b>	$\checkmark$	<b></b>	
3	-	DONE		Bool		false		<b></b>	$\checkmark$	<b></b>	
4	-	ERROR		Bool		false		<b></b>	$\checkmark$	<b></b>	
5	-	STATUS	5	Word		16#0		<b></b>	$\checkmark$	<b></b>	
6		Add r	ew>								

- 6. Set the baud rate and parity according to your probe's specifications. In our case 19200 for the baud rate and 0 for the parity.
- 7. Add the variables from the Data block to MB\_COMM\_LOAD\_DB



- 8. Set the PORT variable to the communication module that is installed on your PLC
- 9. To change the stop bits, under the Program blocks folder, open the System blocks folder then the Program resources folder, then open MB\_COMM\_LOAD\_DB. At the bottom of MB\_COMM\_LOAD\_DB you will find the STOP\_BITS where you can set the appropriate value for your probe. We set the value as 2.



	)evio	es	]				
- Péñ					🗐 🔿		
				: L			
<b>•</b> [	) Decis	ogle p	rohe quide				
- [	51	igie pi Addu	nobe guide				
	-	Devic	es & petworks				
	000 • •						
	-		evice configuration				
		V. 0	nline & diagnostics				
	•	Pr	rogram blocks				
		-	Add new block				
			Main [OB1]				
			CONVERT TO FLOAT [FB1	1			
			CONVERT_TO_FLOAT_DB	[DB6]			
			DATA [DB5]				
			MB_CONFIG [DB4]				
			PORT_CONFIG [DB2]				
		- 6	System blocks				
		-	🖥 Program resources				
			MB_COMM_LOAD	[FB1080]			
			MB MASTER (FB10	811			
			MB_COMM_LOAD_	DB [DB1]			
			MB_MASTER_DB [D]	B3]			
	MB	_CO	MM_LOAD_DB				
		Name	•	Data type	Start val	ue	Retain
1		▼ In	put				
2		•	REQ	Bool	false		
з	-00	•	PORT	PORT	16#FFF	F	[
4		•	BAUD	UDInt	0		
5	-00	•	PARITY	UInt	0		
6	-	•	FLOW_CTRL	UInt	0		Ē
7	-00	•	RTS_ON_DLY	UInt	0		Ē
8	-00	•	RTS_OFF_DLY	UInt	0		Ē
9	-	•	RESP_TO	UInt	1000		Ē
10		<b>-</b> 0	utput				Ē
11		•	DONE	Bool	false		Ē
12		•	ERROR	Bool	false		Ē
13		•	STATUS	Word	16#0		-
14	-	▼ In	Out				
15			MB_DB	MB_BASE			Ē
16	-	▼ St	tatic	_			Ē
17	-		ICHAR GAP	UInt	0		Ē
18			RETRIES	UInt	2		Ē
19	-		WRREC STATUS	Word	2		
20	-		RDREC STATUS	Word	2		Ē
21			SEC STATUS	Word	2		
22	-		Port CFG SFB	Array[0.,25] of Byte			-
23	-		Send CFG SFB	Array[0.,16] of Byte			
24	-		Rev CEG SEB	Array[0, 60] of Byte			
25		. 1	STOP BITS	USInt	2		
20		-	2101_0112	oome	4		



## 3 Communicating with the HMP7 Probe

In this section we will show you how to use MB\_MASTER function block to communicate with the HMP7 probe, which values to choose for the variables and how to store the holding registers.

- 1. Drag an empty box to an empty network and type MB\_MASTER into it or select the block from the instruction window on the right-hand side of the screen.
- 2. The Call options window opens where you can change the name of the block and confirm it.

*PORT_CONFIG*. BAUD — BAUD 0 — PARITY ?? — MB_DB ~	SIAIUS — SIAIUS	Call options	Data block Name MB_MASTER_DB Number Manual	×
3 Network 2:			<ul> <li>Automatic</li> </ul>	
Comment			If you call the function block as a single instance, the function block saves its data in its own instance data block.	n
!!				
MB_MASTE	R			
EN DEC	ENO	-		
KEQ	DONE			
MB_ADDR	BUSY			
/ MODE	ERROR			
?? — DATA_ADDR	STATUS			
?? — DATA_LEN				
?? — DATA_PTR			more	
			OK Cancel	

3. Create a new Data block to store the variables for MB\_MASTER



4. Uncheck the "Optimized data access" box under attributes in properties.

Project tree		Ω		sing
Devices				
210 200			<b>a</b>	
				N
single probe quide				
Add new device				1 4
Devices & networks				2
PLC_1 [CPU 1212C AC/	DC/RI	vl		-
Device configuration	n			
😨 Online & diagnosti	cs			
🔻 🔙 Program blocks				
🗳 Add new block				
🎦 Main [OB1]				
MB_CONFIG [DB	41			
PORT_CONFIG [E	DE	Open		
🕨 🔙 System blocks	Х	Cut	Ct	rl+X
Technology objects		Сору	Ct	rl+C
🕨 🛅 External source file	s 🗎	Paste	Ct	rl+V
🕨 🚂 PLC tags		Copy as text		
PLC data types	~	Delete		Del
Watch and force tal	ь ^	Rename		F2
Online backups	-	Compile		
Traces		Complie Download to device		
OPC UA communication	a 🖌	Go online	Ct	rl+K
Device proxy data	5	Go offline	Ct	rl+M
Program into	100	Connected of the actual values		
PLC alarm text lists		Load spanshots as actual values	105	
Local modules	12	Load start values as actual val	dues	
Geographic advices		Copy snapshots to start value	s	•
Cross-device functions	ata .	o ist server		
Common data		Quick compare		•
Documentation setting	, <b>i</b> n	Search in project	Ct	trl+F
Languages & resource	, 🖿	Generate source from blocks		•
Version control interfa	c 🗴	Cross-references		F11
Gonline access	×	Cross-reference information	Shift+	F11
ig Card Reader/USB memory		Call structure		
		Assignment list		
		Switch programming languag	e	•
		Know-how protection		
	_	knownow protection		
	-	Print	Ct	trl+P
	1	Print preview		
	Q	Properties	Alt+E	nter
IR CONFIG [DB4]				
Conoral Tauta				
General				
General	Δ++	ributos		
Information	лц	induces		
Time stamps				
Compilation	F	Only store in load memo	ry	
Protection				
Attributes		Data block write-protecte	a in i	ine de
Developing		Optimized block access		

General Text	S
General Information	Attributes
Time stamps	
Compilation	Only store in load memory
Protection	Data block write-protected in the device
Attributes Download with	Optimized block access Data block accessible from OPC UA Data block accessible via Web server
< m >	
	OK Cancel

- 5. Add the following variables with the data types to the new data block:
  - MB\_ADDR, UInt
  - MODE, USInt
  - DATA\_ADDR, UDInt

×



- DATA\_LEN, UInt
- DONE, Bool
- BUSY, Bool
- ERROR, Bool
- STATUS, Word
- REQ, Bool

	MB	_COI	IFIG								
		Name	•	Data type	Offset	Start value	Retain	Accessible f	Writa	Visible in	Setpoint
1		▼ St	tatic								
2	-	•	MB_ADDR	UInt		1			$\checkmark$	<b></b>	
3	-	•	MODE	USInt		0			$\checkmark$	<b></b>	
4	-	•	DATA_ADDR	UDInt		40001			$\checkmark$		
5	-	•	DATA_LEN	UInt		4			$\checkmark$	<b></b>	
6		•	DONE	Bool		false		<b></b>	<b>~</b>	<b></b>	
7		•	BUSY	Bool		false			$\checkmark$	<b></b>	
8		•	ERROR	Bool		false		<b></b>	<b>~</b>	<b></b>	
9		•	STATUS	Word		16#0			$\checkmark$	<b></b>	
10		•	REQ	Bool 🔳		false			<b></b>		
11			<add new=""></add>								

Table 13- 150 Data types for the parameters

Parameter and ty	/pe	Data type	Description		
REQ	IN	Bool	0=No request 1= Request to transmit data to Modbus slave		
MB_ADDR	IN	V1.0: USInt	Modbus RTU station address:		
		V2.0: UInt	Standard addressing range (1 to 247) Extended addressing range (1 to 65535)		
			The value of 0 is reserved for broadcasting a message to all Modbus slaves. Modbus function codes 05, 06, 15 and 16 are the only function codes supported for broadcast.		
MODE	IN	USInt	Mode Selection: Specifies the type of request (read, write, or diagnostic). See the Modbus functions table below for details.		
DATA_ADDR	IN	UDInt	Starting Address in the slave: Specifies the starting address of the data to be accessed in the Modbus slave. See the Modbus functions table below for valid addresses.		
DATA_LEN	IN	UInt	Data Length: Specifies the number of bits or words to be accessed in this request. See the Modbus functions table below for valid lengths.		
DATA_PTR	IN	Variant	Data Pointer: Points to the M or DB address (non-optimized DB type) for the data being written or read.		
DONE	OUT	Bool	The DONE bit is TRUE for one scan, after the last request was completed with no error.		
BUSY	OUT	Bool	<ul> <li>0 – No MB_MASTER operation in progress</li> </ul>		
			1 – MB_MASTER operation in progress		
ERROR	OUT	Bool	The ERROR bit is TRUE for one scan, after the last request was termi- nated with an error. The error code value at the STATUS parameter is valid only during the single scan where ERROR = TRUE.		
STATUS	OUT	Word	Execution condition code		

#### 1258

S7-1200 Programmable controller

System Manual, V4.2.3, 08/2018, A5E02486680-AL

6. We want to read the holding registers, to do so we must choose Mode 0, and the Modbus Address starting from 40001. Address 40001 corresponds to register 1.



MODE	Modbus Function	Data length	Operation and data	Modbus Address
0	01	1 to 2000 1 to 1992 <sup>1</sup>	Read output bits: 1 to (1992 or 2000) bits per request	1 to 9999
0	02	1 to 2000 1 to 1992 <sup>1</sup>	Read input bits: 1 to (1992 or 2000) bits per request	10001 to 19999
0	03	1 to 125 1 to 124 <sup>1</sup>	Read Holding registers: 1 to (124 or 125) words per request	40001 to 49999 or 400001 to 465535
0	04	1 to 125 1 to 124 <sup>1</sup>	Read input words: 1 to (124 or 125) words per request	30001 to 39999
1	05	1	Write one output bit: One bit per request	1 to 9999
1	06	1	Write one holding register: 1 word per request	40001 to 49999 or 400001 to 465535
1	15	2 to 1968 2 to 1960 <sup>1</sup>	Write multiple output bits: 2 to (1960 or 1968) bits per request	1 to 9999
1	16	2 to 123 2 to 122 <sup>1</sup>	Write multiple holding registers: 2 to (122 or 123) words per request	40001 to 49999 or 400001 to 465535
2	15	1 to 1968 2 to 1960 <sup>1</sup>	Write one or more output bits: 1 to (1960 or 1968) bits per request	1 to 9999
2	16	1 to 123 1 to 122 <sup>1</sup>	Write one or more holding registers: 1 to (122 or 123) words per request	40001 to 49999 or 400001 to 465535
11	11	0	Read the slave communication status word and event counter. The status word indicates busy (0 – not busy, 0xFFFF - busy). The event counter is incremented for each successful completion of a message.	
			Both the DATA_ADDR and DATA_LEN operands of MB_MASTER are ignored for this function.	
80	08	1	Check slave status using data diagnostic code 0x0000 (Loopback test – slave echoes the request)	
			1 word per request	

Table 13- 151 Modbus functions

1260

S7-1200 Programmable controller

System Manual, V4.2.3, 08/2018, A5E02486680-AL

7. Set the variables to their appropriate places on MB\_MASTER\_DB



8. Once you have created MB\_MASTER\_DB it is time to pass the instance of it to the MB\_DB argument of MB\_COMM\_LOAD\_DB



		%DB1 MB_COMM_ LOAD_DB*	
	MB	COMM_LOAD	
	EN	E	мо ————————————————
%M1.0 "FirstScan"	REQ	DO	DB2.??? "PORT_CONFIG". NE - DONE
<b>269</b> Local~CB_ 1241_(RS485)		ERF	DB2.??? "PORT_CONFIG". COR — ERROR
DB2.??? "PORT_CONFIG".	PORT	STAT	DB2.??? "PORT_CONFIG". TUS — STATUS
0	PARITY		
		▼ Instance DB of	DB1 A
"MB_CONFIG"	.00	Global DB	DB4
"MB_MASTER_DR"		Instance DB of	DB3
		instance pp of	
			~

Now it is time to create a new Data block to store the values from the holding registers

1. Create a new data block to store the holding registers by clicking on the "Add new block" under Program blocks.

Add new block					×
Name:					
Daid					
	Type:	🧧 Global DB	•		
OB	Language:	DB	-		
Organization	Number:	5			
		🔘 Manual			
		<ul> <li>Automatic</li> </ul>			
FB	Description:				
Function block	Data blocks (DB	is) save program data.			
FC					
Function					
В					
Data block					
	more				
Additional inform	ation		_		
Add new and open				ОК	Cancel

2. Uncheck "Optimized block access" in Attributes under properties.



< III >

Project tree				
Devices				
ÊŃ				
r 🗖 single or	ohe quide			
Single pr	new device			
- Aud n	es & petwor	rks		
	1 [CPU 1212	C A	C/DC/RIv]	
	evice config	urat	ion	
V. Or	nline & diag	nos	tics	
👻 🔜 Pro	ogram block	s		
	Add new b	lock		
	Main [OB1]	]		
5	DATA [DB5	-	Onen	
	MB_CONFI		open	
	PORT_CON	Ж	Cut	Ctrl+X
• 🕞	System blo	1	Сору	Ctrl+C
🕨 🙀 Te	chnology ob		raste	Ctrl+V
🕨 🐻 Ex	ternal sourc		Copy as text	
🕨 🍋 PL	.C tags	×	Delete	Del
🕨 📴 PL	C data type:	1	Rename	F2
🕨 🔜 Wa	atch and for		Compile	•
🕨 🚺 Or	nline backup		Download to device	
🕨 💽 Tra	aces	ø	Go online	Ctrl+K
🕨 🐼 OF	PC UA comm	1	Go offline	Ctrl+M
🕨 🗎 De	evice proxy o	10.	Snapshot of the actual va	lues
De Pro	ogram info	10.	Load snapshot of the actual va	lvalues
E PL	.C alarm text		Load start values as actu	al values
🕨 🧎 Lo	cal module	-	Copy snapshots to start v	alues 🕨
🕨 📙 Ungro	ouped devic	da		
🕩 📷 Secur	rity settings	919	Quick compare	•
Cross	-device fund	٩.	Search in project	Ctrl+F
🕨 🧃 Comn	non data		Generate source from blo	cks 🕨
Docur	mentation s		Cross-references	E14
🕨 🚺 Langu	uages & res		Cross-reference information	rii on Shift⊥F11
Versio	on control ir		Call structure	on onne+rri
Online a	ccess		Assignment list	
🔄 📑 Card Rea	der/USB me	_	Cuitale and a second second	
			Switch programming lang	juage 🕨
			Know-how protection	
			Print	Ctrl+P
		4	Print preview	
		0	Properties	Alt+Enter
	1		inopendes	Areachter
DATA [DB5	1			
Genera	lex	ts		
General			Attributos	
Informat	tion		Autoutes	
Time sta	mps			
Compilat	tion		Only store in load i	memory
Protectio	on		Data block write-p	rotected in the
Attribute	s			otected in the
Downloa	d with		Optimized block a	ccess
		4	🛃 Data block access	ible from OPC U
		-	Data block access	ihle via Webse
		•	- Data block access	ione via web se

3. Create an array of UInt with the same number of elements as the value for DATA\_LEN you selected.

X

Cancel

ОК

DATA											
	Name			Data type	Offset	Start value	Retain	Accessible f	Writa	Visible in	Setpoint
-	•	St	atic								
-00	•	•	REGISTERS	Array[03] o 🔳 💌						<b></b>	
	1	•	REGISTERS[0]	UInt		0		$\checkmark$	<b>v</b>	<b>V</b>	
		•	REGISTERS[1]	UInt		0		$\checkmark$	<b>v</b>	<b>V</b>	
		•	REGISTERS[2]	UInt		0		$\checkmark$	<b>V</b>	<b>V</b>	
		•	REGISTERS[3]	UInt		0		$\checkmark$	<b>V</b>	<b>V</b>	
	•		<add new=""></add>								

4. Add the array to the DATA\_PTR variable of MB\_MASTER



# 4 Converting word addresses to human readable 32-bit floating point values:

The data sent via Modbus is split into two 16-bit words, we need to combine the two words into a 32-bit floating point value to get a readable value. We will create a new Function block that moves and converts the bits into a 32-bit float.

- 1. Create a new Function block to convert the values by clicking on the "Add new block" under Program blocks and selecting function block. Make sure that the language selected is LAD.
- 2. Uncheck "Optimized block access" under attributes in properties



Add new block					×
Name:					
CONVERT_TO_FLOAT					
Organization block	Language: Number:	LAD 2 Manual Automatic	<b>•</b>		
Function block	Description: Function blocks ar so that they remai	re code blocks that sto in available after the b	ore their values per lock has been exec	nanently in instance data blocks uted.	,
=					
Function					
В					
Data block	more				
> Additional inform	ation				
<ul> <li>Additional Information</li> </ul>	auon				
Add new and open				OK Cancel	

- 3. We will take advantage of the AT overlay to combine the two words into a 32-bit floating point value. First create the Real variable into which the two words are combined in the Output section of the function block variables.
- 4. Directly under the Real variable, create another variable with the Data type as "AT"

co	CONVERT_TO_FLOAT								
-	Name	Data type	Offset	Default val	Accessible f	Writa	Visible in	Setpoint	
	💌 Input								
	🛯 = 🕨 Data	Array[03] of UInt			<b></b>	<b></b>	<b></b>		
	Add new>								
	🔻 Output								
	Relative_Humidity	Real		0.0	<b></b>	<b></b>	<b></b>		
	RH_Parts AT"	Array[01] of UInt							
	= Temperature	Real		0.0		<b></b>			
	T_Parts	AT I		0.0	$\checkmark$	<b>~</b>	<b></b>		
	Add new>	AOM_IDENT	^						
	🔻 InOut	AT	=						
	Add new>	Array[01] of							
	🔻 Static	Bool							
	Add new>	Byte							
	🔻 Temp	CONN_ANY							
	Add new>	CONN_OUC							
	<ul> <li>Constant</li> </ul>	CONN_PRG	¥.						
	Add new>								

5. The AT variable Data type should then change to that of an array, change the array Data type to UInt and the Array limits to 0..1



CO	CONVERT_TO_FLOAT										
	Name				Data type	Offset	Default val	Accessible f	Writa	Visible in	Setpoint
	•	Inj	out								
	•	•	Data		Array[03] of UInt				<b></b>	<b>~</b>	
	•		<add new=""></add>								
	•	Ou	itput								
-00	•		Relative_Humi	lity	Real		0.0		<b></b>		
-00		٠	RH_Parts	AT "	Array[01] of UInt						
		•	RH_Parts[0]		UInt						
		•	RH_Parts[1]		UInt						
-00	•		Temperature		Real		0.0		<b>~</b>		
		•	T_Parts	AT "T	зу[01] of Bool 🔳 🔻						
	•		<add new=""></add>			Lun d					
	٠	In(	Dut		Data type:	UInt	<b></b>				
	•		<add new=""></add>		Array limits:	01					
-00	•	St	atic		-						
	•		<add new=""></add>		Examples:	099 or 0	99,010				
	•	Те	mp								
	•		<add new=""></add>								
	•	Co	nstant								
	•		<add new=""></add>								

6. Add into the input section, an array of UInt with the same number of elements as the Holding register array.

CO	CONVERT_TO_FLOAT										
Name					Data type	Offset	Default val	Accessible f	Writa	Visible in	Setpoint
	•	Inp	put								
	-	۲	Data		Array[03] of UInt				<b></b>		
	•		<add new=""></add>								
-00	•	Ou	utput								
-00	•		Relative_Hum	idity	Real		0.0		<b></b>	<b>~</b>	
-00		•	RH_Parts	AT"	Array[01] of UInt						
-00		•	RH_Parts[0	]	UInt						
-00		•	RH_Parts[1	]	UInt						
-00	•		Temperature		Real		0.0		<b>~</b>		
-00		٠	T_Parts	AT "T	Array[01] of UInt						
-00		•	T_Parts[0]		UInt						
-		•	T_Parts[1]		UInt						
	•		<add new=""></add>								
-	•	In	Out								
	•		<add new=""></add>								
-	•	Sta	atic								
	•		<add new=""></add>								
	•	Те	mp								
	•		<add new=""></add>								
	•	Co	instant								
	•		<add new=""></add>								

7. Now it is time to move the holding register data into the overlaying array. We get the data from the probe in little endian format, we must then reverse the order of the words when we move them to the overlaying array. In other words, the second element of the data array goes to the first element of the target array and the first element of the data array goes to the second element of the target array.



•	Network 1:
	Comment
	MOVE     MOVE       EN     ENO       #Data[1]     IN       IN     OUT1       #RH_Parts[0]     #Data[0]       IN     OUT1
•	Network 2:
	Comment
	MOVE         MOVE           EN         ENO           #Data[3]         IN           IN         OUT1           #T_Parts[0]         #Data[2]           IN         OUT1

- 8. Add the function block to a new network in the main program block and have the done bit from MB\_MASTER trigger the function block.
- 9. Set the data array as the input and save the outputs in another variable.





# 5 Triggering a MB\_MASTER cycle

In this section we will show the ladder logic that continuously triggers a new MB\_MASTER cycle after each completed cycle. We want the new cycle to be triggered after the probe has responded or it has ended in an error. We added a closed BUSY contact to make sure that a new request does not attempt to trigger a new cycle while MB\_MASTER is already busy with the previous request. We added the "FirstScan" open contact to trigger the first cycle before the done or the error bit is activated.

