G&L Modbus Serial Example Revision A 6-3-2019

Note!! This example was done with the Digital MMC Smartdrive and Drive Resident Control 16 Axis. The programmer is responsible for any settings and wiring that are different due to differences in hardware.

Also note that the Modbus Serial ASFB library is not free-ware and must be purchased from your local Kollmorgen supplier.

First I created a folder called Modbus Serial Example under the C:\G&L Motion Control Data directory. This will be used later in this application note.



I put the Modbus Serial zip file in a folder I created to allow the files to be unzipped to the new folder.



Unzipping and viewing the contents there is a folder called "ModbusSerial" and a subfolder called "Serial" which contains a large number of files which will be explained shortly.

3ndu601_ND0.zip	^	Name	^	Date modified	Туре	Size	
AKD BASIC Servo Drive Firmware (AKD-T-ICAN-01-17-00-000).zip		C MASTEX.bin		5/18/2011 10:30 AM	BIN File	29 KB	
ascii_sr600		C_MASTEX.frc		5/13/2011 8:18 AM	FRC File	1 KB	
ascii_sr600.zip		C_MASTEX.lbk		5/18/2011 1:11 PM	PiCPro Ladder Bac	2 KB	
Desktop		C_MASTEX.LDO		5/18/2011 1:12 PM	PiCPro Ladder File	2 KB	
Documents		C_MASTEX.map		5/18/2011 10:30 AM	Linker Address Map	4 KB	
Deumlande		C_MASTEX.rbk		5/18/2011 1:11 PM	RBK File	3 KB	
		C_MASTEX.rem		5/18/2011 1:12 PM	REM File	3 KB	
Kollmorgen Visualization Builder IDE EN (REV 2.22.353).zip		C_MEMX.LBK		7/2/1999 8:42 AM	PiCPro Ladder Bac	4 KB	
MON161S3		C_MEMX.LDO		6/4/2001 1:20 PM	PiCPro Ladder File	4 KB	
MON161S3.zip		C_MEMX.RBK		7/2/1999 8:42 AM	RBK File	7 KB	
Music		C_MEMX.REM		6/4/2001 1:20 PM	REM File	7 KB	
P7000_Tools_ 2.11.zip		C_MOD.bak		2/28/2012 11:11 AM	BAK File	36 KB	
PiCProMonV16 1.zip		E C_MOD.LIB		2/28/2012 11:11 AM	Object File Library	35 KB	
Dictures		C_MODAUX.bak		2/28/2012 11:11 AM	BAK File	5 KB	
1 Clares		III C_MODAUX.LIB		2/28/2012 11:11 AM	Object File Library	5 KB	
videos		C_MODCRC.lbk		5/11/2011 2:43 PM	PiCPro Ladder Bac	4 KB	
SYSTEM & DATA (C:)		C_MODCRC.Ido		2/28/2012 11:11 AM	PiCPro Ladder File	4 KB	
ENG (\\RADVSVFP02\DATA\$) (F:)		C MODCRC.rbk		5/11/2011 2:43 PM	RBK File	3 KB	

I copied the entire ModbusSerial folder and pasted it into the C:\G&L Motion Control Data for each of location.

3ndu601_ND0.zip	^	Name	Date modified	Туре	Size
AKD BASIC Servo Drive Firmware (AKD-T-ICAN-01-17-00-000).zip		Applications V16.1.1	2/25/2019 11:21 AM	File folder	
ascii_sr600		Applications V18.0.1	2/24/2019 9:15 AM	File folder	
💡 ascii_sr600.zip		Applications16_1	2/25/2019 10:50 AM	File folder	
Desktop		CIP Over Ethernet ASFB V2.2	2/25/2019 10:53 AM	File folder	
Documents		Ethernet IP Test	2/26/2019 3:23 PM	File folder	
		📙 Modbus Serial Example	3/1/2019 12:02 PM	File folder	
Kollmorgen Visualization Builder IDE EN (REV 2 22 353) zin		Modbus TCP Example	4/24/2019 4:12 PM	File folder	
		- ModbusSerial	2/24/2019 8:59 AM	File folder	
MON161S3		🔤 Open Modbus TCP ASFB V1.0	2/22/2019 9:28 AM	File folder	
MON161S3.zip		PiCPro V16.1 Monitor Edition	9/13/2018 4:20 PM	File folder	
h Music		PiCPro V16.1 Professional Edition	2/25/2019 10:42 AM	File folder	
P7000_Tools_ 2.11.zip		PiCPro V18.0 Monitor Edition	1/15/2019 12:33 PM	File folder	
PiCProMonV16_1.zip		PiCPro V18.0 Professional Edition	2/19/2019 6:01 PM	File folder	
Dictures		Applications16_1.zip	6/13/2014 2:52 PM	Compressed (zipp	374,330 KB

I created a new subfolder called ModbusSerial_ASFB in the C:\G&L Motion Control Data folder to group the required files using the method described in the manual shown below.

Remember the *.LIB files and *.LDO source files must reside in the same folder.

1.3 Installation

The **Modbus** software disk contains the files listed below. The Main group includes the ASFB library (LIB), source ladders for the ASFBs (LDOs), and remark files containing the comments in the source ladders (.REMs). The Example group includes the example LDO and REM files. The Auxiliary group contains the LIB, LDOs, and REMs for the UDFBs used in the source ladders for the ASFBs. NOTE: It should never be necessary for you to access any of the files in the Auxiliary group. The LIB is required in order for the ASFB to work and the LDOs allow you to view the source ladders when troubleshooting if necessary.

Follow the guidelines found at the beginning of the manual. Always make a back up copy of the disk and store the original in a safe place. The recommended destination' directory for each file is listed in the last column.

Group	File	Description	Directory
Main	C_MOD.LIB	The library containing the application specific function block used to perform Modbus communications.	ASFB
	C_MODSLV.LDO	The source ladder for the transceiver function block.	ASFB
	C_MODSLV.REM	The remark file for the source ladder	ASFB
	C_MODMST.LDO	The source ladder for the C_MODMST function block.	ASFB
	C_MODMST.REM	The remark file for the source ladder	ASFB
Example	C_MASTEX.LDO C_MASTEX.REM	The example for Modbus master LDO from which you can build a new application LDO or to which you can merge an existing one.	Working
	C_MODEX.LDO C_MODEX.REM	The example for Modbus slave LDO from which you can build a new application LDO or to which you can merge an existing one.	Working
Auxiliary	C_MODAUX.LIB	The library that holds all the function blocks used in the source ladder for the ASFB.	ASFB
	C_MODCRC.LDO C_MODCRC.REM	Source ladder Remark file	ASFB ASFB
	C_MODMEM.LDO C_MODMEM.REM	Source ladder Remark file	ASFB ASFB
	C_MODMOV.LDO C_MODMOV.REM	Source ladder Remark file	ASFB ASFB
	C_MODPAK.LDO C_MODPAK.REM	Source ladder Remark file	ASFB ASFB
	C_MODUNP.LDO C_MODUNP.REM	Source ladder Remark file	ASFB ASFB

The 14 files for the "ASFB" folder are shown below.

3ndu601_ND0.zip	↑ Name	Date modified	Туре	Size
AKD BASIC Servo Drive Firmware (AKD-T-ICAN-01-17-00-000).zip	C MODUNP.re	m 2/24/2019 9:16 AM	REM File	2 KB
ascii_sr600	C_MODUNP.Id	o 2/24/2019 9:16 AM	PiCPro Ladder File	2 KB
ascii_sr600.zip	C_MODSLV.ren	2/24/2019 9:16 AM	REM File	13 KB
Desktop	C_MODSLV.Ido	2/24/2019 9:16 AM	PiCPro Ladder File	17 KB
Documents	C_MODPAK.rer	n 2/24/2019 9:16 AM	REM File	1 KB
	C_MODPAK.Ide	2/24/2019 9:16 AM	PiCPro Ladder File	2 KB
	C_MODMST.re	m 2/24/2019 9:16 AM	REM File	10 KB
Kolimorgen visualization Builder IDE EN (REV 2.22.353).zip	C_MODMST.Id	o 2/24/2019 9:16 AM	PiCPro Ladder File	16 KB
MON161S3	C_MODMOV.re	m 2/24/2019 9:16 AM	REM File	1 KB
MON161S3.zip	C_MODMOV.Id	o 2/24/2019 9:16 AM	PiCPro Ladder File	2 KB
Music	C_MODCRC.re	m 2/24/2019 9:16 AM	REM File	3 KB
P7000_Tools_ 2.11.zip	C_MODCRC.Id	2/24/2019 9:16 AM	PiCPro Ladder File	4 KB
PiCProMonV16 1.zip	C_MODAUX.LI	2/24/2019 9:16 AM	Object File Library	5 KB
Pictures		2/24/2019 9:16 AM	Object File Library	36 KB
Videos				
SYSTEM & DATA (C:)				

In this application the Digital MMC Drive Resident Control will be the Modbus Slave.

Example	C_MASTEX.LDO C_MASTEX.REM	The example for Modbus master LDO from which you can build a new application LDO or to which you can merge an existing one.	Working
	C_MODEX.LDO C_MODEX.REM	The example for Modbus slave LDO from which you can build a new application LDO or to which you can merge an existing one.	Working

I saved the C_MODEX.LDO and C_MODEX.REM files to the "Working" directory which will be the Modbus Serial Example folder created at the beginning of this application note.



These files can be renamed if desired but I left them as is.

Next I created a project also saved to the Modbus TCP Example folder.

Once the project is created the following paths were setup.

The MAIN.LDO is the C_MODEX.LDO copied into the Modbus Serial Example "working" directory.

The PicPro Library paths point to the standard libraries for this version of PicPro, the ModbusSerial_ASFB folder (created just previously), and the standard ASFB library for this verion of PicPro (part of the Applications disk install).



Under the View pull-down menu->Hardware Declarations I configured this for the hardware of my demo. This will depend on your hardware.

	r A
ile Edit Tools Help	
► Hardware	- I/O Expansion Option
Master Rack Slot 1 Empty Slot 2 MMC Digital Resident 16 Axis Servo 1.3M Application Slock 1 In/Out 24V DC (8 In 8 Out Src.) Block 2 Empty Block 3 Empty Block 5 Empty Block 5 Empty Block 6 Empty Block 8 Empty Block 9 Empty Block 10 Empty Block 11 Empty Block 12 Empty Block 13 Empty Block 13 Empty Block 15 Empty Block 15 Empty	 Master Rack Only Remote I/O Block I/O I/O Config/Scan Options None Reconfigurable Block I/O and continue to scan with Master Rack, Remote Rack or Block I/O failures Reconfigurable Block I/O and continue to scan with Master Rack, Block or ASIU failures Pool Memory Option Extend Pool Memory

Note there are serial port settings in the Software Declarations.

I left them as is. This may vary depending on your hardware.

Software Declarations : Main Ladder - [C_MODEX.LDO]

File Edit Tools Help

Name	Туре	Α.	I/O Point	Initial Value	Long Name
REV00	BOOL				Example\.LDO\Revision\Num
SLV1	<fb>C_MODSLV</fb>				Modbus\Slave\Driver\
BOOLS	BOOL(01)				Booleans:\On Modbus:\00001 to
INTEGER	INT(099)				Integers:\On Modbus:\40001 to\
PORTADDR	STRING[15]			USER:\$00	Seria/Port\Name\
PORTCFG	STRING[15]			9600,E,8,1,N\$00	Seria/Port/Configure/String
RCMD	STRUCT				Received\Commands\Informatio\
ADDRESS	USINT				Address of\Incomming\Comman
.FUNCTION	USINT				Function\Number of\Incomming\
	END_STRUCT				
MOD_OK	BOOL				init OK\flag\\
MOD_FAIL	BOOL				init\failure\flag\
MOD_ERR	INT				init\failure\error code\
MOD_MSG	BOOL				message\received\flag\
EXCEPT	BOOL(07)				User\Definable\Exception\Status
MOD_TERR	BOOL				trans\action\error flag\

Here is the Modbus Slave function block. Once downloaded we will animate it.

C_MODSLV	C_MODSLV
Communications- Modbus slave function block Allows the PiC900 to be used as a slave with the Modbus protocol.	EN OK ADDR FAIL PORT ERR CFG RCMD BOOL BSIZ DATA DSIZ EXPT R ASCI

See the Modbus ASFB manual for more details on the I/O of this function block.

A topic of interest is the input for BOOL (type of data) which starts at BOOLS[0] and the input BSIZ indicates the size of the BOOLS array which is 2. This can be expanded (later).

H Network #4 This function	on er	ables	the PiC9	00 to be a Modbu	s Slave ov	ver an RS232 co	onnection.
		- SLV1 - C_M	ODSLV				
		EN05	ок	MOD_OK 			
1	≻.	ADDR	FAIL				
PORTADDR	≻.	PORT	ERR	→ MOD_ERR MOD_MSG			
PORTCFG	≻.	CFG	RCMD				
BOOLS[0]	≻.	BOOL	TERR	·····()			
2	≻_	BSIZ	CODE	$\rightarrow MOD_CODE$			
INTEGER[0]	≻.	DATA					
100	≻.	DSIZ					
EXCEPT[0]	<u>≻-</u>	EXPT					
RCMD	≻.	R					•
		ASCI					

Prior to downloading the sample ladder here are details on the hardware used in this example and the wiring scheme. Refer to the appropriate hardware/installation manual for your specific hardware.

For serial Modbus the User Serial Port (C3) is used on the Drive Resident Digital MMC Control.



Figure 13-1: The Drive Resident Digital MMC Control

Although the C3 port has both RS232 and RS485 interface capability for this demonstration I used RS232. I used a USB to Serial converter and a DSUB9 connector to DSUBHD15 cable to make the serial connection between my PC and the User Port. Check your hardware manual for your PC or HMI to determine what your connections should be.

	PC		User Port
Pin	Description	Pin	Description
3	ТХ	9	RX
2	RX	10	ТХ
5	GND	8	GND

	Table 13-9: User Port Pin Assignments					
Pin	Signal	In/Out	Connector Pinout			
1	NC	N/A				
2	N/C	N/A	Ţ			
3	N/C	N/A	Ţ			
4	RS232 Data-terminal-ready (12 Vdc)	Out				
5	RS232 Request-to-Send	Out	15-pin HD male D-sub			
6	N/C	N/A				
7	RS232 Clear- to-Send	In	11/01			
8	Signal Ground	In/Out	000			
9	RS232 Receive Data	In				
10	RS232 Transmit Data	Out	15005			
11	N/C	N/A				
12	RS485 Receive Data +	In	Ţ			
13	RS485 Receive Data -	In	Ī			
14	RS485 Transmit Data +	Out	Ī			
15	RS485 Transmit Data -	Out	1			
Connector Shell	Drain	In				



This is consistent with what is offered in the Serial Modbus ASFB manual:

Cable connections

The **pinouts** for the various **Modbus** communications connections are shown below. Choose the one for your system.

PiC900 to an operator interface

(1 O-pin sc	rew terminal)	(25-pin female)		
GND RECV TRANS	8 9 10	7 2 3	GND TRANS RECV	
TICH (5	10		NE C V	

PiC900 to a PC

PIC900 CPL	to	а	PC
(1 O-pin screw	terminal)	(9-	pin female)
GND 8		5	GND
RECV 9		3	TRANS
TRANS 10		2	RECV

Finally, I saved the ladder changes and project and then compiled and downloaded it to the control and animated. In my case I used Ethernet to connect to the Digital MMC Drive Resident Control so my serial connection (USB to Serial in my case) would be used for Serial Modbus and to view the PicPro ladder online I used Ethernet.

PRJ	IP: 192.168.0.80

H Network #4 This function enables the PiC900 to be a Modbus Slave over an RS232 connection. - SLV1 -C_MODSLV MOD OK EN05 ОК **(**s)-4 MOD_FAIL 1 ADDR FAIL -(<mark>s</mark>)-> \neg 'USER:\$00' +0 PORTADDR PORT ERR → MOD_ERR \rightarrow '9600,E,8,1,N\$00' MOD_MSG PORTCFG > CFG RCMD \leftarrow \rightarrow \rightarrow MOD_TERR NO BOOLS[0] BOOL TERR $\langle \rangle$ \geq +0 2 BSIZ CODE → MOD_CODE \geq +0 INTEGER[0] \geq DATA 100 DSIZ \geq NO EXCEPT[0] > EXPT RCMD R > ASCI End of Module

Note the E_MODSVR is enabled, SCON_OK and error outputs are all zero.

Next using Modbus Poll as the Modbus Serial master and to check if values can be written.

RTU mode is selected as follows:

onnection Setup		×
Connection Serial Port	~	OK
Serial Settings		Cancel
USB Serial Port (COM2)	~	Mode
9600 Baud \sim		● RTU ○ ASCII
8 Data bits 🛛 🗸		Response Timeout 1000 [ms]
Even Parity 🛛 🗸		Delay Between Polls
1 Stop Bit 🛛 🗸	Advanced	20 [ms]
Remote Modbus Server		
IP Address or Node Name		
192.168.0.80		~
Server Port	Connect Timeout	IPv4
502	3000 [ms]	◯ IPv6

RTU mode is selected by the C_MODSLV block as follows:

	•	SLV1 -		· · · · · · · · · · · · · · · · · · ·
		СМ	ODSLV	
	•			. MOD_OK
		EN05	ОК	(s)
	•			MOD_FAIL
1	\succ	ADDR	FAIL	(s)(
'U SER:\$00'				+0
PORTADDR	\succ	PORT	ERR	\rightarrow MOD_ERR
'9600,E,8,1,N\$	00'			MOD_MSG
PORTCFG	\succ	CFG	RCMD	<u>├</u> ───()──
NO				MOD_TERR
BOOL S[0]	\succ	BOOL	TERR	<u>├</u> ───()──
				+0
. 10	\succ	BSIZ	CODE	\rightarrow MOD_CODE
+1234				
INTEGER[0]	\succ	DATA		
. 100	≻.	DSIZ		
NU		EVOT		
EXCEPT[0]	<u> </u>	EXPT		
DCMD				
. RCMD	<u> </u>	ĸ		
If not not	-	ASCI		
then RTU	-	1301		
mode				

Per the Modbus Slave ASFB the PORTCFG input is for 9600, E, 8, 1 so Modbus Poll is configured with the same settings:

onnection Setup		\times
Connection		ОК
Serial Port	~	Cancel
USB Serial Port (COM2)	~	Mode
9600 Baud \sim		● RTU ○ ASCII
8 Data bits 🛛 🗸		Response Timeout
Even Parity \sim		Delay Between Polls
1 Stop Bit 🛛 🗸	Advanced	20 [ms]
Remote Modbus Server		
IP Address or Node Name		
192.168.0.80		~
Server Port	Connect Timeout	IPv4
502	3000 [ms]	⊖ IPv6

Next In Modbus Poll setup a Read/Write Definition. A very important setting is the Slave ID. This value needs to match the input value of ADDR in the C_MODSLV function block in Picpro. In this example it is 1.

	PiCPro Professional Edition - [Main Ladder - [C_MODEX.LDO]]						
Į	File Edit Online Cor	mpile Ladder	View Window	Help			
		₿ ? №	97 🔓				
		📲 🛃 🌌	1				
	A_DT_T 💽 🔐				_		
	_ SLV			Read/Write Definition	×		
•			мор_ок	Slave ID: 1 OK			
•	· · · · ·		MOD_FAIL	Function: 16 Write Multiple Registers V			
•	1 ADD 'USER:\$00'	R FAIL -	(\$) . +0	Address: 0 Protocol address. E.g. 40011 -> 10			
•	PORTADDR - POR '9600,E,8,1,N\$00'	T ERR	\rightarrow MOD_ERR MOD_MSG	· Quantity: 2			
		RCMD .		Scan Rate: 40 [ms] Apply			
			()	Disable			

Initially the starting address will be zero and not using Base 1. If you want the convention to follow the chart in the Modbus Serial ASFB manual then the PLC Addresses (Base 1) checkbox needs to be checked. With the first test we will attempt to write to the first to INTEGERS in the G&L.

Read/Write	Definition			\times	
Slave ID:	1			OK	
Function:	16 Write Multiple Reg	gisters 🗸 🗸		Cancel	
Address: Quantitur	0 Protoco	laddress. E.g	4001	1 -> 10	
Scan Rate: Disable Read/	40 [ms] Write Disabled e on error		Rea	Apply d/Write Once	
View Rows 10 0 20 0 50 0 100 0 Fit to Quantity					
Hide Alias Columns PLC Addresses (Base 1) Address in Cell Enron/Daniel Mode					

On connection there are no errors so for demonstration purposes I set the first 2 registers to values 1234 and 5678 respectively.

💬 Mbpoll1		
Tx = 4345: Err	= 1: ID = 1: F =	16: SR = 40ms
Alia	as 00000	
0	1234	
1	5678	
2		
3		
4		
5		
6		
7		
8		
9		

Now switching back to the animated ladder in PicPro I used the View List to monitor the values.

Enable	Name	Туре	Value	
X	INTEGER[0]	Signed Decimal	+1234	
X	INTEGER[1]	Signed Decimal	+5678	
				~

Keep in mind some Modbus TCP masters start their address at 1, 40001, 400001, etc. where 1 will likely be equivalent to DATA[0] which is an offset in addressing.

The following yields the same result (Base 1 addressing):

Read/Write	Definition		×
Slave ID:	1		OK
Function:	16 Write Multiple Re	gisters 🗸	Cancel
Address:	1 Protoco	I address. E.g.	40011 -> 10
Quantity:	2		
Scan Rate:	40 [ms]		Apply
Disable Read/	Write Disabled e on error		Read/Write Once
View Rows ① 10	○ 20 ○ 50 ○	100 🔿 Fit to) Quantity
Hide A	ulias Columns ss in Cell	PLC Addr	esses (Base 1) aniel Mode

Note the Modbus TCP ASFB manual assumes a base 1 addressing scheme. Also note the array is shown as BOOL(x) in the chart but in the sample project the array is named BOOLS(x). Likewise the integer array is shown as DAT(x) in the chart but INTEGER(x) in the sample project.

Message Addressing

BOOL	EANS	INTEGERS		
Modbus	PiC900	Modbus PiC90		
00001	BOOL(0)	40001	DAT(0)	
00002	BOOL(1)	40002	DAT(1)	
•				
00999	BOOL(998)	40999	DAT(998	

The addressing between the PiC900 and Modbus is as follows:

To test the Boolean data I setup Modbus Poll to use function 15-Write Multiple Coils.

Read/Write	Definition			×
Slave ID:	1			OK
Function:	15 Write Multiple Coi	ls ~		Cancel
Address:	0 Protoco	I address. E.g.	10011	-> 10
Quantity:	2			
Scan Rate:	40 [ms]			Apply
Disable Read/	Write Disabled e on error		Read	I/Write Once
View Rows	○ 20 ○ 50 ○	100 🔿 Fit to	Quan	tity
Hide A	ulias Columns ss in Cell	PLC Addr	esses (iniel Mi	Base 1) ode

To demonstrate I added BOOLS[0] and BOOLS[1] to the View List in PicPro. NO=0 or OFF and YES=1 or ON.

Enable	Name	Туре	Value
х	INTEGER[0]	Signed Decimal	+1234
X	INTEGER[1]	Signed Decimal	+5678
Х	BOOLS[0]	Boolean	NO
X	BOOLS[1]	Boolean	NO

Turning on BOOLS[0] in Modbus Poll:

ත් Modbus Poli - Mbpoli1	_	×	PiCPro Pro	fessional Editio	on - [View List: C_N	IODEX.rtd]	
File Edit Connection Setup Functions Display View Window Help		ľ.	🔳 File Edit	Online Co	mpile Ladder \	/iew Window	Help
🗅 📂 🖬 🎒 🗙 🛅 🖳 🏛 💷 05 06 15 16 17 22 23 TC 🗵 💡 😵				X 🖻 🖻	₿ ? № 5	1 1	
			🕶 🔛 🚯	3 3 3	Ha 🛃 🏂 🕺	2 👼	
Tx = 14447: Err = 0: ID = 1: F = 15: SR = 40ms			A_DT_T	•			
Alias 00000			Enable	Name	Туре	Value	
0 1			X	INTEGER[0]	Signed Decimal	+1234	
]	X	INTEGER[1]	Signed Decimal	+5678	
			X	BOOLS[0]	Boolean	YES	
2			X	BOOLS[1]	Boolean	NO	<u> </u>
3							
4							
5							
6							

Turning on BOOLS[1] in Modbus Poll:

🛍 Modbus Poll - Mbpoll1	_	\times	FiCPro Pro	fessional Editio	on - [View List: C_MO	DEX.rtd]	
File Edit Connection Setup Functions Display View Window Help			File Edit	Online Cor	mpile Ladder Vie	w Window	Help
🗅 🗃 🖬 🎒 🗙 🛅 🗒 🏩 📖 🕮 05 06 15 16 17 22 23 TC 🛛 💡 🦞				X 🖻 🖻	a 🔋 <table-cell> 💖</table-cell>	6	
				3 3	19 1 9 19 19	a	
1x = 16031: Err = 0: ID = 1: F = 15: SR = 40ms			A_DT_T	- 2 :			
Alias 00000			Enable	Name	Туре	Value	
			X	INTEGER[0]	Signed Decimal	+1234	
] <u>x</u>	INTEGER[1]	Signed Decimal	+5678	
			х	BOOLS[0]	Boolean	NO	
			X	BOOLS[1]	Boolean	YES	
3						[]	
4			d				
5							
6			a				
7							
8							

Note there are 100 INTEGERS in the setup of the sample Modbus Serial ASFB but only 2 BOOLS.

You can expand it by changing the BSIZ from 2 to X within the limit of the size of the BOOLS array (0..998).

For example:



It is important to note that the value of BSIZ must match the size of the array of BOOLS

Software Declarations : Main Ladder - [C_MODEX.LDO]

File Edit Tools Help

	Name	Туре	Α.	I/O Point	Initial Value	Long Name
	BOOLS	BOOL(01)				Booleans:\On Modbus:\00001 to
	INTEGER	INT(099)				Integers:\On Modbus:\40001
	PORTADDR	STRING[15]			USER:\$00	Seria/Port\Name\
	PORTCFG	STRING[15]			9600,E,8,1,N\$00	Seria/Port/Configure/String
Γ	RCMD	STRUCT				Received\Commands\Informatio\

Name	Туре	A. I/O Point	Initial Value	Long Name
BOOLS	BOOL(01)			Booleans:\On Modbus:\0000
INTEGER	INT(099)	Help		Integers:\On Modbus:\40001 to\
PORTADDR	STRING[15]	Insert		Serial\Port\Name\
PORTCFG	STRING[15]	Make Array	,N\$00	Serial/Port/Configure/String
RCMD	STRUCT	Modify Attributes	>	Received\Commands\Informatio\
ADDRESS	USINT	Purge Unused		Address of\Incomming\Comman
FUNCTION	USINT	Delete		Function\Number of\Incomming\
	END_STRUCT			
MOD OK	BOOL			init OK\flag\\

To extend the array size, highlight the BOOLS Type and right-click and then choose Make Array.

In this example I extended the BOOLS from 0..1 to 0..9 or from 2 to 10 elements. So input 10 into the Enter Array Size box and then the Enter button on your PC's keyboard.

Software Declarations : Main Ladder - [C_MODEX.LDO]

Software Declarations : Main Ladder - [C_MODEX.LDO]

	_		_			
Fil	e Eo	dit	00	s	He	lp

Name	Туре	Α.	I/O Point	Initial Value	Long Name
BOOLS	BOOL(01)				Booleans:\On Modbus:\0000
INTEGER	INT(099)				Integers:\On Modbus:\40001 to\
PORTADDR	STRING[15]			USER:\$00	Seria/Port/Name/
PORTCFG	STRING[15]			9600,E,8,1,N\$00	Serial/Port/Configure/String
RCMD	STRUCT				Received\Commands\Informatio\
ADDRESS	USINT				Address of\Incomming\Comman
.FUNCTION	USINT				Enter Array Size: 10 mming\
	END_STRUCT				,
MOD_OK	BOOL				init OK\flag\\
MOD FAIL	BOOL				init\failure\flan\

Software Declarations : Main Ladder - [C_MODEX.LDO]

File Edit Tools Help

Name	Туре	Α.	I/O Point	Initial Value	Long Name
BOOLS	BOOL(09)				Booleans:\On Modbus:\0000
INTEGER	INT(099)				Integers:\On Modbus:\40001 to\
PORTADDR	STRING[15]			USER:\$00	Seria/Port\Name\
DODTOEO	CTDIMO1461			0000 E 0 4 NE00	Social/Doct/Coofigure/String

Save and Close the Software Declarations. Another addition will be a departure from monitoring using the View List and add the capability to viewing the values in the ladder itself. Using the MOVE function block as shown:



I added the two MOVE blocks, saved the ladder, and the performed a Compile&Download.

Testing BOOLS from Modbus Poll:

·					월 Modbus Poll - Mbpoll1
				, i	File Edit Connection Setup Functions Display View Window Help
			MOVE		🗅 🚔 🖨 🎒 🗙 🛅 🗒 🚊 🕮 05 06 15 16 17 22 23 TC 🖻 💡 😵
		EN	ок	-	
YES BOOLS[0] NO	≻.	IN1	OUT1	-	Tx = 1571: Err = 0: ID = 1: F = 15: SR = 40ms
BOOLS[1]	≻-	IN2	OUT2	-	Alias 00000
BOOLS[2]	≻.	IN3	OUT3	-	
BOOLS[3] YES	≻.	IN4	OUT4		
BOOLS[4]	≻-	IN5	OUT5	-	4 1
BOOL S[5] YES	≻-	IN6	OUT6	-	6 1
BOOL S[6]	≻-	IN7	OUT7	-	
BOOLS[7]	≻.	IN8	OUT8		9 0
BOOLS[8]	≻.	IN9	OUT9		
BOOLS[9]	≻.	IN10	OUT10	-	
		l		Ļ	

Testing INTEGER using Modbus Poll:



Using KVB with Modbus Serial

Next I will demonstrate using KVB software to do the same thing as Modbus Poll.

I started by creating a new project and selecting Kollmorgen->Modbus Master RTU/TCPIP as shown below. Note this communication driver supports both Serial (RTU) Modbus and Ethernet based Modbus TCP.

Choose Target	Choose Controller	Select Location
Choose your target in the menu	Choose your preferred controller	Select the location of your project
below	or OPC server in the menu below	in the menu below
Controllers Select brand DEMO G & L Motion Control Kollmorgen OPC UA Server URL:	Select protocol	Next > < Previous Finish Cancel

Under Tags->Controllers->Settings:

Project Explorer 🗸 🗘 🗙	Screen1 × Tags ×			
Screens (1) All Screens	🔁 Tags			
	Tags Controllers Triggers Poll Groups Index Registers			
	Home			
Screen1				
	Add Delete		Controlle	Snow Selection •
	Name	ID	Active	
	> Controller1			₹
0-0-0				
Search 🛞				
 Functions (4) 				
Alarm Server				
Multiple Languages				
Security				
Tags				

Under the Settings tab I changed the Communications mode to Serial. As with Modbus Poll, the ID or in this case called Default Station needs to be set to the same number as in the C_MODSLV's ADDR input value.

PiCPro Professional Edition - [Main Ladder - [C_MODEX.LDO]]	– 🗆 X
I File Edit Online Compile Ladder View Window Help	_ & ×
	Modbus Master RTU/TCPIP ×
	Settings Stations
	Modbus Master RTU/TCPIP 5.02.02
	Settings
	Communication mode Serial
MODIOK	Default station 1
	Modbus protocol RTU
MOD FAIL NO	32-bit word mapping Motorola
	Addressing Decimal
115FR-500' +0 NO	Start address 0-based
	2 Max block size (words) 16
'9600.E.8.1.N500' MOD MSG NO	Force function code 0x10 Disable
	3 - String swap Enable
NO MOD_TERR NO	Silent time (ms) 20

I left the Start address as the default 0-based. If you want the addressing to look like the table in the Modbus Serial ASFB manual you need 1-based but in this example I left it at 0-based because I like the Modbus address count to be the same as the index of the INTEGER[x] array. I also added 20msec of silent time.

Modbus Master RTU/TC	PIP			\times			
Settings Stations							
Modbus Master RTU/TC	PIP 5.02.02						
Settings							
Communication mode		Serial					
Default station		1					
Modbus protocol		RTU					
32-bit word mapping		Motorola					
Addressing		Decimal					
Start address		0-based	0-based				
Max block size (words)	16					
Force function code 0	x10	Disable					
String swap		Enable					
Silent time (ms)		20					
Clock register		40000					
Communication me	ethod						
Communication metho	d	Serial		•			
	OK	Cancel	Apply	Help			



As mentioned before, RTU mode is select in the C_MODSLV as follows:

Scrolling down under Communications Method I selected Serial and set the serial port settings to match that of my PC (which I am using as the Modbus Serial Master) This will need to be set according to your hardware that you are using for the Modbus Serial Master.

Modbus Master RTU/TCPIP

Communication method Communication method								
Communication method								
	Serial							
Port	COM1							
Baud	9600							
Parity	Even 💌							
Data bits	8							
Stop bits	1							
Tunneling protocol	TCP							
Tunneling IP address	192.168.98.2							
Tunneling port	6004							
Tunneling local UDP port	0							
E Advanced								
E Routing	•							
Parity Set the parity for the selected COM port.								

The Stations tab is for Ethernet based communications so I left it at default.

 \times

Next I created 4 tags 2 of the BOOL type and 2 of the INT16 type. Note the addressing is 0-based.

Screen1 × Tags ×									
👌 Tags									
Tags Controllers Triggers	Poll Groups Ind	dex Registers							
Home									
	Column	is Visible							Filter
Add 👻 Delete	▼ ✓ Scali	ing [a Exchange	✓ Others						
Add Delete Tag	▼ Scali	ing 🛛 🗄	Others Controllers		Scaling				Others
Add • Delete Tag Name	Scali Data	ing Exchange	Others Controllers Data Type	Controller 1	Scaling Offset	Gain	Read Expr	Write Expr	Others Description
Add Delete Tag Name BOOLS_0	Cali Data Type BOOL	ing Exchange	Others Controllers Data Type BIT	Controller 1 00000	Scaling Offset 0	Gain 1	Read Expr	Write Expr	Others Description
Add Delete Tag Name BOOLS_0 BOOLS_1	Data Type BOOL BOOL	ing [a Exchange Access Right ReadWrite ReadWrite	Controllers Data Type BIT BIT	Controller 1 00000 00001	Scaling Offset 0	Gain 1	Read Expr	Write Expr	Others Description
Add Delete Tag Name BOOLS_0 BOOLS_1 INTEGER_0	Cali Cali Data Data Type BOOL BOOL INT16	ing 5 a Exchange Access Right ReadWrite ReadWrite ReadWrite	Controllers Data Type BIT BIT INT 16	Controller 1 00000 00001 40000	Scaling Offset 0 0 0 0	Gain 1 1 1	Read Expr	Write Expr	Others Description

Next for Screen 1 I added 2 Buttons and 2 Analog Numerics. I changed the labels on the buttons accordingly and put text labels above the 2 Analog Numerics.

	INTEGER[0]
BOOLS[0]	#
	INTEGER[1]
BOOLS[1]	#

Next it is required to associate a tag and action to each graphic on the screen.

First with the BOOLS[0] button. Set the Mouse Down and Mouse Up as shown so the operation of the button is on press BOOLS[0] will be 1 or ON and on release BOOLS[0] will be 0 or OFF.

100								
J	General Actions							
	Select Action	•	Set Analog	•	Set Analog	•		
			BOOLS_0	•	BOOLS_0	•		
	Mouro Lonvo	5	1 Mouro Down	E.	0 Mouse Lie	5		
1	INIOUSE Leave		Mouse Dowit		Mouse op			
T								
								į.
	•							
			102.0		INTEGER[0]			
			F					
		115	BOOLS[0]		#	538.0		
		110.				000.0		
					INTEGER[1]			
			P0015[1]		#			
			BOOLS[1]		#			
			202.0					
			303.0					

Repeat for the BOOLS_1 button:



Next the Analog Numeric fields will be configured. The tag is selected, Integer type, and On Input is setup to limit the input from the HMI to be within a valid range for a 16 bit signed integer.

Home Project System	Insert View I	Dynamics General Actio	ons				
Integer • Zero Fill	Prefix	INTEGER_0 •	Auto Stretch Font		🗹 On Input 🗌 On Display		
Limit Characters to 10	Suffix	Select Security Groups 🔹	Disable Operator Input		Lower Limit -32768		
Number of Decimals 1		Default *	Transparent		Upper Limit 32767		
Disalan Fransk	Prefix/Suffix	Tag / Security	Format	Text Alignment	Validation		
Project Explorer 🚽 🗸 🗙	Screen1 × Tags ×				95		
▼ Screens (1)							
All Screens							
			•				
Screen1							
					102.0	-	
			-		INTEGER[0]	
				POOL 5[0]	#		
				BOOLSTON	- #	271.0	
					INTEGER[1]	1	
				BOOLS[1]	#		
Search							
vearen Q					303.0		
▼ Functions (4)							

Repeat for INTEGER[1]:



After Save, Build, Run....



On press of the BOOLS[0] button in run-time:

On press of the BOOL[1] button in run-time:



From the Modbus Poll the values were set in the INTEGER array. Initially KVB in run-time shows INTEGER[0] as a value of 1 and INTEGER[1] as a value of 2.



Clicking on the 2 data fields and changing the values:



This concludes the Modbus Serial Example.