

Sample Project

Firmware version of the TRM202 on the LCD of the PR200

The objective of this sample project is to provide an example of how string data can be read by the PR200 Programmable Relay and be shown on its LCD display. In this example, the String Data is the TRM202 firmware version that is stored in its Modbus registers. As Modbus Master, the PR200 is supposed to read the firmware string from the corresponding registers over the RS485 serial communication interface.

This document describes devices used in this project, the program structure and its functional principle.

1. Devices and signal types

Device	Signal type	Interface	Modbus settings
TRM202-H2-RI	Modbus RTU	RS485	Address: 6 Register address: 4100 and 4102

There are two jumpers, XP4 and XP5, on the RS485 interface card of the PR200. They can be set either to Master or to Slave:



By default upon delivery, they are set to Slave. To use the interface as Master, the jumpers should set to the M position:



The TRM202 Process Controller is connected to the PR200 over the RS485 interface (Slot 2 is used in this project).

Compatible with
PR200.24.2.2
PR200.230.2.2



The akYtec ALP software is configured as shown in the figures below:

Device configuration		-		×
⊡ Device Display Clock	Data transmission via Modbus over RS485 interface. Line length without repeater up to 1200 m. Up to 16 Slaves can be connected to the Master interface.			
⊡Interfaces	As standard 👺 Factory settings			
	Interface RS485			
Extension modules	Slot number 2			
	Mode Master -			
i⊞ Analog	Protocol Modbus -			
Outputs	Baud rate 9600 💌			
Digital	Parity none 🔻			
	Stop bits 2			
	Data bits 8 V			
	Comment			
]		
	Read	d	Close	
Device configuration		_		×
Device				
Display				
Clock	Name IRM2U2 Address b			
RS485, Slot 2, Master	Time-out (ms) 100 Burst request			
TRM202, 6	Status variable <pre><mo></mo></pre> Start query <pre><mo></mo></pre>			
Extension modules	Change register order			
€…Analog	REAL 2 1 4 3			
i⊞ Digital	Comment			
	h 🔁 📥 🛄 Name	Version_TRM20		Ĥ
	Parameter in FB Type Register address Type Version TRM202 r1 INT 4100	INT	•	
	Version_TRM202_r2 INT 4102 Register	4100		
	Read function	0x03 👻		
	* Write function	0x06 👻		
	No under of maniatam	Write by change		
	Number of registers			
	Start reading	< 012		-
	R	ead	Close	

The variables Version_TRM202_r1 and Version_TRM202_r2 are addressed to registers 4100 and 4102. These registers contain the data of the TRM202's firmware version.



2. Program structure

Each byte of the Version_TRM202_r1 and Version_TRM202_r2 variables is converted to decimal ASCII codes using the **ASCII_Numb** macro. This macro extracts a desired byte of the corresponding register and returns its decimal value that can be processed by the **Dynamic box** display element present in the Display Manager of the akYtec ALP. The macro can be downloaded from the Component Manager (akYtec ALP: Main Menu > Device > Component Manager).

Here is the ASCII table:

Decimal	Hex	Char	Decimal	Hex	Char	J Decima	al Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	0	96	60	`
1	1	[START OF HEADING]	33	21	1	65	41	Α	97	61	а
2	2	[START OF TEXT]	34	22		66	42	В	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	С	99	63	с
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	е
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	1.1	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	н	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	- 1	105	69	i i
10	Α	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	В	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	С	[FORM FEED]	44	2C	,	76	4C	L	108	6C	1
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	1. A.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	1	79	4F	Ο	111	6F	ο
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	Р	112	70	р
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	S
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	т	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	У
26	1A	[SUBSTITUTE]	58	3A	1.1	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	١	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DE

Since the first 45 symbols are not required to represent the firmware version of the TRM202, the **ASCII_Numb** macro takes it into account with the following formula:

Q1 = INT value - 45.

Therefore, the ASCII character "." changes its decimal code from 46 to 1.

To implement the ASCII table in akYtec ALP, the **Dynamic box** display element is used in this sample project:

Compatible with
PR200.24.2.2
PR200.230.2.2



Pro	operties: Dynamic box7		ņ	×
	≜ ↓ □			
~	Coordinates			^
	X:	7		
	Y:	1		
~	Parameters			
	Variable	[SR_CHAR_7]		
~	Row list	< edit >		
	0	*		
	1			
	2	/		
	3	0		
	4	1		
	5	2		
	6	3		
	7	4		
	8	5		
	9	6		
	10	7		
	11	8		
	12	9		
	13	:		۷

First, each converted-to-int byte, the output of the **ASCII_Numb** macro, has to be assigned to the **Variable** parameter of the Dynamic box. Then the ASCII table can be created with the use of the **Row list** parameter:

~	Coordinates	5				
	X:	7				
	Y:	1				
~	Parameters					
	Variable	[SR	_CHAR_7]			
~	Row list	< edi	t >		\sim	
	0	Nr.	ID	Characters	-	1
	1	0	•	1		
	2	- 1				
	3		•	-	- 1	
	4	2	/	1	_	
	5	3	0	1		
	6	4	1	1		
	7	5	2	1		
	8		2		-	1
	9	6	3	1		J
	10	/				
	11	8				
	12	Э				
	13					

Compatible with
PR200.24.2.2
PR200.230.2.2



The whole ASCII table that the **Row list** contain is:

Nr	ID	Nr	ID	Nr	ID	Nr	ID
0	*	21	В	42	W	63	I
1		22	С	43	Х	64	m
2	/	23	D	44	Y	65	n
3	0	24	E	45	Z	66	0
4	1	25	F	46	[67	р
5	2	26	G	47	١	<mark>68</mark>	q
6	3	27	Н	48]	69	r
7	4	28	I	49	۸	70	s
8	5	29	J	50	_	71	t
9	6	30	K	51	``	72	u
10	7	31	L	52	а	73	V
11	8	32	М	53	b	74	W
12	9	33	N	54	с	75	х
13	:	34	0	55	d	76	У
14	;	35	Р	56	е	77	z
15	<	36	Q	57	f	78	*
16	=	37	R	58	g		
17	>	38	S	59	h		
18	?	39	Т	60	i		
19	@	40	U	61	j		
20	A	41	V	62	k		

The Nr. column must be filled in with decimal ASCII codes. Then these codes are supposed to be compared with the current value of the **Variable** parameter. When the match is detected, the corresponding value of the **ID** column will be showed on the PR200's display. If the value is outside of the range, a default character "*" will be shown:

V	е	r	s	i	0	n						
*	*	*	*	*	*	* (*	•				

Compatible with
PR200.24.2.2
PR200.230.2.2



3. Variable Table

The following figures list all the variables used in this sample project.

Local variables:

Select a variable or create a new one									
Parameter in FB 🛛 🔻	Data type		Persistence	Default value	Used in project	Comment	ي ک		
SR_CHAR_7	INT	-		0	Yes		anda		
SR_CHAR_6	INT	-		0	Yes		ā		
SR_CHAR_5	INT	-		0	Yes		Se l		
SR_CHAR_4	INT	-		0	Yes		1 vice		
SR_CHAR_3	INT	-		0	Yes		H R		
SR_CHAR_2	INT	-		0	Yes		S-4		
SR_CHAR_1	INT	-		0	Yes		5.0		
SR_CHAR_0	INT	-		0	Yes		lot 2		

Network variables (Slot 2):

Select a network variable or create a new one											
TRM202, 6											Sta
Name	me Data type		Read function		Write function		Register address	Bit number	Comment		ndard
Version_TRM2	INT	-	0x03	•	0x06	•	4100				S.
Version_TRM202_r2	2 INT	-	0x03	-	0×06	-	4102				Vice
< no >	BOOL	-	0x01	-	0×05	-	0	0			
-											IS-485, Slot 2