

Compatible with
PR200.24.2.2
PR200.230.2.2

Sample Project

Firmware version of the TRM202 on the LCD of the PR200

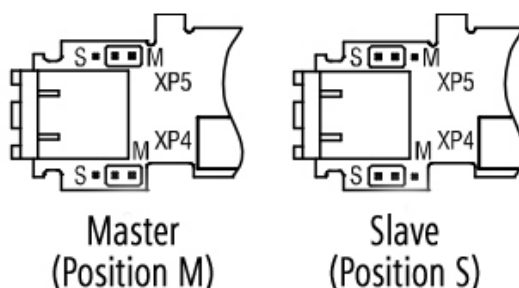
The objective of this sample project is to provide an example of how string data can be read by the PR200 Programmable Relay and be shown on its LCD display. In this example, the String Data is the TRM202 firmware version that is stored in its Modbus registers. As Modbus Master, the PR200 is supposed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



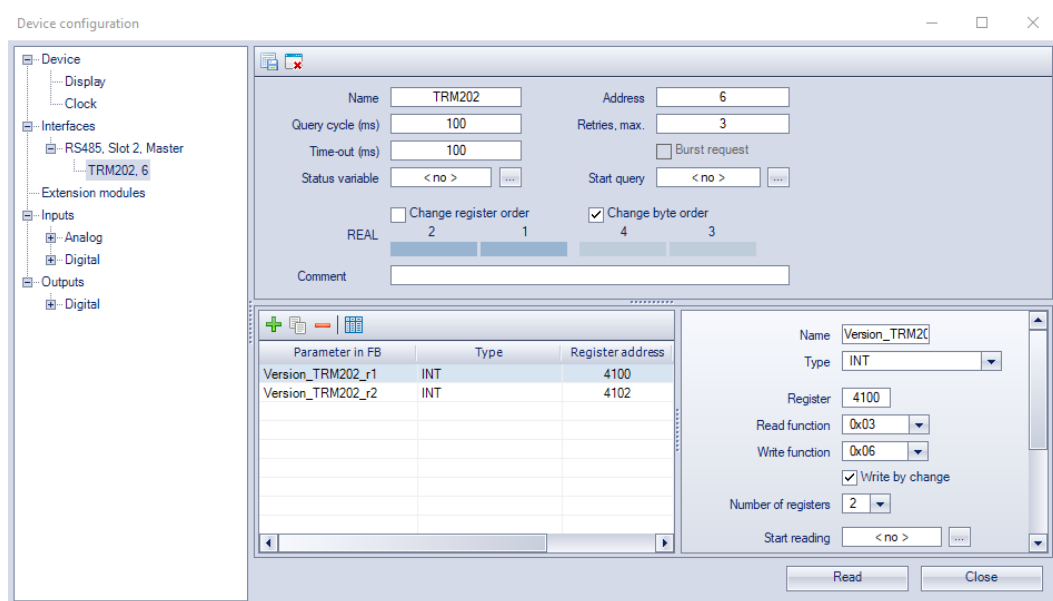
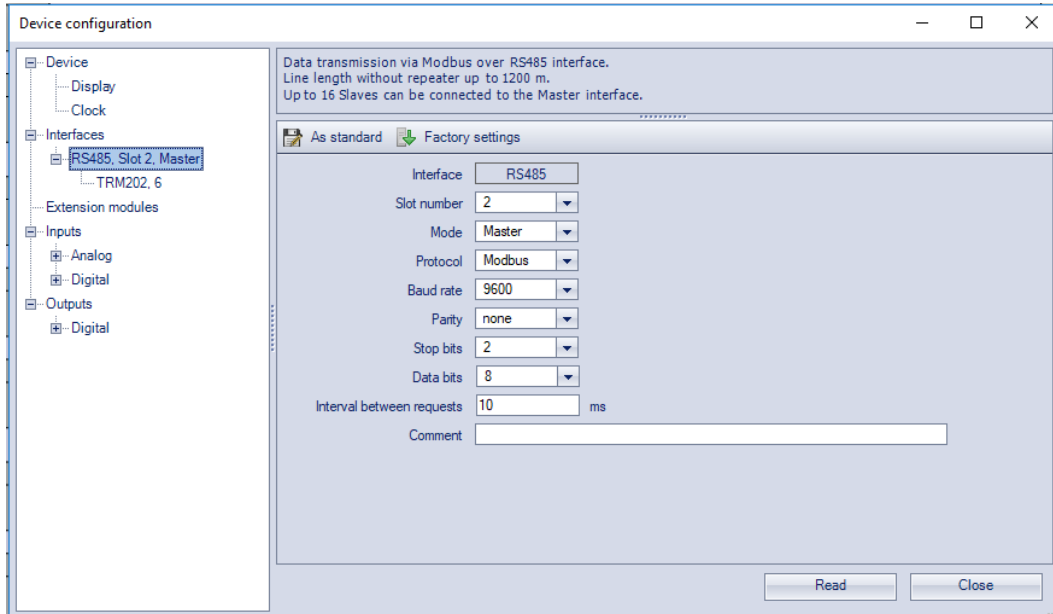
By default upon delivery, they are set to Slave. To use the interface as Master, the jumpers should be set to the M position:



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:



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.

Compatible with
PR200.24.2.2
PR200.230.2.2

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:

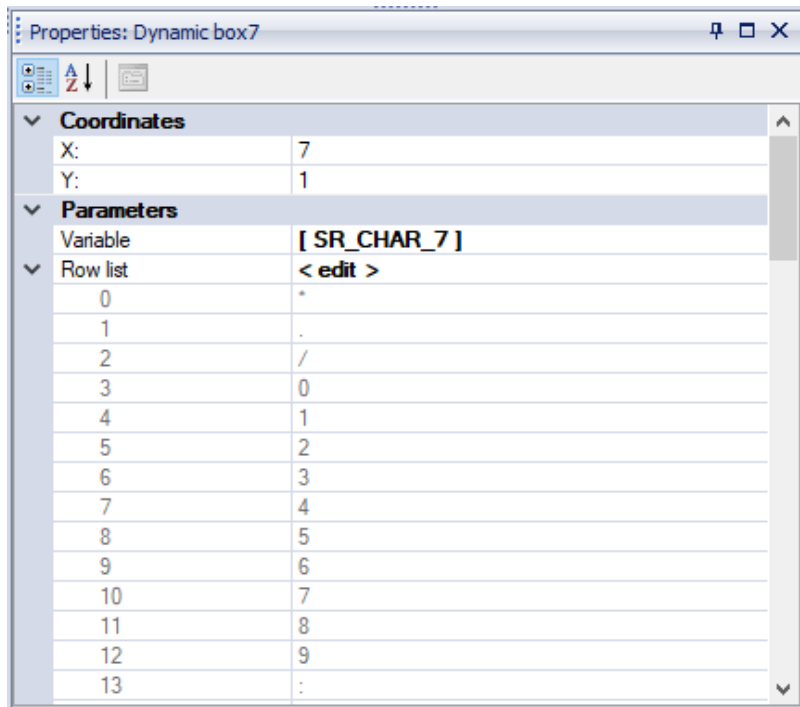
Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
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	T	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	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	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:

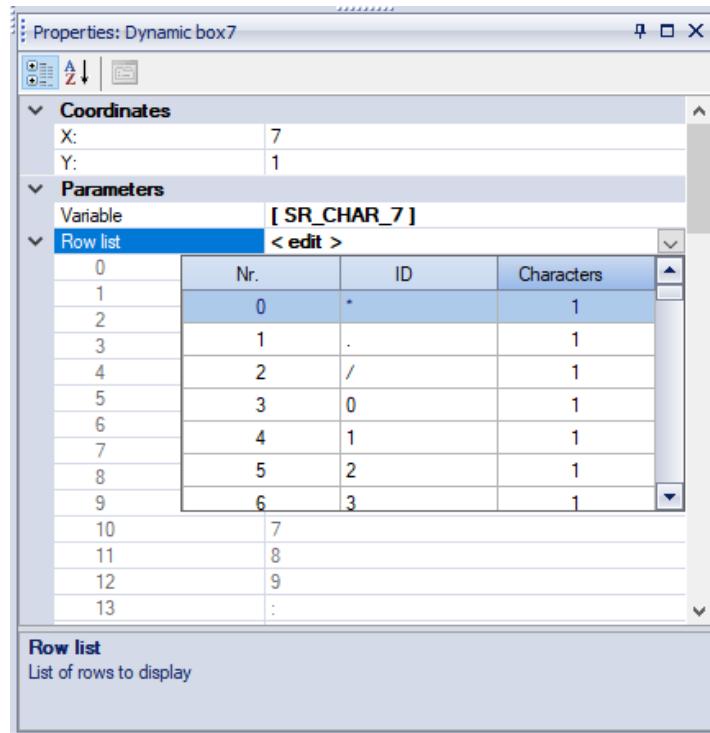
$$Q1 = \text{INT value} - 45.$$

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:



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:

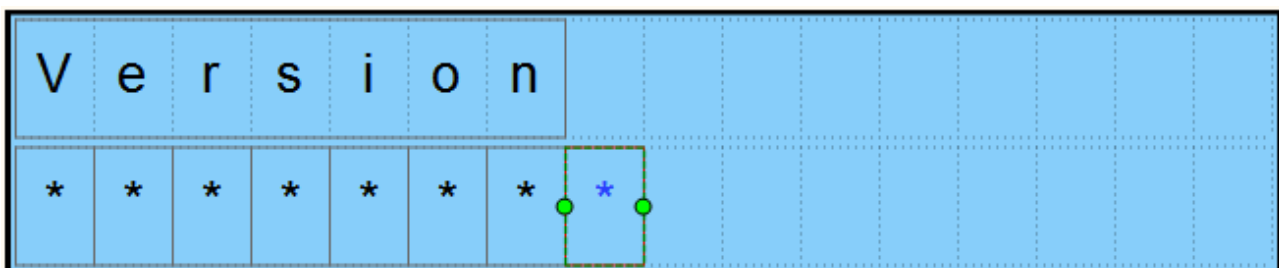


Compatible with
PR200.24.2.2
PR200.230.2.2

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

Nr	ID	Nr	ID	Nr	ID	Nr	ID
0	*	21	B	42	W	63	l
1	.	22	C	43	X	64	m
2	/	23	D	44	Y	65	n
3	0	24	E	45	Z	66	o
4	1	25	F	46	[67	p
5	2	26	G	47	\	68	q
6	3	27	H	48]	69	r
7	4	28	I	49	^	70	s
8	5	29	J	50	_	71	t
9	6	30	K	51	`	72	u
10	7	31	L	52	a	73	v
11	8	32	M	53	b	74	w
12	9	33	N	54	c	75	x
13	:	34	O	55	d	76	y
14	;	35	P	56	e	77	z
15	<	36	Q	57	f	78	*
16	=	37	R	58	g		
17	>	38	S	59	h		
18	?	39	T	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:



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:

Parameter in FB	Data type	Persistence	Default value	Used in project	Comment
SR_CHAR_7	INT	<input type="checkbox"/>	0	Yes	
SR_CHAR_6	INT	<input type="checkbox"/>	0	Yes	
SR_CHAR_5	INT	<input type="checkbox"/>	0	Yes	
SR_CHAR_4	INT	<input type="checkbox"/>	0	Yes	
SR_CHAR_3	INT	<input type="checkbox"/>	0	Yes	
SR_CHAR_2	INT	<input type="checkbox"/>	0	Yes	
SR_CHAR_1	INT	<input type="checkbox"/>	0	Yes	
SR_CHAR_0	INT	<input type="checkbox"/>	0	Yes	

Network variables (Slot 2):

Name	Data type	Read function	Write function	Register address	Bit number	Comment
Version_TRM2...	INT	0x03	0x06	4100		
Version_TRM202_r2	INT	0x03	0x06	4102		
<no >	BOOL	0x01	0x05	0	0	