Kernel communication protocol

USER MANUAL

rev. 0.1

29 November 2012

Indice

1.1 General notes

Each Kernel Sistemi device develop a communication protocol which allow to makes available to any external device the internal variable's values. Is possible create a network (with the RS422 or RS485 connection) with a master device and more slaves, each one of these slaves must have a different node address. The master will communicate with each one slave one by one.

1.2 Communication frame

The communication frame is “question] answer” type, the master device send a data packet and the slave answer immediately. The communication package is made only of ASCII characters: **'0'...'9' and 'A'...'F'** with the exception of the “**STX**” character which represent the transmission data start, and the character “**ETX**” which represent the end of the transmission and the character “**EOT**” which ends the write data block.
The master's query frame, starts always with the STX character, followed from two characters which represents the slave node address (up to 255). The next character is the command which can needs other characters in according to the command types. At the packet's end is inserted a control **checksum**, it's verify the message matching, the checksum is made of two ASCII characters, they are calculated as hexadecimal representation of the sum modulus 256, of the transmitted character code. The packet's end always with the character ETX. The possible presence of the character CR (0x0d) must be always ignored.

Master:		**STX**	**ad0**	**ad1**	**CMD**	**cmd parameters**	**ck0**	**ck1**	**ETX**
Slave:		**STX**	**answer parameters**				**ck0**	**ck1**	**ETX**

Some explanations:

STX	ASCII character <start of text>: 0x02
ad1, ad0	NET address ['0' '0' … 'F' 'F']
CMD	ASCII character which identify the command
cmd parameters	One or more parameters which depends from the command (looks commands list)
ck1,ck0	Hexadecimal checksum value calculated as representation of the sum modulus 256 of the characters from <ad1> to the last character.
EXT	ASCII character <end of text> : 0x03

1.3 Commands list

In the following table are indicated all the command codes

CMD	Meaning	Parameters
d	Data word reading	a3 a2 a1 a0 n1 n0
D	Data word writing	a3 a2 a1 a0 d3 d2 d1 … EOT

1.4 Data words reading

This command allow to read one or more memory locations inside the PLC. Each memory location has a word size (16bits) and it's characterized from an address between zero and a maximum which depends from the PLC model.

Master querying

STX	ad1	ad0	d	a3	a2	a1	a0	n1	n0	ck1	ck0	ETX

ad1, ad0 = slave node address
d = data words reading command
a3, a2, a1, a0 = first data to read [0000...FFFF]
n1, n0 = number of data to read [00...FF]

Slave answer

STX	d03	d02	d01	d00	d13	d12	d11	d10	...	...	ck0	ck1	ETX

d03, d02, d01, d00 = first read data's value [0000...FFFF]
d13, d12, d11, d10 = second read data's value [0000...FFFF]

1.5 Data word writing

This command allow to write one or more memory locations inside the PLC. Each location has a word size (16bits) and it's characterized from an address between zero and a maximum which depends from the PLC model.

Master querying

STX	ad1	ad0	D	a3	a2	a1	a0	d03	d02	d01	d00	d13	d12	d11	d10	...	EOT

ad1, ad0 = slave node address
D = word writing command
a3, a2, a1, a0 = node address first data to write [0000...FFFF]
d03, d02, d01, d00 = first location value to write [0000...FFFF]
d13, d12, d11, d10 = second location value to write [0000...FFFF]
…
EOT = end data character

Slave answer

STX	ACK	0	6	EOT
02	06	30	36	03

1.6 Example 1

Two words readings from the address 0x100 of the PLC with node address n.2 (0x100's value = 100 and 0x101's value = 1000)

Master querying

STX	0	2	d	0	1	0	0	0	2	E	9	ETX
02	30	32	64	30	31	30	30	30	32	45	39	03

02 = slave node address
d = reading command
0100 = first data address to read
02 = number of locations to read
E9 = checksum (30+32+64+30+31+30+30+30+32)

Slave answer

STX	0	0	6	4	0	3	E	8	A	A	ETX
02	30	30	36	34	30	33	45	38	41	41	03

0064 = value inside 0x100
03E8 = value inside 0x101
AA = checksum (30+30+36+34+30+33+45+38)

1.7 Example 2

Writing of value 100 at the internal address 0x100 and writing the value 1000 at the internal address 0x101

Master querying

STX	0	2	D	0	1	0	0	0	0	6	4	0	3	E	8	EOT	1	5	ETX
02	30	32	44	30	31	30	30	30	30	36	34	30	33	45	38	4	31	35	3

02 = slave node address
D = writing command
0100 = first data address to write
0064 = first location value to write
03E8 = second location value to write
EOT = end data character
15 = checksum (30+32+44+30+31+30+30+30+30+36+34+30+33+45+38+04)

Slave answer

STX	ACK	0	6	EOT
02	06	30	36	03

ACK = acknowladge character of right writing
06 = checksum

1.8 Communication errors

The master must manage also the communication errors, indeed is possible that the slave is absent or switched-off, or that the transmission has problems. If there is a receiving error or a checksum error, the slave don't answer, so the master should have a timeout within which waiting a reply; if there is no answer, the master could repeat the transmission or declare the slave off-line.
In case of attempt of non-existent data writing/reading, the slave will answer with a “not acknowledge” message.

Slave answer

STX	NAK	1	6	ETX
02	16	31	36	03

NAK = not acknowledge character of error