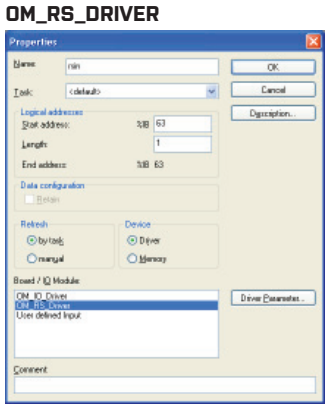
OMC 8000
RS DRIVER

RusAutomation

SHARED MEMORY %M3
Shared memory %M3.0 - %M3.3071, 3 kB in total, is used for communication between the PLC, HMI and other devices. MULTIPROG is unable to place variables into this memory and it also does not monitor overlapping of these variables. This can be an advantage if only a part of the data in the memory needs changing. Data in this part of the memory cannot be stored as retain (back up) data.

RETAIN MEMORY
Retain memory is used for storing data which must not be lost during the absence of power supply. MULTIPROG automatically places into this memory all the data indexed Retain. The size of this memory is 2 kB out of which 16 byte is used by the system.

OM_RS_DRIVER


OM_RS_Driver is used for controlling the RS485 communication, which is connected to input UNI1.4 (L-, terminal 12) and UNI1.4 (L+, terminal 12). A group of either inputs or outputs in the length of 1 Byte needs to be created in order to enter the parameters of this driver. The address of this Byte is not important, because OM_RS_Driver accesses the shared memory directly. Its setting can look like this example:

OM_RS_Driver allows for the use of 6 communication protocols. All of them share the same setting of communication speed in the range of 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400 Baud. Except for the universal ASCII protocol, communication is realized with these parameters: 8 bits data, 1 stop, no parity.

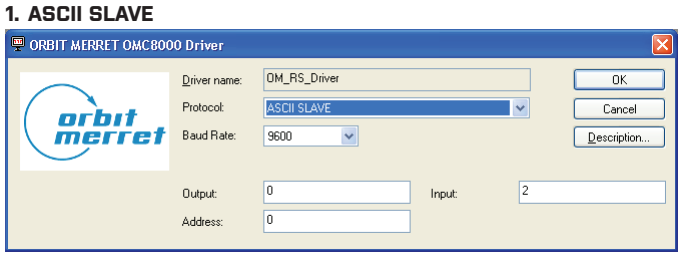
For ASCII master and ASCII slave protocols it is necessary to define in the program a data type String8, which is used for recording data and reading out of data.

This is done in section Data Types according to the following definition:

```
TYPE
  String8:STRING(8);
END_TYPE
```

MULTIPROG stores strings along with other information, so in the memory there are stored 5 + byte length, according to the following pattern:

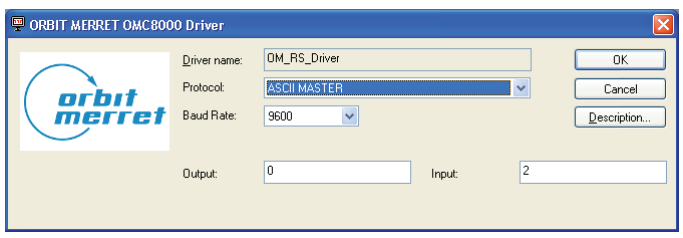
MLL MLH LL LH	D D ... D D ... D 0
MLL/MLH	maximum length lower/higher byte (maximum of 32.767 bytes)
LL/LH	current length lower/higher byte
D	data
0	byte with code 0

1. ASCII SLAVE


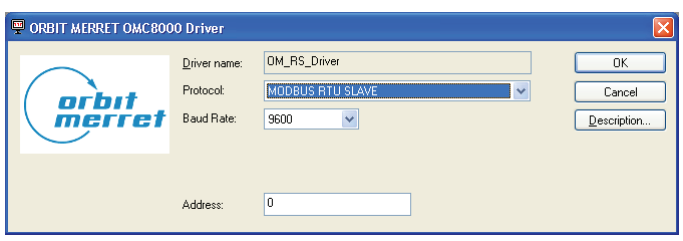
- OMC 8000 communicates as any Orbit Merret device and has three parameters:
  - number of data for output, data are stored from the start of the shared memory
  - number of data for input, data follow output data
  -
- there can be a maximum of 232 data
- output data can be read out using commands 1A..1Z, 1a..1z, ... , 4A..4Z, 4a..4z
- output data can also be read out via command 7W. Data are divided by semicolon and have a variable length of 0 – 8 bytes
- output data for RS cannot be changed
- input data are entered by command 5A..5Z, 5a..5z, ... , 8A..8Z, 8a..8z, every parameter has maximum of 8 characters
- in the shared memory strings are stored according to the following table:

Write/Read	Address	Write/Read	Address	Write/Read	Address	Write/Read	Address
1A/5A	0	2A/6A	676	3A/8A	1352	4A/9A	2028
1B/5B	13	2B/6B	689	3B/8B	1365	4B/9B	2041
1C/5C	26	2C/6C	702	3C/8C	1378	4C/9C	2054
1D/5D	39	2D/6D	715	3D/8D	1391	4D/9D	2067
1E/5E	52	2E/6E	728	3E/8E	1404	4E/9E	2080
1F/5F	65	2F/6F	741	3F/8F	1417	4F/9F	2093
1G/5G	78	2G/6G	754	3G/8G	1430	4G/9G	2106
1H/5H	91	2H/6H	767	3H/8H	1443	4H/9H	2119
1I/5I	104	2I/6I	780	3I/8I	1456	4I/9I	2132
1J/5J	117	2J/6J	793	3J/8J	1469	4J/9J	2145
1K/5K	130	2K/6K	806	3K/8K	1482	4K/9K	2158
1L/5L	143	2L/6L	819	3L/8L	1495	4L/9L	2171
1M/5M	156	2M/6M	832	3M/8M	1508	4M/9M	2184
1N/5N	169	2N/6N	845	3N/8N	1521	4N/9N	2197
1O/5O	182	2O/6O	858	3O/8O	1534	4O/9O	2210
1P/5P	195	2P/6P	871	3P/8P	1547	4P/9P	2223
1Q/5Q	208	2Q/6Q	884	3Q/8Q	1560	4Q/9Q	2236
1R/5R	221	2R/6R	897	3R/8R	1573	4R/9R	2249
1S/5S	234	2S/6S	910	3S/8S	1586	4S/9S	2262
1T/5T	247	2T/6T	923	3T/8T	1599	4T/9T	2275
1U/5U	260	2U/6U	936	3U/8U	1612	4U/9U	2288
1V/5V	273	2V/6V	949	3V/8V	1625	4V/9V	2301
1W/5W	286	2W/6W	962	3W/8W	1638	4W/9W	2314
1X/5X	299	2X/6X	975	3X/8X	1651	4X/9X	2327
1Y/5Y	312	2Y/6Y	988	3Y/8Y	1664	4Y/9Y	2340
1Z/5Z	325	2Z/6Z	1001	3Z/8Z	1677	4Z/9Z	2353
1a/5a	338	2a/6a	1014	3a/8a	1690	4a/9a	2366
1b/5b	351	2b/6b	1027	3b/8b	1703	4b/9b	2379
1c/5c	364	2c/6c	1040	3c/8c	1716	4c/9c	2392
1d/5d	377	2d/6d	1053	3d/8d	1729	4d/9d	2405
1e/5e	390	2e/6e	1066	3e/8e	1742	4e/9e	2418
1f/5f	403	2f/6f	1079	3f/8f	1755	4f/9f	2431
1g/5g	416	2g/6g	1092	3g/8g	1768	4g/9g	2444
1h/5h	429	2h/6h	1105	3h/8h	1781	4h/9h	2457
1i/5i	442	2i/6i	1118	3i/8i	1794	4i/9i	2470

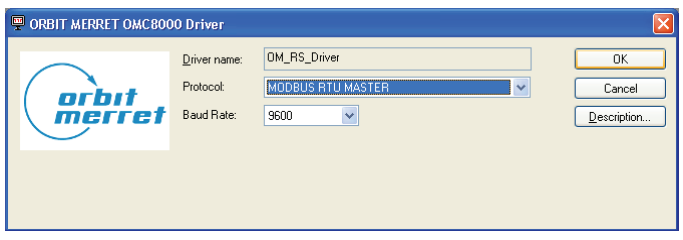
Write/Read	Address	Write/Read	Address	Write/Read	Address	Write/Read	Address
1j/5j	455	2j/6j	1131	3j/8j	1807	4j/9j	2483
1k/5k	468	2k/6k	1144	3k/8k	1820	4k/9k	2496
1l/5l	481	2l/6l	1157	3l/8l	1833	4l/9l	2509
1m/5m	494	2m/6m	1170	3m/8m	1846	4m/9m	2522
1n/5n	507	2n/6n	1183	3n/8n	1859	4n/9n	2535
1o/5o	520	2o/6o	1196	3o/8o	1872	4o/9o	2548
1p/5p	533	2p/6p	1209	3p/8p	1885	4p/9p	2561
1q/5q	546	2q/6q	1222	3q/8q	1898	4q/9q	2574
1r/5r	559	2r/6r	1235	3r/8r	1911	4r/9r	2587
1s/5s	572	2s/6s	1248	3s/8s	1924	4s/9s	2600
1t/5t	585	2t/6t	1261	3t/8t	1937	4t/9t	2613
1u/5u	598	2u/6u	1274	3u/8u	1950	4u/9u	2626
1v/5v	611	2v/6v	1287	3v/8v	1963	4v/9v	2639
1w/5w	624	2w/6w	1300	3w/8w	1976	4w/9w	2652
1x/5x	637	2x/6x	1313	3x/8x	1989	4x/9x	2665
1y/5y	650	2y/6y	1326	3y/8y	2002	4y/9y	2678
1z/5z	663	2z/6z	1339	3z/8z	2015	4z/9z	2691

2. ASCII MASTER


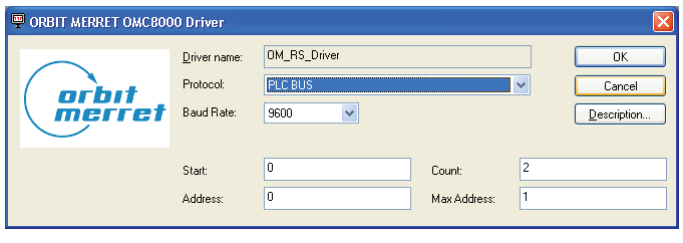
- OMC 8000 projects information on displays OM xxxRS and reads data from instruments OM and it has two parameters:
  - number of data for output, data are stored from the start of shared memory
  - number of data for input, data follow output data
- output data are transmitted via command 9, where every parameter has 0 - 8 characters (i.e. #009888.888<CR>)
- input data are received via command 7W, and into the memory are stored chronologically, where there are no data, "NoData" is stored. For example OM 402UNI stores two strings – value from channel A and MF, for OMU 408UNI there will be 9 strings
- addresses of individual strings correspond with those from the above table

3. MODBUS RTU SLAVE


- OMC 8000 behaves as a standard slave with entire memory accessible as HOLDING registry (address 40000) Register 40000 = %MW3.0, 40001 = %MW3.2, 40002 = %MW3.4 and has a single parameter:
  - address on a MODBUS line. Addresses can be entered in a range of 1 – 247
- Maximum of 64 registers can be sent out and received at once
- **!ATTENTION!** PLC system uses bytes which are placed into the memory in a reverse order and this means that in multi-word items individual words will be in a reverse order.
  For example: In PLC long at address 100 with a value of 0x87654321 with command AA 03 00 32 00 02 CR CR it returns AA 03 04 43 21 87 65 CR CR (CR CR ... 16 bit CRC)
- Implemented commands:
  - 3 for reading
  - 6 and 16 for writing

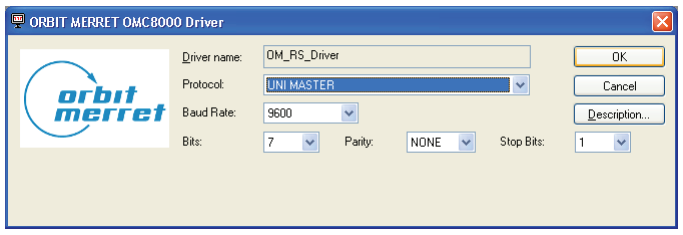
4. MODBUS RTU MASTER


- uses a universal sending-out function block RsSend
- behind the Count of the sent-out bits it also adds CRC
- parameter AsString, EndChar, EC_Count have no meaning I this case
- answer is accepted whole, INCLUDING CRC
- FB outputs are controlled as follows:
  - "Done" is set for the duration of sending out the command, at least one program cycle
  - "Received" is set if an answer is received
- if data cannot fit into the shared memory, they will not be stored at all and "Error" is set
- "Error" is also set in case of TO, or CRC error, of if the number of sent-out data is too high

5. PLC BUS


- special, binary protocol for the fastest possible refreshing of data between instruments where UDP communication over ETHERNET is not desirable. It has four parameters:
  - beginning of data in shared memory, which it shares with other PLCs
  - number of these data
  - address on the bus 0 – 31. Addresses must start from 0, come in a chronological order and be unique
  - maximum address on the bus. This parameter is identical for all PLCs
- PLCs alternate in their transmission in a cyclic manner
- loop is broken, PLCs with address 0 repeats its broadcast after 5 seconds

6. UNI MASTER



- has 3 parameters (Bits, Parity, Stop bits) for entering all communication parameters
- function block RSSend with all its parameters is used for transmission and reception
- the length of the buffer is 136 characters max. Characters which do not fit within are discarded Buffer overflow is not indicated.
- if recording AsString is selected, then only as many characters are stored as many are permitted If the number of received characters is higher, error is indicated
- if data do not fit into the shared memory, they will not be stored at all and an error is indicated

THE FOLLOWING FU AND FB WILL BE DESCRIBED IN HELP OF LIBRARY OM_FW_LIB

ReadRSBusy - FU
Returns true, if previous command in UNI_MASTER or in MODBUS_MASTER mode is running

RsSend - FB

has these parameters	Start	address in the shared memory, where broadcast data start (MODBUS_MASTER) address in the shared memory, where broadcast string starts (UNI)
	Count	number of data to be broadcast
	Answer	answer will be required
	AsString	will be stored as a string
	AnswAdr	address in the shared memory where the space for received data starts
	EndChar	will wait for the end character
	EC_Count	end character a number of characters to be received
	TimeOut	timeout for the reception of answer, maximum of 65.535 s
returns	Done	when the data broadcast was successful Sets itself for the duration of processing of the entire command (including reception) for at least one program cycle. Does not set itself if the line is busy with another FB RSSend
	Received	when data reception was succesful. Sets itself for one program cycle
	Error	timeout or incorrect data size or a different error

CtrlStrToBuf - FB
Parses String on the input side into buffer. Please pay **attention** to the buffer length.
It will be able to parse these codes

\\	\	0x5C
%%	%	0x25
\s	STX	0x02
\e	ETX	0x03
\t	TAB	0x08
\l	LF	0x0A
\r	CR	0x0D
\n	CRLF	0x0D 0x0A

ignores other unknown codes, nothing appears in the buffer and an error is indicated.

%HH	?	0xHH	character with code HH, provided HH is not a hexa-number then characters % and the following two characters are discarded and the buffer is empty

ООО “РусАвтоматизация”
454010 г. Челябинск, ул. Гагарина 5, оф. 507
тел. 8-800-775-09-57 (звонок бесплатный), +7(351)799-54-26, тел./факс +7(351)211-64-57
info@rusautomation.ru; русавтоматизация.рф; www.rusautomation.ru;