The following explains the commands used for the dialogue with the MFR-modules, to help to create your own application programs. Any errors, or version dependent changes can not be excluded. In any case, the results of programming should be tested directly to the interface with appropriate tools, such as a terminal program, to minimize unpleasant surprises.

Command Scheme		
Command	**Parameter**	**CR**
1 Character	Parameter list of any length, coded as 40h + upper or lower 4-bit-nibble of data-byte, yielding printable ASCII-Characters from '@' to 'O'. For each data-byte, there must be provided two characters in the order MSB, LSB..	0Dh

Cmd	Parameters	Send to module	Receive from module
Pure interface functions. Which are normally processed by an application program.			
X (58h)	none	Forces restart of module. It is a soft reset, the proper communication with the module is required!	Not generated by the module Module-ID is emitted after restart (e.g.: XSP01R)
O (4Fh)	none	Retrieves output levels	Not generated by the module
	1. Character Channel 7..4	Sets outputs 0..7 as byte	Level of outputs 0..7 a) as a response b) automatically by level changing
	2. Character Channel 3..0		
	3. Character Mask 7..4	Optional – If present: Affects only those outputs for which the mask bit is set. *** Firmware V 1.10 or newer***	Not generated by the module
	4. Character Mask 3..0		
o (6Fh)	none	Retrieves level of outputs, as in 'O'	Not generated by the module
	1. Character Bit-Address	Asserts at bit address (0..7) the status (0..1). Other outputs are not affected. *** Firmware V 1.10 or newer***	Not generated by the module
	2. Character Bit-Status		
I (49h)	none	Retrieves input levels	Not generated by the module
	1. Character Channel 7..4	Only for test purposes It results in an OR operation with the corresponding physical inputs	Level of inputs 0..7 a) as a response b) automatically by level changing
	2. Character Channel 3..0		
D (44h)	1. Character t / 16	Activates the communication monitoring time **t** in units of 100 ms. Any missing commands 'O' or 'I' during the time **t**, will result in resetting all outputs to the passive state. The value **t** = 0 disables the function monitoring. *** Firmware V 1.10 or newer***	Not generated by the module
	2. Character t % 16		
Other commands that control the internal functions of the MFR module they are usually processed by the MFR configuration program. If the module is only used as an I / O interface, these commands are irrelevant.			

ZEB AUTOMATION LIMITED
Wakefield, West Yorkshire, GB
Branch Office Germany
Obentrautstr. 35, D-10963 Berlin
☎+49-30-6212667 +49-30-25294347 info@zeb-automation.de

Cmd	Parameters	Send to module	Receive from module
Q (51h)	none	Retrieves parameter list from module The expected response: Name (**N**), Version (**V**), Serial # (**S**), ID (**U**)	Not generated by the module
N (4Eh)	none	Retrieves the name of the module	A user defined module-name, with a maximal length of 20 printable characters.
	Maximal 20 Characters	Ignored by the module	
n (6Eh)	Maximal 20 Characters	Sets a user defined module-name, with a maximal length of 20 printable characters.	Not generated by the module
V (56h)	String	Ignored by the module	Version of Firmware as 'Version.Compilation'
S (53h)	Block 16 Characters	Ignored by the module	Unique serial-number of module in hexadecimal notation, in order of Component-ID, Number-Byte 0..5, BCC
Z (5Ah)	1. Character Channel #	Retrieves the function-data of selected channel	Function data of selected channel, 1 byte for each: function index, inputs mask, t1-value, t1-timebase, t2-value, t2-timebase, flag1, flag2
	16 Characters Data	Not necessary	
z (7Ah)	1. Character Channel #	Sets the function-data of selected channel	Not generated by the module
	16 Characters Data		
U (55h)	1. Character Type of Output	Ignored by the module	Module-ID Type of Output : L = Solid-State, R = Relay
	String		More Module-ID shortcuts .. Type of Interface (Param[2]): E = Ethernet, R = RS-232, U = USB
T (54h)	1. Character Message #	Ignored by the module	User defined partial message. Because of memory structure, 1 message consists of 2 parts (even + odd #). (8 messages = 16 partial messages)
	Maximal 10 Characters		
t (74h)	1. Character Message #	Sets user defined messages. Because of memory structure, 1 message consists of 2 parts (even + odd #). (8 messages = 16 partial messages)	Not generated by the module
	Maximal 10 Characters		

Hints for Programming

The procedure for programming the interface is somewhat dependent on the programming language and the structural design, so no single approach will be proposed. Usually, all programming languages provides some methods, to access real communication ports, either directly or indirectly through additional libraries. In the download area of our web presence, we have provided some programming examples for a possible approach.

Sending is generally no problem. The desired and composed output string will be passed to the output object, either as character by character or as a complete string. It is assumed that the appropriate interface has been configured in the right way. Whether the information, returned by the respective transmission routines, will evaluated or not, is at the sole discretion of the programmer. The MFR module receives the string in a receive buffer, and evaluates the string after the first seen

CR (0Dh) as a function of the first character in buffer. Then, the buffer will be emptied. If the first character in buffer can not be identified as a known command, the following characters are ignored. Therefore, it should be noted that no stray characters are allowed in the transmitting output channel. This problem occurs occasionally when initializing interfaces. If it can not be avoided, there should be emitted a CR (0Dh) before the first useful command. This will cause that the receive buffer is emptied. An appended LF (0Ah) does not interfere, it will be ignored on reading. In rapid succession of output commands, and especially when not applied the evaluation of status messages of the sending routine, it is possible that, during the command-evaluation in the MFR-module, the next command will arrive in the receive buffer, but is no longer seen. The same happens when a final CR is lost. The CR of the following command only executes the previous, incomplete command. The corresponding command for this CR is forfeited.

Receiving is a bit trickier. If hardware and programming capabilities allow it, it should be used a buffered method, to receive the messages from the MFR module. To raise an event for the evaluation of a received message, it can be triggered from an incoming CR, as the message contains only printable characters. As a function of the program, and the type of the transmitted sequence of commands, several messages of the MFR module can be found in the background buffer, the buffer-reading must be repeated until the buffer is empty. It should be noted, that the module can send unsolicited, or without specific request, state changes of the inputs and outputs. Inputs and outputs are cyclically scanned, at a rate of 100-ms. If there is detected a change against the respective previous state, a message is generated. If only the answer to a specific request is to be evaluated, the receive buffer must be emptied before, before transmitting the request. Otherwise an older message, not the wanted one, may be erroneously analyzed and may reflect an incorrect state. When working without background buffer, which can actually only be useful in case of the serial RS232 interface, the receive loop must be designed so that the transmission speed of 9600 baud can be processed safely, otherwise some characters may be lost. Especially the using of high-level languages and older (slower) hardware is therefore dangerous. In these cases it may be a better solution, to address the control registers of the communication objects directly.
The MFR-Module, with RS232-Interface, does not generate modem-control-signals, it uses only the signals Rx, Tx and GND. Unless the control signals for data transfer are mandatory, they must be generated elsewhere.

It should be noted that, if virtual COM ports or USB or direct Ethernet connections are used, that these interfaces have their own timing, and that a block of data to be transferred will not necessarily be received as a single block. That must be taken into account when evaluating the received messages from a MFR-Module.

Some helping things ..

Character	Bit 7/3	Bit 6/2	Bit 5/1	Bit 4/0	Character	Bit 7/3	Bit 6/2	Bit 5/1	Bit 4/0
@	0	0	0	0	**H**	1	0	0	0
A	0	0	0	1	**I**	1	0	0	1
B	0	0	1	0	**J**	1	0	1	0
C	0	0	1	1	**K**	1	0	1	1
D	0	1	0	0	**L**	1	1	0	0
E	0	1	0	1	**M**	1	1	0	1
F	0	1	1	0	**N**	1	1	1	0
G	0	1	1	1	**O**	1	1	1	1

ZEB AUTOMATION LIMITED
Wakefield, West Yorkshire, GB
Branch Office Germany
Obentrautstr. 35, D-10963 Berlin
☎+49-30-6212667 +49-30-25294347 info@zeb-automation.de

LSB / MSB	0_	1_	2_	3_	4_	5_	6_	7_
_0	NUL	DLE	SP	0	@	P	`	p
_1	SOH	DC1	!	1	A	Q	a	q
_2	STX	DC2	„	2	B	R	b	r
_3	ETX	DC3	#	3	C	S	c	s
_4	EOT	DC4	$	4	D	T	d	t
_5	ENQ	NAK	%	5	E	U	e	u
_6	ACK	SYN	&	6	F	V	f	v
_7	BEL	ETB	'	7	G	W	g	w
_8	BS	CAN	(	8	H	X	h	x
_9	HT	EM	)	9	I	Y	i	y
_A	LF	SUB	*	:	J	Z	j	z
_B	VT	ESC	+	;	K	[	k	{
_C	FF	FS	,	<	L	\	l	|
_D	CR	GS	-	=	M	]	m	}
_E	SO	RS	.	>	N	^	n	~
_F	SI	US	/	?	O	_	o	DEL

ZEB AUTOMATION LIMITED
Wakefield, West Yorkshire, GB
Branch Office Germany
Obentrautstr. 35, D-10963 Berlin
☎+49-30-6212667 +49-30-25294347 info@zeb-automation.de