

Anwendungs-Programmierung der MFR-Modul Serie

Stand: 2010-01-28 * Seite 1 von 4

Nachfolgend werden die verwendeten Befehle für den Dialog mit den MFR-Modulen erläutert, um das Erstellen eigener Anwendungs-Programme zu erleichtern. Eventuelle Fehler oder versionsübergreifende Änderungen sind jedoch nicht auszuschließen. In jedem Falle sollten die Ergebnisse der eigenen Programmierung direkt an der Schnittstelle mit geeigneten Tools, wie z.B. einem Terminalprogramm kontrolliert werden, um unliebsame Überraschungen von vornherein zu minimieren.

Befehlsaufbau		
Befehl	Parameter	CR
1 Buchstabe	Parameterliste beliebiger Länge, codiert in der Form 40h + obere oder untere 4 Bit des Nutzbytes. Daraus ergeben sich druckbare ASCII-Zeichen im Bereich von '@' bis 'O'. Für jedes Nutzbyte müssen daher 2 Zeichen in der Reihenfolge MSB, LSB generiert werden.	0Dh

Befehl	Parameter	Zum Modul senden	Vom Modul empfangen
Reine Interface-Funktionen. Die üblicherweise vom Applikationsprogramm verarbeitet werden.			
X (58h)	ohne	Erzwingt den Neustart des Moduls. Es ist ein Soft-Reset, der die Dialogfähigkeit des Moduls voraussetzt!	Wird vom Modul nicht generiert Das Modul sendet jedoch nach dem Neustart die Identität (z.B.: XSP01R)
	ohne	Fordert den Status der Ausgänge an	Wird vom Modul nicht generiert
O (4Fh)	1. Zeichen Kanal 7..4	Setzt die Ausgänge 0..7 gem. übergebenen Bitmuster	Status der Ausgänge 0..7
	2. Zeichen Kanal 3..0		a) als Antwort auf Anforderung b) automatisch bei Veränderung
	3. Zeichen Maske 7..4	Optional - Wenn vorhanden: Es werden nur die Ausgänge beeinflusst, deren Maskenbit gesetzt ist. *** ab Firmware V 1.10 ***	Wird vom Modul nicht generiert
	4. Zeichen Maske 3..0		
o (6Fh)	ohne	Fordert den Status der Ausgänge an, wie 'O'	Wird vom Modul nicht generiert
	1. Zeichen Bit-Adresse	Setzt nur an der Bit-Adresse (Ausgang 0..7) den Bit-Status (0 oder 1), andere Ausgänge werden nicht berührt. *** ab Firmware V 1.10 ***	Wird vom Modul nicht generiert
	2. Zeichen Bit-Status		
I (49h)	ohne	Fordert den Status der Eingänge an	Wird vom Modul nicht generiert
	1. Zeichen Kanal 7..4	Nur für Testzwecke Bewirkt ODER-Verknüpfung mit dem zugehörigen physikalischen Eingang	Status der Eingänge 0..7
	2. Zeichen Kanal 3..0		a) als Antwort auf Anforderung b) automatisch bei Veränderung
D (44h)	1. Zeichen t / 16	Setzt und aktiviert die Dialog-Überwachungszeit t in 100-ms-Einheiten. Bleiben die Kommandos O oder I während der Zeit t aus, werden alle Ausgänge in den passiven Status zurückgesetzt. Der Zeitwert t=0 deaktiviert die Überwachungsfunktion. *** ab Firmware V 1.10 ***	
	2. Zeichen t % 16		

ZEB AUTOMATION LIMITED

Wakefield, West Yorkshire, GB

Niederlassung Deutschland

Obentrautstr. 35, D-10963 Berlin

☎ +49-30-6212667 📠 +49-30-25294347 ✉ info@zeb-automation.de

Anwendungs-Programmierung der MFR-Modul Serie

Stand: 2010-01-28 * Seite 2 von 4

Befehl	Parameter	Zum Modul senden	Vom Modul empfangen
Weitere Kommandos, die die internen Funktionen des MFR-Moduls steuern, sie werden üblicherweise vom MFR-Konfigurationsprogramm verarbeitet. Wird das Modul lediglich als I/O-Interface genutzt, sind diese Kommandos belanglos.			
Q (51h)	ohne	Fordert Parameterliste aus dem Modul an Die erwartete Rückmeldung besteht aus: Name (N), Version (V), Seriennummer (S), ID (U)	Wird vom Modul nicht generiert
N (4Eh)	ohne	Fordert den Namen des Moduls an	Ein freier Name, bestehend aus maximal 20 druckbaren Zeichen
	Maximal 20 Zeichen	Wird vom Modul nicht ausgewertet	
n (6Eh)	Maximal 20 Zeichen	Setzt einen frei vergebbaren Namen, bestehend aus maximal 20 druckbaren Zeichen.	Wird vom Modul nicht generiert
V (56h)	Offene Zeichenfolge	Wird vom Modul nicht ausgewertet	Versionsnummer der Firmware in der Notierung 'Version.Kompilation'
S (53h)	Block 16 Zeichen	Wird vom Modul nicht ausgewertet	Einmalige Seriennummer des Moduls in hexadezimaler Notierung in der Reihenfolge Baustein-Typ, Num0..5, BCC
Z (5Ah)	1. Zeichen Kanal #	Fordert die Funktionseinstellungen des gewählten Kanals an	Funktionsdatenblock des Kanals, je 1 Byte für: Funktionsindex, Eingangsmaske, t1-Wert, t1-Zeitbasis, t2-Wert, t2-Zeitbasis, Flag1, Flag2
	16 Zeichen Datenblock	Nicht erforderlich	
z (7Ah)	1. Zeichen Kanal #	Setzt den Funktionsdatenblock des Kanals	Wird vom Modul nicht generiert
	16 Zeichen Datenblock		
U (55h)	1. Zeichen Ausgang	Wird vom Modul nicht ausgewertet	Modultyp-Identifikation Ausgang : L = Halbleiter, R = Relais
	Offene Zeichenfolge		Modultyp-Identifikation Kurzbezeichnung beinhaltet Schnittstellentyp (Param[2]): E = Ethernet, R = RS-232, U = USB
T (54h)	1. Zeichen Meldung #	Wird vom Modul nicht ausgewertet	Gespeicherte Meldungstexte bis zu 20 Zeichen Systembedingt besteht 1 Meldungstext aus 2 Teilmeldungen (gerade + ungerade #) max. 8 Meldungen (= 16 Teilmeldungen)
	Maximal 10 Zeichen		
t (74h)	1. Zeichen Meldung #	Setzt frei definierbare Meldungstexte Systembedingt besteht 1 Meldungstext aus 2 Teilmeldungen (gerade + ungerade #) max. 8 Meldungen (= 16 Teilmeldungen)	Wird vom Modul nicht generiert
	Maximal 10 Zeichen		

Hinweise zur Programmierung

Die Vorgehensweise bei der Schnittstellenprogrammierung ist etwas von der Programmiersprache und der strukturellen Konzeption abhängig, daher kann kein allgemein gültiges Konzept vorgeschlagen werden. In der Regel lassen alle Programmiersprachen, entweder direkt oder indirekt über ein-

ZEB AUTOMATION LIMITED

Wakefield, West Yorkshire, GB

Niederlassung Deutschland

Obentrautstr. 35, D-10963 Berlin

☎ +49-30-6212667 📠 +49-30-25294347 ✉ info@zeb-automation.de

Anwendungs-Programmierung der MFR-Modul Serie

Stand: 2010-01-28 * Seite 3 von 4

gebundene Funktions-Bibliotheken, den Zugriff auf Ausgabekanäle zu. Im Downloadbereich unserer Web-Präsenz haben wir für die eine oder andere Programmiersprache Beispiele für eine mögliche Vorgehensweise bereitgestellt.

Das **Senden** ist in der Regel problemlos. Es wird der gewünschte Ausgabestring zusammengestellt und entweder zeichenweise oder als komplette Zeichenkette dem Ausgabeobjekt übergeben. Es wird vorausgesetzt, dass die zu benutzende Schnittstelle zu Beginn entsprechend parametrisiert wurde. Ob von den jeweiligen Senderroutinen zurückgegebene Statusinformationen ausgewertet werden, liegt einzig im Ermessen des Programmierers. Das MFR-Modul empfängt die Zeichenfolge in einem Empfangspuffer und wertet die Zeichenfolge nach Empfang eines CR (0Dh) in Abhängigkeit des ersten Zeichens aus. Anschließend wird der Empfangspuffer zurückgesetzt. Ist das erste Zeichen nicht als definiertes Kommando zu interpretieren, wird der Rest nicht mehr 'gesehen'. Daher ist zu beachten, dass sich keine vagabundierenden Zeichen im Ausgabekanal befinden dürfen. Dieses Problem tritt gelegentlich bei der Initialisierung von Schnittstellen auf. Falls es nicht vermieden werden kann, sollte vor dem ersten sinnvollen Kommando ein CR ausgegeben werden, damit im MFR-Modul der Empfangspuffer geleert wird. Ein vom System an das CR angehängenes LF (0Ah) stört nicht, es wird beim Empfang überlesen. Bei rascher Folge von Ausgabekommandos und besonders bei Nichtauswertung von Statusmeldungen der angewandten Senderroutine ist nicht auszuschließen, dass während der Auswertung im MFR-Modul das nachfolgende Kommando zwar im Empfangspuffer landet, aber nicht mehr 'gesehen' wird. Das gleiche passiert, wenn ein abschließendes CR verloren geht. Das CR eines nachfolgenden Kommandos führt dann lediglich zur Ausführung des vorherigen, unvollständigen Kommandos. Der zum CR gehörende Funktionsbefehl geht somit verloren.

Das **Empfangen** ist etwas diffiziler. Sofern Hardware und Programmierfunktionen es zulassen, sollte ein Hintergrund-Puffer für den Empfang der Zeichenfolgen vom MFR-Modul verwendet werden. Da ausschließlich druckbare Zeichen vorkommen, kann auf das abschließende CR getriggert werden, um ein Empfangsereignis zur Auswertung auszulösen. Da in Abhängigkeit des Programms und der Art der ausgesandten Befehlssequenz mehrere Antworten des MFR-Moduls im Empfangspuffer eingelaufen sein können, muss das Auslesen wiederholt werden, bis der Puffer leer ist. Dabei ist zu beachten, dass das Modul auch ohne besondere Anforderung Zustandsänderungen der Ein- und Ausgänge unaufgefordert senden kann. Die Eingänge werden zyklisch im 100-ms-Takt abgefragt, wird eine Änderung des jeweilig letzten Status festgestellt, erfolgt eine entsprechende Mitteilung. Soll nur die Antwort auf eine gezielte Anfrage ausgewertet werden, muss vor dem Aussenden des Anforderungsbefehls der Empfangspuffer geleert werden, andernfalls werden frühere, noch nicht abgeholte, Mitteilungen ausgewertet, die möglicherweise ein falschen Zustand widerspiegeln. Wird ohne Hintergrundpuffer gearbeitet, was eigentlich nur bei der seriellen RS232-Schnittstelle sinnvoll sein kann, ist die Empfangsschleife so zu gestalten, dass die Übertragungsgeschwindigkeit von 9600 Baud sicher verarbeitet werden kann, da sonst Zeichen verloren gehen können. Besonders bei Hochsprachen-Funktionen und 'langsamer Hardware' ist daher Vorsicht geboten. In diesen Fällen ist es besser, die Register der Kommunikationsbausteine direkt anzusprechen. Modem-Steuersignale werden vom MFR-Modul mit RS232-Schnittstelle weder generiert noch ausgewertet, es werden nur die Leitungen Rx und Tx genutzt. Sofern die Steuersignale für den Datentransfer zwingend erforderlich sind, müssen sie anderweitig erzeugt werden.

Werden virtuelle COM-Ports bzw. die direkte USB- oder Ethernet-Verbindung genutzt, ist zu beachten, dass diese Schnittstellen ihr eigenes Zeitverhalten haben und dass ein zu übertragender Datenblock nicht zwangsläufig auch als ein einziger Block empfangen werden muss. Das muss u.U. in der Empfangsauswertung entsprechend berücksichtigt werden.

ZEB AUTOMATION LIMITED

Wakefield, West Yorkshire, GB

Niederlassung Deutschland

Obentrautstr. 35, D-10963 Berlin

☎ +49-30-6212667 📠 +49-30-25294347 ✉ info@zeb-automation.de

Anwendungs-Programmierung der MFR-Modul Serie

Stand: 2010-01-28 * Seite 4 von 4

Kleine Hilfsmittel

Zeichen	Bit 7/3	Bit 6/2	Bit 5/1	Bit 4/0	Zeichen	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

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

Niederlassung Deutschland

Obentrautstr. 35, D-10963 Berlin

☎+49-30-6212667 📠+49-30-25294347 ✉info@zeb-automation.de

Application Programming of MFR-Module Series

Issue: 2011-11-29 * Page 1 of 4

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)
	none	Retrieves output levels	Not generated by the module
O (4Fh)	1. Character Channel 7..4	Sets outputs 0..7 as byte	Level of outputs 0..7
	2. Character Channel 3..0		a) as a response b) automatically by level changing
	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
	2. Character Channel 3..0		a) as a response b) automatically by level changing
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

Application Programming of MFR-Module Series

Issue: 2011-11-29 * Page 2 of 4

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

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

Application Programming of MFR-Module Series

Issue: 2011-11-29 * Page 3 of 4

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

Application Programming of MFR-Module Series

Issue: 2011-11-29 * Page 4 of 4

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