MCR for Programmers – How I could use the MCR ? V 0.5

The MCR is a 4 or 8 SmartCard multiplexer, which multiplexes the serial RX/TX and RTS lines over an Altera CPLD to the cards. Four clocks are generated on the MCR which are selectivity switched over the Altera CPLD to the cards. Reset state (the RTS Line) of the Card and the chosen clocks are maintained for each card separately. The clock settings could be programmed to the internal EEPROM, so the last programmed frequency for each slot will be restored at next power up. Even the Reset state is maintained for each slot when switching to other slots, so it is possible the hold cards in reset, when working with an other one. Two serial status lines are providing additional information to the Host. This is at first the CTS line which goes to zero when a change of the cards happens (will be cleared after a Get Status Command) and second the DCD line, stated that all Slots are empty on the MCR.

The MCR has a command and an operating mode. The command mode is archived by setting the DTR line to logical 1. In this mode the ATMega is connected to the RX/TX lines and will receive and execute commands, which are send with 9600 Baud 8N1. As time is rare, the commands are short and the MCR don't answers most time and don't echos to the incoming serial stuff.

All Slots are internally numbered from 0 to 7, for the user from 1 to 8.

Any questions ? e-mail: mcr@irata.de
Info page: http://mcr.irata.biz

Attention: As some unwanted EEPROM writes happened out in the real world a version 2 of the Firmware is released which needs the new 'p' command in front of a clock and time out save command. Please look below.

	Status Request
Command	HEX 0x3f (ASCII “?”)
Length	1 byte
Answer	2 bytes
Detail	First byte: Slot usage – every bit in this byte stands for one card slot. When a card is inserted the corresponding bit goes 1. Bit 0 (LSB) stands for the Slot 0, the Bit 7 (MSB) for Slot 7.
Example	(Answer) 0x03 0x01 Slot 0 and 1 are used and Slot 1 is the active one.

	Select Slot
Command	HEX 0x73 (ASCII “s”) and one additional byte with the slot number (0..7)
Length	2 bytes
Answer	None
Detail	Slots are internally numbered from 0 to 7. Only the last 3 bits from the incoming number will be used.
Example	(Command) 0x73 0x03 This is selecting the Slot 3

	Get Version
Command	HEX 0x76 (ASCII “v”)
Length	1 byte
Answer	1 byte
Detail	Gives back the Firmware version number.
Example	(Answer) 0x02 Version 2 - this is the second version with EEPROM write protect.

	Get Type
Command	HEX 0x74 (ASCII “t”)
Length	1 byte
Answer	1 byte
Detail	Gives back the MCR Type. 4 is for MCR4 8 is for MCR8
Example	(Answer) 0x08 You are playing with an MCR8

	Get Serial
Command	HEX 0x6e (ASCII “n”)
Length	1 byte
Answer	2 bytes
Detail	Gives back an short, unsigned int with MSB in the first byte.
Example	(Answer) 0x00 0x01 Serial number 1 - this is the prototype.

	Echo Test
Command	HEX 0x65 (ASCII “e”) and one test byte
Length	2 bytes
Answer	1 byte
Detail	Gives back the test byte. Is simply for testing purposes, e.g. to test if someone translate codes.
Example	(Command) 0x65 0x41 (Answer) 0x41

	Display “something”
Command	HEX 0x64 (ASCII “d”) and 7 seven more bytes
Length	8 bytes
Answer	None
Detail	This command writes “direct” to the LED Matrix. The seven Bytes represents the seven lines and the lower five Bits in each Byte the five columns. After a constant time the default display contents will reappear. The time constant will be defined in a later version of this document.
Example	0x64 0x1f 0x00 0x00 x00 0x00 x00 0x1f This will switch on the LED's in the upper and the lower line. Nice to play with.

	Display ASCII char
Command	HEX 0x61 (ASCII “a”) and two following bytes
Length	3 bytes
Answer	None
Detail	The first Byte is the ASCII Code of the char that will be displayed, the second Byte is the time until default display contents will reappear. A new Display ASCII Char command will instantly overwrite the current one. The exact time will be defined in a later version of this document.
Example	0x61 0x41 0xff Displays an “A” with the maximum time.

	Set Clocks
Command	HEX 0x63 (ASCII “c”) and two coded bytes
Length	3 bytes
Answer	None
Detail	The two coded Bytes represent one 16 Bit Word. The first Byte is the most significant part, the second one the least significant part. For each slot there are two bits in this word reserved (Bit 1..0 for Slot 0, Bit 3..2 for Slot 1... Bit 15..14 for Slot 7). With these 2 bits you can choose between the 4 clocks. Binary “00” is for the Clock Crystal the first generates, “01” for the second Crystal, “10” for the third, and “11” for the last one. When this command happens, the clocks on all slots will be stopped, reprogrammed and then the clocks will be restarted. This may cause some issues to the cards with glitch and/or jitter detection, so it may a good idea to put all cards/slots to reset, then probe the right frequency to the one selected card/slot, and after wards “restart” all the cards.
Example	(Answer) 0x63 0xc0 x34 Slot 7 is clocked with the last Crystal (8,00 MHz) Slot 6 is clocked with the first Crystal (3,57 MHz) Slot 5 is clocked with the first Crystal (3,57 MHz) Slot 4 is clocked with the first Crystal (3,57 MHz) Slot 3 is clocked with the first Crystal (3,57 MHz) Slot 2 is clocked with the third Crystal (6,00 MHz) Slot 1 is clocked with the second Crystal (3,68 MHz) Slot 0 is clocked with the first Crystal (3,57 MHz)

	Get the Clock Settings
Command	HEX 0x67 (ASCII “g”)
Length	1 byte
Answer	2 bytes
Detail	The two bytes have the same coding as the “Set Clocks” Command
Example	(Answer) 0x10 0x01 Slot 7 is clocked with the first Crystal (3,57 MHz) Slot 6 is clocked with the second Crystal (3,68 MHz) Slot 5 is clocked with the first Crystal (3,57 MHz) Slot 4 is clocked with the first Crystal (3,57 MHz) Slot 3 is clocked with the first Crystal (3,57 MHz) Slot 2 is clocked with the first Crystal (3,57 MHz) Slot 1 is clocked with the first Crystal (3,57 MHz) Slot 0 is clocked with the second Crystal (3,68 MHz)

	Save the clock Settings
Command	HEX 0x6d (ASCII “m”)
Length	1 byte
Answer	None
Detail	The last chosen clock setting is written to the internal EEPROM. At power up this EEPROM is read out and the clocks are set. As EEPROMs have the feature to get broken if they are written to often, please don't try this as often as possible, just only as often as necessary. Attention needs 'p' command in front.
Example	I hope no one needs one here.

	Set TimeOut
Command	HEX 0x6f (ASCII “o”)
Length	3 bytes
Answer	None
Detail	Set the time the MCR won't answer after a change from an invalid to an valid Signal at the RS232 Interface. This is intended to don't stall an device which maybe waits for something at boot on the serial interface. Attention needs 'p' command in front.
Example	0x6f 0x08 0x01 Set's a timeout of 0x801

	Get TimeOut
Command	HEX 0x72 (ASCII “r”)
Length	1 byte
Answer	2 bytes
Detail	Gives back the saved timeout.
Example	(Answer) 0x08 0x01 Timeout is 0x0801

	Enable EEPROM write
Command	HEX 0x70 (ASCII “p”) and two code bytes (0xab and 0xba)
Length	3 byte
Answer	None
Detail	The command following this command can write the EEPROM (look at the “m” and “r” commands). After any following command (known or unknown) the EEPROM write is disabled and must be enabled again if needed.
Example	0x70 0xab 0xba