DIP SWITCH	3x10100 4x10100 I:10099	0,0x0000 B:00 00			UINT16 R/O	
Returns the current setting of the Dip switches. For ULTRA SLIM IOs The current value of the DIP switches: Bit 0: DIP Switch 1 (=0:OFF, =1:ON) Bit 1: DIP Switch 2 (=0:OFF, =1:ON) Bit 2: DIP Switch 3 (=0:OFF, =1:ON) Bit 3: DIP Switch 4 (=0:OFF, =1:ON) For BIG IOs: The current value of the DIP switches: Bit 0: DIP Switch 1 (=0:OFF, =1:ON) Bit 1: DIP Switch 2 (=0:OFF, =1:ON) Bit 2: DIP Switch 3 (=0:OFF, =1:ON) Bit 3: DIP Switch 4 (=0:OFF, =1:ON) Bit 4: DIP Switch 5 (=0:OFF, =1:ON) Bit 5: DIP Switch 6 (=0:OFF, =1:ON) Bit 6: DIP Switch 7 (=0:OFF, =1:ON) Bit 7: DIP Switch 8 (=0:OFF, =1:ON)						
SOFTWARE RESET						
RESET	1x06001 2x06001 I:6000	0,0x00 B:00		N/A:NO CHANGE	BIT R/W	NO
Performs a software reset, whenever 1 is written to this register. If the host writes to this register 1, the module executes a soft reset (reboot).						
RESET	3x06001 4x06001 I:6000	0,0x0000 B:00 00		N/A:NO CHANGE	UINT16 R/W	NO
Performs a software reset, whenever 1 is written to this register. If the host writes to this register 1, the module executes a soft reset (reboot).						
PRODUCT DATA						
HW_GROUP	3x65201 4x65201 I:65200	8545,0x2161 B:21 61			UINT16 R/O	
This is the group of hardware of the current product						
SW_GROUP	3x65202 4x65202 I:65201	4096,0x1000 B:10 00			UINT16 R/O	
This is the group of software of the current product						
SW_VERSION	3x65203 4x65203 I:65202	4352,0x1100 B:11 00			UINT16 R/O	
		SW VERSION:1.1.0				
This is the current software version of the firmware						
SW_AUTHOR	3x65204 4x65204 I:65203	18771,0x4953 B:49 53			UINT16 R/O	

This is the current software author of the firmware						
MODBUS SETTINGS						
UNIT_ID	3x65222 4x65222 I:65221	65535,0xFFFF B:FF FF			UINT16 R/O	
		UNIT ID:255				
If the host reads this register, the current defined unit ID is returned.						
HINT:This settings will be active after you repower or reset your device !!						
BAUD_RATE	3x65223 4x65223 I:65222	4294967295,0xFFFFFFFF B:FF FF FF FF	38400	38400	UINT32 R/W	NO
		57600Bd		ENTER BAUD RATE		
This is the current configured baud rate for DIP switch mode DIP1=ON, DIP2=ON (default is 57600bd) DIP switch settings: DIP1-DIP2 OFF-OFF:9600bd ON-OFF:19200bd OFF-ON:38400bd ON-ON:default 57600bd or the defined baud rate						
Valid baud rates are: 300bd 600bd 1200bd 2400bd 4800bd 9600bd 19200bd 38400bd all other:57600bd						
HINT:This settings will be active after you repower or reset your device !!						
PARITY	3x65225 4x65225 I:65224	65535,0xFFFF B:FF FF		N/A:NO CHANGE	UINT16 R/W	NO
		NO PARITY		SELECT PARITY		
If the register is read out, the currently set parity of the serial interface is returned. Writing a value to this register will change the new parity in FLASH. This will only take effect after a restart of the module. This can be triggered by writing to the RESET SYSTEM register. Parity values are 0: no parity 1: even parity 2: odd parity						
STOP BITS	3x65226 4x65226 I:65225	65535,0xFFFF B:FF FF		N/A:NO CHANGE	UINT16 R/W	NO
		ONE STOPBIT		SELECT STOPBITS		

f the register is read out, the currently set number of stop bits of the serial interface is returned.
Writing a value to this register will change the new number of stop bits in the FLASH. This will only take effect after a restart of the module. This can be triggered by writing to the RESET SYSTEM register.

Values for stop bits are
1: one stop bit
2: two stop bits

GET VERSION	ASCII READ COMMAND	#VERSION<CR> Result: #VERSION:<VersionHi>,<VersionMed>,<VersionLo><CR>	ASCII	
	TX	#VERSION<CR>		
	RX	#255,VERSION:1.1.0<CR>		
		Current SW version:1.1.0		
Returns the version number of the module VersionHi: Version number high (1..255) VersionMed: Version number medium (1..255) VersionLo: Version number low (1..255)				
GET TYPE	ASCII READ COMMAND	#TYPE<CR> Result: #TYPE:<Type><CR>	ASCII	
	TX	#TYPE<CR>		
	RX	#255,TYPE:RESI-SMI16-SIO<CR>		
		Current module type:RESI-SMI16-SIO		
Returns the current module type				
GET OWNER	ASCII READ COMMAND	#OWNER<CR> Result: #OWNER:<Owner><CR>	ASCII	
	TX	#OWNER<CR>		
	RX	#255,OWNER:RESI<CR>		
		Current owner:RESI		
Returns the current owner of the module				
GET CREATOR	ASCII READ COMMAND	#CREATOR<CR> Result: #CREATOR:<Creator><CR>	ASCII	
	TX	#CREATOR<CR>		
	RX	#255,CREATOR:DI HC SIGL,MSC<CR>		
		Current creator:DI HC SIGL,MSC		
Returns the current creator of the module				
GET COPYRIGHT	ASCII READ COMMAND	#COPYRIGHT<CR> Result: #COPYRIGHT:<Copyright><CR>	ASCII	
	TX	#COPYRIGHT<CR>		
	RX	#255,COPYRIGHT:2016,2020 BY RESI AND DI HC SIGL,MSC WWW.RESI.CC<CR>		
		Current copyright:2016,2020 BY RESI AND DI HC SIGL,MSC WWW.RESI.CC		
Returns the current copyright of the module				
GET DIP SWITCH	ASCII READ COMMAND	#GDIP<CR> Result: #GDIP:<DIPSwitchDec>,<DIPSwitchHex><CR>	ASCII	
	TX	#GDIP<CR>		
	RX	#255,GDIP:4,0x4<CR>		
		Current DIP SWITCH settings:0000.0100		

Returns the current setting of the Dip switches as decimal number and as hexadecimal number. DIPSwitchDec DIPSwitchHex The current value of the DIP switches: Bit 0: DIP Switch 1 (=0:OFF, =1:ON) Bit 1: DIP Switch 2 (=0:OFF, =1:ON) Bit 2: DIP Switch 3 (=0:OFF, =1:ON) Bit 3: DIP Switch 4 (=0:OFF, =1:ON)				
ASCII COMMANDS				
SET MODBUS ADDRESS	ASCII WRITE COMMAND	#SMBADR:<UNITID><CR> Result: #OK<CR>	ASCII	NO
	UNITID	123		
	TX	#SMBADR:123<CR>		
	RX	N/A		
Redefines the unit ID of the module. This change will affect the MODBUS/RTU communication immediately. As a Unit IO you can use the values 0dec to 255dec. HINT: The new settings are activated after a system reboot or power off on cycle!				
SET MODBUS BAUDRATE	ASCII WRITE COMMAND	#SMBBAUD:<BAUD><CR> Result: #OK<CR>	ASCII	NO
	BAUD	128000:128000BD		
	TX	#SMBBAUD:128000<CR>		
	RX	N/A		
Sets a new baud rate in the FLASH For ULTRA SLIM IOs RESI-xxx-SIO: This baudrate is only used, if DIP switch mode DIP1=ON+DIP2=ON (BR) (default is 57600bd) For BIG IOs RESI-xxx-SIO: This baudrate is only used, if DIP switch mode DIP7=ON (PARAMETER) (default is 57600bd) The following baudrates are allowed: 300bd, 600bd, 900bd, 1200bd, 2400bd, 4800bd, 9600bd, 19200bd, 38400bd, 57600bd, 115200bd, 128000bd 230400bd, 250000bd, 256000bd HINT: The new setup parameters will be active after a restart of the module.				
SET MODBUS PARITY	ASCII WRITE COMMAND	#SMBPAR:<PARITY><CR> Result: #OK<CR>	ASCII	NO
	PARITY	ODD:ODD PARITY		
	TX	#SMBPAR:ODD<CR>		
	RX	N/A		

Sets a new parity for the serial interface. MBParity: NONE: no parity EVEN: even parity ODD: odd parity HINT: The new setup parameters will be active after a restart of the module.				
SET MODBUS STOPS	ASCII WRITE COMMAND	#SMBSTOP:<STOPBIT><CR> Result: #OK<CR>	ASCII	NO
	STOPBIT	TWO:TWO STOPBITS		
	TX	#SMBSTOP:TWO<CR>		
	RX	N/A		
Sets a new amount of stop bits for the serial interface. MBStops ONE: one stop bit TWO: two stop bits HINT: The new setup parameters will be active after a restart of the module.				
SET MODBUS PARAMS	ASCII WRITE COMMAND	#SMBPARAMS:<UNITID>,<BAUD>,<PARITY>,<STOPBIT><CR> Result: #OK<CR>	ASCII	NO
	UNITID	3		
	BAUD	115200:115200BD		
	PARITY	EVEN:EVEN PARITY		
	STOPBIT	TWO:TWO STOPBITS		
	TX	#SMBPARAMS:3,115200,EVEN,TWO<CR>		
	RX	N/A		
Sets all parameters for serrial interface				
GET MODBUS ADDRESS	ASCII READ COMMAND	#GMBADR<CR> Result: #GMBADR:<MBUnitDec>,<MBFLASHDec>,<MBUnitHex>,<MBFLASHHex><CR>	ASCII	
	TX	#GMBADR<CR>		
	RX	#255,GMBADR:255,0xFF,65535,0xFFFF<CR>		
		Current MODBUS unit ID:255,0xFF,65535,0xFFFF		
Shows the current used MODBUS/RTU or ASCII unit address and shows also the stored unit address in the FLASH memory, which is only used if the DIP switch for the bus address is set to 0. MBUnitDec,MBUnitHex The current used MODBUS/RTU unit or ASCII address for communication MBFLASHDec,MBFLASHHex The internal stored MODBUS/RTU unit address or ASCII address from the FLASH memory, if the DIP switch DIP3 is OFF.				
GET MODBUS BAUDRATE	ASCII READ COMMAND	#GMBBAUD<CR> Result: #GMBBAUD:<BaudRate><CR>	ASCII	
	TX	#GMBBAUD<CR>		

	RX	#255,GMBBAUD:57600<CR>		
		Current baudrate:57600		
This is the current configured baud rate in the FLASH For ULTRA SLIM IOs RESI-xxx-SIO: This baudrate is only used, if DIP switch mode DIP1=ON+DIP2=ON (BR) (default is 57600bd) For BIG IOs RESI-xxx-SIO: This baudrate is only used, if DIP switch mode DIP7=ON (PARAMETER) (default is 57600bd) The following baudrates are allowed: 300bd, 600bd, 900bd, 1200bd, 2400bd, 4800bd, 9600bd, 19200bd, 38400bd, 57600bd, 115200bd, 128000bd 230400bd, 250000bd, 256000bd				
GET MODBUS PARITY	ASCII READ COMMAND	#GMBPAR<CR> Result: #GMBPAR:<MBParity><CR>	ASCII	
	TX	#GMBPAR<CR>		
	RX	#255,GMBPAR:NONE<CR>		
		Current parity:NONE		
Shows the current configured parity of the serial interface. MBParity NONE: no parity EVEN: even parity ODD: odd parity				
GET MODBUS STOP	ASCII READ COMMAND	#GMBSTOP<CR> Result: #GMBSTOP:<MBStop><CR>	ASCII	
	TX	#GMBSTOP<CR>		
	RX	#255,GMBSTOP:ONE<CR>		
		Current stopbit(s):ONE		
Shows the current configured parity of the serial interface. MBParity NONE: no parity EVEN: even parity ODD: odd parity				
GET MODBUS PARAMS	ASCII READ COMMAND	#GMBPARAMS<CR> Result: #GMBPARAMS:<MBUnitDec>,<MBFLASHDec>,<MBUnitHex>,<MBFLASHHex>, <MBBaudrateDec>,<MBBaudrateHex>,<MBParity>,<MBStops><CR>	ASCII	
	TX	#GMBPARAMS<CR>		
	RX	#255,GMBPARAMS:255,0xFF,65535,0xFFFF,57600,0xE100,NONE,ONE<CR>		
		Current MODBUS unit ID used:255		
		Current MODBUS unit ID in FLASH:65535		
		Current baudrate in FLASH:57600		
		Current parity in FLASH:NONE		
		Current stopbit(s) in FLASH:ONE		
Returns the complete settings for serial interface				
ASCII COMMANDS				
RESET	ASCII WRITE COMMAND	#RST<CR> Result: #OK<CR>	ASCII	NO

	TX	#RST<CR>		
	RX	N/A		
Executes a software reset (Reboot) of the module.				
FACTORY RESET	ASCII WRITE COMMAND	#FRST<CR> Result: #OK<CR>	ASCII	NO
	TX	#FRST<CR>		
	RX	N/A		
Performs a factory reset of all internal saved parameters				

SLAVE ID	3x00001 4x00001 I:0	0,0x0000 B:00 00		0:SLAVE 1	UINT16 R/W	NO
		Current slave ID:0		ENTER SLAVE ID		
Current slave ID for the next slave command slave ID can be 0 to 15						
CURRENT MANUFACTURER ID						
MANUFACTURER ID	3x00002 4x00002 I:1	0,0x0000 B:00 00		0:ALL MANUFACTURERS	UINT16 R/W	NO
		Current manufacturer ID:ALL MANUFACTURERS		SELECT MANUFACTURER ID		
Current manufacturer ID for the next manufacturer command Manufacturer ID can be 0 to 15						
CURRENT SLAVES BITFIELD						
SLAVES BITFIELD	3x00003 4x00003 I:2	0,0x0000 B:00 00		15	UINT16 R/W	NO
		Current slaves IDs:0000.0000.0000.0000		ENTER NEW BITFIELD for SLAVES		
Current bitfield for slaves for the next slaves command Bit 0: Slave ID 0 Bit 1: Slave ID 1 ... Bit 15: Slave ID 15						
POSITIONS+IDs						
POSITION	3x00004 4x00004 I:3	0,0x0000 B:00 00		65535	UINT16 R/W	NO
		New position:0		ENTER NEW POSITION		
New position for POS1 or POS2 commands in the range 0 to 65535 (0=0% 65535=100%)						
SLAT POSITION	3x00005 4x00005 I:4	0,0x0000 B:00 00		255	UINT16 R/W	NO
		New slat position:0		ENTER NEW SLAT POSITION		
New slat position for STEP UP or STEP DOWN commands in the range 0 to 255 (0=0% 255=100%)						
ID	3x00006 4x00006 I:5	0,0x00000000 B:00 00 00 00		100000	UINT32 R/W	NO
		New SMI motor ID:0		ENTER NEW SLAVE ID		
New ID of SMI motor ID for WRITE ID command						
ID	3x00007 4x00007 I:6	0,0x00000000 B:00 00 00 00		100000	UINT32R R/W	NO
		New SMI motor ID:0		ENTER NEW SLAVE ID		
New ID of SMI motor ID for WRITE ID command						

POSITIONS+IDs						
POSITION	3x00011 4x00011 I:10	0,0x00000000 B:00 00 00 00		100000	UINT32 R/W	NO
		New position:0%		ENTER NEW POSITION		
New position for POS1 or POS2 commands in the range 0 to 100000 (0=0% 100000=100%)						
SLAT POSITION	3x00013 4x00013 I:12	0,0x00000000 B:00 00 00 00		511000	UINT32 R/W	NO
		New slat position:0		ENTER NEW SLAT POSITION		
New slat position for STEP UP or STEP DOWN commands in the range 0 to 511000 (0=0% 511000=100%)						
POSITIONS+IDs						
POSITION	3x00021 4x00021 I:20	0,0x00000000 B:00 00 00 00		100000	UINT32R R/W	NO
		New position:0%		ENTER NEW POSITION		
New position for POS1 or POS2 commands in the range 0 to 100000 (0=0% 100000=100%)						
SLAT POSITION	3x00023 4x00023 I:22	0,0x00000000 B:00 00 00 00		511000	UINT32R R/W	NO
		New slat position:0		ENTER NEW SLAT POSITION		
New slat position for STEP UP or STEP DOWN commands in the range 0 to 511000 (0=0% 511000=100%)						
POSITIONS+IDs						
POSITION	3x00031 4x00031 I:30	0.000000,0x00000000 B:00 00 00 00		100,000	FLOAT32 R/W	NO
		New position:0,000%		ENTER NEW POSITION		
New position for POS1 or POS2 commands in the range 0 to 100.000%						
SLAT POSITION	3x00033 4x00033 I:32	0.000000,0x00000000 B:00 00 00 00		511,000	FLOAT32 R/W	NO
		New slat position:0,000%		ENTER NEW SLAT POSITION		
New slat position for STEP UP or STEP DOWN commands in the range 0 to 511.000% (0=0% 511.000=100%)						
POSITIONS+IDs						
POSITION	3x00041 4x00041 I:40	0.000000,0x00000000 B:00 00 00 00		100,000	FLOAT32R R/W	NO
		New position:0,000%		ENTER NEW POSITION		
New position for POS1 or POS2 commands in the range 0 to 100.000%						
SLAT POSITION	3x00043 4x00043 I:42	0.000000,0x00000000 B:00 00 00 00		511,000	FLOAT32R R/W	NO
		New slat position:0,000%		ENTER NEW SLAT POSITION		
New slat position for STEP UP or STEP DOWN commands in the range 0 to 511.000% (0=0% 511.000=100%)						

POSITIONS+IDs						
POSITION	3x00051 4x00051 I:50	0.000000,0x00000000 B:00 00 00 00 00 00 00 00		100,000	DOUBLE64 R/W	NO
		New position:0,000%		ENTER NEW POSITION		
New position for POS1 or POS2 commands in the range 0 to 100.000%						
SLAT POSITION	3x00055 4x00055 I:54	0.000000,0x00000000 B:00 00 00 00 00 00 00 00		511,000	DOUBLE64 R/W	NO
		New slat position:0,000%		ENTER NEW SLAT POSITION		
New slat position for STEP UP or STEP DOWN commands in the range 0 to 511.000% (0=0% 511.000=100%)						
POSITIONS+IDs						
POSITION	3x00061 4x00061 I:60	0.000000,0x00000000 B:00 00 00 00 00 00 00 00		100,000	DOUBLE64R R/W	NO
		New position:0,000%		ENTER NEW POSITION		
New position for POS1 or POS2 commands in the range 0 to 100.000%						
SLAT POSITION	3x00065 4x00065 I:64	0.000000,0x00000000 B:00 00 00 00 00 00 00 00		511,000	DOUBLE64R R/W	NO
		New slat position:0,000%		ENTER NEW SLAT POSITION		
New slat position for STEP UP or STEP DOWN commands in the range 0 to 511.000% (0=0% 511.000=100%)						
SLAVE COMMANDS						
SLAVE 0 COMMAND	3x00101 4x00101 I:100	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		

Starts a new command for selected slave. Each register stands for one slave.
Valid commands are:
0:STOP
1:MOVE UP
2:MOVE DOWN
3:MOVE POS1
4:MOVE POS2
5:MOVE POSITION
6:MOVE UP STEP
7:MOVE DOWN STEP
8:READ ID
9:READ POSITION
10:SYNCHRONIZE
11:DIAGNOSE
12:WRITE POS1
13:WRITE POS2
14:SET SLAVE

SLAVE 1 COMMAND	3x00102 4x00102 I:101	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 2 COMMAND	3x00103 4x00103 I:102	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 3 COMMAND	3x00104 4x00104 I:103	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 4 COMMAND	3x00105 4x00105 I:104	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 5 COMMAND	3x00106 4x00106 I:105	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 6 COMMAND	3x00107 4x00107 I:106	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 7 COMMAND	3x00108 4x00108 I:107	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 8 COMMAND	3x00109 4x00109 I:108	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 9 COMMAND	3x00110 4x00110 I:109	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 10 COMMAND	3x00111 4x00111 I:110	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 11 COMMAND	3x00112 4x00112 I:111	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		

SLAVE 12 COMMAND	3x00113 4x00113 I:112	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 13 COMMAND	3x00114 4x00114 I:113	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 14 COMMAND	3x00115 4x00115 I:114	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE 15 COMMAND	3x00116 4x00116 I:115	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
SLAVE xx COMMAND	3x00200 4x00200 I:199	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
Starts a new command for slave previously stored in register SLAVE ID						
MANUFACTURER COMMANDS						
MANUFACTURER 0 COMMAND	3x00201 4x00201 I:200	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
Starts a new command for selected manufacturer Valid commands are: 0:STOP 1:MOVE UP 2:MOVE DOWN 3:MOVE POS1 4:MOVE POS2 5:MOVE POSITION 6:MOVE UP STEP 7:MOVE DOWN STEP 8:READ ID 9:READ POSITION 10:SYNCHRONIZE 11:DIAGNOSE 12:WRITE POS1 13:WRITE POS2 14:SET SLAVE						
MANUFACTURER 1 COMMAND	3x00202 4x00202 I:201	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		

MANUFACTURER 2 COMMAND	3x00203 4x00203 I:202	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 3 COMMAND	3x00204 4x00204 I:203	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 4 COMMAND	3x00205 4x00205 I:204	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 5 COMMAND	3x00206 4x00206 I:205	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 6 COMMAND	3x00207 4x00207 I:206	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 7 COMMAND	3x00208 4x00208 I:207	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 8 COMMAND	3x00209 4x00209 I:208	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 9 COMMAND	3x00210 4x00210 I:209	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 10 COMMAND	3x00211 4x00211 I:210	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 11 COMMAND	3x00212 4x00212 I:211	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 12 COMMAND	3x00213 4x00213 I:212	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		

MANUFACTURER 13 COMMAND	3x00214 4x00214 I:213	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 14 COMMAND	3x00215 4x00215 I:214	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER 15 COMMAND	3x00216 4x00216 I:215	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
MANUFACTURER x COMMAND	3x00300 4x00300 I:299	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
Starts a new command for manufacturer previously stored in register SLAVE ID						
SLAVES COMMANDS						
SLAVES xx COMMAND	3x00301 4x00301 I:300	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
Starts a new command for slave group previously stored in register SLAVES BITFIELD Valid commands are: 0:STOP 1:MOVE UP 2:MOVE DOWN 3:MOVE POS1 4:MOVE POS2 5:MOVE POSITION 6:MOVE UP STEP 7:MOVE DOWN STEP 8:READ ID 9:READ POSITION 10:SYNCHRONIZE 11:DIAGNOSE 12:WRITE POS1 13:WRITE POS2 14:SET SLAVE						
SLAVES xx COMMAND	3x00400 4x00400 I:399	0,0x0000 B:00 00		2:MOVE DOWN	UINT16 R/W	NO
		Current command:STOP		SELECT COMMAND		
Starts a new command for slave group previously stored in register SLAVES BITFIELD						
SLAVE COMMANDS						

POSITION	3x01001 4x01001 I:1000	0,0x0000 B:00 00		10000	UINT16 R/W	NO
		Current position:0%		ENTER NEW POSITION		
Current position of last READ POSITION command in the range 0 to 10000 (0=0% 10000=100%) or new position for next POS1 or POS2 command						
SLAT POSITION	3x01002 4x01002 I:1001	0,0x0000 B:00 00		255	UINT16 R/W	NO
		Current slat position:0%		ENTER NEW SLAT POSITION		
Current slat position for last READ POSITION command or new position for STEP UP or STEP DOWN commands in the range 0 to 511 (0=0% 511=100%)						
SLAVE COMMAND	3x01003 4x01003 I:1002	0,0x0000 B:00 00		3845	UINT16 R/W	NO
SLAVE ID		15:SLAVE 16				
COMMAND		5:MOVE POSITION				
		Current slave ID:0-SLAVE 1				
		Current command:0-STOP				
Sends the selected command to the selected slave ID Valid commands are: 0:STOP 1:MOVE UP 2:MOVE DOWN 3:MOVE POS1 4:MOVE POS2 5:MOVE POSITION 6:MOVE UP STEP 7:MOVE DOWN STEP 8:READ ID 9:READ POSITION 10:SYNCHRONIZE 11:DIAGNOSE 12:WRITE POS1 13:WRITE POS2 14:SET SLAVE						
MANUFACTURER COMMANDS						
POSITION	3x01011 4x01011 I:1010	0,0x0000 B:00 00		10000	UINT16 R/W	NO
		Current position:0%		ENTER NEW POSITION		
Current position of last READ POSITION command or new position for POS1 or POS2 commands in the range 0 to 10000 (0=0% 10000=100%)						
SLAT POSITION	3x01012 4x01012 I:1011	0,0x0000 B:00 00		255	UINT16 R/W	NO
		Current slat position:0%		ENTER NEW SLAT POSITION		
Current slat position for last READ POSITION command or new position for STEP UP or STEP DOWN commands in the range 0 to 511 (0=0% 511=100%)						

MANUFACTURER COMMAND	3x01013 4x01013 I:1012	0,0x0000 B:00 00		2056	UINT16 R/W	NO
MANUFACTURER ID		8:GROENINGER Antriebstechnik				
COMMAND		8:READ ID				
		Current manufacturer ID:0-ALL MANUFACTURERS				
		Current command:0-STOP				
Sends the selected command to the selected slave IDValid commands are: 0:STOP 1:MOVE UP 2:MOVE DOWN 3:MOVE POS1 4:MOVE POS2 5:MOVE POSITION 6:MOVE UP STEP 7:MOVE DOWN STEP 8:READ ID 9:READ POSITION 10:SYNCHRONIZE 11:DIAGNOSE 12:WRITE POS1 13:WRITE POS2 14:SET SLAVE						
SLAVES COMMAND						
POSITION	3x01021 4x01021 I:1020	0,0x00000000 B:00 00 00 00		100000	UINT32 R/W	NO
		Current position:0,000%		ENTER NEW POSITION		
Current position of last READ POSITION command or new position for POS1 or POS2 commands in the range 0 to 100000 (0=0% 100000=100%)						
SLAT POSITION	3x01023 4x01023 I:1022	0,0x00000000 B:00 00 00 00		511000	UINT32 R/W	NO
		Current slat position:0,000%		ENTER NEW SLAT POSITION		
Current slat position for last READ POSITION command or new position for STEP UP or STEP DOWN commands in the range 0 to 511 (0=0% 511=100%)						
SLAVES COMMAND	3x01025 4x01025 I:1024	0,0x00000000 B:00 00 00 00		4294901760	UINT32 R/W	NO
SLAVE 1		1:YES				
SLAVE 2		1:YES				
SLAVE 3		1:YES				
SLAVE 4		1:YES				
SLAVE 5		1:YES				
SLAVE 6		1:YES				
SLAVE 7		1:YES				
SLAVE 8		1:YES				
SLAVE 9		1:YES				
SLAVE 10		1:YES				

SLAVE 11		1:YES				
SLAVE 12		1:YES				
SLAVE 13		1:YES				
SLAVE 14		1:YES				
SLAVE 15		1:YES				
SLAVE 16		1:YES				
COMMAND		0:STOP				
		Current slave ID:0-SLAVE 1				
		Current command:0-STOP				
Sends the selected command to the selected slaves Valid commands are: 0:STOP 1:MOVE UP 2:MOVE DOWN 3:MOVE POS1 4:MOVE POS2 5:MOVE POSITION 6:MOVE UP STEP 7:MOVE DOWN STEP 8:READ ID 9:READ POSITION 10:SYNCHRONIZE 11:DIAGNOSE 12:WRITE POS1 13:WRITE POS2 14:SET SLAVE						
SLAVES COMMANDS						
POSITION	3x01031 4x01031 I:1030	0,0x00000000 B:00 00 00 00		100000	UINT32R R/W	NO
		Current position:0,000%		ENTER NEW POSITION		
Current position of last READ POSITION command or new position for POS1 or POS2 commands in the range 0 to 100000 (0=0% 100000=100%)						
SLAT POSITION	3x01033 4x01033 I:1032	0,0x00000000 B:00 00 00 00		511000	UINT32R R/W	NO
		Current slat position:0,000%		ENTER NEW SLAT POSITION		
Current slat position for last READ POSITION command or new position for STEP UP or STEP DOWN commands in the range 0 to 511 (0=0% 511=100%)						
SLAVES COMMAND	3x01035 4x01035 I:1034	0,0x00000000 B:00 00 00 00		4294901762	UINT32R R/W	YES
SLAVE 1		1:YES				
SLAVE 2		1:YES				
SLAVE 3		1:YES				
SLAVE 4		1:YES				
SLAVE 5		1:YES				
SLAVE 6		1:YES				
SLAVE 7		1:YES				

SLAVE 8		1:YES			
SLAVE 9		1:YES			
SLAVE 10		1:YES			
SLAVE 11		1:YES			
SLAVE 12		1:YES			
SLAVE 13		1:YES			
SLAVE 14		1:YES			
SLAVE 15		1:YES			
SLAVE 16		1:YES			
COMMAND		2:MOVE DOWN			
		Current slave ID:0-SLAVE 1			
		Current command:0-STOP			

Sends the selected command to the selected slaves
Valid commands are:
0:STOP
1:MOVE UP
2:MOVE DOWN
3:MOVE POS1
4:MOVE POS2
5:MOVE POSITION
6:MOVE UP STEP
7:MOVE DOWN STEP
8:READ ID
9:READ POSITION
10:SYNCHRONIZE
11:DIAGNOSE
12:WRITE POS1
13:WRITE POS2
14:SET SLAVE

PLAIN MODE

ACTIVATE PLAIN MODE	3x05556 4x05556 I:5555	0,0x0000 B:00 00		21321	UINT16 R/W	NO
				CODE FOR PLAIN MODE		

Writing 0x5349 or 21321 enables the PLAIN mode of the gateway.

With this command you can switch to plain mode. You will see it, that the STATE LED flashes very fast!
This mode will use the current setting of the DIP switch tfor RS485 or RS232.
It ALWAYS switches the serial RS232 or RS485 to 2400Baud, 8 data bits, no parity and 1 stop bit!
You can use your SMI Easy Monitor Software now to send commands directly to the SMI bus or to commission your SMI devices or tho change slave addresses.
Because in the SMI Easy Montor software you cannot change the baudrate settings, we do this hard coded settings for 2400bd!

After you have finsihed to work with the external software, you can change a DIP switch or disconnect the device from power. The gateway will leave this mode and restart to normal operation.
Whenever you change a DIP switch or you disconnet the gateway from power, this mode will end. But you can send #RESI<CR> to reboot the gateway for normal mode too!

For Ethernet devices: Every incoming socket data is send to the SMI bus and every received data from the SMI bus is send plain to the socket.
Whenever you change a DIP switch or you disconnet the gateway from power, this mode will end. But you can send #RESI<CR> to reboot the gateway for normal mode too!
As a sign of this plain mode, the STATE LED flashes very fast.

SMI ANSWER						
SMI ANSWER STATE	3x10001 4x10001 I:10000	0,0x0000 B:00 00			UINT16 R/O	
		Last answer:0				
Returns the last aswer state for the recent SMI command The following states are possible: 0:NONE 1:PENDING: No answer yet received 2:TIMEOUT: No answer within time, maybe no SMI device 100:ACK: SMI motor answers with ACK 101:NACK: SMI motor answers with no ACK 102:POSITION: Actual positon from SMI motor received 1000:DIAG_ALL_STOP: Answer to DIAGNOSTIC command, no motor moves 1001:DIAG_ONE_UP: Answer to DIAGNOSTIC command, one motor moves up 1002:DIAG_ONE_DOWN: Answer to DIAGNOSTIC command, one motor moves down 1003:DIAG_DIFF_DIRECTIONS: Answer to DIAGNOSTIC command, motors move in different directions 1004:DIAG_ONE_ERR: Answer to DIAGNOSTIC command, at least one motor has an error 2000:SYNCHRONIZE: Answer to SYNCHRONIZE received 2001:READ_ID: Actual ID from SMI motor received 2002:WRONG_CRC: Answer received, but with invalid bytes						
SMI ANSWER SYNCHRONISATION	3x10002 4x10002 I:10001	0,0x0000 B:00 00			UINT16 R/O	
		Last answer for SYNCHRONIZE:0				
Returns the last received answer value for command SYNCHRONIZE						
SMI ANSWER READ ID	3x10003 4x10003 I:10002	0,0x00000000 B:00 00 00 00			UINT32 R/O	
		Last answer for READ ID:0				
Returns the last received Slave ID for command READ ID						
SMI ANSWER READ ID	3x10005 4x10005 I:10004	0,0x00000000 B:00 00 00 00			UINT32R R/O	
		Last answer for READ ID:0				
Returns the last received Slave ID for command READ ID						
SMI ANSWER READ POSITION	3x10007 4x10007 I:10006	0,0x00000000 B:00 00 00 00			UINT32 R/O	
		Last answer for READ POSITION:0				
Returns the last received position for command READ POSITION in format 0-65535 for 0-100%						
SMI ANSWER READ POSITION	3x10008 4x10008 I:10007	0,0x00000000 B:00 00 00 00			UINT32 R/O	
		Last answer for READ POSITION:0,00%				
Returns the last received position for command READ POSITION in format 0-10000 for 0-100%						

SMI ANSWER READ POSITION	3x10021 4x10021 I:10020	0,0x00000000 B:00 00 00 00			UINT32 R/O	
		Last answer for READ POSITION:0,000%				
Returns the last received position for command READ POSITION in format 0-100000 for 0-100%						
SMI ANSWER READ POSITION	3x10031 4x10031 I:10030	0,0x00000000 B:00 00 00 00			UINT32R R/O	
		Last answer for READ POSITION:0,000%				
Returns the last received position for command READ POSITION in format 0-100000 for 0-100%						
SMI ANSWER READ POSITION	3x10041 4x10041 I:10040	0.000000,0x00000000 B:00 00 00 00			FLOAT32 R/O	
		Last answer for READ POSITION:0,000%				
Returns the last received position for command READ POSITION in format 0-100.000%						
SMI ANSWER READ POSITION	3x10051 4x10051 I:10050	0.000000,0x00000000 B:00 00 00 00			FLOAT32R R/O	
		Last answer for READ POSITION:0,000%				
Returns the last received position for command READ POSITION in format 0-100.000%						
SMI ANSWER READ POSITION	3x10061 4x10061 I:10060	0.000000,0x00000000 B:00 00 00 00 00 00 00 00			DOUBLE64 R/O	
		Last answer for READ POSITION:0,000%				
Returns the last received position for command READ POSITION in format 0-100.000%						
SMI ANSWER READ POSITION	3x10071 4x10071 I:10070	0.000000,0x00000000 B:00 00 00 00 00 00 00 00			DOUBLE64R R/O	
		Last answer for READ POSITION:0,000%				
Returns the last received position for command READ POSITION in format 0-100.000%						

SMI COMMANDS FOR ONE SLAVE				
SMI SLAVE MOVE UP	ASCII WRITE COMMAND	#SMISLVUP:<SLAVE><CR> #SMI SLAVE UP:<SLAVE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	TX	#SMISLVUP:15<CR>		
	RX	N/A		
Starts a SMI move up movement for SMI slave SLAVE: Valid slave address 0-15				
SMI SLAVE MOVE DOWN	ASCII WRITE COMMAND	#SMISLVDN:<SLAVE><CR> #SMI SLAVE DOWN:<SLAVE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	TX	#SMISLVDN:15<CR>		
	RX	N/A		
Starts a SMI move down movement for SMI slave SLAVE: Valid slave address 0-15				
SMI SLAVE STOP	ASCII WRITE COMMAND	#SMISLVSTOP:<SLAVE><CR> #SMI SLAVE STOP:<SLAVE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	TX	#SMISLVSTOP:15<CR>		
	RX	N/A		
Stopps a current SMI movement for SMI slave SLAVE: Valid slave address 0-15				
SMI SLAVE MOVE TO POSITION	ASCII WRITE COMMAND	#SMISLVPOS:<SLAVE>:<POSITION><CR> #SMI SLAVE POS:<SLAVE>:<POSITION><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	POSITION	20,50		
	TX	#SMISLVPOS:15:20.5<CR>		
	RX	N/A		
Starts a SMI movement for SMI slave to the specific position SLAVE: Valid slave address 0-15 POSITION: A valid position between 0 and 100.0 percent				

SMI SLAVE MOVE UP IN STEPS	ASCII WRITE COMMAND	#SMISLVUPST:<SLAVE>:<ANGLE><CR> #SMI SLAVE UP STEP:<SLAVE>:<ANGLE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	ANGLE	10,00		
	TX	#SMISLVUPST:15:10<CR>		
	RX	N/A		
Starts a SMI angle up movement for SMI slave to the specific angle in degrees SLAVE: Valid slave address 0-15 ANGLE: A valid angle between 0 and 511.0 degree				
SMI SLAVE MOVE DOWN IN STEPS	ASCII WRITE COMMAND	#SMISLVDNST:<SLAVE>:<ANGLE><CR> #SMI SLAVE DN STEP:<SLAVE>:<ANGLE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	ANGLE	10,00		
	TX	#SMISLVDNST:15:10<CR>		
	RX	N/A		
Starts a SMI angle up movement for SMI slave to the specific angle in degrees SLAVE: Valid slave address 0-15 ANGLE: A valid angle between 0 and 511.0 degree				
SMI SLAVE WRITE POSITION1	ASCII WRITE COMMAND	#SMISLVWPOS1:<SLAVE>:<POSITION><CR> #SMI SLAVE WRITE POS1:<SLAVE>:<POSITION><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	POSITION	20,00		
	TX	#SMISLVWPOS1:15:20<CR>		
	RX	N/A		
Stores a new position in the SMI actuator as position 1 SLAVE: Valid slave address 0-15 POSITION: A new position between 0 and 100.0 percent				
SMI SLAVE MOVE TO POSITION1	ASCII WRITE COMMAND	#SMISLVPOS1:<SLAVE><CR> #SMI SLAVE POS1:<SLAVE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	TX	#SMISLVPOS1:15<CR>		
	RX	N/A		
Starts a SMI movement for SMI slave to stored position 1 of the SMI actor SLAVE: Valid slave address 0-15				

SMI SLAVE WRITE POSITION2	ASCII WRITE COMMAND	#SMISLVWPOS2:<SLAVE>:<POSITION><CR> #SMI SLAVE WRITE POS2:<SLAVE>:<POSITION><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	POSITION	30,00		
	TX	#SMISLVWPOS2:15:30<CR>		
	RX	N/A		
Stores a new position in the SMI actuator as position 2 SLAVE: Valid slave address 0-15 POSITION: A new position between 0 and 100.0 percent				
SMI SLAVE MOVE TO POSITION2	ASCII WRITE COMMAND	#SMISLVPOS2:<SLAVE><CR> #SMI SLAVE POS2:<SLAVE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	TX	#SMISLVPOS2:15<CR>		
	RX	N/A		
Starts a SMI movement for SMI slave to stored position 2 of the SMI actor SLAVE: Valid slave address 0-15				
SMI SLAVE READ POSITION	ASCII WRITE COMMAND	#SMISLVRPOS:<SLAVE><CR> #SMI SLAVE READ POS:<SLAVE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	TX	#SMISLVRPOS:15<CR>		
	RX	#255,OK<CR>		
Sends a command to read out the actual position of the affected SMI slave SLAVE: Valid slave address 0-15				
SMI SLAVE DIAGNOSE	ASCII WRITE COMMAND	#SMISLVDIAG:<SLAVE><CR> #SMI SLAVE DIAGNOSE:<SLAVE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	TX	#SMISLVDIAG:15<CR>		
	RX	#255,OK<CR>		
Sends a command to the affected SMI slave, to request diagnose information about the SMI slave. SLAVE: Valid slave address 0-15				
SMI SLAVE SYNCHRONIZE	ASCII WRITE COMMAND	#SMISLVSYNC:<SLAVE><CR> #SMI SLAVE SYNCHRONIZE:<SLAVE><CR> Result: #OK<CR>	ASCII	NO

	SLAVE	15		
	TX	#SMISLVSYNC:15<CR>		
	RX	#255,OK<CR>		
Sends a SYNCHRONIZE command to the affected SMI slave, to request information about the SMI slave. SLAVE: Valid slave address 0-15 The slave will return 1 byte data: The lower 4 bits stand for the manufacturer ID, the upper 4 bits represent the actor type				
SMI SLAVE READ ID	ASCII WRITE COMMAND	#SMISLVRID:<SLAVE><CR> #SMI SLAVE READ ID:<SLAVE><CR> Result: #OK<CR>	ASCII	NO
	SLAVE	15		
	TX	#SMISLVRID:15<CR>		
	RX	#255,OK<CR>		
Sends a READ ID command to the affected SMI slave, to request the 32 bit unique ID of the SMI slave. SLAVE: Valid slave address 0-15 The slave will return 4 byte data: The four bytes for the 32 bit unique ID of the SMI actor				
SMI COMMANDS FOR MANUFACTURER				
SMI MANUFACTURER MOVE UP	ASCII WRITE COMMAND	#SMIMANUUP:<MANUFACTURER><CR> #SMI MANUFACTURER UP:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANUUP:15<CR>		
	RX	N/A		
Starts a SMI move up movement for SMI slaves with suitable manufacturer code MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers				
SMI MANUFACTURER MOVE DOWN	ASCII WRITE COMMAND	#SMIMANUDN:<MANUFACTURER><CR> #SMI MANUFACTURER DOWN:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANUDN:15<CR>		
	RX	N/A		
Starts a SMI move down movement for SMI slaves with suitable manufacturer code MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers				
SMI MANUFACTURER STOP	ASCII WRITE COMMAND	#SMIMANUSTOP:<MANUFACTURER><CR> #SMI MANUFACTURER STOP:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANUSTOP:15<CR>		
	RX	N/A		

Stops a current SMI movement for SMI slaves with suitable manufacturer code MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers				
SMI MANUFACTURER MOVE TO POSITION	ASCII WRITE COMMAND	#SMIMANUPOS:<MANUFACTURER>:<POSITION><CR> #SMI MANUFACTURER POS:<MANUFACTURER>:<POSITION><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	POSITION	20,50		
	TX	#SMIMANUPOS:15:20.5<CR>		
	RX	N/A		
Starts a SMI movement for SMI slaves with suitable manufacturer code to the specific position MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers POSITION: A valid position between 0 and 100.0 percent				
SMI MANUFACTURER MOVE UP IN STEPS	ASCII WRITE COMMAND	#SMIMANUUPST:<MANUFACTURER>:<ANGLE><CR> #SMI MANUFACTURER UP STEP:<MANUFACTURER>:<ANGLE><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	ANGLE	10,00		
	TX	#SMIMANUUPST:15:10<CR>		
	RX	N/A		
Starts a SMI angle up movement for SMI slaves with suitable manufacturer code to the specific angle in degrees MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers ANGLE: A valid angle between 0 and 511.0 degree				
SMI MANUFACTURER MOVE DOWN IN STEPS	ASCII WRITE COMMAND	#SMIMANUDNST:<MANUFACTURER>:<ANGLE><CR> #SMI MANUFACTURER DN STEP:<MANUFACTURER>:<ANGLE><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	ANGLE	10,00		
	TX	#SMIMANUDNST:15:10<CR>		
	RX	N/A		
Starts a SMI angle up movement for SMI slaves with suitable manufacturer code to the specific angle in degrees MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers ANGLE: A valid angle between 0 and 511.0 degree				
SMI MANUFACTURER WRITE POSITION1	ASCII WRITE COMMAND	#SMIMANUWPOS1:<MANUFACTURER>:<POSITION><CR> #SMI MANUFACTURER WRITE POS1:<MANUFACTURER>:<POSITION><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	POSITION	20,00		
	TX	#SMIMANUWPOS1:15:20<CR>		

	RX	N/A		
Stores a new position in the SMI slaves with suitable manufacturer code as position 1 MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers POSITION: A new position between 0 and 100.0 percent				
SMI MANUFACTURER MOVE TO POSITION1	ASCII WRITE COMMAND	#SMIMANUPOS1:<MANUFACTURER><CR> #SMI MANUFACTURER POS1:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANUPOS1:15<CR>		
	RX	N/A		
Starts a SMI movement for SMI slaves with suitable manufacturer code to stored position 1 of the SMI actor MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers				
SMI MANUFACTURER WRITE POSITION2	ASCII WRITE COMMAND	#SMIMANUWPOS2:<MANUFACTURER>:<POSITION><CR> #SMI MANUFACTURER WRITE POS2:<MANUFACTURER>:<POSITION><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	POSITION	30,00		
	TX	#SMIMANUWPOS2:15:30<CR>		
	RX	N/A		
Stores a new position in the SMI slaves with suitable manufacturer code as position 2 MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers POSITION: A new position between 0 and 100.0 percent				
SMI MANUFACTURER MOVE TO POSITION2	ASCII WRITE COMMAND	#SMIMANUPOS2:<MANUFACTURER><CR> #SMI MANUFACTURER POS2:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANUPOS2:15<CR>		
	RX	N/A		
Starts a SMI movement for SMI slaves with suitable manufacturer code to stored position 2 of the SMI actor MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers				
SMI MANUFACTURER READ POSITION	ASCII WRITE COMMAND	#SMIMANURPOS:<MANUFACTURER><CR> #SMI MANUFACTURER READ POS:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANURPOS:15<CR>		
	RX	#255,OK<CR>		
Sends a command to read out the actual position of the affected SMI slaves with suitable manufacturer code MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers				

SMI MANUFACTURER DIAGNOSE	ASCII WRITE COMMAND	#SMIMANUDIAG:<MANUFACTURER><CR> #SMI MANUFACTURER DIAGNOSE:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANUDIAG:15<CR>		
	RX	#255,OK<CR>		
Sends a command to the affected SMI slaves with suitable manufacturer code, to request diagnose information about the SMI slave. MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers				
SMI MANUFACTURER SYNCHRONIZE	ASCII WRITE COMMAND	#SMIMANUSYNC:<MANUFACTURER><CR> #SMI MANUFACTURER SYNCHRONIZE:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANUSYNC:15<CR>		
	RX	#255,OK<CR>		
Sends a SYNCHRONIZE command to the affected SMI slaves with suitable manufacturer code, to request information about the SMI slave. MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers The slave will return 1 byte data: The lower 4 bits stand for the manufacturer ID, the upper 4 bits represent the actor type				
SMI MANUFACTURER READ ID	ASCII WRITE COMMAND	#SMIMANURID:<MANUFACTURER><CR> #SMI MANUFACTURER READ ID:<MANUFACTURER><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	TX	#SMIMANURID:15<CR>		
	RX	#255,OK<CR>		
Sends a READ ID command to the affected SMI slaves with suitable manufacturer code, to request the 32 bit unique ID of the SMI slave. MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers The slave will return 4 byte data: The four bytes for the 32 bit unique ID of the SMI actor				
SMI MANUFACTURER SET SLAVE	ASCII WRITE COMMAND	#SMIMANUSSLV:<MANUFACTURER>:<ID>:<SLAVE><CR> #SMI MANUFACTURER SET SLAVE:<MANUFACTURER>:<ID>:<POSITION><CR> Result: #OK<CR>	ASCII	NO
	MANUFACTURER	15		
	ID	0x12345678		
	SLAVE	15		
	TX	#SMIMANUSSLV:15:0x12345678:15<CR>		
	RX	N/A		
Stores a new slave address in the SMI actuator with the suitable manufacturer code and specified 32 bit unique ID. MANUFACTURER: Valid manufacturer code from 1-15 or 0 for all manufacturers ID: the 32 bit unique ID of the SMI actor as hexadecimal value (e.g. 0x12345678) SLAVE: Valid slave address 0-15				
SMI COMMANDS FOR A GROUP OF SLAVES				

SMI SLAVES MOVE UP	ASCII WRITE COMMAND	#SMISLVSUP:<SLAVELIST><CR> #SMI SLAVES UP:<SLAVELIST><CR> Result: #OK<CR>	ASCII	NO
	SLAVELIST	0,1,3,7,9,15		
	TX	#SMISLVSUP:0,1,3,7,9,15<CR>		
	RX	N/A		
Starts a SMI move up movement for SMI slave SLAVELIST: Valid list of affected slave addresses <Slave1>,<Slave2>,...,<Slaven> e.g. 0,3,7,15				
SMI SLAVES MOVE DOWN	ASCII WRITE COMMAND	#SMISLVSDN:<SLAVELIST><CR> #SMI SLAVES DOWN:<SLAVELIST><CR> Result: #OK<CR>	ASCII	NO
	SLAVELIST	0,1,3,7,9,15		
	TX	#SMISLVSDN:0.1.3.7.9.15<CR>		
	RX	N/A		
Starts a SMI move down movement for SMI slave SLAVELIST: Valid list of affected slave addresses <Slave1>,<Slave2>,...,<Slaven> e.g. 0,3,7,15				
SMI SLAVES STOP	ASCII WRITE COMMAND	#SMISLVSSTOP:<SLAVELIST><CR> #SMI SLAVES STOP:<SLAVELIST><CR> Result: #OK<CR>	ASCII	NO
	SLAVELIST	0,1,3,7,9,15		
	TX	#SMISLVSSTOP:0,1,3,7,9,15<CR>		
	RX	N/A		
Stops a current SMI movement for SMI slave SLAVELIST: Valid list of affected slave addresses <Slave1>,<Slave2>,...,<Slaven> e.g. 0,3,7,15				
SMI SLAVES MOVE TO POSITION	ASCII WRITE COMMAND	#SMISLVSPOS:<SLAVELIST>:<POSITION><CR> #SMI SLAVES POS:<SLAVELIST>:<POSITION><CR> Result: #OK<CR>	ASCII	NO
	SLAVELIST	0,1,3,7,9,15		
	POSITION	20,50		
	TX	#SMISLVSPOS:0,1,3,7,9,15:20.5<CR>		
	RX	N/A		
Starts a SMI movement for SMI slave to the specific position SLAVELIST: Valid list of affected slave addresses <Slave1>,<Slave2>,...,<Slaven> e.g. 0,3,7,15 POSITION: A valid position between 0 and 100.0 percent				
SMI SLAVES MOVE UP IN STEPS	ASCII WRITE COMMAND	#SMISLVSUPST:<SLAVELIST>:<ANGLE><CR> #SMI SLAVES UP STEP:<SLAVELIST>:<ANGLE><CR> Result: #OK<CR>	ASCII	NO

	SLAVELIST	0,1,3,7,9,15		
	ANGLE	10,00		
	TX	#SMISLVSUPST:0,1,3,7,9,15:10<CR>		
	RX	N/A		
Starts a SMI angle up movement for SMI slave to the specific angle in degrees SLAVELIST: Valid list of affected slave addresses <Slave1>,<Slave2>,...,<Slaven> e.g. 0,3,7,15 ANGLE: A valid angle between 0 and 511.0 degree				
SMI SLAVES MOVE DOWN IN STEPS	ASCII WRITE COMMAND	#SMISLVSDNST:<SLAVELIST>:<ANGLE><CR> #SMI SLAVES DN STEP:<SLAVELIST>:<ANGLE><CR> Result: #OK<CR>	ASCII	NO
	SLAVELIST	0,1,3,7,9,15		
	ANGLE	10,00		
	TX	#SMISLVSDNST:0,1,3,7,9,15:10<CR>		
	RX	N/A		
Starts a SMI angle up movement for SMI slave to the specific angle in degrees SLAVELIST: Valid list of affected slave addresses <Slave1>,<Slave2>,...,<Slaven> e.g. 0,3,7,15 ANGLE: A valid angle between 0 and 511.0 degree				
SMI SLAVES MOVE TO POSITION1	ASCII WRITE COMMAND	#SMISLVSPOS1:<SLAVELIST><CR> #SMI SLAVES POS1:<SLAVELIST><CR> Result: #OK<CR>	ASCII	NO
	SLAVELIST	0,1,3,7,9,15		
	TX	#SMISLVSPOS1:0,1,3,7,9,15<CR>		
	RX	N/A		
Starts a SMI movement for SMI slave to stored position 1 of the SMI actor SLAVELIST: Valid list of affected slave addresses <Slave1>,<Slave2>,...,<Slaven> e.g. 0,3,7,15				
SMI SLAVES MOVE TO POSITION2	ASCII WRITE COMMAND	#SMISLVSPOS2:<SLAVELIST><CR> #SMI SLAVES POS2:<SLAVELIST><CR> Result: #OK<CR>	ASCII	NO
	SLAVELIST	0,1,3,7,9,15		
	TX	#SMISLVSPOS2:0,1,3,7,9,15<CR>		
	RX	N/A		
Starts a SMI movement for SMI slave to stored position 2 of the SMI actor SLAVELIST: Valid list of affected slave addresses <Slave1>,<Slave2>,...,<Slaven> e.g. 0,3,7,15				
SMI COMMANDS FOR SENDING SMI FRAME				
SMI SEND FRAME	ASCII WRITE COMMAND	#SMISND:<BYTE1>,<BYTE2>,..,<BYTEN> or CRC<CR> #SMI SEND FRAME:<BYTE1>,<BYTE2>,..,<BYTEN> or CRC<CR> Result: #OK<CR>	ASCII	NO
	BYTELIST	60,05,CRC		

	TX	#SMISND:60,05,CRC		
	RX	N/A		
Sends a special SMI telegram using a list of bytes. Each byte consists out of a 8 bit hexadecial number in the range of 00-FF separated with a comma. The last byte can be 00-FF or CRC for indicating, that the gateway should calculate a checksum for the frame and use this value				
SMI COMMANDS FOR ANSWERS				
SMI CHECK ANSWER	ASCII READ COMMAND	#SMICHKANS<CR> #SMI CHECK ANSWER<CR> Result: #SMICHKANS:<AnswerText><CR>	ASCII	
	TX	#SMICHKANS<CR>		
	RX	#255,SMICHKANS:POSITION:13434,0x347A,20.50<CR>		
		Answer to last SMI command:POSITION:13434,0x347A,20.50		
Returns the answer state of the last SMI communication NONE: Nothing was communicated on the SMI since last restar of gateway PENDING: A SMI communication is currently active and not finished TIMEOUT: The last SMI protocol generates a timeout without an answer of any SMI actors WRONG_CRC:The last received SMI answer is corrupted, the checksum of the answer is wrong ACK: A positive acknowledgement was received NACK: A negative acknowledgement was received POSITION:<Position>: The last received SMI answer contains a Position in percent (0-100.0) DIAG_ALL_STOP: The last received SMI answer indicates, that all engines are stopped DIAG_ONE_UP: The last received SMI answer indicates, that at least one engine moves up DIAG_ONE_DOWN: The last received SMI answer indicates, that at least one engine moves down DIAG_DIFFERENT_DIRECTIONS: The last received SMI answer indicates, that at least two engines move in different directions DIAG_ONE_ERR: The last received SMI answer indicates, that at least one engine reports an internal error SYNCHRONIZE: The last received SMI answer reports the byte for the SYNCHRONIZE command. The lower 4 bits are the manufacturer code, the upper 4 bits are the type of the SMI actor READ_ID:The last received SMI answer reports the four bytes for the READ_ID command. The four bytes for the 32 bit unique ID of the SMI actor.				
SMI ANSWER	ASCII READ COMMAND	#SMIANS<CR> #SMI ANSWER<CR> Result: #SMIANS:<Length>:<Byte1>,<Byte2>,..,<ByteN><CR>	ASCII	
	TX	#SMIANS<CR>		
	RX	#255,SMIANS:5:EF,45,34,7A,1E<CR>		
		Answer frame to last SMI command:5:EF,45,34,7A,1E		
Returns the answer telegram of the last SMI communication. Lenght: The amount of received bytes as a decimal number Byte1-ByteN: The bytes of the protocol as hexadecimal 8 bit values each				
SMI COMMANDS FOR TRANSPARENT GATEWAY				
SMI PLAIN MODE	ASCII READ COMMAND	#SMIPLAINMODE<CR> #SMI PLAIN MODE<CR> Result: NONE, Gateway is in PLAIN mode with 2400Bd, 8 data bits, no parity, 1 stop bit	ASCII	
	TX	#SMIPLAINMODE<CR>		
	RX	#255,SMICHKANS:POSITION:13434,0x347A,20.50<CR>		

With this command you can switch to plain mode. You will see it, that the STATE LED flashes very fast!
This mode will use the current setting of the DIP switch tfor RS485 or RS232.
It ALWAYS switches the serial RS232 or RS485 to 2400Baud, 8 data bits, no parity and 1 stop bit!
You can use your SMI Easy Monitor Software now to send commands directly to the SMI bus or to commission your SMI devices or tho change slave addresses.
Because in the SMI Easy Montor software you cannot change the baudrate settings, we do this hard coded settings for 2400bd!

After you have finsihed to work with the external software, you can change a DIP switch or disconnect the device from power. The gateway will leave this mode and restart to normal operation.
Whenever you change a DIP switch or you disconnet the gateway from power, this mode will end. But you can send #RESI<CR> to reboot the gateway for normal mode too!

For Ethernet devices: Every incoming socket data is send to the SMI bus and every received data from the SMI bus is send plain to the socket.
Whenever you change a DIP switch or you disconnet the gateway from power, this mode will end. But you can send #RESI<CR> to reboot the gateway for normal mode too!
As a sign of this plain mode, the STATE LED flashes very fast.

