CONVERTER STATUS						
CONVERTER STATUS	3x05051 4x05051 I:5050	0,0x0000 B:00 00			UINT16 R/O	
DIP SWITCH	3x10010 4x10010 I:10009	11,0x000B B:00 0B			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)						
SOFTWARE RESET						
RESET	1x06001 2x06001 I:6000	0,0x00 B:00		N/A:NO CHANGE	BIT R/W	YES
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	YES
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	8337,0x2091 B:20 91			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 cuirrent software version of the firmware						
SW_AUTHOR	3x65204 4x65204 I:65203	18771,0x4953 B:49 53			UINT16 R/O	
This is the cuirrent software author of the firmware						
MODBUS SETTINGS						
UNIT_ID	3x65222 4x65222 I:65221	65535,0xFFFF B:FF FF		N/A:NO CHANGE	UINT16 R/W	NO

		UNIT ID:255				
f the host reads this register, the current programmed unit ID is returned. All values above unit ID 255 define also the unit ID 255. If the host write a new value into this register, the new value will be stored in the FLASH as the new unit ID. The new unit ID is activated after a power off/power on cycle or a software reboot of the module. The host can execute a reboot in writing to the register RESET SYSTEM. NOTE:DIP switch 4 must set to OFF to activate this unit ID, otherwise the unit ID is 255.						
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						

ASCII COMMANDS				
GET VERSION	**ASCII READ COMMAND**	#VERSION<CR> #VER<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> #TYP<CR> Result: #TYPE:<Type><CR>	ASCII	
	TX	#TYPE<CR>		
	RX	#255,TYPE:RESI-1RO-SIO<CR>		
		Current module type:RESI-1RO-SIO		
Returns the current module type				
GET OWNER	**ASCII READ COMMAND**	#OWNER<CR> #OWN<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> #CRE<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> #COPY<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**	#GET DIP<CR> #GDIP<CR> Result: #GDIP:<DIPSwitchDec>,<DIPSwitchHex><CR>	ASCII	
	TX	#GET DIP<CR>		
	RX	#255,GDIP:15,0xF<CR>		
		Current DIP SWITCH settings:1111		
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**	#SET MODBUS ADDRESS:<UNITID><CR> #SETMBADR:<UNITID><CR> Result: #OK<CR>	ASCII	NO
	UNITID	1		
	TX	#SET MODBUS ADDRESS:1<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**	#SET MODBUS BAUDRATE:<BAUD><CR> #SETMBBAUD:<BAUD><CR> Result: #OK<CR>	ASCII	NO
	BAUD	57600:57600BD		
	TX	#SET MODBUS BAUDRATE:57600<CR>		
	RX	N/A		
Sets a new baudrate for the serial interface, if DIP Switches DIP1=ON and DIP2=ON. The following baudrates are allowed: 300bd 600bd 1200bd 2400bd 4800bd 9600bd 19200bd 38400bd all others are interpreted as 57600bd HINT: The new setup parameters will be active after a restart of the module.				

SET MODBUS PARITY	**ASCII WRITE COMMAND**	#SET MODBUS PARITY:<PARITY><CR> #SETMBPAR:<PARITY><CR> Result: #OK<CR>	ASCII	NO
	PARITY	NONE:NO PARITY		
	TX	#SET MODBUS PARITY:NONE<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**	#SET MODBUS STOP:<STOPBIT><CR> #SETMBSTOP:<STOPBIT><CR> Result: #OK<CR>	ASCII	NO
	STOPBIT	ONE:ONE STOPBIT		
	TX	#SET MODBUS STOP:ONE<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**	#SET MODBUS PARAMS:<UNITID>,<BAUD>,<PARITY>,<STOPBIT><CR> #SETMBPARAMS:<UNITID>,<BAUD>,<PARITY>,<STOPBIT><CR> Result: #OK<CR>	ASCII	NO
	UNITID	1		
	BAUD	57600:57600BD		
	PARITY	NONE:NO PARITY		
	STOPBIT	ONE:ONE STOPBIT		
	TX	#SET MODBUS PARAMS:1,57600,NONE,ONE<CR>		
	RX	N/A		
Sets all parameters for serrial interface				
GET MODBUS ADDRESS	**ASCII READ COMMAND**	#GET MODBUS ADDRESS<CR> #GMBADR<CR> Result: #GMBADR:<MBUnitDec>,<MBFLASHDec>,<MBUnitHex>,<MBFLASHHex><CR>	ASCII	
	TX	#GET MODBUS ADDRESS<CR>		
	RX	#255,GMBADR:255,0xFF,65535,0xFFFF<CR>		
		Current MODBUS unit ID for DIP4=OFF: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**	#GET MODBUS BAUDRATE<CR> #GMBBAUD<CR> Result: #GMBBAUD:<BaudRate><CR>	ASCII	
	TX	#GET MODBUS BAUDRATE<CR>		
	RX	#255,GMBBAUD:57600<CR>		
		Current baudrate for DIP1+2=ON:57600		

Returns the current defined baud rate for the serial interface, if DIP switches DIP1=ON and DIP2=ON.
The following baudrates are allowed:
300bd
600bd
1200bd
2400bd
4800bd
9600bd
19200bd
38400bd
all others are interpreted as 57600bd

GET MODBUS PARITY	**ASCII READ COMMAND**	#GET MODBUS PARITY<CR> #GMBPAR<CR> Result: #GMBPAR:<MBParity><CR>	ASCII	
	TX	#GET MODBUS PARITY<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**	#GET MODBUS STOP<CR> #GMBSTOP<CR> Result: #GMBSTOP:<MBStop><CR>	ASCII	
	TX	#GET MODBUS STOP<CR>		
	RX	#255,GMBSTOP:ONE<CR>		
		Current stopbit(s):ONE		

Returns the current configured amount of stop bits for the serial interface.
MBStops
ONE: one stop bit
TWO: two stop bits

GET MODBUS PARAMS	**ASCII READ COMMAND**	#GET MODBUS PARAMS<CR> #GMBPARAMS<CR> Result: #GMBPARAMS:<MBUnitDec>,<MBFLASHDec>,<MBUnitHex>,<MBFLASHHex>, <MBBaudrateDec>,<MBBaudrateHex>,<MBParity>,<MBStops><CR>	ASCII	
	TX	#GET MODBUS PARAMS<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**	#RESET<CR> #RST<CR> Result: #OK<CR>	ASCII	NO
	TX	#RESET<CR>		
	RX	N/A		
Executes a software reset (Reboot) of the module.				
FACTORY RESET	**ASCII WRITE COMMAND**	#FACTORY RESET<CR> #FRST<CR> Result: #OK<CR>	ASCII	NO
	TX	#FACTORY RESET<CR>		
	RX	N/A		

YOUR INDIVIDUAL MODBUS REGISTER MAPPING IS USED, WHICH YOU HAVE CONFIGURED!

ASCII COMMANDS				
GET CONFIG SIZE	**ASCII READ COMMAND**	#GET CONFIG SIZE<CR> #GCS<CR> Result: #GCS:<ConfigSize><CR>	ASCII	
	TX	#GET CONFIG SIZE<CR>		
	RX	#255,GCS:1<CR>		
		Current configured lines in the configuration:1 lines		
This command returns the current amount of config lines stored in the gateway				
READ CONFIG LINE	**ASCII READ COMMAND**	#READ CONFIG:<LINE><CR> #RC:<LINE><CR> Result: #KNX:I:<LINE>=<MBREGISTER>,<MBDATATYPE>,<MBREGISTERCOUNT>,<INTERVAL>, <KNXGROUP>,<KNXDATATYPE>,<KNXDIRECTION>,<FACTOR><CR>	ASCII	
	LINE	1		
	TX	#READ CONFIG:1<CR>		
	RX	#255,KNX:I:1=5,UINT16,1,0,12.3.38,FLOAT16,RW,1.500000<CR>		
		Current configuration line number:1 lines		
		Current MODBUS register index:5		
		Current MODBUS data type:UINT16		
		Current MODBUS register count:1		
		Current interval:0s		
		Current KNX group address:12.3.38		
		Current KNX group data type:FLOAT16		
		Current KNX transfer direction:RW		
		Current KNX multiplication factor:1.500000		
This command returns the desired configuration line from the internal FLASH memory.				
<LINE> stands for the Index in the configuration table starting with 1 and ending with n according to the actual size of the configuration table size as a decimal number.				
<MBREGISTER> stands for the starting index of the holding registers, starting with 1 for the first holding register 4x00001 and ending by 65535 for the last holding register 4x65535 as a decimal number.				
<MBDATATYPE> defines a datatype for the MODBUS registers. This is an ASCII text string in capital letters. Choose one of the following strings: UINT16, SINT16, UINT32, SINT32, UINT32R, SINT32R, FLOAT32, FLOAT32R, DOUBLE64, DOUBLE64R, GENERIC, ASCII or ERR. See the explanation of the MODBUS datatypes for more details about this strings.				
<MBREGISTERCOUNT> defines the amount of MODBUS holding registers, which are used by this configuration entry. For example a UINT16 needs 1 register, a UINT32 or a FLOAT32 needs 2 registers.				
<INTERVAL> stands for a time interval in seconds as a decimal number for an automatic polling request on the KNX bus for this KNX group address. This is for future use and not used now!				
<KNXGROUP> defines the KNX group address with the format <Hi>.<Medium>.<Low>. KNX group addresses in the range from 0.0.0 to 31.7.255 are valid here.				

<KNXDATATYPE> is a string defining the data type of the incoming or outgoing KNX data.
The system uses the following ASCII strings in capital letters:
ERR, BIT, TWOBITS, FOURBITS, SIXBITS, CHARACTER, UINT8, SINT8, UNIT16, SINT16, FLOAT16, TIME, DATE, UINT32, SINT32, FLOAT32, STRING, GENERIC and DATETIME.
See the explanation of the KNX datatypes for more details about this strings.

<KNXDIRECTION> is an ASCII text string defining the communication direction for this entry on the KNX bus.
The following ASCII string in capital letters are valid:
ERR, R, W, RW
See the explanation for KNX directions for more details about this string.

<FACTOR> is a float value defining the multiplication factor for incoming KNX telegrams and the division factor for outgoing KNX telegrams.
Use the float format 1234.567. Don't use a comma as a comma sign!

CLEAR CONFIG	**ASCII WRITE COMMAND**	#CLEAR CONFIG<CR> #CC<CR> Result: #OK<CR>	ASCII	NO
	TX	#CLEAR CONFIG<CR>		
	RX	N/A		
This command clears the complete configuration in the internal FLASH memory of the gateway				
ADD CONFIG LINE	**ASCII WRITE COMMAND**	#ADD CONFIG:<MBREGISTER>,<MBDATATYPE>,<INTERVAL>, <KNXGROUP>,<KNXDATATYPE>,<KNXDIRECTION>,<FACTOR><CR> #AC:<MBREGISTER>,<MBDATATYPE>,<INTERVAL>, <KNXGROUP>,<KNXDATATYPE>,<KNXDIRECTION>,<FACTOR><CR> Result: #OK:<LINE><CR>	ASCII	NO
	MBREGISTER	5		
	MBDATATYPE	UINT16		
	INTERVAL	0		
	KNXGROUP	12.3.38		
	KNXDATATYPE	FLOAT16		
	KNXDIRECTION	READ-WRITE		
	FACTOR	1,5		
	TX	#ADD CONFIG:5,UINT16,0,12.3.38,FLOAT16,READ-WRITE,1.5<CR>		
	RX	N/A		
This command adds this line to the configuration in the internal FLASH memory of the gateway				

<MBREGISTER> stands for the starting index of the holding registers, starting with 1 for the first holding register 4x00001 and ending by 65535 for the last holding register 4x65535 as a decimal number.

<MBDATATYPE> defines a datatype for the MODBUS registers. This is an ASCII text string in capital letters.
Choose one of the following strings:
UINT16, SINT16, UINT32, SINT32, UINT32R, SINT32R, FLOAT32, FLOAT32R, DOUBLE64, DOUBLE64R, GENERIC, ASCII or ERR.
See the explanation of the MODBUS datatypes for more details about this strings.

<INTERVAL> stands for a time interval in seconds as a decimal number for an automatic polling request on the KNX bus for this KNX group address. This is for future use and not used now!

<KNXGROUP> defines the KNX group address with the format <Hi>.<Medium>.<Low>.
KNX group addresses in the range from 0.0.0 to 31.7.255 are valid here.

<KNXDATATYPE> is a string defining the data type of the incoming or outgoing KNX data.
The system uses the following ASCII strings in capital letters:
ERR, BIT, TWOBITS, FOURBITS, SIXBITS, CHARACTER, UINT8, SINT8, UNIT16, SINT16, FLOAT16, TIME, DATE, UINT32, SINT32, FLOAT32, STRING, GENERIC and DATETIME.
See the explanation of the KNX datatypes for more details about this strings.

<KNXDIRECTION> is an ASCII text string defining the communication direction for this entry on the KNX bus.
The following ASCII string in capital letters are valid:
ERR, READ, WRITE, READ-WRITE, READWRITE, R, W, RW
See the explanation for KNX directions for more details about this string.

<FACTOR> is a float value defining the multiplication factor for incoming KNX telegrams and the division factor for outgoing KNX telegrams.
Use the float format 1234.567. Don't use a comma as a comma sign!

GET CONFIG	**ASCII READALL COMMAND**	#GET CONFIG<CR> #GC<CR> Result: #OK<CR>	ASCII	
	TX	#GET CONFIG<CR>		
	RX	#255,KNX:I:1=5,UINT16,1,0,12.3.38,FLOAT16,RW,1.500000<CR> #255,OK<CR>		

This command returns the complete configuration from the internal FLASH memory.

WRITE VALUE	**ASCII WRITE COMMAND**	#WRITE VALUE:<LINE>=<FLOATVALUE><CR> #WV:<LINE>=<FLOATVALUE><CR> Result: #OK<CR>	ASCII	NO
	LINE	2		
	FLOATVALUE	23,5		
	TX	#WRITE VALUE:2=23.5<CR>		
	RX	N/A		

Writes a new value into the defined configuration line <LINE>.
If the configuration defines a write operation on the KNX bus, the corresponding KNX telegram is generated.
<LINE>: The number of the requested configuration line in the range of 1..n, where n is the last line of the current converter table.
<FLOATVALUE>: The new value for the MODBUS registers formatted as a float number

READ VALUE	**ASCII READ COMMAND**	#READ VALUE:<LINE><CR> #RV:<LINE><CR> Result: #RV:<FLOATVALUE><CR>	ASCII	
	LINE	1		
	TX	#READ VALUE:1<CR>		
	RX	#255,RV:0.000000<CR>		
		Current content of MODBUS register of configuration line 1:0.000000		

Returns the current value of the requested configuration line <LINE>
<LINE>: The number of the requested configuration line in the range of 1..n, where n is the last line of the current converter table.
<FLOATVALUE>: The current value in the MODBUS registers formatted as a float number.

SET KNX ADDRESS	**ASCII WRITE COMMAND**	#SET KNX ADDRESS:<KNXADDRESS><CR> #SKNXADR:<KNXADDRESS><CR> Result: #OK<CR>	ASCII	YES
	KNXADDRESS	15.15.250		
	TX	#SET KNX ADDRESS:15.15.250<CR>		
	RX	N/A		
Redefines the KNX address of the module for KNX bus communication. <KNXADDRES>: The new KNX address for the communication on the KNX bus in the format <Hi>.<Med>.<Low> e.g. 15.15.255 Address range from 0.0.0 to 15.15.255				
GET KNX ADDRESS	**ASCII READ COMMAND**	#GET KNX ADDRESS<CR> #GKNXADR<CR> Result: #GKNXADR:<KNXAdrDec>,<KNXAdrHex>,<KNXAdrKNX><CR>	ASCII	
	TX	#GET KNX ADDRESS<CR>		
	RX	#255,GKNXADR:65530,0xFFFA,15.15.250<CR>		
		Current configured KNX address of module:15.15.250		
Returns the current KNX address of the module. <KNXAdrDec>,<KNXAdrHex>: The current configured KNX address for the communication on the KNX bus. <KNXAdrKNX>: The current configured KNX address for the communication on the KNX bus in the format <Hi>.<Med>.<Low> e.g. 15.15.255				