This is the group of hardware of the current product						
HW_TYPE	3x65202 4x65202 I:65201	1,0x0001 B:00 01			UINT16 R/O	
This is the type of hardware of the current product						
SW_VERSION	3x65203 4x65203 I:65202	272,0x0110 B:01 10			UINT16 R/O	
		SW VERSION:0.1.0				
This is the current software version of the firmware						
SW_AUTHOR	3x65204 4x65204 I:65203	21321,0x5349 B:53 49			UINT16 R/O	
This is the current software author of the firmware						
MANUFACTURER	3x65205 4x65205 I:65204	1380275017,0x52455349 B:52 45 53 49			UINT32 R/O	
This is the current software author of the firmware						
NUMBER OF DIGITAL INPUTS	3x65207 4x65207 I:65206	0,0x0000 B:00 00			UINT16 R/O	
		Number of DIS:0				
This is the current software version of the firmware						
NUMBER OF DIGITAL OUTPUTS	3x65208 4x65208 I:65207	0,0x0000 B:00 00			UINT16 R/O	
		Number of DOS:0				
This is the current software version of the firmware						
NUMBER OF ANALOG INPUTS	3x65209 4x65209 I:65208	0,0x0000 B:00 00			UINT16 R/O	
		Number of AIS:0				
This is the current software version of the firmware						
NUMBER OF ANALOG OUTPUTS	3x65210 4x65210 I:65209	0,0x0000 B:00 00			UINT16 R/O	
		Number of AOS:0				
This is the current software version of the firmware						
NUMBER OF UNIVERSAL IN/OUTPUTS	3x65211 4x65211 I:65210	0,0x0000 B:00 00			UINT16 R/O	
		Number of AIOX:0				
This is the current software version of the firmware						

NUMBER OF SPECIAL INPUTS	3x65212 4x65212 I:65211	0,0x0000 B:00 00			UINT16 R/O	
		Number of special inputs:0				
This is the current software version of the firmware						
NUMBER OF SPECIAL OUTPUTS	3x65213 4x65213 I:65212	0,0x0000 B:00 00			UINT16 R/O	
		Number of special outputs:0				
This is the current software version of the firmware						
FEATURE1	3x65214 4x65214 I:65213	2,0x0002 B:00 02			UINT16 R/O	
		Feature:RS485				
This is the feature list of the controller: 0:NONE, 1:RS232, 2:RS485, 3:KNX, 4:DALI, 5:MBUS, 6:LORA, 7:LTE, 8:2xETHERNET						
FEATURE2	3x65215 4x65215 I:65214	0,0x0000 B:00 00			UINT16 R/O	
		Feature:NONE				
FEATURE3	3x65216 4x65216 I:65215	0,0x0000 B:00 00			UINT16 R/O	
		Feature:NONE				
FEATURE4	3x65217 4x65217 I:65216	0,0x0000 B:00 00			UINT16 R/O	
		Feature:NONE				
FEATURE5	3x65218 4x65218 I:65217	0,0x0000 B:00 00			UINT16 R/O	
		Feature:NONE				
FEATURE6	3x65219 4x65219 I:65218	0,0x0000 B:00 00			UINT16 R/O	
		Feature:NONE				
FEATURE7	3x65220 4x65220 I:65219	0,0x0000 B:00 00			UINT16 R/O	
		Feature:NONE				
FEATURE8	3x65221 4x65221 I:65220	0,0x0000 B:00 00			UINT16 R/O	
		Feature:NONE				
MODBUS WATCHDOG						
MODBUS WATCHDOG TIME	3x65222 4x65222 I:65221	0,0x0000 B:00 00		50	UINT16 R/W	NO

		Actual watchdog time in 1/100s:0 -> 0,0s				
Writing a value onto this register defines a new time for the internal communication watchdog timer. The value is a timespan in 1/100s. =0: The communication watchdog is disabled =1..65535: Communication watchdog will be trigegred after x 1/100s pause on communication line In case of an communication watchdog, the module sets all outputs to the states defined in the configuration output registers Reading this register will return the current stored time from the internal FRAM						
RASPBERRY PI WATCHDOG TIMER	3x65223 4x65223 I:65222	0,0x0000 B:00 00		50	UINT16 R/W	NO
		remaining watchdog time in 10ms:0 -> 0,000s				
Enables or disables the WATCHDOG Timer for the Raspberry Pi module. 1..65535: Time for Watchdog in x10 Milliseconds (Maximum 655,35 seconds) =0: No Watchdog is generated						
RTC REAL TIME CLOCK						
RTC YEAR	3x65231 4x65231 I:65230	24,0x0018 B:00 18		24	UINT16 R/W	NO
		Actual RTC year:24				
Returns the current year of the internal real time clock in the range of 24 to 99. Writing to this register prepares the setting of a new time.						
RTC MONTH	3x65232 4x65232 I:65231	2,0x0002 B:00 02		1	UINT16 R/W	NO
		Actual RTC month:2				
Returns the current month of the internal real time clock in the range of 1 to 12 Writing to this register prepares the setting of a new time.						
RTC DAY	3x65233 4x65233 I:65232	29,0x001D B:00 1D		1	UINT16 R/W	NO
		Actual RTC day:29				
Returns the current day of the internal real time clock in the range of 1 to 31 Writing to this register prepares the setting of a new time.						
RTC HOUR	3x65234 4x65234 I:65233	18,0x0012 B:00 12		12	UINT16 R/W	NO
		Actual RTC month:18				
Returns the current hour of the internal real time clock in the range of 0 to 23 Writing to this register prepares the setting of a new time.						
RTC MINUTE	3x65235 4x65235 I:65234	0,0x0000 B:00 00		45	UINT16 R/W	NO
		Actual RTC hour:0				
Returns the current minute of the internal real time clock in the range of 0 to 59 Writing to this register prepares the setting of a new time.						
RTC SECOND	3x65236 4x65236 I:65235	23,0x0017 B:00 17		30	UINT16 R/W	NO
		Actual RTC second:23				

Returns the current second of the internal real time clock in the range of 0 to 59 Writing to this register prepares the setting of a new time.						
RTC DAY OF WEEK	3x65237 4x65237 I:65236	4,0x0004 B:00 04		5:FRIDAY	UINT16 R/W	NO
		Actual RTC week day:THU		SELECT DAY OF WEEK		
Returns the current day of week in the range 1 to 7 1:MON, 2:TUE, 3:WED, 4:THU, 5:FRI, 6:SAT, 7:SUN Writing to this register writes a new date and time and weekday to the RTC						
DIP SWITCH STATUS						
DIP SWITCH	3x65501 4x65501 I:65500	85,0x0055 B:00 55			UINT16 R/O	
Returns the current setting 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)						
CPU DATA						
SERIAL1	3x65521 4x65521 I:65520	34,0x0022 B:00 22			UINT16 R/O	
Serial number of module as 96 bit unsigned integer number						
SERIAL2	3x65522 4x65522 I:65521	24,0x0018 B:00 18			UINT16 R/O	
SERIAL3	3x65523 4x65523 I:65522	22291,0x5713 B:57 13			UINT16 R/O	
SERIAL4	3x65524 4x65524 I:65523	20547,0x5043 B:50 43			UINT16 R/O	
SERIAL5	3x65525 4x65525 I:65524	13361,0x3431 B:34 31			UINT16 R/O	
SERIAL6	3x65526 4x65526 I:65525	8246,0x2036 B:20 36			UINT16 R/O	
		SERIAL:22001800135743503134362O				
Serial number of module as 96 bit unsigned integer number						
CPU TEMPERATURE	3x65527 4x65527 I:65526	5061,0x13C5 B:13 C5			UINT16 R/O	
		Current internal temperature of CPU:50,61°C				
Current internal temperature of CPU in ° Celsius multiplied by 100.						

CPU VOLTAGE	3x65528 4x65528 I:65527	333,0x014D B:01 4D			UINT16 R/O	
		Current supply voltage of CPU:3,33V				
Current internal supply voltage of CPU in Volt multiplied by 1000.						
CPU BATTERY	3x65529 4x65529 I:65528	317,0x013D B:01 3D			UINT16 R/O	
		Current battery voltage of CPU:3,17V				
Current internal backup battery voltage of CPU in Volt multiplied by 1000.						
RESETs						
RASPBERRY PI RESET	3x65534 4x65534 I:65533	0,0x00 B:00 00		1:PERFORM RASPBERRY PI RESET	BIT R/W	NO
Resets the Raspberry Pi						
RASPBERRY PI RESET	3x65534 4x65534 I:65533	0,0x0000 B:00 00		1:PERFORM RASPBERRY PI RESET	UINT16 R/W	NO
Resets the Raspberry Pi						
FACTORY RESET	1x65535 2x65535 I:65534	0,0x00 B:00		1:PERFORM FACTORY RESET	BIT R/W	NO
Performs a factory reset of all internal saved parameters						
FACTORY RESET	3x65535 4x65535 I:65534	0,0x0000 B:00 00		1:PERFORM FACTORY RESET	UINT16 R/W	NO
Performs a factory reset of all internal saved parameters						
RESET	1x65536 2x65536 I:65535	0,0x00 B:00		1:PERFORM ARM RESET	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). Be aware that you will losse all serial interfaces in USB!!!						
RESET	3x65536 4x65535 I:65535	0,0x0000 B:00 00		1:PERFORM ARM RESET	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). Be aware that you will losse all serial interfaces in USB!!!						

Returns the current setting of the Dip switches. =0: DIP is OFF =1: DIP is ON						
DIP SWITCH DIP2	1x65002 2x65002 I:65001	0,0x00 B:00			BIT R/O	
DIP SWITCH DIP3	1x65003 2x65003 I:65002	1,0x01 B:01			BIT R/O	
DIP SWITCH DIP4	1x65004 2x65004 I:65003	0,0x00 B:00			BIT R/O	
DIP SWITCH DIP5	1x65005 2x65005 I:65004	1,0x01 B:01			BIT R/O	
DIP SWITCH DIP6	1x65006 2x65006 I:65005	0,0x00 B:00			BIT R/O	
DIP SWITCH DIP7	1x65007 2x65007 I:65006	1,0x01 B:01			BIT R/O	
DIP SWITCH DIP8	1x65008 2x65008 I:65007	0,0x00 B:00			BIT R/O	
LED1:GREEN						
LED1:GREEN SET TO OFF	1x65009 2x65009 I:65008	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil sets the LED to OFF						
LED1:GREEN SET TO ON	1x65010 2x65010 I:65009	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil sets the LED to ON						
LED1:GREEN INVERT LED STATE	1x65011 2x65011 I:65010	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil inverts the actual LED state						
LED1:GREEN BLINK	1x65012 2x65012 I:65011	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with last defined time						
LED1:GREEN FLASH	1x65013 2x65013 I:65012	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with last defined times						

LED1:GREEN PULSE	1x65014 2x65014 I:65013	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with last defined time						
LED1:GREEN BLINK 5s	1x65015 2x65015 I:65014	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 5s ON-5s OFF cycle						
LED1:GREEN BLINK 1s	1x65016 2x65016 I:65015	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 1s ON-1s OFF cycle						
LED1:GREEN BLINK 250ms	1x65017 2x65017 I:65016	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 250ms ON-250ms OFF cycle						
LED1:GREEN BLINK 50ms	1x65018 2x65018 I:65017	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 50ms ON-50ms OFF cycle						
LED1:GREEN FLASH 5s-1s	1x65019 2x65019 I:65018	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 5s ON-1s OFF cycle						
LED1:GREEN FLASH 1s-250ms	1x65020 2x65020 I:65019	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 1s ON-250ms OFF cycle						
LED1:GREEN FLASH 500ms-100ms	1x65021 2x65021 I:65020	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 500ms ON-100ms OFF cycle						
LED1:GREEN FLASH 300ms-50ms	1x65022 2x65022 I:65021	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 300ms ON-50ms OFF cycle						
LED1:GREEN PULSE 1s	1x65023 2x65023 I:65022	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 1s ON						
LED1:GREEN PULSE 500ms	1x65024 2x65024 I:65023	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 500ms ON						
LED1:GREEN PULSE 250ms	1x65025 2x65025 I:65024	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 250ms ON						

LED1:GREEN PULSE 100ms	1x65026 2x65026 I:65025	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 100ms ON						
LED1:GREEN PULSE 20ms	1x65027 2x65027 I:65026	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 20ms ON						
LED2:WHITE						
LED2:WHITE SET TO OFF	1x65029 2x65029 I:65028	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil sets the LED to OFF						
LED2:WHITE SET TO ON	1x65030 2x65030 I:65029	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil sets the LED to ON						
LED2:WHITE INVERT LED STATE	1x65031 2x65031 I:65030	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil inverts the actual LED state						
LED2:WHITE BLINK	1x65032 2x65032 I:65031	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with last defined time						
LED2:WHITE FLASH	1x65033 2x65033 I:65032	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with last defined times						
LED2:WHITE PULSE	1x65034 2x65034 I:65033	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with last defined time						
LED2:WHITE BLINK 5s	1x65035 2x65035 I:65034	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 5s ON-5s OFF cycle						
LED2:WHITE BLINK 1s	1x65036 2x65036 I:65035	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 1s ON-1s OFF cycle						
LED2:WHITE BLINK 250ms	1x65037 2x65037 I:65036	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 250ms ON-250ms OFF cycle						
LED2:WHITE BLINK 50ms	1x65038 2x65038 I:65037	????		N/A:DO NOTHING	BIT W/O	NO

Writing 1 to this coil start symmetrical blinking of LED with 50ms ON-50ms OFF cycle						
LED2:WHITE FLASH 5s-1s	1x65039 2x65039 I:65038	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 5s ON-1s OFF cycle						
LED2:WHITE FLASH 1s-250ms	1x65040 2x65040 I:65039	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 1s ON-250ms OFF cycle						
LED2:WHITE FLASH 500ms-100ms	1x65041 2x65041 I:65040	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 500ms ON-100ms OFF cycle						
LED2:WHITE FLASH 300ms-50ms	1x65042 2x65042 I:65041	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 300ms ON-50ms OFF cycle						
LED2:WHITE PULSE 1s	1x65043 2x65043 I:65042	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 1s ON						
LED2:WHITE PULSE 500ms	1x65044 2x65044 I:65043	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 500ms ON						
LED2:WHITE PULSE 250ms	1x65045 2x65045 I:65044	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 250ms ON						
LED2:WHITE PULSE 100ms	1x65046 2x65046 I:65045	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 100ms ON						
LED2:WHITE PULSE 20ms	1x65047 2x65047 I:65046	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 20ms ON						
LED3:RED						
LED3:RED SET TO OFF	1x65049 2x65049 I:65048	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil sets the LED to OFF						
LED3:RED SET TO ON	1x65050 2x65050 I:65049	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil sets the LED to ON						

LED3:RED INVERT LED STATE	1x65051 2x65051 I:65050	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil inverts the actual LED state						
LED3:RED BLINK	1x65052 2x65052 I:65051	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with last defined time						
LED3:RED FLASH	1x65053 2x65053 I:65052	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with last defined times						
LED3:RED PULSE	1x65054 2x65054 I:65053	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with last defined time						
LED3:RED BLINK 5s	1x65055 2x65055 I:65054	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 5s ON-5s OFF cycle						
LED3:RED BLINK 1s	1x65056 2x65056 I:65055	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 1s ON-1s OFF cycle						
LED3:RED BLINK 250ms	1x65057 2x65057 I:65056	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 250ms ON-250ms OFF cycle						
LED3:RED BLINK 50ms	1x65058 2x65058 I:65057	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 50ms ON-50ms OFF cycle						
LED3:RED FLASH 5s-1s	1x65059 2x65059 I:65058	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 5s ON-1s OFF cycle						
LED3:RED FLASH 1s-250ms	1x65060 2x65060 I:65059	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 1s ON-250ms OFF cycle						
LED3:RED FLASH 500ms-100ms	1x65061 2x65061 I:65060	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 500ms ON-100ms OFF cycle						
LED3:RED FLASH 300ms-50ms	1x65062 2x65062 I:65061	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 300ms ON-50ms OFF cycle						

LED3:RED PULSE 1s	1x65063 2x65063 I:65062	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 1s ON						
LED3:RED PULSE 500ms	1x65064 2x65064 I:65063	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 500ms ON						
LED3:RED PULSE 250ms	1x65065 2x65065 I:65064	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 250ms ON						
LED3:RED PULSE 100ms	1x65066 2x65066 I:65065	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 100ms ON						
LED3:RED PULSE 20ms	1x65067 2x65067 I:65066	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 20ms ON						
LED4:YELLOW						
LED4:YELLOW SET TO OFF	1x65069 2x65069 I:65068	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil sets the LED to OFF						
LED4:YELLOW SET TO ON	1x65070 2x65070 I:65069	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil sets the LED to ON						
LED4:YELLOW INVERT LED STATE	1x65071 2x65071 I:65070	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil inverts the actual LED state						
LED4:YELLOW BLINK	1x65072 2x65072 I:65071	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with last defined time						
LED4:YELLOW FLASH	1x65073 2x65073 I:65072	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with last defined times						
LED4:YELLOW PULSE	1x65074 2x65074 I:65073	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with last defined time						
LED4:YELLOW BLINK 5s	1x65075 2x65075 I:65074	????		N/A:DO NOTHING	BIT W/O	NO

Writing 1 to this coil start symmetrical blinking of LED with 5s ON-5s OFF cycle						
LED4:YELLOW BLINK 1s	1x65076 2x65076 I:65075	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 1s ON-1s OFF cycle						
LED4:YELLOW BLINK 250ms	1x65077 2x65077 I:65076	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 250ms ON-250ms OFF cycle						
LED4:YELLOW BLINK 50ms	1x65078 2x65078 I:65077	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start symmetrical blinking of LED with 50ms ON-50ms OFF cycle						
LED4:YELLOW FLASH 5s-1s	1x65079 2x65079 I:65078	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 5s ON-1s OFF cycle						
LED4:YELLOW FLASH 1s-250ms	1x65080 2x65080 I:65079	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 1s ON-250ms OFF cycle						
LED4:YELLOW FLASH 500ms-100ms	1x65081 2x65081 I:65080	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 500ms ON-100ms OFF cycle						
LED4:YELLOW FLASH 300ms-50ms	1x65082 2x65082 I:65081	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start asymmetrical flashing of LED with 300ms ON-50ms OFF cycle						
LED4:YELLOW PULSE 1s	1x65083 2x65083 I:65082	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 1s ON						
LED4:YELLOW PULSE 500ms	1x65084 2x65084 I:65083	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 500ms ON						
LED4:YELLOW PULSE 250ms	1x65085 2x65085 I:65084	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 250ms ON						
LED4:YELLOW PULSE 100ms	1x65086 2x65086 I:65085	????		N/A:DO NOTHING	BIT W/O	NO
Writing 1 to this coil start one time pulse of LED with 100ms ON						
LED4:YELLOW PULSE 20ms	1x65087 2x65087 I:65086	????		N/A:DO NOTHING	BIT W/O	NO

Writing 1 to this coil start one time pulse of LED with 20ms ON						
DIP SWITCH STATUS						
DIP SWITCH	3x65501 4x65501 I:65500	85,0x0055 B:00 55			UINT16 R/O	
Returns the current setting 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)						
LED1:GREEN						
LED1:GREEN STATE	3x65502 4x65502 I:65501	4,0x0004 B:00 04		1:SET TO ON	UINT16 R/W	NO
		State of LED:FLASH				
Returns the current state of the LED Writing to this register will set a new state for the LED 0: Switch LED permanent OFF 1: Switch LED permanent ON 2: Invert last state of LED 3: Start symmetrical blinking of LED with TIME1 ON and TIME1 OFF 4: Start asymmetrical flashing of LED with TIME1 ON and TIME2 OFF 5: Start one time pulse of LED with TIME1 ON and inifinite OFF						
LED1:GREEN TIME1	3x65503 4x65503 I:65502	1000,0x03E8 B:03 E8		1000	UINT16 R/W	YES
		Actual time 1 in ms:1000				
Returns the current time1 for blink,flash and pulse ON time in Milliseconds Writing to this register sets a new time in the range 20-65534ms						
LED1:GREEN TIME2	3x65504 4x65504 I:65503	2000,0x07D0 B:07 D0		2000	UINT16 R/W	YES
		Actual time 2 in ms:2000				
Returns the current time2 for blink and flash OFF time in Milliseconds Writing to this register sets a new time in the range 20-65534ms						
LED2:WHITE						
LED2:WHITE STATE	3x65505 4x65505 I:65504	4,0x0004 B:00 04		1:SET TO ON	UINT16 R/W	NO
		State of LED:FLASH				

Returns the current state of the LED Writing to this register will set a new state for the LED 0: Switch LED permanent OFF 1: Switch LED permanent ON 2: Invert last state of LED 3: Start symmetrical blinking of LED with TIME1 ON and TIME1 OFF 4: Start asymmetrical flashing of LED with TIME1 ON and TIME2 OFF 5: Start one time pulse of LED with TIME1 ON and inifinite OFF						
LED2:WHITE TIME1	3x65506 4x65506 I:65505	1000,0x03E8 B:03 E8		1000	UINT16 R/W	YES
		Actual time 1 in ms:1000				
Returns the current time1 for blink,flash and pulse ON time in Milliseconds Writing to this register sets a new time in the range 20-65534ms						
LED2:WHITE TIME2	3x65507 4x65507 I:65506	2000,0x07D0 B:07 D0		2000	UINT16 R/W	YES
		Actual time 2 in ms:2000				
Returns the current time2 for blink and flash OFF time in Milliseconds Writing to this register sets a new time in the range 20-65534ms						
LED3:RED						
LED3:RED STATE	3x65508 4x65508 I:65507	4,0x0004 B:00 04		1:SET TO ON	UINT16 R/W	NO
		State of LED:FLASH				
Returns the current state of the LED Writing to this register will set a new state for the LED 0: Switch LED permanent OFF 1: Switch LED permanent ON 2: Invert last state of LED 3: Start symmetrical blinking of LED with TIME1 ON and TIME1 OFF 4: Start asymmetrical flashing of LED with TIME1 ON and TIME2 OFF 5: Start one time pulse of LED with TIME1 ON and inifinite OFF						
LED3:RED TIME1	3x65509 4x65509 I:65508	1000,0x03E8 B:03 E8		1000	UINT16 R/W	YES
		Actual time 1 in ms:1000				
Returns the current time1 for blink,flash and pulse ON time in Milliseconds Writing to this register sets a new time in the range 20-65534ms						
LED3:RED TIME2	3x65510 4x65510 I:65509	2000,0x07D0 B:07 D0		2000	UINT16 R/W	YES
		Actual time 2 in ms:2000				
Returns the current time2 for blink and flash OFF time in Milliseconds Writing to this register sets a new time in the range 20-65534ms						
LED4:YELLOW						
LED4:YELLOW STATE	3x65511 4x65511 I:65510	4,0x0004 B:00 04		1:SET TO ON	UINT16 R/W	NO
		State of LED:FLASH				

Returns the current state of the LED Writing to this register will set a new state for the LED 0: Switch LED permanent OFF 1: Switch LED permanent ON 2: Invert last state of LED 3: Start symmetrical blinking of LED with TIME1 ON and TIME1 OFF 4: Start asymmetrical flashing of LED with TIME1 ON and TIME2 OFF 5: Start one time pulse of LED with TIME1 ON and inifinite OFF						
LED4:YELLOW TIME1	3x65512 4x65512 I:65511	1000,0x03E8 B:03 E8		1000	UINT16 R/W	YES
		Actual time 1 in ms:1000				
Returns the current time1 for blink,flash and pulse ON time in Milliseconds Writing to this register sets a new time in the range 20-65534ms						
LED4:YELLOW TIME2	3x65513 4x65513 I:65512	2000,0x07D0 B:07 D0		2000	UINT16 R/W	YES
		Actual time 2 in ms:2000				
Returns the current time2 for blink and flash OFF time in Milliseconds Writing to this register sets a new time in the range 20-65534ms						

HEART BEAT	ASCII READ COMMAND	#HB<CR> Result: #HB<CR>	ASCII	
	TX	#HB<CR>		
	RX	#255,HB<CR>		
Sends an Heartbeat to test the communication				
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-C4-A<CR>		
		Current module type:RESI-C4-A		
Returns the current module type				
GET FEATURES	ASCII READ COMMAND	#FTRS<CR> Result: #FTRS:<Type><CR>	ASCII	
	TX	#FTRS<CR>		
	RX	#255,FTRS:RESI-C4-A,RS485<CR>		
		Current module type:RESI-C4-A		
		Number of digital inputs:RS485		
		Type of digital inputs:N/A		
Returns the current module features				
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:2015-24 BY RESI AND DI HC SIGL,MSC WWW.RESI.CC<CR>		
		Current copyright:2015-24 BY RESI AND DI HC SIGL,MSC WWW.RESI.CC		
Returns the current copyright of the module				
GET SERIAL NUMBER	ASCII READ COMMAND	#SN<CR> Result: #SN:<Serial><CR>	ASCII	
	TX	#SN<CR>		
	RX	#255,SN:220018001357435031343620<CR>		
		Current serial number:220018001357435031343620		
Returns the current serial number of the module				
SET BOX NAME	ASCII WRITE COMMAND	#SETBOXNAME:<BOXNAME><CR> Result: #OK<CR>	ASCII	YES
	BOXNAME	MYBOX		
	TX	#SETBOXNAME:MYBOX<CR>		
	RX	N/A		
Sets a new box name for the controller				
GET BOX NAME	ASCII READ COMMAND	#BOXNAME<CR> Result: #BOXNAME:<BoxName><CR>	ASCII	
	TX	#BOXNAME<CR>		
	RX	#255,BOXNAME:NONAME<CR>		
		Current box name:NONAME		
Returns the current box name of the module. If no box name is defined, the value NONAME is returned				
GET INTERNAL STATUS	ASCII READ COMMAND	#INTSTAT<CR> Result: #INTSTAT:<Status><CR>	ASCII	
	TX	#INTSTAT<CR>		
	RX	#255,INTSTAT:I2C1:0,I2C2:0<CR>		
		Current internal status:I2C1		
Returns the device specific internal status				
GET DIP SWITCH	ASCII READ COMMAND	#GDIP<CR> Result: #GDIP:<DIPSwitchDec>,<DIPSwitchHex><CR>	ASCII	
	TX	#GDIP<CR>		
	RX	#255,GDIP:255,0xFF<CR>		
		Current DIP SWITCH settings:11111111		

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) 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)				
SYSTEM 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. Be aware that you will losse all serial interfaces in USB!!!				
FACTORY RESET	ASCII WRITE COMMAND	#FRST<CR> Result: #OK<CR>	ASCII	NO
	TX	#FRST<CR>		
	RX	N/A		
Executes a factory reset of the module				
WATCHDOG TIMER	ASCII WRITE COMMAND	#WD:<WDTIME><CR> Result: #OK<CR>	ASCII	NO
	WDTIME	10		
	TX	#WD:10<CR>		
	RX	N/A		
Enables or disables the WATCHDOG Timer for the Raspberry Pi module. WDTIME: 1..3600000: Time for Watchdog in Milliseconds (Maximum 60 Minutes) =0: No Watchdog is generated HINT: The Watchdog is internally handled every 10ms, so every value below 10 will reset immediately the Raspberry Pi computer.				
SET IO WATCHDOG TIMER	ASCII WRITE COMMAND	#SIOWATCHDOG:<IOWDTIME><CR> Result: #OK<CR>	ASCII	NO
	IOWDTIME	10		
	TX	#SIOWATCHDOG:10<CR>		
	RX	N/A		
Sets a new time for the internal IO WATCHDOG Timer. <IOWDTIME> is a time in 100ms. =0: No IO Watchdog is used HINT: The Watchdog is internally handled every 100ms, if the Timer reaches 0, all internal IOS will be set to a preconfigured state. Every ASCII command or MODBUS request will reset this timer.				
GET IO WATCHDOG TIMER	ASCII READ COMMAND	#GIOWATCHDOG<CR> Result: #GIOWATCHDOG:;:<IOWDTIME><CR>	ASCII	
	TX	#GIOWATCHDOG<CR>		
	RX	#255,GIOWATCHDOG:0,0x0<CR>		

Returns the current time for the internal IO WATCHDOG Timer. <IOWDTIME> is a time in 100ms. =0: No IO Watchdog is used HINT: The Watchdog is internally handled every 100ms, if the Timer reaches 0, all internal IOS will be set to a preconfigured state. Every ASCII command or MODBUS request will reset this timer.				
CPU PARAMETERS				
GET CPU VOLTAGE	ASCII READ COMMAND	#GCPUTEMP<CR> Result: #GCPUTEMP:<CPUTemp><CR>	ASCII	
	TX	#GCPUTEMP<CR>		
	RX	#255,GCPUTEMP:54.3877<CR>		
		Current internal temperature of CPU:54.3877°C		
Current internal temperature of CPU in ° Celsius.				
GET CPU VOLTAGE	ASCII READ COMMAND	#GCPUVOLT<CR> Result: #GCPUVOLT:<CPUVoltage><CR>	ASCII	
	TX	#GCPUVOLT<CR>		
	RX	#255,GCPUVOLT:3.3364<CR>		
		Current supply voltage of CPU:3.3364V		
Current internal supply voltage of CPU in Volt.				
GET CPU BATTERY	ASCII READ COMMAND	#GCPUBATT<CR> Result: #GCPUBATT:<CPUBatteryVoltage><CR>	ASCII	
	TX	#GCPUBATT<CR>		
	RX	#255,GCPUBATT:3.1786<CR>		
		Current backup battery voltage of CPU:3.1786V		
Current internal backup battery voltage of CPU in Volt.				

GET REAL TIME CLOCK	ASCII READ COMMAND	#GRTC<CR> Result: #GRTC:YMD,<YEAR>,<MONTH>,<DAY>,HMS,<HOUR>,<MINUTE>,<SECOND>,<WEEKDAY> ,DOK,<DATEOK>,TOK,<TIMEOK><CR>	ASCII	
	TX	#GRTC<CR>		
	RX	#255,GRTC:YMD,24,1,1,HMS,0,3,3,MON,DOK,1,TOK,1<CR>		
		Current date DD.MM.YYYY:1.1.2024		
		Current time HH.MM.SS (24h):00:03:03		
		Current Weekday:MON		
		Battery buffered date is ok:YES		
		Battery buffered time is ok:YES		
Shows current RTC time of battery backup RTC on module				
ASCII COMMANDS:REAL TIME CLOCK				
SET REAL TIME CLOCK	ASCII WRITE COMMAND	#SRTC:YMD,<YEAR>,<MONTH>,<DAY>,HMS,<HOUR>,<MINUTE>,<SECOND>, <WEEKDAY><CR> Result: #OK<CR>	ASCII	YES
	YEAR	2020		
	MONTH	4		
	DAY	23		
	HOUR	8		
	MINUTE	20		
	SECOND	23		
	WEEKDAY	THU		
	TX	#SRTC:YMD,20,4,23,HMS,8,20,23,THU<CR>		
	RX	N/A		
Executes a software reset (Reboot) of the module.				

GET FRAMSIZE	ASCII READ COMMAND	#GFRAMSIZE<CR> Result: #GFRAMSIZE:<FRAMType>,<FRAMSize>,<UsedSizeDEC>,<UsedSizeHEX><CR>	ASCII	
	TX	#GFRAMSIZE<CR>		
	RX	#255,GFRAMSIZE:FM25L16B_G,2kB,2,0x2<CR>		
Reads the actual type and size of the used FRAM. The <UsedSize> describes the internal used space in bytes of the FRAM				
ASCII COMMANDS:FRAM				
GET FRAM16	ASCII READ COMMAND	#GFRAM16:<INDEX><CR> Result: #GFRAM16:<INDEXDEC>,<VALUEDEC>,<INDEXHEX>,<VALUEHEX><CR> or #GFRAM16:<INDEXDEC>,ERR,<INDEXHEX>,ERR<CR>	ASCII	
	INDEX	350		
	TX	#GFRAM16:350<CR>		
	RX	#255,GFRAM16:350,0,0x15E,0x0<CR>		
		FRAM Index in bytes:350		
		FRAM Value in decimal:0		
Reads the actual UINT16 value (2 bytes) of FRAM memory <INDEX>. <INDEX> is a BYTE index in the FRAM strogae starting with 0.				
GET FRAM32	ASCII READ COMMAND	#GFRAM32:<INDEX><CR> Result: #GFRAM32:<INDEXDEC>,<VALUEDEC>,<INDEXHEX>,<VALUEHEX><CR> or #GFRAM32:<INDEXDEC>,ERR,<INDEXHEX>,ERR<CR>	ASCII	
	INDEX	350		
	TX	#GFRAM32:350<CR>		
	RX	#255,GFRAM32:350,0,0x15E,0x0<CR>		
		FRAM Index in bytes:350		
		FRAM Value in decimal:0		
Reads the actual UINT32 value 4 bytes) of FRAM memory <INDEX>. <INDEX> is a BYTE index in the FRAM strogae starting with 0.				
GET FRAMDBL	ASCII READ COMMAND	#GFRAMDBL:<INDEX><CR> Result: #GFRAMDBL:<INDEXDEC>,<VALUEDBL>,<INDEXHEX>,<VALUEDBL><CR> or #GFRAMDBL:<INDEXDEC>,ERR,<INDEXHEX>,ERR<CR>	ASCII	
	INDEX	400		
	TX	#GFRAMDBL:400<CR>		
	RX	#255,GFRAMDBL:400,0,0x190,0<CR>		
		FRAM Index in bytes:400		
		FRAM Value in decimal:0		
Reads the actual DOUBLE value 8 bytes) of FRAM memory <INDEX>. <INDEX> is a BYTE index in the FRAM strogae starting with 0.				
ASCII COMMANDS:FRAM				
SET FRAM16	ASCII WRITE COMMAND	#SFRAM16:<INDEX>,<VALUE><CR> Result: #SFRAM16:OK<CR> or #SFRAM16:ERR<CR>	ASCII	YES
	INDEX	350		
	VALUE	1234		
	TX	#SFRAM16:350,1234<CR>		

	RX	N/A		
Writes a new UINT16 value (2 byte) into FRAM memory <INDEX>. <INDEX> is a BYTE index in the FRAM strogae starting with 0.				
SET FRAM32	ASCII WRITE COMMAND	#SFRAM32:<INDEX>,<VALUE><CR> Result: #SFRAM32:OK<CR> or #SFRAM32:ERR<CR>	ASCII	YES
	INDEX	350		
	VALUE	123456		
	TX	#SFRAM32:350,123456<CR>		
	RX	N/A		
Writes a new UINT32 value (4 byte) into FRAM memory <INDEX>. <INDEX> is a BYTE index in the FRAM strogae starting with 0.				
SET FRAMDBL	ASCII WRITE COMMAND	#SFRAMDBL:<INDEX>,<DOUBLEVALUE><CR> Result: #SFRAMDBL:OK<CR> or #SFRAMDBL:ERR<CR>	ASCII	YES
	INDEX	400		
	DOUBLEVALUE	3,1415926		
	TX	#SFRAMDBL:400,3.1415926<CR>		
	RX	N/A		
Writes a new DOUBLE value (8 byte) into FRAM memory <INDEX>. <INDEX> is a BYTE index in the FRAM strogae starting with 0.				

GET LED1	ASCII READ COMMAND	#GLED1<CR> Result: #GLED1:<LEDMode>,<LEDStateDec>,<LEDStateHex><CR>	ASCII	
	TX	#GLED1<CR>		
	RX	#255,GLED1:OFF,0,0x0<CR>		
		Current LED state:OFF LED ist currently 0		
Returns the current state of the LED1:GREEN on the cover of module				
LED COMMANDS:LED1:GREEN				
SET LED1 OFF	ASCII WRITE COMMAND	#SL1OFF<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL1OFF<CR>		
	RX	N/A		
Sets the current state of the LED1:GREEN on the cover of module to OFF				
SET LED1 ON	ASCII WRITE COMMAND	#SL1ON<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL1ON<CR>		
	RX	N/A		
Sets the current state of the LED1:GREEN on the cover of module to ON				
SET LED1 INVERT	ASCII WRITE COMMAND	#SL1INV<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL1INV<CR>		
	RX	N/A		
Inverts the current state of the LED1:GREEN on the cover of module from ON to OFF or from OFF to ON				
SET LED1 PULSE	ASCII WRITE COMMAND	#SL1PULSE:<PULSETIME><CR> Result: #OK<CR>	ASCII	YES
	PULSETIME	1000		
	TX	#SL1PULSE:1000<CR>		
	RX	N/A		
Sets the current state of the LED1:GREEN on the cover of module to PULSE and defines the one time pulse duration in Milliseconds between 1 and 60000				
SET LED1 BLINK	ASCII WRITE COMMAND	#SL1BLINK:<BLINKTIME><CR> Result: #OK<CR>	ASCII	YES
	BLINKTIME	1000		
	TX	#SL1BLINK:1000<CR>		
	RX	N/A		
Sets the current state of the LED1:GREEN on the cover of module to PULSE and defines the one time pulse duration in Milliseconds between 1 and 60000				
SET LED1 FLASH	ASCII WRITE COMMAND	#SL1FLASH:<ONTIME>,<OFFTIME><CR> Result: #OK<CR>	ASCII	YES
	ONTIME	200		
	OFFTIME	3000		
	TX	#SL1FLASH:200,3000<CR>		

	RX	N/A		
Sets the current state of the LED1:GREEN on the cover of module to FLASH and defines the on and off intervals in Milliseconds between 20 and 600000				

GET LED2	ASCII READ COMMAND	#GLED2<CR> Result: #GLED2:<LEDMode>,<LEDStateDec>,<LEDStateHex><CR>	ASCII	
	TX	#GLED2<CR>		
	RX	#255,GLED2:OFF,0,0x0<CR>		
		Current LED state:OFF LED ist currently 0		
Returns the current state of the LED2:WHITE on the cover of module				
LED COMMANDS:LED2:WHITE				
SET LED2 OFF	ASCII WRITE COMMAND	#SL2OFF<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL2OFF<CR>		
	RX	N/A		
Sets the current state of the LED2:WHITE on the cover of module to OFF				
SET LED2 ON	ASCII WRITE COMMAND	#SL2ON<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL2ON<CR>		
	RX	N/A		
Sets the current state of the LED2:WHITE on the cover of module to ON				
SET LED2 INVERT	ASCII WRITE COMMAND	#SL2INV<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL2INV<CR>		
	RX	N/A		
Inverts the current state of the LED2:WHITE on the cover of module from ON to OFF or from OFF to ON				
SET LED2 PULSE	ASCII WRITE COMMAND	#SL2PULSE:<PULSETIME><CR> Result: #OK<CR>	ASCII	YES
	PULSETIME	1000		
	TX	#SL2PULSE:1000<CR>		
	RX	N/A		
Sets the current state of the LED2:WHITE on the cover of module to PULSE and defines the one time pulse duration in Milliseconds between 1 and 60000				
SET LED2 BLINK	ASCII WRITE COMMAND	#SL2BLINK:<BLINKTIME><CR> Result: #OK<CR>	ASCII	YES
	BLINKTIME	1000		
	TX	#SL2BLINK:1000<CR>		
	RX	N/A		
Sets the current state of the LED2:WHITE on the cover of module to PULSE and defines the one time pulse duration in Milliseconds between 1 and 60000				
SET LED2 FLASH	ASCII WRITE COMMAND	#SL2FLASH:<ONTIME>,<OFFTIME><CR> Result: #OK<CR>	ASCII	YES
	ONTIME	200		
	OFFTIME	3000		
	TX	#SL2FLASH:200,3000<CR>		

	RX	N/A		
Sets the current state of the LED2:WHITE on the cover of module to FLASH and defines the on and off intervals in Milliseconds between 20 and 600000				

GET LED3	ASCII READ COMMAND	#GLED3<CR> Result: #GLED3:<LEDMode>,<LEDStateDec>,<LEDStateHex><CR>	ASCII	
	TX	#GLED3<CR>		
	RX	#255,GLED3:OFF,0,0x0<CR>		
		Current LED state:OFF LED ist currently 0		
Returns the current state of the LED3:RED on the cover of module				
LED COMMANDS:LED3:RED				
SET LED3 OFF	ASCII WRITE COMMAND	#SL3OFF<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL3OFF<CR>		
	RX	N/A		
Sets the current state of the LED3:RED on the cover of module to OFF				
SET LED3 ON	ASCII WRITE COMMAND	#SL3ON<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL3ON<CR>		
	RX	N/A		
Sets the current state of the LED3:RED on the cover of module to ON				
SET LED3 INVERT	ASCII WRITE COMMAND	#SL3INV<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL3INV<CR>		
	RX	N/A		
Inverts the current state of the LED3:RED on the cover of module from ON to OFF or from OFF to ON				
SET LED3 PULSE	ASCII WRITE COMMAND	#SL3PULSE:<PULSETIME><CR> Result: #OK<CR>	ASCII	YES
	PULSETIME	1000		
	TX	#SL3PULSE:1000<CR>		
	RX	N/A		
Sets the current state of the LED3:RED on the cover of module to PULSE and defines the one time pulse duration in Milliseconds between 1 and 60000				
SET LED3 BLINK	ASCII WRITE COMMAND	#SL3BLINK:<BLINKTIME><CR> Result: #OK<CR>	ASCII	YES
	BLINKTIME	1000		
	TX	#SL3BLINK:1000<CR>		
	RX	N/A		
Sets the current state of the LED3:RED on the cover of module to PULSE and defines the one time pulse duration in Milliseconds between 1 and 60000				
SET LED3 FLASH	ASCII WRITE COMMAND	#SL3FLASH:<ONTIME>,<OFFTIME><CR> Result: #OK<CR>	ASCII	YES
	ONTIME	200		
	OFFTIME	3000		
	TX	#SL3FLASH:200,3000<CR>		

	RX	N/A		
Sets the current state of the LED3:RED on the cover of module to FLASH and defines the on and off intervals in Milliseconds between 20 and 600000				

GET LED4	ASCII READ COMMAND	#GLED4<CR> Result: #GLED4:<LEDMode>,<LEDStateDec>,<LEDStateHex><CR>	ASCII	
	TX	#GLED4<CR>		
	RX	#255,GLED4:OFF,0,0x0<CR>		
		Current LED state:OFF LED ist currently 0		
Returns the current state of the LED4:YELLOW on the cover of module				
LED COMMANDS:LED4:YELLOW				
SET LED4 OFF	ASCII WRITE COMMAND	#SL4OFF<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL4OFF<CR>		
	RX	N/A		
Sets the current state of the LED4:YELLOW on the cover of module to OFF				
SET LED4 ON	ASCII WRITE COMMAND	#SL4ON<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL4ON<CR>		
	RX	N/A		
Sets the current state of the LED4:YELLOW on the cover of module to ON				
SET LED4 INVERT	ASCII WRITE COMMAND	#SL4INV<CR> Result: #OK<CR>	ASCII	YES
	TX	#SL4INV<CR>		
	RX	N/A		
Inverts the current state of the LED4:YELLOW on the cover of module from ON to OFF or from OFF to ON				
SET LED4 PULSE	ASCII WRITE COMMAND	#SL4PULSE:<PULSETIME><CR> Result: #OK<CR>	ASCII	YES
	PULSETIME	1000		
	TX	#SL4PULSE:1000<CR>		
	RX	N/A		
Sets the current state of the LED4:YELLOW on the cover of module to PULSE and defines the one time pulse duration in Milliseconds between 1 and 60000				
SET LED4 BLINK	ASCII WRITE COMMAND	#SL4BLINK:<BLINKTIME><CR> Result: #OK<CR>	ASCII	YES
	BLINKTIME	1000		
	TX	#SL4BLINK:1000<CR>		
	RX	N/A		
Sets the current state of the LED4:YELLOW on the cover of module to PULSE and defines the one time pulse duration in Milliseconds between 1 and 60000				
SET LED4 FLASH	ASCII WRITE COMMAND	#SL4FLASH:<ONTIME>,<OFFTIME><CR> Result: #OK<CR>	ASCII	YES
	ONTIME	200		
	OFFTIME	3000		
	TX	#SL4FLASH:200,3000<CR>		

	RX	N/A		
Sets the current state of the LED4:YELLOW on the cover of module to FLASH and defines the on and off intervals in Milliseconds between 20 and 600000				

DIGITAL INPUTS				
GET DIGITAL INPUTS	**ASCII READ COMMAND**	#GDIS<CR> Result: #GDIS:<DISDec>,<DISHex><CR>	ASCII	
	TX	#GDIS<CR>		
	RX	#255,GDIS:0,0x0<CR>		
		Current status of digital inputs:00.0000		
Returns the current state of all 6 digital inputs as decimal number and as hexadecimal number. DISDec, DISHex The current state of all digital inputs: Bit 0: State of DI1 (=0:OFF, =1:ON) Bit 1: State of DI2 (=0:OFF, =1:ON) Bit 2: State of DI3 (=0:OFF, =1:ON) Bit 3: State of DI4 =0:OFF, =1:ON) Bit 4: State of DI5 (=0:OFF, =1:ON) Bit 5: State of DI6 (=0:OFF, =1:ON) Bit 6..63: Always 0				
GET DIGITAL INPUT DIx	**ASCII READ COMMAND**	#GDI<DINR><CR> Result: #GDI<DINR>:<DIxDec>,<DIxHex><CR>	ASCII	
	DINR	1		
	TX	#GDI1<CR>		
	RX	#255,GDI1:0,0x0<CR>		
		Current status of digital input DI1:0=OFF		
Returns the current state of the digital input Dix as decimal number and as hexadecimal number. X stands for the desired digital input between 1 and 6. DIxDec, DIxHex: The current state of the digital input x: =0: Digital input is OFF =1: Digital input is ON				
GET ALL CHANGES	**ASCII READ COMMAND**	#GAC<CR> Result: #GAC:<ChangesDec>,<ChangesHex><CR>	ASCII	
	TX	#GAC<CR>		
	RX	#255,GAC:0,0x0<CR>		
		Current change counter:0		
Returns the counter for changes on all digital inputs. As soon as the module detects a short keypress or long key press or long key release event, this counter is incremented by 1. If this values has changed sience the last polling request, the host knows, that at least one digital input has changed its state.				
CHANGE ALL DIS	**ASCII READ COMMAND**	#CADIS<CR> Result: #CADIS:<ChangeDI1Dec>,...,<ChangeDI6Dec>,<ChangeDI1Hex>,...,<ChangeDI6Hex><CR>	ASCII	
	TX	#CADIS<CR>		
	RX	#255,CADIS:0,0,0,0,0,0,0x0,0x0,0x0,0x0,0x0,0x0<CR>		
		Current counter for changes on DI1:0		
		Current counter for changes on DI2:0		
		Current counter for changes on DI3:0		
		Current counter for changes on DI4:0		
		Current counter for changes on DI5:0		
		Current counter for changes on DI6:0		

Returns for each digital input the counter for changes. As soon as the module detects a signal change on a digital input, the change counter for the affected digital input is incremented by 1. A signal change can be: Detection of a short keypress Detection of the start of a long keypress Detection of a release of a long keypress				
CHANGE DIx	**ASCII READ COMMAND**	#CDI<DINR><CR> Result: #CDI<DINR>:<ChangesDec>,<ChangesHex><CR>	ASCII	
	DINR	1		
	TX	#CDI1<CR>		
	RX	#255,CDI1:0,0x0<CR>		
		Current counter for changes on digital input DI1:0		
Returns for digital input <DINR> the counter for signal changes. As soon as the module detects a signal change on a digital input, the change counter for the affected digital input is incremented by 1. A signal change can be: Detection of a short keypress Detection of the start of a long keypress Detection of a release of a long keypress				
SHORT KEY ALL DIS	**ASCII READ COMMAND**	#SKADIS<CR> Result: #SKADIS:<ShortKeyDI1Dec>,...,<ShortKeyDI6Dec>, <ShortKeyDI1Hex>,...,<ShortKeyDI6Hex><CR>	ASCII	
	TX	#SKADIS<CR>		
	RX	#255,SKADIS:0,0,0,0,0,0,0x0,0x0,0x0,0x0,0x0,0x0<CR>		
		Current counter for short keypress events on DI1:0		
		Current counter for short keypress events on DI2:0		
		Current counter for short keypress events on DI3:0		
		Current counter for short keypress events on DI4:0		
		Current counter for short keypress events on DI5:0		
		Current counter for short keypress events on DI6:0		
Returns for each digital input the counter for short keypress events. As soon as the module detects a short keypress on a digital input, the counter for the affected digital input is incremented by 1.				
SHORT KEY DIx	**ASCII READ COMMAND**	#SKDI<DINR><CR> Result: #SKDI<DINR>:<ShortKeyDec>,<ShortKeyHex><CR>	ASCII	
	DINR	1		
	TX	#SKDI1<CR>		
	RX	#255,SKDI1:0,0x0<CR>		
		Current counter for short keypress events on digital input DI1:0		
Returns for digital input <DINR> the counter for short keypress events. As soon as the module detects a short keypress on a digital input, the counter for the affected digital input is incremented by 1.				
LONG KEY START ALL DIS	**ASCII READ COMMAND**	#LKSADIS<CR> Result: #LKSADIS:<LongKeyStartDI1Dec>,...,<LongKeyStartDI6Dec>, <LongKeyStartDI1Hex>,...,<LongKeyStartDI6Hex><CR>	ASCII	
	TX	#LKSADIS<CR>		
	RX	#255,LKSADIS:0,0,0,0,0,0,0x0,0x0,0x0,0x0,0x0,0x0<CR>		
		Current counter for long keypress start events on DI1:0		
		Current counter for long keypress start events on DI2:0		
		Current counter for long keypress start events on DI3:0		
		Current counter for long keypress start events on DI4:0		

		Current counter for long keypress start events on DI5:0		
		Current counter for long keypress start events on DI6:0		
Returns for each digital input the counter for long keypress start events. As soon as the module detects the start of a long keypress on a digital input, the counter for the affected digital input is incremented by 1.				
LONG KEY START DIx	**ASCII READ COMMAND**	#LKSDI<DINR><CR> Result: #LKSDI<DINR>:<LongKeyStartDec>,<LongKeyStartHex><CR>	ASCII	
	DINR	1		
	TX	#LKSDI1<CR>		
	RX	#255,LKSDI1:0,0x0<CR>		
		Current counter for long keypress start events on digital input DI1:0		
Returns for digital input <DINR> the counter for long keypress start events. As soon as the module detects the start of a long keypress on a digital input, the counter for the affected digital input is incremented by 1.				
LONG KEY END ALL DIS	**ASCII READ COMMAND**	#LKEADIS<CR> Result: #LKEADIS:<LongKeyEndDI1Dec>,...,<LongKeyEndDI6Dec>, <LongKeyEndDI1Hex>,...,<LongKeyEndDI6Hex><CR>	ASCII	
	TX	#LKEADIS<CR>		
	RX	#255,LKEADIS:0,0,0,0,0,0,0x0,0x0,0x0,0x0,0x0,0x0,0x0<CR>		
		Current counter for long keypress end events on DI1:0		
		Current counter for long keypress end events on DI2:0		
		Current counter for long keypress end events on DI3:0		
		Current counter for long keypress end events on DI4:0		
		Current counter for long keypress end events on DI5:0		
		Current counter for long keypress end events on DI6:0		
Returns for each digital input the counter for long keypress end events. As soon as the module detects the end of a long keypress on a digital input, the counter for the affected digital input is incremented by 1.				
LONG KEY END DIx	**ASCII READ COMMAND**	#LKEDI<DINR><CR> Result: #LKEDI<DINR>:<LongKeyEndDec>,<LongKeyEndHex><CR>	ASCII	
	DINR	1		
	TX	#LKEDI1<CR>		
	RX	#255,LKEDI1:0,0x0<CR>		
		Current counter for long keypress end events on digital input DI1:0		
Returns for digital input <DINR> the counter for long keypress end events. As soon as the module detects the end of a long keypress on a digital input, the counter for the affected digital input is incremented by 1.				
RISE ALL DIS	**ASCII READ COMMAND**	#RADIS<CR> Result: #RADIS:<RiseDI1Dec>,...,<RiseDI6Dec>,<RiseDI1Hex>,...,<RiseDI6Hex><CR>	ASCII	
	TX	#RADIS<CR>		
	RX	#255,RADIS:0,0,0,0,0,0,0x0,0x0,0x0,0x0,0x0,0x0,0x0<CR>		
		Current counter for rising edges on DI1:0		
		Current counter for rising edges on DI2:0		
		Current counter for rising edges on DI3:0		
		Current counter for rising edges on DI4:0		
		Current counter for rising edges on DI5:0		
		Current counter for rising edges on DI6:0		
Returns for each digital input the counter for rising edges. As soon as the module detects a rising edge on a digital input, the rising edge counter for the affected digital input is incremented by 1.				
RISE DIx	**ASCII READ COMMAND**	#RDI<DINR><CR> Result: #RDI<DINR>:<RiseDec>,<RiseHex><CR>	ASCII	

	DINR	1		
	TX	#RDI1<CR>		
	RX	#255,RDI1:0,0x0<CR>		
		Current counter for rising edges on digital input DI1:0		
Returns for digital input <DINR> the counter for rising edges. As soon as the module detects a rising edge on a digital input, the rising edge counter for the affected digital input is incremented by 1.				
FALL ALL DIS	**ASCII** **READ** **COMMAND**	#FADIS<CR> Result: #FADIS:<FallDI1Dec>,...,<FallDI6Dec>,<FallDI1Hex>,...,<FallDI6Hex><CR>	ASCII	
	TX	#FADIS<CR>		
	RX	#255,FADIS:0,0,0,0,0,0,0x0,0x0,0x0,0x0,0x0,0x0<CR>		
		Current counter for falling edges on DI1:0		
		Current counter for falling edges on DI2:0		
		Current counter for falling edges on DI3:0		
		Current counter for falling edges on DI4:0		
		Current counter for falling edges on DI5:0		
		Current counter for falling edges on DI6:0		
Returns for each digital input the counter for falling edges. As soon as the module detects a falling edge on a digital input, the falling edge counter for the affected digital input is incremented by 1.				
FALL DIx	**ASCII** **READ** **COMMAND**	#FDI<DINR><CR> Result: #FDI<DINR>:<FallDec>,<FallHex><CR>	ASCII	
	DINR	1		
	TX	#FDI1<CR>		
	RX	#255,FDI1:0,0x0<CR>		
		Current counter for falling edges on digital input DI1:0		
Returns for digital input <DINR> the counter for falling edges. As soon as the module detects a falling edge on a digital input, the falling edge counter for the affected digital input is incremented by 1.				
RESET COUNTERS	**ASCII** **WRITE** **COMMAND**	#RC<CR> Result: #OK<CR>	ASCII	NO
	TX	#RC<CR>		
	RX	N/A		
Resets all internal counters for digital inputs and events on this digital inputs to 0.				

DIGITAL OUTPUTS				
SET DIGITAL OUTPUTS	**ASCII WRITE COMMAND**	#SDOS:<OutAllDOS><CR> Result: #OK<CR>	ASCII	YES
	DO1	1:ON		
	DO2	0:OFF		
	DO3	1:ON		
	DO4	0:OFF		
	DO5	0:OFF		
	DO6	0:OFF		
	TX	#SDOS:5<CR>		
	RX	#255,OK<CR>		
Sets all 6 digital outputs to the new state OutAllDOS The new state for all digital outputs Bit 0: State of DO1 (=0:OFF, =1:ON) Bit 1: State of DO2 (=0:OFF, =1:ON) Bit 2: State of DO3 (=0:OFF, =1:ON) Bit 3: State of DO4 (=0:OFF, =1:ON) Bit 4: State of DO5 (=0:OFF, =1:ON) Bit 5: State of DO6 (=0:OFF, =1:ON) Bits 6-31: Always 0				
SET DIGITAL OUTPUT DOx	**ASCII WRITE COMMAND**	#SDO<DONR>:<Out><CR> Result: #OK<CR>	ASCII	NO
	DONR	2		
	DO1	0:OFF		
	TX	#SDO2:0<CR>		
	RX	N/A		
Sets the new state for digital output DOx. The state is defined with <Out>. Out The new state of the digital output DOx: =0: digital output is OFF =1: digital output is ON				
GET DIGITAL OUTPUTS	**ASCII READ COMMAND**	#GDOS<CR> Result: #GDOS:<DOSDec>,<DOSHex><CR>	ASCII	
	TX	#GDOS<CR>		
	RX	#255,GDOS:5,0x5<CR>		
		Current status of digital outputs:00.0101		
Returns the current state of the 6 digital outputs as decimal number and as hexadecimal number. DOSDec, DOSHex The current state of the digital outputs: Bit 0: State of DO1 (=0:OFF, =1:ON) Bit 1: State of DO2 (=0:OFF, =1:ON) Bit 2: State of DO3 (=0:OFF, =1:ON) Bit 3: State of DO4 (=0:OFF, =1:ON) Bit 4: State of DO5 (=0:OFF, =1:ON) Bit 5: State of DO6 (=0:OFF, =1:ON)				
GET DIGITAL OUTPUT DOx	**ASCII READ COMMAND**	#GDO<DONR><CR> Result: #GDO<DONR>:<DOxDec>,<DOxHex><CR>	ASCII	
	DONR	2		

	TX	#GDO2<CR>		
	RX	#255,GDO2:0,0x0<CR>		
		Current status of digital output DO2:0=OFF		
Returns the current state of the digital output DOx as decimal number and as hexadecimal number. X stands for the number of the digital output from 1 to 6. DOxDec, DOxHex The current state of the digital output DOx: =0: relay output is OFF =1: relay output is ON				
SET DOx	**ASCII WRITE COMMAND**	#SDO<DONR>:<DO><CR> Result: #OK<CR>	ASCII	NO
	DONR	2		
	DO	1:ON		
	TX	#SDO2:1<CR>		
	RX	N/A		
This command sets the digital output DOx to the new state DO. DO:The new state of the digital output DOx: =0: relay output is OFF =1: relay output is ON				
SET DIAGNSTIC DIGITAL OUTPUTS	**ASCII WRITE COMMAND**	#SDDOS:<DiagnosticDOS><CR> Result: #OK<CR>	ASCII	YES
	DO1	1:ENABLE		
	DO2	1:ENABLE		
	DO3	1:ENABLE		
	DO4	1:ENABLE		
	DO5	1:ENABLE		
	DO6	1:ENABLE		
	TX	#SDDOS:63<CR>		
	RX	#255,OK<CR>		
Sets the diagnstic mode for all 6 digital outputs to the new diagnstic mode DiagnosticDOS The new diagnstic state for all digital outputs Bit 0: Diagnostic state of DO1 (=0:DISABLED, =1:ENABLED) Bit 1: Diagnostic state of DO2 (=0:DISABLED, =1:ENABLED) Bit 2: Diagnostic state of DO3 (=0:DISABLED, =1:ENABLED) Bit 3: Diagnostic state of DO4 (=0:DISABLED, =1:ENABLED) Bit 4: Diagnostic state of DO5 (=0:DISABLED, =1:ENABLED) Bit 5: Diagnostic state of DO6 (=0:DISABLED, =1:ENABLED) Bits 6-31: Always 0				
SET DIAGNOSTIC DIGITAL OUTPUT DOx	**ASCII WRITE COMMAND**	#SDDO<DONR>:<DiagnosticDOx><CR> Result: #OK<CR>	ASCII	NO
	DONR	2		
	DiagnosticDOx	0:DISABLE		
	TX	#SDDO2:0<CR>		
	RX	N/A		
Sets the new diagnostic state for digital output DOx. The diagnostic state is defined with <DiagnosticDOx>. The new diagnostic state of the digital output DOx: =0: diagnostic mode for digital output is DISABLED =1: diagnostic mode for digital output is ENABLED				

GET DIAGNOSTIC DIGITAL OUTPUTS	**ASCII READ COMMAND**	#GDDOS<CR> Result: #GDDOS:<DiagnosticDOSDec>,<DiagnosticDOSHex><CR>	ASCII	
	TX	#GDDOS<CR>		
	RX	#255,GDDOS:63,0x3F<CR>		
		Current diagnostic mode of digital outputs:11.1111		
Returns the current diagnostic mode of the 6 digital outputs as decimal number and as hexadecimal number. DiagnosticDOSDec, DiagnosticDOSHex The current diagnostic mode of the digital outputs: Bit 0: Diagnostic mode of DO1 (=0:DISABLED, =1:ENABLED) Bit 1: Diagnostic mode of DO2 (=0:DISABLED, =1:ENABLED) Bit 2: Diagnostic mode of DO3 (=0:DISABLED, =1:ENABLED) Bit 3: Diagnostic mode of DO4 (=0:DISABLED, =1:ENABLED) Bit 4: Diagnostic mode of DO5 (=0:DISABLED, =1:ENABLED) Bit 5: Diagnostic mode of DO6 (=0:DISABLED, =1:ENABLED) Bits 6-31: Always 0				
GET DIAGNOSTIC DIGITAL OUTPUT DOx	**ASCII READ COMMAND**	#GDDO<DONR><CR> Result: #GDDO<DONR>:<DiagnosticDOxDec>,<DiagnosticDOxHex><CR>	ASCII	
	DONR	2		
	TX	#GDDO2<CR>		
	RX	#255,GDDO2:1,0x1<CR>		
		Current diagnostic mode of digital output DO2:1=ENABLED		
Returns the current diagnostic mode of the digital output DOx as decimal number and as hexadecimal number. X stands for the number of the digital output from 1 to 6. DiagnosticDOxDec, DiagnosticDOxHex The current diagnostic mode of the digital output DOx: =0: diagnostic mode for digital output is DISABLED =1: diagnostic mode for digital output is ENABLED				
GET DIAGNOSTIC STATUS DIGITAL OUTPUTS	**ASCII READ COMMAND**	#GDSDOS<CR> Result: #GDSDOS:<DiagStateDOSDec>,<DiagStateDOSHex><CR>	ASCII	
	TX	#GDSDOS<CR>		
	RX	#255,GDSDOS:48,0x30<CR>		
		Current diagnstic status of digital outputs:11.0000		
Returns the current diagnostic state of the 6 digital outputs as decimal number and as hexadecimal number. Be aware that you have to enable the diagnstic mode for the DOs, if you want to use this feature. DiagStateDOSDec, DiagStateDOSHex The current diagnstic state of the digital outputs: Bit 0: Diagnostic state of DO1 (=0:NO FAULT, =1:FAULT) Bit 1: Diagnostic state of DO2 (=0:NO FAULT, =1:FAULT) Bit 2: Diagnostic state of DO3 (=0:NO FAULT, =1:FAULT) Bit 3: Diagnostic state of DO4 (=0:NO FAULT, =1:FAULT) Bit 4: Diagnostic state of DO5 (=0:NO FAULT, =1:FAULT) Bit 5: Diagnostic state of DO6 (=0:NO FAULT, =1:FAULT) Bits 6-31: Always 0				
GET DIAGNOSTIC STATUS DIGITAL OUTPUT DOx	**ASCII READ COMMAND**	#GDSDO<DONR><CR> Result: #GDSDO<DONR>:<DiagStateDOxDec>,<DiagStateDOxHex><CR>	ASCII	
	DONR	2		
	TX	#GDSDO2<CR>		
	RX	#255,GDSDO2:0,0x0<CR>		
		Current diagnostic status of digital output DO2:0=NO FAULT		

Returns the current diagnostic state of the digital output DOx as decimal number and as hexadecimal number. X stands for the number of the digital output from 1 to 6. DiagStateDOxDec, DiagStateDOxHex The current diagnostic state of the digital output DOx: =0: Diagnostic state for digital output x is NO FAULT =1: Diagnostic state for digital output x is FAULT				
GET THERMAL STATUS DIGITAL OUTPUT GROUPS	**ASCII** **READ** **COMMAND**	#GTSDOGS<CR> Result: #GTSDOGS:<ThermalDOGSDec>,<ThermalDOGSHex><CR>	ASCII	
	TX	#GTSDOGS<CR>		
	RX	#255,GTSDOGS:0,0x0<CR>		
		Current thermal status of digital output groups:0000		
Returns the current thermal state of the corresponding output group as decimal number and as hexadecimal number. ThermalDOGSDec,ThermalDOGSHex The current thermal state of the digital output group: Bit 0: Thermal state of digital output group #1 (DO1-DO6) (=0:NO FAULT, =1:FAULT) Bits 1-31: Always 0				
GET THERMAL STATUS DIGITAL OUTPUT GROUPx	**ASCII** **READ** **COMMAND**	#GTSDOG<DOGRP><CR> Result: #GTSDOG<DOGRP>:<ThermalDOGxDec>,<ThermalDOGxHex><CR>	ASCII	
	DOGRP	1		
	TX	#GTSDOG1<CR>		
	RX	#255,GTSDOG1:0,0x0<CR>		
		Current thermal status of digital output group DOG1:0=NO FAULT		
Returns the current thermal state of the digital output group DOGRP as decimal number and as hexadecimal number. X stands for the number of the digital output group from 1 to 1. ThermalDOGxDec, ThermalDOGxHex The current thermal state of the digital output group DOGRP: =0: Thermal state for output group is OK (NO FAULT) =1: Thermal state for output group is FAULT Digital output group #1 is DO1-DO6.				
GET POWER STATUS DIGITAL OUTPUT GROUPS	**ASCII** **READ** **COMMAND**	#GPSDOGS<CR> Result: #GPSDOGS:<PowerDOGSDec>,<PowerDOGSHex><CR>	ASCII	
	TX	#GPSDOGS<CR>		
	RX	#255,GPSDOGS:0,0x0<CR>		
		Current power status of digital output groups:0000		
Returns the current power supply state of the corresponding output group as decimal number and as hexadecimal number. PowerDOGSDec,PowerDOGSHex The current power supply state of the digital output group: Bit 0: Power supply state of digital output group #1 (DO1-DO6) (=0:NO FAULT, =1:FAULT) Bits 1-31: Always 0				
GET POWER STATUS DIGITAL OUTPUT GROUPx	**ASCII** **READ** **COMMAND**	#GPSDOG<DOGRP><CR> Result: #GPSDOG<DOGRP>:<PowerDOGxDec>,<PowerDOGxHex><CR>	ASCII	
	DOGRP	1		
	TX	#GPSDOG1<CR>		
	RX	#255,GPSDOG1:0,0x0<CR>		
		Current power status of digital output group DOG1:0=NO FAULT		

Returns the current power supply state of the digital output group DOGRP as decimal number and as hexadecimal number. X stands for the number of the digital output group from 1 to 1.
PowerDOGxDec, PowerDOGxHex
The current power supply state of the digital output group DOGRP:
=0: Power supply state for output group is OK (NO FAULT)
=1: Power supply state for output group is FAULT

Digital output group #1 is DO1-DO6.

GET SPI STATUS DIGITAL OUTPUT GROUPS	**ASCII** **READ** **COMMAND**	#GSSDOGS<CR> Result: #GSSDOGS:<SPIDOGSDec>,<SPIDOGSHex><CR>	ASCII	
	TX	#GSSDOGS<CR>		
	RX	#255,GSSDOGS:0,0x0<CR>		
		Current SPI status of digital output groups:0000		

Returns the current SPI communication state of the corresponding output group as decimal number and as hexadecimal number.
SPIDOGSDec,SPIDOGSHex
The current SPI communication state of the digital output group:
Bit 0: SPI communication state for digital output group #1 (DO1-DO6) (=0:NO FAULT, =1:FAULT)
Bits 1-31: Always 0

GET SPI STATUS DIGITAL OUTPUT GROUPx	**ASCII** **READ** **COMMAND**	#GSSDOG<DOGRP><CR> Result: #GSSDOG<DOGRP>:<SPIDOGxDec>,<SPIDOGxHex><CR>	ASCII	
	DOGRP	1		
	TX	#GSSDOG1<CR>		
	RX	#255,GSSDOG1:0,0x0<CR>		
		Current SPI status of digital output group DOG1:0=NO FAULT		

Returns the current SPI communication state of the digital output group DOGRP as decimal number and as hexadecimal number. X stands for the number of the digital output group from 1 to 1.
SPIDOGxDec, SPIDOGxHex
The current SPI communication state of the digital output group DOGRP:
=0: SPI communication state for output group is OK (NO FAULT)
=1: SPI communication state for output group is FAULT

Digital output group #1 is DO1-DO6.