SEAK | LIGHTING CONTROL & ENERGY EFFICIENCY MANAGER

Description of the MODBUS (RTU) protocol for SEAK modulators
v3.8.5.3

Version compatibility

MODULATOR TYPE	FIRMWARE VERSION	POWERLINE COMMUNICATION
PANTER	2.02 - 2.04	QM-50-SSI1
LUMiBOX	2.08 - 2.30	QM-50-SSI1 / QM-50-SSI3

If you have higher version of firmware, check www.seakenergetics.com for new version of protocol.

LUMiBOX FIRMWARE	MODBUS VERSION	POWERLINE COMMUNICATION
2.08	3.8.2	QM-50-SSI1 / QM-50-SSI3
2.11	3.8.3	QM-50-SSI1 / QM-50-SSI3
2.13	3.8.4	QM-50-SSI1 / QM-50-SSI3
2.14 - 2.16	3.8.4.1 - 3.8.4.6	QM-50-SSI1 / QM-50-SSI3
2.21	3.8.4.7	QM-50-SSI1 / QM-50-SSI3 (asynchro)
2.22	3.8.5.0	QM-50-SSI1 / QM-50-SSI3 (asynchro)
2.27	3.8.5.2	QM-50-SSI1 / QM-50-SSI3 (asynchro)
2.30	3.8.5.3	QM-50-SSI1 / QM-50-SSI3 (asynchro)

Modbus RTU frame

	MODBUS message					
Start	Address	Function	Data	CRC		End
≥ 3.5 char	8 bits	8 bits	N x 8 bits	16 bits		≥ 3.5 char
				Lo	Hi	

Modbus functions

FUNCTION	ACCESS	DESCRIPTION
0x03	READ	Read holding registers
0x10	WRITE	Write multiple registers

Control instructions

REGISTRY ADDRESS	FUNCTION	DESCRIPTION	PNT	LUMiBOX								PAGE
				2.08	2.11	2.13 - 2.17	2.21	2.22	2.25	2.28	2.30	
0X0000	0X10	Dimming of all luminaires (BROADCAST)	●	●	●	●	●	●	●	●	●	4
0X0001	0X10	Dimming of group	●	●	●	●	●	●	●	●	●	4
0X0002	0X10	Dimming of luminaire	●	●	●	●	●	●	●	●	●	4
0X0003	0X10	Add group to luminaire	●	●	●	●	●	●	●	●	●	5
0X0004	0X10	Delete one group from luminaire	●	●	●	●	●	●	●	●	●	5
0X0005	0X10	Delete all groups from luminaire	●	●	●	●	●	●	●	●	●	5
0X0006	0X10	Set luminaire reference		●	●	●	●	●	●	●	●	6
0X0007	0X10	Store luminaire reference in memory		●	●	●	●	●	●	●	●	6
0X0008	0X10	Assign ID to luminaire		●	●	●	●	●	●	●	●	6
0X0009	0X10	Change luminaire ID		●	●	●	●	●	●	●	●	7
0X000A	0X10	Set electricity meter value	●									7
0X000B	0X10	Set active/connected phases	●									7
0X000C	0X10	Set consecutive phases modulation	●									8
0X000D	0X10	Set UART setup	●	●	●	●	●	●	●	●	●	8
0X000E	0X10	Set modulator address and terminator setup		●	●	●	●	●	●	●	●	8
0x000F	0X10	Send request for getting status from range of luminaires IDs		●	●	●	●	●	●	●	●	9
0X0010	0X10	Send request for status of luminaire		●	●	●	●	●	●	●	●	9
0X0011	0X10	Set modulator response bottom reference		●	●	●	●	●	●	●	●	9
0X0012	0X10	Start current measurement		●	●	●	●	●	●	●	●	10
0x0014	0x10	Set depth of modulation		●	●	●	●	●	●	●	●	10
0x0015	0x10	Send echo data		●	●	●	●	●	●	●	●	10
0x0016	0X10	EcoStreet - install sequence for control over night for all luminaires		●	●	●	●	●	●	●	●	11
0X0017	0X10	EcoStreet - install sequence for control over night for group		●	●	●	●	●	●	●	●	12
0X0018	0X10	EcoStreet - install sequence for control over night for one luminaire		●	●	●	●	●	●	●	●	13
0X0019	0X10	EcoStreet - install sequence for fixed control for all luminaires		●	●	●	●	●	●	●	●	14
0x001A	0X10	EcoStreet - install sequence for fixed control for one group		●	●	●	●	●	●	●	●	15
0x001B	0X10	EcoStreet - install sequence for fixed control for one luminaire		●	●	●	●	●	●	●	●	16
0x001C	0X10	EcoStreet - remove sequence for all luminaires		●	●	●	●	●	●	●	●	17
0x001D	0X10	EcoStreet - remove sequence for group		●	●	●	●	●	●	●	●	17
0x001E	0X10	EcoStreet - remove sequence for one luminaire		●	●	●	●	●	●	●	●	17
0x001F	0X10	Check sequence identification number		●	●	●	●	●	●	●	●	18
0x0020	0X10	Check presence of any sequence in luminaire		●	●	●	●	●	●	●	●	18
0x0021	0X10	Set limit of maximal power input		●	●	●	●	●	●	●	●	18
0x0022	0X10	Set constant light output (CLO)		●	●	●	●	●	●	●	●	19

Control instructions

REGISTRY ADDRESS	FUNCTION	DESCRIPTION	PNT	LUMiBOX								PAGE
				2.08	2.11	2.13 - 2.17	2.21	2.22	2.25	2.28	2.30	
0x0023	0X10	Transfer data for all luminaires		●	●	●	●	●	●	●	●	19
0x0024	0X10	Transfer data for one group		●	●	●	●	●	●	●	●	20
0x0025	0X10	Transfer data for one luminaire		●	●	●	●	●	●	●	●	20
0x0026	0X10	Send request for getting data from one luminaire		●	●	●	●	●	●	●	●	21
0x0027	0X10	Set SSI protocol type		●	●	●	●	●	●	●	●	21
0x0028	0X10	Set luminaire depth of modulation		●	●	●	●	●	●	●	●	21
0x0029	0X10	Set modulator response bottom and top reference		●	●	●	●	●	●	●	●	22
0x002A	0x10	Send request for level and status of luminaire			●	●	●	●	●	●	●	22
0x002B	0x10	Send request for getting level and status from range of luminaires IDs			●	●	●	●	●	●	●	22
0x002C	0x10	Set motion mode settings (all luminares - old version)			●	●	●	●	●	●	●	23
0x002D	0X10	Set luminaire fade time / step			●	●	●	●	●	●	●	23
0x002E	0x10	Set DALI driver mode				●	●	●	●	●	●	24
0x002F	0X10	Set response asynchro mode					●	●	●	●	●	24
0x0030	0x10	Set motion mode settings for all luminaires or group						●	●	●	●	25
0x0031	0X10	Set motion mode settings for one luminaire						●	●	●*	●*	26
0x0032	0x10	Configure scene for all luminares or group						●	●	●*	●*	27
0x0033	0X10	Configure scene for one luminaire						●	●	●	●	28
0x0034	0x10	Remove scene (all scenes) for all luminaires or group						●	●	●	●	29
0x0035	0X10	Remove scene (all scenes) for one luminaire						●	●	●	●	29
0x0036	0x10	Activate scene for all luminaires or group						●	●	●	●	29
0x0037	0X10	Activate scene for one luminaire						●	●	●	●	30
0x0039	0x10	Set SSR initial state						●	●	●	●	30
0x003A	0X10	Set response reference mode							●	●	●	30
0x003C	0x10	Send request for motion and status of luminaire								●	●	31
0x003D	0x10	Send request for getting motion and status from range of luminaires IDs								●	●	31
0x003E	0x10	Set luminaire type for all luminaires or group									●	32
0x003F	0x10	Set luminaire type for one luminaire									●	32
0x0107	0x10	Prepare luminaire parameters in response buffer		●	●	●	●	●	●	●	●	33

* changed control instruction (added new parameters)

Request instructions

REGISTRY ADDRESS	FUNCTION	DESCRIPTION	PNT	LUMiBOX								PAGE
				2.08	2.11	2.13 - 2.17	2.21	2.22	2.25	2.28	2.30	
0X0064	0X03	Request state and type of the modulator	●	●	●	●	●	●	●	●	●	35
0x0065	0X03	Request consumed energy	●									35
0x0066	0X03	Request firmware version	●	●	●	●	●	●	●	●	●	35
0X0067	0X03	Request count of set phases	●									36
0X0068	0X03	Request modulator temperature	●	●	●	●	●	●	●	●	●	36
0X0069	0X03	Request consecutive modulation setup	●									36
0X006B	0X03	Request stored luminaire status		●	●	●	●	●	●	●	●	37
0x006C	0x03	Request stored luminaire status from luminaires		●	●	●	●	●	●	●	●	37
0X006D	0X03	Request measured current		●	●	●	●	●	●	●	●	37
0X006E	0X03	Request modulator response bottom reference		●	●	●	●	●	●	●	●	38
0X006F	0X03	Request uptime counter		●	●	●	●	●	●	●	●	38
0X0070	0X03	Request voltage		●	●	●	●	●	●	●	●	38
0x0075	0x03	Request depth of modulation		●	●	●	●	●	●	●	●	39
0x0076	0x03	Request modulator address and terminator setup		●	●	●	●	●	●	●	●	39
0x0077	0x03	Request stored echo data		●	●	●	●	●	●	●	●	39
0x0078	0x03	Request stored data from one luminaire		●	●	●	●	●	●	●	●	40
0x0079	0x03	Request full range modulator temperature		●	●	●	●	●	●	●	●	40
0x007A	0x03	Request SSI protocol type		●	●	●	●	●	●	●	●	40
0x007B	0x03	Requst modulator bottom and top reference		●	●	●	●	●	●	●	●	41
0x007C	0x03	Request stored level and status from one luminaire			●	●	●	●	●	●	●	41
0x007D	0x03	Request stored luminaire level and status from luminaires			●	●	●	●	●	●	●	42
0x007E	0x03	Request asynchro data from buffer					●	●	●	●	●	42
0x007F	0x03	Request response asynchro mode					●	●	●	●	●	43
0x0080	0x03	Request response reference mode							●	●	●	43
0x0082	0x03	Request stored motion and status from one luminaire								●	●	43
0x0083	0x03	Request stored luminaire motion and status from luminaires								●	●	44

Control instructions

Dimming of all luminaires (BROADCAST)				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0000*			*Dimming of all luminaires (BROADCAST)*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Level*	*0 - 100 %*
		Lo		
Crc	*Value*			

Instruction for dimming all luminaires.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0000*	*Dimming of all luminaires (BROADCAST)*
Registry count	*0x0001*	
Crc	*Value*	

Dimming of group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0001*			*Dimming of group*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Group*	*1 - 199*
		Lo		
	2	*Hi*	*Level*	*0 - 100 %*
		Lo		
Crc	*Value*			

Instruction for dimming luminaires within the group.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0001*	*Dimming of group*
Registry count	*0x0002*	
Crc	*Value*	

Dimming of luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0002*			*Dimming of luminaire*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*ID*	*ID (see Table of ID range)*
		Lo		
	2	*Hi*	*Level*	*0 - 100 %*
		Lo		
Crc	*Value*			

Instruction for dimming a single luminaire with this ID.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0002*	*Dimming of luminaire*
Registry count	*0x0002*	
Crc	*Value*	

Control instructions

Add group to luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0003*			*Add group to luminaire*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*ID*	*ID (see Table of ID range)*
		Lo		
	2	*Hi*	*Group*	*1 - 199*
		Lo		
Crc	*Value*			

Instruction to add a luminaire to a group.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0003*	*Add group to luminaire*
Registry count	*0x0002*	
Crc	*Value*	

Delete one group from luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0004*			*Delete one group from luminaire*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*ID*	*ID (see Table of ID range)*
		Lo		
	2	*Hi*	*Group*	*1 - 199*
		Lo		
Crc	*Value*			

Instruction to remove a luminaire from a group.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0004*	*Delete one group from luminaire*
Registry count	*0x0002*	
Crc	*Value*	

Delete all groups from luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0005*			*Delete all groups from luminaire*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*ID*	*ID (see Table of ID range)*
		Lo		
Crc	*Value*			

Instruction to remove a luminaire from all groups.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0005*	*Delete all groups from luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Control instructions

Set luminaire reference				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0006*			*Set luminaire reference*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Reference*	*10 - 38*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0006*	*Set luminaire reference*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for setting demodulation reference.
Reference wil be set temporarily. Value will be lost after power supply reset of luminaire. For permanent setting use Instruction Store luminaire reference in memory.
WARNING! Setting the reference too low may cause wrong data recognition.

Store luminaire reference in memory				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0007*			*Store luminaire reference in memory*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*0x0000*	-
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0007*	*Store luminaire reference in memory*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for saving the reference.
Reference that was set using Instruction Set luminaire reference will be stored in luminaire memory permanently.
WARNING! Setting the reference too low may cause wrong data recognition.

Assign ID to luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0008*			*Assign ID to luminaire*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*ID*	*ID (see Table of ID range)*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0008*	*Assign ID to luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Instruction to assign an ID to the luminaire.
It is only possible to assign ID to a luminaire with ID=0 (blank).

Control instructions

Change luminaire ID				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0009*			*Change luminaire ID*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Actual ID*	*ID (see Table of ID range)*
		Lo		
	2	*Hi*	*New ID*	*ID (see Table of ID range)*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0009*	*Change luminaire ID*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for changing luminaire ID.

Set electricity meter value				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x000A*			*Set electricity meter value*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Value*	*Measured electric energy * 10*
		Lo		
	2	*Hi*	*Value*	
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x000A*	*Set electricity meter value*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for setting the electricity meter value.

Set active/connected phases				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x000B*			*Set active/connected phases*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Number of connected phases*	*1-3*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x000B*	*Set active/connected phases*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for setting number of active/connected phases.
For PNT device it is possible to set number of phases, which are connected. For example if 3 phases are set and only 2 are used, modulator will report an error.

Control instructions

Set consecutive phases modulation				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x000C*			*Set consecutive phases modulation*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*value*	*0 - simultaneous* *1 - consecutive*
		Lo		
Crc	*Value*			

Instruction for setting the consecutive phases modulation.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x000C*	*Set consecutive phases modulation*
Registry count	*0x0001*	
Crc	*Value*	

Set UART setup				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x000D*			*Set UART setup*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Baud rate*	*See table UART setup*
		Lo	*Parity and stopbits*	*See table UART setup*
Crc	*Value*			

Instruction for setting the UART parameters.
New UART settings will take effect after power supply RESET.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x000D*	*Set UART setup*
Registry count	*0x0001*	
Crc	*Value*	

Set modulator address and terminator setup				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x000E*			*Set modulator address and terminator setup*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Termination resistor*	*LUMiBOX v1.04: 1 - active, 0 - inactive* *LUMiBOX v2.08: 0 - active, 1 - inactive*
		Lo	*Address*	*New address: 1 - 31*
Crc	*Value*			

Instruction for changing modulator address.
New address settings will take effect after power supply RESET.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x000E*	*Set modulator address and terminator setup*
Registry count	*0x0001*	
Crc	*Value*	

Control instructions

Send request for getting status from range of luminaires Ids				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x000F*			*Send request for getting status from luminaires*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Luminaire start ID*	*1 - 255*
		Lo		
	2	*Hi*	*Luminaire end ID*	*1 - 255*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x000F*	*Send request for getting status from luminaires*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for sending a request to get actual luminaire status from a set of (up to 100) luminaires. To retreive the statuses returned use read instruction (0x006C).

Send request for status of luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0010*			*Send request for status of luminaire*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*ID*	*1 - 255*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0010*	*Send request for status of luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for sending a request to get status of a luminaire. To retrieve the luminaire status (once Lumibox is in OK status) use request Instruction (0x006C).

Set modulator response bottom reference				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0011*			*Set modulator response bottom reference*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Bottom reference*	*20 - 255*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0011*	*Set modulator response bottom reference*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for setting modulator response bottom reference.
WARNING! Setting too high or too low reference may cause wrong data recognition.

Control instructions

Start current measurement				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0012*			*Start current measurement*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*0x0000*	-
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0012*	*Start current measurement*
Registry count	*0x0001*	
Crc	*Value*	

Instruction to start current mesaurement in Lumibox. To retrieve the measured current use request Instruction (0x006D).

Set depth of modulation				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0014*			*Set depth of modulation*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Depth of modulation*	*0 - low; 1 - high*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0014*	*Set depth of modulation*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for setting depth of modulation.

Send echo data				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0015*			*Send echo data*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*ID*	*0 - 255*
		Lo		
	2	*Hi*	*Echo byte*	*0 - 255*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0015*	*Send echo data*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for testing response from luminaire. To retrieve the testing echo byte, use request Instruction (0x0077).
Use luminaire ID=0 to test 2-way communication of luminaire with no ID assigned yet.

Control instructions

EcoStreet - install sequence for control over night for all luminaires				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0016*			*Install sequence for control over night for all luminaires*
Registry count	*0x0003 + 2n + 2m*			
Byte count	*0x06 + 4n + 4m*			
Data	*1*	*Hi*	*Sequence ID*	*1 - 255*
		Lo		
	2	*Hi*	*Number of intervals (n)*	*Number (n) of intervals before middle of night: 2 - 5*
		Lo		
	*(1*2) + 1*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + 2*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(n*2) + 1*	*Hi*	*Length of nth interval*	*1 - 240*
		Lo		
	*(n*2) + 2*	*Hi*	*Intensity of nth interval*	*0 - 100*
		Lo		
	*(n*2) + 3*	*Hi*	*Number of intervals (m)*	*Number (n) of intervals after middle of night: 1 - 5*
		Lo		
	*(1*2) + (n*2) + 2*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + (n*2) + 3*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(m*2) + (n*2) + 2*	*Hi*	*Length of mth interval*	*1 - 240*
		Lo		
	*(m*2) + (n*2) + 3*	*Hi*	*Intensity of mth interval*	*0 - 100*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0016*	*Install sequence for control over night for all luminaires*
Registry count	*0x0003 + 2n + 2m*	
Crc	*Value*	

Instruction for installing sequence for control over night for all luminaires.
ID of sequence – unique identification number of sequence (for testing presence of sequence)
Intervals – Sum of intervals lengths before middle of night must be exactly 480 min. Sum of intervals lengths after middle of night must be max 480 min. Length of the last interval must be exactly 1 min. For example, the sequence may consist 4 intervals before middle of night (75, 60, 135, 90, 120) and 2 intervals after middle of night (175, 1).

Control instructions

EcoStreet - install sequence for control over night for one group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0017*			*Install sequence for control over night for one group*
Registry count	*0x0004 + 2n + 2m*			
Byte count	*0x08 + 4n + 4m)*			
Data	*1*	*Hi*	*Group*	*1 - 199*
		Lo		
	2	*Hi*	*Sequence ID*	*1 - 255*
		Lo		
	3	*Hi*	*Number of intervals (n)*	*Number (n) of intervals before middle of night: 2 - 5*
		Lo		
	*(1*2) + 2*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + 3*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(n*2) + 2*	*Hi*	*Length of nth interval*	*1 - 240*
		Lo		
	*(n*2) + 3*	*Hi*	*Intensity of nth interval*	*0 - 100*
		Lo		
	*(n*2) + 4*	*Hi*	*Number of intervals (m)*	*Number (n) of intervals after middle of night: 1 - 5*
		Lo		
	*(1*2) + (n*2) + 3*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + (n*2) + 4*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(m*2) + (n*2) + 3*	*Hi*	*Length of nth interval*	*1 - 240*
		Lo		
	*(m*2) + (n*2) + 4*	*Hi*	*Intensity of nth interval*	*0 - 100*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0017*	*Install sequence for control over night for one group*
Registry count	*0x0004 + 2n + 2m*	
Crc	*Value*	

Instruction for installing sequence for control over night for luminaire with this group.
ID of sequence – unique identification number of sequence (for testing presence of sequence)
Intervals – Sum of intervals lengths before middle of night must be exactly 480 min. Sum of intervals lengths after middle of night must be max 480 min. Length of the last interval must be exactly 1 min. For example, the sequence may consist 4 intervals before middle of night (75, 60, 135, 90, 120) and 2 intervals after middle of night (175, 1).

Control instructions

EcoStreet - install sequence for control over night for one luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0018*			*Install sequence for control over night for one luminaire*
Registry count	*0x0004 + 2n + 2m*			
Byte count	*0x08 + 4n + 4m*			
Data	*1*	*Hi*	*Luminaire ID*	*ID (see Table of ID range)*
		Lo		
	2	*Hi*	*Sequence ID*	*1 - 255*
		Lo		
	3	*Hi*	*Number of intervals (n)*	*Number (n) of intervals before middle of night: 2 - 5*
		Lo		
	*(1*2) + 2*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + 3*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(n*2) + 2*	*Hi*	*Length of nth interval*	*1 - 240*
		Lo		
	*(n*2) + 3*	*Hi*	*Intensity of nth interval*	*0 - 100*
		Lo		
	*(n*2) + 4*	*Hi*	*Number of intervals (m)*	*Number (n) of intervals after middle of night: 1 - 5*
		Lo		
	*(1*2) + (n*2) + 3*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + (n*2) + 4*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(m*2) + (n*2) + 3*	*Hi*	*Length of nth interval*	*1 - 240*
		Lo		
	*(m*2) + (n*2) + 4*	*Hi*	*Intensity of nth interval*	*0 - 100*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0018*	*Install sequence for control over night for one luminaire*
Registry count	*0x0004 + 2n + 2m*	
Crc	*Value*	

Instruction for installing sequence for control over night for luminaire with this ID.
ID of sequence – unique identification number of sequence (for testing presence of sequence)
Intervals – Sum of intervals lengths before middle of night must be exactly 480 min. Sum of intervals lengths after middle of night must be max 480 min. Length of the last interval must be exactly 1 min. For example, the sequence may consist 4 intervals before middle of night (75, 60, 135, 90, 120) and 2 intervals after middle of night (175, 1).

Control instructions

EcoStreet - install sequence for fixed control for all luminaires				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0019*			*Install sequence for fixed control for all luminaires*
Registry count	*0x0002 + 2n*			
Byte count	*0x04 + 4n*			
Data	*1*	*Hi*	*Sequence ID*	*1 - 255*
		Lo		
	2	*Hi*	*Number of intervals (n)*	*1 - 10*
		Lo		
	*(1*2) + 1*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + 2*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(n*2) + 1*	*Hi*	*Length of nth interval*	*1 - 240*
		Lo		
	*(n*2) + 2*	*Hi*	*Intensity of nth interval*	*0 - 100*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0019*	*Install sequence for fixed control for all luminaires*
Registry count	*0x0002 + 2n*	
Crc	*Value*	

Instruction for installing sequence for fixed control for all luminaires.
ID of sequence – unique identification number of sequence (for testing presence of sequence)
Intervals – Sum of intervals lengths must be max 960 min. Length of the last interval must be exactly 1 min. For example, the sequence only with one interval will have only one 1 min length interval. For example, lengths of intervals for sequence with 5 intervals: 60, 120, 75, 240, 1.

Control instructions

EcoStreet - install sequence for fixed control for one group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x001A*			*Install sequence for fixed control for one group*
Registry count	*0x0003 + 2n*			
Byte count	*0x06 + 4n*			
Data	*1*	*Hi*	*Group*	*1 - 199*
		Lo		
	2	*Hi*	*Sequence ID*	*1 - 255*
		Lo		
	3	*Hi*	*Number of intervals (n)*	*1 - 10*
		Lo		
	*(1*2) + 2*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + 3*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(n*2) + 2*	*Hi*	*Length of nth interval*	*1 - 240*
		Lo		
	*(n*2) + 3*	*Hi*	*Intensity of nth interval*	*0 - 100*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x001A*	*Install sequence for fixed control for one group*
Registry count	*0x0003 + 2n*	
Crc	*Value*	

Instruction for installing sequence for fixed control for luminaire in this group.
ID of sequence – unique identification number of sequence (for testing presence of sequence)
Intervals – Sum of intervals lengths must be max 960 min. Length of the last interval must be exactly 1 min. For example, the sequence only with one interval will have only one 1 min length interval. For example, lengths of intervals for sequence with 5 intervals: 60, 120, 75, 240, 1.

Control instructions

EcoStreet - install sequence for fixed control for one luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x001B*			*Install sequence for fixed control for one luminaire*
Registry count	*0x0003 + 2n*			
Byte count	*0x06 + 4n*			
Data	*1*	*Hi*	*Luminaire ID*	*ID (see Table of ID range)*
		Lo		
	2	*Hi*	*Sequence ID*	*1 - 255*
		Lo		
	3	*Hi*	*Number of intervals (n)*	*1 - 10*
		Lo		
	*(1*2) + 2*	*Hi*	*Length of 1st interval*	*1 - 240*
		Lo		
	*(1*2) + 3*	*Hi*	*Intensity of 1st interval*	*0 - 100*
		Lo		
	...			
	*(n*2) + 2*	*Hi*	*Length of nth interval*	*1 - 240*
		Lo		
	*(n*2) + 3*	*Hi*	*Intensity of nth interval*	*0 - 100*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x001B*	*Install sequence for fixed control for one luminaire*
Registry count	*0x0003 + 2n*	
Crc	*Value*	

Instruction for installing sequence for fixed control for luminaire with this ID.
ID of sequence – unique identification number of sequence (for testing presence of sequence)
Intervals – Sum of intervals lengths must be max 960 min. Length of the last interval must be exactly 1 min. For example, sequence only with one interval will have only one 1 min length interval. For example, lengths of intervals for sequence with 5 intervals: 60, 120, 75, 240, 1.

Control instructions

EcoStreet - remove sequence for all luminaires				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x001C*			*EcoStreet - remove sequence for all luminaires*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*0x0000*	-
		Lo		
Crc	*Value*			

Instruction for removing a sequence from all luminaires.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x001C*	*EcoStreet - remove sequence for all luminaires*
Registry count	*0x0001*	
Crc	*Value*	

EcoStreet - remove sequence for group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x001D*			*EcoStreet - remove sequence for group*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Group*	*1 - 199*
		Lo		
Crc	*Value*			

Instruction for removing a sequence from the luminaire in this group.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x001D*	*EcoStreet - remove sequence for group*
Registry count	*0x0001*	
Crc	*Value*	

EcoStreet - remove sequence for one luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x001E*			*EcoStreet - remove sequence for one luminaire*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Luminaire ID*	*ID (see Table of ID range)*
		Lo		
Crc	*Value*			

Instruction for removing a sequence from the luminaire with this ID.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x001E*	*EcoStreet - remove sequence for one luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Control instructions

Check sequence identification number				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x001F*			*Check sequence identification number*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Sequence ID*	*0 - 255*
		Lo		
Crc	*Value*			

Instruction for sending identification number to check presence of the sequence.
If luminaire contains sequence with this sequence ID, it goes to STANDBY.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x001F*	*Check sequence identification number*
Registry count	*0x0001*	
Crc	*Value*	

Check presence of any sequence in luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0020*			*Check presence of any sequence in luminaire*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*0x0000*	*0*
		Lo		
Crc	*Value*			

Instruction for checking presence of any sequence in the luminaire.
If luminaire contains some sequence, it goes to STANDBY.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0020*	*Check presence of any sequence in luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Set limit of maximal power input				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0021*			*Set limit of maximal power input*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Limit of maximal power input*	*50 - 100 %*
		Lo		
Crc	*Value*			

Instruction for setting the maximal power input for all luminaires.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0021*	*Set limit of maximal power input*
Registry count	*0x0001*	
Crc	*Value*	

Control instructions

Set constant light output (CLO)				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0022*			*Set constant light output (CLO)*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Period to achieve 100% power input*	*1 - 20 (years)*
		Lo		
	2	*Hi*	*Initial power input*	*60 - 100 %*
		Lo		
Crc	*Value*			

Instruction for setting the constant light output (CLO) for all luminaires.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0022*	*Set constant light output (CLO)*
Registry count	*0x0002*	
Crc	*Value*	

Transfer data for all luminaires				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0023*			*Transfer data for all luminaires*
Registry count	*0x0001 + n*			
Byte count	*0x02 + 2n*			
Data	*1*	*Hi*	*Number of data (n)*	*1 - 20*
		Lo		
	1 + 1	*Hi*	*1st value*	*0 - 255*
		Lo		
	...			
	1 + n	*Hi*	*Nth value*	*0 - 255*
		Lo		
Crc	*Value*			

Instruction for sending the data for all luminaires.
This function is reserved for future use.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0023*	*Transfer data for all luminaires*
Registry count	*0x0001 + n*	
Crc	*Value*	

Control instructions

Transfer data for one group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0024*			*Transfer data for one group*
Registry count	*0x0002 + n*			
Byte count	*0x04 + 2n*			
Data	*1*	*Hi*	*Group*	*1 - 199*
		Lo		
	2	*Hi*	*Number of data (n)*	*1 - 20*
		Lo		
	2 + 1	*Hi*	*1st value*	*0 - 255*
		Lo		
	...			
	2 + n	*Hi*	*Nth value*	*0 - 255*
		Lo		
Crc	*Value*			

Instruction for sending the data for the luminaire in this group.
This function is reserved for future use.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0024*	*Transfer data for one group*
Registry count	*0x0002 + n*	
Crc	*Value*	

Transfer data for one luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0025*			*Transfer data for one luminaire*
Registry count	*0x0002 + n*			
Byte count	*0x04 + 2n*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	2	*Hi*	*Number of data (n)*	*1 - 20*
		Lo		
	2 + 1	*Hi*	*1st value*	*0 - 255*
		Lo		
	...			
	2 + n	*Hi*	*Nth value*	*0 - 255*
		Lo		
Crc	*Value*			

Instruction for sending the data for the luminaire whit this ID.
This function is reserved for future use.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0025*	*Transfer data for one luminaire*
Registry count	*0x0002 + n*	
Crc	*Value*	

Control instructions

Send request for getting data from one luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0026*			*Send request for getting data from one luminaire*
Registry count	*0x0003*			
Byte count	*0x06*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	2	*Hi*	*Start register*	*1 - 16*
		Lo		
	3	*Hi*	*End register*	*1 - 16*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0026*	*Send request for getting data from one luminaire*
Registry count	*0x0003*	
Crc	*Value*	

Instruction for sending a request for getting data from one luminaire. To retreive requested data use request instruction (0x0078).
It is possible to set different range of registers (1 - 6, 5 - 8, ..) or set only one register (1 - 1, 5 - 5). This function is not implemented in luminares yet.

Set SSI protocol type				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0027*			*Set SSI protocol type*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*SSI protocol type*	*1 - activate SSI 1* *2 - activate SSI3*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0027*	*Set SSI protocol type*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for setting SSI protocol type in Lumibox.

Set luminaire depth of modulation				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0028*			*Set luminaire depth of modulation*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Depth of modulation*	*0 - Low, 1 - Middle, 2 - High*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0028*	*Set luminaire depth of modulation*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for setting luminaire depth of modulation.
WARNING! Changing depth of modulation without changing modulator response bottom and top reference may cause wrong data recognition.

Control instructions

Set modulator response bottom and top reference				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0029*			*Set modulator response bottom reference*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Bottom reference*	*20 - 255*
		Lo		
	2	*Hi*	*Top reference*	*40 - 255*
		Lo		*(must be greater than bottom referecnce at least 20)*
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0029*	*Set modulator response bottom reference*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for setting modulator response bottom and top reference.
WARNING! Setting too high or too low reference may cause wrong data recognition.

Send request for level and status of luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x002A*			*Send request for level and status of luminaire*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*ID*	*1 - 255*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x002A*	*Send request for level and status of luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for sending a request to get actual dimming level and status of a luminaire. To retrieve the actual dimming level and status of luminare, use request instruction (0x007C).

Send request for getting level and status from range of luminaires Ids				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x002B*			*Send request for getting level and status from luminaires*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Luminaire start ID*	*1 - 255*
		Lo		
	2	*Hi*	*Luminaire end ID*	*1 - 255*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x002B*	*Send request for getting status from luminaires*
Registry count	*0x0002*	
Crc	*Value*	

Instriction for sending a request to get actual dimming level and status from a set of (up to 100) luminaires. To retrieve dimming level and status from requested luminaires use request instruction (0x007D).

Control instructions

Set motion mode settings (all luminares - old version)				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x002C*			*Set motion mode settings (all luminares - old version)*
Registry count	*0x0004*			
Byte count	*0x08*			
Data	*1*	*Hi*	*Enable / disable motion mode*	*Enable = 1* *Disable = 0*
		Lo		
	2	*Hi*	*Level*	*0 - 100%*
		Lo		
	3	*Hi*	*Motion level type*	*Direct level = 0* *Additional level = 1*
		Lo		
	4	*Hi*	*Hold on time*	*0 - 120* *time = (0 - 120) * 5s*
		Lo		
Crc	*Value*			

Instruction for setting motion mode parameters.
Direct motion level type - luminaire after capturing motion goes to Level directly.
Additional motion level type - luminaire after capturing motion adds Level to current level.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x002C*	*Set motion mode settings (all luminares - old version)*
Registry count	*0x0004*	
Crc	*Value*	

Set luminaire fade time / step				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x002D*			*Set luminaire fade step*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Fade time / step*	*DALI fade (for dali drivers):* *0 - No fade, 1 - 0.7s, 2 - 1.0s, 3 - 1.4s, 4 - 2.0s, 5 - 2.8s, 6 - 4.0s, 7 - 5.7s, 8 - 8.0s, 9 - 11.3s, 10 - 16.0s, 11 - 22.6s, 12 - 32.0s, 13 - 45.3s, 14 - 64.0s, 15 - 90.5s* *SSI fade (for other drivers):* *0 - 20 (step = (0 - 20) * 10ms)*
		Lo		
Crc	*Value*			

Instruction for setting luminaire fade step.
DALI fade time is standard DALI fade which is set directly to DALI driver
SSI fade step determines time which luminode waits between two dimming steps. Set 0 to set no fade.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x002D*	*Set luminaire fade step*
Registry count	*0x0001*	
Crc	*Value*	

Control instructions

Set DALI driver mode				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x002E*			*Set DALI driver mode*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*ID*	*0 - All luminaires* *1 - 255 - ID*
		Lo		
	2	*Hi*	*DALI driver mode*	*0 - one DALI driver mode* *1 - two DALI drivers mode*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x002E*	*Set DALI driver mode*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for setting DALI driver mode. Set ID 0 to send command to all luminaires.
DALI driver mode means how many drivers are connected to LUMiNODE.
It is possible to connect 1 or 2 DALI drivers. If 2 DALI drivers are connected to LUMiNODE, they must have DALI address set to 0 and 1.

Set response asynchro mode				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x002F*			*Set response asynchro mode*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*ID*	*0 - disabled* *1 - enabled*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x002F*	*Set response asynchro mode*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for setting response asynchro mode.
If enabled, LUMiBOX receives incoming response data (only special commands) from devices (LUMiCHARGER) all the time.
If disabled, LUMiBOX receives response data only defined time after requested command.
WARNING! Asynchro response supports LUMiBOX SLM-160A and SLM-140A with firmware version higher than 2.21 (including).

Control instructions

Set motion mode settings for all luminaires or group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0030*			*Set motion mode settings for all luminaires or group*
Registry count	*0x0005*			
Byte count	*0x0A*			
Data	*1*	*Hi*	*All luminaire / group*	*0 - All luminaires* *1 - 199 - Group*
		Lo		
	2	*Hi*	*Enable / disable motion mode*	*Enable = 1* *Disable = 0*
		Lo		
	3	*Hi*	*Level*	*0 - 100%*
		Lo		
	4	*Hi*	*Motion level type*	*Direct level = 0* *Additional level = 1*
		Lo		
	5	*Hi*	*Hold on time*	*0 - 120* *time = (0 - 120) * 5s*
		Lo		
	6	*Hi*	*Asynchro mode*	*Disabled = 0* *Enabled (no dimming after motion captured) = 1* *Enabled (with dimming after motion captured) = 2*
		Lo		
Crc	*Value*			

Instruction for setting motion mode parameters.
Direct motion level type - luminaire after capturing motion goes to Level directly.
Additional motion level type - luminaire after capturing motion adds Level to current level.
In FW version 2.28 added new parameter: Asynchro mode.

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0030*	*Set motion mode settings for all luminaires or group*
Registry count	*0x0005*	
Crc	*Value*	

Control instructions

Set motion mode settings for one luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0031*			*Set motion mode settings for one luminaire*
Registry count	*0x0005*			
Byte count	*0x0A*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	2	*Hi*	*Enable / disable motion mode*	*Enable = 1* *Disable = 0*
		Lo		
	3	*Hi*	*Level*	*0 - 100%*
		Lo		
	4	*Hi*	*Motion level type*	*Direct level = 0* *Additional level = 1*
		Lo		
	5	*Hi*	*Hold on time*	*0 - 120* *Time = (0 - 120) * 5s*
		Lo		
	6	*Hi*	*Asynchro mode*	*Disabled = 0* *Enabled (no dimming after motion captured) = 1* *Enabled (with dimming after motion captured) = 2*
		Lo		
	7	*Hi*	*Asynchro ID*	*ID of motion sensor (1 - 16)*
		Lo		
Crc	*Value*			

Instruction for setting motion mode parameters.
Direct motion level type - luminaire after capturing motion goes to Level directly.
Additional motion level type - luminaire after capturing motion adds Level to current level.
In FW version 2.28 added new parameters: Asynchro mode and Asynchro ID

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0031*	*Set motion mode settings for one luminaire*
Registry count	*0x0005*	
Crc	*Value*	

Control instructions

Configure scene for all luminares or group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0032*			*Configure scene for all luminares or group*
Registry count	*0x000A*			
Byte count	*0x14*			
Data	*1*	*Hi*	*All luminaires / group*	*0 - All luminaires* *1 - 199 - Group*
		Lo		
	2	*Hi*	*Scene number*	*0 | 1 - 15*
		Lo		
	3	*Hi*	*Mode*	*1 - Level* *2 - Tc* *3 - RGB-XY* *4 - RGBWAF*
		Lo		
	4	*Hi*	*Level*	*0 - 100 %*
		Lo		
	5	*Hi*	*Data 1*	*See Table of scene data*
		Lo		
	6	*Hi*	*Data 2*	*See Table of scene data*
		Lo		
	7	*Hi*	*Data 3*	*See Table of scene data*
		Lo		
	8	*Hi*	*Data 4*	*See Table of scene data*
		Lo		
	9	*Hi*	*Data 5*	*See Table of scene data*
		Lo		
	10	*Hi*	*Data 6*	*See Table of scene data*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0032*	*Configure scene for all luminares or group*
Registry count	*0x000A*	
Crc	*Value*	

Instruction for configuring scenes in LUMiNODE for all luminaires or group of luminaires.
Set scene number to 0 for trying configuration in driver. Scene activation is not needed. This configuration will not be saved permanently.
For each scene can be configured one of 4 modes.
Mode 1 - Level: in this mode is set only dimming level of luminaire (all other data is set to 0).
Mode 2 - Tc: in this mode it is possible to set color temperature of light (color temperature adjusting) Tc is set in Kelvin units.
Mode 3 - XY: in this mode it is possible to set color using XY coordinates of chromaticity diagram (CIE color space chromaticity diagram 1931).
Mode 4 - RGBWAF: in this mode it is possible to set color mixing 1 - 6 channels (Red, Green, Blue, White, Amber, Freecolor).
In all color modes it is possible to change intensity of color by setting Level value.

Control instructions

Configure scene for one luminare				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0033*			*Configure scene for one luminare*
Registry count	*0x000A*			
Byte count	*0x14*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	2	*Hi*	*Scene number*	*0 | 1 - 15*
		Lo		
	3	*Hi*	*Mode*	*1 - Level* *2 - Tc* *3 - XY* *4 - RGBWAF*
		Lo		
	4	*Hi*	*Level*	*0 - 100 %*
		Lo		
	5	*Hi*	*Data 1*	*See Table of scene data*
		Lo		
	6	*Hi*	*Data 2*	*See Table of scene data*
		Lo		
	7	*Hi*	*Data 3*	*See Table of scene data*
		Lo		
	8	*Hi*	*Data 4*	*See Table of scene data*
		Lo		
	9	*Hi*	*Data 5*	*See Table of scene data*
		Lo		
	10	*Hi*	*Data 6*	*See Table of scene data*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0033*	*Configure scene for one luminare*
Registry count	*0x000A*	
Crc	*Value*	

Instruction for configuring scenes in LUMiNODE for all luminaires or group of luminaires.
Set scene number to 0 for trying configuration in driver. Scene activation is not needed. This configuration will not be saved permanently.
For each scene can be configured one of 4 modes.
Mode 1 - Level: in this mode is set only dimming level of luminaire (all other data is set to 0).
Mode 2 - Tc: in this mode it is possible to set color temperature of light (color temperature adjusting) Tc is set in Kelvin units.
Mode 3 - XY: in this mode it is possible to set color using XY coordinates of chromaticity diagram (CIE color space chromaticity diagram 1931).
Mode 4 - RGBWAF: in this mode it is possible to set color mixing 1 - 6 channels (Red, Green, Blue, White, Amber, Freecolor).
In all color modes it is possible to change intensity of color by setting Level value.

Control instructions

Remove scene from all luminaires or group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0034*			*Remove scene from all luminaires or group*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*All luminaires / group*	*0 - All luminaires* *1 - 199 - Group*
		Lo		
	2	*Hi*	*Scene number*	*0 - Remove all scenes* *1 - 15 - Scene number to remove*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0034*	*Remove scene from all luminaires or group*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for removing scene (all scenes) from LUMiNODE for all luminaires or group of luminaires.

Remove scene from one luminare				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0035*			*Remove scene from one luminare*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	2	*Hi*	*Scene number*	*0 - Remove all scenes* *1 - 15 - Scene number to remove*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0035*	*Remove scene from one luminare*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for removing scene (all scenes) from LUMiNODE for one luminaire.

Activate scene for all luminaires or group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0036*			*Activate scene for all luminaires or group*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*All luminaires / group*	*0 - All luminaires* *1 - 199 - Group*
		Lo		
	2	*Hi*	*Scene number*	*1 - 15 - Scene number to activate*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0036*	*Activate scene for all luminaires or group*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for activating scene in LUMiNODE for all luminaires or group of luminaires.

Control instructions

Activate scene for one luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0037*			*Activate scene for one luminaire*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	2	*Hi*	*Scene number*	*1 - 15 - scene number to activate*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0037*	*Activate scene for one luminaire*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for activating scene in LUMiNODE for one luminaire

Set SSR initial state				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0039*			*Set SSR initial state*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	2	*Hi*	*Initial state*	*0 - OFF* *1 - ON*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0039*	*Set SSR initial state*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for setting initial state of LUMiNODE SSR-1000

Set response reference mode				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x003A*			*Set response reference mode*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*SSI protocol type*	*0 - negative* *1 - positive* *2 - auto*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x003A*	*Set response reference mode*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for setting responde reference mode in LUMiBOX.
It is possible to set different mode according to load characteristic.
Auto mode automatically evaluates actual load characteristic and chooses between positive and negative mode.

Control instructions

Send request for motion and status of luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x003C*			*Send request for motion and status of luminaire*
Registry count	*0x0001*			
Byte count	*0x02*			
Data	*1*	*Hi*	*ID*	*1 - 255*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x003C*	*Send request for motion and status of luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for sending a request to get actual status of motion input and status of a luminaire. To retrieve status of motion input and status from requested luminaires use request instruction (0x0082).

Send request for getting motion and status from range of luminaires Ids				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x003D*			*Send request for getting level and status from luminaires*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Luminaire start ID*	*1 - 255*
		Lo		
	2	*Hi*	*Luminaire end ID*	*1 - 255*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x003D*	*Send request for getting status from luminaires*
Registry count	*0x0002*	
Crc	*Value*	

Instriction for sending a request to get actual status of motion input and status of luminaire from a set of (up to 100) luminaires. To retrieve status of motion input and status from requested luminaires use request instruction (0x0083).

Control instructions

Set luminaire type for all luminaires or group				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x003E*			*Set luminaire type for all luminaires or group*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*All luminaires / group*	*0 - All luminaires* *1 - 199 - Group*
		Lo		
	1	*Hi*	*Luminaire type*	*0 - DALI* *1 - 0-10V* *2 - auto DALI detection*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x003E*	*Set luminaire type for all luminaires or group*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for setting type of luminare connected to LUMiNODE (using this instruction is possible with multiple-output LUMiNODE SDM-110DA) for all luminaires or group of them.
Auto DALI detection sets DALI type only for the LUMiNODEs which have connected luminaire to DALI interface (luminaire has to communicate with LUMiNODE). Otherwise 0-10V type is set to LUMiNODE.

Set luminaire type for one luminaire				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x003F*			*Set luminaire type for one luminaire*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	1	*Hi*	*Luminaire type*	*0 - DALI* *1 - 0-10V* *2 - auto DALI detection*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x003F*	*Set luminaire type for one luminaire*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for setting type of luminare connected to LUMiNODE (using this instruction is possible with multiple-output LUMiNODE SDM-110DA) for one luminaire.
Auto DALI detection sets DALI type only for the LUMiNODEs which have connected luminaire to DALI interface (luminaire has to communicate with LUMiNODE). Otherwise 0-10V type is set to LUMiNODE.

Control instructions

Prepare luminaire parameters in response buffer				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x10*			
Registry address	*0x0107*			*Prepare luminaire parameters in response buffer*
Registry count	*0x0002*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Luminaire ID*	*1 - 255*
		Lo		
	2	*Hi*	*Parameters mode*	*0 - luminaire parameters* *1 - groups*
		Lo		
Crc	*Value*			

Response		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x10*	
Registry address	*0x0107*	*Prepare luminaire parameters in response buffer*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for preparing luminare parameters in LUMiNODE response buffer.
For getting requested data data it is necessary to use 0x0026 - Send request for getting data from one luminaire and after that read received data with 0x0078 - Request stored data from one luminaire.
It is possible to request range of registers for getting requested parameters (see Table of luminaire parameters).

Error response for control instruction

Error response		
Name	**Value**	**Description**
Address	***Address***	
Function	***0x10 + 0x80***	*Function + 0x80*
Error code	***Error code***	*See Table of error codes*
Crc	***Value***	

Request instructions

Request state and type of device		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0064*	*Request state and type of device*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting actual state of device and device type.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Device type*	*See Table of device types*
		Lo	*State of device*	*See Table of device states*
Crc	*Value*			

Request consumed energy		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0065*	*Request consumed energy*
Registry count	*0x0003*	
Crc	*Value*	

Instruction for getting consumed energy from modulator electricity meter.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x06*			
Data	*1*	*Hi*	*State of device*	*See Table of modulator states*
		Lo		
Data	*2*	*Hi*	*Energy upper bytes*	*Consumed energy * 10*
		Lo		
Data	*3*	*Hi*	*Energy lower bytes*	
		Lo		
Crc	*Value*			

Request firmware version		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0066*	*Request firmware version*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting firmware version of the modulator.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Version*	*Firmware version (version.subversion)*
		Lo	*Sub-version*	
Crc	*Value*			

Request instructions

Request count of set phases		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0067*	*Request count of set phases*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting count of active phases set in the modulator (it is not count of connected phases).

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Connected phases*	*Count of set phases*
		Lo		
Crc	*Value*			

Request modulator temperature		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0068*	*Request modulator temperature*
Registry count	*0x0003*	
Crc	*Value*	

Instruction for getting the temperature of each phase of the modulator. Instruction is for 3-phase device such as PNT modulator and temperature is non negative value.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x06*			
Data	*1*	*Hi*	*L1 temperature*	*Temperature in Celsius degrees. Temperature is measured for each modulated phase separately.*
		Lo		
Data	*2*	*Hi*	*L2 temperature*	
		Lo		
Data	*3*	*Hi*	*L3 temperature*	
		Lo		
Crc	*Value*			

Request consecutive modulation setup		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0069*	*Request consecutive modulation setup*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting the consecutive modulation settings.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Modulation setup*	*0 - simultaneous modulation* *1 - consecutive modulation*
		Lo		
Crc	*Value*			

Request instructions

Request stored luminaire status		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x006B*	*Request stored luminaire status*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting received status from the luminaire.
Use this Instruction after control instruction 0X0010.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Full info byte*	*See tables Response info bytes*
		Lo	*Short info byte*	
Crc	*Value*			

Request stored luminaire status from luminaires		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x006C*	*Request stored luminaire status from luminaires*
Registry count	*n*	*Luminaires count (1 - 100)*
Crc	*Value*	

Instruction for getting received status from the set of luminaires (up to 100).

Use this Instruction after control instruction 0x000F.
In this instruction you don't need to set range of luminaires, but only its count.
Range of luminaires is set in control instruction 0x000F.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02 * n*			
	1	*Hi*	*Full info byte*	*See tables Response status info bytes*
		Lo	*Short info byte*	
	...			
	n	*Hi*	*Full info byte*	*See tables Response status info bytes*
		Lo	*Short info byte*	
Crc	*Value*			

Request measured current		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x006D*	*Request measured current*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting measured current from the modulator.
Use this Instruction after control instruction 0X0012.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Current (A)*	*Measured current in ampers*
		Lo	*Current (mA)*	*Fractional part of value (0x03 = 300 mA)*
Crc	*Value*			

Request instructions

Request modulator response bottom reference		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x006E*	*Request modulator response bottom reference*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting modulator response bottom reference.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Reference*	*Bottom reference of response channel (20 - 255)*
		Lo		
Crc	*Value*			

Request uptime counter		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x006F*	*Request uptime counter*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for getting actual value of uptime counter.
Uptime counter is 32bit counter, which is equal to zero at device power-on and increment every second by 1.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Counter - Hi*	*Uptime counter (in seconds)*
		Lo		
Data	*2*	*Hi*	*Counter - Lo*	
		Lo		
Crc	*Value*			

Request voltage		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0070*	*Request measured voltage*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting actual mains voltage measured by modulator (tolerance +/- 5%).

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Voltage (V)*	*Measured voltage*
		Lo		
Crc	*Value*			

Request instructions

Request depth of modulation		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0075*	*Request depth of modulation*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting modulator depth of modulation.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Depth of modulation*	*1 - high, 0 - low*
		Lo		
Crc	*Value*			

Request modulator address and terminator setup		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0076*	*Request modulator address and terminator setup*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting modulator address and terminator setup.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Terminator*	*LUMiBOX v1.04: 1 - active, 0 - inactive* *LUMiBOX v2.08: 0 - active, 1 - inactive*
		Lo	*Address*	*Address*
Crc	*Value*			

Request stored echo data		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0077*	*Request stored echo data*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting modulator address and terminator setup.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Echo byte*	*Received echo byte (0 - 255)*
		Lo	*Data info byte*	*See tables Response data info byte*
Crc	*Value*			

Request instructions

Request stored data from one luminaire		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0078*	*Request stored data from one luminaire*
Registry count	*n*	*Data count (1 - 16)*
Crc	*Value*	

Instruction for getting stored data from one luminaire. Use this Instruction after control instruction 0x0026.
This function is reserved for future use.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02 * n*			
Data	*1*	*Hi*	*Data byte*	*Received data byte (0 - 255)*
		Lo	*Data info byte*	*See tables Response data info byte*
	...			
	n	*Hi*	*Full info byte*	*Received data byte (0 - 255)*
		Lo	*Short info byte*	*See tables Response data info byte*
Crc	*Value*			

Request full range modulator temperature		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0079*	*Request full range modulator temperature*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting the temperature of modulator.
Temperature is full range value (it could be also negative value).

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Temperature*	*Full range temperature of modulator (data type: signed short)*
		Lo		
Crc	*Value*			

Request SSI protocol type		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x007A*	*Request SSI protocol type*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting SSI protocol type set in the modulator.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*SSI protocol type*	*1 - activated SSI 1* *2 - activated SSI3*
		Lo		
Crc	*Value*			

Request instructions

Requst modulator response bottom and top reference		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x007B*	*Requst modulator response bottom and top reference*
Registry count	*0x0002*	
Crc	*Value*	

Instruction for getting modulator response bottom and top reference.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Bottom reference*	*Bottom and top reference of response channel (20 - 255)*
		Lo		
Data	*2*	*Hi*	*Top reference*	
		Lo		
Crc	*Value*			

Request stored luminaire level and status from luminaire		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x007C*	*Request stored level and status from luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Instruction for getting received actual dimming level and status from the luminaire.
Use this Instruction after control instruction 0x002A.
WARNING! Received actual level may be inaccurate. Accepted tolerance between set and received level is max 3%.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Actual level byte*	*See tables Response info bytes*
		Lo	*Short info byte*	
Crc	*Value*			

Request instructions

Request stored luminaire level and status from luminaires		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x007D*	*Request stored luminaire level and status from luminaires*
Registry count	*n*	*Luminaires count (1 - 100)*
Crc	*Value*	

Instruction for getting received actual dimming level and status from the set of luminaires (up to 100).
Use this Instruction after control instruction 0x002B.
In this instruction you don't need to set range of luminaires, but only its count.
Range of luminaires is set in control instruction 0x002B.
WARNING! Received actual level may be inaccurate. Accepted tolerance between set and received level is max 3%.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02 * n*			
	1	*Hi*	*Actual level byte*	*See tables Response status info bytes*
		Lo	*Short info byte*	
	...			
	n	*Hi*	*Actual level byte*	*See tables Response status info bytes*
		Lo	*Short info byte*	
Crc	*Value*			

Request asynchro data from buffer		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x007E*	*Request asynchro data from buffer*
Registry count	*0x0006*	
Crc	*Value*	

Instruction for getting asynchro data from buffer.
LUMiBOX stores received asynchro data in buffer. If buffer is empty, all received data is 0.
In LUMiCHARGER RFID mode (2) asynchro data contains ID of LUMiCHARGER and user's card ID for authorisation.
Received asynchro data has to be proccesed immediately, because buffer may contains multiple asynchro data from different devices, which are waiting for processing.

Asynchro data are removed from buffer after reading.

WARNING! Asynchro response supports LUMiBOX SLM-160A and SLM-140A with firmware version higher than 2.21 (including).

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x0C*			
Data	*1*	*Hi*	*Asynchro mode*	*0 - No data* *1 - Motion group mode (not yet implemented)* *2 - LUMiCHARGER RFID mode*
		Lo		
Data	*2*	*Hi*	*Group / ID*	*Group (in mode 1)* *LUMiCHARGER ID (in mode 2)*
		Lo		
Data	*3*	*Hi*	*Data*	*0 (in mode 1)* *RFID ID byte (in mode 2)*
		Lo		
Data	*4*	*Hi*	*Data*	*0 (in mode 1)* *RFID ID byte (in mode 2)*
		Lo		
Data	*5*	*Hi*	*Data*	*0 (in mode 1)* *RFID ID byte (in mode 2)*
		Lo		
Data	*6*	*Hi*	*Data*	*0 (in mode 1)* *RFID ID byte (in mode 2)*
		Lo		
Crc	*Value*			

Request instructions

Request response asynchro mode		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x007F*	*Request response asynchro mode*
Registry count	*0x0001*	
Crc	*Value*	

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Reponse mode*	*0 - disabled* *1 - enabled*
		Lo		
Crc	*Value*			

Instruction for getting response asynchro mode set in the modulator.
WARNING! Asynchro response supports LUMiBOX SLM-160A and SLM-140A with firmware version higher than 2.21 (including).

Request response reference mode		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0080*	*Request response reference mode*
Registry count	*0x0001*	
Crc	*Value*	

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x04*			
Data	*1*	*Hi*	*Reference mode - set*	*0 - negative* *1 - positive* *2 - auto*
		Lo		
Data	*2*	*Hi*	*Actual reference mode*	*0 - negative* *1 - positive*
		Lo		
Crc	*Value*			

Instruction for getting response reference mode set in the modulator.
Reference mode - set determines reference mode which is set in modulator.
Actual reference mode shows which reference mode is actual in use (it is relevant only in auto reference mode)
WARNING! Asynchro response supports LUMiBOX SLM-160A and SLM-140A with firmware version higher than 2.24 (including).

Request stored motion and status from luminaire		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0082*	*Request stored motion and status from luminaire*
Registry count	*0x0001*	
Crc	*Value*	

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02*			
Data	*1*	*Hi*	*Actual level byte*	*See tables Response info bytes*
		Lo	*Short info byte*	
Crc	*Value*			

Instruction for getting received actual status of motion input and status of luminaire from the luminaire.
Use this Instruction after control instruction 0x003C.

Request instructions

Request stored motion and status from luminaires		
Name	**Value**	**Description**
Address	*Address*	
Function	*0x03*	
Registry address	*0x0083*	*Request stored motion and status from luminaires*
Registry count	*n*	*Luminaires count (1 - 100)*
Crc	*Value*	

Instruction for getting received actual status of motion input and status of luminaire from the set of luminaires (up to 100).
Use this Instruction after control instruction 0x003D.
In this instruction you don't need to set range of luminaires, but only its count.
Range of luminaires is set in control instruction 0x003D.

Response				
Name	**Value**			**Description**
Address	*Address*			
Function	*0x03*			
Byte count	*0x02 * n*			
	1	*Hi*	*Motion byte*	*See tables Response status info bytes*
		Lo	*Short info byte*	
	...			
	n	*Hi*	*Motion byte*	*See tables Response status info bytes*
		Lo	*Short info byte*	
Crc	*Value*			

Error response for request instruction

Error response		
Name	**Value**	**Description**
Address	***Address***	
Function	***0x03 + 0x80***	*Function + 0x80*
Error code	***Error code***	*See Table of error codes*
Crc	***Value***	

Table of device types

DEVICE TYPE ID	DEVICE TYPE
0x00	3 - phase modulator PANTER PNT340 or PNT360
0x80	1 - phase modulator LUMiBOX SLM
0xFF	LUMiCHARGER RS485

Table of device states

STATE CODE	STATE	DESCRIPTION
0x01	Ok	*The device is ready for coding.*
0x02	Ok after coding	*This code is returned at the first request on state of successful coding.*
0x03	Overheat during coding	*This code is returned if the device was overheated during coding, thus the coding was not successful.*
0x04	Coding	*This code is returned if the coding has not finished yet and there is no error.*
0x05	Overheat	*Information of an overheated device. It is necessary to wait for its cool-down.*
0x06	Missing power supply	*Information that there is no power supply connected to the power modulators.*
0x07	Other error	*Other specified problem.*
0x08	Modulator error	*Detected hardware problem with modulator. This status remains until reset.*
0x09	Measuring	*Device is measuring current or receiving luminaire response.*
0x0A	Overheat after coding	*This code is returned if the device was overheated after successful coding. After read, state changes to 0x0005 - Overheat.*
0x0B	Overvoltage	*Information that was measured overvoltage (voltage over 260V). It is necessary to wait for normal voltage*
0x0C	Overvoltage after coding	*This code is returned if the device measured overvoltage after successful coding. After read, state changes to 0x000B - Overvoltage.*
0x0D	Modulation error after coding	*Modulator detected problem during modulation and command was not modulated correctly.*

Table of error codes

1. If the device received a CRC error code, it will not send a response.
2. If the device is unable to perform an instruction, it will send one of the following codes:

ERROR CODE	ERROR	DESCRIPTION
0x01	Illegal function	*Master has sent a function different than 0x03 or 0x10. Other instructions are not accepted.*
0x02	Illegal data address	*Master has sent an invalid address for the data writing. It is possible to write data only on addresses stated in the table of supported commands.*
0x03	Illegal data value	*Master has sent invalid data. It is necessary to hold onto ranges of data stated in the table of supported commands.*
0x04	Slave device failure	*Error of a component in the PANTER modulator basically means overheat or other error of a modulator. In this case it is necessary to verify the state of the device by sending the instruction for verifying state.*

Response info status bytes

FULL INFO BYTE

Bit	7	6	5	4	3	2	1	0
Value	LUMiNODE ID <7:0>							

ACTUAL LEVEL BYTE

Bit	7	6	5	4	3	2	1	0
Value	ACTUAL LUMINODE LEVEL (+/- 3%) <7:0>							

MOTION BYTE

Bit	7	6	5	4	3	2	1	0
Value	ACTUAL STATUS OF MOTION INPUT(0 - no motion, 1 - motion captured) <7:0>							

SHORT INFO BYTE

Bit	7	6	5	4	3	2	1	0
Value	-	-	-	-	-	Short status <2:0>		

SHORT STATUS <2:0>

000	*No data received from LUMiNODE*
001	*Luminaire OK*
010	*Luminaire error*
011	*Wrong data received from LUMiNODE*
100	*No luminaire connected to LUMiNODE*
101	*DALI communication error*

Response info data byte

DATA INFO BYTE

Bit	7	6	5	4	3	2	1	0
Value	-	-	-	-	-	-	Short info <1:0>	

SHORT INFO <2:0>

00	*No data received from LUMiNODE*
01	*Data OK*
11	*Wrong data received from LUMiNODE*

Table of scene data

		MODE		
DATA	LEVEL	Tc	XY coordinates	RGBWAF
1	0	Tc value (1000K- 20000K)	X coordinate (1 - 65534)	Red color (0 - 254)*
2	0	0	Y coordinate (1 - 65534)	Green color (0 - 254)*
3	0	0	0	Blue color (0 - 254)*
4	0	0	0	White color (0 - 254)*
5	0	0	0	Amber color (0 - 254)*
6	0	0	0	Free color (0 - 254)*

*** 255 is MASK value**

Table of LUMiNODE parameters

PARAMETERS

BYTE	1	2	3	4	5	6	7	8
DATA	FW version 1 (Major)	FW version 2 (Minor1)	FW version 3 (Minor2)	FW version 4 (Config.)	Reference	Depth of modulation	Temperature	Motion type / level
BYTE	9	10	11	12	13	14	15	16
DATA	Motion enabled / hold on time	Max limit	CLO years	CLO reduced power	CLO actual level	Sequence ID	Luminaire type	

It is possible to request range of bytes (for example byte 1 - 4 to get firmware version of LUMiNODE or for example 11 - 13 to get CLO settings)

GROUPS

BYTE	1	2	3	4	5	6	7	8
DATA	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8
BYTE	9	10	11	12	13	14	15	16
DATA	Group 9	Group 10	Group 11	Group 12	Group 13	Group 14	Group 15	Group 16

UART setup

HIGH BYTE VALUE	BAUD RATE
0	1200
1	1800
2	2400
3	4800
4	9600
5	115200

PARITY AND STOP BITS SETUP

Bit	7	6	5	4	3	2	1	0
Value	-	-	-	-	-	Stop bits	Parity <1:0>	

PARITY SETUP (BIT 1-0):

00	*None parity*
01	*Even parity*
10	*Odd parity*

STOP BITS SETUP (BIT 2):

0	*1 stop bit*
1	*2 stop bits*

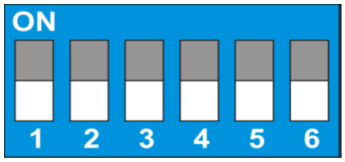
Table of ID range

MODULATOR TYPE	ID RANGE
3 - phase modulator PANTER PNT340 or PNT360	1 - 65535
1 - phase modulator LUMiBOX SLM	1 - 255

Device address

				DEVICE ADDRESS (DIP SWITCH) <4:0>				
Bit	7	6	5	4	3	2	1	0
PANTER	0	0	0	x	x	x	x	x
BROADCAST	0	0	0	0	0	0	0	0

DIP switch



Switches at the position of 1 – 5 define the address of the device on the RS-485 line. DIP 1 is the least significant bit of the address and DIP 5 is the most significant bit of the address.

The switch at position 6 is used for connecting/disconnecting a termination resistor.

The modulator responds only on the address which is set on the switches. For changing the address, it is necessary to set a new address and restart the modulator.

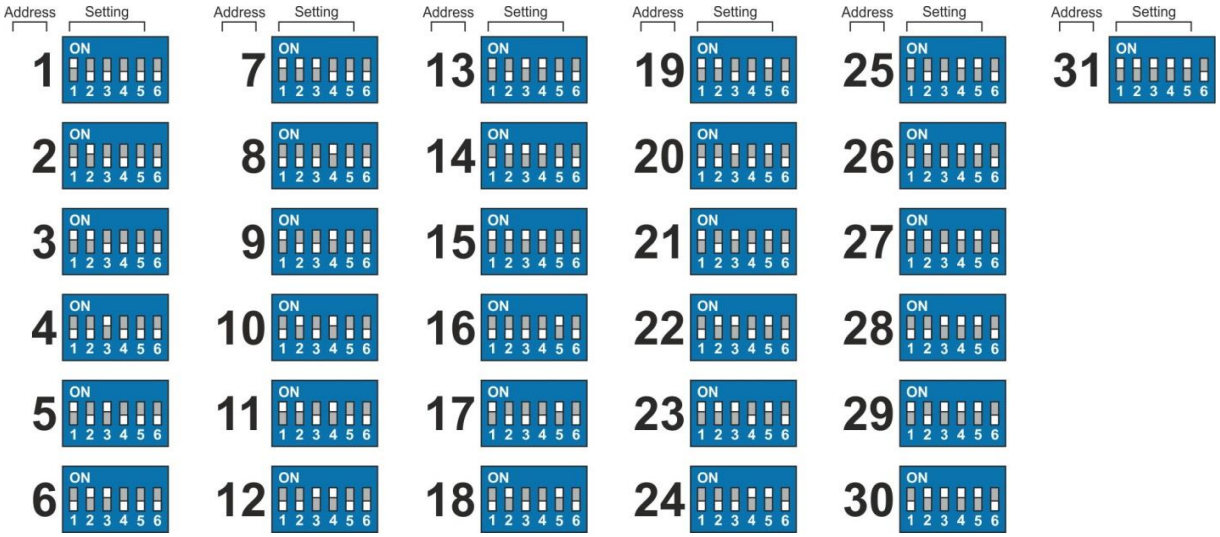
Maximum of 31 PANTER modulators can be addressed on one line with addresses: 0x01 to 0x1F.

Setup when all DIP switches are off on PANTER unit is reserved for setup with initial RS485 parameters. When you have PANTER with unknown RS485 setup, use reserved switch setup 00000 and PANTER will react on address 0x01 on initial RS485 setup parameters shown at the top of this document.

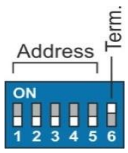
Address calculation according to DIP SWITCH setting

Calculation based on the position of switches on the position 1 – 5 (bit 1 - 5)

Device address = (bit 1) x 2^0 + (bit 2) x 2^1 + (bit 3) x 2^2 + (bit 4) x 2^3 + (bit 5) x 2^4



Terminator setting:



Example of using some functions

Dimming of luminaire procedure

1. *Send command* ***Dimming of luminaire (0x0002).***
2. ***Read state (0x0064)*** *until it changes from* ***CODING*** *to* ***OK AFTER CODING*** *and after that to* ***OK.***

Current measurement procedure

1. *Send command* ***Start current measurement (0x0012).***
2. ***Read state (0x0064)*** *until changes from* ***MEASURING*** *to* ***OK AFTER CODING*** *and after that to* ***OK.***
3. *Read measured current with command* ***Request measured current (0x006D)***

Receiving luminaire status procedure

1. *Send command* ***Send request for status of luminaire (0x0010).***
2. ***Read state (0x0064)*** *until changes from* ***MEASURING*** *to* ***OK AFTER CODING*** *and after that to* ***OK.***
3. *Read received data* ***Request stored luminaire status (0x006C).***

Procedure consists of modulating request command to powerline, receiving response from luminaire and reading luminaire status from SLM unit.

Check 2-way comunication procedure

1. *Send command* ***Send echo data (0x0015)***
2. ***Read state (0x0064)*** *until changes from* ***MEASURING*** *to* ***OK AFTER CODING*** *and after that to* ***OK.***
3. *Read received data* ***Request for stored echo data (0x0077)***

Procedure for checking powerline 2-way comunication by sending „echo data“ and receiving it back.