



u::Lux Message Protocol (UMP)

Protocol Description

www.u-lux.com

office@u-lux.com

Tel: +43/662/450 351-13

Fax: +43/662/450 351-16

u::Lux GmbH

Rechtes Salzachufer 42

5020 Salzburg

Austria



Contents

Introduction	4
UDP-User Datagram Protocol.....	4
Functionality of UDP	4
Port-Structure	4
Structure of an UDP packet.....	4
Used Ports	5
Overview	6
General Information.....	6
Version System	6
Authorisations	6
Data Types.....	6
Starting Process	7
Driver Development	7
Video Transmission.....	8
The Descriptor	9
Overview.....	9
Detailed Description	9
Messages.....	12
Message Structure.....	12
MessageIDs	13
ID-State	15
ID-Init.....	19
ID-Lux	20
ID-Signature.....	21
ID-PageCount	22
ID-IDList	23
ID-Control.....	24
ID-Activate.....	28
ID-PageIndex	29
ID-DateTime	30
ID-Value	32
ID-EditValue	34
ID-RealValue.....	35
ID-LED	37
ID-Text	38
ID-Event	41
ID-I2C-Temperature.....	42
ID-I2C-Humidity	43
ID-I2C-CO2	44
ID-I2C-IN2	45
ID-I2C-VOC	46
ID-AudioVolume	47
ID-AudioStop.....	48
ID-AudioPlayLocal	49
ID-AudioPlayRemote	52
ID-AudioRecord.....	56
ID-VideoState.....	59

u::Lux Message Protocol (UMP)



ID-VideoStart..... 61
ID-VideoStop 62
XX-VideoStream..... 63
XX-VideoStreamEx..... 66
Version Management..... 70



Introduction

The u::Lux Switch uses the Ethernet as a communication medium. The Ethernet is the most frequently used bus system worldwide. The whole Internet is for example based on this network technology. The Ethernet is also one of the most effective and safe bus systems!

In order to communicate with other devices, such as control components (PLC) or other u::Lux Switches, the UDP protocol is used.

UDP-User Datagram Protocol

UDP is a connectionless transport protocol and works on the fourth transport layer of the OSI layer model. It thus has a similar task as the connection-oriented TCP. While TCP is sending verifications at data reception, UDP waives. This has the advantage that the packet header is much smaller and the communication becomes faster.

Functionality of UDP

Contrary to TCP, UDP is missing almost all control functions. As a result the data packets become leaner and are easier to process. The UDP packets are transferred directly to the application. The application itself is responsible for a reliable data transfer.

Port-Structure

The commonality of UDP and TCP is the port-structure which allows multiple applications to have several connections over the network at the same time. In each UDP data packet a number is stored defining a port. Behind this port there is an application or service listening on the port and receiving the UDP's data. UDP ensures that data will not be transferred to the wrong application.

Structure of an UDP packet

UDP packets are composed of the header area and the data area. The header contains all information which allow a quite orderly data transmission and identify an UDO packet as such.

u::Lux Message Protocol (UMP)



Used Ports

The u::Lux Switch uses several UDP ports:

- For audio transmissions port 0x88A4 is used.
- For software updates and management tasks (e.g. allocation of IP addresses, etc.) the port 0x88A8 is used.
- For the communication protocol, which is described in this document, port **0x88AC** is used.

The current information to be transmitted may be found in the data area of the UDP packet.

We call the data format, used in this data area of u::Lux, “u::Lux Message Protocol“ or “UMP” which is described further on the following pages.



Overview

General Information

The data range is basically built by the same system. At the beginning there is the descriptor. In principle two different data ranges are used: The descriptor at the beginning and the user data, packed in a message following the descriptor.

Version System

The frame version consists of a 16-bit value which contains the version information of the frame. That means for example the value 0x0108 stands for version 1.08 and value 0x0201 stands for version 2.01.

We use this value which is divided into two parts – as main version (MSB) and sub version (LSB). The main version has to match with the main PLC always and in all circumstances. If the main PLC receives a packet which does not match with the main version the packet will be rejected. The u::Lux Switch does the same! The reason for this lies in possible extensions of the protocol. The main version is increased if the protocol is extended in such a way that data structures (e.g. the descriptor size changes) move or modify in such a way that the compatibility is at risk. With minor changes or extensions the sub version is increased. The PLC as a frame version has to send always the current implemented protocol version!

Important: The u::Lux Switch sends with every packet the current version. A PLC must ensure that it not only “understands” the protocol but also the correct version!

Authorisations

<i>Read Only</i>	Value can only be read and does not affect the writing
<i>Read/Write</i>	Value can be read and written
<i>Write Only</i>	Values in case of messages, which only can be sent and not requested
<i>Reserved</i>	<i>Reserved</i> shall be sent with 0, when reading, the value has to be ignored
<i>Fixed</i>	A given (fixed) value

Data Types

All data types are transmitted in the Little-Endian Format (LSB first)!

Byte	Size: 8 Bit,	value range: 0-255
Word	Size: 16 Bit,	value range: 0-65.535
Integer	Size: 16 Bit,	value range: -32.768 bis +32.767
Long	Size: 32 Bit,	value range: 0-4.294.967.295
Text	Size: 8 Bit per letter,	termination takes place through one Byte with value 0.



Starting Process

When a voltage is applied to the control (PLC) and the u::Lux NetCon Power/u::Lux NetInj Power these devices are started. The u::Lux Switches are also supplied with voltage through u::Lux NetCon Power/u::Lux NetInj Power. As soon as the u::Lux Switch is started the InitRequest and TimeRequest-Flag are set.

As the u::Lux Switches don't have a real-time clock (battery change would be necessary) the PLC has to provide the time data. These Flags are set in order to communicate an uninitialized state of the switch to the PLC.

If the InitRequest-Flag is set alone or in combination with the TimeRequest-Flag the switch sends approx. every 5 seconds an ID_STATE message, an ID-CONTROL message, an ID-IDList message and an ID-PageCount message to the descriptor.

The PLC has to respond to these requests with an ID-Control message (incl. ControlFlags). Only then the InitRequest-Flag in the Switch is deleted.

Before, all real – and edit values of the used actuators (have been communicated with the ID-IDList command) have to be set – ideally with as few Ethernet packets as possible.

The TimeRequest Flag will only be deleted when the switch receives an ID-DateTime packet (incl. time and date).

If the TimeRequest-Flag is set alone the switch sends approx. every 5 seconds a descriptor and an ID_STATE message.

Driver Development

The PLC has to create a table which saves the allocation of SwitchID, corresponding IP address and appropriate ActorIDs (currently up to 64)!

In the PLC it will be projected only with SwitchID and ActorIDs. These values are provided from u::Lux Config as constants and can be imported into the PLC-control software e.g. TwinCAT.

If an EditValue is changed by a switch the PLC has to check the SwitchID-table and it has to send to all switches which use the same ActorID an ID-EditValue packet with the new changed value.

Alternatively also a Broadcast would be possible, whereby the SwitchID has to be set to 0.

The same applies for the RealValues (real value changes).



Video Transmission

On the switch there can be displayed videos or arbitrary images which can be transmitted through a server. Therefore at the projecting a “video” (placeholder for the later displayed video) has to be available on the corresponding page. A video can be inserted in a function layout or in a page layout. Per page only one video has to be available, the size of the video is arbitrary.

As long as the video is not active it will be displayed on the screen (transparently or in a certain colour, depending on the projecting). If you want to display the video you absolutely have to note the following procedure:

1. First the video state with ID-VideoState is requested. The response contains Flags which can indicate that a video is played currently (VideoActive) and that means that a second video can't be displayed. Additionally there is the Flag VideoPossible which indicates whether a video can be displayed at all. The response contains additionally the range (the X/Y coordinates of the left upper and right lower corner) in which the video is displayed.
2. When the video can be started the command ID-VideoStart has to be sent. This command contains a unique identification of the videostream (SequenceID).
3. Now the video data has to be sent to the switch. For that purpose it will be started with the top line (Bounds.Top) and in each case as many pixels as possible have to be packed in one frame. As the case may be the last packet contains fewer lines in its frame (see calculation methodology and example at the XX-VideoStream).
4. The transmission process is repeated until either the ID-VideoState message is received which communicates that the video is finished (e.g. through a page change) or until the video shall be stopped. In this case the video has to be stopped with ID-VideoStop!



The Descriptor

The descriptor is part of any frame. It serves for the identification of the frame and contains important information about the data processing.

Overview

The following table provides an overview of the contained fields.

	Offset		Designation
	Decimal	Hex	
Descriptor	0-1	0x00-0x01	FrameID
	2-3	0x02-0x03	FrameLength
	4-5	0x04-0x05	FrameVersion
	6-7	0x06-0x07	PackageID
	8-9	0x08-0x09	ProjectID
	10-11	0x0A-0x0B	FirmwareVersion
	12-13	0x0C-0x0D	SwitchID
	14-15	0x0E-0x0F	DesignID
	16- (16+n)	0x10-(0x10+n)	Messages (n = length of the message)

Detailed Description

The descriptor is part of the subsequently described data blocks

Offset 0: FrameID (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	Frame Identification code <i>Fix code to identify the frames. At all messages (ID-XXX) the value 0x8601 is used. At video transmission (VideoStream) the value 0x8602 is used!</i>	<i>Fixed</i>	0x8601 0x8602

u::Lux Message Protocol (UMP)



Offset 2: FrameLength (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	Data length of the frame <i>This value contains the whole length (the length of the descriptor plus the length of the data).</i>	Read/Write	

Offset 4: FrameVersion (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	Version of the frames <i>With this value future extensions can be realized.</i>	Read/Write	

Offset 6: PackageID (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	ID of the packet <i>This field contains a freely selectable value which can't be 0. If a response is sent to a command the same value will be given in the response. Therefore it can be checked if the data packet was received correctly.</i> <i>If one participant sends an event (no response to a command) the PackageID has the value 0!</i>	Read/Write	Not 0

Offset 8: ProjectID (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	ID of the project <i>During configuration of the u::Lux Switch by u::Lux Config this value will be assigned. This serves to check if the participant uses the correct project.</i>	Read Only	

Offset 10: FirmwareVersion (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	Version of the Firmware <i>This value indicates the current version of the firmware. Through the version there can be e.g. determined which functions are supported by the participant and what the features are.</i>	Read/Write	

u::Lux Message Protocol (UMP)



Offset 12: SwitchID (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	ID of the switch <i>During configuration of the u::Lux Switch by u::Lux Config this value will be assigned. Every switch in a project has a unique ID.</i>	<i>Read/Write</i>	

Offset 14: DesignID (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	ID of the Design <i>During configuration of the u::Lux Switch by u::Lux Config this value will be assigned. Every switch in a project has a unique ID.</i>	<i>Read Only</i>	

Offset 16: Messages (variabel size)		Descriptor	
Bit	Description/Note	Mode	Default
	<i>The switch is sending always with its unique ID and accepts only packets which have an ID of 0 or if the ID corresponds to the internal ID.</i>	<i>Read/Write</i>	



Messages

Several messages can be transmitted in one UDP-Frame. The messages are connected directly in series without separator. The descriptor has to be always at the beginning of the message!

Message Structure

The following table provides an overview of the contained fields.

	Offset		Designation
	Decimal	Hex	
Message	0	0x00	MessageLength
	1	0x01	MessageID
	2-3	0x02-0x03	ActorID
	4	0x04	Data with variable size

Offset 0: MessageLength (Byte)		Descriptor	
Bit	Description/Note	Mode	Default
	Full length of the individual message. <i>MessageID+MessageLength+Length of the data</i>	<i>Read/Write</i>	

Offset 1: MessageID (Byte)		Descriptor	
Bit	Description/Note	Mode	Default
	ID for identification of the message type <i>See MessageIDs</i>	<i>Read/Write</i>	

Offset 2: ActorID (Word)		Descriptor	
Bit	Description/Note	Mode	Default
	ID for identification of the control object <i>Has always the value 0 at general tasks</i>	<i>Read/Write</i>	

Offset 4: Data (variabel size)		Descriptor	
Bit	Description/Note	Mode	Default
	Data <i>See Detailed description of the individual messages</i>	<i>Read/Write</i>	



MessageIDs

The following table shows the currently possible values for the MessageID. Values which are not included have to be ignored by the PLC!

Name	Decimal	Hex	Function
ID-State	1	0x01	Read/Request of the StateFlags
ID-Init	2	0x02	Sets certain StateFlags
ID-Lux	3	0x03	Transmission of the brightness
ID-Signature	4	0x04	Signature/Checksums message
ID-PageCount	14	0x0E	Read/request page number
ID-IDList	15	0x0F	Read/Request the ActorID List
ID-Control	33	0x21	Read/Request/Write ControlFlags
ID-Activate	45	0x2D	Activates or Deactivates the switch
ID-PageIndex	46	0x2E	Read/Request/Write the PageIndex
ID-DateTime	47	0x2F	Read/Request/Write time & Date
ID-Value	65	0x41	Read/Request/Write RealValue, EditValue and LEDs of a certain actor
ID-EditValue	66	0x42	Read/Request/Write EditValues of a certain actor
ID-RealValue	67	0x43	Read/Request/Write RealValues of a certain actor
ID-LED	68	0x44	Read/Request/Write LEDs of a certain actor
ID-Text	69	0x45	Read/Request/Write texts of a certain actor
ID-Event	81	0x51	Transmission of an events (key(s) pressed or released)
ID-I2C-Temperature	113	0x71	Transmission of the temperature
ID-I2C-Humidity	114	0x72	Transmission of humidity
ID-I2C-CO2	115	0x73	Transmission of CO2-Concentration
ID-I2C-IN2	116	0x74	Transmission of IN2-State
ID-I2C-VOC	117	0x75	Transmission of VOC-Concentration
ID-AudioVolume	145	0x91	Read/Request/Write volume
ID-AudioStop	146	0x92	Stops the audio output or audio recording
ID-AudioPlayLocal	152	0x98	Playback of a local saved sound
ID-AudioPlayRemote	153	0x99	Playback of an audio stream
ID-AudioRecord	154	0x9A	Recording and streaming a sound

u::Lux Message Protocol (UMP)



Name	Decimal	Hex	Function
ID-VideoState	161	0xA1	Read/Request the videostate
ID-VideoStart	162	0xA2	Start a videotransmission
ID-VideoStop	163	0xA3	Stops a videotransmission
XX-VideoStream			Sending videodata



ID-State

With the message ID-State the StateFlags are transmitted. The StateFlags are sent automatically from the switch and if required they can be requested as this message is sent with MessageLength=0x04 (without the StateFlags).

	Offset		Designation
	Decimal	Hex	
ID-State	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x01
	2-3	0x02-0x03	ActorID=0x00
	4-7	0x04-0x07	StateFlags

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: StateFlags (Long)		StateFlags	
Bit	Description/Notes	Mode	Default
31-30	Reserved <i>Currently not used, therefore value 0!</i>		
29	I2CMotionDetectorValid <i>1=I2C-AddOn MotionDetector valid If this Bit is set an AddOn with motion detector is connected and active</i>	Read Only	0
28	I2CVOCValid <i>1=I2C-AddOn VOC-Sensor valid If this Bit is set an AddOn with VOC-sensor is connected and active (i.e. the VOC values are valid)</i>	Read Only	0
27	I2CIN2Valid <i>1=I2C-AddOn IN2 valid If this Bit is set an AddOn IN2 (2 inputs) is connected and active (i.e. the state of the keys is valid)</i>	Read Only	0
26	I2CCO2Valid <i>1=I2C-AddOn CO2-Sensor valid If this Bit is set an AddOn with CO2-sensor is connected and active (i.e. the CO2 values are valid)</i>	Read Only	0
25	I2CHumidityValid <i>1=I2C-AddOn Humidity valid If this Bit is set an AddOn with humidity-sensor is connected and active (i.e. the humidity values are valid)</i>	Read Only	



Offset 4: StateFlags (Long)		StateFlags	
Bit	Description/Notes	Mode	Default
24	I2CTemperatureValid <i>1=I2C-AddOn Temperature valid If this Bit is set an AddON with temperature-sensor is connected and active (i.e. the temperature is valid)</i>	Read Only	
23-13	Reserved <i>Currently not used. Therefore value 0!</i>	Reserved	
12	LUXValid <i>If this Bit is set the u::Lux Switch contains a brightness sensor which provides and measures the brightness in Lux. (The analog brightness sensor is only available starting with u::Lux Display serialnumber 2700)</i>		
11	I2CMotionDetector <i>1=AddOn MotionDetector detects the object This Bit indicates if the optional AddOn MotionDetector has detected an object within its range (typical approx. 5m). This function can be used e.g. as motion detector to switch on the light in the corridor. Through the Bit 11 in the ControlFlags there can be adjusted if in case of changes of the motion detector a packet shall be sent.</i>	Read Only	
10-8	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	
7	InternalError <i>1=serious error If this Bit has the value 1 the participant has a serious error and has to be checked.</i>	Read Only	0
6	InitRequest <i>1=InitRequest active If this Bit is set the participant needs a packet with the current ControlBlock. As soon as the participant receives a packet with a ControlBlock this Bit will be deleted. This function serves primarily to inform the PLC that the participant (e.g. after a software update) has restarted.</i>	Read Only	



Offset 4: StateFlags (Long)		StateFlags	
Bit	Description/Notes	Mode	Default
5	<p>TimeRequest <i>I=TimeRequest active</i> If this Bit is set the participant needs a TimeSync packet. As soon as the participant receives a packet this Bit will be deleted. But the Bit can be set again at any other time!</p>	Read Only	
4	<p>IntroActive <i>I=The Intro is active</i> This Bit indicates if the Intro-Animation is active.</p>	Read Only	
3	<p>AudioActive <i>I=Audio active</i> This Bit indicates if an Audio Function is active. It doesn't matter if it's a playback (Play) or recording (Record)! Through the Bit 3 can be adjusted if in case of changes of the audiostate a packet shall be sent.</p>	Read Only	
2	<p>DisplayActive <i>I=Diplay active</i> This Bit indicates if the display and the display illumination are switched on. Through Bit 2 in the ControlFlags can be adjusted if in case of changes of the display illumination a packet shall be sent.</p>	Read Only	
1	<p>ProximitySensor <i>I=Object within range</i> This Bit indicates if the proximity sensor has detected an object within its range (approx. 25cm). This function can be used e.g. as motion detector to switch on the light in the corridor. Through Bit 1 in the ControlFlag there can be adjusted if in case of changes of the proximity sensor a packet shall be sent.</p>	Read Only	
0	<p>LightSensor <i>I=light</i> This Bit indicates if the LightSensor classifies the ambient light as light or dark. Through Bit 0 in the ControlFlags there can be adjusted if in case of changes of the light sensor a packet shall be sent.</p>	Read Only	

u::Lux Message Protocol (UMP)



Note to the StateFlags: All Bits with exception of the following list are updated in real time!

The exceptions are as follows:

- LightSensor
The LightSensor is only updated if the Bit ProximitySensor is not set or the “intelligent brightness sensor” in the settings of the u::Lux Config project is deactivated.
- I2C-values Temperature, Humidity, CO2, VOC, and the relative Bits I2CTemperatureValid, I2CHumidityValid, I2CCO2Valid and I2CVOCValid are updated approx. every 15 seconds. Is the Flag I2CPlugAndPlay set in the ControlFlags, the update interval is approx. 2,5 seconds.
- If an AddOn-MD (motion detector) is connected its state is updated approx. 8 times per seconds (every 125ms). Changes will be sent immediately.
- If an AddOn-IN2 (2 digital inputs) is connected its state will be updated approx. 8 times per second (every 125ms). Changes will be sent immediately.



ID-Init

With the message ID-State the switch is set to the state in which it is after a reset or after applying the voltage, the display of the switch will not change. For this certain Bits (depending on the InitFlags) will be set in the StateFlags.

	Offset		Designation
	Decimal	Hex	
ID-Init	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x02
	2-3	0x02-0x03	ActorID=0x00
	4-5	0x04-0x05	InitFlags
	6-7	0x06-0x07	InitCode

For the description of MessageLength, MessageID and ActorID please refer to chapter Message Structure.

Offset 4: InitFlags (Word)		ID-Init	
Bit	Description/Notes	Mode	Default
12-15	Restart <i>If these 4 Bits (all 4) are set the u::Lux Switch carries out a restart (Reset).</i>	Write Only	
7-11	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	
6	SetInitRequest <i>If this Bit is set the Bit InitRequest is set in the StateFlags.</i>	Write Only	
5	SetTimeRequest <i>If this Bit is set the Bit TimeRequest is set in the StateFlags.</i>	Write Only	
0-4	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	

Offset 6: InitCode (Word)		ID-Init	
Bit	Description/Notes	Mode	Default
	Security Code <i>Only if the InitCode has the value 0x5AA5 the InitFlags will be accepted, otherwise the InitFlags will be ignored!</i>	Write Only	0x5AA5

u::Lux Message Protocol (UMP)



ID-Lux

This message is always sent if the analog brightness sensor value has changed and the bit I2CLuxChangeRequest is set in the ControlFlags. Value changes are transmitted immediately.

This message can also be requested by sending this message with MessageLength=0x04 (without the data).

IMPORTANT:

The analog brightness sensor is only available starting with u::Lux Display serialnumber 2700, the bit LuxValid in the StateFlags iss set in this case!

	Offset		Designation
	Decimal	Hex	
ID-LUX	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x03
	2-3	0x02-0x03	ActorID=0x00
	4-5	0x04	Lux
	6	0x06	Valid
	7	0x07	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: Lux (Word)		ID-I2C-IN2	
Bit	Description/Note	Mode	Default
	Lux <i>Indicates the brightness value in Lux. The value is only valid if Valid has the value 0x01.</i>	<i>Read Only</i>	

Offset 6: Valid (Byte)		ID-I2C-IN2	
Bit	Description/Note	Mode	Default
	Validity <i>Only if this Byte has the value 0x01 the brightness value Lux is valid!</i>	<i>Read Only</i>	0

Offset 7: Reserved (Byte)		ID-I2C-IN2	
Bit	Description/Note	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	

u::Lux Message Protocol (UMP)



ID-Signature

This message is not supported any more and was removed.



ID-PageCount

With this command the number of the projected pages is transmitted. This message can be also requested as this message is sent with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-PageCount	0	0x00	MessageLength=0x06
	1	0x01	MessageID=0x0E
	2-3	0x02-0x03	ActorID=0x00
	4	0x04	PageCount
	5	0x05	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: PageCount (Byte)		ID-PageCount	
Bit	Description/Note	Mode	Default
	Number of pages <i>PageCount contains the number of the projected pages.</i>	<i>Read Only</i>	

Offset 5: Reserved (Byte)		ID-PageCount	
Bit	Description/Note	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	



ID-IDList

With this command the ActorIDList is transmitted. Every switch contains up to 64 ActorIDs, in which every ActorID stands for a controllable element, e.g. a load (light) or a sensor (temperature sensor). This message can be also requested as the message is sent with MessageLength=0x04 (without the data).

	Decimal	Hex	Designation
ID-IDList	0	0x00	MessageLength=0x06+2*ActorIDCount
	1	0x01	MessageID=0x0F
	2-3	0x02-0x03	ActorID=0x00
	4-5	0x04-0x05	ActorIDCount
	6-7	0x06-0x07	ActorIDList[ActorIDCount]

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: ActorIDCount (Word)		ID-IDList	
Bit	Description/Notes	Mode	Default
	ActorID number <i>Contains the number of ActorIDs (0..64)</i>	<i>Read Only</i>	

Offset 6: ActorIDList[0..n] (Word)		ID-IDList	
Bit	Description/Notes	Mode	Value
	ActorID List <i>Contains a list of ActorIDs, the length of the list is determined by the value ActorIDCount.</i>	<i>Read Only</i>	



ID-Control

With this command the ControlFlags are transmitted. With the ControlFlags the action of the switch can be influenced. This message can also be requested as the message is sent with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-Control	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x21
	2-3	0x02-0x03	ActorID=0x00
	4-7	0x04-0x07	ControlFlags

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: ControlFlags (Long)		ControlFlags	
Bit	Description/Notes	Mode	Default
31	I2CPlugAndPlay <i>1=I2C-AddOn PlugAndPlay</i> <i>This Bit is only used for demonstrational purposes and in operation the value has always to be 0!</i>	<i>Read/Write</i>	0
30-29	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	
28	I2CVOCChangeRequest <i>1=I2C-AddOn VOC Request active</i> <i>If this Bit is set a packet will be sent every time the VOC-value is changing.</i>	<i>Read/Write</i>	0
27	I2CIN2ChangeRequest <i>1=I2C-AddOn IN2 Request active</i> <i>If this Bit is set a packet will be sent every time one of the inputs of the IN2 AddOn is changing.</i>	<i>Read/Write</i>	0
26	I2CCO2ChangeRequest <i>1=I2C-AddOn CO2 Request active</i> <i>If this Bit is set a packet will be sent every time the CO2-value is changing.</i>	<i>Read/Write</i>	0
25	I2CHumidityChangeRequest <i>1=I2C-AddOn Humidity Request active</i> <i>If this Bit is set a packet will be sent every time the humidity value is changing.</i>	<i>Read/Write</i>	0



Offset 4: ControlFlags (Long)		ControlFlags	
Bit	Description/Notes	Mode	Default
24	I2CTemperatureChangeRequest <i>1=I2C-AddOn Temperature Request active</i> <i>If this Bit is set a packet will be sent every time the temperature value is changing.</i>	Read/Write	0
23-16	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	
15-14	Control of the Display background illumination <i>00=background illumination automatically, day Mode</i> <i>01= background illumination automatically, night Mode</i> <i>10= background illumination is always on</i> <i>11= background illumination is always off</i> <i>The background illumination of the display is controlled depending on these Bits and the project settings.</i>	Read/Write	00
13-12	Lock mode <i>00=no lock</i> <i>01=only navigation keys are locked</i> <i>10=all keys are locked</i> <i>11=all keys are locked and the logo is displayed</i>	Read/Write	00
11	I2CMotionDetectorChangeRequest <i>1=I2CMotionDetectorChangeRequest is active</i> <i>If this Bit is set a packet will be sent every time the state of the motion sensor changes (object within range or object outside range).</i>	Read/Write	0
10	Keep Alive <i>1= Keep Alive activated</i> <i>If this Bit is set every time a packet will be sent when the assigned Timeout time in the project is elapsed after the last packet has been sent.</i>	Read/Write	0
9	Filter Change <i>1=Filter Change activated</i> <i>If this Bit is set the number of changing packets is limited and max. every 5 seconds (if there are changes) a packet is sent if there are changes of the brightness, the proximity sensor, the Display state and the I2C AddOns Temperature and Humidity.</i> <i>If this Bit is not set changes will be immediately signaled through a packet.</i>	Read/Write	0



Offset 4: ControlFlags (Long)		ControlFlags	
Bit	Description/Notes	Mode	Default
8	<p>Frame-Confirmation <i>1=ACK-Frame will be sent</i> <i>If this Bit is set with every received packet a response packet will be sent as confirmation. The response packet contains all sent messages.</i></p>	Read/Write	0
7	<p>LuxChangeRequest <i>1=LuxChangeRequest is active</i> <i>If this Bit is set a ID-Lux packet will be sent every time the brightness sensor changes it's value</i></p>	Reserviert	
6	<p>Reserved <i>Currently not used, therefore value 0!</i></p>	Reserved	
5	<p>VolumeChangeRequest <i>1=VolumeChangeRequest is active</i> <i>If this Bit is set a packet will be sent every time the user changes the volume.</i></p>	Read/Write	1
4	<p>PageChangeRequest <i>1=PageChangeRequest is active</i> <i>If this Bit is set a packet will be sent every time the page changes. Starting with firmwareversion 2.03 on set Bit PageChangeRequest the answer ID-PageIndex is also sent even the Page has NOT changed!</i></p>	Read/Write	1
3	<p>AudioActiveChangeRequest <i>1=AudioActiveChangeRequest is active</i> <i>If this Bit is set a packet will be sent every time the Audio-state (loudspeaker or microphone on/off) is changed.</i></p>	Read/Write	0
2	<p>DisplayActiveChangeRequest <i>1=DisplayActiveChangeRequest is active</i> <i>If this Bit is set a packet will be sent every time the state of the display (on or off) changes.</i></p>	Read/Write	0
1	<p>ProximitySensorChangeRequest <i>1=ProximitySensorChangeRequest is active</i> <i>If this Bit is set a packet will be sent every time the state of the proximity sensor changes (object within range or object outside range)</i></p>	Read/Write	0

u::Lux Message Protocol (UMP)



<i>Offset 4: ControlFlags (Long)</i>		<i>ControlFlags</i>	
Bit	Description/Notes	Mode	Default
0	LightSensorChangeRequest <i>1=LightSensorChangeRequest is active</i> <i>If this Bit is set a packet will be sent every time the state of light sensor (light or dark) changes.</i>	<i>Read/Write</i>	0



ID-Activate

With this command the u::Lux Switch can be activated or deactivated. Activating means that the display of the switch is activated. Deactivating means that the display of the switch is deactivated and the screensaver (if projected) is displayed. Requesting this message is not possible.

	Offset		Designation
	Decimal	Hex	
ID-Activate	0	0x00	MessageLength=0x06
	1	0x01	MessageID=0x2D
	2-3	0x02-0x03	ActorID=0x00
	4	0x04	Flags
	5	0x05	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: Flags (Byte)		ID-PageIndex	
Bit	Description/Notes	Mode	Default
7-2	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	
1	DisplayDeactivate <i>1=If this Bit is set the display of the u::Lux Switch goes into standby mode. This Bit is only valid if the Bit DisplayActive has the value 0! The value 0 has no function.</i>	<i>Read/Write</i>	0
0	DisplayActivate <i>1=If this Bit is set the u::Lux Switch is activated and the display is switches on. If this Bit is set the Bit DisplayDeactivate is ignored! The value 0 has no function.</i>	<i>Read/Write</i>	0

Offset 5: Reserved (Byte)		ID-PageIndex	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	



ID-PageIndex

With this command the number of the current page is transmitted. The switch sends this command if the page index (page number) changes and if the Bit PageChangeRequest is set in the ControlFlags. If you call the screensaver the page index will not change! With this message a PLC can indicate a certain page. This message can also be requested if the message is sent with MessageLength=0x04 (without the data).

IMPORTANT starting with firmwareversion V2.03: If the PageChangeRequest Bit ist set in the ControlFlags, this Command is always (independent of an Change of PageIndex) answered with an ID-PageIndex answer!

	Offset		Designation
	Decimal	Hex	
ID-PageIndex	0	0x00	MessageLength=0x06
	1	0x01	MessageID=0x2E
	2-3	0x02-0x03	ActorID=0x00
	4	0x04	PageIndex
	5	0x05	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: PageIndex (Byte)		ID-PageIndex	
Bit	Description/Notes	Mode	Default
	Displaxed Page <i>Contains the value of the displayed page (0=1. page, 1=2. page, 31=32. page)</i>	<i>Read/Write</i>	0

Offset 5: Reserved (Byte)		ID-PageIndex	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	



ID-DateTime

This message serves basically to synchronise time and date of the u::Lux Switch. Since there is no realtime clock installed in the u::Lux Switch it has to be adjusted precisely from external.

This packet has to be sent from a central place which contains an exact time. We recommend sending this packet as Broadcast because then all switches receive at the same time the date/time. The packet has to be sent after every full hour change. Further more this packet has to be sent when the central control starts or when an u::Lux Switch requests the packet by setting the Bit TimeRequest in the StateFlags.

	Offset		Designation
	Decimal	Hex	
ID-DateTime	0	0x00	MessageLength=0x0C
	1	0x01	MessageID=0x2F
	2-3	0x02-0x03	ActorID=0x00
	4	0x04	Second
	5	0x05	Minute
	6	0x06	Hour
	7	0x07	DayOfWeek
	8	0x08	Day
	9	0x09	Month
	10-11	0x0A-0x0B	Year

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: Second (Byte)		ID-DateTime	
Bit	Description/Notes	Mode	Default
	Second <i>Contains the current second in the range 0-59</i>	Read/Write	

Offset 5: Minute (Byte)		ID-DateTime	
Bit	Description/Notes	Mode	Default
	Minute <i>Contains the current minute in the range 0-59</i>	Read/Write	

u:Lux Message Protocol (UMP)



Offset 6: Hour (Byte)		ID-DateTime	
Bit	Description/Notes	Mode	Default
	Hour <i>Contains the current hour in the range 0-23</i>	Read/Write	

Offset 7: DayOfWeek (Byte)		ID-DateTime	
Bit	Description/Notes	Mode	Default
	DayOfWeek <i>Contains the current day of the week in the range 0-6. 0=Sunday, 1=Monday, 2=Tuesday, 3=Wednesday, 4=Thursday, 5=Friday, 6=Saturday</i>	Read/Write	

Offset 8: Day (Byte)		ID-DateTime	
Bit	Description/Notes	Mode	Default
	Day <i>Contains the current day in the range 1-31</i>	Read/Write	

Offset 9: Month (Byte)		ID-DateTime	
Bit	Description/Notes	Mode	Default
	Month <i>Contains the current month in the range 1-12 (1=January)</i>	Read/Write	

Offset 10: Year (Word)		ID-DateTime	
Bit	Description/Notes	Mode	Default
	Year <i>Contains the current year (e.g. 2012 = Year 2012)</i>	Read/Write	



ID-Value

With this command the edit value (EditValue – this value can be changed by the user of the u::Lux Switch) and up to 4 real values (RealValue – this value can't be changed by the user of the u::Lux Switch) of a certain actor are transmitted. This command contains EditValue and RealValues – if only one variable shall be set please use the command ID-EditValue or ID-RealValue! This message can also be requested by sending the message with MessageLength=0x04 (without the data) In this case the response contains all 4 real values.

	Offset		Designations
	Decimal	Hex	
ID-Value	0	0x00	MessageLength=0x08, 0xA, 0xC, 0xE
	1	0x01	MessageID=0x41
	2-3	0x02-0x03	ActorID=0x0001..0xFFFF
	4-5	0x04-0x05	EditValue
	6-7	0x06-0x07	RealValues[0]
	8-9	0x08-0x09	RealValues[1]
	10-11	0x0A-0x0B	RealValues[2]
	12-13	0x0C-0x0D	RealValues[3]

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: EditValue (Integer)		ID-Value	
Bit	Description/Notes	Mode	Default
	Edit Value <i>This value is the edit value of the corresponding actor. The value is changed directly by the user and contains e.g. the setpoint temperature or e.g. the edit value of the lamp brightness. A change on the switch is not communicated through this command but through the command ID-EditValue to the control!</i>	Read/Write	



Offset 6: RealValues[0] (Integer)		ID-Value	
Bit	Description/Notes	Mode	Default
	RealValue <i>This parameter is the 1. Real value of the corresponding actor. This value can't be changed through the user-interface of the u::Lux Switch and serves only as feedback from the central control. Through this parameter e.g. the real temperature can be indicated while through the EditValue the set temperature is adjusted by the user.</i>	Read/Write	

Offset 8: RealValues[1] (Integer)		ID-Value	
Bit	Description/Notes	Mode	Default
	RealValue <i>This parameter is the 2. real value of the corresponding actor. Other description: See RealValues[0].</i>	Read/Write	

Offset 10: RealValues[2] (Integer)		ID-Value	
Bit	Description/Notes	Mode	Default
	RealValue <i>This paramter is the 3. real value of the corresponding actor. Other description: See RealValues[0].</i>	Read/Write	

Offset 12: RealValues[3] (Integer)		ID-Value	
Bit	Description/Notes	Mode	Default
	RealValue <i>This paramter is the 4. real value of the corresponding actor. Other description: See RealValues[0].</i>	Read/Write	

u::Lux Message Protocol (UMP)



ID-EditValue

With this command the EditValue of a corresponding actor is transmitted. This message can also be requested by sending this message with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-EditValue	0	0x00	MessageLength=0x06
	1	0x01	MessageID=0x42
	2-3	0x02-0x03	ActorID=0x0001..0xFFFF
	4-5	0x04-0x05	EditValue

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: EditValue (Integer)		ID-EditValue	
Bit	Description/Notes	Mode	Default
	EditValue <i>This value is the edit value of the corresponding actor. The value is changed directly by the user and contains e.g. the edit value for the lamp brightness. A change on the switch is transmitted through this command to the control.</i>	Read/Write	0



ID-RealValue

With this command the RealValue of a corresponding actor is transmitted. This message can also be requested by sending the message with MessageLength=0x04 (without data). In this case the response contains all four real values.

	Offset		Designation
	Decimal	Hex	
ID-RealValue	0	0x00	MessageLength=0x06, 0x08, 0x0A, 0x0C
	1	0x01	MessageID=0x43
	2-3	0x02-0x03	ActorID=0x0001..0xFFFF
	4-5	0x04-0x05	RealValues[0]
	6-7	0x06-0x07	RealValues[1]
	8-9	0x08-0x09	RealValues[2]
	10-11	0x0A-0x0B	RealValues[3]

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: RealValues[0] (Integer)		ID-RealValue	
Bit	Description/Notes	Mode	Default
	RealValue <i>This parameter is the 1. real value of the corresponding actor. The value can't be changed through the user-interface of the u::Lux Switch and serves only as feedback from the central control. Through this parameter the e.g. real temperature can be indicated while through the EditValue the set temperature is adjusted by the user.</i>	Read/Write	0

Offset 6: RealValues[1] (Integer)		ID-RealValue	
Bit	Description/Notes	Mode	Default
	RealValue <i>This parameter is the 2. real value of the corresponding actor. Other description: See RealValues[0].</i>	Read/Write	0



Offset 8: RealValues[2] (Integer)		ID-RealValue	
Bit	Description/Notes	Mode	Default
	RealValue <i>This parameter is the 3. real value of the corresponding actor. Other description: See RealValues[0].</i>	Read/Write	0

Offset 10: RealValues[3] (Integer)		ID-RealValue	
Bit	Description/Notes	Mode	Default
	RealValue <i>This parameter is the 4. real value of the corresponding actor. Other description: See RealValues[0].</i>	Read/Write	0



ID-LED

With this command the LEDs of a certain actor can be transmitted. This message can also be requested by sending this message with MessageLength=0x04 (without data).

	Offset		Designation
	Decimal	Hex	
ID-LED	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x44
	2-3	0x02-0x03	ActorID=0x0001..0xFFFF
	4-7	0x04-0x07	LEDColorAndBlinkMode[0..3]

The number of the LEDs is based on the function layout of the corresponding actor. If the actor has only one Led the index 0 is used, with 2 LEDs index 0 and 1 is used and with 4 LEDs all indices (0..3) are used. The index itself is relative and independent from the position of the function layout!

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4/5/6/7: LEDColorAndBlinkMode[0..3] (Byte)		ID-LED	
Bit	Description/Notes	Mode	Default
7	Override <i>If this Bit is set the settings of the UMP protocol for the control of the LEDS is used, with deleted Bit the project settings are used.</i>	Read/Write	0
6-4	Blinkmode <i>These Bits indicate the blinking speed of the LED. At the value 0x0 the LED illuminates constantly, at the value 0x1 slow (approx. 1Hz), at the value 0x2 medium (approx. 2Hz), at value 0x3 fast (approx. 4 Hz), at value 0x4 asymmetrically (½ second on, ¼ second off), at value 0x5 asymmetrical (½ second on, 4 seconds off), at value 0x6 asymmetrical (½ seconds on, 16 seconds off) and at value 0x7 asymmetrical (½ second on, 32 seconds off)</i>	Read/Write	0
3	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	
0-2	Color <i>Color of the LED: 0x0=off, 0x1=red, 0x2=green, 0x3=yellow, 0x4=blue, 0x5=magenta, 0x6=cyan, 0x7=white.</i>	Read/Write	0



ID-Text

With this command user defined texts of a certain actor can be transmitted. This message can be requested by sending this message with MessageLength=0x0C (without text).

Please note that the request is different from the other requests! At the request the value TextID has to be set, Color, Flags and Reserved must have value 0!

	Offset		Designation
	Decimal	Hex	
ID-Text	0	0x00	MessageLength=0x0C+Textlänge
	1	0x01	MessageID=0x45
	2-3	0x02-0x03	ActorID=0x0001..0xFFFF
	4-7	0x04-0x07	Color
	8	0x08	TextID
	9	0x09	Flags
	10-11	0x0A-0x0B	Reserved
	12	0x0C..	Text

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

The length of the message is depending on the text length, whereby the text length corresponds to the number of characters in the text (incl. termination character). If you want to **delete** the text the RemoveFlag has to be set. The text entry will be deleted from the memory, information about color, Flags, etc. are lost!

Offset 4: Color (Long)		ID-Text	
Bit	Description/Notes	Mode	Default
25-31	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	0
16-24	BlueIntensity <i>Indicates the blue component of the desired text color</i>	<i>Read/Write</i>	0
8-15	GreenIntensity <i>Indicates the green component of the desired text color</i>	<i>Read/Write</i>	0
0-7	RedIntensity <i>Indicates the red component of the desired text color</i>	<i>Read/Write</i>	0



Offset 8: TextID (Byte)		ID-Text	
Bit	Description/Notes	Mode	Default
	TextID <i>Indicates the index of the text, value 0 is corresponding to u::Lux Config Variable TEXT.0 and value 9 to u::Lux Config Variable TEXT.9. Value range: 0 to 9.</i>	Write Only	0

Offset 9: Flags (Byte)		ID-Text	
Bit	Description/Notes	Mode	Default
6-7	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	0
5	Remove <i>If this Bit is set the text is removed from the memory. Important: This Flag has to be set alone. As soon as other Flags are set this Flag will be ignored!</i>	Write Only	
4	Visible <i>This Bit indicates if the text is visible (1) or not (0). The Bit is only accepted if the Bit UpdateVisibility has value 1.</i>	Read/Write	0
3	UpdateVisibility <i>This Bit indicates if the Bit Visible shall be accepted (1) or not (0).</i>	Read/Write	0
2	UpdateText <i>This Bit indicates if the text shall be accepted (1) or not (0).</i>	Read/Write	0
1	UpdateColor <i>This Bit indicates if the color shall be accepted (1) or not (0).</i>	Read/Write	0
0	UseColor <i>This Bit indicates if the color shall be used (1) or the project defined text colour (0).</i>	Read/Write	0

Offset 10: Reserved (Word)		ID-Text	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	0

u::Lux Message Protocol (UMP)



Offset 12: Text (Text)		ID-Text	
Bit	Description/Notes	Mode	Default
	<p>Text</p> <p><i>Indicates the text whereby as last character the termination character has to follow (Byte value 0).</i></p> <p><i>All visible characters are allowed (e.g. a-z, A-Z, 0-9, special characters like e.g. [%\$). Other control characters (e.g. CarriageReturn, LineFeed) are not allowed!</i></p>	<i>Read/Write</i>	



ID-Event

This message is always sent when the user pushes or releases a key and the appropriate function layout is used as event. Requesting this message is not possible.

	Offset		Designation
	Decimal	Hex	
ID-Event	0	0x00	MessageLength=0x06
	1	0x01	MessageID=0x51
	2-3	0x02-0x03	ActorID=0x0001..0xFFFF
	4	0x04	KeyState
	5	0x05	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: KeyState (Byte)		ID-Event	
Bit	Description/Notes	Mode	Default
7-4	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	
3	KeyState for 4. key (if available in function layout) <i>1=pressed; 0=not pressed</i> <i>The position is related relatively to the function layout!</i>	<i>Read Only</i>	
2	KeyState for 3. key (if available in function layout) <i>1=pressed; 0=not pressed</i> <i>The position is related relatively to the function layout!</i>	<i>Read Only</i>	
1	KeyState for 2. key (if available in function layout) <i>1=pressed; 0=not pressed</i> <i>The position is related relatively to the function layout!</i>	<i>Read Only</i>	
0	KeyState for 1. key <i>1=pressed; 0=not pressed</i> <i>The position is related relatively to the function layout!</i>	<i>Read Only</i>	

Offset 5: Reserved (Byte)		ID-Event	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	



ID-I2C-Temperature

This message is always sent when an AddOn is connected which provides a temperature value and the Bit I2CTemperatureChangeRequest is set in the ControlFlags. Value changes are transmitted at a distance of approx. 15 seconds. This message can also be requested by sending this message with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-I2C- Temperature	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x71
	2-3	0x02-0x03	ActorID=0x00
	4-5	0x04	Temperature
	6	0x06	Valid
	7	0x07	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: Temperature (Integer)		ID-I2C-Temperature	
Bit	Description/Notes	Mode	Default
	Temperature value <i>This value contains the absolute temperature value. The value 257 corresponds to 25,7°C. The value is only valid if the Byte Valid has the value 0x01!</i>	Read Only	0

Offset 6: Valid (Byte)		ID-I2C-Temperature	
Bit	Description/Notes	Mode	Default
	Validity <i>Only if this Byte has value 0x01 the temperature value is valid!</i>	Read Only	0

Offset 7: Reserved (Byte)		ID-I2C-Temperature	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	



ID-I2C-Humidity

This message is always sent if an AddOn is connected which provides a humidity value and the Bit I2CHumidityChangeRequest is set in the ControlFlags. Value changes are transmitted at a distance of approx. 15 seconds.

This message can also be requested by sending this message with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-I2C-Humidity	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x72
	2-3	0x02-0x03	ActorID=0x00
	4-5	0x04	Humidity
	6	0x06	Valid
	7	0x07	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: Humidity (Word)		ID-I2C-Humidity	
Bit	Description/Notes	Mode	Default
	Humidity value <i>This value contains the relative humidity. The value 588 corresponds to 58,8%. The value is only valid if the Byte Valid has the value 0x01!</i>	Read Only	0

Offset 6: Valid (Byte)		ID-I2C-Humidity	
Bit	Description/Notes	Mode	Default
	Validity <i>Only if this Byte has value 0x01 the humidity value is valid.</i>	Read Only	0

Offset 7: Reserved (Byte)		ID-I2C-Humidity	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	



ID-I2C-CO2

This message is always sent if an AddOn is connected which provides a CO2 value and the Bit I2CCO2ChangeRequest is set in the ControlFlags. Value changes are transmitted at a distance of approx. 15 seconds.

This message can also be requested by sending this message with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-I2C-CO2	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x73
	2-3	0x02-0x03	ActorID=0x00
	4-5	0x04	CO2
	6	0x06	Valid
	7	0x07	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: CO2 (Integer)		ID-I2C-CO2	
Bit	Description/Notes	Mode	Default
	CO2 Concentration in ppm <i>If the Bit CO2Valid is set in the StateFlags, this value contains the current CO2 concentration in ppm. The value range is 400-4000 ppm. During warming up the Bit CO2Valid is set, but the value is -1. If the CO2 sensor has an error the Bit CO2Valid is also set, but then the value is -2.</i>	Read Only	0

Offset 6: Valid (Byte)		ID-I2C-CO2	
Bit	Description/Notes	Mode	Default
	Validity <i>Only if this Byte is 0x01, the CO2 value is valid!</i>	Read Only	0

Offset 7: Reserved (Byte)		ID-I2C-CO2	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	



ID-I2C-IN2

This message is always sent if an AddOn IN2 is connected and the Bit I2CIN2ChangeRequest is set in the ControlFlags. Value changes are transmitted immediately.

This message can also be requested by sending this message with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-I2C-IN2	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x74
	2-3	0x02-0x03	ActorID=0x00
	4-5	0x04	InputState
	6	0x06	Valid
	7	0x07	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: InputState (Word)		ID-I2C-IN2	
Bit	Description/Notes	Mode	Default
15-2	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	
1	InputState for 2. input <i>1=pressed; 0=not pressed</i>	<i>Read Only</i>	
0	InputState for 1. input <i>1=pressed; 0=not pressed</i>	<i>Read Only</i>	

Offset 6: Valid (Byte)		ID-I2C-IN2	
Bit	Description/Notes	Mode	Default
	Validity <i>Only if this Byte has the value 0x01 the InputState is valid!</i>	<i>Read Only</i>	0

Offset 7: Reserved (Byte)		ID-I2C-IN2	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	



ID-I2C-VOC

This message is always sent if an AddOn is connected which provides a VOC value and the Bit I2CVOCChangeRequest is set in the ControlFlags. Value changes are transmitted at a distance of approx. 15 seconds.

This message can also be requested by sending this message with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-I2C-VOC	0	0x00	MessageLength=0x08
	1	0x01	MessageID=0x75
	2-3	0x02-0x03	ActorID=0x00
	4-5	0x04	VOC
	6	0x06	Valid
	7	0x07	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: VOC (Integer)		ID-I2C-VOC	
Bit	Description/Notes	Mode	Default
	VOC Concentration in ppm <i>If the Bit VOCValid is set in the StateFlags, this value contains the current VOC concentration in ppm. The value range is 125-600 ppm. During warming up the Bit VOCValid is set, but the value is -1. If the VOC sensor has an error the Bit VOCValid is also set, but then the value is -2.</i>	Read Only	0

Offset 6: Valid (Byte)		ID-I2C-VOC	
Bit	Description/Notes	Mode	Default
	Validity <i>Only if this Byte is 0x01, the VOC value is valid!</i>	Read Only	0

Offset 7: Reserved (Byte)		ID-I2C-VOC	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	Reserved	



ID-AudioVolume

With this command the volume of the switch is transmitted. This message can also be requested by sending this message with MessageLength=0x04 (without the data).

	Offset		Designation
	Decimal	Hex	
ID-AudioVolume	0	0x00	MessageLength=0x06
	1	0x01	MessageID=0x91
	2-3	0x02-0x03	ActorID=0x00
	4	0x04	Volume=0..100
	5	0x05	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: Volume (Byte)		ID-AudioVolume	
Bit	Description/Notes	Mode	Default
	Volume <i>Defines the volume whereby value 0=Off and value 100 correspond to the max. volume. The volume influences only playback but no records. The volume can be arbitrary changed during a playback. At alarm playbacks this value is ignored and the max. volume is used.</i>	<i>Read/Write</i>	100

Offset 5: Reserved (Byte)		ID-AudioVolume	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	



ID-AudioStop

With this command a running audio record or audio playback can be stopped immediately. Requesting this command is not possible!

	Offset		Designation
	Decimal	Hex	
ID-AudioStop	0	0x00	MessageLength=0x06
	1	0x01	MessageID=0x92
	2-3	0x02-0x03	ActorID=0x00
	4	0x04	StopFlags
	5	0x05	Reserved

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: StopFlags (Byte)		ID-AudioStop	
Bit	Description/Notes	Mode	Default
7-2	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	
1	StopAlarm <i>If this Bit is set a running alarm playback is stopped immediately. This Bit can be combined with the Bit StopNormal.</i>	<i>Write Only</i>	
0	StopNormal <i>If this Bit is set a running recording or playback is stopped immediately. An alarm playback will not be stopped. This Bit can be combined with Bit StopAlarm.</i>	<i>Read/Write</i>	

Offset 5: Reserved (Byte)		ID-AudioStop	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	



ID-AudioPlayLocal

With this command a sound saved on the u::Lux Switch can be played locally. Requesting this command is not possible!

	Offset		Designation
	Decimal	Hex	
ID-AudioPlayLocal	0	0x00	MessageLength=0x10
	1	0x01	MessageID=0x98
	2-3	0x02-0x03	ActorID=0x00
	4	0x04	Volume
	5	0x05	Equalizer
	6	0x06-0x07	PlayFlags
	8-9	0x08-0x09	IncVolumeTime
	10-11	0x0A-0x0B	Repeats
	12-13	0x0C-0x0D	DelayBeforeRepeat
	14-15	0x0E-0x11	SoundIndex

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: Volume (Byte)		ID-AudioPlayLocal	
Bit	Description/Notes	Mode	Default
	<p>Volume</p> <p><i>Defines the volume whereby value 0=Off and value 100 correspond to the max. volume. If the Alarm-Bit is set in the PlayFlags, this value will be ignored and the volume is set to maximum. Otherwise (Alarm-Bit deleted) the volume can also be changed during the playback with the message ID-AudioVolume or directly by the user through the user interface.</i></p>	Read/Write	



Offset 5: Equalizer (Byte)		ID-AudioPlayLocal	
Bit	Description/Notes	Mode	Default
	<p>Equalizer</p> <p>Through this parameter the Equalizer-Mode is selected. The following modes are available: 0=Normal, 1=Rock, 2=Jazz, 3=Classik, 4=Pop, 5=Userdefined (Reserved).</p> <p>The Equalizer Mode can't be changed during a playback.</p>	Read/Write	0

Offset 6: PlayFlags (Word)		ID-AudioPlayLocal	
Bit	Description/Notes	Mode	Default
15-6	<p>Reserved</p> <p>Currently not used, therefore value 0!</p>	Reserved	
5	<p>IncrementVolume</p> <p>If this Bit is set the playback is not started immediately with the adjusted volume, it starts at 0 percent incrementing in a certain time distance until the adjusted volume is reached. The duration, until the max. volume is reached, is indicated by the parameter IncVolumeTime.</p>	Read/Write	0
4	<p>DontChangeVolume</p> <p>If this Bit is set the volume transmitted in this message will not be transfered and the current volume will not be changed. But an alarm playback will be played with full volume.</p>	Read/Write	0
3	<p>NoAudioPage</p> <p>If this Bit is set the audio page (if available) adjusted in the design will NOT be displayed during a playback.</p> <p>If this Bit is cleared, the Audiopage (if available) will be displayed, if the duration of the Sound is at minimum 3 seconds long.</p>	Read/Write	
1-2	<p>Reserved</p> <p>Currently not used, therefore value 0!</p>	Reserved	
0	<p>Alarm</p> <p>If this Bit is set, during playback the volume is held fix on maximum and can't be changed. The playback can only be stopped by setting the Bit StopAlarm at the message ID-AudioStop.</p>	Read/Write	



Offset 8: IncVolumeTime (Word)		ID-AudioPlayLocal	
Bit	Description/Notes	Mode	Default
	Time for increasing volume <i>This parameter indicates time in milliseconds to reach the adjusted volume with increasing volume (Bit 5 of the PlayFlags has to be set).</i>	Read/Write	

Offset 10: Repeats (Word)		ID-AudioPlayLocal	
Bit	Description/Notes	Mode	Default
	Repeats Playbacks <i>This parameter is valid only with local playbacks. It indicates the number of repeats. 0 means no repeat, 1=1 repeat and so on.</i>	Read/Write	

Offset 12: DelayBeforeRepeat (Word)		ID-AudioPlayLocal	
Bit	Description/Notes	Mode	Default
	Time between the repeats <i>This parameter is valid only with local repeats. It indicates the time in milliseconds between the repeats.</i>	Read/Write	

Offset 14: SoundIndex (Word)		ID-AudioPlayLocal	
Bit	Description/Notes	Mode	Default
	Audio-Identification code <i>This parameter contains the index of the audiofile which shall be played. The first audiofile has index 1.</i>	Read/Write	



ID-AudioPlayRemote

With this command an audiostream (audiodata sent via network) can be played. Requesting this command is not possible!

	Offset		Designation
	Decimal	Hex	
ID- AudioPlayRemote	0	0x00	MessageLength=0x20
	1	0x01	MessageID=0x99
	2-3	0x02-0x03	ActorID=0x00
	4	0x04	Volume
	5	0x05	Equalizer
	6-7	0x06-0x07	PlayFlags
	8-9	0x08-0x09	IncVolumeTime
	10-11	0x0A-0x0B	SequenceID
	12-13	0x0C-0x0D	BytesPerFrame
	14-15	0x0E-0x0F	Reserved 1
	16-19	0x10-0x13	DelayBetweenFrames
	20-23	0x14-0x17	IPAddress
	24-29	0x18-0x1D	AMSNetID
30-31	0x1E-0x1F	Reserved 2	

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: Volume (Byte)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Volume <i>Defines the volume whereby value 0=off and value 100 corresponds to the max. volume. If the Alarm-Bit is set in the PlayFlags, this value is ignored and the volume is set to the maximum. Otherwise (Alarm-Bit deleted) also during the playback the volume can be changed with the message ID-AudioVolume or directly by the user through the user interface.</i>	Read/Write	



Offset 5: Equalizer (Byte)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	<p>Equalizer</p> <p><i>Through this parameter the Equalizer-Mode is selected. The following modes are available: 0=Normal, 1=Rock, 2=Jazz, 3=Classik, 4=Pop, 5=userdefined (Reserved).</i></p> <p><i>The Equalizer Mode can't be changed during the playback!</i></p>	Read/Write	

Offset 6: PlayFlags (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
15-6	<p>Reserved</p> <p><i>Currently not used, therefore value 0!</i></p>	Reserved	
5	<p>IncrementVolume</p> <p><i>If this Bit is set, the playback will not be played immediately with the adjusted volume but starting at 0 percent incrementing with a certain time distance until the adjusted volume is reached. The duration to reach the max. volume is indicated through the parameter IncVolumeTime.</i></p>	Read/Write	
4	<p>DontChangeVolume</p> <p><i>If this Bit is set the volume transmitted in this message will not be transferred and the current volume will not be changed. But an alarm playback will be played with full volume.</i></p>	Read/Write	
3	<p>NoAudioPage</p> <p><i>If this Bit is set the audio page (if available) adjusted in the design will NOT be displayed during a playback.</i></p> <p><i>If this Bit is cleared, the Audiopage (if available) will be displayed, if the duration of the Sound is at minimum 3 seconds long.</i></p>	Read/Write	
2	<p>Reserved</p> <p><i>Currently not used, therefore value 0!</i></p>	Reserved	
1	<p>IgnoreAMSNetID</p> <p><i>If this Bit is set during playback the verification of compliance of the AMSNetID is deactivated.</i></p>	Read/Write	



Offset 6: PlayFlags (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
0	Alarm <i>If this Bit is set, during playback the volume is held fix on maximum and can't be changed. The playback can only be stopped by setting the Bit StopAlarm at the message ID-AudioStop.</i>	Read/Write	

Offset 8: IncVolumeTime (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Time for increasing volume <i>This parameter indicates time in milliseconds to reach the adjusted volume with increasing volume (Bit 5 of the PlayFlags has to be set).</i>	Read/Write	

Offset 10: SequenceID (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Audio-Identification code <i>This parameter is compared at reception with the sent SequenceID and the playback works only at compliance.</i>	Read/Write	

Offset 12: BytesPerFrame (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Bytes per Ethernet Frame <i>This parameter indicates the number of databytes per Ethernet Frame. This parameter must currently always have the value 882!</i>	Read/Write	882

Offset 14: Reserved (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>		



Offset 16: DelayBetweenFrames (Long)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Time between every Ethernet Frame <i>Indicates the time between every Ethernet Frame in microseconds. This parameter is not used yet, but must currently always have the value 20000; this corresponds to 20 milliseconds.</i>	Read/Write	20000

Offset 20: IPAddress (Long)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	IP Address to receive Audio-Data <i>This parameter contains the IP-address to which should be listened. If the IP-address (255.255.255.255) is used, every audio stream is accepted but the SequenceIDs have to correlate!</i>	Read/Write	

Offset 24: AMSNetID (6 Bytes)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	AMSNetID Address to receive Audio-Data <i>This parameter contains the AMSNetID to which should be listened. The AMSNetID can also be ignored if the corresponding Bit is set in the PlayFlags.</i>	Read/Write	

Offset 30: Reserved (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>		



ID-AudioRecord

Serves for recording and streaming, requesting is not possible!

	Offset		Designation
	Decimal	Hex	
ID-AudioRecord	0	0x00	MessageLength=0x20
	1	0x01	MessageID=0x9A
	2-3	0x02	ActorID=0x00
	4-5	0x04-0x05	RecordFlags
	6-7	0x06-0x07	SequenceID
	8-9	0x08-0x09	BytesPerFrame
	10-11	0x0A-0x0B	Reserved
	12-15	0x0C-0x0F	DelayBetweenFrames
	16-19	0x10-0x13	MicrophoneSecurityID
	20-23	0x14-0x17	IPAddress
	24-29	0x18-0x1D	AMSNetID
	30-31	0x1E-0x1F	Reserved 2

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: RecordFlags (Word)		ID-AudioRecord	
Bit	Description/Notes	Mode	Default
15-8	Reserved <i>Currently not used, therefore value 0!</i>		
7	RecordFilter <i>This function is not implemented yet!</i>	<i>Read/Write</i>	0
6	MicrophoneHighSensitivity <i>If this Bit is set the highest sensitivity at the microphone is adjusted. Otherwise the microphone works with normal sensitivity.</i>	<i>Read/Write</i>	0
5-0	Reserved <i>Currently not used, therefore value 0!</i>		



Offset 6: SequenceID (Word)		ID-AudioRecord	
Bit	Description/Notes	Mode	Default
	Audio-Identification code <i>This parameter serves to identify the audiostream. Only if the transmitter- and receiver Sequence-ID are corresponding the audiostream is played.</i>	Read/Write	

Offset 8: BytesPerFrame (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Bytes per Ethernet Frame <i>This parameter indicates the number of data bytes per Ethernet Frame. This parameter must currently have the value 882!</i>	Read/Write	882

Offset 10: Reserved (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>		

Offset 12: DelayBetweenFrames (Long)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Time between every Ethernet Frame <i>Indicates the time between every Ethernet Frame in microseconds. This parameter must currently always have the value 20000; this corresponds to 20 milliseconds.</i>	Read/Write	20000

Offset 16: MicrophoneSecurityID (Long)		ID-AudioRecord	
Bit	Description/Notes	Mode	Default
	Microphone Security-ID <i>This parameter has to correspond to the microphoneSecurityID adjusted in the project; otherwise a record can't be started. The parameter serves to avoid misuse of the microphone.</i>	Read/Write	

u::Lux Message Protocol (UMP)



Offset 20: IP (Long)		ID-AudioRecord	
Bit	Description/Notes	Mode	Default
	IP Address for sending Audio-Data <i>This parameter contains the IP-Address where the audio data shall be sent to. If the IP-Address 0xFFFFFFFF (255.255.255.255) is used, the audio data will be sent to all participants (Broadcast).</i>	Read/Write	

Offset 24: AMSNetID (6 Bytes)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	AMSNetID Address for sending Audio-Data <i>This parameter contains the AMSNetID which shall be used in the Header of the Audio packet.</i>	Read/Write	

Offset 30: Reserved (Word)		ID-AudioPlayRemote	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>		



ID-VideoState

With this command the video output size is requested by sending this message with MessageLength=0x04 (without the bounds). This message will also be sent if the Flag VideoActive changes, which means if a video is started or stopped. A running video will be stopped automatically by ID-VideoStop, a firmware or project update or by changing the display page (e.g. calling up a sub menu)!

	Offset		Designation
	Decimal	Hex	
ID-VideoState	0	0x00	MessageLength=0x10
	1	0x01	MessageID=0xA1
	2-3	0x02-0x03	ActorID=0x00
	4-7	0x04-0x07	VideoStateFlags
	8-9	0x08-0x09	Bounds.Left
	10-11	0x0A-0x0B	Bounds.Top
	12-13	0x0C-0x0D	Bounds.Right
	14-15	0x0E-0x0F	Bounds.Bottom

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure. *Bounds* contains the coordinates within which the video is drawn. These coordinates have to be taken into account at the transmission of the videostream.

Offset 4: VideoStateFlags (Long)		ID-VideoState	
Bit	Description/Notes	Mode	Default
31-30	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	
1	VideoPossible <i>1=playing a video is possible</i> <i>If this Bit is set playing a video is possible. The corresponding output position and size is described by Bounds.</i>	<i>Read Only</i>	0
0	VideoActive <i>1=A video is currently played</i> <i>This Bit is set through the command ID-VideoStart and it will be deleted by the command ID-VideoStop.</i>	<i>Read/Write</i>	0



Offset 8: Bounds.Left (Integer)		ID-VideoState	
Bit	Description/Notes	Mode	Default
	Bounds.Left <i>Contains the X-coordinate of the left upper corner.</i>	<i>Read Only</i>	0

Offset 10: Bounds.Top (Integer)		ID-VideoState	
Bit	Description/Notes	Mode	Default
	Bounds.Top <i>Contains the Y-coordinate of the left upper corner.</i>	<i>Read Only</i>	0

Offset 12: Bounds.Right (Integer)		ID-VideoState	
Bit	Description/Notes	Mode	Default
	Bounds.Right <i>Contains the X-coordinate of the right lower corner.</i>	<i>Read Only</i>	0

Offset 14: Bounds.Bottom (Integer)		ID-VideoState	
Bit	Description/Notes	Mode	Default
	Bounds.Bottom <i>Contains the Y-coordinate of the right lower corner.</i>	<i>Read Only</i>	0



ID-VideoStart

With this command a video transmission is started. Requesting this message is not possible. Please use for this purpose the command ID-VideoState!

	Offset		Designation
	Decimal	Hex	
ID-VideoStart	0	0x00	MessageLength=0x0C
	1	0x01	MessageID=0xA2
	2-3	0x02-0x03	ActorID=0x00
	4-7	0x04-0x07	StartFlags
	8-11	0x08-0x0B	SequenceID

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure. The command is only executed if the SequenceID is larger than 0, currently no video is played and if playing a video is possible!

Offset 4: StartFlags (Long)		ID-VideoStart	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	

Offset 8: SequenceID (Long)		ID-VideoStart	
Bit	Description/Notes	Mode	Default
	Video-Identification code <i>This parameter determines the SequenceID of the Videostream. The SequenceID of the streamdata has to correspond to this SequenceID, and then videodata can be received! The SequenceID must not be 0!</i>	<i>Write Only</i>	



ID-VideoStop

With this command a videotransmission is stopped. Requesting this message is not possible. Please use the command ID-VideoState!

	Offset		Designation
	Decimal	Hex	
ID-VideoStop	0	0x00	MessageLength=0x06
	1	0x01	MessageID=0xA3
	2-3	0x02-0x03	ActorID=0x00
	4-7	0x04-0x07	StopFlags
	8-11	0x08-0x0B	SequenceID

For the description of MessageLength, MessageID and ActorID please refer to chapter Message structure.

Offset 4: StopFlags (Long)		ID-VideoStop	
Bit	Description/Notes	Mode	Default
	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	

Offset 8: SequenceID (Long)		ID-VideoStop	
Bit	Description/Notes	Mode	Default
	Video-Identification code <i>This SequenceID has to correspond to the SequenceID, which passed during startup, otherwise the command is ignored!</i>	<i>Write Only</i>	



XX-VideoStream

With this command video data are sent to the switch.

Important: In the descriptor this command has not FrameId 0x8601 but 0x8602 and it is only one message with a special format and not individual aligned messages!

	Offset		Designation
	Decimal	Hex	
XX-VideoStream	0-3	0x00-0x03	StreamFlags
	4-7	0x04-0x07	SequenceID
	8-9	0x08-0x09	StartLine
	10-11	0x0A-0x0B	LineCount
	12-13	0x0C-0x0D	StartColumn
	14-15	0x0E-0x0F	ColumnCount
	16-xx	0x10-xx	Videodata

Offset 0: StreamFlags (Long)		XX-VideoStream	
Bit	Description/Notes	Mode	Default
31-1	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	
1	DirectDraw <i>If this bit is set, all incoming data is directly written to the display. No Designelements (like Texts, Images and Imageanimations) will be drawn. Even if there are no Designelements in the Design, drawing will be faster if this bit is set.</i>	Write Only	
0	Acknowledge <i>If this Bit is activated the participant sends an Acknowledgepacket back to the sender as soon as the data have been successfully processed and the next packet can be sent. The Acknowledgepacket consists only of the descriptor (as it was sent to the u::Lux Switch) and is therefore only 16 Bytes large.</i>	Write Only	



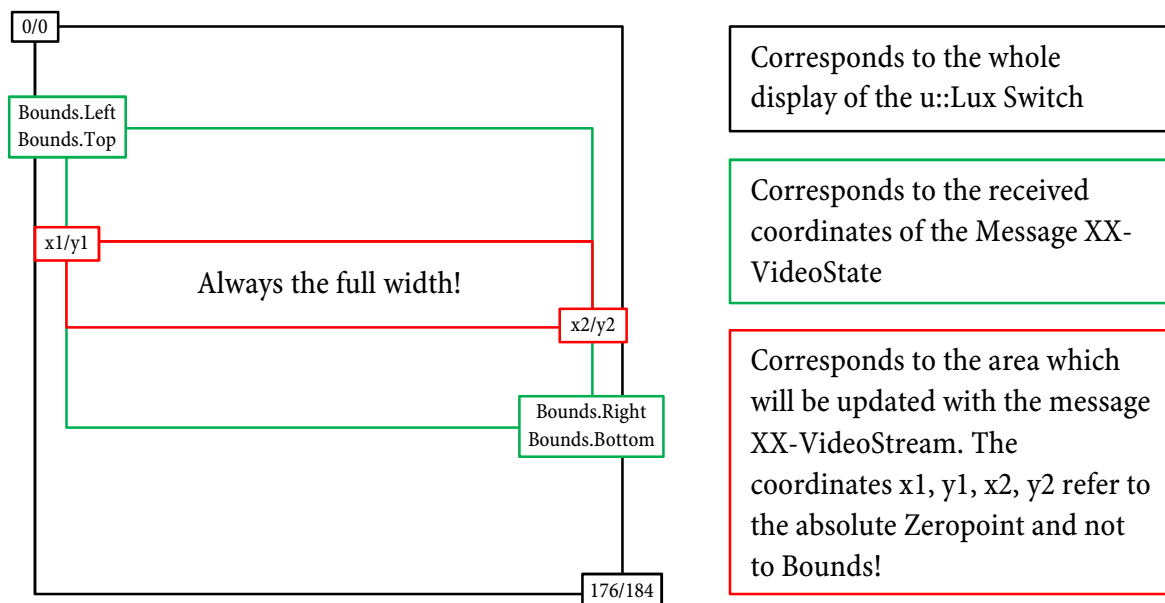
Offset 4: SequenceID (Long)		XX-VideoStream	
Bit	Description/Notes	Mode	Default
	Video-Identification code <i>This SequenceID has to correspond to the SequenceID, which passed during startup, otherwise the command is ignored!</i>	Write Only	

Offset 8: StartLine (Word)		XX-VideoStream	
Bit	Description/Notes	Mode	Default
	StartLine <i>Contains the start line of the video data in this frame. This ist the y1 coordinate.</i>	Write Only	

Offset 10: LineCount (Word)		XX-VideoStream	
Bit	Description/Notes	Mode	Default
	Linecount <i>Contains the number of lines of the video data in this frame. This is the value of y2-y1</i>	Write Only	

x1 is not transmittend and has the value of Bounds.Left, same for x2 which has the value Bounds.Right. Bounds was received with the Message XX-VideoState.

Diagram of the grafic areas:





Offset 12: Videodata (Words)		XX-VideoStream	
Bit	Description/Notes	Mode	Default
	<p>Videodata</p> <p><i>Contains a list of Words with the corresponding pixels in the format RGB 565. The number of pixels corresponds to the number of contained lines multiplied with the number of pixels per line (Bounds.Right-Bounds.Left) multiplied with 2. A maximum of approx. 704 pixels (1408 Bytes) can be contained in one message! Per packet there must always be contained a whole number of ENTIRE lines!</i></p>	Write Only	

Please note also the Chapter Video transmission.

Detail Description

The number of pixels has to be calculated with the coordinates which are determined with the ID-VideoState. For every line Bounds.Right-Bounds.Left Pixel are required, that means (Bounds.Right-Bounds.Left)*2 Bytes. A maximum of 1408 Bytes (704 Pixels) can be transmitted in one Frame, for this reason the number of the lines being transmitted is calculated as follows: rounding off(704/number of Pixel per line)

Example

If ID-VideoState contains e.g. the coordinates Left=10, Up=20, Right=80, Down=65, so (10/20 – 80/70), several XX-VideoStream messages have to be sent in succession to fill out the whole space. Per line (80-10) = 70 pixels are contained. Thus per frame $704/70 = (\text{rounding off}) 10$ lines can be transmitted.

At first the first line block (StartLine=20) is transmitted, then the second (StartLine=30) and the others until StartLine=60. After this only 5 lines are required, thus only $5*70$ pixels are packed into the last frame.

With a fullscreen video the Bounds are currently at (0/0) – (176/184), so a total of $176*184=32384$ Pixels. In the future this could be change maybe because of the use of other displays.



XX-VideoStreamEx

With this command video data are sent to the switch. Instead of the message XX-VideoStream the data is not transmitted with full width, but it can be used a own rectangle area within the videoarea.

Important: In the descriptor this command has not FrameId 0x8601 but 0x8603 and it is only one message with a special format and not individual aligned messages!

	Offset		Designation
	Decimal	Hex	
XX-VideoStreamEx	0-3	0x00-0x03	StreamFlags
	4-7	0x04-0x07	SequenceID
	8-9	0x08-0x09	StartLine
	10-11	0x0A-0x0B	LineCount
	16-xx	0x10-xx	Videodata

Offset 0: StreamFlags (Long)		XX-VideoStreamEx	
Bit	Description/Notes	Mode	Default
31-1	Reserved <i>Currently not used, therefore value 0!</i>	<i>Reserved</i>	
1	DirectDraw <i>If this bit is set, all incomming data is directly written to the display. No Designelements (like Texts, Images and Imageanimations) will be drawn. Even if there are no Designelements in the Design, drawing will be faster if this bit is set.</i>	<i>Write Only</i>	
0	Acknowledge <i>If this Bit is activated the participant sends an Acknowledgepacket back to the sender as soon as the data have been successfully processed and the next packet can be sent. The Acknowledgepacket consists only of the descriptor (as it was sent to the u::Lux Switch) and is therefore only 16 Bytes large.</i>	<i>Write Only</i>	



Offset 4: SequenceID (Long)		XX-VideoStreamEx	
Bit	Description/Notes	Mode	Default
	Video-Identification code <i>This SequenceID has to correspond to the SequenceID, which passed during startup, otherwise the command is ignored!</i>	Write Only	

Offset 8: StartLine (Word)		XX-VideoStreamEx	
Bit	Description/Notes	Mode	Default
	StartLine <i>Contains the start line of the video data in this frame. This is the y1 coordinate.</i>	Write Only	

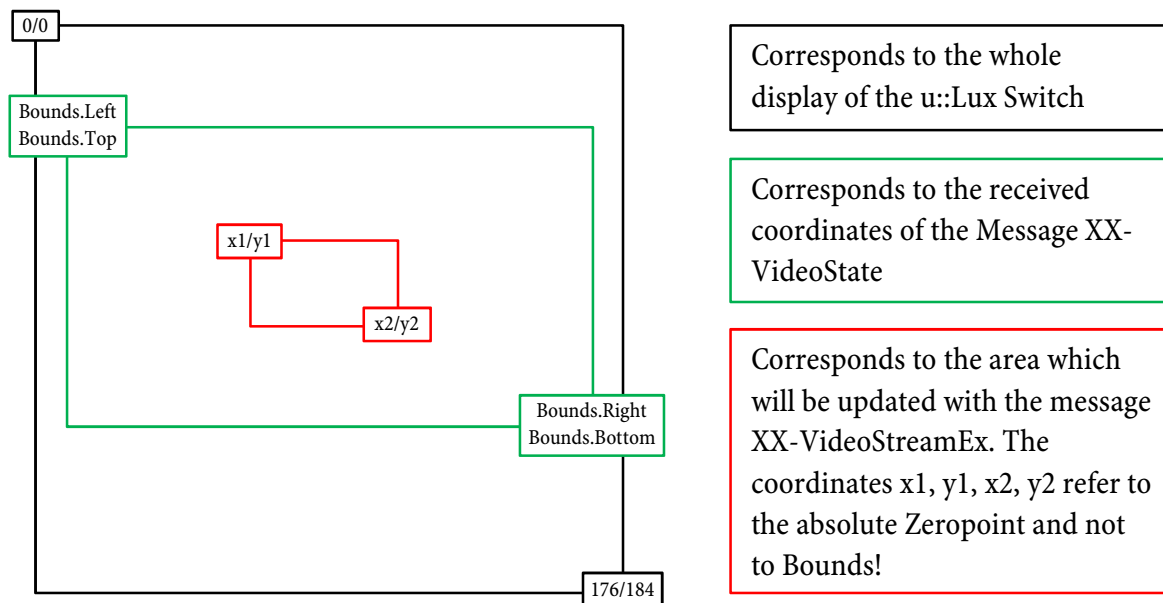
Offset 10: LineCount (Word)		XX-VideoStreamEx	
Bit	Description/Notes	Mode	Default
	Linecount <i>Contains the number of lines of the video data in this frame. This is the value of y2-y1</i>	Write Only	

Offset 12: StartColumn (Word)		XX-VideoStreamEx	
Bit	Description/Notes	Mode	Default
	Startrow <i>Contains the start row of the video data in this frame. This is the x1 coordinate.</i>	Write Only	

Offset 14: ColumnCount (Word)		XX-VideoStreamEx	
Bit	Description/Notes	Mode	Default
	Columncount <i>Contains the number of rows of the video data in this frame. This is the value of x2-x1</i>	Write Only	



Diagram of the graphic areas:



Offset 16: Videodata (Words)		XX-VideoStreamEx	
Bit	Description/Notes	Mode	Default
	<p>Videodata</p> <p><i>Contains a list of Words with the corresponding pixels in the format RGB 565. The number of pixels corresponds to the number of contained lines multiplied with the number of pixels per line (Bounds.Right-Bounds.Left) multiplied with 2. A maximum of approx. 702 pixels (1404 Bytes) can be contained in one message! Per packet there must always be contained a large number of ENTIRE lines!</i></p>	Write Only	

u::Lux Message Protocol (UMP)



Please note also the Chapter Video transmission.

Detail Description

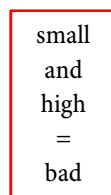
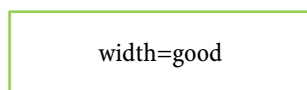
The number of pixels is calculated by the number of lines multiplied by the number of rows. The maximum number of pixels in one frame is 702, which is 1404 bytes.

Example

If ID-Videostate returns the coordinates Left=10, Top=20, Right=80 and Bottom=65, so (10/20 – 80/65) and within this the area Left=15, Top=25, Right=35 and Bottom=60 should be refreshed (all numbers refer to the absolute zeropoint of the display), so (Right-Left)*(Bottom-Top) = (35-15)*(60-25) = 20*35 = 700 pixel must be transmitted.

A line contains (35-15) = 20 pixel. So per frame $702/20 =$ (round down!) 35 lines can be transmitted. The number of columns should always be as large as possible, because it is faster to draw lines instead of rows.

The following example will show this!



With a fullscreen video the Bounds are currently at (0/0) – (176/184), so a total of $176*184=32384$ Pixels. In the future this could change maybe because of the use of other displays.



Version Management

Version	Date	Name	Notes
1.00	18.12.2012	AK	Version 1.00
1.01	12.02.2013	NICZ	Basic version
2.01	11.03.2013	AK	Publication
2.02	21.03.2013	AK	Addition of Audio Messages
2.03	25.03.2013	AK	The Flag NoAudioPage was added at Messages ID-AudioPlayLocal and ID-AudioPlayRemote
2.04	23.04.2013	AK	Chapter Message structure exchanged with MessageIDs, several minor corrections
2.05	26.06.2013	AK	ID-AudioPlayLocal SoundIndex value for first Audiofile changed
2.06	29.07.2013	AK	Addition of ID-VideoState, ID-VideoStart, ID-VideoStop
2.07	17.10.2013	AK	Addition of MotionSensor in the StateFlags and MotionSensorChangeRequest in the ControlFlags
2.08	03.03.2014	AK	Addition of ID-Activate command
2.09	04.03.2014	AK	Addition of I2CCO2Valid in the StateFlags, Addition of I2CCO2ChangeRequest in the ControlFlags, Addition of ID-I2C-CO2 message
2.10	20.03.2014	AK	ID-Value and ID-RealValue extended to 4 RealValues
2.11	09.04.2014	AK	Several corrections, Extension up to 64 Actor-IDs per u::Lux Switch
2.12	07.05.2014	AK	Error in the MessageIDs Overview-Table (I2C-MessageIDs changed). At the commands were the correct MessageIDs. Addition of I2CIN2Valid in den StateFlags, Addition of I2CIN2ChangeRequest in den ControlFlags, Addition of ID-I2C-IN2 Message
2.13	13.05.2014	AK	Addition of ID-Text command.
2.20	26.05.2014	AK	At all audio messages the Play- and RecordFlags have been changed from Byte to Word! The message ID-AudioPlayRemote and ID-AudioRecord have been adjusted to the EAP Protocol. Further more the Audioframe (XX-AudioStream) was described.
2.21	31.07.2014	AK	XX-AudioStream 1. Variabel description changed



Version	Date	Name	Notes
2.22	04.08.2014	AK	Correction of the length of ID-AudioPlayRemote and ID-AudioRecord.
2.23	08.08.2014	AK	ID-Text Text usable characters defined.
2.24	01.10.2014	AK	Contact updated
2.25	16.10.2014	AK	Driver Development Sending a Broadcast was changed (SwitchID must be 0).
2.26	22.10.2014	AK	Addition of Messages ID-Lux and ID-Signature.
2.27	13.11.2014	AK	Addition of ID-Text Remove Flag
2.28	21.11.2014	AK	XX_AudioStream Packet description removed from UMP Protocol and inserted in the USP Protocol.
2.29	04.12.2014	AK	Addition of ACK Flag at the StreamFlags of command XX-VideoStream.
2.31	10.07.2015	AK	XX_VideoStreamEx added.
2.32	15.07.2015	AK	DirectDraw Flag on XX_VideoStream and XX_VideoStreamEx added.
2.33	20.08.2015	AK	Addition of I2CVOCValid in the StateFlags, Addition of I2CVOCChangeRequest in the ControlFlags, Addition of ID-I2C- message
2.34	31.03.2016	AK	ID-Lux Informationen updated. Addition of LUXChangeRequest in the ControlFlags Doku on ID-Text (Controlcharacters) updated.
2.35	18.04.2016	AK	NoAudioPage Bit additional information added.
2.36	22.09.2016	AK	ID-VideoStart SequenceID must not be 0 added.
2.37	14.03.2017	AK	ID-Activate Flags description improved
2.38	19.06.2017	AK	Set FrameVersion to 0x0200
2.39	20.02.2018	AK	I2CMotionDetectorValid added (Firmware V1.9D) AMSNetId adjusted from Long to 6 Bytes, DelayBetweenFrames Description expanded.
2.40	16.04.2018	AK	MotionSensor in MotionDetector renamed.
2.41	07.07.2020	AK	PageChangeRequest Bit in ControlFlags and ID-PageIndex Command Documentation for Firmware V2.03 extended.
2.42	27.10.2020	AK	ID-Signature Message removed. MessageLength on ID-VideoStop corrected.
2.43	26.11.2020	AK	Minimal text modifications
2.44	13.04.2021	AK	Defaultvalue for FrameVersion removed.