DALI-2 IoT4

Modbus Manual

Central Control Device

Central Control Device for 4 DALI-lines

DALI-2 IoT4 Central Control Device

1. ModBus TCP Access

1.1 General

Modbus TCP/IP is a type of the serial modbus protocol for TCP/IP networks using port 502. The Modbus standard provides several functions for data exchange. The DALI-2 IoT4 functions as a Modbus **Server** i.e. it only responds to Modbus Register Reads/Writes initiated by a Modbus Client, and cannot initiate any Modbus transaction on the network by itself.

1.2 Frame Structure

A Modbus TCP/IP Frame has a header called “MBAP Header” and consists of the following 7 byte:

Name	Length (bytes)	Function
Transaction identifier	2	For synchronization between messages of server & client
Protocol identifier	2	Zero for Modbus/TCP
Length field	2	Number of remaining bytes in this frame
Unit identifier	1	Slave address (255 if not used) – used for line selection
Function Code	1	Modbus Function Codes as described below
Data	n	Number of bytes as needed

1.2.1 Unit Identifier: DALI Bus Line Selection:

In the DALI-2 IoT4 the “Unit identifier” is used to select which bus is addressed for certain registers. It’s used in this format:

Bit	7	6	5	4	3	2	1	0
Description	*0*				*Line 3*	*Line 2*	*Line 1*	*Line 0*

Binary selection for bus lines e.g.

sending on line 0: UID = 0x01
sending on line 0 and line 1: UID = 0x03

Note: in some central control units the UID is referred to simply as “id”.

This is i.e. used for sending DALI commands. With the help of the unit identifier DALI commands can be sent either to one or multiple DALI-lines.

1.3 ModBus Commands

Supported Modbus functions:

Function Name	Function Code	Description
Read Multiple Holding Registers	03	Read Data Blocks From Device
Write Multiple Holding Registers	16	Write Data Blocks To Device
Read/Write Holding Registers	23	First Write, then Read from Specific Address, function used to send DALI commands

With the help of the mentioned functions Modbus registers can be accessed and data can be exchanged between a Modbus client (the Control Unit in the System initiating transaction) and the server (DALI-2 IoT 4).

1.4 Frame Examples

Read multiple registers (FC 03):

Request

Byte 0: 03 (Read multiple registers)
Byte 1-2: Register Address (also called „reference number“)
Byte 3-4: Word count (1-125)

Response

Byte 0: 03 (Read multiple registers)
Byte 1: Byte count of response
Remaining Bytes: Register values

Exceptions

Byte 0: FC = 0x83
Byte 1: exception code = Illegal Function (01) or Illegal Data Address (02)

Write multiple registers (FC 16):

Request

Byte 0: 0x10 (Write multiple registers)
Byte 1-2: Register Address (also called „reference number“)
Byte 3-4: Word count (1-100)
Byte 5: Byte count
Remaining Bytes: Register values

Response

Byte 0: 0x10 (Write multiple registers)
Byte 1-2: Register Address (also called „reference number“)
Byte 3-4: Word count

Exceptions

Byte 0: FC = 0x90
Byte 1: exception code = Illegal Function (01) or Illegal Data Address (02)

Read/Write registers (FC 23):

Request

Byte 0: 0x17 (Read/Write registers)

Byte 1-2: READ Register Address (also called „read reference number“)

Byte 3-4: Word count for read (1-125)

Byte 5-6: WRITE Register Address (also called „write reference number“)

Byte 7-8: Word count for write (1-100)

Byte 9: Byte count

Remaining Bytes: Register values

Response

Byte 0: 0x17 (Read/Write registers)

Byte 1: Byte count

Remaining Bytes: Register values

Exceptions

Byte 0: FC = 0x97

Byte 1: exception code = Illegal Function (01) or Illegal Data Address (02)

1.5 ModBus Registers

Register	Name	Length (Word)	Read/Write	Function
1	Polling Configuration	4	RW	Enable/Disable Polling from DALI-2 IoT4
10	Network Configuration	7	R	Read Network Configuration (DHCP or static)
20	System Configuration	32	R	Read System Configuration and Nametag
100	Write DALI Command	6	W	Write DALI Command (100&101 used with FC 23)
101	Read DALI Command	5	R	Read Answer from previously sent Command
6100-7123	Read Sensor Values		R	Read 24bit sensor value (depending on type)
9000-9063	Query Actual Level and Short Address	1	R	Query Actual level and Short Address of given devices
9100-9163	Query Status	1	R	Query DALI Status and Extended Status of given devices

1.6 Register Details

1.6.1 Register 1 – Polling Configuration

If polling is activated, the DALI-2 IoT4 cyclically polls status and actual level of the DALI ballasts. If polling is inactive the status and actual level of the device may be wrong (you will then receive the internal calculated refence value of the ballast which in most cases is correct, but can deviate e.g. in case of communication error or lamp failure).

Polling Configuration			
Byte	Name	Line	Description
0	Config	0	Bit 0: 1=Enable 0=Disable
1	Reserved		Reserved for future use
2	Config	1	Bit 0: 1=Enable 0=Disable
3	Reserved		Reserved for future use
4	Config	2	Bit 0: 1=Enable 0=Disable
5	Reserved		Reserved for future use
6	Config	3	Bit 0: 1=Enable 0=Disable
7	Reserved		Reserved for future use

1.6.2 Register 10 – Network Configuration

With the network configuration the network settings can be read out. The network settings cannot be changed via Modbus, but only via the Restful API available at the *device-IP/docs*

Network Configuration		
Byte	Name	Description
0	DHCP	0x01=DHCP 0x00=Static
1-4	IP Address	i.e. 1=192 2=168 3=0 4=99
5-8	Subnet Mask	i.e. 5=255 6=255 7=255 8=0
9-12	Gateway	i.e. 9=192 10=168 11=0 12=1

1.6.3 Register 20 – System Configuration

With Modbus register 20 info about hardware, firmware, serial number, production data etc. as well as the name of the device (maximum length of 30 characters) can be read out. The name can be change via the Restful API available at the *device-IP/docs/post/info*.

System Configuration		
Byte	Name	Description
0-29	Nametag	Up to 30 characters nametag
30-31	HW Version	Major.Minor
32-35	Serial number	4 Byte serial number
36-39	Article number	4 Byte article number
40	LTDT	Lunatone specific device type
41	Build number	FW Version build number
42	FW Minor	FW Version minor
43	FW Major	FW Version major
44-45	Production	Week,Year
46-63	Info	Device info
62	reserved	reserved
63	reserved	reserved

1.6.4 Register 100 – Write DALI-Command

For direct access to the DALI-lines Modbus Register 100 and 101 are used.

Write Dali Command			**Example RECALL MAX A0**	
byte	**Name**	**Description**	**Value**	**Meaning**
0	**CmdByte**	Command Byte = 0x12 always	0x12	Required
1	**Sequence number**	Command Sequence number (will be sent back)	0x01	sequence nr. set to 1
2	**Control**	Command Control byte Bit 7: unused, set to 0 Bit 6: if set no data is sent out on the DALI line (used to test connection status) Bit 5: send twice, cmd will be sent twice on DALI-line (required for some DALI commands) Bit 4: Send DTR before DALI command (only with DALI-16 and eDALI commands) Bit 3: Send DALI Device Type before DALI command (only with DALI-16 and eDALI commands) Bit 2: Send “Set Actual Level to DTR” before DALI command (only with DALI-16 commands, not to be used with DALI-16 DAP commands) Bit 1: unused, set to 0 Bit 0: unused, set to 0	0x00	do not send twice, no DTR set, no device type before command
3	**Mode**	Command Mode Byte **Value** 0,1: not used **2**: send DALI answer (8Bit, DATA_LO) **3**: send DALI (16 Bit, DATA_MID, DATA_LO) **4**: send eDALI (25Bit, DATA_HI, DATA_MID, DATA_LO) 5: reserved **6**: send 3Byte DALI (24Bit, DATA_HI, DATA_MID, DATA_LO) 7: reserved 8: reserved 12: reserved	0x03	Send DALI 16bit command

4	**Reserved**		0x00	reserved
5	**DALI High**	Highest Dali Byte (DATA_HI)	0x00	HIGH byte not needed for 16bit DALI
6	**DALI Mid**	Mid Dali Byte (DATA_MI)	0x01	DALI Frame containing info: A0 and Command "Recall MAX"
7	**DALI Low**	Low Dali Byte (DATA_LO)	0x05	
8	**DTR**	Value to be set to DTR	0x00	No DTR set
9	**Priority**	Priority for DALI command (not applicable if control bit 2 or control bit 4 are set)	0x00	No priority set
10	**Device type**	Device type to be sent	0x00	No device type to declare

Note for writing *Registers 100 – Sending DALI commands*: it is recommended to use the Function Code FC 23 "Read/Write Holding Registers". Herewith the registers are written and read and it can be captured if the DALI command was successfully sent on the DALI bus. The answer will contain a status information if the command could not be sent (e.g. bus short) or the DALI answer to the according command.

For some example DALI Frames or looking up the necessary DALI frames see sample selection:
https://www.lunatone.com/wp-content/uploads/2024/12/IoT4-Access-via-Modbus-Sample-Collection.pdf

Example*: send "RECALL MAX to A0 on Line 0":*

UID = 1 to reach line 0,

FC 23: write/read multiple registers - number registers write: 6, start register 100, (base address = 0)

byte	0	1	2	3	4	5	6	7	8	9	10
name	*cmd*	*#*	*control*	*mode*	*res*	*high*	*mid*	*low*	*DTR*	*Prio.*	*DT*
value (hex)	0x12	0x00	0x00	0x03	0x00	0x00	0x01	0x05	0x00	0x00	0x00

number registers read: 5 → received answer interpretation see next section

1.6.5 Register 101 – Read DALI Answer

Read DALI Command			Example received Answer From section 1.2 Write/Read Recall Max A0	
Byte	Name	Description	Value	Meaning
0	Cmd Byte	Command Byte = 0x12 always	0x12	mandatory
1	Status	Command Status byte: High nibble: reserved (value is always 7) Low nibble: status stated by value below value 1: DALI answer ="NO" value 2: DALI 8bit data value 7: Error/Info, if set: Collision / DALI answer = "Yes": DATA_LO=1; DALI-line short circuit: DATA_LO=2;	0x81	Low nibble: value =1 Answer is "NO" (as no answers are received to control commands)
2	Reserved		0x00	
3	Reserved		0x00	
4	Reserved		0x00	
5	Answer	DALI_LO (answer to previous command)	0x00	
6	Reserved		0x00	
7	Sequence number	Command sequence number same as previously sent	0x01	answer to command sequence nr 1
8	Reserved		0x12	
9	Reserved		0x01	

Note: Answers can only be read from single lines (on reading Answers with UID set to multiple lines, the answer from the lowest line is returned. On reading answers from zones, the answer from line 0 will be returned)

Example received 8bit answer to sending DALI command :

Byte:	0	1	2	3	4	5	6	7	8	9
Hex:	0x12	0x71	0x00	0x00	0x00	0x00	0x00	0x01	0x00	0x00

Example 2: DALI bus-status can also be registered via sending a DALI query e.g. “query lamp failure” broadcast (register 100) and analysing the answer in register 101:

- byte 1: 0x7**1** and byte 5: 0x**00** - no answer on the DALI bus: no device has a failure
- byte 1: 0x7**2** and byte 5: DALI data – answer of a single device, meaning there is a single device error, with the failure status as described for register 9100
- byte 1: 0x7**7** and byte 5: 0x0**1** – there is a collision on the DALI bus: multiple devices answer: there are multiple lamp failures
- byte 1: 0x7**7** and byte5: 0x0**2** - line short, there is a short circuit or the DALI bus line is not powered

1.6.6 Registers 6100 to 7123 – Read Sensor Values

Query the most recent sensor values of specified sensors. Addressing of sensors via the Restful API and configuration for sensor polling is required. On the webpage *device-IP/docs/post/scan* an addressing of each line is necessary to register the sensors in the DALI-2 IoT4. Each DALI-2 sensor instance will be listed separately (supported sensors: all Lunatone DALI-2 CS, DALI-2 CS-THP, DALI-2 CS THP AQ). Via the endpoint *device-IP/get/sensors* the sensor list and their indices can be read. Via the endpoint *device-IP/post/sensor-values* the sensor values can be refreshed.

The indices given for sensors indicate the register for querying the sensor value: index 0 corresponds to register 6100.

Query most recent Sensor Values		
Byte	Name	Description
Use the sensor index for selection of the register		
0	Sensor value	low byte of sensor value
1	Sensor value	high byte of sensor value

Interpretation of the returned sensor values:

- Motion sensor values:
  - 0: No Movement - Vacant
  - 1: Movement
  - 2: Occupied
  - 4: Still vacant
  - 6: Still occupied
  - 8: Movement active

- Light sensor values, DALI-2 CS and DALI-2 LS [0lux … 2500lux]
  Light Level (lux) = *answer*
- Temperature value DALI-2 CS THP (-AQ) [-20°C … 80°C]
  Temperature (°C) = *answer/10* – 20
- Humidity value DALI-2 CS THP (AQ) [0% … 100%]
  Humidity (%) = *answer*/10
- Pressure value DALI-2 CS THP (AQ) [300hPa ... 1100hPa]
  Pressure (hPa) = *answer*+300
- Air quality index value DALI-2 CS THP AQ - range [0 … 500]
  Air Quality = *answer*

1.6.7 Registers 9000 to 9063 – Query Actual Level and Short Address

Query actual level and short address of DALI control gear. The actual level of up to 64 device can be read with one command only. Addressing of devices via the Restful API and configuration for device polling is required. On the webpage *device-IP/docs/post/scan* an addressing of each line is necessary to register the devices in the DALI-2 IoT4. Via *device-IP/get/devices* the devices addresses and indices can be read. Via the polling settings register 1, or *device-IP/post/device-polling* the polling configuration can be set up

Query Actual Level and Short address (ballasts)		
Byte	Name	Description
Use Unit identifier for Bus selection! Command is usable for up to 64 registers which are equal to DALI addresses. 9000 with 1 word would return level and address from control gear 0. 9000 with 64 words returns actual level and address of all control gears.		
0	Actual level	DALI actual level
1	Short address	DALI short address (255 if unaddressed > no device)

Interpretation of the returned actual level:

The actual level is returned as value between 0-254, where 0 represents 0% and 254 100%, the light level in between in % can be calculated from the “actual level” as such:

$$\text{Light output} = 10^{\frac{actualLevel-1}{\frac{253}{3}}-1}\ \%$$

1.6.8 Registers 9100 to 9163 – Query Status

Query status of DALI control gear. The status of up to 64 device can be read with one command only. The status includes the DALI device status and info about communication. This query only makes sense if automatic polling (ModBus Register 1) is activated (can also be set via Restful API) – otherwise the answer can differ from the current status of the DALI devices.

Query Status (ballasts)		
Byte	Name	Description
Uses Unit identifier for Bus selection! Command is usable like command 9000.		
0	Extended Status	AxxxxxxC(bin) → A .. device is addressed, C .. Communication Error (device is not answering to poll), (x: reserved, 0)
1	DALI Status	DALI device status

Interpretation of the returned DALI status:

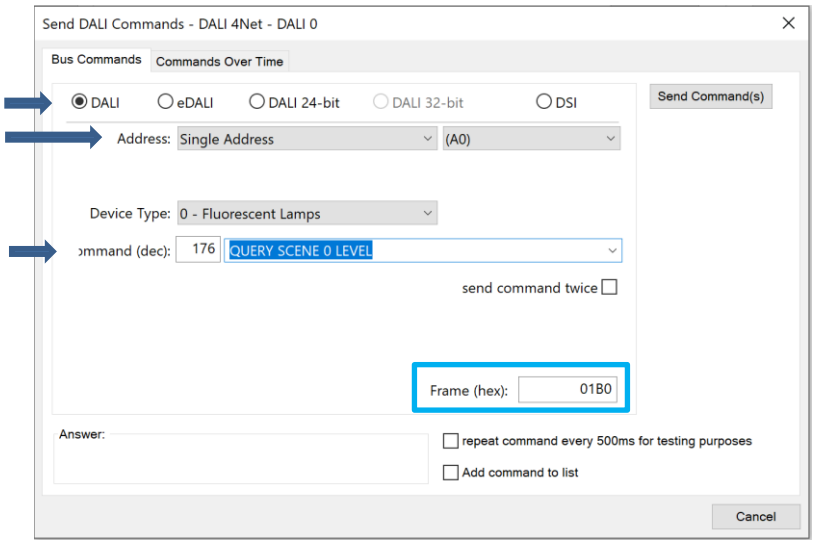
The Returned DALI Status contains following information:

Bit	Description	Value
0	Control Gear Failure	"1" = Yes
1	Lamp Failure	"1" = Yes
2	Lamp On	"1" = Yes
3	Limit Error	"1" = Yes
4	Fade Running	"1" = Yes
5	Reset State	"1" = Yes
6	Short Address is MASK	"1" = Yes
7	Power Cycle Seen	"1" = Yes

2. Examples

2.1 Looking up DALI Frames

Selection of Frame: DALI HI, DALI MID, DALI LO:
For Information on the DALI Commands to send – please refer to the DALI Cockpit DALI Command Tool (Menu DALI Bus > Send DALI Commands…).



By selecting the needed options (type: (DALI, eDALI, DALI-2), address, and command) the correct data can be read from the frame (hex).
For most common DALI Frames see last page for **Table of common DALI Frames**

2.2 Example: Send DALI Control Command

Scene 0 to group 0

Type	Hex Data	Address	Command
DALI16 IAP	81 10	G0	GOTO SCENE 0

Send to IP of the DALI-2 IoT4,
Set UNIT ID 1 for DALI Line 0
Function code FC23: to write/read multiple registers
number of registers to write: 6
number of registers to read: 5
Start from Register: 100
DALI command: DALI Scene 0 - Group 0 (DALI frame: 0x8110)

WORD	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
Hex:	12	01	00	03	00	00	**81**	**10**	00	00	00	00
comment	Command Byte 0x12 always	sequence number (set to 2)	Control set to 0	Mode set to DALI 16 bit	Res.	DALI Hight (unused for DALI 16bit)	**DALI Mid**	**DALI Low**	DTR (not used here)	Priority (not used here)	Device Type (not used here)	Res.

Received 8bit answer to sending DALI command:

Byte:	0	1	2	3	4	5	6	7	8	9
Hex:	0x12	0x71	0x00	0x00	0x00	0x00	0x00	0x01	0x00	0x00

Status: "No Answer" → frame sent on bus successfully , no error

2.3 Example: Query DALI Status

Sending Query Status and reading the response

Type	Hex Data	Address	Command
DALI16 Query	01 90	A0	QUERY STATUS
DALI8 Answer	04		= 4 (0x04)

Send to IP of the DALI-2 IoT4,
Set UNIT ID 2 for DALI Line 1
Function code FC23: to write/read multiple registers -
number of registers to write: 6
number of registers to read: 5
Start from Register: 100
DALI command: DALI Query Status - Address 0 (DALI frame: 0x0190)

WORD	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
Hex:	12	03	00	03	00	00	**01**	**90**	00	00	00	00
comment	command byte 0x12 always	sequence number (set to 3)	control set to 0	mode set to DALI 16 bit	res.	DALI High (unused for DALI 16bit)	**DALI Mid**	**DALI Low**	DTR (not used here)	Priority (not used here)	Device Type (not used here)	res.

Response:

Byte:	0	1	2	3	4	5	6	7	8	9
Hex:	0x12	0x72	0x00	0x00	0x00	0x00	0x04	0x03	0x12	0x03

Interpretation: Status 0x72, low nibble value 2: DALI 8 bit Data
→DALI 8bit Data : 0x04 according to DALI Status info below: no failure/error, lamp is on, no fade running, addressed (address is not mask),…

DALI Status Information

Bit	Description	Value	Answer 0x04
0	Control Gear Failure	"1" = Yes	0
1	Lamp Failure	"1" = Yes	0
2	Lamp On	"1" = Yes	1
3	Limit Error	"1" = Yes	0
4	Fade Running	"1" = Yes	0
5	Reset State	"1" = Yes	0
6	Short Address is MASK	"1" = Yes	0
7	Power Cycle Seen	"1" = Yes	0

2.4 Example: Send RGB Colour Commands

Sending a DALI DT8 colour command requires several commands: sending the value for each colour and then activating the set colour.

Type	Hex Data	Address	Command
DALI16 Special	A3 00	*	DATA TRANSFER REGISTER= 0 (0x00)
DALI16 Special	C3 00	*	DATA TRANSFER REGISTER 1= 0 (0x00)
DALI16 Special	C5 FE	*	DATA TRANSFER REGISTER 2= 254 (0xFE)
DALI16 Special	C1 08	*	ENABLE DEVICE TYPE 8
DALI16 AppExt D8	09 EB	A4	SET TEMPORARY RGB DIMLEVEL
DALI16 Special	C1 08	*	ENABLE DEVICE TYPE 8
DALI16 AppExt D8	09 E2	A4	ACTIVATE

Since 3 DTRs (0,1 and 2) need to be written it cannot be included into one registry write.
(In other applications (e.g. set fade time), where only 1 DTR is set, the control byte (byte 2) and DTR data byte (byte 8) could be used to send DTR commands previous to other commands, with one registry write)
In this example the answers between writing are not mentioned, to make sure all commands are sent to the DALI bus and to capture errors the answers should be evaluated before sending the next command.

Writing register DTR like DALI commands with DALI DTR command – for blue (A300 (red=0), C300 (green=0), C5FE (blue=254))

WORD	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
comment	Command Byte 0x12 always	sequence number (set to 1)	Control set to 0	Mode set to DALI 16 bit	Res.	DALI Hight (unused for DALI 16bit)	**DALI Mid**	**DALI Low**	DTR (not used here)	Priority (not used here)	Device Type (not used here)	Res.

set value red: DTR 0 to 0x00

Hex:	12	01	00	03	00	00	**A3**	**00**	00	00	00	00

set value green: DTR1 to 0x00

Hex:	12	02	00	03	00	00	**C3**	**00**	00	00	00	00

set value blue: DTR2 to 0xFE

Hex:	12	03	00	03	00	00	**C5**	**FE**	00	00	00	00

Writing register Device Type 8 + DALI Data = 0x09EB for Set Temporary Dim Level Address 4

WORD	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
Hex:	12	04	08	03	00	00	**09**	**EB**	00	00	08	00
comment	Command Byte 0x12 always	sequence number (set to 4)	Control use to send Device Type	Mode set to DALI 16 bit	Res.	DALI Hight (unused for DALI 16bit)	**DALI Mid**	**DALI Low**	DTR (not used here)	Priority (not used here)	Device Type = 8	Res.

Writing register Device Type 8 + DALI Data = 0x09E2 for Activate to Address 4

WORD	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
Hex:	12	05	08	03	00	00	**09**	**E2**	00	00	08	00
comment	Command Byte 0x12 always	sequence number (set to 5)	Control use to send Device Type	Mode set to DALI 16 bit	Res.	DALI Hight (unused for DALI 16bit)	**DALI Mid**	**DALI Low**	DTR (not used here)	Priority (not used here)	Device Type = 8	Res.

2.5 Example Read Memory Location – Memory Bank 0

Reading a Memory Bank requires several commands – settings the memory bank number to DTR1, setting the entry to read of the memory bank to DTR, sending the query: read memory bank location to address A6.

Type	Hex Data	Address	Command
DALI16 Special	C3 00	*	DATA TRANSFER REGISTER 1= 0 (0x00)
DALI16 Special	A3 00	*	DATA TRANSFER REGISTER= 0 (0x00)
DALI16 Query	0D C5	A6	READ MEMORY LOCATION
DALI8 Answer	11		= 17 (0x11)
DALI16 Query	0D C5	A6	READ MEMORY LOCATION
DALI8 Answer	D3		= 211 (0xD3)

1st send command: DTR1 set to memory bank 0 → DTR1 = 0
2nd send command: DTR to first entry → DTR = 0
3rd send "Read Memory location": 0x05C5
4th continue reading following memory bank entries with "Read Memory location"

DTR command can be included with the following ModBus command:
setting bit 4 of the control byte to send DTR before DALI command
setting byte 8 to the DTR data 0x00

Writing register DTR1 = 0 to set to memory bank 0

WORD	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
Hex:	12	01	00	03	00	00	**C3**	**00**	00	00	00	00
comment	Command Byte 0x12 always	sequence number (set to 1)	Control set to 0	Mode set to DALI 16 bit	Res.	DALI Hight (unused for DALI 16bit)	**DALI Mid**	**DALI Low**	DTR (not used here)	Priority (not used here)	Device Type (not used here)	Res.

Writing register DTR = 0 + Read memory Location from address A6 (= 0x0DC5)

Hex:	12	02	10	03	00	00	**0D**	**C5**	**00**	00	00	00
comment	Command Byte 0x12 always	sequence number (set to 2)	Control bit 4 set to use DTR	Mode set to DALI 16 bit	Res.	DALI Hight (unused for DALI 16bit)	**DALI Mid**	**DALI Low**	DTR set to 00	Priority (not used here)	Device Type (not used here)	Res.

Response:	0x1272 0x0000 0x**0011** 0x0002 0x0000

Sequential entries of the memory bank can be read by continuing "read memory location" queries – entry number is automatically incremented with each read (DALI standard)

Read Next Entry:

Send	0x1203 0x0003 0x00 **0x0DC5** 0x00 0x00	Read Memory Location A6
Response:	0x1272 0x0000 0x**00D3** 0x0003 0x0000	

2.6 Example Query DALI-2 Instance

To query a DALI instance the DALI-2 device address and the instance number need to be known. To query the values the commands “query input” and “query input value latch” (in case the value is longer than 1byte) are used.

Query Sensor Values/ Pushbuttons via Instances:

1. Query Input Value
2. Read Answer
3. Query Input Value latch (in case of light sensor)
4. Read Answer
5. Evaluate Answer

The evaluation of the response is dependent on the queried instance.

DALI command: DALI Query Input Value- Address 0^2 Instance nr. 0 (DALI frame: 0x01018C)

WORD	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
Hex:	12	05	00	06	00	**01**	**01**	**8C**	00	00	00	00
comment	command byte 0x12 always	sequence number (set to 5)	control set to 0	mode set to DALI 24 bit	res.	**DALI High**	**DALI Mid**	**DALI Low**	DTR (not used here)	Priority (not used here)	Device Type (not used here)	res.

Response:

Byte:	0	1	2	3	4	5	6	7	8	9
Hex:	0x12	0x72	0x00	0x00	0x00	0x00	0xFF	0x05	0x12	0x03

Interpretation: Status 0x82, low nibble value 2: DALI 8 bit Data

→DALI 8bit Data : 0xFF according to instance type different meaning

DALI-2 MC Pushbutton (instance type 1): 0x00 button is released, 0xFF while pressed

DALI-2 CS Motion Sensor (instance type 2): 0x00 no motion , 0xAA occupied (within hold time of last motion), 0xFF motion.

DALI-2 CS light sensor (instance type 0): Input Value + Input Value Latch to be evaluated

e.g. Input value latch: 0x15, input value 0xAE → bin: 0001 0101 1010 1110

0-2046lux - 11bit, resolution: 1lux

in decimal: 1454 lux

See device datasheets for details of evaluation of response

Table of “Query input value” DALI Frames for different instance numbers and addresses:

Command	**Address**	**Modbus Frame to send**
Query Input Value	A0^2 instance Nr0	0x1200 0x0006 0x00**01** **0x008C** 0x0000 0x0000
	A1^2 instance Nr0	0x1200 0x0006 0x00**03** **0x008C** 0x0000 0x0000
	A2^2 instance Nr0	0x1200 0x0006 0x00**05** **0x008C** 0x0000 0x0000
	…	
	A62^2 instance Nr0	0x1200 0x0006 0x00**7D** **0x008C** 0x0000 0x0000
	A63^2 instance Nr0	0x1200 0x0006 0x00**7F** **0x008C** 0x0000 0x0000

	A0[2] instance Nr1	0x1200 0x0006 0x0001 **0x018C** 0x0000 0x0000
	…	
	A0[2] instance Nr2	0x1200 0x0006 0x0005 **0x028C** 0x0000 0x0000
	…	
Query Input Value Latch	A0[2] instance Nr0	0x1200 0x0006 0x0001 **0x008D** 0x0000 0x0000
	A1[2] instance Nr0	0x1200 0x0006 0x0003 **0x008D** 0x0000 0x0000
	…	

3. Python Example Code

Setting up the connection

Host: 192.168.0.98 → IP Address of the DALI-2 IoT4 - Port: 502

Unit id defines which DALI Line or zone is addressed - it is set bitwise to address DALI Line or Zone:

Bit:	7	6	5	4	3	2	1	0
	0				*DALI 3*	*DALI 2*	*DALI 1*	*DALI 0*

Set up connection: (device IP 192.168.0.98, addressing line0)

```
from pyModbusTCP.client import ModbusClient

client = ModbusClient(host='192.168.0.98', port=502)
client.connect()
```

Send DALI Control Command Scene 0 to group 0

Type	Hex Data	Address	Command
DALI16 IAP	81 10	G0	GOTO SCENE 0

```
arguments = {
    'read_address':    101,
    'read_count':      5,
    'write_address':   100,
}

# write to group 0, line 0 (unit id = 1) - scene 0
write_registers = [0x1202, 0x0003, 0x0000, 0x8110, 0x0000, 0x0000]
unit_id= 0x01
answer = client.readwrite_registers(values=write_registers, slave=unit_id, **arguments)
print([hex(i) for i in answer.registers])
```

Query DALI Status A0 as in “Example: Query DALI Status”

Type	Hex Data	Address	Command
DALI16 Query	01 90	A0	QUERY STATUS
DALI8 Answer	04		= 4 (0x04)

```
arguments = {
    'read_address':    101,
    'read_count':      5,
    'write_address':   100,
}
#query status line 1 (unit id = 2) address 0
write_registers = [0x1203, 0x0003, 0x0000, 0x0190, 0x0000, 0x0000]
unit_id= 0x02
answer = client.readwrite_registers(values= write_registers, slave=unit_id, **arguments)
print([hex(i) for i in answer.registers])
```

Set RGB Colour values as in “Example: Send RGB Colour Commands”

In these examples the answers between writing are not evaluated, to make sure all commands are received and sent to the DALI bus and to capture errors the answers should be evaluated before sending the next command.

```
unit_id= 0x01
arguments = {
    'read_address':    101,
    'read_count':      5,
    'write_address':   100,
    'slave':        unit_id,
}
# write to line 1, address 4, DT8 colour command blue:

# data transfer register=0 0x00 (red)
client.readwrite_registers(values=[0x1201, 0x0003, 0x0000, 0xA300, 0x0000, 0x0000], **arguments)

# data transfer register=1 0x00 (green)
client.readwrite_registers(values=[0x1202, 0x0003, 0x0000, 0xC300, 0x0000, 0x0000], **arguments)

# data transfer register 2= 0xFE (blue)
client.readwrite_registers(values=[0x1203, 0x0003, 0x0000, 0xC5FE, 0x0000, 0x0000], **arguments)

# enable DT8 + Set temp dim level
client.readwrite_registers(values=[0x1204, 0x0803, 0x0000, 0x09EB, 0x0000, 0x0800], **arguments)

# enable DT8 + activate command
client.readwrite_registers(values=[0x1205, 0x0803, 0x0000, 0x09E2, 0x0000, 0x0800], **arguments)
```

Close Client Connection

After reading and sending all required data close the client connection.

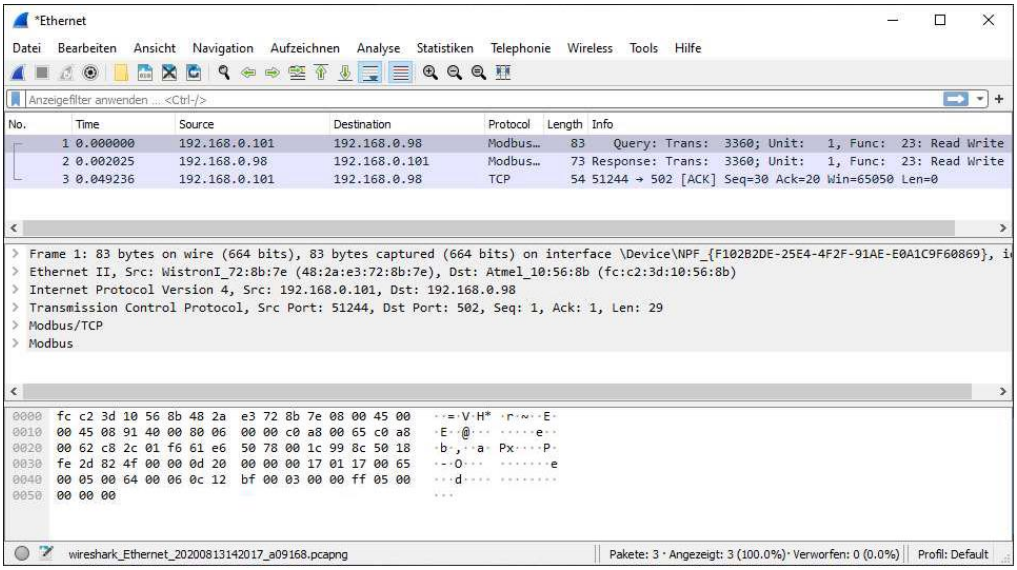
```
client.close
```

4. Example – Wireshark

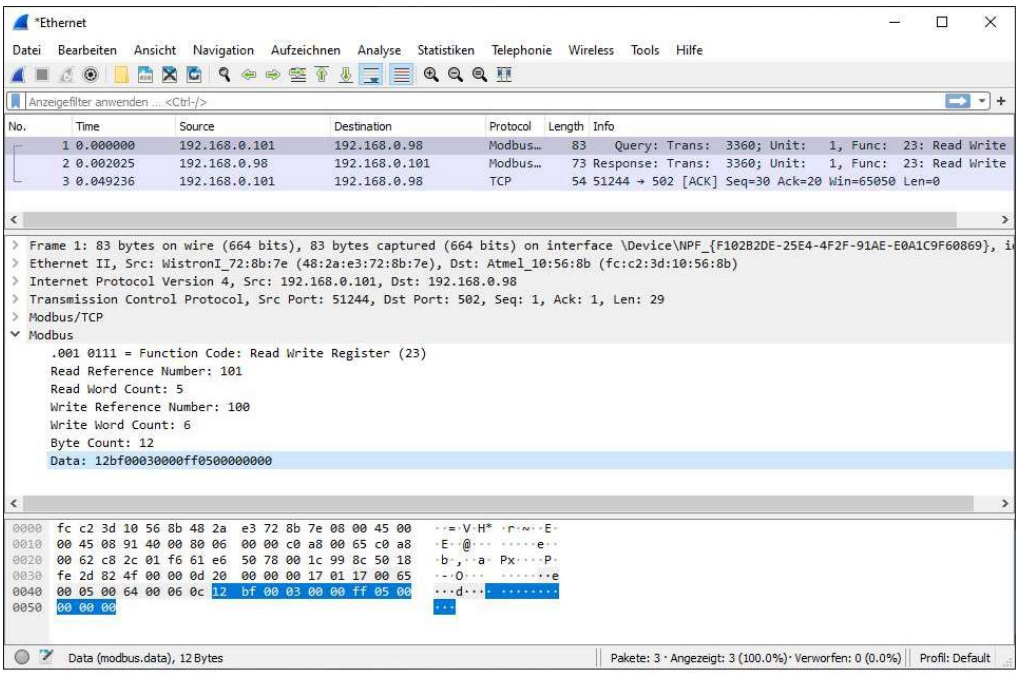
The examples below were made with the program Wireshark, to analyse the communication on the Modbus layer.

4.1 DALI command „RECALL MAX“ to Broadcast on line 0

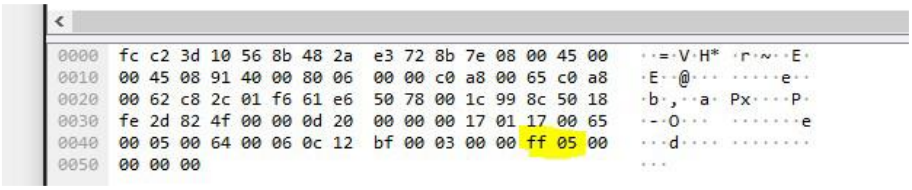
Whole Frame:



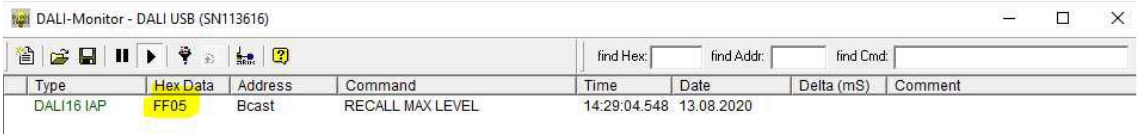
Data:



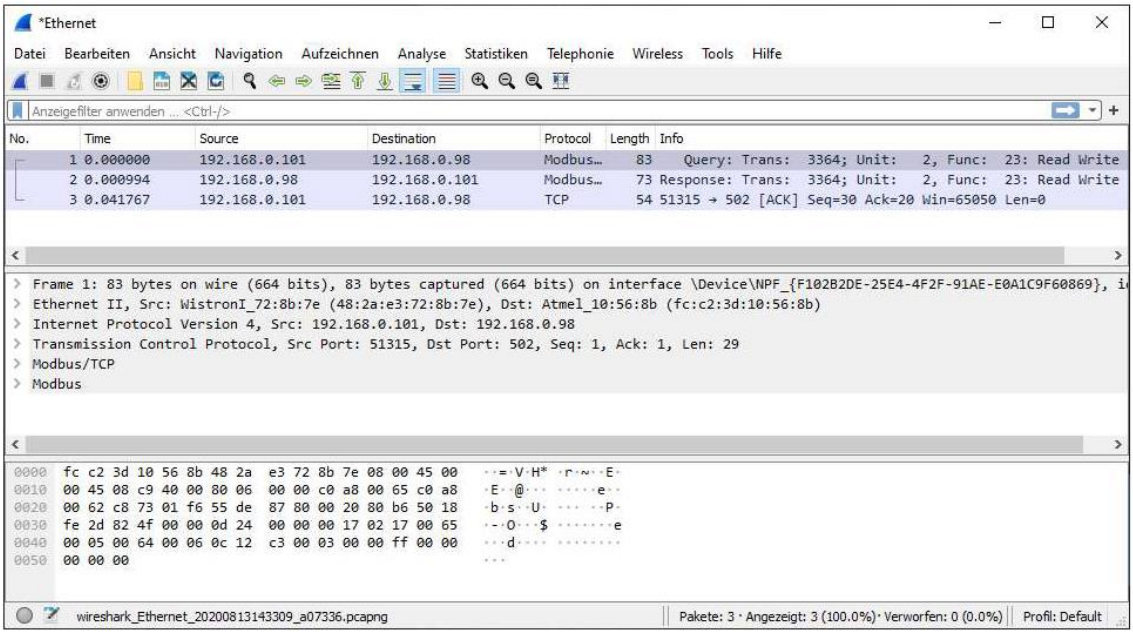
DALI Frame:



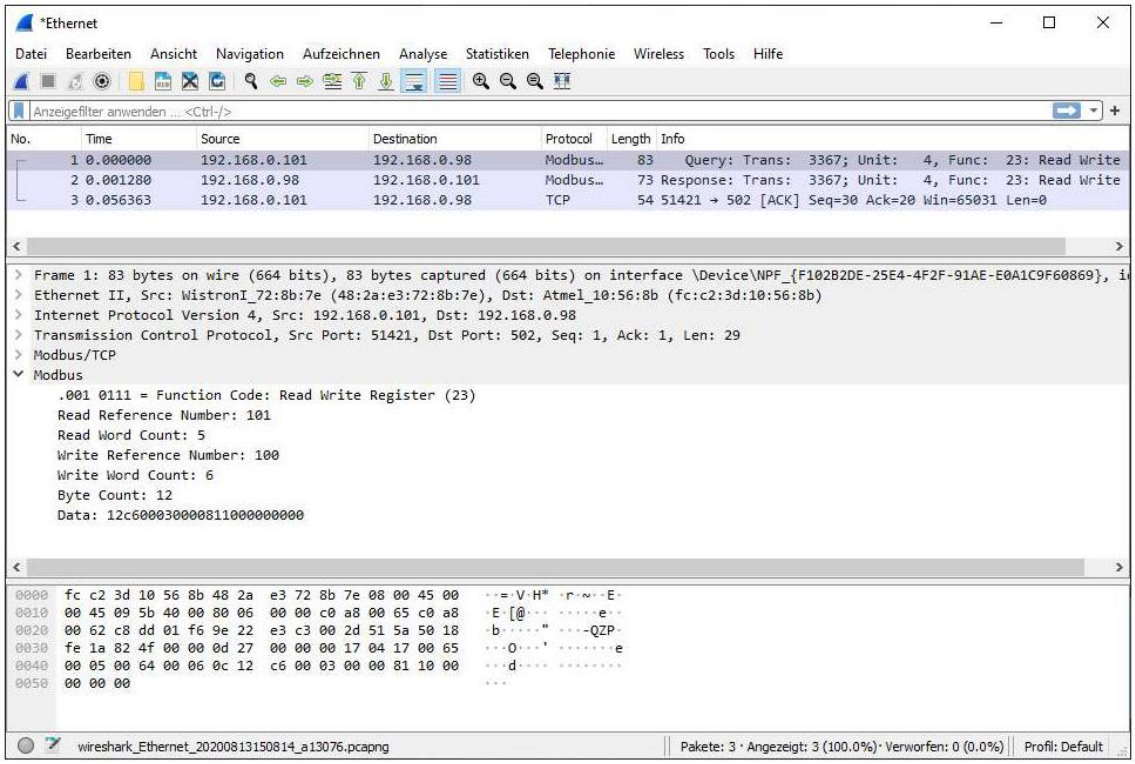
DALI Monitor:



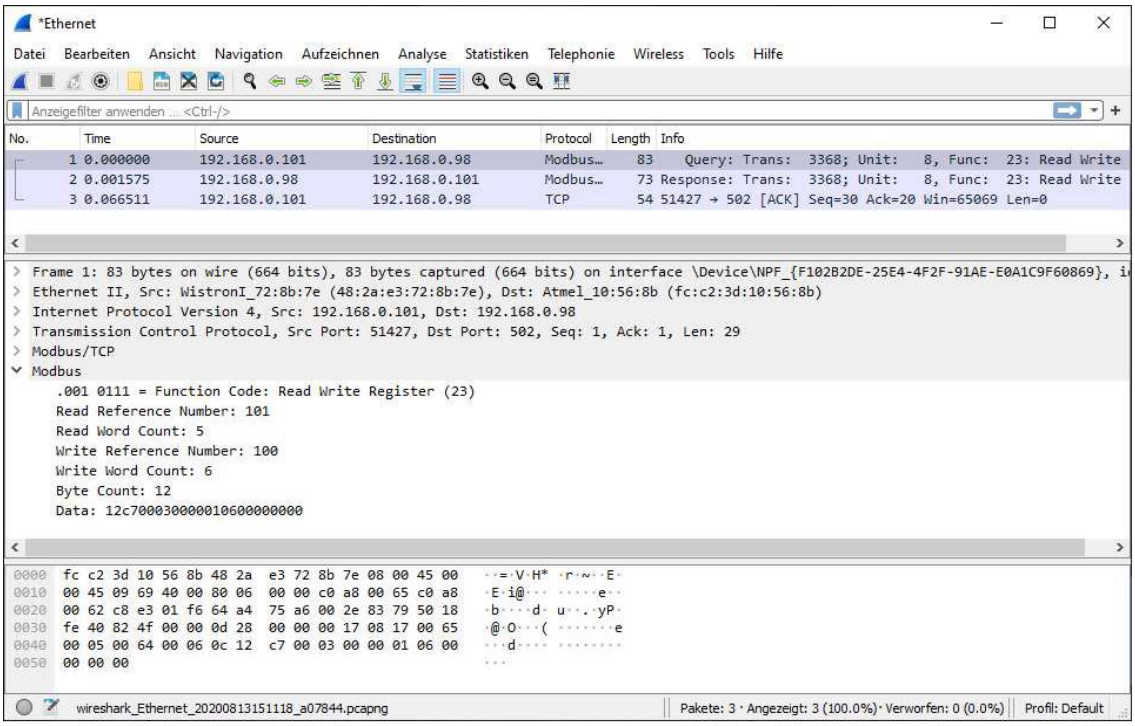
4.2 DALI command “OFF” to Broadcast on Line 1



4.3 DALI command “GO TO Scene 0” to group 0 on Line 2



4.4 DALI command “RECALL MIN LEVEL” to single address A0 on Line 3



5. Table of common DALI Frames

Command	to Address -->	Register	Words	Modbus Frame to send					
DAP 1%	Broadcast	100	6	0x1201	0x0003	0x0000	**0xFE56**	0x0000	0x0000
	A0	100	6	0x1201	0x0003	0x0000	**0x0056**	0x0000	0x0000
	A1	100	6	0x1201	0x0003	0x0000	**0x0256**	0x0000	0x0000
	…								
	A62	100	6	0x1201	0x0003	0x0000	**0x7C56**	0x0000	0x0000
	A63	100	6	0x1201	0x0003	0x0000	**0x7E56**	0x0000	0x0000
	G0	100	6	0x1201	0x0003	0x0000	**0x8056**	0x0000	0x0000
	G1	100	6	0x1201	0x0003	0x0000	**0x8256**	0x0000	0x0000
	...								
	G14	100	6	0x1201	0x0003	0x0000	**0x9C56**	0x0000	0x0000
	G15	100	6	0x1201	0x0003	0x0000	**0x9E56**	0x0000	0x0000
DAP 50%	Broadcast	100	6	0x1201	0x0003	0x0000	**0xFEE5**	0x0000	0x0000
	A0	100	6	0x1201	0x0003	0x0000	**0x00E5**	0x0000	0x0000
	A1	100	6	0x1201	0x0003	0x0000	**0x02E5**	0x0000	0x0000
	…								
	A62	100	6	0x1201	0x0003	0x0000	**0x7CE5**	0x0000	0x0000
	A63	100	6	0x1201	0x0003	0x0000	**0x7EE5**	0x0000	0x0000
	G0	100	6	0x1201	0x0003	0x0000	**0x80E5**	0x0000	0x0000
	G1	100	6	0x1201	0x0003	0x0000	**0x82E5**	0x0000	0x0000
	...								
	G14	100	6	0x1201	0x0003	0x0000	**0x9CE5**	0x0000	0x0000
	G15	100	6	0x1201	0x0003	0x0000	**0x9EE5**	0x0000	0x0000
DAP 100%	Broadcast	100	6	0x1201	0x0003	0x0000	**0xFEFE**	0x0000	0x0000
	A0	100	6	0x1201	0x0003	0x0000	**0x00FE**	0x0000	0x0000
	A1	100	6	0x1201	0x0003	0x0000	**0x02FE**	0x0000	0x0000
	…								
	A62	100	6	0x1201	0x0003	0x0000	**0x7CFE**	0x0000	0x0000
	A63	100	6	0x1201	0x0003	0x0000	**0x7EFE**	0x0000	0x0000
	G0	100	6	0x1201	0x0003	0x0000	**0x80FE**	0x0000	0x0000
	G1	100	6	0x1201	0x0003	0x0000	**0x82FE**	0x0000	0x0000
	...								
	G14	100	6	0x1201	0x0003	0x0000	**0x9CFE**	0x0000	0x0000
	G15	100	6	0x1201	0x0003	0x0000	**0x9EFE**	0x0000	0x0000
OFF	Broadcast	100	6	0x1201	0x0003	0x0000	**0xFF00**	0x0000	0x0000
	A0	100	6	0x1201	0x0003	0x0000	**0x0100**	0x0000	0x0000
	A1	100	6	0x1201	0x0003	0x0000	**0x0300**	0x0000	0x0000
	…								
	A62	100	6	0x1201	0x0003	0x0000	**0x7D00**	0x0000	0x0000
	A63	100	6	0x1201	0x0003	0x0000	**0x7F00**	0x0000	0x0000
	G0	100	6	0x1201	0x0003	0x0000	**0x8100**	0x0000	0x0000
	G1	100	6	0x1201	0x0003	0x0000	**0x8300**	0x0000	0x0000

	...								
	G14	100	6	0x1201	0x0003	0x0000	**0x9D00**	0x0000	0x0000
	G15	100	6	0x1201	0x0003	0x0000	**0x9F00**	0x0000	0x0000

MAX	Broadcast	100	6	0x1201	0x0003	0x0000	**0xFF05**	0x0000	0x0000
	A0	100	6	0x1201	0x0003	0x0000	**0x0105**	0x0000	0x0000
	A1	100	6	0x1201	0x0003	0x0000	**0x0305**	0x0000	0x0000
	…								
	A62	100	6	0x1201	0x0003	0x0000	**0x7D05**	0x0000	0x0000
	A63	100	6	0x1201	0x0003	0x0000	**0x7F05**	0x0000	0x0000
	G0	100	6	0x1201	0x0003	0x0000	**0x8105**	0x0000	0x0000
	G1	100	6	0x1201	0x0003	0x0000	**0x8305**	0x0000	0x0000
	...								
	G14	100	6	0x1201	0x0003	0x0000	**0x9D05**	0x0000	0x0000
	G15	100	6	0x1201	0x0003	0x0000	**0x9F05**	0x0000	0x0000

MIN	Broadcast	100	6	0x1201	0x0003	0x0000	**0xFF06**	0x0000	0x0000
	A0	100	6	0x1201	0x0003	0x0000	**0x0106**	0x0000	0x0000
	A1	100	6	0x1201	0x0003	0x0000	**0x0306**	0x0000	0x0000
	…								
	A62	100	6	0x1201	0x0003	0x0000	**0x7D06**	0x0000	0x0000
	A63	100	6	0x1201	0x0003	0x0000	**0x7F06**	0x0000	0x0000
	G0	100	6	0x1201	0x0003	0x0000	**0x8106**	0x0000	0x0000
	G1	100	6	0x1201	0x0003	0x0000	**0x8306**	0x0000	0x0000
	...								
	G14	100	6	0x1201	0x0003	0x0000	**0x9D06**	0x0000	0x0000
	G15	100	6	0x1201	0x0003	0x0000	**0x9F06**	0x0000	0x0000

SCENE 0	Broadcast	100	6	0x1201	0x0003	0x0000	**0xFF10**	0x0000	0x0000
	A0	100	6	0x1201	0x0003	0x0000	**0x0110**	0x0000	0x0000
	A1	100	6	0x1201	0x0003	0x0000	**0x0310**	0x0000	0x0000
	…								
	A62	100	6	0x1201	0x0003	0x0000	**0x7D10**	0x0000	0x0000
	A63	100	6	0x1201	0x0003	0x0000	**0x7F10**	0x0000	0x0000
	G0	100	6	0x1201	0x0003	0x0000	**0x8110**	0x0000	0x0000
	G1	100	6	0x1201	0x0003	0x0000	**0x8310**	0x0000	0x0000
	...								
	G14	100	6	0x1201	0x0003	0x0000	**0x9D10**	0x0000	0x0000
	G15	100	6	0x1201	0x0003	0x0000	**0x9F10**	0x0000	0x0000

SCENE 1	Broadcast	100	6	0x1201	0x0003	0x0000	**0xFF11**	0x0000	0x0000
	A0	100	6	0x1201	0x0003	0x0000	**0x0111**	0x0000	0x0000
	A1	100	6	0x1201	0x0003	0x0000	**0x0311**	0x0000	0x0000
	…								
	A62	100	6	0x1201	0x0003	0x0000	**0x7D11**	0x0000	0x0000
	A63	100	6	0x1201	0x0003	0x0000	**0x7F11**	0x0000	0x0000
	G0	100	6	0x1201	0x0003	0x0000	**0x8111**	0x0000	0x0000
	G1	100	6	0x1201	0x0003	0x0000	**0x8311**	0x0000	0x0000
	...								

G14	100	6	0x1201	0x0003	0x0000	**0x9D11**	0x0000	0x0000
G15	100	6	0x1201	0x0003	0x0000	**0x9F11**	0x0000	0x0000

Additional Information

DALI-Cockpit –configuration tool from Lunatone for DALI systems, free when used with a Lunatone DALI interface device
https://www.lunatone.com/en/product/dali-cockpit/

Lunatone DALI products
https://www.lunatone.com/

Lunatone datasheets and manuals
https://www.lunatone.com/downloads-a-z/

Contact

Technical Support: support@lunatone.com

Requests: sales@lunatone.com

www.lunatone.com

Disclaimer

Subject to change. Information provided without guarantee. The datasheet refers to the current delivery.

The compatibility with other devices must be tested in advance to the installation.

