

---

11/2013

# Technical Manual

## MDT Multimedia Gateway



SCN-MMG01.01

## 1 Content

1 Content .....	2
2 Introduction.....	3
2.1 Connector pin assignment.....	3
3 Communication Objects .....	4
3.1 Overview.....	4
3.2 Default settings of the communication objects .....	5
4 Reference-ETS-Parameter .....	6
4.1 General .....	6
4.2 Header/Footer.....	7
4.3 KNX => RS232 (1 Bit).....	8
4.4 KNX => RS232 (1 Byte).....	8
4.5 RS232 => Bit-value.....	9
4.6 RS232 => Byte-value.....	9
4.7 RS232 => Text String.....	11
4.8 Macro functions .....	12
4.9 Convert RS232 Byte status into KNX Bit values.....	13
5 Examples.....	14
5.1 Settings for T+A K8 series Multiroom Receiver.....	14
5.2 Settings for Löwe TV.....	18
5.3 Settings for Russound.....	19
6 Index .....	21
6.1 List of figures .....	21
6.2 List of tables.....	21
7 Attachment.....	22
7.1 Statutory requirements.....	22
7.2 Routine disposal .....	22
7.3 Assemblage.....	22
7.4 Datasheet .....	23

## 2 Introduction

The MDT Multimedia Gateway offers possibilities for connecting Multimedia devices to the KNX-Bus. Connection between Multimedia Gateway and Multimedia device is established by a RS232 interface, which is included in delivery.

The Gateway offers prepared settings for devices of the Companies Russound, Löwe TV and the T+A K8 series. Via the setting Universal Plaintext and Universal Hex other devices with the appropriate data format can be controlled.

### 2.1 Connector pin assignment

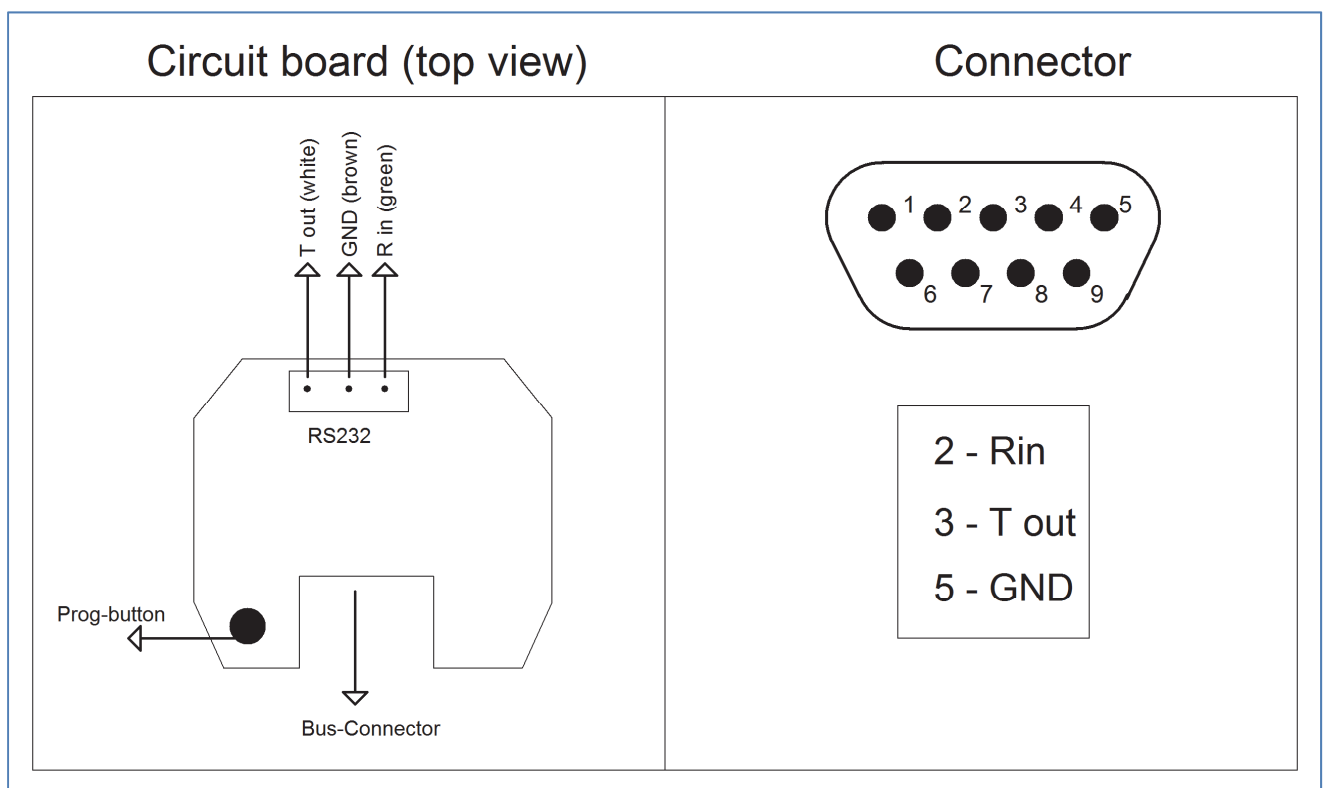


Figure 1: Connector pin assignment

### 3 Communication Objects

#### 3.1 Overview

No.	Function	Usage
0-39	KNX => RS232 Command	Sends as well at a Bit value of 1 as at a value of 0 a command to the Multimedia Device. Telegrams for a 0-command as for a 1-command can be set at the parameters.
40-79	KNX => RS232 Command	Sends only at a Bit value of 1 a command to the Multimedia Device. Telegrams can only be adjusted for a 1-command at the parameters.
80-99	KNX => RS232 Command	The received byte value is assigned to the Multimedia device, e.g. a value for the loudness.
100-109	RS232 => KNX Status	The received value of the Multimedia device causes sending a 0-command or a 1-command via this communication object. Different telegrams can be adjusted, for sending a logical 0 or a logical 1.
110-119	RS232 => KNX Status	The received telegram of the Multimedia Device causes sending a 0-command via this communication object.
120-129	RS232 => KNX Status	The received telegram of the Multimedia Device causes sending a byte-value via this communication object.
130-134	RS23 => KNX Status Text	The received telegram of the Multimedia Device causes sending a string as 14Byte telegram to the bus.
135-140	Macro	As soon as the adjusted value is received at this communication object, the adjusted sequence of commands is preceded.
141-147	RS232 => KNX Bit Status	The adjusted Byte-Object is converted in to single Bit status objects.

Table 1: Communication objects

### 3.2 Default settings of the communication objects

The following chart shows the Default settings of the communication objects:

Default Settings									
No.	Button	Function	Length	Priority	C	R	W	T	U
0-39	Object 0-39	KNX => RS232 Command	1 Bit	Low	X		X		
40-79	Object 40-79	KNX => RS232 Command	1 Bit	Low	X		X		
80-99	Object 80-99	KNX => RS232 Command	1 Byte	Low	X		X		
100-109	Object 100-109	RS232 => KNX Status	1 Bit	Low	X	X		X	
110-109	Object 110-109	RS232 => KNX Status	1 Bit	Low	X	X		X	
120-129	Object 120-129	RS232 => KNX Status	1 Byte	Low	X	X		X	
130-134	Object 130-134	RS232 => KNX Status Text	14 Byte	Low	X	X		X	
135-140	Object 135-140	Macro	1 Bit	Low	X		X		
141-147	Object 135-140	RS232 => KNX Bit Status	1 Bit	Low	X	X		X	
149	Error	Failure message	1 Bit	Alarm	X	X		X	

Table 2: Default settings of the communication objects

You can see the default values for the communication objects from the upper chart. According to requirements the priority of the particular communication objects as well as the flags can be adjusted by the user. The flags allocates the function of the objects in the programming thereby stands C for communication, R for Read, W for write, T for transmit and U for update.

## 4 Reference-ETS-Parameter

### 4.1 General

The following figure shows the general settings:

General settings	
Manufacturer / Device	T+A K8 Series
Baud rate	115200
Parity	no
Stopbit	1 Stopbit
Send pause between RS232 telegrams	40 ms
Detect data end	via length specification in data
Position of length specification	2
Offset of length specification	4
Data end (Timeout)	2 ms

Figure 2: General settings

At this menu the general settings of the device must be recorded. If one of the predefined producers is in use, this producer can be adjusted at this menu. There are predefined commands available for this producer, which can be used directly. The settings for the transfer should be made from the manual of the device.

## 4.2 Header/Footer

The following figure shows the settings for the Header and the Footer of the telegram:

Header / Footer	
Number of Bytes for Header H1	3 Byte
Header Hex H1	
Number of Bytes for Header H2	3 Byte
Header Hex H2	0104C4
Number of Bytes for Header H3	no
Number of Bytes for Footer F1	1 Byte
Footer Hex F1	02
Checksum F1	T+A / Metz
End mark F1	no
Number of Bytes for Footer F2	1 Byte
Footer Hex F2	02
Checksum F2	T+A / Metz
End mark F2	no
Number of Bytes for Footer F3	only checksum and end mark
Checksum F3	off
End mark F3	no

Figure 3: Header and Footer

Because most of the producer use a fixed device identification at the begin and the end of the device as well as a Checksum at the end, up to three header and footer can be adjusted. The Header is put in front of the telegram and the footer is put at the end of the telegram.

Which Header and Footer shall be used can be adjusted for each group of telegrams.

There are two predefined settings for calculating the checksum: T+A/Metz and Russound.

The checksum of T+A/Metz devices is calculated as follows:  $(\text{Sum Byte } 1-5)/0xFF$ .

The checksum of Russound devices is calculated as follows:  $(\text{Sum of all command bytes} + \text{decimal value } 6) \text{ masked with } 0xF7$ .

If other checksums shall be used, the checksum at the Header Footer menu must be deactivated and must be calculated for every command individually.

### 4.3 KNX => RS232 (1 Bit)

The following figure shows the available settings for 1 Bit communication objects:

Bit Obj 0-9 KNX => RS232 value 0/1	
Selection Header	H1
Selection Footer	F1
Object 0 at value = 0	00700502020000F123000000000001
Object 0 at value = 1	00700502020000F123000100000001

Figure 4: KNX => RS232 (1 Bit)

The Group of the objects from 0 to 39 can transmit telegrams to the multimedia device as well at a 0-command as at a 1-command. The Group of the objects from 40 to 79 can transmit telegrams to the multimedia device only at a 1-command.

The transmitted telegram contains of the adjusted Header (here H1) + the adjusted telegram + the adjusted Footer (here F1).

### 4.4 KNX => RS232 (1 Byte)

The following figure shows the available settings for 1 Byte communication objects:

Byte Obj 80-99 KNX => RS232	
Selection Header for Objects 80-89	H1
Selection Footer for Objects 80-89	F1
Selection Header for Objects 90-99	no Header
Selection Footer for Objects 90-99	no Footer
Object 80	00700502020000F12100%10000001

Figure 5: KNX => RS232 (1 Byte)

Received byte values from the KNX-Bus can be transmitted to the multimedia device via the objects 80-99.

The character string “%1” in the telegram indicates that the Byte value is calculated into a relative value from 0 to 100% and transmitted as this to the device. The character string “#1” in the telegram indicates that the Byte value is transmitted directly as absolute value to the device. At this setting the KNX value is limited to 100.



### 4.5 RS232 => Bit-value

The following figure shows the available settings for the sending of 1 Bit KNX-status objects at defined telegrams:

Bit Obj 100-109 RS232 => KNX send value 0/1	
Send value = 0 by object 100	<input type="text"/>
Send value = 1 by object 100	<input type="text"/>
Send value = 0 by object 101	<input type="text"/>
Send value = 1 by object 101	<input type="text"/>

Figure 6: RS232 => Bit-value

At the objects 100 to 109 telegrams can be adjusted as well for sending a logical 0 and a logical 1. At the objects 110 to 119 telegrams can only be adjusted for sending a logical 1.

The whole telegram, with Header and Footer, must be adjusted for the status objects. The Header and Footer functions are not used at this group. Hex values which should not be evaluated must be entered with “##”.

### 4.6 RS232 => Byte-value

The following figure shows the available settings for the sending of 1 Byte KNX-status objects at defined telegrams:

Byte Obj 120-129 RS232 => KNX send Byte value	
Send Byte value by object 120 (#0 / %0)	<input type="text" value="@1\$L64#0#1\$0"/>
Text (String) contains	<input type="text" value="yes"/>
Text length contains in data	<input type="text" value="yes"/>
Text length reduce by	<input type="text" value="1"/>
Text option	<input type="text" value="cut to long text (&gt;14 Byte)"/>

Figure 7: RS232 => Byte-value

For adjusting the telegram following options are available:

- %x  
The data is transmitted to the object 120 + x (x = 0-9) as percental value (0-100%)
- #x  
The data is transmitted to the object 120 + x (x = 0-9) directly (0-255)
- ##  
The data byte is not evaluated
- \$x  
String/Text data are transmitted to the object 130 + x
- \$L  
Adjustment of the length of the evaluating string
- \$\$  
The string is not evaluated

By using the options, the telegram, which contains several data, can be evaluated according to the specific wishes.

Additional the following settings can be adjusted:

- String/Text contains  
indicates if a String/Text is included in the telegram
- Text length contains in data  
indicates if an information of the length of the character string is included in the telegram
- Text length reduce by  
indicates if the text shall be shortened, e.g. for fading out the Checksum  
Example: The information of the length in the telegram is 10, but this information contains parts of the command, so the real length of the text must be shortened by 2.
- Text option  
indicates if character strings which are longer than 14 Byte shall be shortened or transmitted to the next telegram.  
Example: The data has a length of 20Byte. By using the option, object 130 can display the first 14 Byte and object 131 can display the last 6 Byte.

Because as well at the options as at the settings possibilities for forwarding telegrams are available, it must be kept in mid which objects are already in use.

The adjusted command 01\$L64#0#1\$0 refers to T+A K8 and has the following meaning:

- 01 --> Header of the command
- \$L means that the telegram transmit the length of the string at this place
- 64 --> according to T+A documentation the Status 1 is evaluated
- #0 means that the status byte 1 is transmitted to object 120
- #1 means that the status byte 2 is transmitted to object 121
- \$0 means that the string is transmitted to object 130

Comprehensive examples for T+A are available at **5.1 Settings for T+A K8 series Multiroom Receiver.**

## 4.7 RS232 => Text String

The following figure shows the available settings for sending 14 Byte Text-Strings:

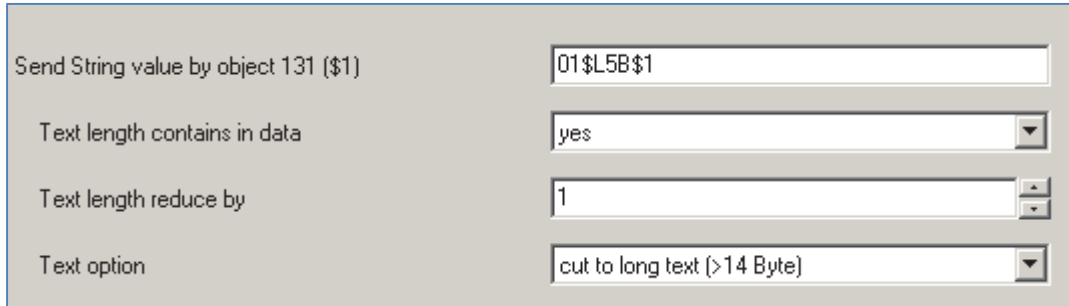


Figure 8: RS232 => Text String

The following options are available for the adjusted telegram:

- \$x  
String Data is transmitted to the object 130 + x
- \$L  
Adjustment of the length of the string which shall be evaluated
- ##  
Data byte will not be evaluated
- \$#  
String will not be evaluated

By using the options it is possible to analyze and demount the telegram according the specific wishes. Additional the following settings can be done:

- Text length contains in data  
indicates if information about the length of the string is included in the telegram
  - if yes: Text length reduce by  
indicates if the text shall be shortened, e.g. for fading out the checksum
  - if no: Exact length  
adjustment of the length of the text
- Text option  
indicates if strings, which are longer than 14 Byte shall be shortened or transmitted to the next objects

Because as well at the options as at the settings possibilities for forwarding telegrams are available, it must be kept in mid which objects are already in use.

Also from the fuction RS232 => Byte-value, values can be transmitted to this group.

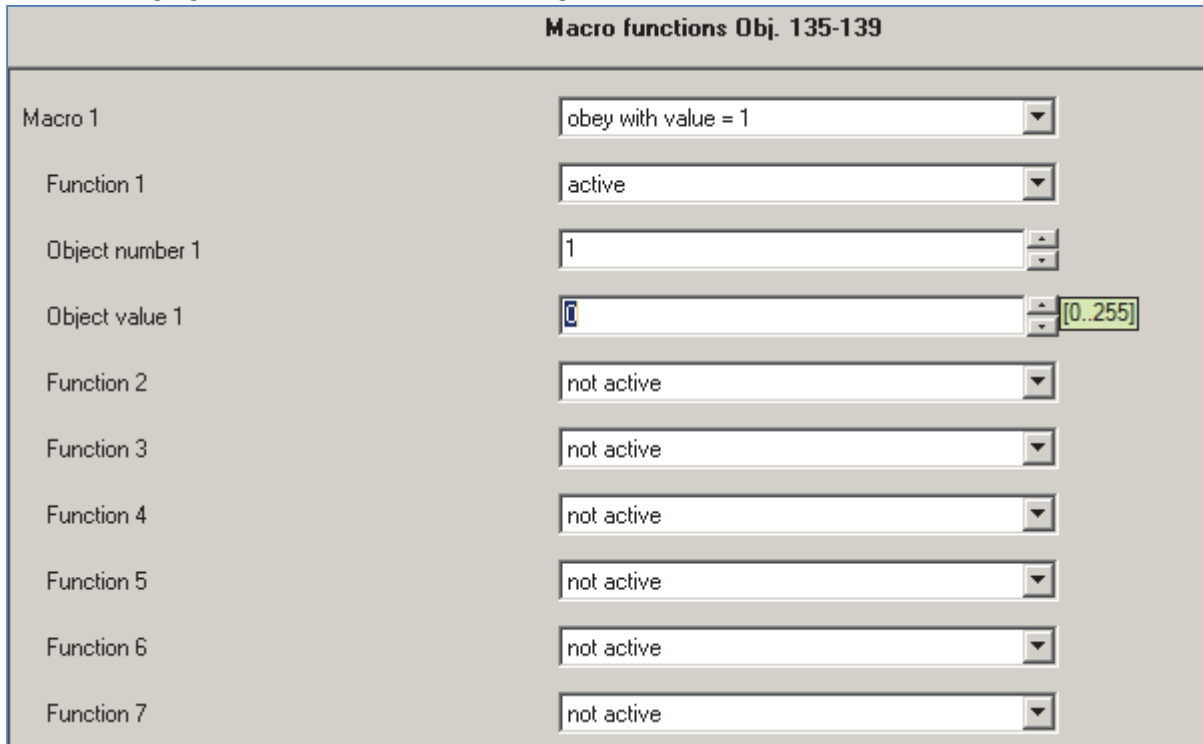
The adjusted command 01\$L64#0#1\$0 refers to T+A K8 and has the following meaning:

- 01 --> Header of the command
- \$L means that the telegram transmit the length of the string at this place
- 64 --> according to T+A documentation the Status 3 is evaluated
- \$1 means that the string is transmitted to object 131

Comprehensive examples for T+A are available at **5.1 Settings for T+A K8 series Multiroom Receiver.**

## 4.8 Macro functions

The following figure shows the available settings for the macro functions:



Macro functions Obj. 135-139	
Macro 1	obey with value = 1
Function 1	active
Object number 1	1
Object value 1	[0..255]
Function 2	not active
Function 3	not active
Function 4	not active
Function 5	not active
Function 6	not active
Function 7	not active

Figure 9: Macro functions

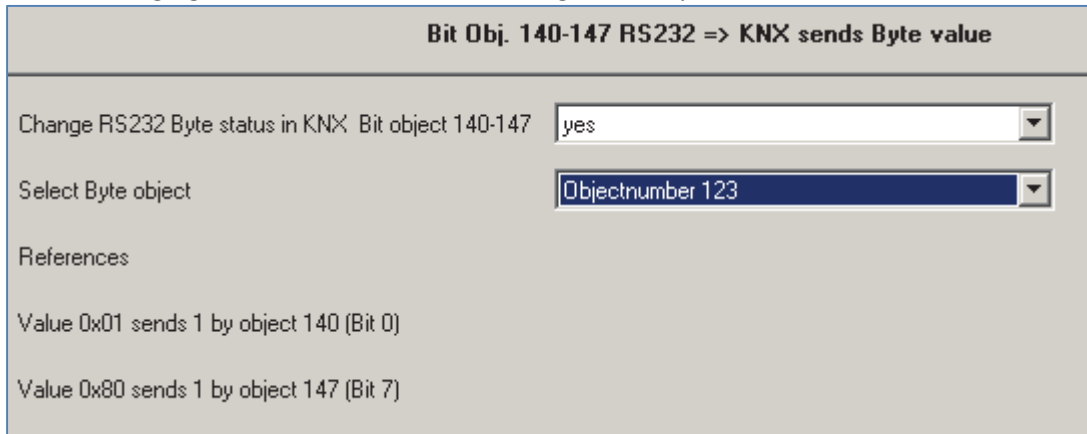
The Macro functions enable the sequential sending of multiple commands via activating one object. Up to 7 commands can be selected for each macro.

For selecting the commands, the objects 0-99 are available. Each object can be called with an individual adjusted value. The byte values are transmitted 1:1 and will not be calculated into a relative value.

Additional can be selected which value (0/1) activates the macro function.

## 4.9 Convert RS232 Byte status into KNX Bit values

The following figure shows the available settings for this parameter:



**Bit Obj. 140-147 RS232 => KNX sends Byte value**

Change RS232 Byte status in KNX Bit object 140-147

Select Byte object

References

Value 0x01 sends 1 by object 140 (Bit 0)

Value 0x80 sends 1 by object 147 (Bit 7)

Figure 10: Convert RS232 Byte status into KNX Bit values

This function enables converting a byte status, which is sent from the Multimedia Device and contains several states, into individual bit values. The received byte value is divided into 8 Bit-values and the Bit 0 is sent to object 140, the Bit 1 to object 141.

Example:

Bit0 Speaker zone 1	=> Obj 140
Bit1 Speaker zone 2	=> Obj 141
Bit2 Speaker zone 3	=> Obj 142
Bit3 Speaker zone 4	=> Obj 143
...	
bit7 System on	=> Obj 147

## 5 Examples

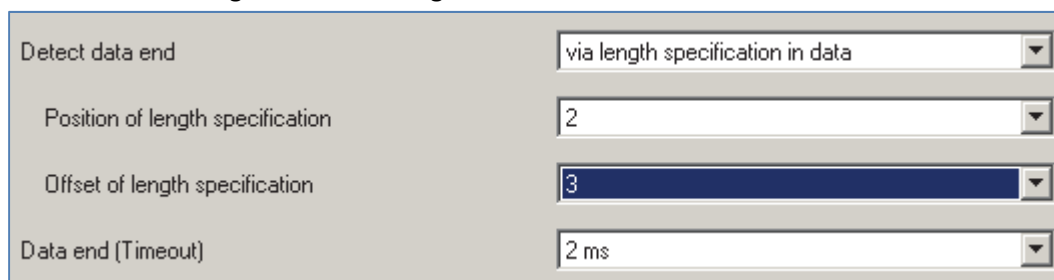
### 5.1 Settings for T+A K8 series Multiroom Receiver

The settings of the parameter are described according to an example:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 6	Byte 7
Header	Length of the telegram	adress	command	Flag Byte	checksum
0x01	0x03	0xC4	0x57	0x02	0xE4

Table 3: Telegram Example

- End of the telegram via the telegram:



The screenshot shows a configuration window for 'Detect data end'. It contains four settings, each with a dropdown menu:

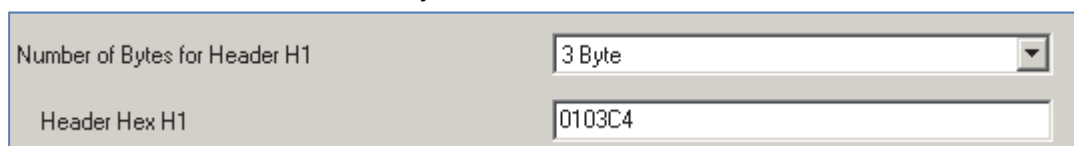
- Detect data end:** via length specification in data
- Position of length specification:** 2
- Offset of length specification:** 3
- Data end (Timeout):** 2 ms

Figure 11: Example 1, general settings

At this example, the length of the telegram is transmitted at the second byte, so the position of length specification is assigned to 2.

To the end of the telegram (without checksum) are 3 bytes remaining (Offset of length specification = 3). So the whole length of the telegram is defined. The data end (timeout) can be adjusted to the minimum value, because the baud rate of the RS232 is 115200, which makes crossovers of telegrams impossible.

- The telegram example contains start values before the command and end values after the command. These values are adjusted for the Header and Footer.



The screenshot shows a configuration window for header settings. It contains two settings:

- Number of Bytes for Header H1:** 3 Byte
- Header Hex H1:** 0103C4

Figure 12: Example 1 - Settings for Header

At first the length of the Header must be adjusted. Now a value for the Header can be assigned, here 0103C4.

Number of Bytes for Footer F1	1 Byte
Footer Hex F1	02
Checksum F1	T+A / Metz
End mark F1	no

Figure 13: Example 1 - Setting for Footer

At first the length of the Footer must be adjusted, here 1 Byte. Now the value from the example, Byte 6, must be signed in.

The checksum is calculated via the software by choosing the option T+A/Metz.

You have not to sign in the End mark, because the length of the telegram is adjusted via the general settings.

- For proceeding a command via 1 Bit object, a command from the manual of T+A Amplifier with the address 0xC4 must be signed into the parameter “Bit Object 0-39: KNX=>RS232 Value 0/1”. Here the value 0x57 must be signed in for this parameter, which switches the device on or off.
- The next example for T+A Amplifier is a telegram for adjusting the loudness of the Amplifier via a 1 Byte object:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Header	Length of the telegram	address	command	Wert	Flag Byte	Checksum
0x01	0x04	0xC4	0x00	0x19	0x02	0xE4

Table 4: Telegram Example --> Byte value

For executing this command, the Hex value 0x00 must be signed in for the parameter “Byte Object 80-99: KNX=>RS232”. As option for this parameter the character %1 is set behind the value 0x00. So the value 00%1 must be signed in for this object. The option in combination with the telegram affects that the Byte value is calculated into a percental value and transmitted as this to the multimedia device.

- For receiving status values from T+A, the telegrams must be signed in the parameter. The Gateway can receive Hex-values as Universal Clear text and transmit these telegrams as DPT-values to KNX. An exemplary status telegram can look as follows:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6...	...Byte 22	Byte 23
Header	Length of the telegram	address	Status-Byte 1	Status-Byte 2	Resource name 0...	Resource name 16	Checksum
0x01	0xXX	0x64	0x81	0xC0	0xXX	0xXX	0xXX

Table 5: Example 1, Receiving status telegrams

**Byte 1 and 3:**

At the settings for the objects 120-129, this telegram must be signed in the line for telegram 120. The Header and the address (Byte 1 and 3) are inscribed 1:1 for the parameter.

**Byte 2:**

The length of the telegram (Byte 2) shall be taken from the RS232 telegram. So the Code \$L is signed in, which indicates that the value for the length of the string is transmitted here.

**Byte 4 and 5:**

For taking the byte values for Status 1 and 2 (Byte 4 and 5) the codes #0 and #1 is used. So the values are transmitted 1:1 to the objects 120 and 121.

**Byte 6 – 22:**

The Clear text is transmitted with the bytes 6 to 22 with the maximum of 16 signs. The code \$0 is signed in to transmit the character string to object 130. If a RS232 telegram with character string is send from the device, the Multimedia Gateway sends this content via the object 130 to the KNX Bus.



- By using the Macro function, it is possible to perform several commands. The object numbers and values can be assigned user defined.

Example for the Macro function 1:

3 Commands are performed with the object 135.

- 1) Switch the device on via the object 0 with the value 1:

Macro 1	obey with value = 1
Function 1	active
Object number 1	0
Object value 1	1

Figure 14: Example Macro 1, Function 1

- 2) Set the mode to Stereo:

Function 2	active
Object number 2	43
Object value 2	1

Figure 15: Example Macro 1, Function 2

- 3) Set the loudness of area 1 to 50%:

Function 3	active
Object number 3	80
Object value 3	128

Figure 16: Example Macro 1, Function 3

## 5.2 Settings for Löwe TV

- The RS232 commands contain of ASCII signs and must be transmitted as Clear text to the device. For recognizing the End of data a control character for the vertical spacing must be transmitted:

Detect data end	via data end mark
Byte for received end	LF (0x0A)
Data end (Timeout)	5 ms

Figure 17: Example 2, general functions

- For performing a command with a 1 Bit value, the Clear text values with the appropriate commands must be signed in:

Object 0 at value = 0 (off)	power off
Object 0 at value = 1 (TV on)	power tv

Figure 18: Example 2, 1-Bit command

So the TV is switched off by sending a “0” to object 0 and switched on by sending a “1” to this object.

- At this example the Multimedia Gateway can transmit a value for the loudness:

Object 80	data volume %1
-----------	----------------

Figure 19: Example 2, 1 Byte value

With the Code %1, the Byte value is calculated to 0-100%. So if you send the value 0x128 to the Byte object, the Gateway transmits the value 50% to the TV.

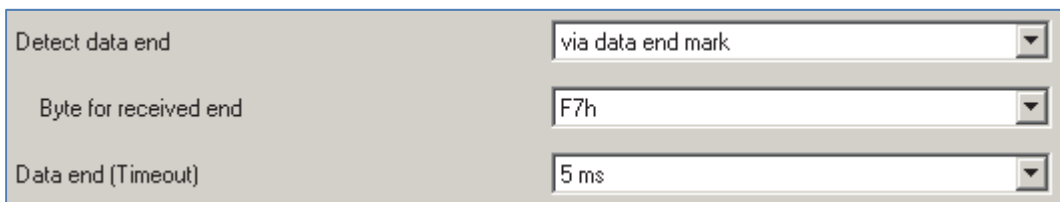
### 5.3 Settings for Russound

- The settings are described via the following example:

1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2
F	0	0	7	0	0	7	0	0	0	0	0	F	2	0	0	0	0	0	0	X	F
0	0	0	F	0	0	0	5	2	2	0	0	1	3	0	1	0	0	1	X	7	

Table 6: Example 3, Telegram Russound

- Byte 22 is the end of the telegram:



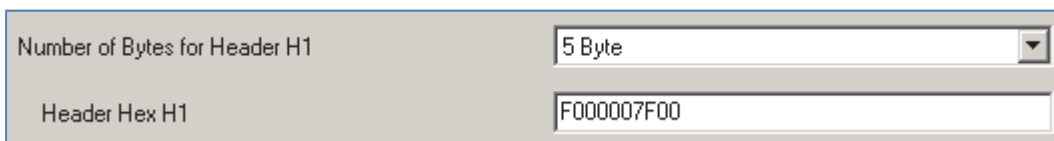
Detect data end: via data end mark

Byte for received end: F7h

Data end (Timeout): 5 ms

Figure 20: Example 3, general settings

- The first 5 Bytes are at every RS232-Russound telegram the same. So these are signed in for the Header:

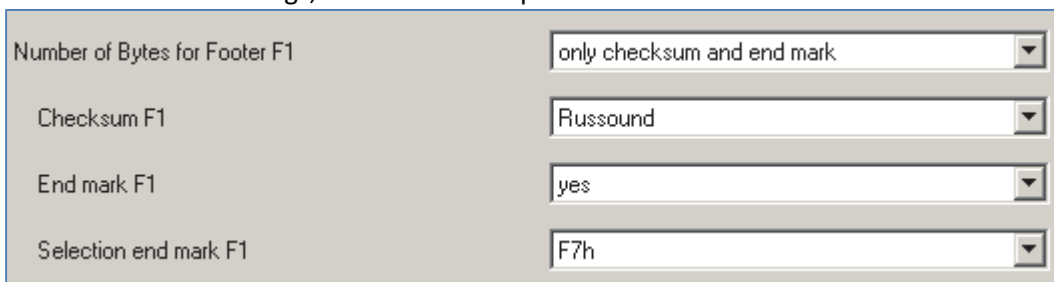


Number of Bytes for Header H1: 5 Byte

Header Hex H1: F000007F00

Figure 21: Example 3, Header settings

- At the Footer settings, the checksum is predefined for Russound. The End mark is 0xF7;



Number of Bytes for Footer F1: only checksum and end mark

Checksum F1: Russound

End mark F1: yes

Selection end mark F1: F7h

Figure 22: Example 3, Footer settings

- Bytes 6 to 20 are the command of this telegram. So these values are signed into the parameter:

Object 0 at value = 0	00700502020000F123000000000001
Object 0 at value = 1	00700502020000F123000100000001

Figure 23: Example 3, 1 Bit command

At this example the Bit-value 0 switches the area 1 off and the Bit-value 1 switches the area on.

## 6 Index

### 6.1 List of figures

Figure 1: Connector pin assignment.....	3
Figure 2: General settings.....	6
Figure 3: Header and Footer.....	7
Figure 4: KNX => RS232 (1 Bit).....	8
Figure 5: KNX => RS232 (1 Byte).....	8
Figure 6: RS232 => Bit-value.....	9
Figure 7: RS232 => Byte-value.....	9
Figure 8: RS232 => Text String.....	11
Figure 9: Macro functions.....	12
Figure 10: Convert RS232 Byte status into KNX Bit values.....	13
Figure 11: Example 1, general settings.....	14
Figure 12: Example 1 - Settings for Header.....	14
Figure 13: Example 1 - Setting for Footer.....	15
Figure 14: Example Macro 1, Function 1.....	17
Figure 15: Example Macro 1, Function 2.....	17
Figure 16: Example Macro 1, Function 3.....	17
Figure 17: Example 2, general functions.....	18
Figure 18: Example 2, 1-Bit command.....	18
Figure 19: Example 2, 1 Byte value.....	18
Figure 20: Example 3, general settings.....	19
Figure 21: Example 3, Header settings.....	19
Figure 22: Example 3, Footer settings.....	19
Figure 23: Example 3, 1 Bit command.....	20

### 6.2 List of tables

Table 1: Communication objects.....	4
Table 2: Default settings of the communication objects.....	5
Table 3: Telegram Example.....	14
Table 4: Telegram Example --> Byte value.....	15
Table 5: Example 1, Receiving status telegrams.....	16
Table 6: Example 3, Telegram Russound.....	19

## 7 Attachment

### 7.1 Statutory requirements

The above-described devices must not be used with devices, which serve directly or indirectly the purpose of human, health- or lifesaving. Further the devices must not be used if their usage can occur danger for humans, animals or material assets.

Do not let the packaging lying around careless, plastic foil/ -bags etc. can be a dangerous toy for kids.

### 7.2 Routine disposal

Do not throw the waste equipment in the household rubbish. The device contains electrical devices, which must be disposed as electronic scrap. The casing contains of recyclable synthetic material.

### 7.3 Assemblage



#### **Risk for life of electrical power!**

All activities on the device should only be done by an electrical specialist. The county specific regulations and the applicable EIB-directives have to be observed.

**MDT Multimedia Gateway, surface mounted**

Version		
SCN-MMG01.01	Multimedia Gateway	Surface mounted

The MDT Multimedia Gateway controls multimedia devices via RS232 connection.

- Database for T+A K8 series, Loewe TV and Russound multiroom systems
- General purpose for plain text and HEX data
- 1,5m connection cable

The MDT Multimedia Gateway is a flush mounted device for fixed installation in dry rooms.

For project design and commissioning of the Multimedia Gateway it is recommended to use the ETS3f/ETS4 or later. Please download the application software at [www.mdt.de/Downloads.html](http://www.mdt.de/Downloads.html)

SCN-MMG01.01



- Production in Germany, certified according to ISO 9001
- Database for **T+A K8 series, Loewe TV** and **Russound** multiroom systems
- **General purpose for plain text and HEX data**
- **Bidirectionally data transfer**
- 1,5m connection cable
- Integrated bus coupling unit
- 3 years warranty

<b>Technical Data</b>	SCN-MMG01.01
<b>Permitted wire gauge</b>	
KNX busconnection terminal	0,8mm Ø, solid core
<b>Supply voltage</b>	KNX Bus
<b>Power consumption KNX bus typ.</b>	< 0,3W
<b>Operation temperature range</b>	0 to + 45°C
<b>Enclosure</b>	IP 20
<b>Dimensions MDRC (Space Units)</b>	2SU

**Exemplary circuit diagram SCN-MMG01.01**

