Application Note AN1800 HA57 EVO User Manual

Exported on 16.08.2022

pei tel Communications GmbH
info@peitel.com
www.peitel.com

Table of Contents

0 History

Date	Revision	Author	Comments
August 2018	1.0	BP	First Release
January 2019	1.1	RaSc	Update of the first release
August 2022	1.2	RaSc	Update buzzer frequency formula

Table 1: History

0.1 Related Documents

No.	Name	Remarks
1	AN1902 Documentation API HA57 EVO	Download from www.peitel.de
2	AN1903 AppLoader HA57 SDK & HA57 EVO	Download from www.peitel.de
3	AN1904 LogoLoader for Handsets	Download from www.peitel.de

Table 2: Related Documents

1 Introduction

This document describes the features of the pei tel handset HA57 EVO. The hardware as well as the available terminal protocols HA20x and HA400 for the communication with the handset will be described in detail in this document.

The essential difference between the HA400 and HA20x protocol is the signalling of key events.

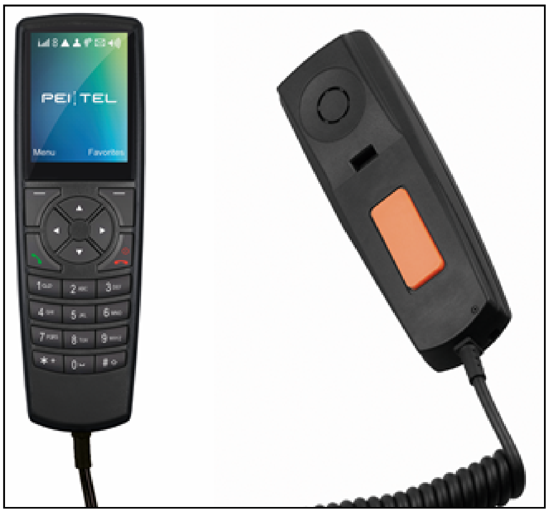


Figure 1: HA57 EVO

The adjacent figure is showing the upper side and bottom side of the handset HA57 EVO.

The handset consists of the following main components:

- Microphone & Earpiece
- Key panel with 20 keys
- Graphic 2.4'' colour LCD with background lighting
- Spiral cable with 10-pin Western connector
- PTT button (Push to Talk)

To connect the handset to a target system a 10-pin Western system connector is used. Besides the audio signals, a RS232 interface is available at the connector for communication with the target system.

In the remainder of this document, the controlling system is referred to as the **"Host"** or **"Host system"** and handset as the **"Client"**. To enable communication between the Host and the Client the so-called **"Terminal Interface"** has been defined. The Terminal Interface consists of the **"RS232 Interface"** (hardware, see 2.4 Communication Settings) and the **"Terminal Protocols"** (software, see 3 Firmware (Terminal Protocols) ff). These protocols are based on the Terminal Protocols HA20x and HA400 used in the handset HA87, HA88 and HA57 SDK for a downwards compatibility of the handsets. Additional commands have been defined to address the enhanced features of the handset HA57 EVO. The selection of the Terminal Protocol to be used can be done via the Setup Menu as described in section 4.4 Emulation.

The handset works as

- **Output device (transmitter)**
  When recognizing a pressed key, a respective key code from the chosen protocol is sent to the host system. This can initiate respective actions.
- **Input device (receiver)**
  The handset is controlled with control sequences by the host system. In this way, for example, a text can be displayed on the handsets display.

The following figure illustrates the application principle:

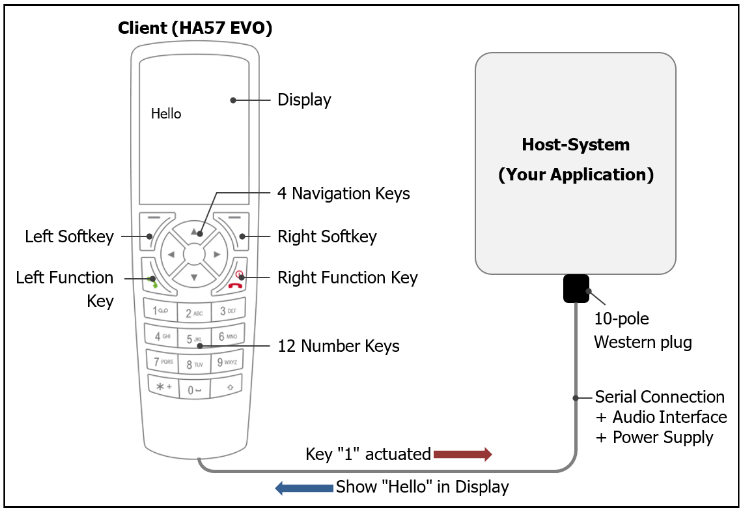


Figure 2: Handset usage principle

The HA57 EVO offers the possibility to integrate user applications. See also chapter 5 Own Software.

2 Overview Hardware Components

2.1 Performance Features

General:

- ARM 32-bit Cortex-M3 CPU (72 MHz frequency)
- HOOK detection via magnet (wear-free)
- Balanced operation of earpiece
- Balanced operation of microphone
- Electret microphone system
- Volume regulation for microphone and earpiece by command
- Continuously adjustable brightness adjustment

Hardware interfaces:

- UART 115200 V24 level (default)

Specifications*:

- 2.4'' colour graphic display (RGB TFT LCD) with 240x320 Pixel & 262,144 colours
- Power supply: 5 to 35 Volt
- Average power consumption: max. 0.8W/ ca. 160 mA at 5 V (lighting + audio switched on)
- Microphone amplification adjustable in 5 dB steps in a range of about 40 dB
- Microphone frequency range: 200 Hz to 7 kHz at -10 dB
- Microphone impedance: approx. 200 Ω
- Microphone sensitivity (in LRGP): 60 mV/Pa (-24 dBV/Pa ± 3 dB) (default)
- Earpiece amplification adjustable in 3 dB steps in a range of about 25 dB_{SPL}
- Earpiece frequency range: 200 Hz to 7 kHz at -10 dB
- Earpiece impedance: approx. 600 Ω
- Earpiece sensitivity: 25 dBPa/V ± 3 dB (default)
- Earpiece input voltage (max.): 80 mV
- Operating temperature range: -20 °C to +70 °C
- Dimensions: 164 mm x 51 mm x 29 mm (without cable)
- Spiral cable length: approx. 650 mm to 3000 mm (extended)
- Customer specific logo at start-up (Boot logo)
- Flash memory 446k (for SDK)
- SRAM memory 64k (for SDK)

* To keep pace with technical progress, we reserve the right to make improvements without prior notice.

2.2 Block Wiring Diagram

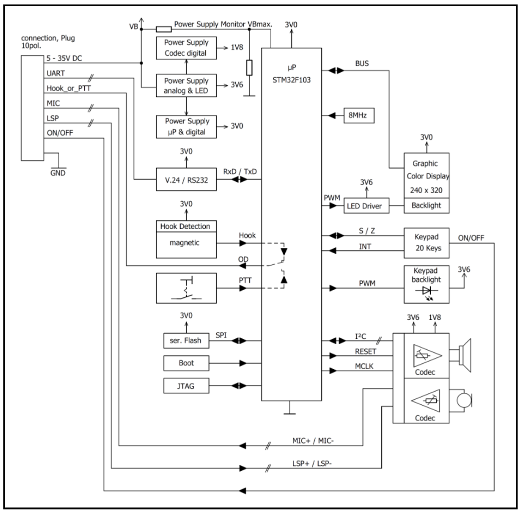


Figure 2: Block wiring diagram

2.3 Connector Pin Configuration

In the basic version, the 10-pin Western connector has the following pin configuration:

PIN	Signal	Level/Remarks
1	ON/OFF	Open-Drain (Right Function Key)
2	MIC -	See section 2.1 Performance Features
3	MIC +	
4	LS -	See section 2.1 Performance Features
5	LS +	
6	Hook or PTT	Open-Drain See section 4.5 Mapping of Pin 6 of the Connector
7	TxD	See section 2.1 Performance Features
8	RxD	
9	GND	Power Supply
10	VBATT (+5 … +35V)	

Table 3: System connector pin configuration

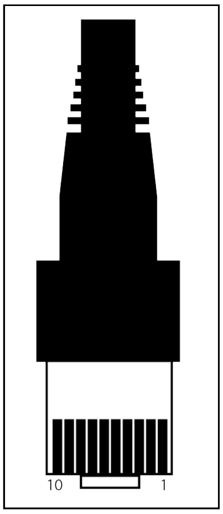


Figure 3: System connector

2.4 Communication Settings

The data stream from the Client (handset) to the Host (target system) is completely independent from the data stream from the Host to the Client (full duplex).

The common interface is a serial interface (RS232-V.24). The communication settings are:

Baud (default):	115200 Baud
Data bits:	8
Parity:	None
Stop bits:	1
Handshake:	None

Short notation: 115200,8,N,1

The hardware interface is part of the Terminal Interface.

3 Firmware (Terminal Protocols)

The Terminal Protocols are in addition to the hardware RS232 interface the second part of the Terminal Interface and allow the Host system to use the handset as a "Terminal", i.e. as an input and output device, controlled via a serial interface.

Die Terminal-Protokolle sind neben der hardwareseitigen RS232-Schnittstelle der zweite Teil der Terminalschnittstelle und erlauben dem Host-System, den Handapparat als "Terminal", d.h. Ein- und Ausgabegerät, über eine serielle Schnittstelle zu steuern.

For compatibility with the handsets HA87, HA88 and HA57 SDK both type of protocols used in these handsets - HA20x and HA400 - have been integrated in the new handset and can be selected via the setup program. See section 4.4 Emulation.

The essential difference between the HA20x and HA400 protocol is the signaling of key events. For example, with the HA20x protocol it is not possible to detect the start of the keystroke, however only one byte for signaling is required while with the HA400 protocol six bytes are required. Please see sections 3.1.1.1 Keypad Input and 3.2.1.1 Key Events for more details.

3.0 Display Organization

Both protocols use identical display organization which will be described prior to the protocol description.

3.0.1 General Organization of the Display

The display has a resolution of 240x320 (width, height) and a 565 RGB colour space (up to 256k colours via colour table). For normal operation, it is divided into three zones:

- Symbol Bar (Icons)
- Text Area (Text)
- Softkey Bar (SoftkeyL – Navigation – SoftkeyR)

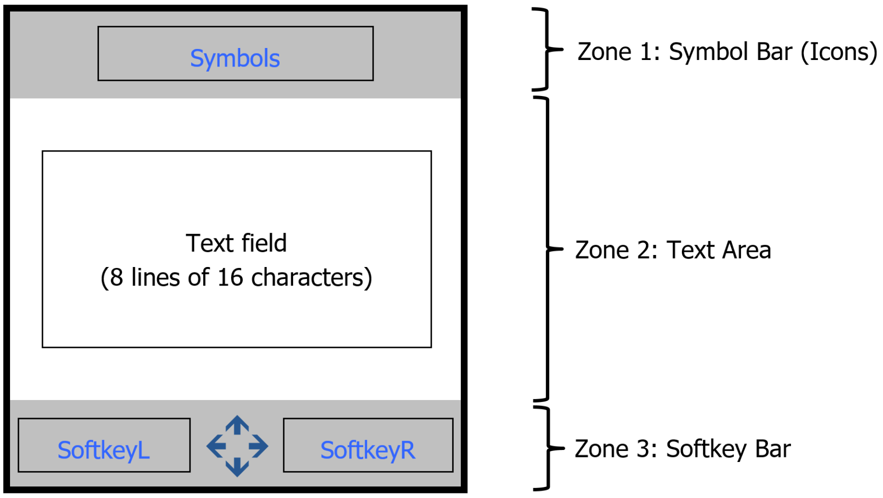


Figure 4: Organization of the display in NORMAL mode

In comparison to the versions HA87/88, all display modes that hide the Symbol Bar or the Softkey Bar for reasons of space limitations are not required anymore, as sufficient space is available for the text area.

A sub-division of the selection of display modes are the text modes that are only available for the text field. These text modes are pre-defined settings for the font used for display, the orientation and the matching of the given character codes (0-255) for a graphic symbol of the font (see also Code Table).

The text field, at normal usage (ASCII mode) consists of 8 rows with 16 columns (characters) each. Using alternative fonts will change the number of rows and columns. Rows and columns of the text field are free addressable (cursor concept) starting at 0 (upper left corner). Text is over-written row by row from left to right and accordingly downwards at the end of a row. In text mode, the overwritten characters are displayed immediately.

Non-overwritten parts of text keep the respective content. When switching the text mode, the text field is deleted and the cursor is positioned at home position (0).

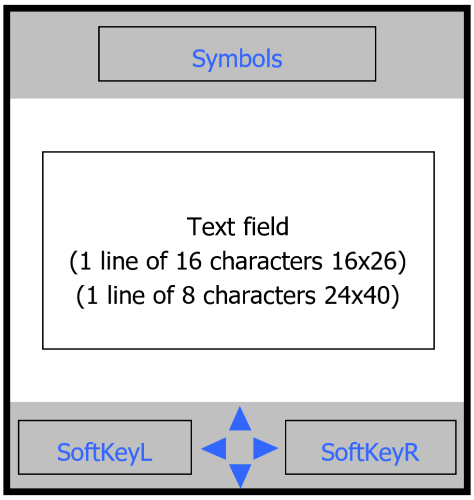

Symbols
Text field
(1 line of 16 characters 16x26)
(1 line of 8 characters 24x40)
SoftKeyL
SoftKeyR

Figure 5: Display organization in BIGSIZE mode

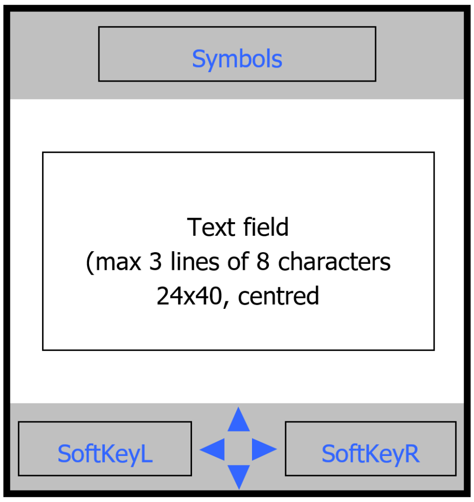


Figure 6: Display organization in CENTRE mode

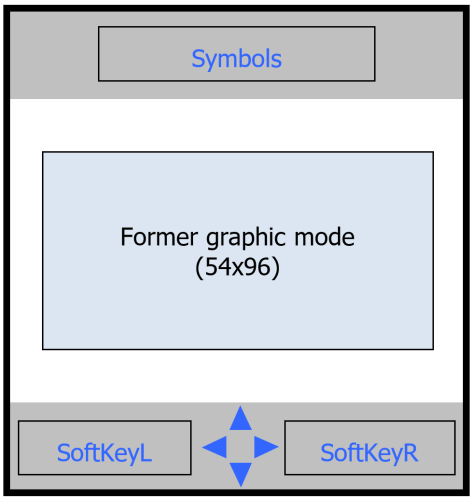


Figure 7: Display organization in GRAPHIC mode 54x96

Graphic modes can be activated independently of the text modes and relate to the total display area. The graphic memory reserved for these modes is filled up first, and then at activation, displayed once. "Graphic" only stands for small rectangles of a given graphic colour or of the text background colour.

The graphic mode was originally developed for a monochrome display and the display of pictograms and custom-made fonts. The graphic memory supports 120x160 pixels and is nearly compatible with the display memory of the HA87 (128x160) which is due to the geometry of the current display. The memory is logically divided into fields of 8x8 pixels each, which makes 20 rows and 15 columns. Depending on the chosen graphic mode, only distinctive rows of the display area are displayed. Therefore, the 3 display areas (symbol bar, text field and softkey bar) can be displayed independently, additionally to the graphic mode. A row of graphic content fills the display entirely from left to right and overwrites, where necessary, the existing content of the display fields.

For the terminal protocols (HA20x and HA400) graphic data are coded in a maximum 20 rows and are transferred in complete rows (all 15 columns).

It is also possible to use these graphic modes with the SDK, but the SDK also contains alternative functions for real graphics or for the integration of fonts.

3.0.2 Graphics Display Modes

The display, or parts of it, can be used for graphics. This section explains the graphic modes.

In general, a graphical mode can be imagined as transparencies that are put on top of the current display. The following figure shows the principle of the graphic-modes.

In this figure a combination of the display modes DISP_MODE_ASCII and DISP_MODE_GR_4ROWS is used. The graphic shaded grey (here the pei tel logo) is put on top of the current display, so that the resulting display will look like shown on the right-hand side.

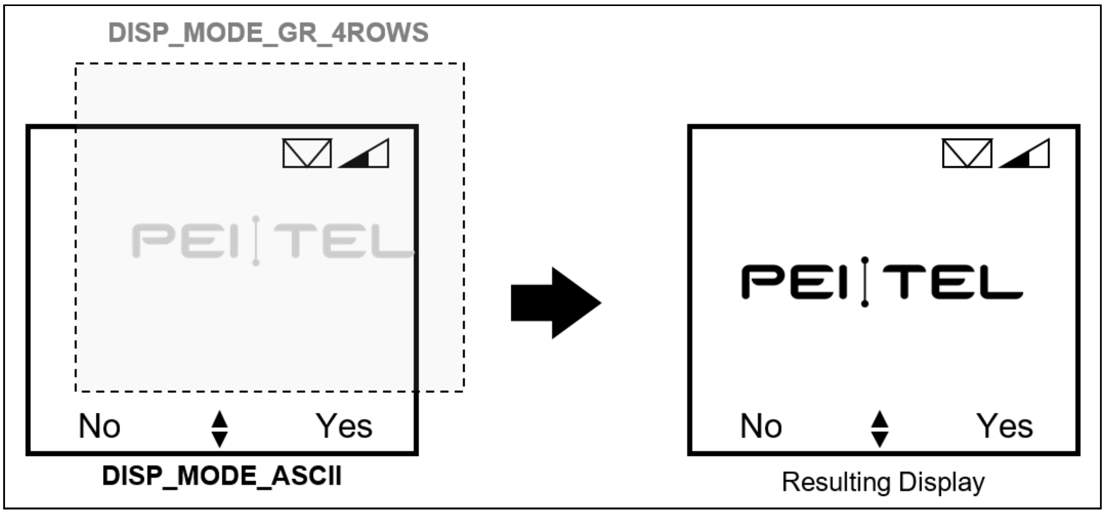


Figure 8: Principle of GRAPHIC modes of the HA57 EVO

The applied coding scheme can be found in the appendix 6.1 Coding Scheme of the Graphic Modes.

3.0.2.1 The Modes DISP_MODE_GR_1_8 to DISP_MODE_GR_17_20

A line consists of 120x8 pixels. Each pixel can be addressed and therefore activated. Depending on the selected graphic display mode, up to 20 lines can be displayed.

All together, there are 120x160 pixels that can be addressed in these modes.

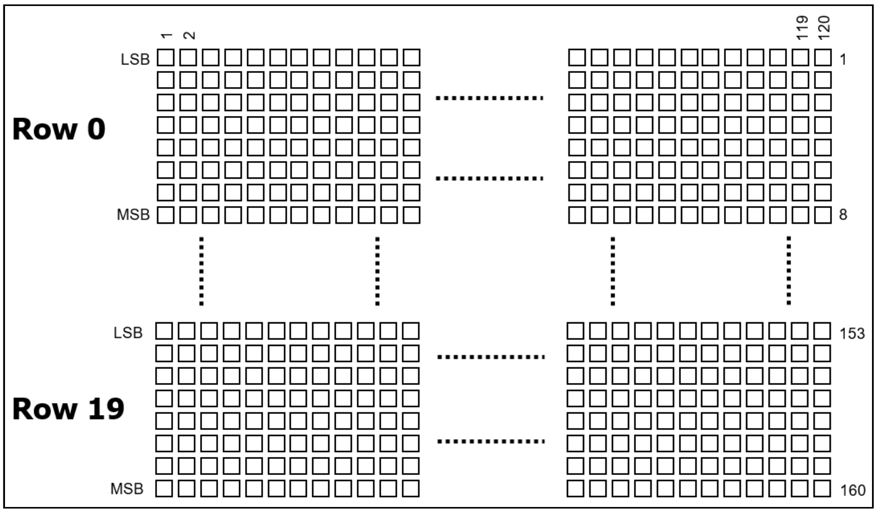


Figure 9: Display Structure 120x160

The size of the graphic area is 120x160 pixels. The graphic field is displayed on the display scaled by factor 2. The graphic mode of the HA57 SDK supported 16 columns. When a 16th column is transferred to the HA57 EVO, it is ignored. When choosing a graphic mode, additionally an 8-bit mask can be indicated, which refers to the rows to be activated in the respective mode. This allows to display any row between 0 and 19.

Line 0	**DISP_MODE_GR_1_8:** Line 0 to line 7 can be used for graphics purposes.
Line 1	
Line 2	
Line 3	
Line 4	
Line 5	
Line 6	
Line 7	
Line 8	**DISP_MODE_GR_9_16:** Line 8 to line 15 can be used for graphics purposes.
Line 9	
Line 10	
Line 11	
Line 12	
Line 13	
Line 14	
Line 15	
Line 16	**DISP_MODE_GR_17_20:** Line 16 to line 19 can be used for graphics purposes.
Line 17	
Line 18	
Line 19	

Figure 10: DISP_MODE_GR_5ROWS

3.0.2.2 The modes DISP_MODE_DISP_MODE_GR_xROW and GR_x

These modes assure the compatibility to the HA57 with a very small display field (width 96 pixels, height 54 pixels). This small graphic field is placed, when these graphic modes are selected, right in the middle of the display. Symbols and softkeys are accordingly not faded out. The modes 1 to 7 make the selection of the rows displayed in the field. For the new modes, this conducted by the bit mask.

> **Note**
>
> The graphic memory for row 0 is placed in the upper text field for these modes. There are at maximum 7 rows and 12 columns. Do not mix up the old modes with the new modes!

A short description of the available modes using the names from older documentations:

DISP_MODE_GR_4ROW:	Row 1-4 of the graphic memory are displayed (4 rows)
DISP_MODE_GR_5ROW:	Row 0-4 of the graphic memory are displayed (5 rows)
DISP_MODE_GR_6ROW:	Row 0-6 of the graphic memory are displayed (7 rows)
DISP_MODE_GR_7:	Row 1-6 of the graphic memory are displayed (7 rows)
DISP_MODE_GR_8:	Identical with DISP_MODE_GR_6ROW
DISP_MODE_GR_9:	Row 1-2 of the graphic memory are displayed (2 rows)
DISP_MODE_GR_10:	Row 3-4 of the graphic memory are displayed (2 rows)
DISP_MODE_GR_11:	Row 5-6 of the graphic memory are displayed (2 rows)

The position of the displayed rows is fixed on the display field. Therefore, an activation of the mode 9 - 11 altogether shows the same result as does the activation of the mode 7.

3.0.3 Symbol Bar

Predefined symbols (Icons) can be displayed on the symbol bar. These symbols are displayed in a special, also predefined position. The symbol bar is divided into 7 fields:

Signal Strength	Missed Calls	Roaming	Audio Mode	Microphone Mute	SMS	Volume

Figure 11: Position of certain Symbols on the Symbol Bar

3.0.4 Softkey Bar

The softkey bar is located in the lower part of the display. It is divided into 3 fields:

SoftkeyL	Navigation	SoftkeyR

Figure 12: Softkey Bar

These fields are assigned to keys of the keypad as shown below:

Field "SoftkeyL":	Key "Left Softkey"
Field "SoftkeyR":	Key "Right Softkey"
Field "Navigation":	Keys "Navigation UP", "Navigation DOWN", "Navigation LEFT", "Navigation RIGHT"

3.1 Terminal Protocol HA20x

3.1.0 Presentation Rules

In the following, the commands are presented with as much precision as possible. First, there is a general description of the command providing information about the command behavior. A "Syntax" paragraph describes the way to use the command. A "Parameter(s)" paragraph describes the required parameter value(s) where applicable. A "Reply" paragraph provides possible responses of a command, where applicable. Finally, an "Example" paragraph provides an example of the command. Figures are provided where necessary. The prefix "0x" is used for a hexadecimal number.

The following basic rules are applied:

- The prefix "0x" is used for a hexadecimal number
- Control-Characters (not printable characters) are placed within <>-brackets. For example, <ESC> equals the ESC-character, which is (0x1B). It is just one character! It is not "E" + "S" + "C"!
- Any other characters are ASCII coded characters
- Variable values are shown in ()-brackets
- Do not use spaces unless it is quoted
- <CR> represents "Carriage Return". The hexadecimal value of <CR> is 0x0D
- <LF> represents "Line Feed". The hexadecimal value of <LF> is 0x0A
- The prefix "//" is used for comments

Command Line

Commands are transmitted to the handset within a "frame". Commands always start with the <ESC>-Character (0x1Bh) and end with <CR><LF> (0x0Dh followed by 0x0Ah). <LF> (0x0Dh) is optional. In this document <CR> <LF> are used as termination.

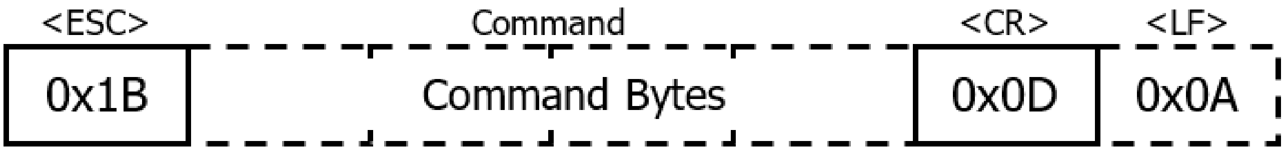


Figure 13: Frame of Terminal Protocol HA20x

There is no response from the handset to incorrect or formally correct commands. Only queries get a response.

Example:

<ESC>[(Row);(Column)H<CR><LF>

This example consists of the following parts:

<ESC>	ESCAPE character 0x1B
[	Character "["
(Row)	Variable value for "Row"
;	Character ";"
(Column)	Variable value for "Column"
H	Character "H"
<CR>	Carriage Return
<LF>	Line Feed

For this example, the following 8 bytes are actually needed to be transferred:

<ESC>	[	Row	;	Column	H	<CR>	<LF>
0x1B	0x5B	Row	0x3B	Column	0x48	0x0D	0x0A

3.1.1 Handset as Output Device (Transmitter)

The HA57 EVO sends answers to corresponding control commands and the input from the keypad.

3.1.1.1 Keypad Input

The distinction is made between three different modes when pressing a key:

- **Short-Key-Press-Mode (SKPM)**
  A key is pressed only for a "short" time. The key event is transmitted as soon as the key is released. In the transmitted key code, the eighth bit is set to "0".
- **Long-Key-Press-Mode (LKPM)**
  If a key remains pressed, a key code with a set eighth bit is transmitted to the host system (bit 8 = 1) after an adjustable time (*Time1*).
- **Autorepeat-Key-Press-Mode (AKPM)**
  If a key remains pressed, this is reported to the host system on expiry of the adjustable time (*Time2*) by means of a third key code. Pressing the key for longer than this time triggers this key event again on expiry of the time (*Time2*).

Keypad entries are transmitted by one byte.

Key	**SKPM**		**LKPM**	**AKPM**
	ASCII	Hex	Hex	Hex
Left Softkey	L	0x4C	0xCC	0xEC
Right Softkey	R	0x52	0xD2	0xE2
Left Function key	A	0x41	0xC1	n.a.
Right Function key	E	0x45	0xC5	n.a.
Navigation key UP	U	0x55	0xD5	0xE5
Navigation key DOWN	D	0x44	0xC4	0xE4
Navigation key LEFT	Y	0x59	0xD9	0xE9
Navigation key RIGHT	X	0x58	0xD8	0xE8
Key "1"	1	0x31	0xB1	0xF1
Key "2"	2	0x32	0xB2	0xF2
Key "3"	3	0x33	0xB3	0xF3
Key "4"	4	0x34	0xB4	0xF4

Key "5"	5	0x35	0xB5	0xF5
Key "6"	6	0x36	0xB6	0xF6
Key "7"	7	0x37	0xB7	0xF7
Key "8"	8	0x38	0xB8	0xF8
Key "9"	9	0x39	0xB9	0xF9
Key "0"	0	0x30	0xB0	0xF0
Key "*"	*	0x2A	0xAA	0xEA
Key "#"	#	0x23	0xA3	0xEB
Hook switch signal "HOOK OFF"	h	0x68	n.a.	n.a.
Hook switch signal "HOOK ON"	H	0x48	n.a.	n.a.
PTT (pressed)	Z	0x5A	n.a.	n.a.
PTT (released)	z	0x7A	n.a.	n.a.

Table 4: Key Codes of Terminal Protocol HA20x

Figure 14 shows the timings of the key codes sent for the "Left Softkey". If the key is released within *Time1* the SKPM key code 0x4C will be sent. If not released within *Time1*, the LKPM key code 0xCC will be sent after the time *Time1*. Note that SKPM key code is not sent, as the key wasn't released within time *Time1*! From now on there will be always the AKPM key code 0xEC sent after *Time2* has expired!

For the setting of the times *Time1* and *Time2* there is a special command **<ESC>IT(Time1);(Time2)<CR><LF>**) available. Please refer to corresponding section for more information about this command.

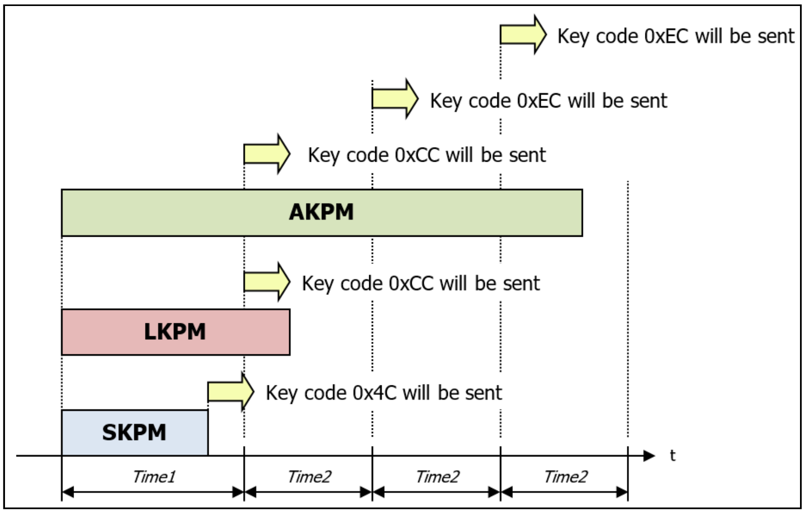

Key code 0xEC will be sent
Key code 0xEC will be sent
Key code 0xCC will be sent
AKPM
Key code 0xCC will be sent
LKPM
Key code 0x4C will be sent
SKPM
t
Time1
Time2
Time2
Time2

Figure 14: Key events HA20x

SKPM: Short-Key-Press-Mode

LKPM: Long-Key-Press-Mode

AKPM: Autorepeat-Key-Press-Mode

3.1.1.2 Power-On

After switching on, the HA57 EVO initialises itself internally according to the predefined configuration. If the HA57 EVO is started with the HA20x emulation, the sequence

<ESC>INIT<CR><CR><LF>

is sent on completion of the initialization. A connected device is thereby informed that the HA57 EVO is ready.

Attention

When the handset is connected to the power supply, the software version string is sent with Baudrate 115200. This is required for testing during production and **cannot** be turned off!

3.1.2 Handset as Input Device (Receiver)

In the direction towards the handset, the distinction is made between two types of data records. There are text messages that that are to be displayed in the display of the HA57 EVO, and control sequences that control the behaviour of the HA57 EVO. Both record types have in common that they are terminated by <CR> (\r; 0x0D) and <LF> (\n; 0x0A). The <LF> is optional.

Text messages

Text messages cannot begin with an <ESC> character. They are terminated with <CR> <LF>.

The text appears on the screen starting at the current cursor position.

When formatting text within a message only <LF> is approved. The following text on <LF> begins in the new row.

The previously selected character set is used to display the text.

Syntax

```
Text<CR><LF>
```

Parameter

```
None
```

Example

```
<ESC>ID0<CR><LF>

// Set display mode 0 (DISP_MODE_ASCII)

<ESC>[3;0H<CR><LF>

// Set cursor to position 3;0 (Row 3, Column 0)

Hello<CR><LF>

// Write text "Hello" in Line 3
```

Control commands

Control commands are initiated by an <ESC> (0x1B). All characters between <ESC> and the termination are passed to the command interpreter. A reaction is carried out only to commands requesting information. All other commands are not answered.

3.1.2.1 Display Mode HA20x

The display mode can be selected using this command (see also section 3.0 Display Organization). In addition, the display can be switched on or off. In non-graphic modes, the display mode should usually be selected **before** sending text to the display. In graphic modes, all data should be sent **first**, before selecting the display mode!

Syntax

```
<ESC>ID(Mode)<CR><LF>
```

Parameter

```
(Mode):
D           Switch off display
E           Switch on display

0,5         DISP_MODE_ASCII
1,8         DISP_MODE_SMS
2,4,6,7,9   DISP_MODE_TB
10          DISP_MODE_BIG_SIZE
11          DISP_MODE_GR_4ROW
12          DISP_MODE_GR_5ROW
13          DISP_MODE_GR_6ROW
14          DISP_MODE_GR_7
15          DISP_MODE_GR_8
16          DISP_MODE_GR_9
17          DISP_MODE_GR_10
18          DISP_MODE_GR_11
20          DISP_MODE_CENTER
21          DISP_MODE_UTF8

G0XX        DISP_MODE_GR_1_8        1st to 8th Line (Row)
G1XX        DISP_MODE_GR_9_16       9th to 16th Line (Row)
G2XX        DISP_MODE_GR_17_20      17th to 20th Line (Row)

XX          OR-operation of lines to be displayed as a hexadecimal number with:
01          1st Line of group
02          2nd Line of group
04          3rd Line of group
...
80          8th Line of group
03          1st and 2nd Line of group
FF          all 8 Lines of group
```

Example

```
(Text)
<ESC>ID0<CR><LF>

// Select display mode DISP_MODE_ASCII
```

```
(Graphic)
<ESC>IDG1FF<CR><LF>

// Select display mode DISP_MODE_GR_9_16
// Lines 8 to 15 are overlaid with graphic

(Graphic)
<ESC>IDG055<CR><LF>

// Select display mode DISP_MODE_GR_1_8
// Lines 0,2,4,6 are overlaid with graphic
```

3.1.2.2 Cursor ON

This command switches the cursor on. The cursor is only shown in the display modes DISP_MODE_ASCII, DISP_MODE_SMS and DISP_MODE_TB.

Syntax

```
<ESC>&C<CR><LF>
```

Parameter

```
None
```

Example

```
<ESC>&C<CR><LF>

// Switch on cursor
```

3.1.2.3 Cursor OFF

This command switches the cursor off.

Syntax

```
<ESC>&D<CR><LF>
```

Parameter

None

```
<ESC>&D<CR><LF>

// Switch off cursor
```

3.1.2.4 Set Cursor

This command allows the cursor to be placed at a specific position for the display modes DISP_MODE_ASCII, DISP_MODE_SMS, DISP_MODE_TB and DISP_MODE_UTF. Text is always written starting from the current cursor position. Note: This command does not switch the cursor on or off. In non-graphic modes, the cursor is set at the home (start) position when switching between two display modes.

```
<ESC>[(Row);(Column)H<CR><LF>
```

Parameter

```
(Row):      The line on which the cursor is to be set.
(Column):   The column in which the cursor is to be set. The value range is 0-15.
            If a column >15 is specified, the command will be ignored.
            Both parameters must be transmitted ASCII-coded, i.e. for column 15 the
following 2 bytes must be transmitted: 0x31 for "1" and 0x35 for "5"!
```

Example

```
<ESC>[3;15H<CR><LF>

// Set cursor in line 3 and column 15.
```

3.1.2.5 Clear Display

This command clears the text area of the display (and overwrites in the same time all activated graphics in this area), and the text memory and the cursor to home position. Properties like chosen text mode, reversed out text and colours are kept unchanged.

Syntax

```
<ESC>&#<CR><LF>
```

Parameter

None

Example

```
<ESC>&#<CR><LF>

// Clears the entire display
```

3.1.2.6 Clear Line

This command clears the text from the current cursor position to the end of this line. To clear a complete line, the cursor must first be set to the starting position of the line.

Syntax

```
<ESC>[K<CR><LF>
```

Parameter

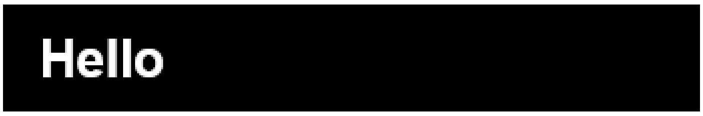
None

Example

```
<ESC>[3;0H<CR><LF>

// Set cursor to position 3;0 (Line 3, Column 0)

<ESC>[K<CR><LF>

// Clear line 3
```

3.1.2.7 Reversed Text Line

A line of text can be shown reversed out. Only one line at a time can be shown reversed out (highlighting effect). A reversed line of text looks as follows:



Hello

Syntax

```
<ESC>IZ(Line)<CR><LF>
```

Parameter

```
(Line):
0       Switch off reversing
1 - 8   Show line reversed out (1 corresponds to line 0).
        Only available in display modes:
        - DISP_MODE_ASCII
        - DISP_MODE_TB
        - DISP_MODE_SMS
```

Example

```
<ESC>&#<CR><LF>

// Clear display + Cursor to home position

Hello<CR><LF>

// Write "Hello" to display

<ESC>IZ1<CR><LF>

// Reverse out line1
```

3.1.2.8 Text of Left Softkey (SoftkeyL)

This command writes text to field "SoftkeyL". This field is assigned to the key "Left Softkey". A variable amount of characters (typically 8) is shown left justified. The actual amount of characters depends on the chosen font and the specific characters. The coding complies with the current text mode.

Syntax

```
<ESC>IK1(Text)<CR><LF>
```

Parameter

```
(Text):
Text to be displayed in field "SoftkeyL" (0-7 Characters)
```

Example

```
<ESC>IK1No<CR><LF>

// Writes "No" to field "SoftkeyL"
```

3.1.2.9 Text of Right Softkey (SoftkeyR)

This command writes text to field "SoftkeyR". This field is assigned to the key "Right Softkey". A variable amount of characters (typically 8) is shown left-justified. The actual amount of characters depends on the chosen font and the specific characters. The coding complies with the current text mode.

Syntax

```
<ESC>IK2(Text)<CR><LF>
```

Parameter

```
(Text):
Text to be displayed in field "SoftkeyR" (0-7 Characters)
```

Example

```
<ESC>IK2Yes<CR><LF>

// Writes "Yes" to field "SoftkeyR"
```

3.1.2.10 Clear Softkey

With this command, the contents of both Softkey fields is cleared.

Syntax

```
<ESC>IK(Key)<CR><LF>
```

Parameter

```
(Key):
0   Clear both left and right softkey fields
1   Clear left softkey field "SoftkeyL" only
2   Clear right softkey field "SoftkeyR" only
```

Example

```
<ESC>IK0<CR><LF>

// Clear both Softkey fields

<ESC>IK2<CR><LF>

// Clear right softkey field "SoftkeyR" only
```

3.1.2.11 Set Softkey Properties

This command specifies the properties of the softkey fields "SoftkeyL" and "SoftkeyR". If it is wished that a Softkey flashes then the flashing overwrites the respective graphic line with the Softkey content after short time.

Syntax

```
<ESC>IK3(Kx)(Mode)<CR><LF>
```

Parameter

```
(Kx):
1    SoftkeyL
2    SoftkeyR

(Mode):
0    Normal
1    Text flashes
```

Example

```
<ESC>IK311<CR><LF>

// SoftkeyL flashing text
```

3.1.2.12 Set Brightness

This command is used to set the brightness with the display switched on. The current value in percent of the brightness in relation to the maximum value [= 100%] can be calculated by using the formula: Brightness [in %] = Value x 5. The value of brightness is stored persistently in the device.

Syntax

```
<ESC>IA(Value)<CR><LF>
```

Parameter

```
(Value):
Value for brightness in the range 6 (dark) to 20 (bright).
Values < 6 are set to 6!
Brightness in relation to the maximum level [=100%] can be calculated by using the
formula: Brightness [in %] = Value x 5.
```

Example

```
<ESC>IA10<CR><LF>

// Set brightness to a medium level (50%)
```

3.1.2.13 Query Brightness

This command is used to get the current value of the brightness setting. The current value in percent of the brightness in relation to the maximum value [= 100 %] can be calculated by using the formula: Brightness [in %] = Returned Value x 5.

Syntax

```
<ESC>IA<CR><LF>
```

Parameter

```
Parameter:  None
```

Reply

```
The current setting of the brightness in the format:
<ESC>IA(Value)<CR>

(Value):
Value for brightness in the range 6 (dark) to 20 (bright).
The current value in percent of the brightness in relation to the maximum value [= 100%
] can be calculated by using the formula:

Brightness [in %] = Returned Value x 5.

Note: The returned value is ASCII coded. Values >9 require 2 bytes.
```

Example

```
<ESC>IA<CR><LF>

// Query current brightness setting
```

```
Possible Reply:

<ESC>IA10<CR>

// Brightness is set to 10 (medium level [= 50 %] in relation to the maximum level of 100 %)
```

3.1.2.14 Query Hook State

This command returns the current state of the hook switch. This is used to determine whether the handset is ON-HOOK or OFF-HOOK. The host system can then institute appropriate actions, such as enabling the audio path.

Syntax

```
<ESC>IH<CR><LF>
```

Parameter

None

Reply

```
The current state of the hook switch in the format:
<ESC>IH(State)<CR>

State):
h Handset is OFF-HOOK (lifted)
H Handset is ON-HOOK
```

Example

```
<ESC>IH<CR><LF>

// Query hook state

Possible Reply:

<ESC>IHh<CR>

// Handset is OFF-HOOK (lifted)
```

3.1.2.15 Query Version

Query the current software version of the Terminal Protocol.

Syntax

```
<ESC>&V<CR><LF>
```

Parameter

```
None
```

Reply

```
Current software version of Terminal Protocol in the format:

<ESC>&V(Version)<CR><LF>
```

Example

```
<ESC>&V<CR><LF>

// Query of the software version at the terminal interface

Possible Reply:

<ESC>&VHA57 V.01.00 11.07.2011<CR><LF>
```

3.1.2.16 Set Key Event Times

This command enables the individual setting of the two times *Time1* and *Time2* for the detection of key events (SKPM, LKPM and AKPM, see also section 3.1.1.1 Keypad Input). The values are stored permanently.

Syntax

```
<ESC>IT(Time1);(Time2)<CR><LF>
```

Parameter

```
(Time1):
Time1 in steps of 100 ms. Valid values are in the range from 4 (0.4 s) to 50 (5.0 s).
The standard value is 12 (1.2 s).

(Time2):
Time2 in steps of 100 ms. Valid values are in the range from 4 (0.4 s) to 50 (5.0 s).
The standard value is 12 (1.2 s). If Time2 is set to 0, the AKPM mode is switched off.
```

Example

```
<ESC>IT20;0<CR><LF>

// Set Time1 to 20 (2.0 s) and switch off AKPM mode
```

3.1.2.17 Backlight in On-Hook Mode

This command is used to configure the background lighting of the display and the keypad. 3 basic modes are possible:

- Permanently ON
- Permanently OFF
- Automatic

In "automatic" mode, the background lighting is activated automatically as soon as a key is pressed. The lighting goes off after an adjustable time. Each key-press restarts the timer. By means of a flag, the keypad lighting can be switched on permanently, while the display lighting goes off after the set time. In the two other modes, the lighting is either ON or OFF, regardless of whether a key is pressed.

```
<ESC>IE(M/D)<CR><LF>

// Set Mode or Duration
```

Parameter

```
(M/D): Lighting mode as follows:

0       Backlight is "Permanently OFF"
E       Backlight is "Permanently ON"
A       "Automatic" mode and restart of the timer, i.e. the lighting is switched on
immediately for the set period of time. Pressing a key also switches on the lighting
for the set period of time.
```

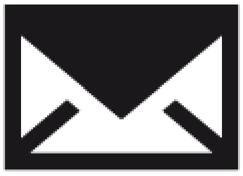
```
S        "Automatic" mode and no restart of the timer. Pressing a key switches on the
lighting for the set period of time.
R        Resets the switch-on time to the default value of 30 s and starts the
"Automatic" mode.
e        Sets the flag for permanent keypad lighting.
r        Clears the flag for permanent keypad lighting.
1...n    Setting of the switch-on time in steps of 1 s.
```

Examples

```
<ESC>IE0<CR><LF>

// Backlight is "Permanently OFF"

<ESC>IE3<CR><LF>

// Set the switch-on time to 3s

<ESC>IEA<CR><LF>

// Set "Automatic" mode and restart timer
```

3.1.2.18 "Read SMS" Symbol

This command switches the "Read SMS" symbol on or off. The "Read SMS" symbol is displayed in the Symbol Bar (see also section 3.0.3 Symbol Bar) in the field "SMS". A flashing symbol is always in the front and overwrites all display content in this area.



Syntax

```
<ESC>IS(State)<CR><LF>
```

Parameter

```
(State):
0        Deactivate symbol "Read SMS"
```

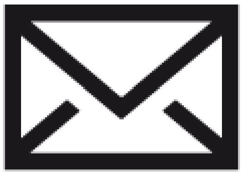
```
1       Activate symbol "Read SMS"
255     Activate flashing symbol "Read SMS"
```

Example

```
<ESC>IS1<CR><LF>

// Show symbol "Read SMS"
```

3.1.2.19 "Unread SMS" Symbol

This command switches the "Unread SMS" symbol on or off. The symbol can also be displayed flashing. A flashing symbol is always in the front and overwrites all display content in this area.



Syntax

```
Syntax:       <ESC>IU(State)<CR><LF>
```

Parameter

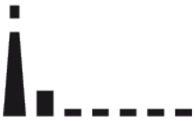
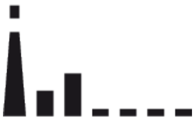
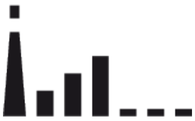
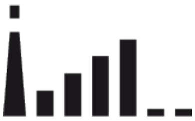
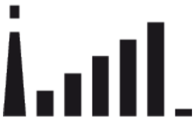
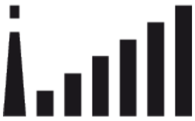
```
(State):
0       Deactivate symbol "Unread SMS"
1       Activate symbol "Unread SMS"
255     Activate flashing symbol "Unread SMS"
```

Example

```
<ESC>IU255<CR><LF>

// Show symbol "Unread SMS", flashing
```

3.1.2.20 "Signal Strength" Symbol

This command switches the symbol "Signal strength" on or off. Up to 6 signal bars can be used to display the signal strength.

					
1 signal bar	2 signal bars	3 signal bars	4 signal bars	5 signal bars	6 signal bars

Syntax

```
<ESC>IF(State)<CR><LF>
```

Parameter

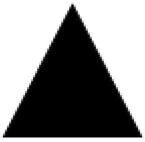
```
(State):
0        Deactivate symbol "Signal Strength"
1-6      Activate the symbol as follows:
         1:              Symbol with 1 signal bar
         6 and higher:   Symbol with 6 signal bars
```

Example

```
<ESC>IF3<CR><LF>

// Show symbol with 3 bars
```

3.1.2.21 "Roaming" Symbol

The symbol "Roaming" is located in the middle of the symbol bar and looks as follows:



Syntax

```
<ESC>IR(State)<CR><LF>
```

Parameter

```
(State):
0   Deactivate symbol "Roaming"
1   Activate symbol "Roaming"
```

Example

```
<ESC>IR1<CR><LF>

// Show symbol "Roaming"
```

3.1.2.22 "Missed Calls" Symbol

The symbol "Missed Calls" is represented only by a number in the symbol bar. For more than 9 calls a 9 is displayed.

Syntax

```
<ESC>IP(Value)<CR><LF>
```

Parameter

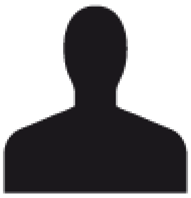
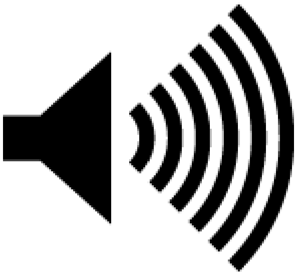
```
(Value):
0       Deactivate symbol "Missed Calls"
1-9     Activate symbol with the corresponding value
```

Example

```
<ESC>IP1<CR><LF>

// Show symbol with the value "1"
```

3.1.2.23 "Volume Private Mode" Symbol

The volume symbol is displayed in conjunction with the symbol "Private Mode". The potentiometer for volume control of the speaker is not adjusted. The value sent with this command only modifies display of the symbol.

	
Private Mode	Volume

Syntax

```
<ESC>IL(Value)<CR><LF>
```

Parameter

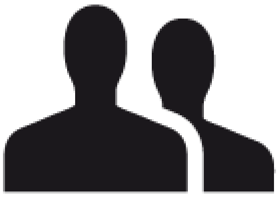
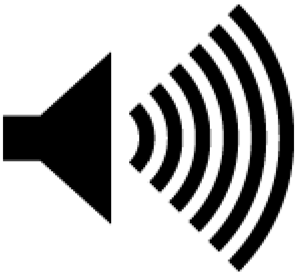
```
(Value):
0       Deactivate all volume symbols, the symbol for the private mode remains active
after activation.
1-10    Activate the symbol "volume private mode" with the corresponding value.
        1:              Show symbol with minimum value.
        10 or higher:   Show symbol with maximum value.
```

Example

```
<ESC>IL5<CR><LF>

// Show the symbol with an average value.
```

3.1.2.24 "Volume Handsfree Mode" Symbol

The volume symbol is displayed in conjunction with the symbol "Handsfree Mode". The potentiometer for volume control of the speaker is not adjusted. The value sent with this command only modifies display of the symbol.

	
Handsfree Mode	Volume

Syntax

```
<ESC>Il(Value)<CR><LF>

// (I + lower case L)
```

Parameter

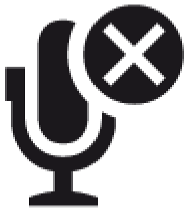
```
(Value):
0       Deactivate all volume symbols, the symbol for the hands-free mode remains
active after activation.
1-10    Activate the symbol "volume hands-free mode" with the corresponding value
        1:              Show symbol with minimum value.
        10 or higher:   Show symbol with maximum value.
```

Example

```
<ESC>Il5<CR><LF>

// Show the symbol with an average value. If the "Mute" symbol was (is) active, the
"Mute" symbol is displayed. The value can however be updated.
```

3.1.2.25 "Mute" Symbol

Only the presentation of the symbol is changed. Use the command 3.1.2.36 Microphone Mute for muting the microphone independently.


Mute

Syntax

```
<ESC>IM(State)<CR><LF>
```

Parameter

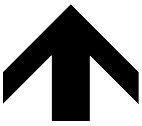
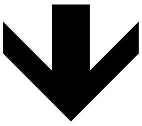
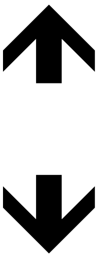
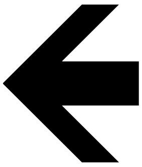
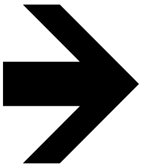
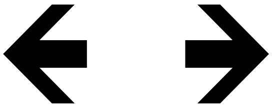
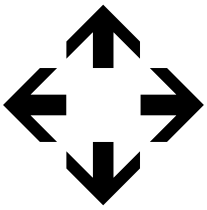
```
(State):
0   Deactivation of the "Mute" symbol.
1   Activation of the "Mute" symbol
```

Example

```
<ESC>IM1<CR><LF>

// Show symbol "Mute".
```

3.1.2.26 "Navigation" Symbols

The "Navigation" symbols are displayed in the field "Navigation" of the Softkey Bar (see also section 3.0.4 Softkey Bar) and represent the four navigation keys of the keypad. 6 symbols are implemented.

upwards	downwards	up- and downwards	left	right	left and right	All 4 directions

Syntax

```
<ESC>IW(State)<CR><LF>
```

Parameter

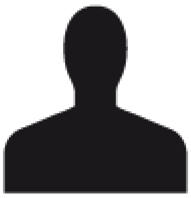
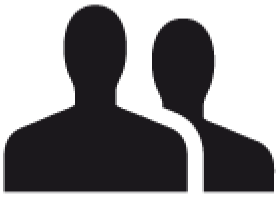
```
(State):
0   Deactivate all navigation symbols
1   Show symbol "Navigation downwards"
2   Show symbol "Navigation upwards"
3   Show symbol "Navigation up- and downwards"
4   Show symbol "Navigation left"
5   Show symbol "Navigation right"
6   Show symbol "Navigation left and right"
7   Show symbol "all 4 directions"
8   Show symbol "Navigation upwards and left"
9   Show symbol "Navigation upwards and right"
10  Show symbol "Navigation upwards, left and right"
11  Show symbol "Navigation downwards and left"
12  Show symbol "Navigation downwards and right"
13  Show symbol "Navigation downwards, left and right "
14  Show symbol "Navigation upwards, downwards and left"
15  Show symbol "Navigation upwards, downwards and right"
```

Example

```
<ESC>IW3<CR><LF>

// Show symbol "Navigation up- and downwards"
```

3.1.2.27 "Audiomode" Symbols

The "Audiomode" symbols are normally automatically switched on in combination with the "Volume" symbol. This command offers the possibility to switch it on independently and especially, to switch it off.

	
Private Mode	Handsfree Mode

Syntax

```
<ESC>IY(State)<CR><LF>
```

Parameter

```
(State):
0 Switch symbol off
1 Activate "Handsfree Mode" symbol
2 Activate "Privat Mode" symbol
```

Example

```
<ESC>IY2<CR><LF>

// Show "Privat Mode" symbol.
```

3.1.2.28 Loading Animation

The handset provides a loading animation.

Syntax

```
<ESC>IO(State)<CR><LF>
```

Parameter

```
(State):
0 Animation off
1 Animation on
```

Example

```
<ESC>IO1<CR><LF>

// Show a pulsating circle animation in the text area of the display.
```

3.1.2.29 Progress Bar

The handset provides an animation for the display of progress.

Syntax

```
<ESC>II(f)<CR><LF>
```

Parameter

```
(f):
Progress in percent ( 0 ... 100 )
```

Example

```
<ESC>II50<CR><LF>

// Shows the progress at 50 % in the text area of the display.
```

3.1.2.30 Set Fonts

Separate fonts can be set for the zones text area and softkeys. See also section 6.4 Fonts.

Syntax

```
<ESC>IDF(Area),(Font)<CR><LF>
```

Parameter

```
(Area):
0 Text area text size 8x12
1 reserved
2 Font for softkeys SoftkeyL + SoftkeyR

Font):
0 Font 15 x 26
1 Font 15 x 26B
2 Font 12 x 16
3 Font 12 x 16B

// Currently, for area 0, only font 0 and 1 are supported!
```

Example

```
<ESC>IDF2,1<CR><LF>

// Select Font 1 for the Softkey areas
```

3.1.2.31 Set Colours

For the 3 display areas, front (fonts, symbols, icons) and background colours can be chosen separately from the colour space RGB555. The colour selection is activated immediately and remains active until this command is used again. The colour selection for the graphic modes is not enabled until the graphic is requested and as well won't be applied to already displayed graphics. The background of graphics is always the background of the text area. After a reset, the colour setting of the setup will be valid. A permanent storage is not required, as the application will change the colours during run time again. The colour space is restricted to 5 Bits per colour value for compatibility reasons. For displays with higher resolution, the missing LSB (lowest significant bits) values are 0.

Syntax

```
<ESC>C(Area),(r),(g),(b)<CR><LF>
```

Parameter

```
(Area):
0 Font colour of the text field
1 Background colour of the text field
2 Symbols colour for the symbol bar (icons)
3 Background colour for the symbol bar
4 Font colour of the softkey fields
5 Background colour of the Softkey fields
6 Graphic

(r), (g), (b):
0..31 Colour in RGB format (colour space RGB555)
```

Example

```
<ESC>C0,0,0,0<CR><LF>

// Set font colour to BLACK for text area
```

3.1.2.32 Send & Show Graphic Data

With this command uncompressed or compressed data can be sent to the handset. The data is copied to the memory but not displayed yet

Note

The required display mode needs to be selected afterwards!

Hints:

1. It is recommended to transfer all rows of the graphic first, and then to select the required display mode.
2. Select only the graphic display modes you really need. For example, if you would like to put a graphic in row 1 and row 2, then only select DISP_MODE_GR_9 and not one of the others.

Syntax

```
<ESC>G(Row),(Data)z<CR><LF>
```

Parameter

```
(Row):
r0-r19 Line (row) where the data shall be displayed

(Data):
xxx Uncompressed or compressed data ASCII-coded
00                  Clear line (row)
```

Examples

```
<ESC>Gr0,00z<CR><LF>

// Clear Line 0

<ESC>Gr0,a00aFFa331B1F0Ez<CR><LF>

// Compressed data for a user defined character "P" in Line 0

<ESC>IDG001<CR><LF>

// Update display
```

3.1.2.33 Set Earpiece Volume

There are 9 steps available for adjusting the volume of the earpiece in the handset.

Syntax

```
<ESC>IV(Value)<CR><LF>
```

Parameter

```
(Value):         Sound Pressure Level SPL@10mV [dB SPL] as follows:
0         90.2
1         93.0
2         95.8
3         99.0
4         101.9
5         105.3
```

```
6         108.2
7         111.1
8         114.0

// If no parameter is specified, the value stored in Setup is applied.
// (Value) is limited to 8, default is 3.
```

Example

```
<ESC>IV4<CR><LF>

// Set Sound Pressure Level to 101,9 dB SPL
```

3.1.2.34 Set Microphone Gain

There are 10 steps available for adjusting the volume of the microphone in the handset.

Syntax

```
<ESC>IG(Value)<CR><LF>
```

Parameter

```
(Value):          Gain [in dB] as follows:
0         -38
1         -34
2         -29
3         -24
4         -19
5         -14
6         -10
7         4
8         0,8
9         0,8

// (Value) is limited to 9, default is 3. With the default setting, the sensitivity of
the microphone path is ~40 mVeff/Pa ±3dB (direction LRGP).
```

Example

```
<ESC>IG4<CR><LF>

// Set gain to -19 dB
```

3.1.2.35 Mute Earpiece

The amplifier of the earpiece can be deactivated. Turning off the amplifier will mute the earpiece.

Syntax

```
<ESC>M(mute)<CR><LF>
```

Parameter

```
(mute):
0         Earpiece active (no mute), amplifier turned on
1         Earpiece deactivated, (mute), amplifier turned off
```

Example

```
<ESC>M1<CR><LF>

// Turn off amplifier, earpiece muted
```

3.1.2.36 Mute Microphone

The amplifier of the microphone can be deactivated. This will mute the microphone. The "Mute" symbol will not be faded in or out!

Syntax

```
<ESC>N(mute)<CR><LF>
```

Parameter

```
(mute):
0   Microphone active (no mute), amplifier turned on
1   Microphone deactivated, (mute), amplifier turned off
```

Example

```
<ESC>N1<CR><LF>

// Turn off amplifier, microphone muted
```

3.1.2.37 Buzzer Function

This command activates the amplifier of the earpiece and can be used to play an external provided sound signal for a time in steps of 10 ms to simulate a buzzer function. The audio amplifier of the HA57 EVO is activated for the requested time. If the HOOK signal is activated (handset hung up), the volume is set to maximum level for the requested time. The sound needs to be provided externally over the audio path.

Syntax

```
<ESC>IB(Value)<CR><LF>
```

Parameter

```
(Value):

// Time to activate the amplifier of the earpiece in steps of 10 ms.
// Range is from 0 (short) to 255 (long).
```

Example

```
<ESC>IB10<CR><LF>

// External sound should be played for 100 ms.
```

3.1.2.38 Buzzer Tone

This command is used to activate a sound signal for a certain amount of time in steps of 10 ms. The sound output comes from the internal buzzer.

The buzzer is operated by a rectangle generator, whose frequency is calculated as follows:

freq = 101265 kHz / (scale + 1)

The value 0 for scale or time disables the sound. Due to the resonance and frequency behavior of the buzzer, the volume during frequency sweep (from about 200 Hz) will vary.

Syntax

```
<ESC>IB(time);(scale)<CR><LF>
```

Parameter

```
(time):
Amount of time (x 10 ms) the sound is required to be played.
Values above 10 seconds are limited to 10 seconds (1000).

(scale):
Pitch. A higher value will lower the tone. The value ranges from 0 to 255.
```

Example

```
<ESC>IB1;40<CR><LF>

// Generate a key tone (beep).
```

3.1.2.39 Scrolling Text

For animated representation of texts that are longer than one line, scrolling text can be displayed. The running direction is from right to left. A scrolling text on the display cannot be overwritten.

The text to be displayed should be placed in "" (quotation marks) to avoid misinterpretation of special characters. A text, which is shorter than the line length, is positioned centred in the display at the beginning.

The frequency indicates, in which period of time the text is moved by exactly one display pixel. The speed depends on the display resolution. To assure that the parameters gain a similar speed compared with the older devices, the minimum frequency was reduced from 10 to 5 ms.

Syntax

```
<ESC>F,(Freq),(Font),(Pos),(Text)<CR><LF>
```

Parameter

```
(Freq):
0       Text fixed
1..255  Frequency in steps of 5 ms)

Font):
Font and font size as follows:
0   Font size   15x26 (regular)
1   Font size   15x26B
2   Font size   24x40
3   Font size   12x16
4   Font size   12x16B

(Pos):
0..12   Line in which the scrolling text is shown

(Text): Text to be scrolled (max 100 characters)
```

Example

```
<ESC>F,3,2,10,"Sampletext"<CR><LF>

// Text in line 10 is scrolled every 15 ms one pixel to the left
```

3.1.2.40 Set Mode of Loudspeaker Input

This command is futile. It only exists for reasons of compatibility. The response is always "unbalanced".

Syntax

```
<ESC>L(s)<CR><LF>
```

Parameter

```
(s):
0   Balanced speaker mode (balanced input audio signal)
1   Unbalanced speaker mode (default; unbalanced input audio signal)
```

3.1.2.41 Get Mode of Loudspeaker Input

This command queries the current mode of the loudspeaker amplifier. The response is always "unbalanced".

Syntax

```
<ESC>L?<CR><LF>
```

Parameter

none

Reply

```
<ESC>L: 1<CR><LF>

(S):
1 Unbalanced
```

3.2 Terminal Protocol HA400

3.2.0 Presentation Rules

In the following, the commands are presented with as much precision as possible. First, there is a general description of the command providing information about the command behaviour. A "Syntax" paragraph describes the way to use the command. A "Parameter(s)" paragraph describes the required parameter value(s) where applicable. A "Reply" paragraph provides possible responses of a command, where applicable. Finally, an "Example" paragraph provides an example of the command. Figures are provided where necessary. The prefix "0x" is used for a hexadecimal number.

The following basic rules are applied:

- The prefix "0x" is used for a hexadecimal number

- Control-Characters (not printable characters) are placed within <>-brackets. For example, <ESC> equals the ESC-character which is 0x1Bh. It is just one character! It is not "E" + "S" + "C"!
- Any other characters are ASCII coded characters
- Variable values are shown in ()-brackets
- Do not use spaces unless it is quoted
- <CR> represents "Carriage Return". The hexadecimal value of <CR> is 0x0D
- <LF> represents "Line Feed". The hexadecimal value of <LF> is 0x0A
- The prefix "//" is used for comments

Command Line

The byte-oriented protocol uses some sort of a frame. Commands always start with the <ESC>-Character (0x1Bh) and end with <CR> (0x0Dh) or <CR><LF> (0x0Dh followed by 0x0Ah). In this document both <CR> and <LF> are used.

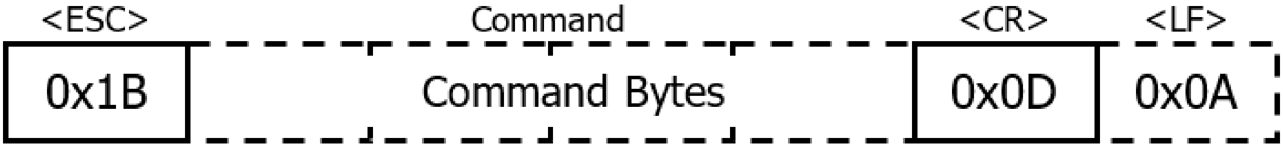


Figure 15: Frame of Terminal Protocol HA400

To keep data traffic as low as possible, only queries get a response. Please bear in mind that there is no response from the handset to incorrect or formally correct commands.

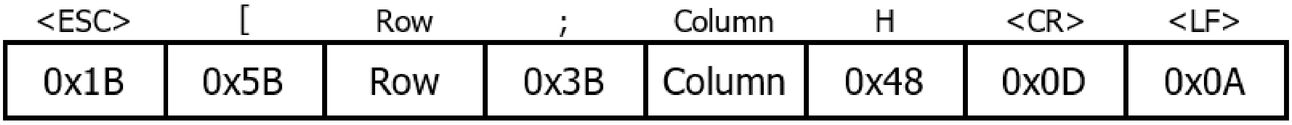
Example:

<ESC>[(Row);(Column)H<CR><LF>

This example consists of the following parts:

<ESC>	ESCAPE character 0x1B
[	Character "["
(Row)	Variable value for "Row"
;	Character ";"
(Column)	Variable value for "Column"
H	Character "H"
<CR>	Carriage Return
<LF>	Line Feed

For this example, the following 8 Bytes (real byte stream) are transferred:



<ESC>	[	Row	;	Column	H	<CR>	<LF>
0x1B	0x5B	Row	0x3B	Column	0x48	0x0D	0x0A

3.2.1.1 Key Events

The distinction is made between five different modes when pressing a key:

- **Start-Key-Event SKE**
  This event occurs, when a key is pressed the first time.
- **Long-Key-Event LKE**
  After an adjustable period of time (*Time1*) a still pressed key will result in a Long-Key-Event. Switches like hook or PTT do not have a Long-Key-Event.
- **Repeated-Key-Event RKE**
  After an also adjustable period of time (*Time2*) a still pressed key will result in a Repeated-Key-Event. Switches like hook or PTT do not have a Repeated-Key-Event.
- **End-Key-Event EKE**
  When a key is released, an End-Key-Event occurs.
- **False-Key-Event (FKE)**
  When a key-error is detected, a False-Key-Event occurs. For example, pressing two keys at the same time can be a False-Key-Event.

3.2.1.2 Key Event Messages

For each key event a key event message is sent via the serial link to notify the host system about the key event. The host might then initiate appropriate actions.

In general, the structure of such a message is as follows:

Syntax

```
<ESC>K(Key)(Event)<CR><LF>                (6 Bytes)
```

Parameter

```
(Key): Key code of the triggering key. The key code for the keys can be obtained from
Table 5.

(Event): The triggering event as follows:
s       Start-Key-Event
l       Long-Key-Event                    (lower case "L")
r       Repeated-Key-Event
e       End-Key-Event
```

```
F       False-Key-Event

The following messages are sent according to the key event:
Start-Key-Event-Message (SKEM):         <ESC>K(Key)s<CR><LF>
Long-Key-Event- Message (LKEM):         <ESC>K(Key)l<CR><LF>
Repeated-Key-Event- Message (RKEM):     <ESC>K(Key)r<CR><LF>
End-Key-Event- Message (EKEM):          <ESC>K(Key)e<CR><LF>
False-Key-Event- Message (FKEM):        <ESC>KFx<CR><LF>

Since it is not possible to assign a key code when 2 keys are pressed at the same
time, the False-Key-Event is an exception from the rule.
This case can be detected by examining the message for the "F" after the <ESC> and "K".
 Any message containing the "F" has to be considered as invalid!
```

Example

```
The following key event messages for key "1" will be sent to the host system:
Event       ASCII Stream            Hex Stream (real byte stream)
SKE         <ESC>K1s<CR><LF>        0x1B, 0x4B, 0x31, 0x73, 0x0D, 0x0A
LKE         <ESC>K1l<CR><LF>        0x1B, 0x4B, 0x31, 0x6C, 0x0D, 0x0A
RKE         <ESC>K1r<CR><LF>        0x1B, 0x4B, 0x31, 0x72, 0x0D, 0x0A
RKE         <ESC>K1r<CR><LF>        0x1B, 0x4B, 0x31, 0x72, 0x0D, 0x0A
EKE         <ESC>K1e<CR><LF>        0x1B, 0x4B, 0x31, 0x65, 0x0D, 0x0A
```

3.2.1.3 Key Codes

Keypad entries are transmitted according to the following table (ASCII coded):

Key	Code	SKEM	LKEM	RKEM	EKEM
Left Softkey	L	KLs	KLl	KLr	KLe
Right Softkey	R	KRs	KRl	KRr	KRe
Left Function key	A	KAs	KAl	KAr	KAe
Right Function key	E	KEs	KEl	KEr	KEe
Navigation key UP	U	KUs	KUl	KUr	KUe
Navigation key DOWN	D	KDs	KDl	KDr	KDe
Navigation key LEFT	Y	KYs	KYl	KYr	KYe

Key	Code	SKEM	LKEM	RKEM	EKEM
Navigation key RIGHT	X	KXs	KXl	KXr	KXe
Key "1"	1	K1s	K1l	K1r	K1e
Key "2"	2	K2s	K2l	K2r	K2e
Key "3"	3	K3s	K3l	K3r	K3e
Key "4"	4	K4s	K4l	K4r	K4e
Key "5"	5	K5s	K5l	K5r	K5e
Key "6"	6	K6s	K6l	K6r	K6e
Key "7"	7	K7s	K7l	K7r	K7e
Key "8"	8	K8s	K8l	K8r	K8e
Key "9"	9	K9s	K9l	K9r	K9e
Key "0"	0	K0s	K0l	K0r	K0e
Key "*"	*	K*s	K*l	K*r	K*e
Key "#"	#	K#s	K#l	K#r	K#e
Contact "HOOK"	H	KHs	n.a.	n.a.	KHe
Key "PTT"	P	KPs	n.a.	n.a.	KPe

Table 5: Key Codes of Terminal Protocol HA400

Remarks

- The rest position of the hook switch is "ON-HOOK"! So removing the Handset from the rest will cause the Start-Key-Event.
- The "IT" command allows the user to switch off the LKEM and RKEM messages, so that only the Start and end event will be transferred.

Examples:

The following timing diagram shows chronological process of the events and which are sent for pressing key "1" on the keypad.

When the key is pressed, the Start-Key-Event occurs and the corresponding SKEM message is sent. If the key is kept pressed, the Long-Key-Event occurs after the time *Time1* and the LKEM message is sent to the host system. If the key is still kept pressed, the Repeated-Key-Event occurs and will cause the Repeated-Key-Event-Message (RKEM) to be sent. Finally, when the key is released, the End-Key-Event will occur and the End-Key-Event-Message will be sent.

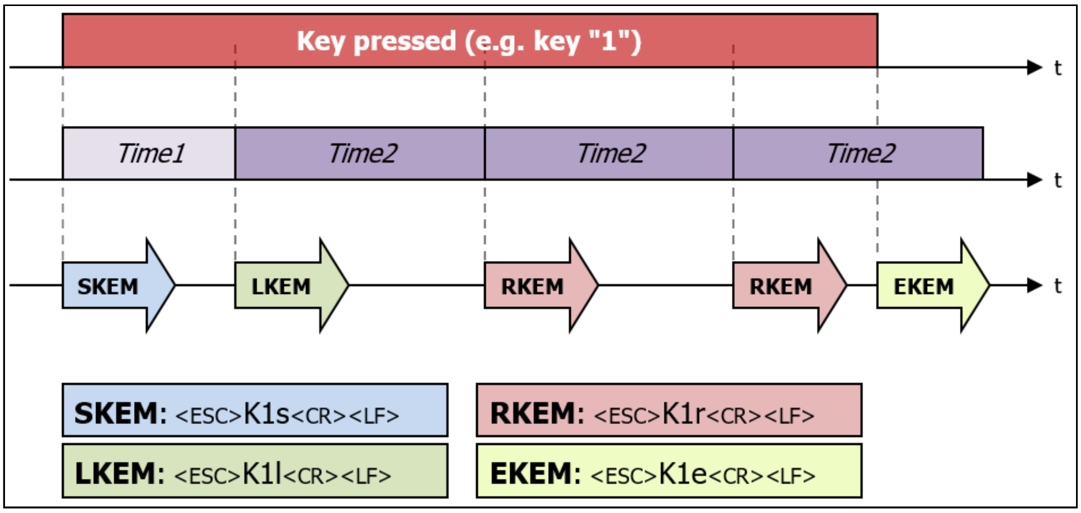


Figure 16: Timing diagram key events

In the following timing diagram shows a faulty key stroke, when a second key "2" is pressed while the key "1" is pressed. The period of time both keys are pressed, is considered to be invalid.

Please note that for key "1", no End-Key-Event message is sent. Each message, which contains an "F" is considered invalid. The next valid message is the Start-Key-Event message, after key "1" was released.

Please study the diagram shown next to get a better understanding of the processes.

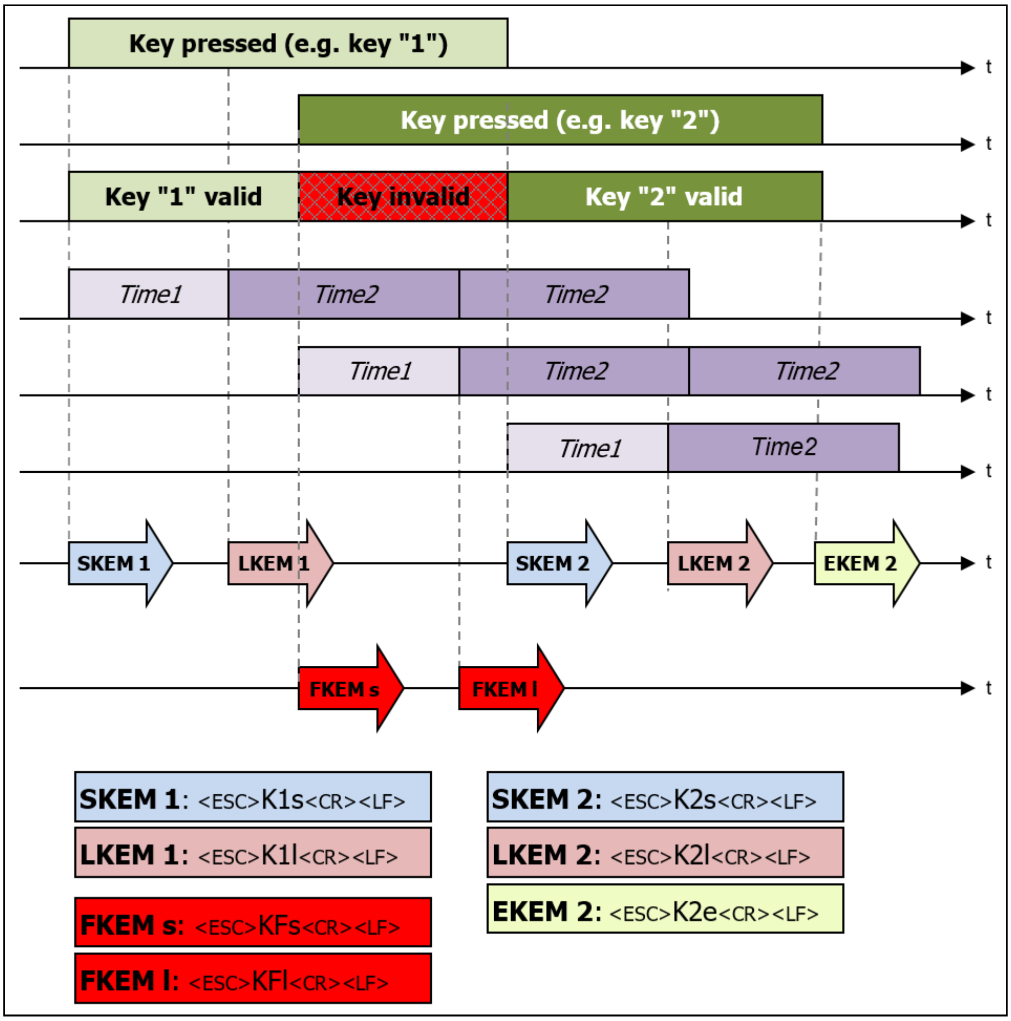


Figure 17: Timing diagram invalid key event

3.2.1.4 Power-On

When the handset is connected to power supply, the sequence

<ESC>INIT<CR><CR><LF>

is sent once using the selected baud rate. This sequence can be used to detect the moment, when the handset is connected to the host system.

Notice

When the handset is connected to the power supply, the software version string of the internal bootloader is sent via the serial interface with baud rate 115200.
This is required for testing during production and **cannot** be turned off!

3.2.2 Handset as Input Device (Receiver)

In the direction towards the handset, the distinction is made between two types of data records. There are text messages that that are to be displayed in the display of the HA57 EVO, and control sequences that control the behaviour of the HA57 EVO. Both record types have in common that they are terminated by <CR> (\r; 0x0D) and <LF> (\n; 0x0A). The <LF> is optional.

Text messages

Text messages cannot begin with an <ESC> character. They are terminated with <CR> <LF>.

The text appears on the screen starting at the current cursor position.

When formatting text within a message only <LF> is approved. The following text on <LF> begins in the new row.

The previously selected character set is used to display the text.

Syntax

```
Text<CR><LF>
```

Parameter

None

Example

```
<ESC>IDM2<CR><LF>

// Set display mode DISP_MODE_05_16

<ESC>[3;0H<CR><LF>

// Set cursor to position 3;0 (Row 3, Column)

Hello<CR><LF>

// Write text "Hello" in row 3
```

Control commands

Control commands are initiated by an <ESC> (0x1B). All characters between <ESC> and the termination are passed to the command interpreter. A reaction is carried out only to commands requesting information. All other commands are not answered. The syntax of the protocols HA20x and HA400 is very similar, so special attention has to be paid when implementing in the host system!

3.2.2.1 Display Mode HA400

The display mode can be selected using this command (see also section 3.0 Display Organization). In addition, the display can be switched on or off. In non-graphic modes, the display mode should usually be selected **before** sending text to the display. In graphic modes, all data should be sent **first**, before selecting the display mode!

Syntax

```
<ESC>IDM(Mode)<CR><LF>
```

Parameter

```
(Mode):
D       Switch off display
E       Switch on display
0       DISP_MODE_ASCII
1       DISP_MODE_TB
2       DISP_MODE_SMS
4       DISP_MODE_BIG_SIZE
5       DISP_MODE_GR_4ROWS
6       DISP_MODE_GR_5ROWS
7       DISP_MODE_GR_6ROWS
8       DISP_MODE_7
9       DISP_MODE_8
10      DISP_MODE_9
11      DISP_MODE_10
12      DISP_MODE_11
13      DISP_MODE_CENTER
21      DISP_MODE_UTF8

G0XX    DISP_MODE_GR_1_8        1st to 8th Line (Row)
G1XX    DISP_MODE_GR_9_16       9th to 16th Line (Row)
G2XX    DISP_MODE_GR_17_20      17th to 20th Line (Row)

XX      OR-operation of lines to be displayed as a hexadecimal number with:
01      1st Line of group
02      2nd Line of group
04      3rd Line of group
...
80      8th Line of group
03      1st and 2nd Line of group
FF      all 8 Lines of group
```

Example

```
(Text):
<ESC>IDM0<CR><LF>

// Set display mode DISP_MODE_ASCII

(Graphic):
<ESC>IDG1FF<CR><LF>

// Set display mode DISP_MODE_GR_9_16
// Lines 8 to 15 are overlayed with graphic

(Graphic):
<ESC>IDG055<CR><LF>

// Set display mode DISP_MODE_GR_1_8
// Lines 0, 2, 4, 6 are overlayed with graphic
```

3.2.2.2 Cursor ON/OFF

This command switches the cursor on. The cursor is only shown in the display modes DISP_MODE_04_16, DISP_MODE_04_16_SMS, DISP_MODE_05_16 and DISP_MODE_05_16_SMS.

Syntax

```
<ESC>&C(Mode)<CR><LF>
```

Parameter

```
(Mode):
0   Switch on cursor
1   Switch off cursor
```

Example

```
<ESC>&C1<CR><LF>

// Switch on cursor
```

3.2.2.3 Set Cursor

This command allows the cursor to be placed at a specific position for the display modes DISP_MODE_ASCII, DISP_MODE_SMS and DISP_MODE_TB. Text is always written starting from the current cursor position.

> **Note**
>
> This command does not switch the cursor on or off. In non-graphic modes, the cursor is set at the home (start) position when switching between two display modes.

Syntax

```
<ESC>&H(Row);(Column)<CR><LF>
```

Parameter

```
(Row):
The line on which the cursor is to be set. The value range depends on the chosen
display mode. In a 5 line display mode for example, the area ranges from 0 to 4.

(Column):
The column in which the cursor is to be set. The value range is 0-15.

Both parameters must be transmitted ASCII-coded, i.e. for column 15 the following 2
 bytes must be transmitted: 0x31 for "1" and 0x35 for "5"!
```

Example

```
<ESC>&H3;15<CR><LF>

// Set cursor in line 3 and column 15.
```

3.3.2.4 Query Cursor Position

This command returns the current position of the cursor.

Syntax

```
<ESC>&H?<CR><LF>
```

Parameter

```
None
```

Reply

```
Current cursor position in the format:

<ESC>&H:<SPACE>(Row);(Column)<CR><LF>

(Row):      Line (row) of current cursor position
(Column):   Column of current cursor position
```

Example

```
<ESC>&H?<CR><LF>

// Query cursor position

Possible Reply:

<ESC>&H: 3;15<CR><LF>

// Current cursor position is line 3 and column 15
```

3.2.2.5 Clear Display

This command clears the text area of the display (and overwrites therefore the activated graphics in this area), and the text memory and sets the cursor to home position. The chosen test mode, inversion and colours stay unchanged.

Syntax

```
<ESC>&D<CR><LF>
```

Parameter

```
None
```

Example

```
<ESC>&D<CR><LF>

// Clears the entire display
```

3.2.2.6 Clear Line

This command clears the text from the current cursor position to the end of this line. To clear a complete line, the cursor must first be set to the starting position of the line.

Syntax

```
<ESC>&K<CR><LF>
```

Parameter

None

Example

```
<ESC>&H3;0<CR><LF>

// Set cursor to position 3;0 (Line 3, Column 0)

<ESC>&K<CR><LF>

// Clear line 3
```

3.2.2.7 Reversed Text Line

A line of text can be shown reversed out. Only one line at a time can be shown reversed out (highlighting effect). A reversed line of text looks as follows:

Syntax

```
<ESC>IZ(Line)<CR><LF>
```

Parameter

```
(Line):
0       Switch off reversing
1 - 8   Show line reversed out (1 corresponds to line 0).
```

Example

```
<ESC>&D<CR><LF>

// Clear display + Cursor to home position

Hello<CR><LF>

// Write "Hello" to display

<ESC>IZ1<CR><LF>

// Reverse out line 1
```

3.2.2.8 Text of Left Softkey (SoftkeyL)

This command writes text to field "SoftkeyL". This field is assigned to the key "Left Softkey". A maximum of 8 (normally 7) characters is displayed, the real amount of the displayed characters depends on the configured font.

Syntax

```
<ESC>IK1(Text)<CR><LF>
```

Parameter

```
(Text):
Text to be displayed in field "SoftkeyL" (0-7 Characters)
```

Example

```
<ESC>IK1No<CR><LF>

// Writes "No" to field "SoftkeyL"
```

3.2.2.9 Text of Right Softkey (SoftkeyR)

This command writes text to field "SoftkeyR". This field is assigned to the key "Right Softkey". A maximum of 8 (normally 7) characters is displayed, the real amount of the displayed characters depends on the configured font.

Syntax

```
<ESC>IK2(Text)<CR><LF>
```

Parameter

```
(Text):
Text to be displayed in field "SoftkeyR" (0-7 Characters)
```

Example

```
<ESC>IK2Yes<CR><LF>

// Writes "Yes" to field "SoftkeyR"
```

3.2.2.10 Clear Softkey

With this command, both fields of the softkeys "SoftkeyL" and "SoftkeyR" can be cleared.

Syntax

```
<ESC>IK(Key)<CR><LF>
```

Parameter

```
(Key):
0   Clear both left and right softkey fields
1   Clear left softkey field "SoftkeyL" only
2   Clear right softkey field "SoftkeyR" only
```

Examples

```
<ESC>IK0<CR><LF>

// Clear both left and right softkey fields

<ESC>IK2<CR><LF>

// Clear right softkey field "SoftkeyR" only
```

3.2.2.11 Set Softkey Properties

Specifies the properties of the softkey fields "SoftkeyL" and "SoftkeyR". When a softkey flashes, the flashing overwrites the respective graphic lines again with the softkey content after a short amount of time.

Syntax

```
<ESC>IK3(Kx)(Mode)<CR><LF>
```

Parameter

```
(Kx):
1   SoftkeyL
2   SoftkeyR

(Mode):
0   Normal
1   Text flashes
```

Example

```
<ESC>IK311<CR><LF>

// SoftkeyL flashing text
```

3.2.2.12 Set Brightness

The brightness value can be set within the range of 30 % to 100 %. The higher the value, the higher is the brightness of the backlight. Parameters outside the allowed range are automatically corrected. The brightness value is stored permanently.

Syntax

```
<ESC>IN(Value)<CR><LF>
```

Parameter

```
(Value):
Value for brightness in the range 30 (dark) to 100 (bright).
```

Example

```
<ESC>IN50<CR><LF>

// Set brightness to a medium level (50 %). The setting will be permanently saved for
the state "display ON"
```

3.2.2.13 Increase Brightness

This command increases the brightness by 1 %. When the maximum permitted value of 100 [= 100 %] has been reached, a "?" will be replied. The new value is not stored permanently.

Syntax

```
<ESC>IN+<CR><LF>
```

Parameter

```
None
```

Example

```
<ESC>IN+<CR><LF>

// Increase brightness by 1 step
```

3.2.2.14 Decrease Brightness

This command decreases the brightness by 1 %. When the minimum permitted value of 30 [= 30 %] has been reached, a "?" will be replied. The new value is not stored permanently.

Syntax

```
<ESC>IN-<CR><LF>
```

Parameter

```
Parameter:  None
```

Example

```
<ESC>IN-<CR><LF>

// Decrease brightness by 1 step
```

3.2.2.15 Query Brightness

This command queries the current value of brightness in percent of the maximum brightness.

Syntax

```
Syntax:        <ESC>IN?<CR><LF>
```

Parameter

```
None
```

Reply

```
Current value of brightness in the format:

<ESC>IN:<SPACE>(Value)<CR>

(Value):
Brightness in percent
```

Example

```
<ESC>IN?<CR><LF>

// Query current setting of brightness

Possible Reply:

<ESC>IN: 50<CR>

// Current value is 50 %
```

3.2.2.16 Query Hook State

This command returns the current state of the hook switch. This is used to determine whether the handset is ON-HOOK or OFF-HOOK. The host system can then institute appropriate actions, such as enabling the audio path.

Syntax

```
<ESC>KH?<CR><LF>
```

Parameter

```
None
```

Reply

```
The current state of the hook switch in the format:

<ESC>KH:<SPACE>(State)<CR><LF>

(State):
h   Handset is OFF-HOOK (lifted)
H   Handset is ON-HOOK
```

Example

```
<ESC>KH?<CR><LF>

// Query hook state

Possible Reply:

<ESC>KH: h<CR><LF>

// Handset is OFF-HOOK (lifted)
```

3.2.2.17 Query PTT State

This command returns the current state of the PTT switch.

Syntax

```
<ESC>KP?<CR><LF>
```

Parameter

None

Reply

```
Current state of PTT switch in the format:

<ESC>KP:<SPACE>(State)<CR><LF>

(State):
p   PTT button is currently not pressed or not existing
P   PTT button is currently pressed
```

Example

```
<ESC>KP?<CR><LF>

// Query PTT state

Possible Reply:

<ESC>KP: p<CR><LF>

// PTT button is currently not pressed
```

3.2.2.18 Query Version

Query the current software version of the Terminal Interface.

Syntax

```
<ESC>&V?<CR><LF>
```

Parameter

None

Reply

```
Current software version of Terminal Interface in the format:

<ESC>&V:<SPACE>(Version)<CR><LF>
```

Example

```
<ESC>&V?<CR><LF>

// Query software version of the Terminal Interface

Possible reply:

<ESC>&V: HA57 V.01.00 11.07.2011<CR><LF>
```

3.2.2.19 Set Key Event Times

This command enables the individual setting of the two times *Time1* and *Time2* for the detection of key events (SKE, LKE, RKE and EKE, see also section). The values are permanently stored.

Syntax

```
<ESC>IT(Time1);(Time2)<CR><LF>
```

Parameter

```
(Time1):
Time1 in steps of 100 ms. Valid values are in the range from 4 (0.4 s) to 50 (5.0 s).
The standard value is 12 (1.2 s).

(Time2):
Time2 in steps of 100 ms. Valid values are in the range from 4 (0.4 s) to 50 (5.0 s).
The standard value is 12 (1.2 s).

If Time2 is set to 0, then the Repeated-Key-Event-Message (RKEM) is switched off.
If Time1 and Time2 are set to 0, Long-Key-Event-Message (LKEM) and Repeated-Key-Event-
Message (RKEM) are switched off.
Only Start-Key-Event-Message (SKEM) and End-Key-Event-Message (EKEM) are enabled.
```

```
<ESC>IT0;0<CR><LF>
```

```
// Set Time1 and Time2 to 0 and disable LKEM and RKEM messages
```

3.2.2.20 Query Key Event Times

This command queries the current settings of the key event times *Time1* and *Time2*.

```
<ESC>IT?<CR><LF>
```

Parameter

```
None
```

Reply

```
Current settings of key event times Time1 and Time2 in the format:

<ESC>IT:<SPACE>(Time1);(Time2)<CR><LF>

(Time1):
Time, when Long-Key-Event (LKE) will occur, in steps of 100 ms

(Time2):
Time, when Repeated-Key-Event (RKE) will occur, in steps of 100 ms
```

Example

```
<ESC>IT?<CR><LF>

// Query key event times

Possible Reply:

<ESC>IT: 0;0<CR><LF>

// LKEM and RKEM are disabled
```

3.2.2.21 Set Backlight Mode and Duration

This command is used to configure the backlight of the display and the keypad of the handset. There are 5 modes available:

Normal Mode (automatic)	Mode 0
Normal Mode (permanent)	Mode 1
PWM Mode (automatic)	Mode 2
PWM Mode (permanent)	Mode 3
Permanent off Mode	Mode 4

In PWM modes, the brightness is adjustable by the IN command. In Normal modes the brightness cannot be adjusted.

Automatic means that the backlight is switched on when a key is pressed. After an adjustable period of time, the backlight is switched off automatically. With every key press the backlight timer is restarted. It is also possible that the backlight of the keypad remains turned on and just the backlight of the display is turned off after the period of time has expired. This can be achieved by setting a flag in Normal Mode (automatic), Mode 0. Permanently means, that the backlight remains on all the time.

At delivery, the following settings are applied:

- Automatic
- Time 30 sec
- Brightness 60 %

Syntax

```
<ESC>IE(M/D)<CR><LF>

// Set Mode and Duration
```

Parameter

```
(M/D):
Backlight mode. The following parameters are supported:
0       Backlight is permanently OFF (Mode 4)
E       Backlight is permanently ON (Mode 1)
A       Automatic mode and activation of backlight for the duration as stored in
EEPROM (Mode 0)
S       Automatic mode, but backlight is not turned on (Mode 0)
```

```
P       Backlight is switched on permanently in PWM Mode. The brightness can be set
with the IN command (Mode3).
R       Restarts the backlight with the selected parameters (Mode and Duration).
e       Sets the flag for the permanent backlight of the keypad. Can only be set in
mode 0. Is not a mode, just sets a flag.
r       Clears the flag for the permanent backlight of the keypad and restarts the
backlight. Is not a Mode, clears just a flag.
s       Wie r, aber Beleuchtung wird nicht eingeschaltet
1..250  Duration of backlight in steps of 1s in the range from 1 to 250.
```

Examples

```
<ESC>IE0<CR><LF>

// Turn off backlight permanently

<ESC>IE10<CR><LF>

// Set duration of backlight to 10 s
```

3.2.2.22 Query Backlight Mode and Duration

Command to retrieve the current backlight mode and duration.

Syntax

```
<ESC>IE?<CR><LF>
```

Parameter

None

Reply

```
Current backlight mode and duration in the format:

<ESC>IE:<SPACE>(Mode);(Duration)<CR><LF>

(Mode):
0   Normal Mode (automatic)
1   Normal Mode (permanent)
```

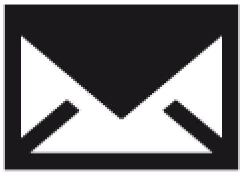
```
2   PWM Mode (automatic)
3   PWM Mode (permanent)
4   Permanent off Mode

(Duration):

Duration in seconds
```

Example

```
<ESC>IE?<CR><LF>

// Query backlight mode and duration

Possible Reply:

<ESC>IE: 2;15<CR>

// PWM Mode (automatic), Mode 2, 15 seconds
```

3.2.2.23 " Read SMS " Symbol

This command switches the "Read SMS" symbol on or off. The "Read SMS" symbol is displayed in the Symbol Bar (see also section 3.0.3 Symbol Bar) in the field "SMS". A flashing symbol overwrites graphic lines after a short amount of time.



Syntax

```
<ESC>IS(State)<CR><LF>
```

Parameter

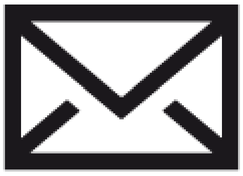
```
(State):
0       Deactivate symbol "Read SMS"
1       Activate symbol "Read SMS"
255     Activate flashing symbol "Read SMS"
```

Example

```
<ESC>IS1<CR><LF>

// Show symbol "Read SMS"
```

3.2.2.24 "Unread SMS" Symbol

This command switches the "Unread SMS" symbol on or off. The symbol can also be displayed flashing. A flashing symbol overwrites graphic lines after a short amount of time.



Syntax

```
<ESC>IU(State)<CR><LF>
```

Parameter

```
(State):

0       Deactivate symbol "Unread SMS"
1       Activate symbol "Unread SMS"
255     Activate flashing symbol "Unread SMS"
```

Example

```
<ESC>IU255<CR><LF>

// Show symbol "Unread SMS", flashing
```

3.2.2.25 "Signal Strength" Symbol

This command switches the symbol "Signal strength" on or off. Up to 6 signal bars can be used to display the signal strength. It is placed in the symbol bar in the field "Signal Strength".

1 signal bar	2 signal bars	3 signal bars	4 signal bars	5 signal bars	6 signal bars

Syntax

```
<ESC>IF(State)<CR><LF>
```

Parameter

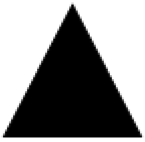
```
(State):
0        Deactivate symbol "Signal Strength"
1-6      Activate the symbol as follows:
         1:              Symbol without signal bar
         6 or higher:    Symbol with 6 signal bars
```

Example

```
<ESC>IF3<CR><LF>

// Show symbol with 3 bars
```

3.2.2.26 "Roaming" Symbol

The "Roaming" symbol is located in the middle of the symbol bar and looks as follows:



Syntax

```
<ESC>IR(State)<CR><LF>
```

Parameter

```
(State):
0   Deactivate symbol "Roaming"
1   Activate symbol "Roaming"
```

Example

```
<ESC>IR1<CR><LF>

// Show symbol "Roaming"
```

3.2.2.27 "Missed Calls" Symbol

The symbol "Missed Calls" is represented only by a number in the symbol bar.

More than 9 calls are represented by the figure 9.

Syntax

```
<ESC>IP(Value)<CR><LF>
```

Parameter

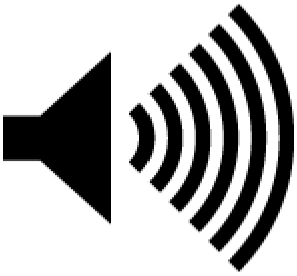
```
(Value):

0       Deactivate symbol "Missed Calls"
1-9     Activate symbol with the corresponding value
```

Example

```
<ESC>IP1<CR><LF>

// Show symbol with the value "1"
```

3.2.2.28 "Volume Private Mode" Symbol

The volume symbol is displayed in conjunction with the symbol "Private Mode". The potentiometer for volume control of the speaker is not adjusted. The value sent with this command only modifies display of the symbol.

	
Private Mode	Volume

Syntax

```
<ESC>IL(Value)<CR><LF>
```

Parameter

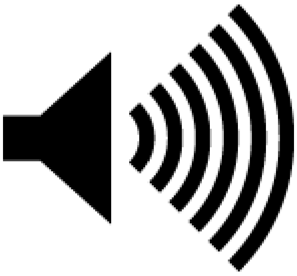
```
(Value):
0       Deactivation of all volume symbols
1-10    Activation of the symbol "Volume Private Mode" with the corresponding value.
        1:              Show symbol with min. value
        10 or higher:   Show symbol with max. value
```

Example

```
<ESC>IL5<CR><LF>

// Show the symbol with an average value
```

3.2.2.29 "Volume Handsfree Mode" Symbol

The volume symbol is displayed in conjunction with the symbol "Handsfree Mode". The potentiometer for volume control of the speaker is not adjusted. The value sent with this command only modifies display of the symbol.

	
Handsfree Mode	Volume

Syntax

```
<ESC>IJ(Value)<CR><LF>
```

Parameter

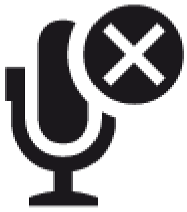
```
(Value):
0       Deactivation of all volume symbols
1-10    Activation of the symbol "Volume Handsfree Mode" with the corresponding value.
        1:              Show symbol with min. value
        10 or higher:   Show symbol with min. value
```

Example

```
<ESC>IJ5<CR><LF>

// Show the symbol with an average value
```

3.2.2.30 "Mute" Symbol

Use command 3.2.2.53 Mute Microphone for muting the microphone independently.


Mute

Syntax

```
<ESC>IM(State)<CR><LF>
```

Parameter

```
(State):
0   Deactivation of the symbol "Mute".
1   Activation of the symbol "Mute".
```

Example

```
<ESC>IM1<CR><LF>

// Show symbol "Mute".
```

3.2.2.31 "Navigation" Symbol

The "Navigation" symbols are displayed in the field "Navigation" of the Softkey Bar (see also section 3.0.4 Softkey Bar) and represent the four navigation keys of the keypad. 6 symbols are implemented.

↑	↓	↑ ↓	←	→	← →	✥
upwards	downwards	up- and downwards	left	right	left and right	All 4 directions

Syntax

```
<ESC>IW(State)<CR><LF>
```

Parameter

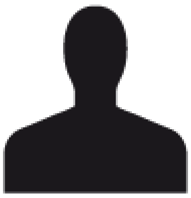
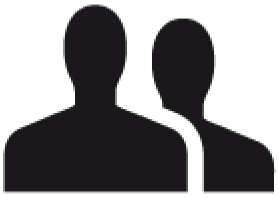
```
(State):
0   Deactivate all navigation symbols
1   Show symbol "Navigation downwards"
2   Show symbol "Navigation upwards"
3   Show symbol "Navigation up- and downwards"
4   Show symbol "Navigation left"
5   Show symbol "Navigation right"
6   Show symbol "Navigation left and right"
7   Show symbol "All 4 directions"
8   Show symbol "Navigation upwards and left"
9   Show symbol "Navigation upwards and right"
10  Show symbol "Navigation upwards, left and right"
11  Show symbol "Navigation downwards and left"
12  Show symbol "Navigation downwards and right"
13  Show symbol "Navigation downwards, left and right"
14  Show symbol "Navigation upwards, downwards and left"
15  Show symbol "Navigation upwards, downwards and right"
```

Example

```
<ESC>IW3<CR><LF>

// Show symbol "Navigation up- and downwards"
```

3.2.2.32 "Audiomode" Symbol

The "Audiomode" symbols are normally automatically switched on in combination with the "Volume" symbol. This command offers the possibility to switch it on independently and especially, to switch it off.

	
Private Mode	Handsfree Mode

Syntax

```
<ESC>IY(State)<CR><LF>
```

Parameter

```
(State):
0    Symbol OFF
1    Activate symbol "Handsfree mode"
2    Activate symbol "Privat mode"
```

Example

```
<ESC>IY2<CR><LF>

// Show symbol "Privat mode".
```

3.2.2.33 Loading Animation

The handset provides a loading animation.

```
<ESC>IO(State)<CR><LF>
```

Parameter

```
(State):
0   Animation off
1   Animation on
```

Example

```
<ESC>IO1<CR><LF>

// Show a pulsating circle animation in the text area of the display.
```

3.2.2.34 Progress Bar

The handset provides an animation for the display of progress.

Syntax

```
<ESC>II(f)<CR><LF>
```

Parameter

```
(f):
Progress in percent ( 0 ... 100 )
```

Example

```
<ESC>II50<CR><LF>

// Shows the progress at 50 % in the text area of the display.
```

3.2.2.35 Set Fonts

Separate fonts can be set for the zones Text Area and Softkeys (SoftkeyL and SoftkeyR). See also section 6.4 Fonts.

Syntax

```
<ESC>IDMF(Area),(Font)<CR><LF>
```

Parameter

```
(Area):
0 Text area
1 reserved
2 Font for softkeys SoftkeyL + SoftkeyR

(Font):
0 Font 15 x 26
1 Font 15 x 26B
2 Font 12 x 16
3 Font 12 x 16B

Currently, for area 0 only the values/fonts 0 and 1 are supported!
```

Example

```
<ESC>IDMF2,1<CR><LF>

// Select Font 1 for SoftkeysL and SoftkeyR
```

3.2.2.36 Set Colours

For the 3 display areas, front (fonts, symbols, icons) and background colours can be chosen separately from the colour space RGB555. The colour selection is activated immediately and remains active until this command is used again. The colour selection for the graphic modes is not enabled until the graphic is requested and as well won't be applied to already displayed graphics. The background of graphics is always the background of the text area. After a reset, the colour setting of the setup will be valid. A permanent storage is not required, as the application will change the colours during run time again. The colour space is restricted to 5 Bits per colour value for compatibility reasons. For displays with higher resolution, the missing LSB (lowest significant bits) values are 0.

Syntax

```
<ESC>C(Area),(r),(g),(b)<CR><LF>
```

Parameter

```
(Area):
0 Font Colour of Text Area
1 Background colour of Text Area
2 Font colour of Symbol Bar
3 Background colour of Symbol Bar
4 Font colour of softkeys SoftkeyL + SoftkeyR
5 Background colour of Softkey Bar
6 Graphic

(r), (g), (b):
0..31 Colour in RGB format (Colour Space RGB555)
```

Example

```
<ESC>C0,0,0,0<CR><LF>

// Set font colour to BLACK for Text Area
```

3.2.2.37 Restart of the Handset

This command performs a software restart of the handset. The bootloader string will not be sent.

Syntax

```
<ESC>&O0<CR><LF>
```

Parameter

```
None
```

Example

```
<ESC>&O0<CR><LF>

// Restart handset
```

3.2.2.38 Query Serial Number

With this command, the serial number of the handset can be retrieved.

Syntax

```
<ESC>&S?<CR><LF>
```

Parameter

```
None
```

Reply

```
Serial Number in the format:

<ESC>&S:<SPACE>(SN)<CR><LF>

(SN):
xxx     Serial number of handset
ERROR   No serial number programmed
```

3.2.2.39 Buzzer Function

This command activates the amplifier of the earpiece and can be used to play an external provided sound signal for a time in steps of 10 ms to simulate a buzzer function. The audio amplifier of the HA57 EVO is activated for the requested time. If the HOOK signal is activated (handset hung up), the volume is set to maximum level for the requested time. The sound needs to be provided externally over the audio path.

Syntax

```
<ESC>IB(Value)<CR><LF>
```

Parameter

```
(Value):
Time to activate the amplifier of the earpiece in steps of 10 ms.
Range is from 0 (short) to 255 (long).
```

Example

```
<ESC>IB10<CR><LF>

// Activate earpiece amplifier for a time of 100 ms
```

3.2.2.40 Buzzer Tone

This command is used to activate a sound signal for a certain amount of time in steps of 10 ms. The sound output comes from the internal buzzer.

The buzzer is operated by a rectangle generator, whose frequency is calculated as follows:

freq = 101265 kHz / (scale + 1)

The value 0 for scale or time disables the sound. Due to the resonance and frequency behavior of the buzzer, the volume during frequency sweep (from about 200 Hz) will vary.

Syntax

```
<ESC>IB(time);(scale)<CR><LF>
```

Parameter

```
(time):
Amount of time (x 10 ms) the sound is required to be played.
Values above 10 seconds are limited to 10 seconds (1000).

(scale):
Pitch. A higher value will lower the tone. The value ranges from 0 to 255.
```

Example

```
<ESC>IB1;40<CR><LF>

// Generate a key tone (beep).
```

3.2.2.41 Set Baud Rate

This command changes the baud rate of the serial interface on the fly. That means, that all communication is stopped, the serial interface is closed and opened again with the new baud rate. The selected baud rate will be saved permanently in EEPROM. At delivery, the baud rate is 115,200 Baud.

Syntax

```
<ESC>IX(Baud)<CR><LF>
```

Parameter

```
(Baud): Baud rate as follows:
9600             Baud rate 9,600 Baud
14400            Baud rate 14,400 Baud
19200            Baud rate 19,200 Baud
28800            Baud rate 28,800 Baud
38400            Baud rate 38,400 Baud
57600            Baud rate 57,600 Baud
76800            Baud rate 76,800 Baud
115200           Baud rate 115,200 Baud
```

Example

```
<ESC>IX9600<CR><LF>

// Set baud rate to 9,600 Baud
```

3.2.2.42 Query Baud Rate

Query the current baud rate.

Syntax

```
<ESC>IX?<CR><LF>
```

```
Parameter:  None
```

Reply

```
Baud rate in the format:

<ESC>IX:<SPACE>(Baud)<CR><LF>

(Baud):
9600             Baud rate 9,600 Baud
14400            Baud rate 14,400 Baud
19200            Baud rate 19,200 Baud
28800            Baud rate 28,800 Baud
38400            Baud rate 38,400 Baud
57600            Baud rate 57,600 Baud
76800            Baud rate 76,800 Baud
115200           Baud rate 115,200 Baud
```

Example

```
<ESC>IX?<CR><LF>

// Query baud rate

Possible Reply:

<ESC>IX: 9600<CR><LF>

// Baud rate is 9,600 Baud
```

3.2.3.43 Send & Show Graphic Data

With this command uncompressed or compressed data can be sent to the handset. The data is copied to the memory but not displayed yet.

Hinweis

Select the corresponding graphical mode afterwards to update the display!

Hints

1. It is recommended to transfer all rows of the graphic first, after that select the required display mode.
2. Select only the graphic mode(s) you need. For example, if you would like to put a graphic in row 1 and row 2, then select DISP_MODE_GR_9 and not one of the other.

Syntax

```
<ESC>G(Row),(Data)z<CR><LF>
```

Parameter

```
(Row):
r0-r19   Line (row) where the data shall be displayed

(Data):
xxx      Uncompressed or compressed data ASCII-coded
00       Clear line (row)
```

Examples

```
<ESC>Gr0,00z<CR><LF>

// Clear Line 0

<ESC>Gr0,a00aFFa331B1F0Ez<CR><LF>

// Compressed data for a user defined character "P" in Line 0

<ESC>IDM9<CR><LF>
// Update display
```

3.2.2.44 Set Earpiece Volume

There are 9 steps available for adjusting the volume of the earpiece in the handset. The volume is stored persistently.

Syntax

```
<ESC>IV(Value)<CR><LF>
```

Parameter

```
(Value): Sound Pressure Level SPL@10mV [dBSPL] as follows:
0         90.2
1         93.0
2         95.8
3         99.0
4         101.9
5         105.3
6         108.2
7         111.1
8         114.0

If no parameter is specified, the value stored in Setup is applied.
(Value) is limited to 8, default is 1.
```

Example

```
<ESC>IV4<CR><LF>

// Set Sound Pressure Level to 101.9 dB SPL ein
```

3.2.2.45 Query Earpiece Volume

This command is used to query the current setting of the earpiece's volume.

Syntax

```
<ESC>IV?<CR><LF>
```

Parameter

None

Reply

```
Current setting of volume in the format:

<ESC>IV:<SPACE>(Value)<CR><LF>

(Value):         Sound pressure level SPL@10mV [dB SPL] as follows:
0         90.2
1         93.0
2         95.8
3         99.0
4         101.9
5         105.3
6         108.2
7         111.1
8         114.0
```

Example

```
<ESC>IV?<CR><LF>

// Query volume setting

Possible Reply:

<ESC>IV: 4<CR><LF>

// Volume setting is 4 (101.9 dBSPL)
```

3.2.2.46 Decrease Volume Earpiece

This command increases the volume of the earpiece by 1 step. When the max value (8) has already been reached, a "?<CR><LF>" is returned.

Syntax

```
<ESC>IV+<CR><LF>
```

Parameter

None

Example

```
<ESC>IV+<CR><LF>

// Increase volume of earpiece by 1 step
```

3.2.2.47 Decrease Earpiece Volume

This command decreases the volume of the earpiece by 1 step. When the min value (0) has already been reached, a "?<CR><LF>" is returned.

Syntax

```
<ESC>IV-<CR><LF>
```

Parameter

None

Example

```
<ESC>IV-<CR><LF>

// Decrease volume of earpiece by 1 step
```

3.2.2.48 Set Microphone Gain

To control the gain of the microphone path a digital potentiometer is used. 10 steps are available.

The microphone gain is stored persistently.

Syntax

```
<ESC>IG(Value)<CR><LF>
```

Parameter

```
(Value):         Gain [in dB] as follows:
0          -38
1          -34
2          -29
3          -24
4          -19
5          -14
6          -10
7          4
8          0.8
9          0.8

(Value) is limited to 9, default is 3. With the default setting, the sensitivity of
the microphone path is ~40 mV eff/Pa ±3 dB (direction LRGP).
```

Example

```
<ESC>IG4<CR><LF>

// Set gain to -19 dB
```

3.2.2.49 Query Microphone Gain

This command is used to query the current setting of the microphone's gain.

Syntax

```
<ESC>IG?<CR><LF>
```

Parameter

```
None
```

Reply

```
Current setting of microphone's gain in the format:

<ESC>IG:<SPACE>(Value)<CR><LF>

(Value):         Gain [in dB] as follows:
0         -38
1         -34
2         -29
3         -24
4         -19
5         -14
6         -10
7         4
8         0.8
9         0.8
```

Example

```
<ESC>IG?<CR><LF>

// Query gain of microphone path

Possible Reply:

<ESC>IG: 4<CR><LF>

// Gain is 4 (-19 dB)
```

3.2.2.50 Increase Microphone Gain

This command increases the gain of the microphone by 1 step. When the max value (9) has already been reached, a "?<CR><LF>" is returned.

Syntax

```
<ESC>IG+<CR><LF>
```

Parameter

```
None
```

Example

```
<ESC>IG+<CR><LF>

// Increase gain of microphone by 1 step
```

3.2.2.51 Decrease Microphone Gain

This command decreases the gain of the microphone by 1 step. When the min value (0) has already been reached, a "?<CR><LF>" is returned.

Syntax

```
<ESC>IG-<CR><LF>
```

Parameter

```
None
```

Example

```
<ESC>IG-<CR><LF>

// Decrease gain of microphone by 1 step
```

3.2.2.52 Mute Earpiece

This command controls the amplifier for the earpiece. Turning off the amplifier will mute the earpiece.

Syntax

```
<ESC>M(mute)<CR><LF>
```

Parameter

```
(mute):
0 Earpiece active (no mute), amplifier turned on
1 Earpiece deactivated, (mute), amplifier turned off
```

Example

```
<ESC>M1<CR><LF>

// Turn off amplifier, earpiece muted
```

3.2.2.53 Mute Microphone

This command controls the amplifier for the microphone. Turning off the amplifier will mute the microphone. The microphone mute symbol will not be shown!

Syntax

```
<ESC>N(mute)<CR><LF>
```

Parameter

```
(mute):
0 Microphone active (no mute), amplifier turned on
1 Microphone deactivated, (mute), amplifier turned off
```

Example

```
<ESC>N1<CR><LF>

// Turn off amplifier, microphone muted
```

3.2.2.54 Scrolling Text

For animated representation of texts that are longer than one line, scrolling text can be displayed. The running direction is from right to left. A scrolling text on the display cannot be overwritten.

The text to be displayed should be placed in "" to avoid misinterpretation of special characters. A text, which is shorter than the line length, is positioned centred in the display at the beginning.

The frequency indicates, in which period of time the text is moved by exactly one display pixel. The speed depends on the display resolution. To assure that the parameters gain a similar speed compared with the older devices, the minimum frequency was reduced from 10 to 5 ms.

Syntax

```
<ESC>F,(Freq),(Font),(Pos),(Text)<CR><LF>
```

Parameter

```
(Freq):
0       Text fixed
1..255  Frequency in steps of 5 ms

(Font): Font and font size as follows:
0         Font size        15x26 (normal)
1         Font size        15x26B
2         Font size        24x40
3         Font size        12x16
4         Font size        12x16B
```

```
(Pos):
Line in which the scrolling text is shown 0...12

(Text):
Text to be scrolled (max 100 characters)
```

Example

```
<ESC>F,3,2,10,"Sampletext"<CR><LF>

// Text in line 10 is scrolled every 15 ms one pixel to the left
```

4 Setup

The setup menu is a utility tool implemented into the HA57 EVO. It is used to make some basic settings for the handset. Additionally, some information about the handset can be retrieved.

The setup menu can be entered during runtime of the emulations by pressing the left function key (green symbol) and the key 7 (S) **simultaneously** for about 2 seconds. The text "SETUP" appears on the display.

The setup menu is entered using the right softkey "menu".

It contains the submenus:

1	Brightness	"Brightness"
2	Factory set.	"Factory settings"
3	Logo	"Selection of the start-up logo"
4	Emulation	"Active emulation"
5	HW-Pin	"Mapping of pin 6 of the connector"

The submenus are described in the following chapters.

The setup can be exited from the main setup menu at any time with the right function key. The exit from the setup always results in a restart of the handset.

The SDK allows to remove single sections or the entire Setup for security related applications.

4.1 Brightness

For readability reasons the backlight of the display cannot be turned off completely. In the "off" state, the backlight can be dimmed only so far that readability is still guaranteed. The brightness for the "on" state and the "off" state can be separately set with the submenu "Brightness".

Press key "1" in the main setup menu to get to the submenu "Brightness". The current setting for the brightness for the "on" state is displayed. By using the navigation keys UP and DOWN the brightness for the "on" state can be adjusted in steps of 5 % in the range from 30 % to 100 %.

Press softkey "Off" to get to the setting for the "off" state. By using the navigation keys UP and DOWN the brightness for the "off" state can be adjusted in steps of 1 % in the range from 0 % to 20 %.

The range from 21 % to 29 % remains unused, so that a distinction between the "on" and "off" state of the backlight is guaranteed.

Press softkey "On" to return to the menu for the "on" state.

The currently active menu is displayed by "Display ON" or "Display OFF".

By pressing the right softkey "Menu", you return to the main setup menu, pressing the right function key exits the setup and will reboot the handset.

4.2 Factory Settings

Press key "2" in the main setup menu to get to the submenu "Factory settings". The display will then show a prompt.

By pressing the right softkey "Yes", the contents of the user defined parameter memory is erased. The handset automatically reboots and re-initializes the HA57 EVO with its factory settings (default values).

By pressing the left softkey "No" will cancel the operation and the main setup menu appears. The current settings are not changed.

4.3 Logo

Press key "3" in the main setup menu to get to the logo submenu.

There are three possibilities on offer for the start-up logo:

1	default Logo
2	customer Logo
3	no Logo

The second possibility, customer logo, is only accepted, when a customer logo was uploaded first. When the selection is confirmed, the chosen logo is displayed for a short time.

4.4 Emulation

Press key "4" in the main setup menu to get to the submenu "Emulation". Here it is possible to set the emulation HA57 EVO. Choose from the emulations:

1	pei tel HA20x	"Protokoll HA20x"
2	pei tel HA400	"Protokoll HA400"
3	User Application	

When delivered, the emulation "pei tel HA20x" is set using the protocol "HA20x".

If a user application was registered to the system with its name, this name appears as a third choice.

The currently active emulation is characterized by a selection bar. Use the navigation keys UP or DOWN, or use the keys 1-2 (3 in the presence of a user application), to select the emulation. By pressing the left soft-key "Set" the selected emulation is referred to as the emulation to be used after a restart.

By pressing the right softkey "Menu" in the submenu "Emulation", you return to the main setup menu, pressing the right function key exits the setup and will reboot the handset.

4.5 Mapping of Pin 6 of the Connector

Press key "5" in the main setup menu to get to the submenu "HW-Pin". In this section, the function of the pin 6 of the connector can be chosen. The handset offers either HOOK signal or PTT signal for choice. The selection is done by using the arrow keys UP and DOWN and the "Set" softkey.

The softkey "Menu" gets you back to the main menu.

4.6 Info Display

Press the softkey "Info" to access the device information menu.

It contains:

- Software version
- Date of the software release
- Serial number of the HA57 EVO
- Switch-on counter
- Detected maximum operation voltage

5 Own Software

The handset HA57 EVO is able to execute user applications created by a customer itself. The user application code is created under an ARM development environment and downloaded to the SDK using the tool **AppLoader HA57 EVO**. A Software Development Kit (SDK) is available for application developers.

Enabling your application is described in section 4.4 Emulation.

The Setup menu is also available when an application was activated.

5.1 User Applications

It is therefore possible, taking advantage of the available hardware to create completely new applications. Here, an extensive catalogue of functions for initialization and control of the individual hardware modules is provided. Through the use of complex features, the application software can be very slim. Detailed knowledge of the hardware is not required.

5.2 Command Extension

An alternative to using a user application, the SDK of the HA57 EVO offers the possibility to keep using the HA20x or HA400 protocols and to define additional commands or to change or hide available commands. After the activation of uploaded software in the Setup, customized commands are available in addition to the chosen protocol. Respective received texts and commands can be intercepted and edited. It is also possible to manipulate keypad input.

6 Protocol Description LogoLoader

The following chapter describes the protocol sequence of the LogoLoader application used for HA57 EVO devices. It is intended for users with own connected hardware, e.g. a bord computer and the requirement to change the start logo using the own hardware.

Loading a logo file requires to activate the service mode of the handset. The service mode adds AT sequences to the normal protocol mode, especially the command to load and switch the logo.

6.1 Command Sequence

- Start the service mode
- Transfer the logo file
- Activate the customer logo
- Reset the service mode -> device restart with new logo

6.2 AT Command Descriptions

Syntax

```
AT+SERVICE=N
```

Parameter

```
N: State service mode
1 - service mode on
0 - service mode off
```

Reply

```
Service mode on:
Software version, compilation date and loader checksum, final OK
```

```
Service mode off:
OK and restart of the device starting with software version and compilation date
```

Example

```
> AT+SERVICE=1
< HA57 EVO V.1.0.6
< Sep 11 2019 KV: E694
<
< OK
```

Note

It is strongly recommended to deactivate SDK applications, when available, to avoid interferences with the service command parser.

The following commands are only available (and answered correctly) in active service mode:

Syntax

```
AT*LDLOGO
```

Reply

```
OK

// Flash loader started.
```

The command starts a flash-loader to transfer the logo file. Await 0x15 (NAK) and start the line transfer. After an EOT symbol is sent or errors occur, the interpreter returns to command mode. The flash-loader sequence is described in the next section.

Syntax

```
AT*SETLOGO=L
```

Parameter

```
L:
1 - custumer logo
0 - peitel logo (default)
```

Reply

```
OK

// Logo mode was set.
```

Example

```
> AT*SETLOGO=1
< OK
```

Check the set logo configuration:

Syntax

```
AT*SETLOGO?
```

Reply

```
*SETLOGO: L
OK
```

Example

```
> AT*SETLOGO?

< *SETLOGO: 1
< OK
```

6.3 Flash Loader for Customer Logo Files

The protocol of the hexfile loader is inspired by the x-modem protocol. The table below shows the used protocol tokens.

Symbol	**Hex**	**Description**
SOH	01	Start of 128 byte line
EOT	04	End of raw mode
ACK	06	Acknowledge a line
NAK	15	Leave the current protocol mode
CAN	18	Cancel transfer (error)
SUB	1A	Line fill byte

CRC		Sum of the raw hex bytes (1 byte)

Table 6: Protocol Tokens

The logo bmp file has to be compiled to a sequence of protocol lines with a 1 byte sequence number 1 increased by 1, byte size 128:

```
<SOH><seq><~seq><128 raw hex bytes or <SUB> to fill><CRC>
```

At end of file, the last line has to be filled with <SUB> bytes and a final

```
<EOT>
```

must be sent to leave the raw protocol mode.

Each protocol line is answered with

```
<ACK>
```

If the received line was ok (CRC check, framing) or in case of errors with

```
<CAN>
```

A sent <CAN> ends the transfer mode. In that case, the partially transferred logo data remain in flash.

It is important to wait for the <ACK> byte before sending the next line because the receive lines are flashed directly and the receiver hardware could be busy.

Example

```
> AT*LDLOGO<CR><LF>

< <CR><LF>OK<CR><LF><CR><LF>

short delay to remove the logo flash area

< <NAK>

now the device is ready to receive protocol lines:
```

```
> 01 01 FE 42 4D 38 58 02 00 00 00 00 00 36 00 00 00 28 00 00 00 F0 00 00 00 40 01 00
00 01 00 10 00 00 00 00 00 02 58 02 00 12 0B 00 00 12 0B 00 00 00 00 00 00 00 00 00 00
 FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF
7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F 8D

< 06

> 01 02 FD FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF
7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF
7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F 80

< 06

> 01 03 FC FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF
7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF
7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F 80

< 06

...



> 01 B1 4E FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF
7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
00 00 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A
1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A
1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 9A

< 06

> 04

< <ACK><CR><LF>

Back to command mode
```

Note

The BMP interpreter of the device firmware is extremely simple, no compression, no colour tables, fixed size and colour format.

For the HA57 EVO the bmp-file must start with the byte sequence:

```
42 4D BMP magic

38 58 02 00 size of the BMP file in bytes (14 + 40 + 240 * 320 * 2) round up mod 4

00 00 00 00 reserved, 0

36 00 00 00 offset to picture data

---------------------

28 00 00 00 Info header size

F0 00 00 00 Width 240 pixel

40 01 00 00 Height 320, buttom up

01 00 planes 1

10 00 color deep 16 (RGB565)
```

The tool LogoLoader checks that sequence if an HA57 EVO is connected.

7 Appendix

7.1 Coding Scheme of Graphic Modes

The coding scheme used now becomes clear if we look at a line in more detail. For every column, the LSB (least significant bit) is the top pixel and the MSB (most significant bit) is represented by the lowest pixel. This therefore gives 8 bits per column, where a display pixel is represented by a "1" at the corresponding bit position:

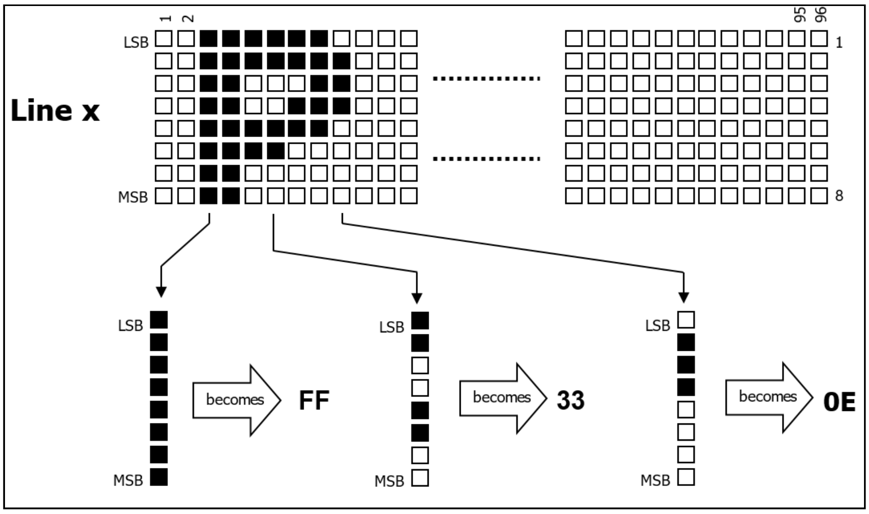


Figure 18: Coding scheme of graphic modes

For the example above, this would give the following values for the first 10 columns:

0000**FF**FF33**33**1B1F**0E**00

The following points must thereby be taken into account:

- The resulting values are coded in hexadecimal. The values are then transmitted ASCII-encoded to the handset. For example, to transmit "FF", the value 0 x 46 ("F") must first be sent, followed by a further 0x46 ("F")!
- Upper- and lower-case letters must be taken into account. Uppercase letters must be used for the hexadecimal values A-F!
- In order to fill the entire display with a graphic, a large amount of graphic data would have to be transmitted. A line would therefore require at least 2 x 96 bytes. To reduce the amount of data, a compression algorithm has been implemented, as described in the next section.

7.2 Compression Algorithm

In order to fill the entire display with a graphic, a large amount of graphic data would have to be transmitted. This would take some time even at a baud rate of 115200 Baud. A compression function is used in order to reduce the amount of data.

The compression makes use of the process of using a factor several times in the event of equal recurring values. Now, to distinguish between "normal data" and compressed data, lowercase letters are used for the factor. So, for example, if the value "00" has to be transmitted five times, "d00" is sent instead. The following mapping applies for the factor (see Compression table):

2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	D	e	f	g	h	i	j	k	l

14	15	16	17	18	19	20	21	22	23	24	25	26
m	n	o	p	Q	r	s	t	u	v	w	x	y

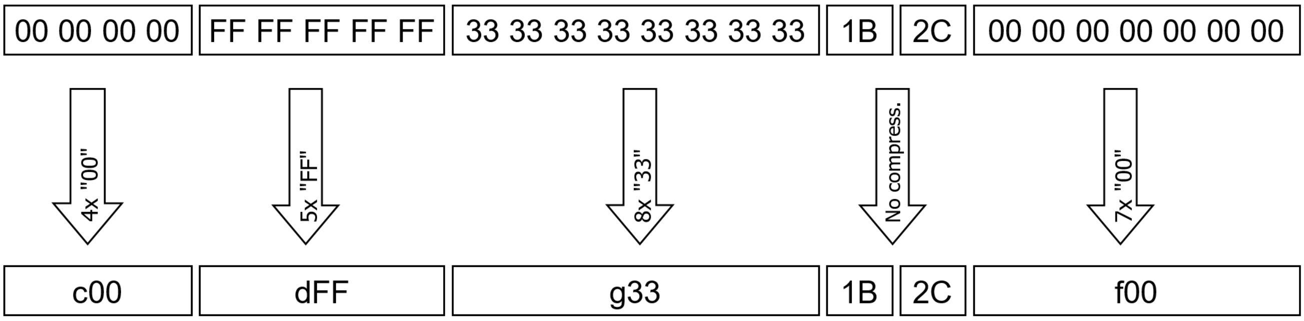
Example:

The data stream with the following 52 bytes

00 00 00 00 FF FF FF FF FF 33 33 33 33 33 33 33 33 1B 2C 00 00 00 00 00 00 00

can be compressed to: c00 dFF g33 1B 2C f00

Now only 16 bytes have to be transmitted instead of 52 bytes.



The empty spaces between the values are inserted only for reasons of clarity. The real data stream may not have these spaces!

The actual data stream to be transmitted for the graphic data would then be:

c	0	0	d	F	F	g	3	3	1	B	2	C	f	0	0
0x63	0x30	0x30	0x64	0x46	0x46	0x67	0x33	0x33	0x31	0x42	0x32	0x43	0x66	0x30	0x30

7.3 Code Tables

The table below shows the character set used in the divers' display modes. A byte value (HEX code) of the serial data transfer or the buffer of the SDK is displayed accordingly to this table. For empty fields in the table will generate a replacement character, actually, a space.

If there is no coding for a character, the text will be interpreted according to UTF-8 and the display of the characters will be in accordance with the character set of the firmware. Using the SDK, the available fonts can be replaced or enhanced.

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x00				empty
0x01	£	£	£	
0x02	§	§	§	
0x03	¥	¥	¥	
0x04	È	È	È	
0x05	É	É	É	
0x06	Ù	Ù	Ù	
0x07	Ì	Ì	Ì	
0x08	Ò	Ò	Ò	
0x09	Ç	Ç	Ç	
0x0A				Carriage Return
0x0B	Ø	Ø	Ø	
0x0C	ø	ø	ø	
0x0D				Line Feed (New Line)
0x0E	Á	Á	Á	
0x0F	á	á	á	
0x10	►	►	►	
0x11	_	_	_	Underscore

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x12	↕	↕	↕	
0x13	‼	‼	‼	
0x14	¶	¶	¶	
0x15	§	§	§	
0x16	-	-	-	
0x17	↨	↨	↨	
0x18	↑	↑	↑	
0x19	↓	↓	↓	
0x1A	→	→	→	
0x1B				ESC
0x1C	Æ	Æ	Æ	
0x1D	Æ	Æ	Æ	
0x1E	ß	ß	ß	
0x1F	É	É	É	
0x20				Space
0x21	!	!	!	
0x22	"	"	"	
0x23	#	#	#	
0x24	$	$	$	
0x25	%	%	%	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x26	&	&	&	
0x27	'	'	'	
0x28	(	(	(	
0x29	)	)	)	
0x2A	*	*	*	
0x2B	+	+	+	
0x2C	,	,	,	
0x2D	-	-	-	
0x2E	.	.	.	
0x2F	/	/	/	
0x30	0	0	0	
0x31	1	1	1	
0x32	2	2	2	
0x33	3	3	3	
0x34	4	4	4	
0x35	5	5	5	
0x36	6	6	6	
0x37	7	7	7	
0x38	8	8	8	
0x39	9	9	9	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x3A	:	:	:	
0x3B	;	;	;	
0x3C	<	<	<	
0x3D	=	=	=	
0x3E	>	>	>	
0x3F	?	?	?	
0x40	@	|	@	
0x41	A	A	A	
0x42	B	B	B	
0x43	C	C	C	
0x44	D	D	D	
0x45	E	E	E	
0x46	F	F	F	
0x47	G	G	G	
0x48	H	H	H	
0x49	I	I	I	
0x4A	J	J	J	
0x4B	K	K	K	
0x4C	L	L	L	
0x4D	M	M	M	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x4E	N	N	N	
0x4F	O	O	O	
0x50	P	P	P	
0x51	Q	Q	Q	
0x52	R	R	R	
0x53	S	S	S	
0x54	T	T	T	
0x55	U	U	U	
0x56	V	V	V	
0x57	W	W	W	
0x58	X	X	X	
0x59	Y	Y	Y	
0x5A	Z	Z	Z	
0x5B	[	Ä	[	
0x5C	\	Ö	\	
0x5D	]	Ñ	]	
0x5E	^	Ü	^	
0x5F	_	§	_	
0x60	`	¿	`	
0x61	a	a	a	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x62	b	b	b	
0x63	c	c	c	
0x64	d	d	d	
0x65	e	e	e	
0x66	f	f	f	
0x67	g	g	g	
0x68	h	h	h	
0x69	i	i	i	
0x6A	j	j	j	
0x6B	k	k	k	
0x6C	l	l	l	
0x6D	m	m	m	
0x6E	n	n	n	
0x6F	o	o	o	
0x70	p	p	p	
0x71	q	q	q	
0x72	r	r	r	
0x73	s	s	s	
0x74	t	t	t	
0x75	u	u	u	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x76	v	v	v	
0x77	w	w	w	
0x78	x	x	x	
0x79	y	y	y	
0x7A	z	z	z	
0x7B	{	ä	{	
0x7C	|	ö	|	
0x7D	}	ñ	}	
0x7E	~	ü	~	
0x7F	Δ	ã	Δ	
0x80	Ç	@	Ç	
0x81	ü		ü	
0x82	é		é	
0x83	â		â	
0x84	ä		ä	
0x85	à		à	
0x86	å		å	
0x87	ç		ç	
0x88	ê		ê	
0x89	ë		ë	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x8A	è		è	
0x8B	ï		ï	
0x8C	î		î	
0x8D	ì		ì	
0x8E	Ä	Ä	Ä	
0x8F	Å	Ä	Å	
0x90	É		É	
0x91	æ		æ	
0x92	Æ		Æ	
0x93	ô		ô	
0x94	ö		ö	
0x95	ò		ò	
0x96	û		û	
0x97	ù		ù	
0x98	ÿ		ÿ	
0x99	Ö		Ö	
0x9A	Ü	ö	Ü	
0x9B	¢	ü	¢	
0x9C	£		£	
0x9D	¥		¥	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0x9E	Pt		Pt	
0x9F	ƒ		ƒ	
0xA0	á			
0xA1	í		Ë	
0xA2	ó			
0xA3	ú			
0xA4	ñ			
0xA5	Ñ		S	
0xA6	ã		I	
0xA7			Ï	
0xA8	¿		J	
0xA9				
0xAA				
0xAB				
0xAC				
0xAD	Í			
0xAE	<<			
0xAF	>>			
0xB0			A	
0xB1			Б	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0xB2			В	
0xB3			Г	
0xB4			Д	
0xB5			Е	
0xB6			Ж	
0xB7			З	
0xB8			И	
0xB9			Й	
0xBA			К	
0xBB			Л	
0xBC			М	
0xBD			Н	
0xBE			О	
0xBF			П	
0xC0	À		Р	
0xC1	Á		С	
0xC2	Â		Т	
0xC3	Ã		У	
0xC4	Ä		Ф	
0xC5	Å	Ä	Х	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0xC6	Æ		Ц	
0xC7	Ç		Ч	
0xC8	È		Ш	
0xC9	É		Щ	
0xCA	Ê		Ъ	
0xCB	Ë		Ы	
0xCC	Ì		Ь	
0xCD	Í		Э	
0xCE	Î		Ю	
0xCF	Ï		Я	
0xD0	Đ		а	
0xD1	Ñ		б	
0xD2	Ò		в	
0xD3	Ó		г	
0xD4	Ô		д	
0xD5	Õ		е	
0xD6	Ö		ж	
0xD7		Ö	з	
0xD8	Ø		и	
0xD9	Ù		й	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0xDA	Ú		к	
0xDB	Û		л	
0xDC	Ü		м	
0xDD	Ý	Ü	н	
0xDE	Þ		о	
0xDF	ß		п	
0xE0	à		р	
0xE1	á		с	
0xE2	â		т	
0xE3	ã		у	
0xE4	ä		ф	
0xE5	å	ä	х	
0xE6	æ		ц	
0xE7	ç		ч	
0xE8	è		ш	
0xE9	é		щ	
0xEA	ê		ъ	
0xEB	ë		ы	
0xEC	ì		ь	
0xED	í		э	

HEX Code	DISP_MODE_ASCII	DISP_MODE_SMS DISP_MODE_TB	DISP_MDOE_CYR	Remarks
0xEE	î		ю	
0xEF	ï		я	
0xF0	@			
0xF1	ñ		ë	
0xF2	ò			
0xF3	ó			
0xF4	ô			
0xF5	õ		s	
0xF6	ö	ö	i	
0xF7	÷		ï	
0xF8	ø		J	
0xF9	Æ			
0xFA	ú			
0xFB	û			
0xFC	ü	ü		
0xFD	ý		§	
0xFE	þ			
0xFF				

Table 7: Character codes

7.4 Fonts

The following fonts are available:

- SMALL
- SMALL BOLD
- NORMAL
- NORMAL BOLD
- LARGE

All fonts are based on a pixel coding of the Google font RobotoMono (UTF 33-511, 1023-1279, selected additional characters) with the heigths 20, 28 and 40 pixels, matched to the utilized display. See also https://fonts.google.com/specimen/Roboto+Mono.

Alle Fonts basieren auf einer pixel-Kodierung des Google-Fonts RobotoMono (UTF 33-511, 1023-1279, ausgewählte Zusatzzeichen) mit den Höhen 20, 28 und 40 Pixel abgestimmte auf das genutzte Display, siehe auch https://fonts.google.com/specimen/Roboto+Mono.

Sections 3.1.2.1 Display-Modus HA20x and 3.2.2.1 Display-Modus HA400 describe how to use the fonts.

7.5 License

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

END OF DOCUMENT