

PLE protocol specifications

Version: 6fd1631 10.02.2016

Introduction

This document specifies PiccoLink Emulation (PLE) protocol. PLE is PiccoLink over TCP/IP. Protocol is implemented in TS8 mobile barcode and RFID scanner and PLE server SDK. Protocol is compatible with various vendors Handheld data terminals and Retail/warehouses/logistics software.

Although the PLE protocol and server supports handling the data through serial port, the functions can be implemented only if the terminal device is equipped with BlueTooth serial port interface.

Software developers, testers and network administrators are the specifications target audience.

Message from host to terminal

Message frame structure

Field	Length (bytes)	Values		
SOH	1	Start of header (fixed value 0x01)		
LENGTH	1	Number of bytes located in DATA field		
PREFIX_A	1	bit 0 - "1" indicates that message is repeated by a repeater		
		bit 1 - "1" indicates that message goes from Host to terminal		
		bit 2 - Ignored when frame sent from Host to terminal		
		bit 3 - "1" when frame is sent independently by host (terminal should be in receiver mode to receive this frame)		
PREFIX_A	1	bit 4-7 - Value indicates the field strength of the radio signal		
		ID_H	1	High byte of the terminal serial number
		ID_L	1	Low byte of the terminal serial number
		DATA		Contains actual data and the commands. ASCII values from 0x0E ... 0x1F are reserved for the commands and should not be sent within a command if the protocol specifications states otherwise
CRC_H	1	High byte of the checksum		
CRC_L	1	Low byte of the checksum		

List of commands

CmdCode	Command Name	Length (bytes)	Syntax	Description
0x06	ACK	1	06	Acknowledgement from the Host
0x07	BEEP	1	07	Generates a beep
0x08	SET_PARAMETERS	6 + L	08 PIN_H PIN_L ADDR_H ADDR_L L DATA	Writes L bytes of DATA to ADDR if the PIN code is correct
0x09	TAB	1	09	Tabulator, moves the cursor 4 positions forward
0x0A 0x01	EMULATE_KEY	5	0A 01 KEY 00 00	Emulates the KEY press as if the KEY was pressed by the user of the Terminal
0x0A 0x02	SENDKEYS	5	0A 02 MSK_LOW MSK_MID MSK_HIGH	Sets the selected keys to direct mode in which they will be sent directly to the host when pressed (instead of being handled by the UI as in normal mode)
0x0A 0x04	PASSWORD	5	0A 04 P1 P2 P3	Sets selected fields to the "password style" input fields
0x0A 0x08	SHIFT_STATE	5	0A 08 SHIFT 00 00	Sets Shift OFF (if SHIFT = 00) or ON (SHIFT != 00)
0x0A 0x10	SERIAL_TOUT	5	0A 10 TIMEOUT 00 00	Not implemented
0x0B	NEW_RFID_FIELD	6	0B POS L STYLE FIRST LAST	Definition of an RFID Input Field
0x0C	RFID_READ_PAGES	3	0C FIRST LAST	Read memory of RFID tag
0x0D	LINEFEED	1	0D	Carriage Return + Line Feed
0x0E	MOVE_CURSOR	2	0E POS	Moves the cursor to POS
0x0F	NEW_FIELD	4	0F POS L STYLE	Creates a new input field at position POS, with length L and attributes set by the STYLE bitfield
0x10	BEEP_EX	2 + L	10 L DURATION DELAY ...	Generates a beep sequence with specified durations of beep and delay
0x11	MODIFY_FIELD	3	11 POS FCMD	Modifies the attributes of the field at position POS

0x12	CLEAR	4	12 CCMD POS_START POS_STOP	Clears fields and/or static text from the display
0x13	SET_VIEW	2	13 ROW	Sets the viewing window
0x14	POPUP	2 + L	14 L TEXT	Displays the pop-up window with TEXT
0x16	FIELD_TEXT	3 + L	16 POS L TEXT	Fill (presets) the field with the TEXT
0x17	BUTTON	3 + L	17 POS L TEXT	Creates a “button style” input field
0x18	SEND_PING	2	18 DELAY	Send PING command to HOST when DELAY has expired
0x19	WRITE_SERIAL	2 + L	19 L DATA	Not implemented
0x1A	READ_SERIAL	2	1A DELAY	Not implemented
0x1B	NEW_FIELD_EX	4	1B POS L STYLE	Creates a new input field at position POS, with length L and attributes set by the STYLE bitfield
0x1C	RECEIVER	2	1C MODE	Sets the receiver mode on/off
0x1D	FORM_ID	3	1D ID_H ID_L	Sets an ID to the current form
[none]	STATIC_TEXT		ASCII TEXT	All bytes >= 0x20 are treated as static text and written to the current cursor position

ACK

Acknowledgement from the Host (1 byte).

Syntax: 06

Clears all fields that have the “clear on ack” attribute set.

BEEP

Generates a beep (1 byte).

Syntax: 07

SET_PARAMETERS

Sets or modifies the Terminal configuration.

Syntax: 08 | PIN_H | PIN_L | ADDR_H | ADDR_L | L | DATA

PIN_H | PIN_L - 4-digit PIN code (0...9999) to access the configuration.

ADDR_H | ADDR_L - Configuration register that will be written.

L - Length of data.

DATA - Data to be written to the register.

TAB

Tabulator, moves the cursor 4 positions forward (1 byte).

Syntax: 09

EMULATE_KEY

Emulates a key press of the Terminal as if pressed by the user (5 bytes)

Syntax: `0A | 01 | KEY | 00 | 00`

`KEY` values are defined in the following table:

KEY Value	Physical key
1	F1
2	F2
3	F3
4	F4
5	F5
6	SCAN
7	DEL
8	Special Characters (lower left corner)
9	[unused]
10	UP arrow
11	DOWN arrow
12	OK
13	0
14	1
15	2
16	3
17	4
18	5
19	6
20	7
21	8
22	9

SENDKEYS

Sets the selected keys to direct mode in which they will be sent directly to the host when pressed (instead of being handled by the UI as in normal mode). (5 bytes)

Syntax: `0A | 02 | MSK_LOW | MSK_MID | MSK_HIGH`

`MSK_LOW`, `MSK_MID`, `MSK_HIGH` are defined in the following table:

Byte	Bit	Physical key
MSK_LOW	0	F1
MSK_LOW	1	F2
MSK_LOW	2	F3
MSK_LOW	3	F4
MSK_LOW	4	F5
MSK_LOW	5	SCAN
MSK_LOW	6	DEL
MSK_LOW	7	Special Characters (lower left corner)
MSK_MID	0	[unused]
MSK_MID	1	UP arrow
MSK_MID	2	DOWN arrow
MSK_MID	3	OK
MSK_MID	4	0
MSK_MID	5	1
MSK_MID	6	2
MSK_MID	7	3
MSK_HIGH	0	4
MSK_HIGH	1	5
MSK_HIGH	2	6
MSK_HIGH	3	7
MSK_HIGH	4	8
MSK_HIGH	5	9

PASSWORD

Sets selected fields to the “password style” input fields. In case the user of the Terminal enters some text to these fields, asterisks (*) will appear on the terminal screen (5 bytes).

Syntax: `0A | 04 | P1 | P2 | P3`

Up to 3 fields can be made “password style” by using this command once.

`P1` - Position of the first field that will become “password style” input field.

`P2` - Position of the second field that will become “password style” input field (value 255 means “not used”).

`P3` - Position of the third field that will become “password style” input field (value 255 means “not used”).

SHIFT_STATE

Sets or resets the Shift state (5 bytes).

Syntax: `0A | 08 | SHIFT | 00 | 00`

`SHIFT` - 0x00 shift OFF

any other value switches Shift ON.

SERIAL_TOUT

Not implemented.

NEW_RFID_FIELD

Defines a new RFID input field on the screen (6 bytes).

Syntax: `0B | POS | L | STYLE | FIRST | LAST`

`POS` - Starting point of the field (0 - 239).

`L` - Field length (1 - 63).

`STYLE` - Field parameters.

`FIRST` - First block of memory that will be read from the RFID tag.

`LAST` - Last block of memory that will be read from the RFID tag.

The priority of this command is equal to the NEW_FIELD command. This affects the order of execution of commands within one message frame.

`STYLE` bits can be set or cleared to enable or disable the function.

Bit	Name	Description
0	SND_ENTER	Field is sent to HOST by pressing the OK key on this field.
1	NO_SEND	Field is not sent if the SND_ALL command occurs, unless the field itself gave the SND_ALL command.
2	SND_ALL	All fields in the form (except NO_SEND fields) are sent to the host when the OK key is prssed on this field.
3	TAG_ID	The unique ID of the RFID tag is read and copied to this field. HEX to ASCII conversion is performed on this data. FIRST and LAST bytes are ignored if this bit is set.
4	FLD_LINE	Field is underlined (_____)
5	FLD_READER	Field can be filled with RFID scanner data.
6	READER_DEFAULT	If another field is active and it has not the FLD_READER bit set, the RFID scanner is activated and the data goes to this field automatically, even if the active field is not an RFID field itself. Only one READER_DEFAULT field can be defined per form.
7	FLD_ACTIVE	This field will be set active

`SND_ENTER`, `SND_ALL` and `FLD_READER` combinations:

<code>SND_ENTER</code>	<code>SND_ALL</code>	<code>FLD_READER</code>	Description
0	0	0	Field cannot be read with an RFID reader. If the user presses the OK key on this field, nothing is sent to the server, only the focus moves to the next field.
0	0	1	Field can be read with an RFID reader, but the data is not automatically sent to the server. If the user presses the OK key on this field, nothing is sent to the server, only the focus moves to the next field.
0	1	0	Unused. The Terminal acts as if <code>SND_ENTER</code> bit is set.
0	1	1	Unused. The Terminal acts as if <code>SND_ENTER</code> bit is set.
1	0	0	Field cannot be read with an RFID reader. In case the user presses the OK key on this field, it is sent to the server.
1	0	1	Field is automatically sent to the server either by reading data from the RFID reader or by pressing the OK key on this field.
1	1	0	Field cannot be read with an RFID reader. In case the user presses the OK key on this field, all the fields on this form are sent to the server, except fields marked as <code>NO_SEND</code> .
1	1	1	All the fields on the form (except the <code>NO_SEND</code> fields) are sent to the server either by reading data from the RFID reader or by pressing the OK key on this field, even if the <code>NO_SEND</code> bit is set on this field.

RFID_READ_PAGES

Reads data from an RFID tag (3 bytes).

Syntax: `@C | FIRST | LAST`

`FIRST` - First block of memory that will be read from the RFID tag.

`LAST` - Last block of memory that will be read from the RFID tag.

The Terminal activates its RFID scanner after getting this command. The scanner remains ON until a tag has been successfully read or until the timeout specified in the configuration file expires. This command does not interact with the terminal display.

In case the RFID tag read succeeds, the Terminal sends the data to the Host using `RFID_DATA` command.

LINEFEED

Moves the cursor to the beginning of next line (1 byte).

Syntax: `0D`

MOVE_CURSOR

Moves the cursor to the selected position (2 bytes).

Syntax: `0E | POS`

`POS` - New cursor position (0...239).

NEW_FIELD

Defines a new input field on the screen (4 bytes).

Syntax: `0F | POS | L | STYLE`

`POS` - Starting point of the field (0 - 239).

`L` - Field length (1 - 63).

`STYLE` - Field parameters.

`STYLE` bits can be set or cleared to enable or disable the function.

Bit	Name	Description
0	SND_ENTER	Field is sent to HOST by pressing the OK key on this field.
1	NO_SEND	Field is not sent if the SND_ALL command occurs, unless the field itself gave the SND_ALL command.
2	SND_ALL	All fields in the form (except NO_SEND fields) are sent to the host when the OK key is pressed on this field.
3	TAG_ID	The unique ID of the RFID tag is read and copied to this field. HEX to ASCII conversion is performed on this data. FIRST and LAST bytes are ignored if this bit is set.
4	FLD_LINE	Field is underlined (_____)
5	FLD_READER	Field can be filled with RFID scanner data.
6	READER_DEFAULT	If another field is active and it has not the FLD_READER bit set, the RFID scanner is activated and the data goes to this field automatically, even if the active field is not an RFID field itself. Only one READER_DEFAULT field can be defined per form.
7	FLD_ACTIVE	This field will be set active

SND_ENTER, SND_ALL and FLD_READER combinations:

SND_ENTER	SND_ALL	FLD_READER	Description
0	0	0	Field cannot be filled with data from laser- or external scanner. If the user presses the OK key on this field, nothing is sent to the server, only the focus moves to the next field.
0	0	1	Field can be filled with data from laser- or external scanner, but the data is not automatically sent to the server. If the user presses the OK key on this field, nothing is sent to the server, only the focus moves to the next field.
0	1	0	Unused. The Terminal acts as if SND_ENTER bit is set.
0	1	1	Unused. The Terminal acts as if SND_ENTER bit is set.
1	0	0	Field cannot be filled with data from laser- or external scanner. In case the user presses the OK key on this field, the data entered by the user via terminal keyboard is sent to the server.
1	0	1	Field is automatically sent to the server either by reading data from laser- or external scanner or by pressing the OK key on this field.
1	1	0	Field cannot be filled with data from laser- or external scanner. In case the user presses the OK key on this field, all the fields on this form are sent to the server, except fields marked as NO_SEND.
1	1	1	All the fields on the form (except the NO_SEND fields) are sent to the server either by reading data from laser- or external scanner or by pressing the OK key on this field, even if the NO_SEND bit is set on this field.

BEEP_EX

Generates a beep sequence with specified durations of beep and delay (2 + L bytes).

Syntax: 10 | L | BEEP_ON | BEEP_OFF |

Length of BEEP_ON and BEEP_OFF duration bytes.

BEEP_ON - Beep ON duration (DURATION = BEEP_ON * 10 ms).

BEEP_OFF - Beep OFF duration (DURATION = BEEP_OFF * 10 ms).

MODIFY_FIELD

Modifies field attributes (3 bytes).

Syntax: 11 | POS | FCMD

POS - Field position (0...239).

FCMD - Action to be taken on this field.

Bits of FCMD are defined as follows:

Bit	Name	Description
0	FLD_REMOVE	Removes the field
1	FLD_CLEAR	Clears the field (also clears locked fields)
2	[Reserved]	[Reserved]
3	FLD_LOCK	
4	[Reserved]	[Reserved]
5	[Reserved]	[Reserved]
6	[Reserved]	[Reserved]
7	FLD_ACTIVE	Activates the field (focus moves to this field)

CLEAR

Clears fields and/or static text from the display (4 bytes).

Syntax: `12 | CCMD | POS_START | POS_STOP`

`CCMD` - Flags describing the clearing process.

`POS_START` - First position on the screen to be cleared (0...239).

`POS_STOP` - Last position on the screen to be cleared (0...239).

Bits of `CCMD` are defined as follows:

Bit	Name	Description
0	CLEAR_TXT	Removes static text from POS_START to POS_STOP (including POS_START and POS_STOP positions)
1	CLEAR_FLD	Removes any fields that intersect with the region POS_START...POS_STOP
2	CLEAR_FLDDATA	Clears field data from POS_START to POS_STOP
3	[Reserved]	[Reserved]
4	[Reserved]	[Reserved]
5	[Reserved]	[Reserved]
6	[Reserved]	[Reserved]
7	[Reserved]	[Reserved]

SET_VIEW

Sets the viewing window.

Syntax: `13 | ROW`

`ROW` - Selects the virtual display row number that will be placed to the first row on the physical display (0..4).

POPUP

Displays a message to the user in a separate pop-up window.

Syntax: `14 | L | TEXT`

`L` - Length of text.

`TEXT` - Text to be displayed in the pop-up window. May include the cursor control commands (`TAB`, `LINEFEED` and `MOVE_CURSOR`).

FIELD_TEXT

Fills (presets) the field with text (3 + L bytes).

Syntax: `16 | POS | L | TEXT`

`POS` - Position of field to be filled.

`L` - Length of text.

`TEXT` - Text to the field.

BUTTON

Creates a “button style” input field (3 + L bytes).

Syntax: `17 | POS | L | TEXT`

This command will create a locked input field that will be preset with text.

`POS` - Position of button (0...239).

`L` - Length of “button style” input field.

`TEXT` - Text for the button.

SEND_PING

Request the Terminal to start sending PING requests to the Host periodically.

Syntax: `18 | DELAY`

`DELAY` - Delay in seconds (1...99) or 0 for stopping periodical PING requests.

Note that the “new style” of PING behaviour is to send the PING requests periodically until the Host disables it by sending the PING command with parameter DELAY = 0.

WRITE_SERIAL

Not implemented.

READ_SERIAL

Not implemented.

NEW_FIELD_EX

Defines a new input field on the screen (4 bytes).

Syntax: `1B | POS | L | STYLE`

`POS` - Starting point of the field (0 - 239).

`L` - Field length (1 - 63).

`STYLE` - Field parameters.

`STYLE` bits can be set or cleared to enable or disable the function.

Bit	Name	Description
0	SND_ENTER	Field is sent to HOST by pressing the OK key on this field.
1	NO_SEND	Field is not sent if the SND_ALL command occurs, unless the field itself gave the SND_ALL command.
2	SND_ALL	All fields in the form (except NO_SEND fields) are sent to the host when the OK key is pressed on this field.
3	[Reserved]	
4	OVR	Overwrite mode. When OVR bit is set and the field becomes active, all the field data will become 'selected' and will be overwritten when new data is entered to this field.
5	FLD_READER	Field can be filled with laser- or external BlueTooth scanner data.
6	READER_DEFAULT	Indicates that this field will receive the data from laser- or external scanner from any fields on the form that itself do not have the FLD_READER bit set. Note that only one field on the form can have this bit set! Also note to set the FLD_READER bit of this field, otherwise the READER_DEFAULT bit has no effect.
7	FLD_ACTIVE	This field will be set active

`SND_ENTER`, `SND_ALL` and `FLD_READER` combinations:

SND_ENTER	SND_ALL	FLD_READER	Description
0	0	0	Field cannot be filled with data from laser- or external scanner. If the user presses the OK key on this field, nothing is sent to the server, only the focus moves to the next field.
0	0	1	Field can be filled with data from laser- or external scanner, but the data is not automatically sent to the server. If the user presses the OK key on this field, nothing is sent to the server, only the focus moves to the next field.
0	1	0	Unused. The Terminal acts as if SND_ENTER bit is set.
0	1	1	Unused. The Terminal acts as if SND_ENTER bit is set.
1	0	0	Field cannot be filled with data from laser- or external scanner. In case the user presses the OK key on this field, the data entered by the user via terminal keyboard is sent to the server.
1	0	1	Field is automatically sent to the server either by reading data from laser- or external scanner or by pressing the OK key on this field.
1	1	0	Field cannot be filled with data from laser- or external scanner. In case the user presses the OK key on this field, all the fields on this form are sent to the server, except fields marked as NO_SEND.
1	1	1	All the fields on the form (except the NO_SEND fields) are sent to the server either by reading data from laser- or external scanner or by pressing the OK key on this field, even if the NO_SEND bit is set on this field.

RECEIVER

This command makes the Terminal to listen messages from the Host continuously. This mode is meant primarily for sending the pop-up messages and beeps, because other commands can disturb other activities on the Terminal GUI. However, using other commands in receiver mode is not prevented in any way, it is left to the responsibility of the Host application developer to use the commands in a way they do not mess up the GUI.

When `RECEIVER` mode is ON, **pop-up messages must be acknowledged as read by the user by pressing the DEL key** (instead of pressing any key in normal mode). This prevents accidental key press before reading the pop-up message.

Syntax: `1C | MODE`

`MODE` - Receiver mode.

The `MODE` byte is defined as follows:

Value	Description
0	RECEIVER mode is OFF
1 or 3	RECEIVER mode is ON. After receiving an asynchronous message from Host, it will be acknowledged with RECEIVER_ACK message.
2 or 4	RECEIVER mode is ON. Asynchronous messages from host will not be acknowledged.

Note that `RECEIVER` modes 1 and 3 are handled equally by Terminal, as well as modes 2 and 4. The Terminal recognises all these `RECEIVER` mode values (0, 1, 2, 3 and 4) to be compatible with Host software written for other types of Hand Terminals using proprietary radio link for communication.

FORM_ID

Sets an ID for the current form. If the FORM_ID is defined, it will be included at the beginning of the data part of every FIELD_DATA message from Terminal to Host. FORM_ID = 0x0000 will disable the FORM_ID.

Syntax: `1D | ID_H | ID_L`

`ID_H` - High byte of FORM_ID

`ID_L` - Low byte of FORM_ID

STATIC_TEXT

All bytes $\geq 0x20$ are treated as static text and written to the current cursor position. The cursor position increases by 1 after the operation.

Messages from terminal to host

Message frame structure

SOH	LENGTH	PREFIX_A	PREFIX_B	ID_H	ID_L	ID_X	DATA	CRC_H	CRC_L
Field	Length (bytes)	Values							
SOH	1	Start of header (fixed value 0x01)							
LENGTH	1	Number of bytes located in DATA field							
PREFIX_A	1	bit 0 - "1" indicates that message is repeated by a repeater							
		bit 1 - "0" indicates that message goes from terminal to Host							
		bit 2 - "0" when receiver mode is ON, "1" when receiver mode is OFF							
		bit 3 - "0" Normal message, transaction has been started from the hand terminal.							
		bit 4-7 - Value indicates the field strength of the radio signal							
PREFIX_B	1	bit 0-3 - Message number (0...15). Incremented by 1 after a successful transaction							
		bit 4-6 - Terminal battery level							
		bit 7 - Reserved							
ID_H	1	High byte of the terminal serial number							
ID_L	1	Low byte of the terminal serial number							
ID_X	1	Extra ID, that can be set by user							
DATA		Contains actual data and the commands. ASCII values from 0x0E ... 0x1F are reserved for the commands and should not be sent within a command if the protocol specifications states otherwise							
CRC_H	1	High byte of the checksum							
CRC_L	1	Low byte of the checksum							

List of commands

CmdCode	Command Name	Length (bytes)	Syntax	Description
0x0A 0x01	KEYSTROKE	3	0A 01 KEY	User pressed KEY that was previously set to Direct Mode
0x0C	RFID_DATA	2 + L	0C L DATA	Data read from the RFID tag's memory.
0x10	FIELD_DATA	3 + L	10 POS L DATA	Field data from the terminal
0x11	SERIAL_DATA	2 + L	11 L DATA	Not implemented
0x18	PING	1	18	The terminal sends this command after DELAY has expired since receiving the SEND_PING command from the host.
0x1C	RECEIVER_ACK	1	1C	Terminal has received a message from the HOST that was sent independently while the terminal was in RECEIVER mode
0x1D	FORM_ID	3	1D ID_H ID_L	If FORM_ID is defined by the host, the terminal includes the FORM_ID in every FIELD_DATA command
[none]	FUNCTION_KEY	2 or 3	46 F_NUM	User pressed a Function key.

KEYSTROKE

Terminal reports that user pressed a key that was previously set to direct mode by the Host using the `SENDKEYS` command.

Syntax: `0A | 01 | KEY`

`KEY` values are defined in the following table:

KEY Value	Physical key
1	F1
2	F2
3	F3
4	F4
5	F5
6	SCAN
7	DEL
8	Special Characters (lower left corner)
9	[unused]
10	UP arrow
11	DOWN arrow
12	OK
13	0
14	1
15	2
16	3
17	4
18	5
19	6
20	7
21	8
22	9

RFID_DATA

This is the response to the `RFID_READ_PAGES` command.

Syntax: `0C | L | DATA`

`L` - Data length.

`DATA` - Data from tag's memory.

Data length depends on the tag's memory structure and on the `FIRST` and `LAST` fields in the server request.

FIELD_DATA

The Terminal uses this command to submit data from input fields to the Host.

Syntax: `10 | POS | L | DATA`

`POS` - Position of input field.

`L` - Data length.

`DATA` - Data from input field.

SERIAL_DATA

Not implemented.

PING

The Terminal sends this command to the Host in case the Host has requested it by using the `SEND_PING` command.

Syntax: `18`

RECEIVER ACK

Terminal has received a message from the HOST that was sent independently while the terminal was in `RECEIVER` mode.

Syntax: `1C`

Note that this message is sent only in case the Terminal is in `RECEIVER` mode 1 or 3.

FORM_ID

If the `FORM_ID` is defined by the Host, this command is included at the beginning of data part in every `KEYSTROKE`, `FIELD_DATA`, `PING`, `RECEIVER_ACK` and `FUNCTION_KEY` message to the Host.

Syntax: `ID | ID_H | ID_L`

`ID_H` - High byte of `FORM_ID`.

`ID_L` - Low byte of `FORM_ID`.

FUNCTION_KEY

User pressed one of the Function keys (F1...F10).

Syntax: `46 | F_NUM`

Note that this command does not have a special command code, the Terminal just sends the Function key that was pressed by the user as ASCII text. Possible values are as follows:

ASCII	HEX
F1	46 31
F2	46 32
F3	46 33
F4	46 34
F5	46 35
F6	46 36
F7	46 37
F8	46 38
F9	46 39
F10	46 31 30