



KEYBOARD SERVICE

INDEX

INTRODUCTION	3
COMMANDS	4
XPEAK_COMMAND_CANCEL_READ (v.0706)	5
XPEAK_COMMAND_GET_CAPABILITIES (v.0706)	6
XPEAK_COMMAND_READ (v.0706)	8

INTRODUCTION

This documentation details the specific set of commands for keyboard devices. These, along with *common commands* conform the complete set of commands available for keyboard devices. Sometimes, certain *common commands* can be overwritten within a specific service, because they change their behavior. In the case of **Keyboard Service**, following commands are overwritten:

- [XPEAK_COMMAND_GET_CAPABILITIES](#)

All commands described here meet the *xpeak* specification [General Message Format](#)

COMMANDS

 **XPEAK_COMMAND_CANCEL_READ** (0x706013D)

Version: 0706

Description:

Disables the read operation of the keyboard reader.

 **Result:**

See Common Results

XPEAK_COMMAND_GET_CAPABILITIES (0x7060017)

Version: 0706

Description:

Returns the device capabilities. Depending on them, the application behavior should be different and should be adapted to the peripheral characteristics.

Result:

- **boolean SecureModeSupported**
If *false*, the keyboard only can be used in clear mode. If *true*, the keyboard can be used in secure mode for operations combined with a security service.
- **boolean CanSendKeyPressedEvents**
If *true*, the keyboard can send a *KeyPressed* when a key is pressed during the execution of the [XPEAK_COMMAND_READ](#) command. If *false*, no *KeyPressed* events will be thrown.
- **boolean AutoDisplay**
If *true*, the keyboard writes automatically the value of the pressed keys on a Display during the execution of the command [XPEAK_COMMAND_READ](#). If *false*, the pressed keys are not displayed.
- **Unicode SupportedCharacterKeys**
Contains the character keys supported by the keyboard. Every character key is represented by its Unicode value. The length of this field will be determined by the number of character keys supported by the keyboard.
- **int[] SupportedControlKeys**
Contains the control keys supported by the keyboard.

Possible values are:

- XPEAK_KEYBOARD_CONTROL_KEY_ENTER (0x70600F0)
- XPEAK_KEYBOARD_CONTROL_KEY_CLEAR (0x70600F1)
- XPEAK_KEYBOARD_CONTROL_KEY_BACKSPACE (0x70600F2)
- XPEAK_KEYBOARD_CONTROL_KEY_TAB (0x70600F3)
- XPEAK_KEYBOARD_CONTROL_KEY_INSERT (0x70600F4)
- XPEAK_KEYBOARD_CONTROL_KEY_SUPR (0x70600F5)
- XPEAK_KEYBOARD_CONTROL_KEY_ESC (0x70600F6)
- XPEAK_KEYBOARD_CONTROL_KEY_F1 (0x70600F7)
- XPEAK_KEYBOARD_CONTROL_KEY_F2 (0x70600F8)
- XPEAK_KEYBOARD_CONTROL_KEY_F3 (0x70600F9)
- XPEAK_KEYBOARD_CONTROL_KEY_F4 (0x70600FA)
- XPEAK_KEYBOARD_CONTROL_KEY_F5 (0x70600FB)

- XPEAK_KEYBOARD_CONTROL_KEY_F6 (0x70600FC)
- XPEAK_KEYBOARD_CONTROL_KEY_F7 (0x70600FD)
- XPEAK_KEYBOARD_CONTROL_KEY_F8 (0x70600FE)
- XPEAK_KEYBOARD_CONTROL_KEY_F9 (0x70600FF)
- XPEAK_KEYBOARD_CONTROL_KEY_F10 (0x7060100)
- XPEAK_KEYBOARD_CONTROL_KEY_F11 (0x7060101)
- XPEAK_KEYBOARD_CONTROL_KEY_F12 (0x7060102)
- XPEAK_KEYBOARD_CONTROL_KEY_ARROW_UP (0x7060108)
- XPEAK_KEYBOARD_CONTROL_KEY_ARROW_DOWN (0x7060109)
- XPEAK_KEYBOARD_CONTROL_KEY_ARROW_LEFT (0x706010A)
- XPEAK_KEYBOARD_CONTROL_KEY_ARROW_RIGHT (0x706010B)

- **int BeepOnPress**

This value indicates the capability of the keyboard to beep when a key is pressed during the execution of the command [XPEAK_COMMAND_READ](#). Possible values are:

Option	Description
XPEAK_KEYBOARD_BEEP_ON_PRESS_ALWAYS (0x70600EB)	This value indicates that the keyboard always beeps when a key is pressed. The parameter <i>BeepOnPress</i> of the command XPEAK_COMMAND_READ is not used, since it is not possible to disable the beep on key press.
XPEAK_KEYBOARD_BEEP_ON_PRESS_NEVER (0x70600EC)	This value indicates that the keyboard never beeps when a key is pressed. The parameter <i>BeepOnPress</i> of the command XPEAK_COMMAND_READ is not used, since it is not possible to enable the beep on key press.
XPEAK_KEYBOARD_BEEP_ON_PRESS_CONFIGURABLE (0x70600ED)	This value indicates that the keyboard can beep or not depending on the value of the parameter <i>BeepOnPress</i> of the command XPEAK_COMMAND_READ .

XPEAK_COMMAND_READ (0x7060106)

Version: 0706

Description:

This command enables the keyboard to read pressed keys and finishes immediately. The service continues reading the keys pressed by the user during the time specified in the parameter *Timeout*. The service should empty the pressed keys buffered before the execution of the command and read the keys pressed until the reading operation ends.

The reading operation ends when the command [XPEAK_COMMAND_CANCEL_READ](#) is executed, when a `TerminateKey` is pressed, or when the Maximum Length has been reached and the flag *AutoEnd* is enabled. The command can also finish when the *Timeout* expires.

The events `XPEAK_RESULT_CHARACTER_KEY_PRESSED` (0x706010C) and `XPEAK_RESULT_CONTROL_KEY_PRESSED` (0x7060107) will be thrown when a key is pressed. This events will be thrown only if the parameter *SendEvents* is enabled.

This command will return no data. When the reading operation ends, the event `XPEAK_RESULT_DATA_READ` (0x7060142) will be thrown containing an Unicode value with the keys buffered during the execution of the operation.

Parameters:

- **Unicode ActiveCharacterKeys**

Specifies the enabled character keys during the execution of the command represented by their Unicode value.

The character keys not contained in this field will not take effect if they are pressed during the execution of the command. All the keys of this parameter have to be included in the *SupportedCharacterKeys* field of the command [XPEAK_COMMAND_GET_CAPABILITIES](#).

An empty field indicates that every supported character key is active during the execution of this command.

- **int[] ActiveControlKeys**

Contains the enabled control keys during the execution of the command.

The control keys not contained in this array, will not take effect if they are pressed during the execution of the command. All the keys of this parameter have to be included in the *SupportedControlKeys* response of the command [XPEAK_COMMAND_GET_CAPABILITIES](#).

An empty array indicates that every supported control key is active during the execution of this command.

- **boolean BeepOnPress**

If *true*, the keyboard will beep when a key is pressed. If this value is *false* no beep will sound during the execution of the command.

This value will take effect only when the value of *BeepOnPress* of the command `XPEAK_COMMAND_GET_CAPABILITIES` has the value `XPEAK_KEYBOARD_BEEP_ON_PRESS_CONFIGURABLE` (0x70600ED).

- **boolean SendEvents**

If *true*, the service will send a *KeyPressed* event each time an active key is pressed.

This value will take effect only when the value of *CanSendKeyPressedEvents* of the command `XPEAK_COMMAND_GET_CAPABILITIES` has the value *true*.

- **int Timeout**

Specifies the timeout, expressed in milliseconds, while the keyboard will read keys. A value less than 0 indicates that no timeout is applied. When the timeout expires, the command will finish with the result `XPEAK_RESULT_TIMEOUT` (0x7060002).

Events:

`XPEAK_RESULT_CONTROL_KEY_PRESSED` (0x7060107) Since 0706

This event is thrown when a active control key is pressed. This event will be thrown only if the parameter *SendEvents* is *true*. In this case, the event will be thrown even if the *MaximumLength* has been reached.

- **int ControlKeyPressed**

The value of the control key. This value has to be any of the values included in the parameter *ActiveControlKeys*.

`XPEAK_RESULT_CHARACTER_KEY_PRESSED` (0x706010C) Since 0706

This event is thrown when a character key is pressed. This event will be thrown only if the parameter *SendEvents* has the value *true* and the *MaximumLength* has not been reached.

- **Unicode CharacterKeyPressed**

Contains the Unicode value of the pressed character key. The length of this Unicode field has to be 1. The value of the pressed key has to be included in the parameter *ActiveCharacterKeys*, otherwise the event will not be sent.

If the value of the parameter *Mode* is `XPEAK_KEYBOARD_READ_MODE_PIN` (0x7060104) or `XPEAK_KEYBOARD_READ_MODE_MANUAL_KEY` (0x7060105), the event will contain the Unicode character '*' instead of the pressed key.

Some keyboards include keys with several characters, for example "00" or "000". In that case the

service will send as many events as necessary. For example, if "00" is pressed, two events with the Unicode value '0' will be sent.

XPEAK_RESULT_DATA_READ (0x7060142) Since 0706

This event is thrown when a the reading operation finishes. It contains the keys pressed during the operation.

The reading operation ends when one of the premises of the following table occurs.

Condition	Event thrown
A <i>TerminateKey</i> is pressed	XPEAK_RESULT_DATA_READ (0x7060142)
The <i>MaximumLength</i> has been reached and <i>AutoEnd</i> is enabled	XPEAK_RESULT_DATA_READ (0x7060142)
The command XPEAK_COMMAND_CANCEL_READ is executed	XPEAK_RESULT_DISABLED (0x7060173)
The <i>Timeout</i> has expired	XPEAK_RESULT_TIMEOUT (0x7060002)
A hardware error occurs	XPEAK_RESULT_HARDWARE_ERROR (0x7060059)
A connection error occurs	XPEAK_RESULT_CONNECTION_ERROR (0x706005B)
A user atcion error occurs	XPEAK_RESULT_USER_ACTION_ERROR (0x706005A)

- **Unicode PressedCharacterKeys**

Returns a Unicode value with the character keys buffered during the execution of the reading operation. The length of this value will be determined by the number of keys pressed and it will not be longer than the *MaximumLength* parameter.

Every returned key must be included in the parameter *ActiveCharacterKeys*.

- **int TerminateControlKey**

Returns the value of the terminate control key that caused the end of the reading operation. If the operation finished due to any other reason different from a terminate control key pressed, the value of this result will be XPEAK_KEYBOARD_CONTROL_KEY_NONE (0x706010D).

Only the constants contained in the parameter *TerminateControlKeys* and the constant XPEAK_KEYBOARD_CONTROL_KEY_NONE (0x706010D) are allowed in this field.