Embedded USB Host Fujitsu FTP-6xx Printer Class Driver User Guide

Version 1.00

For use with USBH Class Driver for Fujitsu FTP-6xx Printer module versions 1.01 and above

Exported on 02/12/2019

Table of Contents

1 System Overview

This chapter contains the fundamental information for this module.

The component sections are as follows:

- Introduction – describes the main elements of the module.
- Feature Check – summarizes the main features of the module as bullet points.
- Packages and Documents – the *Packages* section lists the packages that you need in order to use this module. The *Documents* section lists the relevant user guides.
- Change History – lists the earlier versions of this manual, giving the software version that each manual describes.

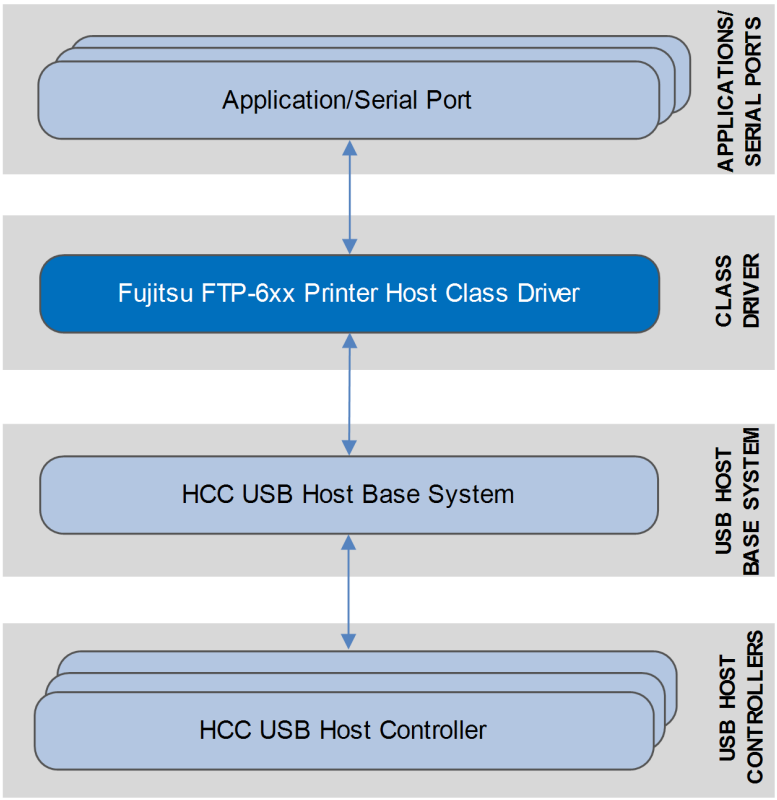
1.1 Introduction

This guide is for those who want to implement an embedded USB printer host class driver to control Fujitsu FTP-6xx printer USB devices.

The package provides a Fujitsu FTP-6xx printer host class driver for a USB stack. Two extra packages are provided free with this to support the following common printer languages:

- ESC/POS® – an Epson® language supported by a variety of printers.
- PDL PCL 5 – printers from many different manufacturers support this Hewlett-Packard Page Description Language (PDL).

The system structure is shown in the diagram below:



This shows how the lower layer interface of the host class driver is designed to use HCC Embedded's USB Host Interface Layer. This layer is standard over different host controller implementations; this means that the code is unchanged, whichever HCC USB host controller it is interfaced to.

Note: For detailed information about this layer, refer to the HCC USB Host Base System User Guide that is shipped with the base system.

The package provides a set of API functions for controlling access to a Fujitsu FTP6xx printer. These are described here, with separate sections for module and printer management.

1.2 Feature Check

The main features of the class driver are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Designed for integration with both RTOS and non-RTOS based systems.
- Compatible with all HCC USB host controllers.
- Supports the ESC/POS® and PDL PCL 5 printer languages.
- Supports multiple devices connected simultaneously.
- Supports all Fujitsu FTP-6xx printers that comply with the USB printer specification.

1.3 Packages and Documents

Packages

The table below lists the packages that you need in order to use this module:

Package	Description
hcc_base_doc	This contains the two guides that will help you get started.
usbh_base	The USB host base package. This is the framework used by USB class drivers to communicate over USB using a specific USB host controller package.
usbh_cd_printer_fujitsu_ftp6xx	The USB device printer host class driver package described by this document.
usbh_cd_printer_pdl_escpos	The PDL ESC/POS® printer class driver package. This is provided with the **fujitsu_ftp6xx** package and described by this document.
usbh_cd_printer_pdl_pcl5	The PDL PCL 5 printer class driver package. This is provided with the **fujitsu_ftp6xx** package and described by this document.

Documents

For an overview of HCC's embedded USB stacks, see Product Information on the main HCC website.

Readers should note the points in the HCC Documentation Guidelines on the HCC documentation website.

HCC Firmware Quick Start Guide

This document describes how to install packages provided by HCC in the target development environment. Also follow the *Quick Start Guide* when HCC provides package updates.

HCC Source Tree Guide

This document describes the HCC source tree. It gives an overview of the system to make clear the logic behind its organization.

HCC USB Host Base System User Guide

This document defines the USB host base system upon which the complete USB stack is built.

HCC Embedded USB Host Class Driver for Fujitsu FTP-6xx Printer User Guide

This is this document.

1.4 Change History

This section describes past changes to this manual.

- To view or download manuals, see USB Host PDFs.
- For the history of changes made to the package code itself, see History: usbh_cd_printer_fujitsu_ftp6xx, History: usbh_cd_printer_pdl_escpos, and History: usbh_cd_printer_pdl_pcl5.

The current version of this manual is 1.00.

Manual version	Date	Software version	Reason for change
1.00	2019-02-12	1.01	First online release.

2 Source File List

This section describes all the source code files included in the system. These files follow the HCC Embedded standard source tree system, described in the HCC Source Tree Guide. All references to file pathnames refer to locations within this standard source tree, not within the package you initially receive.

Note: Do not modify any files except the configuration file.

2.1 API Header Files

These files in **src/api** within each package are the only files that should be included by an application using this module. **These files should only be modified by HCC**.

File	Package	Description
api_usbh_printer_fujitsu_ftp6xx.h	Fujitsu	Defines the main API functions; see Application Programming Interface.
api_usbh_printer_pdl_escpos.h	ESC/POS	ESC/POS® API functions.
api_usbh_printer_pdl_pcl5.h	PDL PCL5	PDL PCL 5 API functions.

2.2 Configuration File

The file **src/config/config_usbh_printer_fujitsu_ftp6xx.h** contains the configurable parameters. Configure these as required. For details of the options, see Configuration Options.

2.3 Source Code Files

The source code files are in **src/usb-host/class-drivers/printer** within each package. **These files should only be modified by HCC**.

File	Package	Description
usbh_printer_fujitsu_ftp6xx.c	Fujitsu	Defines the main elements.
pdl/usbh_printer_pdl_escpos.c	ESC/POS	Defines the ESC/POS® elements.
pdl/usbh_printer_pdl_pcl5.c	PDL PCL5	Defines the PDL PCL 5 elements.

2.4 Version Files

These files in **src/version** within each package contain the components' version numbers. The version number is checked by all the modules that use this module to ensure system consistency over upgrades.

File	Package	Description
ver_usbh_printer_fujitsu_ftp6xx.h	Fujitsu	Main package version.
ver_usbh_printer_pdl_escpos.h	ESC/POS	ESC/POS® version.
ver_usbh_printer_pdl_pcl5.h	PDL PCL5	PDL PCL 5 version.

3 Configuration Options

Set the system configuration options in the file **src/config/config_usbh_printer_fujitsu_ftp6xx.h**. This section lists the available options and their default values.

USBH_PRINTER_FUJITSU_FTP6XX_MAX_UNITS

The number of printers supported. The default is 2.

USBH_PRINTER_FUJITSU_FTP6XX_VID

The vendor ID, based on which device is identified. The default is 0x0430.

USBH_PRINTER_FUJITSU_FTP6XX_PID

The product ID, based on which the device is identified. The default is 0x0623.

USBH_PRINTER_FUJITSU_FTP6XX_ERROR_SET_FLAGS

This defines the flags. If any flag is in the set state, printing is not possible.

The default is: `( PRN_FUJITSU_FTP6XX_ERROR_SUM | PRN_FUJITSU_FTP6XX_PAPER_OUT )`

USBH_PRINTER_FUJITSU_FTP6XX_ERROR_CLEARED_FLAGS

This defines the flags. If any flag is in the cleared state, printing is not possible. The default is 0.

4 Application Programming Interface

This section documents the Application Programming Interface (API). It includes all the functions that are available to an application program.

4.1 Module Management

The functions are the following:

Function	Description
usbh_prn_fujitsu_ftp6xx_init()	Initializes the module and allocates the required resources.
usbh_prn_fujitsu_ftp6xx_start()	Starts the module.
usbh_prn_fujitsu_ftp6xx_stop()	Stops the module.
usbh_prn_fujitsu_ftp6xx_delete()	Deletes the module and releases the resources it used.

usbh_prn_fujitsu_ftp6xx_init

Use this function to initialize the class driver and allocate the required resources.

> **Note:** You must call this before any other function.

Format

```
int usbh_prn_fujitsu_ftp6xx_init ( void )
```

Arguments

Parameter
None.

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_start

Use this function to start the class driver.

> **Note:** Call **usbh_prn_fujitsu_ftp6xx_init()** before this function.

Format

```
int usbh_prn_fujitsu_ftp6xx_start ( void )
```

Arguments

Parameter
None.

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_stop

Use this function to stop the class driver.

Format

```
int usbh_prn_fujitsu_ftp6xx_stop ( void )
```

Arguments

Parameter
None.

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_delete

Use this function to delete the class driver and release the associated resources.

Format

```
int usbh_prn_fujitsu_ftp6xx_delete ( void )
```

Arguments

Parameter
None.

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

4.2 Printer Management

The functions are the following:

Function	Description
usbh_prn_fujitsu_ftp6xx_send()	Sends data to the printer.
usbh_prn_fujitsu_ftp6xx_receive()	Gets data from the printer.
usbh_prn_fujitsu_ftp6xx_read_id()	Gets an IEEE 1284 ID string from the printer.
usbh_prn_fujitsu_ftp6xx_print()	Prints the created job.
usbh_prn_fujitsu_ftp6xx_soft_reset()	Sends a soft reset command to the printer.
usbh_prn_fujitsu_ftp6xx_get_port_hdl()	Gets the printer port handle.
usbh_prn_fujitsu_ftp6xx_get_port_status()	Gets the current status of a printer.
usbh_prn_fujitsu_ftp6xx_present()	Checks whether a printer is connected.
usbh_prn_fujitsu_ftp6xx_register_ntf()	Registers a notification function for a specified event type.

usbh_prn_fujitsu_ftp6xx_send

Use this function to send data to the printer.

Format

```
int usbh_prn_fujitsu_ftp6xx_send (
    t_usbh_unit_id   uid,
    uint8_t *        buffer,
    uint32_t         length,
    uint32_t *       slength )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id
buffer	The data to send.	uint8_t *
length	The number of bytes to send.	uint32_t
slength	On return, the number of bytes sent.	uint32_t *

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_receive

Use this function to get data from the printer.

Format

```
int usbh_prn_fujitsu_ftp6xx_receive (
    t_usbh_unit_id   uid,
    uint8_t *        buffer,
    uint32_t         length,
    uint32_t *       rlength )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id
buffer	Where to put the incoming data	uint8_t *
length	The number of bytes to receive.	uint32_t
rlength	On return, the number of bytes received.	uint32_t *

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_read_id

Use this function to get an IEEE 1284 ID string from the printer.

The first two bytes in the string are the string length in big-endian format.

Format

```
int usbh_prn_fujitsu_ftp6xx_read_id (
    t_usbh_unit_id   uid,
    uint8_t *        buffer,
    uint32_t         length,
    uint32_t *       rlength )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id
buffer	The buffer to hold the ID.	uint8_t *
length	The length of the buffer.	uint32_t
rlength	On return, the number of bytes received.	uint32_t *

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_print

Use this function to print the created job.

Format

```
int usbh_prn_fujitsu_ftp6xx_print (
    t_usbh_unit_id   uid,
    uint8_t *        buffer,
    uint32_t         length )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id
buffer	A pointer to the data to print.	uint8_t *
length	The number of bytes to print.	uint32_t

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_soft_reset

Use this function to send a soft reset command to the printer.

Note: Call this function after any communication error.

Format

```
int usbh_prn_fujitsu_ftp6xx_soft_reset ( t_usbh_unit_id uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_get_port_hdl

Use this function to get the printer port handle.

Format

```
t_usbh_port_hdl usbh_prn_fujitsu_ftp6xx_get_port_hdl ( t_usbh_unit_id uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id

Return Values

Return value	Description
The port handle.	Successful execution.
USBH_PORT_HDL_INVALID	Invalid port handle.

usbh_prn_fujitsu_ftp6xx_get_port_status

Use this function to get the current status of a printer.

Format

```
int usbh_prn_fujitsu_ftp6xx_get_port_status (
    t_usbh_unit_id   uid,
    uint8_t *        status )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id
status	Where to put the status bitfield.	uint8_t *

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_present

Use this function to check whether a printer is present.

Format

```
int usbh_prn_fujitsu_ftp6xx_present ( t_usbh_unit_id uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id

Return Values

Return value	Description
0	No printer is present.
1	A printer is present.
Else	See Error Codes.

usbh_prn_fujitsu_ftp6xx_register_ntf

Use this function to register a notification function for a specified event notification type.

When a device is connected or disconnected, the notification function is called.

Note: It is the user's responsibility to provide any notification functions required by the application. Providing such functions is optional.

Format

```
int usbh_prn_fujitsu_ftp6xx_register_ntf (
    t_usbh_unit_id   uid,
    t_usbh_ntf       ntf,
    t_usbh_ntf_fn    ntf_fn )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id
ntf	The notification type.	t_usbh_ntf
ntf_fn	The notification function to be used when an event occurs.	t_usbh_ntf_fn

Return Values

Return value	Description
USBH_SUCCESS	Successful execution.
Else	See Error Codes.

4.3 ESC POS Management

This section describes the ESC/POS® printer functions. Using this API, a simple text-only ESC/POS® document can be generated in RAM.

The functions are the following:

Function	Description
escpos_init()	Initializes the module and allocates the required resources.
escpos_start_job()	Starts a print job.
escpos_end_job()	Closes a print job.
escpos_set_line_spacing()	Sets the number of lines printed in one inch. This sets the vertical spacing between lines.
escpos_default_spacing()	Sets the vertical line spacing to the default.
escpos_underline_on()	Turns underlining on.
escpos_underline_off()	Turns underlining off.
escpos_emphasize_on()	Turns emphasis (bold text) on.
escpos_emphasize_off()	Turns emphasis (bold text) off.
escpos_set_char_size()	Sets the number of lines printed in one inch. This sets the vertical spacing between lines.
escpos_set_font()	Sets the font.
escpos_print_text()	Prints the contents of a buffer.
escpos_cut_paper()	Cuts the paper.
escpos_set_barcode_height()	Sets the bar code height in dots.
escpos_set_barcode_width()	Sets the bar code width in dots.
escpos_print_barcode()	Prints a bar code.

escpos_init

Use this function to initialize the ESC/POS® module.

> **Note:** You must call this before any other function.

Format

```
int escpos_init ( void )
```

Arguments

Parameter
None.

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_start_job

Use this function to start a printing job.

This call resets the printer to its default state. Font selections, page properties, and so on all return to their default state.

Format

```
int escpos_start_job ( escpos_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_end_job

Use this function to close a print job.

Format

```
int escpos_end_job (
    escpos_buffer_t *   buf,
    uint8_t             feed_lines )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *
feed_lines	The number of line feeds to output after the print.	uint8_t

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_set_line_spacing

Use this function to set the number of lines printed in one inch. This sets the vertical spacing between lines.

Format

```
int escpos_set_line_spacing (
    escpos_buffer_t *   buf,
    uint8_t             line_spacing )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *
line_spacing	Lines printed per inch. 1 is equal to 30 dots. This is 4.23 mm or 1/6 inch.	uint8_t

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_default_spacing

Use this function to set the vertical line spacing to the default.

Format

```
int escpos_default_spacing ( escpos_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_underline_on

Use this function to turn underlining on.

Format

```
int escpos_underline_on ( escpos_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_underline_off

Use this function to turn underlining off.

Format

```
int escpos_underline_off ( escpos_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_emphasize_on

Use this function to turn emphasis (bold text) on.

Format

```
int escpos_emphasize_on ( escpos_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_emphasize_off

Use this function to turn emphasis (bold) off.

Format

```
int escpos_emphasize_off ( escpos_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_set_char_size

Use this function to set the number of lines printed in one inch. This sets the vertical spacing between lines.

Format

```
int escpos_set_char_size (
    escpos_buffer_t *   buf,
    uint8_t             width,
    uint8_t             height )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *
width	1 = normal character width, 2 = double size, and so on.	uint8_t
height	1 = normal character height, 2 = double size, and so on.	uint8_t

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_set_font

Use this function to set the font.

Format

```
int escpos_set_font (
    escpos_buffer_t *   buf,
    uint8_t             font)
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *
width	0 = font A, 1 = font B, and so on.	uint8_t

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_print_text

Use this function to print the contents of a buffer.

Format

```
int escpos_print_text (
    escpos_buffer_t *   buf,
    char *              text)
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *
text	The text to print.	char *

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_cut_paper

Use this function to cut the paper.

Format

```
int escpos_cut_paper ( escpos_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_set_barcode_height

Use this function to set the bar code height in dots.

The range for the TM-T88V printer is 1 .. 255. The default is 162.

Format

```
int escpos_set_barcode_height (
    escpos_buffer_t *   buf,
    uint8_t             height )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *
height	The bar code height in dots.	uint8_t

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_set_barcode_width

Use this function to set the bar code width in dots.

The range for the TM-T88V printer is 2 .. 6. The default is 3.

Format

```
int escpos_set_barcode_width(
    escpos_buffer_t *   buf,
    uint8_t             width )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	escpos_buffer_t *
width	The bar code width in dots.	uint8_t

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

escpos_print_barcode

Use this function to print a bar code.

Valid combinations of bar code types, bar code data and length are described in the printer's documentation.

Format

```
int escpos_print_barcode (
    escpos_buffer_t *   buf,
    t_barcode           type,
    uint8_t *           barcode,
    uint8_t             length )
```

Arguments

Parameter	Description	Type
buf	A pointer to the destination buffer.	escpos_buffer_t *
type	The bar code type. Refer to the GS k command.	t_barcode
barcode	A pointer to the bar code buffer to print.	uint8_t *
length	The length of the bar code buffer.	uint8_t

Return values

Return Value	Description
ESCPOS_SUCCESS	Successful execution.
ESCPOS_ERR_BUF_FULL	The buffer is full.

4.4 PDL PCL 5 Management

PCL 5 is a Printer Control Language that is understood by many printers. These functions can be used to format print output that conforms to this standard.

Function	Description
pcl5_init()	Initializes the module and allocates the required resources.
pcl5_start_job()	Starts a printing job.
pcl5_end_job()	Closes a printing job.
pcl5_set_col()	Sets the column the next character is printed to.
pcl5_set_row()	Sets the row the next character is printed to.
pcl5_set_l_margin()	Sets the left margin to the specified column.
pcl5_set_r_margin()	Sets the right margin to the specified column.
pcl5_set_t_margin()	Sets the top margin of the page.
pcl5_set_fixed_spacing()	Switches to fixed spacing characters.
pcl5_set_proportional_spacing()	Switches to proportional spacing characters.
pcl5_set_fixed_pitch()	Sets the number of fixed space characters per inch.
pcl5_set_proportional_height()	Sets the height of proportional characters.
pcl5_line_spacing()	Sets the number of lines printed in one inch. This sets the space between lines.
pcl5_set_style()	Sets the character style.
pcl5_set_underline_on()	Turns underlining on.
pcl5_set_underline_off()	Turns underlining off.
pcl5_set_weight()	Sets the stroke weight.
pcl5_select_symbol_set()	Selects the symbol set.
pcl5_eject_page()	Issues a page eject command.
pcl5__print_text()	Copies a specified zero-terminated string to a PCL 5 buffer.
pcl5_print_transparent()	Copies characters with the specified length to the PCL 5 buffer.
pcl5_img_start()	Starts an image printing job.

Function	Description
pcl5_img_rgbdata()	Copies image data into a PCL5 buffer.
pcl5_img_finish()	Completes an image printing job.

pcl5_init

Use this function to initialize the PCL 5 module.

Note: You must call this before any other PCL 5 function.

Format

```
int pcl5_init ( void )
```

Arguments

Parameter
None.

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_start_job

Use this function to start a printing job.

This call resets the printer to its default state. Font selections, page properties, and so on all return to their default state.

Format

```
int pcl5_start_job ( pcl5_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_end_job

Use this function to close a job.

Format

```
int pcl5_end_job ( pcl5_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_col

Use this function to set the column the next character is printed to.

Format

```
int pcl5_set_col (
    pcl5_buffer_t *   buf,
    int               col )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
col	The column number.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_row

Use this function to set the row the next character is printed to.

Format

```
int pcl5_set_row (
    pcl5_buffer_t *   buf,
    int               row )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
row	The row number.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_l_margin

Use this function to set the left margin to the specified column.

Any Carriage Returns used will return to this position.

Format

```
int pcl5_set_l_margin (
    pcl5_buffer_t *   buf,
    int               col )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
col	The column number.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_r_margin

Use this function to set the right margin.

Depending on the printer, characters printed after this column may be lost, or wrapped to the next line.

Format

```
int pcl5_set_r_margin (
    pcl5_buffer_t *   buf,
    int               col )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
col	The column number.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_t_margin

Use this function to set the top margin of the page.

Format

```
int pcl5_set_t_margin (
    pcl5_buffer_t *   buf,
    int               row )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
row	The row number.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_fixed_spacing

Use this function to switch to fixed spacing characters.

Format

```
int pcl5_fixed_spacing ( pcl5_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_proportional_spacing

Use this function to switch to proportional spacing characters.

Format

```
int pcl5_proportional_spacing ( pcl5_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_fixed_pitch

Use this function to set the number of fixed space characters per inch.

This command has no effect on proportional characters.

Format

```
int pcl5_set_fixed_pitch (
    pcl5_buffer_t *   buf,
    int               pitch )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
pitch	The number of fixed space characters per inch.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_proportional_height

Use this function to set the height of proportional characters.

This command has no effect on fixed space characters.

Format

```
int pcl5_set_proportional_height (
    pcl5_buffer_t *   buf,
    int               height )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
height	The height of the proportional characters.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_line_spacing

Use this function to set the number of lines printed in one inch. This sets the space between lines.

Format

```
int pcl5_set_line_spacing (
    pcl5_buffer_t *   buf,
    line_spacing_t    lpi )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
lpi	Lines printed per inch.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_style

Use this function to set the character style.

See the structure *chr_style_t* in the file **api_pcl5.h** for possible styles. Not all styles are supported by all character configurations. Please check your printer's documentation for supported combinations.

Format

```
int pcl5_set_style (
    pcl5_buffer_t *   buf,
    chr_style_t       style )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
style	The style.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_underline_on

Use this function to turn underlining on.

Format

```
int pcl5_underline_on ( pcl5_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_underline_off

Use this function to turn underlining off.

Format

```
int pcl5_underline_off ( pcl5_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_set_weight

Use this function to set the stroke weight.

The weight value is in the range of -7...+7. A smaller value means a thicker line is used for characters. The default value is zero.

Format

```
int pcl5_set_weight (
    pcl5_buffer_t *   buf,
    int               weight )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
weight	The stroke weight.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_select_symbol_set

Use this function to select the symbol set.

All PCL 5 printers support multiple symbol sets. A symbol set specifies the character symbols that are available in the character table. For example, the code 0xFA in the PC-8 table is a double right arrow. The same code in the Roman-8 table is a font symbol. Symbol sets are identified by a short string. For example, the id of the PC-8 symbol set is "10U".

For symbol set details, refer to your printer's documentation.

Format

```
int pcl5_select_symbol_set (
    pcl5_buffer_t *   buf,
    char *            id )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
id	The symbol set.	char *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_eject_page

Use this function to issue a page eject command.

Format

```
int pcl5_eject_page ( pcl5_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_print_text

Use this function to copy a specified zero-terminated string to a PCL 5 buffer.

The printer control characters Line Feed and Carriage Return are not changed and are interpreted by the printer. Thus you can start a new line by printing "\r\n".

Format

```
int pcl5_print_text (
    pcl5_buffer_t *   buf,
    char *            text )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
text	The text.	char *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_print_transparent

Use this function to copy characters with the specified length to the PCL 5 buffer.

The printer does not treat printer control characters as special characters. The character in the current character table for all values will be printed.

This call can be used to print special symbols mapped to printer control characters. Examples are 0x01 (smiley) and 0x02 (reverse smiley) in the PC-8 symbol set.

Format

```
int pcl5_print_transparent (
    pcl5_buffer_t *   buf,
    char *            text,
    int               length )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to print.	pcl5_buffer_t *
col	The text.	char *
length	The length.	int

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_img_start

Use this function to start an image printing job.

This call copies the commands needed to start an image print to a PCL5 buffer.

> **Note:** You must call this before copying image data to the buffer by using **pcl5_img_rgbdata()**.

Format

```
int pcl5_img_start (
    pcl5_buffer_t *   buf,
    uint16_t          width,
    uint16_t          height,
    uint16_t          resolution )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer to copy data to.	pcl5_buffer_t *
width	The image width in pixels.	uint16_t
height	The image height in pixels.	uint16_t
resolution	The image resolution in Dots Per Inch (DPI). Valid DPI values are 75, 100, 150, 200, 300, and 600.	uint16_t

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_img_rgbdata

Use this function to copy image data into a PCL5 buffer. The image must be in standard RGB format.

Each pixel is represented by 3 x 8 bits of red, green, and blue color data. All the bytes of a given pixel must be copied before the next pixel is copied.

The top left pixel must be copied first, then remaining pixels in the first row, continuing with the remaining rows.

Format

```
int pcl5_img_rgbdata (
    pcl5_buffer_t *   buf,
    char *            src,
    uint16_t          len )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer.	pcl5_buffer_t *
src	The image width.	char *
len	The image height.	uint16_t

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

pcl5_img_finish

Use this function to complete an image printing job.

Note: You must call this after all image data is transferred with the **pcl5_img_rgbdata()**.

Format

```
int pcl5_img_finish ( pcl5_buffer_t * buf )
```

Arguments

Parameter	Description	Type
buf	A pointer to the buffer holding the image.	pcl5_buffer_t *

Return Values

Return Value	Description
PCL5_SUCCESS	Successful execution.
PCL5_ERR_BUF_FULL	The buffer is full.

4.5 Error Codes

Class Driver Error Codes

If a function executes successfully it returns with a USBH_SUCCESS code, a value of 0. The following table shows the meaning of the error codes:

Return Code	Value	Description
USBH_SUCCESS	0	Successful execution.
USBH_SHORT_PACKET	1	IN transfer completed with short packet.
USBH_PENDING	2	Transfer still pending.
USBH_ERR_BUSY	3	Another transfer in progress.
USBH_ERR_DIR	4	Transfer direction error.
USBH_ERR_TIMEOUT	5	Transfer timed out.
USBH_ERR_TRANSFER	6	Transfer failed to complete.
USBH_ERR_TRANSFER_FULL	7	Cannot process more transfers.
USBH_ERR_SUSPENDED	8	Host controller is suspended.
USBH_ERR_HC_HALTED	9	Host controller is halted.
USBH_ERR_REMOVED	10	Transfer finished due to device removal.
USBH_ERR_PERIODIC_LIST	11	Periodic list error.
USBH_ERR_RESET_REQUEST	12	Reset request during enumeration.
USBH_ERR_RESOURCE	13	OS resource error.
USBH_ERR_INVALID	14	Invalid identifier/type (HC, EP HDL, and so on).
USBH_ERR_NOT_AVAILABLE	15	Item not available.
USBH_ERR_INVALID_SIZE	16	Invalid size.
USBH_ERR_NOT_ALLOWED	17	Operation not allowed.
USBH_ERROR	18	General error.

ESC/POS Error Codes

If an ESC/POS function executes successfully, it returns with an ESCPOS_SUCCESS code, a value of 0. The following table shows the meaning of the error codes:

Return Value	Value	Description
ESCPOS_SUCCESS	0	Successful execution.
ESCPOS_ERR_BUF_FULL	1	The buffer is full.

PCL 5 Error Codes

If a PCL 5 function executes successfully, it returns with a PCL5_SUCCESS code, a value of 0. The following table shows the meaning of the error codes:

Return Value	Value	Description
PCL5_SUCCESS	0	Successful execution.
PCL5_ERR_BUF_FULL	1	The buffer is full.

4.6 Types and Definitions

This section describes the notification function, the standard notification codes, and the status bits that are defined in the API header file.

t_usbh_ntf_fn

The **t_usbh_ntf_fn** definition specifies the format of the notification function. It is defined in the USB host base system in the file **api_usb_host.h**.

Format

```
int ( * t_usbh_ntf_fn )(
    t_usbh_unit_id   uid,
    t_usbh_ntf       ntf )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	t_usbh_unit_id
ntf	The notification code.	t_usbh_ntf

Notification Codes

The standard notification codes shown below are defined in the USB host base system in the file **api_usb_host.h**. This module has no specific notification codes of its own.

Notification	Value	Description
USBH_NTF_CONNECT	1	Connection notification code.
USBH_NTF_DISCONNECT	2	Disconnection notification code.

Printer Status Bits

Note: These are unique bits in a set of bit values.

The printer information status bits are as follows:

Printer information status bits	Value	Description
PRN_FUJITSU_FTP6XX_INFO_RX_BUFFER_STATUS	1U << 1	The device has no paper.
PRN_FUJITSU_FTP6XX_INFO_OFFLINE	1U << 3	The device is selected.
PRN_FUJITSU_FTP6XX_INFO_PAPER_FEED	1U << 6	The device encountered no error.

The printer error information status bits are as follows:

Printer error status bits	Value	Description
PRN_FUJITSU_FTP6XX_ERROR_VOLTAGE	1U << 8	The voltage is incorrect.
PRN_FUJITSU_FTP6XX_ERROR_RX_DATA	1U << 9	The received data is incorrect.
PRN_FUJITSU_FTP6XX_ERROR_HEAD_UP	1U << 10	Head up error.
PRN_FUJITSU_FTP6XX_ERROR_HW	1U << 13	Hardware error.
PRN_FUJITSU_FTP6XX_ERROR_THERMAL_OR_MOTOR	1U << 14	Thermal or motor error.

The paper detect information status bits are as follows:

Printer paper status bits	Value	Description
PRN_FUJITSU_FTP6XX_PAPER_NEAR_END	1U << 16	There is little paper left.
PRN_FUJITSU_FTP6XX_PAPER_MARK_DETECTION	1U << 17	The device.
PRN_FUJITSU_FTP6XX_PAPER_OUT	1U << 18	The device has no paper.

escpos_buffer_t

The **escpos_buffer_t** structure defines the ESC/POS® buffer.

Element	Type	Description
start	char *	The position of the first character.
index	uint32_t	The index.
length	uint32_t	The length of the buffer.

pcl5_buffer_t

The *pcl5_buffer_t* structure defines the PCL buffer.

Element	Type	Description
start	char *	The position of the first character.
index	uint32_t	The index.
length	uint32_t	The length of the buffer.

5 Sample Code

This section shows example code for the class driver.

5.1 Initialization

This example shows the code used to initialize a USB host with the class driver.

```
/*
** Initialize USB host with Printer class driver.
*/

int usb_host_init ( void )
{
    int  rc;
    rc = hcc_mem_init();

    /* Initialize USB host module */
    if ( rc == USBH_SUCCESS )
    {
        rc = usbh_init();
    }

    /* Initialize specific USB host controller */
    if ( rc == USBH_SUCCESS )
    {
        rc = usbh_hc_init( 0, usbh_stm32uh_hc, 0 );
    }

    /* Initialize Printer class driver module */
    if ( rc == USBH_SUCCESS )
    {
        rc = usbh_prn_fujitsu_ftp6xx_init();
    }

    /* Start the Printer class driver */
    if ( rc == USBH_SUCCESS )
    {
        rc = usbh_prn_fujitsu_ftp6xx_start();
    }

    /* Start the USB host stack */
    if ( rc == USBH_SUCCESS )
    {
        rc = usbh_start();  /* Start the USB host */
    }

    return rc;

} /* usb_host_init */
```