
CYUSB3KIT-003 with SP605 xilinx Documentation

Release latest

Oct 08, 2021

Contents

1	Getting Started	3
1.1	Dependencies	3
1.2	Cloning repository	3
1.3	Compiling and Running	3
2	Constructor	5
3	Programming	7
3.1	Download fx3 firmware	7
3.2	Program Device	7
4	Bulk Methods	9
4.1	Send Buffer	9
4.2	Recive Buffer	9
4.3	Send Text Files	10
5	Print information	11
6	Others	13
	Index	15

Author

J. Agustin Barrachina

Version 1.0 of 2019/02/27

class FX3USB3Connection

The cpp code is a class that enables easy connect and communication with a Cypress CYUSB3KIT-003 and hopefully to other Cypress devices. The projec was done in the context of communicating with a SP605 Xilinx board but it should be useful for other applications.

Content

1.1 Dependencies

1. `sudo apt install cmake git libusb-1.0-0-dev`
2. Download and install Cypress official software for linux from [EZ-USB_FX3_Software_Development_Kit](#)

1.2 Cloning repository

1. `sudo apt install git`
2. `cd` to the folder you want to have the project
3. `git clone https://github.com/NEGU93/CYUSB3KIT-003_with_SP605_xilinx.git`

1.3 Compiling and Running

1. `sudo apt install cmake`
2. `cd fx3_manager_cpp_source/`
3. `cmake .`
4. `make`
5. `./testing_cpp_code`

CHAPTER 2

Constructor

Upon construction, the object connects to the desired Cypress device. For initializing the device there are 3 options according to the need.

FX3USB3Connection : **FX3USB3Connection** (vid, pid)

This option is used to give the device a VID and PID of the device that has to connect

FX3USB3Connection : **FX3USB3Connection** (nullptr)

When cypress is installed, a file is created on `/etc/cyusb.conf` with the description of all Cypress possible devices. Use the `nullptr` option when connection to any standard Cypress device.

FX3USB3Connection : **FX3USB3Connection** ()

FX3USB3Connection : **FX3USB3Connection** ("path/to/conf/file")

If VID and PID is not good enough, you can create a `.conf` file with the description of all the parameters the device must have. By default, `conf/device.conf` is used. If not it is necessary to give the constructor the file path.

3.1 Download fx3 firmware

```
int FX3USB3Connection::download_fx3_firmware(char *filename, char *tgt_str =  
const_cast<char *>("ram"), unsigned short vid  
= 0, unsigned short pid = 0)
```

Description Upload a .img firmware to the FX3 device

Receives

- filename: name of the firmware.img file to be programmed
- (Optional) tgt_str:
 - “ram” (Default)
 - “i2c”
 - “spi”
- (Optional) pid & vid: If after programming, the board is supposed to change both vid and pid values it must be passed as parameters.

Returns

- 0 on success
- EINVAL: if filename or tgt_str values were incorrect
- LIBUSB_ERROR code on other failures

3.2 Program Device

```
int FX3USB3Connection::program_device(char *fpga_firmware_filename)
```

Description Program FPGA with the file passed as input.

Returns

- 0 on success
- LIBUSB_ERROR_TIMEOUT if the transfer timed out
- LIBUSB_ERROR_PIPE if the control request was not supported by the device
- LIBUSB_ERROR_NO_DEVICE if the device has been disconnected
- another LIBUSB_ERROR code on other failures

4.1 Send Buffer

`int FX3USB3Connection::send_buffer (unsigned char *buf, int sz, unsigned int end_ptr = 0x01)`

Description Sends the data stored on 'buf' of size 'sz' to the cypress device

Receives

- `unsigned char *buf`: pointer to a buffer where the data to be sent is stored
- `unsigned int data_count`: Size of data to be sent
- (Optional) `unsigned int end_ptr`: endpoint number where to send the data (default 0x01)

Returns

- 0 on success (and populates transferred)
- `LIBUSB_ERROR_TIMEOUT` if the transfer timed out (and populates transferred)
- `LIBUSB_ERROR_PIPE` if the endpoint halted
- `LIBUSB_ERROR_OVERFLOW` if the device offered more data, see Packets and overflows
- `LIBUSB_ERROR_NO_DEVICE` if the device has been disconnected
- another `LIBUSB_ERROR` code on other failures

4.2 Recive Buffer

`int FX3USB3Connection::receive_buffer(unsigned char *buf, unsigned int data_count, unsigned int end_ptr = 0x01)`

Description Reads data from endpoint 0x81 (default) to buf and returns

Receives

- `unsigned char *buf`: pointer to a buffer where the readed data will be stored

- `unsigned int data_count`: Size of expected amount of bytes to read
- (Optional) `unsigned int end_ptr`: endpoint number where to read the data (default 0x81)

Returns Length of data readed the size of data read (should be same as `data_count`)

4.3 Send Text Files

`void FX3USB3Connection::send_text_file` (bool *verbose*)

Description Uses send and receive buffer functions to send a text file and reads same size response It then checks the readed data is the same as the data sendd

Receives

- `bool verbose`: True if print results and false if silent mode

Print information

`int FX3USB3Connection::print_devices()`

Description Prints all USB devices BUS, VID, PID and bcd.

Returns

- number of USB devices on success
- libusb error upon error.

`int FX3USB3Connection::print_device_descriptor()`

Description Prints the connected device descriptor.

Returns LIBUSB error return status

`int FX3USB3Connection::print_config_descriptor()`

Description Prints the connected device configuration

Returns LIBUSB error return status

`libusb_device_descriptor FX3USB3Connection::get_device_descriptor()`

Returns `libusb_device_descriptor` with the information of the connected device.

`int FX3USB3Connection::soft_reset()`

Description Send a reset command to the FX3 device and then reconnects to it.

Returns

- 0 on success
- cyusb error if not

`int FX3USB3Connection::claim_interface(int interface)`

Description This program is a CLI program to claim an interface for a device which has an unclaimed interface

Receives Interface to claim

F

FX3USB3Connection (C++ *class*), 1
FX3USB3Connection::claim_interface (C++
 function), 13
FX3USB3Connection::download_fx3_firmware
 (C++ *function*), 7
FX3USB3Connection::FX3USB3Connection
 (C++ *function*), 5
FX3USB3Connection::get_device_descriptor
 (C++ *function*), 11
FX3USB3Connection::print_config_descriptor
 (C++ *function*), 11
FX3USB3Connection::print_device_descriptor
 (C++ *function*), 11
FX3USB3Connection::print_devices (C++
 function), 11
FX3USB3Connection::program_device (C++
 function), 7
FX3USB3Connection::send_buffer (C++ *func-*
 tion), 9
FX3USB3Connection::send_text_file (C++
 function), 10
FX3USB3Connection::soft_reset (C++ *func-*
 tion), 13