1. Overview

1.1 Overview

This specification describes the simplified CPLD control tool (hereinafter, this tool) functions for R-Car M3/H3/V3M/V3H Starter Kit, V3U Falcon and S4 Spider. This tool works on Linux environment.

1.2 Function

This tool supports the reading/writing matter of CPLD registers, based on that, users can control the target board in many functions: resetting, SoC boot mode configuration, LPDDR4 frequency, on-board peripheral configuration, etc…

1.3 Reference

1.3.1 Related Documents

The following table shows the document related to this module.

Table 1-1 Related document

Number	Issue	Title
-	Renesas Electronics	R-Car M3/H3/V3M/V3H Starter Kit/S4 Spider User's Manual
-	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7795SIPB0010S
-	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7796SIPB0011S
-	Renesas Electronics	R-CarV3U-SiP System Evaluation Board Falcon Hardware Manual
-	Renesas Electronics	R-CarS4-SiP System Evaluation Board Spider Hardware Manual

1.4 Notice

- This tool was only tested on M3/H3 Starter Kit and one-passed tested for V3M/V3H Starter Kit and V3U Falcon. The detail testing will be conducted when more devices are available.
- Make sure that the CPLD firmware version of the devices matches the following table:

Table 1-2 CPLD firmware version

Board name	Version date (yyyy/mm/dd)
M3/H3 Starter Kit	2019/04/08
V3M Starter Kit	2020/08/18
V3H Starter Kit	2021/06/14
V3U Falcon	2021/05/18
S4 Spider	2020/07/21

- Privilege for USB access: since this tool will access to USB devices on Linux system, users need root permission to execute it successfully, users can select either one of two methods:
  - Run the tool with sudo command.
  - Configure Ubuntu Linux to allow a particular user can run the tool without sudo command:

    1. Add the user to group dialout:

    ```
    $sudo usermod -aG dialout $USER
    ```

    2. Create file "/etc/udev/rules.d/10-cpld-control.rules" (need to run as root permission) with content as below:

    ```
    ATTRS{idProduct}=="6001", ATTRS{idVendor}=="0403", MODE="660", GROUP="dialout"
    ATTRS{idProduct}=="6010", ATTRS{idVendor}=="0403", MODE="660", GROUP="dialout"
    ATTRS{idProduct}=="6011", ATTRS{idVendor}=="0403", MODE="660", GROUP="dialout"
    ATTRS{idProduct}=="6014", ATTRS{idVendor}=="0403", MODE="660", GROUP="dialout"
    ```

    3. Reload udev rules:

    ```
    $sudo udevadm control --reload
    $sudo udevadm trigger
    ```

1.5 Terminology

The following table shows the terminology related to this module.

Table 1-3 Terminology

Terms	Explanation
CPLD	Complex Programmable Logic Device
SPI	Serial Peripheral Interface
SMI	Serial Management Interface
I^{2}C	Inter-Integrated Circuit
MPSSE	Multi-Protocol Synchronous Serial Engine
FTDI	Future Technology Devices International, is a Scottish privately held semiconductor device company, specializing in Universal Serial Bus (USB) technology

2. Operating Environment

2.1 Hardware Environment

The following table lists the hardware needed to use this module.

Table 2-1 Hardware specification

Name	Version	Manufacture
The above-mentioned R-Car boards	-	Renesas Electronics
Ubuntu PC (x86/x64) or FEAT box (ARM/ARM64) or another Linux device	-	-

Module Configuration

Figure 2-1 shows the software configuration in which this module is used.

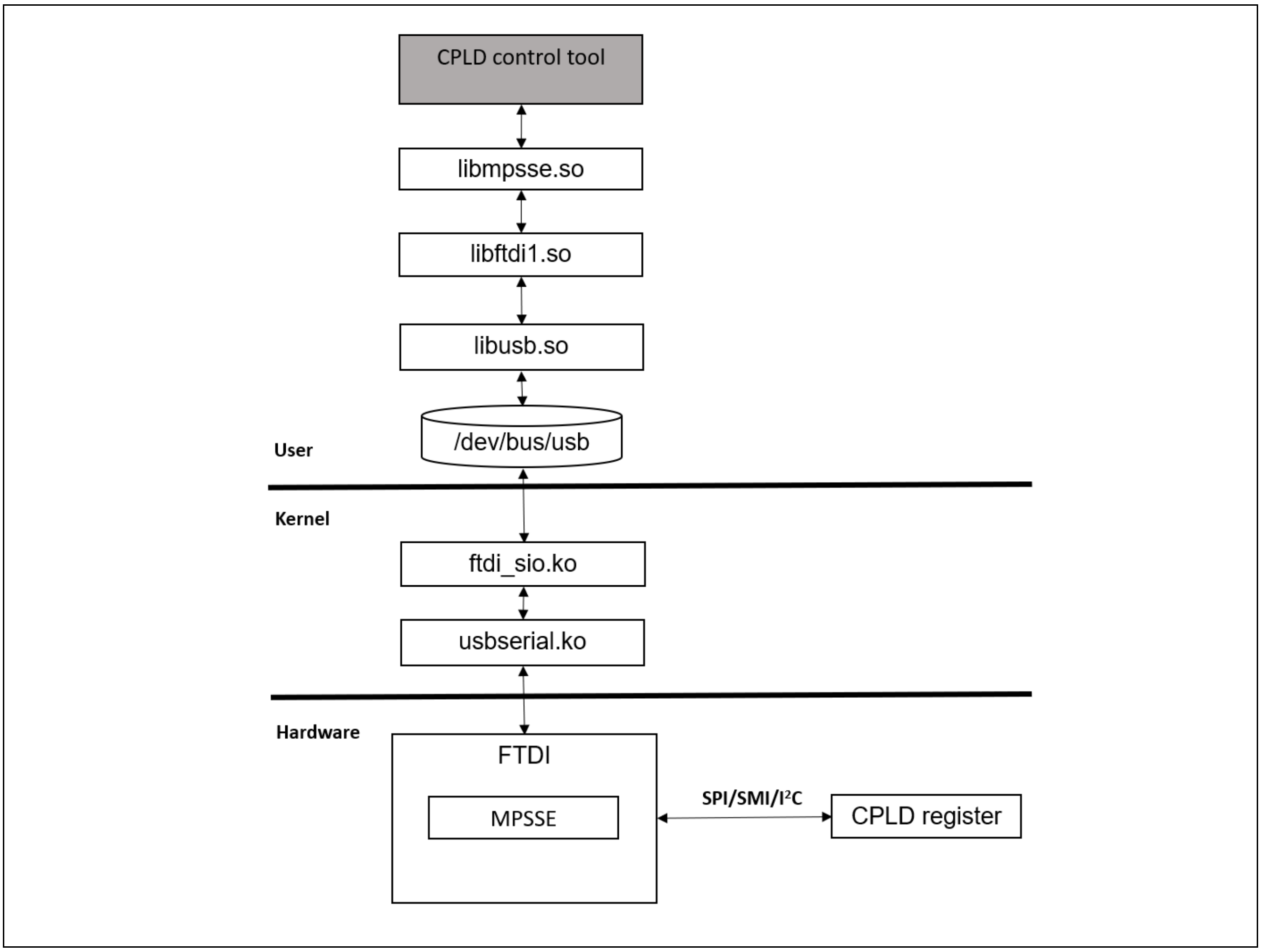

Figure 2-1 Software block diagram

(*) SPI protocol is used for H3/M3 Starter Kit, SMI protocol is used for V3M Starter Kit and I^{2}C protocol is used for V3U Falcon/V3H Starter Kit/S4 Spider

2.2 Flowchart

This section describes the basic sequences of the simplified version of CPLD control tool

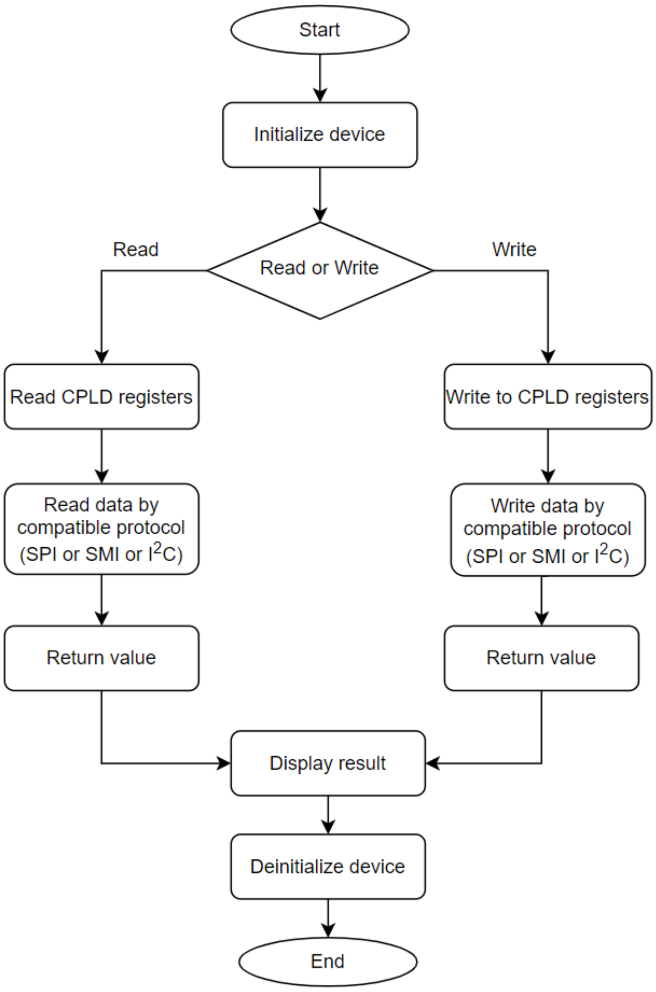

Figure 2-2 Flowchart of CPLD control tool

Notice:

- Users need to send the address of CPLD register as input parameter. The CPLD registers are described in the document mentioned in chapter 1.3.1 Related Documents.
- The communication protocol is implemented by software, depends on which board is using, the tool will select the suitable protocol automatically.

3. Command Specification

This chapter describes the command specifications of CPLD control tool. Beside the two essential commands are reading and writing CPLD register, there are also some additional commands for supporting purpose.

3.1 CPLD help command

A common help option

3.1.1 Synopsis

```
# cpld-control –h
```

3.1.2 Description

Display a simple description for all the supported commands and their syntax.

3.1.3 Command option

-h: option “Help”

3.1.4 Notice

None

3.2 CPLD list command

List all the available devices

3.2.1 Synopsis

```
# cpld-control –l
```

3.2.2 Description

This command is used to list all the available FTDI devices which are plugged in. Because a board controller can control many boards via CPLD, we need know the FTDI device serial number of each board to avoid the wrong destination mistake.

3.2.3 Command option

-l: option “List”

3.2.4 Notice

- This command will access USB devices on Linux system, users need Privilege for USB access. Please refer to chapter 1.4 Notice for the detail
- This function searches the FTDI device based on the Vendor ID and the Product ID. If there are many boards using the same FTDI devices are plugged in at the same time, they will still be listed sufficiently.

3.3 CPLD change serial command

Change the serial number or FTDI device.

3.3.1 Synopsis

```
# cpld-control –c <Board name> <Old FTDI serial number> <New FTDI serial number>
```

3.3.2 Description

This command is used to change the serial number of FTDI device on the boards. Users are recommended to change the default serial number to their expected number.

3.3.3 Command option

-c: option “Change serial number”

<Board name>: The name of the using board.

M3SK:	M3 Starter Kit
H3SK:	H3 Starter Kit
V3MSK:	V3M Starter Kit
V3HSK:	V3H Starter Kit
V3U:	V3U Falcon
S4:	S4 Spider

Example:

Change serial number of board V3U from 000001 to 000040:

```
$sudo ./cpld-control –c V3U 000001 000040
```

3.3.4 Notice

- The serial number can be alphabet, digit or special character and should not have whitespace character.
- This command will access to USB devices on Linux system, users need Privilege for USB access. Please refer to chapter 1.4 Notice for the detail.
- The serial number should be set to a unique number for each board to avoid the wrong destination mistake.
- If there are many boards having the same serial number are plugged in at the same time, the command will change only one of them.

3.4 CPLD read command

Read the specific CPLD registers or all registers.

3.4.1 Synopsis

```
# cpld-control –r <Board name> <FTDI serial> <register1> <register2> <register3>…
```

3.4.2 Description

This command is used to read the register value of CPLD. It can be used to read a specific register or all registers of CPLD. Users must specify FTDI serial number they are using and the address of the registers they want to read.

3.4.3 Command option

-r: option "Read"

<Board name>: The name of the using board.

M3SK:	M3 Starter Kit
H3SK:	H3 Starter Kit
V3MSK:	V3M Starter Kit
V3HSK:	V3H Starter Kit
V3U:	V3U Falcon
S4:	S4 Spider

<FTDI serial>: the serial number of the FTDI chip is being used. It can be acquired by the CPLD list command.

<register1> <register2> <register3>…: the register addresses. No register address means users want to read all of them.

Example:

Read all V3U (serial 000001) CPLD registers value:

```
$sudo ./cpld-control –r V3U 000001
```

Read V3U (serial 000001) CPLD Dip switch 50 register value:

```
$sudo ./cpld-control –r V3U 000001 0x0020
```

3.4.4 Notice

- The register addresses must be in hexadecimal values and must not exceed 64bits.
- This command will access to USB devices on Linux system, users need Privilege for USB access. Please refer to chapter 1.4 Notice for the detail.
- In case users want to interact with many boards, they need to make sure each board have the unique FDTI serial number. Article 3.3 CPLD change serial command can be the help.
- If users specify an invalid address, the tool will return error and stop.

3.5 CPLD write command

Write the desired values to the specific registers

3.5.1 Synopsis

```
# cpld-control –w <Board name> <FTDI serial> <register1> <value1> <register2> <value2> <register3> <value3>…
```

3.5.2 Description

This command is used to write the desired value to the specific registers of CPLD. It can be used to write to a specific register or many registers of CPLD. Users must specify FTDI serial number they are using and the address of the registers they want to write to.

3.5.3 Command option

-w: option “Write”

<Board name>: The name of the using board.

M3SK:	M3 Starter Kit
H3SK:	H3 Starter Kit
V3MSK:	V3M Starter Kit
V3HSK:	V3H Starter Kit
V3U:	V3U Falcon
S4:	S4 Spider

<FTDI serial>: the serial number of the FTDI chip is being used. It can be acquired by the CPLD list command.

<register1> <register2> <register3>…: the registers addresses.

<value1> <value2> <value3>…: the desired values.

Example:

Write the value 0x01 to V3U (serial 000001) CPLD register address 0x0024 to reset V3U board:

```
$sudo ./cpld-control –w V3U 000001 0x0024 0x01
```

3.5.4 Notice

- The register addresses and values must be in hexadecimal values and must not exceed 64bits.
- This command will access to USB devices on Linux system, users need Privilege for USB access. Please refer to chapter 1.4 Notice for the detail.
- In case users want to interact with many boards, they need to make sure each board have the unique FDTI serial number. Article 3.3 CPLD change serial command can be the help.
- If users specify an invalid address, the tool will return error and stop.
- If users specify an overflow value of the register, it will be truncated and the least significant bit will be considered as the value. For example: the register I2C address on CPLD of V3U (address 0x0022) is an 8-bit register. If users specify a 9-bit value (0x1FF) for it, then the written value is truncated (0xFF).

3.6 CPLD write non-volatile command

Write the desired values to the specific registers

3.6.1 Synopsis

```
# cpld-control –wnv <Board name> <FTDI serial> <register1> <value1> <register2> <value2> <register3> <value3>…
```

3.6.2 Description

This command is used to write the desired value to the specific registers of CPLD and the value will be kept after power cycle. It can be used to write to a specific register or many registers of CPLD. Users must specify FTDI serial number they are using and the address of the registers they want to write to.

This function is only supported by V3M/V3H Starter Kit, V3U Falcon and S4 Spider

There is only an amount of registers that can be written non-volatile and the amount is also different in each board.

3.6.3 Command option

-wnv: option “Write non-volatile”

<Board name>: The name of the using board.

V3MSK:	V3M Starter Kit
V3HSK:	V3H Starter Kit
V3U:	V3U Falcon
S4:	S4 Spider

<FTDI serial>: the serial number of the FTDI chip is being used. It can be acquired by the CPLD list command.

<register1> <register2> <register3>…: the registers addresses.

<value1> <value2> <value3>…: the desired values.

Example:

Write the value 0x0100 to V3U (serial 000001) CPLD register address 0x1000 to save the PCB version (1.0)

```
$sudo ./cpld-control –wnv V3U 000001 0x1000 0x0100
```

3.6.4 Notice

- The register addresses and values must be in hexadecimal values and must not exceed 64bits.
- This command will access to USB devices on Linux system, users need Privilege for USB access. Please refer to chapter 1.4 Notice for the detail.
- In case users want to interact with many boards, they need to make sure each board have the unique FDTI serial number. Article 3.3 CPLD change serial command can be the help.
- If users specify an invalid address, the tool will return error and stop.
- If users specify an overflow value of the register, it will be truncated, and the least significant bit will be considered as the value. For example: the register PCB version on CPLD of V3U (address 0x1000) is a 16-bit register. If users specify a 17-bit value (0x12345) for it, then the written value is truncated (0x2345)

4.Integration

4.1 Folder Structure

The CPLD control tool directory configuration is shown below.

```
[CPLD control tool]

 cpld-control/src
   |-- Makefile              : Makefile
   |-- inc/
           |-- cpld.h        : Header file of CPLD, include CPLD register addresses and device structures
           |-- spi.h         : Header file of SPI protocol.
           |-- smi.h         : Header file of SMI protocol.
           |-- i2c.h         : Header file of I$^{2}$C protocol.
   |-- src/
           |-- cpld.c        : Source file of CPLD, extract CPLD register info, process all users commands.
           |-- spi.c         : Source file of SPI protocol, process read/write data by SPI protocol
           |-- smi.c         : Source file of SMI protocol, process read/write data by SMI protocol
           |-- i2c.c         : Source file of I$^{2}$C protocol, process read/write data by I$^{2}$C protocol
           |-- main.c        : Source file of main function, navigate users commands.
```

Figure 4-1 Folder structure (R-Car M3/H3/V3M/V3H Starter Kit, V3U Falcon, S4 Spider)

4.2 Hardware Connection

The CPLD tool can be compiled and executed on any Linux device (PC, FEAT box, etc...) and control target board via USB Serial interface.

The below figure shows an example of hardware layout on V3U Falcon board. It's similar to M3/H3/V3M/V3H Starter Kit/S4 Spider.

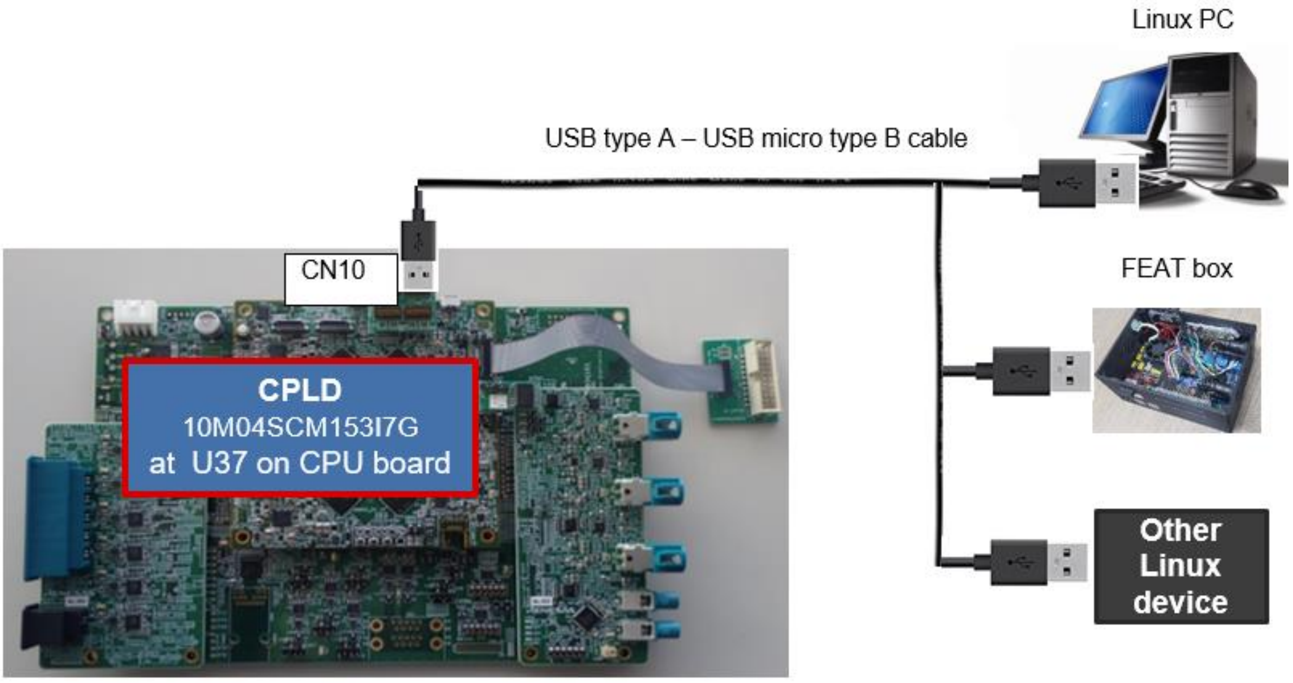

Figure 4-2 Hardware connection layout on V3U Falcon

4.3 Installation Procedure

4.3.1 Dependency software, library

- **libusb-1.0.23**

(1) The libusb-1.0.23 can be download from this repository:
https://github.com/libusb/libusb/releases/download/v1.0.23/libusb-1.0.23.tar.bz2

(*) Make sure that the libudev-dev is installed or udev usage is disabled by option --enable-udev

(2) Set the path (environment variable) to the compiler

```
$ export CC=/(Specify the path to the compiler to use)
```

Example

```
$ CC=gcc
```

(3) Run the build

```
$./configure
```

Or

```
$./configure --enable-udev=no (to disable udev usage)
$ make
$ make install
```

- **libconfuse-2.5**

(1) The libconfuse-2.5 can be download from this repository:
https://www.intra2net.com/en/developer/libftdi/download/confuse-2.5.tar.gz

(2) Set the path (environment variable) to the compiler

```
$ export CC=/(Specify the path to the compiler to use)
```

Example

```
$ CC=gcc
```

(3) Run the build

```
$./configure
$ make
$ make install
```

(4) Notice

When facing an issue with script config.guess as below

```
Checking build system type... support/config.guess: unable to guess system type
This script, last modified 2004-03-12, has failed to recognize the operating system you are using. It is
advised that you download the most up to date version of the config scripts from
```

Users can copy the guess script in their environment to this build folder by the command:

```
$ cp /usr/share/misc/config.guess support/config.guess
```

Then retry the step (3)

- **libftdi1-1.5**

(*) Make sure that all dependencies, especially libusb-1.0.23 and libconfuse-2.5 is already installed.

(1) The libftdi1 can be download from this repository:

https://www.intra2net.com/en/developer/libftdi/download/libftdi1-1.5.tar.bz2

(2) Set the path (environment variable) to the compiler

```
$ export CC=/(Specify the path to the compiler to use)
```

Example

```
$ CC=gcc
```

(3) Run the build

```
$ cd libftdi1-1.5/
$ mkdir build
$ cd build
$ cmake ..
$ make
$ make install
```

- **libmpsse-1.3**

(*) Make sure that all dependencies, especially libftdi1-1.5 is already installed.

(*) Make sure that swig (http://www.swig.org/download.html) is installed or disable python usage by option --disable-python

(1) The libmpsse-1.3 can be download from this repository:

https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/libmpsse/libmpsse-1.3.tar.gz

(2) Set the path (environment variable) to the compiler

$ export CC=/(Specify the path to the compiler to use)

Example

```
$ CC=gcc
```

(3) Run configure file

```
$ cd src/
$./configure
```

Or

$./configure --disable-python (to disable python usage)

(4) Edit Makefile and build

Add "-DLIBFTDI1=1" to Makefile of MPSSE line #32 to prevent error using old version of libftdi (libftdi-0.x)

```
$vi Makefile
```

Add (CC) $(CFLAGS) $(LDFLAGS) -DLIBFTDI1=1 -c fast.c to line #32 and save.

```
$ make
$ make install
```

NOTE: After compiling and installing all new shared libraries, command "ldconfig" should be executed to create a necessary links and cache to them

- **ftdi_sio and usbserial driver**

Most Linux distro with kernel version at least 2.4.3 will include these drivers by default.

Users can check to see if the driver is loaded:

```
$dmesg | grep -i ftdi
```

And they should see something like:

```
usb 1-11: Manufacturer: FTDI
ftdi_sio 1-11:1.0: FTDI USB Serial Device converter detected
usb 1-11: FTDI USB Serial Device converter now attached to ttyUSB0
```

If the device is not recognized be the kernel. Please troubleshoot and then try to manually load it.

For more information, you can read the summary about ftdi_sio and usbserial driver on http://ftdi-usb-sio.sourceforge.net/

4.3.2 CPLD control tool

(1) Set the path (environment variable) to the compiler

```
$ export CC=/(Specify the path to the compiler to use)
```

Example

```
$ CC=gcc
```

(2) Run the build

```
$ cd source
$ make clean
$ make
```

If the build is successful, the binary file named “cpld-control” will be generated.

5. Sample log

The display sample on V3U Falcon of this tool is shown below.

```
$sudo ./cpld-control -h
CPLD control version 1.7

The valid <Board name>: M3SK, H3SK, V3HSK, V3MSK, V3U, S4

./cpld-control -h ...................................................... Print this help.
./cpld-control -l ...................................................... List available devices.
./cpld-control -c <Board name> <Old serial number> <New serial number>... Change FTDI serial number
./cpld-control -r <Board name> <FTDI iSerial> .......................... Print all CPLD registers.
./cpld-control -r <Board name> <FTDI iSerial> <reg>* ................... Print 1 CPLD register.
                                 *One or more <reg> can be specified.
./cpld-control -w <Board name> <FTDI iSerial> [<reg> <val>]* ........... Write CPLD register(s).
                                 *One or more [<reg> <val>] pairs can be specified.
./cpld-control -wnv <Board name> <FTDI iSerial> [<reg> <val>]* ......... Write non-volatile CPLD register(s).
                                 *One or more [<reg> <val>] pairs can be specified.

$ sudo ./cpld-control -l
FT232R: 0861
FT2232: 40

$ sudo ./cpld-control -c V3U 40 000040
Using device V3U with iSerial: 40

Serial number has been changed, please run cpld-control -l to re-check!

$ sudo ./cpld-control -l
FT232R: 0861
FT2232: 000040

$ sudo ./cpld-control -r V3U 000040
Using device V3U with iSerial: 000040

PRODUCT         0x0000: 0xB8A779A0
VERSION         0x0004: 0x20200728
MODE_SET        0x0008: 0x00000000006100A8
MODE_NEXT       0x0010: 0x00000000006100A8
MODE_LAST       0x0018: 0x00000000006100A8
DIPSW50         0x0020: 0xD4
I2C_ADDR        0x0022: 0xE0
```

```
RESET           0x0024: 0x40
POWER_CFG       0x0025: 0x01
PERI_CFG        0x0030: 0x08
UART_CFG        0x0036: 0x00
UART_STATUS     0x0037: 0x00
CNT_POWER       0x0080: 0x00000000
CNT_RESET       0x0084: 0x00000000
PCB_VERSION     0x1000: 0x0000
SOC_VERSION     0x1002: 0x0000
PCB_SN          0x1004: 0xFFFFFFFF
MAC             0x1008: 0xFFFFFFFFFFFF

$ sudo ./cpld-control -w V3U 000040 0x0008 0x00010000006100BE 0x0024 0x01
Using device V3U with iSerial: 000040

Writing register 0x0008 with value 0x00010000006100BE
Writing register 0x0024 with value 0x01
PRODUCT         0x0000: 0xB8A779A0
VERSION         0x0004: 0x20200728
MODE_SET        0x0008: 0x00010000006100BE
MODE_NEXT       0x0010: 0x00010000006100BE
MODE_LAST       0x0018: 0x00010000006100BE
DIPSW50         0x0020: 0xD4
I2C_ADDR        0x0022: 0xE0
RESET           0x0024: 0x40
POWER_CFG       0x0025: 0x01
PERI_CFG        0x0030: 0x08
UART_CFG        0x0036: 0x00
UART_STATUS     0x0037: 0x00
CNT_POWER       0x0080: 0x00000000
CNT_RESET       0x0084: 0x00000000
PCB_VERSION     0x1000: 0x0000
SOC_VERSION     0x1002: 0x0000
PCB_SN          0x1004: 0xFFFFFFFF
MAC             0x1008: 0xFFFFFFFFFFFF

$ sudo ./cpld-control -wnv V3U 000040 0x1002 0x0100
Using device V3U with iSerial: 000040

Writing register 0x1002 with value 0x0100
PRODUCT         0x0000: 0xB8A779A0
VERSION         0x0004: 0x20200728
```

```
MODE_SET        0x0008：0x00010000006100BE
MODE_NEXT       0x0010：0x00010000006100BE
MODE_LAST       0x0018：0x00000000006100A8
DIPSW50         0x0020：0xD4
I2C_ADDR        0x0022：0xE0
RESET           0x0024：0x40
POWER_CFG       0x0025：0x01
PERI_CFG        0x0030：0x08
UART_CFG        0x0036：0x00
UART_STATUS     0x0037：0x00
CNT_POWER       0x0080：0x00000000
CNT_RESET       0x0084：0x00000000
PCB_VERSION     0x1000：0x0000
SOC_VERSION     0x1002：0x0100
PCB_SN          0x1004：0xFFFFFFFF
MAC             0x1008：0xFFFFFFFFFFFF
```