Celoxica

# Platform Developer's Kit

## FTU2 User manual

Authors: RG

Document number: 1

Customer Support at http://www.celoxica.com/support/

| Celoxica in Europe | Celoxica in Japan | Celoxica in the Americas |
|---|---|---|
| T: +44 (0) 1235 863 656 | T: +81 (0) 45 331 0218 | T: +1 800 570 7004 |
| E: sales.emea@celoxica.com | E: sales.japan@celoxica.com | E: sales.america@celoxica.com |

# Contents

# Conventions

A number of conventions are used in this document. These conventions are detailed below.

 Warning Message. These messages warn you that actions may damage your hardware.

 Handy Note. These messages draw your attention to crucial pieces of information.

Hexadecimal numbers will appear throughout this document. The convention used is that of prefixing the number with '0x' in common with standard C syntax.

Sections of code or commands that you must type are given in typewriter font like this:

```
void main();
```

Information about a type of object you must specify is given in italics like this:

```
copy SourceFileName DestinationFileName
```

Optional elements are enclosed in square brackets like this:

```
struct [type_Name]
```

Curly brackets around an element show that it is optional but it may be repeated any number of times.

```
string ::= "{character}"
```

# Assumptions & Omissions

This manual assumes that you:

- have used Handel-C or have the Handel-C Language Reference Manual
- are familiar with common programming terms (e.g. functions)
- are familiar with MS Windows

This manual does not include:

- instruction in VHDL or Verilog
- instruction in the use of place and route tools
- tutorial example programs. These are provided in the Handel-C User Manual

# 1 Celoxica FTU2 program

The Celoxica File Transfer Utility (FTU2) allows you to configure the FPGA on a RC100 or RC200 board via a parallel port cable.

The FTU2 also allows you to interact with Flash memory/SmartMedia on the RC100 and RC200. You can download a BIT file, write data from a file or read data from the memory.

The FTU2 can be run from a graphical interface, or via the command line. If you want to interact with the SmartMedia card on the RC200, or download a BIT file to Flash memory on the RC100 you need to use the command-line version of the program.

## 1.1 Using the FTU2 from the GUI

The graphical version of the FTU2 allows you to:

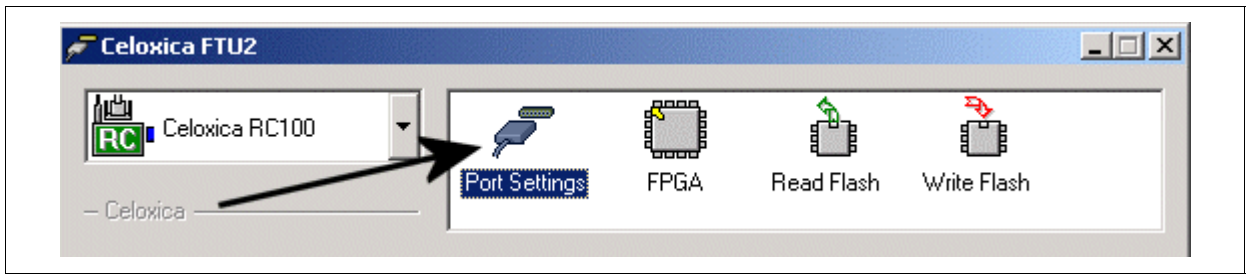
- configure the FPGA, or read from or write to Flash memory on the RC100.
- configure the FPGA on the RC200.

To start the program: select Programs>Celoxica>Platform Developer's Kit>FTU2 program from the Start menu.

Before you write to or read from your board:

1. Check the board is connected to your PC and switched on.
2. Select RC100 or RC200 from the drop-down menu in the top left-hand corner.
3. Test the port settings.

### *1.1.1 FTU2 port settings*




You need to configure the port settings for the FTU2 program the first time you use it with a particular PC or board. After this, you should not need to change the settings.

### Port Address

The default parallel port for the FTU2 is LPT1. To change this:

- Select the LPT2 button

OR

- Select the Other - address: button and write an address in hexadecimal in the box next to it

If you specify a random address in the box next to the Other - address button you could cause your PC to crash.

### Port Mode

By default all 3 parallel port tests are selected. This will lead to faster communication between your PC and the FPGA or Flash memory on the board. However some of the tests may cause problems on certain PCs. If there is a problem when you test the port settings, try unticking some of these tests.

### Testing port settings

Once you have set the address and port mode (or have left the default settings):

1. Check that the board is connected to your PC and switched on.
2. Press the Test Settings button.

## *1.1.2 Configuring the FPGA*

To use the graphical version of the FTU2 program to download BIT files onto the FPGA:

1. Press the FPGA icon at the top of the screen. This allows you to access the relevant settings.
2. Use the Browse button to select the file you want to download. Recently selected files will be shown in the box on the screen.
3. Configure the FPGA using one of these methods:
   - Double-click a file from the list on the screen.
   - Select the file from the list and press the Configure button.
   - Right-click a file and select Configure FPGA with this file.

If you want to edit the list of recent files you can Browse to select files, or press Remove to delete files.

If the .bit file you select is not configured for the FPGA on your board, there will be a red exclamation mark (!) next to the file name. You may damage your FPGA if you use the wrong kind of .bit file.

### *1.1.3 Writing files to Flash memory*

To use the graphical version of the FTU2 program to write a file to Flash memory on the RC100 board:

1. Press the Write Flash icon at the top of the screen. This allows you to access the relevant settings.
2. Press the Query Device button. This will detect which Flash device is installed on your board. You do not need to do this if the device is already displayed in the Flash Device box.
3. Press the Add button to browse for a file. An Enter Start Location box will appear.
4. Enter the Flash location you want to start writing the material to. You can specify this in bytes, or as a vector. The FTU2 program will calculate and display the End Location.
5. Press the Configure button.
   Progress will be displayed in a dialog. The dialog disappears once the download is complete and "OK" will be displayed in the Status field in the Flash Writing Schedule box if the download was successful.

You can write files of any name. By default, files with a .bit extension are treated as FPGA configuration files and others are treated as binary files. You can change this by right-clicking on a file and toggling the Write as an FPGA file setting.

To change the Start Location of a file, right-click it in the file list and choose Set Start Location.

If you want to write to SmartMedia on the RC200, use the command-line version of the FTU2.

#### Using lists of files

You can create a list of files and write them all in a single session. Use the Add and Remove buttons to create your list and press the Configure button once the list is complete. You can save your file list, together with the Start/End location settings by pressing the Save List button. Press the Open List button to access previous lists.

By default, saved lists have an .ini extension, but you can specify any name.

### *1.1.4 Reading data from Flash memory*

To use the graphical version of the FTU2 program to write material from Flash memory on the RC100 board to a file on your PC:

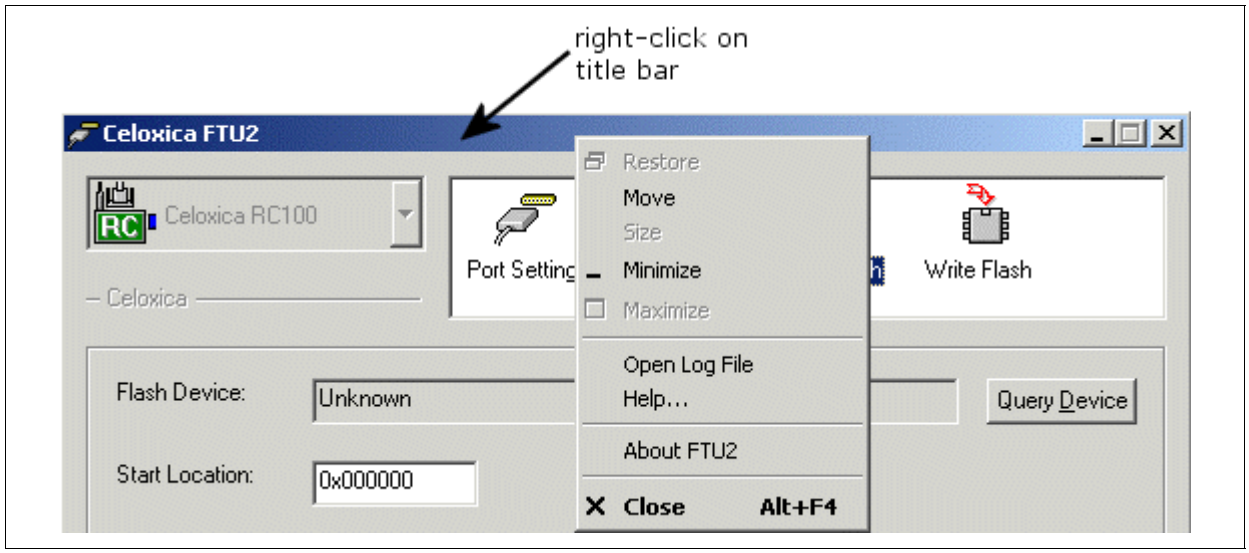
1. Press the Read Flash icon at the top of the screen. This allows you to access the relevant settings.
2. Press the Query Device button. This will detect which Flash device is installed on your board. You do not need to do this if the device is already displayed in the Flash device box.
3. Enter the Start Location of the material you want to read from Flash. You can use hexadecimal or decimal numbers. If you use decimal numbers they will be converted to hexadecimal.
4. Enter the number of bytes you want to read from this location in the File Size box.
5. Choose an Output File. You can browse to select an existing file, or write the path and name for a new file. If you choose an existing file, the content will be overwritten.
6. Press the Read button.

The name of the output file can have any extension. The default extension is `.bin`.

If you want to read from SmartMedia on the RC200, use the command-line version of the FTU2.

### 1.1.5 FTU2 log files

The FTU2 program creates a log of all the commands executed within a session. To view this, right-click on the title bar of the program and select Open Log File.




The log will open in your default text editor. The log is erased when you close the FTU2 program.

## 1.2 Using the FTU2 from the command line

In addition to the graphical version of the FTU2, you can run the FTU2 from the command line.

The command-line version of the FTU2 allows you to perform the following operations on the RC100 or RC200:

- configure the FPGA.
- read from Flash memory/SmartMedia.
- write to Flash memory/SmartMedia.
- download a BIT file to Flash memory/SmartMedia.

FTU2 commands are of the form:

`RC`***`X`***`00CmdFTU <`***`action`***`> [<`***`switch`*** `> [`***`parameter`***`] ...] [<`***`filename`***`>]`

Some of the actions have required and optional switches, some of which take a further parameter. If a file name is used, it is always the last parameter.

### *1.2.1 RC100 command-line FTU2*

FTU2 commands for the RC100 are of the form:

`RC100CmdFTU <action> [<switch> [parameter] ...] [<filename>]`

Some of the actions have required and optional switches, some of which take a further parameter. If a file name is used, it is always the last parameter.

#### Actions

The following actions may be performed for the RC100.

| Command | Action | Description |
|---|---|---|
| `-c` | Configure FPGA | Configures the RC100 with the specified BIT file. You must supply the name and path of the file. For example, if you were running the command line in the same directory as your BIT file:<br>`RC100CmdFTU <-c> <MyFile.bit>` |
| `-x` | Clear FPGA | Erases the RC100 FPGA. Requires no other parameters. |
| `-r` | Read data from Flash | Reads data from Flash memory into a file. You must specify the name and path of a file to store data in, a start location in the Flash memory and the number of bytes to read. For example:<br>`RC100CmdFTU <-r> <-l 0x10000> <-n 0xEA60> <FlashData.txt>` |
| `-d` | Write data to Flash | Writes a data file into the Flash memory. You must specify the name and path of the file to read, and a start location in Flash memory. If you want to write a BIT file, use the `-b` command. |
| `-b` | Write `BIT` file to Flash | Writes an FPGA configuration `BIT` file into the Flash memory. You must specify the name and path of the file to read, and a start location in Flash memory. |

#### Switches

The following switches can be used with the RC100.

| Command | Meaning | Description |
|---|---|---|
| `-l` ***FlashAddress*** | Start location | Required for `–r`, `-d` and `–b` actions. Has no effect on other actions. ***FlashAddress*** can be specified in decimal or hexadecimal notation and specifies the location in Flash at which a read or write operation should start. For example: `-l 0x10000` or `–l 65536` |
| `-n` ***Bytes*** | File size | Required for the `–r` action. Has no effect on other actions. ***Bytes*** specifies the number of bytes to read from the Flash memory, in decimal or hexadecimal notation. For example: `-n 0xEA60` or `–n 60000` |
| `-q` ***PortAddress*** | Parallel port base location | Can be used with any action. Allows you to specify a parallel port address (in hexadecimal) other than the default of 0x378. Ensure that the specified address is correct, or your system may become unstable. |
| `-t` ***Mode*** | Use EPP or ECP mode | Can be used with any action. ***Mode*** can be epp, ecp or ecp/epp. Causes the FTU2 to check for ECP and/or EPP capabilities in the parallel port, and use them if possible. For example, to test and use EPP only, use `–t epp`. To test for ECP and EPP, and use them, use `-t ecp/epp`<br><br>If this switch is not used, the default behaviour is to test for ECP and EPP , but then use software emulation. Using this switch can result in higher performance. |

| | | |
|---|---|---|
| -e | Force software EPP emulation | Can be used with any action. Forces the FTU2 to ignore the results of ECP or EPP tests on the parallel port, and use software emulation instead.<br><br>The default behaviour is to use software emulation, but this switch may be useful if you wish to test ECP or ECP using the -t switch, but still use emulation for the actual communication. |

### *1.2.2 RC200 command-line FTU2*

FTU2 commands for the RC200 are of the form:

RC200CmdFTU <***action***> [<***switch***> [***parameter***] ...] [<***filename***>]

Some of the actions have required and optional switches, some of which take a further parameter. If a file name is used, it is always the last parameter.

If you are targeting SmartMedia, you are recommended to use logical addressing, rather than physical addressing. Logical addressing recognises (and avoids) reserved and bad blocks in the SmartMedia, such that visible block addresses are always good blocks. To use logical addressing, you need to format the card to a Celoxica specification, by using the –f action described below.

When using a SmartMedia card with the RC200 it is possible to write data to it such that it will no longer work in a digital camera or other similar device. This is more likely to occur where physical addressing has been used to access the card. To return a SmartMedia card to a state where a digital camera can format it, use the –s action, as described below.

#### Actions

The following actions may be performed for the RC200. Required switches are indicated where appropriate.

| Command | Action | Description |
|---|---|---|
| -c | Configure FPGA | Configures the RC200 with the specified BIT file. You must supply the name and path of the file. For example, if you were running the command line in the same directory as your BIT file:<br>`RC200CmdFTU <-c> <MyFile.bit>` |
| -x | Clear FPGA | Erases the RC200 FPGA. Requires no other parameters. |
| -r | Read data from SmartMedia | Reads data from SmartMedia card into a file. You must specify the name and path of a file to store data in, a start location in the SmartMedia card and the number of bytes to read. For example:<br>`RC200CmdFTU <-r> <-l 0x32> <-n 0xEA60> <FlashData.txt>` |
| -d | Write data to SmartMedia | Writes a data file to the SmartMedia card. You must specify the name and path of the file to read, and a start location in SmartMedia card. If you want to write a BIT file, use the -b command. |
| -b | Write BIT file to SmartMedia | Writes an FPGA configuration BIT file to the SmartMedia card. You must specify the name and path of the file to read, and a start location in SmartMedia card. |
| -s | Test and restore physical format of SmartMedia | Returns a SmartMedia card to a state where most digital cameras can format it for normal use. This is achieved by testing every block on the SmartMedia card and marking any that are found bad, erasing all good blocks (so they are ready to be written to), and writing a valid CIS (Card Information Structure) and MBS (Master Boot Sector) to the card. Performing this action may take some time, as it involves writing and erasing every block in the SmartMedia card several times. |

| | | |
|---|---|---|
| -f | Perform logical format of SmartMedia | Performs a logical format of a SmartMedia card on the RC200 to the Celoxica specification. This is required before the card can be accessed from the FTU2 or the FPGA using logical addressing. It is recommended that a logical format is performed on all SmartMedia cards before further use with the FTU2 or RC200. |

### Switches

The following switches can be used with the RC200.

| Command | Meaning | Description |
|---|---|---|
| -p ***BlockAddress*** | Physical address location | Required for –r, -d and –b actions unless the –l switch is used. Has no effect on other actions. BlockAddress specifies the SmartMedia block address to start a read or write operation at, in decimal or hexadecimal notation. For example: -p 0x32 or –p 50<br>Physical addressing does not take account of bad or reserved blocks, so there is a risk of overwriting data and reserved blocks. Logical addressing avoids these risks. When using physical addressing, bad blocks will be read and written, resulting in possible data loss. To stop this behaviour, use the –k switch. |
| -k | Skip bad blocks when using physical addressing | Can only be used with the –p switch for physical addressing. Causes the FTU2 to ignore bad blocks when reading from or writing to the SmartMedia card. Note that this can cause the data written to take up more blocks than expected, carrying the risk of overwriting subsequent data. |
| -l ***LogicalAddress*** | Logical address location | Required for –r, -d and –b actions unless the -p switch is used. Has no effect on other actions. LogicalAddress specifies the location in SmartMedia at which a read or write operation should start, in decimal or hexadecimal notation. For example: -l 0x32 or –l 50<br>You must perform a logical format (-f action) before using logical addressing. |
| -n ***Bytes*** | File size | Required for the –r action. Has no effect on other actions. Bytes specifies the number of bytes to read from SmartMedia, in decimal or hexadecimal notation. For example: -n 0xEA60 or –n 60000<br>The FTU2 will tell you how many blocks will be read, although the file saved will contain the number of bytes requested. |

| | | |
|---|---|---|
| -q ***PortAddress*** | Parallel port base location | Can be used with any action. Allows you to specify a parallel port address other than the default of 0x378. Ensure that the specified address is correct, or your system may become unstable. |
| -t ***Mode*** | Use EPP or ECP mode | Can be used with any action. Mode can be epp, ecp or ecp/epp. Causes the FTU2 to check for ECP and/or EPP capabilities in the parallel port, and use them if possible. For example, to test and use EPP only, use –t epp. To test for ECP and EPP, and use them, use -t ecp/epp<br>If this switch is not used, the default behaviour is to test for ECP and EPP, but then use software emulation. Using this switch can result in higher performance. |
| -e | Force software EPP emulation | Can be used with any action. Forces the FTU2 to ignore the results of ECP or EPP tests on the parallel port, and use software emulation instead.<br>The default behaviour is to use software emulation, but this switch may be useful if you wish to test ECP or ECP using the -t switch, but still use emulation for the actual communication. |

# 1.3 Working with SmartMedia

You must use FTU2 from the command-line (not the GUI) to access SmartMedia devices on RC200 and RC203 boards.

The FTU2 utility supports both physical addressing and Celoxica-formatted logical addressing. Logical addressing accommodates bad blocks on otherwise useable media.

## *1.3.1 Examples*

**Write bitfile "mybitfile.bit" to SmartMedia at logical block 4**

rc200cmdftu -b -l 0x04 mybitfile.bit

**Write bitfile "mybitfile.bit" to SmartMedia at physical block 2**

rc200cmdftu -b -p 0x02 mybitfile.bit

**Write binary file "mydatafile" to SmartMedia at logical block 2**

rc200cmdftu -d -l 2 mydatafile

**Read 0x1000 bytes from SmartMedia starting at logical block 4 to "mediacopy.bin"**

rc200cmdftu -r -l 0x04 -n 0x1000 mediacopy.bin

For further information, see also: ***RC200 command-line FTU2*** (see page 11)

# 2 Index