



D50x Development Tool Chain

Documentation

This document describes the tools used for D50x gateway product family firmware development.

Document history:

Rev.	Creator Date	Creator Name	Released Date	Released Name	Comments	Ident-Number
1	1 Nov. 2011	Hofer	14 Nov 2011	Hofer	See chapter "Document History"	d-005468-010937
2	5 Nov 2012	Hofer	7 Nov 2012	Rapka		d-005468-013114

duagon AG
Riedstrasse 12
CH-8953 Dietikon
Phone +41 44 743 73 00
Fax +41 44 743 73 15

www.duagon.com



Table of Contents

- 1. Introduction.....3**
- 2. Configuration Overview.....4**
- 3. Firmware Tool Chain.....5**
 - 3.1 Software Development Tools..... 5
 - 3.2 Development Kit.....6
 - 3.2.1 USB-Blaster..... 7
 - 3.2.2 Serial Debug Cable..... 7
 - 3.3 Compilation..... 8
 - 3.4 Programming..... 9
 - 3.5 Dump Flash Image..... 11
 - 3.5.1 Dump Full Flash Image..... 11
 - 3.5.2 Dump Flash Image without file system "/etc"..... 11
- 4. The Command Line Interface.....12**
 - 4.1 Install and Setup uCon..... 12
 - 4.2 Debugging with CLI..... 14
 - 4.3 Up / downloading files..... 15
- 5. Links to other duagon Documents.....16**
- 6. Document History.....17**

1. Introduction

The functionality of the D50x gateway product family is based on an FPGA configuration and a firmware image.

The FPGA configuration and a demo firmware are supplied by duagon AG.

The customer can use the demo firmware as a basis for his own D50x firmware.

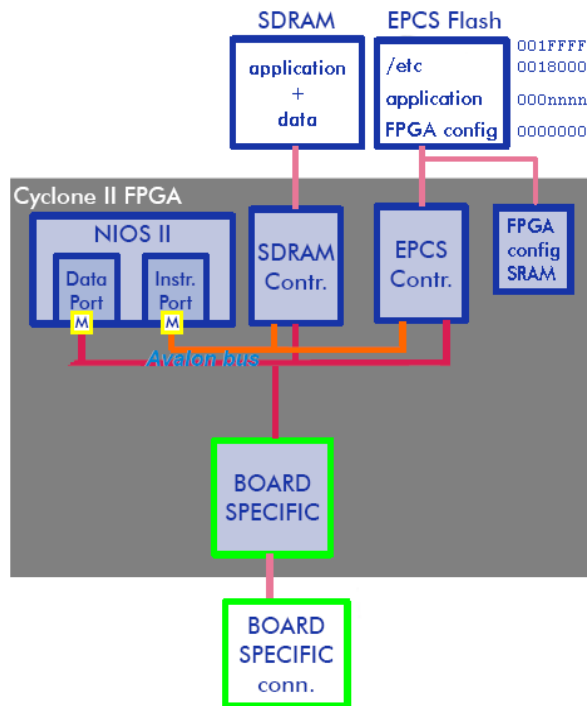
To develop the firmware, a series of tools are required. This document describes the tools used for D50x firmware development. Basically all versions of the D50x gateway product family are similar.

The D50x gateway product family provides a hardware platform to translate between Ethernet and various serial interfaces or field buses:

- D501, data sheet d-002540-nnnnnn
- D502, data sheet d-002552-nnnnnn
- D503, data sheet d-004090-nnnnnn
- D504, data sheet d-004091-nnnnnn

2. Configuration Overview

The D50x gateway product family boards have an EPCS flash memory device, a SDRAM device, and application specific peripherals:



The FPGA configuration and the firmware are stored in the flash. The flash also contains a Flash File System.

After powering, the FPGA configuration is loaded from the flash into the FPGA internal configuration memory by the FPGA's configuration cycle (known as Active Serial Configuration).

After the FPGA is configured, a bootstrap program will read the application from the flash, load it into the SDRAM and start its execution.

3. Firmware Tool Chain

duagon AG supplies the FPGA configuration and a basic firmware.

All D50x product family boards are shipped pre-programmed with a demo application. The same demo application is also contained in the development library. The product specific development library is required for application development.

3.1 Software Development Tools

To compile and program the D50x product family boards with a new firmware, the tools used are supplied by Altera.

Primarily the Nios II Embedded Design Suite (EDS) is required. The Nios II EDS contains the GNU C compiler and linker, which are used to create the firmware.

Before installation of the Nios II EDS, first the Quartus II Web Edition Software must be installed. duagon recommends a specific version of the Altera design software:

Design Software	Version
Quartus II Web Edition	11.1
NIOS II EDS	11.1

Remarks:

- All Design Software can be downloaded from the Altera website "<https://www.altera.com/>", e.g. go to the "Download Center" or search for "Quartus II Web Edition Software v11.1".
The recommended Altera Design Software packages are free of charge.
- Install the software packages in the order as listed in the table above.
- duagon does not provide any support for Altera's Design Software.

3.2 Development Kit

The duagon provided Development Kit contains the hardware components required for application development. It can be ordered at sales@duagon.com (order code d-005672-nnnnnn). Figure 1 shows the application development related parts included in the development kit.

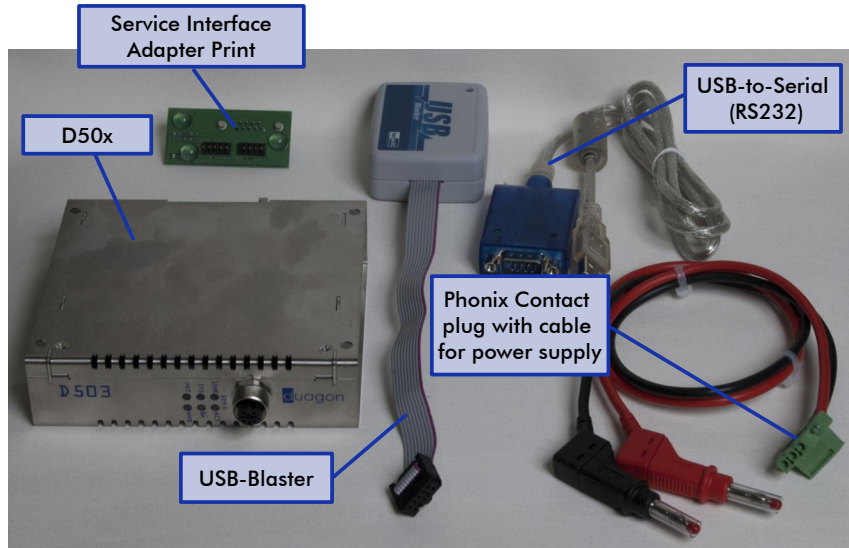


Figure 1: Development Kit Components

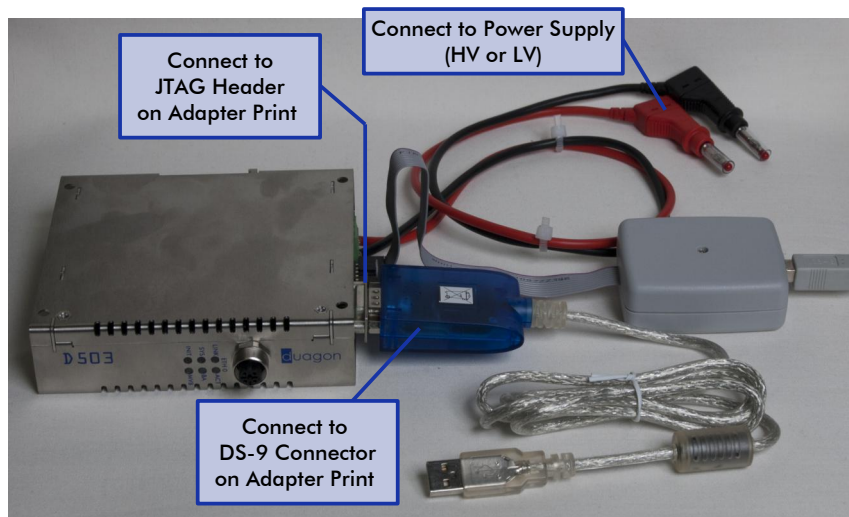


Figure 2: Hardware Setup (i.e. connection of cables to D50x)

3.2.1 USB-Blaster

The USB-Blaster connects via an USB cable to the PC. On the other end, connect the header to the JTAG connector on the adapter print. Notice that Pin1 is marked on the adapter print and on the flexible ribbon of the USB-Blaster.

3.2.2 Serial Debug Cable

Connect the SER connector on the adapter print directly via USB-to-RS232 converter cable to the PC. Notice, no null-modem cable is required.

3.3 Compilation

The easiest way to compile is to open a Nios II Command Shell. The Nios II Command Shell can be opened by executing following ".bat" file:

```
"C:\altera\11.1\nios2eds\Nios II Command Shell.bat"
```

The following commands must be performed in the Nios II Command Shell.

Change the working directory to the `Makefile` location:

```
cd /cygdrive/c/duagon/D50x/devel/app /cygdrive/c/duagon/D50x/devel/app
```

D50x, where x is the product number, assuming the D50x sources were copied to `C:\duagon\D50x` and enter the `make` command:

```
make
```

Notes:

1. The Nios II Command Shell under Windows is based on the Cygwin, a Unix-like environment for Windows. Therefore, it uses Unix-like commands.
2. It is usually good to use `make clean` before `make` when many changes are made, specially on ".h" files.

`make` will create the `app.elf` file; the name `app` is defined in the `Makefile` by the `APP_NAME` variable.

3.4 Programming

Once the `app.elf` file is generated, it can be converted to a flash file and programmed in the flash memory.

Programming requires power applied the board and an USB-Blaster correctly installed and connected to the JTAG connector (see previous section "Development Kit").

After programmed to the flash, the configuration is not lost after power off.

The flash can be programmed by typing in the Nios II command Shell:

```
make program
```

A successful programming will show something like:

```
duagon:~/devel/app> make program
-- Generating flash file
elf2flash --epcs --input=app.elf --output=app.flash
--after=../../support/pld/pld.flash
-- Load programmer PLD
quartus_pgm -c USB-Blaster -m JTAG -o "P;../../support/pld/programmer-
.sof@1"
Warning (210120): Cyclone IV E information is incomplete. The ISP clamp
functionality will be disabled.
Info: *****
Info: Running Quartus II 32-bit Programmer
Info: Version 11.1 Build 259 01/25/2012 Service Pack 2 SJ Web Edition
Info: Copyright (C) 1991-2011 Altera Corporation. All rights reserved.
Info: Your use of Altera Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Altera Program License
Info: Subscription Agreement, Altera MegaCore Function License
Info: Agreement, or other applicable license agreement, including,
Info: without limitation, that your use is for the sole purpose of
Info: programming logic devices manufactured by Altera and sold by
Info: Altera or its authorized distributors. Please refer to the
Info: applicable agreement for further details.
Info: Processing started: Thu Nov 8 10:30:02 2012
Info: Command: quartus_pgm -c USB-Blaster -m JTAG -o
P;../../support/pld/programmer.sof@1
Info (213045): Using programming cable "USB-Blaster [2-2]"
Info (213011): Using programming file ../../support/pld/programmer.sof
with checksum 0x002353B7 for device EP4CE15F17@1
Info (209060): Started Programmer operation at Thu Nov 8 10:30:04 2012
Info (209016): Configuring device index 1
Info (209017): Device 1 contains JTAG ID code 0x020F20DD
Info (209007): Configuration succeeded -- 1 device(s) configured
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Thu Nov 8 10:30:06 2012
Info: Quartus II 32-bit Programmer was successful. 0 errors, 1 warning
Info: Peak virtual memory: 104 megabytes
Info: Processing ended: Thu Nov 8 10:30:06 2012
Info: Elapsed time: 00:00:04
Info: Total CPU time (on all processors): 00:00:01
-- Programming at EPCS address 0x0F000000
nios2-flash-programmer --epcs --base=0x0F000000 ../../support/pld/pld-
flash app.flash
Using cable "USB-Blaster [2-2]", device 1, instance 0x00
Resetting and pausing target processor: OK
Checksummed/read 59kB in 4.2s
Erased 1344kB in 14.1s (95.3kB/s)
Programmed 1286KB +58KB in 147.5s (9.1KB/s)
Did not attempt to verify device contents
Leaving target processor paused
duagon:~/devel/app>
```

If there is any programming problem, check the cables and proper powering of the device.

Alternatively, the steps executed automatically by "make program" can also be performed manually:

1. Convert the ELF file to a flash file

For this step, the duagon supplied `pld.flash` file is also needed in the app directory (see product specific documentation on demo application):

```
elf2flash --epcs --input=app.elf --output=app.flash --after=pld.flash
```

2. Load flash programmer FPGA configuration

Following command directly loads a flash programmer FPGA configuration to the FPGA. For this step, the duagon supplied `programmer.sof` file is also needed in the app directory (see product specific documentation on demo application):

```
quartus_pgm -c USB-Blaster -m JTAG -o "P;programmer.sof@1"
```

3. Program the flash files

```
nios2-flash-programmer --epcs --base=0x0F000000 pld.flash app.flash
```

4. Turn power off and then on again for the new firmware to start

3.5 Dump Flash Image

3.5.1 Dump Full Flash Image

The flash image file can be created by dumping the flash content.

The flash image file contains following data:

1. FPGA configuration
2. embedded application
3. /etc file system (typically contains the configuration file `device.cfg`)

Following command creates the flash image file `image.flash`:

```
make dump
```

The file `image.flash` can either be downloaded via webserver (e.g. for remote update) or via JTAG.

The command to download the `image.flash` file via JTAG is similar to to the program command from the previous section:

```
quartus_pgm -c USB-Blaster -m JTAG -o "P;programmer.sof@1"
```

```
nios2-flash-programmer --epcs -base=0x0F000000 --program image.flash
```

3.5.2 Dump Flash Image without file system "/etc"

Sometimes it make sense to exclude the `/etc` file system from the flash image.

Remind: The `/etc` file system typically contains the configuration file `device.cfg` (e.g. configuration of Ethernet interface). In addition the `/etc` file system may contain customer specific files.

Following command creates the flash image file `image_nocfg.flash`:

```
make dump_nocfg
```

4. The Command Line Interface

The Command Line Interface (CLI) provides useful informations for debugging and for simple tests.

For the connection of the serial debug console, see section "Development Kit".

4.1 Install and Setup uCon

Install a terminal program, if it is not already available on your PC. For the purposes of this document, uCon is the used terminal program.

- uCon (http://www.umonfw.com/releases/ucon_install.exe)

Start uCon and select the proper COM port and set the baud rate to 115.2K as shown below:

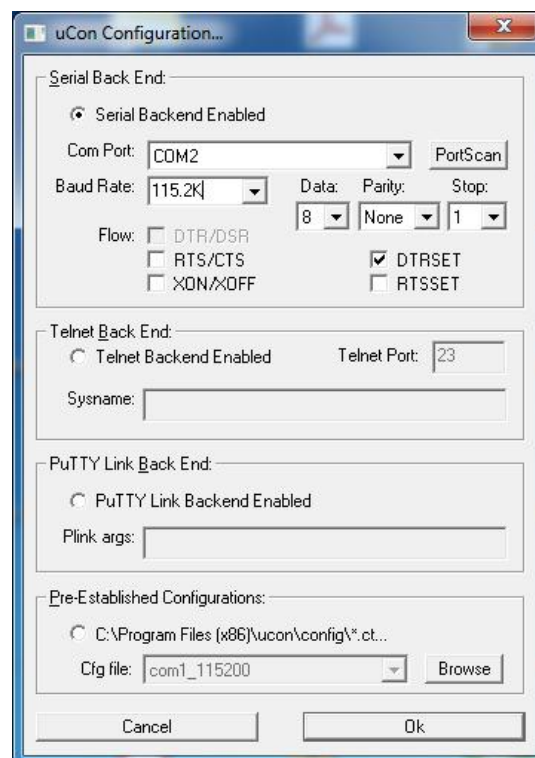


Figure 3: uCon Configuration

After setting up the serial console and the cables, turn the power on. A screen similar to the following should appear:

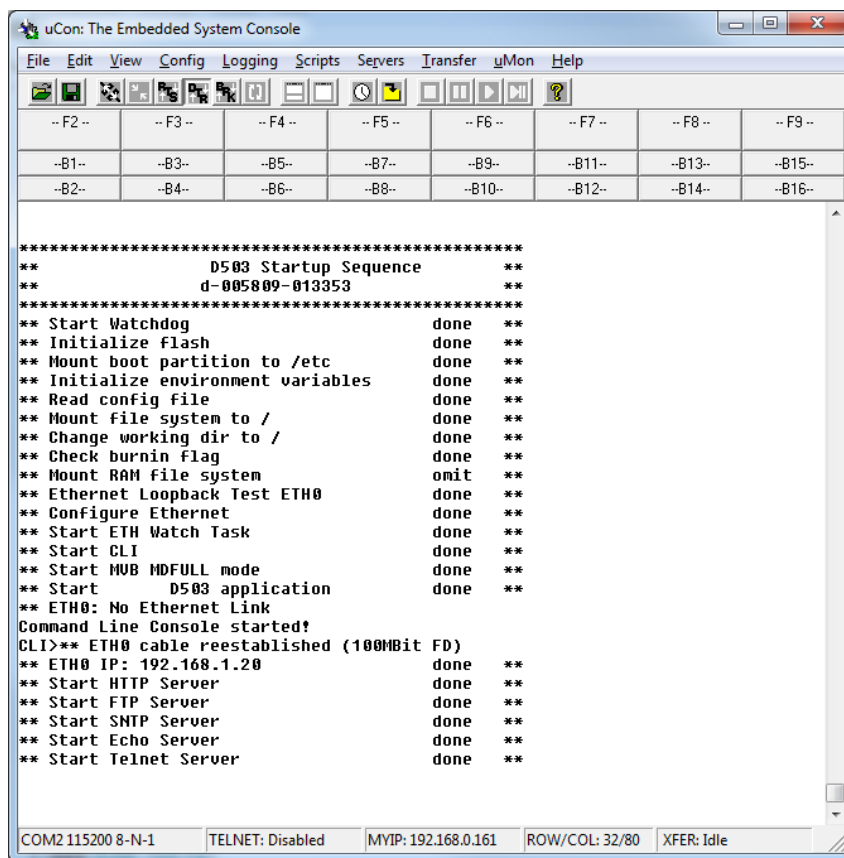


Figure 4: Device Startup

Typing `help` (followed by 'Enter') will show all available commands.

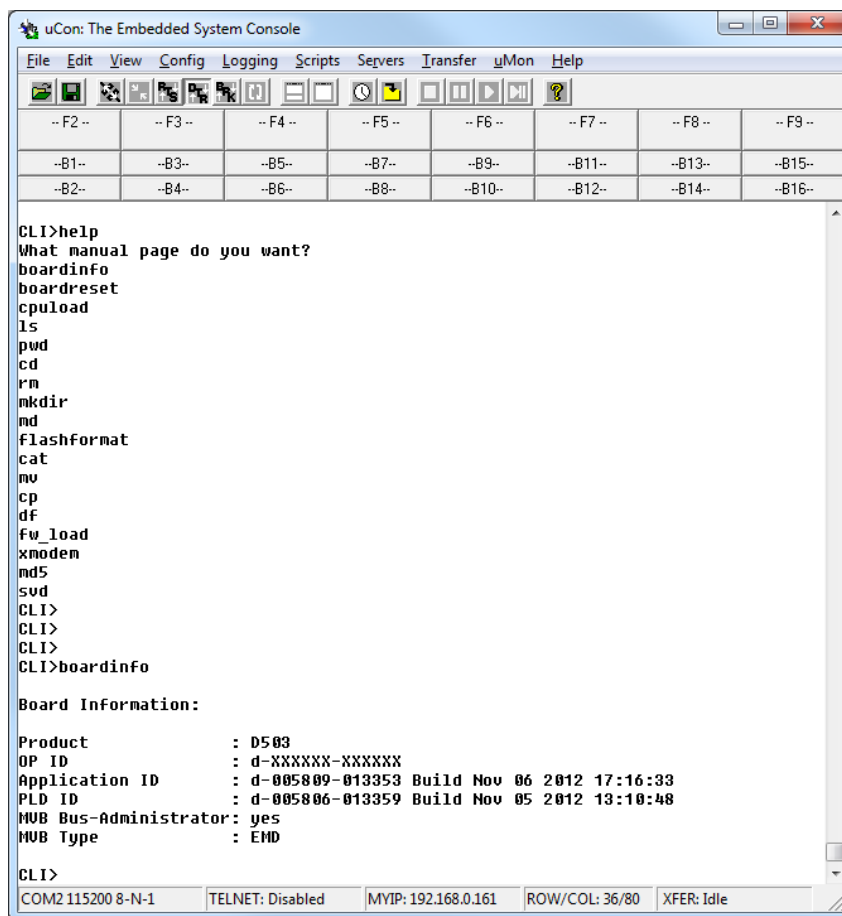


Figure 5: Help Command

The screen above shows the available commands and a typical output for the command `boardinfo`.

Type `help` command to have a quick help for the desired command.

For more information about the CLI, please refer to the Command Line Interface (CLI) data sheet.

4.2 Debugging with CLI

For debugging a newly developed application, insert

```
diag_printf(format, variable, ...);
```

in the source code (`diag_printf` works exactly like the C `printf` function). The output of this function will appear in the CLI.

4.3 Up / downloading files

The uCon terminal program can be used to up / download files from / to the device file system. This can be useful to modify the configuration files.

The procedures below shows how to up / download the `device.cfg` file located in the `/etc` folder.

Uploading to the PC

To upload the `device.cfg` file from the device to the PC (see Figure 6 and Figure 7):

- enter `cd /etc` at the CLI line
 - select the menu entry 'Transfer/Xmodem'
 - enter in the text box 'File' the destination path and file in the PC
 - enter in the text box 'Cmd':
`xmodem -u -F device.cfg`
 - click on 'Recv'
- The `device.cfg` file will be stored in the destination PC file.

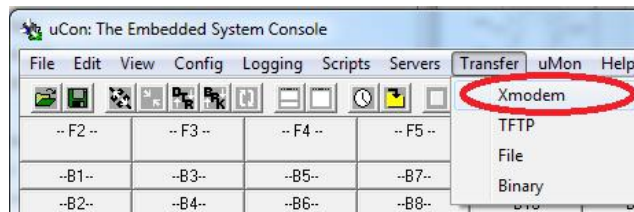


Figure 6: Select Xmodem Menu

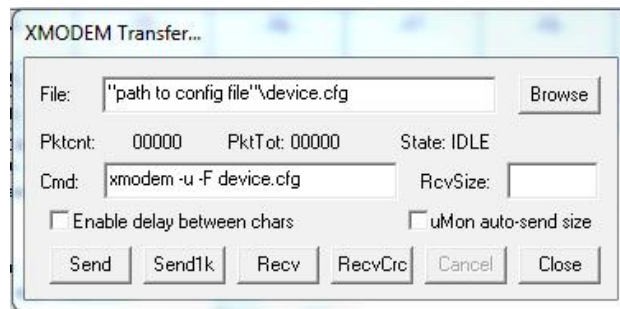


Figure 7: Xmodem Transfer

Downloading from the PC

To download the `device.cfg` file from the PC to the device:

- enter `cd /etc` at the CLI line
 - select the menu entry 'Transfer/Xmodem'
 - enter in the text box 'File' the destination path and file in the PC
 - enter in the text box 'Cmd':
`xmodem -d -F device.cfg`
 - click on 'Send'
- The `device.cfg` file will be stored in the device `/etc` directory.

5. Links to other duagon Documents

D501 RS232-Ethernet Gateway, Data Sheet, d-002540-nnnnnn

D501 Data Sheet. Provides information on connectors.

D502 RS485/RS422-Ethernet Gateway, Data Sheet, d-002552-nnnnnn

D502 Data Sheet. Provides information on connectors.

D503 MVB-Ethernet Gateway, Data Sheet, d-004090-nnnnnn

D503 Data Sheet. Provides information on connectors.

D504 CAN-Ethernet Gateway, Data Sheet, d-004091-nnnnnn

D504 Data Sheet. Provides information on connectors.

Command Line Interface (CLI), Data Sheet, d-002357-nnnnnn

This document describes the features of the Command Line Interface (CLI) available on duagon products.

6. Document History

d-005468-013114

- Updated to new duagon template
- Added section "Dump Flash Image"

d-005468-010937

- First release