FRONTGRADE Gaisler

USER MANUAL

RELEASED JANUARY 2025

Quick start guide for GR740-MINI Evaluation board

GR740-MINI

GR740-MINI Quick Start Guide

GR740-MINI-QSG
Jan 2025, Version 1.2

Frontgrade Gaisler AB | Kungsgatan 12 | SE-411 19 | Göteborg | Sweden | frontgrade.com/gaisler

Table of Contents

1. Introduction

1.1. Overview

This document is a quick start guide for the GR740-MINI Development Board.

The purpose of this document is to get users quickly started using the board.

For a complete description of the board please refer to the GR740-MINI Board User's Manual [RD-1].

The GR740 system-on-chip is described in the GR740 Data Sheet and User’s Manual, [RD-2].

The CertusPro FPGA is described in the CertusPro-NX Family Data Sheet [RD-3].

This quick start guide does not contain as many technical details and is instead how-to oriented. However, to make the most of the guide the user should have glanced through the aforementioned documents and should ideally also be familiar with the GRMON3 debug monitor, [RD-4] .

1.2. IP addresses

The GR740-MINI board is equipped with two Ethernet ports that have default IP addresses 192.168.0.24 and 192.168.0.51. The GR740-MINI should not be connected to an existing network where these addresses may be already occupied. Please see Chapter 4 before conneceting to the network or for more information.

1.3. Handling

ATTENTION : OBSERVE PRECAUTIONS FOR HANDLING ELECTROSTATIC SENSITIVE DEVICES

This unit contains sensitive electronic components which can be damaged by Electrostatic Discharges (ESD). When handling or installing the unit observe appropriate precautions and ESD safe practices.

When not in use, store the unit in an electrostatic protective container or bag.

When connecting/disconnecting cables, ensure that the unit is in an un-powered state.

This equipment has SpW ports that use Low Voltage Differential Signalling (LVDS) which has limited common mode voltage protection. Please refer to the user’s manual for instructions on how to ensure that the grounds of equipment are connected together when using SpaceWire.

1.4. References

Table 1.1. References

RD-1	GR740-MINI User’s Manual [https://gaisler.com/index.php/products/boards/gr740-mini]
RD-2	GR740 Data Sheet and User’s Manual [https://gaisler.com/gr740]
RD-3	CertusPro-NX Family Data Sheet [https://www.latticesemi.com/Products/FPGAandC-PLD/CertusPro-NX]
RD-4	GRMON3 User's Manual [https://www.gaisler.com/doc/grmon3.pdf]
RD-5	GRLIB IP Core User’s Manual [https://www.gaisler.com/products/grlib/grip.pdf]
RD-6	RTEMS homepage [https://www.rtems.org]

RD-7	LEON/ERC32 RTEMS Cross Compilation System (RCC) [https://www.gaisler.com/index.php/products/operating-systems/rtems]
RD-8	RCC User's manual [https://gaisler.com/anonftp/rcc/doc]
RD-9	Frontgrade Gaisler RTEMS driver documentation [https://gaisler.com/anonftp/rcc/doc]
RD-10	Bare C Cross-Compilation System [https://www.gaisler.com/index.php/products/operating-systems/bcc]
RD-11	BCC User's Manual [https://www.gaisler.com/doc/bcc2.pdf]
RD-12	VxWorks 7 SPARC architectural port and BSP [https://www.gaisler.com/index.php/products/operating-systems/vxworks-7]

2. Overview

The GR740-MINI board is software development and evaluation platform for the GR740 (a quad-core radiation tolerant microprocessor) and the CertusPro-NX FPGA. A variant of the CertusPro-NX qualified for use in space is available from Frontgrade in the form of the CertusPro-NX-RT family. shown in Figure 2.1.

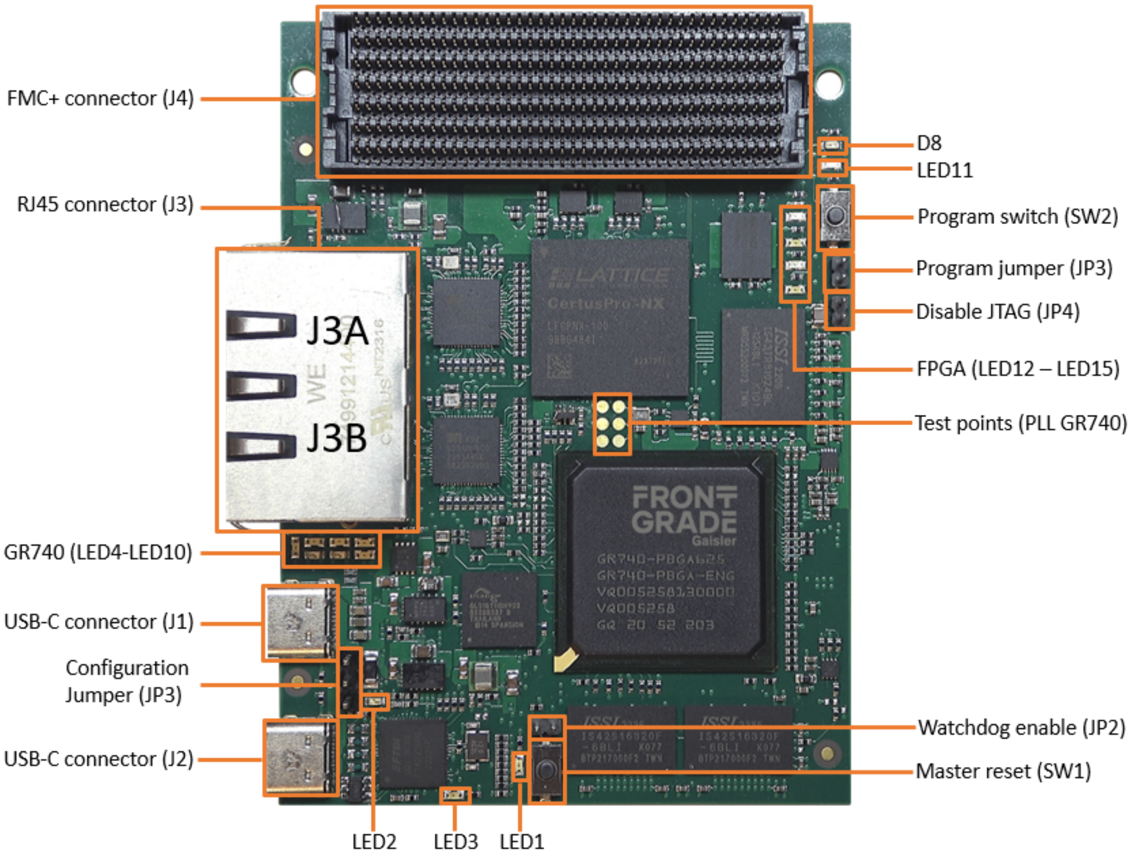


Figure 2.1. Top view of GR740-MINI board

The GR740-MINI architecture includes the following parts:

- 2 x USB-C connectors for debug and power
- GR470 Processor
  - 1 x USB interface via FTDI FT4232HP providing JTAG and UART
  - 1 x Ethernet for communication and debug
  - 4 x SpaceWire channels to FMC+ connector
  - 256MiB SDRAM
  - 128MiB FLASH
- CertusPro-NX FPGA
  - 1 x USB interface via FTDI FT4232HP providing JTAG and UART
  - 1 x Ethernet for communication and debug
  - 4 x SerDes to FMC+ connector
  - LVDS to FMC+ connector
  - 3V3 IO to FMC+ connector
  - I2C link to FMC+ connector
  - 1GiB DDR3 memory
  - 512Mib SPI FLASH
- Intercommunication between GR740 and CertusPro-NX
  - PCI link 32bit 33MHz
  - GMII/MII
  - 4 x SpaceWire channels

3. Board Configuration

3.1. Overview

The primary sources of information for the development board is the GR740-MINI User’s Manual, [RD-1]. The primary information for the Processor is GR740 Data Sheet and User’s Manual [RD-2] and for the FPGA Certus-Pro-NX Family Data Sheet [RD-3].

3.2. Prerequisites

- GR740-MINI board
- USB-C Cable
- Host computer (Windows/Linux)
- GRMON3 hardware debugger (See for Section 4.2 for license options)
- Optional; 5V Power adapter (min 15W)

3.3. Pre programmed

The GR740-MINI Board is preproggramed when delivired. The GR740 is booted from the prom memory with helloblink.prom, built with MKPROM2 se Section 5.3. The boot file is necessary to initialize the SDRAM, L2-cache, clock-gating unit and UART pin-multiplexing, example is available for download on the webpage www.gaisler.com/gr740-mini. The FPGA is configurated from the SPI memory which is flashed with a bitstream, EX-1.bit, please se Chapter 7 for more information about the bitstream.

3.4. Power Up the board

The GR740-MINI board is powered using either of the USB-C connectors (J1 or J2). Therefor there are two ways to power up the board and it is accomplished by attaching a 2-pin jumper to the 3-pin head (JP5), see Figure 3.1.

The overall power requirement for the board is 15W. Note that both the UCB-C cable and source must deliver at least 3A, 5V (15W). Observe that not all sources with USB-C sockets support that. Check the source's manual before connecting to the GR740-MINI board.

J2 is considered as the main power supply, and the configuration jumper is set to this as default JP5: 1-2 mode (blue). This mode can be used when communicating with USB-C (data and power through the same cable), Ethernet or FMC+ connector.

If J2 is connected to a source that cannot provide the minimum power requirement (for example a computer port or a docking station), then J1 can be connected to a external power source. Set jumper JP5 to 2-3 (red) mode to use J1 as the power source.

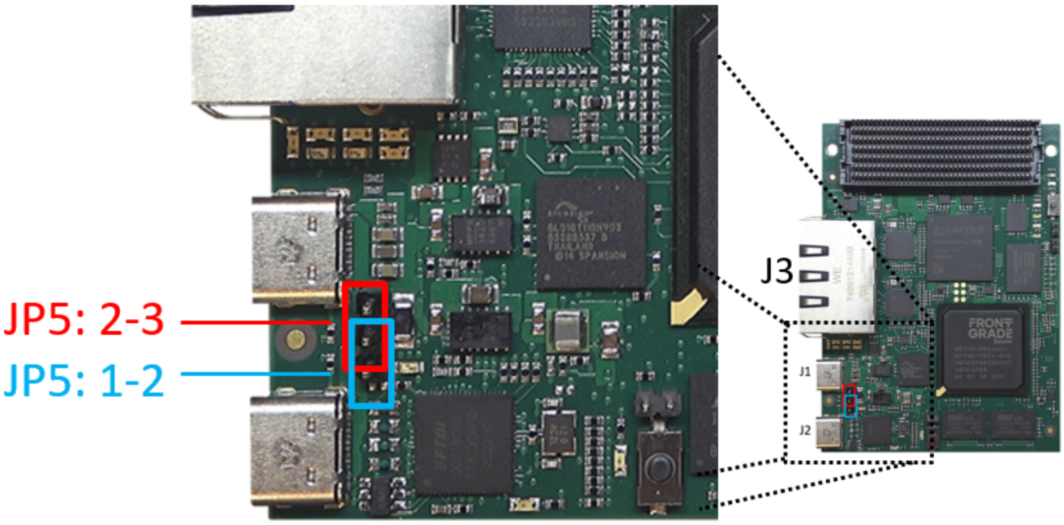


Figure 3.1. Jumper configuration for power supply

3.5. LEDs indication

With the preprogramed devices some LEDs on board are suppost to be on/blinkning during the start-up, see the corresponding parts below.

3.5.1. Power

When powering the board there are 3 LEDs for indicating the availability and communication status.

LED1 : ON, Indicates that 3V3 is available

LED2 : OFF, Light when USB is in suspended mode

LED3 : ON, Indicate normal USB operational mode

3.5.2. GR740

The are LEDs correspondning the GR740, see Figure 3.2 and the list below for behavior.

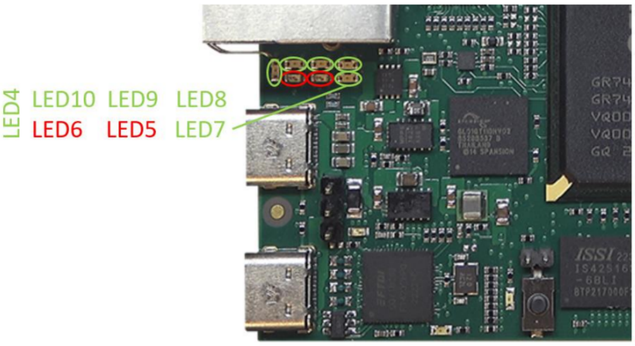


Figure 3.2. LEDs corresponding GR740

LED4 : ON, Indicates debug support unit is active

LED6 : Blinkning

LED7 : Blinkning

LED8 : Blinkning

LED9 : Blinkning

LED10 : Blinkning

3.5.3. FPGA

When the FPGA is configurated there are two LEDs for indicating the configuration status, se below.

D8 : OFF, Indicates that a error occur during configuration

LED11 : ON, Indicates when configuration completed successfully

3.6. Switch buttons

3.6.1. Master reset (SW1)

There is a switch button (SW1) to reset the board. The reset signal will reset the GR740, the Ethernet PHYs, flash memory (GR740) and the FPGA (for the provided example bitstreams). It will not trigger an FPGA reconfiguration or interrupt USB communication

3.6.2. Program switch (SW2)

This switch is connected to the FPGA and will initiate configuration sequence when asserted.

3.7. FTDI chip

A FTDI FT4323HP chip is connected to USB-C connector J2 and provides Serial to USB conversion up to 4 ports. In Table 3.1 the port are listed. The corresponing "vendor id" is 0403 and "product id" is 6043 for the FTDI chip.

Table 3.1. FTDI port assignment

Port	Function	Target
A	JTAG	GR740
B	JTAG	CertusPro-NX
C	UART	GR740
D	UART	CertusPro-NX

Verify that the board is recognized by the host computer. In windows go to Device Manager, in linux use the **lsusb** command in the terminal. If you cannot see the device please refer to Chapter 8

4. GRMON3 hardware debugger

4.1. Overview

GRMON3 is a debug monitor used to develop and debug GRLIB/LEON systems. The target system, including the processor and peripherals, is accessed on the AHB bus through a debug-link connected to the host computer. GRMON3 has GDB support which makes C/C++ level debugging possible by connecting GDB to the GRMON3's GDB socket. With GRMON3 one can for example:

- Inspect LEON and peripheral registers
- Upload applications to RAM with the **load** command.
- Program the FLASH with the **flash** command.
- Control execution flow by starting applications (**run**), continue execution (**cont**), single-stepping (**step**), inserting breakpoints/watchpoints (**bp**) etc.
- Inspect the current CPU state listing the back-trace, instruction trace and disassemble machine code.

The first step is to set up a debug link in order to connect to the board. The following section outlines which debug interfaces are available and how to use them on the GR740-MINI Evaluation Board. After that, a basic first inspection of the board is exemplified.

4.2. GR740 License options

When temporary using (borrowing) a GR740-MINI board the GRMON3 evaluation version which is freely available and can be used to operate the GR740-MINI board. The evaluation version does not support any other GR740 boards. The evaluation version is limited in certain regards compared with the GRMON3 professional product. When purchasing a GR740-MINI board a professional license is required. GRMON3 can be downloaded from the GRMON3 homepage [RD-4].

The following table summarizes the GRMON3 license options for GR740.

Table 4.1. GRMON3 license options for GR740. All references to GRMON3 is for version 3.3.6 or later.

Program version	License	Supported hardware	Supported FPGA designs
GRMON3 professional	Professional	All GR740 systems	All GRLIB designs
GRMON3 evaluation	Evaluation • No cost • No registration	GR740-MINI (borrowed)	• The provided example bitstreams • GRLIB designs with open-source IP cores (GPLv2 license), see GRLIB IP Core User’s Manual [RD-5]

4.3. Debug-link alternatives for GR740

4.3.1. Connecting via the FTDI USB/JTAG interface

Please see GRMON3 User's Manual for how to set up the required FTDI driver software. Verify the serialnumber and that GRMON3 is able to detect the board with the following command

```
grmon -ftdi -jtaglist
```

If it's not showing any devices please refer to Chapter 8

Due to the SDRAM configuration on the board the **-sddcs** flag must be set. **-jtagserial** is used to connect to the board where "A" is indicating port A on the FTDI chip listed in Table 3.1. Connect the PC to the board by using a USB cable to the J2 connector and issue the following command:

```
grmon -ftdi -jtagserial MySerialNumberXXA -sddcs
```

4.3.2. Connecting via the Ethernet debug interfaces

Note: Debugging using Ethernet is not supported by grmon-eval

The GR740 supports one Ethernet debug communication link (EDCL) in the default configuration of the GR740-MINI. The debug link have default address 192.168.0.24. The GR740-MINI should not be connected to an existing network where this address may be already occupied.

If another address is wanted for the Ethernet debug link then the JTAG debug link must be used to connect GRMON3 to the board. The EDCL IP address can then be changed using GRMON3's **edcl** command. This new address will persist until next system reset.

Ethernet debug link traffic is routed to the Debug AHB bus as default. Ethernet debug link traffic can be routed either to the Master I/O AHB bus or to the Debug AHB bus by changing a bootstrap signal. In order to control the LEON processors the debug link must be routed to the Debug AHB bus, otherwise GRMON3 will not be able to use the debug link to access the Debug Support Unit.

After reset the Ethernet debug communication link will attempt to configure the Ethernet PHY. In order for this to succeed, the Ethernet port must be connected to a switch or other networking equipment.

Connect the Ethernet to connector J3B. Due to the SDRAM configuration on the board the **-sddcs** flag must be set. Issue the following command:

```
grmon -eth 192.168.0.24 -sddcs
```

4.4. Debug-link alternatives for CertusPro

The interfaces to debug the CertusPro can vary depening on how the FPGA is programmed. In this chapter the interfaces that will be provided from our pre-built bitsreams will be described.

4.4.1. Connecting via the FTDI USB/UART interface

Please see GRMON3 User's Manual for instructions how to connect GRMON3 to a board using a serial UART connection. The PC is connected using a USB cable to the USB connector J2 and then start GRMON3 with the **-uart PORTNAME** debug-link option and device name.

Example in Linux:

```
grmon -uart /dev/ttyUSB3
```

Example in Windows:

```
grmon -uart \\.\COM6
```

4.4.2. Connecting via the Ethernet debug interfaces

Note: Debugging using Ethernet is not supported by grmon-eval

Ethernet debug communication link (EDCL) have default address 192.168.0.51 The GR740-MINI should not be connected to an existing network where these addresses may be already occupied.

If another address is wanted for the Ethernet debug link then the UART debug link must be used to connect GRMON3 to the board. The EDCL IP address can then be changed using GRMON3's **edcl** command. This new address will persist until next system reset.

The Ethernet debug link traffic is routed to the Debug AHB bus.

After reset the first Ethernet debug communication link will attempt to configure the Ethernet PHY. In order for this to succeed, the Ethernet port must be connected to a switch or other networking equipment. Connect the Ethernet to connector J3A and issue the following command:

```
grmon -eth 192.168.0.51
```

4.5. First steps

The previous sections have described which debug-links are available and how to start using them with GRMON3. The subsections below assume that GRMON3, the host computer and the GR740-MINI board have been set up so that GRMON3 can connect to the board.

When connecting to the board for the first time it is recommended to get to know the system by inspecting the current configuration and hardware present using GRMON3. With the **info sys** command more details about the system is printed and with **info reg** the register contents of the I/O registers can be inspected. Below is a list of items of particular interest:

- AMBA system frequency is printed out at connect, if the frequency is wrong then it might be due to noise in auto detection (small error). See `-freq` flag in the GRMON User's Manual [RD-4].
- Memory location and size configuration is found from the **info sys** output.

4.6. Connecting to the board

4.6.1. GR740

The transcript below shows an example session with GRMON3. GRMON3 is started with the `-u` flag in order to redirect UART output to the GRMON3 terminal.

```
user@user:~/ $ grmon -ftdi -jtagserial MySerialNumberXXA -sddcs -u

  GRMON debug monitor v3.3.6 64-bit eval version

  Copyright (C) 2023 Frontgrade Gaisler - All rights reserved.
  For latest updates, go to https://www.gaisler.com/
  Comments or bug-reports to support@gaisler.com

  This eval version will expire on 28/04/2024


Parsing -ftdi
Parsing -sddcs
Parsing -u

Commands missing help:
 echotrace
 package

JTAG chain (1): GR740
  Device ID:           0x740
  GRLIB build version: 4153
  Detected system:     GR740 rev1
  Detected frequency:  250.0 MHz

  Component                            Vendor
  JTAG Debug Link                      Frontgrade Gaisler
  GRSPW2 SpaceWire Serial Link         Frontgrade Gaisler
  EDCL master interface                Frontgrade Gaisler
  EDCL master interface                Frontgrade Gaisler
  LEON4 SPARC V8 Processor             Frontgrade Gaisler
  LEON4 SPARC V8 Processor             Frontgrade Gaisler
  LEON4 SPARC V8 Processor             Frontgrade Gaisler
  LEON4 SPARC V8 Processor             Frontgrade Gaisler
  IO Memory Management Unit            Frontgrade Gaisler
  AHB-to-AHB Bridge                    Frontgrade Gaisler
  L2-Cache Controller                  Frontgrade Gaisler
  AHB Memory Scrubber                  Frontgrade Gaisler
  IOMMU secondary master i/f           Frontgrade Gaisler
  AHB-to-AHB Bridge                    Frontgrade Gaisler
  LEON4 Debug Support Unit             Frontgrade Gaisler
  AHB/APB Bridge                       Frontgrade Gaisler
  AMBA Trace Buffer                    Frontgrade Gaisler
  AHB/APB Bridge                       Frontgrade Gaisler
  AHB/APB Bridge                       Frontgrade Gaisler
  Muxed FT DDR/SDRAM controller        Frontgrade Gaisler
  Memory controller with EDAC          Frontgrade Gaisler
  GRPCI2 PCI/AHB bridge                Frontgrade Gaisler
  GRSPW Router                         Frontgrade Gaisler
  LEON4 Statistics Unit                Frontgrade Gaisler
  GRPCI2 Trace buffer                  Frontgrade Gaisler
  Generic UART                         Frontgrade Gaisler
  Generic UART                         Frontgrade Gaisler
  General Purpose I/O port             Frontgrade Gaisler
  Multi-processor Interrupt Ctrl.      Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
```

```
GRSPW Router DMA interface           Frontgrade Gaisler
GRSPW Router DMA interface           Frontgrade Gaisler
GRSPW Router DMA interface           Frontgrade Gaisler
GRSPW Router DMA interface           Frontgrade Gaisler
GR Ethernet MAC                      Frontgrade Gaisler
GR Ethernet MAC                      Frontgrade Gaisler
CAN Controller with DMA              Frontgrade Gaisler
CAN Controller with DMA              Frontgrade Gaisler
SPI Controller                       Frontgrade Gaisler
Clock gating unit                    Frontgrade Gaisler
MIL-STD-1553B Interface              Frontgrade Gaisler
AHB Status Register                  Frontgrade Gaisler
AHB Status Register                  Frontgrade Gaisler
General Purpose I/O port             Frontgrade Gaisler
General Purpose Register             Frontgrade Gaisler
Temperature sensor                   Frontgrade Gaisler
General Purpose Register Bank        Frontgrade Gaisler
CCSDS TDP / SpaceWire I/F            Frontgrade Gaisler
LEON4 Statistics Unit                Frontgrade Gaisler
64-bit PC133 SDRAM Controller        Frontgrade Gaisler

Use command 'info sys' to print a detailed report of attached cores
```

4.6.2. CertusPro

Please see Chapter 7 for the output corresponding the FPGA design.

5. Software Development Environment

5.1. Overview

Frontgrade Gaisler provides a comprehensive set of software tools to run several different operating systems. The GR740 platform supports the following:

BCC — the Bare C Cross-Compiler System is a toolchain to compile bare C or C++ applications directly on top of the processor without the services provided by an operating system

RTEMS — a hard Real Time Operating System. Frontgrade Gaisler provides RCC, a toolchain to develop and compile RTEMS applications specifically for the LEON

Linux — the open source operating system. Board Support Packages and tools to ease the compilation and deployment of the kernel are provided

VxWorks — an embedded real-time operating system developed by WindRiver. Frontgrade Gaisler provides a LEON architectural port (HAL) and a Board Support Package (BSP) in full source code

Frontgrade Gaisler also provides a set of debug tools. The GR740 platform is supported by the following:

GRMON — Used to run and debug applications on GR740-MINI hardware. See (Chapter 4).

Developer tools are generally provided for both Linux and Windows host operating systems. Frontgrade Gaisler also provides an integrated, easy-to-use solution to help programmers with the task of developing for the LEON. The LEON Integrated Development Environment for Eclipse (LIDE) is an Eclipse plug-in integrating compilers, software and hardware debuggers in a graphical user interface. The plugin makes it possible to cross-compile C and C++ application for LEON, and to debug them on either simulator and target hardware (TSIM or GRMON3).

The recommended method to load software onto a LEON board is by connecting to a debug interface of the board through the GRMON3 hardware debugger (Chapter 4). Execution of programs by a PROM-loaded boot loader is also possible.

5.2. Bare C Cross-Compiler System

5.2.1. Overview

The Bare C Cross-Compiler (BCC for short) is a GNU-based cross-compilation system for LEON processors. It allows cross-compilation of C and C++ applications for LEON2, LEON3 and LEON4. This section gives the reader a brief introduction on how to use BCC together with the GR740-MINI Evaluation Board. It will be demonstrated how to build an an example program and run it on the GR740-MINI using GRMON3.

The BCC toolchain includes the GNU C/C++ cross-compiler 7.2.0, GNU Binutils, Newlib embedded C library, the Bare-C run-time system with LEON support and the GNU debugger (GDB). The toolchain can be downloaded from [RD-10] and is available for both Linux and Windows. Further information about BCC can be found in [RD-11].

The installation process of BCC is described in [RD-11]. The rest of this chapter assumes that **sparc-gaisler-elf-gcc** is available in the `PATH` variable.

5.2.2. Compiling with BCC

The following command shows an example of how to compile a typical *hello, world* program with BCC.

```
$ cat hello.c
#include <stdio.h>

int main(void)
{
        printf("hello, world\n");
        return 0;
}

$ sparc-gaisler-elf-gcc -mcpu=leon3 -O2 -g hello.c -o hello.elf
```

All GCC options are described in the gcc manual. Some of the most common options are:

Table 5.1. BCC's GCC compiler relevant options

`-g`	generate debugging information - recommended for debugging with GDB
`-msoft-float`	emulate floating-point - must be used if no FPU exists in the system
`-O2`	optimize for speed
`-Os`	optimize for size
`-Og`	optimize for debugging experience
`-qsvt`	use the single-vector trap model
`-mflat`	enable flat register window model. The compiler will not emit `SAVE` and `RESTORE` instructions.

It is recommended to use the options

```
  -mcpu=leon3
```

with GR740. For more details, see [RD-10].

5.2.3. Running application with GRMON3

Once your application is compiled, connect to your GR740-MINI with GRMON3. The following log shows how to load and run the the application using breakpoint. Note that the console output is redirected to GRMON3 by the use of the `-u` command line switch, so that the application standard output is forwarded to the GRMON3 console.

```
grmon3> load hello.elf
  40000000 .text                     23.6kB /  23.6kB   [===============>] 100%
  40005E70 .data                      2.7kB /   2.7kB   [===============>] 100%
  Total size: 26.29kB (806.59kbit/s)
  Entry point 0x40000000
  Image hello.elf loaded

grmon3> bp main
  Software breakpoint 1 at <main>

grmon3> run

  CPU 0:  breakpoint 1 hit
          0x40001928: b0102000  mov  0, %i0  <main+4>
  CPU 1:  Power down mode

grmon3> step
  0x40001928: b0102000  mov  0, %i0  <main+4>

grmon3> step
  0x4000192c: 11100017  sethi  %hi(0x40005C00), %o0  <main+8>

grmon3> cont
hello, world

  CPU 0:  Program exited normally.
```

Alternatively you can run GRMON3 with the `-gdb` command line option and then attach a GDB session to it. For further information see Chapter 3 of [RD-11].

5.3. Boot Loaders

Frontgrade Gaisler provides two boot loaders for the LEON2, LEON3 and LEON4 processors listed below for more information. The boot loaders covers different use cases and requirements on software quality level. The boot loaders are all capable of booting all the supported Operating Systems provided by Frontgrade Gaisler.

MKPROM2
MKPROM2 is a free open-source boot loader supporting a minimal system initialization, extraction of a single ROM application image into main memory and booting it. No system self-tests are performed by MKPROM2.

GRBOOT
The GRBOOT boot loader software is based on the GR712RC Boot SW using the same ECSS software engineering standards previously used to guarantee a high reliability for flight. By isolating mission and device specific parts into BSPs and generalizing the implementation, GRBOOT provides similar a reusable feature set for systems based on LEON3/4FT processor devices acting as either payload or OBC.

One or more application images can be located in parallel flash or SPI flash. Multiprocessor application booting is supported.

GRBOOT is available for GR712RC and GR740 based systems together with the appropriate quality proofs, documentation and test suites. A version without references to the ESA requirements documents is also available.

6. FPGA development Environment

6.1. Lattice Radiant Software

Lattics provides a software development tool for the FPGA development. Lattice Radiant software is available for both linux and windows enviorment.

For download and information about licensing please see webpage: www.latticesemi.com/latticeradiant.

6.2. Program the FPGA

The FPGA can be programmed in the volatile (RAM) memory or non-volatile (SPI flash) memory. The FPGA is programmed with JTAG through the FTDI chip.

6.2.1. Lattice programmer

TBC

6.2.2. GRMON3

GRMON3 can be used to program the FPGA in both linux and windows enviorment, see sections below for prerequisites. Note that GRMON version 3.3.6 or higher is needed.

In order to program the FPGA using GRMON3 it is needed to connect the debug link to GR740 via JTAG. Please se section Section 4.3.1 for how to connect. In GRMON3, the following command is used to program the RAM

```
grmon3> fpgaload ex-1.bit
     write to ram
     Jtag frequency : requested 1.00MHzOpen file: DONE
  Parse file: DONE
  Enable configuration: DONE
  SRAM erase: DONE
  Loading: [==================================================] 100.00%
  Done
  Disable configuration: DONE
```

To program the SPI memoory use the -f flag, see below:

```
grmon3> fpgaload ex-1.bit -f
  write to flash
        Jtag frequency : requested 1.00MHzOpen file DONE
        Parse file DONE
        Enable configuration: DONE
        SRAM erase: DONE
        Detected: Macronix MX25L51245G 1024 sectors size: 512Mb
        00000000 00000000 00000000 00
        Erasing: [==================================================] 100.00%
        Done
        Writing: [==================================================] 100.00%
        Done
        Refresh: DONE
```

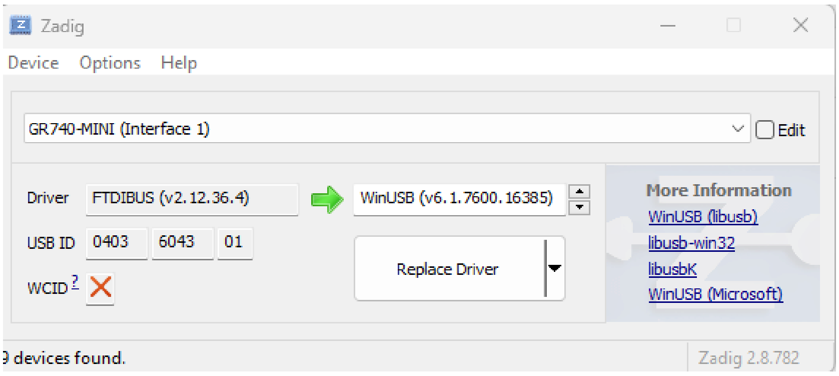
6.2.2.1. Linux prerequisites

Prerequisites: the following programs needs to be installed:

- libusb-1.0
- libftdi1
- libudev
- libhidapi-hidraw
- libz
- glibc-2.29 or higher
- libstdc++.so.x.x.21 or higher

6.2.2.2. Windows prerequisites

To be able to program the FPGA you must replace the driver for the FT4232 interface connected to FPGA. The tool Zadig can be used for this purpose and can be downloaded from https://zadig.akeo.ie/ . Start Zadig and click on "Options" and enable "List All Devices" from the drop-down menu. Select GR740-MINI (Interface 1) and set the driver to WinUSB as in Figure 6.1. Click on "Replace Driver"



GR740-MINI (Interface 1) □ Edit

Driver FTDIBUS (v2.12.36.4) → WinUSB (v6.1.7600.16385)

USB ID 0403 6043 01

WCID ? ✗

9 devices found.

Zadig 2.8.782

Figure 6.1. Zadig interface

7. Example Bitstreams

7.1. Overview

For the GR740-MINI board, example bitstreams for the CertusPro-NX are available for download on the webpage www.gaisler.com/gr740-mini. The bitstreams are based on an architecture centered around the AMBA Advanced High-speed Bus (AHB), to which high-bandwidth units are connected. Low-bandwidth units are connected to the AMBA Advanced Peripheral Bus (APB) which is accessed through an AHB to APB bridge.

In the sections below we shortly present the different bistreams and corresponding IP cores also a simplified interactions is shown. The AHB/APB memory mapping and interrupts are not included in this document but, can be displayed for each design using the **info sys** command in GRMON3.

For more information considering the operations of each individual core, please refer to the grip documentation available at GRLIB IP Core User’s Manual

7.2. EX-1

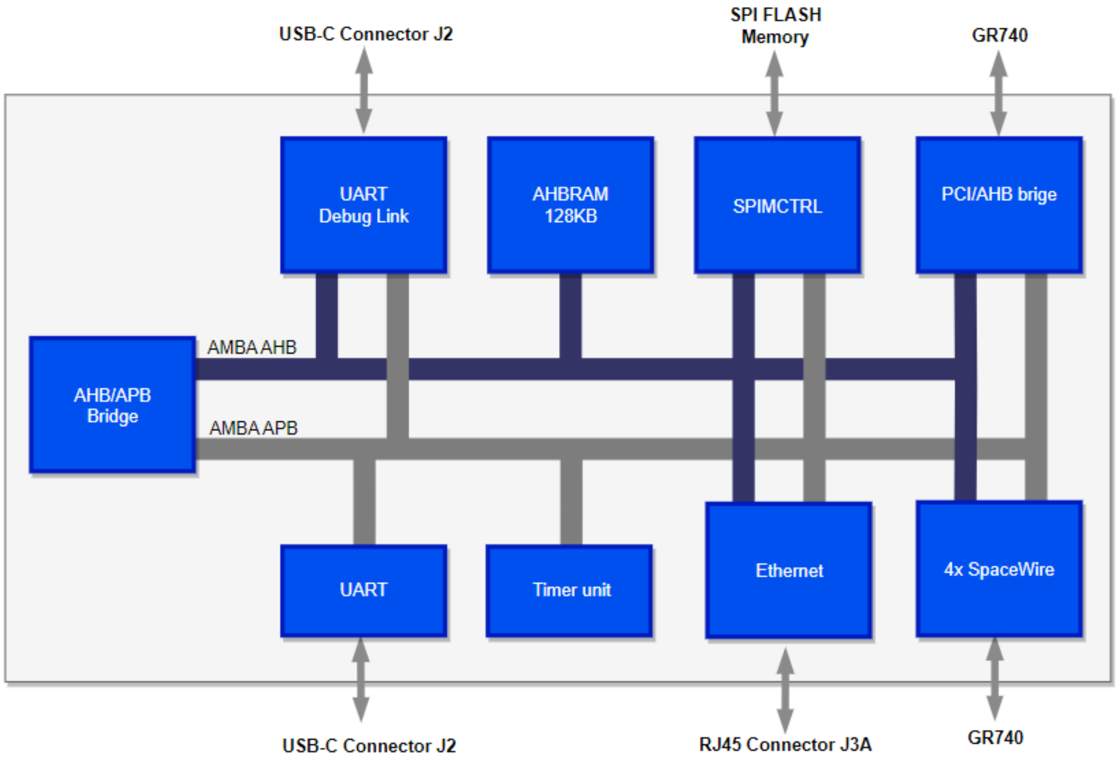


Figure 7.1. Architectural block diagram of bitstream EX-1

Expected output when connecting with GMON3.

```
user@user:~/ $ grmon -uart /dev/ttyUSB3 -u

GRMON debug monitor v3.3.6 64-bit eval version

Copyright (C) 2023 Frontgrade Gaisler - All rights reserved.
For latest updates, go to https://www.gaisler.com/
Comments or bug-reports to support@gaisler.com

This eval version will expire on 28/04/2024

Parsing -uart /dev/ttyUSB3
Parsing -u

using port /dev/ttyUSB3 @ 115200 baud
Device ID:           0x801
```

```
  GRLIB build version: 4283
  Detected frequency:  50.0 MHz

  Component                            Vendor
  AHB Debug UART                       Frontgrade Gaisler
  GR Ethernet MAC                      Frontgrade Gaisler
  GRPCI2 PCI/AHB bridge                Frontgrade Gaisler
  GRSPW2 SpaceWire Serial Link         Frontgrade Gaisler
  GRSPW2 SpaceWire Serial Link         Frontgrade Gaisler
  GRSPW2 SpaceWire Serial Link         Frontgrade Gaisler
  GRSPW2 SpaceWire Serial Link         Frontgrade Gaisler
  AHB/APB Bridge                       Frontgrade Gaisler
  SPI Memory Controller                Frontgrade Gaisler
  Single-port AHB SRAM module          Frontgrade Gaisler
  Generic UART                         Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler

  Use command 'info sys' to print a detailed report of attached cores
```

7.2.1. UART

This design includes UART IP cores thats connect to J2 through a FTDI chip. Please refere to Section 4.4.1 for how to interface GRMON3 using UART to the FPGA.

7.2.2. SPIMCTRL

This IP block allows the user to access the SPI Flash memory featured on the board. The user can set the IP and the memory to work in Extended, Dual or Quad SPI. By issuing **spim flash detect**, GRMON3 can recognise the memory and therefore load the needed settings to operate.

```
grmon3> spim flash detect
         Got manufacturer ID 0xc2 and device ID 0x201a
         Detected device: Macronix MX25L51245G
```

7.2.3. PCI

In this bitstream the FPGA is configured as a target and can therefore not been found using the **info -reg pci0** command. Since the FPGA is a target, only the GR740 can perform an access. The GR740 is the host and need to perform configuration when seting up the PCI.

```
******  GR740  ******
grmon3> pci init
grmon3> pci conf
  PCI devices found:
  Bus 0 Slot 0 function: 0 [0x0]
        Vendor id: 0x1ac8 (Aeroflex Gaisler)
        Device id: 0x55 (Unknown device)
        BAR 0: 80000008 [64MB]
        BAR 1: 84000008 [1MB]
grmon3> # Sets the AMBA adress for the RAM (FPGA)
grmon3> pci wcfg32 0:0:0 0x44 0x40000000
grmon3> # Read the RAM memory corresponing the FPGA
grmon3> mem 0x80000000
  0x80000000  46504741  20436572  74757350  726f0000    FPGA CertusPro..
  0x80000010  00000000  00000000  00000000  00000000    ................
  0x80000020  00000000  00000000  00000000  00000000    ................
  0x80000030  00000000  00000000  00000000  00000000    ................
```

7.2.4. Ethernet

This bitstream provides ethernet in 10/100/1000 Mbit speed. By default the PHY is set to 1000Mbit mode and the IP address is 192.168.0.51. The ethernet is connected to connector J3A. To evalauate how the PHY is configurated use the **mdio info** command. Please refere to Section 4.3.2 for how to interface GRMON3 with ethernet to the FPGA.

```
grmon3> mdio info
  greth0: PHY address 2
  Model: Micrel KSZ9031
  Link: 1000Mbps Full Duplex (autoneg on)
```

7.2.5. SpaceWire

There are four SpaceWire in this design, all connecting to the GR740. See Table 7.1 for the SpaceWire connection between the devices.

Table 7.1. SpaceWire connection

SpaceWire port (FPGA)	SpaceWire port (gr740)
grspw0	spwrtr0::port5
grspw1	spwrtr0::port6
grspw2	spwrtr0::port7
grspw3	spwrtr0::port8

To establish a SpaceWire link, there is need to set one link to "autostart" and the other one to "linkstart". This can be performed with the following commands:

```
****** FPGA ******
grmon3> set grspw0::ctrl::as 1
grmon3> set grspw1::ctrl::as 1
grmon3> set grspw2::ctrl::as 1
grmon3> set grspw3::ctrl::as 1

****** GR740 ******
grmon3> set spwrtr0::pctrl_5::ls 1
grmon3> set spwrtr0::pctrl_6::ls 1
grmon3> set spwrtr0::pctrl_7::ls 1
grmon3> set spwrtr0::pctrl_8::ls 1
```

7.3. EX-2

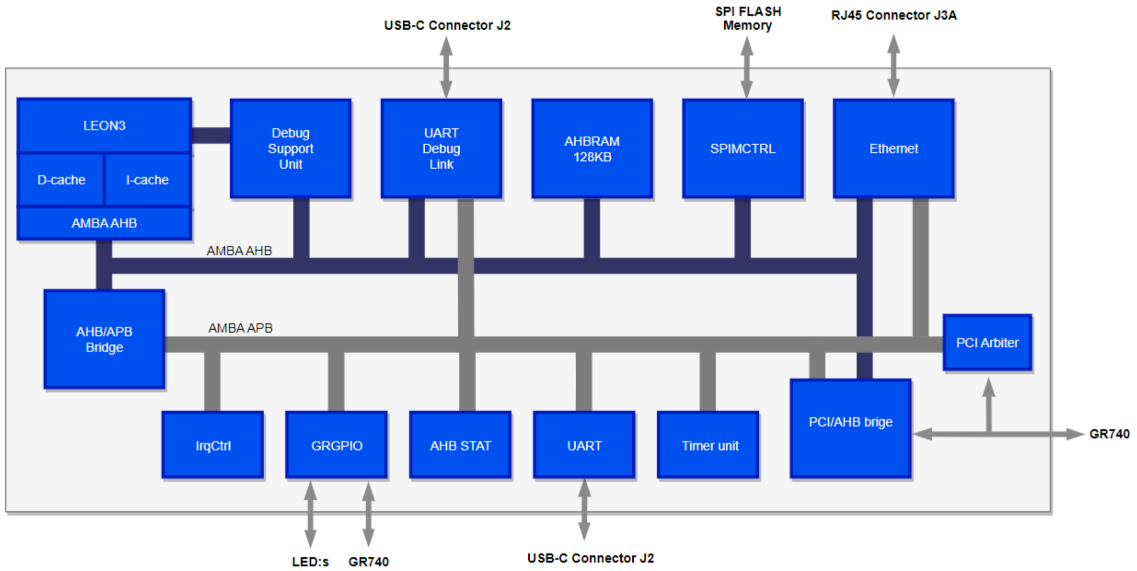


Figure 7.2. Architectural block diagram of bitstream EX-2

Expected output when connecting with GMON3.

```
user@user:~/ $ grmon -uart /dev/ttyUSB3 -u

GRMON debug monitor v3.3.6 64-bit eval version

Copyright (C) 2023 Frontgrade Gaisler - All rights reserved.
For latest updates, go to https://www.gaisler.com/
Comments or bug-reports to support@gaisler.com
```

```
This eval version will expire on 28/04/2024

Parsing -uart /dev/ttyUSB3
Parsing -u

using port /dev/ttyUSB3 @ 115200 baud
Device ID:           0x801
GRLIB build version: 4288
Detected frequency:  50.0 MHz

Component                            Vendor
LEON3 SPARC V8 Processor             Frontgrade Gaisler
AHB Debug UART                       Frontgrade Gaisler
GR Ethernet MAC                      Frontgrade Gaisler
GRPCI2 PCI/AHB bridge                Frontgrade Gaisler
LEON3 Debug Support Unit             Frontgrade Gaisler
AHB/APB Bridge                       Frontgrade Gaisler
SPI Memory Controller                Frontgrade Gaisler
Single-port AHB SRAM module          Frontgrade Gaisler
LEON3 Statistics Unit                Frontgrade Gaisler
Generic UART                         Frontgrade Gaisler
Modular Timer Unit                   Frontgrade Gaisler
General Purpose I/O port             Frontgrade Gaisler
AHB Status Register                  Frontgrade Gaisler
Multi-processor Interrupt Ctrl.      Frontgrade Gaisler
PCI Arbiter                          European Space Agency

Use command 'info sys' to print a detailed report of attached cores
```

7.3.1. LEON 3

The LEON3 is instantiated in the GP (General Purpose) configuration. The following log shows how to load and run an application by steping using breakpoints. Note that the console output is redirected to GRMON3 by the use of the **-u** command line switch, so that the application standard output is forwarded to the GRMON3

```
grmon3> load hello.elf
             40005f80 .rodata            128B              [===============>] 100%
             40006000 .data              1.2kB /   1.2kB   [===============>] 100%
  Total size: 25.16kB (118.98kbit/s)
  Entry point 0x40000000
  Image hello.elf loaded
grmon3> bp main
  Software breakpoint 1 at <main>
grmon3> run
  Breakpoint 1 hit
  0x40001254: b0102000  mov  0, %i0  <main+4>
grmon3> step
  0x40001254: b0102000  mov  0, %i0  <main+4>
grmon3> step
  0x40001258: 11100017  sethi  %hi(0x40005c00), %o0  <main+8>
grmon3> cont
  hello, world
    Program exited normally
```

Below is shown some corresponing signals to LEON3, tied to the LEDs on the board.

PROC_ERRORN : LED12

DSU_ACTIVE : LED13

7.3.2. GPIO

Through this IP block, the user can control GPIO signals to the following components:

GPIO0 : LED14 (active HIGH)

GPIO1 : LED14 (active HIGH)

GPIO2 : GR740::GPIO2[0]

GPIO3 : GR740::GPIO2[1]

GPIO4 : GR740::GPIO2[2]

GPIO5 : GR740::GPIO2[3]

GPIO6 : GR740::GPIO2[4]

The following command sequence shows one example of usage of the gpio register.

```
grmon3> # Set gpio 0 to HIGH (LED 14)
grmon3> set gpio0::iooutput 0x00000001
grmon3> # Set gpio 1 to HIGH (LED 15)
grmon3> set gpio0::iooutput 0x00000002
```

7.3.3. UART

This design includes UART IP cores thats connect to J2 through a FTDI chip. Please refere to Section 4.4.1 for how to interface GRMON3 using UART to the FPGA.

7.3.4. SPIMCTRL

This IP block allows the user to access the SPI Flash memory featured on the board. The user can set the IP and the memory to work in Extended, Dual or Quad SPI. By issuing **spim flash detect**, GRMON3 can recognise the memory and therefore load the needed settings to operate

```
grmon3> spim flash detect
         Got manufacturer ID 0xc2 and device ID 0x201a
         Detected device: Macronix MX25L51245G
```

7.3.5. PCI

In this design the FPGA is instantiated both as a target and master. The GR740 is the host and need to do some setup to configurate the PCI. There is a pci arbiter (IP block) instantiated in the design which purpose is to handle multiple masters. Example below show access to the RAM/SDRAM.

```
******  GR740  ******
grmon3> pci init
grmon3> pci conf
  Bus 0 Slot 0 function: 0 [0x0]
        Vendor id: 0x1ac8 (Aeroflex Gaisler)
        Device id: 0x55 (Unknown device)
        BAR 0: 80000008 [64MB]
        BAR 1: 84000008 [1MB]
grmon3> # Sets the AMBA adress to the RAM to the FPGA
grmon3> pci wcfg32 0:0:0 0x44 0x40000000
grmon3> # Read the RAM memory corresponing the FPGA
grmon3> mem 0x80000000
  0x80000000  46504741  20436572  74757350  726f0000    FPGA CertusPro..
  0x80000010  00000000  00000000  00000000  00000000    ................
  0x80000020  00000000  00000000  00000000  00000000    ................
  0x80000030  00000000  00000000  00000000  00000000    ................
grmon3> # Reading Status and Command register
grmon3> pci cfg32 0:0:0 0x04
  0x02100002
grmon3> # Enable bus masters individually, set bit 2 (BM) high
grmon3> pci wcfg32 0:0:0 0x04 0x0210006

******  FPGA  ******
grmon3> # Read the SDRAM memory corresponing the GR740
grmon3> mem 0xc0000000
  0xc0000000  47523734  30205261  64486172  64207175    GR740 RadHard qu
  0xc0000010  61642d63  6f726500  53504152  43000000    ad-core.SPARC...
  0xc0000020  00000000  00000000  00000000  00000000    ................
  0xc0000030  00000000  00000000  00000000  00000000    ................
```

7.3.6. Ethernet

In this design ethernet is only supported in 10 Mbit mode, and therefore the user need to configure the PHY using the **wmdio** command. By default the IP address is 192.168.0.24 and the ethernet is connected to connector J3A. Please refere to Section 4.4.2 for how to interface GRMON3 with ethernet to the FPGA.

```
grmon3> wmdio 2 0 0x0100
  Writing dev0: PHY addr: 2  Reg addr: 0  Value: 100
```

```
grmon3> mdio info
  greth0: PHY address 2
  Model: Micrel KSZ9031
  Link: 10Mbps Full Duplex (autoneg off)
```

7.4. EX-3

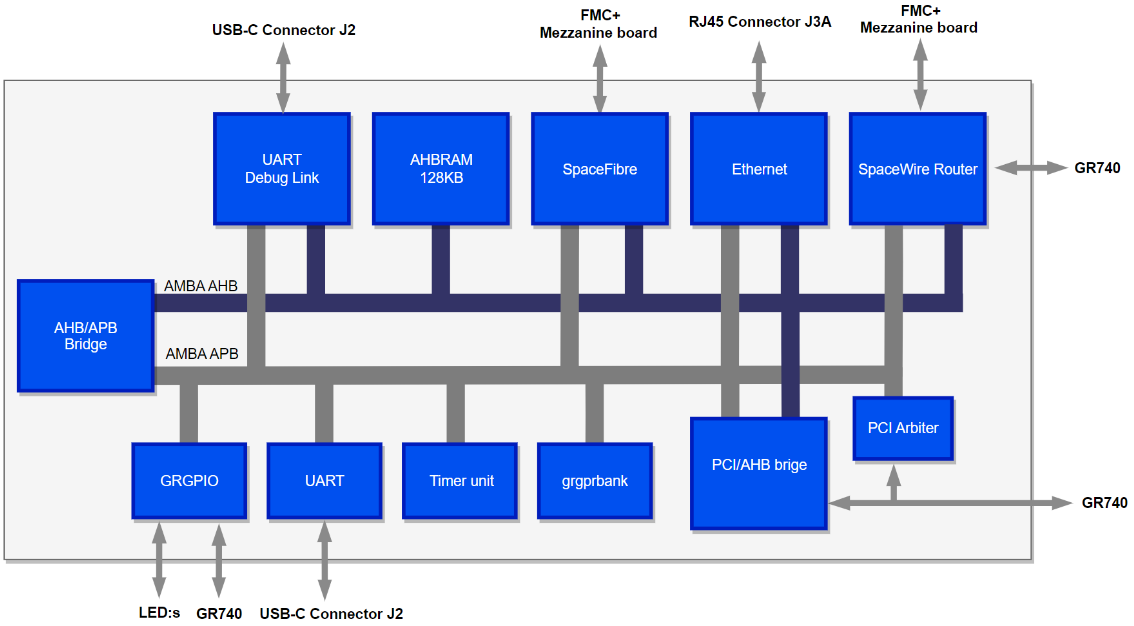


Figure 7.3. Architectural block diagram of bitstream EX-3

Expected output when connecting with GMON3.

```
user@user:~/ $ grmon -uart /dev/ttyUSB3 -u

GRMON debug monitor v3.3.6 64-bit eval version

Copyright (C) 2023 Frontgrade Gaisler - All rights reserved.
For latest updates, go to https://www.gaisler.com/
Comments or bug-reports to support@gaisler.com

This eval version will expire on 28/04/2024

Parsing -uart /dev/ttyUSB3
Parsing -u

using port /dev/ttyUSB3 @ 115200 baud
Device ID:           0x801
GRLIB build version: 4288
Detected frequency:  50.0 MHz

Component                             Vendor
AHB Debug UART                        Frontgrade Gaisler
GR Ethernet MAC                       Frontgrade Gaisler
GRPCI2 PCI/AHB bridge                 Frontgrade Gaisler
GRPCI2 DMA interface                  Frontgrade Gaisler
GRSPW Router DMA interface            Frontgrade Gaisler
GRSPFI SpaceFibre Serial Link         Frontgrade Gaisler
AHB/APB Bridge                        Frontgrade Gaisler
Single-port AHB SRAM module           Frontgrade Gaisler
GRSPW Router                          Frontgrade Gaisler
Generic UART                          Frontgrade Gaisler
Modular Timer Unit                    Frontgrade Gaisler
General Purpose I/O port              Frontgrade Gaisler
PCI Arbiter                           European Space Agency
General Purpose Register Bank         Frontgrade Gaisler

Use command 'info sys' to print a detailed report of attached cores
```

7.4.1. UART

This design includes UART IP cores thats connect to J2 through a FTDI chip. Please refere to Section 4.4.1 for how to interface GRMON3 using UART to the FPGA.

7.4.2. SPIMCTRL

This IP block allows the user to access the SPI Flash memory featured on the board. The user can set the IP and the memory to work in Extended, Dual or Quad SPI. By issuing **spim flash detect**, GRMON3 can recognise the memory and therefore load the needed settings to operate

```
grmon3> spim flash detect
         Got manufacturer ID 0xc2 and device ID 0x201a
         Detected device: Macronix MX25L51245G
```

7.4.3. PCI

In this design the FPGA is instantiated both as a target and master. The FPGA is the host and need to do some setup to configurate the PCI. There is a pci arbiter (IP block) instantiated in the design which purpose is to handle multiple masters. Example below show access to the RAM/SDRAM.

```
******  FPGA  ******
grmon3> pci init
grmon3> pci conf
  Bus 0 Slot 0 function: 0 [0x0]
        Vendor id: 0x1ac8 (Aeroflex Gaisler)
        Device id: 0x740 (Unknown device)
        IRQ INTA#  LINE: 0
        BAR 0: c8000008 [128MB]
        BAR 1: c0000008 [128MB]
        BAR 2: 0 [8MB]
grmon3> # Sets the AMBA adress to the SDRAM to the GR740
grmon3> pci wcfg32 0:0:0 0x44 0x00000000
grmon3> # Read the RAM memory corresponing the GR740 (byte twisted)
grmon3> mem 0xC8000000
  0xc8000000  34375247  61522030  72614864  75712064    47RGaR 0raHduq d
  0xc8000010  632d6461  0065726f  00000001  00000001    c-da.ero........
  0xc8000020  0020d091  00000001  00000001  00000001    . ..............
  0xc8000030  0020d091  00000001  00000001  00000001    . ..............
grmon3> # Reading Status and Command register
grmon3> pci cfg32 0:0:0 0x04
  0x02100002
grmon3> # Enable bus masters individually, set bit 2 (BM) high
grmon3> pci wcfg32 0:0:0 0x04 0x0210006

******  GR740  ******
grmon3> # Read the RAM memory corresponing the FPGA (byte twisted)
grmon3> mem 0x80000000
  0x80000000  41475046  72654320  50737574  00006f72    AGPFreC Psut..or
  0x80000010  00000000  00000000  00000000  00000000    ................
  0x80000020  00000000  00000000  00000000  00000000    ................
  0x80000030  00000000  00000000  00000000  00000000    ................
```

7.4.4. Ethernet

This bitstream provides ethernet in 10/100/1000 Mbit speed. By default the PHY is set to 1000Mbit mode and the IP address is 192.168.0.51. The ethernet is connected to connector J3A. To evalauate how the PHY is configurated use the **mdio info** command. Please refere to Section 4.3.2 for how to interface GRMON3 with ethernet to the FPGA.

```
grmon3> mdio info
  greth0: PHY address 2
  Model: Micrel KSZ9031
  Link: 1000Mbps Full Duplex (autoneg on)
```

7.4.5. GPIO

Through this IP block, the user can control GPIO signals to the following components:

GPIO0 : LED14 (active HIGH)

GPIO1 : LED14 (active HIGH)

GPIO2 : GR740::GPIO2[0]

GPIO3 : GR740::GPIO2[1]

GPIO4 : GR740::GPIO2[2]

GPIO5 : GR740::GPIO2[3]

GPIO6 : GR740::GPIO2[4]

The following command sequence shows one example of usage of the gpio register.

```
grmon3> # Set gpio 0 to HIGH (LED 14)
grmon3> set gpio0::iooutput 0x00000001
grmon3> # Set gpio 1 to HIGH (LED 15)
grmon3> set gpio0::iooutput 0x00000002
```

7.4.6. General Purpose Register Bank

In the design there is a general-purpose register bank implemented. This register bank enables selection of clock source to the SerDes (SpaceFibre) block in the FPGA. For more information about the clock architecture in the FPGA please refer to https://www.latticesemi.com/view_document?document_id=53257 for the Technical Note (CertusPro-NX SerDes/PCS User Guide).

In this bitstream only quad 1 is implemented. Each quad can however choose between their own ref clock (the onboard clock) or the external clocks (connected to the FMC connector). Please see the GR740-MINI Board User's Manual for schematics. The register bank contains 5 register, where register 5 is the configuration register, desciberd below.

15	14 8	7	6	5	4	3	2	1	0
	RESERVED			R*				R*	
0	0	0	0	0	0	0	0	0	0
rw	r	rw	rw	r	rw	rw	rw	r	rw

R* = RESERVED

15	Active-low reset signal for SerDes. Must be set to 1 to enable SerDes.
14:8	RESERVED
7	Directly controls the active-high enable signal for the X7 100.00 MHz external oscillator which provides the quad-local reference clock for SerDes quad 1.
6	RESERVED
5	Reference clock source selection for SerDes quad 1. 0: Use quad-local reference clock (requires bit 7 = 1). 1: Use shared reference clock from internal mux (see bit 4)
4	Shared reference clock mux control for quad 1. This bit only has an effect when bit 5 = 1. 0: Select SD_EXT0_REFCLK (connected to GBTCLK1 from FMC+ connector). 1: Select SD_EXT1_REFCLK (connected to GBTCLK0 from FMC+ connector).
3	Directly controls the active-high enable signal for the X5 156.25 MHz external oscillator which provides the quad-local reference clock for SerDes quad 0.
2	RESERVED
1	Reference clock source selection for SerDes quad 0. 0: Use quad-local reference clock (requires bit 3 = 1). 1: Use shared reference clock from internal mux (see bit 0)
0	Shared reference clock mux control for quad 0. This bit only has an effect when bit 1 = 1. 0: Select SD_EXT0_REFCLK (connected to GBTCLK1 from FMC+ con-

nector). 1: Select SD_EXT1_REFCLK (connected to GBTCLK0 from FMC+ connector).

The following command sequence shows one example of usage of the controll register.

```
grmon3> # Sets quad 1 ref clock as clock source
grmon3> wmem 0x80010010 0x8080
grmon3> # Validating selected frequency
grmon3> set a [mem 0x80010000 4]; set t0 [clock micros]; after 100; set b [mem 0x80010000 4];
set t1 [clock micros]; puts [expr 2*1024*($b - $a)/(1.0*($t1 - $t0))];
  0x80010000  0000610b                                ..a.
  0x80010000  000074a6                                ..t.
  100.90819131390874
grmon3> # Sets quad 1 external clock as clock source
grmon3> wmem 0x80010010 0x8030
grmon3> # Validating selected frequency
grmon3> set a [mem 0x80010000 4]; set t0 [clock micros]; after 100; set b [mem 0x80010000 4];
set t1 [clock micros]; puts [expr 2*1024*($b - $a)/(1.0*($t1 - $t0))];
  0x80010000  00001c02                                ....
  0x80010000  00003468                                ..4h
  124.95416715508146
```

7.4.7. SpaceFibre

In this design only one quad (quad 1) is implemented. The SpaceFibre link is connected to DP0 signals on the FMC connector. Refere to the GR740-MINI Board User's Manual for schematics. See Section 7.4.9 for a application using the SpaceFibre.

7.4.8. SpaceWire Router

In this design the SpaceWire port 1 is connected to the FMC connector (LA14, LA09, LA01 and LA18) and port 2 is connected to the port 6 of the GR740. Refere to the GR740-MINI Board User's Manual for schematics. See Section 7.4.9 for a application using the SpaceWire Router.

7.4.9. SpaceFibre-to-SpaceWire bridge application

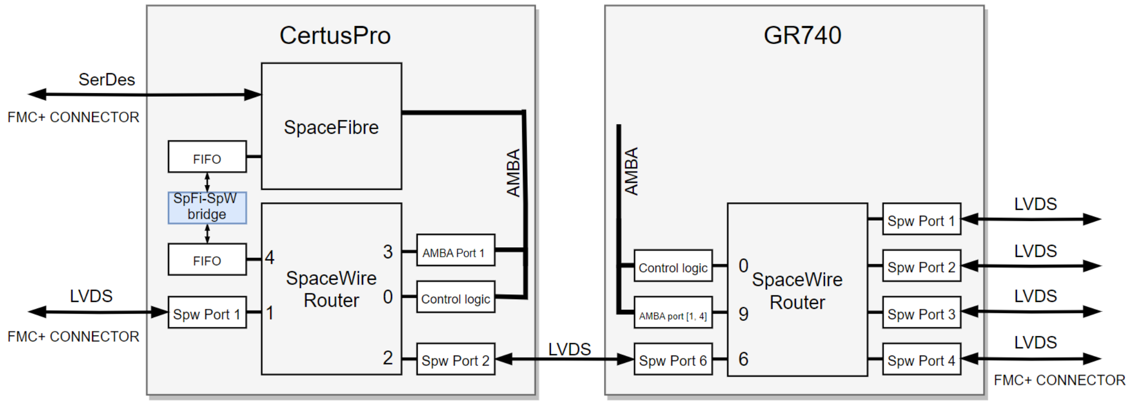


Figure 7.4. SpaceWire-to-SpaceFibre bridge

This design support the application of using a bridge between SpaceFibre and SpaceWire. Data transfer is bidirectional. Make sure to set appropriate clock source using the control register described above. The following program listing show how configure the bridge and how to transmitt data from GR740 to SpaceFibre.

```
******  FPGA  ******
grmon3> # Set the SpaceFibre to autostart
grmon3> set grspfi0::cctrl::as 1
grmon3> # Set the SpaceWire to autostart
grmon3> set spwrtr0::pctrl_1::as 1
grmon3> # Set the external equipment to be in "run-state"
grmon3> # GRSPFI0 setup: forward Virtual Channel 0 data to external FIFO port.
grmon3> # Virtual Channel 0: All timeslots allowed, BW allocation 95%
grmon3> set grspfi0::defaddr::defmsk 0xff
```

```
grmon3> set grspfi0::defaddr::defaddr 0xfe
grmon3> set grspfi0::dc0map 0x1
grmon3> set grspfi0::vc0ctrl::bd 1
grmon3> set grspfi0::vc0ctrl::efc 0x0
grmon3> set grspfi0::vc0ctrl::vbw 0x000d
grmon3> set grspfi0::dc0ctrl 0x3
grmon3> set grspfi0::spfien 0x1
grmon3> set grspfi0::vc0ts1 0xffffffff
grmon3> set grspfi0::vc0ts2 0xffffffff
grmon3> ## GRSPW0 and SPW router: enable and initialize time-codes
grmon3> set spwrtr0::tc::re 0x1
grmon3> set spwrtr0::tc::en 0x1
grmon3> set grspw0::ctrl::rs 0x1
grmon3> set grspw0::time::timecnt 0
grmon3> set grspw0::ctrl::re 0x1
grmon3> set grspw0::ctrl::pm 0x1
grmon3> set grspw0::defaddr::defmask 0xff
grmon3> set grspw0::defaddr::defaddr 0xfe
grmon3> set grspw0::dkey::destkey 0x00
grmon3> set grspw0::dma0rxmax 0x100

******  GR740  ******
grmon3> # Set the SpaceWire to autostart
grmon3> set spwrtr0::pctrl_6::ls 1
grmon3> set grspw1::defaddr::defmask 0xff
grmon3> set grspw1::defaddr::defaddr 0xfe
grmon3> # Configurate the tx descpitor
grmon3> set grspw1::dma0td 0x00001000
grmon3> wmem 0x00001000 0x3000 0x00003000 0x400 0x00003000
grmon3> wmemb 0x00003000 6 4 0xfe 0x55 0xaa
grmon3> set grspw1::dma0ctrl::td 1
grmon3> set grspw1::dma0ctrl::te 1
```

8. Frequently Asked Questions / Common Mistakes / Know Issues

8.1. Why is there no /dev/ttyUSB on Linux?

On Linux, the ftdi_sio kernel module is responsible for creating /dev/ttyUSB. Prior to kernel version 6.1 ftdi_sio does not automatically recognize the FT4232HP serial converter that is used on the GR740-MINI board. However, in earlier versions of Linux it is possible to temporarily (persisting until reboot) add support for the FT4232HP by adding the corresponding USB VID:PID identification to the driver. This can be done with the below command:

```
>>> echo 0403 6043 > /sys/bus/usb-serial/drivers/ftdi_sio/new_id
```

8.2. Why am I getting "invalid cable"?

On Linux this happens when the USB device is in use by another program. This can for example happen if another instance of GRMON is already connected to the selected cable. But the more common situation is that the ftdi_sio kernel module has attached it to a /dev/ttyUSB serial device.

The availability for all ports can be printed with the command "grmon -ftdi -jtaglist". Whether the ftdi_sio module is attached to a particular port can be seen in the output of "lsusb -t". The ftdi_sio module can unloaded using

```
>>> rmmod ftdi_sio
```

which will detach all /dev/ttyUSB serial ports and make them available for use by GRMON. It is also possible to selectively detach the /dev/ttyUSB corresponding to the desired port using the shell-script "ftdi_sio_detach.sh". This script can be found in the share-folder in the GRMON installation. Typically "grmon-pro-3.x.x/linux/share/grmon/tools/ftdi_sio_detach.sh". To remove /dev/ttyUSB0:

```
>>> ./ftdi_sio_detach.sh /dev/ttyUSB0
```

If there are no other FTDI devices connected to the host than the one on the GR740-MINI board, then /dev/ttyUSB0-3 will normally correspond to FT4232HP port A-D. I.e. /dev/ttyUSB0 will be port A (GR740 JTAG) and /dev/ttyUSB1 will be port B (FPGA JTAG).

8.3. Can I use GRMON2?

Using GRMON2 with the GR740-MINI is not supported. GRMON2 does not recognize the USB-JTAG converter used on this board. In principle, it may be possible to connect GRMON2 to the FPGA using UART, and to the GR740 through Ethernet. But neither configuration is tested or supported

9. Revision History

Table 9.1. Revision History

Issue	Date	Section	Description
1.0	2023-11-02		First release of this document
1.1	2024-01-24	6. Example Bitstreams	Updated section 6.2
		6. Software Development Enviroment	Chapter added
		7. Example Bitstreams	Added section 7.3 and 7.3
		8. Frequently Asked questions / ...	Chapter added
1.2	2025-01-13	4.2. GR740 License options	Updated text
		4.3.1	Updating serialnumber to MySerialNumberXXA
		4.3.2	Changing text from UART to JTAG
		9. Revision History	Chapter added

10. Support

For support contact the Frontgrade Gaisler support team at support@gaisler.com.

When contacting support, please identify yourself in full, including company affiliation and site name and address. Please identify exactly what product that is used, specifying if it is an IP core (with full name of the library distribution archive file), component, software version, compiler version, operating system version, debug tool version, simulator tool version, board version, etc.

There is also an open forum available at https://grlib.community.

Frontgrade Gaisler AB
Kungsgatan 12
411 19 Göteborg
Sweden
frontgrade.com/gaisler
sales@gaisler.com
T: +46 31 7758650
F: +46 31 421407

Frontgrade Gaisler AB, reserves the right to make changes to any products and services described herein at any time without notice. Consult the company or an authorized sales representative to verify that the information in this document is current before using this product. The company does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by the company; nor does the purchase, lease, or use of a product or service from the company convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual rights of the company or of third parties. All information is provided as is. There is no warranty that it is correct or suitable for any purpose, neither implicit nor explicit.