APPLICATION NOTE

RELEASED 2/18/16

# Remote Boot Over SpaceWire RMAP

**Table 1: Cross Reference of Application Products**

| Product Name | Manufacturer Part Number | SMD# | Device Type | Internal PIC[1] Number: |
|---|---|---|---|---|
| LEON 3FT | UT699 | 5962-08228 | ALL | WG07 |
| | UT699E | 5962-13237 | | WQ02 |
| | UT700 | 5962-13238 | | WQ03 |

PIC=Product Identification Code

## 1.0 Overview

This application note describes the SpaceWire (SpW) Boot bundle. It provides a complete method for remote boot over SpaceWire RMAP (Remote Memory Access Protocol). The SpWBoot bundle supplies source code for loading a simple application to the target device from the master device. The master boot application uploads and starts a mkprom2 boot image on the target device. The target application is extracted from the boot image. Once control is transferred to the target application, the memory area occupied by the boot image becomes available to the target application. The SpaceWire RMAP has the advantage of booting one or more target devices without any further active involvement of the target hardware.

## 2.0 RMAP

The SpaceWire RMAP provides read and write access to a target node directly by the SpaceWire interface, without further involvement of the target hardware. The RMAP capable SpaceWire interfaces included in CAES' LEON processors provide access to the entire addressable AMBA space, including RAM, ROM and APB configuration registers.

### 2.1 Target memory layout

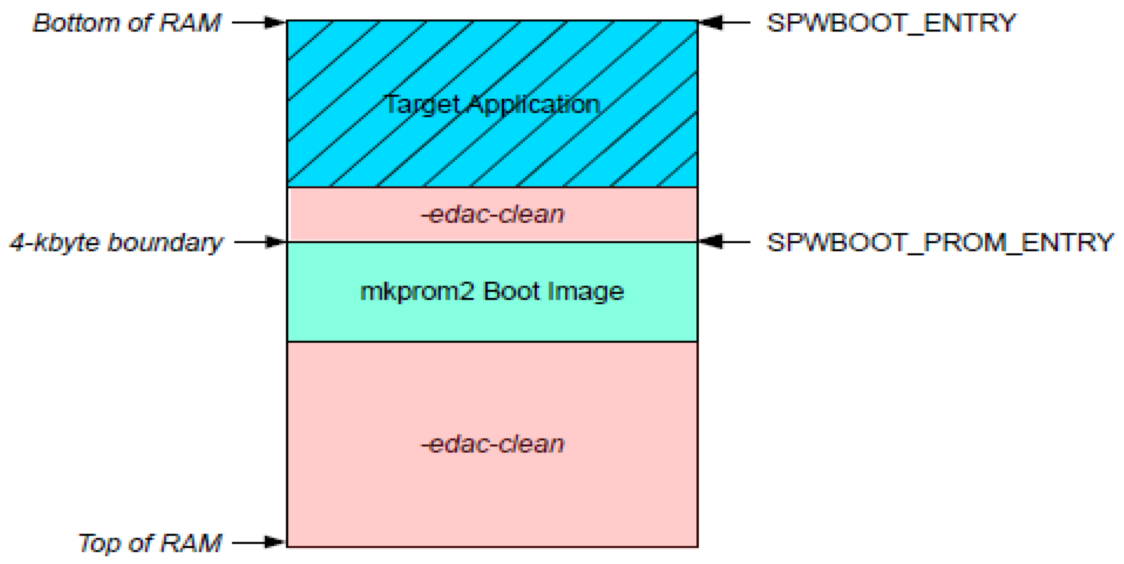


Figure 1: Memory layout of the target RAM

# Remote Boot Over SpaceWire RMAP

The mkprom2 boot image uploaded over SpaceWire is linked to the 4 kbyte boundary following the last allocated address of the target application. In order to safely initialize the whole memory, the user should use mkprom2 –edac-clean switch which clears memory areas below and above the boot image. The compilation process requires two invocations of mkprom2 to get the right PROM image size for edac initialization. It is important to avoid reconfiguration of the memory controller during execution of the boot image from RAM or the memory controller location when set to an arbitrary address by the mkprom2–mctrl switch in the boot image. The target application entry point is chosen as a defined memory address-unused at that stage of the boot process.

**2.2 SpWBoot bundle**

The SpWBoot bundle consists of the following items:

- Spwboot_rmap.h: defined as C style arrays of 32-bit words of address and data for configuring the target's hardware and memory control registers.
- Spwboot_rmap_img.h: Image for the target device defined as C style arrays of 32-bit words of address and data created by the Makefile.
- spwboot.c: The master boot application, which includes the boot image to download on the target platform.
- spwapi.c, spwapi.h: Driver for the GRSPW2 core
- obj2array.c: Tool to convert the boot image to a C style array for inclusion in the master boot application and to extract various address information from ELF object files
- bdinit.c: Initialization routine to be included in the mkprom2 boot process. It initializes the first word in the target memory following the boot image in case the size of the boot image is not 8-byte aligned. This alignment is required for the routine to initialize EDAC memory (mkprom2-edac-clean option)
- Makefile: creates the target mkprom2 image and master boot application.

# Remote Boot Over SpaceWire RMAP

### 2.3 Boot Hardware Setup

- Basic SpaceWire hardware setup

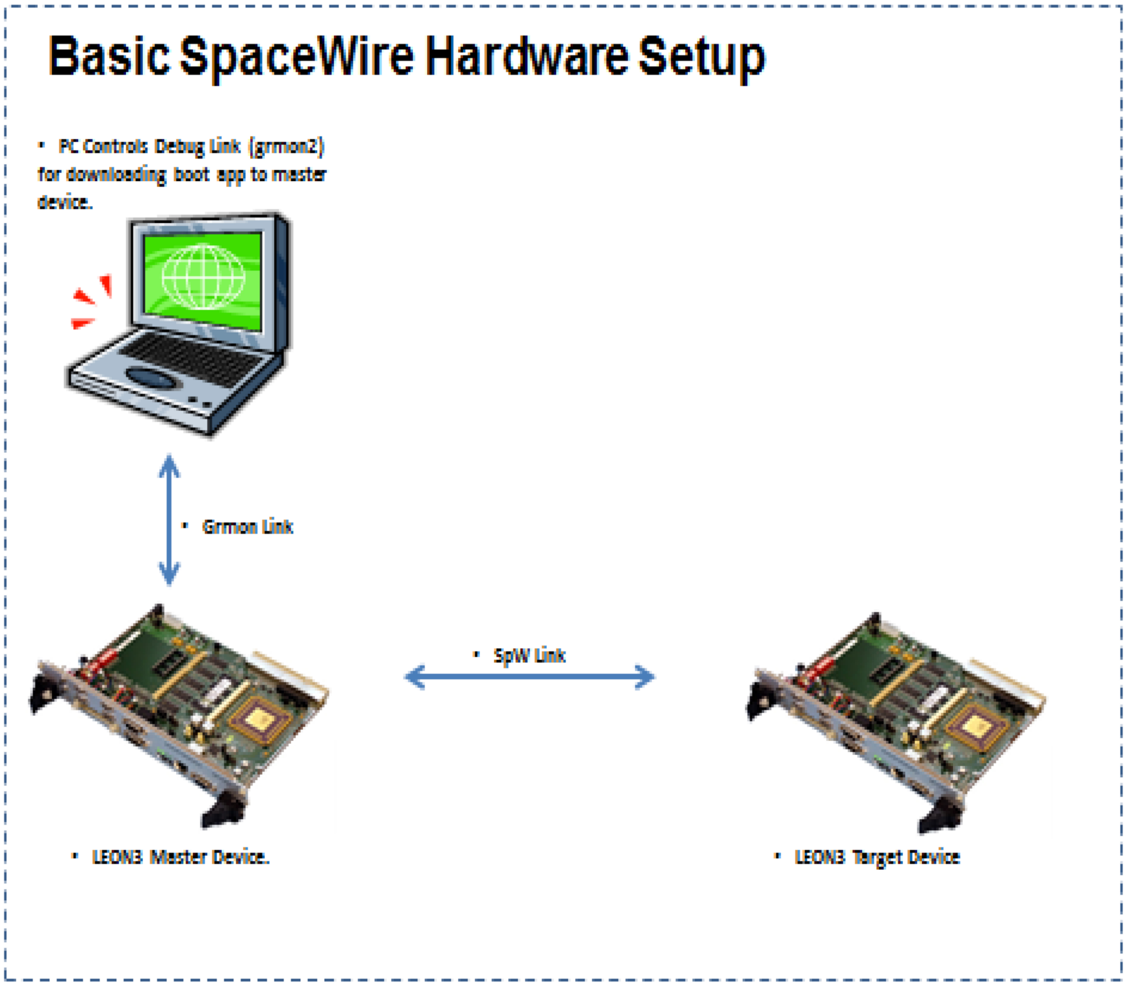

Basic SpaceWire Hardware Setup
• PC Controls Debug Link (grmon2) for downloading boot app to master device.
• Grmon Link
• SpW Link
• LEON3 Master Device.
• LEON3 Target Device

Figure 2: Basic Setup

# Remote Boot Over SpaceWire RMAP

- SpaceWire Router for multiple LEON3 connections

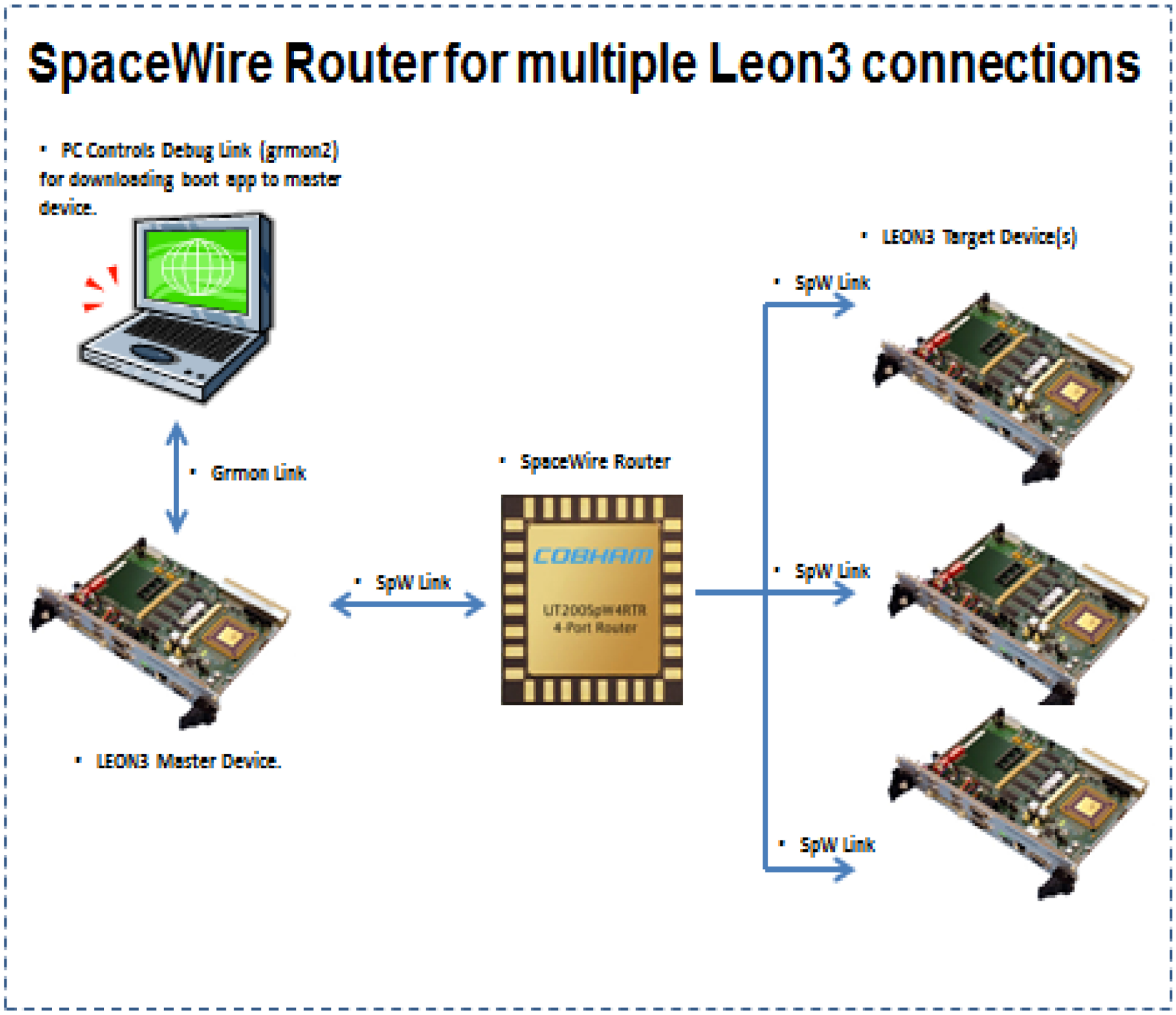

SpaceWire Router for multiple Leon3 connections
PC Controls Debug Link (grmon2) for downloading boot app to master device.
LEON3 Target Device(s)
SpW Link
Grmon Link
SpaceWire Router
COBHAM
UT200SpW4RTR 4-Port Router
SpW Link
SpW Link
SpW Link
LEON3 Master Device.

Figure 3: Multiple LEON3 Setup

## 3.0 Master boot application build

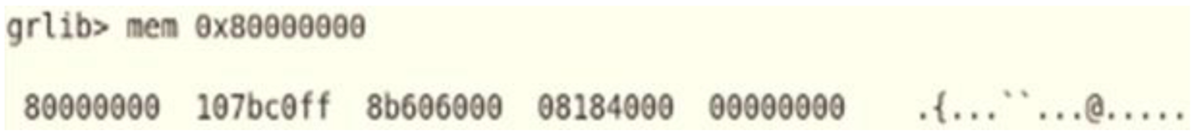
GRMON2, mkprom2 and the cross-toolchain binaries are required to be in the search path to create the master boot application.

# Remote Boot Over SpaceWire RMAP

### 3.1 Master boot application board support configuration

The Makefile configuration reflects target and master platforms for creating the boot application. At least the following adjustments are necessary at a minimum:

- SPWBOOT_TARGET_APP: SpaceWire uploads and starts the application binary file on the target platform
- MKPROM_GEN_ARGS: Generic arguments to mkprom2 for the boot image on the target platform. In particular, -freq and -ramsize are essential, the latter to achieve a correct stack setup.
- RAM_START, RAM_END: Information about the system RAM is necessary in order to provide for a correct initialization of the whole EDAC memory. The -edac-clean option to mkprom2 is used to initialize the whole RAM area before and after the boot image.
- GRSPW_MASTER, GRSPW_TARGET: SpaceWire ports for master and target device.
- spwboot_rmap.h: Configures the target's hardware and memory control registers, figure below shows an example of the memory configuration for SDRAM.



```
grlib> mem 0x80000000

 80000000  107bc0ff  8b606000  08184000  00000000    .{...``...@.....
```

### 3.2 Master boot application compilation

The master boot application has the following steps:

- make spwboot will invoke mkprom2 and create the application binary for the master device.

## 4.0 Boot process

The boot process transfers data over RMAP from a LEON based master platform to the target device, which requires the SpaceWire link clock divisor register to have proper settings. This configuration process requires the target LEON3 to have the DSU enabled.

### 4.1 SpaceWire link

The Clock Divisor Register of the target platform's SpaceWire requires a proper value with respect to the core frequency to achieve a functional SpaceWire link. The clock divisor bits in the Clock Divisor Register are set based on the logic levels of GPIO [7:4] on power-up, see Figure 4. They may be set by software after the boot process. Please refer to the target user manual for details.

# Remote Boot Over SpaceWire RMAP

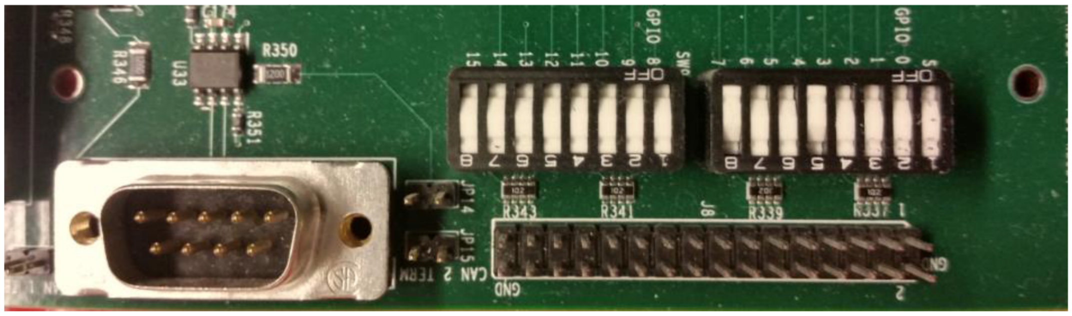
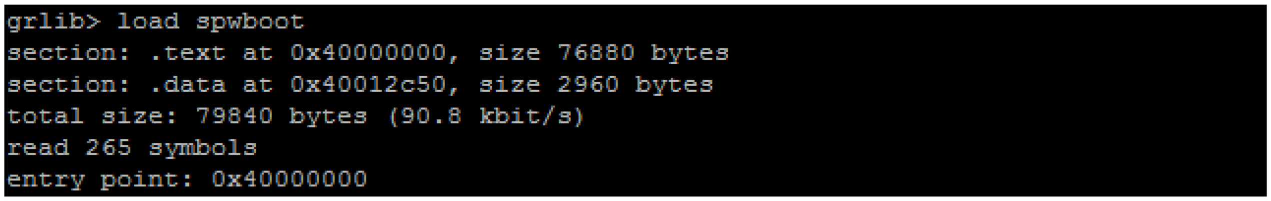


Figure 4: GRSPW2 link divider register value from GPIO[7:4] on UT700 board

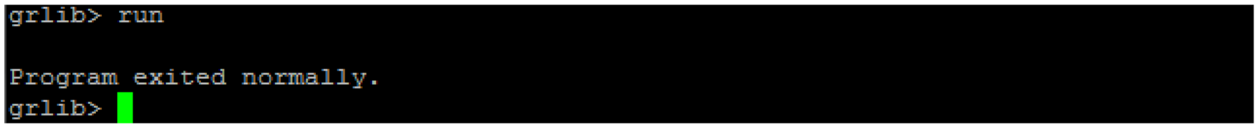
### 4.2 Load SpWBoot (boot application) on master device

Change < –uart /dev/ttyUSB2 > to proper serial device
$ grmon –uart /dev/ttyUSB2 –nosram
Load spwboot on master device.



```
grlib> load spwboot
section: .text at 0x40000000, size 76880 bytes
section: .data at 0x40012c50, size 2960 bytes
total size: 79840 bytes (90.8 kbit/s)
read 265 symbols
entry point: 0x40000000
```

Run the application.



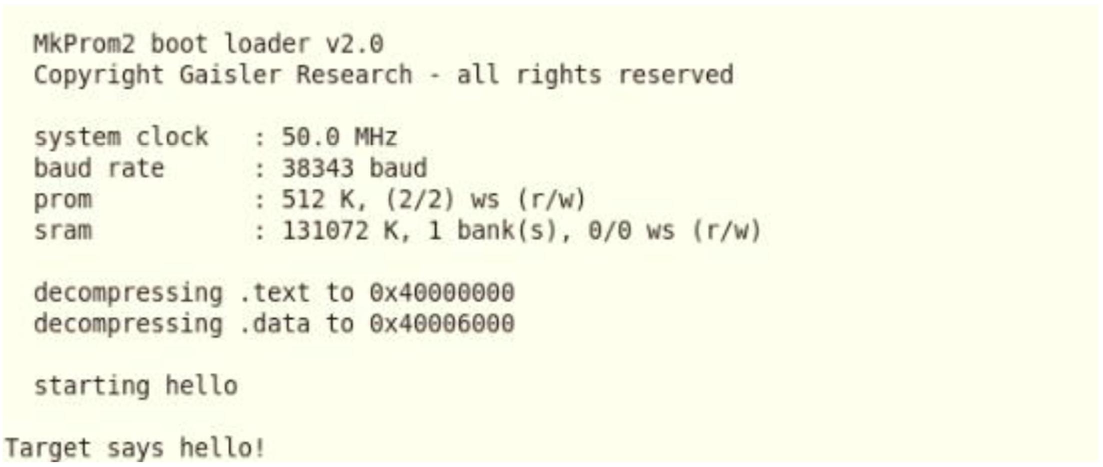
```
grlib> run

Program exited normally.
grlib>
```

### 5.0 Booting the target device with demo application

The terminal window displays the message in Figure 5 with the target device booting to mkprom2 and the 'hello world' application.

# Remote Boot Over SpaceWire RMAP



```
  MkProm2 boot loader v2.0
  Copyright Gaisler Research - all rights reserved

  system clock   : 50.0 MHz
  baud rate      : 38343 baud
  prom           : 512 K, (2/2) ws (r/w)
  sram           : 131072 K, 1 bank(s), 0/0 ws (r/w)

  decompressing .text to 0x40000000
  decompressing .data to 0x40006000

  starting hello

Target says hello!
```

Figure 5: Message from target device

## 6.0 Reference

Datasheet: cobhamaes.com
Application Note Info: cobhamaes.com
Application Note : cobhamaes.com
Source Code: cobhamaes.com
Debug Interfaces: cobhamaes.com

### *Revision History*

| Date | Rev. # | Change Description |
|---|---|---|
| 02/18/2016 | 4 | CAES' suggested references |

APPLICATION NOTE

RELEASED 2/18/16

# Remote Boot Over SpaceWire RMAP