

VXWORKS 6.9 ON ALTERA CYCLONE V AND
ZEDBOARD. LINUX DEBUG ON ALTERA CYCLONE V



MATTIA CIOLLI
MATTEO POLSINELLI
DAVIDE GIANCOLA

INDICE

1	INTRODUCTION	1
2	ALTERA CYCLONE V VXWORKS BOOT	2
2.1	Overview	2
2.2	Preloader	7
2.3	Bootloader	8
2.4	From scratch	9
2.4.1	Compiling the Hardware Design	9
2.4.2	Compiling the Preloader	9
2.4.3	Device Tree	10
2.4.4	Vivado	10
3	ZEDBOARD	12
3.1	Overview	12
3.2	Building the kernel image	13
3.3	Generating the bootloader	13
3.3.1	Bootloader	13
3.3.2	Bootloader configuration	14
3.3.3	Errors and bugs	14
3.4	XMD console	14
3.4.1	Connected to the board	14
3.4.2	RAM busy	15
3.4.3	Missing component	15
3.4.4	Directly in RAM	16
4	ANOTHER ZEDBOARD	17
4.1	Wrong kernel image	17
4.2	U-BOOT	17
5	LINUX ON CYCLONE V	19
5.1	Debug on Linux	19
6	CONCLUSIONS	25
I	APPENDIX	26
A	ONLINE QUESTION	27
A.1	VxWorks question	27
A.2	BlaBla question	27
	BIBLIOGRAFIA	28

ELENCO DELLE FIGURE

Figura 1	Boot flow	2
Figura 2	Additional boot flow	2
Figura 3	Altera Cyclon V	3
Figura 4	Altera Cyclon V switchs	3
Figura 5	MSEL	4
Figura 6	Shown with Putty application	5
Figura 7	Msel switchs	5
Figura 8	socfpga folder	6
Figura 9	Boot flow	7
Figura 10	Boot flow	8
Figura 11	Boot cyclonV guide	8
Figura 12	From scratch	9
Figura 13	GHRD	9
Figura 14	Quartus Error	10
Figura 15	Compiling schema	10
Figura 16	Device tree	10
Figura 17	Device tree error	11
Figura 18	Zedboard	12
Figura 19	Bootloader build	13
Figura 20	Boot	13
Figura 21	Code	14
Figura 22	XMD connect	15
Figura 23	Error	17
Figura 24	Another error	17
Figura 25	Wrong kernel image	18
Figura 26	Boot problem	18
Figura 27	Debug terminal	19
Figura 28	Router connection	20
Figura 29	Internet properties	21
Figura 30	Connection test	21
Figura 31	Commands	22
Figura 32	License error	22
Figura 33	Download on borad	23
Figura 34	Download success	23
Figura 35	Settings	24
Figura 36	Debug	24

INTRODUCTION

La mobilità delle persone e delle merci sono una componente essenziale del mercato interno dell'Unione Europea (UE) ed è di fondamentale importanza garantire la sua fattibilità al fine di salvaguardare la crescita economica. La rete ferroviaria ha un ruolo strategico in questo contesto almeno sotto due punti di vista:

ALTERA CYCLONE V VXWORKS BOOT

2.1 OVERVIEW

The following figure depicts the typical boot flow:

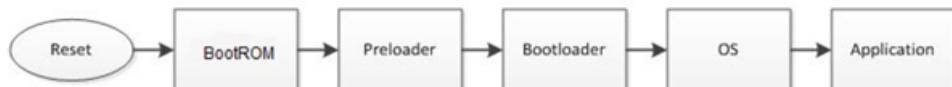


Figura 1: Boot flow

Additional boot flows are possible, as shown in the following diagram:

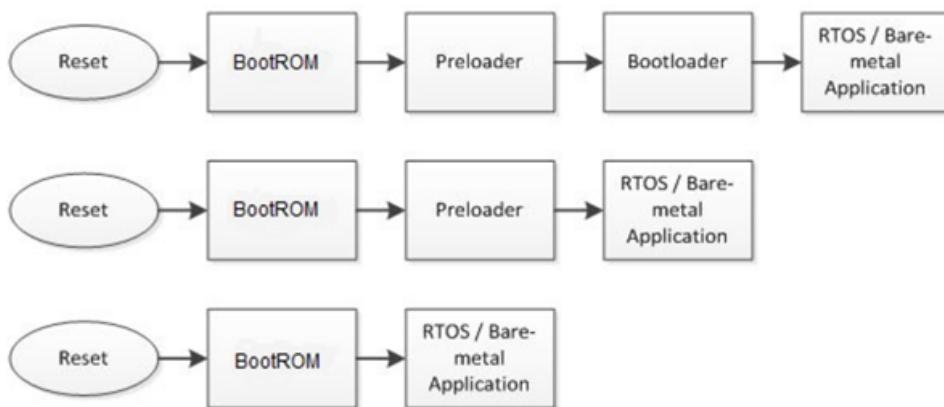


Figura 2: Additional boot flow

The HPS boot process starts when the processor is released from reset, and jumps to the reset vector address, located in the Boot ROM address space.

Typically, the main responsibilities of the Boot ROM are:

- Detect the selected boot source;
- Perform minimal HPS initialization;
- Load the next boot stage (typically the Preloader) from Flash to OCRAM and jump to it.

The behavior of the Boot ROM is influenced by the BSEL and CSEL options (rev B, C). Typical board switches layout found in the references.

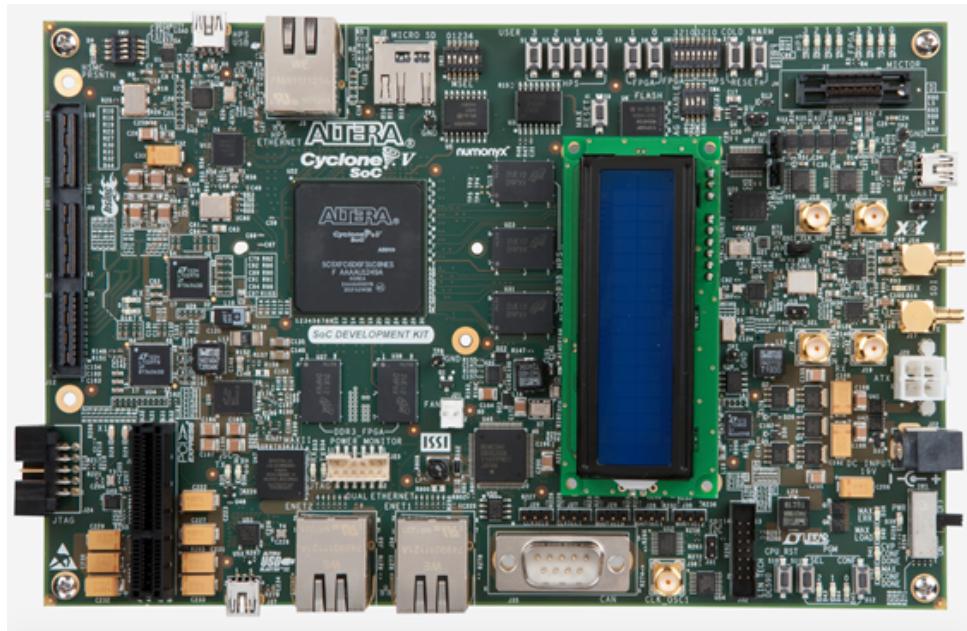


Figura 3: Altera Cyclon V

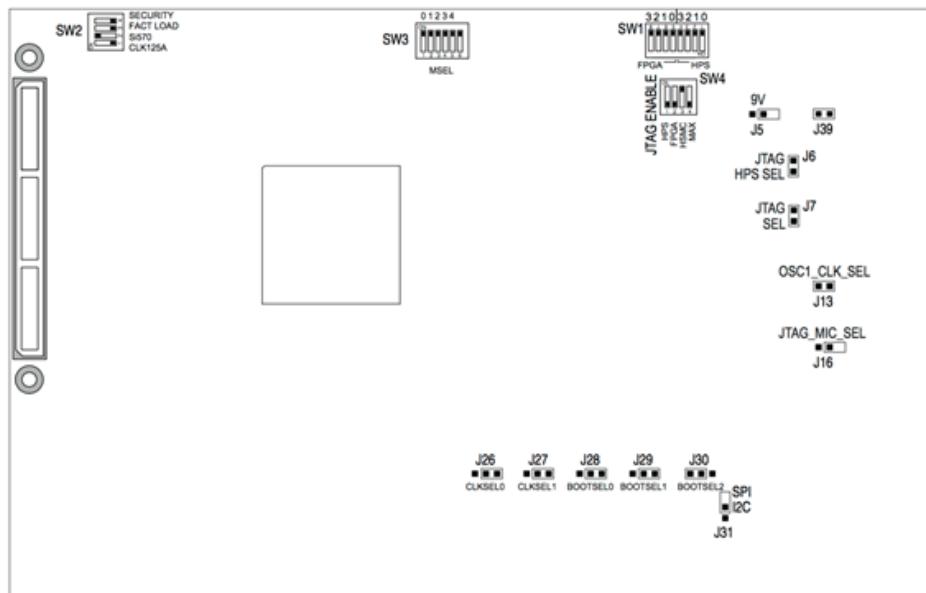


Figura 4: Altera Cyclon V switchs

Our board. The switches are configured for FPGA Working Mode.

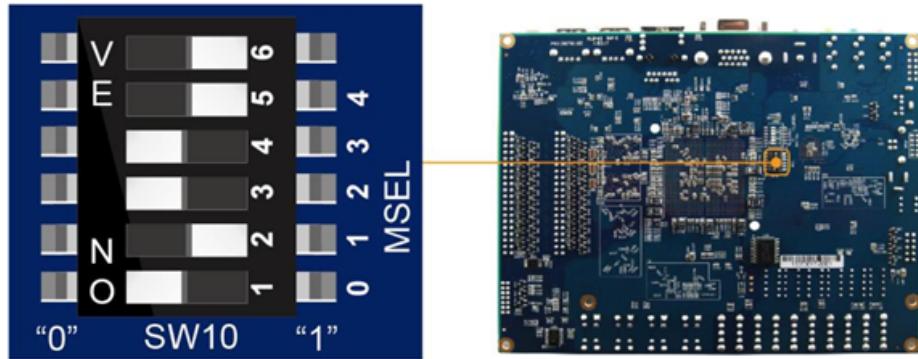


Figura 5: MSEL

Running Linux to check the correct configuration of the board switches.

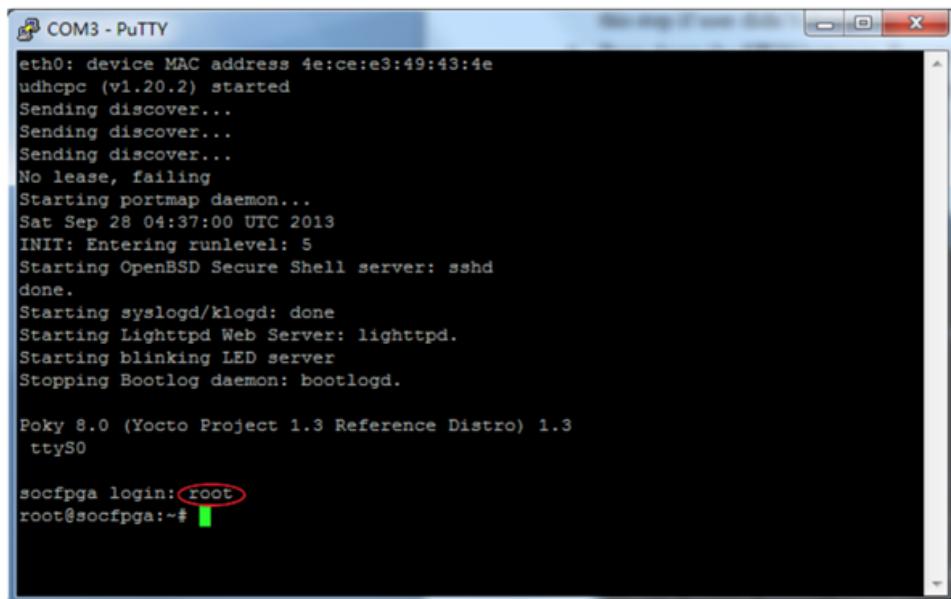
- **What is necessary:**

1. SD card (At least 4 gb);
2. Win32DiskImager.exe (<http://sourceforge.net/projects/win32diskimager/> to flash the Linux Image on the SD);
3. Pre-built SD Card Image (http://www.terasic.com/downloads/cd-rom/de1-soc/linux_BSP/DE1_SoC_SD.zip.);
4. Putty.

- **What is inside the default image:**

1. SPL Pre-loader;
2. U-boot;
3. Device Tree Blob;
4. Linux Kernel;
5. Linux Root File system.

- **MSEL CONFIGURATION**



A screenshot of a Windows-style terminal window titled "COM3 - PuTTY". The window displays a black background with white text representing a system boot log. The log includes messages such as "eth0: device MAC address 4e:ce:e3:49:43:4e", "udhcpc (v1.20.2) started", "Sending discover...", "No lease, failing", "Starting portmap daemon...", "Sat Sep 28 04:37:00 UTC 2013", "INIT: Entering runlevel: 5", "Starting OpenBSD Secure Shell server: sshd done.", "Starting syslogd/klogd: done", "Starting Lighttpd Web Server: lighttpd.", "Starting blinking LED server", "Stopping Bootlog daemon: bootlogd.", "Poky 8.0 (Yocto Project 1.3 Reference Distro) 1.3", "ttyS0", and "socfpga login: root". The "root" prompt is highlighted with a red oval.

Figura 6: Shown with Putty application

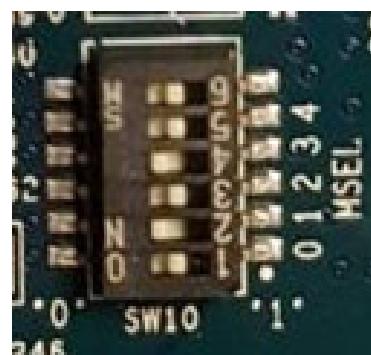


Figura 7: Msel switchs

The preloader configures clocking, IOCSR, pinmuxing, DDRAM and loads the second-stage bootloader(like U-boot or in our case the VxWorks bootloader) into DDRAM.

Folder	File	Generated	Description
board/altera/socfpga/	build.h	yes	Preloader Build Parameters
	iocsr_config_arria5.c	yes	I/O Pin Configuration Blob (Arria V)
	iocsr_config_arria5.h	yes	
	iocsr_config_cyclone5.c	yes	I/O Pin Configuration Blob (Cyclone V)
	iocsr_config_cyclone5.h	yes	
	pinmux_config.h	yes	Pin Muxing Parameters Header
	pinmux_config_arria5.c	yes	Pin Muxing Parameters (Arria V)
	pinmux_config_cyclone5.c	yes	Pin Muxing Parameters (Cyclone V)
	pll_config.h	yes	Clock Configuration
	reset_config.h	yes	Reset Configuration
	Makefile		Makefile
	socfpga_common.c		Common Functions
	socfpga_arria5.c		Displays Board Name (Arria V)
	socfpga_cyclone5.c		Displays Board Name (Cyclone V)
	timestamp_config.h		Timestamp File

Figura 8: socfpga folder

2.2 PRELOADER

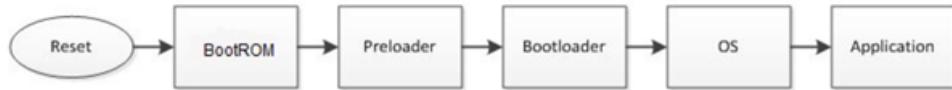


Figura 9: Boot flow

The modifications made to the uboot-socfpga/board/altera/socfpga/build.h file are shown below in bolded text:

```
1 /* Enable FAT partition support when booting from SDMMC. */
2 #define CONFIG_PRELOADER_FAT_SUPPORT (1)
3
4 /* When FAT partition support is enabled, this specifies the * FAT partition where the boot image is located.
   */
5 #define CONFIG_PRELOADER_FAT_BOOT_PARTITION (1)
6
7 /* When FAT partition supported is enabled, this specifies the * boot image filename within a FAT partition to
   be used as fatload payload.*/
8 #define CONFIG_PRELOADER_FAT_LOAD_PAYLOAD_NAME "bootloader.bin"
```

Another change where the preloader loads the FPGA file was made to the uboot-socfpga/include/configs/ socfpga_common.h file:

```
1 /* FPGA programming support with SPL FPGA RBF file source (with mkimage header) is located within the same
   boot device which stored the subsequent boot image (U-Boot). */
2
3 /* enabled program the FPGA */
4 #define CONFIG_SPL_FPGA_LOAD
```

2.3 BOOTLOADER

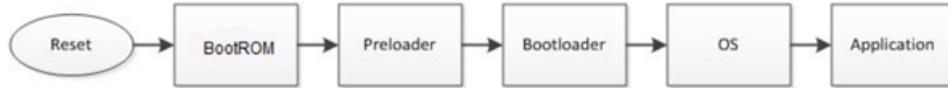


Figura 10: Boot flow

We have built the VxWorks bootrom file using Wind River tools and the alt_soc_gen5 BSP using the workbench of Wind River.

```
1 -A arm -T firmware -C none -O vxworks -a 0x08000040 -e 0 -n "vxWorks bootloader for SoC FPGA" -d bootrom.bin bootloader.bin
```

Finally we have put everything on the SD Card and we had tried to boot the system with no results.

This method is described by the official documentation from the Intel/Altera site. Also the software is provided.

The screenshot shows a technical document titled "Booting VxWorks with the Altera Cyclone V and Arria V Development Boards". The page is dated 2014.10.17 and includes links for "AN-724", "Subscribe", and "Send Feedback".

Purpose of this Document

This document describes how to take a VxWorks® bootrom that is built using the Wind River® BSP, alt_soc_gen5, combine it with a preloader built from an Altera® FPGA design, and boot the Altera Cyclone® V and Arria® V SoC development boards using QSPI or SD/MMC.

Note: This document does not cover how to configure or build the Altera preloader or VxWorks bootrom.

What is Needed to Build this System

The following is needed to build this system:

- An Altera Cyclone V or Arria V SoC development board to run the software.
- A Wind River workbench development environment and license to build the VxWorks bootrom.
- An SD/MMC card to hold the boot software for SD/MMC or help program QSPI. Alternatively, you can use the Altera Quartus programmer to program QSPI instead of programming through the SD/MMC.
- The Win32 Disk Imager program to write the image to an SD/MMC card. You can also use the Linux program dd, but be careful with the device name. You can download the Windows tool from the SourceForge® website.

This package contains pre-built binaries to get started, including:

- Prefbuilt SD/MMC images for Cyclone V and Arria V development boards.
- Prefbuilt preloader images for both QSPI and SD/MMC and for both Cyclone V and Arria V development boards.
- Prefbuilt version of the linux utility mkimage for Windows, used to wrap the VxWorks bootrom with the proper header.

This release also contains a script, make_sdimage.sh, to help create a new SD/MMC image on a linux system.

© 2014 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRON, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/legal.html. Altera warrants performance of its products to the specifications for a limited time and limited applications as described in the standard warranty. Altera reserves the right to make changes to any products or services at any time without notice, but is not obligated to notify a customer of any such changes or of any inference product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

101 Innovation Drive, San Jose, CA 95134

ALTERA

Figura 11: Boot cyclonV guide

2.4 FROM SCRATCH

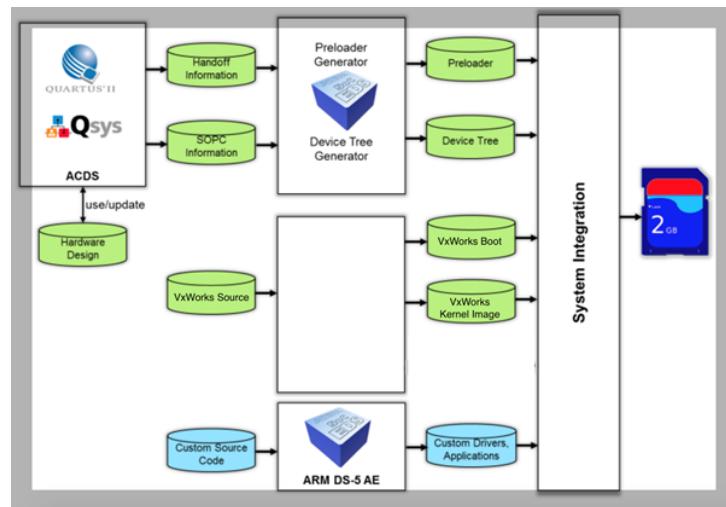


Figura 12: From scratch

2.4.1 Compiling the Hardware Design

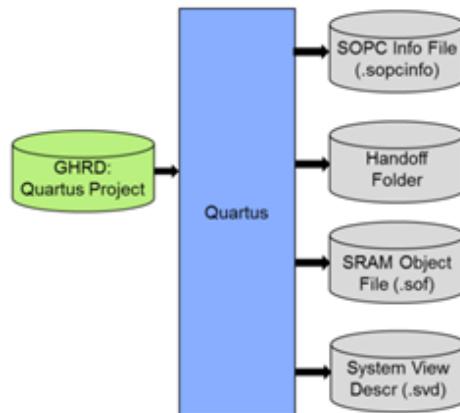


Figura 13: GHRD

<https://rocketboards.org/foswiki/view/Documentation/GSRDCompileHardwareDesign>.

2.4.2 Compiling the Preloader

<https://rocketboards.org/foswiki/view/Documentation/GSRDPreloader>.

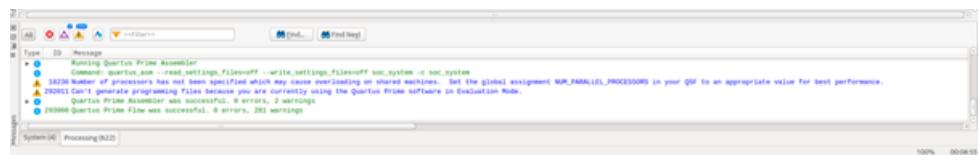


Figura 14: Quartus Error

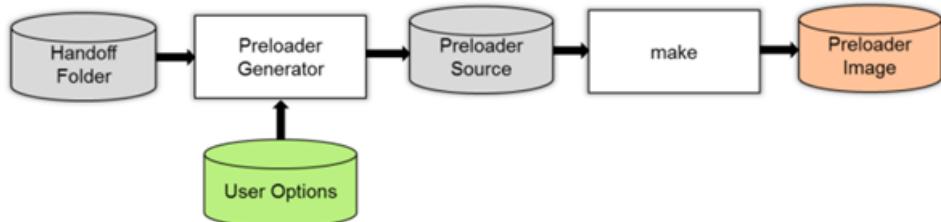


Figura 15: Compiling schema

2.4.3 Device Tree

The device tree is a data structure for describing hardware. Given the correct device tree, the same compiled kernel can support different hardware configurations within a wider architecture family.

For ARM, use of device trees has become mandatory for all new SoCs.

This can be seen as a remedy to the vast number of forks (of Linux and Das U-boot) that has historically been created to support (marginally) different ARM boards.

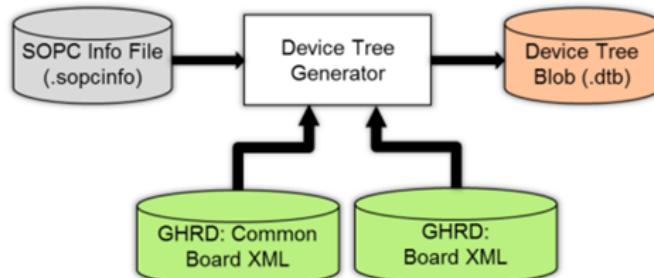


Figura 16: Device tree

<https://rocketboards.org/foswiki/view/Documentation/GSRDDeviceTreeGenerator>.

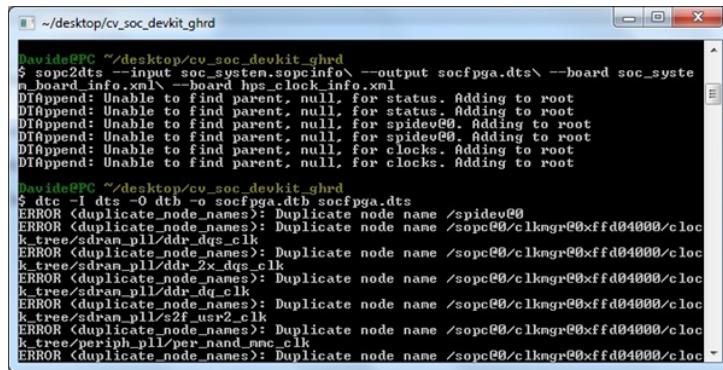
2.4.4 Vivado

We used Vivado on the workstation to build We built the VxWorks image for the Cyclone V. Building the device tree it gave an error, so we forced the output.

We put on the SD the image, the device tree and the bootloader.

We tried to boot but nothing showed up, neither with the forced device tree, nor with

the .dts file already provided.



The screenshot shows a terminal window with the following command and its output:

```
David@PC:~/desktop/cv_soc_devkit_ghrd$ socp2dts --input soc_system.socpinfo --output socfpga.dts --board soc_system_board_info.xml --board hps_clock_info.xml
DIFAppend: Unable to find parent, null, for status. Adding to root
DIFAppend: Unable to find parent, null, for status. Adding to root
DIFAppend: Unable to find parent, null, for spidev@0. Adding to root
DIFAppend: Unable to find parent, null, for spidev@0. Adding to root
DIFAppend: Unable to find parent, null, for spidev@0. Adding to root
DIFAppend: Unable to find parent, null, for clocks. Adding to root
DIFAppend: Unable to find parent, null, for clocks. Adding to root

David@PC:~/desktop/cv_soc_devkit_ghrd$ dtc -I dts -O dtb -o socfpga.dtb socfpga.dts
ERROR <duplicate_node_names>: Duplicate node name /spidev@0
ERROR <duplicate_node_names>: Duplicate node name /soc@0/clkmgr@0xffff04000/clk_k_tree/sdram_pll/addr_dqs_clk
ERROR <duplicate_node_names>: Duplicate node name /soc@0/clkmgr@0xffff04000/clk_k_tree/sdram_pll/addr_2x_dqs_clk
ERROR <duplicate_node_names>: Duplicate node name /soc@0/clkmgr@0xffff04000/clk_k_tree/sdram_pll/addr_dq_clk
ERROR <duplicate_node_names>: Duplicate node name /soc@0/clkmgr@0xffff04000/clk_k_tree/clkctrl_nand_ns2_clk
ERROR <duplicate_node_names>: Duplicate node name /soc@0/clkmgr@0xffff04000/clk_k_tree/periph pll/per_nand_mmc_clk
ERROR <duplicate_node_names>: Duplicate node name /soc@0/clkmgr@0xffff04000/clk_k_tree/periph pll/per_nand_mmc_clk
ERROR <duplicate_node_names>: Duplicate node name /soc@0/clkmgr@0xffff04000/clk_k_tree/periph pll/per_nand_mmc_clk
```

Figura 17: Device tree error

ZEDBOARD

After having failed to boot VxWorks on the Cyclone V and waiting for Quartus licence, we decided to try to boot the same RTOS on the Zedboard, a Zynq-7000 based board. This board was chosen because we have seen that it has official documentation and guides and it's one of the boards supported by VxWorks.

Moreover there are programs like Vivado to develop and debug software.

3.1 OVERVIEW

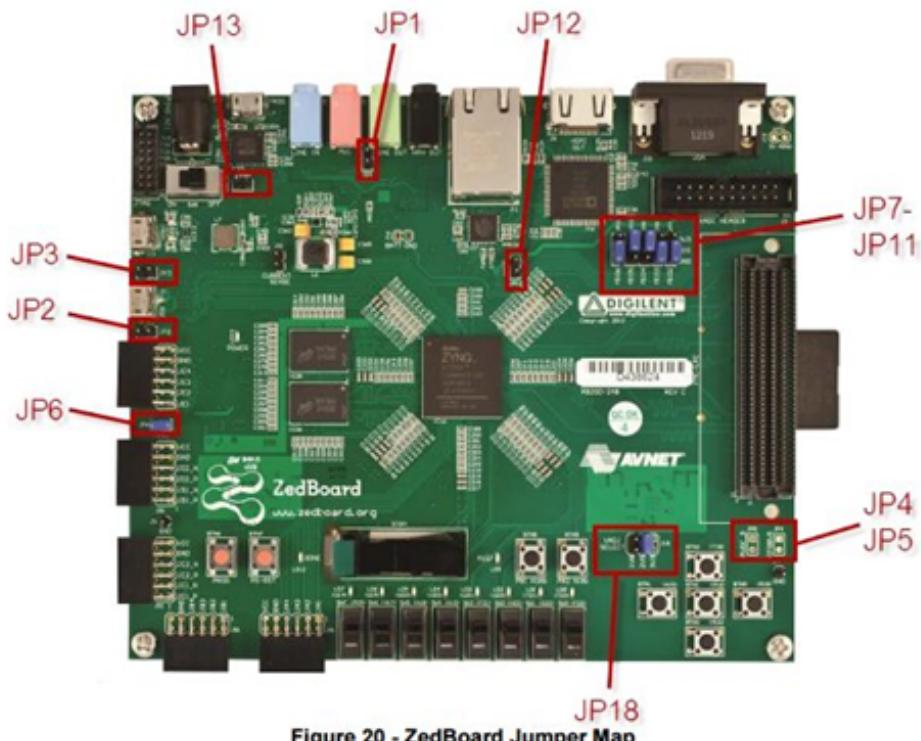


Figure 20 - ZedBoard Jumper Map

Figura 18: Zedboard

- SD Card;
- 512 MB DDR3 (128M x 32) o 256 Mb QSPI Flash;
- USB 2.0 FS USB-UART bridge;
- Dual ARM Cortex-A9 MPCore Up to 667 MHz.

3.2 BUILDING THE KERNEL IMAGE

We have built the kernel image with Windriver workbench following this guide:
[Guide](#)

We created the .bif file with the following format:

```
1 ZC702_bif_for_VxWorks:  
2 {  
3 [bootloader]zynq_fsbl_0.elf  
4 bootROM.elf  
5 }
```

And downloaded the .elf file from the official Xilinx website.

3.3 GENERATING THE BOOTLOADER

We built the bootloader using the Vivado TCL shell using “bootgen” following this guide <http://www.wiki.xilinx.com/Prepare+boot+image> omitting the “I” parameter because was not recognised.

```
Vivado% bootgen -image boot.bif -o boot.bin  
WARNING: [Common 17-259] Unknown Tcl command 'bootgen -image boot.bif -o boot.bin' sending command to the OS shell for execution.  
Vivado%
```

Figura 19: Bootloader build

3.3.1 Bootloader

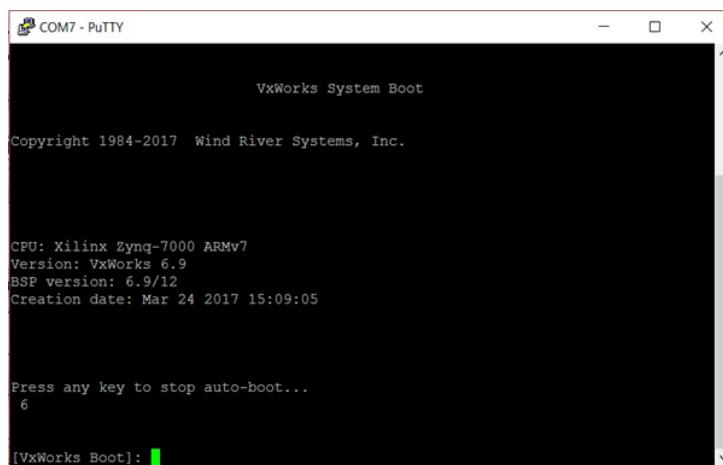


Figura 20: Boot

We installed the Cypress drivers for the serial interface.
We copied boot.bin e vxWorks on the SD card and connected with Putty.
The bootloader started up.

3.3.2 Bootloader configuration

We stopped the autoboot, then we inserted the following instructions as suggested from the guide:

Guide

Type **C** at the boot prompt, and press **Return** to start the boot configuration.
Change the boot device to **fS** and press **Return** until you reach the file name.
Change to **/sd0:1/vxWorks** and press **Return** until you reach **other (o)**.
If no entry exists, type **gem0**. Press **Return** and the boot prompt opens.
Type **@** to proceed the boot process.
Type **i** to display all running tasks.

Figura 21: Code

3.3.3 Errors and bugs

After some attempts it gave us the following error

```
1 Host Name: bootHost
2 Target Name: vxTarget
3 User: target
4 ERROR: ipcom_drv_eth_init: drvname:, drvunit: 0
5 0x1100e5c (tRootTask): task 0x1100e5c has had a failure and has been stopped.
6 0x1100e5c (tRootTask): fatal exception in a kernel task or stack overflow!
7 Instantiating /sd0:0 as rawFs, device = 0x10001
```

We noticed that the parameters inserted previously are not stored in the SD but in internal memory. So resetting, rebooting, formatting the SD and installing Linux had no effect in deleting those values.

3.4 XMD CONSOLE

We used the Xilinx SDK (Software development kit), a software installed with Vivado, used to develop embedded applications.

We tried to put the bootloader and the image directly in RAM with the XMD console included in the SDK.

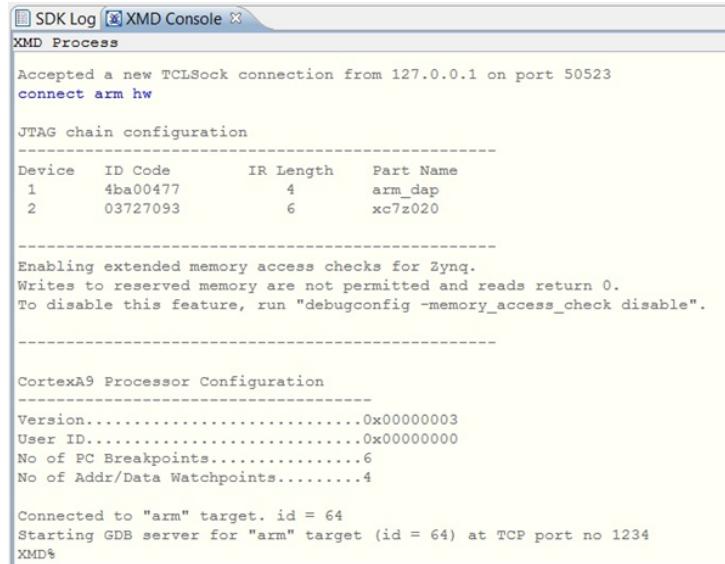
Trying to connect with the connect command we have discovered that 2 usb cable were needed: the first for the serial, the second connected to the PROG port.

3.4.1 Connected to the board

Following this guide https://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug1043-embedded-system-tools.pdf pg.73, we inserted the command

“connect arm hwCortexA9” without result.

Following the help of the console we discovered that the right command is “connect arm hw”.



The screenshot shows the XMD Console window with the following output:

```
SDK Log XMD Console X
XMD Process
Accepted a new TCLSock connection from 127.0.0.1 on port 50523
connect arm hw

JTAG chain configuration
-----
Device ID Code IR Length Part Name
1 4ba00477 4 arm_dap
2 03727093 6 xc7z020
-----
Enabling extended memory access checks for Zynq.
Writes to reserved memory are not permitted and reads return 0.
To disable this feature, run "debugconfig -memory_access_check disable".
-----
CortexA9 Processor Configuration
-----
Version..... 0x00000003
User ID..... 0x00000000
No of PC Breakpoints..... 6
No of Addr/Data Watchpoints..... 4

Connected to "arm" target. id = 64
Starting GDB server for "arm" target (id = 64) at TCP port no 1234
XMD%
```

Figura 22: XMD connect

3.4.2 RAM busy

Once connected we tried to put the files directly in RAM. we executed the command “dow -data boot.bin ox00100000” giving us the following error:

```
1 AP transaction error (DP CTRL_STAT=0xf0000021)
2 Error Address = 0x00100000, Size = 0x00000004
```

We discovered that the problem wasn't the register but the fact that the processor was not stopped and the memory was busy

3.4.3 Missing component

We tried to manually stop and reset the processor with “stop” and “rst” commands. After some researches we discovered that ps7_init.tcl was missing. We downloaded it and we used it to stop the processor:

```
1 source ps7_init.tcl
2 ps7_init
3 ps7_post_config
4 stop
```

3.4.4 Directly in RAM

We were able to stop the processor and to put the files in RAM with the “dow” command. Then we gave the “con” and “run” command but nothing showed up in Putty.

```
1 XMD% con  
2  
3 RUNNING> XMD%
```

After this we asked help on the knowledge forum to resolve the stackoverflow problem but no one was able to help us <https://ask.windriver.com/en/questions/35789/zedboard-boot-vxworks-69-problem/>.

ANOTHER ZEDBOARD

We changed board and the bootloader started well without giving the stackoverflow error but randomly didn't start.

After having tried the same parameters previously inserted, the bootloader couldn't initialize inet and some network settings.



Figura 23: Error

```
"inet on ethernet" boot parameter not set.  
ERROR: ipcom_drv_eth_init: drvname:, drvunit: 0
```

Figura 24: Another error

4.1 WRONG KERNEL IMAGE

Following this guide [Guide](#) we tried to boot the vxWorks image but it can't get the kernel image.

4.2 U-BOOT

We rebuilded the image following the configurations parameters provided by the guide but we had the same error.

```

Zynq> fatls mmc 0
      system volume information/
 586244  boot.bin
 14671  devicetree.dtb
 205888  fsbl-zc702-zynq7.elf
3006660  u-boot.elf
3827288  uimage
8487331  uramdisk.image.gz
 4096  .trashes
       .trashes/
       .spotlight-v100/
 3675048  vxworks

8 file(s), 3 dir(s)

Zynq> fatload 0x5000000 vxworks
** Bad device specification 0x5000000 vxworks **
Zynq> fatload mmc 0 0x5000000 vxworks
reading vxworks
3675048 bytes read in 325 ms (10.8 MiB/s)
Zynq> fatload mmc 0 0x4000000 devicetree.dtb
reading devicetree.dtb
14671 bytes read in 19 ms (753.9 KiB/s)
Zynq> bootm 0x5000000 - 0x4000000
Wrong Image Format for bootm command
ERROR: can't get kernel image!
Zynq>

```

Figura 25: Wrong kernel image

```

U-Boot 2016.07 (Dec 12 2016 - 23:04:43 -0700)

Model: Zynq ZC702 Development Board
Board: Xilinx Zynq
I2C: ready
DRAM: ECC disabled 1 GiB
MMC: sdhci@e010000: 0
SF: Detected S25FL256S_64K with page size 256 Bytes, erase size 64 KiB, total 32
    MiB
In:   serial@e0001000
Out:  serial@e0001000
Err:  serial@e0001000
Model: Zynq ZC702 Development Board
Board: Xilinx Zynq
Net:  ZYNQ GEM: e000b000, phyaddr 7, interface rgmii-id
eth0: ethernet@e000b000
Hit any key to stop autoboot:  0
Zynq>
Zynq>

```

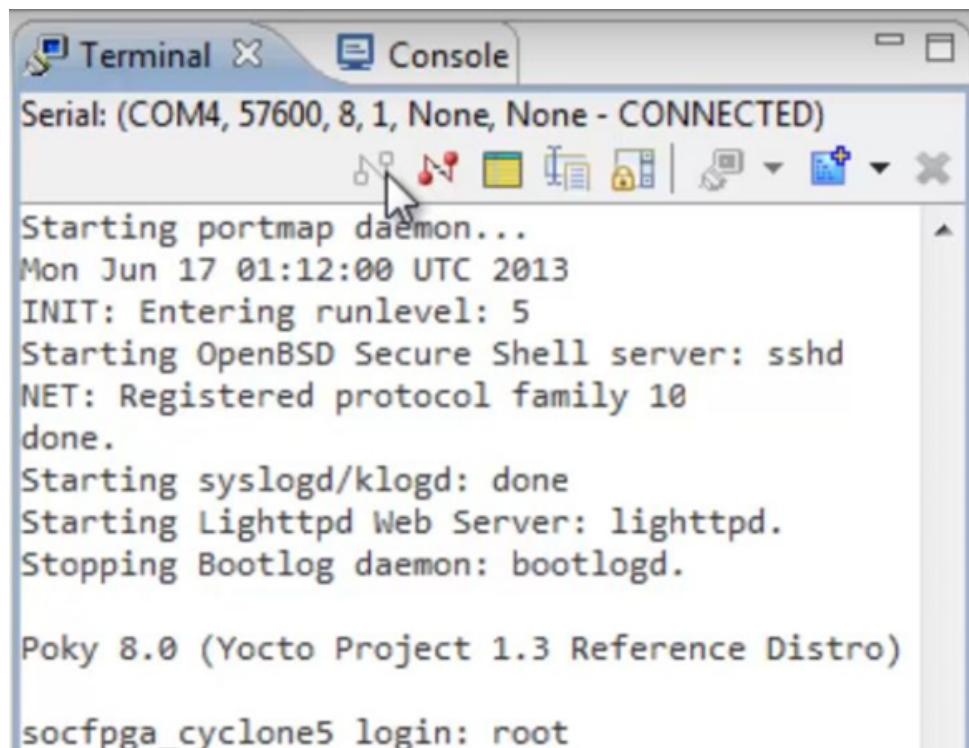
Figura 26: Boot problem

LINUX ON CYCLONE V

We have downloaded the system image from the Altera website and put it on the SD card with Win32DiskImager.exe.

We connected with Putty and it showed that Linux booted up successfully.

5.1 DEBUG ON LINUX



```
Terminal X Console
Serial: (COM4, 57600, 8, 1, None, None - CONNECTED)
Starting portmap daemon...
Mon Jun 17 01:12:00 UTC 2013
INIT: Entering runlevel: 5
Starting OpenBSD Secure Shell server: sshd
NET: Registered protocol family 10
done.
Starting syslogd/klogd: done
Starting Lighttpd Web Server: lighttpd.
Stopping Bootlog daemon: bootlogd.

Poky 8.0 (Yocto Project 1.3 Reference Distro)

socfpga_cyclone5 login: root
```

Figura 27: Debug terminal

To debug on Linux with the Cyclone V was needed an Eclipse based environment called DS-5 debugger included in the intelFPGA software.

We connected it to the serial and the boot was showed in its terminal panel.

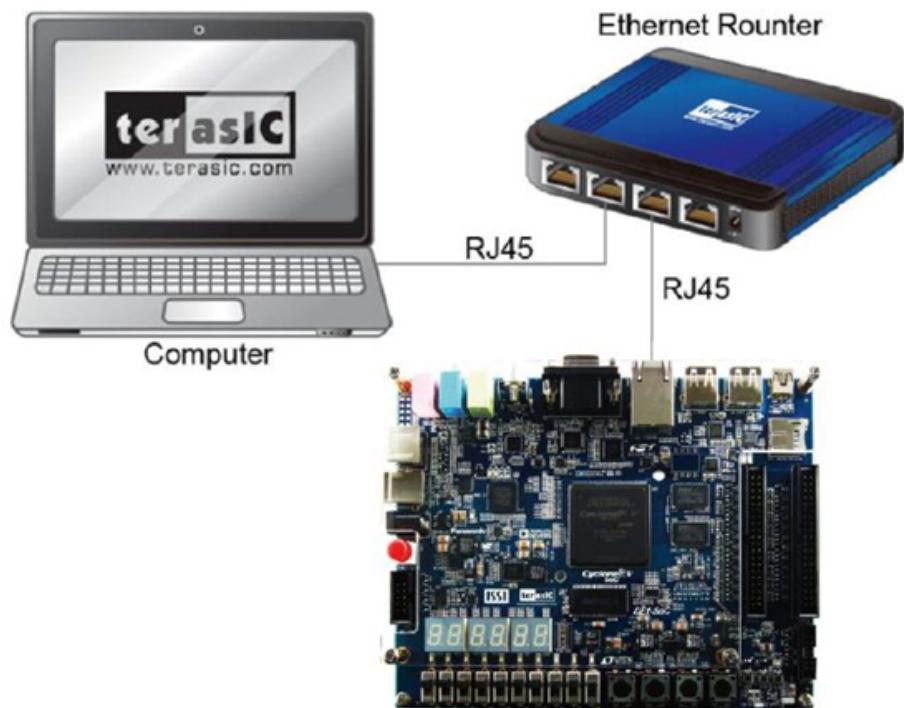


Figura 28: Router connection

As told in "My First HPS guide provided by Altera, a router was needed to connect the PC with the board via Ethernet.

However the router was only needed for its DHCP protocol, so we manually configured the board with ifconfig etho 192.168.1.4 netmask 255.255.255.0 up in the terminal. In Windows in LAN settings we configured the network as in the image. We executed the ping command and it worked.

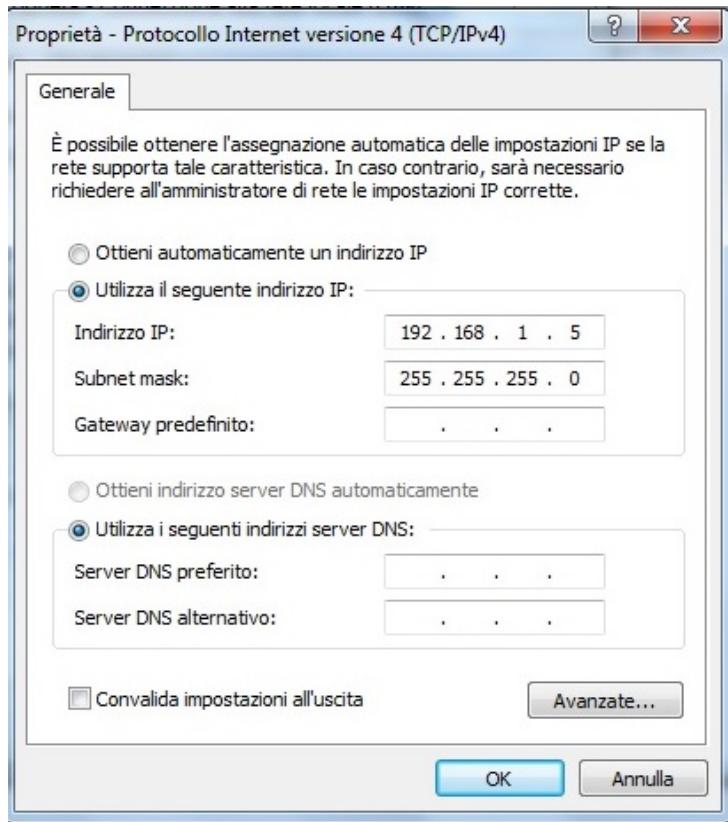


Figura 29: Internet properties

```
C:\Users\David>ping 192.168.1.4

Esecuzione di Ping 192.168.1.4 con 32 byte di dati:
Risposta da 192.168.1.4: byte=32 durata=1ms TTL=64
Risposta da 192.168.1.4: byte=32 durata<1ms TTL=64
Risposta da 192.168.1.4: byte=32 durata<1ms TTL=64
Risposta da 192.168.1.4: byte=32 durata=1ms TTL=64
```

Figura 30: Connection test

We executed the following commands to activate the ssh protocol. Then in the remote system panel we defined an ssh only connection with host name the board IP and a connection name. We opened it and it requested a password, but was only necessary to write root in the host name to gain the access.

```
root@socfpga_cyclone5:~# cp /etc/ssh/sshd config{,.orig}
root@socfpga_cyclone5:~# echo PermitEmptyPasswords yes >> /etc/ssh/sshd_config
root@socfpga_cyclone5:~# start-stop-daemon -K -x /usr/sbin/sshd
stopped /usr/sbin/sshd (pid 608)
root@socfpga_cyclone5:~# start-stop-daemon -S -x /usr/sbin/sshd
root@socfpga_cyclone5:~#
```

Figura 31: Commands

Once established a connection to test the debug function, we opened a software made by Altera: intelFPGA-embedded-examples-software-Altera-SoC FPGA-Blinking-LED-Linux-GNU.tar.gz

When we tried to build the project an error showed up due to a licensing problem:

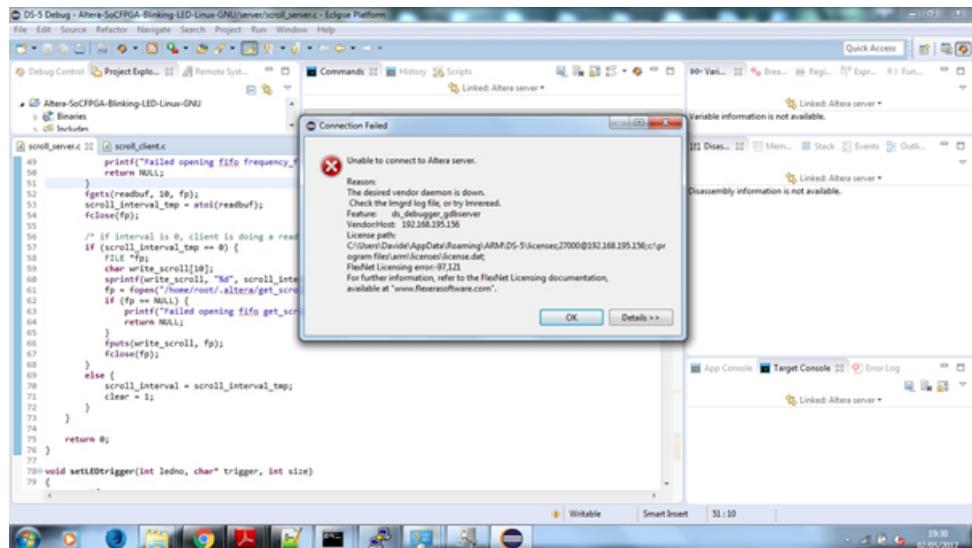


Figura 32: License error

Once downloaded the licence, it builded the project. Then we created a new configuration in the panel “Debug configuration” for the application server side following these steps:

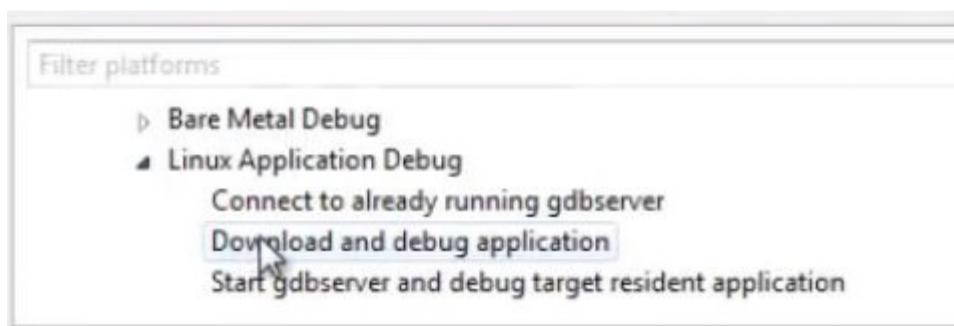
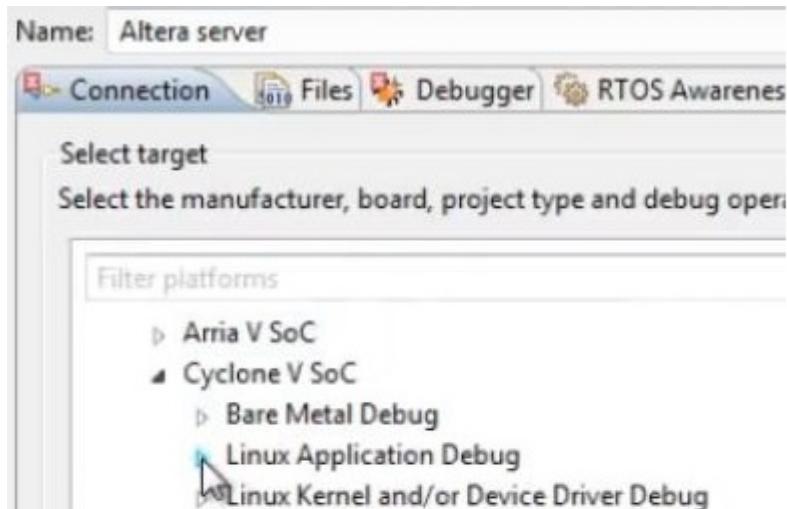


Figura 33: Download on board

DS-5 Debugger will download your application to the target system and then start a new gdbserver session to debug the application.

Figura 34: Download success

Same steps for the client side but changing the following parameters.
Click on Altera server and start debug.

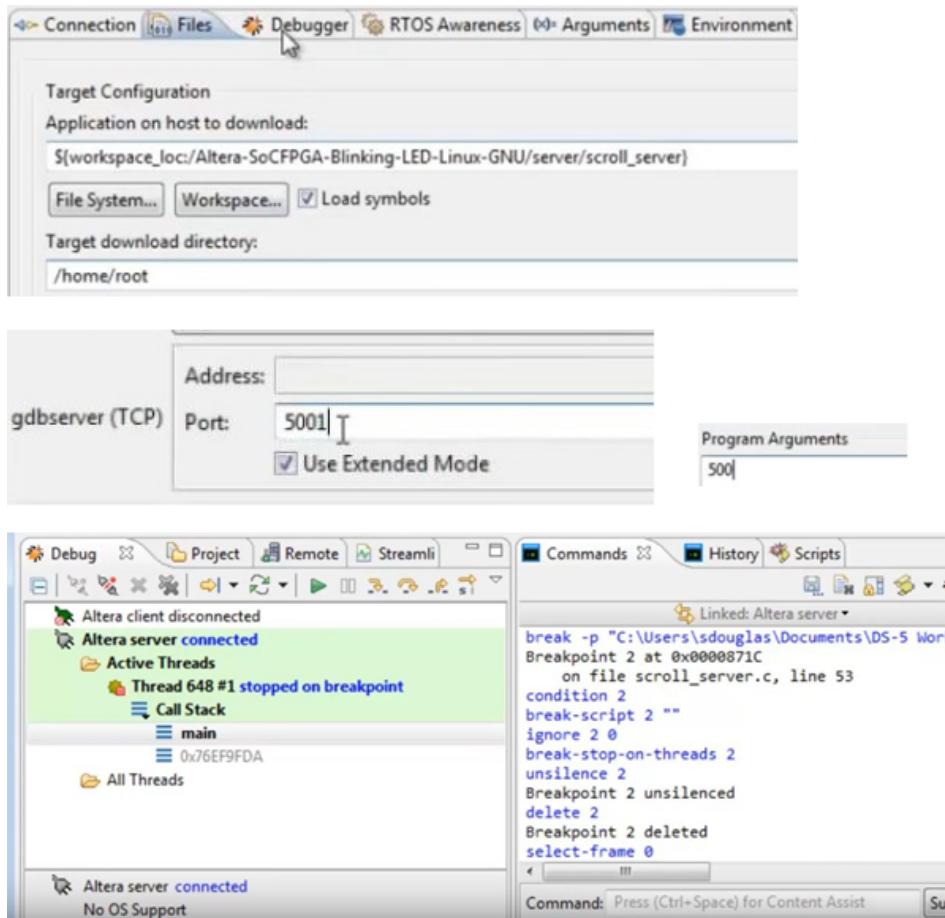


Figura 35: Settings

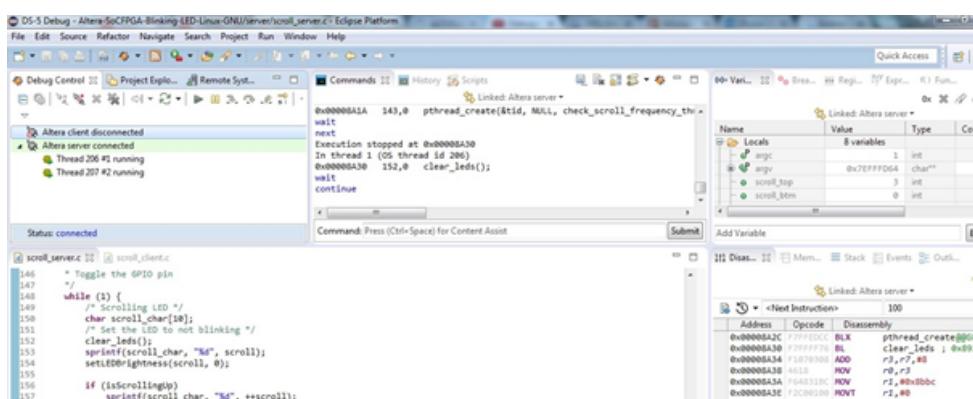


Figura 36: Debug

6

CONCLUSIONS

La mobilità delle persone e delle merci sono una componente essenziale del mercato interno dell'Unione Europea (UE) ed è di fondamentale importanza garantire la sua fattibilità al fine di salvaguardare la crescita economica. La rete ferroviaria ha un ruolo strategico in questo contesto almeno sotto due punti di vista:

Parte I
APPENDIX

A

ONLINE QUESTION

A.1 VXWORKS QUESTION

A.2 BLABLA QUESTION

- <https://www.altera.com/products/fpga/cyclone-series/cyclone-v-overview.html>
- https://knowledge.windriver.com/en-us/000_Products/000/020/010/000
- <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836&PartNo=4>
- <http://dl.altera.com/16.1/?edition=standard>
- <https://support.dce.felk.cvut.cz/psr/cvicensi/target/>
- <https://dl.altera.com/soceds/>
- https://www.altera.com/support/support-resources/download/rtos_tools.html
- <https://marketplace.windriver.com/index.php?bsp&on=details&bsp=12660>
- https://www.windriver.com/licensing/documents/wr_product_install_licensing_developers_guide_2.6.pdf
- https://www.xilinx.com/support/documentation/application_notes/xapp1114.pdf
- https://knowledge.windriver.com/@api/deki/files/241699/vxworks_bsp_developers_guide_6.9.pdf
- https://rocketboards.org/foswiki/view/Documentation/PreloaderUbootCustomization131#HPS_Boot_Flow
- <https://rocketboards.org/foswiki/view/Documentation/WebHome>
- https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-v/cv_5400a.pdf
- <http://zedboard.org/sites/default/files/documentations/GS-AES-Z7EV-7Z020-G-V7.pdf>
- http://zedboard.org/sites/default/files/documentations/CY7C64225_Setup_Guide_1_3.pdf
- http://zedboard.org/sites/default/files/documentations/ZedBoard_HW_UG_v2_2.pdf
- https://www.xilinx.com/support/documentation/application_notes/xapp1158-zynq-7000-vxworks-bsp.pdf
- https://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug1043-embedded-system-tools.pdf

- <http://www.wiki.xilinx.com/Prepare+boot+image>
- <https://forums.xilinx.com/t5/Embedded-Development-Tools/AP-transaction-Error/td-p/369465>