

Creating Embedded Linux Solutions

Course Workbook

Table of Contents

About this Workbook	<u>Page 3</u>
Pre-Lab: Workshop Pre-requisites	<u>Page 4</u>
Lab 3: Creating Multi Processor Solutions	<u>Page 7</u>

About this Workbook

The contents of this workbook are created by Aduvo Engineering & Training, Ltd.

If you have any questions about the contents, or need assistance, please contact Adam Taylor at adam@adiuvoengineering.com.

Pre-Lab

Creating Embedded Linux Solutions

Required Hardware

Arty Z7-20

SD Card

Micro USB Cable

Downloads and Installations

Step 1 – Download and install the following at least 1 day prior to the workshop. This may take a significant amount of time and drive space.

Watch the video available [here](#) to show how to configure the installation

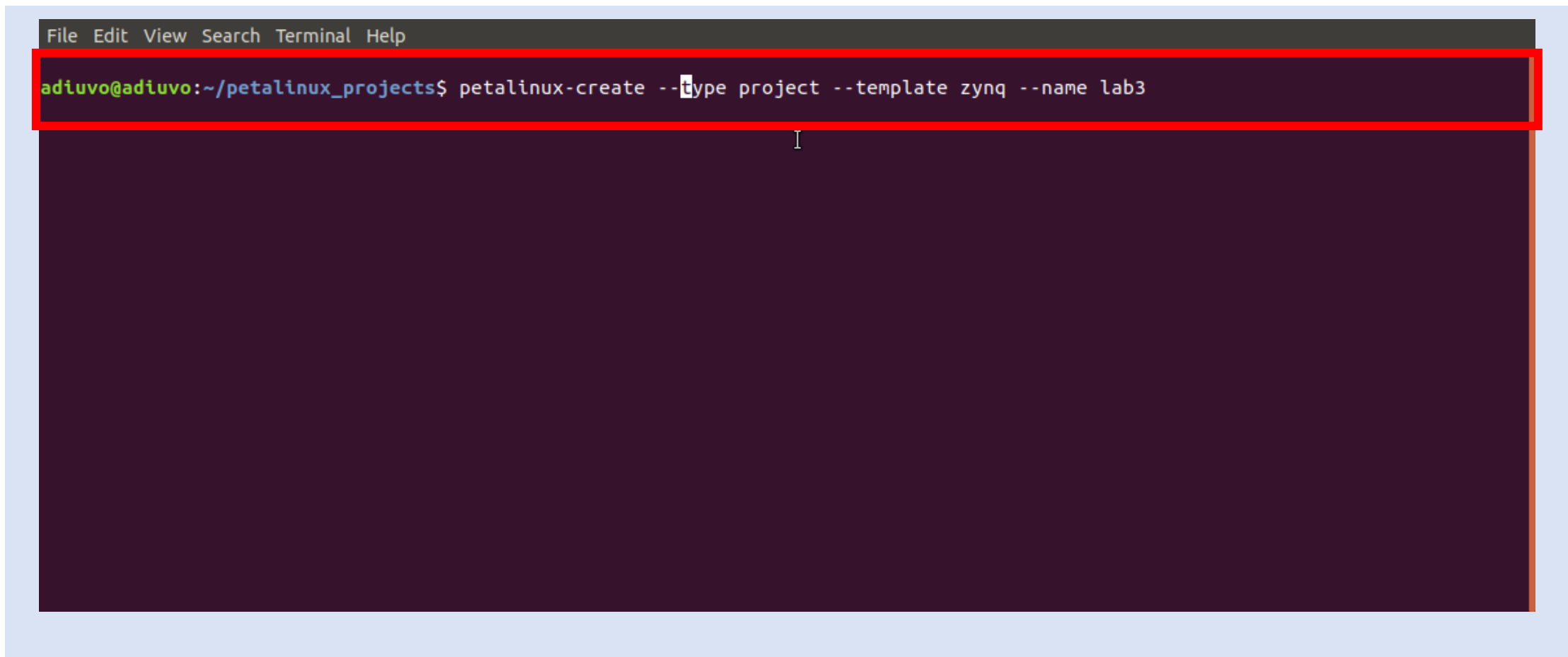
Vitis 2021.1	Download
PetaLinux 2021.1	Download

Lab 3

Creating Embedded Linux Solutions

Lab 3: Creating Embedded Linux Solutions

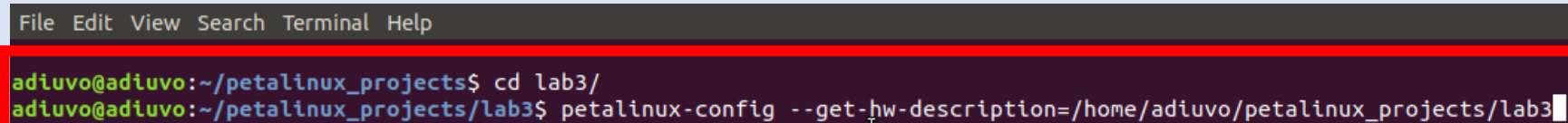
Step 1 – Using a Linux Development machine with PetaLinux 2021.1 installed create a new project

A terminal window with a dark background and light text. The menu bar at the top includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The prompt is 'adiuvo@adiuvo:~/petalinux_projects\$'. The command entered is 'petalinux-create --type project --template zynq --name lab3'. A red rectangular box highlights the command line. A cursor is visible at the end of the command.

```
File Edit View Search Terminal Help
adiuvo@adiuvo:~/petalinux_projects$ petalinux-create --type project --template zynq --name lab3
```


Lab 3: Creating Embedded Linux Solutions

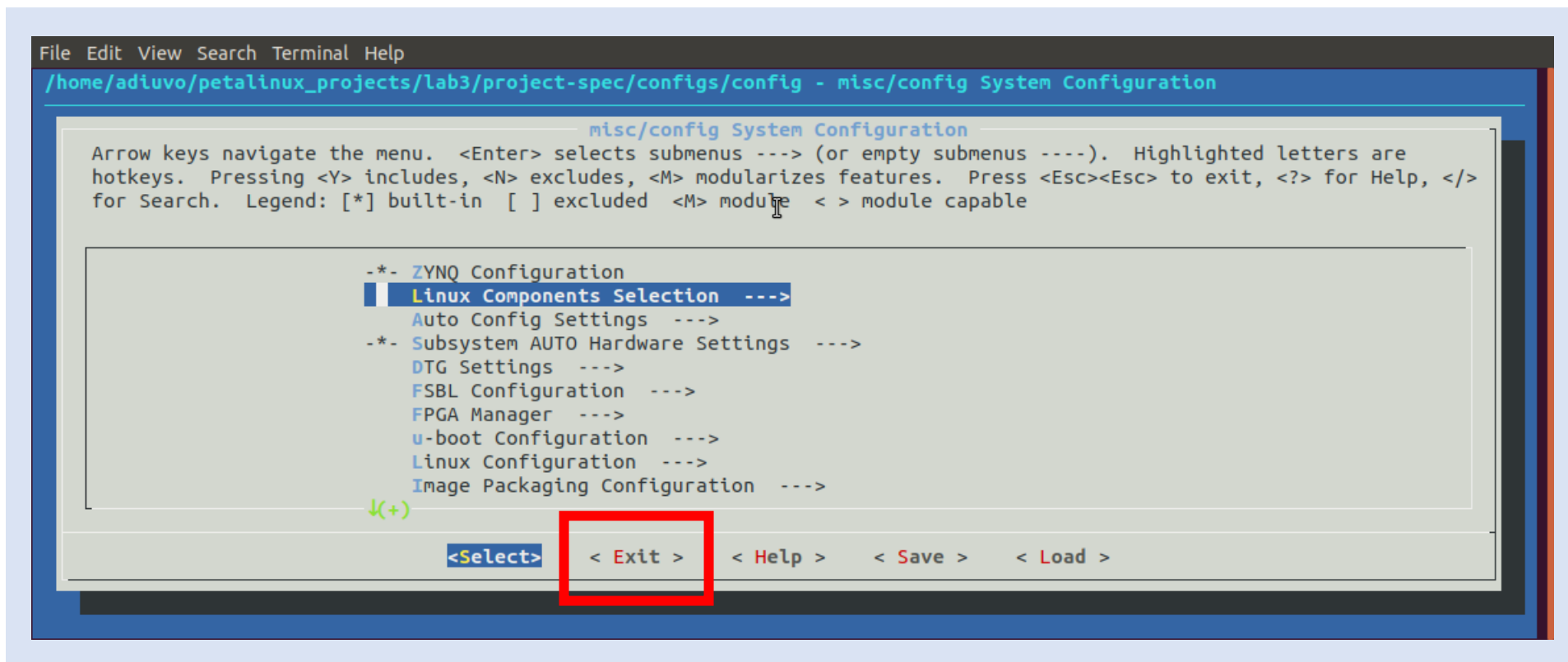
Step 2 – Change directory into the project, import the XSA exported from Lab One

A terminal window with a dark background and light text. The window title bar shows 'File Edit View Search Terminal Help'. The terminal content shows two lines of text: the first line is 'adiuvo@adiuvo:~/petalinux_projects\$ cd lab3/' and the second line is 'adiuvo@adiuvo:~/petalinux_projects/lab3\$ petalinux-config --get-hw-description=/home/adiuvo/petalinux_projects/lab3'. A red rectangular box highlights the first two lines of the terminal output.

```
File Edit View Search Terminal Help
adiuvo@adiuvo:~/petalinux_projects$ cd lab3/
adiuvo@adiuvo:~/petalinux_projects/lab3$ petalinux-config --get-hw-description=/home/adiuvo/petalinux_projects/lab3
```

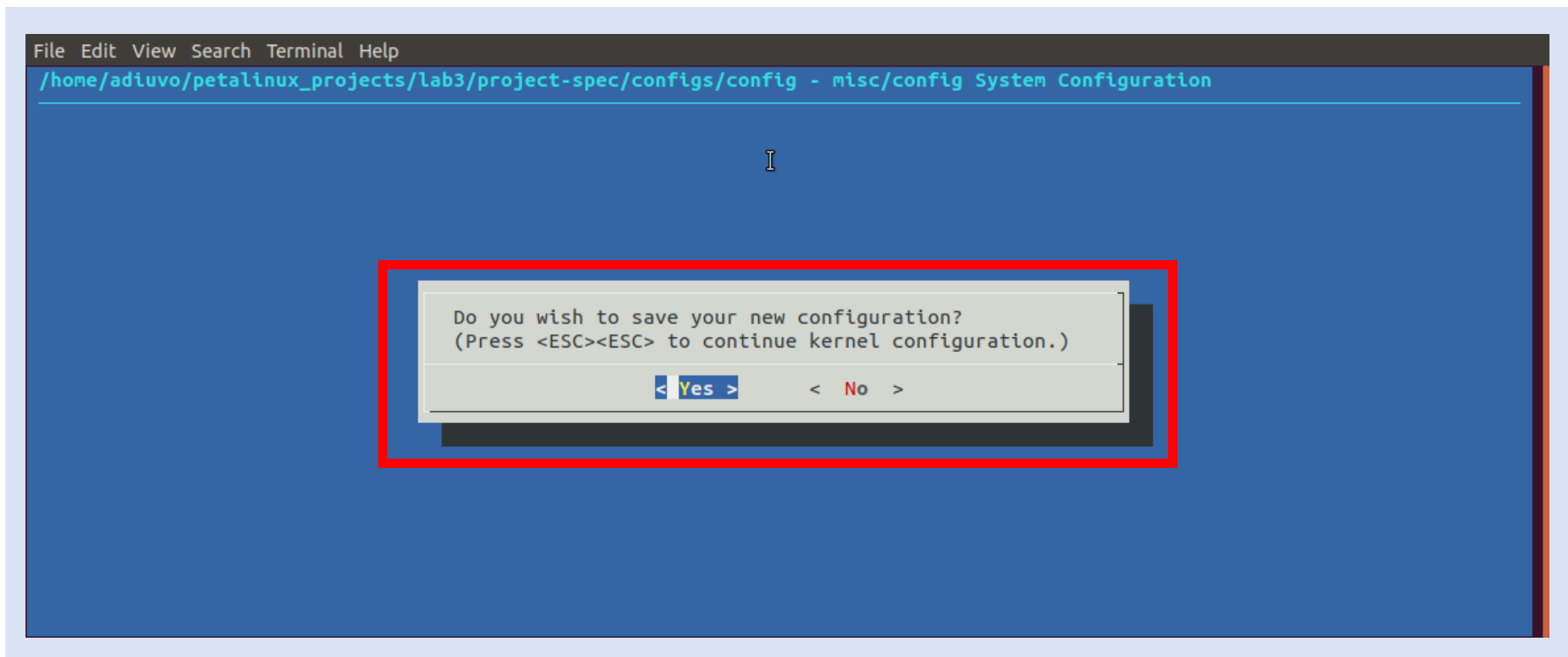
Lab 3: Creating Embedded Linux Solutions

Step 3 – Exit the dialog which is opened following the XSA Import.



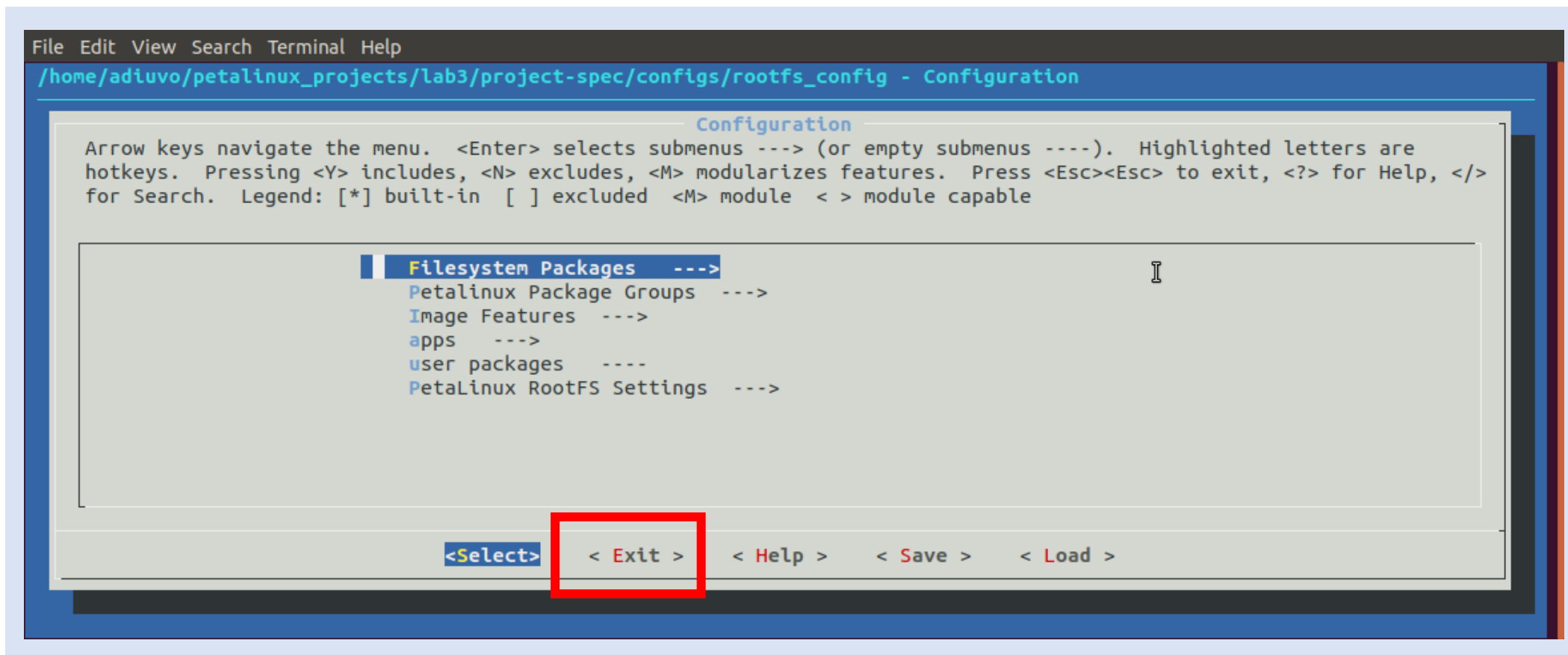
Lab 3: Creating Embedded Linux Solutions

Step 4 – When asked save any changes



Lab 3: Creating Embedded Linux Solutions

Step 5 – Enter the command `petalinux-config -c rootfs` to examine the rootfs dialog. Close without saving changes after exploring the settings.



```
File Edit View Search Terminal Help
/home/adiuvo/petalinux_projects/lab3/project-spec/configs/rootfs_config - Configuration

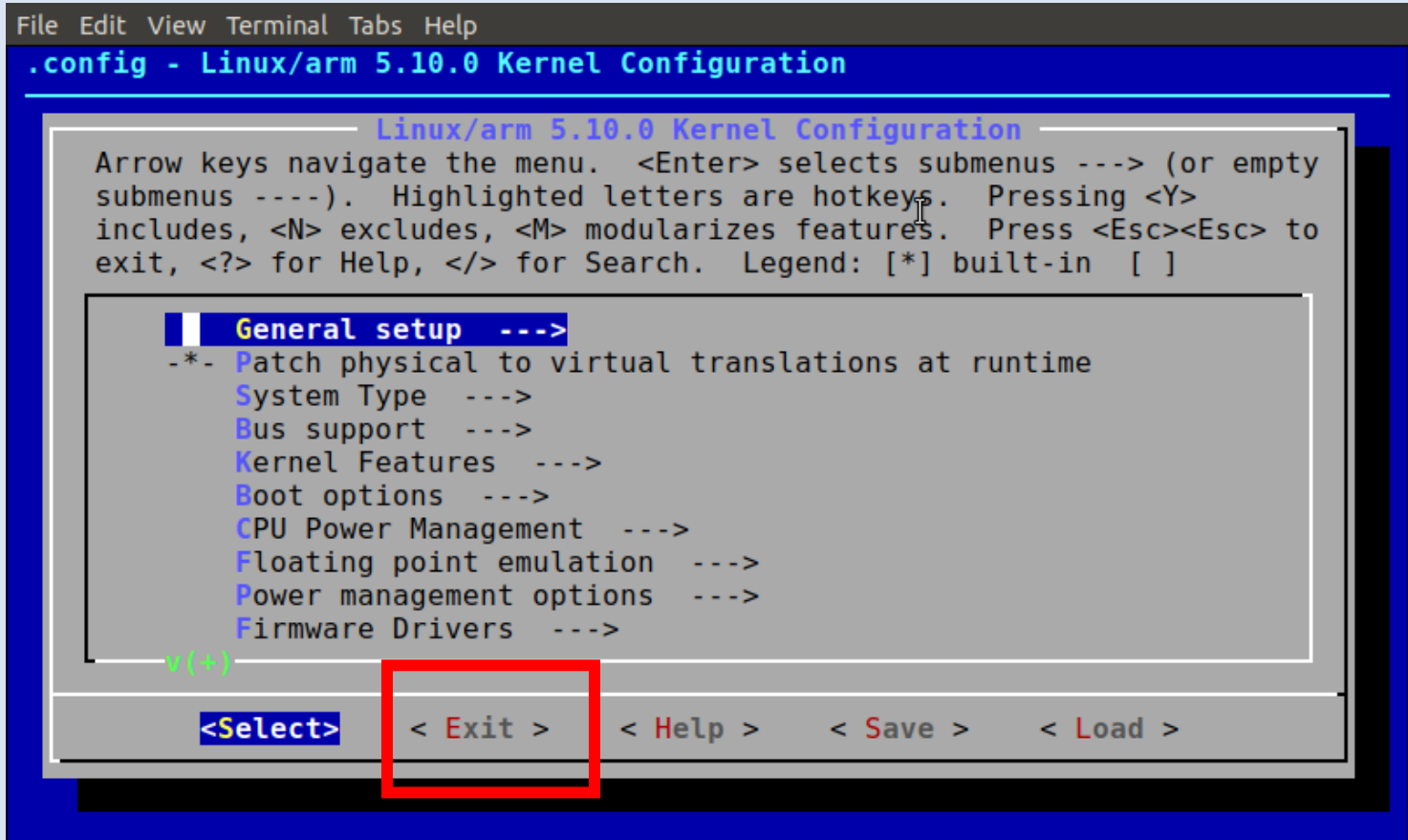
Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

Filesystem Packages --->
Petalinux Package Groups --->
Image Features --->
apps --->
user packages ----
Petalinux RootFS Settings --->

<Select> < Exit > < Help > < Save > < Load >
```

Lab 3: Creating Embedded Linux Solutions

Step 6 – Enter the command `petalinux-config -c kernel` to view the config settings, do not change anything and exist after exploring.



```
File Edit View Terminal Tabs Help
.config - Linux/arm 5.10.0 Kernel Configuration

Linux/arm 5.10.0 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

General setup --->
-* Patch physical to virtual translations at runtime
System Type --->
Bus support --->
Kernel Features --->
Boot options --->
CPU Power Management --->
Floating point emulation --->
Power management options --->
Firmware Drivers --->

v(*)
<Select> < Exit > < Help > < Save > < Load >
```

Lab 3: Creating Embedded Linux Solutions

Step 7 – Build the petalinux project with the command `petalinux-build` – this may take some time.

```
File Edit View Search Terminal Help
adiuvo@adiuvo:~/petalinux_projects$ cd lab3/
adiuvo@adiuvo:~/petalinux_projects/lab3$ petalinux-config --get-hw-description=/home/adiuvo/petalinux_projects/lab3
[INFO] Sourcing buildtools
INFO: Getting hardware description...
INFO: Renaming design_1_wrapper.xsa to system.xsa
[INFO] Generating Kconfig for project
[INFO] Menuconfig project
configuration written to /home/adiuvo/petalinux_projects/lab3/project-spec/configs/config

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO] Extracting yocto SDK to components/yocto. This may take time!
[INFO] Sourcing build environment
[INFO] Generating kconfig for Rootfs
[INFO] Silentconfig rootfs
[INFO] Generating plnxtool conf
[INFO] Adding user layers
[INFO] Generating workspace directory
adiuvo@adiuvo:~/petalinux_projects/lab3$ petalinux-build
```

Lab 3: Creating Embedded Linux Solutions

Step 8 – Package the petalinux project to create the boot.bin necessary

```
File Edit View Search Terminal Help
adiuvo@adiuvo:~/petalinux_projects/lab3$ cd images/linux/
adiuvo@adiuvo:~/petalinux_projects/lab3/images/linux$ petalinux-package --boot --fsbl zynq_fsbl.elf --u-boot u-boot.elf --fpga
system.bit --force
[INFO] Sourcing buildtools
INFO: Getting system flash information...
INFO: File in BOOT BIN: "/home/adiuvo/petalinux_projects/lab3/images/linux/zynq_fsbl.elf"
INFO: File in BOOT BIN: "/home/adiuvo/petalinux_projects/lab3/images/linux/system.bit"
INFO: File in BOOT BIN: "/home/adiuvo/petalinux_projects/lab3/images/linux/u-boot.elf"
INFO: File in BOOT BIN: "/home/adiuvo/petalinux_projects/lab3/images/linux/system.dtb"
INFO: Generating Zynq binary package BOOT.BIN...

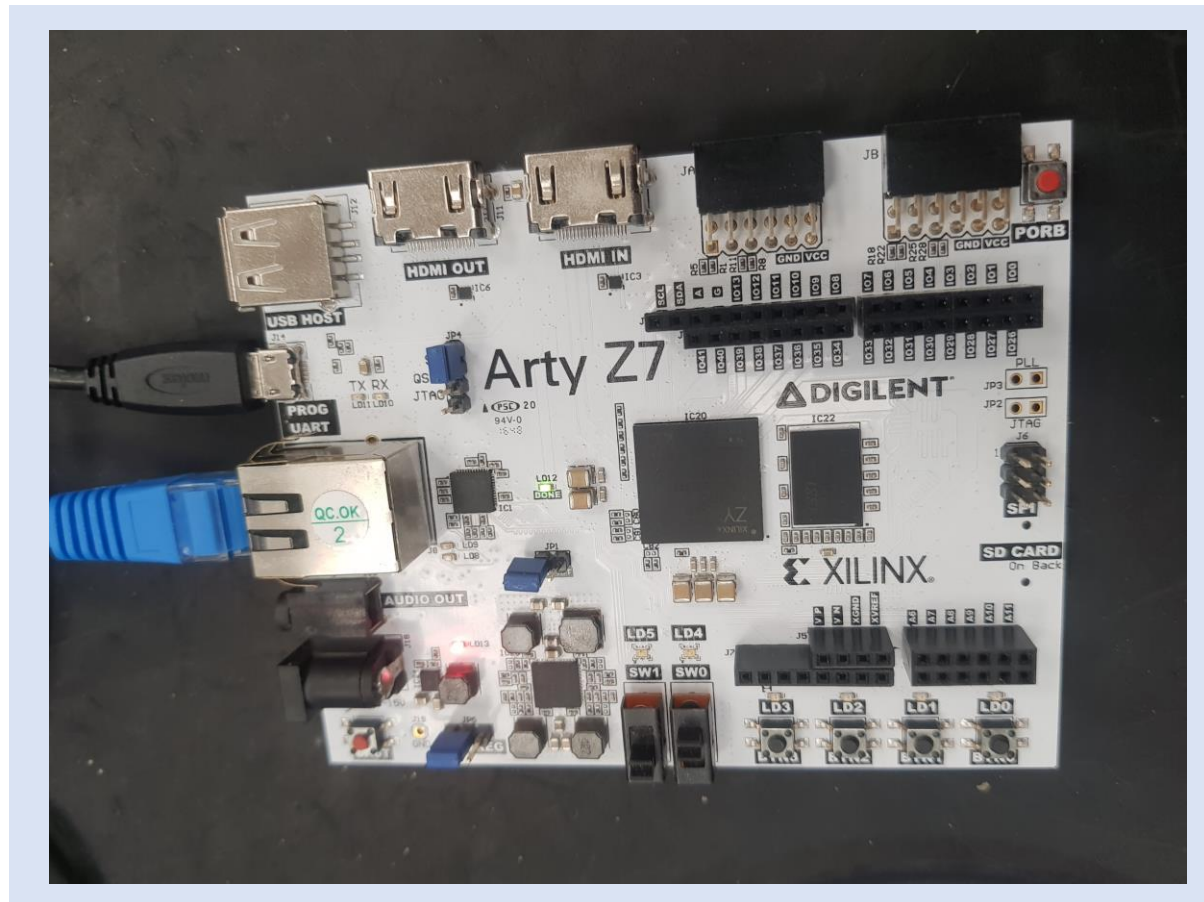
***** Xilinx Bootgen v2021.1
**** Build date : May 28 2021-21:36:22
** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

[INFO] : Bootimage generated successfully

INFO: Binary is ready.
WARNING: Unable to access the TFTPBOOT folder /tftpboot!!!
WARNING: Skip file copy to TFTPBOOT folder!!!
adiuvo@adiuvo:~/petalinux_projects/lab3/images/linux$
```

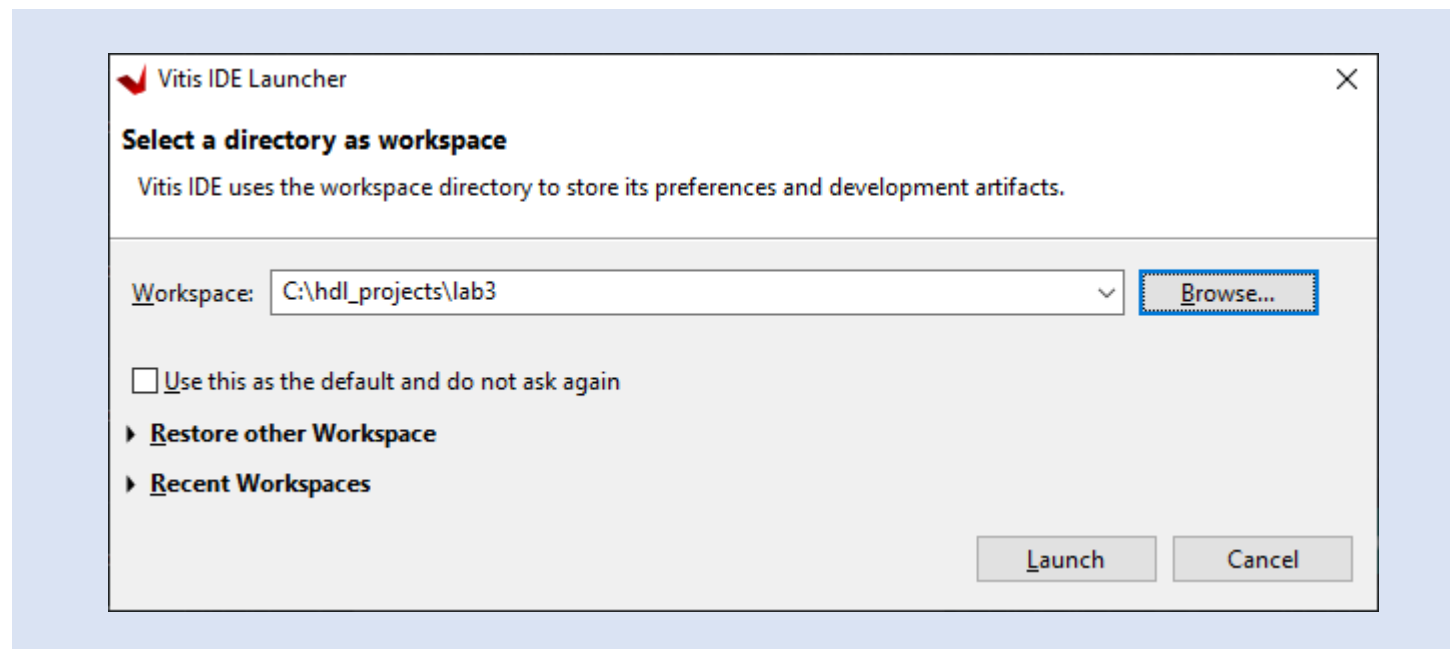
Lab 3: Creating Embedded Linux Solutions

Step 9 – Copy the image.ub boot.bin and boot.scr to the SD Card, insert in the Arty Z7 – ensure it is set for SD Card boot. Connect the power and Ethernet cable.



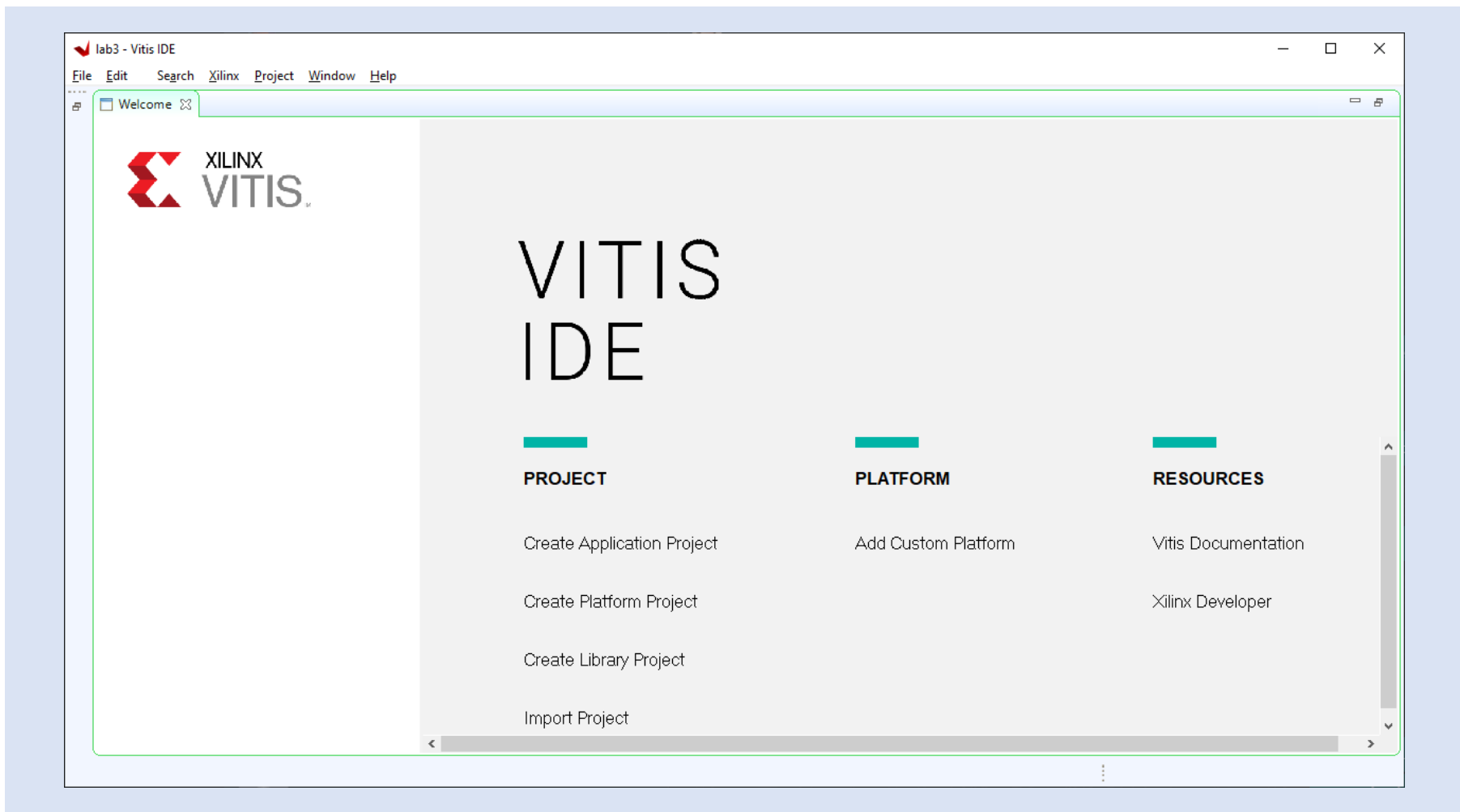
Lab 3: Creating Embedded Linux Solutions

Step 10 – Vitis will start and ask for the workspace, create a new directory using browse to store the projects.



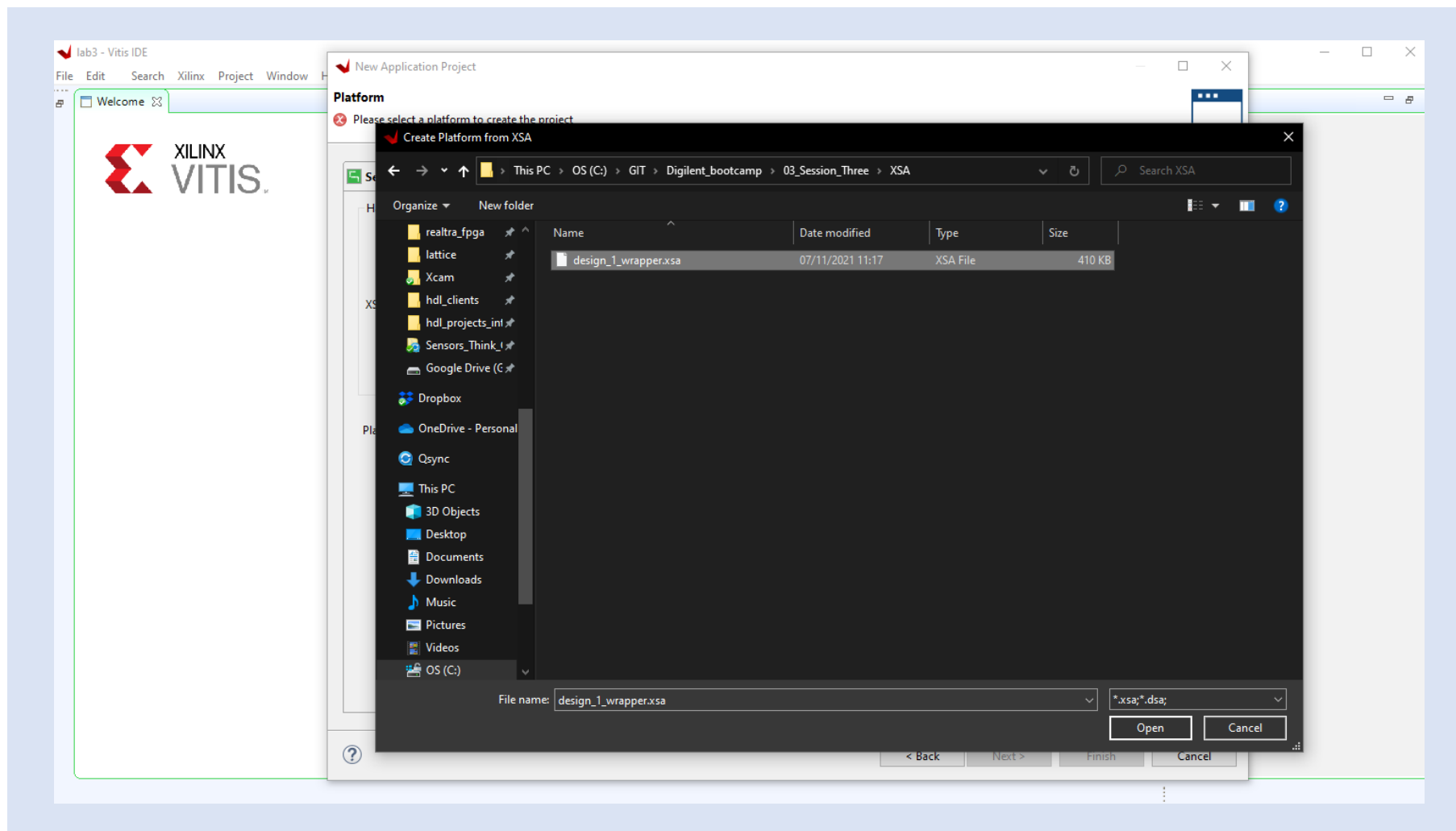
Lab 3: Creating Embedded Linux Solutions

Step 11 – Select Create Application Project



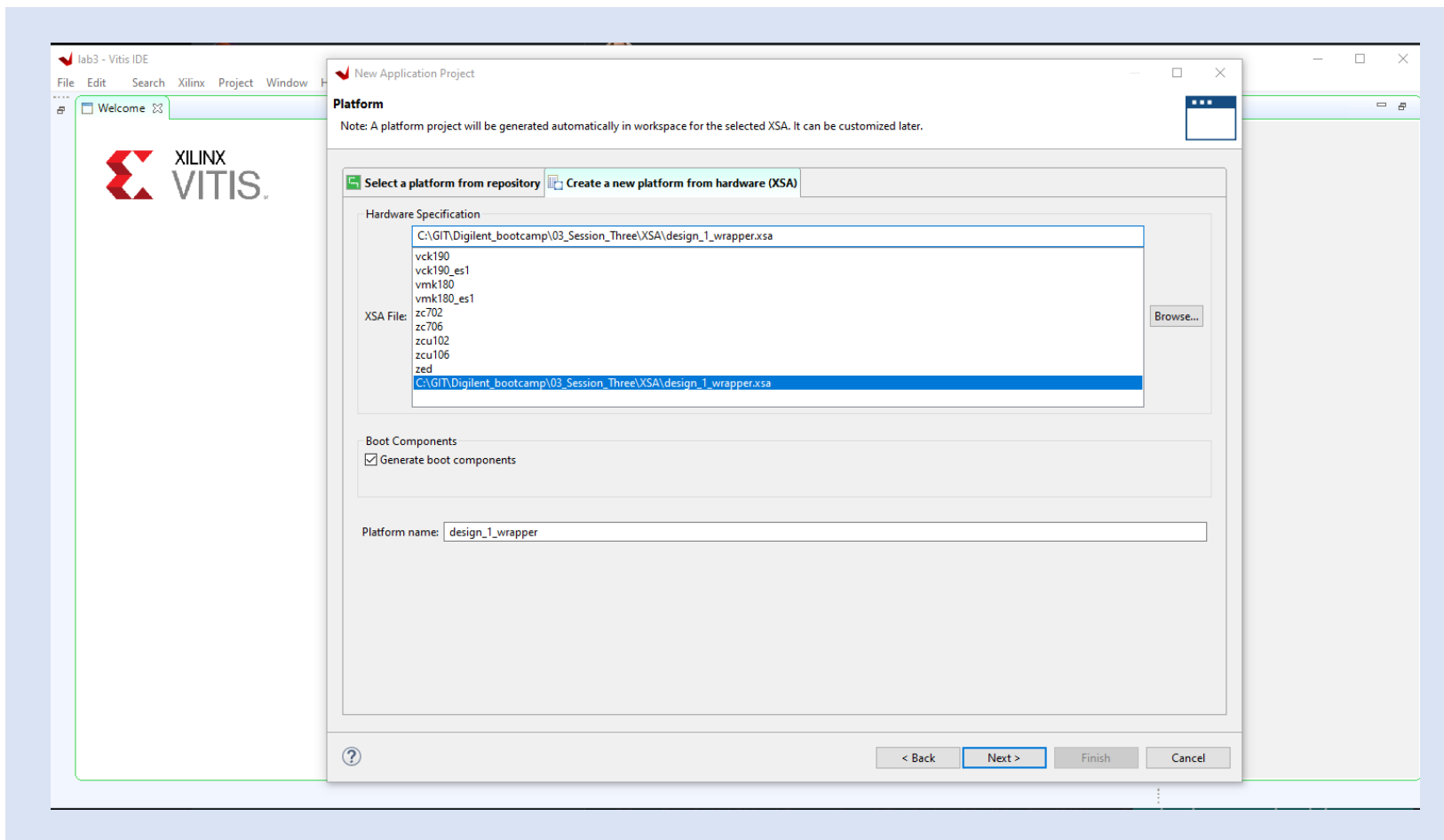
Lab 3: Creating Embedded Linux Solutions

Step 12 – Select the XSA downloaded with GitHub Repository under lab three



Lab 3: Creating Embedded Linux Solutions

Step 13 – Click next



Lab 3: Creating Embedded Linux Solutions

Step 14 – Enter a name for the project and select the processors in SMP configuration

New Application Project

Application Project Details
Specify the application project name and its system project properties

Application project name: test

System Project
Create a new system project for the application or select an existing one from the workspace

Select a system project
+ Create new...

System project details
System project name: test_system

Target processor
Select target processor for the Application project.

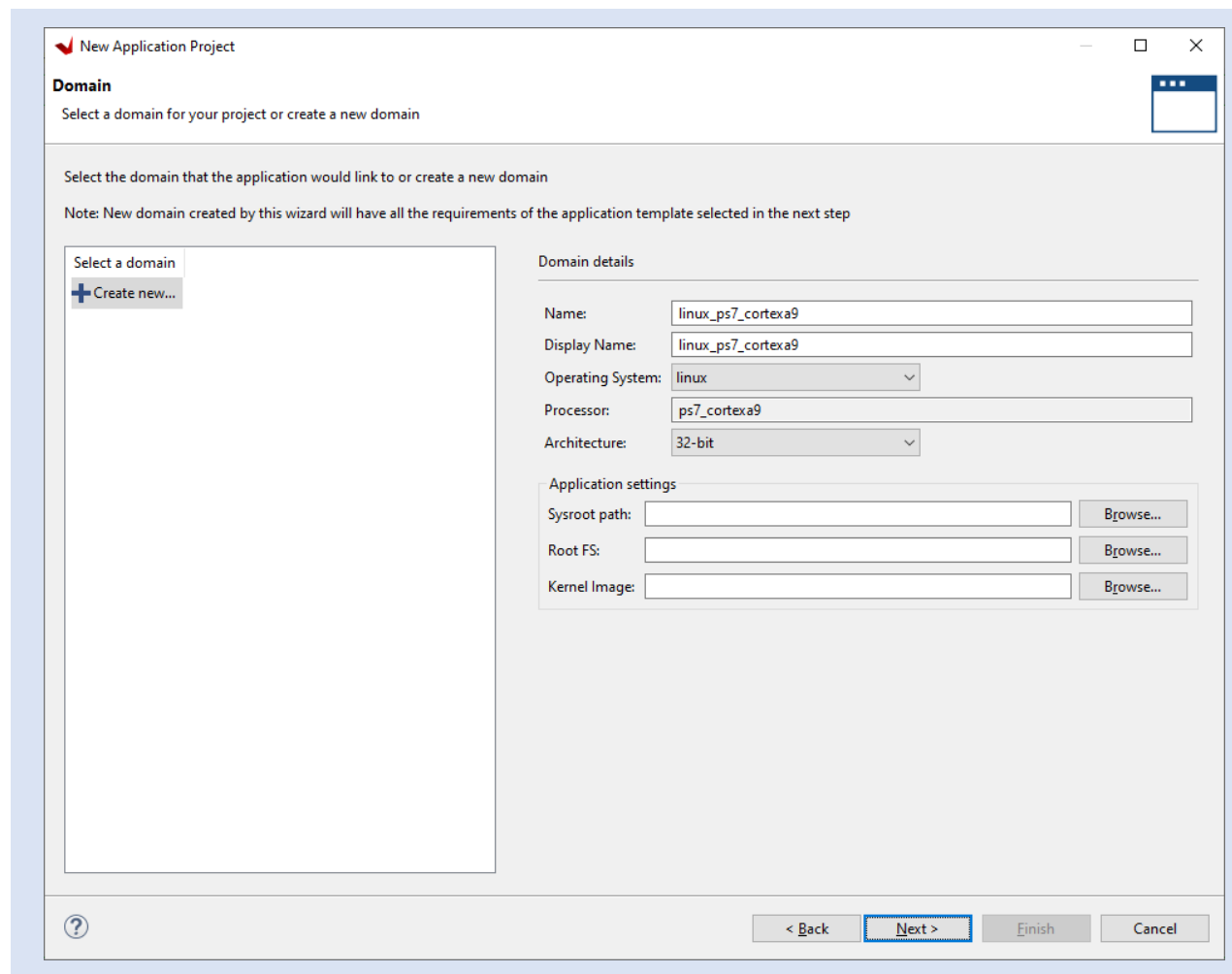
Processor	Associated applications
ps7_cortexa9_0	
ps7_cortexa9_1	
ps7_cortexa9 SMP	test

Show all processors in the hardware specification

< Back Next > Finish Cancel

Lab 3: Creating Embedded Linux Solutions

Step 15 – Click Next on the Domain page



New Application Project

Domain
Select a domain for your project or create a new domain

Select the domain that the application would link to or create a new domain

Note: New domain created by this wizard will have all the requirements of the application template selected in the next step

Select a domain
+ Create new...

Domain details

Name: linux_ps7_cortexa9
Display Name: linux_ps7_cortexa9
Operating System: linux
Processor: ps7_cortexa9
Architecture: 32-bit

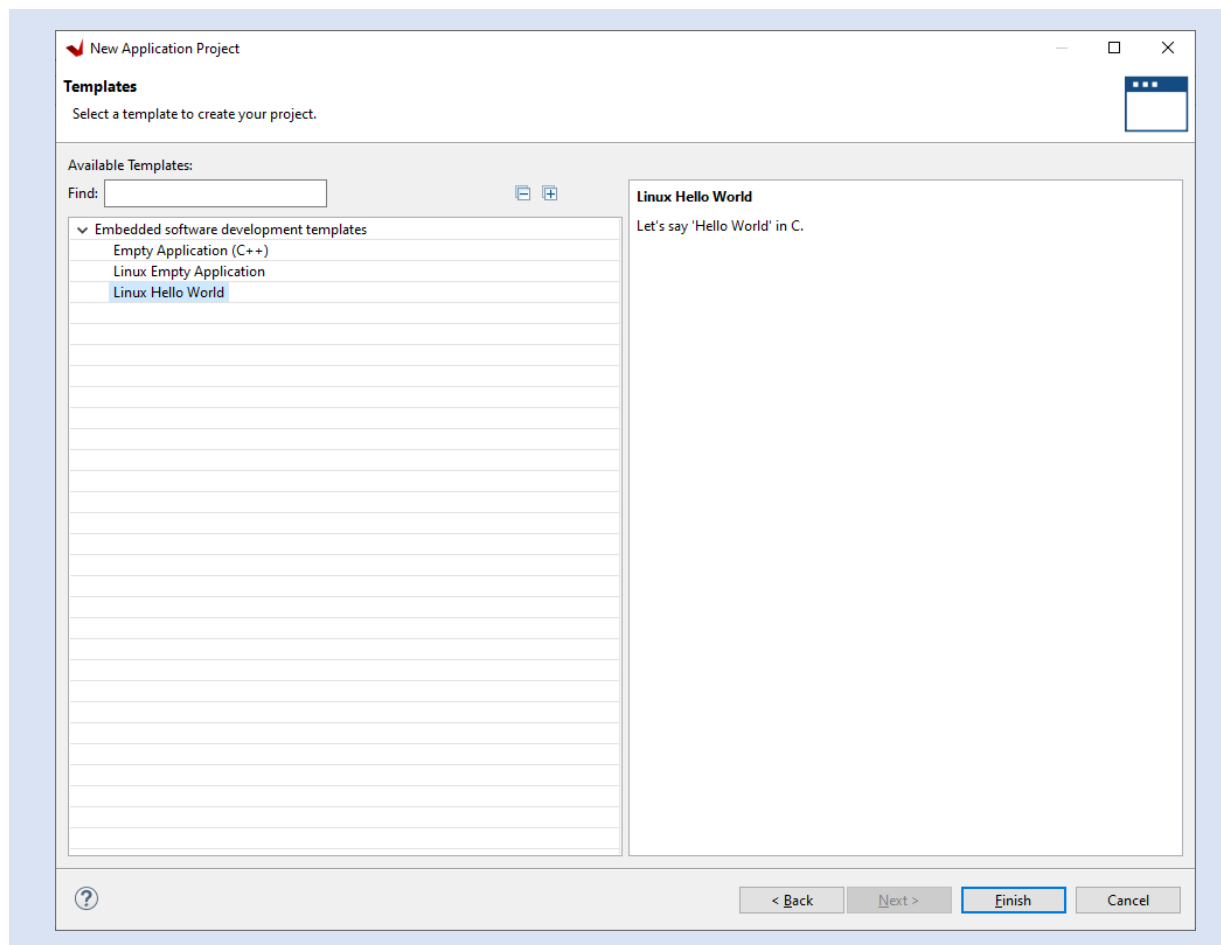
Application settings

Sysroot path: Browse...
Root FS: Browse...
Kernel Image: Browse...

< Back **Next >** Finish Cancel

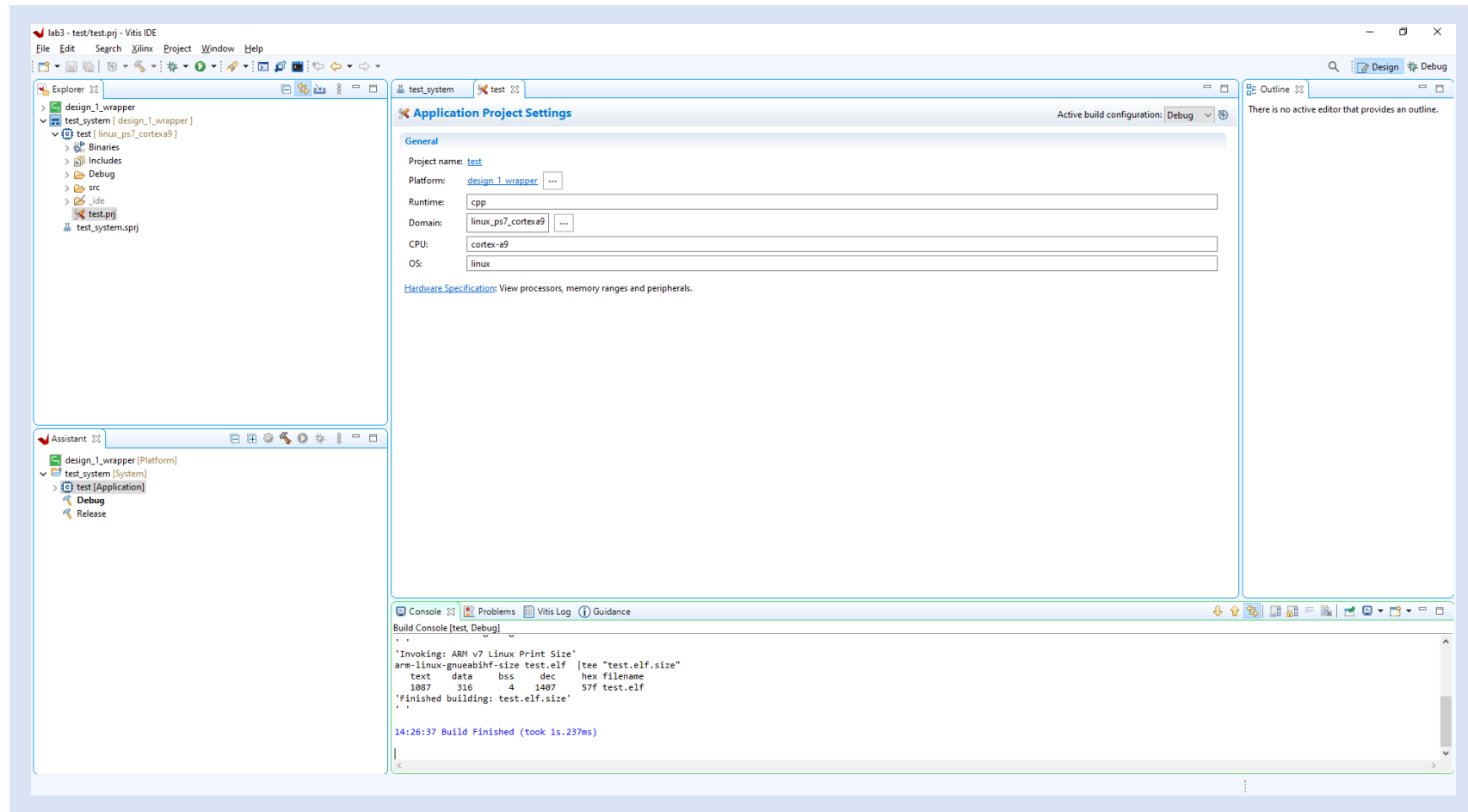
Lab 3: Creating Embedded Linux Solutions

Step 16 – Select the Linux Hello World application and click finish



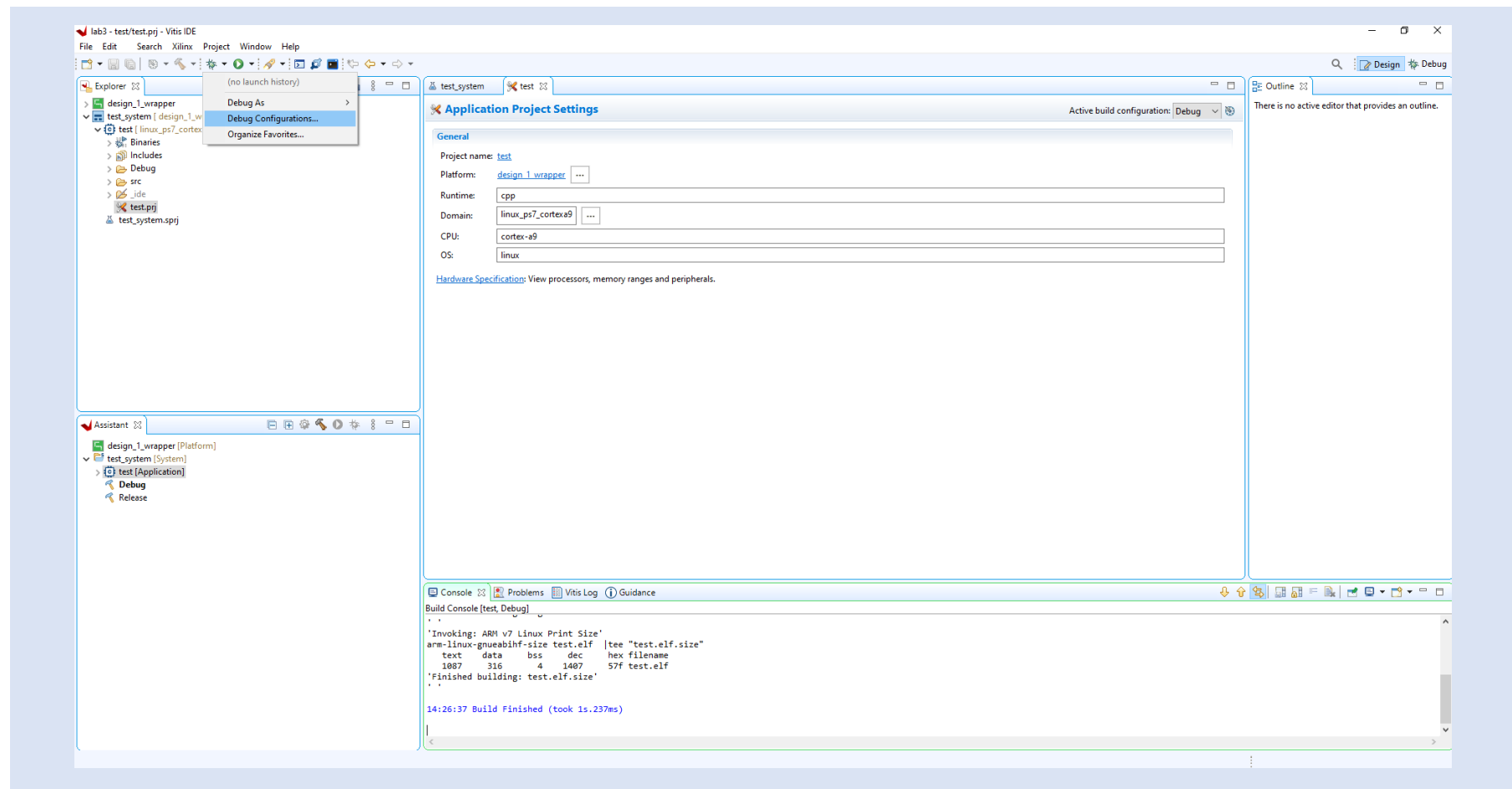
Lab 3: Creating Embedded Linux Solutions

Step 17 – Click on the Hammer to build the application



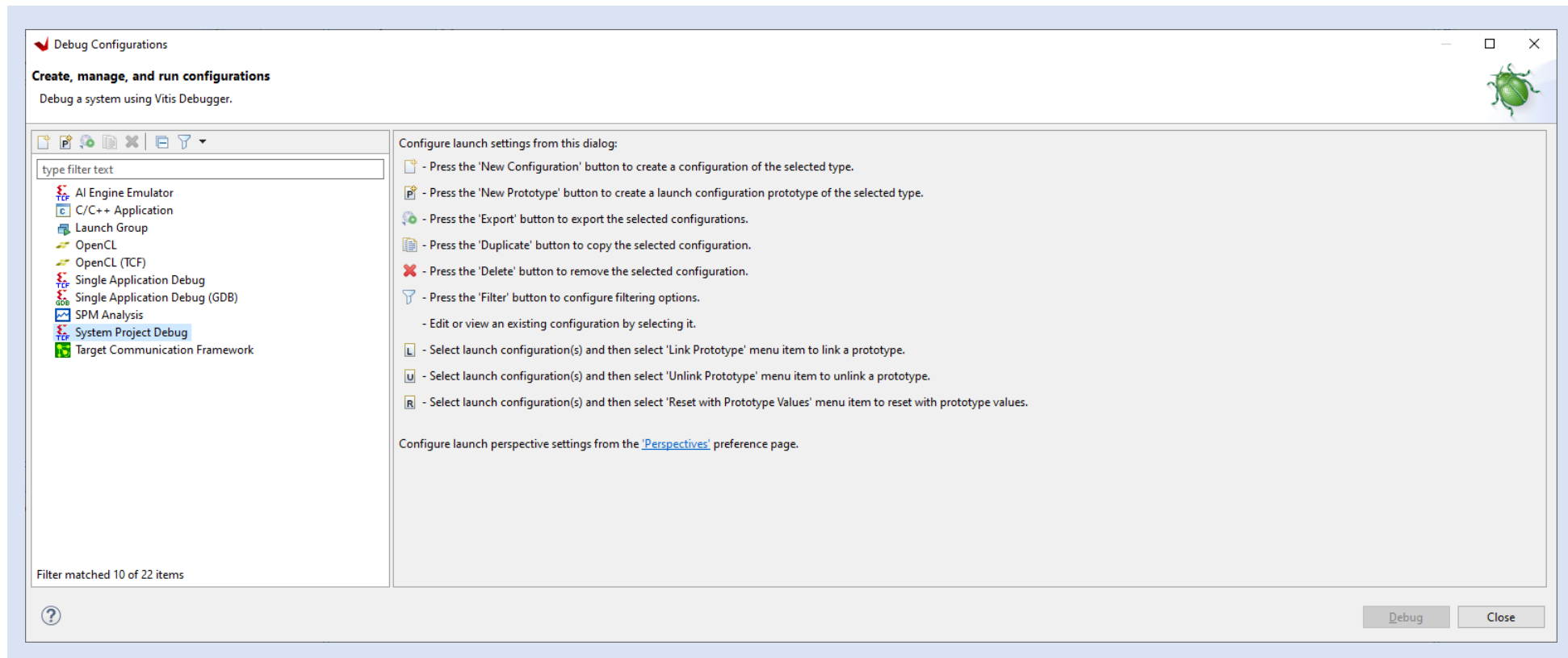
Lab 3: Creating Embedded Linux Solutions

Step 18 – Select the arrow by the green debug icon and select debug configuration



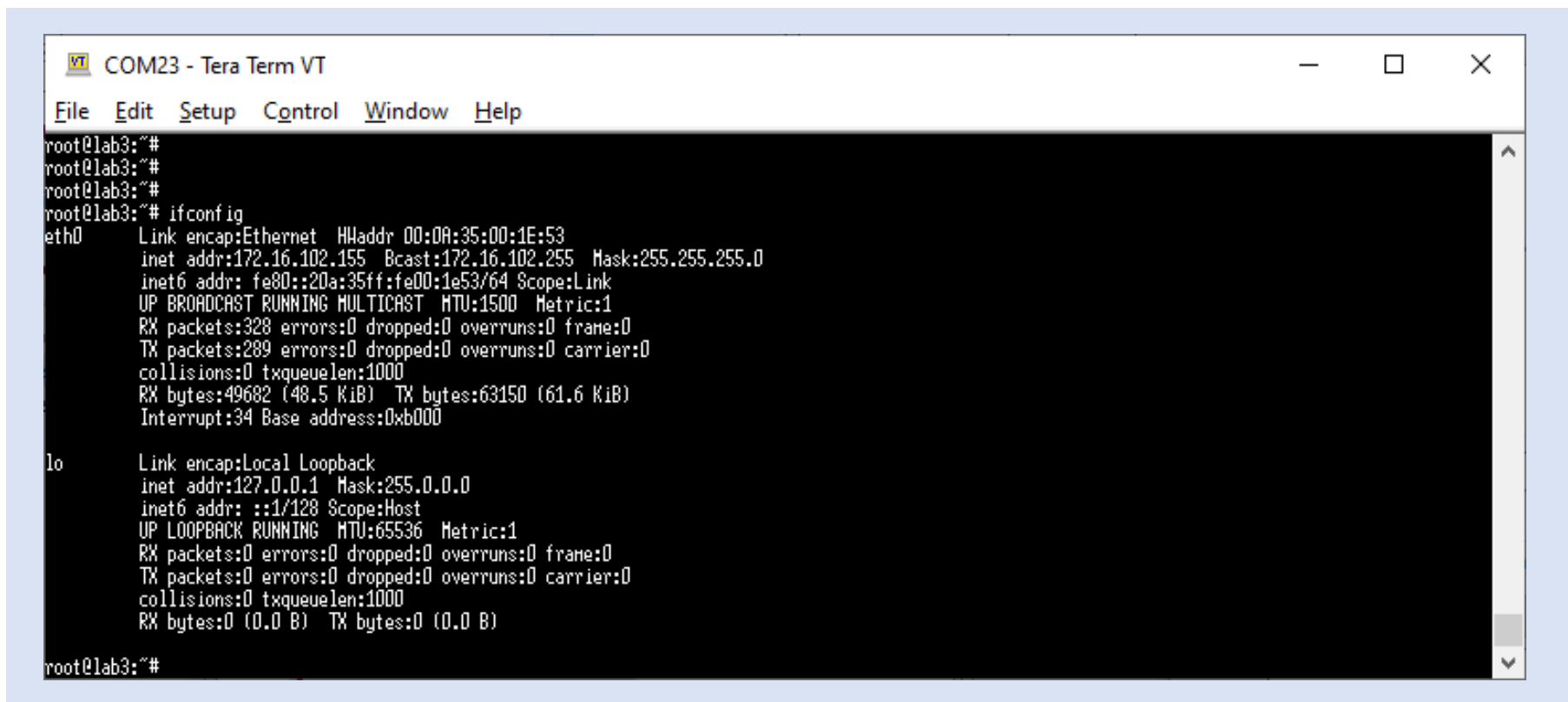
Lab 3: Creating Embedded Linux Solutions

Step 19 – Select System Project Debug and click new



Lab 3: Creating Embedded Linux Solutions

Step 20 – Open a serial terminal 115200:n:1 and type the command ifconfig to list the IP Address of the board



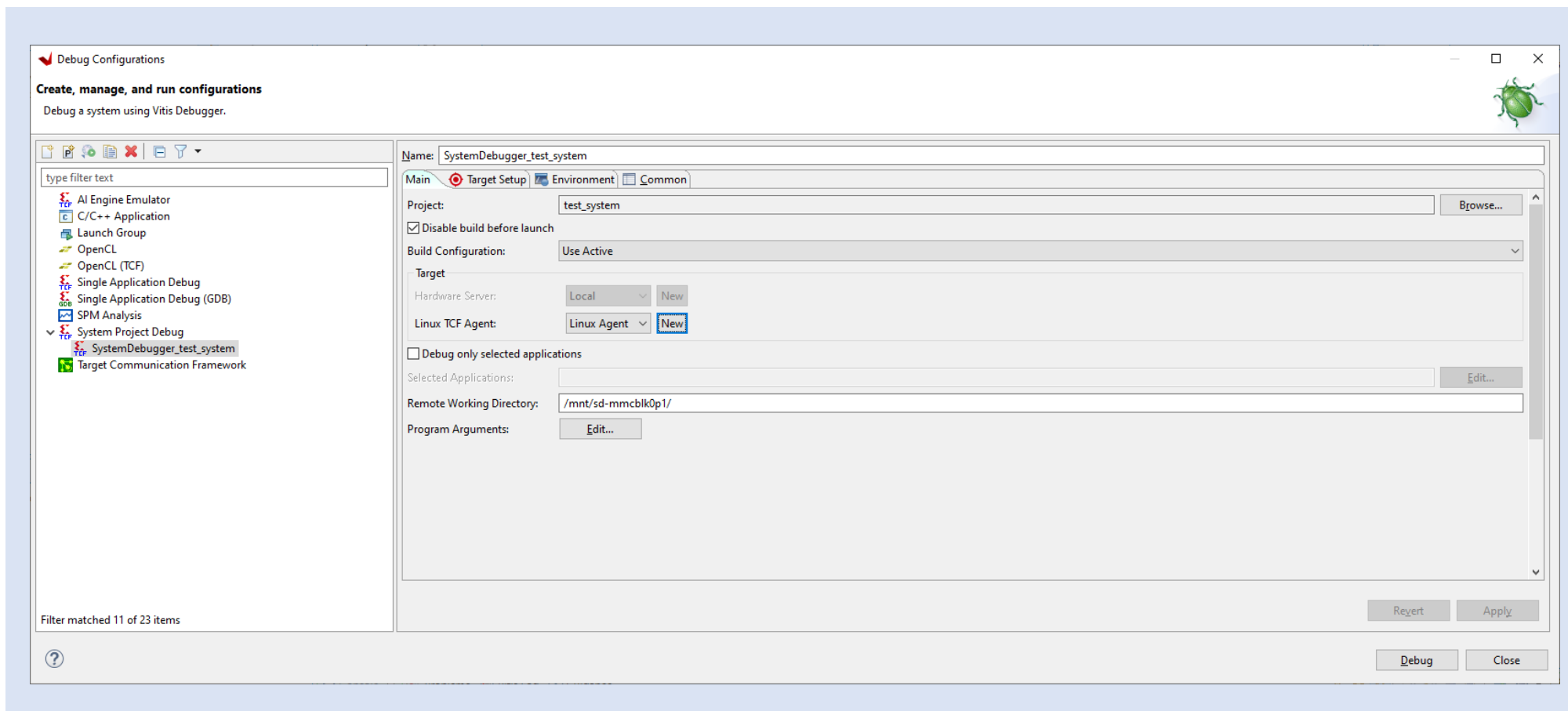
```
COM23 - Tera Term VT
File Edit Setup Control Window Help
root@lab3:~#
root@lab3:~#
root@lab3:~#
root@lab3:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0A:35:00:1E:53
          inet addr:172.16.102.155 Bcast:172.16.102.255 Mask:255.255.255.0
          inet6 addr: fe80::20a:35ff:fe00:1e53/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:328 errors:0 dropped:0 overruns:0 frame:0
          TX packets:289 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:49682 (48.5 KiB) TX bytes:63150 (61.6 KiB)
          Interrupt:34 Base address:0xb000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

root@lab3:~#
```

Lab 3: Creating Embedded Linux Solutions

Step 21 – Change the build configuration to Hardware and for Linux TCF Agent select new



Lab 3: Creating Embedded Linux Solutions

Step 22 – Enter a name and enter the Ethernet IP address – Click test connection to verify the board can be seen.

Target Connection Details

New Target Connection
Creates new configuration for connecting to a target.

Target Name

Set as default target

Specify the connection type and properties

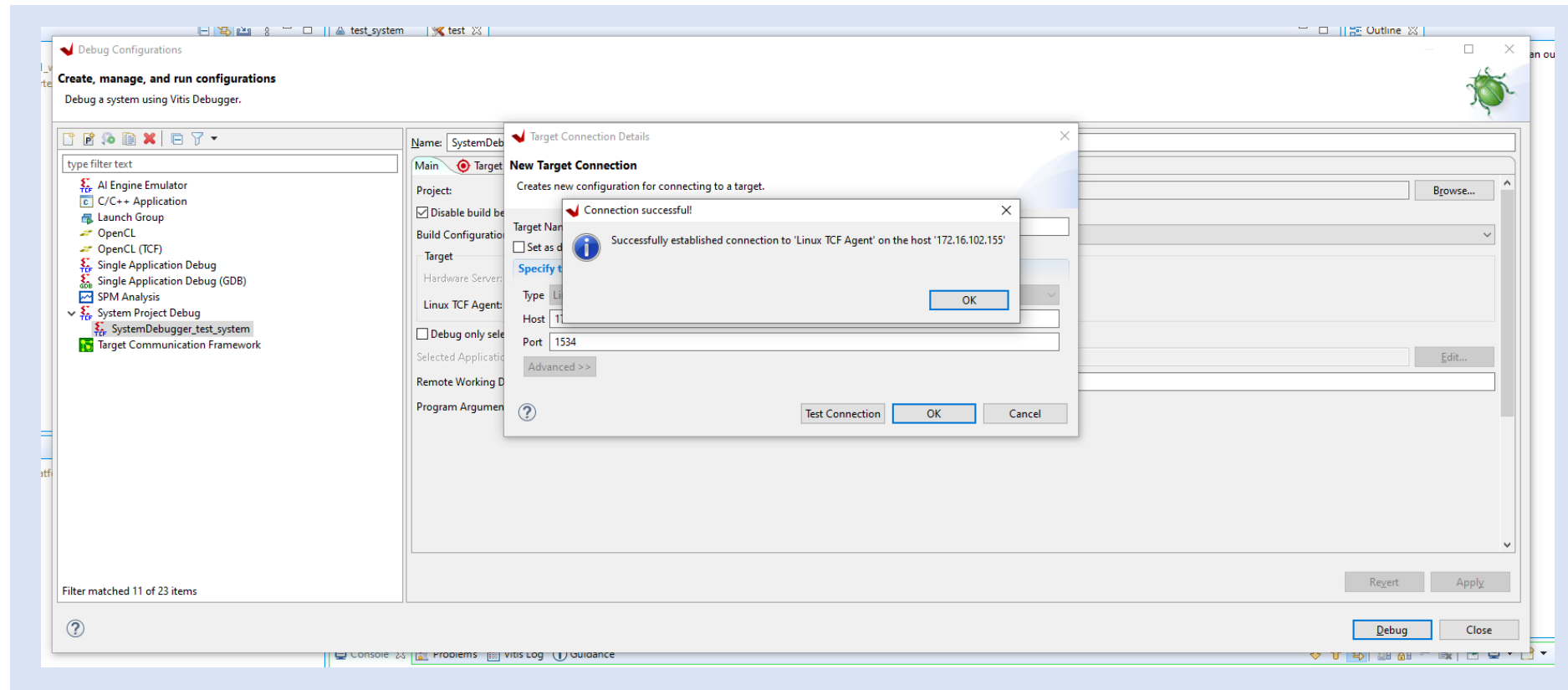
Type

Host

Port

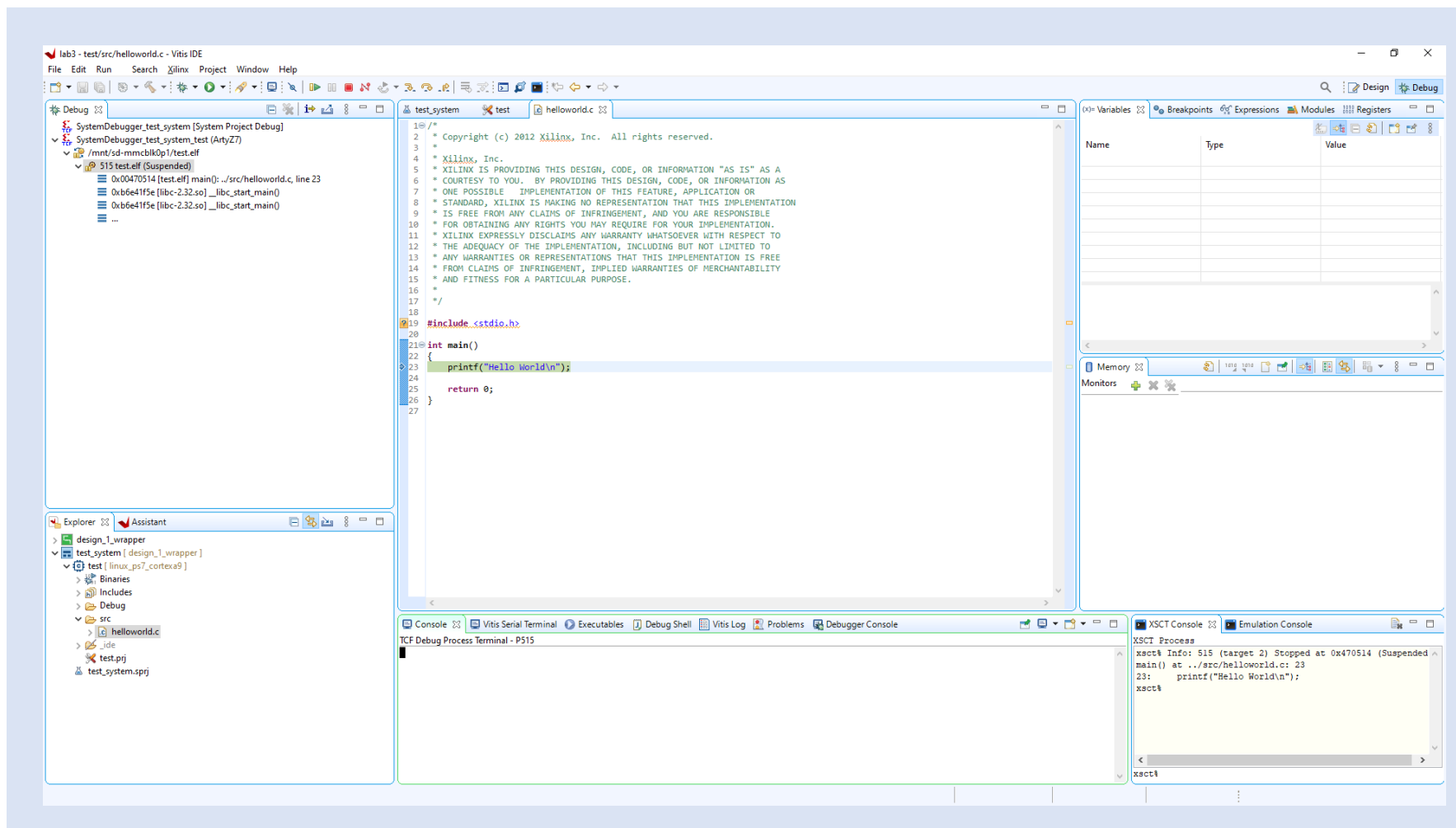
Lab 3: Creating Embedded Linux Solutions

Step 23 – Click OK when the connection is successful, click OK on the target connection and the click debug



Lab 3: Creating Embedded Linux Solutions

Step 24 – The application will be downloaded to the target and paused for execution, press the run button



Lab 3: Creating Embedded Linux Solutions

Step 25 – Click on the run button and you will see the hello world appear in the console.

