**Distributor ID: Ubuntu**

**Description: Ubuntu 14.04.5 LTS**

**Release: 14.04**

**Vivado v2016.2 (64-bit)**

Download link: <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html>

**Xilinx USB Cable Drivers**

1. Disconnect all Xilinx USB cables from the host computer.

2. Open a shell or terminal console.

3. Download install\_drivers.tar.gz from: secure.xilinx.com/webreg/clickthrough.do?cid=103670

4. Extract the driver script and its support files by typing: tar xzvf install\_drivers.tar.gz.

The extraction creates a directory named install\_drivers in the current directory.

5. Navigate to the install\_drivers directory by typing: cd install\_drivers.

6. Run the script by typing: ./install\_drivers.

7. When the installation is complete, connect the Xilinx USB cable to the desired USB port. If the STATUS indicator on the cable illuminates, then the driver installation completed successfully.

**CRATE’S firmware: via LNLS – Last updated on Jan/17.**

**AFC’s firmware: Commit 7c66824eb4311ad4d8c2d374f1cc355ebd5ee1fe from branch devel by LNLS GITHUB**

**Parts in code that must be changed for each computer:**

**AFC\_Loopback2:**

**interface\_loopback.py**

Must change “comando2” string for the current GTP.tcl location in BotaoGTP and the current RTM\_IO.tcl location in BotaoRTM.

BotaoGTP:

“comando2 = “/opt/Xilinx/Vivado/2016.2/bin/vivado -source PATH/TO/AFC\_Loopback2/GTP.tcl”

BotaoRTM:

“comando2 = “/opt/Xilinx/Vivado/2016.2/bin/vivado -source or PATH/TO/AFC\_Loopback2/RTM.tcl”

**GTP.tcl:**

Must change the line 4 and 8.

set\_property PROGRAM.FILE {PATH/TO/AFC\_Loopback2/example\_ibert\_7series\_gtp\_0.bit} [lindex [get\_hw\_devices] 0]

**RTM\_IO.tcl:**

Must change the line 4:

set\_property PROGRAM.FILE {PATH/TO/AFC\_Loopback2/design\_1\_wrapper.bit} [lindex [get\_hw\_devices] 0]

Line 5:

set\_property PROBES.FILE {PATH/TO/AFC\_Loopback2/debug\_nets.ltx} [lindex [get\_hw\_devices] 0]

**AFC\_Omnysis\_Test2:**

**Installing xdotool:**

sudo apt-get install xdotool

**TESTE\_AUTOMATIZADO\_DDR3/script2g.sh**

Must change the line 34:

xdotool type "xdownload 0 PATH/TO/AFC\_Omnysis\_Test2/TESTE\_AUTOMATIZADO\_DDR3/mem5000/project\_1/project\_1.sdk/Mem/Debug/Mem.elf"

Line 42:

xdotool type "xelf\_start\_address PATH/TO/AFC\_Omnysis\_Test2/TESTE\_AUTOMATIZADO\_DDR3/mem5000/project\_1/project\_1.sdk/Mem/Debug/Mem.elf"

**Libraries**

**Installing the ipmitool**

sudo apt-get update

sudo apt-get install ipmitool

**Installing the pyipmi**

sudo apt-get update

sudo apt-get install ipmitool

**Installing the pIp**

First of all, pip3 has to be installed (as we are using Python 3+). To install pip, securely download [get-pip.py](https://bootstrap.pypa.io/get-pip.py).

Then run the following:

python get-pip.py

Or in case of pip already installed, upgrade it to the latest version:

python -m pip install --upgrade pip

**scipy.interpolate**

**Library = SciPy Stack**

Install the SciPy stack packages with pip. It is recommend a *user* install, using the --user flag to pip (note: don’t use sudo pip, that will give problems). This installs packages for your local user, and does not need extra permissions to write to the system directories:

pip3 install --user numpy scipy matplotlib ipython jupyter pandas sympy nose

For user installs, make sure your user install executable directory is on your PATH. Here are example commands for setting the user PATH:

*# Consider adding this at the end of your ~/.bashrc file*

export PATH="$PATH:/home/your\_user/.local/bin"

**matplotlib.pyplot**

**Library = Matplotlib**

You might prefer to use your package manager. matplotlib is packaged for almost every major Linux distribution.

Debian / Ubuntu :

sudo apt-get install python-matplotlib

Fedora / Redhat :

sudo yum install python-matplotlib

**serial**

**Library = pyserial**

**There are also packaged versions for some Linux distributions:**

Debian/Ubuntu:

sudo apt-get install python-serial

sudo apt-get install python3-serial

Fedora / RHEL / CentOS / EPEL:

sudo yum install pyserial

Some distributions may package an older version of pySerial. These packages are created and maintained by developers working on these distributions.

**paramiko**

**Library = paramiko**

The recommended way to get Paramiko is to install the latest stable release via [pip](http://pip-installer.org/):

pip3 install paramiko

**PyQT4**

**Library = PyQT4**

**qt4-designer (must be qt4)**

**Debian/Ubuntu:**

sudo apt-get install python-qt4 qt4-designer

**sip (Must be 4.19.1 or above)**

Debian/Ubuntu:

<https://www.riverbankcomputing.com/software/sip/download>

And then go to the sipfile/doc/installation.html

**pyqt4**

Debian/Ubuntu:

<https://www.riverbankcomputing.com/software/sip/download>

And then go to the pyqtfile/doc/installation.html

**QT**

**Converting QT file to Python**

The process to convert a *.ui* QT file to a *.py* file, is through the command *pyuic4* from the library pyqt4 following:

pyuic4 (-x) input.ui -o output.py

Where input, output and are the conversion files.