

# Remote Access to Vivado for FPGA Development for Phase 4 Ground

2016-11-24 kb5mu

I've installed Vivado on an Ubuntu virtual machine, following the recommendations of Pavel Demin as documented at <http://pavel-demin.github.io/red-pitaya-notes/development-machine/>. For now, this is a standard WebPack installation, the same as you could easily install for yourself, suitable for development with the Red Pitaya. At some future time, this may become a fully licensed version, able to synthesize for the larger Xilinx parts.

This document describes how you can access this tool remotely, from a Mac, Linux, or Windows computer.

## Notes

If you do much at the VM's command prompt, you'll soon find that a lot of useful commands are missing from the VM. For example, there's no "man". These can be installed the usual way, with apt-get. You will have access to do so via sudo.

I don't know what happens if more than one vivado is running. Probably nothing good. We may need to arrange some kind of lockout if conflicts arise

The host machine is an HP Envy laptop with an Intel Core i7-6560U CPU at 2.2 GHz, quad core, 16 GB RAM. Its network connection is AT&T UVerse, 50/5 Mbps shared with the rest of my household's needs.

The intention is for this system to be up 24/7 for as long as it is useful. If the system isn't up when you need it, ping paul\_williamson on the Phase 4 Ground team on Slack.

## Expert Level Instructions

Email your SSH public key to [red-pitaya@mustbeart.com](mailto:red-pitaya@mustbeart.com) and ask to be added. Once I've done so, you can use SSH with trusted X11 port forwarding enabled to log in to the VM at ptw.dyndns.org, port 2222. The command to start the tool is simply "vivado".

You can skip the rest of this document.

## Step-By-Step from a Mac or Linux

### Step 0, Key Setup

Make sure you have a local SSH key. It would be in `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`. If you don't already have those files, do this:

```
ssh-keygen
```

and you should now have them. If that didn't work, you might have to install openssh first (Linux only).

### Step 1, if you have the login password

Authorize your key for SSH access on the VM. To do that, run:

```
ssh-copy-id -p 2222 red-pitaya@ptw.dyndns.org
```

and you'll have to authenticate, this once, with the login password for the user red-pitaya on the VM.

### Step 1, if you don't have the login password

Email your public key (that is, `~/.ssh/id_rsa.pub`) to [red-pitaya@mustbeart.com](mailto:red-pitaya@mustbeart.com), and wait for your access to be set up manually.

### Step 2, Starting the Local X Server

On the Mac, look in Applications/Utilities for XQuartz. If it's not there, download it from <http://www.xquartz.org> and install it there. Double-click on it to start your local X server.

On Linux, you already have an X server running, unless you're booted in text only mode. In that case, you'll need to figure out what to do to start your local X server. It might be as easy as:

```
startx
```

### Step 3, Connecting to the VM

Log on to the VM and establish port forwarding for X:

```
ssh -Y -p 2222 red-pitaya@ptw.dyndns.org
```

and answer yes when it asks about the new host. You should get an ordinary looking login with no weird error messages.

### Step 4, Run Vivado

Run the tool:

```
vivado &
```

and you should get the GUI for Vivado in an X window on your local desktop. The ampersand is optional and just runs vivado in the background so you can have your terminal session back.

### Step 5, Extra Terminal Sessions

If you want more terminals into the VM you can repeat step 3 from additional local terminal sessions. You don't need to use the `-Y` flag that enables X port forwarding, unless you want to run X programs from the new shell. (No X programs are currently installed other than vivado.)

## Step-By-Step from Windows

I'm assuming here that your Windows machine isn't already set up for X and SSH. If it is, you can skip ahead.

### Step 0, Install Networking Software

Download Xming from <https://sourceforge.net/projects/xming> and install it. The defaults are fine. Xming shows up in the tray at the bottom of the screen.

Download PuTTY from <http://putty.org/> and install it. Here too, the defaults are fine.

### Step 1, Key Setup

From the Start menu run PuTTYgen. Click Generate and wiggle the mouse as directed to create a new public/private key pair. Click Save public key and give it a filename. Click Save private key and give it a different filename. Leave PuTTYgen open for now.

### Step 2, Set Up an SSH Session

From the Start menu run PuTTY. Under "Host Name (or IP address)" put "red-pitaya@ptw.dyndns.org" (without the quotation marks) and for Port put 2222. In the Category column click on the + next to SSH under Connection and then click on Auth. Under "Private key file for authentication" give it the filename you used for the private key file above. Back in the Category column, under Connection and SSH, click on X11. Check the box labeled "Enable X11 forwarding". Now scroll back to the top of the Category column and click on Session. Make up a name and put it into the box under "Saved Sessions". Click Save.

### Step 3, if you have the login password

Click Open. You should see a terminal window and a password prompt. Put in the login password for the user red-pitaya, and you should be logged in. Now, on the VM, edit `~/.ssh/authorized_keys` (use nano or whatever editor you like). Get ready to insert text on a new line at the bottom of the file.

Go back to PuTTYgen and select everything in the little window labeled "Public key for pasting into OpenSSH authorized\_keys file" (by dragging the mouse over it) and hit Ctrl-C to copy it. You can close PuTTYgen now.

Go back to your PuTTY session and right-click to paste the key into the file. Save the file and exit the editor. Log out (by typing "exit"). You can close PuTTY now.

### Step 3, if you don't have the login password

Email your public key file (saved from PuTTYgen above) (NOT the private key file) to [red-pitaya@mustbeart.com](mailto:red-pitaya@mustbeart.com) and ask to have it added to `authorized_keys`. In that case, you'll now need to wait until it has been added manually.

### Step 4, Connecting to the VM

To start a session, first make sure Xming is visible in the tray. If it isn't, run Xming from the Start menu.

Then, run PuTTY and double-click on the saved session name you made up. You should get a terminal window, all logged in with no funny errors.

#### Step 5, Run Vivado

Run the tool:

```
vivado &
```

and you should get the GUI for Vivado in an X window on your local desktop. The ampersand is optional and just runs vivado in the background so you can have your terminal session back.

#### Step 6, Extra Terminal Sessions

If you want more terminals into the VM you can repeat step 4 from additional local terminal sessions.

## How It Works

Just for documentation.

There are two things that had to be configured to make this work. First, I set up a port forward in my main router to send port 2222 to the IP address corresponding to Moriarty, the host machine for the VM. Second, in the Network settings of the VM, I set a port forward from host port 2222 to VM port 22 (the standard port for SSH). The choice of 2222 was arbitrary, it just has to be the same in both port forwards and in these user instructions. Of course, this relies on my already having set up a dynamic DNS hostname with automatic IP address updating.

In the future, we may eliminate the VM from this configuration, running Vivado directly on a Linux host. These instructions should still be largely applicable.