**ASUS Tinker board Software Installation**

**Note: this is based on installing TinkerOS_Debian V1.8**

**1. Install Tinker OS (Debian)**

Download latest Debian image from https://www.asus.com/uk/Single-Board-Computer/Tinker-Board/HelpDesk_Download/

On the web page under **Drivers & Tools** select **Others** for the OS.

This will them change top show the number of files and the link to **Others** which you should click on.

Scroll down the page until you get to the section for **TinkerOS_Debian V1.8** and click on the link to download the OS.

Unzip image

Copy image to memory card using **dd** on Linux or **Win32DiskImager** on Windows.

Boot the Tinker board with it attached to your network. It will use DHCP to get an IP address.

Note that the display will work but may not be the correct size and the touch screen may not be working.

Connect to the Tinker board using ssh from another system. You can do all this on the Tinker board with a keyboard and mouse attached, but I find it is easier to ssh. (ssh linaro@<IP address>).

Update the system (warning this may take some time to complete and respond with **y** to any questions):

```
sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get install -f
```

Reboot the system.

**2. Install WiringPi GPIO package**

Either connect to the Tinker board using ssh or open a terminal window on the tinker board if you have connected a keyboard an mouse.

From the home directory create a directory called wiringPi and cd into it. Download the WiringPi package:

```
wget http://dlcdnet.asus.com/pub/ASUS/mb/Linux/Tinker_Board_2GB/GPIO_API_for_C.ZIP
```

This will create a file called GPIO_API_for_C.ZIP. Unzip the file using the command:

unzip GPIO_API_for_C.ZIP

run the build script using the command:

sudo ./build

**2. Install piHPSDR**

Either connect to the Tinker board using ssh from another system or open a terminal window on the Tinker board.

In the home directory (/home/linaro) download the binary package. This will overwrite any existing file with the same name. If you want to keep the previous version then you should rename the existing tinker.tar file before performing the wget command:

wget -N https://github.com/g0orx/pihpsdr/raw/master/release/tinker.tar

The /home/linaro directory should now contain the tinker.tar file.

Extract the files (will create directory pihpsdr):

tar xvf tinker.tar

Setup the system for piHPSDR:

cd pihpsdr
./install.sh

The first time you run install.sh it will install fftw3. You may be asked several questions to which you should simply hit the return key to accept the default answer.

The system is now ready to run piHPSDR.

You are now ready to run the application by double tapping (clicking) on the desktop icon.

Using a touch screen it is sometimes hard to do a double click so you can also start piHPSDR by tapping on **Menu** and then tapping on **Other** and then tapping on **pihpsdr**.

**3. Auto start piHPSDR when the system is booted**

If you want to set up the system so that pihpsdr starts automatically when the system is booted do the following from either an ssh connection or a terminal window screen:

nano ~/.config/lxsession/LXDE-pi/autostart

add the following line to the end of the file:

@/home/linaro/pihpsdr/start_pihpsdr.sh

and save the file.

Next time you boot the system it will start pihpsdr.

**4. Updating piHPSDR**

You can update the piHPSDR with the latest version of the software by repeating step 2 above. It will overwrite the pihpsdr binary and the shared libraries. Running the install.sh script will copy the updated libraries.

It will not overwrite or remove the FFTW3 wisdom file (wdspWisdom) or the radio configuration files which are stored as text files with the name made of the radio's MAC address and the suffix .props (i.e. 00-04-A3-6A-1E-60.props).

**ASUS Tinker board  Rotary Encoders amd Buttons**

Currently pihpsdr supports a high resolution optical rotary encoder for tuning and 2 low resolution encoders with a built in push switch for AF Gain/AGC Gain and RF Drive/RF Tune Drive.

The high resolution encoder is a 600 ppr optical encoder, and the low resolution encoders are 24 ppr (KY-040) with built in push switch.

Connection to the Raspberry Pi 2:

VFO: 600 ppr optical encoder.

Green – GPIO-17 (pin 11)
White – GPIO-18 (pin 12)
Black - Ground
Red - +3.3v

AF Gain/Mic Gain: 24 ppr encoder (KY-040):

CLK – GPIO-20 (pin 38)
DAT – GPIO-26 (pin 37)
SW – GPIO-25 (pin 22) (LOCK)
+ - +3.3v
GND - Ground

RF Drive/Tune Drive: 24 ppr encoder (KY-040):

CLK – GPIO-16 (pin 36)
DAT – GPIO-19 (pin 35)
SW – GPIO-08 (pin 24)
+ - +3.3v
GND - Ground

AGC: 24 ppr encoder (KY-040):

CLK – GPIO-04 (pin 7)
DAT – GPIO-21 (pin 40)
SW – GPIO-07 (pin 26)
+ - +3.3v
GND - Ground

All the following SPST momentary push to make buttons have one side connected to ground:

Band Up/Down - GPIO-13 (pin 33)
Band Stack Up/Down - GPIO-12 (pin 32)
Mode Up/Down - GPIO-06 (pin 31)
Filter Up/Down - GPIO-05 (pin 29)
Noise Up/Down - GPIO-24 (pin 18)
AGC Up/Down - GPIO-23 (pin 16)

| | |
|---|---|
| MOX/Tune | - GPIO-27 (pin 13) |
| Function | - GPIO-22 (pin 15) |