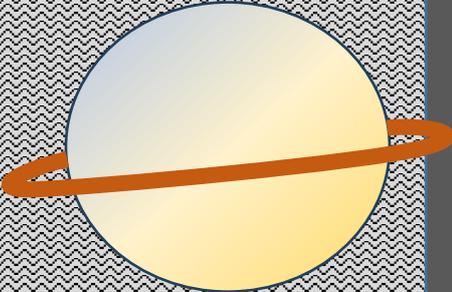


openHPSDR



ANAN-G2

User Guide

OpenHPSDR radio
With Saturn SDR



This document contains the words Apache, ANDROMEDA, SATURN, ANAN in reference to the Apache Labs Transceiver products.

<http://www.apache-labs.com>

In cooperation with G8NJJ, N1GP, VK6PH, NRØV, W5WC, K5SO, LA2NI, and the OpenHPSDR Hardware and Software Projects <http://openhpsdr.org>

All images and manufacturer data are copied here with permission of the owner Apache Labs

Note: Customer is responsible for FEDEX shipping and all local Customs, Tariff, VAT, Taxes and incidental charges within their delivery [address] Country. Please contact Apache Support support@apache-labs.com for details regarding your order or shipping charges.

Apache Labs LLC, Inc. - Declarations of Conformity

The ANAN-G2 complies with FCC Part 97 rules for the Amateur Radio Service. It has been confirmed by the relevant authorities that the ANAN-G2 DOES NOT require FCC certification. Under rule 97 Amateur radio equipment with the exception of amplifiers and scanning receivers are exempt from certification, however, must conform to rule 97 which states that harmonic and spurious emissions must be below 43dB of the transmitted out up to 30MHz and 60dB beyond that.

Contents

Contents

1	Key Features.....	5
1.1	Benefits.....	5
1.2	Architecture.....	6
1.3	Writing Style.....	7
1.4	SATURN Design Team.....	7
2	Controls and Connections.....	8
2.1	Powering On and Off.....	8
2.1.1	Radios with front panel controls.....	8
2.1.2	Radios with no front panel controls.....	8
2.2	Touchscreen.....	9
2.3	Front Panel Connections.....	10
2.4	Rear Panel Connections.....	11
3	Installation.....	14
3.1	Important Operation tips.....	14
3.2	Heat Dissipation.....	14
3.3	Quick Start.....	14
3.4	Getting Started with piHPSDR.....	15
3.5	Getting Started with Thetis.....	16
3.6	Getting started with the Raspberry Pi.....	17
3.6.1	Desktop Environment.....	18
3.6.2	Networking.....	18
3.6.3	SSH Command Line Access.....	19
4	Linux Applications.....	20
4.1	p2app.....	20
4.1.1	p2app Command Line Options.....	20
4.2	piHPSDR.....	20
4.3	Desktop “Helper” Applications.....	22
4.3.1	Flashwriter.....	22
4.3.2	Audiotest.....	23
4.3.3	Biascheck.....	24
4.3.4	AXI ReaderWriter.....	25
4.4	Pre-loaded software.....	25
4.5	Running Software Automatically after Power Up.....	26

5	Connecting Accessories.....	27
5.1	Linear Amplifier.....	27
5.1.1	PTT Out	27
5.1.2	ALC.....	27
5.1.3	Puresignal.....	27
6	Specifications.....	28
7	Maintenance	29
7.1	PC Maintenance	29
7.2	File Locations	29
7.3	Software Installation	30
7.3.1	Driver and SATURN Linux Software.....	30
7.3.2	piHPSDR	30
7.4	Reflashing firmware	31
7.5	Replacing the File System	32
7.5.1	Reflashing on-board eMMC memory	33
7.6	Block Diagrams.....	37
8	Support	38
8.1	Warranty.....	38
8.2	References.....	38

1 Key Features

The ANAN-G2 Amateur Radio Transceiver is a modern direct-down-conversion [DDC] receiver, and direct-up-conversion [DUC] transmitter, Software Defined transceiver covering the HF bands plus 6 Meters. It uses the SATURN FPGA board with integrated DSP processing and a Raspberry Pi processor. SATURN has a proud heritage from the TAPR Hermes Transceiver introduced in 2012. It is based upon the established 7000DLE mk2 RF, DSP and Raspberry Pi single board computer.

SATURN firmware updates can be easily installed via running a simple linux desktop application, eliminating the need for special mechanical or programming adapters. It is not necessary to open the ANAN-G2 cabinet to update the firmware.

The ANAN-G2 transceiver has been designed to work with several client Software Defined Radio applications. It can be used with:

- piHPSDR, running on the Raspberry Pi itself. piHPSDR was originally written by John Melton G0ORX and developed further by Christophe Wullen DL1 YCF and Rick Koch N1GP.
- THETIS Software - the latest SDR processing from the OpenHPSDR line and is based on the proven WDSP libraries. Thetis connects via the Ethernet interface, using the HPSDR "Protocol 2" interface.
- Both have full support for dual receivers, antenna diversity and PureSignal TX linearization.
- Other SDR programs can be used including those written for Windows®, Linux, MacOS/X®, Android, RaspberryPi, and other operating systems, using the HPSDR Protocol 2 interface.

1.1 Benefits

ANAN-G2 – Second to none!

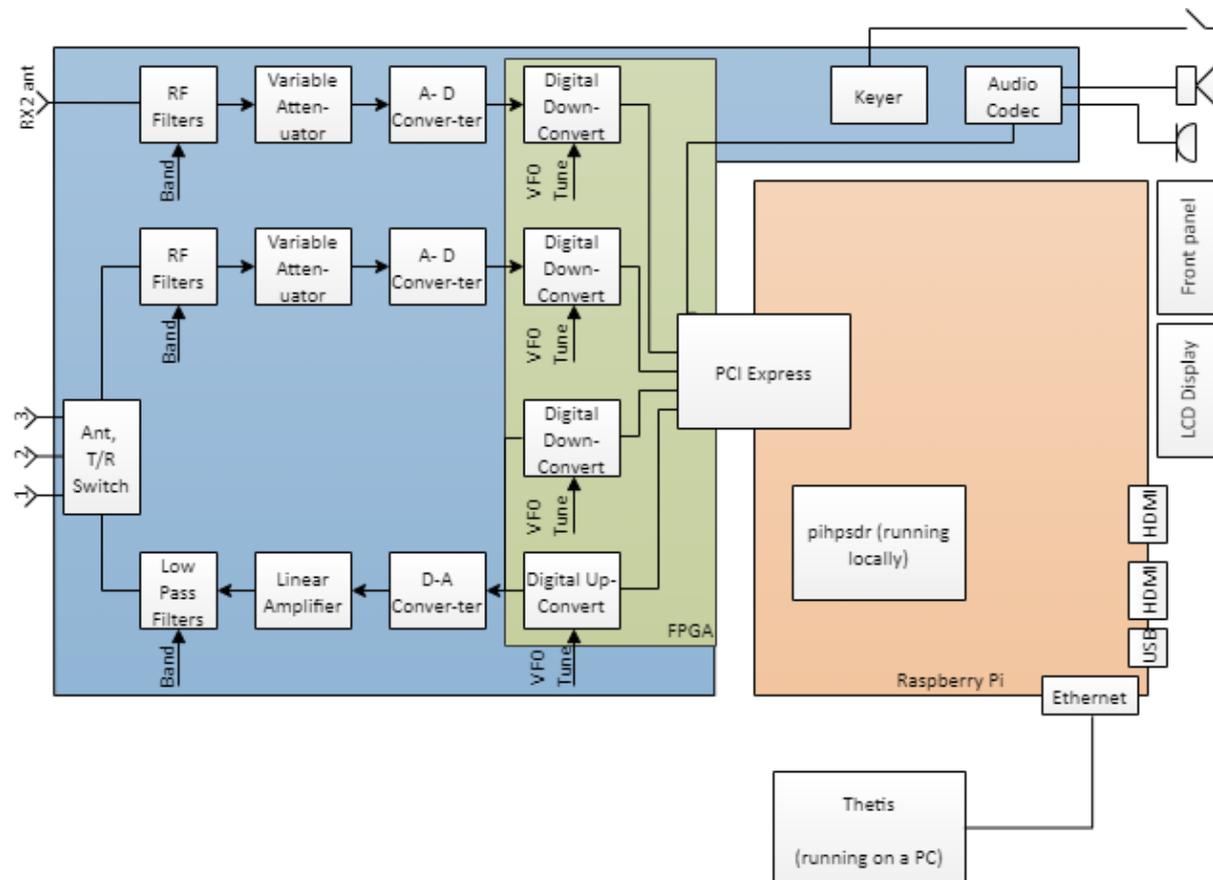
- Using Direct Down Conversion with an ultra low phase noise clock yields an RMDR of 116dB @ 2KHz separation, this means that close in weak signals will not be masked by the receiver's phase noise.
- The transmitter specifications are also off the chart, use of a new 16-bit DAC with an ultra low noise clock source results in transmit phase noise better than any other product available in the market.
- Use of LDMOS drivers and an optimized final Amplifier stage with adaptive Predistortion Algorithm (PureSignal) yields transmit IMD of better than 60dB, this is at least 20dB better than any Class A transmitter and over 30dB better than the competition.
- Use of two 16-bit phase synchronous ADCs allow for advanced applications such as Diversity reception for ultimate noise mitigation and effects of signal fading.
- Higher FPGA DSP Processing power
- Hardware capable of Remote operation (Future feature set currently not available)
- Stand Alone operation

The ANAN-G2 HF & 6M 100W SDR Transceiver offers top of the line performance in a rugged package, it is based on the work of the OpenHPSDR community.

The new PA board incorporates the following updates over the previous 100W generation:

- PA Gain redistribution for better IMD
- Thermally compensated Bias
- Current/Voltage/Temp Sensors
- Internal fan and option for external fan
- Independent BPFs and 6M LNAs for each ADC; option to ground ADC2 on TX
- Improved TX SNR and higher duty cycle TX

1.2 Architecture



1. Your radio hardware provides a transmit/receive path. There are two fully separate RX paths each with its own signal conditioning, digitising and processing.
2. Received signals are filtered, attenuated then digitised using two high performance analogue-to-digital converter. The A-D converters cover the whole HF band. Digital processing in the Field Programmable Gate Array (FPGA) converts a part of the HF band to zero centre frequency, and transfers the data to the PC via Ethernet. The bandwidth provided can be selected, but is usually in the range 48KHz – 1.536MHz.
3. A Raspberry pi4 Compute Module provides a local, high performance processor. The Pi runs desktop linux and can host many applications. A high performance PCI Express interface moves data between the FPGA and the Pi.
4. A client Software Defined Radio application (for example piHPSDR or Thetis) connects to the receiver data streams and provides further signal processing to reduce noise, to select the signals of interest and demodulate them. Its user interface provides a view of the band activity in the downconverted signal is shown, and controls to tune to the required signal and demodulate it successfully.
5. Received audio is routed to the speaker connectors. Additional PC connected speakers are also possible.
6. On transmit, the client application provides the initial signal processing to optimise the transmit signal. Audio processing is available to enhance a voice signal. A CW keyer is provided. Sampled data is sent to the radio hardware via PCI Express.
7. The radio hardware upconverts the TX signal to the required frequency, sets its signal level, amplifies and filters it. Amplifier linearization is available: a sample of the signal from the

linear amplifier is downconverted back through the receiver, and compared with the “intended” TX signal. The difference is processed by Puresignal algorithm to apply pre-distortion so that the final output from the linear amplifier has approximately 20dB lower spurious signal content than a “normal” HF transmission.

1.3 Writing Style

In this manual:

- A control / setting name is highlighted **bold**
- The user setting for that control is underlined.
- Menu > Equalizer means open the “Equaliser” setting on the program’s menu
- Menu > Setup > DSP > RX2 means open the setup form using Setup on the menu, then select the DSP tab, RX 2 sub-tab

So for example this instruction would indicate a gain control setting: Set the **RX1 AF** encoder to 30.

1.4 SATURN Design Team

SATURN is a new DSP board that draws upon previous work from the HPSDR family. Contributors to its content include :

Laurence Barker G8NJJ
Rick Koch N1GP
Abhishek Prakash
Dr. Warren C. Pratt, NR0V
Kjell Karlsen, LA2NI
Phil Harman, VK6PH
Doug Wigley, W5WC
Dr. Joe Martin, K5SO
John Melton, G0ORX/N6LYT
Adam Farson AB4OJ/VA7OJ

2 Controls and Connections

The front panel provides the first port of call to control your radio. Two version of the radio are currently available:

	<p>A version with no local display: this can be used with attached monitors for locally hosted client software such as piHPSDR. It can be used “headless” to connect to a PC client application such as Thetis.</p>
	<p>A version with a front panel 7” touchscreen display, rotary controls and pushbuttons. A touchscreen display provides a PC display from the PC control application; rotary encoders and pushbuttons provide access to the most commonly used controls.</p>
	<p>A different display version using the “Andromeda” display is also planned. This panel has more controls, and hands off processing for them to an Arduino microcontroller to reduce the load on the Raspberry Pi.</p>

2.1 Powering On and Off

The radio has a “soft” power button on the front panel. This is used both to turn power on, and to turn it off again. It is important to shut the power down properly using this button: otherwise the file system in your Raspberry Pi computer will become corrupted. The controls are different depending on whether you have front panel controls or not.

2.1.1 Radios with front panel controls

To power the radio **on**: press and release the button. A short press is sufficient. The power LED will illuminate.

To power the radio **off**:

1. Shut down the operating system. In PiHPSDR, click Menu and then Shutdown. You may also click on the Raspberry Pi icon at top left on the screen; select Logout from the menu then Logout on the Shutdown Options menu.
2. After the computer has completed shutting down, wait for approximately 10-15 seconds and then press and hold the power button for around 3 seconds until the power switches off. The front panel LED will go out.

2.1.2 Radios with no front panel controls

The procedure is slightly different as you may not have a computer display available.

To power the radio **on**: press and release the button. A short press is sufficient. The power LED will illuminate RED.

To power the radio **off**: press and release the power button. The LED will change to WHITE when the computer has shut down. Wait for approximately 10-15 seconds and then press and hold the button for around 3 seconds until power turns off completely; the LED will extinguish.



Power switch with Raspberry Pi running



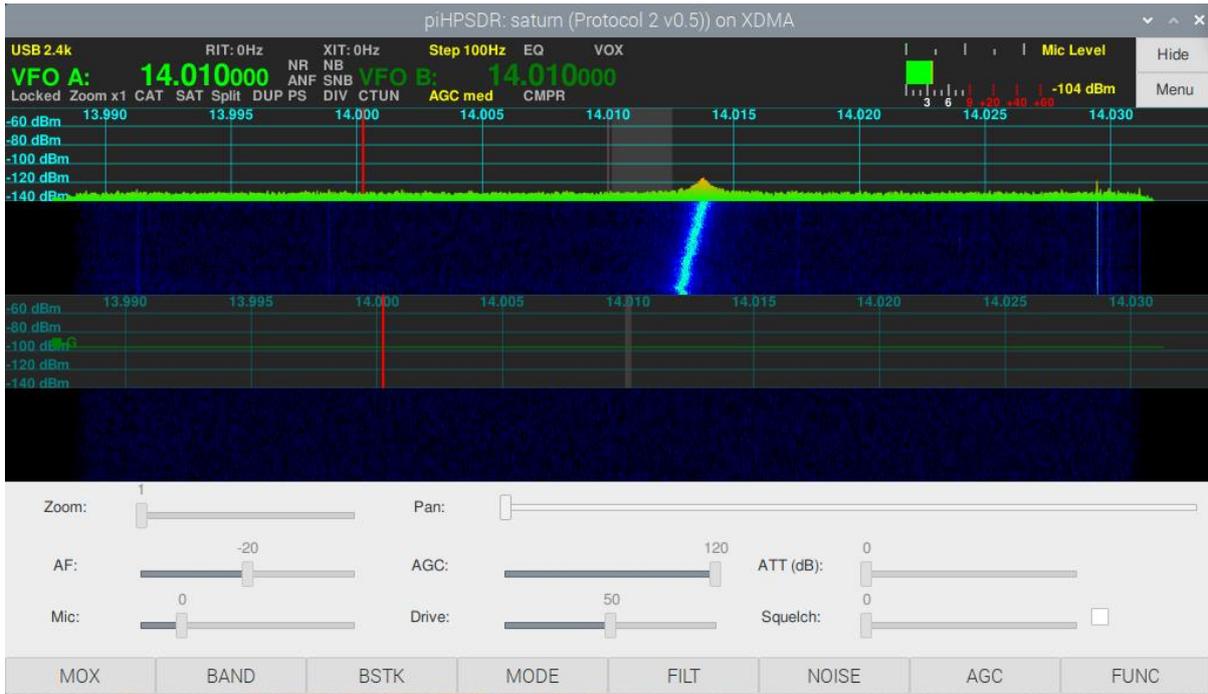
Power switch with Raspberry Pi Off

2.2 Touchscreen

The touchscreen, on those models with front panel controls, is a full function display with capacitive touch sensing. This is a full function Linux desktop; see Raspberry Pi documentation for more details. With an attached mouse and keyboard, it presents a full function desktop computer. Additional displays can be connected to the rear of the radio using HDMI connectors.

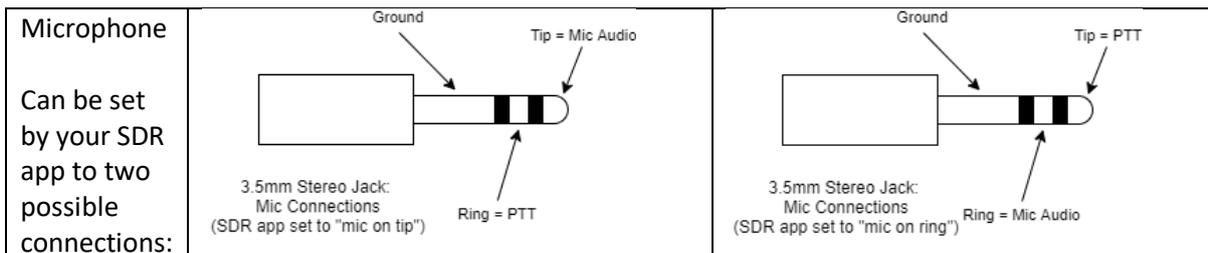
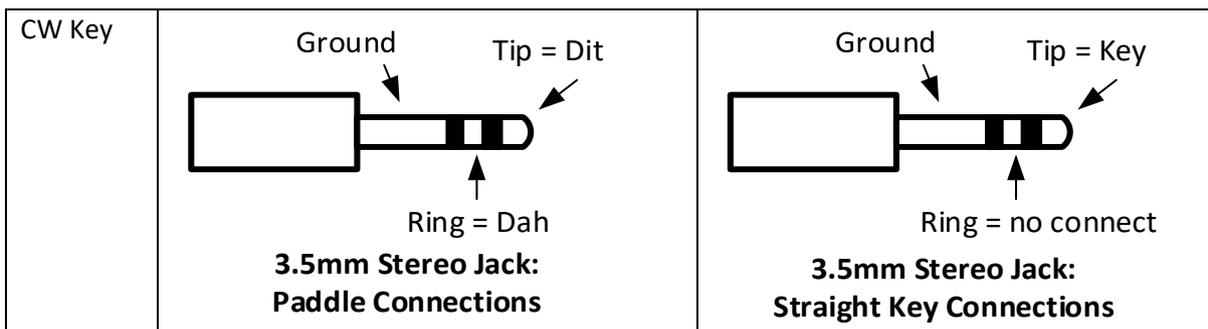


The front panel display can provide the front panel display for piHPSDR. In this mode it presents the front panel for your radio, allowing completely standalone operation.



2.3 Front Panel Connections

CW Key	3.5mm stereo jack socket. Connect a CW (morse) key: this may be a “straight” key or iambic keyer. Do not use a mono jack plug: TX mode will be continuously asserted!
Phone	Connect stereo headphones. This is a standard 3.5mm stereo jack socket. The internal speakers will be muted when connected.
MIC	3.5mm stereo jack socket.
USB	A USB socket connected to the Raspberry Pi; useful to connect memory sticks etc.



Use the “audiocheck” desktop app to test your microphone connection.

The microphone carries both microphone audio and PTT signals. Your DSP program (for example piHPSDR or Thetis) has controls to select how the signals are connected.

- PTT, if enabled, is achieved by connecting the PTT input to Ground
- A DC bias for the audio signal – needed by condenser “electret” microphones – can also be selected by your client app.

We are aware that some, but not all, PC headsets suffer loss of signal level when connected, because of the PTT connection. Using an external cable to “break” the wire to the ring will eliminate this.

2.4 Rear Panel Connections

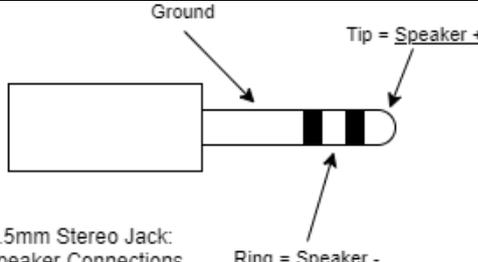


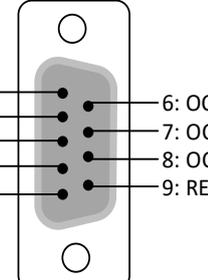
RF	GND	RF earthing stud
	ANT 1 - 3	3 independent TX & RX1 Antenna connectors
	RF Bypass	RX input bypassing RX filtering. Puresignal feedback input for an external PA
	EXT1	RX input bypassing the T/R switch
	XVTR In	Transverter RF input
	ADC2	RX2 Antenna input
	XVTR	Low power TX signal to Transverter
Power	13.8V DC	PowerPole® 13.8V DC, 30A power supply input
Raspberry Pi	Wi-Fi	Connection to the Wi-Fi antenna connector on the Raspberry Pi computer.
	USB	USB2 Type A connectors. A keyboard and mouse can be connected.
	LAN	Gigabit ethernet connection: connect to a PC to use Thetis, either directly or via your network.
	HDMI 1, 2	Full function HDMI outputs to connect external displays. Either connector may be used.
Aux Radio	OC Out	7 user configurable Open Collector Outputs
	Dig In	A 3.5mm jack with User digital inputs 1 & 2 Dig in 1: can be used as active low TX Inhibit Dig in 2: not currently supported
	Line in	A 3.5mm jack with Left and right TX audio line inputs

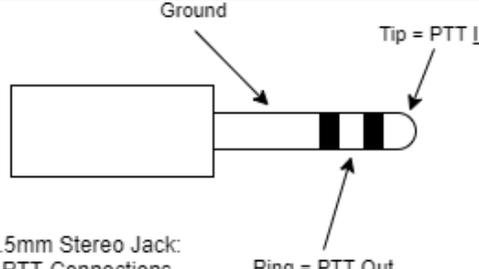
PTT In/Out	A 3.5mm jack with connections for PTT in (for example allowing an external PTT switch) and PTT out (for example to key an attached linear amplifier). PTT Input: short this signal to ground to initiate PTT PTT output: grounded if TX active (eg to key external equipment)
L SPK, R SPK	Connection for left and right speakers
XLR	Input for a balanced microphone
CW Key	Duplicates the front panel CW key input
HP	Duplicates the front panel headphone input.
Mic	Duplicates the front panel microphone input

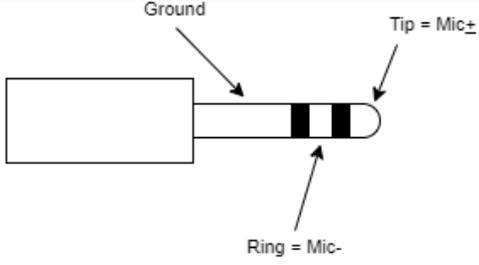
If you have a radio with front panel display, then the CW key, headphones and microphone connections are duplicates of those on the front panel.

CW Key	3.5mm stereo jack socket. Wiring as described for the front panel connection.
Phone	Connect stereo headphones. This is a standard 3.5mm stereo jack socket.
MIC	3.5mm stereo jack socket. Wiring as described for the front panel connections.

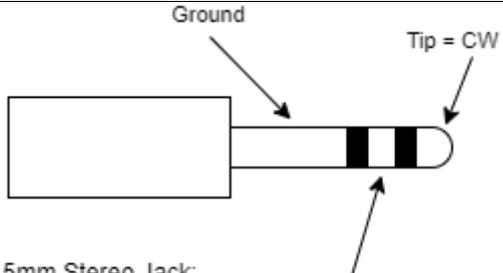
Rear panel speaker connectors	 <p>3.5mm Stereo Jack: Speaker Connections</p>	<p>External speakers should be wired between Speaker + and Speaker -.</p> <p>Use 4Ω or 8Ω, at least 2W speakers.</p>
-------------------------------	--	--

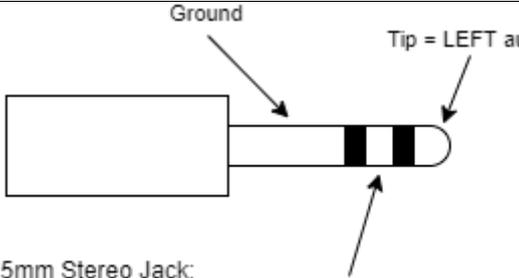
Open collector pinouts:		<p>GND = 0V</p> <p>OC1-OC7 are separate open collector outputs. Each connects to 0V when active.</p> <p>REF should be connected to the highest supply voltage used by OC1-OC7 (typically 5V or 12V)</p>
-------------------------	---	---

PTT In/Out	<p>Connections for both PTT in and PTT out are provided using a 3.5mm jack plug.</p> <p>PTT in: connect to ground to enter transmit mode.</p> <p>PTT out: this signal is grounded when in TX; use for keying linear amplifiers etc</p>	 <p>3.5mm Stereo Jack: PTT Connections</p>
------------	--	--

<p>Balanced Microphone</p>	<p>A connection for a balanced microphone input is available.</p> <p>This uses a 6.3mm jack on the radio, and a cable can be made to connect to an XLR socket.</p> <p>This will only work with balanced microphones designed for this kind of input.</p>	 <p>6.3mm Stereo Jack: Balanced Microphone Connections</p>
----------------------------	--	--

If you have a balanced microphone terminating in an XLR plug, you will need to make a cable to an XLR socket. Wire pin 1 to Ground, pin 2 to Mic+ and pin 3 to Mic-

<p>Dig In (two digital inputs)</p>	<p>CW Key: if connected to ground, this will key the TX when in a CW mode</p> <p>TX Disable: when connected to ground, this will signal to your SDR client app to request TX disable. (In Thetis, this input has to be enabled in the Setup - > TX menu).</p>	 <p>3.5mm Stereo Jack: Digital Input Connections</p>
--	---	---

<p>Line In (analogue input)</p>	<p>The line input can be used to connect other audio sources – for example audio processing equipment or digital mode signals.</p> <p>Note that only the LEFT audio channel is used with HPSDR radios.</p>	 <p>3.5mm Stereo Jack: Line Input Connections</p>
-------------------------------------	--	---

3 Installation

3.1 Important Operation tips

Do not cover rear or bottom ventilation holes. Proper air circulation throughout the ANAN-G2 Transceiver is necessary to prevent possible overheating and failure.

The radio is factory calibrated to 100W into a 50 ohm Dummy Load. Operating at higher levels will damage the radio. Please review CCW and ICAS specifications.

Data/AM/FM and other continuous duty modes must be operated at 30W or less to remain within the operational limits of the radio.

Input voltage must be 13.8vdc +/- 5%, well regulated at 30A Peak and proper POLARITY must be observed.

DISCONNECT the Antenna during Thunderstorms or when not in use.

Please ensure that you select the correct hardware model during software setup: ANAN-G2 100W.

Install the supplied rubber feet on the bottom of the case for proper cooling.

3.2 Heat Dissipation

ANAN-G2 uses a heavy duty Aluminium heat sink to dissipate heat with temperature controlled internal and optional external fan.

ICAS [100W] operation

SSB, CW and all ICAS modes [Intermittent Commercial/Amateur Service] 100W PEP

The Power Amplifier MOSFET transistor has characteristics that far exceed the needs of normal Transceiver operation.

3.3 Quick Start

Make sure that you have the following equipment and accessories:

- A 13.8v 30A Power Supply
- 50 Ohm Dummy load capable of at least 100W load capacity
- DC Power Cable (Supplied)
- Optional Keyboard/Mouse (either USB or Wireless)

With the Radio switched off:

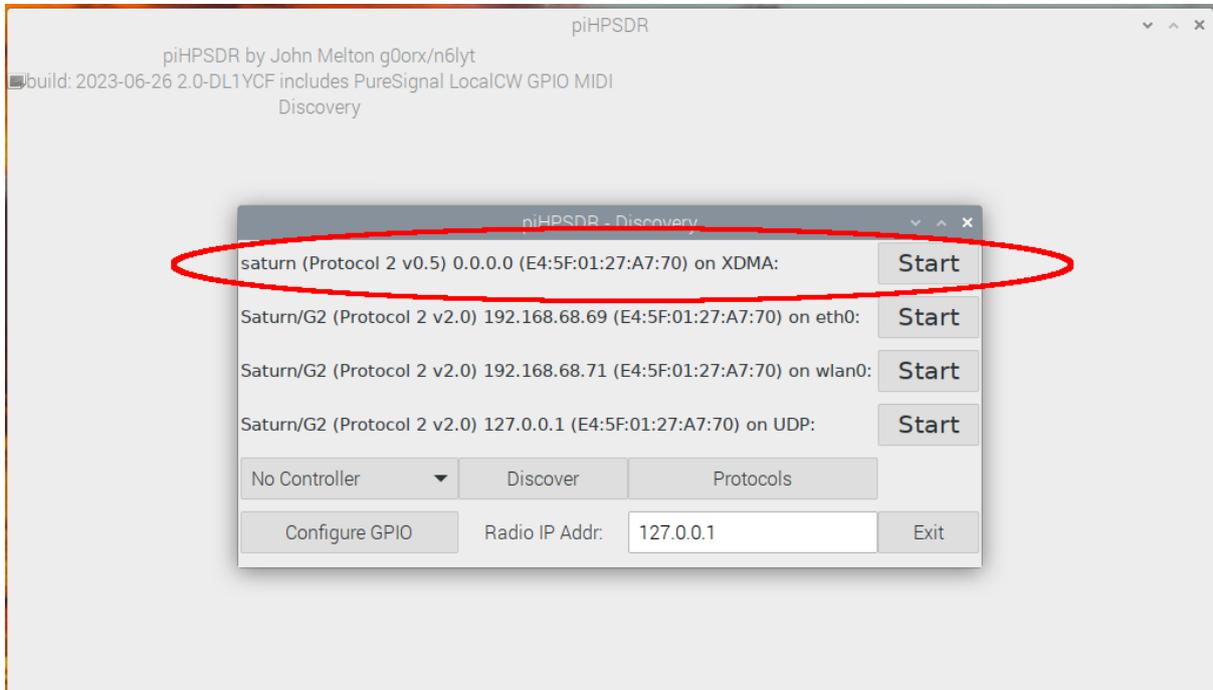
1. Connect the Keyboard/Mouse to one of the USB ports, if required
2. Connect the DC power cable to the PowerPole connector. Please ensure that the RED side corresponds with the Positive supply.
3. Connect the Dummy Load to the ANT1 connector.
4. Connect an ethernet network cable from the rear panel network connector to your home network.
5. Switch on the Radio: you should see the power button light up and then the internal PC booting. Once linux boots and is ready you are ready to launch your SDR client application (see next sections).
6. Once you have tested basic functionality you may remove the Dummy load and connect the Antenna ensuring good SWR before you are ready to go on Air!

It is possible to connect additional external displays if required using the rear HDMI connectors.

3.4 Getting Started with piHPSDR

piHPSDR will automatically launch on startup on the G2 with Display. If you have a radio without a front panel display, you can still run it but you will need an attached monitor, mouse and keyboard.

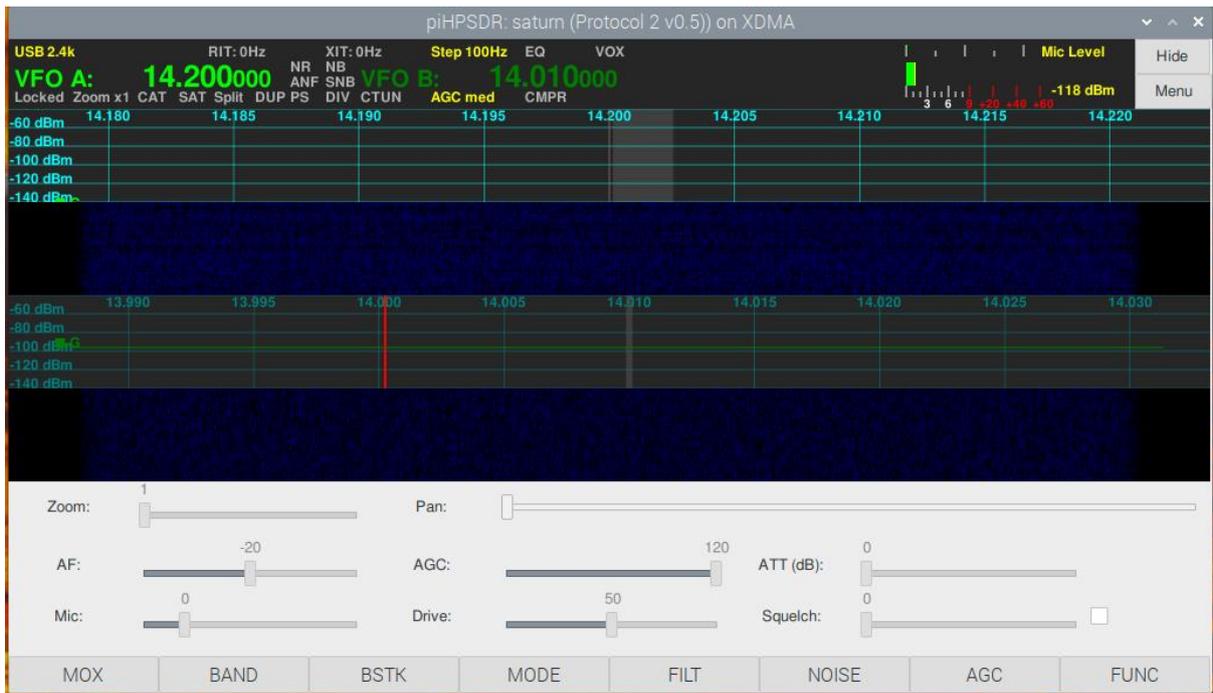
Double click the piHPSDR icon¹. The program will find the available radio hardware and display a selection form. Click “Start” on the tab showing a connection via XDMA:



If you have a front panel controller, it is configured using the drop-down box marked “no controller” here. Select **G2 Front Panel**

When you click Start, piHPSDR begins execution!

¹ If you don't have that icon, see section 7.5.2



There is a manual for piHPSDR in its software repository here:

<https://github.com/dl1cyf/pihpsdr/tree/master/release/documentation>

Use the Memu button to access the menus. Top level settings are in the Radio menu, including microphone settings.

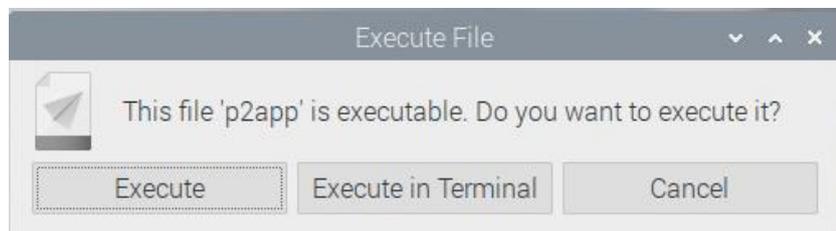
3.5 Getting Started with Thetis

To use Thetis, you will need your radio connected to the same network as the PC you will run Thetis from. You can use your pi's web browser to check the network connection first.

Once your radio is running (from the steps in section 3.3) you are ready to run Thetis.

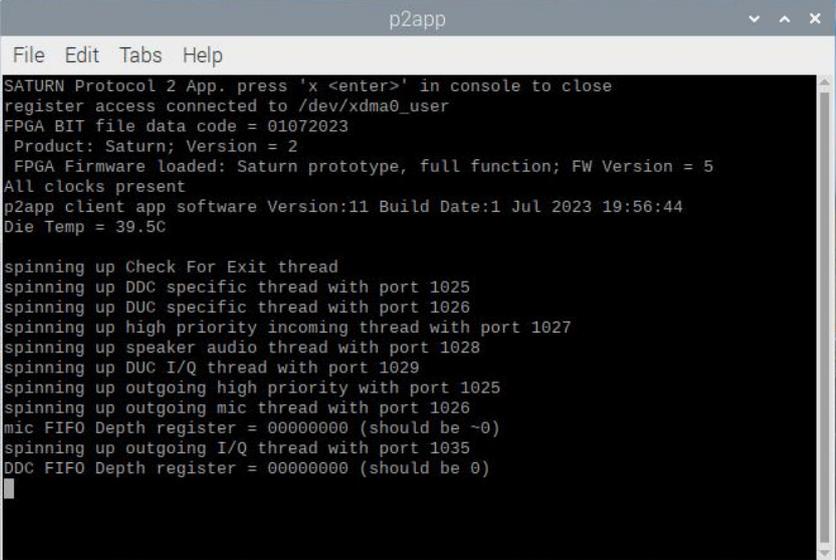
1. On the G2 with display, close piHPSDR and launch the p2app from the desktop by double clicking on its icon.

If you get this screen,
**select Execute in
 Terminal**



(if you want to remove this warning see section 3.6.4)

p2app running in a terminal window looks like this:



```
p2app
File Edit Tabs Help
SATURN Protocol 2 App. press 'x <enter>' in console to close
register access connected to /dev/xdma0_user
FPGA BIT file data code = 01072023
Product: Saturn; Version = 2
FPGA Firmware loaded: Saturn prototype, full function; FW Version = 5
All clocks present
p2app client app software Version:11 Build Date:1 Jul 2023 19:56:44
Die Temp = 39.5C

spinning up Check For Exit thread
spinning up DDC specific thread with port 1025
spinning up DUC specific thread with port 1026
spinning up high priority incoming thread with port 1027
spinning up speaker audio thread with port 1028
spinning up DUC I/Q thread with port 1029
spinning up outgoing high priority with port 1025
spinning up outgoing mic thread with port 1026
mic FIFO Depth register = 00000000 (should be ~0)
spinning up outgoing I/Q thread with port 1035
DDC FIFO Depth register = 00000000 (should be 0)
```

2. On G2 with NO display this does not need to be done and the p2app will launch automatically on startup.
3. Start Thetis as normal on your chosen PC
4. Click Setup -> HW Select on the menu.
5. Select **Radio Model** to Anan-G2
6. Click OK
7. Start Thetis using its on/off button. It will discover and connect to your radio.

There is a manual for Thetis available here:

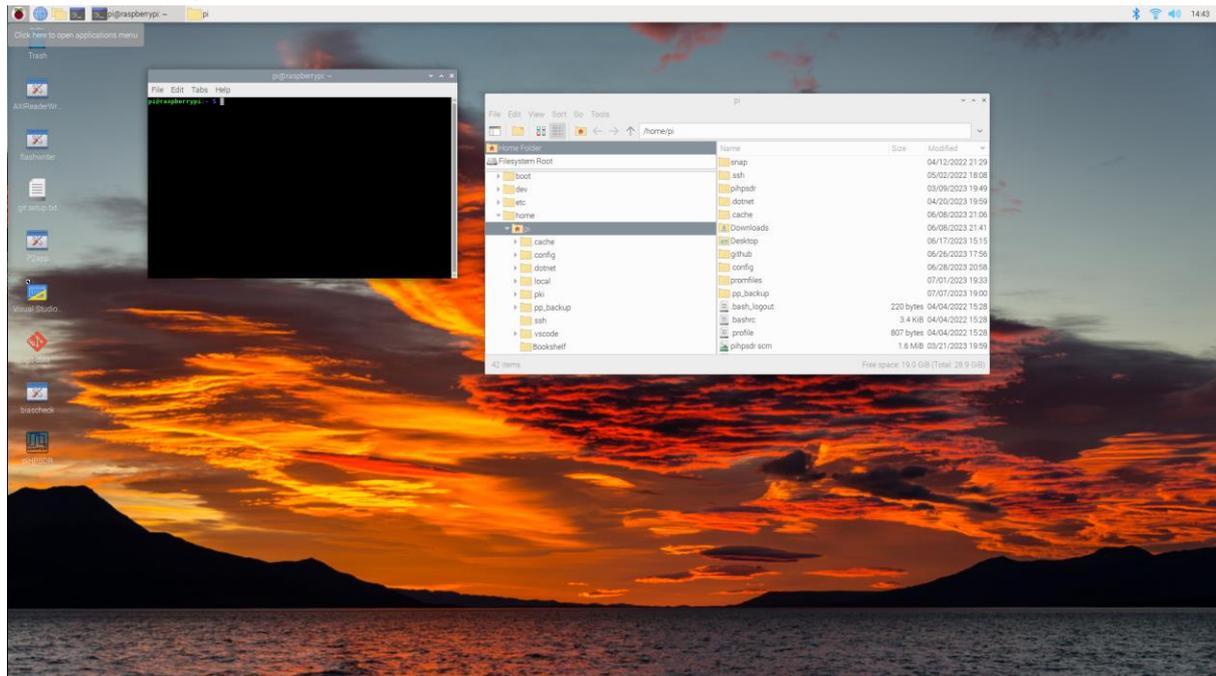
https://apache-labs.com/download_file.php?downloads_id=1020

3.6 Getting started with the Raspberry Pi

In normal use you will not need to interact much with the Raspberry Pi. However it is a full desktop computer and you may want to make use of its features from time to time.

The easiest way to do this is with a keyboard, mouse and monitor. If you have a radio with front panel display, it should already be displaying the Raspberry Pi desktop. If you don't have a front panel monitor, plug a PC monitor into one of the two rear panel HDMI connectors. It is also possible to interact with the operating system using a command line interface through a network connection, but this is only for those that like command lines!

3.6.1 Desktop Environment



With a monitor connected, there is a standard desktop environment. You use mouse and keyboard in the conventional way. On the desktop there is a command prompt, and a file manager window.

Along the top of the screen is the status bar with several icons:

	Opens a menu with tabs for a number of pre-loaded programs. The “preferences” menu allows access to a number of configuration options. There is a “logout” icon on this menu for safe shutting down.
	Opens a web browser.
	Opens a file manager window.
	Opens a command prompt window
	Bluetooth options
	Network connections. Hovering over this icon gives a summary of connections. Clicking on it allows you to choose a WiFi access point to connect to.
	Adjust computer system volume. Does not affect audio from the radio if connected to the rear panel speakers, or to the headphone sockets.
	Shows the time. Double clicking shows a calendar

3.6.2 Networking

If you have plugged your radio into an ethernet connection: it will have been allocated an IP address automatically and will be visible on your network. If you have not plugged it in, you may be able to establish a WiFi connection. WiFi is NOT recommended for use with PC SDR client applications such as Thetis.

The Raspberry Pi website provides some documentation on setting up your network:

<https://www.raspberrypi.com/documentation/computers/configuration.html#configuring-networking>

There are many other website will provide help with this.

<https://www.seeedstudio.com/blog/2021/01/25/three-methods-to-configure-raspberry-pi-wifi/>

<https://stevessmarthomeguide.com/configuring-networking-raspberry-pi/>

3.6.3 SSH Command Line Access

There is a tutorial on setting up command line access:

<https://raspberrypi-guide.github.io/networking/connecting-via-ssh>

3.6.4 Suppressing Warning When you Execute Programs

It is simple to remove the warning when you double click desktop icons:

1. Open your file manager from the taskbar
2. Select **Edit -> Preferences -> General**
3. Tick **Don't ask options on launch executable file**
4. Press **Close**

4 Linux Applications

ANAN-G2 is supplied with some linux applications that may be useful to you. These have been developed by the open source developer community. You may install further programs as you need them.

4.1 p2app

p2app is a command-line application that presents a “protocol 2” interface to PC client applications such as Thetis. p2app is a “middleware” program: it reads samples from the receiver and sends them to the remote client; it receives transmit samples and sends them to the transmit hardware. It provides audio in and out, and manages all of the register settings to control the radio hardware.

p2app is pre-installed. If you have an ANAN-G2 without front panel, it will be loaded automatically after power up. That means you can simply power up the radio hardware, and once the operating system is running you can run Thetis. Thetis will discover the connection to the radio across your network automatically.

If you need to run p2app you need to use a linux command line. Note that this is case sensitive!

Open a command window giving you a linux prompt. Then type:

- `cd github/Saturn/sw_projects/P2_app`
- `./p2app`

The program will start, and display the software and FPGA firmware version information.

4.1.1 p2app Command Line Options

To run for normal use:

```
./p2app
```

Or it can be run as `./p2app <optional arguments>`

Where <optional arguments> can be:

<code>-f <freq in Hz></code>	Turns on DDC test frequency source for development
<code>-i saturn</code>	Identifies board as “Saturn” (default)
<code>-i orionmk2</code>	Identified board as “Orion mk 2”
<code>-m xlr</code>	selects XLR microphone input
<code>-m jack</code>	selects 3.5mm microphone jack (default)
<code>-s</code>	skip checking for “x” button to exit (used when run at startup)

To exit: type `x<enter>`

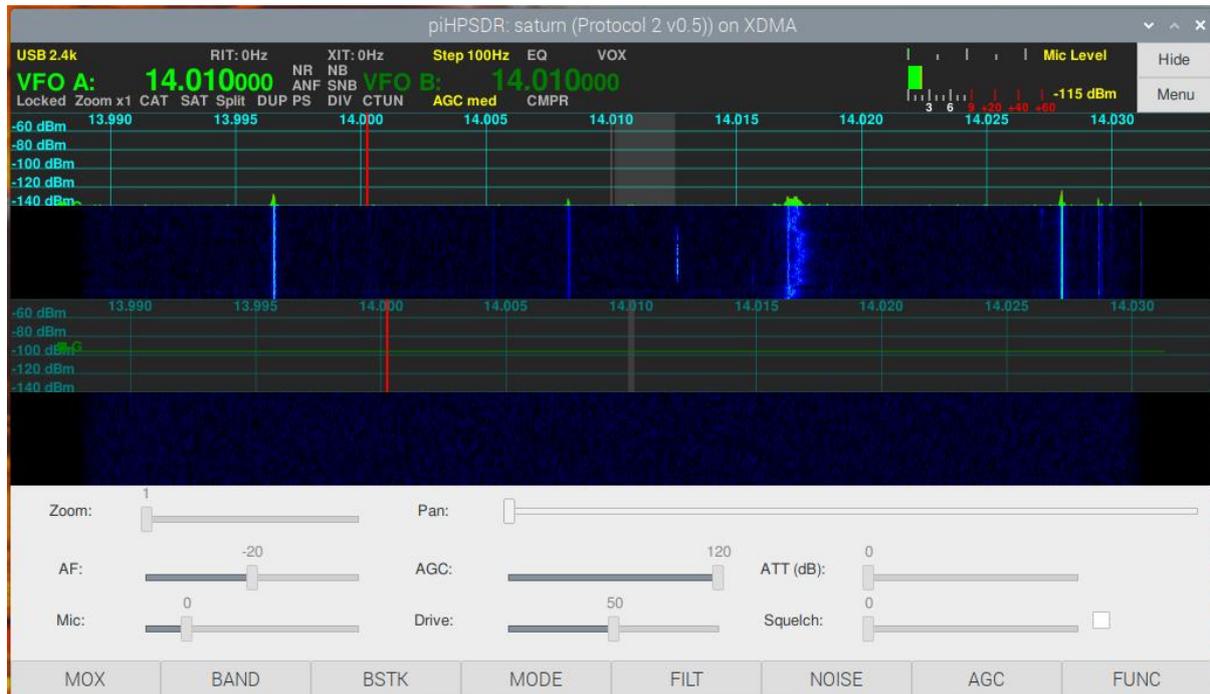
(note no text is displayed as you type; just do it!)

(note that both Thetis and piHPSDR can now select the balanced microphone input, so there should be no need now for the `-m xlr` option)

4.2 piHPSDR

piHPSDR is a desktop SDR client application written for the Raspberry Pi. This program enables a complete radio to be operated within your ANAN-G2 if you have a version with a front panel. If you

don't have the front panel you can plug in an external monitor, keyboard and mouse and use piHPSDR.



piHPSDR is pre-installed on your radio. If you have a model with a front panel display, it will start automatically.

piHPSDR will make the best use it can of the available processing on the Raspberry pi. 96KHz with one receiver or 48KHz with two receivers are the recommended sampling rates. If the rate is set too high gaps may appear in the audio stream.

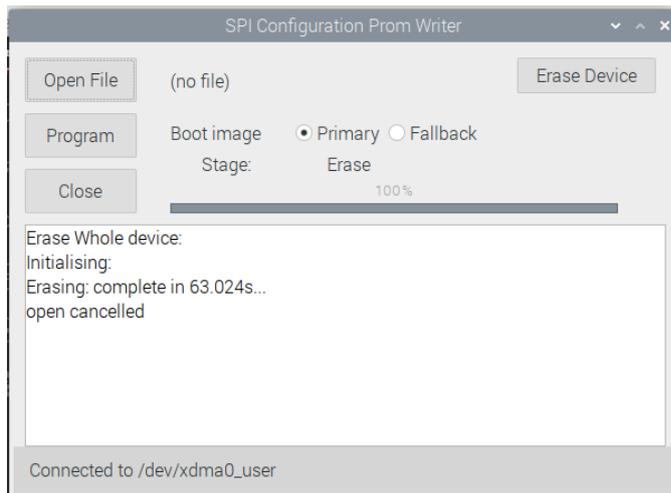
4.3 Desktop “Helper” Applications

4.3.1 Flashwriter

This is an application to program the FPGA configuration memory from a BIT file. You will need to run this program if new FPGA firmware is made available and you want to install it.

The program may already be installed onto your linux desktop. If not you can run it simply from a linux command prompt. Open a command window then type:

- `cd github/Saturn/sw_tools/flashwriter`
- `./flashwriter`



Usage:

- | | |
|--------------|--|
| Open File | Opens a dialog to select the file to be programmed into the memory,
The only format accepted is a .BIN file (a Xilinx binary format) |
| Boot Image | Selects which image to reprogram
Primary (this is the normal image to program)
Fallback (CAUTION: only reprogram if absolutely necessary) |
| Program | When pressed initiated programming; this may take 2 minutes or so. The steps are displayed and progress for each step is shown in the bar display. |
| Close | Close the application. |
| Erase Device | Erases the complete FPGA. Don't use this unless you know why you are doing it – this is not a normal programming stage! |

If your power supply fails during the programming process, the image will be corrupted. Your radio will still load, but using a “fallback” image. This is likely to be older FPGA code. You will need to repeat the process when power has been restored.

It is possible to reflash the “fallback” image although the need for doing that is rare. To do so you would select the **Fallback** button. The fallback image is over twice the size and takes a few minutes to program.

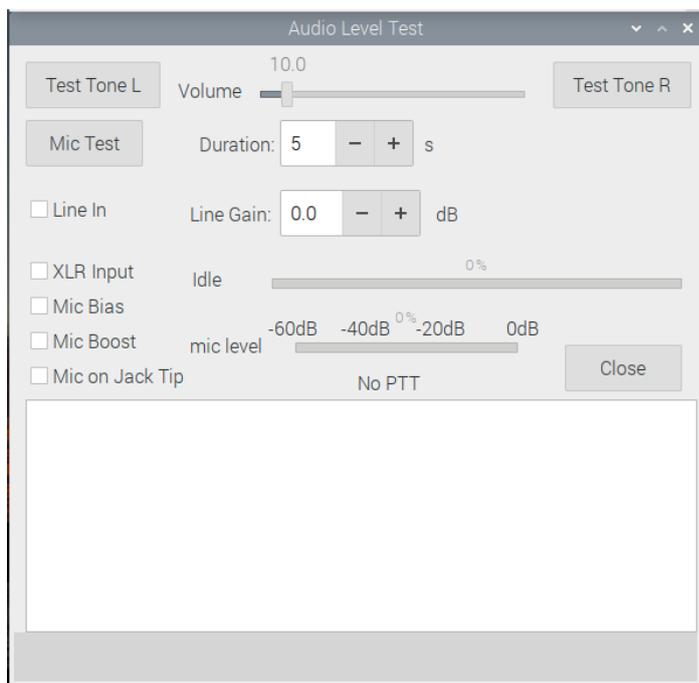
4.3.2 Audiotest

This is a simple application for checking audio inputs and outputs. For example it can be used to test if a new headset or microphone is working.

The program may already be installed onto your linux desktop. If not you can run it simply from a linux command prompt. Open a command window then type:

- `cd github/Saturn/sw_projects/audiotest`
- `./audiotest`

(Please note: p2app should not running at the same time or it will not function correctly. Follow the instructions in section 4.5 to terminate the version of p2app that runs after power up)



Usage:

- | | |
|----------------|--|
| Test tone L, R | Plays a test tone with selectable volume while the button is pressed into the Left or Right audio channels. Left channel gets a lower frequency tone. |
| Mic Test | Enables recording for the (adjustable) duration followed by playback of the recording. A progress bar shows how much time has elapsed, and the current operation (record/replay/idle) is indicated.

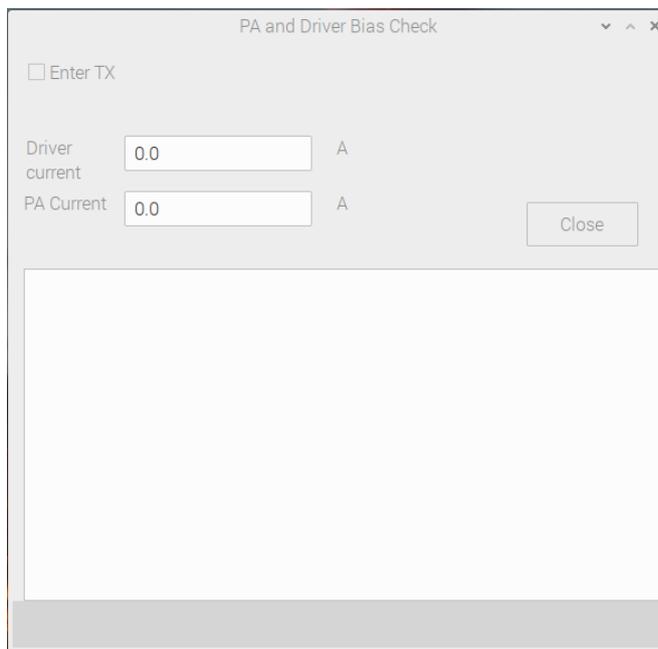
The mic level bar shows a log representation of signal level, with max signal at the right. It is scaled from -60dB to 0dB. |
| Line In | If checked, selects the Left line input instead of the microphone. (You still need to press the Mic Test button!) |
| Line in Gain | Selects the gain for the line input channel, between +12dB and -34.5dB. |
| XLR Input | If checked, selects the balanced microphone input instead of the 3.5mm jack. |
| Mic Bias | If selected, applies bias to the 3.5mm mic input. |
| Mic Boost | If checked, increases the audio gain of the Codec by 20dB. Typically needed for dynamic microphones and not for electret ones. |

Mic on Jack Tip	Selects the microphone signal to be on the tip of the 3.5mm jack. If not selected the ring terminal will be used.
Close	Close the application.

4.3.3 Biascheck

This is a simple application for setting up driver and PA bias currents. It puts the radio into transmit with no signal present, so the potentiometers to adjust bias can be set. **This is not intended for user operation.** The program may already be installed onto your linux desktop. If not you can run it simply from a linux command prompt. Open a command window then type:

- `cd github/Saturn/sw_projects/biascheck`
- `./biascheck`



Usage:

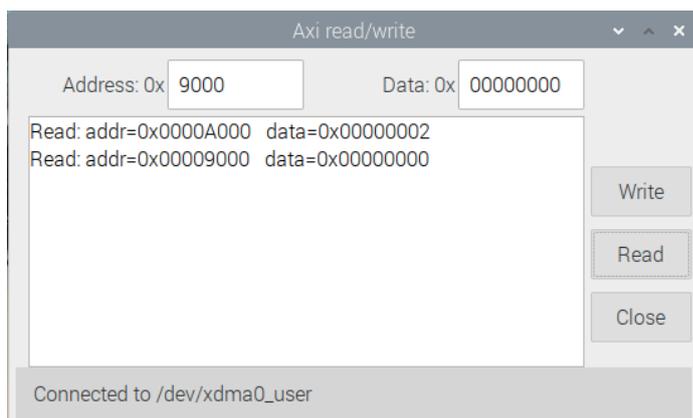
Enter TX	When checked, the radio enters TX with no signal present. Any current in the driver stage or PA is purely the bias current.
Driver Current	Shows the current into the driver stage on the Saturn board. Each bias potentiometer on Saturn should be fully anticlockwise initially; advance the 1 st to obtain 0.1A reading then advance the second until 0.2A is displayed.
PA Current	Shows the current into the PA stage on the RF board. Each bias potentiometer on the RF board should be fully anticlockwise initially; advance the 1 st to obtain 1.0A reading then advance the second until 2.0A is displayed.
Close	Close the application.

4.3.4 AXI ReaderWriter

This is a simple application for register reading and writing. **It is intended for development, not for normal use** and it is very possible to crash the raspberry Pi by incorrect usage!

You can run it simply from a linux command prompt. Open a command window then type:

- `cd github/Saturn/sw_tools/axi_rw`
- `./axi_rw`



Usage:

Address	Hexadecimal address to be accessed. Should be in the range 0 to 0x1FFC, in steps of 4
Data	Hexadecimal data to be written, or read from a register.
Write	Causes the data value entered to be written into the register with specified address
Read	Reads data from the register at the specified address.
Close	Causes the data value entered to be written into the register with specified address

Note that if the address does not correspond to a valid register, this may hang the computer!

4.4 Pre-loaded software

ANAN-G2 has some software pre-loaded at power up to make your operating experience easier. The software depends on the version.

- If you have a radio with a front panel display: piHPSDR will be pre-loaded.
- If you have a radio without a front panel display: p2app will be pre-loaded.

piHPSDR will eventually include the “p2app” functionality.

4.5 Running Software Automatically after Power Up

It is possible to set programs to run automatically after power up. This is a standard linux function and seek advice from websites covering the Raspberry Pi if in doubt. These instructions cover running the p2app application, but others can be started similarly.

Please be careful when doing this: it requires administrator permission, and can cause harm if incorrectly completed. The settings have been stored in file `/etc/xdg/lxsession/LXDE-pi/autostart` which can be edited with a text editor. As shipped it will be configured as follows:

With display:	@lxpanel --profile LXDE-pi @pcmanfm --desktop --profile LXDE-pi @xscreensaver -no-splash @/home/pi/github/pihpsdr/pihpsdr
No display:	@lxpanel --profile LXDE-pi @pcmanfm --desktop --profile LXDE-pi @xscreensaver -no-splash @/home/pi/github/Saturn/sw_projects/P2_app/p2app

Use your favourite text editor to edit the file, with root access for example:

```
pi@raspberrypi:~ $ sudo geany /etc/xdg/lxsession/LXDE-pi/autostart
```

If you need to terminate the auto-running app you need to find its process id, then kill it

```
pi@raspberrypi:~ $ ps -ax | grep p2app
```

```
741 ? Sl 30.00 ./p2app
```

```
pi@raspberrypi:~ $ sudo kill 741
```

5 Connecting Accessories

5.1 Linear Amplifier

5.1.1 PTT Out

Most modern Linear Amplifiers have a connection for keying the amplifier. The radio includes a back panel 3.5mm jack connection for this purpose.

The PTT Out feature is best used to switch 25vdc and <1A. Exceeding these values may damage the PTT output circuit.

How to test your PTT Linear Amplifier circuit BEFORE connecting your Apache Transceiver:

1. Power your Linear, connect the control cable to your linear but do not connect the other end to your Apache Transceiver.
2. Using a DC Voltmeter, measure the voltage between the centre pin and the outer connection of the cable. This must be less than 25vdc.
3. Connect a DC Ammeter between the centre pin and the exciter connection of the cable. This should ACTIVATE the Linear Amplifier and give you a clear measurement of the current requirements. The ANAN-G2 PTT circuit should not draw more than 1A.
4. If either of these values are out of the required range, then use an external keying interface. There are many keying interfaces available from many sources such as MFJ, Elecraft, DX Engineering.

Important: Connecting a Linear Amplifier that does not meet the above voltage and current requirements is likely to cause damage to the Apache Transceiver linear switching transistor and extreme cases, serious damage to other parts. Damage to the radio caused in this manner is not covered by Warranty.

Linear Amplifier Keying ISOLATION adapters

<http://www.dxengineering.com/search/product-line/kd9sv-products-sv-kr-keying-relays>

<http://www.ameritron.com/Product.php?productid=ARB-704>

<http://www.sultantronics.com/product.html>

5.1.2 ALC

ANAN-G2 does not offer ALC control for an external amplifier. It is important to set the drive level so that the amplifier is not driven into saturation. Drive shows a % of full power; setting the PA gain in the setup form is the best way to achieve this.

5.1.3 Puresignal

To get Puresignal linearization your amplifier will need a coupler to take a sample of the TX signal and provide it back to ANAN-G2. It should be just above 0dBm at normal operation. The signal should be connected to the RX Bypass input connector. That input needs to be enabled on the Antenna tab of your client SDR application.

6 Specifications

General Specifications:

- Architecture: Direct Sampling DDC/DUC Transceiver
- Interface: PCI-Express to Raspberry Pi; Ethernet from Raspberry Pi
- Phase Noise (Clock): -149dB @ 10Khz
- TCXO Stability (Typical): +/- .1 PPM
- Modes: CW, SSB, NFM, AM, Digital
- Antenna Ports: Three BNC 50 ohms Software Configurable Ports, Three BNC ADC1 Bypass and Loop in ports, One BNC for RX2
- Frequency Resolution: 1 Hz

Electrical Specifications:

- 13.8v DC @ 25A, 3A Receive/25A Transmit

Mechanical Specifications:

- 5Kg (approx. Weight)
- Dimensions: 365mm (L) x 310mm (W) x 140mm (H) (Not including extrusions)
- Rugged Aluminium Extrusion with internal fan.

Receiver Specifications:

- Receiver Architecture: Direct Down Conversion
- Dual 16 bit Phase Synchronous ADCs
- Independent filter banks for each ADC
- 6M LNAs
- Frequency Coverage: 9Khz to 60MHz
- Attenuator: 1-30dB step attenuator
- Reciprocal Mixing Dynamic Range (RMDR): 116dB @ 2Khz offset
- Receiver Phase noise: -149dB @ 10Khz
- Image rejection: 90dB
- Hardware support for 10 independent receivers assignable to either ADC

Transmitter Specifications:

- Transmitter Architecture: Direct Up Conversion
- DAC Resolution: 16 bit
- RF Output Power: 100W PEP SSB, FM, RTTY, Digital; 1-30W AM, 100W CW
- IMD: IMD3 typically -68dB below PEP @ 100W output on 20M
- Harmonics: Typically better than -43dBc on HF and -60dBc on 6M
- Carrier and Opposite Sideband Suppression: Better than -80dBc
- Transverter IF Output: 0dBm to 15dBm

Front Panel Specifications:

- 7" touchscreen display, 1024x600 pixels
- 1 high resolution VFO encoder (480 steps per revolution)
- Dual shaft encoders for analogue controls, with user programmable functions
- User programmable pushbuttons

7 Maintenance

7.1 Computer Maintenance

A Raspberry pi4 Compute Module provides the processing within the radio's chassis. This is a normal linux based Raspberry Pi and requires normal computer maintenance and precautions. It is the user's responsibility to ensure that it is adequately protected from viruses or other malicious attacks, and that it is backed up adequately.

A mouse and keyboard can be connected to the front or rear panel USB sockets using a simple hub if required.

7.2 File Locations

Your linux filesystem is set up in the normal way. There are some locations you will need to be aware of.

As a general guide, **do not make changes to other folders in the filesystem unless you know what you are doing**. Linux can be very unforgiving!

/home/pi	This is the "home" location for your account. You can store any files you want in this location. If you take screenshots using the Prtsc key, the image files will be put here Configuration files for some of your applications including pihpsdr will be stored here.
/home/pi/Desktop	Copies of files that are on your computer desktop.
/home/pi/Downloads	Any files you download using a web browser will be located here.
/home/pi/github/Saturn	This is a copy of the Saturn software repository
/home/pi/github/Saturn /FPGA	Files for the FPGA design are in this folder. You will find the files to reflash the FPGA here: saturnprimary.bit (the primary FPGA image file) saturnfallback.bit (the fallback configuration file)
/home/pi/github/Saturn /desktop	Desktop shortcut files for some of our applications. Copy these to your desktop to make them available as clickable icons.
/home/pi/github/Saturn /linuxdriver	This is the code for the PCI Express device driver
/home/pi/github/Saturn /project_documentation	Useful documentation
/home/pi/github/Saturn /sw_tools	Useful software tools are located here
/home/pi/github/Saturn /sw_projects	Radio related software projects are located here
/home/pi/github/Saturn /sw_projects/P2_app	This is the location for "p2app", the server program that makes samples available to remote client applications such as Thetis.
/github/wdsp	This is where the DSP code used by pihpsdr is located
/github/pihpsdr	This is a copy of the pihpsdr software repository
/github/Waveshare-DSI-LCD	Drivers for front panel touchscreen
/media	If you plug in a USB memory stick, the files will be mounted here.

7.3 Some common Linux Commands

This is not a substitute for reading a proper linux reference but may help get you started!

Remember linux commands and filenames are case sensitive.

Command	Function
ls	List files
ls -a	List files, including hidden files
cd <foldername>	Change directory to folder <foldername>
cd ~	Change to home folder
rm <filename>	Delete specified file
rmdir <foldername>	Delete named folder (which must be empty)
./<filename>	Execute the file <filename> in the current folder
top	Displays the CPU usage of current programs. Ctrl-c to exit. (This is a good way to see if p2app is running because it will show up!)
ps -ax	Shows a list of all executing processes
geany >><filename>	Opens the geany test editor in the named file
shutdown	Shuts down the computer

7.4 Software Installation

If it is necessary to re-install any of the supplied software, please follow these steps.

7.4.1 Driver and SATURN Linux Software

1. Open a command window
2. Type `cd github/Saturn`
3. Type `git pull`
4. Type `cd sw_projects/P2_app`
5. Type `make`

There are instructions to set up the files for the raspberry Pi from scratch provided in the Saturn repository, project_documentation folder. The Saturn repository is located at:

<https://www.github.com/laurencebarker/Saturn>

7.4.2 piHPSDR

piHPSDR can be copied and installed from the repository: <https://github.com/dl1ycf/pihpsdr>

Instructions are available in the release/documentation folder.

NOTE: The github repository URL for piHPSDR has been changed from

<https://github.com/n1gp/pihpsdr>

To

<https://github.com/dl1ycf/pihpsdr>

To see which URL yours is from:

1. Open a command window
2. Type `cd github/pihpsdr`
3. Type `git remote -v`

If your URL is <https://github.com/n1gp/pihpsdr> you will need to change it.

To change it (**which only needs to be done once**):

4. You should still be in directory: github/pihpsdr
5. Type make clean
6. Type git checkout master
7. Type git pull
8. Type git remote set-url origin <https://github.com/dl1ycf/pihpsdr.git>
9. Type git pull

7.5 Missing Desktop Icons

If you are missing desktop icons you can recreate them simply.

7.5.1 Icon for p2app

1. Open a file editor. Your Raspberry pi will have several, and an editor called “geany” can be loaded from the programming menu.
2. Enter (or paste) the text below into your editor window

```
[Desktop Entry]
Name=p2app
Comment=Saturn FPGA Register Read/Write
Icon=/usr/share/pixmaps/obconf.png
Exec=/home/pi/github/Saturn/sw_projects/P2_app/p2app
Type=Application
Encoding=UTF-8
Terminal=true
Path=/home/pi/github/Saturn/sw_projects/P2_app
StartupNotify=false
Name[en_US]=P2app
X-KeepTerminal=true
Comment[en_US]=Saturn protocol 2 server
```

3. Use **save as** to save it to your desktop with filename p2app.desktop
4. Using your file browser, open your /home/pi/Desktop folder
5. Select file p2app.desktop (may shows as p2app with no extension)
6. Right click and select **properties**
7. Select the **Permissions** tab
 - a. Set **View Content:** to Anyone
 - b. Set **Change content:** to Only owner
 - c. Set **Execute:** to Anyone
8. Press **OK**

7.5.2 Icon for piHPSDR

1. Open a file editor. Your Raspberry pi will have several, and an editor called “geany” can be loaded from the programming menu.
2. Enter (or paste) the text below into your editor window

```
[Desktop Entry]
Version=1.0
Type=Application
Terminal=false
Name[en_GB]=piHPSDR
Exec=/home/pi/github/pihpsdr/pihpsdr
```

```
Icon=/home/pi/pihpsdr/hpsdr_icon.png
Name=piHPSDR
```

3. Use **save as** to save it to your desktop with filename piHPSDR.desktop
4. Using your file browser, open your /home/pi/Desktop folder
5. Select file piHPSDR.desktop (may shows as piHPSDR with no extension)
6. Right click and select **properties**
7. Select the **Permissions** tab
 - a. Set **View Content:** to Anyone
 - b. Set **Change content:** to Only owner
 - c. Set **Execute:** to Anyone
8. Press **OK**

7.6 Reflashing firmware

1. Run the flashwriter desktop application (see section 4.3.1)
2. Download a new firmware .BIT file
3. Click **Open File**, navigate to the .BIT file and click **open**
4. The BIT filename will now be displayed
5. Ensure the buttons are set to **Primary**
6. Click **Program**
7. The device will be programmed, and progress will be shown on the display.

7.7 Replacing the File System

The Linux file system is just like a PC file system: if mistreated it can lose data. In the worst case, your Raspberry Pi may no longer boot and the filesystem will need to be reinstalled. Apache Labs will make available replacement images if this happens.

There are two forms of file system depending on the variant of Raspberry Pi. These are not interchangeable and you can't have both:

- A file system based on a micro SD card, which is loaded in a socket on Saturn. This is easily reflashed using another PC.
- A file system based on "eMMC" memory device on the Raspberry Pi itself. This is more complicate to reflash, and needs a cable connected to the Saturn board itself

To reflash either memory type, you will need to remove the radio's top panel then identify which kind of file system you have.

- Begin with Saturn powered off.
- Remove the top cover.
- The Saturn DSP board is the large circuit board assembled to the heatsink. The raspberry Pi is on the right hand side, near the rear panel.
- At the front edge of the board to the right is an SD Card socket.

If there is an SD card in the socket, you have an SD card filesystem that can be reflashed by inserting that card into a card reader on a laptop or PC.

If there is no SD card, you have an eMMC file system and it will need to be reflashed “in situ” using a USB cable.

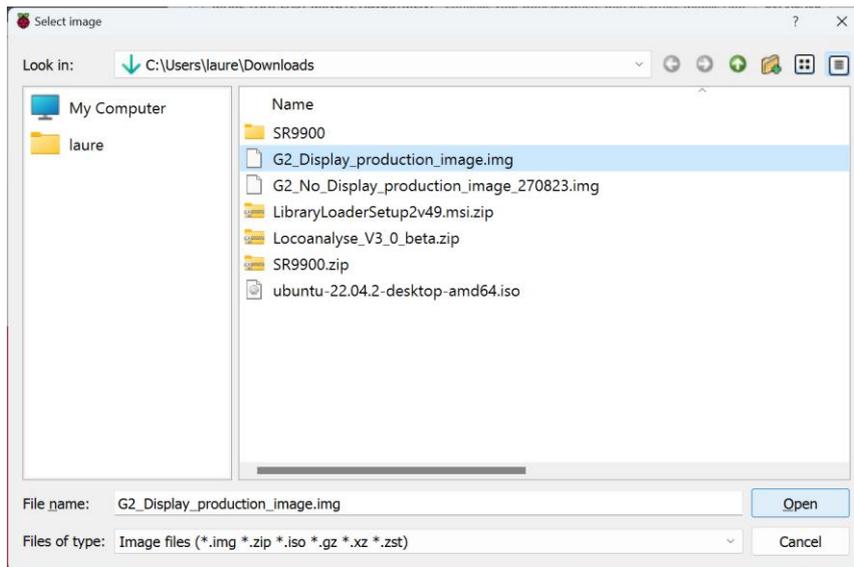


7.7.1 Reflashing an SD Card

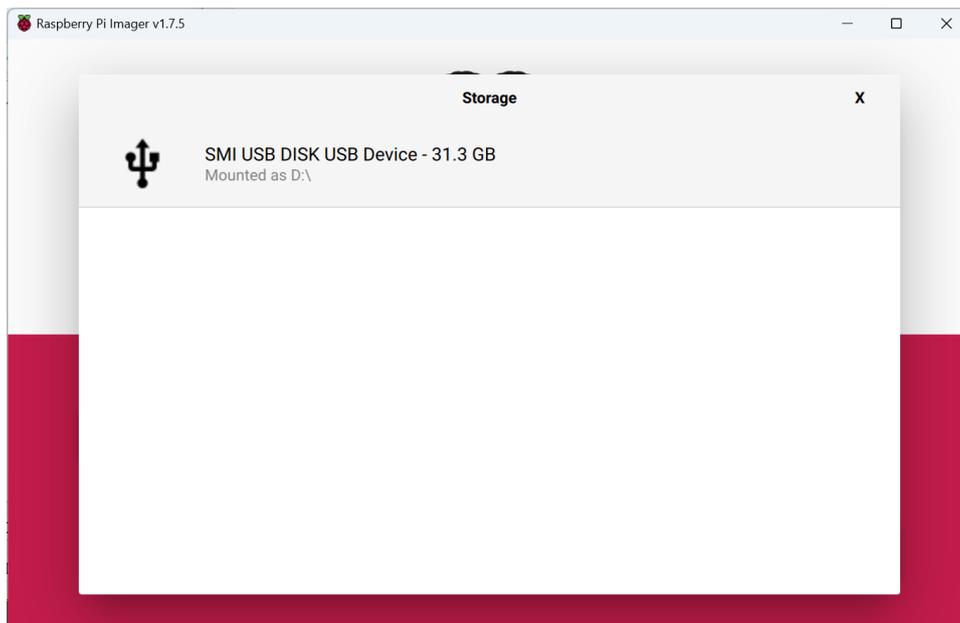
1. Download the Raspberry Pi Imager from <https://www.raspberrypi.com/software/>
2. Click “open file” when your browser completes the download
3. Click “yes” to allow the installer to make changes, then “install” on the installer page



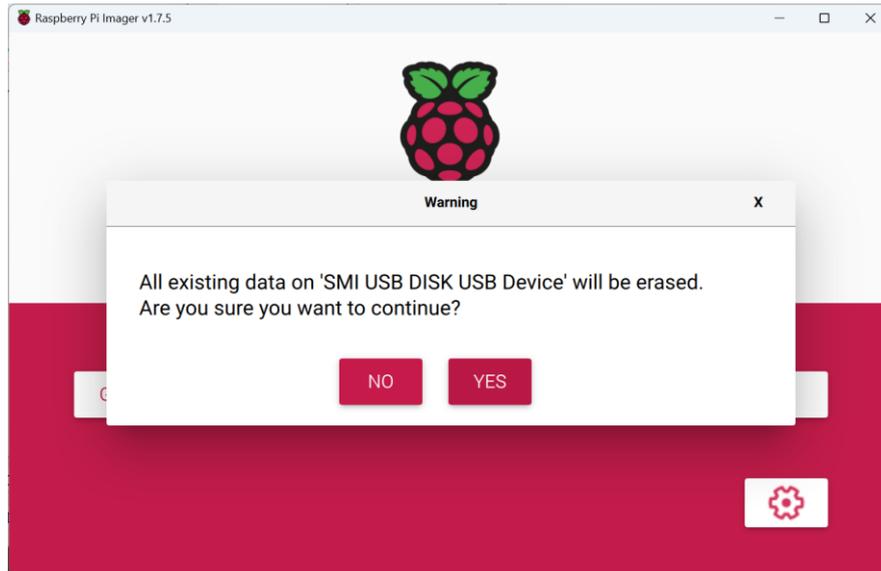
4. Download a drive image from Apache Labs:
 - a. With display: https://storage.googleapis.com/anandg2/G2_Display_production_image.img
 - b. Without display: https://storage.googleapis.com/anandg2/G2_No_Display_production_image_270823.img
 - c. (Note the Raspberry pi images are for 32 bit operating system and ~7GByte.)
5. Insert a suitable SD card into your computer (laptops usually have a port for this, or you may need a USB card reader)
6. Run Raspberry Pi Imager from your start menu
7. Click “yes” to allow it to make changes
8. Click CHOOSE OS and select “use custom”
9. navigate to the .img file you downloaded
10. click “open”



11. click CHOOSE STORAGE and navigate to your SD card
12. Click on the selected drive. Make sure you have the correct drive!



13. Click WRITE then YES on the next screen



14. The data will be written to the drive. This will take several minutes.
15. When the write has completed you will get a message. Remove the drive and insert it into the SD card socket on the Saturn board in your G2.
16. If you try to reflash an existing linux boot drive, you will need to reformat first. There is an option for this on the first screen of the Raspberry Pi Imager.

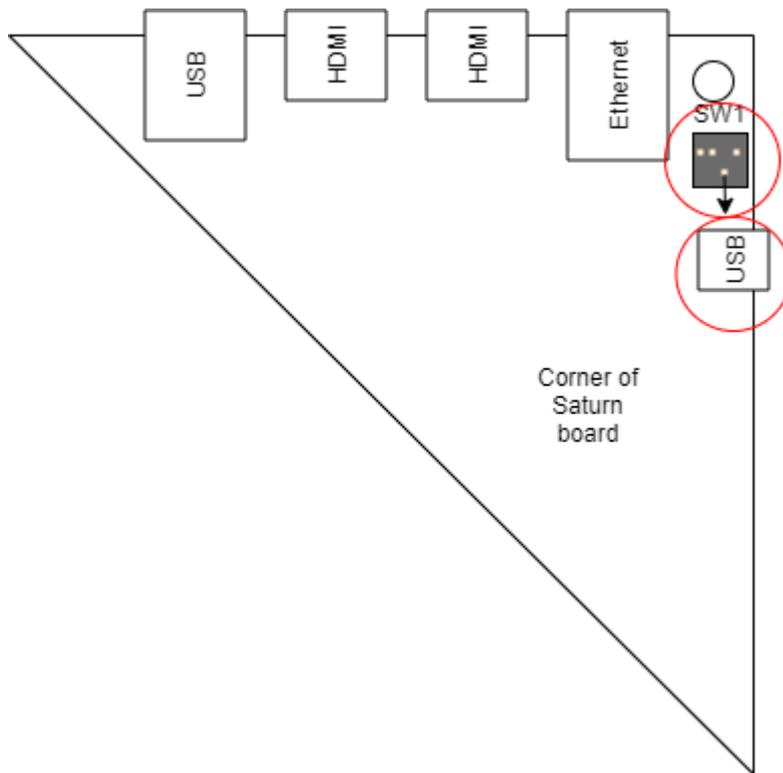
7.7.2 Reflashing on-board eMMC memory

The procedures are well covered on third party websites, for example:

<https://www.raspberrypi.com/documentation/computers/compute-module.html>

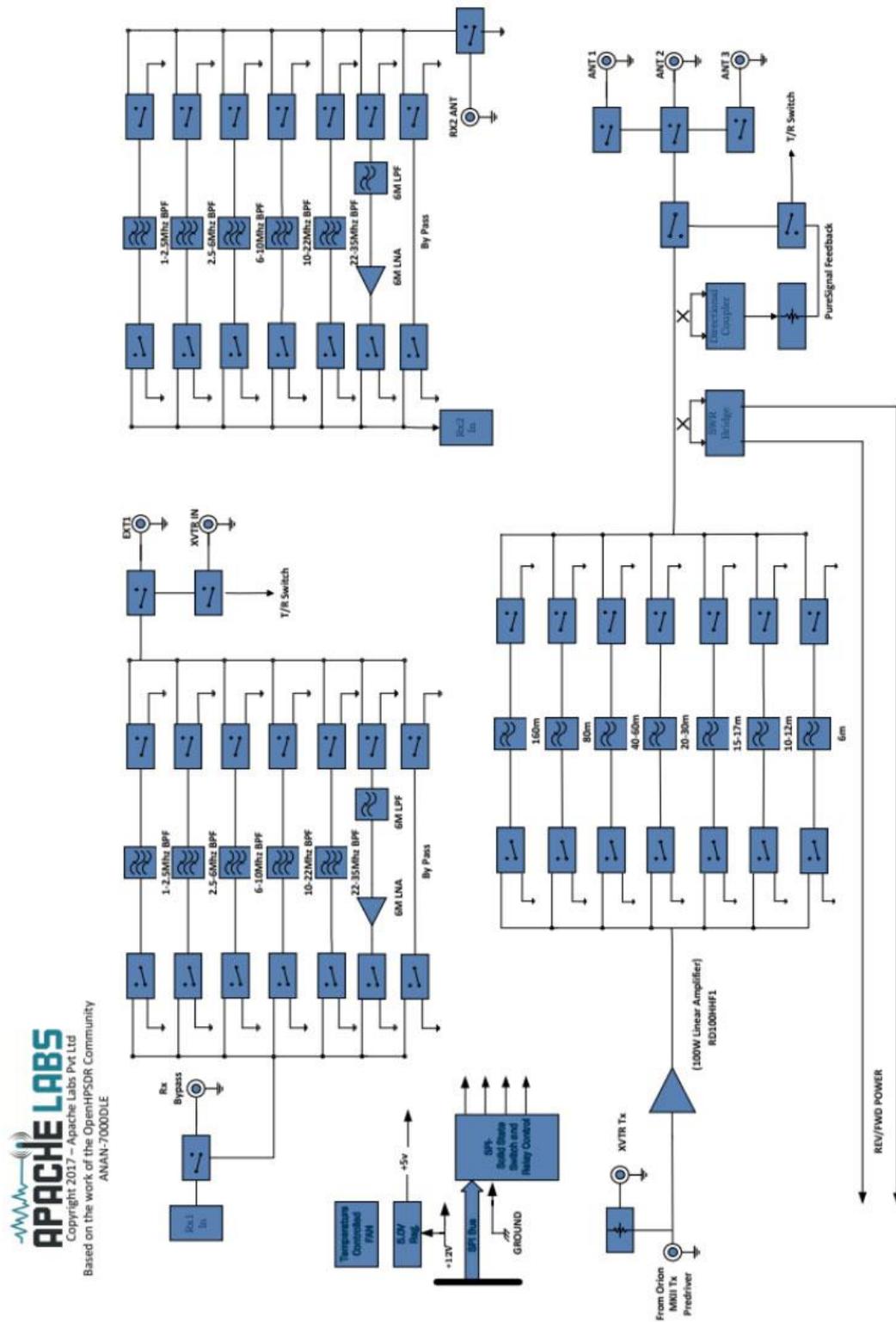
<https://www.jeffgeerling.com/blog/2020/how-flash-raspberry-pi-os-compute-module-4-emmc-usbboot>

- Begin with Saturn powered off.
- Remove the top cover.
- The micro USB connector on the side of the Saturn board is the one needed for reflashing the memory.
- Use a USB cable to connect that to your PC.
- Locate the 4 position switch SW1 adjacent to the ethernet connector.
- Use a pencil or similar tool to set the second switch in from the edge to the position shown.
- Power up Saturn and follow the instructions on the Raspberry Pi website.



After the process is complete you will need to reset the SW2 position. All 4 switches should be next to the board mounting hole for normal use.

7.8 Block Diagrams



8 Support

8.1 Warranty

We warrant that our products are free of defects in material and workmanship and extend this warranty under intended use and normal service conditions to the original owner for a period of 12 months from the date of purchase. This warranty does not apply to any product that has been repaired [Unauthorized] or altered in any manner and is void if damaged by accident, neglect, unreasonable use, improper installation, lack of proper impedance matching [High SWR], improper tuning, over driving power limits, damage due to use of any third party accessories, or any other cause not arising specifically out of defects in material or workmanship.

Our obligations are limited to repairing or replacing, at our sole discretion, any product or part that is returned to the mailing address, all shipping or transportation charges must be pre-paid, accompanied by proof of purchase and which examination reveals to have been defective within the warranty period stated above.

Our Warranty does not imply any obligation other than herein stated. Any implied warranties, including but not limited to fitness for a particular purpose, are limited in duration for the above one year period.

We shall not be liable under this warranty, or any implied warranty, for loss of use of the product or for any other consequential loss or damage incurred by the purchaser.

Warranty -- Apache Labs website has a link to explain their warranty and other pertinent information:

<http://www.apache-labs.com/terms-of-use.html> under "Terms of Use".

IMPORTANT

The FPGA and other components on the SATURN PCBs can be damaged by Electrostatic Discharge [ESD]. When handling the PCB, you should take ESD precautions [e.g. use a work bench with an antistatic mat on it and wear a grounded wrist strap].

When powered on do not touch the top of any IC in order to determine its temperature. In particular, do not touch the Switch Mode Regulator IC. Touching its leads can change the switching frequency and alter the output voltage.

8.2 References

- [1] Apache Labs community forum: <https://apache-labs.com/community/index.php>
- [2] Open HPSDR Project Website: <http://openhpsdr.org/>
- [3] Saturn repository: <https://www.github.com/laurencebarker/Saturn>
- [4] piHPSDR repository: <https://github.com/dl1ycf/pihpsdr>
- [5] Thetis repository: <https://github.com/TAPR/OpenHPSDR-Thetis>
- [6] Raspberry pi source code repository: <https://github.com/raspberrypi>
- [7]