

Peaberry SDR V2 by AE9RB

Welcome to homebrew SDR. This version of the manual assumes you have assembled and tested hardware. If you have a kit or need to troubleshoot hardware then you want the other manual.

The Peaberry SDR V2 is designed as a platform for experimentation with SDR applications. These applications are typically free or open source and will have a steep learning curve. Most of the complexity of SDR is in the software. If your desire is to explore software then you will have a lot of rewarding fun. If you buy an assembled Peaberry because you are looking for a more turnkey solution then you may be disappointed. Especially if you run into software compatibility problems.

Traditionally, a homebrew SDR transceiver would be connected to a second sound card installed in your computer. This means your receiver performance is completely dependent on the quality of the sound card. Even the transmitter circuits can operate differently since there's no standard for output voltage. I believe this has held back advancement of inexpensive zero-IF designs since using a sound card complicates sharing of experiment results.

The Peaberry SDR V2 is designed to solve the sound card problem. It includes a DAC/ADC that is known to have the proper characteristics for SDR. The receiver has an incredible 99dB of dynamic range (two-tone, 2kHz, 3rd order, ARRL method).

This manual will cover configuring the Peaberry SDR V2 under a Windows operating system. Windows XP or newer is known to work. The application we'll be using is HDSDR. When done, your radio and software will be fully ready for use.

You're welcome to use another operating system or application but you'll have to figure it out on your own. There are dozens of compatible applications across all modern operating systems. Please remember that the Peaberry SDR is produced entirely by one person. Radio software is vast and complex so I am unable to provide application support. Use the Peaberry SDR support forum or the application's support forum to connect with other users who can help.

<http://AE9RB.com/forum/viewforum.php?f=4>

Additional Equipment Required

- CW or SSB receiver
- 13.8V 1A regulated power supply
- Power supply wire, 20AWG or better
- 50 ohm antenna, BNC terminated
- 50 ohm dummy load
- Computer to run DSP software

□ Connections

A power plug is included with the radio. Wire it up to your 13.8V DC power supply with the center positive. The radio is fused for 0.75A but a 1A or greater power supply is recommended.

The Peaberry SDR is a transceiver so it expects an antenna that can transmit well. For example, a vertical with ground radials, a dipole, a yagi, etc. If you simply connect a random wire to the center of the BNC then receiver performance will be poor.

If you bought an assembled radio from AE9RB.com then it came with a USB cable. This, obviously, connects the radio to your computer.

The KEY jack is for a morse code key or paddle. Not all applications support this. HDSDR does not yet send CW well and does not support this jack.

The AMP jack will trigger an external RF amplifier. Many amps include an RF detector and timer but if it has a manual trigger then you don't have to worry about SSB missing a first syllable or cutting out during a long pause. The AMP jack is designed for CMOS/TTL triggered amps. It will not allow enough current to directly trigger a relay.

The ATU jack supports the Elecraft T1 automatic antenna tuner. The Peaberry SDR will signal the tuner when you change bands. There is no configuration or setup needed. Simply connect the tuner with a stereo patch cable.

You can also connect the ATU jack to an Arduino and write a custom program to control an antenna switch or change bands on an amplifier. An example program which can read the band data is available in the source code repository.

<https://github.com/AE9RB/peaberry>

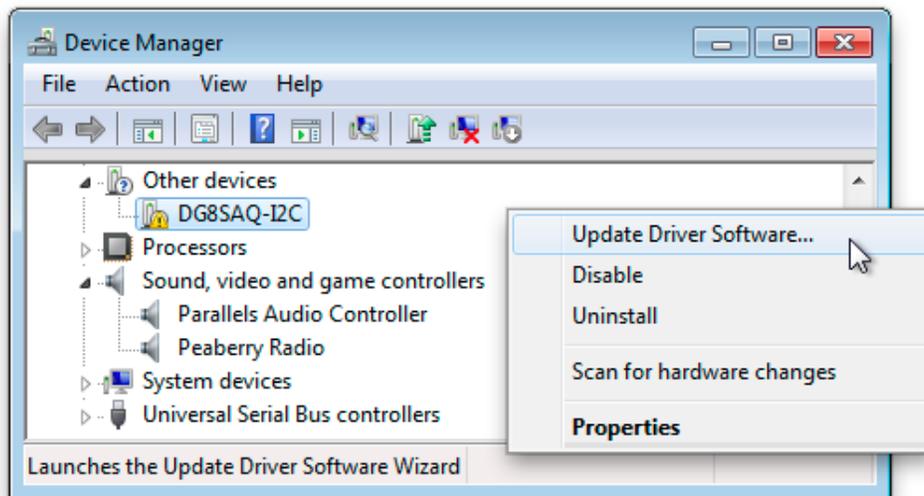
□ Windows Device Driver

Windows requires a device driver. Other operating systems do not. You can get a ready-to-use libusb-win32 driver from the support forum:

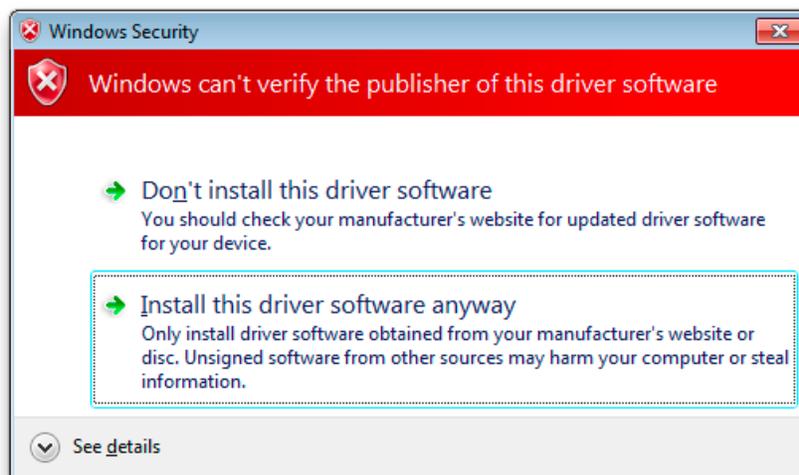
<http://AE9RB.com/forum/viewtopic.php?f=4&t=96>

Unzip the driver and launch Device Manager. The “Peaberry Radio” appears as a sound card and should have been automatically installed by Windows because it is a standard USB audio device. You should also have a “DG8SAQ-I2C” device which needs a device driver.

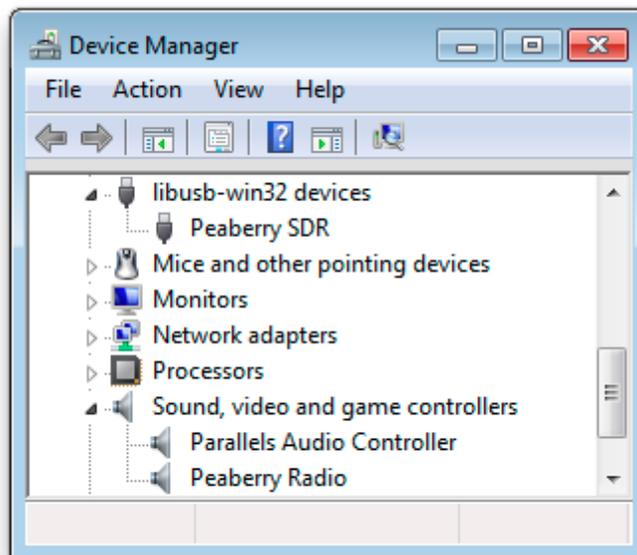
The steps vary slightly with different version of Windows. The examples here are from Windows 7. Right click on “DG8SAQ-I2C” and choose “Update Driver Software...”.



Select “Browse my computer for driver software” and provide the path where you unzipped the driver. Select “Next” and you’ll probably get a security warning. Install the driver anyway.



If everything goes well then the other/unknown “DG8SAQ-I2C” device is replaced by “Peaberry SDR” showing as a libusb-win32 device.



Not seeing the Peaberry Radio sound device?

If you have ever used the automatic installer (.exe) from PE0FKO for a SoftRock then you must uninstall and delete the driver. After selecting “Uninstall” from Device Manager, check the box to delete all files.

Errors with Windows 8?

You must disable driver signature enforcement in Windows 8. Do an internet search for “disable driver signature enforcement” if you do not know how to do this.

□ HSDR Application

Install the latest version of HSDR. It's a standard Windows installer. Take note of the path it installs to. If you already have HSDR installed then run the "delete_settings.cmd" so you have a clean start. If at any time you have a problem following this guide, come back to this beginning and run "delete_settings.cmd" to start anew.

<http://www.hdsdr.de/>

Install the latest version of CFGSR. The setup.exe generally works better than the msi installer.

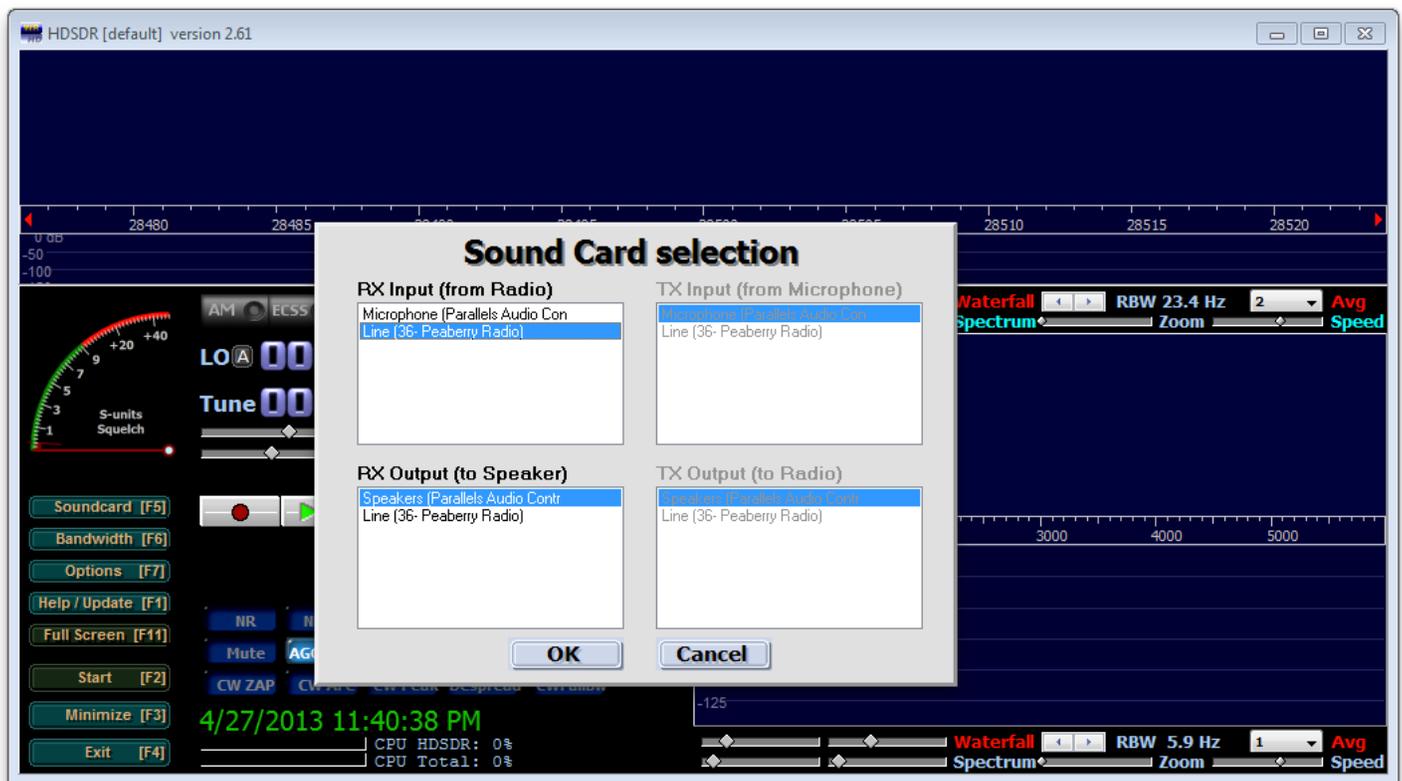
<http://pe0fko.nl/CFGSR/>

You have to figure out where CFGSR was installed. It doesn't always tell you. It's different on the various versions of Windows. When you find it, somewhere within is a ExtIO_Si570.dll file. Copy this file to the HSDR folder. Here are the paths for my Windows 7 system:

C:\Program Files (x86)\CFGSR\ExtIO_Si570\ExtIO_Si570.dll

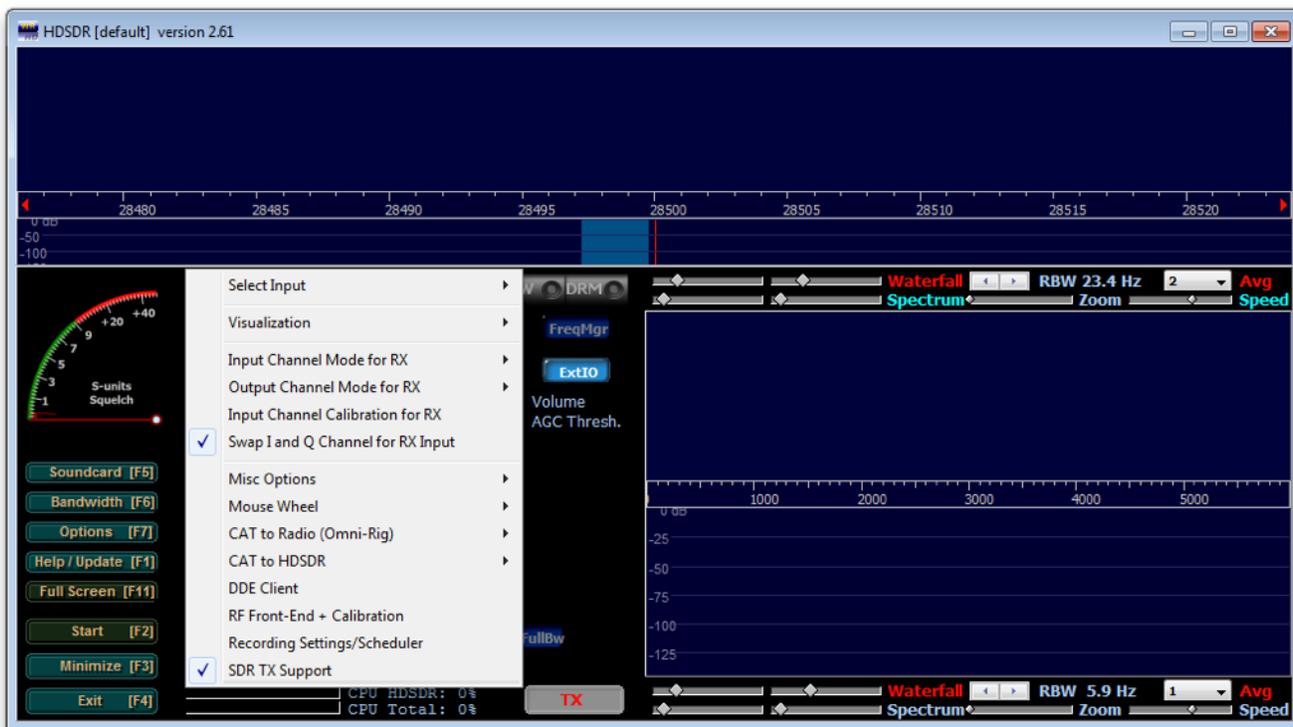
C:\Program Files (x86)\HSDR\ExtIO_Si570.dll

Do not run the CFGSR tool. It will conflict with HSDR. All we needed was the DLL. Launch HSDR now.

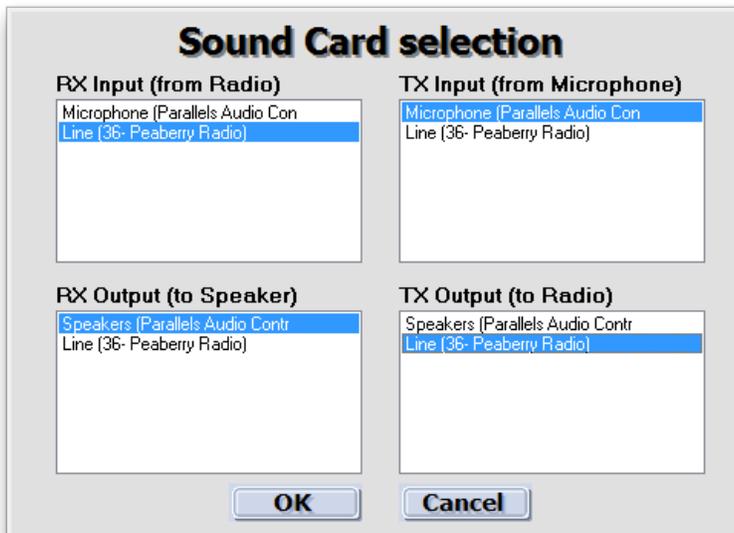


Set the “RX Input (from Radio)” to “Peaberry Radio”. Set the “RX Output (to Speaker)” to wherever you want to hear audio. Press OK.

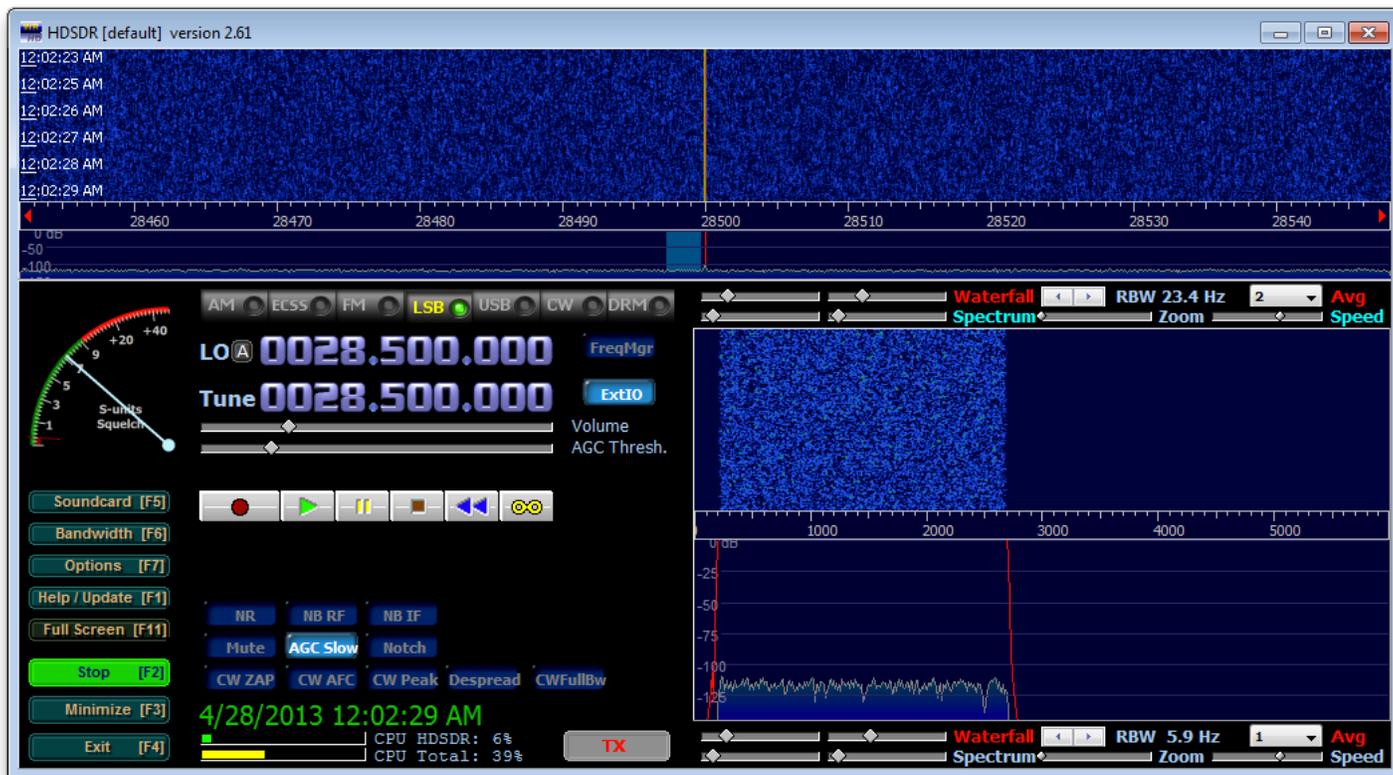
Click on the “Options [F7]” button. Enable “Swap I and Q channel for RX Input”.



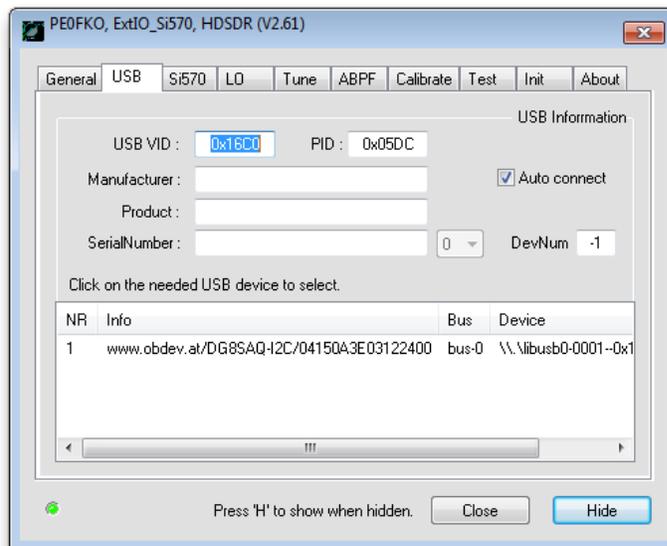
Click on the “Options [F7]” button again. Select “SDR TX Support”. Choose “Yes” after reading the warning. Set the “TX Input (from Microphone)” to where you have a microphone. Set the “TX Output (to Radio)” to “Peaberry Radio”. Press OK.



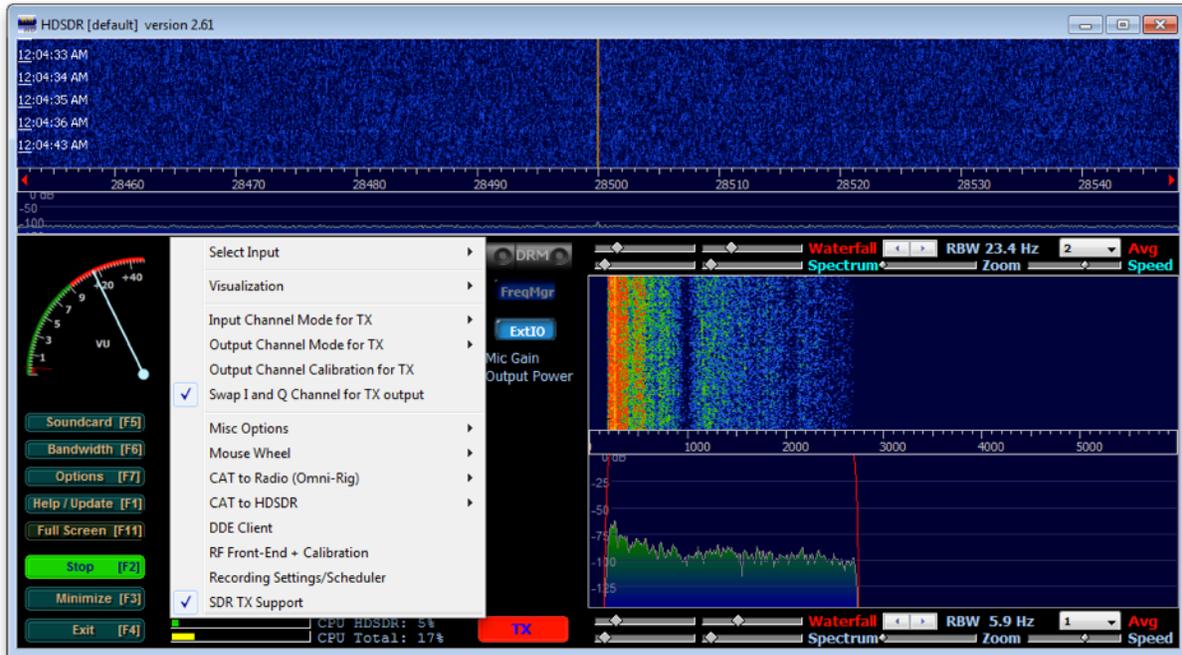
Click on the “Bandwidth [F6]” button. Select 96000 for the sampling rate. Close the dialog with the “X” button. Click on the “Start [F2]” button. The radio will begin working and you’ll hear static from the speaker.



Click on the “ExtIO” button. You should see a message about the Si570 being connected. If not, you can try to figure out what’s going on from the USB tab.



Connect the radio to a dummy load. Click on the “TX” button. The LED on the radio will blink rapidly while HDSDR is transmitting.



Click on the “Options [F7]” button. Enable “Swap I and Q channel for TX Output”. You must be transmitting to do this.

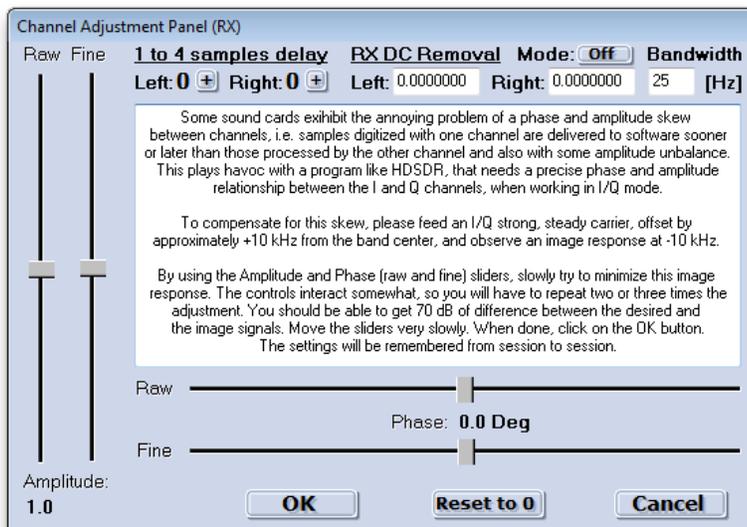
Press “TX” to turn off the transmitter. All of the HDSDR setup that is unique to the Peaberry SDR V2 is now complete.

The BS170 MOSFETs Q6-Q8 will be destroyed if you transmit significant power out of band or into a poorly matched antenna system. Replace all three if this happens.

□ RX Adjustment

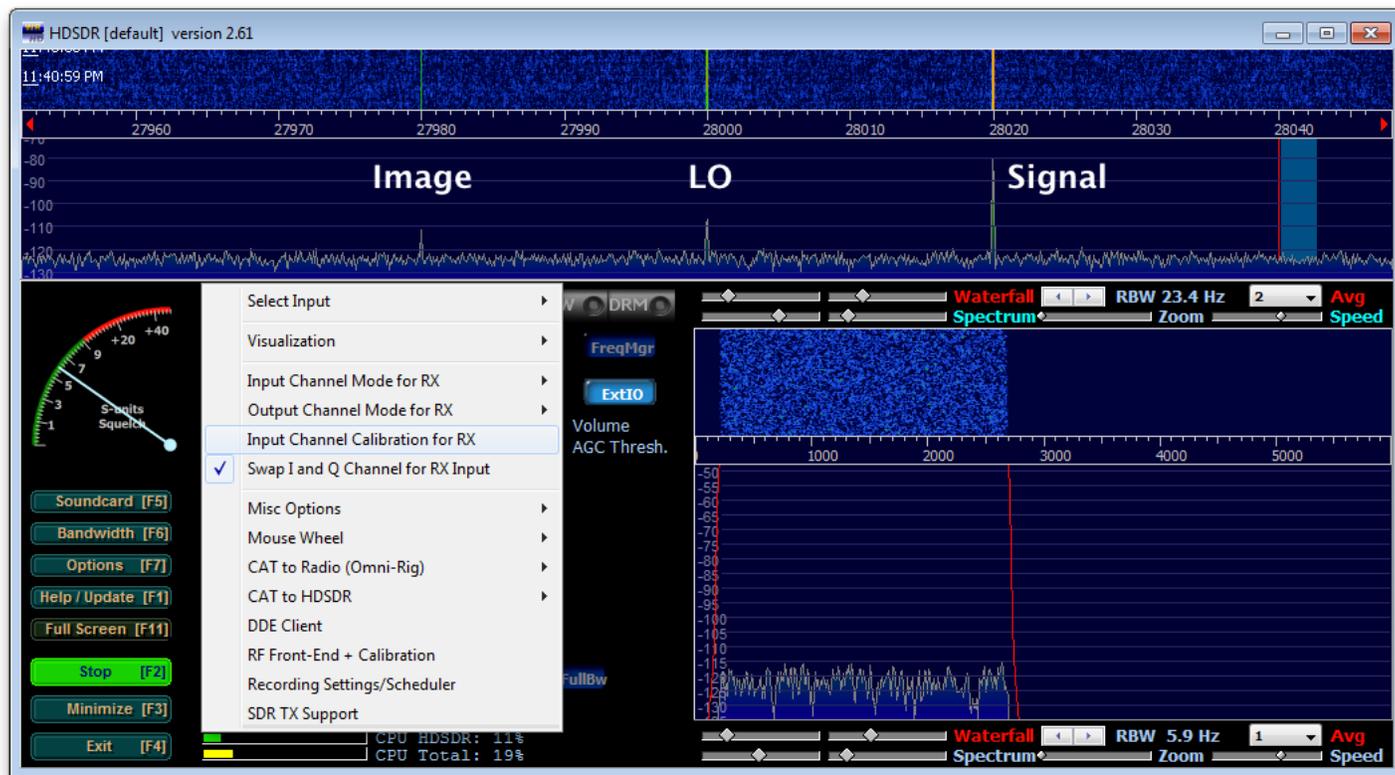
Zero-IF radios like the Peaberry SDR must have the I/Q data adjusted to compensate for hardware variances. Some software attempts to do this automatically with varying degrees of success. HSDR is adjusted manually with perfect results.

In HSDR, click “Options [F7]” then select “Input Channel Calibration for RX”. A dialog with sliders and some help text will appear. Ignore the settings about sample delays.



You will need a strong signal to make the adjustment. A signal generator is easiest, but a strong CW or AM station can also be used. Adjust the sliders until the image is completely gone. HSDR does not save this per-band so you will have to make the adjustment every time you change bands.

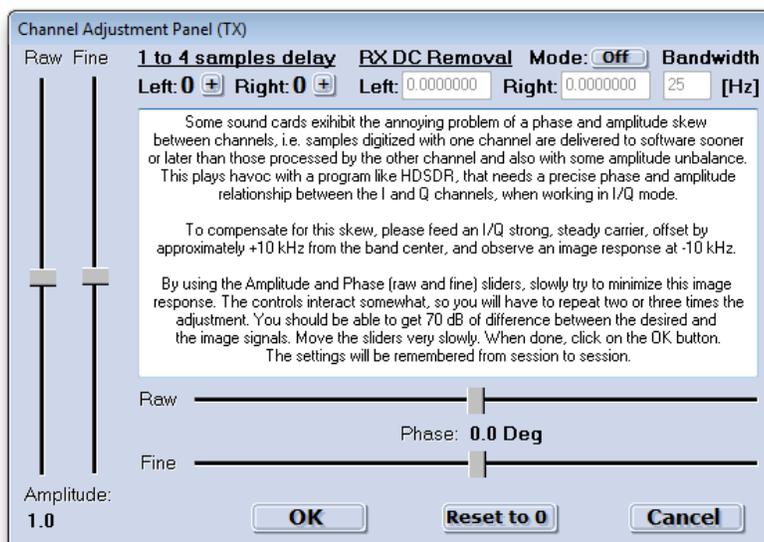
Newer versions of HSDR can remove the DC spike that appears at the LO frequency. Changing the mode to “Auto” is all you need.



□ TX Adjustment

The transmitter must have the I/Q data adjusted just like the receiver. If you fail to do so then your image will not be suppressed enough.

In HDSDR, while transmitting, click “Options [F7]” then select “Output Channel Calibration for TX”. A dialog with sliders and some help text will appear. Ignore the settings about sample delays and DC removal. The help text here is for the receiver and not very helpful.



For example on 10m: Set mode to AM, LO to 28.400, tune to 28.380, and begin transmitting into a 50Ω dummy load. Use another SSB or CW receiver to monitor the AM carrier on 28.420. Adjust the sliders to minimize the strength of the received signal.

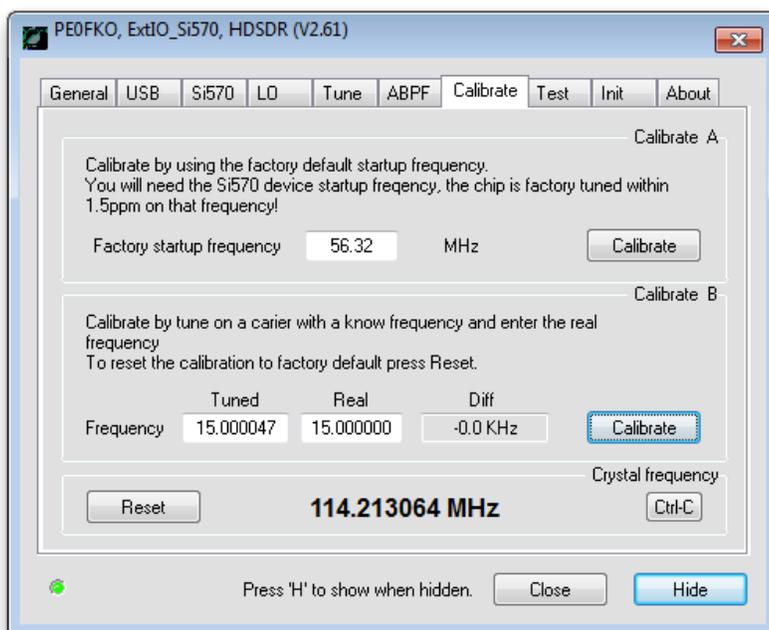
□ Si570 Calibration

You can perform a manual calibration to improve frequency accuracy of the Si570.

Use the “Calibrate” tab in the ExtIO utility. “Calibrate B” is used to make a manual adjustment. “Calibrate A” will restore the factory calibration.

□ Get on the air!

Your Peaberry SDR V2 is ready to operate. Get on the air and enjoy it. HDSDR is only one of many applications that can be used. You may prefer a different one.



Digital modes are typically done with “virtual” audio cables connecting a SSB application like HDSDR with a digital mode application like fldigi. Digital mode configurations are beyond the scope of this documentation.

73 David AE9RB

Appendix A: LED Status

A power-on self-test function is included in the Peaberry SDR V2. The status LED sends error messages in morse code at 5 WPM.

Status	Information
Off	PSoC not initialized (or LED failure)
Steady On	Normal operation.
Rapid Blinking	Transmitter active.
“BOOT”	Bootloader active. Ready to receive new firmware over USB.
“XTAL”	24 MHz crystal X1 failure.
“I2C”	I2C bus failure. Both PCM3060 and Si570 are unreachable.
“SI570”	Si570 U9 not responding.
“PCM3060”	PCM3060 U10 not responding.

By default, the included LED D4 is driven at 5mA with current limiting by R2. You may prefer to install the LED on your enclosure and increase the power. Pretty much any LED can be used so long as you don't exceed 16mA. Be careful if you select a complete assembly as some will include a resistor which you do not need. A recommended part is the Dialight 5592201007F (Digi-Key 350-2422-ND).

Appendix B: Channel Reversal

The "Peaberry Radio" sound card that shows up on your computer carries the I/Q radio signal. This will appear as regular stereo device with, for example, the left channel containing the I data and the right channel containing the Q data. There is no standard for which channel carries which data. Most applications will let you configure this but some do not. Tune the LO to one of these magic frequencies and the channel reversal will immediately take effect and be saved to EEPROM.

LO Frequency	Effect
33.333333	Normal. The HDSDR configuration in this guide assumes this setting.
33.444444	Receive is reversed.
33.555555	Transmit is reversed.
33.666666	Both are reversed.

Appendix C: USB Commands

The USB command set that was developed by DG8SAQ and PE0FKO for the Atmel AVR has been re-implemented for the PSoC. This makes the Peaberry SDR V2 extremely compatible with free and open source software for homebrew radios.

Command - Direction	Function
0x00 - IN size: 2	Returns version number. Most significant byte is major version. Always returns V0.0.
0x02 - IN size: 1	Deprecated. Read pin status.
0x20 - IN valueHi: register index: value	Write a byte value to a Si570 register. Only value 0x01 can be written to register 0x87 to request a reset of the Si570. The Si570 is not actually reset, but command 0x3D will return the factory setting after issuing.
0x30 - OUT size: 6	Deprecated. Set LO frequency by Si570 register. If your software still uses this command, tell it to use a crystal frequency of 114.285 MHz.
0x32 - OUT size: 4	Set current LO frequency. Coded as fixed point in 11.21 bits. The Si570 operates at 4 times the tuning frequency. For example: (float)frequency * 4 * (1<<21)
0x33 - OUT size: 4	Set Si570 crystal frequency. 8.24 bits format. This is changed to calibrate your radio.
0x3A - IN size: 4	Get current LO frequency.
0x3C - IN size: 4	Get Si570 startup frequency. 11.21 bits format. Always 56.32 MHz.
0x3D - IN size: 4	Get Si570 crystal frequency. 8.24 bits format. Returns current setting. Use command 0x20 first to obtain factory setting.
0x3F - IN size: 6	Get Si570 control registers 7:12.
0x50 - IN valueLo: PTT size: 1	Set PTT and read CW paddle status. Bit 0x01 is high to transmit.
0x51 - IN size: 1	Read CW paddle status. Bit 0x20 is KEY_0 and 0x02 is KEY_1.

Appendix D: Firmware Updates

New firmware is distributed in a ZIP file containing two formats of the ROM. Two methods of updating the firmware are available.

USB Bootloader

When the BOOT jumper is in the position nearest the USB jack the LED will blink BOOT in morse code. In this mode the radio is ready to accept firmware. You will need the free PSoC Creator (2.1 or newer) from Cypress to use this method.

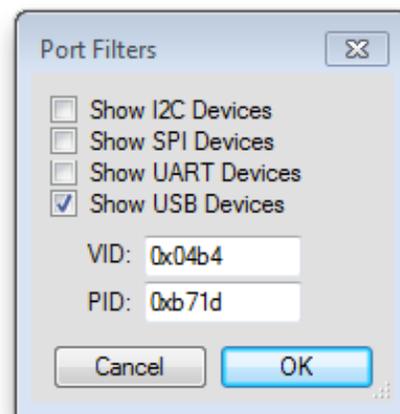
Get it here: <http://www.cypress.com/go/creator>

If Cypress asks for a company name, use “personal/hobbyist”.

Install the software as per the directions from Cypress. It’s an ordinary Windows installer so you shouldn’t have much trouble. Launch the PSoC Creator application after it’s installed.

From the menu of PSoC Creator, choose “Tools” then “Bootloader Host...”. This will launch the Bootloader Host application.

Press the “Filters...” button in Bootloader Host. A dialog will pop up. Ensure that “Show USB Devices” is checked and the VID and PID are as shown here. The other checkboxes do not matter. Press “OK”.

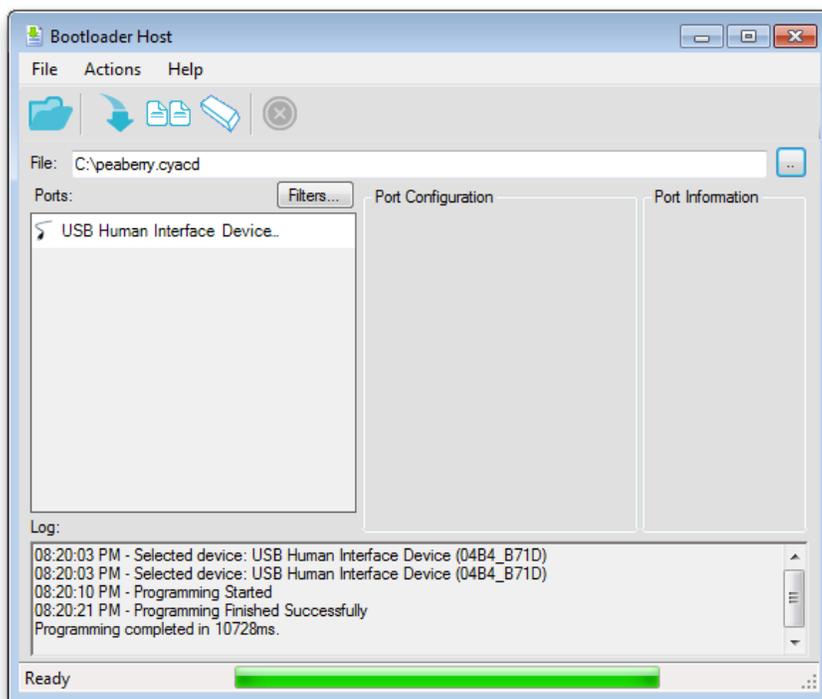


Make sure the BOOT jumper is on the USB jack side and the radio is powered and connected. You should see a “USB Human Interface Device” in the ports list.

Choose “File” then “Open...” from the menu. Select the .CYACD file for the firmware you wish to load.

Now choose “Actions” then “Program” from the menu. You may need to click the ports list or relaunch Bootloader Host to get this enabled.

In a few seconds the new firmware will be loaded and you’ll get a success message. Move the jumper back and restart your radio by interrupting power.



SWD Programming

A 10-pin SWD connector is included with every radio for connecting a Cypress MiniProg3 or other programmer. The connector isn't keyed so please observe the PCB silkscreen when connecting your programmer. This method is for advanced users who should be familiar with the process. Here are the critical settings you need to know:

- File Path: Choose the .HEX file
- Programming Mode: Reset
- Connector: 10p
- Protocol: SWD
- Voltage: 5.0V
- Clear EEPROM: Disable (or Si570 calibration is reset)

