

Installing & using usbsoftrock on Ubuntu

Introduction

This short note describes the installation, including compilation, of **usbsoftrock** on a computer running Ubuntu. The **usbsoftrock** software allows the control of:

- a QRP-2000 frequency synthesizer
- any Si570 oscillator using DG8SAQ firmware
- a Si570 equipped softrock
- a Si570 equipped softrock including the mobo (motherboard) option

usbsoftrock

The usbsoftrock project was originally hosted by Google¹. There are now versions hosted on GitHub². The necessary USB configuration is no longer as described in the README file.

Ubuntu

This procedure has been tested on Ubuntu 12.04 (LTS), but should be identical on later versions³.

Origins &c.

This document was first prepared in 2012, and slightly updated in 2020 & 2023.



Copyright © 2023 R A Gape

Licensed under the Creative Commons Attribution-ShareAlike 4.0 International License⁴.

RAG, G8DQX: 2012, 2020, 2023-06-27

¹ At <http://code.google.com/p/usbsoftrock/>. NB this is now an **archive** page. Taking a local copy is **strongly** recommended. (Also at <https://web.archive.org/web/20170103133851/https://code.google.com/archive/p/usbsoftrock/downloads>.)

² What appears to be a straight copy at <https://github.com/alexlee188/usbsoftrock-alex>, and another version at <https://github.com/8cH9azbsFifZ/usbsoftrock/tree/master/usbsoftrock-1.0.2>.

³ YMMV.

⁴ See <https://creativecommons.org/licenses/by-sa/4.0/>.

usbsoftrock Installation

The procedures below will:

- a) configure the host system to recognise a QRP-2000 or Si570 equipped softrock correctly and consistently
- b) download and compile **usbsoftrock**

USB configuration

- Create a **udev** file (as root: use **sudo**) **/etc/udev/rules.d/51-DG8SAQ-I2C.rules** containing:

```
SUBSYSTEMS=="usb" ATTRS{idVendor}=="16c0" ATTRS{idProduct}=="05dc"  
MODE=="0666" SYMLINK+="DG8SAQ-I2C"
```

- Issue command⁵:

```
sudo udevadm control --reload-rules
```

At this point, plugging in a QRP-2000 or Si570 equipped softrock should result in a kernel message that a new low speed USB device has appeared, using `uhci_hcd`.

Alternative USB configuration for Softrock

As above, but the file name should be **51-usbsoftrock.rules**, containing:

```
SUBSYSTEM=="usb", ACTION=="add", ATTR{idVendor}=="16c0",  
ATTR{idProduct}=="05dc", MODE="0666", SYMLINK+="softrock",  
GROUP="dialout"
```

Download, compile & install usbsoftrock

- Install the following packages, if not already present:
 - **build-essential**⁶
 - **libusb**⁷
 - **libusb-dev**⁸
 - **libncurses5**
 - **libncurses5-dev**

⁵ This is not necessary on Ubuntu 11.04 (Natty) and presumably later.

⁶ Brings in gcc &c.

⁷ Will already be installed.

⁸ Presumably required.

- Download the **usbsoftrock** source⁹, as a tarred & gzipped file¹⁰ (version 1.0.1 is recommended¹¹)
- unpack the source directory to a convenient location in your home directory
- open a terminal (if not already open)
- change directory (**cd** command) to the **usbsoftrock** source folder¹²
- issue the command:

```
./configure
```

- this command should complete quickly
- issue the command:

```
make
```

- this will invoke the compiler, should complete quite quickly, and will deliver a compiled **usbsoftrock** in the source directory
- issue the command:

```
sudo make install
```

- this will copy the compiled **usbsoftrock** to the local installation directory¹³
- **usbsoftrock** may then be tested by issuing the command:

```
usbsoftrock
```

- which should result in a screen of usage information

⁹ Available from <http://code.google.com/p/usbsoftrock/downloads/list>.

¹⁰Filename will be **usbsoftrock-1.0.1.tar.gz**, or similar.

¹¹Unless your application requires softrock mobo support, in which case version 1.0.2 may be necessary.

¹²The directory name will be **usbsoftrock-1.0.1** or similar.

¹³The local installation directory is **/usr/local/bin**, and this is included in the default path by Ubuntu.

Using `usbsoftrock`

As a hack straight, but edited, lift from the **README** file:

This utility for Unix platforms is designed for controlling Tony Parks' optional I2C adapter for the RXTX 6.3 with the standard DG8SAQ firmware and PE0FKO 15.10 and the Mobo software on a AT90USB162.

`usbsoftrock` should compile anywhere `libusb` exists, with adjustments to library includes in the makefile, if required.

The CW Key interface on the board is **not supported**, and would require a `dæmon` for useful results as a CW keyer.

Prerequisites

- `libusb 1.0`
- `ncurses` (and headers i.e. `libncurses5-dev` package)
- `gcc`

Installation

```
./configure
```

```
make
```

```
make install
```

libusb and permissions on Linux

The `udev` rules noted earlier should sort these issues out.

Usage

prompt>: `./usbsoftrock [OPTION] COMMAND`

OPTION is one or more of

- a Advanced firmware present, i.e. let the firmware calculate registers
- d Enter a mode that listens for commands via UDP.
- h <freq MHz> Enable subharmonic (/3) sampling from frequency (DEFAULT off)
- i <address> I2C address in DECIMAL (DEFAULT = 85 (0x55))
- m <multiplier> Multiplication factor for frequency (DEFAULT = 4)
- p <port num> Port to listen for UDP datagrams (DEFAULT = 19004)
- s <startup frequency MHz> Factory programmed startup frequency (DEFAULT = 56.32)
- v Verbose output (fairly useful)

- vv Even more verbose output (debugging)
- x <calibrated xtall freq MHz> Corrected XTALL frequency of Si570 device calculated through the use of the calibrate command immediately after startup.

COMMAND is one of

- calibrate (may require -s option)
- getfreq
- getregisters
- interactive
- ptt {on | off}
- set bpf {on | off} (PE0FKO only)
- set freq <frequency in MHz>
- set si570_multiplier <decimal factor> (PE0FKO only)
- set startup <frequency in MHz> (PE0FKO only)
- set xtall <frequency in MHz> (PE0FKO only)
- status

Examples

Original DG8SAQ Firmware

Turn on softrock, plug in usb-i2c interface

```
usbsoftrock calibrate
```

Or if you have a different startup freq than 56.32

```
usbsoftrock -s 40.000 calibrate
```

It will return something like

```
fXTALL = 114.182074
```

Then take the value from that and use the following:

```
usbsoftrock -x 114.182074 getfreq
```

```
usbsoftrock -x 114.182074 setfreq 14.040
```

By default the program will calculate the Si570 register values locally when setting the frequency. If you specify option -a (advanced firmware) then it will use the set frequency by value command and let the interface calculate the registers. This enables features such as automatic BPF selection and smooth tuning in Fred PE0FKO's firmware.

PE0FKO Usage

Additional Support with v15.10:

- **set startup freq**

Sets the frequency of the si570 on startup

- **set si570_multiplier <decimal factor>**

Sets the oscillator multiplier value in the attiny firmware.

- **set xtall freq**

Set the calibrated oscillator value

- **set bpf {on|off}**

Enable or disable the BPF filter. Needs to be disabled for RXTX 6.3 otherwise changing frequency will sometimes key the transmitter (by default).

- **set bpf_point <index> <freq MHz>**

Set the crossover frequency for index. 4 BPF means 3 cross over points 0, 1, 2 e.g. the following would set it up as the default for the electronically switched BPF kit from Tony:

```
set bpf_point 0 4.0
```

```
set bpf_point 1 8.0
```

```
set bpf_point 2 16.0
```

- **-a flag**

forces use of the set/get freq by value commands which is a more generic interface. Recommended for this firmware, especially if the xtall freq has been stored in the device as you no longer need to specify -x on the command line.

Additional Support with v15.12, TF3LJ or Mobo AT90USB162 Variants

- **set lpf_addr**
- **set lpf_point**
- **set lpf [on|off]**
- **set bpf_addr**
- **set si570_multiplier <band> <decimal factor>**

Sets the oscillator multiplier value in the attiny firmware for the band specified. For 4 BPF this would be 0, 1, 2, 3. If you use this then you probably want to use option **-m 1** to disable the default x 4 multiplication inside usbsoftrock.

Calibration is the same as the other firmware, however one has the option to store the crystal frequency in the firmware.

```
usbsoftrock calibrate
```

```
usbsoftrock set xtall 114.182074
```

and the startup frequency can be changed from the default 7.050 to something else:

```
usbsoftrock set startup 14.080
```

Since I use an RXTX 6.3:

```
usbsoftrock set bpf off
```

Afterwards the **status** command will produce something like:

```
andrew@msi-wind:~/dev/usbsoftrock$ ./usbsoftrock status
```

```
Version      : 15.10
Frequency    : 14.080000 (x 4.00)
Startup Freq: 14.080000 (x 4.00)
Xtall Freq   : 114.182074
Smooth Tune  : 3500 PPM
LO Subtract  : 0.000000
Multiply     : 1.000000
Filter Bank 1:
  CrossOver[0] = 16.000000
  CrossOver[1] = 32.000000
  CrossOver[2] = 64.000000
  BPF Enabled: 0
```

6m ABPF Example (15.12)

I built my ABPF kit for 80-6M with 6m as filter #4, not as specified in the builders notes.

Once off configuration:

```
usbsoftrock set bpf_point 0 8.0
```

```
usbsoftrock set bpf_point 1 16.0
```

```
usbsoftrock set bpf_point 2 35.0
```

```
usbsoftrock set bpf on
```

```
usbsoftrock set si570_multiplier 0 4.0
```

```
usbsoftrock set si570_multiplier 1 4.0
```

```
usbsoftrock set si570_multiplier 2 4.0
```

```
usbsoftrock set si570_multiplier 3 1.33333
```

This will give me subharmonic sampling ($/3$) on 6m and allow direct frequency setting as follows:

```
usbsoftrock -m 1 -a set freq 50.050
```

i.e. set CF to 50.050 received = 66.7333 Si570 output

-m 1 disables any multiplication in usbssoftrock and lets the circuit sort things out¹⁴.

Interactive Mode

```
usbssoftrock -a interactive
```

will enter a **curses** based screen enabling continuous tuning in 100kHz, 10kHz, 1kHz, 100 and 10Hz with the **q, a, w, s, e, d, r, f, t, g** keys as in **PowerSDR**.

x will exit.

p will toggle ptt on/off.

UDP Interface

Specifying the **-d flag** will put usbssoftrock into a mode where it will listen by default on **port 19004** for UDP datagrams. The port can be changed by the **-p** option.

A subset of features is available as commands (to be sent individually in each datagram):

- **set ptt on | off**
- **set bpf on | off**
- **set freq xxxxxxx**
- **get freq**
- **quit**

usbssoftrock will respond to a UDP request with a datagram beginning with either ok or error. In the case of get freq, the frequency will be after the ok i.e. **ok 14.080**

This interface is deliberately intended to be similar to the update/command protocol in the AB2KT branch of DTTSP in order to simplify writing clients.

¹⁴**How** is left as an exercise for the interested reader!