

# QtSoundModem

## Introduction

QtSoundModem (QtSM) is a multi-platform port of [UZ7HO's SoundModem](#). Many thanks to Andy for allowing me to convert his excellent software. There are versions for Windows and Linux. If you are not familiar with SoundModem I suggest you look at the User Guide on Andrei's site before trying to use this program. I've only documented differences from the original program.

**This is Beta Software - don't be suprised if it doesn't perform perfectly**

## Installation

The program can be download from:

<http://www.cantab.net/users/john.wiseman/Downloads/Beta/QtSoundModem> Binary for x86 Linux systems

<http://www.cantab.net/users/john.wiseman/Downloads/Beta/piQtSoundModem> Binary for Raspberry Pi

<http://www.cantab.net/users/john.wiseman/Downloads/Beta/QtSoundModem.zip> Program and Qt runtime for Windows.

The program uses the [Qt Framework](#), so needs some runtime files.

The Windows .zip file contains all the needed Qt Files, so you just need to unzip to an appropriate place. It is possible on some older versions of Windows that you may need some compiler runtime files, eg MSVCP140.dll. If you get a message about these being missing when you run the program download and run either

[https://aka.ms/vs/16/release/vc\\_redist.x86.exe](https://aka.ms/vs/16/release/vc_redist.x86.exe) or

[https://aka.ms/vs/16/release/vc\\_redist.x64.exe](https://aka.ms/vs/16/release/vc_redist.x64.exe)

On Linux you may need to install the Qt5 package if you don't already have any other Qt5 programs on your machine:

```
sudo apt-get install qt5-default  
sudo apt-get install libqt5serialport5
```

If you need to run on other Linux or similar systems you may be able to compile from source. First you need to install Qt Serial Port support, which isn't part of the standard Qt installation, the pulseaudio headers and an fft library:

```
sudo apt-get install libqt5serialport5-dev libfftw3-dev libpulse-dev
```

Then download

<http://www.cantab.net/users/john.wiseman/Downloads/Beta/QtSMSource.zip>, unzip and run qmake then make

QtSoundModem keeps it's configuration information in a .ini file, which is saved in the directory QtSoundModem is started from.

## Differences from original SoundModem.

### General

QtSM is generally compatible with Soundmodem V 1.08, including FX.25 features. All modem types except BFSK FEC 4x100 are supported.

The Filter parameters are not configurable.

### Sound Card Handling

Sample Rate is fixed at 12000

When running two channels the original SoundModem uses the left audio for the first and right for the second. Mine allows either channel to be allocated to either side. As a lot of radio interfaces are actually mono this makes little difference but does mean if you can access the two sides separately you could run QtSoundModem on one and another mode on the other. If you have both channels running on the same side you only get one waterfall.

### PTT

As well as the normal RTS/DTR on a COM port QTSM supports:

Using PTT pins on sound card interfaces based on the CM108 or CM119 Chips

Using GPIO pin(s) on platforms like the Raspberry Pi.

CAT Commands. You need to specify (in hex) the strings to be sent to the radio for PTT on and off.

HAMLIB (rigctl)

### Running without the GUI

QtSM can run without the graphics display. Start with command line parameter nogui.

This works on both Windows and Linux, but on Windows the only way to close the

program is to kill it using Task Manager.

It is possible to configure the program by editing the QtSoundModem.ini, but that is far from easy, so it is better if you can run the program in gui mode first, configure, then restart in nogui mode.

John Wiseman G8BPQ

April 2020

Updated April 2022