

Using the QSSTV-sBitx Edition

I use it with an external hdmi monitor and have not tried it on the 7" display.

Anyone who uses the sBitx 64 bit v4.2 - ELF Image published by the development team will have noticed that QSSTV is also included among the radio programs. The original application had to be modified in several aspects, and we can thank Willi W4WHL for the modifications.

The sBitx controller is the RPi, which also has to handle the operation of the radio and the program that transmits the image. The QSSTV connects to the radio via an audio interface.

Everything is in one place here, so this connection had to be solved in the software.

After this was done, it turned out that the reception signal needed to be larger to have a visible image.

For this, the incoming signal, sync, had to be amplified, which was implemented in the program with automatic gain control.

The output signal during transmission was too high, so there is a slider in the qsstv Configuration/ Sound, with which the transmission level can be reduced to to the level required for transmission.

Important warning: sometimes, due to bad habits, we handle the mouse and buttons incorrectly, which results in the screen freezing. In such a case, close the qsstv program and restart the application.

Continue where you left off.

I managed to make 8 sstv qso in a row, the program never crashed.

To do this, you need to prepare the images, the Templates and do not double-click anywhere, even by accident.

The program stores the received and given images and templates in a directory with the lowercase 'qsstv'.

You can see the received images in the Gallery, where it is worth using the right mouse button and deleting unnecessary images containing only noise.

After this introduction, the details:

The program can work in SSTV and DRM modes.

In SSTV mode, it has three selectable tabs, which are the Receive, Send and Image Gallery options.

Now we will deal with SSTV mode.

We start the program by starting sBitx and selecting the frequency (typically 14.230 kHz) and DIGI mode.

The frequency display should be visible on the login screen, then there is a CAT connection, and we see something similar on the startup screen (not an error message).



When starting QSSTV for the first time, we set the window size we want to see, then enter the Configuration (Option/Configuration) and fill in the data necessary for operation. The Sound and CAT data are filled in, if they are missing, you can see the necessary information in the image:

The screenshot shows the 'Configuration' window with the 'Audio Interface' tab selected. The 'Input Audio Device' is set to 'plughw:CARD=Loopback,DEV=1 -- Loopback, Loopback PCM'. The 'Output Audio Device' is set to 'plughw:CARD=Loopback_1,DEV=0 -- Loopback, Loopback PCM'. The 'TX Volume' is set to 100. The 'Input Clock Frequency' and 'Output Clock Frequency' are both set to 48000. The 'Sound Input' section has 'From sound card' selected. The 'Sound Output' section has 'To sound card' selected. The 'Maximum recording size (in MB)' is set to 100. The 'Cancel' and 'OK' buttons are at the bottom right.

The screenshot shows the 'Configuration' window with the 'Special Serial Port' tab selected. The 'Enable PTT serial interface' checkbox is checked. The 'PTT Serial Port' is set to '/dev/ttyS0'. The 'Hamlib Control' section has 'Enable Hamlib Cat Interface' checked. The 'Radio Model' is set to '2 Hamlib,NET rigctl'. The 'Parity' is set to 'None', 'Databits' to '8', 'Baudrate' to '9600', 'StopBits' to '1', and 'Handshake' to 'None'. The 'PTT Control via' section has 'CAT (Data port)' selected. The 'Restart CAT Interface' button is highlighted in green. The 'XMLRPC Interface' section has 'Enable XMLRPC Interface' checked. The 'Port' is set to '4532'. The 'TX on Delay' is set to '0.0' in seconds. The 'Cancel' and 'OK' buttons are at the bottom right.

Before receiving the image, it is advisable to set sBitx Volume=40, IF=50 to around and prepare by selecting agc=fast-mid-slow.

In qsstv, the reception level and its range can be fine-tuned with the Max dB, Range, Avg settings shown at the bottom of the right side. You can check this in the incoming image and the waterfall window, depending on the propagation and the quality of your antenna.

It also affects the image quality, so it is advisable to check Auto Slant, which is intended to correct the skew of the image. Sensitivity= DX and Mode= Auto or whatever you selected.

Be patient, because as you can see, there are several options for the correct setting.

After we have managed to take a picture and we think we want to send it, we must first go to the transmission window, where we can make the preparations. The selected picture is stored in the TX Stock (folder icon), for which we can create a caption with the Template editor (paint palette icon).

After creating a Template, do not forget to save it (as an image, when we only want to send a picture with a permanent caption, or as a template, when it also contains macros). In the transmission window, there are some fields to fill in from the keyboard, which we will see on the transmission screen if the Template box is checked. The use of macros helps in filling in the Template, so before transmitting, you only need to fill in the station's callsign, name and report before clicking the transmit button.

It does not affect reception if you fill in the transmission window while receiving the picture, as long as sufficient information is available. This way you can immediately respond to the station you see.

If you want to send the received image back in the image you have compiled, you can either respond immediately after the received image has been built or you can do it by clicking on the image shown in small size in the transmission window and searching for it among the received images.

Note the mode of the received image and the time of the image reception, then you will be successful, because this is how the image is stored in the folder.



There are many small tricks and options that you need to try out. It all depends on how useful you find this mode to spend your time.

What may be important: I made some modifications to the sBitx I am using before using it for this purpose. The output stage FETs in the device heat up during sustained PTT, and this heat must be dissipated. There is a large heat sink and it can be cooled with an external fan, and the RPi can also heat up. In my case, the output stage has been replaced with AlN (Aluminum Nitride) with better thermal conductivity instead of mica insulators, and a quiet fan cools the heat sink. Furthermore, the RPi4 4GB is installed, cooled by a quiet 50x50x10 mm fan. The pursuit of perfection has narrowed down to the best solution.

QSSTV author Johan ON4QZ wrote about using the program here:
<https://www.qsl.net/on4qz/qsstv/manual/index.html>