

# The HRC SSTV Notebook



This notebook is intended as a practical quick start guide to the great world of amateur slow scan TV (SSTV). Hopefully you will find this an interesting place to find out what you need to know to get on the air, have some fun and learn more about another aspect of amateur radio.

You don't have to read this notebook from beginning to end, in fact we encourage you to skip around. Use the hyperlinks and bookmarks to speed you on the way to enjoying SSTV.

SSTV is a simple way to transmit images, but that doesn't really cover it well enough – perhaps it is quicker to establish what it isn't! The only similarity between SSTV and everyday commercial TV is that they are both used to transmit images. However, commercial broadcast TV and the amateur equivalent, ATV (Amateur TV), are used to transmit moving images, whereas SSTV is a means to transmit still images over narrow and often noisy wireless communications channels. With SSTV you can have a QSO where you can also share pictures of the shack, QTH, areas of local interest and even immediately exchange QSL cards.

We hope that you will find this notebook to be a rich source of information on the topic of SSTV, but more than anything we want you to give it a try. To accomplish that we have set up the document to get you going first, and provide more detail later.

Compiled and edited by

David, KI6AWR

with help from

Carl, W6KGO,

Shawn, KG6ZHC

Tips and inspiration from John, KE6PID.

# TABLE OF CONTENTS

<b>GETTING READY FOR SSTV .....</b>	<b>3</b>
HARDWARE .....	3
<i>Transceiver</i> .....	3
<i>Interface</i> .....	3
<i>PC</i> .....	3
SOFTWARE .....	3
<b>MMSSTV SET UP .....</b>	<b>4</b>
BASIC SETUP (RECEIVE) .....	4
ADVANCED SETUP (RECEIVE) .....	5
BASIC SETUP (TRANSMIT) .....	6
<b>GETTING ON THE AIR .....</b>	<b>7</b>
OPERATING FREQUENCIES .....	7
RECEIVING SSTV WITH MMSSTV .....	7
TRANSMITTING SSTV WITH MMSSTV .....	8
<i>Image capture</i> .....	8
<i>Loading Images</i> .....	8
<i>Selecting the mode</i> .....	9
<i>Sending Images</i> .....	9
<i>Image History</i> .....	10
<i>SSTV on VHF</i> .....	10
<i>SSTV on HF</i> .....	10
<i>Signal Reports</i> .....	11
THE FINER POINTS .....	11
<i>Aspect Ratio</i> .....	11
<i>Stretching</i> .....	11
<i>Cropping</i> .....	11
<i>Rotated images</i> .....	12
<i>Picture quality adjustment</i> .....	12
<i>Adding your callsign</i> .....	12
<b>NUTS AND BOLTS .....</b>	<b>13</b>
THE ROOTS OF SSTV .....	13
SCANNING SYSTEM BASICS .....	14
<i>Raster scanning</i> .....	14
<i>Resolution, bandwidth and bit rate</i> .....	15
COLOR TV BASICS .....	16
CLOSE UP ON SSTV MODES .....	17
<i>Analog modes</i> .....	17
<i>What SSTV signals are made of</i> .....	18
<i>Digital modes</i> .....	19
WHERE DID MY RESOLUTION GO? .....	20
<i>The audio subcarrier</i> .....	20
<i>The audio channel spectrum</i> .....	21
<i>A better compromise?</i> .....	23
WHAT IS SLANT? .....	24
<i>What causes slant?</i> .....	24
<i>How bad can it get?</i> .....	24
<i>How can I correct slant?</i> .....	24
HOME BREW INTERFACES TO VARIOUS TRANSCEIVERS .....	26
<i>FT-817</i> .....	26
<i>FT-60R</i> .....	27
OTHER RESOURCES .....	28
<i>On line</i> .....	28
<i>In print</i> .....	28

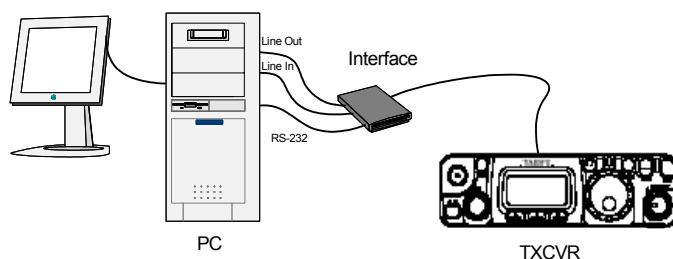
# Getting Ready for SSTV

SSTV today is divided into two broad categories; analog and digital SSTV. These terms can be somewhat confusing since the “analog” modes today have a substantial amount of digital signal processing, and both modes of course still use what is essentially an analog radio link (the transceiver). We will explore the differences later, so in the meantime this section will get you started with the analog modes.

Assuming that you have a transceiver, working antenna and a PC you already have most of what you will need. The following section fills in some of the blanks – please read on.

## Hardware

From the hardware perspective an SSTV station consists of a PC, an interface a transceiver and of course an antenna.



**Figure 1. Essence of an SSTV station**

## Transceiver

Because SSTV signals are designed for use in a voice channel, almost all transceivers designed for phone operation can be used. All that you need is the ability to connect the transceiver (rig) to the PC sound card and provide a means for the PC to switch the transceiver into transmit mode. Great results have been routinely made with base station transceivers, mobile rigs and also HTs. A wide variety of transceivers are in use with SSTV without problem. If you are looking for obstacles, look elsewhere!

## Interface

The interface provides the means to connect the PC to the transceiver. Connecting a sound card to a rig is not complex and doesn't require much in the way of specialized electronic skills. If you want to build your own interface see the section [Home brew interfaces](#) later in this notebook. There are several off the shelf commercial offerings that will work well. Note that this interface will also work for the popular PSK31 "digital" mode.

## PC

Most recent PCs will work very well for analog SSTV. The software popularly used in the Hayward Radio Club (MMSSTV to be discussed in the next section) was written specifically for the Windows Operating System. Unfortunately as with other specialized applications, software for other operating systems is harder to come by. In addition to the choice of operating system, the PC will need to have a good sound card. This can be a problem for laptops, which until recently have tended to have sound cards that are not quite good enough for the demands of SSTV signal processing. All is not lost however as external sound cards connecting to the PC via a USB connection are fairly inexpensive and can work well.

## Software

There are a variety of different software packages available for analog SSTV. The one used most commonly on the Hayward net is MMSSTV which is freeware downloadable by clicking on the following link: <http://mmhamsoft.amateur-radio.ca/>

The program was written by Makoto Mori, JE3HHT and is an easy to use, well written reliable program that runs on PCs using the Win95/98/ME/NT/2000 and Windows XP operating systems. Thanks a lot Makoto- this is where the real magic is!

# MMSSTV Set Up

Although there are several other software packages, to keep this note simple we will be focusing only on MMSSTV and how to use it.

MMSSTV is a great example of software using DSP in your PC. The soundcard in the PC is used to convert from analog to digital for the received images, and digital to analog when you are sending images. DSP is used to encode and decode the tones for the AFSK modulation, and the PC is used to store and display the images.

Download the software, install and open the application and you should see the main control panel as shown below.

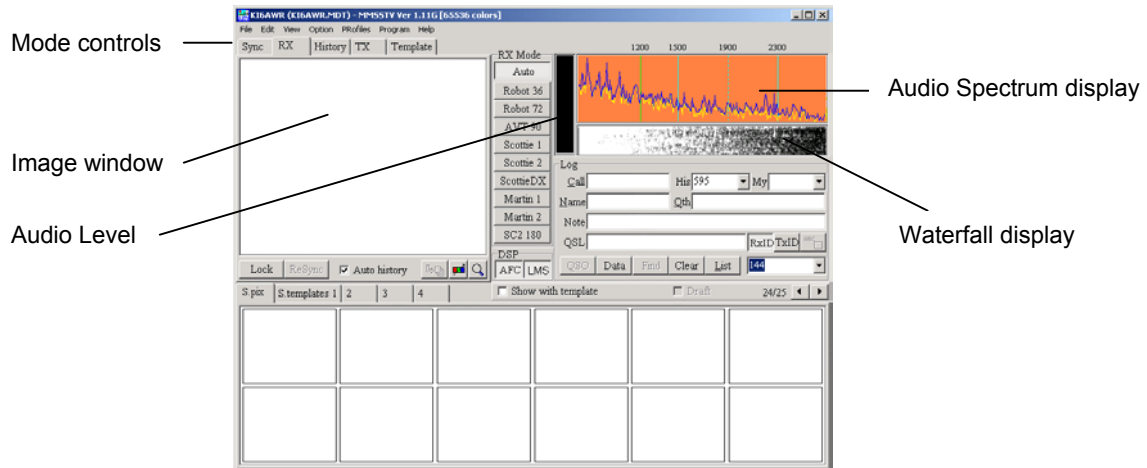


Figure 2. MMSSTV Receive screen (blank)

The software defaults are good enough to get started without any set-up (that can be done later).

For instant gratification, connect up the interface between the transceiver and the PC, tune into an SSTV transmission, click the RX button (mode controls in Figure 2), and watch the picture appear in the image window.

## Basic Setup (Receive)

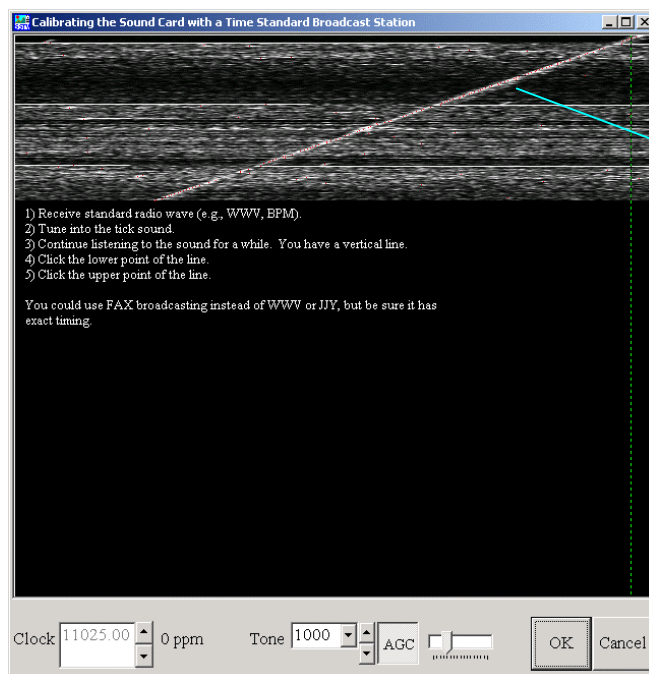
MMSSTV is very unfussy, particularly when used on VHF with FM, but nonetheless there are some setup parameters that should be set to optimize performance. These settings are provided through the MMSSTV Option/SetupMMSSTV page as described below:

1. Audio Input Level
  - Click on the **Options** tab
  - Select **Soundcard Input Level**
  - While receiving an image, set the line in or microphone level controls so that the Audio Level bar (Fig 2) is about 50% If you can't find an image find a voice transmission and set the level to about 30%.
2. Auto Slant (clock offset compensation)
  - Click on the **Options** tab
  - Select **"Setup MMSSTV"**
  - Click on the **RX** tab
  - Check the **"AutoSlant"** check box (more on that later in the section - [What is slant?](#) )

## Advanced Setup (Receive)

The clock rate of the soundcard is critical to correct operation of the DSP. The clock rate of the sound card cannot be adjusted directly, but the DSP can compensate for a clock which is off frequency. The following procedure is a part of the MMSSTV program and provides a simple and accurate way to establish the amount of compensation required:

- Connect the audio output of an HF receiver to the audio input of the soundcard
- Tune the receiver to WWV on 2.5, 5, 10, 15 or 20 MHz.
- Click on the MMSSTV **Options** tab
- Select **"Setup MMSSTV"**
- Click on the **Misc** tab



This line is far from vertical!

**Figure 3. Soundcard compensation window (clock offset)**

- Click the **"Adj"** button in the clock section at the bottom of the **Misc** page (another window will open).
- Allow the display to show for several minutes. You should have a display similar to that shown below.
  - Use the mouse to position the cursor at the bottom of the line and left click
  - Use the mouse to position the cursor at the top of the line and left click
  - Click OK
- If the amount of adjustment is small then you may continue using the program. If however it is large (as above), then you should close MMSSTV and restart it. If there was a lot of adjustment required then it is recommended to perform the procedure again to fine tune.
- For more information on slant see [What is slant?](#)

## Basic Setup (Transmit)

A few simple steps are required to get the transmit side of the system working.

1. PTT control
  - Click on the Options tab
  - Select **"Setup MMSSTV"**
  - Click on the **TX** tab
  - From the PTT port drop down box select the serial port you are going to use to activate the transceiver PTT (usually but not always COM1)
  - **OR**  
If your transceiver has VOX capability you can key it by using the MMSSTV VOX option. Click the **VOX tone** "Standard" radio button to enable this feature
  - Check the **"Exclusive lock"** check box
  - From the **"Tune button"** drop down box select 1200 (this is useful but not strictly necessary)
  - In the **"Template:"** text entry box enter your call sign (useful if you want to have CW ID)
  - In the CW ID box check the **"CW"** radio button
  - Set the speed control to just past the mid point of the Fast/Slow slider control (fast is really fast)

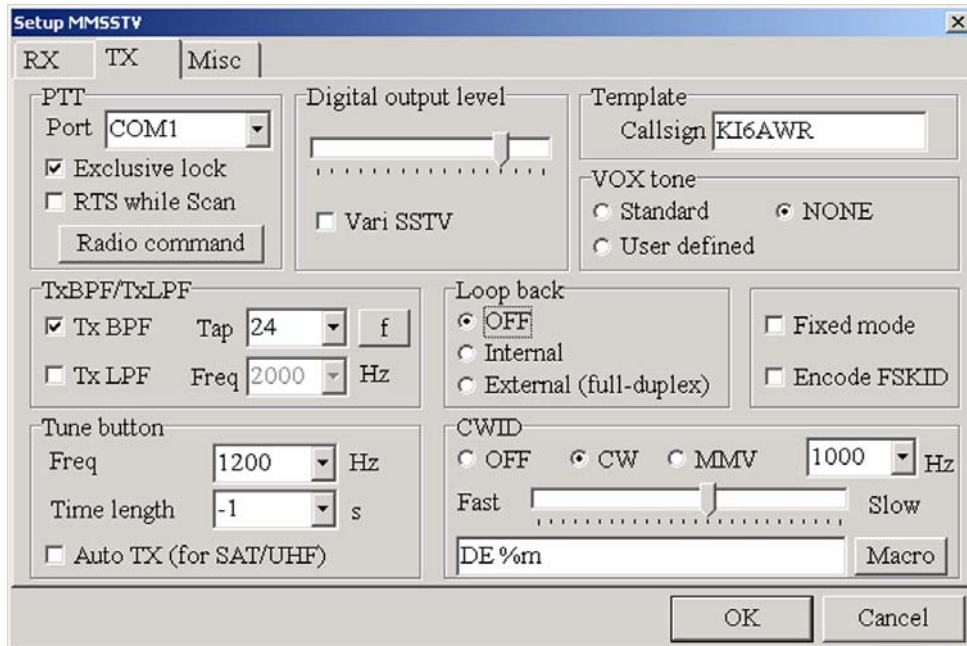


Figure 4. MMSSTV TX Setup page

# Getting on the air

## Operating Frequencies

Where to listen? Here is a list of frequencies by band to check out for SSTV activity.

Band	F (MHz)
160	1.890
80	3.845
40	7.171
20	14.230, 14.233
15	21.340
10	28.680, 28.690, 28.700
6	50.680
2	145.500, 145.600 (simplex)

## Receiving SSTV with MMSSTV

MMSSTV will start to receive automatically as soon as it detects a valid SSTV signal as long as the **Auto** button is set on the RX page (Figure 2). It determines this based on the setting of the Auto start section of the Set MMSSTV tab (center of Fig 5).

Be sure also to make sure that the "Auto" button has been selected on the SSTV main page and the program will automatically switch to the correct mode at the beginning of the transmission.

If you don't see the received image make certain that you are looking at the RX page by clicking on the RX button on the main screen.

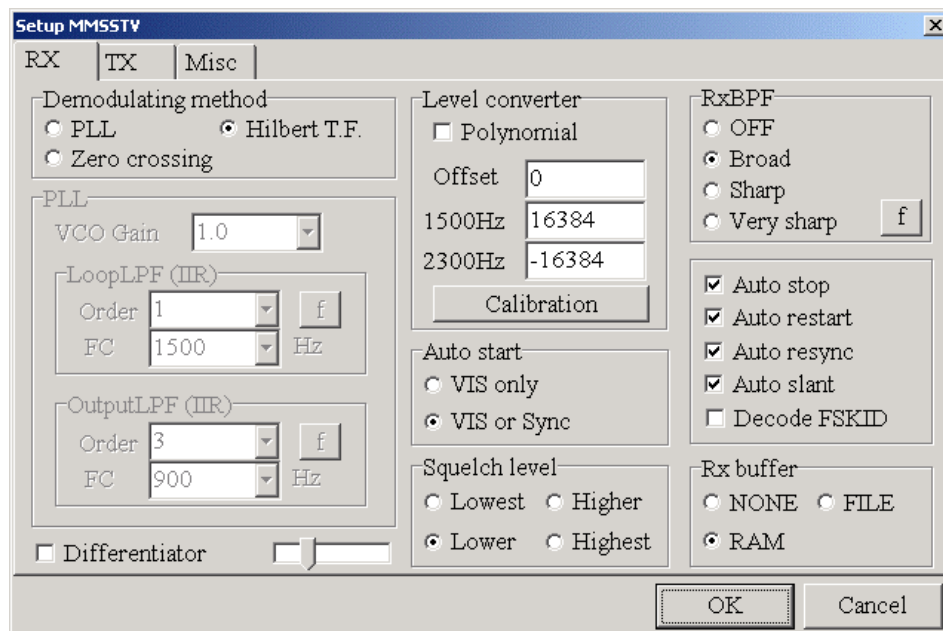


Figure 5. MMSSTV RX Setup screen

## Transmitting SSTV with MMSSTV

Transmitting images is only a little more difficult than receiving them. The added difficulties amount to loading the images, and selecting the appropriate mode.

### *Image capture*

PCs are now very capable devices for capturing, storing and processing digital images greatly simplifying the task.

Scottie and Martin modes use the following image formats:

Number of pixels per line	320
Number of lines per picture	256
Aspect Ratio	5:4
Native file size (uncompressed)	246 kbytes
Formats	JPEG, BMP

You can use any of the normal methods to import an image into the PC including:

Scanner – download film originated photos or hard copy printed material (typically via a USB port)

Digital Camera - download your own digital images. (via a USB port or digital memory card reader)

Web downloads (choose JPEG if you don't plan to do a lot of post processing)

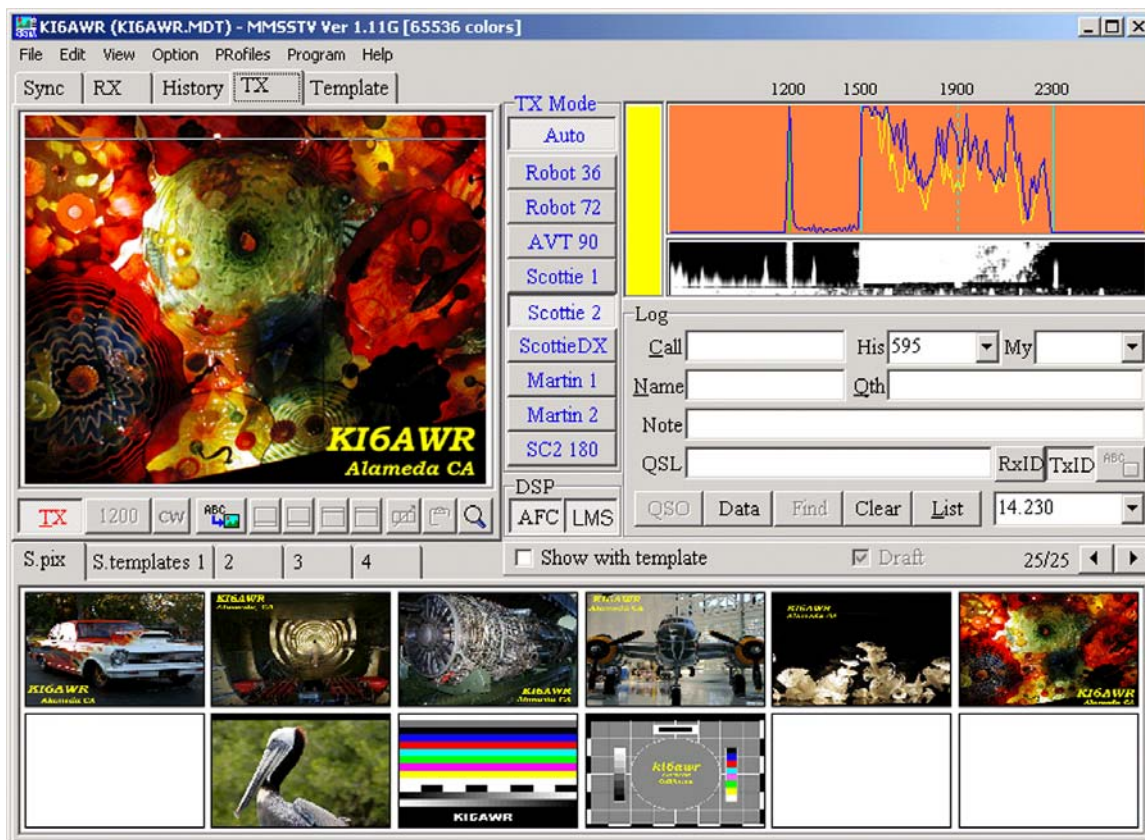
The most common type of digital image is a JPEG file and the MMSSTV program will accept this type of file directly. If your original image is a format other than JPEG or BMP (TIFF or GIF for example) you will need to convert them into JPEG or BMP before you can use them with MMSSTV. There are many photographic software packages available commercially and also as freeware that can size and adjust images and store them in JPEG format. JPEG is recommended in general as it is not proprietary and is supported by almost all PC platforms and image oriented applications.

### *Loading Images*

Martin or Scottie images are transmitted with 320x256 resolution, although 320x240 is also supported by padding the image with a small header. Images for transmission are stored as thumbnails in a pallet below the image window. To place images into this pallet, you can either drag and drop them in directly, or right click in the pallet window and choose the "Load File" option. Note that the pallet can store around 300 images in multiple pages (page size is dependent upon your display screen resolution). Scroll between pages using the scroll buttons above the pallet at the right hand end.

If the image is not the correct size a simple editor is automatically launched. This editor provides sizing and cropping capability, but it is most suitable for making minor adjustments to the image. For drastic changes in size and aspect ratio it is recommended that you use one of the many photo editing software applications.





**Figure 6. MMSSTV TX screen**

After you have loaded images into the pallet, you may either click and drag the desired image from the pallet into the TX window, or double click on the image that you wish to send and it will be automatically copied into the transmit window. (If the RX window is open, it will automatically close and the TX window will open).

### **Selecting the mode**

The TX mode selection buttons are located to the right of the image window. The column of buttons provides a subset (9) of the more popular of the 43 modes that MMSSTV supports. The list may be customized, but the default list shown in Figure 6 is usually quite adequate.

To select a TX mode, left click on the mode desired.

### **Sending Images**

Click the "TX" button and the scan should begin. You should see a thin horizontal line in the transmit window gradually moving down to the bottom of the image. This indicates the line that is currently being scanned. The transmission will automatically stop at the end of the image and MMSSTV will return to the receive mode. To stop the transmission before the end left click on the "TX" button again.

## Image History

Received images can be automatically saved to a history file. To view history click on the **"History"** tab above the receive window.

MMSTV sets an upper limit to the number of images that you can save in the history folder. The default is 32 and after you have reached this limit the earlier images will be over-written. You can increase the upper limit from the Option/ MMSTV Setup/Misc tab. Click on the History Max drop down box to set the upper limit (up to 256).

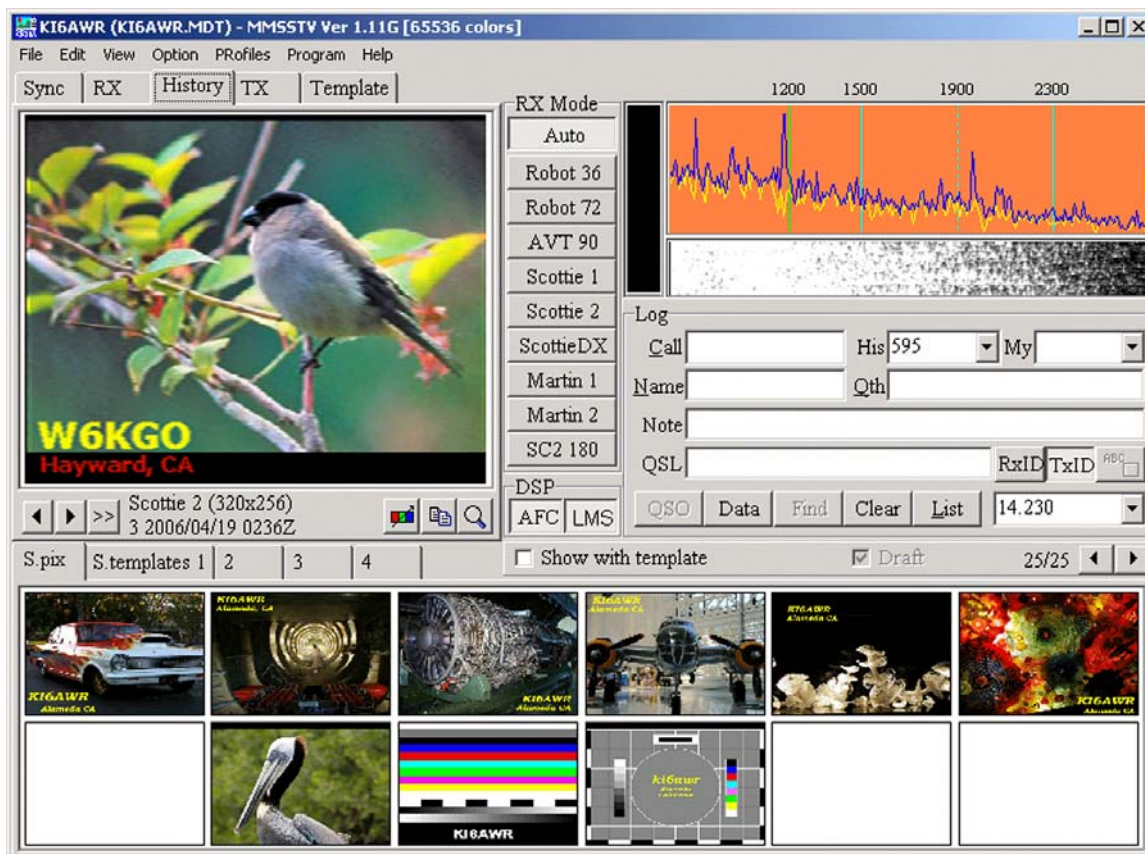


Figure 7. MMSSTV History screen

Use the horizontal scroll buttons at the bottom of the receive window to select through the stored images, or click on the **"Always show thumbnails of History"** button in the MMSSTV **View** menu tab.

## SSTV on VHF

Lack of fading and good signal to noise on VHF make it an ideal place to get started with SSTV. Operation can be accomplished using either FM or SSB. Local nets provide a forum for groups of SSTV enthusiasts to participate in image oriented round tables. The Hayward Radio Club currently operates an analog SSTV net using FM on Tuesday evenings following the 7:00 pm net on the K6EAG repeaters. This net is unique in the Bay area in that it provides an opportunity to operate on SSTV on the 6 meter band, in addition to the 2 meter and 70 centimeter bands.

For longer range contacts without using repeaters, as with phone operation, SSTV will be more successful when using SSB.

## SSTV on HF

VHF provides an easy way to get started, but it is suited only for local communications (unless Echolink and IRLP are included in the discussion). Making long distance QSOs with SSTV

normally involves using SSB on the HF bands. New challenges abound when making SSTV contacts on HF as the effects of lower signal to noise ratios, and QSB will reduce the quality of received images. The long transmission time involved in SSTV means that often it is more effective to make contact by voice first before attempting to send an image.

Operation with SSB is a little more critical because unlike FM, the tuning of the receiver will affect the frequency of the received tones. Fine tune the receiver while viewing the waterfall display and adjust tuning until the line representing the sync aligns with the 1200 mark on the MMSSTV display.

### Signal Reports

Image modes such as ATV and SSTV use a modified version of the familiar old RST system known as RSV. This is similar to the familiar RST system, but in place of the "Tone" report there is a "V" report. V ranges from 0 (noise and no image) to 1 (noisy image) through to 5 (no visible noise) with intermediate levels in between. This is a very subjective report, but when operating on 2m FM, it is quite normal to have a 595 signal report, indicating readability 5, signal strength 9 and video quality 5.

### The finer points

#### Aspect Ratio

The aspect ratio of a picture is the ratio of the width to the height. For example the aspect ratio of pictures produced by 35 mm cameras is 3:2 and digital cameras typically produce pictures with an aspect ratio of 4:3.

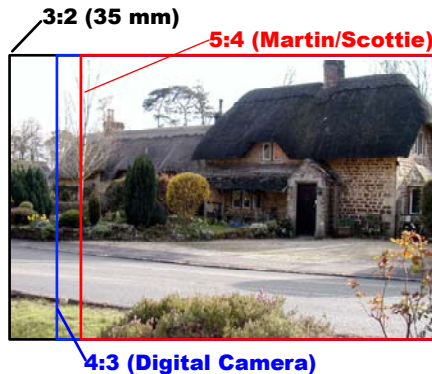


Figure 8. Sources with different aspect ratios

Scottie or Martin modes are 5:4 aspect ratio which neither film or digital cameras typically provide, so cropping or stretching of the picture will be required before it can be used. This simple operation is supported in MMSSTV by an application known as the Picture Clipper. (The Picture Clipper opens automatically if you try to drop an image into the TX pallet that is not the correct size). The figure above shows the area that will be required to crop to make it fit within the 5:4 aspect ratio. The picture clipper provides a 5:4 "crop frame" that can be moved around the image so that you can choose the area that you wish to retain.

#### Stretching

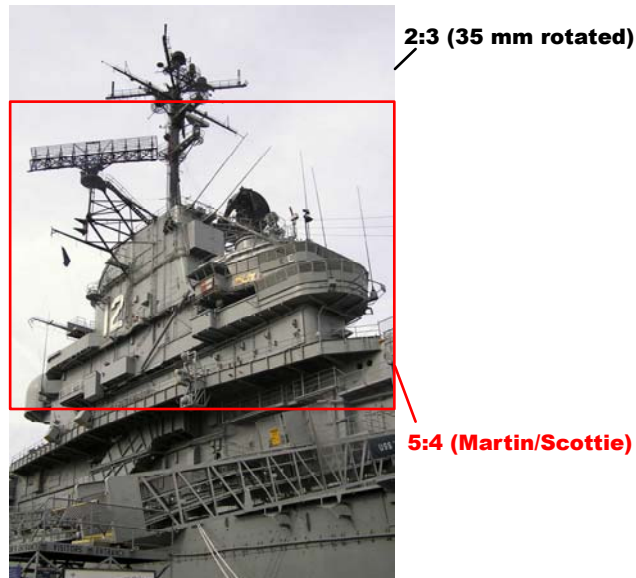
If the picture you wish to transmit is not a 5:4 aspect ratio it will have to be edited. If the aspect ratio is close to this, then you may wish to stretch the image to get it to fit. Stretching has the advantage that you will keep all of the original image, but with large amounts of stretch the image may no longer look normal, making objects too fat or too skinny. If this is the case, then it is recommended to crop (clip) the image. As you can see from Figure 8 above the amount of stretch from 4:3 to 5:4 is modest, but 3:2 might be a problem.

#### Cropping

If the picture you wish to transmit is considerably different from 5:4 aspect ratio you may have to crop the image to make it fit. Clipping (or cropping) has the advantage that you will not distort the aspect ratio of the image, but you will lose some of the original image. Cropping reduces the number of pixels in the image without manipulating the pixels that you have decided to keep. This is different than what happens when you resize (stretch or shrink) an image. When you resize an image new pixels are generated from the original pixels using filtering algorithms, and there can be a noticeable degradation in picture quality as a result.

### ***Rotated images***

The real challenge comes when using images that have been rotated. Then a 3:2 aspect ratio becomes a 2:3 aspect ratio. The Scottie and Martin aspect ratios are fixed at 5:4 with no alternative vertically oriented format. Simply cropping or stretching an image in this case typically yields poor results as shown in the following figure.



**Figure 9. Rotated 35 mm image**

One solution to the rotated image problem is to reduce the size of the image so that the image is contained within a 5:4 area, and then overlay it on a colored background. MMSSTV provides this capability, but it is difficult to manipulate accurately. If you are particular about matching panels on either side of the image (as in the figure below) you will be better off using a dedicated photo editing application.



**Figure 10. Rescaled image with side borders**

### ***Picture quality adjustment***

MMSTV provides simple editing to adjust the picture quality. This includes adjustment of contrast, brightness, hue and saturation of an image. The sharpness filter is also quite effective to improve a "soft" looking image, or to soften an image if required. Take note however that if you are using one of the "higher" speed modes such as Scottie2 or Martin2, the loss of horizontal resolution cannot be compensated for by using the sharpness control. The sharpness control affects the sharpness of both vertical and horizontal edges, but since the loss of frequency response affects only edges of vertical lines, the use of excessive amounts of sharpness may seriously over-sharpen horizontal edges.

### ***Adding your callsign***

When you are using MMSSTV you have the choice of embedding your callsign directly into an image, or adding it to a picture from a template. The best position to locate your callsign is



dependent upon the composition of the image. To ensure legibility of your callsign, you should use a relatively large font size and choose a color that is a good contrast to the image behind it. Adding a black edge or drop shadow to the callsign often improves the look and improves legibility.

Unless you are using images from a camera phone, it is almost certain that you will have to downsize the image before transmission. You will obtain best results if you add your callsign after you have downsized the original image as this will ensure better image quality. Because of the low resolution of SSTV images, it is a good idea to use anti-aliasing for the callsign. Most digital photography software programs will allow you to add text, and generally it will be anti-aliased by default. If you end up with jagged looking text, make sure that you haven't turned off the anti-aliasing feature.

## Nuts and bolts

### The Roots of SSTV

SSTV has been around for about 40 years and has been through several major steps to the point today that it works very well without the need for almost any special equipment. This section provides the essential details of the development of SSTV without exploring it in a blow by blow fashion.

The first step of SSTV began in the late 1950s. The amount of data involved in transmitting an image at the time made it prohibitive for use on the HF radio bands. Commercial TV was relegated to the VHF and UHF bands principally because each TV channel requires about 5MHz of spectrum (525 line NTSC). The compression ratio needed to squeeze a TV image into a voice channel in round figures is about 2000:1 ( 5MHz/2.5kHz) which was then a serious challenge. At the time signal processing techniques for data compression were not available so the options were basically to reduce the amount of data by limiting the resolution and also to send the data at a reduced rate. A critical issue at that time was there was no way to store a received image, and without storage viewing an image was difficult. The problem of viewing an image was solved by using CRTs with long persistence phosphors which could display an image for about 6-10 seconds. It was this time that established the maximum frame time of 8 seconds. The CRTs were designed for use with radar and because they were circular this set the optimum aspect ratio as 1:1 which means the number of pixels per line was the same as the number of lines. With techniques used at the time, the available bandwidth would accommodate about 15 lines per second (120 pixels per line). These images were monochrome only.

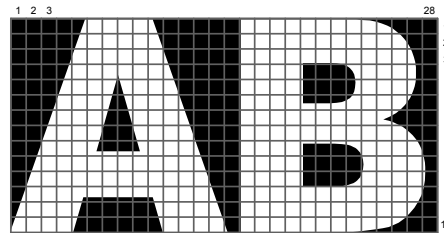
The next major step for SSTV was moving beyond the limitations set by the CRT. The addition of digital memory to store both the original and the received images enabled the images to be viewed on a normal TV set. Hardware oriented interfaces using digital ICs for storage made the capture and display much more practical and was the result of much experimentation by radio amateur SSTV enthusiasts. Enabling use with a normal TV set also opened up the possibility of transmitting color signals and the idea of transmitting color components (Red, Green and Blue) sequentially was adopted. This was accomplished neatly using field sequential scanning by equipping a black and white camera with a color wheel with red, green and blue filters. An early commercial offering was the Robot series of SSTV converters. The popularity of these devices can be seen by the number of modes still bearing the name Robot. The Robot 1200 used an early microprocessor and was consequently programmable. This enabled one of the earliest forms of hacking. Martin Emmerson (G3OQD) and Ed Murphy (GM3SBC) used the Robot 1200 as a platform to develop their own color SSTV formats, the now familiar Martin and Scottie SSTV Modes which we will review in the next section.

The final major step for SSTV was ushered in by the arrival of the PC. Initially the PC provided a convenient display device, and as a host for dedicated SSTV cards that plugged into the ISA expansion slots. More recently the incredible power available in modern PCs has enabled the use of software based digital signal processing (DSP) providing a low cost platform for storing and processing images, encoding and decoding them into a wide variety of SSTV modes as well as surfing the web, doing email, balancing your check book and all the other aspects that you take for granted today! Today's PC can support both analog or digital SSTV modes using downloadable software with both commercial and freeware offerings.

## Scanning system basics

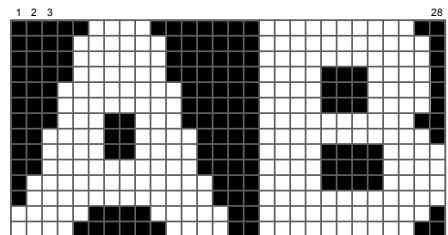
### ***Raster scanning***

TV images use a system of raster scanning invented by Philo Farnsworth in 1921 (at the age of 15). With this technique an image is divided into rows (lines) of pixels. Essentially the scanning process divides the original image into a mosaic of elements (known as pixels) something like a tiled floor, with each individual tile being a different color. The color value of each pixel (tile) is then transmitted sequentially starting from the top left of the image and reading out the first row of pixels. The process is then repeated for the next line of pixels until all lines have been sent.



**Figure 11. Original Image**

The image shown in Figure 11 represents the original image with a tile pattern (sample grid) overlaid on top of it.



**Figure 12. Sampled Image**

The image shown in Figure 12 shows the sampled image where square pixels have been substituted for each of the original "pixels". The jagged appearance is due to a phenomena known as aliasing and looks particularly bad in this case because of the very coarse sampling grid (24x14). Aliasing results from the fact that in the original image not all edges of the image fall completely inside the sample grid. This jagged appearance can be reduced by a technique known (not surprisingly) as anti-aliasing.

Digital still cameras also use the idea of rows and columns of pixels which makes conversion of digital camera images to SSTV images very simple.

## **Resolution, bandwidth and bit rate**

### **Bandwidth requirements (maximum bandwidth)**

When an image is transmitted the pixels in each line are transmitted sequentially from left to right and line by line. Each pixel is allocated a very precise amount of time based on a clock known as the pixel clock.

The finest detail in an image generates the highest bandwidth. The finest detail occurs when there is a sequence of alternating black and white pixels (horizontally). Each pixel is transmitted for a full cycle of the pixel clock which means that a black and white sequence will take exactly two pixel clocks yielding the following simple relationship:

$$\text{Maxbandwidth} = \frac{\text{Pixelclock}}{2}$$

So how do we find out what the pixel clock frequency is? Sometimes this information is available directly, but often you will have to calculate it based on the frame frequency and the total number of pixels.

$$\text{Pixelclock} = \text{TotalNumberpixels} \times \text{Frame\_rate}$$

so

$$\text{Maxbandwidth} = \frac{\text{Pixelclock}}{2} = \frac{\text{Numpixels / line} \times \text{Number\_of\_lines} \times \text{Frame\_rate}}{2}$$

For example the maximum bandwidth of Scottie2 =  $320 \times 256 \times 0.14/2 = 1.728 \text{ kHz}$

NOTE: To keep this discussion simple we used the number of active pixels per line in place of the total number of pixels per line. The total number of pixels includes the sync and blanking intervals, but these are generally small. In this case the bandwidth is underestimated by about 4%.

### **Effective bit rate**

Another interesting metric is the effective bit rate that these modes provide. We can evaluate this easily enough by recalling that the image has a fixed number of pixels, and a defined period that it takes to transmit them.

Using Scottie2 as an example the total number of pixels is  $320 \times 256 \times 3 = 245,760$  pixels per frame which is just under 2 Mb.

Since it takes 71 seconds to transmit, the effective bit rate is 27 kbps. ScottieDX in comparison is 2 Mb/269 seconds, about 7,400 bps.

How does Scottie2 get such a high bit rate? – well in reality it doesn't! (see [Where did my resolution go?](#) for more information).

## Color TV basics

Raster scanning solves the problem of sending an image electronically, but initially it supported only the transmission of monochrome images. To emulate black and white photography, these monochrome images were displayed originally on CRTs with a white phosphor and it became known alternatively as Black and White Television. How then is color information transmitted?

Psycho-visually, color information is the perception of light of different wavelengths. Surprisingly it was discovered that a mixture of only three different wavelengths of light provides a good simulation of an image with a continuous spectrum of colors. The three colors used were red, green and blue (RGB). Other colors in the visual spectrum can be produced as a mixture of these three primary colors. For example the color yellow can be synthesized by a mixture of red and green colors. This is known as additive mixing.

The scientific definition of these colors was carefully described and provides a reasonably satisfactory result.

Having discovered that red, green and blue images projected to the eye will provide an acceptable color image what remains to be done is to originate and display these three color components.

The origination of the color components for natural images is accomplished by splitting the original image into red, green and blue components which is accomplished by optical filters placed in front of three imaging devices.

In the case of SSTV, the three images are transmitted as if they are three monochrome images sent one after the other. Early analog SSTV formats transmitted a complete red image, followed by a complete green image and finally a complete blue image. These were stored at the receiver and after all three frames had been received were displayed as a single color image. This is known as a frame sequential color system. Later analog SSTV modes (Scottie and Martin for example) use a line sequential color system. In this method a line of the red image is transmitted, followed by a line (same line number of course!) of the green image, followed by a line of the blue image. This process is then repeated for the next line number until the whole frame has been transmitted (Note that due to a design quirk in the development of the Martin modes the order of transmission for Martin modes is G,B,R, not R,G,B).

At the receiving end, the three color components are displayed by illuminating colored pixels or colored phosphors so that the eye perceives a single colored image.

Some color TV systems (ATV and SSTV for example) use a more efficient method of encoding the color information. Instead of transmitting the red green and blue components directly, these three components are combined in a specific way to produce three new components known as luminance and color difference signals. Advantage can be taken of the fact that the human eye perceives details primarily in terms of differences in intensity, which is represented by the luminance signal. The luminance signal is essentially a black and white signal and the color difference signals are two additional signals that provide the information describing the color in the image. Because the eye gleans information about edges and fine lines from the luminance signal, the color difference signals can be filtered to reduce the bandwidth required for each of them without significantly affecting the displayed result. At the receiving end, this the encoded signal is then decoded back to RGB and voila – a sharper looking image without using more bandwidth.

Formats such as MP73 and some of the early Robot formats exploit this convenient aspect of the human psycho-visual system. Your science \$\$ at work!



## Close up on SSTV modes

### Analog modes

Technological advances in SSTV transmission have been accomplished over a period of almost 40 years and as a result there is a very long list of analog SSTV formats. Most of these have fallen into disuse so today this is easily simplified into a few very popular formats.

The two most popular analog SSTV medium format families (known as modes) are:

- Scottie
- Martin

The Scottie modes are attributed to Ed Murphy, GM3SBC, and the Martin modes are attributed to Martin Emmerson, G3OQD. Each of these modes have several variants suited for different applications, and both provide full 24 bit color with 320 x 256 resolution images.

Scottie modes

- Scottie DX
- Scottie 1
- Scottie 2

Martin modes

- Martin 1
- Martin 2

Why are there several variants of each format?

The answer is that one size doesn't fit all. The rate at which information can be transmitted is related to the signal to noise ratio of the communication channel used. In amateur radio this can range from very good, VHF FM for example, to very poor, HF DX for example. The different variants are designed to cope with different types of use. The DX variants are designed to work with signals having QSB and relatively low signal to noise ratio typically of DX contacts.

They support the same image resolution, but transmit at a slower rate to provide as much opportunity as possible to discriminate the signal from the noise.

Mode	Comment	VIS	Resolution	Lines/min	TX time (sec)
Scottie DX	Very slow TX time, more robustness, good sharpness	76	320 x 256	57.12653528	269 +
Scottie 1	Slow TX time, pretty good sharpness	60	320 x 256	140.1148942	110 +
Scottie 2	Faster TX time, reduced sharpness	56	320 x 256	216.0067214	71+
Martin 1	Slow TX time, pretty good sharpness	44	320 x 256	134.3947532	114
Martin 2	Faster TX time, reduced sharpness	40	320 x 256	264.5525975	58

The VIS entry in the table above is for information only. This code is used by the software to automatically detect the mode and is not user configurable.

### What SSTV signals are made of

An SSTV signal is made up of several parts, that are transmitted sequentially.

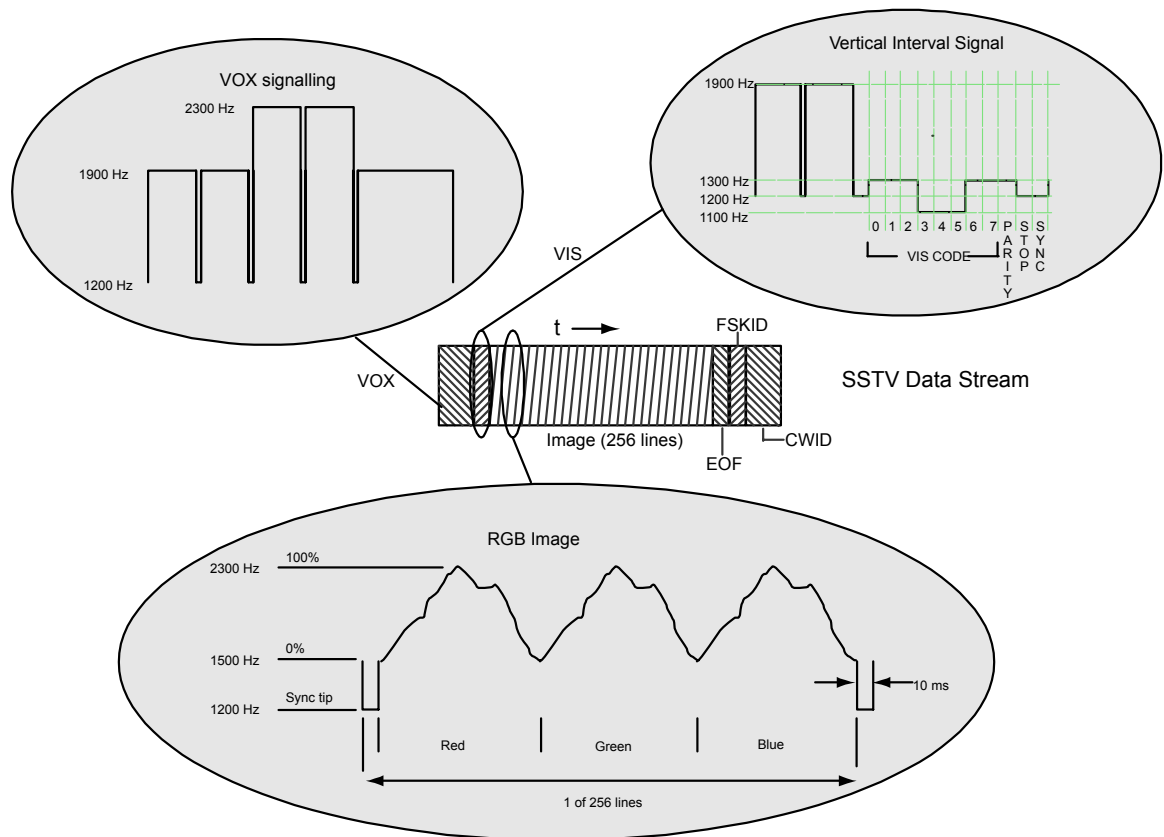
#### Header

- VOX
- VIS
- Image data

#### End of message

- EOF
- FSKID
- CWID

The header contains the Vertical Interval Signal (VIS) which is used by the SSTV encoder to inform the decoder what mode to expect, and also to signal the start of the image part of the message. The VIS code is a 7 bit binary code transmitted as using frequency shift keying (FSK) where a binary "1" is represented by an 1100 Hz tone, and a binary "0" is represented by a 1300 Hz tone. The code is preceded by a start bit and followed by a parity bit and a stop bit.



**Figure 13. Scottie2 signal components**

MMSSTV can be set up to key the transmitter using VOX. To make sure that none of the VIS is clipped by the VOX delay in the transmitter, a special VOX activation signal can be transmitted automatically prior to the VIS (see detail in Figure 13).

Following the header VIS, the image is transmitted line by line. Martin and Scottie modes have 256 lines per picture (MMSSTV also supports 240 line Martin and Scottie pictures by adding a 16 line blank header to the picture). Martin and Scottie modes use a line sequential color system in which the three different primary colors (R,G and B) are transmitted one after the other. The order in which they are transmitted is different between the Martin and Scottie modes. Scottie modes transmit a red line, followed by the corresponding green line and finally the blue line (see Figure 13). Martin modes however transmit the green line first, followed by blue and then red. The order is not important as long as the decoder operates in the order appropriate to the standard. Each sequence of R,G, B lines is preceded by a sync pulse which is used to identify the start of each line (unlike broadcast TV, this sync pulse is not used to resynchronize each line which can result in a problem known as Image Slant – see [What is slant?](#)).

At the end of the image, an end of file sequence is added to inform the decoder that the total image has been received. MMSSTV can also be set up to include additional ID information. ID information (your call sign) can be transmitted digitally using FSK, or CW, or both. If you use both methods then the FSKID period precedes sending of the call sign by CW.

### ***Digital modes***

The analog modes are simple to use and effective under situations with good signals, but the techniques used do not provide optimum performance by today's standards. Much better quality images can be transmitted reliably using digital techniques for image transmission in place of the audio FM methods used in analog SSTV. In addition, digital SSTV has no limitations on image aspect ratio. It is simply transferring a file. Since we are transferring a file, a key advantage of digital SSTV is error correction. If any picture segments in the file have errors, the receiving station simply clicks on the BSR (bad segment request) button and the transmitting station can resend just those segments that contained errors.

The dominant digital SSTV mode is called DRM (Digital Radio Mondiale). DRM is actually an open standard digital radio system (<http://www.drm.org/system/technicalaspect.php>). The amateur radio SSTV implementation of DRM is limited to a bandwidth of approximately 2.5 kHz. Using sophisticated modulation and error correction techniques, transmission speeds of 2300 bits/sec are typically achieved. Therefore, if we assume a maximum 120 second transmission time, we can send a maximum file size of 34K bytes. Fortunately, Digital SSTV programs include image compression routines that can dramatically reduce picture file size. Additional information can be found in the Yahoo digssstv group (<http://groups.yahoo.com/group/digssstv/>).

## Where did my resolution go?

If you get the chance to see your own transmitted image, you might be surprised to see that it is a little less sharp than you saw in your TX window. Why is that? Well it might just be the choice of mode that you are using. This section looks into some of the important subtleties associated with the different modes.

### The audio subcarrier

The image information has to be transferred faithfully using the voice channel of your rig, but unfortunately unlike voice (which the rig is of course designed to accomplish) it cannot be transmitted directly. Because the image data has a d.c. component, and a voice channel cannot transmit a d.c. component some sort of magic must be performed. This is a similar to the problem of transmitting digital data down a telephone line and requires the use of a modem. Although you may not physically see it, SSTV does have a built in modem (with the MMSSTV program it is a software modem running on your PC).

The encoding side of analog SSTV simply translates each pixel intensity to a specific frequency of a constant amplitude audio tone. This is a form of FM which makes it largely independent of the amplitude, a highly desirable characteristic for a radio link which is subject to QSB and QRN. This tone is known as a subcarrier because this signal will be modulated again in the transceiver on the main carrier (the transmit frequency of the transceiver).

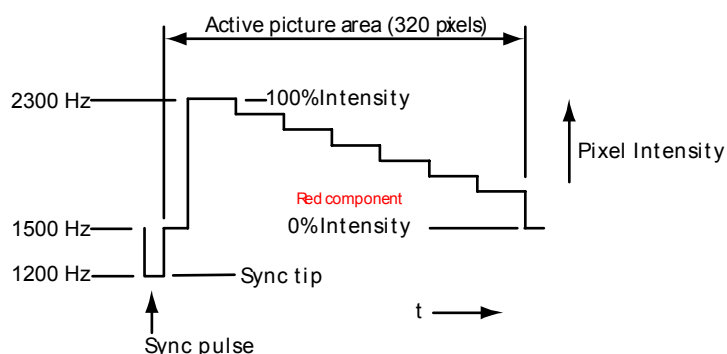
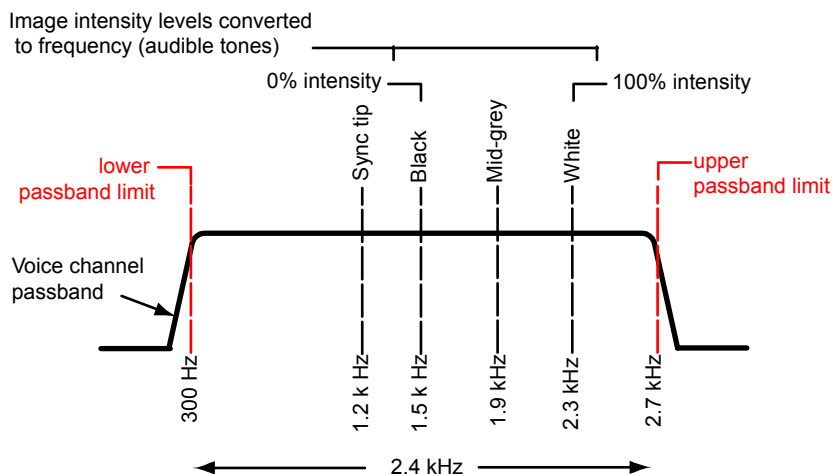


Figure 14. Red component of an SSTV line

The frequency of the subcarrier corresponds to the brightness of each pixel. It is easiest to think about this from the point of view of a black and white image. If the pixel to be transmitted is a black pixel (0% intensity) then the frequency of the subcarrier will be 1500 Hz (see Figure 14). If however the pixel is white, then the subcarrier frequency will be 2300 Hz. Any intermediate shade of grey will be represented by a tone with a frequency proportional between these two extremes. For example mid grey (50% intensity) will have a frequency of  $50\% \times (2300 - 1500) + 1500 \text{ Hz} = 1900 \text{ Hz}$



**Figure 15. Subcarrier frequencies within the voice channel passband**

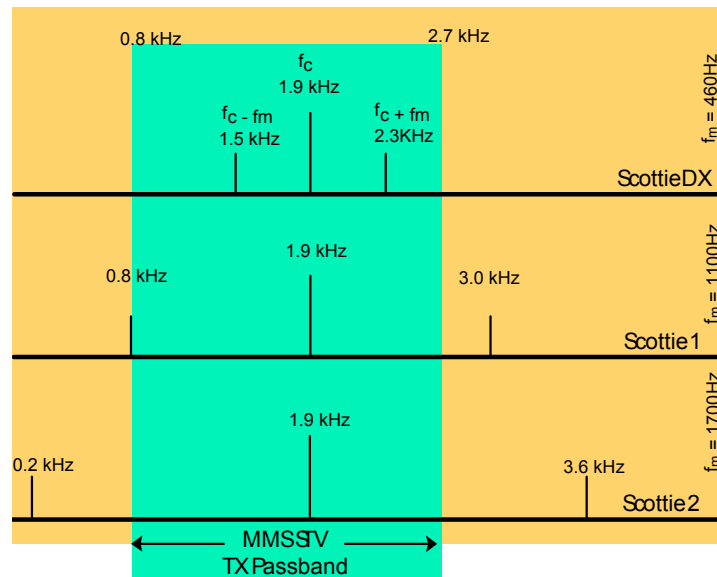
Figure 15 shows this simple scheme, and illustrates how neatly this fits into the audio channel of an SSB or FM transmitter (or even AM if you are so inclined). This method looks like it will work very nicely with the subcarrier fitting nicely into the voice channel passband, so off we go. Not to so fast though, this happy state is only true if the subcarrier is sitting at a single frequency. This would be true if the whole line or frame is a single color, but that makes for a pretty dull image. Areas of fine detail generate higher bandwidths (see [Resolution and bandwidth](#)) which changes the frequency of the subcarrier dynamically with an important consequence.

### The audio channel spectrum

Being interested radio hams we know all about FM. Remember that the amount of spectrum space that FM signals require is related to the modulation index. In this case the peak to peak deviation is the dynamic range of the video (from black to white) which is  $2.3 \text{ kHz} - 1.5 \text{ kHz} = 800 \text{ Hz}$ . The modulating frequency ( $f_m$ ) is the rate at which changes in pixel intensity can be transmitted which in the case of Scottie2 can be as high as  $1.8 \text{ kHz}$  (maximum bandwidth).

$$\text{modulation index} = \frac{\Delta f}{f_m} = \frac{400}{1800}$$

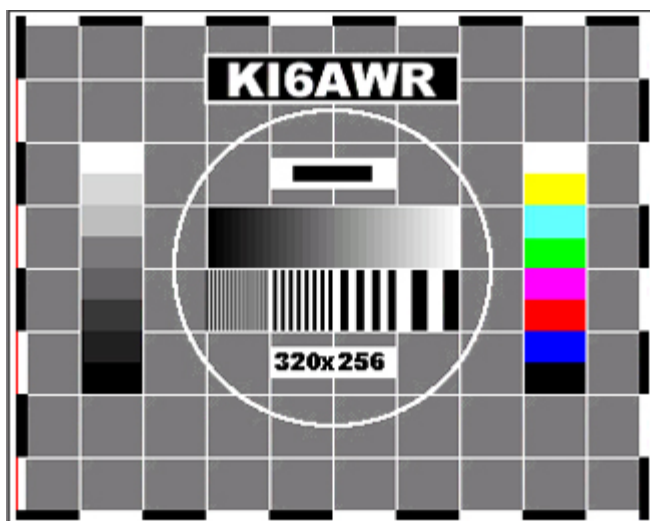
Since the modulation index is less than one, the signal is defined as a narrow band FM (NBFM) signal. The low deviation ratio of NBFM means that we need only be concerned with the first pair of sidebands (since the others will be of negligible amplitude). These sidebands are located symmetrically about the subcarrier frequency and offset by the modulation frequency  $f_m$ . At last, we have come to the problem!



**Figure 16. Frequency spectrum for three Scottie modes**

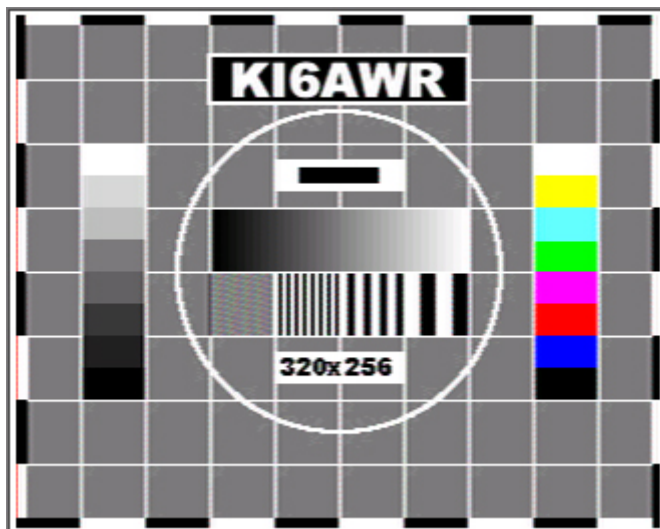
In Figure 16 we have plotted the position of the carrier and sidebands for three different modes, ScottieDX, Scottie1, and Scottie2. As you can see, when you are using Scottie2 those precious little sidebands fall right outside the voice channel passband, and with Scottie1 only the lower sideband makes it (barely). MMSSTV provides bandpass filtering on both the encode and decode side, so even before you send this signal to your rig, sidebands outside of the voice channel will be neatly disposed of. So what will you see? Since both sidebands have been very neatly chopped off, all that you are left with is the subcarrier, which is sitting happily in the middle. If you were supposed to be seeing alternating patterns of black and white bars, instead you will see plain mid-grey. All of that resolution just fell into an analog equivalent bit bucket.

The first of the figures below shows a received image using ScottieDX mode. Transmission time is very long, but the image quality is very close to the original.



**Figure 17. ScottieDX received image**

Figure 18 shows an image transmitted using the Scottie1 mode. This mode takes 109 seconds to transmit a frame. Results are not as good but reasonably acceptable. The highest resolution bars are missing but the second set of bars (2 pixels wide per bar) is reasonably clear.

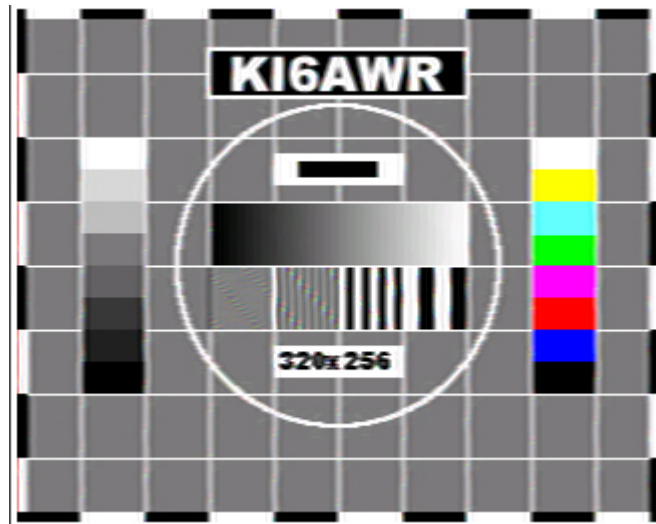


**Figure 18. Scottie1 received image**

Saving the quickest for last., Scottie2 is the fastest of the modes with a transmit time of 71 seconds. This speed comes at a price.

Scottie2 resolution is understandably the worst of the three Scottie modes as far as resolution is concerned. Figure 19 shows the received image using Scottie2.

Even though the difference between the original image and Scottie2 is very noticeable, with less demanding images Scottie2 can still deliver acceptable looking results.



**Figure 19. Scottie2 received image**

Notice that the highest resolution grating (left hand end of the resolution gratings) is completely blank and there are artifacts in the next lowest resolution gratings.

Beware! If you are thinking that you can improve the situation by pre-sharpening an image before transmission, as you are likely to be very disappointed. An aspect of video scanning systems is that the high bandwidth components are generated by the horizontal scanning system. You can see this in the figure above since the vertical lines of the grid pattern are notably less sharp than the horizontal lines. If you over-sharpen the image, horizontal edges will be over-sharpened, but vertical edges will be little improved. Not really what we are looking for.

### ***A better compromise?***

This resolution problem is caused by the scanning frequency generating a spectrum that exceeds the capacity of the channel passband. To solve the problem we need to narrow the spectrum by reducing the scanning frequency. This can be achieved by using an analog form of image compression. This is the same type of analog compression scheme that is used in the NTSC and PAL TV formats and uses a different method of transmitting the color information. Instead of transmitting the red, green and blue color components (each with the same bandwidth), a black and white image is transmitted (monochrome) together with two color difference components (with lower bandwidth). Since detail is perceived by the eye by the luminance (brightness) and not the color of an image, the color information is transmitted with 50% less resolution and the luminance (black and white image) is allocated more bandwidth.

An example of this is the MP73 format. The resolution of the color channels is 50% of the original resolution, which provides images that are as sharp as Scottie1 images (109 second transmit time) but require only the same time as Scottie2 (71 seconds), a 35% improvement in time.

## What is slant?

Slant is a distortion of the received image due to differences between sample clock of the PC at the transmitter and frequency of the sample clock of the PC at the receiver. This distortion skews the received image as shown in the figure below.

### What causes slant?

Slant is caused because analog SSTV protocols such as Martin or Scottie use an asynchronous transmission protocol. This means that the clock in the soundcard of the receiving PC is not synchronized to the soundcard clock of the transmitting PC. Because of the difference in clock rates, the effective display width of each pixel received will be effectively different from the intended width. The result of this is that with each line the image will be increasingly displaced. If the transmit clock is faster than the receive clock, the image will become slanted towards the right, and if the transmit clock is slower than the receive clock the received image will appear to slant to the left.

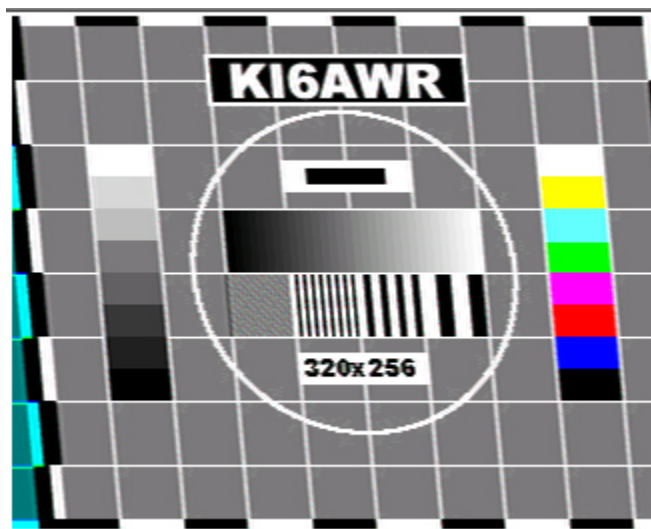


Figure 20. Scottie1 with transmit clock offset by +1Hz

### How bad can it get?

In MMSTV, the decoder uses the sync pulses at the start of the picture to align the start of the picture to the top left corner of the display. The horizontal sync pulses are then ignored (except if lock is lost and re-lock option has been selected). The time discrepancy between the transmit and receive clocks causes a displacement error that accumulates for the entire picture.

In the previous figure, note that the slant was due to a frequency difference of only 1 Hz (90 ppm) between the TX and RX clocks (nominal clock frequency 11,025 Hz.). Martin and Scottie modes each have 320 active pixels per line x 3 (R,G,B) and after taking into account sync width, this is about 1000 pixels per line. Since there are 256 lines per frame that means there are 256,000 clock cycles per frame.

An error of 90 ppm will then cause an error of  $256,000 \times 90 \times 10^{-6} = 23$  pixels. Over 256 lines, this is a skew of more than 5 degrees which is pretty impressive for a 1Hz error.

### How can I correct slant?

You can manually compensate for the error in your PC clock with a clever alignment procedure built in to MMSSTV. (See [Advanced Setup \(Receive\)](#) for a method to compensate for the offset)

Additionally MMSSTV can be set to automatically correct for small differences in the sample clock of the transmit and receive ends of the communications link. This process is known as Auto slant. The Auto slant process measures the average time between sync pulses and since this should represent a known number of pixels (320 active pixels per line plus the width of the sync pulse) the clock rate of the original signal can be estimated and the difference can be compensated for. For greater accuracy this averaging is performed over a number of lines.



Even if you have calibrated MMSTV to compensate for the offset in your sound card clock, you may find the signal you are receiving has not been compensated so it is generally best to keep this feature turned on.

## Home brew interfaces to various transceivers

### FT-817

The interfaces shown below were designed to provide low cost but effective connectivity between the transceiver and the PC sound card. The first of these interfaces was designed for the popular Yaesu 817 all mode QRP transceiver using the data input on the rear of the unit.

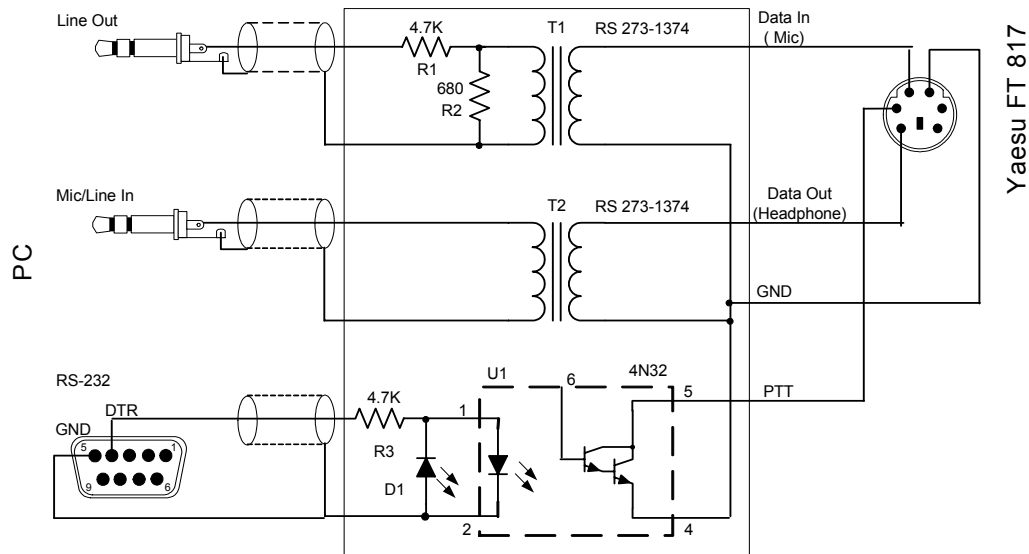


Figure 21. Yaesu FT 817 SSTV interface

The interface is designed to provide DC isolation between the PC and the transceiver to prevent ground loops that may introduce hum into the PC or transceiver. Audio signals are isolated using 1:1 isolation transformers and the control (PTT) signal is isolated using an opto-isolator.

Line out from the PC is AC coupled using an audio frequency isolation transformer (Radio Shack #273-1374). The audio level is attenuated (18dB) by R1 and R2 to match the input requirements of the FT 817.

Audio out from the FT 817 is AC coupled to the PC using the same type of audio frequency isolation transformer.

The RS-232 output of the PC controls the PTT of the FT 817. DC isolation is provided by an opto isolator. When the RS-232 output goes negative (Receive mode) diode D1 conducts to protect the opto-isolator from excessive reverse bias. Since D1 is an LED it will illuminate and so provides a simple way to indicate that the interface is in the receive (ready to send) mode. If you don't care for the status indicator, replace D1 with a 1N4148 general purpose diode.

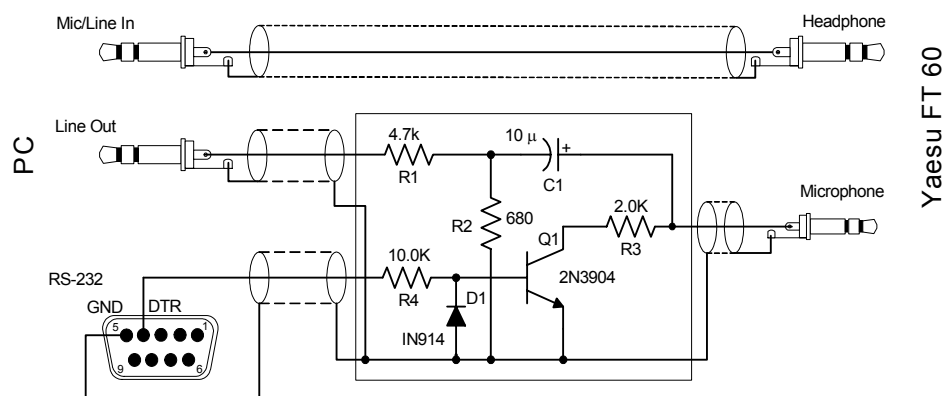
This all fits very easily into a Radio Shack project box #270-283A.

The above approach uses a com port DTR pin for PTT control. However, if your transceiver accepts serial data commands, MM-SSTV can send transmit and receive commands over the serial link. The advantage of this approach is that, in addition to PTT, you can program MM-SSTV to send commands that turn off mike gain and/or compression prior to transmitting the SSTV signal. Also, after sending the receive command at the end of the SSTV transmission, you can program MM-SSTV to restore mike gain and compression if desired for voice communication.

Finally if you have a rig with VOX capability, you can use the MMSSTV vox control in place of the PTT control and eliminate the opto-isolator completely.

## FT-60R

The following interface was designed for use with an HT operating on battery power. This is an alternative and very simple interface using direct coupling since the HT will be floating from ground. The ground connections of the PC and the rig are connected together directly which eliminates the need for isolation transformers or opto-isolators.



**Figure 22. Yaesu FT 60R SSTV interface**

This interface is designed for use with the Yaesu FT 60R hand held transceiver, but other transceivers will be similar. This uses the microphone input and headphone output of the transceiver.

Note that the audio output from the FT 60R is coupled directly into the PC line input. This can often be accomplished with a standard stereo cable.

The line output from the PC must be AC coupled to the FT 60R microphone input because the PTT operation also uses the microphone connection.

Q1 provides the level shifting needed to convert the bipolar RS-232 signal from the PC to the contact closure that the FT 60R requires.

R1 and R2 (&R3) form an attenuator to drop the audio level from PC line level by 20dB to avoid overdriving the HT microphone input.

This all fits very easily into a Radio Shack project box #270-283A.

NOTE: The FT 60R uses a four pole jack which is not readily available. In addition to building the interface you will need to purchase an adapter (Yaesu CT-44 or similar) so that you can use standard three pole (stereo) connectors.

## Other resources

### *On line*

The following web site is very useful as it has a massive number of links to all aspects of SSTV:

<http://www.kiva.net/~djones/>

Dave Jones, KB4YZ

### *In print*

A lot of information on the topic of SSTV and more can be found in the following publication:

The ARRL Image Communications Handbook

-- by Dr. Ralph E. Taggart, WB8DQT

