Proposal for SSTV Mode Specifications

JL Barber (N7CXI), Silicon Pixels

Presented at the Dayton SSTV forum, 20 May 2000

Forward

When I first began investigating the requirements for designing an SSTV system, one of the first obstacles I ran into was the lack of *accurate* information on the exact format and timings of the many SSTV modes in use. All the information I found was either generalized, or based on the results of reverse engineering existing systems. As so often noted, this has resulted in many compatibility problems over the years. This paper is a beginning at changing that. The reason I call it a *beginning* is that other developers may not agree that the specifications presented here should become standards. Even so, every venture must start somewhere. If the developer's community at large can agree on reasonable standards, we'll be happy to adopt them. In the meantime, we present these proposed specifications for use or publication by anyone who finds them useful.

Data sources

The specifications presented in this paper include only those that we have "hard" data for. In the case of Scottie, Martin, Robot, and Wrasse modes, our source is the firmware for the Robot 1200C. For other modes:

Pasokon "P" modes	John Langner	(author)
"PD" modes	Don Rotier	(author)
FAX480	Dr. Ralph Taggart	(author)

Note: Those familiar with our ChromaPIX and W95SSTV systems will note the absence of specifications for the AVT modes. Since we have no solid data on AVT from the author, we have not included them here.

Numeric precision

The timings specified here are generally given in milliseconds, (ms) except where other units work better for examples. Scan times have been rounded to .001ms, which is adequate precision for any known application. Total transmission times are not calculated precisely, since the exact lengths aren't relevant for development purposes. In the case of the Robot firmware, we can provide the 8031 clock frequency and "counts", if absolute precision is desired.

Disclaimer

While every attempt has been made to ensure the accuracy of this information, we can't absolutely guarantee it to be completely without error.

Layout

Rather than present mode timings with pictures, we've chosen to "step through" each mode, in sequence. We believe this method will be more useful to developers, as it reveals the sequence of events more readily than an image.

VIS Code and Robot calibration header

All standard SSTV modes utilize a unique digital code that identifies the mode to a receiving system. The code is called the VIS, or Vertical Interval Signal code. Although the entire calibration header is often referred to as the "VIS code", the code itself is only a part of it. The seven-bit code is transmitted least-significant-bit (LSB) first, and uses "even" parity.

Calibration header with VIS code

TIME(ms)	FREQUENCY(hz)	IDENTITY
300	1900	Leader tone
10	1200	break
300	1900	Leader tone
30	**1200**	**VIS start bit**
30	**bit 0**	**1100hz = "1", 1300hz = "0"**
30	**bit 1**	**""**
30	**bit 2**	**""**
30	**bit 3**	**""**
30	**bit 4**	**""**
30	**bit 5**	**""**
30	**bit 6**	**""**
30	**PARITY**	**Even=1300hz, Odd=1100hz**
30	**1200**	**VIS stop bit**

Note that all mode specifications begin immediately after the VIS stop bit.

SCOTTIE MODES

"Scottie" modes, although the most commonly used modes in the US, are unusual for two reasons. The combination of the two has caused considerable confusion over the years:

- The first scan line (only) begins with an out-of-sequence 9.0ms "starting" sync pulse, at 1200hz.
- The "regular" sync pulse is positioned between the Blue and Red scans, rather than at the line break, as with other modes.

VIS CODES

Scottie 1	60d (decimal)
Scottie 2	56d
Scottie DX	76d

COLOR MODE	RGB (1500-2300hz luminance range)
SCAN SEQUENCE	Green, Blue, Red
NUMBER OF LINES	256
NORMAL DISPLAY RES	320x256 (including 16-line header)

TRANSMISSION TIME

Scottie 1	109.6 seconds (not including cal. header/VIS)
Scottie 2	71.1 seconds
Scottie DX	268.9 seconds

COLOR SCAN TIME

Scottie 1	138.240ms	(.4320ms/pixel @ 320 pixels/line)
Scottie 2	88.064ms	(.2752ms/pixel @ 320 pixels/line)
Scottie DX	345.6ms	(1.0800ms/pixel @ 320 pixels/line)

TIMING SEQUENCE

(1) "Starting" sync pulse (first line only!)	9.0ms	1200hz
(2) Separator pulse	1.5ms	1500hz
(3) Green scan		
(4) Separator pulse	1.5ms	1500hz
(5) Blue scan		
(6) Sync pulse	9.0ms	1200hz
(7) Sync porch	1.5ms	1500hz
(8) Red scan		

After the first line, repeat steps 2 – 8 for all following lines.

MARTIN MODES
Martin modes are straightforward, with the sync pulse occurring at line breaks.

VIS CODES
Martin 1 44d (decimal)
Martin 2 40d

COLOR MODE	RGB (1500-2300hz luminance range)
SCAN SEQUENCE	Green, Blue, Red
NUMBER OF LINES	256
NORMAL DISPLAY RES	320x256 (including 16-line header)

TRANSMISSION TIME

Martin 1	114.3 seconds (not including cal. header/VIS)
Martin 2	58.06 seconds

COLOR SCAN TIME

Martin 1	146.432ms	(.4576ms/pixel @ 320 pixels/line)
Martin 2	73.216ms	(.2288ms/pixel @ 320 pixels/line)

TIMING SEQUENCE

(1) Sync pulse	4.862ms	1200hz
(2) Sync porch	0.572ms	1500hz
(3) Green scan		
(4) Separator pulse	0.572ms	1500hz
(5) Blue scan		
(6) Separator pulse	0.572ms	1500hz
(7) Red scan		
(8) Separator pulse	0.572ms	1500hz

Repeat the above sequence for 256 lines.

ROBOT 36 COLOR
Robot 36 is probably the most complex of all SSTV modes:

- It uses Y, R-Y, B-Y color encoding
- The R-Y color information is averaged for two lines, and transmitted on even lines.
- The B-Y color information is averaged for two lines, and transmitted on odd lines.
- The R-Y and B-Y scans have only ½ the period (44ms) of the Y scan. (88ms)
- Even lines use a 1500hz "separator" pulse, while odd lines use 2300hz.

VIS CODE 8d (decimal)

COLOR MODE	Y, R-Y, B-Y (see Appendix B)
SCAN SEQUENCE	Y, R-Y (even lines) Y, B-Y (odd lines)
NUMBER OF LINES	240

TRANSMISSION TIME 36 seconds (not including cal. header/VIS)

Note: For clarity, 2 complete lines are shown below

TIMING SEQUENCE

(1) Sync pulse	9.0ms	1200hz
(2) Sync porch	3.0ms	1500hz
(3) Y scan	88.0ms	
(4) "Even" separator pulse	4.5ms	1500hz
(5) Porch	1.5ms	1900hz
(6) R-Y scan	44ms	
(7) Sync pulse	9.0ms	1200hz
(8) Sync porch	3.0ms	1500hz
(9) Y scan	88.0ms	
(10) "Odd" separator pulse	4.5ms	2300hz
(11) Porch	1.5ms	1900hz
(12) B-Y scan	44ms	

Repeat the sequence above for 240 lines.

ROBOT 72 COLOR
Robot 72 is somewhat simpler than Robot 36, but still suffers from different scan periods for the Y versus the R-Y and B-Y scans.

VIS CODE 12d (decimal)

COLOR MODE	Y, R-Y, B-Y (see Appendix B)
SCAN SEQUENCE	Y, R-Y, B-Y
NUMBER OF LINES	240

TRANSMISSION TIME 72 seconds (not including cal. header/VIS)

TIMING SEQUENCE (one line shown)

(1) Sync pulse	9.0ms	1200hz
(2) Sync porch	3.0ms	1500hz
(3) Y scan	138ms	
(4) Separator pulse	4.5ms	1500hz
(5) Porch	1.5ms	1900hz
(6) R-Y scan	69ms	
(7) Separator pulse	4.5ms	2300hz
(8) Porch	1.5ms	1500hz
(9) B-Y scan	69ms	

Repeat the above sequence for 240 lines.

WRASSE SC2-180
SC2-180 is probably the simplest RGB SSTV mode.

VIS CODE 55d (decimal)

COLOR MODE	RGB (1500-2300hz luminance range)
SCAN SEQUENCE	Red, Green, Blue
NUMBER OF LINES	256

TRANSMISSION TIME 182 seconds (not including cal. header/VIS)

COLOR SCAN TIME 235.000ms (.7344ms/pixel @ 320 pixels/line)

TIMING SEQUENCE (one line shown)

(1) Sync pulse	5.5225ms	1200hz
(2) Porch	0.500ms	1500hz
(3) Red scan		
(4) Green scan		
(5) Blue scan		

Repeat the above sequence for 256 lines.

PASOKON "P" modes

The "P" modes are all designed to display at 640x496, including a 16-line header. They differ somewhat from other "families" of modes, in that the sync pulse and porch periods vary with the sub-mode. When John Langner originally authored them, his intent was that the pixel clock, sync, and porch periods all be evenly divisible into a standard RS232 clock rate.

VIS CODES:

P3 113d (decimal)
P5 114d
P7 115d

COLOR MODE	RGB (1500-2300hz luminance range)
SCAN SEQUENCE	Red, Green, Blue
NUMBER OF LINES	496

TRANSMISSION TIMES

P3	203 seconds (not including cal. header/VIS)
P5	304.6 seconds
P7	406.1 seconds

COLOR SCAN TIMES:

P3	133.333ms (.2083ms/pixel @ 640 pixels/line)
P5	200.000ms (.3125ms/pixel @ 640 pixels/line)
P7	266.666ms (.4167ms/pixel @ 640 pixels/line)

SYNC PERIODS

P3	5.208ms	1200hz
P5	7.813ms	1200hz
P7	10.417ms	1200hz

PORCH PERIODS

P3	1.042ms	1500hz
P5	1.563ms	1500hz
P7	2.083ms	1500hz

TIMING SEQUENCE (one line shown)

(1) Sync pulse
(2) Porch
(3) Red scan
(4) Porch
(5) Green scan
(6) Porch
(7) Blue scan
(8) Porch

Repeat the above sequence for 496 lines.

PD Modes
The PD modes were developed jointly by Don Rotier and Paul Turner. Although they're a little complicated, they represent a fair trade-off between color accuracy and transmission time.

VIS CODES

PD50	93d (decimal)
PD90	99d
PD120	95d
PD160	98d
PD180	96d
PD240	97d
PD290	94d

COLOR MODE	Y, R-Y, B-Y (See Appendix B)
SCAN SEQUENCE	Y, R-Y, B-Y

NORMAL DISPLAY RESOLUTION

PD50	320x256	(all include 16-line header)
PD90	320x256	
PD120	640x496	
PD160	512x400	
PD180	640x496	
PD240	640x496	
PD290	800x616	

TRANSMISSION TIMES

PD50	49.7 seconds
PD90	90.0 seconds
PD120	126.1 seconds
PD160	160.9 seconds
PD180	187.1 seconds
PD240	248.0 seconds
PD290	288.7 seconds

COLOR SCAN TIMES (Y, R-Y, B-Y)

PD50	91.520ms
PD90	170.240ms
PD120	121.600ms
PD160	195.584ms
PD180	183.040ms
PD240	244.480ms
PD290	228.800ms

TIMING SEQUENCE

Note: two complete lines are shown.

(1) Sync pulse	20.000	1200hz
(2) Porch	2.080ms	1500hz
(3) Y scan (from odd line)		
(4) R-Y scan averaged for two lines		
(5) B-Y averaged for two lines		
(6) Y scan (from even line)		

Repeat until correct number of lines are transmitted for sub-mode.

FAX480 mode

FAX480 is a monochrome FAX, rather than a "true" SSTV mode. It uses a 5-second FAX tone "header" followed by a phasing area, instead of a Robot calibration header and VIS code. It was developed by Dr. Ralph Taggart, WB8DQT

We've included it here because it's useful for monochrome image transmission, and is often included with SSTV software. (ChromaPIX included)

COLOR MODE	Monochrome (1500-2300hz luminance range)
NORMAL DISPLAY RES	512x480
NUMBER OF LINES	480
SCAN TIME	.512ms x 512 pixels = 262.144ms

HEADER (start tone)

(1)	2.05ms	2300hz
(2)	2.05ms	1500hz

Repeat this sequence 1220 times, for a total of 5.002 seconds.

PHASING INTERVAL (20 lines)

(1)	Sync	5.12ms	1200hz
(2)	White	.512ms x 512 pixels = 262.144ms	

Repeat this sequence 20 times, for a total of 20 lines.

TIMING SEQUENCE

(1) Header
(2) Phasing interval
(3) Sync 5.12ms 1200hz
(4) Scan line

Repeat 3-4 until 480 lines have been transmitted.

Note: Some systems (ChromaPIX included) use the 20-line phasing area for video.)

Appendix A RGB Color Encoding

Of the two encoding methods used in SSTV, RGB is by far the simplest. The calculations below are intended for use with non-linear 24-bit RGB color, which is standard for modern computer-based systems.

24-bit RGB is straightforward. 8-bit (0-255) Red, Green, and Blue values are "packed" into a single 24-bit identity. Red values occupy the lowest 8 bits, followed by Green, then Blue. This allows for a maximum of 16,777,216 unique combinations of colors, which is generally adequate for any but the highest-end imaging systems.

To extract individual 8-bit R,G,B values from a 24-bit RGB encoded pixel :

RedByte = colorval AND 255
GreenByte = (colorval AND 65280) / 256
BlueByte = (colorval AND 16711680) / 65536

If your programming language and platform supports it, shifting and masking the 24-bit value may be slightly faster than using the "generic" method above, since it would avoid the division.

SSTV systems use the frequency range of 1500-2300hz to represent the range of brightness values from pure black to pure white. Because of this, we must convert our 8-bit R,G,B values into the appropriate frequency:

Frequency = 1500 + (ColorByte * 3.1372549)

Conversely, to derive the 8-bit R,G, or B value from a frequency:

ColorByte = (Frequency – 1500) / 3.1372549

Obviously, look-up tables could be calculated in advance, avoiding the necessity of performing floating-point multiplies and divisions for encoding and decoding every pixel in every scan.

For receiving purposes, the easiest way to recombine the individual R,G, and B values back into a 24-bit RGB identity is to use the RGB(n,n,n) macro provided with most modern compilers. If that isn't possible :

RedVal = ColorByte (directly, no conversion required)
GreenVal = ColorByte * 256
BlueVal = ColorByte * 65536

Once the R, G, and B values have been scaled correctly, they can be simply added together to form the 24-bit identity:

ColorVal = Red + Green + Blue

Appendix B Y, R-Y, B-Y Color Encoding

(Y, R-Y, B-Y) encoding is used in Robot and PD modes, as well as the seldom-used Wrasse SC2-120 mode. This method of encoding is also referred to as (Y/C), (Y'U'V'), (Y'CBCR), and other names. Although all these refer to the same overall method of encoding, the exact scalings used vary somewhat by implementation and developer.

The encoding used in ChromaPIX and W95SSTV is compliant with ITU BT. Rec. 601. (the 1953 NTSC standard) Because there has been no clearly-defined standard, the scalings described below may vary slightly from those used by other developers.

To convert non-linear RGB to [Y, R-Y, B-Y] (scaled 0-255) :

Y = 16.0 + (.003906 * ((65.738 * R) + (129.057 * G) + (25.064 * B)))

RY = 128.0 + (.003906 * ((112.439 * R) + (-94.154 * G) + (-18.285 * B)))

BY = 128.0 + (.003906 * ((-37.945 * R) + (-74.494 * G) + (112.439 * B)))

Again, as with RGB encoding, these values can be converted to frequency, for SSTV transmission :

Frequency = 1500 + (v * 3.1372549)
(where 'v' is the Y, R-Y, or B-Y value)

Conversely, Y, R-Y, B-Y values scaled 0-255 may be converted back to non-linear RGB:

R = 0.003906 * ((298.082 * (Y – 16.0)) + (408.583 * (RY – 128.0)))

G = 0.003906 * ((298.082 * (Y – 16.0)) + (-100.291 * (BY – 128.0)) + (-208.12 * (RY – 128.0)))

B = 0.003906 * ((298.082 * (Y – 16.0)) + (516.411 * (BY – 128.0)))