

2446 Stream Server

Model 2446 | Ver1.0.1 | Oct. 08

SENSORAY | embedded electronics



Designed and manufactured in the U.S.A

SENSORAY | p. 503.684.8005 | email: info@SENSORAY.com | www.SENSORAY.com

7313 SW Tech Center Drive | Portland, OR 97203

TABLE OF CONTENTS

LIMITED WARRANTY.....	3
SPECIAL HANDLING INSTRUCTIONS.....	4
INTRODUCTION.....	5
Feature Summary.....	5
Specifications.....	5
SOFTWARE.....	6
Feature Summary.....	6
SDK Reference.....	7
Release Notes.....	7
General Notes.....	7
Command Description.....	7
Command Reference.....	8
Stream Server Startup.....	16
Stream Server configuration (/etc/s2246/ss0.conf).....	17
Description of parameters in ss0.conf file.....	17
Example 1: Change video input to SVideo Channel IN1.....	20
Example 2: Change video bitrate to 5Mbit/s.....	20
Example 3: Change to MPEG2 Program Stream.....	21
Example 4: Change back to MPEG2 Transport stream.....	22
Appendix A: Distributables.....	22
Appendix B: Perl script to start streaming and update captions continuously.....	23

Limited warranty

Sensoray Company, Incorporated (Sensoray) warrants the hardware to be free from defects in material and workmanship and perform to applicable published Sensoray specifications for two years from the date of shipment to purchaser. Sensoray will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The warranty provided herein does not cover equipment subjected to abuse, misuse, accident, alteration, neglect, or unauthorized repair or installation. Sensoray shall have the right of final determination as to the existence and cause of defect.

As for items repaired or replaced under warranty, the warranty shall continue in effect for the remainder of the original warranty period, or for ninety days following date of shipment by Sensoray of the repaired or replaced part, whichever period is longer.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. Sensoray will pay the shipping costs of returning to the owner parts that are covered by warranty. A restocking charge of 25% of the product purchase price, or \$105, whichever is less, will be charged for returning a product to stock.

Sensoray believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, Sensoray reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult Sensoray if errors are suspected. In no event shall Sensoray be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, SENSORAY MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF SENSORAY SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. SENSORAY WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEROF.

Third party brands, names and trademarks are the property of their respective owners.

Special handling instructions

The circuit board contains CMOS circuitry that is sensitive to Electrostatic Discharge (ESD).

Special care should be taken in handling, transporting, and installing circuit board to prevent ESD damage to the board. In particular:

- Do not remove the circuit board from its protective anti-static bag until you are ready to install the board into the enclosure.
 - Handle the circuit board only at grounded, ESD protected stations.
 - Remove power from the equipment before installing or removing the circuit board.
-

Introduction

Model 2446 is a USB video and audio capture device. It allows advanced annotation and MPEG streaming.

Feature Summary

- MPEG1/2/4 capture at full frame rate and full resolution.
- 2 Composite inputs, 2 S-video inputs, SDI/HDSDI input.
- 1 DVI output, 1 Composite output, 1 S-Video output.
- Annotation/Caption/On screen display (OSD) – multiple text overlays with up to 8 windows
- Compact and robust design.
- Powered through the USB port.

Specifications

Inputs	Video: composite (BNC), S-video (DIN), 75 Ohm Audio: line in stereo (2xRCA), microphone mono (3.5 mm 3-position jack), 10 kOhm.
Electret microphone bias current, max	3 mA
Input video formats	NTSC (M), PAL (BDGHIMN)
Output formats	MPEG1, MPEG2 (MP@ML), MPEG4 (SP@L3 with B-frames), Motion JPEG
Output resolutions	320x240, 720x480 (NTSC, 30 frames/sec) 320x288, 720x576 (PAL, 25 frames/sec)
OSD	max 96 characters per frame, 16x16 font
Snapshot	JPG and BMP formats, concurrent with capture/preview
Power	+5 V, 470 mA (through USB port)

Software

Feature Summary

The 2446 stream server encodes MPEG1, MPEG2(MP@ML) or MPEG4(SP@L3) at standard resolution. The encoded stream may be saved to file on the server or streamed over an IP Ethernet network. The network streaming supports RTSP and UDP streaming. For UDP streaming, the multiplexed formats are MPEG Program Stream(MPEG1, MPEG2) or MPEG Transport stream(MPEG4).

SDK Reference

Release Notes

V.1.0.1:

Initial release.

General Notes

A client application "scmd" is used to communicate with the stream server. The command syntax is similar to the 2416 Sensoray stream server product. Additionally, the user may communicate with the stream server using regular TCP sockets.

An example command scenario is below. In this example, the 2446 is at the IP address 10.135.1.199, port 17740(default port). The client wants the server to stream UDP MPEG4 (TS mux) to the address of 10.10.10.21, port 1234. A caption of "Sensoray Inc" is desired at location (x, y) = (100, 150). The client uses the scmd program to automatically make the correct socket calls. It is assumed the stream server has been configured for MPEG4, TS mux format prior to the calls below.

1) Start stream:

```
"scmd 10.135.1.199 17740 [STREAMUDP^10.10.10.21^1234^]"
```

2) Caption command(extended version):

```
"scmd 10.135.1.199 17740 [CAPTEXT2^0^100^150^"Sensoray Inc"^arial^14^green^1^255^]"
```

Command Description

The stream server operates by default on port 17740 (see advanced configuration section to change this port). It communicates using TCP sockets. A socket can be opened to the stream server and commands sent. The command timeout is 5 seconds. If multiple commands need to be sent with more than 5 seconds in between, the client should send a PING command(see below) to keep the connection alive.

Every command is formatted as follows:

Bytes 0-3: packet id

Bytes 4-7: command length

Bytes 8-X: command

The "packet id" is sent back in the response. The command length is the length of the command string after the first 8 bytes of header. Every command is delimited by '[' and ']' characters.

The "scmd" client (included in the SDK), sets up a separate TCP connection for each command. Users who don't want to use the TCP interface can then use the command line to issue commands.

The usage of scmd is as follows:

"scmd IP_ADDRESS PORT COMMAND" where IP_ADDRESS is the address of the server, PORT is the port, and COMMAND is the full text string of the command.

The response message is as follows:

Bytes 0-3: packet id

Bytes 4-7: response length

Bytes 8-X: response string

If the command was accepted, an "OK" is sent; otherwise an error message is sent.

Example Perl scripts are shown in the Appendix section.

Command Reference

[STREAMUDP^ip_address^port^]

Streams MPEG compressed video over UDP to ip_address. In general, UDP stream is used for pre-multiplexed streams on the server. Supported multiplexed formats for MPEG streaming are TS(Transport stream) and MPEGPS(program stream). For MPEG4, only TS is supported.

Parameters

ip_address

The ip_address to stream. Can be unicast or multicast. Multicast occurs when the address is in the range 224.0.0.0-239.255.255.255.

port

The UDP port. Port 1234 is the default port.

Returns

OK response on success

[STOP^]

Stops the stream server.

Parameters

None.

Returns

OK response on success. Error message otherwise.

[PING^]

Checks if connection still alive and also keeps connection alive. Command server connection has 5 second timeout.

Parameters

None.

Returns

ACK response on success

[CAPTEXT^index^x^y^string^font^font_size^color^trans^region^]

Creates an annotation on the video image. CAPTEXT provides full control of all text annotation parameters. CAPSHORT below provides more limited control with a shorter command. All caption commands update the caption windows in memory. To upload the caption to the hardware, use the "CAPUPDATE" command. This is done so that multiple changes can be done before doing an CAPUPDATE because an CAPUPDATE to the hardware may take up to one video frame due to the size of the data. See CAPUPDATE command below.

index

Index of Caption/Annotation/OSD window.

x

x position of caption.

y

y position of caption.

string

value of caption. If spaces are in text, use double quotes. Eg. "Test string"

font

font type of the text. ARIAL, TIMES, COUR for arial, times new roman and courier, or provide the full path to a custom TTF font file on the 2446 server.

font_size

Size of the font.

color

Color of the text: black, silver, gray, navy, white, maroon, red, purple, fuchsia, green, lime, olive, yellow, navy, blue, teal, aqua.

trans

transparency of the text. 0-none, 1-transparent. If using no transparency, see "BACKCOL" command for background window color

region

region mask of the text. 1-Monitor(DVI, Analog Outputs), 2-Compression,4-Snapshots. Use region=7 for annotation on all regions.

Returns

OK response on success

[CAPSHORT^index^x^y^string^region^]

Creates an annotation on the video image with arial font, 16 point, transparent background and black text.

Parameters

index

Index of Caption/Annotation/OSD window.

x

x position of caption.

y

y position of caption.

string

value of caption. If spaces are in text, use double quotes. Eg. "Test string"

region

region mask of the text. 1-Monitor(DVI, Analog Outputs), 2-Compression,4-Snapshots. Use region=7 for annotation on all regions.

Returns

OK response on success

[CAPIMAGE^index^x^y^image_path^region^]

Adds a 24bit BMP image as a caption overlay.

Parameters

index

Index of Caption/Annotation/OSD window.

x

x position of caption.

y

y position of caption.

image_path

full path to the caption image. Image MUST be in 24bit BMP or PCX file format.

region

region mask of the text. 1-Monitor(DVI, Analog Outputs), 2-Compression,4-Snapshots. Use region=7 for annotation on all regions.

Returns

OK response on success

[BACKCOL^color^]

Sets background color for text annotations without transparency.

Parameters

color

Background color of the text: black, silver, gray, navy, white, maroon, red, purple, fuchsia, green, lime, olive, yellow, navy, blue, teal, aqua.

Returns

OK response on success

[CAPCLR^]

Clears all captions.

Parameters

None.

Returns

OK response on success

[CAPMOVE^{index}^x^y]

Moves caption at specified index to new position.

Parameters

index

Index of Caption/Annotation/OSD window.

x

new x position of caption.

y

new y position of caption.

Returns

OK response on success

[CAPUPDATE[^]]

Updates the annotations. Will update the hardware with the new overlay information from the Caption commands. CAPUPDATE should be done at most once every 33 milliseconds or once per video frame. More frequent updates may decrease the overlay performance.

Parameters

None

Returns

OK response on success

[STREAMRTP^{ip_address}^{port}]

Streams RTSP to *ip_address* and *port* and set stream server to RTP mode.

Parameters

ip_address

IP address. If no immediate streaming is desired, set *ip_address* to 0.0.0.0. This puts the server into unicast RTSP mode. The client can open the stream with `rtsp://server_ip_address:8554`. To stream to multicast, use a multicast address of 225.0.0.1 to 239.255.255.255.

port

multicast port

Returns

OK response on success

[AINPUT]

Change the audio input on the fly. Use SET command and AUDIO_INPUT parameter to change audio input on startup.

Parameters

input

Audio input. Use 0 for line, 1 for microphone.

Returns

OK response on success

[SET^{name}value]

Sets one of the stream server parameters. See Stream Server Configuration section.

Parameters

name

Name of parameter to change. Eg. VIDEO_INPUT

value

value of parameter

Returns

OK response on success

[GET^{name}]

Retrieves value of parameter.

Parameters

name

Name of parameter to retrieve. Eg. VIDEO_INPUT

Returns

OK response on success

[SAVE]

Saves all the configuration parameters changed with SET commands to the working config file. Use STORE to update the permanent file loaded when the system reboots.

Parameters

None

Returns

OK response on success

[RESTART^]

Restarts the stream server. After parameters changed with SAVE and SET command(s), RESTART will load the new parameters. Some parameters require a restart. Other parameters can be changed on the fly.

Parameters

None

Returns

OK response on success

[STORE^]

Copies the working /var/s2246/ss0.conf onto the read-only /etc/s2246/ss0.conf, so that the next time the system boots it will have the new settings.

Parameters

file_path

Full file path on the server to record to.

Returns

OK response on success.

[SNAPSHOT^filepath^type^]

Takes a snapshot. Uses merge method(for interlaced fields) specified in config file as parameter SNAPSHOT_MERGE. Default merge method is 2(MERGE 2 files).

Parameters

filepath

Full filepath for the snapshot. Snapshot will be saved locally on the stream server. It is up to the caller to make sure there is enough space. Snapshot extension will be appended to

the filepath. Eg. Filepath=/mnt/cflash/test_snap and type =1, saved file will be /mnt/cflash/test_snap.jpg.

type

Type mask of the snapshot(1-JPEG, 2-BMP, 4-PPM)

Returns

OK response on success

[CLKCAP^x^y^font_size^color^trans^region^on_off^]

Controls optional caption clock thread(text clock string). Caption clock thread will automatically place a clock caption in overlay index 0 and update the time every second.

Parameters

x

x position of clock

y

y position of clock.

color

Color of the text: black, silver, gray, navy, white, maroon, red, purple, fuchsia, green, lime, olive, yellow, navy, blue, teal, aqua.

trans

transparency of the text. 0-none, 1-transparent. If using no transparency, see "BACKCOL" command for background window color

region

region mask of the text. 1-Monitor(DVI, Analog Outputs), 2-Compression,4-Snapshots. Use region=7 for annotation on all regions.

on_off

on_off setting of the clock(1-on, 0-off).

Returns

OK response on success.

[ENCODE^FILE^file_path^]

Encode file writes a file using the current COMPRESSION and MUX settings. It will continuously write to the local server disk. The user is responsible for stopping the stream before the disk space runs out.

Parameters

file_path

Full file path on the server to record to.

Returns

OK response on success.

```
[ENCODE^UDPFILE^ip_address^port%file_path^]
```

Encode UDPFILE writes a file using the current COMPRESSION and MUX settings while streaming UDP to the IP address specified. It will continuously write to the local server disk. The user is responsible for stopping the stream before the disk space runs out.

Parameters

ip_address

ip address for UDP streaming.

port

ip port.

file_path

Full file path on the server for recording.

Returns

OK response on success.

Stream Server Startup

By default, the stream server loads when the Linux OS boots up. /usr/etc/services contains the driver loading commands followed by the stream server startup. It should not be required to change this.

Additions to rc.local:

```
modprobe s2246s_ez
modprobe s2246w_ez
modprobe s2246s
modprobe s2246w
mkdir -p /var/s2246
cp /etc/s2246/ss0.conf /var/s2246/ss0.conf
sleep 8
/usr/sbin/scserver -cd /var/s2246/ss0.conf
```

If required, the stream server can be stopped from the command line using `/usr/sbin/scserver -k`. It can be restarted with `/usr/sbin/scserver`. The 2446 as shipped should not require starting and stopping the stream server. It only needs restarted in this manner if for instance the PORT 17740 of the server needs changed to a different value. In this case, one can edit the `ss0.conf` file and change `SERVER_PORT` to the desired port and then can the above commands to stop and then restart the stream server.

(Note: `scserver` itself will run `/usr/sbin/str-server` to allow streaming and file recording of the MPEG stream.).

Stream Server configuration (/etc/s2246/ss0.conf)

The file `/etc/s2246/ss0.conf` contains the parameters for the stream server. For protection, the root file system is read only so the config file is copied to `/var/s2246/ss0.conf` and run from there. In general, this file should not be edited by hand. The commands `SET`, and `SAVE`, and `RESTART` should be used to change them. The command `STORE` will copy the working `/var/s2246/ss0.conf` file to `/etc/s2246/ss0.conf` for permanent storage.

Each line in the `ss0.conf` config file specifies a parameter. The values should always be enclosed in quotes. The "=" sign should appear after each parameter name. EG.

```
PARAMETER_NAME="PARAMETER_VALUE"
```

A backup of the original configuration file is saved as `/etc/s2246/ss0.orig`. If your configuration becomes corrupt, this file may be copied to `ss0.conf` to restore the default settings.

Description of parameters in ss0.conf file

SERVER_NAME

Name of the stream server

DEBUG_LEVEL

Debug level of the stream server. Logs to `/var/log/ss17740.log` and `/var/log/ms17740.log`.

"0" Quiet, "1" error, "2" info, "3" verbose, higher values reserved for development only.

MPEG_DEVICE_NAME

`/dev` device name of the compression chip driver (default `"/dev/s2246w_0"`)

FPGA_DEVICE_NAME

`/dev` device name of FPGA device (default `"/dev/s2246s_0"`)

INTERFACE_ADDRESS

Local server interface IP address. Set to "0.0.0.0" for default.

SERVER_PORT

Server port for commands (def 17740)

SERVER_TYPE

Stream type for autostart. EUDP or ERTTP
Use ERTTP if using the RTSP model for streaming
Use EUDP if using UDP for streaming

AUTOSTART

autostart the server after startup(immediate streaming) YES or NO

SSS_VERSION

Sensaray stream version number

DESTINATION_ADDRESS

Stream destination address for autostart mode.
(224.0.0.0-239.255.255.255 for multicast UDP)
(225.0.0.0-239.255.255.255 for multicast RTP)

DESTINATION_PORT

Stream destination port for autostart mode. "17760"

TTL

time to live for outgoing ip packets.

MPEG_STREAM

stream type. Currently only "MUX" (multiplexed) type is supported. VES streamed if using RTP.

MUX

Sets the multiplex format for file saving and UDP streaming. Uses the ffmpeg open source library to multiplex the compressed stream. Mpeg program stream-"MPEG". Mpeg transport stream (default) - "TS". Other mux types may be supported at a future date. Not used for RTSP streaming.

COMPRESSION

"0"-MPEG4, "1"-MPEG, "2"-MPEG2, "3"-MJPEG(future), "4"-H263(future)

Note: not all combinations of MUX and COMPRESSION are supported.

Current tested and supported system formats

MPEG2 with MUX=TS (recommended default setting for VLC UDP streaming)

MPEG4 with MUX=TS

MPEG1 with MUX=MPEG

MPEG2 with MUX=MPEG

BIT_RATE

MPEG bit rate in bits/second. If non-zero, overrides default settings(default 0).

GOP_SIZE

Override GOP size defaults. Use 0 for default settings.

GOP_MODE

Override GOP mode default settings. "0"-Ionly, "1"-IP, "2"-IPB, "3"-default.

VIDEO_SYSTEM

Video system ("PAL" or "NTSC")

VIDEO_INPUT

"0"-Composite channel IN0, "1"- Composite channel IN1, "2"- SVideo channel IN0, "3"-
Svideo channel IN1, "5"-SDI, "6" HDSDI

AUDIO_INPUT

Audio input channel: "0" = line input L/R, "1"- microphone input.

AUDIO_BIT_RATE

Audio bit-rate (kbps): default 192.

CAPTION_CLOCK

caption clock on "1" off "0"

CAPTION_CLOCK_X

x position of clock

CAPTION_CLOCK_Y

y position of clock

CAPTION_CLOCK_SCALE

Clock text font-size ("12"- "49")

CAPTION_CLOCK_TRANS

Clock text transparency attribute: "1"-transparent(default), "0"-not transparent

CAPTION_CLOCK_COLOR

Color of clock/time text: "green", "silver", "gray", "white", "maroon", "red", "purple", "fuchsia"
"lime", "olive", "navy", "blue", "teal", or "aqua".

CAPTION_CLOCK_REGION

Clock region mask: "1"-Monitor, "2"-MPEG, "4"-still. Use "7" for all regions.

CAPTION_BACKCOLOR

Default caption background window color:

"green", "silver", "gray", "white", "maroon", "red", "purple", "fuchsia"
"lime", "olive", "navy", "blue", "teal", or "aqua".

SNAPSHOT_MERGE

"0"-none, "1"-duplicate field, "2"-merge both fields(default), "3"-interpolate one field.

Example 1: Change video input to SVideo Channel IN1

Changing the video input is done by changing the VIDEO_INPUT parameter in the configuration file.

To change to SVideo IN1, perform the following.

1) If stream server running, send stop command:

[STOP^]

2) Set the VIDEO_INPUT parameter to 3.

[SET^VIDEO_INPUT^3^]

3) Save the parameter file

[SAVE^]

4) Restart the stream server.

[RESTART^]

Example 2: Change video bitrate to 5Mbit/s

1) If stream server running, send stop command:

[STOP^]

2) Set the BIT_RATE parameter to 5000000.

[SET^BIT_RATE^5000000^]

3) Save the parameter file

[SAVE^]

4) Restart the stream server.

[RESTART^]

Example 3: Change to MPEG2 Program Stream

1) If stream server running, send stop command:

[STOP^]

2) Set the MUX parameter to "MPEG".

[SET^MUX^MPEG^]

3) Set the COMPRESSION parameter to 2 for MPEG-2.

[SET^COMPRESSION^2^]

4) Save the parameter file

[SAVE^]

5) Restart the stream server.

[RESTART^]

6) Send a start streaming command if desired.

[STREAMUDP^10.10.10.21^1234^]

Example 4: Change back to MPEG2 Transport stream.

1) If stream server running, send stop command:

```
[STOP^]
```

2) Set the MUX parameter to "TS".

```
[SET^MUX^TS^]
```

3) Set the COMPRESSION parameter to 2 for MPEG-2.

```
[SET^COMPRESSION^2^]
```

4) Save the parameter file

```
[SAVE^]
```

5) Restart the stream server.

```
[RESTART^]
```

6) Send a start streaming command if desired.

```
[STREAMUDP^10.10.10.21^1234^]
```

Appendix A: Distributables

- /usr/sbin/scserver (stream control and caption/config server)
- /usr/sbin/str-server (stream server)
- /usr/etc/s2246/images/*.bmp (bitmaps)
- /usr/etc/s2246/default_setting_live.txt (default MPEG Compression settings. Do not edit.)
- /usr/etc/s2246/ss0.conf (saved stream server configuration file. Do not edit directly.)
- /usr/etc/s2246/ss0.orig (original stream server configuration. Do not edit)
- /usr/etc/s2246/ss0.desc (description of ss0.conf)

- /lib/firmware/dec_XX.ini (video decoder init files. Do not edit.)
- /lib/firmware/enc_XX.ini (video encoder init files. Do not edit.)
- /lib/firmware/fix_setting.txt (advanced chip config. Do not edit)
- /lib/firmware/go7007fw.bin (chip firmware file. Chip firmware may be loaded directly from driver source in newer revisions)
- /lib/firmware/go7007sb.bin (chip firmware file.)
- /lib/firmware/s2246.bin (Firmware file for FPGA hardware)
- /lib/firmware/tlv320.ini (audio chip initialization. Do not edit)
- /lib/modules/.../kernel/drivers/media/video/s2246s.ko (FPGA driver)
- /lib/modules/.../kernel/drivers/media/video/s2246w.ko. (MPEG driver)
- /lib/modules/.../kernel/drivers/media/video/s2246s_ez.ko. (FPGA USB loader)
- /lib/modules/.../kernel/drivers/media/video/s2246w_ez.ko. (MPEG USB loader)

Appendix B: Perl script to start streaming and update captions continuously.

```
#!/usr/bin/perl
# ovltest2_tcp.pl
# @Copyright: Sensoray Company Inc. 2007(C)
# Script to start a UPDSTREAM, and update captions remote
# with the 2246 streaming server.
# Uses TCP sockets directly(not scmd).
# Opens the socket and sends the commands.
#
# 1) sends start stream command
# 2) Sends caption commands
# 3) Moves captions around
# 4) stops stream when key pressed
# Example usage.
# 2446 is at IP 10.135.1.199(default port 17740)
# We want this 2446 to stream to 10.10.10.21, port 1234
# User would issue this command
# "./ovltest2_tcp.pl 10.135.1.199 17740 10.10.10.21 1234"
```

```
use Shell;
use Term::ReadKey;
use IO::Socket;
```

```

$pkt_id = 0x00000000;
$numArgs = $#ARGV +1;

if($numArgs < 4) {
    print "Usage: ovltest2_tcp.pl serverip serverport destinationIP port\n\n";
    exit 0;
}
$scs_server_ip = $ARGV[0]; #/"127.0.0.1"; #scs_server ip address
$scs_server_port = $ARGV[1];# /"17740"; #scs_server port
$dest_ip = $ARGV[2];
$dest_port = $ARGV[3];

#create the socket
my $sock = new IO::Socket::INET ( PeerAddr => $scs_server_ip,
                                PeerPort => $scs_server_port,
                                Proto => 'tcp');

sub send_cmd {
    my $cmd = shift; # name doesn't matter
    my $tmp1;
    print $sock $cmd;
    $tmp1 = "";
    $sock->recv($tmp1, 128);

    return;
}

sub form_cmd {
    my $cmd = shift;
    $len = length($cmd);
    $pkt_id++;
    $scmd = pack("I", $pkt_id) . pack("I", $len);
    $cmd = $scmd . $cmd;
    return $cmd;
}

die "could not create socket" unless $sock;

#form and send the Stream command
$start_cmd = "[STREAMUDP^" . $dest_ip . "^" . $dest_port . "^]";
$start_cmd = form_cmd $start_cmd;
send_cmd $start_cmd;

#initialize all the positions
$CurX1 = 20;
$CurY1 = 5;
$CurX2 = 302;
$CurY2 = 2;
$start = $CurX1;
$end = $start+200;
$Inc = 1;
$Inc2 = 2;
$CurX4 = 400;
$CurY4 = 100;
$IncX4 = 3;
$IncY4 = -5;

$CurX5 = 600;

```

```

$CurY5 = 400;
$IncX5 = -1;
$IncY5 = 2;
$LastX5, LastY5;

$CurX7 = 350;
$CurY7 = 5;
$IncX7 = 2;
$IncY7 = -1;

$file1 = "/etc/s2246/images/CrossHairSml.bmp";
$file2 = "/etc/s2246/images/logo.bmp";
$file3 = "/etc/s2246/images/X.bmp";
$file4 = "/etc/s2246/images/O.bmp";
$file5 = "/etc/s2246/images/T.bmp";

$cmd1="[CAPIMAGE^1^" . $CurX1 . "^" . $CurY1 . "^" . $file1 . "^255^]";
$cmd1 = form_cmd $cmd1;

$cmd2="[CAPIMAGE^2^" . $CurX2 . "^" . $CurY2 . "^" . $file2 . "^255^]";
$cmd2 = form_cmd $cmd2;

$cmd3="[CAPIMAGE^3^" . $CurX3 . "^" . $CurY3 . "^" . $file3 . "^255^]";
$cmd3 = form_cmd $cmd3;

$cmd4="[CAPIMAGE^4^" . $CurX4 . "^" . $CurY4 . "^" . $file4 . "^255^]";
$cmd4 = form_cmd $cmd4;

$cmd5="[CAPIMAGE^5^" . $CurX5 . "^" . $CurY5 . "^" . $file5 . "^255^]";
$cmd5 = form_cmd $cmd5;

$text6 = $CurX5 . "," . $CurY5;
$cmd6="[CAPTEXT^6^" . $CurX5 . "^" . ($CurY5+10) . "^" . $text6 . "^arial^14^green^1^255^]";

$cmd6 = form_cmd $cmd6;

#send all the initial captions/annotations
send_cmd $cmd1;
send_cmd $cmd2;
send_cmd $cmd3;
send_cmd $cmd4;
send_cmd $cmd5;
send_cmd $cmd6;
#send the update command( see comments below on Update)
$cmd_update = "[CAPUPDATE^]";
$cmd_update = form_cmd $cmd_update;

print "press any key to stop\n";
ReadMode('cbreak');

#move the overlays in an endless loop(until key pressed)
while(1) {

    if( defined ( $char = ReadKey(-1))) {
        #stop the stream server
        $stop_cmd = "[STOP^]";
        $stop_cmd = form_cmd $stop_cmd;
    }
}

```

```

        send_cmd $stop_cmd;
        close($sock);
        ReadMode('normal');
        system("clear");
        print "done \n";
        exit;
    }
    $CurX1+=$Inc;
    $CurY1+=$Inc;
    if(($CurX1<20) || ($CurX1 > 220)) {
        $Inc *= -1;
    }
    $CurX5+=$IncX5;
    $CurY5+=$IncY5;
    if(($CurX5<200) || ($CurX5 > 630)) {
        $IncX5 *= -1;
    }
    if(($CurY5<20) || ($CurY5 > 440)) {
        $IncY5 *= -1;
    }
    }

    $CurY2+=$Inc2;
    if(($CurY2<2) || ($CurY2 > 400)) {
        $Inc2 *= -1;
    }
    }
    $cmd1 = "[CAPMOVE^1^" . $CurX1 . "^" . $CurY1 . "^]";
    $cmd1 = form_cmd $cmd1;

    $cmd2 = "[CAPMOVE^2^" . $CurX2 . "^" . $CurY2 . "^]";
    $cmd2 = form_cmd $cmd2;

    $cmd5 = "[CAPMOVE^5^" . $CurX5 . "^" . $CurY5 . "^]";
    $cmd5 = form_cmd $cmd5;

    #form the update command. This loads the new
    #Annotation into the hardware
    #If moving a bunch of things at once, it is best to
    #Use all the CAPMOVE commands first, THEN call the update.
    $cmd_update = "[CAPUPDATE^]";
    $cmd_update = form_cmd $cmd_update;

    send_cmd $cmd1;
    send_cmd $cmd2;
    send_cmd $cmd5;

    #only update once per frame (30fps NTSC)
    #Updating sooner could result in slower response
    #Because the hardware takes some time to load all the overlay
    #data via USB
    #NOTE: if sleeping longer than 5 seconds, make sure to send a PING
    #command [PING^] or the socket will close
    usleep(30000);
    send_cmd $cmd_update;
}

```