

# Video Integration

## Introduction

This section will walk through the necessary steps to acquire a video stream from the Logitech C920 webcam and the Seek Thermal Camera over the Wi-Fi connection.

\*In order for these to work, the devices must be plugged into the Beaglebone via USB and properly recognized. This can be verified by checking the output of the command: `# lsusb`

## Logitech Stream Setup



*Image 1: Logitech C920*

For the implementation of a video stream, we will be using a service known as MJPG-Streamer. This service will allow us to send the feed of the camera's frames to an IP address for anyone with access to the Wi-Fi network to see.

First, you'll need to download the MJPG-Streamer package along with its dependencies:

```
# pacman -S mjpg-streamer libjpeg imagemagick gcc libv4l-dev
```

To get all the necessary mjpg-streamer files, download the folder "mjpg-streamer-code-182.zip" from our <Downloads> page.

Using WinSCP, move the zip file you just downloaded into a folder of your creation that you will navigate to whenever you wish to start the streaming service. Next, using 7zip, extract the contents of the folder with the following commands:

```
# pacman -S p7zip
```

```
# 7za x mjpg-streamer-code-182.zip
```

If the file "videodev2.h" is not under /usr/include/linux/, run the following command:

```
# sudo ln -s /usr/include/linux/videodev2.h /usr/include/linux/videodev.h
```

Now that everything is installed, you can start the service simply by executing a command. The following command is what our team used for a 60fps 640x280 resolution stream over the IP of alarm. This command (as is) must be executed inside the directory named "mjpg-streamer" from the .zip file contents:

```
# ./mjpg_streamer -i \"/input_uvc.so -r 640x280 -f 60" -o \"/output_http.so -w \"/www" &
```

The ampersand (&) tacked onto the end simply lets the process run unmonitored in the background. To be able to type commands after this one, hit ctrl+C. The stream is available in several different browser modes/windows for you to choose from at <http://alarm:8080> when mjpg-streamer is running.

To stop the stream, the system must be powered off if the ampersand was used.

Other helpful documentation for mjpg-streamer: <http://skillfulness.blogspot.com/2010/03/mjpg-streamer-documentation.html>

## Seek Thermal Stream Setup



*Image 2: Seek Thermal Camera*

Get required packages:

```
# pacman -S python2 python2-pillow opencv python2-numpy python2-pyusb
```

Download the python file “processVideo.py” from our <Downloads> page and place into the directory of your choice.

Running this program from here will NOT work correctly because there is no configured display port for the program to show frames. Running the program using:

```
# python2 processVideo.py
```

Should return something like “Gtk Warning: cannot open display”. To fix this, the program needs to have a remote display to project the frames to, i.e. a server. If this is not the error you receive, it is most likely unable to compile due to a missing package. Make sure they line up with all the imported modules at the top of the python program.

In our setup, we use X11 forwarding to send the compressed frames over to the server for decompression, processing, and display to the user. This is completed by using an XWin Server as part of Cygwin/X.

To get XWin Server on your computer, follow the instructions at this link:

<http://x.cygwin.com/docs/ug/setup.html#setup-cygwin-x-installing>

To open XWin Server, navigate to the root directory (folder = cygwin). Under this folder, go into “bin” and look for “run.exe”. If this doesn’t work, try typing “xwin” into your start menu search bar and see if it comes up there. Just a search for “xwin” on the correct drive should return the correct result.

Once you open up XWin Server, ssh into the Beaglebone:

```
$ ssh -XY root@alarm
```

The `-X` enables X11 forwarding on the server's side. The `-Y` forces authentication credentials. On the Beaglebone, X11 forwarding must be enabled in a different way, although the change will last until you change it back. To do this, using WinSCP, navigate to `/etc/ssh` and open the "`sshd_config`" file in a text editor (Notepad++ is highly recommended). Once opened, scroll down until you see the lines below (line 102):

```
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost no
```

Change these lines to:

```
X11Forwarding yes
#X11DisplayOffset 10
X11UseLocalhost no
```

Additional packages are required for X11 forwarding:

```
# pacman -S xorg xauth
```

The "`xauth`" package is absolutely necessary for this method of streaming, otherwise the server you're using will see the client (Beaglebone) as restricted and will be unable to capture frames from the camera.

Now you finally have everything you need to correctly run the program. This will open up a display on the server's machine (your laptop, in this case) showing a video feed from the thermal camera. To run the program, as before, navigate to the directory where `processVideo.py` is stored and type:

```
# python2 processVideo.py
```

Typing `ctrl+C` will close the program.

From boot, once in the correct directory, the sequence for viewing the stream will be:

1) Start XWin Server

2) `$ ssh -XY root@alarm`

3) `# python2 processVideo.py`

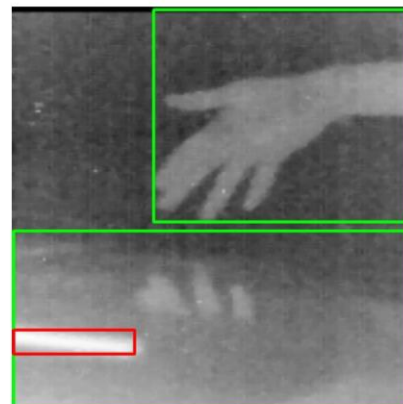


Image 3: Seek Thermal Visual Output with Bounding