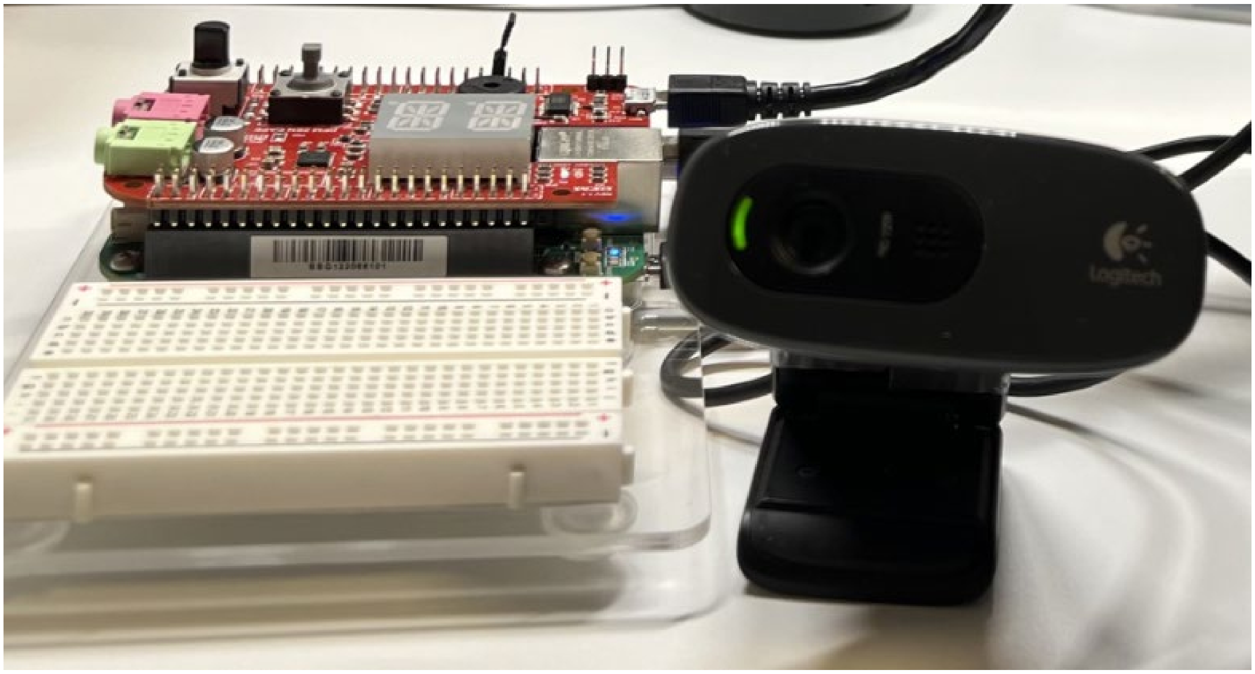
CMPT 433

Using Webcam to Send Video Stream through UDP



Group Member:

Patrick Zhao,

Hao Chen,

Sen Wang

April 14, 2024

This part is based on a guide post on Dr. Brian's website written by a student in the previous semester, "How-to Detect Motion using USB Webcam and OpenCV" and "How To Stream Video Through Udp Using A Webcam On BeagleBone Green With Audio" .

This document guides the user through:

1. Connect the Webcam to BeagleBone through USB port.
2. Install the OpenCV library.
3. Modified the provides codes to adapt target devices.
4. Establish UDP connection and playing the video frame through VLC media player.

Table of Contents

Formatting:

1. Host(computer) commands starting with (host)$ are Linux console commands:

```
(host)$ echo "Hello world"
```

2. Target (BeagleBone) commands start with (bbg)$:

```
(bbg)$ echo "On embedded board"
```

3. Almost all commands are case sensitive.

1. Initial Setup

1. Plug the USB camera into the USB port of target (BeagleBone Green).

2. After plugging in the USB camera, use Linux command 'lsusb' on target:

```
(target) $ lsusb
Bus 001 Device 002: ID 046d:0825 Logitech, Inc. Webcam C270
```

3. Assume the target has Internet access (Networking guide), install Open CV[1] on the target by the following command:

   - (target) $ **sudo apt-get install libv4l-dev**

     ```
     ...
     0 upgraded, 4 newly installed, 0 to remove and 22 not upgraded.
     Need to get 401 kB of archives.
     After this operation, 848 kB of additional disk space will be used.
     ...
     ```

   - (target) $ **sudo apt-get install libopencv-dev**

     ```
     ...
     0 upgraded, 206 newly installed, 0 to remove and 22 not upgraded.
     Need to get 118 MB of archives.
     After this operation, 385 MB of additional disk space will be used.
     ...
     ```

   - (target) $ **sudo apt-get install ffmpeg**

     ```
     ...
     0 upgraded, 36 newly installed, 0 to remove and 22 not upgraded.
     Need to get 44.4 MB of archives.
     After this operation, 73.5 MB of additional disk space will be used.
     ...
     ```

   Note: From the previous guide, we also need to run the command 'sudo apt-get install libav-tools'. However, package 'libav-tools' is not available but is referred to by another package. This mean that the package has been obsoleted and replaced by the package 'ffmpeg'.

[1] OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

2. Codes Modified

1. Download the code boneCV from GitHub and found the code ‘**capture.c’**.

2. Change code on ‘capture.c’:

```
    if (force_format==2){
        fmt.fmt.pix.width    = 1920;
        fmt.fmt.pix.height   = 1080;
        fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_H264;
        fmt.fmt.pix.field     = V4L2_FIELD_INTER_ACED;
    }
```

to:

```
    if (force_format==2){
        fmt.fmt.pix.width    = 640;
        fmt.fmt.pix.height   = 360;
        fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_MJPEG;
        fmt.fmt.pix.field     = V4L2_FIELD_NONE;
    }
```

You can check the resolution support by the camera using the following command:

```
(target) $ v4l2-ctl --list-formats-ext -d /dev/video0
```

We will change the resolution to 640 x 360; other resolutions supported by the camera can also be used. We use low resolution, MJPEG encoding, and ‘V4L2_FIELD_NONE’ to reduce computational demands and bandwidth usage and optimize the performance.

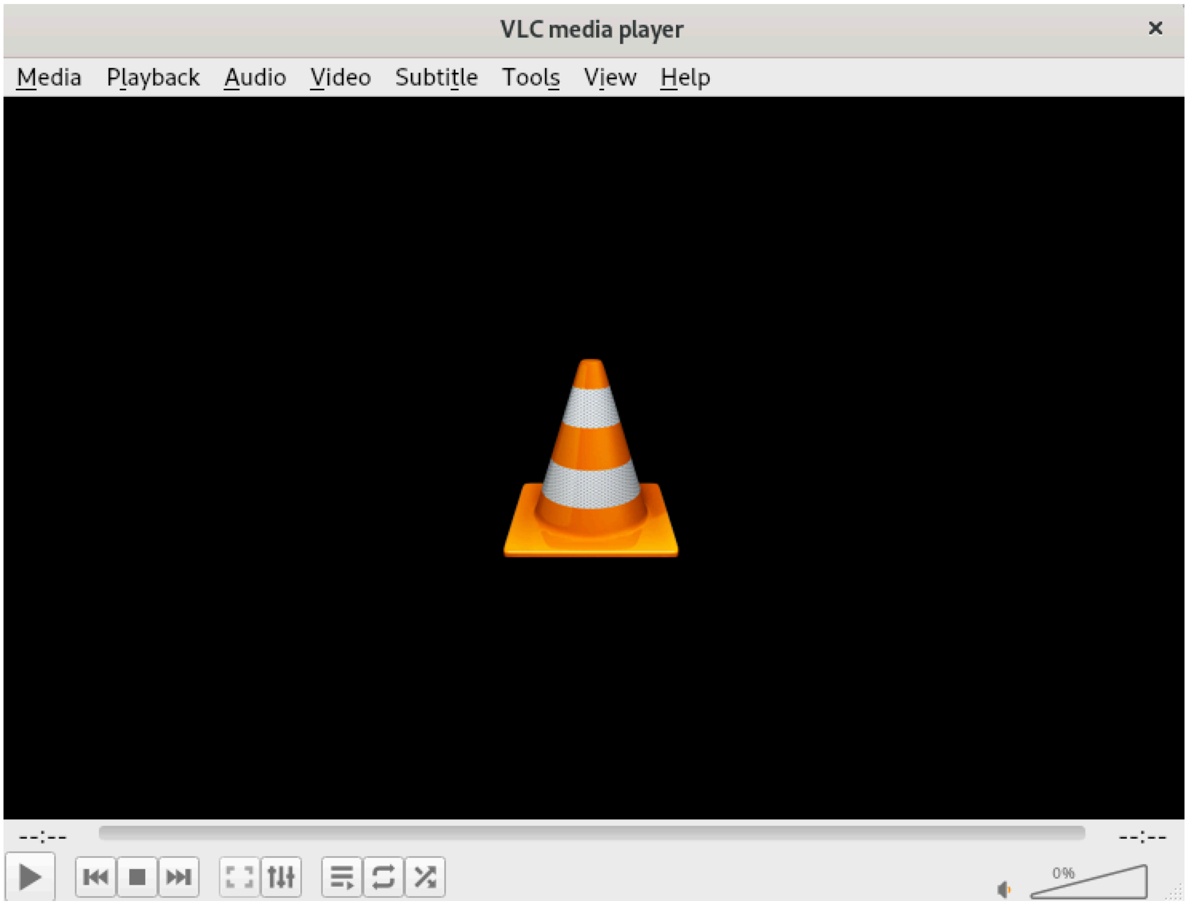
3. Compile the code through Makefile on the target; change the DEST_DIR to your cmpt433 public folder, and after compile, you can enter ‘install’ to move the compiled file to the destination folder (cmpt433 public folder).

```
CROSS_TOOL = arm-linux-gnueabihf-
CC = $(CROSS_TOOL)gcc
CFLAGS = -Wall -g -std=c99 -D_POSIX_C_SOURCE=200809L -Werror -Wshadow
LFLAGS = -L$(HOME)/cmpt433/public/v4l2_lib_BB -lv4l2

TARGET = capture_exec
SRC = capture.c
DEST_DIR = /home/user/cmpt433/public/myApps/

all: $(TARGET)

$(TARGET): $(SRC)
	$(CC) $(CFLAGS) $(SRC) -o $(TARGET) $(LFLAGS)

install: $(TARGET)
	cp $(TARGET) $(DEST_DIR)

clean:
	rm -f $(TARGET)

.PHONY: all install clean
```

3. Install Media Player (VLC)

1. On host, install vlc media player:

```
(host) $ sudo apt install vlc
```



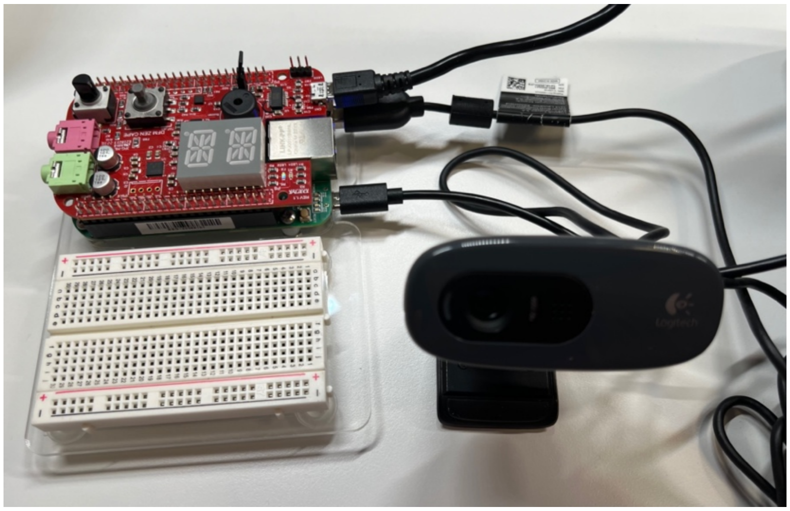
2. Run program on the target:

```
(target) $ ./capture_exec -F -o -c0 | ffmpeg -i pipe:0 -vcodec copy -f mjpeg udp://192.168.7.1:1234
```

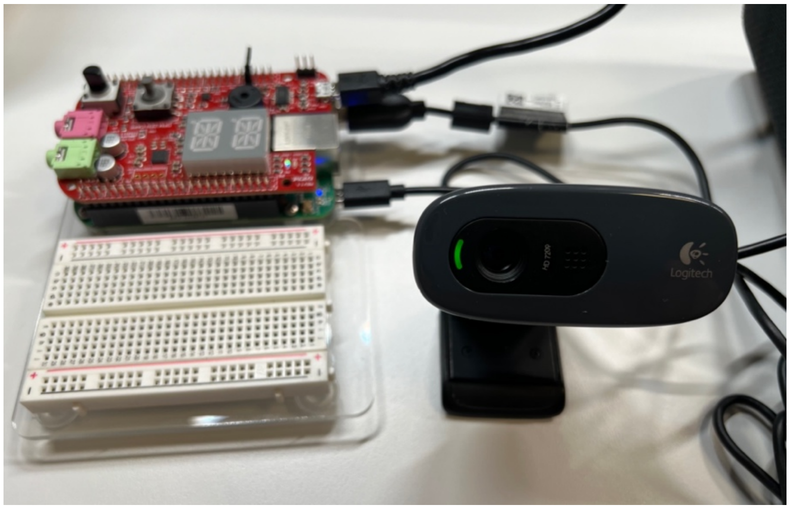
- **'-F':** Activates a specific force format option (in our case, it will set the video format to MJPEG and a resolution of 640 x 360).
- **'-o':** Output the video stream to stdout.
- **'-c0':** Specifies an infinite number of frames to capture.
- **'|':** This symbol pipes the stdout from 'capture_exec' to the stdin of 'ffmpeg'.
- **'ffmpeg':** A multimedia framework to processed and stream video.

- **‘-i pepe:0’:** Tells ‘ffmpeg’ to take input from the piped stream (‘pipe:0’ refers to stdin).
- **‘vccodec copy’**: Instructs ‘ffmpeg’ to copy the video codec directly without reencoding, preserving the original quality and format.
- **‘-f mjpeg’:** Specifies the output format as MJPEG.
- **‘udp://192.168.7.1:1234’:** Directs ‘ffmpeg’ to stream the output video to the given UDP address and port, making the video accessible across the network.

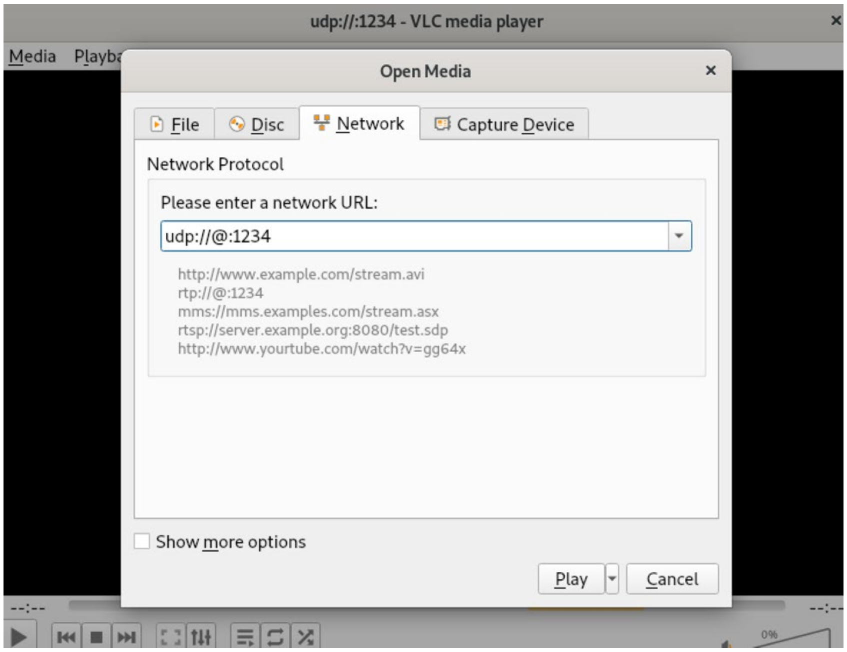
Before running the program, the camera status:



After running the program, the camera status:



3. Open VLC, click ‘Media – Open Network Stream’, set the network URL ‘**udp://@:1234’**, and click the play button. Then, the VLC will show the video stream.



udp://:1234 - VLC media player

Open Media

Network Protocol

Please enter a network URL:

udp://@:1234

```
http://www.example.com/stream.avi
rtp://@:1234
mms://mms.examples.com/stream.asx
rtsp://server.example.org:8080/test.sdp
http://www.yourtube.com/watch?v=gg64x
```

□ Show more options

4. Troubleshooting:

1. If you cannot download the package on the host, please double-check that the host has internet access and enough space.

2. You can check your camera recognized by the BeagleBone:

   (target) $ **lsusb**

3. If you encounter “No packagr ‘openCV found’, enter the following command to ensure the OpenCV is installed on the target:

   (target) $ **sudo apt-get update**

   (target) $ **apt-get install libopencv-dev**

   It will install OpenCV and set the path ‘pkg-config’.

4. If the camera shows a green light, but no frames are sent through UDP, please double-check the camera supporting format:

   ```
   (target) $ v4l2-ctl --list-formats-ext -d /dev/video0
   ```

   Make sure you change the code format.

5. If the program immediately ends after running:

   ```
   …
   select timeout
   pipe:0: Invalid data found when processing input
   ```

   Please change the code in the **capture.c:**

   ```
   /* Timeout. */
                   tv.tv_sec = 2;
                   tv.tv_usec = 0;
   ```

   to:

   ```
   /* Timeout. */
                   tv.tv_sec = 5;
                   tv.tv_usec = 0;
   ```

Then, the program will wait longer to receive video data before returning due to timeout. This may help reduce or avoid timeout errors in some conditions, such as delays in data transfer.

Reference:

1. How to stream OpenCV processed images over the network, https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2019-student-howtos/StreamOpenCVImages.pdf

2. How To Stream Video Through Udp Using A Webcam On BeagleBone Green With Audio, https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2023-student-howtos/StreamUdpVideoAndAudioOnBBG.pdf

3. GitHub "boneCV", https://github.com/derekmolloy/boneCV

4. Networking Guide for BeagleBone (Green or Black), https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/guides/files/Networking.pdf