SFU

How To Guide:
Write an Automated Script for Streaming a Webcam from Beaglebone to a Webpage

Group Name: Quad Lock
Submitted on December 7th, 2022

Members
Abigail Monte de Ramos
Devon Sandhu
Jagpreet Grewal
Zohra Khan Durani

Table Of Contents

Table Of Contents **2**

Introduction **2**

Required Hardware **2**

Required Software **2**

Host Packages 3

BeagleBone Green Packages 3

Wiring **3**

Camera and Beaglebone Green 3

Coding **3**

Integration 4

Scripts 4

Host Scripts 4

BBG Scripts 5

Processes 6

Troubleshooting **6**

References **7**

Introduction

This guide provides step-by-step instructions on how to create automated scripts to stream a webcam to a webpage. This guide references existing guides from previous semesters that can guide you to set-up and stream a webcam, but the aim of this guide is to help create automated scripts to open, refresh the webpage, and begin streaming to allow a smoother start of the streaming with a shorter latency.

Required Hardware

BeagleBone Green
C720 Webcam

Required Software

1) Before compiling, please run the following commands:

Host Packages

```
$ sudo apt-get install nodejs npm FFmpeg openssh-server xdotool expect
$ npm install express
$ sudo apt install nfs-kernel-server nfs-common
```

BeagleBone Green Packages

```
$ sudo apt-get install openssh-server expect
$ sudo apt install nfs-kernel-server nfs-common
```

2) Make sure you have the NFS server working by following the NFSGuide_BeagleBone guide by Brian Fraser or an equivalent.

Wiring

Camera and Beaglebone Green

3) Connect the webcam to your Beaglebone

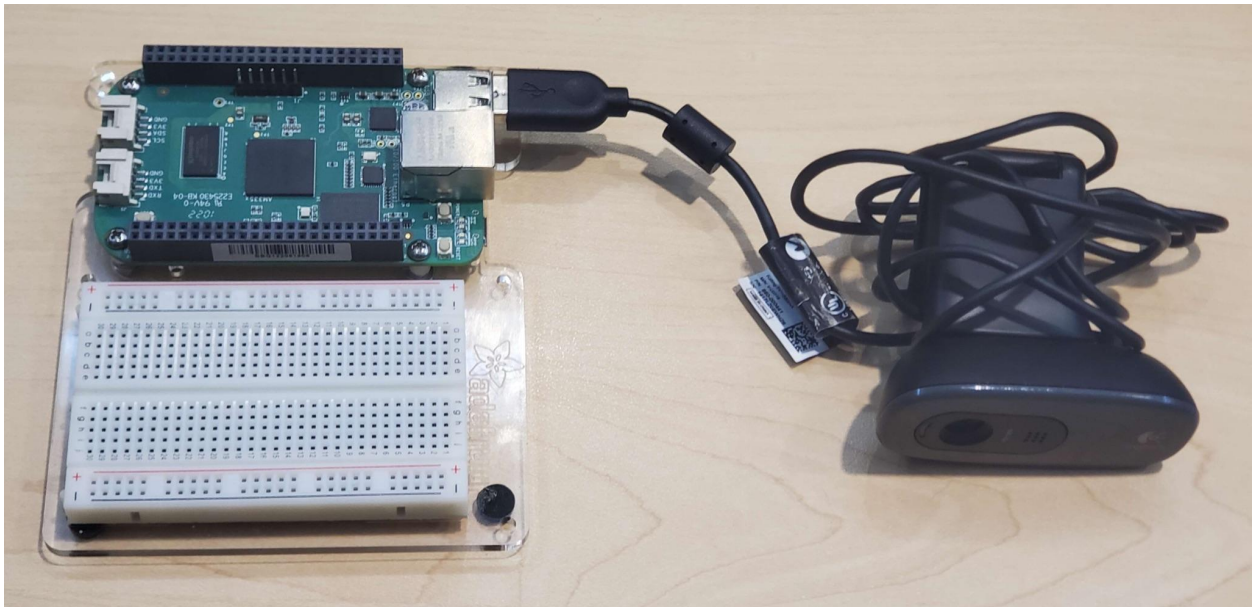

Figure 1: Camera is connected to the BBG's USB port

Coding

4) You can obtain the code for the camera by going to the link in [2]. Scroll down to the Spring 2022 Guides and download the support files from: *How-To Streaming Webcam From BeagleBone To NodeJS Server, with support files* [1].

5) Then, you will need to make the following changes. In the capture directory, go to the capture.c function `static void init_device(void)` and scroll to these lines:

```
fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
      fprintf(stderr, "Force Format %d\n", force_format);
        if (force_format) {
            if (force_format==2){
                  fmt.fmt.pix.width       = 720;
                  fmt.fmt.pix.height      = 720;
```

6) There, make `fmt.fmt.pix.width` and `fmt.fmt.pix.height` equal your camera's pixel width and height. Different cameras may have a different resolution. Other resolutions may work, but there is no guarantee.

Integration

Scripts

7) In order to integrate and automate the actions required to run the camera code such as opening and refreshing the webpage, use the following scripts and save them all to your /cmpt433/work/AnyName directory :

Host Scripts

prepareWebpage.sh

```
#! /bin/bash
# This runs the webpage in a different terminal. Otherwise, the Beaglebone
# messages will show up mixed with server messages.
gnome-terminal -x npm start &
sleep 2
gnome-terminal -x /usr/bin/firefox 192.168.7.1:3000 &
sleep 4
# if errors occur, run the following commands to find and kill whatever
# rogue camera process is running
# $ lsof -i tcp:3000
# Insert PID below
# $ kill -9 <PID>
```

refreshWebpage.sh

```
WID=`xdotool search --name "Mozilla Firefox" | head -1`
xdotool windowactivate $WID;xdotool key F5
```

BBG Scripts

bbgStreamScript.sh

```
#!/usr/bin/expect
# if you get an error,
# you may need to edit /etc/ssh/sshd_config on the host machine.
# Comment out these lines:
# X11Forwarding yes
# X11DisplayOffset 10
# X11UseLocalhost yes
set timeout 60
set -o errexit
set -o nounset
# ???? represents your username and password on the host
# You may need to change the directories
# If you have any executables that run when logging in to the host or BBG,
# please disable them
set password ????
spawn ssh -X ????@192.168.7.1
expect "?assword:*"
send -- "$password\r"
send -- "\r"
sleep 2
expect "?$"
send -- {export DISPLAY=:0.0}
send -- "\r"
send -- {~/cmpt433/work/quad-lock/refreshWebpage.sh}
send -- "\r"
expect "?$"
send -- {exit}
send -- "\r"
interact
```

bbgStartCapture.sh

```
#! /bin/bash
set timeout 30
sleep 3
for i in {1..10}
do
    ./capture
    ./bbgStreamScript.sh &
done
```

Processes

8) Launching the **prepareWebpage.sh** script from code causes errors with the other scripts targeting the webpage, so it must be run manually on the host before the program is executed. Otherwise, we can use forks in c to launch the other scripts after certain conditions are met.

```
pid_t pid = fork();
if(pid <0 ){
    fprintf(stderr, "Fork failed\n");
    exit(-1);
}
// Make the child process execute the script
// The BBG scripts must be in the same directory as the executables
else if(pid == 0){
    execl("/bin/bash","sh","./bbgStartCapture.sh",NULL);
} else{
    printf("Child %d running\n",pid);
}
while(true){
    // End of program
    Alarm_initAndSound();
    Common_sleep_ms(500);
}
```

Troubleshooting

- Ensure that your Virtual machine username and password are entered in the specified areas in the **bbgStreamScript.sh** script
- Disable any programs that start when you log in like a LED and button game
- You may need to edit /etc/ssh/sshd_config on the host machine. Comment out these lines:
  X11Forwarding yes
  X11DisplayOffset 10
  X11UseLocalhost yes
- Run the following commands to kill whatever camera process is running on the host. It will rarely be necessary but sometimes the camera does not shutdown properly.
  lsof -i tcp:3000
  kill -9 <PID>

References

[1] https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/ensc351/links/files/2022-student-howtos/StreamingWebcamFromBeagleBoneToNodeJSServer.pdf

[2] https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/ensc351/links

[3] https://github.com/derekmolloy/boneCV/blob/master/capture.c