# IOT Based Audio Video Surveillance

[1]Mohammed Abdullah Khan,[2]Mohd Asadullah Khan,[3]Syed Sabair Hussain, [4]Mohammed ArifuddinSohel
[1]Diploma, B.E,[2]B.E,[3]B.E, [4]B.E, M.S, Ph.D
[1]ECE,
[1]MJCET, Hyderabad,India.

***Abstract:***Fast paced technological growth in the field of IoT has made surveillance a very promising application area. With great extent of smartness over the things that are controlled by internet security is one of the applications that everyone needs to be controlled remotely. This paper presents an IoT application for live video and audio streaming that can be used for surveillance from anywhere and at anytime. This system provides the IP address based live streaming using a Raspberry Pi module with Fast forward moving picture expert group (FFMPEG). A MIC and loud speaker for audio connectivity is also provided in addition to a USB Camera for video connectivity.

***IndexTerms*– surveillance, IP addresses, FFMPEG, live streaming**.

## I. INTRODUCTION

In today's world of rapidly evolving technologies with tremendous advancements in information and security sector and hectic schedules making one's life fast-track, there is a growing demand for surveillance. Surveillance refers to monitoring a place of interest using some or the other kind of security like audio and video surveillance audio could be with the help of Mic and loud speaker and video can be with the help of CCTV these both are based on IP address. In addition, FFMPEG provides highly efficient codec algorithm that can meet requirements of real-time video / audio analysis [9].

The proposed work has ability to control live video and audio streaming over the internet via IP address. Streaming and controlling is possible from anywhere over the world. This paper is structured as follows. Section II presents the remote surveillance system, Section III presents booting up the raspberry pi and installing FFMPEG, Section IV shows the block diagram of the system. Section V explains the working and system testing, finally the paper is concluded with Section VI and section VII as future scope.

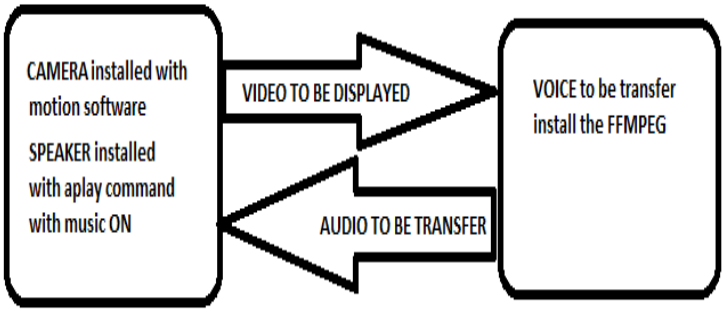
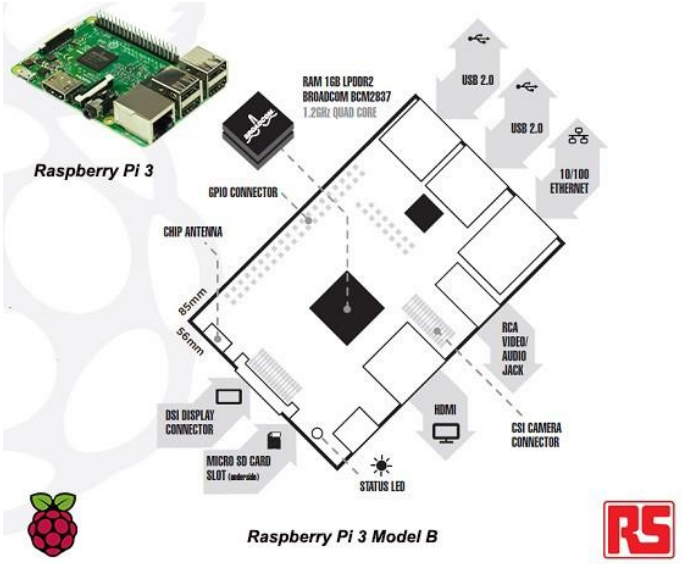



Fig. a. Basic concept of our project

## II. REMOTE SURVEILLANCE SYSTEM

*INTRODUCTION:*

Our project deals with building a remote surveillance system at affordable price tag using a credit-card sized, inexpensive, internationally used and acclaimed computer board called RASPBERRY PI. The project is aimed at building an IP based camera which when installed at a place of interest connects to the internet using Wi-Fi and it enables us to view the live video stream on any internet enabled device like Laptop/PC/Smartphone from anywhere in the world [12]. The Raspberry Pi layout is shown below:




## III. REMOTE SURVEILLANCE BOOTING AND INSTALLING FFMPEG

In this chapter, the installation of Raspbian operating system and Booting up of Pi along with remote GUI access and wireless network connectivity will be discussed. Installation can be done using a computer or a laptop. If we do not want any external display we can also use the software called PUTTY by which just we need to enter the IP address and type user name as "pi" and password as "raspberry" then a command window will be open just we need to enter "startlxde" now you can use windows OS as well as Raspbian OS on laptop [3].

Making the connection wireless helps in making the Pi mobile and we can install the whole surveillance system at any place of our interest inside the range of the wireless network [2].

Thus making the whole system more sophisticated and up to the market standards. The method is explained as follows. Get the latest firmware drivers by running the following commands:

sudo apt-get update

sudo apt-get upgrade

Check and make sure your Wi-Fi dongle has installed correctly (LSUSB):

Lsusb

Now we need to install webcam for that we need to type some commands on terminal window of raspberry pi:

sudo apt-get install fswebcam

fswebcam --no-banner -r 640x480 image.jpg

After installing this webcam only the image will be captured &For live surveillance we need to install motion software to get live streaming via IP address:

sudo apt-get update &&sudo apt-get install etcher-electron

sudo apt-get update &&sudo apt-get upgrade

sudo apt-get install motion

Now we need to apply the changes in" motion.conf" file as shown:

***Accessing the Live Video Stream:***

In order to achieve a stable feed from the webcam the following changes should be made in the motion.conf file:

daemon on #default off (This allows the motion to run in the background)

framerate 30 #default 2 (increased framerate)

width 640 #default 320 (changed width to match that of the webcam)

height 480 #default 240 (changed height to match that OF the webcam)

threshold 2000 #default 1500 (Detection in change of no. of pixels)

pre_capture 2 #default 0 (captures 2 frames before motion was detected and adds that to the videos to make them smoother)

post_capture 5 #default 0 (captures 5 frames after motion was detected and adds that to the videos to make them smoother)

output_normal off #default on (this disables storing images, since we only require video)

ffmpeg_video_codec msmpeg4 #default swf (msmpeg4 is accepted by windows media player, hence easier to play on Windows)

target_dir /etc/motion/media #default /tmp/motion (changed the directory where videos will be stored because the videos (or) images stored in tmp file are deleted once the Raspberry Pi is shutdown )

webcam_maxrate 5 #default 1 (increase the max frame rate on live stream)

webcam_localhost off #default on (allows us to set up a live stream of the webcam)

Now in order to activate the video stream the following command is typed in the terminal:

sudo motion

Now we can access live streaming as shown below:

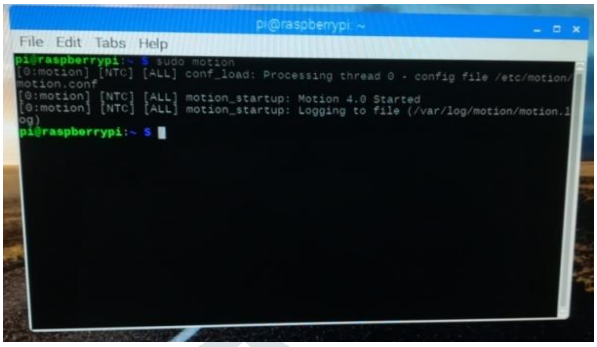



Fig. 1. Command used to active motion software

Now we can access live by simply entering an IP address as shown below:

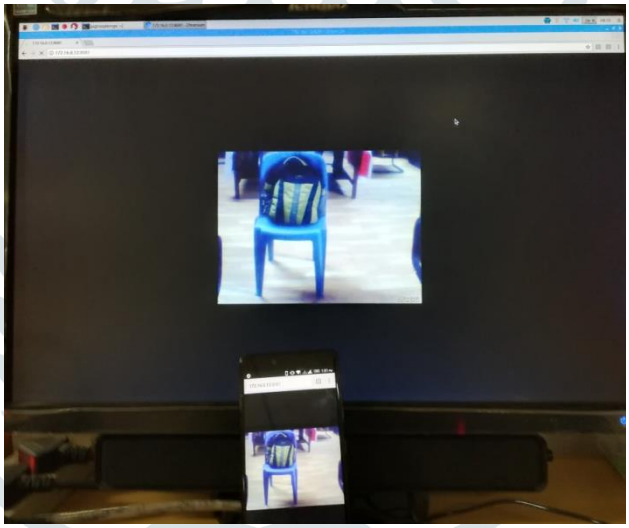


Fig. 2. Live stream by IP address

***Accessing the Live Audio:***

***FFmpeg on Raspberry Pi 3 with h264 support***

Raspberry Pi 3 Model B with Raspbian Jessie Lite

```
sudo apt-get update
sudo apt-get upgrade
sudosh -c 'echo "deb http://www.deb-multimedia.org jessie main non-free" >> /etc/apt/sources.list.d/deb-multimedia.list'
sudosh -c 'echo "deb-src http://www.deb-multimedia.org jessie main non-free" >> /etc/apt/sources.list.d/deb-multimedia.list'
sudo apt-get update
sudo apt-get install deb-multimedia-keyring
sudo apt-get update
sudo apt-get remove ffmpeg
sudo apt-get install build-essential libmp3lame-dev libvorbis-dev libtheora-dev libspeex-dev yasmpkg-configlibfaac-dev libopenjpeg-dev libx264-dev
cd /usr/src/
sudo apt-get install git
```

```
sudogit clone git://git.videolan.org/x264
cd x264/
sudo ./configure --host=arm-unknown-linux-gnueabi --enable-static --disable-opencl
sudo make
sudo make install
cd /usr/src
sudogit clone https://github.com/FFmpeg/FFmpeg.git
cdFFmpeg/
sudo ./configure --arch=armel --target-os=linux --enable-gpl --enable-libx264 --enable-nonfree
sudo make -j4
sudo make installffmpeg -encoders # test it works
```

***1.Install and test FFMPEG on Raspberry Pi 3 (with Docker)***

Docker containers were used, so ffmpeg was not installed directly on the RPi. The docker image has been pushed tohttps://hub.docker.com/r/tgogos/ffmpeg/, you can save time by pulling it and then start testing... If you'd like to reproduce the whole process take a look at the section below "Download ffmpeg source code / compile

***Test environment***

- Raspberry Pi 3
- Host OS: Raspbian Jessie Lite image from the official site)
- Docker image to start with (FROM): resin/rpi-raspbian:jessie-20160831
- Video file: Big Buck Bunny by Blender Foundation https://www.youtube.com/watch?v=YE7VzlLtp-4

***Download FFMPEG source code / compile***

* Beware that this compilation procedure takes a few hours to finish

```
apt-get update
apt-get upgrade
apt-get install git
git clone git://source.ffmpeg.org/ffmpeg.git
cdffmpeg
apt-get install build-essential
apt-get install pkg-config
./configure
make
make install

# STEP 1 - Raspberry Pi (eth0: 10.143.0.246):
# big_buck_bunny.mp4 is not provided within the docker image
# if it is available on the docker host (RPi), use `docker cp...` to copy it inside the image
ffmpeg -re -i big_buck_bunny.mp4 -vcodec mpeg4 -an -b 1024k -s 640x480 -f mpegts udp:10.143.0.245:9999?pkt_size=1316

# STEP 2 - Demo pc (eth0: 10.143.0.245):
ffplay udp://10.143.0.246:9999
```

***2 Test with Docker & OVS***

Open vSwitch must be installed on the docker host which is the Raspberry Pi, not on any other docker container. Network configuration comes from this source: https://developer.ibm.com/recipes/tutorials/using-ovs-bridge-for-docker-networking/

```
# first add and configure the bridge
```

```
sudoovs-vsctl add-br ovs-br1
sudoifconfig ovs-br1 192.168.0.1 netmask 255.255.255.0 up
exportpubintf=eth0
exportpriintf=ovs-br1
sudoiptables -t nat -A POSTROUTING -o $pubintf -j MASQUERADE
sudoiptables -A FORWARD -i $priintf -j ACCEPT
sudoiptables -A FORWARD -i $priintf -o $pubintf -m state --state RELATED,ESTABLISHED -j ACCEPT

# then, run a container without network
docker run --net=none --name=ffmpeg_nonet_privileged --privileged -itdtgogos/ffmpeg:latest
docker exec ffmpeg_nonet_privilegedifconfig # this should print out only 'lo'

# add a NIc
sudoovs-docker add-port ovs-br1 eth0 ffmpeg_nonet_privileged --ipaddress=192.168.0.2/24 --gateway=192.168.0.1
docker exec ffmpeg_nonet_privilegedifconfig # this should print out 'lo' and 'eth0'

# test (RPi: 10.143.0.246, Demo pc: 10.143.0.245)
# big_buck_bunny.mp4 is not provided within the docker image
# if it is available on the docker host (RPi), use `docker cp...` to copy it inside the image

docker attach ffmpeg_nonet_privileged
ffmpeg -re -i big_buck_bunny.mp4 -vcodec mpeg4 -an -b 1024k -s 640x480 -f mpegts udp:10.143.0.245:9999?pkt_size=1316
ffplay udp://10.143.0.246:9999 # for the Demo pc



# commands you might need later...
sudoiptables --flush
sudoovs-vsctl del-brname_of_bridge
```

***3. Test with Docker (as transcoder)***

```
# 1 - Demo PC sends a video
# 2 - Docker container with ffmpeg transcodes it and sends it back


# STEP 1 - Demo pc (eth0: 10.143.0.245):
# start sending the video file at udp port 9999...
ffmpeg -re -i big_buck_bunny.mp4 -vcodec mpeg4 -an -b 1024k -s 640x480 -f mpegts udp://10.143.0.246:9999?pkt_size=1316


# STEP 2 - Raspberry Pi (eth0: 10.143.0.246):
# run a container and expose udp port 9999
docker run -itd --name=ffmpegTranscoder -p 9999:9999/udptgogos/ffmpeg
docker attach ffmpegTranscoder
# send back through udp port 9998
ffmpeg -re -i udp:10.143.0.245:9999 -vcodec mpeg4 -b:v 2048 -f mpegts udp:10.143.0.245:9998?pkt_size=1316


# STEP 3 - Demo pc (eth0: 10.143.0.245):
ffplay udp://10.143.0.246:9998
```

***Raspberry Pi into an audio spying device:***

If you are a spy or work for the CIA or the FBI, this will come in handy in your knowledge. Given the size and price of the Raspberry Pi, it occurred to me that using it as a spying device (which are actually really expensive) would be really appropriate.

In this case, I'll explain how to convert your Raspberry Pi into an audio streaming device, or audio spying device, however suits best to you. Use it under your own responsibility.

***Plugging the microphone and loading the module:***

The Raspberry Pi has an audio output jack, but unfortunately there is no input port. We will then need a USB microphone. If you have one, keep reading, if not, go buy one ;) (or plug a webcam with microphone).

Once the microphone is plugged in, you might have to load the audio module by typing the following command if it isn't.

sudomodprobe snd_bcm2835

***Recording and playing a test file:***

Now we can try to record some audio into a file by running:

arecord -D plughw:1,0 test.wav

Just press CTRL+C once you think you've got enough recording. Now let's play it to see if it works! But first, plug your earphones to the audio output of your Raspberry Pi

aplay test.wav

Did you hear anything? Great! Your Raspberry is ready to "rock"ord some audio!

If you want to record louder or adjust some parameter, you can use the alsamixer tool to play with the input/output levels of your microphone.

**alsamixer**

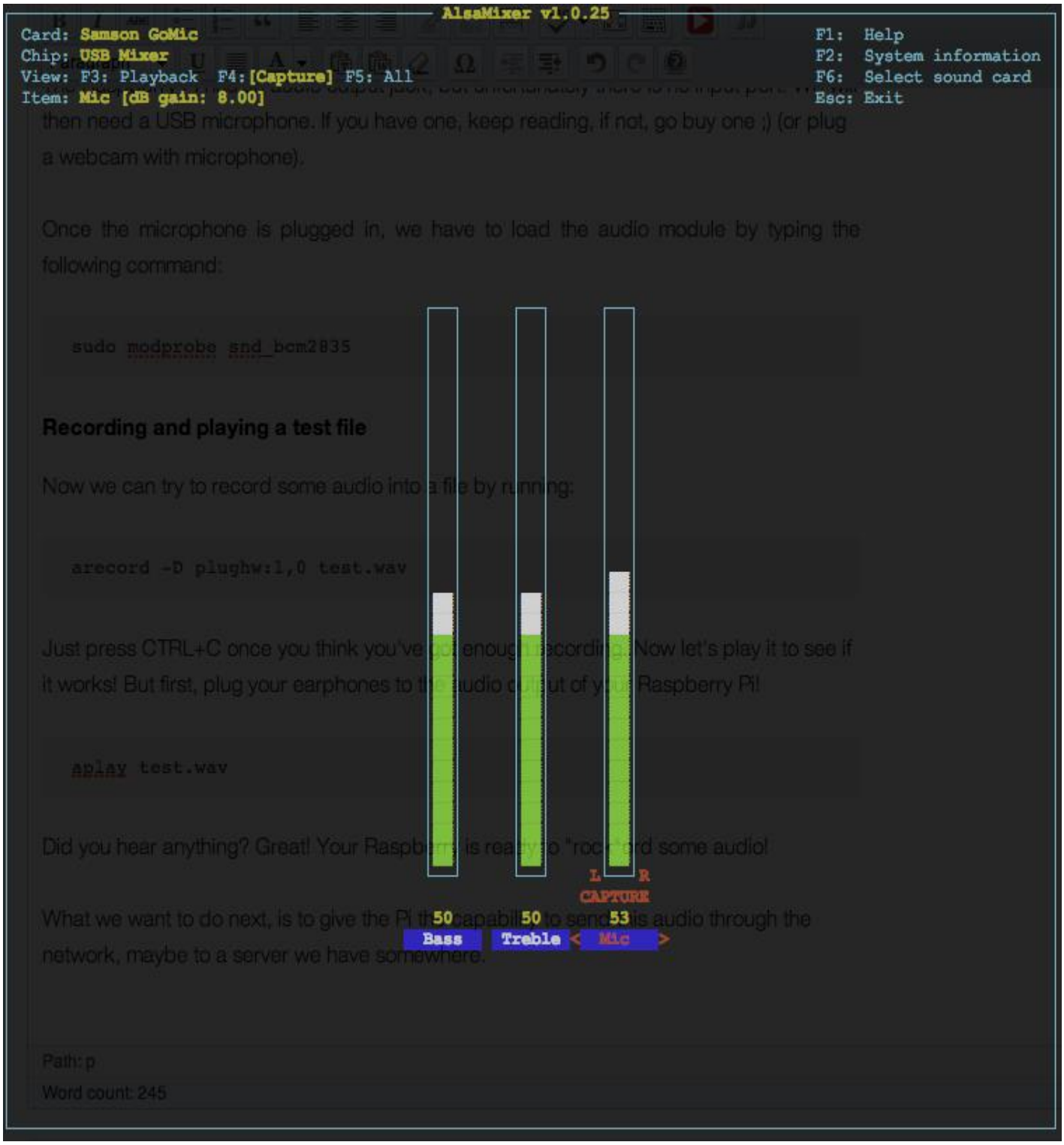



Fig. b. AUDIO MIXER

***Alsamixer:***

Once you have set your settings, remember to store your changes:

**sudoalsactl store**

***Streaming the audio to another PC:***

What we want to do next, is to give the Pi the capability to send this audio through the network, maybe to a server we have somewhere. In order to do that, we will pipe the audio from the microphone into an ssh communication to the destination computer. As simple as:

**arecord -D plughw:1,0 -f dat | ssh -C user@remoteipaplay -f dat**

Now you have everything set! Switch on your speakers in the destination PC, and hear what the Raspberry Pi is spying from hundreds of miles! [4].

## IV. WORKING

As the system is switched on camera starts recording the video and the voice being transfer, through IP address i.e. Using Wi-Fi. Raspberry Pi2 communicates with Raspberry pi3 via server.At client-side client opens the web page and request the server via entering valid IP address into address bar of web page. User can watch and hear live video and audio from anywhere in the world. Motion and the position of system can be controlled via pressing the control buttons that are provided on web page. Also, the movement of camera is possible with control buttons available on web page [6].

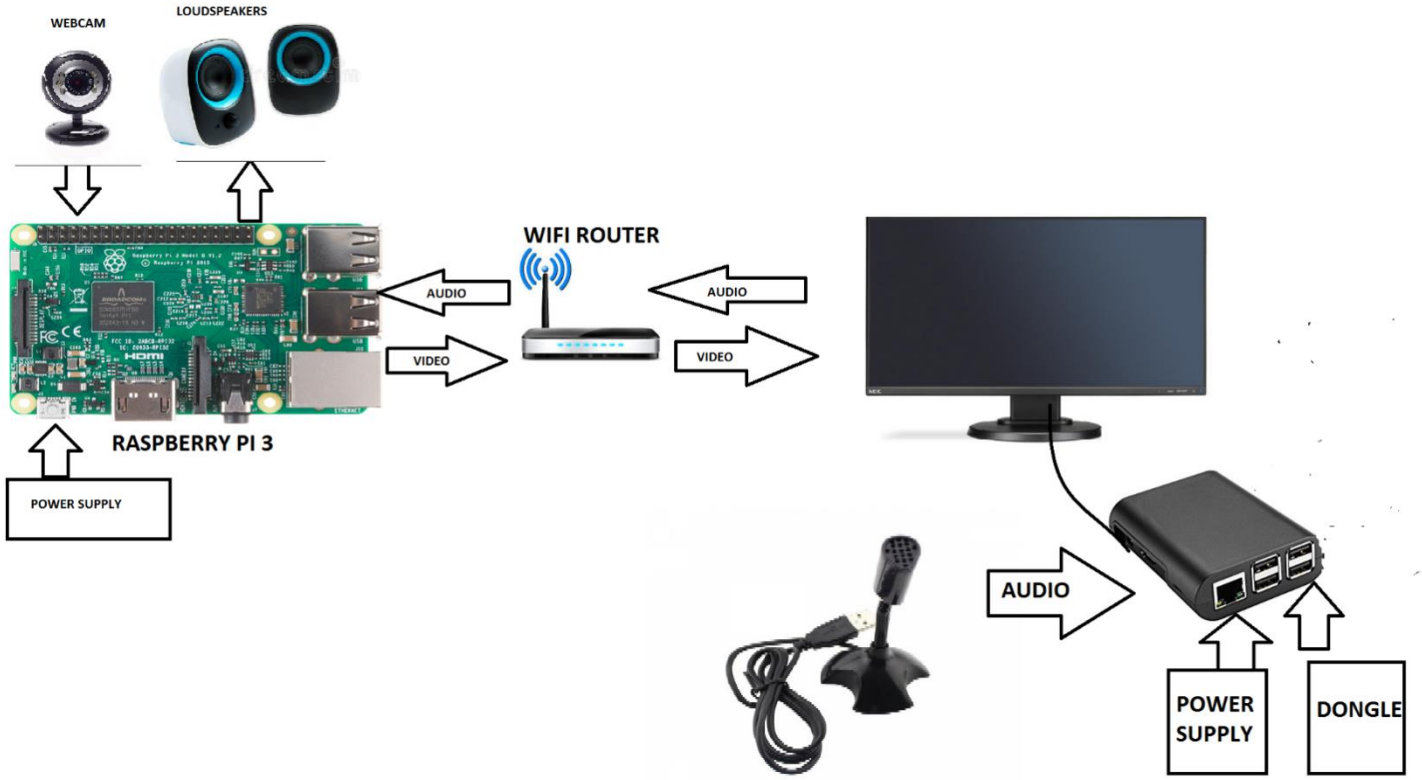



Fig. 3. BLOCK DAIGRAM

Fig. 3 shows there are two parts in our project

1) Camera will transfer the video at the destination
2) And the other part that is audio transfer is done in the reverse of the camera as shown in above fig

This video and audio can be watched and surveillance lively on webpage.

## V. TESTING AND RESULTS

Testing was carries out at different levels. Testing of connection between Raspberry Pi2 and Raspberry pi3 and different component like camera, speakers and mic are successfully observed. Testing of whether system is connecting to internet or not is successfully observed.

### *A. Problems faced while testing:*

Some problems that are faced during test are:

1. Improper internet connection due to poor signal strength in some area.

2. Some delay obtained while transmission the audio from one pi to another.
3. Problem to upload video or to store.

***B. Hardware used:***

*At the setup of audio transfer:*

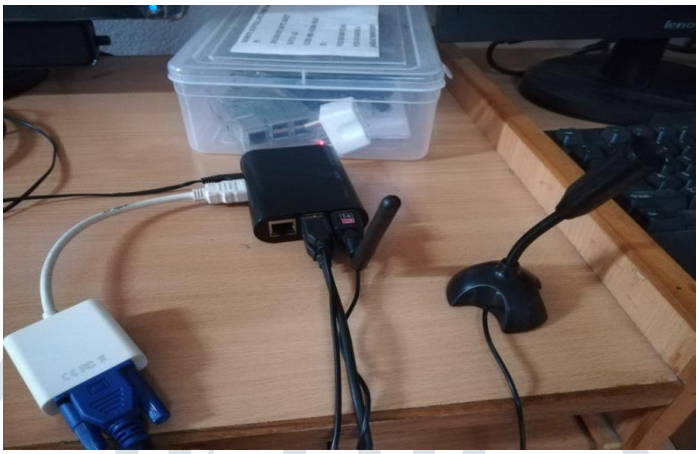


Fig. 4. Photo of audio setup

*Now at video surveillance setup*:

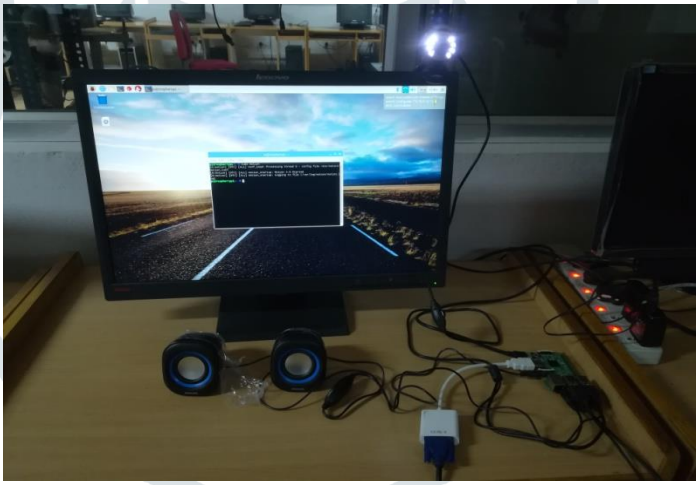


Fig. 5. Photo of video setup

***C. Results:***

*At audioside:*

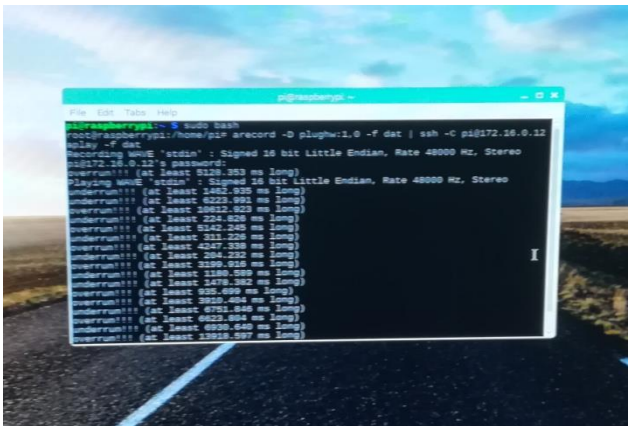


Fig. 6. Output of audio

*At video side:*

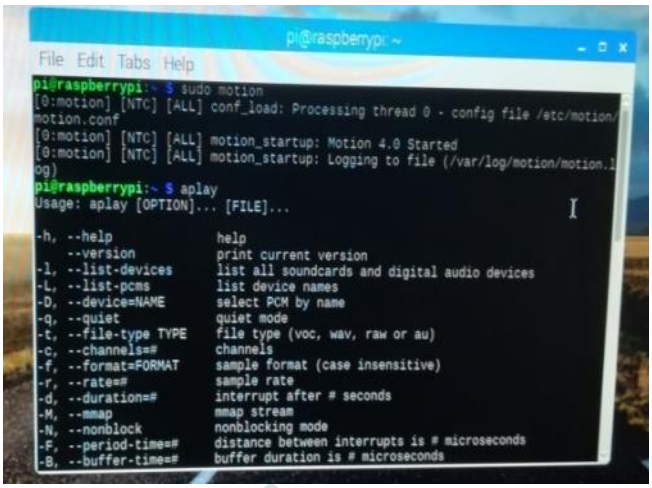



Fig. 7. Output of video setup

***D. Advantages of the System:***

1. System is portable and can be used in many places
2. Low cost and efficient.
3. Common man with basic knowledge of computer can operate it easily.
4. Controlling of video movement and with audio of whole module from web page is possible with PC as well as Android based mobile phone.

.

## VI. CONCLUSION

In this paper a Remote surveillance system was designed using Raspberry Pi's which detects motion and also transmit voice. whenever there is a change in pixel values and it facilitates the viewing of live video feed of the surveillance camera on any internet enabled device like a Smartphone/Laptop/PC with a modern browser support from anywhere in the world. How the video surveillance is obtained same and parallel y the audio is been transmitted.

## VII. FUTURE SCOPE

The uses and applications of these camera systems traverse along various domains. A few aspects of this work were not fully improved due to the different possible combinations of trade-offs and limited time availability. To mention a few: The performance of the system can be enhanced by mounting light sources along with the surveillance system to improve lighting conditions. Another useful feature could be sending email notification whenever motion is detected. The surveillance system can be extended to work as a home automation system with the addition of sensors like temperature and humidity sensors etc. 76 Integrating the surveillance system with GSM module for SMS.Providing power back-up by adding a battery pack to the surveillance system. Or opting for solar power for back-up

## REFERENCES

[1] RupaliIkhankar, Varun Kuthe, SarangUlabhaje, ShrutiBalpande, MahendraDhadwe, "Pibot: The Raspberry Pi Controlled Robot For Surveillance & Live Streaming ", 2015 and International Conference on Industrial Instrumentation and Control (ICIC), 2014.
[2] Andrew S. Tanenbaum. "Computer Networks-third Edition", Pearson Education, 2007
[3] Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes. "SSH, The Secure Shell: The Definitive Guide Second edition", O'Reilly Media, May 2005.
[4] https://www.jeffreythompson.org/blog/2014/11/13/installing-ffmpeg-for-raspberry-pi/ https://github.com/tgogos/rpi_ffmpeg
[5] https://blog.mutsuda.com/raspberry-pi-into-an-audio-spying-device-7a56e7a9090e
[6] "Implementation of Cloud Based Live Streaming for Surveillance" was published by Neel Oza, N. B. Gohil in International Conference on 6-8 April 2016 at (IEEE)
[7] Kenny Ju Min Yeoh and Hwee Ling Wong, "Web-based Remote Navigational Robot for Multiclass Human-Robot Interaction", 2012 IEEE conference on Sustainable Utilization and Development in Engineering and Technology.

[8] Viren Pereira, VandykAmsdemFernandes and JunietaSequeira, "Low Cost Object Sorting Robotic Arm using Raspberry Pi", 2014 *IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS)*, September 26-27,2014, Trivandrum.
[9] The FFmpeg developers. FFmpeg Documentation. May 2013
[10] F. Mano Anthony Nayagam, K.M. Sharmilee. "Web Hosting and Live Streaming Using Raspberry Pi", International conference on innovations in Engineering & Technology (ICIET), 2014.
[11] Sneha Singh, PradnyaAnap, Yogesh Bhaigade, Prof.J.P.Chavan. "IP Camera Video Surveillance Using Raspberry Pi", International Journal of Advanced Research in Computer & Communication Engineering, Feb 2015.
[12] Sneha Singh, Tina M. Patil,Kajal J. JadhavMr.Piyush Rode titled "Design and Development of Remote control Surveillance Vehicle", International Journal of Advancements in Research & Technology, Nov 2013.
[13] Sarthak Jain, Anant Vaibhav, Lovely Goyal. "Raspberry Pi based Interactive Home Automation System through E-mail", International Conference on Reliability, Optimization and Information Technology -ICROIT, Feb 2014
[14] AamirNizam Ansari, Mohamed Sedkyl, Neelam Sharma, Anurag Tyagil. "An Internet of Things Approach for Motion Detection using Raspberry Pi", International Conference on Intelligent Computing and Internet of Things (IC1T), 2015