

# **Embedded Server Design for Media Streaming in Multi Media Access**

V.Sai Kumar<sup>1</sup>, T.Vinay Simha Reddy<sup>2</sup>

M-Tech, Dept of ECE, Malla Reddy College of Engineering and Technology, Maisammaguda, Telangana, India<sup>1</sup>

Assistant professor, Dept of ECE, Malla Reddy College of Engineering and Technology, Maisammaguda,  
Telangana, India<sup>2</sup>

**ABSTRACT:** The arrangement of spilling media server exploits low power utilization and elite of RASPBERRY PI implanted processor, and makes utilization of the Linux portion as the installed advancement stage which is anything but difficult to tweak for a specific application. This paper breaks down and presents the outline procedure of an Embedded Streaming Media Server. The equipment piece and the product engineering of the framework are expounded. As per the constant transmission prerequisites of the sound and video spilling of interactive media, a continuous stream media server is outlined in light of the RTP/RTCP convention. The gushing media server receives simultaneous server calculation. The acknowledgment of code depends on the UDP convention correspondence model of the attachment organize programming which is under nature of Linux. The framework is blame tolerant with superior and the misfortune rate of the bundle is additionally low. The server outlined is particularly reasonable for the versatile video checking application.

**KEYWORDS:-**RaspberryPi, Camera, RTP/RTCP.

## **I. INTRODUCTION**

In the rapid development of the Internet today, people have a lot of requirements about the network information. In addition to get pictures and text of information, people most of the time need audio and video media information. Because the audio and video containing a large amount of data, the traditional HTTP download mode requires longer time. Some of the media applications such as network video conference, live television hope that waiting delay time is as short as possible and with good fluency of transmission. In order to improve the audio-visual effects of the network, streaming media technology emerges as the times require. Streaming media technology which is a continuous media technology to be transmitted in the network supports audio and video files to play while they are downloaded. The audio and video files are played fluently as they are played on a local file. Streaming media server also is a key platform which the operators use to provide users with the video service.[2] Its main function includes acquisition of the media content, cache, scheduling and transmission playing. Streaming media server is also the most important part of it. This paper takes advantage of Raspberry Pi, and uses embedded Linux as embedded software platform to realize an embedded streaming media server which is applied in video monitoring. The realization of code is based on the UDP protocol communication model of the socket network programming which is under the environment of Linux. Owing to the advantages of small volume, low power consumption, low cost and reliability of the embedded streaming media system, an embedded streaming media system is applied more and more widely in video monitoring products.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 7, July 2018

## II. PROPOSED SYSTEM

The proposed system has been designed to give the live video transfer the block diagram shows that the project is divided into six sections.

Implementation is discussed in section 1. Block diagram is showed in section 2. RTSP Server is discussed in section 3. Software development is discussed in section 4. Experimental results are shown in section 5. Conclusion is discussed in section 6. References are showed in section 7. The raspberry pi will be somewhere at a remote place, in surveillance area, spying the activities on webpage using LAN.

## III. IMPLEMENTATION

### Architecture Diagram:

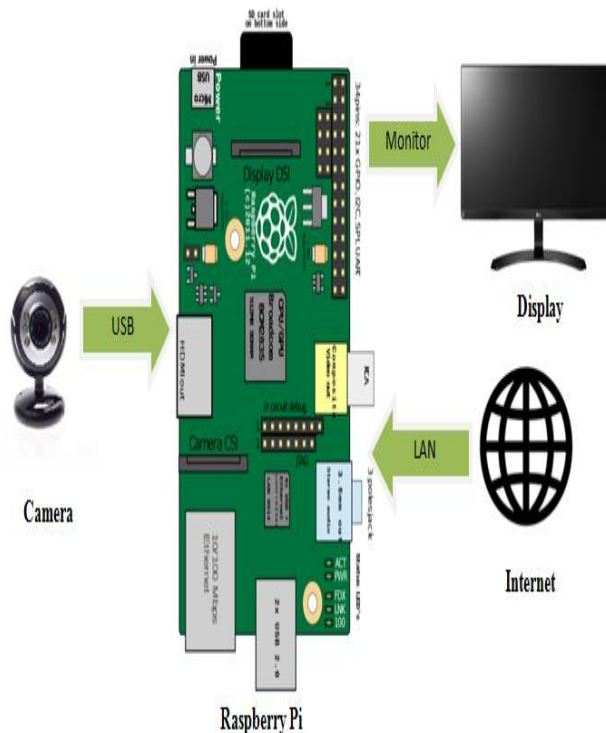


Fig1: Architecture Diagram

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 7, July 2018

## Schematic Diagram:

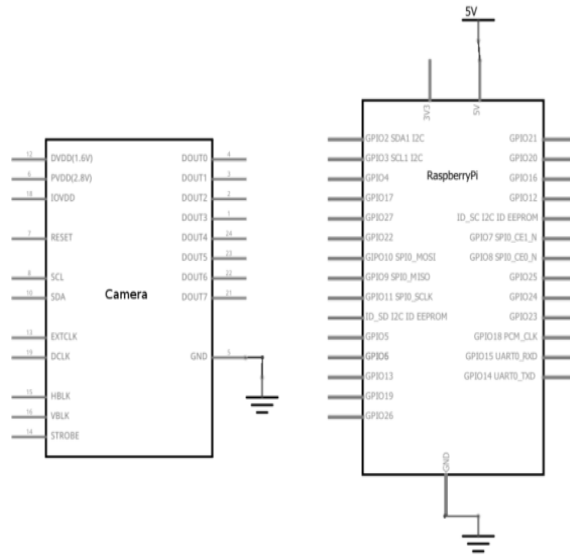


Fig2: Schematic diagram

## IV. BLOCK DIAGRAM

Power Supply is taken from 5V adaptor for raspberry pi controller. Camera is connected to controller using USB port. Internet is connected through LAN port.[2] The Capturing Video is displayed in the IoT page.

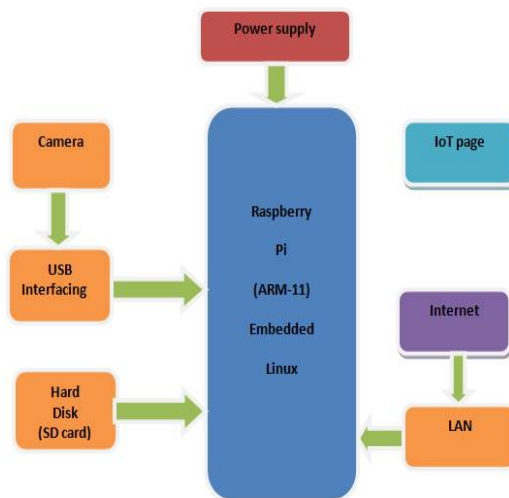


Fig3:Block Diagram

# International Journal of Innovative Research in Science, Engineering and Technology

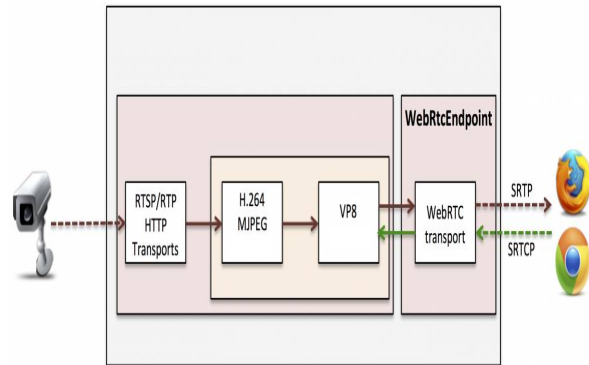
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 7, July 2018

## RTSP Streaming in Raspberry pi

At a certain day I wanted to do some video surveillance, see if my kids are safe playing in the street. After a little while searching on the net for viable camera systems I quickly realized that professional HD IP cameras don't come cheap these days. Since I'm on a tight budget, I started searching for alternatives for this expensive equipment, and since I was already playing around with a Raspberry Pi decided to go further in that direction. [3] So I ordered a CSI camera module on some vendor outlet website, and started tinkering with it. Luckily for me there are tons of articles on the net with information about the Raspberry Pi and it's camera module. Since I already owned a Synologynas with surveillance station software on it, the choice to build a compatible IP camera for this purpose was logical, since surveillance station already had a website and mobile applications where I could view my video feed.



If I wanted to build my IP camera to be compatible with my Synologynas, it had to support either MJPEG or H.264 RTSP streaming. I knew there is a GPU in the Raspberry PI capable of doing H.264 encoding, and a utility called raspivid to drive all this and get this to the stdout of our Rasbian OS, so it seemed logical to take benefit of that knowledge and go the H.264 way for streaming.[4] [5] Now that we had a way to get our video stream, the only thing that remained was pipe this raw video feed into some RTSP server for dispersion to my Synologynas. After searching the net (again) for a while found that there are source-code libraries for standards-based [6] [7] RTP/RTCP/RTSP/SIP multimediasstreaming, suitable for embedded and/or low-cost streaming applications provided by Live Networks, you can find their website at [http://192.168.1.17/stream\\_simple.html](http://192.168.1.17/stream_simple.html) and put on SD card with win32diskimager.

This is a minimal Rasbian installation which is compatible with the official Foundation Raspbian distribution. (it's totally not a requirement, you can just as easily start off with the official foundation distribution) After booting up, log in with root and raspberry as password, then perform below steps to expand the root partition, default this is only 512 Mb, we want this bigger for the rest of our software:

## Flow chart:

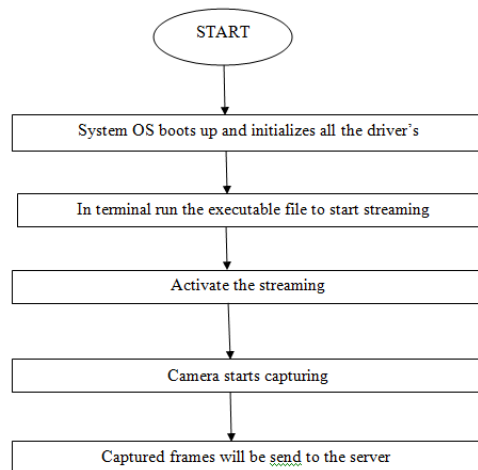


Fig: 4Flow chat of video streaming Operation

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 7, July 2018

## RTSP Sever

Fig: 5 Functionalmodules Architecture ofstreaming media

## V. SOFTWARE DEVELOPMENT

### Raspbian:

Raspbian uses PIXEL, Pi Improved windows Environment, Lightweight as its main desktop environment as of the latest update. It is composed of a modified LXDE desktop environment and the Openbox stacking window manager with a new theme and few other changes. [5] The distribution is shipped with a copy of computer algebra program Mathematical and a version of Minecraft called MinecraftPias well as a lightweight version of Chromium as of the latest version.

### Raspberry pi

The Raspberry Pi 3+ utilizes a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-center ARM Cortex-A53 processor, with 512 KB RAM.

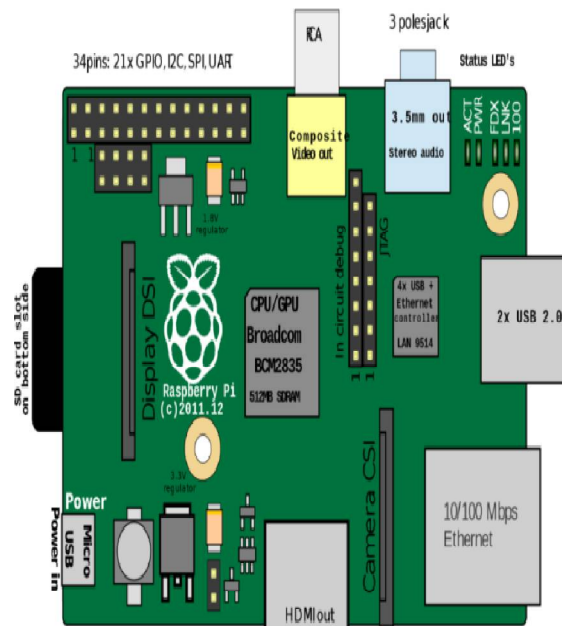


Fig:6Software Development Kit

### Camera

A Camera is a camcorder that feeds or streams its picture continuously to or through a PC to a PC organizes. Whenever "caught" by the PC, the video stream might be spared, seen or sent on to different systems by means of frameworks, for example, the web, and messaged as a connection. [7] At the point when sent to a remote area, the video stream might be spared, seen or on sent there. Not at all like an IP camera (which associates utilizing Ethernet or Wi-Fi), a webcam is for the most part associated by a USB link, or comparable link, or incorporated with PC equipment, for example, workstations.

## International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 7, July 2018



Fig:7 Camera

### Steps followed to start streaming:

1. Login to pi  
Username: pi  
Password: raspberry
2. Select Directory path  
pi@raspberrypi:~\$ cd motion  
pi@raspberrypi:~/motion\$ cd mjpg-streamer/
3. Run the executable file  
pi@raspberrypi:~/motion/mjpg-streamer \$ sudo ./mjpg\_streamer -i "/input\_uvc.so -f 10 -r 640x480 -n -y" -o  
"/output\_http.so -w ./www -p 80"
4. Open the link and observe the video streaming  
[http://192.168.1.17/stream\\_simple.html](http://192.168.1.17/stream_simple.html)  
[https://grizzliest-ruff-7414.dataplicity.io/stream\\_simple.html](https://grizzliest-ruff-7414.dataplicity.io/stream_simple.html)

## VI. EXPERIMENTAL RESULTS

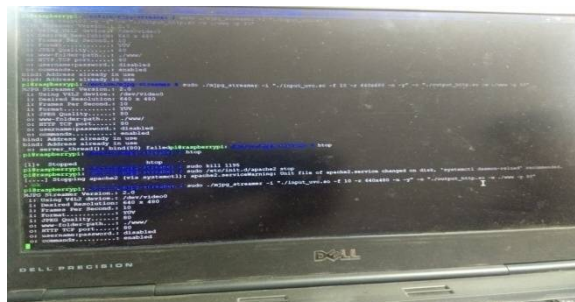


Fig: 8 Commands executing from the system

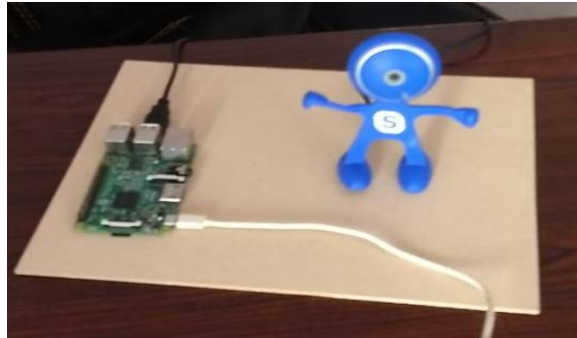


## International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 7, July 2018



**Fig9:** Working Prototype



**Fig10:** Out Put Display

### VII. CONCLUSION

This paper is primarily based on the embedded streaming media server which is used within the video monitoring. We advise a brand new technique to study it. System of streaming media server takes gain of low strength consumption and excessive performance of RASPBERRY PI embedded processor, and makes use of the Linux kernel which is simple to customize for a selected software. Using actual-time shipping protocol of RTP/RTCP, we are able to assemble a server which could transmit media facts circulation such as H.264 and so forth. It carries three popular technology, which incorporates embedded System, Internet and Multimedia. In this paper, the device has a fault-tolerant and excessive performance. The loss fee of the packet is low, so it is increasingly extensively implemented in the video tracking and network media.

### REFERENCES

- [1] Mokhtarian. K, " Capacity Management of Seed Servers in Peer-to- Peer Streaming Systems With Scalable Video Streams ", IEEE Transactions on Multimedia. vol. 15, no. 1, 2013, pp. 181-194.
- [2] JianzheZhang,YongpingZhang,Yuanqing He , Design of the remote monitoring server as for streaming media digital video, J. Technology Wind, 2011, pp. 45-46.
- [3] Zhou, X.;Xu, C.-Z, " Harmonic proportional bandwidth allocation and scheduling for service differentiation on streaming servers ",IEEE Transactions on Parallel and Distributed Systems .vol. 15, no. 9, 2004, pp. 835-848.
- [4] Kai Chao Yang, Jiashung Wang, "Unequal Error Protection for Streaming Media Based on Rateless Codes", IEEE Transactions on Computers. vol. 61, no. 5, 2012, pp. 666-675.
- [5] Shiann-TsongSheu, Chi-Hao Huang, " Mixed P2P-CDN system for media streaming in mobile environment ", Wireless Communications and Mobile Computing Conference (IWCMC), July 2011, pp. 657-660.
- [6] XiangdongLiu,Kui Liu, Designing qos-based streaming media server, J. Computer Applications And Software, 2011,pp. 26-31.
- [7] WenyanCi, XueliChen,SuhuaCai, "A Buiding of Streaming Player Client Based on ARM Processor", International Conference on Biomedical Engineering and Computer Science, 2010, pp. 881-884.