

## ***Multimedia Sensing based on Internet of Multimedia Things (IoMT) using API's***

***Yogesh S. Jadhav<sup>1</sup>, Anandrao B. Kakade<sup>2</sup>***

*Department of Electronics and Telecommunication Engineering*

*Rajarambapu Institute of Technology,*

*Rajaramnagar, Sangli (M.S.), India*

***Corresponding Authors: yogeshjadhav1957@gmail.com<sup>1</sup>, anandrao.kakade@ritindia.edu<sup>2</sup>***

### ***Abstract***

*Multimedia sensing in IoMT will be implemented as first module. Raspberry pi platform which is based on ARM and which has an inbuilt Broadcom video and graphics core will be used for this module.*

*Multimedia sensing in Internet of multimedia things will be implemented as first module. Raspberry pi platform which is based on ARM and which has an inbuilt Broadcom video and graphics core will be used for this module. The Open MAX IL is an API which provide interface and communication within the components.*

***Keywords: Raspberry pi, OpenMAX IL, GStreamer***

### **INTRODUCTION**

Multimedia sensing in IoMT will be implemented as first module. Raspberry pi platform which is based on ARM and which has an inbuilt Broadcom video and graphics core will be used for this module.

J. Barba et al. [1] have presented the use of OMX layer for the communication between the components. This component provide better multimedia platform for the communication. They also presented an HW version of OpenMAX IL which provide whole information of the OMX IL. HW

implementation is used to increase the output of OMX based multimedia platform.

Mickael Lanoë et al. [2] have introduced the progress of multimedia application based on embedded system. They introduced hardware and software frameworks for this type of growth. This type of framework access more energy and power. They designed the methodology for power and energy consumption of multimedia application which is useful in embedded system. This technology is executed in Gstreamer framework which is running on Linux operating system. Also these designs are used to measure the power consumption during compressed audio and video streams.

David de la Fuente et al. [3] have developed the designs which provide UML/MARTE methodology. They introduced increasing complexity multimedia system. The concept for designing the complex system is to develop electronic system level.

Nimmi S et al. [4] have designed the basic concepts of video streaming using GStreamer through OpenMAX IL GNU Radio platform. The Gstreamer is a framework which provide many types of

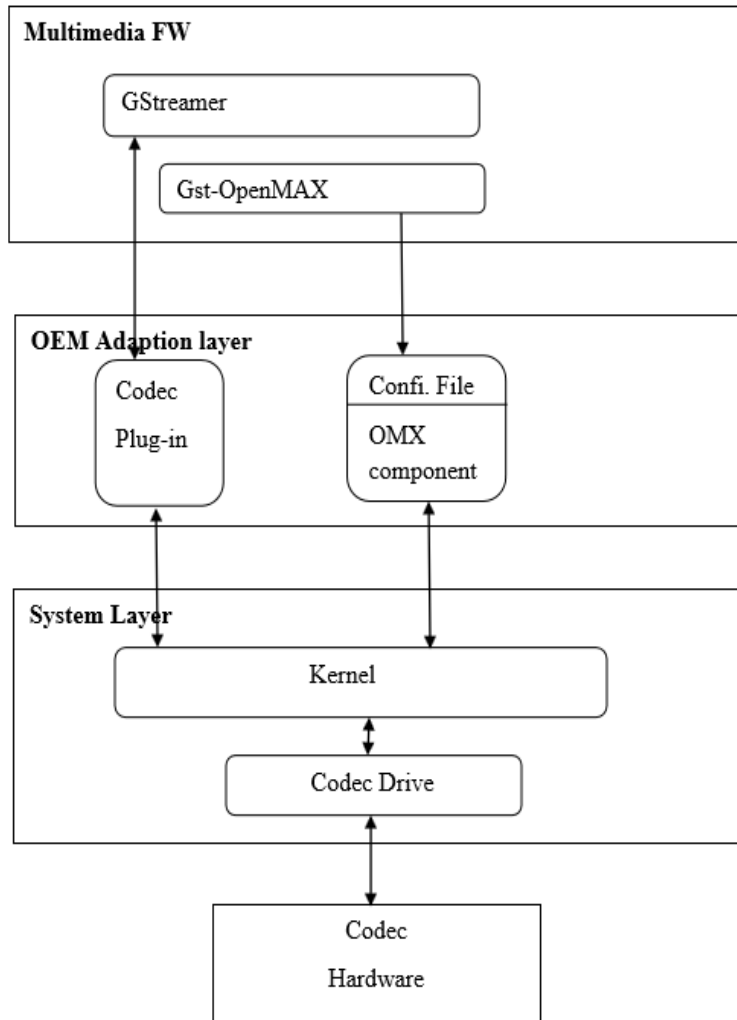
multimedia applications. Gstreamer is a multimedia engine in Linux and other multimedia player. Gstreamer operates in pipeline structure. This structure is designed to generate number of multimedia applications such as video edition, media player etc.

G. Sundari et al. [5] have presented the necessity of different video bit stream techniques. The main focus is a flow of video using Gstreamer. The video is taken by web camera which is interfaced with pc. Then it encode by H.264 encoder. The flow of compressed video is through Gstreamer which gives good compression ratio. H.264 is encoder which compress the video and hence good compression ratio is occurred.

A driver stack would be developed on the video core which will be made Open MAX compliant later. Once the video components are open API compliant, GStreamer plug-in will be develop for the multimedia components. Using the plug-in a small pipeline will be designed which eventually should be driven by a Gstreamer application.

The OpenMAX IL is an API which provide interface and communication within the components. The purpose of the software

interface is to take components with different initialization and commands.



**Fig. 1 Proposed Block Diagram**

Multimedia sensing in IoMT will be implemented as first module. Raspberry pi platform which is based on ARM and which

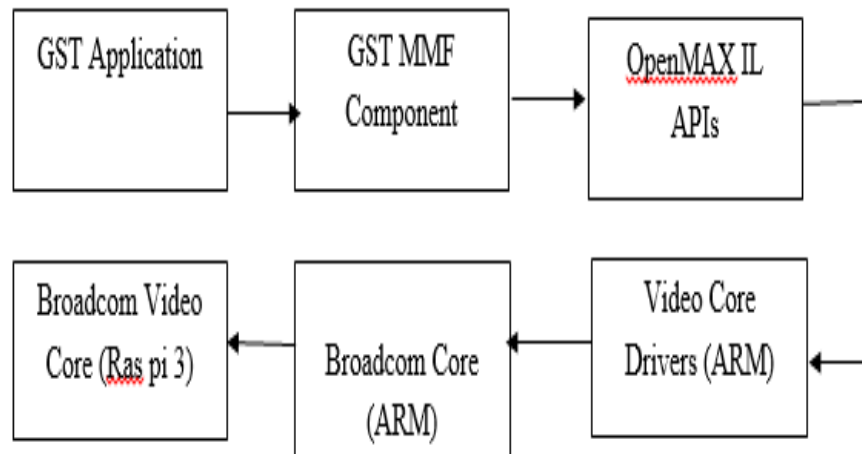
has an inbuilt Broadcom video and graphics core will be used for this module.

Fig. 1 presents proposed block diagram of system, is to encode and decode the sensed raw multimedia data. In video codec the one time video is to be compressed, after that every time it can be decoded to get desired results. It requires complex encoder and simple decoders.

As shown in Fig. 2, a driver stack will be developed on the video core which will make OpenMAX compliant later. Based on how far and well the project goes, the decision of whether to implement video

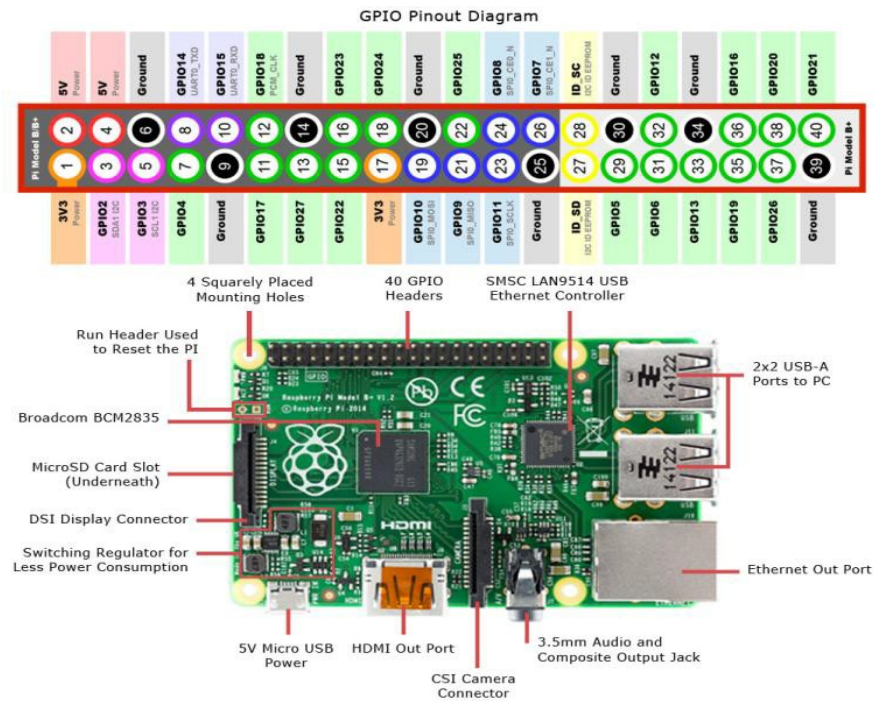
encoder and graphics functionality will be taken. The proposed system will start with the video decoder functionality and make it open API compliant.

Once the video components are open API compliant, GStreamer plug-in will be develop for the multimedia components. Using the plug-in a small pipeline will be designed which eventually should be driven by a GStreamer application.



**Fig. 2 Basic Block Diagram**

**1. Raspberry Pi 3:-**



**Fig. 3 Raspberry Pi 3 Module with GPIO Pin-out Diagram**

Raspberry Pi 3 controller is the latest controller which uses python programming language. Fig. 3 shows Raspberry Pi 3 Module with GPIO Pin-out Diagram. As shown in fig.2, 40 GPIO pins can be used for interfacing RPI 3 with different type of sensors, dc motors, stepper motor etc.

- USB ports-4
- Full size HDMI
- Camera Module V2 can be directly connected using CSI port.
- Touch screen can be directly connected to DSI port.

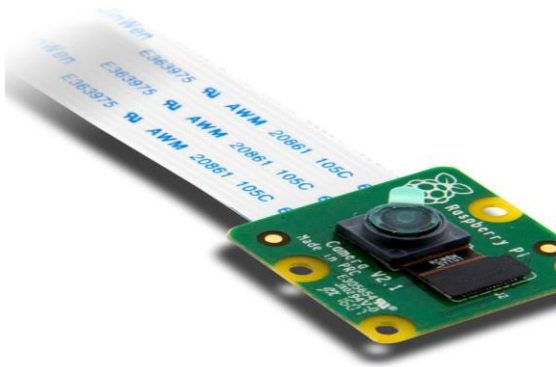
**Feature of RPI 3 Model B:**

- 40 GPIO Pins
- SD card expandable up to 32 GB
- BCM2837Broadcom Processor
- 1.2GHz Quad-Core ARM Cortex-A53
- Inbuilt WIFI and Bluetooth
- RAM-1GB

**2. Camera Module:-**

Fig. 4 shows Raspberry Pi 3 camera Module. 8 megapixel Sony IMX219 Camera module V2 is used to obtain the image of the object and to monitor baby through it. Camera module is used for taking snaps as well as for video capturing. This camera is directly connected to the RPi 3 at camera CSI connector provided on the

board; also we can connect separate webcam using USB port on RPi 3.



**Fig. 4 Raspberry Pi camera module**

**Specification:-**

- Sony IMX219 sensor.
- 8 megapixels captures images of 3280 x 2464 pixels.

- Capture video at 1080p30, 720p60 and 640x480p90 resolutions.
- Good white balance and image quality

**3. Hardware:-**

Fig. 5 shows hardware setup for encoding and decoding of streams. It shows the interfacing of raspberry pi with camera. HDMI cable is connected between screen and raspberry pi module.



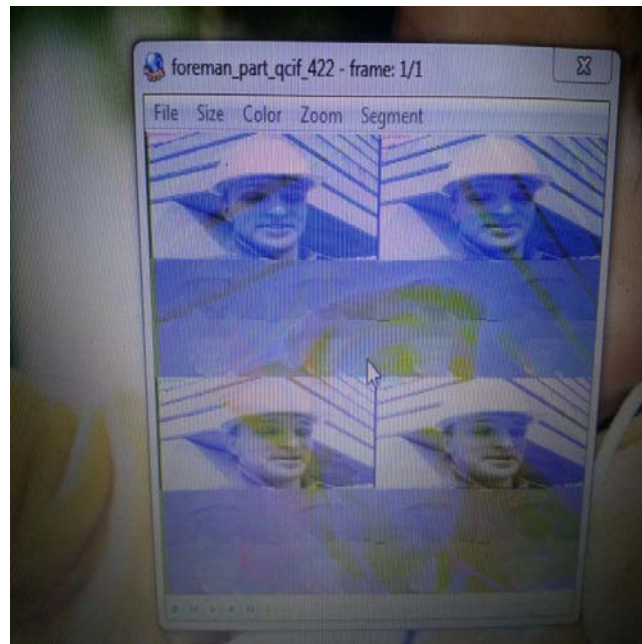
**Fig. 5 Hardware Setup**

**II. RESULTS**

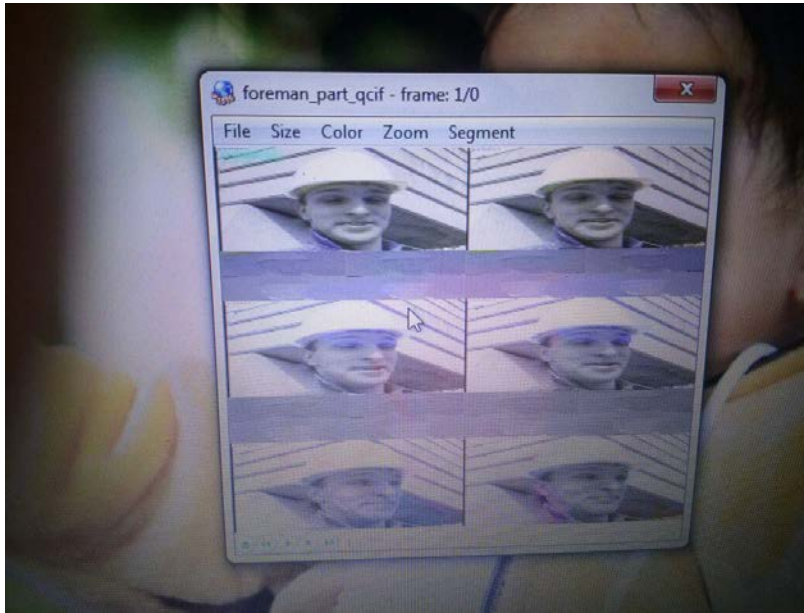
```

pi@raspberrypi: ~/Desktop/tejad/JM/bin
File Edit Tabs Help
-----
Profile IDC : 100
Image Format : 176x144 (176x144)
Color Format : 4:2:0 (8:8:8)
-----
POC must = frame# or field# for SIIRs to be correct
-----
Frame      POC Pic#  QP  SnrY  SnrU  SnrV  Y:U:V Time(ms)
-----
Warning: could not read from reconstructed file
00000(IDR)  0    0   10           4:2:0   44
00002( P )  4    1   15           4:2:0   28
00001( b )  2    2   10           4:2:0   34
-----
Average SNR all frames -----
SIIR Y(dB) : 0.00
SIIR U(dB) : 0.00
SIIR V(dB) : 0.00
Total decoding time : 0.107 sec (28.037 fps)[3 frm/107 ms]
-----
Exit JM 19 (FRExt) decoder, ver 19.0
Output status file      : log.dec
3 frames are decoded.
[Inferior 1 (process 2936) exited normally]
(gdb)
  
```

**Fig. 6 Decoder Output Window**



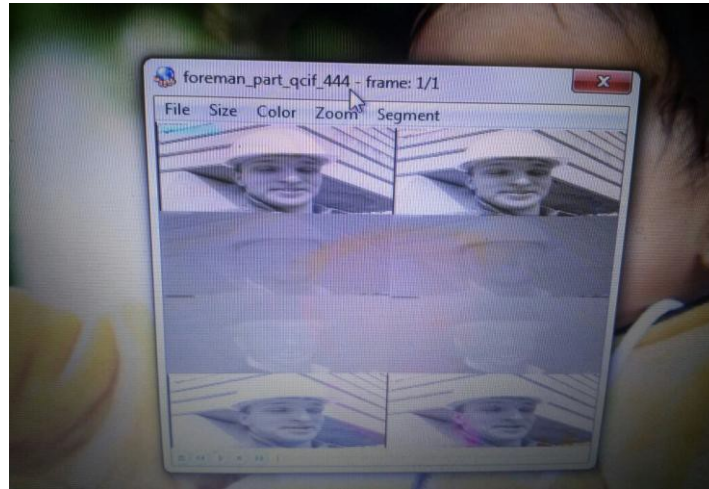
**(a)**



(b)



(c)



(d)

**Fig. 7 Output of Decoder with different streams**

The Fig. 7 shows the output of decoded streams in this streams are decoded by using JM references module. In this streams are available in .h264 format and after decoding streams are converted into YUV format.

## CONCLUSIONS

The Raspberry Pi 3 is successfully used to run JM reference module for encoding and decoding. Different streams with .h264 format are encoded and decoded. Also decoded outputs are played in YUV player. The advantage of decoding is to decompress the data which is present in .h264 format and speed of decoding will also increase.

## REFERENCES

- [1] , “UML/marte methodology for automatic SYSTEMC code generation of OpenMAX multimedia applications”, digital system design, pp.467 – 470, 2013.
- [2] Nimmi S, Saranya V, Gandhiraj R, Green Computing Communication and Electrical Engineering (ICGCCEE), International Conference, pp.1 – 6, 2014.
- [3] G. Sundari, T. Bernatin, “H.264 Encoder using Gstreamer”, International Conference on Circuit4, 2015.