REMOTE VIDEO MONITORING SYSTEM USING RASPBERRY PI 3 BASED ON GPRS MODULE AND PPP PROTOCOL

S. N. Kale[1], Prof. R. D. Patil[2]
ME Electronics Engineering, Associate Professor

Abstract

Video monitoring has played important role now in day to day life for security purpose. With wireless technology we can connect to remote area. Also due to development in embedded technology, it is possible to build inexpensive system for remote area monitoring. This paper presents implementation of remote video monitoring system using Raspberry Pi 3 based on GPRS (General Packet Radio Service) network and PPP protocol. Transmitter hardware takes Raspberry Pi 3 with ARM Cortex-A53 (64 Bit) processor as centralized unit along with 1GB LPDDR2 SDRAM, USB camera and GPRS module for internet connectivity which again reduces overall cost of the system. Application program is used to capture images using USB camera and compressed through libjpeg library for JPEG compression. Captured images are transmitted via SIM900A dual band GSM/GPRS module using PPP protocol to FTP server. This system is supported by Raspbian Linux operating system based on Dabian. Monitoring center downloads image data from FTP server by using curl client utility and displays using MJPG streamer. In this system we can view images as well as can store images sent by transmitter for future use. With the combination of Raspberry Pi and GPRS module, the monitoring system has the advantage of low cost and low power consumption.

Index Terms: **Raspberry Pi, USB Camera, GPRS, PPP, MJPG streamer**

I. INTRODUCTION

Recent years monitoring facilities are required and useful for our daily life security. From small firms to many large companies some kinds of video capturing systems have introduced to keep their security.

Traditional video surveillance system can generally achieve close distance monitoring, by using the computer system as a monitoring unit connected to surveillance camera with coaxial cable. But today in this digital world, the video surveillance systems based on embedded technology are more advantageous compared to the traditional surveillance systems. It provides high performance at low cost and good stability. Meanwhile, it provides many advantages, such as enhanced design simplicity, compact construct, portable, low power consumption, long-distance transmission.

For wide coverage areas and hard-to-wire locations for remote monitoring such as traditional river monitoring systems, it is difficult to construct a system because river environments are not well suitable for wire or fiber transmission lines. Here wireless GPRS technology gives an excellent alternative for this. As it is easy to set up and configure, the cost of operating a GPRS monitoring system can be substantially lower, since the cost is directly proportional to the volume of the data and the frequency with which it is transmitted.

II. SYSTEM ARCHITECTURE

In this system, frames are captured from a USB camera which is connected to Raspberry Pi. Captured frames are compressed into JPEG format. These frames are transferred by the GPRS network under the control of the Raspberry pi to FTP server.

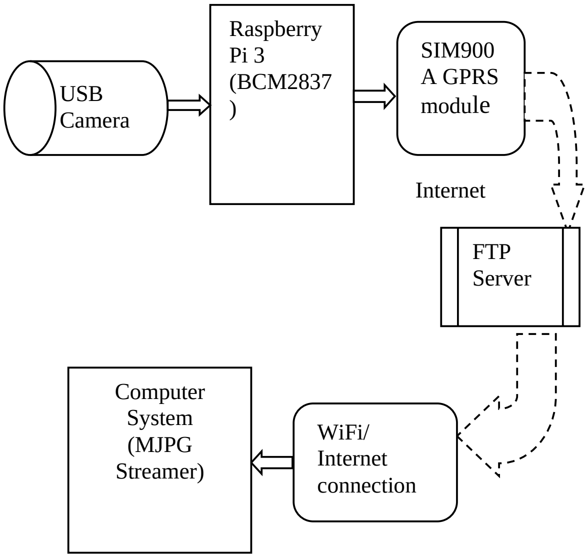

USB Camera
Raspberry Pi 3 (BCM2837)
SIM900 A GPRS module
Internet
FTP Server
Computer System (MJPG Streamer)
WiFi/ Internet connection

Fig 1 Block diagram of overall system

Video frames are downloaded using client utility program such as curl. The monitor client will receive these frames which are streamed using MJPG Streamer. Wireless video monitor system provide a practical solution for remote wireless monitoring with low cost.

III. RASPBERRY PI 3 SYSTEM

In this system we have used RASPBERRY PI 3 model B with1.2GHz Quad-Core ARM Cortex-A53 processor. The raspberry pi is a very powerful, minicomputer with the dimensions of credit card. It Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT. This system is very easy to operate and user friendly as it comes with GUI and command prompt.

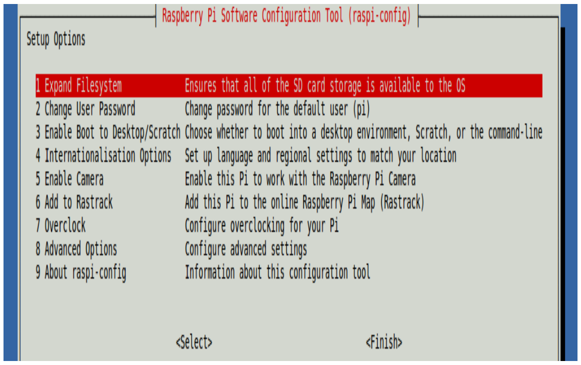
The brain of the Raspberry pi is the Broadcom BCM2837 “system on chip,” which includes the main components needed for a computer system. These include the central processing unit (processor), which handles the main workload; the graphics processing unit (GPU) such as Dual Core VideoCore IV® Multimedia Co-Processor which provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode , which accelerates the process of producing the complicated graphics to see on screen; and the random access memory (RAM) 1GB LPDDR2 memory module which acts as somewhere for the CPU to keep the information that it is working on.

The Raspberry Pi 3 has four USB ports, allowing connect it to keyboards, mice, WiFi dongles, USB camera and USB sticks containing files. This system comprises the general purpose input/output (GPIO) pins. They’re a set of connections that have various functions, but their main one is to allow you to connect to the Raspberry Pi with an electronic circuit. The Raspberry Pi 3 features the same 40-pin general-purpose input-output (GPIO) header as all the Pis going back to the Model B+ and Model A+. Any existing GPIO hardware will work without modification; the only change is a switch to which UART is exposed on the GPIO’s pins, but that’s handled internally by the operating system.

A. Preparation of operating system for Raspberry Pi 3

This device supports Raspbian, a Debian-based Linux distribution, Windows 10 IOT Core, RISC OS, and specialized media center distributions. It
promotes Python and Scratch as the main programming language, with support for many other languages.

Raspbian operating system is selected to control functioning of Pi. First it is required to prepare SD card with operating system which is installed using WIN32DISKIMAGER utility in Windows operating system. After preparation, pi is booted using SD card and configured by using raspi-config configuration menu.



```
Raspberry Pi Software Configuration Tool (raspi-config)
Setup Options

1 Expand Filesystem              Ensures that all of the SD card storage is available to the OS
2 Change User Password           Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment, Scratch, or the command-line
4 Internationalisation Options   Set up language and regional settings to match your location
5 Enable Camera                  Enable this Pi to work with the Raspberry Pi Camera
6 Add to Rastrack                Add this Pi to the online Raspberry Pi Map (Rastrack)
7 Overclock                      Configure overclocking for your Pi
8 Advanced Options               Configure advanced settings
9 About raspi-config             Information about this configuration tool

              <Select>                      <Finish>
```

Fig 2 configuration tool window

EXPAND FILE SYSTEM option is enabled so that all of the SD card storage is available to the OS. Password can be changed using CHANGE USER PASSWORD option. With ENABLE BOOT TO DESKTOP/SCRATCH, booting into a desktop environment, Scratch, or the command line is selected. INTERNATIONALISATION Options sets up the language and regional settings to match location. In advanced option

SERIAL option is enabling for shell and kernel messages on the serial connection.

IV. APPLICATION PROGRAM DESIGN FOR FRAME CAPTURE AND COMPRESSION

To capture frames an application program is written in C language which is based on Video For Linux Two API. V4L2 drivers are implemented as kernel modules, loaded manually by the system administrator or automatically when a device is first opened. The driver modules plug into the "videodev" kernel module. Driver supports video capturing through /dev/vide0. To open and close V4L2 devices application program uses the open() and close() function, respectively.

A. Open the video equipment: The device name of USB camera in Linux is /dev/video0. The Linux kernel supports UVC (USB video device class) based cameras. UVC is a USB device class that describes devices capable of streaming video like webcams, digital camcorders, transponders, analog video converters, television tuners, and still-image cameras. Logitech c 170, 5 Mp web camera is connected to Pi which is UVC based so that there is no requirement for drivers needed to capture images from the camera.

B. Initialization of video device: Devices are programmed using the ioctl() function. The VIDIOC_QUERYCAP ioctl is available to check if the kernel device is compatible with specification given by V4L2 API, and to query the functions and I/O methods supported by the device. Applications can use the VIDIOC_G_CROP and VIDIOC_S_CROP ioctls to get and set top left corner, width and height of image. The v4l2_pix_format structure defines the format and layout of an image in memory. Image formats are negotiated with the VIDIOC_S_FMT ioctl. YUYV image format is selected using V4L2_PIX_FMT_YUYV function.

C. Frame capture: memory mapped streaming I/O method is used to map buffers in device memory into the application’s address space. To allocate device buffers applications call the VIDIOC_REQBUFS ioctl with the desired number of buffers and buffer type.

D. YUV to RGB conversion: captured frames are converted into RGB format. YUV format are used for television broadcast however many displays (almost all DVI inputs) only except RGB.

E. JPEG compression: The JPEG compression encoding scheme is used in the this system which is based on the DCT (Discrete Cosine Transform) sequence. The JPEG encoding under Linux can use the library Libjpeg, which is used under linux standard library, and its function is to compress the picture in accordance with a certain percentage of compressed into JPEG format pictures. Most of the software operating on JPEG are based on this library. In this system libjpeg8 library package is installed using sudo apt-get install libjpeg8-dev on command line.

V. PROGRAM DESIGN FOR NETWORKING

A. sim900A GSM/GPRS module

GSM/GPRS Modem-RS232 is built with Dual Band GSM/GPRS engine- SIM900A, works on frequencies 900/ 1800 MHz. It features GPRS multi-slot class 10/class 8(optional) and support GPRS coding schemes CS-1,CS-2, CS-3,CS-4.The Modem is coming with RS232 interface, which allows connect PC as well as microcontroller with RS232 Chip(MAX232). The baud rate is configurable from 9600-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface. There is GPRS data transfer (PPP or TCP or UDP) in progress. In this case, power consumption is related with network settings (e.g. power control level), uplink / downlink data rates and GPRS configuration (e.g. used multi-slot settings).The onboard Regulated Power supply allows to connect wide range unregulated power supply. Using this modem can make audio calls, SMS, Read SMS; attend the incoming calls and internet through simple AT commands.

The modem needed only 3 wires (Tx, Rx, GND) except Power supply to interface with microcontroller/Host PC. The built in Low Dropout Linear voltage regulator allows to connect wide range of unregulated power supply (4.2V -13V).

Raspberry Pi is connected to the internet through a GPRS cellular data connection using GPRS module. The Raspberry Pi serial port consists of two signals (a 'transmit' signal, TxD and a 'receive' signal RxD) made available on the GPIO header. To connect to another serial device, the 'transmit' of pi is connected to the 'receive' of the GPRS module, and vice versa. The Ground pins of the two devices should be connected together. On the Raspberry Pi 3 the second serial port is called /dev/ttyS0 and is by default mapped to the GPIO pins 14 and 15.

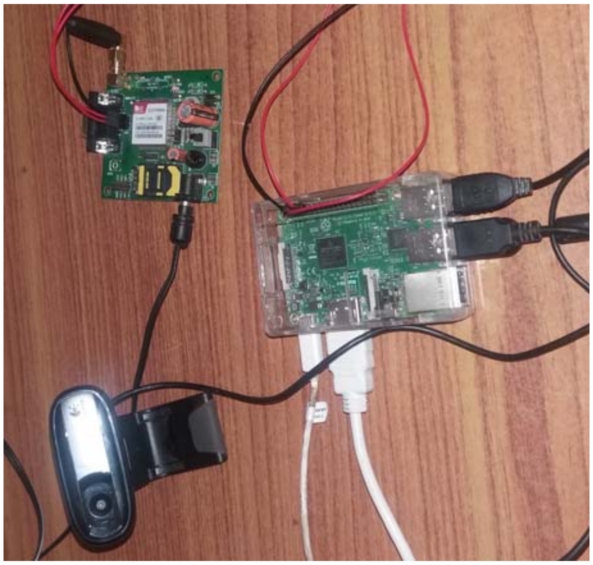


Fig 3 Interfacing of USB camera and GPRS module to Raspberry Pi

A. Access of the Internet using GSM / GPRS Modem (SIM900A) and PPP protocol

PPP is most commonly used data link protocol. It is used to connect the Home PC to the server of ISP via a modem.

PPP is the protocol used for establishing internet links over dial-up modems, DSL connections, and many other types of point-to-point links. The pppd daemon works together with the kernel PPP driver to establish and maintain a PPP link with another system (called the peer) and to negotiate Internet Protocol (IP) addresses for each end of the link. Pppd can also authenticate the peer and/or supply authentication information to the peer. PPP can be used with other network protocols besides IP.

Chat is a program that can perform simple handshaking between a PPP client and server during connection setup, such as exchange usernames and passwords. Chat is also responsible for causing your modem to dial the ISP's phone number and other simple tasks.

Chat itself is automatically invoked by pppd when started. A simple shell script invokes chat to handle the negotiation. Script file should be placed in the */etc/ppp* (as root).

Required and previlaged options are defined in script file which is placed in /etc/ppp/peers directory

B. Uploading images to FTP server

Captured images with number of frames set by user are uploaded through FTP client utility to FTP server. Shell script is used to connect, upload images to FTP server.

C. Multithreading in Application program

In most modern operating systems it is possible for an application to split into many "threads" that all execute concurrently. In order to take full advantage of the capabilities provided by threads, a standardized programming interface was required.

For UNIX systems, this interface has been specified by the IEEE POSIX 1003.1c standard (1995).

Implementations adhering to this standard are referred to as POSIX threads, or Pthreads. Most hardware vendors now offer Pthreads in addition to their proprietary API's.

The POSIX thread libraries are a standards based thread API for C/C++. It is most effective on multi-processor or multi-core systems where the process flow can be scheduled to run on another processor thus gaining speed through parallel or distributed processing. Threads require less overhead than "forking" or spawning a new process because the system does not initialize a new system virtual memory space and environment for the process.

As Raspbian is Linux based dabian package, it also support multithreading via POSIX thread libraries. The function ‘pthread_create () can start execute a function as a separate thread within the main process which calls the ‘pthread_create (). The prototype of the function ‘pthread_create () is defined in the header file <pthread.h>. Thread synchronization is achieved by using the function pthread_join( ).

VI. MONITORING AT RECEIVER

At receiver side we have computer system with Linux based Ubuntu operating system which is used to receive and display frames continuously by using application programs. Frames are received using internet via FTP client such as curl and frames are displayed using MJPG

Streamer program**.** Internet connectivity is provided via Wi-Fi or 3G dongle.

A. MjpgStreamer

It is a simple utility to stream images or video via HTTP. MJPG-streamer", is a command line application that copied JPG-frame from a single input plug-in to multiple outputs plug-in. It can be used to stream JPEG files over an IP-based network from the webcam to a viewer like Firefox, and Video lan client or even to a Windows Mobile device running the TCPMP-Player. It uses hardware compression in order to reduce CPU cycles of the server. It is light and utilize less CPU hence ideal for embedded devices and regular servers. It recovers JPG images from UVC-compatible webcams, system files or other input plugins and distributes them as M-JPEG via HTTP to browsers, VLC or others. We can use input_file plugin to collect images from folder, ouput_http plugin to stream files and output plugin to store JPEG images of the input plug-in in a specified folder.

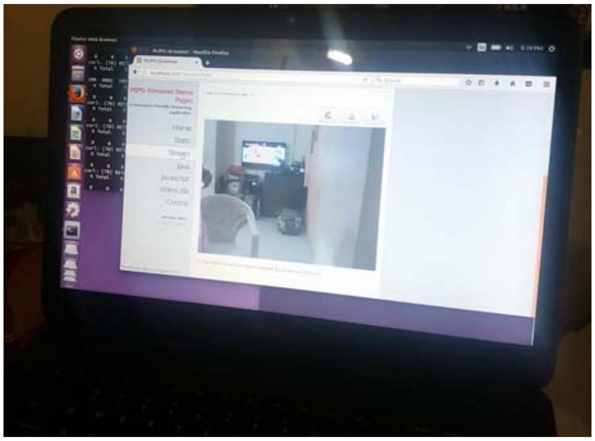


Fig 4 streaming of video at receiver side using MJPG Streamer

VII. CONCLUSION

With this system jpeg images are captured with size 320x240 and compressed by 15-20% of original image. Due to multithreading it is possible to capture images and connect to server simultaneously which increases performance of the system. After compression, images are uploaded to FTP server in 5-7 seconds. This may vary according to strength of network signal. This system is a good solution in areas where continuous streaming is not required. We can reduce cost as it doesn’t require static IP and also GPRS system require less bandwidth as compared to 3g and next upgraded version network. As images are available to receiver side, we can use those images for future use.

REFERENCES

[1] Lihui Zhao,Chunbao Huo, Hongzhe Yang, “The Design of Remote Video Monitoring System Based on S3C2416 and GPRS,” IEEE Trans. on System Science and Engineering, , pp. 385-388, July 2012.

[2] Denan Li, Zhiyun Xiao,, “Design of Embedded Video Capture System Based on ARM9,” IEEE , 978-1-4244-8039-5/11, 2011.

[3] Nagaraja G. and Sharada P.N., “ Design of Remote Security System Using Embedded Linux Based Video Streaming” (IJCAR) ISSN 2305-9184 Volume 2, Number 2 (April 2013), pp. 50-56.

[4] Ruturaj Shete, Mayuri Sabale , “VIDEO SURVEILLANCE USING RASPBERRY PI ARCHITECTURE” Discovery ANALYSIS The International Daily journal ,ISSN 2278 – 5469,pp 37-43,Nov 2015

[5] Sanjana Prasad1, P.Mahalakshmi2, A.John Clement Sunder3, R.Swathi4, “Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, pp 7107-7109