

Remote Lab Implementation on an Embedded Web Server

Alexander P J,
PG student,
EEE Dept, College of Engineering,
Guindy, Anna University,
Chennai, Tamil Nadu, India
alexanderpj.ceg@gmail.com ,

N. Radhakrishnan
Assistant Professor
EEE Dept, College of Engineering,
Guindy, Anna University,
Chennai, Tamil Nadu, India
rk1905@annauniv.edu

Abstract— The design, implementation and evaluation in setting up a remote laboratory for programming microcontrollers on an embedded web server residing over an ARM (Advanced RISC Machine) microprocessor based embedded system is proposed in this paper. The hex files or c codes generated with the help of various client machine IDE's (Integrated Development Environment) can be transferred to the server through the TCP (Transmission Control Protocol) port 80 which in turn programs the microcontroller attached to beaglebone with help of various software tools like avrdude, mspware etc. The system is designed by setting up an Apache web-server over the Beaglebone Black single board computer. Client machines can access the server through the static IP address assigned. The evaluation of the remote lab is done by programming Atmel AT89s52 microcontroller remotely and by performing a webcam interface to beaglebone thereby achieving video streaming, the real time video result can be viewed at the client end.

Keywords—ARM, Beagleboneblack, TCP, avrdude, mspware, AT89s52

I. INTRODUCTION

Because of the rapid development and expansion in the field of internet and microprocessors the application performed by those to the end user also kept on increasing and demanding. And with the advent of Internet of Things (IoT) the task of monitoring and other telemetric activities like remotely programming or re-programming based on the environment conditions pertaining to the remote location became challenging. This paper discusses and designs on such issues of remotely programming microcontrollers, where a prototype of remote lab is setup with the help of beaglebone black which can program any microcontrollers attached to it if it is specified and setup properly.

This paper also comes up with a design solution of an embedded web-based remote monitoring system for the environment in the laboratories, which realizes the local management and remote publishing applications for large-scale dynamic data of sensor networks and video images[2]

The design proposed in this paper is expected to answer the issues often raised in the implementation of the remote lab. The remote labs find much application in

industrial [3] and educational institutions[4]. Its advantages over conventional PC servers are explained below

- Implementation on Embedded Web Server for the remote lab will provide advantages like cost effectiveness because the required hardware is cheaper when compared to using a PC server and software is built on open source utilities and API's, the power required is very optimum since the system is running on minimum requirements and the dimension of equipment are also smaller providing easy installation and maintenance[6].
- Because the Embedded Web Server working with pure HTML and PHP instructions that opened the port is port 80 only, so that the firewall is safe.
- Remote labs can be multiuser, so that at an instant multiple users can log onto to the system and through proper context switching all can access the system resources depending on processing speed of processor and the number of experimental module installed.
- Other telemetric activities like remotely programming or re-programming based on the environment conditions pertaining to the remote location became challenging. This paper discusses and designs on such issues of remotely programming microcontrollers.

II. GENERAL ARCHITECTURE

The central processing unit is a beaglebone black board consisting of ARM cortex A-8 processor[8]. The general framework of remote lab can be classified into two. The remote monitoring system consisting of various sensors whose calibrated values can be published on the server and the expansion board which can be programmed remotely which is the Arduino uno in this design

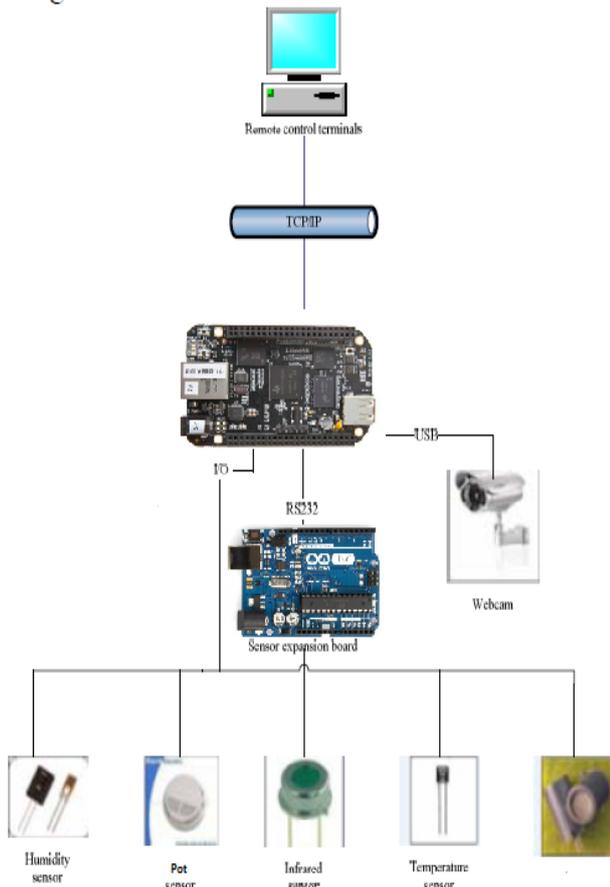


Figure 1. General Architecture of Remote Lab

A usb webcam is also attached to see the real-time visuals in cases where the serial expansion board programming is done for some DC-motor control applications or some relay systems.

III. DESIGN OF SYSTEM HARDWARE

A. Beaglebone Black

The AM335x 1GHz ARM Cortex-A8 is the hard core of beaglebone black. It has a 512MB DDR3 RAM (Dual Data Rate Random Access Memory) with 2GB 8-bit extended Multi Media Controller (eMMC) on-board flash storage. For high performance graphics operations it is having a 3D NEON graphics accelerator which computes all the floating point calculations[8]. It has Ethernet HDMI ports. This feature of Ethernet connectivity is extracted for this project.

ARM Servers are the new entrants into a server market dominated by the x86 architecture. The goal of the Ubuntu ARM Server project is to match the server and workload functionality currently enjoyed by x86 servers on the ARM architecture with optimal performance.

B. Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328 microcontroller. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller by simply connecting it to a computer with a USB cable or powering it with a AC-to-DC adapter or battery, the bootloader will call to the starting location of the flash memory where the program is residing and the program gets executed.

IV. DESIGN OF SYSTEM SOFTWARE

The beaglebone is loaded with a Ubuntu OS (Operating System) Ubuntu is a debian (derivative) based linux operating system. Ubuntu server pack is installed into the board since its fast, secure, robust, light weight, open source that uses the same APT repositories as the Ubuntu Desktop Edition.

Apache web server is installed over it. Apache supports a variety of features, many implemented as compiled modules which extend the core functionalities ranging from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP. Sensing the sensor values and passing it to the system calls is performed with the python shell scripting and the server side scripting is performed with PHP (Personal Home Page) scripting

A. Remote Programming System

The host can upload the c files to be compiled and bud into the flash memory of the microcontroller. Linux AVR Tool chain is used for programming the attached microcontroller which includes

- 1) A text editor
- 2) GNU C compiler for AVR microcontroller family (avr-gcc) and libraries to generate executable file
- 3) A utility for converting the executable file to hex codes
- 4) A programming tool such as AVRDUDE to dump the hex code to the flash memory of the microcontroller

The avr-gcc (GNU c compiler for avr microcontrollers) that has been installed over the Ubuntu OS residing on the beaglebone compiles the file and returns a machine dependable executable code which can be run from the shell. The avr-gcc also performs post build operations like optimizing the size of the executable file to suite the processor and setting up of fuses of the EEPROM of the processor. We can resort the avr-objcopy tool is used to convert this executable file into machine dependable .hex code since only .hex files can be written into the flash memory of the microcontroller.

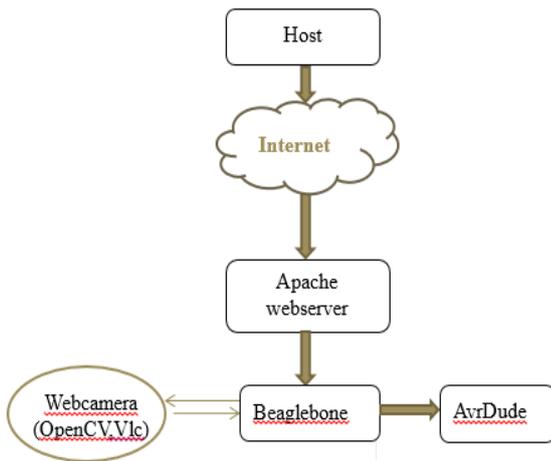


Figure 2. Remote programming system overview

Burning the .hex codes into the flash memory of the microcontroller can be performed through the avrdude tool called either through the system call from remote terminal or through the php script running on the local device. The avr-dude provides option for determining the baud rate for writing, bit rate if we are using Joint Test Action Group (JTAG) emulators for emulation purposes, erasing the flash memory before programming in-order to remove the garbage bits set by the previous program run on the microcontroller, bit stuffing the unallocated areas of memory, run time calibration, memory type etc. Once flashed avrdude raises the interrupt which can be used to trigger the video streaming commands

B. Remote Monitoring System

- Main login page

The main login page is used for accepting the user credentials and verifying with the database within the server thereby authenticating that a valid user is using the remote lab. If a match is found the user is directed to the next page else the user is given with proper warning and sent back to the login page. This operation will be carried out in a loop until a match is found

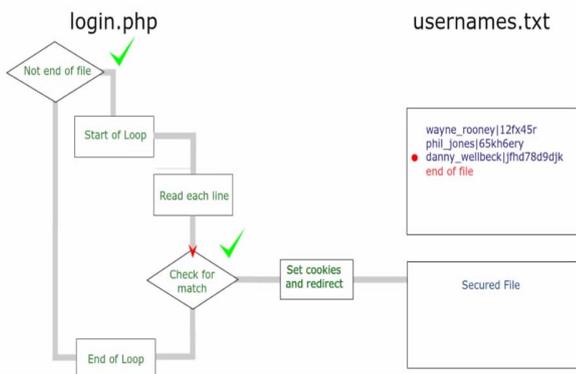


Figure 3. User authentication flow chart

- Publishing Monitoring Data on Web page

The flow chart below explains the execution flow of the monitoring system. The login page is created with PHP the login id and password will be compared with the system database and only if same the main page is displayed. Later the sensing values obtained from various sensors connected to beagle bone are captured using the python shell scripts and the files containing the details are sent given to the server and the results are displayed on the main page and video streaming is carried out by triggering the webcam including all its associates drivers and streaming is carried out through the TCP port 8080

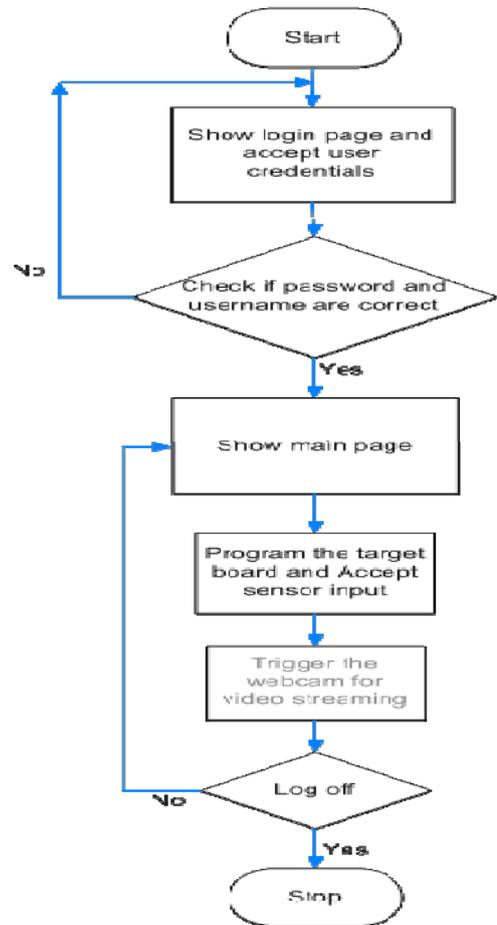


Figure 4. Execution flow chart of monitoring system

- Video streaming using embedded linux

In publishing the images or streaming a video on webpage the process can be split into three.

1. Image acquisition,
2. Image compression
3. Dynamically displays

Image acquisition module completes the capture of the video images through the video device driver V4L (video4linux) in the embedded Linux kernel. V4L is the basis of imaging systems of Linux system and provides a range of application programming interface function for the control of the video devices[7].

The stream of image data acquired in the memory is saved as files and is compressed for transmission through the network. The compression defines whether to compress as moving picture format (for video) or single image (in lossy or lossless formats). This design compress raw data in the MJPEG algorithm[7] (video) to generate data stream. The compressed images can be passed through various image processing algorithms. Opencv (Open Computer Vision) can be used for writing these algorithms. The opencv binary executable provided along with the image processing libraries can be used optimizing the algorithm based on the target machine being used. Finally, results are published in web.

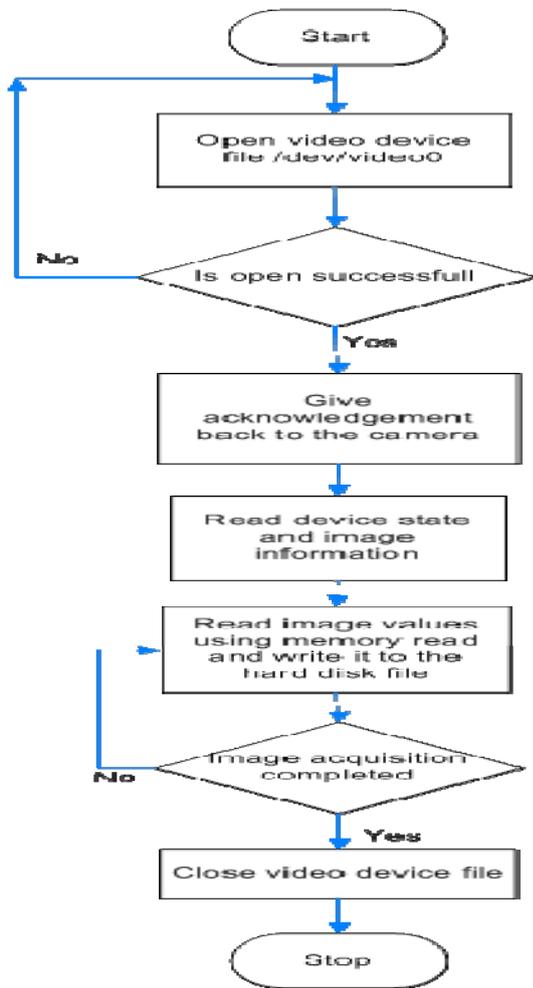


Figure 5. Flow chart of video capturing

V. EXPERIMENTAL RESULTS

A. Remote laboratory home page

A static IP address is assigned to beaglebone when logged into the server from a remote system the webpage obtained contains option to upload hex file and submitting it. On submission the background the PHP script is being run for executing system commands for compiling the code and triggering avrdude

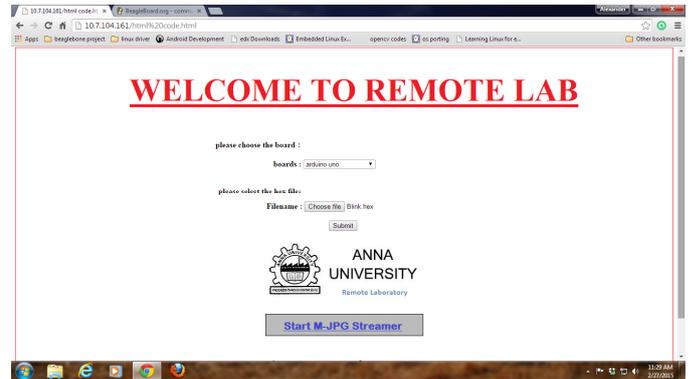


Figure 6. Remote lab home page

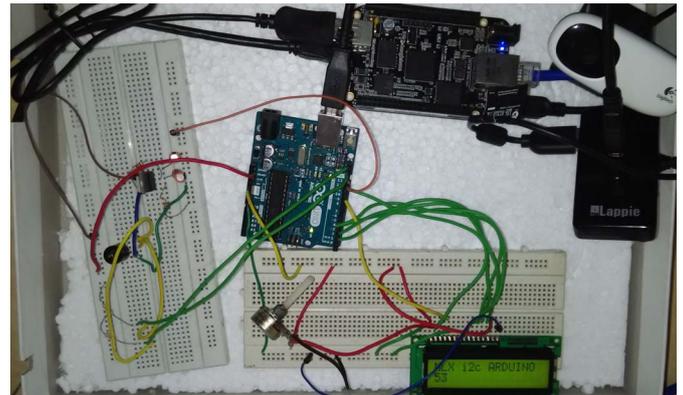


Figure 7. Experimental setup for remote programming

B. Video streaming using MJPG streaer

The edited snapshot of the video being captured by the webcam at usb0 and streamed using MJPG streaer[7] is displayed below. Video resolution of 720x480 is obtained and static images from the video can also be obtained.

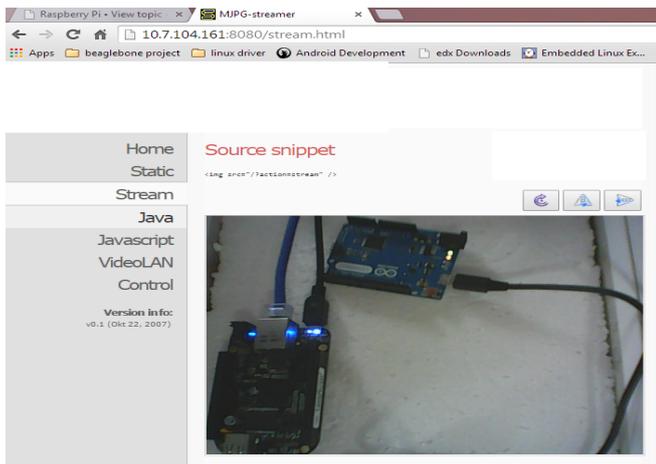


Figure 8. Video streaming

VI. CONCLUSION AND FUTURE WORK

The remote laboratory concept is not new, but this paper focusses on implementing the same on an embedded web server thereby reducing the implementation cost, power consumption, boot-up and runtime, expanding the opportunities for students to work on target boards and also by designing an embedded web-based remote monitoring system to realize the local management and remote publishing applications for large scale dynamic data of sensor networks and video images[2]. The experimental results show that the system designed realizes safe and convenient remote monitoring and local management of the environment in laboratories and has high availability, reliability and popularization[1].

Security and real time response are the two other key features for remote lab which are beyond the scope of this paper so as part of future work performing MD5 algorithm[6] for user authentication and file transfer can improve security and implementing the whole system on a real time operating system can be challenging. Optimizing the OpenCV codes for various vision based algorithms for a particular DSP processor can also be performed as future work of this paper.

C. Image processing being performed with OpenCV

OpenCV is much faster than the Matlab executable since the executable code size is much smaller and that the codes are written in C++ language and linked using the OpenCV shared object libraries. OpenCV also added support for graphics processing unit (GPU) residing on beaglebone and gives optimized codes by performing operations using the NEON floating point unit of the ARM cortex A8 system on chip (SOC). OpenCv is faster than Matlab in some algorithm from 4 to 30 times and in case of Erosion algorithm up to 100 times[5]. A result of a RGB image to gray image conversion performed using opencv in beaglebone is shown below



Figure 9. RGB to gray conversion

References

- [1] Bose, R., "Virtual Labs Project: A Paradigm Shift in Internet-Based Remote Experimentation," Access, IEEE , vol.1, no., pp.718,725, 2013
- [2] Liu Yang; Linying Jiang; Kun Yue; Heming Pang, "Design and Implementation of the Lab Remote Monitoring System Based on Embedded Web Technology," Information Technology and Applications (IFITA), 2010 International Forum on , vol.2, no., pp.172,175, 16-18 July 2010
- [3] Pradarelli, B.; Latorre, L.; Flottes, M.-L.; Bertrand, Y.; Nouet, P., "Remote Labs for Industrial IC Testing," Learning Technologies, IEEE Transactions on , vol.2, no.4, pp.304,311, Oct.-Dec. 2009
- [4] Kulkarni, Anup; Jhunjunwala, Ashok, "Development of a remote lab with micro-processor & DSP with FPGA accelerator," Communications (NCC), 2013 National Conference on , vol., no., pp.1,5, 15-17 Feb. 2013
- [5] Matuska, S.; Hudec, R.; Benco, M., "The comparison of CPU time consumption for image processing algorithm in Matlab and OpenCV," ELEKTRO, 2012 , vol., no., pp.75,78, 21-22 May 2012
- [6] Shuang Wang; Huiyang Zhao, "Design and implementation of remote virtual network lab," Systems and Informatics (ICSAI), 2012 International Conference on , vol., no., pp.1005,1007, 19-20 May 2012
- [7] Yeoh, K.J.M.; Hwee Ling Wong, "Web-based remote navigational robot for multiclass human-robot interaction," Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2012
- [8] <http://linux.org/BeagleBoneBlack>