RESEARCH ARTICLE OPEN ACCESS

Design and Development of Remote Monitoring and Controlling System Based on Embedded Web Technology

Pooja Kullarkar*, Kanchan Dhote**
*(Department of Electronics & Communication Engineering, RTM University, Nagpur, India
Email: pooja.kullarkar@gmail.com)
**(Department of Electronics & Communication Engineering, RTM University, Nagpur, India
Email: kanchan.dhote@rediffmail.com.)

ABSTRACT
The application of Remote Monitoring and Controlling Systems is used to implement a monitoring and controlling system using ARM Intelligent Monitoring and Controlling Center which uses Samsung's processor (S3C6410) as its major controller. The environmental circumstances present within laboratory can be monitored using sensors and other devices are connected to sensor controlling board. We are distributing predefined standards from sensor board to ARM11 board through RS-232 cable which is linked to UART port of ARM11 board. When a person is came inside lab, the person's image can be captured by webcam and transfer it to controller which transmits data to remote PC through Ethernet via FTP. Once data is to be found at server, we can analysis data at remote PC on web page with unique IP address. User can observed continuous streaming of video with sensor's data. If we want to control devices based on sensor's data is possible through web page from remote location using HTTP protocol which always requests the server for controlling devices. We are focusing on difficult points of developing GUI applications based on Qt/Embedded and Linux drivers for various types of sensors for multiple purpose labs.
Keywords: - FTP, HTTP, QT/Embedded, RS-232 cable, UART.

I. INTRODUCTION

Monitoring and controlling systems from remote locations has been increasing in day to day life which makes easy to control and monitor condition from any place at any time. The embedded systems uses sensor controlling board in which 8-bit microcontroller as the main controller has been widely used in different fields, other than most of these applications in the low-level stage of stand-alone use of the embedded system. S3C6410 consists of ARM 11 processor, several media and graphic co-processors and various peripheral IPs. It is practicable to be relevant the high performance 32-bit microprocessors such as S3C6410, embedded Linux system and Qt/embedded GUI application to practical industrial control in certain occasion.

At present the management of the Domestic laboratories in the research institutes has issues of deprived real time, high cost and low accuracy. It is complex to identify the quality of the environment of the laboratory. So the Intelligent Monitoring and Controlling System should be developed for giving early warning, remote control, and additional functions. This paper focuses on the process and complex points in the application of embedded GUI based on Qt / Embedded and Linux device driver in the laboratory environment intelligent monitoring system.

Typically, programming, digital logic design and often a computer architecture course are prerequisites for the more advanced embedded systems or microprocessor design course that is the focus of this paper. For software development in the embedded systems industry, the C/C++ family of languages is still used in the large majority of new designs, according to annual industry surveys. Many embedded systems, microcontroller, or microprocessor design courses started out with low-cost 32-bit processors with limited capabilities, but most of the development effort in industry has moved on to modern System on-a-Chip (SOC) 32-bit devices that contain a reduced instruction set computer (RISC) processor with volatile memory, non-volatile flash memory, and a wide assortment of standard I/O interfaces, all on a single chip. According to annual industry surveys of embedded designers, 70% of new designs now utilize an operating system (OS), and 59% include networking. The widespread development of these new embedded devices with networking. Now that a single-chip microcontroller already contains the processor, memory, and numerous I/O interfaces with built in hardware controllers, it is appropriate to use a higher level of abstraction in such a course. An increased focus can be placed networking. The embedded C/C++ application programming interface (API) libraries are used to enhance the output, basic operating system concepts and rapid prototyping of devices. This paper describes the experience gained developing a laboratory to support development of these devices; it will primarily focus on the new technologies used in the student instructional laboratories during the first three offerings of the new course.

II. SYSTEM TOPOLOGY

The system is divided into two parts which are ARM Intelligent Monitoring Center and Sensor Controlling Centre. The peripherals are connected with it.

III. OVER ALL SYSTEM PLAN

The total system plan is showed in figure1.

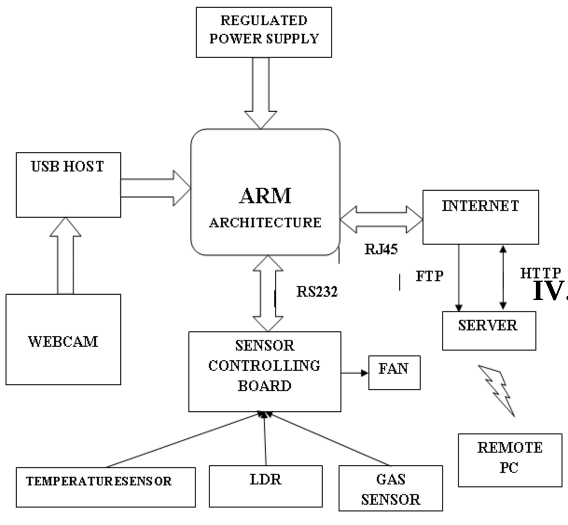

Figure1: Total System Block Diagram

The system is mainly made with ARM controller unit. The design in this paper applies S3C6410s 32-bit ARM microprocessor which takes ARM11 as its core which consist of only 66 microprocessor. This microprocessor has rich resources, including, UART, USB, NAND and NOR Flash, Clock, SDRAM, LCD, Ethernet Interface, RS232 Interface, JTAG, Regulated power supply, etc. These modules can help achieve Ethernet services along with Zigbee technology. This system consists of sensor controlling board using one or more IC 8051 is connected with ARM 11 development board. The Zigbee sensor technology is used at both boards for transmission and receiving process. ARM 11 development board works as computer or controller having the speed up to 533 to 750 MHz S3C6410X consists of ARM 11 processor, a number of media and graphic co-processors and various external IPs. ARM11 processor is connected to several memory controllers through 64-bit AXI-bus. IC 8051 is having speed up to 12MHz and is externally connected to ADC but don't handle USB camera. This camera is connected to ARM 11 that has two USB port. In RMCS project, we implement only one USB hub that connects multiple USB which requires USB extension box. This requires one hub and one USB camera for making prototype.

Embedded Linux operating system and boa web server run on the main controller to manage various types of equipments including sensors, USB camera. The sensors mainly used in our system are atmospheric conditions sensing sensors like Temperature, Gas and LDR sensors. USB camera monitors all the conditions inside Lab. Communication from sensor board to ARM board is done using UART through RS-232 cable. These monitored conditions are viewed at PC on web page by using FTP protocol by providing specific IP address. To control devices from web page we use HTTP protocol and link is established through public network i.e., internet network.

IV. SOFTWARE DESIGN OF ARM HOST CONTROLLER

4.1. Transplantation of Linux Operating System

The transplant of Linux operating system is related with the hardware. Its modification prefers to the Linux operating system according to the concrete hardware platform to make it running on this hardware platform very well. The system's kernel edition is 2.6.32. The Linux operating system's transplant needs to complete three works: boot loader transplant, Linux kernel transplant and root filing system transplant. Boot loader is a small program which loads operating system kernel into memory and transfer control to it. The mainly role is initializing hardware equipment(including I/O, the special function register), establishing the memory space map and bringing the environment of the system's hardware and software to an appropriate state. The Linux operating system's kernel is able to provide good support to the ARM processor and manage most of components which connect to the periphery of the controller. The embedded Linux kernel only requires providing support to the hardware which will be used; therefore we may cut the kernel according to the practical application.

4.2. Application design

The overall software structure of the ARM main controller is shown in Figure 2. The system adopts Linux as the operating system of the ARM's main controller. The work needed to be done is: the porting of Linux in ARM11 board, the programming of the serial driver is used to read values from serial port and display the values on GUI window. The programming of touch screen driver is used for controlling the devices. The implementation of Web server is for monitoring and controlling from remote location. The system adopts of the graphical user interface based on QT/E and establishes a QT user interface to optimize the human-computer interaction environment.

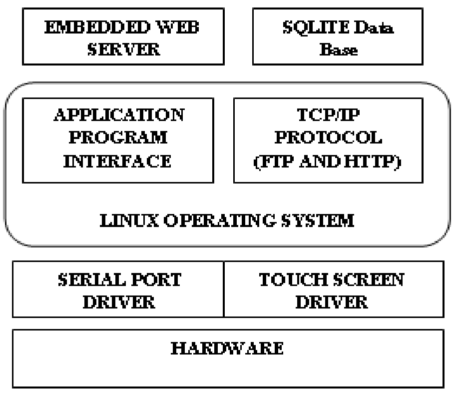

Figure 2

4.3. Local Data Management & SQLite and Embedded GUI

The local data management for sensor networks includes three parts which are data storage, data analysis & data display and Storage of Sensor Data. The ARM Intelligent Monitoring Center acquires sensor data via the sensor expansion board. Sensor expansion board with AT89S52 controller as its master chip is used for sensor and communicating with intelligent monitoring center. SCM software is the task of tracking and controlling changes in software include revision control and establishing baselines that used to obtain the status of each sensor and monitoring local data. It is able to send out an interrupt request signal. ARM intelligent monitoring center send control commands to SCM according to the definition of serial communication protocol, which returns the data of specified sensors according to the commands. The sensor local data is acquired and updated in a fixed time interval managed by SQLite database. However, when the frequency of data acquisition is high, the data in the embedded database will be drastically increased and there is a need to clear the historical data, or the access performance of the database will gradually decline.

V. THE DESIGN OF GUI AND DEVICE DRIVER OF THE LAB ENVIRONMENT INTELLIGENT MONITORING SYSTEM

The design of GUI for embedded systems is different from that of traditional data computing class software, which frequently handles mouse or keyboard events to complete a explicit calculation, while the former mostly handle events caused by touch screen and other kinds of external devices. since the embedded systems is resource-constrained, the propose mode of the GUI of the conventional PC, the memory consumption of which is relatively large and take up more CPU time, is not suitable for embedded systems. In this paper, Remote Monitoring and Controlling Systems uses Qt/Embedded under embedded Linux as its GUI application development platform, should be capable of fully satisfying the constraint of embedded system resources. As QT uses C++ as its programming language, it can implement hybrid programming with Linux C. The header files include both QT-API library and Linux system calls libraries. Write the Linux system calls as part of the slots functions which can respond to specific signals in order to achieve the combination of Qt / Embedded and Linux-C. Certainly, to achieve reading and writing of a specific device file, there should be device drivers which give reading and writing operation interface functions. Therefore, we require completing the preparing, configuring and modifying of the drivers of sensors, cameras and other external expansion device of S3C6410 microprocessor. The Laboratory Intelligent Monitoring and Controlling System use QT to complete GUI on the ARM head-end machine to achieve the graphical display of data collected by different sensors. This paper focuses on elaborating the design of the Linux drivers of various types of sensors and qtopia application in the system.

With the help of QT designer, the programmer can quickly develop relevant GUI window and adjust the size and position on that window. Also includes buttons and labels. By using labels we are displaying the current temperature, Light intensity concentrations of carbon dioxide and harmful gases concentration in the laboratory environment and showing whether the infrared sensors open or not. Using radio buttons we can also control (turn on/ off) the devices.

VI. PROGRAM IMPLEMENTATION

```
//Capturing image from camera
#include <QtGui/QApplication>
#include "mainwindow.h"
 int main(int argc, char *argv[])
{
CvCapture * camera = cvCreateCameraCapture(0);
assert(camera);
IplImage * image=cvQueryFrame(camera);
assert (image);
QApplication a(argc, argv);
MainWindow *w = new MainWindow(camera,0);
w->show();
}
MainWindow::~MainWindow()
{
delete ui ;
}
// Reading sensor values through serial (RS232) cable
void MainWindow::repeatTimer()
{
Tempval2;lightval2;voltval2;int p=0 ,length1;
 if(count== 5 || count==10 || count==15)
write(MainWindow::secom , "$N2!",3) ;
length =read(MainWindow::secom,buffer,19);
qDebug("bugger[%d]= %c",++p,buffer[0]);
buffer[length]='\0' ;
qDebug("this  light buff= %s  10= %d 11=%d
12=%d\n",voltbuff ,voltval2, Tempval2,lightval2)
}
buffer[0]='\0';
qDebug("tempval= %d" ,Tempval2);
qDebug("lightval= %d" ,lightval2);
qDebug("voltavla= %d" ,voltval2);
ui->leTemp2->setText(QString::number(Tempval2));
ui->leVoltage2->setText(QString::number(voltval2));
ui->leLDR2->setText(QString::number(lightval2));
returnfile =fopen("/root/returndata.txt","r") ;
QFile file1("/www/index.txt");
file1.open(QFile::WriteOnly);
QTextStream out(&file1);
printf("Content-type: text/html\r\n\r\n,charset=iso-
8859-1\",13,10");
//Uploading data, images on server
dir.rename("/www/index.txt","/www/index.html");
system ("cp /www/index.html /bin/index.html");
if (flgip == 0)
{
system ("ftpup");
flgip = 1;
}
system("rm /root/returndata.txt");
buffer[0]='\0' ;
tempbuff[0]='\0';ltbuff[0]='\0' ;
voltbuff[0]='\0' ;
qDebug("this si last repcount = %d",count);
count++ ;
if(count > 15)
count = 0;
}
```

VII. EXPERIMENTAL RESULT

The remote monitoring of sensors present in laboratory and video image information which is implemented by the combination of embedded SQLite database and embedded boa web server is shown in Figure 3. Also it can control the devices from remote location with Zigbee technology. The remote video monitoring system implements both real time image monitoring and automatically captured photo image files when any person enter into laboratory.

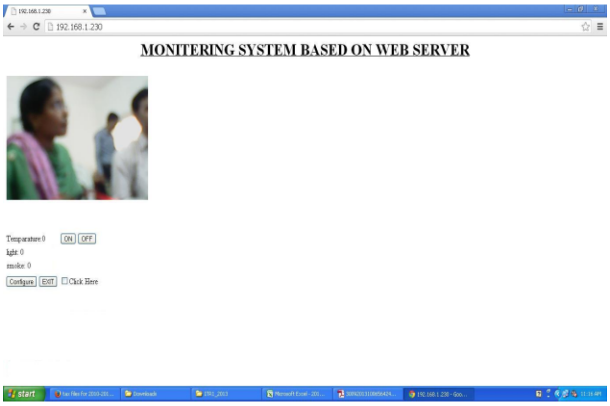

Figure 3

VIII. CONCLUSION

This paper is for monitoring conditions inside the laboratory and controls the parameters through web page. Build an embedded web server to implement the data of sensor network and video images to achieve remote monitoring and controlling which is based on ARM11 architecture. It develops a laboratory intelligent monitoring system with S3C6410 microprocessor as its major controller, elaborating the complex points of the development of the GUI applications based on Qt / Embedded and Linux drivers for various types of sensors in the project. We can develop this project in industrial sectors also will have better performance and broader market prospect.

ACKNOWLEDGMENT

This work was supported in part by a grant from my project guide Ms. Kanchan Dhote, (Assistant Professor in M. Tech Department of Electronic & Communication Engineering & Technology), Nagpur University, India and I express my sincere appreciation to her for constant encouragement.

REFERENCES

[1] Wang Xianchun, Guo Jierong, Hu Weiwen, and Fan Xiping, *"Design and Implementation of Embedded Web Server Based on ARM and Linux," Micro Computer Information, vol. 23(5-2)*, 2007, pp. 164-165.

[2] Jacek W, *"Embedded Internet technology in process control devices," IEEE Internet Computing, Vol. 34,* 2000.

[3] Douglas, *"Engineering Web Technologies for Embedded Applications," IEEE Internet Computing*, May/June 1998.

[4] Wang Tianmiao, Publication: *Embedded System Design and Case Development*, Beijing: Tsinghua University Press, 2002.

[5] Liu Yingshui, Xiao Zhengyu, and Sun Wei, *"Embedded Web Server Based on ARM and Linux," Microcontrollers & Embedded Systems*, June 2007, pp. 14-21.

[6] *Samsung Electronics Co Ltd. Users' Manual S3C6410* V0.12 [M] March, 2004.

[7] Yun Sin-quan, Lu Qiang, Qian Pei-del, *Implementation of Linux application based on Qt / Embedded [J]*, Computer Application and Software, 2006,23 (2): 105-107.

[8] Chen Kun, Chen Yun-qiu, Liu Xin. *Application design based on Qt / Embedded and embedded Linux [J]*. Computer and Digital Engineering, 2009,37 (1).

[9] Ying Feng, Feng Yu-fen. "*The design and implementation of home gateway based on SQLite database [J]", Micro-computer information*, 2008, 24 (5-2): 24-26. 914

[10] K. Ricks, D. Jackson, and W. Stapleton, "*An embedded systems curriculum based on the IEEE/ACM model curriculum," IEEE Trans Educ., vol. 51, no. 2*, pp. 262–270, May 2008.

[11] A. Behrens, L. Atorf, R. Schwann, B. Neumann, R. Schnitzler, J. Balle, T. Herold, A. Telle, T. Noll, K. Hameyer, and T. Aach, *"MATLAB meets LEGO Mindstorms—A freshman introduction course into practical engineering," IEEE Trans Educ., vol. 53, no. 2*, pp. 306–317, May 2010.

[12] J. Gómez-de-Gabriel, A. Mandow, J. Fernández-Lozano, and A. García-Cerezo, "*Using LEGO NXT mobile robots with LabVIEW for undergraduate courses on mechatronics," IEEE Trans Educ., vol. 54, no. 1*, pp. 41–47, Feb. 2011.

[13] EE Times Group, New York, NY, *Embedded Market Study*, Apr. 19, 2010.