COLLEGE NET WIFI ENABLED DATA ACQUISITION NETWORK USING OPENMOKO

Dated: 24th March, 2009

Mentored By:
Mr. Dhananjay V. Gadre

By:
Saurabh Gupta (81/EC/05)
Vijay Majumdar (97/EC/05)

ACKNOWLEDGEMENT

We would like to thank Mr. Dhananjay V. Gadre, Assistant Professor, Electronics and Communication Engineering Division, NSIT, for his timely help and suggestions whenever we faced difficulties anywhere in the project. We would also like to thank the whole Openmoko community and developers to help us throughout this project with their valuable guidance, suggestions, comments and feedbacks. Finally, we thank gtk+ community and PHP, MYSQL mailing list members for replying to our doubts and queries during the course of this project.

Contents

1. Introduction

The project aims to develop a data acquisition system on an embedded device (Openmoko) with wireless facility to send and store the data digitally on a centralized server. **Data acquisition** is the sampling of the real world to generate data that can be manipulated by a computer[1]. Acquired data are displayed, analyzed, and stored on a computer, either using custom made software, or custom displays and control can be developed using various general purpose programming languages such as C, C++, Python, PHP and the database can be handled using standard structured query language (SQL).

In a data acquisition system, there are several components which are integrated together and proper interface between these sub modules is necessary. The main parts of it are data acquiring unit, communication interface and data processing unit. In this project, communication is done wirelessly and data acquiring unit is a portable hand held embedded device. The following figure shows the main component of a data acquisition system.

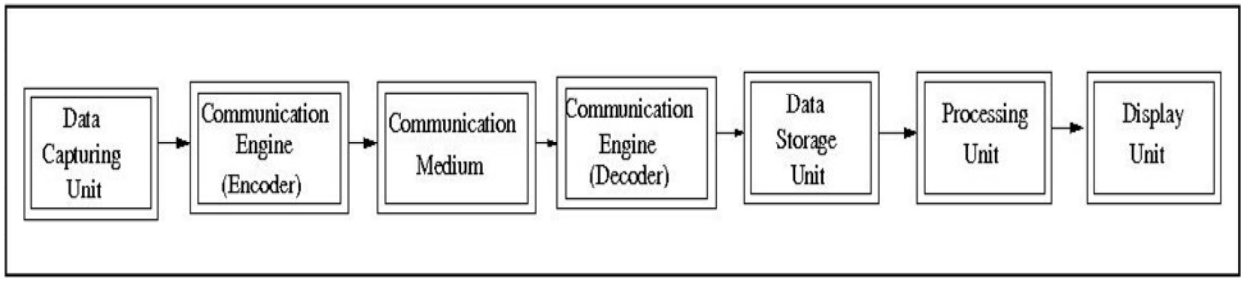


Figure 1: Modules present in a data acquisition system

This project aims to implement data acquisition system for acquisition of data on an hand held portable device. The data can be of any form. In this project, the application is built to facilitate teachers to take attendance, notes and other student's data and sending it wirelessly to a central server where data processing takes place and result can be viewed on an intra-college website. The various modules for data acquisition system in this project are:

Data Capturing Unit: It captures the data from the real world in digital form. This will be a portable touch sensitive embedded device called Neo Freerunner using software stack named Openmoko[2]. The device consists of ARM 16/32 processor with 250/400 MHz frequency. However, the application is developed in a way that it can be ported on any portable device provided it has a linux kernel running on it.

Communication Engine: This works to set up the communication and to encode/decode the acquired data in proper format. The communication is done to transfer the data from Data capturing unit and Data storage unit. In this project, this is a program written in C++ which works on the network layer to create a socket connection with the central server.

Communication Medium: The medium is Wireless communication using Wi-Fi protocol[3].

Data Storage Unit: This is a server with enough memory (more than 40GB) to store data in the form of databases which can be read using SQL.

Processing Unit: This is a computer which sufficient processing resource. It interfaces with the data storage unit and processes the data.

Display Unit: The processed data can be seen in form of tables and graphical user interface through an intra-college website. Every teacher has an account on the website and the result can be seen after logging in with the password set by user (teachers here).

This project is a combination of hardware units and software modules. Various frameworks and platform are used to build the whole application. The target devide for this project is openmoko.

1.1 Openmoko Framework:

Openmoko is a project to create mobile phones with an open software stack. **Openmoko Inc releases hardware**: phones to run the open source software stack. The first phone was the Neo 1973, followed by the current model, Neo FreeRunner. The Openmoko stack, which includes a full X server, allows users and developers to transform mobile hardware platforms into unique products. Its license gives developers and users freedom to cosmetically customize their device or radically remix it. It grants them the freedom, for example, to transform a phone into a medical device or point of sale device or the freedom to simply install their own favorite software. Beyond freeing the software on our devices it has also released its CAD files under Creative Commons.

1.1.1 Software Framework

The framework consists of many cross compilers for C, C++, python and many more. All libraries support for building applications can be found here. The software architecture of Openmoko is given in the next figure.

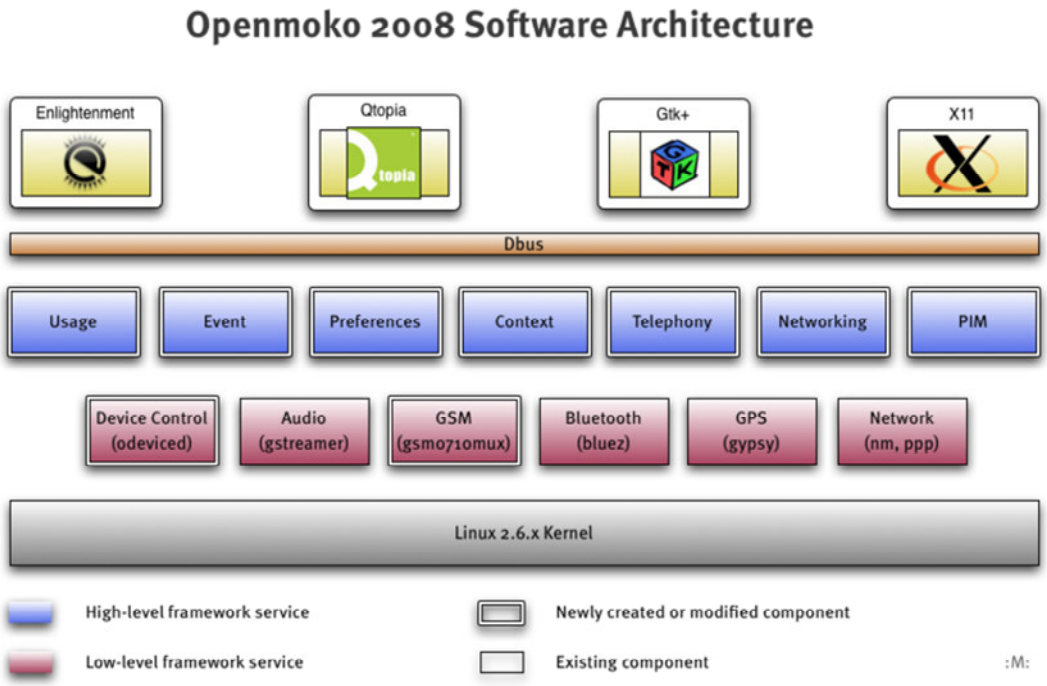


Figure 2: Openmoko Software Framework

1.1.2 Hardware Framework

The device is a portable touch sensitive mobile phone which has the following hardware specifications and supports the following with software also.

Figure 3: Neo Freerunner Device

Hardware Electrical

- 400/500 MHz Samsung 2442B Processor/SOC (400 minimum, ARM920T core, ARMv4T)
- Boot code in NAND FLASH or 2MB NOR FLASH (optional design)
- 128 MB SDRAM total, 64 MB CPU internal, 64 MB external
- 256MB NAND Flash MCP package.

Display

- Topploy VGA, 16 bit color depth
- resolution: 480 x 640 pixels
- size: 43mm x 58mm (1.7” x 2.27”)
- White LED backlight. Required brightness is 200 NIT minimum.

WiFi 802.11 b/g transceiver

- TX power at 11 Mbps: 13 dBm minimum
- RX sensitiveity at 11 Mbps: -89 dBm desired, -83 dBm minimum

Serial interfaces (UART)

- A-GPS or GPS
- GSM/GPRS

Accelerometer

- 2x accelerometer required
- 3 axis sensing

GPS and AGPS

- GPS chipset receiver and antenna
- Sensitivity at Antenna port: -157 dBm tracking on chipset specification

1.2 Openmoko Distributions

Openmoko distributions are designed to run on various mobile devices, with the primary aim of supporting Openmoko Inc.'s Neo FreeRunner phones. They are GNU/Linux distributions -- complete operating systems with more or less user applications. There are many software distributions available in Openmoko. The speech recognition application is built mainly for OM distributions. The following distributions are available in Openmoko currently:

- OM 2007.
- OM 2008.8
- FSO
- QT extended
- Debian
- Android

1.3 Openmoko Tools for development

There are a lot of tools available for developing application on the Openmoko platform.

1. **Openmoko toolchain**: A prebuilt toolchain is available for directly developing applications for Openmoko and cross compilation. The speech recognition project is developed using this toolchain. It is available for i686 and x86_64 hardware platforms.

2. **Openembedded**: The popular openembedded and bitbake platform is also available for compiling and building applications on Openmoko platform. The whole distribution image can be compiled using bitbake utility.

3. **Emulator:** Applications developed for Openmoko can be tested and emulated using the qemu emulator available for Openmoko. Qemu loads the available Openmoko distribution image and then emulates the device and is used for testing purposes.

2. Implementation

The implementation of this project involves usage of various platforms and frameworks. It includes working developing software application, graphical user interface, interfacing of Neo Freerunner with server through Wi-Fi etc. Following are the brief steps involved in this project:

1. Developing application for communication between the data capturing unit and storage unit.
2. Developing graphical user interface to facilitate the user acquire and record the data in a comfortable and easy way.
3. Developing protocols and database structure to be used in the system.
4. Porting all applications on the embedded device along with full customization support.
5. Set up of a central data server to communicate with data capturing unit.
6. Developing an intra college website with many configurable options and graphical user interface using LAMP technology.

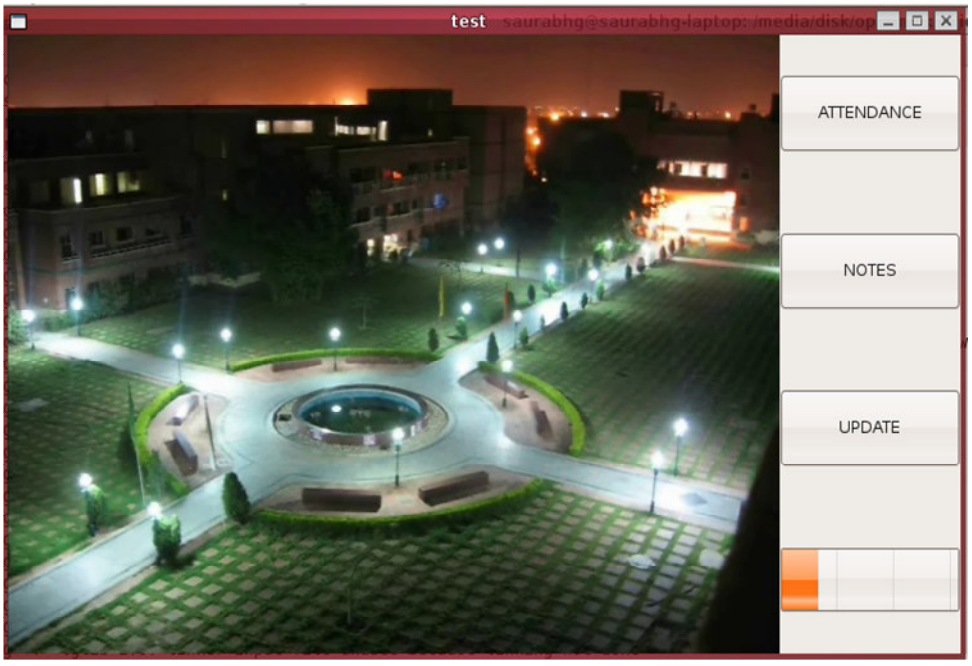
3. Result And Discussion

Several modules in this project have been tested and integrated. The latest version of the code developed in this project can be obtained from
http://attendance-on-openmoko.googlecode.com/svn/trunk/ .
Communication engine through wi-fi has been developed between the Openmoko Device and central database server. The exchange of data and files takes place in a robust way through a defined format. Proper error handling and treatment of failure cases is insured.

3.1 GRAPHIC USER INTERFACE

Graphical user interface has been developed to ease the data acquisition procedure. The snapshots of GUI developed are:

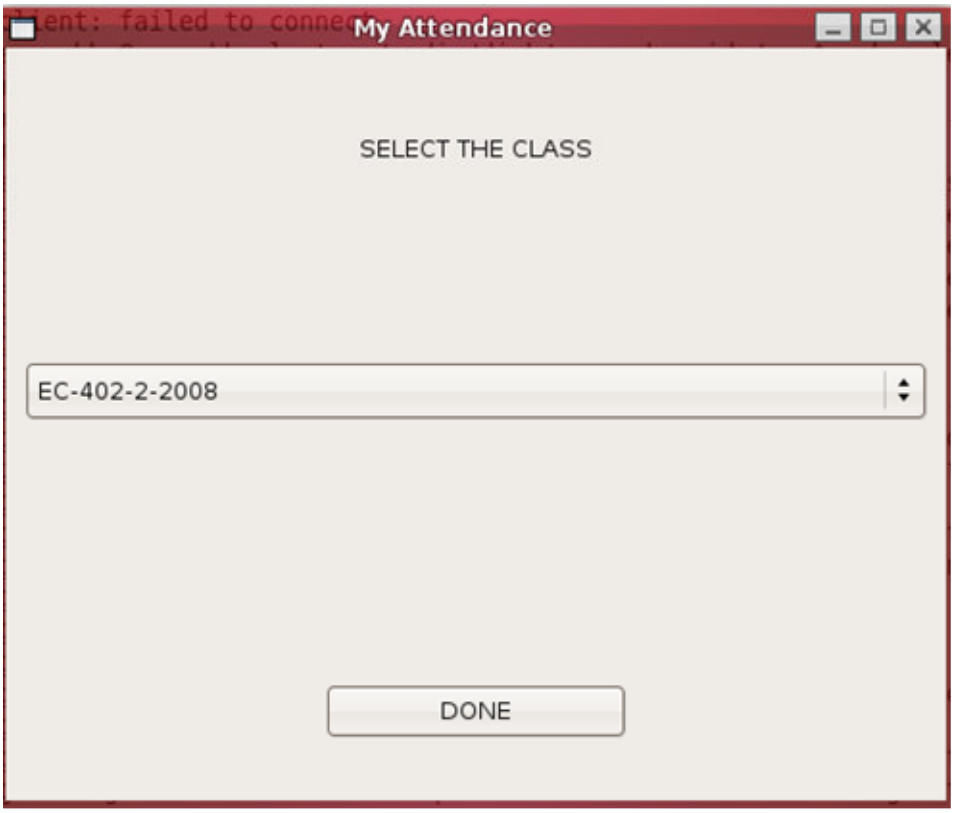
SNAP 1:



Teacher can select from here

1. If he wants to take attendance for a particular class.
2. If he wants to write down notes about students of any class.
3. Or he needs to update the list of classes he takes from the central server.

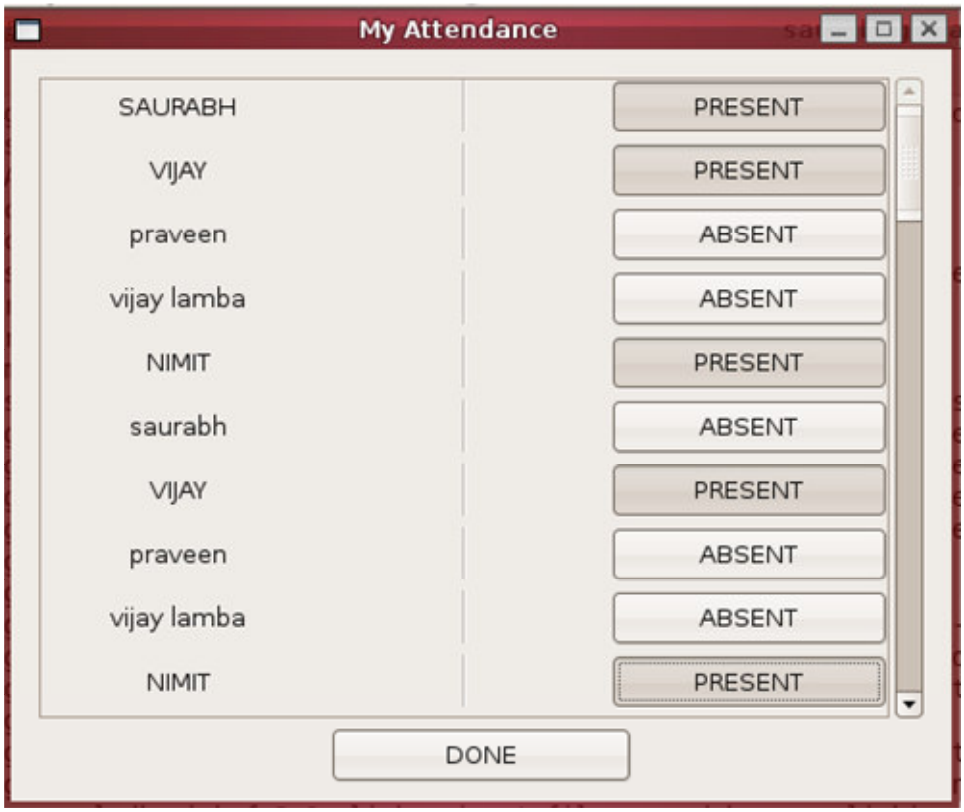
SNAP 2:



Teacher can select

1. The class whose attendance he wants to take.

SNAP 3:



This section

1. Displays the list of students registered in the class.
2. Teacher can mark the attendance from here.
3. The attendance is sent to the central server automatically after marking the attendance.

3.2 COMMUNICATION ENGINE DEVELOPMENT

This part is a separate process which works with the GUI. GUI part and communication engine communicate with each other through IPC (inter process communication). This engine is responsible for sending and retrieval of data from the Central Database.

The communication is done on the network layer implemented in C++ and PHP. Socket programming is used extensively to bind the socket with an available network. On Openmoko side, communication protocol is written in C++ and data is transferred through TCP/IP protocol. On server, a similar application is running which receives/sends data on a pre defined port and IP. This project

If the connection cannot be built at the time attendance is taken, the ouput file is stored locally. Next time, whenever a communication is made through wi-fi, all pending files will be sent and database will be updated. A provision is also made by which the the data stored on Openmoko can be sent to database server by attaching it with PC and a proper GUI will be made for that. This feature is helpful in case wi-fi connection is not working in the college.

4. Remaining Work

4.1 Website Development

To access the recorded data and display it, an intra college website is to be developed using LAMP(Linux-Apache-MySQL-PHP) technology. This requires setting up a server having static IP address along with proper log-in facility in a secured way.

4.2 More GUI support inclusion

We are aiming at better GUI support inclusion with the current application. So that the application become more responsive and User Interacting.

4.3 Deployment of whole system in NSIT

This will include the deployment of a Wi-Fi system in NSIT without enough range and payload capacity. In this LAN technology a Server setup will be employed with Static IP , the underlying technology will be LAMP (LINUX- Apache – PHP - My SQL) on a Linux system, which will operate 24x7 with proper battery backup.

Embedded Hardware Device will be distributed to every teacher with pre stored data about the classes. Every teacher will have his account

Conclusion

The project results in a very useful application which allows data recording in a more automated, secured and comfortable way. The teacher's task of taking attendance is simplified as no manual work is involved in maintaining the attendance statistics.

In addition, the presence of a central database obviates the need for regular submitting of attendance to the administration, which is itself taken care of. Time, is automatically saved as any data can be retrieved within no time. Besides having the facility for marking attendance and notes, this system can be extended to any activity in the college work e.g. data acquisition system and automatic storage in library related activities etc.

Besides recording data manually, more advanced features can be integrated with this project. For example, speech recognition API can be integrated with it and then attendance can be marked just by speaking the roll numbers of the present students. In speech recognition, teacher voice and words are pre stored in the device and it will be recognized as "PRESENT" or "ABSENT" and attendance will be marked.

Another possibility is to integrate the system with image processing API. A camera can be integrated with the device and it will capture the image of the whole class. Face recognition software will run on the device and mark the attendance of the present students. The current target device Openmoko's hardware supports the facility for camera.

References

1. http://en.wikipedia.org/wiki/Data_acquisition
2. http://wiki.openmoko.org/wiki/Main_Page
3. http://en.wikipedia.org/wiki/WiFi
4. http://code.google.com/p/attendance-on-openmoko/
5. http://attendance-on-openmoko.googlecode.com/svn/trunk/ .