



EDUCATIONAL BLUETOOTH QUIZZING APPLICATION FOR ANDROID PHONE BASED ON RASPBERRY PI

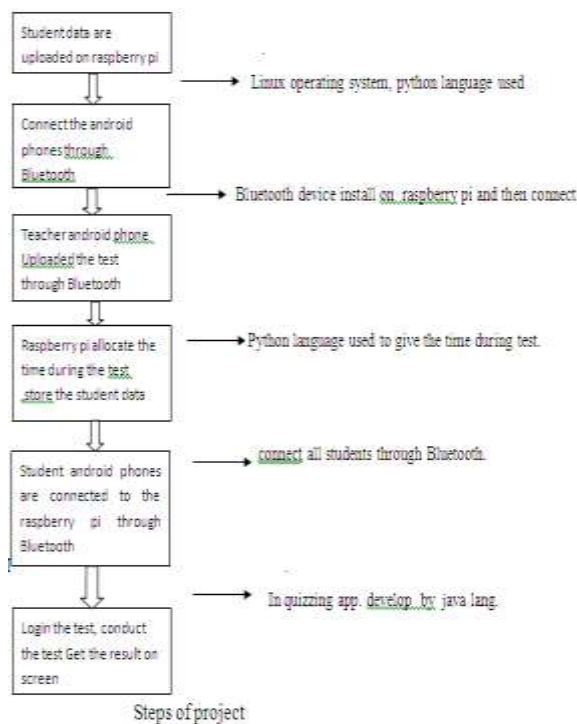
Prof-Zadbuke Ashwini S¹, Miss-Amruta Khot V², Miss-Kharat Sandhya S³,
Miss-supriya shelar P⁴, Miss -khomane shital V⁵
1,2,3,4,5 S.B.Patil College Of Engineering Indapur

Abstract : Nowadays, the developments in mobile learning give an opportunity to students and instructors to enhance their learning and teaching respectively. To increase the student knowledge, attendance, interest to learning the subjects. Bluetooth technology allow electronic device to communicate wirelessly. This paper is about Bluetooth quizzing system for student using RASPBERRY PI. It will utilize a quizzing system to allow the many students for giving answer. When Bluetooth is connected the can register and choose the option to complete the quiz. The result is automatically sent when quiz is done .Then data analysis can done.

Keyword: Raspberry pi, Bluetooth device, embedded system, educational application, quizzing application

I. INTRODUCTION

Currently mobile technology offers to deliver the study material on class through Bluetooth. In this paper do the quiz is done in classroom by using Bluetooth and raspberry pi. Bluetooth is device that connected to the raspberry in USB port. Android phones connected to the raspberry pi through Bluetooth. To connect the android phone to raspberry pi through Bluetooth we will use python language. Python's beauty is that it is accessible to beginning programmers and allows them to tackle interesting problems more quickly than many other, more complex languages that have steeper Learning curve. Steps:



We will design the quiz application to complete the quiz in classroom. To design quiz application will use java language. The data will store the SD card on raspberry pi. Only those students will register and give the test whose data can be store in raspberry pi.

Flow of project:

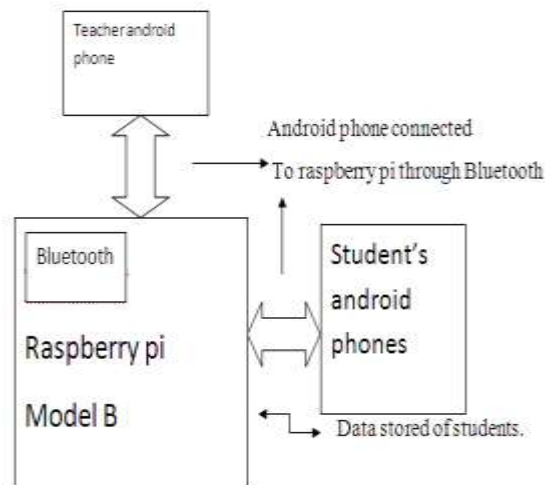
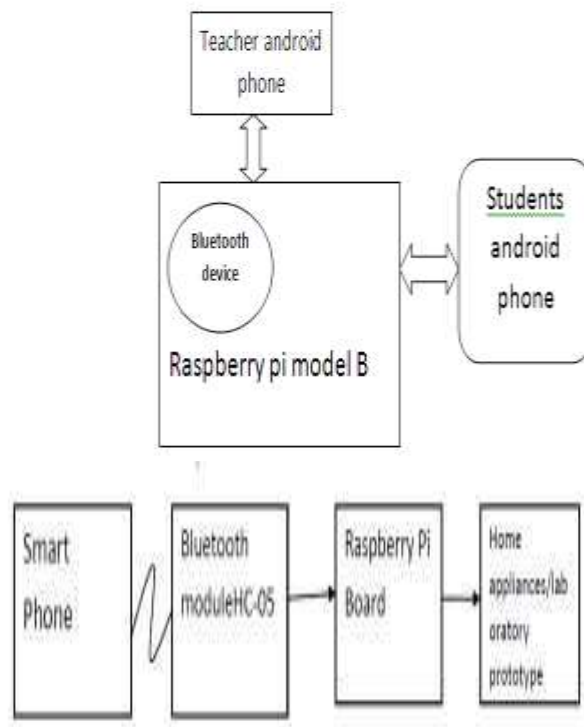


Fig. flow of project

BLOCK DIAGRAM

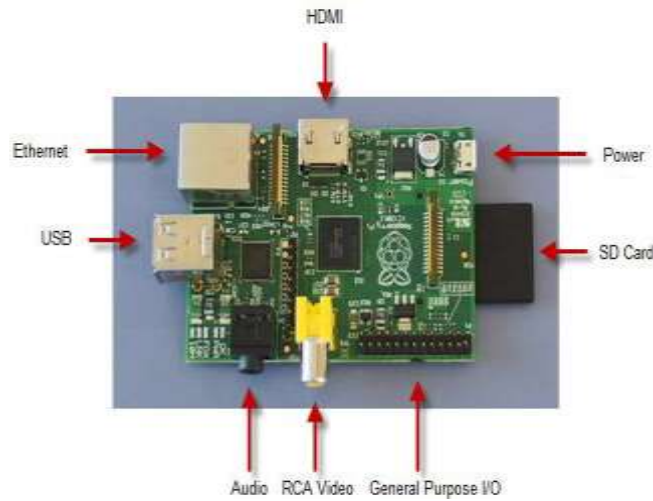


Block Diagram

II. RASPBERRY PI

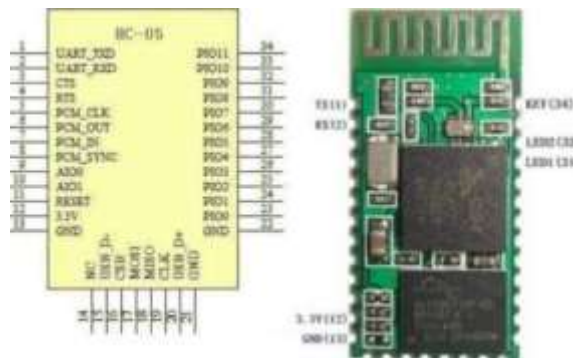
Raspberry pi useful because programmers tend to tinker with their development machine, and tinkering can break things. Generally this doesn't damage the machine itself, but it can require you to reinstall the system, which can mean a bit of lost data, and it can put the machine out of action for a few hours.

Raspberry Pi Diagram



FEATURES	SPECIFICATION
CPU	700MHz ARM1176-JZFS
GPU	Broadcom Video Core IV
Memory	256MB LPDDR2-800
Video	HDMI, composite
Audio	HDMI, stereo analog
USB	2 x USB2.0 (model B)
Storage	SD card
Networking	10/100 Ethernet
Power	5V micro USB

BLUETOOTH:



III. BLUETOOTH MODULE:

Bluetooth Smart technology is a wireless communications system intended to replace the cables connecting many types of devices, from mobile phones and headsets to hear monitors and medical equipment. In my project I am using another module for communication between voices from smart phone to raspberry pi by using Bluetooth module. For this purpose

HC-05

module being used and the specifications of Bluetooth module as follows.

- Bluetooth protocol: Bluetooth Specification
 - v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: $\leq 4\text{dBm}$, Class 2
- Sensitivity: $\leq -84\text{dBm}$ at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps,
- Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75Centigrade

Bluetooth module(HC-05) Raspberry pi GPIO

Bluetooth module(HC-05)	Raspberry pi GPIO
Vcc	Vcc(pin #2)
Rxd	Rxd(pin #8)
Txd	Txd(pin #10)
GND	GND(pin #6)

Pin connection from Bluetooth to Raspberry

RASPBERRY PI CONFIGURATION

Here Raspberry Pi used for controlling purpose and it is almost like a small computer so to start with this we have to configure the board. The programming language here going to use is python. We are going to running a recent release of Raspbian on my Raspberry Pi. Commands for required Installations as follows

1. Sudo apt-get install python-dev for python installation
2. Sudo apt-get install python-rpi-gpio for accessing GPIO

Pins of Raspberry pi

Both the commands can be done by in online manner we can done it in offline also. By default the Raspberry Pi is

configured to write boot time messages to the serial port, and also to start a login console on it. Unfortunately, the default baud rate that the RPi uses for its serial port is 115200 bps, while the

Bluetooth module comes preconfigured from factory to 9600 bps. It is easier to configure the RPi to use 9600 bps so we will try that first. There are two config files that need to be updated.

File /boot/cmdline.txt contains the kernel options that are used to boot the system. In my Raspbian based system this file contains the following options:

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200  
kgdboc=ttyAMA0,115200 console=tty1  
root=/dev/mmcbk0p2 rootfstype=ext4 elevator=deadline  
rootwait
```

The interesting options are console and kgdboc, because these configure the serial port device /dev/ttyAMA0 to 115200 bps. We need to change these two configurations to 9600 bps. After you make these changes the file should read

```
dwc_otg.lpm_enable=0 console=ttyAMA0,9600  
kgdboc=ttyAMA0,9600 console=tty1  
root=/dev/mmcbk0p2 rootfstype=ext4 elevator=deadline  
rootwait
```

The second configuration file is /etc/inittab. Inside this file you have to locate the following line:

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

This tells the system to start a terminal on the serial port, and again it uses 115200 bps to configure the port. You have to change this line to use 9600 bps:

```
T0:23:respawn:/sbin/getty -L ttyAMA0 9600 vt100.
```

Remember that these are configuration files so they are not writable to the default pi user, to edit these files you have to use sudo. As far as text editors, Raspbian provides two, vi and pico. For example, to edit cmdline.txt with pico you would run the following command:

```
$ sudo pico /boot/cmdline.txt
```

The system will ask for us password and after that we will be able to make changes to this file. I recommend that we save original copies of these config files in case we make a mistake.

IV. LITERATURE REVIEW

A mobile quizzing system through which students can answer short questions in lectures with the use of Bluetooth. The main objective of this system was to provide the students with different technology options that would enhance their learning experience, as well as, encourage them to attend lectures thereby increasing the level of participation.

Raspberry pi useful because programmers tend to tinker with their development machine, and tinkering can break things. Generally this doesn't damage the machine itself, but it can require you to reinstall the system, which can mean a bit of lost data, and it can put the machine out of action for a few hours.

V. CONCLUSION

A Bluetooth quizzing application was developed for use in lectures. To help lecturers administer quizzes and in turn help students revise work previously learnt. Students will benefit by revising work done in previous classes. Bluetooth transmission in Raspberry Pi controller board with utmost accuracy. Also when there is a mismatch between the sent and received data, we were able to detect it at all instances and notify the client system. Future works could improve on this method for supporting more simultaneously Bluetooth Connections on the server side.

REFFRANCE

1. Core Version 3.0. Accessed August 24, 2012. <https://www.bluetooth.org/en-us/specification/adopted-specifications>.
2. Hosny, W. 2007. "Power Engineering Mobile Education Technology." 2007 42nd International Universities Power Engineering Conference, 4-6 September, 2007, 971-974. Brighton
3. International Telecommunications Union. 2011. "Mobile Cellular telephone subscriptions. Accessed May 24, 2013. http://www.itu.int/ITU-D/icteye/Reporting/ShowReportFrame.aspx?ReportName=/WTI/CellularSubscribersPublic&ReportFormat=HTML4.0&RP_intYear=2011&RP_intLanguageID=1&RP_bitLiveData=False.
4. Miller, Michael. 2001. *Discovering Bluetooth*. Sybex Incorporated.
5. Kumar, C. Bala, Paula J. Kline, and Timothy J Thompson. 2004. *Bluetooth application programming with Java API*. Elsevier Incorporated.
6. J. Bray, C. F. Sturman, "Bluetooth 1.1: Connect without Cable", *Pearson Education*, edition 2, 2001.
7. A. S. Huang, L. Rudolph, "Bluetooth essentials for programmers", *Cambridge University Press*, 2007
8. Powers, Shawn. "The open-source classroom: your first bite of raspberry pi." *Linux Journal* 2012.224 (2012): 7
9. Mitchell, Gareth. "The Raspberry Pi single-board computer will revolutionize computer science teaching [For & against]." *Engineering & Technology* 7.3 (2012): 26-26.
10. L. Rabiner, —A tutorial on Hidden Markov Models and selected applications in speech recognition, *Proceedings of the IEEE*, vol. 77, pp. 257-286, 1989.
11. www.imagesco.com/articles/hm2007/SpeechRecognitionTutorial01.html
12. V. Ramya, B. Palaniappan, Bobby George, Embedded System for Automatic Irrigation of Cardamom Field using
13. Xbee-PRO Technology, *International Journal of Computer Applications* (0975 – 8887) Volume 53– No.14, September 2012.
14. F. L. Alegre, "Application of ANN and HMM to Automatic
15. Speaker Verification", *IEEE LATIN AMERICA*
16. *TRANSACTIONS*, Vol. 5, No. 5, pp. 329-337, Sept. 2007
17. 14) <http://appinventor.mit.edu/explore>