First-Person Shooter Gaming Simulator

Tajmalludin Bin Yaakob
Faculty of Electrical Engineering
Universiti Teknologi MARA
40450 Shah Alam, Selangor, Malaysia
taj.jacob@gmail.com

***Abstract*—In this paper, I have developed a device that worked with first-person shooter (FPS) game for personal computer (PC) platform. The idea of this project is to change the way of playing the FPS game from using the conventional method; mouse and keyboard to the real gun model. This project was using the Arduino Due as the microcontroller and the accelerometer ADXL335 as the main subject besides joystick and keyboard. This project also studies the behavior of The ADXL335 that can measure three types of data; acceleration, gravity and angle.**

***Index Terms*—FPS, PC, ADXL335, joystick**

I. Introduction

First-person shooter (FPS) is a video game genre centered on weapon and projectile weapon-based combat through a first-person perspective with shooting as the main action of the game [1]. Counter Strike: Condition Zero, Call Of Duty, Battlefield are the examples of FPS games [2]. For PC platform, this type of game usually played by using mouse and keyboard. The mouse usually uses to aim and shoot the target meanwhile the keyboard use to control the movement, change weapons, and other actions. Figure 1 shows the gameplay of Counter Strike: Condition Zero.

Figure 1

II. PROBLEM STATEMENTS

In Malaysia, the video game industry is not a very popular industry. In contrary, America especially they make huge amounts of money from this industry. In 2010, The NPD Group/Retail Tracking Service reported about 15.9 billion US dollars for total the consumers spend their money on computer and video games [3]. The video game evolve from the way of user interface to game controller. The video game console, Wii brings the new dimension of gameplay. Instead of using a typical game controller, they come out with The Wii Remote™ Plus controller [4], a game controller based on motion detection by swing, swipe, thrust, or turn the controller.

Average gamer spends about 8 hours per week playing video games [5]. The idea coming when the conventional FPS game for PC platform required the player to use a mouse and keyboard as the game controller. This method makes the player sit on the chair and facing the screen too close. A report from the American Optometric Association reviews that the exposure of a computer screen can cause Computer Vision Syndrome (CVS) and the symptoms occur such as eyestrain, headaches, blurred vision and neck or shoulder pain [6]. Furthermore, sitting in a chair for a long hours cause the electrical activity in human muscles slows down and the calorie-burning rate drops to one calorie per minute [7]. This project will work on how to overcome these problems.

III. OBJECTIVES

This device was designed for FPS game for the PC platform. Its design will look like a real weapon so the player will experience the game like handling the real gun and improved the shooting skill. This gadget already in the market for platform PS3, Xbox 360 and Nintendo Wii. Nevertheless, there is no device such a like this for the PC platform. The main idea is to transform the mouse and the keyboard into one device.

There are four main objectives of this project. The first objective is to give the player a new experience of playing the FPS game. Next is to transform the mouse and the keyboard in a single device. Besides that, the device is user friendly, which means the player only need to plug and play. Finally, to study the accelerometer to work as a mouse [8].

IV. METHODOLOGY

Counter Strike: Condition Zero was chosen as the platform to test the device. Every setting in this game was synchronized to fit in the device. For this project, the device was designed to look like the real assault rifle so the player will feel like holding the real weapon. Figure 2 shows the model of The Target. The position of each button was designed to make the player comfortable therefore can enhance the skills of the player. The Arduino Due was chosen as the microcontroller for this project.

Figure 2

Figure 3 shows the block diagram of the device. There are three inputs and also three outputs. For the inputs, 3 components was used which is the accelerometer ADXL335, the 2-axis joystick and the push buttons. The accelerometer works as mouse cursor replacing the regular optical mouse. The optical mouse required the smooth surface in order to work. With the accelerometer, the way of controlling the mouse will change by pointing the accelerometer to the target. The 2-axis joystick works as a virtual keyboard. It will give analog signal to the microcontroller. The challenge here is to change the analog signal into a digital signal. The push buttons will be the inputs for making the virtual keyboard.

For the outputs, there will be 3 components will be expected; virtual mouse, virtual keyboard and vibration motor. For the vibration motor, it only vibrates when the left click is clicked.

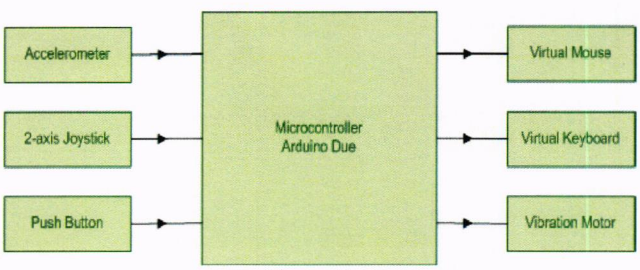

Figure 3

Before starting the programming, the flowchart must be made so the coding well organized. The flow chart of The Target shown in Figure 4. The programming was written in The Arduino Development Environment. The coding starts by determining the pins. For the push buttons of keyboard and mouse, the pins were set to digital pin where else the accelerometer and the 2-axis joystick was set to analog pin.

The second step is to initialize code for keyboard and mouse. For the Arduino Due, it enables to emulate a USB mouse or keyboard to an attached computer by using a specific command. In this step also, the pin mode must be specified whether it is as input or output. By referring the block diagram in figure 3, the pin for accelerometer, joystick, keyboard and mouse will be set as input and the vibration motor pin as output. All of this second step was put in void setup command. For the next step, it will divide into 3 major coding; push buttons, 2-axis joystick and accelerometer.

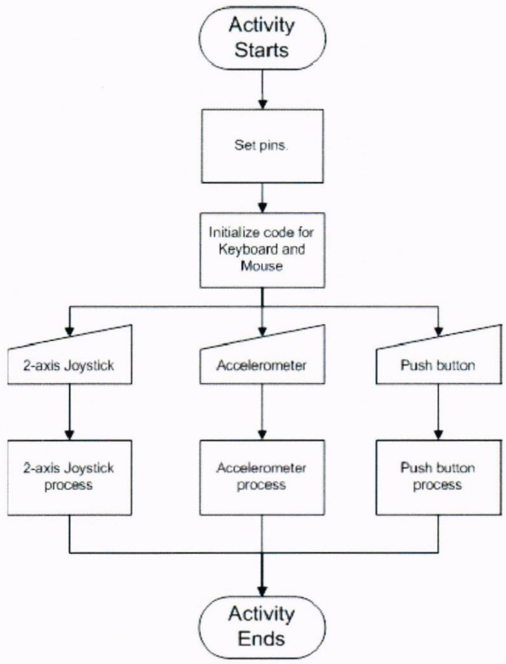

Figure 4

The push buttons were used as keyboard and mouse clicks. Figure 5 shows the flow chart of push button programming. The programming was divided into 2: keyboard and mouse. For keyboard programming, the selected key was chosen based on the setting in Counter Strike game. To make the Arduino Due works as a keyboard, the command used was 'keyboard.begin();'. This command must be written in void setup column. To send a keystroke to a connected computer, the command 'Keyboard.write();' was used. Every time the push button is high, it will send the keystroke to the computer. The delay 100miliseconds must be put after 'Keyboard.write();'. Only ASCII characters that are on the keyboard are supported.

In order to make the Arduino Due works as a mouse, the command 'Mouse.begin();' was used. When the push button was high, the mouse will act as right click or left click. In addition to the left click, the vibration motor will act whenever the push button is high. The function of the vibration motor is to give a feeling to the player to experience like shooting with real guns.

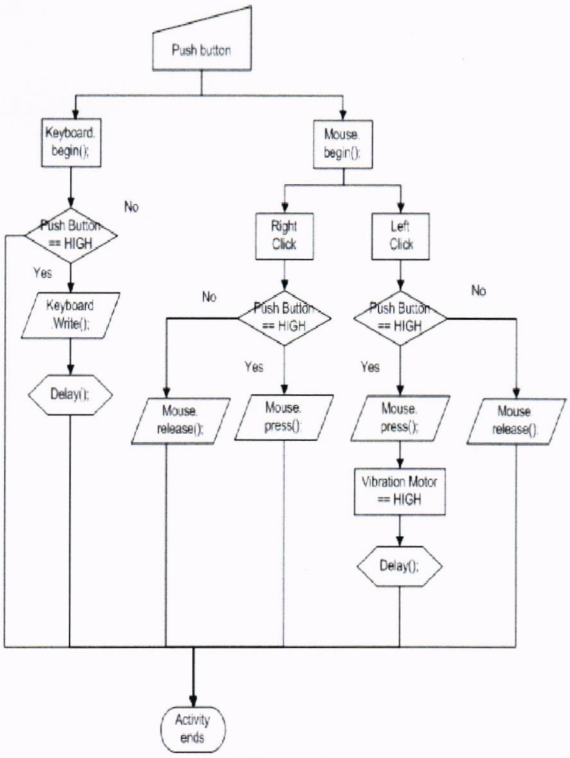

Figure 5

The accelerometer was planned to be as a mouse pointer in this project [9]. Figure 6 shows the flowchart of the accelerometer ADXL335. The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. It measures acceleration with a minimum full-scale range of ±3 *g*. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

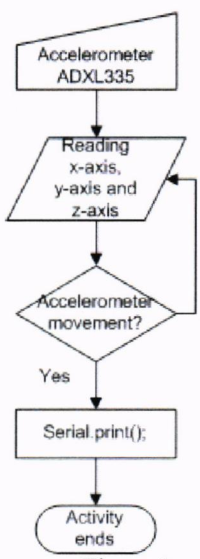

Figure 6

The accelerometer was connected to the analog pin to read the axis. There are 3 types of readings was taken; voltage, gravity, and angle. For voltage's reading,

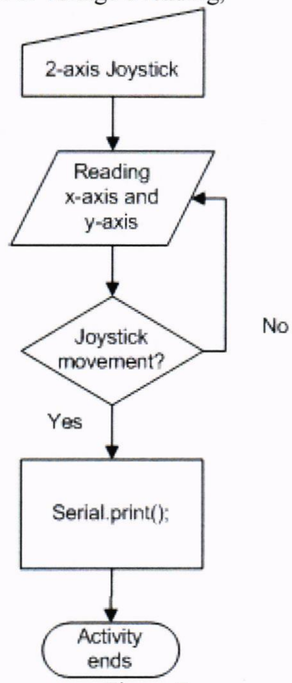

Figure 7

Figure 7 shows the flowchart of 2-axis joystick. The reading for x-axis and y-axis was taken in millivolt unit.

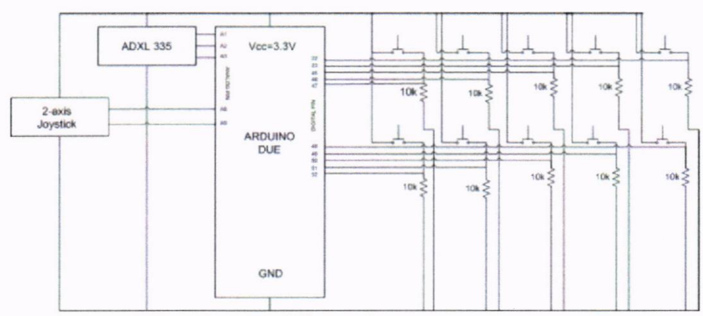

Figure 8

Figure 8 shows the circuit diagram of the project. The push buttons were connected to the digital pin where else the accelerometer and the joystick was connected to the analog pin. There are two ports in the Arduino Due, programming port and native USB port. The programming port is the default for uploading sketches and communicating with the Arduino. It was used to take the reading from the accelerometer and the joystick. The native USB port enables the Due to emulate a USB mouse or keyboard to an attached computer.

V. RESULTS AND DISCUSSIONS

Upon implementing on Arduino Due, several readings for joystick and accelerometer was taken to understand their behavior. Figure 7 shows the graph of 2-axis joystick voltage versus time. The reading was taken by moving the joystick in different directions. For the first ten seconds, the position of the joystick is static and the reading for x-axis is in average 514mV where else the y-axis is in average 500mV. Then move it up for 10 seconds and the average readings for x-axis is remain 500mV but for y-axis was dropped to 0mV. At 30s to 40s, the joystick was moving down and the average reading for x-axis still maintains 500mV but the y-axis get high about 1023mV. At 50s to 60s, the joystick was moved to the right. This time the average reading of x-axis became high which is 1023mV meanwhile the y-axis is 500mV. When joystick was moved to the left, the ex-axis's reading was dropped to 0mV but the y-axis is maintaining 500mV.

The initial idea of using a joystick is to use it as a keyboard which is given the keystroke output. When the joystick is moving, it will give keystroke such as w, s, a, and d as the setting for moving the boat in the game. Since the accelerometer has problem to make it as mouse pointer that will explain in the next paragraph, so the joystick was changed the role from keyboard to mouse pointer.

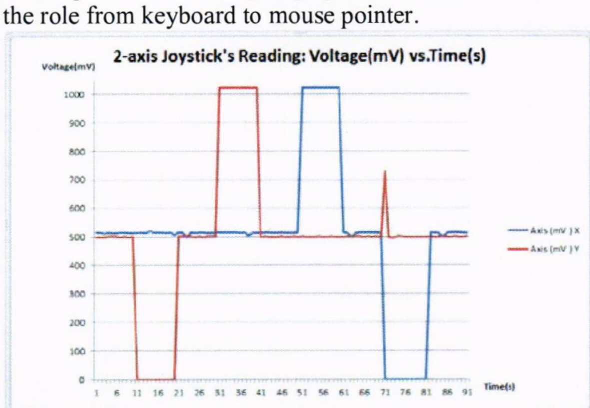

Figure 9

Table 9.1 shows the average of reading based on the position of the joystick.

Direction	Static	Up	Down	Right	Left
x-axis(mV)	514	514	514	1023	0
y-axis(mV)	500	0	1023	500	500

Table 9.1

The accelerometer can measure three types of data; acceleration, gravity and angle. Before taking the reading, they must be soldered to get good reading or else it will not give a good result. Every data was collected to study its behavior. In every readings, the position of the accelerometer was changed every 10 seconds.

Time(s)	10	20	30	40	50	60	70	80	90
Position	Static	Up	Static	Down	Static	Right	Static	Left	Static

Figure 10 shows the accelerometer's graph of voltage against time. The reading was taken straight from the serial port without any adjustment at the coding. Table 10.1 shows the reading at different position.

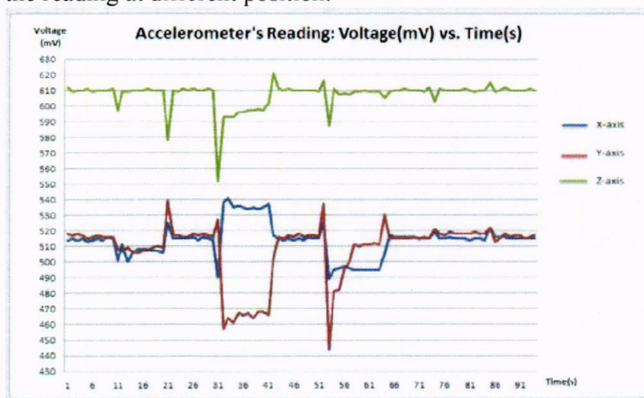

Figure 10

Direction	Static	Up	Down	Right	Left
x-axis(mV)	514	501	490	525	495
y-axis(mV)	518	508	527	537	511
z-axis(mV)	612	597	552	616	609

Table 10.1

Figure 11 shows the accelerometer's graph of gravity against time. From the ADXL335 datasheet at 3.3V power, we should expect an axis to read 1.65V when it has zero acceleration, and the voltage should typically change by 330 mV per G of acceleration. The signal from our analog to digital converter gives us a number from 0 to 1023. It was called these "ADC units". 0V maps to 0 ADC units, 3.3V maps to 1023 ADC units and I assume it is linear in between. This means that zero acceleration on an axis should give us a reading of 512 ADC units on the pin for that axis. Also, a change of 1 ADC unit in our signal corresponds to a voltage difference of 3.3V/1023 ADC units = 3.226 mV/ADC unit. Since the datasheet says 1G typically corresponds to 330 mV voltage difference, we expect that:

330 mV/G = 330 mV/G × (1023 ADC units) / 3.3 V = 102.3 (ADC units)/G

From the equation above, the analog outputs from the accelerometer can be turned into gravity(G) unit. Table 11.1 shows the reading of gravity at different position. Logically, the reading of accelerometer must be zero when in static position but the reading shows a little different. This happened because of the soldering part at accelerometer bending a little.

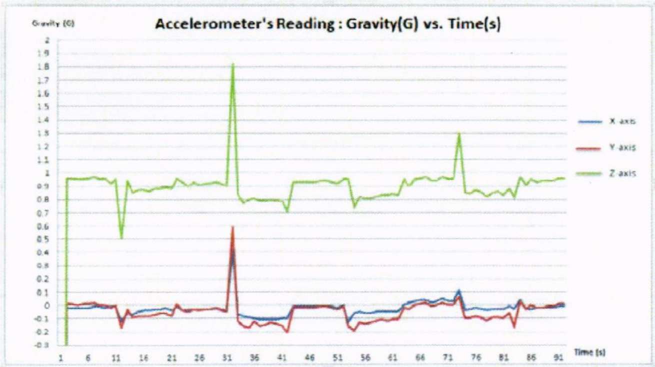

Figure 11

Direction	Static	Up	Down	Right	Left
x-axis(G)	-0.02	-0.12	-0.03	-0.01	0.05
y-axis(G)	0.03	-0.05	-0.01	-0.01	-0.03
z-axis(G)	0.95	0.67	0.96	0.95	0.94

Table 11.1

Figure 12 shows the accelerometer's graph of degree against time. This reading was taken slight different than previous reading. The accelerometer was flipped to 90degree from front to back and from right to left. Table 12.1 shows the reading from different angles.

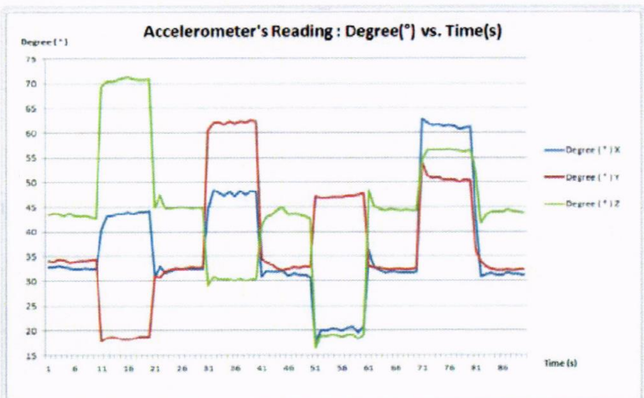

Figure 12

Direction	Static	Up	Down	Right	Left
x-axis(°)	32.69	43.19	48.34	20.04	62.74
y-axis(°)	34.18	18.43	62.03	46.75	53.97
z-axis(°)	43.38	70.45	30.84	18.94	54.69

Table 12.1

The oscilloscope was used as additional to understand the behavior of the accelerometer. Figure 13 shows the graph of x-axis while Figure 14 shows the graph of y-axis. Meanwhile Figure 15 shows the graph of z-axis. From these figures, we can say that x-axis and y-axis have similar behavior which is when the accelerometer was tilted, the reading was changed and remain until it back from the static position.

The z-axis has different behavior from other axes. If the accelerometer moved up, the reading changed and go back to the static position. After understanding the behavior of accelerometers, the next thing to understand is how making the accelerometer as a mouse.

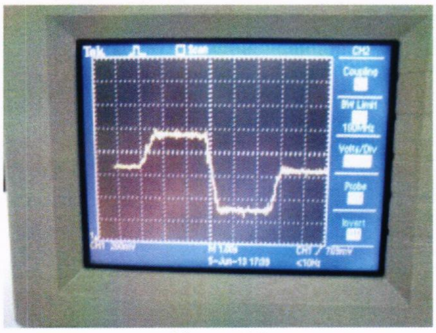

Figure 13

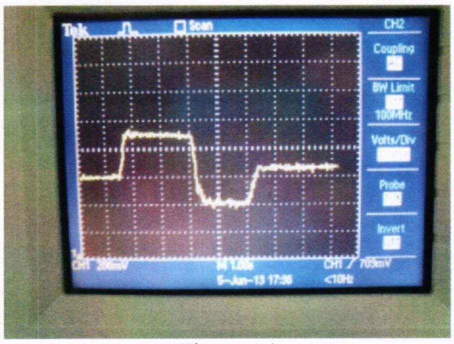

Figure 14

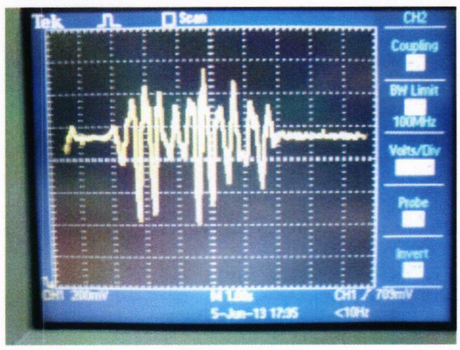

Figure 15

To make the Arduino act as a mouse, the cable was connected to the native USB port. Then the command 'Mouse.begin();' was written in void setup column. Next to move the cursor, the command 'Mouse.move(x-axis, y-axis, 0);' was used. This command only needs two axis to make the movement so the x-axis and z-axis was taken from the accelerometer. The y-axis in mouse command was replaced by z-axis. However, things not simple as it think, the properties of z-axis is different than x-axis and y-axis. This property makes it hard to do the coding. When the test was made, the z-axis work properly even it was not smooth so do the x-axis.

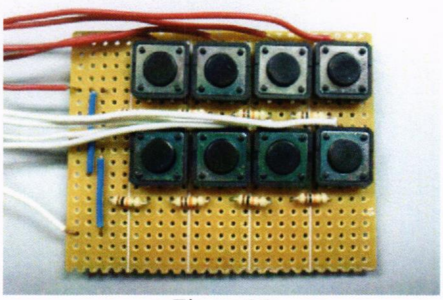

Figure 16

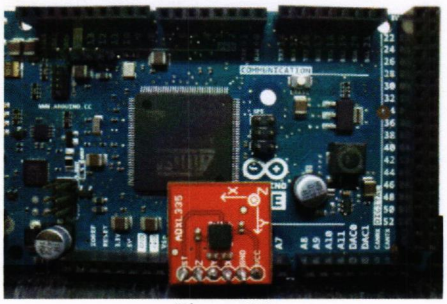

Figure 17

Figure 16 shows the circuit of push buttons that used as a keyboard. When working with the push button, the delay must be inserted in the coding or else it will not work properly. Figure 17 shows the Arduino Due with the accelerometer ADXL335 mounted on it for testing purpose. Figure 18 shows the model of the device. All the components were planted into this model.

Figure 18

Table 13 shows the summary of the results.

Subsystem	A test conducted	Result
The Push Buttons	The push button will work as a keyboard and mouse clicks when it HIGH and do nothing when it LOW.	It worked as planned.
The 2-axis Joystick	The reading of x-axis and y-axis was collected to understand its behavior.	The joystick becomes the mouse pointer replacing the mouse.
The Accelerometer ADXL335	There are three types of reading was taken; the acceleration, gravity and the angle.	The mouse pointer was moved but not like normal and need to improve.

Table 13

VI. CONCLUSIONS

In conclusions, all of the objectives were successfully achieved except the accelerometer was not working smoothly as planned. The fact that the behavior of the accelerometer was different between the axis make it hard to do the coding. In addition, for future development, this project can be upgraded by working with android environment and working with Smartphones.

REFERENCES

[1] http://datenform.de/fpseng.html Nov. 18th 2012

[2] http://www.complex.com/video-games/2011/02/the-50-best-first-person-shooters-of-all-time/#gallery Nov. 18th 2012

[3] www.theesa.com/facts/pdfs/ESA_EF_2011.pdf Nov. 18th 2012

[4] http://www.nintendo.com/wii/what-is-wii#/controls Nov. 19th 2012

[5] http://www.esrb.org/about/video-game-industry-statistics.jsp Nov. 19th 2012

[6] www.aoa.org/documents/EffectsComputerUse.pdf Nov. 19th 2012

[7] http://lifehacker.com/5879536/how-sitting-all-day-is-damaging-your-body-and-how-you-can-counteract-it Nov. 19th 2012

[8] Seungbae Lee, Gi-Joon Nam, Junseok Chae, Hanseup Kim, and Alan J. Drake, "Two-Dimensional Position Detection System with MEMS Accelerometer for MOUSE Applications", Department of EECS, University of Michigan

[9] Arghya Dev Biswas Aryan, Md. Fahmid Wasof, Md. Mahmudul Hasan, "Design and Implementation Of An Accelerometer Based Wireless Mouse", Department of Electrical and Electronics Engineering BRAC University Dhaka, Bangladesh, December 2011