EP 315 Report

Group - Ashwin Hegde, Vishal MV, Gautam Reddy

Initial Proposal

Our initial proposal was to construct an air mouse which detects the tilt w.r.t vertical and moves the cursor appropriately. The information from the accelerometer to the arduino board is sent through a RF transmitter. The arduino performs basic processes and conveys this information to the computer through a serial port. System functions on the computer use this information to perform mouse-like functions.

Stage-1

In the first stage, we tested each component separately. We required a dual axis accelerometer which detects acceleration of the order of a 'g'. We zeroed in on the MMA7361L 3-axis accelerometer since we felt the third axis can be used for scrolling purposes. The accelerometer gives analog values in the range 0.8V to 2.3V for each axis. We tested the accelerometer by reading it through the analog pins of the Arduino. The Arduino has an in-built ADC which showed values between 150 and 500. We used these values to calibrate the mouse movements accordingly.

Stage-2

In the next stage, we tested the switches used for left and right clicks. We used the interrupt function of the arduino to detect a click. The interrupt is called when the corresponding pin detects a change in the logic value. The first change is interpreted as a mouse press and the second change is a release. We could not implement the scroll function through the accelerometer due to unreliability. These were implemented using two more switches. Pressing it stops all other mouse functions and goes into a loop until it is released.

Stage-3

In the third stage, the coding part of the project was done. The Arduino code reads two analog pins(x and y) and four digital pins (2 clicks , scroll up and scroll down) with a given sampling time. It then combines this data and generates a string with 4 bytes reserved each for the x and y values from the accelerometer and 1 byte each for the digital values. The Java code uses the RXTX library to access the Serial port. It implements a `listener' which waits for Serial data to be available. The code then reads it and decodes the string to get back each quantity. Then, we use the Robot class to generate mouse events on the computer based on the data collected from the Serial port.

Problems Encountered

1. The switches were found to be unstable initially. The interrupts detected change when the switches weren't being pressed. Solution: This problem was solved when the switches were soldered

2. We couldn’t use the arduino to detect only rising and falling edges as it was giving random values. Solution: We coded the interrupt to detect change and used the fact that the first change is always a rise.

3. Sometimes, the arduino detected only the rising but not the falling edge. This could pose a serious problem when translated into a mouse function and can lead to the system hanging and other problems. Solution : We assumed that each click lasts at most for a second. If the mouse press lasted longer than one second, we automatically released it. The drag function of the mouse cannot be used due to this.

4. The zero level value of the accelerometer kept changing due to unknown factors. Solution: We subtracted the zero level value from this to obtain relative tilt. To remove this problem, we constantly updated the zero level value.

5. The COM port was not being detected. Solution:The code was modified so that it checks all the COM ports.

Final results and suggestions for improvements

The switches worked individually but were slightly glitchy when the accelerometers was connected to the system. The scrolls worked perfectly. Mouse motion was reasonable.

Improvements: A few algorithms can be implemented to remove noise. The calibration with accelerometer readings can be improved. The optimum sampling rate can be found out with trial and error. The entire device is portable and can be made compact easily. Wireless communication can be implemented to make a Kinect/Wii Remote kind of device.

Arduino Code

```
volatile int rise[2]= {0,0},fall[2]={0,0},stat[2]={1,1},flag[2] = {1,1} ;

//number of rises and falls

volatile int left = 0, right = 0; // left and right click

volatile int tiltx , tilty, n = 1000000 , m = 0;

char str[13]; //string for encoding data and sending through serial port

int lx = 340, ly = 325; //zero level value

float x = 0, y = 0;

int count = 0;


void setup()

{
  Serial.begin(9600);
  attachInterrupt(1,Count,CHANGE);
  attachInterrupt(0, Count1, CHANGE);
  str[12]='\0';
}

void loop()
{
  if(flag[0]==0) // to prevent multiple click detection
  {
    delay(100);
    flag[0]=1;
  }
  if(flag[1]==0)// to prevent multiple click detection
  {
    delay(100);
    flag[1]=1;
  }
  m++;
  if(stat[0] == 1 && stat[1] == 1) m = 0 ; //to prevent problem 3(see above)
  if(m > 20){    //to prevent problem 3(see above)
    if(stat[0] == 2){
      fall[0]++;
      left = 2;
      stat[0] = 1 ;
      m=0;
      flag[0] = 0 ;
    }
  }
```

```
while(digitalRead(6)){ //scroll
  delay(50);
  str[0] = (lx/1000) + '0';
  str[1] = (lx/100) % 10 + '0';
  str[2] = (lx/10) % 10 + '0';
  str[3] = (lx%10) + '0';
  str[4] = (ly/1000) + '0';
  str[5] = (ly/100) % 10 + '0';
  str[6] = (ly/10) % 10 + '0';
  str[7] = (ly%10) +'0';
  str[10] = '1';
  str[11] = '0';
  str[8] = str[9] = '0';
  Serial.print(str);
}
while(digitalRead(7)){ //scroll
  delay(50);
  str[0] = (lx/1000) + '0';
  str[1] = (lx/100) % 10 + '0';
  str[2] = (lx/10) % 10 + '0';
  str[3] = (lx%10) + '0';
  str[4] = (ly/1000) + '0';
  str[5] = (ly/100) % 10 + '0';
  str[6] = (ly/10) % 10 + '0';
  str[7] = (ly%10) +'0';
  str[10] = '0';
  str[11] = '1';
  str[8] = str[9] = '0';
  Serial.print(str);
```

```
 }
 x += analogRead(4);  //accelerometer x
 delay(25);
 y += analogRead(5); //accelerometer y
 count++;
 if(count == 3)
 {
   tiltx = x/3;
   tilty = y/3;     // to control the variation
   str[0] = (tiltx/1000) + '0';    //encoding the string
   str[1] = (tiltx/100) % 10 + '0';
   str[2] = (tiltx/10) % 10 + '0';
   str[3] = (tiltx%10) + '0';
   str[4] = (tilty/1000) + '0';
   str[5] = (tilty/100) % 10 + '0';
   str[6] = (tilty/10) % 10 + '0';
   str[7] = (tilty%10) +'0';
   str[10] = '0';
   str[11] = '0';
   str[8] = left + '0';
   str[9] = right + '0';
   Serial.println(str);
   x = y = 0;
   count = 0;
 }
 left = right = 0; //reset
 delay(25); // sampling rate = 25millis


}
```

```
void Count() //interrupt functions
{
  if(flag[0])
  {
    if(stat[0]==1)
    {
      rise[0]++;
      left = 1;
      stat[0]=2;
    }
    else
    {
      fall[0]++;
      left = 2;
      stat[0]=1;
    }
    flag[0]=0;
  }
}

void Count1() //interrupt functions
{
  if(flag[1])
  {
    if(stat[1]==1)
    {
      rise[1]++;
      right = 1;
      stat[1]=2;
    }
    else
    {
      fall[1]++;
      stat[1]=1;
    }
    flag[1]=0;
  }
}
```

Java code

```
import java.awt.Robot;
import java.awt.event.InputEvent;
import java.io.InputStream;
import java.io.OutputStream;
import gnu.io.CommPortIdentifier;
import gnu.io.SerialPort;
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;
import java.util.Enumeration;

public class SerialTest implements SerialPortEventListener {
	SerialPort serialPort;
	/* x, y -> decides the x and y coordinates of the mouse pointer
	other variables are self-explanatory */
	int x, y, leftclick, rightclick, scrollup, scrolldown, xpos, ypos;
	int drag, limitx, limity, scrollspeed, lx, ly;
	//drag decides the sensitivity. limitx and limity are tolerance levels.
```

```
    //lx and ly are the zero level values.


   /** The port we're normally going to use. */
    private static final String PORT_NAMES[] = {
                    "/dev/tty.usbserial-A9007UX1", // Mac OS X
                    "/dev/ttyUSB0", // Linux
                    "/dev/ttyUSB1",
                    "/dev/ttyUSB2",
                    "/dev/ttyUSB3",
                    "COM3", // Windows
                    };
    /** Buffered input stream from the port */
    private InputStream input;
    /** The output stream to the port */
    private OutputStream output;
    /** Milliseconds to block while waiting for port open */
    private static final int TIME_OUT = 2000;
    /** Default bits per second for COM port. */
    private static final int DATA_RATE = 19200;


    public void initialize() {
            CommPortIdentifier portId = null;
            Enumeration portEnum = CommPortIdentifier.getPortIdentifiers();


            // iterate through, looking for the port
            while (portEnum.hasMoreElements()) {
                    CommPortIdentifier currPortId = (CommPortIdentifier)
portEnum.nextElement();
                    for (String portName : PORT_NAMES) {
                            if (currPortId.getName().equals(portName)) {
```

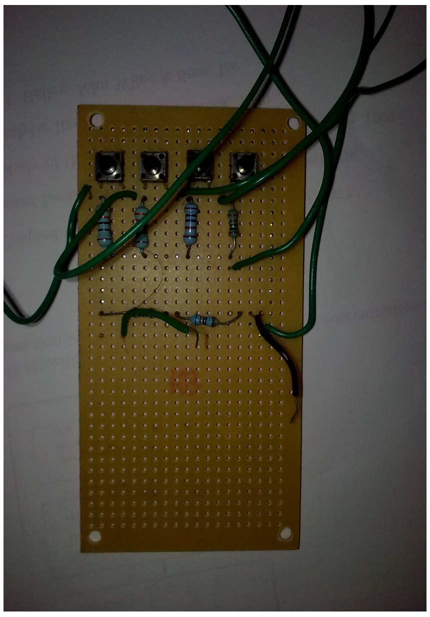
```
                        portId = currPortId;
                        break;
                    }
                }
            }

            if (portId == null) {
                System.out.println("Could not find COM port.");
                return;
            }

            try {
                // open serial port, and use class name for the appName.
                serialPort = (SerialPort)
portId.open(this.getClass().getName(),
                        TIME_OUT);

                // set port parameters
                serialPort.setSerialPortParams(DATA_RATE,
                        SerialPort.DATABITS_8,
                        SerialPort.STOPBITS_1,
                        SerialPort.PARITY_NONE);

                // open the streams
                input = serialPort.getInputStream();
                output = serialPort.getOutputStream();

                // add event listeners
                serialPort.addEventListener(this);
                serialPort.notifyOnDataAvailable(true);
```

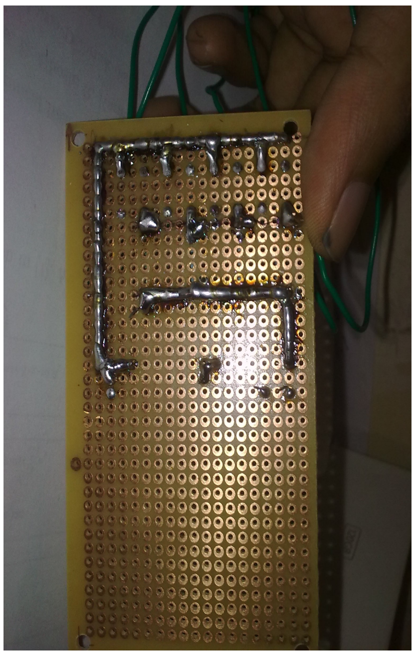
```
		} catch (Exception e) {
			System.err.println(e.toString());
		}
		/*setting initial values*/
		x =  y =  leftclick = rightclick = scrollup = scrolldown = 0;
		xpos = 683; ypos = 384;
		limitx = 60;
		limity = 60;
		scrollspeed = 5;
		lx = 340;
		ly = 325;
		drag = 5;
	}

	/**
	 * This should be called when you stop using the port.
	 * This will prevent port locking on platforms like Linux.
	 */
	public synchronized void close() {
		if (serialPort != null) {
			serialPort.removeEventListener();
			serialPort.close();
		}
	}

	/**
	 * Handle an event on the serial port. Read the data and print it.
	 */
	public synchronized void serialEvent(SerialPortEvent oEvent) {
```

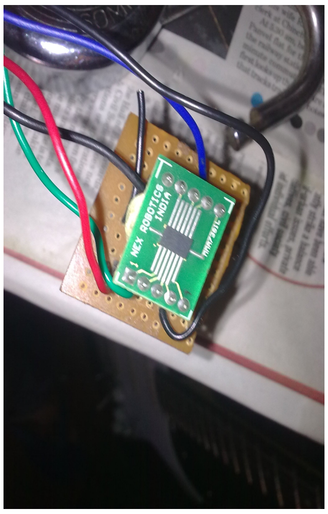
```
if (oEvent.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
        try {
                Thread.sleep(20);
                int available = input.available();
                byte chunk[] = new byte[available];
                input.read(chunk, 0, available);

                //Process the serial data, decoding the string
                x = y = 0;
                for(int i = 0; i < 4; i++)
                        x = x*10 + (chunk[i] - '0');
                for(int i = 0; i < 4; i++)
                        y = y*10 + (chunk[i+4] - '0');
                leftclick = chunk[8] - '0';
                rightclick = chunk[9] - '0';
                scrollup = chunk[10] - '0';
                scrolldown = chunk[11] - '0';
                /*tolerance and drag execution*/
                if(x - lx <= limitx && x - lx >= -limitx)
                        x = 0;
                else
                        x = (x - lx)/drag;
                if(y - ly <= limity && y - ly >= -limity)
                        y = 0;
                else
                        y = (y - ly)/drag;

                System.out.println(x);
                System.out.println(y);
```

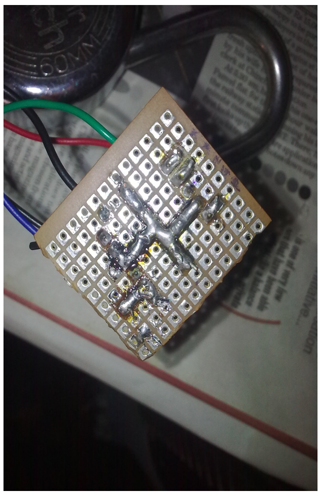
```
//Execute mouse events
Robot robo = new Robot();
xpos += x;
ypos += y;

/*screen resolution*/
if(xpos > 1366)
        xpos=1366;
if(ypos > 768)
        ypos = 768;
//movement
robo.mouseMove(xpos, ypos);
//leftclick
if(leftclick == 1)
        robo.mousePress(InputEvent.BUTTON1_MASK);
else if(leftclick == 2)
        robo.mouseRelease(InputEvent.BUTTON1_MASK);
//rightclick
if(rightclick == 1)
{
        robo.mousePress(InputEvent.BUTTON3_MASK);

        robo.mouseRelease(InputEvent.BUTTON3_MASK);
}
//scroll
if(scrollup == 1)
        robo.mouseWheel(-scrollspeed);
if(scrolldown == 1)
        robo.mouseWheel(scrollspeed);
```

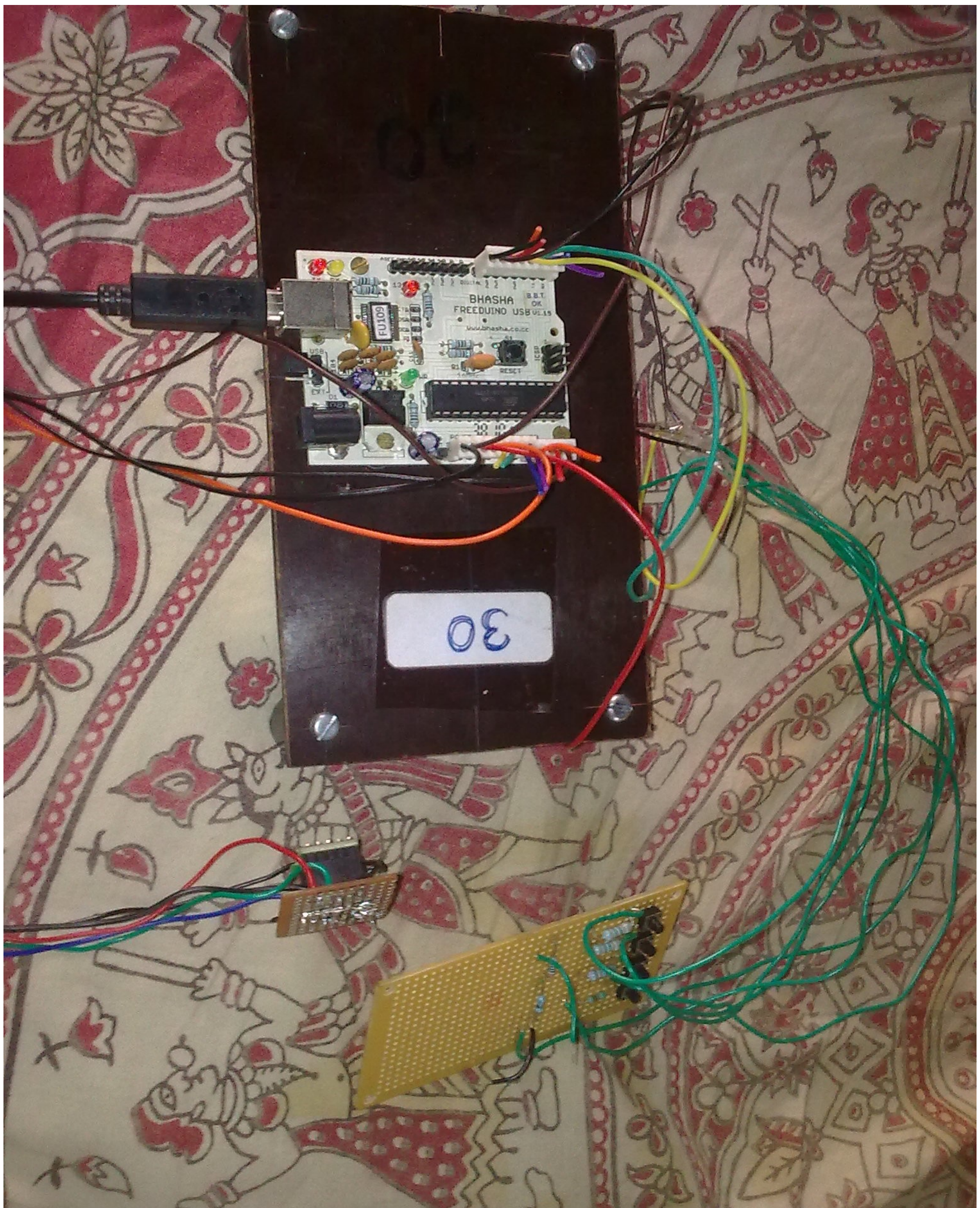
```
                        // Displayed results are codepage dependent
                        System.out.print(new String(chunk));
                    } catch (Exception e) {
                        System.err.println("---");
                    }
                }
                // Ignore all the other eventTypes, but you should consider the other
ones.
        }

        public static void main(String[] args) throws Exception {
                SerialTest main = new SerialTest();
                main.initialize();
                System.out.println("Started");
        }
}
```






NEX ROBOTICS INDIA




BHASHA
FREEDUINO USB
www.bhasha.co.cc
RESET
FU109
30