Islamic University of Technology

CSE 4510

Software Development

Project Report

Mouse Pointer Control Through Hand Orientation

Date of Submission: 10.05.2019

Group: B1

Submitted By:

Name	ID
Mohammad Ishrak Abedin	160041051
Saad Bin Ashraf	160041068
Rizvi Ahmed	160041076

Contents

1. Project Overview

A Hand mounted mouse is a device that uses hand movement to navigate the mouse pointer and fingertips to generate clicks. Its objective is to make the mouse portable and removing the hindrance of having a surface below a typical mouse to move the mouse pointer around the screen.

2. Motivation of the project

The main goal of our project is to build a light weight, mobile, cheap and simple system that would allow one to control mouse pointer movement in a computer using hand orientation and perform clicks using finger taps. We had to achieve this by using gyroscopes, accelerometer and physical wires.

Though technologies exist for mouse pointer movement using hand gestures, most of them use sophisticated sensors, motion detection through IR or even works by tracking muscle movements. However, these technologies are quite expensive and out of general people's reach. Though some cheap technologies exist that perform cursor movement by tracking hand gestures through camera, they are neither accurate nor suitable for persistent use. Moreover, in darker environments, they are completely incapable of performing what they are supposed to do.

We were able to overcome these problems by developing such a technology, that will solely depend on the orientation of hand for cursor movement. We have put a very light weight glove-like exoskeleton on hand that houses one gyroscope and accelerometer. Small wires have been fit on the contact points of finger when they are completely wrapped up. The positions can be modified and are subject to user preference and capability. By calibrating and tracking the orientation of gyroscope, it is possible to measure the orientation of hand accurately. Physical wires are used to simulate clicks.

The main motivation for us behind developing this system is its flexibility and versatile use cases. Main among them is the fact that even with a very weak wrist or hand, this system can be operated easily since it will be light weight. It makes it ideal for people with paralysis or other physical disabilities. Second one is its portability and mobility. Since the whole device will be resting on hand and communicating wirelessly, it does not require any kind of surface

to be operated and allows the user to move while operating the device as long as the user is within connection range. It makes it perfect for presentations or showcasing where the user needs to operate using mouse. Finally, it can also be used for casual cases, like watching movies in a computer from distance and controlling it just by hand orientation.

3. Scope of the Work

3.a. Project Area

The Project covers working with hardwares such as Arduino nano, MPU 6050 housing gyroscope and accelerometer and some basic wires to implement click function.

In the software side, we worked with Arduino IDE to take input from the MPU and the wires. We used C# to use the input to map it to the mouse in the computer.

3.b. Work Planning

In the first phase, we wished to develop the system using Arduino modules and sensors. Since Arduino offers vast range of modules and it has a large user base and library, it was easier for us to debug and develop the very initial project. It also provided us the opportunity to develop the driver software and test run at an early phase. We used Arduino Nano to keep the device light weight and small even in the first phase. Once we were done designing the prototype hardware and software, it was planned to be converted into a single circuit unit by designing proper circuit boards with ICs and sensors.

4. Project Users

The main stakeholders of this project are normal people who wishes to perform day to day computer activities more conveniently. Since this mouse is portable and requires no surface, it can be used not being close or attached to any table. It is useful for giving presentations or drawing from a distance from the computer. Also people with weak hands or physical disabilities may find it very useful to use this as a mouse. It is lightweight, hand mounted, easy to use, so it'll take very little time getting used to and then can be used for smooth experience.

5. UML Diagrams

Dataflow Diagram:

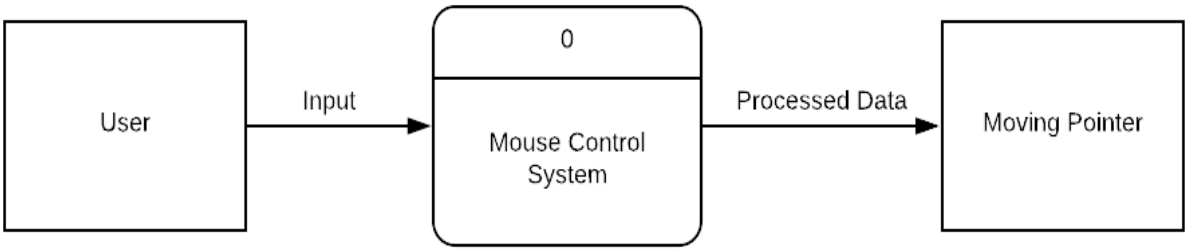

Figure # 5.1: Context Diagram

Context Diagram:

Our whole system takes a main input from user and based on that input, moves mouse pointer.

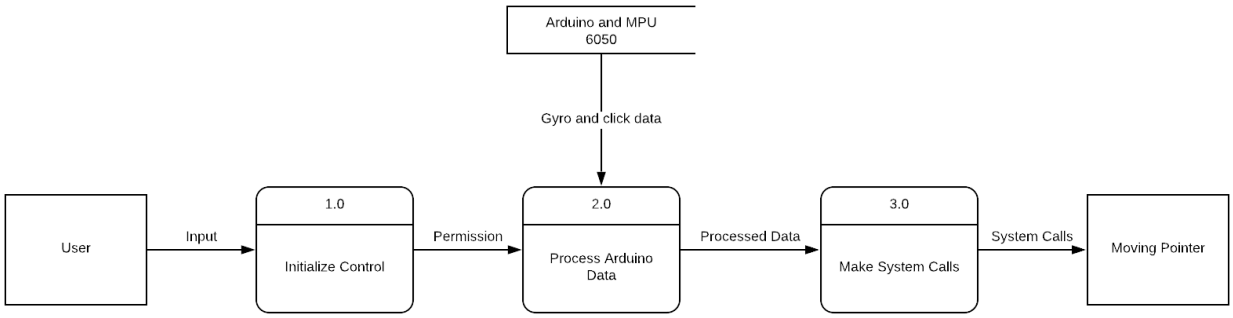

Figure # 5.2: Diagram 0

Diagram 0:

First the user initializes the device using the driver. Once the connection has been properly established, the controller takes gyroscope and click data from Arduino and MPU 6050 and analyzing it, makes system call to move the mouse pointer.

Class Diagram:

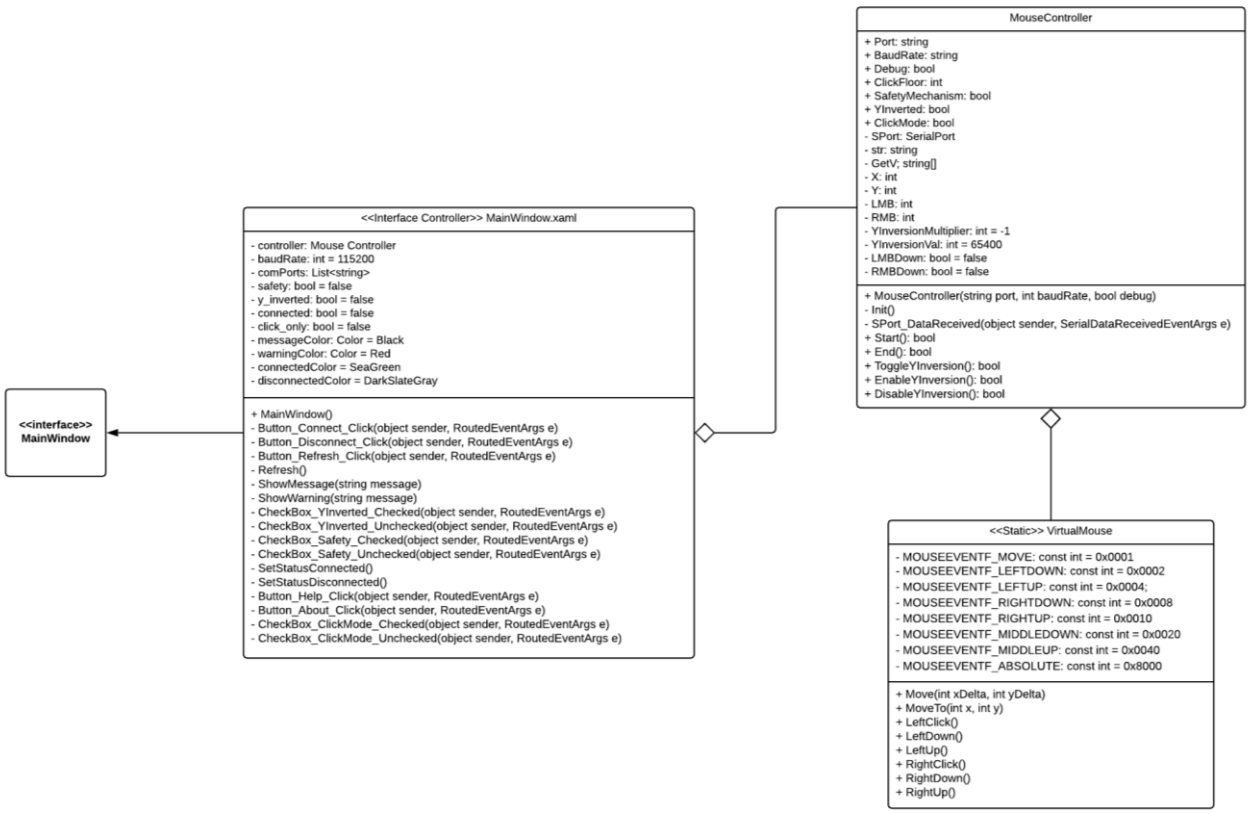

Figure # 5.3: Class Diagram

The driver consists of 3 main classes along with the interface. MainWindow.xaml is the controller class for interface. It accommodates user interaction and provides a way to enable/disable the device and toggle options. This controller class aggregates MouseController class, which is the main controller class for the mouse manipulation. It receives events from serial port and by analyzing the data from Arduino, it provokes methods of VirtualMouse class, which it aggregates. VirtualMouse class serves as a host for dynamic link library which makes system calls based on the provoked methods to control mouse pointer and perform clicks.

Swim Lane Diagram:

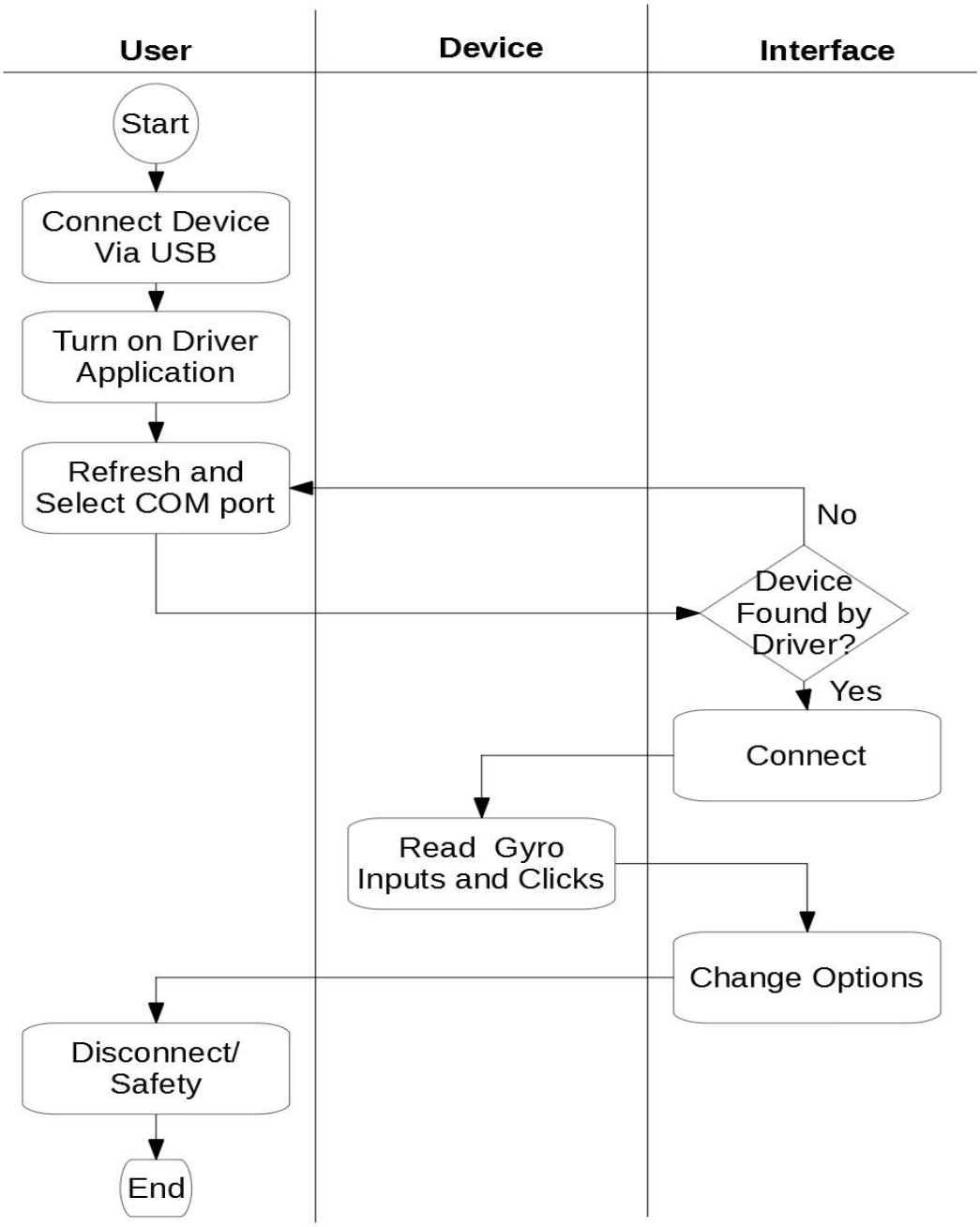

Figure # 5.4: Swim lane Diagram

Swim lane diagram shows how the active events take place in the system. User starts the flow of command by first connecting Arduino in the USB port and then opening the driver application and selecting the COM port. If the device is found, it is connected and data from device is sent to the

controller to control mouse pointer and perform clicks. Meanwhile, user can change options for the device, like inverting y axis movement or toggling between click or drag mode using the user interface. Once the user is done, user terminates the connection by disconnecting the device.

Sequence Diagram:

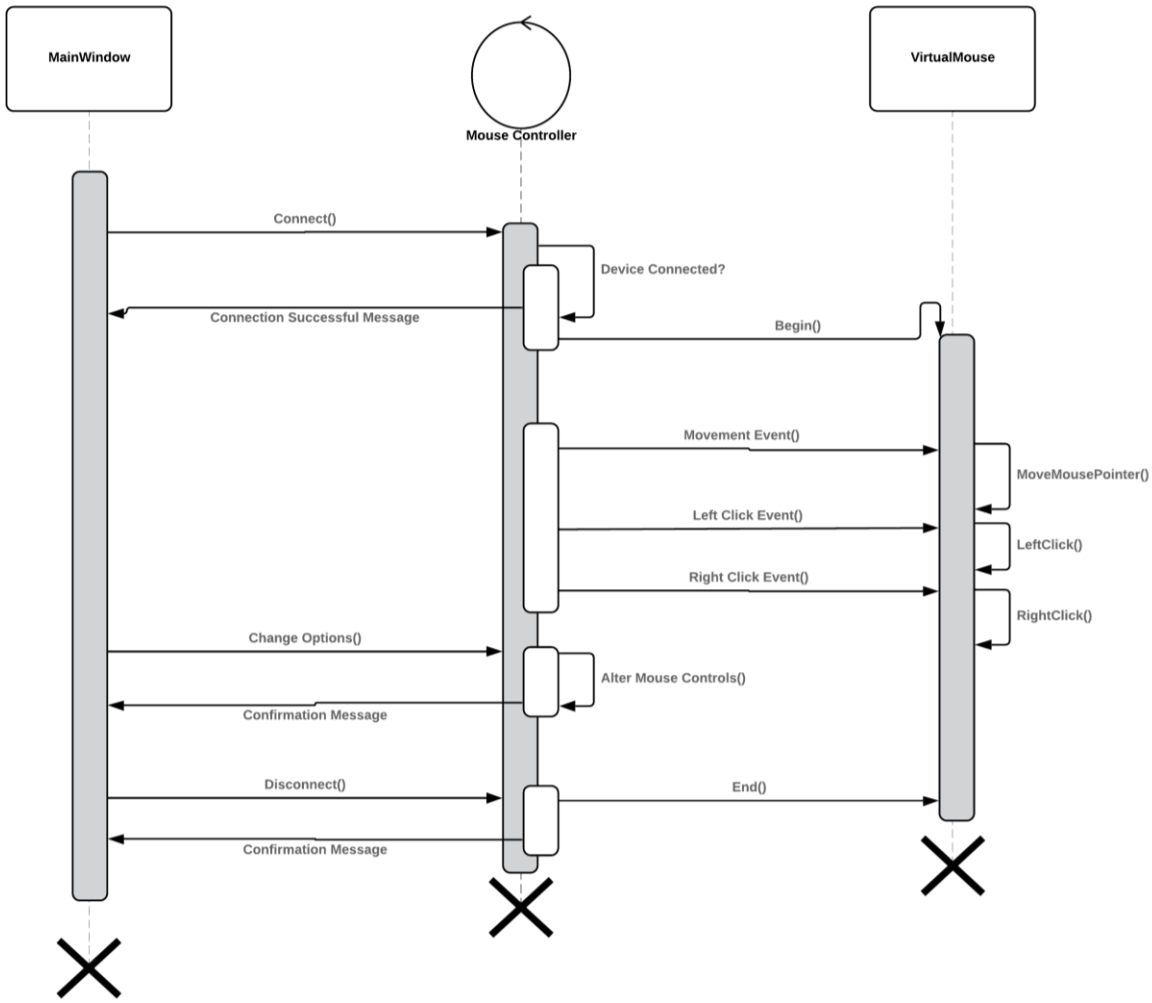

Figure # 5.5: Sequence Diagrams

Sequence diagram depicts the interaction sequence between different classes. After Main Window has initialized a connect request, Mouse Controller checks whether the device is

connected or not, and if connected, sends a confirmation message back to window and initializes interaction with Virtual Mouse. Based on the data from Arduino, Mouse controller performs mouse movement and clicks with

the help of virtual mouse. Mouse controller also manages option settings which the user can change by issuing change option commands from the Main Window, which in term will call Mouse Controller. When user wants to end the session, a Disconnect request is sent from Main Window, which in terms, by processing, Mouse Controller ends the connection with Virtual Mouse. It also sends a confirmation message back to Main Window.

6. Requirements

Functional Requirements:

- Imitate mouse usage by using a hand mounted gloves.
- Tilt and rotate the hand to get mouse pointer movement.
- Touching the index and the middle finger to the thumb to generate mouse clicks.
- Perform other activities such as dragging or double clicks by using click and movement together or double tapping.

Non Functional Requirements:

Performance:

We have achieved optimal performance by using C# to process the raw data input we got from the gyroscope. The data was taken and passed using Arduino IDE and then processed to map it to get mouse position. We had to take running average of inputs to filter out unwanted jitters so that we could get smooth movement. And because the driver is optimized, the performance was smooth and lag free.

Portability:

This device is connected to the computer via a USB cable. We intend to achieve portability later by adding a Bluetooth module in the device and connecting it remotely via Bluetooth. Then we would be able to send data and process it without being physically attached.

We also want to achieve cross platform usability by creating drivers for different operating systems.

7. Prototypes and Design

7.a. Prototypes and sketches

Our product is still in prototype phase. It can be seen as a fully operational 'Selected Functionality' prototype. We have implemented only the basic left and right clicking and dragging mechanism with mouse pointer movement using the gyroscope. Due to lack of time, we are still away from implementing the wireless and enabling/disabling feature for the device.

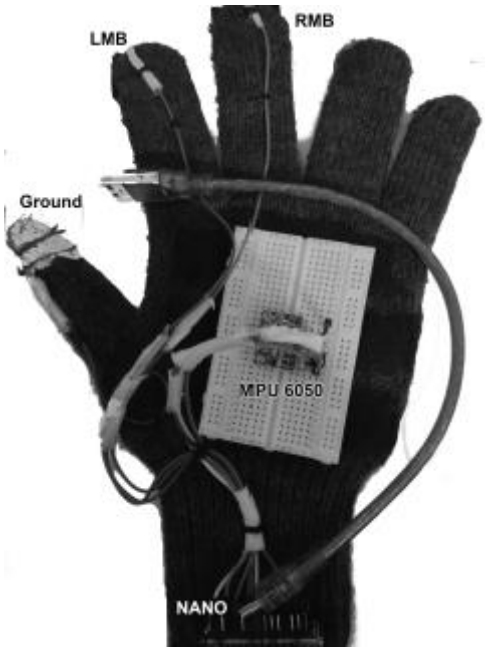

Figure # 7.a.1: Device Setup

Sketch: The back end or the code for the Arduino is written in Arduino's native language which is a derivative of C/C++. MPU 6050 has an onboard DMP or Digital Motion Processor. By providing timely interrupts to the gyro-accelero combo MPU 6050, we can retrieve the gyroscope processed data by the DMP.

We took last 10 running average of DMP data to smooth out the gyroscope reading and remove jitters. Click is currently implemented by manipulating the random noise of analogue pins in Arduino. When any pin is open in Arduino, it provides a random value between 0 to1023 which continuously flickers. So, last 10 values can only be 0 if the pin is grounded. Based on this nature, by taking running sum of last 10 readings, we implemented click. To increase contact with the ground, the ground wire on top of thumb is wrapped around in aluminum foil to increase conductible surface. Same cannot be done with the LMB and RMB wires since it will increase their noise picking region and make them capacitive sensors. For this reason, LMB and RMB are kept as far apart as possible. The sketch diagram is given below.

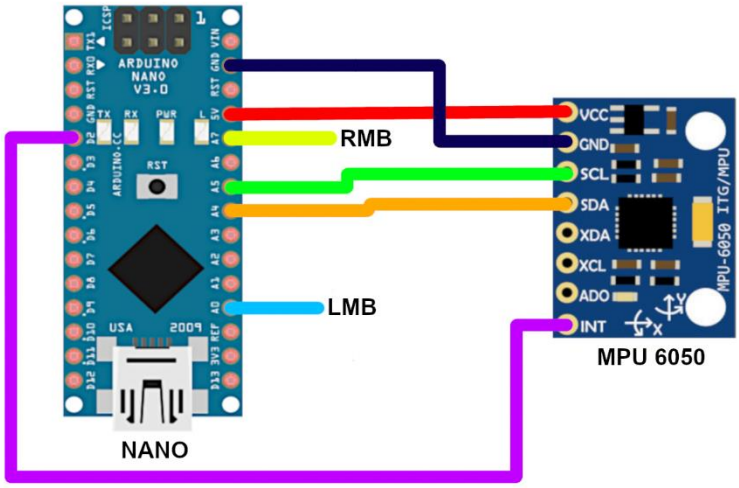

Figure #7.a.2: Pin diagram for Arduino Nano and MPU 6050.

7.b. UI Designs

The whole front end of the project is done using DotNet framework 4.6.1. The UI is designed using XAML or eXtensible Application markup language. It is connected to front end codes using Visual C#. By taking data from gyroscope and Arduino, front end controls mouse by sending system calls through a dll(dynamic link library). A full working detail can be found in the documentation of code attached at the end of the project. A figure with different components of the UI is given below.

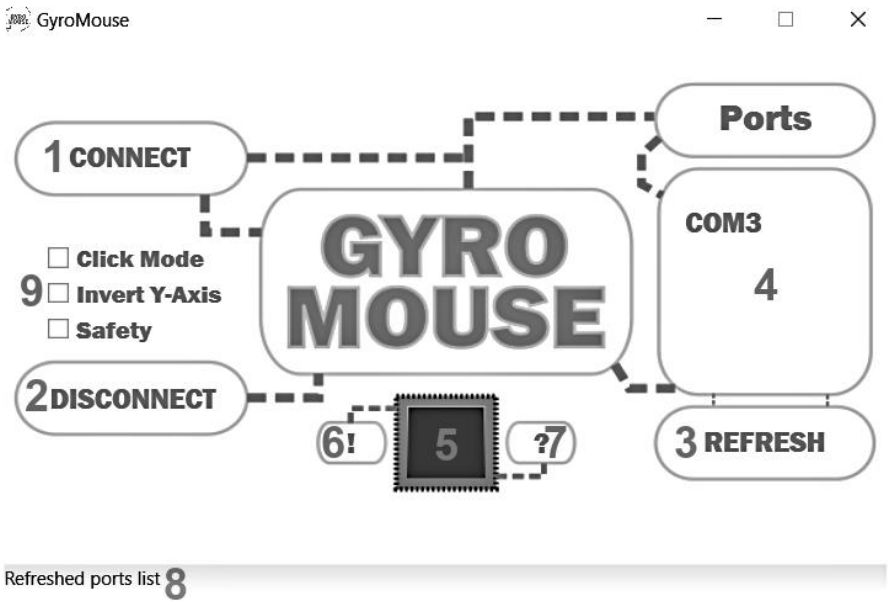

Figure #7.b.1: UI Components

Component Description:

1. Connect: This button is used to establish connection to the selected COM port in COM port list marked as 4. It also starts the mouse control.
2. Disconnect: This button is used to disconnect current connection and stop mouse control using the gyroscope.
3. Refresh: This button is used to refresh the COM port list marked as 4.

4. COM port list: After starting the application, this list will auto fetch all available COM ports. If the device is connected after the application has been started, ‘Refresh’ button can be pressed to manually refresh the COM port list. A valid COM port needs to be selected from here before pressing the ‘Connect’ button.
5. Indicator: The indicator turns Sea Green color when a valid connection has been established. Slate Gray color indicates that there is currently no connection.
6. Help: This button shows a brief description about how to operate the driver application.
7. About: This button shows the list of developers.
8. Output Log: This section is used to show messages and warnings. Messages appear black while warnings are shown in red.
9. Option check boxes: This check boxes are used to toggle various control options.
    - Click Mode: If checked, left action in device will perform only click and disable dragging.
    - Invert Y Axis: If checked, the Y axis mapping of mouse pointer with controller will be inverted.
    - Safety: Built for debug purpose. While checked, moving the mouse pointer to the top left corner of the screen will force disconnect the controller.

8. Proposed System Architecture

8.a. Architectural Block Diagram

The following diagram gives an overall overview of how our device and software works between different layers of abstraction.

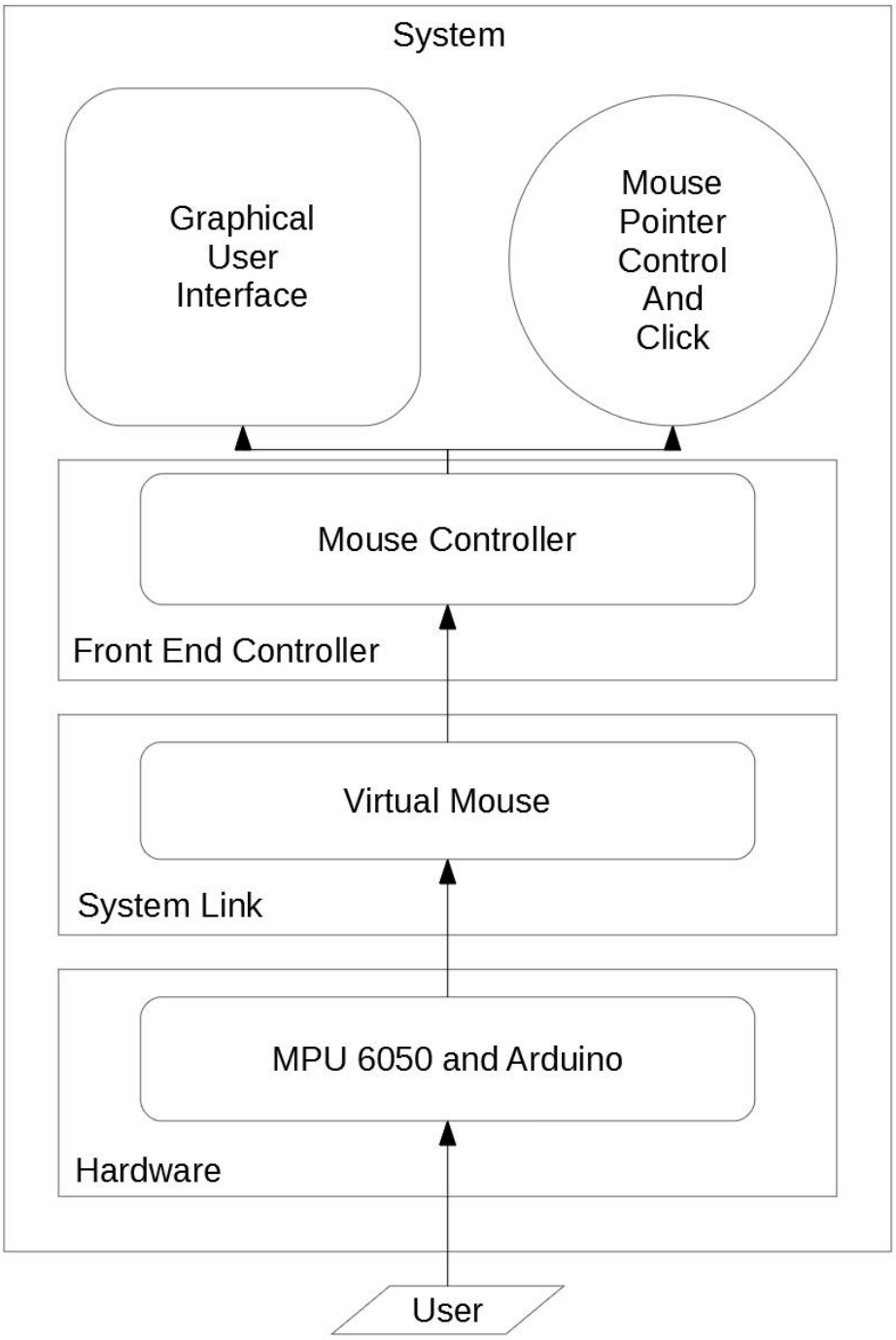

Figure # 8.a.1: Architectural Block Diagram

In hardware layer, Arduino and MPU 6050 works in tandem to send gyroscope and click data to the computer. In the system layer, Virtual Mouse class, using dynamic link library, performs system calls to move the mouse and perform clicks. The Virtual Mouse class is managed by MouseController class which, in fact, is a 'facade' class and simplifies the inputs for Virtual Mouse class. And at the top, sits the Interface which the user uses to interact with the driver and control mouse pointer.

8.b. Software Process Model Followed

For the development of our controller and its driver, we followed spiral modeling. We worked in increments and in each increment, we had certain goals that we had to meet. Due to shortage of time, our final product is still in prototype phase though many features were added during the increments. In our first increment, we added just movement readings using MPU 6050 gyroscope and Arduino Nano. In our second increment, we developed the clicking and dragging mechanism. Third increment was building the front end UI and driver application. We took some time here due our problem with PyAutoGUI and finally switched to Visual C# and DotNet which solved our problem. In our fourth increment, we mounted the device on a simple glove, fine tuned the values and tested the usability of the product. We are currently standing at the end of fourth increment.

9. Software Test, Planning and Evaluation

Due to shortage of time, rigorous and controlled testing of the software could not be carried out. Primary tests of the hardware and related software were carried out by the developers to remove bugs and hardware related issues. In future, we plan to continue this project and in that case, we wish to add more features, switch the whole circuit to a PCB and then we can carry out full fledged testing.

10. Conclusion and Future Work

At this moment the whole system is built based on Arduino nano and MPU 6050 combined together, making it somewhat bigger and less convenient. Building a PCB consisting only the necessary things and laying it flat(omitting the vertical pins and soldering the connections) on the hand would make it very sleek and lot less pronounced.

Also this device is not wireless. Using a module or replacing nano with Arduino Leonardo (or similar boards), we can add wireless functionalities to it.

We intended to make a mouse which is more user friendly and not conventional to achieve portability. The lack of necessity of using a surface to operate a mouse would give us a great opportunity at attaching one of the most used device to our hands and reduce the need of more space. It would give us immense portability and enables us to use devices like tablets, or minimalistic desktops easily.