# **CHAPTER 1**

# **INTRODUCTION**

The Development of information technology is increasing day by day in our society. A computer system is very essential and important part of the human life. Because of the information technology the time taken for all the environmental things is very less. The user interface of the personal computer has developed from a text-based command line to a graphical interface with keyboard as well as mouse are taken as a inputs. But for human being they are inconvenient and also they are very much unnatural. So to overcome these drawback some attractive alternative has to be taken place. The use of hand gestures provides an attractive alternative to these interface devices for human computer interaction.

Gesture recognition has been a research area which received much attention from many research communities such as human computer interaction and image processing. The increase in human-machine interactions in our daily lives has made user interface technology progressively more important. Physical gestures as intuitive expressions will greatly ease the interaction process and enable humans to more naturally command computers or machines.

Human computer interaction (HCI) also referred as Man-Machine Interaction (MMI) refers to the relation between the human operator and the computer or rather specifically the machine. Since the role of human operator is very significant in operating the machine, it is important that the two main characteristics should be considered when designing a HCI system i.e., the functionality and usability. Here system functionality refers to the set of functions or services that the system equips to the users, while system usability refers to the level and scope that the system can operate and perform specific user purposes efficiently. Any system that achieves a suitable balance between these concepts can be considered as influential performance and powerful system.

A gesture controlled air mouse using an accelerometer is one kind of mouse which can be operated by the movement of hand by placing an accelerometer on it. The system uses accelerometer to detect the users hand tilt in order to direct mouse movement on the monitor. The change in the acceleration values of the accelerometer are transmitted to the PC, where in the software applications take control and moves the mouse cursor. It is very much compact and easy to use. More advantage is that its direction and motion can be controlled just by hand movement. The use of accelerometer makes this project compact and enables it to move effectively in 4 directions in PC.Cheap components makes it more accessible for the users. Though it consists of less components circuitry is easier and thus troubleshooting becomes easier.

**1.1 PROBLEM STATEMENT**

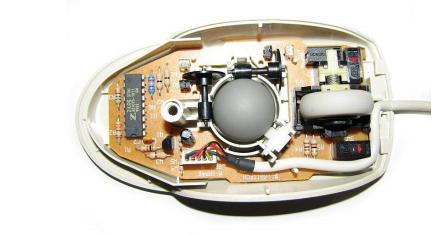
Computer usage is at an all-time high in India and many people use computer mice on a daily basis for extended periods of time. It is common that every device has its own disadvantages, especially computer devices. Analysing of various type mouse has been and the problems are generalized as follows. The below points describes the normal problems that present physical mouse suffers:

* Optical mouse may subject to wear and tear.
* Many hardware components are require for optical mouse.
* Optical mouse are not adjustable to all environments and varies according to the environment.
* All wired mouse have its own lifespan and requires large surface to operate.
  1. **EXISTING SYSTEM**

There exist many types of physical computer mouse in the modern technology. The following section will discuss the types and differences about the physical mouse need to operate only on flat surface.

**Mechanical Mouse**

The other name of this mechanical mouse is trackball mouse and is mostly used in 1990s.There ball which is there inside is separated by two rotating rollers to detect movement made by it. One of the rollers is used for detecting left/right motion and the other roller is for detecting forward/backward motion. The ball is made of steel and covered with a fine layer of rubber. The purpose of using rubber is to make the detection more precise. The three main functions performed by that mouse are left click, right click and scrolling. Because of the friction between the mouse ball and the rollers itself, the mouse are prone to degradation. The overtime usage of the mouse may cause rollers to degrade, thus making it unable to detect the motion correctly. Even though if switches are replaced, they also cause mechanics within to be loosed and no longer will detect any click.



**Fig 1.2.1: Mechanical Mouse with Top Cover Removed**

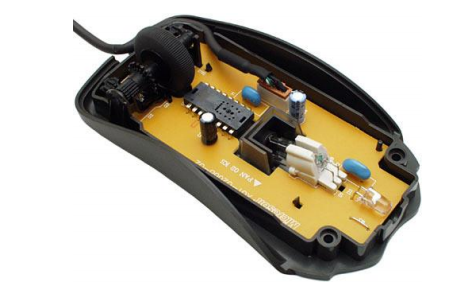
The following table describes the advantages and disadvantages of the Mechanical Mouse.

**Table 1.2.1: Advantages and Disadvantages of Mechanical Mouse**

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * User can control the system by moving the mouse. * Provides precise mouse tracking movements. | * Prone to degradation of the mouse rollers and button switches, causing to be faulty. * There is a need of long flat surface. |

**Optical and Laser Mouse**

A mouse that commonly used in these days. The motions of optical mouse depend on the Light Emitting Diodes (LEDs) which are used to detect movements relative to the underlying surface, while the laser mouse is an optical mouse that uses coherent laser lights. Comparing it with mechanical mouse, the optical mouse no longer depend on the rollers to determine its movement, instead it uses an imaging array of photodiodes. The purpose of implementing this optical and laser light is to eliminate the limitations of degradation that plagues the current predecessor, giving it more durability while offers better resolution and precision. Besides all these advantages there exist some disadvantages. It is unable to detect its motion on the polished surface. And also long term usage without a proper cleaning or maintenance may leads to dust particles trap between the LEDs, which will cause both optical and laser mouse having surface detection difficulties. There is still a chance that the mouse may prone to degradation of the button switches, which will make the mouse to function improperly unless it is disassembled and repaired.



**Fig 1.2.2**. **Optical Mouse with top Removed**

The following table describes the advantages and disadvantages of the optical and laser mouse.

**Table 1.2.2: Advantages and Disadvantages of Optical and Laser Mouse**

|  |  |
| --- | --- |
| Advantage | Disadvantage |
| * Gives better precision with less hand movements. * Supports longer life span. | * Prone to button switches degradation. * Function improperly on a polished surface. |

* 1. **PROPSED SYSTEM**

In the proposed system, a physical mouse will be developed which works in the air. If a person moves the mouse in air then it will work, thereby no need of using the surface, overcoming the problems of mouse not working on rough surfaces, and reduces the space needed to operate mouse.

The air mouse will be able to perform three main functions. They are

1. Left click
2. Right click
3. Moving the cursor

The implementation is done using a sensor named accelerometer to sense the hand gestures. Accelerometer is a motion sensor which sense changes in motion in any of the three axes. In here the accelerometer will be at the user side attached to the breadboard.

# **CHAPTER 2**

# **LITERATURE REVIEW**

**2.1 PRESENT WORK:**

` Optical mouse have less range, can operate under certain distance only. The distance is nothing but the length of their connecting cable and require a surface to work on. Wireless mouse is no more different than Optical mouse except that it doesn’t require a wire to operate, but still needs a surface to operate. HUMAN COMPUTING INTRACTION (HCI) is one of the important and emerging area of research were people try to improve the computer technology by increasing the interaction between Human and Computer. Gesture is a very natural human communication capability. A unique feature of the gesture communication channel is that it allows one to act on one’s environment as well as to retrieve information from it. With the improvement in technology, it became common to interact with devices by using natural gestures. The idea that natural, comfortable motions can be used to control computers is opening the way to a host of the input devices that look and feel very different from the keyboard and mouse. The advancement of technology in the field of sensors made it possible to design a humanoid for any application. Efforts are being made to reduce the gap between a human and a machine.

# **CHAPTER 3**

# **SYSTEM ANALYSIS**

**3.1 FUNCTIONAL REQUIREMENTS**

* The services of Gesture controlled mouse are useful for all the tech savvy’s who work more on computers.
* Services of Gesture controlled mouse are useful in designing, device control and gaming.

**3.2 NONFUNCTIONAL REQUIREMENTS**

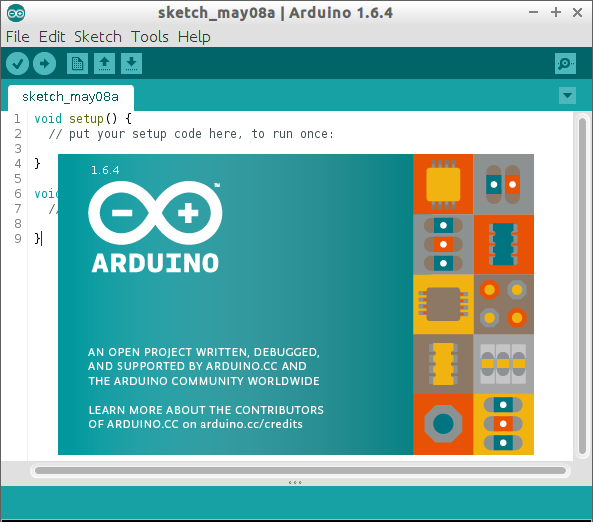
* **Scalability :** Scalability is a non-functional property of a system that describes the ability to appropriately handle increasing (and decreasing) workloads. Our gesture based mouse is capable of working continuously without any problem regarding stuck.
* **Maintainability:** Maintainability is the measure of ability to successfully repair or fix the product after manufacturing, usually in the field, and over time. Even though a repair occurs we can easily find it and fix it.

**3.3 SOFTWARE REQUIREMENTS:**

**Arduino:**

Arduino consists of both software, and IDE (Integrated Development Environment) that runs on your computer, and a physical programmable circuit board (often referred to as a microcontroller) used to write and upload computer code to the physical board.Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and a microcontroller kits for building digital devices and interactive objects which sense and control objects in the physical world. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the Arduino board. All the operating system Windows, Mac OS, and Linux support Arduino software. The code in this Arduino is written in simple Java based on Processing and other open-source software.

 This software can be used with any Arduino board like Arduino UNO, pro micro, Leonardo, Duemilanove, Nano, Macro etc...  The Arduino platform has become quite popular for people working in the field of electronics. No special devices are required to upload the code into the Arduino boards, a single USB cable is sufficient to upload it. And the software uses simplified version of C++, making users to learn programming easily and quickly.

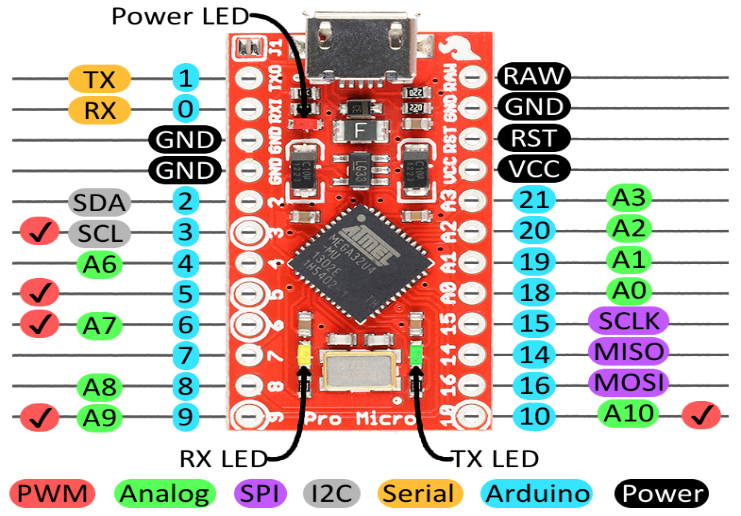


**Fig 3.3.1: Arduino Software**

* 1. **HARDWARE REQUIREMENTS**

**Arduino Pro Micro:**

Arduino Pro Micro is one of the Arduino boards available from the Arduino Company. The Pin out layout for Arduino Pro micro is as follows. All of the Pro Micro’s I/O and power pins are broken out to two, parallel headers. Some pins are for power input or output, other pins are dedicated I/O pins. Further, the I/O pins can have special abilities, like analog input. Here’s a map of which pin is where, and what special hardware functions it may have:



**Fig 3.4.1: Pinlayout of Arduino Pro Micro**

Delving a little further into which pins do what.

**Power Pins**

There are a variety of power and power-related nets broken out:

* **RAW** is the unregulated voltage input for the Pro Micro. If the board is powered via USB, the voltage at this pin will be about 4.8V. On the other hand, if the board is powered externally, through this pin, the applied voltage can be up to 12V.
* **VCC** is the voltage supplied to the on-board ATmega32U4. This voltage will depend on whether you’re using a 3.3V/8MHz Pro Micro or a 5V/16MHz version, it’ll be either 3.3V or 5V respectively. This voltage is regulated by the voltage applied to the RAW pin. If the board is powered through the ‘RAW’ pin (or USB), this pin can be used as an output to supply other devices.
* **RST**can be used to restart the Pro Micro. This pin is pulled high by a 10k&Ohm; resistor on the board, and is active-low, so it must be connected to ground to initiate a reset. The Pro Micro will remain “off” until the reset line is pulled back to high.
* **GND,** of course, is the common, ground voltage (0V reference) for the system.

**I/O Pins**

The Pro Micro’s I/O pins – 18 in all – are multi-talented. Every pin can be used as a digital input or output, for blinking LEDs or reading button presses. These pins are referenced in the Arduino IDE via an integer value between 0 and 21. (The A0-A3 pins can be referenced digitally using either their analog or digital pin number).

Nine pins feature analog to digital converters (ADCs) and can be used as analog inputs. These are useful for reading potentiometers or other analog devices using the analog Read ([pin]) function.

There are five pins with pulse width modulation (PWM) functionality, which allows for a form of analog inputs using the analog Write ([pin], [value]) function. These pins are indicated on-board with a faint, white circle around them.

There are hardware UART (serial), I2C, and SPI pins available as well. These can be used to interface with digital devices like serial LCDs, XBees, IMUs, and other serial sensors.

The Pro Micro has five external interrupts, which allow you to instantly trigger a function when a pin goes either high or low (or both). If you attach an interrupt to an interrupt-enabled pin, you’ll need to know the specific interrupt that pin triggers: pin 3 maps to interrupt 0, pin 2 is interrupt 1, pin 0 is interrupt 2, pin 1 is interrupt 3, and pin 7 is interrupt 4.

**On-Board LEDs**

There are three LEDs on the Pro Micro. One red LED indicates whether power is present.

The other two LEDs help indicate when data is transferring over USB. A yellow LED represents USB data coming into (RX) the Pro Micro, and a green LED indicates USB data going out (TX).

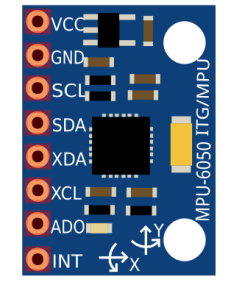
**3.3V or 5V? 8MHz or 16MHz**

Pro Micros come in two flavours, which vary by system voltage and operating frequency. The standard 5V Pro Micro runs at 16MHz, and is very comparable to an  Arduino Leonardo, while the 3.3V version of the Pro Micro runs at half the speed (to remain in the safe operating zone at the lower voltage) – 8MHz.

The operating voltage of your Pro Micro determines the maximum allowable voltage on any of the I/O pins. For example, if you have a 3.3V Pro Micro, don’t interface it with something that outputs 5V.

**MPU 6050:**

MPU 6050 is one of the IMU (Inertial Moment Unit) sensors available. Among other IMU sensors like ADXL 345 accelerometer, ITG 3200 gyroscope, Sparkfun 6 DOF IMU sensor , MPU 6050 is the most reliable and accurate IMU sensor. Apart from being significantly cheaper. MPU 6050 sensor help us get the position of an object attached to the sensor in three-dimensional space. These values are usually in angles to help us to determine its position. They are used to detect the orientation of smartphones, or in wearable gadgets like the Fitbit.

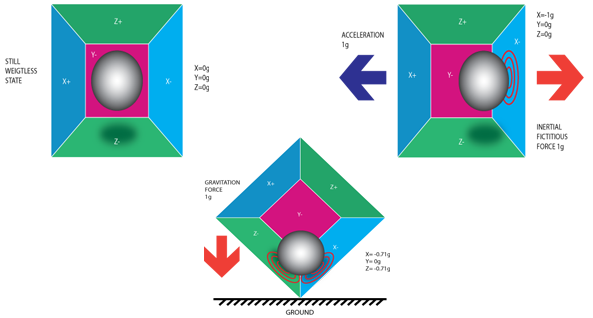


**Fig 3.4.2: MPU 6050 Pins**

The MPU 6050 is a 6 DOF (degrees of freedom) or a six-axis IMU sensor, which means that it gives six values as output: three values from the accelerometer and three from the gyroscope. The MPU 6050 is a sensor based on MEMS (micro electro mechanical systems) technology. Both the accelerometer and the gyroscope are embedded inside a single chip. This chip uses I2C (inter-integrated circuit) protocol for communication. To know the working of the MPU 6050 we should know the working of Accelerometer and Gyroscope.

**Accelerometer Working Principle**

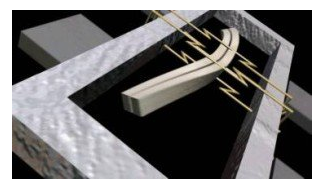
An accelerometer works on the principle of the piezoelectric effect. Imagine a cuboidal box with a small ball inside it, like in the picture above. The walls of this box are made with piezoelectric crystals. Whenever you tilt the box, the ball is forced to move in the direction of the inclination due to gravity. The wall that the ball collides with creates tiny piezoelectric currents. There are three pairs of opposite walls in a cuboid. Each pair corresponds to an axis in 3D space: X, Y, and Z axes. Depending on the current produced from the piezoelectric walls, we can determine the direction of inclination and its magnitude.



**Fig 3.4.3: Piezo Electric Accelerometer**

**Gyroscope Working Principle**

Gyroscopes work on the principle of Coriolis acceleration. Imagine that there is a fork-like structure that is in a constant back-and-forth motion. It is held in place using piezoelectric crystals. Whenever you try to tilt this arrangement, the crystals experience a force in the direction of inclination. This is caused as a result of the inertia of the moving fork. The crystals thus produce a current in consensus with the piezoelectric effect, and this current is amplified. The values are then refined by the host microcontroller.

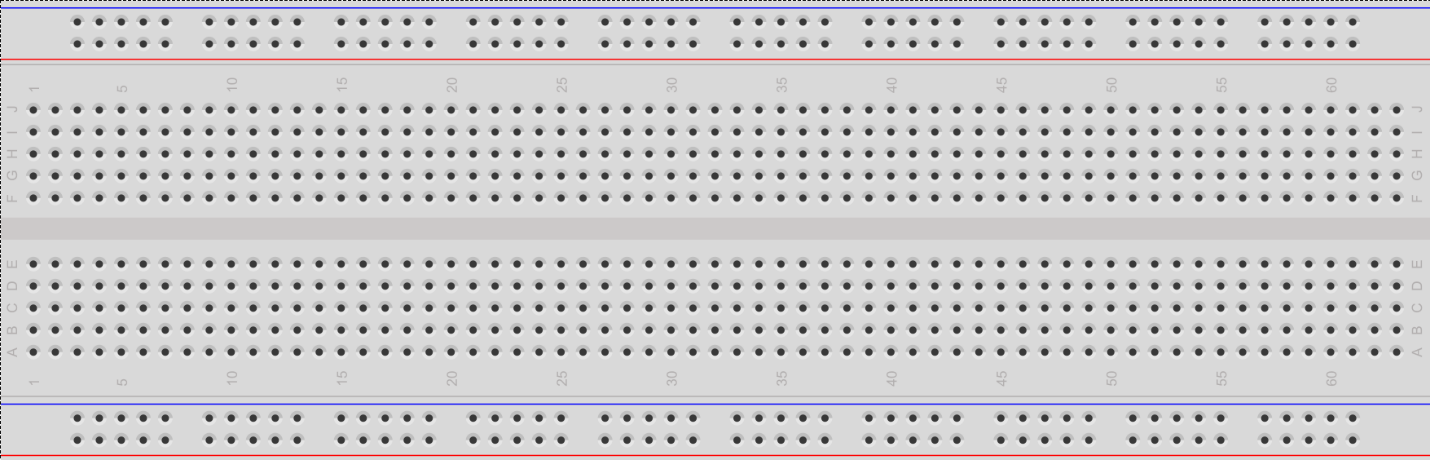


**Fig 3.4.4: Piezo Electric Gyroscope**

**Breadboard:**

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.

The top and bottom rows (the rows indicated by the blue) and are usually the (+) and (-) power supply holes and these move horizontally across the breadboard, while the holes for the components move vertically.

****

**Fig 3.4.5: Breadboard**

**Resistors:**

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit.  Modern resistors are made out of either a carbon, metal, or metal-oxide film, and a thin film of conductive (though still resistive) material is wrapped in a helix around and covered by an insulating material. A resistor works by converting electrical energy into heat, which is dissipated into the air.



**Fig 3.4.6: 10K Ohm Resistor**

For implementing Gesture based mouse control we need two 10k ohm resistors one for left click switch and the other for right click switch.

**Switches:**

A push-button (also spelled pushbutton) or simply button is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. A push to make switch allows electricity to flow between its two contacts when held in. When the button is released, the circuit is broken. This type of switch is also known as a Normally Open (NO) Switch. For implementing the gesture based mouse control using breadboard we need two push button switches each of 12mm square. One switch is used to implement left click operation of a mouse and the other one is used for implementing the right click operation.



**Fig 3.4.7: Push button Switch**

**Jumper cables:**

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont cable) is an electrical wire or group of them in a cable with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.



**Fig 3.4.8: Jumper Cable**

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

**USB cable:**

A USB port is a standard cable connection interface for personal computers and consumer electronics devices. USB stands for Universal Serial Bus, an industry standard for short-distance digital data communications. USB ports allow devices to be connected to each other with and transfer digital data over USB cables. In this project we use USB cable to connect Arduino and the Laptop. One USB cable is sufficient to implement this project.



**Fig 3.4.9: USB cable**

# **CHAPTER 4**

# **SYSTEM DESIGN**

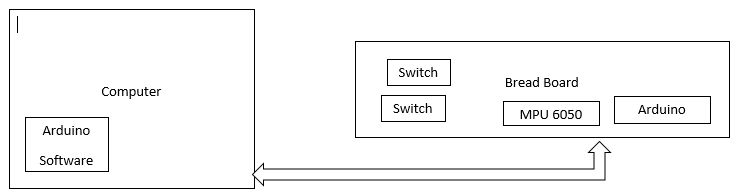
**4.1 INTRODUCTION**

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

It also stresses the importance of system designing phase in the process of system development. System designing in terms of software engineering has its own value and importance in the system development process as a whole.

**4.2 ARCHITECTURE**

**The proposed system architecture uses MPU 6050 as a sensor to detect the motion. MPU 6050 is interfaced to a microcontroller, Arduino Pro Micro in this case. The basic block diagram is as shown in the below figure.**

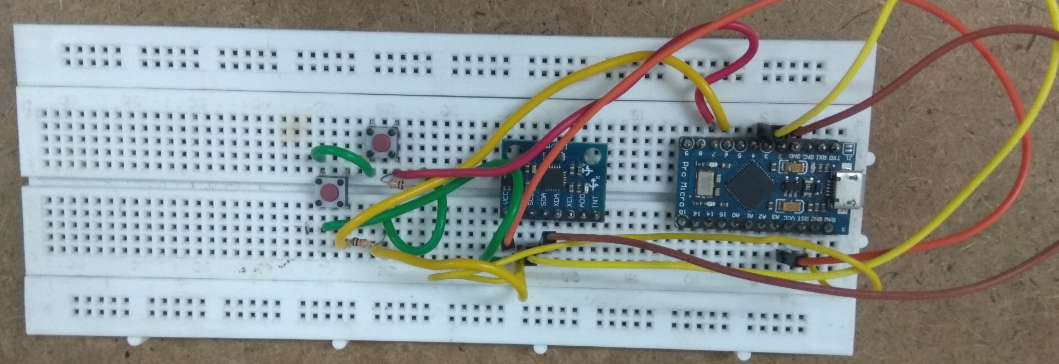


**Fig 4.2.1: Block Diagram for the Architecture**

**The block diagram consist of MPU 6050, Arduino pro micro, Arduino software, breadboard. Switches mainly. And to make the current flow safely resistors are placed in the breadboard. The connection between computer and Breadboard is made through a USB cable.**

**4.3 DESIGN**

**The connections in the bread board between Arduino Pro Micro, MPU 6050, switches should be designed accurately. The GND of sensor is connected to GND of Arduino because we need to supply power to this device. VCC of sensor is connected to VCC of Arduino. The SCL of the MPU6050 is connected to 3rd pin of Arduino Pro Micro. The SDA of MPU6050 is connected to 2nd pin of Arduino. Push button switches are placed in the bread board and resistors are placed in correspondence to the switches. Ground of sensor is connected to both the switches.**



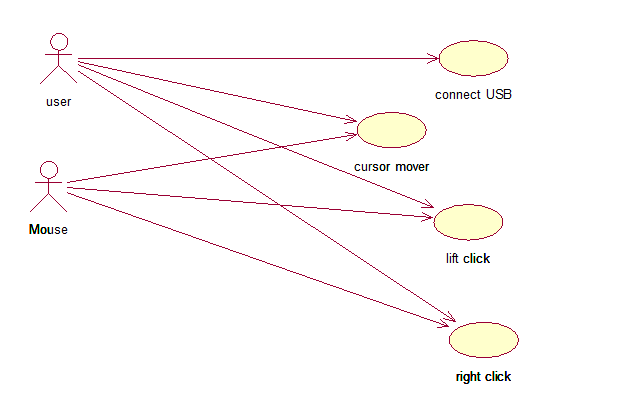
**Fig 4.3.1: Design of the Connections**

**4.4 UML DIAGRAMS**

**The Unified Modelling Language (UML) is a general “purpose modelling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system”. It was created and developed by Grady Booch, Ivar Jackson and James Rum Baugh at Rational Software during 1994-1995 with further development led by the through 1996.**

* **Use-case Diagram:**

**A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.**



**Fig 4.4.1: Use Case diagram**

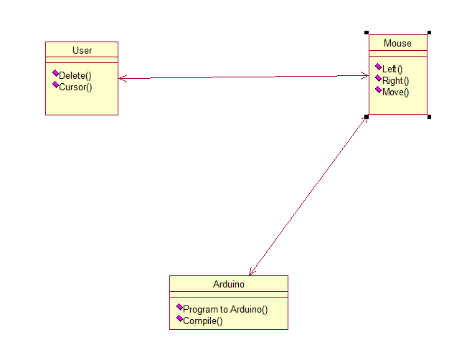
* **Class Diagram:**

**Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.**

**Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.**

**Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.**

**The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.**

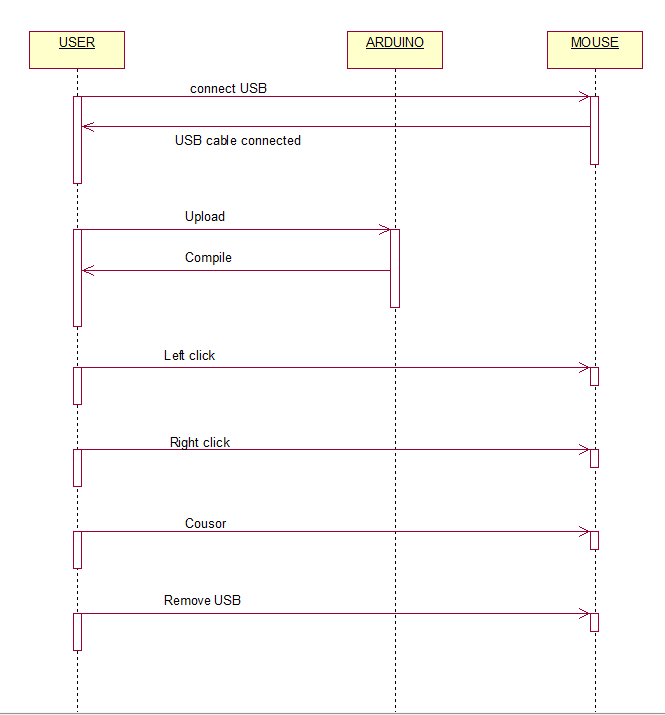
****

**Fig 4.4.2: Class Diagram**

* **Sequence Diagram**

This interactive behaviour is represented in UML by two diagrams known as Sequence diagram and Collaboration diagram. The basic purpose of both the diagrams are similar. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

The sequence diagram has four objects (USER, ARDUINO, and MOUSE)

****

**Fig 4.4.3: Sequence Diagram**

# **CHAPTER 5**

# **IMPLEMENTATION**

**After making all the connections, implementation will be done by connecting all the components and deploying the code into the Arduino board. The code is placed in the Arduino software.**

**CODE:**

**#include <Wire.h>**

**#include <I2Cdev.h>**

**#include <MPU6050.h>**

**#include <Mouse.h>**

**MPU6050 mpu;**

**int16\_t ax, ay, az, gx, gy, gz;**

**int vx, vy;**

**int button1 = 6;**

**int button2 = 7;**

**int buttonState1 = 0;**

**int buttonState2 = 0;**

**void setup()**

**{**

**Serial.begin(9600);**

**Wire.begin();**

**pinMode(button1, INPUT);**

**pinMode(button2, INPUT);**

**mpu.initialize();**

**if (!mpu.testConnection())**

**{**

**while (1);**

**}**

**}**

**void loop()**

**{**

**mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);**

**vx = (gx+15)/150;**

**vy = -(gz-100)/150;**

**Serial.print("gx = ");**

**Serial.print(gx);**

**Serial.print(" | gz = ");**

**Serial.print(gz);**

**Serial.print(" | X = ");**

**Serial.print(vx);**

**Serial.print(" | Y = ");**

**Serial.println(vy);**

**Mouse.move(vx, vy);**

**buttonState1 = digitalRead(button1);**

**buttonState2 = digitalRead(button2);**

**if (buttonState1 == HIGH) {**

**Mouse.press(MOUSE\_LEFT);**

**delay(100);**

**Mouse.release(MOUSE\_LEFT);**

**delay(200);**

**}**

**else if(buttonState2 == HIGH) {**

**Mouse.press(MOUSE\_RIGHT);**

**delay(100);**

**Mouse.release(MOUSE\_RIGHT);**

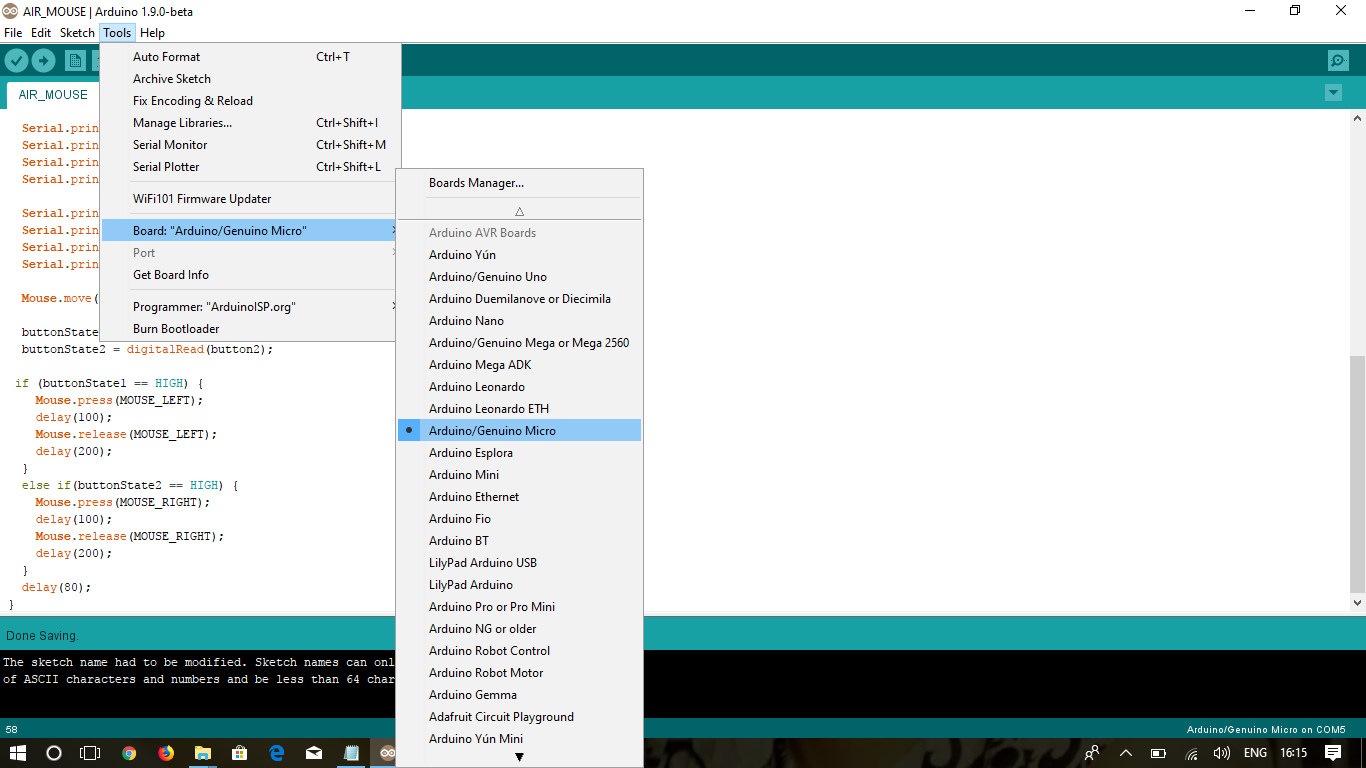
**delay(200);**

**}**

**delay(80);**

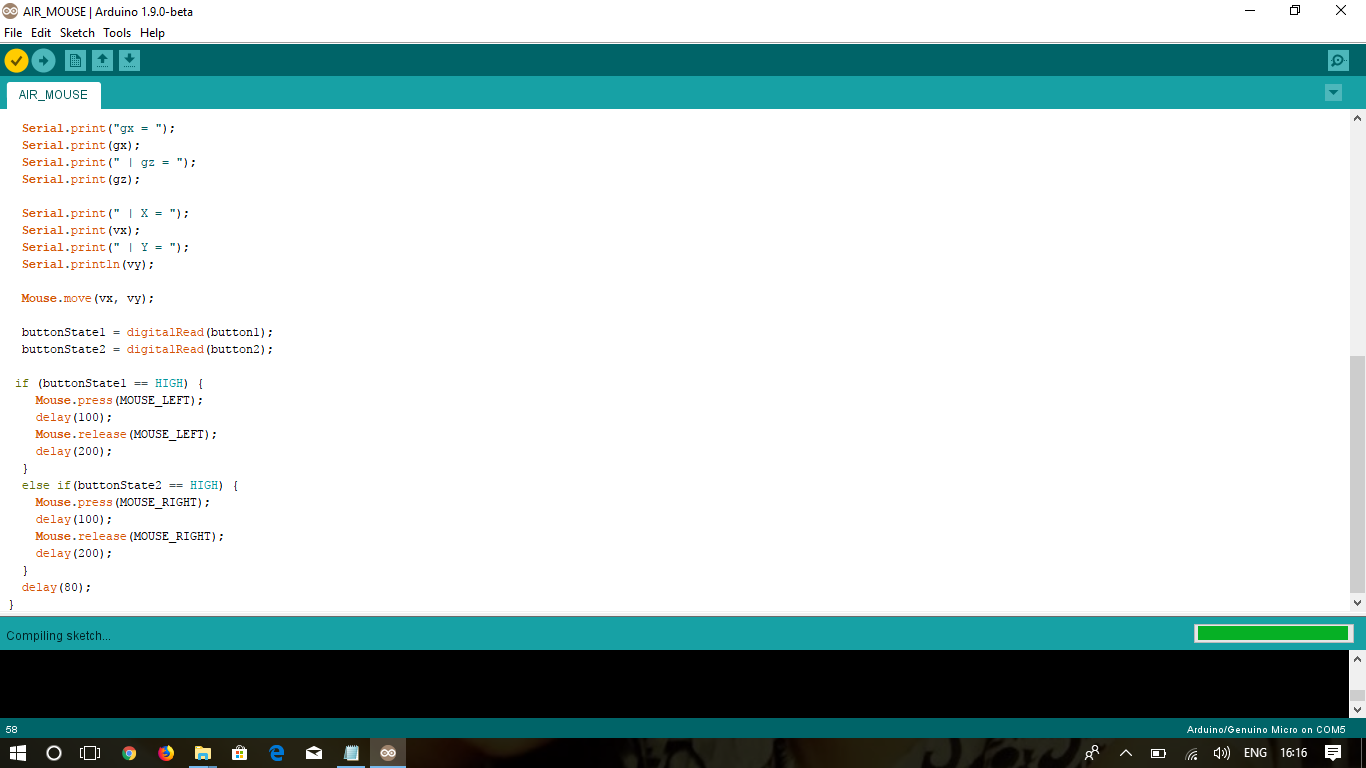
**}**

**After writing the code in the Arduino software we have to change some settings in the software. Go to tools, select Board and select Arduino/Genuino Micro. The following screenshot says its implementation.**



**Fig 5.1: Selecting the Particular Arduino Board**

**Then go the port and select the appropriate port. Later save the code and compile the sketch, if the code is error free it will not show any errors, if there exists errors then compilation errors will be displayed. The screenshot of the compiling will appear as below.**



**Fig 5.2: Picture showing Compiling Sketch**

**If the code is compiled then upload the code into the Arduino board. After doing this connect it through USB cable, it will be implanted and perform all the operations of the mouse. Left click, right click, and cursor movement will work as proposed in the proposed system.**

# **CHAPTER 6**

# **TESTING**

**6.1 Overview of Testing:**

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

Software testing is an investigation conducted to provide stakeholders with information about the [quality](https://en.wikipedia.org/wiki/Software_quality) of the [software](https://en.wikipedia.org/wiki/Software) product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding [software bugs](https://en.wikipedia.org/wiki/Software_bug) (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time,
* is sufficiently usable,
* can be installed and run in its intended [environments](https://en.wikipedia.org/wiki/Operating_environment), and
* Achieves the general result its stakeholder’s desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding [software bugs](https://en.wikipedia.org/wiki/Software_bug) (errors or other defects). The job of testing is an iterative process as when one bug is fixed, it can illuminate other, deeper bugs, or can even create new ones.

Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors.

Software testing can be conducted as soon as executable software (even if partially complete) exists. The [overall approach to software development](https://en.wikipedia.org/wiki/Software_development_process) often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an [agile approach](https://en.wikipedia.org/wiki/Agile_software_development), requirements, programming, and testing are often done concurrently.

Tests can be conducted based on two approaches –

* Functionality testing
* Implementation testing

Software testing methods are traditionally divided into white- and black-box testing.  These two approaches are used to describe the point of view that the tester takes when designing test cases.

**Test Methods:**

* **Black Box Testing:**

Black-box testing is a method of software testing that examines the functionality of an application (what the software does) without going inside the internal structure (White-box Testing). We also have something called Gray-box testing which something is in between. You need no knowledge of how the system is created.

• Black-box testing can be done by a person who only know what the software is supposed to do

• Compare to driving a Car – you don’t need to know how it is built in order to test it.

* **White Box Testing:**

You need to have knowledge of how (Design and Implementation) the system is built. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. While white-box testing can be applied at the [unit](https://en.wikipedia.org/wiki/Unit_testing), [integration](https://en.wikipedia.org/wiki/Integration_testing) and [system](https://en.wikipedia.org/wiki/System_testing) levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

**Testing Levels:**

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified.  Software is tested on various levels:

**1. Unit testing**:

While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

**2. Integration Testing**:

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updating etc.

**3. System Testing**:

System Testing follows Integration Testing.

* It consists of Black-box Tests that validate the entire system against its requirements.
* Checking that a software system meets specifications and that it fulfils its intended purpose.
* Often executed by an independent group (QA group, QA – Quality Assurance).
* Since system tests make sure the requirements are fulfilled, they must systematically validate each requirement in the SRS (Software Requirements Specification).

**4. Acceptance Testing**:

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

# **CHAPTER 7**

# **CONCLUSION**

The aim of the project was to design a Gesture based Mouse Control using Arduino and Accelerometer, which works on the hand gestures and is designed successfully. This paper discusses about the methodologies and algorithms involved in the gesture recognition for Human machine Interaction. Gesture based Mouse is done using Arduino, breadboard and MPU 6050. An accelerometer, and gyroscope are used to sense motion, more particularly acceleration in a given direction. Accelerometer is used as a motion sensor with axis of the accelerometer that is X-axis and Y-axis, which forms a plane of motion, are used to sense. This is a tilt based approach. Change of acceleration or a tilt will have an impact on click and release of the curser of mouse. In the proposed system, the cursor is made to move based on the moment of the accelerometer, not exactly the position. The position based approach can be tried in future, where in the mouse cursor just follows the movement of the finger.

# **CHAPTER 8**

# **FUTURE SCOPE**

# **The future scope of this Gesture Based Mouse is to make the scrolling up and down possible. So that he can do every mouse operation very feasibly. Tilt based approached should be made so perfect that even movement in any angle should be happen accurately. Also adding push buttons to the fingers make it possible to implement left click and right click with just one tap of finger in the air. Another extension we would like to add in the future is that typing the words in the system without using keyword and implementing it by using gesture moments in the air.**

# **CHAPTER 9**

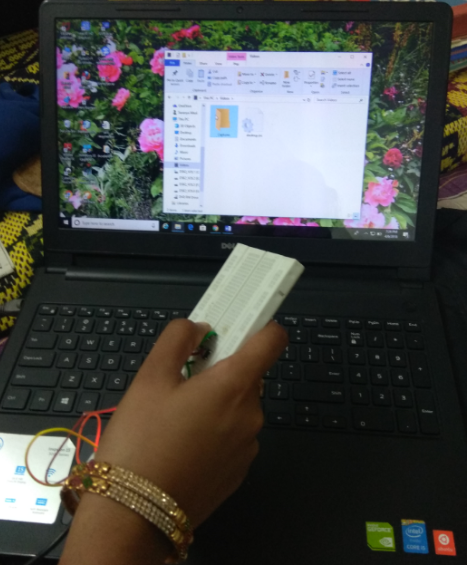
**REFERENCES**

1. Sohel Ahmed, Mohammed Abdullah Zubair, Irshad Basha Shaik “Accelerometer based Wireless Air Mouse using Arduino Micro-controlled board” 2015 Global Conference on Communication Technologies, IEEE.
2. <http://www.mrhobbytronics.com/arduino-accelerometer-mouse/>
3. R.Suriya, Dr.V.Vijayachamundeeswari “A Survey on Hand Gesture Recognition for Mouse Control” at ICICS -2014.
4. .M.M.Maindarkar, Kamthe Netra, Keerthana Reddy “Accelerometer Based Wireless Mouse using Arduino Board” at IJSART - Volume 3 Issue 5 –MAY 2017.

# **CHAPTER 10**

# **APPENDIX**

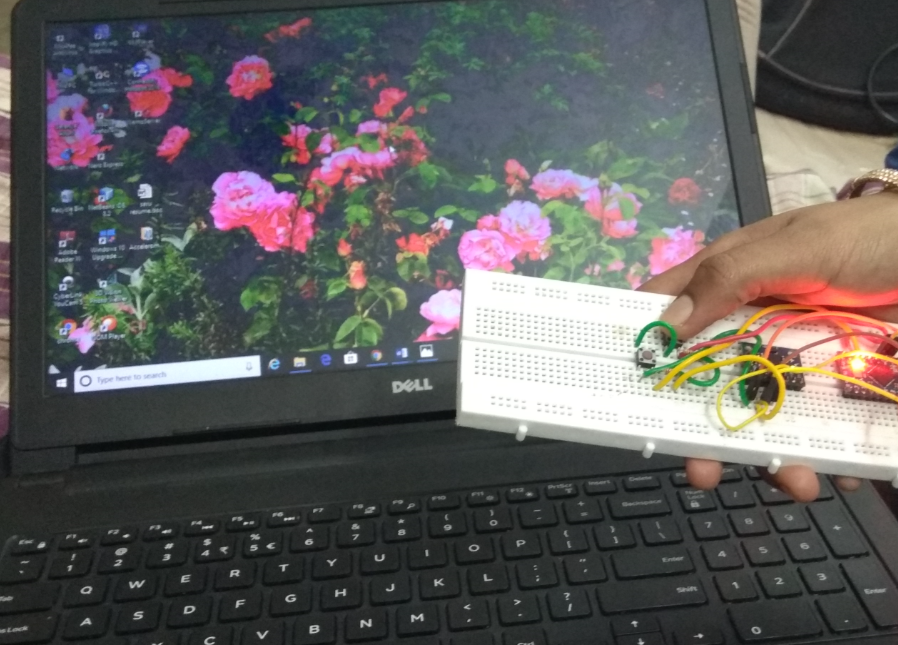
**10.1 Output Screens:**

****

**Fig 10.1.1: Implementation of Left Click**

****

**Fig 10.1.2: Implementation of Right Click**



**Fig 10.1.3: Implementation of Cursor Movement**