3D TOUCHLESS INTERFACE TRACKING USING CAPACITIVE SENSING TECHNOLOGY

Submitted in partial fulfilment of the Degree of

Bachelor of Technology

May – 2015

Sumangal Mangal (111094)

Shruti Grover (111099)

Harshita Solanki (111100)

Name of supervisor - D.S.Saini

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

CERTIFICATE

This is to certify that project report entitled "3D Touchless Interface Tracking Using Capacitive Sensing Technology", submitted by Sumangal Mangal (111094), Shruti Grover(111099), Harshita Solanki(111100) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 25 may, 2015 **Supervisor's Name:** Dr. D.S.Saini

Designation: Associate Professor
Department of Electronics and Communications Technology
Jaypee University Of Information Technology
Waknaghat, Solan

CONTENTS

ACKNOWLEDGEMENT

We take this opportunity to express our sincere thanks and deep gratitude to all those people who extended their wholehearted co-operation and helped us in completing this project successfully. First of all, we would like to thank **Dr. Sunil Bhooshan** (Head of Department, ECE) for creating opportunities .Special thanks to **Dr. D.S Saini**, Project Mentor for all the help and guidance extended to us by him in every stage during our project development. His inspiring suggestions and timely guidance enabled us to perceive the various aspects of the project in a new light. We are highly indebted and grateful for his strict supervision, constant encouragement, inspiration and guidance, which ensure the worthiness of our work.

ABSTRACT

In this project, 3-D technology has been implemented in a cost effective and optimized way which is also simple to understand and experiment with. Combine low-tech materials with some high-tech components and build a completely touchless 3D Tracking Interface.

This has been achieved using a technology called capacitive sensing, which is a way of human touch sensing, that requires little or no force to activate. In this, the sensor plate and our body form a capacitor. The more its capacitance, the more charge it can store. The capacitance of this capacitive touch sensor depends on how close your hand is to the plate.

A 3D workspace is created and connected with Arduino microcontroller which measures the time capacitor (i.e. the touch sensor) takes to charge, giving it an estimate of the capacitance, which in turn transfers this data to Matlab (GUI), which processes this data and shows the corresponding movement using a graphic user interface.

The working of human capacitive sensing using an aluminium foil as a capacitor plate and our hand as another has been shown.

_____	_____
Signature of Students	Signature of Supervisor
Name	Name
Date	Date

LIST OF FIGURES AND TABLES

LIST OF SYMBOLS AND ACRONYMS

SYMBOLS/ACRONYMS	MEANING
GUI	Graphic User Interface
NUI	Natural User Interface
RMS	Root Mean Square
ITO	Indium Tin Oxide
LED	Light Emitting Diode
USB	Universal Serial Bus
ICSP	In-Circuit Serial Programming
PWM	Pulse Width Modulation
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
IDE	Integrated Development Environment

OBJECTIVE

To make 3D touchless hand position tracking system using the concept of capacitive sensing. Ever since the advent of the computer mouse and the GUI based on the Windows and other operating systems next paradigm shift in user interfaces have been explored. Mouse-based GUIs have proven remarkably flexible and robust but a major sea change towards NUIs has been finally seen, not only in the research lab, but also in commercial products aimed at broad consumer audiences. Under the NUI umbrella, there are two broad categories of interfaces: those based on direct touch, such as multi-touch tablets and those based on three-dimensional spatial input such as motion-based games. It is this latter category, which we call three-dimensional user interfaces (3D UIs), that we focus on.

Interface tracking technology detects motion in 3D and requires no special worn-sensors for operation. The system is capable of detecting movements in 3-dimensions without ever having to put your fingers on the screen. By interacting in the line-of-sight of the sensors the motion is detected and interpreted into on-screen movement.

.

CHAPTER 1

INTRODUCTION

With the arrival and gained popularity of hardware that supports Motion Sensing the past few years, there is growing interest in the different ways this technology can be used in. Although the first applications of it is for entertainment purposes (video games) there is now a larger number of scenarios where motion sensing can be employed. These scenarios include the implementation of software in a variety of fields, such as design-related (painting, 3D model design), simulations (for learning or training purposes) and file browsers (e.g. image galleries).

Motion sensing and its derivatives, like gesture recognition, present an alternative way of communicating with software that falls under the umbrella of Touchless Interaction. It is the kind of interaction where the user does not get in touch with any sort of input tool.

In many cases, machines, home appliances and electronic devices have to be controlled by human beings. It is part of everyone's daily life and we are familiar with switches, push buttons, keyboards, knobs and slider controls. Since some time, a new species of control elements invades our life. It started in consumer products like mobile phones and MP3 players but moves into all kind of devices now. Those talks are about touch sensors. Simple electrodes underneath the housing replace mechanical input devices with moving elements. The shape and layout of these sensor electrodes can be designed in a very flexible way, leading to appealing, modern product designs with enhanced usability. A wide range of elements can be implemented with touch sensor electrodes: simple buttons and keyboards, linear or circular sliders, transparent touch elements on displays or even buttons on wooden surfaces. As the sensor electrodes are placed inside the device, no openings are required. The housing is more robust and cost-effective and ideally suited for rough environments where

dust and moisture could creep into the device. Especially for medical applications or devices used in clean environments like in the food industry, capacitive touch control enables hygienic casings.

Conventional mechanical buttons or potentiometers with moving parts have a certain lifetime. Sooner or later they are worn out and do not work reliable anymore. Due to the lack of moving parts, capacitive touch is much more durable. However for highly stressed elements, the surface resp. overlay cover material of the touch electrode has to be considered. Glass or acryl may be better suited as plastics. For single, isolated buttons, even metal can be used, not for complete front covers however as the sensor pads must be isolated from each other.

One positive aspect of the moving parts in conventional push buttons or switches is the tactile feedback to the user. By touching a surface, the user does not "feel" if the push button was triggered. This can be compensated by using optical, acoustical or, a bit more complex, vibration feedback.

- User Interfaces

A UI is the framework of communication between the application and the user and consists of all the elements that allow the user to issue commands and requests to the application. The application in turn, makes use of the interface to organize the content it contains, prompt the user for some action and provide feedback to the user as to how commands and requests are progressing and what their result is. According to the type of UI, different elements might be used to achieve the above or even omit them. User interfaces evolved as part of the broader computing evolution, when advances in hardware and software allowed it. Those advances include the more powerful CPU chips, graphic acceleration capabilities, direct input devices (mouse, keyboard) and the appearance of multi-purpose operating systems (OS) that would make computers fit for a wider audience.

- GUI

GUI are the most popular ones and constitute the primary way of interacting with an application currently. A GUI makes use of visual elements to provoke behavior from the user, respond to that behavior and provide information. Amongst those elements there are icons, buttons, text fields, various types of list boxes, menus and menu items and more. There are also container elements, which organize content and lay it out in a way that is easy to follow, such as panels, group boxes, tabs and other. Another type of elements is that of output ones, which includes labels, text blocks, message boxes, lists views to name a few.

- NUI

NUI is based on design of interactions that feel natural to the human. In that sense, the user and the application communicate in a way that resembles human communication. The aim of such a design model is to make the interface behave in a way that feels real and not iconic. NUI is a relatively broad term since a touch interface can also be a type of NUI. The interface is able to recognize a movement performed by the user and map it to an action. Another important aspect of natural interfaces is the ability to understand speech. A simple example of the first case is adjusting the screen brightness depending on the light conditions of a place, while an instance of the second is to turn off the screen when the user turns his head away from it thus removing focus from the application.

- Capacitive Sensor

A touch screen able to detect the 3D position of an object placed not farther than 5 centimetres from the screen. The sensing is performed by an array of capacitive sensors

integrated into the screen. Due to the fact that the entire interface can be incorporated into the touch screen, it could be used as a 3D hand interface for portable devices such as smart phones, and tablets.

Capacitive sensors can directly sense a variety of things—motion, chemical composition, electric field—and, indirectly, sense many other variables which can be converted into motion such as pressure, acceleration, fluid level, and fluid composition. They are built with conductive sensing electrodes in a dielectric, with excitation voltages on the order of five volts and detection circuits which turn a capacitance variation into a voltage, frequency, or pulse width variation. The range of application of capacitive sensors is extraordinary.

- Motion detectors can detect 10-14 m displacements with good stability, high speed, and wide extremes of environment and capacitive sensors with large electrodes can detect an automobile and measure its speed.

- Capacitive technology is displacing piezo resistance in silicon implementations of accelerometers and pressure sensors, and innovative applications like fingerprint and infrared detectors are appearing on silicon with sensor dimensions in the microns and electrode capacitance of 10 Ff.

- Capacitive sensors in oil refineries measure the percent of water in oil and sensors in grain storage facilities measure the moisture content of wheat.

- In the home, cost-effective capacitive sensors operate soft-touch dimmer switches and help the home craftsman with wall stud sensors and digital construction levels.

- Laptop & computers use capacitive sensors for two-dimensional cursor control, and transparent capacitive sensors on computer monitors are found in retail kiosks

A Capacitor is formed whenever two conductive surfaces are separated by an insulating layer, which is known as the dielectric layer. Capacitance is measured in Farads. There are many

different capacitive sensing algorithms, and the sensors themselves can be constructed from many different materials, such as aluminium, silver, gold, copper, ITO, and various printed (conductive) inks. Typically copper capacitive sensors can be implemented using standard FR4 (fibreglass) printed circuit board material itself as the dielectric layer. Touch screen capacitive sensors typically utilize ITO materials because it can be up to ninety percent (90%) transparent, allowing you to see the screen underneath without too much light loss. All capacitive sensing systems work by measuring a change in capacitance as a human interacts with the sensor, and they differ in the method, or algorithm used to measure that capacitance.

Surface capacitance measurement (sometimes referred to as "active" capacitance measurement) is performed by the application of a small voltage to the sensor, which creates a uniform electrostatic field. When a human comes into contact with the sensor, (and hence the electrostatic field) a capacitor is formed dynamically. The capacitive sensor controller measures the change in capacitance to determine the proximity of the human within the electrostatic field.

Capacitive sensors are non contact by design. That is, they are able to precisely measure the position or displacement of an object without touching it. Because of this the object being measured will not be distorted or damaged and target motions will not be dampened. Additionally, they can measure high frequency motions because no part of the sensor needs to stay in contact with the object, making them ideal for vibration measurements or high speed production line applications.

As mentioned above, the range of a capacitance sensor is dictated by the diameter, or area, of the sensor. The larger the area, the larger the measurement range. Measurement range is typically specified starting when the probe is touching the target. At this point the output from the system is zero volts. When the gap is increased to equal the full scale measurement range of the capacitive system the system output is 10 volts (Vdc). In theory, the probe can

operate anywhere between these two extremes, however, it is not recommended to operate below 10% of the gap. With this said, the ideal operating or standoff distance is somewhere between 5Vdc to 7Vdc which will allow the target to move closer to or further away from the probe without going out of range. The resolution of a capacitive sensor is defined as the smallest amount of distance change that can be reliably measured by a specific system. Capacitance sensors offer extremely high resolution and stability often exceeding that of expensive and complex laser interferometer systems. Because of their ability to detect such small motions, they have been successfully used in many demanding measurement applications including computer disk drive runout, microscope focusing and nano-positioning within highly complex photolithography tools.

The primary factor in determining resolution is the system's electrical noise. If the distance between the sensor and target is constant, the voltage output will still fluctuate slightly due to the "white" noise of the system. It is assumed that, without external signal processing, one cannot detect a shift in the voltage output of less than the random noise of the instrument. Because of this most resolution values are presented based on the peak-to-peak value of noise and can be represented by the following formula: Resolution = Sensitivity X Noise.

Sensitivity is simply the measurement range divided by the voltage output swing of the capacitance amplifier. From the formula, you can see that for a fixed sensitivity the resolution is solely dependent upon the noise of the system. The lower the noise, the better the resolution.

It is important to note that some manufacturers specify resolution based on peak or rms noise, resulting in claims that are 2x and 6x respectively better than peak-to-peak. Although an acceptable method, it is somewhat misleading as most users do not have the ability to decipher voltages changes less than the peak-to-peak noise value.

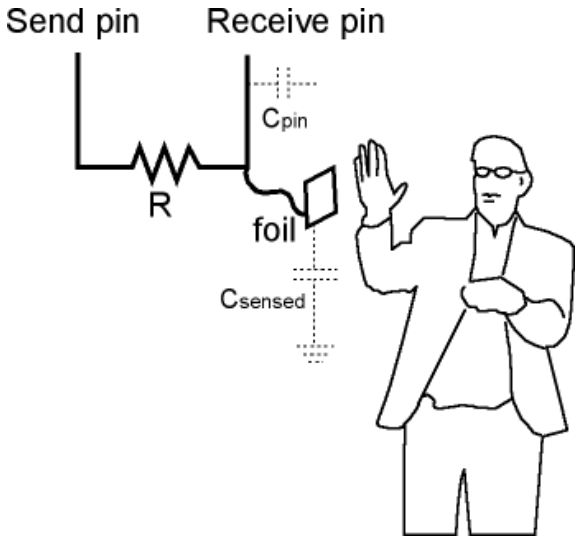

Figure 1

Capacitive touch sensors working

CHAPTER 2

2.1 TECHNICAL DETAILS

2.1.1 Basic Theory

A capacitor is a passive two-terminal electrical component used to store energy electrostatically in an electric field.

A capacitor contains at least two electrical conductors (plates) separated by a dielectric (i.e. insulator). The conductor can be thin films, foil or sintered beads of metal or conductive electrolyte etc. The "non conducting" dielectric acts to increase the capacitor's charge capacity. A dielectric can be glass, ceramic, plastic film , air, vacuums , paper , mica, oxide layer etc. When there is a potential difference across the conductors (e.g., when a capacitor is attached across a battery), an electric field develops across the dielectric, causing positive charge +Q to collect on one plate and negative charge −Q to collect on the other plate. If a battery has been attached to a capacitor for a sufficient amount of time, no current can flow through the capacitor. However, if a time-varying voltage is applied across the leads of the capacitor, a displacement current can flow.

An ideal capacitor is characterized by a single constant value for its capacitance. Capacitance is expressed as the ratio of the electric charge Q on each conductor to the potential difference V between them. The SI unit of capacitance is the farad (F), which is equal to one coulomb per volt (1 C/V). Typical capacitance values range from about 1 pF (10^{-12} F) to about 1 mF (10^{-3} F).

2.1.2 Concept of capacitive sensing

In electrical engineering, capacitive sensing is a technology, based on capacitive coupling, that takes human body capacitance as input. Capacitive sensors detect anything that is conductive or has a dielectric different from that of air. The formula for capacitance is given as

$$C = \varepsilon A/d$$

where

C = Capacitance in farads

ε = Permittivity of the medium

A = Area of the plate

d = Distance between the plates

As we can see that capacitance is a function of distance between the plates, changes in the distance between the surfaces changes the capacitance. It is this change of capacitance that capacitive sensors use to indicate changes in position of a target.

2.1.3 Capacitance and Distance

Non contact capacitive sensors work by measuring changes in an electrical property called capacitance. Capacitance describes how two conductive objects with a space between them respond to a voltage difference applied to them. When a voltage is applied to the conductors, an electric field is created between them causing positive and negative charges to collect on each object (Fig. 2). If the polarity of the voltage is reversed, the charges will also reverse.

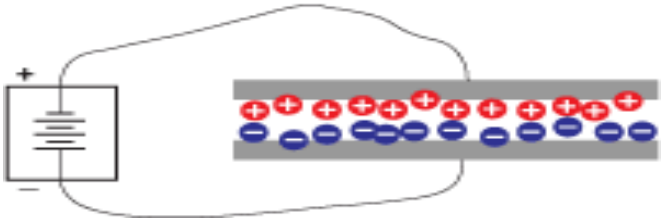

Figure 2

Applying a voltage to conductive objects causes positive and negative charges to collect on each object.

Capacitive sensors use an alternating voltage which causes the charges to continually reverse their positions. The moving of the charges creates an alternating electric current which is detected by the sensor (Fig. 3). The amount of current flow is determined by the capacitance, and the capacitance is determined by the area and proximity of the conductive objects. Larger and closer objects cause greater current than smaller and more distant objects. The capacitance is also affected by the type of nonconductive material in the gap between the objects.

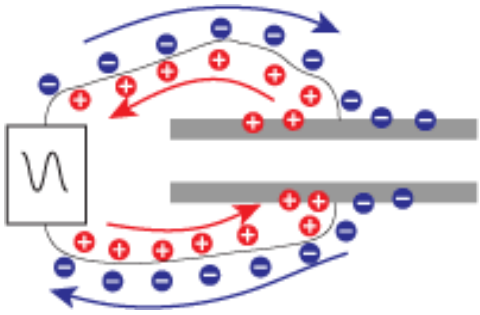

Figure 3

Applying an alternating voltage causes the charges to move back and forth between the objects, creating an alternating current which is detected by the sensor.

In typical capacitive sensing applications, the probe or sensor is one of the conductive objects; the target object is the other. (Using capacitive sensors to sense plastics and other insulators is discussed in the nonconductive targets section.) The sizes of the sensor and the target are assumed to be constant as is the material between them. Therefore, any change in capacitance is a result of a change in the distance between the probe and the target. The electronics are calibrated to generate specific voltage changes for corresponding changes in capacitance. These voltages are scaled to represent specific changes in distance. The amount of voltage change for a given amount of distance change is called the sensitivity. A common sensitivity setting is 1.0V/100µm. That means that for every 100µm change in distance, the output voltage changes exactly 1.0V. With this calibration, a +2V change in the output means that the target has moved 200µm closer to the probe.

In this project we will take a very simple idea- the length of time it takes a capacitor to charge and discharge using interfacing of Arduino and Matlab which will track the position of our hand.

This change in voltage (charging or discharging) occurs after a certain time called time constant given by

$$\tau = RC$$

The equation of charging and discharging of a capacitor is given as follows

Charging $V(t) = V_o(1 - e^{-t/\tau})$

Discharging $V(t) = V_o(e^{-t/\tau})$

The electrode of a touch sensor represents one plate of such a capacitor. The corresponding 2nd plate is represented by the environment of the sensor electrode (to form a parasitic

capacitor C_0) and another conductive object, like a human finger for example (to form touch capacitor C_T). This capacitor, i.e. the sensor electrode, is connected to a measurement circuit. The capacitance of the sensor pad is measured periodically. If a conductive object approaches or touches the electrode, relative permittivity ε_r air/vacuum: ε_r=1 PE: ε_r=2 wood: ε_r=3 ABS: ε_r=4 glass: ε_r=7 water: ε_r=80, the measured capacitance will increase. This change is detected by the measurement circuit and converted into a trigger signal. Considering the formula above, one can see that a bigger pad and a thinner overlaying cover material, leads to a bigger touch capacitance C_T and as a result, a bigger capacitance difference between touched and untouched sensor pad. In other words, the size of the electrode and the covering material influence the sensitivity of the sensor.

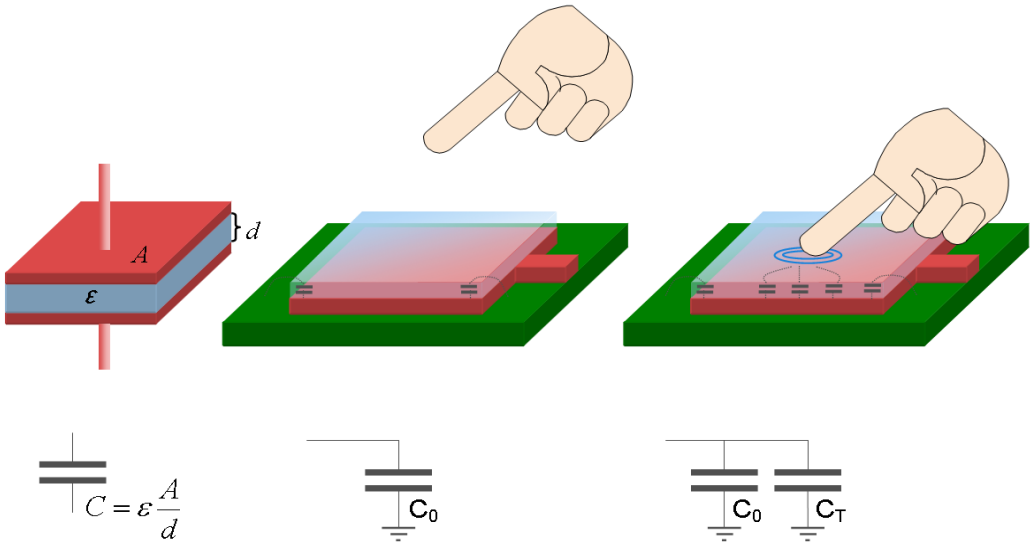

Figure 4

Touch Sensor Principle: Untouched sensor pad with parasitic capacitance C_0, touched sensor pad with additional touch capacitance C_T.

2.2 CIRCUIT DIAGRAM

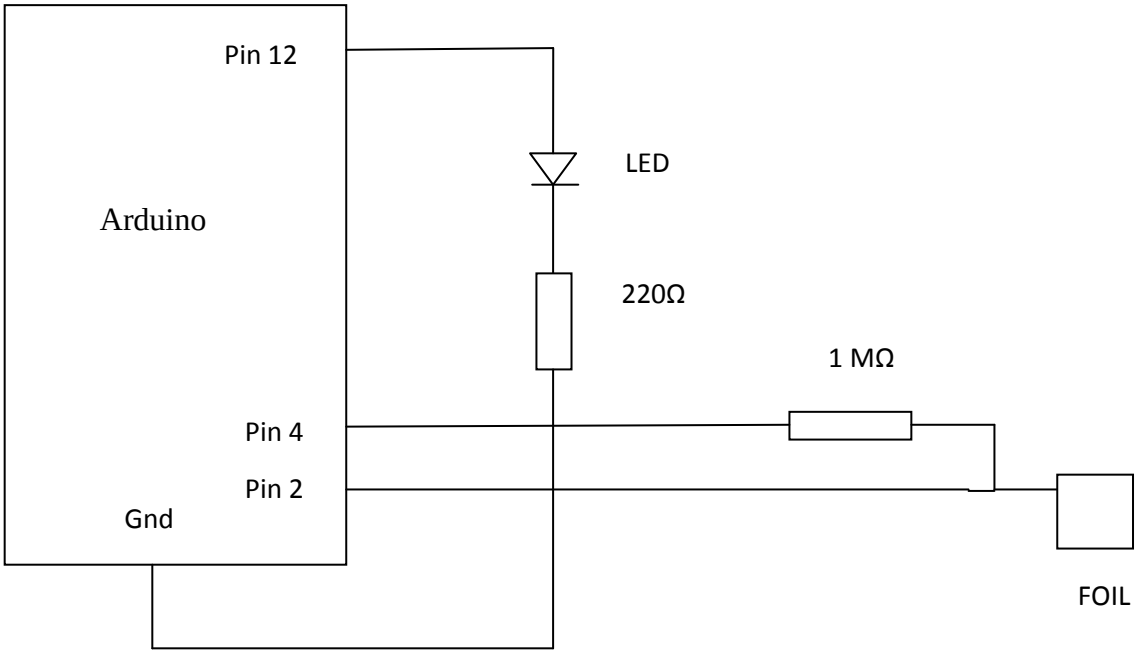

Figure 5

Figure 5 is the schematic where a LED is connected to the Arduino board. This LED is switched on or off based on the charging and discharging of the capacitor consisting of an aluminium foil and our hand as the other plate. This practical served as basis for implementing our objective in three dimensions.

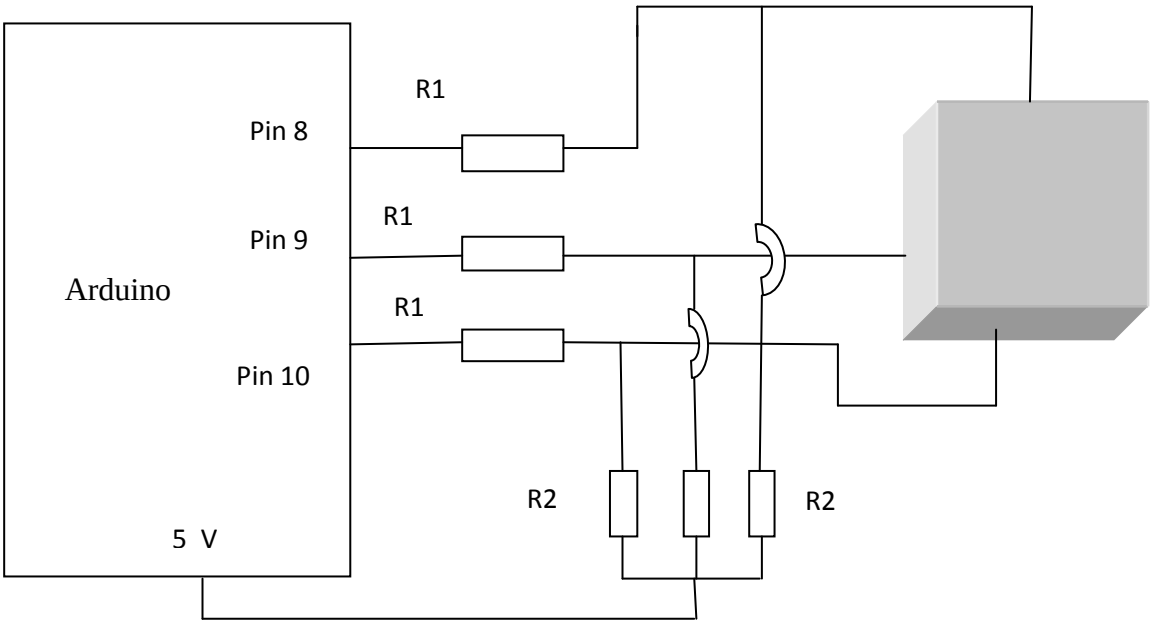

Figure 6

Here , the three 10kΩ resistors are connected to the given pins of the arduino board .The purpose of these resistors is that we do not connect any electronic device(eg, LED) directly with the microcontroller so as to prevent a large amount of instantaneous current flow through the microcontroller which can cause damage to the devices connected.

The three 220kΩ resistors are used for the charging and discharging of the three capacitor plates respectively. The values are decided based on at how much distance do we want our capacitive sensor to detect movement. In our case, any resistance between the range 200kΩ to 500kΩ can be selected.

2.3 HARDWARE USED

2.3.1 Arduino

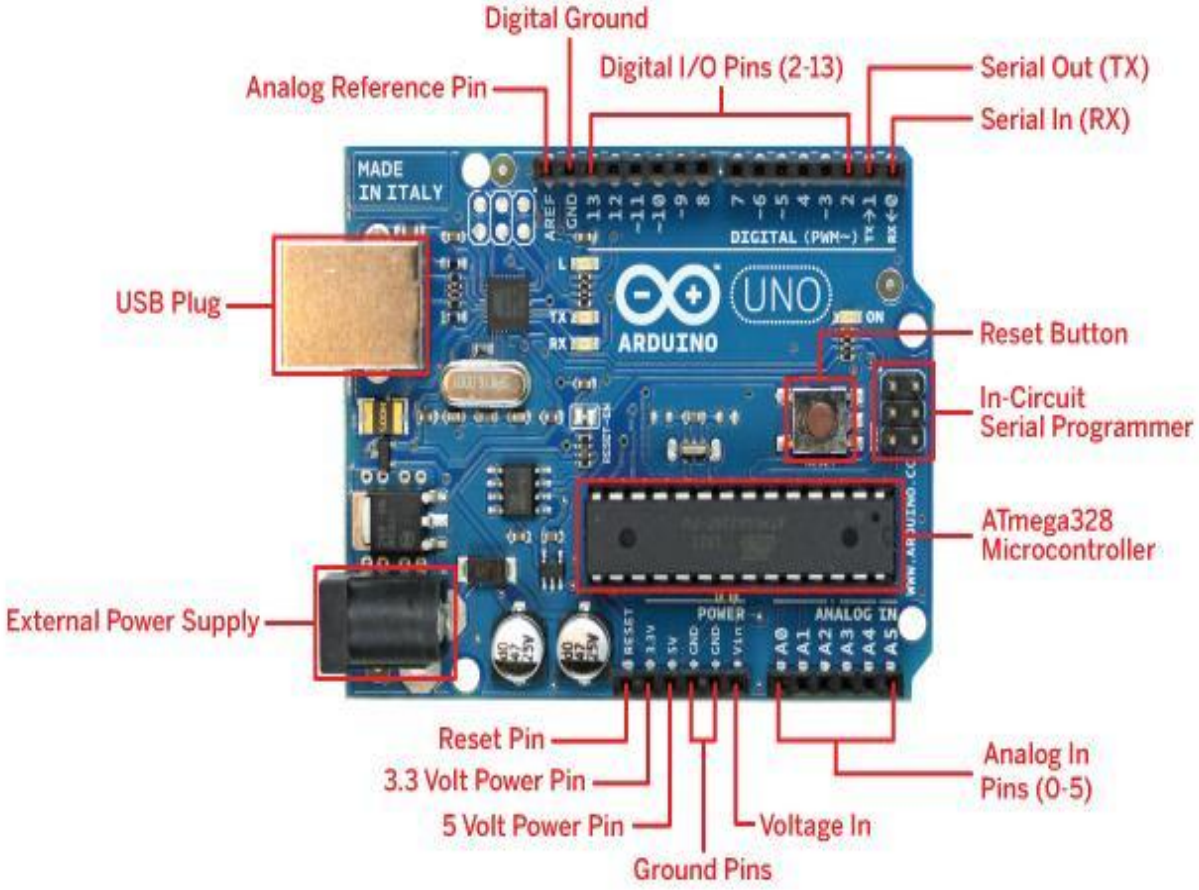

Figure 7

This figure shows the specifics of the microcontroller used in our project, which is ARDUINO UNO

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Table 1-Specifications of Arduino

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

The power pins are as follows:

VIN: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3.3V: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND : Ground pins.

IOREF: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

LED: There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

AREF: Reference voltage for the analog inputs. Used with analogReference().

Reset: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

2.4 SOFTWARE USED

2.4.1 Arduino IDE

The open-source Arduino environment makes it easy to write code and upload it to the i/o board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing, avr-gcc, and other open source software.

2.4.2 Matlab

Matlab is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enabled us to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java.

CHAPTER 3

WORK DESCRIPTION

3.1 METHODOLOGY

This chapter presents the steps of the process that was followed when investigating the Touchless Interaction.

1. Getting acquainted with the arduino board as it plays a major role in the project, its various pin functions, its specifications etc.
2. Checking that the microcontroller is performing the various functions we will require in our project.
3. Programming of arduino in the arduino IDE (Integrated Development Environment) which will take the data from the arduino in the form of change in voltage and send that to the Matlab.
4. Matlab programming to take the input from arduino through serial communication and move the ball corresponding to the input provided from the Arduino.
5. Serial communication between Matlab and arduino as the input is given to arduino which needs to be seen through a graphical user interface (GUI) prepared in matlab in our laptop/computer.
6. Evaluating the different programs and checking them by providing input values of capacitors (dummy values) and correspondingly noting the different outputs, this we ultimately used to decide which values of resistors we need with the capacitor value provided by the aluminium plates used in the project.
7. Then we created a 2-D model and tested the concept of capacitive sensing. We used an LED to confirm that our microcontroller was reading capacitive values and the same were faithfully being transmitted to Matlab.

8. After this, we proceeded on to build our 3-D model using cardboard plates, aluminium foils, resistors, breadboard and wires.
9. Finally, we tested the hand tracking movement algorithm by executing several hand motions in the 3-D space and saw the same being shown on the graphic user interface in the matlab window.

3.2 CODES

3.2.1 For Switching LED

- ARDUINO CODE

```
#include <CapacitiveSensor.h>
// pin 4 sends electrical energy
// pin 2 senses senses a change
capacitiveSensor capSensor = CapacitiveSensor(4,2);
const int ledPin = 12;
void setup() {
        Serial.begin(9600);
        pinMode(ledPin, OUTPUT);
```

```
void loop() {
                long sensorValue = capSensor.capacitiveSensor(30);
                Serial.println(sensorValue);
                 if(sensorValue > 15)
                   {
                        digitalWrite(ledPin, HIGH);
                    }
                 else {
                        digitalWrite(ledPin, LOW);
                      }
                 delay(10);
               }
```

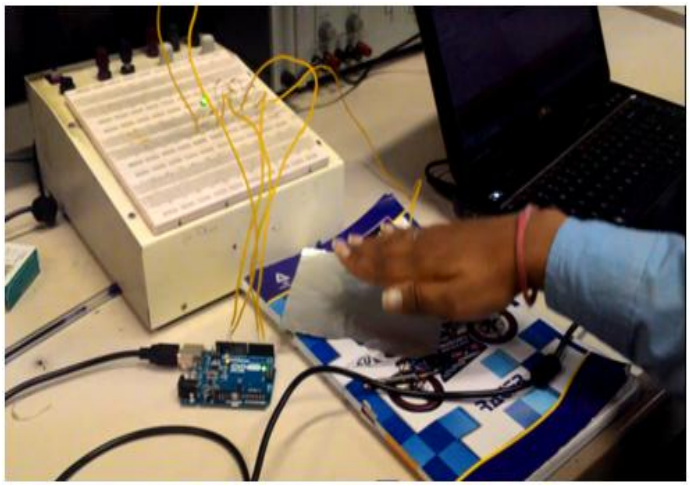

Figure 8(a)

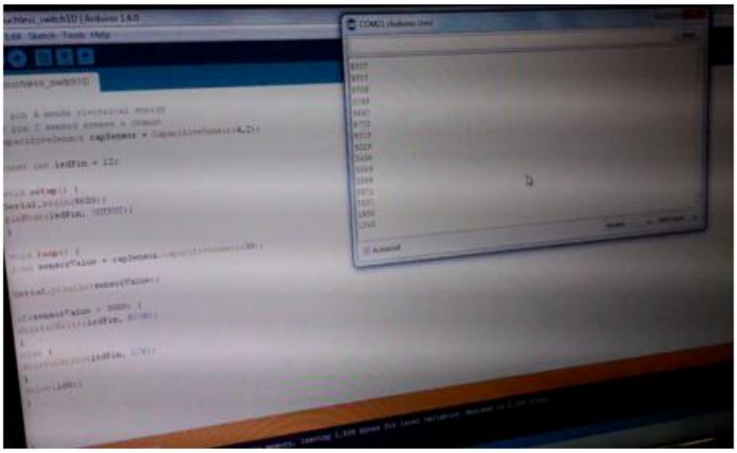

Figure 8(b)

Working code in Arduino for switching LED

Values shown on serial monitor

3.2.2 For 3D Tracking

- ARDUINO CODE

```
void setup() {
                Serial.begin(4800);
                 // unused pins are fairly insignificant,
                // but pulled low to reduce unknown variables
                        for(int i = 2; i < 14; i++) {
                                pinMode(i, OUTPUT);
                                digitalWrite(i, LOW);
                                        }
                   for(int i = 8; i < 11; i++)
                          pinMode(i, INPUT);
                   startTimer();
                  }
void loop() {
                Serial.print(time(8, B00000001), DEC);
                Serial.print(" ");
                Serial.print(time(9, B00000010), DEC);
                Serial.print(" ");
```

```
        Serial.println(time(10, B00000100), DEC);
    }
long time(int pin, byte mask) {
                unsigned long count = 0, total = 0;
                while(checkTimer() < refresh)
                    {
                      // pinMode is about 6 times slower than
                      assigning
                   // DDRB directly, but that pause is important
                       pinMode(pin, OUTPUT);
                        PORTB = 0;
                        pinMode(pin, INPUT);
                        while((PINB & mask) == 0)
                          count++;
                           total++;
                    }
        startTimer();
        return (count << resolution) / total;
```

```
        }
 extern volatile unsigned long timer0_overflow_count;
void startTimer() {
          timer0_overflow_count = 0;
          TCNT0 = 0;
                  }
unsigned long checkTimer() {
                                    return ((timer0_overflow_count << 8) + TCNT0) << 2;
                                  }
```

- MATLAB CODE

```
function sphz();
% create figure
figure('color',[1 1 1],'me','n','nu','off')
he=uicontrol('sty','e','ba','w','un','n','p',[.04 .1 .1 .05],'str','5 0');
hs=uicontrol('sty','e','ba','w','un','n','p',[.04 .25 .1 .05],'str',10);
uicontrol('sty','pu','ba','w','str','Go','ca',@go1)
%uicontrol('sty','te','ba','w','str','X - Z','un','n','p',[.04 .16 .1 .05])
```

```
uicontrol('sty','te','ba','w','str','Speed','un','n','p',[.04 .3 .1 .05])

% create ground and adjust axis

hold on

grid on

axis([-5 5 -5 5 -5 5].*2)

axis square

aa=[-5 0 -5;5 0 -5;5 0 5;-5 0 5;-5 0 -5;5 0 -5;5 0 5;-5 0 5].*2;

bb=[1 2 3 4;5 6 7 8;1 2 6 5;2 3 7 6;3 4 8 7;4 1 5 8];

patch('vertices',aa,'faces',bb,'edgecolor','w','facecolor',[.5.5],'facevertexalphadata',0.5,'
facealpha','flat');

view(3)

rotate3d

xlabel('X')

ylabel('Y')

zlabel('Z')

% create sphere

 h=movsph(1,[0,0],[0,0],25,0);

 % move sphere

        function go1(varargin)
```

```
                pt=str2num(get(he,'string')); %#ok
                deg=str2double(get(hs,'string'));
                h=movsph(1,[rand,rand],pt,deg,0,h);
            end
        end
```

Checking the status of COM ports in Matlab

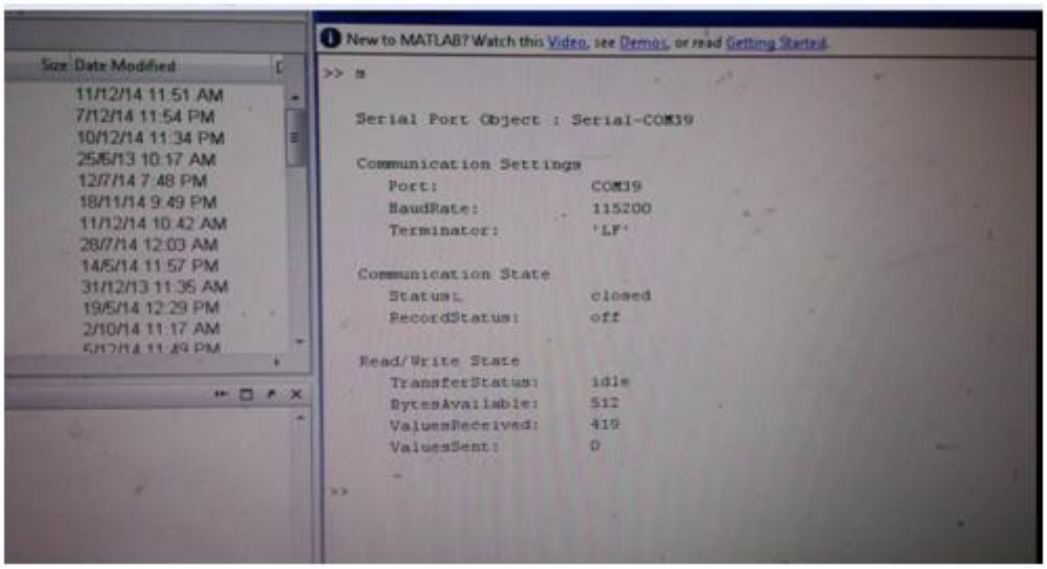

Figure 9

Values obtained on the serial monitor of Arduino IDE

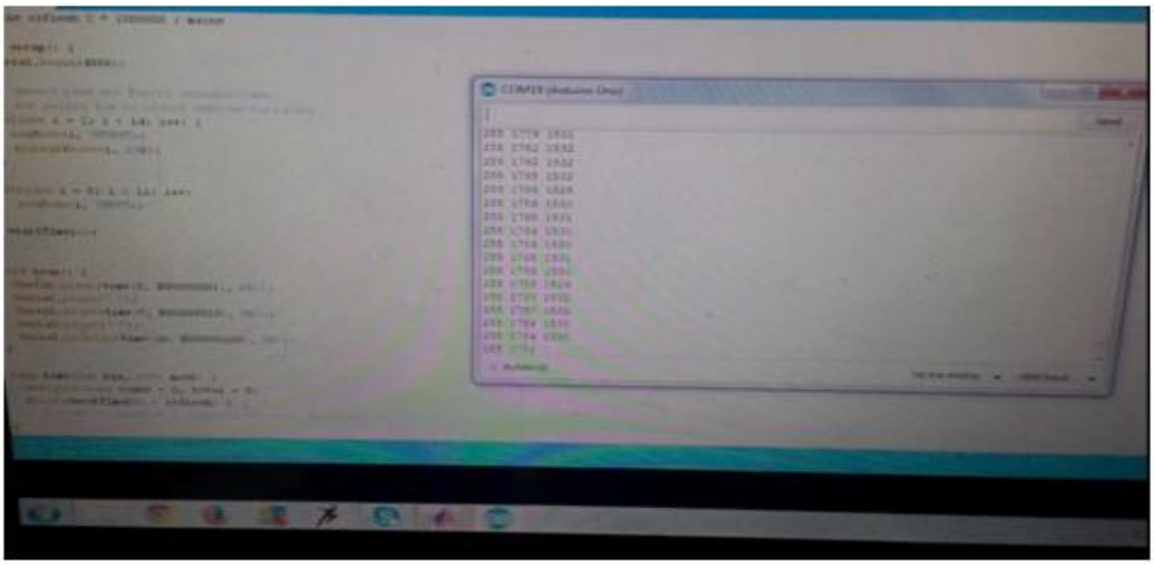

Figure 10

Before taking the input

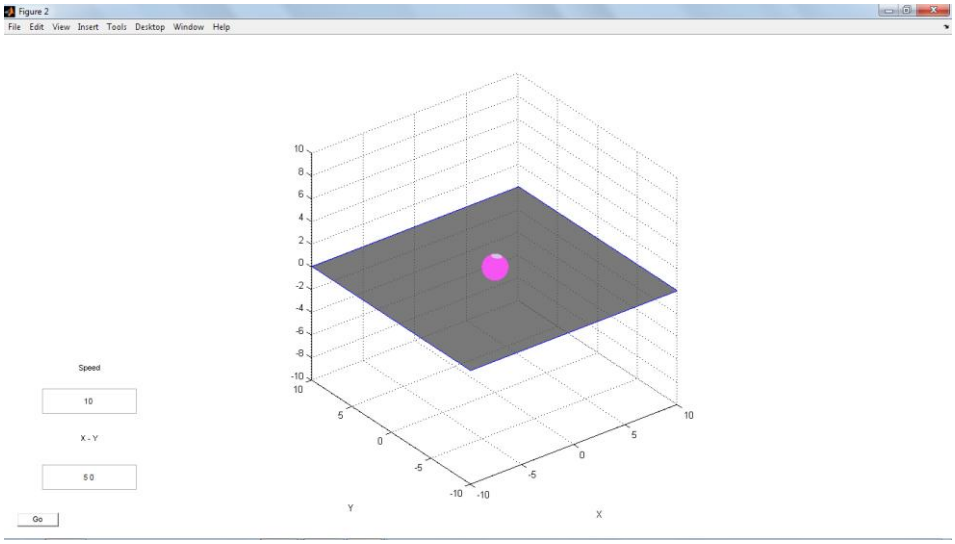

Figure 11

After taking the inputs

- ➢ Movement in x- direction

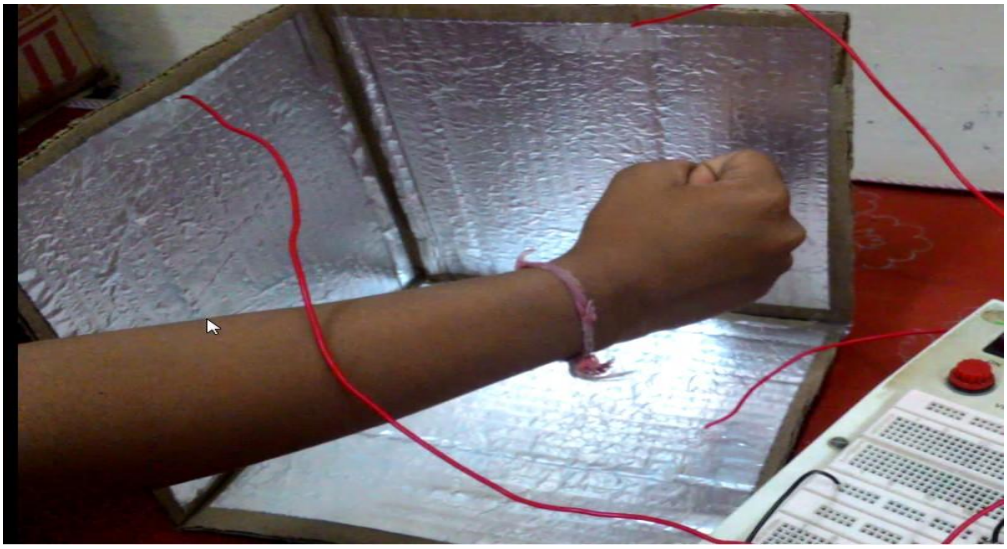

Figure 12

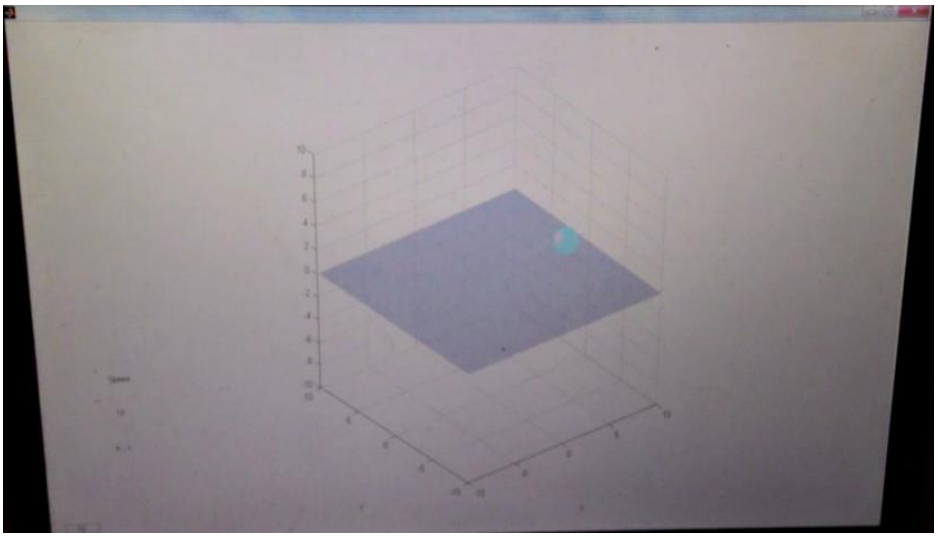

Figure 13

➢ Movement in y- direction

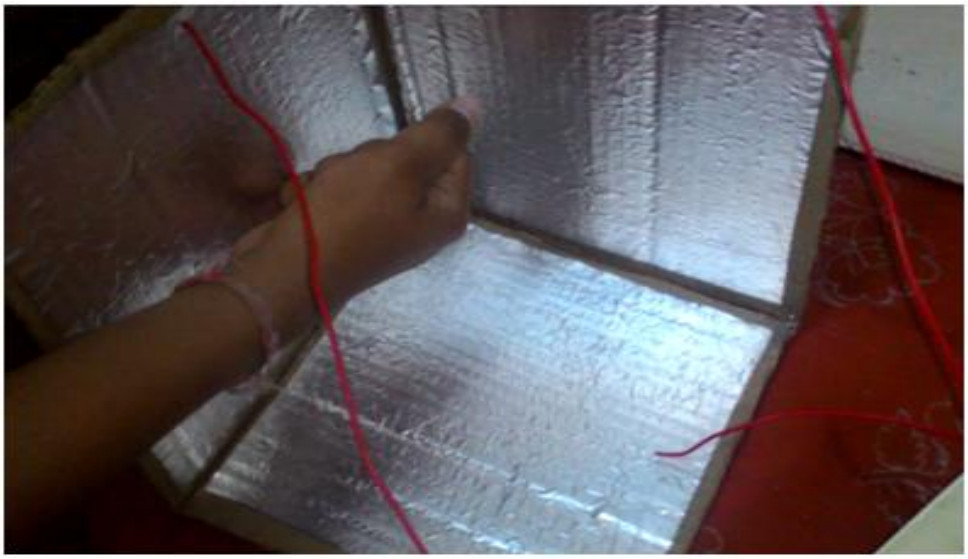

Figure 14

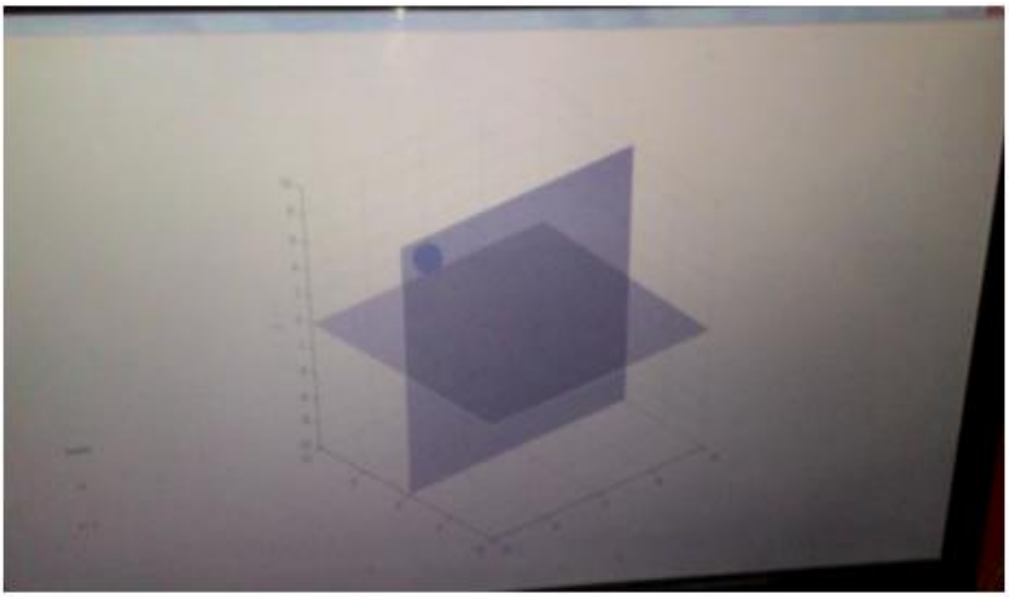

Figure 15

➢ Movement in z-direction

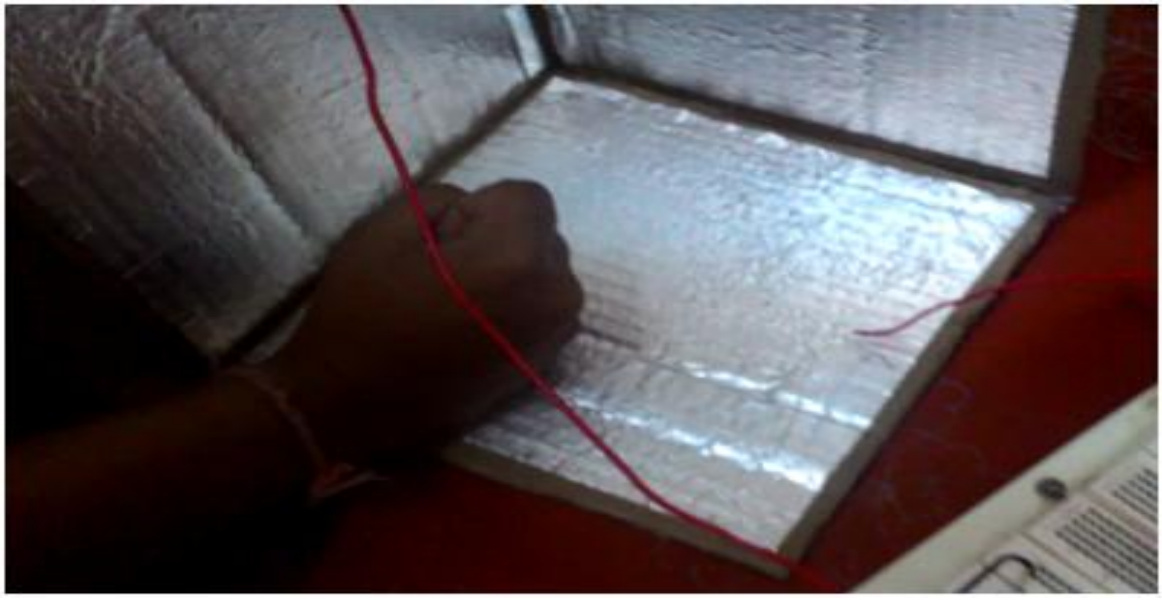

Figure 16

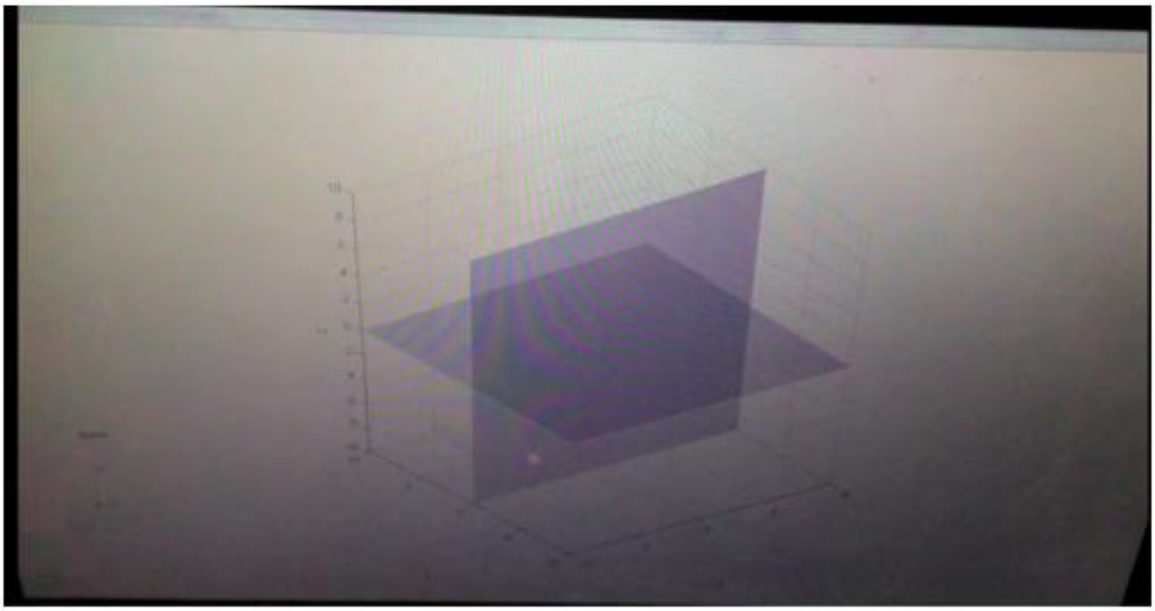

Figure 17

RESULTS

- The microcontroller (arduino UNO) and matlab have been successfully interfaced with each other The input is successfully read by the arduino and the corresponding data, given to matlab shows the motion of a 3D object (i.e. ball) in the (x,y,z) plane.

- We successfully showed that as proximity of the hand with the sensor (aluminium foil) increases or decreases, correspondingly, the current flowing across the LED changed and it glowed(high current flowing) or dimmed(low current flowing) according to it.

- Next we implemented our actual 3-D model and tracked the movements of our hand in the pre-specified 3-D space. We showed that as we moved our hand, the movement was tracked and shown in the graphic user interface of matlab. We successfully showed correct hand position movement..

CHALLENGES

- Serial communication between arduino and matlab proved difficult as we were using two different language platforms, so creating functions which could provide interface between the two was not an easy task.

- The computer is programmed to take the COMPORT of arduino (in our case, it was 18) only once during its initialization, which meant that we had to restart our laptop

every time we had to run our program. This made debugging of program and detecting and correcting errors a very tedious task.

- ➢ Arduino as a whole is a very slow working microcontroller, and we had to do a lot of permutation and combination with the serial communication rate to achieve close to real time tracking conditions.
- ➢ 2-D implementation using the graphic user interface proved much simpler as compared to 3-D implementation. The ball movement in all 3 directions simultaneously using the data provided by arduino was also difficult. We had to further improve our interfacing matlab code to correctly accommodate for it.
- ➢ Sensitivity quickly decreases.
- ➢ Interference from other objects.
- ➢ Relatively small workspace, places limitations on the usability of this interface

FUTURE WORK

- ➢ The work we have done in our project can be further developed into a game, after making certain adjustments for a bigger 3-D tracking space
- ➢ Apart from a game, other user friendly applications which use motion detection can also be developed using this.
- ➢ We have implemented a single user tracking system. This can be made more complex by implementing two or more users tracking system, which tracks the hand movement of many persons simultaneously.
- ➢ We can also control our computers , projectors and mobile phones without actually touching the device.
- ➢ We can modify the Arduino code to use that information to drive a device of our choice! We'll need to take apart the device and wire some inputs from the Arduino

into its circuitry. For eg:- We can drive the throttle of a remote controlled helicopter, varying rotor speeds (and thus the height that the helicopter flies at) depending on the position of our hand.

- ➢ We can also control video or music parameters; sequence a beat or melody.
- ➢ It can also be used to glow LED's with the movement of our hand.

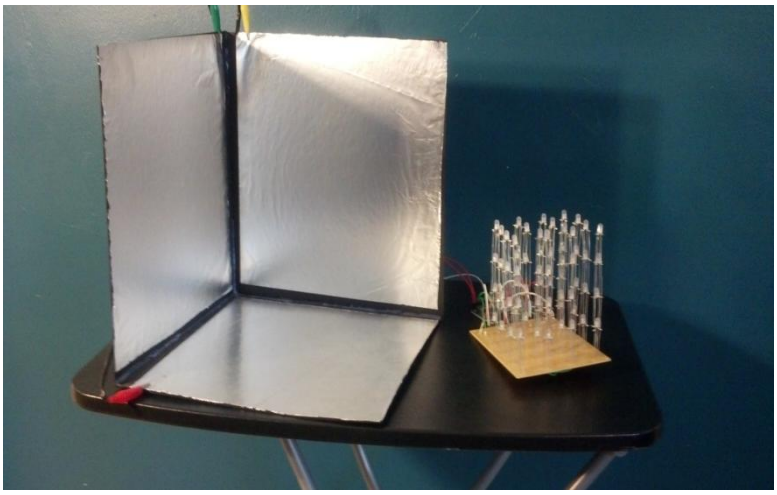

Figure 18

- ➢ Touch less Pads can also be made using this technology.

REFERENCES

[1] 3D Tracking Using Particle Filters, Yasir Salih, Aamir S. Malik, Senior Member IEEE,, Department of Electrical & Electronic Engineering, Universiti Teknologi PETRONAS, Perak, Malaysia. Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE, page 1-4

[2] Automated 3D Motion Tracking Using Gabor, Filter Bank, Robust Point Matching and Deformable Models, Ting Chen*, Member, IEEE, Xiaoxu Wang, Sohae Chung, Dimitris Metaxas, Member, IEEE, and Leon Axel. Medical Imaging, IEEE Transactions on (Volume:29 , Issue: 1 , page 1-11)

[3] Encapsulated Copper Wire and Copper Mesh, Capacitive Sensing for 3-D Printing Applications, Corey Shemelya, Member, IEEE, Fernando Cedillos, Efrian Aguilera, David Espalin, Danny Muse, Ryan Wicker, and Eric MacDonald, Senior Member, IEEE. IEEE SENSORS JOURNAL, VOL. 15, NO. 2, page 1280-1285

[4] www.visionnw.com/3d-technology-techniques.

[5] www.analogrules.com/capacitors.

[6] Larry K. Baxter(1996).Capacitive sensors. John Wiley and Sons

[7] playground.arduino.cc

[8] cache.freescale.com/files/sensors/doc/white_paper/PROXIMITYWP.pdf

[9] www.touchadvance.com/2011/06/capacitive-touch-touch-sensing.

[10] makezine.com/projects/a-touchless-3d-tracking-interface

[11] www.instructables.com/id/DIY-3D-Controller.

[12] publications.lib.chalmers.se/records/fulltext/201511/201511.pdf