TOUCHLESS 3D INTERFACE

V.V.RAVI THEJA REDDY-13BEC0745 M.VAMSI KRISHNA-13BEC0594
A.VISWANATH REDDY-13BEC0665

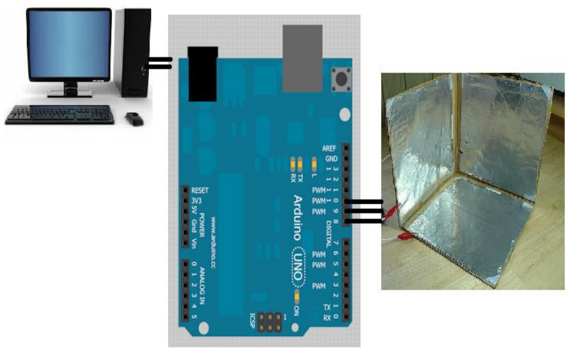
***Abstract*: This report is about the project "Touch less 3D Interface System". Due to ever increasing demand of industrialization, this type of interfacing systems are becoming more and more necessary and also makes our work easy. This module makes you to sit at a place and control a car, robot, rover and a quad copter using a hand. It can also be used to track objects in real time.**

I. *INTRODUCTION*

In a developing and populous country like India, we have to stand in a queue to get demandable services. In this project, we introduced the concept of capacitive sensing. Nearly all sensing of this kind depends upon how long it takes a capacitor will affect the capacitor will affect the capacitance value and the corresponding time constant

We hope that, using our device everybody will feel comfort and will be exciting to use. In this project, we have started with the block diagram of our proposed system. Then we described each part of block diagram of our proposed system. We described our software part by giving code. We also have given the picture of our practical device.

II. *BLOCK DIAGRAM*



III. *DESCRIPTION OF PARTS*

A. Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects. Arduino board senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators. You can tell your Arduino what to do by writing code in the Arduino programming language and using the Arduino development environment.

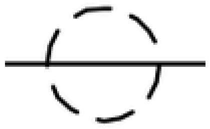
B. Shielded Wire

A Shielded cable is an electrical cable of one or more insulated conductors enclosed by a common conductive layer. The shield may be composed of braided strands of copper (or other metal, such as aluminium), a non-braided spiral winding of copper tape, or a layer of conducting polymer. This is mainly used to reduce the electromagnetic interference (EMI). The electrical environment around the area you are installing the cable determines what is the best choice for that job. An area such as a production/factory floor where heavy equipment is being used is a prime example of a place where you might consider a shielded cable.

The best method to wire shielded cables for screening is to ground the shield at both ends of the cable. Grounding can also be a concern in some installations. If shielded cable is used to connect equipment from two different circuits, a ground loop can occur causing noise on a network line. If the ground voltage difference is great enough it may even cause damage. Installers will often leave one end of the shielded cable terminated with a non-shielded connector. This is often referred to as a "floating" shield. This too can cause a different problem. The floating shield often acts as an antenna that picks up additional noise.

So the solution is to ground the shielded wires at both ends. If ground loop occurs we can go for floating point. And if this also doesn’t work we should use fiber optic cables.

Here in our project we have used Shielded wire because the time constant may be effected due to EMI. We have supplied five volts to the shield at one end as a floating point and also taken measures to reduce antenna effect.



Shielded wire Symbol

C. *Processing Software*

Processing is an open source programming language and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the propose of teaching the fundamentals of computer programming in a visual context, and to serve as the foundation for electronic sketchbooks.

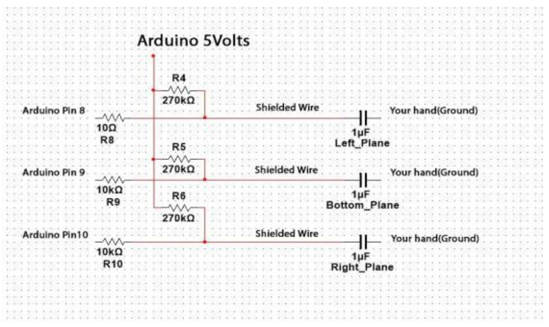
Here in this project we have used processing software to show the output the position of the hand in the cubic box. It gets the values (time constant values) from Arduino through serial communication and normalize the values to display them on the cube.

D. *Aluminium Foils*

Aluminium foil often referred to with the misnomer tin foil, is aluminium prepared in thin metal leaves with a thickness less than 0.2 mm (7.9 mils); thinner gauges down to 6 micrometres (0.24 mils) are also commonly used.

In this project we have used these foils as the capacitors placed in the 3 dimensional axis to calibrate position of the hand. Cut out three pieces of 26cm x 26cm aluminium foil and tape them to the cardboard as shown in the picture.

IV. *CIRCUIT EXPLANATION*



The two parts of the shielded wire the inner core copper wire and outer shielded wire. Instead of grounding the shielded wire we give five volts to reduce electromagnetic interference. So at one end of the shielded wire just connect the inner wire to the aluminium foils and left open the shield (as floating point). The other end we connect to resistors 10k and 270k to the inner part for all the three wires and combine all the 270k resistors to one point connect to five volts. And also connect the same five volts to the shield outside. Now connect the three 10k resistors to 3 pins in Arduino.

In the circuit, hand act as another (second) capacitor.

V. WORKING PRINCIPLE

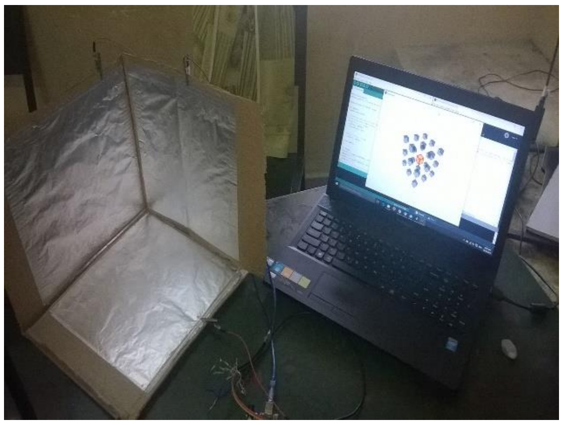
The controller is made up of a cardboard support, 3 plates of aluminium foil, shielded wire, resistors, an Arduino, and a computer with a charger and the proper software.

By putting your fist in the space between the plates, you are creating a capacitor between your fist and the plates, with the plates at some voltage and your body at ground.

By changing the distance between your fist and the plates, you are changing the capacitance. The Arduino can measure this by grounding the plates, then letting them recharge from a constant 5V source, and measuring how long it takes to recharge to a certain threshold. This time is determined by the RC constant (equivalent resistance of the circuit times the capacitance), and since the resistance is constant, it's a measure of the capacitance.

The Arduino can measure the changes in this recharging time relative to an initial calibration, and through the Processing code, it figures out the location of your hand.

Plug the circuit into the Arduino and run the code. Run the Processing sketch and make sure the capacitor can sense the location of your hand.



A. Explanation of the Arduino Code

1) Setup: It sets up the Serial communication port to communicate at 115200 baud rate. (Serial communication works by sending one byte at a time between the computer and the Arduino at a certain number of symbols per second, or baud. 115200 baud is currently the fastest possible communication rate between the computer and the Arduino.)

2) The Arduino then sets every pin to ground (LOW). This is to prevent fields from other pins from potentially affecting the wires going into the Arduino in some sort of an antenna effect. It then initializes pins 8, 9 and 10 to INPUT (remember pin 8 is connected to the left plate, pin 9 is connected to the bottom plate and pin 10 is connected to the right plate). At this time, all of the plates will be charged to 5 volts through the 270k ohm resistors.

3) Loop: The code calls the function "time" for pin 8 (left plate), which first sets pin 8 to OUTPUT and sets the voltage on the pin to a (digital) low (which is close but not necessarily equal to 0V). This will very quickly discharge the plates through the 10k resistors.

4) After sometime during or after this discharge process, the Arduino will set pin 8 to INPUT. At this point, it will start counting upwards, once every clock cycle, until the plate has charged up enough through the 270k ohm resistors that the Arduino sees the voltage at pin 8 has gone back to (digital) high (which is close but not necessarily equal to 5V).

5) Then the Arduino will stop counting, and report the number over the Serial cable to the computer. Then the computer, using the Processing sketch, figures out based on this number where your fist is located relative to the left plate. This same process now repeats for the bottom and right plates. All of this happens in a fraction of a second, repeated over and over.

In this way Arduino constantly reads the position of the hand in the cube.

VI. CODE

A. *Arduino Code*

```
#define resolution 8
#define mains 50
#define refresh 1 * 1000000 / mains
void setup() {
  Serial.begin(115200);
  // 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;
}
```

B. *Processing Code*

```
import processing.serial.*;
import processing.opengl.*;
Serial port;
int serialPort = 2;   // << Set this to be the serial
port of your Arduino - ie if you have 3 ports :
COM1, COM2, COM3 and your Arduino is on
```

```
COM2 you should set this to '1' - since the array is
0 based
int sen = 3; // sensors
int div = 3; // board sub divisions
String got;
int end =10;
Normalize n[] = new Normalize[sen];
MomentumAverage cama[] = new
MomentumAverage[sen];
MomentumAverage axyz[] = new
MomentumAverage[sen];
float[] nxyz = new float[sen];
int[] ixyz = new int[sen];
float w = 256; // board size
boolean[] flip = {
  false, true, false};
int player = 0;
boolean moves[][][][];
PFont font;
void setup() {
  size(800, 600, P3D);
  frameRate(25);
  font = loadFont("TrebuchetMS-Italic-20.vlw");
  textFont(font);
  textMode(SHAPE);
  port = new Serial(this,"COM3", 115200);
  port.bufferUntil('\n');
  for(int i = 0; i < sen; i++) {
    n[i] = new Normalize();
    cama[i] = new MomentumAverage(.01);
    axyz[i] = new MomentumAverage(.15);
  }
  reset();
}
void draw() {
  updateSerial();
  drawBoard();
}
//void SerialEvent(Serial port){
//  got=port.readStringUntil('\n');}
void updateSerial() {
  String cur = port.readStringUntil('\n'); //got;
  //println(cur);//serial.readStringUntil('\n');
  if(cur != null) {
    println(cur);
    String[] parts = split(cur, " ");
    if(parts.length == sen  ) {
      float[] xyz = new float[sen];
      for(int i = 0; i < sen; i++)
        xyz[i] = float(parts[i]);
      if(mousePressed && mouseButton == LEFT)
        for(int i = 0; i < sen; i++)
          n[i].note(xyz[i]);
      nxyz = new float[sen];
      for(int i = 0; i < sen; i++) {
        float raw = n[i].choose(xyz[i]);
        nxyz[i] = flip[i] ? 1 - raw : raw;
        cama[i].note(nxyz[i]);
        axyz[i].note(nxyz[i]);
        ixyz[i] = getPosition(axyz[i].avg);
      }
    }
  }
}
float cutoff = .2;
int getPosition(float x) {
  if(div == 3) {
    if(x < cutoff)
      return 0;
    if(x < 1 - cutoff)
      return 1;
    else
      return 2;
  }
  else {
    return x == 1 ? div - 1 : (int) x * div;
       }
}
void drawBoard() {
  background(255);
  float h = w / 2;
  camera(
    h + (cama[0].avg - cama[2].avg) * h,h +
(cama[1].avg - 1) * height / 2,w * 2,h, h, h, 0, 1, 0);
  pushMatrix();
  // We'll use a stroke.
  noFill();
  stroke(0, 40);
  translate(w/2, w/2, w/2);
  rotateY(-HALF_PI/2);
  box(w);
  popMatrix();
  float sw = w / div;
  translate(h, sw / 2, 0);
  rotateY(-HALF_PI/2);
  pushMatrix();
  float sd = sw * (div - 1);
  translate(
    axyz[0].avg * sd,
    axyz[1].avg * sd,
    axyz[2].avg * sd);
  fill(255, 160, 0, 200);
  noStroke();
  sphere(18);
  popMatrix();
  for(int z = 0; z < div; z++) {
    for(int y = 0; y < div; y++) {
      for(int x = 0; x < div; x++) {
        pushMatrix();
        translate(x * sw, y * sw, z * sw);
        noStroke();
        if(moves[0][x][y][z])
      fill(255, 0, 0, 200); // transparent red
    else if(moves[1][x][y][z])
      fill(0, 0, 255, 200); // transparent blue
    else if(
    x == ixyz[0] &&
      y == ixyz[1] &&
      z == ixyz[2])
      if(player == 0)
        fill(255, 0, 0, 200); // transparent red
      else
        fill(0, 0, 255, 200); // transparent blue
    else
      fill(0, 100); // transparent grey
    box(sw / 3);
    popMatrix();
   }
  }
 }

 stroke(0);
 if(mousePressed && mouseButton == LEFT)
  msg("defining boundaries");
}
void keyPressed() {
 if(key == TAB) {
  moves[player][ixyz[0]][ixyz[1]][ixyz[2]] = true;
  player = player == 0 ? 1 : 0;
 }
}

void mousePressed() {
 if(mouseButton == RIGHT)
  reset();
}
void reset() {
 moves = new boolean[2][div][div][div];
 for(int i = 0; i < sen; i++) {
   n[i].reset();
   cama[i].reset();
   axyz[i].reset();
 }
}
void msg(String msg) {
//we're going to use the console to output.
 println(msg);
}
```

VII. CONCLUSION

We can use this interface with capacitive sensing to run a module like bot, car, a Quad copter or anything with movement of your hand. We can use the coordinates of the cube in Processing and use them. It is the device where you can sit and control an object precisely with your hand.

VIII. ACKNOWLEDGEMENT

We thank Prof.Sathya.P, for giving us the opportunity to do this project.

We are also grateful to VIT University for provision of expertise, and technical support in the implementation. Without their superior knowledge and experience, the Project would lack in quality of outcomes, and thus their support has been essential.

Nevertheless, we express our gratitude towards our colleagues for their kind co-operation and encouragement which helped us in completion of this project.

IX. REFERECES

[1] Aezinia, F and Yifan Wang ; Bahreyni, Behraad, “Touchless capacitive sensor for hand gesture detection”, Sensors, 2011 IEEE, pp.546-549, 28-31 Oct. 2011.
[2] P. Belimpasakis and R. Walsh, "A Combined Mixed Reality and Networked Home Approach to Improving User Interaction with Consumer Electronics," IEEE Transactions on Consumer Electronics, vol. 57, no. 1, pp. 139-144, 2011.
[3] Y. Zhou, Sh. Wang, G. Yu, Z. Yingqiang, "Capacitive sensor highprecision measurement system", 2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME), pp.55-60, 16-18 April 2010.
[4] Z. Radivojevic, et al., “Apparatus, methods and computer program products providing finger-based and hand-based gesture commands for portable electronic device applications”, January, U. S. Patent 0005703, 2008.
[5] http://www.arduino.cc/
[6] http://makezine.com/projects/a-touchless-3d-tracking-interface/
[7] http://www.engineersgarage.com
[8] http://www.instructables.com
[9] https://processing.org/

Prof.Sathya.P

[1] V.V.Ravi Theja Reddy(13BEC0745),B.TECH.
Email-Id:-ravithejareddy99@gmail.com
Phone No. 9629772339.
[2] M.V.Vamsi Krishna(13BEC0594),B.TECH.
Email-Id:-vamsi5987@gmail.com
Phone No. 9787090399.

[3] A.Viswanath Reddy(13BEC00665),B.TECH.
Email-Id:-ambativiswanath1902@gmail.com
Phone No. 8015925405.