

Haptic Blind Spot Alert System

Christopher Nebolsky, Mechanical Engineering Student, *University of South Florida*

Abstract— In everyday transit, the issue of blind spot collisions are a prevalent issue. Currently, there are blind spot detection systems available as factory accessories to manufacturer specific vehicles, but aftermarket detection systems are scarce. Blind spot detection systems all share a common feedback to the driver; visual and/or audio. In this paper, a haptic feedback steering wheel cover will be used in conjunction with ultrasonic sensors to monitor blind spots. This will provide a better alternative to a blind spot alert systems by adding haptic feedback. A preliminary study was performed on human subjects to analyze their response times to different haptic alerts to ensure the most beneficial stimuli is used on the blind spot alert system. However, different modes of vibration showed no significant difference in response time, so the stimuli was chosen in terms of user preference. Subjects explained that they felt a slow pulsing vibration was more effective and distinguishable than a steady vibration.

I. INTRODUCTION

Blind spots are a common nuisance that drivers encounter in every day travel. According to the US Bureau of Transportation Statistics, there are 254.4 million registered passenger vehicles in the United States and approximately 840,000 blind spot collisions occur with 300 fatalities every year [1]. Blind spots are defined as the areas in adjacent lanes of traffic that are blocked by various structures in the automobile or other indirect areas not visible to the driver [2]. Such physical obstructions can potentially cause a collision if a driver performs a lane change when another vehicle is present within the blind spot.

To prevent this from happening, some manufacturers have incorporated blind spot monitoring systems in their vehicles, i.e. Mercedes Benz (BSM), but these systems are factory accessories only installed into their vehicles and not available as an aftermarket product. These blind spot monitoring systems warn the driver by illuminating a light and/or producing an acoustic alert. The driver's visual attention to the road is vital in preventing collisions. Just for perspective, if a driver is traveling at 70 mph and takes their eyes off the road for 2 seconds, they would have traveled 205 ft unattended [2]. By using a warning light, it would require the driver to take their eyes of the road, so the driver is not aware of the environment in front of them.

In order to help prevent blind spot collisions, a haptic blind spot alert system has been produced at the University of South Florida, Tampa, FL, USA. The system utilizes an aftermarket steering wheel cover with multiple vibrations motors, an arduino microcontroller, and two ultrasonic sensors. The alert system detects whether a vehicle is within the blind spot when the turn signal is engaged. If there is a

vehicle present, a haptic vibration is sent to the driver through the steering wheel cover.

Related products for blind spot detection systems and haptic avoidance systems are listed in Section II. The hardware setup and coding are covered in Section III. The experimental setup and procedure is covered in detail in Section IV. The study results are explained in Section V and finally, the conclusions and future recommendations will be covered in Section VI and VII.

II. RELATED PRODUCTS AND SYSTEMS

There are multiple products out today that provide blind spot detection. Volvo offers the "Blind Spot Information System (BLIS)" on their vehicles as an accessory [3]. BLIS utilizes a digital camera to monitor the blind spot region and illuminates a symbol on the mirror housing. Mercedes Benz offers "Blind Spot Assist" also as an accessory on their S and CL-Class vehicles [4]. This system uses a short range radar systems and alerts the driver also by illuminating a symbol located on the side mirror. Toyota also has a blind spot detection system (BSM - Blind Spot Monitor) that works on the same principle as Mercedes and Volvo; illumination of a symbol on the side mirrors. This requires the driver not only to check their side mirrors for approaching vehicles, but also to check for the blind spot symbol. Figure 1 is an image provided by Toyota of the blind spot monitor alert on a Toyota Camry.



Figure 1 Toyota Blind Spot Alert

Select automotive companies offer haptic alerts in some of their vehicles, but none of these systems are for blind spot detection. Mercedes Benz offers a system called "Lane Keep Assist", which monitors road markings for any deviation past a solid line. If the vehicle passes over a solid line, it provides a vibration to the steering wheel to alert the driver[4]. This vibration feature is incorporated into the steering wheel, but is not available for "Blind Spot Assist".

At Kettering University in Michigan, David P. Racine has developed a force feedback steering wheel and gas pedal to help prevent side collisions caused by blind spots [2]. According to their tests results, side collisions caused by blind spots were reduced by 47% in a driving simulator while using his haptic force feedback system. This force feedback system applies a torque to the steering wheel and a force to the gas pedal to prevent the driver from making an improper lane change. This system overrides the control of the vehicle from the driver, however, many individuals want to maintain control at all times. In extreme situations, this system may cause more damage, e.g. if a driver was trying to avoid a stopped vehicle directly in front of them.

For the haptic alert system presented in this paper, collisions are prevented by the driver with the assistance of a haptic response. This haptic response is in the form of a vibration rather than a force feedback that physically pushes you from harm's way. This adds an additional sense(touch) for blind spot recognition, and keeps the drivers attention on the road ahead of them. Another benefit is that this system is universal to any vehicle since the haptic alert is presented through a steering wheel cover.

III. HARDWARE SETUP AND CODING



Figure 2 Hardware Setup

The haptic alert system consists of these simple components:

1. Steering Wheel Cover
2. (4) VPM2 Vibrating Disk Motors
3. Arduino Uno Microcontroller
4. (2) Parallax PING))) Ultrasonic Sensors

A. Steering Wheel Cover Design

The vibration motors are mounted in strategic positions around the steering wheel cover that accommodate the most common hand positions. These locations are the standard "10 and 2", "12 o'clock", and "7 o'clock". The "7 o'clock" position was selected rather than the "5 and 7" due to limited use of the "5 o'clock" position. Most drivers rest their right hands on the gear shift or the upper portion of the wheel. According to the National Highway Traffic Safety Administration, the conventional "10 and 2" position is not safe anymore due to new air bag designs [6]. Fortunately, since the locations of these vibration motors are located all around the steering wheel, the vibrations can be felt throughout the circumference of the cover. The VPM2 vibration motors deliver 1G of vibration strength with a 3V DC supply [7] which is easily sensed and resembles the vibration of a modern cell phone.

B. Arduino Uno Microcontroller Setup

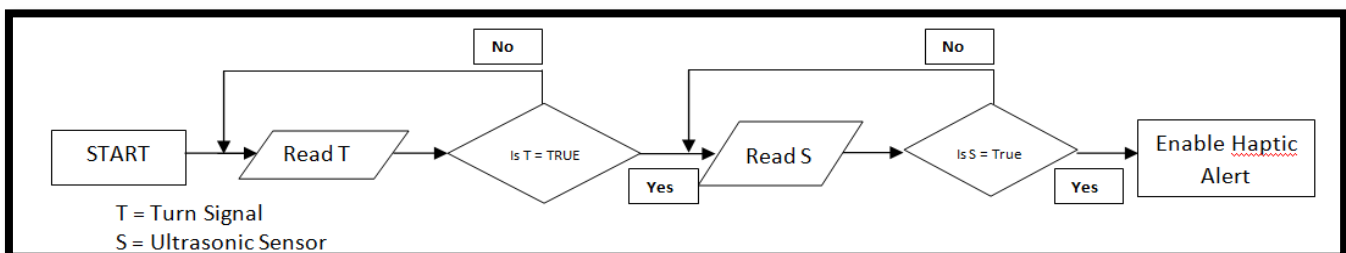
The Arduino Uno microcontroller was selected to control

Figure 3 Microcontroller Algorithm

the electrical components of the system. All coding for the Uno was programmed through Arduino's development environment which uses a simplified version of C/C++ language [8]. The arduino coding uses simple logic to determine when to execute a haptic alert. First, the microcontroller checks whether a turn signal has been enabled. If the turn signal was enabled, it then checks the ultrasonic sensors for a condition. The condition determines if there is an object within a certain range from the sensors. If there is an object present, the microcontroller enables a haptic alert to the steering wheel. Figure 2 below is the basic algorithm that the microcontroller has been programmed. A complete detailed code can be found in the appendix section of this paper.

C. Ultrasonic Sensors

In order for the alert system to operate, there needs to be a means of distance observation within the blind spot region. This is done with the use of ultrasonic sensors which measures distance with an ultrasonic pulse known as sonar [9]. The sensor used for this setup was a Parallax PING))) distance sensor which can measure distances from 3cm to 3m. The ping sensor must be controlled manually by the microcontroller to by generating a short ping and listening for an echo to measure distance. Figure 4 is an image provided by Parallax of how the PING))) operates [9].



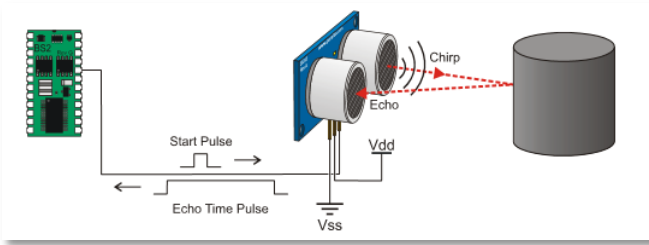


Figure 4 PING))) Sensor

The time it takes for the echo to return to the sensor is directly related to the distance. Generally, a ping of 2 microseconds is pulsed by the sensor. The sound travels at 340 meters per second or 29 microseconds per centimeter. The total time is then divided by two since the pulse travels to and from the object within range, and the distance can be found from the return time. For example, if an object is 2 meters from the sensor, the total time it would take for the sensor to receive an echo is 11.6 milliseconds. This means echo took 5.8 milliseconds to travel back from the object. If we divide 5.8 milliseconds by 29 microseconds, we obtain 200 centimeters or 2 meters. The average highway lane is 3.7 meters wide, and the sensors operating range is 3cm to 3 m. Figure 4 identifies the blind spot range on a standard vehicle, this image was provided by Mercedes for their BSA system [4]. The areas are directly perpendicular to the rear quarter panels of the vehicle. By mounting the sensors on the rear bumper directly behind the rear wheels, the sensors can operate correctly and will detect objects that are present in the adjacent lanes.

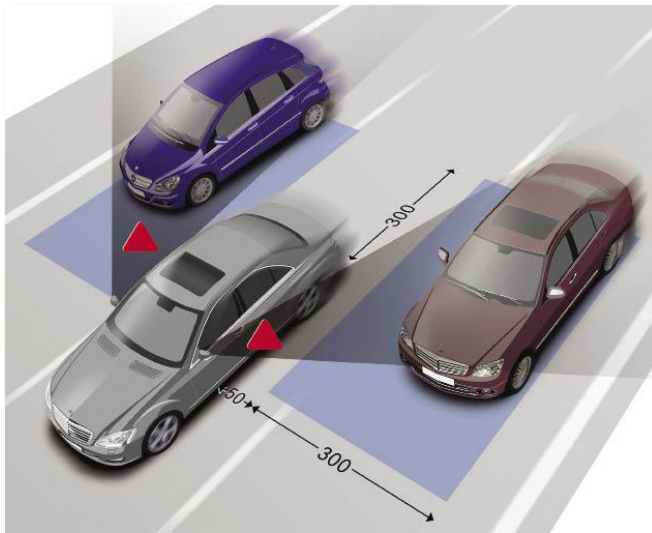


Figure 5 Blind Spot Location

For the operation of the haptic alert system, the distance sensors need to have a form of logic to determine if an object is within the range of the blind spot. The average midsize sedan has a width of 1.8 meters, leaving an estimated 1.9 meters of distance between vehicles on a highway [10]. If an object was to come within 1.9 meters of the ultrasonic sensor when a turn signal is engaged, the microcontroller will alert the driver through the haptic steering wheel cover.

IV. EXPERIMENTAL SETUP AND PROCEDURE

It was hypothesized that sending a pulsing vibration alert to an individual will result in a faster alert and response time. To test this hypothesis, an experimental study program was generated to evaluate the response time of individuals to different vibration alerts.

A. Testing Components

The experimental study was conducted with the following components:

1. Haptic Steering Wheel Cover
2. Arduino Uno Microcontroller
3. Visual Basic Integrated Development Environment

The haptic steering wheel cover was controlled directly by the arduino microcontroller, and the visual basic program works in conjunction with the microcontroller by serial port communication. This allows for a very simple testing apparatus that does not require many components and provides a simple and portable testing station.

B. Testing Program

The testing program was generated using visual basic 2010 express edition, which is an event-driven language development environment. This program presents a user friendly GUI to the test subjects with simple buttons and a video screen. The basic operation of the program was to take basic information from the subject, present a series of haptic alerts at random times, and record the response times of the subjects when they pressed a button labeled "Record Time".

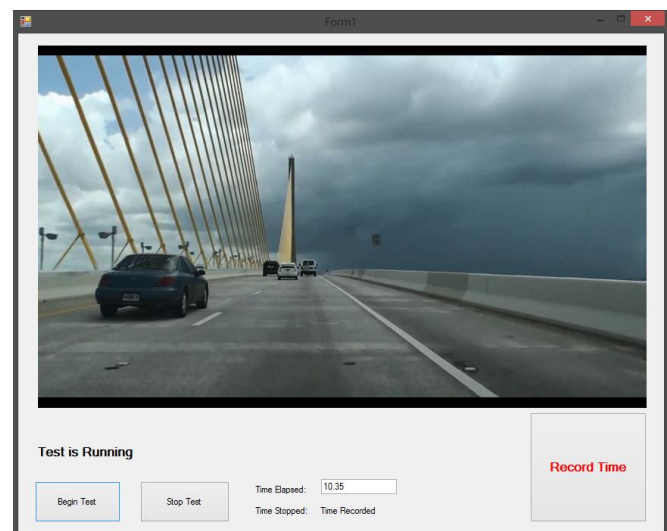


Figure 6 Response Time GUI

The data was written to a text file in real time, and was analyzed by excel for the average time response for each haptic alert. There were three modes presented to the subjects at random times throughout the test, each mode was presented twice for a total of six haptic alerts.

The three modes consisted of:

Table 1Vibration Modes

Mode	Loop	Duration
Steady	Steady for 5 s	5 s
Slow Pulse	On for 500 ms, Off for 500 ms	5 s
Fast Pulse	On for 200 ms, Off for 50 ms	3 s

The vibrations were controlled by the arduino uno within its own coding. Once the subject presses the "Begin Test" button, the visual basic program sends a "1" through the serial port to the arduino. The arduino was coded to look for "1" or "0" to begin or end the test. When it receives a "1", it begins a series of different haptic alerts through the steering wheel cover.

C. Procedure

Each test subject was informed of what to expect from the testing program. The GUI presents a video that runs through the instructions for the study and presents a sample haptic alert to the subject before the test begins. In order to obtain accurate response time results, a video of a vehicle traveling down an interstate was shown to the subject to distract them visually. Below are the procedures given to each subject:

1. Comfortably seat yourself in front of the computer.
2. Enter age and gender into dialog widow.
3. Grasp the steering wheel with your least dominant hand.
4. Click "Begin Test"
5. Click "Record Time" every time a haptic alert is presented
6. When signaled test has completed, click "End Test".

V. RESULTS

13 subjects participated in the study, ranging from 21 to 56 years old. A total of 78 responses were recorded (13 participants x 3 response time tasks x 2 trials) for haptic stimuli alerts. Appropriate statistical analysis was performed by means of ANOVA to study if gender or age had significant effects on response times.

Analysis of Variance					
Source	Sum Sq.	D.F.	Mean Sq.	F	Prob>F
Subject	1.41994	12	0.11833	9.05	0
Vibration Mode	0.04656	2	0.02328	1.78	0.1902
Error	0.31384	24	0.01308		
Total	1.78034	38			

After completing the statistical analysis, it was determined that there is no significant difference between vibrations modes when it comes to alert response times ($F(2,38) = 1.78$, $p=0.1902$). There were some differences between subjects and their respected response times ($F(12,38) = 9.05$, $p = 0$), but no factor set them apart from the study group (2 subjects had slower response times, but varied in age by 30 years).

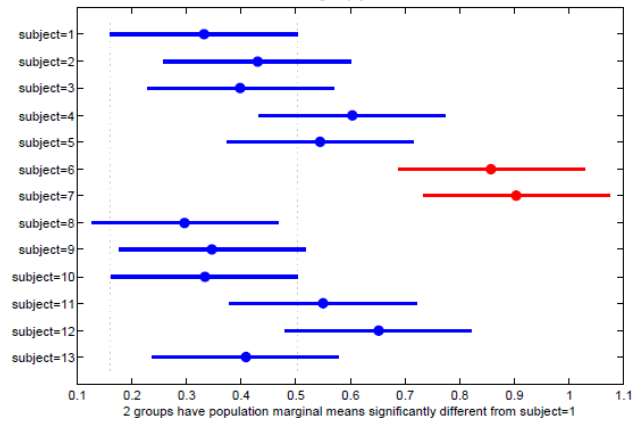


Figure 7 Subject Comparison

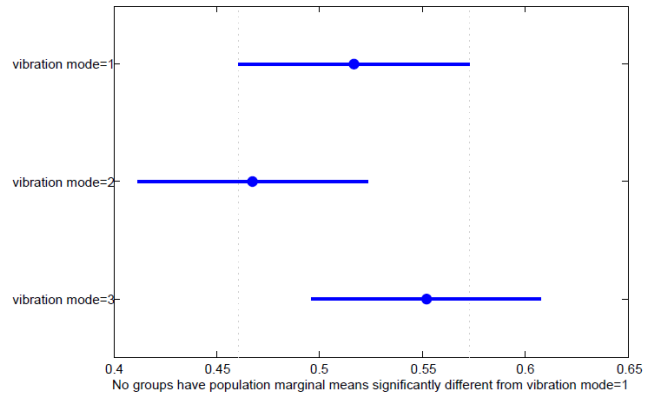


Figure 8 Vibration Mode Comparison

According to a study conducted in [11], the average response time to visual stimuli is 0.517ms and 0.493ms for auditory stimuli. The average response time to haptic alerts in this study was 0.484ms. This shows that the use of a haptic alerts is just as effective as visual and audio alerts. However, haptic alerts should not be the only stimuli presented to a driver when operating a vehicle.

VI. CONCLUSION

With the use of haptic alerts for blind spot monitoring, the driver has an added sense of security when performing lane changes. When a driver is operating a motor vehicle on a highway setting, visual attention is a priority, so to minimize any unnecessary instances of visual distraction benefits the driver's safety. The driver can anticipate a haptic alert from the blind spot alert system before making a visual verification to ensure a proper lane change with no collisions. According to the study that was performed, the response rate to haptic stimuli is just as effective as visual and audio feedback.

Having this product available to the average consumer will effectively decrease collisions caused by improper lane changes. Since the haptic blind spot alert system is universal to any vehicle, it can be sold as an aftermarket product with little modification with installation. No longer will consumers have to purchase specific vehicle brands to obtain blind spot detection features. The cost of production will be

low due to the simple components used in the system, which could lead to excellent profitability.

VII. FUTURE RECOMMENDATIONS

This haptic blind spot alert system can be further modified and enhanced by using different components to amplify the quality and reliability of the product. Some of these features could include:

1. Replacing the Ultrasonic Sensor
2. Utilizing System for other Haptic Features

Instead of using ultrasonic sensors to monitor concentrated areas, a short range radar sensor can be used to monitor the surrounding area of the vehicle. Ultrasonic sensors are easily interfered by multiple environmental factors, which could include rain or dirt. If radar is used in combination with an effective processor, multiple features can be integrated into the system. One of these features could include, back up assist, navigation in poor visual conditions, and collision prevention. For example, if the driver forgets to signal a lane change and proceeds to move into a lane occupied by another vehicle. If the system detects that another vehicle is within a range of comfort, it can automatically send an alert to the driver.

Other features could also dictate how the system operates if certain conditions are met, i.e. vehicle speed or location. Location can be found under the use of a GPS, and if a vehicle is on a designated highway or interstate, the system will know to enable the blind spot alert system. This way the system does not only depend on vehicle speed since speed can be very minimal on interstates during rush hours.

REFERENCES

- [1] *Transportation Statistics Annual Report*. Bureau of Transportation Statistics. Web. 9 April 2013. <http://www.rita.dot.gov/bts/sites/rita.dot.gov/bts/files/publications/transportation_statistics_annual_report/index.html>
- [2] Racine, D. P.; Cramer, N. B.; Zadeh, M.H., "Active blind spot crash avoidance system: A haptic solution to blind spot collisions," IEEE International Symposium on Haptic Audio-Visual Environments and Games (HAVE), pp 1-5, 2010.
- [3] Hanlen, M., *Volvo Launches Blind Spot Information System (BLIS)* Gizmag. Web. 4 April 2013. <<http://www.gizmag.com/go/2937/>>
- [4] *Mercedes Benz Offers Blind Spot Assist*. Mercedes-Benz Dealer News. Web. 4 April 2013. <http://www.mercedesdealernews.com/my_weblog/2007/09/mercedes-offers.html>
- [5] *Mercedes Benz collision avoidance features*. Highway Loss Data Institute. Bulletin. Vol. 29, No.7. April 2012.
- [6] *Get with the times: You're driving all wrong*. NBC News. Web. 9 April 2013. <<http://www.nbcnews.com/business/get-times-youre-driving-all-wrong-518710>>
- [7] *VPM2 Flat Type Vibration Motor Specifications*. Solarbotics. Datasheet. Web. 9 April 2013. <<http://www.robotshop.com/content/PDF/datasheet-vpm2.pdf>>
- [8] *Arduino Uno R3*. Arduino. Web. 9 April 2013. <www.arduino.cc>
- [9] *PING))) Ultrasonic Distance Sensor*. Parallax. Specification Sheet. Web. 9 April 2013. <<http://www.parallax.com/Portals/0/Downloads/docs/prod/acc/28015-PING-Sensor-Product-Guide-v2.0.pdf>>

- [10] Woodyard, C. *Vehicles keep inching up and putting on pounds*. USA Today. Web. 10 April 2013. <http://usatoday30.usatoday.com/money/autos/2007-07-15-little-big-cars_N.htm>
- [11] Ng, A.; Chan, A., "Finger Response Times to Visual, Auditory and Tactile Modality Stimuli," Proceedings of The International MultiConference of Engineers and Computer Scientists 2012, pp1449-1454, 2012.

APPENDIX

1. Arduino Alert System Coding (Attached)
2. Visual Basic Dialog Window Coding (Attached)
3. Visual Basic Study Program Coding (Attached)
4. Arduino Study Program Coding (Attached)
5. MATLAB ANOVA Coding (Attached)
6. MATLAB ANOVA Figures (Attached)

Final Arduino Code

```
const int pingPin = 8; // Designating Pin 8 for Ping Sensor
const int distance = 190; // Alert distance in cm
const int ledPin = 13; // Designating Pin 13 for LED
const int Switch = 9; // Designating Pin 9 for Switch
int ledState = LOW;
long previousMillis = 0;
long interval = 500;

void setup()
{
  Serial.begin(9600); // Begins serial port communication
  DDRD = B11110000; // Setting pins 4-7 as OUTPUTS (5V)
  pinMode(Switch, INPUT); // Setting switch pin as INPUT
  digitalWrite(Switch, HIGH); // Setting Switch Pin as Pull Up Resistor
  pinMode(ledPin, OUTPUT); // Setting Led pin as OUTPUT
}

void loop()
{
  long duration, cm; // establish variables for duration of the ping and the
  distance result in centimeters

  while(digitalRead(Switch) == LOW) // Operates only when Switch is active.
  {
    unsigned long currentMillis = millis();

    if(currentMillis - previousMillis > interval) // Flashes turn signal
    LED
    {
      previousMillis = currentMillis; // save the last time you blinked
      the LED

      if (ledState == LOW) // if the LED is off turn it on and
      vice-versa:
        ledState = HIGH;
      else
        ledState = LOW;

      digitalWrite(ledPin, ledState); // set the LED with the ledState
      of the variable:
    }

    // Operating the PING)) Sensor
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW); // Give a short LOW pulse beforehand
    to ensure a clean HIGH pulse:
    delayMicroseconds(2);
  }
}
```

```

                                Final Arduino Code
HIGH pulse of 5 microseconds.
                                digitalWrite(pingPin, HIGH); // The PING))) is triggered by a
HIGH
                                delayMicroseconds(5);
                                digitalWrite(pingPin, LOW);

                                // The same pin is used to read the signal from the PING))) : a
HIGH
                                // pulse whose duration is the time (in microseconds) from the
sending
                                // of the ping to the reception of its echo off of an object.
                                pinMode(pingPin, INPUT);
                                duration = pulseIn(pingPin, HIGH);

                                // convert the time into a distance
                                cm = microsecondsToCentimeters(duration);

                                Serial.print(cm);
                                Serial.print("cm");
                                Serial.println();

                                delay(100);

of a sensor
                                if (cm <= distance) // Logic for if an object is within the range
                                {
                                    PORTD = B11110000;
                                    delay(500);
                                    PORTD = B00000000;
                                    delay(400);
                                }
                                else // Do nothing if no objects are present in sensor's range.
                                {
                                    PORTD = B00000000;
                                }

                                }
                                digitalWrite(ledPin, LOW); // If switch is not active, Turn off LED
                                PORTD = B00000000; // If switch is not active, No haptic alert
                                }

// Calculating centimeters from time.
long microsecondsToCentimeters(long microseconds)
{
    // The speed of sound is 340 m/s or 29 microseconds per centimeter.
    // The ping travels out and back, so to find the distance of the
    // object we take half of the distance travelled.
    return microseconds / 29 / 2;
}

```

Imports System.Windows.Forms

Public Class Dialog1

Dim Gender As String

Dim Age As Integer

```
Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
    My.Computer.FileSystem.WriteAllText("C:\Users\Christopher\Documents\Chris School\Haptics\Project\Arduino Projects\ComPortTest\test.txt", Gender & Age & ", ", True)
    Me.DialogResult = System.Windows.Forms.DialogResult.OK
    Me.Close()
End Sub
```

```
Private Sub Cancel_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel_Button.Click
    Me.DialogResult = System.Windows.Forms.DialogResult.Cancel
    MainForm.Close()
    Me.Close()
End Sub
```

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
    Gender = "Male,"
End Sub
```

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
    Gender = "Female,"
End Sub
```

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    Age = TextBox1.Text
End Sub
```

End Class


```
Imports System.IO
Imports System.IO.Ports
Imports System.Threading
```

```
'Main Form Coding
Public Class MainForm
```

```
    Shared _continue As Boolean
    Shared _serialPort As SerialPort
    Dim Stopwatch As New System.Diagnostics.Stopwatch
```

```
'Opens Serial Port for Arduino Communication
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Dim startform As New Dialog1
    startform.Show() 'Shows Dialog Box when starting program
    startform.TopMost = True 'Brings Dialog Box to the front of the screen
    SerialPortx.PortName = "com4" 'Enter Com port for Arduino
    SerialPortx.BaudRate = 9600 'Enter Baud Rate for Arduino
    SerialPortx.DataBits = 8
    SerialPortx.Parity = IO.Ports.Parity.None
    SerialPortx.StopBits = IO.Ports.StopBits.One
    SerialPortx.Open() 'Opens Arduino Serial Port for communication
End Sub
```

```
'Begin Test Button Press
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1. Click
    SerialPortx.Write("1") 'Writes "1" to Arduino
    Label2.Text = "Test is Running" 'Display to user to designate test is running
    AxWindowsMediaPlayer1.Ctlcontrols.play() 'Prompts media player to play video
    Stopwatch.Start() 'Begins timer for count recording
    Timer1.Interval = 100
    Timer1.Enabled = True
End Sub
```

```
End Sub
```

```
'End Test Button Press
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2. Click
    SerialPortx.Write("0") 'Writes "0" to Arduino
    Label2.Text = "Test Has Ended" 'Display to user to designate test has ended
    AxWindowsMediaPlayer1.Ctlcontrols.stop() 'Prompts media player to stop video
    Stopwatch.Stop() 'Ends timer count
    Stopwatch.Reset() 'Resets timer
    My.Computer.FileSystem.WriteAllText("C:\Users\Christopher\Documents\Chris School\Haptics\Project\ Arduino Projects\ComPortTest\test.txt", " ; ", True) 'Writes to text file ";" to designate end of test.
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
```

```
    If Stopwatch.IsRunning Then 'Get the elapsed time as a TimeSpan value.
```

```
        UpdateDisplay(Stopwatch.Elapsed)
```

```
    End If
```

```
End Sub
```

```
'Controls timer text box format and display
```

```
Sub UpdateDisplay(ByVal ts)
```

```
    StopwatchTextBox.Text = String.Format("{0:00}.{1:00}", ts.Seconds, ts.Milliseconds / 10)
```

```
End Sub
```

```
'Record Time Button Press
```

```
Private Sub TimeRecord_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TimeRecord.Click
```

```
Label1.Text = StopwatchTextBox.Text      'Writes time to label if the button is pressed
My.Computer.FileSystem.WriteAllText("C:\Users\Christopher\Documents\Chris School\Haptics\Project\
Arduino Projects\ComPortTest\test.txt", _
    StopwatchTextBox.Text & ",", True)    'Writes to text file the time recorded when button is
pressed seperated by a "," for easy excel import.
End Sub
End Class
```

Study Arduino Code

```
void setup()
{
  Serial.begin(9600); // Setting Baud Rate
  DDRD = B11110000; // Setting pins 4 - 7 as OUTPUTS (5V)
}

void loop()
{
  while (Serial.available() == 0);
  int val = Serial.read() - '0'; // Converting value to read
  if (val == 1) // Reading if "1" was sent from Visual Basic Program to begin Test
  {
    delay(16900);
    PORTD = B11110000; // Example Alert (3s) - Writing pins 4-7 as HIGH
    delay(3000);
    PORTD = B00000000; // Writing pins 4-7 as LOW
    delay(27000);
    PORTD = B11110000; // Steady Alert (5s)
    delay(5000);
    PORTD = B00000000;
    delay(15000);
    for(int count = 0; count<5; count++) // Slow Pulse (5s)
    {
      PORTD = B11110000;
      delay(500);
      PORTD = B00000000;
      delay(500);
    }
    delay(8000);
    for(int count = 0; count<12; count++) // Fast Pulse (3s)
    {
      PORTD = B11110000;
      delay(200);
      PORTD = B00000000;
      delay(50);
    }
    delay(12000);
    for(int count = 0; count<5; count++) // Slow Pulse (5s)
    {
      PORTD = B11110000;
      delay(500);
      PORTD = B00000000;
      delay(500);
    }
    delay(18000);
    for(int count = 0; count<12; count++) // Fast Pulse (3s)
```

Study Arduino Code

```
{
  PORTD = B11110000;
  delay(200);
  PORTD = B00000000;
  delay(50);
}
delay(9000);
PORTD = B11110000; // Steady Pulse (5s)
delay(5000);
PORTD = B00000000;
}
else if (val == 0) // If "0" is received from VB, Test has ended.
{
  Serial.println("Test Ended");
  PORTD = B00000000;
}
else // Invalid Input received
{
  Serial.println("Ooopss..!!");
  PORTD = B00000000;
}
Serial.flush();
}
```

```
close all
```

```
clear all
```

```
% Time, Subject, Vibration Mode, Age
```

```
data = [
```

```
0.33 1 1;
```

```
0.335 1 2;
```

```
0.33 1 3;
```

```
0.355 2 1;
```

```
0.47 2 2;
```

```
0.465 2 3;
```

```
0.71 3 1;
```

```
0.215 3 2;
```

```
0.27 3 3;
```

```
0.565 4 1;
```

```
0.575 4 2;
```

```
0.67 4 3;
```

```
0.56 5 1;
```

```
0.515 5 2;
```

```
0.56 5 3;
```

```
0.8 6 1;
```

```
0.645 6 2;
```

```
1.13 6 3;
```

```
0.81 7 1;
```

```
0.82 7 2;
```

```
1.085 7 3;
```

```
0.325 8 1;
```

```
0.235 8 2;
```

```
0.33 8 3;
```

```
0.4 9 1;
```

```
0.295 9 2;
```

```
0.345 9 3;
```

```
0.295 10 1;
```

```
0.305 10 2;
```

```
0.4 10 3;
```

```
0.57 11 1;
```

```
0.57 11 2;
```

```
0.51 11 3;
```

```
0.595 12 1;
```

```
0.685 12 2;
```

```
0.675 12 3;
```

```
0.405 13 1;
```

```
0.415 13 2;
```

```
0.405 13 3;
```

```
];
```

```
[p table stats] = anovan(data(:,1), {data(:,2) data(:,3)}, 'varnames',  
{'subject', 'vibration mode'})
```

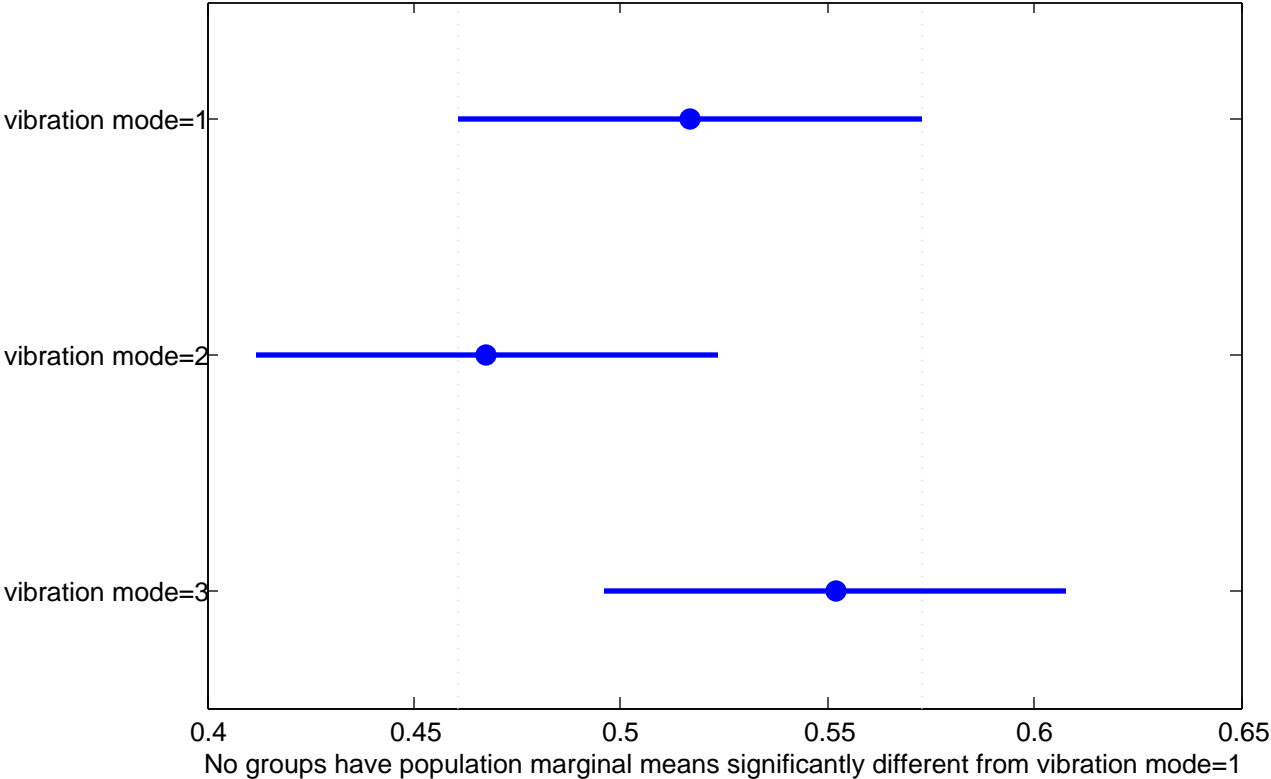
```
pause
```

```
% compare differences between subjects - not typically important to know, but should be  
in the ANOVA
```

```
multcompare(stats, 'dim', 1)
```

```
% compare differences between experiment type  
figure  
[comp8 mean8] = multcompare(stats, 'dim', 2);
```

Click on the group you want to test



Click on the group you want to test

