Element G: Construction of a testable prototype

*To create our final design, we decided to combine two separate concepts. These two concepts consisted of the "Critical and Noncritical Display," and the "Repair Shops/Mechanics" ideas. We began researching Bluetooth technology. By researching Bluetooth technology, we discovered that the technology to fulfill the second part of our design does exist. We learned that Bluetooth is wireless and automatic. It works on a protocol system where there are various commands and responses. Thus, it works on two levels: physical and protocol. The physical level is radio-frequency standard. Secondly, there is the protocol level where different products have to agree on bits sent, how many bits sent at a time, and how the parties in conservation can be sure that the message is received in the same message sent. In Bluetooth networking there is data sent via low-power radio waves. These frequencies allow different Bluetooth abled devices to connect. For, devices to connect there must be universal protocols which allows for a successful connection**[1]**.*

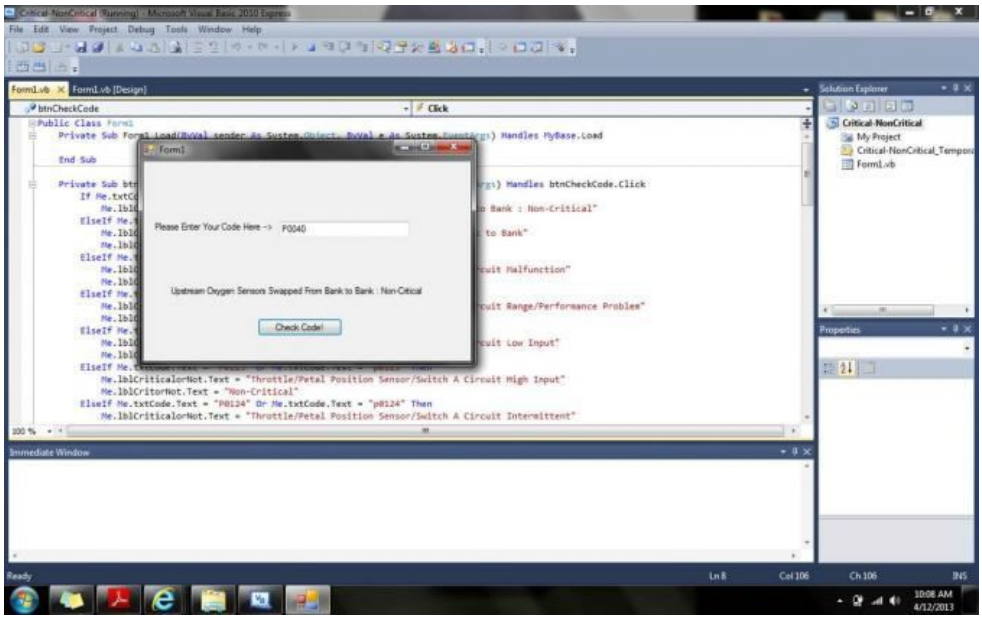
However, with our limited time we were not able to learn all there is about Bluetooth technology to complete this task. Therefore, we focused our energy on the "Critical/Noncritical" display. To our knowledge, the display of this information does not exist which is why we used our efforts to complete and perfect this unique concept.

Design:

- We chose to design a Visual Basic Program to demonstrate our display idea. This program was the best choice considering time constraints and easy access to the program software. The program allowed us to convey a prototype environment of what the ideal Android app would look like. Additionally, it exemplified how the "Critical" and "Non-Critical" elements would be incorporated into our final "dream" design.

Construction of Prototype:

- We downloaded the Visual Basic Express software from Microsoft**[2]**, and implemented coding. Our coding would allow its user to input data retrieved (diagnostic code) from the "Torque Application**[3]**." This Torque app was a downloaded app from the Android Store that is able to communicate with the Bluetooth OBDII reader and generate a diagnostic code. From this app, we had access to the trouble code in the car. This code would then be typed into a text box on the Visual Basic Program. In the end, our goal is to make an app that leaves out the step of typing in the code into the program. Instead, the code would be generated on the app along with this Critical/Noncritical Information. After the code was inserted, the coding looked for the correct "match" and populated the information in an organized layout with the "Critical" and "Non-Critical" section added to a basic description of the code itself.



- To show this idea and test to see if our product works, we used ten critical diagnostic codes and ten noncritical codes. We established these labels alongside our mentor (Mr. Thomas Kennedy Sr.) who provided us insight on why certain codes create a more hazardous environment while others are minor faults within the vehicle. All the noncritical codes included car diagnostic codes that would not cause any immediate damage to the vehicle or create driving hazards. The noncritical code would alert the driver of a problem, but the car would not need immediate attention. The critical codes included any "misfires" which would cause more damage to the vehicle or danger to the driver. The critical description would alert the driver of the problem, and require the immediate attention to the vehicle.
    - Noncritical Codes:
        - P0040, P0041: These codes are noncritical because the driver has typically three to five oxygen sensors throughout the exhaust system. Therefore, when one oxygen sensor faults the other sensors still are able to function.
        - P0120, P0121, P0122, P0123, P0124, P0220, P0221, P0222: These codes are noncritical because they describe a fault in the throttle/pedal position sensor. The acceleration pedal is connected to the throttle cable attached to the carburetor. This system allows airflow to continuously mix with the fuel to complete the air to fuel ratio needed for proper combustion. A sensor only allows the vehicle to keep the revolutions per minute at low rate and fuel efficient speed. These codes are noncritical due to the fact that the "choke (valve that allows air to pass through the intake manifold into the engine itself)" is still open with or without a sensor.
    - Critical Codes:
        - P0300, P0301, P0302, P0303, P0304, P0305, P0306, P0307, P0308, P0309, P0310, P0311, P0312: These codes are

critical because a cylinder misfire described by these codes can lead to a blown engine. For, it can destroy the vehicles crank shaft, cam shaft, pistons, pistons rings, and connecting rods. These materials are crucial to engine performance. If a cylinder misfire is not resolved, other malfunctions with occur with the vehicle.

- In the Program itself, the codes were inputted in If… Then… statements that organized the information by the diagnostic codes. If… Then… statements occur such as this: IF the user types in the trouble code, THEN the Visual Basic Express coding with search for that text in the program itself and populates the given response. Thus, by inputting the code the program finds its corresponding information of a basic description and critical/noncritical.

Visual Basic Code:

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub btnCheckCode_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnCheckCode.Click
        If Me.txtCode.Text = "P0040" Or Me.txtCode.Text = "p0040" Then
            Me.lblCriticalorNot.Text = "Upstream Oxygen Sensors Swapped From Bank to Bank"
            Me.lblCritorNot.Text = "Non-Critical"
        ElseIf Me.txtCode.Text = "P0041" Or Me.txtCode.Text = "p0041" Then
            Me.lblCriticalorNot.Text = "Downstream Oxygen Sensors Swapped From Bank to Bank"
            Me.lblCritorNot.Text = "Non-Critical"
        ElseIf Me.txtCode.Text = "P0120" Or Me.txtCode.Text = "p0120" Then
            Me.lblCriticalorNot.Text = "Throttle/Petal Position Sensor Switch A Circuit Malfunction"
            Me.lblCritorNot.Text = "Non-Critical"
        ElseIf Me.txtCode.Text = "P0121" Or Me.txtCode.Text = "p0121" Then
            Me.lblCriticalorNot.Text = "Throttle/Petal Position Sensor/Switch A Circuit Range/Performance Problem"
            Me.lblCritorNot.Text = "Non-Critical"
        ElseIf Me.txtCode.Text = "P0122" Or Me.txtCode.Text = "p0122" Then
            Me.lblCriticalorNot.Text = "Throttle/Petal Position Sensor/Switch A Circuit Low Input"
            Me.lblCritorNot.Text = "Non-Critical"
        ElseIf Me.txtCode.Text = "P0123" Or Me.txtCode.Text = "p0123" Then
            Me.lblCriticalorNot.Text = "Throttle/Petal Position Sensor/Switch A Circuit High Input"
```

```
        Me.lblCritorNot.Text = "Non-Critical"
    ElseIf Me.txtCode.Text = "P0124" Or Me.txtCode.Text = "p0124" Then
        Me.lblCriticalorNot.Text = "Throttle/Petal Position Sensor/Switch A Circuit Intermittent"
        Me.lblCritorNot.Text = "Non-Critical"
    ElseIf Me.txtCode.Text = "P0220" Or Me.txtCode.Text = "p0220" Then
        Me.lblCriticalorNot.Text = "Throttle/Petal Position Sensor/Switch B Circuit Malfunctoin"
        Me.lblCritorNot.Text = "Non-Critical"
    ElseIf Me.txtCode.Text = "P0221" Or Me.txtCode.Text = "p0221" Then
        Me.lblCriticalorNot.Text = "Throttle/Petal Position sensor/Switch B Circuit Range Performance Problem"
        Me.lblCritorNot.Text = "Non-Critical"
    ElseIf Me.txtCode.Text = "P0222" Or Me.txtCode.Text = "p0222" Then
        Me.lblCriticalorNot.Text = "Throttle/Petal Position Sensor/Switch B Circuit Low Input"
        Me.lblCritorNot.Text = "Non-Critical"
    ElseIf Me.txtCode.Text = "P0300" Or Me.txtCode.Text = "p0300" Then
        Me.lblCriticalorNot.Text = "Random/Multiple Cylinder Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0301" Or Me.txtCode.Text = "p Then" Then
        Me.lblCriticalorNot.Text = "Cylinder 1 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0302" Or Me.txtCode.Text = "p0302" Then
        Me.lblCriticalorNot.Text = "Cylinder 2 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0303" Or Me.txtCode.Text = "p0303" Then
        Me.lblCriticalorNot.Text = "Cylinder 3 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0304" Or Me.txtCode.Text = "p0304" Then
        Me.lblCriticalorNot.Text = "Cylinder 4 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0305" Or Me.txtCode.Text = "p0305" Then
        Me.lblCriticalorNot.Text = "Cylinder 5 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0306" Or Me.txtCode.Text = "p0306" Then
        Me.lblCriticalorNot.Text = "Cylinder 6 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0307" Or Me.txtCode.Text = "p0307" Then
        Me.lblCriticalorNot.Text = "Cylinder 7 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0308" Or Me.txtCode.Text = "p0308" Then
        Me.lblCriticalorNot.Text = "Cylinder 8 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0309" Or Me.txtCode.Text = "p0309" Then
        Me.lblCriticalorNot.Text = "Cylinder 9 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0310" Or Me.txtCode.Text = "p0310" Then
        Me.lblCriticalorNot.Text = "Cylinder 10 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0311" Or Me.txtCode.Text = "p0311" Then
        Me.lblCriticalorNot.Text = "Cylinder 11 Misfire"
        Me.lblCritorNot.Text = "Critical"
    ElseIf Me.txtCode.Text = "P0312" Or Me.txtCode.Text = "p0312" Then
        Me.lblCriticalorNot.Text = "Cylinder 12 Misfire"
        Me.lblCritorNot.Text = "Critical"
```

Refinements

- Our first prototype consisted of a text box for user input. A label then displayed a single line response with the “Critical” or “Non-Critical” information displayed at the end of the description of the code when a button was pushed. After we tested this program, it was noticed that users were not getting the information they truly needed in a safe amount of time. In this test, we discovered that users were taking too long to read the code, and some users did not understand the given information. We wanted to minimize the possible distractions our prototype would create for the driver, and wanted the driver to be able to understand whether their diagnostic code is critical or noncritical. Therefore, we made the “Critical” or “Non-Critical” elements of the Visual Basic program the first thing the user notices by bolding, italicizing, and changing the color of the word to red.

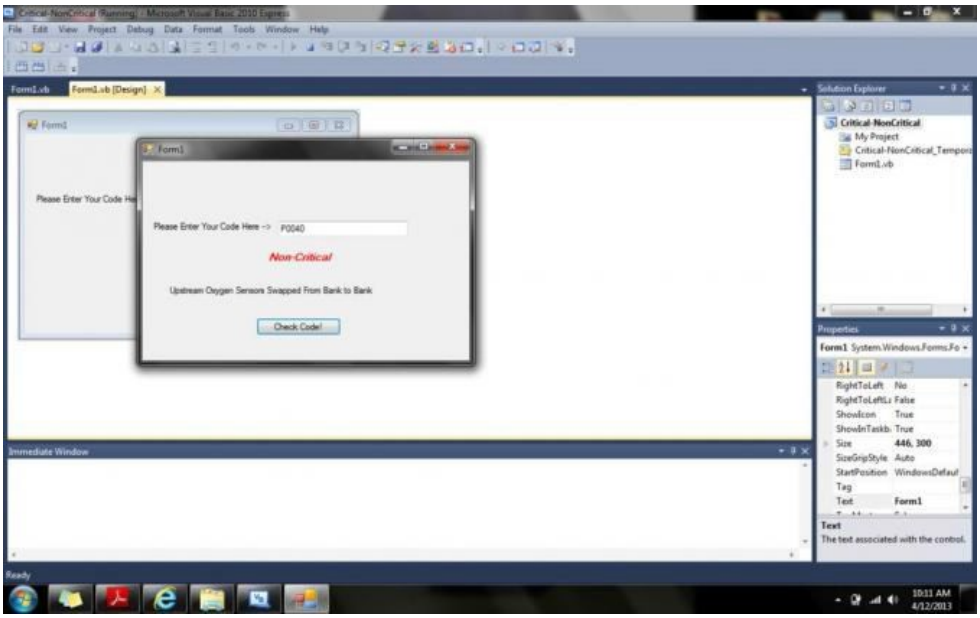
Here the revised prototype shows the “Noncritical” or “Critical” in the first line, and is now red. Thus, this information is what attracts the driver.

Materials for Design

- Our materials did stay the same throughout the project, simply because the solution revolves around a program written in code. The “Torque” App and the Visual Basic program were the only two elements needed to complete and implement our final design. Our dream goal would be to incorporate everything into one easy to use Android, Apple, or Google App without the extra step of inputting the diagnostic code. This input would be skipped by this APP which would display the needed information (Critical or noncritical display) when the APP would communicate with the Bluetooth OBDII reader. Thus, when the APP receives the diagnostic code, it would display the code alongside the critical or noncritical information.

Here is the video of the entire process of our final prototype:

http://w ww.yo utube.com/watch?v=mQ SeIXKAEPE (http://www.youtube.com/watch?v=mQSeIXKAEPE)



[1] Layton, Julia, and Curt Franklin. "How Bluetooth Works." Howstuffworks.com. How Stuff Works, n.d. Web. 5 Mar. 2013.

[2] "Visual Basic 2010 Express." - Overview. N.p., n.d. Web. 07 Feb. 2013.

[3] "Torque Pro (OBD 2 & Car)." - Android Apps on Google Play. N.p., n.d. Web. 24 Apr. 2013.