Logan Brincks, Steven Karrmann, John Polus

4/6/2017

Design Document

Lab 5 – Breathalyzer Circuit

Abstract

The following document describes the assumptions made, derived equations and justification for the design properties of SE4130 Lab 5 submitted in the Spring Semester of 2017 for the class “Real Time and Embedded Systems” at the University of Wisconsin-Platteville directed by Dr. Joe Cliffton

Contents

[Requirement 1 2](#_Toc479178756)

[Requirement 2 4](#_Toc479178757)

[Testing the LCD: 4](#_Toc479178758)

[Testing the Digital Potentiometer: 4](#_Toc479178759)

[Testing the ADC: 4](#_Toc479178760)

[Testing the Dip Switches: 4](#_Toc479178761)

[Expected values of Digital Potentiometer: 4](#_Toc479178762)

[Requirement 3 5](#_Toc479178763)

[Requirement 4 6](#_Toc479178764)

[Requirement 5 7](#_Toc479178765)

[Requirement 6 8](#_Toc479178766)

[Table for BAC ADC indexing: 8](#_Toc479178767)

[Requirement 7 9](#_Toc479178768)

[Equation for temperature adjustment of ADC Blood Alcohol Value: 9](#_Toc479178769)

[Requirement 8 10](#_Toc479178770)

[Requirement 9 11](#_Toc479178771)

[Equation for Celsius from ADC values: 11](#_Toc479178772)

[Equation for Fahrenheit from ADC values: 11](#_Toc479178773)

[Requirement 10 12](#_Toc479178774)

[Testing and Calibration: 13](#_Toc479178775)

# Requirement 1

**Definitions:**

|  |  |  |
| --- | --- | --- |
| Value | Definition | Metric |
|  | The voltage across the load resistor | Volts |
|  | The resistance of the load resister | Ω |
|  | The current through the load resistor | Amps |
|  | The voltage across the sensor | Volts |
|  | The resistance of the sensor | Ω |
|  | The current through the sensor | Amps |

**Given:**

Ohm’s Law:

The voltage across a circuit element is equal to the product of the resistance of that element and the current running through that element.

Kirchhoff’s Current Law: .

The current flowing into a circuit node must equal the current flowing out of the circuit node.

Kirchhoff’s Current Law Corollary:

If there are *n* resistors in series, the total resistance across that series, , is equal to the sum of all of the resistances in that series.

**Derivation of VRL:**

Figure 1 may be simplified to Figure 2 for a more straight-forward graphical representation of the problem.

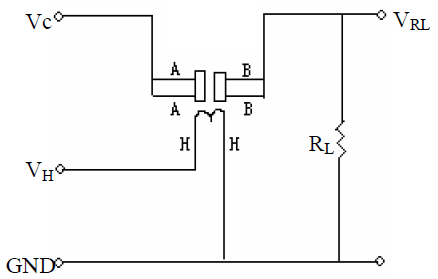


Figure 1 The circuit diagram for the MQ3 alcohol sensor.

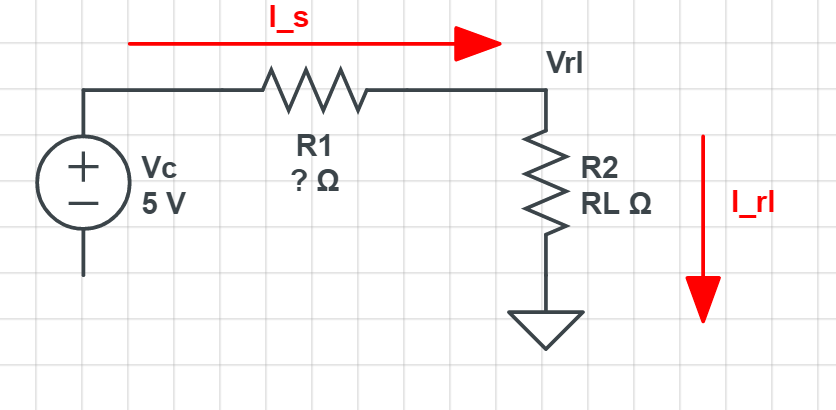


Figure 2 The simplified circuit diagram.

So, by Ohm’s Law, .

|  |  |
| --- | --- |
| Equation | Operation |
|  | Ohm’s Law |
|  | Kirchhoff’s Current Law |
|  | Algebraic substitution |
|  | Distributive Property |
|  | Algebra |
|  | Algebra |
|  | Express in terms of an ADC value |

This is the voltage across the load resistance expressed in terms of an ADC value. Solving this equation for results in the following:

|  |  |
| --- | --- |
| Equation | Operation |
|  | Voltage in terms of an ADC value. |
|  | Multiply by |
|  | Add |
|  | Multiply by |

# Requirement 2

## Testing the LCD:

To properly test the LCD the following procedure should be followed:

|  |  |  |
| --- | --- | --- |
| Procedure | Expected Results | Visual Representation |
| 1. Include LCD.c and LCD.h into a new project |  |  |
| 1. Create a main method and call the InitializeLCD(); method | LCD should initialize and a blinking cursor should be visible on the top line on the left. |  |
| 1. Use the SendData() method to send an individual char to the [SendData(“A”);] | An “A” Character should appear in the first position and the cursor should advance |  |
| 1. Create an array of chars to send as a string:   char s1[6] = “String”; |  |  |
| 1. Send the String using the command [SendString(s1);] | The string should be appended to the end of the initial “A” from step 3 and should now display “AString” |  |
| 1. Move the cursor position to the second line using the command [SetCursor(1, 0);] | The blinking cursor should now be on the second line of the LCD. |  |
| 1. Send the String using the command [SendString(s1);] | The string should be added to the second row and the display should read:  “AString  String” |  |
| 1. Return the cursor to the first position on the top row of the LCD using the command: [ReturnHome();] | The cursor should now be blinking in the first position of the top row. |  |
| 1. Clear the display using the command [ClearDisplay();] | The display should clear and the cursor should be in the first positon of the top row. |  |

## Testing the Digital Potentiometer:

To test the Digital Potentiometer the following procedure should be followed:

|  |  |
| --- | --- |
| Procedure | Observed Results |
| 1. Include DigitalPOT.c and DigitalPOT.h in a new project | N/A |
| 1. Uncomment “define POT\_TESTING” | N/A |
| 1. Program the PIC with these two files. | N/A |
| 1. Using a digital multimeter, measure the resistance across pin 7 (potentiometer 1 output) and the ground. Record the output in the nearest kOhm to the right. |  |
| 1. Wait 5 seconds and record the new value |  |
| 1. Wait 5 seconds and record the new value |  |
| 1. Wait 5 seconds and record the new value |  |
| 1. Wait 5 seconds and record the new value |  |
| 1. Wait 5 seconds and record the new value |  |
| 1. Verify the first value recorded in step 4 is the same as the value recorded in step 9. |  |
| 1. Verify the recorded voltages relate to the table in “Expected values of Digital Potentiometer – Value” |  |

## Testing the ADC:

Note: When setting the DIP Pins, pin 1 is the high order bit and pin 6 is the low order bit. Pins 7 and 8 are unused and therefore their location does not matter in testing. To represent a 1 the toggle should be down towards the pin number and a 0 should be represented with the toggle down away from the printed pin numbers.

|  |  |
| --- | --- |
| Procedure | Verification |
| 1. Include BreathalizerDipHandler.c and BreahalizerDipHandler.h in a new project | N/A |
| 1. Uncomment “#define DIP\_MODE\_TEST” | N/A |
| 1. Program the PIC with these two files. | N/A |
| 1. Set the DIP switch to 000000 and record the value of the top line of the LCD in the box to the right | = “BAC Percent RL ” |
| 1. Set the DIP switch to 100000 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 100001 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 100010 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 100011 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 100100 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 100101 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 110000 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 110001 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 110010 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 111000 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 111001 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 111010 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
| 1. Set the DIP switch to 111111 and record the value of the top line of the LCD in the box to the right | = “Unimplemented” |
|  |  |
|  |  |
|  |  |

## Testing the Dip Switches:

## Expected values of Digital Potentiometer:

|  |  |  |
| --- | --- | --- |
| Resistance | Reference Resistance | Value |
| RL1 | 1 kΩ | 1.2 kΩ |
| RL2 | 5 kΩ | 5.0 kΩ |
| RL3 | 10 kΩ | 9.9 kΩ |
| RL4 | 20 kΩ | 20.1 kΩ |
| RL5 | 50 kΩ | 50.0 kΩ |

# Requirement 3

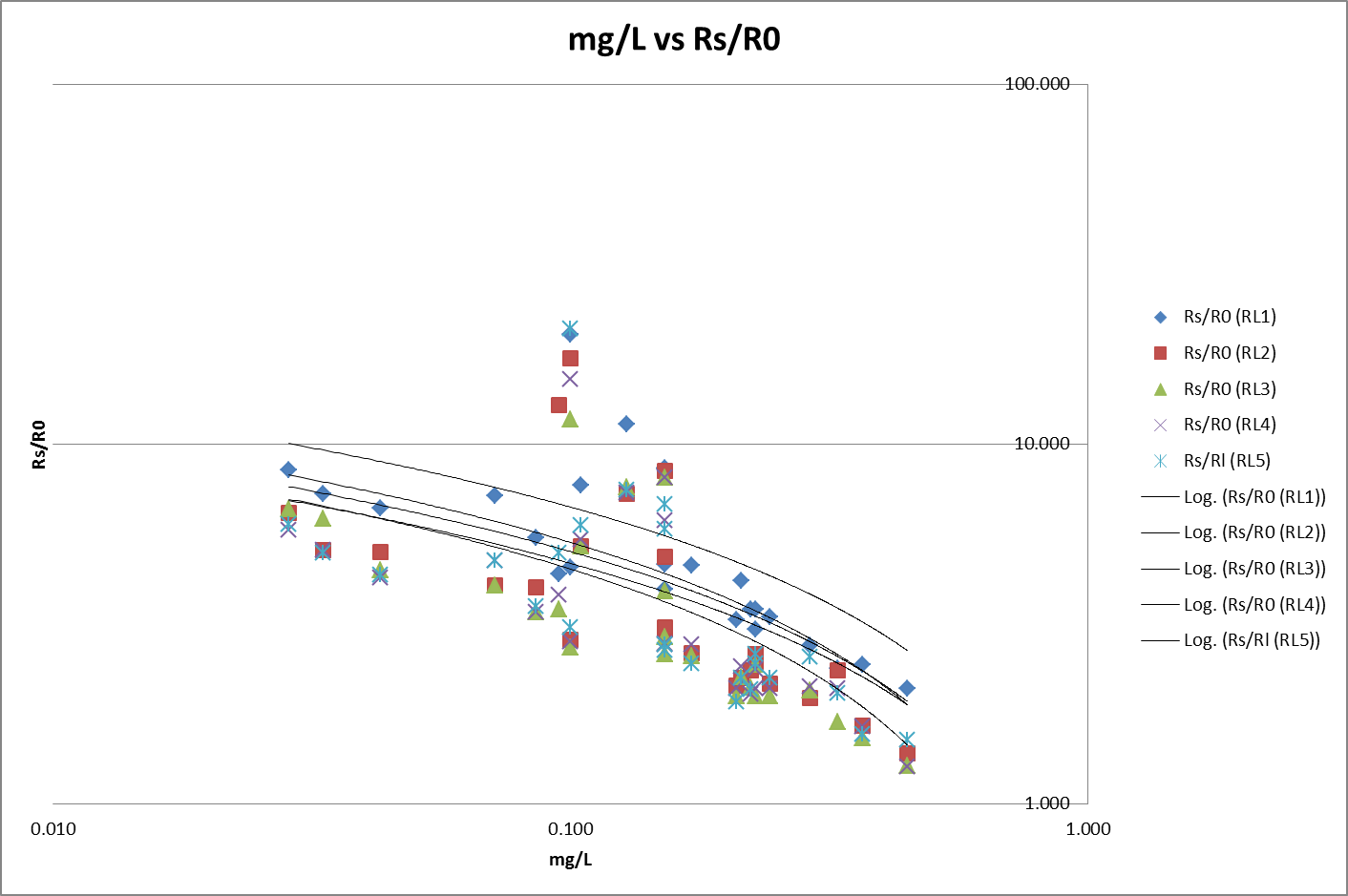
## Table of Observed data:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Subject | Date/Time | BAC | ADC-RL1 | ADC-RL2 | ADC-RL3 | ADC-RL4 | ADC-RL5 |
| Pepe | 4/4 7:30PM | 0.021 | 103 | 260 | 485 | 590 | 720 |
| Alejandro | 4/4 7:30PM | 0.032 | 214 | 561 | 747 | 789 | 916 |
| Alejandro | 4/4 8:30PM | 0.032 | 213 | 422 | 580 | 735 | 900 |
| Pepe | 4/4 8:30PM | 0.022 | 232 | 545 | 685 | 810 | 914 |
| Jose | 4/4 8:30PM | 0.027 | 170 | 459 | 595 | 755 | 890 |
| Alejandro | 4/4 9:00PM | 0.02 | 350 | 323 | 770 | 864 | 930 |
| Pepe | 4/4 8:30PM | 0.015 | 245 | 608 | 740 | 832 | 934 |
| Johnny Boiiiiiiii | 4/4 9:30PM | 0.007 | 243 | 550 | 645 | 821 | 930 |
| JesÜs | 4/4 9:30PM | 0.021 | 340 | 690 | 814 | 900 | 963 |
| Orlando | 4/5 8:00PM | 0.009 | 260 | 554 | 719 | 848 | 941 |
| Jose | 4/5 8:00PM | 0.006 | 215 | 490 | 630 | 800 | 913 |
| Pepe | 4/5 8:00PM | 0.018 | 298 | 611 | 773 | 878 | 955 |
| JesÜs | 4/5 8:00PM | 0.044 | 420 | 752 | 861 | 929 | 985 |
| Alejandro | 4/5 8:00PM | 0.032 | 336 | 674 | 802 | 902 | 971 |
| Orlando | 4/5 8:30PM | 0.032 | 373 | 671 | 821 | 901 | 969 |
| Pepe | 4/5 8:30PM | 0.048 | 435 | 720 | 831 | 918 | 972 |
| JesÜs | 4/5 8:30PM | 0.077 | 493 | 800 | 894 | 948 | 992 |
| Alejandro | 4/5 8:30PM | 0.047 | 403 | 732 | 853 | 932 | 982 |
| Orlando | 4/5 9:00PM | 0.048 | 403 | 710 | 861 | 929 | 975 |
| Pepe | 4/5 9:00PM | 0.051 | 415 | 750 | 861 | 929 | 979 |
| JesÜs | 4/5 9:00PM | 0.094 | 531 | 829 | 913 | 964 | 993 |
| Alejandro | 4/5 9:30PM | 0.045 | 359 | 742 | 843 | 916 | 979 |
| Orlando | 4/5 9:30PM | 0.061 | 462 | 768 | 856 | 928 | 973 |
| Pepe | 4/5 9:30PM | 0.069 | 499 | 732 | 882 | 929 | 983 |
| Jose | 4/5 9:30PM | 0.036 | 337 | 708 | 823 | 902 | 975 |

The table represents the data points collected in our experiments. The ADC values were evaluated in a manner described in the Testing and Calibration tab of this document. The scatter graph shows the data plotted with trend lines and r squared values. The closer the R squared value is to 1, the better fit the trend line is and therefore, the highest R squared value should be the equation we utilize for our own data optimization. In the data we selected the RL1 showed the highest R squared value at R squared of .6778. We observed the data of the higher resistors seemed to top out at around 950 regardless of the BAC levels. Therefore, we have selected the following equation for our RL value:

# Requirement 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ADC R0 | 531.0 | 846.4 | 937.5 | 974.2 | 1004.8 |
|  | 1200 | 5000 | 9900 | 20100 | 50000 |
| mg/L | Rs/R0 (RL1) | Rs/R0 (RL2) | Rs/R0 (RL3) | Rs/R0 (RL4) | Rs/Rl (RL5) |
| 0.100 | 20.185 | 17.337 | 11.714 | 15.143 | 20.940 |
| 0.152 | 8.543 | 4.865 | 3.902 | 6.119 | 5.812 |
| 0.152 | 8.594 | 8.413 | 8.066 | 8.085 | 6.800 |
| 0.105 | 7.705 | 5.181 | 5.211 | 5.426 | 5.934 |
| 0.129 | 11.339 | 7.259 | 7.596 | 7.324 | 7.436 |
| 0.095 | 4.345 | 12.803 | 3.470 | 3.797 | 4.976 |
| 0.071 | 7.176 | 4.032 | 4.038 | 4.737 | 4.741 |
| 0.033 | 7.254 | 5.081 | 6.189 | 5.077 | 4.976 |
| 0.100 | 4.540 | 2.851 | 2.711 | 2.820 | 3.100 |
| 0.043 | 6.632 | 5.001 | 4.465 | 4.258 | 4.336 |
| 0.029 | 8.493 | 6.426 | 6.587 | 5.751 | 5.995 |
| 0.086 | 5.498 | 3.984 | 3.415 | 3.408 | 3.543 |
| 0.210 | 3.244 | 2.129 | 1.987 | 2.088 | 1.920 |
| 0.152 | 4.621 | 3.059 | 2.910 | 2.768 | 2.665 |
| 0.152 | 3.938 | 3.099 | 2.598 | 2.794 | 2.773 |
| 0.229 | 3.055 | 2.486 | 2.440 | 2.360 | 2.611 |
| 0.367 | 2.429 | 1.647 | 1.524 | 1.632 | 1.555 |
| 0.224 | 3.477 | 2.349 | 2.105 | 2.015 | 2.078 |
| 0.229 | 3.477 | 2.604 | 1.987 | 2.088 | 2.450 |
| 0.243 | 3.311 | 2.150 | 1.987 | 2.088 | 2.236 |
| 0.448 | 2.094 | 1.382 | 1.272 | 1.263 | 1.503 |
| 0.214 | 4.180 | 2.237 | 2.255 | 2.410 | 2.236 |
| 0.290 | 2.744 | 1.962 | 2.060 | 2.112 | 2.557 |
| 0.329 | 2.373 | 2.349 | 1.688 | 2.088 | 2.025 |
| 0.171 | 4.600 | 2.628 | 2.566 | 2.768 | 2.450 |

When comparing the graphs from our observed values versus the reported values we noticed some similar qualities. The graphs appear to follow a common trend of regression. We however observed all of our values were considerably higher in the beginning that the data would suggest. This may be due to inaccurate sensors, invalid measuring methods or a number of other environmental variables. We did observe our log graphs seemed to follow a more linear pattern that is also represented in the documentation which would suggest we were recording accurate values that are simply skewed slightly. Again, we observed some outliers that came from potentially different data sources such as different blowing methods, rates, or temperatures that could in the future be analyzed and removed from our data set to see a closer correlation between the mg/L vs Rs/R0.

# Requirement 5

The graphs display a wide variety in the log trend lines. However, when comparing to the graph shown in the gas sensor for the Alcohol trend line we observed a number of qualities. The graph shown in the documentation began at .1 and therefore, we compared our starting point similarly. We observed the starting point in the documentation was close to approximately 2. When comparing this starting point with our data, multiple series displayed a similar starting point, particularly RL3, RL4 and RL5 were closest to this starting value. When compared, RL3 appeared to follow the reported values for the gas sensor. While looking at the data, we observed there were however a number of outliers that could be further changed in the future to find data closer to the trend lines. We have however selected RL3 to represent our data. While the R squared value from Requirement 3 is considerably lower than RL1, the log graphs more closely resemble the desired data set for the alcohol values of the gas sensor.

# Requirement 6

## Table for BAC ADC indexing:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BAC 0  index 0 | BAC 0.01  index 1 | BAC 0.02  index 2 | BAC 0.03  index 3 | BAC 0.04  index 4 | BAC 0.05  index 5 | BAC 0.06  index 6 | BAC 0.07  index 7 | BAC 0.08  index 8 | BAC 0.09  index 9 | BAC 0.10  index 10 |
| 783 | 805 | 828 | 851 | 874 | 897 | 919 | 942 | 965 | 988 | 1011 |

# Requirement 7

To derive a linear equation in the format of aT + b = y where a is a constant, T is the temperature, b is another constant and y is the resultant compensation for temperature we started out by setting up a few sample equations. We assumed this formula for the temperature range of 10° C to 30° C (50° F to 86° F) where the resistance values were fairly linear. We assumed the relationship Rdiv = Rs/R0. We are also making an assumption Rdiv of 1.1 at 10 degree C, 1.0 at 20, and 0.9 at 30. 3. So Vrl decreases by approximately 1 / 1.1 at 10 degrees and increases by about 1 / 0.9 at 30 degrees. To compensate, we must multiply Vrl by 1.1 when the temperature is 10 degrees and by 0.9 when it is at 30 degrees. By setting up the three know equations we have the following relationships:

Thus to solve for “a” we used elimination:

=

Using we can solve for b:

And our resulting linear equation becomes the following.

## Equation for temperature adjustment of ADC Blood Alcohol Value:

This equation will return a constant value when multiplied by the BAC value will adjust BAC upward when temperature is below 20 degrees Celsius and adjust the BAC downward when BAC is above 20 degrees Celsius.

# Requirement 8

In an attempt to select an accurate alpha value we were unsure what to select originally. In doing some research different sources provided different ideas of idea alpha values. The standard rule of thumb indicated alpha should always be less than .5. However, other sources indicated a value between .05 and .3 are better while others indicated a value between .2 and .3 is ideal.

Without performing actual analysis on different alpha values we decided to use an alpha value of .5 for the temperature sensor. This was based on the fact that the temperature sensor was known to be fairly stable in its calculations and therefore less prone to noise on the ADC channel.

When selecting an alpha value for the gas sensor we decided to use an alpha value of .3. This would prevent large spikes from disrupting the data. This smaller alpha value however would also increase the time necessary from taking the initial reading to the point where the data has potentially equalized to an accurate value.

Requirement 9

## Equation for Celsius from ADC values:

Our ADC ranges in value from 0 to 1023, this represents a proportion of the represented voltage from 0 to 5 V. Therefore, by dividing 5V / 1023 we determine each change in ADC range indicates a change of .0048876 Volts. The documentation for the temperature sensor indicates a .01 change in voltage indicates a 1 degree Celsius change. Therefore a change of 1 in the ADC indicates a .48876 degree change in Celsius. Our simple temperature sensor begins at 2 degrees Celsius so to get the actual value of Celsius we need to subtract 2 from the derived equation to get the exact value. Therefore the entire equation is as follows:

## Equation for Fahrenheit from ADC values:

We can then derive an equation for Fahrenheit by knowing the conversion is:

# Requirement 10

In selecting the frequencies for the buzzer to play we decided to use different variations of the “C” note. We selected a High C of about 1048Hz to play for our high note and a Middle C of 261.63Hz for our low note.

As per design requirements, our duty cycle was set to 50%. Therefore to vary the frequency we would need to change either the prescaler or the period of the PWM. We decided to change the period of the PWM to meet this requirement.

The formula for the PWM period is as follows:

Solving for the desired frequency we can use to deduce:

Because we selected a predetermined prescaler of 16 and were running our clock at 4MHz we can deduce the formula to:

And now solving for “High C” (1046.502 Hz):

Because PR2 only stores whole values we set PR2 to 59 to play this note.

Now solving for “Middle C” (261.63Hz):

Because PR2 only stores whole values we set PR2 to 238 to play this note.

# Testing and Calibration:

In an attempt to calibrate the sensor various different data points were tested. In an attempt to standardize the data, all tests were performed at in a room between the temperatures of 65 to 75 degrees Fahrenheit. The recorded BAC values were recorded from a handheld BAC unit “BACtrack”. This allowed for instant results. The BAC value was recorded before ADC values were measured on our circuit.

Furthermore, while blowing into the sensor, a deep breath was taken each time and the breath was held as long as possible. This was done to allow the smoothing procedure a possibility to normalize. Each person also blew into a ring around the test unit to direct a majority of the airflow through the unit and not around the sensor. While this method is unsure if it is accurate, it was standardized in all the tests and was used in all instances of collecting the data.