Electrical Trumpet

-Integration of an Air Pressure Sensor-

By :

Abdul-Wahab Chaarani

Gr.: 4

Spring 2014

"Playing the trumpet is such an intangible experience. The player is compelled to start from the beginning and build anew each day… It isn't like building a house, where we resume where we left the previous accomplishments. Each new day we must rebuild the foundation of our musical structure…"

-Armando Ghitalla-Professional Trumpeter

Table of contents

Abstract

This project was undertaken to improve my original electrical trumpet by integrating the possibility to blow inside a sensor to recreate a real trumpet's range of pitches. The improvements were a success. Nevertheless, the blowing mechanism still needs some development in order to perfect the simulation of blowing inside a real trumpet.

Introduction

Two years ago, I had a multidisciplinary project that combined both my scientific competencies and my harmonic passion for the musical arts. The physics assignment had me conceive an object using the knowledge acquired during the course. In parallel, for my music class, it was requested that I built a simple musical instrument that could play a single *C* major scale (see Appendix F for musical vocabulary). The physics subject at that time was electricity and electrical circuits. I was already experienced with that subject and skilled in building and soldering electrical circuits at that time. Since I used to play the trumpet during my music class, I had decided to build an electrical trumpet. In order words, my instrument used electricity instead of mechanical waves in order to play musical notes. Finally, what differentiates a trumpet from other brass instruments is the fact that there are only three valves that can produce many musical notes depending on the way the player blows into the instrument's embouchure.

At the end, the final product was fairly simple. It had only three buttons that produced a variety of notes using electricity. Nevertheless, I had been unable to integrate the heart and soul of playing the trumpet: blowing air through its tube. I did not have the necessary knowledge in order to do so two years ago. I also did not concentrate on the theoretical side of the project but only on its practical side. Thus, I wasn't able to comprehend the science behind the object in order to make some improvements and further develop its complexity. Consequently, the subject of my research project will be the second attempt in building my original trumpet by incorporating what I call a "blow switch" that will create the same feeling as blowing into a trumpet, making my instrument more realistic. Indeed, it will be rebuilt from scratch.

First, I will include the physics involved in the instrument in the theoretical framework in order to fully understand the object. Next, I will list all of the methods and materials used in the making of the project. Moreover, I will expose all the problems that I have encountered during the building process as well as the adjustments that I have made. Finally, I will compare my final product with its previous version.

Theoretical Framework

Trumpets

Figure 1: A Trumpet

Trumpets are musical instruments. They have the highest pitch in the brass family. They are played by producing "buzzing" sounds that act as constant waves thus producing certain musical notes. The stronger the blow, the higher the amplitude of the wave will be, making the sound much louder. This wave is propelled inside their many valves. The three piston valves are able to modify the frequency of the sound waves by delaying the air, thus changing its period (see *Sound and Speakers* for the link between frequencies and musical notes).

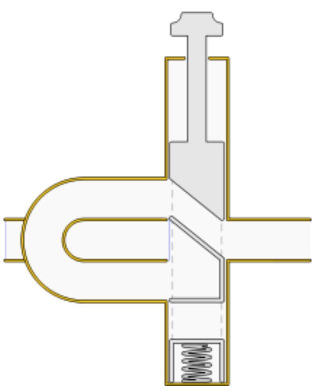


Figure 2: An Open Valve Controlling the Path of the Air

Trumpets are transposing instruments. The most common type is the *B* ♭ trumpet, but *A*, *C*, *D*, *E* ♭, *E*, low *F*, and *G* trumpets are also built. The trumpet that I produced is in *C* because it easily satisfied the initial condition: it must play a simple major *C* scale.

Sound and speakers

The human brain can interpret sounds as the vibration of the eardrum. Rapid changes in air pressure are the most common method to vibrate this very thin piece of skin. Thus, when an object vibrates in a certain environment, such as air or water, we humans hear a sound. The propagation of sound is due to the movement of particles inside the environment. Those particles, in turn, move all the other particles around them, carrying the pulse of the vibration. This pulse is called “sound wave”. Moreover, different sounds are made from different vibrating objects that are caused by different variations of the sound wave’s frequency and amplitude (French, 1966).

First of all, the frequency (f) of a wave is “the number of occurrences of a repeating event per unit of time” (Séguin, 2010). The number of oscillations per second is measured in Hertz (Hz or s^{-1}). The period (T) is the duration (in seconds) of one cycle in a repeating event. Finally, the amplitude (A) is a measure of its change or strength over a single period. The relation between the frequency of a wave and its period is:

$$f = \frac{1}{T} \qquad \textbf{(Eq. 1)}$$

(Séguin, 2010)

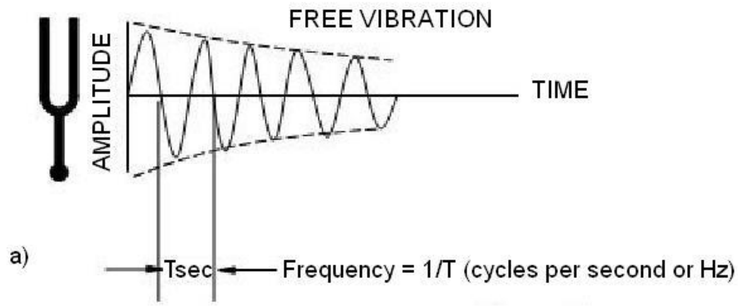


Figure 3: Frequency, Period and Amplitude of a Sound

A higher wave frequency simply means that the air pressure fluctuates faster. We hear this as a higher pitch. When there are fewer fluctuations in a period of time, the pitch is lower. Sound waves with greater amplitudes move our ear drums more, and we register this sensation as a higher volume (Harris, n.d.)

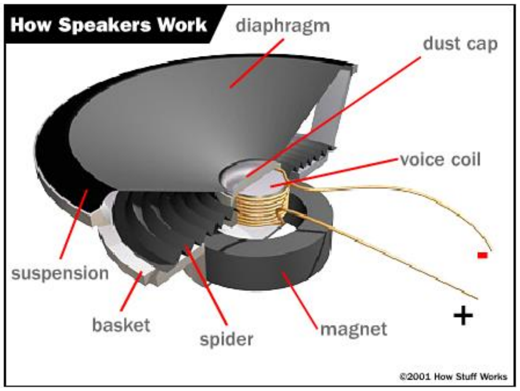


Figure 4: Dissection of a Speaker

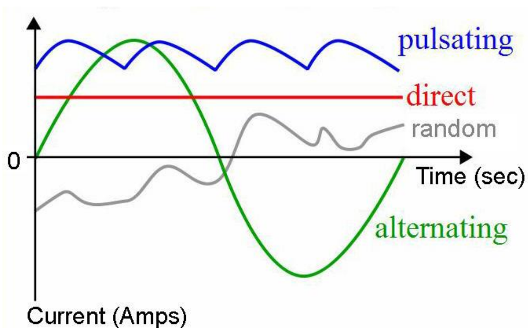


Figure 5: Alternating Current

A speaker is an object that can produce vibrations, with the help of an **alternating electrical current**. The voice coil acts as an electromagnet (solenoid) that is attracted or repulsed by the magnet, as shown in the figure to the left. A solenoid is a conductive wire wrapped around a piece of metal. An electrical current flows inside the wire. This creates a magnetic field around the coiled wire. (Séguin, 2010) The polarity of this artificial magnet depends on the direction of the current. Thus, if it is possible to change the direction of that current significantly fast, it would also be possible to change the polarity of the magnet at that same speed. In other words, the diaphragm will be able to vibrate, producing a certain sound. We can periodically reverse the direction of the current with an alternating current (Harris, n.d.).

Using these ideas, it is possible to produce specific musical notes by correctly timing the alternation of the current. For instance, the middle *A*, or "*la*" in French, has a frequency of 440 Hertz (Michigan Tech, n.d.). Using the first equation, it is possible to calculate its period.

$$f = \frac{1}{T} \rightarrow T = \frac{1}{f} = \frac{1}{440\ Hz} = 2273\ \mu s$$

Therefore, that period represents the amount of time the current has to alternate its polarity in order to hear an *A*. Logically, it must reach its maximum amplitude in half that time and use the other half to reach its minimum. By constantly alternating the current, the frequency that corresponds to the current will produce a sound wave of the same frequency while supplied to a speaker.

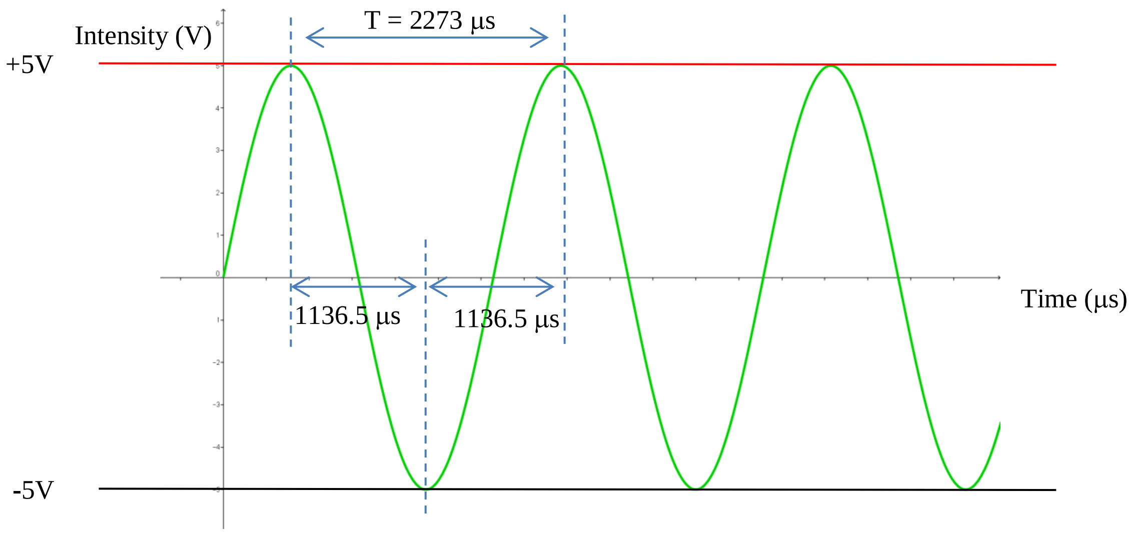


Figure 6: Alternating Current of 440 Hz

As shown in this graph, the current has a sinusoidal periodical function. It holds the characteristics of a sound wave.

Arduino and Microcontrollers

Microcontrollers, or MCUs, are small computers designed for specific applications. They can be programed by programing platforms in order to complete specific actions, such as controlling switches, creating a timer or even an alarm. In order to produce musical notes, I need to create an alternating current with defined periods. I must also map the many combinations using three buttons to recreate the trumpet's fingering. I must therefore write a code and program it into a supporting microcontroller in order to do so. I have chosen to use the *Arduino Uno* programming platform and language because of its user-friendly interface. It also comes with its own MCU. Additionally, *Arduino* does not require much current. It can complete all of its operations with only five volts (5V) (Arduino, n.d.).

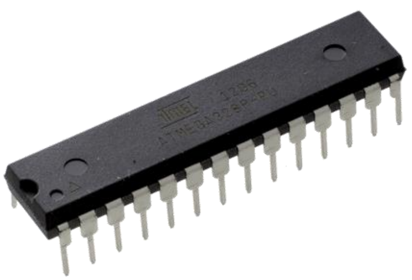


Figure 7: Microcontroller (MCU)

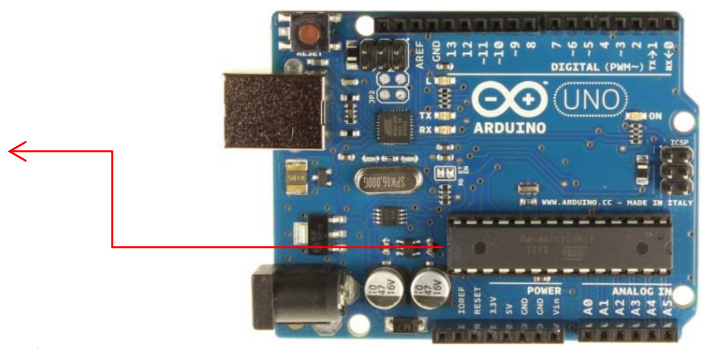


Figure 8: Arduino Uno

Using the previous example, I found that in order to play an *A*, the required alternating current much reach its maximum intensity (+5V) in 1137 μs and its minimum intensity (-5V) in another 1137 μs. Thus, if that current is looped, the connected speaker will vibrate at that same period, producing the note *A*. Using *Arduino*'s programing language, here is an example of the code that can play the note *A*.

```
void loop () {
void A () {
digitalWrite(speaker, HIGH);
delayMicroseconds(1137);
digitalWrite(speaker, LOW);
delayMicroseconds(1137); }
}
```

Key (Arduino, n.d.):

void loop: loops below

digitalWrite(speaker, HIGH) : sets the highest intensity to the port requested (speaker)

void A: name of the loop (A)

digitalWrite(speaker, LOW) : sets the lowest intensity to the port requested

delayMicroseconds(1137): delay in microseconds (waits before next action)

Pressure Sensor

Figure 9: Differential Pressure Sensor

The electrical component that simulates the blowing range of a trumpet is a differential pressure sensor. It detects changes in pressure by comparing the pressure of the blown air to the atmospheric pressure. It translates the intensity of the blow into the voltage of an electrical current.

It is possible to measure the range of intensity of a normal person's blow by using a serial monitor (see *Results and Adjustments*) A trumpet has a range of seven levels of pitches, as shown in the figure below. I have decided to limit my trumpet to five levels to decrease its difficulty. I removed the lowest and the highest playable notes. On the other hand, by doing so, I have decreased the total number of playable notes. Nevertheless, there are still plenty (26 are left.)

As a result, I have been able to assign different ranges of voltage that corresponds to the five middle levels of the trumpet's intensity.

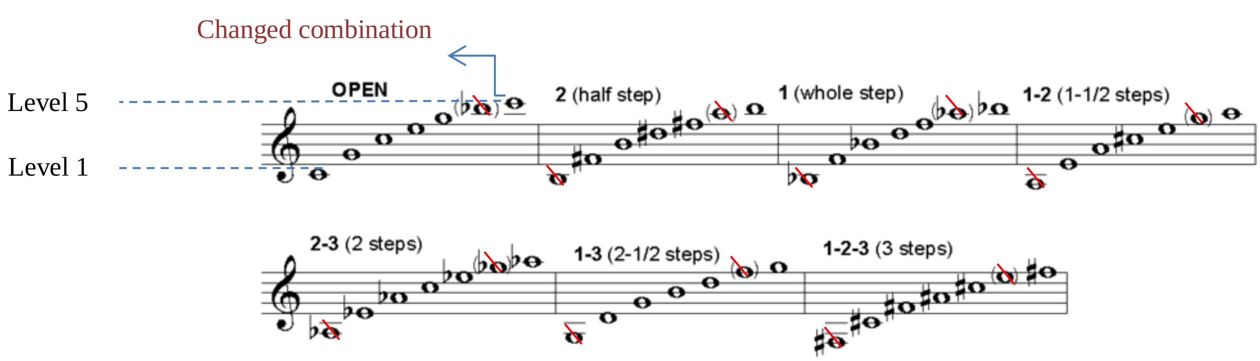


Figure 10: Trumpet's Levels of Pitches

As it is shown in the above figure, there are seven notes that can be played with the same combination of valves on a trumpet by only changing the intensity of the air that flows inside of it(it is much more difficult than it sounds). I have marked in red the notes that I have removed on my trumpet.

Methods and Materials

Material

- Arduino Uno
- Atmega328P-PU Microcontroller
- LM386 Low Voltage Power Amplifier
- 3 Push-button Switches
- 16 MHz Cristal
- 2 22 pF Capacitors
- 1 100 nF Capacitor
- 1 47 µF Capacitor
- 1 10 µF Capacitor
- Soldering Tools
- 1 250 µF Capacitor
- 1 0.05 µF Capacitor
- 1 10 kΩ Variable Resistor
- Speaker
- 8 100 kΩ Resistors
- 8 Light-emitting diodes (LEDs)
- 1 ON/OFF Switch
- 9V Battery
- MPXC201xDT1: 0 to 10kPa, Differential, High Volume Pressure

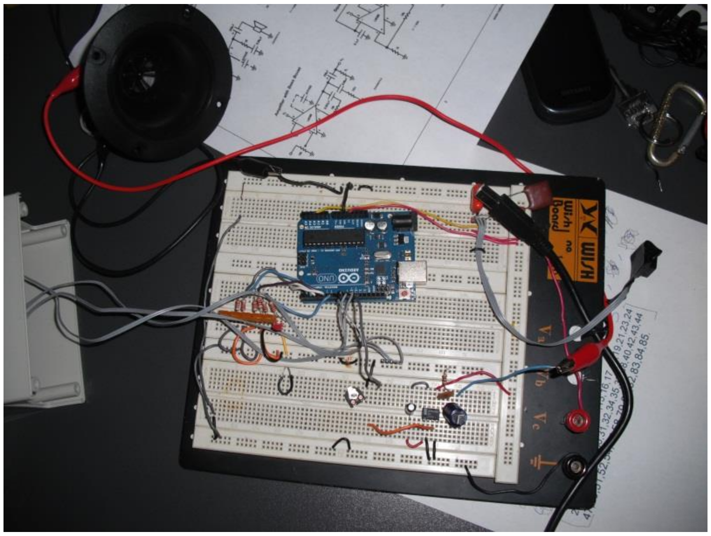


Figure 11: Material Connected on a Breadboard during the Testing Phase

Methods

My personalized circuit only needs the speaker, the sensor, the three buttons, the microcontroller and *Arduino*. Most of the rest are needed to power to amplifier as indicated in its schematic. Indeed, in order to amplify the sound to make it as loud as a trumpet, I have used an amplifying circuit that I found during my research. I copied without any personal modifications. Its schematic in included in Appendix A

The circuit's schematic is contained in Appendix B. It is slightly altered than the one used to build the previous trumpet. It now contains the pressure sensor.

Last time, I installed the whole *Arduino* platform inside the trumpet. Little did I know that it was possible the build a little power supply for the microcontroller so it does not rely on the platform after it has been programmed. The crystal and two capacitors are necessary components, along with the battery, to power the MCU.

Many pictures of the development of the trumpet are contained in Appendix **C**.

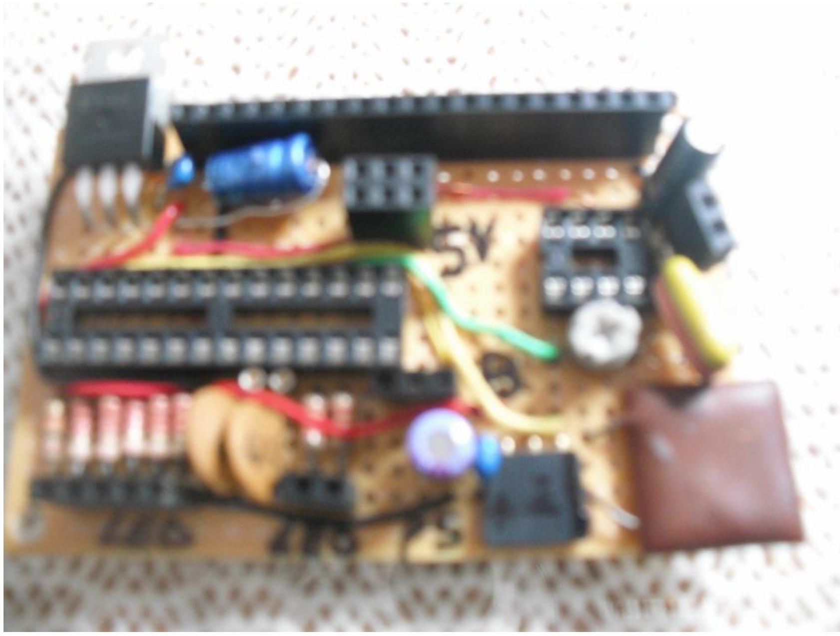


Figure 12: The Internal Circuit

Results and Adjustments

Pressure Monitoring

First of all, the trumpet has its own fingering and uses combination of pressed valves and air intensity to play musical notes. In order to recreate these combinations, I needed to separate the intensity of my own blow into the correct levels.

Arduino's serial monitor had the ability to translate the intensity of my blow into a certain voltage. Since it fluctuates a lot, I then proceeded to limit the intensity into five groups to allow the player to have a bigger range. The stronger the air flow, the higher the level. It took a lot of testing in order to find the correct borders of intensity to create a similar feeling as blowing into the real instrument. Nevertheless, my instrument turned out to be relatively easier to play.

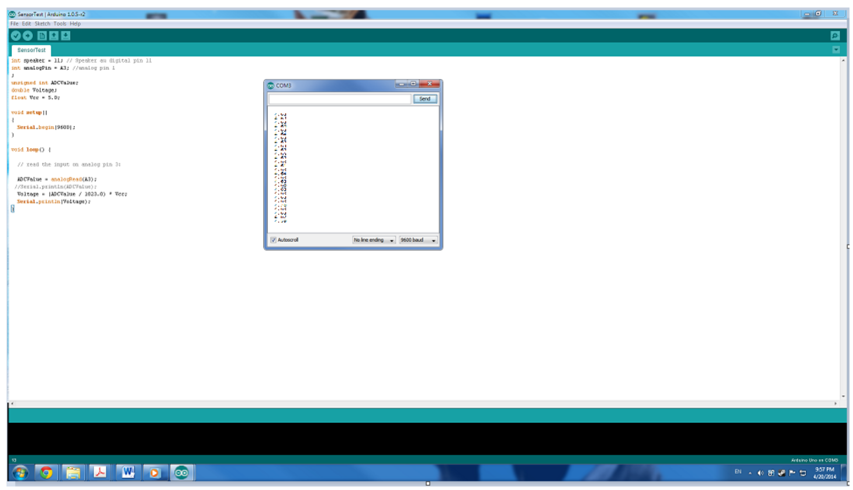


Figure 13: Serial Monitor during the Sensor Test

Nonetheless, even with the sensor, the blowing doesn't feel quite the same as in a real trumpet. The component could not detect vibrations, but only changes in the intensity. On a real trumpet, blowing harder would produce a louder sound, but still play the same note. To change notes, in other words to change levels, we need to vary the pressure of the "buzzing" using our diaphragm. On my trumpet, I could only control the intensity of the air flow. Thus, my object is not a perfect electrical replica of the original instrument. However, it is still a decent simulation. The fingering on my object is exactly the same as on a real trumpet. It is therefore a nice replacement for those that do not have the money to buy the real thing.

Visual Indicators

It was also difficult to reach a particular level when blowing a first note, especially the higher ones. To counter this problem, I installed visual indicators. The five red LEDs represent the five levels (in order form right to left.) There are also two yellow LEDs that signal when we are too close to another level, in order to adjust the air flow. I also had to adjust the code accordingly.

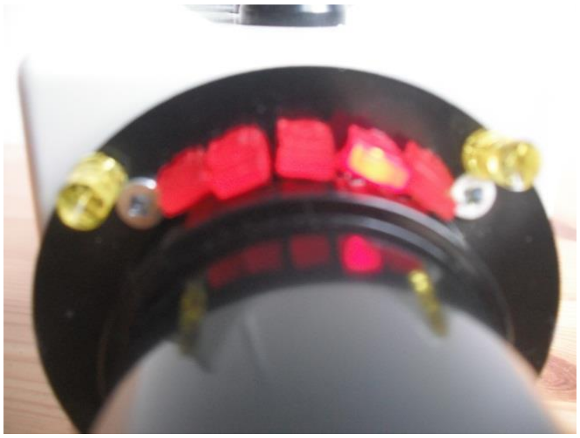


Figure 14: Light Indicators (Levels)

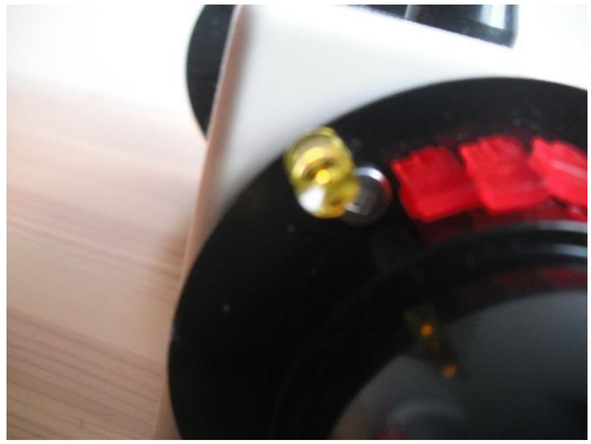


Figure 15: Light Indicators (Critical flow)

In figure 14, I am playing a note from the second level with no particular problem. However, on in figure 15, the yellow light on the left is turned on. The intensity of the flow is a bit too high, which means that the trumpet could possibly end up playing a note in the third level. Therefore, I can adjust my breath until the yellow indicator is turned off.

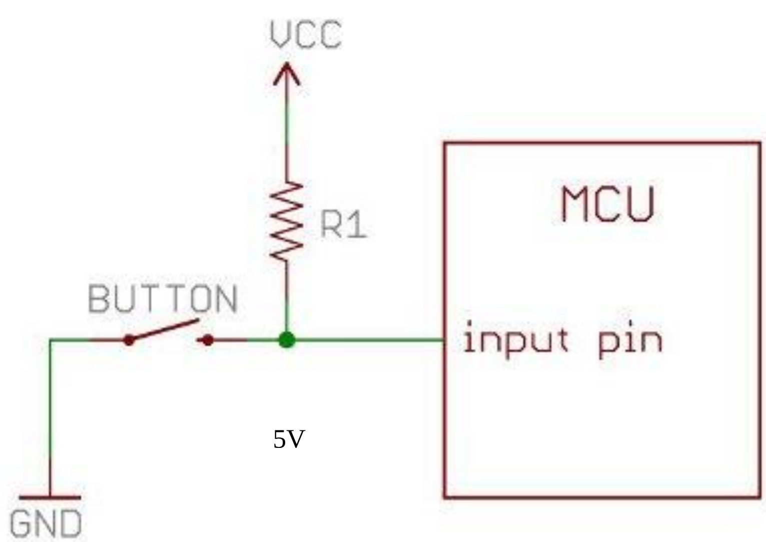
The tail is just for esthetics. It makes the object look more like a real trumpet. It does not hold any specific function.

Pull-up Resistors

At one point during the assembly, the trumpet sometimes played by itself. In other words, when I picked it up, it played some notes. When it stopped and I tried playing a note, the wrong note was played. It was as there was a certain button that was always being pressed. I searched for short-circuits, but I did not find any. I tested the buttons, they all worked fine. After countless hours of trying to figure out what was wrong, I researched the symptoms on the Internet.

It turned out that the microcontroller could not recognize the state of the buttons. When a button is not pressed, is that pin high (pulled to +5V) or low (pulled to 0V)? In plain English, what it electrical potential difference between the MCU and the switches? It is difficult to tell. That unknown state is called *floating* (Sparkfun, n.d.).

To solve that problem, it is best to install *pull-up resistors*. The microcontroller has a tension of 5V. Without a resistor, any electrically charged object with an insignificant voltage, such as dust, could create a minimal difference of electrical potential that would close the circuit. The MCU will read it as a pressed button. With a pull-up resistor, when the button is not pressed, the microcontroller will read a high state (5V), no matter the disturbance. Therefore, the circuit cannot work because there is no electrical potential difference of 5V. Only while connected to the ground (0V-GND), hence while the button is pressed, will the MCU read a voltage of 5V and complete the correct function: to play the note corresponding to the right combination of pressed switches. Only an object with a voltage higher than 5V will be able to bypass the resistor. Fortunately, dust and other ambient object do not possess such electrical tension (Sparkfun, n.d.).



Pull-up resistor

Microcontroller

Filter Capacitors

0V

During the assembly of the amplifying circuit, something unexpected happened. The trumpet started playing music; real songs came out of its speaker. As I sat there looking at it flabbergasted, I realized:

“I built a radio!”

On the spot, I could not explain what happened. I had to stay shocked until a couple days ago actually when, during my physics class, which is also a course on electricity and magnetism, my teacher taught about alternating currents and their effect on capacitors.

As it turned out, without a demonstration, a capacitor’s reactance (X_C) is inversely proportional to its capacity multiplied by the angular frequency of the current (ω):

$$X_C = \frac{1}{\omega C} \qquad \text{(Eq. 2)}$$

(Séguin, 2010)

In electrical systems, a component’s reactance is its opposition to the electrical current. Therefore, in a system fed by an alternating current, a capacitor will offer little resistance to signals that have a higher frequency and much resistance to signals of low frequencies, as the equation suggests.

My musical frequencies range from 260 Hz to 1100 Hz. Radio waves have frequencies that range from 3 kHz to 300 GHz, which are pretty high compared to my notes (French, 1966) As it turned out, I installed the biggest capacitor at a wrong place. Thus, it blocked my own frequencies and acted as an antenna for a radio wave of an exact frequency. Unfortunately, I couldn't change the radio's channels.

Comparison with Previous Trumpet

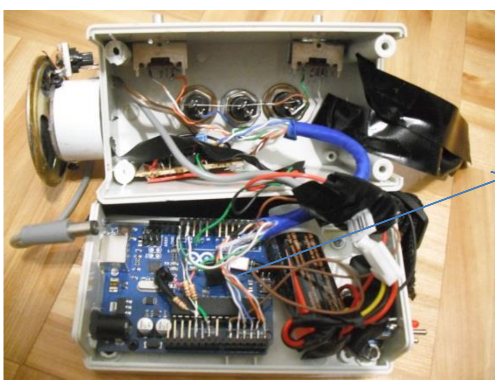
Before

It had a limited amount of combinations because there was no blow function. Switches were used instead

Sharp/Flat and Octave Up/Down switches

"Activate" switch

Replaces the need to blow

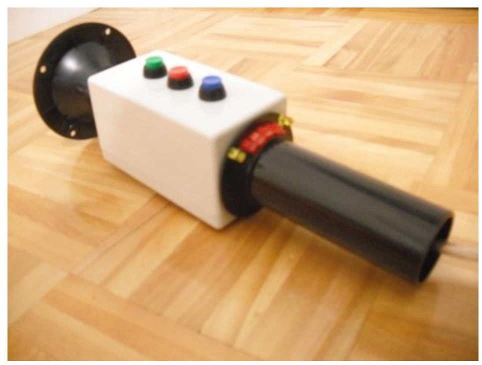


Whole Arduino, completely inserted

The first internal circuit

New speaker

After

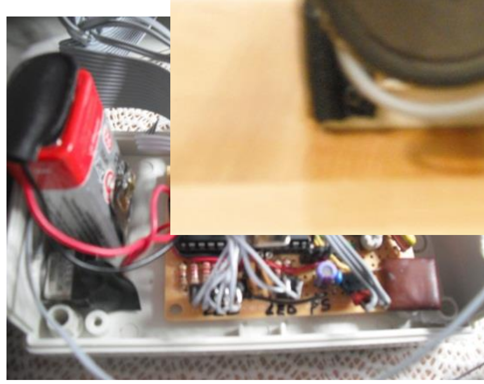


There is now no need for these extra switches. T

into the ser

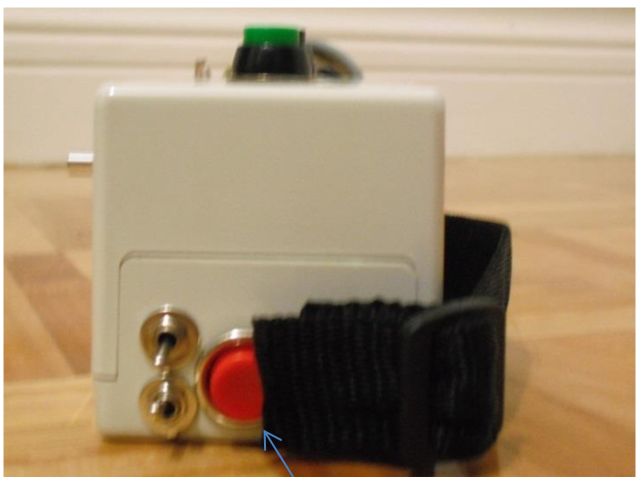
giving acc

combinatio



The new internal circuit

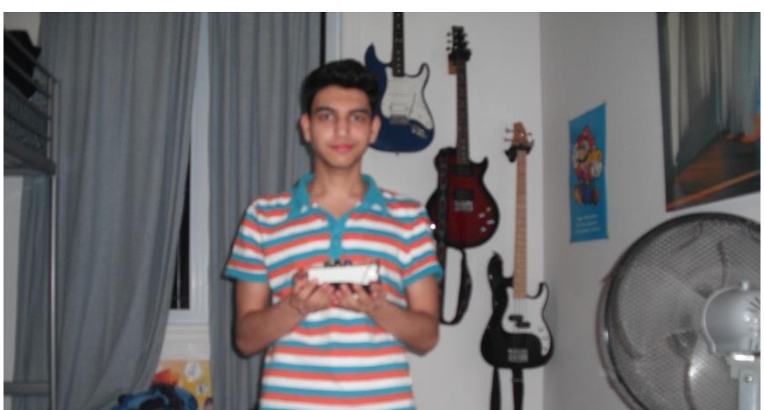
Very similar, it doesn't rely on the Arduino platform anymore.



No tail

- Blowing not possible – switch instead

Had a handle



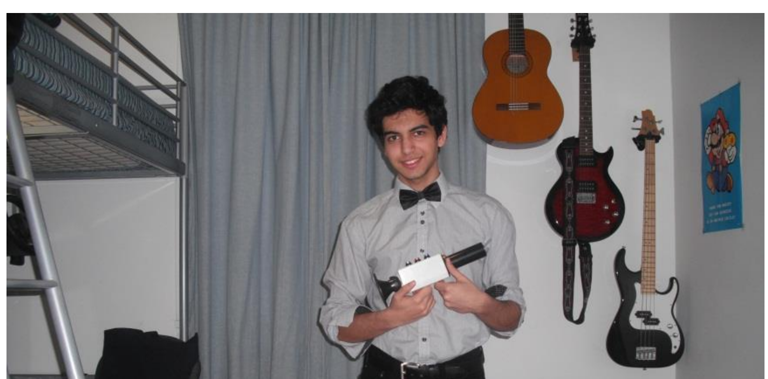
Old trumpet

Young Abdul

New tail-

Blowing now possible – LEDs indicate levels

Handle not necessary anymore



New and improved trumpet

New and improved Abdul

Conclusion

In conclusion, I have been able to achieve the ultimate goal of this project: the integration of a blowing mechanism to recreate an electrical replica of an original trumpet. Even though I have already built a similar object in the past, I have still encountered many problems such as having difficulties calibrating the required air intensity to blow inside the sensor to reach a certain level.

There are many adjustments that could be made in the future to perfect my trumpet. It would be interesting to install a volume potentiometer that could control the intensity of the volume. Furthermore, the improvement of the blow function itself might need some more work since it is difficult to play real melodies with my object. It is also challenging to adapt to this trumpet after playing the real instrument. Finally, it would be wonderful to change the sound of my trumpet. Its current electrical buzzing can become quite annoying after some time.

References

Arduino (n.d.). *Language Refrence*. Retrieved from: http://arduino.cc/en/Reference/HomePage (Retrieved on April- May 2012)

French, A. P. (1966). *Vibrations and Waves*. New-York: W.W Norton & Company

Harris, T. (n.d.). *How Speakers Work*. How Stuff Works. Retrieved from: http://electronics.howstuffworks.com/speaker12.htm (Retrieved on May 2, 2014)

Michigan Tech. (n.d.). *Physics of Music*. Retrieved from http://www.phy.mtu.edu/~suits/notefreqs.html (Retrieved on May 4, 2014)

Séguin, M. (2010). *Physique XXI Tome B : Électricité et magnétisme*. Québec : ERPI

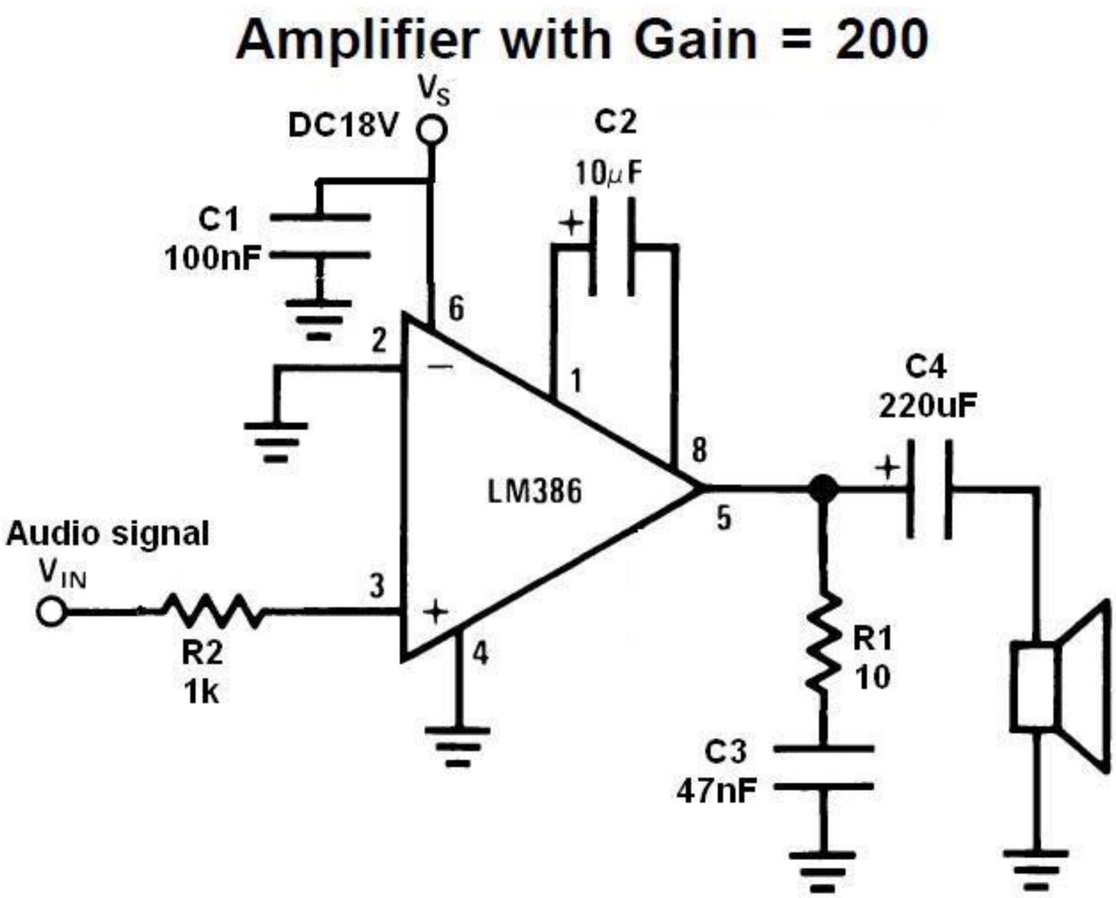
Séguin, M. (2010). *Physique XXI Tome C : Ondes et physique moderne*. Québec : ERPI

Sparkfun. (n.d.) *Pull-up Resistors*. Retrieved from: https://learn.sparkfun.com/tutorials/pull-up-resistors (Retrieved on April-May 2012)

Texas Instruments. (2000). *LM389 Low Voltage Audio Power Amplifier*. Retrieved from http://www.ti.com/lit/ds/symlink/lm386.pdf (Retrieved on April-May 2012)

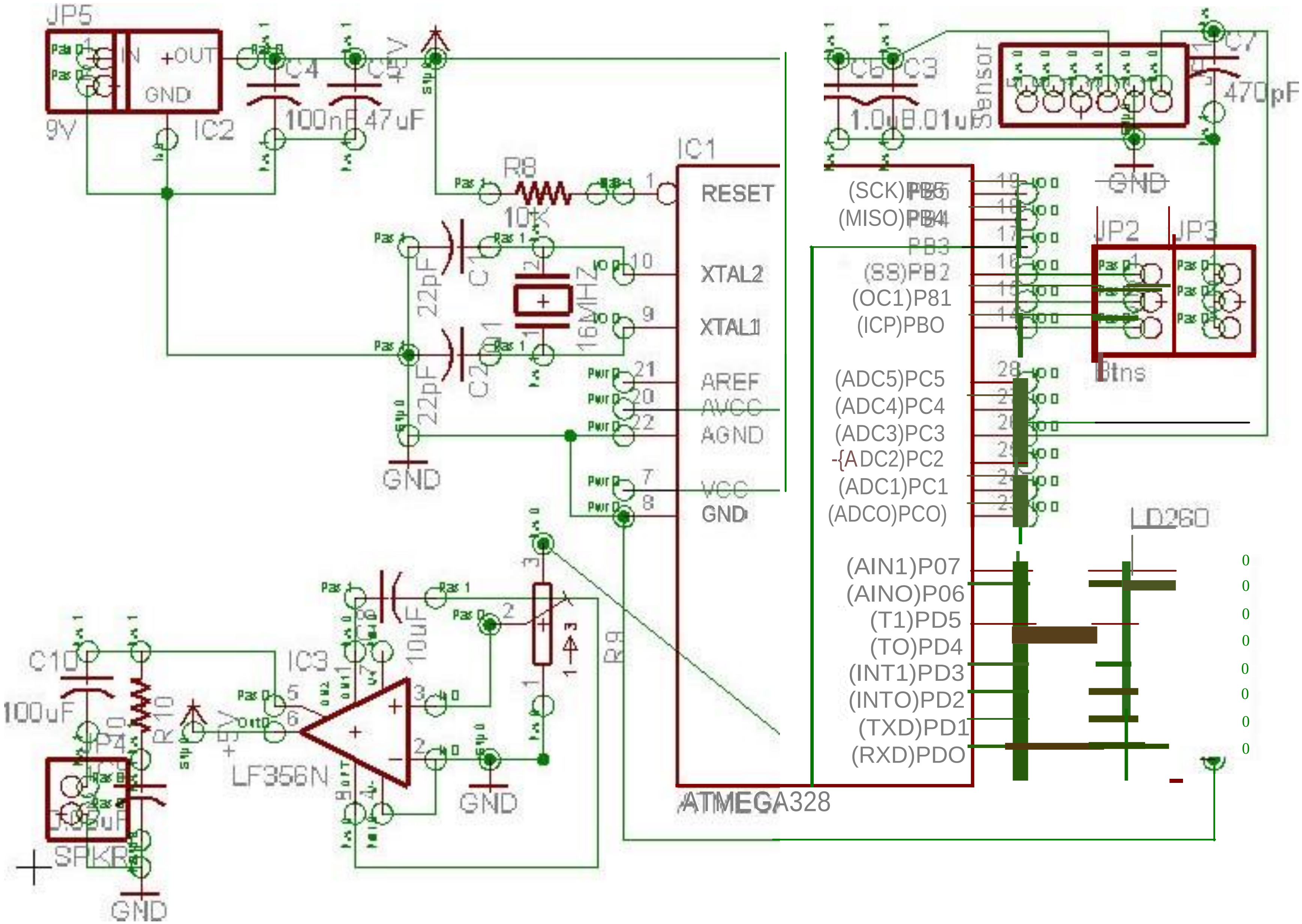
Appendices

APPENDIX A – Amplifier Schematic

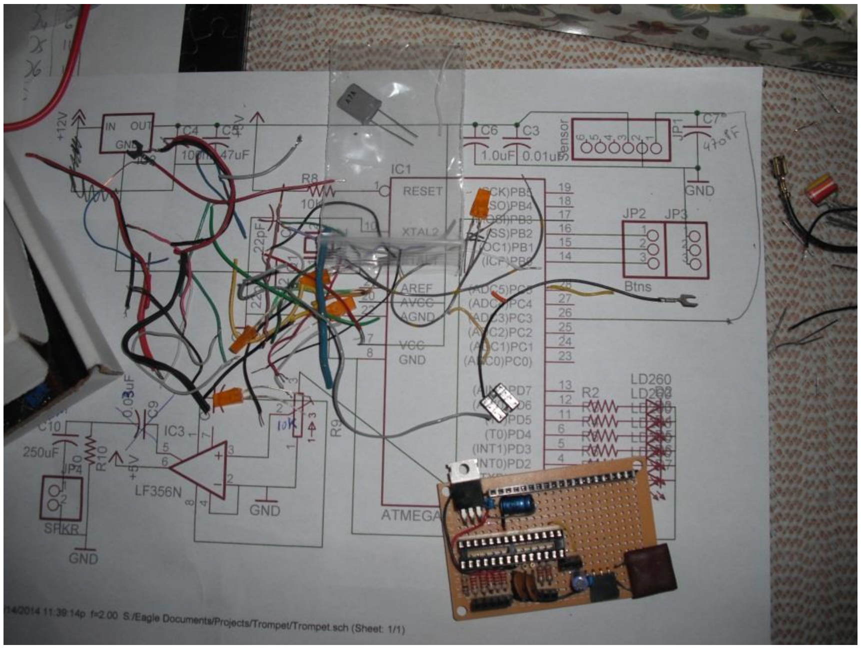


(Texas Instruments, 2000)

APPENDIX B – Complete Circuit Schematic



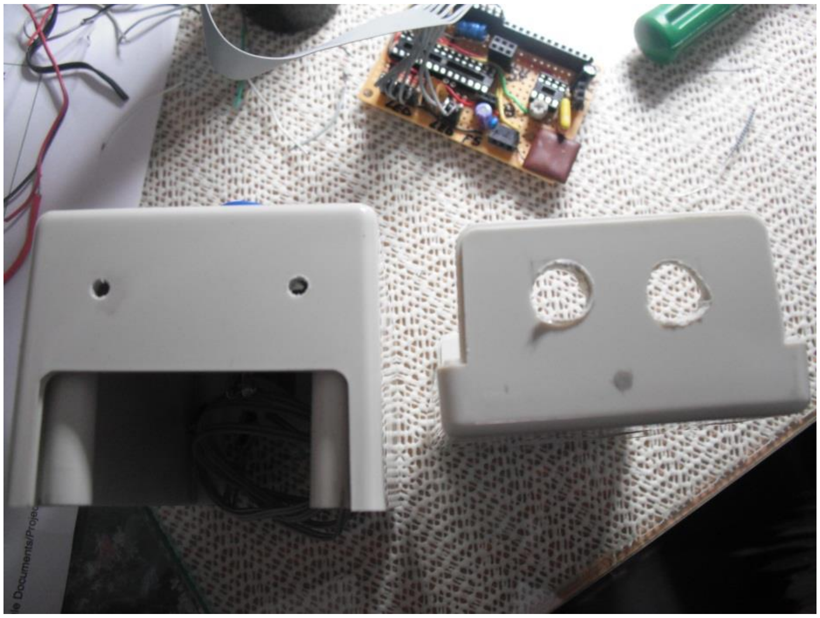
APPENDIX C – Photos of Development



The beginning of the assembly of the internal circuit using my schematic

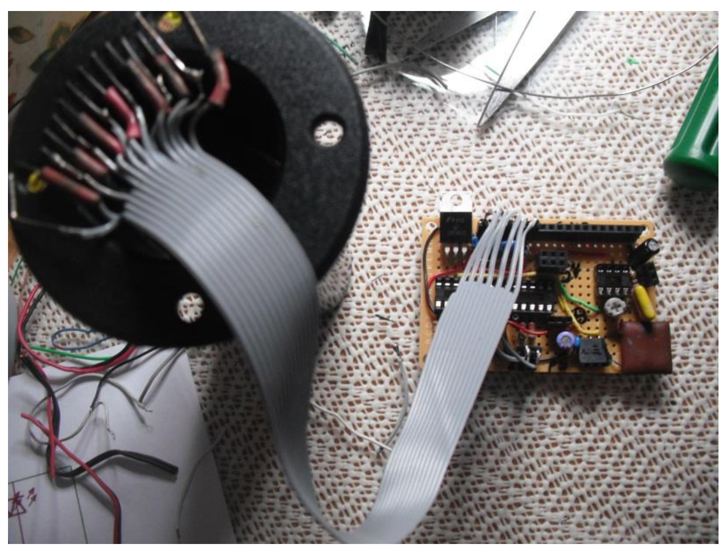
Me trying to take a “selfie” while soldering

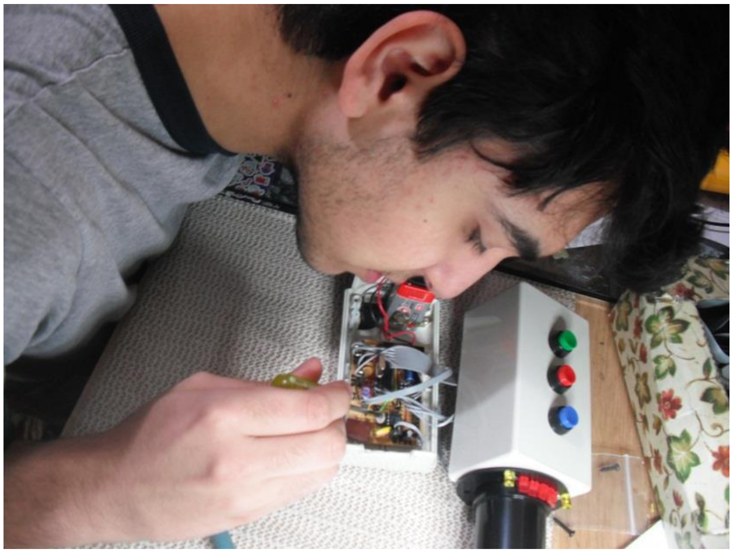
Do not attempt



The drilling of the holes

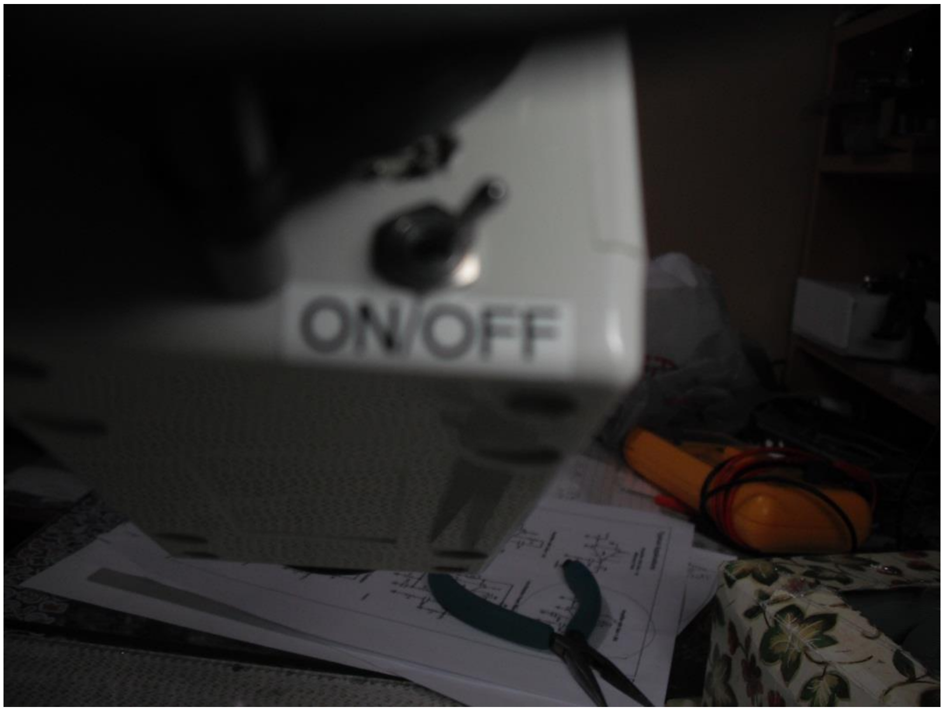
Installation of the light indicators and the tail

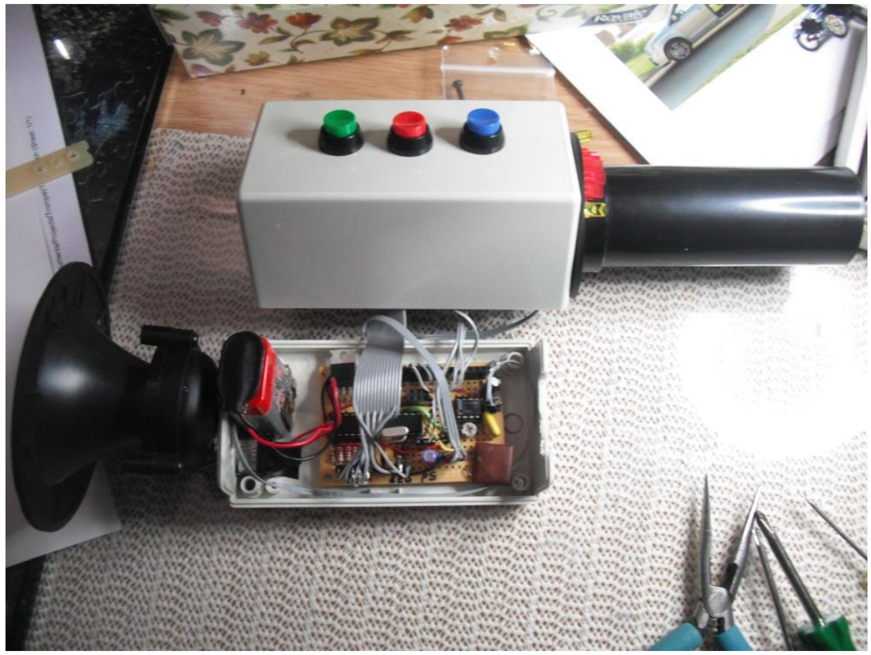




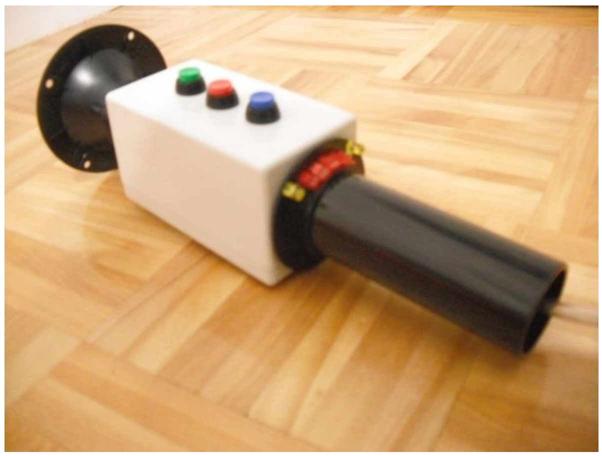
Boxing everything

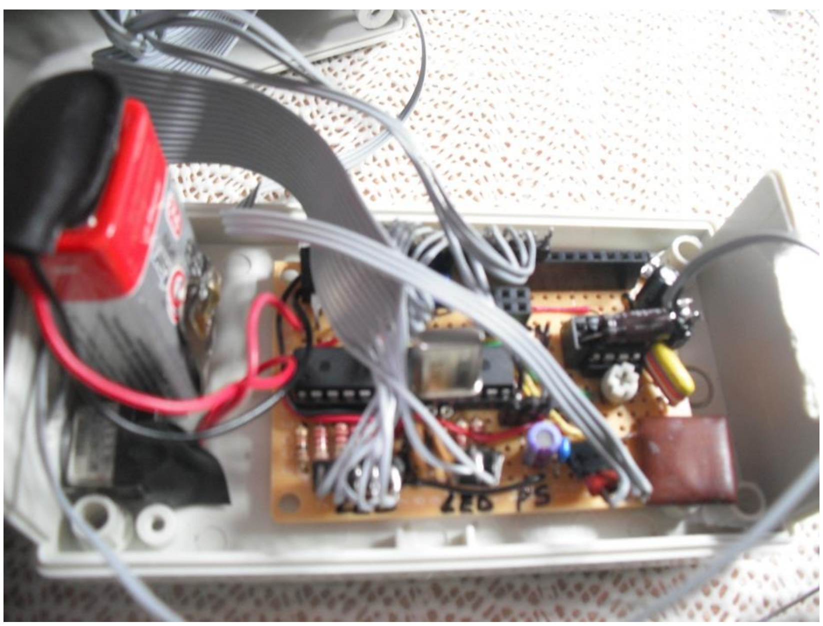
ON/OFF switch





The final product





APPENDIX D – Trumpet Fingerings Chart

(Colored) = Button Pressed

(Empty) = Button Open

“Middle *A*” is A_4

Level 1:

= C_4

= $C^{\#}_4$

= D_4

= $D^{\#}_4$

= E_4

= F_4

= $F^{\#}_4$

Level 2:

= G_4

= $G^{\#}_4$

= A_4

= $A^{\#}_4$

= B_4

Level 3:

= C_5

= $C^{\#}_5$

= D_5

= $D^{\#}_5$

Level 4:

= E_5

= F_5

= $F^{\#}_5$

Level 5:

= G_5

= $G^{\#}_5$

= A_5

= $A^{\#}_5$

= B_5

= C_6

= $C\#_6$

APPENDIX E – Frequencies of Musical Notes

Note	Frequency (Hz)
C_4	261.63
$C^{\#}_4$	277.18
D_4	293.66
$D^{\#}_4$	311.13
E_4	329.63
F_4	349.23
$F^{\#}_4$	369.99
G_4	392.00
$G^{\#}_4$	415.30
A_4	440.00
$A^{\#}_4$	466.16
B_4	493.88
C_5	523.25
$C^{\#}_5$	554.37
D_5	587.33
$D^{\#}_5$	622.25
E_5	659.25
F_5	698.46
$F^{\#}_5$	739.99
G_5	783.99
$G^{\#}_5$	830.61
A_5	880.00
$A^{\#}_5$	932.33
B_5	987.77
C_6	1046.50
$C^{\#}_6$	1108.73

(Michigan Tech. n.d.)

APPENDIX F – Musical Vocabulary

Letter Notation	Note
C	Do
D	Ré
E	Mi
F	Fa
G	Sol
A	La

Scale: Set of musical notes in ascending order. C Major Scale = C–D–E–F–G–A–B–[C]

♭: Flat = (half tone down)

#: Sharp = (half tone up)

Half tone: note between two fundamental notes