Zach D, Cole F, Lucas V

Dr. Dickerson

ECE 1895

13 November 2022

Design Project #2: BOP-IT

Design Overview:

Our project uses a 9V battery to supply power to our circuit. A slide switch is then used to power the device on and off. To begin a new game the user presses the RED button on the device. A buzzer will then beep one to three times and the LCD screen will specify the action that needs to be performed by the user. The three actions are move a joystick, match the YELLOW, GREEN, or BLUE button with the corresponding lit LED, and cover the photo eye sensor. If the user is successful at their attempt the LCD screen will display correct and add a point to their score. If the user is unsuccessful at their attempt the LCD screen will display incorrect and the game will end showing their final score. If the user is successful in their attempt to reach a score of 99 points the game will end, the buzzer will buzz several times, and their final score will be shown on the LCD screen. The time the user will have to complete an action will decrease as their point value on the LCD screen increases.

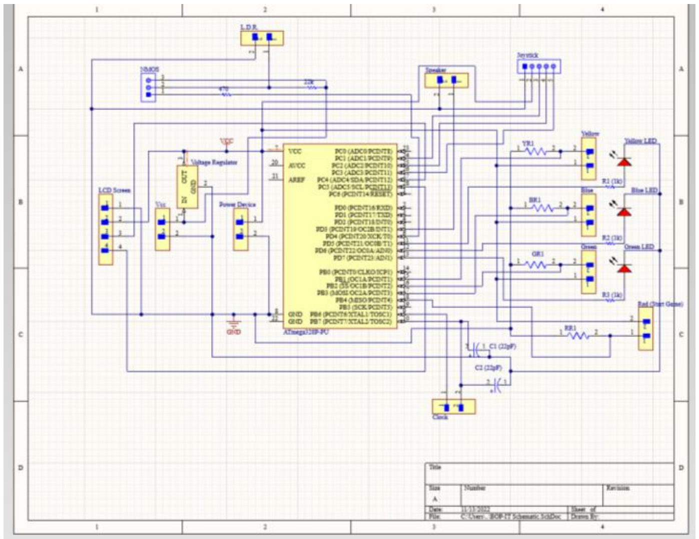


Figure 1: Circuit Schematic

Design Verification:

Hardware:

The circuit was verified in increments using an LED for verification of output. We began by testing the photo eye circuit, which can be found in figure 2 and 3. During initial testing we found that the circuit was not outputting a voltage that was high enough to be detected by the ATMega chip. We then decreased the value of the resistors in the circuit which allowed us to achieve a higher output voltage that was then able to be detected by the chip. We then tested the button matching section where we discovered that the chip was reading in an input value when the buttons were not being pressed, after some further testing we concluded that we needed pull down resistors to prevent this from happening. After implementing the resistors, the buttons functioned as we intended them to. Another input we tested was our joystick where we learned that it gives a default voltage value of 2.5V and increases/decreases depending on the movement of the joystick, this meant that we needed to use the analog pins of the chip to read a voltage change instead of just detecting a voltage value, as well as slight changes to our code. The final piece we tested was our LCD display. During testing we discovered that we had incorrectly mapped our pins to the chip and used digital instead of analog pins which meant we needed to change our PCB design.

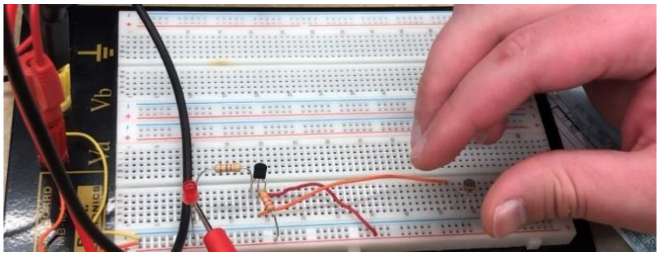


Figure 2: Photoeye Circuit LED OFF

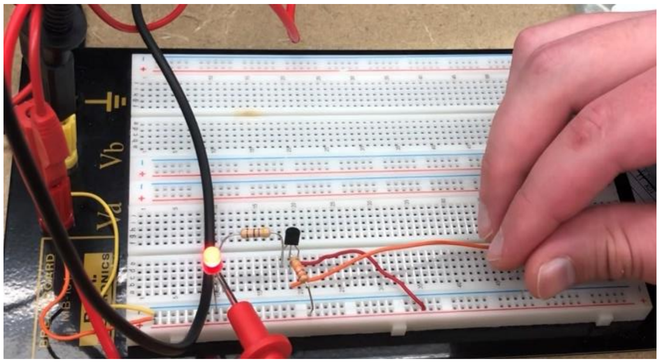


Figure 3: Photoeye Circuit LED ON

Software:

During our hardware testing we discovered many changes that need to be made in order to have our software function correctly. Starting with the simplest changes we switched the joystick to analog pins which allowed us to read in voltage change, we saw that the analog values sat around 600 when not touched and ranged from 0-1300 when moved. This allowed us to set boundaries for a safe area which would not trigger the joystick action when it was not being moved and allow us to detect movement outside of those values. We also needed to test our LCD display to get it to display correctly. Which led us to creating a function that took in two strings, one for each line of the display. This also allowed us to trouble shoot when we were testing our hardware. We also had to test the timing of the game, which was implemented using a while loop that counted down and had a delay of one second for each run.

Electronic Design Implementation:

Our initial PCB design had four buttons, one for the start/new game, and the other three for one of the in-game actions. There was an LED for each of the three game action buttons, a joystick, capacitors, the ATMega Chip, and 2-pin headers for the voltage and ground inputs, a switch, the light dependent resistor, and the clock. There is a 3-pin header for the voltage regulator and a 4-pin header for the LCD screen. The initial design is shown in the picture below.

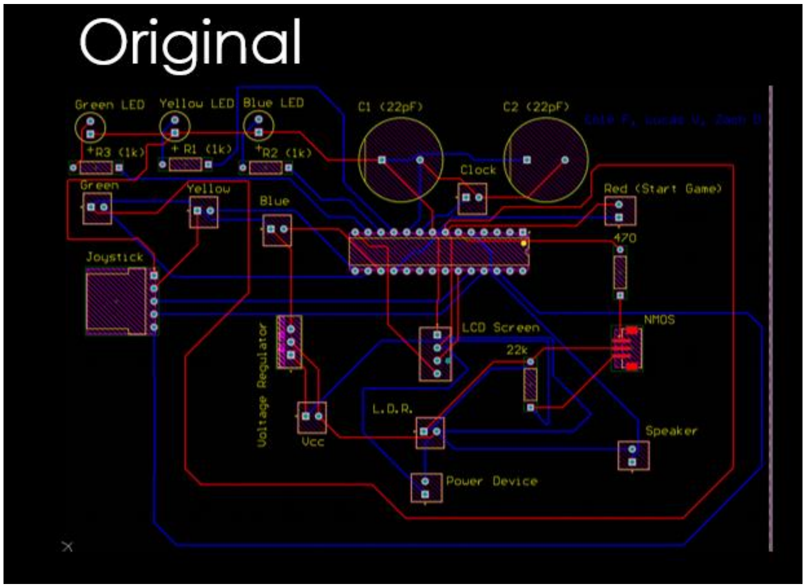


Figure 4: Original PCB Design

The previously mentioned errors with this initial design (missing pulldown resistors, incorrectly pinned LCD screen), in addition to an error where the regulator was between the switch and power which led to the regulator burning when the switch to the rest of the circuit was open, were fixed in the following PCB design. The second and final design had all the same components as the previous design, in addition to pull down resistors to each of the buttons, an updated and correct pinout for the LCD screen, and we put the voltage regulator after the switch to prevent the regulator from frying. The final PCB design is shown in the picture below.

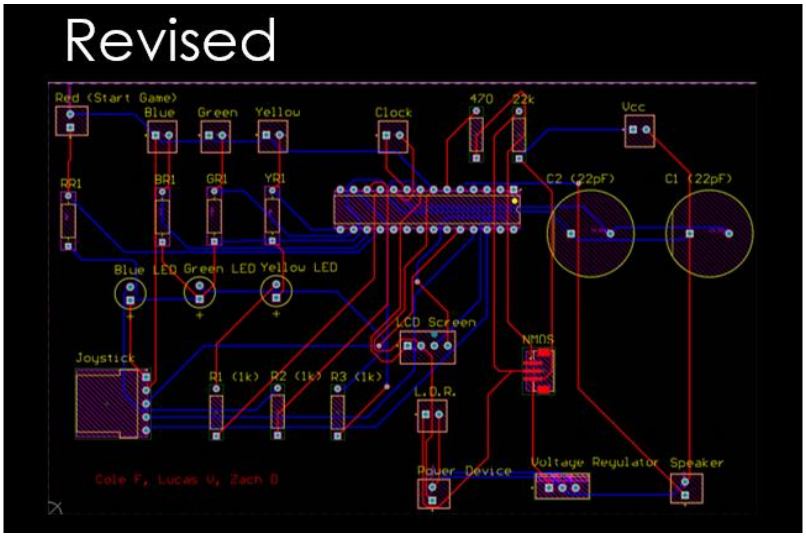


Figure 5: Revised PCB Design

For most of our PCB components, we used 2-pin headers instead of searching for physical components on Altium because we knew we would be attaching wires between the PCB and the component to allow the components to be moved as needed to fit the enclosure. Our PCB layout was designed to put components near their positions on the enclosure.

Software Implementation:

Our software was kept simple, we created functions for reading in the input of each user input, button press, photoeye, joystick. We also created a function for printing out lines to the LCD display. Our main code consisted of two main loops, one to keep the overall game going while the power switch was on and one to keep the round running while the user continued to give correct inputs. The overall game loop contained a while loop that would hold the game until the start game button was pressed. Once pressed a random number would be generated that would pick one of the inputs that the user needed to input, if it was input correctly, it would increase user score and repeat, if it was incorrect, it would display the overall score and end the game and begin waiting for start game again. If the user reached the end game of 99 points the score would be displayed as well as a win game message before it would return to the waiting for new game state.

Enclosure:

Our original design for our enclosure was going to be a Leap Frog gaming console as shown in figure 6. The reasoning behind this design was because it had all the holes necessary for our components, we were just going to need to modify the design to correctly fit our components. The 3 green buttons on the top left were going to be used for our YELLOW, GREEN, and BLUE buttons. The red button on the top right was going to be used for our RED start game button. The speaker at the bottom right was going to be used for our buzzer. The Screen area was going to be used for our LCD screen and the LEDs used for matching the correct buttons. The area at the bottom of the device where the stylus string was attached was going to be where our ON/OFF slide switch was going to be. The area where the stylus was kept on the device was going to be where our photo eye was going to be. The big A and B button was going to be where our GREEN LED and RED LED were going to be for indication of correct/incorrect. The d-pad was going to be the area used for our joystick. After receiving our PCB though we realized that it was too big to be placed in a Leap frog. We then decided to use Fusion 360 software to implement our enclosure design shown in figure 7. The same idea was going to be used but this time we were going to be able to place our components in a more presentable way, along with having more exact holes made for the components. This would allow us to not need to do many modifications, if any to the size of the holes. We later found that after 3 attempts to print out 3D design it wasn't going to work, so we decided to laser cut our design. The same sketches from our Fusion 360 file were used to create the laser cut of our design. The laser cut design was then hot glued together and the components were glued onto the design. After our 3rd design attempt for our enclosure, we were finally able to produce the enclosure shown in figure 11.

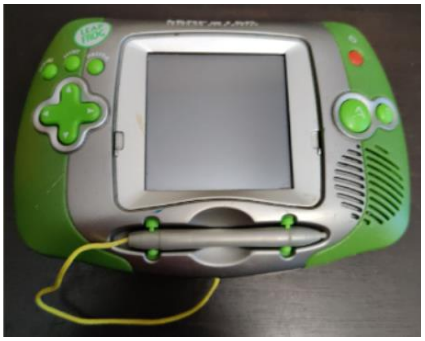


Figure 6: Leap Frog

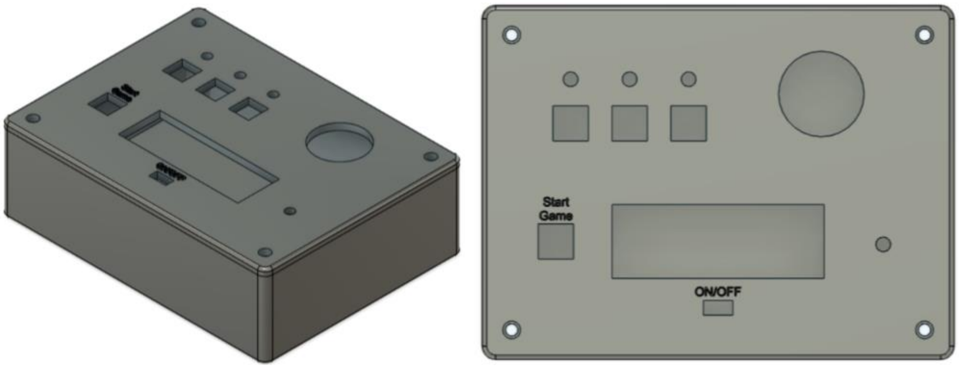


Figure 7: 3D Print

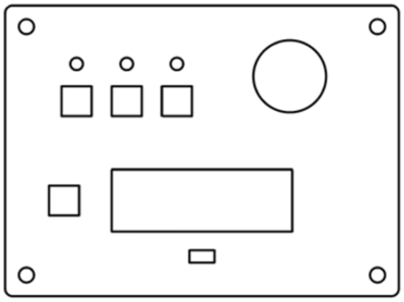


Figure 8: Laser Cut

Assembly and System Integration:

Our assembly required many modifications as we had to order a new PCB after we came across the issues during testing. While we waited for the new PCB to arrive, we assembled the original PCB and attempted to modify it to function as a backup if the new PCB did not arrive in time. To get the original PCB to function we had to first implement pull down resistors to all of the buttons, to achieve this we wired from the button to the resistor to a wire that connected all pull downs to a ground. This worked but added many extra wires which would often get accidentally shorted. We also needed to change the pins from the LCD display to the chip. This was implemented by soldering wires to the same pin holes as the chip on the board which then were wired to the display. These changes allowed our PCB to function with some minor issues caused by the extra wires. Luckily our new PCB had arrived the day before the deadline. We decided to assemble it to make sure we had a fully functional PCB. We made one slight change to the PCB by changing the position of the on/off switch as the original location was causing the voltage regulator to overheat. After fixing that issue the PCB was fully functional with no issues, as well as being much more efficient.

Design Testing:

Our testing and debugging process started with breadboarding pieces of the design. First, we built the light dependent resistor circuit to test the output of this circuit. We found that with the 5 volts from the voltage regulator, you had to cover the entirety of the LDR to get a voltage at the output. In order to fix this, we first changed the LDR to receive the 9 volts input from before the voltage regulator. Additionally, we changed the resistor values in the circuit, making the 470 Ohm resistor a wire. This allowed the LDR to be more reactive and function better as one of our required actions. Next, we tested the buttons. We realized that the buttons had low voltage across them when they weren't being pressed, which was causing the buttons to input randomly. We fixed this issue by adding pull down resistors to the output side of the buttons so that when the buttons weren't being pressed, there was no voltage on the output side. Then we built the entire design on the breadboard to ensure that all the components worked as expected. This is where we incorporated the joystick. We realized that the joystick was constantly inputting. This is because in the neutral position, the joystick outputs at 2.5V, then increases to 5V when the stick is pushed in the positive X or Y direction, and 0V in the negative X and Y direction. We changed the joystick to output to analog pins of the ATMega, combined the X and Y outputs of the joystick, and altered the code associated with the joystick to run on a change in voltage instead of a general output.

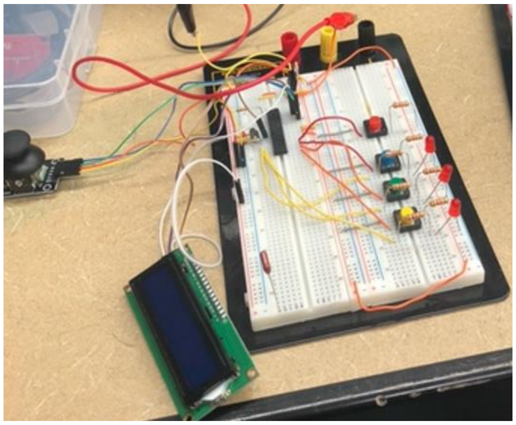


Figure 9: Design Bread-Boarded

After building the original design on the breadboard, we built the same design on our first PCB. After adding external wires and resistors for pulldown resistors for buttons and using more wires to properly connect the LCD to the ATMega, our original design was able to work as expected.

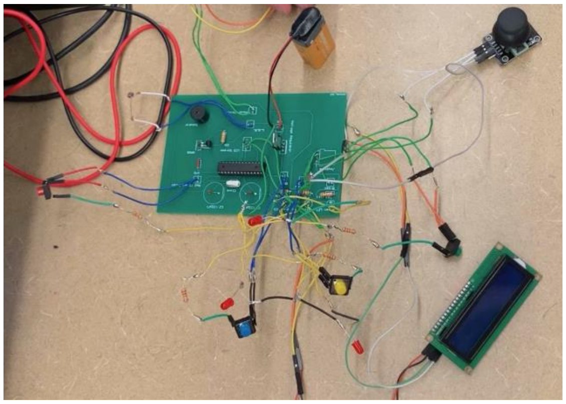


Figure 10: Original PCB Soldered Together

Finally, when the second PCB design was available, we built the circuit again on the new board, double checked the functionality of the design, and put the circuit into the enclosure.

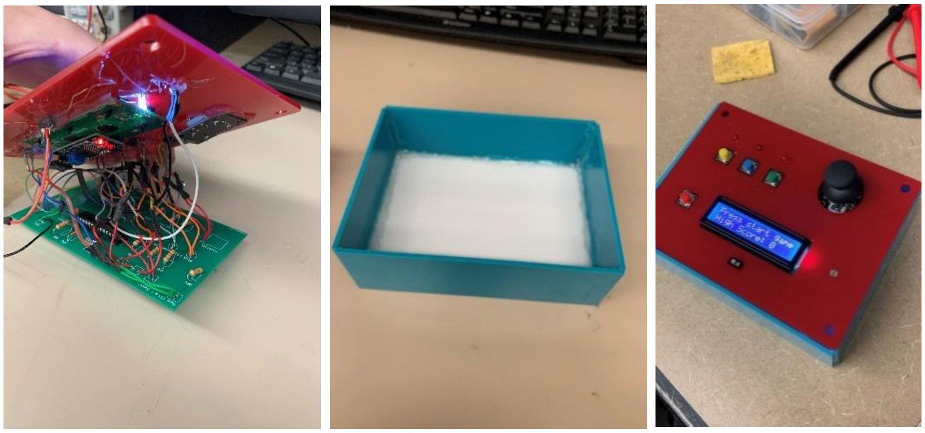


Figure 11: Revised PCB Soldered Together and Glued in Enclosure

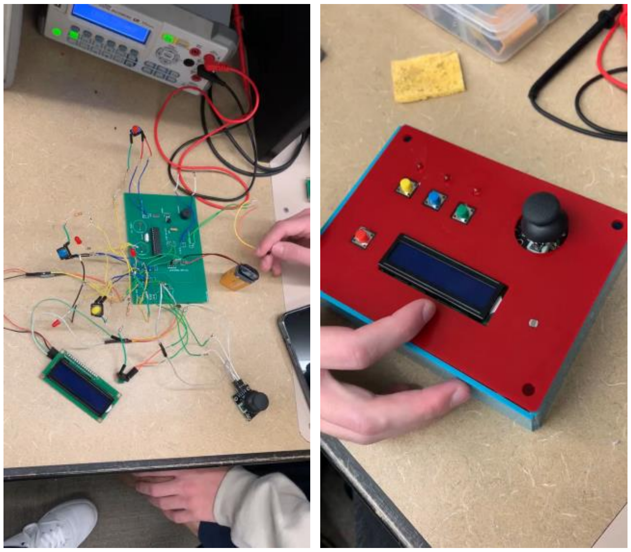


Figure 12: Video of Completed Circuit Functioning

Budget and Cost Analysis:

Estimated cost to manufacture the PCB design: $18

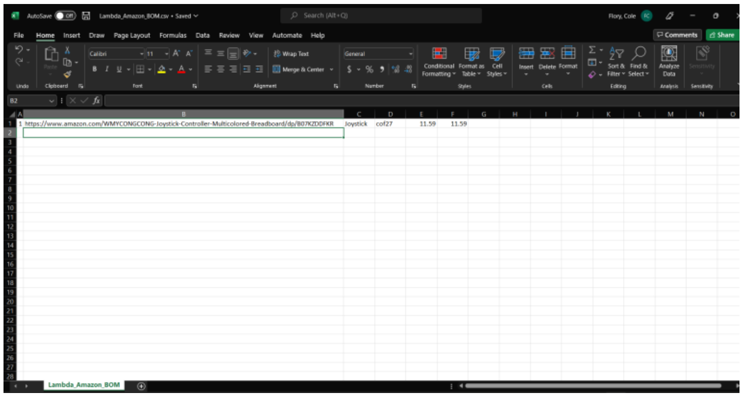
Cost associated with assembly: $165.62

Bill of Materials: $35.76

Estimated Enclosure Design: $383.27

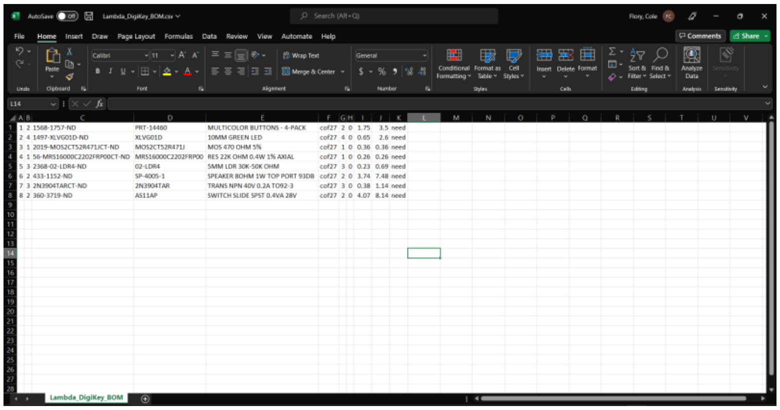
Total: $602.65

Estimated cost of producing 10,000 copies of our design: $6,026,500



	A	B	C	D	E	F
1	1	https://www.amazon.com/WMYCONGCONG-Joystick-Controller-Multicolored-Breadboard/dp/B07KZDDFKR	Joystick	cof27	11.59	11.59

Figure 13: Amazon BOM



A	B	C	D	E	F	G	H	I	J	K
1	2	1568-1757-ND	PRT-14460	MULTICOLOR BUTTONS - 4-PACK	cof27	2	0	1.75	3.5	need
2	4	1497-XLVG01D-ND	XLVG01D	10MM GREEN LED	cof27	4	0	0.65	2.6	need
3	1	2019-MOS2CT52R471JCT-ND	MOS2CT52R471J	MOS 470 OHM 5%	cof27	1	0	0.36	0.36	need
4	1	56-MRS16000C2202FRP00CT-ND	MRS16000C2202FRP00	RES 22K OHM 0.4W 1% AXIAL	cof27	1	0	0.26	0.26	need
5	3	2368-02-LDR4-ND	02-LDR4	5MM LDR 30K-50K OHM	cof27	3	0	0.23	0.69	need
6	2	433-1152-ND	SP-4005-1	SPEAKER 8OHM 1W TOP PORT 93DB	cof27	2	0	3.74	7.48	need
7	3	2N3904TARCT-ND	2N3904TAR	TRANS NPN 40V 0.2A TO92-3	cof27	3	0	0.38	1.14	need
8	2	360-3719-ND	AS11AP	SWITCH SLIDE SPST 0.4VA 28V	cof27	2	0	4.07	8.14	need

Figure 14: DigiKey BOM

Team:

All members of the group played roles in all aspects of the design process, but the main roles were broken up as seen below:

Main roles:

Cole: PCB Design & Enclosure

Lucas: Assembly

Zach: Software

The main methods of communication between the group were text message and email.

Timeline:

Week1 (Oct10-14):

Design proposal

Initial B.O.M.

GitHub and workflow tutorial

Week2 (Oct21-25):

Testing of inputs for parts that arrived, made modifications based on results

Redesigned PCB based on modifications and resubmitted

Began creating code

Week3 (Oct31-Nov4):

Finalized code

Created 3D printing files for enclosure design

Submitted design to be 3D printed in SERC

Week4 (Nov7-11):

Modified enclosure to be laser cut

Laser cut enclosure and assembled it

Assembled original PCB and troubleshooted until functional

Received and assembled new PCB, changed power switch, assembled enclosure with PCB and all buttons

Finalized code and project

Summary, Conclusions and Future Work:

In summary, our project started with testing components on the bread board, writing a program for the ATMega chip, and designing an initial PCB. We then had to redesign and reprint the board after finding small issues with the hardware of the circuit, in addition to making the necessary changes to the software that was then programmed onto the ATMega. Lastly, we designed an enclosure to be laser cut and attached together to house the PCB and all the components. If we had to improve the design, we would add an SD card and a more sophisticated audio output and alter the enclosure to reflect the change. Additionally, we would’ve laid out the components on the PCB differently to avoid stretching the wires connecting the components to the PCB.