Tadd Bindas

CMPT 422N  Dr. Rivas

The Arduino ipod



Abstract:

‘The Arduino iPod’ is a combination/push button iPod made from the components in my Arduino Uno kit. The project is a way to store songs with a combination lock and demonstrate my skills as a computer scientist. Consisting of two Pushbuttons, four LED lights, an LCD display, a Piezo, and a Potentiometer, this project plays songs on the piezo and uses the user’s input of a four-pinned combination to unlock the media. My idea for this project was inspired from my phone lock screen. I wanted to replicate a phone lock screen on my iPhone, and use the input from my phone to unlock something using my Arduino Uno. After a series of twists and turns, the project morphed into an combination iPod which displays a light show when it plays music. This project is so cool because it encompasses most of the elements we learned in class and squeezes them together onto two breadboards.

Introduction:

My idea for this project was inspired from my phone lock screen. I wanted to replicate a phone lock screen on my iPhone, and use the input from my phone to unlock something using my Arduino Uno. I wanted to eventually use this to remotely unlock a door. This, however, proved to be more of a daunting task that initially thought up. Unwilling to spend extra money and unable to get new parts in time, I decided to change my project. I changed the goal of the project to be a safe. After creating the safe, I wanted to add more to showcase more of what I learned in class. I then added another breadboard and a piezo to create a password protected iPod.

Methodology:

I created this project by first creating lab eleven from the ‘Getting Started with Arduino’ book. This lab is called the “Crystal Ball” Lab. The crystal ball lab described how to hook up the LCD display to the Arduino and use an input with it. The input used in the lab was a tilt. The tilt acted like a crystal ball because when shaken or pressed down it changed states. Whenever the tilt’s state was changed, the program picked a random response and displayed it to the LCD display. Using the lab as an outline, I moved the potentiometer to the bottom of the breadboard and replaced the tilt with two push buttons. The bottom button, or the left most button when looking at the LCD display, scrolls through the available options on the screen. The top button, or right if looking at the LCD display, sends the Arduino whatever input is on the LCD display.

Next, I added more elements to the project to further showcase my knowledge of how an Arduino works. Originally, I wanted to attach a piezo to add sound effects to compliment the user pressing the push buttons. The only issue with this idea was that my piezo was broken. Since I could no longer do that idea, I decided to add eight LED lights to indicate when the user enters input for the combination lock. If the user entered correct input a green LED lighted up. A red LED also lighted up for incorrect inputs. I soon ran into an issue where there were too many lights but enough digital pins to have them all work. My solution was to cut the number of LEDs to four. Having four lights solved the space issue and allows for the user to see if they entered a number in or not.

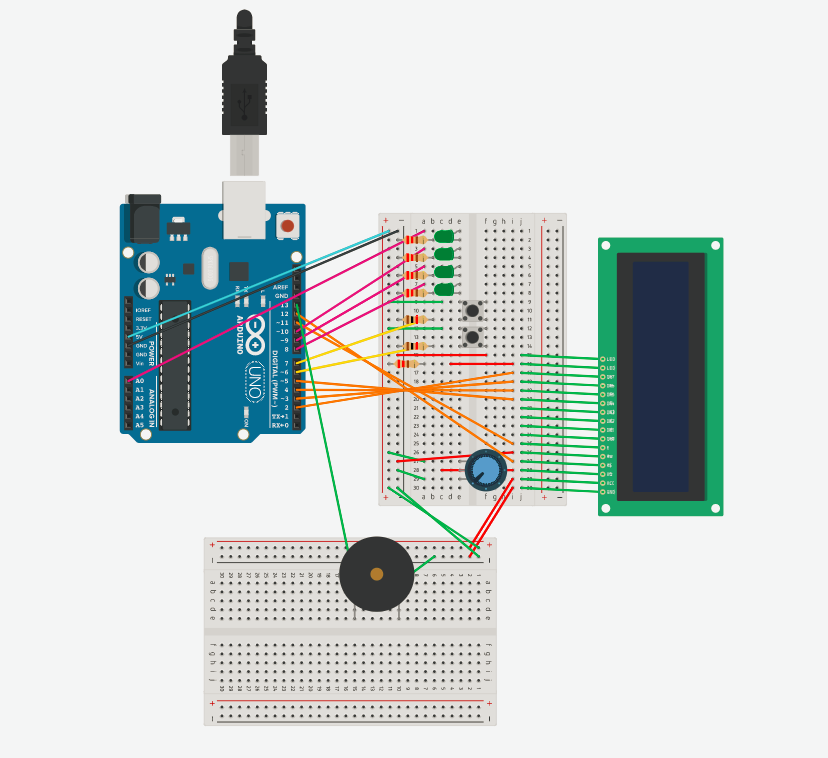
I soon came to realize my project was not challenging enough. I did not go through enough of a learning experience while creating the combination lock and I wanted more out of my project. Next, I borrowed a breadboard and a piezo from a classmate (Marcus Zimmermann) and added the piezo I wanted to have earlier. This piezo was giving me trouble at first because it would not make a tone when plugged into digital input slot 0. I soon moved it to digital slot 13 and moved a light input to the analog side. I learned that the analog inputs act as digital inputs unless specified in the code, something I would have never learned without pushing myself the extra mile. Once the device is unlocked, the user can scroll through a collection of songs on the device and the piezo will play them as the LCD displays the name and artist of the song. The LED lights also flash to the beat of the tune. Lastly, I added a song that plays when the Arduino starts up. Lights flash to the beat, too. This completed my project.

The biggest issue of the project was that my fingers were too fat to successfully insert the wiring into the breadboard without undoing other wiring. Also, it was a pain using the LCD display because I had to re-wire the thing at least ten times. I would have it working, go to bed, and then it would not work. Next, I fried my LCD, so I had to borrow one from a classmate. I broke a wire inside of the breadboard and had to use tweezers to get it out. (Made sure to unplug it first). I was unable to use my original piezo, so I had to borrow one from my classmate and stick it on a different breadboard. Coding the songs was challenging. I used the beats per minute of the song to find the length of the beat in milliseconds. Then I used a frequency chart to match the notes and some sheet music to determine how many milliseconds I would have to play the note for. I learned a lot during this project and would love to had made more songs for the piezo if I had time.

The reason my project is so cool is because the LED light show at the end goes to the beat of “Butterfly Effect” by Travis Scott. The lights and piezo matching up together is very cool. My project was also cool because the only guidance I had was the Arduino Projects book. I completed the project using what I learned in the class and in other labs.

Diagram:

Below is the diagram of my device:



The 5V and ground are connected to the positive and negative ends of my breadboard to feed the board with power and to ground it. Second, I have four green LED lights connected to the breadboard, each with a 220 Ohm Resistor. The lights take up digital inputs: 14, 10, 9, and 8. They light when the device is turned on, the user inputs a number, and in the end if the user correctly enters the passcode.

Next, I have two pushbuttons. The bottom button is connected to digital input 6 and scrolls through options presented on the LCD display. The top button confirms the user input and is connected to digital input 7. Each push button is connected to a one kilo-ohm resistor.

The LCD display is connected to the breadboard using the pins attached to it. In the diagram, there were no pins available, so I had to connect it using wires. From the top of the LCD to the bottom, the top two pins are connected to the power and to ground, respectively. The ground wire uses a 220 Ohm resistor. Next, the LCD display has six input wires which connect to digital inputs 2, 3, 4, 5, 11, and 12. These are for sending and receiving input from the Arduino. Second to last, the LCD has a pin, located three from the left, which connects to the potentiometer. The potentiometer adjusts the contrast of the screen. Lastly, the LCD display has two more wires linking to both power and ground, respectively.

Lastly, the piezo is connected to ground as well as digital input 13. It plays sound via a vibrating metal plate and takes in an input of Hz to make a sound. It was easy to put in.

References:

Arduino Projects Book, liquid crystal library, frequency to note sheet, and sheet music for “Butterfly Effect” by Travis Scott.