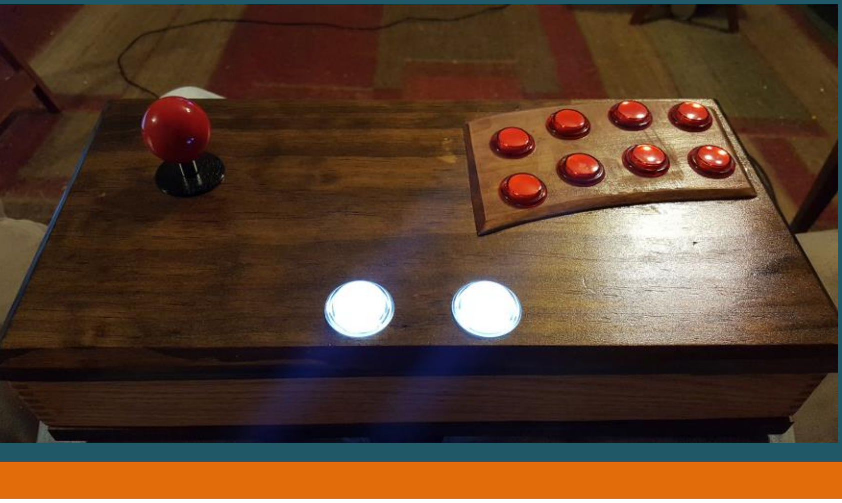
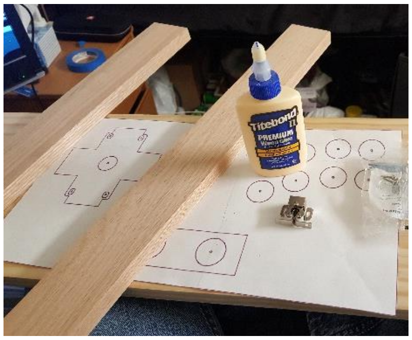


Wireless Bluetooth Joystick

Imgur.com galleries:

Cabinet Build: https://imgur.com/gallery/dA79q
Finished Product: https://imgur.com/gallery/QlinT

This is a project that I'm particularly proud of, as the multifaceted nature of the arcade controller required me to develop my electronic, programming, and even woodworking skills throughout its creation.



Creating the Case

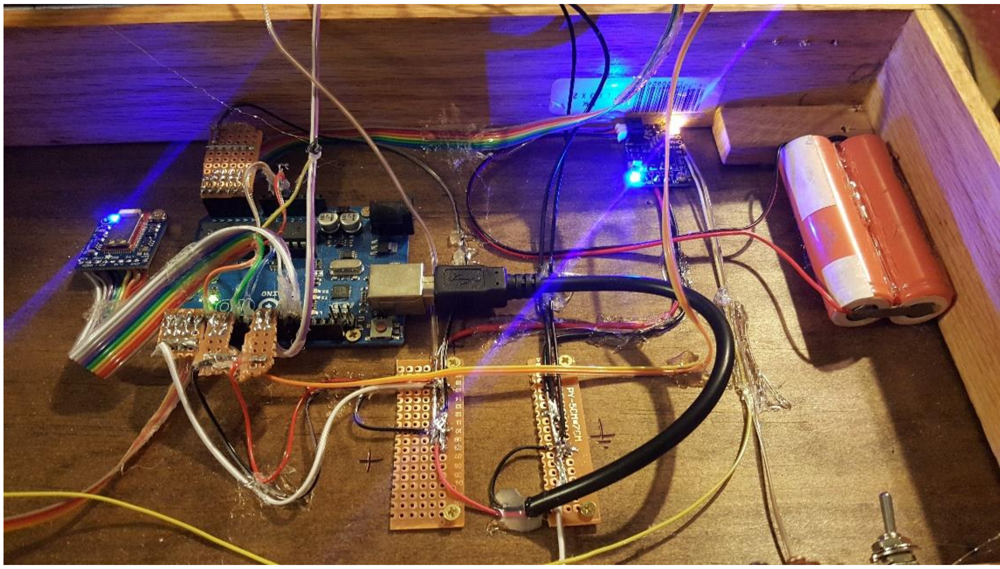
After selecting the joystick and buttons, their layout was drafted in Fusion360. This layout was printed to scale and used to drill out the appropriate holes on the case lid. A simple jig was then constructed to cut and join the cases sides with a box joint. The lid and bottom were made from pine stock while the cases sides were cut from poplar.

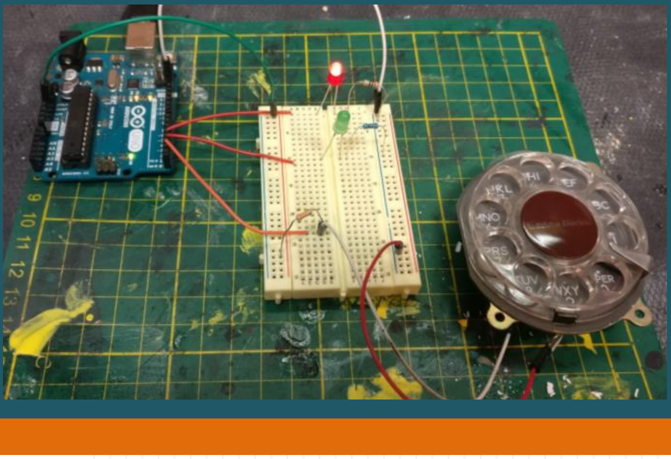
.

The Electronic System

At its core the controller is simply a Bluetooth keyboard, which utilizes an Arduino Uno to interpret button presses and send commands to an Adafruit Bluefruit module accordingly. Coding these Bluefruit commands was far and away the most difficult part of the project, as Adafruit was no longer supporting the module and its documentation was both sparse and hard to track down.

The system is powered by two parallel Li-Po cells, pulled from an old laptop. The 3.7 V cell voltage was stepped up to 5 V using an Adafruit Power-Boost charger. This board contains a boost converter as well as all the circuitry needed to monitor and recharge the voltage of the 2 Li-Po cells.

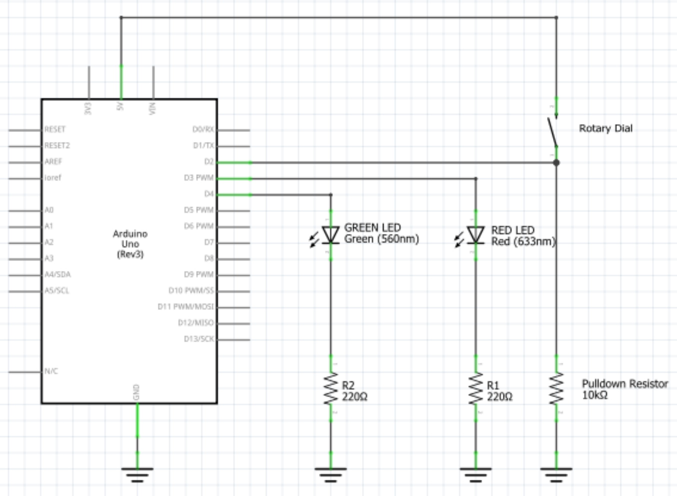


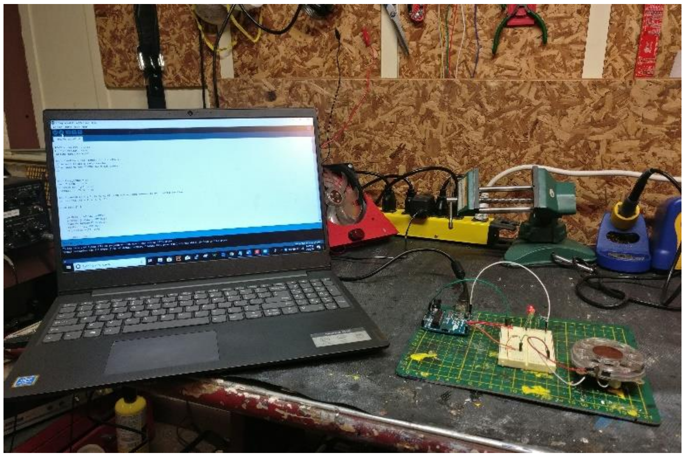


Rotary Phone Arduino Interface

Demonstration:

https://youtu.be/9-KoaOLVdJA



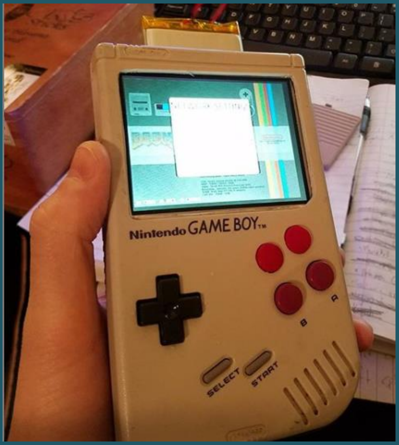


This project was intended to serve as a proof of concept for using a rotary phone’s dial as an interface between a user and the Arduino microcontroller. However, this experiment quickly turned into an opportunity to learn all about the Atmega328P’s various timer and interrupt capabilities and how to manually set registers within the Arduino IDE in order to utilize them.

While disassembling a few legacy phones, I was excited to learn that their dials are completely self-contained devices that operates as a normally closed switch. When the user draws the dial back to a specific number the switch will periodically open that number of times, with specific timing that is dictated by the rotary dial’s inner clockwork. In order to test the possibility of using the device to act as a sort of keypad, the circuit depicted in the top diagram was constructed. Both LEDs have their own digital pins (D3 and D4) which are simply used to communicate whether the correct predetermined code has been entered into the dial. The microcontroller counts the low pulses at the rotary dials pin (D2) in order to determine which number has been entered.

Though the circuit is very straight forward, programming for this operation required an intimate knowledge of the Atmega328P’s timers. A timer had to be utilized in unique ways in order to distinguish one pulse from another, determine when to move on to pulses for the next number, and even debounce the switch.

The project has left me with a much deeper understanding of the registers used to control the 328P and consequently I feel very confident navigating the chips datasheet in order to use any of the registers or features I’m not yet familiar with.



Raspberry Pi Emulator

Demonstration:

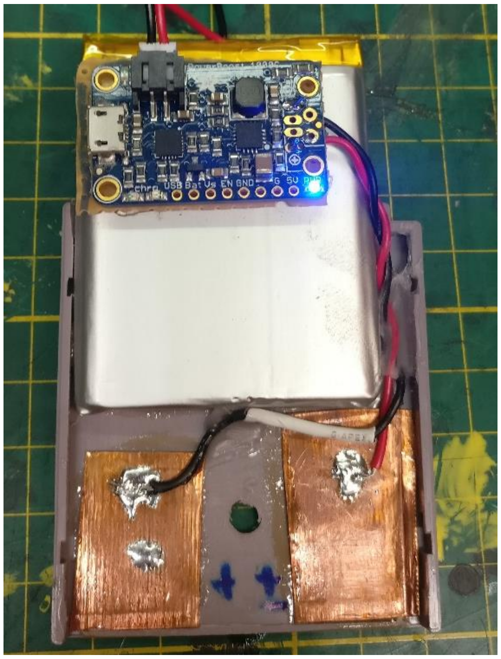
https://youtu.be/zcBz5yVLZEo

Affectionately called the Gameboy-Pi, this handheld console utilizes a Raspberry Pi Zero to run Recall-Box, a popular OS for running all sorts of video game emulators. What's unique about this set up though, is that everything needed for a full gaming experience has been mounted inside of the shell of an original Nintendo Gameboy. Unlike the other projects in this portfolio, I followed various guides step by step to create the Gameboy-Pi. Despite this, the project still required many moments of improvisation and an advanced understanding of working in Linux systems.

The majority of the Pi's GPIO pins are used to handle the input from the 13 buttons, five of which have been added to the Gameboy's original eight. Two additional pins are used for Audio and Video respectively. The screen is a repurposed rear-view camera display, which receives a signal through an AV cable, while the speaker and headphone jack are driven by a modular class D amplifier circuit known as the PAM8403. Since Recall-Box does not default to sending audio or video signals through GPIO pins, various setup scripts had to be edited to accommodate for this display method.

The battery pack is a single Li-Po cell, which is stepped up by another Adafruit Power-Boost charger and mounted on the shell of an old Gameboy game. The 5v from the voltage step up circuit is present on the packs 2 copper pads, so that the system is given power by inserting the cartridge just like an actual Gameboy cartridge.

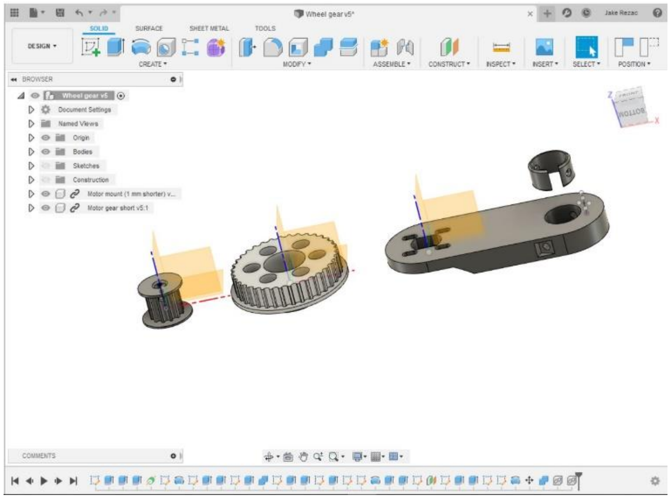
The completed system can be connected to any wi-fi network and accessed over SSH in order to add or update the ROM files on the system. This way, any game supported by Recall-Box can be added to the Gameboy-Pi at the user's discretion.

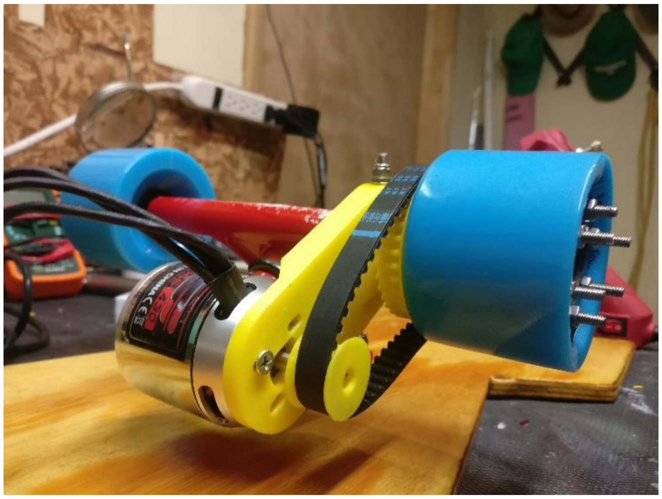


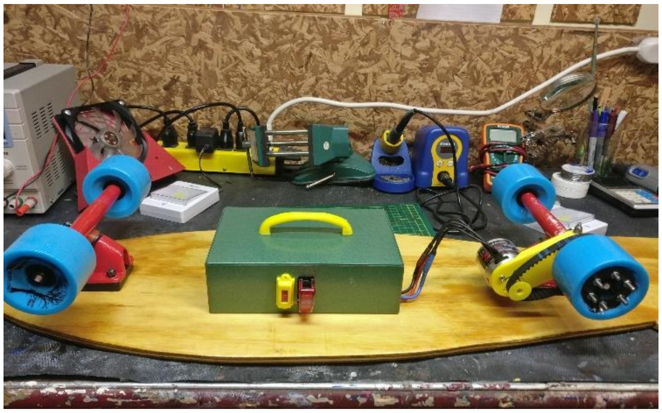
Electric Longboard (Ongoing Project)

Thingiverse Page:

https://www.thingiverse.com/thing:4094054







I began working on the longboard mid-July 2019 so that I would have a reliable way to navigate campus, though soon had to set the project aside to focus on my fall term courses. I've just begun working on the board again and hope to have it ready for the spring term.

So far all of the wood and metal work has been completed, so that all the motor hardware fits together properly and is mounted securely onto the base of the board. The board itself was cut from ¼" plywood, rounded with a hand sander, and treated with linseed oil and 3 coats of polyurethane.

A 430 KV brushless motor rated for a 1750 W (2.35 HP), driven by an 80 Amp Electronic Speed Controller (ESC) and powered from a 6S Li-Po pack, will be the engine of the vehicle. The gears were designed in Fusion 360, based off of other designs I had seen online, and printed on my CR-10 with a 60% infill density in PLA. I'm confident the PLA gears will be able to handle everyday use, but I have been experimenting with lost PLA plaster casting so that I can cast the motor mount in Aluminum if necessary.

Moving forward, work on the longboard will mainly be programming/electrical and the biggest challenge will almost certainly be mounting everything neatly within the housing on the board's underside. At this point the microcontroller is only running a basic sketch to control motor speed with a PWM signal to the dedicated ESC. The final sketch will handle speed control, interfacing with the RC remote, battery voltage monitoring, and breaking functionality.

Other Projects

These projects did not seem large enough to justify giving them an entire page, but I would still be very excited to share them with you and links to project descriptions, demonstrations, or relevant websites have been provided for your reference below:

- **555 Timer PWM Motor Speed Controller**
  https://youtu.be/89Dhx_Mi5jU

- **3D Printed Wire Spool Holder Mounts**
  https://www.thingiverse.com/thing:3329726

- **Quickdraw Mechanism**
  https://youtu.be/m_8ccsk1Nqw