AUTOMATED LOCKER BUILD GUIDE

University at Buffalo | CSE453

Abstract

This documentation is for the Fall 2020 CSE 453 Locker Team to use as a reference to finish implementing the design for a new RFID based locker system.

Evan Brown and Dee Gao

Spring 2020

TABLE OF CONTENTS

Description

Client Details

- **Project:** RFID Based Locker System
- **Client:** Depew High School (5201 Transit Rd, Depew, NY 14043)
- **Point of Contact:** Craig Uhrich <cuhrich@depewschools.org>

Purpose

This build is designed for a student with visual and mental impairments at Depew High School to improve their ease of use when handling their locker on a daily basis. It uses an RFID (Radio Frequency Identification) mechanism that gets unlocked using an RFID key fob that the student carries with them. RFID is a simple alternative solution for someone that would otherwise have trouble using a traditional combination lock on a High School Locker. The locker also includes a mechanical override component that can be accessed by the school staff in case of an emergency or device failure.

The purpose of this build guide is to make the transition between CSE 453 groups as seamless as possible, allowing the new group to be able to complete the project with relatively few problems. This guide is meant to serve as a reference point that will hopefully address any questions that may arise in the future amongst subsequent locker teams.

Design

Parts List

The Spring 2020 Locker Team has already acquired most of the essential parts needed to build the lock. Below is a list of the parts with their links attached in case they need to be reordered:

1. Lock Style Solenoid 12VDC
2. Arduino UNO R3
3. Adafruit Push-button Power Switch Breakout
4. RFID Card Reader - Serial
5. RFID 50mm Round Tag **x5**
6. Diode Rectifier 1N4001 **x3**
7. TIP102 Transistor **x3**
8. LED RGB Diffused **x6**
9. 220ohm Resistor **x20**
10. DeWalt 12v Lithium Ion Battery
11. DeWalt 12v-20v Lithium Ion Battery Charger
12. Speaker: SP-1504 x2
13. Double Sided 3M Adhesive tape
14. Super Glue
15. Door Handle
16. Key Lock
17. Breadboard, wiring, washers/nuts/screws

Note: All parts are in possession of Evan Brown except for:

- These are with Professor Schindler or in Baldy 200C:
  - Adafruit Push-button Power Switch Breakout
  - Double Sided 3M Adhesive Tape
  - Super Glue
- Must be Acquired:
  - Door Handle
  - Key lock

Recommendation: **If you need to order parts, place the order as soon as possible as shipping may take a while!**

Circuit

This is the schematic for the lock that was originally made by the previous locker team in Spring 2019. We have adapted their schematic to our design by adding the speaker and RGB LED below:

Wire Colors

- **Red**: 12V (Active) and Arduino 5V
- **Green**: Signal (0 or 1)
- **Yellow**: RFID Serial Read
- **Black**: Ground (GND)

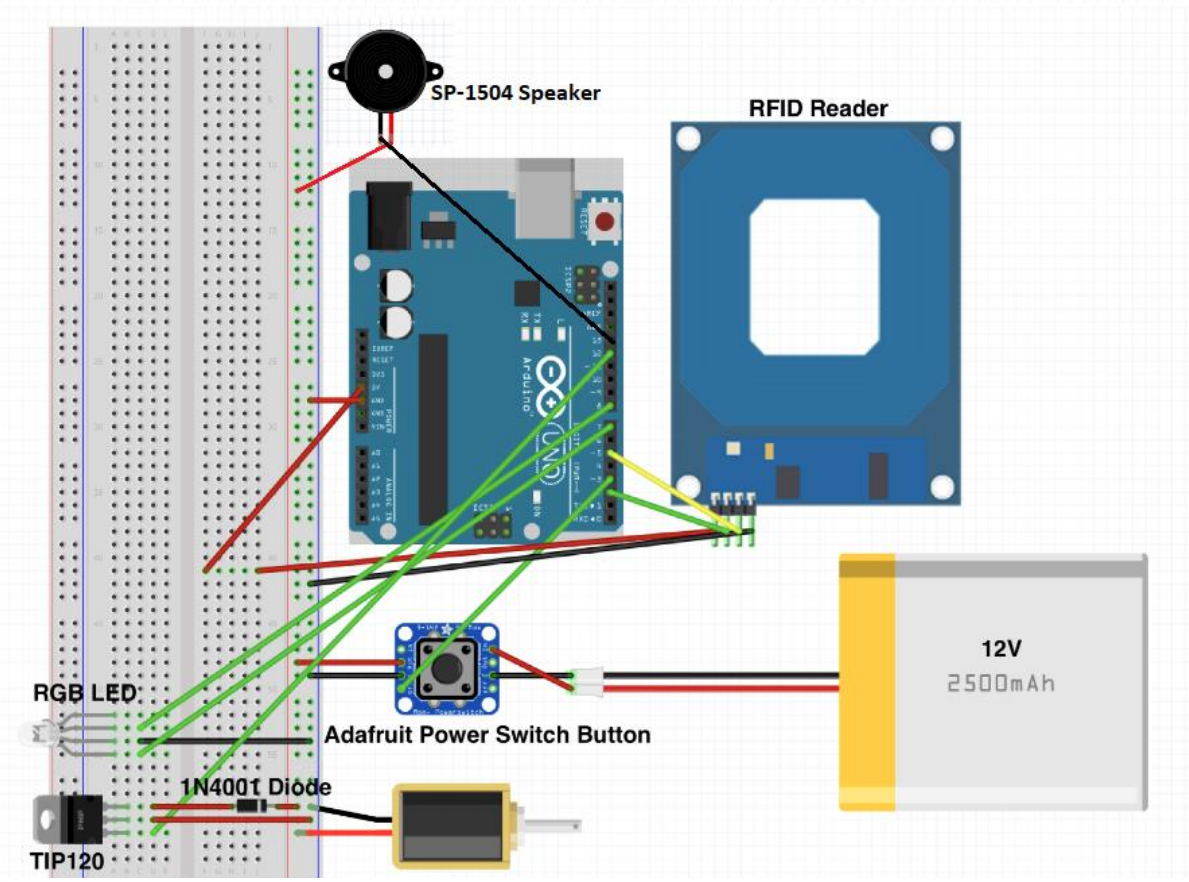

Figure 1. Locker Circuit Schematic

Description

- **Note:** The speaker wires in the diagram are reversed. It should be Red into PIN 13 and Black into the Power Rail.
- **Adafruit Power Switch Button**: This is used to control the power state of the RFID Reader. The battery connects to it from 12V to IN and (-) to G. The button then connects to the circuit with OUT and to ground with G. OFF is connected to the Arduino so the power can be cut after a certain amount of time passes.

- **RGB LED**:
  - **Red** ❼On, but locked
  - **Green** → Unlocked
- **TIP120 Transistor**: Controls the operation of the solenoid, with the BASE of the transistor connected to PIN 13 on the Arduino.
- **RFID Reader**: Having the ENABLE pin connected to 2 and SERIAL pin to 5 sends out signals to ready the RFID reader.

Software

There are two files of Arduino Code: LockerCode.ino and addKeys.ino. LockerCode is the main source code, and addKeys simply reads keys and prints out the RFID tag value on the serial monitor in the IDE. You can add or delete keys in the main code where key values are defined.

Main Flow Chart

This is a flow chart of the logic that the locker should follow for locking and unlocking.

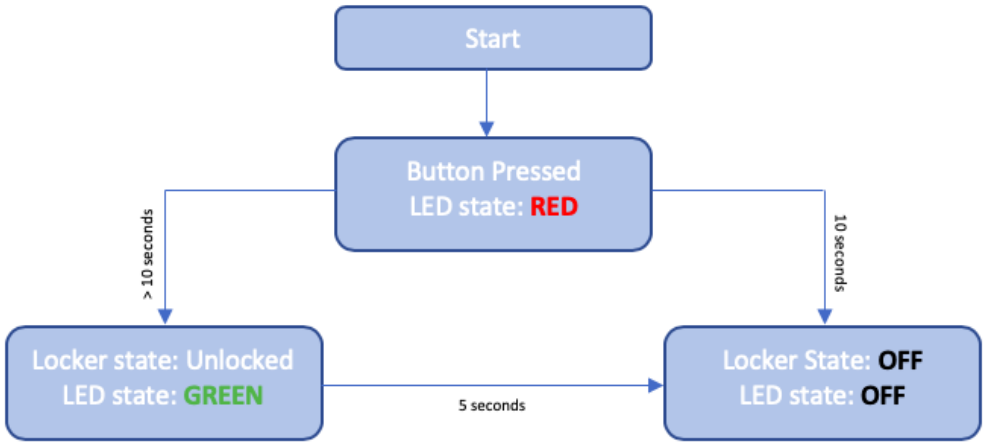

Figure 2. Locker Flow Chart

Programs Needed

1) Arduino IDE
2) Fritzing (schematic design)
3) 3D Printer Software
   a) Flashprint
   b) Fusion 360

3D Printing

The Physical Design is made up of 3D printed parts that are fixed to the inside of the locker. The parts are in the following STL files:

LockSolenoidHolder

This is where the lock solenoid is housed along with the battery holder. Because of this, it is also where the mechanical override mechanism is. If you are going to change anything, it would be this piece depending on how you wanted to do the mechanical override. It is unlikely though that that will be necessary.

LockerInterface

This is where the circuit is housed (Breadboard, Arduino, etc.) and where the RFID scanner is placed. This is a very important part of the locker, as it is directly connected to what the user sees on the front of the locker.

LockerInterfaceLid

This is what connects to the front of the LockerInterface part to make what the user sees on the front of the locker. This is also where the Button and Front Plate are mounted.

Opening Mechanism Folder

This contains the files FrontPlate, KeyGear, and LockGear. FrontPlate covers the RFID Scanner to give a user-friendly look to the front of the locker. KeyGear is connected to the key lock so when turned, catches the LockGear to move the Solenoid and open the locker.

Self-Made Button Folder

This folder contains both the custom Button and Button Holder. The Button should connect to the Adafruit button so that when the user presses it, it triggers the circuit.

For the Button and Front Plate, print these in green material. That is the favorite color of the student and is preferable than the yellow used in the previous build. We never printed these, so we cannot provide more information than was given in the original documentation. However,

they are relatively self-explanatory, and if you get a basic grip of 3D-printing they should be simple to put together.

Mechanical Override

The Mechanical Override uses a sliding mechanism that is created by utilizing the edges of the solenoid and the LockSolenoidHolder part to slide back and forth using the key lock with gears in between.

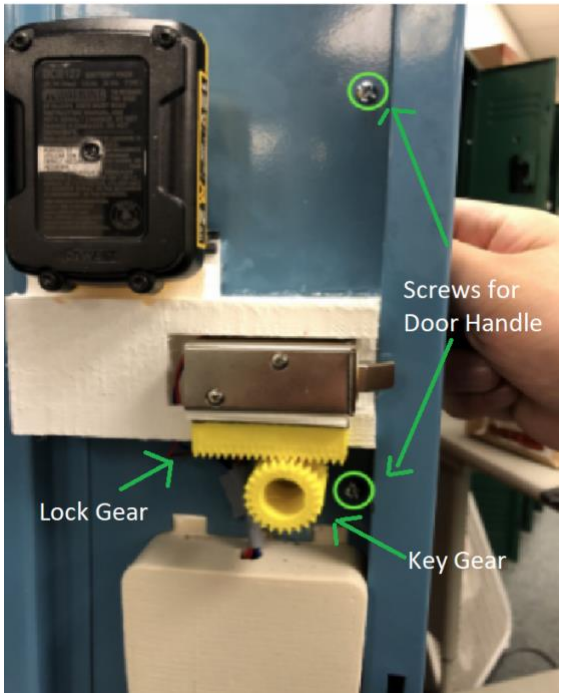

Figure 3. Spring 2019's Locker Design

Figure 4. Lock Solenoid

You can see that the Lock Gear is super glued to the bottom of the Solenoid, which then connects to the Key Gear which can be turned when the key is inserted. To avoid drooping of the Key Gear in the future, you can reverse it, so the gears sit on top of the solenoid. Given the symmetrical nature of the design, this requires little to no change to the parts themselves. The one and only drawback we see with this, is it will reverse the direction the key needs to be turned to lock and unlock, potentially confusing the user who has access to both lockers. However, this is a minor issue and one that can be made clear to the customer.

There are no specific design sources for the key lock itself. However, it shouldn't be anything more complicated than making a hole for it and attaching it to the gear.

Build Specifics and Tips

Battery

The 5V output from the Arduino is not enough power for the Solenoid to function, hence the need of a 12V battery. In order to connect the battery to the circuit, you must take apart the DeWalt Battery Charger and use the connector.

Figure 5. Battery Components

To take apart the charger, you will need a Torx t15 Bit, with a long neck on the screwdriver to be able to reach the screws at the bottom.

The battery should rest on top of the connector when in use and can be removed for charging on occasion.

Button

We purchased a green momentary push button that cannot be used alone, but in conjunction with the Adafruit push button breakout switch. It is solely for aesthetic purposes; however, you may also use the 3D printed model instead if it proves easier.

Figure 6. Green Momentary Push-Button

The momentary push button has not been tested with the Adafruit button due to not having the Adafruit one in our possession in time.

Speaker

The speaker is new to our design. It has a simple implementation, as long as it is connected properly. As mentioned above, the diagram wire colors are reversed, so keep that in mind when connecting it. We recommend having it stick out of the LockerInterface Module and be attached to the inside of the locker. It is magnetic and sticky, so superglue shouldn't be needed (although it doesn't hurt). The code was written for this, but it was only tested separately from the main program. This shouldn't be an issue however as it is a matter of playing a short tone when the locker is powered on and unlocked.

Figure 7. Speaker

The wires on the speaker are quite flimsy and hard to connect, so we recommend stripping the rubber on the wires down and folding them over themselves to make them thicker and more reliable.

Assembly

- ➔ You will need to acquire the locker door to start assembling, so contact the client (Craig) as soon as you get started
- ➔ A cut will need to be made in the locker door to remove the original components and make room for the parts
- ➔ A door handle is also required to be screwed in, as there will be no other way to open the locker without the original mechanism
- ➔ Soldering should be optional. All circuit components should be firmly connected to the breadboard and Arduino (Use a small breadboard so there is more room for components)
- ➔ The circuit is relatively simple and should work with the code provided if put together correctly

Lessons Learned

There were many takeaways from this entire process over the course of two semesters, some through our own faults and some out of our control. We took on this project because we wanted to be part of a real project, one that would help someone. However, due to the circumstances surrounding the pandemic, we were not able to finish the project according to plan because our time on campus had to be cut short. We were not able to utilize the resources that we needed in order to physically work on the project, but we hope that by the time the next semester begins, the situation has improved enough for this project to be completed.

Throughout our two semesters in this class, we learned a few things that may be helpful to future groups moving forward:

- Plan out how you are going to implement your project. You are given a lot of freedom in this class, but don't procrastinate because these are real clients.
- Order your parts as soon as possible! This is probably the most time-consuming step.
- Communicate with your team members to ensure that everyone is on the same page. You are going to be working with these people for the entire year.
- Also, keep open communication with your client! There will be meetings throughout the process with the client to keep them in the loop about how the project is going. We are building the project for them, so it's important that they are up to date with what is going on.
- Be creative!
    - Sometimes, we might not be able to come up with new design ideas right away, especially when there is already a working design in front of us, but don't be afraid to try new things. Since you are given ample time, there is room for things not working the first time
    - However, on the same hand, don't fix what is not broken. If the design is good and it works, there might not be a reason to reinvent it all over again from scratch.
- Most importantly, have fun!

References & Helpful Links

Github: This is where you will get all necessary files, including code, 3D models, and documentation.

https://github.com/shellshocker98/CSE453-Locker-Project

Documentation for other locker designs from Alden High School and Middle School are also included if you wish to reference them.

I recommend going through all of the documents in here before starting, because even though we tried to include all of the important details, we may have missed something small or you may need a bigger picture of the project.

If you have trouble accessing this, or any other questions, please contact us.

Evan: epbrown@buffalo.edu

Dee: dgao@buffalo.edu

I (Evan) am also leaving Kris with my personal email, just in case our Buffalo ones go inactive.