Locker for Student with Special Needs

Design Document

Prepared for:
Depew High School

Prepared By:
Tijmen Van Der Beek, Siquan Wang, Miren Patel , Xiaoang Zhang, Dazhou Liu

School of Engineering and Applied Sciences
University at Buffalo, The State University of New York
Spring 2021

Table of Content

Introduction

The Depew High School locker project focuses on customized lockers for individuals with physical disability. According to the needs of the client, each lock has to be designed specifically for each user to solve their problem with accessing the locker independently. Our locker is to help a student with physical or visual impairments so that they can operate the lockers without any trouble. Everything can be found on Github.

Software

There is one file of Arduino Code: LockerCodeDepew.ino. LockerCodeDepew.ino is the main source code. This can be obtained on our Github.

Important Static Variables

- **tagDelay:** This is the processing delay between each tag read in ms
- **ERRORBlinkRate:** This is the speed at which the led binks in ms
- **LOCKER_TIME:** 30 This is the time the locker has till it turns off in sec when locked
- **UNLOCK_TIME:** 10 This is the time the locker will be unlocked for in sec when Unlocked

Functions

- **void ReadSerial(String &ReadTagString):** This function reads the tag in front of the RFID sensor and passed the string through ReadTagString
- **void clearTags():** This function clears all the tags in EEPROM
- **void loadTags():** This function loads all the tags from the EEPROM into the global tag_list
- **bool writeTag(String data):** This function adds a tag to memory and returns true if successful and false if reached max tags
- **bool checkTag(String data):** Checks the given tag with the registered tags if any of them match it returns true.
- **void setLed (int rVal, int gVal, int bVal):** sets the rgb led color of only the status LED
- **void ledBlink (int rVal, int gVal, int bVal, int halfCycle, int blinkCount):** Blink the RGB LED given the provided RGB value, the time for each half cycle (time off, time on, in ms) and the number of blinks. (Only Status LED)
- **void soundFeedback(int type):** adds sound effects to the locker for better feedback (0 start, 1 unlock, 2 wrong tag)
- **void setBatteryLed():** sets the battery led color based on the voltage of the battery

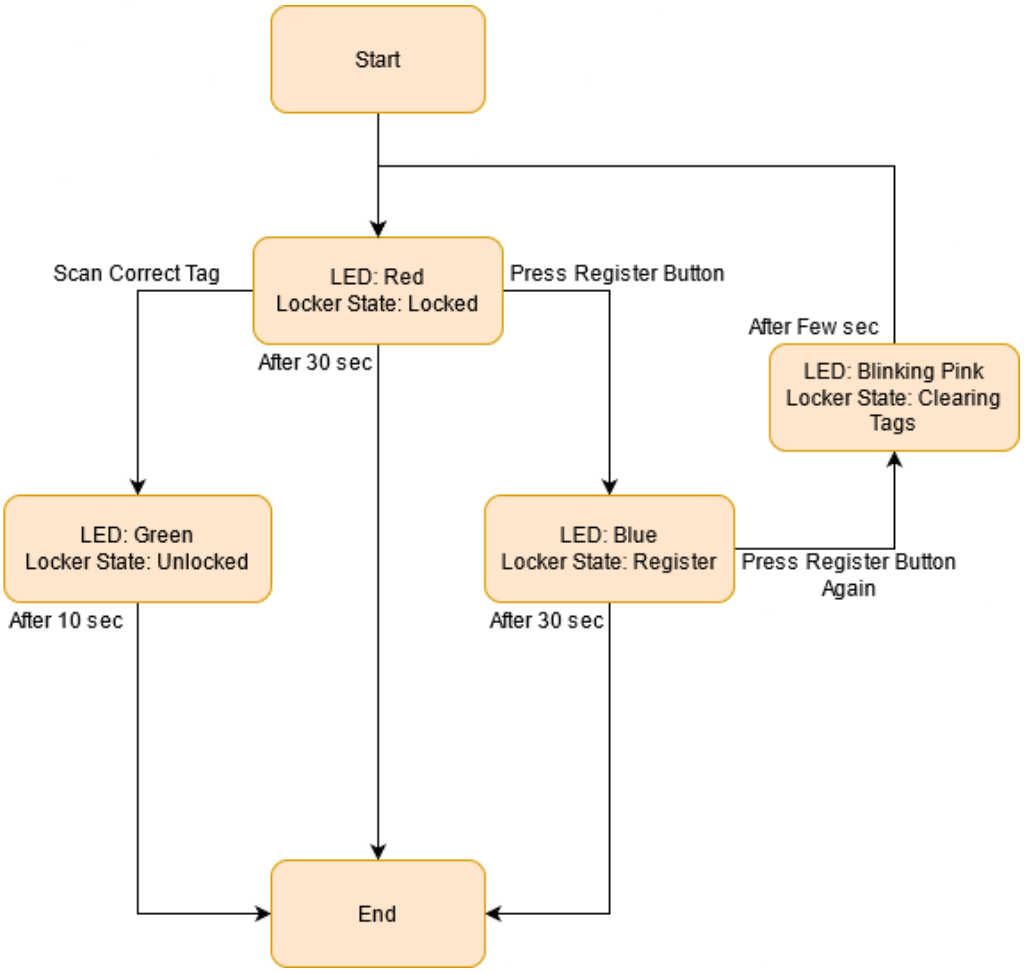
- **void enableTimerInterrupt():** Sets the timer interrupt to interrupt every second (We have a counter to determine the time)
- **void resetTimer():** resets the timer counter back to Locker_Time whenever called

Interrupts

- **void buttonISR():** sets clear_tags to true and sets led to pink
- **ISR(TIMER1_COMPA_vect):** Turns off locker after timer interrupt activates and counter is at 0

Main Flow Chart

This is a flow chart of the logic that the locker should follow.



RGB LED Indicator Key

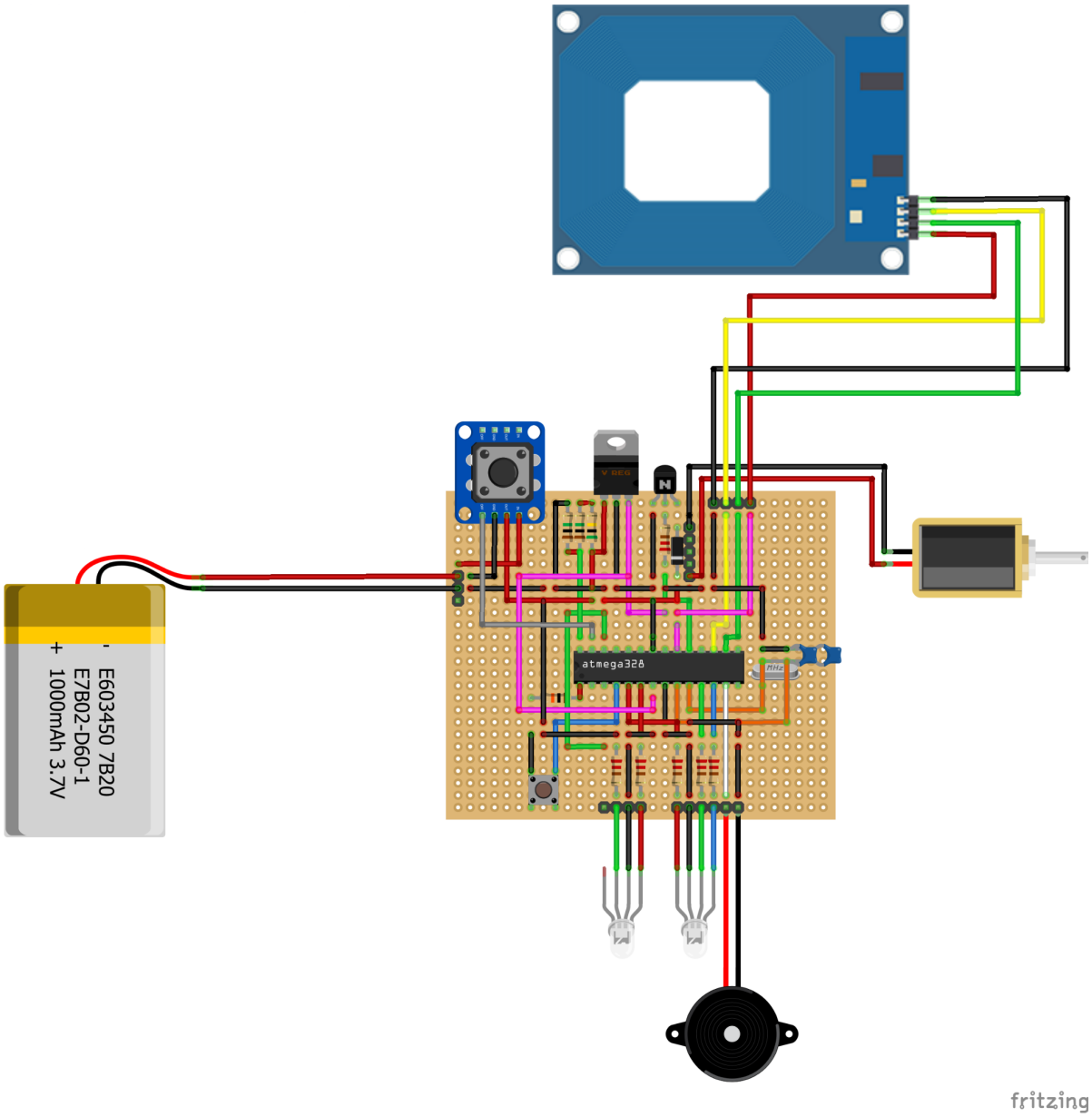
Status LED (LEFT)

State	LED Color
Initial State	OFF
Locked	RED
Unlocked	Green
Processing	Yellow
Register Mode	Blue
If the tag is wrong	Blinks RED
If the tag is correct	Blinks GREEN
Successful Register	Blinks Blue Twice
Already Registered	Blinks Blue Three Times
Unsuccessful Register	Blinks Red
Cleared All Tags	Blinks Pink

Battery LED (RIGHT)

State	LED 2 Color Battery Indicator LED
Initial State	OFF
When the push button is pressed, it turns on the circuit and is ready to read the tags.	**GREEN:** if the battery level is sufficient (battery is greater than 10 V) **RED:** if the battery is at low level (the battery is less than 8 V but greater than 7 V) **Blinking RED:** if the battery is low (the battery is less than 7 V). At this point the circuit won't turn on, battery recharge is necessary
OFF State	OFF

Schematic



Description

- We are using the atmega328p Chip that we program with the arduino and put on our circuit.
- The battery is a 12 3Ah Dewalt battery the one in the schematic is a placeholder image
- Adafruit Power Switch Button: This is the power button for the locker with an off pin on it allowing us to turn off the locker after being idle for a certain amount of time.

- The voltage divider takes the 12 volt power line and reduces so the chip can read it and set the Power Led appropriately. The voltage divider is two 1M Ohms, one 470k Ohm and the output voltage shouldn't exceed 5 volts.
- The Voltage regulator brings down the 12 volt to 5 volt to power the chip and RFID reader.
- The 2N2222 Transistor is used to power the solenoid with the chip.
- Status RGB LED:
    - Red On, but locked
    - Green→ Unlocked
    - Blue→ Register mode
    - Blinking pink→ Clearing all tags
    - Blinking blue→ Successful tag registration
- Power RGB LED:
    - Red Low Battery
    - Green High Battery
    - Blinking Red Out of Power
- We are using headers on the circuit board so parts can be easily replaced.
- We are using 2 22pF capacitors and a 16Mz Clock to run the chip.
- We have a 10k Ohm resistor on the reset pin on the chip.
- The resistors on the LEDs and 2N2222 transistor are 220 Ohms.

Parts List

1. Arduino UNO R3
2. Atmega328 chips (Remove from Arduino board)
3. Adafruit Push-button Power Switch Breakout
4. RFID Card Reader - Serial
5. RFID Tags x5
6. Diode Rectifier 1N4001 x3
7. Voltage Regulator
8. 2N2222A Transistor x3
9. LED RGB Diffused x6
10. 220ohm Resistor x20
11. 1M ohm Resistor x2
12. 470k ohm Resistor x
13. 10k Resistor
14. 22pf Capacitors x2
15. 16Mz Clock
16. DeWalt 12v Lithium Ion Battery
17. DeWalt 12v-20v Lithium Ion Battery Charger

18. 5V 2 Terminals Buzzer x2
19. Key Lock
20. Solenoid
21. PCBs
22. White Filament (Used for LED caps)
23. Breadboard, wiring, washers/nuts/screws
24. Push Button(Activation)
25. Push Button(Reset)
26. Magnetic Push Latches (Optional)

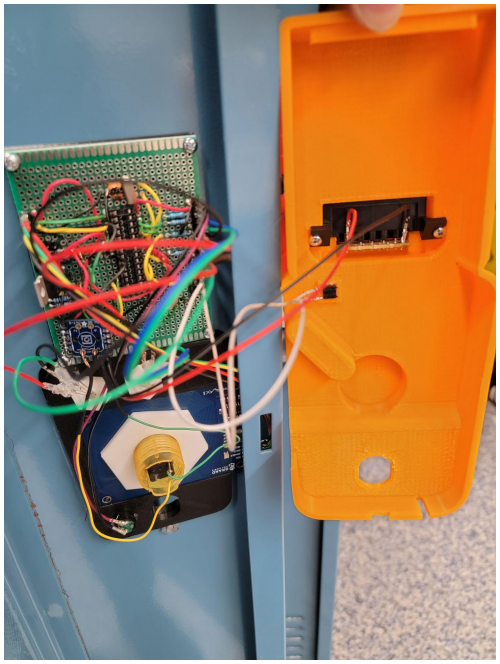
Case

Front

The front of the case has two LEDs, the power button and the lock for the mechanical override.

Inside Front

The inside shows the RFID reader in the center with the button going through it. We glued the speaker in the bottom corner. We have 2 LED caps screwed in with nuts above the RFID reader. On the top is where we screwed the circuit board. On the bottom is where the lock core goes.

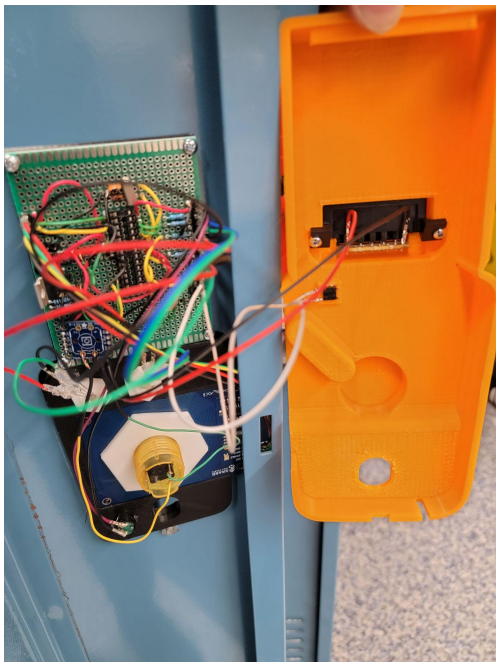


Back

We have the spot for the dewalt battery and the register/clear button below that. The register/clear button is recessed so that it can’t be accidentally pressed and must be pushed with something like a pencil.

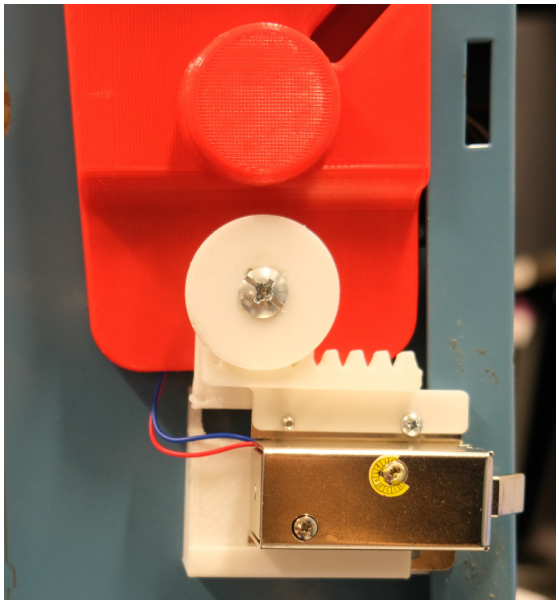
Inside Back

We used the charging insert from the walt charger and screwed it into the locker case. We glued the register/clear button to the back. The prints can be found on Github.



Mechanical Design

The mechanical override utilizes a locker gear and a key gear to horizontally slide the solenoid. The locker gear is a jagged rectangle plate that mounts the solenoid with screw holes. The key gear is screwed to the locker core and is attached to the locker gear.
When a key is inserted into the keyhole and turned, the key gear inside is turned around. As a result, the locker gear with the solenoid screwed to it is slided back and forth. When the solenoid is moved far enough, the solenoid no longer acts as a door latch so that the locker door is unlocked. With the spring mechanism the door must be pushed in before turning the key. The friction of the spring on the solenoid causes the solenoid to move properly. All the 3d parts can be found on our Github.



Upgrades

- Improving the mechanical override with compact gear built with better materials to improve smoothness of unlocking process or changing the whole design to a superior one when the solenoid can be properly mounted.
- Custom manufactured circuit board instead of hand soldering through-hole board to improve stability and durability of the lock and speed up the soldering process.
- Some improvements to the case and overall design of the locker.

Troubleshooting

- The Circuit will not turn on properly if at all without the RFID reader attached to it.
- If the circuit is not working properly or not turning on try replacing the chip.
- If the circuit turns on but the startup sound sounds off clear all the tags with the button on the back. (This will most likely happen with a new chip that was just programed)
- If the circuit turns on at 7v input instead of 12 v input or is not turning on at all, the voltage regulator might be damaged. Swap a new voltage regulator to fix the problem.
- If the status LED is blinking in red color while being powered by 12 v, the voltage divider might be broken. Desoldering the 3 resistors and resoldering on 3 new resistors with the same resistance should fix this issue.
- If solenoid is not working, this could be either the solenoid or the npn transistor. Swap the solenoid first and test the functionality. If still not working, swap the npn transistor on board.
- If any plastic part is broken, you can reprint the part and replace the broken part.

- If the locker does not read the tags, clear all the tag registration and try to re-register one tag. If still not functioning, check the RFID wire connection with the board and retry. If still having trouble, replace the RFID with a new one. If the problem still persists after these steps, replace the processing chip on the board with a new one.
- If the buzzer does not work, check the wire connection with the board first and retry. If still having trouble, replace the buzzer with a new one.

References

We inherit the previous 2 locker teams ideas then improve and expand from them. This is their Github1 Github2.
Our Github.