Fingerprint Protected Voting Machine

Team 68: Brennan Hartigan, Ashwin Dandapani, and Jeremiah Kennedy

Background/Problem

- **Our project tackles a current and prominent issue**
- 2020 election brought about many questions and concerns with regards to voter fraud and miscounted votes
  - Want to create a way to completely disallow and prevent any potential for fraud
- Socioeconomically unfair process with regards to registering to vote
  - Modern day “poll tax”
  - Registering to vote should be as equally accessible for all people
  - KEY POINT: Not everyone can afford a government-issued ID, but everyone has a fingerprint

Our Idea/Solution

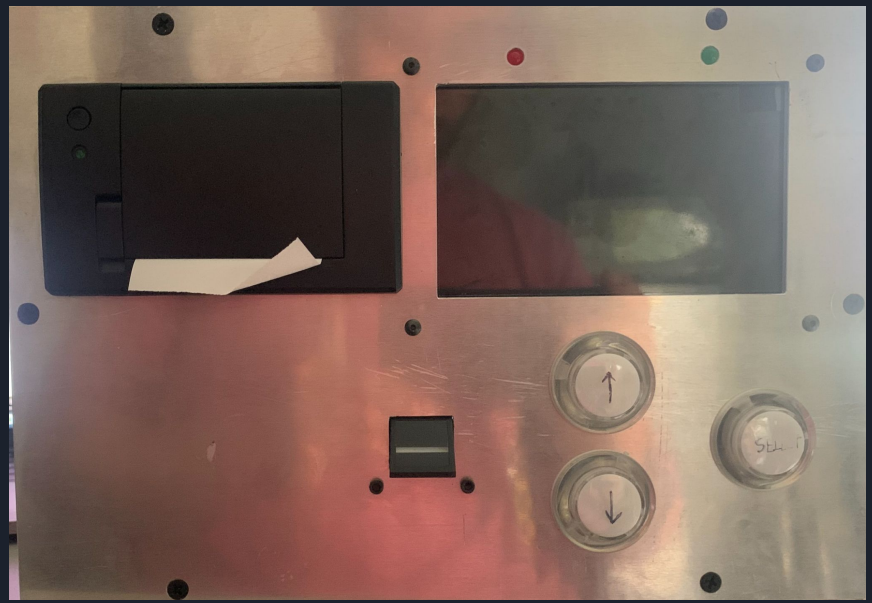
- Voter's fingerprint is completely unique to them, this would be a good form of identification
  - Fingerprint enrollment taken care of by local government
  - Prevents any concerns about potential "modern day poll tax"
- Using a fingerprint, each voter can be identified with near-perfect accuracy and prevented from voting more than once
  - With this is in place, once a user's identification number is mapped to them having voted they will be locked out if they attempt to vote again

High Level Requirements

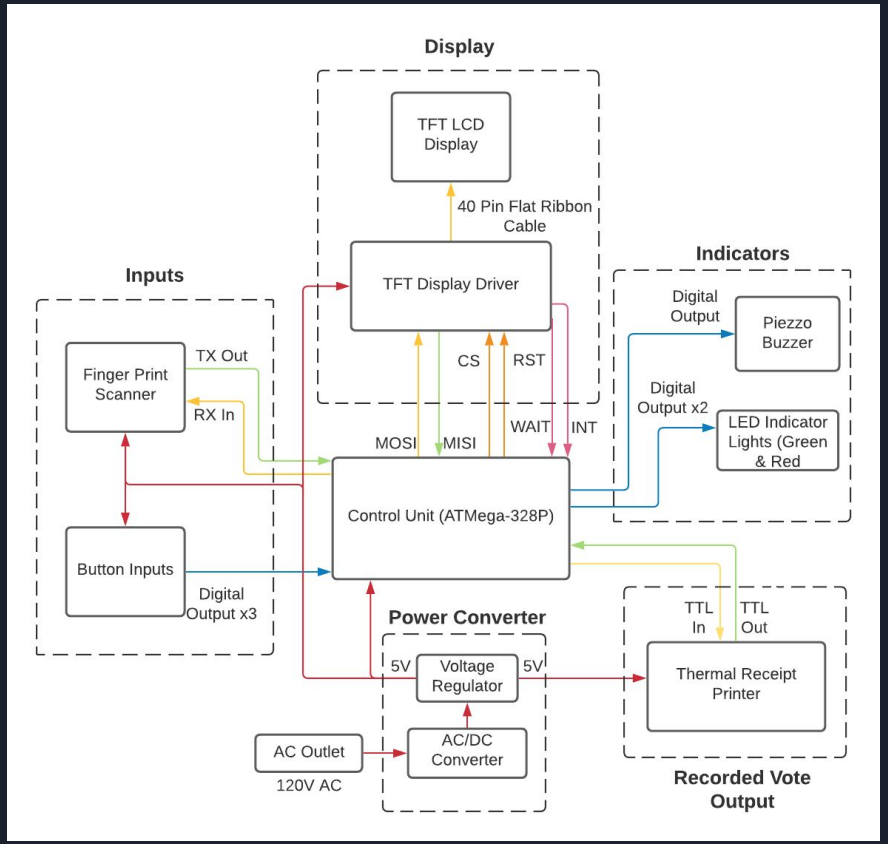
- Voter can scan their fingerprint and successfully make a vote. In order to identify that the voter is allowed to vote, we will look for an accuracy of 95%.
- If the fingerprint of a voter did not have a match after three tries, then the voter is rejected.
- Voter will be able to receive feedback through the LEDs, speaker, and thermal receipt printer after voting is completed.

Features

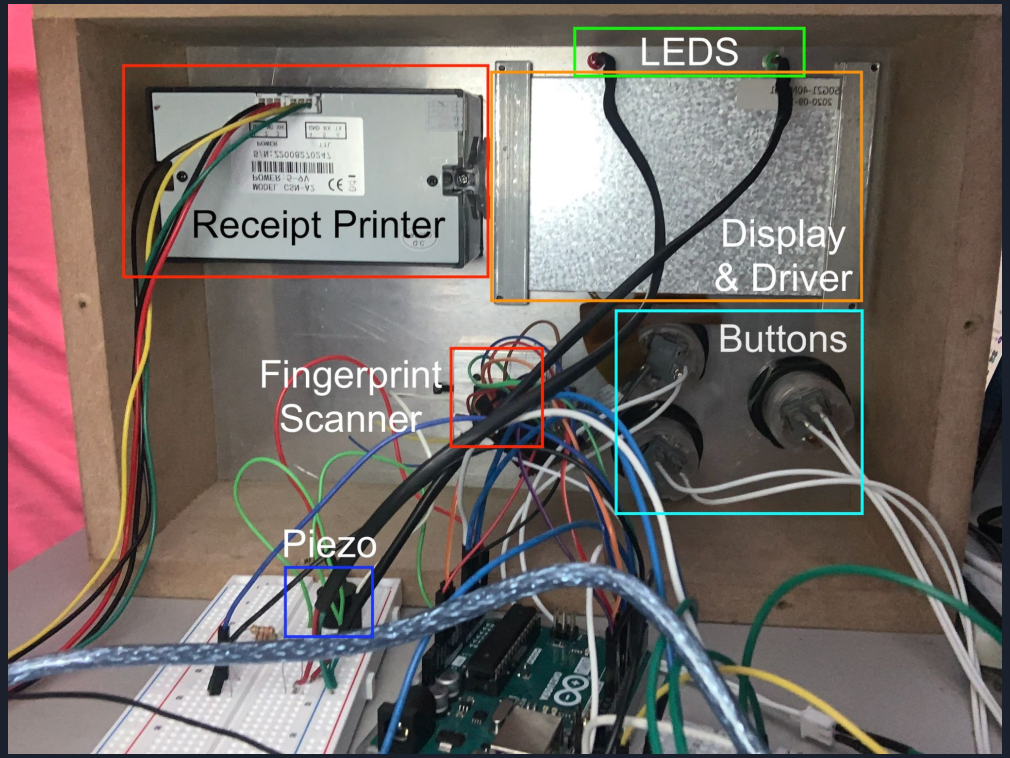
- Fingerprint scanner
- Arcade-style push buttons to navigate UI
- TFT LCD display with driver board
- Thermal receipt printer for ballot dispensing
- Low-cost design, favors mass production



Block Diagram



Physical Design

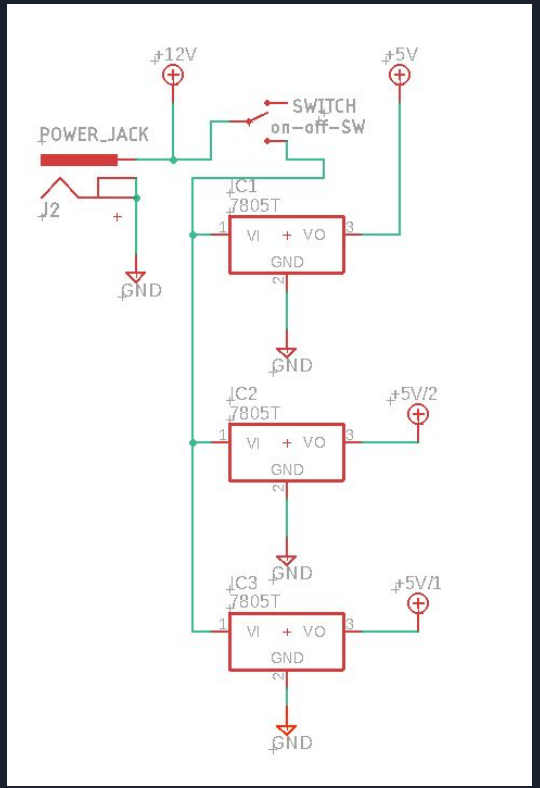


Project Build Timeline

1. Research the problem we are trying to solve, decide on implementation
2. Design schematics, determine what parts will be needed
3. Research and order parts
4. Work with ECEB Machine Shop to design the physical product
5. Begin R&V testing with hardware as it begins to come in
6. Install components into physical housing via ECEB Machine Shop
7. Project on breadboards and Arduino to prototype/debug
8. Transition hardware to PCB, run testing on PCB (N/A)
9. Implement PCB in physical product
10. Demo, presentation

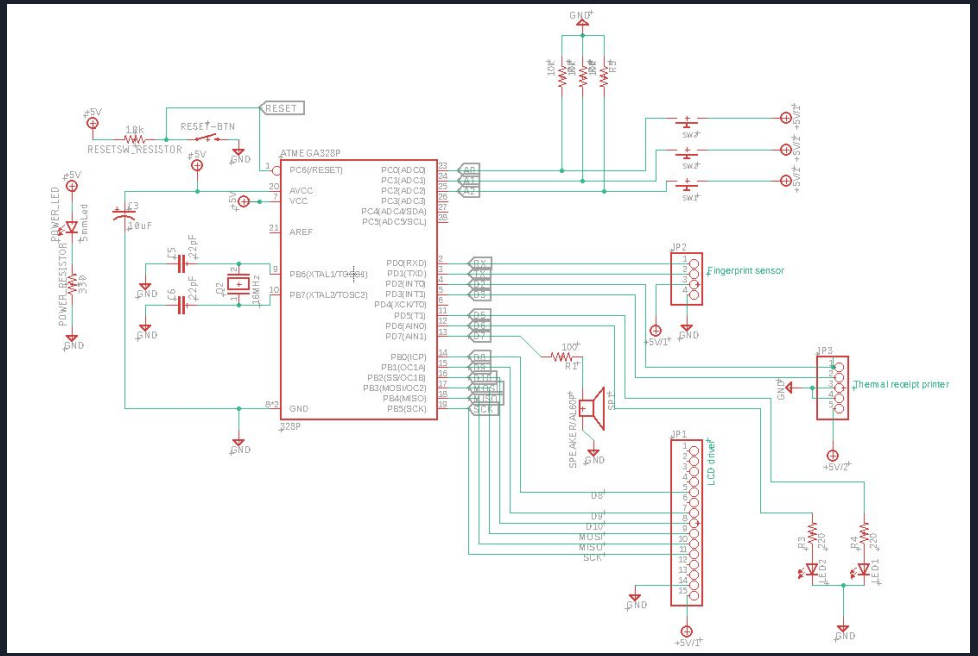
Subsystem 1: Power Supply

- AC/DC Converter
  - Input: 120V AC
  - Output: 12V 5A
- Voltage Regulator (x3)
  - Texas Instruments UA7805CKCT Linear Voltage regulator (LDO)
  - Input: 12V
  - Output: 5V 1.5A
  - Why LDO?
    - Stable & highly accurate output
    - Low-noise output
    - Low cost



Subsystem 2: Control Unit

- Microcontroller: ATmega328P-PU
  - Low-cost
  - Can program using Arduino IDE
  - Lots of resources for debugging and troubleshooting
  - Use Arduino for prototyping and testing



Control Unit Requirements & Verifications

Requirements

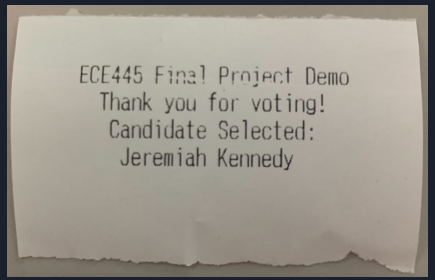
1. The microcontroller must be able to communicate over UART in order to process and transmit the input data from the fingerprint reader.

Verifications

A. Using an LED and the fingerprint scanner, test the microcontroller's ability to process UART signals from the fingerprint scanner in matching. The LED on means a matched fingerprint.

B. Send UART signals to the receipt printer to print out the example text.

Result



Control Unit Requirements & Verifications (cont.)

Requirements	Verifications	Result
2. The microcontroller must be able to receive and send digital TTL signals	A. The LED lights will receive digital signals and light up if a user is granted access or denied B. The buzzer will buzz if a user is denied	 

Control Unit Requirements & Verifications (cont.)

Requirements	Verifications	Result
3. The microcontroller must be able to communicate over SPI in order to echo movements on the screen.	A. The screen will respond to digital input signals from the buttons	

Subsystem 3: Input Devices

- Fingerprint scanner
  - Inputs: 3.3-5V, RX input data from microcontroller
  - Outputs: TX output data to microcontroller
  - Onboard memory can hold up to 128 fingerprints
    - Mass use needs a device with more storage
  - Easy to use with Arduino IDE & Fingerprint Sensor Library
- Buttons
  - Large Arcade-style pushbuttons
  - Input: 5V
  - Cheap, sturdy, reliable
  - Mounted well on our physical design

Software Algorithm: Fingerprint Scanner

- Utilized the Adafruit Fingerprint Sensor Library to enroll fingerprints on the scanner for testing and demo purposes
- Each fingerprint that was enrolled was given a unique integer identification number
- Algorithm:
  - Checked if the scanned finger matches enrolled fingerprint identification number
  - If success then allow the user to enter the voting screen
  - If failure then it will tell the user to retry 2 times before it locks them out

Input Devices Requirements & Verifications: Fingerprint Scanner

Requirements

1. The scanner will have a less than 1 second fingerprint image acquisition time
2. An accurate fingerprint image created within 3 user attempts. If there is an unsuccessful attempt, the screen will display a message asking the user to wipe or clean their finger.

Verifications

1. If the scanner is able to administer the fingerprint and decide a response in a second. The delay will be set in the software.

2.

A. If the scanner is able see a match in the system or if not it makes the user re scan.

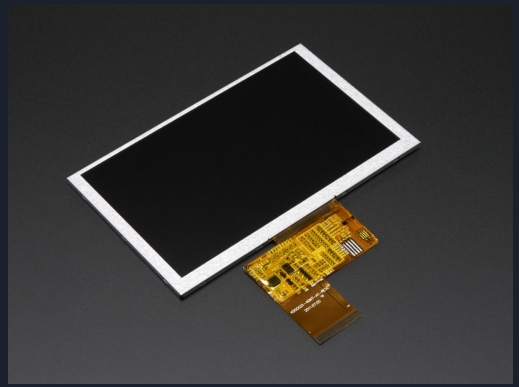
B. If the scanner is able to mention if the user needs to re-scan their finger after checking for a match.

Result

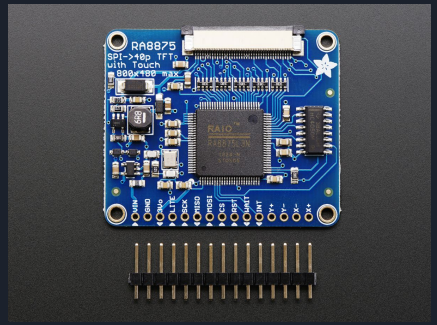
```
Image taken
Image converted
Found a print match!
Found ID #11 with confidence of 214
11
```

Subsystem 4: Display

- Adafruit 5.0” 40-pin TFT LCD Display
  - Low-power
  - Bright
  - Low-cost
  - Easy to mount on physical design

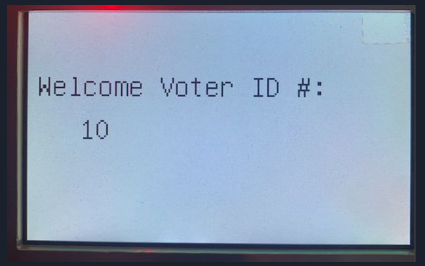


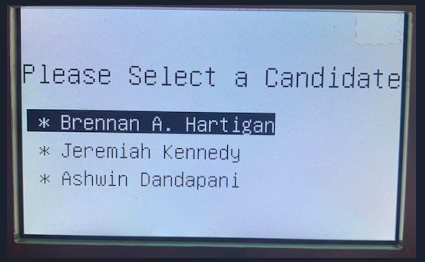
- Adafruit RA8875 TFT Display Driver Board
  - Driver PCB with RA8875 chip
  - Refreshes display at 60Hz
  - Allows display to be used with Arduino IDE, Library for displaying text/images on the screen
  - Communicates with ATmega328P-PU over SPI

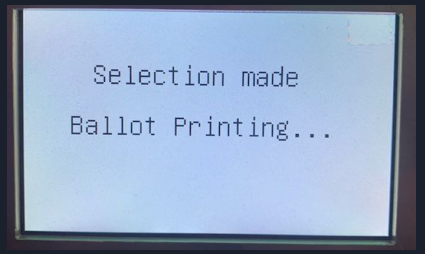


Software Algorithm: LCD Display

- Utilized the Adafruit TFT Library to create display messages on the LCD display for testing and demo purposes
- Each different screen on the display that was used was its own TFT function







Subsystem 5: System Indicators

- LEDs
  - Red for “access denied”
    - Remains ON while the user does not have access to the system
  - Green for “access granted”
    - Remains ON while the user has access to the system and until they are done voting
  - Receive digital 5V TTL signals from microcontroller
- Piezo Buzzer
  - Adafruit PS1240 buzzer
  - Cheap, easy to use with Arduino IDE

Subsystem 6: Recorded Vote Output

- Adafruit Thermal Receipt Printer
  - Most expensive component of the project
    - Costs are reduced for bulk orders for real implementation
  - Communicates with microcontroller via TTL Serial
  - Easy to use with Arduino IDE, Adafruit Thermal Printer library

Functional Testing 1: Granting access to a fingerprint with a match in the system

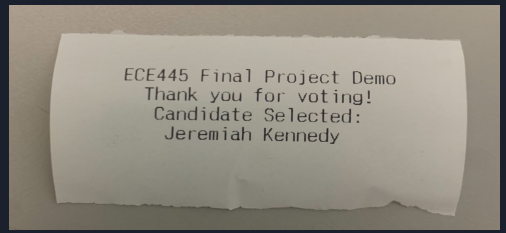
- Scanning correct fingerprint
- LED indicator turns GREEN
- Screen shows “access granted”
- If fingerprint has no match, message is displayed to clean finger, try again
  - LED stays RED
- Result: Success

Functional Testing 2: Using buttons to control the candidate selection process

- After access granted, reaching the “Candidate Selection” screen
- Using buttons to select a candidate
- Subsequent screen messages after this is done
- Result: Success

Functional Testing 3: Printing voter ballot, conclusion of voting process

- Printing the ballot results
- Updating display with relevant messages
- Looping the system program back to the beginning to prompt for a new fingerprint
    - Turning back on RED LED, turning off GREEN LED
- Result: Partial Success on PCB, full success on prototype/breadboard



Functional Testing 4: Consecutive Attempts security

- Lock out a voter if three consecutive failed matches
- Updating display with relevant information
- Each failed attempt accompanied by buzzer tone
- Result: Success

Functional Testing 5: Preventing a voter from voting more than once

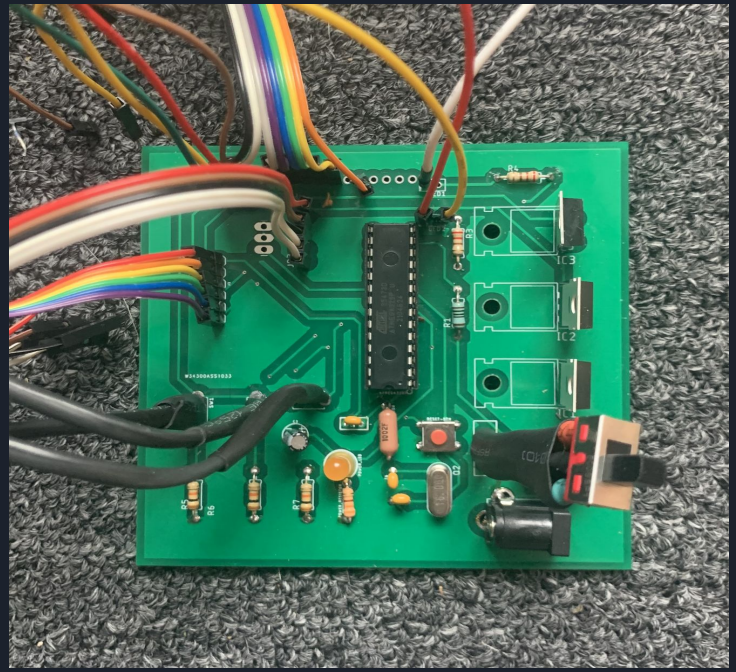
- Keeping track of unique voter IDs by fingerprint, and recording whether each unique ID had voted
- If voter tries to re-access the machine after already voting, update display with appropriate messaging
- Result: Success

Functional Testing 6: “Master” Fingerprint Implementation for Poll Workers

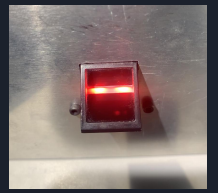
- Machine will prompt the voter to find a poll worker for assistance:
  - 3 Failures
  - Attempt at double voting
- Poll worker scans their fingerprint, system goes back to the main “welcome screen”
- Result: Success

Microcontroller Testing - PCB

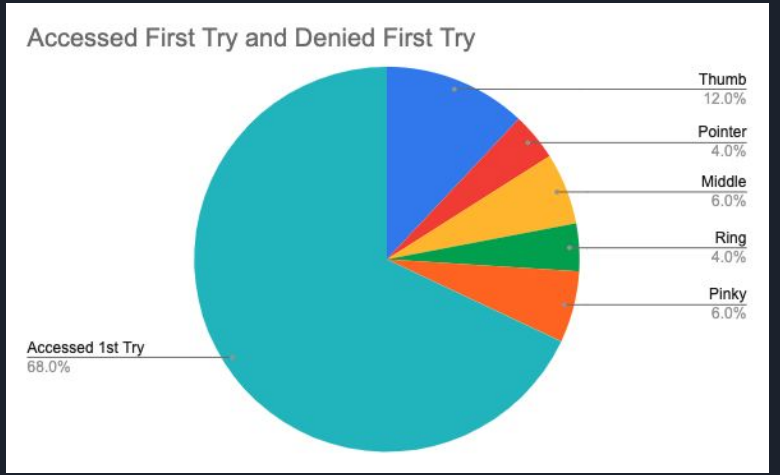
- Successes
  - Flashing Light
  - Button interaction
- Issues
  - Fingerprint Sensor
  - Display
  - Printer



Fingerprint Reliability Tests



- 0% Non-registered fingerprints granted access
- Registered never got locked out
- First Try (with somewhat sweaty)
  - 12% of thumbs asked for clean off
  - 4-6% for other fingers
- Requesting 2nd scan for registered very rare (1-2%)

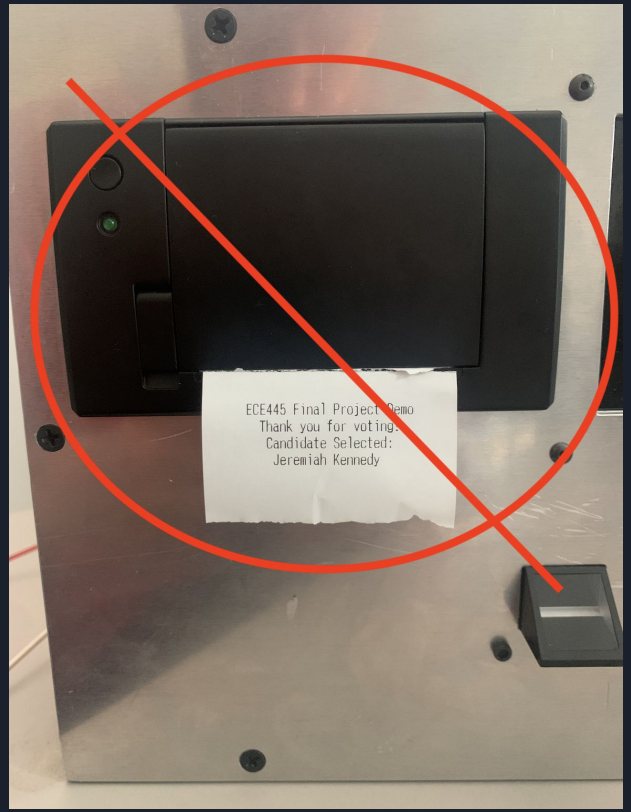


Software Challenges

- Issue with checking double voting
  - Case returned “2”
- 3 Fail would return to welcome
  - Added lockout + poll worker override
- Lack of Dynamic Memory
  - Optimize code or upgrade to ATmega2560

Demo Failures

- Printer not functioning
  - Worked with test code
- Cause: Lack of dynamic memory for local variables
  - Post demo testing
  - Fixes:
    - Use a microcontroller with more DRAM
    - Optimize code so it takes up less memory



Ethics and Safety

- Long process of certification
  - Fed, State, County Level
- Privacy protection
  - Voter demographics only associated with votes
  - Finger prints part of private voter info

Future Improvements

- Completely altering the current voting process not possible immediately
  - Receipt and ballot box for counting votes
  - Make vote counting more virtual to handle potential issues with the receipt printer
  - Storing the vote tally in an online database (no personal voter info)
  - Using a data storing service like Firebase to hold voters' information, whether they are registered, and if they have voted yet as there needs to be more fingerprints enrolled
- Invest in a more accurate fingerprint scanner to prevent any potential mistakes and also expedite the voting process
  - Less likely a voter would need to enter their fingerprint multiple times or accidentally get locked out
  - Quicker and smoother voting process

Credits

- Thank you to everyone who helped our team throughout this semester
  - ECE 445 Course Staff
  - TA: Dean Biskup
  - Professor Fliflet