Arduino Project: ‘Hot Uno’

**Introduction:**

I wanted to make use of some recently acquired electronics hardware, sensors mostly. My idea was to replace a hard copy of my passwords with an overly complex and unsecure solution. I chose to store passwords in an array, only accessed when fire is detected by a flame sensor.

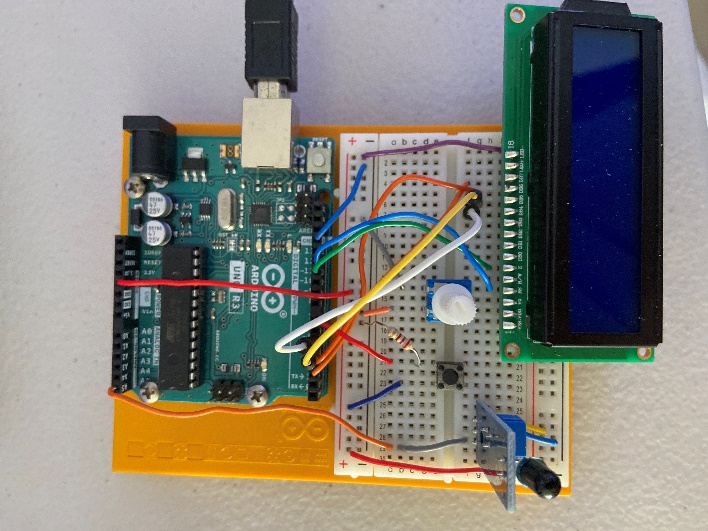
**Methodology:**

1. Hardware:

Connected a flame sensor to the Arduino on analog pin A5 to detect the presence of a flame.

Connected an LCD display via the I2C protocol, allowing password display.

Added a pushbutton to digital pin 7 for user interaction, specifically for scrolling through passwords.



1. Tested Flame Sensor:

Developed and uploaded a basic Arduino sketch to read and print the analog values from the flame sensor to the serial monitor.

Observed the sensor's output to determine the threshold values for flame detection (e.g., lower values indicated flame presence). Typical values when no flame is detected hover around 1,200, 30 when flame detected.

1. Basic Password Display:

Programmed the Arduino to display a list of passwords on the LCD when the flame was detected.

Initially, the passwords were shown sequentially in a loop, automatically after flame detection.

Integrated a button that allows users to scroll through passwords manually, increasing security by requiring physical interaction.

1. Code Improvements:

Introduced a boolean variable (accessGranted) to maintain the state of flame detection.

Allowed password scrolling only if the flame was detected at least once, thus not requiring the flame to be continuously present.

Added simple debouncing to the button press detection to improve reliability and prevent accidental multiple entries from a single press.

Added simple debouncing to the button press detection to improve reliability and prevent accidental multiple entries from a single press.

1. Final Implementation:

Allows scrolling through list of passwords when a flame has been detected once.

**Demo (Photos):**

1. Ambiguous message printed on the LED. The button doesn’t do anything at this point.



1. Flame applied near the flame sensor. Now, once the button is pressed, the secret menu can be accessed.



1. Passwords displayed one by one, index of accessed element in the passwords array is incremented with each button press.



**Future Improvements:**

Expand on authentication process. Fingerprint sensor, keypad PIN required, etc. Code menu could be expanded to include different data. To integrate with other projects, I may store a key needed to decrypt passwords within a password manager program I wrote. I’ll probably use an Arduino Nano in future.