Milestone 4 – Team and Project Commitment/Contract

- **Names, NetId's and Email address of ALL team members**

Name: Maaz Ahmed
netID: mahmed76
Email: mahmed76@uic.edu

Name: Hasan Ali
netID: hali32
Email: hali32@uic.edu

Name: Ammar Idrees
netID: aidree3
Email: aidree3@uic.edu

Name: Umer Qazi
netID: uqazi2
Email: uqazi2@uic.edu

- **Name of your project**

Addiction Prevention Safe

- **Abstract of your project (HARD LIMIT of 100 words or LESS!!!)**

Our project is to implement a safe-like device that is designed to keep the user away from wasting their time on certain tasks (e.g. phone usage, etc). The safe will use some security features such as the requirement of a password as well as a timer which will make the safe unable to open until the timer is complete. For example, if a user wants to stay away from their phone for an hour, they can place their phone into the container, and they will not be able to use their phone until the safe unlocks an hour later.

- **Detailed Project Ideas**

o 3 Required Sections

1. **Overall Description of Project Idea**

- **Information of the project as a whole**

Our project is to implement a safe-like device that is designed to keep the user away from wasting their time on certain tasks (e.g. phone usage, etc). The safe will use some security features (password, fingerprints, etc) as well as a timer which will make the safe unable to open until the timer is complete. The timer is planned to have a default count from one hour, but the user can adjust it accordingly. For example, if a user wants to stay away from their phone for an hour, they can place their phone into the container, and they will not be able to use their phone until the safe unlocks an hour later. Furthermore, the added security features would allow either a user to not have their items stolen once the timer is complete or if a parent wants their child to not have access to the contents of the safe (for example, the parent wants a child to stop using their phone, so now the child can't access it).

- **Information of the Sub-Parts of your Project**

  So this is our main idea. As for the subparts, there are various other things to be considered such what type of fingerprint scanner we wanted to use. We decided on using the optical fingerprint reader sensor module for Arduino. We have to consider the type of lock. We went with a servo motor. There is also the ability to type in a 4 digit pin to unlock the safe. We also wanted to include a timer to display the time remaining until the safe reopens. There will also be LED light. The green will display when the password was entered correctly. The red will display when the password was entered incorrectly.

2. **Initial Project Design including Expected Inputs/Outputs**

The initial design is to use a shoebox that would be the location where the locked valuables will be kept. There will be an LCD screen attached to the shoe box that would function display the status of the shoebox and allow the user to see the password that is being inputted. The plan is to have the first line on the LCD say if the shoe box is locked or unlocked. Then the second line will display the numbers and change once the user enters the lock. "0, 0 ,0" will be the default line so user's password isn't exposed to other people.

To have the ability to enter this password, the initial plan is to have it being accomplished by using a joystick. To enter a new password, a push button would be used. So once, that push button is clicked then you will be allowed to input the password.

Furthermore, there will be another push button that will allow the user to set a new password. These buttons will be labeled so it makes it more convenient for the user to remember which push button does what. Another option we want to implement as well is to have a fingerprint reader to make it even more easier for the user to access the locked box. The last sensor, that we plan to implement is to have a potentiometer. The purpose of the potentiometer would be to control the LCD screen's brightness. So for example, when the user is inputting his password he can dim the brightness. That way it gives him some privacy.

Furthermore, we plan to have LEDs to let the user know if the password entered was correct or wrong. Green will symbolize that password was entered correctly and red will let the user know that they have to re input the password. We also will require the need of servo motor so that the safe will unlock once the correct password is entered.

The initial plan is to have a timer so that it prevents the user from re-entering the password right away and gaining access back to the device. The user can either unlock the box through a 4-5 digit pin or using the fingerprint sensor.

3. **Expected Plan for Use and Communication between the multiple Arduinos**

- ❖ First Arduino
    - ➢ The first arduino will be to set a lock on the box. This arduino will be connected to the metal box that is being used as a safe. This arduino will communicate with the 2nd and 3rd arduino in order to be locked. The 2nd arduino will include the process of setting a timer and the 3rd arduino will help the user set a password. Once a timer and password is set, this arduino will set the lock on. This arduino will also be set up to use it with the leds and potentiometer.
- ❖ Second Arduino
    - ➢ The second arduino will help in setting a timer for the duration of the box being locked. Prior to setting a lock on the box, the user will have the option of how long they want the box locked for. The lock will be then set to a timer and will be available to be unlocked after the time finishes. The user can also have the option of having the box locked for no specific time and can be unlocked when they wish via password unlock.
- ❖ Third Arduino
    - ➢ The third arduino will assist in setting a password that is needed to unlock the box. After the timer is finished, the box is ready to be opened. The 2 ways to open the lock box will be either via fingerprint recognition or a 4-6 digit pin. We will have a fingerprint scanner attached in which the user can set their fingerprint prior to locking the box in order to use this feature. We will also have a 0-9 pin pad to unlock the locked box as well. Similar to the fingerprint scanner, the pin pad can also be set up prior to locking the box. This arduino will also be set up to use it with the joycons.

        - ❖ Fourth Arduino
            - ➢ Finally, the fourth arduino will be used to unlock and open the box. The box will only be able to unlock when it meets the time wait requirement and is unlocked with one of the 2 password options. The box will then open and can be used again at the user's discretion. This arduino will also be set up to use it with the push buttons.

- REQUIRED Supporting Materials
    - o **Timeline of Development done so far (week-by-week format)**
        - Week 08- Research what parts work best to complete project and order them. Finalize schedule within group to work on the project.
        - Week 09- Research what parts work best to complete project and order them
        - Week 10- Install LCD on shoebox. Start wiring arduino
        - Week 11- Complete the coding portion of the project
    - **Timeline of Development to complete (week-by-week format)**
        - Week 12- Figure out the password system using both a joystick and fingerprint scanner
        - Week 13- Do various testing to make sure the final product is working as intended and practice presentation as a group
        - Week 14- Presentation

o UPDATED **List of Materials Expected to be Needed**

- Jumper Wires
- Fingerprint reader
- Senso motor

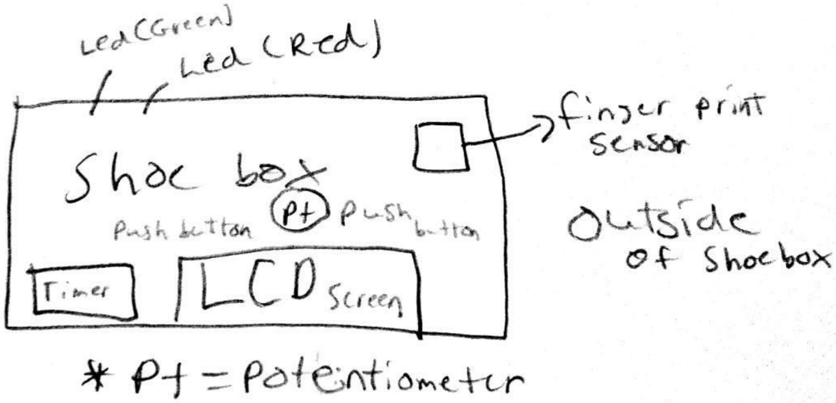
o UPDATED List of References

- https://www.hackster.io/user885477/an-arduino-based-diy-safe-22299b
- https://create.arduino.cc/projecthub/chummer1010/electronic-safe-with-arduino-25d039
- https://www.hackster.io/nickthegreek82/arduino-fingerprint-sensor-tutorial-103bb4
- https://www.youtube.com/
- https://www.arduino.cc
- https://maker.pro/arduino/projects/how-to-make-your-own-fingerprint-scanner-with-arduino-uno

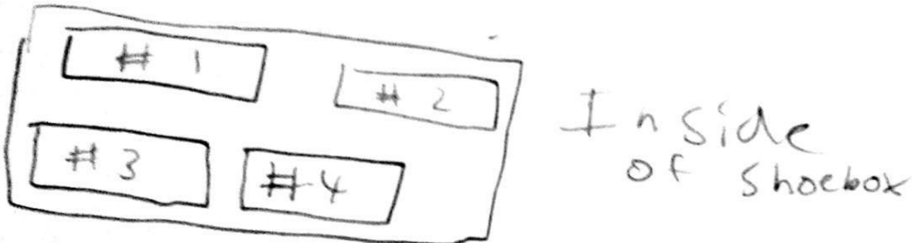
o Inclusion of one or more sketches to help identify your project is REQUIRED!

High Level Diagram of the Project as a whole

Sketch



* Pt = potentiometer



Arduinos will be pinned to the side walls

#3 -> wired with finger print scanner and joycons

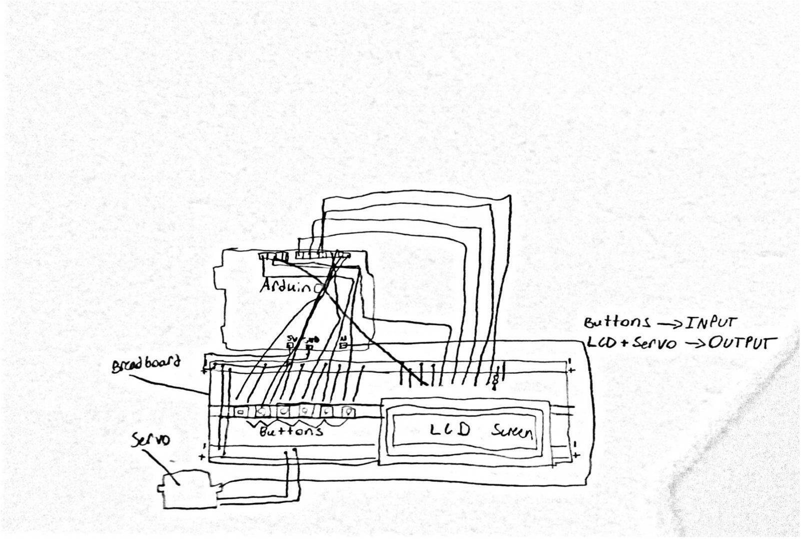
#1 -> wired with lock to open the secured box and Led/potentiometer

#4 -> wired with the lock itself and push buttons

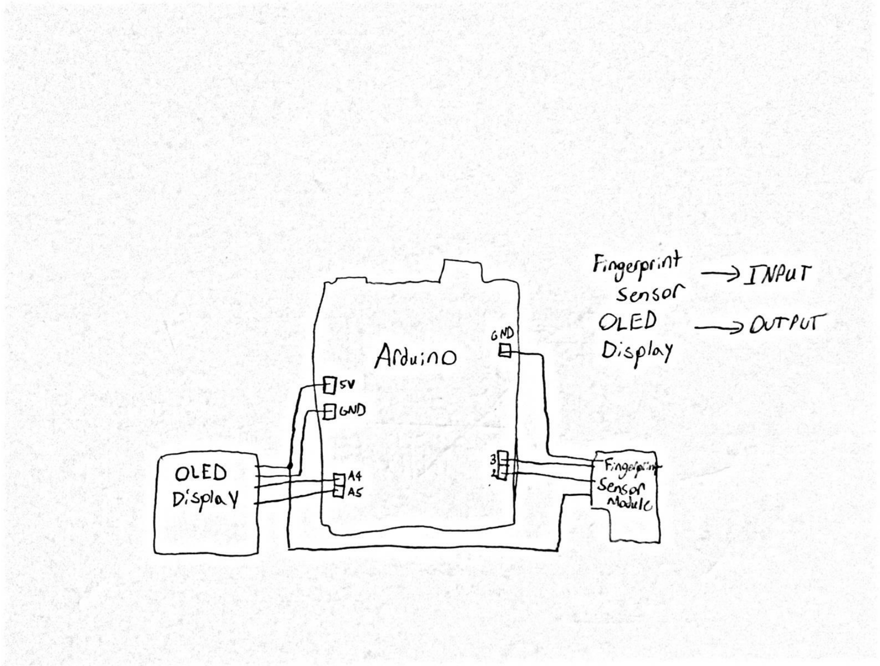
#2 -> wired with timer

Lower level diagram showing what is connected to each Arduino. (One diagram per an arduino)

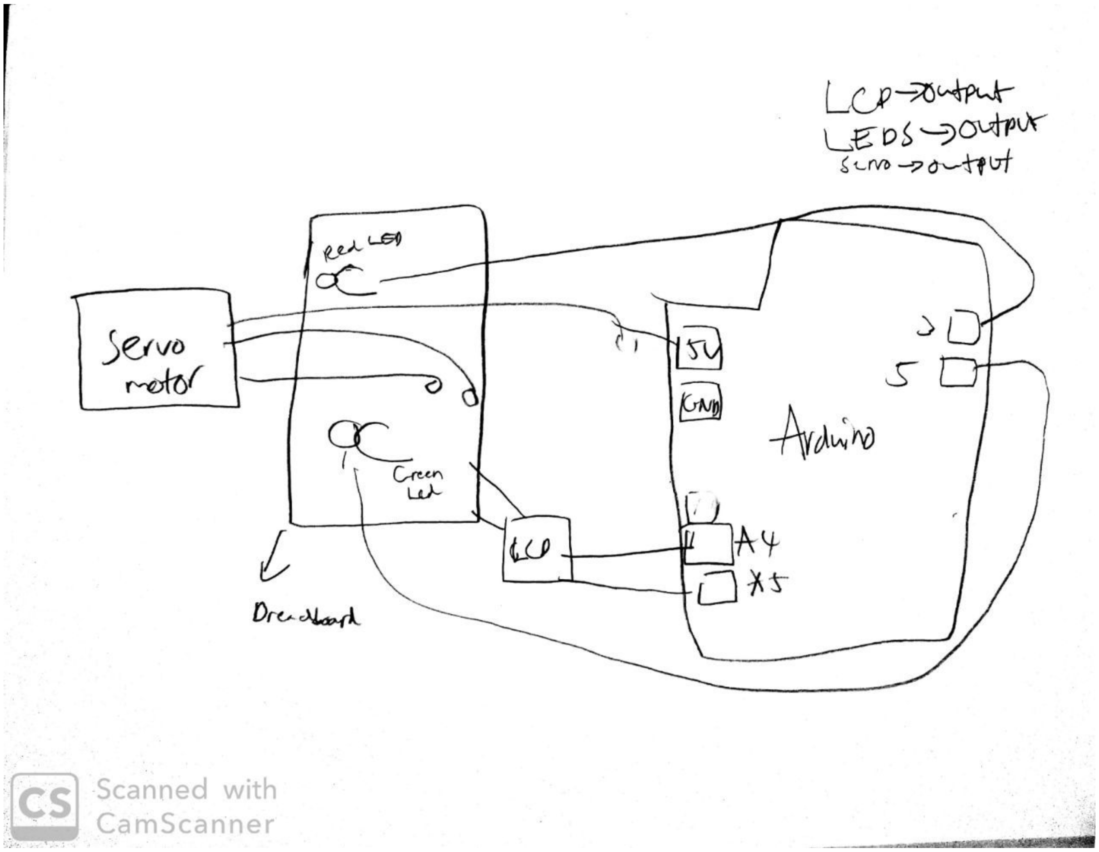
Arduino #1 (Attaching the lock to the safe):



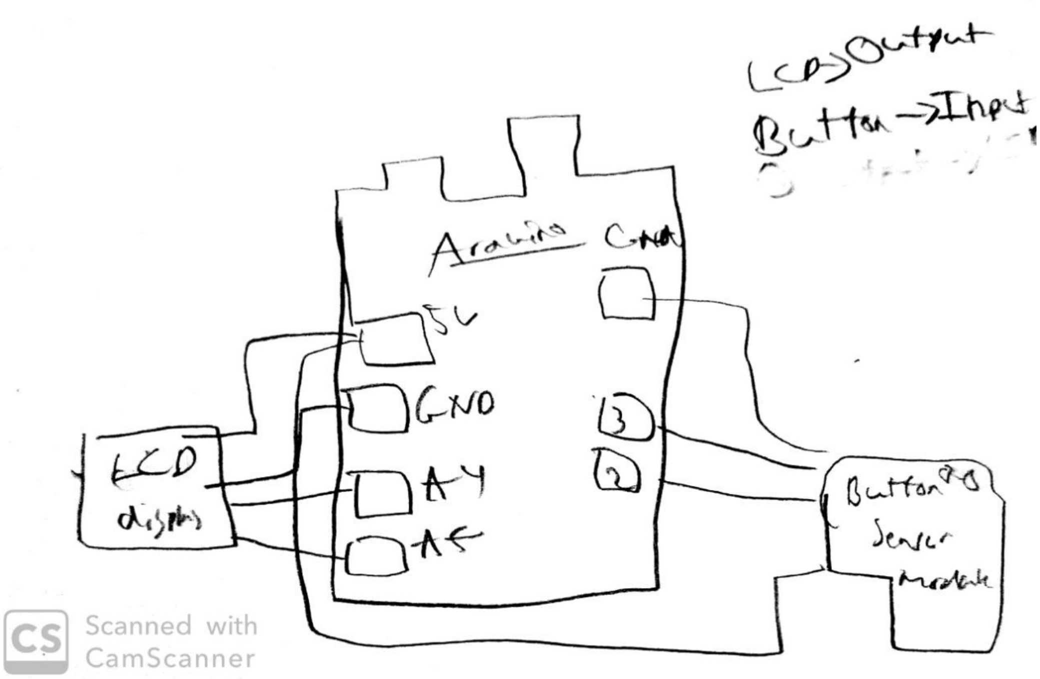
Arduino #2 (Adding the fingerprint sensor):



Arduino #3 (Attaching the LEDS):



Arduino #4 (To add the timer):



Code:

```
//Still in developmental stage
#include <LiquidCrystal.h>
#include <EEPROM.h>
#include <Servo.h>

int address = 0;
static unsigned long SaveTimer;
static unsigned long SaveDelay = (30 * 1000);


char CODE[10] = "1234E";
char Str[10];
char CodeLength = 4;
int Pos = 0;
bool Unlocked;
static unsigned long DisplayTimer;
static unsigned long DisplayDelay = 200;

LiquidCrystal lcd(12, 11, 9, 8, 7, 6);


int buttonPin1 = 2;
int buttonPin2 = 3;
int buttonPin3 = 4;
int buttonPin4 = 5;

int enterbutton = 10;
int clearlockbutton = 13;

Servo myServo;

void setup() {

  myServo.attach(A1);

  int EEPROMCodeOK = true;
  for (Pos = 0; Pos <= (CodeLength); Pos++) {
    Str[Pos] =  EEPROM.read(Pos);
    if (!(strrchr("1123456789", Str[Pos]))) {
      // not a valid code
      EEPROMCodeOK = false;
    }
  }
  Pos++;
  Str[Pos] =  EEPROM.read(Pos);
  if (Str[CodeLength + 1] != 'E') EEPROMCodeOK = false;
  if (EEPROMCodeOK) {
    Str[CodeLength + 2] = '\0';
    strncpy(CODE, Str, CodeLength + 1);
  }
  ClearCode();

  pinMode(buttonPin1, INPUT_PULLUP);
  pinMode(buttonPin2, INPUT_PULLUP);
  pinMode(buttonPin3, INPUT_PULLUP);
  pinMode(buttonPin4, INPUT_PULLUP);

  pinMode(enterbutton, INPUT_PULLUP);
  pinMode(clearlockbutton, INPUT_PULLUP);

  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("Hello.");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Password:");

  DisplayTimer = millis() + 200;
}


void loop() {

  Lock();
```

```
Pos = constrain(Pos, 0, CodeLength);

int buttonState1 = digitalRead(buttonPin1);
int buttonState2 = digitalRead(buttonPin2);
int buttonState3 = digitalRead(buttonPin3);
int buttonState4 = digitalRead(buttonPin4);

int clButtonState = digitalRead(clearlockbutton);
int enterButtonState = digitalRead(enterbutton);

lcd.setCursor(9, 0);

if (buttonState1 == LOW) {
  Str[Pos] = '1';
  Pos++;
  Str[Pos] = '\0';
  delay(250);
  while (digitalRead(buttonPin1) == LOW);

}

else if (buttonState2 == LOW) {
  Str[Pos] = '2';
  Pos++;
  Str[Pos] = '\0';
  delay(250);
  while (digitalRead(buttonPin2) == LOW);

}

else if (buttonState3 == LOW) {
  Str[Pos] = '3';
  Pos++;
  Str[Pos] = '\0';
  delay(250);
  while (digitalRead(buttonPin3) == LOW);
}
```

```
else if (buttonState4 == LOW) {
  Str[Pos] = '4';
  Pos++;
  Str[Pos] = '\0';
  delay(250);
  while (digitalRead(buttonPin4) == LOW);

}
else if (enterButtonState == LOW) {
  Str[Pos] = 'E';
  Pos++;
  Str[Pos] = '\0';
  delay(250);
  while (digitalRead(buttonPin1) == LOW);
   if (strcmp (Str,CODE) == 0) {
    Unlocked = true;
    lcd.setCursor(0, 0);
    lcd.print(" Access Granted");
    delay(2000);
    lcd.clear();
    lcd.print("    Unlocked");
  }
  else if (SaveTimer > millis() && (Pos + 1) == CodeLength) {

    strcpy(CODE, Str);
    for (Pos = 0; Pos <= (CodeLength + 1); Pos++) {
      EEPROM.write(Pos, Str[Pos]);
    }
    lcd.setCursor(0, 0);
    lcd.print("Saving Code:");
    lcd.setCursor(0, 1);
    lcd.print(Str);

    Unlocked = true;
  }

  else {

    lcd.clear();
    lcd.print(" Access Denied.");
    delay(2000);
    lcd.clear();
    lcd.print("Password:");

  }

  while (Unlocked) {
    Unlock();
    if (digitalRead(clearlockbutton) == LOW) {
      delay(200);
      lcd.clear();
      lcd.print("    Locked");
      delay(2000);
      lcd.clear();
      Unlocked = false;
      SaveTimer = millis() + 30000;
    }
  }

  ClearCode();


}

else if (clButtonState == LOW) {
  delay(500);

  while (clearlockbutton == LOW);
  if ((millis() - SaveTimer) > 4500) {

  }

  ClearCode();

}

if ( (long)( millis() - DisplayTimer ) >= 0) {
  DisplayTimer += DisplayDelay;
```

```
    lcd.setCursor(9, 0);
    lcd.print(Str);
    lcd.print("     ");

  }
}

void ClearCode() {

  Pos = 0;
  Str[Pos] = '\0';
  lcd.setCursor(0, 0);
  lcd.print("Password:");
  lcd.setCursor(0, 1);
  lcd.print("          ");

}


//TODO
void fingerPrintSensor() {

}

//TODO
void communicateWith() {

}
void Unlock() {

  myServo.write(150);

}

void Lock() {

  myServo.write(50);

}
```

Description of the original work being attempted by your project.

- Integrating a timer with the safe that unlocks the safe automatically when the time is done. This will ensure that the user can't access the device before the timer ends and thus they stay focused on their task at hand. Also also allow the user to have the ability to unlock the safe using a fingerprint scanner. This makes it more convenient for the user and they don't have to remember an unique 4-6 digit pin. We hope our original modifications help make this better for users and allow them to be more focused when needing to get important tasks done.