Milestone 6 – 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 a specific time, they can place their phone into the container, and they will not have access to their phone until it allows you to re-enter a password.

• **Detailed Project Ideas**

o 3 Required Sections

1. **Overall Description of Project Idea**

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

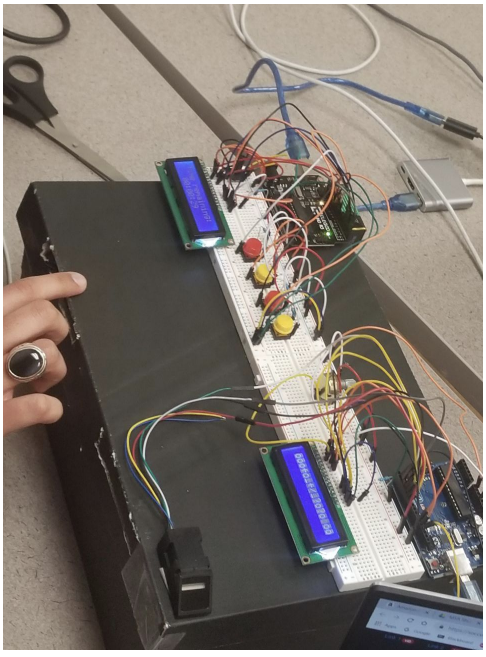
Our project is to implement a safe-like device that is designed to keep the user away from wasting their time on certain tasks (e.g. phone usage, etc). The safe will use some security features (password, fingerprints, etc) as well as a timer which will make the safe unable to open until the timer is complete. Instead of

just the basic one lock, we had two locks. Thus, it acts like 2 factor verification to accesses the box. We had a shoebox to act as as the safe.

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**

  Since there were 4 group members, we used 4 arduinos for this project. Each arduino played a critical role in ensuring our final product worked correctly. We had one arduino with the 4 buttons and a LCD display. The purpose of this arduino was to allow the user to enter the 4 digit pin. The LCD display was to make it more user friendly. It would display to enter the pin and show it as well on the screen. Furthermore, this is where the timer would display and start running if the correct 4 digit pin was entered. This communicated with the second arduino which was the servo motor for one half of the shoebox. The servo motor was crucial to make sure the box was locked. The third arduino was an LCD display and fingerprint reader. The fingerprint reader was just another extra measure of security. The LCD display would prompt user to scan their finger and then confirm confirmation when it was successful. This worked with the 4th arduino which was the second servo motor. You had to get both forms of security measures correct to unlock the whole box.



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

In our final design, we used a shoebox to act as the safe and the location of where the locked valuables will be kept. We had two LCDs on the cover of the shoebox. One LCD was attached on the left side and had the buttons attached to it. Thus, this would be the display for the 4 digit pin portion. This LCD is where we prompted to enter the 4 digit pin and it also displayed the timer. It would say "Time remaining" on the first line and the second line would display the time left in the format "00:00:00" to reflect how long the user has to wait to reenter the password. This LCD screen was attached with the 4 buttons. The buttons would allow the user to enter the 4 digit pin. The button order reflected to the number order so for example if you hit the second button it would make the number entered: 0100. Once the password is entered correctly, it would unlocks one side of the box but you would still need to use your fingerprint to unlock the second side of the box.

The second LCD on the right hand side was used for the fingerprint scanner. So it would prompt the user asking them to "Scan their finger". Once the user scans their finger and it matches it would say "Found match". At that point, the second lock will unlock and then you would have the ability to grab your device out of the box.

Inside the shoebox, we had the two servo motors. Each servo motor was attached to a different arduino. We had one servo motor communicate with the button/LCD arduino and then the second servo motor would communicate with the fingerprint scanner/LCD. That would be a total of 9 external devices. The four buttons and fingerprint scanner would be the input devices. The output devices would be the 2 LCD screens and 2 servo motors.

We implemented so that there is a timer so that it prevents the user from re-entering the password right away and gaining access back to the device. The user has to unlock the box by entering the 4 digit pin correctly and having the correct fingerprint.

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

So for communication we decided to do it where we have two arduinos communicating together at once. Originally, we wanted all 4 arduinos to communicate with each other, but unfortunately ran into issues with that.

- First Arduino and Second Arduino
  - The first arduino had the four buttons and the LCD screen. This was how the user would input the 4 digit pin. The second arduino was one of the servo motors. We communicated between these two arduinos using serial communication. So we use pin 0 and pin

1 on each arduino to communicate. Pin 0 is the Rx,so the receiver and Pin 1 is the Tx, so the sender. Thus, the buttons were telling the servo motor to unlock once the password was entered correctly. In the code portion, we used Serial.Write and Serial.Available to communicate. Serial.Write writes the binary data to the serial port. Serial.Available gets the number of bytes available for reading from the serial port.

- ❖ Third and Fourth Arduino
  - ➢ The third arduino was the fingerprint scanner. For this, the user would just scan their finger and then the (fourth arduino) second servo motor would unlock. We used the same form of communication as stated above for the four buttons and the first servo motors.

- REQUIRED Supporting Materials
  - o **Final Timeline of Development (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- Start making each individual arduino
    - Week 10- Start attaching arduinos to shoebox and testing them
    - Week 11- Complete the coding portion of the project
    - Week 12- Figure out how to implement communication between each pair of arduinos
    - Week 13- Do various testing to make sure the final product is working as intended and practice presentation as a group
    - Week 14- Presentation
    - Week 15 - Demonstration

o List of Materials that were Needed

- Jumper Wires
- Fingerprint reader
- 2 Senso motor
- Resistors
- 4 Arduinos
- LEDs for testing
- Box
- Scissors

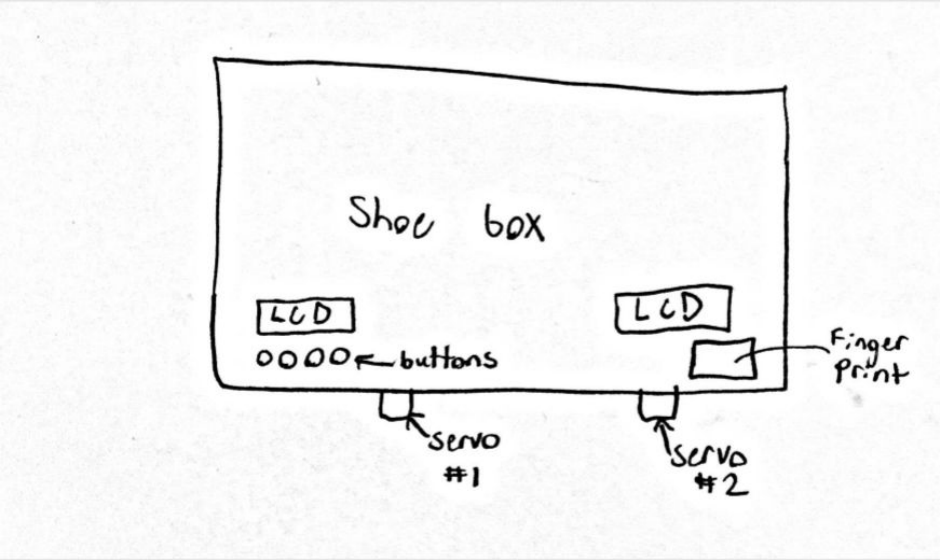
o List of References

- 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
- https://www.instructables.com/id/fingerprint-arduino-with-16x2-LCD/

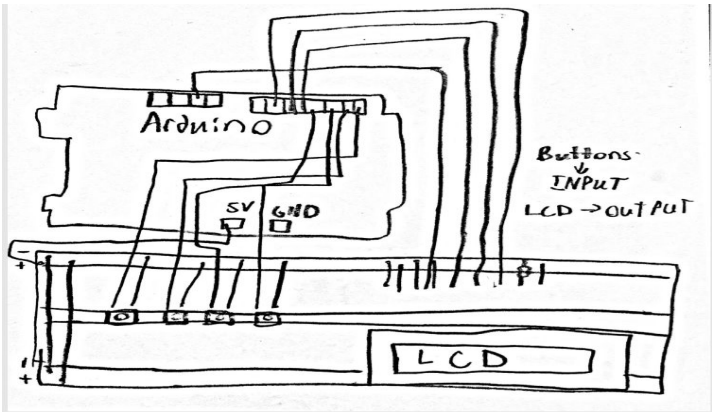
o Inclusion of one or more sketches to help identify your project is REQUIRED!

High Level Diagram of the Project as a whole

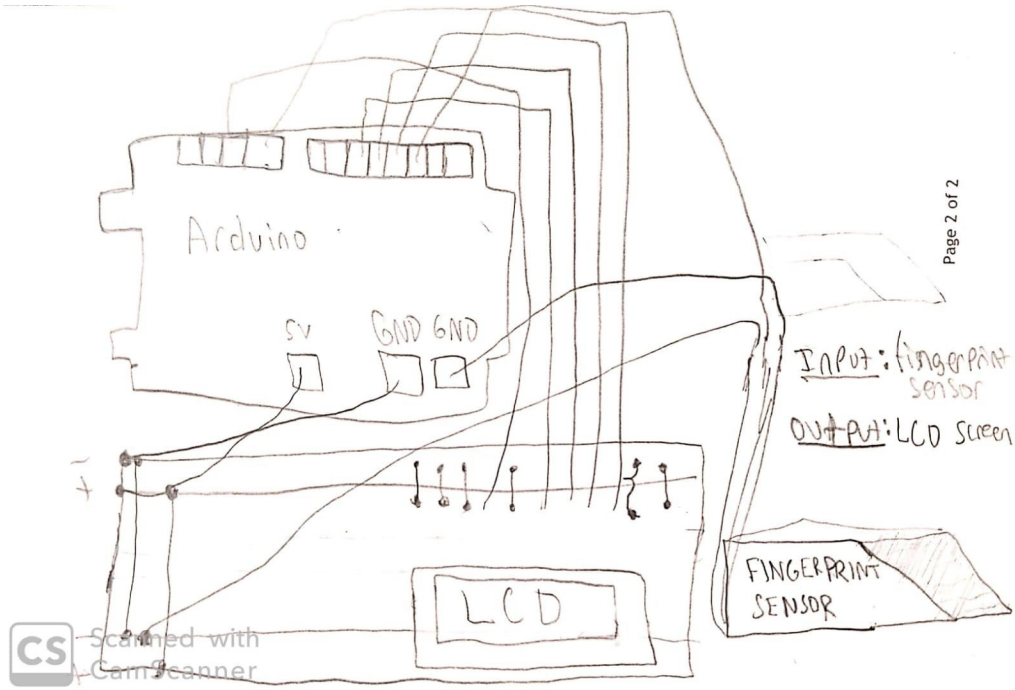


Lower level diagram showing what is connected to each Arduino. (One diagram per an arduino)

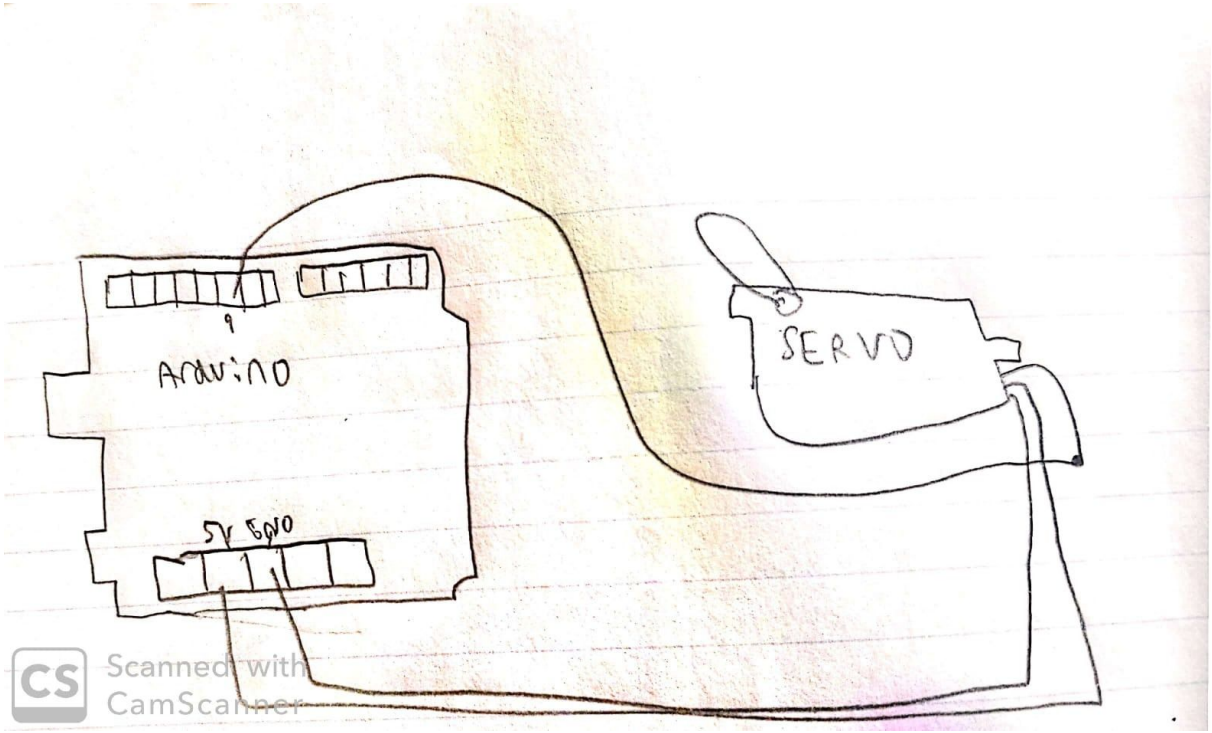
Arduino #1 (Entering code on LCD screen):



Arduino #2 (Fingerprint Sensor + LCD display):



Arduino #3 and #4 (Both Servo motors):



Description of the original work being attempted by your project.

- We implemented a two factor authentication by adding two servo motors and thus requiring 2 forms of security for the user. First one, was the 4 digit pin. The second original work was implementing a fingerprint scanner. It was a thing we purchased separately and its purpose to ensure the user who locked the box is the one who is actually unlocking it.. Also we integrated 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. Thus, this makes it more unique than a just a regular safe. 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.

Final Code(Includes 3 files):

Password.ino file

```
//Group #60
//Name: Maaz Ahmed, netID: mahmed76
//Name: Hasan Ali, netID: hali32
//Name: Ammar Idrees, netID: aidree3
//Name: Umer Qazi, netID: uqazi2
//Project name: Addiction Prevention Safe
//Abstract: 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 a specific time, they can place their phone into the container,
//and they will not be able to use their phone until it allows you to re-enter a password.

/*The code here utilizes four buttons to enter a 4 digit pin that will be displayed on an LCD
 * screen. To set a password, it must be hardcoded into this program (currently 1 2 3 4). Note
 * that the wiring is a bit fragile and can easily make the LCD unreadable, if that is the case,
 * resetting the arduino should remedy the issue.
 */
//Using LCD library
#include <LiquidCrystal.h>

//Buttons will be used to enter 4 digit code
int buttons[] = {8, 9, 10, 11};
//They will be all 0s by default
int code[] = {0, 0, 0, 0};
//Correct code is here
int correctCode[] = {1, 4, 1, 8};
//Set up LCD and create flag that will determince if correct code is entered
LiquidCrystal lcd(13, 12, 5, 4, 3, 2);
```

```
bool correct = false;
bool isLocked = false;
int timer = 30;

void setup() {
  //All buttons will be input
  for (int i = 0; i < 4; ++i) {
    pinMode(buttons[i], INPUT);
  }

  //Set up LCD, and prompt user to enter password
  lcd.begin(16, 2);
  lcd.print("Hello.");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Enter Password");
  delay(2000);

  Serial.begin(9600);
}

void enterPassword() {
  while (true) {
    //Clear screen, and have an array to keep track of button states
    lcd.clear();
    int buttonStates[4];

    lcd.setCursor(0,0);

    /*Print each digit on the screen, everytime a button is pressed, increment by 1. In the
     * case that the digit gets to 9, pressing again will reset it to 0
     */
    for (int i = 0; i < 4; ++i) {
      Serial.print(code[i]);
      lcd.print(code[i]);
      buttonStates[i] = digitalRead(buttons[i]);
      if (buttonStates[i] == HIGH) {
        code[i] += 1;
```

```
      if (code[i] > 9) {
        code[i] = 0;
      }
    }
  }

  //Continuosly check if code is correct, set bool flag accordingly
  for (int i = 0; i < 4; ++i) {
    if (code[i] != correctCode[i]) {
      correct = false;
      break;
    }
    correct = true;
  }

  /*If the bool flag states true, then display to user that the password is correct, and
   * communicate to another arduino to lock/unlock. After that, reset code to all 0s
   */
  if (correct) {
    Serial.println("Password Correct");
    lcd.clear();
    lcd.print("Password Correct");
    Serial.write('A');
    delay(1000);
    for (int i = 0; i < 4; ++i) {
      code[i] = 0;
    }
    break;
  }
  delay(250);
 }
}

void loop() {
 //If the safe is not locked, then enter password to lock
 if (!isLocked) {
  enterPassword();
  isLocked = true;
 }
```

```
  //Otherwise, wait until timer ends, then enter password unlock
  else {
    while (timer > 0) {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Time remaining: ");
      lcd.setCursor(3, 1);
      lcd.print("00:00:");
      lcd.print(timer);
      timer--;
      delay(1000);
    }
    isLocked = false;
    timer = 30;
    enterPassword();
  }

  delay(250);
}
```

Servo.ino file

```
//Two of the arduinos will be using this code. All it does is simply lock or unlock
//We will be using the Servo library
#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int lock = 0; //variable to determine if locked or not
int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
}

void loop() {
```

```
  //if arduino data was sent/arduino 2 can read arduino 1s data
  if (Serial.available() > 0) {
    //read what the serial.write data bytes are
    if (Serial.read()=='A') {
      if(lock==1){//locking
        myservo.write(50);
        lock=0;
      }
      else{//unlocking
        myservo.write(150);
        lock=1;
      }
    }
  }
}
```

FingerPrintLCD.ino file

```
//The majority of the code here was given by the Adafruit library. Our work is any of the code
that interacts with the LCD
//(The work we did will be commented on)
//Fingerprint library and LCD library
#include <Adafruit_Fingerprint.h>
#include <LiquidCrystal.h>

//Pins 2 and 3 are for the fingerprint. 12, 13, 4, 5, 6, and 7 are the LCD
SoftwareSerial mySerial(2, 3);
LiquidCrystal lcd(12, 13, 4, 5, 6, 7);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  //Set up screen and prompt user to scan their finger.
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("Scan finger");
```

```
  delay(1000);
  while (!Serial);  // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit finger detect test");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }

  finger.getTemplateCount();
  Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
  Serial.println("Waiting for valid finger...");
}

void loop()                     // run over and over again
{
 //Continuously prompt the user to scan their finger, and run the scanner to find an ID
 lcd.clear();
 lcd.print("Scan finger");
 getFingerprintIDez();
  delay(50);            //don't ned to run this at full speed.
}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
```

```
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
  Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
```

```
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  // found a match!

  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);

  //Once match is found, display to User and communicate to arduino to lock/unlock
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Found match!");
  Serial.write('A');
  delay(1000);

  return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
```

```
  //Same as above: display to user and communicate to lock/unlock
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Found match!");
  Serial.write('A');
  delay(1000);

  return finger.fingerID;
}
```