

DR. PILL

Automatic Pill Dispenser

MEET THE TEAM



Jonathan
Cerniaz

- Fingerprint and Facial Recognition
- Circuit Design



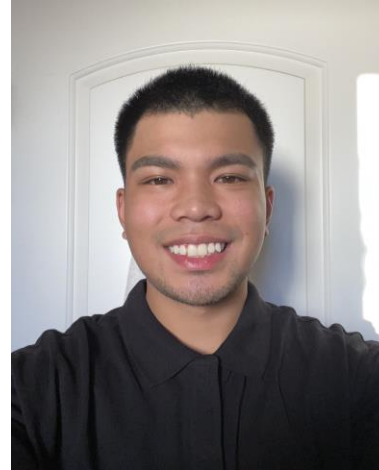
Jehmel
Espiritu

- Software Development of Interface



Jeremy
Espiritu

- Software Development of Interface



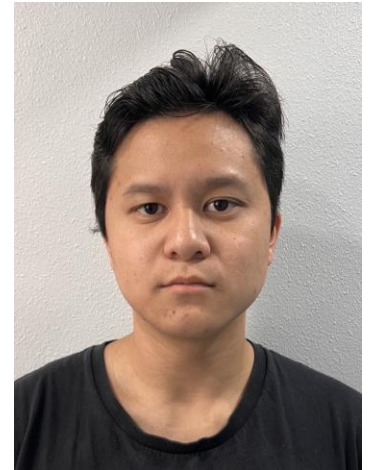
Joseph
Guzman

- Object Recognition
- Board to Board Communication
- 3D Design



Afzal
Hakim

- Fingerprint and Facial Recognition
- Log tracking and Security



Lee Roger
Ordinario

- Pill Storage System and Organization
- PCB Design and Soldering



Executive Summary

Dr. Pill is an automatic pill dispenser that simplifies medication management. Our device aims to facilitate the process by freeing up more time for doctors and allowing patients to take their medication accurately by eliminating the common issue of medication errors and missed dosages. Designated for pharmacies, nursing homes, personal households.





OUR 3 DEMOS

1. Dispense Mechanism & Storage
2. Displaying User Interface
3. Authentication and Board to Board Communication

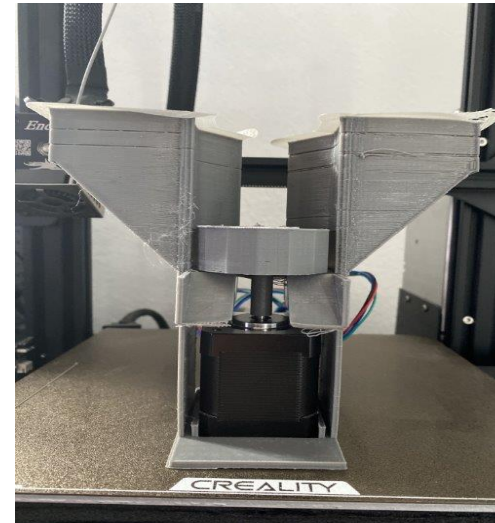
Final: Completed 'Dr. Pill'
The Automatic Pill Dispenser



PREVIOUS DEMONSTRATION

1. 3D Design (Dispense Mechanism & Storage)
(Computer <--> MobaXTerm <--> STM32/Motors)

2. Displaying User Interface
(RaspberryPi/LCD <--> MobaXTerm <--> STM32/Motors)



TODAY'S DEMO

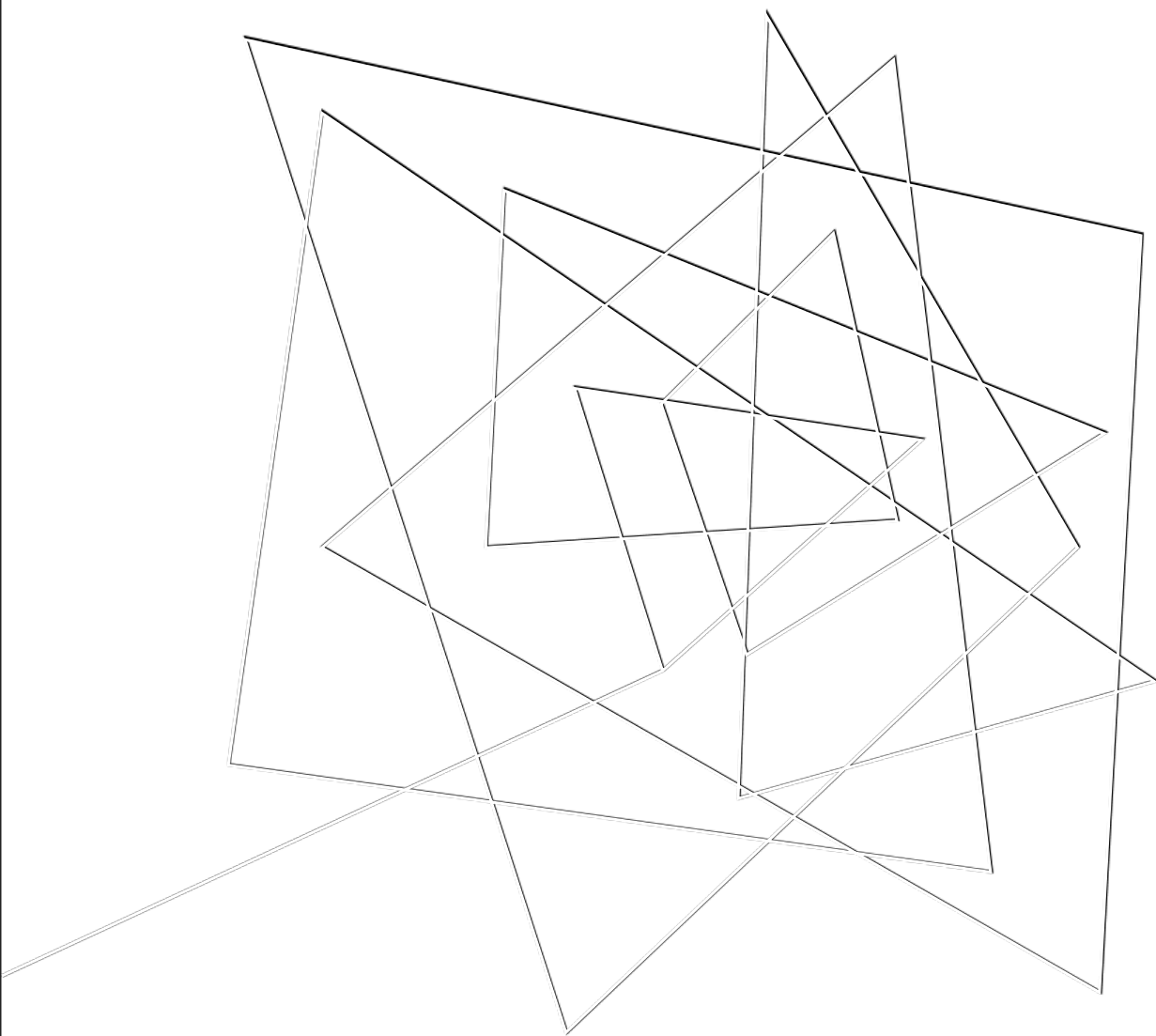
Authentication and Board to Board Communication

Requirements: Fingerprint and Facial Recognition working for specified users, object detecting different pills

Constraints: Authentication devices accuracy and Motor response from GUI input

Success: Fingerprint and facial recognition sensor successfully integrated with our GUI.





DEMO TIME

TECHNICAL DETAILS

```
import cv2
import os
from datetime import datetime
from picamera2 import Picamera2
import time
```

```
PERSON_NAME = "afzal"
```

```
def create_folder(name):
    dataset_folder = "dataset"
    if not os.path.exists(dataset_folder):
        os.makedirs(dataset_folder)

    person_folder = os.path.join(dataset_folder, name)
    if not os.path.exists(person_folder):
        os.makedirs(person_folder)
    return person_folder
```

```
def capture_photos(name):
    folder = create_folder(name)
```

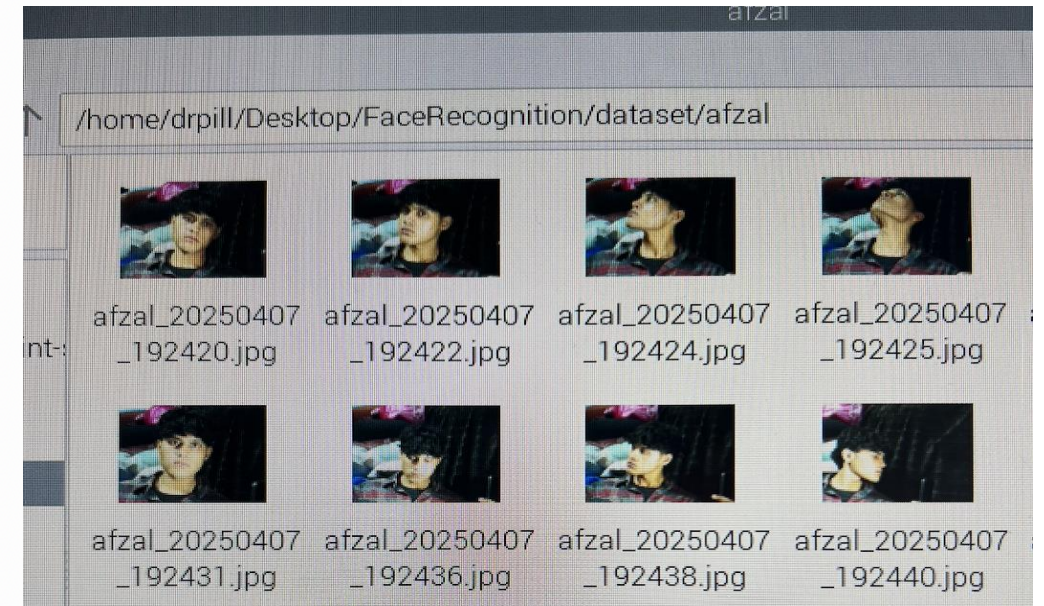
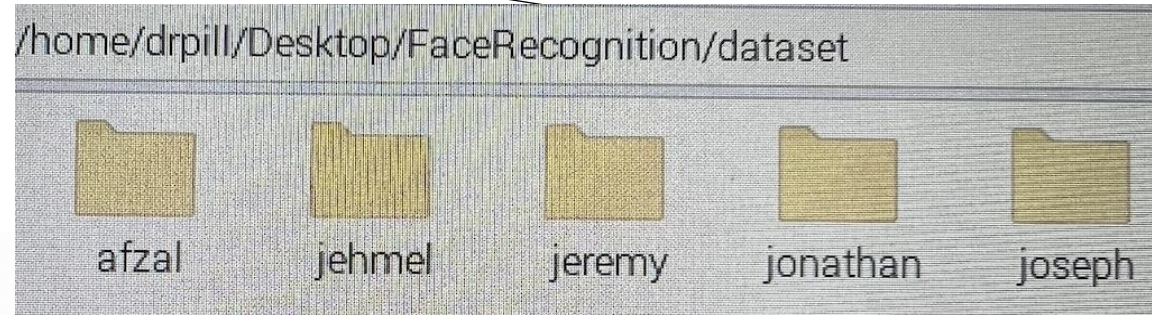
```
    # Initialize the camera
```

```
    picam2 = Picamera2()
    picam2.configure(picam2.create_preview_configuration(main={"format": 'XRGB8888', "size": (640, 480)}))
    picam2.start()
    photo_count = 0
```

```
    print(f"Taking photos of {name}. Press SPACE to take photo, press 'enter' to finish.")
```

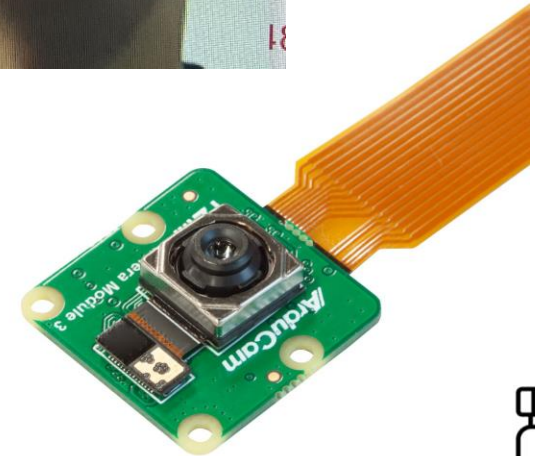
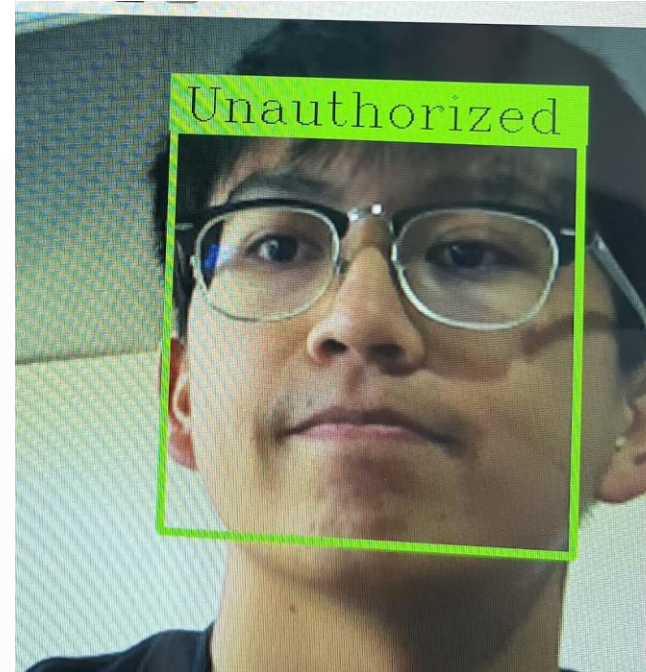
```
    while True:
```

```
        # Capture frame from Pi Camera
        frame = picam2.capture_array()
```



TECHNICAL DETAILS

```
def perform_biometrics(method):  
    hide_all_screens()  
  
    if method == "Facial Recognition":  
        facial_text.show()  
        start_time = time.time()  
        timeout = 5 # max 5 seconds to detect face  
  
        while True:  
            frame = picam2.capture_array()  
            processed_frame = process_frame(frame, selected_profile)  
            display_frame = draw_results(processed_frame)  
  
            cv2.imshow('Video', display_frame)  
  
            if any(name != "Unauthorized" for name in face_names):  
                recognized_name = next(name for name in face_names if name != "Unauthorized")  
                facial_text.hide()  
                info("Welcome", f"{recognized_name} recognized!")  
                switch_panel(selected_profile)  
                break  
  
            if time.time() - start_time > timeout:  
                facial_text.hide()  
                info("Error", "Face not recognized. Try again.")  
                request_auth_method(selected_profile)  
                break  
  
        cv2.waitKey(1)  
  
    cv2.destroyAllWindows()
```

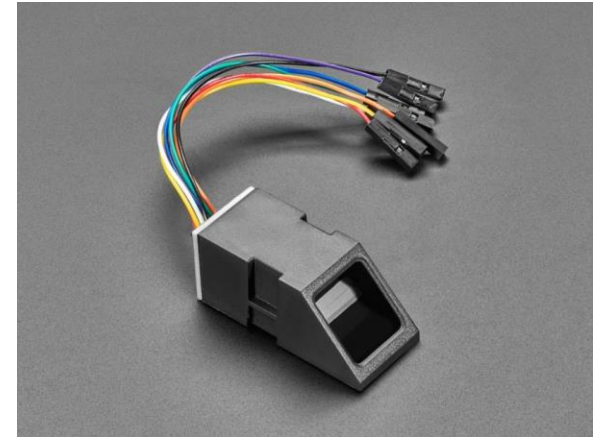


TECHNICAL DETAILS

```
import serial
import adafruit_fingerprint

# Setup the fingerprint sensor (connects via UART)
uart = serial.Serial("/dev/ttyAMA0", baudrate=57600, timeout=1)
finger = adafruit_fingerprint.Adafruit_Fingerprint(uart)

# Maps fingerprint IDs to roles
FINGERPRINT_USERS = {
    1: "Staff",
    2: "Admin",
    3: "Staff",
    4: "Patient",
    5: "Patient",
    6: "Patient"
}
```



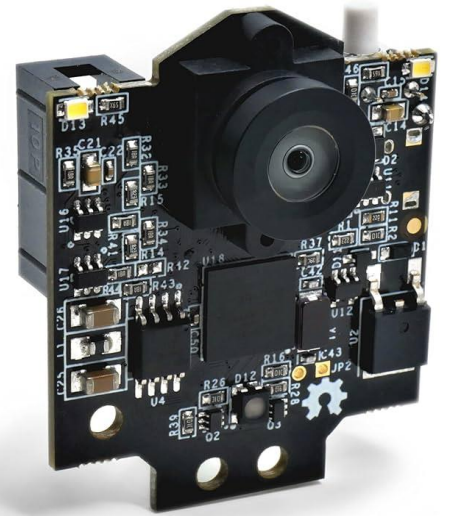
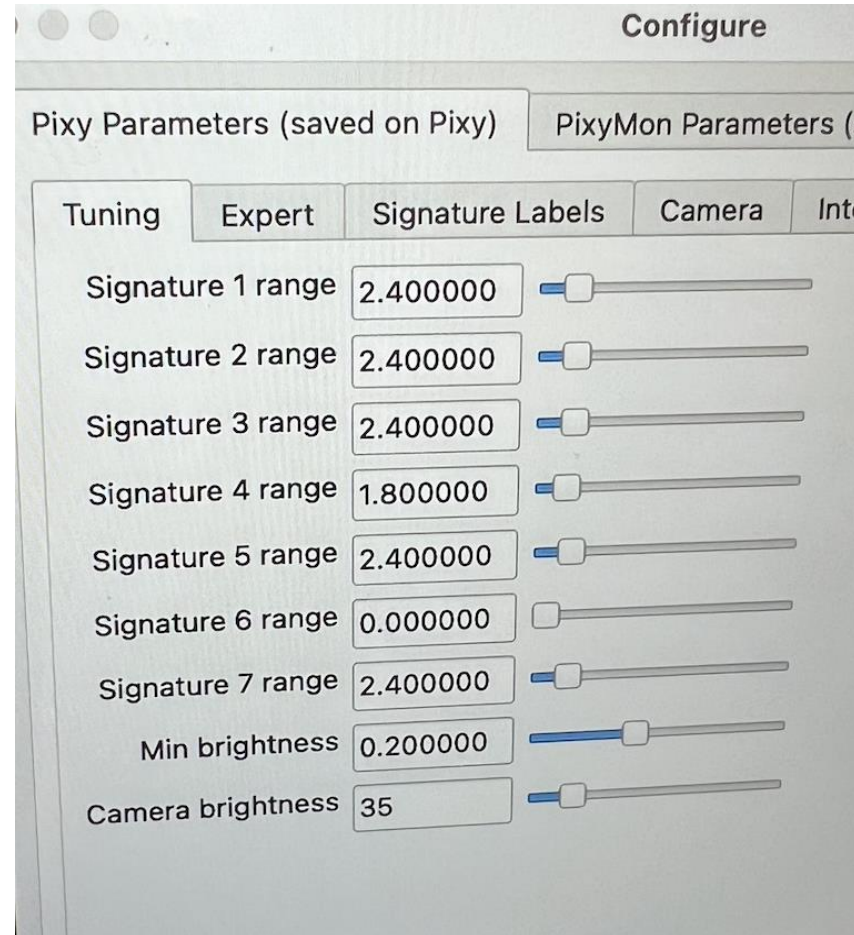
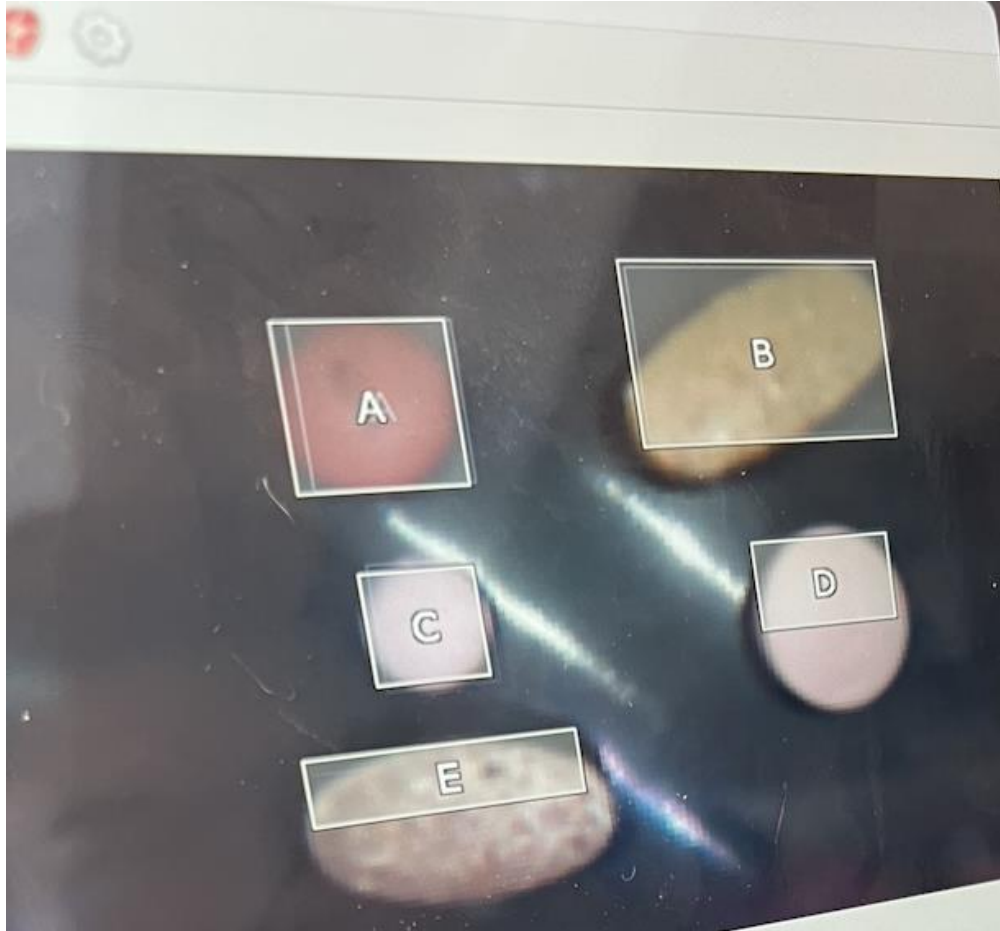
TECHNICAL DETAILS

```
# Function to try and scan a fingerprint
def get_fingerprint():
    if finger.get_image() != adafruit_fingerprint.OK:
        return False
    if finger.image_2_tz(1) != adafruit_fingerprint.OK:
        return False
    if finger.finger_search() != adafruit_fingerprint.OK:
        return False
    return True

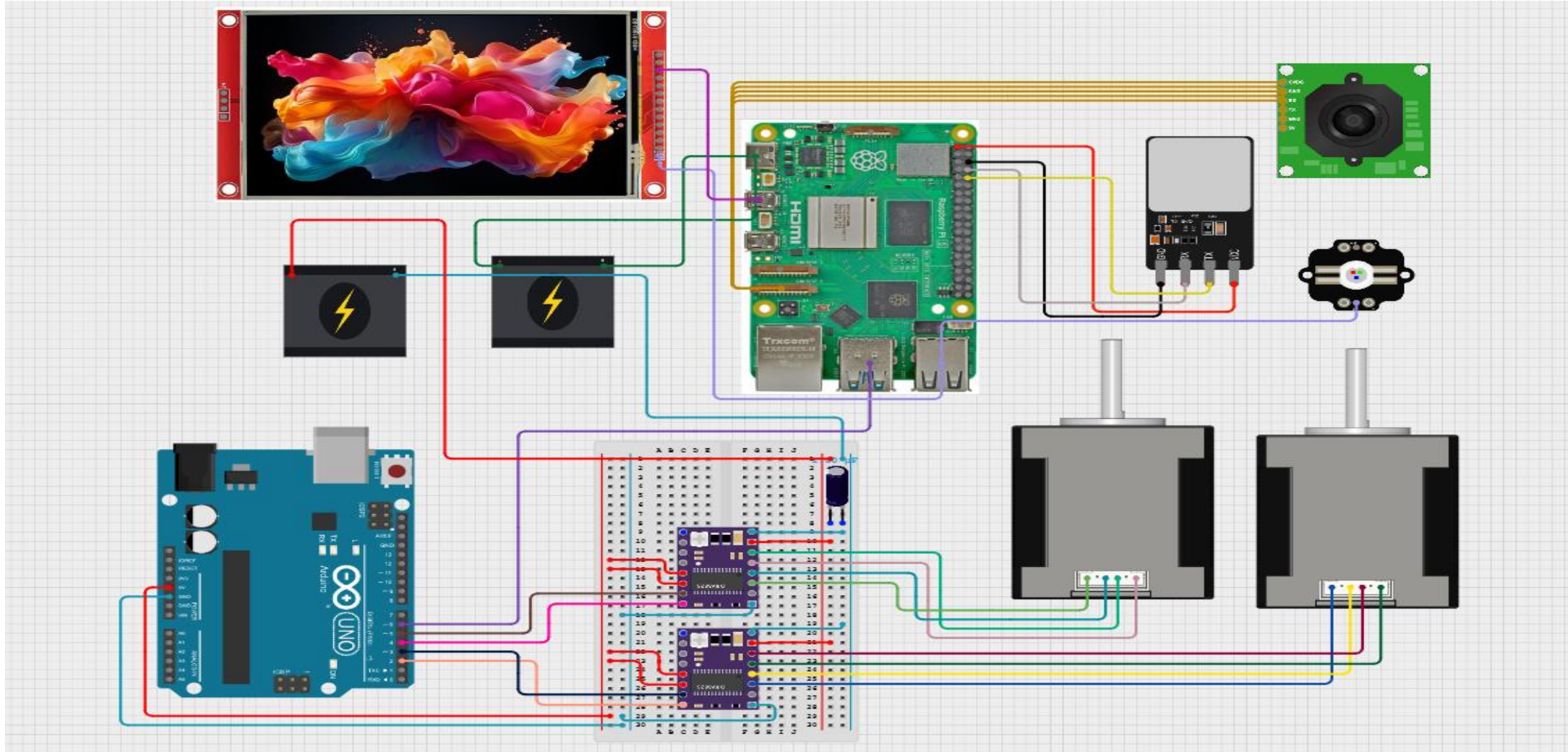
# Main fingerprint authentication logic
def fingerprint_auth():
    print("Place your finger on the scanner.")
    timeout = time.time() + 5 # Scan for 5 seconds
    while time.time() < timeout:
        if get_fingerprint():
            uid = finger.finger_id
            name = FINGERPRINT_USERS.get(uid, "unknown")
            print(f"Fingerprint matched! ID: {uid}, Role: {name}")
            return
    print("Fingerprint not recognized.")
```



TECHNICAL DETAILS



TECHNICAL DETAILS



CHALLENGES

- **Car got stolen** – This unexpected event caused us delays as our fingerprint and facial recognition device was in the car that got stolen.
- **New fingerprint and facial recognition devices** – We switched to a new biometric device which required us to do more research and understand how to code and integrate it properly.
- **Challenging to code** – Writing the software to integrate all hardware components, especially the new security features was complex and time consuming.
- **Pixy2 object recognition issue** – The Pixy2 sensor detects pills based on color, so all white pills are being recognized as the same which made it hard for us to differentiate between 2 white colored pills. Lighting inconsistency obscured the camera's ability to recognize pills.
- **Board-to-board communication difficulties** – After setting up the fingerprint and facial recognition devices separately, making them work together across different boards caused integration issues.

A California Vehicle Report form for a stolen vehicle. The form is titled 'STOLEN VEHICLE' and contains fields for reporting agency, location, date, time, and details of the vehicle. It includes a section for 'REASON FOR STOP' with checkboxes for various conditions like 'UNLAWFUL STOP', 'VEHICLE VIOLATION', etc. There is also a section for 'STOLEN VEHICLE / COMPONENT' and 'EMBEZZLED VEHICLE'. The form is filled out with handwritten information, including the name 'HAKIM ATAL JAMIL' and the vehicle details '2014 HYUNDAI SANTA FE'. The bottom of the form has a section for 'REMARKS' and a signature line.

OVERCOMING CHALLENGES

- We ended up using camera module 2 and adafruit fingerprint scanner thats connected by strip cable, and fingerprint scanner by pins and uart, since the previous device is gone
- Trial and error with code and testing a number of commands in command prompt to identify the correct port for CDC setup for board to board communication
- The use of different colored pills to avoid the issue of inaccuracy in identifying pills. The Pixy2 camera has a built in light that we used for the main source of lighting



FINAL DEMO

The complete device 'Dr. Pill' The Automatic Pill Dispenser

- **Log Tracking** – The system keeps a record of user interactions, pill dispensing events and authentication activity.
- **Fully Completed GUI** – A user friendly and responsive interface is ready making it easy for all users to navigate the system.
- **Pill Dispensing with Object Recognition** – Pills are dispensed automatically with the Pixy2 sensor confirming both the number and type of pills dispensed.

All components (GUI, fingerprint, face recognition, motors, pill sensor) working together seamlessly without manual intervention.





THANK YOU

