

Smart Door Security System: Installation and Execution

1. Installation Steps

1.1 Environment Setup

1. Install Python 3.x

Download and install Python 3.x from the official website if it's not already installed.

2. Install Required Libraries

Run the following command in the terminal to install the necessary Python libraries:

3. `pip install flask RPi.GPIO requests opencv-python-headless`
 4. install Cmake From official website and OpenCv Libraries and also install dlib libraries by the command
 - *`pip install opencv-python`*
 - *`Pip install flask`*
 - *`Pip install dlib`*
 - *`Pip install Face_recognition`*
-

1.2 Hardware Setup

1. Camera Module

- Connect the Camera Module to the USB port on the PC.
- Enable the camera in the Module.

2. Keypad

- Connect the keypad using GPIO pins
- Use the specified row and column pins for proper configuration.

3. Stepper Motor

- Connect the stepper motor to GPIO pins for door control.
-

1.3 Project File Structure

smart-door-security-system

├─ app.py

├─ env setup

| └─ Feed Trained image

├─ captured_images of Unknown

| └─ Unknown.jpg # (captured visitor images)

2. Execution Steps

2.1 Start the Application

1. Open the terminal and navigate to the project directory:

```
cd /smart-door-security-system
```

2. Run the Flask application:

```
python app.py
```

3. Access the Web Interface

```
http://localhost:5000
```

3. System Usage

1. Detect Visitors

- 1.1 When the keypad is activated or the face is detected:

- The system captures a photo of the visitor using the camera module.
- The captured image is saved in the captured_images/unknown folder and displayed on the web interface.

- 1.2 Use the keypad to enter the password if the face recognition fails.

2. Door Control

- The stepper motor rotates to open the door upon successful authentication.
-

4. Error Handling

1. Button Press Not Detected

- Check that the button is correctly connected to GPIO pin 18 and ensure proper resistor configuration.

2. Camera Issues

- Verify the camera is securely attached and the interface is enabled in Raspberry Pi settings.

3. Web Interface Not Loading

- Confirm the Flask app is running and check for any errors in the terminal logs.

4. Keypad Issues

- Check the keypad connections and ensure proper GPIO pin configuration.

5. Key Components

1. Keypad Configuration

- The keypad uses GPIO pins with the following row and column configuration:

```
ROW_PINS = [22, 18, 2, 3]
COL_PINS = [8, 10, 9, 11]
```

2. Stepper Motor Control

- The stepper motor is controlled using the following GPIO pins:

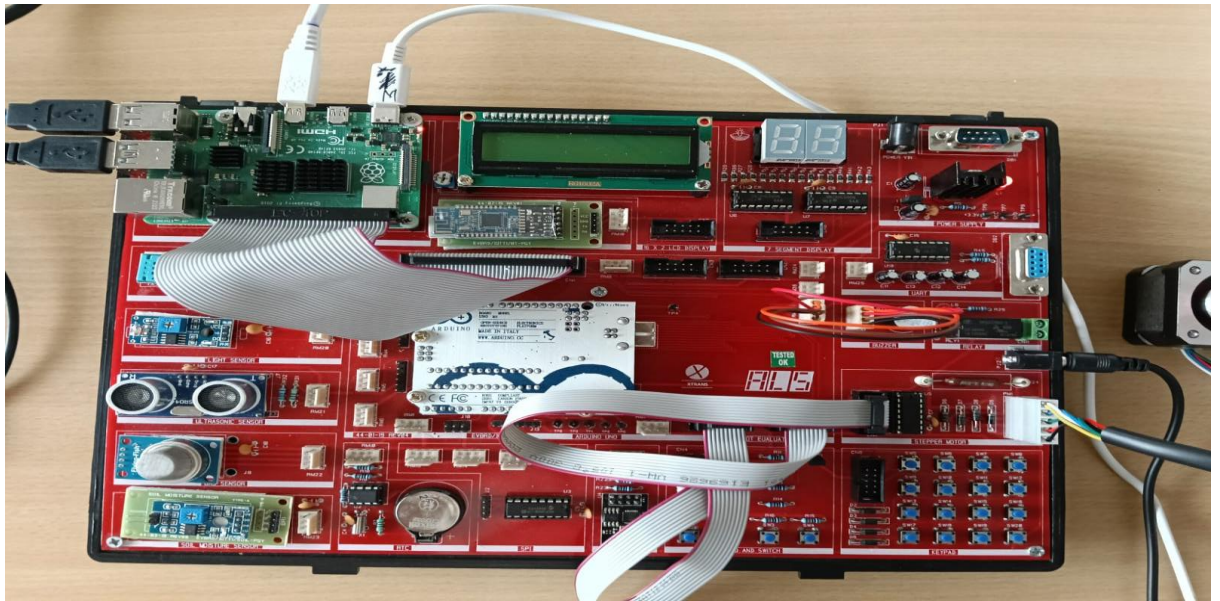
```
StepPins = [13, 4, 6, 5]
```

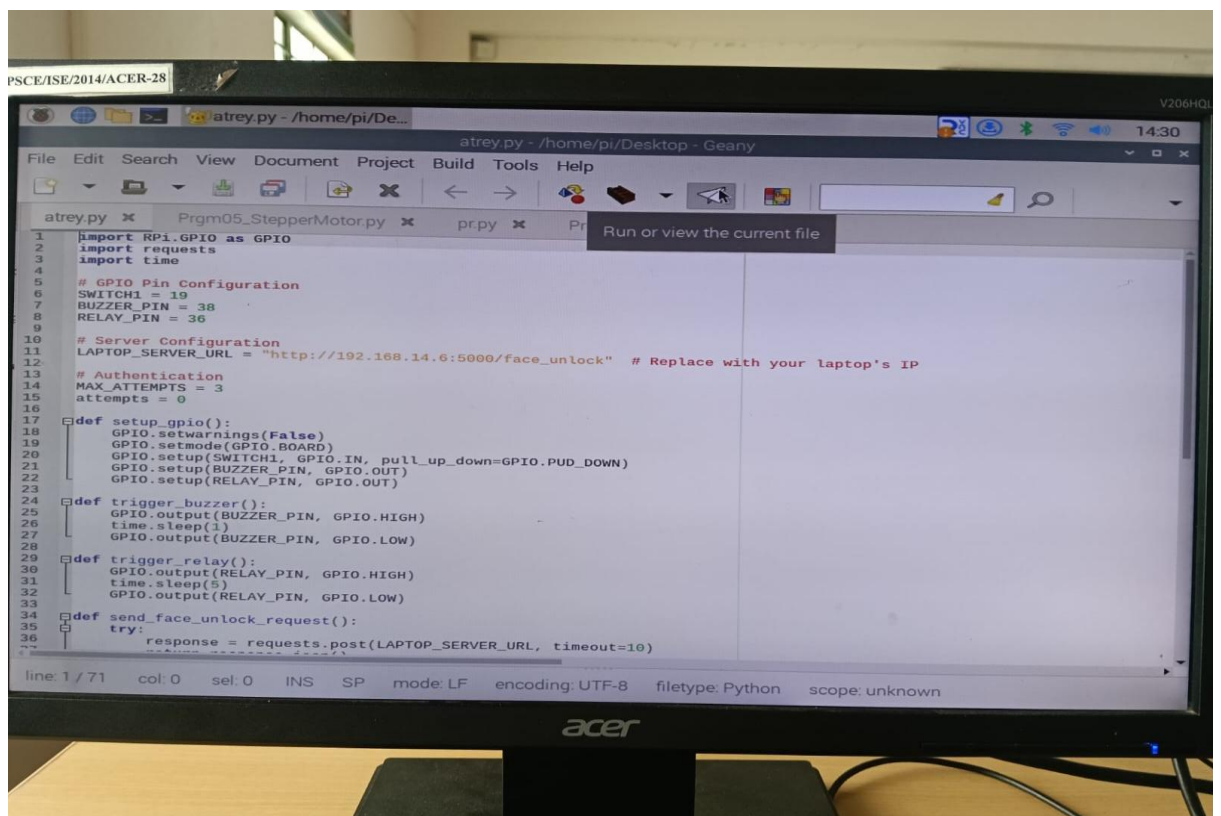
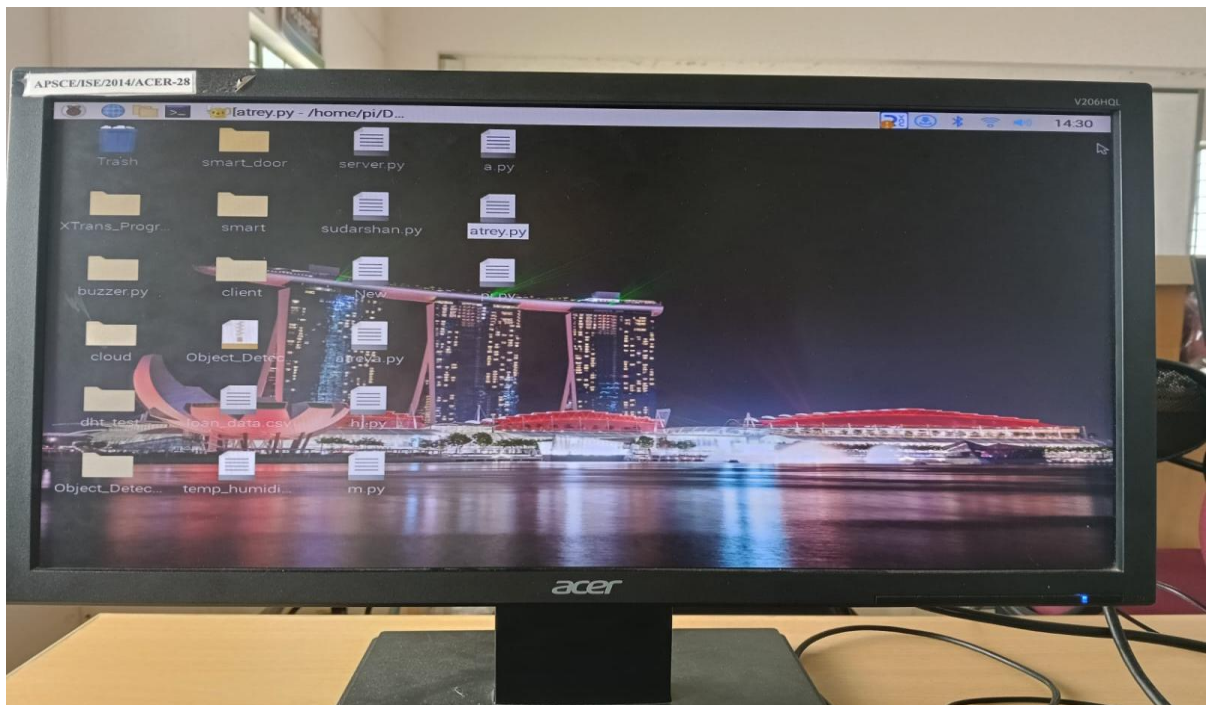
- The motor rotates based on predefined sequences to open and close the door.

```
from flask import Flask, jsonify
import cv2
import face_recognition
import os
from datetime import datetime
```

```
game.py > load_authorized_face
1 from flask import Flask, jsonify
2 import cv2
3 import face_recognition
4 import os
5 from datetime import datetime
6
7 app = Flask(__name__)
8
9 AUTHORIZED_FACE_ENCODING = None # Load your authorized face encoding here
10
11 # Directory to store unknown face images
12 UNKNOWN_FACES_DIR = "unknown_faces"
13 os.makedirs(UNKNOWN_FACES_DIR, exist_ok=True)
14
15 def load_authorized_face():
16     global AUTHORIZED_FACE_ENCODING
17     image = face_recognition.load_image_file("atreya.jpg") # Replace with your image
18     AUTHORIZED_FACE_ENCODING = face_recognition.face_encodings(image)[0]
19
20 def save_unknown_face(frame):
21     """Save the frame with unknown face as an image file."""
22     timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
23     filename = os.path.join(UNKNOWN_FACES_DIR, f"unknown_face_{timestamp}.jpg")
24     cv2.imwrite(filename, frame)
25
26 @app.route("/face_unlock", methods=["POST"])
27 def face_unlock():
28     cap = cv2.VideoCapture(0)
```

```
192.168.14.30 - - [10/Dec/2024 14:51:32] "POST /face_unlock HTTP/1.1" 200 -
192.168.14.30 - - [10/Dec/2024 14:51:40] "POST /face_unlock HTTP/1.1" 200 -
192.168.14.30 - - [10/Dec/2024 14:51:49] "POST /face_unlock HTTP/1.1" 200 -
192.168.14.30 - - [10/Dec/2024 14:52:17] "POST /face_unlock HTTP/1.1" 200 -
192.168.14.30 - - [10/Dec/2024 14:53:32] "POST /face_unlock HTTP/1.1" 200 -
```






```
pr.py - /home/pi/Desktop - geany_run_script_CH5...
geany_run_script_CH5JY2.sh
File Edit Tabs Help

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

/home/pi/Desktop/pr.py:30: RuntimeWarning: This channel is already in use, conti
nuing anyway. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(pin, GPIO.OUT)
/home/pi/.local/lib/python3.9/site-packages/pad4pi/rpi_gpio.py:127: RuntimeWarni
ng: A physical pull up resistor is fitted on this channel!
  GPIO.setup(self._row_pins[i], GPIO.IN, pull_up_down=GPIO.PUD_UP)
Security System Active. Press # after entering password.
Requesting face detection...
Unauthorized face detected. Please enter password on the keypad.

45 # Keypad Factory and Setup
46 factory = rpi_gpio.KeypadFactory()
47 keypad = factory.create_keypad(keypad=KEYPAD, row_pins=ROW_PINS, col_pins=COL_PINS)
48
49 # Keypad Callback Function
50 def process_key(key):
51     global entered_password, password_entry_needed, face_recognition_enabled
52     if password_entry_needed:
53         if key == '#':
54             if entered_password == PASSWORD:
55                 print("Password correct! Door is opening.")
```

```
File Edit Tabs Help

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

/home/pi/.local/lib/python3.9/site-packages/pad4pi/rpi_gpio.py:127: RuntimeWarni
ng: A physical pull up resistor is fitted on this channel!
  GPIO.setup(self._row_pins[i], GPIO.IN, pull_up_down=GPIO.PUD_UP)
Security System Active. Press # after entering password.
Requesting face detection...
Unauthorized face detected. Please enter password on the keypad.
Current Input: 4
Current Input: 47
Password correct! Door is opening.

22 [GPIO.LOW, GPIO.HIGH, GPIO.LOW, GPIO.HIGH, GPIO.LOW, GPIO.HIGH]
23 [GPIO.LOW, GPIO.HIGH, GPIO.LOW, GPIO.HIGH, GPIO.LOW, GPIO.HIGH]
24 [GPIO.LOW, GPIO.HIGH, GPIO.LOW, GPIO.HIGH, GPIO.LOW, GPIO.HIGH]
```

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

```
/home/pi/.local/lib/python3.9/site-packages/pad4pi/rpi_gpio.py:127: RuntimeWarning:  
ng: A physical pull up resistor is fitted on this channel!  
GPIO.setup(self._row_pins[1], GPIO.IN, pull_up_down=GPIO.PUD_UP)  
Security System Active. Press # after entering password.  
Requesting face detection...  
Authorized face detected! Door is opening.
```