

Unlock the Future

Home Security System with Face Recognition, One-Tap Live Surveillance, and Break-in Alert



Introduction

Have you ever been locked out of your dorm/apartment before? Or, have you ever lost a package because the delivery guy put it at your doorstep and someone else just took it? If your answer is yes to either of these questions, then our product can change your life. We use face recognition technology to unlock doors, which provides a powerful alternative solution to keypad and NFC based smart locks. Not everyone has a NFC-capable phone, but everyone has a face, right?

In addition to the amazing feature of leaving your keys at home for eternity, these new technologies sprung a new array of security concerns with its hackable nature. What if someone hacked the NFC key or have a picture of you, and they can open your door when you are not home? We solve this question by making full use of the current parts and

incorporating motion detection to our system, so that whenever movement is detected outside your door, the device would send the live video captured by the camera to your phone, and let you know of any suspicious activity.

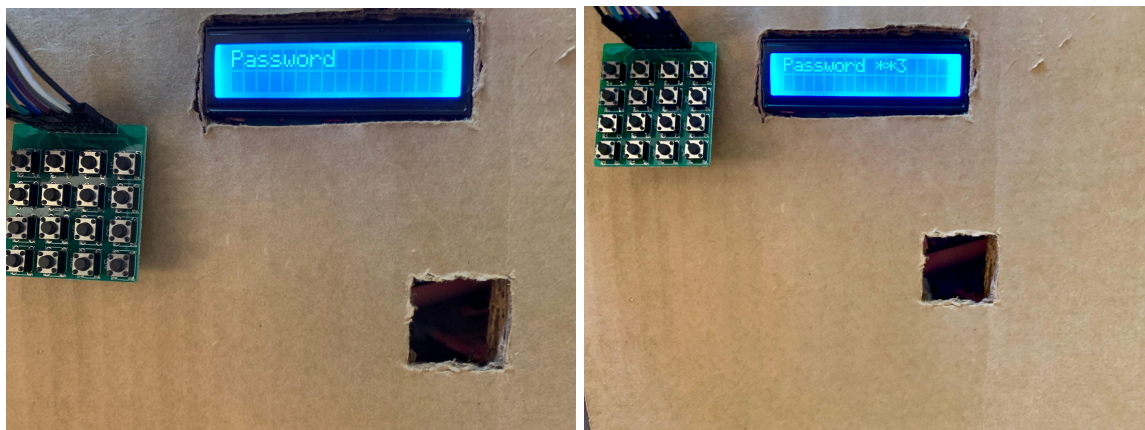
Features:

- Facial recognition
- Security alert
- Live surveillance video
- Lock your door remotely
- Low cost
- Easy to install

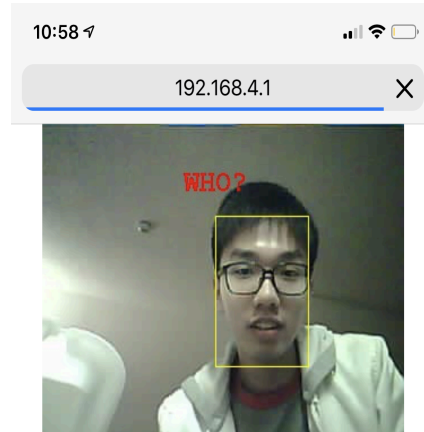
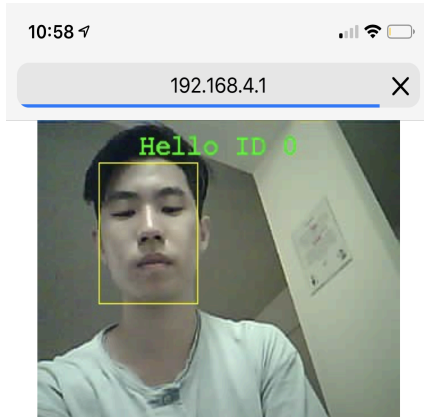
User's Manual

- Unlocking

1. Enter the six digit code. The code is unique to each resident/visitor.



-
2. Place your face in front of the camera. If the code you enter matches the identity detected by the camera, the screen will read “Welcome, <Your Name>” and unlock the door. Otherwise, user will be prompted to retry.



- Remote Security:

1. You will receive an alert on your mobile device when movement is detected outside your door. Alternatively, you can choose to manually enter a designated URL to a browser on any device.
2. You should see the live feed from the camera installed on the door. Recorded faces will have green caption, whereas strangers will have a red “WHO?” above their face
3. Based on this information, you can either choose to unlock the door for the visitor or call the police if you see suspicious activities.

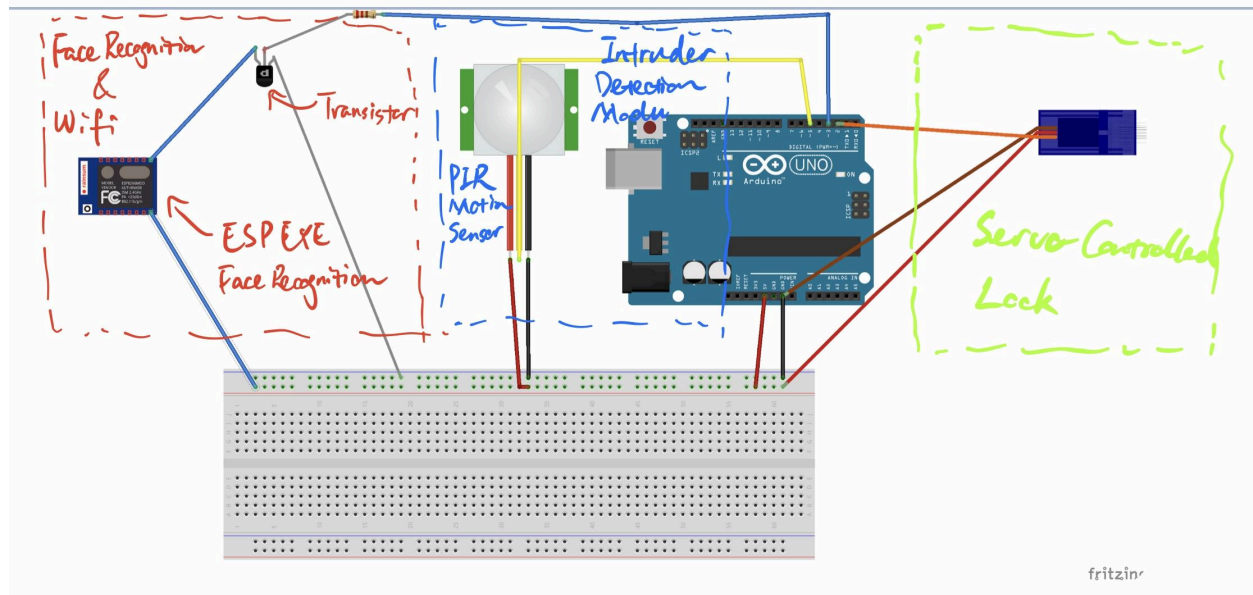
Project Breakdown

Budget & List of Materials

- \$25 ESP EYE Camera + WiFi module
- \$- Arduino Uno
- \$- PIR Motion Sensor
- \$- 2NP2222 Transistor

- \$6 Continuous Rotation Servo Motor
- \$1 Screw
- \$3 Slide Bolt

Jun 1, 2019 at 5:04 PM



Components Explained

- Face Recognition Module:

Comprised of ESP-EYE and transistor. ESP-EYE incorporates a microcontroller, a WiFi module, and a camera. This module enables our system to recognize recorded faces. For example, you can store your family's face images in the ESP-EYE and let them open the door without keys. We also use the ESP-EYE to capture and send video outside the door and run the face recognition program. The transistor is used to switch this module on and off to lower power consumption.

- Intruder Detection Module:

Uses a PIR motion detector. If the camera detect long time motions in front of the door, it will send an alarm to the user and the user can see who is outside.

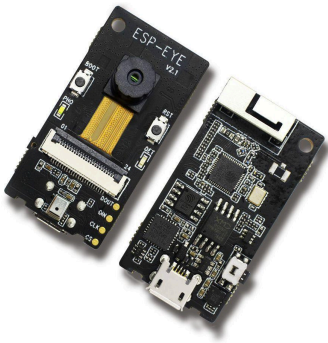
- **Arduino Controlled Slide Bolt Lock**

Uses a continuous servo attaching to a bolt with a nut on it. The nut is connected with the handle of the door lock which spinning the bolt in a fixed position will move the nut. The handle of the door will be pushed or pulled depending on the direction the bolt spins.

Core Technologies

(Why, How, What)

ESP-EYE Face Recognition Program



- ESP-EYE is an IoT development board made by company Espressif. It has a camera, a microphone, a BluFi module, and receives its power from USB cable.
- ESP-WHO & ESP-IDF IoT Design Framework

These are programming frameworks by Espressif. We referenced their sample program of a face recognition server. The program uploads the video stream captured by the camera to an http server and outputs information based on the face detected is recorded or not.

Microsoft Azure Custom Vision API

- The custom vision API from Microsoft Azure learns to recognize specific content in imagery and becomes smarter with training and time.
- This API works better to develop phone app to alert user of intruders than running the recognition program itself. However, due to time constraints, we were unable to apply this technology to our project.

Arduino Controller Program

- Set-up: the program first give 30 second to allow the PIR sensor to calibrate. Then, it displays the prompt for user to enter password.
- In the loop function of the arduino code, first a function getPassword is called to get the input from the keypad, store it in an array, and display the digit on LCD, while changing the previous digit to "*."
- Then, the program reads from the digital pin of the PIR Sensor to tell if motion is detected, and display an alert to the LCD screen.

LCD Display

- Uses LiquidCrystal Arduino Library to display text and interacts with the user.

PIR Motion Sensor

- Motion Sensor that detects the movement of infrared-emitting objects.

Keypad

- 4x4 Keypad that uses 8 digital pin on arduino to interact with it.

Continuous Servo

- Servo that spins without an angle limit.

Development Process

- **Stage 1:**

We came up with the idea of door because several break in happened to our friends. We think smart locks can help to solve security issues.

- **Stage 2:**

We tried several detectors to find suitable parts that could detect motions. Finally, we found that ESP-EYE has the most powerful functions and it can also deal with images.

- **Stage 3:**

We studied and wrote the code for ESP-EYE and connected it with wifi. So we can access the image from our phone. Then we use modulo to control a motor which helps to move the lock.

- **Stage 4:**

We built the circuit and a cardboard prototype for the smart lock.

- **Stage 5:**

We connected all parts together and did tests on it.

Help & Inspirations:

Advice from mentors, help from internet

- Find videos online about how to build a lock
- Find explanations and instructions about ESP-EYE
- Uses github repo: <https://github.com/espressif/esp-who>;
<https://github.com/andriyadi/esp32-custom-vision>
