

**TRƯỜNG ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ  
MINH TRƯỜNG ĐẠI HỌC BÁCH KHOA**

---



**THỰC TẬP ĐỒ ÁN ĐA NGÀNH**

**Xây dựng hệ thống giám sát camera phát hiện người lạ**

**GVHD: LÊ ĐÌNH THUẬN**

<b>STT</b>	<b>Nhóm Vibihai</b>	<b>MSSV</b>
1.	Hồ Trọng Tín	1920064
2.	Nguyễn Tuấn Kiệt	1927021
3.	Nguyễn Hữu Nghĩa	1927027
4.	Huỳnh Phạm Quốc Anh	1920001
5.	Mai Thị Kim Cương	1927007

## MỤC LỤC

I.	<i>Giới thiệu đề tài:</i>	3
II.	<i>Mô tả hệ thống:</i>	3
III.	<i>Functional Requirements</i>	3
IV.	<i>Non-Functional Requirements</i>	3
V.	<i>Use Case Scenario</i>	4
VI.	<i>Cơ sở dữ liệu:</i>	8
VII.	<i>Kiến trúc hệ thống:</i>	8
VIII.	<i>User story:</i>	10

## Mục lục hình

Hình 1	<i>Usecase</i>	4
Hình 2	<i>ERD diagram</i>	8
Hình 3	<i>Flow chart</i>	8
Hình 4	<i>Deployment view</i>	9

## **I. Giới thiệu đề tài:**

Công nghệ AI nhận dạng khuôn mặt là công nghệ cho phép nhận dạng một người cụ thể từ ảnh hoặc 1 đoạn video. Cách thức làm việc của công nghệ này là so sánh hình ảnh khuôn mặt với những hình ảnh sẵn có trong cơ sở dữ liệu để đưa ra kết quả.

Trong thời đại công nghiệp hóa - hiện đại hóa, công nghệ đang phát triển không ngừng nên vấn đề an toàn luôn được con người đặt sự quan tâm lên hàng đầu.

Project là sự ý tưởng kết hợp giữa công nghệ nhận diện khuôn mặt để xây dựng một hệ thống an ninh.

## **II. Mô tả hệ thống**

### **1. Actor:**

- Người sử dụng có thể xác minh bằng khuôn mặt.
- Người quản lý giám sát và quản lý thông tin ra vào
- Camera thu nhận hình ảnh gửi về server xác minh.
- Người quản lý có thể đăng ký khuôn mặt mới.
- Người quản lý có thể xem lại lịch sử ra vào.

### **2. System:**

- Hệ thống nhận diện và xác minh bằng khuôn mặt.
- Thông báo khi có người lạ.

## **III. Functional Requirements**

### **1. Người sử dụng:**

- Người sử dụng có thể đăng ký khuôn mặt mới tại camera của hệ thống.

### **2. Người quản lý:**

- Thêm, sửa và xóa thông tin người dùng

### **3. Hệ thống:**

- Hệ thống nhận biết được người đã đăng ký
- Nhận biết được người chưa đăng ký khuôn mặt và gửi thông báo đến mobile app.

## **IV. Non-Functional Requirements**

### **1. Camera :**

- Camera liên tục stream hình ảnh gửi về server

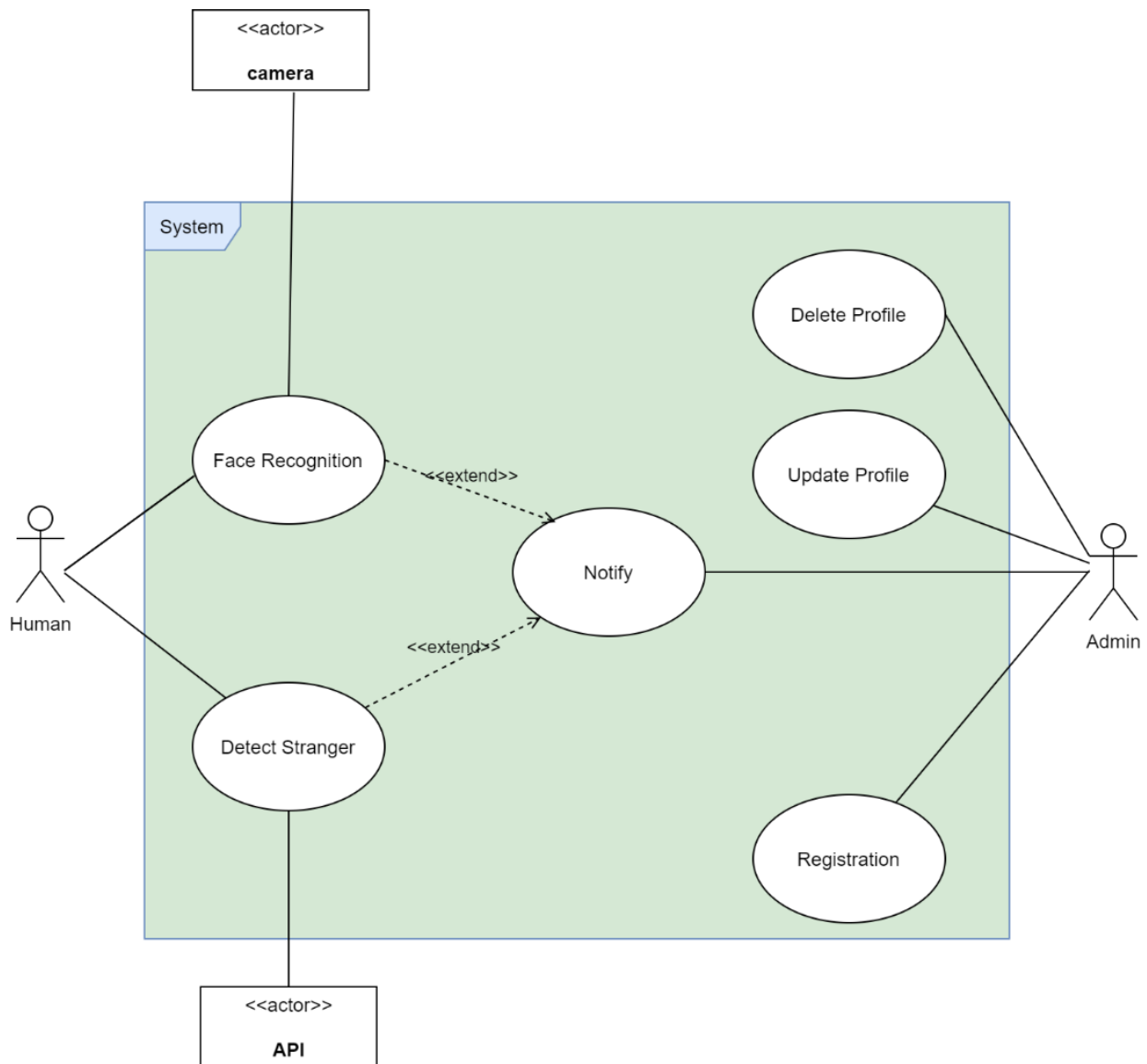
### **2. Main server (Django/Nodejs):**

- Hệ thống hoạt động liên tục 24/7.
- Có thể xử lý liên tục hình ảnh được gửi về từ camera.
- Tính toán song song để tăng tốc độ xử lý.

### 3. Mobile app (Flutter):

- Giao diện đơn giản, dễ sử dụng.
- Có khả năng nhận thông báo đẩy.
- Hoạt động được với tín hiệu mạng yếu.

## V. Use Case Scenario



**Hình 1 Usecase**

## 1. Nhận diện khuôn mặt

Tên Use Case	Nhận diện khuôn mặt
Mã số Use Case	UC01
Mô tả tóm tắt	Hệ thống có khả năng nhận diện khuôn mặt dựa trên dữ liệu hình ảnh được gửi về từ camera, biết được thông tin người dùng dựa trên hình ảnh khuôn mặt nếu là người dùng đã đăng ký thông tin.
Các bước thực hiện	<ul style="list-style-type: none"><li>● Bước 1: Camera gửi ảnh về server.</li><li>● Bước 2: Server xử lý hình ảnh, nhận diện khuôn mặt.</li><li>● Bước 3-1: Nếu người dùng đã đăng ký → Log thông tin ra vào.</li><li>● Bước 3-2: Nếu người dùng chưa đăng ký → Gửi thông báo đẩy đến mobile app cho admin làm việc.</li></ul>
Điều kiện thoát	Không
Yêu cầu đặc biệt	Không
Yêu cầu trước khi thực hiện	Có kết nối internet
Điều kiện sau khi thực hiện	Không

## 2. Gửi thông báo

Tên Use Case	Gửi thông báo
Mã số Use Case	UC02
Mô tả tóm tắt	Hệ thống gửi thông báo đến mobile app của người giám sát hệ thống.
Các bước thực hiện	Hệ thống gửi thông báo đến mobile app của người giám sát hệ thống
Điều kiện thoát	Thông báo được gửi đi thành công
Yêu cầu đặc biệt	Không

Yêu cầu trước khi thực hiện	Có kết nối internet
Điều kiện sau khi thực hiện	Không

### 3. Đăng ký khuôn mặt

Tên Use Case	Đăng ký khuôn mặt
Mã số Use Case	UC03
Mô tả tóm tắt	Đăng ký khuôn mặt người dùng vào cơ sở dữ liệu
Các bước thực hiện	<ul style="list-style-type: none"> <li>● Người quản lý mở ứng dụng di động đã cài sẵn</li> <li>● Hướng camera ứng dụng vào khuôn mặt người dùng cần đăng ký và ấn nút chụp</li> <li>● Người quản lý bấm nút đăng ký để gửi hình ảnh về server trung tâm.</li> </ul>
Điều kiện thoát	Người dùng ấn nút Thoát
Yêu cầu đặc biệt	
Yêu cầu trước khi thực hiện	Máy di động cần có camera sau và kết nối internet
Điều kiện sau khi thực hiện	Thông tin người dùng được gửi đến server thành công

### 4. Sửa khuôn mặt đã đăng ký

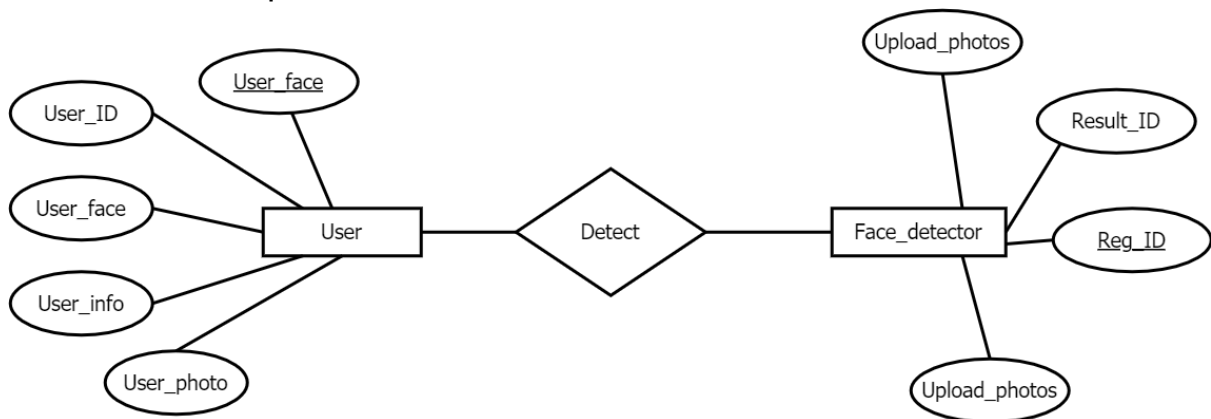
Tên Use Case	Sửa khuôn mặt đã đăng ký
Mã số Use Case	UC04
Mô tả tóm tắt	Sửa hình ảnh khuôn mặt người dùng trên cơ sở dữ liệu
Các bước thực hiện	<ul style="list-style-type: none"> <li>● Người quản lý mở ứng dụng di động đã cài sẵn</li> <li>● Chọn hình ảnh cần sửa</li> <li>● Hướng camera ứng dụng vào khuôn mặt người dùng cần chụp hình lại và ấn nút chụp</li> </ul>

	<ul style="list-style-type: none"> <li>● Người quản lý bấm nút đăng ký để gửi hình ảnh về server trung tâm</li> </ul>
Điều kiện thoát	Người dùng ấn nút Thoát
Yêu cầu đặc biệt	
Yêu cầu trước khi thực hiện	Máy di động cần có camera sau và kết nối internet
Điều kiện sau khi thực hiện	Thông tin chỉnh sửa được gửi đến server thành công

## 5. Xóa khuôn mặt đã đăng ký

Tên Use Case	Xóa khuôn mặt đã đăng ký
Mã số Use Case	UC05
Mô tả tóm tắt	Xóa hình ảnh khuôn mặt người dùng trên cơ sở dữ liệu
Các bước thực hiện	<ul style="list-style-type: none"> <li>● Người quản lý mở ứng dụng di động đã cài sẵn</li> <li>● Chọn người cần xóa</li> <li>● Người quản lý bấm nút xóa để xóa thông tin người sử dụng và thông tin người đó tại server trung tâm</li> </ul>
Điều kiện thoát	Người dùng ấn nút Thoát
Yêu cầu đặc biệt	
Yêu cầu trước khi thực hiện	Máy di động cần kết nối internet
Điều kiện sau khi thực hiện	Thông tin xóa được gửi đến server thành công

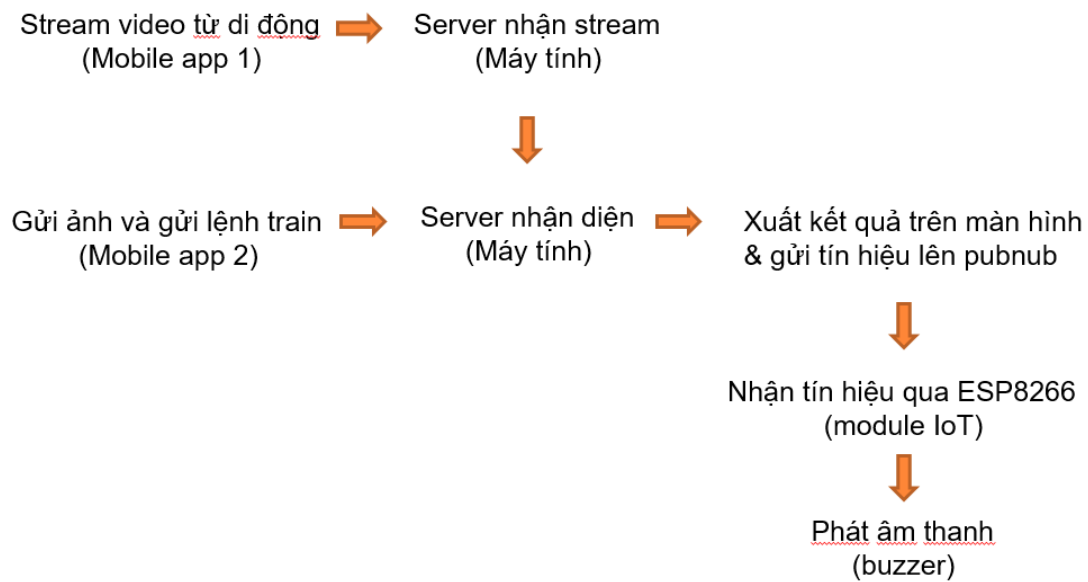
## VI. Cơ sở dữ liệu:



Hình 2 ERD diagram

## VII. Kiến trúc hệ thống:

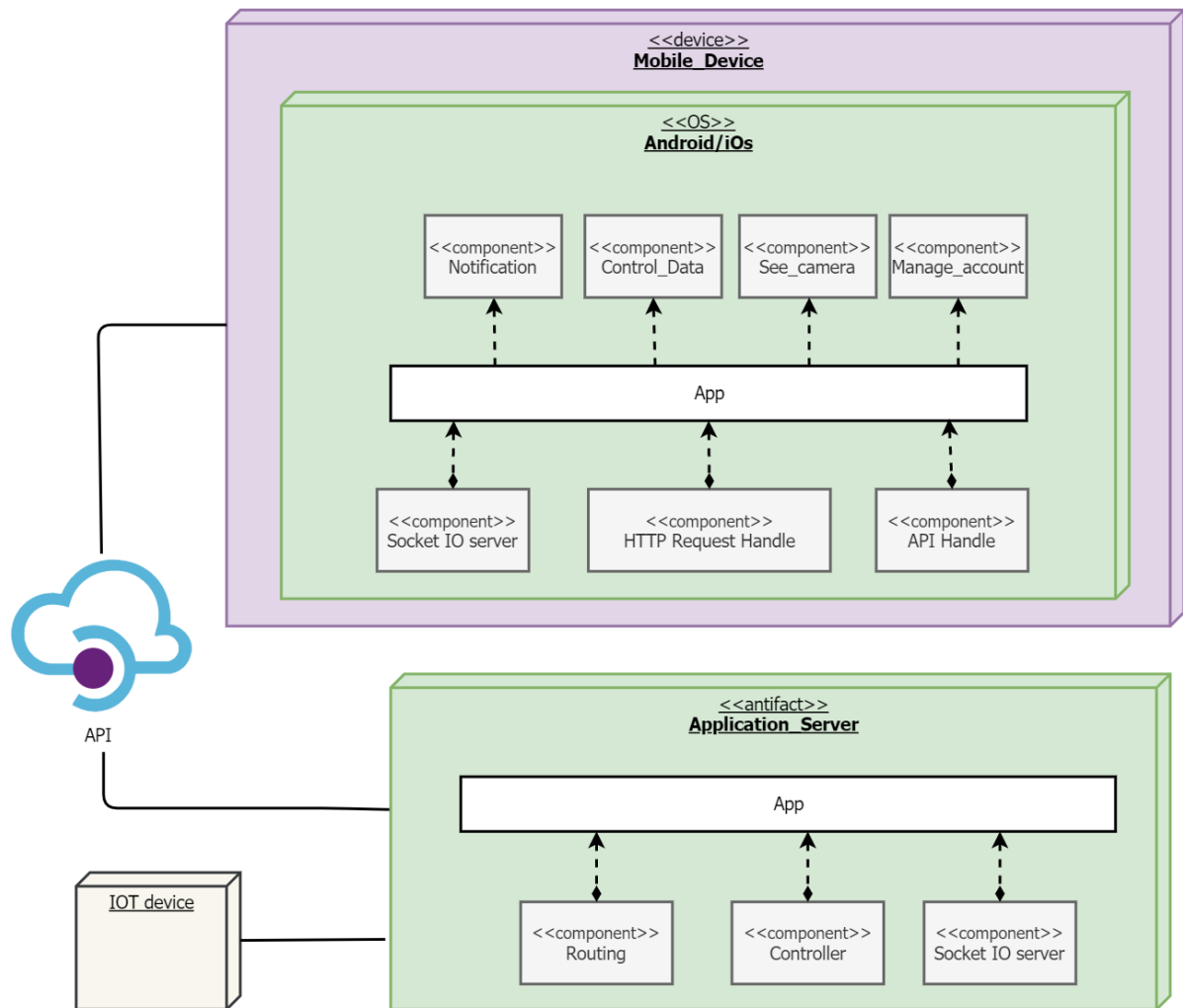
### ① Flow chart



Hình 3 Flow chart



② Deployment views:



**Hình 4 Deloyment view**

Hệ thống gồm 2 thành phần chính là Mobile App và Server. Thành phần tích hợp bên ngoài là PubNub REST API

### VIII. User story:

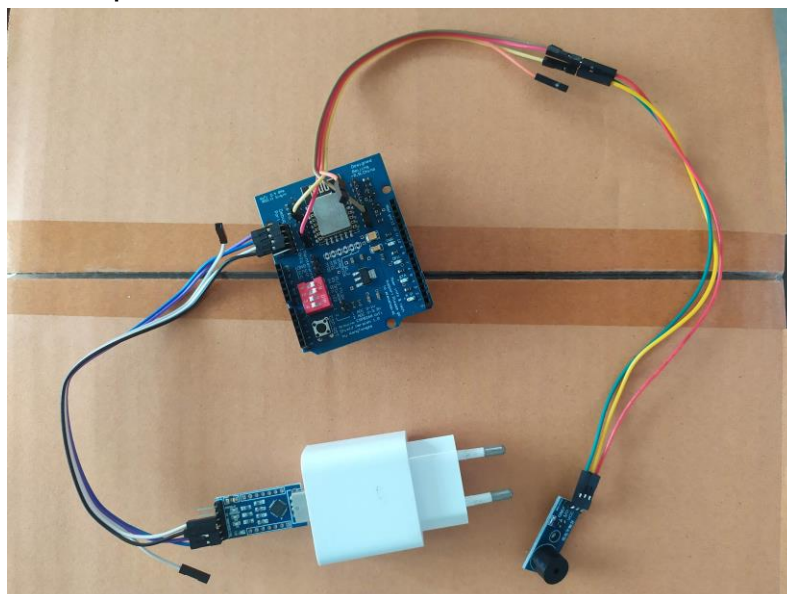
- ① Đăng Nhập:
  - . Người dùng truy cập vào ứng dụng.
- ② Tải dữ liệu:
  - Người dùng cập nhật thông tin

The screenshot shows a mobile application interface with a status bar at the top displaying '9:16 | 81,1KB/s' and signal indicators. The app has a blue header bar with the text 'Upload file to server'. Below the header, there are two green buttons: 'Capture Image' and 'Browse Image'. In the center, there is a square image of a man with dark hair wearing a light-colored shirt. Below the image, there are two input fields: one labeled 'URL' containing the text '192.168.10.10' and another labeled 'ID' containing the text 'nghianh'. At the bottom, there are two buttons: a yellow 'Upload' button and a red 'Train' button. At the very bottom of the screen, the text 'Status: Ready' is displayed.

- ③ Kiểm tra thông tin người dùng:



- ④ Thông báo khi có người lạ
- Thiết bị Module ESP8265 và loa buzzer

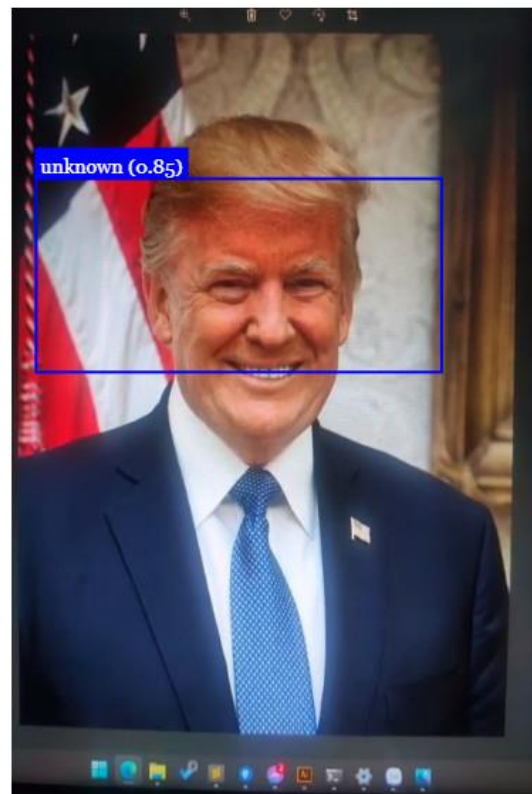


⑤ Kiểm tra hoạt động:

Với những hình ảnh đã được train, hệ thống hiển thị thông tin của người đó



Với những người không nằm trong dữ liệu, sẽ hiển thị không rõ (unknown)  
Đồng thời chuông báo sẽ thông báo đến người quản lí.



**IX. Tài liệu tham khảo:**

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.jsonx](http://arduino.esp8266.com/stable/package_esp8266com_index.jsonx)

<https://www.pubnub.com/docs/sdks/arduino/index#wifi101-shield-sample>

<https://www.npmjs.com/package/node-media-server>

<https://github.com/pubnub/arduino/tree/bda126eb054ff4327c5f789eaf74f92c4f064df4>

<https://github.com/pubnub/arduino/blob/master/examples/PubNubDemo/PubNubDemo.ino>