2017

IoT Project Document

fit5140 advanced mobile systems

IoT pROJECT PART B.

QIUXIAN CAI 27010767

YINGCHEN LIU 26981068

Table of Contents

[1 Project Description 3](#_Toc497595857)

[2 System Architecture 3](#_Toc497595858)

[3 Circuit Diagrams 4](#_Toc497595859)

[4 Libraries/APIs Used 5](#_Toc497595860)

# 1 Project Description

Lazy Mails is an IoT system aimed at intelligently differentiating variety of mails in a physical mailbox. The system distinguishes useful letters from junk mails and send notifications to the owner’s mobile phone, which saves the users’ time for checking mails everyday and avoid them from missing important letters.

# 2 System Architecture

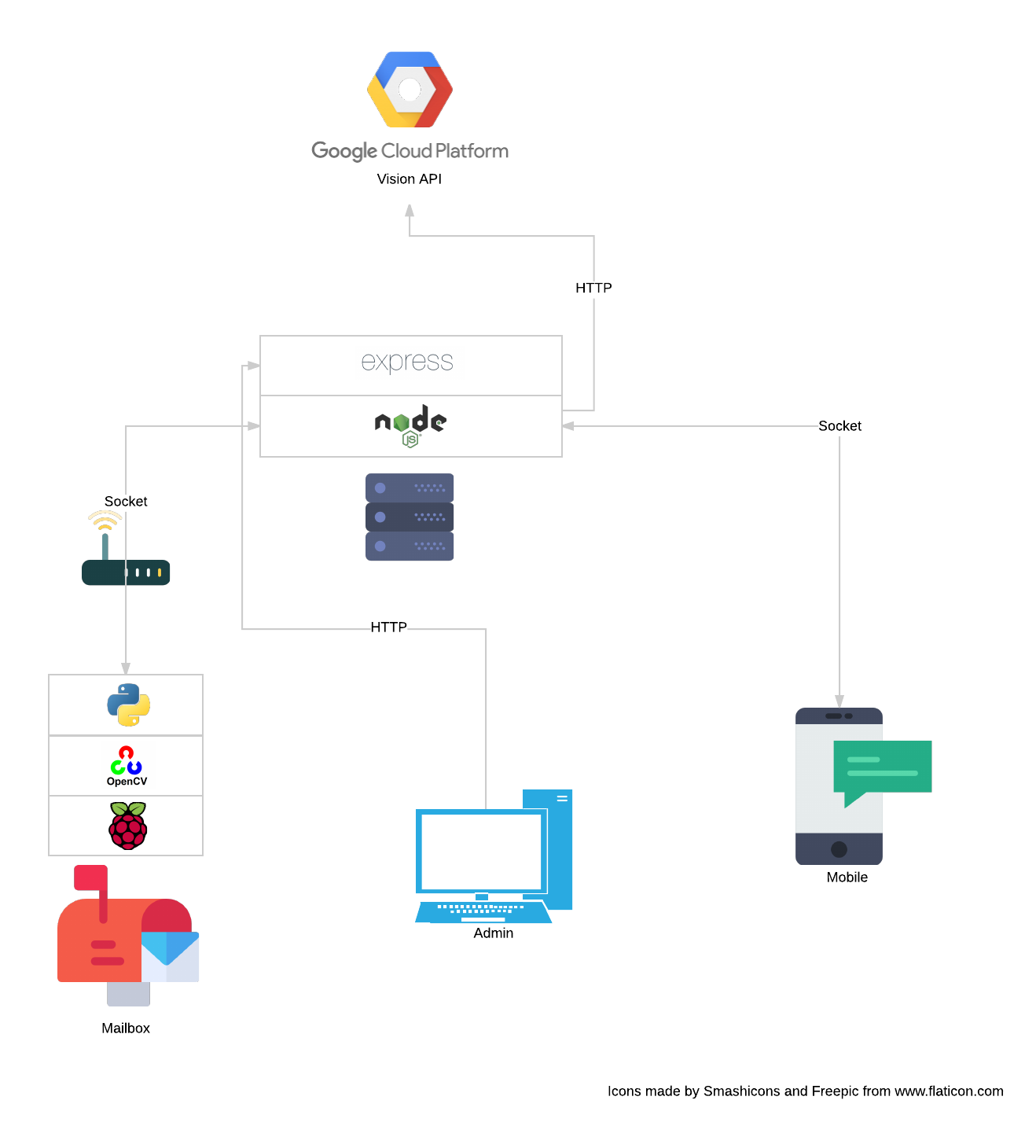


Figure 1. System Architecture

A Raspberry Pi is installed in the physical mailbox with a camera and motion detectors (see 3 Circuit Diagrams). We use three motion detectors because it can increase the success rate in detecting an arriving mail. The camera is used to capture the photo of the mailbox. By using background subtraction in OpenCV with Python, mail can be cropped. The Raspberry Pi sends a message containing the photo of the received mail to the server via TCP/IP protocol. The server then sends a request to Google Vision API via HTTP protocol for logo, text detection and labelling. After that, some post-processes, such as image rotating and mail classification, will be done before a notification can be sent to the mobile phone also via TCP/IP. By using multi-threading, this system can support multiple mailboxes. Also, for one mailbox, it supports multiple users. In addition, a web-based admin system built with ExpressJS enables administrators to manage the system easily using a web browser. For the server, MongoDB is used to store mailbox, user and mail’s information. PM2, which is a production process manager for Node.js, is used for keeping the server running all the time.

# 3 Circuit Diagrams

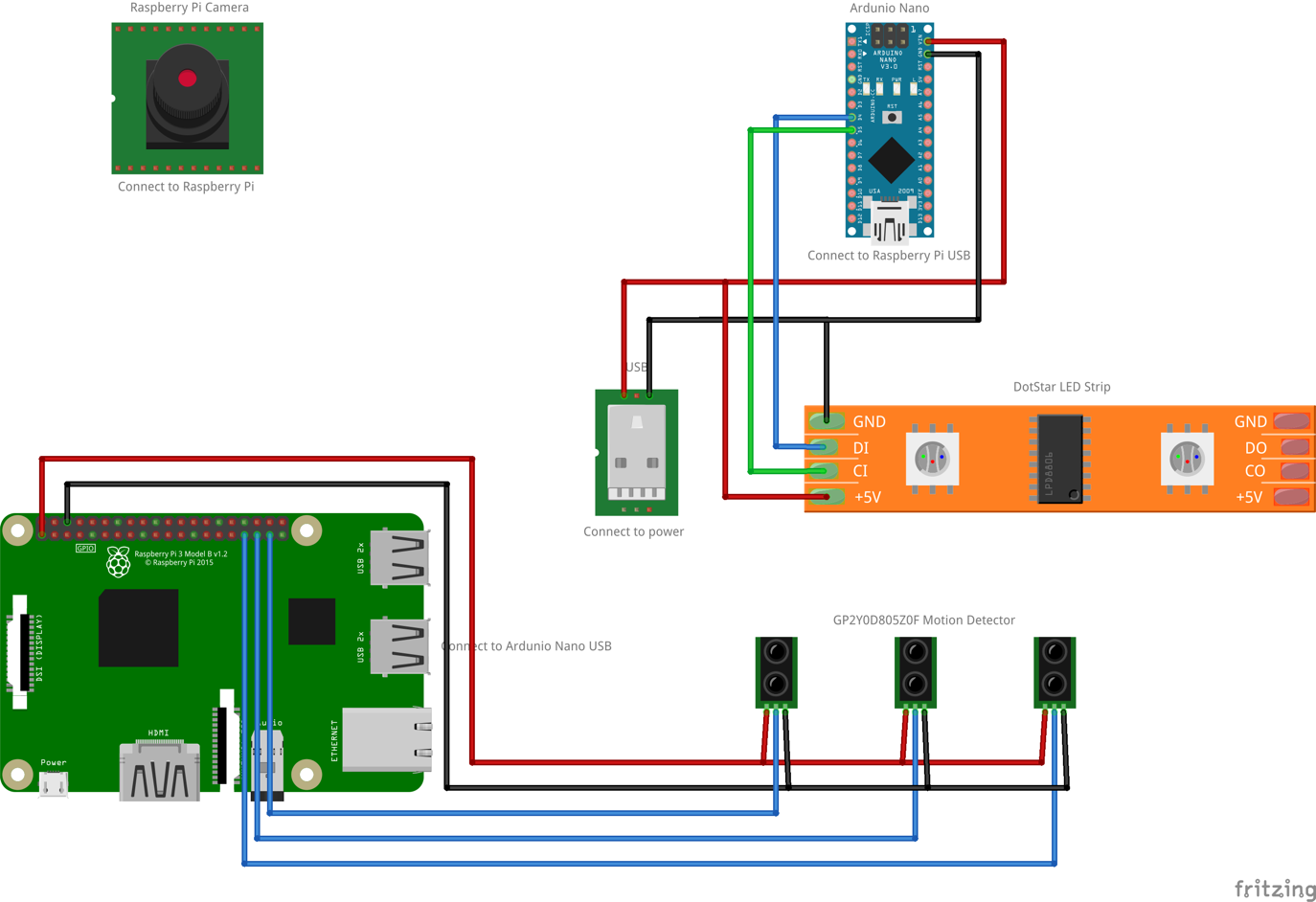


Figure 2. Circuit Diagrams

In a LazyMails box, to ensure that the camera works properly, an Arduino Nano is also installed to control a LED strip (5V-based). The Raspberry Pi uses serial to control the Arduino, so that the LED can be turned on or off. Another USB cable is used for power the LED strip as it requires high current.

# 4 Libraries/APIs Used

* iOS Application
  + QRCodeReader

As this project supports multiple users for any single mailbox, user register is required. QRCodeReader makes it easier for users in a family to link their accounts to a mailbox by scanning the QRCode provided by us. It is extremely hard to develop a QRCode scanner by ourselves, so we decided to use this library.

* + SwiftValidator

It is convenient to use this library to do validation as it has some useful and good built-in validators such as email and length. It is unlikely for us to write a perfectly working regular expression.

* + Whisper

It is used to display an in-app notification on the top of the screen. As there is not enough time for us to implement this functionality, we focused more on the core function of this application, so we have chosen this library.

* Server
  + ExpressJS with EJS template engine

This is a very powerful web framework.

* + Google Vision API

As computer vision is not the main focus of this assignment, we have chosen an API to extract information from images.

* + Axios

A library for making HTTP request. In this app, we used it to call Google Vision API.

* + Anchorme

It is used to extract URLs from text.

* + Basic-auth

We used it to do basic authentication for web-based admin system.

* + Fuzzyset.js

We used this to match user’s name on envelops in order to distinguish if a mail is important to the user or not. Because Google Vision ORC is not correct at 100%, we need to use this to do fuzzy math.

* + Image-size

This is used to get the size of an image.

* + Jimp

This is used to do image processing. In this app, we only use it to rotate an image.

* + Md5

We used it to generate a unique filename for an image, also it is used to encrypt password.

* + Moment

A datetime library for parsing date string.

* + Monk

A 3rd-party library for making query to MongoDB. It simplifies the process of connecting and cursor handling.

* + Serialize-error

We used it to serialize JavaScript error messages, so that they can be sent via JSON.

* Raspberry Pi
  + OpenCV

This is used to image processing. It uses machine learning to subtract object form background. In this app, we used it to extract the top-most mail from a stack of others.

* + Numpy

Required by OpenCV.

* + Pillow

Image processing. For example, in this app, we used it to compress images.

* + PiCamera

Used to capturing images from the camera.

* + PySerial

We used it to make connection to the Arduino Nano, so that we can turn the LED on and off.