

List of Hardware and Software

Project: Mail Gobbler 9000 (MG9K - Smart Mailbox/Dropbox)

Document: List of Hardware and Software

Team members: Luan Vo, Adam Cytrynowski, Jackie Chan, Brendan Truong

The following is a list of all hardware and software modules used to date, as of MDR:

Hardware Modules:

1. ESP8266

- Used as a means of communication between our **Arduino Uno** and our AWS database. Establishes

2. MCR12 Barcode Scanner

- Reads all incoming barcodes and sends each number as an ASCII byte, all within our loop. Each value is then stored in a list, and when the list has completed and no more data is being transmitted to the barcode reader, the list is sent to AWS and then cleared, waiting for the next set of barcode values.

3. 12V - Solenoid

- This solenoid is the locking mechanism of the mailbox. When the system is in off state, the solenoid is in the locking position.

4. Push button

- Power button of the mailbox, switching between on and off state.

5. ATmega328P - Brains of operation, transport data from barcode scanner to ESP8266.

6. Pressure Plate - Used to detect whether or not mail has arrived. If mail has arrived, a change of resistance will occur, and that change will be recorded as an incoming package.

Software Modules:

1. Backend Module

The following tools are used within the Backend Module, which as a whole acts as a means of data storage and manipulation for the system along with being the pipeline for communication between the system's hardware and front end mobile application.

a. Amazon Web Services (AWS)

AWS is a cloud computing platform that offers various tools and services that enable our means of communication and data storage.

i. AWS IoT Core

IoT Core is a service that allows us to easily connect our ESP Wi-Fi module for communication with the hardware via topics to publish or subscribe to, while offering means to integrate with other AWS services.

ii. AWS Lambda

Lambda is a serverless computing platform that offers the use of dynamic functions. This allows us to program functionality that may be called and utilized at any time. We have functions coded in lambda to make data comparisons from our hardware to our database, update our database, communicate messages from the mobile app to the hardware, and to notify users via push notifications of real time updates/changes.

iii. AWS AppSynch

AppSynch is a service that easily creates a GraphQL API, via a user defined schema. This allows us to create queries and mutations that operate in the backend, that we may utilize within our mobile application. Example queries are those that allow us to extract table values of our barcodes and delivery logs to display for the user.

iv. AWS DynamoDB

DynamoDB offers the ease of creating NoSQL databases so that we may store our data. We utilize three tables: the first being `barcode_table`, that stores our barcodes the user uploads and whether or not they have been delivered, the second is the `mail_table` which stores the time and date of when mail arrives, and lastly is the `package_table` that stores scanned barcodes and the time that they were scanned.

v. AWS IAM

IAM allows us to manage functionalities in the backend, to restrict certain functions in accessing only the required services/data to uphold our security and avoid mishaps.

vi. AWS EC2

EC2 is a means of utilizing a virtual machine, which we use for backend testing. By treating a virtual machine as an IoT thing, we were able to debug and analyze backend functionalities.

vii. AWS Cognito

Cognito allows users to sign up, sign in, and access the database. It is simply a means of authentication through the mobile app to avoid malicious users.

viii. AWS Simple Notification Service (SNS)

SNS is a service that allows us to easily integrate push notifications, so that whenever our DynamoDB tables are updated, relevant notifications may be sent to the mobile user's device.

2. Frontend Module

The tools in this category were used for creating the user mobile application, which enables the user to be able to unlock the mailbox via the app, upload barcodes into the backend, and check previous logs of barcodes, letter-mail and packages.

a. App Creation

i. xCode

xCode is the IDE created by Apple for the development of iOS applications. Code is written in Swift, and the interface can be created using either Storyboard or SwiftUI (2.0). xCode has a built-in device simulator which allows the programmer to test their app within the IDE.

ii. Google Drawing

Google Drawing allows for the use of shapes to create charts and diagrams. In this case, it was used to create the app icon.

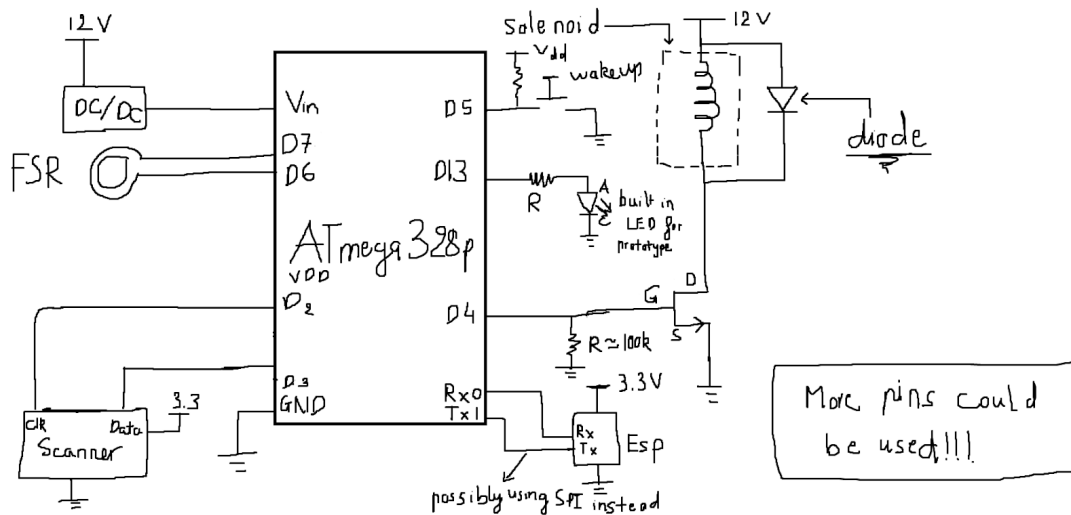
iii. Github

Github is a hosting service that offers distributed version control and source code management, allowing developers to push/pull their version of the code.

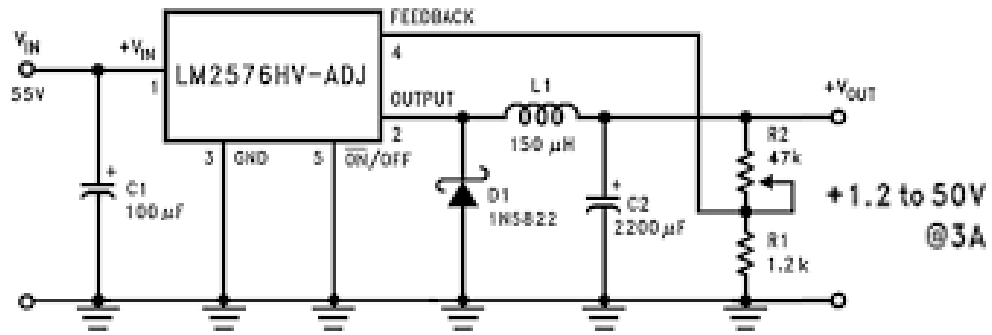
The following is a list of all hardware and software modules we plan to add or implement in the next stages of the project, **during the Spring Semester:**

Hardware Modules:

- 1. Atmel Studio
- 2. PCB migration plan



- 3. Buck converter



Software Modules:

1. HTML and CSS
 - a. We'll be programming in HTML and CSS for our project's website in the coming semester.
2. Github Pages
 - a. We'll host our website with Github Pages unless directed otherwise.