EEC 172 Lab 6 Report

Alejandro Torres
almtorres@ucdavis.edu

ABSTRACT

In this paper, I present my process of utilizing my CC3200 LaunchPad and AWS to create a security system. I set up the CC3200 to connect with the AWS IoT and utilized SNS to send push notifications to an associated email. The board also utilizes the built-in accelerometer to detect whether the device has been moved or not. The user will utilize the universal remote controller to activate the security system.

General Terms

AWS IoT means Amazon Web Services Internet of Things. SNS means Simple Notification System.

1. INTRODUCTION

This laboratory was divided into two main sections. The first part involved interfacing the CC3200 LaunchPad with AWS to update and retrieve the shadow state of my registered CC3200 device.

After that, the focus was on using a Simple Notification Service (SNS) to allow registered users to receive notifications whenever the shadow state was updated. After creating the SNS, I utilized the remote and decoder logic from lab 3 to register a password to activate the security system.

The final part had to do with configuring the interface of the security system and initializing the accelerometer to detect whether or not the device had been moved. This will then call the previous parts of the lab to tie together all of the operations of the security system.

2. Procedures

2.1 Connecting AWS and SNS

Initially, I created a new Thing in my AWS specifically for my security system. I did this to make a new device shadow and created the necessary certificates and keys to initialize the connection between the CC3200 LaunchPad and the AWS server.

Next, I created a new rule and subscription to enable the SNS communication system. I registered it with the desired email address, which will be the one to receive the notifications that get sent via my post function.

2.2 Security System

To set up the security system, I configured a welcome screen to display on the OLED screen, prompting the user to activate the system by setting a new password. The welcome screen conveys a message instructing the user to enter the password, which is facilitated using the IR controller (with decoding logic from lab 3). The password is displayed on the screen to inform the user of the password they are setting. Once the user sets the password, they receive an SNS notification to their email confirming that the "Password is set."

Following this, the system becomes active, and its status is displayed on the OLED screen. The security system is designed to trigger only when the CC3200 LaunchPad is moved. To detect movement, I accessed the accelerometer by reading from and writing to the addresses that hold the x and y coordinates. These values are used to determine if the device has been moved.

When movement is detected, the user has the opportunity to enter the password to deactivate the security system. A countdown timer starts running when the device is moved. If the user fails to enter the correct password within the allotted time, the system transitions from the "active" state to the "intruder" state. In the intruder state, the OLED screen displays an intruder warning, and the user receives an email notification stating, "WE HAVE AN INTRUDER."

3. Problems and Solutions

I will discuss the problems encountered during the lab and how they were resolved.

3.1 Decoder

There was an issue with obtaining consistent results when decoding the buttons on the remote. I used the same decoder logic from Lab 3, which had provided consistent results. However, when I applied the same logic to my code, it was not consistently reading the decoded values. It's possible that the implementation of a countdown timer when the BMA detected movement may have interfered with the timing of the GPIO interrupt.