

Project Work Progress Report No 4

Date: 03/31/2019

Team #: 33

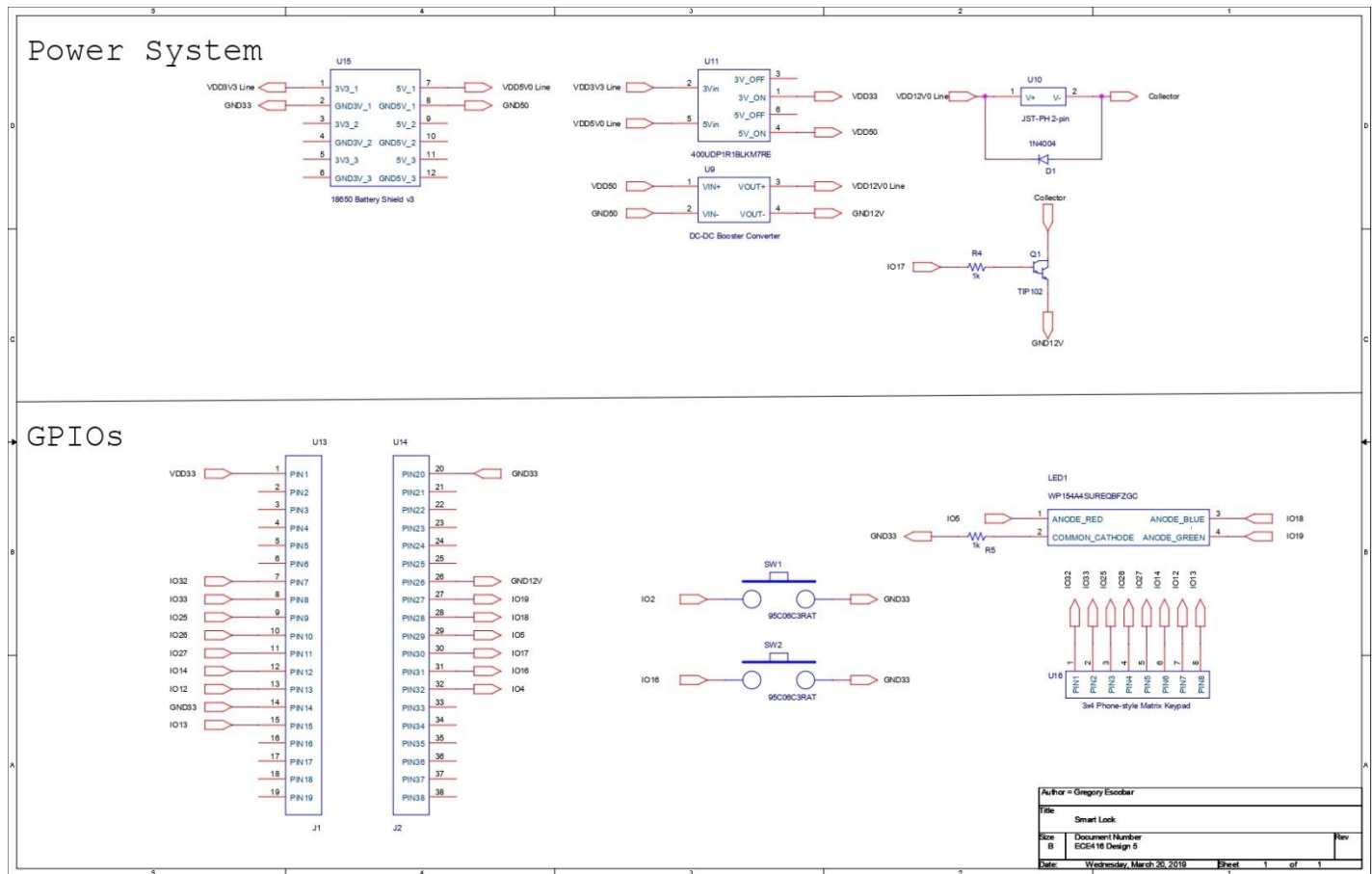
Project Title: Anti-theft Package Security Home System (APSHS)

Submitted by Gregory Escobar, Stephen Benavides, Shawn Carnevale, Geomar Reyes

1. Project progress since last report:

Hardware

The power system was tested in different scenarios. Results obtained showed that this system could power the microcontroller (ESP32), the solenoid and LEDs in case of a power outage for at least 10 hours due to the high capacity 18650 lithium cell. Solenoid was tested at different voltages in order to find the optimal input that would consume as low power as possible. GPIOs and power pins were finalized to proceed to PCB design.



Software:

The ESP 32 software has been successfully implemented into the hardware. The team was able to test the Wi-Fi connection by using a cellphone with hotspot included as a router to transmit a signal between the ESP32 and the APP by using the mobile network provided by AT&T. The test was successful, so we were able to send the right signal from the APP to the lock to open or close it at will. Our next goal was to check if the app was able to communicate across networks that are intended to be used by outside guests such as the ones that you get at a coffee shop, they are usually open for everyone and unsafe. For this test, we used NJITguest with a student credential to access. Unfortunately, the NJIT network asks for a specific set of permissions and confirmations when accessing their network; therefore, the test for direct connection with the NJIT network was unsuccessful. Then we decided to use a “router” to redirect the signal from the NJIT into the ESP 32. We were able to do so by sharing hotspot with the cellphone to the esp32, but instead of using the mobile connection we shared the NJIT Wi-Fi through the phone. Thus, proving that we can connect our device on any network that has a router to redirect the signal.

Application

- Our goal was to add a background for the app, we made it look more professional and user friendly
- We tested the communication between the hardware and the app through the ESP 32
- A link was successfully created between the app and the ESP32
- The user interface can store the username and password by using a TinyDB to store that information in the app's cache
- The user interface sends error messages when the username or password is incorrect

Network Framework

- Connection has been made successfully between the lock, the esp32 and the application.
- The ESP 32 can now build a web server to send and receive requests from the exterior.
- A protocol has been implemented into the esp32 to change the way in which we access into the server. We no longer have to put an IP address into the search bar of any device connected in LAN, now we can use an specific name to make it easier to access i.e. from <http://192.168.43.57> we changed it to <http://smartlock.local>
- The code that requests signals from the ESP 32 has been modified using an encrypted password, and the only way to decrypted is by using the app.

Future Upgrades

- Implement a method to send messages to the user when initializing the smart lock for the first time
- Implement a method to send messages to the user every time the door has been access
- The messages will be sent to the costumer via text messages and email to make it safer. The user will have control over everything that happens around the lock on real time

- Create a dedicated web server <http://smartlock.local> where the user is going to be able to know the status of the lock. i.e. How long the door has been active for as well as the temperature of the internal mechanisms.

2. Milestones achieved:

- PCB design was finalized
- Components were solder and tested onto the PCB
- Keypad implementation initialized
- Connection between the app, ESP 32 and lock was finalized
- Encrypted passwords have been implemented
- Encrypted requests from the server has been implemented
- The ESP32 only responds to the users with the app
- The app has become the solely virtual key for the lock

3. Is project on schedule? YES X NO

4. Next steps: (elaborate specifically on any problems listed in 3 and if the answer in 4. Is NO)

Keypad Implementation: The roadblocks we mainly encountered was with the keypad. Since the keypad was only Arduino compatible product, we had some issues with getting the keypad to work properly with the ESP32 and how to connect it to the ESP32. When we would press a 4 on the keypad the program would output a 6 and vice versa. After a further investigation as to why this was happening, we found out it had to do with the 4 and 6 being different resistances than the other keys. To fix this we had to modify our code and connect each key to different GPIOs to get the right results. After this, we are planning to add the keypad to our security system by allowing the user to select a 4-digit passcode.