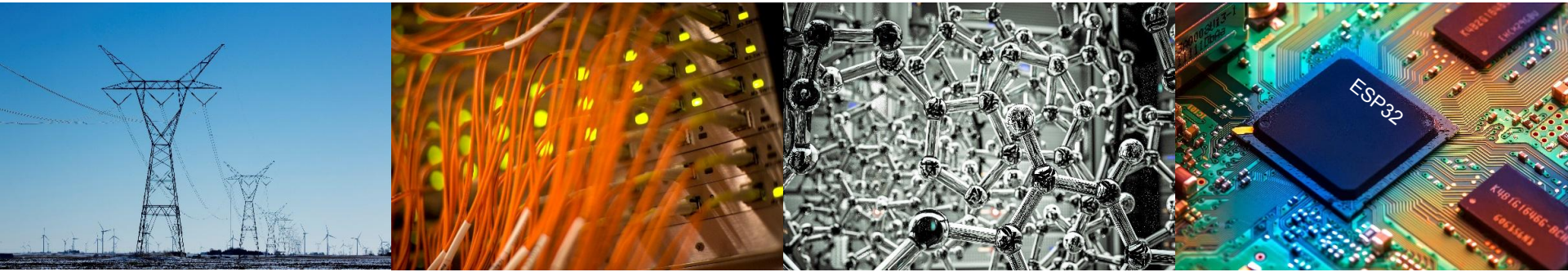
Secure Mailbox with Mobile Connectivity

Neehar Sawant, Roshun Navin, Avadh Patel

ECE 445
Team 26



ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

Objective

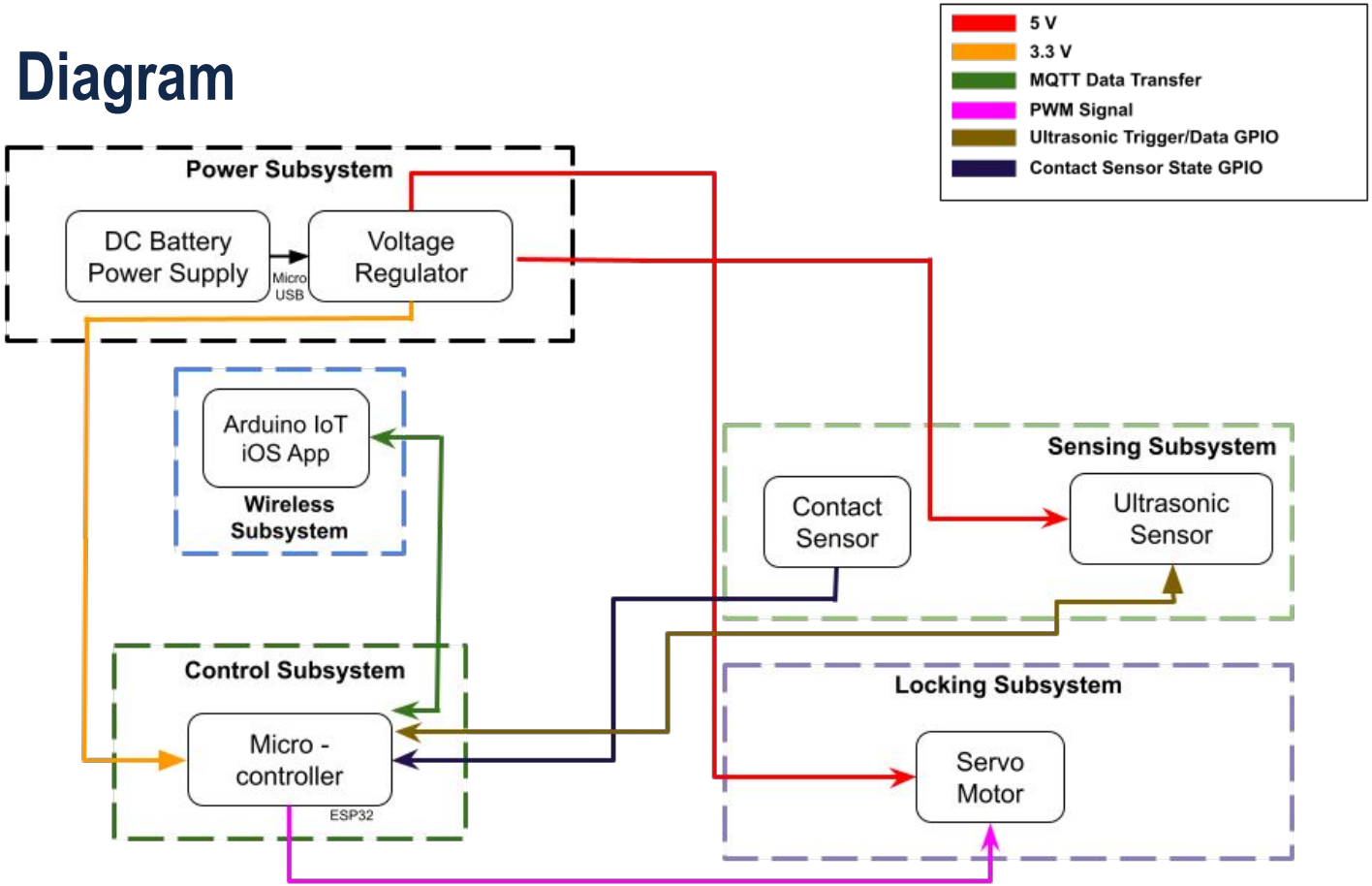
Problem: Mail theft from unsecured mailboxes

Solution: Automatically locking mailbox with mobile control and notifications

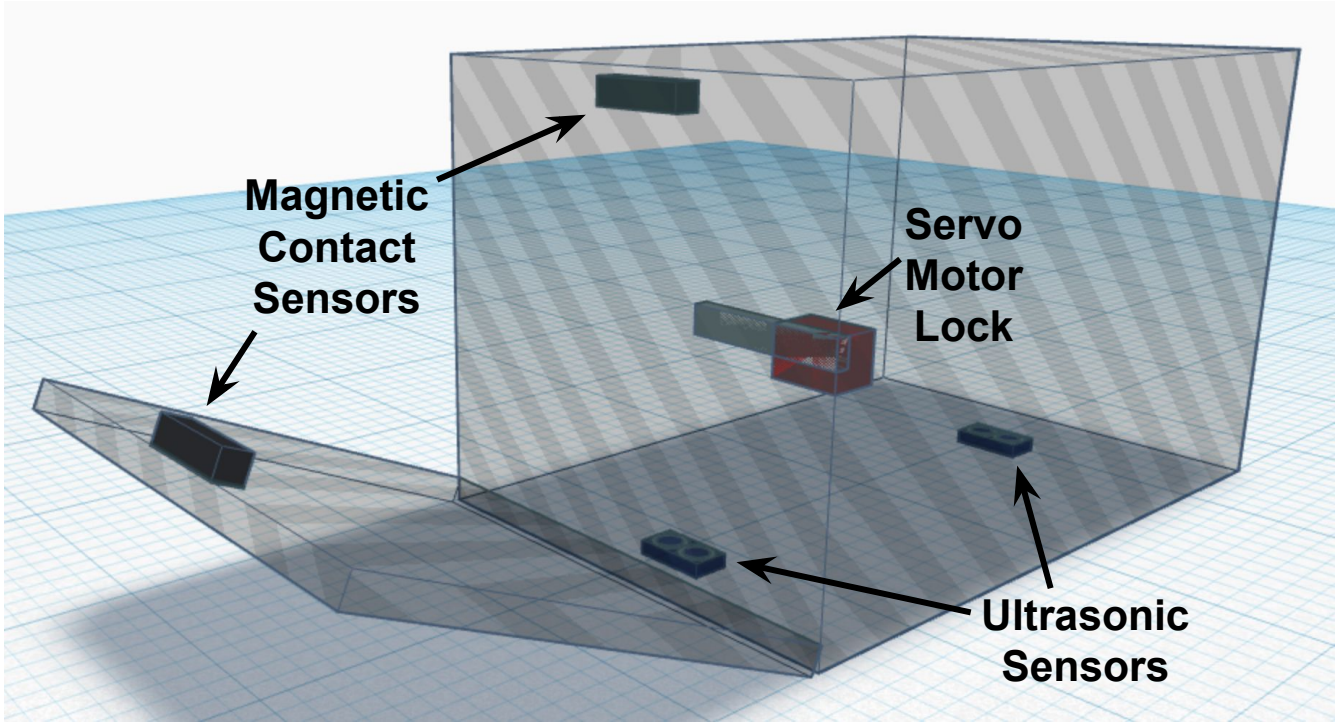
High Level Requirements

1. The mailbox must **automatically lock within 10 seconds** of the door being closed if mail is placed inside and there is no schedule set to leave it unlocked.
2. The mailbox must **lock and unlock within 5 seconds of pressing the corresponding button** on the application.
3. The mailbox must **send a notification within 30 seconds of an action** being made on the mailbox. This includes opening and closing the mailbox as well as whether mail is present.

Block Diagram

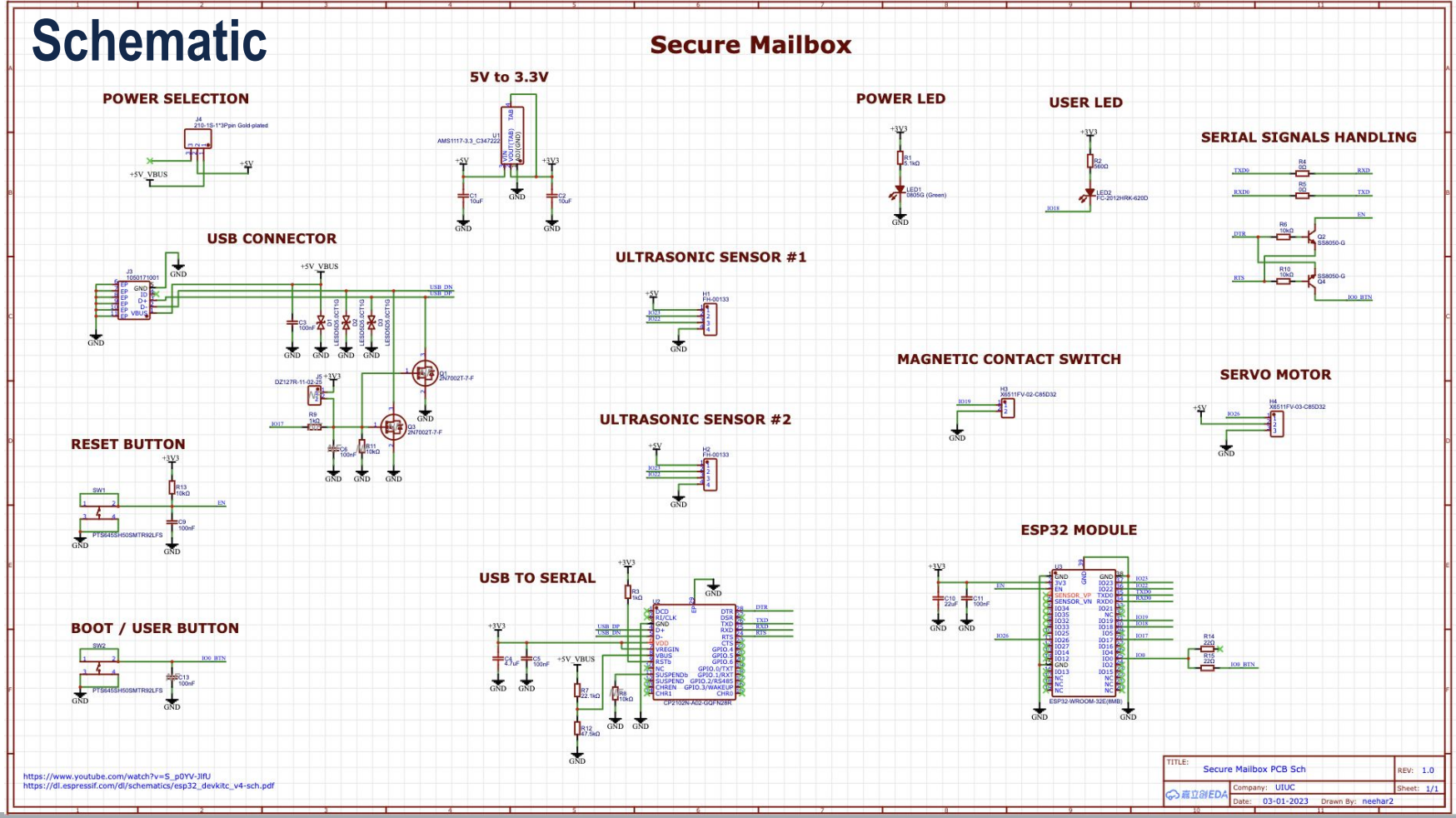


Original Mailbox Design



Completed Design

Schematic


Secure Mailbox
POWER SELECTION
5V to 3.3V
POWER LED
USER LED
SERIAL SIGNALS HANDLING
USB CONNECTOR
ULTRASONIC SENSOR #1
MAGNETIC CONTACT SWITCH
SERVO MOTOR
ULTRASONIC SENSOR #2
RESET BUTTON
ESP32 MODULE
BOOT / USER BUTTON
USB TO SERIAL
https://www.youtube.com/watch?v=S_p0YV-JIfU
https://dl.espressif.com/dl/schematics/esp32_devkitc_v4-sch.pdf
TITLE: Secure Mailbox PCB Sch
REV: 1.0
Company: UIUC
Sheet: 1/1
Date: 03-01-2023
Drawn By: neehar2

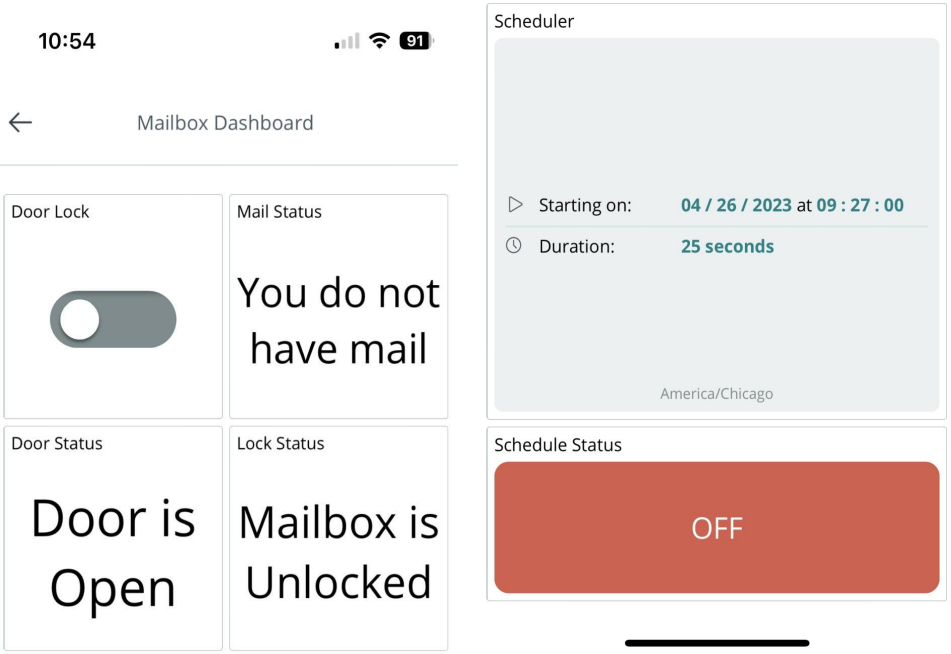
ESP32 with Built in WiFi

- ESP32-WROOM processes mailbox logic and sensor data
- Allows for mobile application communication
- Surface mounted on PCB

Power

- 10,000mAh battery bank delivers 5V to PCB via micro USB
- 5V delivered to ultrasonic sensors and servo motor
- 3.3V delivered to ESP32-WROOM

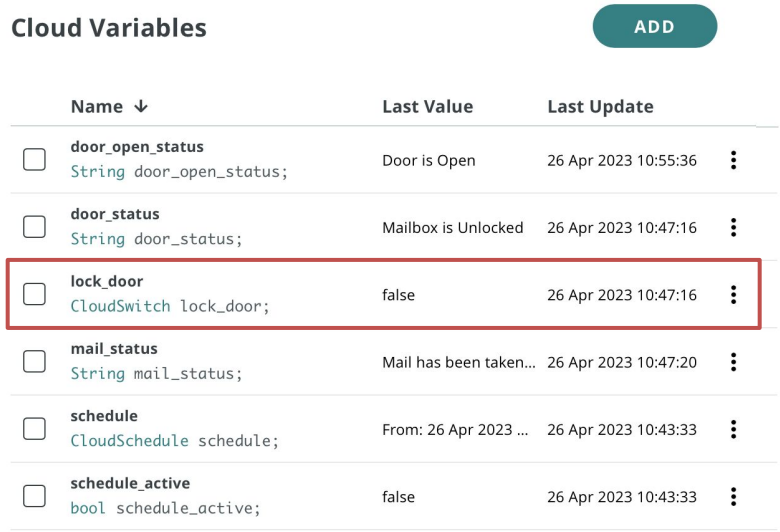
Mobile Application UI



- Gives user access to manual lock and unlock
- Set a schedule for time where mailbox is unlocked
- Real time status of door, lock, schedule, and mail

Arduino IoT Cloud

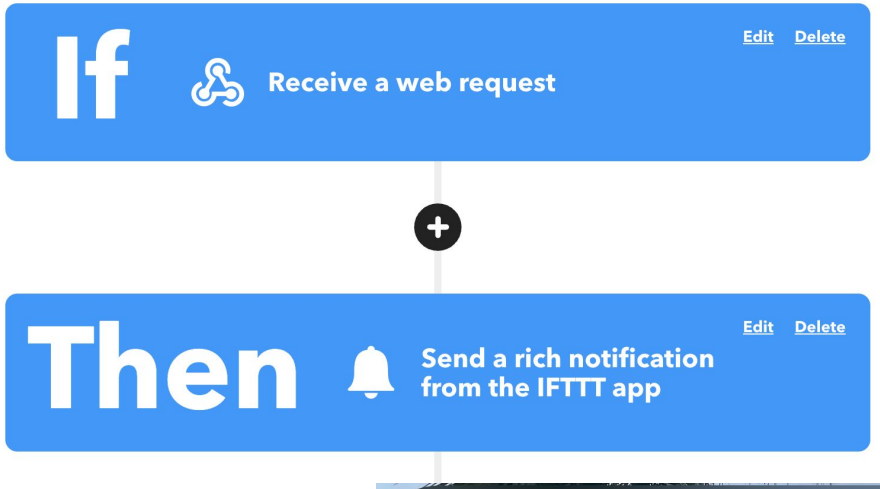
- Arduino IoT Cloud allows anyone connected to the internet to interact with hardware
- Functions will run if cloud variables are changed (such as locking the door)

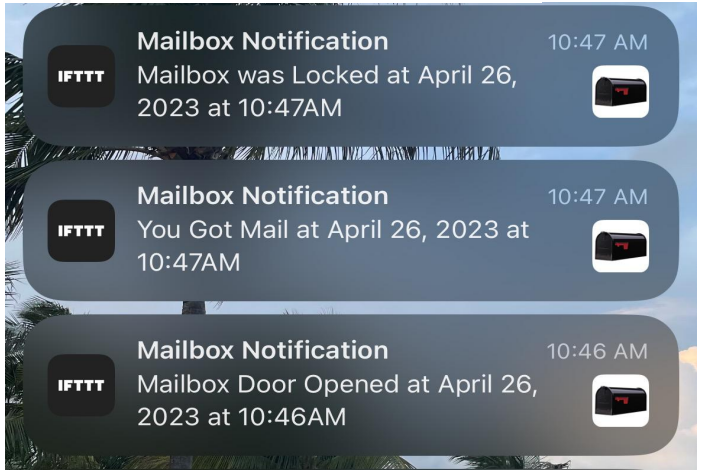



```
/*
  Since LockDoor is READ_WRITE variable, onLockDoorChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLockDoorChange()  {
  // Add your code here to act upon LockDoor change
  if(lock_door == 1){
    lockDoor();
  }
  else{
    unlockDoor();
  }
}
```

Mobile Notifications

- IFTTT allows notifications to be sent to anyone with the app
- Web request from software -> IFTTT web server -> user's phone






```
void sendMessage(String message){
  http.begin("https://maker.ifttt.com/trigger/door/with/key/keyvalue?value1="+message);
  http.GET();
  http.end();
}
```

Project Video

Quantitative Analysis

Action	Time (s)*
automatically locking mailbox when mail is present	0.35
locking/unlocking mailbox from mobile app	0.48
updating mobile app dashboard	0.86
receiving push notification for mailbox updates	2.21
locking/unlocking mailbox from schedule	0.74

**average of 15 attempts*

Ultrasonic	Distance (cm)
Lower bound	2.35
Upper bound	18.93

Challenges

- MQTT protocol and notifications being blocked by WiFi network
- Connectivity issues between AWS IoT and ESP32
- Soldering WROOM and SMD components
- 3D printing design
- Placing ultrasonic sensors

Conclusions

What We Learned

- Cloud Applications
- Arduino C controlling sensors
- Wireless connectivity protocols
- PCB Design
- PWM signals

What We Would Do Differently

- Adding button for manual lock
- 3D print the top of the mailbox separately
- Housing for sensors/motor in mailbox

Further Improvements

- Metal structure instead of 3D printed plastic
- Camera for live feeds and increased range of detecting mail
- More robust locking mechanism
- Potentially use solar power and a larger mAh rechargeable battery
- Ensure waterproofing

Thank you!
Q&A