Drawer Organization System

Michael Grawe, Nathan Marchosky, Michael Stoens

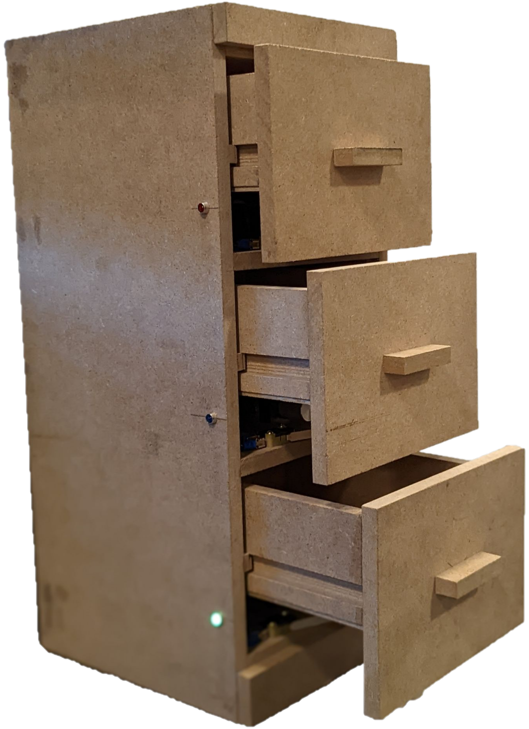
Team 11
TA: Stasiu Chyczewski

Problem

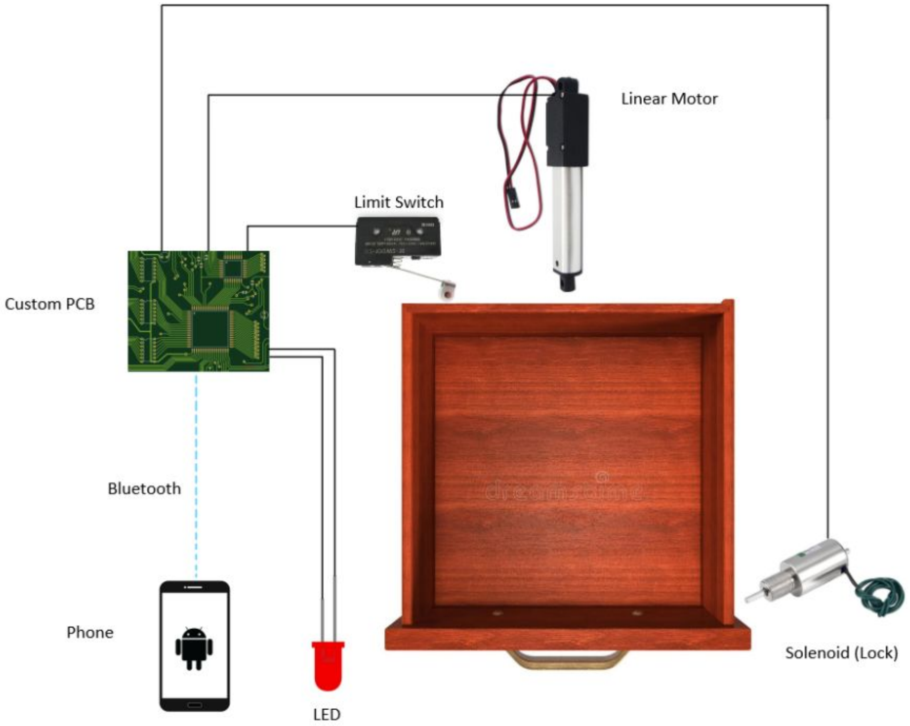
Can’t find your items in drawers?

Solution

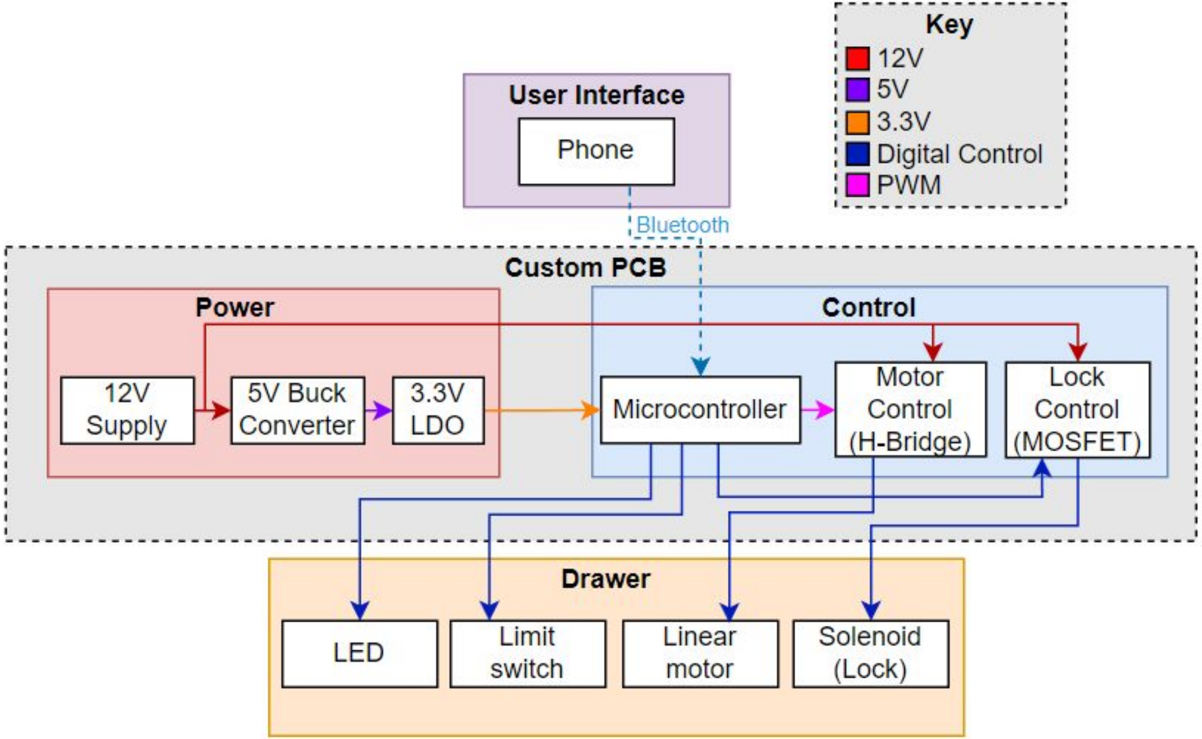
cricket 4:32
LTE 100%
Friday, May 20 72°F
Assistant
Photos
Play Store
Google



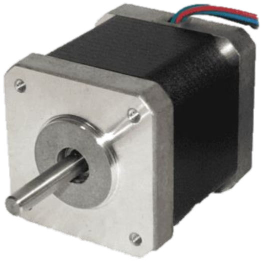
Original Design

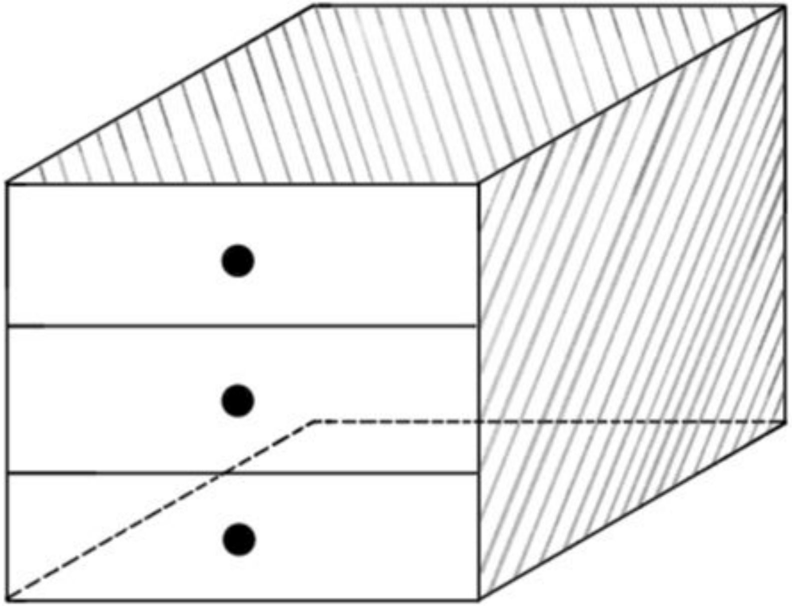


Original Design: Block Description

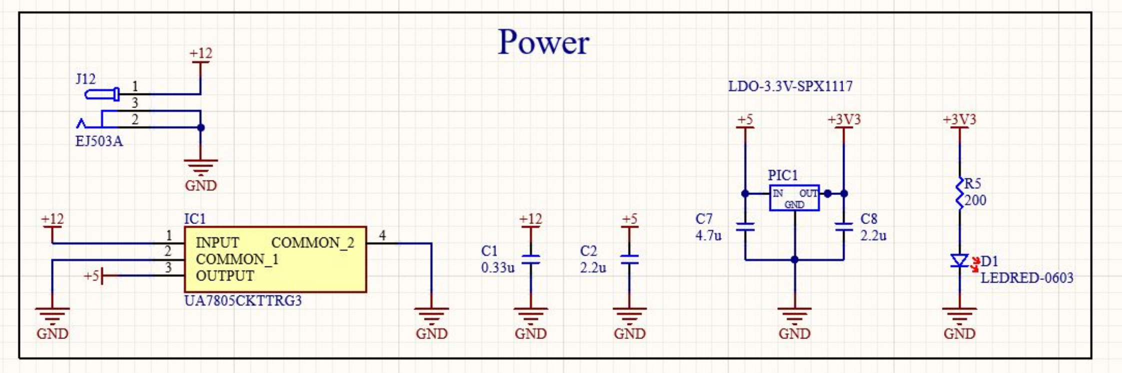


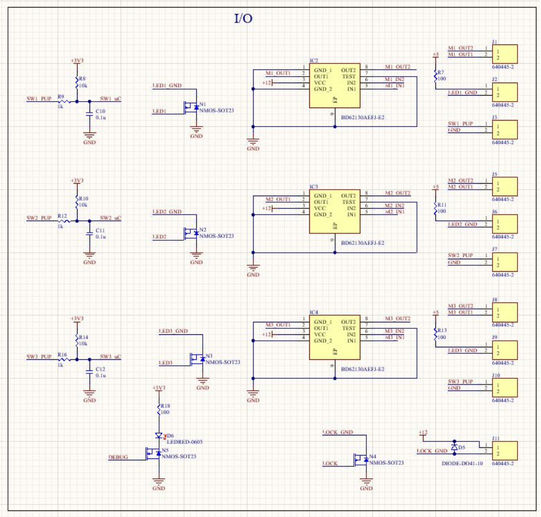
Design Process: Initial Steps

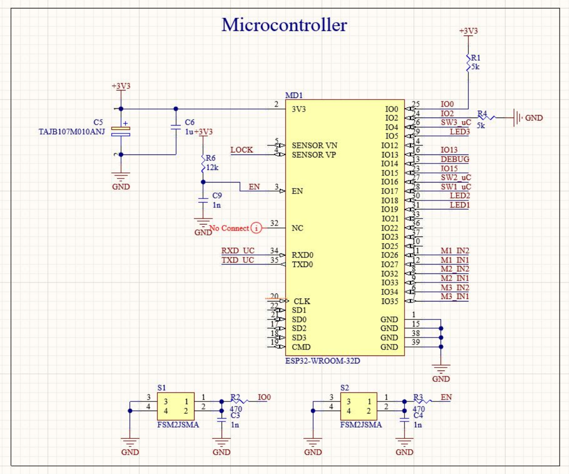


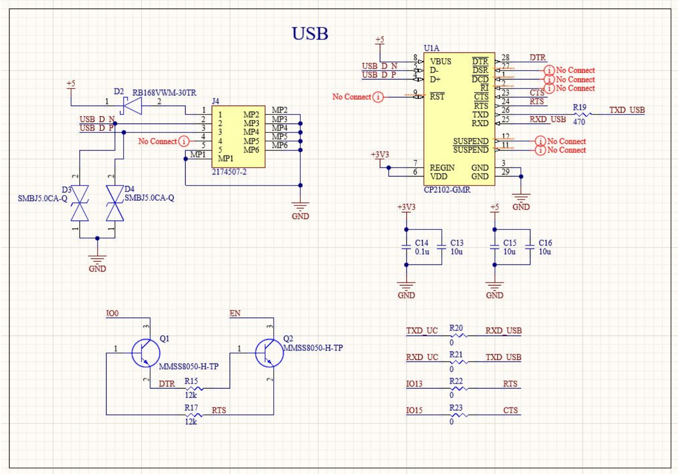


PCB Design

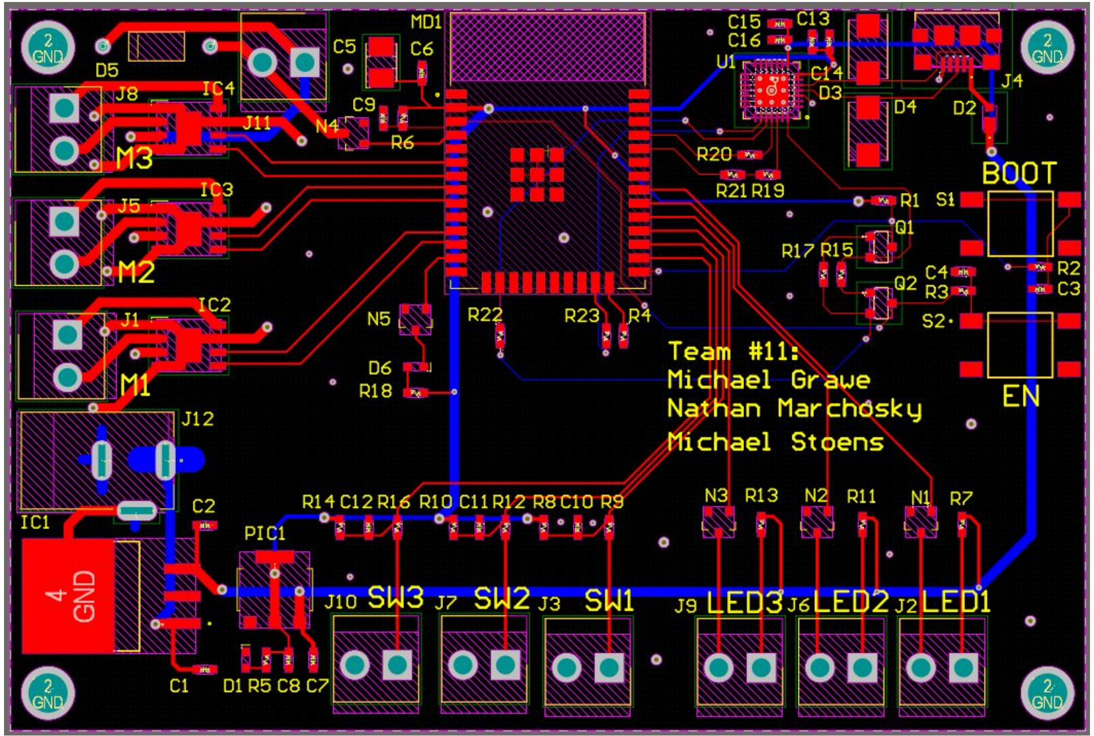

Power
J12
EJ503A
+12
GND
IC1
INPUT
COMMON_1
OUTPUT
COMMON_2
UA7805CKTTRG3
+5
C1
0.33u
C2
2.2u
LDO-3.3V-SPX1117
+3V3
PIC1
IN
OUT
GND
C7
4.7u
C8
2.2u
R5
200
D1
LEDRED-0603

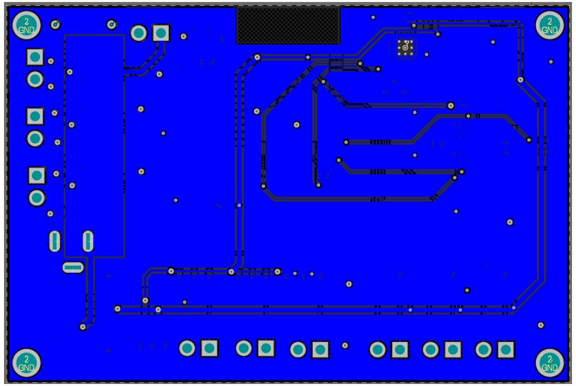

I/O
BD62130AEFJ-E2
NMOS-SOT23
640445-2
LEDRED-0603
DIODE-DO41-10

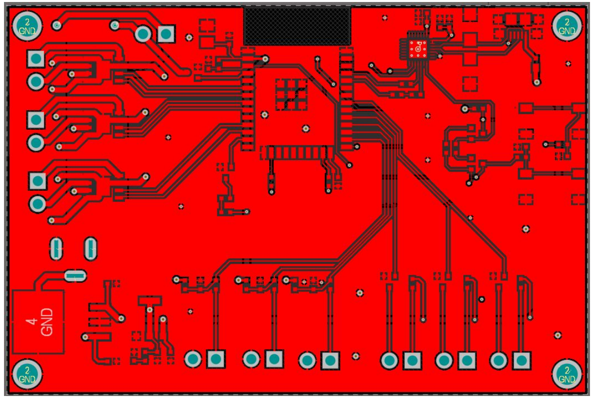

Microcontroller
MD1
ESP32-WROOM-32D
+3V3
C5
TAJB107M010ANJ
C6
1u
R6
12k
C9
1n
LOCK
EN
No Connect
RXD_UC
TXD_UC
R1
5k
R4
5k
GND
S1
S2
FSM2JSMA
R2
470
R3
470
C3
1n
C4
1n
IO0


USB
U1A
CP2102-GMR
D2
RB168VWM-30TR
J4
2174507-2
D3
D4
SMBJ5.0CA-Q
No Connect
R19
470
TXD_USB
RXD_USB
C14
0.1u
C13
10u
C15
10u
C16
10u
Q1
Q2
MMSS8050-H-TP
R15
12k
R17
12k
DTR
RTS
R20
R21
R22
R23
0
TXD_UC
RXD_UC
IO13
IO15
CTS

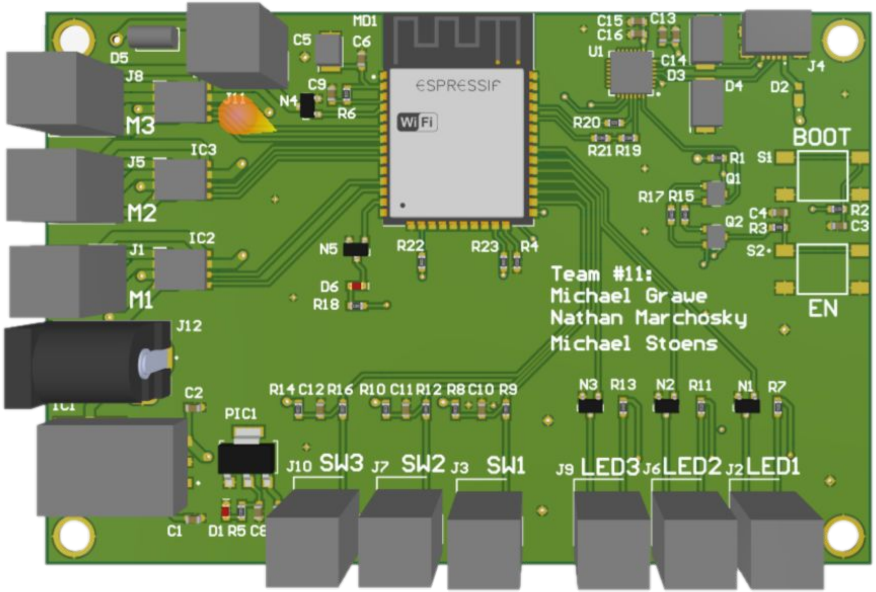
PCB Design

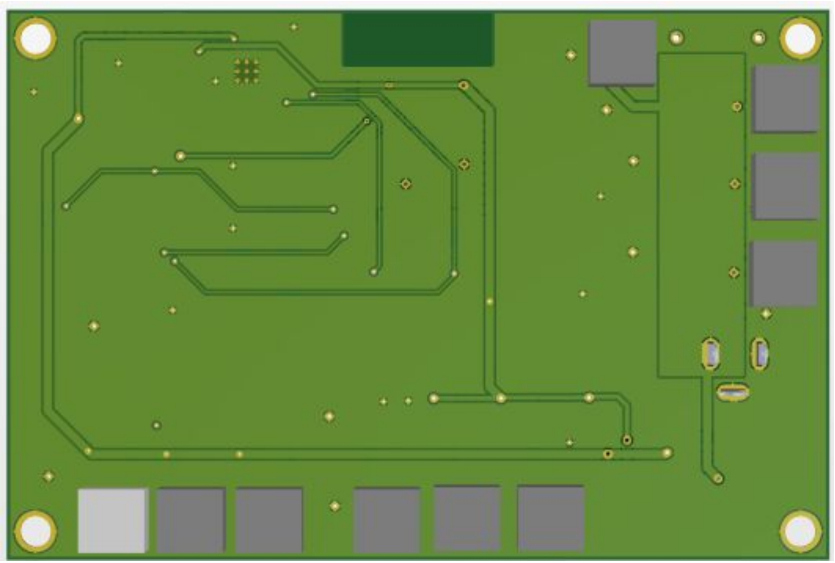

2 GND
D5
MD1
C5
C6
C15
C16
C13
U1
C14
D3
D4
D2
J4
J8
IC4
J11
N4
C9
R6
M3
R20
R21 R19
BOOT
R1
S1
J5
IC3
M2
Q1
R17 R15
C4
R3
R2
C3
Q2
S2
J1
IC2
M1
N5
R22
R23
R4
D6
R18
Team #11:
Michael Grawe
Nathan Marchosky
Michael Stoens
EN
J12
IC1
C2
PIC1
R14 C12 R16 R10 C11 R12 R8 C10 R9
N3
R13
N2
R11
N1
R7
4 GND
J10
SW3
J7
SW2
J3
SW1
J9
LED3
J6
LED2
J2
LED1
C1
D1 R5 C8 C7


2 GND


2 GND
4 GND

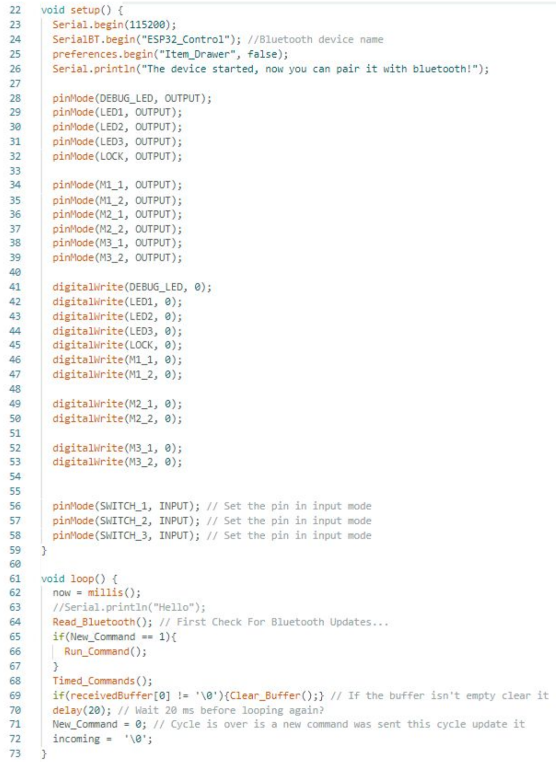
PCB Design


M3
M2
M1
J8
J5
J1
J12
IC3
IC2
IC1
D5
C5
C6
MD1
C9
N4
R6
J11
ESPRESSIF
Wi Fi
C15
C16
C13
U1
C14
D3
D4
D2
J4
R20
R21 R19
BOOT
R1
S1
Q1
R17 R15
Q2
C4
R3
R2
C3
S2
EN
N5
R22
R23
R4
D6
R18
Team #11:
Michael Grawe
Nathan Marchosky
Michael Stoens
C2
PIC1
R14 C12 R16 R10 C11 R12 R8 C10 R9
N3 R13 N2 R11 N1 R7
J10 SW3 J7 SW2 J3 SW1
J9 LED3 J6 LED2 J2 LED1
C1
D1 R5 C8



Design Process: FW Development

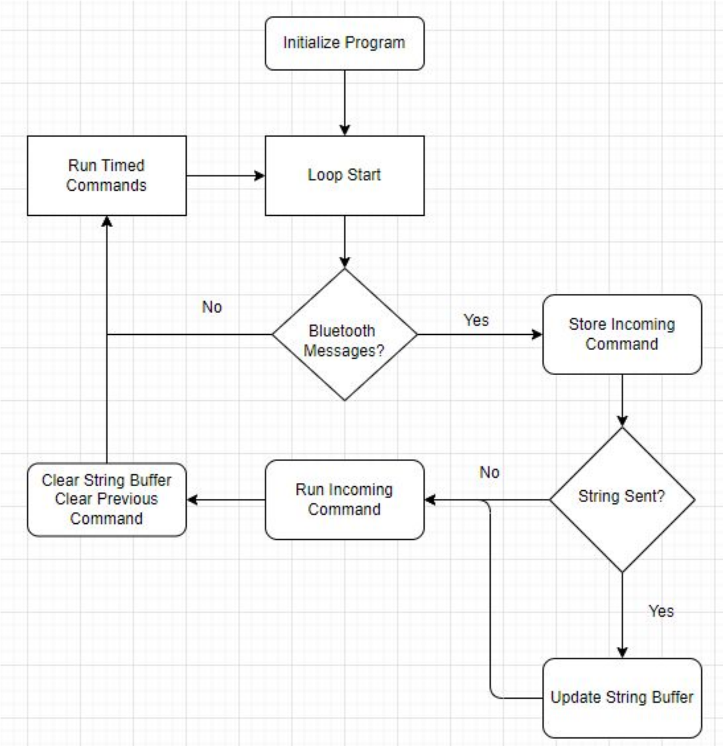
Arduino IDE



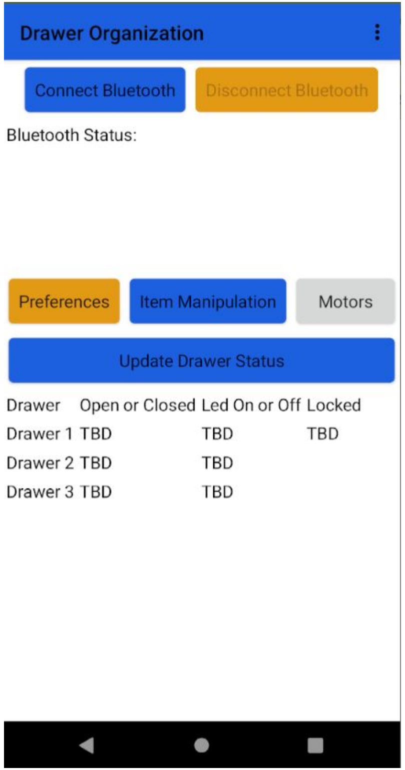
```
void setup() {
  Serial.begin(115200);
  SerialBT.begin("ESP32_Control"); //Bluetooth device name
  preferences.begin("Item_Drawer", false);
  Serial.println("The device started, now you can pair it with bluetooth!");

  pinMode(DEBUG_LED, OUTPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LOCK, OUTPUT);

  pinMode(M1_1, OUTPUT);
  pinMode(M1_2, OUTPUT);
  pinMode(M2_1, OUTPUT);
  pinMode(M2_2, OUTPUT);
  pinMode(M3_1, OUTPUT);
  pinMode(M3_2, OUTPUT);

  digitalWrite(DEBUG_LED, 0);
  digitalWrite(LED1, 0);
  digitalWrite(LED2, 0);
  digitalWrite(LED3, 0);
  digitalWrite(LOCK, 0);
  digitalWrite(M1_1, 0);
  digitalWrite(M1_2, 0);

  digitalWrite(M2_1, 0);
  digitalWrite(M2_2, 0);

  digitalWrite(M3_1, 0);
  digitalWrite(M3_2, 0);


  pinMode(SWITCH_1, INPUT); // Set the pin in input mode
  pinMode(SWITCH_2, INPUT); // Set the pin in input mode
  pinMode(SWITCH_3, INPUT); // Set the pin in input mode
}

void loop() {
  now = millis();
  //Serial.println("Hello");
  Read_Bluetooth(); // First Check For Bluetooth Updates...
  if(New_Command == 1){
    Run_Command();
  }
  Timed_Commands();
  if(receivedBuffer[0] != '\0'){Clear_Buffer();} // If the buffer isn't empty clear it
  delay(20); // Wait 20 ms before looping again?
  New_Command = 0; // Cycle is over is a new command was sent this cycle update it
  incoming =  '\0';
}
```

Design Process: FW Development

App ESP32 Interaction

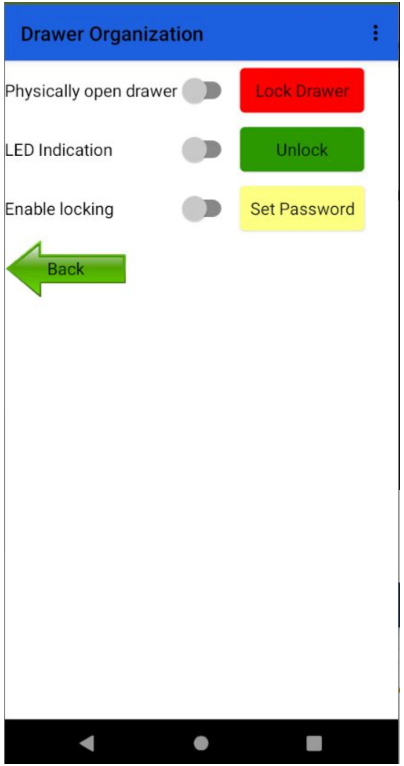


App Development



Bluetooth Status:

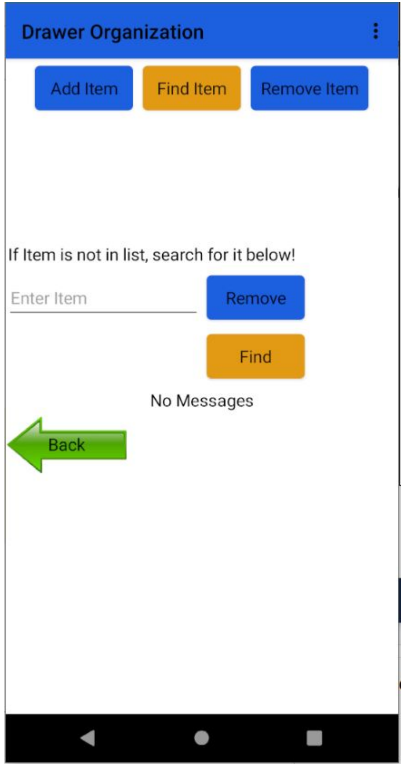
Drawer	Open or Closed	Led On or Off	Locked
Drawer 1	TBD	TBD	TBD
Drawer 2	TBD	TBD	
Drawer 3	TBD	TBD	



Physically open drawer ○

LED Indication ○

Enable locking ○

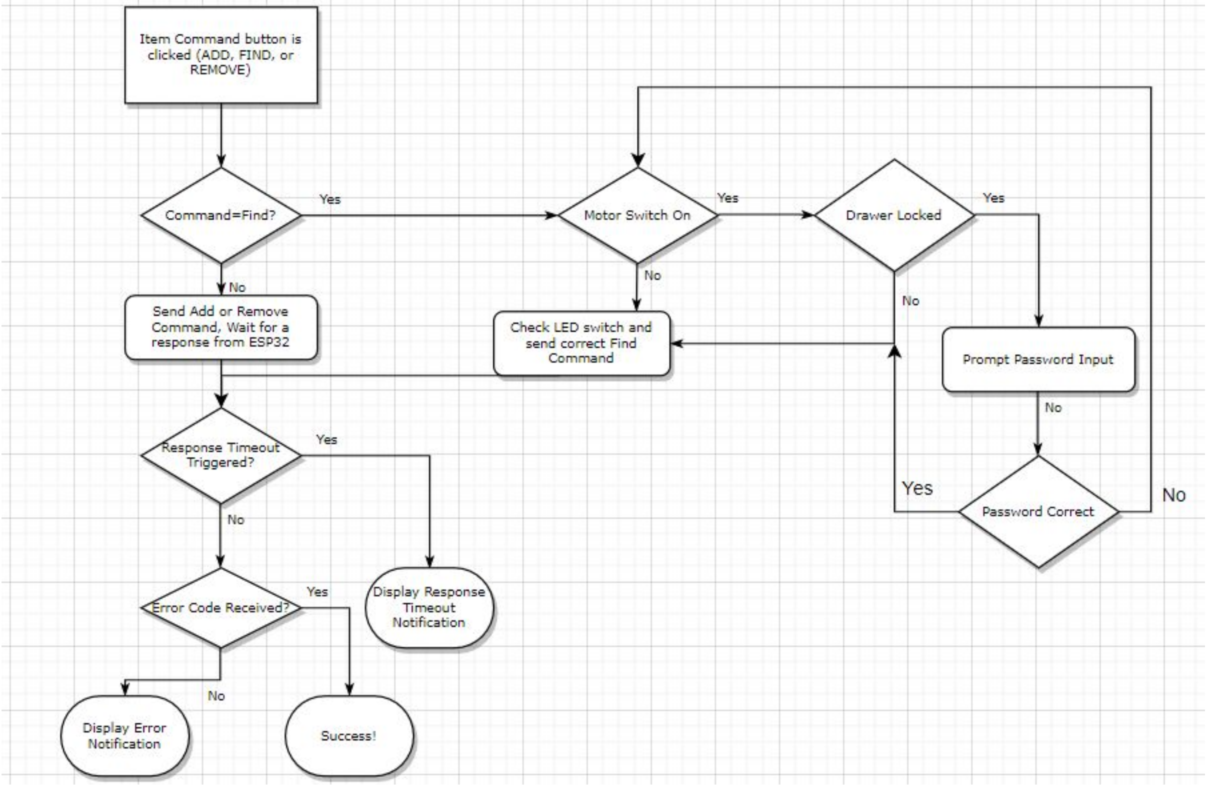


If Item is not in list, search for it below!

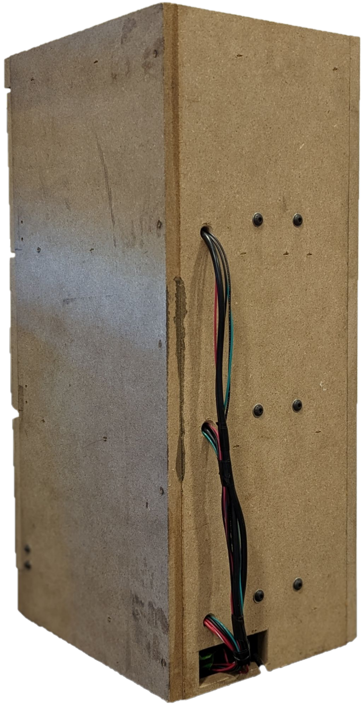
Enter Item

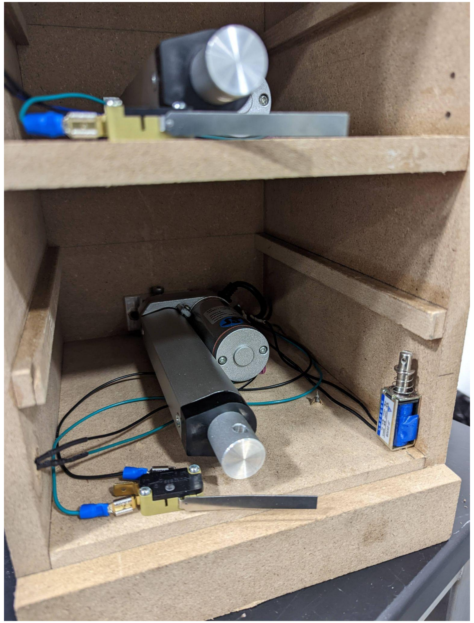
No Messages

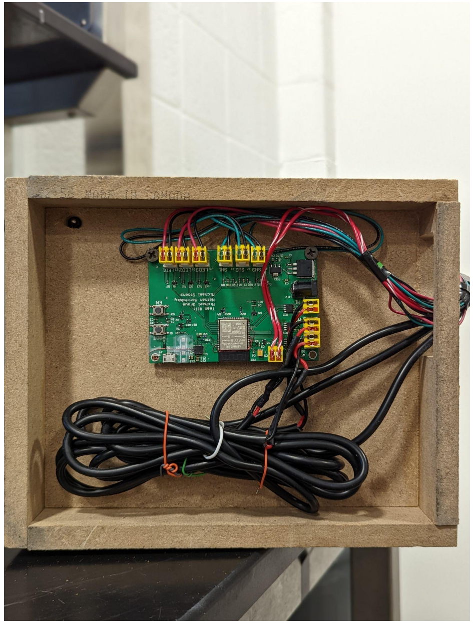
App Development: Flowchart


Item Command button is clicked (ADD, FIND, or REMOVE)
Command=Find?
Yes
No
Send Add or Remove Command, Wait for a response from ESP32
Motor Switch On
Yes
No
Check LED switch and send correct Find Command
Drawer Locked
Yes
No
Prompt Password Input
No
Password Correct
Yes
No
Response Timeout Triggered?
Yes
No
Display Response Timeout Notification
Error Code Received?
Yes
No
Display Error Notification
Success!

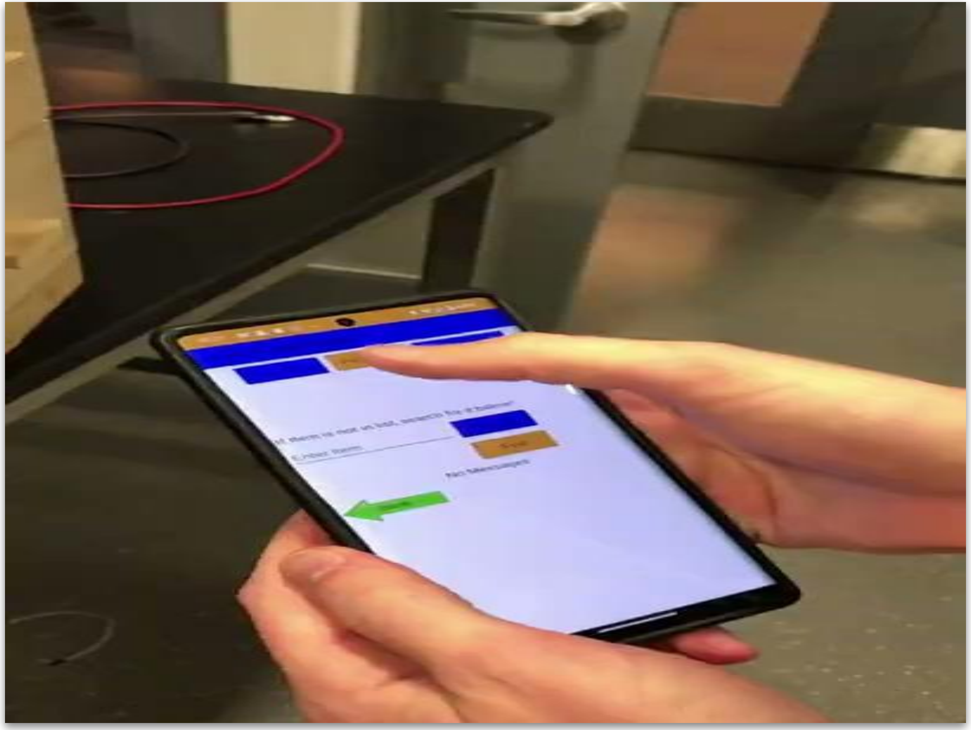
Design Process: Putting it all Together







Demonstration



Quantitative Results

Parameter	Desired	Actual
Radius of Bluetooth Connection	>21 feet (for size of the average room)	41.2 feet - SUCCESS
Latency between App Command and Drawer Reaction	<1 second	<1 second, measured by stopwatch -SUCCESS
Per Component Current Draw	<0.75A per motor <1A for solenoid lock	0.6A average per motor 0.72A for solenoid - SUCCESS

Successes and Challenges

- Overall, project was successful
  - All high level requirements met, drawers are fully functional with app
  - Final Product looks complete and could be used as is
- Challenges:
  - Overcurrent Drawn when motors switch direction
  - Severed Bluetooth Connection when changing screens on Android App
  - Incorrect Labeling of GPI pins as GPIO pins
  - Could Not Program ESP32 due to improper pin connection of USB-UART chip

Key Takeaways

- Thinking critically about mechanics and ease of implementation (How to build the drawers)
- Working as a team to come up with ideas for debugging hardware issues (More systematic debugging)
- Applying information learned in various hardware and software ECE courses in creative ways

Further Work and Improvements

- Increase size of drawer unit
- Add more drawers
- Increase speed of motors
- Add retraction capability to drawers
    - Using a weak magnet attached to the push-point on drawer, as well as push-point on motor, retraction could be achieved while still allowing manual operation

Acknowledgements/Q&A

?