



OUTDOOR SMART DOG FEEDER

Electrical & Computer Engineering

Team 14: Lucas Duduit, T'Andra Newby, Haojiang Shi

April 30th, 2024

WHY DO WE NEED IT?

Feeding dogs on a timely manner can be hard. Overeating can occur if feeding amount is not controlled. Most dog feeders are indoors only.

OBJECTIVE

Desired amount of food will be accurately dispensed at feed time.

The door will open at feed time when dog is within the range of the RFID communication. The feeder will operate outdoors without supplying power from the household.

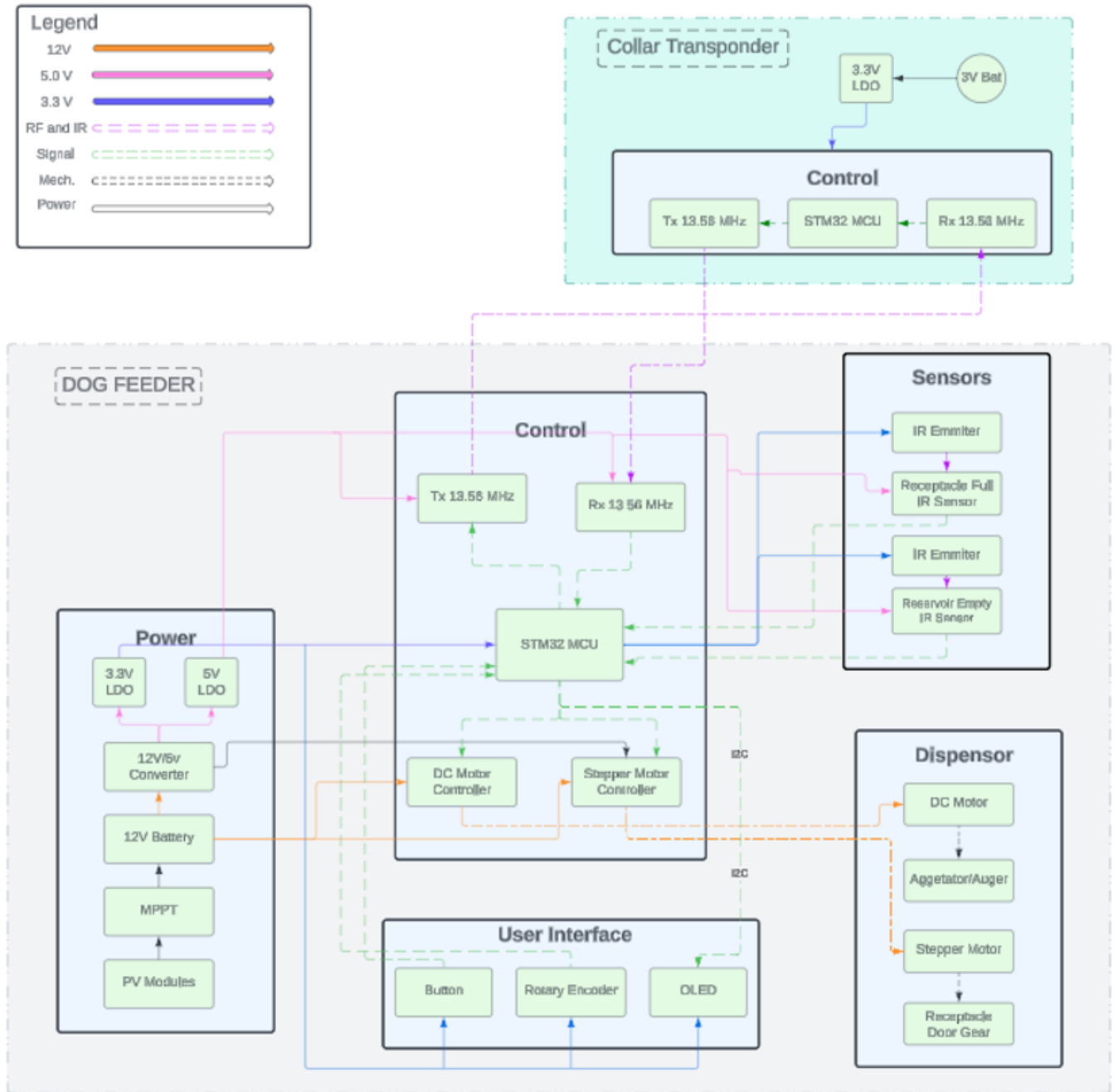
HIGH LEVEL REQUIREMENTS

- Active RFID Communication
- Network Control
- Power Management

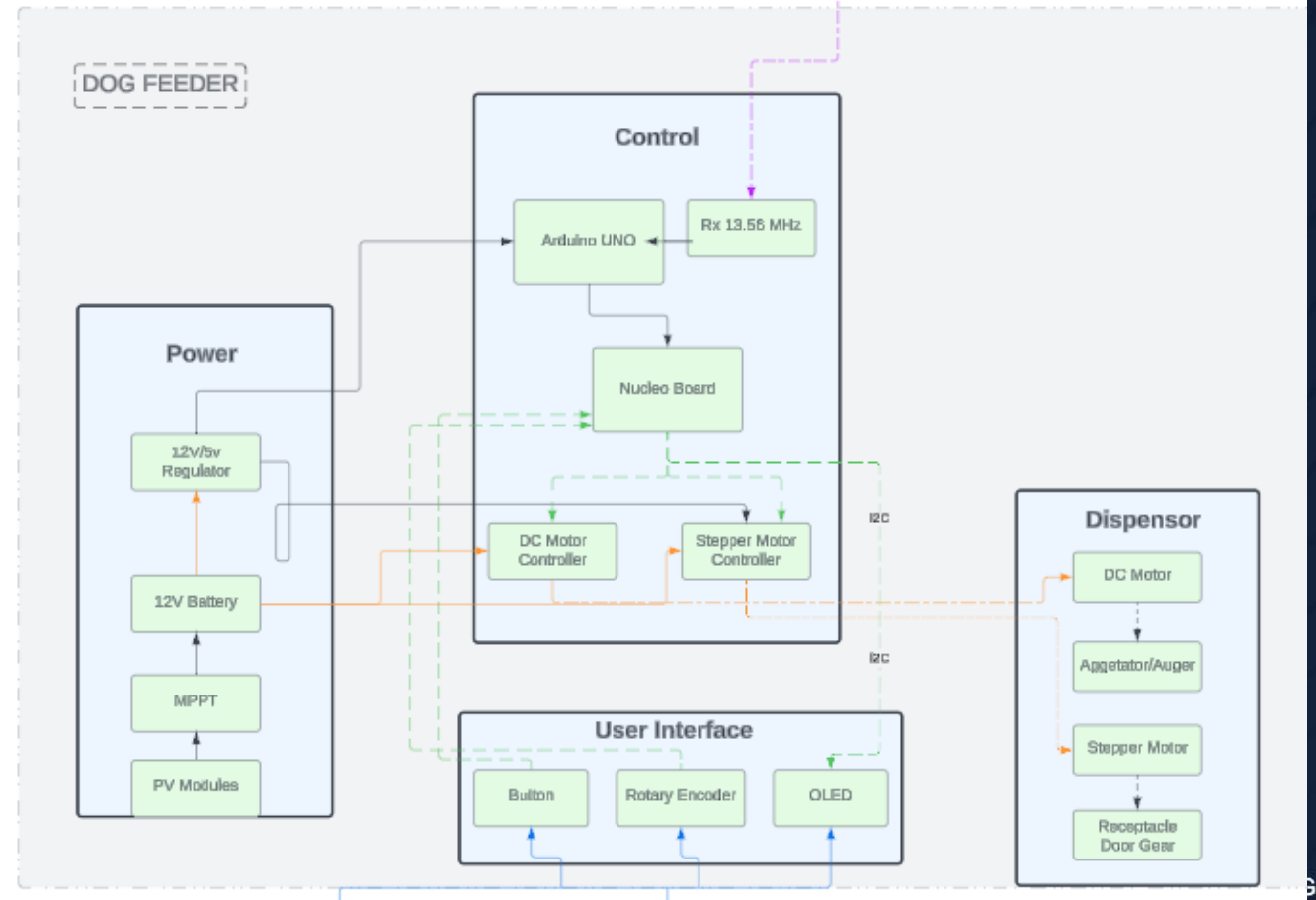
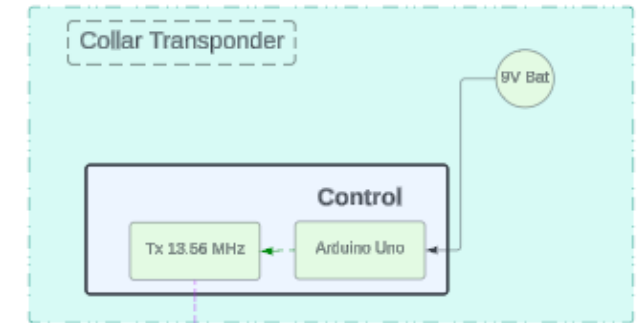
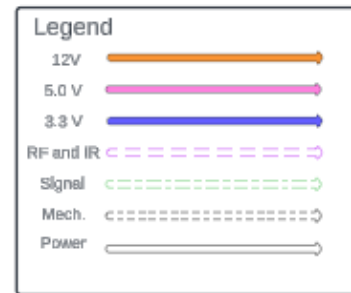


EVOLUTION OF DESIGN

INITIAL DESIGN



FINAL DESIGN



INITIAL DESIGN

ELECTRICAL & COMPUTER ENGINEERING



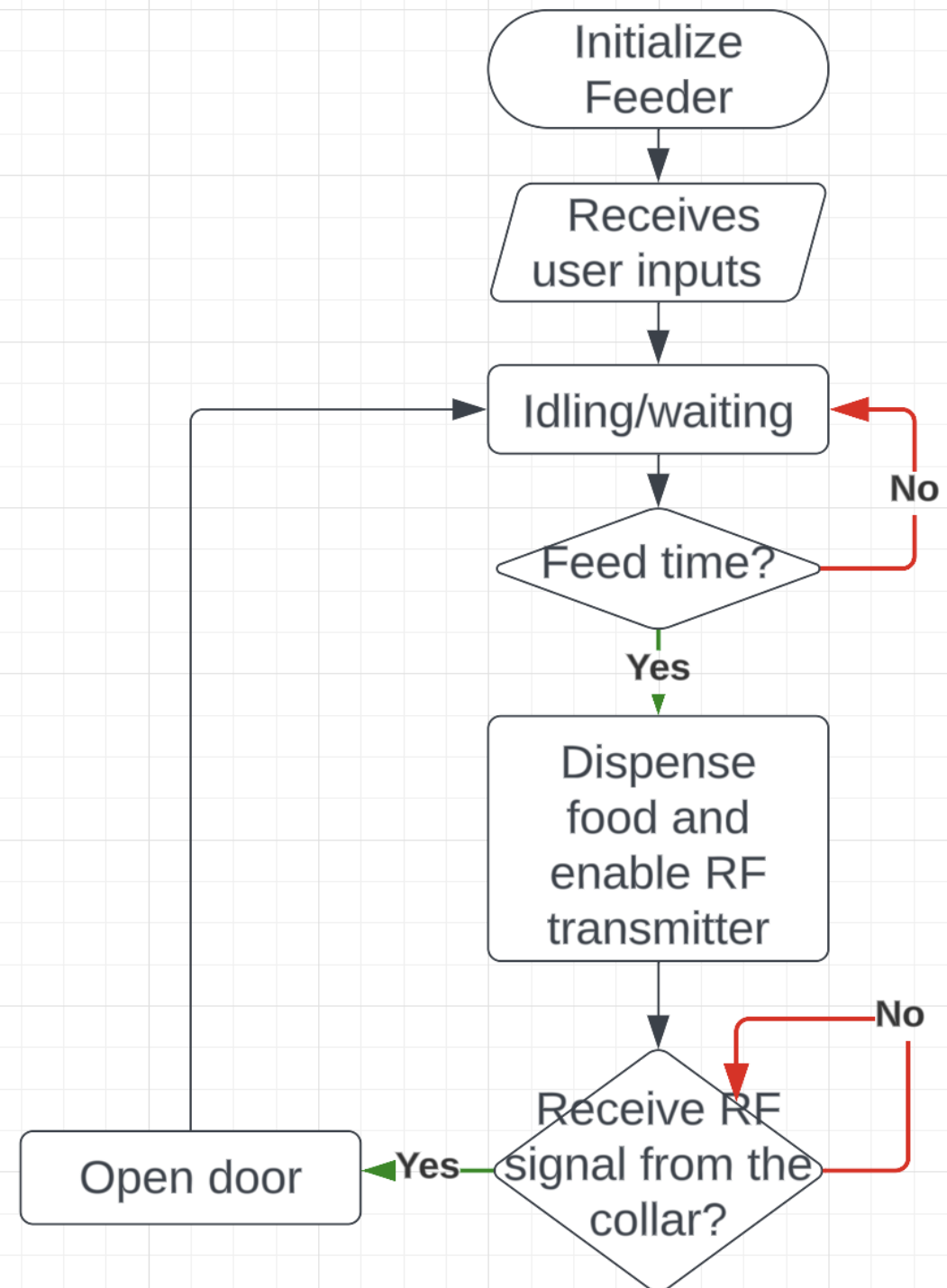
FINAL BUILD



POWER MANAGEMENT

POWER USE AND PRODUCTION		ENERGY PER DAY
IDLE (24hr)	1.8 W	2.59 kJ
RUNNING (max use)	18.5 W	1.11 kJ
SOLAR (8 hr)	10 W	4.8 kJ

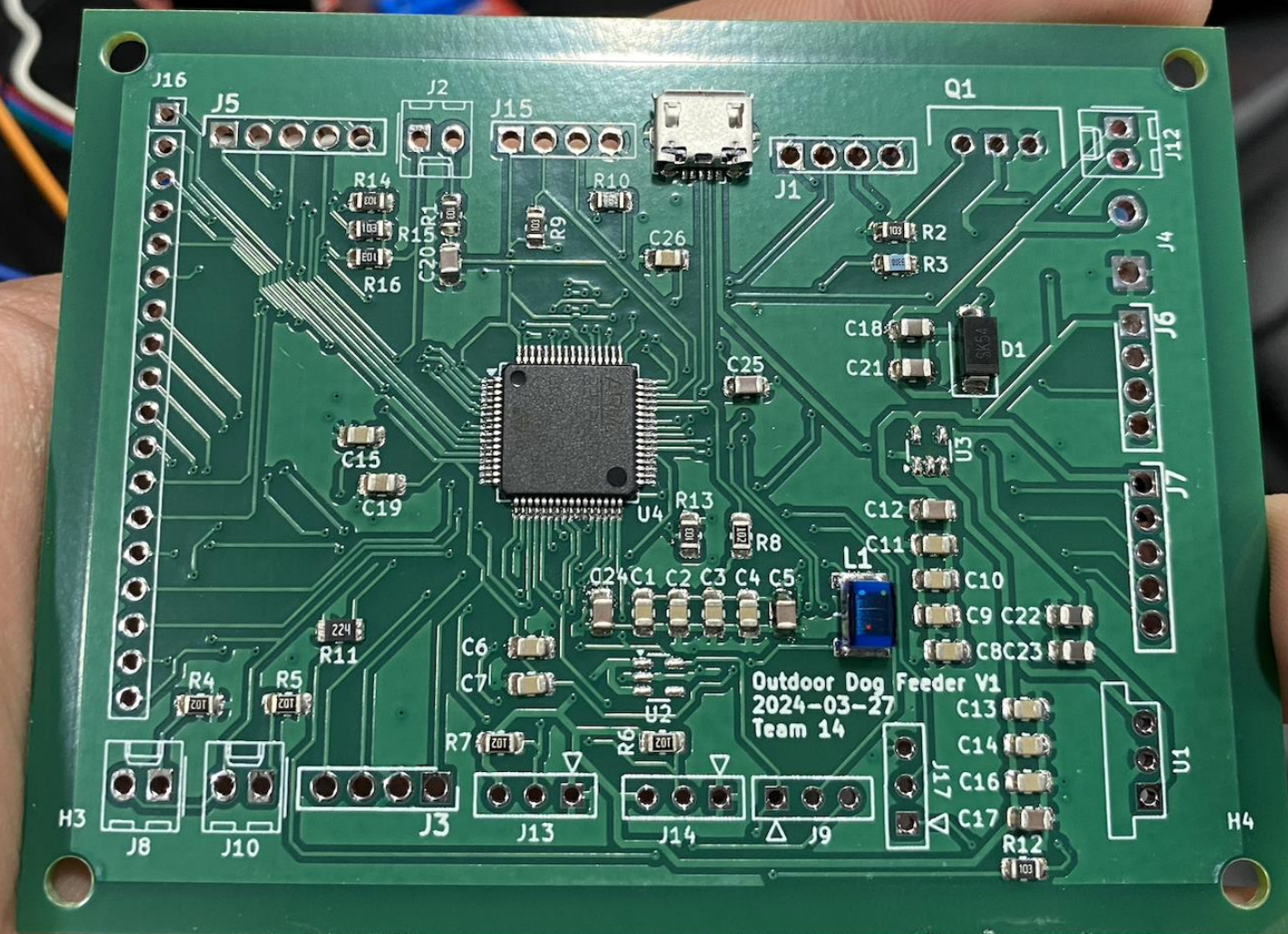
CONTROL FLOWCHART



A photograph of a concrete staircase with a metal handrail, set against a concrete wall. A white, three-dimensional zigzag line is mounted on the wall, following the path of the stairs. The right side of the image is overlaid with a dark blue background featuring a diagonal line pattern.

CHALLENGES

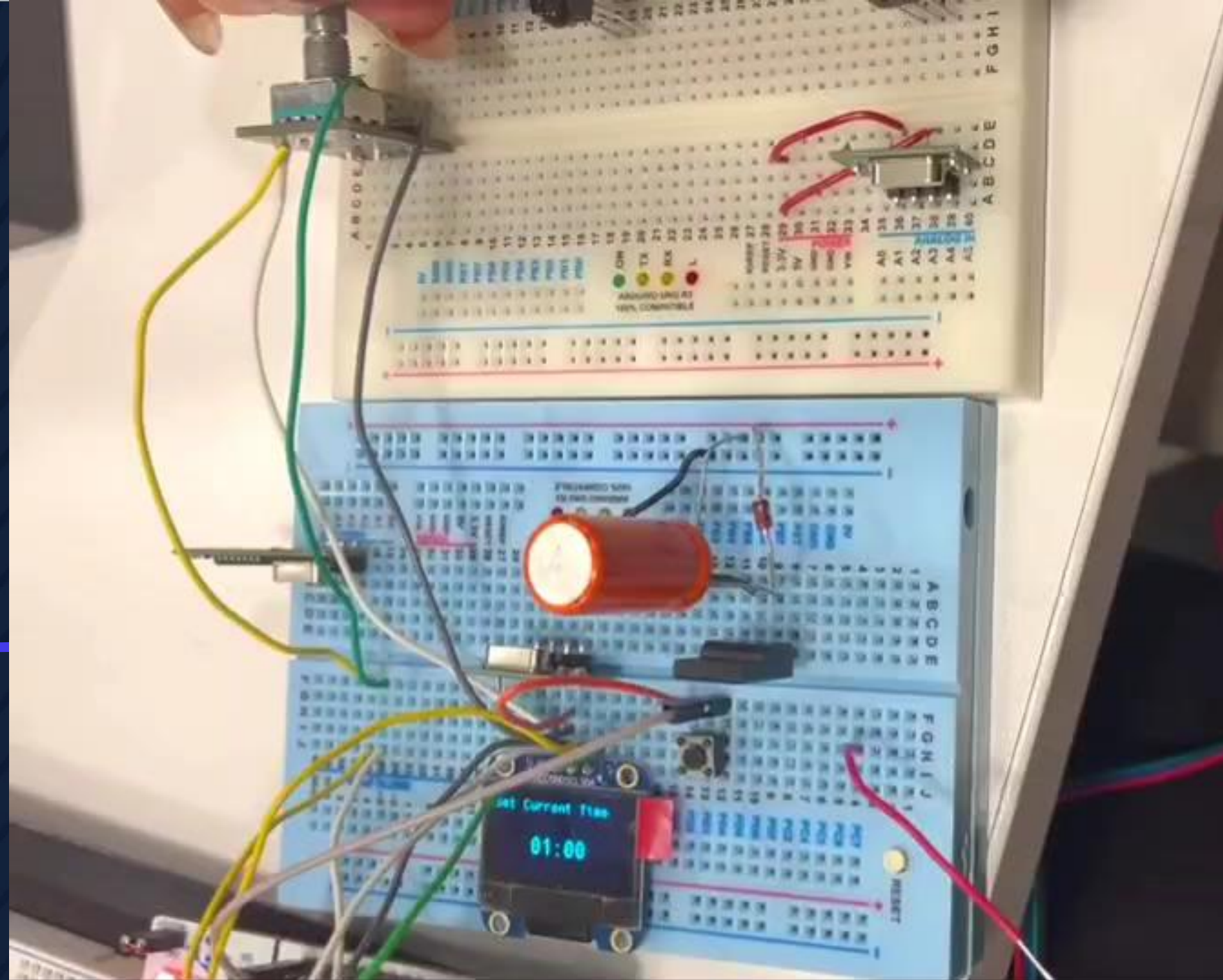
PCB



A composite graphic on the left side of the slide. It features a concrete staircase with a metal handrail, overlaid with a white zigzag line that follows the path of the stairs. The background of the entire slide is dark blue with diagonal lines.

RESULTS

USER INTERFACE



RFID PROGRAMING

```
// Include libraries
#include <RH_ASK.h>
#include <SPI.h>

const int FEED_TIME = 2;
const int OPEN_LID = 3;

// Msg 2 size expected
const byte tamMsg2 = 4;

// Create ASK objects (Tx on pin 12 & Rx on pin 11)
RH_ASK driver;

void setup() {
  Serial.begin(9600);
  // Initialize object
  driver.init();
  pinMode(FEED_TIME, INPUT);
  pinMode(OPEN_LID, OUTPUT);
  digitalWrite(OPEN_LID, LOW);
}

void loop() {
  if(digitalRead(FEED_TIME)){ //pin is high from stm 32
    // Send message
    const char *msg1 = "RANGE";
    driver.send((uint8_t *)msg1, strlen(msg1));
    driver.waitPacketSent();
    Serial.print("Msg sent: ");
    Serial.println(msg1);
    delay(50);
  }
}
```

```
// Receive msg from board 2
uint8_t msg2[tamMsg2];
uint8_t msg2Len = sizeof(msg2);

bool listened = false;
int t = 0;
while ((! listened) && (t < 15)) {
  listened = driver.recv(msg2, &msg2Len);
  t++;
  delay(50);
}
if (listened) {
  if(msg2[0] == '0' && msg2[1] == 'P'){
    Serial.print("YAY\n");//set pin high for opening lid t
    digitalWrite(OPEN_LID, HIGH);
    // delay(1000);//idk how long
    // digitalWrite(OPEN_LID, LOW);
  }
  delay(500);
} else {
  Serial.print("NOTHING");
  digitalWrite(OPEN_LID, LOW);
}
}
else{
  digitalWrite(OPEN_LID, LOW);
}
}
```


CONCLUSION

- Achievements and Failures
- Lessons Learned
- Future Direction



QUESTIONS?



THANK YOU!