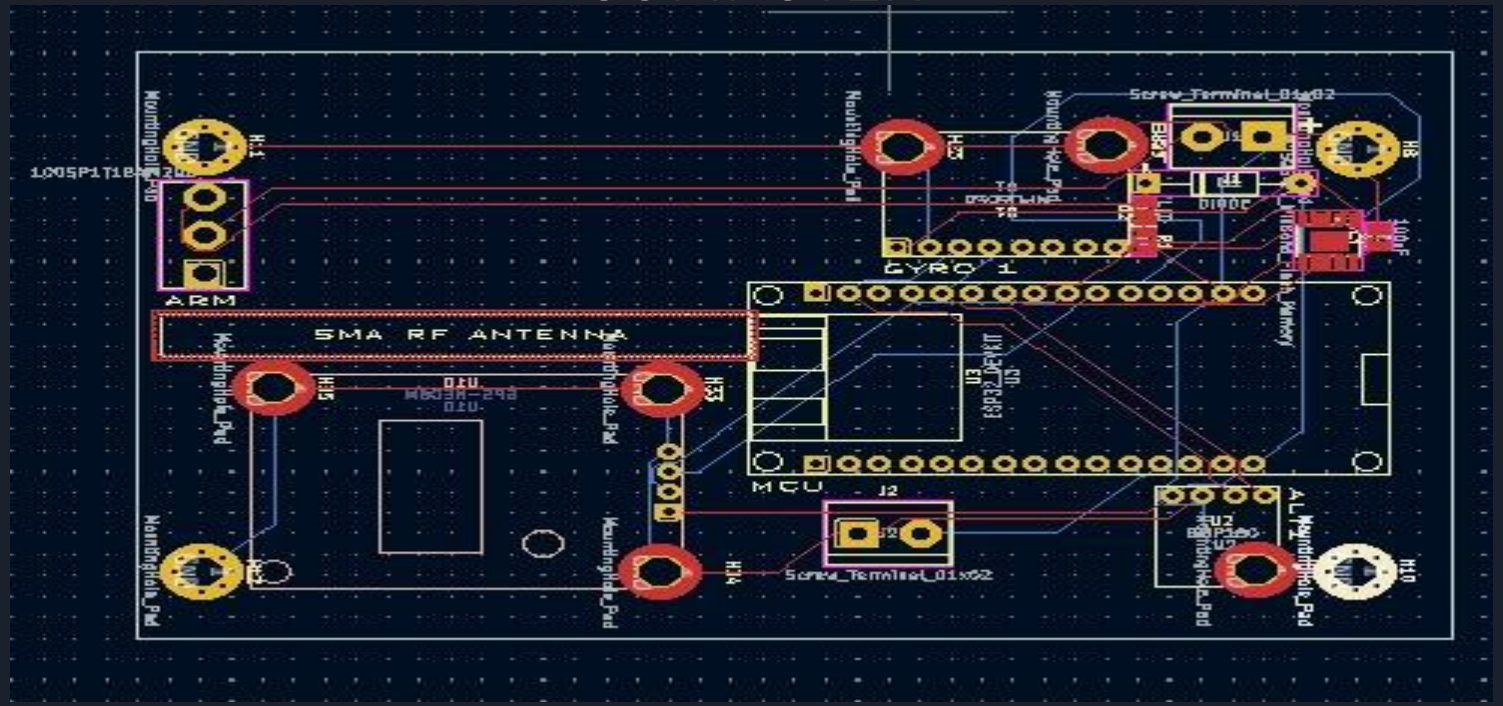
RECOVERY TEAM PRESENTATION (B2)

WEEK 3

TASKS ACCOMPLISHED

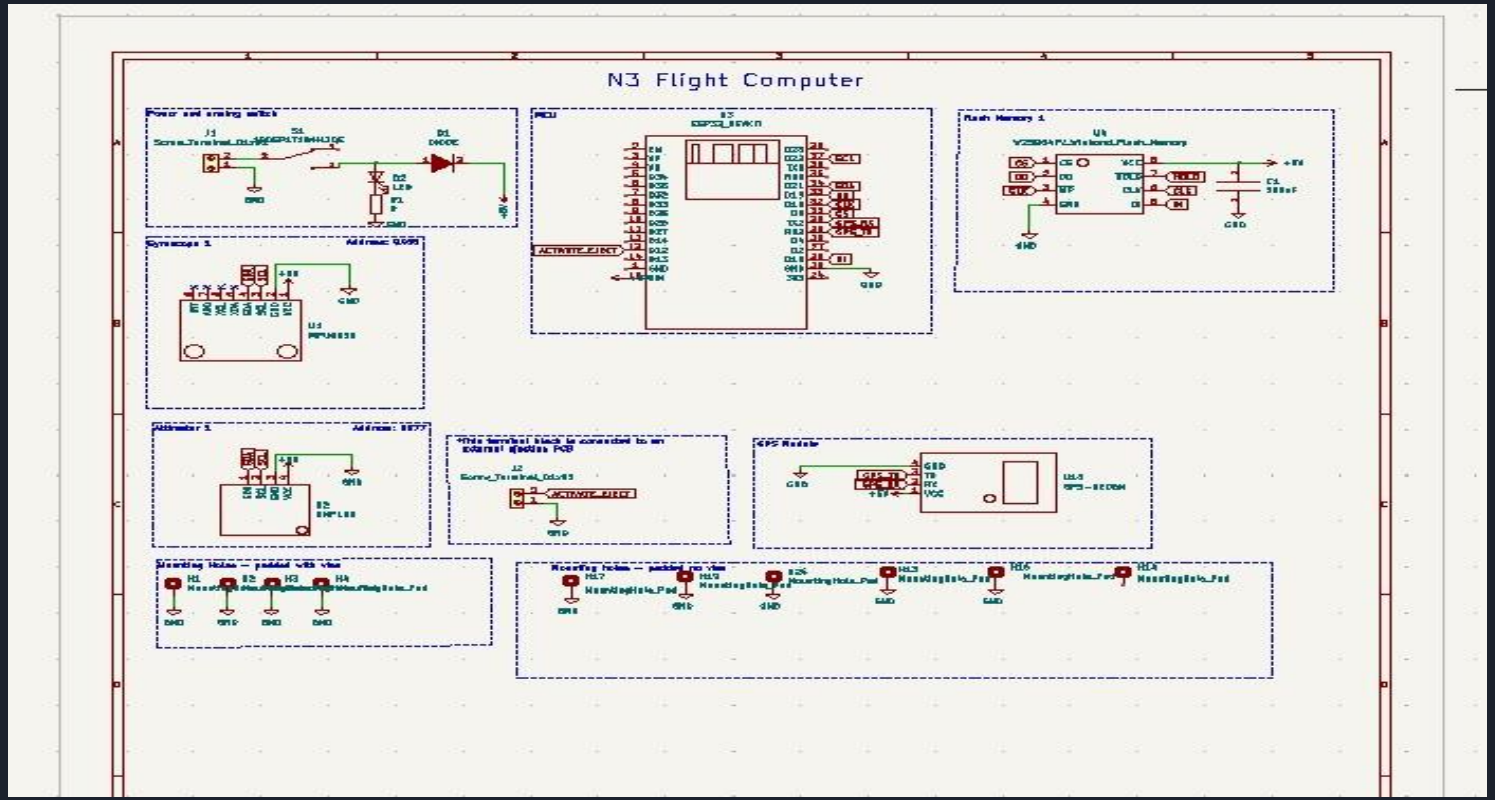
- Acquiring components to make the Flight Computer.
- Implementing Over The Air(OTA) updates.
- Finished fabricating the steel piston chamber and attaching the PVC end cap.
- Rectifying the code for the piston test.
- Creating a simple project board .

ACQUIRING COMPONENTS TO MAKE THE FLIGHT COMPUTER

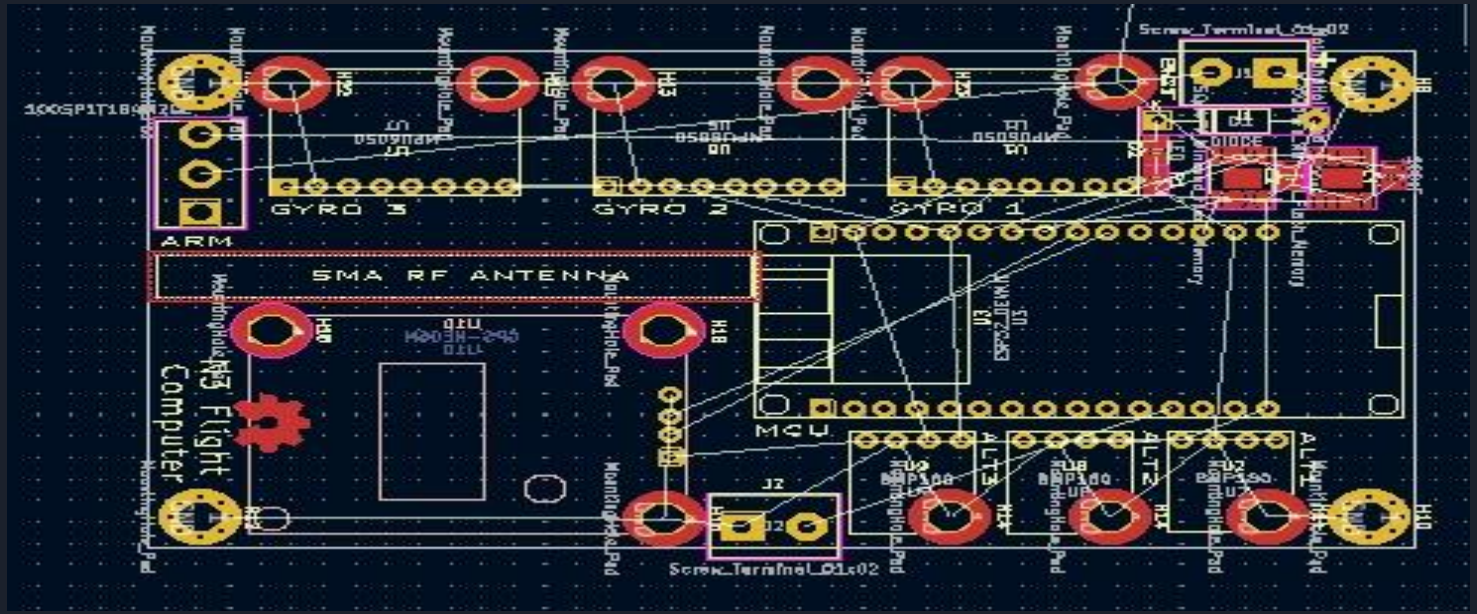


This the final PCB that we are using.

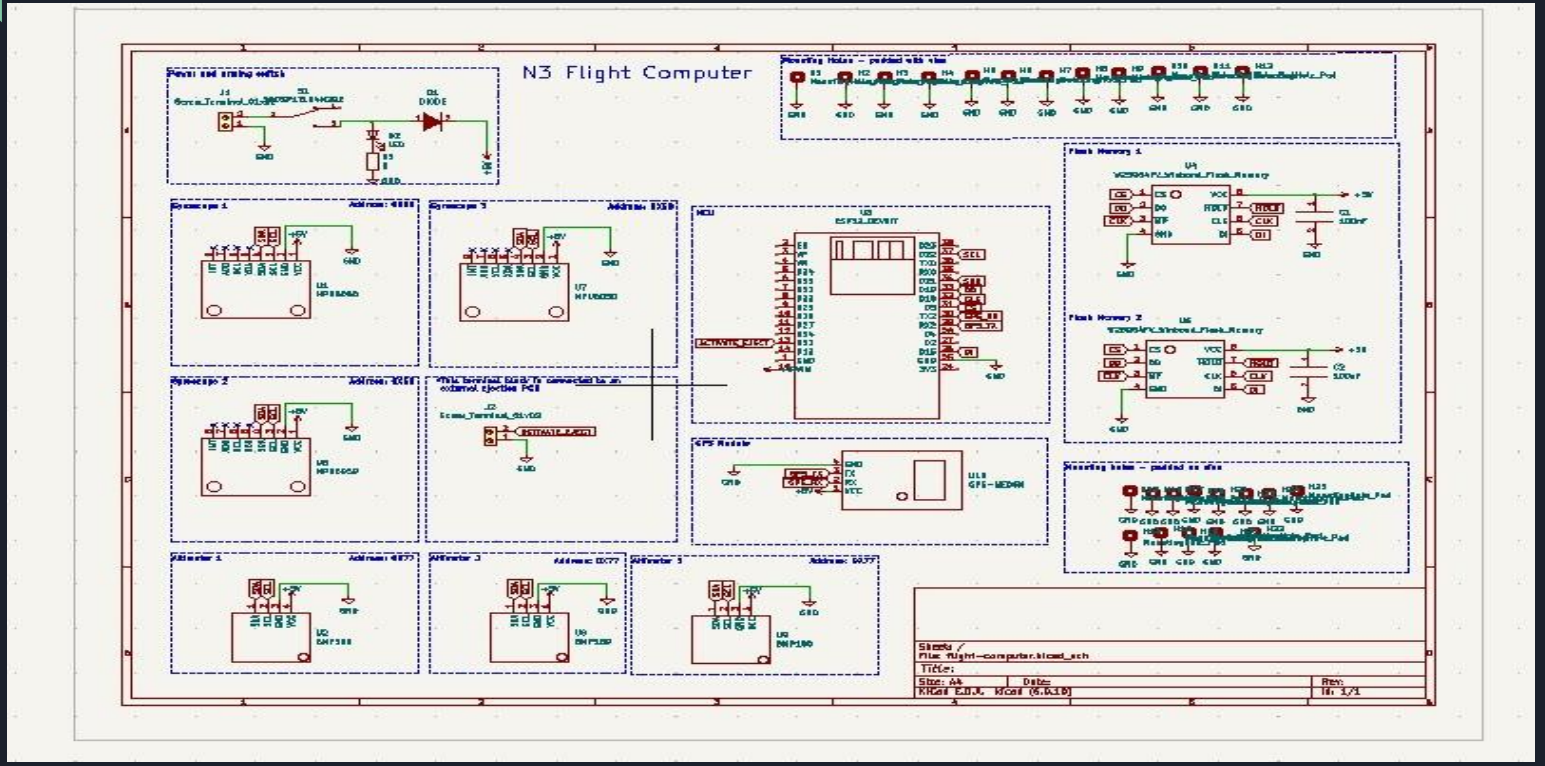
FINAL SCHEMATICS



INITIAL PCB


GYRO 3
GYRO 2
GYRO 1
ARM
SMA RF ANTENNA
MCU
Flight Computer
Screw_Terminal_01x02

These are the initial schematics.



OTA UPDATES IMPLEMENTATION

The code was added to both the ground station and Telemetry repository.

```
#include <WiFi.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>

//const char* ssid = "";
//const char* password = "";

void setupOTA() {

  // Port defaults to 3232
  // ArduinoOTA.setPort(3232);

  // Hostname defaults to esp3232-[MAC]
  ArduinoOTA.setHostname("Nakuja");

  // No authentication by default
  ArduinoOTA.setPassword("ProjectN3");

  // Password can be set with it's md5 value as well
  // MD5(admin) = 21232f297a57a5a743894a0e4a801fc3
  //ArduinoOTA.setPasswordHash("21232f297a57a5a743894a0e4a801fc3");
```

```
  ArduinoOTA
    .onStart([]() {
      String type;
      if (ArduinoOTA.getCommand() == U_FLASH)
        type = "sketch";
      else // U_SPIFFS
        type = "filesystem";

      // NOTE: if updating SPIFFS this would be the place to unmount SPIFFS using SPIFFS.end()
      Serial.println("Start updating " + type);
    })
    .onEnd([]() {
      Serial.println("\nEnd");
    })
    .onProgress([](unsigned int progress, unsigned int total) {
      Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
    })
    .onError([](ota_error_t error) {
      Serial.printf("Error[%u]: ", error);
      if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
      else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
      else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
      else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
      else if (error == OTA_END_ERROR) Serial.println("End Failed");
    });

```

This will be tested once the Flight computer is fabricated.

```
  ArduinoOTA.begin();

  Serial.println("Ready");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}

void handleOTA() {
  ArduinoOTA.handle();
}
```

PISTON PROGRESS

- Piston test software finalized
- Piston chambers finalized

- Weight comparison


SF-430
174


SF-400
230
CAPACITY:

- Additional crimson powder prepared

RECTIFYING THE CODE FOR THE PISTON TEST

Previously the user specified a delay time which caused a pause in the entire running of the code

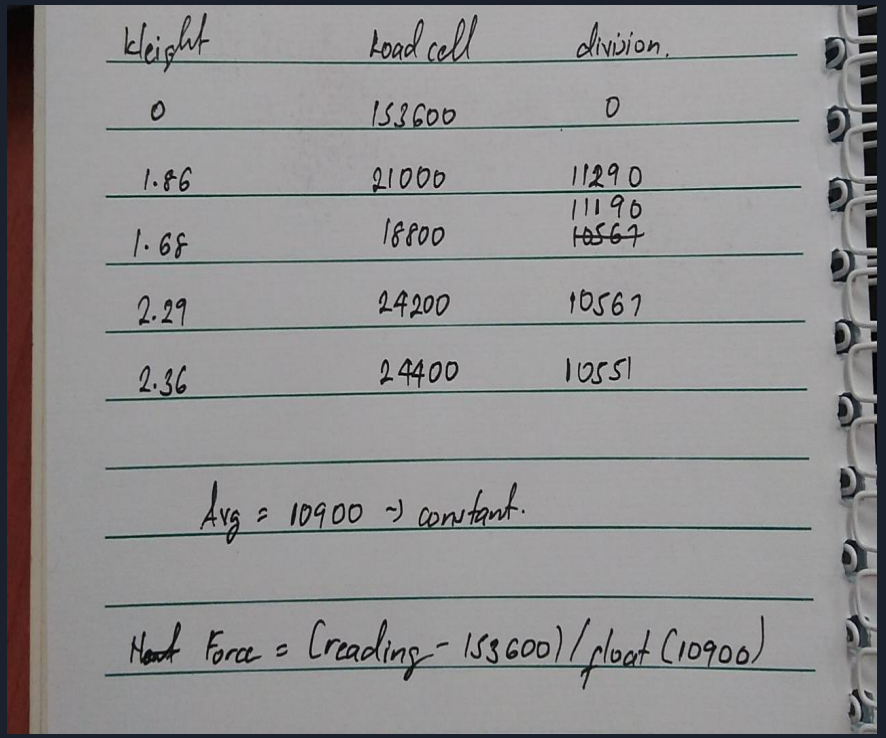
The nichrome wire is turned off once the load cell reads a value greater than 0

This also enabled us to determine the time between ignition command and firing of the crimson powder.

```
20  int real_average_reading, raw_average_reading;
21
22  bool led_state = 0;
23  const int led_pin = 14;
24  int nichrome_high_time = 0;
25  String dummy_force = "700";
26
27  void connect_to_wifi(){
28    // Connect to Wi-Fi
29    WiFi.begin(SSID, PASSWORD);
30    while (WiFi.status() != WL_CONNECTED) {
31      delay(1000);
32      debugln("[+]Connecting to WiFi..");
33    }
34
35    // Print ESP Local IP Address
36    debugln(WiFi.localIP());
37  }
38
39  /* Load cell functions */
40  HX711 load_cell;
41
42  void setup_load_cell(){
43    load_cell.begin(load_cell_dt, load_cell_sck);
```

CALIBRATION OF THE LOAD CELL

We carried out tests that gave us these results.



CHALLENGES FACED THIS WEEK

- Two ESP32 microcontrollers fried.
- We were unable to acquire all the components so as to make the flight computer hence the etching has been pushed to next week.
- EasyEDA doesn’t have the footprints , hence they are to be made from scratch.

NEXT WEEK'S TASKS

- Finishing the piston test
- Determining the ejection algorithm
- Fabricating of the Flight Computer PCB