**Rocket Flight Controller**

This repository contains the design files and documentation for a custom-designed Printed Circuit Board (PCB) serving as a flight controller for a rocket. The PCB integrates various sensors, communication modules, and control mechanisms, all managed by a central microcontroller.

**Features**

* **Microcontroller Slot:** Dedicated footprint for an ESP32-WROOM-32 microcontroller, enabling comprehensive control over all integrated components.
* **BMP280 Sensor Integration:** Onboard support for the BMP280 barometric pressure sensor, connected via I2C protocol for altitude and atmospheric data acquisition.
* **GNSS L89 Module Support:** Dedicated footprint for the L89 GNSS module, connected via UART protocol for precise location and velocity tracking.
* **MicroSD Card Slot:** Included slot for a microSD card, interfaced with the microcontroller via SPI protocol for data logging during flight.
* **XBee S2C Pro Integration:** Footprint for an XBee S2C Pro RF communication module, facilitating real-time data transmission with configurable band, sample rate, and frequency.
* **MOSFET Switching Circuits(IRF 540 N):** Two MOSFET-based switching circuits are incorporated, controlled by the microcontroller, to enable two distinct switching activities during flight (e.g., parachute deployment, ignition).
* **Voltage Control and Protection:** Resistors (100kΩ and 10kΩ connected in parallel for voltage division) are strategically placed to protect MOSFETs and ensure proper voltage levels at outputs.
* **Compact Circular Design:** The PCB is designed in a circular form factor suitable for integration into a rocket body.

**Block Diagram**

A[ESP32-WROOM-32 Microcontroller] --> B(BMP280 Sensor - I2C)

A --> C(GNSS L89 Module - UART)

A --> D(MicroSD Card - SPI)

A --> E(XBee S2C Pro - RF Communication)

A --> F1(MOSFET Circuit 1)

A --> F2(MOSFET(IRF 540 N) Circuit 2)

F1 --> G1(Output 1 - e.g., Parachute)

F2 --> G2(Output 2 - e.g., Ignition)

E --> H(Ground Station / Receiving Unit)

D --> I(Data Logging)

**PCB Design**

The PCB was designed using Proteus for schematic capture and layout, and further verified with Cadence tools. Simulations were performed in Wokwi prior to PCB fabrication.

**Top Layer (2D Render)**

This image shows the top silkscreen and component placement as rendered in the design software, highlighting the ESP32, XBee, and BMP280 module footprints.

**Bottom Layer (2D Render)**

This render displays the bottom side of the PCB, showing the footprints for the GNSS L89 module and the SD card slot, along with the resistors for voltage control.

**Physical Bottom View**

A photograph of the fabricated PCB's bottom side, clearly showing the SD card slot and GNSS L89 module area.

**Layer View (Routing)**

This image provides a detailed view of the PCB layers, illustrating the routing of traces and connections between various components and protocols (I2C, UART, SPI).

**Assembled PCB**

A photograph of the fully assembled PCB, with the microcontroller, XBee module, and other components mounted, ready for integration.

**Schematics**

The complete schematics for this flight controller, designed in [Proteus Design Suite](https://www.labcenter.com/) and verified with [Cadence Design Systems](https://www.cadence.com/), will be provided here. This includes detailed circuit diagrams for the microcontroller connections, sensor interfaces, communication modules, MOSFET switching circuits, and power management.

**Simulation**

Prior to PCB fabrication, extensive simulations were conducted using Wokwi to validate the circuit functionality and component interactions. This section will include links to the Wokwi projects and any relevant simulation results.

* [Wokwi Simulation Link 1](https://www.google.com/search?q=https://wokwi.com/your-project-link-1) (Placeholder)

**Bill of Materials (BOM)**

A comprehensive list of all components required to assemble this flight controller PCB, including part numbers, manufacturers, and quantities.

* BOM.csv (Placeholder for CSV file)

**Firmware**

The firmware for the ESP32 microcontroller, written in Arduino/C++, will be located in this section. It will include code for:

* Reading data from BMP280 (I2C)
* Interfacing with GNSS L89 (UART)
* Logging data to MicroSD card (SPI)
* Controlling XBee S2C Pro for data transmission
* Managing MOSFET switching activities
* Flight state logic and data processing
* firmware/ (Placeholder for firmware source code directory)

**Getting Started**

Instructions on how to replicate this project, from ordering the PCB to flashing the firmware and testing the flight controller.

1. **Order PCB:** Use the Gerber files (to be provided) to order the PCB from your preferred manufacturer.
2. **Gather Components:** Refer to the BOM for a list of necessary components.
3. **Assemble PCB:** Solder all components onto the PCB.
4. **Flash Firmware:** Upload the provided firmware to the ESP32 microcontroller.
5. functionalities are working correctly.

**Schematic**