**General Overview of Control sections:**

Structural Overview:

The control subsystem is the main convergence point for all of the other subsystems in some form or another. Each sub system has data that can be taken into the esp32 processor for manipulation, or the esp32 can send signals to a particular sub system.

Communication overview:

[More in depth look at MQTT connection here]

The ESP32 has the ability to connect to WiFi and thus send data to the command website for display. This is facilitated by initiating WiFi connection, followed by connecting the ESP32 to an MQTT server. From here commands such as sub() and pub() can be used to subscribe and publish to the MQTT server for receiving and sending commands/data respectively.

**Drive subsystem:**

Motor system: Using an I2C connection to an H-Bridge, the ESP32 receives commands from the user app and then sends signals to the motors to perform different instructions, such as turning or moving.

**Optic Sensor system:**

**Diagram

Description automatically generated**The optic sensor is used in conjunction with another important component, that being the gyroscope. The gyroscope makes use of its ability to track rate of rotation to track the current angle of the rover (starting at 0 degrees). Using this angle, basic trigonometry can then be done to determine the direction of travel of the rover and in turn the X-Y coordinates. This is implemented as shown below: [optic sensor; one dimension of movement is shown, as it is all that is needed. Depending on angle the x/y coordinate changes.] The Chip Select is then set to active high at the end of the use of the optic sensor in order to initiate the SPI bus for the FPGA input, which is sent an active low at the same time.

**Gyroscope system:**

The gyroscope system is implemented with an SEN0142 MPU6050 purchased as an extra component at a cost of £x. It tracks the angle of the rover relative to the start angle. This is implemented by code that, if a threshold of rotation is detected (>=0.1rad/s) it will update the angle by multiplying the detected rotational speed by the sampling time of 0.1seconds, giving a fairly accurate angle track.

**Data processing:**

**FPGA data:**

The FPGA sends in data in the form of an SPI bus input to the esp32 through its SPI pins. This signal is then processed and converted to give the colour of the alien detected, the approximate pixel width and approximate angle (0 to 75 degrees). This is then processed further to give approximate distance, and then to calculate using the rover’s current position and angle relative to start the approximate location of the alien detected. An active HIGH is then sent to the Chip select to disable the FPGA’s input stream.

**Radar data:**

The radar data functions in much the same way. The radar will detect a fan under the ground. An approximate distance will be ascertained by basic trigonometry. This is then sent to the command.

**Battery data:**

The battery discharges voltage at a given value range between 0 and 3.3V (the maximum threshold of the esp32’s analogue read function). This is then converted into a percentage value and sent to command.

**Ultrasonic sensor:**

The ultrasonic sensor solves a specific problem to do with the detection of buildings. By giving the sensor code parameters to neglect detections near the boundaries of the arena, it is able to detect objects in front of it, and then will process them as being either an alien (if FPGA is currently detecting one) or as a building if the FPGA is not currently detecting anything in that area.

Diagram

Description automatically generatedPINOUT REFERENCE: Connection Diagram

|  |  |
| --- | --- |
| **PIN** | **USE** |
| GPIO 4 | OPTIC SENSOR, MOUSECAM\_RESET |
| GPIO 5 | OPTIC SENSOR, CHIP\_SELECT |
| GPIO 18 | OPTIC SENSOR, FPGA, SCK |
| GPIO 19 | OPTIC SENSOR, FPGA, MISO |
| GPIO 23 | OPTIC SENSOR, FPGA, MOSI |
| GPIO 3 | FPGA CHIP\_SELECT |
| GPIO 21 | GYROSCOPE, SDA |
| GPIO 22 | GYROSCOPE, SCL |
| GPIO 25 | ULTRASONIC SENSOR, TRIG\_PIN |
| GPIO 26 | ULTRASONIC SENSOR, ECHO\_PIN |
| GPIO 12 | H-BRIDGE, IN2 |
| GPIO 13 | H-BRIDGE, IN1 |
| GPIO 14 | H-BRIDGE, ENA |
| GPIO 15 | H-BRIDGE, ENB |
| GPIO 16 | H-BRIDGE, IN4 |
| GPIO 17 | H-BRIDGE, IN3 |