Bryon Catlin - Engineering Project Portfolio

Florida Poly Phoenix Racing - Solar Car

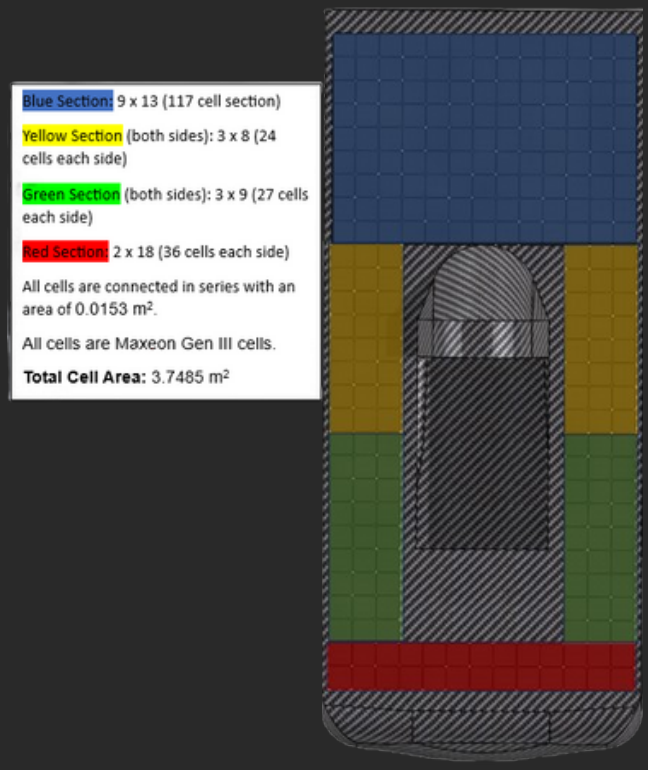
As a Phoenix Racing team lead, I designed all solar array and solar power delivery solutions. The solar car was designed to be powered by one rear motor. As a team lead, I collaborated with a team to design and manage all electrical systems onboard the final design of the vehicle. Our team achieved a final score of ninety-three laps or 300 miles in this endurance competition, placing 7th out of 24 highly competitive teams.

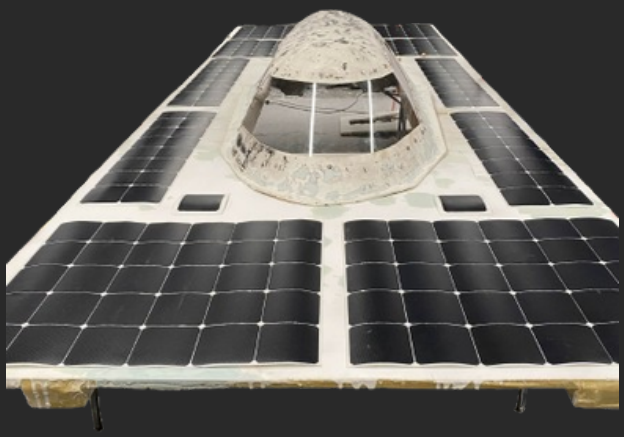
During the Phoenix Racing project, I was required to meet specific design constraints such as a desired output voltage of at least 120 volts. This voltage would allow systems to maintain stability and meet the maximum required voltage required by the motor and battery input.

Performing the calculation by including the number of solar cells and connecting them in series, it can be observed that the theoretical voltage fulfilled the design requirements:

0.7 (output/cell) * 255 (total cells) = 178.5 volts.

Another design constraint which the team needed to maintain was a maximum of four-square meters area of solar cells. In order to ensure proper aerodynamics and weight distribution, the optimal area was determined to be 3.7485 square meters of solar array area.

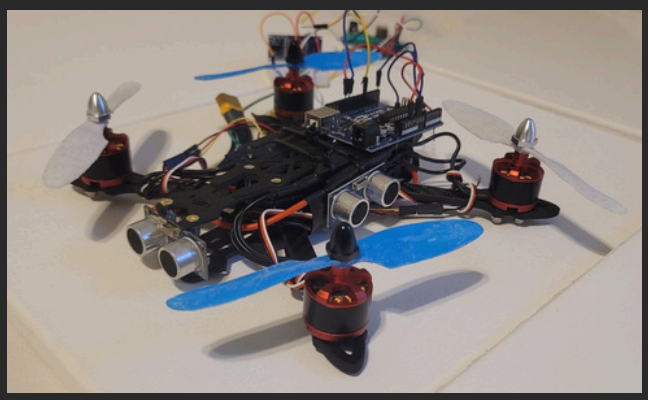




After receiving and verifying the layout of the cells properly fit the body of the vehicle, the solar array was tested to verify the actual voltage in both overcast and sunny weather conditions. These tests were conducted to verify the stability of the output of the arrays depending on weather conditions. During each of our tests the arrays outputted slightly below expected theoretical voltages due to added resistance from heat and non-ideal weather conditions. The design was completed using two Maximum Power Point Transfer units, allowing for increased reliability, a stable voltaic supply, and a higher current delivery to the battery.

Motion Controlled Quadcopter

The motion-controlled quadcopter is a device programmed using embedded C (AVR C), implementing an ATMega328P Arduino microprocessor. This project was completed as an engineering partnership. My work enabled the success of stability control, electric motors, and feedback sensors. The final design allowed users to control the speed of each motor through an ultrasonic distance sensor. The control program utilized control feedback loops for sensor-based stabilization.



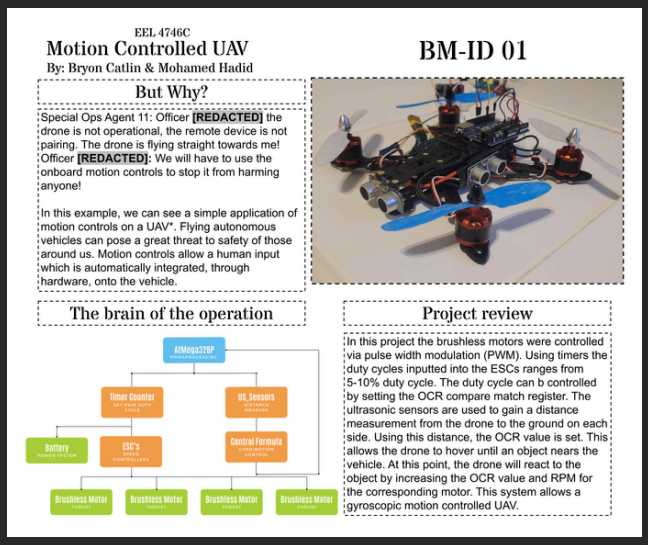
```c
// measure time for return
uint16_t ultrasonic_measure_distance(int x) {
    uint16_t count = 0;

    // Wait for the echo pin to go high
    while(!(PIND & (1 << x)));

    // Measure the time for which the echo pin stays high
    while(PIND & (1 << x)) {
        count++;
        _delay_us(1);
    }
    return count;
}
```

Because this project was programmed in embedded C, many constraints of the low-level library added to the complexity of the system. Displayed on the left is a snippet of code displaying the control of each different pin for the ultrasonic sensor's distance measurement. Traditional high level programming permits the use of IDEs such as Arduino IDE and C++ libraries. These libraries would allow for a simple measurement instead of a manually created timer measurement. Many other examples of these complexities can be observed when analyzing the distance feedback calculation and motor control.

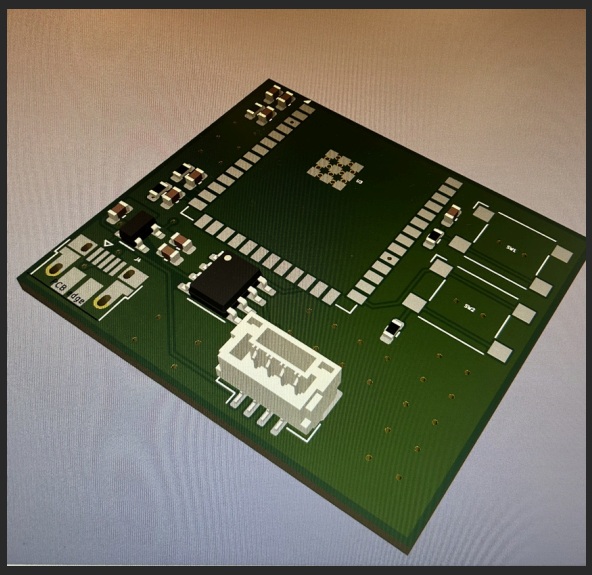
The project was demonstrated in a showcase which displayed the functionality and milestones achieved of this project. Given that the quadcopter was primarily controlled through ultrasonic distance sensors, the quadcopter would detect and avoid obstacles and users. The goal of this project was to demonstrate the added safety of sensors and rangefinders on quadcopters. One challenge we had to overcome during this project was the noise created by the other sensors. Because all four sensors operated concurrently, the sensors which were in the process of measurement would detect other pulses as a received sound. This caused unpredictable reactions. This noise was dampened by mounting the sensors in opposing angles, decreasing the proximity of the pulses. Another potential considered solution was the use of lidar range finding rather than ultrasound. While lidar would be a more costly alternative, the measurements would be much less susceptible to noise and the outside environment.



UAS Remote ID Module

The Unmanned Aircraft System (UAS) Remote ID module broadcasts a preset identification number through Wi-Fi and Bluetooth transmissions. The module, which was completed as a solo project, achieved consistent data transmission and cost-effectiveness. As the designer for this Printed Circuit Board (PCB), I used KiCAD software to design the product. The final design has been manufactured for retail sale. The Remote ID module allows UAS manufacturers to be compliant with Federal Aviation Administration regulations. The module is currently implemented on five UAS.

The Remote ID module uses firmware developed through the Ardupilot open-source project. Using ArduRemoteID firmware, the ESP32 was loaded with remote identification parameters and customization. During the design phase of the project, the range of the wireless transmitter became a primary concern in ensuring Federal Aviation Administration (FAA) compliance. To ensure that the FAA would be able to identify the craft, the antenna and module was mounted on the aircraft in an ideal position to avoid noise generated in the broadcast frequencies (2.4-2.48 GHz).



Through the use of KiCAD software, the design was completed in an effective manner before the assigned deadline, enforced by the FAA. The PCB was completed in two iterations. The first iteration, displayed on the left, included a GH connector making it much easier to prototype. It was printed without a status LED to provide user feedback. The second iteration shown in the first image included alternate clickmate connectors for improved interoperability among other aircraft hardware. On top of the addition of a new connector and LED, the second iteration also included mounting holes which matched a standard M3 mounting pattern among the target user's systems.