Testing

Because our project involves many components, testing was first performed on each component to ensure they worked properly before testing the entire system:

|  |  |
| --- | --- |
| **Components** | **Test** |
| Motors/ESCs and Transmitter/Receiver | Confirm all motors spin by changing transmitter input |
| Power Distribution Board | Connect all ESCs to power distribution, power with single battery, and confirm all motors spin |
| 5V switching Regulator | Confirm battery input was converted to 5V output |

As described in the hardware section, the drone system consists of a variety of components: a transmitter, a receiver, a gyro, a 5V regulator, 4 motors, and 4 electronic speed controllers (ESCs). The first test we performed were to make sure all of our hardware components worked. We tested each motor and ESC combo by powering it with a battery, connecting tito the receiver, and varying the throttle to check if the motor would spin. WE repeated this test with each motor and conrifmed that all 4 motros and escs functioned as expected. This test also established the receiver and transmitter were functioning properly because they signal was transmitted from successfully to make the motor spin. Next we tested the power distribution board by plugging in all four ESCs to the board, connecting each ESC to the receiver, powering the distribution board with a single battery, and confirming each motor would respond to transmitter input. Finally we tested the 5V switching regulator to confirm it could convert the battery’s 11.1V to 5V by checking the output with a multimeter.

Once each component was tested, we moved on to begin completing the critical milestones necessary to complete our project.

|  |  |
| --- | --- |
| Critical Milestones | Technical Description |
| Read Gyro data | I2C communication from Gyro to FRDM K64 |
| Control Motors using FRDM K64 | Write PWM from FRDM K64 |
| Interpret Receiver input using FRDM K64 | Read PWM from FRDM K64 |
| Untethered Power | Power all electronics using 5V regulator without USB input |
| Make Drone react to Transmitter | Transmitter input translate to appropriate drone movements |
| Make Drone React to Gyro | Gyro input translate to appropriate drone movements |
|  |  |
|  |  |

Our first milestone was to successfully read gyro data. First we wired the gyroscope to the FRDM board using I2C for data transmission. Then we implemented and modified an I2C library found on mbed to communicate with the gyroscope. First we set the appropriate addresses and registers necessary for communication. Then we used serial to read back the roll, pitch, and yaw values from the gyroscope. We used RealTerm, a Serial tErminal software to establisha serial connectiona dn monitor the data stream coming from the gyro. After some trouble shooting, we successfully read roll, pitch, and yaw, values from the gyro. Next we tested to see if the data make sense. We would accelerate the gyro in one axis and check to see that axis value increase. We did this for each access, determingin which direction of movement cause a positive and negative value. We recorded these direction and noted which axis would be used for roll pitch and way based on the placement of the gyro on the drone.

Our next goal was to control the motors directly fomr the FRDM borard cvai PWM without a transmitter and receiver. We implemented a PWMOut library to control the PWM input to each ESC. WE connected each ESC ground wire to ground and signal wire to a PWM port on the FRDM. Then we created 4PWMOut objects and set their pulse widths to 1500us. After we flashed the code, all motors successfully spun. We then tried varying the pulse width from high to low using a for loop to find the pulse width corresponding to the motor’s maximum and minimum speeds. We found that each motor’s pulse width range was between 1 and 2ms.

Next, we needed to have the fRDM board read the PWM signals generated by the receiver. We implemented a PWMIn library from mbed to help us determined the pulsewidth using interrupts and timers to measure the time difference between the rising and falling edge of the signal. We tested this by using serial to print the pulewidths of each channel to RealTerm. To test if was working correctly, we varied the transmitter inputs and confirmed each channel’s pulseidwth varied from 1 – ms as the channel input was swept form minium to maxmimum position. After scaling the values apporoptiaely, each cahnnel’s pulsewidth changed appropriately based on the transmitter input.

Because the drone is designed to fly, we could not rely on the USB cable as a permantetn power source. The next test consisted of ensureing all electornic compoenets (receiver, gyro, and FRDM board) could eb powered via a lituium polymer battery. We referenced the FRDM baord’s datasheet and found the board could be powered externlally via the Vin GPIO port. But because the Vin can only handle 5-9V, we used the 5V linear regulator to step down the voltage. We connected the battery to the regulator, and the regulator output to the Vin port on the FRDM board. The FRDM board succesffuly received power. Then we tried powering the Reciver, which taeks a 4-6V input form the 5v GPIO port in the FRDM board. Unfortunately, the reveicer would not trnu on. Instead we directly powered the reciver using the 5V regulator and this worked succesffuly. Finally, we powered the gysocope using the 3V# GPIO port on the fRDM board. We plugged int the battery to the regulator and all components were succesffulyl powered.

The next test was to confirm that the motros would respond to transmitter input. After successfully reading PWM form the receiver and writing PWM to the motors, this test was ensure the transmitter movemetns would result in the correct response by the drone. Because this test was done without propellers moutned to the motors, we monitored the pulsewdith of each motor to tell when it was at a high or low speed. We used serial to print the PWM values for each motor every 250 cycles of the main loop. This test was broken into four componeents, throttle, roll, pitch, and yaw.

|  |  |  |
| --- | --- | --- |
| **Channel** | **Input** | **Motor Response** |
| Throttle | Increase | 1  2  3  4  1  2  3  4  1  2  3  4  1  2  3  4 |
| Decrease | 1  2  3  4  1  2  3  4  1  2  3  4  1  2  3  4  1  2  3  4  1  2  3  4  1  2  3  4 |
| Roll | Right | 1  2  3  4 |
| Left | 1  2  3  4 |
| Pitch | Forward | 1  2  3  4 |
| Backward | 1  2  3  4 |
| Yaw | Right | 1  2  3  4 |
| Left | 1  2  3  4 |

First the throttle was swept form high to low to ensure all four motors simultaneously changed speeds. Then yaw was tested by moving the yaw chennl right and left and

Full System Tests

* Drone stabilization, isolate axis, roll pitch , yaw

Challenges

Shorting to frame

5V rail

Main loop time

flipping