Arduino-Nano Receiver

Architecture notes

I have the code for the RF receiver on Arduino Nano. The Arduino acts as SPI master to suck data from an RF module, NRF 24. I could port the code to a different (faster) processor, but don't plan on doing so unless there is overwhelming reason to.

I have the code written for Raspberry PI Zero W to measure elevation (acoustically) and horizontal position (using a mini camera). This is unlikely ever to be ported to another processor as the camera has a dedicated port on the RPI (Raspberry PI).

To avoid implementing a complete flight controller, at this time, I plan on doing the altitude and XY corrections in the Arduino.

Both the Arduino Nano and the Raspberry PI Zero W have one each of SPI and I2C. As the NRF24 is accessed as SPI slave, the corrections will be accessed by the Arduino acting as an I2C master.

Timing

RF (from NRF24) overhead unknown payload 6 bytes pulled by SPI. Guess 2 uS.

Altitude, X and Y offsets, and later rotation pulled by I2C from C Zero W. Say four bytes. Guess 0.5 mS. This is probably fast enough, if not: data could be packed into nibbles.

Implementation Plans

I have designed a PCB to mount the Arduino Nano, the NRF24, 3.3V power supply, the PWM outputs, and I2C interface to the Raspberry PI Zero W. The time from sending the Gerber files to receiving the boards was four days. Cost: $5 for 10 boards. Because it was my first order, the shipping (normally about $17) was waived.

I will build two boards and program one as a transmitter with fake data to validate the receiver. When this is working, I will debug the transmitter (with the GUI) connected to an RF module.