Building a portable ground control station for R/C planes

Sean McNamara

Mentored by Mr. Christopher Rogers

SMA Science and Mathematics Academy

Introduction

Unmanned aerial vehicle flying is an accessible hobby that is growing in popularity with over 670,000 new flyers registered in 2016 alone (Huerta, 2017). One new possibility is to create a game where pilots can duel each other for entertainment. The objective of this project was to make this possible by creating a control station that will give the user improved ergonomics and the additional controls needed to play the game. Integrating this with Aidan Zoretic's IR hit detection system would create the base of such a game, with the opportunity to be expanded to other applications (such as first person flying).

Materials and Methods

The planes were built using foam board templates from FliteTest™, a hobby website dedicated to providing information needed to start flying. Initially the FT Versa, (a flying wing design) was chosen for its high durability and the ease of mounting a camera or any other needed components to it. But it proved to be difficult to fly, and weight balance was an issue so the more basic FT Flyer (shown in Figure 1) was used instead. Construction of the FT Flyer only required a few cuts and hot glue, as opposed to the scoring and folding required for the Versa. Repairs were made after crashes.

Figure 1: Completed FT Flyer (26" wingspan).

Figure 2: The basic control setup.

A joystick was then modified so that the two potentiometers inside (shown in Figure 2) could be connected to an Arduino® Mega to read the analog values. A throttle lever was constructed with laser cut plywood, a 20,000 Ω potentiometer, some brass tubing, a short wooden dowel, and two bolts. The controls were mounted to a basic rolling cart with Velcro® (Figure 2). All of the wiring was soldered to a small daughterboard that is mounted to the Arduino® to keep it secure (Figure 3).

Materials and Methods (cont.)

For the Arduino® to control the plane it must communicate through the radio transmitter. Through online research it was found that most radio transmitters use Pulse Position Modulation (PPM). The radio transmitter was tested with an oscilloscope to determine its particular parameters. It starts at 3.3 V and has low (0 V) pulses for 400 µs then goes back to 3.3 V after a variable period of time. The period between each low pulse ranges from 600 µs to 1600 µs, depending on analog values from the controller. Each pulse delay determines the analog value for that control channel which determines the position of a servo or throttle power.

Code was written for an Arduino® to receive analog inputs from the joystick and throttle and generates a PPM output (See graphs 1 and 2). Code was found online with a similar function, which the code used here was based on. These values were mapped to a value ranging from 600 µs to 1600 µs, which will represent the length of time after the pulse for that control. A voltage divider was connected to the PPM output to reduce the 5 V from the Arduino® to 3.3 V to avoid damaging the transmitter. Each of these pulses and delays creates a sequence that repeats every 20 ms, which is output through an S-video cable to the transmitter.

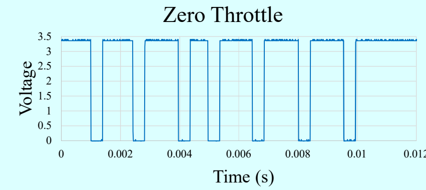
Figure 3: The Arduino® Mega with the daughterboard attached.

Results

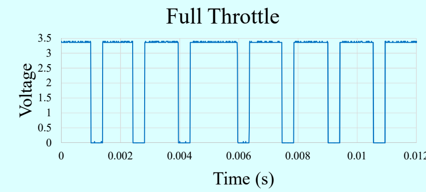
The Arduino® code functioned well with no noticeable delay. It generated a PPM output (Graphs 1 and 2) that was able to effectively communicate with the transmitter. This resulted in smooth responsive controls. The code fully links the Arduino® with the transmitter with the ability to accommodate for more controls that would be needed for an advanced plane. The controller currently lacks the ability to control a more complex plane. The controller also includes a trigger button that can be wired for other uses.

The modified joystick and built throttle were found by users to be comfortable to use. The only ergonomic issue found was that the throttle felt loose and was easy to move unintentionally.

Results (cont.)



Graph 1: This is the PPM stream generated with a throttle value of 0. The delay between the 3rd and 4th voltage drops is at its minimum value.



Graph 2: This is the PPM stream generated with a full throttle value. The delay between the 3rd and 4th voltage drops is at its maximum value.

Conclusion

The Arduino code and all controls (both joystick axes and throttle) worked as intended, and allow for easy modification. More controls could also be added, such as foot pedals to accommodate more complex planes. Currently this system only works with PPM based transmitters with these specific parameters. Different code would be needed for compatibility with other radio transmitters. It has a trigger button to be used as an input for the game, but this does not have any functionality on its own.

A great addition to this project that was not realized would be the addition of a first person view (FPV) camera system. Flying with an FPV system would allow for the use of a screen overlay that displays information to the pilot from sensors on the plane (such as battery status, pitch, roll, altitude etc.). With the IR aerial dogfight game the overlay would show key information for the game (such as a crosshair, hit markers, victor, etc.). These features can be used outside of gaming as well and make this project a low-cost aerial drone system.

References

Huerta, M. (2017, January 6) Speech–"Drones: A story of revolution and evolution" Retrieved from: https://www.faa.gov/news/speeches/news_story.cfm?newsId=21316