ENGR100-Quadcopter: Custom Design Project Final Proposal

Samuel Hofmann, Christopher Van Dyke, Hassan Kadiri

October 6, 2020

The purpose of this proposal is to provide guidance on the structure of the custom design project. This proposal will function as a final proposal for our project.

1 Context

We are seeking to develop a system that enables camera surveillance quadcopters to take off when they detect enough daylight to capture clear photos. This will be achieved by equipping the quadcopter with an analog light sensor, enabling it to measure the amount of ambient light and takeoff once that measurement reaches a set threshold.

2 Value(s) Provided

Having an autonomous quadcopter can be both difficult and expensive technology to design and implement, especially when it involves controlling complex systems. We propose a simplified way of having quadcopters take off and land via a light source, therefore enabling the quadcopter to function almost independently from the manual controls. This is a relatively cheap and simple solution that, when implemented, would enable a fleet of surveillance drones to begin surveillance without human intervention.

3 Approach

Equip the quadcopter with an upward-facing light intensity sensor. This sensor will function as a receiver and measure the ambient light intensity of the current area it is in. Depending on both the intensity of the light detected and its duration, the quadcopter would either take off and hover at a fixed height or land. Minor adjustments to position and orientation may occur to ensure smooth transitions and executions of the quadcopter's movement.

We plan to have the quadcopter take flight once it detects the light intensity of the room to be at standard light levels in the lab (meaning, light levels with all lights on, such as when we work in the lab on a typical day) for one second. This will prevent a brief flash from setting the system into motion. We will use the sensor to measure readings in the room before we attach it to the quadcopter to determine what this cutoff level will be (Estimated to be about 500 lux). Once the light levels dip back below the threshold for five seconds, the quadcopter will signal to itself to land.

The quadcopter may only work in dark areas depending on the measuring intensity of the sensor, but we might be able to make it function in lighter areas if the difference in light intensity is large enough. We will also have a toggle/trigger to control the function of the light sensors in case of accidentally triggering one of the functions.

4 System Architecture

One light sensor will be added to the quadcopter. The Arduino code will also need to be modified to interface correctly with the sensors. Figure 1 illustrates the suggested alignment and angle of each sensor. Both sensors will be detecting light intensities, then the data will be sent to the Arduino which will compute the mean light intensity. A filter will be used to ensure the outlying data will not trigger any functionality of the quadcopter. Once the mean value, post-filtering, has been above the threshold for one second, the Arduino will signal to the BeagleBone to take off. Similarly, once readings fall below the threshold, the quadcopter will begin the landing procedure.

The sensor will output different voltages based on its reading of the Illuminance (in Lux) it detects. We will filter then use these outputs to set the threshold output needed to tell the quadcopter to take off and land.

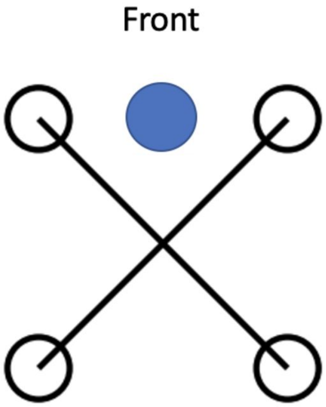

Figure 1:

5 Components

- Arduino Uno
- Photoresistor
- 4-5 jumper wires
- Resistors (10k Ohm, 3.3k Ohm,
- Will require the use of the BeagleBoneBlue
- Small Bread Board
- 4-pin Male JST connector
- USB-A to USB-B connector

6 Main Challenges and Milestones

- Design/create the physical structure that will house the sensor and Arduino (0.5 Weeks)
- Mount and interface sensor with Ardunio (0.5 week)
- Modify Arduino code to support the new light sensor and its functionality (1 weeks)
- Testing, Debugging, and revisions (1 weeks)

7 Changes From Initial Proposal

- One sensor instead of two.
- The sensor will be a photoresistor rather than a more complex light-intensity sensor.
- Added resistors and a voltage diverter to required materials.
- Modified milestones and length of each to fit with time restrictions in the lab.
- No library code for Arduino IDE needed.
- Added Arduino Uno, USB connector, and 4-pin Male JST connector to components list and removed a few items that we didn’t end up needing.