Quadrocopter Flight Control

Functional Description and System Block Diagram

By: Eric Backman

Advisor: Dr. Malinowski

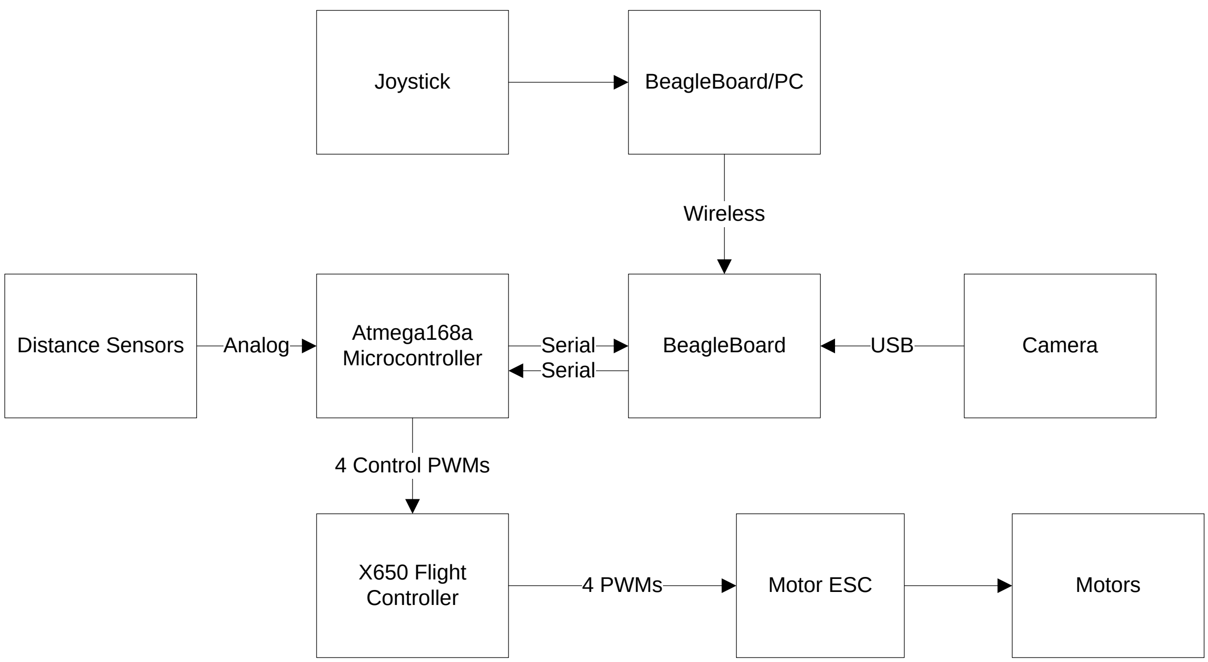
Date: 10/11/12

Introduction

Quadrocopters are ideal for small scale flying robots due to their ability to hover like a helicopter without the need to change the rotor blade pitch angle. This simplifies their design and control. Last year, Brad Bergerhouse, Nelson Gaske, and Austin Wenzel worked on an autonomous quadrocopter. They constructed the quadrocopter platform, installed a real time operating system on BeagleBoard, and started sensor implementation.

The primary goal for this year's project is to get the quadrocopter flying and responding to inputs from sensors using microcontroller Linux. The platform is already built and needs controllers and sensors to function. A joystick will be connected to another Beagleboard or PC that will connect wirelessly to the onboard controller. This will allow a user to control the quadrocopter. Simple object avoidance will be programmed to prevent the quadrocopter from crashing, as well as simple flight patterns to allow for semi-autonomous flight and navigation. Lastly, a camera will be integrated into the system to allow for better navigation.

Block Diagram



Functional Description

This system will work by taking the inputs from the sensors and sending them to the analog-to-digital converters on the Atmega168a chip. These digital signals will then be sent to the BeagleBoard through a serial port. The BeagleBoard will take these inputs and use them to influence the command coming either from a controller or based on pre-programmed autonomous decisions. It will generate a signal for yaw, pitch, roll and elevation. Since it is tough to generate PWMs from Linux, it will send these signals back to the Atmega168a which will generate the required PWMs and give them to the built-in flight controller. The flight controller will take the commands and use those to run the motors.

Subsystems

Sensors
Distance sensors will be chosen and integrated into the system to provide object avoidance. These will generate an analog voltage that will be converted into a distance and implemented into the controls on BeagleBoard. The sensor array will be designed so that all sides are watched preventing any collisions. One sensor will point up to prevent running into the ceiling and one will point down to allow for landing. Another 4-6 will be put on the sides in a way that will allow for accurate and simple navigation. One issue that is involved with sensors on a quadrocopter is how tilting effects the distance to objects. When the quadrocopter tilts to move, the sensors are no longer perpendicular to the wall. This makes it difficult to know the actual distance to the wall as there is no method for determining the angle. This will be handled by making sure the angle never exceeds an angle where it is impossible to detect a wall in time to stop.

Controllers
BeagleBoard will start by receiving commands wirelessly but eventually will have navigation for autonomous flight. It will use these commands as well as the sensor inputs to create commands for yaw, pitch, roll and elevation to be sent to the PWM (Pulse Width Modulation) generating microcontroller. The PWMs are pulse signals that vary the duty cycle to encode information or provide different amounts of power. Remote Control PWM standards use a 50 Hz signal that has a pulse length from 1 ms to 2 ms to stand for 0% -100%. It will send these commands to the microcontroller via SPI. The microcontroller will generate the PWMs that are inputs to the flight controller. The flight controller converts the control PWMs which represent yaw, pitch, roll, and elevation into commands for the four motors. The flight controller also uses a built in accelerometer to handle stability controls. It generates the PWMs for each motor that are sent directly to the motor electronic speed controls (ESCs) which generates the power PWMs that actually run the motors.

Camera
The camera will be placed at the bottom of the quadrocopter and allowing it to help in navigation. Simple digital signal processing will be implemented in the BeagleBoard to create navigation such as color following or landing on a color coded landing pad.

References

[1] Brad Bergerhouse, Nelson Gaske, Austin Wenzel. Aerial Collision Avoidance System. Senior Project, Electrical and Computer Engineering Department, Bradley University, May 2012, http://cegt201.bradley.edu/projects/proj2012/quadcptr/

[2] Introduction to Autonomous Mobile Robots, 2/ed., by R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza, MIT Press, 2011, ISBN: 978-0262015356