CSE466 Blimp Project – Phase Three

AUTUMN 2011

OBJECTIVES

<u>In this lab you will DO the following:</u>

- Develop a PC-side GUI for remote control of your blimp.
- Implement autonomous altitude control using the IR proximity sensor.
- Document your project.

DELIVERABLES

At the beginning of the next lab period you will demonstrate the performance of your blimp bot.

Additionally, a set of documentation is required which will cover all three phases of the blimp lab. This documentation should take the form of **both a paper and a web page,** both presenting the same content. The completed paper should be 6 - 8 printed pages of text and figures. The webpage should include all text and figures from the paper, but may also include videos or other media. The webpage will eventually be made public, so please keep that in mind when choosing content. Here are the requirements for the report:

- 6-8 pps paper format; webpage
- Sections for paper:
  - Introduction
  - Related work: what else has been done with blimps (here in CSE, elsewhere)
  - System Design & Implementation
    - Description of HW
    - Description of Firmware
    - Description of PC SW / GUI / Control algorithm
    - To include:
      - Wired UART communication
      - Wireless communication protocol (error detection?, net communication rate)
      - Motor control
      - High level control --- height control
      - Extras / application
  - Results
    - What worked
    - What didn't work
    - Characterization of controller (error levels, control loop rate)
  - Future work
    - What else could / should be done with this system
    - Suggested modifications for next generation of HW
- Figures, etc to include
  - Photo(s) of system
  - Video(s) of system in operation
  - SW block diagrams
  - Time series of sensor data, illustrating control system in operation
  - Source code listings (web only)

RESOURCES

These documents and web resources will be useful in completion of the lab and/or in answering the questions posed. Additional resources are described in the procedures below.

- Class discussion board
- BlimpBot schematics and component datasheets on course website
- MSP430F5510 Datasheet
- MSP430X5XX Family User's Guide
- MSP430F2274 Datasheet
- MSP430X2XX Family User's Guide
- MSP430 Code Examples
- MSP430 Optimizing C/C++ Compiler User's Guide

IMPORTANT NOTES

There are several very important rules to follow when using the blimp hardware. If you do not follow these rules, there's a good chance you'll release the magic smoke and have to wait to have your hardware repaired.

1. **ALWAYS POWER ON THE BLIMP BATTERY SWITCH BEFORE CONNECTING THE DEBUGGER**. Never connect the debugger when the battery switch is OFF or when the battery is disconnected or dead.
2. **NEVER short the motor or battery terminals.** The motor driver ICs and battery are very good at sourcing HUGE currents, and things will be damaged. Use caution when probing VBAT or motor terminals.
3. **ALWAYS plug in the battery with the proper polarity.** There is no protection against a reversed battery!
4. **ALWAYS keep the battery voltage above 3.4V.** When you notice the regulated voltage on VDD starting to drop below 3.3 V, it is time to charge the battery (HINT: Your code should watch for this "drop out" condition...)
5. **ALWAYS disconnect the battery when not using the blimp.** This helps ensure the blimp will not be left ON and deplete the battery.

PART 7: PC-SIDE GUI

In this section you will develop a GUI for wireless control of the blimp. This will require implementing bi-directional serial communications from a PC-side application in order to both send commands to the blimp and receive sensor data and/or state information from the blimp. The GUI should communicate with the blimp via the existing CLI developed in Phase 1; no new control protocol is required.

The following is a list of the actions which should be available to the user when operating your GUI:

- Select Run or Stop mode
- Select or deselect autonomous altitude control mode
- Control motors by varying direction and duty cycle
- Query IR proximity sensor (one or all) and view results
- Extra credit: View low battery alarm indicator
- Extra credit: View compass heading

The details of this GUI are up to you. However, the GUI must have some buttons or other graphical features (it can't simply be a textbox containing your CLI). Additionally, it's recommended that you put some thought into the motor control interface: a well-designed interface will allow you to fly the blimp from the GUI with ease.

The optional extra credit features should be implemented only AFTER the remainder of the lab and documentation is done, as they may be significantly time consuming.

GOALS:

- Develop a PC-side GUI which provides the functionality listed above.
- Use the GUI to fly your blimp around!

PART 8: AUTONOMOUS ALTITUDE CONTROL

The vertical infrared proximity sensor may be used to estimate the height of the blimp from the floor, and so can be used in the feedback loop for an autonomous altitude control system. This control loop may be incorporated into the blimp firmware, or may be implemented within the PC-side application. However, the blimp must be able to accept and run other commands while in autonomous altitude control mode (this is an argument for using the PC-side application to handle the control loop).

Select one of the control algorithms from the lecture slides and use it to vary the effective thrust of the vertical fan based on the difference between the IR sensor value and a desired set point. The slides describing control methods can be found here:

http://www.cs.washington.edu/education/courses/cse466/11au/calendar/10-control-posted.pdf

NOTE: It's recommended that you start with a simple control algorithm, such as the "Bang-Bang controller". If you have time, work your way up to a more complex controller.

NOTE: When tested prior to the lab, the IR sensor was capable of accurately detecting the floor with 1 - 2 inch resolution from a distance of several feet. If your performance varies significantly from these figures, find help.

GOALS:

- Use one of the control algorithms described in the lecture slides to maintain a constant height.

PART 9: ANALYSING PERFORMANCE

In this section you will quantify the performance of your autonomous altitude control system. The performance metric will be the mean-squared error between the sensor value and the desired set point over a period of time.

You will collect a time series of IR sensor measurements as the blimp adjusts to an abrupt change in the set point value:

- Place blimp into autonomous altitude control mode with a given set point value.
- Wait for blimp to stabilize at that altitude
- Begin collecting IR sensor data
- Wait 10 seconds
- Abruptly change the set point value, which will cause the blimp to adjust its height.
- Wait 10 seconds
- Stop collecting IR sensor data

Compute the mean-squared error between the set point values and the IR sensor values over the entire 20-second long event. This number should be present in your report, along with a time series plot of the IR sensor data and set point values.

EXTRA CREDIT: DO SOMETHING INTERESTING!

Here are some suggestions for extra functionality you can implement:

- Maintain a compass heading.
- Simple collision avoidance using horizontal IR sensor channels.
- Find an IR beacon (use your F2012 or 2013 to flash an IR LED).