# Lab Outline
EE 234

Andy Parks
Partner: Gustavo Gomez

January 21, 2012

**Abstract**

The purpose of this report is to outline what content should be in and how to organize a lab report. First an abstract should outline the report, it should give the reader an brief overview of the report. The abstract is different from the introduction in that it outlines specifics from the experiment or project completed. It should touch on each of the three body sections of the report, *Software-Hardware Design*, *Test Procedure and Results* and *Conclusion - Why I Care...* , giving the main points of each section. The reader should be able to summarize the main ideas in the report by only reading the abstract. Although it is to provide breadth, it should be no more than 130 words.

## Introduction

This section is intend to introduce the reader to the concepts covered in this lab. Assume the reader has a basic technical knowledge of what a Microcontroller is. It is acceptable to assume that the reader is also taking your class, their knowledge is accumulative. For example if the lab is focusing on interrupts, it is not necessary to explain what registers are, however it is important to explain (in your own words) what interrupts are. You should also include a few sentences about the application of the concepts covered. This will not only help me see you understand the concepts but should also help reinforce your knowledge of the concepts.

## Software Design

How does your software work? How did the design come to be? This section is focused at answering these two questions. In order to answer how your software works I suggest using flow charts. This will give the reader a visual to communicate your design. Would you rather read a paragraph about how the fetch decode execute cycle works, or a few short sentences combined with a visual?

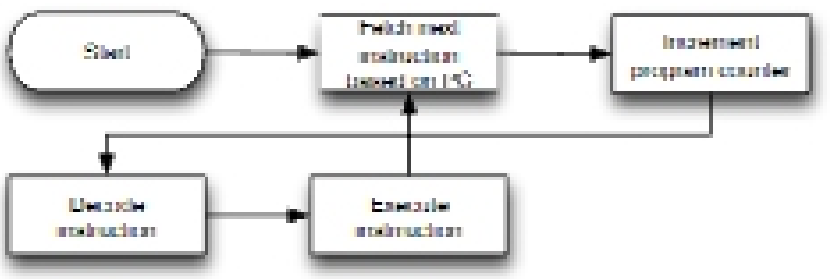


**Figure 1:** Fetch, Decode, Execute

Design considerations and tradeoffs should be reflected on when answering how the software was designed. Did you have to store data in SRAM or the stack, due to limited registers? Did you need to limit the number of ROBO-MAL instructions due to op-code length? I am interested in your though process and how you designed your final product.

## Hardware Design

Only knowing how the software works is often only half the picture in lab. Often it is important to understand how the hardware works. How was it connected? How did it interact? If someone outside of lab were to try and use the code for your robot, they would need to know how to configure the hardware as well as the software. Pictures and diagrams may be helpful in relaying this information. For example, a recent project I was working required I create this simple block diagram showing connections.

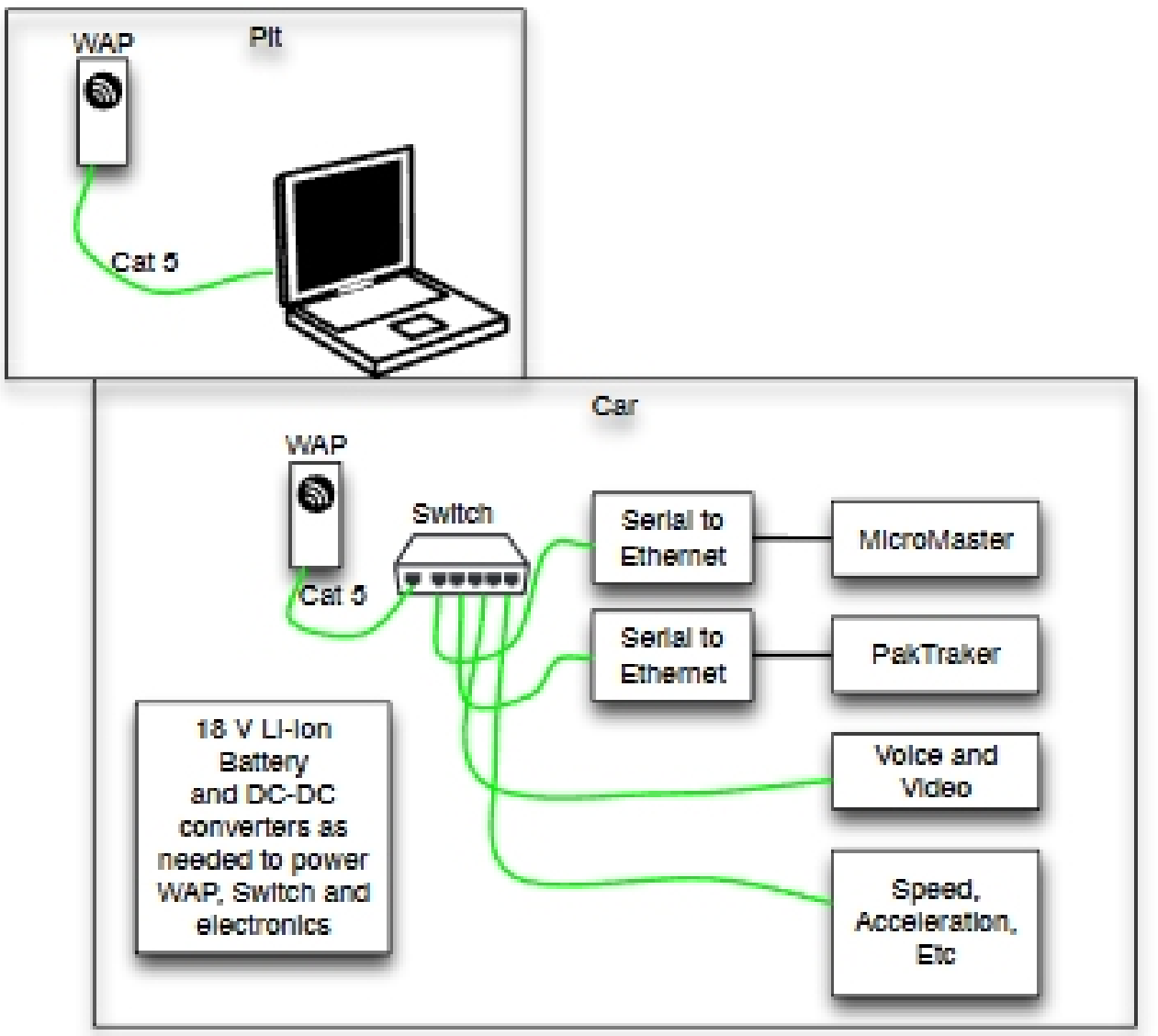


**Figure 2:** EV Team Communications

## Test Procedure and Results

Designing and describing the software and hardware is great, but did it work how you expected. Developing a test procedure and testing the product is very important. This section should be used to explain the test procedure used and the results of these tests. For example did you test your line following robot by building your own course? How did the robot preform? Or did you use the AVR Studio debugger to test a series of conditions? I am interested in how you verified the project did what it was supposed to (before doing the demo) and how it preformed compared to your expectations.

## Answers to Questions

Please answer the questions that Andy O'Fallon presents at the end of each lab.