SPI device interfacing with RTU
Team: Richard Lin, Jerry Chen, Allen Tang
EE149/249 Project Charter, Fall 2014

Project Goal
In this project, we will examine implementing traditional microcontroller peripherals on a RTU (real-time unit) coprocessor.

Project Approach
We will start with bit-banging SPI, then communicating with an digital accelerometer and gyroscope, and implementing a simple IMU. A majority of the RTU implementation will be provided, so we will be using that as our starting point. The SPI protocol implementation will likely be a state machine, and we provide an analysis of timing precision capabilities and guarantees.

Resources
At minimum, nothing except a computer - all development can be done in simulation with test vectors. However, it would be fun (and make for a cool demo) to have actual hardware to play with. This will require a FPGA on which to deploy the RTU RTL, along with devices (like an accelerometer) to interface with.

Schedule
10/21 This document.
10/28 Midterm cram: let's be honest, not much else is getting done.
11/4 Learn existing RTU implementation and protocols planned for implementation.
11/11 Get RTU infrastructure working (simulator at least, FPGA if possible).
11/18 Code bit-bang SPI module in RTU, including host interface.
11/25 Analyze timing guarantees for SPI, determine maximum frequency.
12/2 Implement accelerometer protocol RTU on top of SPI module.
12/9 Implement other low-level protocols in RTU, like I^2C, PWM, or NeoPixel control.
Provide timing analyses if possible, re-using work for SPI above.
12/16 Polish and prepare demo.

Risk and Feasibility
This project is relatively feasible if we are able to stick to our schedule and do not encounter any unforeseen problems. The main risk is learning the existing RTU code and toolchain, but the researcher on the project will be able to help us. Since actual hardware is not strictly necessary, the risk associated with physical devices is low.