Union College
ECE/CS 352/552
Final Project Guidelines

A one page primary proposal including as much of the information asked for in the project step 1, detailed project proposal, at the end of this document. This preliminary proposal is due Thursday April 17th, 2014.

The objective of the final design project is to integrate the programming and interfacing knowledge you have gained in the course to produce a prototype embedded system. Here are some project ideas. You may need to purchase some parts for your project if the ECE shop does not stock them and I will be happy to purchase these for you. You should do the background reading to make the best decision you can on what components to buy. Most components are not very expensive. Here are some sites that you should find useful:

http://www.sparkfun.com/ (Often, there are breakout boards available which makes it easier to connect to sensors)
http://www.jameco.com/
http://www.mouser.com/
http://www.digikey.com/

For the most part, you want to find sensors that will be reasonably easy to interface with and you want to use DIP packaging if you can find it or breakout boards.

Project ideas

Here are some general ideas for projects:

Security system - this can model a home, car, or other security system. The idea is to have sensors, user inputs, and actuators and indicators. You might consider implementing password protection, motion detection, light detection, timers, alarms, etc.

Meter - the idea here is to implement some sort of measurement instrument. The goal would be to display the measured entity in real world units as accurately as possible. Some entities will be quite simple (like temperature - been there, done that), and others would be more complex (motor speed, human walking rate, degree of cloudiness, etc). Select something reasonable.

Game - there are many games that can be implemented with a microcontroller. Tic-tac-toe, dice games, battleship, etc. For this type of project it is suggested to keep the inputs and outputs quite simple, since the programming of the game algorithm is often a challenge.

Control system - a closed or open loop control system can be a lot of fun to implement. You must find a sensor, user input, and actuator that you can interface, and then you can use the various methods discussed in class to implement the control. A temperature controller (where you can use a light bulb to heat a container) is one example.

Robot – you can come up with a project which uses the Robo51 (I have some of these). This has a lot of sensors and actuators already in place which avoids difficulties sometimes encountered in interfacing these to the microcontroller. You can also add on to this robot or modify it.

Here are some specific project titles from previous years to give you some food for thought:

2008	**2010**
Weather station	FM Transmitter
Digital protractor	Digital Anemometer and Weathervane
Tic Tac Toe	Smart Headlights
Digital Alarm	Minesweeping Robot
Traffic controller	House Alarm System
Interfacing a cellphone to the 8051	Magnetic Repulsor
Home monitoring system	Campus Trolley Robot
Psychic test	Alarm Clock

Guidelines

You only have a little about 2 - 3 weeks for this project, and you should plan on spending a good amount of time in the final week on debugging. Therefore, you should keep the project relatively simple. We will dedicate a couple of classes to debugging but you will need to work on it outside of class time. If the project is portable, you can sign out the board and work on it from home or at times convenient to you.

The project will be broken into steps:

1. Write detailed project proposal. This should contain:
    - A detailed description of how the system will behave
    - List sensors and actuators required (inputs and outputs)
    - List interfaces will be used (I2C, SMB, SPI, Serial etc.)
    - Include a block diagram of the system.
    - Include a schematic of the hardware, including the way it will be connected to the 8051
2. Interface the hardware and write simple test programs to ensure that the interface works – you can write simple assembly or C code for the interface.
3. Write the software for the main part of the project to tie together inputs and outputs.
4. Test for as many possible scenarios as possible.
5. Write a project report that describes the design, implementation, and testing results.