Project 1
Project 1 Objectives

At the end of this project, you should be able to:

• create simple microcontroller-based circuits,
• read digital information from a switch,
• compute errors between desired and actual state variables, and
• convey information about sensors using a set of LEDs.
Part 1: Circuit

• Mount Arduino board to your solderless breadboard. Mount both to the Frisbee
• Connect and mount the compass. Compass must be 12” above the ground (at least)
• 4 LEDs in circle: will use to display heading or heading error
• 10-LED bar: will use in next project to display distances
• Add a switch
Part 2: Compass Interface

Must implement:

```
int16_t get_orientation(void)
```

- Returns heading in 10ths of a degree.
- Range: -1799 to 1800 (0 is North)
Part 2: Compass Interface II

In the `main()` function: include a while(1) loop that:

- Gets the orientation from the compass
- Computes an orientation error between a goal and the current orientation (part 3)
- Depending on the state of the switch, displays (part 3) either the orientation or the error.
- `delay_ms(100)`
Part 3: Sensor Processing and Display

Must implement:

```c
int16_t compute_orientation_error(
    int16_t orientation_goal, int16_t orientation)
```

- Returns the difference between `orientation_goal` and `orientation`
- Return value range: -1799 to 1800 (0 means orientation is at orientation_goal)
Part 3: Sensor Processing and Display II

Must implement:

```c
void display_orientation(int16_t theta)
```

- Changes the 4 LEDs to indicate theta (range is -1799 ... 1800)
- Must encode at least 8 different orientations
Part 4: Hovercraft

- Mount lift fan
- Start mounting batteries and other fans
Demonstration/Presentation

• All group members must be present
• 4-5 slide presentation (see project spec)
• Demonstration
• All group members must be able to answer questions about the hardware or software
• Code review
Code

• Check in code to subversion tree (it should be clear which files are for project 1)
• Code must be documented (see project specification for an example)
Other Components into Subversion Tree

• Presentation file
• Circuit diagram: Must use EagleCad
Personal Reports

• Next week you will receive a request from Catme to fill out an evaluation of you and your group

• Must be filled in to receive project grade
Group Grade

- 35%: Project implementation
- 30%: Demonstration/presentation of working project (to either of the TA or the instructor)
- 35%: Code documentation and circuit diagram
Personal Programming Components

• Everyone must accumulate at least two during the semester
• To receive credit, you must be the primary designer, implementer and debugger of the component
• Your other group members should still help!
Personal Grade

• For all parts not including the personal programming components: each group member should contribute about equally

• If this is the case, then your personal grade will be equal to the group grade

• If not, then the personal grades will be adjusted appropriately
Next ...

• Finish sign-offs of:
  – Programming the Atmel processor
  – Group: attach LEDs to the Atmel and control them (could be the same ones for the project)

• Start on project 1