

Project 1: Addressing Multiple Digital Lines

Questions?

Project 1

- Hardware:
 - Wire in a set of LEDs
 - (leave room for future components)
- Software:
 - Provide interface functions for the LEDs
 - Write a test `main()` function

Project 1: Heading Display

4 LEDs in a circle:

- Represent heading with 8 different illumination patterns
- Interface function:

```
void display_heading(int16_t heading)
```

 - Heading = 10ths of a degree. Value between -1799 and 1800
 - Left-handed coordinate system
- Do not deviate from this specification!

Project 1: Heading Velocity Display

10 LEDs in a line (use bar graph):

- Represent heading velocity
- Pairs of LEDs are connected to a single digital pin
- Interface function:

```
void display_heading_velocity(int16_t velocity)
```

- Velocity = 10ths of a degree/sec.

Value between -3000 and 3000

- Illuminate the two center LEDs if rate is near zero

Project 1: Test Function

- Add switch to circuit
- In `loop()`
 - One switch state:
 - Increment a simulated heading from -1799 to 1800
 - Display heading
 - If heading reaches 1800, reset to -1799
 - Other switch state:
 - Increment heading velocity from -3000 to 3000
 - Display
 - If velocity reaches 3000, reset to -3000

Code Specifications as Contracts

- You are implementing code that will be used for future projects and by your other group members
- Implement exactly the functions that we ask for
 - Name of function must be as specified
 - Parameter types and names must be as specified
 - Return values must be as specified

Documentation

Project-level documentation

- Top of each C (and H) file
- Project #
- Date
- Group number
- Group members
- Group member responsible for the software

Documentation

Function-level documentation:

- Summarize what the function does in a sentence or two
 - This is for future users of your function
- Explicitly document the inputs and outputs of the function
 - Include variable names and meaning of the variables
 - Discuss any other effects that the function has (e.g., changing pin state)

Documentation

- In-Line documentation:
 - Document the ***meaning*** of individual lines of code or small groups of lines
 - Document what you are doing and why
- See the project 1 specification for a link to an example

Project Groups

- Use assigned groups
- For each project, one person must take the lead on the software

Due by the Project Deadline

- Documented code (“ino” files) checked-in to the subversion tree
 - Useful for sharing with us and with your group members
- Demonstration/code review with me or the TA
 - Complete by Monday following the deadline
 - If completed before the deadline, then you may make changes for an improved grade

Due Shortly after the Project Deadline...

Personal Report: Catme will ask you to fill out a survey

- This will be used to provide feedback to you and your group members
- I will also use this to detect asymmetries in group member participation

Grading

Personal programming component:

- Each group member must collect 3 personal programming components over the semester
- One is available for each of the 10 projects
- Grading is generally binary (completed or not)
- A programming component will not count as yours if another group member must make substantial changes to the code in order to complete the project

Grading II

Group grade

- Assess circuit, program, functionality and documentation
- See the rubric for details
- In most cases, the group grade will be given to each group member
 - In cases of significant, repeated asymmetries, grades will be balanced to reflect contribution

Hardware to Borrow

Bin in the lab will have:

- Spare parts
- Tools