Roller Coaster Design Project
Lab 4 Supplement: C Crash Course
The C Language

- Many similarities to Matlab, but some very important differences, each covered in turn in the following slides:
  - Strongly Typed
  - Vectors very different
  - Syntax differences
C Variables

- Strongly Typed
  - When you create a variable, you must also specify what sort of data is stored within the variable.
- Uninitialized – start with random data in them.
- Types:
  - Integers (int/short/long)
    - Store integer type data (-4, 2, 111)
    - Are (16/8/32) bits long
    - Can be signed (default, 1-bit used for sign) or unsigned (all bits used for magnitude, but always positive)
    - Will 'overflow' when value stored in them exceeds available bits
Types (cont')

- Types:
  - Floating Point (float/double)
    - Store floating point data (3.14, -0.00000134)
    - Are (32/64) bits long
    - Are always signed, and can represent a HUGE range of values
    - Can be tricky to work with, as it can be very hard to predict the details of how they work. Using direct comparisons (== or !=) can be VERY dangerous, logically
Types (cont')

- **Types:**
  - Character types (char, string)
    - Used to store character data
    - Usually, chars are used in arrays (see below)
    - Can be tricky to work with without more in-depth discussion of how C/C++ works

- **Arrays:**
  - Collections of other data types
  - To make an array, add [#] to the end of the variable name, with # being equal to the number you want.
  - Use [n] to access the nth element of the array
Variable declarations

- Examples of variable declarations in C:
  - `int myInteger;`
  - `float pi = 3.14;`
  - `char myName[10] = “Dr. Smith”;`
Constants

- `#define` can be used to create a constant
  - `#define pi 3.14`
  - Now, everywhere 'pi' shows up in your code, it will be replaced with '3.14' before the code is compiled
Operators

- Work pretty much exactly as they do in Matlab
- Order of operations still holds
- Vector math is NOT implemented, so '.' operators do not exist
- In C, 'not equal' is '!='
- C supports 'in-line' or 'compound' operators:
  - n++ → n = n + 1
  - n -- → n = n – 1
  - n += a → n = n + 1
  - n *= b → n = n * b
  - etc...
- n++ (etc) can be used in equations
Flow Control

- C supports if statements, while loops, for loops and case statements, like Matlab did, but with different syntax:

```c
if (pinFiveInput < 500)
{
    // action A
}
else
{
    // action B
}
```
Flow Control

for (initialization; condition; increment)
{
    // statement(s);
}

while(expression)
{
    // statement(s)
}
More Resources

- www.Arduino.cc
- CSE 1222, CSE 2221