Oh the Arduinos You'll Program

A too-fast intro

The Arduino IDE

- Integrated Development Environment
  - Allows you to program, compile, and load programs from one place
- Official one from Arduino.cc
- Many others

Programing

- How all computers are controlled
- Lots of languages
- Generally divided into a range of ‘low’ to ‘high’ level programing
- Arduino uses basically C++

Your Toolbox

- Variables
- Assignments/Math
- Functions
- Conditionals
- Loops
- The Arduino Library

Variables

- Containers to store information
- Lots of types
- Today we need only ‘int’
- Defined by saying “int foo;” or “int foo = 5;”

Assignments/Math

- Most of your standard math symbols (* , / , + , -)
- Parentheses group like you would expect
- More complicated functions like exponents, sums, square roots, need a library
- Also bitwise and shifting operators (and &, or |, not ~, xor ^, left shift <<)
- Variables are assigned by saying “foo = (bar + bat) / baz;”

Functions

- Functions allow code to be reused
- Int getNthBit(int input, int bitNum){
  int result = (input & (1 << bitNum)) >> bitNum;
  return result;
  }
- Foo = getBitNum(1337, 4); // foo is now ‘1’
  // Anything after ‘//’ is a “comment” and not run as code

Conditionals

- Conditionals allow for comparing values and running different sections of code depending on the comparison
- Void printBigOrSmall(int input){
  if (input > 300){
  printf(“%d is big!\n\r”, input);
  } else {
  printf(“%d is small.\n\r”, input);
  }
  }

Loops

- Loops allow sections of code to be repeated until some condition is no longer met
- While(1){ // Infinite loop, does this forever
  printf("All work and no play makes %s a dull boy.\n\r", name);
  }
- For(I = 1; I <= 100 ; I++)
  printf("%d Mississippi\n\r", I);
  printf("Ready or not, here I come!\n\r");

The Arduino Library

- A set of functions that let you change pins, read pins, and communicate
- The ones that you will need to use immediately are these:
- pinMode([Pin Number], [Pin Type (INPUT, INPUT_PULLUP, OUTPUT)])
- digitalWrite([Pin Number], [Pin Mode (LOW, HIGH)])
- digitalRead([Pin Number])
- Everything is documented at arduino.cc (click the ‘Reference’ tab)

The Project

- Blinking LEDs
- With buttons!
- And computer control!
- And counting!

The Most Basic Program

- (Do this with me)
- Setup function
- Loop function

Adding a Button

- Connect one side of the button to a pin
- Connect the other side to ground
- pinMode([Button Pin Name], INPUT_PULLUP);
- digitalRead([Button Pin Name])
- This reads the opposite of what you'd expect, 1 (true) when the button isn't pressed and 0 (false) when it is. This can be fixed by putting a '!' in front of that statement

Four Projects

- First, when you press the button, make the LED turn on
- Second, when you press the button, make the LED switch
- Third, when you send some specific character over serial make the LED switch
- Fourth, wire up 4 LEDs, and set them so that they display a number in binary and have it go up when you hit the button. Add serial commands if there's time

Serial? What?

- In setup put “Serial.begin(9600);”
- If there are characters to read Serial.available() will return a value that isn’t 0
- Serial.read() will return the oldest character that’s still stored (In ascii)
- You can compare those values to characters by using single quotes (‘a’) will match to an “a“ character that was returned by Serial.read()
- You can reply with printf or Serial.write(character)