Microcontroller Lab 4 (Class 13)

"Catching up and playing around"

Given all the excitement last week, I thought we should devote this week to catching up. Also, I'm going to give you a bunch of sample sketches and ideas for working with the collection of sensors and parts in your kit.

Reading

Accessory Kit -- list of items in your kit

- [ ] **Example 4.1 - Servo**

A servo (such as the one in your kit) is a motor which moves over a fixed range (often 180 degrees) whose position can be set precisely based on a control signal. This servo (like most others) expects a PWM signal (like you encountered in lab 2). Once it is wired up it just takes one function call to move it to a specific angle in the range 0-180 degrees.

Wiring:

Black wire	GND
Red wire	+5V *from the breadboard*. (Arduino +5V insufficient current)
White wire	Arduino pin 9 (or any PWM-capable pin)

(be sure to connect Arduino GND to breadboard GND)

Programming:

```
Servo myservo;          // create a servo object
// (in setup)
  myservo.attach( 9, 800, 2400);
// (in loop)
  myservo.write( 90);   // move to position in degrees
```

- [ ] **Example 4.2 - Light Sensor**

The light sensor is a photodiode with a frequency response similar to that of your eye. They're used in things like phones to dim the screen to match ambient light.
One annoying thing you can do is to use the light sensor to modulate a tone from a speaker. Let's try this example from: http://arduino.cc/en/Tutorial/Tone2.

Wiring:

Light sensor LONG lead   Arduino pin A0, with 12k resistor to GND

(resistor value 8k-12k is fine)

Light sensor SHORT lead	+5V
Speaker lead 1	Arduino pin 9
Speaker lead 2	GND

Programming:

```
void setup() {  }

void loop() {
  int sensorReading = analogRead(A0);
  // map sensor range 10-800 to tone 120-1500Hz
  int thisPitch = map(sensorReading, 10, 800, 120, 1500);
  tone(9, thisPitch, 10);
  delay(1);     // delay in between reads for stability
}
```

▢ **Example 4.3 - Duck Shooting Game**

Do you have a laser pointer handy? How steady is your hand?
Build the duck shooter to find out!

Wiring:

Light sensor LONG lead	Arduino pin A0, with 12k resistor to GND (resistor value 8k-12k is fine)
Light sensor SHORT lead	+5V
Speaker lead 1	Arduino pin 9
Speaker lead 2	GND
Servo	Arduino pin 10
plus 5V and GND	

Attach a paper duck to the servo arm. Make a paper target around the light sensor.
Aim the laser pointer at the light sensor...

Write a sketch which counts up when the light sensor is hit by the laser, and counts down otherwise. When it reaches a threshold, you win!

▢ **Example 4.4 - RGB LED**

An RGB LED is three LEDs (red, green, blue) in one package, so in principle any color can be produced by mixing. It is the big fat LED with four leads (a few kits don't have them). The long lead is the common cathode (GND) connection.
Orient the LED so that the leads point down and the long lead is 2nd from the left.

Then the pins are (R, GND, G, B) from left to right.

Wiring:

Long lead	GND
Leads 1, 3, 4	Arduino PWM-capable pins (i.e. 3, 5, 6)
	Through 100 ohm resistors

Programming:

Use **analogWrite()** to set the relative intensities of R/G/B.

The "RGB" color space is not intuitive for picking specific colors, so often an alternative such as "HSV" (Hue, Saturation, Value) is used. The algorithm for converting is well-documented here:

http://en.wikipedia.org/wiki/HSL_and_HSV#From_HSV

There is a sample implementation in my **Lab_4_4** example sketch.

- [ ] **Example 4.5 - Accelerometer**

The small red circuit board in your kit is a 3-axis accelerometer. It provides an analog output in the range 0-3.3V, for a +/-1.5G or +/-6G acceleration along each of 3 axis as marked on the PCB. Hooking it up requires a bit of soldering

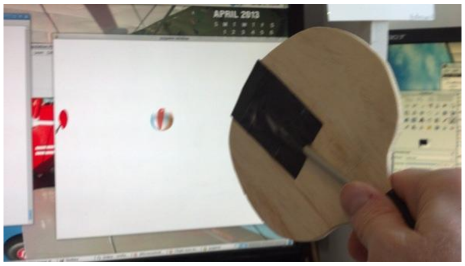
Wiring:

ST	GND
GSEL	GND
0GD	-n.c.-
SLP	+3.3V
XOUT	Arduino analog input i.e. A0
YOUT	Arduino analog input i.e. A1
ZOUT	Arduino analog input i.e. A2
GND	GND
VCC	3.3V

Note: be sure to use 3.3V (from the Arduino is fine), not 5V to power this board.

Programming:

Just read the analog inputs. Use your imagination!
I attached mine to a ping-pong paddle and made a "wii pong" game.
(the Arduino reads the accelerometer and senses paddle movement, then sends updates over the USB interface to a script using the python PyGame library)



- ▢ **Example 4.6 - Force Sensing Resistor**

The force-sensing resistor (user guide) is open-circuit with no force, and provides a resistance which roughly follows a power-law down to about 100 ohms.

Wiring:

Connect one terminal to GND
Connect the other to +5V through a resistor (10k is good)
and to an Arduino analog input.

Programming:

Just read the analog input. With a few of these and a bit of programming you could make a drum machine.

- ▢ **Example 4.7 - IR Remote Receiver**

This device receives IR remote control signals modulated at 38kHz, compatible with many popular remote controls. It detects a light signal through a narrow pass-band filter and provides a logic signal which is active when the modulated signal is present.

Wiring:

1	Arduino digital input
2	5V
3	GND

Programming:

To simply detect that any code is being received, just test the logic level with **digitalRead().** Decoding which button is pressed is complex, and it is best to use a library to capture and decode the pulse train. There is an excellent tutorial here:

http://learn.adafruit.com/ir-sensor/overview