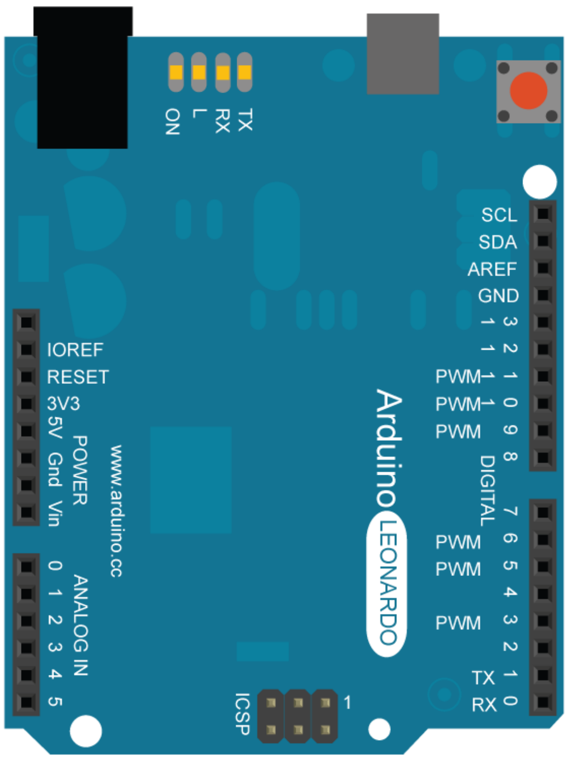
ESE 141:
INTRODUCTORY ROBOTICS

INTRODUCTION TO ARDUINO



WHAT IS ARDUINO?

http://www.arduino.cc/

Arduino is an open-source electronics prototyping platform/microcontroller based on flexible, easy-to-use hardware and software. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language. This class uses the *Leonardo* board, which is the most recent version. Arduino is super cool and easy to use. Make use of it when you bring it home!

Want to read more in depth about Arduino?
http://www.princeton.edu/~ffab/media___downloads_files/IntroArduinoBook.pdf

DOWNLOADING AND SETUP

The software to program the board can be downloaded below.
Arduino IDE (Integrated Development Environment) : http://arduino.cc/en/Main/Software
Downloading Guide: <link it here>

PINS

~ 5, ~6 – Motor control ~3 - Buzzer 7, 8, 12 – LED A0, A1 – Distance A2, A3 – Floor	Gnd – Ground 5V – Power
Yellow Wire	Power Wires (Black and Red)

PARTS OF AN ARDUINO PROGRAM

http://arduino.cc/en/Reference/HomePage

The Arduino language (based on Wiring) is implemented in C/C++, and therefore has some differences from the Processing language, which is based on Java. Arduino programs can be divided in three main parts: structure, values (variables and constants), and functions. The first thing you want to do in an Arduino program is to set up any variables you want to have access to throughout the entire program. The next part is the "setup" function that initializes the pins and defines what each pin does. Finally, we have a "loop" function, which is main function. This function is what runs the logic of the program. The process of the "loop" function runs over and over again until you stop or unplug the robot.

BLINKING CIRCUIT

This is a simple program for a blinking circuit. It turns on the LED for one second and then turns it off for one second.

```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led1 = 13;
int led2 = 12;
int led3 = 8;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  digitalWrite(led3, HIGH);
  delay(1000);
  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);
  digitalWrite(led3, LOW);
  delay(1000);               // wait for a second
}
```

BUZZER

The buzzer turns on when button is being pressed.

```
// set pin numbers:
const int buzzer = 3;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
const int ledPin2 =  12;      // the number of the LED pin
const int buttonPin = 1;     // the number of the pushbutton pin

int buttonState = 0;         // variable for reading the
                                //pushbutton status

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(buttonPin, OUTPUT);

  digitalWrite(ledPin2, HIGH);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    tone(buzzer, 400);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
    tone(buzzer, 0);
  }
}
```

BUTTON

Turns on and off a light emitting diode(LED) connected to digital pin 13, when pressing a pushbutton attached to pin 2. Note that on most Arduinos there is already an LED on the board attached to pin 13.

The circuit:
* LED attached from pin 13 to ground
* pushbutton attached to pin 2 from +5V
* 10K resistor attached to pin 2 from ground

```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 1;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the
pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

SHARP SENSOR

This is an analogue serial reader. It reads an analog input on pin 0, prints the result to the serial monitor (accessed with Ctrl-Shift-M). Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

```
int led1 = 13;

// the setup routine runs once when you press reset:
void setup() {
  pinMode(led1, OUTPUT);
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  digitalWrite(led1, HIGH);
}

// the loop routine runs over and over again forever:
void loop() {
  int leftSharp = analogRead(A0);
  int rightSharp = analogRead(A1);

  // print out the value you read:
  Serial.print(leftSharp, DEC);
  Serial.print(", ");
  Serial.println(rightSharp, DEC);
  delay(100);        // delay in between reads for stability
}
```

Functions
http://arduino.cc/en/Reference/Functio
nDeclaration

RUNNING THE PROGRAM

Copy and paste the code into a new Arduino file.
Choose Tools > Board > Arduino Leonardo
Choose Tools > Serial Port > USB Port #
Verify (Click the circle checkmark button)
Plug in the robot to your computer.
Upload (Click the circle sideways arrow button)

The program in on your robot now!
You can detach the micro USB cable now, if you have batteries powering your robot.

You can watch the Serial monitor to see your print statements. Click **Tools > Serial Monitor** to watch the variables. You can also press **Ctrl-Shift-M**.