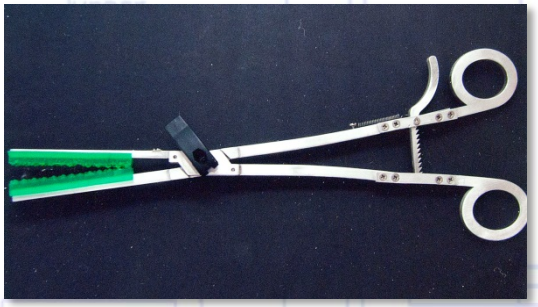
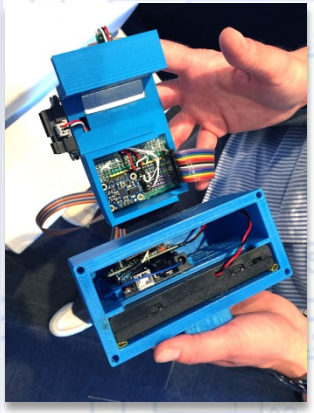
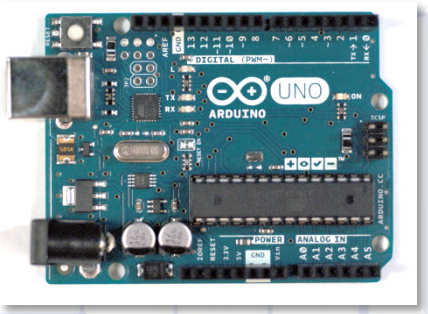
Introductory Medical Device Prototyping

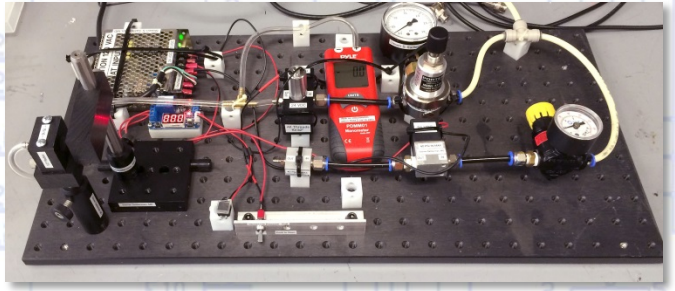
Arduino Part 3

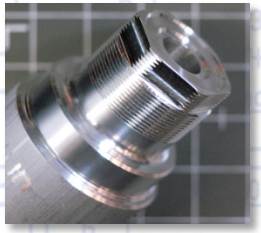
Prof. Steven S. Saliterman, http://saliterman.umn.edu/
Department of Biomedical Engineering, University of Minnesota











Topics

- More functions:
  - Math functions
  - Trigonometry functions
  - Random numbers
  - Interrupts
    - Examples – Button push and an 8 bit counter.
  - Sensor example
- Arduino Lab 1

Math Functions: abs()

- abs(x)
  - Returns:
    - x if x >= 0
      - e.g. x=5, then abs(x)=5
    - -x if x is < 0
      - e.g. ix=-4, then abs(x)=4

max()…

- max(x, y)
  - Returns the larger of x vs. y
    - e.g. sensVal = max(sensVal, 20);
      - Assigns sensVal to the larger of sensVal or 20 .
      - In this example effectively ensuring that it is at least 20.

min()…

- min(x, y)
  - Returns the smaller of two numbers
    - e.g. sensVal = min(sensVal, 100);
      - Assigns sensVal to the smaller of sensVal or 100.
      - In this example, ensuring that it never gets above 100.

Constrain()…

- constrain(x, a, b) - Constrains a number to be within a range.
  - x: the number to constrain, all data types
  - a: the lower end of the range, all data types
  - b: the upper end of the range, all data types
  - Returns
    - x: if x is between a and b
    - a: if x is less than a
    - b: if x is greater than b
  - For example, if *sensVal* was read as 140 from a sensor, the values assigned to *sensVal* in each case would be:
    - sensVal = constrain(sensVal, 10, 150), answer 140
    - sensVal = constrain(sensVal, 50, 100), answer 100
    - sensVal = constrain(sensVal, 150, 250), answer 150

map()…

- map(value, fromLow, fromHigh, toLow, toHigh)
  - Re-maps a number from one range to another. That is, a value of fromLow would get mapped to toLow, a value of fromHigh to toHigh, values in-between to values in-between, etc..
  - Use integer math (fractions are truncated).
  - Example:
    - void setup() {}
      void loop()
      {
          int val = analogRead(0);
          val = map(val, 0, 1023, 0, 255);
          analogWrite(9, val);
      }

pow()…

- pow(base, exponent) - Value of a number raised to a power.
  - base: the number (float), exponent: the power to which the base is raised.
  - Example (serial print a binary sequence, 2 to the power of 0…49)
    - float i = 0;
      long result;

      void(setup){
        Serial.begin(9600L);
      }

      void(loop){
        Serial.println("Base 2: ");
        for(i=0; i<50; i++){
          result=pow(2 ,i);
          Serial.println(result)
      }

sq() and sqrt()…

- sq(x)
  - The square of the number: x^2
    - Returns a double
- sqrt(x)
  - Calculates the square root of a number.
    - Returns a double

Trigonometry

- sin(rad) - Calculates the sine of an angle (in radians). The result will be between -1 and 1.
  - rad: the angle in radians (*float*)
  - returns the sine of the angle (*double)*
- cos(rad) - Calculates the cos of an angle (in radians). The result will be between -1 and 1.
  - rad: the angle in radians (float)
  - Returns the cos of the angle ("double")
- tan(rad) - Calculates the tangent of an angle (in radians). The result will be between negative infinity and infinity.
  - rad: the angle in radians (float)
  - Returns the tangent of the angle (double)

Random Numbers

- randomSeed(seed)
  - Initializes the pseudo-random number generator, causing it to start at an arbitrary point in its random sequence.
  - This sequence is always the same.
  - For a better random seed, initialize with analogRead() on an unconnected pin.
- random(max) and random(min, max) are variations.

Interrupts

- Responsible for making the processor respond to important events.
  - This could be a button push, or other external event that you need react to immediately.
  - With the appropriate signal, the processor turns its attention to executing specific code related to the event.
  - The goal of an external interrupt is to not miss the event!
  - Two external interrupt pins: INT0 and INT1 which map to Arduino Uno pins 2 and 3.

ISR…

- Interrupt Service Routines (ISR)
  - Fast, short code that is executed with an interrupt.
  - Only one can run at a time.
  - Takes no parameters and returns none.
    - Use global variables to pass information between the main program and the ISR.

Implementing an ISR…

- Required code:
    - attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)

  (Pin number, name of ISR, and mode - when should the interrupt be triggered.)
- Interrupt trigger “modes:”
    - LOW to trigger the interrupt whenever the pin is low.
    - CHANGE to trigger the interrupt whenever the pin changes value.
    - RISING to trigger when the pin goes from low to high.
    - FALLING for when the pin goes from high to low.
    - HIGH to trigger the interrupt whenever the pin is high. (Due, Zero and MKR1000 boards)

About Pre- and Post- Operators…

- n++ (post) returns the current value n and then increments n.
- ++n (pre) increments n and returns the new value of n.

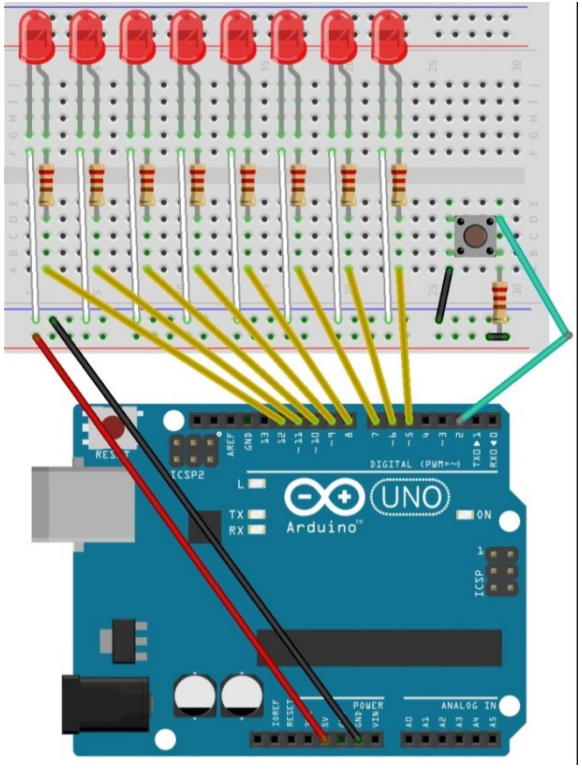
Interrupt Example…

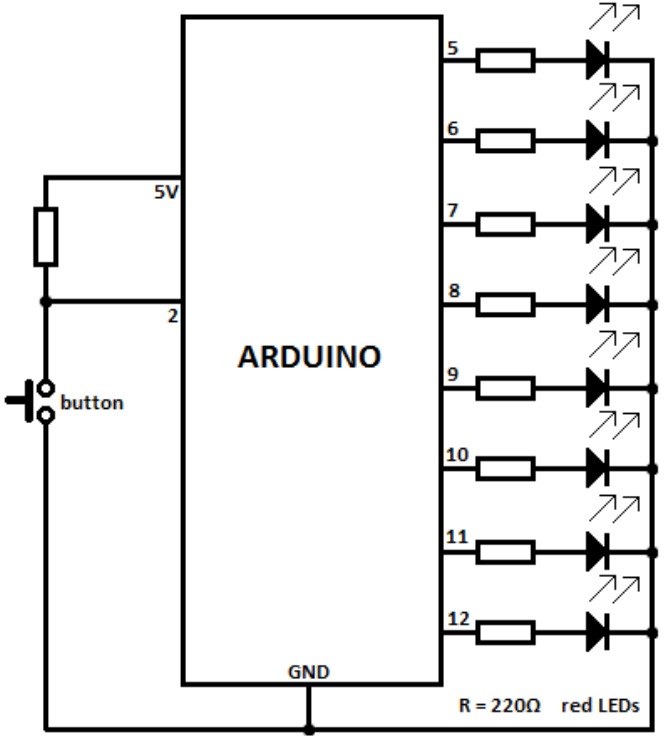
Task: A button on digital pin 2 is made to turn on an LED on pin 13.

```
const byte LED = 13;
const byte BUTTON = 2;

// Interrupt Service Routine (ISR)
void switchPressed ()
{
  if (digitalRead (BUTTON) == HIGH)
    digitalWrite (LED, HIGH);
  else
    digitalWrite (LED, LOW);
}  // end of switchPressed

void setup ()
{
  pinMode (LED, OUTPUT);  // so we can update the LED
  digitalWrite (BUTTON, HIGH);  // internal pull-up resistor
  attachInterrupt (digitalPinToInterrupt (BUTTON), switchPressed, CHANGE);  // attach interrupt handler
}  // end of setup

void loop ()
{
  // loop doing nothing
}
```


Gammon, N. http://gammon.com.au/interrupts

Example: 8-Bit Binary Counter





Marian, P. Arduino 8 bit Binary LED Counter. www.electroschematics.com

Binary Counter…

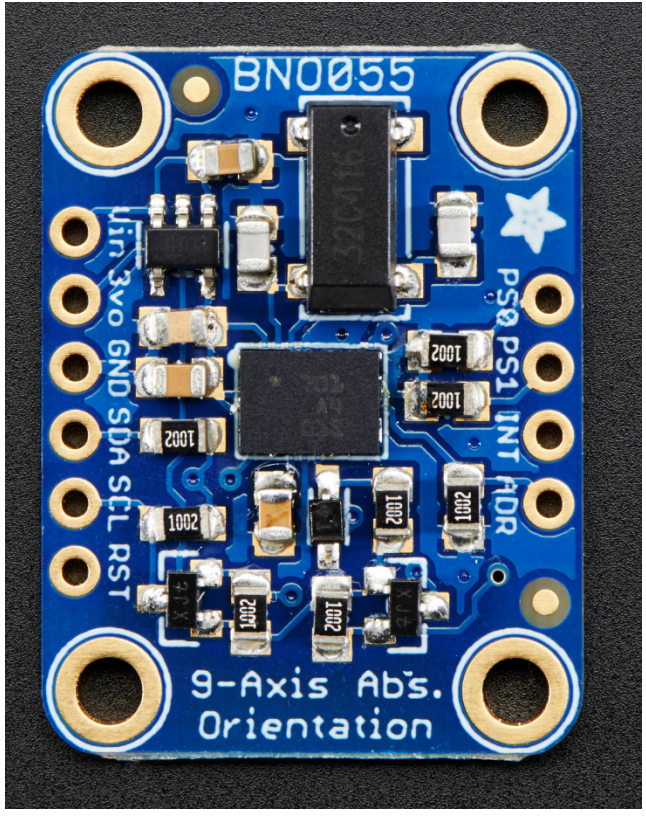
```
int button = 2;                          // pin to connect the button
int presses = 0;                         // variable to store number of
presses
long time = 0;                           // used for debounce
long debounce = 100;                     // how many ms to "debounce"
const byte numPins = 8;                  // how many LEDs
int state;                               // used for HIGH or LOW
byte pins[] = {5, 6, 7, 8, 9, 10, 11, 12};          // LED Pins

void count()  // function count the button presses
{
if(millis() - time > debounce)  presses++; //debounce pushbutton
 time = millis();
}

void setup()
{
for(int i = 0; i < numPins; i++)         // set LED pins to outputs
 {
   pinMode(pins[i], OUTPUT);
 }
 pinMode(button, INPUT);
 attachInterrupt(0, count, LOW);     // pin 2 is interrupt 0 on UNO
}
```

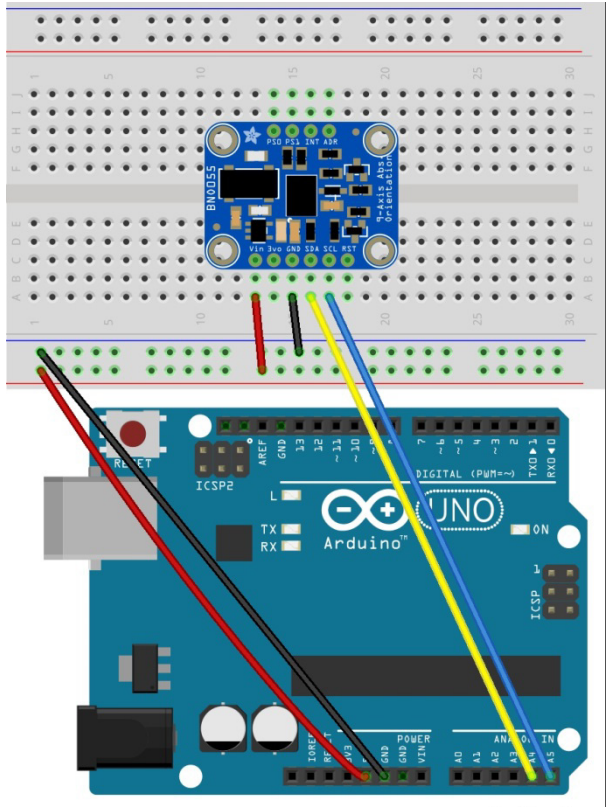
```
void loop()
{
 /* convert presses to binary and store it as a string */
 String binNumber = String(presses, BIN);

int binLength = binNumber.length();  //get length of string

  if(presses <= 255)  // if we have less or equal to 255 presses
  {
   for(int i = 0, x = 1; i < binLength; i++, x+=2)
    {
      if(binNumber[i] == '0') state = LOW;
      if(binNumber[i] == '1') state = HIGH;
      digitalWrite(pins[i] + binLength - x, state);
    }

   // do something when we reach 255
  }
}

// http://www.electroschematics.com/9809/arduino-8-bit-binary-LED
```

Marian, P. Arduino 8 bit Binary LED Counter. www.electroschematics.com

Sensor Example

Absolute Orientation Sensor BNO005





Left –image courtesy of Adafruit

Arduino Serial Monitor…

COM5 (Arduino/Genuino Uno)

Send

```
0               CALIBRQ¥Orientation Sensor Raw Data Test

Current Temperature: 32 C

Calibration status values: 0=uncalibrated, 3=fully calibrated
X: 0.00 Y: 0.00 Z: 0.00         CALIBRATION: Sys=0 Gyro=0 Accel=0 Mag=0
X: 0.00 Y: 0.00 Z: 0.00         CALIBRATION: Sys=0 Gyro=0 Accel=0 Mag=0
X: 0.00 Y: 0.00 Z: 0.00         CALIBRATION: Sys=0 Gyro=0 Accel=0 Mag=0
X: 0.00 Y: 0.00 Z: 0.00         CALIBRATION: Sys=0 Gyro=0 Accel=0 Mag=0
X: 0.00 Y: 0.00 Z: 0.00         CALIBRATION: Sys=0 Gyro=0 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
X: 0.00 Y: -0.94 Z: 0.00                CALIBRATION: Sys=0 Gyro=3 Accel=0 Mag=0
```

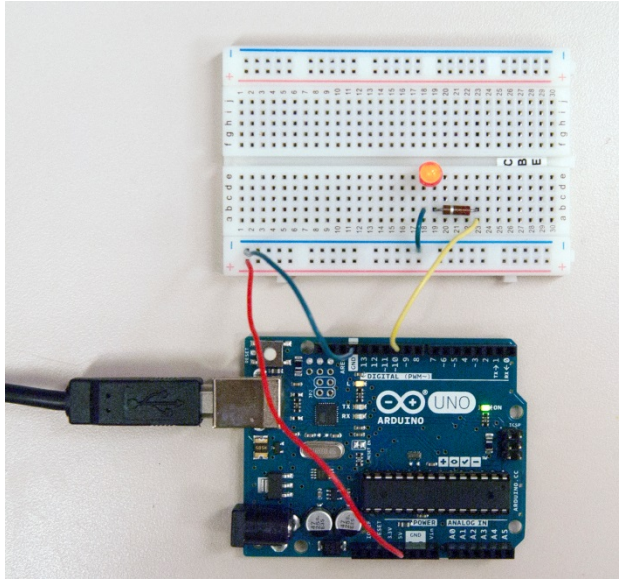
Arduino Lab

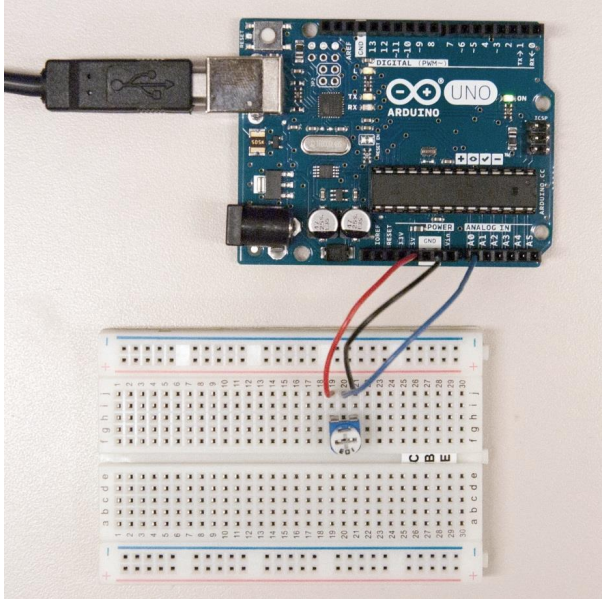
Prof. Steven S. Saliterman

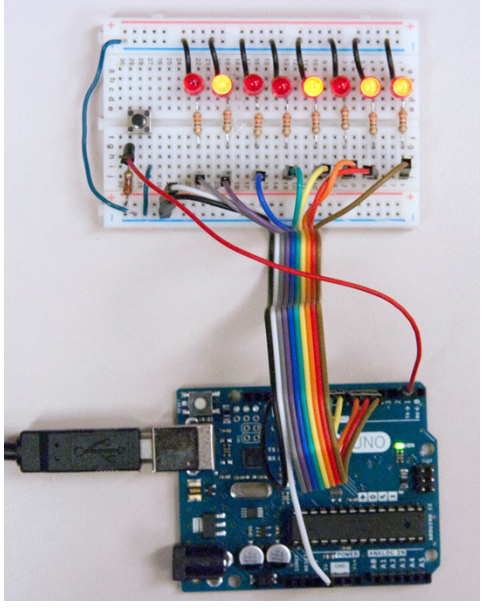
Jumper Wires, Cables & Power Adaptor…

Power Adapter
for Arduino

Breadboarding Technique…







Summary

- More functions:
  - Math functions
  - Trigonometry functions
  - Random numbers
  - Interrupts
    - Examples – Button push and an 8 bit counter.
  - Sensor example
- Arduino Lab 1