week 03

Digital Input and Output

RGB LEDs fade with PWM

Digital vs. Analog

Digital signal

Difference between two possible states

“whether or not”

Is the cat on the mat or not?

Analog signal

continuous range of possible states

“how much” or “stronger” “faster” “brighter”

How heavy is the cat that’s on the mat?

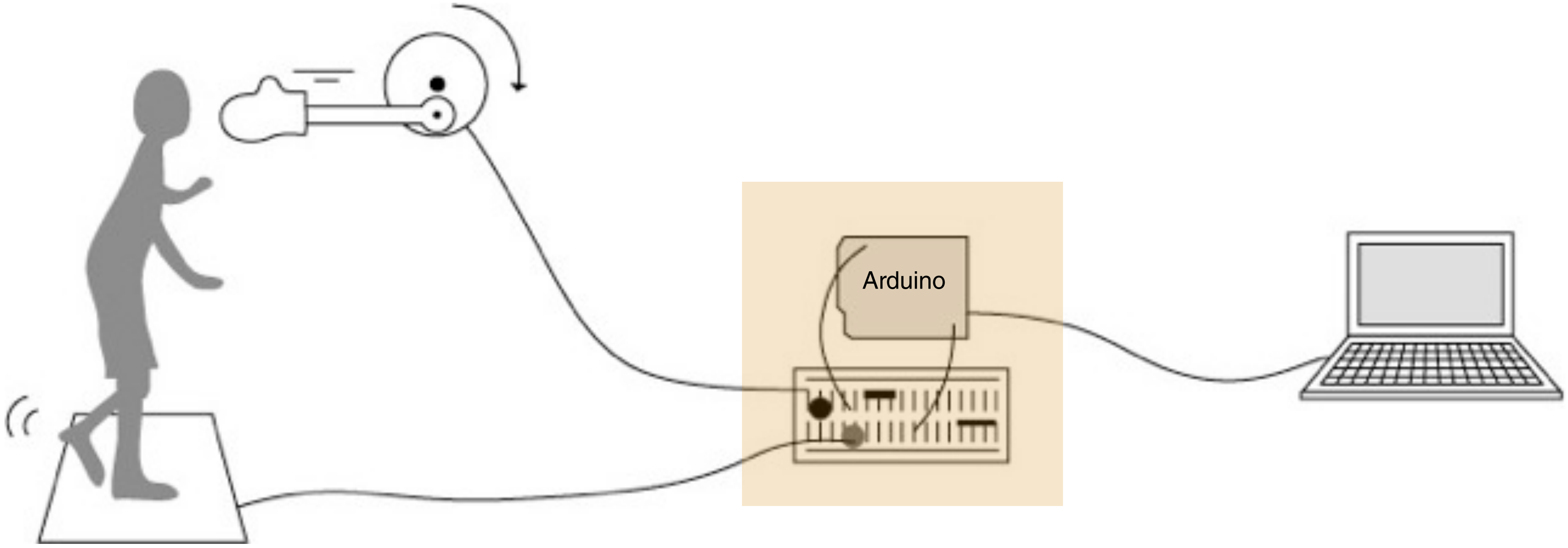
Digital vs. Analog

Binary vs. continuous signals

- Binary / Digital = “whether or not”
- Continuous / Analog signal = “how much” or “faster,” “brighter,” etc.

Microcontrollers

Output Transducers
actuators (e.g.,
motors, buzzers)



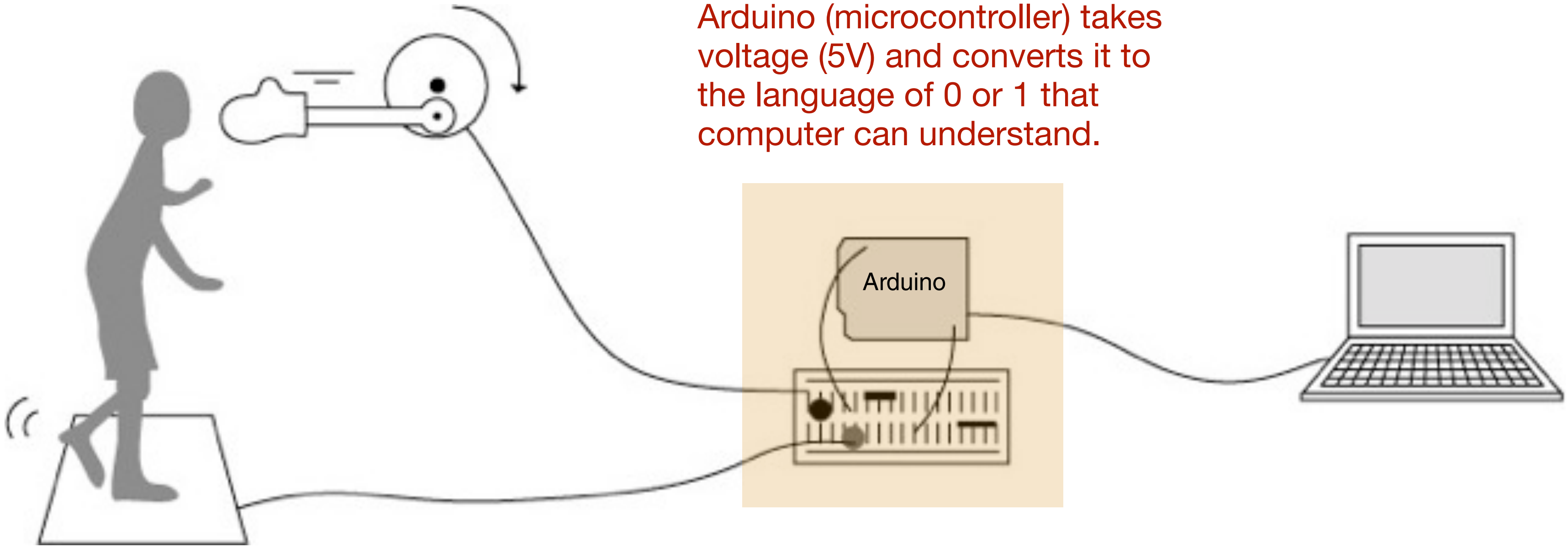
Input Transducers
sensors (e.g., switches,
levers, sliders, etc.)

Illustration adapted and modified from O’Sullivan and Igoe

Microcontrollers

Output Transducers
actuators (e.g., motors, buzzers)

Arduino (microcontroller) takes voltage (5V) and converts it to the language of 0 or 1 that computer can understand.



Input Transducers
sensors (e.g., switches, levers, sliders, etc.)

Illustration adapted and modified from O'Sullivan and Igoe

Digital vs. Analog

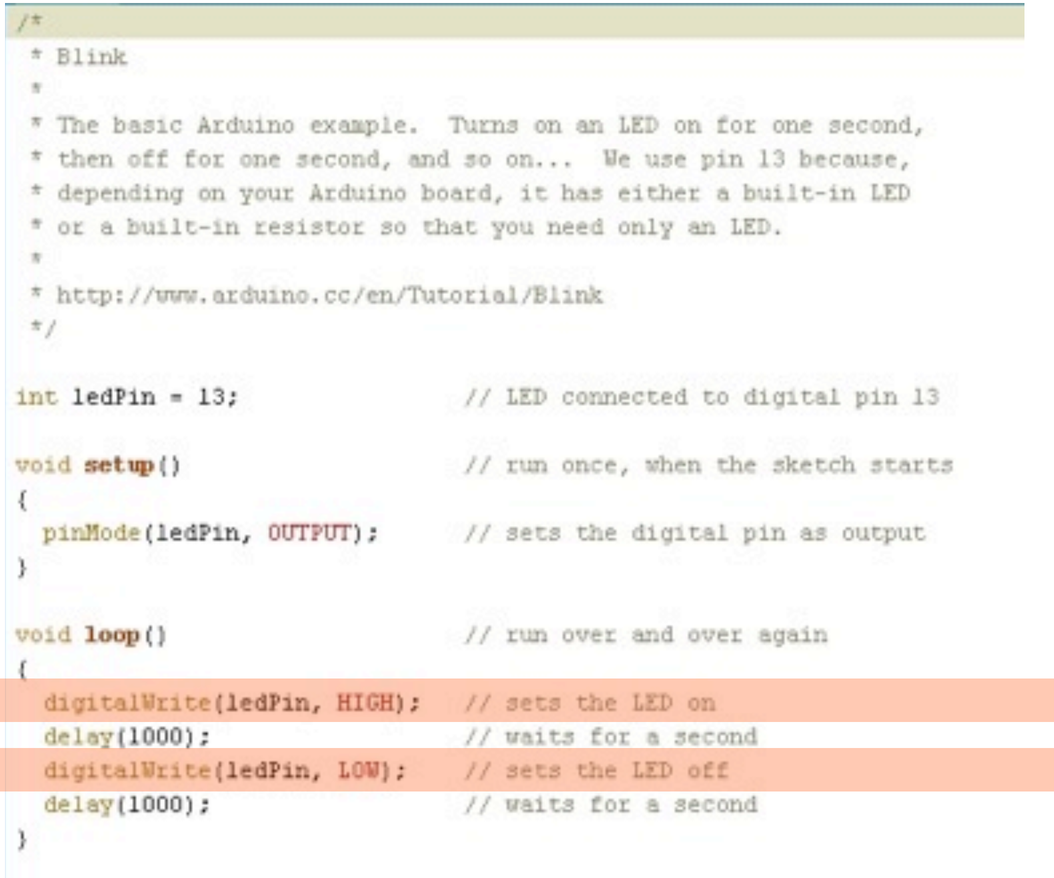
Internally, all microprocessors compute binary: 0 or 1 (0V or 5V)

In general, most microprocessors output only binary (0V or 5V)

Specifically, Arduino output pins can only be LOW (0V) or HIGH (5V)

Digital Output

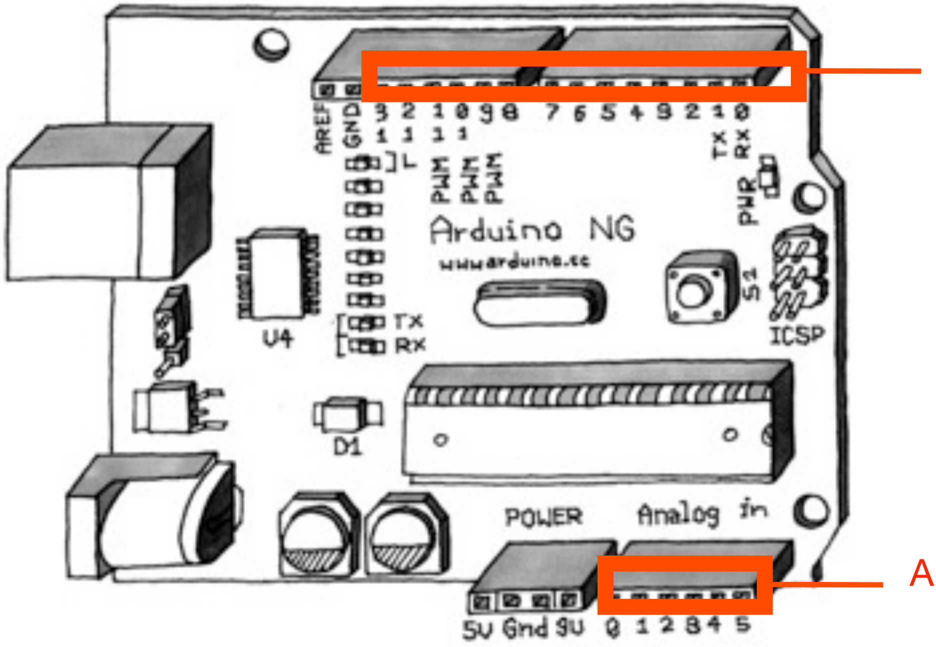
Blinking LED



```
/*
 * Blink
 *
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

Digital vs. Analog

Two states (binary signal) vs. multiple states (continuous signal)



Digital Input and Output

Analog Input (next week!)

Can We Do Analog Out?

LED with 23% brightness?

23% brightness

100% brightness

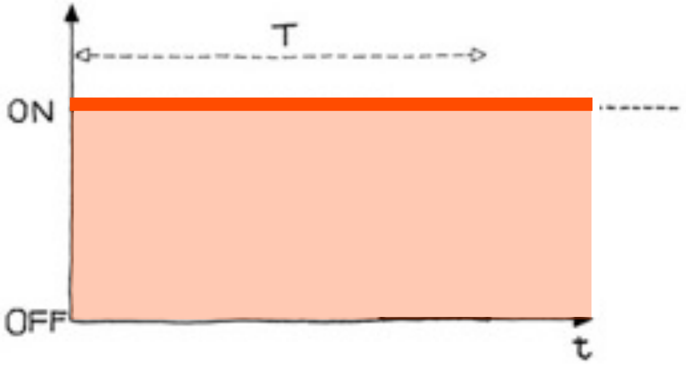
Pulse Width Modulation (PWM)

Most microprocessors can only output binary: LOW (0V) or HIGH (5V)

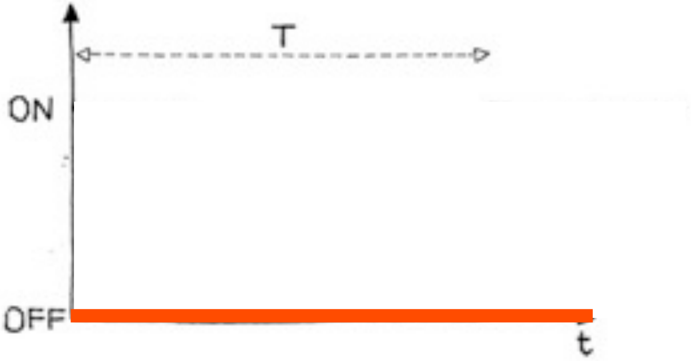
So you fake it with PWM, Pulse Width Modulation

It gives you an illusion of analog values, in between LOW and HIGH

PWM

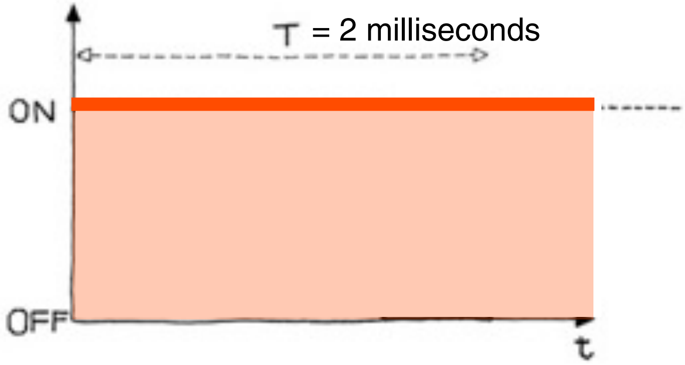


100% brightness

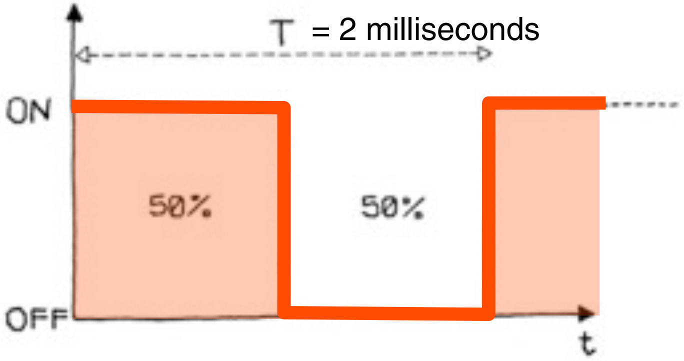


0% brightness

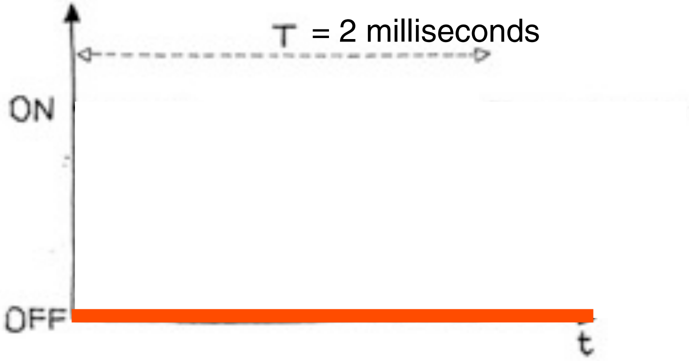
PWM



100% brightness

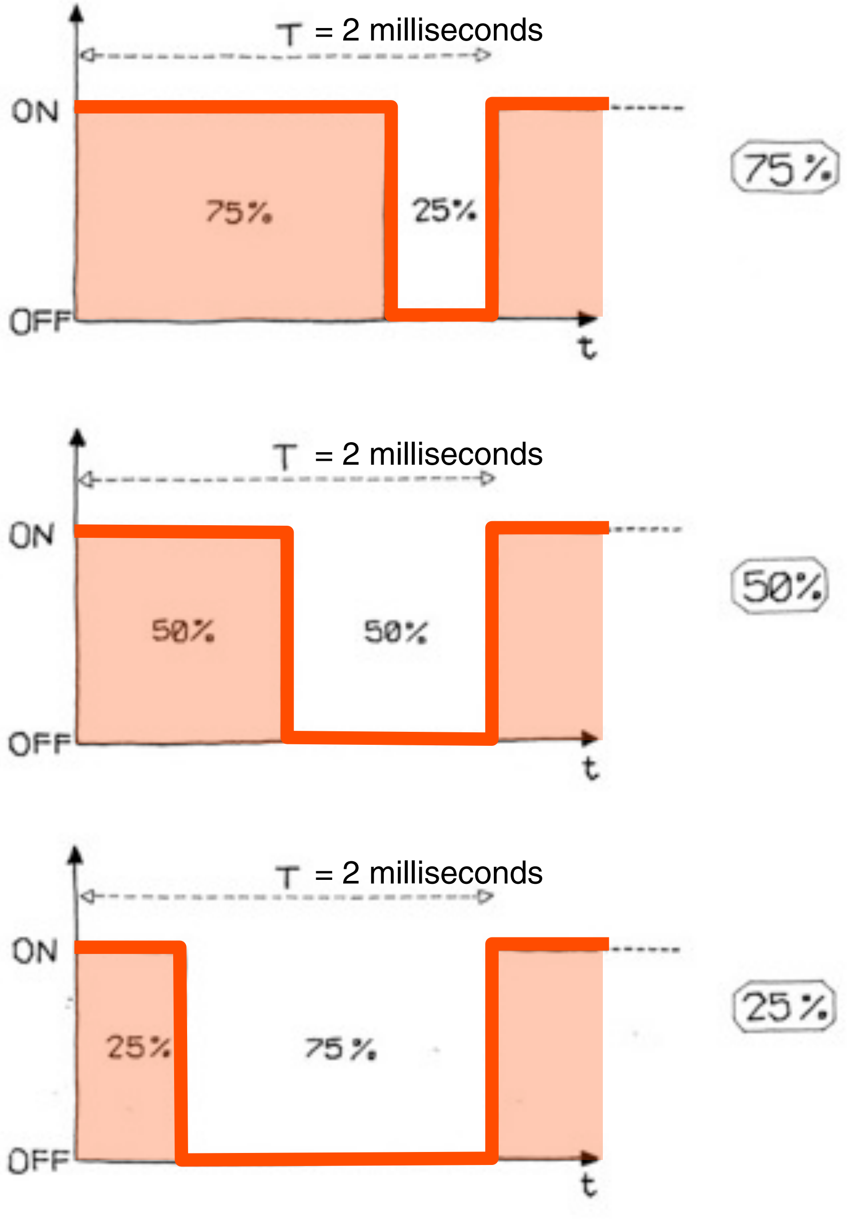


50% brightness

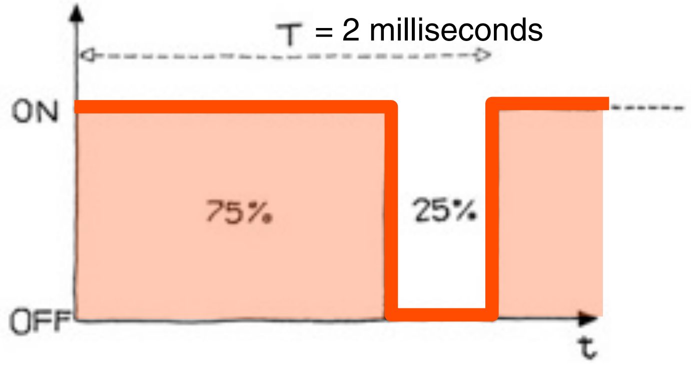


0% brightness

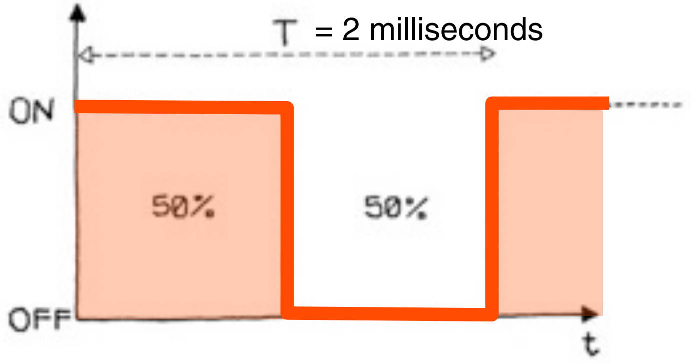
PWM


T = 2 milliseconds
ON
OFF
75%
25%
t
75%
T = 2 milliseconds
ON
OFF
50%
50%
t
50%
T = 2 milliseconds
ON
OFF
25%
75%
t
25%

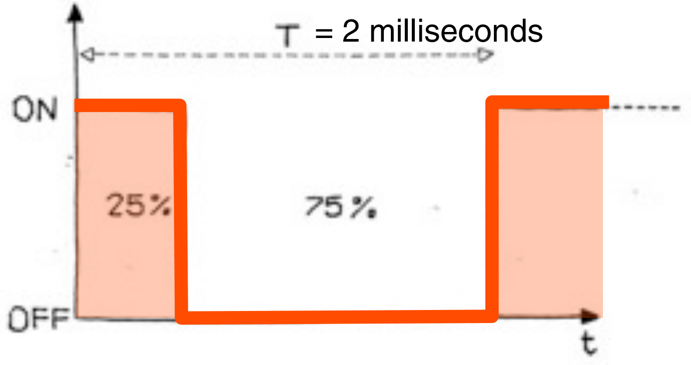
PWM


T = 2 milliseconds

75% brightness


T = 2 milliseconds

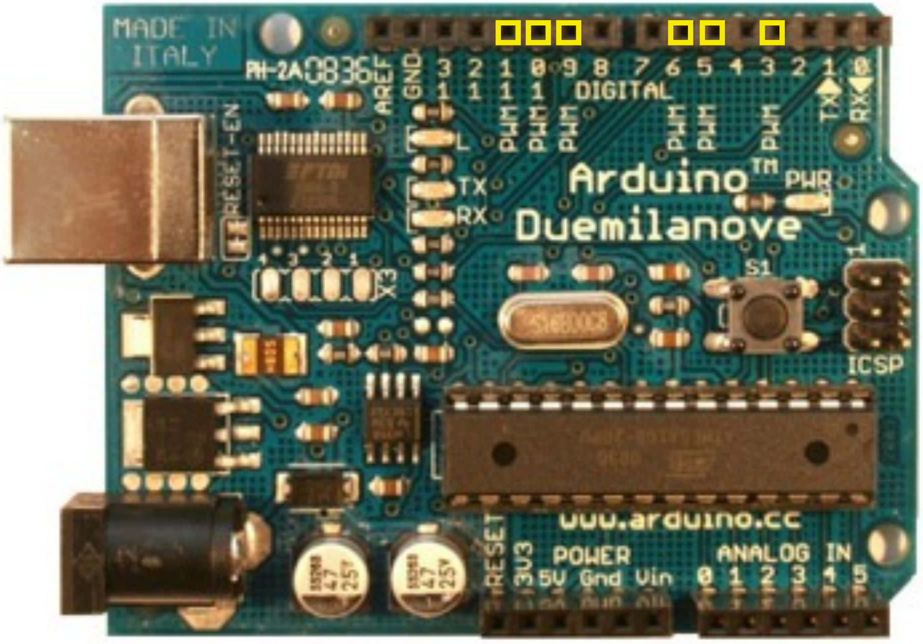
50% brightness


T = 2 milliseconds

25% brightness

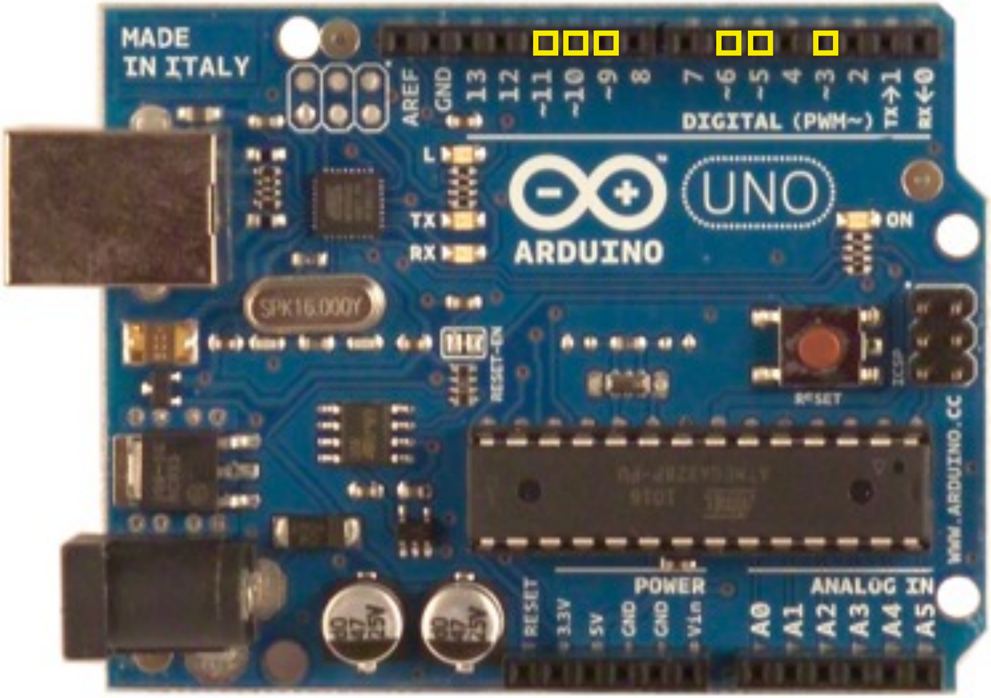
Pulse Width Modulation (PWM)

Your Arduino board has built in PWM circuits,
on pins 3, 5, 6, 9, 10, and 11



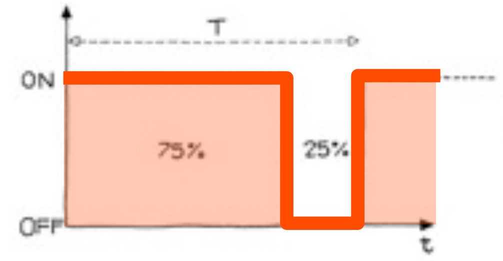
Pulse Width Modulation (PWM)

Your Arduino board has built in PWM circuits,
on pins 3, 5, 6, 9, 10, and 11



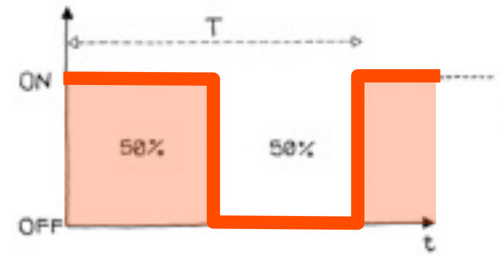
analogWrite(pin, value)

The duty cycle: between 0 and 255



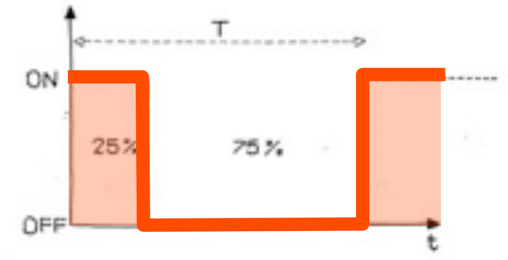
analogWrite(11, 191)

75% brightness



analogWrite(11, 127)

50% brightness



analogWrite(11, 64)

25% brightness

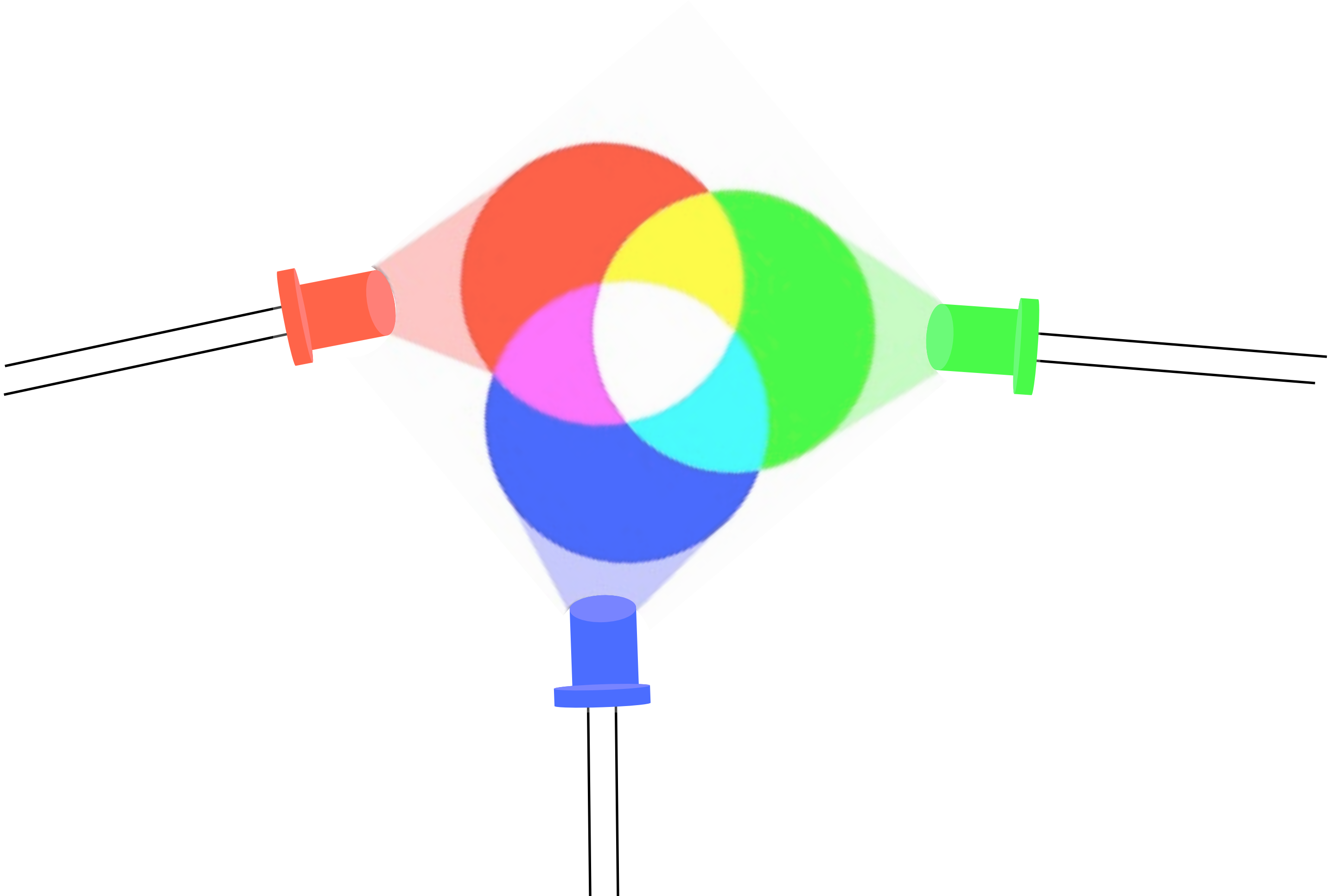
In Class Exercise

Make a color mixer with RGB LEDs

Exercise with digital input and output, and PWM

Color Mixer with RGB LEDs

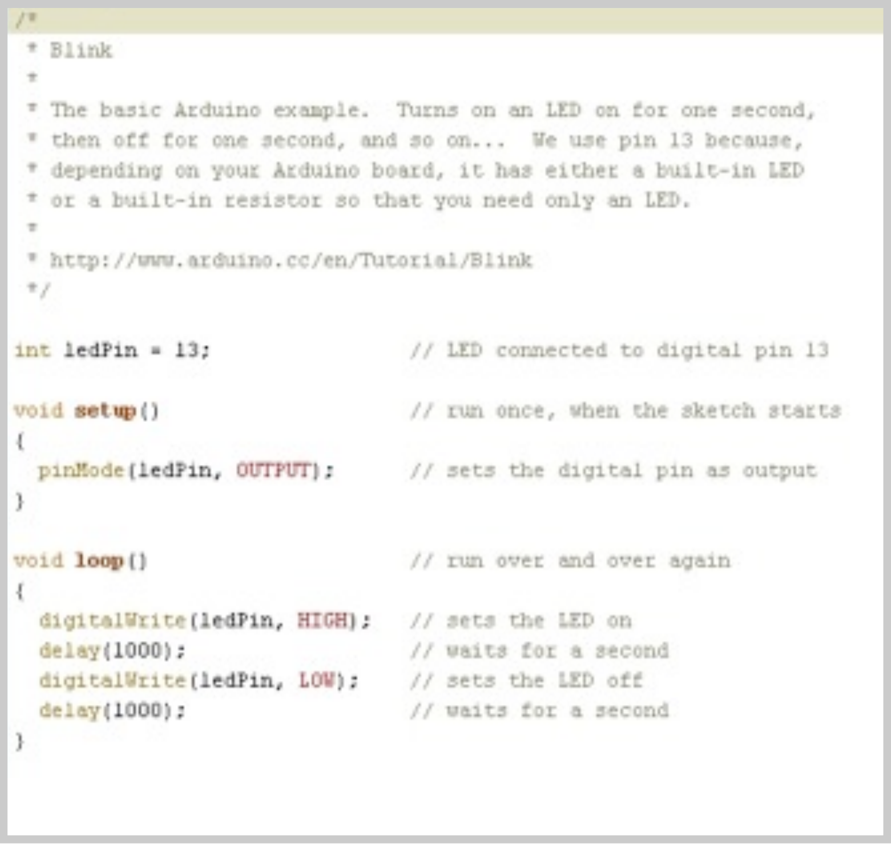
Make any colors with Red, Green, and Blue LEDs, except black



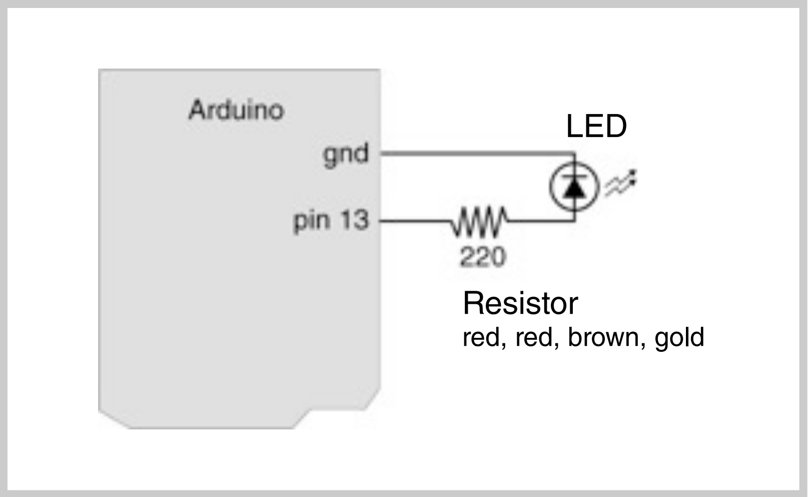
In Class Exercise

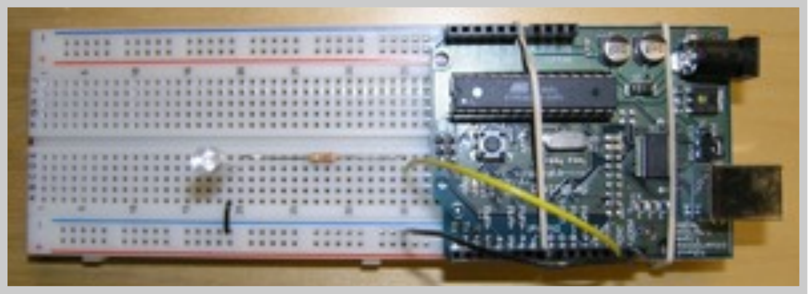
1. LED Blink
2. LED Fade
3. Circuit with 3 LEDs
4. RGB LED Fade
5. Serial RGB LED

Blinking LED (c.f. homework)



```
/*
 * Blink
 *
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

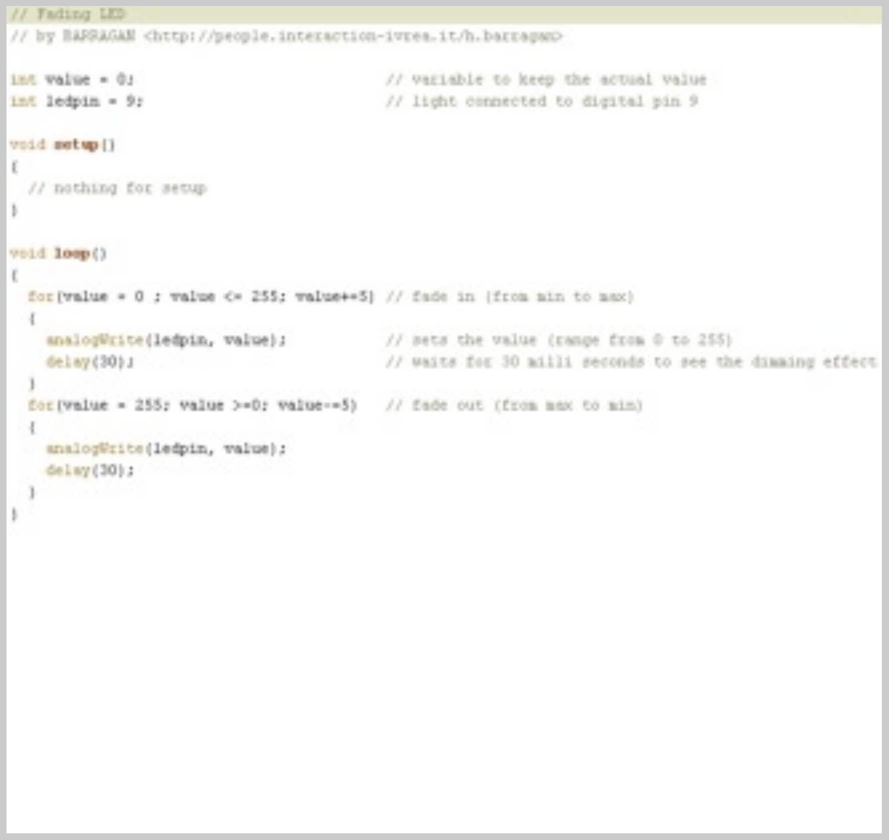




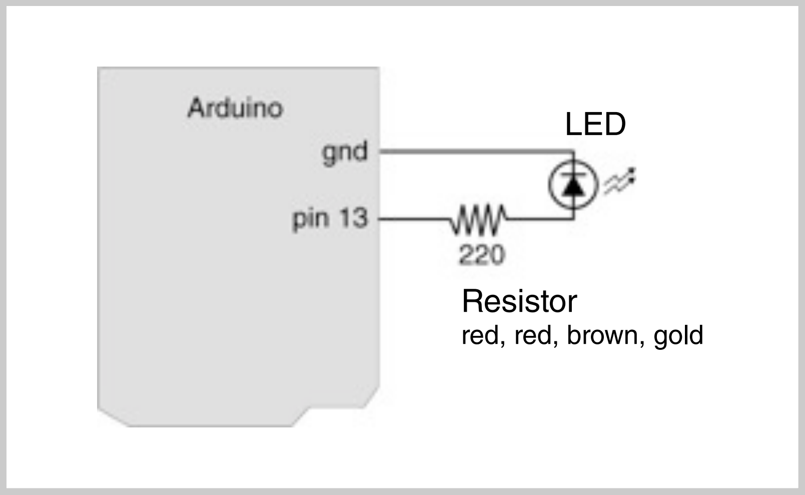
In Class Exercise

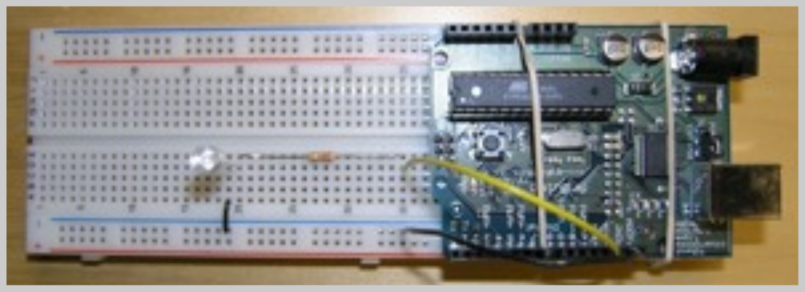
1. LED Blink
2. **LED Fade**
3. Circuit with 3 LEDs
4. RGB LED Fade
5. Serial RGB LED

LED Fade



```
// Fading LED
// by BARRAGAN <http://people.interaction-ivrea.it/h.barragan>

int value = 0;                            // variable to keep the actual value
int ledpin = 9;                           // light connected to digital pin 9

void setup()
{
  // nothing for setup
}

void loop()
{
  for(value = 0 ; value <= 255; value+=5) // fade in (from min to max)
  {
    analogWrite(ledpin, value);           // sets the value (range from 0 to 255)
    delay(30);                            // waits for 30 milli seconds to see the dimming effect
  }
  for(value = 255; value >=0; value-=5)   // fade out (from max to min)
  {
    analogWrite(ledpin, value);
    delay(30);
  }
}
```



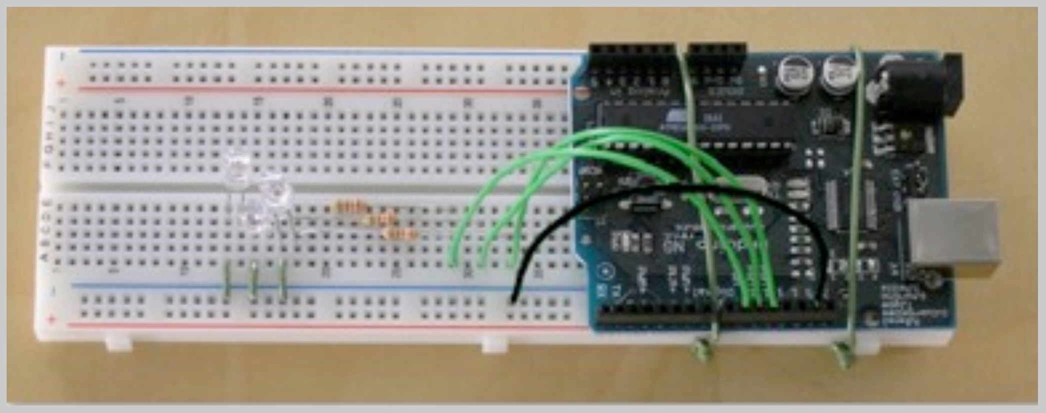


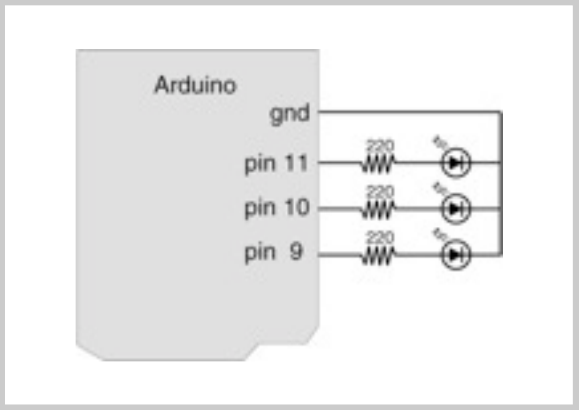
In Class Exercise

1. LED Blink
2. LED Fade
3. **Circuit with 3 LEDs**
4. RGB LED Fade
5. Serial RGB LED

Circuit with 3 LEDs

Plug three LEDs, red, green, and blue and make different colors



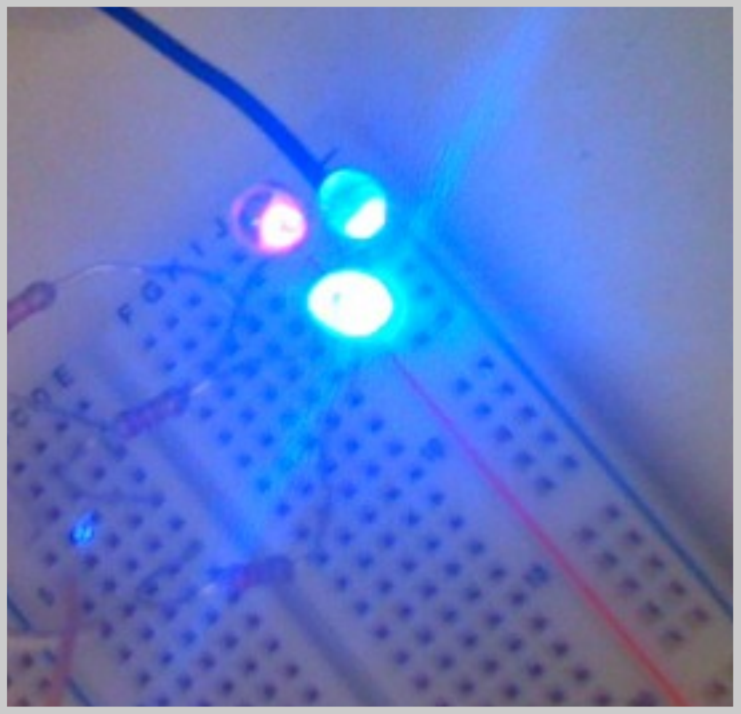


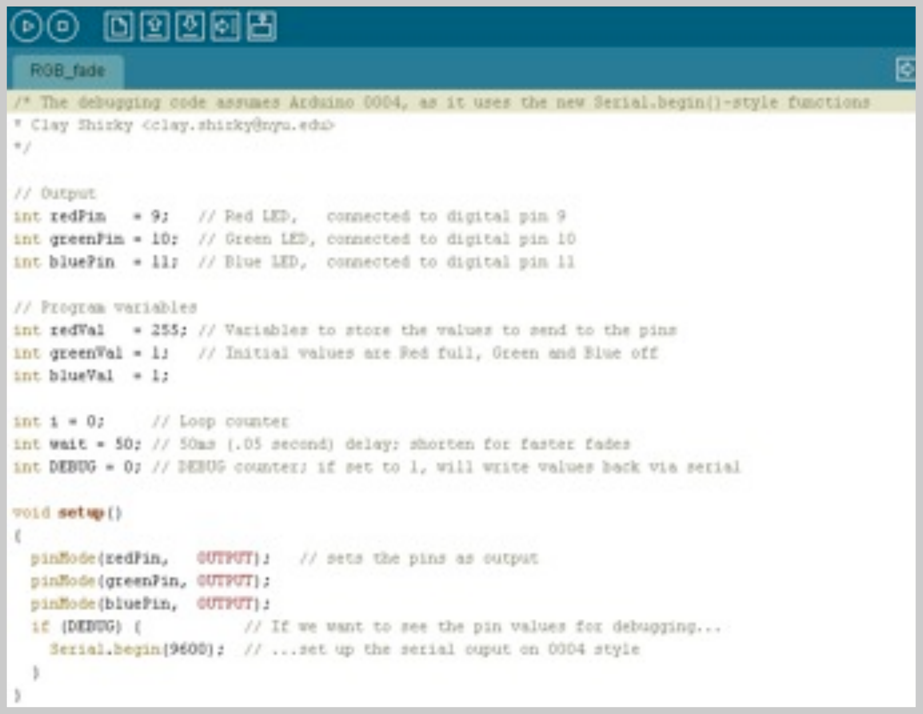
In Class Exercise

1. LED Blink
2. LED Fade
3. Circuit with 3 LEDs
4. **RGB LED Fade**
5. Serial RGB LED

RGB LED Fade

Slow color fading and mixing

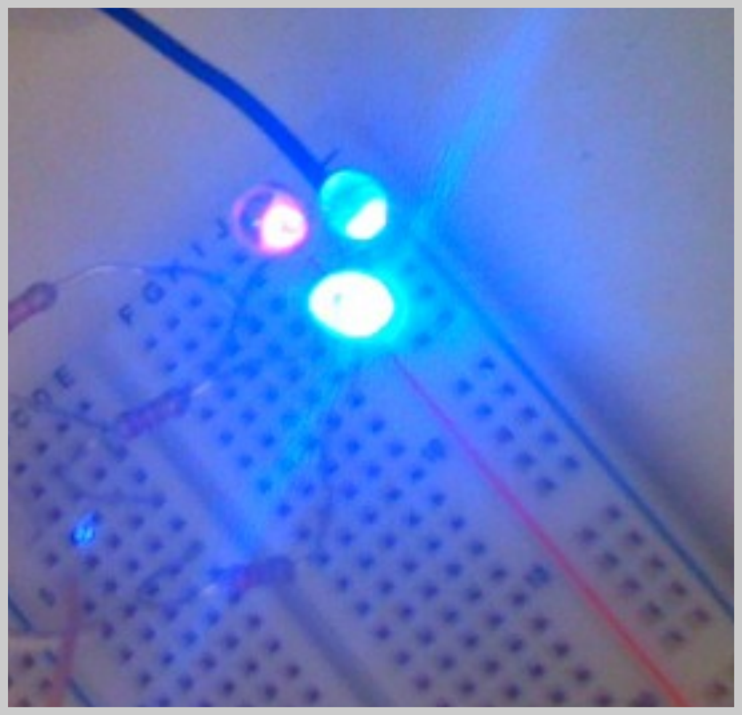


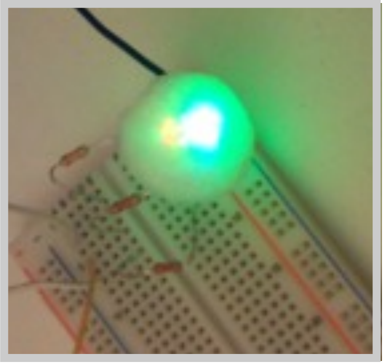


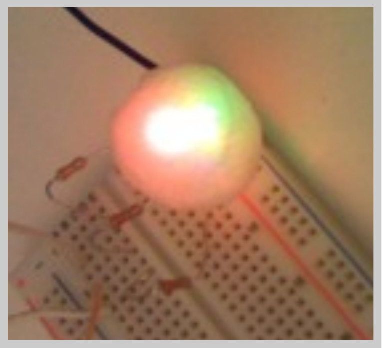
```
/* The debugging code assumes Arduino 0004, as it uses the new Serial.begin()-style functions
* Clay Shirky <clay.shirky@nyu.edu>
*/

// Output
int redPin   = 9;   // Red LED,   connected to digital pin 9
int greenPin = 10;  // Green LED, connected to digital pin 10
int bluePin  = 11;  // Blue LED,  connected to digital pin 11

// Program variables
int redVal   = 255; // Variables to store the values to send to the pins
int greenVal = 1;   // Initial values are Red full, Green and Blue off
int blueVal  = 1;

int i = 0;      // Loop counter
int wait = 50; // 50ms (.05 second) delay; shorten for faster fades
int DEBUG = 0; // DEBUG counter; if set to 1, will write values back via serial

void setup()
{
  pinMode(redPin,   OUTPUT);   // sets the pins as output
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin,  OUTPUT);
  if (DEBUG) {           // If we want to see the pin values for debugging...
    Serial.begin(9600);  // ...set up the serial ouput on 0004 style
  }
}
```

Diffuser

Experiment with different materials.



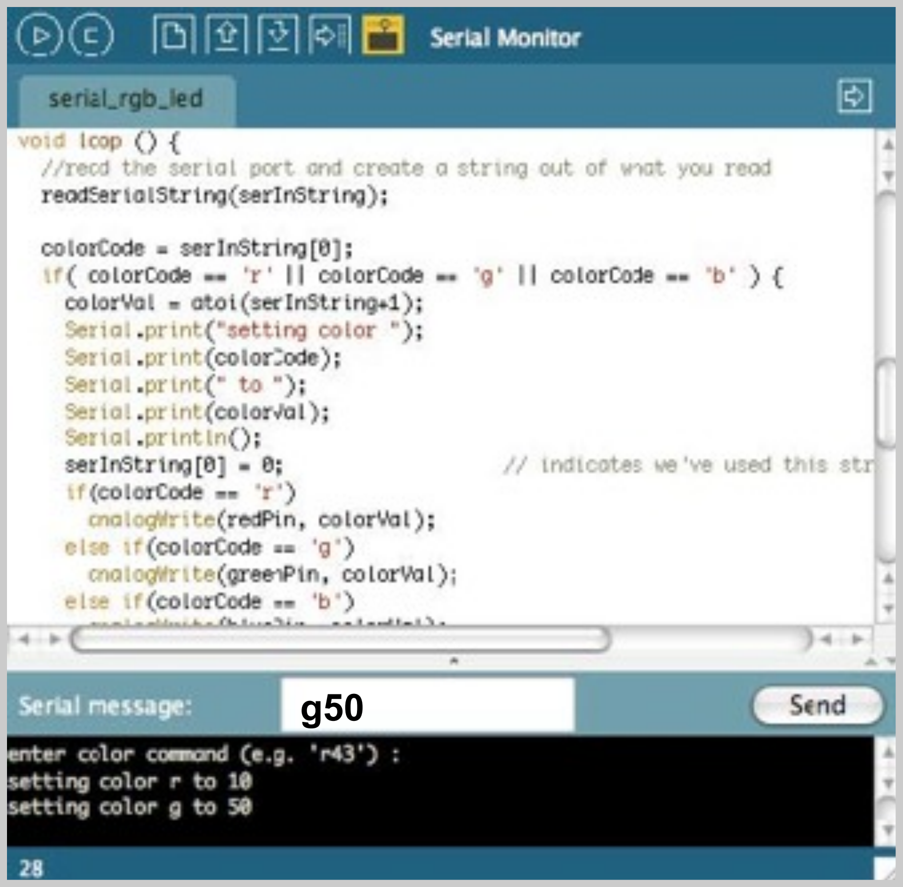




In Class Exercise

1. LED Blink
2. LED Fade
3. Circuit with 3 LEDs
4. RGB LED Fade
5. **Serial RGB LED**

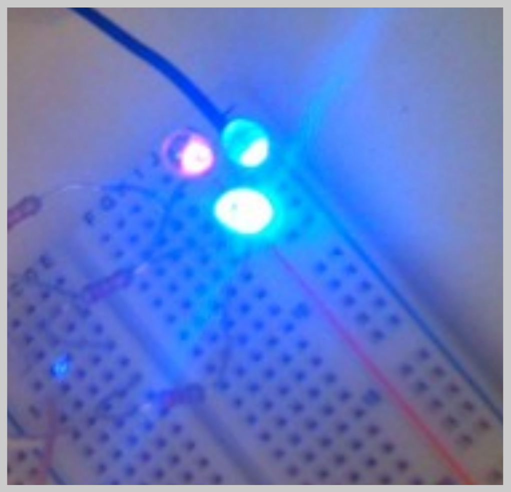
Serial RGB



```
void loop () {
  //read the serial port and create a string out of what you read
  readSerialString(serInString);

  colorCode = serInString[0];
  if( colorCode == 'r' || colorCode == 'g' || colorCode == 'b' ) {
    colorVal = atoi(serInString+1);
    Serial.print("setting color ");
    Serial.print(colorCode);
    Serial.print(" to ");
    Serial.print(colorVal);
    Serial.println();
    serInString[0] = 0;                   // indicates we've used this str
    if(colorCode == 'r')
      analogWrite(redPin, colorVal);
    else if(colorCode == 'g')
      analogWrite(greenPin, colorVal);
    else if(colorCode == 'b')
```

g50

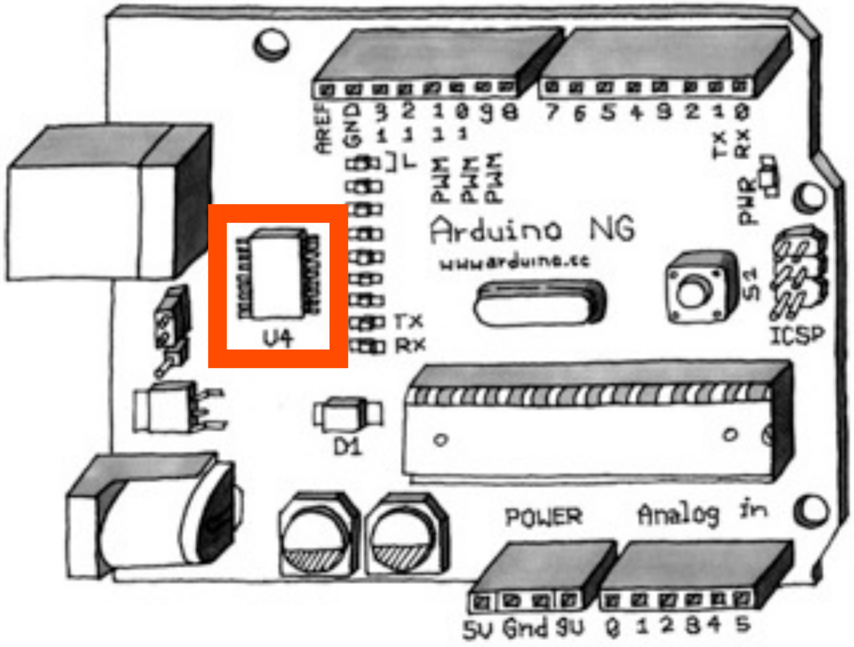
```
enter color command (e.g. 'r43') :
setting color r to 10
setting color g to 50
```



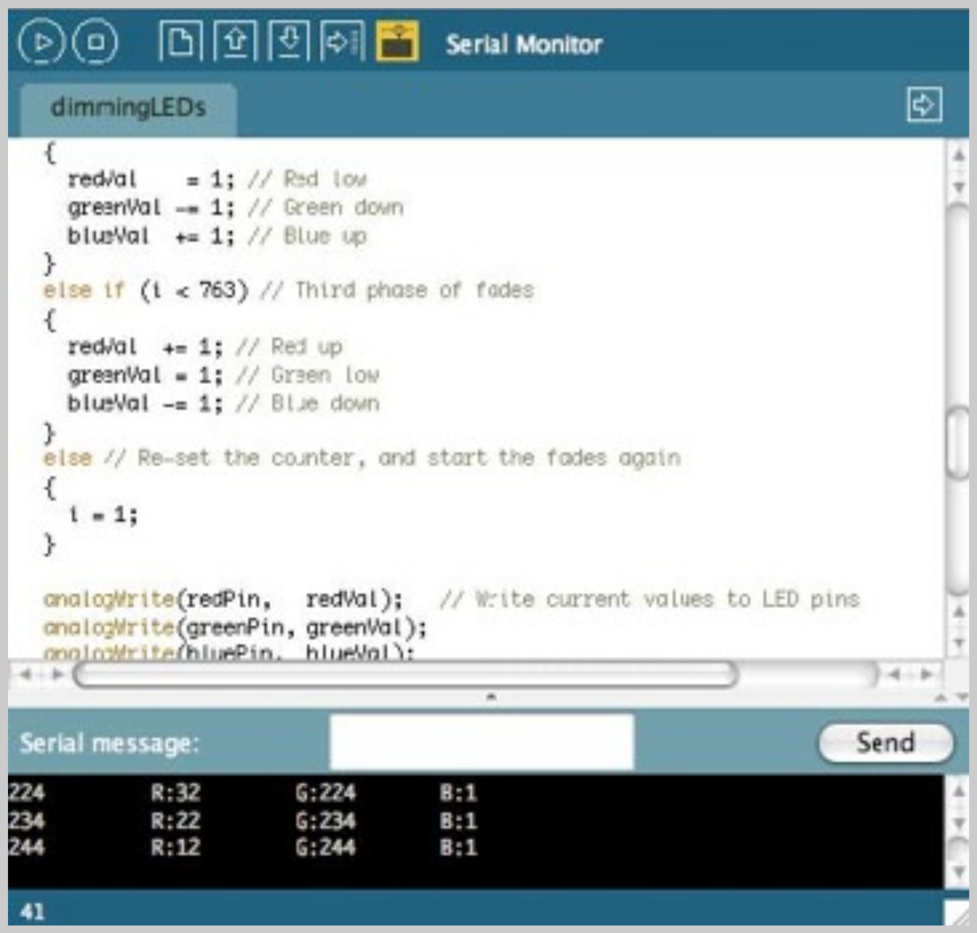
Tell it to mix 50 red, 100 green, and 20 blue... Etc.

Arduino Board

- USB to serial



Serial Monitor



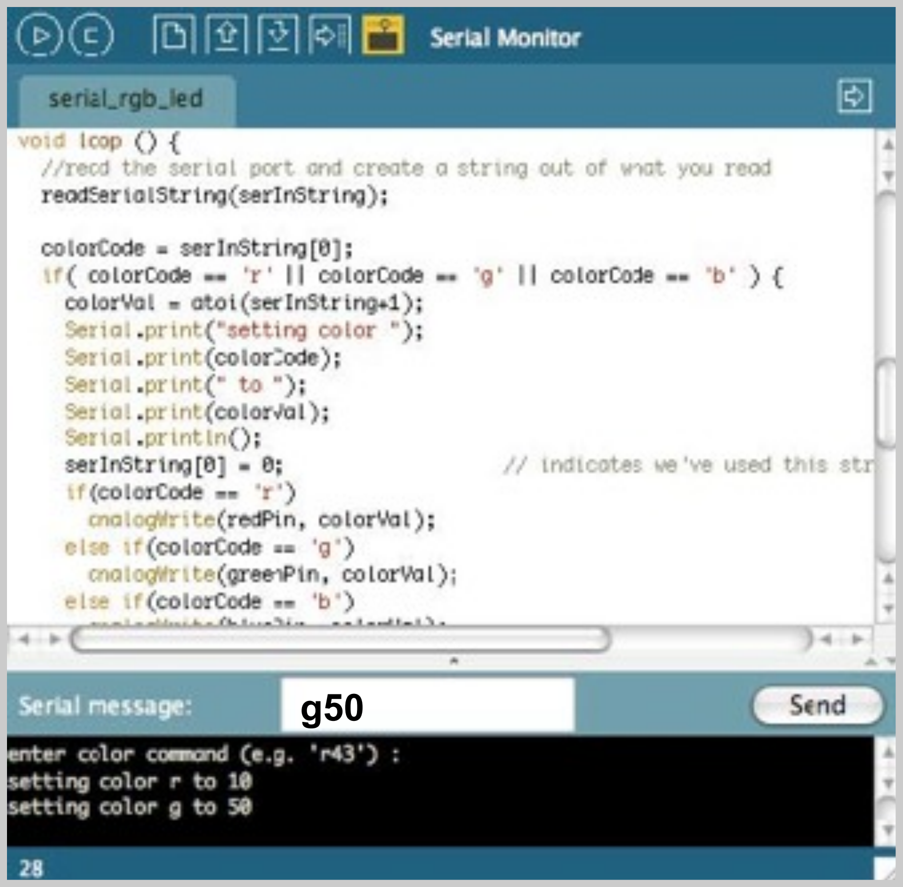
```
{
  redVal     = 1; // Red low
  greenVal -= 1; // Green down
  blueVal  += 1; // Blue up
}
else if (i < 763) // Third phase of fades
{
  redVal  += 1; // Red up
  greenVal = 1; // Green low
  blueVal -= 1; // Blue down
}
else // Re-set the counter, and start the fades again
{
  i = 1;
}

analogWrite(redPin,   redVal);   // Write current values to LED pins
analogWrite(greenPin, greenVal);
analogWrite(bluePin,  blueVal);
```

```
224         R:32        G:224       B:1
234         R:22        G:234       B:1
244         R:12        G:244       B:1
```

Serial Communication

Serial.begin()	e.g., Serial.begin(9600)
Serial.print()	e.g., Serial.print(colorVal)
Serial.read()	

Serial RGB



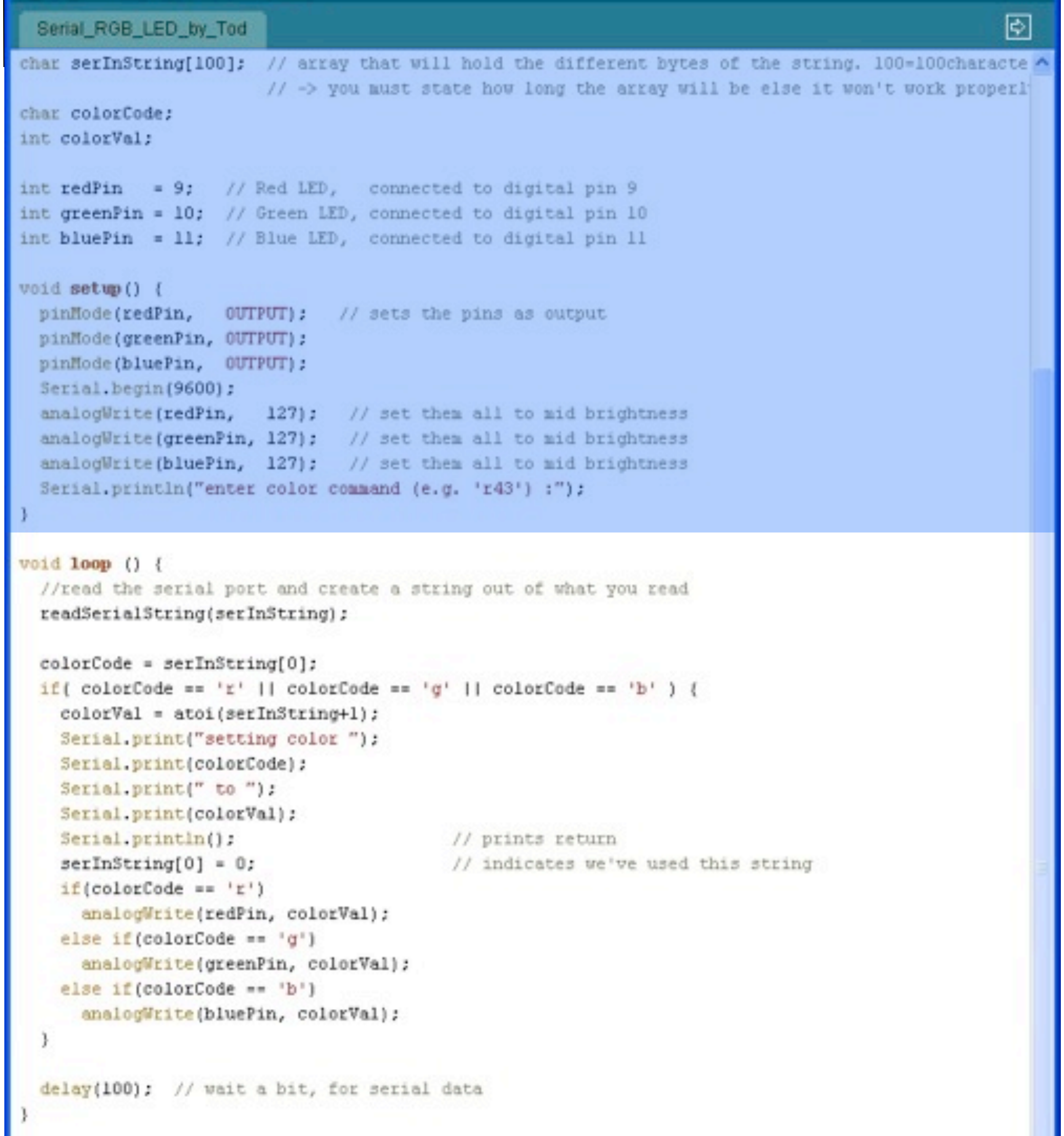
```
void loop () {
  //read the serial port and create a string out of what you read
  readSerialString(serInString);

  colorCode = serInString[0];
  if( colorCode == 'r' || colorCode == 'g' || colorCode == 'b' ) {
    colorVal = atoi(serInString+1);
    Serial.print("setting color ");
    Serial.print(colorCode);
    Serial.print(" to ");
    Serial.print(colorVal);
    Serial.println();
    serInString[0] = 0;                   // indicates we've used this str
    if(colorCode == 'r')
      analogWrite(redPin, colorVal);
    else if(colorCode == 'g')
      analogWrite(greenPin, colorVal);
    else if(colorCode == 'b')
```

Serial message: **g50** Send

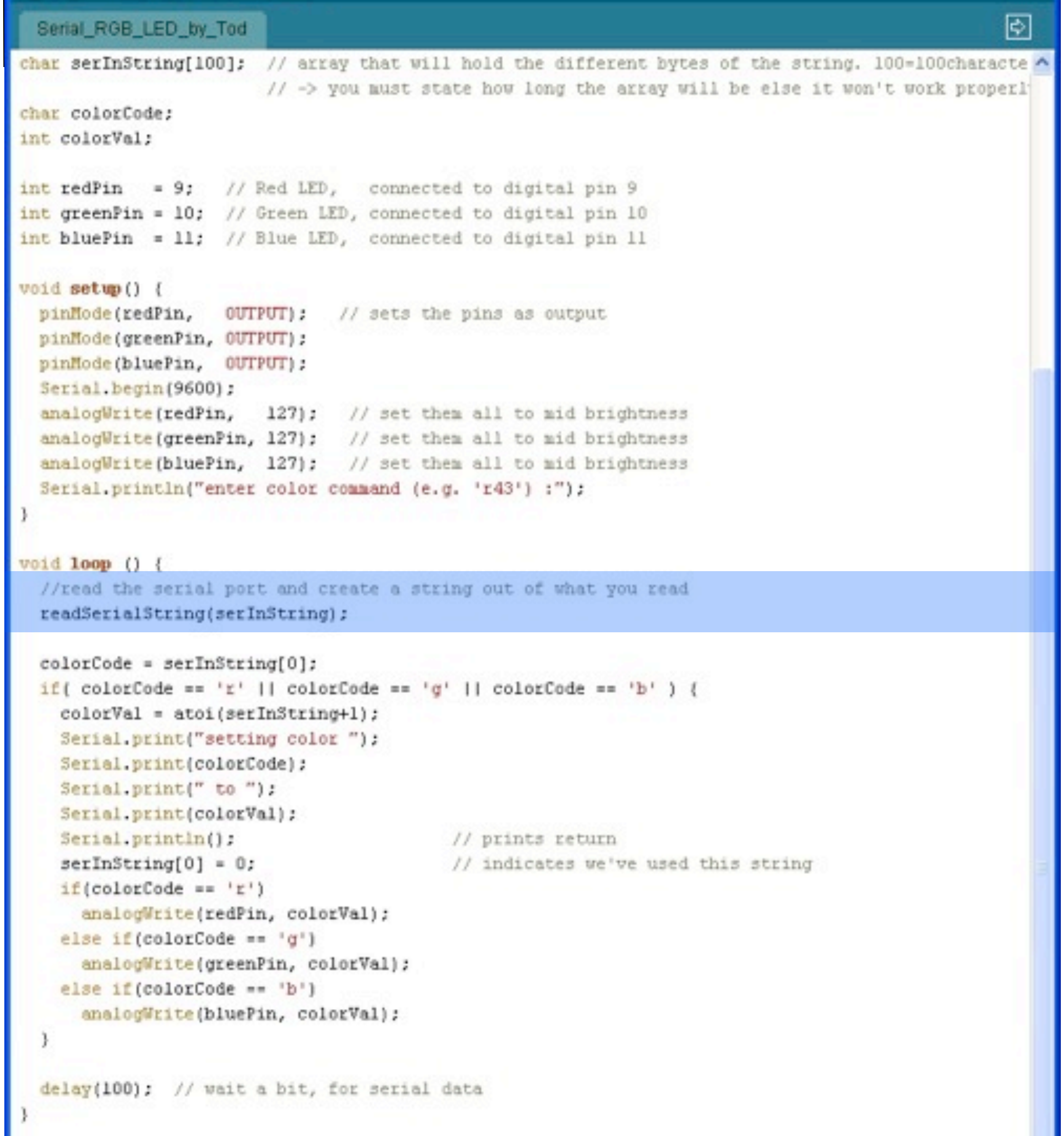
```
enter color command (e.g. 'r43') :
setting color r to 10
setting color g to 50
```

28

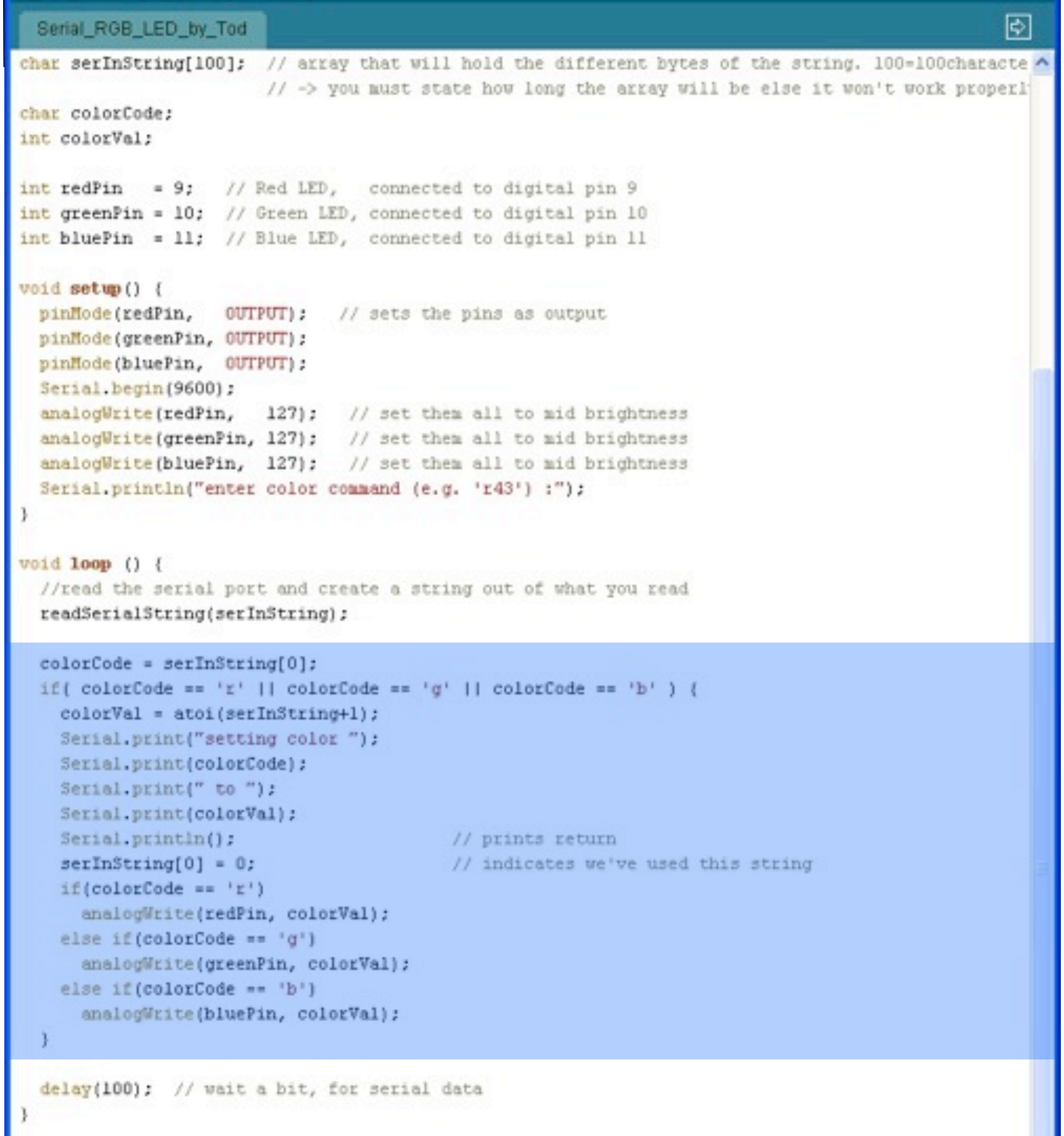
Tell it to mix 50 red, 100 green, and 20 blue... etc.



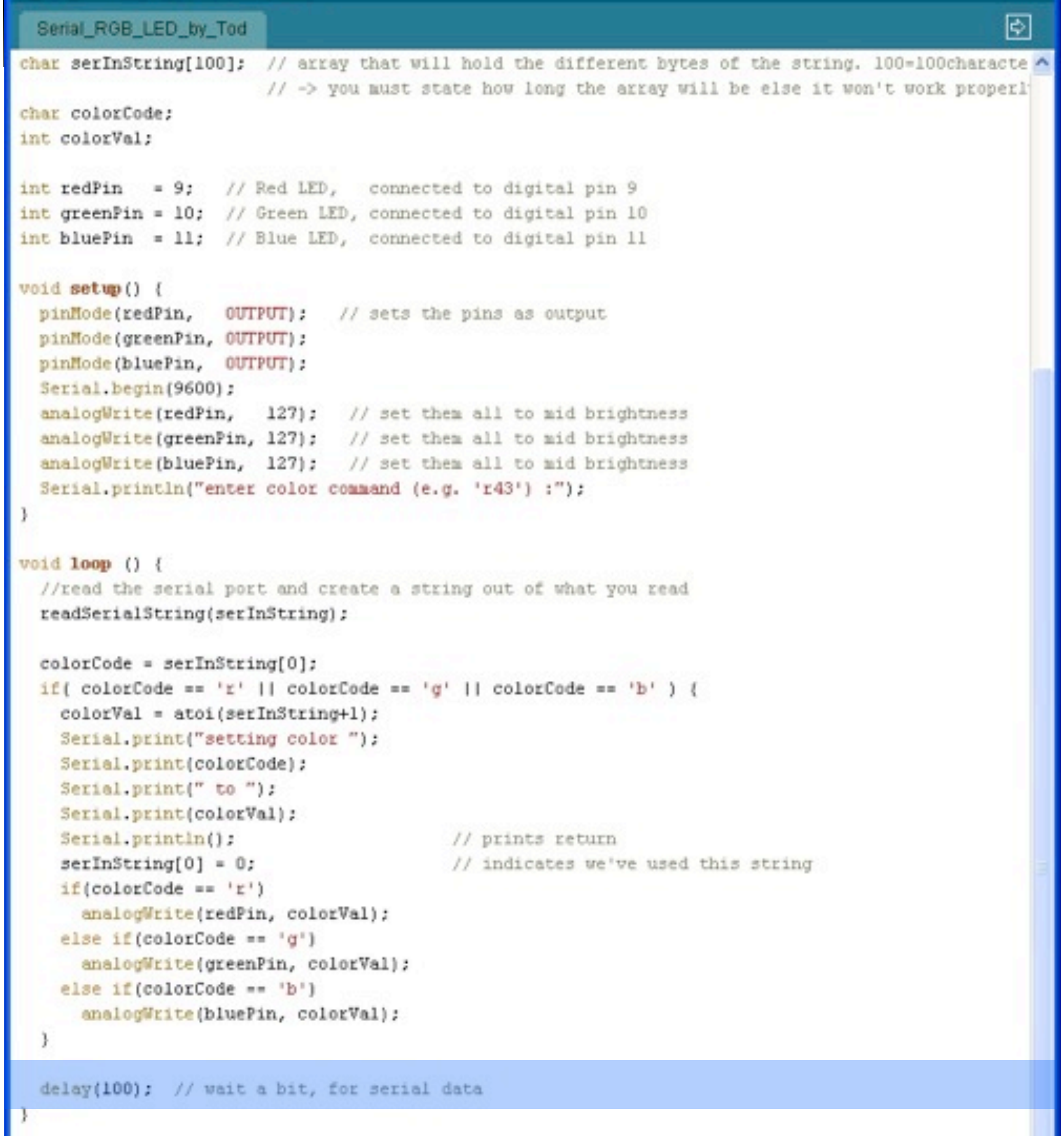
```
char serInString[100];  // array that will hold the different bytes of the string. 100=100characte
                        // -> you must state how long the array will be else it won't work properl
char colorCode;
int colorVal;

int redPin   = 9;   // Red LED,   connected to digital pin 9
int greenPin = 10;  // Green LED, connected to digital pin 10
int bluePin  = 11;  // Blue LED,  connected to digital pin 11

void setup() {
  pinMode(redPin,   OUTPUT);   // sets the pins as output
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin,  OUTPUT);
  Serial.begin(9600);
  analogWrite(redPin,   127);   // set them all to mid brightness
  analogWrite(greenPin, 127);   // set them all to mid brightness
  analogWrite(bluePin,  127);   // set them all to mid brightness
  Serial.println("enter color command (e.g. 'r43') :");
}

void loop () {
  //read the serial port and create a string out of what you read
  readSerialString(serInString);

  colorCode = serInString[0];
  if( colorCode == 'r' || colorCode == 'g' || colorCode == 'b' ) {
    colorVal = atoi(serInString+1);
    Serial.print("setting color ");
    Serial.print(colorCode);
    Serial.print(" to ");
    Serial.print(colorVal);
    Serial.println();                     // prints return
    serInString[0] = 0;                   // indicates we've used this string
    if(colorCode == 'r')
      analogWrite(redPin, colorVal);
    else if(colorCode == 'g')
      analogWrite(greenPin, colorVal);
    else if(colorCode == 'b')
      analogWrite(bluePin, colorVal);
  }

  delay(100);  // wait a bit, for serial data
}
```



```
char serInString[100];  // array that will hold the different bytes of the string. 100=100characte
                        // -> you must state how long the array will be else it won't work properl
char colorCode;
int colorVal;

int redPin   = 9;   // Red LED,   connected to digital pin 9
int greenPin = 10;  // Green LED, connected to digital pin 10
int bluePin  = 11;  // Blue LED,  connected to digital pin 11

void setup() {
  pinMode(redPin,   OUTPUT);   // sets the pins as output
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin,  OUTPUT);
  Serial.begin(9600);
  analogWrite(redPin,   127);   // set them all to mid brightness
  analogWrite(greenPin, 127);   // set them all to mid brightness
  analogWrite(bluePin,  127);   // set them all to mid brightness
  Serial.println("enter color command (e.g. 'r43') :");
}

void loop () {
  //read the serial port and create a string out of what you read
  readSerialString(serInString);

  colorCode = serInString[0];
  if( colorCode == 'r' || colorCode == 'g' || colorCode == 'b' ) {
    colorVal = atoi(serInString+1);
    Serial.print("setting color ");
    Serial.print(colorCode);
    Serial.print(" to ");
    Serial.print(colorVal);
    Serial.println();                   // prints return
    serInString[0] = 0;                 // indicates we've used this string
    if(colorCode == 'r')
      analogWrite(redPin, colorVal);
    else if(colorCode == 'g')
      analogWrite(greenPin, colorVal);
    else if(colorCode == 'b')
      analogWrite(bluePin, colorVal);
  }

  delay(100);  // wait a bit, for serial data
}
```



```
char serInString[100];  // array that will hold the different bytes of the string. 100=100characte
                        // -> you must state how long the array will be else it won't work properl
char colorCode;
int colorVal;

int redPin   = 9;   // Red LED,   connected to digital pin 9
int greenPin = 10;  // Green LED, connected to digital pin 10
int bluePin  = 11;  // Blue LED,  connected to digital pin 11

void setup() {
  pinMode(redPin,   OUTPUT);   // sets the pins as output
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin,  OUTPUT);
  Serial.begin(9600);
  analogWrite(redPin,   127);   // set them all to mid brightness
  analogWrite(greenPin, 127);   // set them all to mid brightness
  analogWrite(bluePin,  127);   // set them all to mid brightness
  Serial.println("enter color command (e.g. 'r43') :");
}

void loop () {
  //read the serial port and create a string out of what you read
  readSerialString(serInString);

  colorCode = serInString[0];
  if( colorCode == 'r' || colorCode == 'g' || colorCode == 'b' ) {
    colorVal = atoi(serInString+1);
    Serial.print("setting color ");
    Serial.print(colorCode);
    Serial.print(" to ");
    Serial.print(colorVal);
    Serial.println();                        // prints return
    serInString[0] = 0;                      // indicates we've used this string
    if(colorCode == 'r')
      analogWrite(redPin, colorVal);
    else if(colorCode == 'g')
      analogWrite(greenPin, colorVal);
    else if(colorCode == 'b')
      analogWrite(bluePin, colorVal);
  }

  delay(100);  // wait a bit, for serial data
}
```



```
char serInString[100];  // array that will hold the different bytes of the string. 100=100characte
                        // -> you must state how long the array will be else it won't work properl
char colorCode;
int colorVal;

int redPin   = 9;   // Red LED,   connected to digital pin 9
int greenPin = 10;  // Green LED, connected to digital pin 10
int bluePin  = 11;  // Blue LED,  connected to digital pin 11

void setup() {
  pinMode(redPin,   OUTPUT);   // sets the pins as output
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin,  OUTPUT);
  Serial.begin(9600);
  analogWrite(redPin,   127);   // set them all to mid brightness
  analogWrite(greenPin, 127);   // set them all to mid brightness
  analogWrite(bluePin,  127);   // set them all to mid brightness
  Serial.println("enter color command (e.g. 'r43') :");
}

void loop () {
  //read the serial port and create a string out of what you read
  readSerialString(serInString);

  colorCode = serInString[0];
  if( colorCode == 'r' || colorCode == 'g' || colorCode == 'b' ) {
    colorVal = atoi(serInString+1);
    Serial.print("setting color ");
    Serial.print(colorCode);
    Serial.print(" to ");
    Serial.print(colorVal);
    Serial.println();                     // prints return
    serInString[0] = 0;                   // indicates we've used this string
    if(colorCode == 'r')
      analogWrite(redPin, colorVal);
    else if(colorCode == 'g')
      analogWrite(greenPin, colorVal);
    else if(colorCode == 'b')
      analogWrite(bluePin, colorVal);
  }

  delay(100);  // wait a bit, for serial data
}
```

In Class Exercise

1. LED Blink
2. LED Fade
3. Circuit with 3 LEDs
4. RGB LED Fade
5. Serial RGB LED

Homework (due next Tuesday, Feb 8)

Part I: Design a good diffuser for your RGB LEDs

e.g., ping pong ball, styrofoam, mylar, cottons, icecream cone?

Part II: Modify the Serial RGB code for new keyboard input:

- **Basic**: Control the RGB values with multiple key presses (e.g., instead of typing “r127” to set the Red LED to 50% brightness, count how many times the characters “r” “g” “b” were pressed. E.g., press “r” 5 times to get it to 50%, 8 times for 80%, 10 times for 100%, 11 times to go back to 0%, etc.)
- **Advanced**: Find new ways of controlling the colors of LEDs using the keyboard

Post both parts on the course website (photo, descriptions, code)

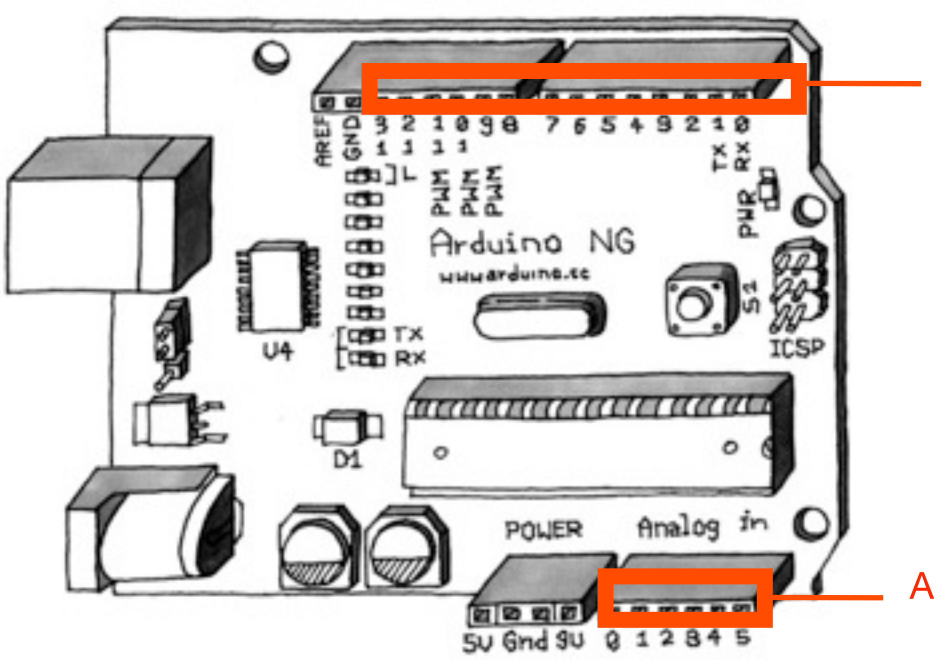
Supplement Readings

More on PWM and LED fading: Chapter 6 (p. 112-114) of O’Sullivan and Igoe

More on microcontroller in general: Chapter 4 (p.49-63)

Next Wednesday: Analog Input

Two states (binary signal) vs. multiple states (continuous signal)



Digital Input and Output

Analog Input (next week!)