Reading Analog Data

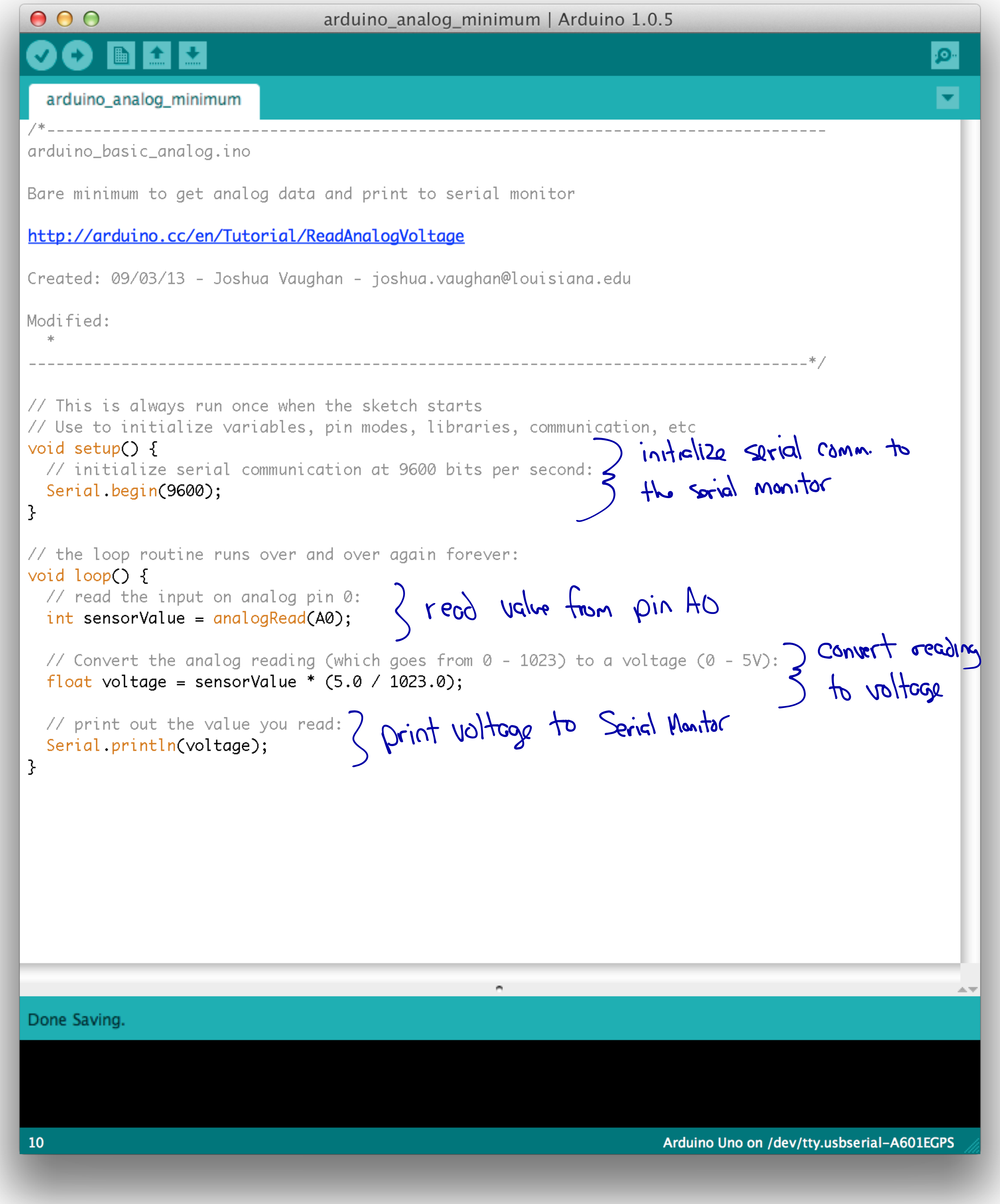
$$\underbrace{\texttt{int}}_{\text{var type should be int}}\ \underbrace{\texttt{variable\_name}}_{\text{variable name to store value in}} = \underbrace{\texttt{analogRead}}_{\text{function to read data}}\underbrace{\texttt{(analog\_pin)}}_{\text{pin to read (A0-A5)}};$$

This will be between 0-1023

$$\left.\begin{array}{l}\text{analog } 5V = 1023\\ \text{analog } 0V = 0\end{array}\right\}\ \text{create linear function from reading to analog voltage } (y=mx+b)$$

$$\text{Slope} = \frac{5-0}{1023-0} = \frac{5}{1023} \qquad \text{y-intercept} = 0$$

$$\text{so}\quad \text{voltage} = \frac{5}{1023}\left(\text{sensor reading}\right)$$



```
/*------------------------------------------------------------------------------------
arduino_basic_analog.ino

Bare minimum to get analog data and print to serial monitor

http://arduino.cc/en/Tutorial/ReadAnalogVoltage

Created: 09/03/13 - Joshua Vaughan - joshua.vaughan@louisiana.edu

Modified:
  *
-------------------------------------------------------------------------------------*/

// This is always run once when the sketch starts
// Use to initialize variables, pin modes, libraries, communication, etc
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);

  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);

  // print out the value you read:
  Serial.println(voltage);
}
```

} initialize serial comm. to the serial monitor

} read value from pin A0

} convert reading to voltage

} print voltage to Serial Monitor

Done Saving.

10

Arduino Uno on /dev/tty.usbserial-A601EGPS

Writing Analog Data (actually PWM)

```
pinMode(digital_pin,INPUT);    or

pinMode(digital_pin,OUTPUT);
```

variable for pin #

Defines digital_pin as an INPUT

Defines digital_pin as an OUTPUT

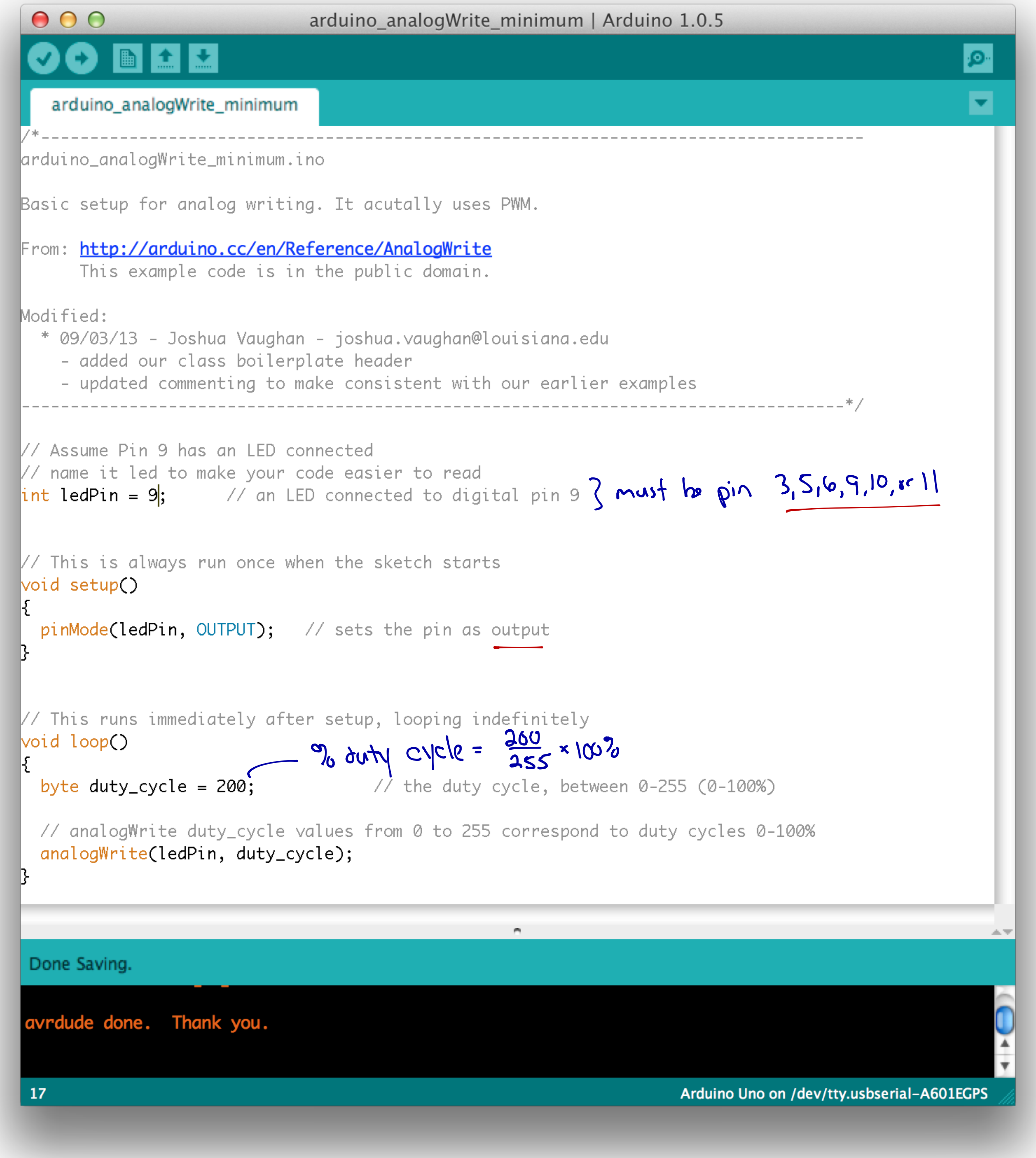
```
analogWrite(digital_pin, duty_cycle);
```

function name

pin to control must be OUTPUT

duty cycle → 0-255 = 0-100%

☆ Only available on PWM pins (3,5,6,9,10,11 on the RedBoard) ☆



arduino_analogWrite_minimum | Arduino 1.0.5

arduino_analogWrite_minimum

```
/*------------------------------------------------------------------------------------
arduino_analogWrite_minimum.ino

Basic setup for analog writing. It acutally uses PWM.

From: http://arduino.cc/en/Reference/AnalogWrite
      This example code is in the public domain.

Modified:
  * 09/03/13 - Joshua Vaughan - joshua.vaughan@louisiana.edu
    - added our class boilerplate header
    - updated commenting to make consistent with our earlier examples
----------------------------------------------------------------------------------*/

// Assume Pin 9 has an LED connected
// name it led to make your code easier to read
int ledPin = 9;      // an LED connected to digital pin 9


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


// This runs immediately after setup, looping indefinitely
void loop()
{
  byte duty_cycle = 200;            // the duty cycle, between 0-255 (0-100%)

  // analogWrite duty_cycle values from 0 to 255 correspond to duty cycles 0-100%
  analogWrite(ledPin, duty_cycle);
}
```

} must be pin 3,5,6,9,10,or 11

% duty cycle = $\frac{200}{255} \times 100\%$

Done Saving.

```
avrdude done.  Thank you.
```

17

Arduino Uno on /dev/tty.usbserial-A601EGPS

Reading Digital Data

```
pinMode(digital_pin,INPUT);    or

pinMode(digital_pin,OUTPUT);
```

Defines digital_pin as an INPUT

variable for pin #

Defines digital_pin as an OUTPUT

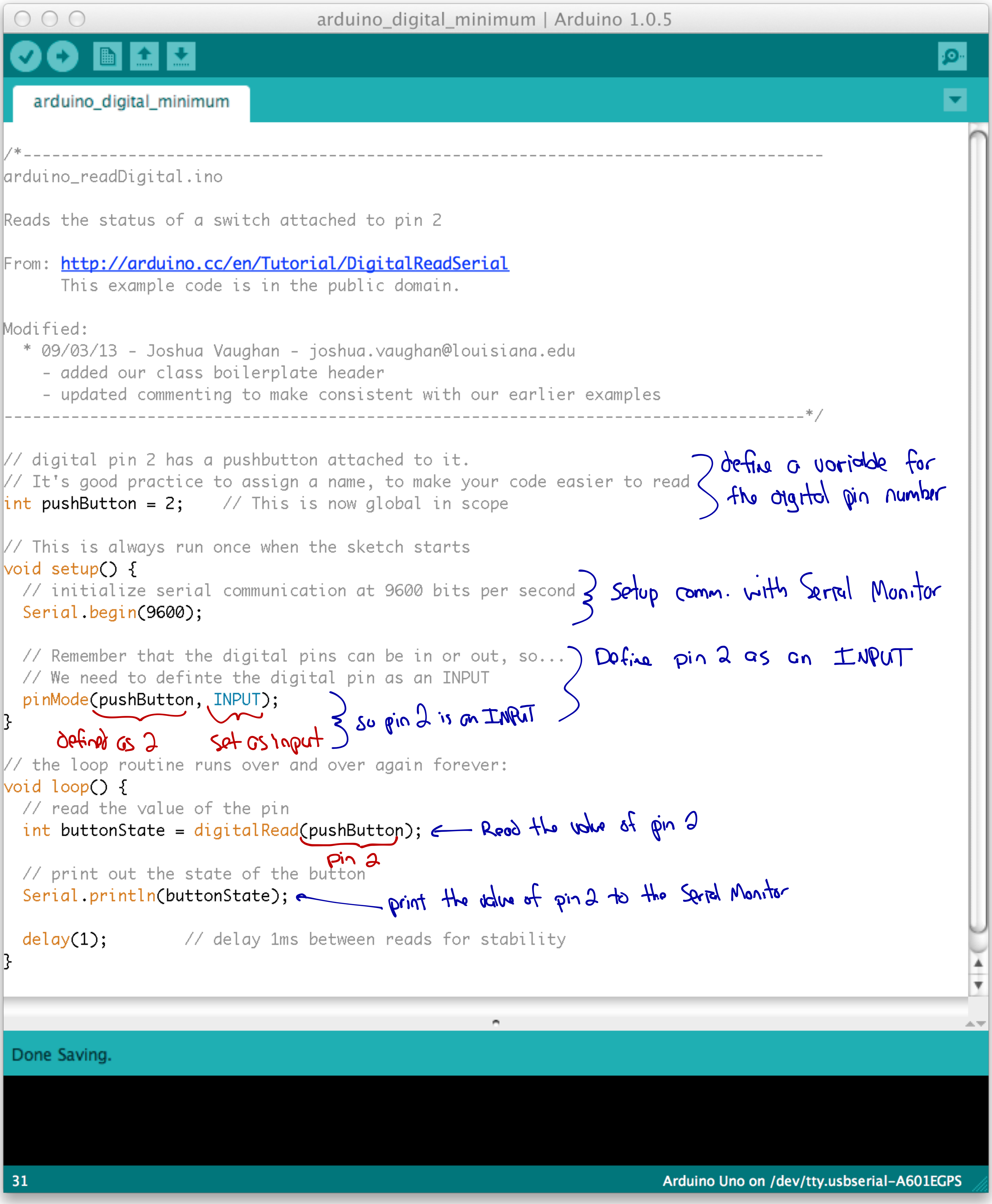
```
int variable_name = digitalRead(digital_pin);
```

name to store value in

function to read digital

pin number or variable defined as pin number

☆ This pin must previously have been defined as an INPUT ☆


arduino_digital_minimum | Arduino 1.0.5
arduino_digital_minimum
Done Saving.
31
Arduino Uno on /dev/tty.usbserial-A601EGPS

```
/*---------------------------------------------------------------------------------
arduino_readDigital.ino

Reads the status of a switch attached to pin 2

From: http://arduino.cc/en/Tutorial/DigitalReadSerial
      This example code is in the public domain.

Modified:
  * 09/03/13 - Joshua Vaughan - joshua.vaughan@louisiana.edu
    - added our class boilerplate header
    - updated commenting to make consistent with our earlier examples
---------------------------------------------------------------------------------*/

// digital pin 2 has a pushbutton attached to it.
// It's good practice to assign a name, to make your code easier to read
int pushButton = 2;    // This is now global in scope

// This is always run once when the sketch starts
void setup() {
  // initialize serial communication at 9600 bits per second
  Serial.begin(9600);

  // Remember that the digital pins can be in or out, so...
  // We need to definte the digital pin as an INPUT
  pinMode(pushButton, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the value of the pin
  int buttonState = digitalRead(pushButton);

  // print out the state of the button
  Serial.println(buttonState);

  delay(1);        // delay 1ms between reads for stability
}
```

define a variable for the digital pin number

Setup comm. with Serial Monitor

Define pin 2 as an INPUT

so pin 2 is an INPUT

defined as 2

set as input

Read the value of pin 2

pin 2

print the value of pin 2 to the Serial Monitor

Writing Digital Data - Just On/Off

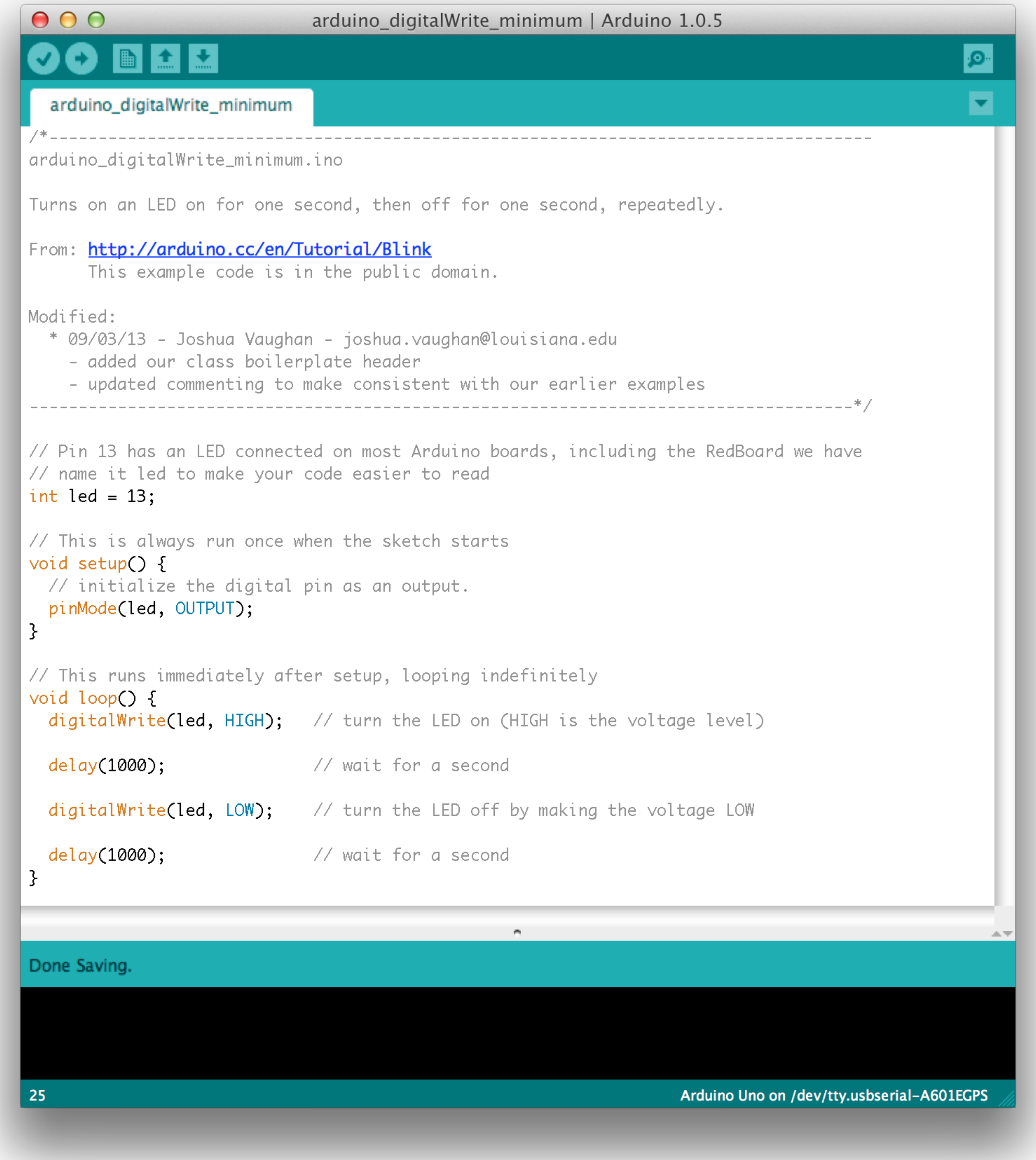
```
digitalWrite(digital_pin, HIGH); or
```

HIGH is on (5V - high voltage)

Pin must be OUTPUT

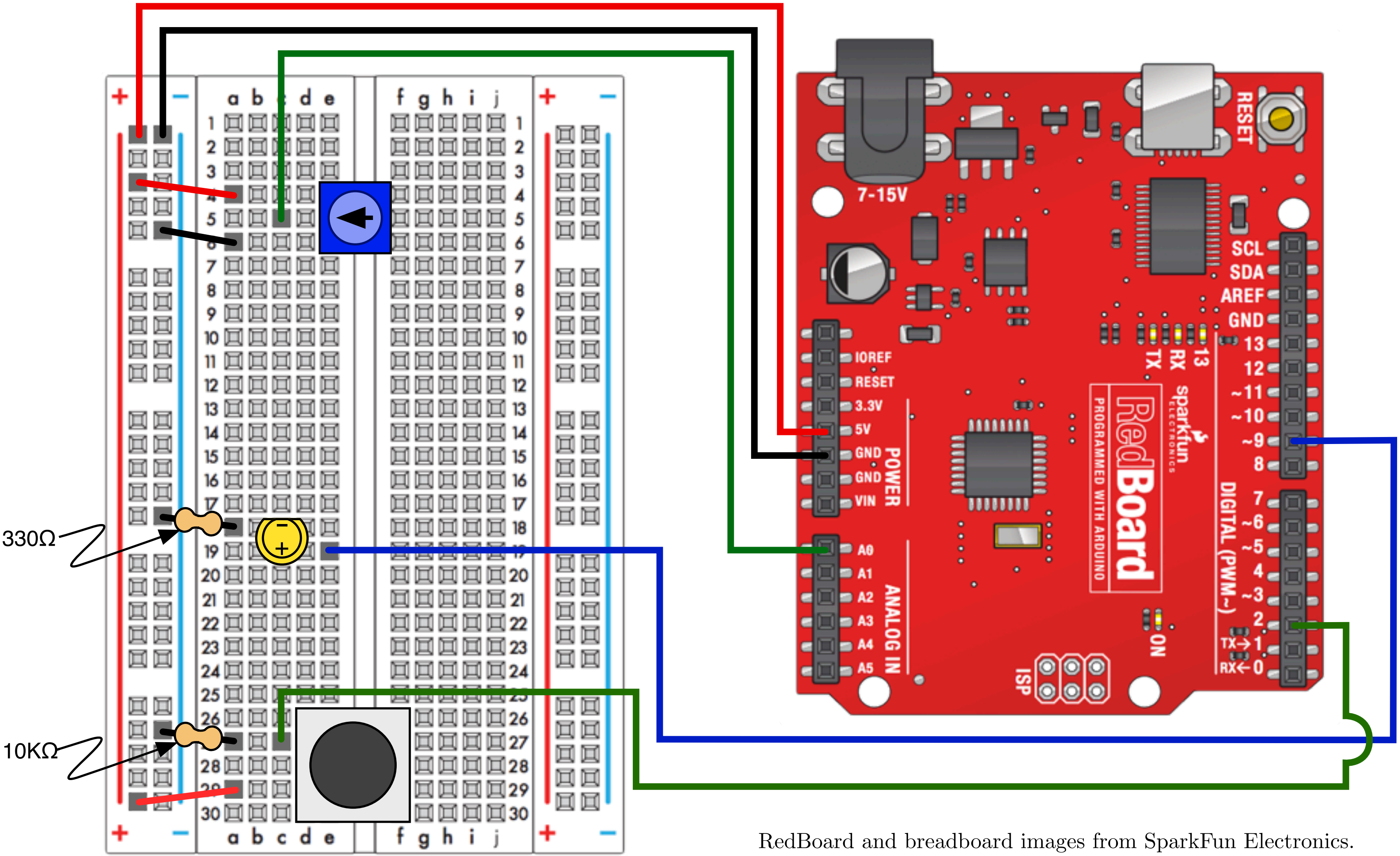
```
digitalWrite(digital_pin, LOW);
```

LOW is off (0V - low voltage)


arduino_digitalWrite_minimum | Arduino 1.0.5
arduino_digitalWrite_minimum

```
/*--------------------------------------------------------------------------------------
arduino_digitalWrite_minimum.ino

Turns on an LED on for one second, then off for one second, repeatedly.

From: http://arduino.cc/en/Tutorial/Blink
      This example code is in the public domain.

Modified:
  * 09/03/13 - Joshua Vaughan - joshua.vaughan@louisiana.edu
    - added our class boilerplate header
    - updated commenting to make consistent with our earlier examples
-------------------------------------------------------------------------------------*/

// Pin 13 has an LED connected on most Arduino boards, including the RedBoard we have
// name it led to make your code easier to read
int led = 13;

// This is always run once when the sketch starts
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// This runs immediately after setup, looping indefinitely
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)

  delay(1000);               // wait for a second

  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW

  delay(1000);               // wait for a second
}
```

Done Saving.



RedBoard and breadboard images from SparkFun Electronics.