

# Reading Analog Data

```
int variable_name = analogRead(analog_pin);
```

Var type  
should be  
int

variable name to store  
value in

This will be between 0-1023

function to read data

pin to read (A0-A5)

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

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** arduino\_analog\_minimum | Arduino 1.0.5
- Toolbar:** Standard Arduino toolbar with icons for file operations.
- Code Editor:** The code for "arduino\_analog\_minimum.ino" is displayed:

```
/*-----  
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);  
    } read value from pin A0  
    // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):  
    float voltage = sensorValue * (5.0 / 1023.0);  
    } convert reading to voltage  
    // print out the value you read:  
    Serial.println(voltage);  
}
```
- Serial Monitor:** Shows the message "Done Saving."
- Bottom Status Bar:** Shows "Arduino Uno on /dev/tty.usbserial-A601EGPS".

Handwritten annotations in blue ink are present in the code editor area:

- A brace groups the first two lines of the code: "analog 5V = 1023" and "analog 0V = 0". An arrow points from this brace to the text "create linear function from reading to analog voltage (y=mx+b)".
- A brace groups the "Serial.begin(9600)" call and the "void setup()" block. An arrow points from this brace to the text "initialize serial comm to the serial monitor".
- A brace groups the "analogRead(A0)" call and the "int sensorValue" declaration. An arrow points from this brace to the text "read value from pin A0".
- A brace groups the "Serial.println(voltage)" call and the "float voltage" declaration. An arrow points from this brace to the text "print voltage to Serial Monitor".

## Writing Analog Data (actually PWM)

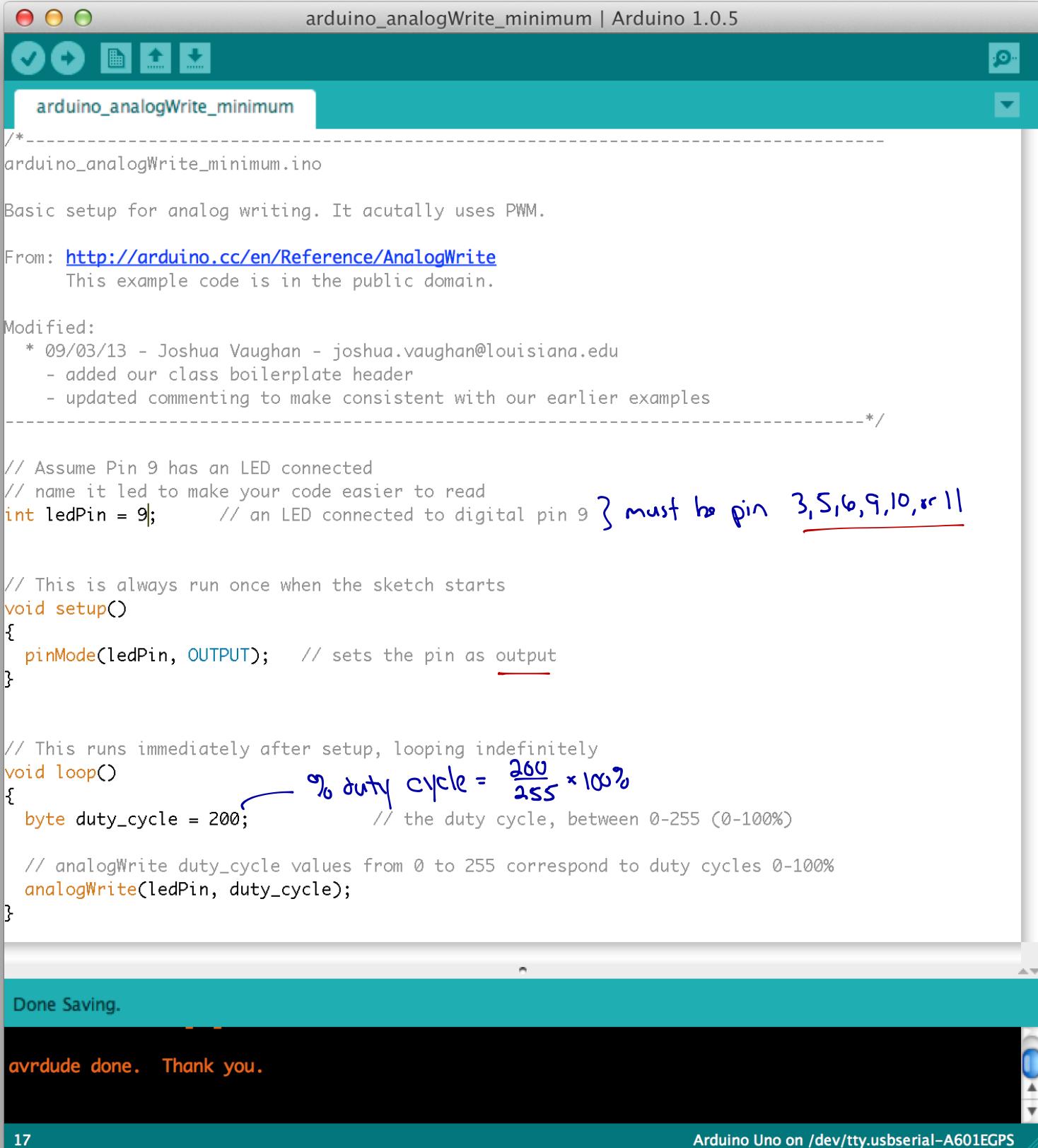
`pinMode(digital_pin, INPUT);` or Defines digital-pin as an INPUT  
`pinMode(digital_pin, OUTPUT);` Defines digital-pin as an OUTPUT

Variable for pin #

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

// 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(ledPin, duty_cycle);
}

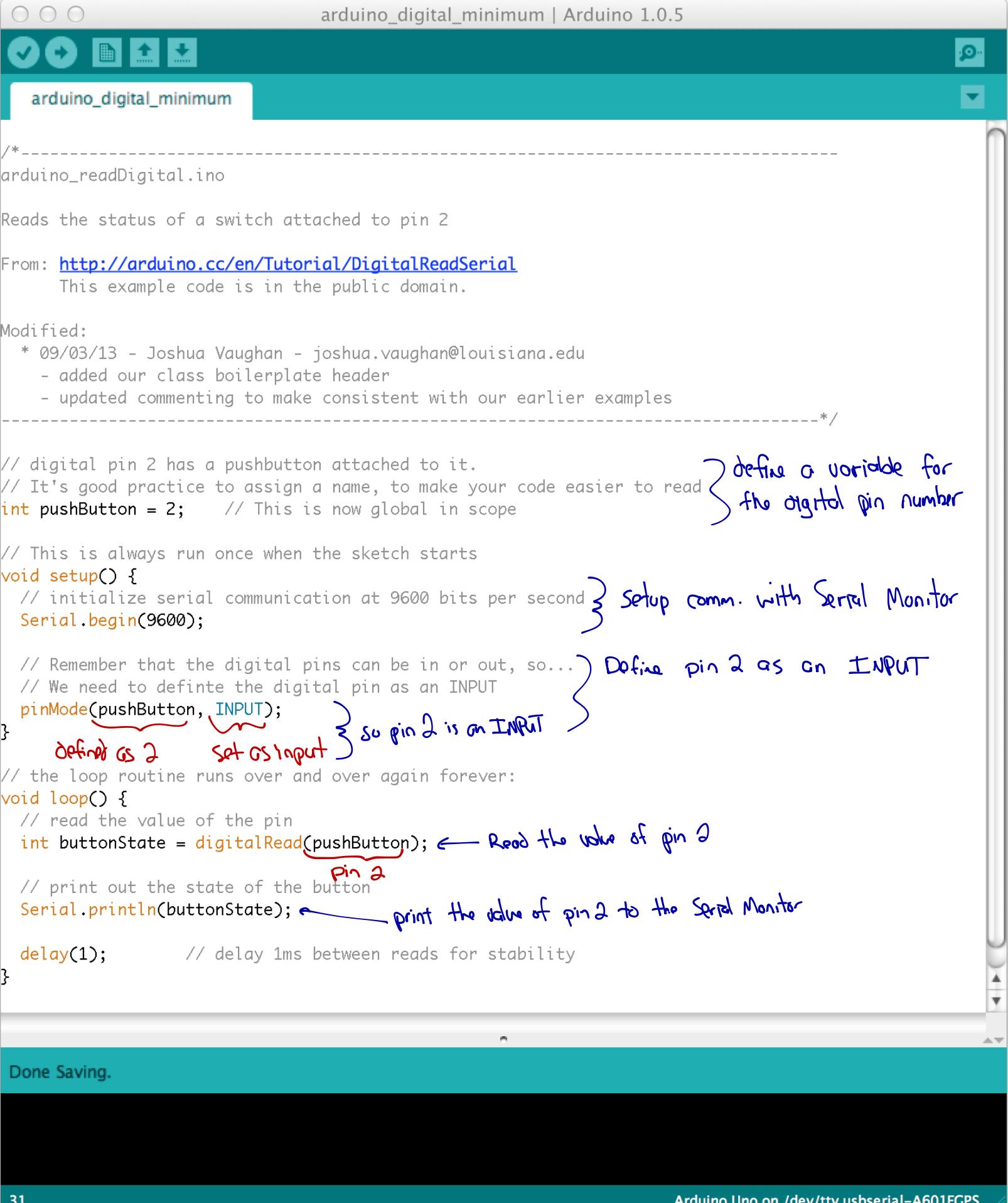
Done Saving.

avrduke done. Thank you.
```

## Reading Digital Data

`pinMode(digital_pin, INPUT);` or Defines digital-pin as an INPUT  
variable for pin #  
`pinMode(digital_pin, OUTPUT);` 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 ★



The screenshot shows the Arduino IDE interface with the title bar "arduino\_digital\_minimum | Arduino 1.0.5". The code editor contains the following sketch:

```
/*
  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 } define a variable for
// the digital pin number

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

  // Remember that the digital pins can be in or out, so...
  // We need to define the digital pin as an INPUT
  pinMode(pushButton, INPUT); } Define pin 2 as an INPUT
} } so pin 2 is an INPUT

// the loop routine runs over and over again forever:
void loop() {
  // read the value of the pin
  int buttonState = digitalRead(pushButton); } Read the value of pin 2
  pin 2
  // print out the state of the button
  Serial.println(buttonState); } print the value of pin 2 to the Serial Monitor
  delay(1); // delay 1ms between reads for stability
}
```

Annotations in blue and red are present in the code:

- A brace groups the line `int pushButton = 2;` with the note "define a variable for the digital pin number".
- A brace groups the `pinMode(pushButton, INPUT);` call with the notes "Define pin 2 as an INPUT" and "so pin 2 is an INPUT". A red annotation "Define as 2" is next to "pushButton" and "set as input" is next to "INPUT".
- A brace groups the `digitalRead(pushButton);` call with the note "Read the value of pin 2". A red annotation "pin 2" is next to "pushButton".
- A brace groups the `Serial.println(buttonState);` call with the note "print the value of pin 2 to the Serial Monitor".

The status bar at the bottom of the IDE shows "Done Saving." and "Arduino Uno on /dev/tty.usbserial-A601EGPS".

## 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)

The screenshot shows the Arduino IDE interface with the title bar "arduino\_digitalWrite\_minimum | Arduino 1.0.5". The code editor contains the following code:

```
/*-----  
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.
```

The status bar at the bottom right shows "Arduino Uno on /dev/tty.usbserial-A601EGPS".

