

```
/*  
Fade
```

This example shows how to fade an LED on pin 9 using the `analogWrite()` function.

The `analogWrite()` function uses PWM, so if you want to change the pin you're using, be sure to use another PWM capable pin. On most Arduino, the PWM pins are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.

This example code is in the public domain.

```
https://www.arduino.cc/en/Tutorial/BuiltInExamples/Fade  
*/
```

```
int led = 9;           // the PWM pin the LED is attached to  
int brightness = 0;    // how bright the LED is  
int fadeAmount = 5;    // how many points to fade the LED by  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // declare pin 9 to be an output:  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  // set the brightness of pin 9:  
  analogWrite(led, brightness);  
  
  // change the brightness for next time through the loop:  
  brightness = brightness + fadeAmount;  
  
  // reverse the direction of the fading at the ends of the fade:  
  if (brightness <= 0 || brightness >= 255) {  
    fadeAmount = -fadeAmount;  
  }  
  // wait for 30 milliseconds to see the dimming effect  
  delay(30);  
}
```

```
/*
```

```
Fading
```

This example shows how to fade an LED using the `analogWrite()` function.

The circuit:

- LED attached from digital pin 9 to ground through 220 ohm resistor.

created 1 Nov 2008

by David A. Mellis

modified 30 Aug 2011

by Tom Igoe

This example code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Fading>

```
*/
```

```
int ledPin = 9; // LED connected to digital pin 9

void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0; fadeValue <= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for (int fadeValue = 255; fadeValue >= 0; fadeValue -= 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```

```

/*
Switch statement
Demonstrates the use of a switch statement. The switch statement allows you
to choose from among a set of discrete values of a variable. It's like a
series of if statements.
To see this sketch in action, put the board and sensor in a well-lit room,
open the Serial Monitor, and move your hand gradually down over the sensor.
The circuit:
- photoresistor from analog in 0 to +5V
- 10K resistor from analog in 0 to ground
created 1 Jul 2009
modified 9 Apr 2012
by Tom Igoe
This example code is in the public domain.
https://www.arduino.cc/en/Tutorial/BuiltInExamples/SwitchCase
*/

// these constants won't change. They are the lowest and highest readings you
// get from your sensor:
const int sensorMin = 0;    // sensor minimum, discovered through experiment
const int sensorMax = 600;  // sensor maximum, discovered through experiment

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

void loop() {
  // read the sensor:
  int sensorReading = analogRead(A0);
  // map the sensor range to a range of four options:
  int range = map(sensorReading, sensorMin, sensorMax, 0, 3);

  // do something different depending on the range value:
  switch (range) {
    case 0: // your hand is on the sensor
      Serial.println("dark");
      break;
    case 1: // your hand is close to the sensor
      Serial.println("dim");
      break;
    case 2: // your hand is a few inches from the sensor
      Serial.println("medium");
      break;
    case 3: // your hand is nowhere near the sensor
      Serial.println("bright");
      break;
  }
  delay(1); // delay in between reads for stability
}

```

Servo Sweep 180*

```
#include <Servo.h>
```

```
Servo myservo; // create servo object to control a servo  
// twelve servo objects can be created on most boards
```

```
int pos = 0; // variable to store the servo position
```

```
void setup() {  
  myservo.attach(9); // attaches the servo on pin 9 to the servo object  
}
```

```
void loop() {  
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees  
    // in steps of 1 degree  
    myservo.write(pos); // tell servo to go to position in variable 'pos'  
    delay(15); // waits 15ms for the servo to reach the position  
  }  
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees  
    myservo.write(pos); // tell servo to go to position in variable 'pos'  
    delay(15); // waits 15ms for the servo to reach the position  
  }  
}
```

```
/*
 * HC-SR04 example sketch
 *
 * https://create.arduino.cc/projecthub/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-036380
 *
 * by Isaac100
 */
```

```
const int trigPin = 9;
const int echoPin = 10;
```

```
float duration, distance;
```

```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration*.0343)/2;
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);
}
```

```
/*
  LiquidCrystal Library - Hello World
```

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD and shows the time.

The circuit:

- * LCD RS pin to digital pin 12
- * LCD Enable pin to digital pin 11
- * LCD D4 pin to digital pin 5

- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 2
- * LCD R/W pin to ground
- * LCD VSS pin to ground
- * LCD VCC pin to 5V
- * 10K resistor:
- * ends to +5V and ground
- * wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008
 by David A. Mellis
 library modified 5 Jul 2009
 by Limor Fried (<http://www.ladyada.net>)
 example added 9 Jul 2009
 by Tom Igoe
 modified 22 Nov 2010
 by Tom Igoe
 modified 7 Nov 2016
 by Arturo Guadalupi

This example code is in the public domain.

<https://docs.arduino.cc/learn/electronics/lcd-displays>

```
*/

// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```