

```
/*SPINNING A MOTOR
```

Use a transistor to spin a motor at different speeds. We'll also show you how to input data from the serial port (see the serialSpeed() function below). Motors are the basis for thousands of things in our daily lives, and the Arduino can control them. Here we'll use pulse-width modulation (PWM) to vary the speed of a motor.

The Arduino pins are strong enough to light small LEDs (up to 40 milliAmps), but they're not strong enough to run motors and other power-hungry parts. (This motor needs 50-100mA). Because the motor needs more current than an Arduino pin can provide, we'll use a transistor to do the heavy lifting. A transistor is a solid-state switch. When we give it a small amount of current, it can switch a much larger current.

You can turn a transistor on and off using the digitalWrite() function, but you can also use the analogWrite() function to vary the speed of the motor. The analogWrite() function pulses a pin, varying the width of the pulse from 0% to 100%. We call this technique "PWM", for "Pulse-Width Modulation".

One thing to keep in mind is that when you lower the speed of a motor using PWM, you're also reducing the torque (strength) of the motor. For PWM values below 50 or so, the motor won't have enough torque to start spinning. It will start spinning when you raise the speed a bit.

```
*/
```

```
// We'll be controlling the motor from pin 9.  
// This must be one of the PWM-capable pins.
```

```
const int motorPin = 9;
```

```
void setup()  
{  
  // Set up the motor pin to be an output:  
  
  pinMode(motorPin, OUTPUT);  
  
  // Set up the serial port:  
  
  Serial.begin(9600);  
}
```

```
void loop()  
{  
  // Here we've used comments to disable some of the examples.  
  // To try different things, uncomment one of the following lines  
  // and comment the other ones. See the functions below to learn  
  // what they do and how they work.  
  
  // motorOnThenOff();  
  // motorOnThenOffWithSpeed();  
  // motorAcceleration();  
  serialSpeed();  
}
```

```

}

// This function turns the motor on and off like the blinking LED.
// Try different values to affect the timing.

void motorOnThenOff()
{
  int onTime = 3000; // milliseconds to turn the motor on
  int offTime = 3000; // milliseconds to turn the motor off

  digitalWrite(motorPin, HIGH); // turn the motor on (full speed)
  delay(onTime); // delay for onTime milliseconds
  digitalWrite(motorPin, LOW); // turn the motor off
  delay(offTime); // delay for offTime milliseconds
}

// This function alternates between two speeds.
// Try different values to affect the timing and speed.

void motorOnThenOffWithSpeed()
{
  int Speed1 = 200; // between 0 (stopped) and 255 (full speed)
  int Time1 = 3000; // milliseconds for speed 1

  int Speed2 = 50; // between 0 (stopped) and 255 (full speed)
  int Time2 = 3000; // milliseconds to turn the motor off

  analogWrite(motorPin, Speed1); // turns the motor On
  delay(Time1); // delay for onTime milliseconds
  analogWrite(motorPin, Speed2); // turns the motor Off
  delay(Time2); // delay for offTime milliseconds
}

// This function slowly accelerates the motor to full speed,
// then back down to zero.

void motorAcceleration()
{
  int speed;
  int delayTime = 20; // milliseconds between each speed step

  // accelerate the motor

  for(speed = 0; speed <= 255; speed++)
  {
    analogWrite(motorPin, speed); // set the new speed
    delay(delayTime); // delay between speed steps
  }

  // decelerate the motor

  for(speed = 255; speed >= 0; speed--)
  {
    analogWrite(motorPin, speed); // set the new speed
    delay(delayTime); // delay between speed steps
  }
}

```

```
// This function will let you type a speed into the serial
// monitor window. Open the serial monitor code editor window. Then
// type your desired speed into the small text entry bar at the
// bottom of the window and click "Send" or press return. The motor
// will then operate at that speed. The valid range is 0 to 255.
```

```
void serialSpeed()
{
  int speed;

  Serial.println("Type a speed (0-255) into the box above,");
  Serial.println("then click [send] or press [return]");
  Serial.println(); // Print a blank line

  // In order to type out the above message only once,
  // we'll run the rest of this function in an infinite loop:

  while(true) // "true" is always true, so this will loop forever.
  {
    // First we check to see if incoming data is available:

    while (Serial.available() > 0)
    {
      // If it is, we'll use parseInt() to pull out any numbers:

      speed = Serial.parseInt();

      // Because analogWrite() only works with numbers from
      // 0 to 255, we'll be sure the input is in that range:

      speed = constrain(speed, 0, 255);

      // We'll print out a message to let you know that the
      // number was received:

      Serial.print("Setting speed to ");
      Serial.println(speed);

      // And finally, we'll set the speed of the motor!

      analogWrite(motorPin, speed);
    }
  }
}
```