# Simple GPIO Control using Pi4J.

The following example demonstrates the simple control of a GPIO pin on the Raspberry Pi.

## Source Code

The source code for this example is included in the github repository:
https://github.com/Pi4J/pi4j/tree/master/pi4j-example/src/main/java/ControlGpioExample.java

```java
/*
 * #%L
 * **********************************************************************
 * ORGANIZATION  :  Pi4J
 * PROJECT       :  Pi4J :: Java Examples
 * FILENAME      :  ControlGpioExample.java
 *
 * This file is part of the Pi4J project. More information about
 * this project can be found here:  http://www.pi4j.com/
 * **********************************************************************
 * %%
 * Copyright (C) 2012 - 2013 Pi4J
 * %%
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 * #L%
 */

import com.pi4j.io.gpio.GpioController;
import com.pi4j.io.gpio.GpioFactory;
import com.pi4j.io.gpio.GpioPinDigitalOutput;
import com.pi4j.io.gpio.PinState;
import com.pi4j.io.gpio.RaspiPin;

/**
 * This example code demonstrates how to perform simple state
 * control of a GPIO pin on the Raspberry Pi.
 *
 * @author Robert Savage
 */
public class ControlGpioExample {

    public static void main(String[] args) throws InterruptedException {

        System.out.println("<--Pi4J--> GPIO Control Example ... started.");

        // create gpio controller
        final GpioController gpio = GpioFactory.getInstance();

        // provision gpio pin #01 as an output pin and turn on
        final GpioPinDigitalOutput pin = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_01, "MyLED",
PinState.HIGH);
        System.out.println("--> GPIO state should be: ON");

        Thread.sleep(5000);

        // turn off gpio pin #01
        pin.low();
        System.out.println("--> GPIO state should be: OFF");

        Thread.sleep(5000);
        // toggle the current state of gpio pin #01 (should turn on)
        pin.toggle();
        System.out.println("--> GPIO state should be: ON");

        Thread.sleep(5000);

        // toggle the current state of gpio pin #01  (should turn off)
        pin.toggle();
        System.out.println("--> GPIO state should be: OFF");

        Thread.sleep(5000);

        // turn on gpio pin #01 for 1 second and then off
        System.out.println("--> GPIO state should be: ON for only 1 second");
        pin.pulse(1000, true); // set second argument to 'true' use a blocking call

        // stop all GPIO activity/threads by shutting down the GPIO controller
        // (this method will forcefully shutdown all GPIO monitoring threads and scheduled tasks)
        gpio.shutdown();
    }
}
```

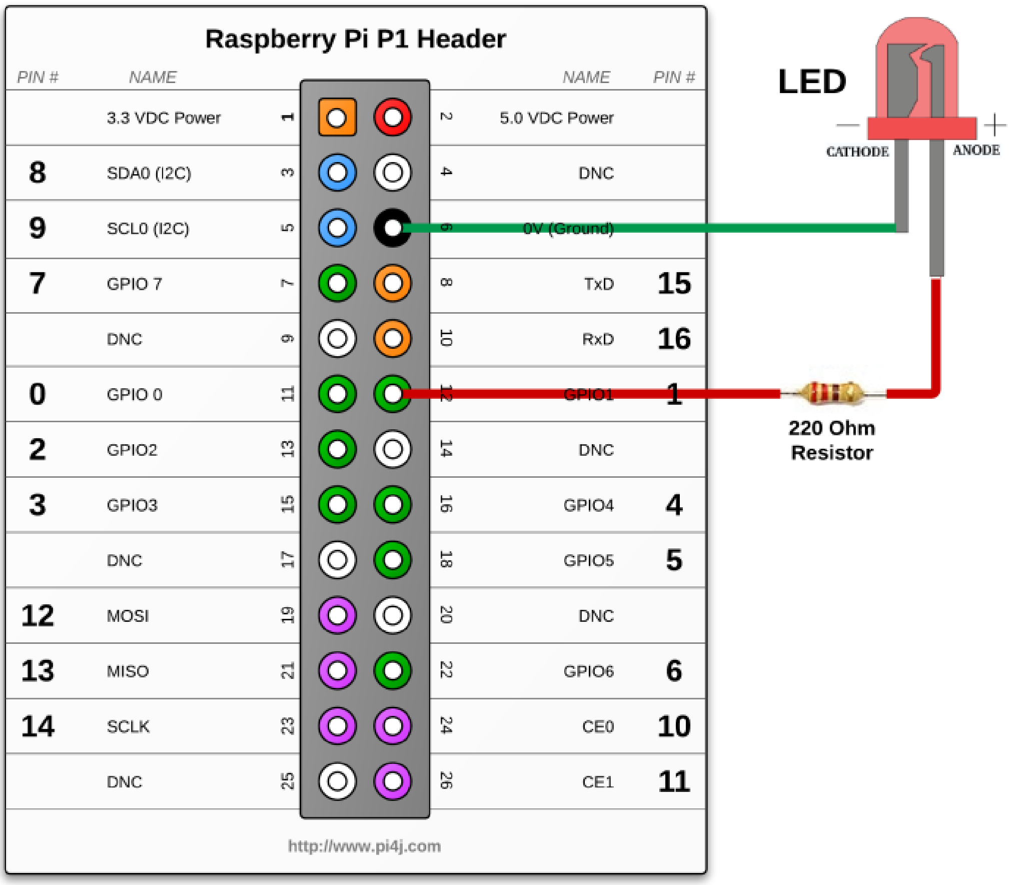
## JavaDoc

The following JavaDoc links are the primary interfaces used to control the Pi's GPIO state:

- com.pi4j.io.gpio.GpioFactory
- com.pi4j.io.gpio.Gpio
- com.pi4j.io.gpio.GpioPin
- com.pi4j.io.gpio.PinState

## Wiring Diagram

The following circuit can be used in conjunction with this sample code.



(click here for hi-resolution image)

## Navigate

If you have not already downloaded and installed the Pi4J library on the RaspberryPi, then view this page for

instructions on where to download and how to install Pi4J:
Download & Install Pi4J

First, locate the *ControlGpioExample.java* source file in the samples folder of the Pi4J installation on the RaspberryPi.
You can use the following command on the Pi's console or SSH terminal to navigate to this path:

```
cd /opt/pi4j/examples
```

## Compile

Next, use the following command to compile this example program:

```
javac -classpath .:classes:/opt/pi4j/lib/'*' -d . ControlGpioExample.java
```

## Execute

The following command will run this example program:

```
sudo java -classpath .:classes:/opt/pi4j/lib/'*' ControlGpioExample
```

## Output

You should see the attached LED perform as follows:

●
Turn ON for 5 seconds

●
Turn OFF for 5 seconds

●
Turn ON for 5 seconds

●
Turn OFF for 5 seconds

●
Turn ON for 1 second

●
Turn OFF