

# relay8

Arduino and chipKit library support for relay boards

## Manual



## PREFACE:

This library is just a quick and easy way to control relay boards. The library supports from 1 to 8 relays.

This library supports relays board with both active high and active low inputs. To set the correct parameter for your relay board you will have to edit [relay8.h](#). Set line 59 to `#define OUTPUT_ON_LEVEL HIGH` if your board requires a **HIGH** signal to switch a relay on. If your board requires a **LOW** signal to switch a relay on you can edit line 32 to be `#define OUTPUT_ON_LEVEL LOW`.

You can always find the latest version of the library at <http://electronics.henningkarlsen.com/>

If you make any modifications or improvements to the code, I would appreciate that you share the code with me so that I might include it in the next release. I can be contacted through <http://electronics.henningkarlsen.com/contact.php>.

For version information, please refer to `version.txt`.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

## FUNCTIONS:

**relay8(pin1 [,pin2 [,pin3 [,pin4 [,pin5 [,pin6 [,pin7 [,pin8]]]]]]]);**

The main class constructor.

Parameters:      pin1:    I/O pin for relay #1  
                  pin2:    <optional> I/O pin for relay #2  
                  pin3:    <optional> I/O pin for relay #3  
                  pin4:    <optional> I/O pin for relay #4  
                  pin5:    <optional> I/O pin for relay #5  
                  pin6:    <optional> I/O pin for relay #6  
                  pin7:    <optional> I/O pin for relay #7  
                  pin8:    <optional> I/O pin for relay #8

Usage:            relay8 relay(2, 3, 4, 5); // Initialize the library for 4 relays on pins 2, 3, 4 and 5.

**numberOfRelays();**

Get the number of relays currently controlled by the library.

Parameters:      none  
Returns:          <int> Number of relays currently controlled by the library  
Usage:            int relays = relay.numberOfRelays(); // Get the number of relays

**on(relay);**

Switch on one relay.

Parameters:      relay: Number of the relay to switch on [1-8]  
Usage:            relay.on(3); // Switch on relay #3

**off(relay);**

Switch off one relay.

Parameters:      relay: Number of the relay to switch off [1-8]  
Usage:            relay.off(3); // Switch off relay #3

**allOn();**

Switch on all the relays.

Parameters:      None  
Usage:            relay.allOn(); // Switch on all the relays

**allOff();**

Switch off all the relays.

Parameters:      None  
Usage:            relay.allOff(); // Switch off all the relays

#### **cycle([delay\_time]);**

Cycle through all relays from #1 to the last, then back down to #1 again.

Parameters:      delay\_time: **<optional>** Time in ms each relay will remain switched on. Default is 250ms.  
Usage:            relay.cycle(); // Cycle through all relays

#### **cycleUp([delay\_time]);**

Cycle up through all relays from #1 to the last.

Parameters:      delay\_time: **<optional>** Time in ms each relay will remain switched on. Default is 250ms.  
Usage:            relay.cycleUp(); // Cycle through all relays

#### **cycleDown([delay\_time]);**

Cycle down through all relays from the last down to #1.

Parameters:      delay\_time: **<optional>** Time in ms each relay will remain switched on. Default is 250ms.  
Usage:            relay.cycleDown(); // Cycle through all relays

#### **chaseUpOn([delay\_time]);**

Switch on all relays in sequence from #1 to the last relay.

Parameters:      delay\_time: **<optional>** Time in ms to delay before switching on the next relay. Default is 250ms.  
Usage:            relay.chaseUpOn(); // Switch on all relays in sequence from #1 to the last relay

#### **chaseUpOff([delay\_time]);**

Switch off all relays in sequence from #1 to the last relay.

Parameters:      delay\_time: **<optional>** Time in ms to delay before switching off the next relay. Default is 250ms.  
Usage:            relay.chaseUpOff(); // Switch off all relays in sequence from #1 to the last relay

#### **chaseDownOn([delay\_time]);**

Switch on all relays in sequence from the last relay down to #1.

Parameters:      delay\_time: **<optional>** Time in ms to delay before switching on the next relay. Default is 250ms.  
Usage:            relay.chaseDownOn(); // Switch on all relays in sequence from the last relay down to #1

#### **chaseDownOff([delay\_time]);**

Switch off all relays in sequence from the last relay down to #1.

Parameters:      delay\_time: **<optional>** Time in ms to delay before switching off the next relay. Default is 250ms.  
Usage:            relay.chaseDownOff(); // Switch off all relays in sequence from the last relay down to #1