

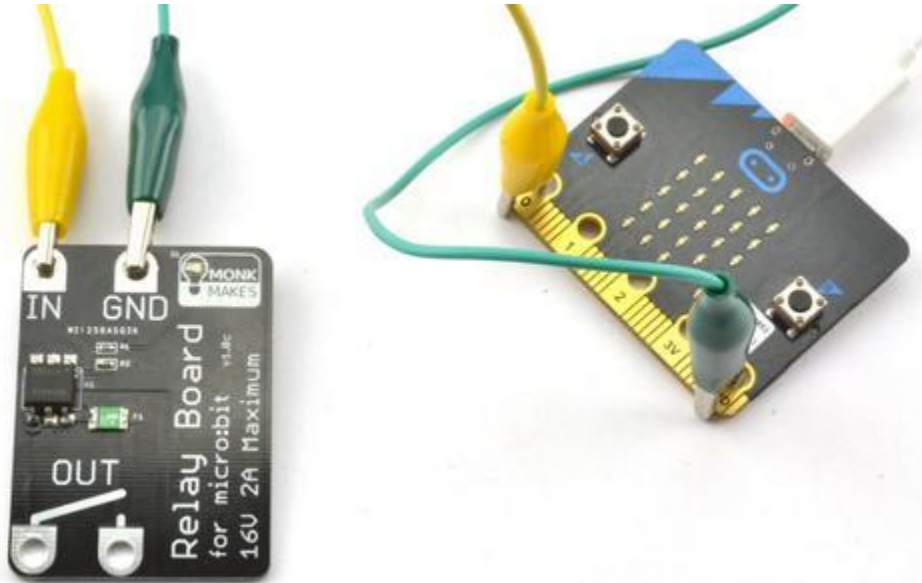
## Relay for micro:bit /MNK00061



The MonkMakes Relay for micro:bit is a solid-state (no moving parts) relay that allows an output of a micro:bit to turn things on and off.

A micro:bit can turn an LED on and off directly, but anything more powerful requires something like a relay or a transistor. Using a transistor to switch something on and off requires a shared ground connection with the micro:bit and a knowledge electronics that you or your students may not be ready for. The MonkMakes Relay for micro:bit is much easier to use, acting like a simple micro:bit controlled switch.

This relay can be used to switch low voltage devices such as light bulbs, a motor, a small heating element or even a string of 12V LED lighting. The voltage needs to be kept under 16V, but the relay will automatically protect itself against too much current.



## FEATURES

- Solid-state relay (up to 2 Amps)
- Active LED indicator
- Resettable 'polyfuse' to protect against over-current

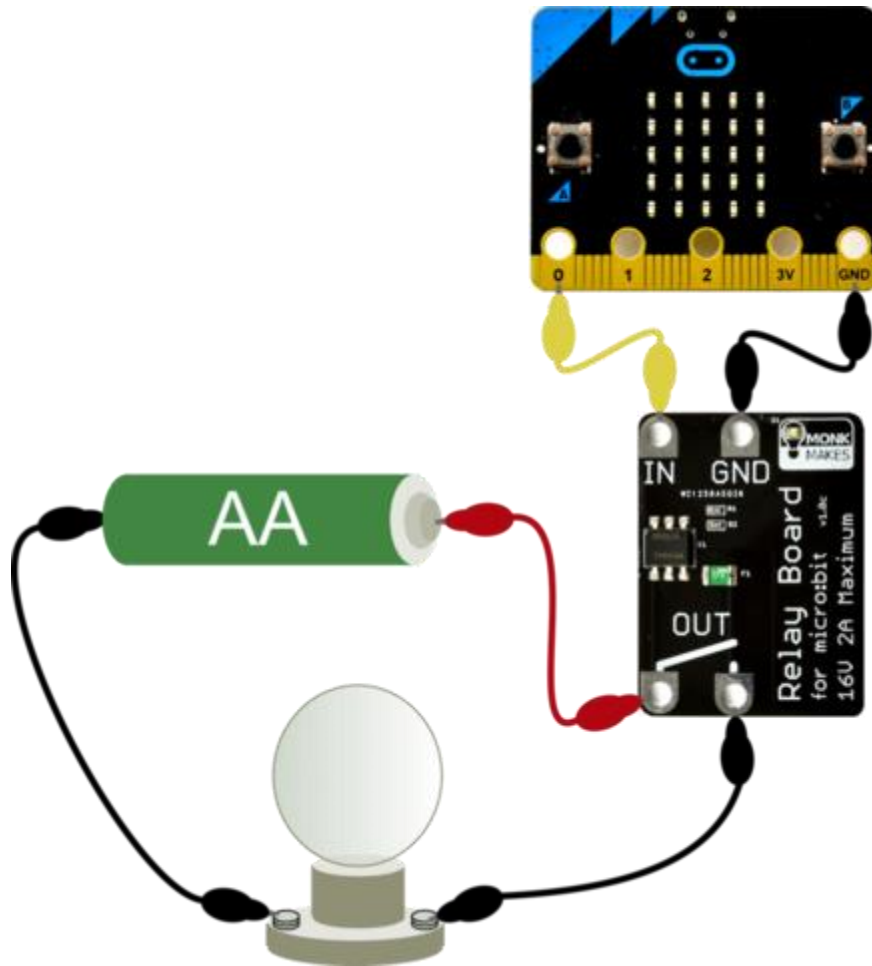
## Getting Started

Connecting your micro:bit

The Relay requires just two connections to the micro:bit. One to GND (ground) and one to whatever pin is to be used to control the relay's switching action.

When attaching the alligator clips to the micro:bit, make sure that the clips are perpendicular to the board so that they are not touching any of the neighbouring connectors on the micro:Bit edge connector.

Here's an example of how you could wire up a MonkMakes Relay for micro:bit to turn an old fashioned light bulb on and off.



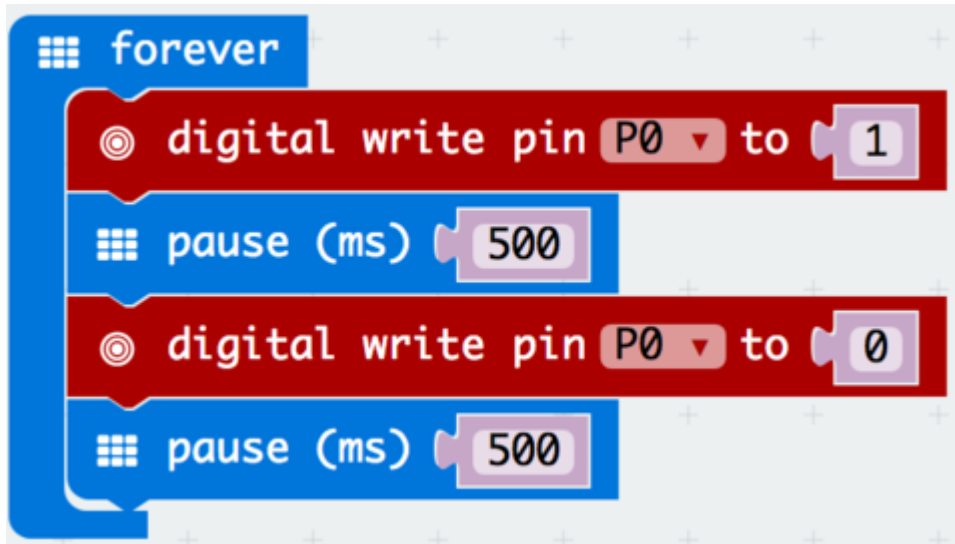
## HEX File

The quickest way to try out your relay is to [DOWNLOAD THIS HEX FILE >](#) and then copy it onto your micro:bit. The program will turn the relay on and off once a second.

## JavaScript Blocks Editor

Set the controlling pin to 1 and the relay contacts will close, set it to 0 and the contacts will open again. Its as simple as that. So, to make your relay turn on and off once a second, [open the Blocks Editor](#), add a *forever* block and then the *digital write* blocks from the *pins* category and the *pauses* from the *basic* category.

You can also click on the image of the blocks below to open up the Blocks Editor on this project.



## MicroPython

Paste the following code into the [Python window](#) and then Download the file and copy it onto your your micro:bit.

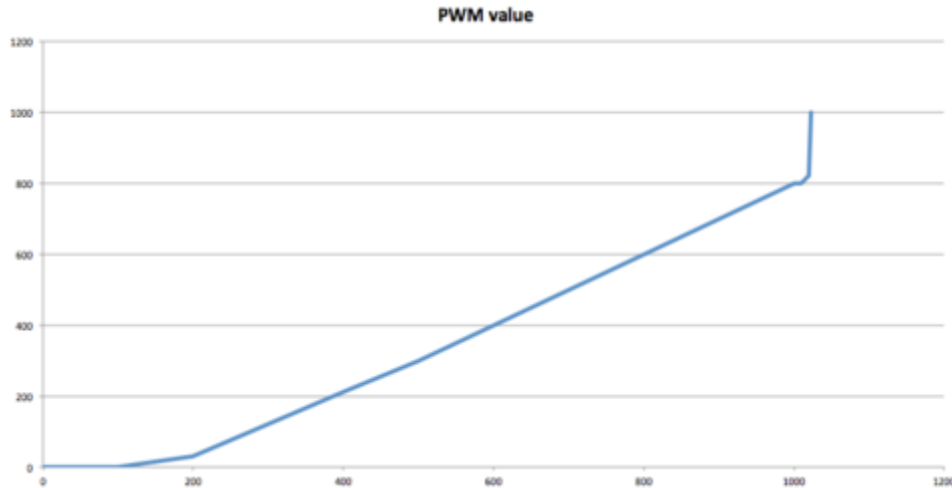
```
from microbit import *

while True:
    pin0.write_digital(True)
    sleep(500)
    pin0.write_digital(False)
    sleep(500)
```

## Analog Outputs

The latest MonkMakes Relay for micro:bit can do more than just switch things on and off. It can also be used with micro:bit analog outputs. Look closely at your Relay for micro:bit and is it has the version number **v1ev** (under the word 'Board') then it can be used with the 'analog write' block in the blocks editor or the 'write\_analog' function in MicroPython. If your board has the version number **v1e** then it is not suitable for use with analog outputs – sorry you were unlucky to get one of the small batch of first boards to be sold.

The output of the Relay for micro:bit is not linear at low PWM and high PWM values as the following chart illustrates.



The y-axis shows the current in mA for a test load resistor supplied from a constant voltage source. The x-axis is the analog write value (0 to 1023). As you can see, there is a dead zone up to a analog output value of about 100, followed by a relatively good linear region right up to about 1000, after which the output effectively becomes 'on'.

The tests were carried out at the default PWM frequency of 50Hz for the micro:bit. Lower frequency PWM is expected to produce more linear results.

