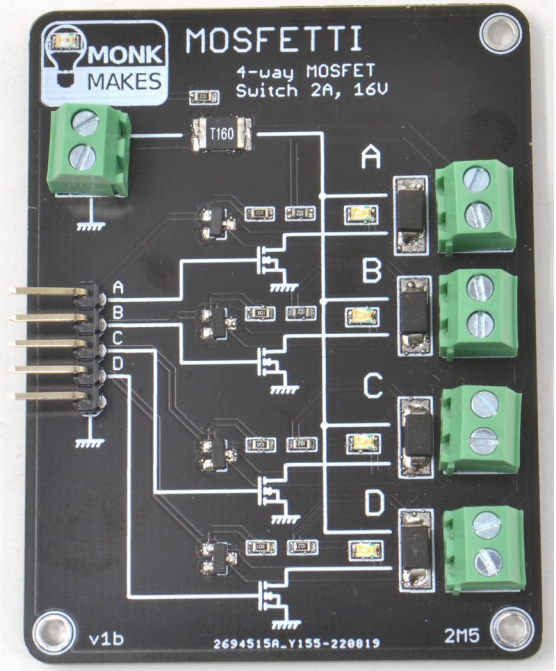
Instructions:

MOSFETTI



4-way MOSFET switch. Perfect for cars, motorhomes, model railways and other low voltage DC projects.

Compatible with Arduino, Raspberry Pi Pico, Raspberry Pi 4, Beagleboard and any 3V or 5V microcontroller.

Instructions version 1b.

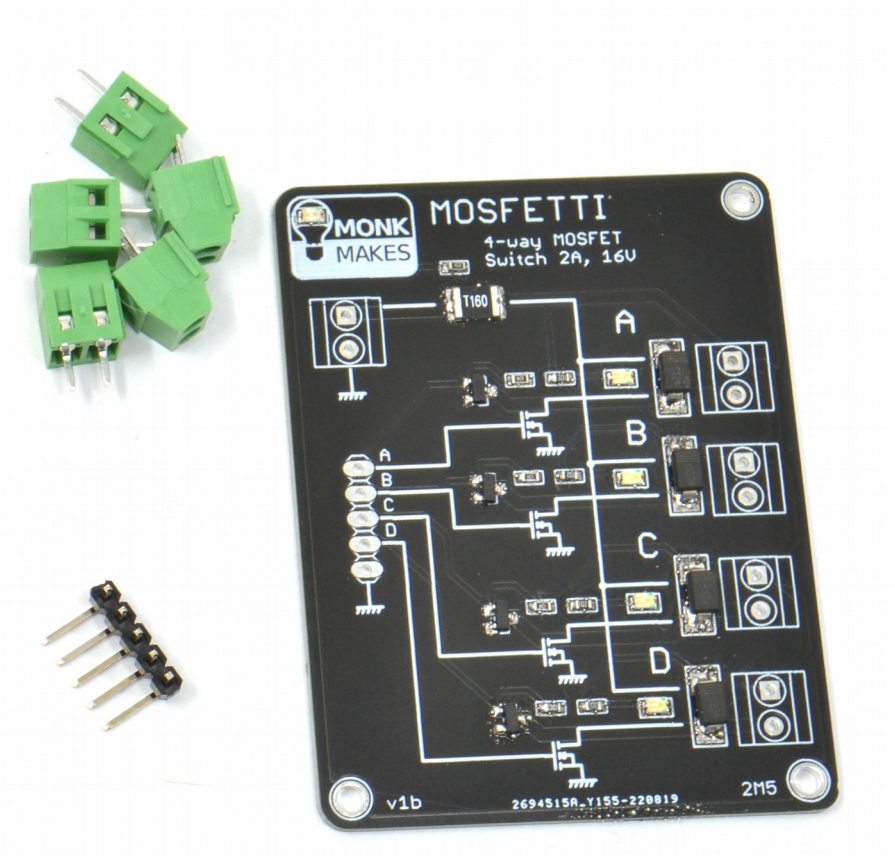
Table of Contents

Warning

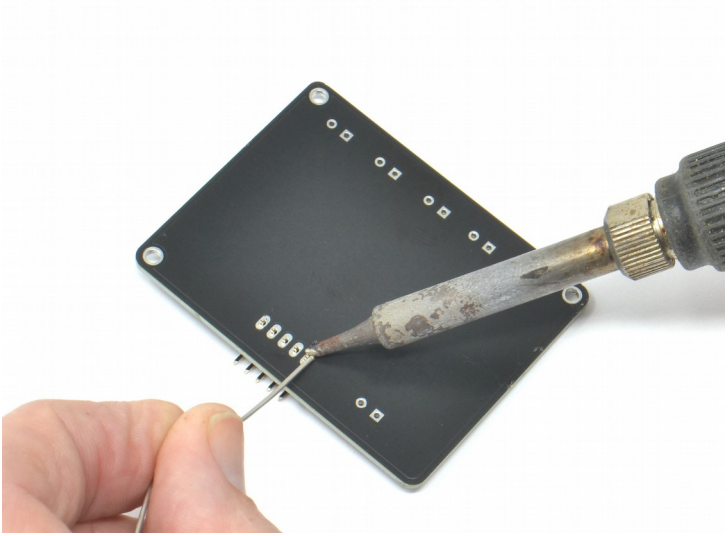
This product is for switching low voltage DC only. Under no circumstances should it be connected to AC.

Assembly

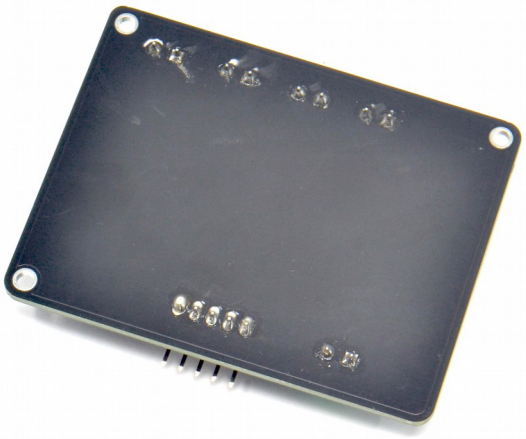
The Mosfetti comes as a kit, with all the surface mount components already soldered on. You just need to solder on the header pins and screw terminals. Some projects may be better with leads soldered directly to the pads.



Solder the header pins first, as they are the shortest. Put the pin headers in place then flip the board over onto its back and solder the pins. Sometimes a piece of adhesive putty is useful to keep the pins in position until the first pin is soldered.



Next solder the screw terminals into place, being careful to ensure that the part of the screw terminal into which the wire goes is towards the outside of the board.

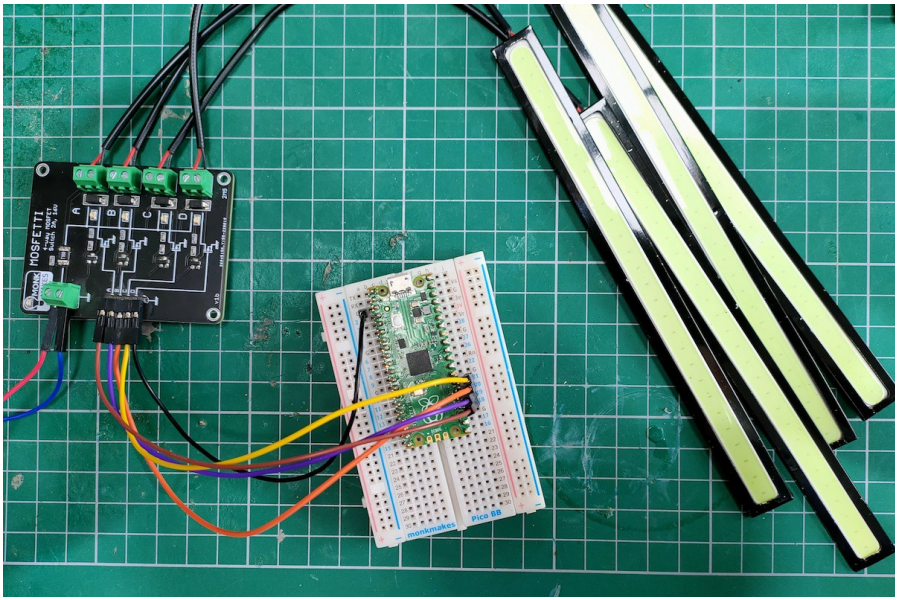


Using the Mosfetti

You can think of the Mosfetti as a 4-channel switch controllable from your favourite microcontroller or single-board-computer. You can switch DC loads such as motors and LED lighting modules (low-voltage DC only) that use far too much current to be connected to a GPIO pin directly. The Mosfetti is compatible with any 3 or 5V microcontroller, such as an Arduino or a Raspberry Pi Pico. You can also connect a Mosfetti to a single-board-computer with GPIO pins such as a Raspberry Pi 4.

Here's an example of how you might wire up a 12V DC power supply and four 12V LED lamps and then control it all from a microcontroller.

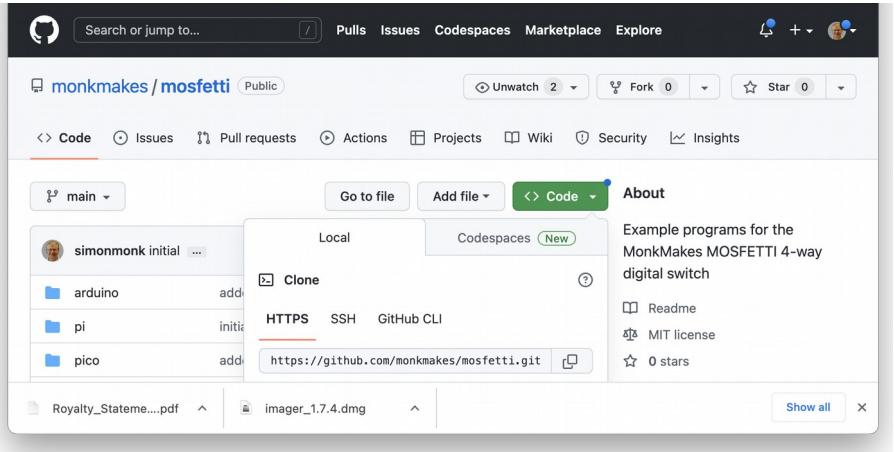
This example has been made with a Raspberry Pi Pico, and Arduino Uno and a Raspberry Pi 4.



The board has one *input* screw terminal for the supply voltage, and four *outputs*, to which motors, or LED strips or pumps can be attached. Each of the four channels labelled A to D and is controlled by the corresponding pin on the 5 way pin header. On the 5 pin header, the fifth pin is the ground or GND pin which must be connected to the microcontroller or single-board-computer controlling the Mosfetti.

Downloading the Examples

To download the ZIP archive containing example programs for all platforms, visit https://github.com/monkmakes/mosfetti



Click on the Code button, select Download ZIP and then extract the downloaded ZIP archive.

If you are familiar with git and would prefer to download the examples using the command line, then you can do so with the command:

```
$ git clone https://github.com/monkmakes/mosfetti.git
```

With the extracted archive, you will find folders called:

- pico – MicroPython examples for the Raspberry Pi Pico
- arduino – Arduino, ESP32 and other boards that can be programmed with the Arduino IDE
- pi – Raspberry Pi Python examples

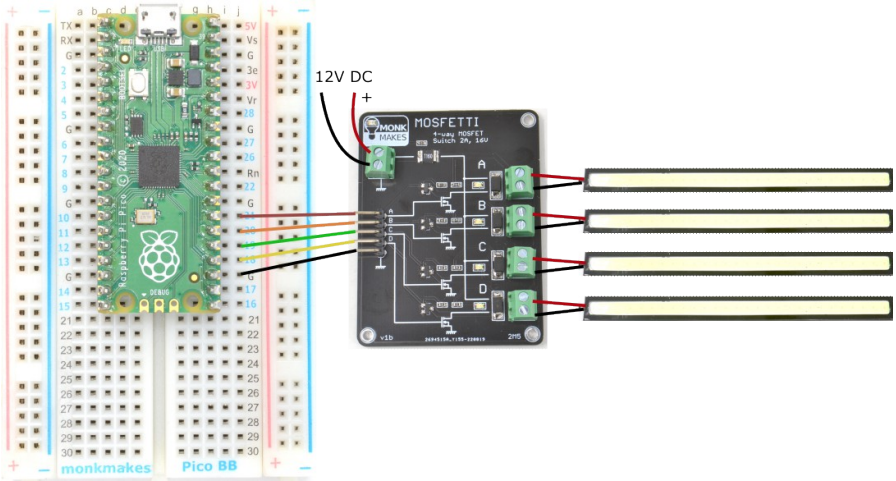
Raspberry Pi Pico

You will need

To build this example project you will need the following items:

- A MonkMakes Mosfetti Board
- A Raspberry Pi Pico
- Solderless Breadboard. The MonkMakes Breadboard for Pico has the Pico pinout printed on it, making pin identification a lot easier (https://monkmakes.com/pico_bb.html).
- 5 x female to male jumper wires
- 4 x LED lamp modules (12V). We used 6W COB lighting strips intended for use in cars.
- 12V power supply

Wiring – Raspberry Pi Pico



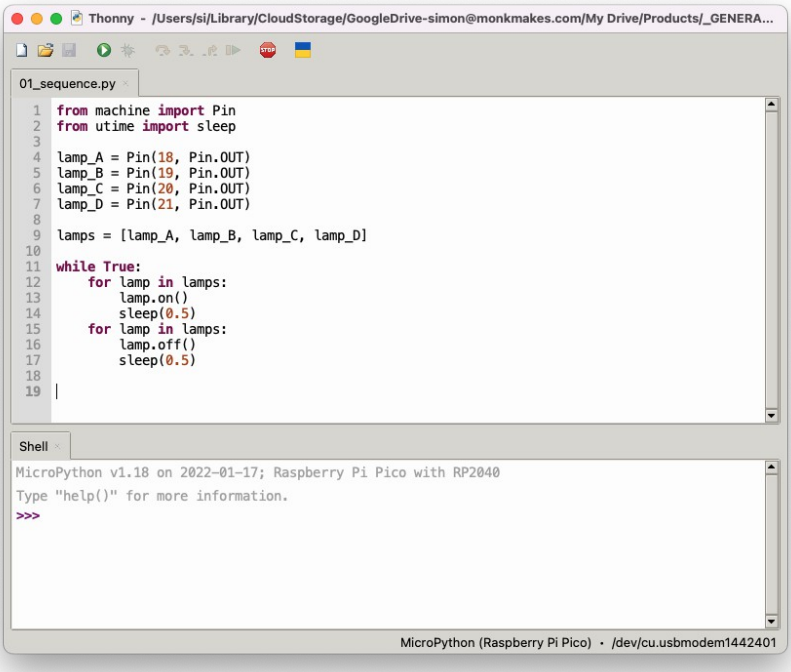
12V DC should be connected to the Mosfetti and the Pico powered from USB. The female to male jumper leads are used to connect the Pico's GND connection to the Mosfetti's GND and four of the Picos GPIO pins (18, 19, 20 and 21) are connected to the Mosfetti control pins A to D.

Example Software

In the *pico* folder of the Examples folder that you downloaded (see Page 6) you will find two programs: 01_sequence.py and 02_fade.py.

Turning Lamps on and off

Open 01_sequence.py in Thonny (https://thonny.org/) and run it on your Pico. You should see the LED lamps light up in turn and then when all are lit, turn off in sequence.



Here is the code for 01_sequence.py.

```
from machine import Pin
from utime import sleep

lamp_A = Pin(18, Pin.OUT)
lamp_B = Pin(19, Pin.OUT)
lamp_C = Pin(20, Pin.OUT)
```

```
lamp_D = Pin(21, Pin.OUT)

lamps = [lamp_A, lamp_B, lamp_C, lamp_D]

while True:
    for lamp in lamps:
        lamp.on()
        sleep(0.5)
    for lamp in lamps:
        lamp.off()
        sleep(0.5)
```

Each of the four lamps is associated with a different control pin. A list (`lamps`) is created that contains all four lamps, making it easy to iterate over each one in turn.

PWM

As well as turning things on and off, the Mosfetti is also capable of Pulse Width Modulation (PWM) to control the brightness of a lamp, or the speed of a motor. The example in 02_fade.py illustrates this.

```
from machine import Pin, PWM
from utime import sleep

lamp_A = PWM(Pin(18, Pin.OUT), 1000)
lamp_B = PWM(Pin(19, Pin.OUT), 1000)
lamp_C = PWM(Pin(20, Pin.OUT), 1000)
lamp_D = PWM(Pin(21, Pin.OUT), 1000)

lamps = [lamp_A, lamp_B, lamp_C, lamp_D]

while True:
    for lamp in lamps:
        for brightness in range(0, 255):
            lamp.duty_u16(brightness * 256)
            sleep(0.01)
    for lamp in lamps:
        lamp.duty_u16(0)
```

This time, each control pin is a PWM pin and the brightness is set using the `lamp_duty_u16` method. This expects a brightness value of between 0 and 65535. The example program increases the brightness in steps of 256.

Try running the example, and you should see how the first lamp gradually works its way up to full brightness and then the next light and so on.

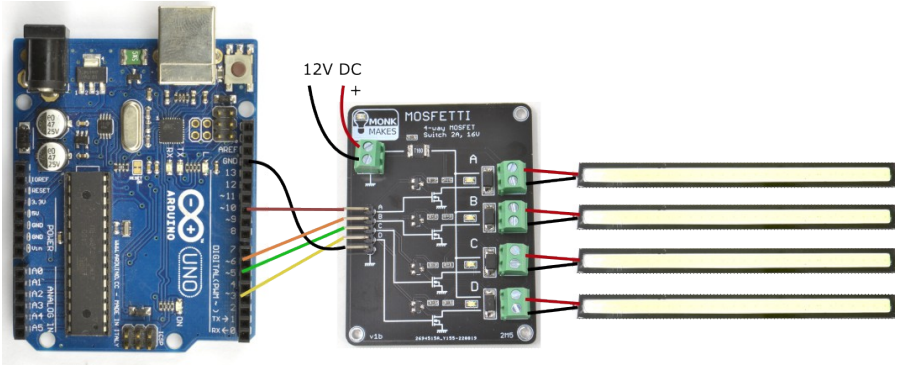
Arduino

You will need

To build this example project you will need the following items:

- A MonkMakes Mosfetti Board
- An Arduino Uno, or other Arduino board. If you use another Arduino -compatible board like an ESP32-based board, then you will need to change the pins that you use. Pick pins that are PWM capable.
- 5 x female to male jumper wires.
- 4 x LED lamp modules (12V). We used COB lighting strips intended for use in cars.
- 12V power supply

Wiring – Arduino Uno



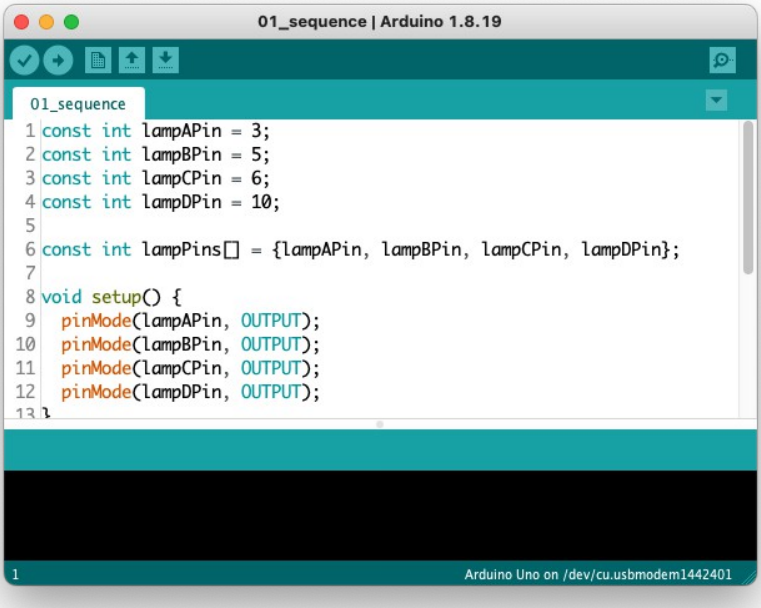
Connect the Arduino and Mosfetti grounds, and four control pins using female to male jumper wires. Use Arduino Uno pins 3, 5, 6 and 10, as these are PWM capable.

Example Software

In the *arduino* folder of the Examples folder that you downloaded (see Page 6) you will find two programs: 01_sequence.ino and 02_fade.ino.

Turning Lamps on and off

Open 01_sequence.ino in the Arduino IDE and upload it to the Arduino. You should see the LED lamps light up in turn and then when all are lit, turn off in sequence.



Here is the code for this:

```
const int lampAPin = 3;
const int lampBPin = 5;
const int lampCPin = 6;
const int lampDPin = 10;

const int lampPins[] = {lampAPin, lampBPin, lampCPin,
lampDPin};

void setup() {
  pinMode(lampAPin, OUTPUT);
  pinMode(lampBPin, OUTPUT);
  pinMode(lampCPin, OUTPUT);
  pinMode(lampDPin, OUTPUT);
}
```

```
void loop() {
    for (int i = 0; i < 4; i++) {
      digitalWrite(lampPins[i], HIGH);
      delay(500);
    }
    for (int i = 0; i < 4; i++) {
      digitalWrite(lampPins[i], LOW);
      delay(500);
    }
}
```

PWM

As well as turning things on and off, the Mosfetti is also capable of Pulse Width Modulation (PWM) to control the brightness of a lamp, or the speed of a motor. The example in 02_fade.ino illustrates this.

```
const int lampAPin = 3;
const int lampBPin = 5;
const int lampCPin = 6;
const int lampDPin = 10;

const int lampPins[] = {lampAPin, lampBPin, lampCPin, lampDPin};

void setup() {
  pinMode(lampAPin, OUTPUT);
  pinMode(lampBPin, OUTPUT);
  pinMode(lampCPin, OUTPUT);
  pinMode(lampDPin, OUTPUT);
}

void loop() {
    for (int i = 0; i < 4; i++) {
      for (int brightness = 0; brightness < 255; brightness++) {
        analogWrite(lampPins[i], brightness);
        delay(10);
      }
    }
    for (int i = 0; i < 4; i++) {
      digitalWrite(lampPins[i], LOW);
    }
}
```

For each lamp in turn analogWrite is used to ramp the brightness up from 0 (off) to 255 (fully on).

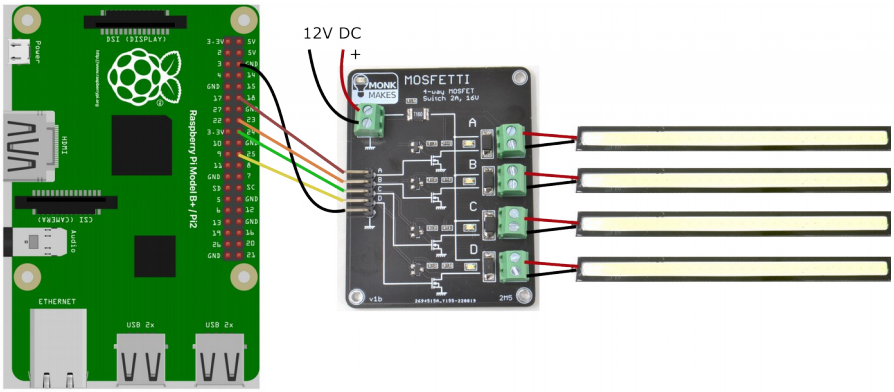
Raspberry Pi

You will need

To build this example project you will need the following items:

- A MonkMakes Mosfetti Board
- A Raspberry Pi 4. Older versions of Raspberry Pi will also work. The Raspberry Pi should have the latest version of Raspberry Pi OS installed.
- 5 x female to female jumper wires.
- A Raspberry Leaf GPIO template (http://www.monkmakes.com/leaf.html) will make it easier to work out which pin is which.
- If you have a Raspberry Pi 400, then a GPIO adapter will make it easier to connect the Mosfetti. (http://www.monkmakes.com/pi_400_gpio.html).
- 4 x LED lamp modules (12V). We used 6W COB lighting strips intended for use in cars.
- 12V power supply

Wiring



Use female to female jumper wires to connect the ground and control lines of the Raspberry Pi to the Mosfetti. I used GPIO pins 18, 23, 24 and 25 of the Raspberry Pi.

Example Software

In the *pi* folder of the Examples folder that you downloaded (see Page 6) you will find two programs: 01_sequence.py and 02_fade.py.

Turning Lamps on and off

On your Raspberry Pi, open a terminal session and if you haven't already done so, download the example code and change to the pi folder using the commands below:

```
$ git clone https://github.com/monkmakes/mosfetti.git
$ cd mosfetti/pi
```

You can then run the first example program like this:

```
$ python 01_sequence.py
```

Here's the code.

```
from gpiozero import LED
from time import sleep

lamp_A = LED(18)
lamp_B = LED(23)
lamp_C = LED(24)
lamp_D = LED(25)

lamps = [lamp_A, lamp_B, lamp_C, lamp_D]

while True:
    for lamp in lamps:
        lamp.on()
        sleep(0.5)
    for lamp in lamps:
        lamp.off()
        sleep(0.5)
```

The code uses the gpiozero module to control the GPIO pins. Each lamp is assigned its own pin and a list defined to contain all four lamps.

PWM

As well as turning things on and off, the Mosfetti is also capable of Pulse Width Modulation (PWM) to control the brightness of a lamp, or the speed of a motor. The example in 02_fade.py illustrates this.

```
from gpiozero import PWMLED
from time import sleep

lamp_A = PWMLED(18)
lamp_B = PWMLED(23)
lamp_C = PWMLED(24)
lamp_D = PWMLED(25)

lamps = [lamp_A, lamp_B, lamp_C, lamp_D]

while True:
    for lamp in lamps:
        for brightness in range(0, 255):
            lamp.value = brightness / 255
            sleep(0.01)
    for lamp in lamps:
        lamp.value = 0
```

In this code, the `gpiozero` class `PWMLED` is used, as we want to control the brightness of the lamps.

To set the brightness of a lamp, its `value` property is set to a number between 0.0 and 1.0.

Troubleshooting

Problem: The orange power LED in the MonkMakes logo on the Mosfetti does not light.

Solution: Make sure that the power terminal block is properly connected to a DC voltage between 3 and 16V with the correct polarity.

Problem: The orange power LED in the MonkMakes logo on the Mosfetti is lit, but none of the green output LED's light when a control pin is taken high.

Solution: Make sure that there is not a short circuit, or a load greater than 2A on one of the outputs. Also, make sure that the GPIO pin you are using is set to be an output in your code and is properly connected to the GPIO pin. Che

Problem: I accidentally short-circuited one of the outputs. Have I killed my Mosfetti.

Solution: Not necessarily, the Mosfetti includes a self-resetting fuse, that will often (but not always) protect the Mosfetti against overloading. Disconnect the load and let the Mosfetti cool down for a half a minute and then try the board again. If the green indicator LEDs light, then it shows that the output is switching OK.

Support

You can find the Product's information page here: https://monkmakes.com/mosfetti including a datasheet for the product.

If you need further support, please email support@monkmakes.com.

MonkMakes

As well as this kit, MonkMakes makes all sorts of kits and gadgets to help with your electronics projects. Find out more, as well as where to buy here: https://monkmakes.com you can also follow MonkMakes on Twitter @monkmakes.