

PicoBuck Hookup Guide V12

Introduction

Please note that this tutorial is for the newest version of the PicoBuck, V12. If you have an older version of the PicoBuck, please refer to this tutorial.

Developed in collaboration with Ethan Zonca of profusion, the PicoBuck is a small-size, triple-output, constant current LED driver. By default, each channel is driven at 330mA; that current can be reduced by either presenting an analog voltage or a PWM signal to the board. Version 12 of the board adds a solderable jumper that can be closed to increase the maximum current to 660mA. It also increased the voltage rating on the various components on the board, allowing the board to be used up to the full 36V rating of the AL8805 part.



Suggested Reading

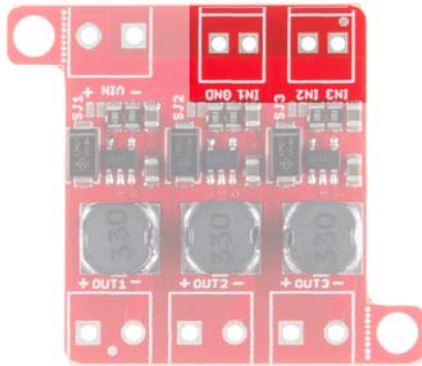
Here are some topics you should know before using the PicoBuck. Have a look if you need more information.

- LEDs
- Resistors
- Voltage
- Current
- Electric Power
- Pulse Width Modulation

PicoBuck Overview

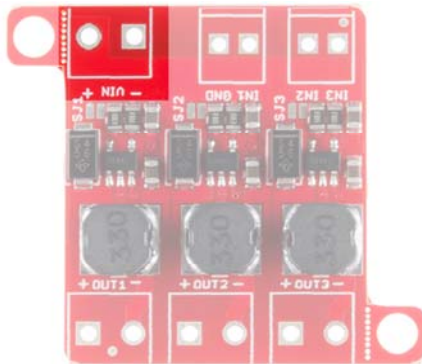
Since the PicoBuck is a constant current driver, the current drawn from the supply will drop as supply voltage rises. At 12V, the PicoBuck drives the three LEDs on our Rebel Triple Play board at 350mA per LED while drawing less than 350mA **total** from the power supply.

Three signal inputs are provided for dimming control. You can use the PWM signal from an Arduino or your favorite microcontroller to dim each channel individually, or you can tie them all to the same PWM for simultaneous dimming. A separate ground pin (labeled GND) is provided to reference against the controlling module for accuracy. The pin spacing for the two pairs of pins is 0.1", but the two pairs are slightly 0.2" apart, to allow for a 2.54mm pitch screw terminal pair to be used, or for a five-position standard 0.1" header with the middle pin removed. Dimming can be done by an analog voltage (20%-100% of max current by varying voltage from .5V-2.5V) or by PWM (so long as PWM minimum voltage is less than .4V and maximum voltage is more than 2.4V) for a full 0-100% range.

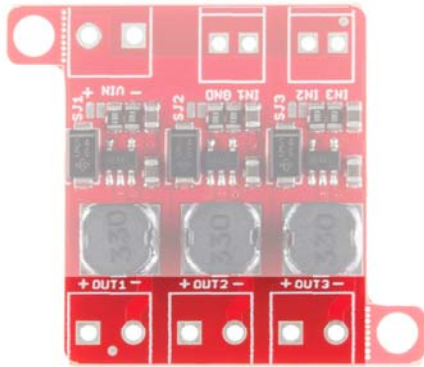


Signal Inputs

The power supply pads are sized for 3.5mm screw terminals, as are the output pads. Each output is independent from the other two.



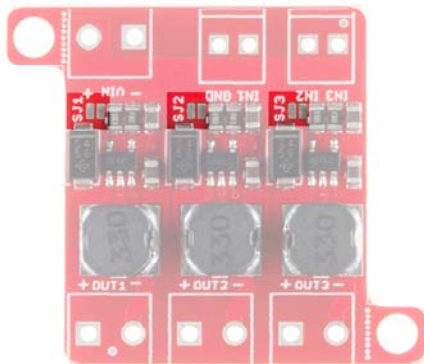
Power Input



Output Pads

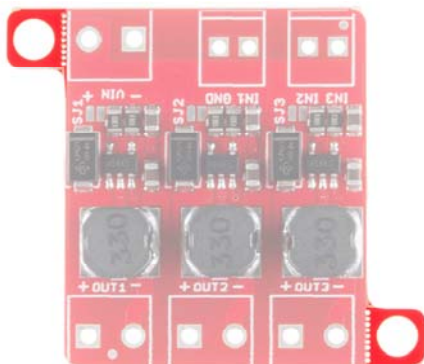
Note: The PicoBuck **cannot** be used to drive a common anode or common cathode LED or LED string, and the individual channel +/- pads **must not** be connected to one another.

A small jumper is provided for each channel to allow you to increase the drive strength from 330mA to 660mA. More information on this can be found below.



Solder Jumpers

Two mounting holes for 4-40 or M3 screws are provided on either side of the board. They are perforated so they can be easily snapped off with a pair of pliers, if a smaller footprint is desired.

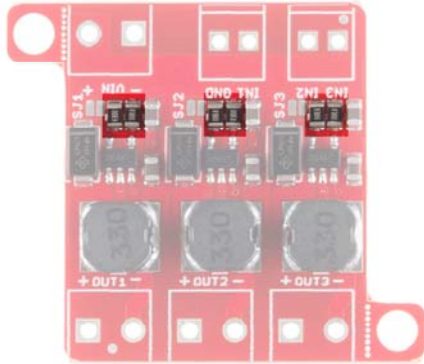


Mounting Holes

It is possible to increase the maximum current of the PicoBuck board up to 1A per channel; to do so, replace the three current sense resistors with smaller values. To calculate the new value for the resistor, use this formula:

$$I_{LED} = 0.1 / R_{set}$$

Thus, for a 1A current, you'd want a 0.1Ω resistor. Don't forget to be wary of current ratings. At 1A, the sense resistor will be dissipating 1/10W, so you probably want a resistor of at least 1/8W rating. The package is a standard 0805.



Current Set Resistors

Closing the Current Setting Jumpers



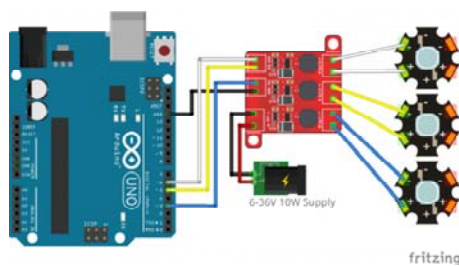
As you can see from the image above, the solder jumper doesn't need to be closed particularly neatly. All of the pads in its vicinity are connected to it anyway, so if you glob a little extra solder on, it's no big deal. Just be careful not to actually short the resistors, as in the rightmost circuit!

Dimming with a Microcontroller

As mentioned earlier, the PicoBuck's three channels can be individually dimmed by either varying the input voltage of the channel from 0-2.5V or using a PWM signal. The PWM signal from an Arduino board is perfectly suited for this.

Connecting one LED per channel

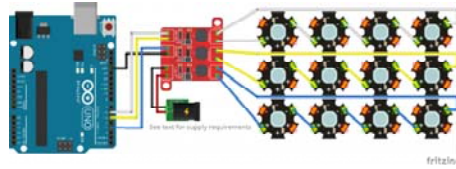
Here's a diagram showing how to connect the PicoBuck to an Arduino.



Note: Despite using an external supply for the PicoBuck, the grounds of the two boards must be connected! If the power supply is 12V or less, the Arduino can be powered from it as well, but do not attempt to power the PicoBuck from the Arduino!

Also note that each channel must be independently connected to the + and - connections of the LED it is to drive! **Do not** connect the + or - connections of any two channels together.

More than one LED per channel



Multiple LEDs can be connected in series, as shown, and the supply voltage should be at least 2-3V higher than the sum of the forward voltages of the LEDs.

Multiple LEDs can be connected per channel; they should be connected in series, as shown above, and the power supply voltage must be at least 1-2V higher than the sum of the forward voltages of the LEDs.

For instance, our blue 3W LEDs have a forward voltage of 3.2V to 3.8V. To be on the safe side, use the highest voltage in the range. If you want to connect four of them, you'd need a power supply of ~17V or greater ($3.8V + 3.8V + 3.8V + 3.8V = 15.2V$; add 2V of "head room").

Since 17V is greater than the Arduino can tolerate on its input, we have to provide an external supply for the Arduino as well. This can be the standard 5V USB supply.

It's perfectly acceptable to mix colors either between channels or on one channel, so long as all of the LEDs can handle the current (330mA or 660mA, depending on the jumper setting). Just make sure that the power supply voltage is high enough to handle the sum voltages of the highest voltage string. There is also no requirement that the three strings of LEDs have the same forward voltage of LEDs across them; you could have one white LED on channel 1, two red LEDs on channel 2, and four green LEDs on channel 3.

Code example

Code for controlling this device is trivial; simply use the `analogWrite()` function to adjust the brightness via PWM.

```
PicoBuck_Example (https://codebender.cc/code/AL880514) Download
referrer=sparkfun) by sparkfun (https://codebender.cc/user/sparkfun?
referrer=sparkfun) Breakout Example (/? (https://codebender.cc/code/AL880514)
3 * Mike Hord @ SparkFun Electronics
4 * Nov 5 2015 referrer=sparkfun) referrer=sparkfun
5 *
6 * A simple example showing how to control a PicoBuck with
7 *
8 * License: http://opensource.org/licenses/MIT
9 *
10 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF AN
11 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
12 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
13 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DA
14 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTH
15 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
16 * THE SOFTWARE.
17 */
18
19 const int CHL_1 = 3;
20 const int CHL_2 = 5;
21 const int CHL_3 = 6;
22
23 void setup()
24 {
25     pinMode(CHL_1, OUTPUT);
26     pinMode(CHL_2, OUTPUT);
27     pinMode(CHL_3, OUTPUT);
28 }
29
30 void loop()
```

Resources and Going Further

Thanks for reading. Below are all the documents and links you'll need to learn even more about the PicoBuck.

- [PicoBuck Schematic](#)
- [PicoBuck GitHub Repository](#)
- [AL8805W5 Datasheet](#)
- [Profusion Page](#)