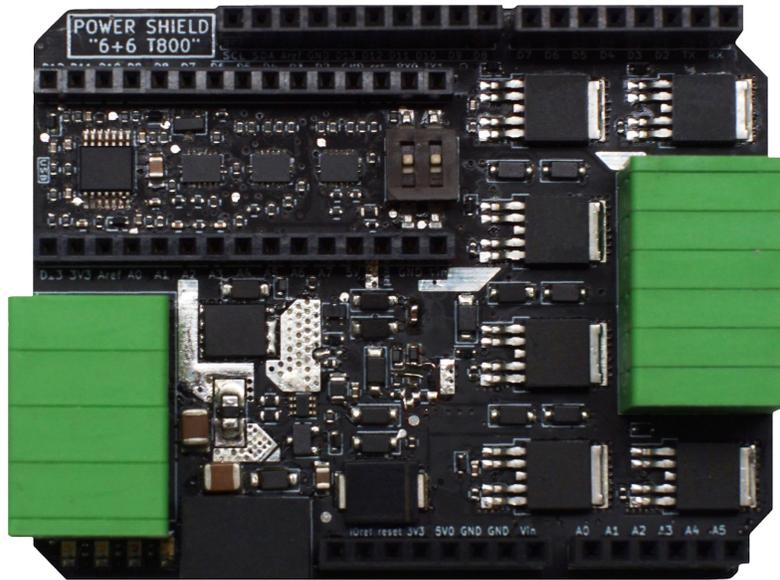


Multifunctional, Stackable & Multi-Protected 800W *POWER SHIELD 6+6 T800* for Arduino



[Fig. 1 – POWER SHIELD 6+6 T800, top view]

User manual

Updated: August 29, 2018

www.v-vTech.com

Table of Contents

[Table of Contents](#)

[Product overview](#)

[Electrical safety](#)

[Main features](#)

[Pin configuration](#)

[Compatibility](#)

[Block diagram](#)

[Simplified circuit diagram](#)

[Power supply](#)

[Input/output voltages](#)

[Current, limits and protection](#)

[Impedance](#)

[Protection](#)

[FeedBacks](#)

[LEDs on board](#)

[Outputs to analogue inputs](#)

[Frequency](#)

[Duty cycle vs. frequency](#)

[How to connect](#)

[Connecting the power supply](#)

[Low voltage](#)

[Connecting motors](#)

[Connecting DC motors](#)

[DC motor direction of rotation](#)

[Stepper motor \(UniPolar only\)](#)

[Connecting halogens](#)

[Connecting LED strips](#)

[High efficiency \(LED\) driving \(with step-down function\)](#)

[Driving 100W 35V LEDs easily](#)

[Voltage booster \(step-up converter\)](#)

[Connecting the transformer](#)

[Mixed connection application](#)

[Stacking – double power – double outputs](#)

[Multiplying PS outputs](#)

[Merging PS outputs](#)

[Demo programs for Arduino](#)

[Download and test sketches](#)

[Additional tips and tricks](#)

[Control the PS directly with push buttons](#)

[Use PS as Nano expansion board](#)

[General tips and warnings](#)

[Troubleshooting](#)

[Best wishes!](#)

[Contacts](#)

Product overview

POWER SHIELD 6+6 T800 (referred to as PS) is an add-on to Arduino-type boards. PS is designed to handle strong (up to 25A) currents. Each board has 6 channels for controlling 6 loads. Each power output can be turned on and off and its power can be adjusted with pulse-width modulation (PWM). Loads can be of any type (inductive, resistive or capacitive).

To get more channels and more power, PS boards can be stacked.

PS also has many safeguards to save time for professionals when creating a secure system and to prevent inexperienced users from burning the system.

In addition, this PS board still has 4 colour LED indicators and 3 backlinks. The backlinks lead to Arduino analogue inputs. One of the backlinks is an analogue current meter.

Electrical safety

1. This device is sensitive to electrostatic discharge (ESD).
2. Always use input fuses to prevent fire in case of product failure.
3. For high loads and high frequencies, please note that the device may become hot. Do not touch when operating and for some period thereafter.
4. Use in circuits not specified in this manual is at user's own risk.
5. Users should be 18 years or older and have good knowledge of electronics or use this device with professional oversight.

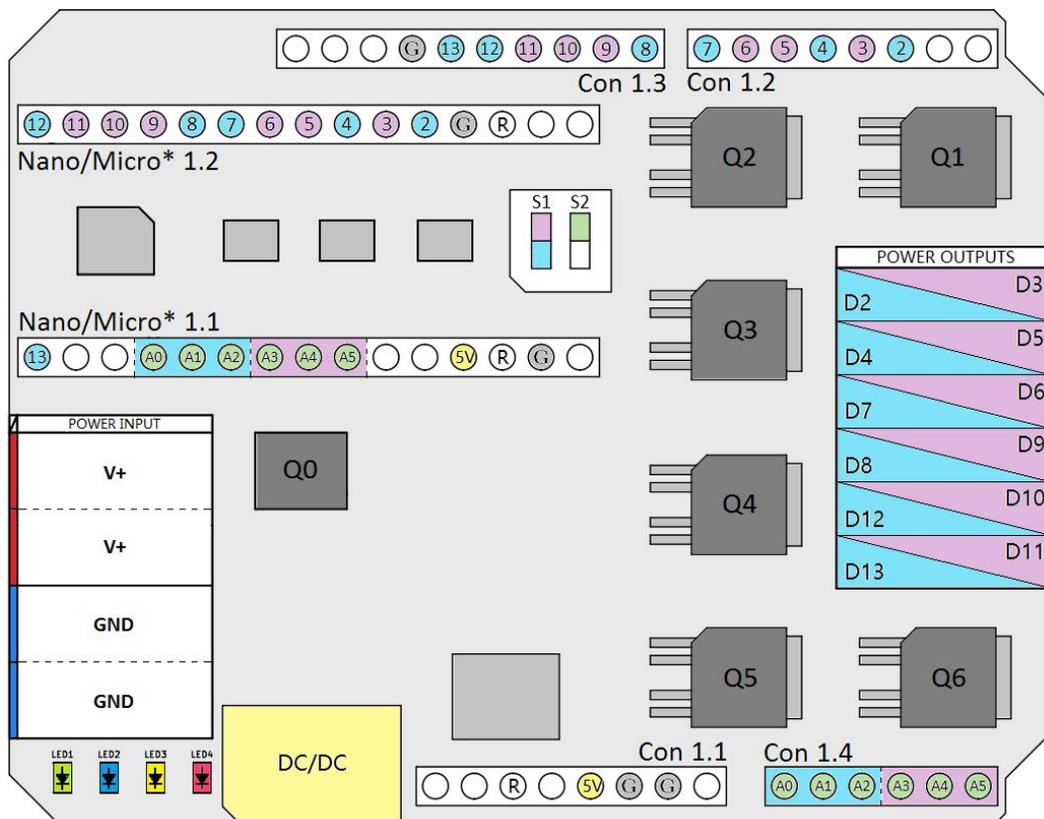
Main features

1. Wide operating input voltage range 6.5...32V (32V is the absolute maximum).

2. Each output can control independent voltage sources in the range of 1...32V.
3. High output frequencies:
 - Without heat sinks: at full load 0...2kHz, fully covering Arduino board's default PWM speed.
 - With additional heat sinks on transistors: 4...25kHz at full power. 25...100kHz with smaller currents.
4. Dozens of protections for beginners and professionals:
 - FlashBack® (FlyBack) protection
 - Five-level overcurrent protection!
 - Overtemperature protection
 - Power input polarity protection
 - Too low logic +5V supply *warning* & protection
 - Logic input & analogue output ESD protection
 - Logic circuit overvoltage protection (TVS & PPTC)
 - Logic circuit overcurrent protection with 0.5A PPTC
5. Three FeedBack outputs to Arduino's analogue (user selectable) inputs. Current measure expression 1V/10A (1V/20A when two boards are stacked).
6. Four coloured status LED indicators on the PCB.
7. Input/output logic levels are universal 5.0/3.3/2.5V and work with many boards.
8. Integrated high efficiency step-down DC/DC converter 0.5A to supply Arduino +5V pins and the module itself.
9. Pins compatible with most popular Arduino boards (including NANO, UNO, and MEGA).
10. High quality components and power connectors with spring-latch technology.
11. Maximum continuous total output power up to **800W**. Maximum output power when stacked up to **1.5kW**.
12. Operating temperature: -20...+50° degrees Celsius.

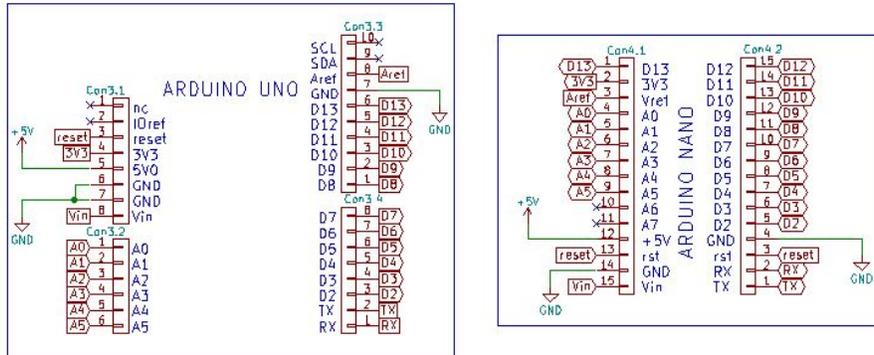
Pin configuration

Has colour-coded pins (matching the Arduino map) used by the PS board.



[Fig. 2 – PS top plan]

Arduino Nano connections are connected in parallel with UNO connections and are fully compatible:



[Fig. 3 – Relationship between Arduino UNO and Arduino NANO shield headers]

Compatibility

PS is compatible with almost all Arduino-based development boards, making it quite versatile. In addition to working for intended use cases it is also compatible with processor platforms both already available on the market and rapidly developing. Here are just a few of them:

Original products

- Arduino/Genuino MEGA
- Arduino UNO
- Arduino NANO
- Intel® Edison
- Analog Devices EVAL-ADICUP360
- Arduino Duemilanove
- Arduino YUN
- Intel® Cyclone 10 LP Evaluation Board
- Arduino Micro
- Arduino Leonardo
- Arduino 101
- BARE CONDUCTIVE Touch Board
- Arduino ZERO
- Arduino DUE
- Arduino PRO
- Arduino M0
- And more

Third-party products

- Infineon XMC4700
- ST NUCLEO
- ST NUCLEO (NANO size)
- KeyDuino
- FireBird32
- RaspDuino (transition to Raspberry Pi)
- Cytron ARM Cortex M0
- And more

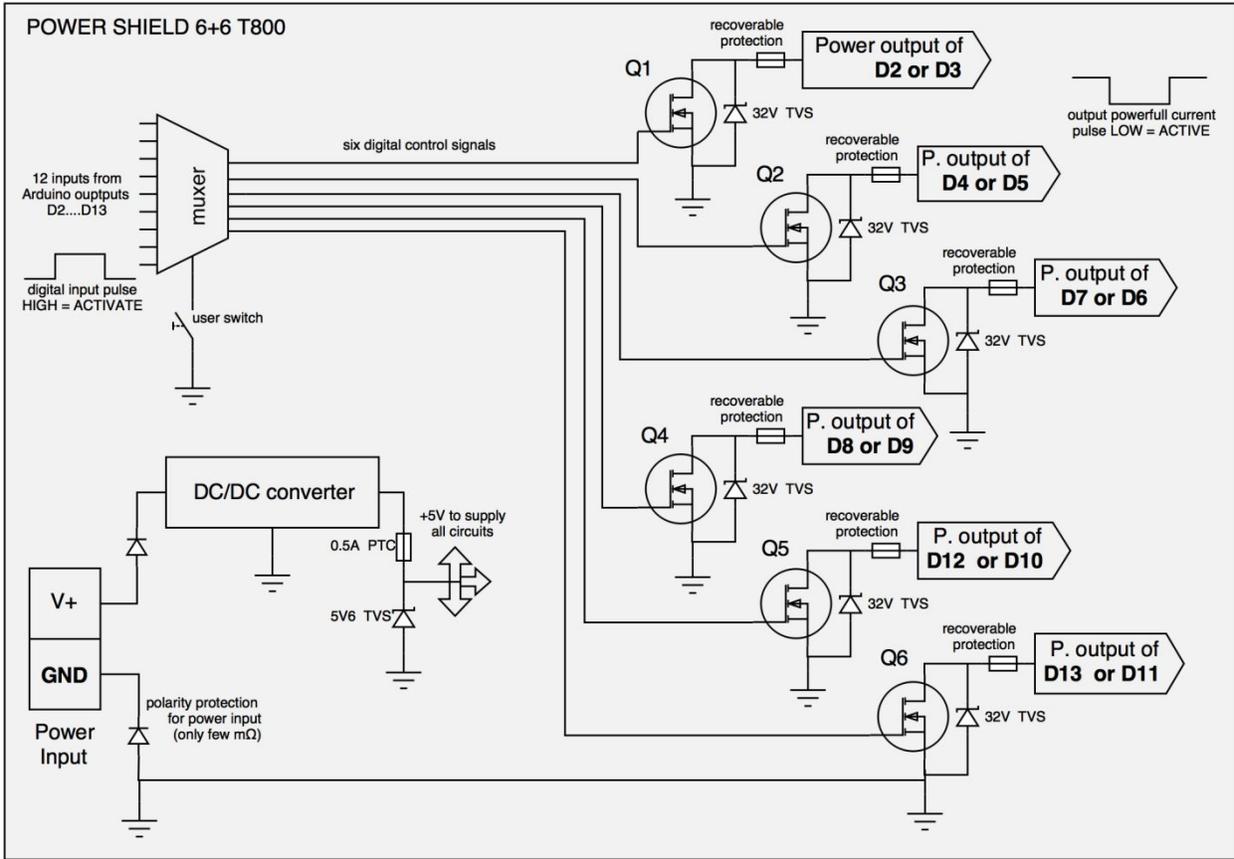
You can see the extended list of products including many other microcontrollers here:

https://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems

If you need to control more than 6 channels in PWM mode, below is a list of boards that we have checked and that have 12 PWM channels:

- ARDUINO MEGA (8-bit), DUE (8-bit), M0 (8-bit), M0 PRO (8-bit), TIAN (8-bit), PRIMO (12-bit), STAR OTO (16-bit)

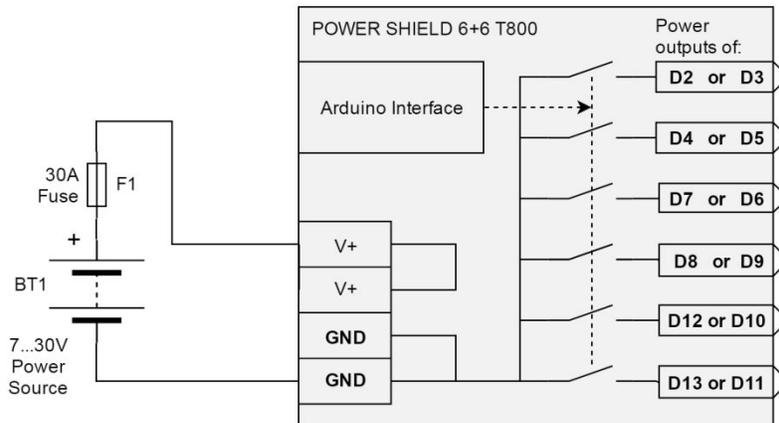
(Note: not all Arduino boards that have this feature have been fully checked by v-vTech.)



[Fig. 5 – Simplified internal structure]

Power supply

The board has a DC/DC (BUCK step-down high efficiency) converter that provides a stable +5V power supply to the PS logic circuits and powers the attached Arduino board and other possible expansion modules connected to the top. The converter (for external devices) can deliver a current of 450mA, though we recommend to use only up to 300mA.



[Fig. 6 – Very simplified internal structure with a power source connected]

Be careful also before plugging the power supply into the PS (outputs or input)! Make sure the source voltage does not exceed 32V (absolute maximum)! A lower voltage is better – we recommend 30V or less. If you exceed 32V, you can permanently damage the product.

Also, be careful when connecting polarity wires. The polarity protection will only trip when you connect the common (+) and V+ inputs to the same power source and load(s) (see Fig. 9). Unfortunately, polarity protection cannot protect the device if you do not connect anything to the V+ input. Protection will only work partially if you connect multiple voltage sources – in this case, the polarity protection will only work for the power source that is connected to V+. If you reverse polarities for those which are connected separately (for example, Fig. 11, closed loops through the low-side with inverted polarities), you will have the opposite current, which will burn the output drivers.

All nominal parameters specified in this document are determined by testing the board at room temperature, +15...+25°C.

Input/output voltages

The power supply voltage is 7...30V. The absolute minimum and maximum is 6...32V. Any PS can control very low voltages, e.g. even from 100mV. Therefore, evaluate the maximum output current according to the output resistance (see Fig. 7). The minimum input voltage is significant for the Step down converter to run, which can power the 5V PS circuit and the Arduino controller, as well as to determine the sufficient voltage for controlling the input current transistor Q1 gate. Therefore, if you work with voltage lower than +6V, you must power the board in a different way, i.e. from an Arduino board, USB, or Vin (according to the specific input voltage requirements of the Arduino board). In this case, it is best to leave both V+ terminals unconnected; otherwise, protection may be triggered and the outputs blocked.

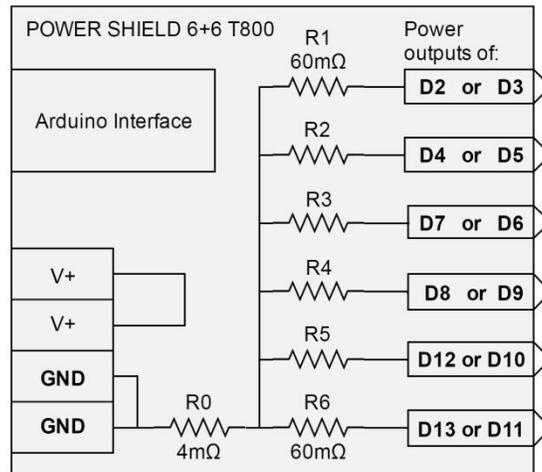
Output voltage can be different at each of the 6 channels because of the Low-Side driver's benefits!

Wide operational voltage range – outputs can control loads on 1...32V voltage supplies!

Current, limits and protection

- **Maximum DC input current protection** 25A [moderate reaction]
Response time at GND input current, i.e. sudden increase from ~1A to ~45A (or more) is ~250 ms.
- **Maximum input pulse current protection** ~280A [quick reaction]
Response time ~2ms. Warning: peak output power can reach 8kW!
- **Thermal direct current protection per channel** 6...8A [slow reaction]
The higher the temperature of a specific output driver, the faster the reaction. 1...100s.
- **Current limitation via channel (when warning indicator is off)** ~40A [quick reaction]
Response time about ~1ms.
- **Current limitation for the channel (when warning indicator is on)** ~15 A [quick reaction]
Response time about ~1ms.

Impedance



[Fig. 7 – Power circuit impedance]

These imaginary impedances $R_0 \dots R_6$ were deduced from the following operating conditions: input voltage 24V, current 23A (100% of duty cycle). In other words, the voltage drop between the input GND and the output terminals was measured and the impedance was calculated:

$$R_0 = \sim 3.61\text{m}\Omega, \quad R_1 \dots R_6 = \sim 56.7\text{m}\Omega.$$

Protection

- Input polarity protection. By reversing the potential between V+ and GND, the protection completely disconnects the GND, including from the outputs (since the device is Low-Side type). This ensures that no negative current is supplied to the outputs.
- Fifth-level current protection (for more details, see [Current, limits and protection](#)).
- Logic circuits 5V overvoltage protection with 5V6 TVS.
- Logic circuits overcurrent protection with 0.5A PPTC at +85°C (at higher temperatures, overcurrent protection may be activated at a lower current).
- V+ input ESD (and short overvoltage peaks) protection with bipolar 33V TVS component.
- All signal inputs and outputs have ESD HBM JESD22-A114F protection up to 2000V.

FeedBacks

LEDs on board

- **Green** – Power okay; +5V logic voltage supply is present (higher than 4V)
- **Blue** – Overcurrent indicator; flashes when (pulsed or constant) input current is higher than about 20A

- **Yellow** – Overtemperature *warning*; one or more of the 6 channels have reached critical temperature due to being short-circuited or overloaded. PS will continue working but with lower peak [current limits](#) per channel. Usually, the current will remain limited until you shut off the power or pull the reset pin.
- **Red** – ERROR when protection is on for a few seconds (PS is rebooting)

Outputs to analogue inputs

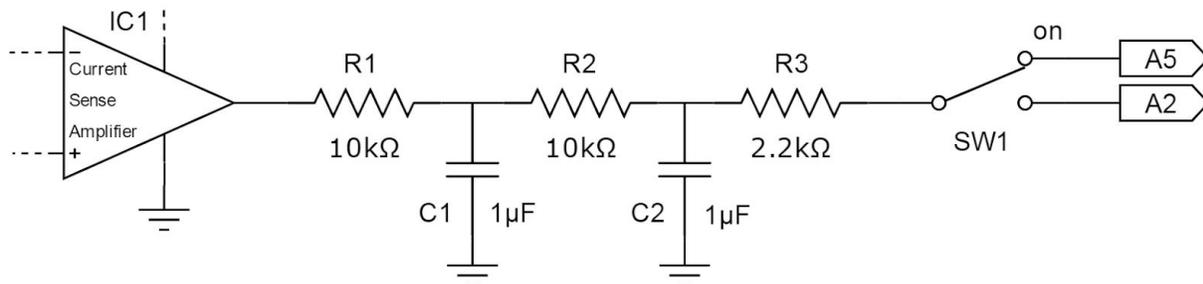
- **Analogue** current expression – 1V <==> 10A; 1V <==> 20A when stacked at input merge setting

- Digital (usually latched) **warning** (High out) output – warns about critical temperature or high current

- Digital **protection on** signal (high when active)

You can turn off the FeedBack with the second switch (lowering it to the OFF position), and then the analogue pins will remain unconnected. On the other hand, if the FeedBacks are turned on, you can still ignore them and send their signals to any analogue inputs because the PS sends all FeedBack signals only through 22k resistors; therefore, the PS outputs are protected from short circuits to either GND or to + 5V.

The analogue current measurement signal is outputted via the dual RC filter to the A5 or A2 inputs:



[Fig. 8 – PS internal circuit: output filter for current sense amplifier]

To get FeedBacks to the Arduino analogue in, first turn SW2 to the ON position. Next, with the channel group switch SW1, select the analogue channels for feeding the FeedBacks.

If SW1 is ON, FeedBacks are outputted to:

A5 – Analogue current measure signal

A4 – Warning signal

A3 – Protection on signal

If SW1 is off, FeedBacks are outputted to:

A2 – Analogue current measure signal

A1 – Warning signal

A0 – Protection on signal

Frequency

Absolute maximum output frequency with current up to 0.5A output per channel	- 100kHz
Maximum frequency at full power using additional heat sinks and/or fan	- 25kHz
Maximum safe output frequency at full power (without active cooling)	- 4kHz
Recommended output frequency for long-term work up to	- 2kHz

Duty cycle vs. frequency

Possible duty cycle ranges for pulse-width modulation:

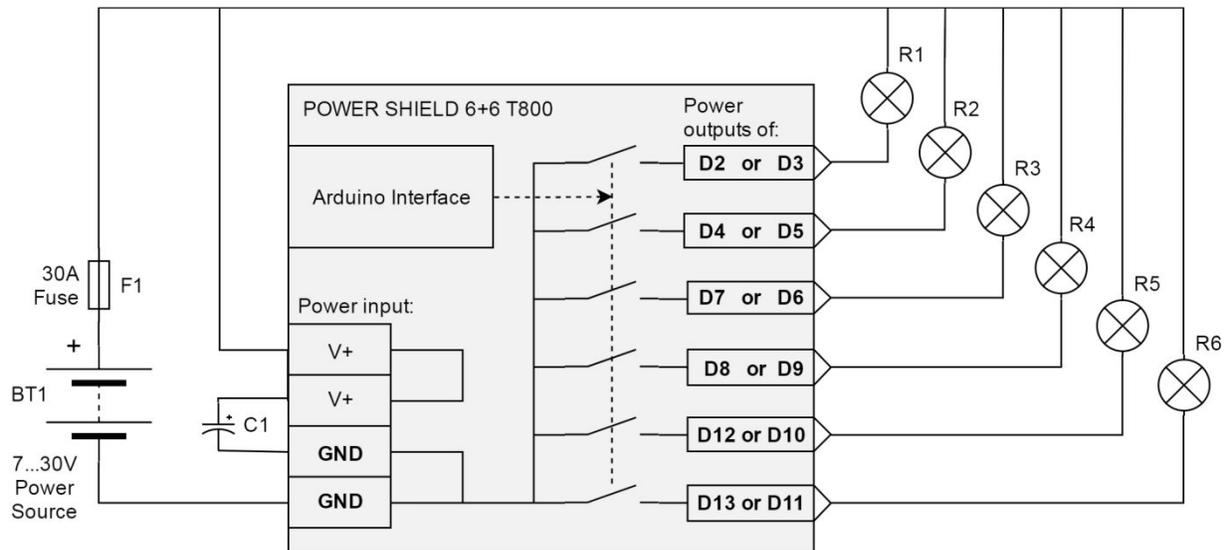
- 1...98% @ 0...2kHz
- 6...90% @ 2...10kHz
- 10...80% @ 10...20kHz
- 25...55% @ 20...50kHz
- only ~45% @ 50..100kHz

Note that after reaching the duty cycle limits, the outputs will always be at 0% or 100%. For example, in the case of 50 kHz, the 0%...~ 24% outputs will be completely desaturated, and ~56%...100% will be fully saturated. This is because these limits are no longer transmitting the frequency but only completely deactivate or fully activate the outputs.

The rising and falling fronts have a duration of ~ 2 μ s. The front times are measured at 24V and 100 W halogen bulb loads at 90% duty cycle and 4.4kHz operating frequency.

How to connect

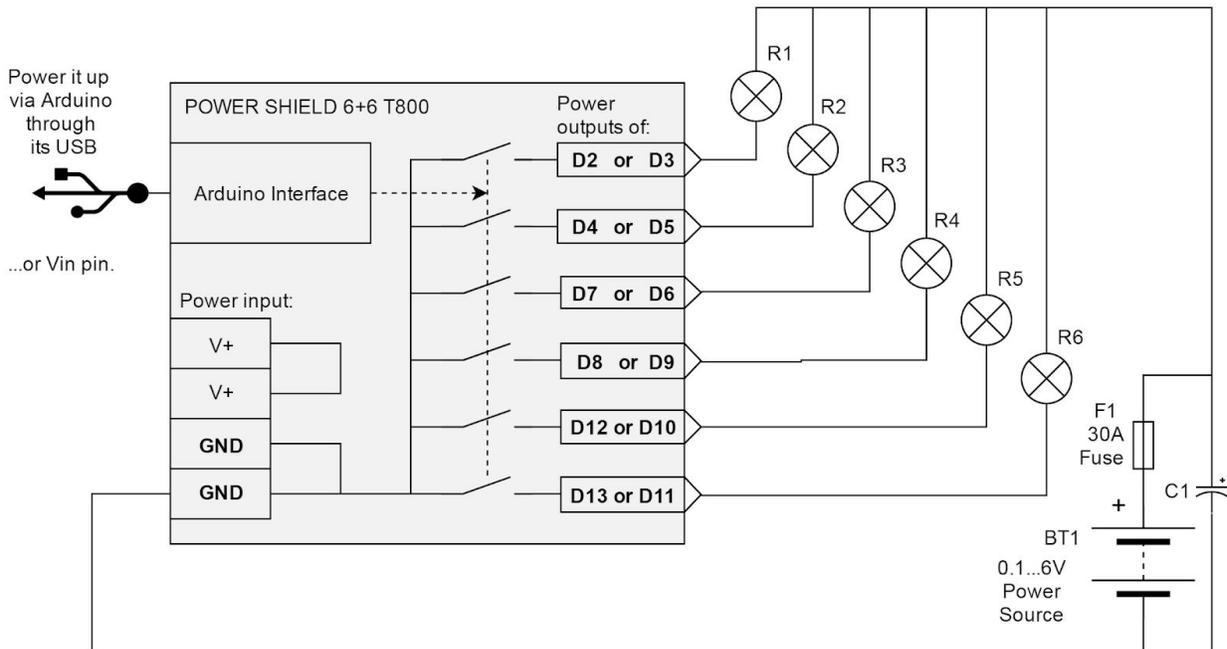
Connecting the power supply



[Fig. 9 – The simplest connection diagram for utilisation of all channels]

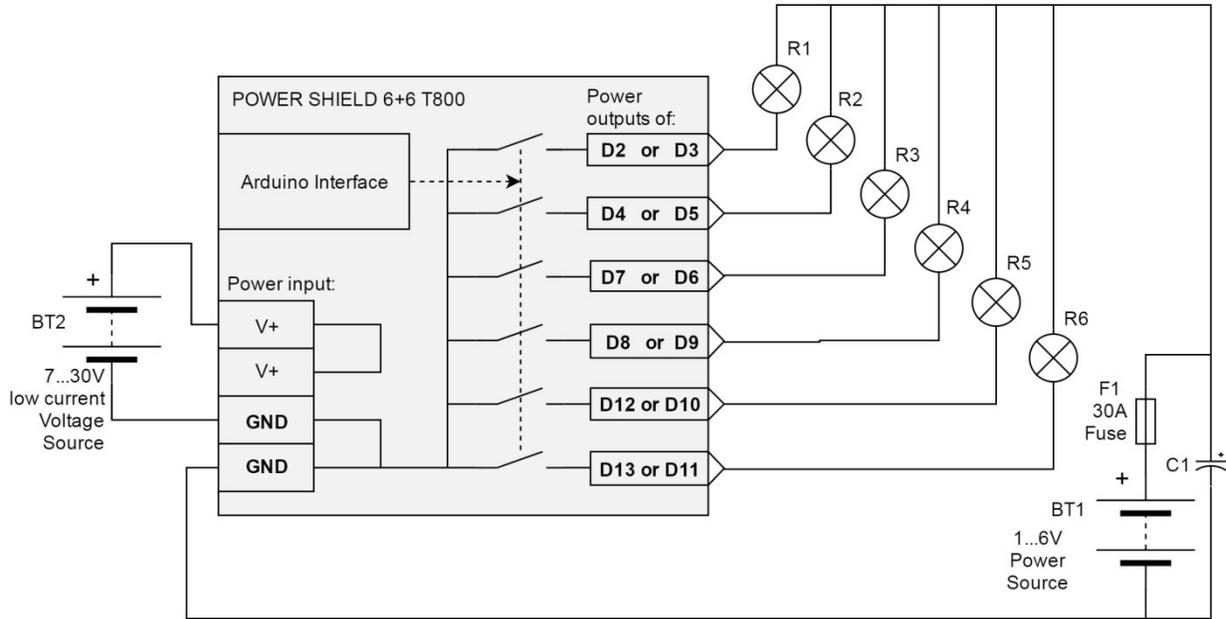
In parallel to the power supply, we recommend that you install the electrolytic C1 capacitor – it is better to use larger capacities, respectively, according to the loads. Let's say that you want to work with full power. In that case, it would be best to use 10'000 μ F / 50V. The rated voltage of the capacitor should be higher than the power supply by +20....30%. For example, if you work with 12V, the capacitor voltage should be 16V or higher. We also recommend using an F1 fuse. Although we have tried to put as many protections as possible, still a fuse or mechanical fuse is the most reliable protection. You can also select the current strength of the fuse depending on the range of the maximum current you will be using. Let's say that you know you use up to 10A. In that case, use a 15...20A fuse, but no more than 30A per one PS board.

Low voltage



[Fig. 10 – PS logical circuits powered by Arduino USB]

Here is an example of how the V+ input voltage is not sufficient for the stepdown converter in the PS (DC/DC) and polarity protection when powered with a supply voltage lower than 6V; in this case, you must leave V+ unconnected. Here, the product will be powered from the Arduino board's 5V pin. Hopefully you know how to power the Arduino, for example via USB. Another option is to connect a separate voltage source to V+ on the PS. The example below shows two power supplies, where BT2 is dedicated to powering the PS (and Arduino) boards:



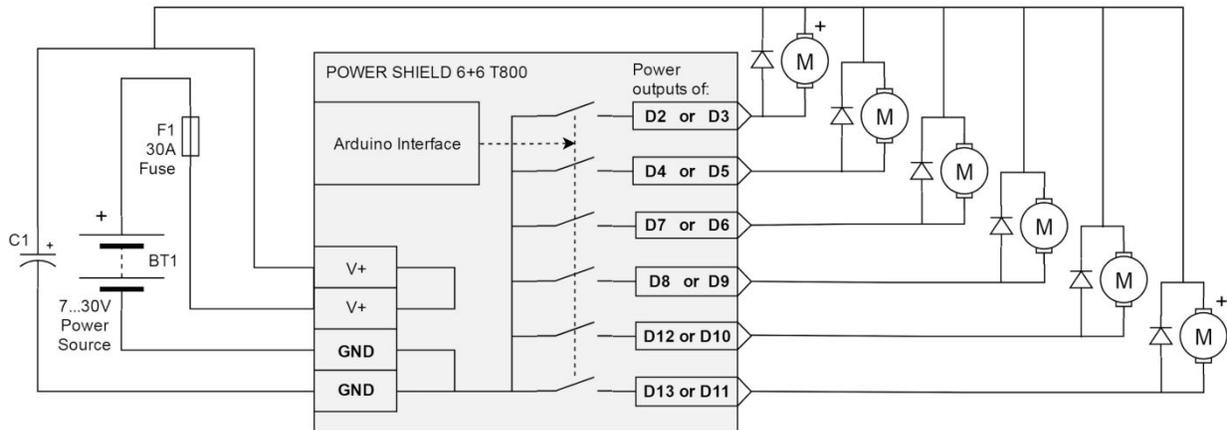
[Fig. 11 – Power supply of the logic circuits from a separate source]

Here, the BT1 voltage source can range from 0.01V to 32V as it does not power the board.

Connecting motors

This board can be used to control brushed DC motors or UniPolar Stepper motors (which have 5, 6 or 8 wires).

Connecting DC motors



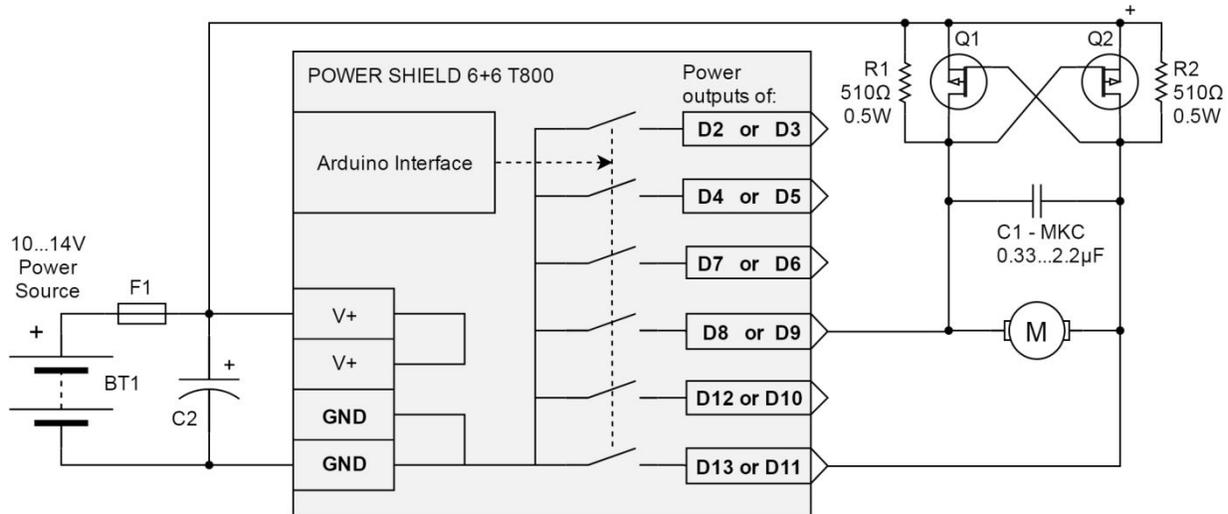
[Fig. 12 – DC motor wiring diagram]

When connecting DC motors, we definitely recommend adding high-speed diodes for FlyBack currents, for example Schottky diodes 1N5822 or 1N5819. This will prevent damage to the PS and unnecessary tripping of protections. The maximum load for the diode will be created by a

heavy loaded powerful motor controlled by the PWM signal. The DC motor polarity can be freely reversed; doing so will only impact the direction of rotation.

DC motor direction of rotation

Although the PS is used to control DC motors in one direction of rotation, by adding 2 extra transistors you can get an H-Bridge which allows you to control the motor in both directions.



[Fig. 13 – H-Bridge]

Some advice:

Use a P-Channel MOSFET transistor with a much higher U_{gs} voltage. It is best to have U_{gs} voltages of +/-20V or higher.

We have tested this circuit with **IRF5305** – Q1 and Q2. The circuit began to work with a C1 – MKC capacitor connected to motor terminals and resistors R1 and R2 with 510Ω resistance. Do not forget to connect the C1 – MKC capacitor; it is very important!

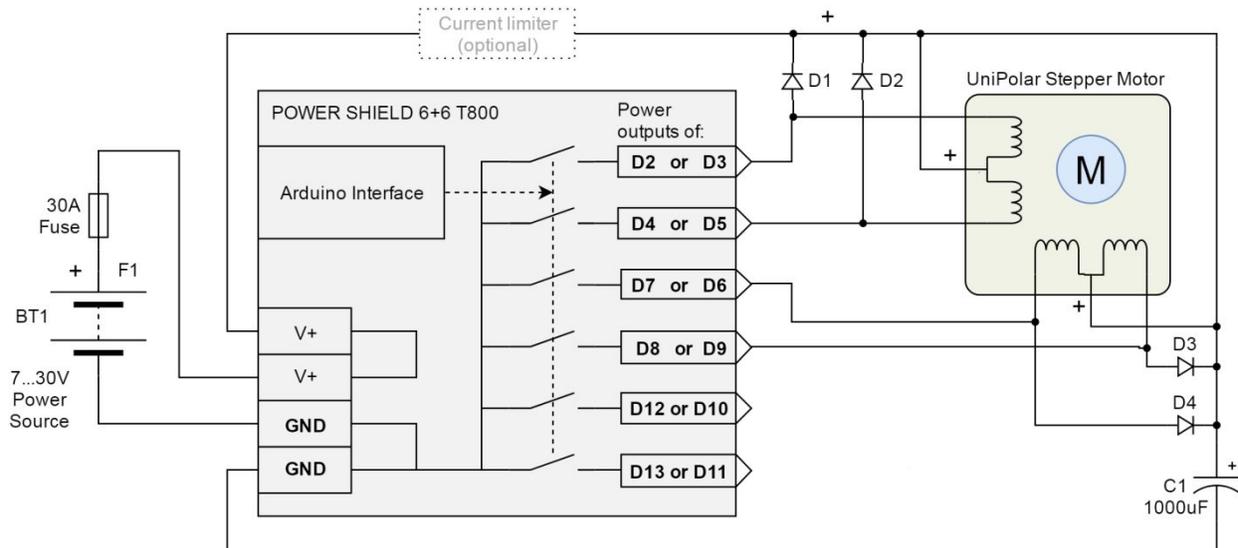
Control your motors only with default PWM frequencies (0...2kHz).

Make sure that your power supply voltage does not drop below 10V because this is very important for gate inputs for MOSFETs.

Do not open both channels at the same time; if you do, you will get a short circuit and burn your MOSFETs.

If Q1 and Q2 transistors heat up significantly, attach them to heatsinks.

Stepper motor (UniPolar only)



[Fig. 14 – UniPolar stepper motor]

- With this device, you can only control UniPolar type Stepper motors (which have 5, 6 or 8 connection wires). The circuit diagram gives an example of how to connect a 6-wire stepper motor. Note: If you want to control a motor rated less than 7V (for example, 5V), connect it to a separate power supply.
- If you want a more stable motor step and better performance, we recommend installing a current limiter – the **Buck Step Down Constant Current Regulator series** work best. Select the current limitation depending on the motor (from 0.1A to 5A).
- The current limiter will greatly improve the performance of the motor when it needs to be rotated slowly and acts as a brake mechanism (with direct current), thanks to which the motor will heat up much less! We recommend 1000 μ F capacity for the C1. Alternatively, for testing, instead of the current limiter you can use an incandescent lamp (halogen) or a separate power supply that has an integrated current limiting capability.
- You can stack two PS boards and simultaneously control as many as three UniPolar stepper motors; one Arduino microcontroller will be sufficient for this. And if you want to limit the current here, be aware that you have to limit it for each motor separately with a separate limiter.
- You can use UF4004 diodes from this series or similar.

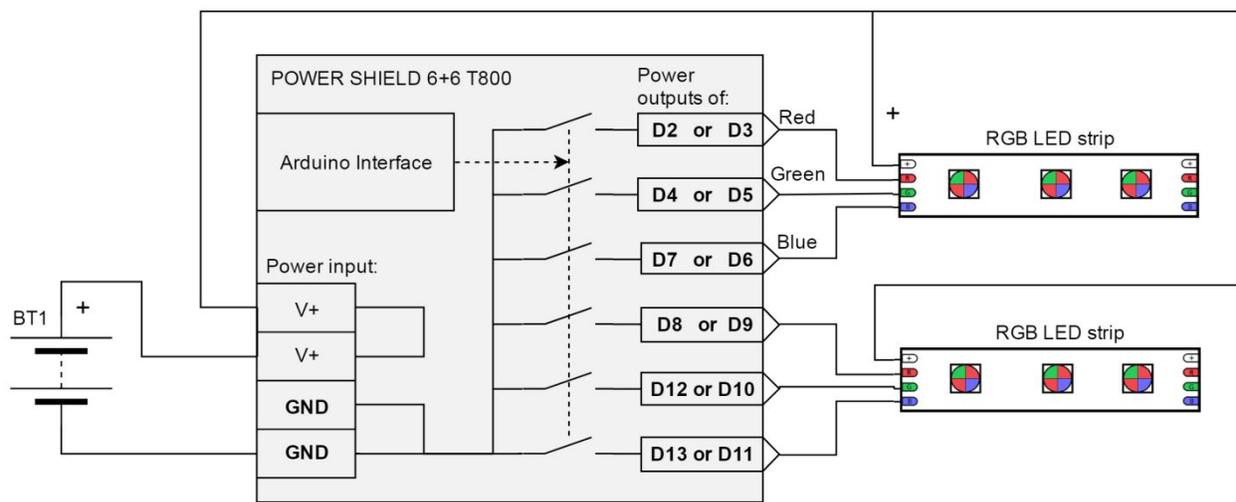
Connecting halogens

We recommend using 24V halogens for up to 70W per channel or 12V for up to 35W per channel. Of course, it is possible to connect more powerful elements, but then it will be difficult to start them because the halogen lamps have non-linear volt-ampere characteristics: the

resistance of the cold filament is ~14 times less than that of the illuminating, and therefore the start current is automatically ~14 times higher.

Using halogen lamps with the lowest duty cycles and at the highest frequencies (more than 2kHz) can generate (due to inductance) narrow but very strong back current peaks that can trigger the FlashBack detector and activate the protection, which means switching off the outputs. Therefore, we recommend using additional FlyBack diodes in a similar way as for controlling motors (see Fig. 12). After this, everything is the same; just connect halogens instead of motors.

Connecting LED strips



[Fig. 15 – LED RGB Strip]

With one PS, you can control up to 2 RGB LED strips or 6 monochrome LED strips.
With two PS boards, these numbers are doubled.

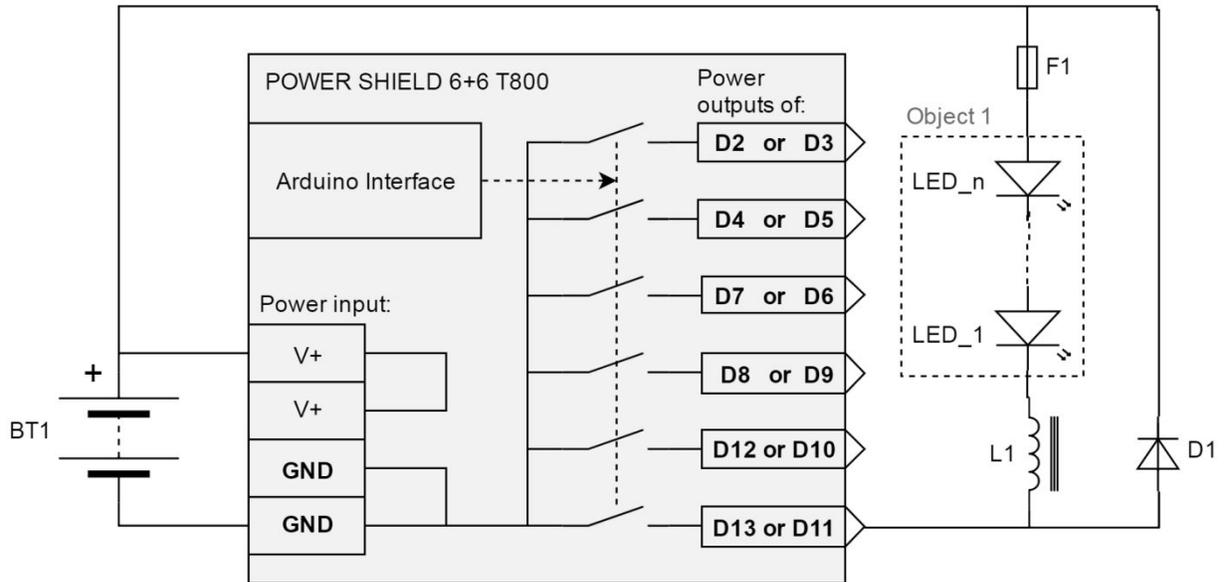
You can connect up to 600W of 24V bands to one board.

You can connect up to 300W of 12V bands to one board.

For example, if the 24V band has a power of 14.4W/m, its maximum length will be 40m.

High efficiency (LED) driving (with step-down function)

We offer this principle of diode control only to those who already have experience in electronics and have sufficient knowledge of the C language when programming Arduino microcontrollers because for this you will need to write a FeedBack program to select the correct PWM signal to power the Object 1 (LEDs) with a direct current. Be sure to use the F1 fuse because selecting the wrong PWM signal can easily create overcurrent in Object 1.



[Fig. 16 – Inductor for driving LEDs]

The circuit is very useful when you want to control 1W and more powerful (Object 1 voltage = ~2...25V) LEDs with minimal losses (which could be caused by traditional LED current limitation via a resistor).

To use this method, you will need to set the default Arduino PWM pin frequency to 8kHz instead of the standard 0.5kHz. Otherwise, if you use a lower default frequency (0.5...1kHz), you will need a very high inductance (~15mH) coil. With an 8kHz coil, the inductance of 500µH...2mH will be sufficient (see Fig. 19).

The inductive coil works similarly to a transformer, so the current through Object 1 will be almost as many times higher as the voltage (drop) on Object 1 is lower. For example, if the input has $U_{BT1} = 15V$ and $I_{output\ average} = 1A$, then $U_{Object\ 1} = 5V$ and $I_{Object\ 1} = up\ to\ 3A$.

The current of Object 1 can be approximately calculated as follows:

$$I_{Object\ 1} = \frac{U_{BT1} \cdot I_{output\ average}}{U_{Object1}} \eta = \frac{15V \cdot 1A}{5V} 0.9 = 2.7A$$

Quiescent η is usually $\approx \approx 90\%$. It depends on many factors.

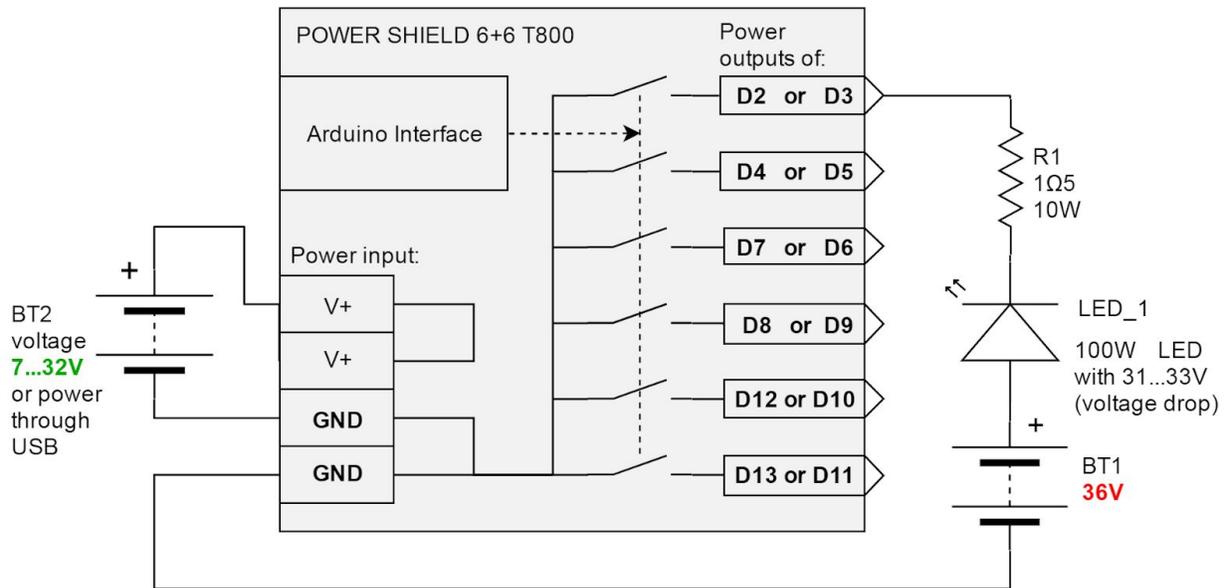
The transformation ratio K in this case will be: $K = \frac{U_{BT1}}{U_{Object1}} = 3$

To determine the duty cycle limits for controlling a particular object, you can use the current meter integrated for current measurement (which outputs a signal to analogue pins).

Once you've set the limits for one channel, you can continue to apply them to other channels with the same components connected to them.

This way you can power a variety of objects, for example Peltier elements, heating elements, or hydrogen trays, or even charge any battery. Also, the circuit will work more efficiently.

Driving 100W 35V LEDs easily



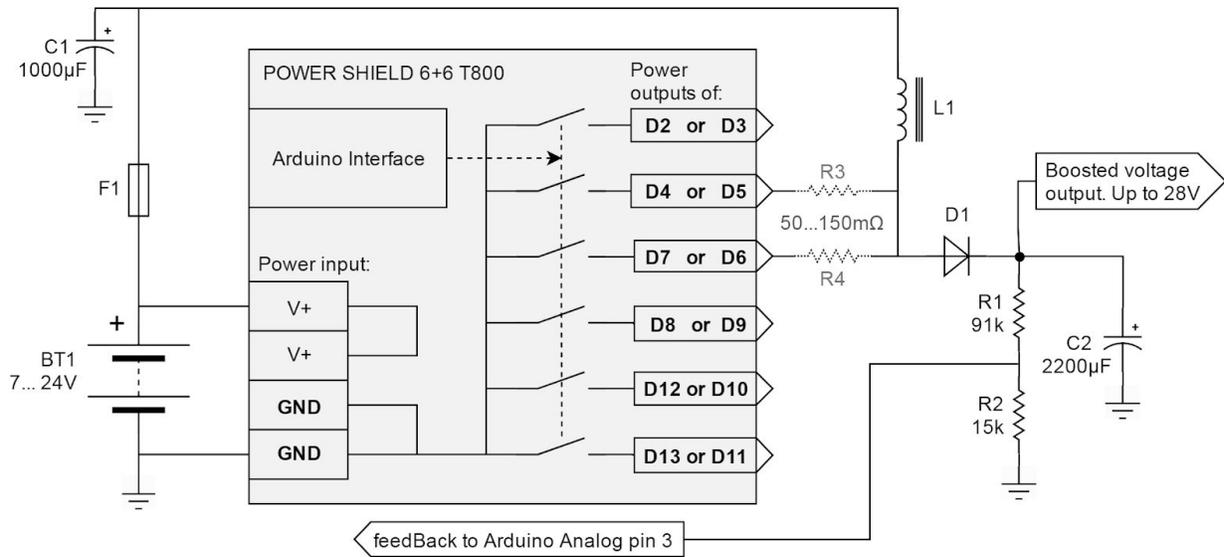
[Fig. 17 – 100W LED]

Here is an example of how you can exclusively control **31–33V 100W** LEDs with a 36V power supply. Since the voltage used exceeds the rated voltage of the PS, the following requirements must be met:

- Use resistor R1 with minimum parasitic inductance.
- Important: do not power the board with a 36V (BT1) power supply! It is used only for the LED circuit(s) described here.
- Use a separate power supply for powering the board logic (see Low voltage).
- For long-term use, calculate R1 according to the specific LED specifications and power supply. For example, in our case, a voltage of 4 V dropped across the resistor and we selected a resistance of 1.5Ω. This way we limited the maximum current to 2.7A.
- Be careful with frequency increases (overcharging). Above 2kHz, the operation of the device may be influenced by parasitic inductance. For heavy loads, we recommend using only the default frequency.
- **Only 30...35V LEDs are suitable for this circuit diagram. For other cases, over-voltage cannot be used at this point!** The 30V LED is an exception because the difference in power voltage drop on it is taken off from the BT1; therefore, this voltage in this case will not damage the PS outputs.

In this way, you can control (adjust power or turn ON and OFF) 6 independent 100W LEDs using all channels. In total, you will have up to 600W LED power, or you can connect LEDs in sequence and use the same power across all channels.

Voltage booster (step-up converter)



[Fig. 18 – Step up converter – voltage multiplier]

The circuit diagram is for advanced users of electronics and those experienced with programming Arduino microcontrollers. It can increase voltage from 7V to 28V with ~90% efficiency.

For this circuit diagram, you will need to:

1. Configure the board PWM frequency as high as possible (we recommend 8kHz)
2. Choose a high-quality, high-current toroid inductor (~470µH...2.2mH & <0.3Ω or similar component)

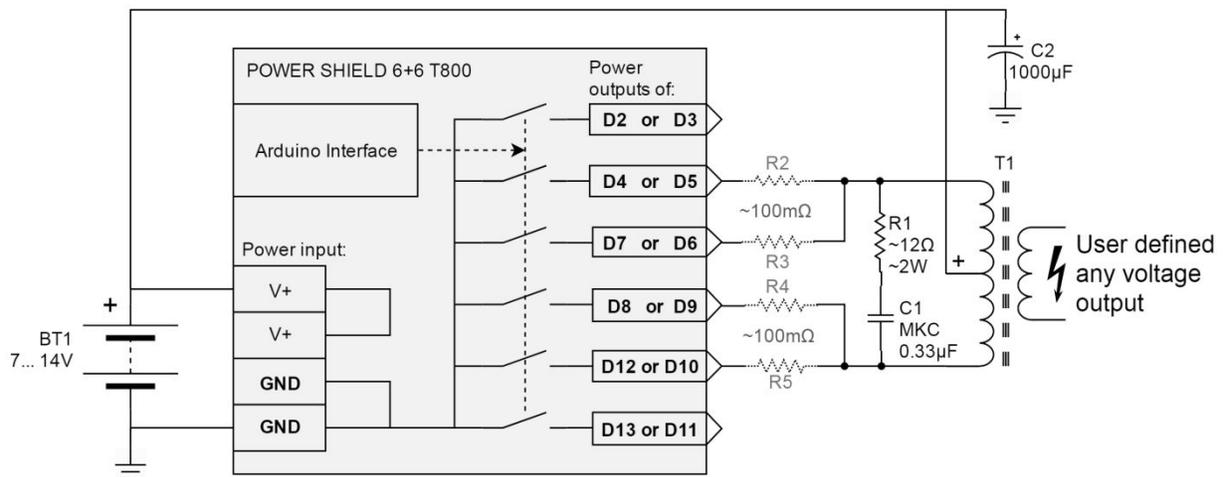


[Fig. 19 – Toroid inductor]

3. Program the microcontroller so that the circuit automatically adjusts the duty cycle, inversely proportional to the returning FeedBack voltage. Make sure to never reach more than 80% of the duty cycle! Otherwise, you will short-circuit the coil. Also, your code must disconnect the output(s) if the voltage exceeds 28V.
4. The feedback in the best case should be supplied to the following Arduino PINs:
5. ***A3** or A4 when SW1 is up
6. ***A0** or A1 when SW1 is down

7. If you use C1 or especially if you use the C2 capacitor, use as large of a capacity as possible. Increasing C2 will reduce current instability, and the FeedBack will automatically measure the voltage more accurately.
8. D1 Schottky diode could be, for example, 1N5822.
9. If you want to control higher currents, you can connect the PS outputs in parallel. We recommend separating them with the R3 and R4 resistors. This will allow you to distribute the output current more evenly.
10. Also, if you parallel the outputs (through the R3 and R4 resistors), before doing so, make sure that your chosen digital outputs work synchronously (at the same frequency and coincide with the duty cycle phase).

Connecting the transformer



[Fig. 20 –Transformer with centre-tap application]

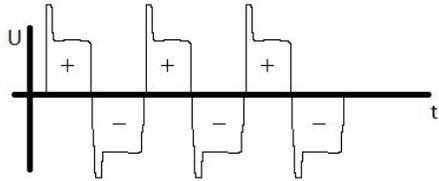
With this circuit diagram, you can generate almost any secondary voltage from ~1V to ~5000V (attention – potentially high voltage – be careful and only use if you have appropriate education). This circuit diagram is for advanced users only because you need to select the correct transformer and properly program the outputs.

First of all, it is very important to configure the outputs so they can open alternatively: respectively, D5 with D6 for upper and D9 with D10 for lower, primary winding poles. It is also very important that the duty cycle of each signal is not greater than 48% if you work at frequencies up to 1kHz and 42% if you work at frequencies up to 10kHz.

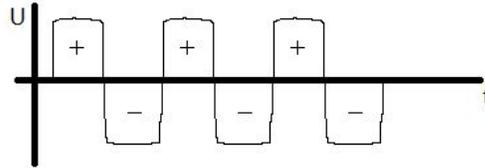
If you want to build a high frequency switching supply, use a ferrite core transformer with about 20 turns in primary winding (2x10W). Set the frequency at about 8kHz.

If you work with a 50/60Hz (mains frequency) transformer and, for example, you want to make a 110...120V or 220...240V inverter (attention – potentially high voltage – be careful, only use if you have appropriate skills), you can simply use any standard electric steel transformer.

It is very important to choose R1 and C1 correctly – they are intended to reduce the switching peaks. The values of the passive components presented here have been tested with a ferrite transformer. If you do not suppress the peaks, they will trigger FlashBack protection and your device will shut down. The diagrams below show signal forms between the side edges of the primary winding with and without peak filtering (with and without RC suppressor):

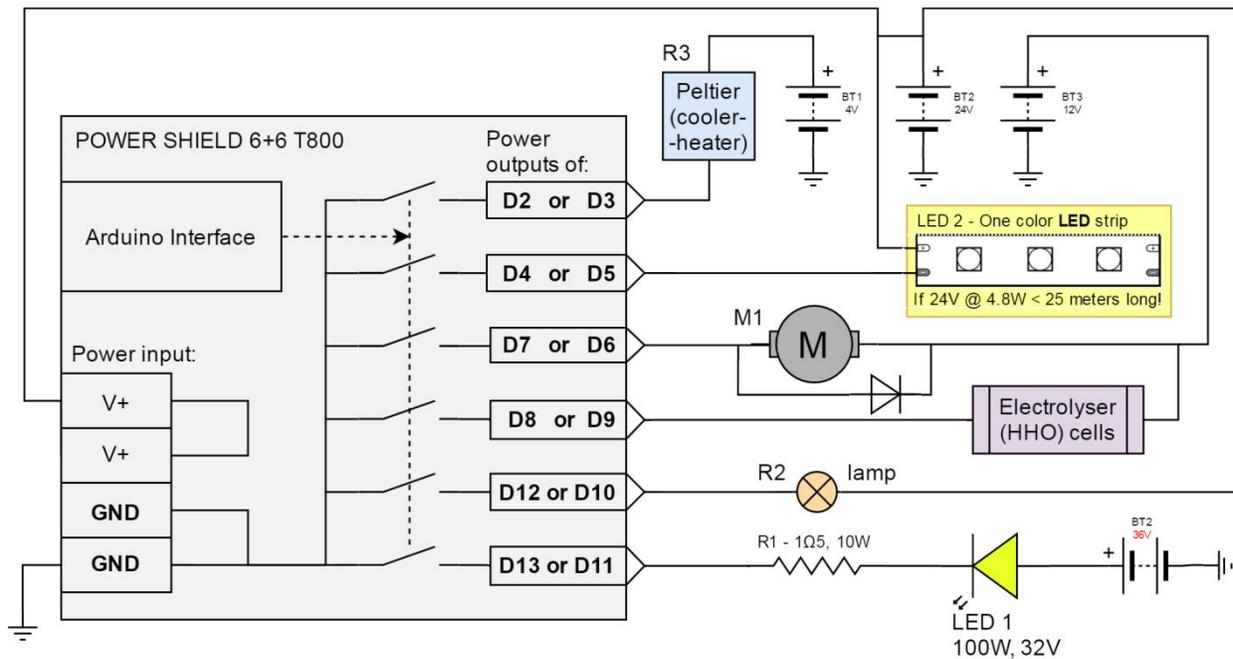


[Fig 21.1 – Transformer signal without RC filter]



[Fig 21.2 – Transformer signal with RC filter]

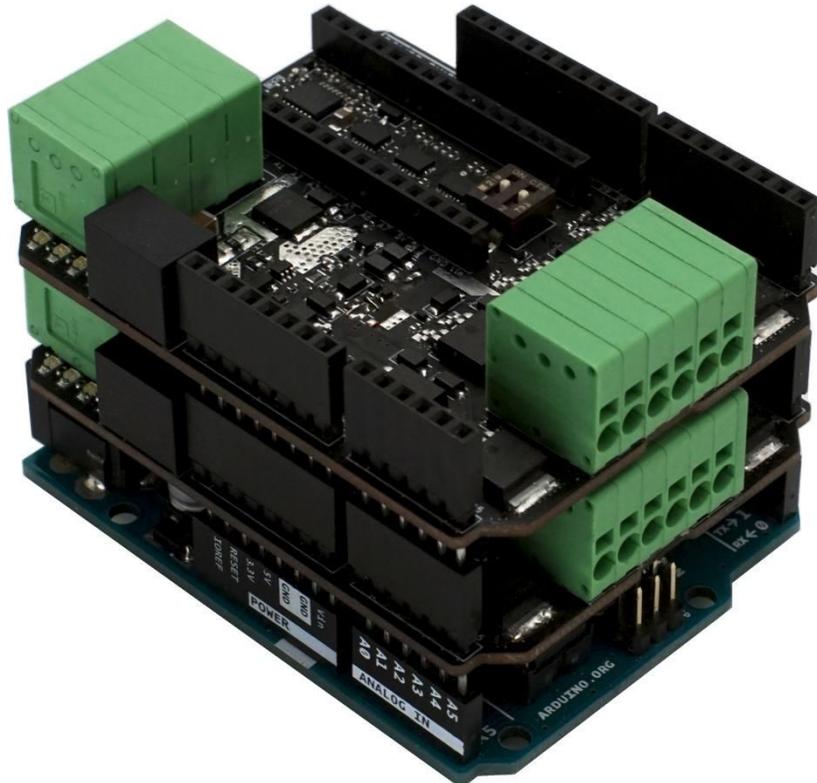
Mixed connection application



[Fig. 22 –Various loads connect]

This circuit diagram demonstrates the flexibility and versatility of PS. You can connect both different types of loads as well as different voltage supply sources.

Stacking – double power – double outputs

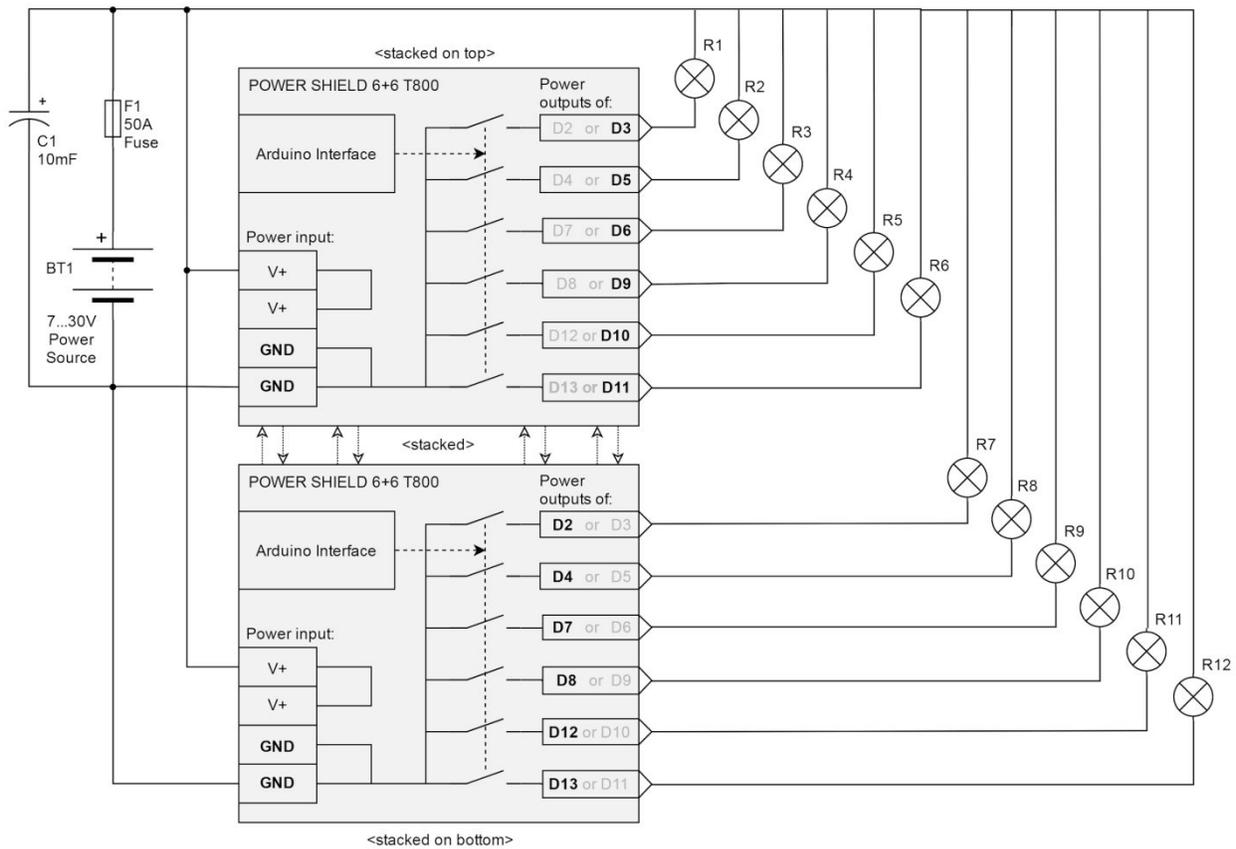


[Fig. 23 – Arduino UNO + two stacked PSs to use 12 channels to get output power up to 1.5kW]

By stacking, you can expand output channels from 6 to 12 and also have FeedBacks from both stacked boards. You can select which Arduino ports to use – PWM or non-PWM depending on the Arduino board.

Only one Arduino controller is required to control both PSs, in any case. For this use case, for each PS board you need to set different SW1 switch positions and then you will have 12 independent (separately controlled) channels. If both SW1s are in identical positions, the outputs will be duplicated on both expansion boards; in other words, 6 Arduino digital outputs will control 12 outputs (each Arduino output will control 2 PS outputs). Remember that SW1 also switches feedbacks to analogue inputs.

Multiplying PS outputs



[Fig. 24 – Example of two stacked PSs]

We recommend installing the board that will work with PWM signals on top because it will get hotter. This way it will be easier to dissipate the heat. If both boards will work in PWM mode (for example, using the Arduino MEGA microcontroller because it has 15 PWM channels) with the maximum load applied, then use active cooling (i.e. a high-flow server fan). The best ventilation direction is from right to left.

Merging PS outputs

If you want to control higher currents, you can easily do this by connecting outputs in parallel. This way you will have 6 higher current outputs and the current will be about 1.8 times higher. When connecting outputs in parallel, do not forget to set SW1 on both boards to the same position. Also, to make sure that the currents between the merged outputs are completely the same, use 50...100mΩ resistors.

Demo programs for Arduino

Download and test sketches

You can download [two programs \(sketches\)](#) and upload them to your Arduino microcontroller. One of them is simplified program for beginners and other one is for more advanced or professional users.

Once you have uploaded one of these programs and connected the Arduino to the PS board(s), connect something to the outputs to test them. For testing, we recommend connecting the bulbs as shown in Fig. 24 ([Stack PS to multiply outputs to 12](#)) or 6 simple LED strips. On one board, switch SW1 to ON and on the other board switch SW1 to OFF.

This sketch controls all main outputs from 2 to 13. Duty cycle signals are output on PWM outputs (3, 5, 6, 9, 10, 11), while the switching signals are output on the remaining outputs (2, 4, 7, 8, 12, 13).

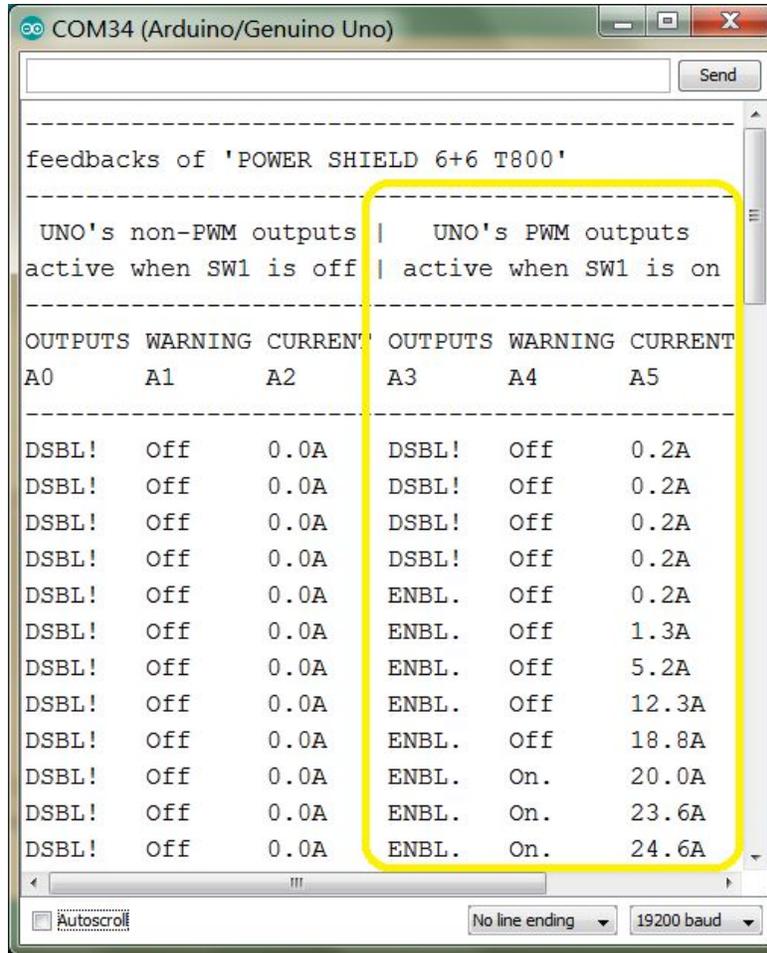
Therefore, if you are using only one board, switch SW1 depending on which outputs you want – PWMs or simple. For example, on Arduino MEGA, all outputs D2...D12 are PWM. Therefore, you can program the outputs so that PWM operates even when SW1 is off.

Analog inputs (A0...A5) are used here for FeedBack reception from PS.

If SW2 is ON, the shield returns three types of FeedBacks:

- **A0 or A2 – Error; all outputs disabled for a few seconds (red blinking LED)**
- **A1 or A4 – Warning (yellow LED)**
- **A2 or A5 – Analogue output voltage of 1V equivalent to 10A (20A for stacked version when both SW1 switches set into same position) of input current (blue LED)**

All of these inputs can be monitored by turning on the Serial Monitor (with a baud rate of 19200). The return of FeedBack signals (to A0, A1, A2 or A3, A4, A5) will depend on the position of the SW1 switch (outputs to analogue inputs).

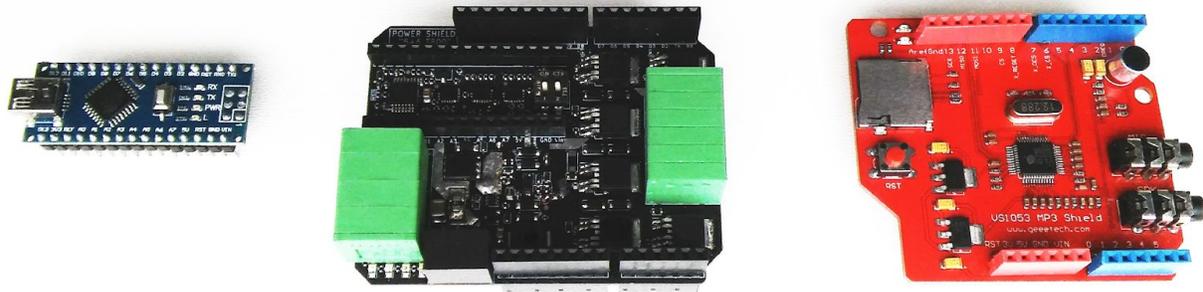


[Fig. 25 – FeedBack outputs to Arduino serial monitor form one PS]

In the case of stacking, when the SW1 of one board is ON and the other is OFF, this window will display the parameters on the left and right.

All 6 FeedBacks will run simultaneously, and 2 ammeters will work.

Use PS as Nano expansion board



[Fig. 26.1 – POWER SHIELD + Arduino NANO + some other shield]

Because PS is compatible with almost all Arduino-type boards, it can also be used as an expansion adapter from Arduino NANO (or MICRO*) to the Arduino UNO platform for connecting it to other extension modules. Use the low-profile male pin headers to connect Arduino NANO.



[Fig. 26.2 – POWER SHIELD + Arduino NANO + another shield]

Since the PS is supported here, you can arrange for better cooling at higher loads; put a fan to blow along the board or place the device in a well-ventilated location.

General tips and warnings

- Only operate with a safe constant voltage up to +32V (absolute maximum).
- For protection against short circuits, always connect the power supply via a 10...30A fuse (~25% higher than the maximum load).
- For the V+ and GND power supply input for one board you can use 2.5mm² cross-section monolithic (or soldered) wires or 1.5mm² with ferrules.
- For outputs, use 1 monolithic wire or soldered wires or 0.75mm² with ferrules.
- Do not put (balancing) resistors directly in the inputs. Resistors must be separated by wires at least 10cm long.
- Connect the power supply only when you are sure that everything is connected correctly.

- The maximum possible operating temperatures are -20°C ... $+50^{\circ}\text{C}$, but to reach the maximum power performance (without additional cooling), use the board only within the temperature range 0 ... $+30^{\circ}\text{C}$.
- If the board is placed in a closed housing, take care to ensure good air circulation (from the top, bottom and sides) to ensure good cooling of the board. Of course, if you use the board with only low load (total current up to 15A) and without PWM, the board should not overheat.
- To improve cooling, you can glue a thin heatsink to the bottom of the board (especially in the zone under the outputs) through a silicone gasket.
- Use coloured wires to make it easier to distinguish your connections: **red V+ for the input**; black for GND input and six power outputs.
- Use the device only in a dry place.
- If you connect the board not to Arduino microcontrollers but to controllers of other manufacturers (or any generators or similar), namely in a non-standard way, ensure that the input signal amplitude does not exceed that of the power supply (+5V). Otherwise, if necessary, protection cannot block input signals. This problem will never occur if you connect the PS as intended.

Troubleshooting

The **red LED blinks** and the **green LED is ON continuously** in normal operating mode. After connecting the power supply, it will blink for several seconds. This means that the board is starting up. If blinking occurs again:

- A. Check that you have correctly connected the FlyBack diodes (see Fig. 11 in the DC motors section)
- B. Check that everything is connected correctly – to check that your circuit has no errors, you can compare it with the circuit diagram in the description.

Flashing red LED and green LED is OFF (turns ON only occasionally and very briefly):

- A. Check if your supply voltage (and power) is sufficient, as the green LED turns OFF when the input voltage drops (albeit very briefly).
- B. Check that you have sufficient capacity of the capacitor on the PS power input.

Yellow LED and green LED are ON – yellow turns ON (and usually is latched) due to:

- A. Overcurrent on one or more channels.
- B. Overheating of one or more output transistors.

No signs of operation, no LED is ON:

- A. You have reversed the polarity
- B. No voltage, or voltage from power supply is too low

You activated the outputs and then the **blue LED blinked** while the **yellow warning LED** went ON along with it:

- A. The load connected is probably too high or the output has short-circuited.

Yellow warning LED is on – you can reset this status in three ways:

- A. By pushing the reset button on the Arduino
- B. By supplying very low (0.0V) digital signal to any of the reset pins
- C. By turning off the power and waiting until all the LEDs go out, then turning the power on again

Best wishes!

We wish you divine creativity to realize your great ideas!

We hope that this PS will allow you to implement your projects.

Contacts

Product authors:
Vytautas Janušonis
Valdas Mikėnas

All rights reserved ® and Copyright ©

www.v-vTech.com