# Table of Contents

## Overview 3

## Parts 4
- Powerstrip
- Housing, connectors and standoffs
- Electronics to control powerstrip

## Circuit Construction 6
- Circuit Details
- Prepare Connectors
- Use PermProto HAT to Mount ULN2803A

## Building the Housing 12

## Configure & Test 14
- Declare Outlets
Welcome to the second guide in reef-pi series.

See the first part of this series here [here](#).

In this guide, we will be building a power controller that can turn on/off AC equipment in a variety of ways (on-demand, timers, macro etc). Think of it as an automated power strip.

This guide assumes you have a working reef-pi installation, if not, start with the first guide [here](#)

In the first guide, we have gone through setting up reef-pi on a Raspberry Pi and test the equipment logic using a simple LED. We'll use the same software configuration with ancillary electronics and parts to control real AC equipment (like fan, heater, pumps etc.)
Parts

In addition to a Raspberry Pi to run reef-pi, we'll need a controllable power strip and some ancillary electronics for the control circuit. The control circuit will be built on top of a PermaProto HAT and will be connected to different connectors using jumper wires. The PermaProto HAT and all the connectors will be mounted on the plastic enclosure using standoffs.
## Powerstrip

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x American DJ SR P8 power strip</td>
<td>8 channel controllable power strip</td>
<td><a href="https://www.amazon.com/dp/B007EYHBVQ/">https://www.amazon.com/dp/B007EYHBVQ/</a></td>
</tr>
<tr>
<td>1 x Power supply</td>
<td>12 V 1A DC power adapter</td>
<td><a href="https://www.adafruit.com/product/798">https://www.adafruit.com/product/798</a></td>
</tr>
<tr>
<td>1 x DB9 cable</td>
<td>DB9 connector wire</td>
<td><a href="https://www.amazon.com/gp/product/B00QM8ZP5E/">https://www.amazon.com/gp/product/B00QM8ZP5E/</a></td>
</tr>
</tbody>
</table>

## Housing, connectors and standoffs

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x Nylon standoffs</td>
<td>Standoff to mount Raspberry Pi on housing</td>
<td><a href="https://www.adafruit.com/product/3299">https://www.adafruit.com/product/3299</a></td>
</tr>
<tr>
<td>1 x DB9 connector</td>
<td>Panel mount db9 connector</td>
<td><a href="https://www.amazon.com/dp/B073H9BKZ5/">https://www.amazon.com/dp/B073H9BKZ5/</a></td>
</tr>
<tr>
<td>1 x Plastic enclosure</td>
<td>Plastic enclosure</td>
<td><a href="https://www.amazon.com/dp/B0723DW8JM/">https://www.amazon.com/dp/B0723DW8JM/</a></td>
</tr>
<tr>
<td>1 x Barrel connector</td>
<td>2.1mm panel mount barrel connector</td>
<td><a href="https://www.amazon.com/dp/B01ERPVCV1Y">https://www.amazon.com/dp/B01ERPVCV1Y</a></td>
</tr>
</tbody>
</table>

## Electronics to control powerstrip

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x ULN2803A</td>
<td>Darlington transistor to control 12V relays</td>
<td><a href="https://www.adafruit.com/product/970">https://www.adafruit.com/product/970</a></td>
</tr>
<tr>
<td>1 x Solid core wire</td>
<td>Solid core wire for building jumpers on permaproto hat</td>
<td><a href="https://www.adafruit.com/product/1311">https://www.adafruit.com/product/1311</a></td>
</tr>
<tr>
<td>1 x Perma Proto HAT</td>
<td>Perma proto hat for mounting ULN2803a</td>
<td><a href="https://www.adafruit.com/product/2310">https://www.adafruit.com/product/2310</a></td>
</tr>
<tr>
<td>1 x IC socket</td>
<td>18 pin IC socket</td>
<td><a href="https://www.adafruit.com/product/2204">https://www.adafruit.com/product/2204</a></td>
</tr>
</tbody>
</table>
Circuit Construction

Circuit Details

American DJ SR P8 (the power strip), uses 12V relays to control each of its eight AC outlets. Each of these relays in turn represented by one pin in the Line In DB9 connector. We'll use reef-pi to generate the required 12V control signal for each of the 8 outlets. This signal will be created in the controller enclosure and exposed via a DB9 connector. We can then connect the enclosure with the power strip using a DB9 cable.

Internally, reef-pi will use one of the Raspberry Pi's GPIO for each outlet signal. Since Raspberry Pi GPIO produces 3.3V, we'll use a ULN2803 Darlington transistor to convert it to 12V. ULN2803 is an 8 channel Darlington transistor which works out perfectly.

Following is the pinout of ULN2803A, we'll connect each of the In pin to a Raspberry Pi GPIO pin, and Out pin to DB9 connector. COM will be connected to 12V and 0V to common GND (shared across LM2596, ULN2803A, and Raspberry Pi). ULN2803A is a sink type driver, hence we'll connect 12V power to the GND pin of the DB9 connector.

With the power strip control circuit operating at 12V and Raspberry Pi at 5V, we'll need a power source for each of this. Though it is possible to power them separately, it's inconvenient. We'll use a single 12V power supply and an LM2596 buck converted to convert 12V to 5V (you can use a 12V to 5V UBEC as well).
The LM2596 module allows variable voltage step down using the trimmer. Before wiring it to the rest of the circuit, tune it to generate 5V output by rotating the trimmer screw. LM2596 modules are cheap and effective in stepping down anywhere from 30V to 3.3V at 2.5A.

This is the fritzing breadboard view of the entire circuit. We are using a perma proto hat to mount ULN2803. Notice the DB9 connector exposing 8 12V control signal, and a barrel jack connector for 12V power source. The ULN2803A circuit is built on top of a PermaProto HAT, which also used to build a 12V rail to connect power source (barrel jack), LM2596 input pin, DB9 connectors GND pin (since ULN2803 is a sink type driver).

This entire circuit will be housed inside the enclosure. Female barrel jack connector will be used to connect power source with enclosure, and db9 connector for the powerstrip. ULN2803 circuit will be built on top of a PermaProto HAT.

Before starting soldering or drilling the plastic enclosure, arrange all the main components (connectors, pi, PermaProto HAT) to get a tentative idea. We'll use same arrangements to draw the drilling holes for the standoffs in next section.
Prepare Connectors

All connectors (DB9 and barrel jack) will be wired with the PermaProto HAT using jumper wires. For this, I start with male-female jumper wires, then strip off the male end, tin it with soldering iron.

We’ll be powering this build with DC 2.1 barrel jack connector based 12V power supply. A panel mount female barrel jack connector is fixed on the controller housing.

Solder the jumper wires to the barrel connector. I prefer to use appropriate colored (red for +Ve, black for GND) jumper wires for individual pins. This simplifies understanding the overall circuit.
To determine the +Ve and GND pins in barrel connectors, I generally plug in the power source and use a multimeter to identify which pins represents +Ve and GND.

The DB9 connector we are using has screw terminals, we can simply screw in one end of the jumper wire while the other end can be connected to male headers (that connected to ULN2803A output pin).

It's OK to use other types (not screw terminal) of DB9 terminals as well.

Use PermProto HAT to Mount ULN2803A

On the Perma Proto HAT, ULN2803 is mounted using a 18 pin IC socket, which is soldered on the HAT. Pi GPIO pins are connected to ULN2803 input pins using 22
AWG solid core jumper wires. We'll also solder male header pins for each of the output of ULN2803, as well as COM.

Connect 8 individual GPIO with each of the In pins of ULN2803A.

Solder male header pins to LM2596 module. We'll use female-female jumper wires to connect the input pin of this module with 12V rail on PermaProto HAT. The output pins will be connected back to PermProto HAT's 5V rail. It is safe to connect all GND together (LM2596 can share a common input and output GND).
Solder male header pins to LM2596 module. We'll use female-female jumper wires to connect the input pin of this module with 12V rail on PermaProto HAT. The output pins will be connected back to PermProto HAT's 5V rail. It is safe to connect all GND together (LM2596 can share a common input and output GND).

Adjust the LM2596 module for precisely 5.1 V output. For this, connect the 12V power source to the input pin (use the barrel connector we have built in the earlier step) and a multimeter (with alligator clip) to the output pin. Rotate the trimmer pot screw till multimeter indicates 5.1-5.2V. LM2596 is fairly precise, you should be able to configure exactly 5.1V.

We have not built the housing yet, but for reference, this is the final state we are looking at. You don't have to follow the exact layout, thanks to the PermaProto HAT's design, there's plenty other ways to construct this, as long as the connections are proper (follow the breadboard schematic), we are good. Notice the 12V rails on PermaProto HAT, a set of 3 male header pins located in the same column as ULN2803A pin 10 or COM pin. The right side of the PermaProto HAT has male header pins for GND. In the build image below, I have passed the red jumper wires beneath the PermaProto HAT to reduce visual clutter.
Building the Housing

With the circuit constructed, we'll move on to building the housing. We'll start with a 7x4" plastic enclosure from uxcell. It does not need to be this specific enclosure, you can choose something else if you have prior familiarity. Some of the reef-pi users have built their own 3D printed enclosures.

Large Plastic Project Enclosure - Weatherproof with Clear Top
Store your project safe and sound in this nice weatherproof box with a clear top. We picked up this box because we like the machinable ABS plastic body and tough clear polycarbonate...
https://www.adafruit.com/product/905

We'll be mounting connectors and the PermaProto HAT inside the housing. PermaProto HAT (and Raspberry Pi) will be mounted using nylon standoffs. While barrel jack connected or DB9 connector will be directly screwed in.

Arrange all the parts to get an idea and mark the exact locations that needs to be drilled (for connectors or standoffs).

I use a marker pen to highlight the parts that needs to be drilled.

DB9 connector and barel jack connector will be mounted on the side.
PermaProto HAT and LM2596 module will be mounted on the back.

I use dremel to drill the bigger holes for DB9 and barrel jack connector, A drill driver to for the standoff holes.

That's it. once holes are drilled, mount all the components and connect the pins with jumper wires.
Test that you are getting 5V on PermProto HAT (coming from LM2596) before connecting Raspberry Pi. Once tested, connect Raspberry Pi and your reef-pi should controller should be good to go. Screw in the enclosure lid.

Next, we'll configure reef-pi UI and test our new power controller.

**Configure & Test**

**Declare Outlets**

With circuit and enclosure completed, our physical controller is done. Connect the controller's with American DJ SR P8 power strip using a DB9 cable, Connect DC 12V power source to the controller, and reef-pi should boot up. Next we'll declare all the outlets and equipment required to test our build.
Head to Configuration section of the reef-pi user interface (UI). Under configuration section, click on the connector sub-section. We’ll create one outlet for each of the GPIO pins. Make sure the GPIO pin numbers reflect our wiring.

Create one outlet for each of the 8 GPIO pins.

Next we'll create some virtual equipment for testing these outlets.

Test with Equipment

Navigate to the Equipment section and create a virtual equipment, associate it with an outlet.

Declare 8 equipment for each of the 8 outlets.
Once all eight equipment is declared, we can click on the on/off button and reef-pi to turn on/off corresponding AC outlet. You should be able to hear the sound relay latching on/off. Validate each of the AC outlets is indeed turning on/off from the reef-pi equipment UI by plugging in an AC equipment (such as table lamp) to the relevant AC outlet in the power strip.

Configure Dashboard

reef-pi allows showing the current state of equipment (on or off) as a chart in the main dashboard. To enable this, navigate to the Dashboard tab and click on the Configure button on left side. Configure the dashboard to have one row and two columns. The first cell will show health metrics (CPU & memory usage) and the second cell will show the equipment chart. Select the appropriate charts by clicking on the drop-down menu inside individual cells.

Click on the Update button to save the dashboard settings and then click on the Back to dashboard button to go back to dashboard view. You should see the equipment chart in the dashboard now. Each equipment item is represented as a vertical bar, with color indicating their state (red for off and green for on). This chart is refreshed automatically after every 10 seconds, so charts can be at most 10 seconds stale.
At this point, we can use reef-pi to turn on/off any AC equipment on-demand.

Schedule on/off with Timers

reef-pi allows controlling AC equipment periodically using timers. Timers are popular to automate various reef keeping chores (for example light turn on/off at a fixed time every day).

Navigate to the Timers tab and create a timer with following details. reef-pi uses Linux `cron` like grammar to specify timers.

The above timer tells reef-pi after every 10 seconds, turn on equipment Eq6 and then turn after 2 seconds. In short, Eq6 will be turned on for 2 seconds after every 10 seconds. Cron is a very powerful language that allows a variety of periodicity to be expressed concisely.

Macro

The macro feature in reef-pi allows executing a series of steps sequentially on demand. A step can be triggering an action against an equipment or waiting. In short,
macro allows users to turn a set of equipment on or off in specific order with a single click of a button. A popular use of this feature is to automate feed mode or weekly water changes, where pumps in reef tanks are turned off during the feeding of corals & fish or to replace tank water with clean saltwater. Following is an example of a macro in reef-pi, it will trigger different actions against different equipment with certain delays in-between them.

![Example of a macro in reef-pi](image)

Once declared, this macro can be triggered by single click of a button

![Macro trigger button](image)

Although reef-pi is aimed for reef tanks, the power controller feature can be used in any general purpose AC equipment control, like home automation or even making a Christmas lighting controller.

Finally, the power control module is a foundational layer in reef-pi. Other modules, like temperature controller and water level controller, requires the power controller feature.

In our next reef-pi guide we'll be extending this power controller to monitor temperature and turn on/off a heater or cooler to maintain a stable tank temperature.