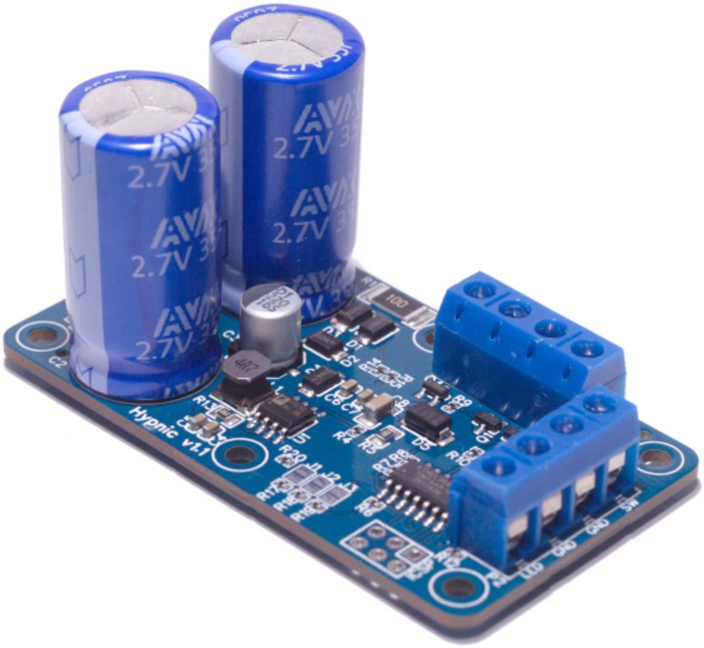
FALK

Hypnic Power Manager

User Manual v1.0



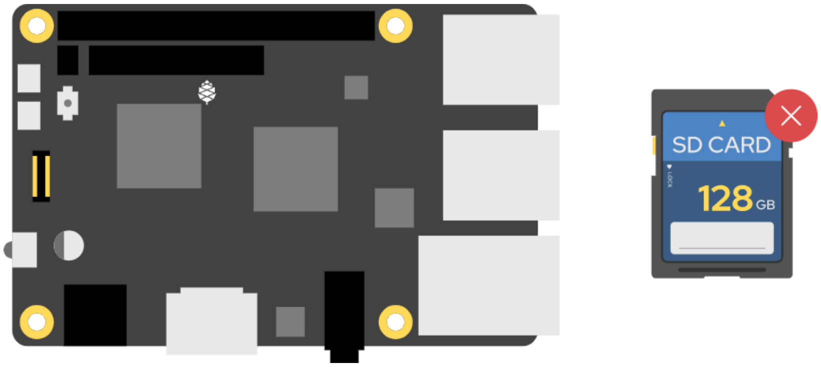
Hardware Revision v1.1

Contents

Introduction

The Hypnic Power Manager is a small board designed to provide power management and safe shutdown to the Raspberry Pi and other Single Board Computers. It’s ideal for hard-to-reach devices that don’t need to be on when the main power supply has failed.

All Single Board Computers run an operating system (OS), generally from an SD card but sometimes from eMMC storage. As the OS boots, it loads components of itself, and running services, into memory. Changes you make are often saved in memory and only written to your SD card/eMMC/USB key occasionally - and as the OS performs it’s shutdown sequence. Should your device lose power while running, any data changed since the last write-down is lost. Often this is harmless as the OS can detect and fix the errors, but sometimes (and there’s no telling when) this can be catastrophic. The result is that the SD card is corrupted and either a service or even the OS itself becomes unusable, rendering the whole thing unbootable.

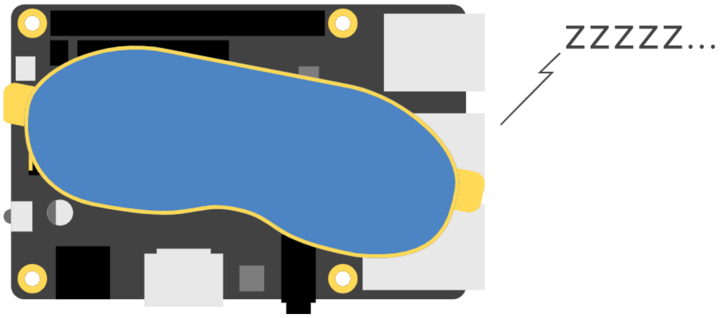


Unlike traditional UPS-style power managers, which use LiPo batteries to provide minutes or hours of uptime, the Hypnic uses supercapacitors. These provide enough power that there’s time to safely shutdown the Single Board Computer (SBC) and prevent corruption. Depending on the connected SBC (and it’s peripherals), the Hypnic can provide enough power for 15-30s of runtime - plenty of time for a safe shutdown sequence! Best of all, supercapacitors require no maintenance. They have no “battery memory” and don’t need replacing.

Why use a Hypnic instead of a UPS? Often, when the power is lost it’s fine for the device to power down, usually because the service it was providing is now impossible until the power is restored. Examples of this would be in audio players (distros like VolumIO come to mind), media devices (Kodi, etc.) and anywhere else where a power loss would mean the device can’t perform it’s duties. There’s no point having your irrigation controller turn on/off the sprinklers when the sprinkler valves themselves have no power to open! When the power is restored, the Hypnic will automatically power the device back on.

What's Up With The Name?

You're probably wondering about the name - Hypnic. Hypnic Jerk is the name for the small twitch your body often makes just as you're falling asleep. While the actual cause is unknown, some believe that it's your body's attempt to protect you as you fall asleep. Sound familiar? The Hypnic Power Manager detects when power is lost and the SBC must sleep, and does so in a safe manner!



Features

The Hypnic Power Manager is designed to "get out of the way". It's simple to use and generally won't need any maintenance at all. All you need to do is wire it up once and forget all about it! This is what it can do for you:

- Provides up to 5V@3A to a single board computer
- Covers a brownout event (temporary power loss) of up to two seconds
- Full support for a connected momentary power button
- Performs a graceful shutdown of the SBC when external power is lost or power button is pressed
- Completely powers off the SBC after shutdown
- Automatically powers on the SBC when external power is restored
- Provides system status via a connected LED
- Eliminates battery maintenance through the use of supercapacitors
- Supports DIYers by providing full access to the firmware and microcontroller

Hardware

Overview

Connectors

P1 - SBC Connection

Pin 1	Safe
Pin 2	Halt
Pin 3	Ground
Pin 4	5V

P2 - Front Panel Connection

Pin 1	LED+
Pin 2	LED- (Ground)
Pin 3	Switch- (Ground)
Pin 4	Switch+

P3 - Input Power

Pin 1	5V
Pin 2	Ground

Specifications

Input Voltage	5V
Input Current (max)	3.5A
Output Voltage	5V
Output Current (max)	3A
Microcontroller	ATtiny44
Supercapactors	2 x 30F
Quiescent Current (standby)	~1mA
Quiescent Current (max)	~100mA
Connectors	SBC Connector (4P Screw Terminal) Front Panel Connector (4P Screw Terminal) Input Power (2P Screw Terminal)

Installation

Installation of the Hypnic Power Manager is such a breeze, you could do it in your sleep (or just before…)

Software

First of all, install the operating system of your choice onto an SD card for your SBC. For the Raspberry Pi, this is usually Raspbian. The Hypnic software currently supports major Linux distributions that leverage systemd (this includes Raspbian, Armbian and most major Linux distributions).

Once your SBC is up and running, connect it to the internet if possible - this will make the installation easier. Most installations will include the tools required, but ensure you have wget installed before your continue:

```
sudo apt install wget -y
```

Once installed, you can begin the software installation with the following command:

```
wget -qO -
https://raw.githubusercontent.com/gilphilbert/hypnic/main/sbc/install.sh | sudo
bash
```

This will run a script to install the Hypnic software onto your device. The installation will download the required packages for installation, download the Hypnic software components and install two services: one to receive Halt notifications from the Hypnic device and one to send a Safe signal back. During the installation, you will be asked which pins the Halt and Safe connections are connected to. Enter your selection here, or use the defaults. You can change them later if you need to.

Software - Offline

If you do not have internet connectivity, you can download the latest installer from the GitHub repository as a zip file:

https://github.com/gilphilbert/hypnic/archive/refs/heads/main.zip

Simply extract the zip file, navigate to the sbc directory and run the installer:

```
unzip main.zip
cd hypnic/sbc
bash install.sh
```

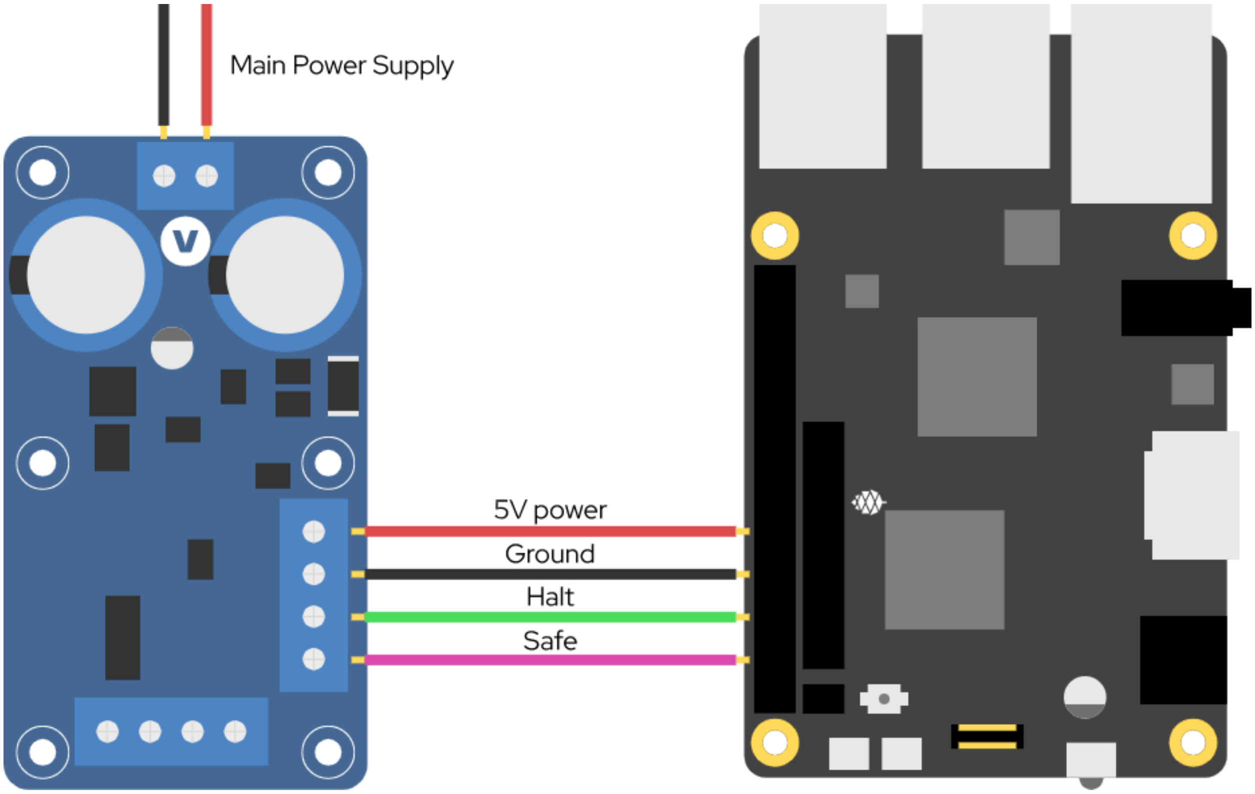
Hardware

Now the fun part! Located header P1 and connect the 5V line to an available 5V connector on the SBC's GPIO header. You can find this documented in the manual for your SBC. Connect the GND connection to any available Ground connection on the GPIO header.

Next, connect the Halt and Safe connections to available GPIO pins. The installation for the Raspberry Pi uses BCM GPIO 17 (physical pin 11) for the Halt message and BCM GPIO 27 (physical pin 13) for the Safe pin by default. Unless you're using these pins for other purposes, these pins are the easiest to use.

Finally, connect a suitable 5V power supply to the input power connector on the Hypnic device. A power supply that offers 3.5A or greater is recommended, although less will do for smaller SBCs. Connect the cables as shown on the device either side of header P3.

Your connections for P1 and P3 should be as follows:



Once connected, provide power to the Hypnic device. It will automatically power on and start up the SBC.

The Hypnic device can also support a power button, using a momentary switch, and an LED to provide device status. Connect these to the SBC using the header P2. The LED anode (+) should be connected to the LED pin and the cathode (-) should be connected to the adjacent GND pin. The switch should be connected to the SW pin and the remaining GND pin - which way doesn’t matter.

Signals

As you've seen, the Hypnic device has just a few connections to the SBC - two are power and two are signal pins. The power pins should be self-explanatory; the ground pin is constant ground while the 5V pin is switched on and off to turn the SBC on and off.

Pin	Direction	Purpose
Halt	Hypnic > SBC	Shutdown instruction
Safe	Hypnic < SBC	Shutdown complete

The Halt signal is supplied by the Hypnic to the SBC. It's purpose is to instruct the SBC to begin a graceful shutdown sequence (via `sudo poweroff`). This is achieved by changing the state of the Halt pin, which is interpreted by the Hypnic State Service which then triggers the shutdown process. This provides the operating system on the SBC to flush it's memory and unmount filesystems - important steps to prevent corruption.

Once the shutdown sequence is complete, the final service run is the Hypnic Shutdown Service. This service is only triggered on a shutdown/halt, not on a reboot. When it runs, it changes the state of the Safe pin, notifying the Hypnic that the shutdown sequence is complete. At this point, the Hypnic will turn off the power supply to the SBC.

To turn the SBC back on, simply press the switch wired to the Switch and Ground pins on the Hypnic.

The default pin configuration is as follows:

Single Board Computer	Halt Pin	Safe Pin
Raspberry Pi (B+ and later)	GPIO 17 (physical pin 11)	GPIO 27 (physical pin 13)

Using The Hypnic Device

The Hypnic device is relatively simple in operation: the device starts with the power disconnected. The Hypnic will detect when power is applied and will power on the SBC. The Hypnic service on the SBC then monitors the GPIO pin connected to the Halt connection on the Hypnic. When either the power supply to the Hypnic is lost, or if the power button is pressed, the Hypnic will send a signal to the SBC using the Halt pin. The software detects this and starts a graceful shutdown of the operating system. Meanwhile, the LED attached to the Hypnic will flash. Once the SBC shutdown sequence is completed, the final action is to send a signal to the Hypnic device. When the Hypnic device receives the connection, the power to the SBC is disconnected and the LED is turned off.

Switch

In the powered off state, the switch simply provides power to the SBC. When the device is powered on, pressing the switch sends a Halt signal to the SBC using the Halt pin. The Hypnic will now wait for the SBC to confirm power down (Safe). If a Safe signal has not been received after 20 seconds, the Hypnic device will disconnect the power to the SBC.

Status LED

The status LED is designed to show you what state the SBC is in. There are three states for the LED:

State	Meaning
Off	The SBC is currently powered off
On	The SBC is currently powered on
Flashing	The Hypnic device has sent a halt signal to the SBC and is waiting for a Safe signal to disconnect the power

DIY Customization

As an open source device, the design files and firmware are publicly available for the Hypnic Power Manager, covered under an open source license that allows individuals to modify the device as they wish. Please be aware that customized firmware can't be supported and any damage you cause is yours to fix/repair/replace.

Getting the Firmware

The current firmware is available on the Hypnic GitHub repository:

https://github.com/gilphilbert/hypnic

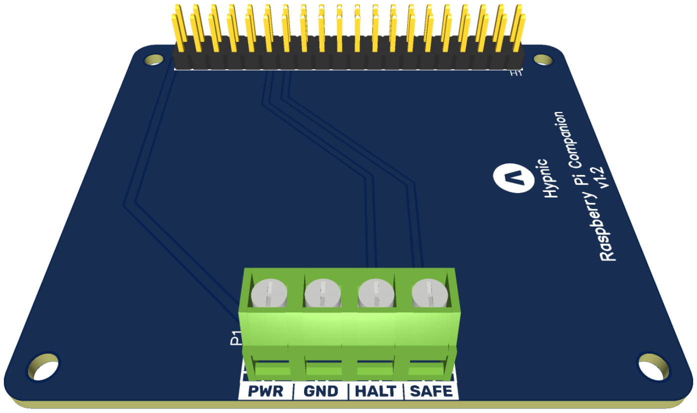
The Hypnic is developed using PlatformIO, an IDE for microcontroller development that's integrated into Visual Studio Code. To get started, install Visual Studio Code then add the PlatformIO extension. Clone the Hypnic firmware locally:

```
git clone https://github.com/gilphilbert/hypnic.git
```

Open the **`hypnic`** folder in Visual Studio code and navigate to the **`src`** directory. Inside you will find **`main.cpp`** - the firmware file for the Hypnic.

Edit the firmware as required, then use an ICSP-capable programmer (such as Sparkfun's AVR Pocket Programmer) to flash the modified firmware to the Hypnic using the ICSP header on the bottom right of the board. You can flash the firmware from within PlatformIO, although you will need to modify the **`upload_protocol`** line in **`platformio.ini`** if you use a different programmer.

Raspberry Pi Companion Board



Introduction

As described in the sections above, the Hypnic power manager requires connectivity to the Single Board Computer - 5V and Ground to power the device and two additional general purpose IO pins (GPIO) to control the power on/off signals. The Hypnic can be connected directly to the GPIO header on the Single Board Computer but this prevents other devices being stacked on top of the SBC. In most instances this is fine, but sometimes full access to the header is required for your application. In this event, the Raspberry Pi Companion Board can be used to provide a stackable HAT solution for the Raspberry Pi, allowing the Hypnic to be connected to the Raspberry Pi but still allowing full access to the GPIO.

The Raspberry Pi Companion board fits the Raspberry Pi's HAT specification, providing access to the underlying GPIO and connects the Hypnic to the default pins configured in the software utility, pins 17 and 27. The function of these pins and the connection to the Raspberry Pi is described in the following section.

Raspberry Pi GPIO Usage

The Raspberry Pi Companion Board connects the Hypnic to the following pins:

- 11 (BCM pins 17) - Halt signal from the Hypnic
- 13 (BCM pin 27) - Safe signal to the Hypnic board

The Hypnic provides power to the Raspberry Pi using the 5V and Ground pins

GPIO (BCM)	Name	Physical	Physical	Name	GPIO (BCM)
	3.3V	1	2	5V	
2	SDA	3	4	5V	
3	SCL	5	6	GND	
4	GPIO7	7	8	TXD	14
	GND	9	10	RXD	15
17	GPIO 0	11	12	GPIO 1	18
27	GPIO 2	13	14	GND	
22	GPIO 3	15	16	GPIO 4	23
	3.3V	17	18	GPIO 5	24
10	MOSI	19	20	GND	
9	MISO	21	22	GPIO 6	25
11	SCLK	23	24	CE0	8
	GND	25	26	CE1	7
0	SDA	27	28	SCL	1
5	GPIO 21	29	30	GND	
6	GPIO 22	31	32	GPIO 26	12
13	GPIO 23	33	34	GND	
19	GPIO 24	35	36	GPIO 27	16
26	GPIO 25	37	38	GPIO 28	20
	GND	39	40	GPIO 29	21