

# Raspberry Pi ON/OFF Power Controller

*Power up your Raspberry Pi with a latching push-button controlled MOSFET switch*

It would be convenient to use a momentary contact push button switch to turn ON and turn OFF your Raspberry Pi (RPI). A press of a button should apply power to the micro USB header, and keep it ON while the Raspberry Pi initializes and starts its application programs. There should be several options for powering OFF the Raspberry Pi:

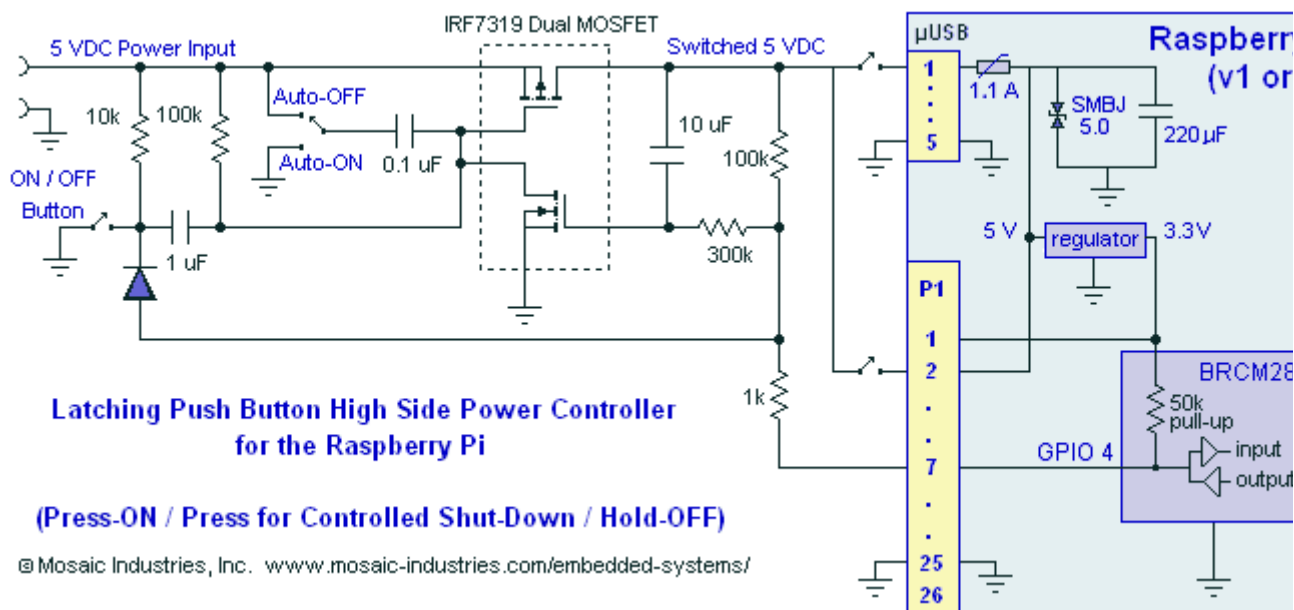
## Chapter Contents

Using the Raspberry Pi GPIO 4 signal for both input and output

Power latching circuit operation and software control

- The RPi should be able to turn itself OFF fully autonomously, under program control, automatically on an internal timer, or in response to a network signal.
- You should be able to initiate a turn OFF sequence by momentarily pressing an OFF button. The Raspberry Pi should then shut-down its application program in an orderly way, and turn itself OFF.
- If the RPi fails to turn itself OFF or crashes, you should be able to press the button longer to force it OFF.

A soft power switch with this behavior can be implemented using a high-side latching MOSFET power switch. The latching and toggle ON/OFF power switch circuit for controlling microcontrollers shown [here](#) is easily adapted for controlling a Raspberry Pi in this manner. Here's the electronic circuit to use:



**Fig. 1** Push button ON/OFF latching MOSFET power switch circuit for the Raspberry Pi

The schematic shows you two options<sup>1)</sup> for applying the switched 5V power to your Raspberry Pi. You can apply it through the micro USB connector, in which case it passes through a PolySwitch™ resettable fuse (rated at 1.1 A)<sup>2)</sup> before being applied to the rest of the RPi circuitry, or, you can apply it to header P1, in which case it bypasses the fuse. The advantage of applying power to P1 and bypassing the fuse is that the 5V is then passed on to the USB connectors unimpeded, where it can drive high current USB devices.<sup>3)</sup>

Whichever pins you choose for applying the 5V to the board, you should use the GPIO 4 signal on P1 pin 7 for feedback to the circuit. The Raspberry Pi is then able to shut itself OFF by driving that pin low.

The power controller conveniently uses two MOSFETs packaged together, the [IRF7319](#), which allows you to build a very small circuit.

## Using the Raspberry Pi GPIO 4 signal for both input and output

---

The basic circuit operation is discussed [here](#) and [here](#), but there are a few considerations in its design I should mention:

- The circuit uses a single Raspberry Pi input/output pin – the GPIO 4 signal. Because the GPIO signal numbering is different from and inconsistent with the underlying BCM2835 chip pin numbering and the RPi connector numbering, let me be explicit: We use the GPIO 4 signal which appears on pin 7 of the 26-pin header P1. Some signals differ between RPi Rev 1 and Rev 2 – but the GPIO 4 signal appears on pin 7 of P1 for both versions.
- By default GPIO 4 is initialized as an input, but idles high because it is pulled up by an internal resistor. While we don't know the precise value of the pull-up resistor, it's probably in the range of 40 KΩ - 100 KΩ as discussed [here](#). Being pulled high doesn't affect the circuit much during startup, as the internal pull-up resistor is effectively across the 100K resistor of the circuit. As soon as the circuit powers up, in response to a button press, the GPIO 4 input is pulled up internally. It actually acts as a very weak output pin, helping to pull up the gate of the N-MOSFET, and keeping the circuit ON.
- The circuit's 100 KΩ does pull the input towards 5V, and in general you shouldn't connect an input to anything greater than 3.3V as that will forward bias the input substrate diode and inject current into the RPi's 3.3V internal supply rail. However, in this case, the current injected is very small, only about  $(5-3.3-0.5)/100\text{ K}\Omega = 17\text{ }\mu\text{A}$ , and poses no harm of burn-out or distortion of internal signals. The RPi's input pin limits the voltage applied to the N-MOSFET gate to about 3.9 V, but that is still more than sufficient to keep the MOSFET fully turned ON.
- The 1 KΩ protects the Raspberry Pi pin from excessive current in the event that it is inadvertently set as an output high while the button is held down.
- While the circuit is in the OFF state, it draws essentially no quiescent current. The only current drawn is the inconsequential reverse leakage current through the diode and MOSFET transistors.

## Power latching circuit operation and software control

---

- The circuit uses two MOSFET transistors in a latching feedback arrangement. The high-side PMOS power switch provides power and is controlled by the complementary NMOS transistor.
- When input power is first applied, the MOSFET power switch comes up in its OFF state if the jumper is in the Auto-OFF position, or ON in the Auto-ON position. You may replace the jumper with fixed wiring. While OFF the circuit consumes no power, and when ON only the P-MOSFET's small ON resistance dissipates power, and that is extremely low.
- When your Raspberry Pi is OFF, a momentary button press turns it ON.
- When your Raspberry Pi is ON, a momentary button press doesn't affect the power switch, but it can be detected by the RPi if GPIO 4 is configured to produce an interrupt on a falling edge, or if polled frequently enough. Your RPi can then implement an orderly shut-down.
- You can also use GPIO 4 to turn OFF the power. After the RPi detects a button press, or at any other time determined by its application program, GPIO 4 can be reconfigured as an output and set low. That forces the MOSFET latch to switch OFF after several seconds.
- Ready or not, your RPi can be forced to power down by pressing and holding the button for more than three seconds.

When the Raspberry Pi is first powered up, GPIO 4 is automatically configured as an input with passive pull-up. That is the behavior we want, so you don't need to do anything to guarantee it. Subsequently, the application program should do the following:

- It should configure GPIO 4 to call an interrupt service routine when it sees a falling edge or an input low.
- The interrupt service routine should initiate a shut-down of the any applications running and whatever I/O they use.
- Once the applications have been shut-down safely, your program should reconfigure GPIO 4 as an output and set it low.
- Setting the output low resets the MOSFET switches to their OFF state after several seconds, turning OFF your Raspberry Pi.

---

The content of this page is provided under a  
[Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

---

## Notes:

- 1)  
The options are shown as switches; as a practical matter you would wire up one path or the other.
- 2)  
Although the PolySwitch™ is unspecified on the schematic, it is probably part number [miniSMDC110F-2](#), with 1.1 A working current and 2.2 A trip current.
- 3)  
At least in the case of RPi Rev 2. In RPi Rev 1 the dual USB connectors receive power through a resettable fuse, a [miniSMDC014](#) (140 mA working current, 280 mA trip current). On Rev 1.1 boards the fuses were replaced with 0 ohm shorts, and on Rev 2 boards the 5V is applied directly to the connectors.