

# GPIO Electrical Specifications

*Raspberry Pi input and output pin voltage and current capabilities*

The [Raspberry Pi](#) provides general purpose digital input/output pins (called GPIO pins) that you can use for reading digital logic signals or for outputting digital logic levels. The outputs do not have much current capability, but you can drive LEDs or other low current devices. Of course to use them effectively and safely, you need to know their voltage levels and drive capability, and unfortunately, the RPi vendors don't supply the GPIO pin specs.

It is a source of frustration to Raspberry Pi users that the electrical characteristics of the input/output pins of the Broadcom BCM2835 SoC (system-on-chip) are not adequately specified in the Raspberry Pi documentation or processor datasheet.

This page tries to address that inadequacy by piecing together information from the [Raspberry Pi foundation](#), other sites including the [eLinux Wiki](#), and datasheets of other processors. However, please treat any updated specifications at [RaspberryPi.org](#) as definitive.

On this page we'll attempt to assemble coherent electrical specs by looking at comparable SoCs. We'll show that the electrical specs of comparable products do differ. Consequently, you won't find definitive specs for the RPi here, but I think you will find the information to be useful enough to make sound engineering decisions when interfacing to the RPi GPIO pins.

The RPi's Broadcom BCM2835 incorporates an ARM1176 processor core. The ARM processors are low-cost yet high performance processors used in smart phones, digital TVs, eReaders and other media devices. ARM licenses the processor to other companies, like Broadcom, who combine it with various I/O modules and incorporate it into system-on-chip (SoC) designs. ARM also provides the physical IP for the digital inputs and outputs (GPIO) as cell libraries.

In this document we piece together what is known of the electrical specifications for the general purpose input/output pins from the various RPi documents and from comparable SoC devices that also use the ARM1176JZF5 core.

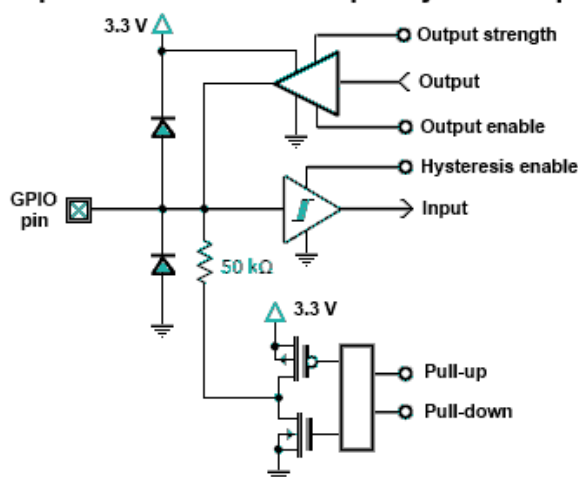
## GPIO pin circuitry

The Raspberry Pi's GPIO pins are quite versatile, and you can modify many of their characteristics from software. You can turn on/off input pin hysteresis, limit output slew rate, and control source and sink current drive *capability* from 2 mA to 16 mA in 2 mA increments. These properties are set for the GPIO block as a whole, not on a pin-by-pin basis.

Source/sink current capability does not limit the current into or out of the pin, but only specifies the maximum current for which the output signal high/low voltage specifications will be met. If misused, output pins can be damaged by excessive current irrespective of the source/sink current programmed. After a reset, the RPi comes up with the GPIO outputs set to 8 mA drive capability.

The following diagram shows a simplified schematic of the I/O pin circuitry:

### Equivalent Circuit for Raspberry Pi GPIO pins



### Chapter Contents

- [GPIO pin circuitry](#)
- [Input and output pin electrical specifications](#)
- [RPi GPIO input voltage and output current limitations](#)
- [GPIO power-on state](#)
- [Raspberry Pi driver software for configuring and setting GPIO pins](#)
- [GPIO pinout on header P1](#)
- [Finding your board's revision number](#)

**Fig. 1** Equivalent circuit for a Raspberry Pi GPIO pin. The input diodes are actually parasitic FETs which can not handle any significant current.

There are a few subtleties and notable features of the circuit:

- The internal diodes shown in [the figure](#) are not really substrate diodes, but they are actually parasitic FETs. Electrically, their I-V characteristic looks like a diode's, but with a greater forward drop and a more gradual knee. They may protect against low current transient events caused by transient out-of-range voltages applied to the pins, but they are not intended to protect against the application of voltages greater than the supply voltage or less than ground, even with an external series resistor. In brief, you should never deliberately forward bias those "diodes". Consequently, you can not safely place an external pull-up resistor to 5V on the I/O pin. That would forward bias a parasitic FET and owing to its poor internal impedance to the chip's internal power rail it may overheat, or worse, it may bias up parts of the chip to voltages greater than they can handle. So, don't do it!
- The input gate detects whether the input voltage level is low or high by comparing it to a threshold voltage. Normally the voltage threshold is about 1.8V, but it isn't guaranteed; it can be anywhere between the maximum input low and minimum input high, that is, between about 0.8 and 2.0V.
- To provide noise immunity and prevent chatter on transitions, you can configure the input gate to act as a Schmitt trigger, with input hysteresis. With hysteresis there are different threshold voltages for rising and falling transitions. Broadcom doesn't specify the degree of hysteresis, but [Table 1](#) offers some representative values from other ARM1176-based devices. For a better understanding of how it works, you can read our more detailed description of [input pin hysteresis for the Freescale 9S12/HCS12 MCU](#). That description pertains to a different processor, but the operation is the same.
- The output drives to either 3.3V (high) or 0V (low). Only one driver is shown in the schematic, but in actuality there are several wired in parallel, with software controlling which are enabled, so that the output drive impedance can be varied to offer 2 mA to 16 mA source and sink current capability, in 2 mA increments.
- While a pull-up/down resistor value of 50 K $\Omega$  is shown in the figure, Broadcom doesn't specify its value. It is likely to be in the 40 to 100 K $\Omega$  range, as shown in [Table 1](#). The transistors that enable the pull-up/down resistor contribute their appreciable nonlinear resistance to it, so the effective resistance is not constant over the input pin voltage range. The pull-up behavior is somewhat of a cross between constant resistance and constant current behavior.

## Input and output pin electrical specifications

---

In lieu of actual measurements, the following specifications are provide as good faith estimates that you may use as guidelines in your designs. These specifications are derived from the Raspberry Pi datasheets as well as datasheets of other other ARM1176-based SoCs, specifically these:

[Freescale MCIMX31](#) multimedia application processor;  
[Renasas  \$\mu\$ PD77630A](#) mobile multimedia processor; and,  
[Development board for the ARM1176JZF](#)

Their GPIO specs are summarized in the following table:

GPIO input/output pin electrical characteristics	
<b>Output low voltage</b> $V_{OL}$	< 0.40 V <sup>1)</sup> < 0.66 V <sup>2)</sup> < 0.40 V <sup>3)</sup> < 0.40 V <sup>4)</sup>
<b>Output high voltage</b> $V_{OH}$	> 2.40 V <sup>5)</sup> > 2.64 V <sup>6)</sup> > 2.90 V <sup>7)</sup>
<b>Input low voltage</b> $V_{IL}$	< 0.80 V <sup>8)</sup> < 0.54 V <sup>9)</sup> < 1.15 V <sup>10)</sup>
<b>Input high voltage</b> $V_{IH}$	> 2.00 V <sup>11)</sup> > 2.31 V <sup>12)</sup> > 2.15 V <sup>13)</sup>
<b>Hystereses</b>	> 0.25 V <sup>14)</sup> 0.66 – 2.08 V <sup>15)</sup>
<b>Schmitt trigger input low threshold</b> $V_{T-}$	1.09 – 1.16 V <sup>16)</sup> 0.9 V <sup>17)</sup>
<b>Schmitt trigger input high threshold</b> $V_{T+}$	2.24 – 2.74 V <sup>18)</sup> 0.90 V <sup>19)</sup>
<b>Pull-up/down resistance</b>	40 – 65 K $\Omega$ <sup>20)</sup> 100 K $\Omega$ <sup>21)</sup>
<b>Pull-up/down current</b>	< 50 $\mu$ A <sup>22)</sup> < 28 $\mu$ A <sup>23)</sup>
<b>Pin capacitance</b>	5 pF <sup>24)</sup>
<b>Bus hold resistance</b>	5-11 K $\Omega$ <sup>25)</sup>

**Table 1** Speculative Raspberry Pi GPIO input output pin electrical specifications

## RPi GPIO input voltage and output current limitations

You should keep the following limitations in mind when using the GPIO pins:

- These are 3.3 volt logic pins. A voltage near 3.3 V is interpreted as a logic one while a voltage near zero volts is a logic zero. A GPIO pin should never be connected to a voltage source greater than 3.3V or less than 0V, as prompt damage to the chip may occur as the input pin substrate diodes<sup>26)</sup> conduct. There may be times when you may need to connect them to out-of-range voltages – in those cases the input pin current must be limited by an external resistor to a value that prevents harm to the chip. I recommend that you never source or sink more than 0.5 mA into an input pin.
- To prevent excessive power dissipation in the chip, you should not source/sink more current from the pin than its programmed limit. So, if you have set the current capability to 2 mA, do not draw more than 2 mA from the pin.
- Never demand that any output pin source or sink more than 16 mA.
- Current sourced by the outputs is drawn from the 3.3 V supply, which can supply only 50 mA maximum. Consequently, the maximum you can source from all the GPIO outputs simultaneously is less than 50 mA. You may be able to draw transient currents beyond that limit as they are drawn from the bypass capacitors on the 3.3 V rail, but don't push the envelope!
- There isn't a similar limitation on sink current. For sink current, the pertinent limitation is that imposed by maximum chip power dissipation. Even so, you can safely sink up to 16 mA each into any number of GPIO pins simultaneously. In the worst case, the output pins (if configured to the 16mA high current drive capability) have a maximum output low voltage of about 0.4 V<sup>27)</sup>, and their internal circuitry dissipates only 6.4 mW worst case. Even sinking 16 mA into 16 pins simultaneously would produce only 0.1024 W, that is, about a tenth of a watt. However, depending on the source of the current, transient sink currents may make demands on the board's bypass capacitors, so you may not want to switch all the outputs to sink the maximum current synchronously, if you require fast, clean transitions.
- Do not drive capacitive loads. Do not place a capacitive load directly across the pin. Limit current into any capacitive load to a maximum transient current of 16 mA. For example, if you use a low pass filter on an output pin, you must provide a series resistance of at least

$$3.3\text{V}/16\text{mA} = 200\ \Omega.$$

## GPIO power-on state

The Raspberry Pi's Broadcom SoC does its own power-on initializations of its GPIO pins. However, these initializations may be undone or modified by the kernel's startup software.

## Raspberry Pi driver software for configuring and setting GPIO pins

Software for controlling the GPIO pins is readily available. You can consult the following sources:

[RPI Low-level peripherals](#)

[GPIO C library](#)

[GPIO python library](#)

## GPIO pinout on header P1

The GPIO signals and power supplies are brought out to a 26-pin header, designated *P1* on the PCB's silkscreen. *P1* provides two rows of pins, odd and even. Pin 1 of header *P1* is the corner pin closest to the *P1* label.

### Caution -- Pins are numbered unconventionally

Note that the orientation of *P1* differs from the convention used in most connector placement. On most computer boards the header would be rotated 180°, so that the odd-numbered row would lie alongside the board edge, instead of the even-numbered row. That way, the wires or ribbon cables running from the edge of the board are numbered from left to right. The Raspberry Pi board violates the convention, placing its even-numbered rows along the board's edge. So be careful in making your connections.

The following table identifies the signal names delivered to each of the pins of *P1*, for Rev-1 and Rev-2 boards:

P1: 26-pin header			
Signal	Pins		Signal
+3.3 V	—1	2—	+5 V
GPIO_02 <i>GPIO_00</i> (SDA1)	—3	4—	+5 V
GPIO_03 <i>GPIO_01</i> (SCL1)	—5	6—	GND
GPIO_04 (GPCLK0)	—7	8—	GPIO_14 (UART0_TxD)
GND	—9	10—	GPIO_15 (UART0_RxD)
GPIO_17	—11	12—	GPIO_18 (PCM_CLK)
GPIO_27 <i>GPIO_21</i>	—13	14—	GND
GPIO_22	—15	16—	GPIO_23
+3.3 V	—17	18—	GPIO_24
GPIO_10 (SPI_0_MOSI)	—19	20—	GND
GPIO_09 (SPI_0_MISO)	—21	22—	GPIO_25
GPIO_11 (SPI_0_SCLK)	—23	24—	GPIO_08 (SPI_0_CE0_N)
GND	—25	26—	GPIO_07 (SPI_0_CE1_N)
Notes:			
1. Signals in plain type are for Rev 2 boards, signals for Rev 1 boards are shown in <i>italics</i> .			
2. Primary functions for signals are shown in parentheses.			

Notes:

- The signal names used in the above table are the Broadcom chip signal names – these are the

names to use when programming the board. The RPi board provides alternate descriptive names for the same signals, and those names are often confusing. For example, `GPIO_14` at the chip is called `TXD0` at the connector, a name that illustrates its primary function, but, `GPIO_17` at the chip becomes `GPIO_GEN0` at the connector, a renaming that can only cause confusion.

- All GPIO signals default to inputs except for `GPIO_14` and `GPIO_15` (on pins 8 and 10), which are configured by default as `UART0_TxD` and `UART0_RxD` respectively.
- You may draw no more than 50 mA from the +3.3 V supply.
- The board uses approximately 700 mA from the +5 V supply. You may draw current from the +5 V pins provided the sum of your current and the board's 700 mA can be provided by the supply you provide to the board.
- You may deliver power to the board through header *P1* at pins 2 or 4 as well as delivering it to the microUSB socket, except that power delivered through *P1* bypasses the 1.1 A PolySwitch resettable fuse.
- Pins 3 and 5 are pulled up on the board to the 3.3V supply by 1.8 kΩ resistors to enable them to be used easily as [I2C communication](#) pins.

## Finding your board's revision number

To determine the revision number for your board you can execute the following command from a prompt in a terminal window,

```
cat /proc/cpuinfo
```

If the `cpuinfo` number begins with 1000 then your Raspberry Pi has been over-voltaged or over-clocked (which permanently flips a write-once memory bit in the processor). Your hardware revision is simply the last 4 digits of the number (i.e. 0002 in this example).

You can find a table associating the `cpuinfo` number and the board revision at the [eLinux wiki](#)<sup>28)</sup>, which is reproduced here for your convenience as the following table.

cupinfo number	Release Date	Model	PCB Revision	Memory	Notes
Beta	Q1 2012	B (Beta)	?	256MB	Beta Board
0002	Q1 2012	B	1.0	256MB	
0003	Q3 2012	B	1.0/ECN0001	256MB	no fuses and D14 removed
0004	Q3 2012	B	2.0	256MB	Mfg by Sony
0005	Q4 2012	B	2.0	256MB	Mfg by Qisda
0006	Q4 2012	B	2.0	256MB	Mfg by Egoman
0007	Q1 2013	A	2.0	256MB	Mfg by Egoman
0008	Q1 2013	A	2.0	256MB	Mfg by Sony
0009	Q1 2013	A	2.0	256MB	Mfg by Qisda
000d	Q4 2012	B	2.0	512MB	Mfg by Egoman
000e	Q4 2012	B	2.0	512MB	Mfg by Sony
000f	Q4 2012	B	2.0	512MB	Mfg by Qisda

For further information on the electrical specifications of the GPIO pins see:

- [GPIO Datasheet Addendum - GPIO Pads](#)

And for more, and possibly more up-to-date information on the various hardware revisions of the boards see:

- [RPI Hardware History](#)

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

## Notes:

- 1) [RealView® Platform Baseboard for ARM1176JZF-S](#)
- 2) [Freescale MCIMX31 multimedia application processor](#)
- 3)

[Renasas μPD77630A mobile multimedia processor](#)

4)

[JESD8C.01 JEDEC interface standard for 3.3V devices](#)

5), 8), 11)

RealView® Platform Baseboard for ARM1176JZF-S

6)

Freescale MCIMX31 multimedia application processor

7), 10), 13), 15), 16), 18), 20), 22), 25)

Renasas μPD77630A mobile multimedia processor

9), 12), 14), 21), 23)

Freescale MCIMX31 multimedia application processor

17), 19)

0.5\*Vcore, Freescale MCIMX31 multimedia application processor

24)

At the processor, discounting PCB trace and connector capacitances; including those may double or triple this value. RealView® Platform Baseboard for ARM1176JZF-S

26)

actually parasitic FETs, shown as diodes in [Figure 1](#)

27)

It's likely that the designers conform at a minimum to the JESD8C.01 JEDEC interface standard for 3.3 V devices, which requires a maximum output low of 0.40 V or less, at the stated maximum current.

28)

Please consult [eLinux wiki](#) for up-to-date info on RPi revisions.

This page is about: GPIO Electrical Specifications, Raspberry Pi Input and Output Pin Voltage and Current Capability – The Raspberry Pi provides general purpose digital input/output pins (called GPIO pins) that you can use for reading digital logic signals or for outputting digital logic levels. The outputs do not have much current capability, but you can drive LEDs or ...

[User Agreement](#) | [Privacy Policy](#) | [End User License Agreement](#) | [Sales Terms and Conditions](#) | [Life Support Policy](#) | [Trademarks](#) | [All Policies](#)

© Mosaic Industries, Inc. All rights reserved.