

# Demystifying Microcontroller GPIO Settings

**4 JUNE 2018 BY PHILLIP JOHNSTON • LAST UPDATED 21 SEPTEMBER 2021**

Anyone who writes software for microcontrollers will have to configure and manage general purpose input/output (GPIO) pins. On their surface, GPIO configuration seems simple: pins are either input or output, and they can be high or low.

This mental model will be true until you come across a fancy processor with a plethora of configuration options, or an electrical engineer will request pin settings that you don't understand ("please make this line Hi-Z").

This guide will help you understand the different pin configuration options that are provided on modern microcontrollers.

## **Table of Contents:**

### 1. Background Information

1. Tri-state Logic
2. High-impedance
3. Floating
4. Pull-up
5. Pull-down
6. Current Sink
7. Current Source

### 2. Input and Output Modes

1. GPIO Input Modes
2. GPIO Output Modes
  1. Push-Pull Output
  2. Open-Drain Output
  3. Open-Collector

### 3. Speed

4. High Drive
5. Further Reading

## Background Information

Before we dive into configuration options, it's useful to understand some general descriptions and terms that are associated with GPIO and IO signals.

### Tri-state Logic

Most modern GPIO pins are implemented as a tri-state buffer. This means that the GPIO pin can effectively assume three values:

1. Logical 0 (connection to ground)
2. Logical 1 (connection to VCC)
3. High-impedance (also called “floating”, “Hi-Z”, “tri-stated”)

### High-Impedance

When a line is put into a high-impedance state, the output is effectively removed from the circuit. This allows multiple circuits or devices to share the same output lines and is commonly utilized to implement communication busses. Failure to utilize a high-impedance state when it's required leads to IO contention and short-circuits.

### Floating

A signal is said to be “floating” when its state is indeterminate, meaning that it is neither connected to VCC or to ground. The signal's voltage will “float” to match the residual voltage.

The term “floating” is often used interchangeably to describe a pin which is in the high-impedance state.

### Pull-up

Pull-ups are resistors that connect a signal to VCC. Pull-ups are used to set a default state when the signal is floating.

Recall that when an input pin is in high-impedance mode and not driven by external sources, it is floating at a residual voltage level. Pull-up resistors prevent the pin from floating by forcing the signal to VCC when it is not being actively driven. When another source drives the signal low (connects to ground), the pull-up is overridden and the input pin will read a '0'.

Many microcontrollers supply internal pull-up configuration options. Sometimes, a specific pull-up resistor value is required which necessitates using an external pull-up instead of a chip's internal pull-up.

## Pull-down

Pull-downs are resistors that connect an signal to ground. Pull-downs are used to set a default state when the signal is floating. When another source drives the signal high (connects to VCC), the pull-down is overridden and the input pin will read a '1'.

Many microcontrollers supply internal pull-down configuration options. Sometimes, a specific pull-down resistor value is required which necessitates using an external pull-down instead of a chip's internal pull-down.

## Current Sink

A “current sink” means that current is flowing into a pin, node, or signal. For digital IO, a current sink provides the ground connection to the load.

## Current Source

A “current source” is the opposite of a current sink: the current is flowing out of a pin, node or signal. For digital IO, a current source provides the voltage source to the load.

Both a current source and a current sink have current flowing, but in different directions.

# Input and Output Modes

The primary configuration option for a GPIO pin is *input* or *output*.

## GPIO Input Modes

When a GPIO is configured as an input, it can be used to read the state of the electrical signal. Configuring a GPIO as an input puts the pin into a high-impedance state.

In general, there GPIO inputs are primarily configured in one of three ways:

1. High-impedance (often the default – floats if not driven)
2. Pull-up (internal resistor connected to VCC)
3. Pull-down (internal resistor connected to Ground)

Most GPIO input pins also feature internal hysteresis, which prevents spurious state changes on the pins. Usually hysteresis is a built-in feature rather than a configurable setting.

## GPIO Output Modes

When a GPIO is configured as an output, it can be used to drive a signal high or low. There are primarily two configuration options for GPIO outputs: push-pull and open-drain.

### Push-pull output

Push-pull is the default GPIO output setting in most cases. A push-pull GPIO has the ability to both source and sink current.

With a push-pull GPIO, a transistor connects to VCC or GND to drive a signal high or low. When the output goes low, the signal is actively “pulled” to ground, and when the output goes high it is actively “pushed” to VCC.

### Open-Drain Output

Unlike push-pull, an open-drain output can only sink current. The output has two states: low and high-impedance. In order to achieve a logical high output on the line, a pull-up resistor is used to connect the open-drain output to the desired output voltage level.

You can think of an open-drain GPIO as behaving like a switch which is either connected to ground or disconnected.

Open-drain GPIO can typically be configured in two different modes:

- Open-drain

- Open-drain with internal pull-up

Most applications which utilize open-drain circuitry utilize external pull-ups on open-drain outputs. Often, internal pull-up values are not sufficient for the target circuitry.

Open-drain outputs are useful when multiple gates or pins are connected together, such as with the I2C bus. When a device is not using the bus, the open-drain output is in high-impedance mode and the voltage level is pulled high by the pull-up resistor. When a device drives the output low, all connected lines will go low, as they are tied together.

Another common use for open-drain outputs is having multiple external devices drive a single, active-low interrupt pin on a microcontroller.

## Open-Collector

An “open collector” is functionally the same as an “open drain”. An “open collector” refers to a current sink on a BJT transistor output, while an “open drain” refers to a current sink on a FET output.

I encounter “open collector” pins more on component datasheets than I do on microcontrollers.

## GPIO Speed

GPIO speed controls the slew rate, or the rate at which a signal can change between low/high values (the “rise time” and “fall time”). Speed configuration options are described as “speed”, “slew rate”, “frequency”, and “high-frequency mode”.

By increasing the GPIO speed, you increase the rate of change of the output voltage (reducing rise time). However, power consumption and noise radiated by the circuit increases along with the GPIO speed. By default, you should keep GPIO speed low unless there is a specific reason for increasing it.

## High Drive

High-drive GPIO are push-pull pins that are capable of providing more current than typical pins. While you must check each chip's datasheet to understand the current capacity of your pins, typical push-pull GPIO can source/sink around  $\pm 8\text{mA}$ , while a high-drive pin can source/sink up to  $\pm 40\text{mA}$ .

High-drive pins enable your microcontroller to directly drive IO that requires higher-than-normal current, such as an LED. Using high-drive pins can help simplify the electrical design and reduce cost by eliminating the need for external current amplifying circuitry.

## Further Reading

- Three-state Logic
- Open Collector
- Push-pull Output
- What is an open drain?
- Understanding Microcontroller Pin Input Output Modes
- Push-pull/open drain; pull-up/pull-down
- STM32 Understanding GPIO Settings

## Acknowledgments

Thanks to Gabriel Staples for identifying corrections needed by the article.



**UNCATEGORIZED**



**FEATURED, GPIO, MICROCONTROLLERS**

## 11 Replies to “Demystifying Microcontroller GPIO Settings”

**Siddiq Habib**

**2 JANUARY 2019 AT 15:36**

If i configure a GPIOpin to O/P (say PushPull mode) will i be able to still read the status of this GPIO pin, or should i configure it to I/P and read the value?

**Phillip Johnston**

**3 JANUARY 2019 AT 00:35**

This depends on the architecture. Some chips will let you read the value that is currently set, others will cause that pin to change to input mode.

**Siddiq Habib**

**2 JANUARY 2019 AT 15:47**

You mentioned above that in I/P mode the GPIO is in High-impedance by default.  
What will be the GPIO state when configured to O/P in Pushpull mode ( i think it will be not pulled at all – which is HighZ)?

**Phillip Johnston**

**3 JANUARY 2019 AT 00:34**

This may be dependent on the chip, but if you set the push-pull pin to output mode, I would expect the value to be GND.

**QM**

**16 JANUARY 2019 AT 10:28**

Took a note of the "background information", that will make a valuable reminder.  
I think pull-up and pull-down can also be used when the GPIO is being used as output, to hold their status, for low power operation, when the micro is chip is being rebooted.  
Am I correct?  
Is this the same of push-pull output?

## Phillip Johnston

16 JANUARY 2019 AT 21:21

Depending on the architecture, you can set a pull-up/pull-down with a GPIO output. However, this will not usually work when the microchip is being rebooted: in that scenario, the pins revert to their reset state.

To enforce a pin state when the micro is being rebooted, you will need external pull-up/pull-down resistors.

## Gabriel Staples

24 APRIL 2019 AT 20:44

Great article! I've posted a link to it at the bottom of my Stack Overflow answer here: <https://electronics.stackexchange.com/questions/28091/push-pull-open-drain-pull-up-pull-down/354993#354993>. I have 2 minor corrections though:

1) spelling: "change change" → "can change": "GPIO speed controls the slew rate, or the rate at which a signal change change between"

2) Change the word "transistor" to "BJT transistor": "An "open collector" refers to a current sink on a transistor output, while an "open drain" refers to a current sink on a FET output."

Contrasting "transistor" with "FET" implies a FET is not a transistor, which is absolutely incorrect, since the very acronym F-E-T stands for "Field-Effect *Transistor*"

([https://en.wikipedia.org/wiki/Field-effect\\_transistor](https://en.wikipedia.org/wiki/Field-effect_transistor)).

## Phillip Johnston

24 APRIL 2019 AT 20:53

Hi Gabriel,

I've made the corrections. Thanks for writing in!

Cheers,

Phillip



**Samuel**

**19 JANUARY 2021 AT 07:15**

Thank you!

**Moises**

**24 JULY 2021 AT 02:26**

Thanks for sharing this.

**Nelu**

**11 SEPTEMBER 2022 AT 04:45**

Helpful and nice explanations! Thanks!