# NOVASOM\_M7.GPIO module basics

## Importing the module

To import the NOVASOM\_M7.GPIO module:

import NOVASOM\_M7.GPIO as GPIO

By doing it this way, you can refer to it as just GPIO through the rest of your script.

## Pin numbering

There are two ways of numbering the IO pins on a NOVASOM M7 board within NOVASOM\_M7.GPIO. The first is using the BOARD numbering system. This refers to the pin numbers on the P1 header of the M7 board. The advantage of using this numbering system is that your hardware will always work, regardless of the board revision of the NOVASOM\_M7. You will not need to rewire your connector or change your code.

The second numbering system is the BCM numbers. This is a lower level way of working - it refers to the channel numbers on the Broadcom SOC embedded on the Raspberry Pi. You have to always work with a diagram of which channel number goes to which pin on the NOVASOM M7 board. Your script could break between revisions of NOVASOM M7 boards.

To specify which you are using using (mandatory):

GPIO.setmode(GPIO.BOARD)

# or

GPIO.setmode(GPIO.BCM)

To detect which pin numbering system has been set (for example, by another Python module):

mode = GPIO.getmode()

The mode will be GPIO.BOARD, GPIO.BCM or None

## Warnings

It is possible that you have more than one script/circuit on the GPIO of your NOVASOM M7 board. As a result of this, if NOVASOM\_M7.GPIO detects that a pin has been configured to something other than the default (input), you get a warning when you try to configure a script. To disable these warnings:

GPIO.setwarnings(False)

## Setup up a channel

You need to set up every channel you are using as an input or an output. To configure a channel as an input:

GPIO.setup(channel, GPIO.IN)

(where channel is the channel number based on the numbering system you have specified (BOARD or BCM)).

More advanced information about setting up input channels can be found [here](https://sourceforge.net/p/raspberry-gpio-python/wiki/Inputs/).

To set up a channel as an output:

GPIO.setup(channel, GPIO.OUT)

(where channel is the channel number based on the numbering system you have specified (BOARD or BCM)).

You can also specify an initial value for your output channel:

GPIO.setup(channel, GPIO.OUT, initial=GPIO.HIGH)

## Setup more than one channel

You can set up more than one channel per call. For example:

chan\_list = [11,12] # add as many channels as you want!

# you can tuples instead i.e.:

# chan\_list = (11,12)

GPIO.setup(chan\_list, GPIO.OUT)

## Input

To read the value of a GPIO pin:

GPIO.input(channel)

(where channel is the channel number based on the numbering system you have specified (BOARD or BCM)). This will return either 0 / GPIO.LOW / False or 1 / GPIO.HIGH / True.

## Output

To set the output state of a GPIO pin:

GPIO.output(channel, state)

(where channel is the channel number based on the numbering system you have specified (BOARD or BCM)).

State can be 0 / GPIO.LOW / False or 1 / GPIO.HIGH / True.

## Output to several channels

You can output to many channels in the same call (release 0.5.8 onwards). For example:

chan\_list = [11,12] # also works with tuples

GPIO.output(chan\_list, GPIO.LOW) # sets all to GPIO.LOW

GPIO.output(chan\_list, (GPIO.HIGH, GPIO.LOW)) # sets first HIGH and second LOW

## Cleanup

At the end any program, it is good practice to clean up any resources you might have used. This is no different with NOVASOM\_M7.GPIO. By returning all channels you have used back to inputs with no pull up/down, you can avoid accidental damage to your NOVASOM M7 board by shorting out the pins. Note that this will only clean up GPIO channels that your script has used. Note that GPIO.cleanup() also clears the pin numbering system in use.

To clean up at the end of your script:

GPIO.cleanup()

It is possible that don't want to clean up every channel leaving some set up when your program exits. You can clean up individual channels, a list or a tuple of channels:

GPIO.cleanup(channel)

GPIO.cleanup( (channel1, channel2) )

GPIO.cleanup( [channel1, channel2] )

## NOVASOM M7 Board Information and RPi.GPIO version

To discover information about your RPi:

GPIO.RPI\_INFO

To discover the NOVASOM M7 board revision:

GPIO.RPI\_INFO['P1\_REVISION']

To discover the version of RPi.GPIO:

GPIO.VERSION