Docs » MicroPython libraries » machine — functions related to the hardware »
class ADC – analog to digital conversion

class ADC – analog to digital conversion

Usage:

```
import machine

adc = machine.ADC()             # create an ADC object
apin = adc.channel(pin='GP3')   # create an analog pin on GP3
val = apin()                    # read an analog value
```

Constructors

class **machine.ADC**(*id=0, *, bits=12*)

Create an ADC object associated with the given pin. This allows you to then read analog values on that pin. For more info check the pinout and alternate functions table.

> Warning
>
> ADC pin input range is 0-1.4V (being 1.8V the absolute maximum that it can withstand). When GP2, GP3, GP4 or GP5 are remapped to the ADC block, 1.8 V is the maximum. If these pins are used in digital mode, then the maximum allowed input is 3.6V.

Methods

ADC.channel(*id, *, pin*)

Create an analog pin. If only channel ID is given, the correct pin will be selected. Alternatively, only the pin can be passed and the correct channel will be selected. Examples:

```
# all of these are equivalent and enable ADC channel 1 on GP3
apin = adc.channel(1)
apin = adc.channel(pin='GP3')
apin = adc.channel(id=1, pin='GP3')
```

ADC.init()

Enable the ADC block.

`ADC.deinit()`

Disable the ADC block.

class ADCChannel — read analog values from internal or external sources

ADC channels can be connected to internal points of the MCU or to GPIO pins. ADC channels are created using the ADC.channel method.

`machine.adcchannel()`

Fast method to read the channel value.

`adcchannel.value()`

Read the channel value.

`adcchannel.init()`

Re-init (and effectively enable) the ADC channel.

`adcchannel.deinit()`

Disable the ADC channel.