

class RTC – real time clock

The RTC is an independent clock that keeps track of the date and time.

Example usage:

```
rtc = machine.RTC()
rtc.init((2014, 5, 1, 4, 13, 0, 0, 0))
print(rtc.now())
```

Constructors

class `machine.RTC(id=0, ...)`

Create an RTC object. See `init` for parameters of initialization.

Methods

RTC.init(*datetime*)

Initialise the RTC. Datetime is a tuple of the form:

```
(year, month, day[, hour[, minute[, second[, microsecond[, tzinfo]]]])
```

RTC.now()

Get the current datetime tuple.

RTC.deinit()

Resets the RTC to the time of January 1, 2015 and starts running it again.

RTC.alarm(*id, time, *, repeat=False*)

Set the RTC alarm. Time might be either a millisecond value to program the alarm to current time + `time_in_ms` in the future, or a `datetime` tuple. If the time passed is in milliseconds, `repeat` can be set to `True` to make the alarm periodic.

RTC.alarm_left(*alarm_id=0*)

Get the number of milliseconds left before the alarm expires.

RTC.cancel(*alarm_id*=0)

Cancel a running alarm.

RTC.irq(*, *trigger*, *handler*=None, *wake*=machine.IDLE)

Create an irq object triggered by a real time clock alarm.

- **trigger** must be **RTC.ALARM0**
- **handler** is the function to be called when the callback is triggered.
- **wake** specifies the sleep mode from where this interrupt can wake up the system.

Constants

RTC.ALARM0

irq trigger source