Docs » MicroPython libraries » `esp32` — functionality specific to the ESP32

# `esp32` — functionality specific to the ESP32

The `esp32` module contains functions and classes specifically aimed at controlling ESP32 modules.

## Functions

**`esp32.wake_on_touch`**(*wake*)

Configure whether or not a touch will wake the device from sleep. *wake* should be a boolean value.

**`esp32.wake_on_ext0`**(*pin*, *level*)

Configure how EXT0 wakes the device from sleep. *pin* can be `None` or a valid Pin object. *level* should be `esp32.WAKEUP_ALL_LOW` or `esp32.WAKEUP_ANY_HIGH`.

**`esp32.wake_on_ext1`**(*pins*, *level*)

Configure how EXT1 wakes the device from sleep. *pins* can be `None` or a tuple/list of valid Pin objects. *level* should be `esp32.WAKEUP_ALL_LOW` or `esp32.WAKEUP_ANY_HIGH`.

**`esp32.raw_temperature`**()

Read the raw value of the internal temperature sensor, returning an integer.

**`esp32.hall_sensor`**()

Read the raw value of the internal Hall sensor, returning an integer.

## The Ultra-Low-Power co-processor

*class* **`esp32.ULP`**

This class provides access to the Ultra-Low-Power co-processor.

**`ULP.set_wakeup_period`**(*period_index*, *period_us*)

Set the wake-up period.

**`ULP.load_binary`**(*load_addr*, *program_binary*)

Load a *program_binary* into the ULP at the given *load_addr*.

`ULP.run`(*entry_point*)

Start the ULP running at the given *entry_point*.

## Constants

`esp32.WAKEUP_ALL_LOW`

`esp32.WAKEUP_ANY_HIGH`

Selects the wake level for pins.