

# Deep Sleep Shield

The schematic of the Deep Sleep Shield is available as a [PDF File](#).

## Pinout

The pinout of the Deep Sleep Shield is available as a [PDF File](#).



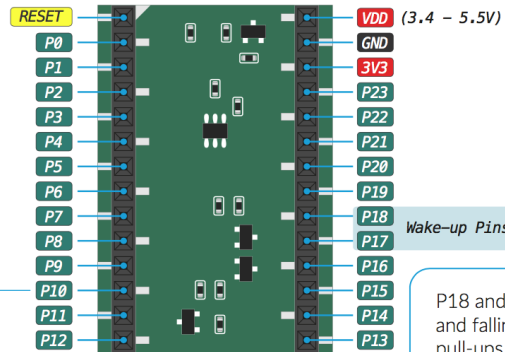
**Please note** - In order to use the **Deep Sleep** feature of this shield with a Pycom Development Device (i.e. LoPy), you will need to include/use a MicroPython library with your project code.

This library, along with example code, can be found at [github.com/pycom/pycom-libraries](https://github.com/pycom/pycom-libraries)

**\*\*The Deep Sleep Shield does not use the `machine.deepsleep([time_ms])` method!\*\***

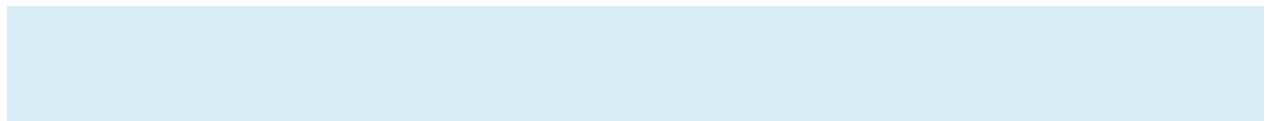
Timer wake up is configurable by software as well (1 second resolution)

*DEEP\_SLEEP\_ONEWIRE\_COMM\_PIN*



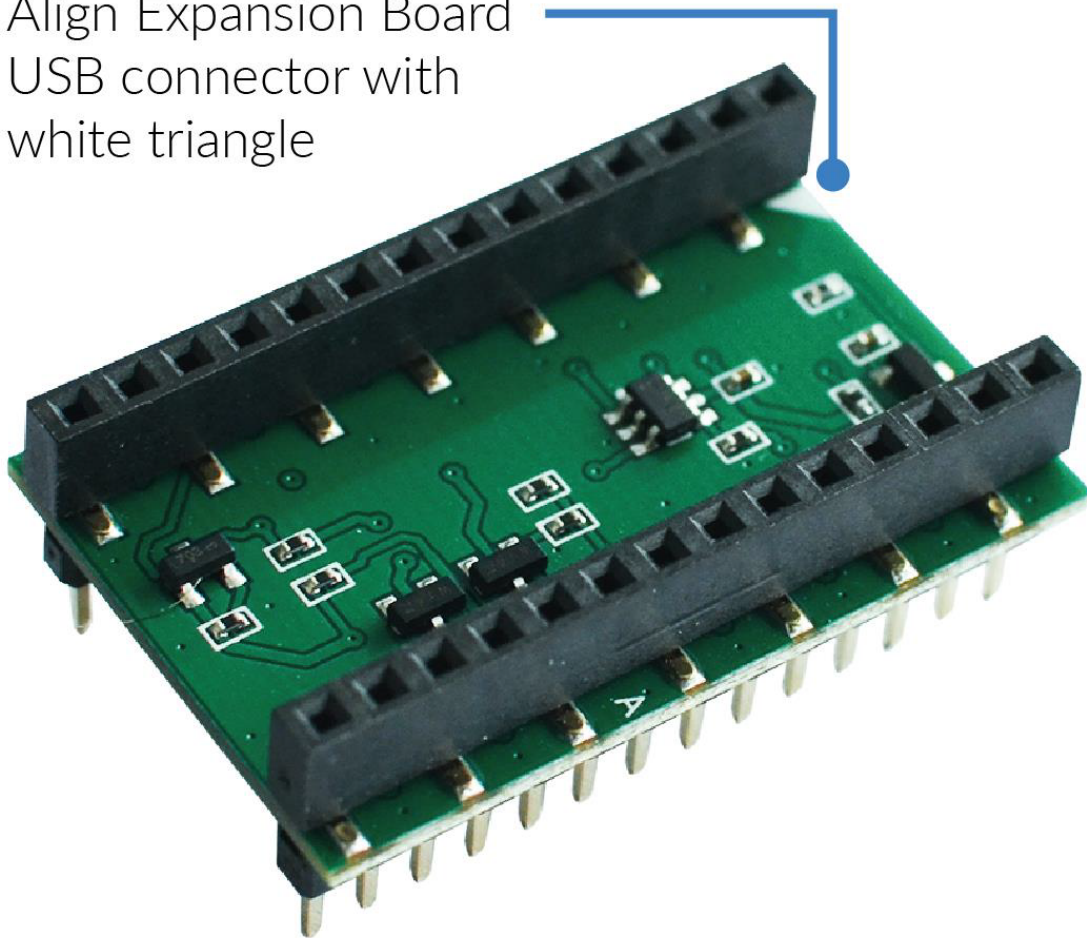
Wake-up Pins

P18 and P17 are able to wake up on rising and falling edge. These two pins have internal pull-ups configurable by software (Pull-downs if needed must be added externally)



To correctly connect a WiPy 2.0, LoPy or SiPy to the Deep Sleep Shield, align the white triangle on the Shield with the LED of the Pycom Device. Once the Pycome Device is seated onto the Deep Sleep Shield, this can then be connected to the Expansion Board.

Align Expansion Board  
USB connector with  
white triangle



# Deep Sleep API

This chapter describes the library which controls the Deep Sleep Shield. This includes the controls for external interrupts and timer setup of the deep sleep functionality.

To use this library, please upload the associated [Deep Sleep Library](#) to `/lib` on the target Pycom Device.

## Quick Example

```
from deepsleep import DeepSleep

ds = DeepSleep()

# get the wake reason and the value of the pins during wake up
wake_s = ds.get_wake_status()
print(wake_s)

if wake_s['wake'] == deepsleep.PIN_WAKE:
    print("Pin wake up")
elif wake_s['wake'] == deepsleep.TIMER_WAKE:
    print("Timer wake up")
else: # deepsleep.POWER_ON_WAKE:
    print("Power ON reset")

ds.enable_pullups('P17') # can also do ds.enable_pullups(['P17', 'P18'])
ds.enable_wake_on_fall('P17') # can also do ds.enable_wake_on_fall(['P17', 'P18'])

ds.go_to_sleep(60) # go to sleep for 60 seconds
```

## DeepSleep

The Deep Sleep Shield allows for waking up via a user trigger and also via an external interrupt (i.e. Accelerometer, Button).

### *Constructors*

#### **`class DeepSleep()`**

Creates a DeepSleep object, that will control the board's sleep features. For example;

```
ds = DeepSleep()
```

### *Methods*

#### **`deepsleep.enable_auto_poweroff()`**

This method allows for a critical battery voltage to be set. For example, if the external power source (e.g. LiPo Cell) falls below 3.3V, turn off the Pycom Device. This is intended to protect the hardware from under voltage.

#### **`deepsleep.enable_pullups(pins)`**

This method allows for pullup pins to be enabled. For example, if an external trigger occurs, wake the Pycom Device from Deep Sleep. `pins` may be passed into the method as a list, i.e. `['P17', 'P18']`.

**`deepsleep.disable_pullups(pins)`**

This method allows for pullup pins to be disabled. For example, if an external trigger occurs, wake the Pycom Device from Deep Sleep. `pins` may be passed into the method as a list, i.e. `['P17', 'P18']`.

**`deepsleep.enable_wake_on_raise(pins)`**

This method allows for pullup pins to trigger on a rising voltage. For example, if an external rising voltage triggers occurs, wake the Pycom Device from Deep Sleep. `pins` may be passed into the method as a list, i.e. `['P17', 'P18']`.

**`deepsleep.disable_wake_on_raise(pins)`**

This method allows for disabling pullup pins that trigger on a rising voltage. `pins` may be passed into the method as a list, i.e. `['P17', 'P18']`.

**`deepsleep.enable_wake_on_fall(pins)`**

This method allows for pullup pins to trigger on a falling voltage. For example, if an external falling voltage triggers occurs, wake the Pycom Device from Deep Sleep. `pins` may be passed into the method as a list, i.e. `['P17', 'P18']`.

**`deepsleep.disable_wake_on_fall(pins)`**

This method allows for disabling pullup pins that trigger on a falling voltage. `pins` may be passed into the method as a list, i.e. `['P17', 'P18']`.

**`deepsleep.get_wake_status()`**

This method returns the status of the pins at wakeup from deep sleep. The method returns a `dict` with the states of `wake`, `P10`, `P17`, `P18`.

**`deepsleep.set_min_voltage_limit(value)`**

This method relates to the `enable_auto_poweroff` method and allows the user to specify the minimum power off voltage as a value.

**`deepsleep.go_to_sleep(seconds)`**

This method sends the board into deep sleep for a period of `seconds` or until an external interrupt has triggered (see `set_pullups`).

**`deepsleep.hw_reset()`**

This method resets the PIC controller and resets it to the state previous to the pins/min-voltage being set.

Please note that more functionality is being added weekly to these libraries. If a required feature is not available, feel free to contribute with a pull request at the [Pycom Libraries GitHub repository](#).