

Code example: Power control for sensors and LED

Mercury V1 manual · v1.0

<https://www.altimetercloud.com/support/mercuryv1/code-power-control/>

The Mercury altimeter has sensors and the Neopixel LED's turned off by default. This feature exists so that items can be turned off in deep or light sleep as needed to save power and operate at maximum efficiency. It does mean though that you need to turn the power on before you use items.



Using Arduino IDE? Our online programmer includes Mercury_Pins.h by default so the pin names work without issue. If you are using Arduino IDE or another programmer, copy the Mercury_Pins.h tab content and paste it into the top of your program.

```
/*
 * Mercury V1 (ESP32-C6) Power control example
 * Turns on the Status LED power as well as the IMU power (Gyroscope and Accelerometer)
 */
#include "Mercury_Pins.h"

void setup() {
  pinMode(VACC, OUTPUT);
  pinMode(LEDPOWER, OUTPUT);
  digitalWrite(VACC, HIGH);
  digitalWrite(LEDPOWER, HIGH);
}

#pragma once
/*
 * Mercury (ESP32-C6) Pin Definitions
 * Board-specific GPIO assignments
 */

// — Status LED (NeoPixel) —
#define LEDPOWER 3 // NeoPixel power (drive HIGH to enable)
#define LED 2 // NeoPixel data signal

// — I2C Bus —
#define SDA 21 // I2C data
#define SCL 22 // I2C clock

// — Sensor Power —
#define VACC 20 // Sensor power rail (drive HIGH to enable)

// — General Purpose Ports —
#define GP06 6 // GP06 port
#define GP07 7 // GP07 port

// — High Current Output —
#define OUT1 5 // High current output (e.g. pyro / relay)

// — Battery Bar LEDs —
#define BL1 4 // Battery LED 1 (lowest)
#define BL2 14 // Battery LED 2
#define BL3 15 // Battery LED 3
```

```
#define BL4      18 // Battery LED 4
#define BL5      19 // Battery LED 5 (highest)
```

```
// — Indicators —
```

```
#define DISK     8 // Disk activity LED
```

```
// — Analogue / Detection —
```

```
#define BATIN    0 // Battery voltage (1:1 divider)
```

```
#define USBDETECT 1 // USB power detect (HIGH = USB present)
```