

第1章 GPIO操作とタイマ割り込みの基礎

マイコンI/Oの基本…
LED点滅

宮田 賢一

リスト1 LEDの点灯と消灯led.py

```

1: from machine import Pin
2: import time
3:
4: led = Pin(config.DEFAULT_LED, Pin.OUT)
5: for _ in range(10):
6:     led.on()
7:     time.sleep_ms(200)
8:     led.off()
9:     time.sleep_ms(200)

```

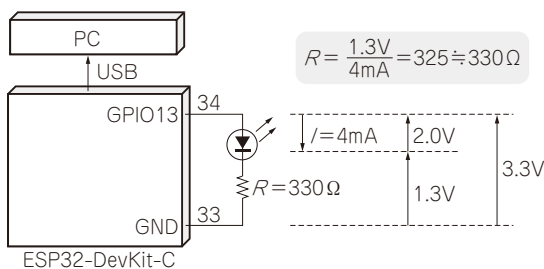


図1 ESP32-DevKit-CにLEDを接続する

● 動作の正常/異常を知る手段に便利なLED

システムの動作状態を表示したり、取得したデータを可視化したりすることは、ユーザ・インターフェースとして重要な要素です。第4部ではMicroPythonによるデータ表示手法を説明します。

LEDはシンプルな表示デバイスではありますが、単に電源ONを示すために点灯させるだけではなく、通信中など、特定の状況の間点滅させることで正しく動作していることを示すこともできます。また、マイコン上のプログラムのデバッグ中に何らかの不正な状態に陥ったことを示すために、コンソール画面がなくてもLEDの点灯でエラー状態を示すという使い方もあるでしょう。

● マイコンのGPIO端子を使って制御する

LEDはマイコンのGPIO (General Purpose Input/Output) を使って制御します。MicroPythonではmachineモジュールに含まれるPinクラス(第6部第1章)を使ってGPIOの制御ができます。

リスト2 タイマ割り込みによるLED点滅led_timer.py

```

1: from machine import Pin
2: from machine import Timer
3: import time
4:
5: state = 0
6: def toggle_led(t):
7:     global state
8:     led.value(state)
9:     state = 1 if state == 0 else 0
10:
11: led = Pin(config.DEFAULT_LED, Pin.OUT)
12:
13: t = Timer(config.DEFAULT_TIMER_ID)
14: t.init(period=100, callback=toggle_led)
15: time.sleep(3)
16: Timer.deinit(t)
17: # タイマが非初期化するまで待つ
18: time.sleep(0.1)
19: led.off()

```

プログラマが指定したタイミングでLEDの点灯または消灯をしたい場合は、Pinクラスのonメソッドとoffメソッドを必要な場所に記述するだけです。

リスト1は200ms間隔でLEDを10回点滅させるプログラムです。ラズベリー・パイPico/Pico WではオンボードのLEDが点滅します。ESP32にはオンボードLEDがないので、図1に示した回路図に従ってGPIOにLEDを外付けすれば、同じようなLEDの点滅を制御できます。

● タイマ割り込みによるバックグラウンド点滅

パイロット・ランプのように、マイコン・ボードが動作中であることを知らせるために、LEDを点滅させたい場合もあるでしょう。そのようなときにはタイマ割り込みを使った方法が便利です。タイマ割り込みを使うことにより、プログラムのメイン・ループ中でLEDの点滅間隔を制御するようなコードを書かずに、バックグラウンドで自動的に点滅させることが可能です。

MicroPythonでタイマ割り込みを扱うためのクラスはmachine.Timerです(第6部第1章)。リスト2はタイマ割り込み版のLED点滅プログラムです。

みやた・けんいち