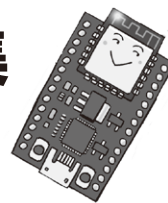


700円マイコンESP32ではじめる

逆引きMicroPythonプログラム集



角 史生

第3回

静電容量, 磁気, 赤外線(人感),
温湿度/気圧の検出

MicroPythonは、リソースの少ないマイコン上でPython 3と同じようにプログラミングできる環境の実現を目指して開発された言語処理系で、プロトタイプ開発に向いています。

プロトタイプ開発では、試作、テスト、修正を繰り返しながら開発が進みますが、MicroPythonを用いることでトライ&エラーが容易になります。

今回は、ESP32-WROOM-32 (Espressif Systems, 以降はESP32と表記)の内蔵センサや、外部センサ・モジュールを使って、測定データを見える化します。

2-1 静電容量(タッチ・センサ)で
指を検知する

● ESP32の静電容量検知機能を使う

ESP32は、静電容量の変化を検知できる入力端子を

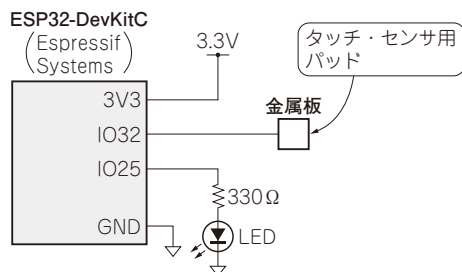


図1 タッチ・センサ用パッドと接続する回路

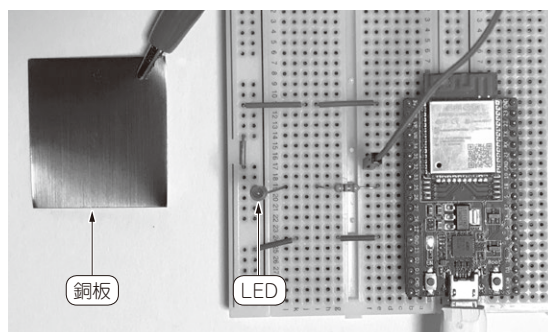


写真1 銅板で作成したタッチ・センサ用パッドを接続した様子

備えています。今回、指に触れることで変化する静電容量をmachineモジュールのTouchPadクラスを用いることでタッチ・センサとして使用します。

ESP32でタッチ・センサとして使える端子は次の通りです。

IO0, IO2, IO4, IO12, IO13, IO14, IO15, IO27, IO32, IO33

● 回路

回路図を図1に示します。写真1のように、IO32に金属板などで作成したタッチ・センサ用パッドを接続します。金属板に指が触れるとLEDが点灯し、離れるとLEDが消灯します。

● プログラム

タッチ・パッドの静電容量の変化に応じてLEDが点灯するプログラムをリスト1に示します。

2-2 内蔵ホール・センサで磁力を
検知する

ESP32には磁力を検知できるホール・センサが内蔵されており、hall_sensor関数を用いることで

リスト1 タッチ・パッドの静電容量の変化に応じてLEDが点灯するプログラム

```
from machine import TouchPad
from machine import Pin
import utime

touch_threshold = 390 # 閾値を390に設定
touch = TouchPad(Pin(32)) # GPIO32をタッチ・パッドに設定

led = Pin(25, Pin.OUT)
led.off()
while True:
    sense_value = touch.read() # タッチ・パッドの値を取得
    print(sense_value)
    if sense_value < touch_threshold:
        print("LED On") # センサの値が閾値以下になった場合
        led.on() # タッチされていると判断してLEDを点灯させる
    else:
        led.off() # そうでないならLEDを消灯する
    utime.sleep_ms(200) # 200msecウエイト(少し待つ)
```