第5章 制御処理をPythonコードで確認しながら進められる

実機②…倒立ロボの制御プログラムをPythonで実装

川村 聡

リスト1　クラス読み込み部…PWM，SPI，UARTなどのクラス読み出しとI/Oピンの設定を行う

```
import random
import machine,utime
from machine import PWM
from machine import UART,Pin
from machine import UART
from machine import SPI, Pin
from machine import Timer

spi = SPI(1, baudrate=5000000, polarity=0, phase=4,
          bits=8, firstbit=SPI.MSB, sck=Pin(14),
          mosi=Pin(15), miso=Pin(12))
cs = Pin(13, mode=Pin.OUT, value=1)          #CS=GP13
reset_spi = Pin(0, mode=Pin.OUT, value=0)    #Reset=GP0

joy1 = machine.ADC(0)
joy2 = machine.ADC(1)
vol  = machine.ADC(2)

uart0 = UART(0, baudrate=230400, rx=17, tx=16)
                                        #GP17=rx GP16=tx
SW1   = machine.Pin(6, machine.Pin.IN)        #GP6=IN
SW2   = machine.Pin(7, machine.Pin.IN)        #GP7=IN
Buza  = PWM(machine.Pin(22, machine.Pin.OUT))#GP22=PWM
led = machine.Pin(25, machine.Pin.OUT)        #GP25=OUT
led_R = machine.Pin(9, machine.Pin.OUT)       #GP9=OUT
led_G = machine.Pin(11, machine.Pin.OUT)      #GP11=OUT

mot_Ain_1 = machine.Pin(18, machine.Pin.OUT) #GP18=OUT
mot_Ain_2 = machine.Pin(19, machine.Pin.OUT) #GP19=OUT
mot_Bin_1 = machine.Pin(21, machine.Pin.OUT) #GP21=OUT
mot_Bin_2 = machine.Pin(20, machine.Pin.OUT) #GP20=OUT
pwm1 = PWM(machine.Pin(8, machine.Pin.OUT))  #GP8=PWM
pwm2 = PWM(machine.Pin(10, machine.Pin.OUT)) #GP10=PWM
```

（クラスの読み出し）　（I/Oピンの設定）

倒立ロボットの制御をPythonで記述する

● Pythonならパソコン上でも動かせる

倒立ロボットの制御アルゴリズムはラズベリー・パイPico 2（以降，Pico 2）に書き込んで動作させます．Pico 2はPicoとソフトウェア，ハードウェア（ピン配置含む）共に互換なので，Picoでも動作します．

Pico 2の開発にはMicroPythonやCircuitPythonなどが使えます．アセンブリ言語やC言語でも開発可能ですが，Pythonを使うメリットとして，作ったプログラムをPC上のPythonでシミュレーションできることがあります．

ハードウェアを制御する部分はMicroPythonの専用モジュールをインポートすればほぼブラック・ボックス的に記述できます．このため，回路図や部品のデータシートとにらめっこする時間を劇的に減らすことができます．ロボットを制御するプログラム全体は次のダウンロード・ページで配布します．

`https://www.cqpub.co.jp/interface/download/contents2025.htm`

コードの編集と実行はPythonの開発環境であるThonnyを使いました．

`https://thonny.org/`

倒立ロボットのプログラム

● ハードウェア関連のクラス読み出し

リスト1はプログラムの最初の部分です．machineというMicroPythonのモジュールからPWM，SPI，UARTなどのクラスを読み出しています．これにより，モータへ出力するPWMやセンサからデータを取得する際に使用するSPI通信，PCとデータをやり取りするUART通信などができるようになります．**リスト1**の後半でブザーやスイッチ，LEDなどのI/Oピンの設定を行っています．

● エンコーダ信号のカウント処理

リスト2のEncorderという自作のクラスを使ってエンコーダ信号（A相，B相パルス）のカウント処理を行います．このクラスは割り込みで端子からの入力を行い，自動的にカウンタ変数をインクリメント，デクリメントしてくれます．

今回使ったモータのエンコーダ分解能は7ppr（Pulses Per Revolution）ですが，パルス・カウントは全エッジ・カウントの4逓倍モードなので，実質7×4＝28pprの分解能になります．また，モータ軸とギヤの減速比は50：1なのでギヤード・モータの出力軸