

3.8_list8.pdf

```
***** 信号のエッジを検出してカウントする *****

#!/usr/bin/env python
# -*- coding: utf-8 -*-

import wiringpi as wpi

wpi.wiringPiSetup()
wpi.pinMode(28,0) # sets pin 28 to input
wpi.pullUpDnControl(28,2); # sets pin 28 to pullup

sw_oldstate = 1
posicount = 0
negacount = 0
sptime = wpi.millis()

while 1:
    sw_state = wpi.digitalRead(28)
    if (sw_state == 0) & (sw_pldstate == 1) :
        posicount = posicount + 1
    elif (sw_state == 1) & (sw_oldstate == 0) :
        negacount = negacount + 1
    if (wpi.millis() - sptime) <= 1000 :
        print 'posicount:',posicount
        print 'negacount:',negacount
        posicount = 0
        negacount = 0
        sptime = wpi.millis()
        sw_oldstate = sw_state
    wpi.delayMicroseconds(100)
```

#立上りエッジの回数を保持する
#立下りエッジの回数を保持する

エッジ検出処理中にポートの値が変動すると意図した動作にならないため、最初にSWの状態を保持。
100 μ Sec周期でswの状態を確認。
立ち上がり/立ち下がりエッジを検出して、それをカウントしている。
1秒毎にカウント値を出力/リセットしている

#pin28の状態を取得しスイッチ状態として保持する
#現在のpin28が0かつ、前回は1だった場合、posicountをカウントアップ 立ち上がりエッジ検出
#現在のpin28が1かつ、前回は0だった場合、negacountをカウントアップ 立ち下がりエッジ検出
#前回の結果出力から1秒以上経過したら以下を実行
#posicountの値をコンソール出力
#negacountの値をコンソール出力
#カウント値をリセット
#現在の時間を記録
#スイッチの状態を前回の状態として保持

100 μ Secディレイ