

3.7_list7.pdf

```

***** 信号のエッジを検出して間隔を測定する *****

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

import wiringpi as wpi

wpi.wiringPiSetup()
wpi.pinMode(28,0)          #pin28を入力に設定する
wpi.pullUpDnControl(28,2); #pin28をプルアップに設定する

sw_oldstate = 1            #スイッチ入力の前回の値を保持する
positime = 0               #立上りを検出した時の時間を保持
negatime = 0               #立下りを検出した時の時間を保持

while 1:
    sw_state = wpi.digitalRead(28)

    if (sw_state == 0) & (sw_oldstate == 1) :
        negatime = wpi.micros()
        print "HIGH pluse width:",(negatime - positime)

    elif (sw_state == 1) & (sw_oldstate == 0):
        positime = wpi.micros()
        print "LOW pluse width:",(positime - negatime)

    sw_oldstate = sw_state
    wpi.delayMicroseconds(100)

wpi.microsはwiringPiSetupを実行してからの経過時間を取得することができる

```

このプログラムは100mSec周期でスイッチの状態を確認している
100mSec前の状態と異なる場合にエッジを検出している

whileループの最初にpin28の状態を保持している。if文の処理中にポートの値が変動するのを防いでいる

#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(つまりスイッチON)かつ、前回pin28が1(つまりスイッチがOFFだった)場合、以下を実行
#立下りエッジを検出したとみなし、その時の時間を保存
#立下りを検出した時間から、立ち上がりエッジを検出した時間を引き算する事で、Hiの時間をコンソールに出力する
#同様に現在のpin28が1で前回は0なら、立ち上がりエッジを検出したとみなし、Lowの時間を出力

#スイッチの状態を前回の状態として保持
#100μSecの待ち ⇒ 100μSec周期でエッジ検出を繰り返す