

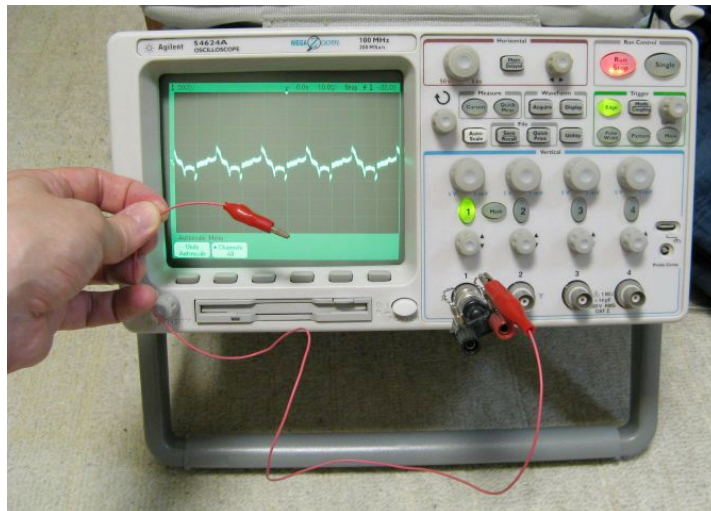
■付録基板のプッシュ・スイッチSW1読みとり時のノイズ

●裸の付録基板はノイズに注意

付録基板は表面に部品や端子、配線パターンが露出しています。意図的に47k Ω という大きなプルダウン抵抗を使っていることもあり、付録基板を裸のまま使っていると基板の配線パターンや部品に触れた指から伝わるノイズのためにうまくプッシュ・スイッチSW1の状態を読みとれない場合があります。

どのようなノイズかというとは商用電源AC100Vライン（50Hzまたは60Hzの正弦波）からの誘導雑音です。SW1を押そうとした時に、指先がプリント基板の配線パターンや部品に触れると人体を通じてAC100Vの商用電源ラインから50Hz/60Hzのノイズ成分が入力ポートに加わってしまいます。（図A）

スイッチ入力回路のプルダウン抵抗が小さければ誘導ノイズのレベルも小さくなり、その影響は無視出来ます。しかし、付録基板のプルダウン抵抗値は意図的にかなり大き目の47k Ω としているので、スイッチがオフの状態を入力ポートに加わる雑音電圧が入力信号のHi/Loを判別する閾値2.5Vを越えると、本来入力レベルがLoレベル（スイッチ・オフ）であるはずのものをHiレベル（スイッチ・オン）と誤検出してしまうことになります。（プルダウン抵抗を47k Ω と大きな値にしているのは、10k Ω 程度のプルアップ抵抗を外付けして、プルアップ入力端子として使用することも出来るようにしたためです）



図A AC100V商用電源ラインからの誘導ノイズ波形
オシロスコープの入力につないだリード線の絶縁被覆を指でつまんでいるだけで
約1Vpp（波形の振幅が約1V）のノイズを拾ってしまう。

リードの先端の金属端子（みのむしクリップ）を直接指でさわるとノイズ・レベルは軽く5Vppを越える。

●シールド・ケースに入れた機器ではプルダウン抵抗が大きくても問題は無い

きちんとシールドされたシャーシの中に入っている機器ではプルダウン抵抗の値が47k Ω でも問題となるようなことはありませんので安心して下さい。AC100Vラインからの誘導ノイズが混入する可能性があるのは、裸の基板やソルダーレス・ブレッドボードを使ったバラックセットで実験する時だけです。指が配線や部品に接触することが無ければ問題ありません。

●付録基板のプッシュ・スイッチSW1は爪楊枝などで押してください

付録基板でノイズ問題を回避する一番簡単な方法は、プッシュ・スイッチSW1を押すときに基板上の配線パターンや部品に触れないようにすることです。指ではなく爪楊枝などを使ってスイッチを押せば安心です。（爪楊枝の尖った先ではなくお尻の側で押してください）

●値の小さなプルダウン抵抗追加でも対処可能

きちんと回路的に対策するならば、47k Ω のプルダウン抵抗と並列に10k Ω 程度の抵抗を外付けすれば良いでしょう。こうすれば指でSW1を押そうとして、付録基板上の配線パターンや部品に触れてしまっても安心です。

●ソフト的なノイズ対策も可能

ソフト的にノイズ対策をすることも可能です。10ms～50ms程度の間隔で複数回入力ポートの状態を読みとって、読みとった値がすべてHiレベル（5V）であればプッシュ・スイッチSW1はオン、それ以外の場合はSW1はオフと判定するようにすればノイズの問題を回避出来ます。このような複数回のデータ読み込みはチャタリング除去にも用いられる手法です。

実際に複数回読み取りの効果をデモンストレーションするサンプル・プログラムnoise1.cのソースを図Bに示します。このプログラムを起動すると緑色のLEDが点灯します。SW1を押すと緑のLEDが消えて赤色のLEDが点きます。

まずSW2をRUNポジションにしてプログラムを起動してください。この状態ではノイズ対策のためにSW1の状態をセンスするのに複数回読み取りをおこなっています。SW1は押さずに、基板のSW1付近の端子や部品を指で触っても誤検出はしません。（緑色のLEDが点灯したまま）次にSW2をLDポジションにして、SW1は押さずに端子部分を触ってみてください。この状態では一度だけしかSW1の状態読み取りをしませんので、指で端子部分を何度か触っているとノイズによる誤検出が生じて赤色のLEDが点灯するはずですが、

もし皆さんが作成されたプログラムでノイズによるスイッチ読み取り誤りが問題になるようなことがあった場合は、noise1.cを参考に対策をおこなってください。なお、付録基板搭載のdsPIC30F2012に書き込まれているブートローダーもプッシュ・スイッチSW1の状態読み取り時にはソフト的なノイズ対策をしています。

●8月号掲載プログラムではノイズ対策は不要です

8月号の記事に掲載した付録基板のサンプル・プログラムでは、プッシュ・スイッチSW1読み取り時のノイズ対策は不要です。理由は、8月号のプログラムでは読みとったSW1の状態によって分岐するような処理はしていないからです。（単にSW1を押している間、点灯するLEDの色を変えたりしているだけなので、間欠的な誤読み取りがあっても問題ない）

```

1: // switch read test program
2:
3: #include <p30F2012.h>
4:
5: //-----
6: // read SW1 only once
7:
8: unsigned sw1status1() {
9:     return _RD9;          // read SW1
10: }
11:
12: //-----
13: // read SW1 8 times
14:
15: unsigned sw1status2() {
16:     unsigned i, j, x;
17:
18:     x=0;
19:     // read SW1 8 times
20:     for (i=0; i<8; i=i+1) {
21:         x=(x<<1)+_RD9;
22:         for (j=0; j<1000; j=j+1) { asm("nop"); asm("nop"); }
23:     }
24:     if (x==0xff) {
25:         return 1;
26:     } else {
27:         return 0;
28:     }
29: }
30:
31: //-----
32:
33: void main(void) {
34:     _TRISD8=1;                // RD8 is input (slide swithc, SW2)
35:     _TRISD9=1;                // RD9 is input (push switch, SW1)
36:     _TRISC13=0;               // RC13 is output (red/green LED)
37:
38:     _RC13=0;
39:     while(1) {                // forever loop
40:         if (_RD8==0) {        // if sw2='LD'
41:             if (sw1status1()==1) _RC13=1; // read SW1 once
42:         } else {              // sw='RUN'
43:             if (sw1status2()==1) _RC13=1; // read SW1 8 times
44:         }
45:     }
46: }

```

図B スイッチの複数回読み取り処理のサンプル・プログラムnoise1.c

プッシュ・スイッチSW1を一度押すと点灯するLEDの色が緑から赤に変わる。

スライド・スイッチSW2がRUNポジションの時は、関数sw1status2()でSW1を8回連続して読みとり、8回すべてがHiレベルの時のみSW1が押されたと判定する。

SW2がLDの時は、関数sw1statsu()でSW1の状態を1回だけ読みとる。