AVRp00：汎用入出力ﾎﾟｰﾄの入出力形態設定

序説

一部の例外を除いて、AVRの汎用ﾎﾟｰﾄは双方向です。通常、これらは応用目的に従って、ﾌﾟﾛｸﾞﾗﾑの先頭で入出力の方向指定と出力値の設定(初期化)が行われます。多くの資料で、この手順を次のように記載しています。

```
;       * 方向設定⇒初期値設定の順でﾎﾟｰﾄBの全ﾋﾞｯﾄを出力に設定 *

        LDI     R16, 0b11111111     ; 出力指定値を取得
        OUT     DDRB, R16           ; ﾎﾟｰﾄB全ﾋﾞｯﾄ出力設定
        LDI     R16, 0b00001111     ; 初期値を取得
        OUT     PORTB, R16          ; ﾎﾟｰﾄB出力値初期化
```

この例はﾎﾟｰﾄBの全ﾋﾞｯﾄを出力として使用し、初期値は上位4ﾋﾞｯﾄが0(Low)で、下位4ﾋﾞｯﾄが1(High)です。このように方向設定と初期値設定の順番が基本的に思考順と一致していますので特に問題を感じないかもしれません。

これに反して、逆の設定順も有り得ます。即ち、初期値設定後に方向設定を行う方法です。

```
;       * 初期値設定⇒方向設定の順でﾎﾟｰﾄBの全ﾋﾞｯﾄを出力に設定 *

        LDI     R16, 0b00001111     ; 初期値を取得
        OUT     PORTB, R16          ; ﾎﾟｰﾄB出力値初期化
        LDI     R16, 0b11111111     ; 出力指定値を取得
        OUT     DDRB, R16           ; ﾎﾟｰﾄB全ﾋﾞｯﾄ出力設定
```

一見両者間で特段の違いがないように思えます。然しながら、両者に於ける初期化時のﾎﾟｰﾄ出力変化を考察すると明らかな違いがあります。**図1.**は前者の順、**図2.**は後者の順での結果を表し、共に(1)は0(Low)を、(2)は1(Hihg)を初期値とする場合を示します。両者間の違いは次の通りです。

- 方向設定⇒初期値設定の順の方が2ｸﾛｯｸ早く出力に切り替わります。
- 方向設定⇒初期値設定の順で初期値が1(Hihg)の場合に2ｸﾛｯｸ間0(Low)が出力されます。

図1. 方向設定⇒初期値設定の順で出力として使用するﾎﾟｰﾄの電源投入時ﾎﾟｰﾄ初期化

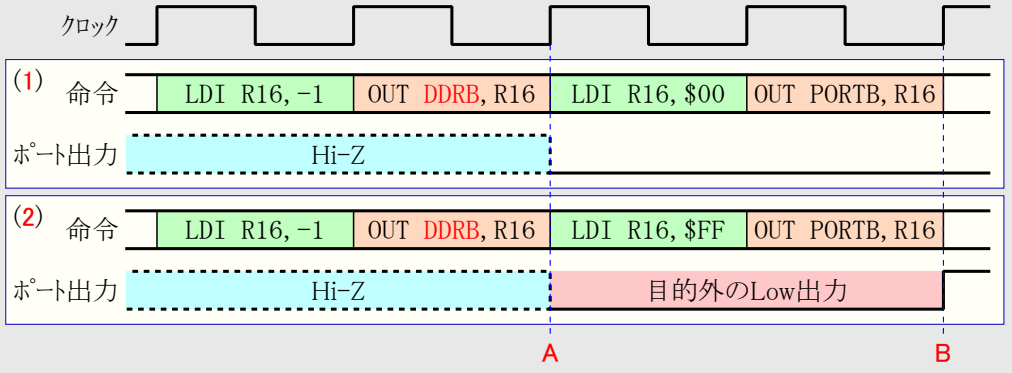
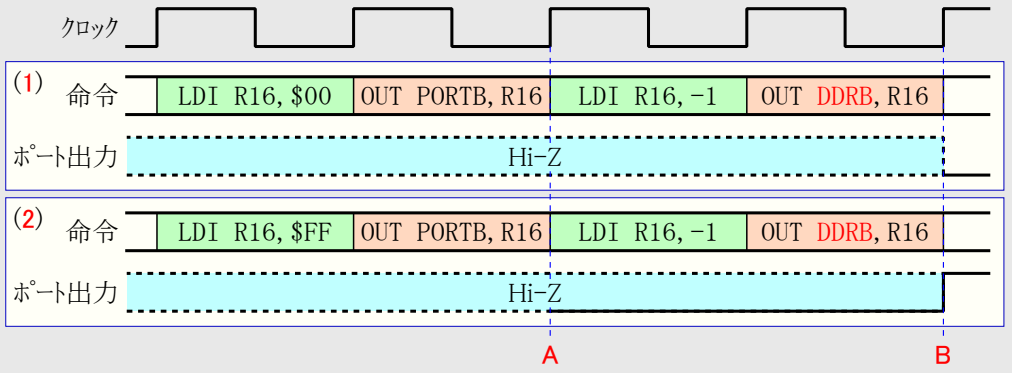


図2. 初期値設定⇒方向設定の順で出力として使用するﾎﾟｰﾄの電源投入時ﾎﾟｰﾄ初期化



HERO and heavy friends

8ﾋﾞｯﾄ AVR® マイクロ コントローラ

応用記述

Rev. AVRs00-12/13

前頁の図1.と図2.間で明らかなように、入力から出力への切り換わりが方向設定⇒初期値設定順ではA点であるのに対し、初期値設定⇒方向設定順ではB点となり、方向設定⇒初期値設定順の方が2ｸﾛｯｸ早く変更されることが判ります。この違いは一般的に問題視されないでしょう。

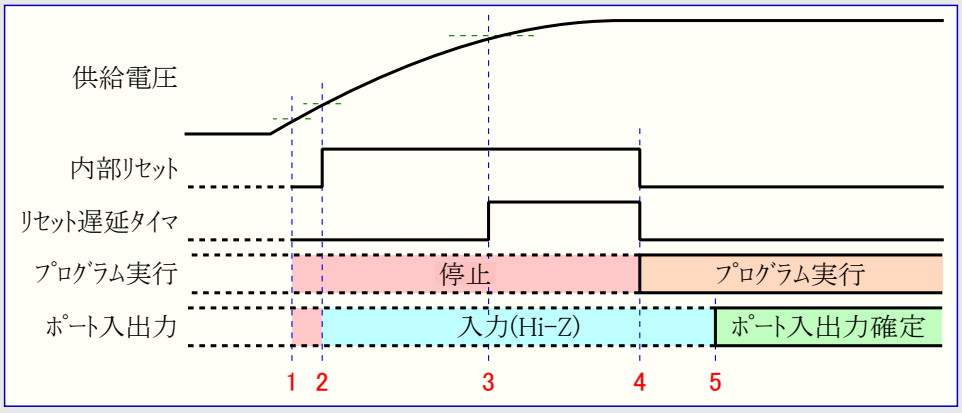
方向設定⇒初期値設定の順で初期値が0(Low)の場合(図1.の(1)参照)は全く問題なく、前記の切り換え応答時間も考慮すると、この場合は方向設定⇒初期値設定の順での初期化が最善でしょう。けれども、初期値が1(Hihg)の場合は図1.のA点からB点間で示されるように、目的外の0(Low)出力が2ｸﾛｯｸ間出現します。これはその出力ﾎﾟｰﾄへの接続物に依存して問題になるかもしれません。

ﾎﾟｰﾄ初期化

ここでﾘｾｯﾄ直前を含めて、CPUの動きを簡単に考察します。電源投入時からの遷移は次のようになります。

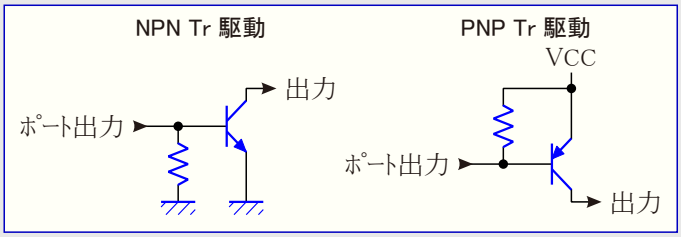
1. 0Vからの供給電圧上昇が電源ONﾘｾｯﾄ(POR)開始電圧に達する前に、CPUのCMOS論理回路が動作可能な電圧に達すると、不定状態が発生します。一般的にCPUはPOR回路が先行して動作するように考慮されているので、この状態は無視できます。
2. 0Vからの供給電圧上昇が電源ONﾘｾｯﾄ(POR)開始電圧に達すると、CPUの内部ﾘｾｯﾄが有効になり、入出力ﾎﾟｰﾄは既定の初期値になります。
3. 供給電圧が更に上昇して電源ONﾘｾｯﾄ(POR)終了電圧または低電圧検出(BOD)ﾘｾｯﾄ電圧に達すると、ﾘｾｯﾄ遅延ﾀｲﾏが起動されます。
4. ﾘｾｯﾄ遅延ﾀｲﾏの計時完了で内部ﾘｾｯﾄが解除され、ﾌﾟﾛｸﾞﾗﾑの実行が開始されます。
5. 一般的にﾌﾟﾛｸﾞﾗﾑの先頭でﾎﾟｰﾄの初期化が行われます。

図3. 電源投入からﾎﾟｰﾄ初期化までの流れ



ﾘｾｯﾄ開始(図3.の1)からﾎﾟｰﾄ初期化完了時(図3.の5)までの間は各ﾎﾟｰﾄが入力なので、各ﾋﾟﾝはHi-Z状態です。これはﾎﾟｰﾄ出力に接続される回路に於いて、その回路の入力が開放状態と等価であることを意味します。通常、開放入力が許されないので、このような入力はﾌﾟﾙｱｯﾌﾟまたはﾌﾟﾙﾀﾞｳﾝ抵抗器によって、駆動部出力がない場合の入力値固定化が行われます。ﾎﾟｰﾄがTrを駆動する時、換言するとﾎﾟｰﾄ出力に接続される回路の入力がTrの場合の例を右に示します。

図4. 代表的なTr入力例



ﾘｾｯﾄ開始からﾎﾟｰﾄ初期化完了時までの時間に対し、一般的に対応回路の動作が充分に遅い場合、または特に問題とならない場合には、このﾌﾟﾙｱｯﾌﾟまたはﾌﾟﾙﾀﾞｳﾝ抵抗器を省略する場合があります。 例えば、Trの出力に繋がっている対象物の応答速度が50msで、CPUのﾘｾｯﾄ時間が10msとした場合、概ね10ms間の不安定状態では対象物が応答しないので、無視することができます。但し、現実には次の点も考慮しなければなりません。

- 電源投入時の場合はCPUが確実にﾘｾｯﾄ状態へ移行する前に、或る程度の電圧が印加されている状態が存在します。勿論、ｼｽﾃﾑによってはﾘｾｯﾄ時間中も電源電圧が上昇しているかもしれません。この場合は電源電圧と応答時間の両者の関係から、対象物の駆動部が対象物を動かさない、または対象物自体が動かないことを保証しなければなりません。
- 電源供給中にﾘｾｯﾄが発生する可能性(例えばWDT)がある応用では、そのﾘｾｯﾄへの移行直前の対象物の状態を考慮しなければなりません。例えば電源投入時は対象物の初期応答が充分に遅いために問題とならなくても、定常状態の応答がより速い場合は電源供給中に起こる(例えばWDT)ﾘｾｯﾄに反応するかもしれません。

そこでﾎﾟｰﾄ初期化ﾌﾟﾛｸﾞﾗﾑを検討すると、やはり方向設定⇒初期値設定の順で初期値が1(Hihg)の場合に問題となるのは明らかで、殆どの場合で、この手順を使用すべきではありません。基本的に初期値設定⇒方向設定の順が推奨されます。希望する初期値が0(Low)の場合にだけ、方向設定⇒初期値設定の順が使用でき、より速い出力切り替えが必要な場合に有効でしょう。

入出力変更

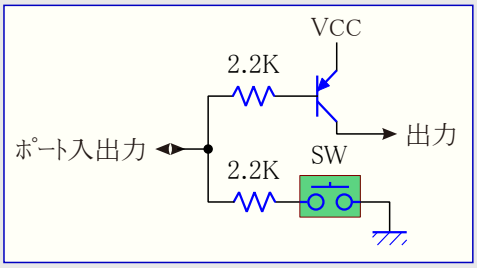
多くの応用ではﾎﾟｰﾄの入出力方向がﾘｾｯﾄ直後の初期設定から変更されないでしょう。特定の応用では動作中にﾎﾟｰﾄの方向を変更しなければならないことがあります。例えば、外部ﾒﾓﾘ支援を持たないﾃﾞﾊﾞｲｽで外部ﾒﾓﾘを使用する場合に、外部ﾒﾓﾘのﾃﾞｰﾀに使用されるﾎﾟｰﾄは外部ﾒﾓﾘの読み書きに応じて入力または出力に切り換えなければなりません。このような動作中の方向変更は、前記の「序説」項で記述したようなﾎﾟｰﾄ方向ﾚｼﾞｽﾀとﾎﾟｰﾄ出力ﾚｼﾞｽﾀ間の設定順によって問題が起き得ます。

ﾎﾟｰﾄ入出力切り換えを必要とする応用の殆どでは、入出力切り換えに伴う入出力間での衝突を回避する何らかの対策が講じられているでしょう。例えば上記の外部ﾒﾓﾘの場合は、外部ﾒﾓﾘのﾃﾞｰﾀ線の方向を切り換える信号線があり、ﾎﾟｰﾄ入出力を切り換えるのに先立って外部ﾒﾓﾘのﾃﾞｰﾀ線の方向を切り換えることによって衝突を回避できます。このような場合にはﾎﾟｰﾄ入出力切り替えの手順が問題になることはないでしょう。

ﾎﾟｰﾄ入出力切り替え手順を考察するために図5.の回路を用います。通常この回路はｽｲｯﾁ入力に対応してTrがON/OFFし、CPUは内部ﾌﾟﾙｱｯﾌﾟ付き入力とすることでｽｲｯﾁ入力を読むことができます。CPUは必要な時にﾎﾟｰﾄを出力に切り換えることによってｽｲｯﾁの状態に拘らずTrをON/OFFできます。TrをOFFする時にTrのﾍﾞｰｽをVCCに引き上げるのにCPUの内部ﾌﾟﾙｱｯﾌﾟを利用していることに注意してください。これは本項での問題点を顕著に表すためで、一般的に推奨される方法ではありません。

以下の考察に於ける変更直前の状態はﾎﾟｰﾄ方向ﾚｼﾞｽﾀ値が0(入力)、ﾎﾟｰﾄ出力ﾚｼﾞｽﾀ値が1(ﾌﾟﾙｱｯﾌﾟ付き入力)です。各種の組み合わせが図6.～9.で示されますが、ﾎﾟｰﾄﾋﾟﾝの実線は黒がﾎﾟｰﾄ出力による値で、赤はLowの場合がｽｲｯﾁON、Highの場合が内部ﾌﾟﾙｱｯﾌﾟによる値を示します。また、( )付きの"SBI PORTB,0"命令は結果的に直前値と同じ値を上書きすることになるので、省略できます。

図5. 入出力切り換え用参照回路例



入力から出力への変更

入力から出力への設定変更に関して、方向変更⇒出力値設定の順で行った場合のﾀｲﾐﾝｸﾞが図6.で、出力値設定⇒方向変更の順で行った場合のﾀｲﾐﾝｸﾞが図7.で示されます。両図共に(1)と(2)はｽｲｯﾁがONされている時の切り換えを示し、(3)と(4)はｽｲｯﾁがOFFされている時の切り換えを示します。同じく切り換え時に期待されるTr動作は、(1)の場合がON(Low)継続、(2)の場合がON(Low)⇒OFF(High)、(3)の場合がOFF(High)⇒ON(Low)、(4)の場合がOFF(High)継続です。

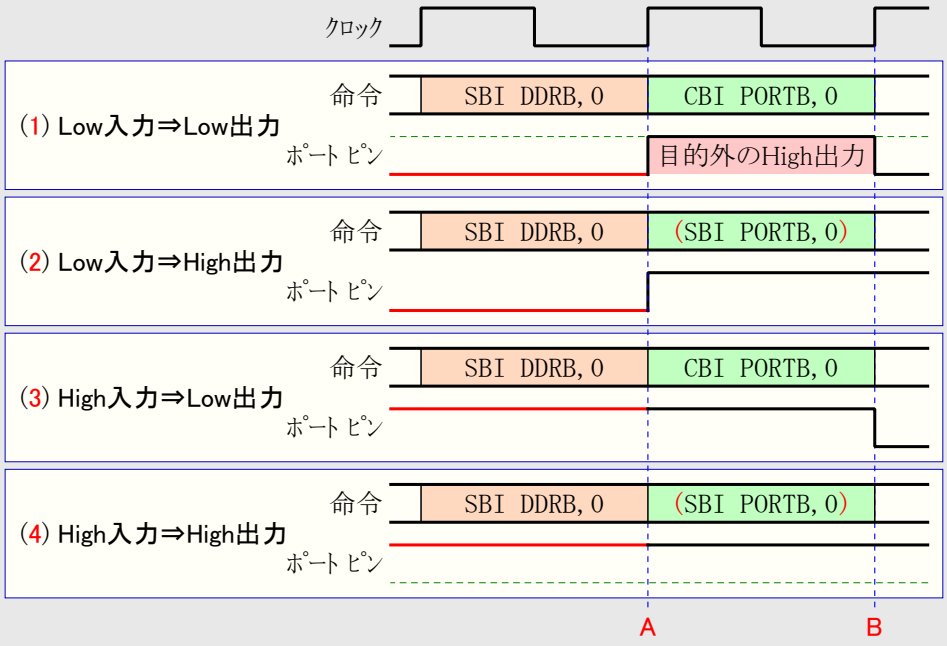
方向変更⇒出力値設定順での入力から出力への変更

(4)の場合は全く問題ありません。(2)と(3)の場合も切り換わり点がAとBで位置が異なるだけで特に問題はありません。(1)の場合はAとBの間で1ｸﾛｯｸ幅の意図せぬ1(High)が出力されて、Trを一瞬OFFしてしまいます。これは内部ﾌﾟﾙｱｯﾌﾟを有効とするためにﾎﾟｰﾄ出力ﾚｼﾞｽﾀ値が1の状態で、ﾎﾟｰﾄ方向ﾚｼﾞｽﾀを出力(1)に設定したことによります。

応用によっては意図せぬ出力が1ｸﾛｯｸ幅なら無視できるかもしれません。けれどもその場合は方向変更と出力値設定を行う命令が連続して実行されることを保証しなければなりません。例えばこの命令間で割り込みが処理されると、予期せぬ時間に渡って意図せぬ値が出力されます。

従って、この場合はLow入力時にLowを出力する切り換えに、方向変更⇒出力値設定順での変更方法が使用できないことを意味します。

図6. 方向変更⇒出力値設定の順で、入力から出力への切り換え



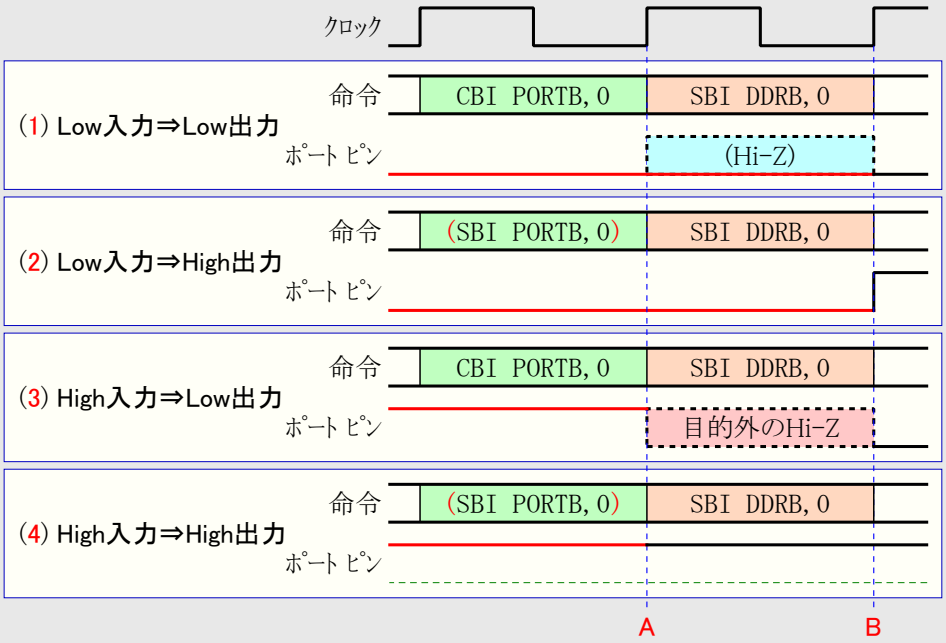
出力値設定⇒方向変更順での入力から出力への変更

(4)の場合は全く問題ありません。(2)の場合も切り換わり点がB位置であることを除いて特に問題はありません。(1)の(3)の場合はAとB間で1ｸﾛｯｸ幅分の意図せぬHi-Zが出現します。一見、(1)の場合はｽｲｯﾁがLowに引いているので実害がなさそうですが、AとBの間でｽｲｯﾁが開放された場合に(3)と同じ状態になります。これは方向を切り換える前に内部ﾌﾟﾙｱｯﾌﾟを解除した結果です。故にLow出力の場合にだけ発生しています。

応用によっては(1)と(3)の問題を無視できます。一般的に入出力容量や浮遊容量などが存在するために、例えHi-Zになったとしても直ぐにその影響が現れません。影響が現れるには一定の時間を必要とし、その時間は前記の容量と入力ｲﾝﾋﾟｰﾀﾞﾝｽなどに依存します。図5.の回路例で(3)の場合は容量を放電する大きな要素がTrのﾍﾞｰｽ入力ｲﾝﾋﾟｰﾀﾞﾝｽだけなので、概ね1MHz程度以上のｼｽﾃﾑ ｸﾛｯｸなら問題にならないでしょう。より低いｼｽﾃﾑ ｸﾛｯｸでは、これが問題となる可能性が増します。勿論、この場合も出力値設定と方向変更を行う命令が連続して実行されることが前提です。

従って、この場合はLowを出力する切り換えに出力値設定⇒方向変更順での変更法を用いない方が良いことを意味します。

図7. 初期値設定⇒方向設定の順で、入力から出力への切り換え



出力から入力への変更

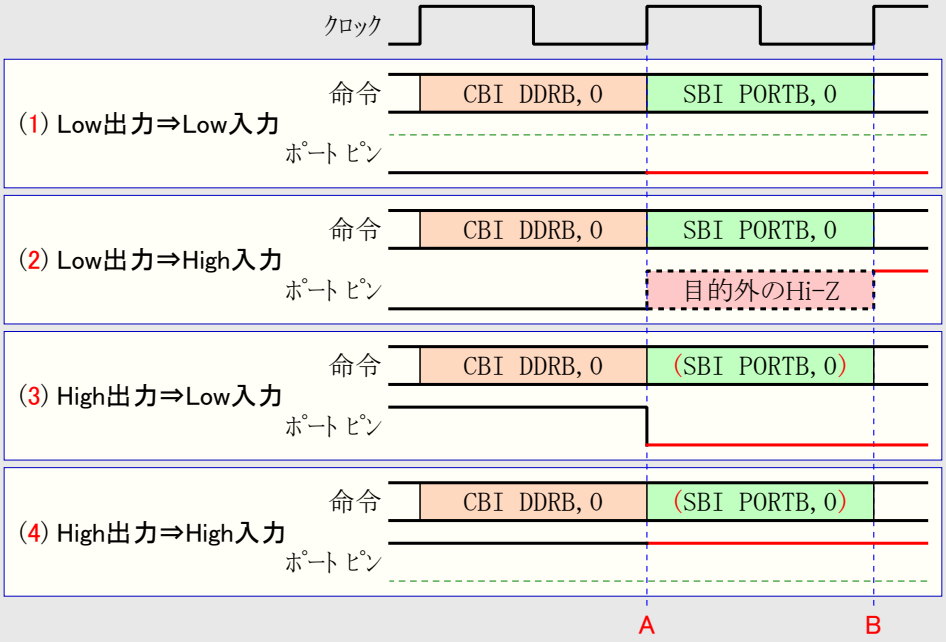
出力から入力への設定変更に関して、方向変更⇒出力値設定の順で行った場合のﾀｲﾐﾝｸﾞが図8.で、出力値設定⇒方向変更の順で行った場合のﾀｲﾐﾝｸﾞが図9.で示されます。両図共に(1)と(3)はｽｲｯﾁがONされている時の切り換えを示し、(2)と(4)はｽｲｯﾁがOFFされている時の切り換えを示します。同じく切り換え時に期待されるTr動作は、(1)の場合がON(Low)継続、(2)の場合がON(Low)⇒OFF(High)、(3)の場合がOFF(High)⇒ON(Low)、(4)の場合がOFF(High)継続です。

方向変更⇒出力値設定順での出力から入力への変更

(1)、(3)、(4)の場合は全く問題ありません。(2)の場合はAとBの間で1ｸﾛｯｸ幅分の意図せぬHi-Zを生じます。これは条件によって問題になりますが、多くの応用では問題にならないでしょう。これは前記のシステム クロック依存の件と基本的に同じです。その場合は長時間のHi-Z状態がCPUをﾗｯﾁｱｯﾌﾟさせる要因になるかもしれません。

従って、この場合はLow出力時にHighを入力する切り換えに、方向変更⇒出力値設定順での変更方法を用いない方が良いことを意味します。

図8. 方向変更⇒出力値設定の順で、出力から入力への切り換え

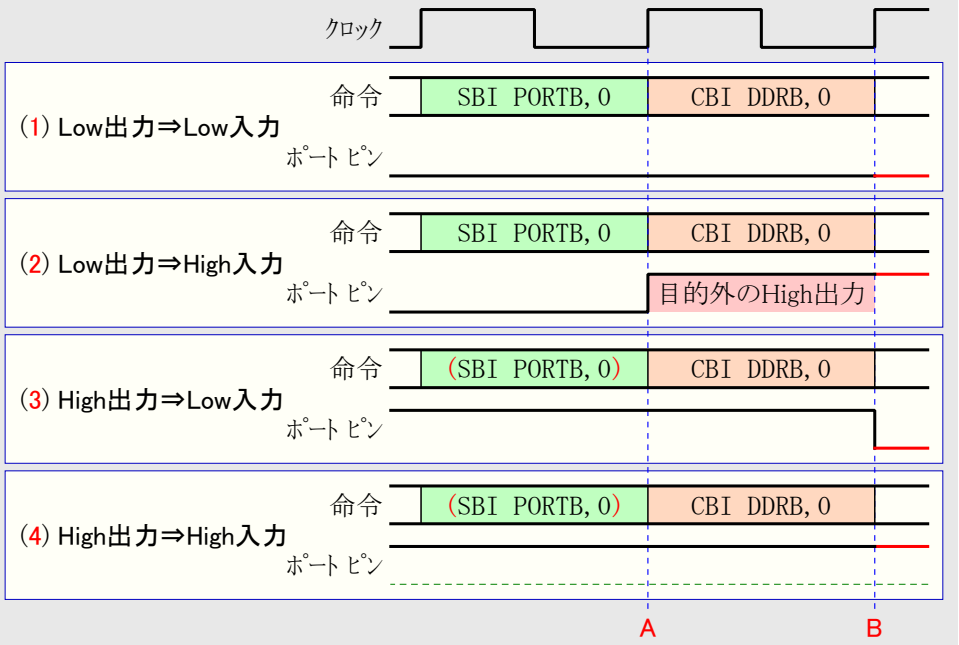


出力値設定⇒方向変更順での入力から出力への変更

(1)、(3)、(4)の場合は全く問題ありません。(2)の場合はAとBの間で1ｸﾛｯｸ幅分の意図せぬ1(High)が出力されますが、期待する値も1(High)なので、基本的に問題とならないでしょう。けれども、この区間で外部がLow駆動する可能性がある場合は大きな問題です。

従って、この場合はLow入力時にHhgを出力する切り換えに、出力値設定⇒方向変更順での変更方法が使用できないことを意味します。

図9. 初期値設定⇒方向設定の順で、出力から入力への切り換え



纏め

これまでの検討結果として、最善の入出力切り換え方法は条件に応じて異なることが判ります。それでは変更直前の状態を読み取ってからそれに応じた方法を選択すれば良いのかと言うと、それは現実的に無意味です。前述の考察は基本的に変更中の入力変化がないとの条件で行われています。現実的にも、読み取った変更直前の値は、実際の変更時に異なっているかもしれません。

これは入力値の変化点が出力変化点と完全に一致することが不可能で、且つCPUが入力変化点を制御できないため、現実的な検討では次のように考察しなければならないことを意味します。

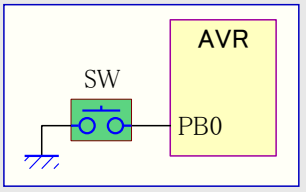
- 入力から出力への変更を検討する場合は、Low入力⇒Low出力とHigh入力⇒Low出力(図6.と図7.での各々(1)と(3))、Low入力⇒High出力とHigh入力⇒High出力(図6.と図7.での各々(2)と(4))の両組み合わせは、それらの組み合わせ内が同じ手順で変更されなければなりません。
- 出力から入力への変更の場合は、Low出力⇒Low入力とLow出力⇒High入力(図8.と図9.での各々(1)と(2))、High出力⇒Low入力とHigh出力⇒High入力(図8.と図9.での各々(3)と(4))の両組み合わせは、それらの組み合わせ内が同じ手順で変更されなければなりません。

従って、総合的な結果としては、より問題となる目的外の値出力に着目して、これが起こらない条件、即ち入力から出力への切り換えは出力値設定⇒方向変更の順、出力から入力への切り換えは方向変更⇒出力値設定の順が無難です。多くの応用に関してこの結果を適用できますが、個々の応用に対して前述の考察と同様な考慮が行われるべきです。

応用例

必然的に双方向通信を行う応用などを除いて、動作中にポートを意味を持って切り換える応用は少なく、殆どの応用は電源投入時のポート初期化以外にポートの方向を切り換えることがないでしょう。ここでは少し別の観点からポートの方向切切り換えの応用例を示します。この応用例は当然ポート方向切り換えを行ないますが、その目的はスイッチ入力の節電です。例とする回路を図10.に示します。

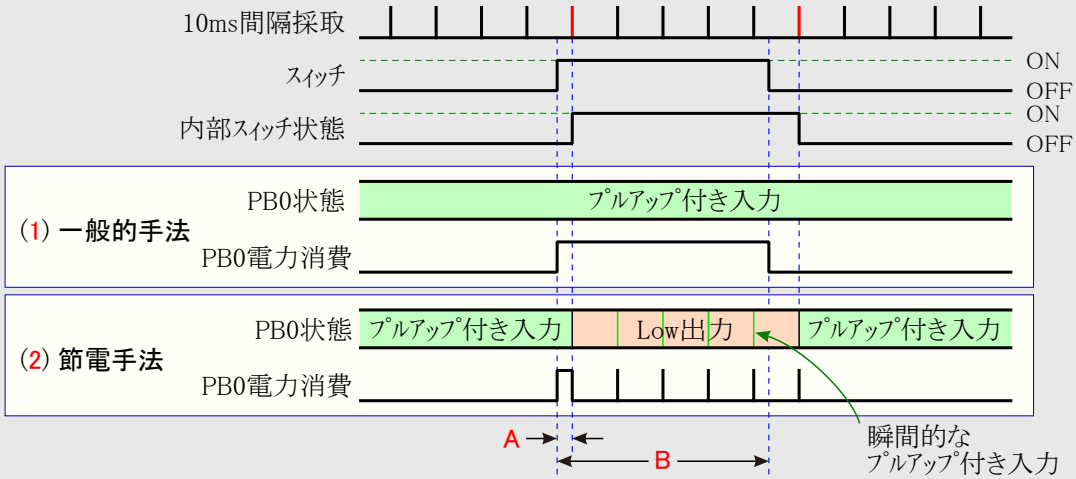
図10. 応用例回路



勿論基本的な使用法は内部プルアップ付き入力です。スイッチは10ms毎に採取され、ONの場合は何らの反応をするものとします。これは極ありふれた仕様です。ここでスイッチON時のプルアップ電流に注目します。スイッチがOFFの場合はピンの入力漏れ電流程度しか消費しませんが、ONの場合は概ねVCC(V)÷プルアップ抵抗(Ω)分の電流が流れ続けます。

これを10ms毎の採取でONを検出したなら、ポートをLow出力に切り換えて、以降の10ms毎に再びプルアップ付き入力に再設定してスイッチの状態を確認します。そこで未だONのままなら、またLow出力に切り換えて、以下同様にします。OFFへの復帰を検出した場合はプルアップ付き入力のままでスイッチ検出処理を抜けます。これによって初期状態と同じになります。

図11. 一般的なスイッチ採取と本例でのスイッチ採取の比較



上記の処理によってスイッチON時の内部プルアップ抵抗消費電流を大きく削減できます。これは図11.で示され、一般的手法ではスイッチONの間中(B区間)大きなプルアップ電流が流れますが、節電手法ではスイッチONの瞬間から次の10ms間隔時間までの間(A区間)と各10ms間隔処理での瞬間的なプルアップ付き入力切替区間でしか大きなプルアップ電流は流れません。この瞬間的なプルアップ付き入力切替区間は全体に大きく影響を及ぼさないので、以下の考察では無視しています。

仮に100ms間スイッチがONされた場合のON間の差を比較すると、VCC=5Vでの0V入力時の代表的なプルアップ抵抗電流が約140μAなので、一般的手法のこの区間の平均電流は約140μAです。節電手法に於ける最大プルアップ電流区間はスイッチONの瞬間から次の10ms間隔時間なので、その時間は最小が0ms、最大が10msで平均5msになり、ON区間のプルアップ抵抗消費電力比は5ms÷100ms=0.05で5%になり、故にこのON区間の平均プルアップ抵抗消費電流は140μA×0.05=7μAになります。

けれども、一般手法に対して、節電手法ではスイッチON区間内でポート方向切り換えを行う命令が余分に実行されます。休止形態を使用しない応用では無関係ですが、休止形態を使用している場合はこれらの命令による動作分だけ休止形態期間が短縮され、従って逆に消費電力を増加させます。これらの命令は基本的にスイッチON区間内の10ms間隔処理内で、その処理毎に4命令の追加になります。

この電力消費増加はシステム クロックに依存しますが、例えば10MHz動作と100kHz動作の場合は100倍の処理能力差があるので、仮に100kHzで100%動作とした場合、10MHzでは1%動作に相当し、99%の区間がパワーダウン動作にできることになり、これに関する吟味があまり意味を持たなくなります。参考までに100%動作での10MHzと100kHz動作に於ける増加分を計算すると、10MHz動作の場合は1周期命令が1s間に10,000,000回実行されるので、その1s間中の追加命令数は1000ms÷10ms×4命令=400命令、増加分は400÷10,000,000=0.00004で0.004%になります。仮にVCC= 5V,10MHzの標準動作消費電流が6mAとした場合は、6mA×0.00004=0.00024で0.24μAの増加になり、差し引き7μA−0.24μA=6.76μAの節電になります。100kHz動作の場合は1s間の総クロック数が100,000で、増加分は400÷100,000=0.004で0.4%になります。VCC=5V,100kHzの標準動作消費電流が0.1mAとした場合は、0.1mA×0.004=0.0004で0.4μAの増加になり、差し引き7μA−0.4μA=6.6μAの節電になります。

実際にポートを入力からLow出力に切り換える方法は基本がLow入力⇒Low出力ですが、チャタリングなどによって変更中にスイッチがOFFになる可能性があるため、High入力⇒Low出力も考慮しなければなりません(図6.と図7.の(1)と(3)参照)。けれども、その場合のHigh入力は外部から駆動されるHighではないので、本例の場合はどちらの方法でも問題なく使用できます。

ポートをLow出力から入力へ切り換える場合も同様に基本がLow出力⇒High入力ですが、Low出力⇒Low入力も考慮しなければなりません(図8.と図9.の(1)と(2)参照)。図9.の(2)の場合で、変更中にスイッチON状態が起こると短絡になるため、この方法は使用できません。従ってポートをLow出力から入力にする方法は方向変更⇒出力値(プルアップ)設定の順でなければなりません。

HERO and heavy friends

本書はAVRの応用に関する補助情報AVRp00の記録です。

青字の部分はﾘﾝｸとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。