

入力ポートのインターフェース

制御用マイコンの入力としては、デジタル信号とアナログ信号が考えられますが、本章ではデジタル信号との基本的なインターフェースについて説明します。

2-1 PICマイコンの入力ポートの特性

■ 入力ポートの構造と特性

● 等価的な回路

PICマイコンの入力ポートには、デジタル入力ポートとアナログ入力ポートがあります。ここでは前者の特性を説明します。デジタル入力ポートは前出の表1-1-1に示したようにTTLコンパチブル・バッファ付きの入力ピンと、シュミット・トリガ・バッファ付きの入力ピンがあります。これらの一例を図2-1-1に示します。

CMOSデバイスですから、入力ポートは高入力インピーダンスです。そのため同じ電源電圧系ならばインターフェース相手の回路から電流が流れることはありません。しかし、各I/Oピンには保護用ダイオードが入っていますから、高い電源系と不用意にインターフェースすると図2-1-2のように入力ピンを通じてPICマイコンへ電流が流れこんでしまいます。これらの保護用ダイオードは、等価回路や回路図ではたいてい省略されていますから注意してください。

また、高速なインターフェースでは、入力容量を無視できないことがあります。

● 入力特性

入力ポートはTRISレジスタの対応する各ビットを1に初期化することで機能します。入力ポートで判定される電圧レベルは、図2-1-3のようにTTLレベルとシュミット・トリガ・レベルによって異なります。

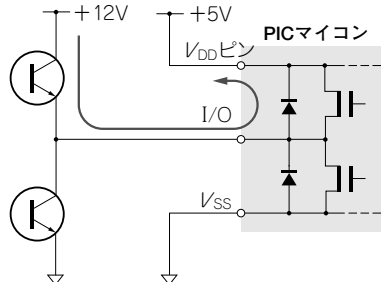


図2-1-2 高電源系と不用意にインターフェースすると入力ピンを通じてマイコン側電源へ電流が流れる

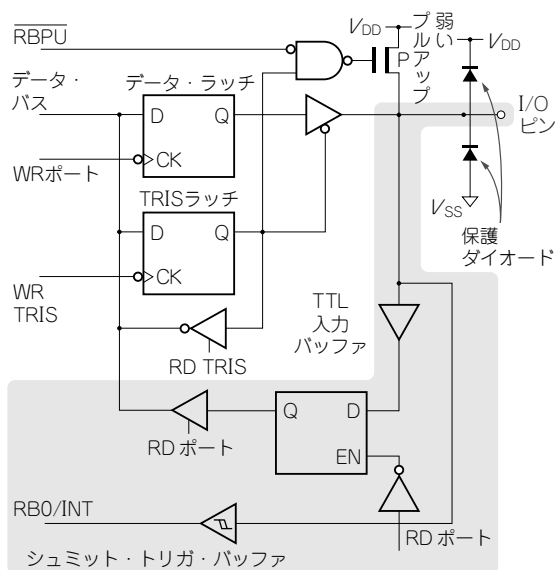
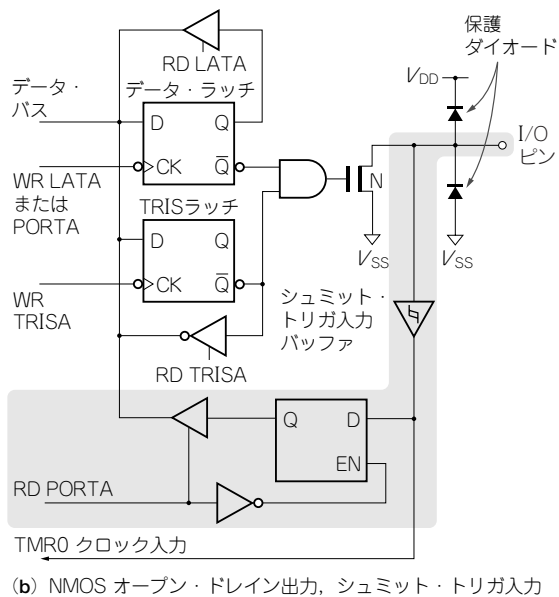
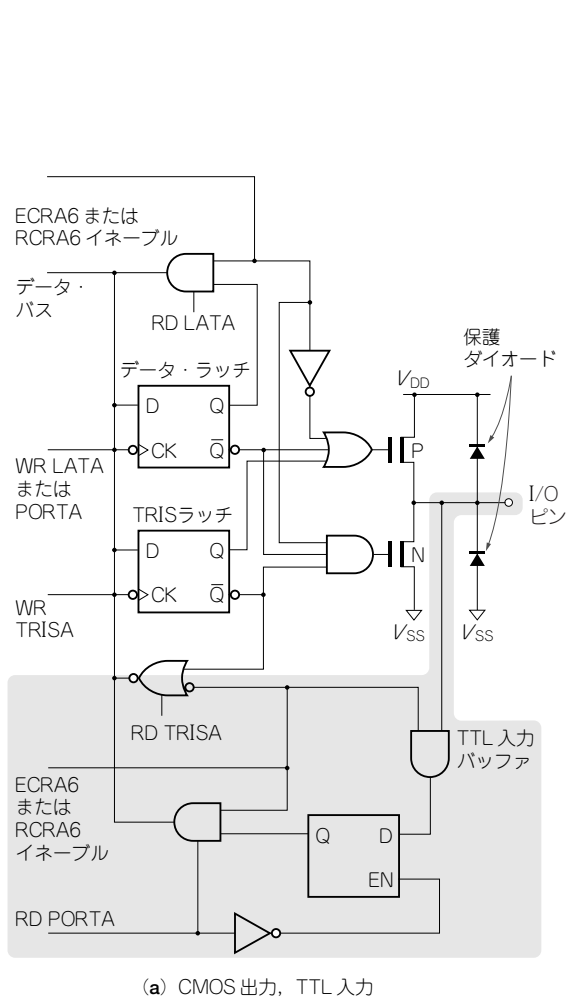


図2-1-1 PICマイコンのデジタル入力ポートの種類

図2-1-4と図2-1-5は実測例です。

どのポートがTTL入力なのかシュミット・トリガ入力なのかは、PICマイコンの品種によって割り当てが異なるので一概にはいえませんが、外部インターフェースの可能性が高いポートCやポートDはシュミット・トリガ入力であることが多いようです。

▶ TTL入力

電源電圧が5Vの場合は2V以上がHレベル、0.8V以下がLレベルです。ほかの電圧では電源電圧の

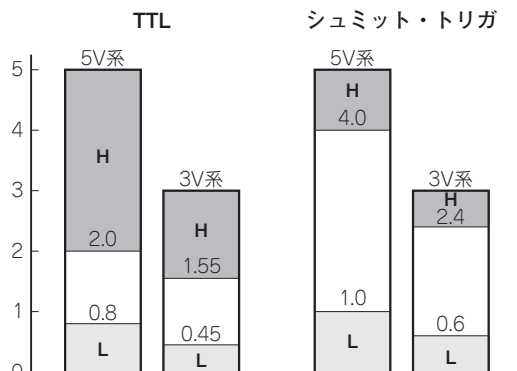
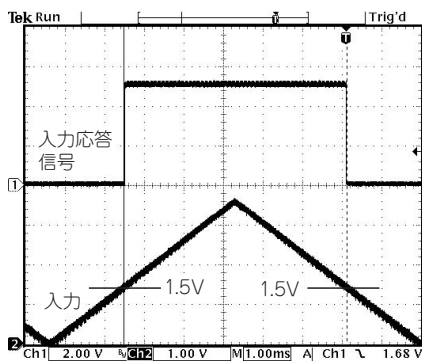
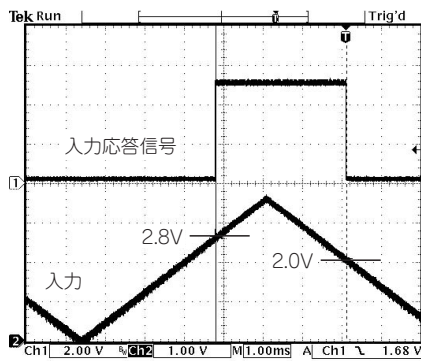


図2-1-3 TTL入力とシュミット・トリガ入力の判定レベル

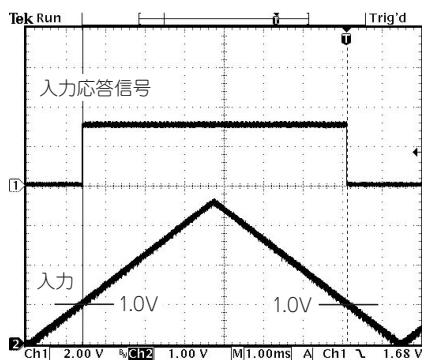


(a) TTL入力

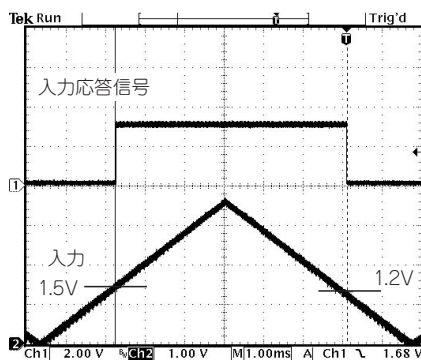


(b) シュミット・トリガ入力

図2-1-4 電源電圧5V時の入力特性(1ms/div., 上: 2V/div., 下: 1V/div.)



(a) TTL入力



(b) シュミット・トリガ入力

図2-1-5 電源電圧3V時の入力特性(1ms/div., 上: 2V/div., 下: 1V/div.)

25% + 0.8V以上がHレベルで、電源電圧の15%以下がLレベルとなるので、3V系ではそれぞれ1.55Vと0.45Vです。実測してみると5V電源では約1.5Vで、3V電源では約1Vで変化します。

▶シュミット・トリガ入力

電源電圧の80%以上がHレベルで、20%以下がLレベルになります。5V系ではそれぞれ4Vと1Vで、

3V系ではそれぞれ2.4Vと0.6Vです。実測してみると、5V電源ではL→Hが約2.8V、H→Lが約2Vです。3V電源ではL→Hが約1.5V、L→Hが約1.2Vで変化していました。

■ 入力ポートからデータを読み込む方法

ポートから数値データを読み込む方法は2種類あります。1バイト読み込む方法と、1ビットだけ読み込む方法です。

● 1バイト読み込み

▶ アセンブリ言語の記述例

```
MOVLW  H'FF'      ;ポートBを入力ポートに設定するためFFhをWREGへロード
MOVWF  TRISB     ;TRISBレジスタへ書き込む
MOVF   PORTB     ;ポートBから1バイト読み込む
COMF   PORTC     ;ポートCの各ビットを反転した1バイト読み込む
```

▶ C言語の記述例

```
TRISB=0xFF;      //ポートBを入力ポートに設定する
n=PORTB;         //ポートBから1バイト読み込む
n=~PORTC;        //ポートCから各ビットを反転して1バイト読み込む
```

● 1ビット読み込み

ポートからビット入力するときは、ビット検査スキップ命令を使って次のように記述します。

▶ アセンブリ言語の記述例

```
BSF    TRISC, 6   ;ポートCのビット6を入力ポートに設定する
BTFSS  PORTC, 6   ;ポートCのビット6がHレベルなら次命令をスキップして処理Xへ
GOTO   other     ;Lレベルなら別の処理へ
    処理X
    .....
BTFSC  PORTA, 4   ;ポートAのビット4がLレベルなら次命令をスキップして処理Yへ
GOTO   other     ;Hレベルなら別の処理へ
    処理Y
```

▶ C言語の記述例

```
TRISCbits.TRISC6=1;          //ポートCのビット6を入力ポートに設定する
if ( PORTCbits.RC6== 1 )     //ポートCのビット6がHレベルなら処理Xを実行
{
    処理X
}
    .....
if ( !PORTAbits.RA4 )        //ポートAのビット4がLレベルなら処理Yを実行
{                             //(このような書き方もできる)
    処理Y
}
```