

40Vの高さから100Vの点を計ると60Vと測定されてしまいますよね。「基準」を意識することはとても大事なわけですが、その基準を大地としているわけです。

さて、家庭用や工場への電力線は必ずどこかで、もじどおり、「接地」(地中に接続)され、基準が確保されています。ところが、組込み製品は接地できないものがほとんどです。

しかし、このような場合でも、組込み製品に内蔵されている電子回路には必ずグランド(訳せば「接地」)なのですが、漢字の意味に引っ張られるのを避けるためにあえてカタカナ用語で記します)が存在します。それは、電子回路が動作するためには基準電位、あるいは基準電位を確保する仕掛けが必要だからです。

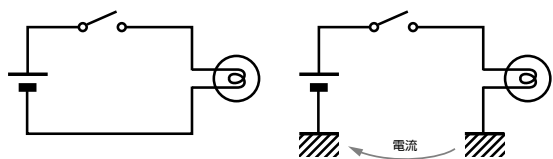
たとえば、TTLロジックと呼ばれる、5Vを論理Highとして扱う電子回路は、0Vの基準が確立しているのです。相対的に5Vを5Vとして、つまり、論理Highとして判定できるわけです。グランドが不確かだと、電子回路の部分部分で「こっちは5Vを出力しているぞ」、「こっちはそれを3Vと判定したぞ(3Vは多くのTTL回路ではHighでもLowでもない不定領域)」といった具合になり、回路が正しく動作しません。

グランドにはもう2つ、大事な役割があります。

2番目の役割は、電流の面から見たグランド、つまり、電流を流す経路としてのグランドです。

図3.34左は、電流を流して信号を伝える接続の例です。電池にスイッチをつなぎ、電球をON/OFFしています。電流ですから「行き」と「戻り」があるわけですが、この「戻り」の部分に図3.34右のようにグランド接続を使うのです。

図3.34：回路としてのグランド



グランドに電流を流す場合、グランドが不十分、つまり、電気接続として不十分で接触不良や電気抵抗があったりする場合、当然、電線に流れる信号はその影響を受けます。

3番目の役割は、「保安接地」と呼ばれるものです。電子レンジや洗濯機に

は必ずアース端子が付いていて、緑色の電線が繋がっているはずですが、これが保安接地の電線(アース線)で、実際にその先を大地に埋めることが要請されています(集合住宅では各戸に接地端子が準備され、そこに接続します)。これは、機器の故障により危険な電圧が内部に発生したときに、それを大地に逃がしてユーザーの安全を確保しようという仕掛けです。

さて、図3.34にすでに記号が登場しましたが、回路記号ではグランドを図3.35のどれかで表記します。

図3.35：グランドを表す回路記号



グランドは、先にも書いたとおり、電子回路が動作するためにとても重要な役割を持っています。ぜひ、この役割を覚えておいてください。グランドの接続状態がおかしいとプログラムが読み取る信号の論理値が不安定になったりするわけですから、知っておけば信号関係のトラブルの解決に役立つはずですよ。

### ■オープンコレクタ/オープンドレイン

マイコンやデバイスの出力回路では「オープンコレクタ」や「オープンドレイン\*」がよく見られます。マイコンをはじめとしたIC(Integrated Circuit)はトランジスタを組み合わせて作られます。ここではトランジスタの中で「バイポーラトランジスタ」と呼ばれるトランジスタを使った出力回路を説明しましょう。

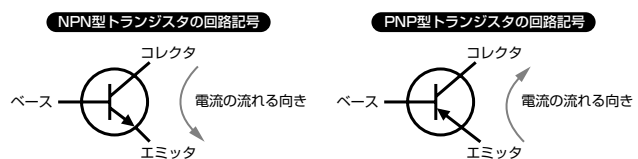
バイポーラトランジスタにはNPN型とPNP型の2種類がありますが、それぞれの回路図上での記号と端子名とを次ページ図3.36に示します。

\*バイポーラトランジスタではなく電界効果トランジスタ(Field Effect Transistor, FET)のドレイン端子に外部のデバイスを接続して使用する出力回路をオープンドレインという。

次ページ図3.37のようにコレクタ端子に外部のデバイスを接続して使用する出力デバイスを「オープンコレクタ(デバイス)」といいます。

オープンコレクタでは、コレクタとエミッタの間にスイッチがあり、そのスイッチをベースに与える信号でON/OFFする、と考えるとわかりやすいでしょう。NPNトランジスタを使用するオープンコレクタ回路では、プログラムから見るとベースへの信号がHighのときにスイッチON、LowのときにOFFとなります。

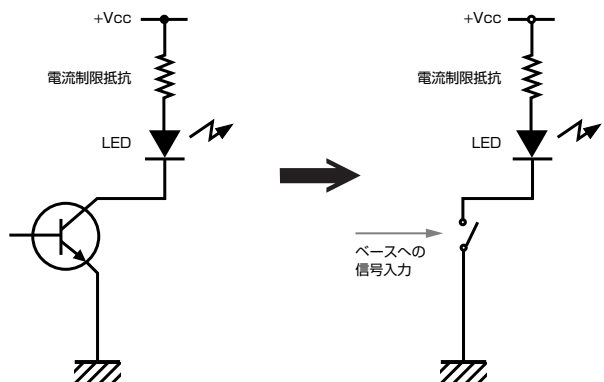
図3.36：トランジスタの回路記号と端子名



たとえば、オープンコレクタの端子にLED (Light Emission Diode: 発光ダイオード) を接続し、プログラムからLEDを点灯/消灯させる回路は、図3.37左のようになります。また、この等価回路 (論理的に同じ意味を持つ回路) は図3.37右のようになります\*。

\*Vccは電源電圧を意味する。デジタル回路の設計により、5V、3.3V、1.2Vなどさまざまな電圧がある。

図3.37：オープンコレクタの接続と等価回路



この回路では、ベースへの信号によりトランジスタをONにすると、コレクタとエミッタとの間が導通し (スイッチがONになり)、LEDに電流が流れてLEDが発光します。

具体的には、たとえば、この出力がアドレス0xff00のビット0に割り当てられていて、このビットへの論理値1の出力で点灯、論理値0の出力で消灯となっている場合、点灯のためには次のようなプログラムを書くことになります。

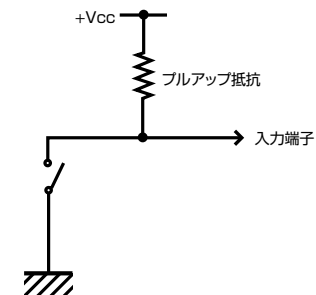
```
#define BIT0    (0x01)
unsigned char *p_port, portval;
p_port = (unsigned char *)0xff00;
:
/* portval へのさまざまな操作 */
:
*p_port = portval | BIT0; /* LEDを点灯させる */
```

### ■プルアップとプルダウン

マイコンやデバイスの入力回路では、「プルアップ」や「プルダウン」がよく見られます (プルアップ入力である、プルダウン入力である、のように表現します)。

これは回路の入力端子に、電圧Highと電圧Lowの状態を安定して与えるように、プルアップ抵抗やプルダウン抵抗と呼ばれる抵抗素子を使用するものです。図3.38は、プルアップ抵抗をスイッチ入力回路に使用している例です。

図3.38：プルアップ抵抗の使用例

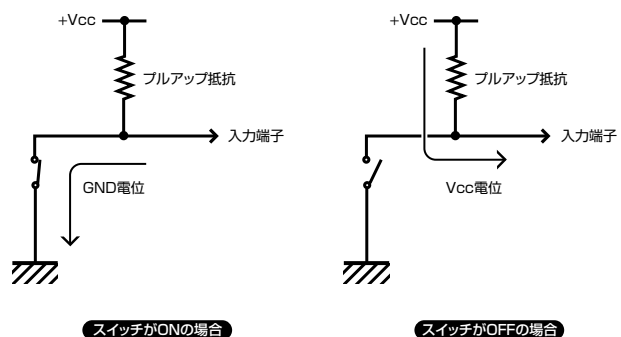


この回路でプルアップ抵抗がどのような役割を持つかを、次ページ図3.39で説明します。

図3.39左では、スイッチをONにしています。これにより入力端子がグランド電位と同じ (電位Low) となります。

図3.39右では、スイッチをOFFにしています。これにより、入力端子がプルアップ抵抗を介してVcc電位に引っ張り上げられ (Pull-upされ)、電位Highとなります。

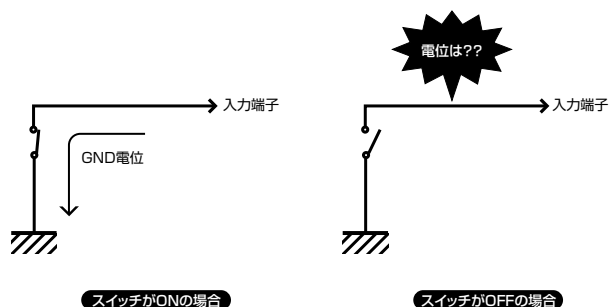
図3.39：プルアップ抵抗の役割



ここで、プルアップ抵抗が接続されていない場合を見てみましょう。図3.40を見てください。左の図では問題なく入力端子が電位Lowであることを知ることができるでしょう。しかし、右の図では入力端子がどこにも接続されない状態\*となり、その電位は「不定」となります。プルアップ抵抗の重要性が理解できるでしょう。

\*このような接続を「浮いている」という。

図3.40：プルアップ抵抗がない場合の入力回路図



同様の役割がプルダウン抵抗にもあります。プルダウン抵抗がどのような役割を持つかを、図3.41で見てみましょう。

図3.41：プルダウン抵抗の役割

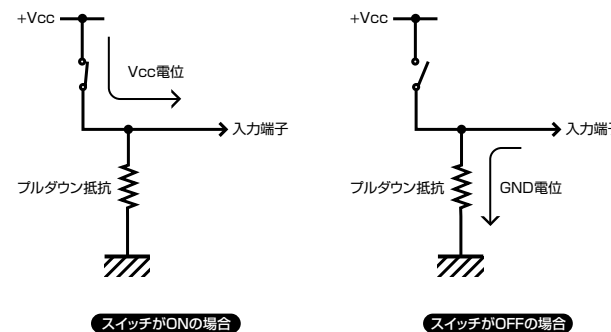


図3.41左では、スイッチをONにしています。これにより、入力端子がVcc電位と同じ（電位High）となります。

図3.41右では、スイッチをOFFにしています。これにより入力端子がプルダウン抵抗を介してグラウンド電位に引きずり下げられ（Pull-downされ）、電位Lowとなります。

プルアップ抵抗と同様に、右の図でプルダウン抵抗がないと入力端子の電位が不定になってしまうことがわかるでしょう。

プルアップ抵抗/プルダウン抵抗がどのような役割を持っているかを理解できましたか？ プルアップ抵抗を使用する場合とプルダウン抵抗を使用する場合は、スイッチの状態と入力論理値がちょうど逆の関係ですね。でも、これを丸暗記するのではなく、実際の回路図を見るときにここまでの説明を思い出し、皆さん自身でスイッチと論理値との関係をそのつど判断できるようになりましょう。

また、マイコンによっては、入力ポートの内部にプルアップ抵抗を接続するかどうかをレジスタへのデータ設定により制御できるものがあります。その設定を漏らしてしまうと……どうなるかはわかりですね\*？ スイッチなどの入力回路で、マイコンまでの間にプルアップ抵抗やプルダウン抵抗が見当たらないときは、この設定が必要かどうかをマイコンのマニュアルで確認してみましょう。

\*図3.40のようになり、スイッチ入力の状態を正しく知ることができない。

### 3.6.5 ハードウェアの特性

ここからは、ハードウェアが持つ独特の性質を見ていきましょう。

#### ■ チャタリング

電氣的に導通をON/OFFするスイッチやリレーデバイスでは、金属どうしを機械的に付いたり、離したりしてON/OFFを実現するものがあります（模式図を図3.42に掲げました）。このような仕掛けを持つ場合、接点どうしの接触時に「ばたばたっ」と短い時間に接点が付いたり離れたりするという動作がよく見られます。この結果、接点からの信号が図3.43のようになります。この現象のことをチャタリング (*Chattering*) といいます（技術者の間では「チャタる」といういい回しもされます）。

図3.42：典型的な機械接点スイッチ

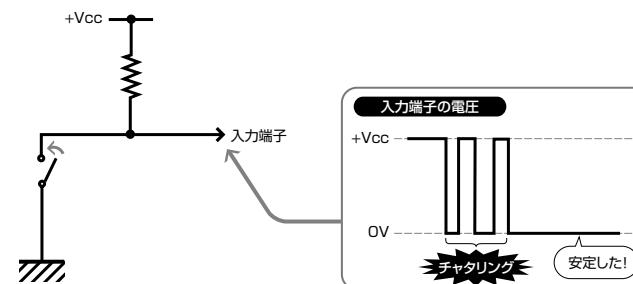


このようなチャタリングは、スイッチがなんども押されたといった誤認識のもととなります。そこで、電気回路的にチャタリングを発生しにくくしたり、ソフトウェアでチャタリングを避けた正確な判断をするプログラムを組んだりすることになります。

以下は、ソフトウェアでチャタリングを回避するロジックの例です。

- ある時間間隔において何回かスイッチ状態をチェックし、n回連続してON状態であればスイッチONとして判断する
- チャタリングの周期よりも長い時間間隔でスイッチの状態をポーリングする

図3.43：チャタリング



#### ① ポーリング——状態チェックの常套手段

ポーリングとは、繰り返しの中で状態をチェックする処理のことです。プログラムで書くとき次のようになります。

```
#define BIT0 (0x01)
#define BIT1 (0x02)
unsigned char port_save=0x00, portval1, portval2;
while(1) { /* 無限ループ */
    /* ポートAの状態を変数portval1に読み込む */
    :
    if (portval1 & BIT0) {
        /* ポートAのビット0が1の場合に対応した処理 */
    }
    else if (portval1 & BIT1) {
        /* ポートAのビット1が1の場合に対応した処理 */
    }

    /* ポートBの状態を変数portval2に読み込む */
    :
    if (portval2 != port_save) {
        /* ポートBの状態変化に対応した処理 */
        port_save = portval2;
    }
    :
}
```

ポーリングでは、チェックの順序（とそれに関係する処理の順序）が決まっています。「今回だけこの処理を先にして！」という要望には応えることができません。また、ポーリング