

■ 2 進数の組合せ

コンピュータでデジタルデータを用いる場合 “2 進数” が利用されています。これは現在のコンピュータが、電圧の “高い：1” と “低い：0” のふたつの状態しか利用できないためです。

この電圧の “高い：1” と “低い：0” を用いることで様々な情報を扱うためこのふたつの状態、つまり 2 進数を組み合わせて様々な状態を表す事を考えました。例えば “文字” を扱えるようするには下記のような組合せになります。

数字 0～9 (10 通り)	2^4 で 16 通り
アルファベット小文字 a～z (26 通り)	2^5 で 32 通り
アルファベット大文字 A～Z (26 通り)	同上

全て識別するためには $10 + 26 + 26 = 62$ 通り 2^6 で 64 通り

文字だけではなく、空白、改行などの記号も使いたいため、現在では文字や記号を表現するための最小の組合せは **7bit** を使って 2^7 の 128 通りを使います。これが “**ASCII 文字コード**” となります。

でも、ここで問題が！ コンピュータでいくら文字や数字を扱えるようにしても、結局コンピュータには電圧の “高い：1” と “低い：0” しか理解ができず、ただ “文字” を判別できるだけで何もできません。

そこで、CPU に対する命令をあらかじめ準備しておき、それを 2 進数の組合せで呼び出せば、なんかできんじゃない？ と思って考えたのです。その例として、下記のような命令体系を作ってみました。

命令表

2bit $2^2 = 4$ 通り

00	右手をあげる	01	左手をあげる	10	右手を下げる	11	左手を下げる
-----------	---------------	-----------	---------------	-----------	---------------	-----------	---------------

3bit $2^3 = 8$ 通り

000	右手をあげる	001	左手をあげる	010	右手を下げる	011	左手を下げる
100	右手を開く	101	左手を開く	110	右手を閉じる	111	左手を閉じる

4bit $2^4 = 16$ 通り

0000	右手をあげる	0001	左手をあげる	0010	右手を下げる	0011	左手を下げる
0100	右手を開く	0101	左手を開く	0110	右手を閉じる	0111	左手を閉じる
1000	体を右に回す	1001	体を左に回す	1010	体を右に傾ける	1011	体を左に傾ける
1100	原点復帰	1101	笑う	1110	“OK” と言う	1111	何もしない (nop)

4bit 使う場合、右手をあげて、左右に体を振るという動作はこのような数列になります。

右手をあげる	体を右に回す	体を左に回す	体を右に回す	体を左に回す	終了
0000	1000	1001	1000	1001	-



隣の人に命令してみましょう！

動くときのルール！！

- ・1 処理 1 秒程度、“何もしない”も 1 回 1 秒としてください。
- ・命令書は組合せ番号のみ伝えてください。

いかがでしょうか、このままでもできるのですが、bit 数が増え、命令数が増えると表現できづらくなります。そこで 0、1 の並びを 4bit ずつに分け“16 進数”することにします。そうすると先ほどの命令は、次のようになります。

右手をあげる	体を右に回す	体を左に回す	体を右に回す	体を左に回す	終了
0000	1000	1001	1000	1001	-
0	8	9	8	9	-

0x08989 です。

2 進数は 1bit 増やすと、命令数は 2 倍になります。この調子で、bit 数を増やしてゆけば、どんどん複雑な命令が準備でき、より細かな制御ができるようになります。

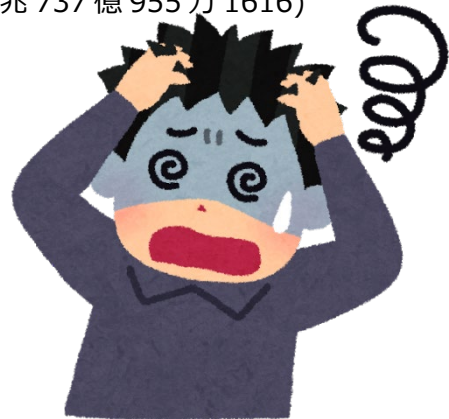
1 チップマイコンでは、8bit、16bit CPU が多いですが、最近では 32bit 以上の bit 数を利用するものが増えてきました。一方、Windows が搭載されるパソコンの CPU は現在 32bit から 64bit に移行しており、皆さんが利用している Windows も、現在 64bitCPU に対応したバージョンを利用しています。

ちなみに、64bit だと組合せ数は

“1844,6744,0737,0955,1616 (1844 京 6744 兆 737 億 955 万 1616)”

※日本語読みを表すために、4 桁区切りで表現しています。

すでに、何を言ってるのか分からない組合せ数ですね。



○チェックポイント・キーワード



- ・2 進数を利用する理由
- ・2 進数の組合せは、1bit 増えるごとに倍になる。
- ・文字を 2 進数で表現する : ASCII 文字コード
- ・組合せ数と処理
- ・16 進数表記