

# 数値の表現方法と ビットごとの論理演算

鹿取 祐二

組み込みシステムでは、特定のビットだけを操作することが多いです。スイッチやLEDはGPIOの特定のビットに接続されています。ビット操作によりスイッチ(SW)に連動してLEDを点灯させることができます。

## 1 整数は8, 10, 16進数で表す

C言語における整数値の表現は「10進数」「8進数」「16進数」の3種類があります。第1章のリスト3では1234と111の加算と減算をしました。分かりやすいのは、10進数ですが、マイコンのGPIOは2進数で操作します。C言語はビット操作に最適な2進表現を持っていません<sup>注1</sup>。2進数の4桁が16進数の1桁になるので、ビット操作では16進数を使います。

表1 10進数, 8進数, 16進数, 2進数の対応表

10進数	8進数	16進数	2進数
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	10	8	1000
9	11	9	1001
10	12	A	1010
11	13	B	1011
12	14	C	1100
13	15	D	1101
14	16	E	1110
15	17	F	1111

表2 1000 (10進数) を8進数と16進数で表すとまるで違う表記になる

10進数	8進数 (10進数の1000と同じ)	16進数 (10進数の1000と同じ)
1000	01750	0x3E8

## ● 頭に0xと書いたら16進数, 頭に0と書いたら8進数

先頭に0(プリフィックス)を付けると、「8進数」として扱われます。「16進数」は先頭に0xのプリフィックスを付けて数字の0~9, 英文字のA~Fを使って記述します。0xのxを含め, A~Fは大文字と小文字のどちらかで記述してもかまいません。表1は各進数の対応表です。

10進数の1000を8進数や16進数で表現すると、表2のようになります。2進数で書くと、1000は「0011110 1000」です。覚えるまで、表1の対応表とにらめっこしてください。

## 2 ビットを操作する演算子「&」「|」「^」

C言語にはマイコンのビット操作を行う演算子が用意されています。論理積(AND)の「&」、論理和(OR)の「|」、排他的論理和(XOR)の「^」の3つがあります。

### ● 特定のビットを0にクリアする「&」

論理積(AND)の真理値を表3に示します。両方の被演算数が1のときだけ結果が1になり、片方の被演算数が0のときは0になります。

A列を元の値, B列をANDする値と考えると, A列の元の値が0でも1でも, B列の0とANDを取れば結果は必ず0になります。B列の1とANDを取ればA列の元の値は変化しません。

この結果から, &は特定のビットを0にクリアするときに使えることが分かります。

リスト1のようなプログラムを実行すると, 0とANDしている変数aの上位バイトは元の値に関係なく必ず0になり, 1とANDしている変数aの下位バイトは元の値を保持した結果になります。

注1: C言語をベースに機能拡張したプログラム言語C++では、2014年12月に改訂されたバージョンC++14から、2進数を表す0b(もしくは0B)が使えるようになった。