16 STEP

スピーカを使おう

STEP16 では、スピーカから音を出してみます。
スピーカを使うのは難しいような気がするかもしれませんが、単純はビープ音は LED の点滅と同じ要領で出すことができます。

16.1　スピーカとは

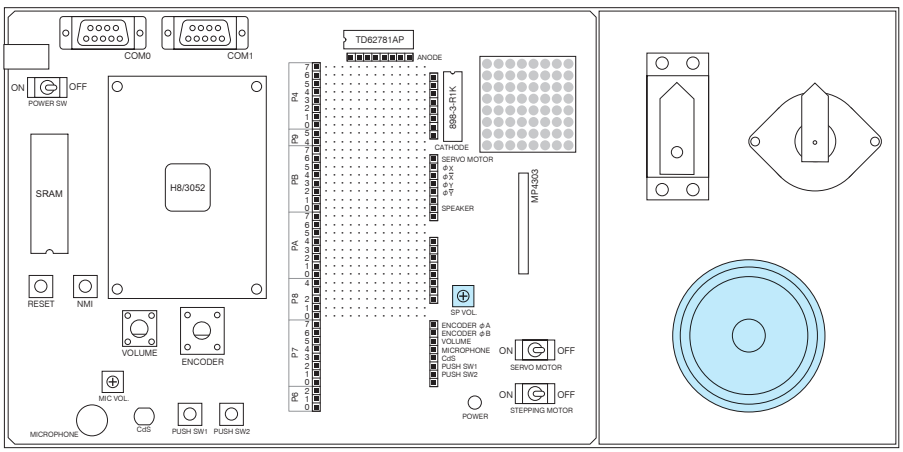


図 16-1　スピーカ

SP VOL は、スピーカの音量調整用です。時計回りにすると音が大きくなり、反時計回りにすると音が小さくなります。SP VOL を回す際は、ドライバーを使用してください。

スピーカは電気信号を空気の振動に変換し「音」として出力する部品です。これまで使ってきたパルス信号でも音として出力することができます。

16.2 スピーカにパルスを送る

パルス周波数によって音の高さが変わります。例えば、音階の基準音として使用される「ラ」の音は 440Hz です。周波数 440Hz の場合、1 秒間に 440 回の波を繰り返しています。
ON-OFF のパルス波形では、図 16-2 のように表せます。

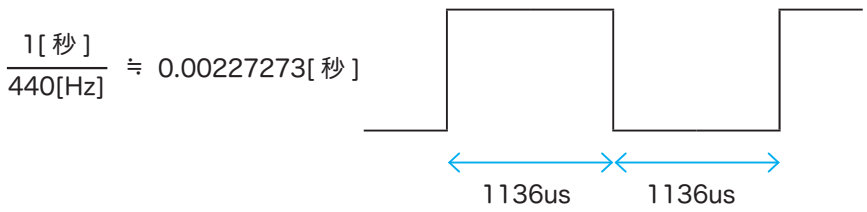
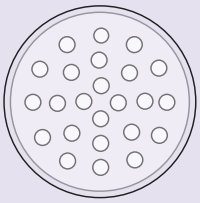


図 16-2 周波数 440Hz のパルス

音は、周波数が上がる (パルス幅が短くなる) と高く聞こえ、下がる (パルス幅が長くなる) と低く聞こえます。
また、人が認識できる周波数の範囲は 20Hz ～ 20kHz です。この範囲外の音、例えばコウモリが出す 30kHz から 100kHz の高周波は人には聞き取ることがきません。

課題 16-1

周波数 440[Hz] を矩形波をスピーカから出力する。



スピーカを配線しましょう。図 16-3 のように追加配線してください。

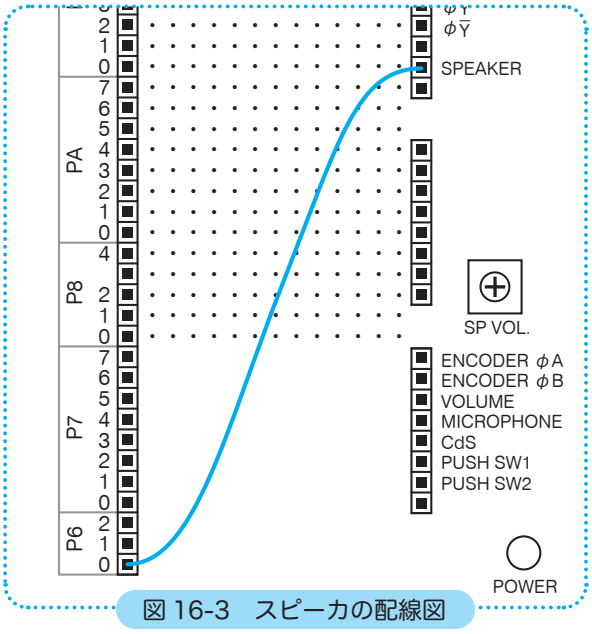


図 16-3　スピーカの配線図

440[Hz] のパルスをスピーカに送るフローチャートは図 16-4 のようになります。
パルス出力はサーボモータの駆動で使いましたね。
440Hz のパルス＝ 1136us のパルス幅ですからフローチャートの待ち時間を 1136us にする必要があるのですが・・・

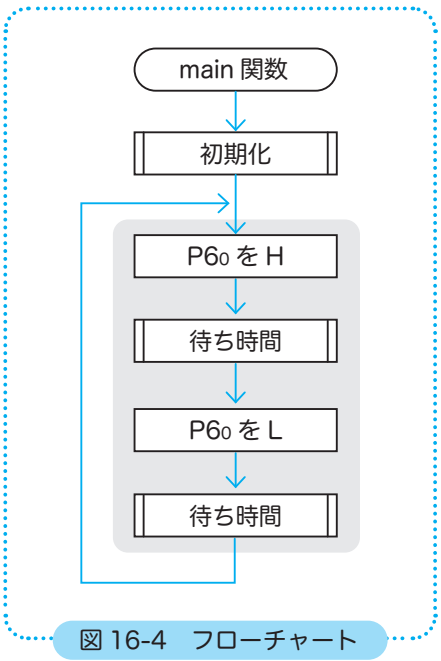


図 16-4　フローチャート

STEP 16 スピーカを使おう

μs 単位の待ち時間を 1ms の関数では作れません。
1ms = 1560 回ループから計算すると、1μs = 1.56 回ループとなり不可能です。
ループ内に k++ を無くすと実測で 1ms = 31250 回ループなので、1μs = 31.25 回ループとなり、31 回ループで 1μs に近い時間になりそうですが、実測すると 6.9μs でした。
1 回ループでも同じ 6.9μs だったので、ループ回数と待ち時間が比例していません。ループによる待ち時間の限界です。

マイコンで正確な時間計測は割込みや ITU（Integrated Timer Unit）を使うべきなのです。
割込みは STEP24 で、ITU については STEP22 で扱います。

正確な 1μs タイマは作れないのですが、課題の音を鳴らすぐらいであれば以下の関数で近い値になります。1136μs の待ち時間は waitUS(1136) で呼び出します。

```
/*
 *  待ち時間関数 [us]
 */
void waitUs(int us)
{
    int i, k;
    for (i = 0; i < us; i++)
        k++;
}
```

16.3　P6DDR と P6DR

ビット：	7	6	5	4	3	2	1	0
	—	$P6_6DDR$	$P6_5DDR$	$P6_4DDR$	$P6_3DDR$	$P6_2DDR$	$P6_1DDR$	$P6_0DDR$
初期値：	1	0	0	0	0	0	0	0
R/W：	—	W	W	W	W	W	W	W

図 16-5　P6DDR と初期値

ビット：	7	6	5	4	3	2	1	0
	—	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
初期値：	1	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

図 16-6　P6DR と初期値

※ ポート 6 は 7 ビットの入出力ポートです。バス制御入出力端子と共用になっています。
本キットでは、$P6_0$ ～ $P6_2$ は CN3 に配置されており利用可能です。
詳細は「H8/3052F ハードウェアマニュアル」をご参照ください。

STEP 16 スピーカを使おう

プログラム例　16-1

```
/************************************************************
    製作者　アドウィン
    解説　　440Hz の音を出す
************************************************************/
#include <3052f.h> // 3052F 固有の定数

/*
 *  初期化関数
 */
void initIO(void)
{
    P6.DDR = 0xFF; // 出力 スピーカ
}

/*
 *  待ち時間関数 [us]
 */
void waitUs(int us)
{
    int i, k;
    for (i = 0; i < us; i++)
        k++;
}

/*
 *  main 関数
 */
int main(void)
{
    initIO(); // 初期化関数の呼び出し
    while (1)
    {
        P6.DR.BIT.B0 = !P6.DR.BIT.B0;
        waitUs(1136); // 周期 [us]
    }

    return 0;
}
```

⚠ スピーカを使った実験は音量にご注意ください。

・プログラムを実行する前に SP VOL を左いっぱいに回して音量を最小にしておく。

・プログラムを実行中に SP VOL で適度な音量に調節する。

16.4　音階

440Hz の音はラの音に聞こえましたか？
「ブーッ」と音が出たと思いますが、音が 1 つでは面白くありません。音階を出せるようにしてみましょう。表 16-1 に、各音階に対応した周波数を載せておきます。

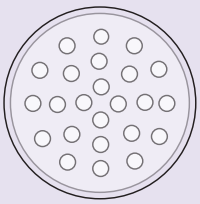
音階	周波数 [Hz]	待ち時間 [us]
ド	261.6	1911
レ	293.6	1702
ミ	329.6	1516
ファ	349.2	1431
ソ	392.0	1270
ラ	440.0	1136
シ	493.8	1026
ド	523.3	955

表 16-1　音階別の周波数

音階は周波数を 2 倍にすると 1 オクターヴ高い音になり、反対に周波数を 1/2 倍にすると、1 オクターヴ低い音になります。低いドと高いドの周波数を比べてみると分かるでしょうか。

課題　16-2

SW1 を押している間は音が鳴り続け、
SW2 を押すたびに、音階が高くなる。
低いド→レ→ミ→ファ→ソ→ラ→シ→高いド→低いド へ戻る



STEP 16 スピーカを使おう

「音階を変える」とは、周波数を変えるということです。周波数を変えるということは、つまり、待ち時間を変えるということになります。音階ごとに待ち時間は決まっているので、配列にしたほうがよさそうです。また、音階用の待ち時間関数は μs 単位ですが、SW のクリックを検出するのでチャタリング対策用の ms 単位の待ち時間関数も用意しておきましょう。

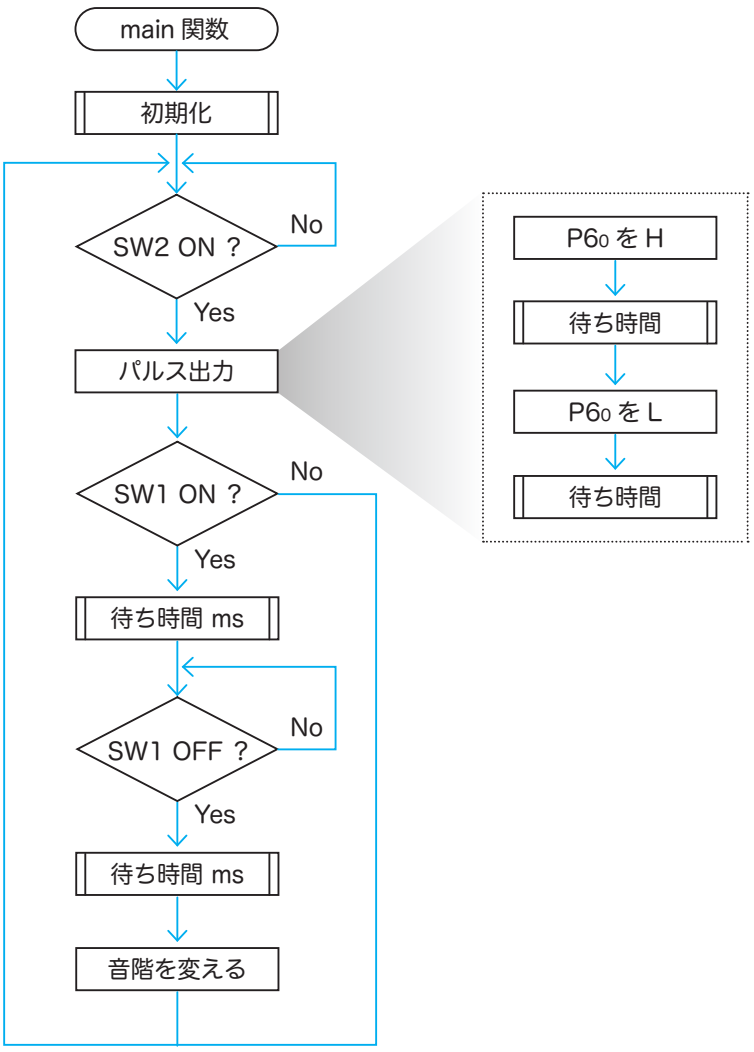


図 16-7　フローチャート

プログラム例　16-2

```
// 各音階での H と L の時間を配列化（ド・レ・ミ・ファ・ソ・ラ・シ・ド）
int onkai[8] = {1911, 1702, 1516, 1431, 1270, 1136, 1026, 955};
```

中略

```
/*
 *  main 関数
 */
int main(void)
{
    initIO();       // 初期化関数の呼び出し
    int index = 0;  // 配列に使う添え字の変数

    while (1)
    {
        // SW1 が ON ならば音を出す
        if (SW1_ON)
        {
            P6.DR.BIT.B0 = !P6.DR.BIT.B0;
            waitUs(onkai[index]); // 音階周期 [us]
        }

        // SW2 を押して離すと配列の添え字の値（音階）を変更
        if (SW2_ON)
        {
            waitMs(10);
            while (SW2_ON)
                ;
            waitMs(10);

            index++; // 音階の変更

            // index 値の補正
            if (index > 7)
                index = 0;
        }
    }

    return 0;
}
```