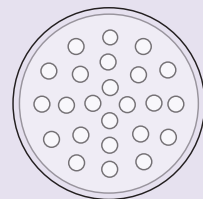


## D/A 変換しよう

STEP20 では、D/A 変換 (デジタル / アナログ) 変換について学習します。デジタル音声データを D/A 変換してスピーカに出力してみましょう。

## 課題 20-1

音声ファイルを再生して、スピーカから「こんにちは」という音声を出させる。



おすけい

## 20.1 D/A 変換

スピーカに方形波のパルスを送って鳴らすビープ音ではなく、声に聞こえる音声を鳴らしてみよう。音声に聞こえるには、どんな波形にすればいいのでしょうか？

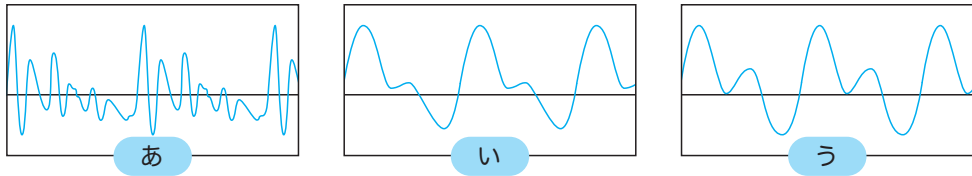


図 20-1 音声波形

図 20-1 を見てください。これは、「あ」「い」「う」の音声波形です。これはアナログ波形ですね。ですから、音声として再生するためにはコンピュータからアナログ波形を出力しなければなりません。

アナログ情報をデジタルに変換することを A/D 変換といいました。逆にデジタル情報をアナログにすることを **D/A 変換** といいます。デジタル化された音声データをスピーカで再生する時には D/A 変換が必要です。CD や MP3 もデジタルデータを D/A 変換して再生しているのです。

音声データは、exercise > include フォルダの中の“wave1.h”に記録されています。「wave1.h」ファイルに記録された配列「wavedata1」には、9008 個分の電圧値が 16 進数データで保存されています。この配列をエクセルで 10 進数に変換し、グラフ化したものが図 20-2 です。これは「こんにちは」の音声波形そのものです。

この配列の各要素を D/A 変換してスピーカから出力させると、「こんにちは」という音声聞こえるというわけです。

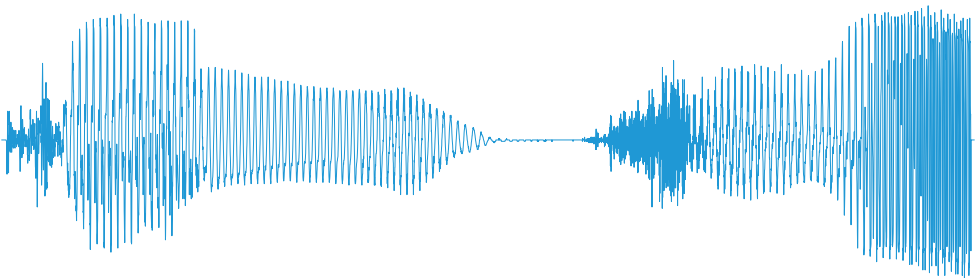
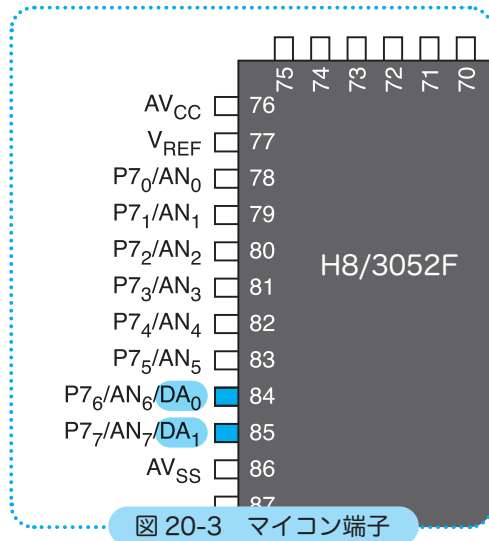


図 20-2 「wavedata1」のグラフ化波形

## 20.2 D/A 変換器

H8/3052F は、2 チャンネルの D/A 変換回路を内蔵し、それぞれ独立して変換を行うことができます。

DA<sub>0</sub>、DA<sub>1</sub> は D/A 変換されたアナログ値が出力されるアナログ出力端子です。それぞれ P7<sub>6</sub>、P7<sub>7</sub> との共用端子です。



## 20.3 D/A コントロールレジスタ【DACR】

DACR は D/A コントロールレジスタの略です。

その名のとおり D/A 変換の動作を制御するレジスタです。

DACR も、A/D 変換用 ADCSR のように内部処理に使われるレジスタなので、対応する外部端子はありません。8 ビットのレジスタで、下位 5 ビットはリザーブビットです。

ビット:	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
初期値:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	—	—	—	—	—

※フラグをクリアするための 0 ライトのみ可能です。

図 20-4 DACR

## 【bit 7 DAOE1】D/A アウトプットイネーブル 1

D/A 変換とアナログ出力を制御するビットです。チャンネル 1 の D/A 変換の許可とアナログ出力端子 DA1 の許可を設定します。1 を書き込むと許可、0 を書き込むと禁止になります。

0	アナログ出力 DA <sub>1</sub> を禁止
1	D/A 変換と DA <sub>1</sub> 出力を許可

## 【bit 6 DAOE0】D/A アウトプットイネーブル 0

D/A 変換とアナログ出力を制御するビットです。チャンネル 0 の D/A 変換の許可とアナログ出力端子 DA0 の許可を設定します。1 を書き込むと許可、0 を書き込むと禁止になります。

0	アナログ出力 DA <sub>0</sub> を禁止
1	D/A 変換を許可

## 【bit 5 DAE】D/A イネーブル

DAOE1、DAOE0 の両チャンネルの D/A 変換を制御するビットです。

0 を書き込むと各チャンネルを独立して、1 を書き込むと一括して制御します。

0	チャンネル 0、1 の独立制御
1	チャンネル 0、1 の一括制御

DACR で設定できるのは 3 ビットだけなので、組み合わせを表にまとめておきます。

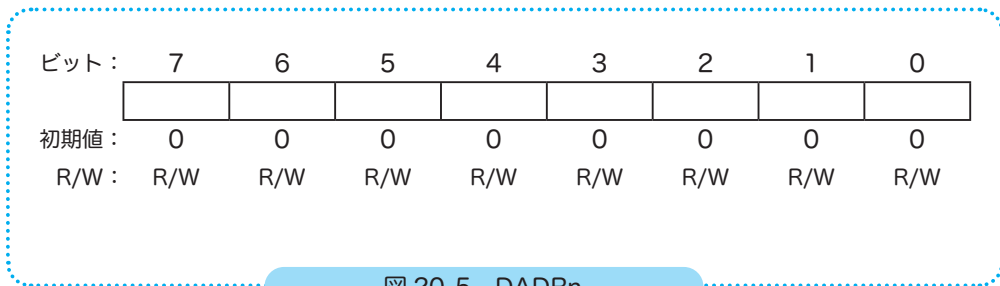
bit 7	bit 6	bit 5	チャンネル 0 DA <sub>0</sub>	チャンネル 1 DA <sub>1</sub>
DAOE1	DAOE0	DAE		
0	0	—	× 禁止	× 禁止
	1	0	○ 許可	× 禁止
		1	○ 許可	○ 許可
1	0	0	× 禁止	○ 許可
		1	○ 許可	○ 許可
	1	—	○ 許可	○ 許可

## 20.4 D/A データレジスタ【 DADR0, DADR1 】

D/A 変換のチャンネルの設定は DACR で、D/A 変換されたアナログ値はアナログ出力端子から電圧出力されるまで分かりましたね？

では、D/A 変換される前のデジタル値はどこに書き込めばいいのでしょうか？

変換の窓口となるのは **DADR** (D/A データレジスタ) です。



DADR0、DADR1 は同じ構成です。個々の DADR は **8 ビット** のレジスタです。DA<sub>0</sub> のレジスタは DADR0、DA<sub>1</sub> のレジスタは DADR1 です。

チャンネルの D/A 変換が許可されていたら、DADR にデジタル値が格納されるとすぐに D/A 変換が始まり、アナログ出力端子に出力されます。

D/A 変換を簡単に言うと「数値を電圧に変換すること」です。

例えば、8 ビットの D/A 変換は  $2^8$  の分解能があり、フルスケールで 0 ~ 256 (0x00 ~ 0xFF) になります。0 ~ 255 の値をアナログ電圧 0 ~ 5 V に変換できるということです。

D/A 変換の仕組みについて理解できたでしょうか？

# STEP 20

## D/A 変換しよう

DA<sub>0</sub> とスピーカを接続します。STEP16 で P6<sub>0</sub> に接続していた配線を図 20-6 のように変更してください。

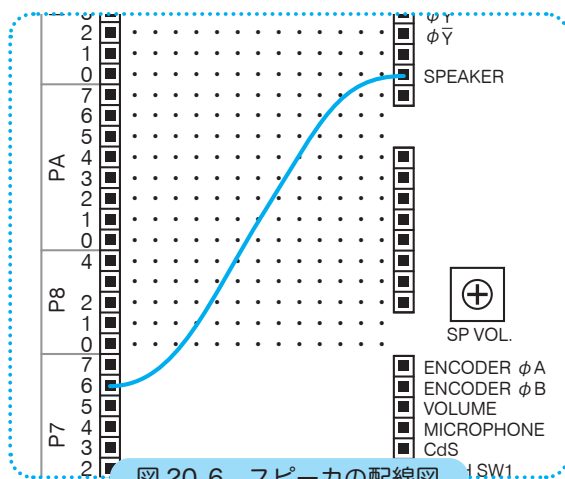


図 20-6 スピーカの配線図

図 20-7 のフローチャートを見て、プログラムの流れを確認しましょう。「SW1 ON?」の条件分岐は、必要であればチャタリング対策してください。

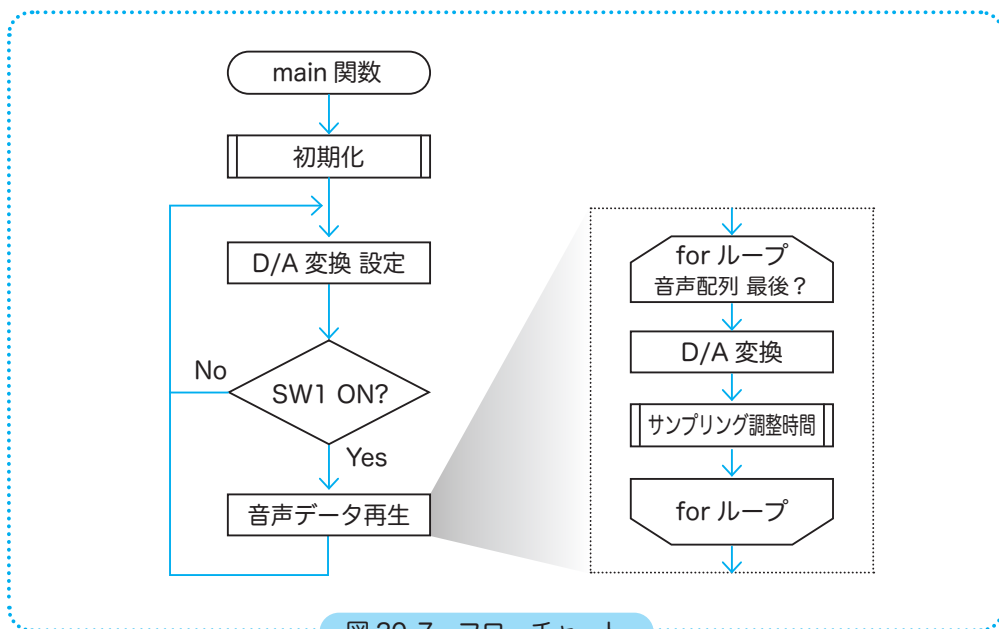


図 20-7 フローチャート

## プログラム例 20-1

```
40  /*
41  * main 関数
42  */
43  int main(void)
44  {
45      int index;
46      initI0(); // 初期化関数の呼び出し
47
48      DA.CR.BIT.DA0E0 = 1; // チャンネル0の D/A 変換とアナログ出力 DA0 の許可
49
50      while (1)
51      {
52          if (SW1_ON)
53          {
54              // チャタリング対策
55              waitMs(10);
56              while (SW1_ON)
57              ;
58              waitMs(10);
59
60              // 配列データを順番に取得 配列数は 9008
61              for (index = 0; index < 9008; index++)
62              {
63                  DA.DR0 = wavedata1[index]; // D/A 変換と再生
64                  waitUs(50);                // サンプリング周波数調整
65              }
66          }
67      }
68
69      return 0;
70  }
```

64 行目にある**サンプリング周波数**とはなんのことでしょうか？

サンプリング周波数 (サンプリングレート) は、A/D 変換や D/A 変換を行う際、1 秒間に何回変換を行うのかを表す周波数のことです。例えば、1 秒間に 100 回 A/D(D/A) 変換を行うとサンプリング周波数は 100Hz ということになります。

A/D 変換と D/A 変換のサンプリング周波数を合わせないと元の音声に聞こえません。つまり、録音と再生のスピードを合わせないと、早送りで聞こえたり、スローに聞こえたりしてしまうということです。

「wave1.h」ファイルのサンプリングレートは **22.050kHz** です。周期 **50 $\mu$ s** 程度で再生すれば元に近い音声になるはずですが、再生時に周期を長くすると低く遅くなり、短くすると高く速くなります。