

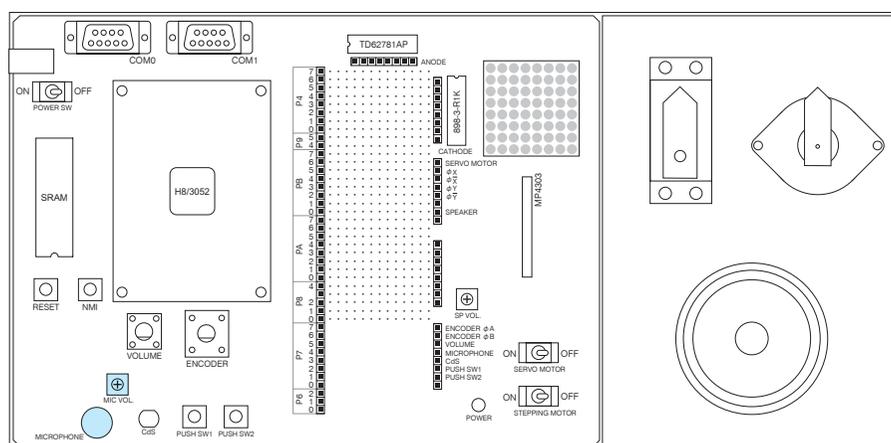
## マイクとスピーカ ～ A/D 変換と D/A 変換～

STEP21 では入力装置であるマイクを使ってみましょう。

マイクはスピーカの逆で、空気の振動を電圧の変化に変える装置です。

マイクから拾った音を入力するのは A/D 変換を使い、音声データをスピーカに出力するのは D/A 変換を使います。

### 21.1 マイクとは



# STEP 21

## マイクとスピーカ ～ A/D 変換と D/A 変換～

### 21.2 録音と再生の手順

録音と再生は以下のような手順で行います。

#### 録音の手順

- ① マイクによって音（空気の振動）をアナログ電圧の波に変換する。
- ② 電圧の波がマイコン端子に送られる。
- ③ 送られたアナログ電圧を A/D 変換し、デジタル値にする。
- ④ デジタル値を記憶させる。

#### 再生の手順

- ⑤ 記憶させたデジタル値を D/A 変換し、電圧の波にする。
- ⑥ 電圧の波がスピーカに送られる。
- ⑦ アナログ電圧の波をスピーカによって、音（空気の振動）にする。

### 課題 21-1

SW1 を押すとマイクから入る音を録音し、SW2 を押すとスピーカから音を再生する。  
インジケータとして録音中は右上の LED が点灯し、再生中は右下の LED が点灯する。  
ボイスレコーダーのような機能ですね。



21.3 配線

それでは、配線作業を行きましょう。

STEP21 ではマイクから拾った音を A/D 変換します。チャンネル 1 を使用するのので、マイクと AN<sub>1</sub> を配線します。スピーカの配線は STEP20 のままです。

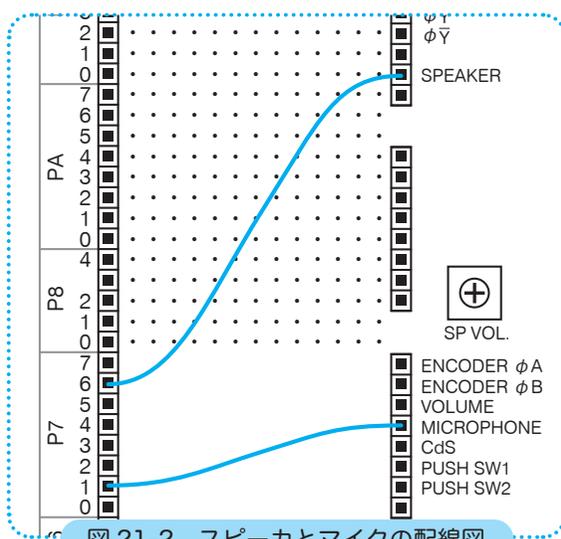


図 21-2 スピーカとマイクの配線図



## プログラム例 21-1

```

48  /*
49  * main 関数
50  */
51  int main(void)
52  {
53      unsigned char data[60000]; // A/D 変換によりできたデジタルデータを格納する配列
54      unsigned int index; // デジタルデータを格納する配列の添え字
55      int i; // サンプリング周波数の調整に使用する変数
56
57      initI0C(); // 初期化関数の呼び出し
58
59      PB.DR.BIT.B0 = 0; // 一番右の列のカソードを L に
60
61      DA.CR.BYTE = 0x5F; // チャンネル 0 (DA0) を使用 スピーカ
62      AD.CSR.BYTE = 0x01; // チャンネル 1 (AN1) を使用 マイク
63
64      while (1)
65      {
66          // SW1 を押すと、録音を開始
67          if (SW1_ON)
68          {
69              // チャタリング対策
70              waitMs(10);
71              while (SW1_ON)
72              ;
73              waitMs(10);
74
75              P4.DR.BIT.B7 = 1; // 録音開始 右上点灯
76
77              // A/D 変換によりできたデジタルデータを配列に格納していく
78              for (index = 0; index < 60000; index++)
79              {
80                  AD.CSR.BIT.ADST = 1; // A/D 変換開始
81                  while (AD.CSR.BIT.ADF == 0) // A/D 変換終了待ちループ
82                  ;
83                  data[index] = (AD.DRB >> 8); // 録音
84                  waitUs(50); // サンプリング周波数調整
85              }
86              P4.DR.BIT.B7 = 0; // 録音終了 全消灯
87          }

```

⋮

53 行目の unsigned char 型、54 行目の unsigned int 型はそれぞれ、符号なしの文字を定義する時と、符号なしの整数を定義する時に用います。

char 型と int 型では、それぞれ -32768 ～ +32767 の文字と -32768 ～ +32767 の整数を定義することができましたが、unsigned char 型と unsigned int 型では、それぞれ 0 ～ 65535 の文字と 0 ～ 65535 の整数を定義できます。

```
88
89 // SW2 を押すと、再生を開始
90 else if (SW2_ON)
91 {
92     // チャタリング対策
93     waitMs(10);
94     while (SW2_ON)
95         ;
96     waitMs(10);
97
98     P4.DR.BIT.B0 = 1; // 再生開始 右下点灯
99
100    for (index = 0; index < 60000; index++)
101    {
102        DA.DR0 = data[index]; // D/A 変換と再生
103        waitUs(50);          // サンプリング周波数調整
104    }
105    P4.DR.BIT.B0 = 0; // 再生終了 全消灯
106 }
107 }
108
109 return 0;
110 }
```