

ミニ・オルガン(SDカード対応)(PIC18F2550)

概要

音譜をテキストファイルに記入し、それを自動演奏させるオルガンを製作しました。

動作原理

<プログラムの流れ>

1. ポートや内臓モジュールを初期化します。
2. SDカードを初期化します。
3. スタートスイッチが押下されるのを待ちます。
4. ファイル(ファイル名は、“organ.txt” 固定)をオープンします。
5. カンマ区切り単位で音譜を読み込みます。
6. 読み込んだ音譜の周波数で音を発生させます。(ポートのON/OFFをソフトで処理)
7. ファイルの最後まで5.6.を繰り返します。
8. 3.へ戻ります。

<演奏までの流れ>

1. パソコン上のテキストエディタで、音譜を作成し、テキストファイルに記入し、SDカードに保存します。
2. SDカードを、ミニ・オルガンのSDカードスロットに挿入します。
3. 電源をONにします。
4. SDカードの初期化が成功するとLEDが3回点滅します。失敗すると点滅したままになります。
5. スタートスイッチを押下すると、演奏がスタートします。演奏中はLEDが点灯します。

<表記> 音階の周波数および音譜の表記は、次の表のようになります。(音譜間はカンマ区切り)

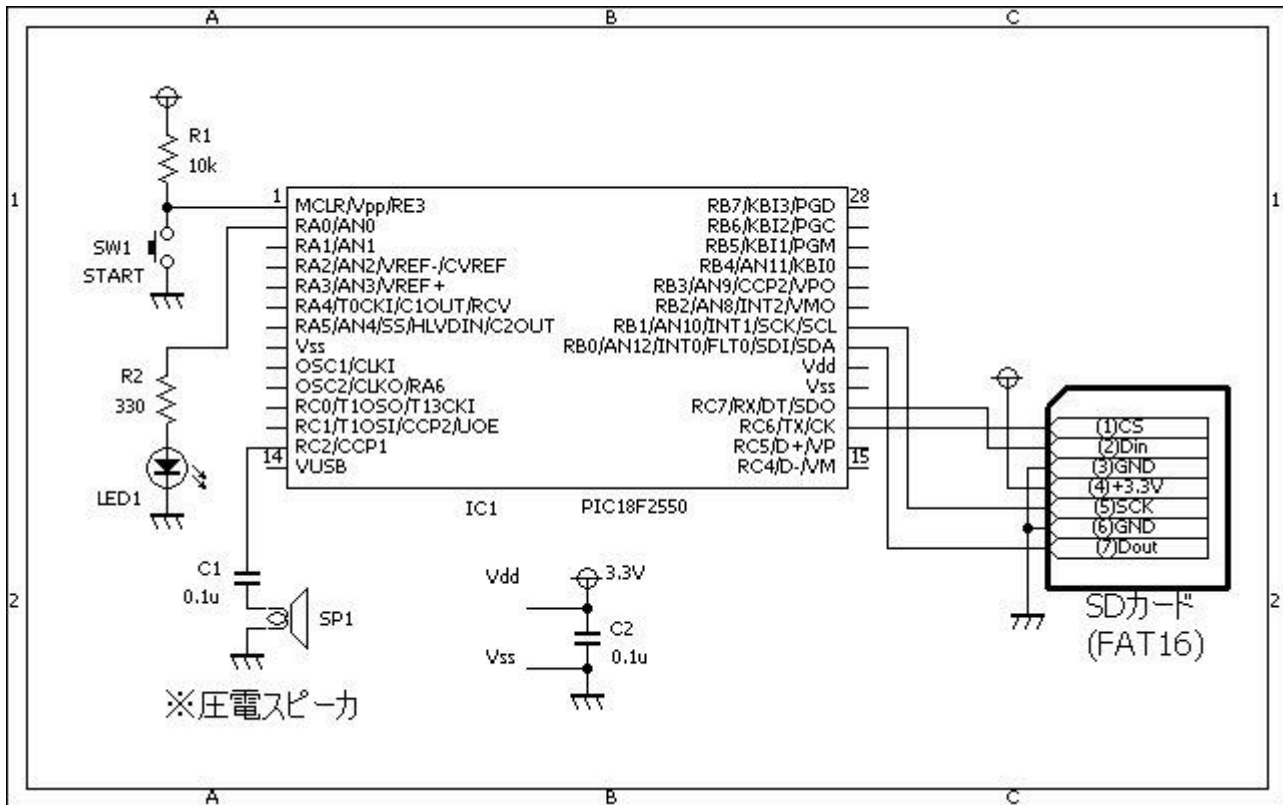
周波数(Hz)	262	294	330	349	392	440	494	523	587	659	699	784	880	988
音階	ド	レ	ミ	ファ	ソ	ラ	シ	ド	レ	ミ	ファ	ソ	ラ	シ
表記	DO	RE	MI	FA	SO	RA	SI	DO2	RE2	MI2	FA2	SO2	RA2	SI2
	-							オクターブ上						

<その他の表記> DO :ド DO-:ドー -:休止記号

<表記例:ドレミファソラシ> DO,RE,MI,FA,SO,RA,SI

<表記例:1オクターブ上のドレミファソラシ> DO2,RE2,MI2,FA2,SO2,RA2,SI2

回路図



ソースコード

[mini_organ.c](#)

```

//*****
*
/*
「簡易自動演奏オルガン(SDC対応)」
*/
//*****
*

#define SW PORTC.F3
#define LED PORTA.F0

#define CR 0x0d
#define LF 0x0a

#define DO 262
#define RE 294
#define MI 330
#define FA 349
#define SO 392
#define RA 440
#define SI 494
#define DO2 523
#define RE2 587

```

```
#define      MI2          659
#define      FA2          699
#define      S02          784
#define      RA2          880
#define      SI2          988

//*****
*

void  init_sdc()
{
    static  short  cnt;
    //□□□□□□□□の初期化
    Spi_Init_Advanced(MASTER_OSC_DIV64, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
    if (Mmc_Fat_Init(&PORTC, 6)) {
        while (1) {
            LED = 1;
            Delay_ms(100);
            LED = 0;
            Delay_ms(100);
        }
    }
    Spi_Init_Advanced(MASTER_OSC_DIV16, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
    for (cnt = 0; cnt < 3; cnt++) {
        LED = 1;
        Delay_ms(300);
        LED = 0;
        Delay_ms(300);
    }
}

//*****
*

void  SwitchONcheck()
{
    while (Button(&PORTE, 3, 1, 0) == 0)
        ;
    while (Button(&PORTE, 3, 1, 1) == 0)
        ;
}

//*****
*

void  doremiProc(int freq, long time)
{
    static  int  cnt, tmp;
    //
```

```
switch (freq) {
case D0:
    tmp = time / (1000000 / D0);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / D0 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / D0 / 2);
    }
    break;
case RE:
    tmp = time / (1000000 / RE);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / RE / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / RE / 2);
    }
    break;
case MI:
    tmp = time / (1000000 / MI);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / MI / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / MI / 2);
    }
    break;
case FA:
    tmp = time / (1000000 / FA);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / FA / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / FA / 2);
    }
    break;
case S0:
    tmp = time / (1000000 / S0);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / S0 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / S0 / 2);
    }
    break;
case RA:
    tmp = time / (1000000 / RA);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / RA / 2);
    }
}
```

```
        PORTC.F2 = 0;
        Delay_us(1000000 / RA / 2);
    }
    break;
case SI:
    tmp = time / (1000000 / SI);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / SI / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / SI / 2);
    }
    break;
case D02:
    tmp = time / (1000000 / D02);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / D02 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / D02 / 2);
    }
    break;
case RE2:
    tmp = time / (1000000 / RE2);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / RE2 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / RE2 / 2);
    }
    break;
case MI2:
    tmp = time / (1000000 / MI2);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / MI2 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / MI2 / 2);
    }
    break;
case FA2:
    tmp = time / (1000000 / FA2);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / FA2 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / FA2 / 2);
    }
    break;
case S02:
    tmp = time / (1000000 / S02);
```

```
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / S02 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / S02 / 2);
    }
    break;
case RA2:
    tmp = time / (1000000 / RA2);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / RA2 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / RA2 / 2);
    }
    break;
case SI2:
    tmp = time / (1000000 / SI2);
    for (cnt = 0; cnt < tmp; cnt++) {
        PORTC.F2 = 1;
        Delay_us(1000000 / SI2 / 2);
        PORTC.F2 = 0;
        Delay_us(1000000 / SI2 / 2);
    }
    break;
}
}

//*****
*

void organProc()
{
    //変数の定義
    static char buf[10], character, cnt;
    static unsigned long fsize, length;
    //[]のファイルのオープン
    Mmc_Fat_Assign("organ.txt", 0);
    Mmc_Fat_Reset(&fsize);
    length = 0;
    cnt = 0;
    //
    while (length < fsize) {
        Mmc_Fat_Read(&character);
        length++;
        //
        switch (character) {
            case ',':
            case CR:
                buf[cnt] = 0x00;
                cnt = 0;
        }
    }
}
```

```
if (strstr(buf, "D02") != 0x00) {
    doremiProc(D02, buf[3] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "RE2") != 0x00) {
    doremiProc(RE2, buf[3] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "MI2") != 0x00) {
    doremiProc(MI2, buf[3] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "FA2") != 0x00) {
    doremiProc(FA2, buf[3] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "S02") != 0x00) {
    doremiProc(S02, buf[3] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "RA2") != 0x00) {
    doremiProc(RA2, buf[3] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "SI2") != 0x00) {
    doremiProc(SI2, buf[3] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "D0") != 0x00) {
    doremiProc(D0, buf[2] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "RE") != 0x00) {
    doremiProc(RE, buf[2] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "MI") != 0x00) {
    doremiProc(MI, buf[2] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "FA") != 0x00) {
    doremiProc(FA, buf[2] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "S0") != 0x00) {
    doremiProc(S0, buf[2] == '-' ? 750000 : 500000);
    break;
}
if (strstr(buf, "RA") != 0x00) {
    doremiProc(RA, buf[2] == '-' ? 750000 : 500000);
    break;
}
```

```
    }
    if (strstr(buf, "SI") != 0x00) {
        doremiProc(SI, buf[2] == '-' ? 750000 : 500000);
        break;
    }
    if (strstr(buf, "-") != 0x00) {
        Delay_ms(100);
        break;
    }
    break;
case ' ':
    break;
case LF:
    Delay_ms(1000);
    break;
default:
    buf[cnt] = character;
    cnt++;
    break;
}
}
}

//*****
*

void main()
{
    //コンパレータは使用しない。
    CMCON = 0b00000111;
    //□□□変換の設定
    ADCON1.PCFG3 = 1;
    ADCON1.PCFG2 = 1;
    ADCON1.PCFG1 = 1;
    ADCON1.PCFG0 = 1;
    //
    OSCCON.IRCF2 = 1;
    OSCCON.IRCF1 = 1;
    OSCCON.IRCF0 = 1;
    //ポートの設定
    TRISA = 0b0000000000;
    TRISB = 0b00000001;
    TRISC = 0b00000000;
    //□□□□□□□□の初期化
    init_sdc();
    //
    while (1) {
        SwitchONcheck();
        LED = 1;
        organProc();
        LED = 0;
    }
}
```

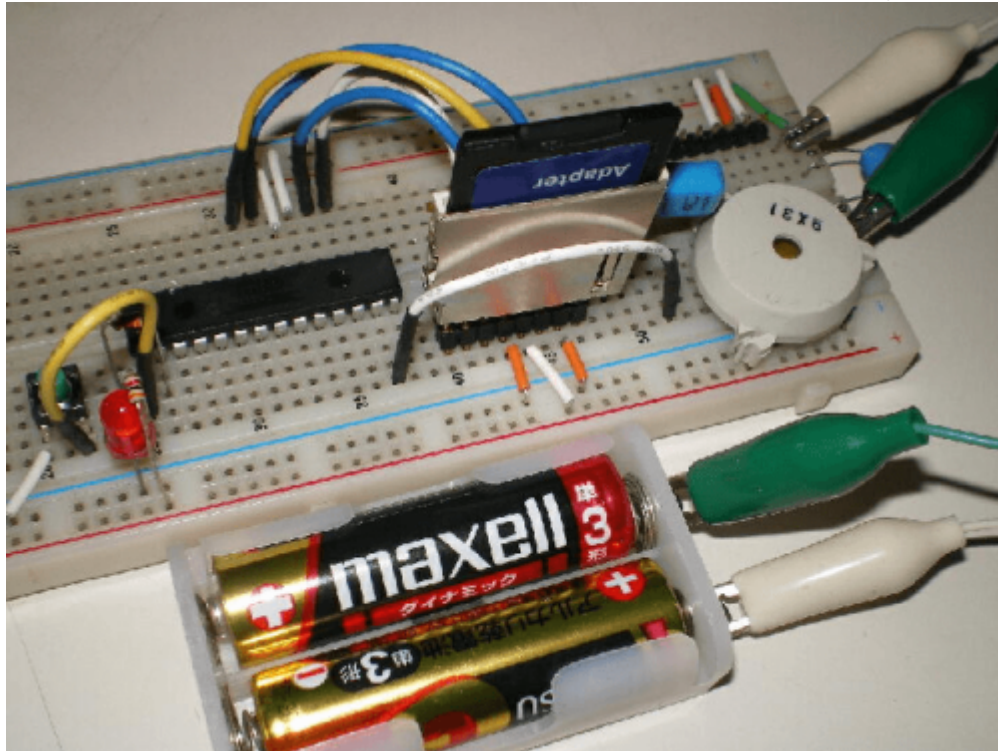
```

}
}

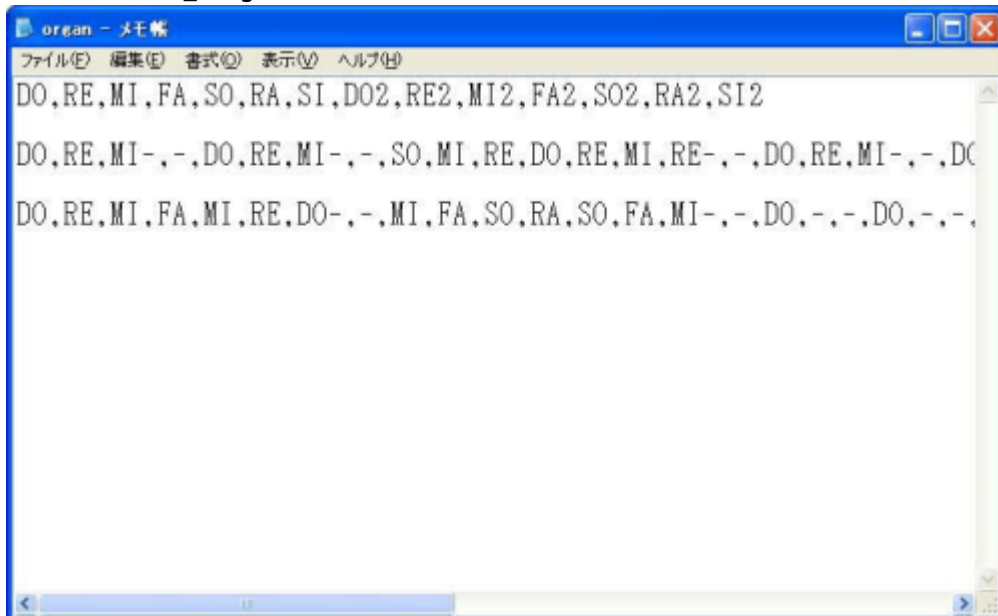
//*****
*
```

動作確認

左側から、スタート用プッシュスイッチ□LED□PIC18F2550□乾電池(単三2本)□SDカード、圧電スピーカ



音譜の例です□“organ.txt”をメモ帳で開いたところです。



<チューリップ> DO,RE,MI-,-,DO,RE,MI-,-,SO,MI,RE,DO,RE,MI,RE-,-,DO,RE,MI-,-,DO,RE,MI-,-

,SO,MI,RE,DO,RE,MI,DO-,-,SO, SO,MI,SO,RA,RA,SO-,-,MI,MI,RE,RE,DO-

<かえるの歌> DO,RE,MI,FA,MI,RE,DO-,-,MI,FA,SO,RA,SO,FA,MI-,-,DO-,-,DO-,-,DO-,-,DO-,-,
,DO,DO,RE,RE,MI,MI,FA,FA, MI-,RE-,DO-

如何でしょうか? もう少し滑らか(自然)に演奏させるには、表記をもっと工夫する必要がありますが、用途によってはこれで十分なのは。。。 😊

著作権表示 **copyright notice**

このページは稲崎様の閉鎖したHPのコピーで、著作権は稲崎様にあります。 [詳細](#) This page is a copy of Mr. Inasaki's closed website, and the copyright is held by him. [Details](#)

From:

<http://www.deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:otherpic:178>

Last update: **2025/10/17 14:29**

