

## 20

## おわりに

このテキストは、2014年7月1日時点での完成版です。

しかし、これで終わりというわけではありません。

より良い教材を目指して、これからも修正、変更を繰り返していくでしょう。

マイコン技術に限らず、世の中のすべてのものが改良され発展し進化していきます。

われわれはそれらに遅れないように日々学習し、新しい技術、新しい概念を習得していかなければなりません。

このテキストを学習した皆さんも、この先を、より良いプログラムを目指して、日々精進して行ってください。

ここまで学習してきた皆さんは、マイコンプログラムの基本は身についたと思います。

これから先は各自が自分の興味に応じてそれぞれの道を進んでください。

自ら興味を持ち、進んで学習すること、それが皆さんに対して著者が真に望むことなのです。

では、この先のまた別の場所で。



## 21

## 著作権に関すること

以下に著作権に関することをまとめておきます。

本文中で利用した、マニュアル、フリー画像などをまとめておきます。

ここで出典を記載していない画像については、著者自身が描いたものです。

- レジスタの説明は H8/3694 グループ ハードウェアマニュアル, ルネサスエレクトロニクスから引用しました。
- アイコン : WPZOOM, <http://www.wpzoom.com/> および 商用利用可の WEB 素材が無料な素材屋, <http://ec-sozai.net/> のアイコンを TEX 用にデータ処理しました。
- P.62 図 3.4:公開されている資料 (MB-RS10Schema.pdf) から転載
- P.186 図 7.9:ロータリスイッチのマニュアルから転載
- P.327 図 12.7:PSD センサのマニュアルから転載
- PSD センサ基板作成 : 電子情報技術科 遠藤幸さん
- コントローラ基板作成:電子情報技術科 藤田聖さん
- TEX マクロの一部は中国職業能力開発大学校の杉山誠先生に頂いたものを変更して利用しました。
- モーション作成ソフトや一部画像、説明については、著者のホームページ (<http://0n0.fc2web.com/>) でも公開しています。いずれも著者本人が製作したものです。



## 22

## 謝辞

電子情報技術科の職員の皆様には、常日頃から何かにつけお世話になっております。

東北職業能力開発大学校の職員の皆様にも、様々な便宜を図っていただいております。

この場を借りてお礼申し上げます。

また、総合製作実習の担当学生、大仁田優喜さん、千葉真悟さん、藤田聖さんにはロボットのコントロール基板作成など協力していただきました。

どうもありがとうございました。



## 関連図書

- [1] H8/3694 グループ ハードウェアマニュアル, ルネサスエレクトロニクス
- [2] H8S、H8/300 シリーズ C/C++コンパイラ、アセンブラ、最適化リンカージェディタ コンパイラパッケージ Ver.6.01 ユーザーズマニュアル, ルネサスエレクトロニクス
- [3] MB-H8A 取扱説明書, サンハヤト株式会社
- [4] MB-RS10 取扱説明書, サンハヤト株式会社
- [5] 島田義人 編著, H8/Tiny マイコン完璧マニュアル, CQ 出版
- [6] 株式会社富士通ラーニングメディア 編著, 標準テキスト 組込みプログラミング ハードウェア基礎, 技術評論社
- [7] 堀桂太郎 著, H8 マイコン入門, 東京電機大学出版局
- [8] Miyata's Robot Factory, <http://www.geocities.jp/mimiin/>
- [9] 回路図エディタ BSch 用ライブラリ, masahiro's electronic lab, <http://www.eleclabo.com/>
- [10] ROTARY CODED SWITCHES S-1000A/2000A カタログ
- [11] WPZOOM, <http://www.wpzoom.com/>
- [12] 商用利用可の WEB 素材が無料な素材屋, <http://ec-sozai.net/>






# A

## 課題解答例

### A.1 ビット演算

#### A.1.1 課題 2.2.1 解答例

 e02211or01.c

```
1  /******  
2  /*  
3  /*  DESCRIPTION :OR の演算  
4  /*  CPU TYPE   :PC  
5  /*  NAME       :Ono Yasuji  
6  /*  
7  /******  
8  
9  #include<stdio.h>  
10  
11 int main(void)  
12 {  
13     printf("0 \ | 0 = %d\n", 0|0);  
14     printf("0 \ | 1 = %d\n", 0|1);  
15     printf("1 \ | 0 = %d\n", 1|0);  
16     printf("1 \ | 1 = %d\n", 1|1);  
17     return(0);  
18 }  
19 }
```

*End Of List*

#### 実行結果 (Borland C++ Compiler 5.5)

```
>bcc32 e02211or01.c ↵  
  
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland  
e02211or01.c:  
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland  
  
>e02211or01.exe ↵  
  
0 | 0 = 0  
0 | 1 = 1  
1 | 0 = 1  
1 | 1 = 1  
  
>
```

---

 ④ e02211not01.c
 

---

```

1  /*****
2  /*
3  /* DESCRIPTION :not の演算
4  /* CPU TYPE   :PC
5  /* NAME       :Ono Yasuji
6  /*
7  /*****
8
9  #include<stdio.h>
10
11 int main(void)
12 {
13     printf("\! 0 = %d\n", !0);
14     printf("\! 1 = %d\n", !1);
15
16     return(0);
17 }
```

---

*End Of List*


---

 📄 実行結果 (Borland C++ Compiler 5.5)
 

---

```

>bcc32 e02211not01.c ↵
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
e02211not01.c:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland

>e02211not01.exe ↵

!0 = 1
!1 = 0

>
```

---

 ④ e02211xor01.c
 

---


```

1  /*****
2  /*
3  /* FILE       :03xor01.c
4  /* DESCRIPTION :XOR の演算
5  /* CPU TYPE   :PC
6  /* NAME       :Ono Yasuji
7  /*
8  /*****
9
10 #include<stdio.h>
11
12 int main(void)
13 {
14     int i;
15
16     printf("0 \^ 0 = %d\n", 0^0);
17     printf("0 \^ 1 = %d\n", 0^1);
18     printf("1 \^ 0 = %d\n", 1^0);
19     printf("1 \^ 1 = %d\n", 1^1);
20
21     return(0);
22 }
```

---

*End Of List*


---


 実行結果 (Borland C++ Compiler 5.5)

```
>bcc32 e02211xor01.c ↵
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
e02211xor01.c:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland

>e02211xor01.exe ↵
0 | 0 = 1
0 | 1 = 0
1 | 0 = 0
1 | 1 = 1

>
```

## A.1.2 課題 2.4.1 解答例


 e02411hyouji01.c

```
1  /*****
2  /*
3  /*  DESCRIPTION :10 進数を 2 進数、16 進数で表示 (1 バイト)
4  /*  CPU TYPE   :PC
5  /*  NAME       :Ono Yasuji
6  /*
7  /*****
8
9  #include<stdio.h>
10
11 #define NUM 200
12
13 int main(void)
14 {
15     int i, n=NUM, j=128;
16
17     printf("decimal %d: ", n);
18     printf("hexadecimal %x: ", n);
19     printf("binary ");
20     for(i=0; i<8; i++){
21         printf("%d", n/j);
22         n=n%j;
23         j=j/2;
24     }
25     printf("\n");
26     return(0);
27 }
28 }
```

---

*End Of List*

---

 実行結果 (Borland C++ Compiler 5.5)

```
>bcc32 e02411hyouji01.c ↵
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
e02411hyouji01.c:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland

>e02411hyouji01.exe ↵
decimal 200: hexadecimal c8: binaly 11001000

>
```

## A.1.3 課題 2.5.1 解答例


## e02511hyouji01.c

```

1  /*****
2  /*
3  /*  DESCRIPTION :10 進数を 2 進数、16 進数で表示 (1 バイト)
4  /*  CPU TYPE    :PC
5  /*  NAME       :Ono Yasuji
6  /*
7  /*****
8
9  #include<stdio.h>
10
11 #define NUM 200
12
13 int main(void)
14 {
15     int n=NUM;
16
17     printf("decimal %d: ", n);
18     printf("hexadecimal %x: ", n);
19     printf("binary ");
20     printf("%d", n/128);
21     n=n & (128-1);
22     printf("%d", n/64);
23     n=n & (64-1);
24     printf("%d", n/32);
25     n=n & (32-1);
26     printf("%d", n/16);
27     n=n & (16-1);
28     printf("%d", n/8);
29     n=n & (8-1);
30     printf("%d", n/4);
31     n=n & (4-1);
32     printf("%d", n/2);
33     n=n & (2-1);
34     printf("%d", n);
35     printf("\n");
36
37     return(0);
38 }

```

End Of List

 実行結果 (Borland C++ Compiler 5.5)

```

>bcc32 02511hyouji01.c
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
02511hyouji01.c:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland

>02511hyouji01.exe

decimal 200: hexadecimal c8: binaly 11001000

>

```

## A.2 LED の点滅

## A.2.1 課題 4.9.1 解答例

## e01\_LED07.c

```

1  /*****
2  /*
3  /* FILE      :01_LED07.c
4  /* DESCRIPTION :LED の消灯が増える
5  /* CPU TYPE   :H8/3694F
6  /*
7  /* LED0 P85
8  /* LED1 P86
9  /* LED2 P87
10 /*
11 /* This file is generated by Renesas Project Generator (Ver.4.9).
12 /*
13 /*****
14
15 #define PCR8 (*((volatile unsigned char *)0xFFEB))
16 #define PDR8 (*((volatile unsigned char *)0xFFDB))
17
18 #define LED_SHIFT 5
19 #define LED_MAX 0x07
20 #define LED_MASK (LED << LED_SHIFT)
21
22 #define WAIT_LOOP 0x3FFFFFL
23
24 void WaitLoop(void)
25 {
26     unsigned long count;
27     for(count=0; count<WAIT_LOOP; count++){
28         ;
29     }
30 }
31
32 void main(void)
33 {
34     unsigned char led_data=0x07; /* LED はすべて点灯 */
35     int i;
36
37     /*****
38     LED 接続端子の初期化
39     *****/
40     PCR8 |= LED_MASK;
41
42     /*****
43     LED の初期状態
44     *****/
45     PDR8 &= (~((led_data<<LED_SHIFT) & LED_MASK)); /* 0 をセット */
46     PDR8 |= (~(led_data<<LED_SHIFT)) & LED_MASK; /* 1 をセット */
47
48     while(1){
49         for(i=0, led_data=LED_MAX; i<3; i++, led_data<<=1){
50             PDR8 &= (~((led_data<<LED_SHIFT) & LED_MASK)); /* 0 をセット */
51             PDR8 |= (~(led_data<<LED_SHIFT)) & LED_MASK; /* 1 をセット */
52             WaitLoop();
53         }
54     }
55 }
56 }

```

---

*End Of List*

---

## A.2.2 課題 4.10.1 解答例

## e01\_LED08.c

```

1  /*****
2  /*
3  /* FILE      :e01_LED08.c
4  /* DESCRIPTION :LED 点灯の往復
5  /* CPU TYPE   :H8/3694F
6  /*
7  /* LED0 P85
8  /* LED1 P86
9  /* LED2 P87
10 /*
11 /* This file is generated by Renesas Project Generator (Ver.4.9).
12 /*
13 /*****
14

```

```

15 #define PCR8 (*((volatile unsigned char *)0xFFEB))
16 #define PDR8 (*((volatile unsigned char *)0xFFDB))
17
18 #define LED_SHIFT 5
19 #define LED0 0x01
20 #define LED2 0x04
21 #define LED_MASK (LED << LED_SHIFT)
22
23 #define WAIT_LOOP 0x3FFFFFFL
24
25 void WaitLoop(void)
26 {
27     unsigned long count;
28     for(count=0; count<WAIT_LOOP; count++){
29         ;
30     }
31 }
32
33 void main(void)
34 {
35     unsigned char led_data=0x01; /* LED1 のみ点灯 */
36
37     /*****
38      LED 接続端子の初期化
39      *****/
40     PCR8 |= LED_MASK;
41
42     /*****
43      LED の初期値設定
44      *****/
45     PDR8 &= (~((led_data<<LED_SHIFT) & LED_MASK)); /* 0 をセット */
46     PDR8 |= (~(led_data<<LED_SHIFT)) & LED_MASK; /* 1 をセット */
47
48     while(1){
49         for(led_data=LED0; led_data<LED2; led_data<=1){
50             PDR8 &= (~((led_data<<LED_SHIFT) & LED_MASK)); /* 0 をセット */
51             PDR8 |= (~(led_data<<LED_SHIFT)) & LED_MASK; /* 1 をセット */
52             WaitLoop();
53         }
54         for(led_data=LED2; led_data>LED0; led_data>=1){
55             PDR8 &= (~((led_data<<LED_SHIFT) & LED_MASK)); /* 0 をセット */
56             PDR8 |= (~(led_data<<LED_SHIFT)) & LED_MASK; /* 1 をセット */
57             WaitLoop();
58         }
59     }
60 }
61 }

```

---

*End Of List*

---

### A.2.3 課題 4.11.1 解答例

#### e01\_LED09.c

```

1  /*****
2  /*
3  /* FILE           :e01_LED09.c
4  /* DESCRIPTION   :LED0,2 の点滅 (XOR 版)(関数化)
5  /* CPU TYPE     :H8/3694F
6  /*
7  /* LED0 P85
8  /* LED1 P86
9  /* LED2 P87
10 /*
11 /* This file is generated by Renesas Project Generator (Ver.4.9).
12 /*
13 /*****
14
15 #define PCR8 (*((volatile unsigned char *)0xFFEB))
16 #define PDR8 (*((volatile unsigned char *)0xFFDB))
17
18 #define LED0 0x01
19 #define LED1 0x02
20 #define LED2 0x04
21
22 #define LED_SHIFT 5
23 #define LED_MASK ((LED0 | LED1 | LED2) << LED_SHIFT)
24
25 #define WAIT_LOOP 0x3FFFFFFL
26
27 void WaitLoop(void)
28 {
29     unsigned long count;
30     for(count=0; count<WAIT_LOOP; count++){
31         ;
32     }
33 }

```

```

34 }
35
36 /*****
37 LED 接続端子の初期化 (LED はすべて消灯)
38 *****/
39 void LedInit(void)
40 {
41     PCR8 |= LED_MASK;
42     PDR8 |= LED_MASK;
43 }
44
45 /*****
46 1 を指定した LED を点灯
47 *****/
48 void LedOn(unsigned char led_data)
49 {
50     PDR8 &= (~(led_data << LED_SHIFT) & LED_MASK); /* 0 をセット */
51 }
52
53 /*****
54 1 を指定した LED を消灯
55 *****/
56 void LedOff(unsigned char led_data)
57 {
58     PDR8 |= ((led_data << LED_SHIFT) & LED_MASK); /* 1 をセット */
59 }
60
61 /*****
62 指定した LED を点灯
63 それ以外は消灯
64 *****/
65 void LedSet(unsigned char led_data)
66 {
67     LedOn(led_data);
68     LedOff(~led_data);
69 }
70
71 void main(void)
72 {
73     unsigned long led_data=0;
74     LedInit();
75     while(1){
76         LedSet(led_data);
77         WaitLoop();
78         led_data ^= (LED0 | LED2);
79     }
80 }
81
82 }

```

---

*End Of List*

---

## A.3 タクトスイッチ

### A.3.1 課題 7.1.1 解答例

sw.h、sw.c は P.179 で定義したものを使いました。

---

#### 📄 e02\_SW04.c

---

```

1  /*****
2  /*
3  /* FILE      :e02_SW04.c
4  /* DESCRIPTION :タクトスイッチを押すたびに LED の点灯が左シフト
5  /* CPU TYPE   :H8/3694F
6  /*
7  /* タクトスイッチ
8  /* SW0 P14
9  /*
10 /*****
11
12 #include "led.h"
13 #include "sw.h"
14
15 void main(void)
16 {
17     unsigned long led_data=LED0;
18

```

```

19 LedInit();      /* LED 接続ポートの初期化 */
20 SwInit();      /* タクトスイッチ接続ポートの初期化 */
21
22 while(1){
23   for(i=0; i<3; i++){
24     LedSet(led_data << i);
25     SwWaitPush(); /* タクトスイッチが押されるまで待つ */
26     SwWaitDetach(); /* タクトスイッチが離されるまで待つ */
27   }
28 }
29 }

```

---

*End Of List*

---

## A.4 ロータリスイッチ

### A.4.1 課題 7.2.1 解答例

sw.h、sw.c は P.194 で定義したものを使いました。

---

#### 📄 e03\_ROT SW02.c

---

```

1  /******
2  /*
3  /* FILE      :e03_ROT SW02.c
4  /* DESCRIPTION :ロータリスイッチの値によって LED の動作を変更
5  /* CPU TYPE   :H8/3694F
6  /*
7  /* ロータリスイッチ
8  /* ROT SW1 P50
9  /* ROT SW2 P51
10 /* ROT SW3 P52
11 /* ROT SW4 P53
12 /*
13 /******
14
15 #include "led.h"
16 #include "sw.h"
17
18 #define LED_MAX (LED0 | LED1 | LED2)
19
20 void LedRightShift( void )
21 {
22   unsigned char led_data;
23
24   while(1){
25     for(led_data=LED2; led_data>0; led_data>>=1){
26       LedSet(led_data);
27       WaitLoop();
28     }
29   }
30 }
31
32 void LedLeftShift( void )
33 {
34   unsigned char led_data;
35
36   while(1){
37     for(led_data=LED0; led_data<=LED_MAX; led_data<<=1){
38       LedSet(led_data);
39       WaitLoop();
40     }
41   }
42 }
43
44 void LedCountUp( void )
45 {
46   unsigned char led_data;
47
48   while(1){
49     for(led_data=0; led_data<=LED_MAX; led_data++){
50       LedSet(led_data);
51       WaitLoop();
52     }
53   }
54 }
55
56 void main(void)
57 {

```



```

58 LedInit(); /* LED 接続ポートの初期化 */
59 RotSwInit(); /* ロータリスイッチ接続ポートの初期化 */
60
61 switch(RotSwGet()){
62 case 0:
63     LedRightShift();
64     break;
65 case 1:
66     LedLeftShift();
67     break;
68 default:
69     LedCountUp();
70     break;
71 }
72 }

```

---

*End Of List*

---

## A.5 IRQ(Interrupt ReQuest)

### A.5.1 課題 8.1.1 解答例

#### intprg.c(修正後)

```

// vector 14 IRQ0
//__interrupt(vect=14) void INT_IRQ0(void) { /* sleep(); */}

```

#### e04\_IRQ02.c

```

1  /*****
2  /*
3  /* FILE      :e04_IRQ02.c
4  /* DESCRIPTION :タクトスイッチを押したら LED が全て点灯
5  /* CPU TYPE   :H8/3694F
6  /*
7  /* タクトスイッチ
8  /* SWO P14/IRQ0
9  /*
10 /*****
11
12 #include <machine.h>
13 #include "led.h"
14 #include "iodef.h"
15
16 static int irq_flag=0;
17
18 void IrqInit(void)
19 {
20     set_imask_ccr(1); /* 割り込み不可 */
21     IO.PMR1.BIT.IRQ0=1; /* IRQ0 入力端子 */
22     IO.PUCR1.BIT.B4=1; /* プルアップを設定 */
23     IEGR1.BIT.IEGO=0; /* IRQ0 端子入力の立ち下がりエッジを検出 */
24     IENR1.BIT.IENO=1; /* IRQ0 端子の割り込み要求 */
25     IRR1.BIT.IRRI0=0; /* IRQ0 割り込み要求フラグクリア */
26     set_imask_ccr(0); /* 割り込み許可 */
27 }
28
29 void main(void)
30 {
31     unsigned char led_data;
32
33     LedInit(); /* LED 接続ポートの初期化 */
34     IrqInit(); /* IRQ 初期化 */
35     while(irq_flag==0){
36     };
37 }
38 set_imask_ccr(1); /* 割り込み不可 */
39 while(1){
40     for(led_data=0; led_data<=LED_MAX; led_data++){
41         LedSet(led_data);
42         WaitLoop();
43     }

```

```

44 }
45 }
46
47 __interrupt(vect=14) void INT_IRQ0(void)
48 {
49     irq_flag=1;
50 }

```

---

*End Of List*

---

## A.6 タイマー V

### A.6.1 課題 10.1.1 解答例

表 A.1 音階と周波数の関係

音階	周波数 (Hz)
ド	1046.50
シ	987.77
ラ♯(シ♭)	932.33
ラ	880.00
ソ♯(ラ♭)	830.61
ソ	783.99
ファ♯(ソ♭)	739.99
ファ	698.46
ミ	659.26
レ♯(ミ♭)	622.25
レ	587.33
ド♯(レ♭)	554.37
ド	523.25
シ	493.88
ラ♯(シ♭)	466.16
ラ	440.00
ソ♯(ラ♭)	415.30
ソ	392.00
ファ♯(ソ♭)	369.99
ファ	349.23
ミ	329.63
レ♯(ミ♭)	311.27
レ	293.66
ド♯(レ♭)	277.18
ド	261.63

**B**

関数仕様およびプログラム一覧  
(H8/3694F)

## B.1 LED 関連関数

LED 関連関数 (H8/3694F)

ヘッダファイル: led.h

LED0 P85

LED1 P86

LED2 P87

```
*****
void LedInit(void);
```

形式 : #include"led.h"  
 void LedInit(void);  
 機能 : LED で使用する I/O ポートを初期化する関数。  
 初期状態は LED すべて消灯。  
 LED を使用する前に実行すること。  
 引数 : なし  
 戻り値: なし

```
*****
void LedOn(unsigned char led_data);
```

形式 : #include"led.h"  
 void LedOn(unsigned char led\_data);  
 機能 : 1 で指定された LED を点灯する。  
 ビットパターンで同時に複数の LED を指定できる。  
 1 で指定された LED 以外 (0 で指定された LED) には変化が無い。  
 引数 : unsigned char led\_data  
 点灯する LED の指定 (LED0 or LED1 or LED2)  
 下位 3 ビット分のみ有効。  
 戻り値: なし

```
*****
void LedOff(unsigned char led_data);
```

形式 : #include"led.h"  
 void LedOff(unsigned char led\_data);  
 機能 : 1 で指定された LED を消灯する。  
 ビットパターンで同時に複数の LED を指定できる。  
 1 で指定された LED 以外 (0 で指定された LED) には変化が無い。  
 引数 : unsigned char led\_data  
 消灯する LED の指定 (LED0 or LED1 or LED2)  
 下位 3 ビット分のみ有効。  
 戻り値: なし

```
*****
void LedSet(unsigned char led_data);
```

形式 : #include"led.h"  
 void LedSet(unsigned char led\_data);  
 機能 : 1 で指定された LED を点灯し、  
 0 で指定された LED を消灯する。  
 ビットパターンで同時に 3 ビット分の LED を指定する。  
 引数 : unsigned char led\_data  
 点灯する LED の指定 (LED0 or LED1 or LED2)  
 下位 3 ビット分のみ有効。  
 戻り値: なし

### led.h

```
1 #ifndef _LED_H_
2 #define _LED_H_
3
4 #include"iodefine.h"
5
6 #define LEDDAT IO.PDR8.BYTE
7 #define LEDIO IO.PCR8
8
9 #define LED0 0x01
10 #define LED1 0x02
11 #define LED2 0x04
```

```

12
13 #define LED_SHIFT 5
14 #define LED_MASK ((LED0 | LED1 | LED2) << LED_SHIFT)
15
16 #define WAIT_LOOP 0x3FFFFFFL
17
18 void LedInit(void);
19 void LedOn(unsigned char led_data);
20 void LedOff(unsigned char led_data);
21 void LedSet(unsigned char led_data);
22 void WaitLoop(void);
23
24 #endif

```

---

*End Of List*

---



---

led.c

---

```

1  /*****
2  /*
3  /* FILE :led.c
4  /* DESCRIPTION :LED 用関数
5  /* CPU TYPE :H8/3694F
6  /*
7  /* LED0 P85
8  /* LED1 P86
9  /* LED2 P87
10 /*
11 *****/
12
13 #include"led.h"
14
15 #define WAIT_LOOP 0x3FFFFFFL
16
17 void WaitLoop(void)
18 {
19     unsigned long count;
20
21     for(count=0; count<WAIT_LOOP; count++){
22         ;
23     }
24 }
25
26 /*****
27 LED 接続端子の初期化 (LED はすべて消灯)
28 *****/
29 void LedInit(void)
30 {
31     LEDIO |= LED_MASK;
32     LEDDAT |= LED_MASK;
33 }
34
35 /*****
36 1 を指定した LED を点灯
37 *****/
38 void LedOn(unsigned char led_data)
39 {
40     LEDDAT &= (~(led_data << LED_SHIFT) & LED_MASK); /* 0 をセット */
41 }
42
43 /*****
44 1 を指定した LED を消灯
45 *****/
46 void LedOff(unsigned char led_data)
47 {
48     LEDDAT |= ((led_data << LED_SHIFT) & LED_MASK); /* 1 をセット */
49 }
50
51 /*****
52 指定した LED を点灯
53 それ以外は消灯
54 *****/
55 void LedSet(unsigned char led_data)
56 {
57     LedOn(led_data);
58     LedOff(~led_data);
59 }

```

---

*End Of List*

---

## B.2 スイッチ関連関数

タクトスイッチ関連関数 (H8/3694F ボード)

ヘッダファイル: sw.h

SWO P14

```
*****
```

```
void SwInit(void);
```

形式 : #include"sw.h"  
void SwInit(void);

引数: なし

戻り値: なし

解説: スイッチ読み取りで使用する I/O ポートを初期化する関数。  
端子をプルアップする。  
スイッチ読み取り関数を使用する前に実行すること。

```
*****
```

```
unsigned char SwGet(void);
```

形式 : #include"sw.h"  
unsigned char SwGet(void);

引数: なし

戻り値: unsigned char

解説: タクトスイッチの状態 (0:押されていない 1:押されている)  
タクトスイッチの状態を読み取る関数。  
チャタリング対策のため同じ値に安定するまで内部で 100 回  
読み出しと比較を行っている。10000 回数読み出しを行っても安定しなかった  
場合は最後に読み取った状態を返す。

```
*****
```

```
void SwWaitPush(void);
```

形式 : #include"sw.h"  
void SwWaitPush(void);

引数: なし

戻り値: なし

解説: タクトスイッチが押されるまで待つ関数。  
SwGet を利用している。

```
*****
```

```
void SwWaitDetach(void);
```

形式 : #include"sw.h"  
void SwWaitDetach(void);

引数: なし

戻り値: なし

解説: タクトスイッチが離されるまで待つ関数。  
SwGet を利用している。

ロータリスイッチ関連関数 (H8/3694F ボード)

ヘッダファイル : sw.h

ROTSW1 P50  
ROTSW2 P51  
ROTSW3 P52  
ROTSW4 P53

```
*****
void RotSwInit(void);
```

形式 : #include"sw.h"  
RotSwInit(void);

引数 : なし

戻り値 : なし

解説 : ロータリスイッチ読み取りで使用する I/O ポートを初期化する関数。  
各端子をプルアップする。  
ロータリスイッチ読み取り関数を使用する前に実行すること。

```
*****
unsigned char RotSwGet(void);
```

形式 : #include"sw.h"  
RotSwGet(void);

引数 : なし

戻り値 : unsigned char

ロータリスイッチの状態 (下位 4 ビットのみ : 0-15)。  
ロータリスイッチのつまみの数値と同じ値が返る。

解説 : ロータリスイッチの状態を読み取る関数。  
チャタリング対策は行っていない。

## sw.h

```

1  #ifndef _SW_H_
2  #define _SW_H_
3
4  #include "iodefine.h"
5
6  #define ROTSWIO  IO.PCR5
7  #define ROTSWDAT IO.PDR5.BYTE
8  #define ROTSWPU  IO.PUCR5.BYTE
9
10 #define SWIO  IO.PCR1
11 #define SWDAT IO.PDR1.BYTE
12 #define SWPU  IO.PUCR1.BYTE
13
14 #define ROTSW1 0x01
15 #define ROTSW2 0x02
16 #define ROTSW3 0x04
17 #define ROTSW4 0x08
18
19 #define SW0 0x01
20 #define SW_SHIFT 4
21
22 #define SW_ROT_MASK ( ROTSW1 | ROTSW2 | ROTSW3 | ROTSW4 )
23 #define SW_MASK ( SW0 << SW_SHIFT )
24
25 #define SW_LOOP 1000 /* タクトスイッチ確認最大回数、安定しなかったら最後の値 */
26 #define SW_COUNT 100 /* 値がこの回数同じだったら、スイッチの値とする */
27
28 void RotSwInit(void);
29 unsigned char RotSwGet(void);
30
31 void SwInit(void);
32 unsigned char SwGet(void);
33 void SwWaitPush(void);
34 void SwWaitDetach(void);
35
36 #endif
```

End Of List

## sw.c

```

1  /******
2  /*
3  /* FILE      :sw.c
4  /* DESCRIPTION :スイッチ用関数
5  /*
```

```

5  /* CPU TYPE      :H8/3694F                                */
6  /*                                                       */
7  /* ロータリスイッチ                                    */
8  /* ROTSW1 P50                                           */
9  /* ROTSW2 P51                                           */
10 /* ROTSW3 P52                                           */
11 /* ROTSW4 P53                                           */
12 /*                                                       */
13 /* タクトスイッチ                                    */
14 /* SW0      P14                                           */
15 /*                                                       */
16 /******
17
18 #include "sw.h"
19
20 /******
21   ロータリスイッチのつながった端子を初期化
22   *****/
23 void RotSwInit(void)
24 {
25     ROTSWIO &= (~SW_ROT_MASK);
26     ROTSWPU |= SW_ROT_MASK; /* プルアップの設定 */
27 }
28
29 /******
30   ロータリスイッチの状態を取得
31   *****/
32 unsigned char RotSwGet(void)
33 {
34     return ((~ROTSWDAT) & SW_ROT_MASK);
35 }
36
37 /******
38   タクトスイッチのつながった端子を初期化
39   *****/
40 void SwInit(void)
41 {
42     SWIO &= (~SW_MASK);
43     SWPU |= SW_MASK; /* プルアップ */
44 }
45
46 /******
47   タクトスイッチの状態を取得
48   *****/
49 unsigned char SwGet(void)
50 {
51     /* i:スイッチの値を確認した回数 */
52     /* count:スイッチの値が連続して同じだった回数 */
53     int i=0, count=0;
54
55     /* old_state:現在と異なる最近のスイッチの値 */
56     /* new_state:現在のスイッチの値 */
57     unsigned char new_state, old_state=255;
58
59     while((i<SW_LOOP) && (count<SW_COUNT)){
60         new_state = (SWDAT & SW_MASK);
61         if(old_state == new_state){
62             count++;
63         }else{
64             count = 0;
65             old_state = new_state;
66         }
67         i++;
68     }
69     return (((~new_state) & SW_MASK) >> SW_SHIFT);
70 }
71
72 /******
73   タクトスイッチが押されるまで待つ
74   *****/
75 void SwWaitPush(void)
76 {
77     while(!SwGet()){
78         ;
79     }
80 }
81
82 /******
83   タクトスイッチが離されるまで待つ
84   *****/
85 void SwWaitDetach(void)
86 {
87     while(SwGet()){
88         ;
89     }
90 }
91

```

---

*End Of List*

---



## B.3 音楽関連関数

音楽関連関数 (H8/3694F ボード)

ヘッダファイル: music.h

SP P76/TMOV

```

*****
void MusicInit(void);

形式 : #include"music.h"
       void MusicInit(void);
引数 : なし
戻り値: なし
解説 : 音楽演奏用のポート初期化し、演奏を始める関数。
       音楽を演奏する際に実行すること。

*****
void StopMusic(void);


形式 : #include"music.h"
       void StopMusic(void);
引数 : なし
戻り値: なし
解説 : 音楽の演奏を停止する関数。

*****
void StartMusic(void);

形式 : #include"music.h"
       void StartMusic(void);
引数 : なし
戻り値: なし
解説 : 音楽の演奏を開始する関数。

*****

```

 music.h

```

1  #ifndef _MUSIC_H_
2  #define _MUSIC_H_
3
4  /*-----*/
5  /*   ファイル名 :   music.h                               */
6  /*   内容 :   音楽用タイマ関数ヘッダファイル             */
7  /*                                                     */
8  /*   SP P76/TMOV                                       */
9  /*                                                     */
10 /*-----*/
11
12 //音の長さの定義 (すべての値が偶数になるように定義)
13 //奇数は拡張性のためにとっておく
14 //TEMPO:0.05s, L1:1.6s
15 #define TEMPO 30      /* L1 の値に反映させる */
16 #define L1  32*TEMPO //全音符長
17 #define L2  (L1/2)  //2 分音符長
18 #define L4  (L1/4)  //4 分音符長
19 #define L8  (L1/8)  //8 分音符長
20 #define L16 (L1/16) //16 分音符長
21 #define L32 (L1/32) //32 分音符長
22 #define F2  ((L1*3)/4) //付点 2 分音符長
23 #define F4  ((L1*3)/8) //付点 4 分音符長
24 #define L3  (L4/3)   //3 連符長
25 #define L6  (L4/6)   //6 連符長
26
27 //音の高さ (音程) の定義
28 #define do0  149 //ド
29 #define dos0 141 //ド#
30 #define re0  133 //レ
31 #define res0 126 //レ#
32 #define mi0  119 //ミ
33 #define fa0  112 //ファ

```

```

34 #define fas0 106 //ファ#
35 #define so0 100 //ソ
36 #define sos0 94 //ソ#
37 #define ra0 89 //ラ 880Hz
38 #define ras0 84 //ラ#
39 #define si0 79 //シ
40 #define do1 75 //ド
41 #define dos1 70 //ド#
42 #define re1 67 //レ
43 #define res1 63 //レ#
44 #define mi1 59 //ミ
45 #define fa1 56 //ファ
46 #define fas1 53 //ファ#
47 #define so1 50 //ソ
48 #define sos1 47 //ソ#
49 #define ra1 44 //ラ 1760Hz
50 #define ras1 42 //ラ#
51 #define si1 40 //シ
52 #define do2 37 //ド
53
54 //音のデータ
55 struct NoteData{
56     unsigned int Freq; //音程
57     unsigned int Len; //長さ
58 };
59
60 void MusicInit(void);
61 void StopMusic(void);
62 void StartMusic(void);
63 #endif

```


---

*End Of List*

---



---

 **music.c**

---

```

1  /*****
2  /*
3  /* FILE           :music.c
4  /* DESCRIPTION   :音楽用関数
5  /* CPU TYPE     :H8/3694F
6  /*
7  /* SP P76/TMOV
8  /*
9  /*****
10
11 /* インクルードファイル */
12 #include<machine.h>
13 #include"iodefine.h"
14 #include "music.h" /* 音階や音の長さの定義 */
15 #include "music_data" /* 楽譜データ */
16
17 struct NoteData CNote; /* Freq: 音程, Len: 音の長さ */
18 int NoteCnt=0; /* 音符の何番目を演奏しているか */
19 int ZeroCnt=30; /* 音の最後の消音 50ms 1.6mS 単位 */
20 unsigned int fdata; /* 音階を格納する変数 */
21 int note_max=NOTEMAX; /* 楽譜データの要素数 */
22
23 /*****
24 音の開始・停止操作 (タイマ V)
25 *****/
26 void StopPwmTimer(void)
27 {
28     TV.TCSR.V.BIT.OS=0;
29 }
30 void StartPwmTimer(void)
31 {
32     TV.TCSR.V.BIT.OS=3; /* コンペアマッチ A でトグル出力 */
33 }
34 /*****
35 音の長さをカウントするタイマの初期化・リセット操作 (タイマ A)
36 *****/
37 void StopLengthTimer(void)
38 {
39     TA.TMA.BIT.CKSI=0xC;
40 }
41 void StartLengthTimer(void)
42 {
43     TA.TMA.BIT.CKSI=5; /* インターバルタイマ 128 分周 */
44 }
45
46 /*****
47 音階タイマの初期化 20MHz 128 分周 (タイマ V)
48 P76/TMOV に出力
49 *****/
50 void PwmTimerInit(void)

```

```

51 {
52     TV.TCRV0.BIT.CKS=3;
53     TV.TCRV1.BIT.ICKS=1; /* 128 分周 */
54     TV.TCRV0.BIT.CCLR=1; /* コンペアマッチ A でクリア */
55     TV.TCSR.V.BIT.OS=3; /* コンペアマッチ A でトグル出力 */
56 }
57
58 /*****
59     音の長さ用タイマの初期化 1.6ms ごとに割り込みをかける設定
60     20MHz 128 分周 (タイマ A)
61 *****/
62 void LengthTimerInit(void)
63 {
64     set_imask_ccr(1);
65     IENR1.BIT.IENTA=1; /* タイマ A のオーバーフロー割り込み要求 */
66     IRR1.BIT.IRRTA=0;
67     set_imask_ccr(0);
68 }
69
70 /*****
71     音をセット
72 *****/
73 void SetData(unsigned int freq)
74 {
75     TV.TCOR.A=freq; /* ON, OFF は半々 エンベロープを入れるならここを変更 */
76 }
77
78 /*****
79     音の長さのタイマ割り込みフラグをクリア
80 *****/
81 void ClearLegthFlag(void)
82 {
83     IRR1.BIT.IRRTA=0;
84 }
85
86 /*-----
87     以下、ハードウェア依存性の無いプログラム
88     ただし、二つのタイマの割り込み関数は修正の必要あり
89     (今の場合音はトグル出力で出しているの割り込み関数は無い)
90 -----*/
91
92 /*****
93     最初のデータをセット 休符のときの処理
94 *****/
95 void StopMusic(void){
96     StopPwmTimer();
97     StopLengthTimer();
98 }
99
100 void StartMusic(void){
101     StartPwmTimer();
102     StartLengthTimer();
103 }
104
105 void SetInitData(void)
106 {
107     CNote=Note[NoteCnt];
108     if(CNote.Freq == 0){
109         StopPwmTimer();
110     }else{
111         SetData(CNote.Freq);
112         StartPwmTimer();
113     }
114 }
115
116 /*****
117     音楽演奏のための初期化関数
118 *****/
119 void MusicInit(void)
120 {
121     PwmTimerInit();
122     LengthTimerInit();
123     SetInitData();
124     StartLengthTimer();
125 }
126
127 /*****
128     IMIA1 割り込み関数 (音の長さ)1.6ms ごとの割り込み
129 *****/
130 __interrupt(vect=19) void INT_TimerA(void)
131 {
132     ClearLegthFlag();
133     if(CNote.Len>0){
134         CNote.Len--; /* エンベロープを入れるならここで対応 */
135     }else if(NoteCnt<(note_max-1) && ZeroCnt>0){ /* 消音を入れて同じ音が続いたときの対処 */
136         if(ZeroCnt==50){
137             StopPwmTimer();
138         }
139         ZeroCnt--;
140     }else if(NoteCnt<(note_max-1) && ZeroCnt<=0){ /* CNote.Len==0 音の終了時 */

```

```
141     ZeroCnt=50;
142     NoteCnt++;
143     CNote=Note[NoteCnt];
144     fdata=CNote.Freq; /* Freq:音階 タイマ A0 */
145     if(fdata==0){ /* 音階データが0の場合には音を出さない */
146         StopPwmTimer(); /* タイマ A0 を停止 */
147     }else{
148         SetData(fdata);
149         StartPwmTimer();
150     }
151 }else{ /* 楽譜の最後で音を止める */
152     StopMusic();
153 }
154 }
```

---

*End Of List*

---

## B.4 シリアル通信関連関数

■ファイル (sci.c)、ヘッダファイル (sci.h) ■

-----  
**void Sci1Init(BaudRate b)**

形式:

```
#include "sci.h"
void Sci1Init(BaudRate b);
```

機能: シリアルポートの初期化を行う関数。  
シリアルポートを用いる場合には、必ず実行する。

引数: BaudRate b(ボーレートを指定、br19200、br38400、br57600)

返却値: 無し

-----  
**void Rx1BuffClear(void)**

形式:

```
#include "sci.h"
void Rx1BuffClear(void);
```

機能: 受信バッファクリア関数(割り込み有り)。  
受信バッファをクリアする関数。

引数: 無し

返却値: 無し

-----  
**int Sci1Write(char c)**

形式:

```
#include "sci.h"
int Sci1Write(char c);
```

機能: 1文字送信関数。  
0x200000 回送信を行って送信できない場合には関数を抜ける。

引数: char c(送信する文字)

返却値: 送信成功なら 0、失敗なら-1 を返す。

-----  
**char Sci1Read(void)**

形式:

```
#include "sci.h"
char Sci1Read(void);
```

機能: 1文字受信関数(割り込み、バッファ有り)。

引数: 無し

返却値: 受信した 1文字を返す。

-----  
**int Sci1Putchar(char c)**

形式:

```
#include "sci.h"
int Sci1Putchar(char c);
```

機能: 1文字送信関数。  
改行 (\n) は CR+LF に変換。

引数: char c(送信する文字。改行 (\n) は CR+LF に変換)

返却値: 送信成功なら 0、失敗なら-1 を返す。

-----  
**int Sci1Puts(char \*str)**

形式:

```
#include "sci.h"
int Sci1Puts(char *str);
```

機能: 文字列送信関数。  
終端文字 (\0) まで文字列を送信。  
改行 (\n) は CR+LF に変換。

引数: char \*str(送信する文字列。最後は終端文字 (\0)。改行 (\n) は CR+LF に変換)

返却値: 送信失敗なら 0、成功なら送信した文字数を返す。

## ■ファイル (sci.c)、ヘッダファイル (sci.h) ■

```
char Sci1Getchar(void)
```

形式:

```
#include "sci.h"
char Sci1Getchar(void);
```

機能: 1 文字受信関数 (割り込み有り)。

1 文字を受信する関数。

CR は改行 (\n) に変換。

引数: 無し

返却値: 受信した 1 文字を返す。

```
void Sci1Gets(char *str)
```

形式:

```
#include "sci.h"
void Sci1Gets(char *str);
```

機能: 文字列受信関数 (割り込み有り)。

改行 (CR) まで文字列を受信する関数。

CR は改行 (\n) に変換。

引数: char \*str (受信する文字列。改行 (CR) まで受信)

返却値: 無し。

## sci.h

```
1 #ifndef _SCI_H_
2 #define _SCI_H_
3
4 /* レジスタ定義 */
5 #define TXD1F SCI3.SSR.BIT.TDRE //送信データフラグ
6 #define RXD1F SCI3.SSR.BIT.RDRF //受信データフラグ
7 #define TD1D SCI3.TDR //送信データ
8 #define RD1D SCI3.RDR //受信データ
9
10 #define SCI_RX_BUFF_SIZE 32
11 #define SCI_RX_BUFF_MASK (SCI_RX_BUFF_SIZE-1)
12
13 typedef enum{
14     br19200=32,
15     br38400=15,
16     br57600=10
17 }BaudRate;
18
19 typedef struct{
20     unsigned short start_index;
21     unsigned short data_count;
22     char data_buff[SCI_RX_BUFF_SIZE];
23 }SciRxBuffer;
24
25 void Sci1Init(BaudRate b);
26 void Rx1BuffClear(void);
27
28 int Sci1Write(char c);
29 char Sci1Read(void);
30
31 int Sci1Putchar(char c);
32 int Sci1Puts(char *str);
33
34 char Sci1Getchar(void);
35 void Sci1Gets(char *str);
36
37 #endif
```

End Of List

## sci.c

```
1 #include <machine.h>
2 #include "iodefine.h"
3 #include "sci.h"
4
5 #define CHECK_LOOP 0x200000
6 #define CR 0x0D
7 #define LF 0x0A
8
9 static SciRxBuffer rx1_buff;
```

```

10
11 /*****
12 シリアルポートを初期化 :Sci1Init
13 *****/
14 void Sci1Init(BaudRate b)
15 {
16     set_imask_ccr(1); /* 割込み不可 */
17     SCI3.SCR3.BYTE = 0x00; /* TE、RE クリア CKE 初期化 */
18     SCI3.SMR.BYTE = 0x00; /* 調歩同期,8 ビット,パリティなし,ストップ 1 ビット n=0 */
19     SCI3.BRR = b;
20     IO.PMR1.BIT.TXD = 1; /* TXD ポートイネーブル */
21     SCI3.SCR3.BYTE = 0x70; /* TE RE 受信割込み */
22     set_imask_ccr(0); /* 割込み許可 */
23 }
24
25 /*****
26 1 バイト送信関数 :Sci1Write
27 *****/
28 int Sci1Write(char c)
29 {
30     unsigned long i;
31     unsigned short write_flag=0; /* 送信データ書き込みフラグ */
32
33     for(i=0; i<CHECK_LOOP; i++){
34         if(TXD1F){ /* 送信データ書き込み可能になるまで待つ */
35             TXD1F=0;
36             write_flag=1;
37             break;
38         }
39     }
40     if(write_flag){
41         TD1D=c; /* 送信データを格納 */
42         return(0);
43     }
44     return(-1);
45 }
46
47 /*****
48 1 バイト受信関数 (割り込み有り)
49 *****/
50 __interrupt(vect=23) void INT_SCI3(void)
51 {
52     char msg;
53     int index;
54
55     if(RXD1F){
56         RXD1F=0;
57         msg=RD1D;
58         if(rx1_buff.data_count < SCI_RX_BUFF_SIZE){
59             index=(rx1_buff.start_index+rx1_buff.data_count) & SCI_RX_BUFF_MASK;
60             rx1_buff.data_buff[index]=msg;
61             rx1_buff.data_count++;
62         }
63     }
64 }
65
66 int Sci1Rx1(char *msg)
67 {
68     if(rx1_buff.data_count > 0){
69         *msg=rx1_buff.data_buff[rx1_buff.start_index];
70         rx1_buff.start_index=(rx1_buff.start_index+1) & SCI_RX_BUFF_MASK;
71         rx1_buff.data_count--;
72         return(0);
73     }
74     *msg=0;
75     return(-1);
76 }
77
78 /*****
79 * 1 文字受信 : Sci1Read
80 *****/
81 char Sci1Read(void)
82 {
83     char c;
84     while(Sci1Rx1(&c)){
85         ;
86     }
87     return(c);
88 }
89
90 /*****
91 受信バッファクリア関数 :Rx1BuffClear
92 *****/
93 void Rx1BuffClear(void)
94 {
95     rx1_buff.start_index=0;
96     rx1_buff.data_count=0;
97 }
98
99 /*****
100 汎用関数
101 *****/

```

```
102
103 /*****
104 1バイト送信関数(改行(\n)はCR+LFに変換):Sci1Putchar
105 *****/
106
107 int Sci1Putchar(char c)
108 {
109     /* 改行(\n)はCR+LFに変換して送信 */
110     if(c=='\n'){
111         if(Sci1Write(CR)){
112             return(-1);
113         }
114         if(Sci1Write(LF)){
115             return(-1);
116         }
117     }else{ /* それ以外はそのまま送信 */
118         if(Sci1Write(c)){
119             return(-1);
120         }
121     }
122     return(0);
123 }
124
125 /*****
126 * 文字列送信(改行(\n)はCR+LFに変換):Sci1Puts
127 *****/
128 int Sci1Puts(char *str)
129 {
130     char *msg;
131
132     /* 終端文字まで1文字ずつ表示 */
133     for(msg=str; *msg!='\0'; msg++){
134         if(Sci1Putchar(*msg)){
135             return(0);
136         }
137     }
138     return(msg-str); /* 表示文字数を返す */
139 }
140
141 /*****
142 * 1文字受信(CRは改行(\n)に変換):Sci1Getchar
143 *****/
144 char Sci1Getchar(void)
145 {
146     char c;
147     c=Sci1Read();
148     if(c==CR){
149         c='\n';
150     }
151     return(c);
152 }
153
154 /*****
155 * 文字列受信(CRは改行(\n)に変換):Sci1Gets
156 *****/
157 void Sci1Gets(char *str)
158 {
159     int i=0;
160     char s;
161
162     while((s=Sci1Getchar()) != '\n'){
163         str[i]=s;
164         i++;
165     }
166     str[i]='\0';
167 }
```

---

*End Of List*

---



## B.5 文字列関連関数

■ファイル (str.c)、ヘッダファイル (str.h) ■

-----  
`int IsDigit(char c)`

形式:

```
#include "str.h"
```

```
int IsDigit(char c);
```

機能: 10 進数文字かチェックを行う関数。

引数: `char c` (10 進数文字かチェックを行う 1 文字)

返却値: 10 進数文字なら 1、それ以外なら 0 を返す。

-----  
`int IsHex(char c)`

形式:

```
#include "str.h"
```

```
int IsHex(char c);
```

機能: 16 進数文字かチェックを行う関数。

引数: `char c` (16 進数文字かチェックを行う 1 文字)

返却値: 16 進数文字なら 1、それ以外なら 0 を返す。

-----  
`int IsEos(char c)`

形式:

```
#include "str.h"
```

```
int IsEos(char c);
```

機能: 「\n」あるいは「\0」文字かのチェックを行う関数。

引数: `char c` (「\n」あるいは「\0」文字かチェックを行う 1 文字)

返却値: 「\n」あるいは「\0」文字なら 1、それ以外なら 0 を返す。

## ■ ファイル (str.c)、ヘッダファイル (str.h) ■

```
int StrDecToLong(char *str, long *n)
```

形式:

```
#include "str.h"
```

```
int StrDecToLong(char *str, long *n);
```

機能: 10 進数文字から数値に変換を行う関数。

引数: char \*str(10 進数文字列), long \*n(long 型の数値へのポインタ)

返却値: 「\n」あるいは「\0」文字まで達していない場合には 0、達している場合には 1 を返す。

```
int StrHexToLong(char *str, long *n)
```

形式:

```
#include "str.h"
```

```
int StrHexToLong(char *str, long *n);
```

機能: 16 進数文字から数値に変換を行う関数。

引数: char \*str(16 進数文字列), long \*n(long 型の数値へのポインタ)

返却値: 「\n」あるいは「\0」文字まで達していない場合には 0、達している場合には 1 を返す。

## ■ ファイル (str.c)、ヘッダファイル (str.h) ■

```
int LongToStrDec(long n, char *str)
```

形式:

```
#include "str.h"
```

```
int LongToStrDec(long n, char *str);
```

機能: 数値から 10 進数文字に変換を行う関数。

引数: long n(long 型の数値), char \*str(10 進数文字列)

返却値: 10 進数文字列の文字数。

```
int LongToStrHex(long n, char *str)
```

形式:

```
#include "str.h"
```

```
int LongToStrHex(long n, int upper, char *str);
```

機能: 数値から 16 進数文字に変換を行う関数。

引数: long n(long 型の数値), int upper(大文字小文字の指定), char \*str(16 進数文字列)

返却値: 16 進数文字列の文字数。

## str.h

```
1 #ifndef _STR_H_
2 #define _STR_H_
3
4 int IsDigit(char);
5 int IsHex(char);
6 int IsEos(signed char c);
7 int StrDecToLong(char *str, long *n);
8 int StrHexToLong(char *str, long *n);
```

```

9 int LongToStrDec(long n, char *buf);
10 int LongToStrHex(long n, int upper, char *buf);
11
12 #endif

```

---

*End Of List*

---



---

📄 **str.c**

---

```

1  #include "str.h"
2
3  /*****
4   10 進数文字かチェック
5   *****/
6  int IsDigit(char c)
7  {
8     if(c>='0' && c<='9'){
9         return(1);
10    }
11    return(0);
12 }
13
14 /*****
15  16 進数文字かチェック
16  *****/
17 int IsHex(char c)
18 {
19     if(c>='0' && c<='9'){
20         return(1);
21     }else if(c>='a' && c<='f'){
22         return(1);
23     }else if(c>='A' && c<='F'){
24         return(1);
25     }
26     return(0);
27 }
28
29 /*****
30  「\n」あるいは「\0」文字かのチェック
31  *****/
32 int IsEos(char c)
33 {
34     if(c=='\n' || c=='\0'){
35         return(1);
36     }
37     return(0);
38 }
39
40 /*****
41  10 進数文字から数値に変換
42  *****/
43 int StrDecToLong(char *str, long *n)
44 {
45     int minus=0;
46
47     /* マイナスかをチェック */
48     if(*str == '-'){
49         minus=1;
50         str++;
51     }
52
53     /* 10 進数文字を数値に変換 */
54     for(*n=0; IsDigit(*str); str++){
55         *n *= 10;
56         *n += (*str - '0');
57     }
58
59     /* マイナスの場合 */
60     if(minus){
61         *n = -1*(*n);
62     }
63
64     /* 終端文字まで達していない場合 */
65     if(!IsEos(*str)){
66         return(0);
67     }
68     return(1);
69 }
70
71 /*****
72  16 進数文字から数値に変換
73  *****/
74 int StrHexToLong(char *str, long *n)
75 {
76
77     /* 16 進数文字を数値に変換 */
78     for(*n=0; IsHex(*str); str++){
79         *n <<= 4;
80         if(*str >= '0' && *str <= '9'){

```

```
81     *n += (*str - '0');
82 }else if(*str >= 'a' && *str <= 'f'){
83     *n += (*str - 'a' + 10);
84 }else if(*str >= 'A' && *str <= 'F'){
85     *n += (*str - 'A' + 10);
86 }
87 }
88
89 /* 終端文字まで達していない場合 */
90 if(!IsEos(*str)){
91     return(0);
92 }
93 return(1);
94 }
95
96 /*****
97  数値を 10 進数文字変換
98  *****/
99 int LongToStrDec(long n, char *buf)
100 {
101     char c;
102     int len = 0;
103     long i, half;
104     int minus=0;
105
106     /* マイナスかをチェック */
107     if(n<0){
108         minus=1;
109         n=-n;
110     }
111
112     /* 10 進文字列へ変換し文字数をカウント */
113     do{
114         if(n == 0){
115             i = 0;
116         }else{
117             i = n % 10;
118         }
119         buf[len] = (char)(i + '0');
120         len++;
121         n /= 10;
122     }while(n != 0);
123     if(minus==1){
124         buf[len]='-';
125         len++;
126     }
127
128     /* 文字の並び順を直す */
129     half = len >> 1;
130     for(i=0; i < half; i++){
131         c = buf[i];
132         buf[i] = buf[(len-1)-i];
133         buf[(len-1)-i] = c;
134     }
135
136     /* 終端文字列の挿入 */
137     buf[len]='\0';
138     return(len);
139 }
140
141 /*****
142  数値を 16 進数文字変換
143  *****/
144 int LongToStrHex(long n, int upper, char *buf)
145 {
146     char c;
147     char a = 'a';
148     int len = 0;
149     long i, half;
150
151     /* 大文字/小文字の設定 */
152     if(upper){
153         a = 'A';
154     }
155
156     /* 16 進文字列へ変換し文字数をカウント */
157     do{
158         i = n & 0x0F;
159         if(i > 9){
160             buf[len] = (char)(i + a - 10);
161         }else{
162             buf[len] = (char)(i + '0');
163         }
164         len++;
165         n >>= 4;
166     }while(n != 0);
167
168     /* 文字の並び順を直す */
169     half = len >> 1;
170     for(i=0; i < half; i++){
171         c = buf[i];
172         buf[i] = buf[(len-1)-i];
173         buf[(len-1)-i] = c;
174     }
```

```

174     }
175
176     /* 終端文字列の挿入 */
177     buf[len]='\0';
178     return(len);
179 }

```

---

*End Of List*

---

## B.6 A/D 変換関連関数

### ■ ファイル (ad.c)、ヘッダファイル (ad.h) ■

```
void AdIinit(void)
```

形式:

```
#include "ad.h"
void AdIinit(void);
```

機能: A/D 変換の端子の初期化を行う関数。  
 シングルモード、ANO に限定  
 A/D 変換を用いる場合には、必ず実行すること。  
 デフォルトでは A/D 変換は停止状態。

引数: 無し  
 返却値: 無し

```
void AdStart(void)
```

形式:

```
#include "ad.h"
void AdStart(void);
```

機能: A/D 変換を開始する関数。

引数: 無し  
 返却値: 無し

```
void AdStop(void)
```

形式:

```
#include "ad.h"
void AdStop(void);
```

機能: A/D 変換を停止する関数。

引数: 無し  
 返却値: 無し

```
unsigned short AdRead(void)
```

形式:

```
#include "ad.h"
unsigned short AdRead(void);
```

機能: A/D 変換の結果を取得する関数。  
 ANO の値のみ。

引数: 無し  
 返却値: unsigned short (A/D 変換の結果 10 ビット)

### 📄 ad.h

```

1 #ifndef _AD_H_
2 #define _AD_H_
3
4 void AdIinit(int scan, int channel);
5 void AdStart(void);
6 void AdStop(void);
7 unsigned short AdRead(void);
8
9 #endif

```

---

*End Of List*

---

---

 ad.c

---

```
1  /*****  
2  /*  
3  /* FILE      :ad.c  
4  /* DESCRIPTION :A/D 変換用関数  
5  /* CPU TYPE   :h8/3694  
6  /*  
7  /* AD01 ANO  
8  /*  
9  /***/  
10  
11 #include "iodefine.h"  
12 #include "ad.h"  
13  
14 void AdInit(void)  
15 {  
16     AD.ADCSR.BIT.ADST = 0;      /* A/D 変換停止 */  
17     AD.ADCSR.BIT.SCAN = 0;     /* モード */  
18     AD.ADCSR.BIT.CKS  = 1;     /* 高速変換 */  
19     AD.ADCSR.BIT.CH   = 0;     /* チャンネル */  
20 }  
21  
22 /*****  
23     AD 変換開始  
24 *****/  
25  
26 void AdStart(void)  
27 {  
28     AD.ADCSR.BIT.ADST = 1;     /* A/D 変換スタート */  
29 }  
30  
31 /*****  
32     AD 変換中止  
33 *****/  
34  
35 void AdStop(void)  
36 {  
37     AD.ADCSR.BIT.ADST = 0;  
38 }  
39  
40 /*****  
41     AD 変換読み取り  
42 *****/  
43  
44 unsigned short AdRead(void)  
45 {  
46     unsigned short ad_r;  
47  
48     while(!AD.ADCSR.BIT.ADF){  
49         ;  
50     }  
51     ad_r=(AD.ADDRA>>6) ;  
52     AD.ADCSR.BIT.ADF=0;  
53     return(ad_r);  
54 }
```

---

End Of List

---

## B.7 歩行ロボット関連関数

■ファイル (pwm.c)、ヘッダファイル (pwm.h) ■

```
void PwmInit(void)
```

形式:

```
#include "pwm.h"
void PwmInit(void);
```

機能: サーボモータ用 PWM 出力の初期化を行う関数。  
デフォルトでは PWM は ON 時間が 1.5ms。

引数: 無し

返却値: 無し

```
void PwmStart(void)
```

形式:

```
#include "pwm.h"
void PwmStart(void);
```

機能: PWM 出力を開始する関数。

引数: 無し

返却値: 無し

```
void PwmStop(void)
```

形式:

```
#include "pwm.h"
void PwmStop(void);
```

機能: PWM 出力を停止する関数。

引数: 無し

返却値: 無し

```
void PwmSetDuty(int ch, unsigned short duty)
```

形式:

```
#include "pwm.h"
void PwmSetDuty(int ch, unsigned short duty);
```

機能: PWM 出力のデューティ比 (ON 時間) をセットする関数。

引数: int ch(モータ番号), unsigned short duty(ON 時間)

返却値: 無し

```
unsigned short PwmGetDuty(int ch)
```

形式:

```
#include "pwm.h"
unsigned short PwmGetDuty(int ch);
```

機能: PWM 出力のデューティ比 (ON 時間) を取得する関数。

引数: int ch(モータ番号)

返却値: unsigned short (PWM 出力のデューティ比 (ON 時間))

```
void PwmSetPose(unsigned char char_duty[])
```

形式:

```
#include "pwm.h"
void PwmSetPose(unsigned char char_duty[]);
```

機能: ポーズの補間 (モーションの再生) をする関数。

引数: unsigned char char\_duty[] (モーションデータ)

返却値: 無し

### pwm.h

```
1 #ifndef _PWM_H_
2 #define _PWM_H_
3
4 #define PWM_MAX_DUTY (12000-1) //2.4mS
5 #define PWM_MIN_DUTY (3000-1) //0.6mS
6 #define PWM_PERIOD (12500-1) //2.5mS
7
8 #define SRV_CH_NUM 8
9
10 #define D_CONST PWM_MIN_DUTY //char_duty と duty の変換式の定数部分
```

```

11 #define D_PROPO 45 //char_duty と duty の変換式の比例係数部分 (12000-3000)/200
12
13 #define SDV 239
14 #define SMD 241
15 #define DIV 242
16 #define SLO 243
17 #define ELO 244
18 #define SSD 249
19 #define SOH 254
20 #define EOT 255
21
22 void PwmInit(void);
23 void PwmStart(void);
24 void PwmStop(void);
25 void PwmSetDuty(int ch, unsigned short duty);
26 unsigned short PwmGetDuty(int ch);
27 void PwmSetPose(unsigned char char_duty[]);
28
29 #endif

```

---

*End Of List*

---



---

 pwm.c

---

```

1 #include "pwm1.h"
2 #include "iodefine.h"
3 #include <machine.h>
4
5 #define P1_MASK 0x07
6
7 char buf[6];
8 unsigned long n;
9 int c;
10 unsigned short srv_curr_index=0;
11 unsigned short srv_ch; /* PWM の出力先 */
12 unsigned short calc_duty_flg; /* 20ms をカウント */
13 unsigned long pwm_duty[SRV_CH_NUM]; /* PWM のデューティのデータ */
14
15 /*****
16 PWM 出力端子を初期化
17 *****/
18 void PwmInit(void)
19 {
20     srv_ch=0;
21     calc_duty_flg=0;
22     IO.PCR1|=0x07;
23     IO.PDR1.BYTE|=(srv_ch & P1_MASK);
24     IO.PDR1.BYTE&=(srv_ch | (~P1_MASK));
25     set_imask_ccr(1);
26     TW.TMRW.BYTE=0x49; /* FTIOB 端子を PWM 出力に設定 */
27     TW.TCRW.BYTE=0xA0; /* 内部クロックの 1/4, コンペアマッチ B で 1 出力 */
28     TW.TIERW.BYTE=0x71; /* A の割り込みを可能にする */
29     TW.TSRW.BYTE=0x70; /* 割り込みフラグをクリア */
30     TW.TCNT=0x0000; /* TCN の初期化 */
31     TW.GRA=PWM_PERIOD; /* 周期 (2.5mS) */
32     TW.GRB=PWM_PERIOD-7500; /* 1.5mS */
33     TW.TMRW.BIT.CTS=1; /* TCNT カウンタスタート */
34     set_imask_ccr(0);
35
36 }
37
38 /*****
39 サーボ動作開始
40 *****/
41 void PwmStart(void)
42 {
43     TW.TMRW.BIT.CTS=1;
44 }
45
46 /*****
47 サーボ動作停止
48 *****/
49 void PwmStop(void)
50 {
51     TW.TMRW.BIT.CTS=0;
52 }
53
54 /*****
55 デューティの設定
56 *****/
57 void PwmSetDuty(int ch, unsigned short duty)
58 {
59     /* サーボチャンネルが範囲外なら無視 */
60     if(ch >= SRV_CH_NUM){
61         return ;
62     }
63
64     if(duty < PWM_MIN_DUTY){ /* デューティを最小値以上に限定 */
65         duty = PWM_MIN_DUTY;

```



```

66     }else if(duty > PWM_MAX_DUTY){ /* デューティを最大値以下に限定 */
67         duty = PWM_MAX_DUTY;
68     }
69     /* デューティを設定 */
70     pwm_duty[ch] = duty;
71 }
72
73
74 /*****
75 デューティの取得 (パラメータから)
76 *****/
77 unsigned short PwmGetDuty(int ch)
78 {
79     /* サーボチャンネルが範囲外なら無視 */
80     if(ch >= SRV_CH_NUM){
81         return (0);
82     }
83     /* デューティを返す */
84     return (pwm_duty[ch]);
85 }
86
87
88 /*-----*/
89 /* ボーズ補間関数 0-200 で指定*/
90 /*-----*/
91 void PwmSetPose(unsigned char char_duty[])
92 {
93     long cur_duty[SRV_CH_NUM], new_duty[SRV_CH_NUM];
94     long diff_duty[SRV_CH_NUM];
95     int i;
96     unsigned short div, count_i;
97
98     div=char_duty[0]; /* 分割数 : 20mS 単位 */
99     for(i=0; i<SRV_CH_NUM; i++){
100         new_duty[i]=D_CONST+D_PROPO*char_duty[i+1]; /* 目標値 */
101         cur_duty[i]=PwmGetDuty(i); /* 現在のデューティ */
102         /* 現在のデューティと目標デューティの差 */
103         if(new_duty[i]-cur_duty[i]>=0){
104             diff_duty[i]=new_duty[i]-cur_duty[i];
105         }else{
106             diff_duty[i]=cur_duty[i]-new_duty[i];
107         }
108     }
109     /* 補間の計算 */
110     for(count_i=1; count_i<=div; count_i++){
111         /* 20mS 待ち */
112         while(!calc_duty_flg){
113             ;
114         }
115         calc_duty_flg=0;
116         /* 20mS ごとの値を計算 */
117         for(i=0; i<SRV_CH_NUM; i++){
118             if(new_duty[i]-cur_duty[i]>=0){
119                 PwmSetDuty(i, cur_duty[i]+(diff_duty[i]*count_i)/div);
120             }else{
121                 PwmSetDuty(i, cur_duty[i]-(diff_duty[i]*count_i)/div);
122             }
123         }
124     }
125 }
126
127 /*****
128 割り込み関数
129 *****/
130 __interrupt(vect=21) void INT_TimerW(void)
131 {
132     /* 2.5mS ごとのコンペアマッチ */
133     TW.TSRW.BIT.IMFA=0;
134     srv_ch++;
135     if(srv_ch>= SRV_CH_NUM){
136         srv_ch=0; /* モータ出力先を 0 に戻す */
137         calc_duty_flg=1;
138     }
139     IO.PDR1.BYTE|=(srv_ch & P1_MASK);
140     IO.PDR1.BYTE&=(srv_ch | (~P1_MASK));
141     TW.GRB=PWM_PERIOD-pwm_duty[srv_ch];
142 }

```

---

End Of List

---

## 歩行ロボットを用いたマイコン実習

より移植性の高いプログラムを目指して

2014年7月1日第3刷

著者 小野泰二

©YASUJI ONO

独立行政法人 高齢・障害・求職者雇用支援機構

東北職業能力開発大学校

電子情報技術科

〒987-2223 宮城県栗原市築館字萩沢土橋 26

TEL 0228-22-2082/FAX 0228-22-2432

本書の一部、あるいは全部を無断で複製することは、  
法律で認められた場合を除き、著作権の侵害となります