

sample8p【 ベータ版 2006/9/16 】

printfデバック1：扱えるデータ型、数値範囲
BCSH7206用

有限会社ビーリバーエレクトロニクス

【 用途 】 BCSH7206開発セット向け

【 プログラム概要 】

char、int、long、float のデータを、printf コマンドで SI0 モニタ (フォースビュー) に表示する。

【 概要 】

SI0モニタ画面にprintf関数で数値、変数、レジスタ、メモリ、ポートなどの内容をを表示するライブラリprintf_SH.hをリリースしました。

一般に、GCC添付のprintf関数は消費メモリが大きく、組み込み機器にそのまま使用できることは稀でした。そこで、コンパクトで、ある程度機能を満たしたprintf関数があれば組み込み機器のデバックもはかどると考えられ、弊社で開発しリリースしたものです。

サイズは7Kバイト程度で組み込んで負担にならないサイズだと思います。

【 printf_SH.o仕様 】

printf関数は以下のデータ形、フォーマット指定子等が使用できます。

扱えるデータ形	フォーマット指定子	サイズ (ビット)	扱える数値範囲 10進数(16進数)
char	%s 文字列	8	-128から127 (0x80~0x7f)
	%c 1文字		
short、int	%d 10進数	16	-32、768~32、767 (0x8000~0x7fff)
	%x 16進数	16	
long	%ld 10進数	32	-2147483648~ 2147483647 (0x80000000~ 7fffffff)
float	%f 浮動小数点1 0進数	32 1	±32766.9999程度 2

1 doubleもfloatと同じ指定子表示できますが、範囲はfloatと同じです。

2 浮動小数点の演算は仮数のビット長が有限であるため「情報落ち」誤差が生じる可能性があります。

また、桁落ち誤差も発生する場合があります。誤差を意識した使用が求められます。

使用できない表記

02%dのような出力桁指定表記、%gべき乗表示、他です。

【 printf 関数を使用するにあたり追加すること 】

sample8p 以外で printf 関数を使用する場合、以下の追加が必要です。

sample8p ディレクトリ内にある printf_SH.h、printf_SH.o ファイルを開発するソースファイルと同じディレクトリに置いてください。

開発するソースファイルに #include "printf_SH.h" と表記してください。

printf_SH.h 内で定義されている関数名は「print_sio」なので、printf として使用する場合、
#define printf print_sio

と、上記デファイン文で定義してください。後述ご参考。

xxx.cmd ファイルで以下の 2 箇所の赤の部分を追加してください。

```
REM Beyond the river 20060613
```

```
REM del *.o
```

```
del *.mot
```

```
sh-elf-gcc -m2 -c -Wall sh_start0.s -o sh_start0.o
```

```
sh-elf-gcc -m2 -O -g -c -Wall sample8p.c -o sample8p.o
```

```
sh-elf-gcc -m2 -nostartfiles -o sample8p.elf sh_start0.o sample8p.o printf_SH.o -Tsh7206_rom.x
```

```
sh-elf-objcopy -O srec sample8p.elf sample8p.mot
```

```
sh-elf-objdump -S sample8p.elf > sample8p.lst
```

```
sh-elf-objdump -h sample8p.elf > sample8p.map
```

以上で、好きなところに printf 関数を記入することができます。sample8p ディレクトリは上記全て実行済ですので、従来通り、sample8p+リターンでコンパイルできます。

【 プログラム内容 】

各データサイズの確認

d (10 進 16 ビット), x (16 進 16 ビット), ld (10 進 32 ビット), f (10 進 浮動小数点) 各表示方法、
違い

文字は全角、漢字対応できません。半角アルファベット、数字を使用してください。

【 プログラム 】

```
/*
```

```
sample8p SH7206 開発セット 【 printf デバック 1 】
```

題名：演算 char、int、long、float のデータを、printf コマンドで SIO モニタ（フォースビュー）

に表示する。

SIO 出力関数名 print_sio

各データサイズの確認

d (10 進 16 ビット), x (16 進 16 ビット), l (10 進 32 ビット), f (10 進 浮動小数点) 各表示方法の違い

文字は全角、漢字表示はできません。半角アルファベット、数字を使用します。(printf_H8 と異なります)

Copyright Beyond the river Inc. 20060916

*/

```
#include "SH7206.h"
#include "mov200MHz.h"
#include "sio.h"
#include "printf_SH.h"

#define printf print_sio

int main(void)
{
char key_data;
unsigned short sdata;
int idata;
long ldata;
float fdata;

//200MHz 動作

mov200MHz();
```

```

//SIO initial

init_SIO1();          //38400bps

printf("printf_test\nPlease push key 1 or 2 or 3");

while(1)
{
    key_data = char_in1();          //キーデータ読み込み

    switch(key_data)
    {
        case '1'://short と int データ

            //数値セット
            sdata = 1000;          //x 表示最大 0xFFFF
            idata = 0x7fff;        //d 表示最大 ±32767

            //SIO モニタ
            printf("\nx=%x d=%d",sdata,sdata);
                //sdata hex 表示 dec 表示
            printf("\nx=%x d=%d",idata,idata);
                //idata hex 表示 dec 表示

            break;

        case '2'://long と float データ

            //数値セット
            ldata = 1000200030+1111222233;
                //l 最大 ±2147483647
            fdata = 12345.1234+12345.1234;
                //f 最大 ±32766.9999 あたりまで扱える

            //SIO モニタ
            printf("\nl=%ld",ldata); //ldata dec 表示
            printf("\nf=%f",fdata); //fdata dec 表示
    }
}

```

```

        break;

    case '3': //char データ

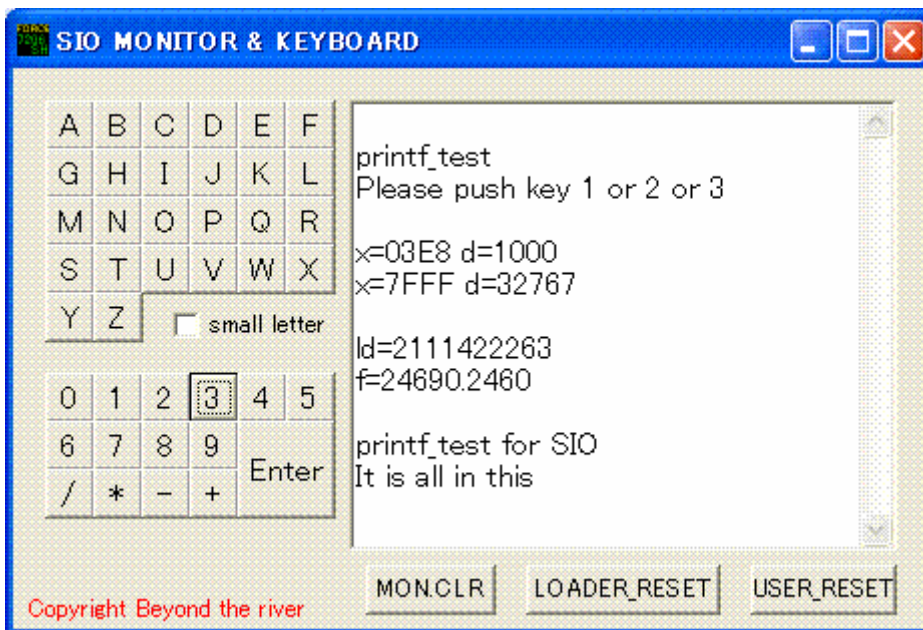
        //SIO モニタ
        printf("%nprintf_test for SIO");
        printf("%nIt is all in this");    //char 表示
        break;

    }

}
}
}

```

【 解説 】



```
#include "printf_SH.h"
```

printf_SH.h が関数のヘッダファイル、printf_SH.o がオブジェクトファイルです。必ずインクルードしてください。

```
#define printf print_sio
```

printf_h8.h 内で定義されている関数名は「print_sio」なので、printf として使用する場合、このデフォイン文で定義してください。もちろん、print_sio のままでも使えます。

```
printf("printf_test\nPlease push key 1 or 2 or 3");
```

「KEYBOARD」 1, 2, 3いずれかのクリックでそれぞれ種類の違う表示をします。

「1」のクリックで short データと int データ (このコンパイラでは同じ16ビット) のデータを16進数 (%x)、10進数 (%d) で表示します。unsigned データも表示できますが、その場合、%x 表示 (0 ~ FFFF) を使用してください。%d ですと 0 x 7FFF 以上はマイナス表示になってしまいます。

```
case '1': //short と int データ
```

```
//数値セット
sdata = 1000;           //x 表示最大 0xFFFF
idata = 0x7fff;        //d 表示最大 ±32767

//SIO モニタ
printf("%nx=%x d=%d", sdata, sdata);
//sdata hex 表示 dec 表示
printf("%nx=%x d=%d", idata, idata);
//idata hex 表示 dec 表示
break;
```

「2」のクリックで long データと float データ (どちらも32ビット) を表示します。

```
case '2': //long と float データ
```

```
//数値セット
ldata = 1000200030+1111222233;
//l 最大 ±2147483647
fdata = 12345.1234+12345.1234;
//f 最大 ±32766.9999 あたりまで扱える

//SIO モニタ
printf("%nl=%ld", ldata); //ldata dec 表示
printf("%nf=%f", fdata); //fdata hex 表示 dec 表示
break;
```

SIO モニタ画面で表示されているように long の整数 ldata = 1000200030+1111222233 = 2111422263 と誤差はありません。浮動小数点の fdata = 12345.1234+12345.1234 = 24690.2468 となつてほしいところですが、24690.2460 と表示されています。小数点以下4桁目の数値が違います。これが先に述べた誤差です。コンパイラによらず浮動小数点演算に付き物の誤差です。全体からみれば 0.00000...% の誤差です。用

途によってはどの程度の誤差が生じているのか、この printf 関数を用いて確認するといいいでしょう。

文字列の表示です。半角のアルファベット、数字が表示できます。全角、漢字は使用できません。

```
case '3'://char データ

//SIO モニタ
printf("%nprintf_test for SIO");
printf("%nIt is all in this"); //char 表示
break;
```

【 printf 関数使用上の注意点 】

無限ループの中で短時間でパソコン側に出力を繰り返すようなプログラムは対応できません。理由は常に 受信データ数 > 表示数 の関係では受信バッファが増加していった最後に設定数をオーバーフローしてしまうためです。

```
NG な例 1
while ( 1 )
{
    printf("time %d",time); // t i m e が連続してパソコン側に出力される
}
```

もし仮に上記のようなプログラムを書いて実行しますと、たいがい「SIO モニタ」は消えます。何回立ち上げても消えます。理由は受信バッファオーバーフローです。プログラムの中の問題のコマンドを削除するか、以下の対策を行い再コンパイルします。再書き込みで直ります。プログラムを書き換えない限り直りません。

対策例 1

表示後、「KEYBORAD」からの 1 文字入力待つ

```
while ( 1 )
{
    printf("time %d",time); // t i m e が連続してパソコン側に出力される
    cf=char_in1();
}
```

対策例 2

表示後、適当なウエイトを置く

```
while ( 1 )
{
    printf("time %d",time);    // t i m e が連続してパソコン側に出力される
    wait(1000);
}
```

対策例 3

表示後、無限ループにする。確認後、コメントとする。

```
while ( 1 )
{
    printf("time %d",time);    // t i m e が連続してパソコン側に出力される
    while(1){};
}
```

【 お問い合わせ 】

〒350-1213 埼玉県日高市高萩 1141-1 TEL 042 (985) 6982 FAX 042 (985) 6720

Homepage : <http://beriver.co.jp> e-mail : support@beriver.co.jp

有限会社ビーリバーエレクトロニクス