

sample9p【 ベータ版 2006/8/26 】

printfデバック2：レジスタ、メモリを見る

BCH83048ONE, BCH83052用

有限会社ビーリバーエレクトロニクス

【 用途 】 BCH83048ONE、BCH83052開発セット向け

【 プログラム内容 】

ポートを読み表示する

配列の全データを見る

配列の特定のデータを見る

特定のメモリのデータを読む

連続したメモリのデータを読む

【 プログラム 】

/*

sample36 学習ボード 【 printf デバック 2 】

題名：レジスタ、メモリの内容をSIOモニタ（フォースビュー）に表示する。

SIO出力関数名 print_sio

ポートを読み表示する

配列の全データを見る

配列の特定のデータを見る

特定のメモリのデータを読む

連続したメモリのデータを読む

このプログラムはフォースライタ Ver1.5(1.4)以上を使ってください。

SIOモニタ画面のバッファ不足（バグ）で長い文字列が表示できません。

Copyright Beyond the river Inc. 20060814

*/

```

#include "h8_300h.h"
#include "printf_H8.h"
#include "sio.h"

#define printf print_sio

#define DATA_RAM (*(volatile unsigned char *)0xfff000)
#define RENZ_RAM ((volatile char *)0xfff010)

int main(void)
{
    char a[]="0123456789ABC";
    char *p;
    //unsigned char c1,c2,c3;

    //SIO イニシャル

    sio_init();

    //1 行目表示
    //入力ポートを読み込み表示する

    P4DDR = 0x00; //P4DDR 0~7 を入力設定
    P4PCR = 0xff; //全てプルアップ

    //1つのメモリの読み書き

    DATA_RAM = 0x41;

    //連続したメモリの読み書き

    p = RENZ_RAM;
    *p = 0x41;p++;
    *p = 0x42;p++;
    *p = 0x43;p++;

```

```

*p = 0x44;p++;
*p = 0x45;p++;
*p = 0x00;
p -= 5;                                //アドレス前に戻す

//SIO モニタ

//1 行目表示
printf("¥nP4DR=%x",P4DR);              //ポートを読む

//2 行目表示
printf("¥nchar a[ ]=%s",&a);          //文字列を読む

//3 行目表示
printf("¥nchar a[5]=%c",&a[5]);       //文字列の中の1文字を読む

//4 行目表示
printf("¥nm1=%x m2=%s",DATA_RAM,p);   //メモリの内容を読む

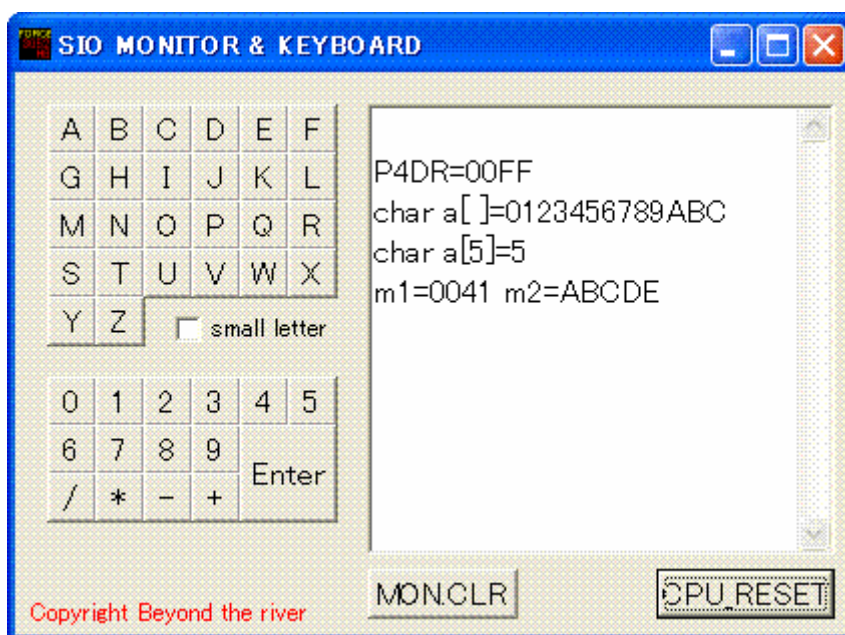
while(1)
{

}

}

```

【 解説 】



```
#define DATA_RAM (*(volatile unsigned char *)0xfff000)
```

0xfff000 という絶対アドレスの名前を定義しています。

```
#define RENZ_RAM ((volatile char *)0xfff010)
```

ポインタでアクセスするために、先頭アドレスを設定しています。

```
char a[]="0123456789ABC";
```

文字列の設定です。

```
char *p;
```

ポインタの名前を設定しています。

```
P4DDR = 0x00; //P4DDR 0~7を入力設定
```

```
P4PCR = 0xff; //全てプルアップ
```

P4 を読み込むので、入力に設定し、内部プルアップを付けています。

```
DATA_RAM = 0x41;
```

0xfff000 番地に 0x41 を書き込みました。

```
p = RENZ_RAM;
```

```
*p = 0x41;p++;
```

```
*p = 0x42;p++;
```

```
*p = 0x43;p++;
```

```
*p = 0x44;p++;
```

```
*p = 0x45;p++;
```

```
*p = 0x00;
```

0xfff010 番地から 0x41,0x42、、 0x45 まで書き込みます。

```

    p -= 5; //アドレス前に戻す
後の都合上、アドレスを先頭に戻しています。
    printf("%nP4DR=%x",P4DR); //ポートを読む
P4DR のデータが SI0 モニタに表示されます。同様の方法で H8 の全てのレジスタは読み込み、表示することができます。なお、表示は 16 ビット固定で、8 ビットレジスタでは上位 00 となり、意味を持ちません。
    printf("%nchar a[ ]=%s",&a); //文字列を読む
文字列を全て表示します。「0123456789ABC」
    printf("%nchar a[5]=%c",&a[5]); //文字列の中の 1 文字を読む
文字列の 5 番目のみ表示します。
//4 行目表示
    printf("%nm1=%x m2=%s",DATA_RAM,p); //メモリの内容を読む
m 1= DATA_RAM=0xffff000 番地のデータを表示します。
m 2 = p (ポインタ p) で示されるアドレスからの文字列を表示します。0 x 4 1、0 x 4 2、、、ですので、文字表示では ABCD、、(アスキーコード)となります。

```

以上、sample8p、sample9p と 2 つのサンプルプログラムで printf デバック、printf 関数の使い方について説明いたしました。特別なハードウェアを必要としないシンプルなデバック環境が構築できると思います。

【 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