建構簡單 Arduino 自動控制系統(二 A) (輸出部份 RS232，移植 C printf 函數)

Author:spuggy0919@gmail.com

學習等級：已會操作 Arduino 開發介面寫程式，基本 C 語言。

Arduino 有一個 Serial RS232 模組，這次將使用這個模組來顯示。
在此教學我們主要在說明如何在 Serial 以 C 的 printf 輸出。

第一步 Serial RS232 是透過 USB 將 RS232 資料回傳在序列監識器(Serial Monitor)中顯示。這個不需要硬體連接，單一 Arduino 板子便可以操作。

第二步 在網頁 http://www.menie.org/georges/embedded/printf-stdarg.html
尋找 printf 的原始碼, 並將其剪貼入專案內，並做三點小修改紅色部份。
1・修改 putchar 爲 Serial.write 函數。
2・加入`#define TEST_PRINTF`，打開測式的部份並將 main 修改爲 test 函數。
3・加入 Setup 及 Loop。
//--------- 以下副程式 printf ---------

```
/*
    Copyright 2001, 2002 Georges Menie (www.menie.org)
    stdarg version contributed by Christian Ettinger

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU Lesser General Public License as published
by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
*/

/*
    putchar is the only external dependency for this file,
    if you have a working putchar, leave it commented out.
    If not, uncomment the define below and
    replace outbyte(c) by your own function call.

#define putchar(c) Serial.write(c) // Modified Here only for Serial
*/


static void printchar(char **str, int c)
{
```

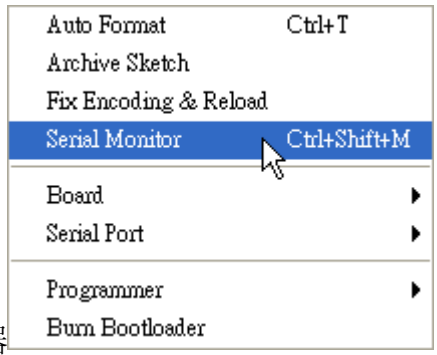
```
// extern int putchar(int c);

   if (str) {
      **str = c;
      ++(*str);
   }
   else  (void)putchar(c);
}

#define PAD_RIGHT 1
#define PAD_ZERO 2

static int prints(char **out, const char *string, int width, int pad)
{
   register int pc = 0, padchar = ' ';

   if (width > 0) {
      register int len = 0;
      register const char *ptr;
      for (ptr = string; *ptr; ++ptr) ++len;
      if (len >= width) width = 0;
      else width -= len;
      if (pad & PAD_ZERO) padchar = '0';
   }
   if (!(pad & PAD_RIGHT)) {
      for ( ; width > 0; --width) {
         printchar (out, padchar);
         ++pc;
      }
   }
   for ( ; *string ; ++string) {
      printchar (out, *string);
      ++pc;
   }
   for ( ; width > 0; --width) {
      printchar (out, padchar);
      ++pc;
   }

   return pc;
}

/* the following should be enough for 32 bit int */
#define PRINT_BUF_LEN 12

static int printi(char **out, int i, int b, int sg, int width, int pad, int
letbase)
{
   char print_buf[PRINT_BUF_LEN];
   register char *s;
   register int t, neg = 0, pc = 0;
   register unsigned int u = i;

   if (i == 0) {
      print_buf[0] = '0';
      print_buf[1] = '\0';
      return prints (out, print_buf, width, pad);
   }

   if (sg && b == 10 && i < 0) {
```

```
        neg = 1;
        u = -i;
    }

    s = print_buf + PRINT_BUF_LEN-1;
    *s = '\0';

    while (u) {
        t = u % b;
        if( t >= 10 )
            t += letbase - '0' - 10;
        *--s = t + '0';
        u /= b;
    }

    if (neg) {
        if( width && (pad & PAD_ZERO) ) {
            printchar (out, '-');
            ++pc;
            --width;
        }
        else {
            *--s = '-';
        }
    }

    return pc + prints (out, s, width, pad);
}

static int print(char **out, const char *format, va_list args )
{
    register int width, pad;
    register int pc = 0;
    char scr[2];

    for (; *format != 0; ++format) {
        if (*format == '%') {
            ++format;
            width = pad = 0;
            if (*format == '\0') break;
            if (*format == '%') goto out;
            if (*format == '-') {
                ++format;
                pad = PAD_RIGHT;
            }
            while (*format == '0') {
                ++format;
                pad |= PAD_ZERO;
            }
            for ( ; *format >= '0' && *format <= '9'; ++format) {
                width *= 10;
                width += *format - '0';
            }
            if( *format == 's' ) {
                register char *s = (char *)va_arg( args, int );
                pc += prints (out, s?s:"(null)", width, pad);
                continue;
            }
            if( *format == 'd' ) {
                pc += printi (out, va_arg( args, int ), 10, 1, width, pad, 'a');
```

```
                continue;
            }
            if( *format == 'x' ) {
                pc += printi (out, va_arg( args, int ), 16, 0, width, pad, 'a');
                continue;
            }
            if( *format == 'X' ) {
                pc += printi (out, va_arg( args, int ), 16, 0, width, pad, 'A');
                continue;
            }
            if( *format == 'u' ) {
                pc += printi (out, va_arg( args, int ), 10, 0, width, pad, 'a');
                continue;
            }
            if( *format == 'c' ) {
                /* char are converted to int then pushed on the stack */
                scr[0] = (char)va_arg( args, int );
                scr[1] = '\0';
                pc += prints (out, scr, width, pad);
                continue;
            }
        }
        else {
        out:
            printchar (out, *format);
            ++pc;
        }
    }
    if (out) **out = '\0';
    va_end( args );
    return pc;
}

int printf(const char *format, ...)
{
        va_list args;

        va_start( args, format );
        return print( 0, format, args );
}

int sprintf(char *out, const char *format, ...)
{
        va_list args;

        va_start( args, format );
        return print( &out, format, args );
}
#define TEST_PRINTF
#ifdef TEST_PRINTF
int test(void)
{
    char *ptr = "Hello world!";
    char *np = 0;
    int i = 5;
    unsigned int bs = sizeof(int)*8;
    int mi;
    char buf[80];

    mi = (1 << (bs-1)) + 1;
```

```
    printf("%s\n", ptr);
    printf("printf test\n");
    printf("%s is null pointer\n", np);
    printf("%d = 5\n", i);
    printf("%d = - max int\n", mi);
    printf("char %c = 'a'\n", 'a');
    printf("hex %x = ff\n", 0xff);
    printf("hex %02x = 00\n", 0);
    printf("signed %d = unsigned %u = hex %x\n", -3, -3, -3);
    printf("%d %s(s)%", 0, "message");
    printf("\n");
    printf("%d %s(s) with %%\n", 0, "message");
    sprintf(buf, "justif: \"%-10s\"\n", "left"); printf("%s", buf);
    sprintf(buf, "justif: \"%10s\"\n", "right"); printf("%s", buf);
    sprintf(buf, " 3: %04d zero padded\n", 3); printf("%s", buf);
    sprintf(buf, " 3: %-4d left justif.\n", 3); printf("%s", buf);
    sprintf(buf, " 3: %4d right justif.\n", 3); printf("%s", buf);
    sprintf(buf, "-3: %04d zero padded\n", -3); printf("%s", buf);
    sprintf(buf, "-3: %-4d left justif.\n", -3); printf("%s", buf);
    sprintf(buf, "-3: %4d right justif.\n", -3); printf("%s", buf);

    return 0;
}
/*
 * if you compile this file with
 *   gcc -Wall $(YOUR_C_OPTIONS) -DTEST_PRINTF -c printf.c
 * you will get a normal warning:
 *   printf.c:214: warning: spurious trailing `%' in format
 * this line is testing an invalid % at the end of the format string.
 *
 * this should display (on 32bit int machine) :
 *
 * Hello world!
 * printf test
 * (null) is null pointer
 * 5 = 5
 * -2147483647 = - max int
 * char a = 'a'
 * hex ff = ff
 * hex 00 = 00
 * signed -3 = unsigned 4294967293 = hex fffffffd
 * 0 message(s)
 * 0 message(s) with %
 * justif: "left     "
 * justif: "    right"
 *  3: 0003 zero padded
 *  3: 3   left justif.
 *  3:   3 right justif.
 * -3: -003 zero padded
 * -3: -3   left justif.
 * -3:   -3 right justif.
 */
#endif
void setup()
{
      Serial.begin(9600);
      Serial.println("printf: testing ");
      test();

}
```

```
void loop()
{
}
```



第三步打勾編譯, 上載再打開序列通訊監視器，這樣便可以看見 printf 測試輸出的部份了，也可以開始使用 printf 函數，以後不管在偵錯或輸出都可以用熟習的 c 函數。