第4章 デバッグ用文字出力関数を作る

起動処理のプログラムでひとまずhello,worldを表示

豊山 祐一

リスト1　デバッグ用UART出力プログラム（kernel¥syslib.c）

```
/* UART0の初期化 */
void tm_com_init(void)
{
    /* ボーレート設定 */
    out_w(UART0_BASE+UARTx_IBRD, 67);
    out_w(UART0_BASE+UARTx_FBRD, 52);
    /* データ形式設定 */
    out_w(UART0_BASE+UARTx_LCR_H, 0x70);
    out_w(UART0_BASE+UARTx_CR, UART_CR_RXE|
          UART_CR_TXE|UART_CR_EN); /* 通信イネーブル */
}

/* デバッグ用UART出力 */
UINT tm_putstring(char* str)
{
    UINT    cnt = 0;

    while(*str) {
        /* 送信FIFOの空き待ち */
        while((in_w(UART0_BASE+UARTx_FR) &
                        UART_FR_TXFF)!= 0);
        /* データ送信 */
        out_w(UART0_BASE+UARTx_DR, *str++);
        cnt++;
    }
    return cnt;
}
```

前章までででマイコンの起動処理のプログラムを作成し，C言語のmain関数が実行可能となりました．Try KernelのOS本体を作り始める前に，デバッグの準備として，文字列の表示機能を実装します．まずはmain関数からhello, worldを表示します．

シリアル経由で文字列を表示するプログラムを作る

● ディスプレイなし環境の定番デバッグ手法

Picoにディスプレイはありませんが，シリアル通信で送信したデータはPicoprobeを経由してPCに送ることができます．PicoprobeはPCでUSB-シリアル変換デバイスとして認識されるので，Tera Term[注1]などのターミナル・エミュレータを実行しておけば，Picoから送られてきた文字列のデータを表示できます．

このようなシリアル通信による文字列の表示機能はプログラムのデバッグの際に便利なので，ディスプレイのない組み込みシステムではよく使われます．

● デバッグ用の文字表示関数を実装する

IEEE 2050-2018規格にはデバッグ機能の仕様は含まれませんが，規格のベースとなったμT-Kernelには，デバッグ用のAPIとしてtm_putstring関数が定められています．そこでTry Kernelでもこの関数を実装します．

tm_putstring関数はC言語の標準関数であるputstringとほぼ同じ機能ですが，標準出力ではなく決められたデバッグ用のシリアル通信出力に文字列データを送ります．

tm_putstring関数のソースコードを**リスト1**に示します．この関数は今後もデバッグに使用するので，kernelディレクトリのsyslib.cに置くこととします．

リスト中にはtm_putstring関数以外にtm_com_init関数が記述されています．これはUARTの初期化を行う関数なので，tm_putstring関数を使用する前に一度だけ実行する必要があります．デバッグ用なので通信速度や形式は固定です．

main関数を実行してhello，worldを表示

準備ができたのでmain関数を作成して実行してみましょう．前章で作成したLED点滅プログラムのmain関数の冒頭にtm_putstring関数によりhello，worldを表示させるコードを追加したのが**リスト2**です．

PCのターミナル・エミュレータにはTera Termを使用します．Tera Termを実行したらメニューから［ファイル］-［新しい接続］を選択し，Picoの接続されているシリアル・ポートを選択します．**図1**に設定の

注1：Tera Termのウェブ・ページ．
https://ja.osdn.net/projects/ttssh2/