実例2：UART

鈴木 量三朗

UARTが使えると デバッグなどで重宝する

● マイコンでUARTが動くとうれしい

何らかのCPUを持つ典型的な組み込みシステム開発では，開発用ボードが届くと，次のようなステップで動作確認を進めることが多いのではないでしょうか．

1. 設計通りに電源周りのLEDが点灯するか確認
2. テスト用のLEDの点灯や点滅（つまりLチカ）の動作を確認
3. ソフトウェア的なライブラリなどを動かす
4. UARTでホストと接続してHello Worldの表示

ここまでできると，バンザイ！初期デバッグ終了といった流れになると思います．

マイコン・ボードはHello Worldと，しゃべることで初めてこの世に接するわけで，そういう意味で組み込みシステムでのHello Worldはソフトウェア的なスタート地点と言えます．

● FPGAでもUARTで内部状態の観測ができる

FPGAを搭載したボードではLチカがスタート地点に相当します．ソフトウェア的な準備が大変になるため，わざわざHello Worldを表示することは，あまり重要視されないでしょう．

それでもUARTは対人という意味で便利なユーザ・インターフェースを提供してくれます．LEDの点灯やオシロスコープによる波形観測と比べると，ボード自身がHello Worldをはじめとした内部状態をしゃべってくれることで，開発は楽になるに違いありません．

Pythonを使った高位合成ツールPolyphonyがUARTを制御することで，ソフトウェア的な開発手法をFPGAのハードウェア開発に提供できれば，便利なことが幾つもありそうです．

ここでは新たにuart_masterというプロジェクトを作ります．FPGAでUARTを制御し，Hello Worldに挑戦してみます．

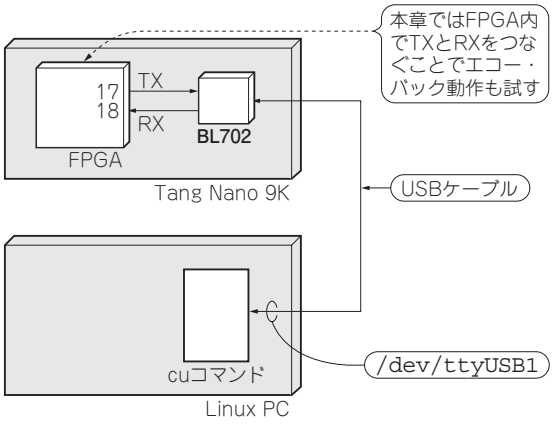

図1　Tang Nano 9KとLinuxとがUARTで通信する場合

Tang Nano 9Kで UART通信するための実験環境

図1に実験環境におけるTang Nano 9K（Sipeed）とLinux PCとの関係を示します．

Tang Nano 9KにはBL702（Bouffalo Lab）というUSB-シリアル変換に使うチップが搭載されています．今回の構成では，このチップからの信号がUSB Type-C経由でLinux PCにつながっています．Linuxからは/dev/ttyUSB1などのシリアル・ポートとして見えています．従ってLinuxから，cuコマンドやscreenコマンドでシリアル・ポートとして/dev/ttyUSB1を指定することで，Tang Nano 9KとUART通信ができます．

FPGAチップ側を見るとFPGAとBL702とはTX（17番ピン）およびRX（18番ピン）でつながっています．FPGA内部でTXとRXをつないでしまえば，Linuxから書き込んだデータがループ・バックします（**リスト1**）．これでLinuxのcuなどのアプリケーションから打ち込んだASCIIコードがUART通信を通してエコー・バックするのを確認できます．このための物理制約ファイルを**リスト2**に示します．

リスト1　FPGA内でTXとRXの信号をつないでエコー・バックさせるVerilogコード

```
module top (
  input wire clk,

  output wire uart_tx,
  input wire uart_rx
);

assign uart_tx = uart_rx;

endmodule
```