

# UARTとA-Dコンバータを使ったプログラミング

組み込み機器において多用されるインターフェースがUART(Universal Asynchronous Receiver Transmitter)だ。UARTさえあれば、マイコン基板とパソコンの間でデータの受け渡しが行える。ときにはデバッグ出力としても活躍するなど、幅広く応用できる。また、今回のFRマイコンはA-Dコンバータを内蔵している。特定のピンに入力された0V~3.3Vの電圧をA-D変換し、デジタル値として読み取ることができる。ここでは、付属マイコン基板とパソコンを接続して文字データを入力する方法と、A-Dコンバータからの入力をUARTへ出力する方法について解説する。(編集部)

平石 郁雄

本誌の連載記事「組み込みマイコンの仕組みを理解しよう」の第3回(2008年4月号, pp.175-183)では、UART(RS-232-C)の基本とFRマイコンでUARTを使うためのレジスタについて解説しました。本稿では、付属FRマイコン基板とパソコンを使って、実際にUARTを使ってみます。

まず、環境を整えます。使用する部品は、USBコネクタが1個、ショート・プラグが1個、そしてリセット・スイッチが1個です(写真1)。また、パソコンには、ターミナル・ソフトウェアが必要です。ここでは、フリー・ソフトウェアの「TeraTerm」(<http://hp.vector.co.jp/authors/VA002416/>)がインストールしてあることを前提に説明します。

ここで解説するプログラムはUART用のサンプルですが、付属マイコン基板とパソコンの間はUSBケーブルで接続します。これは、基板上のCP2102でUSB-UART変換を行っているためです。なお、CPUやUARTの詳細については、上述の連載記事を参照してください。

マイコンからの信号をターミナル・ソフトで表示  
連載の第3回では割り込みを使用しないプログラム例を掲載しました。今回は割り込みを使用したUARTプログラムを紹介します。割り込みの詳細については第6章で説明します。プログラムの流れを図1に、UARTプログラムのAPI(Application Program Interface)を表1に示します。このプログラムは、パソコンからの信号をマイコンで受信し、マイコンからの出力をパソコンのターミナル・ソフトウェア(TeraTerm)で表示するようになっています。

## プログラムの構成

一人で開発するような小規模な組み込みソフトウェアであれば、好きなように書くことができます。しかし、複数のエンジニアが共同で開発するような大規模な組み込みソフトウェアの場合、一般的に「アプリケーション」と「ドライバ」に分けて開発する傾向があります。

ドライバは、ハードウェアを直接制御するソフトウェア部品で、アプリケーションはドライバが提供するAPIを使ってハードウェアにアクセスします。このAPIは非常に重要です。例えば筆者の場合、複数のグループで組み込みプログラムを開発する機会があるのですが、このAPI仕様

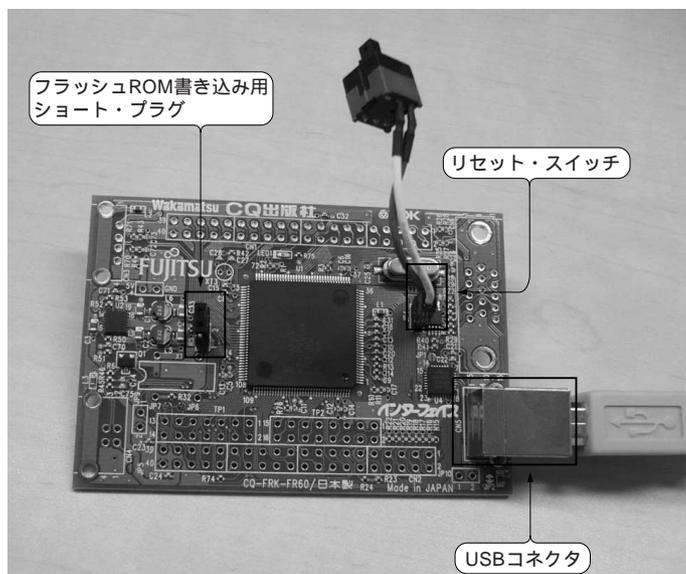


写真1 UARTプログラムで使用する部品

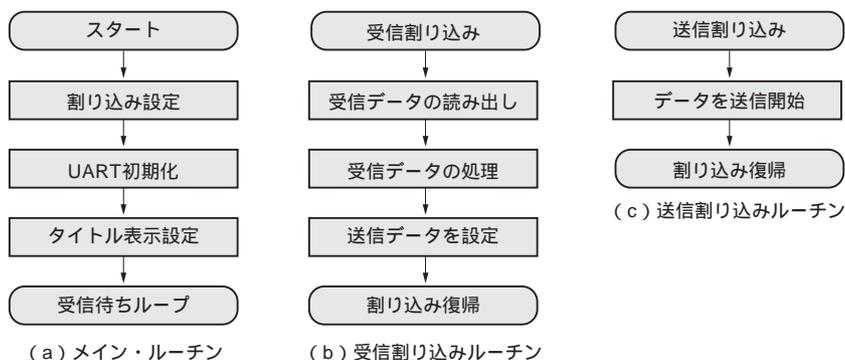


図1 プログラムの流れ

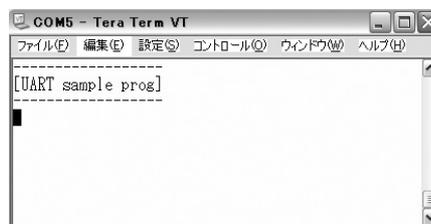


図2 TeraTerm の表示  
リセットを押すと[ UART sample prog ]が表示される。

表1  
UART プログラムの  
API

	名称	型	データ	内容
引き数	なし	なし	なし	なし
戻り値	なし	なし	なし	なし

(a) 初期化関数 `uart_init()`

本関数を呼び出すことで `uart0` の初期化を行う。UART0 を使う前には、本関数をユーザ・メインで実行すること。パリティあり、偶数パリティ、ストップ・ビット1、転送データ・サイズ8ビット、ボーレート19200bps

	名称	型	データ	内容
引き数	<code>size</code>	<code>char</code>	1 ~ 256	送信するデータのサイズを1 ~ 256 の範囲で設定
	<code>data</code>	<code>char A[ 256 ]</code>	任意	本配列データを若い番号から順番に、UART0 から出力する
戻り値	<code>Uart_flag</code>	<code>char</code>	0	正常に送信データを設定完了
			1	送信中のデータがあるため、送信データを設定できていない

(b) 表示設定関数 `uart_display(char size, char data)`

本関数の `data`( 引き数 )に `size`( 引き数 )に設定した個数の表示データを設定することで、UART からデータが送信される。設定したデータが空になるまで繰り返す。

	名称	型	データ	内容
引き数	<code>rec_data</code>	<code>char</code>	受信データ	UART で受信したデータ
	<code>rec_err</code>	<code>char</code>	エラー・データ	UART のステータス・レジスタの値
戻り値	なし	なし	なし	なし

(c) 受信データ処理関数 `uart_rec(char rec_data, char rec_err)`

本関数をメイン・プログラムに作成する。本関数の引き数に受信データとシリアル・ステータス・レジスタが格納されているので、本関数で処理すること。

を作成してから、それぞれのグループで開発を行っています。API 仕様に従っていないプログラムができてきたときには、アプリケーションとドライバを結合するためのプログラムの作成や、面倒な保守が必要になります。

ただ、今回は API の善しあしはとりあえず忘れて、メイン・プログラム部 `user_main.c` (リスト1) と UART ドライバ部 `fr_uart2.c` (リスト2) の構成に分けて作成しています。

`user_main.c` 中の関数(リスト1)

●メイン関数 `main()`

まず、`_EI()` で、PS レジスタの CCR フィールドにある

I フラグを1に設定します。次に、`_set_il(31)` で PS レジスタの ILM フィールドに割り込み最低レベルの31を設定し、0 ~ 30の割り込みレベルが受け付けられるようにします。

そして、`uart_init()` を呼び出して、UART の設定を初期化します。`title()` 関数を呼び出して、TeraTerm にタイトルを表示します(図2)。その後、UART の受信待ち状態になります。

●タイトル表示関数 `title()`

`uart_display` 関数への引き数として、タイトル画面に表示するデータを設定しています。