第4章　パソコンと通信して
マイコンの動作を制御しよう！

試しながら学ぶシリアル・インターフェース操作術

島田 義人
Yoshihito Shimada

本章では，R8C/Tinyマイコンのシリアル・インターフェース（UART：Universal Asynchronous Receiver Transmitter）機能について解説します．UART機能を使って，自由にパソコンとデータ通信を行ってみます．

UART0機能を使いこなすことができるようになると，パソコンからの指示でマイコンを動かせますし，マイコンの動きをパソコンに送り出すことができるようになります．本章を参考にしてぜひとも，UART0機能を使ったさまざまなアプリケーション・プログラムの作成にチャレンジしてみてください．第5章以降では，UART0機能をマイコンのデータ入出力機能として使っていきます．

シリアル・インターフェースの基礎

シリアル通信には，クロック同期式と調歩同期式（非同期式とも呼ばれる）の2種類の通信方式があります．R8C/Tinyマイコンのシリアル・インターフェース機能（UART）は，この二つのシリアル・データ通信方式に対応しています．

■ 通信方式

● クロック同期式

相互間でクロック信号による同期をとり，そのタイミングに従ってデータを送受信します．送受信データの有無に関わらず制御用の信号が流れているため，相手との同期を常に保つことができます．

データがないときには，待ち状態を示す信号をやりとりしなければならない欠点があります．しかし，データを送受信する際に，データの始まりと終了を示す信号が存在しないので，データ転送速度はそのぶん速くなります．

● 調歩同期式

Windows対応のパソコンでは，一般に調歩同期式が使われています．この方式では，データをビット（bit）という単位に分け，1文字ぶんのデータごとに同期をとることによって，送受信間の正常なデータのやりとりを行っています．

データの先頭と最後には，必ずデータの開始を示すスタート・ビットと，データの終了を示すストップ・ビットが付きます．したがってクロック同期式に比べ，調歩同期式の速度は遅くなる欠点はありますが，待ち状態のときに余分な情報を処理する必要がありません．

また，調歩同期式におけるアイドル状態は，マークと呼ばれる‘1’の値をもちます．このマークにより，アイドル状態の場合とケーブルが外れている状態を判別することができます．

そのほか，調歩同期方式ではマルチプロセッサ通信機能を備えており，複数のプロセッサ間で通信回線を共有してデータの送受信を行うことができます．

■ 調歩同期式モードのデータ・フォーマット

ここでは，R8C/Tinyマイコンを使って，Windows対応のパソコンと調歩同期式によるシリアル通信をしてみましょう．

調歩同期式モードの一般的なデータ・フォーマット構造を**図1**に示します．通信データの1キャラクタ，もしくは1フレームは，スタート・ビット（Lレベル）から始まり，送信/受信データ（LSBからMSBの順），パリティ・ビット，ストップ・ビット（Hレベル）の順で構成されています．

Keywords
シリアル・インターフェース，UART，クロック同期式，調歩同期式，スタート・ビット，ストップ・ビット，パリティ，ビット・レート・ジェネレータ，フレーミング・エラー，MB－R8CQ，R8C/Tiny

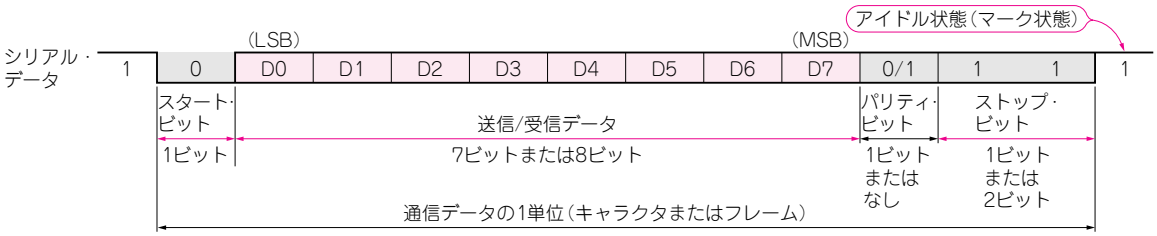

図1　調歩同期式モードの一般的なデータ・フォーマット

● スタート・ビットとストップ・ビット

シリアル通信では，送受信するデータが1列になって送られるため，そのままでは文字データの区切りがわかりません．そこで，文字データの前と後ろに識別のためのビットが付加されています．

この前後に付加されるビットのうち，1文字ぶんのデータ・ビット列の前に付けて，データの先頭を示すビットのことを「スタート・ビット」と呼びます．そして1文字ぶんのデータ・ビット列の後ろに付けて，データの終了を示すビットのことを「ストップ・ビット」と呼びます．すなわち，データの最初と終わりを示すマークのことです．

スタート・ビットは必ず‘0’で，1ビットぶんの幅が割り当てられています．一方のストップ・ビットは，‘1’と決められていますが，ビットの幅は1ビットまたは2ビットがあります．通常は1ビットを使用することが多いようです．

● パリティ・ビット

パリティ・ビット（parity bit）とは，データの送受信を行う際に，データが正しく伝達されたものであるかどうかをチェックするためのビットです．パリティ・ビットは，データ・ビット列の後ろ，ストップ・ビットの前に設けられた，1ビットの幅をもっています．パリティには偶数パリティ（even parity），奇数パリティ（odd parity），そしてパリティなし（non parity）という選択肢があります．

ビットの状態は，送信時に7または8ビットの単位に分けられたデータ・ビットの中で，‘1’になっているビットを数え，その結果によって‘1’か‘0’かの状態をとります．そして受信の際には，その値を参照して送られてきたデータに誤りがないかどうかを検出します．

奇数パリティでは，‘1’の数が必ず奇数個になるようにパリティ・ビットを‘1’か‘0’に調整します．偶数パリティでは，同様に‘1’の数が必ず偶数個になるようにパリティ・ビットを調整します．

具体的に例を示してみましょう．例えば，8ビット・データ1001 1011では‘1’の数が奇数あります．このとき，偶数パリティの場合はパリティ・ビットは‘1’になります．一方，奇数パリティの場合では‘0’になります．

実際にエラーが発生したときに，エラーの存在を知らせることはできますが，それがどのデータ・ビットなのか，その場所を知らせる機能はありません．また，偶数個のエラーがデータ中に発生した場合には，パリティ・ビットでエラーを検出することが不可能であることは明らかです．したがって，通常ではパリティなしで使うことが多いようです．

UART機能関連のレジスタ

R8C/TinyマイコンにはUART機能として1チャネルのUART0が内蔵されています．UART0はクロック同期式と調歩同期式（非同期式）のいずれにも使えるようになっています．

R8C/TinyマイコンのUART0のブロック構成を**図2**に示します．UART0は独立した送信部と受信部を備えているので，送信と受信を同時に行うことができます．また，送信部および受信部ともにダブル・バッファ構造になっているため，連続送信/連続受信が可能です．UART0には，以下に述べる各種のレジスタがあります．

● UART0送受信モード・レジスタ（U0MR）

U0MRは，シリアル・データ通信フォーマットとシリアルI/Oモードを選択するためのレジスタです．U0MRのビット構成を**図3**に示します．

▶パリティ制御ビット（PRYE）

PRYE = ‘1’のとき，送信時はパリティ・ビットを付加し，受信時はパリティ・チェックを行います．

通常はパリティなしでPRYE = ‘0’に設定します．

▶パリティ選択ビット（PRY）

PRY = ‘0’のとき，パリティ・モードを奇数パリティに設定し，PRY = ‘1’のとき偶数パリティに設定します．このビットは，パリティ制御ビットPRYE = ‘1’のときに有効になります．

▶ストップ・ビット長選択ビット（STPS）

STPS = ‘0’で送信時のストップ・ビット長を1ビットに設定し，STPS = ‘1’で2ビットに設定します．