

JinSatoの地下室工房

じん さとう
Jin Sato

— 第17回(最終回) —

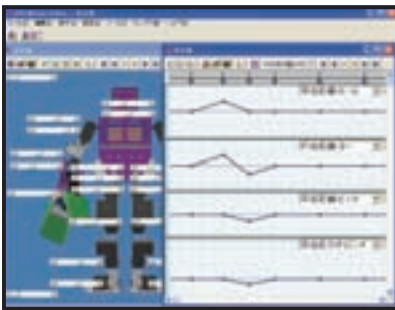
■ はじめに ■

前回、「STK500」という ATMEL 社製の基板を使い、シリアル通信の基本部分を説明しました。今回はその通信の基本部分を使い、データの表現方法やユーザーインターフェースの仕組みについて考えてみたいと思います。

■ モーションエディタ ■

さっそくですが、**画面1**は、数年前に作成したモーションエディタです。このモーションエディタは1つのサーボに1つのスクロールバーがそれぞれ対応しており、スクロールバーを操作することで、サーボモーターを動かすことができるようになっています。

このようなことを行うには、単にシリアル通信ができればよいというだけではなく、具体的にどのようなデータのやり取りを行うかを考える必要があります。



画面1 筆者が作成したモーションエディタ

■ シリアル通信のさまざまな方法 ■

一般に、通信を行う際には、データの送信側と受信側との間でさまざまな取り決めが必要です。ここでは、シリアル通信時のデータの表現方法として、テキストを使う方式とバイナリを使う方式、加えて、イン

ターフェースの作り方として、メニュー方式とコマンド方式について、それぞれを組み合わせた実例をいくつか紹介します。

■ テキストを使ったメニュー方式 ■

テキストとはキーボードから入力できる文字、たとえばここでは、半角の数字とアルファベットの組み合わせとしておきましょう。それらの文字を使って表現できる命令(コマンド)を決めて、ロボットと通信を行う方式です。

テキストコマンド方式は、キーボードから入力できる文字だけを使うので、パソコン側に特別なソフトを用意しなくても、キーボードからコマンドを直接パチパチと入力して操作できるのが魅力です。

上手にメニューを作れば、キーボードで1文字入力しながら対話的に作業を進めることができますし、実装も容易です。初めてモーションエディタを作るときはテキストメニュー方式がお手軽でしょう。

画面2は、EEPROMの内容を操作するためのメニューを表示しています。

このメニューの中では、たとえば「d」と入力し、そのあと、表示したいアドレスを6桁の16進数で入力することにより、特定のメモリアドレスを表示する機能などを提供しています。

画面2のプログラムの一部は、**リスト1**のようになっています。



画面2 EEPROM操作用のメニュー画面

■ テキストを使ったコマンド方式 ■

GUIのインターフェースを使わずに、メニューだけでモーションエディタのようなものを作ることは可能ですが、メニューが増えてゆくとマイコンのメモリが不足することも考えられます。

そこで、メニューに依存しない方法として、数文字のアルファベットを組み合わせたコマンド名を決めて、コマンド入力により操作するようにすれば、メモリ効率もアップします。

たとえば**画面2**のように、EEPROMのメモリ状態を確認する場合ならば、dumpと覚えやすいコマンド名を決めて、それを入力するとメモリ内容を表示するようにすれば、メニューを作る手間が省けますね。

ただし、コマンド名があまり長いとコマンド入力時のタイプミスも考えられますし、そのコマンドに対応したサブルーチン呼び出すプログラミングも大変になります。コマンド名は2文字程度と決めてしまうと、プログラムを作るのも容易になると思います。

たとえば、タクシー会社に電話をして、タクシーを家に配車してもらった場合、タクシー会社の配車係は、客先にもっとも近いタクシーを無線連絡で割り当てるとしています。プログラム中でこういう役割を担うものをディスパッチャー(dispatcher)といいます。また、送られてきたコマンドから適切なサブルーチンなどを呼び出すルーチンのことを、ディスパッチルーチンと呼びます。

リスト2は、送られてきた2文字のコマンドを解釈してサブルーチン呼び出すプログラムの一部です。

このルーチンは、リニアサーチと呼ばれる手法で、リストに登録してある2文字を順番に比較していますので、リストの最後にあるコマンドの検索には時間がかかりま