



マイコンを正しく操縦するための作法

基礎から学ぶC言語講座

岡田 好一

Yoshikazu Okada

最終回 データベースを使うデータ共有の実験

最終回の話題はデータベースとの接続です。複数のマイコンをネットワークで接続する際、お互いに網の目のようにデータ交換するのも一手ですし、そうしたシステムもあるようです。しかし、データを集中管理するとシステムの動作の指定や追跡が容易になります。必要なのはマイコンの相手をする、いわゆるオンライン・システムです。データベースの使用は冗長なのですが、データベース仕立てにしておく、システムの解析や拡張が容易などの利点が期待できます。

R8C/1Bでは無理ですが、上位のマイコンともなるとUNIX級のOSが楽々と動作するようです。ですから、データベース・システムを装置の内部に抱えるのも夢の話ではなくなってきました。今回はWindows PCのデータベース・システム(Microsoft Access)と接続のためのVisual Basicプログラムを利用して、2台のR8C/1Bスタータキットからデータを共有する方法を探ってみます。

開発目標は複数のマイコンの協調動作

初期の計画では、ごく簡単に接続しようと思ったのですが、やってみるとWindows PC側が結構大仕掛けになってしまいました。さすがに現代ではシステムが確実に便利になったぶん、準備作業が膨大になって

います。

そこで最初に開発目標を確認しておきましょう。

冒頭で述べたように、複数のマイコンを協調動作させるには種々の方法があると思います。本稿で想定しているのは、データベース内に簡単なシステムのモデル—といっても数個の整数データ—を用意し、個々のマイコンが適宜問い合わせ動作を決めて行く方法です。通常は動作の目的があって、データベースの内容が上位のコンピュータによって刻々変化して行きます。末端のマイコンは、データベースの内容、つまり指令を確認し、動作してから結果を報告、つまりデータベースにデータを登録します。

中央から積極的に末端に向かって指令しないのは奇異に思えるかもしれませんが、これは考え方ですが、個々の末端の制御方式や処理のベースは末端に任せてしまい、結果の報告でシステムの現状を評価すると想定します。

データベースとは何か

データベースは字義通りにはデータの基地であり、複数の応用プログラムから参照可能なデータ集とそれを支えるプログラム群を指します。データの読み書きだけでなく、検索や集計ができるのが特徴です。

Keyword 1

データベース

あるプログラム専用のファイルを用意すれば高速で便利なのは当然ですが、それでは応用が利きません。

例えば、社員の所属などの基本情報は多数のプログラムから参照されるので、集中管理できなければ面倒なことになります。

データベース(database)は、複数の応用プログラムから利用されることを前提とした計算機上のデータ集です。データの読み書きだけでなく、検索と集計ができます。

応用上、利用者管理、障害管理、ネットワーク管理などが必要とされるので、データベース・マネージメント・シ

ステム(DBMS)はOSに匹敵する複雑なプログラムです。とはいえ、その基礎は1970年代には完成していて、私には改良が続いているだけのように入ります。

データ構造として見ると、キーと呼ばれる値からデータを得る検索表です。

データベースの基礎技術はB-treeと呼ばれるハードディスク上のデータ構造であり、物理構造と論理構造に著しい乖離がある、特徴的なシステムです。

サーバ上のデータベースは、複数同時参照に対応していますし、利用者管理や障害管理まで行いますから、数あるシステムの中でも屈指の複雑さを有するソフトウェア(データベース・マネジメント・システム: DBMS)です。通信も必須なので設定などが多く、接続するまではおおげさです。しかし、データ構造として見ると、キーと呼ばれる数値や文字列などで値を得る検索表であり、原理的な部分は難しくありません。

● リレーショナル・データベース

現在広く使われているリレーショナル・データベース(RDB)では、データの本体は単純な表形式になっています。

RDBのモデルは「集合」であり、1行1件のデータが他のデータと識別できればよく、元来、順序はありません。しかし、実用上、順序はたいせつで、検索速度の鍵を握っています。ですから、内部的にはキー項目で順に並べます。

● 問い合わせ言語 SQL

検索や更新にはSQLと呼ばれる「問い合わせ言語」を使います。ただし、データベースの定義や設定はDBMSを直接使い、案内に従って行るのが通常です。ですから、アプリケーション側からは「SELECT(検索)」「INSERT(挿入)」「UPDATE(更新)」「DELETE(削除)」の四つの文を知っていれば、とりあえずデータベースを使うことができます。

今回はADO.NETと呼ばれる関数群を使います。出てくるのは単純なSQL文のみです。

Microsoft AccessとMicrosoft Visual Basic(VB)を使う

● Accessは個人用のデータベース・ソフトウェア

今回採用したMicrosoft Access 2003はMicrosoft Officeの1ソフトウェアで、個人用のデータベースであり、資料が多く、手軽に使えますし、表計算ソフト

ウェアとの連結も容易です。

個人用なので、Accessは複数同時参照が得意ではありません。そこで、Visual Basicで接続部分を作成します。二つのR8C/1BスタートキットとRS-232Cで通信し、窓口を1本化(逐次化)してからAccessに問い合わせます。Accessから見ると、相手はVisual Basicのみです。なお、使用したVisual Basicはバージョン2005 Express Edition(無料版)です。無料版なので、契約の範囲内で使用します。

本来ならば、端末はDBMSが管理すべきですし、通信は標準プロトコルを遵守しないと危険です。今回は接続自体を主眼とするので、素朴に作成してみました。実際のシステムを作る際には、管理や通信品質についての十分な検討と対策が必要です。

● VBを使ったRS-232Cプログラミング

通信にはスタートキットのRS-232Cラインを使い[要改造。本連載第11回(2007年4月号)を参照]、市販のUSBへの変換器を介してWindows PCに接続します(図12-1)。今回の設定では変換器の導入時に「COM4」と「COM5」の名称がWindowsにより割り振られたので、そのまま利用します。

VB側では「SerialPort」と呼ばれる組み込みのコンポーネントを利用します。コンポーネントはマイクロソフト用語で、特定機能のサブルーチン集と考えればよいでしょう。

ツールボックスにSerialPortがあります。今回はCOM4とCOM5の2回線ですから、フォーム上に二つのSerialPortを配置します(不可視)。デフォルトの名称はそれぞれ「SerialPort1」と「SerialPort2」となります。

単純な通信なら文字列を読み書きするだけで、簡単に接続できてしまいます。ただし、SerialPortのプログラミングにはVBのマルチスレッドの知識が必要です。記述量は少ないものの、説明は簡単ではありません。詳しくは参考文献(1)を参照してください。

Keyword 2

リレーショナル・データベース

リレーショナル・データベース(relational database: RDB)は、現在もっとも人気のあるデータベースの論理構造です。リレーション(関係)とは表のことであり、1行が1件のデータです。初期には用語が混乱していましたが、現在は分かりやすい名称と呼ばれています。関係代数と関係論理が分かれば、RDBを理論的に理解したことになります。

RDBのモデルは集合で、各行は見かけ上も他と識別できることが条件であり、データの順序は理論上は求められていません。

しかし、例えば図書館の本は類似した内容の書籍が近くにあるなど、並べる順序はたいせつです。検索一特に前方一致を高速度化するためにはインデックスと呼ばれる全順序構造を導入する必要があります。

もっとも、図書館の例で分かるように、現実にはデータの順序はきわめて主観的であり、整数などの自明な例を除いて決定版はなく、応用の性質によって最適な条件が変わります。

リレーショナル以外のデータベースのモデルとしては、階層型、ネットワーク型、オブジェクト型があります。