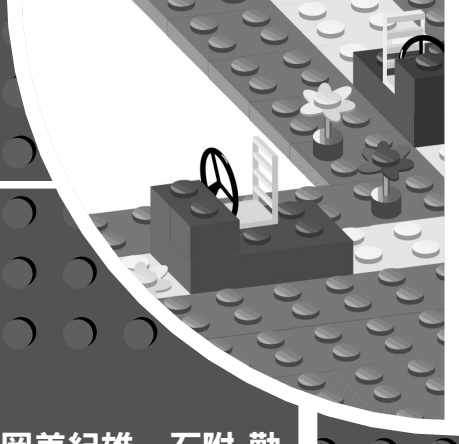

4 CPUコアの活用
—— Ethernetモジュールの設計

●長岡美紀雄，石附 勤

本稿ではPLD向けソフト・マクロのCPUを活用して設計したEthernetモジュールの開発事例を紹介する．CPUコアのリファレンス・デザインに準拠したシステム構成をとることでハードウェアおよびソフトウェアの開発期間の短縮を図った．100Base-TX対応モジュールでは，IPコアの独自設計も行った．（編集部）

筆者らがEthernetモジュールを開発したのは，社内にある工作機械や測定器をLANに接続し，ネットワーク経由で制御/モニタしたいという要望からでした．工作機械や計測器の入出力には，RS-232-Cインターフェースが使われているのが普通です．多チャネルRS-232-Cインターフェース・ボードを装着したパソコンに接続し，専用ソフトウェアで制御したり測定値を記録していました．機器によってはシリアル・インターフェースではなく，接点出力やパラレル出力などというものもありました．

これらの機器をLANに接続するには，一般的なEthernetアダプタでは対応できません．そこで独自に開発を始めました．

システム構成の検討

Ethernetモジュールを開発するにあたり，はじめに要求事項をまとめ，実現性を吟味しながらシステム構成を決定しました（**表1**）．

〔表1〕開発するモジュールへの要求事項

特　徴	説　明
小型化	機器内に組み込むことがあるので，なるべく小型化したい．
柔軟性	一つの設計（基板）でいろいろなインターフェースに対応する．RS-232-C，パラレルなど．
保守性	制御機器の耐用年数（15年以上）の保守が必要． 部品の入手性に注意．
低コスト化	少量生産で低コスト化を図る．開発ツールなどにかかるライセンス料などを抑える必要がある．

●要求事項

1）小型基板サイズ

機器内に組み込むことを考え，なるべく小型化することを目標としました．

2）対象機器ごとにカスタマイズ可能な柔軟性

工作機械や測定器の外部インターフェースは，RS-232-C，パラレルなどさまざまです．インターフェースごとに基板を設計していたのでは，コストがかかってしまいます．一つの設計（基板）でいろいろなインターフェースに対応できるようにくふうします．

3）長期間にわたる部品の入手性

汎用CPUの場合，運が悪いと数年で製造中止（ディスコン）になってしまいます．そうなるとせっかく開発したモジュールも製造中止にせざるをえません．保守もできなくなってしまいます．製造中止になったCPUの後継製品が発売されたとしても，換装しただけで動作するかどうかはわかりません．

制御機器などは，長いものだと15年以上使われることがあります．使用する部品の入手性は大きな問題です．

4）少量生産で低コスト化を図る

LANに接続するということは，すなわちTCP/IPで通信するということです．

ハードウェア的には，CPUとLANコントローラLSIがあれば（トランスやRJ45コネクタなども必要だが），Ethernetインターフェースを構成することができます．しかしハードウェアだけでは動作しません．ソフトウェアも必要です．

〔表2〕
実現方式の検討

ハードウェア	OS	TCP/IP	開発期間	コスト	説　明
H8/300 CPU + NE2000互換 LAN	購入	購入	短い	高い	もっとも慣れたハードウェア構成. OSやプロトコル・スタックのライセンス費用が大きい.
	開発	開発	長い	低い	OSとプロトコル・スタックの開発には時間がかかる. 多用途で活用できるので開発コストは分散可能.
組み込み用ボード	Linux	OS付属	中～長	低い	ボードとOSは比較的低コストですむ. リアルタイム性を保証するにはハードルが高い.
x86ボード	Windows	OS付属	短い	高い	ボードのコストが大きい.

TCP/IPで通信するソフトウェアには，プロトコル・スタックと呼ばれるソフトウェア（ミドルウェア）を利用します．しかし市販のプロトコル・スタックは高価であり，機器1台ごとにロイヤリティを支払わなければならない場合が多いようです．もちろん，適用するハードウェアに合わせてカスタマイズする必要があり，ノウハウと多くの時間が要求されます．

少量生産で低コスト化を図るためには，開発にかかるライセンス料などを抑える必要があります．

●実現方式の検討

筆者の開発環境において，要求仕様を満たすための構成を考え，実現可能性の検討を行いました（表2）．

1）H8/300 CPU + NE2000互換 LAN + 市販TCP/IPスタック

この構成は，筆者らが過去に開発した機器で何度か採用した方法です．筆者らにとっていちばん手慣れた構成で，ハードウェアについては低コストで製作できることがわかっています．

問題はリアルタイムOSとTCP/IPプロトコル・スタックです．過去の設計は受託だったので，相手先企業からの要求で市販のリアルタイムOSと開発環境を使うことができました．しかし，今回のような少量生産では，ライセンス料が問題になってしまいます．

リアルタイムOSとTCP/IPプロトコル・スタックを独自に開発することも検討しています．しかし実現にはまだ時間がかかりそうです．

2）市販の小型Linux搭載ボード + 自作I/O

LASER5の「L-Card+」やセンチュリーシステムズの「FutureNet MA-300」のようなLinux対応の組み込み用CPUボードを活用する方法です．

Linux環境における開発経験はありました．Linuxはオープン・ソースなので，ソース・コードが公開されています．しかし，それらをすべて吟味して製品としての品質を保証するには，けっこうな時間と労力が必要と思われます．

また，今回ターゲットとしている制御機器では，リアルタイム性が要求されます．Linuxにおいてリアルタイム性を保証するには，通常のカーネルにリアルタイムLinuxを移植しなければなりません．ハードルは高くなります．

3）小型x86ボード + ROM化Windows

x86系CPU搭載のPC/AT互換アーキテクチャのボードを使う方法では，いろいろな選択肢があります．なか

4

column　ディスコン問題を解決するソフト・マクロのCPU

パソコン用のCPUであれば5年も経てば時代遅れになってしまい，本体ごと買い換えるというのが普通です．しかし，計測・制御機器は，10年，15年と使われ続けることが珍しくありません．これらの機器の製造メーカは，当然その間，保守を行わなければなりません．ここで問題になるのが，修理用保守部品の確保です．特に最近は，一つのCPUファミリの中で新しく機能強化された製品が次々に発売され，古い品種は比較的早い時期に製造中止（ディスコン）になってしまいます．

CPUの場合，代替品や後継品と称されるものでも完全互換ではなかったり，プログラムに修正が必要になったりと，ハードウェアとソフトウェアの両面で対応しなければならないケースが少なくありません．しかしHDLで記述されたCPUであれば，器となるデバイスが異なっても内部ではまったく同じ機能を構成することが可能です．プログラムの修正の必要もありません．

CPUコアは市販品も多く，インターネット上ではフリーのCPUコア・プロジェクトがいくつも立ち上がっています．将来，40ピンDIPのZ80が完全に世の中から姿を消すことがあったとしても，PLDで構成したZ80でこれを補うことができるかもしれません（パッケージの物理形状変換などは必要だが）．