[第1章]

パソコン，マイクロコントローラ，周辺技術の概要，変遷

パソコンによる制御システム

一昔前（1980年代）ならば，パソコンもいわゆるマイコンとしてシステム・バスの構造を理解して，そこにI/O（Input/Output；入出力装置，ボード）を接続し，外部のコントロールを行うことで，簡単な制御システムを構築することが比較的容易にできていました．

ところが，パソコンの性能が向上するにつれ，ハード的にもソフト的にもそのような「工作」が困難になってきました．その主な要因として，ハード面ではPCI（Peripheral Components Interconnect）バスやUSB（Universal Serial Bus），ソフト面ではWindows NT系の普及です．これらで構成される最近のパソコンでは，一般的なコンピュータやプログラミングの知識やツールを使うだけでは，新たな外部機器の増設は困難になりました．

しかしここにきて，入手しやすくて低価格なPIC（Microchip Technology社）やUSB変換チップの普及によって，USBなどの既存のI/O窓口を利用した簡単な外部機器の実現が可能になってきました．本章では，このように変化してきた現状のパソコンに対して，どのようなI/O増設の方法があるかを広範囲に紹介します．

1.1 制御システムの構築

前述のような理由で，現状のマイコンを利用した制御システムは，**図1.1**と**図1.2**のように分けられます．**図1.1**はパソコンを中心に各種拡張（I/O）ボードを利用して外部機器の制御を行う方法です．この場合，拡張ボードは通常PCIの規格に準じたものを使用し，自作するのは難しいので市販品を購入することになります．また，拡張ボードをドライブするソフトウェアは，メーカが用意した拡張ボードに付属するドライバやツール，あるいは，LabVIEWなどの統合型の計測ツールを利用します．

図1.2はいわゆる「組み込みシステム」と呼ばれるもので，CPUとその周辺を一つのブロックとして制御システムを構築します．従来，組み込みシステムはOSをもたないのが一般的でしたが，組み込みに用いるCPUと周辺回路が高性能になってきたため，リアルタイムOSを搭載する場合もあります．

図1.1のようなパソコンを中心とした方法の場合，高機能なパソコンの機能，つまり高機能な計測ソフトや，大容量のデータ・ストレージ（ハードディスクやDVD-R，フラッシュ・メモリ）が利用できます．しかし，これらを実現するにはそれなりのコストがかかり，たとえば，比較的低速でかつ数ビットあれば間に合うような小規模なシステム構築には大げさになります．

一方，**図1.2**のような「組み込みシステム」の場合，小規模なシステムからコスト・パフォーマンスの良いシステムが構築できますが，複雑な処理を行わせようとすると，しっかりとした開発ツールが必要にな

このアイコンは，章末に用語解説があります

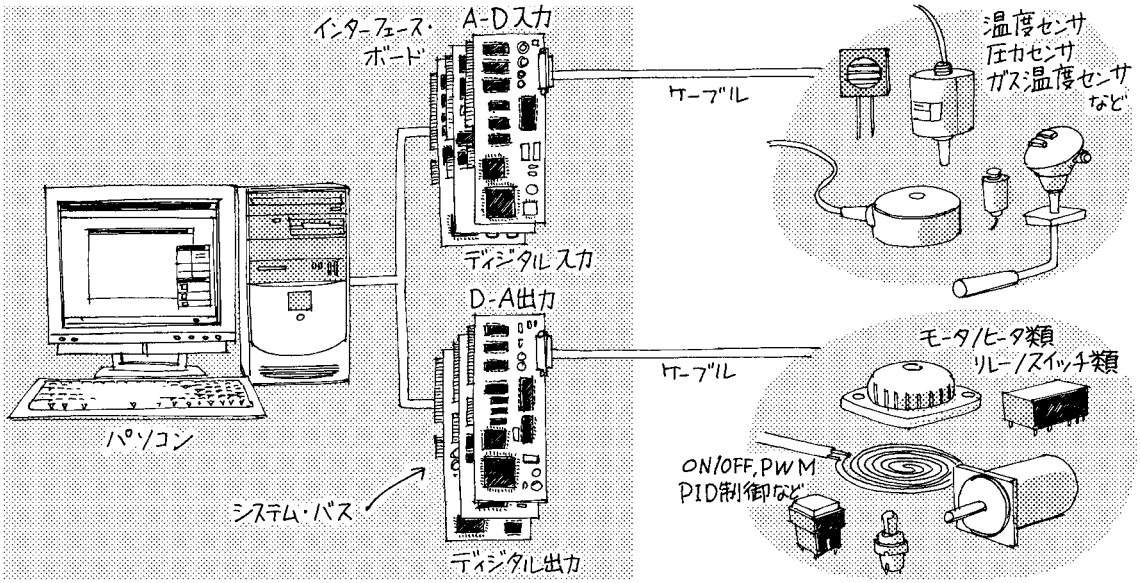


図1.1　パソコンを中心とするシステム

ります．また，データ・ストレージやネットワーク（TCP/IP；インターネット・プロトコル，無線LANなど）の構築となると，その実装は結構な手間となります．

そこで，これらのデメリットを補完する3番目の手法として，パソコンを中心にしてマイクロコントローラを組み合わせた**図1.3**のような構成が考えられます．従来からも3番目の手法はインテリジェントI/Oといった呼び方で存在していましたが，周辺処理を行うマイクロコントローラおよびその通信手段の開発が今ほど簡便でなく，あまり採用されなかったようです．本書では，使い勝手のよくなったPICに代表されるマイクロコントローラとパソコンを組み合わせた，3番目の手法による簡便な外部制御システムの構築方法について解説します．

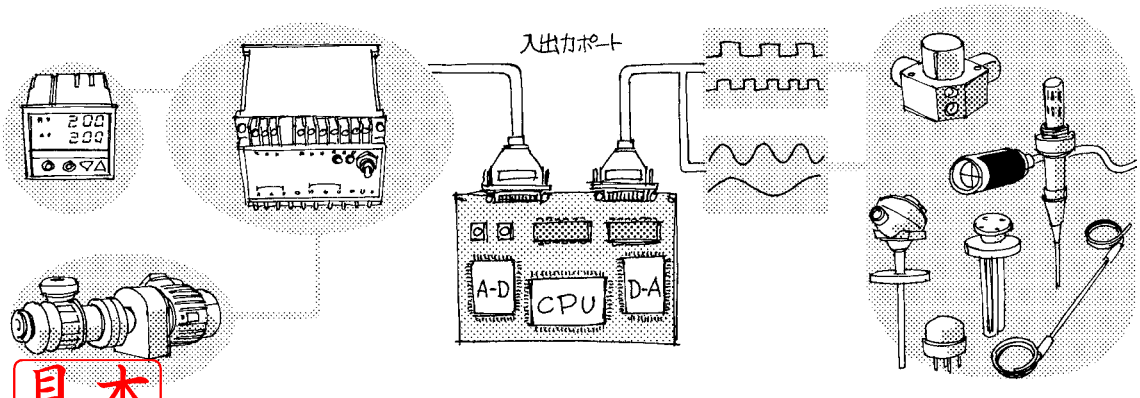


図1.2　組み込みシステム

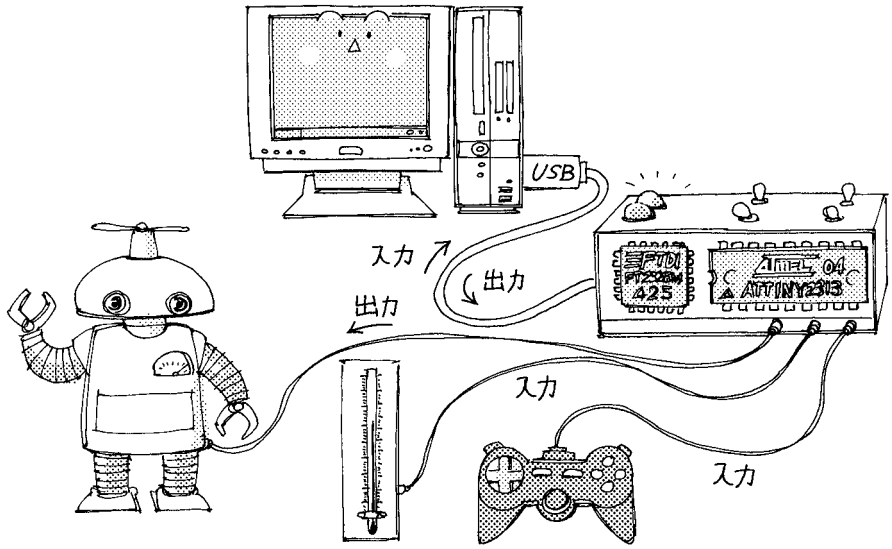


図1.3　パソコンとマイクロコントローラを組み合わせたシステム

1.2　マイクロコンピュータの基本構成

この節では，マイクロコンピュータの基本構造，基本構成をおさらいしながら，パソコンと組み込みシステムの違いとそれぞれに用いられるCPUの違いを紹介します．

マイクロコンピュータは，パソコンから，最小システムに近い組み込みマイコンに至るまで，基本的には，**図1.4**に示すように，CPUを中心に，ROMやRAMのメモリおよびインターフェースなどのLSIチップで構成されます．それぞれのチップは，データ・バス，アドレス・バスおよびコントロール・バスと呼ばれる線群で結ばれています．

このバスでは，CPUから見るとメモリも含め周辺からのデータの読み出しと書き込みの二つの動作が

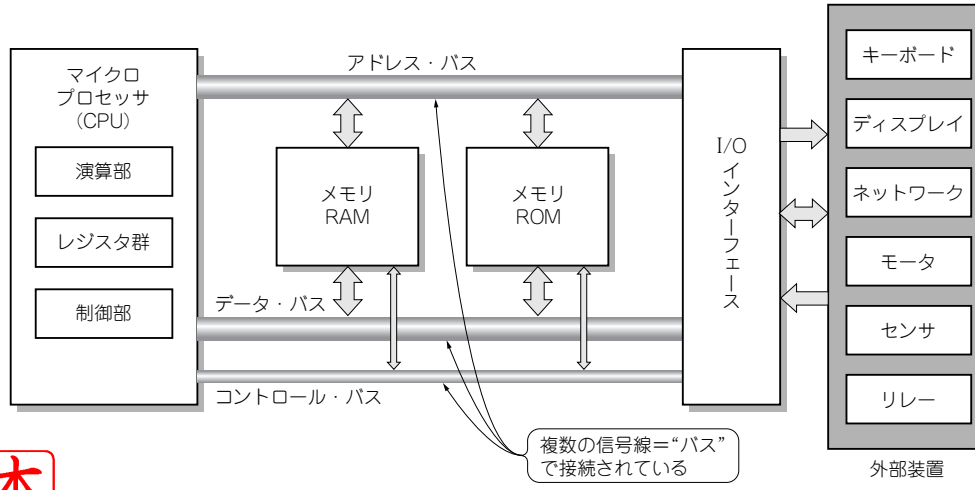


図1.4　マイクロコンピュータの基本構成

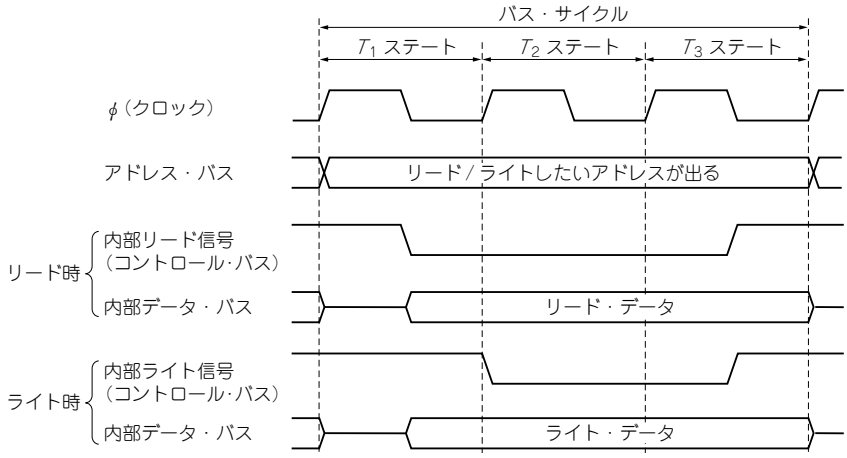


図1.5　バス・タイミング

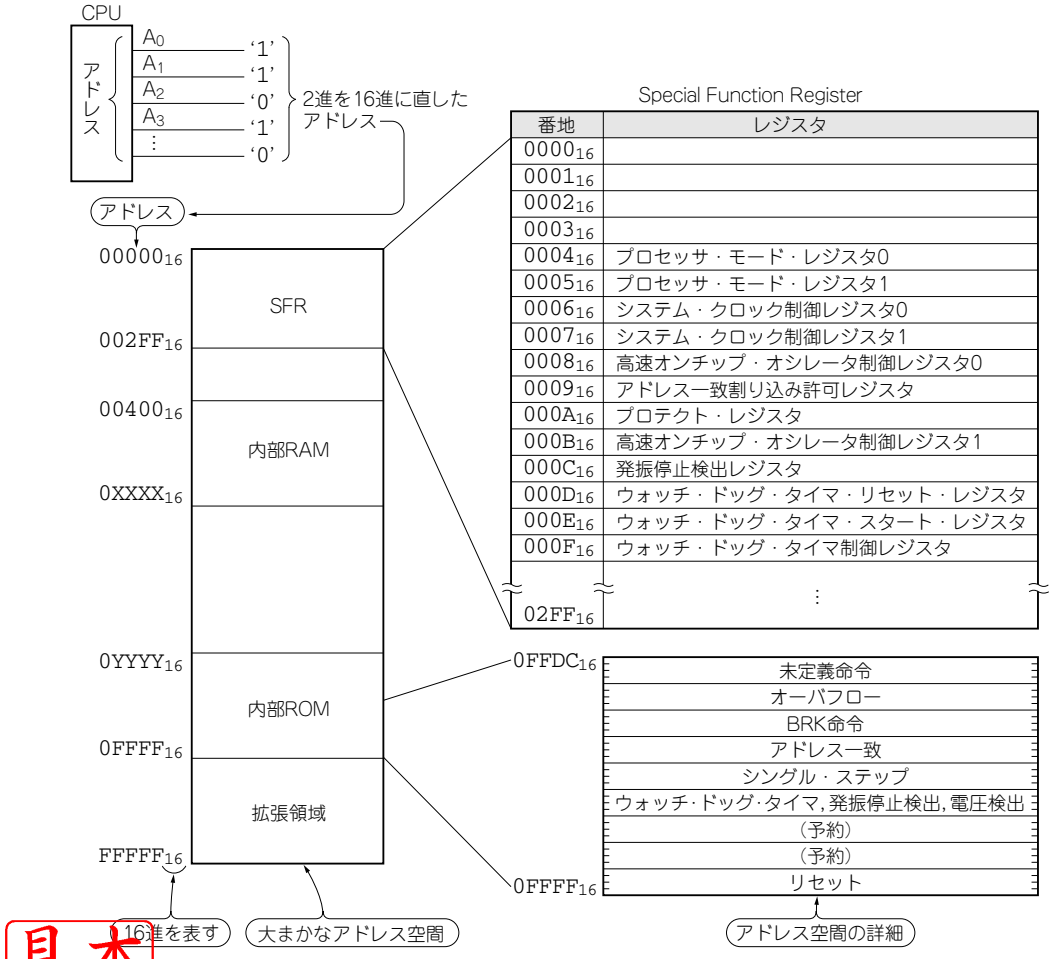


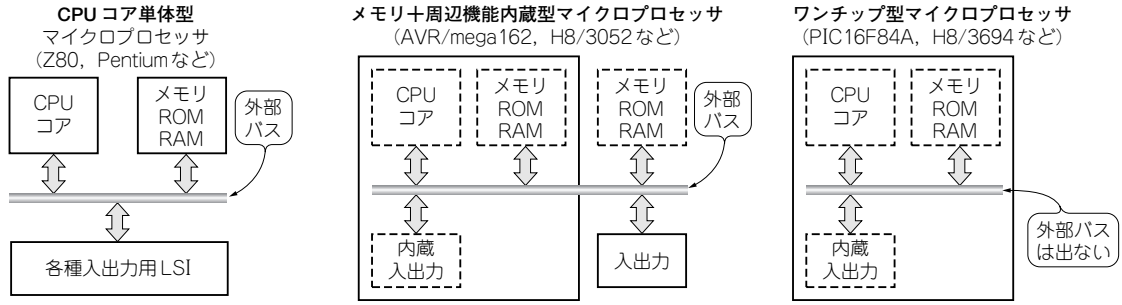
図1.6　アドレス空間

行われます．**図1.5**はマイクロコントローラの一つであるH8系の内部アクセスのバス・タイミングです．データ・アクセスの動作はCPUがアドレスを出力した後，リード，ライトのコントロール線を使って周辺に対して入出力のタイミングを知らせます．H8以外のCPUでも，基本的には同様のリード・サイクルとライト・サイクルが実行されます．

アドレスを使ったバス構造からはアドレス空間という概念が作られます．この空間はメモリ・マップとも呼ばれます．**図1.6**はR8系CPUのメモリ・マップです．データをやりとりする先がアドレス空間にマッピングされるという構造は**図1.4**のようなアドレス・バスとデータ・バスが物理的に存在するシステムだけでなく，ネットワークなど数本のシリアル線のみで構成されるシステムでも実現可能です．

● マイクロプロセッサとマイクロコントローラ

マイクロプロセッサは**図1.7**に示すように，（**a**）CPUコア単体，（**b**）CPUコア＋周辺，（**c**）CPUコア＋周辺＋外部バスなし，といった構成をとるチップに分類できます．パソコンに使われるCPUは自由



（**a**）CPUコア単体型マイクロプロセッサによるシステム　（**b**）メモリ＋周辺機能内蔵型マイクロプロセッサによるシステム　（**c**）ワンチップ型マイクロプロセッサによるシステム

図1.7　マイクロプロセッサとマイクロコントローラ

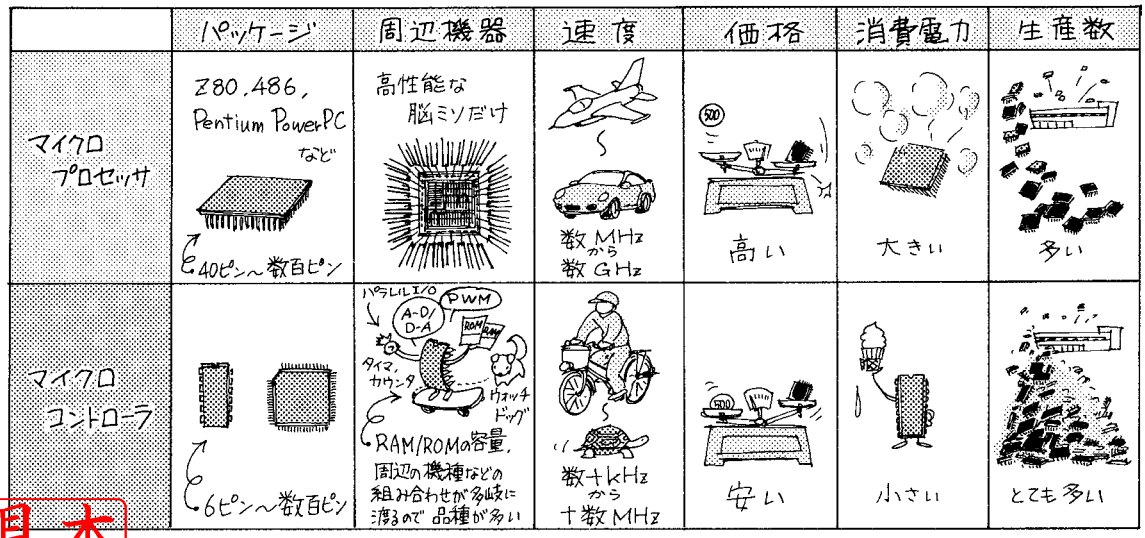


図1.8　マイクロプロセッサとマイクロコントローラの比較

な拡張性が必要なので「CPUコア単体」が用いられ，PICなどの組み込み用CPUは「CPUコア＋周辺＋外部バスなし」が用いられます．CPUに周辺装置を付加したチップをマイクロコントローラやワンチップ・マイコンと呼ぶことも多いようです．

図1.8にマイクロプロセッサとマイクロコントローラの違いを示します．マイクロプロセッサが誕生した当初1970年ごろは，マイクロプロセッサとマイクロコントローラとの違いはあまりありませんでしたが，現在ではそれぞれの用途に合わせて異なる進化を遂げています．

1.3　パソコンの外部インターフェース

この節では，パソコンの外部インターフェースについて紹介します．パソコンの外部インターフェースには，**図1.9**に示すようなものがあります．これらのうち，RS-232Cと以前プリンタが接続されていたパラレル・ポートは，旧来からのインターフェースということでレガシ(Legacy)インターフェースと呼ばれます．

また，これらのインターフェースは「USB-シリアル変換アダプタ」や「USBインターフェースを増設するPCIカード」といった装置があるので，現在でもある程度は互いに変換可能です．

● PCI

PCI(Peripheral Components Interconnec t)バスは，パソコン内部のブロック間を結ぶバスの規格として，Intel社を中心に策定され，現在，ほとんどのパソコンに採用されています．PCI規格では，バス幅32ビット，33MHz動作が基本で，この場合，最大データ転送速度は133MB/sとなります．PCI規格ではバス幅64ビット，66MHz動作で最大533MB/sの高速な仕様まで規定されており，サーバなどの用途で使われています．

さらに，PCI規格をベースに拡張してより高速なデータ転送に対応したPCI-Xや，伝送方法をシリアル伝送に変更してさらに高速化を図ったPCI Expressもあります．また，伝送性能はそのままで組み込みシ

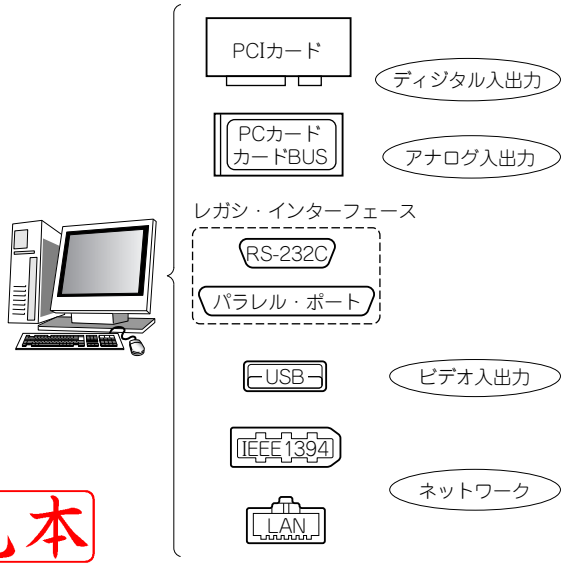


図1.9　マイコンの外部インターフェース

ステム用に小型化したCompact PCIやノート・パソコン用のMini PCI，小型パソコン用のLow Profile PCIと呼ばれるものもあります．

一番ベーシックな32ビットPCIバスでは，それまでのレガシなISA（Industrial Standard Architecture）バスなどとは異なり，32ビットのアドレス・バスと32ビットのデータ・バスが時間分割されて32ビットの信号線でやりとりされます．ですから，最大データ転送速度の133 MB/sは，データが連続して転送されるバースト転送時に得られます．また，PCIでは本来のデータ転送の前にコンフィグレーション・サイクルを行うことにより，PCIバス空間にマッピングされるボード固有のアドレスを自動的に決定できます．この動作は，インターネット（TCP/IP）で使われるDHCP（Dynamic Host Configuration Protocol）のようなものと考えるとわかりやすいでしょう．このように，PCIバスを利用する上での取り決めは多岐にわたり，適合させるボードを作るのは大変難しいといえます．

実際のマイコンとPCIの関係を，最近の平均的なパソコンでよく使われるIntel製865GVチップセットを使った例を**図1.10**に示します．少し前のパソコンではPCIはその名のとおり，ノース・ブリッジとサウス・ブリッジの間に利用されていましたが，現在ではサウス・ブリッジから出ている外部バスの一つという役割のようです．

図1.11に5V，32ビット版のPCIカードのサイズを示します．32ビット版には3.3V版もあり，**図1.12**のようにカードに切り込みを設けることで区別しています．**写真1.1**に5V 32ビット版のスロット，**写真1.2**にLow Profile PCIサイズのユニバーサル32ビット版カードを示します．

● PC Card

PC Cardは，主にノート・パソコン用に，米国の「PCMCIA（Personal Computer Memory Card International Association）」と日本の「JEIDA（Japanese Electronic Industry Development Association）」が共同で策定したカード型拡張機器の規格です．当初は可搬型のメモリとして登場しましたが，その後

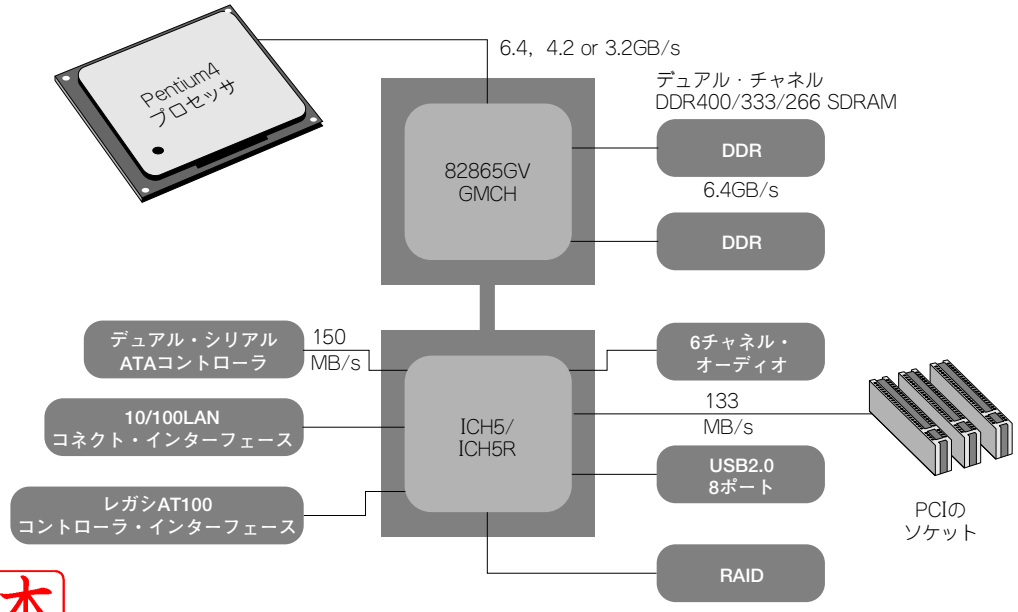


図1.10　マイコン・システムとPCI（865GVチップセットの場合）

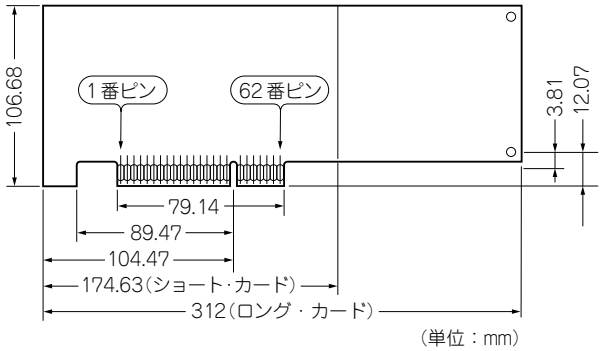


図1.11　5V 32ビット版のPCIカード

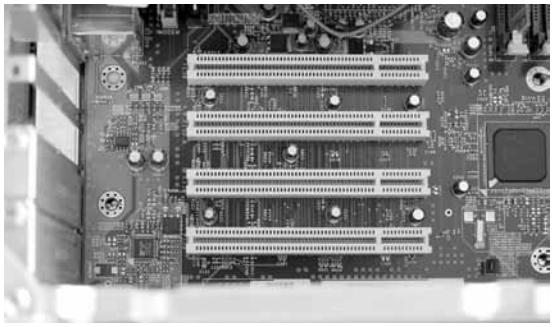


写真1.1　5V 32ビット版のスロット

写真1.3　様々なPC Card

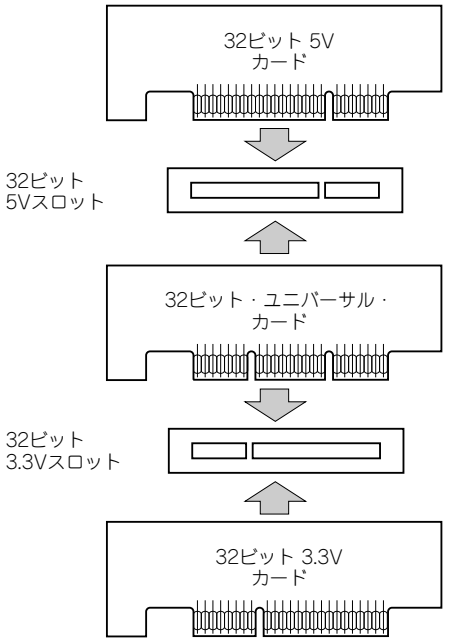


図1.12　各種32ビット版PCIカードとスロット

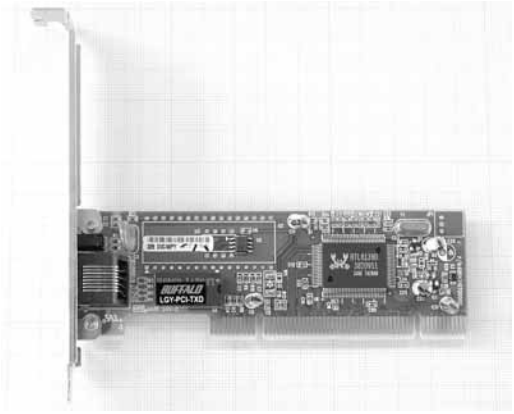


写真1.2　ユニバーサル32ビット版 Low Profile PCIカード

I/O機能が追加され，さらにPCIバスに準ずる機能も包含されました．これらは正式には「PC Card Standard」として定義されますが，大きく分けると，16ビット版の16ビットPC Cardと32ビット版のCardBusの2種類があります．

16ビットPC Cardは旧来のISAバス（PCIバス以前の拡張バス規格，実際的な性能はPCIの数分の1程度）程度の性能，CardBusは33MHz 32ビットのPCIバス程度の性能（前述のように最大転送速度133MB/sといわれています．また，両者のバスで特徴的なことは，16ビットPC Cardは16ビットのデータ・バスと26ビットのアドレス・バスが独立して存在するのに対し，CardBusはPCIバスのように32ビット

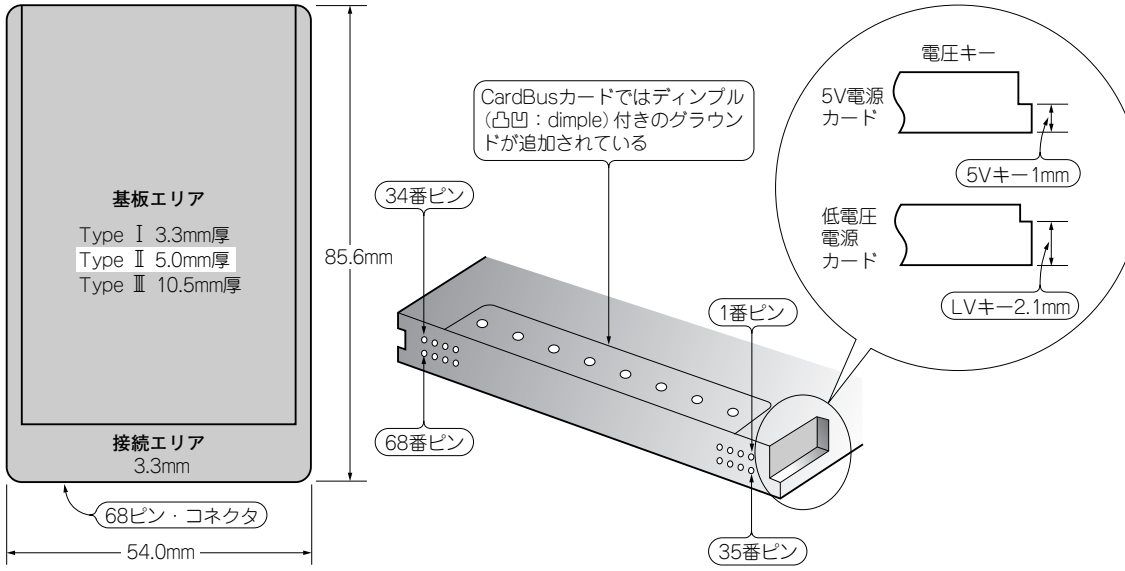


図1.13　PC Cardの外観

のデータ・バスとアドレス・バスが同じところに割り付けられた共用バスとなります．したがって，外観やプラグ＆プレイなどの使い勝手はほぼ同じように見えますが，ハード的にはまったく別物と考えてもよいくらいです．

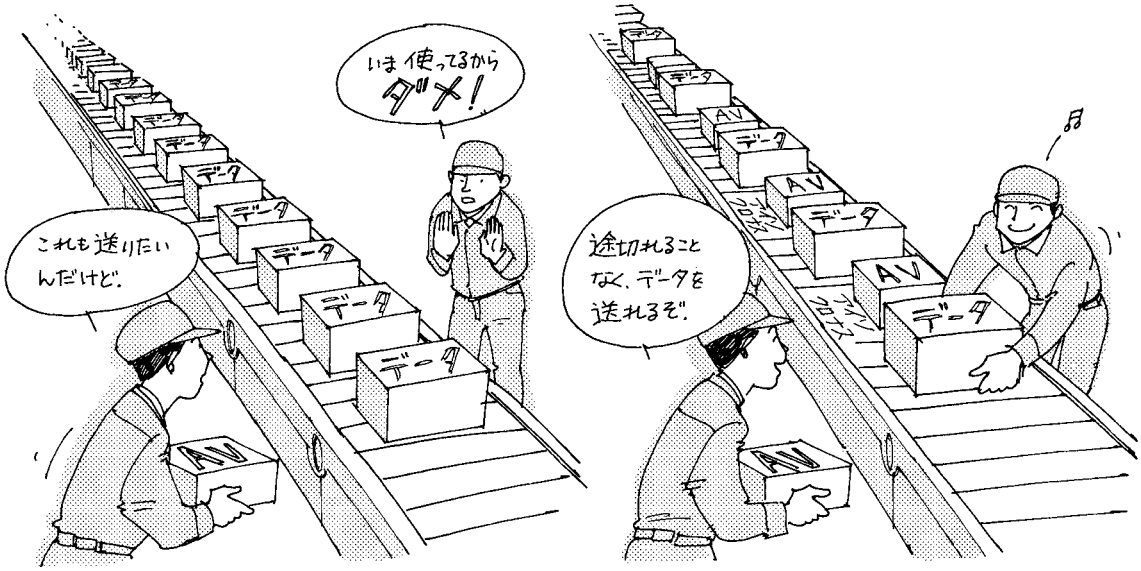
なお，ディジタル・カメラや組み込みシステムで利用されるCompact Flash（CF）Cardは16ビットPC Cardのアドレス・バスの線数を減らしたものです．最近になって，PCIがPCI Expressに進化したように，シリアル伝送を利用したより高速で小型化されたExpress Cardも対応パソコンを含め出まわりはじめています．

図1.13にPC Cardの外観，**写真1.3**に様々なPC Cardを示します．**写真1.3**では左側二つが16ビットPC Card，右側二つが32ビットのCardBusで，4点とも厚さ5mmの一般的に使われているType-IIの形状となります．

● USB

パソコンの世界を，現在のように幅広く使えるように進化させた大きな要因のひとつにUSBがあるのではないかと筆者は考えます．なぜなら，それまでのパソコンは，たとえばプリンタのセントロニクス・パラレルやSCSI（スカジ：ハードディスクなどで使われていたレガシ・インターフェース）のケーブル，コネクタなど，性能はUSB接続と変わらないにしても「ごつくて」「ねじ止め」で「計算機然」としていました．それがUSBになることで，「しなやか」で「スマート」な印象に変わりました．

また，USBの機能として特徴的なことのひとつは，**図1.14**に示すようなアイソクロナス転送を可能にしたことです．つまり，ずっと一つのデバイスがバスを占有しないような仕組みを最初から備えていたので，USBオーディオなどアナログ電子機器の使い勝手がパソコンで実現できるようになりました．また接点が少なくなったのでコネクタのコストも下がり，デバイスを特定するためのID設定などからも解放され，さらに，テーブル・ライトのようなUSB（から電源をとるだけの）小物類なども現れて，パソコン

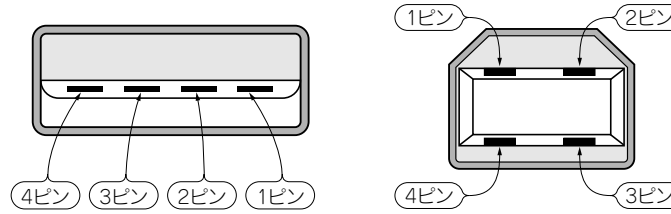


（a）SCSIやイーサネットなど　　（b）USBやIEEE1394のアイソクロナス転送

図1.14　アイソクロナス転送

ピン番号	名　称
1	VBus
2	−Data（D−）
3	＋Data（D＋）
4	GND

（a）ピン配置



（b）標準Aプラグ　　（c）標準Bプラグ

図1.15　USBのコネクタ

をより身近なものにしています．

このように紹介すると良いことずくめのようですが，1点だけ残念な点がありました．それは旧来のRS-232Cなどで可能であった簡単な電子工作レベルの入出力（シリアル）に利用することが難しくなったことです．しかし，後の章で紹介するように，最近になって安価で使い勝手のよいUSB-シリアル変換アダプタなどが簡単に入手できるようになり，この点も克服できるようになりました．

当初（1996年），USBは12 Mbps以下の通信速度で，同時に127個のデバイス（ルート・ハブ以外のハブも数える）が接続可能でした．その後（2000年），480 Mbpsまでの転送速度が追加されたUSB 2.0規格となりました．現在では，USBは通信速度1.5 Mbpsのロー・スピード（キーボード，マウス，ゲーム周辺機器などの接続），通信速度12 Mbpsのフル・スピード（オーディオ，中速ネットワークなどの接続），通信速度480 Mbpsのハイ・スピード〔ビデオ，高速ネットワーク，ハードディスク（HDD：Hard Disk Drive）などの接続〕の三つのモードが利用されます．

USBのコネクタとハードウェアの概要を**図1.15**，**図1.16**に示します．USBの通信は物理的にはたった4本の信号線で行われます．そのうち2本は電源用ですし，信号線は差動信号ですから，信号の系統は

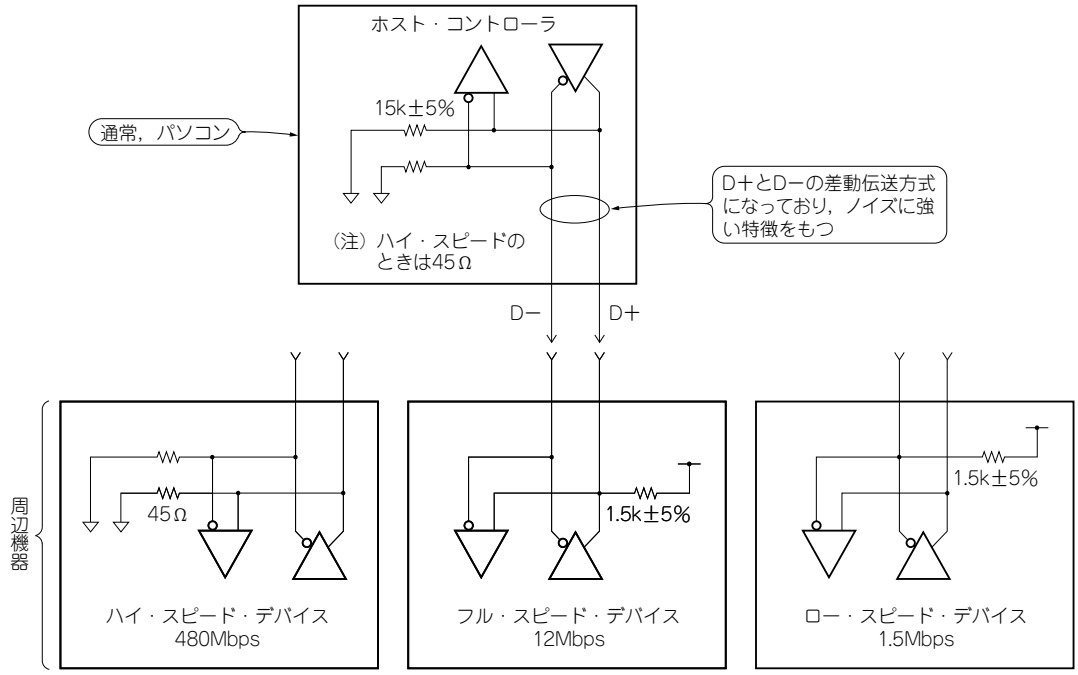


図1.16　USBのハードウェア

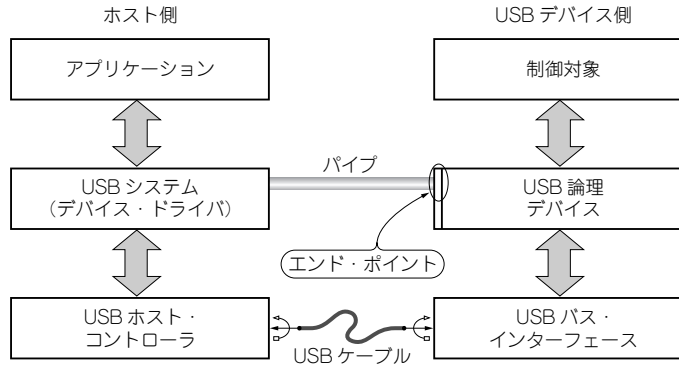


図1.17　USBの構成

1系統しかないわけです．この1系統の信号線を使って半二重通信を行います．このシンプルなハードウェアの上に階層構造をもつプロトコルが構築され，多様な通信を可能にしています．

これらのハードからソフトまでの全体がUSBという規格です．USBの全体構成を**図1.17**に示します．アプリケーションからは，複数の仮想的な通信線がターゲットとの間に結ばれているように見えます．これをパイプと呼びます．

● RS-232C

RS-232C（Recommended Standard 232 version C：アール・エス・ニー・サン・ニ・シー）はパソコンが世に出た当初から使われているシリアル通信の規格です．当初はパソコンとモデムとを接続するためのも

ので，コントロール線の名称にその名残を見ることができます．現在ではEIA-232F，あるいはEIA-574（9ピン版のEIA-232F）と呼ぶのが正式のようです．しかし，パソコンのカタログなどを見ると「シリアル・ポート：RS-232C D-SUB 9ピン，16550A互換」といったEIA準拠という意味の書き方をしていることが多いようです．また，データ通信を行うTxD，RxD以外に数本のコントロール線があり，Windows APIから制御可能です．このコントロール線を使ってフロー制御を行うこともできますが，TD，RD以外の線はあまり使われないようです．なお，16550A互換というのは，PC/ATの後に出てきたPS/2で採用された非同期通信用デバイス（UART：Universal Asynchronous Receiver Transmitter）の型番で，現在では，チップセットの中に機能として搭載されています．

RS-232Cの信号線はパソコン内部でよく使われる5Vや3.3Vのロジック・レベルでなく±15Vが利用されます．したがって，通常のロジックICやCPUなどと接続する際には必ず電圧レベルの変換が必要です．レベル変換が必要ですが，それ以外は単純な非同期通信が行われるだけなので，受け手の設計が簡単にすみ，現在でも簡易な通信方式として根強く利用されています．**図1.18**にPC/AT互換機で使われる9ピン・コネクタと各信号線を示します．

● パラレル・ポート

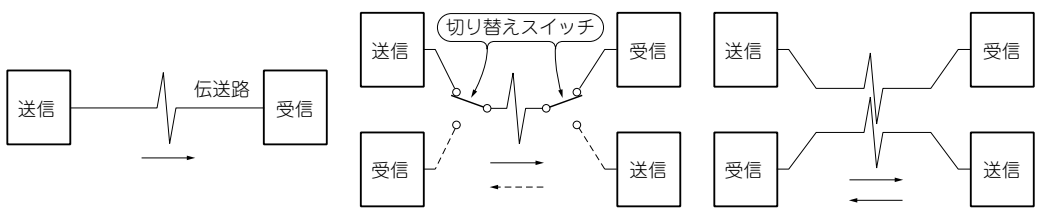
8ビット・パラレル・ポートは，米国のセントロニクス社のプリンタ接続ケーブルの規格をもとに，「セントロニクス準拠」としてパソコンのプリンタとの接続ポートとして標準インターフェースとなりました．その後，双方向通信の可能な「ニブル・モード」「バイト・モード」が追加され，さらに，高機能な

Column … 1.1　半二重と全二重

図1.Aに示すように，双方向通信でも通信路として1系統を交互に利用するものと，2系統を同時に利用する方法があります．それぞれを半二2重，全二重と呼びます．たとえばトランシーバと携帯電話を考えると，トランシーバは交互にしゃべるので半二重，携帯電話は同時にしゃべることができるので全二重となります．

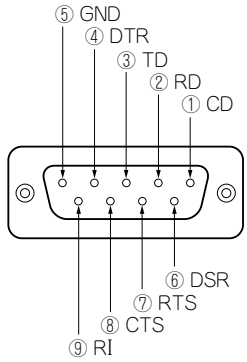
一般的には全二重のほうが使い勝手が良さそうですが，いわゆる上り下りの通信が同時あるいは均等に行われることはむしろ希で，通信路の資源と効率を考えると半二重のほうが効率的な場合が多いようです．ブロードバンドのADSL（Asymmetric Digital Subscriber Line）などもどちらかというと半二重的な通信ともいえます．

ところで，糸電話は半二重でしょうか全二重でしょうか？



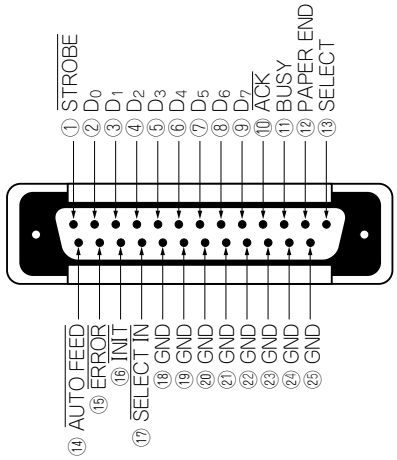
（a）単方向伝送　（b）半二重伝送　（c）全二重伝送

図1.A　半二重と全二重



ピン番号	I/O	論理	略号	原文	機能(もともとの意味)
①	I	正	CD	Carrier Detect	相手のモデムが送信の準備ができている
②	I	負	RD	Receive Data	受信したデータ
③	O	負	TD	Transmission Data	送信するデータ
④	O	正	DTR	Data Terminal Ready	端末(パソコン)の通信準備ができている
⑤			GND	Ground	グラウンド
⑥	I	正	DSR	Data Set Ready	モデムの通信準備ができている
⑦	O	正	RTS	Request To Send	送信要求. 全二重ではつねにON
⑧	I	正	CTS	Clear To Send	送信可(送信を許可する)
⑨	I	正	RI	Ring Indication	電話回線に呼び出しを受けている

図1.18　PC/AT互換機で使われる9ピンRS-232Cのコネクタと信号線
EIA-574規格，通常パソコン側にDサブのオス・コネクタが使われる．データ線は負論理(−5〜−15 Vが'1')，コントロール線は正論理．



ピン番号	I/O	信号名	ピン番号	I/O	信号名
①	O	$\overline{\text{STROBE}}$	⑩	I	$\overline{\text{ACK}}$
②	I/O	D_0 (LSB)	⑪	I	BUSY
③	I/O	D_1	⑫	I	PAPER END(紙なし)
④	I/O	D_2	⑬	I	SELECT(選択)
⑤	I/O	D_3	⑭	O	$\overline{\text{AUTO FEED}}$(自動紙送り)
⑥	I/O	D_4	⑮	I	$\overline{\text{ERROR}}$(エラー)
⑦	I/O	D_5	⑯	O	$\overline{\text{INIT}}$(初期化)
⑧	I/O	D_6	⑰	O	$\overline{\text{SELECT IN}}$(選択)
⑨	I/O	D_7 (MSB)	⑱〜㉕		GND

図1.19　PC/AT互換機で使われる25ピン・パラレル・ポートのコネクタと信号線
IEEE1284規格，パソコン側はDサブのメス・コネクタが使われる．

「ECP (Extended Capabilities Port) モード」，「EPP (Enhanced Parallel Port) モード」が追加されました．当初の出力のみの動作を「互換モード」と呼び，これらをまとめてIEEE1284と呼びます．

これらのモードは，ポートのネゴシエーション 動作時に周辺側から転送モード・データをパソコン側に送信することにより切り替えられます．転送モード・データが受け取れなかった場合は互換モードとなります．互換モード以外はWindows APIではサポートされず，専用のデバイス・ドライバの組み込みによって利用可能になります．**図1.19**にPC/AT互換機で使われる25ピン・パラレル・ポートのコネクタと信号線を示します．

前述のRS-232C，パラレル・ポート，さらにISAバスなどPC/AT時代のインターフェースをレガシ・インターフェースと呼び，新規設計のパソコンの場合，採用しない方向です．

1.4　制御システムのソフトウェア

外部制御を行うためのソフトウェアの作成は，標準的なパソコンでデータのみを処理する場合と異なり，いくつかの仕組みを理解しておく必要があります．この節では，制御システムのソフトウェアの特色について概要を紹介します．

● ネイティブ開発とクロス開発

外部制御を行うためのソフトウェアの場合も，当然開発システムあるいは開発ツールと呼ばれる，コンパイラなどの言語ツール，デバッガ，これらを統合する統合プラットホームなど開発環境が必要です．この開発システムには**図1.20**に示すように大きく分けてネイティブ開発とクロス開発に分けられます．ネイティブ開発あるいはセルフ開発は，開発環境と実行環境が同じマシンあるいは同じ種類CPUのマシンの場合です．クロス開発とは開発環境と実行環境のマシンが異なる，とくにCPUが異なる場合を言います．

たとえば，Windowsアプリケーションをマイクロソフト社の開発ツール＆言語のVisual C++を使って作成する場合はネイティブ開発，Windowsパソコンでマイクロチップ・テクノロジー社のPIC開発統合

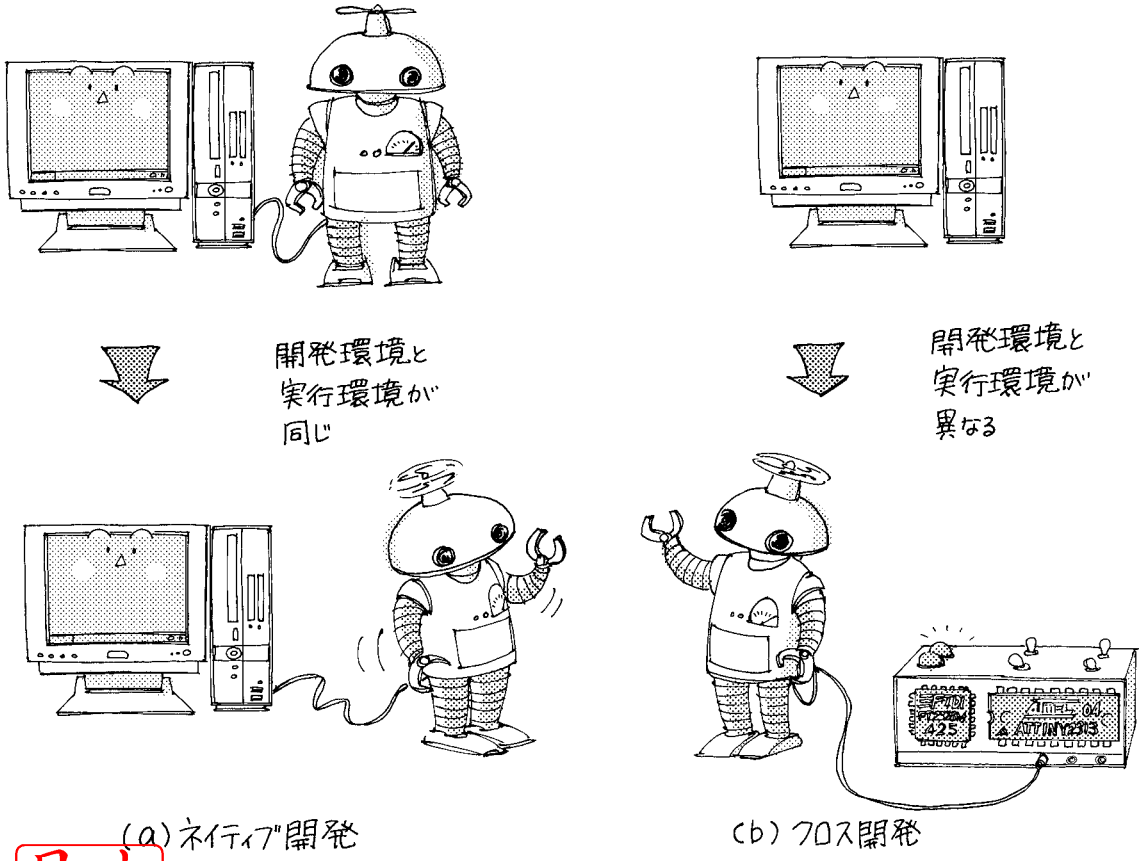


図1.20　ネイティブ開発とクロス開発
開発環境は実行環境と同じマシンで可能とはかぎらない．

環境MPLABを使ってPIC用のプログラムを開発する場合はクロス開発になります．また1.1節で紹介した，「パソコンを中心とするシステム」の場合はネイティブ開発，「組み込みシステム」の場合はクロス開発，「パソコンとマイクロコントローラを組み合わせたシステム」の場合は両方の開発方法を利用することになります．

● アセンブラと高級言語

最近ではクロス開発の場合でも，C言語の開発環境が多くの場合無償あるいは低価格で用意されています．ですから，たとえばPICやAVRなどを使った小規模なシステムでもC言語で開発することも多くなってきました．C言語で開発すれば確かに開発効率は上がります．ですが，初めからC言語で始めるのはあまり適当ではないと筆者は考えます．ある程度CPUの特性を把握するまではアセンブラで組み立てるほうが望ましいと思います．ある程度ようすをつかんでからC言語で開発を始めることをお勧めします．また「パソコンとマイクロコントローラを組み合わせたシステム」の場合は，コントローラ側をシンプルに作ることができるので，アセンブラでも十分といえます．

逆に「パソコンを中心とするシステム」の場合は，Windows APIや追加した外部制御デバイス用のライブラリはC言語から呼び出すスタイルが標準なので，こちらはC言語を中心に考えるのが適当と思います．同様に「パソコンとマイクロコントローラを組み合わせたシステム」の場合も，パソコン側はC言語で開発ということになります．

● ハードウェアのドライブ

キーボードやディスプレイなどパソコンに標準装備されるハードウェアとのやり取りはWindowsの中のデバイス・ドライバが担当します．デバイス・ドライバはハードウェアに対して基本的な操作を提供します．このデバイス・ドライバに対してもっともわかりやすい形で機能を提供するものがWindows APIとなります．**図1.21**に示すようにパソコンに追加した外部機器をドライブする場合もデバイス・ドライバを使用することになります．追加するデバイス・ドライバは，Windowsの管理下で登録され動作

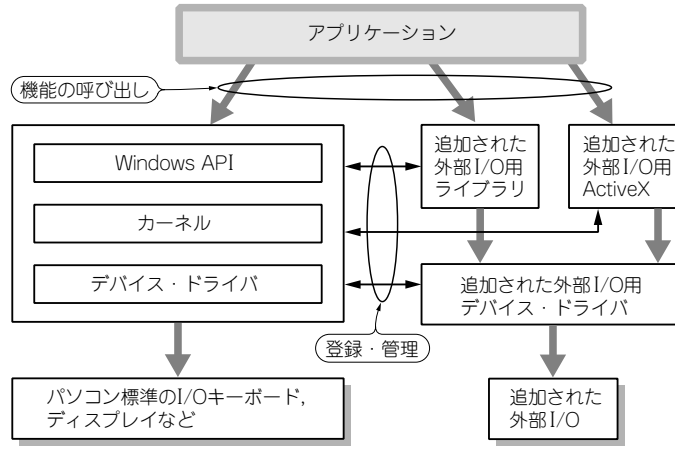


図1.21　アプリケーションからのハードウェアのドライブ
新たに追加されたハードウェアもデバイス・ドライバやライブラリを使えば単純な操作関数の呼び出しで使用できる．