

ATA (IDE) /ATAPI の 歴史と概要

熊谷 あき

1 ATA(IDE)の歴史

● IDEの誕生

磁気記録面からヘッドが少しだけ浮いている、いわゆるウィンチェスター方式のディスク・ドライブが、初めて市場に出てからもうすでに36年が経とうとしています。

IDEの起源は、米国 Seagate Technology 社が開発したハード・ディスク・ドライブ ST-506 だといえます。米国 IBM 社が初代パソコンでこれを採用したため、PC/AT にもそれが引き継がれ、多くのベンダがこの ST-506 互換のハード・ディスク・ドライブを製造・販売してきました。したがって、この ST-506 が発売された 1980 年が、いわば現在の IDE の元年とも呼べる記念すべき年です。

さらに、WD1003 などのハード・ディスク・コントローラによって巨大なシェアを誇った米国 Western Digital 社は、高まるハード・ディスクの需要に応える形で、米国 Compaq Computer 社と共同で、パソコンの拡張スロットを使用せず、直接マザーボード上の 40 ピンのコネクタでハード・ディスクを接続する方法を開発しました。それが 1986 年に仕様がまとめられた IDE インターフェースです。

IDE は ST-506 と BIOS レベルで互換性があり、従来の ST-506 ベースの PC/AT 互換機でも問題なく動作するように工夫されていました。

この IDE という名称の意味には諸説があるようですが、一般的には Integrated Drive (または Device) Electronics の略とされています。

● IDE と SCSI

IDE は CPU が直接ドライブをコントロールするため、簡単なインターフェース回路で接続できます。そのためコスト・メリットが大きく、また不足気味の ISA カード・スロットを使用しなくても済むという利

点があります。

一方、IDE と同じころに使われ始めた SCSI (Small Computer System Interface) は、動作が複雑なので専用のインターフェース・ボードが必要でした。また当時は SCSI も IDE も速度的にそれほど変わりがなかったため、IDE ハード・ディスクが PC/AT 互換機における標準の地位を獲得しました。

● IDE から Enhanced IDE, そして ATA/ATAPI へ

しかし当時の IDE ドライブは、各メーカーごとに仕様が微妙に異なっていたため、1 台目 (マスタ) と 2 台目 (スレーブ) に同一のメーカーのドライブを使用しなければ動作が保証されないような状態でした。

そこで、ハード・ディスク・メーカー各社が集まって仕様を統一し、ANSI (American National Standards Institute) において、1988 年に ATA (AT Attachment) インターフェースという仕様が作成されました (表 1)。

ただ、仕様の統一や標準化には時間がかかるのが常です。ATA-1 が決まったところには、すでにより大容量でより高速な IDE が望まれていました。結果的に、市場には Enhanced IDE として ATA-2 の仕様を先取りしたハード・ディスクが市販されるようになりました。

その後も ATA は拡張を続けました。ATA-3 では低速な転送モードが廃止されたり、CD-ROM ドライブなどのハード・ディスク以外のデバイスを接続するための ATAPI (ATA Packet Interface) が規格化されました。ただし、ATAPI 仕様書は、当初 ATA とは別規格であり、ATA/ATAPI と一本化されたのは ATA-4 の世代からです (したがって、正確には ATA/ATAPI-4 と表記する)。

● 高速化と大容量化

ATA の歴史における大きな変化として、ATA-4 で採用された UltraDMA 転送方式の登場があるでしょう。PIO 転送方式やマルチワード DMA 転送方式では 16M バイト/秒が上限だったところへ、UltraDMA 転送モード 2 ではその倍の 33M バイト/秒まで引き上げ

ATA/ATAPIのハードウェア構成の詳細

熊谷 あき

ここではまず、ATA/ATAPIのハードウェア構成やコネクタ・ピン配置, 電気的特性など, 物理仕様について解説する。また, ATAレジスタのマッピングやレジスタの概要も示す。なお, 以降の解説はすべてATA/ATAPI-5を基本に解説する。なお, 仕様書では負論理の信号は信号名の後に“ $\bar{}$ ”を付けるが, 本文中で示すと区切り文字などと区別がつかなくなるので, ここでは信号名の後に“#”を付けて表記する。
(編集部)

1 デバイスの接続

ATAは, 同一ケーブル上に一つまたは二つのデバイスを接続できます。二つ接続した場合, それぞれのデバイスの識別方法として, ケーブル・セレクト(CSEL)信号を使用する方法と, マスタ/スレーブのジャンパ・ピンを使用する方法の二つがあります。

接続ケーブルは, コネクタが三つあるものが一般的で, ハード・ディスクなどのストレージ・デバイスを二つと, ホスト・コントローラ一つを接続できます。ケーブルの端にホスト・コントローラを接続し, もう一方の端と真中のコネクタにデバイスを接続します。

デバイスを一つだけ接続する場合は, ケーブルの端のコネクタに接続し, 中央のコネクタを空き状態にする必要があります(図1)。中央に接続して, 端のコネクタを空き状態にすると, 反射などで信号が劣化し, 正しく伝送できない場合があるので注意してください。

● デバイス番号の付け方

ATAはホスト・コントローラ一つに対して二つのデバイスを接続できます。同一のケーブルに二つの

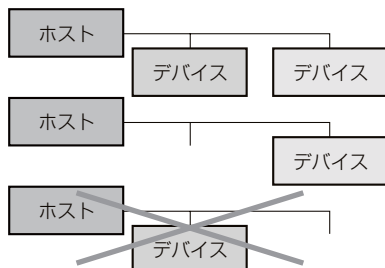


図1 デバイスの接続方法

デバイスを接続する都合上, デバイス番号を与えなければホスト・コントローラ側でそれぞれのデバイスをコントロールできません。

片方をデバイス0(マスタ), もう片方をデバイス1(スレーブ)と呼びます。デバイスを区別する方法として2種類が規定されており, システム設計者はどちらかを選択することができます(図2)。

▶ CSEL信号を用いる方法

ケーブル上でデバイス番号を決定します。そのため, この方法でデバイスを区別する場合, 図2に示すようにケーブルとコネクタの間は, CSEL信号をカットしたケーブルを使用する必要があります。

ホスト側はCSEL信号を常にGNDに接続しておき, デバイス側では信号をプルアップしておきます。よってホスト側とCSEL信号がつながっているとGNDに, つながっていないければ V_{cc} になり, これを利用して自分がマスタなのかスレーブなのかデバイス番号を識別できます。

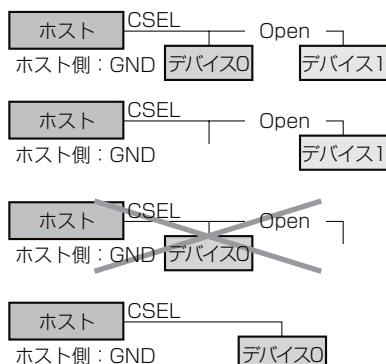


図2 CSEL信号によるデバイス番号の区別

PC カード ATA と TrueIDE

大中 邦彦

● PC カード ATA

ATA (IDE) 規格とは別の形態として、PC カード ATA と呼ばれる規格があります。この規格は、PC カード・スロットにハード・ディスクやフラッシュ・メモリ・カードなどのストレージ・デバイスを接続するためのものです。PC カードとして見ると、PC/AT 互換機の IDE コントローラ相当の機能をもった I/O カードということになるのですが、ほかの種類の PC カードよりも細かな部分まで規格として定められています。

もともと PC カードは、CIS (Card Information Structure) と呼ばれるカードの種類や属性を記した ROM をもっていて、それを解読すればそのカードを使うための細かな情報を得ることができます。しかしそれは、カードの自由度を上げてしまうことになるので、同じような種類のカードであっても細かな仕様は異なるものが多数存在してしまいます。

結果的に、共通に使えるドライバの作成が難しくなり、個別のドライバで対応せざるをえなくなります。個別にドライバが必要ということは、PC カードを汎用的なブート・デバイスとして使用することが難しいということを意味します。

PC カード ATA 規格では、カードをブート・デバイスとして利用できるように基本的な機能を規定しています。たとえば、I/O ベース・アドレスやコンフィグレーション・インデックス番号の対応などを明確に決めているので、ATA カードであることさえわかれば、コンフィグレーション・レジスタやコンフィグレーション番号などを決めうちで使用できるようになっています。

しかし、PC カードを通常の I/O カードとして使用できるようにするには、コンフィグレーションという作業が必要なものに変わりはなく、起動する OS を格納するデバイスとして利用するのは簡単とはいえないものとなっています。

● TrueIDE

そこで、コンフィグレーション作業をせずにいきなり ATA カードを利用する方法として、TrueIDE モード (68 ピン ATA モード) と呼ばれる動作モードをオプションとして定めています。TrueIDE モードは、IDE インターフェースと同等の信号をもった IDE インターフェースと互換の動作モードです。

TrueIDE モードをサポートしたカードならば、OE 端子 (9 番端子) を“L”に固定して PC カードの電源 (V_{CC}) を投入すれば、TrueIDE モードに移行します。表 A に I/O モード・インターフェース、PC カード ATA モード、TrueIDE モードのピン配列を示します。

TrueIDE モードの信号のうち、PC カード ATA モード (I/O インターフェース・モード) との違いを説明しましょう。

まず、IDE と同じようにアドレス・ビットが A0～A2 の 3 本しかありません。それ以外のアドレス・ビットは 0 に固定します。

IDE の CS0# と CS1# は、それぞれ PC カード・インターフェースの CE1# (7 番ピン)、CE2# (42 番ピン) に割り当てられています。また、DASP# と PDIAG# がそれぞれ SPKR# (62 番ピン) と STSCHG# (63 番ピン) へ割り当てられています。

割り込み端子は 16 番ピンに割り当てられており、PC カード・インターフェースのときと同じですが、TrueIDE モードではアクティブ H になるので注意が必要です。REG# (61 番ピン) は“H”に固定します。

TrueIDE モードをサポートしたフラッシュ ATA カードは、OE 端子を“L”に固定してしまえば超小型の IDE ドライブとして動作させられます。自作の組み込みシステムに小型のドライブを接続したいが、PC カード・コントローラを載せるのは大げさすぎるという場合などに非常に便利です。

ただし、あくまでこの規格はオプションなので、すべての ATA カードが TrueIDE モードをサポートし

ATA/ATAPI の 各種データ転送の概要

熊谷 あき

ATA の転送方式には、CPU が直接データ転送を行う PIO 転送方式、PC/AT 互換機に搭載されている DMA コントローラに仕様を合わせたシングル/マルチワード DMA 転送方式、さらに PCI バス時代になって登場した専用の DMA コントローラで実現する UltraDMA 転送方式がある。ここではそれぞれの転送方式について解説する。 (編集部)

1 データ転送を行うのは誰か

● ATA 仕様とデータ転送の実体

ATA ではデータ転送方式によって、データ転送を行う実体が異なります。シングル/マルチワード DMA 転送や UltraDMA 転送は、DMA コントローラがデータ転送を行う転送モードです。ただし、ATA の仕様では DMA コントローラそのものの規定まではありません。それは、あくまでもこれを実装するプラットフォーム側の問題だからです。

次節で説明するように、各転送方式にはそれぞれ各 ATA 信号の制御方法やタイミングが細かく規定されています。しかし、その規定に従って ATA 信号のデータ・バス上に用意したデータを、さらにそのプラットフォームの転送先(ほとんどの場合はそのシステムのメイン・メモリになると思うが)までいかにして転送するかは、システム設計者に任されているのです。

仮に CPU の処理能力をデータ転送のためだけに使ってよなら、ATA としてはマルチワード DMA 転送を使い、そこからシステムのメイン・メモリへの転送に CPU 転送を使ってもかまわないわけです。

● PIO 転送

すでに何度も説明したように、PIO 転送は CPU が直接データの転送を行います。ATA レジスタのステータスを見てデータの準備が整ったかどうかを判断し、データ・レジスタにアクセスしてデータをやりとりし、必要な数だけそれを繰り返します。

もう少し具体的に PC/AT 互換機で説明すると、CPU の IN/OUT 命令で ATA レジスタにアクセスし、

MOV 命令でバッファ・メモリ空間を読み書きし、それを LOOP 命令などで繰り返し実行するわけです [図 1 (a)]。

この方法では、システム・バス上で、ATA レジスタのアクセス・サイクルとメイン・メモリへのアクセス

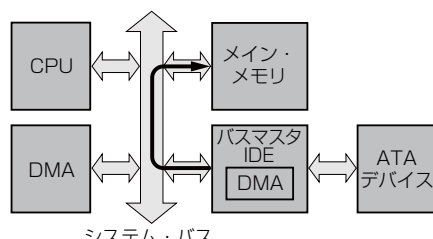
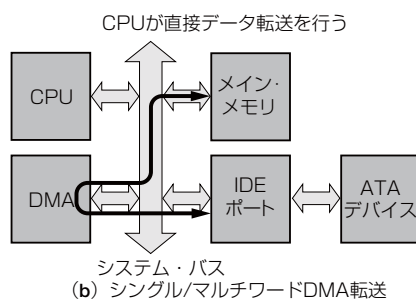
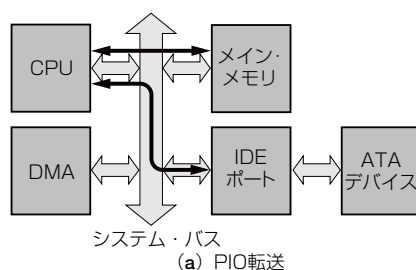


図1 データ転送を行うのは誰か (PC/AT 互換機の場合)

ディスクの構造とパーティション情報の基礎知識

みのうら まこと

1 ディスク装置の基礎知識

● ディスク装置の基本構造

図Aと写真Aに、ディスク装置の基本的な構造を示す。ディスク装置とは、その名のとおり円板上にデジタル・データを記憶する装置を指す。1枚のディスクは、上から見るとトラック(track)と呼ばれる同心円に分割される^{注1}。各トラックは、外側から順に番号が振られ、これをトラック番号と呼ぶ。

さらに、トラックはセクタ(sector)と呼ばれる円弧に分割される。フロッピーディスクなどでは、すべてのトラックが同数のセクタからなるが、すると外周に比べて内周の方が密度が高くなってしまふ。このため、容量を追求する最近のハード・ディスクやMOなどでは、外周の方が多くのセクタからなっている。

1セクタのサイズは固定で、パソコンで用いられるフロッピーディスクやハード・ディスクでは512バイト、以前広く使われていたNEC PC-9801シリーズのフロッピーディスクでは1,024バイト、640Mバイト以上のMOなどでは2,048バイトである。

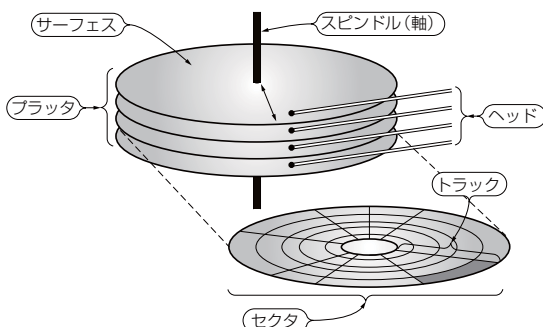
ちなみに、ハード・ディスクやMOなどのセクタ・サイズは、一般的にはハードウェアの仕様で固定されている。しかし、フロッピーディスクの場合、こうし

た分割がフォーマット時に行われ、セクタ・サイズやトラック内のセクタ数などがある程度可変できる。

セクタには、トラックごとにセクタ番号が振られる。

ハード・ディスク装置などでは、一つの装置内にこの円板を複数封入したものが多い^{注2}。円板はプラッタ(platter)と呼ばれるが、ハード・ディスク装置の外側から見た場合、プラッタより問題になるのはその面(サーフェス:surface)である。情報の記録が面上に行われるためである。プラッタの片面しか利用しない場合もある^{注3}。

記録面にはサーフェス番号ともいふべき番号が振られるが、一般に、情報の読み書きを行うヘッド(head)を面ごとに個別でもつドライブが多いため、通常はヘッド番号と呼ぶ。フロッピーディスクの場合はその両面を用いるため、ヘッド数(サーフェス数)は2^{注4}、ハード・ディスクの場合はモデルによって1~十数片



図A ディスク装置の基本構造



写真A 3.5インチ・ハード・ディスク・ドライブの内部

注1: CD-ROMのように、らせん状に1本のトラックで構成されるものもある。

注2: 現在主流のウインチェスター型ハード・ディスクの場合、原理上密閉する必要がある。

注3: 一般的に複数枚を使用するプラッタの場合、一番外側の上下2面は利用されないことがある(図A参照)。

注4: 現在主流の3.5インチで、1.44Mバイトの容量をもつフロッピーディスクを2HDと呼ぶが、この最初の2が2面を指す。

ATA コマンド・プロトコルと ATA コマンドの詳細

熊谷 あき

この章では、各パワー・マネジメント・モードの違いや、それぞれの ATA コマンド・プロトコルの動作の流れ、そしてそれぞれの ATA コマンドのパラメータや戻り値などを解説する。ハード・ディスク・ドライブの読み書きについては、ここで説明されている ATA コマンドを使うだけで十分に制御可能である。 (編集部)

1 パワー・マネジメントと各モード

● 四つのモード

ATA ではパワー・マネジメント機能が標準でインプリメントされています。パワー・マネジメントは四つのモードをもっており、それぞれアクティブ・モード、アイドル・モード、スタンバイ・モード、スリープ・モードです(図1)。電力消費はアクティブ・モードが最大で、次いでアイドル・モード、スタンバイ・モード、スリープ・モードの順に小さくなります。

パワー・マネジメント機能を活用すると、デバイスを使用しない場合や一定期間アクセスがない場合にス

リープ・モードなどへ移行することができ、電力消費を最小限にとどめられます。電力投入時のモードはベンダ定義になりますが、アクティブ・モードまたはスタンバイ・モードで起動します。

● モード遷移

次に、モード遷移の条件について説明します。

アクティブ・モードへは、メディアをアクセスするコマンドを送ったときに遷移します。アイドル・モードへは、アイドル状態にする ATA コマンド (IDLE または IDLE IMMEDIATE コマンド、詳細は後述) を発行したときに遷移します (ベンダ定義による、メディア・アクセスのないときにアイドル・モードへ自動的に遷移するものもある)。

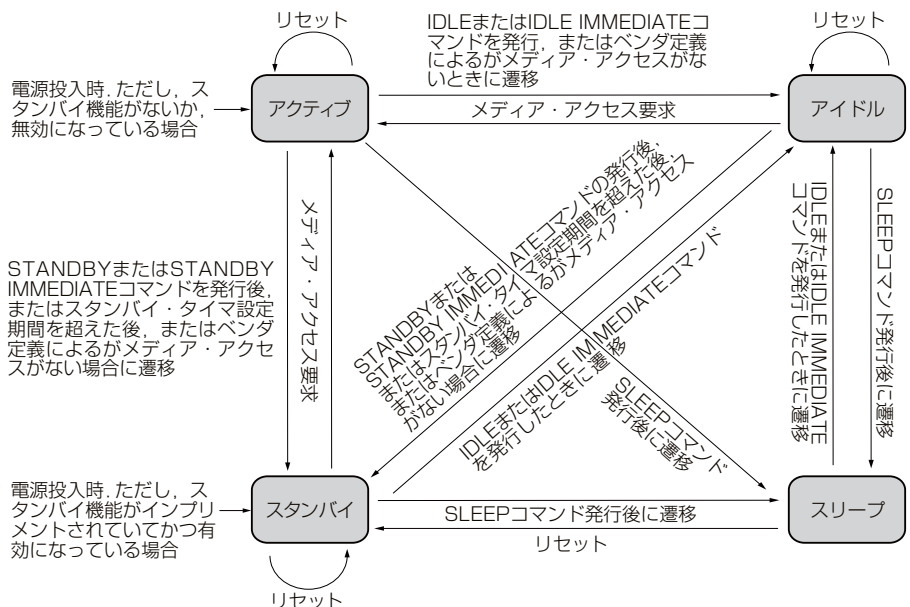


図1 ATA のパワー・マネジメント・モード

ATAPI デバイスの制御方法と パケット・コマンドの詳細

熊谷 あき

従来からのATAコマンドの拡張を最小限にとどめ、従来のATAデバイスを誤動作させないようにパケット・コマンドという概念を採り入れたのがATAPIである。ハードウェアやレジスタ・マップには一切の変更を加えずに、IDEへCD-ROMを接続可能になった。本章ではATAPIデバイスの制御の概要と、ATAPIパケット・コマンドについて解説する。
(編集部)

1 ATAPIの概要

● ATAPIとPACKET Command feature セット

ATAPIとは、PACKET Command featureセットがインプリメントされたATAデバイスの総称です。主なATAPIデバイスは、CD-ROMやDVD-ROMなどのマルチメディア系になります。ATAPIデバイスのコマンド体系は、SCSIのマルチメディア・コマンド・セットと互換性があり、そのコマンド群はSFF 8090などの規格書に記載されています。

ATAPIの考え方は古く、ATA-2のころにIDEインターフェース上でCD-ROMの制御などの複雑なシーケンスやコマンドを使用するために、PACKETコマンドというコマンドとプロトコルを定義し、コマンドとパラメータをセットする上位レイヤとして動作するしくみを考えました。

このコマンドをATAPIパケット・コマンドと呼び、ATAPIパケット・コマンドとパラメータのセットをコマンド・パケットと呼びます。そのため、物理レイ

ヤであるATAの信号をそのまま利用できます。またコマンド・パケットをマルチメディア・コマンド・セットとの橋渡し役として規定しており、その橋渡し役の部分の部分をトランスポート・レイヤと呼んでいます。

ATA/ATAPI-4以降では、ATAの仕様はこのしくみを取り込んだため、別の規格であることを意識する必要はありません。

● ATAPIのコマンド発行のようす

まず、ATAとATAPIのコマンドの概念図を図1に示します。ATAPIパケット・コマンドはATAコマンドの上に位置しており、ATAコマンドを使って

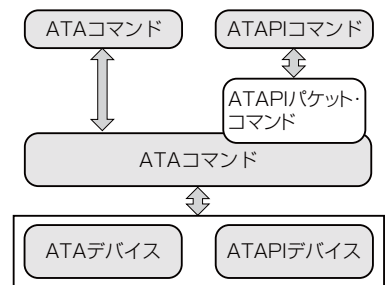


図1
ATAとATAPI
のコマンドの
概念図

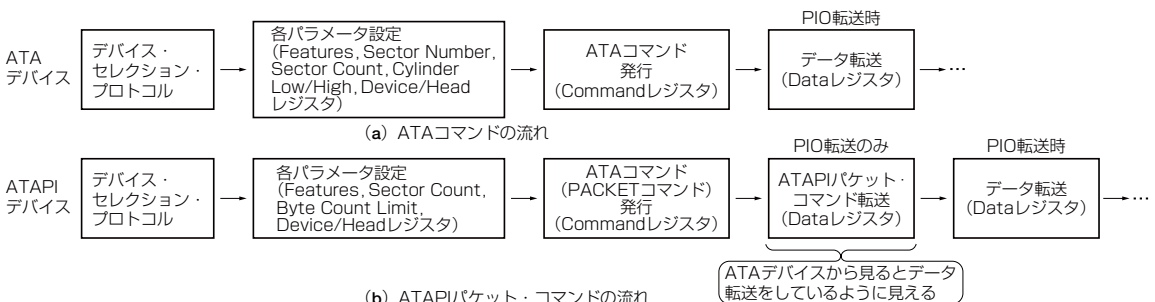


図2 ATAコマンド・プロトコルとATAPIコマンド・プロトコル

ATA/ATAPI デバイス 制御 BIOS の基礎

熊谷 あき

第1章から第4章までは ATA や ATAPI の規格の概要について解説した。この章では ATA や ATAPI デバイスを制御する最も基本的なプログラムを解説する。まず ATA コマンド・プロトコルを処理するための基本サブルーチン群を作成し、その上にいくつかの ATA コマンドの発行ルーチンを実装する。こうして HDD と CD-ROM をアクセスする最も簡単なプログラムを示す。プログラムは異なるプラットフォームにも移植しやすいように記述する。
(編集部)

ここまでで、ATA/ATAPI に関する基本的な部分をひとつおき解説してきました。そこでいよいよ ATA/ATAPI デバイスを制御するための最も簡単な、PIO 転送方式をサポートしたプログラムについて解説します。いわゆる ATA/ATAPI 制御用の簡単な BIOS を作成してみましょう。

1 ATA/ATAPI デバイス制御 BIOS の基本構造

第6章では、PC/AT 互換機 + MS-DOS という環境で実際にハード・ディスク・ドライブ (HDD) や CD-ROM を読み書きします。また、第10章や第11章では、SH-2 や V850、ARM マイコンといった組み込み機器へ ATA インターフェースを組み込むことを想定しています。制御プログラムは C 言語で記述しますが、できるかぎり移植性を考慮して、コンパイラ依存部や動作環境依存部分を切り離れた構造にします。

● int 型サイズ

SH シリーズや V850、ARM マイコンの int 型のサイズは 32 ビットです。しかし、MS-DOS の環境は 16 ビットです。そこでビット・サイズに依存したり、共通でなければ困る変数などには、typedef で型を定義し直してそれを使うようにします。ここでは次のように定義しました。

- UDWORD 32 ビット

```
AT ... unsigned long
```

```
SH ... unsigned int
```

- UWORD 16 ビット

```
AT ... unsigned int
```

```
SH ... unsigned short int
```

- UBYTE 8 ビット

```
AT ... unsigned char
```

```
SH ... unsigned char
```

以降では、明確にサイズを決めたい場合にはこれらの型を使います。数種類の状態保持や成功/失敗といったフラグなど、符号付き 16 ビット・サイズの変数で十分に表現できる部分には、単に“int”を使っています。

● I/O 入出力

x86 系アーキテクチャには I/O 空間があり、PC/AT 互換機では I/O 空間に ATA レジスタが存在しています。しかし、SH シリーズには I/O 空間という概念はなく、メモリ・マップト I/O となります。ATA レジスタに直接アクセスする部分は、ハードウェア依存部に切り分けて実装します。ここでは、I/O 入出力を次のように別関数またはマクロとしました。

- 8 ビット入力

```
in_byte(address)
```

- 16 ビット入力

```
in_word(address)
```

- 8 ビット出力

```
out_byte(address, data)
```

- 16 ビット出力

```
out_word(address, data)
```

● 割り込み処理

さらに、ATAPI 制御では割り込みを使った制御も必要になります。ATA/ATAPI 制御用 BIOS としては、ATA 割り込みが発生した場合は特定の関数が実行されることを前提にプログラムを組みます。

ATA 割り込みが発生してから実際にその関数が呼

ATA/ATAPI デバイス 制御プログラムの応用

熊谷 あき

前章では ATA/ATAPI デバイス制御用の簡単な BIOS を作成し、最も基本的な HDD のセクタ・リード/ライトや CD-ROM ドライブのセクタ・リード方法を解説した。ATA/ATAPI にはこれら HDD や CD-ROM ドライブ以外にもさまざまなデバイスが存在する。ここでは書き込み可能な ATAPI デバイスとして 3.5 インチ MO ドライブや SuperDisk を、リムーバブルな ATA デバイスとして ZIP ディスクを取り上げ、これらのデバイスの制御方法について、ATA/ATAPI アナライザなども使って詳しく解説する。
(編集部)

IDE は PC/AT 互換機に標準に用意されているストレージ用ポートであるため、このポートに接続可能なデバイスにはいろいろなものが用意されています。前章で解説したハード・ディスク・ドライブ (HDD) や CD-ROM ドライブは最も一般的なデバイスですが、ほかにポピュラなデバイスとしては、プロローグの写真にも示した 3.5 インチ光磁気ディスク・ドライブや、大容量フロッピーディスク・ドライブ (FDD) などが存在します。これらのデバイスも、第3章で解説した ATA コマンドや第4章で解説した ATAPI パケット・コマンドで制御可能です。

ここでは、Appendix 3 で紹介している ATA/ATAPI アナライザを使用して、具体的にどのようなコマンドでこれらのデバイスが制御されているかを簡単に解析してみます。そしてそれを参考にして、第5章で作成した基本制御プログラムを組み合わせて、実際にデバイスを制御してみます。

なお、使用した ATA/ATAPI アナライザについての紹介は、Appendix 3 を参照してください。

1 ATA/ATAPI アナライザ による動作解析

第5章で説明した CD-ROM のセクタ・リード・プログラムは、CD-ROM が正常にアクセスできることを前提にしたプログラムでした。リムーバブル媒体では、ドライブにメディアが挿入されていない可能性もあるので、ノット・レディ状態を考慮したプログラムが必要です。CD-ROM ではどのようにしてノット・レディ状態を検出したらよいのでしょうか。

また、HDD や CD-ROM 以外のデバイスは、具体的

にどのようにして制御するのでしょうか。ここでは、1.3G バイト対応 3.5 インチ光磁気ディスク・ドライブ、大容量 FDD として 120M バイト対応版 Super DISK と 100M バイト版 ZIP ドライブの3種類のデバイスを取り上げ、これらのデバイスを Windows 環境に接続し、ATA/ATAPI アナライザを使ってその動作を簡単に調べてみます。

1.1 CD-ROM ドライブ

● リムーバブル・メディアとしての制御

CD-ROM のセクタ・リード方法は、すでに第5章の基本プログラムで解説しました。しかし、第5章のプログラムは最も基本的なものです。

CD-ROM ドライブはディスクを交換可能なリムーバブル・デバイスなので、ディスクが入っていない場合にアクセスした際のエラー処理など、実際の運用ではさまざまな事態が考えられます。安定したドライブを実現するには、これらのさまざまな状況に対して適切にエラー/リトライ処理を考え、プログラムを作成しなければなりません。

● ノット・レディ状態

ここでは Windows がどのようにして CD-ROM ドライブを制御しているか、ATA/ATAPI アナライザを使って調べてみます。まず、ドライブにディスクを入れずに Windows を起動した後、ディスクを入れてフォルダが開くまでの動作を調べたのがリスト 1 です。使用したアナライザは IDE のすべての動作をキャプチャするもので、実際にはもっと膨大なデータになります。ここでは注目すべき点を切り出して、CD-ROM ドライブを制御するために必要なコマンド

ATA/ATAPI アナライザの活用

熊谷 あき

本書の執筆に当たり、ATA/ATAPI コマンドを詳しく解析する必要が出てきました。幸いなことに、ATA/ATAPI アナライザを借用できたので、ここでその借用した ATA/ATAPI アナライザについて紹介します。

今回使用させていただいた ATA/ATAPI アナライザは、IDE-Pocket (東陽テクニカ) という、小型軽量のハンディ・タイプのアナライザです。写真 A に外観を、表 A に仕様を示します。

写真からわかるように、本体は 3.5 インチ・ハード・ディスク・ドライブ (HDD) 程度の大きさで非常にコンパクトです。これ自体にはユーザ・インターフェース機能がないので、IDE-Pocket 自身を制御するには別途パソコンが必要です。そのパソコンとの接続には USB を使うので、ノート・パソコンと組み合わせれば、フィールドでのテスト/デバッグに威力を発揮しそうです。

● アナライザの接続

一般的に ATA/ATAPI アナライザを使う場合、マザーボードとマスタ・デバイス 1 台を IDE ケーブルの両端につなぎます。そして、真ん中のコネクタにア

ナライザを接続します [図 A (a)]。

借用したアナライザには、マザーボードやマスタ・デバイス、スレーブ・デバイス、そしてアナライザ自身を接続するために、IDE ケーブルへコネクタが四つ圧着された特別仕様のケーブルも添付されていました。たしかに、マスタとスレーブの 2 台のデバイスを接続した環境でさらにアナライザも使いたいとなれば、このようなケーブルが必要になります [図 A (b)]。しかし、添付の取り扱い説明書には、ATA の電気的規格を満たしていないため、このケーブルについては動作を保証しないと書かれています。このあたりは注意して使用する必要があるようです。

IDE-Pocket とホスト・パソコンの間は、一般的な USB 周辺機器の接続と同じです。IDE-Pocket に AC アダプタが添付されていることからわかるように、セルフ・パワー・デバイスなので、バス・パワーで動作している USB ハブにも接続できます。

筆者のノート・パソコンにある USB ポートは一つだけで、普段は USB マウスを接続しています。そこで USB のポートを二つにする、コネクタ部が少しゴツくて短いケーブルの付いた、二また USB ケーブル



写真 A
IDE-Pocket (東陽テクニカ)

UltraDMA 転送 制御プログラムの基礎

桑野 雅彦

UltraDMA/100 対応のマザーボードと HDD を組み合わせれば、瞬間最大データ転送レートが 100M バイト/秒というシステムを構築できる。UltraDMA 転送は PCI バス時代になったからこそ実現できた転送方式であるといえる。ここでは UltraDMA 転送の制御を、PC/AT 互換機上で実際にプログラムを作成しながら解説する。UltraDMA 転送に対応したマザーボードと HDD を用意してほしい。
(編集部)

少し前までは、ハード・ディスク・ドライブ(HDD)内にあるバッファ・メモリとの転送では UltraDMA/100 の性能が出るものの、磁気記録面からの読み書き転送レートが UltraDMA/100 までいかないものがほとんどでした。しかし、プラッタの記録密度の向上や回転数の高速化で、最近では真に UltraDMA/100 の転送レートを必要とする HDD も出てきました。このような HDD を UltraDMA/100 を下回る環境で使用した場合、ボトルネックは完全にバス側に出てきます。

高速なデータ転送を実現する UltraDMA 転送はどのようにして制御するか、ここでは実際にプログラムを作成して UltraDMA 転送を体感してください。

1 PC/AT 互換機における IDE のデータ転送

すでに第2章などでも簡単に解説していますが、ここでもう一度 PC/AT 互換機における、IDE のデータ転送について説明します。図1に PC/AT 互換機における IDE のデータ転送のようすを示します。

● PIO 転送のようす

図1(a)は PIO 転送のようすです。まず CPU が ATA のデータ・レジスタに対して I/O リードを行います。CPU のアドレス・バスには、ATA のデータ・レジスタのアドレスが出力されます。すると IDE の信号線上では、データ・レジスタのアドレスが出力され、DIOR# 信号が“L”レベルになり、データの読み出しが行われます。読み出したデータは CPU のデータ・バスに出力され、CPU はそれを読み取ります。次にその読み込んだデータをメモリに書き込むために、今度はアドレス・バスにメモリのアドレスを、そ

してデータ・バスには先ほど読み出したデータを出力し、メモリへの書き込み信号(図では $\overline{\text{MEMW}}$)をアクティブにします。これで1ワードのデータ転送が完了します。

つまり、PIO 転送で1ワードのデータを転送するには、CPU は I/O アクセスとメモリ・アクセスの2回のバス・アクセスを発生させなければならないわけです。もちろん、セクタ・データを読み捨てるとか、セクタ内を0で埋めるという場合は、転送先/元のメモリ・アクセスを発生させなくても可能ですが、一般的なデータ転送では必ずメモリへのアクセスも必要になります。

なお、図ではシステム・バスを ISA バスとしていますが、PCI バスになっても基本は変わりません。

● マルチワード DMA 転送のようす

図1(b)はマルチワード DMA 転送のようすです。HDD 側のデータ転送の準備ができると、DMARQ 信号を“H”レベルにして DMA 転送要求を出力します。これがシステム内の DMA リクエスト信号(図では DRQ_n)に伝わり、PC/AT アーキテクチャの DMA コントローラである 8237A に入力されます。

すると 8237A は DMA 要求を受け付け、システム内の DMA アクノリッジ信号(図では $\overline{\text{DACK}_n}$)をアクティブにします。これが IDE 信号線上では DMACK# となって出力され、HDD 側とシステム側双方の DMA 転送の用意ができました。

実際のデータ転送時は、8237A は IOR# を出力して I/O リード要求を出力すると同時に、アドレス・バスにはメモリ・アドレスを、そしてメモリ書き込み信号である $\overline{\text{MEMW}}$ を出力します。アドレス・バスにメモ

PIO 転送対応 PCI ボードの 設計/製作

山武 一朗

本章では PIO 転送対応の ATA インターフェースを PCI バス上に設計/製作する。一般的に PCI バスを搭載したプラットフォームには IDE コントローラも搭載されているのが普通だが、ここでは ATA インターフェース・ホスト機能実現のための学習用として、CPLD/FPGA などのプログラマブル・デバイスを搭載した PCI 評価用ボードを使い、最も基本的な転送モードである PIO 転送に対応した ATA ホスト・コントローラの設計と、制御ソフトウェアの作成までを解説する。

(編集部)

● 作らなくても IDE はあるじゃん!?

PC/AT 互換機や G3/G4 Power Mac に代表されるように、PCI バスを搭載したシステムには、ほとんどすべてに IDE コントローラも搭載されています。x86 系 CPU や PowerPC 系 CPU を使う場合は、メイン・メモリ・コントローラやパソコン周辺入出力インターフェース機能を集積した、いわゆるチップ・セットと呼ばれる LSI とセットにしてシステムを設計します。このチップ・セットに、PCI バス・コントローラや IDE コントローラも内蔵されているのが普通だからです(図1)。

よって、PCI バス上に FPGA や CPLD などのプログラマブル・デバイスを使って IDE コントローラを実装する場面や必要性は、実はあまりありません。

しかし組み込み分野で、しかも非 AT アーキテクチャを採用するシステムを考えた場合、これが当てはまらない場合も出てきます。PC/AT 互換機向けのチップ・セット、いわゆる SuperI/O と呼ばれる LSI は、AT アーキテクチャに依存した構造のため、x86 系以外の CPU では不要だったり使えない機能も数多く実装されています。また、メーカー間の激しい競争のため世代交代が激しく、組み込み分野で要求される5年や10年といった長期間にわたる部品の安定供給が保証されなかったり(数年後にはもうディスコン!?)、少数ロットでは部品を購入することができないという話も聞きます。

● ATA インターフェースの基本を学習する

そういう意味では、x86 系以外の CPU と PCI バスを搭載したシステムに、IDE コントローラだけを追加したい。しかも、FPGA や CPLD などプログラマブル・デバイスを使って組み込みたい…という要求

は、少なからずあると思われます。また、ATA インターフェースのハードウェアを勉強するためにも、実際にインターフェースを設計/製作してみるというのは重要なことです。

ここでは、ハード・ディスク・ドライブ(HDD)や CD-ROM を制御するのに最も基本的な PIO 転送に対応した ATA インターフェースを、FPGA を使って PCI バス上に実装し、簡単な制御プログラムまでを作

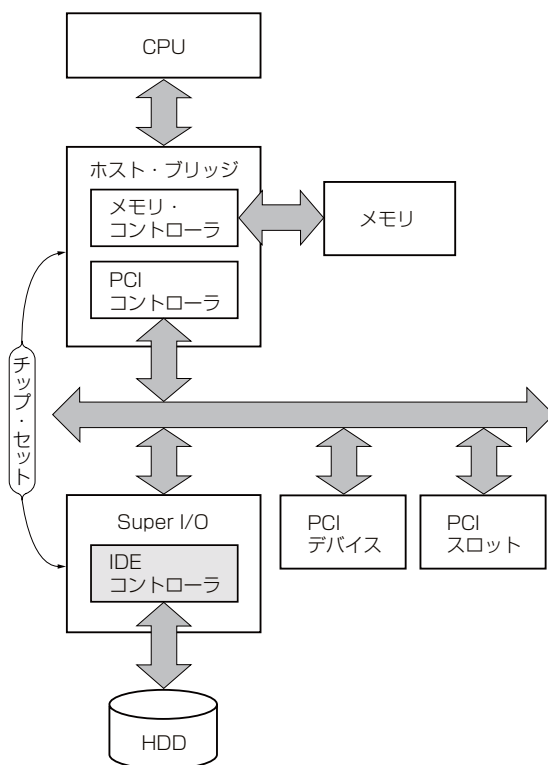


図1 PC/AT 互換機の PCI バスと IDE

UltraDMA 転送対応 PCI ボードの設計/製作

菅原 博英

前章の第8章で設計したホスト・コントローラはPIO転送のみに対応している。ここではパラレル転送方式でより転送レートを高めたUltraDMA転送に対応したホスト・コントローラを、FPGAを使って実現する。

なお、本コントローラ的设计およびテスト・プログラムについては参考文献(1)も参照してほしい。(編集部)

最近はバスのシリアル化というトレンドの中で、シリアルATAが普及しています。今後もこのトレンドは続くでしょう。一方で、パラレルATA(IDE)規格も、今後しばらくは一定の需要があると思います。

これまで、パラレルATAのPIOアクセスまでの設計を扱った書籍はよく見受けられました。しかし、UltraDMAを実際に設計し、動作の確認を行う書籍はあまりありませんでした。そこで本章では、UltraDMAを含むパラレルATAのPCIホスト・バス・アダプタの設計を行います。

1 パラレルATA全体の構成と仕様

設計する、パラレルATAコア(PATA_CORE)の全体構成図を図1に示します。

PCIバスとの接続は、参考文献(1)で詳しく解説されているSATA_COREと共通な内部コア・インターフェースにします。したがって、PCIバス側はSATA_COREの場合と全く同じRTLを使用できます。

なお、今回設計するコアはDMAとしてUltraDMAのみサポートし、MultiWordDMAなどはサポートしません。また、UltraDMAはホスト/デバイスで全く異なる設計になるので、ここではホスト機能だけとします。

2 パラレルATAの規格の概要

● PIOの設計上の着目点

パラレルATAの規格そのものは第1章から第2章を参照してください。

ここでは最低限の説明のために、図2にPIOアクセスのタイミング波形、表1と表2にタイミング規格を示します。

PIOアクセスでの着目点は、まずIORDYの扱いです。規格では、DIOR#/DIOW#のアサートから t_A 時間はIORDYが不定(参照してはいけない)となっている点です。ですから、ホストのシーケンサは t_A 時間以上待ってからIORDYを参照し、次のシーケンスに移行すればよいわけです。

もう一つの着目点は、リード時にホスト側がHi-zとする期間です。開始はDIOR#のアサートからHi-zとし、DIOR#のネゲートから t_{HZ} 間はHi-z状態を保つということです。

あとは、単なるアドレス・デコーダとシーケンサです。

● UltraDMAの設計上の着目点

UltraDMAのリードの波形とタイミング規格を、図3と表3に示します。

多くの解説では、これは最初に説明される最も典型的な波形です。そして、DSTROBEのストロブを

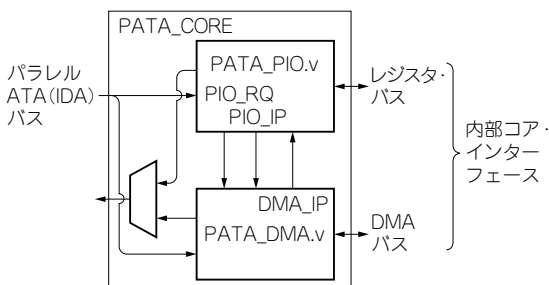


図1 PATA_CORE全体の構成

SH/V850/ARM マイコンに IDE モードで CompactFlash を接続する

山武 一朗

本章では SH や V850, ARM などの各種マイコンに IDE デバイスを接続する事例について解説する。実際に接続するデバイスとして CompactFlash カードを使う。CompactFlash カードを TrueIDE モードで使うと、IDE デバイスが接続されたものとして扱うことができる。 (編集部)

ここでは SH や ARM などのマイコンに、IDE デバイスを接続する方法について解説します。とはいえ、せっかく低消費電力なマイコンへモータを回転させる必要のあるハード・ディスク・ドライブを接続するのはどうかとも思います。そこでモータなどの稼働部品がなく低消費電力でありながら IDE デバイスとして使える記憶媒体として、ここでは CompactFlash カードを使います。CompactFlash カードは TrueIDE モードを備えているので、IDE として制御することができます。

表 1 に CompactFlash カードのピン配置を、表 2 に CompactFlash カードと IDE の信号ピンの対応を示します。

1 マイコンと IDE デバイスとの 接続方法の考察

いきなり SH などのマイコンへ CompactFlash カードを接続する前に、まずは IDE デバイスをマイコンに接続するための条件や回路構成などについて考えてみましょう。

● IDE は 8/16 ビット混在バス

IDE のデータ・バスは信号線が DD0 ~ DD15 まであることから明らかなように、16 ビット幅です。しかし第 1 章 第 2 節の信号線の説明にもあるように、16 本のデータ・バスを使うのはデータ転送時のみで、データ・レジスタ以外の制御レジスタのアクセスは下位 8 ビットのみを使います。

これは一見、気を付ける必要もないほど当たり前の仕様のように思えますが、後述するように、いざマイコンに IDE デバイスを接続しようとしたときに問題

となる重要な仕様の一つです。

● IDE はバス・アクセスそのもの

また第 1 章の PIO 転送モードのアクセス・タイミングを見てもわかるように、IDE デバイスに対するアクセスは、一般的な CPU のいわゆるローカル・バスのアクセスに相当します。ローカル・バスにはほかにもメモリや周辺 LSI が接続されることもあるので、IDE の CS0# や CS1# の生成にはアドレス・デコーダが必要な場合もあるでしょう。

また PIO 転送モードにはモード 0 ~ 4 まで五つの動作モードが規定されていますが、基本的な動作に変わりはなく、モード 0 が低速でモード 4 が高速と、アクセス・タイムが早くなっているだけです。

リード/ライト信号が IORD# や IOWR# となっていますが、これは PC/AT 互換機では IDE が I/O 空間に接続されていたためであり、メモリ用のリード信号やライト信号を使っても全く問題ありません。

以上のことから、データ・バスの幅として 16 ビットの外部バスを持つマイコンであれば、IDE デバイスの接続は簡単です。ただし後述するような、少しややこしい問題点があります。

● データ・レジスタとデータ・レジスタ以外の制御レジスタ

IDE の場合、CS0# で選択される空間はコマンド・ブロック・レジスタ空間、CS1# で選択されるのがコントロール・ブロック・レジスタ空間と呼ばれます。16 ビット幅でアクセスが必要なのはデータ・レジスタのみで、データ・レジスタはコマンド・ブロック・レジスタ空間に割り当てられています。コマンド・ブロック・レジスタ空間にはほかにも制御レジスタがあ

V850 マイコン制御の 液晶表示付き CD プレーヤの製作

館 伸幸

本書の最後に、マイコンに ATAPI インターフェースを実装する事例について解説する。ここではパソコン用 CD-ROM ドライブをマイコンから制御し、CD プレーヤを実現している。マイコンの外部バスにバッファを経由して CD-ROM ドライブを接続している。また第5章や第6章で作成した ATAPI BIOS プログラムをマイコンに移植した。

(編集部)

パソコンを自作していると、いつのまにか押し入れの中に、不要になった CD-ROM ドライブや DVD-ROM ドライブがたまっていたりします。世の中の主流は書き込み可能なドライブなので、単なる CD-ROM ドライブなどは中古なら 1,000 円以下で売られていることもあるようです。本章ではこれらを有効活用し、CD プレーヤを作ってみたいと思います。

昨今ではオーディオ CD や DVD はもちろんのこと、あらゆるフォーマットのディスクを再生できるマルチプレーヤが 5,000 円程度で買えてしまいます。しかし、自分で作ったプレーヤで音楽を聴くというのもよいものです。プログラムを変更すれば、自分好みの機能にアレンジできます。ドライブそのものを補強するなどして音質にこだわった、世界に1台の“マイ”プレーヤに仕上げることも可能です。

1 ATAPI インターフェースのハードウェア

● まずは回路から

図1は今回製作する CD プレーヤの構成です。

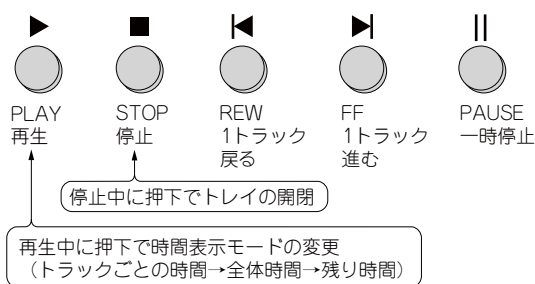


図2 操作スイッチ

一般の CD プレーヤと同じ機能を持たせる。PLAY キーによる表示モード変更は、ソフトウェア設計中に思い付いたもの。

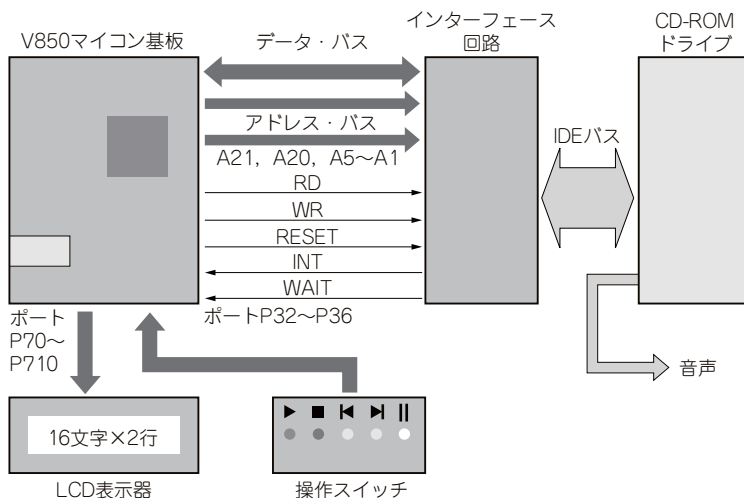


図1 全体のブロック図
LCD 表示と操作スイッチはポートに。CD-ROM ドライブは外部バス(メモリ空間)に配置する。