

## 1-1 小さなコンピュータ「マイコン」の役割をさぐる

### ● マイコンは「小さなコンピュータ」

図1はパソコンの構成例です。パソコンはCPUだけではなく、そのほかにCPUを補助するLSI、メモリ、ハードディスク、電源、キーボード、ディスプレイとさまざまなものから構成されています。CPUはプログラムを読みながら計算や処理を行うもので、パソコンの頭脳と言えます。

図2は一般的なマイコンのチップの内部を模式的に示したものです。図1と比較してみると、構造がよく似ているのわかります。

マイコンは「マイクロ・コンピュータ」や「マイクロ・コントローラ」の略称で、CPUの周りにある部品を、できるかぎり一つのLSIに集積したものです。写真1に実際のマイコンの外観例を示します。

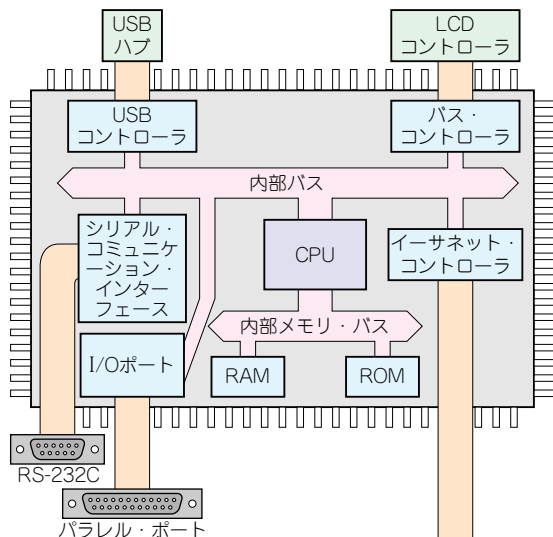


図2 一般的なマイコンのチップの内部  
図1のパソコンと構造がよく似ている

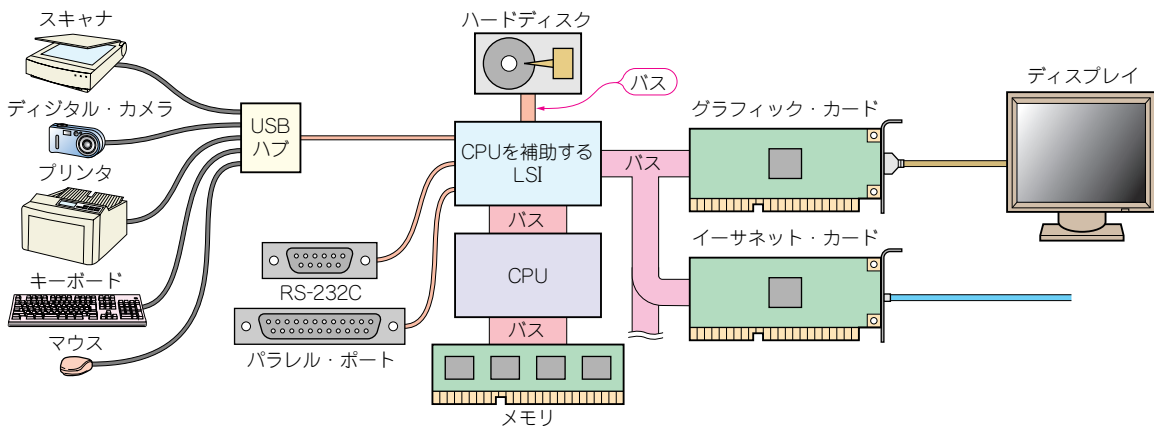


図1 パソコンを構成する部品の例  
CPUのほか、メモリ、ハードディスクなどさまざまなものから構成されている

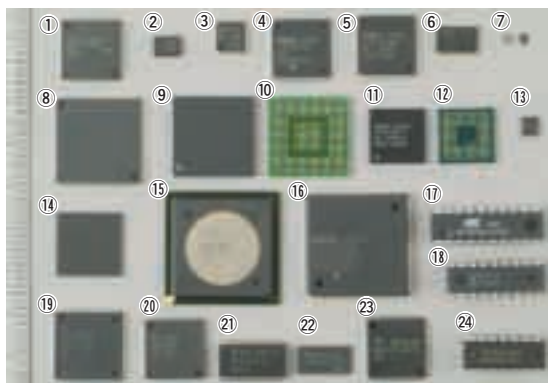


写真1  
実際のマイコンの外観例  
①⑧ H8ファミリー、②③ R8Cファミリー、⑨⑩ SHファミリー(ルネサステクノロジー)  
④⑥⑦⑬ 78Kファミリー、⑤⑪⑫⑬ V850ファミリー(NEC)  
⑭⑮⑲ FRファミリー、⑳ F2MCファミリー(富士通)  
⑰ AVRマイコン(アトメル)  
⑱ PICマイコン(マイクロチップ・テクノロジー)  
㉑～㉒ MSP430ファミリー(テキサス・インスツルメンツ)

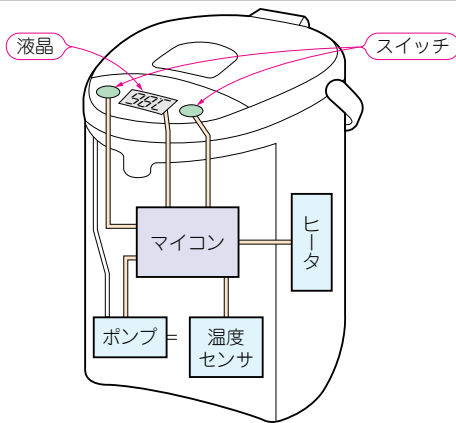


図3 電子ポットの中のマイコンは装置どうしをつなぐハブの役割をしている  
お湯の温度をセンサで検出し、常に適温に保つ

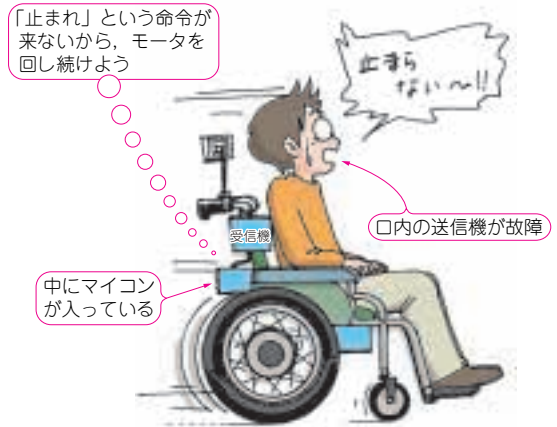


図4 送信機が壊れたときを想定してマイコンの制御内容をプログラムしないと車いすが暴走してしまう  
プロトコルの修正で解決できる

### ● さまざまな装置を連携動作させる役割を担う

パソコンにはたくさんのコネクタが付いていて、いろいろな装置を繋いで動かせます。例えば、キーボードで入力した文字をディスプレイに映し、スキャナで撮った画像をプリンタで印刷できます。

つまり、パソコンはさまざまな周辺装置同士を互いに接続し、データを橋渡ししていると言えます。このとき、データを適切な形に加工したりタイミングを合わせたりといった頭脳プレーも行います。

装置同士を繋ぐ装置を「ハブ(hub)」と言います。ハブとは、英語で「車輪の中心部分」という意味です。マイコン、すなわち「小さなコンピュータ」が電子回路の中で果たしている役割もいわば「ハブ」なのです。

#### ▶ 電子ポットの中のマイコンの役割も「ハブ」

図3は、電子ポットの中に入っているマイコンが、スイッチや液晶などの装置同士を繋ぐ「ハブ」の役割をしている様子を示しています。例えばスイッチによって「保温」モードが選ばれると、マイコンは温度センサでお湯の温度を監視しながらヒータを制御し、常に適切な温度にお湯が保たれるようにします。

#### ▶ ワイヤレス車いすでのマイコンの役割も「ハブ」

イントロダクションで紹介した「ワイヤレス車いす」の中のマイコンが果たしている役割は、やはり「ハブ」です。無線機、LED、モータ、バッテリーなどの性質の違うさまざまな回路や装置も、間にマイコンを入れることでうまく接続できます。

まず無線の受信機から受け取った信号を解析し、受信データにエラーが含まれていないか、近くにいる別の車いすへの信号ではないかなどをチェックし、受け取った信号に含まれる命令に沿ってモータを制御し、車いすを動かします。バッテリーの容量などのステータスをLEDに表示したりもします。

### ● 処理の手順を理解して動作するのが得意

ハブとして動作するマイコンは、装置同士を単に繋ぐだけではなく、データの加工やタイミングの調整、データをチェックしてエラーを検出、といった知的な動作ができます。このような知的な処理はマイコンの得意分野と言ってよいでしょう。

例えば、ワイヤレス車いすで操作を「前に進む」「止まる」の二つとしましょう。マイコンは、次の内容が無線機とモータとの間で橋渡しすることになります。「無線機から受信した信号を解釈し、前に進むという命令だった場合はモータに電流を流し、止まるという命令であれば電流を止める」

このように「何が来たら何をやる」という決められた手順の事を「プロトコル」と言います。マイコンはこういったプロトコルを理解して処理するのが得意な部品です。

### ● 処理の内容を簡単に変更できる

プロトコルを少し変更する場合、マイコンを使っていればプログラムを少し変更するだけで対応できます。回路を作り直す必要が無く、非常に便利です。

前述のプロトコルは単純すぎて、実際に使うには問題があります。例えば「前に進む」という命令を送った後に送信機が壊れてしまったら、図4のように車いすが止まらなくなってしまいます。

プロトコルを少し修正し、「1秒に1回以上前に進む命令が来たときはモータに電流を流し、そうでないときは電流を止める」としておけば、送信機が壊れたときに止まらなくなってしまいう問題が解決します。