

第2章 入出力ポートの設定と 内部ハードウェアの動作を理解する

マイコンから High/Low の 信号を出力してみよう!

山本 秀樹
Hideki Yamamoto

この章では、第1章で作成したプログラムを実行すると、マイコンの内部ハードウェアがどのように動作するのかを見ていきます。特に、**マイコンから信号が出力されるまで**を見てみます。

その前に、プログラミング言語について簡単に説明しましょう。

プログラミング言語のいろいろ

マイコンはプログラムに従って動作します。プログラムを記述する言語は、**プログラミング言語**と呼ばれます。

プログラミング言語として、マイコンが理解できる機械語のほかに、人間にとって理解しやすい言語が使われています。

● マイコンが理解できる機械語(マシン語)

付録マイコンを含む大半のコンピュータでは、“H”、“L”の2値の電気信号により情報のやりとりを行っています。この電気信号に‘1’と‘0’を対応づけることで、情報を**2進数**で表現することができます。

この2進数の1桁は、**1ビット**と呼ばれます。1ビットでは $2^1=2$ 通り、2ビットでは $2^2=4$ 通り、8ビットでは $2^8=256$ 通りの値を表現できます。**8ビットは1バイト**と呼ばれます。

マイコンは、**機械語**というプログラミング言語で記述されたプログラムを、直接解釈して実行できます。機械語のプログラムは2進数で表現され、実行時には対応する電気信号として扱われます。機械語は、**マシン語**とも呼ばれます。

機械語はマイコンの種類ごとに異なります。

● **機械語を読みやすくしたアセンブリ言語(アセンブラ)**
機械語は人間が理解するのは難しいため、**機械語とほぼ1対1に対応し、人間にとってもう少しわかりやすい、アセンブリ言語**が用意されています。

なお、アセンブリ言語という名前は正確な言いかたですが普通はあまり使われず、**アセンブラ**と呼ぶことが多いと思います(筆者もふだんはそう呼んでいる)。ただそうすると、処理系とプログラミング言語を区別できなくなるので、ここではアセンブリ言語と呼ぶことにします。

アセンブリ言語で記述されたプログラムは、

- 機械語に対応しマイコンにより実行される**命令**の部分
- アセンブラへの指示など直接は実行されない**疑似命令**

の部分からなります。疑似命令は、例えばC言語の #include、#define、#pragma などに似たものと考えてもよいでしょう。

図1からわかるように、アセンブリ言語で書かれたソース・コードから、アセンブラにより直接機械語プログラムに変換されるわけではありません。いったん**オブジェクト・モジュール**に変換され、次に**リンカ**によって、オブジェクト・モジュールから実行可能な機械語プログラムに変換されます。ソース・コードが複数ある場合は、アセンブラでそれぞれをオブジェクト・モジュールに変換し、リンカでそれらをまとめて一つの**実行可能プログラム**に変換します。

アセンブリ言語は機械語と対応する言語なので、基本的には似たようなものといえ、詳細はやはりマイコンの種類ごとに異なります。

Keywords

プログラミング言語, 2進数, 機械語, アセンブリ言語, アセンブラ, 命令, 疑似命令, オブジェクト・モジュール, リンカ, C言語, ポート1, 入出力ポート, リセット, 駆動能力, オペランド, ポートの方向, P1レジスタ, アドレス・バス, データ・バス, 制御バス, DRRレジスタ, MOV命令, オペランド, srcオペランド, destオペランド, ポート, BCLR命令, SFR

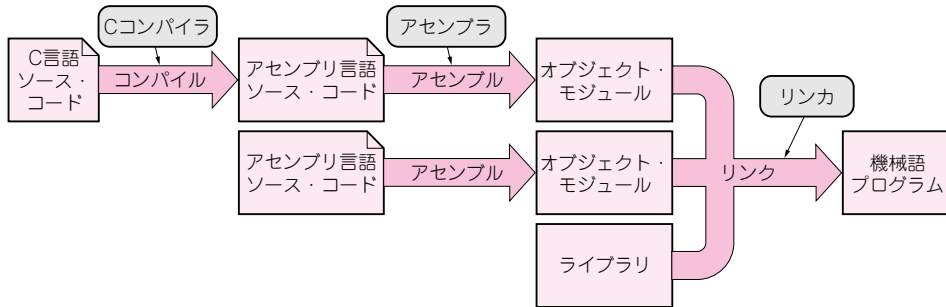


図1 アセンブリ言語のプログラム，C言語のプログラムが機械語プログラムへ変換されるまでの流れ

● マイコンの種類に依存しないC言語

マイコンのプログラム開発で、もっともよく使われているのがC言語でしょう。基本的にマイコンの種類に依存せず、大規模なアプリケーション・プログラムも、ハードウェア制御のようなきめ細かいプログラムも記述できる言語です。

C言語で記述されたプログラムも、図1の流れで変換(コンパイル)されます。一つのシステムのある部分をC言語で記述し、別の部分をアセンブリ言語で記述して、リンカで一つの実行可能プログラムにまとめることもできます。

この特集では、主にアセンブリ言語を使って解説しています。

出力端子を“L”にすればLEDは点灯する

では、第1章で動かしたプログラムの詳細を見てみましょう。

このプログラムをステップ実行したときにわかったように、最初は消灯していたLEDが、ある命令を実行すると点灯しました。命令を実行することで、マイコンの内部状態が変化し、それがLEDの点灯として現れています。

実験基板では、マイコンが“H”を出力しているとき[図2(a)]はLEDが消灯し、“L”を出力しているとき(b)はLEDが点灯します。プログラムで“H”、“L”のどちらを出力するのかにより、LEDの点灯・消灯を制御できます。

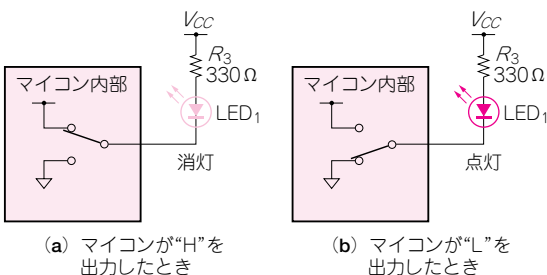


図2 LED点灯・消灯のイメージ

LED点灯プログラムの処理内容

● 第1章で作成したLED点灯プログラムの処理の流れ
プログラム内の個々の命令を順番に見ていく前に、第1章で作成したプログラムで行っている処理の概要を説明します。なお、このプログラムには INCLUDE や SECTION など、ピリオドで始まる命令があります。これらは疑似命令です。疑似命令の意味については、この章の最後でまとめて説明します。

このプログラムでは、リセット後にこのプログラムの先頭から付録マイコンが実行を始めるようにしています。

作成した実験基板では、LED₁をマイコンのP1_1端子(付録マイコンの22番ピン)、LED₂をP1_2端子(21番ピン)に接続しています。これらの端子に“L”を出力すると、接続しているそれぞれのLEDが点灯します。

この二つの端子は、付録マイコンのポート1に接続されています。P1_1はポート1のビット1、P1_2はビット2を表します。ポートと端子の関係のイメージを、図3に示します。

ポート1の端子は入出力端子として利用することができます。初期設定として、駆動能力の設定、出力する値の設定、ポートの方向を出力にする設定を行います。この段階では、LEDは消灯しています。

次にLEDを点灯するように、ポートに出力します。最後に、無限ループに入ってプログラムの実行を止め

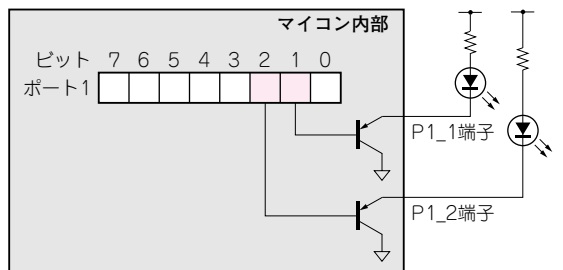


図3 ポートと端子の関係(イメージ)