

コンピュータ工学 講義プリント(11月6日)

今回は、アセンブリ言語のプログラムを作成し、そのプログラムを PIC マイコンに書き込むまでの大まかな手順について説明する。

・必要なソフトウェア・ハードウェア

以下にアセンブリ言語によって PIC マイコンのプログラムを開発するときに必要なソフトウェアとハードウェアを説明する。

(1) パソコン

パソコンはディスプレイ、キーボード、マウス、十分な容量のメモリなど、プログラムの開発に必要なソフトウェア(テキストエディタ、アセンブラなど)を動作させるために必要な機能・性能を持っているため、パソコンのプログラムを開発するには、パソコン自身を用いて開発するのが普通である。

この様に、プログラムを開発したコンピュータ自身で、開発したプログラムを利用する開発方式を、セルフ開発という。(図 1(a)参照)

一方で、PIC などのマイコンは、プログラムの開発に必要なソフトウェアを動作させるために必要な機能・性能を持っていない場合がほとんどである。この場合、マイコンのプログラムの開発にはパソコンを使う必要がある。この様に、プログラムを開発するコンピュータと、開発したプログラムを利用するコンピュータとが異なる開発方式を、クロス開発という。(図 1(b)参照)

PIC マイコンのプログラムの開発には、クロス開発のためのパソコンが必要である。

(2) テキストエディタ

アセンブリ言語や C 言語などでマイコンのプログラムの開発を行う場合、それらのプログラムを入力し、テキストファイルとして保存する必要がある。このような、テキストファイルの入力や保存のためのソフトウェアをテキストエディタ、または単にエディタと呼ぶ。

テキストエディタには、ワープロの様に文字のフォントの種類や色、大きさなどを文書の各所で変えたり、図を挿入したりといった機能はなく、テキスト(装飾のない文字のみで構成された文書)の入力に特化さ

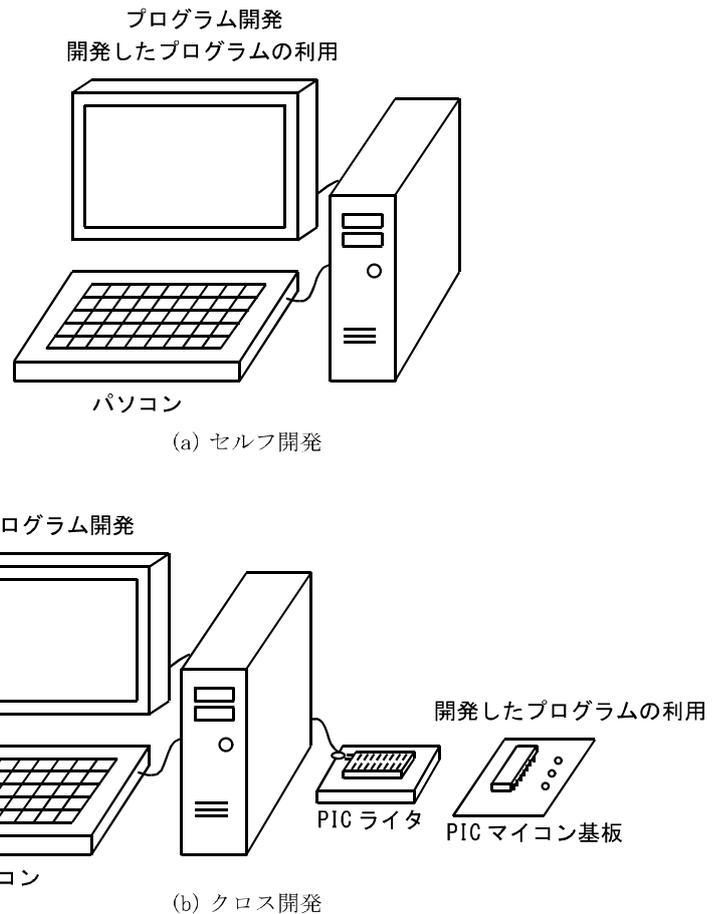


図 1、セルフ開発とクロス開発

れている。一方で、単語の検索などの機能は、ワープロに比べてテキストエディタの方が優れている場合が多い。

(3) アセンブラ

テキストエディタで入力したアセンブリ言語のプログラムのファイルは、アセンブラを使って機械語に変換する必要がある。

マイコンのプログラム開発では、クロス開発を行うため、アセンブラが動くコンピュータ(パソコン)と、生成した機械語が動作するコンピュータ(マイコン)が異なる。このようなアセンブラをクロスアセンブラと呼ぶ。

なお、C 言語などの高級言語でプログラムを開発する場合は、コンパイラにより機械語に変換する必要があるが、クロス開発で使うコンパイラをクロスコンパイラと呼ぶ。

(4) ライタ

アセンブラで生成した機械語のプログラムをマイコンで動作させるには、機械語のプログラムをマイコンに書き込む装置が必要となる。これをライタ(またはプログラマ)と呼ぶ。

ライタにはマイコンを挿入する IC ソケットが付いており(教科書 P.45 の図 2.40 を参照)、この IC ソケットにマイコンを装着した状態で、パソコンから機械語のプログラムを転送すると、マイコンにプログラムが書き込まれる仕組みになっている。プログラムを書き終わったマイコンは、ライタから取り外して、動作させたいマイコン基板の IC ソケットに差し替える。(図 1(b)参照)

なお、最近は基板の小型化のため、IC ソケットを使わずに、小型のマイコン IC を直接基板に半田付けする事が多くなった。(教科書 P.20 の図 2.5(a)の写真に写っている SOIC 型パッケージの PIC16F84A は、基板に直接半田付けするマイコン IC の一例である)この場合は、マイコンを基板から取り外せないで、その代わりにプログラムを転送するためのコネクタをマイコン基板に取り付ける。

このように、マイコンを基板から取り外さずに、プログラムを書き込む方法を、in-circuit programming と呼び、その中でも、プログラムの書き込みに必要な信号線を減らすために、シリアル方式でプログラムを転送する方式を In-Circuit Serial Programming (ICSP)と呼ぶ。ICSP でマイコンにプログラムを書き込むには、ICSP に対応した専用のライタが必要となる。

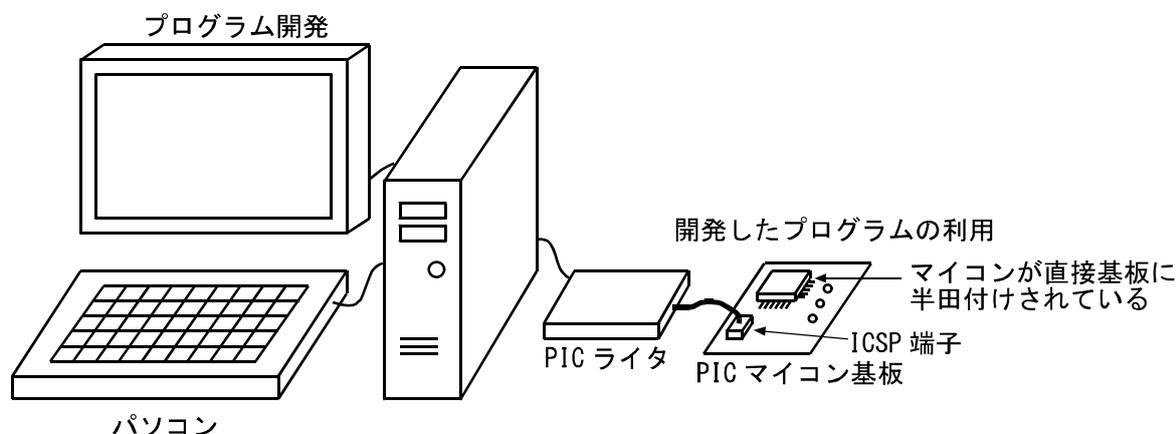


図 2、ICSP を用いたマイコンプログラムのクロス開発

(5) ライティングソフト

ライターでマイコンにプログラムを書き込むためには、パソコン上で、ライターにプログラムを転送するためのソフトウェアが動作している必要がある。このソフトウェアをライティングソフトと呼ぶ。ライティングソフトは、ライターを購入すると付属しているか、あるいはインターネットを利用して、無料でダウンロードできる場合が多い。

上記(1)~(5)のハードウェアやソフトウェアと、プログラムを書き込みたいマイコン、およびそのマイコンを実装する基板があれば、マイコンのプログラム開発が行える。

ただ、後に説明するように、テキストエディタ、アセンブラ、ライティングソフトは、頻繁に交互に切り替えながら使うので、これらが別々のソフトとして用意されていたのでは、パソコンの操作が煩雑になる。そこで、テキストエディタ、アセンブラ(およびコンパイラ)、ライティングソフトなどの機能を一つのプログラムに集約し、それぞれの機能が連携しながら使えるようにした、統合開発環境(IDE; Integrated Development Environment)と呼ばれるソフトウェアが用いられる事が多くなった。

PIC マイコンを製造開発している Microchip 社では、MPLAB X という IDE をインターネットで配布している。(図 3 参照) 機能制限が一部かかるものの、無料でも利用できるため、PIC マイコンのプログラム開発には MPLAB X を使うのが普通である。

余談になるが、MPLAB X の画面は全て英語表記で、ヘルプファイルも全て英語で書かれている。また、PIC マイコンのデータシート(PIC マイコンの仕様や性能、使い方などを詳しく書いた文書)も全て英語で

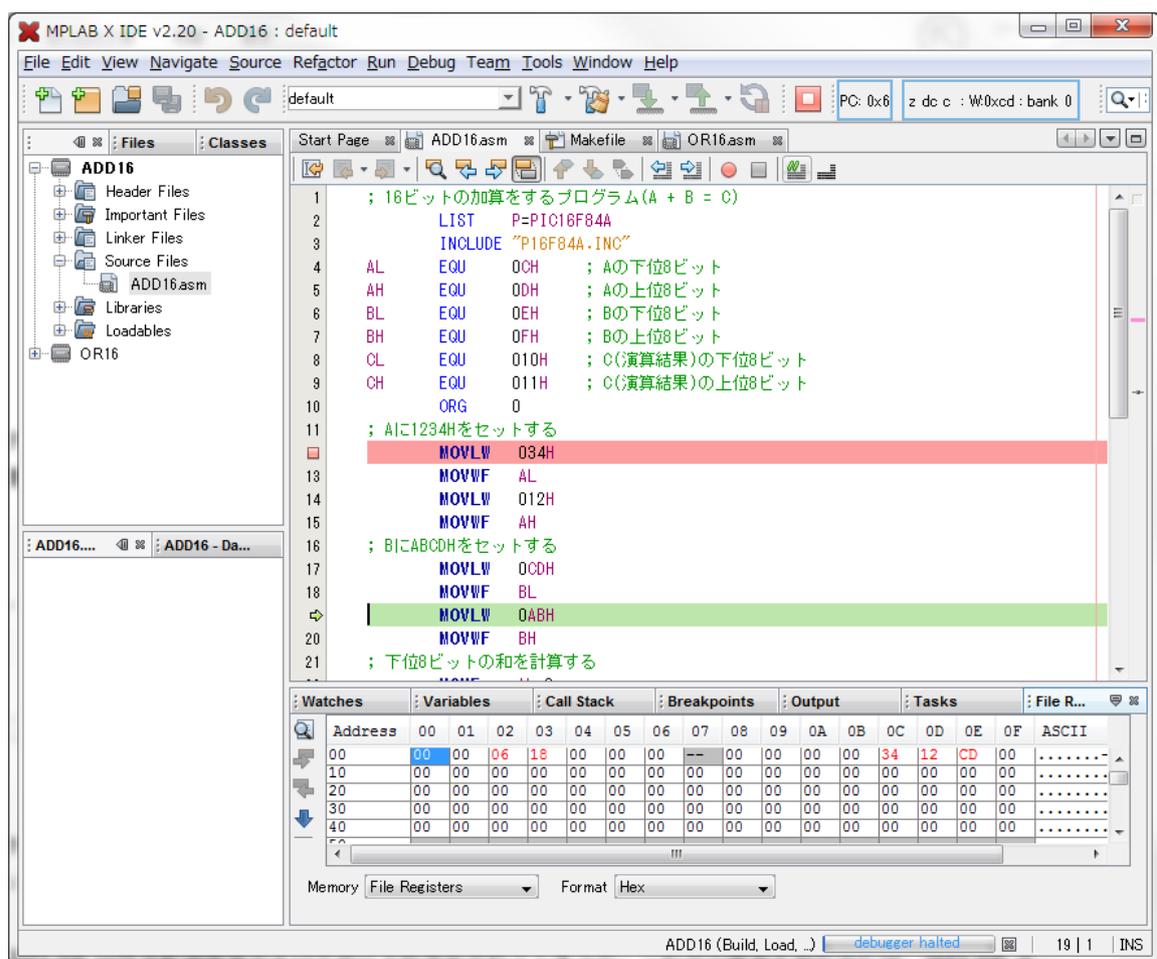


図 3、MPLAB X の画面

書かれている。

PIC マイコンに限らず、電子工学分野の開発を行う場合には、必要な情報が英語で提供されている場合が多いため、業務として開発を行うには、英語の文書を日本語並みに速く読める語学力が必須と考えてよい。

話が元に戻るが、教科書 P.153 以降では、MPLAB X の前身である MPLAB という IDE の使い方が説明されている。しかし、MPLAB は昨年新規開発およびサポートが中止されてしまった。これから開発を始めるには、MPLAB よりも MPLAB X を使う方が好ましい。

PIC マイコンにプログラムを書き込むライターには色々あるが、教科書 P.44~45 に 3 種類の PIC ライターが紹介されている。ただし、これらの商品は全て販売中止になっている。図 2.40 の PIC ライターキットに関しては、後継機種が秋月電子通商から販売されている。

ICSP 型の PIC ライターに関しては、Microchip 社純正の PICKit3 というライターが良く使われる。

図 4 に PICKit3 の写真を示す。左下の黒い 6 ピンのメスコネクタに、マイコンにプログラムを書き込むための信号が出てくる。マイコン基板の ICSP 端子(6 ピン、オス)に PICKit3 のコネクタを差し込んで、プログラムを書き込む。

また、PICKit3 に、DIP IC 用のソケットがついた基板を外付けすれば、旧来の、基板からマイコンを取り出してプログラムを書き込むタイプのライターにもなる。

・プログラムの作成を始めてからマイコンに書き込むまでの処理の流れ

プログラムの作成を始めてから、マイコンに書き込むまでの処理の流れは、おおむね教科書 P.42 の図 2.36 のフローチャートの様になる。プログラムの記述にはテキストエディタ、アセンブルにはアセンブラ、プログラムの書き込みにはライターを使うのは、既に説明したとおりである。

この図で、シミュレータやシミュレーションという言葉が出てくるので、説明しておく。

シミュレーションとは、実際の現象を、数式を解く事などにより、模擬的に再現する事であるが、ほとんどの場合、シミュレーションとはコンピュータを使ったコンピュータシミュレーションを指す。

マイコンの開発で行うシミュレーションは、マイコンの挙動を、パソコンの中で計算により再現する事を指す。シミュレータとは、そのシミュレーションに使うソフトウェアの事である。

MPLAB X(および MPLAB)は、主要な PIC マイコンのシミュレータを内蔵している。そのため、たとえマイコンの実機がなくても、作ったプログラムをシミュレータ上で動作させる事ができる。

実機のマイコンでプログラムを動作させた場合と比較すると、パソコン上で動作するシミュレータは動作が遅いというデメリットはあるものの、プログラムを任意のタイミングで止めたり、W レジスタやファイルレジスタの内容を画面に表示して観察できたり、動作に必要なクロック数を正確に数えたりといった事ができるため、実機でプログラムを動作させる前に、シミュレータで動作を確認する事の意味は大きい。

なお、図 3 は、前回の講義で説明した 16 ビットの足し算を行うプログラムを、MPLAB X のシミュレータで動作させ、MOVWF BL の命令を実行し終わったところ(MOVLW 0ABH の命令を実行する直前)で止めた画面をキャプチャしたものである。画面の右下にはファイルレジスタの内容が一覧表示されているが、



図 4、PICKit3

0CH~EH のアドレスに、期待したとおり、順に 34H、12H、CDH が書き込まれている事が確認できる。

この様に、CPU の内部の様子を観察しながら動作確認できるという点ではシミュレータはとても便利なツールであるが、シミュレーションできるのは、マイコン内部の出来事だけである。マイコンに接続したセンサなどの外部回路のシミュレーションまでは行えない。そのため、シミュレータで動作確認できるプログラムは、意外と限られている。先ほど説明した 16 ビットの足し算プログラムは、マイコンの内部だけで話が完結するので、最もうまくシミュレーションが機能する例の一つである。

なお、教科書の図 2.36 のフローチャートの「アセンブル」という条件判定で、「NG」という分岐条件があるが、これは、プログラムをアセンブルした結果、エラーが出たという事を示している。

当然であるが、例えば MOVWF 命令のつづりを NOVWF と間違えて入力すれば、アセンブラはそれをどう解釈していいのかわからず、エラーを出す。エラーが出ると、機械語のプログラムのファイルは生成されないの、エラーがなくなるようにプログラムを修正するまでは、次のステップに進めないのである。

また、「シミュレーション」という条件判定で「NG」という条件分岐があるが、これはシミュレータで実行した結果、プログラムの動作が期待と異なっていたという事を示している。

ただし、先ほど話したように、シミュレータでうまく動作確認できないようなプログラムも多く、また MPLAB X のシミュレータも、全ての PIC マイコンに対応している訳ではないので、シミュレーションを行うかどうかは任意である。

「実装」という条件判定に「NG」という条件分岐があるが、これは、プログラムを書き込んだマイコンを実際動作させたい基板に実装して確認したところ、基板の挙動が期待と異なっていたという事を示している。

以上のようにして、マイコンを実装した基板の挙動が、想定どおりになるまで、プログラムの修正を繰り返す。

・LED の点滅によるマイコン基板の動作確認(L チカ)

マイコンのプログラム開発を行う場合、マイコン基板が正常に動作しないのは、プログラムミスによるものとは限らない。自分でマイコン基板を設計した場合は、その設計ミスも原因となるし、アセンブラやコンパイラなどの開発ソフトの設定ミスや、マイコンにプログラムを書き込むライタの故障や配線ミスなども考えられる。

この様に、多種多様の異状の原因がある状況では、プログラム開発を進めるのは困難である。そのため、プログラム開発を始める時に、いきなり目的のプログラムを作り始めるのではなく、確実に基板が動作し、開発ソフトの設定や、ライタなどの配線が正しく行われていることを確認する作業をする事が多い。

パソコンと異なり、マイコンにはディスプレイなどの出力装置がないか、あるいはディスプレイがあっても、それが正常に動作する事が保障されるのは、ある程度開発が進んでからの事となる。そのため、画面に”Hello World!”などと表示する、いわゆる「printf デバッグ」という動作確認の手法は、使えない事が多い。

一方で、マイコン基板には、動作モードの表示などの目的で、LED ランプが設けられている事が多い。LED の点灯/消灯は、GPIO の出力を 1 または 0 に切り替えるだけなので、そのプログラムに誤りが混入する可能性は低い。

そのため、マイコン基板の初期の動作確認には、LED の点滅プログラムが使われる事が多い。LED が想定した周期で点滅すれば、マイコン基板全部とは言えなくとも、少なくともマイコン自体は機能しており、コンパイラの設定や、ライタの配線などにも誤りがない事が示される。この様な目的で LED の点滅プログ

ラムを作成・実行する事を、俗に「Lチカ」(LEDチカチカの略)と呼ぶ。

「printfデバッグ」や「Lチカ」は、俗語であるので、それらの言葉を使用する状況に注意が必要だが、技術者の間で広く使われている言葉なので、知っておく方が良いでしょう。

・おわりに

今回は、プログラム開発の大雑把な流れを説明した。具体的な開発ツール(MPLAB など)の使い方については、次々回の講義で説明する予定である。