

## 5 A - 1

## Linux における PIC ライタの開発

蛭原 康尊 アサノ デービッド

信州大学大学院工学系研究科情報工学専攻

## 1 研究の背景と目的

最近、PIC というものがいろいろな分野で使われるようになってきている。例えば、ゲームセンターなどにおいてある両替機や時計、赤外線リモコンなどである。そもそも PIC とは、Microchip Technology 社のマイクロコントローラ(制御用 IC)製品ファミリーの名称で小さな CPU である。PIC には 16F84A や 16F877 など様々な種類があるが主な特徴としては、プログラムの書き換えが簡単で、省電力で電池でも回路を動かすことができるというメリットがある。その PIC というものはプログラムで回路を制御するわけだが、使う言語はアセンブラがよく使われる。そのアセンブラで書かれたプログラム PIC に埋め込んで使うわけだが、そのプログラムを埋め込む時に使う MPLAB は現在 Windows でしか環境が整っていない。しかし、それでは Windows を使えない環境下においては PIC という便利なものが使えないことになる。そこで本開発では、アセンブラ言語から HEX コードを生成し、Linux でも PIC に書き込む処理ができるようにする。

## 2 システム構成

## 2.1 開発の流れ

Linux における PIC ライタの開発の簡単な流れを図 1 に示す。まず、アセンブリ言語で書かれたプログラムファイルを Linux 専用のアセンブラでアセンブルし、Hex ファイルにする。そしてその Hex ファイルを書き込み用ソフトウェアを使って今回は書き込む PIC に限定して書き込む。この際の通信手段はシリアル通信を使用し、ハードウェア側で送られてきたシリアル信号をデコードし PIC にデータを流し込む。

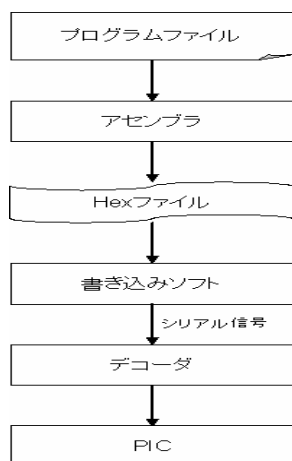


図 1 開発の流れ

## 2.2 Linux 専用アセンブラ

まず、PIC を制御するプログラムを PIC に書き込むためにアセンブラを開発する必要がある。PIC にはアセンブリ言語を直接書き込むことはできない。したがってプログラムを Hex ファイルという直接書き込める形式に変換しなくてはならない。そこでその Hex ファイルを生成するためにアセンブラが必要となる。このアセンブラはアセンブリ言語で書かれた命令を 16 進数に変換し、PIC が読み取れるようなバイナリコードに変換する。

## 2.3 書き込みハードウェア

書き込みハードウェアというのは PC と PIC を直接つなぎ、プログラムを書き込むためのハードウェアである。PC とハードウェアをつなぐ手段としてシリアルポート、パラレルポート、USB などがあるが、今回はノート PC などにも搭載されているシリアルポートを利用してデータの転送を行うものとする。

シリアルポートを使用するにあたり、PC 側から PIC にデータを送る際、PC 側から出た信号を PIC に直接書き込むことはできない。そのため、PC 側から出た信号をデコードして PIC に書き込む必要がある。そのデコーダは PIC マイコンで実現させるものとする(図 2)。PIC は普段使うときは 5 v の電圧をかけるのだが、プログラムを書き込む場合は約 1.2 v の電圧をかけなければならない。つまり 1.2 v の電圧をかけると PIC が書き込みモードに移行する。このハードウェアでは MAX662A という IC を使用し、電源 5 v から 1.2 v を生成させる。MAX662A は非常に安定した 1.2 v を出力してくれる。

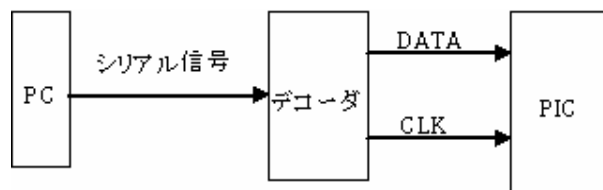


図 2 シリアルポートでの書き込みハードウェアの略図

## 2.4 書き込みソフトウェア

書き込みソフトウェアというのは、PC と PIC を直接つなぐためのハードウェアを制御するソフトウェアである。このソフトウェアでは PIC にプログラムを書き込むためにハードウェアの IC から PIC に 1.2 v の電圧をかけて書き込みモードにしたり、PIC をリセットしたりする。またソフトウェア側でシリアルポートから信号を出力できるように設定しなければならない。

今回は Linux のターミナル上で動作させるので Windows 版のようにウインドウを作成する必要はない。

### 3 システムの実現

#### 3.1 書き込みハードウェア

システムの実現にあたり、まずハードウェアから開発を行う。ハードウェアの回路図を図3に示す[3]。書き込む PIC にデータを書き込む際、必要となるピンは VCC、MCLR、GND、CLOCK、DATA の5つである。始めに MCLR に MAX662A を使用して生成させた 1.2 [v] を与える。それにより PIC は書き込みモードに移行する。次に PC 側から送られてきたシリアル信号をデコーダを介し、RB6 への CLOCK に同期させて RB7 へ DATA を送る。

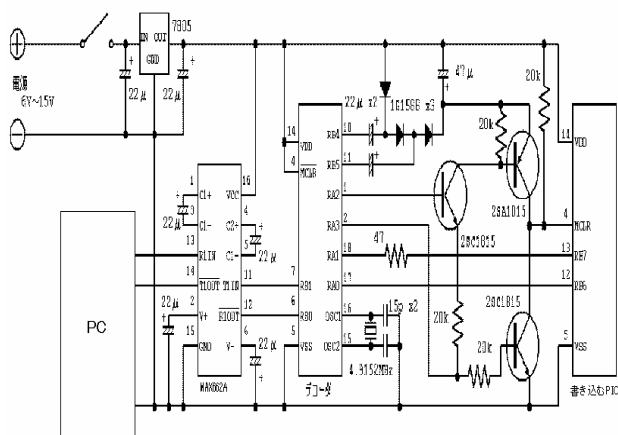


図3 ハードウェアの回路図

#### 3.2 デコーダ

PC 側から送られてきたシリアル信号を PIC にそのまま書き込むことができないので、シリアルポートを使用した PIC ライタにはそのシリアル信号を PIC に書き込める形式にデコードしなければならない。そこでデコーダが必要となる。デコードした信号を書き込むときは、書き込む PIC の CLOCK ピンに CLOCK 信号に同期させて DATA ピンに 1 バイトを 1 つのデータとして書き込む。

デコーダには PIC16C84 を使用し開発を行う。PIC16C84 にはシリアルポート用の専用回路が内蔵されていないため、シリアル接続の制御はソフトウェアで行う。

#### 3.3 書き込みソフトウェア

ハードウェアを扱う際のソフトウェアで行う処理の流れを図4に示す。まず、書き込む PIC に 1.2 [v] の電圧をかけ、書き込みモードに移行させる。次に PC から出力されたシリアル信号をデコーダに入れ、CLOCK に同期させて DATA を送る。最後にデコーダがストップビットを書き込む PIC に出力したら PC 側にも書き込み完了のシリアル信号を出力する。

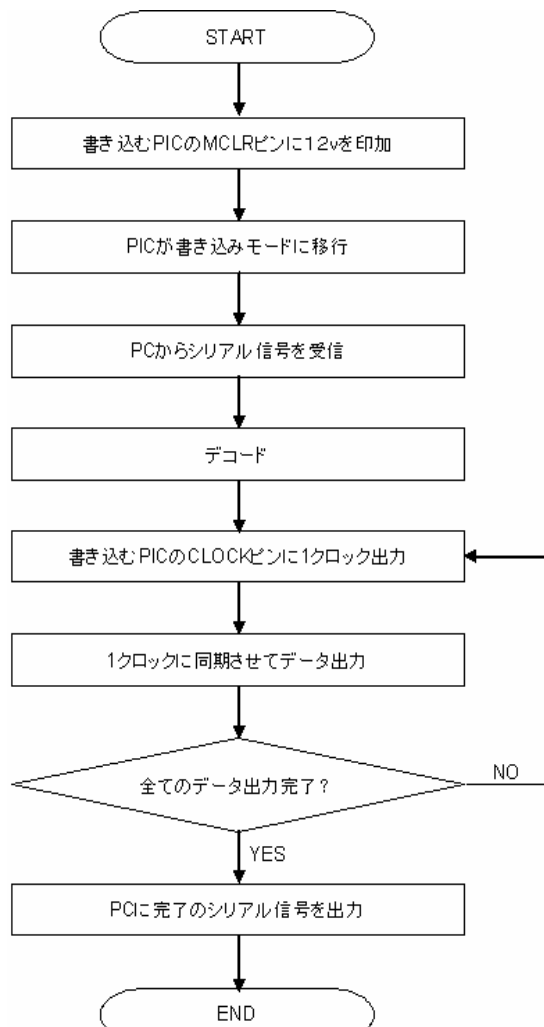


図4 処理の流れ

### 4 まとめ

Linux で PIC マイクロコントローラにプログラムが書き込めるシステムを目標にして、基礎検討を行った。現在は書き込みハードウェアのデコーダを、PIC16C84 を使って開発した。そのためにシリアル通信によって送られてきた信号をどのようにして PIC に渡すかというプログラムのアルゴリズムの基礎検討も行った。

#### 参考文献

- [1] PIC アセンブラ入門  
浅川 毅著 東京電機大学出版局
- [2] C による PIC 活用ブック  
高田 直人著 東京電機大学出版局
- [3] トランジスタ技術 95 年 12 月号  
CQ 出版社