

H8 マイコンによるマトリクス LED の制御

1 年 C 組 青木 雅典

指導教員 米田 隆恒

1. 要約

最近、街中でよく電光掲示板を見るようになった。私も、一度はこのようなものを作りたいと思っていた。そこで今回、マイコン制御の学習を兼ねて、H8 マイコンを使用してマトリクス LED の制御を行った。

キーワード H8 マイコン、マトリクス LED、ダイナミック点灯方式、JIS コード

2. 研究背景と目的

最近、公共施設やお店の広告など、ほとんどの場所で電光掲示板が使用されるようになった。電光掲示板とは、LED(発光ダイオード)を格子状に配置したり、液晶を用いたりすることにより、文字や絵などを表現するものである。私は今回、H8-3664F マイコン(以下 H8 マイコンという)を用いて、8×8 のマトリクス LED(以下 LED という)を制御し、電光掲示板のようなものを作ろうと考えた。

しかし、普通の電光掲示板の仕組みをそのまま H8 に組み込もうとすると H8 自身に文字パターンを記憶させなければならない。すると、どうしても LED 上で表現できる文字の種類に限りが出てしまい、実用性がなくなってしまう。

そこで私は、Visual C# 2008 を用いた自作アプリケーションを作成した。そして、指定した文字をビットマップデータに変換した後、そのデータを RS-232C シリアル通信(以下シリアル通信という)を用いて H8 に送信し、LED 上で文字として表現するという方法を採用した。

ここでは、そこに至るまでの研究経過を報告する。

3. 研究内容

(1) 研究事項

目標は、自作アプリケーションを使用して、パソコンで入力した文字を LED にスクロール表示させることである。



図 1 H8 3664F

(ルネサス エレクトロニクス,
動作周波数 16MHz)

＜研究1＞H8 マイコン単独で、プログラムに組み込んだビットマップデータをLEDにスクロール表示させる。

＜研究2＞パソコンと H8 マイコン間でシリアル通信を行い、キー入力した1文字をリアルタイムでLEDに表示させる。

＜研究3＞自作アプリケーションを使用して、入力した文字列をLEDにスクロール表示させる。

(2) 研究方法とその結果

はじめに、今回の研究で使用する回路を設計・製作する。

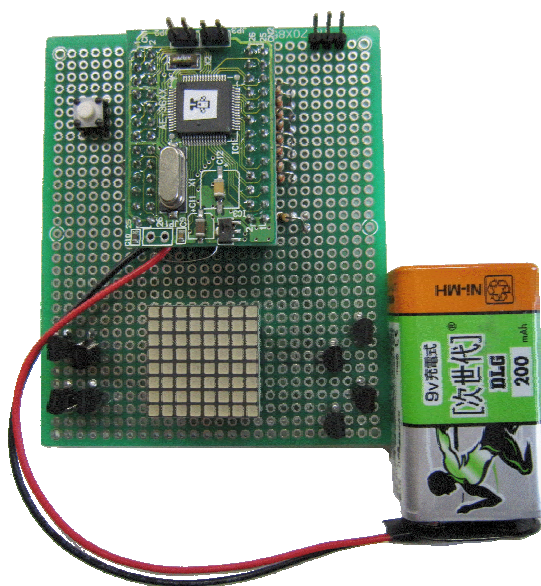


図2 製作したマトリクスLED
制御ボード

この回路には、主に次のような部品を使用している。

- ・ AKI-H8 3664F マイコンボード
(秋月電子通商)
- ・ DotMatrixLED-LT5027E
(SHARP, 赤・緑2色点灯)

- ・ トランジスタ-2SC1815
(東芝, NPN型)
- ・ 炭素皮膜抵抗 1/4W-3種
- ・ 両面スルーホールユニバーサル基板
- ・ その他(配線コード、はんだなど)

今回の研究は、主にこの回路(以下LED制御ボードという)を使用して行っていく。

＜研究1＞

ここでは、LED制御ボードで電光掲示板と同じような動作をさせる。ただし、文字のビットマップパターンはH8マイコン本体に記憶させておく。

まず、プログラムを作成する前に、マトリクスLEDを制御する方法を考えた。今回は、「ダイナミック点灯方式」を採用した。ダイナミック点灯方式とは、人間の目の残像効果を利用したもので、表示させる部分を高速で切り替え、まるでそれぞれの部分が同時に光っているかのように見せる方法である。

この方式を取り入れた理由は、2つある。一つは、LEDのすべてのピンを直接H8マイコンのIOポートに接続していくと、IOポートのピン数が足りなくなってしまうことである。もう一つは、図3のように点灯させたいときに、LEDをH8に直接つないでいると、うまく表示できなくなってしまうということである。

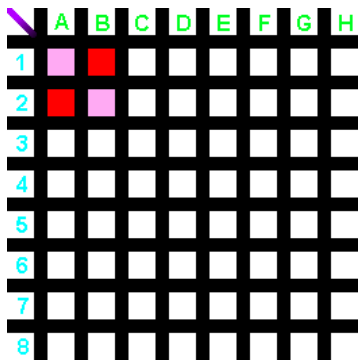


図3 表示させるパターン

■…点灯させたい部分

■…点灯に表示されてしまう部分

ここで、図3について、説明とその解決策を述べておく。図3において、数字側を+電源、アルファベット側を-電源に接続するものとする。

いま、図の濃い部分■(A・2とB・1)を点灯させたいとする。このために、+電源は1と2のみONにし、-電源は、AとBにのみ電流が流れ込むようにする。すると、点灯させたい部分に加え、A・1とB・2(■の部分)も光るようになってしまうため、正しく文字を表現することができなくなってしまう。

そこで、ダイナミック点灯方式を利用する。つまり、まずAにのみ電流が流れ込むようにするとA列のパターンが点灯し、次にBにのみ電流が流れ込むようにするとB列のパターンが点灯する。こうすることで、LEDの点灯場所を一つずつ処理していくことになり、最初のように光らせたくないところが点灯する現象を抑えることができる。

これでLED制御の問題が解決した。次は、実際に電光掲示板のようなプログラムを組んでいく過程である。

LED制御ボードの方は、ダイナミック点

灯の回路を製作した。GND側は、トランジスタ(2SC1815)のスイッチング作用を利用してコントロールする(図4)。一方、+電源側は、それぞれのピン(図3の1~8)をH8マイコンのIOポートに接続し、LEDのON, OFFを出力する。

すると、実際にはLED上で表示する文字を8x8のビットマップデータで書き出したものを縦方向に割り、それを数値化することで表現できることになる(図5)。

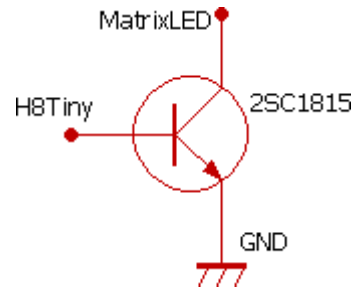


図4 トランジスタの

スイッチング回路部分

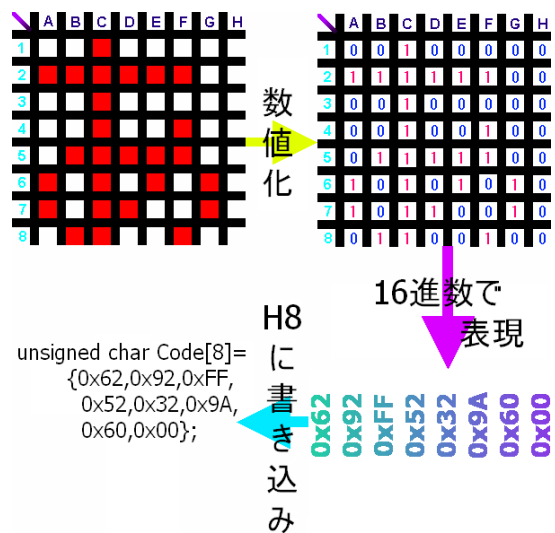


図5 文字パターンをH8マイコンに記憶させるまでの流れ

実際のプログラムは次のように設計した。

まず、表示させたい文字をつなげた状態のビットマップデータを作成し、数値化したものをキャラクタ型の配列変数に入れておく。次に、For 文を回してビットマップデータを一行ずつずらしながら読んでいき、それを LED に表示させる。すると、まるで文字列がスクロールしているかのようにみえるという仕組みである (図 6)。

これで、H8 マイコンのみでマトリックス LED の制御を行い、電光掲示板のようなものを製作することができた。

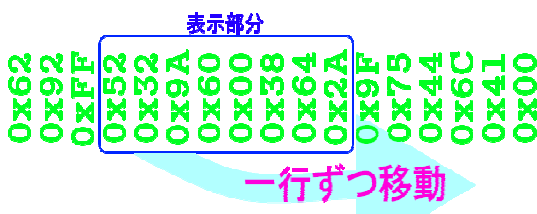


図 6 ビットマップコードを
ずらして読むときのイメージ

<研究 2 >

ここでは、パソコンで入力した文字をリアルタイムで表示させていく。また、今回も文字のビットマップパターンは H8 マイコン本体に記憶させておく。

ここからは、シリアル通信を用いて研究を行っていくため、まずは H8 マイコンをパソコンとシリアル通信させてみた。

私が使用している AKI-H8 3664F マイコンボードには、既にシリアル通信ドライバ IC が実装されているため、マイコン側では、通信速度(ボーレート設定)や割り込み許可などを設定するだけでよい。また、パソコン側では今回、数字や文字の送受信を行うだけなので、ターミナルエミュレーターソフト(Tera Term)を使用して通信を行った。

図 7 は、H8 マイコンとパソコン間で行われるシリアル通信の全体像である。プログラムは、キー入力データをパソコンから H8 マイコンに送信すると、H8 はその文字の表示データを読み込み、LED に表示する、というものである。

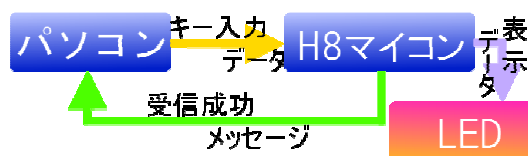


図 7 H8 マイコンとパソコン間での
シリアル通信の全体像

Tera Term では日本語の通信ができないことやプログラム上での処理のしやすさなどを考えて、今回はアルファベットと数字を表示できるようにする。仕組みは簡単であり、パソコンから送られてきた ASCII コードでキーを判定し、条件処理で LED に文字を表示するというものだ。また、ビットマップパターンは、あらかじめ研究 1 と同じ方法で作成しておき、条件処理を行うときに選択して使用する。

これで、入力文字のリアルタイム表示が成功した。対応しているキーならば、どこを押してもリアルタイムに表示できる。しかし、<研究 1 >、<研究 2 >ともに文字のビットマップパターンをすべて手作業で作成しているため、文字の変更が困難であり、あまり実用性が感じられない。

そこで、<研究 3 >ではアルファベットや数字の他に、ひらがなやカタカナ、漢字なども表示でき、8×8 の BDF 形式データならば、どんなフォントも使用できるというようなシステムを開発する。ここで、BDF

形式とは、ビットマップフォントの形式の一種であり、X Window System 用フォントでもある。

<研究 3 >

ここでは、自作アプリケーションを使用して、入力した文字を LED 上にスクロール表示させる。

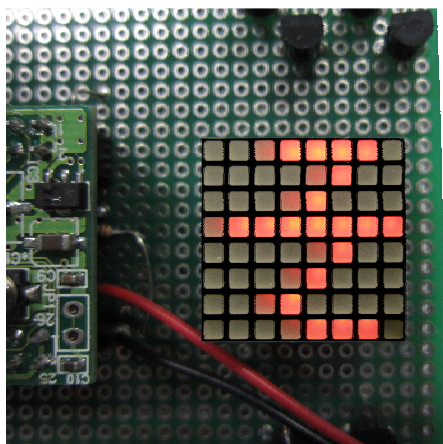


図 8 実際に動作させているところ

開発中の自作アプリケーション(図 8)は、以下のようなことを行う。

- ①表示させる文字列をパソコンから入力(取得)する。
- ②文字列を JIS コードにエンコードし、あらかじめ用意したビットマップデータベース(以下ビットマップ DB という)から JIS コードを検索し、ビットマップデータを取得する。
- ③取得したビットマップデータをバイト配列に格納しておき、シリアル通信で H8 マイコンに送信する。
- ④H8 マイコン側で、受信したデータを研究 1 と同じ方法で、スクロール表示する。

上記の内容をもう少し詳しく説明する。アプリケーションは Visual C#でプログラミングしている。



図 9 開発中の自作アプリケーション

文字列を JIS コード(ISO-2022-JP)にエンコードする理由は、BDF 形式のフォントデータが JIS コードで表記されていることが大きな理由であるが、それ以外にも、エスケープシーケンスから文字種の判別を行うことができるなどのメリットがある。

パソコンでキー入力された文字をビットマップデータへ変換するために、今回は自分で構築したビットマップ DB を使用した。データベースのフォーマットは図 10 のような形式にしている。

私は、8×8 の BDF 形式データから、JIS コードとビットマップデータを抽出し、データベースとして書き出すプログラムも作成した。つまり、これを使用すると、フォントさえあれば様々な文字が使用できるということになり、非常に便利になる。将来的には、このプログラムもメインアプリケーションの機能の一つに取り入れようと考えている。

なお、今回は恵梨沙フォント(恵梨沙フォントPJ, BDFフォント)を使用した。

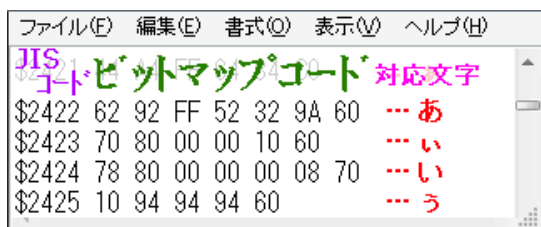


図10 自分で構築したデータベースのフォーマット

H8側のLED制御プログラムは、基本的に研究2と同様なので、ここでは説明しないことにする。

5. 考察

今回製作したLED制御ボードは、主に次のような問題点があった。

- ①赤色点灯時、下2段のみ少し暗くなる。
- ②ときどき、ある一行が光らなくなることがある。
- ③H8メモリ上の関係で、ある一定の文字数よりも多くは表示できない。

これらの問題点の原因と解決方法を私は以下のように考えている。

①については、H8マイコンの構造上、接続しているIOポートのピンは、周りと比べて出力電流が低くなっている。解決策としては、バッファICを利用する、又は使用する抵抗の抵抗値を落とすということが挙げられる(すでに対処している)。

②については、LEDボード裏面の配線に、線を重ねてしまったところがあるため、そこから発生している接触不良かと考えられる。

③については、H8に完全にデータを送っ

てから文字をスクロールしているためである。解決策として、スクロールをしながら少しずつ受信するという方法が挙げられる。

6. 今後の課題

現在の自作アプリケーションは、まだ開発段階であるため、操作性やエラー処理、項目設定などの機能に欠ける。そのため、今後はエラー対処やデザイン、設定項目の増加など機能拡大を図った上で、今まで作成してきたプログラムを一つのアプリケーションとしてまとめていきたい。

また、現在はLEDの赤色の部分しか制御できないので、今後は緑色の部分も制御できるように機能を拡張しようと考えている。

7. 参考文献

- [1] 「ぼくらのマイコン・ロボット工作」 横井浩史、松下光次郎、CQ出版(2006)
- [2] 「Visual C# 2008 逆引き大全 555の極意」 池谷京子、増田智明、国本温子、秀和システム(2009)
- [3] Microsoft社、Visual C# デベロッパーセンター
<http://msdn.microsoft.com/ja-jp/vcsharp/default.aspx>
- [4] Renesas Electronics社、H8/3664グループ データシート
<http://japan.renesas.com/>

8. 謝辞

今回の研究にあたり技術指導をくださった本校サイエンス研究会物理班顧問の米田先生、及びアドバイスをくださったサイエンス研究会物理班の先輩方に、この場をお借りして深く御礼申し上げます。