電気電子工学科の情報・通信実験におけるPBL項目の立ち上げ

－H8/3069Fマイコンを用いた電子回路の設計と製作－

Startup of a Project Based Learning subject for the Information and Communication experiment in the Department of Electrical and Electronic Engineering
－Design and construction of an electronic circuit using an H8/3069F micro computer－

袴田　吉朗*

Yoshiro　HAKAMATA

Abstract: The material outlines a newly settled experimental subject in the Information and Communication experiment in the Department of Electrical and Electronic Engineering. The subject intends a Project Based Learning. Students are ought to use an H8/3069F micro computer as an engine for making an electronic circuit in the subject. Without this, students can freely make electronic circuits. The results of the PBL are outlined in the material. Also given is the knowhow how to use H8/3069F micro computers.

1．　はじめに

電気電子工学科の3年生科目である情報・通信実験においては，従来から「負帰還増幅器の設計と製作」および「マルチバイブレータの設計と製作」という2つのPBLに準ずる実験項目を学生に行わせてきた．これらの項目では「製作」と名前を冠している通り，この製作の過程を通じて学生が自らの考えに基づいて主体的に製作・実験を行うことができるようになっている．しかしながら製作する回路はあらかじめ決められており，PBL項目であるとは必ずしも言えない部分もある．

そこで今年度から「H8マイコンを用いた電子回路の設計と製作」の項目を新設し，学生が主体的に考えて

- マイコンのポートにどのような電子回路，回路素子，センサあるいはモータなどを接続し，
- その接続した回路をどのように制御するかを自分で考え，あるいは図書館で調べたりして
- マイコンにより制御される電子回路を設計し，
- またその回路を動作させる制御プログラムを設計し，マイコンで動作する電子回路を製作して評価する

本格的なPBL項目を実施することにした．

回路の中心的な存在，言わばエンジンとしてH8マイコンを選んだが，それは以下の理由からである．筆者は数年来DSPやPICマイコンを使用して電子回路の開発を行ってきた．特にPICマイコンは，形状が小さく通常のICと同様に実装場所を選ばない特徴がある．この点に大きな魅力を感じ種々の回路の製作に適用してきた．一方市販のH8マイコンは通常ドータボードに搭載されており（搭載せざるを得ず），これをマザーボードに乗せて使用するため，どうしても形状が大型化する．この点で使用するのにためらいがあった．しかし18ピンのPICマイコンは語長が短く，またメモリ容量が小さいためページ切り替えや，バンク切り替えなどに余分な注意を払う必要がある．この点に不満を感じていたことも事実である．一方H8マイコンは，フラッシュROMへ書き込みできる回数が少ないことや，C言語を用いるときにメモリ容量が不足勝ちになることへの不満はあるが，アセンブリ言語を用いる場合にはリソースの点では特に問題は少ないように思える．またメモリ容量に関しては外付けのRAMで対応することも可能である．

H8マイコン以外にも種々のマイコンが開発されており，身の回りを見ても，液晶テレビ，エアコン，冷蔵庫，洗濯機，CDラジカセなどなどの家電製品から，自動車の制御に至るまでいろいろな所でマイコンが電子回路のエンジンとして使用されてきている．特に自動車に関しては，電気自動車はもとより，ハイブリッド・カーや通常の自動車に至るまで，多くのマイコンが使用されるようになってきており，マイコン無しでは自動車は動くことすらできないと言うような現状がある．H8マイコンが全世界における自動車に占める割合が40%にも達しているという事情も仄聞している．

以上のような事情を考慮して本項目の中心的な部品，すなわちエンジンとしてH8/3069F（2MBの外付けDRAM搭載）を使用するマイコンボードを選択することにした．

本資料では，今年度の実施結果をとりまとめ，次年度に向けての参考になるようにした．またH8/3069Fマイコンを使用するときのノウハウを整理した．

2．　実験の日程

情報・通信実験は9月の第4週から開始し，リソースの関係から毎週グループ毎に異なる実験を行わせ，最終的に全項目を終えるように計画して実施してきた．しかしながらマイコンの実験は，教員にとっても学生にとっても初めての経験であり，全員が同時に行った方が何かと都合が良いであろうと判断した．当初H8マイコンの実験には2週間分（コマ数にして6コマ）を割り当てることを予定していたが，これではまだ時間的に十

2013年2月19日受理

*理工学部　電気電子工学科

分ではないであろうとの観測もあった.

そこで予備日等も使用することになる可能性も考慮して，極力終わりの方の日程に組み込むことにした．また設計の過程で必要になった部品を調達する時間的な余裕も考えると，連続した2週間に2回の実験を行う日程では無理があり，最低でも1週間は空ける必要があろうと考えた．これに基づき2案のスケジュールを学生に提示し選択して貰ったところ賛成多数により

- 1回目・・・11月29日
- 2回目・・・12月20日

（12月6日，12月13日は他の実験項目を実施）と言う日程に決定した．なお実際には1月9日の予備日にも第3回目の実験（特にプログラムの作成，マイコンを使ってプログラムを動作させる作業）を行った.

2.1 学生の選んだテーマと実施結果

マイコンボードの数量の関係で，グループ単位（3〜4人）で1つの回路を製作して貰うことにした．全8グループの学生達が選定したテーマと，第3回目の時間の最後に教員，TAおよび学生全員で行った作品コンテストの評価結果を表2.1に示す.

表2.1 実施したテーマおよびコンテストの結果

班名	テーマ	コンテスト結果
1	4×4×4 LEDキューブ	A
2	ドットマトリクスLED(1個)	C
3	スイッチによる点滅LEDの制御	B
4	LEDの点滅	C
5	ドットマトリクスLED(2個)	C
6	LEDの点滅	C
7	圧電ブザーによる音の発生	B
8	LEDの点滅	B

実験実施時点において，マイクロプロセッサ応用の講義を受講している学生はいたものの，H8マイコンに関してはまだ十分に勉強している訳ではなく未熟であった．またこの講義を受講していない学生も多数いた．そのため実験テキストに例題として記載しておいたLED点滅のテーマを無難に（換言すれば安易に）選んだグループが半数を占めたものと思われる.

反面1班の「4×4×4 LEDキューブ」は秀逸であった．特に配線および半田付けは非常に美しくできており，その見事さは賞賛に値した．プログラムの動作も問題なく，青色LEDをプログラム制御により点滅させる訴求力のある作品にできあがっていた．コンテストの結果でも多くの支持を得ており，また羨望の眼を注がれていたものと思われる.

マトリクスLEDを用いた文字（アルファベット）の表示回路には，2つの班がチャレンジした．もし実現できていれば製作した学生達も大きな達成感を味わうことができ，また他の学生達にも興味を喚起できたものと思われる．しかし残念なことに完成には至らなかった．実験終了後に筆者が配線を確認したところ，数ヶ所の配線ミスが見つかった．事前の配線図のチェック，半田付けを行った部分をマーカーペンでなぞらせてミスをなくす，などをより徹底させる必要性を感じた.

7班の圧電ブザーによる音の発生は，3音の発生はできたものの，ドレミファソラシドの音階を発生させるまでには至らなかった．学生達に周波数や周期に関する知識が十分にあるのか否かの懸念が頭をよぎった.

3班の，点滅させるLEDをスイッチ入力により変更する回路は，コンテスト開始時間ギリギリであったが完成させることができた.

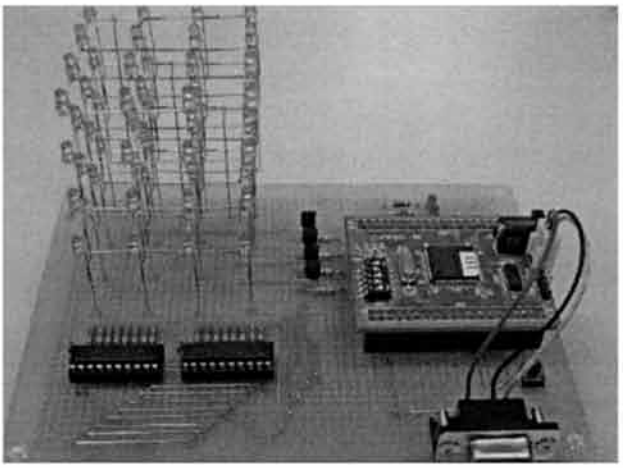


図2.1 4×4×4 LEDキューブ

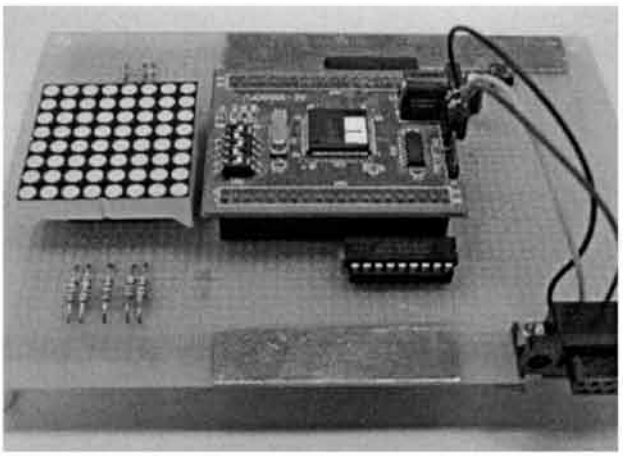


図2.2 ドットマトリクスLED（1個）

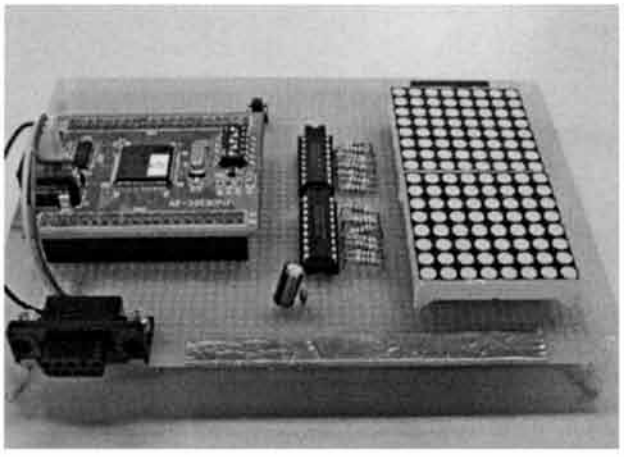


図2.3 ドットマトリクスLED（2個）

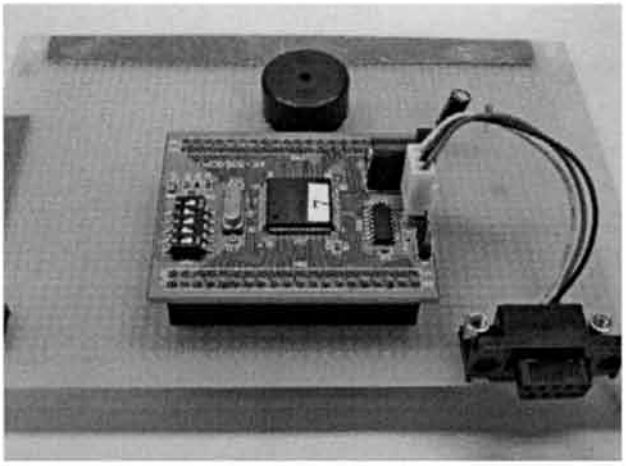


図2.4 圧電ブザーによる音の発生

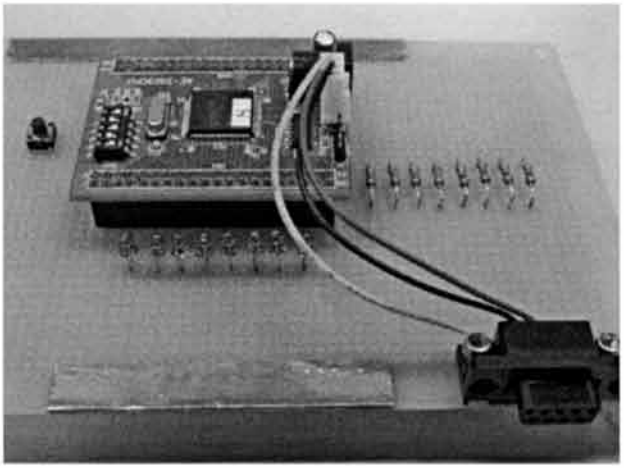


図2.5 LEDの点滅（8班）

2.2 本PBL項目を実施したことにより得られた知見

① 第1回目の「何を作るのかを決める」,「回路図を作成する」,「部品表を作成する」ための時間には当初3コマ（合計5時間）を配分した．また時刻を16時までと限定して「何を作るのか」を提出するように促したが，あらかじめ準備をしていたと思われるグループを除きなかなかテーマが決まらないようであった．

② 回路の製作が遅い

3〜4名の班員で1つの回路を作成するため，作業がタイムシリアルとなり製作を完了させるまでに多くの時間がかかる傾向が見られた．回路製作とプログラムの作成などを分担して行っているグループもあったが，なかなかそのようにはうまく行かないグループも見られた．

レポートの感想には，作業を分担しない学生は手持ちぶさたになり，不公平であるとの意見も書かれていた．

③ 恐らく大半の学生が，H8マイコンのアセンブラはもとより，C言語のプログラム経験も少ない（あるいは全く経験がない）ため，なかなかプログラムができないように見受けられた．

④ 本実験ではマイコンのROMへのプログラム書き込み回数を減少させるため，RAMにプログラムを読み込んで動作させてデバッグを行うRAMエミュレーションを使用した．

そのためまずモニタプログラムをH8マイコンに書き込む必要があった．しかしこれがなかなかできなかった．これは必ずしも学生の未熟さだけが原因ではなく，マイコンとPCを接続するRS232Cコネクタにルーズコンタクトがあったのではないかと推察される蓋然性があった．

⑤ 11月22日にそれまでに実験した項目に関する発表会を実施した．その際に終わりの約1時間を割いて「H8マイコンのいろは」に関する資料を配り，ガイダンスを実施した．しかし実際にPBLを行った時点においては殆どその内容を理解していないか，あるいは記憶していなかった．

⑥ 但し，分からないことでも実験テキストに書かれている内容を見て自分たちで苦闘してみたり，あるいはそれでも分からないことは教員に積極的に質問してきたりしたので，PBLの目的は達成できていたものと思える．

今年度は，実験テキストを作成したのが8月の夏休み中であり，そのときにはPCにWindowsXPマシンを使用した．現3年生のPCがWindows7であることから，10月になってから急遽動作の確認を行ってみた．その結果不幸なことにテキストに記述した内容ではうまくいかないことが判明した．そこでかねてからルネサス社の担当者と親交があると聞いていた電気電子工学科の高橋教授に相談した結果，ルネサス社からWindows7マシンで動作するHEW接続シリアルモニタの提供を受けることができた．またモニタを学生に開示する承諾も貰えたので急遽資料を作成し，11月22日にガイダンスを行った経緯がある．

来年度は必要な資料を全て実験テキストに記載することができるので，よりよい結果になることを期待したい．

3． H8/3069F（2MB外部DRAM付き）の使用方法

3.1 回路図およびI/Oポートの機能

図3.1にマイコンボードの回路図を示す．電源は図左上にあるコネクタ（J1）を介して供給する．3端子レギュレータが使用されているが，ホット側のピンを，メッキ線を用いて半田付けし，3端子レギュレータを使用しないで，直接5Vの安定化電源パックから電源を供給するようにしている．なおクロック周波数は25MHzである．

マイコンのリセット端子は63番ピンであるが，本ドータボードには配線されていない．しかし電源を抜くことによりリセットを代替することができる．HEW接続シリアルモニタの説明書（readme.htm）にはNMI端子（64番ピン）にスイッチを接続してLに落とし，NMI割り込みを掛けてリセットを代替する方法が示されている．これは単にCPUボードの電源を切っただけではHEWは停止せず，結局タスクマネージャでキルしなければならないからである．本実験でもこの方法を使用した．

3.2 使用できるポート

表3.1は各I/Oポートの機能を整理した表である．マイコンには多くの機能が集約されており，マイコンのピンも各種の信号で共用されている．どのようにマイコンを動作させるか，ど

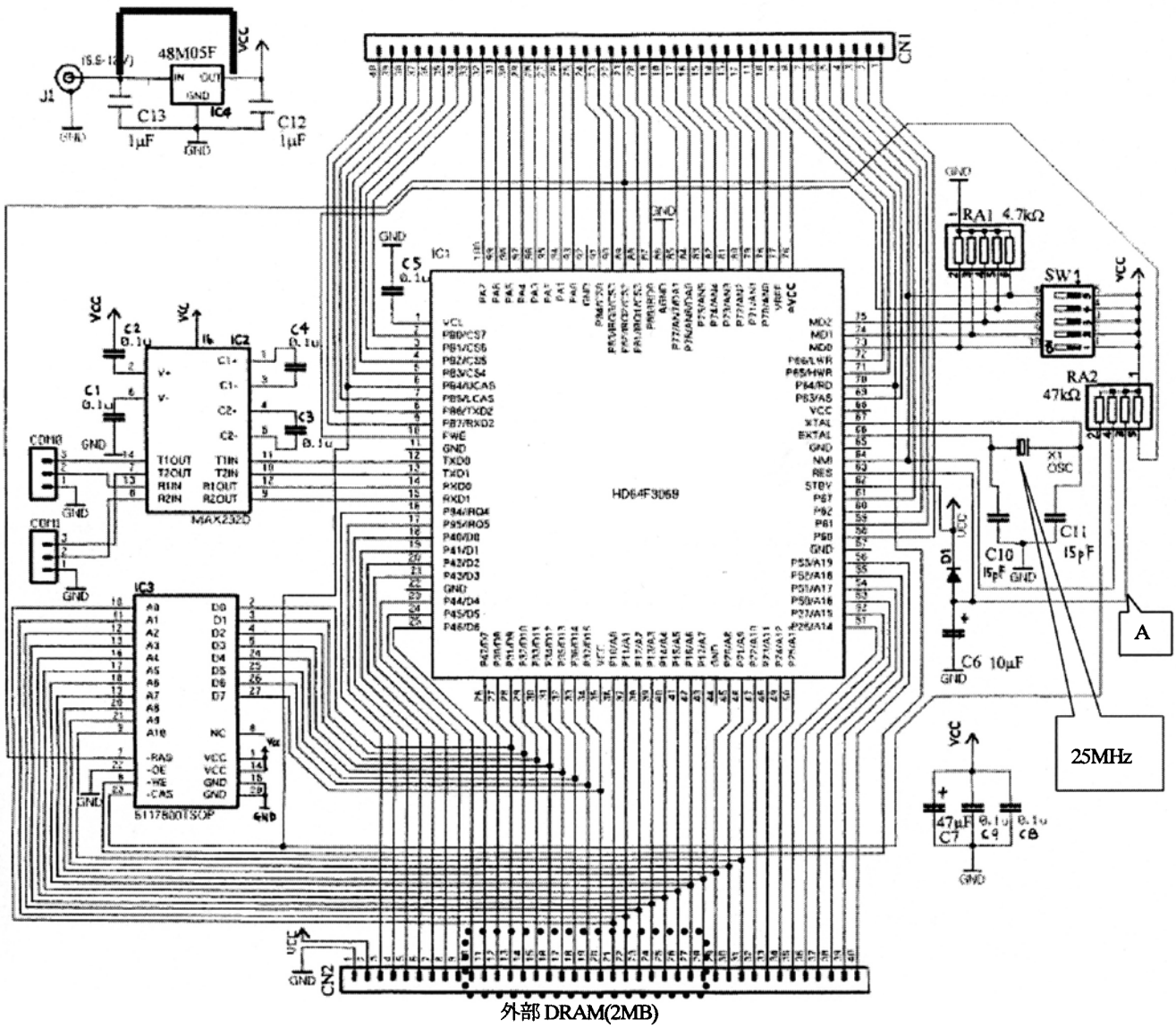


図 3.1　H8/3069F マイコンボードの全体回路図

表 3.1　H8/3069F マイコンの I/O ポートの機能

		DRAM 用	入力ポート	出力ポート	内蔵周辺機器			DRAM 用	入力ポート	出力ポート	内蔵周辺機器
ポート 1	$P1_7$～$P1_0$	○	×	×	−	ポート 8	$P8_4$～$P8_3$	−	○	×	CS 端子，IRQ_3
ポート 2	$P2_2$～$P2_0$	○	×	×	−		$P8_2$	○	×	×	(CS2，IRQ_2)
	$P2_7$～$P2_3$	−	○	×	−		$P8_1$	−	○	×	CS 端子，IRQ_1
ポート 3	$P3_7$～$P3_0$	○	×	×	−		$P8_0$	−	○	○	CS 端子，IRQ_0
ポート 4	$P4_7$～$P4_0$	−	○	○	−	ポート 9	$P9_7$～$P9_0$	−	○	×	SCI0,SCI1
ポート 5	$P5_3$～$P5_0$	−	○	×	−	ポート A	PA_7～PA_0	−	○	○	TPC,タイマ
ポート 6	$P6_2$～$P6_0$	−	○	○	バス制御	ポート B	PB_7～PB_6 PB_3～PB_0	−	○	○	TPC,タイマ，DMA
	$P6_7$～$P6_3$	−	×	×	バス制御		PB_5	○	−	−	/LCAS
ポート 7	$P7_5$～$P7_0$	−	○	×	A/D		PB_4	○	−	−	/UCAS
	$P7_7$～$P7_6$	−	○	○	A/D，D/A						

のピンにどんな信号が出力されるのか(あるいは入力すべきか)を適切に判断して処理を行う必要がある．詳細は「H8/3069F-ZTATTM ハードウェアマニュアル」の P.285～P.350 に書かれている．

表 3.1 において DRAM 用に○がついているポートは，既に外部 DRAM に接続されており，ユーザが使用できないポートで

ある．ユーザは DRAM 用がーになっているポートを使用できるが，入力ポート，出力ポートあるいは内蔵周辺機器でポートを共用しているので，それらが競合しないように適切に使用する必要がある．

これより，入力ポートとして使えるポートの数は多いものの，逆に LED の点灯に使えるような単なる出力ポートとして使えるポートはあまり多くないことが分かる．なお，「H8/3069 F-ZTATTM ハードウェアマニュアル」の P.792 に書かれているように，ポート 1, 2 および 5 以外のポートでは，各ポートに流せるシンク電流，ソース電流の最大許容値が 2mA と小さい．したがってマイコンで直接 LED を駆動するときにはこの許容電流を超えないように電流制限抵抗を大きめに設計するか，あるいはダーリントンシンクドライバ(あるいはソースドライバ)を介して駆動するようにする必要がある．

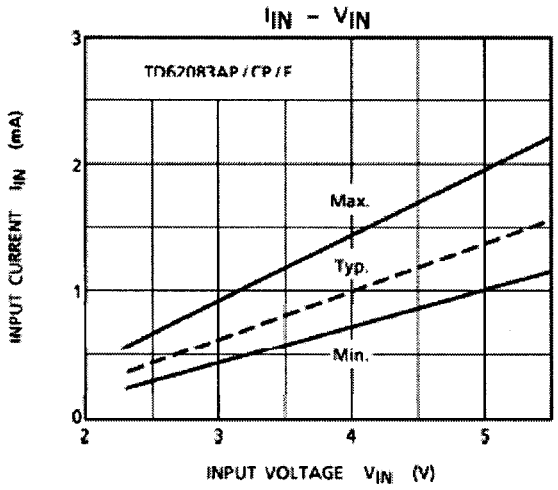


図 3.2 シンクドライバ TD62083APG の入力特性

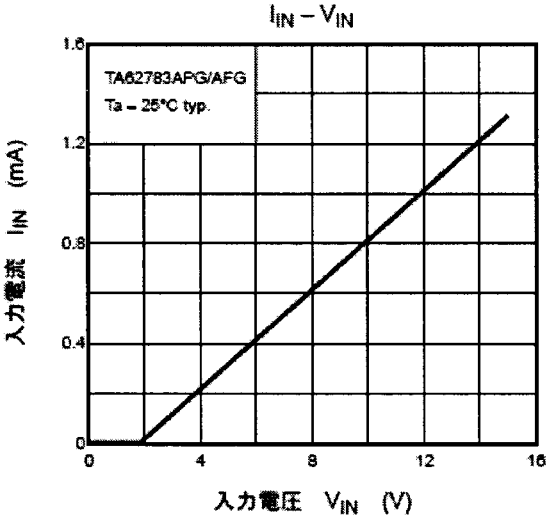


図 3.3 ソースドライバ TD62783APG の入力特性

本実験ではダーリントンシンクドライバ TD62083APG およびダーリントンソースドライバ TD62783APG を使用した．図 3.2 および図 3.3 はカタログから抜粋した両ドライバの入力電圧ー入力電流の特性である．これより 5V 入力時のソースドライバの入力電流は 0.3mA であり，H8 マイコンと直結しても良さそうである．しかしシンクドライバの場合には入力電流の MAX 時約 2mA となり直結は好ましくないと思われる．先の「H8/3069 F-ZTATTM ハードウェアマニュアル」には 2kΩの抵抗を挿入するようにと記載されている．

実際にこれらのドライバを使用したのは，LED マトリクスアレイの回路である．正常な動作ができていないのは，直結したことによりマイコンを破壊した恐れがあるが，現時点ではまだ確認ができていない．

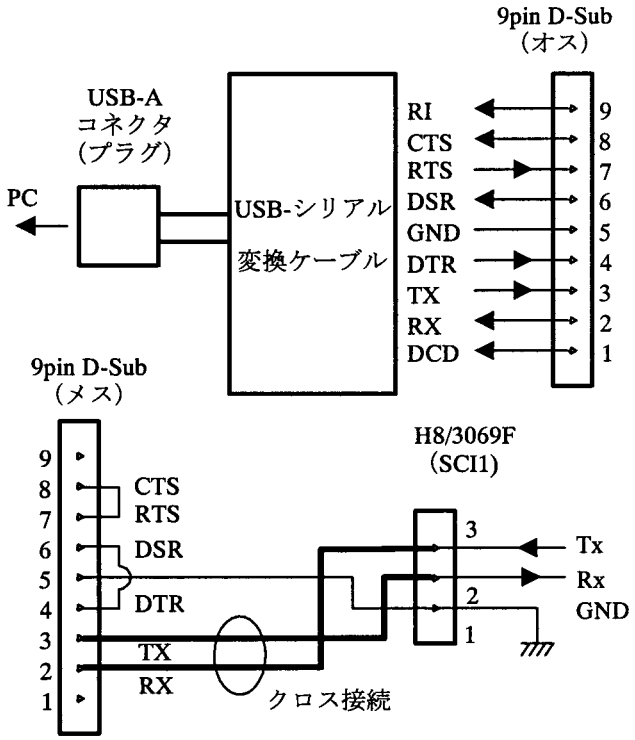


図3.4 USBーシリアル変換ケーブルを用いた PCとH8マイコンの接続

3.3 PC との接続

実験で使用するマイコンボードは 3 チャネルのシリアル通信チャネル（SCI ポート）を有している．そのうち SCI0（COM0）および SCI1（COM1）の 2 チャネルはマイコンのドータボード上のヘッダピンに配線されており，信号を外部に取り出せるようになっている．ユーザの開発したプログラムを PC からマイコンの ROM に書き込むためには，SCI1（COM1，電源端子に近い側のコネクタ）を使用する．また後述するように，プログラムのデバッグも SCI1（COM1）を介して PC と通信しながら行うことになる．

旧来の PC には COM ポートがついているものが多かった(今ではレガシーインタフェースと呼ばれている）が，最近ではこの COM ポートがついていない PC の方が多くなってきており，学生が使用する PC にも COM ポートはついていない．そこで図 3.4 に示す USBーシリアル変換ケーブルを使用して接続した．

参考のためにRS232Cコネクタ（プラグ）の形状を図3.5に示しておく．

なおそのままストレートに接続すると通信ができないので，TX線（2番ピン）とRX線（3番ピン）が接続されるように，H8マイコンを搭載するマザーボード上でクロスさせて配線しておく必要がある（図3.4の太線で示した部分）．また周知のようにフロー制御のための信号であるRTS−CTSおよびDSR−DTRは配線により短絡しておき，PCから送信した信号がそのまま返送されるようにしておく．

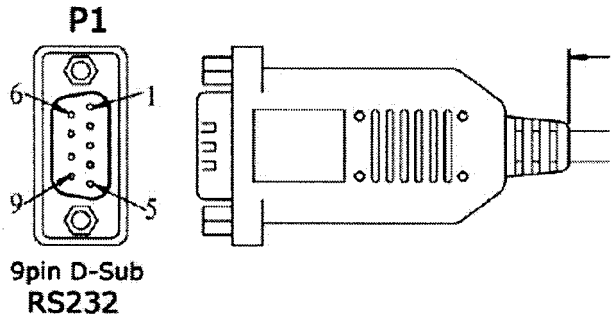


図3.5 RS232Cコネクタ（プラグ）の形状

3.4 H8/3089F（2MB外部DRAMつき）の物理メモリ

H8/3069Fマイコンの物理メモリの具体的な配置(外部DRAM容量2MB，モード5）は図3.6に示す通りである．H8マイコンのROMへの書き込み回数制限は公称100回程度と言われており，これが事実とすれば非常に少ない．そこで通常デバッグ時には，モニタプログラムを用いてユーザプログラムをRAMにロードして動作させるようにし，極力ROMへの書き込み回数を減少させるようにして開発を行う方法が用いられる．本実験でもこの方法を用いた．

HEW接続シリアルモニタモニタの説明書（readme.htm）に書かれているように，モニタが内蔵RAMの一部領域（4KB）をワークエリアとして使用するので，ユーザの使用できる内蔵RAMの領域は約12KBになる．

逆に0〜0xFF番地は実際にはROM領域であるが，ユーザが割り込みベクタを設定することができるようになっている．

H'000000 H'0000FF	割り込みベクタ	
H'000100 H'005FFF	モニタ本体	H8の内蔵ROM (512KB)
H'006000 H'07FFFF	未使用	
H'080000 H'1FFFFF	*	エリア0 (1536KB)
H'200000 H'3FFFFF	*	エリア1 (2048KB)
H'400000 H'5FFFFF	ユーザ プログラム	外付けDRAM エリア2 (2048KB)
	*	
H'FEE000 H'FEE0FF	H8のI/O(1)	H8のI/O(1) (256B)
	*	
H'FFBF20 H'FFDFFF	ユーザ プログラム	H8の内蔵RAM (8416B)
H'FFF700 H'FFFA1F	モニタ用ワークエリア(4096B)	
H'FFF000 H'FFFF00 	ユーザ プログラム スタックエリア ユーザ プログラム	H8の内蔵RAM (3872B)
H'FFFF20 H'FFFFE9	H8のI/O(2)	H8のI/O(2) (202B)
H'FFFFEA H'FFFFFF	*	エリア7 (22B)

図3.6 HEW接続シリアルモニタを用いてRAM動作をさせる場合のメモリマップ

4. モニタプログラムの作成方法

ダウンロードし展開したホルダーRenesas Htermの中にあるreadme.htm（HEW接続組み込み型モニタのカスタマイズ方法）に従ってモニタプログラムをカスタマイズする．

4.1 スタートアップモジュールHardwareSetup.cの作成

H8/3069F（2MB外部DRAM付き）用のスタートアップモジュールの作成結果を図4.1の HardwareSetup.c に示す．

以下に各設定の説明を示す．なお［ ］内に示してある数字はH8/3069F-ZTATTMハードウェアマニュアルにおけるページ数である．

① ABWCR（バス幅コントロールレジスタ）[p.134]

ビット	7	6	5	4	3	2	1	0
	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0

各ビットがエリア0〜エリア7のバス幅を決める．1のとき8ビットアクセス，0にすると16ビットアクセスになる．本実験で使用するDRAMは8ビットアクセス空間を用いる．モード5では，リセット時にABWCRはH'FFにイニシャライズされ，全エリアが8ビットアクセス空間になるので，特に設定する必要はない．

② RTCOR（リフレッシュタイマコントロールレジス）[p.154]

リフレッシュカウンタRTCNTに，システムクロックφを

RTMCSR で設定した値に分周したクロックが入力され，RTCNT の値が RTCOR と比較され，一致したときにリフレッシュサイクルが起動される．なお DRCRB の RCYCE を 1 に設定しておく必要がある．

③ RTMCSR(リフレッシュタイマコントロール／ステータスレジスタ) [p.152]

ビット	7	6	5	4	3	2	1	0
	CME	CMIE	CKS2	CKS1	CKS0	(1)	(1)	(1)

割り込みを使用しないので CME=CMIE=0 (初期値) にしておく．CKS2～CKS0 の値でシステムクロック (φ=25MHz) を何分周して，リフレッシュカウンタ RTCNT (アップカウンタであり，PLL ではない) に供給するかを設定する．

搭載されている DRAM は M5117805F-60TK(OKI)である．この DRAM のデータシートを入手できなかったが類似品 (MSM5117805F) のデータシートによれば，リフレッシュレートは 2048 回／32ms，すなわち 64kHz である．したがってこれを満足するように RTCOL および RTMCSR における CKS の値を選ぶ．システムクロックの値を φ(=25MHz)とすると RTCNT に入力されるクロックは CKS の値により以下のようになる．

CKS

0	カウントなし	
1	φ／2	(=12.5MHz)
2	φ／8	(=3.125MHz)
3	φ／32	(=78.1kHz)
4	φ／128	(=19.5kHz)
5	φ／512	(=48.8kHz)
6	φ／2048	(=12.2kHz)
7	φ／4096	(=6.1kHz)

アップカウンタ RTCNT を用いて 64kHz のクロックを作成するためには，CKS を 1～3 とする必要がある．自由度の多さを考えて CKS=1，すなわち φ／2 (RTMCSR = H'0F) を選ぶものとすれば，リフレッシュレートは以下のようになる．

- RTCOL=195 で約 64kHz のリフレッシュ
- RTCOL=125 で 100kHz のリフレッシュ
-

④ DRCRB (DRAM コントロールレジスタ B) [同, p.149]

ビット	7	6	5	4	3	2	1	0
	MXC1	MXC0	CSEL	RCYCE	(1)	TPC	RCW	RLW

エリア 2 を使用するが，①で述べたように本実験で使用するマイコンではエリア 2 は 8 ビットアクセス空間になっている．

- MXCはカラムアドレスのビット数を指定する．

MXC

0	8ビット
1	9ビット
2	10ビット

になっている．

本実験で使用するDRAMは

- カラムアドレス10ビットのもの MXC=2
- CSELはUCAS，LCASの出力端子の設定である．PB4およびPB5を使用する CSEL=0
- RCYCEはCASビフォアRAS (CBS) リフレッシュの使用を許可する RCYE=1
- TPCはプリチャージサイクルを1ステート挿入する TPC=0
- RCWはリード／ライト時にRAS，CAS間へのウェイト挿入を禁止する RCW=0
- RLWはリフレッシュ時にRAS，CAS間へのウェイト挿入を禁止する RLW=0

以上より DRCRB=B'1001 1000=H'98 とする．

⑤ DRCRA (DRAMコントロールレジスタA) [p.147]

ビット	7	6	5	4	3	2	1	0
	DRAS2	DRAS1	DRAS0	(1)	BE	RDM	SRFMD	RFSHE

DRASAはエリア2～5のそれぞれをDRAMで使うのかSRAMで使うのかを設定する．

DRAS	DRAMを使う空間
0	DRAMは使わない
1	エリア2 (CS2) のみ
2	エリア2 (CS2)，エリア3 (CS3)
3	エリア2，エリア3をCS2だけで4MBの空間として使う
4	エリア2 (CS2)，エリア3 (CS3)，エリア (CS4)
5	エリア2 (CS2)，エリア3 (CS3)，エリア (CS4)，エリア5 (CS5)
6	エリア2，エリア3をCS2，エリア4，エリア5をCS4で使う
7	エリア2～エリア5をCS2で使う

エリア 2 に DRAM が増設されているので，DRAS=1 (DRAS2=DRAS1=0, DRAS0=1) に設定する．

- DRAMが高速ページモードを使える場合には BE=1 に設定すると，連続でメモリを読む場合には早くなる．このDRAMは高速ページモードを使えるので BE=1 とする．
- RDMは高速ページモード時におけるバーストアクセス時に，RASアップモードを使うか，RASダウンモードを使うかの設定である．今回はRASダウンモードを選択し RDM=1 とする．
- SRFMDはセルフリフレッシュをするか否かの設定である．対応していないので SRFMD=0 とする．
- RFSHEはRFSH端子 (P80) から外部にリフレッシュ信号を出すか否かの設定である．必要ないので RFSHE=0とする．RFSHE=0のときポートP80は入出力ポートとして使用できる．

以上より DRCRA=B'0011 1000=H'38

とする．

⑥ ポートディレクションレジスタの設定 [p.290, p.293, p.315]

外部DRAMの容量は2MBである．また8ビットアクセスであり，2^{11}=2048であるから11本のアドレス線（A0～A10）が必要になる．表3.1に示したように，これらのアドレスはポート1（P10～P17）およびポート2の一部（P20～P22）と共用になっている． DDRレジスタを出力に設定するとアドレス線として使用できるので，まずポート1およびポート2における前記のDDRレジスタのビットを1にしている．ポート8（5ビットのポート）に関しては，チップセレクト信号の出力端子として使用するためエリア2（CS2）に対応するビット2をDDR＝1に設定する．上位3ビットはdon't'careである．

P1DDR = H'FF　　アドレス　A0～A7
P2DDR＝H'07　　アドレス　A8～A10
P8DDR＝H'04　　エリア2（CS2）のみ使用

⑦ WCRH, WCRL（ウェイトコントロールレジスタ）[p.136]

ビット	7	6	5	4	3	2	1	0
WCRH	W71	W70	W61	W60	W51	W50	W41	W40
WCRL	W31	W30	W21	W20	W11	W10	W01	W00

W7～W0はそれぞれの外部DRAMエリアのウェイトステートを決める．外部DRAMをエリア2に接続しており，W21およびW20に設定した値によりウェイトステートが決まる．デフォルトでは11になっており3ステートのウェイトになっている．

H8/3069Fのクロックは25MHzであり，1クロック時間は40nsである．使用しているDRAMにおいて，CASからのアクセス時間（t_{CAS}）は最大15nsである．したがってDRAMを接続したエリアCS2のウェイトステートを0（W21=W20=0）とし，DRAMアクセス時のウェイトを0とする．

よって　WCRH＝B'1111 1111＝H'FF,
　　WCRL=B'1100 1111=H'CF　　となる．

なお内蔵メモリおよび内部I/Oレジスタに対するアクセスステート数はWCRH、WCRLの設定値にかかわらず固定である．

⑧ ASTCR(アクセスステートコントロールレジスタ) [p.135]

ビット	7	6	5	4	3	2	1	0
	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0

それぞれのエリアのアクセスステート数を決める．DRAMは高速でありアクセスステート数=2とする．

よって　ASTCR=B'1111 1011=H'FB　とする．

⑨ wait80us（8リフレッシュサイクル分のwaitを入れる）

DRAMの説明書によれば，「電源投入後V_{CC}が規定の電圧に到達してから200μs以上のポーズをとり，その後8回以上のリフレッシュサイクル（CASビフォアRASリフレッシュサイクル時）を加えて下さい」と注記されている．

DRAMのリフレッシュレートが2048/32ms=64kHzであるから，8サイクルでは125μsの遅延時間になる．この遅延時間をプログラムではNOP命令を使って作成している．refresh_loopは1回で20ステートであり，これを25MHzクロックで実行すると0.8μsになる．したがってループ回数は計算上156.25回となる．これを多めに見積もって250回としている．

以上の結果をまとめたものが図4.1の結果である．なおBSCはDRAMやバス関連のレジスタを束ねる上位概念であるが，ハードウェアマニュアルには明示的な記載はない．しかし3069s.hには記載されている．

```
1   #include "iodefine.h"
2
3   void wait_80u(void)
4   {
5       volatile int t= 250;
6       while(t--);
7   }
8
9   void HardwareSetup(void)
10  {
11      BSC.RTCOR           = 125;
12      BSC.RTMCSR.BYTE     = 0x0F;
13      BSC.DRCRB.BYTE      = 0x98;
14      BSC.DRCRA.BYTE      = 0x38;
15
16      P1DDR               = 0xFF;
17      P2DDR               = 0x07;
18      P8DDR               = 0x04;
19
20      BSC.WCR.BYTE.H      = 0xFF;
21      BSC.WCR.BYTE.L      = 0xCF;
22
23      BSC.ASTCR.BYTE      = 0xFB;
24
25      wait_80u();
26  }
```

図4.1 スタートアップモジュールHardwareSetup.cの作成結果

4.2 シリアルチャネルにおけるボーレートの設定

WindowXP+Htermの環境下ではリンケージコマンドファイルを用いて行っていた設定が，HEW接続シリアルモニタでは直接モニタのタブにより設定できるようになった．一部シリアルモニタのボーレートについては計算式を示したおいた方が後々の参考になると思われるので以下に示しておく．

readme.htmには計算式が示されていないが，H8/3069F-ZTATTMハードウェアマニュアルによれば次式で表される．

$$N = \frac{\phi \times 10^6}{64 \times 2^{n-1} \times B} - 1$$

但し：φ・・・システムクロック（=25MHz）
　　n・・・分周比
　　B・・・SCIチャネルのボーレート

n=0, B=38400baudとすれば，N=19（10進数，16進数では13）
n=0, B=19200baudとすれば，N=40（10進数，16進数では28）
n=0, B= 9600baudとすれば，N=80（10進数，16進数では50）
となる．

ボーレートは早いほうが良いが，正しく通信できなければ意味がない．試してみて文字化けのしない最速のボーレートに設

定する．WindowXP+Htermの環境下では，モニタを書き込むときには38400kbaudで良いが，ユーザプログラムの読み込み時には9600baudまでボーレートを下げる必要があった．

HEW接続シリアルモニタの場合には両方とも38400kbaudに設定すれば良い．逆にHtermの場合と違いユーザプログラムの読み込み時に9600baudまでボーレートを下げると動作しない．

5． C言語を用いたプログラム例

ポート6に接続したLEDを，タイマー割り込みを用いて1s間隔で点滅させるプログラムを図5.1に示した．以下このプログラムを例にとり説明する．

```
#include    "3069s.h"
int cnt;
void int_imia0(void)
{
    ITU.TISRA.BIT.IMFA0 = 0;
    cnt -= 1;
    if( cnt == 0)
    {
        P6DR.BYTE ^= 0xFF;
        cnt = 50;
    }
}
void main(void)
{
    P6DDR                = 0xFF;
    P6DR.BYTE            = 0x00;
    ITU0.TCR.BIT.CCLR    = 1;
    ITU0.TCR.BIT.TPSC    = 3;
    ITU0.GRA             = 62499;
    ITU.TISRA.BIT.IMIEA0 = 1;
    ITU.TSTR.BIT.STR0    = 1;
    cnt = 50;
    while(1);
```

図5.1 割り込みによるLED点滅プログラム

プログラムあるいはデータをメモリに配置して管理するのはリンカの役目であり，ユーザがHEWの ビルド／H8_300 Standard Tool Chain／最適化リンカ／セクション の部分において設定する．

「H8S、H8/300シリーズ C/C++コンパイラ、アセンブラ、最適化リンケージエディタ」のp.184, 表9.1に詳細が示されているが，代表的なセクションとそれを配置するメモリ種別を表5.1に示す．

5.1 プログラムをROMに書き込んで動作させる場合

最終的にはプログラムをマイコンのROMに書き込んで動作させる．HEWを用いてC言語のプログラムを作成し ビルド／H8_300 Standard Tool Chain／最適化リンカ／セクション を開くとデフォルトで図5.2のようになっている．

PResetPRG～DのセクションはROMに配置されている．セクションB，RおよびSはRAMに配置されている．この例では外部DRAMには何も配置されていない．プログラムが大きくなるとセクションPが大きくなるので，表5.1の原則を守りつつプログラムが収容できるように境界を調節する必要がある．

なお図5.2には示されていないが，実際には0～0xFF番地には割り込みベクタが配置されている．ベクタテーブルはintprg.cにおいて，例えばタイマー割り込みの場合には

__interrupt(vect=24) void INT_IMIA0(void) {/* sleep(); */}

と言うように＜割り込みベクタ番号＞（vect=24）を記述することによって自動的に生成される．

表5.1 代表的なセクション

領域		セクション		
		名称	属性	メモリ
1	プログラム領域	P	code	ROM
2	定数領域	C	data	ROM
3	初期化データ領域	D	data	ROM
4	未初期化データ領域	B	data	RAM
5	初期化データセクションのアドレス領域	C$DSEC	data	ROM
6	未初期化データセクションのアドレス領域	C$BSEC	data	ROM
7	スタック領域	S	stack	RAM

図5.2 ROM動作時のメモリ配置

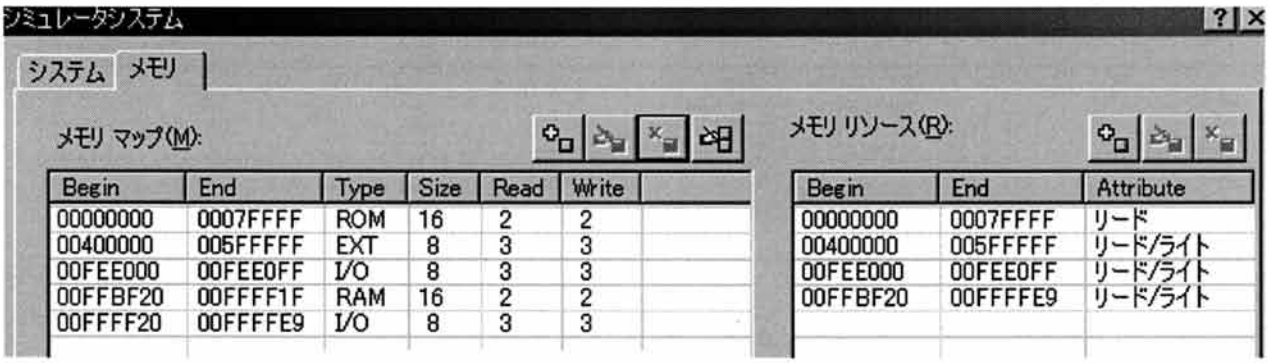
5.2 デバッグにおける設定

(1) HEWのシミュレータを用いたシミュレーション

H8マイコンはROMへの書き込み回数の最大値が100回程度と少ないようである．従ってデバッグを行うには，まずPC上でシミュレーションを行い，その後にPCとパソコンを接続してプログラムをRAMに読み込んでエミュレーションを行う方法が適していると考えられる．これによりROMへの書き込み回数の低減が図れる．

この場合には表5.1および図5.2においてROMに配置されたセクションをRAMに移動させる必要がある．図5.3にメモリ配置の一例を示す．

図5.3 RAM動作時のメモリ配置の例



Begin	End	Type	Size	Read	Write
00000000	0007FFFF	ROM	16	2	2
00400000	005FFFFF	EXT	8	3	3
00FEE000	00FEE0FF	I/O	8	3	3
00FFBF20	00FFFF1F	RAM	16	2	2
00FFFF20	00FFFFE9	I/O	8	3	3

Begin	End	Attribute
00000000	0007FFFF	リード
00400000	005FFFFF	リード/ライト
00FEE000	00FEE0FF	リード/ライト
00FFBF20	00FFFFE9	リード/ライト

図5.4　シミュレータシステムの設定

シミュレータシステムにおけるメモリの設定を図5.4に示す．図3.6の物理メモリの配置を設定すれば良い．図5.4を正しく設定しないと メモリアクセスエラー が発生する．

(2) HEW接続シリアルモニタを用いたエミュレーション

PCとマイコンを接続し，HEWの デバッグの設定 において H8/300H Serial Monitor を選択するとシミュレータを使用する場合と全く同様にしてエミュレーションによりデバッグを行うことができる．セクションの設定は図5.3と全く同様で良い．シリアルモニタ使用時には割り込みベクタを本来はROM領域である0～0xFF番地に設定できるようになっている．したがってintprg.cにおける設定は変更する必要がなく，図5.3のメモリを全てRAM領域に配置すれば良い．この場合にリセット直後に表示した割り込みベクタ領域におけるメモリ内容を図5.5に示す．

またRAMのメモリ配置を先頭アドレスが0xFFBF20となるように変更した場合を図5.6に示す．実際にはRAM領域にマッピングされるが，表面上ROMの内容が書き換えられているように見えている．

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000000	00	40	00	00	7D	1E	EF	F0	5A	1B	10	10	F7	C8	F3	9B
000010	E8	12	C4	C8	76	D7	D7	37	44	42	90	B8	00	40	00	16
000020	00	40	00	18	00	40	00	1A	00	40	00	1C	00	40	00	1E
000030	00	40	00	20	00	40	00	22	00	40	00	24	00	40	00	26
000040	00	40	00	28	00	40	00	2A	83	88	0D	0E	EB	2B	EF	2F
000050	00	40	00	2C	00	40	00	2E	A4	28	48	01	00	40	00	30
000060	00	40	00	32	00	40	00	48	00	40	00	4A	C8	E4	48	D8
000070	00	40	00	4C	00	40	00	4E	00	40	00	50	CD	33	D0	E8
000080	00	40	00	52	00	40	00	54	00	40	00	56	D7	B9	FE	DF
000090	00	40	00	58	00	40	00	5A	00	40	00	5C	00	40	00	5E
0000A0	00	40	00	60	00	40	00	62	00	40	00	64	00	40	00	66
0000B0	00	40	00	68	00	40	00	6A	00	40	00	6C	00	40	00	6E
0000C0	60	F7	4D	00	25	D3	6D	56	00	11	43	26	E6	2E	A0	2B
0000D0	00	40	00	70	00	40	00	72	00	40	00	74	00	40	00	76
0000E0	00	40	00	78	00	40	00	7A	00	40	00	7C	00	40	00	7E

図5.5　RAM動作時における割り込みベクタ

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000000	00	FF	BF	20	7D	1E	EF	F0	5A	1B	10	10	F7	C8	F3	9B
000010	00	00	11	70	00	00	11	C8	00	00	12	1C	00	FF	BF	36
000020	00	FF	BF	38	00	FF	BF	3A	00	FF	BF	3C	00	FF	BF	3E
000030	00	FF	BF	40	00	FF	BF	42	00	FF	BF	44	00	FF	BF	46
000040	00	FF	BF	48	00	FF	BF	4A	83	88	0D	0E	EB	2B	EF	2F
000050	00	FF	BF	4C	00	FF	BF	4E	A4	28	48	01	00	FF	BF	50
000060	00	FF	BF	52	00	FF	BF	68	00	FF	BF	6A	C8	E4	48	D8
000070	00	FF	BF	6C	00	FF	BF	6E	00	FF	BF	70	CD	33	D0	E8
000080	00	FF	BF	72	00	FF	BF	74	00	FF	BF	76	D7	B9	FE	DF
000090	00	FF	BF	78	00	FF	BF	7A	00	FF	BF	7C	00	FF	BF	7E
0000A0	00	FF	BF	80	00	FF	BF	82	00	FF	BF	84	00	FF	BF	86
0000B0	00	FF	BF	88	00	FF	BF	8A	00	FF	BF	8C	00	FF	BF	8E
0000C0	60	F7	4D	00	25	D3	6D	56	00	11	43	26	E6	2E	A0	2B
0000D0	00	FF	BF	90	00	FF	BF	92	00	FF	BF	94	00	FF	BF	96
0000E0	00	FF	BF	98	00	FF	BF	9A	00	FF	BF	9C	00	FF	BF	9E
0000F0	00	FF	BF	A0	00	FF	BF	A2	00	FF	BF	A4	00	FF	BF	A6

図5.6　メモリ配置を変更した場合の割り込みベクタ

6．むすび

電気電子工学科3年生の実験科目である情報・通信実験において今年度から新設し，実施したPBL項目である「H8/3069Fマイコンを使用した電子回路の設計と製作」の立ち上げおよび実施結果について報告した．また回路の中心的な役割を担うH8/3069Fマイコンを使用する上でのノウハウを整理した．

今年度初めて実施したので数々の不測の事態や，課題が見つかり，また結果として多くの知見が得られた．来年度は，これらの結果をベースにしてより良い方向に持って行けるものと期待している．

[謝辞]

電気電子工学科の高橋　久教授にはHEW接続シリアルモニタの件でいろいろとお骨折りを頂いた．深く感謝致します．
またルネサスエレクトロニクス株式会社の鹿取　祐二氏には同モニタをご提供して頂いた上に，当方の回路，プログラムのチェックまでして頂いた．ここに記して深く感謝の意を表します．

[参考文献]

1) ルネサス社，“H8/3069F-ZTATTM ハードウェアマニュアル”
2) ルネサス社，“HEW接続組み込み型モニタのカスタマイズ方法 readme.htm”
3) OKI電気，”M5117805Fデータシート”
4) ルネサス社，”H8S、H8/300 シリーズ C/C++コンパイラ、アセンブラ、最適化リンケージエディタ”
5) H8-MON@YTomo. WEB, “Renesas 組み込みモニタ構築［H8,Win］”