

初学者を対象にした組み込み開発学習教材の製作 - H8を搭載したマイコンボードYOSIMAX-3664の設計と開発 -

吉岡 正博* 小高 知宏* 白井 治彦**
黒岩 丈介*** 小倉 久和***

Development of the Teaching Materials for Embedded Instrument - Manufacturing a Microcomputer Board named "YOSIMAX-3664" that is Mounted H8 Processor -

Masahiro YOSHIOKA*, Tomohiro ODAKA*, Haruhiko SHIRAI**
Jousuke KUROIWA*** and Hisakazu OGURA*

(Received January 25, 2008)

We developed the microcomputer board named YOSIMAX-3664 (YM-3664). The processor of a YM-3664 microcomputer board is H8/3664. The YM-3664 has many input-and-output devices. For example, it includes the speaker, the data entry switch, the light-emitting-diode, the liquid-crystal-display, the servomotor, and the expansion input-output.

The purposes which developed YM-3664 microcomputer board is to use for the teaching materials for beginners. In addition, the YM-3664 also can be used as an embedded development environment. This paper describes the feature and usage of YM-3664.

Key words : Engineering Education, Embedded System, H8/3664, H8/Tiny

1. はじめに

近年、携帯電話やハイテク家電など、いわゆる組み込み業界では市場の急速な拡大および開発期間の短縮や大規模化により、組み込み技術者の不足が懸念されている^[1]。また、新卒者の組み込み開発に対する意識も、つらい業界であるとか、ソフトウェア開発は経験があるがハードウェア開発の知識が無く不安といった、マイナスの印象が多い^[2]。

これらの問題を解決する一つの方法として、大学における組み込み開発の基礎教育が必要である。組み込み開発はハードウェアとソフトウェア双方の知識が必要であり、技術者教育としても非常に有意

義な分野であるといわれている^[3]。そこで今回、福井大学工学部知能システム工学科の実験IIにおいて、組み込み開発を題材とした実習を行いその教材として、初学者でも親しみやすい組み込み開発体験教材を開発した。教材の製作にあたり、ハードウェアはマイコンボードの形式で自主開発しYOSIMAX-3664 (以下YM-3664)とした。本ボードを用いることで、組み込み開発の学習ができ、組み込み機器開発の試作に用いることも可能である。このYM-3664は16ビットマイコンをメインとした学習ボードである。ボード上にはLEDやスピーカやスイッチといった、入出力デバイスを実装しており、開発から実行までをボード単体で行えるという特徴がある。これにより初学者であっても組み込み開発の流れを把握しやすくした。

本論文では、2章と3章でYM-3664の設計と製作について記し、4章で実際にYM-3664を用いた結果を記す。なおYM-3664の詳しい機能やファームウェアなどについては付録とした。

* 大学院工学研究科原子力・エネルギー安全工学専攻

** 工学部技術部

***大学院工学研究科知能システム工学専攻

* Nuclear Power and Energy Safety Engineering Course,
Graduate School of Engineering

** Dept. of Technology, Dept. of Engineering

***Human and Artificial Intelligent Systems Course,
Graduate School of Engineering

2. YM-3664の設計仕様

YM-3664の主な設計方針は以下の3点である。

- ▶ 初学者を対象にした組み込み開発教育における、親しみやすい教材であること。
- ▶ 基礎学習後さらに応用したい場合にも、対応できる様な拡張性を持つこと。
- ▶ 教育以外の利用として、汎用のマイコン開発ボードの機能を有すること。

そして、これらの条件を満たすために、具体的な仕様を決めた。

まず、親しみやすさの面として直感的にわかりやすいインタフェースを採用するべきと考え、ボード上にLEDと液晶とスピーカを実装することを検討した。さらに手で触ると変化が起こるインタフェースとして、温度センサやつまみ入力の実装を考えた。

開発におけるソフトウェア的な配慮としては、開発言語が問題となる。組み込み開発で一般的な言語はC言語であるため、C言語で開発できるCPUが必須条件となる。その上で、開発環境はフリーソフトウェアの形で手に入り、環境構築の容易なものを検討した。さらに、ハードウェアを直感的に制御できるような関数も多数用意した。これにより、組み込み開発の初学者であっても、ごく初歩のC言語の知識さえあれば、ひととおりの機能を試すことができるようにした。

次に、基礎の学習が終了し応用へ進む場合にも、同じボードで対応できるような工夫を検討する。これはボード上のスイッチやスピーカ等で使用していない入出力ピンは全て拡張入出力として、拡張ピンの形式で実装することで実現可能である。これにより、YM-3664と外部機器を接続することでボードに実装されていない機器の制御も行うことが可能である。

最後に、教育用途以外の使用方法として、インタフェースを実装済みの汎用マイコンボードとしても機能するように設計するため、入出力の数を増やすことを検討した。ボード上に数個の入力スイッチ、出力として数個のLEDとスピーカ等を実装し、外部拡張インタフェースも実装すれば汎用の入出力としては十分な数と考えた。

また中心となるCPUモジュールはコネクタによる取り外しを可能とすることで、YM-3664上で動作を確認した後に、別システムにそのまま組み込むことができる考えた。このような設計にしておくと、組み込み機器の試作時にその都度試作機を製作する必要が無く、そのままYM-3664を使うか、あ

るいは追加回路のみ製作すればよいため開発期間の短縮になると思われる。

以上の3点の方針を満足する設計について、その詳細をハードウェアとソフトウェアに分けて以下に述べる。

2.1 ハードウェア

メインとなる処理部にはルネサステクノロジー社の16bit CPUであるH8/3664¹⁴⁾を用いた(図1)。

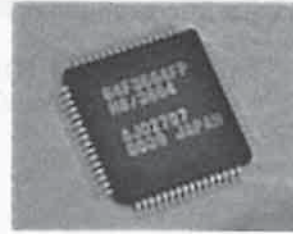


図1 ルネサス テクノロジー製 H8/3664

H8シリーズは採用実績も多く参考資料の入手性も良い。また開発言語は一般的なC言語に対応しており、開発環境もフリーウェアで対応できるためコスト面で有利である。このCPUは図2の様に秋月電子から扱いやすいモジュール¹⁵⁾として販売されている。これは図1のような表面実装部品で提供されるH8/3664を、扱いやすい汎用ピンの形式に変換し、さらに水晶発振子とシリアル通信用の電圧レベル変換チップを搭載した商品である。

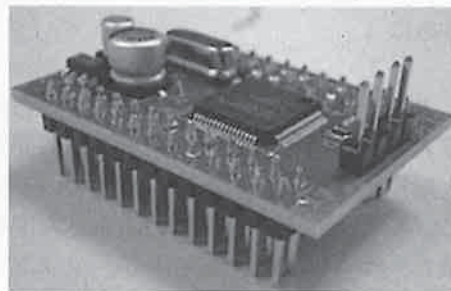
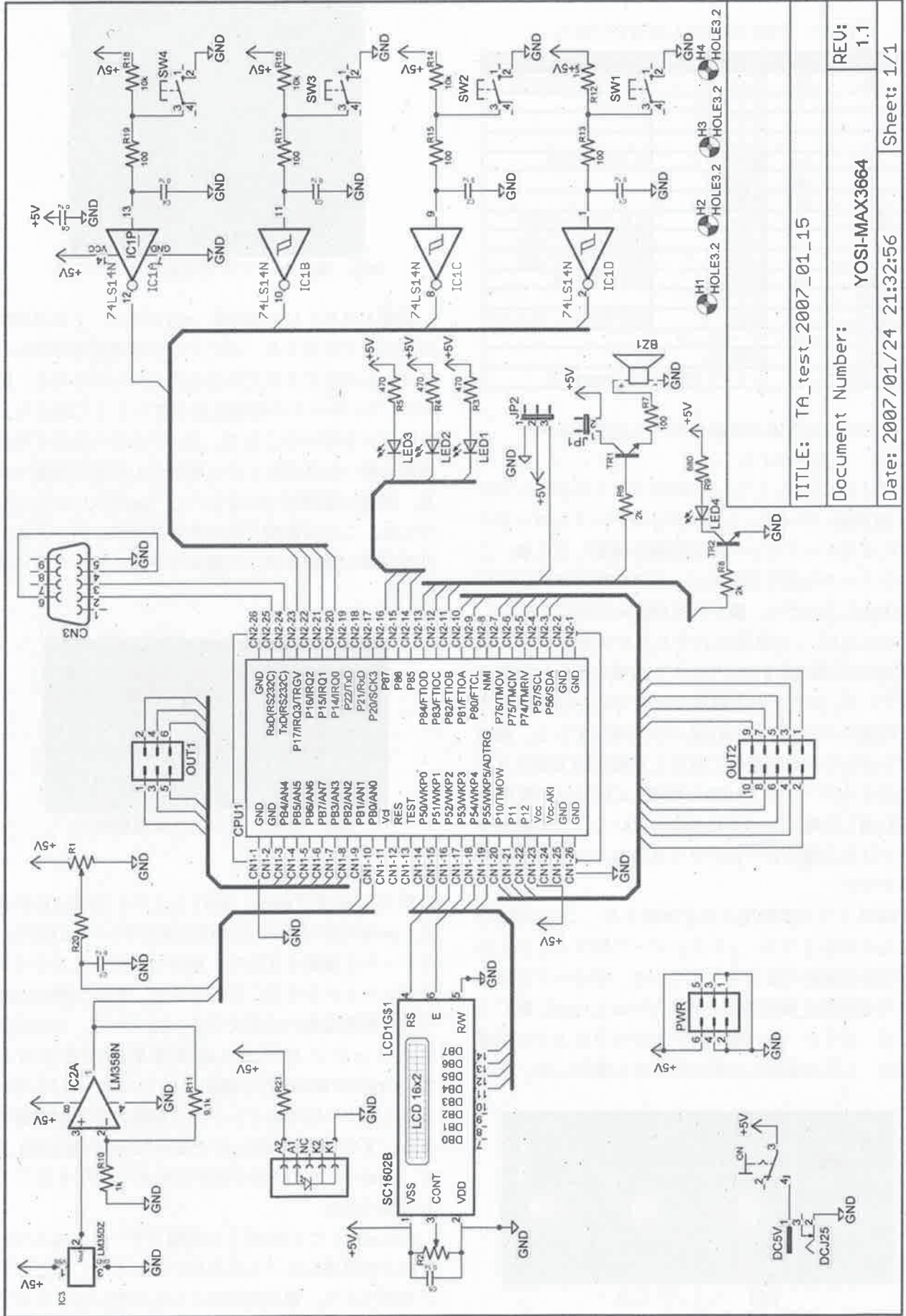


図2 秋月電子製 H8/3664モジュール

YM-3664はこのCPUモジュールに対応するマザーボードの形式で図3の回路図のように設計した。構成要素となる入出力デバイスは表1にまとめた。



TITLE: TA_test_2007_01_15

Document Number: YOSI-MAX3664

Date: 2007/01/24 21:32:56

REV: 1.1

Sheet: 1/1

図3 YM-3664の全回路図

表1 YM-3664の入出力デバイス

IN/OUT	module	element	備考
IN	in0	スイッチ1	タクティルスイッチ
-	-	スイッチ2	-
-	-	スイッチ3	-
-	-	スイッチ4	-
IN	in1	温度センサ	オペアンプ併設
IN	in2	可変抵抗	
IN	ext0	拡張入力	
OUT	out0	LED1	発光ダイオード(赤)
-	-	LED2	発光ダイオード(黄)
-	-	LED3	発光ダイオード(緑)
OUT	out1	LED4	PWM動作可能
OUT	out2	ブザー	
OUT	out3	液晶	16文字2行バックライト
OUT	out4	サーボモータ	
OUT	ext1	拡張出力	
IN, OUT	sci0	シリアル通信	D-SUB9pinメス

それぞれの要素の詳細を以下に記述する。

2.1.1 入力デバイス

はじめにin0として、4つのタクティルスイッチがある(図4)。シュミットトリガインバータとコンデンサによるチャタリング除去回路も実装したため、このスイッチを押すとCPUにパルスが出力されCPUではHigh入力となり、離した場合Low入力となる。

次にin1として温度センサ入力を用意した。これは図5の左端のIC3というシルク印刷の部分に実装されている、IC化された温度センサであるLM35DZ^[6]と汎用オペアンプのLM358^[7]から成っている。温度センサからは周辺温度に依存した電圧出力があり、これをオペアンプで10倍に増幅しCPUのA/D変換回路に適した電圧レベルに変換している。この温度センサは出力電圧がリニアライズされており非常に扱いやすい。

in2として可変抵抗入力を実装した。これは図5の右上にあるような、+ドライバーで回すことにより抵抗値を連続可変するものである。本ボードではこの可変抵抗を補助抵抗を挟んでVccとGndに繋ぐことで、ボリュームの様な使い方ができるように配慮した。これは連続した変化をさせる場合に用いる。

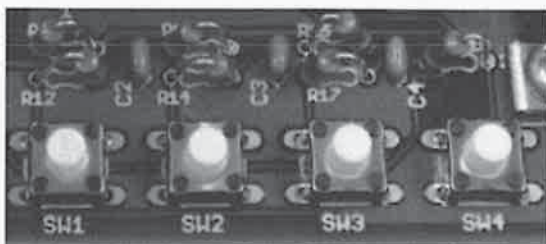


図4 スwitch入力

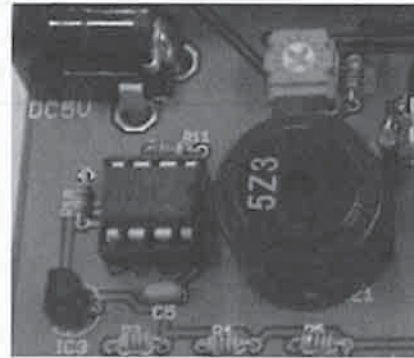


図5 温度センサと可変抵抗とスピーカ

拡張入出力としてはext0, ext1がある。これはCPUの入出力ピンのうち、ボード上の構成要素で使われてないものをできるだけ引き出したものである。図6のようにボードの背面に汎用ピンとして設けた。このピンを用いることで、ボード上のみならず外部回路も用いた応用をしたい場合にも対応が可能である。同様に電源も+5Vを2ピン、Gndを4ピン出力している。これは拡張入出力を外部回路に用いる際、外部回路の電源を別に用意する手間を省くために設置した。

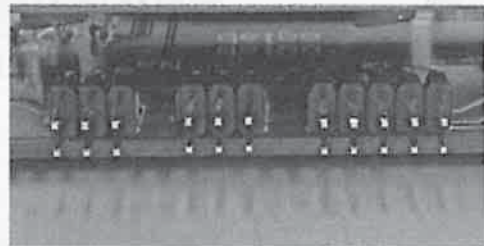


図6 拡張入出力と外部電源出力

2.1.2 出力デバイス

出力にはまずout0, out1として4つのLEDがある。out0は図7のような汎用に扱える3つのLEDで、スイッチと連動させたり、組み合わせることでイルミネーションさせることができる。そして図8がout1であり特殊な扱い方ができるLEDである。このLEDはトランジスタによる高輝度駆動回路を介して、CPUのPWM出力と接続されている。out1を用いることによりLEDのインバータ駆動の学習が可能である。またPWMを用いない場合は汎用のLEDとして、out0の3つのLEDと組み合わせて使用することも可能である。

次にout2として図5右下の電磁ブザーによるスピーカ出力を用意した。これを用いることで、電子ピアノを作ったり、警告用のサイレンに用いることができる。応用製作にエンターテイメント性を加えるこ

とが可能となり、初学者に親しみやすい環境が期待できる。



図7 汎用LED3個とサーボモータ出力ピン



図8 インバータ駆動用LED

標準出力としてout3には液晶を実装した。この液晶は16文字x2行の計32文字表示が可能で、文字は半角カナと英数字を表示できる。その他にも8文字まで自作データを表示させることができる。製品ではコピー機やレジキャッシャー、自販機などのあらゆる製品に組み込まれており、安価で採用実績の多い表示素子である。本ボードではこの液晶を用いて、メッセージを表示したり、デバッグの際の補助に用いる。



図9 16文字x2行表示の液晶

最後の出力要素としてout4のサーボモータがある。これは3ピンの端子で、順に信号出力、+5V出力、Gndとなっている。市場には主に2種類のピン配置のサーボモータが販売されており、本ボードでは向きを変える事でどちらにも対応している。このサーボモータ出力により、サーボモータをライトレーサ等の操舵に用いたり、ロボットの関節駆動の学習用に用いることができる。

その他に特殊な要素として、sci0というシリアル通信要素を実装した。これは図10のような9ピン汎用D-subメスコネクタと、CPUモジュール上に実装

された電圧レベル変換ICから成る。クロス開発時にはこのコネクタを用い、開発以外ではシリアル通信のコネクタとして利用できる。これによりコンピュータとRS-232Cでのシリアル通信を行ったり、外部回路との通信を行うことができ、複雑な制御にも対応可能である。



図10 シリアル通信用コネクタ

以上がYM-3664の全入出力デバイスの詳細である。これら要素を駆動させるために、電源は安定化された5V単一電源が必要となる。電源容量はサーボモータ使用時には数A必要になることもあるが、サーボモータを使わない限り数100mAあれば全機能を動かせるのに十分である。以上の構成要素を図11の様に、CPUを中心とした入出力要素として初学者に理解しやすくまとめた。組み込み開発で重要なのは、開発しようとするシステムにどのような入力と出力があるのか、その働きや関係を把握することである。YM-3664の開発にあたり、入出力を明確にすることで、表1のような入出力を図11の様に把握して開発ができるように工夫した。

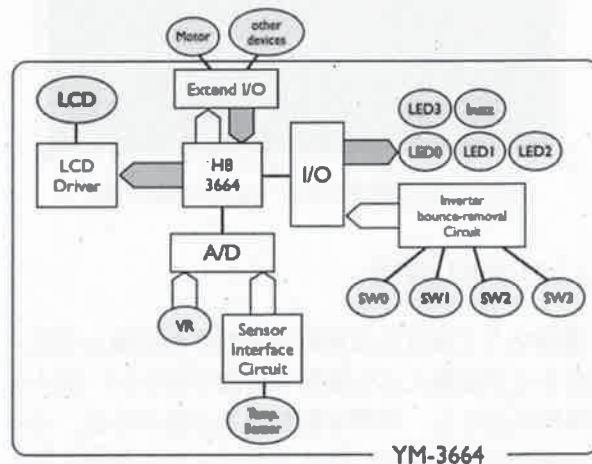


図11 YM-3664の構成図

2.2 開発環境

開発言語はアセンブリ言語かC言語を用いる。本ボードでは基本的なC言語の素養があれば、全機能

を扱えるよう設計した。細かい修正などはアセンブリ言語で対応可能であるが、基本的にはC言語での開発となる。

開発の流れは図12のとおりである。まずC言語でソースとなるプログラムを記述し、コンパイラでアセンブリ言語に変換される。次にアセンブラにより機械語に変換され、リンカによりその他のファイルと結合され実行可能ファイルが完成する。

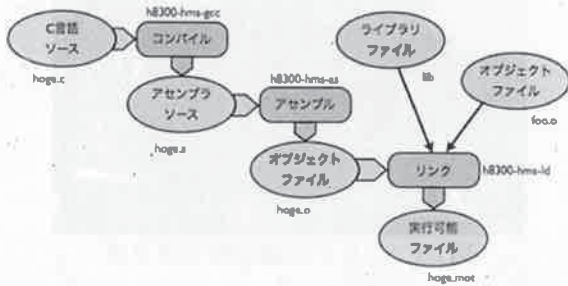


図12 開発フロー

この一連の開発を行うのが、開発環境である。今回は、実験で用いる際、OSとしてLinuxを用いる必要があったため、GPLライセンスで無償配布されているgccツールスイートを用いた。Windowsを用いる場合には、図13であげているGCC Developer Lite[®]のようなGUI環境も用いることができる。



図13 開発環境 GCC Developer Lite

3. YM-3664の製作

教材として使うには複数台の製作が必要となる。数台なら手配線による製作も可能であるが、数十台の規模になると、専用基板を作る必要がある。よって本製作に移る前に、まずユニバーサル基板を用いて図14のような試作をした。この試作により、大体の構成要素のテスト動作を検証する事ができた。

次に試作した回路を回路CADにて、電子図面として記述した。専用基板を作る際、訂正箇所があると

その都度コストと期間がかさむため、完成度を高めるための作業である。

そして、電子図面が完成したらモックアップとして、専用基板を印刷したものをボール紙に貼付け、実際に部品を仮実装する作業を行った。これにより、部品間の干渉などのミスを防ぐことができた。

最後に基板製造会社への発注を行った。届いた基板が図15のものである。そして、全部品を実装して完成したものが図16である。

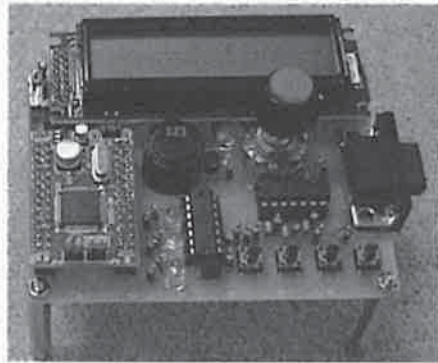


図14 YM-3664の試作版

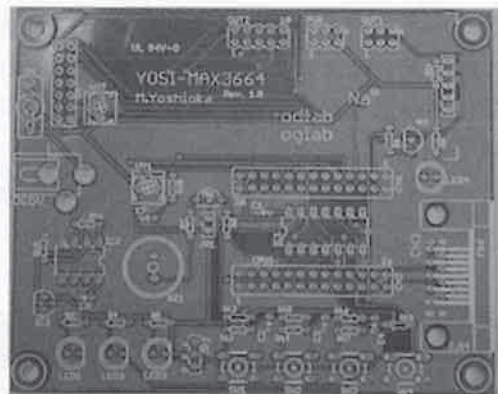


図15 YM-3664基板

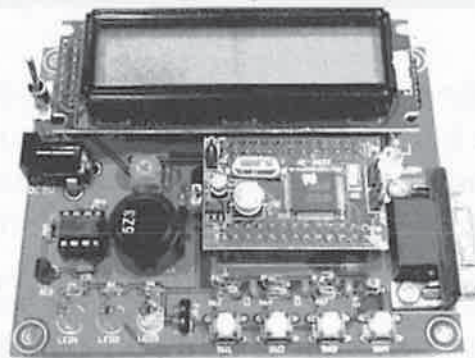


図16 YM-3664完成版

4. YM-3664の適用例

YM-3664には、CPUにH8/3664を搭載し、シリアル通信、LEDやスイッチ等の各種I/O、デバッグや表示用に用いる液晶が全て実装されており、組み込み開発の教育に用いる学習教材や、組み込み機器の試作に用いることが可能である。

本章では、学習ボードとして用いる例に、知能システム工学科の知能システム工学実験IIをとりあげた。また、試作に用いる例として研究室の入退室管理システムの試作例を示した。

4.1 学習ボードとして用いた例 (知能システム工学実験II)

福井大学工学部知能システム工学科2年生の実験「知能システム工学実験II -組み込み用マイコンによるリアルタイム制御-」に於いて図17のようにYM-3664を用いた実習を行った。この実習は、前年度まではPICマイコンによるアセンブリ言語による実習であったが、本年度からはYM-3664を用い、C言語による開発を行った。

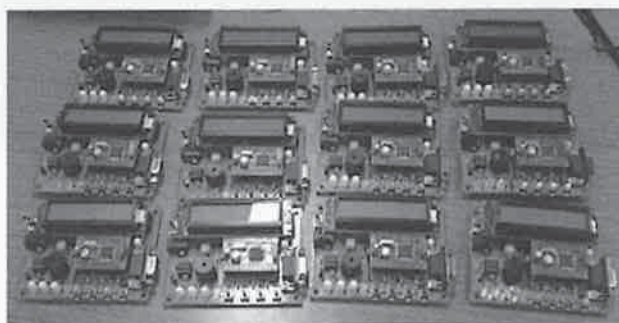


図17 実験に用いたYM-3664

実験内容は表2の通りである。各3時間ずつ第1週と2週に分かれており、第1週でまず基本を学び、第2週で応用として各自オリジナルの動作プログラムを組み込んだ製作を行う。

まず第1週では組み込み開発の背景について触れ、開発目的の理解をさせる。そして、各自のコンピュータ上で開発環境の構築をした。このコンピュータはOSがFedoraCore3というLinux系OSであるため、開発環境にはGNUライセンスのgccツールスイートを用了。開発環境が整ったところで、あらかじめ教材として用意しておいたサンプルプログラムの実行を含め実習に移る。

YM-3664のボード上には入力スイッチや温度センサ、LED、スピーカなどの入出力が実装されており、以上の(1)~(5)のような実習内容をYM-3664単体で行うことができた。また多種の回路を組み合わせることなくYM-3664単体で開発が進められると

いうことは、直感的に開発状況が理解できると考えられ、学生の組み込み開発への親しみやすさが向上すると思われる。

表2 実験内容

	内容
(1) デモプログラム	入力としてスイッチ1~4, 可変抵抗, および温度センサ, 出力として液晶, LEDおよびスピーカを用いる。各スイッチが割り込みの発生源となっており, スイッチに応じて動作するプログラムが切り替わる。プログラム内容は, 1つ目が可変抵抗の操作量に応じてLEDの輝度が高低するものであり, 2つ目が同様にスピーカから出力される音の音程が高低するものである。3つ目はLEDのイルミネーションであり, 最後が温度センサによる室温の液晶への表示である。
(2) 入出力	組み込み開発の基本は, どのような入力に大してどう出力するかである。ここではスイッチを入力としてLEDを出力とした簡単な環境に限定して, マイコンの入出力についてのプログラムを作成する。さらにスイッチのチャタリング現象についても各自調査をさせる。
(3) タイマ制御	組み込み機器ではタイマによる時間制御が必須である。例えばPWM機能を用いるにもタイマが必要であるし, 目覚まし機能を実装するにもタイマが必要となる。ここでは, LEDの点滅間隔をタイマで制御するプログラムを用意し, その間隔の変更をさせる。
(4) A/D変換	声や温度といった, 自然界に存在するあらゆるアナログ情報を組み込み機器で扱うには, いったんデジタル情報に変換する必要がある。ここではそれについて考察を含め, ボード上の温度センサをA/D変換して液晶に表示させるプログラムを各自で改造させた。さらにセンサを扱う上で重要となるオペアンプ回路についても, その動作を実際にデジタルマルチメータを用いて測定することで理解を深める。

	内容
(5) オリジナル	(1)~(4)をひととおり終えたあとに、応用として各自で考えた動作を盛り込むオリジナルのプログラムを作成させた。これは各自が仕様の決定からコード開発およびデバッグまですべての段階を体験するため、組み込み開発の一連の作業が簡単ではあるが理解できると考えている。

4.2 組み込み機器の試作に用いた例 (入退室管理システム)

研究室の入室と退室を管理するシステムを製作する際に、試作段階にてYM-3664を用いた。本システムは研究室内部に操作パネルを配置し、廊下に面したドア部分設置した表示BOXに内部の状況を表示するものであり、構成図は図17である。

今回は表示BOX部分 (図17白黒反転部分) の試作について記す。

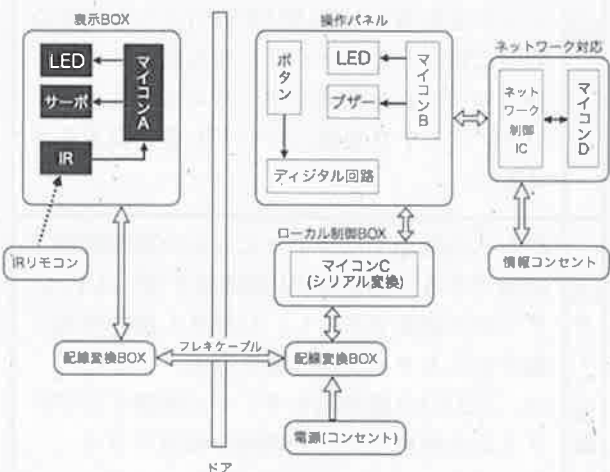


図17 入退室管理システムの構成図

図18が実際の開発の様子である。右奥に見えるのが表示BOXで、左手前がYM-3664である。

このような機器を開発する際には、その都度ユニバーサル基板やブレッドボードを用いて試作回路を製作するのが一般的である。だがYM-3664を用いる事により、ある程度の回路規模なら試作回路の製作を省く事ができる。もし回路規模が大きくなった場合でも、YM-3664には外部拡張が実装されているため、拡張部分だけ外付けの回路にすれば良い。今回の試作では、入力として数個のボタン、出力としてLEDとサーボモータ駆動ができれば良かったため、ボード単体で十分に開発できた。

また、デバッグに於いても、高価なデバッガを使わない限りは、コンピュータとシリアル通信を可能にする回路を特別に作り、コンピュータのコンソールでデバッグするのが一般的である。YM-3664には液晶が実装されており、簡単なデバッグならボード単体で十分可能であった。

最終的に試作が成功し、本製作に移る場合に於いても、YM-3664からCPUモジュールを抜いて、そのまま用いる事ができる。このように、組み込み機器の開発にはYM-3664が非常に便利である。

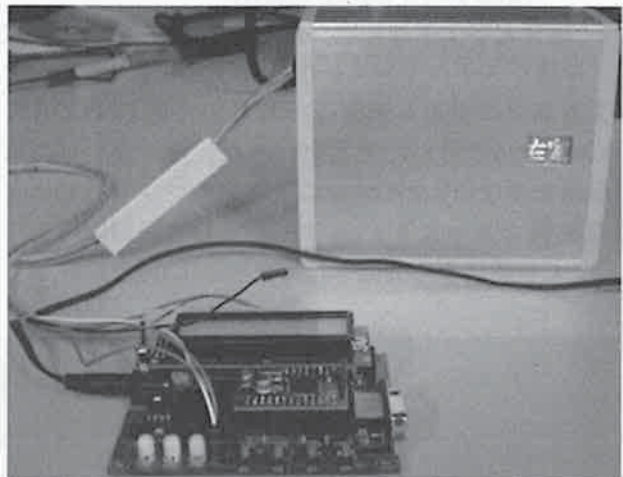


図18 同システムの試作状況

5. 考察とまとめ

初学者を対象にした、組み込み開発の学習教材として16bitマイコンを搭載したマイコン開発ボードを開発しYOSIMAX-3664と名付け複数台製作した。教育用としては15台を製作し、知能システム工学科に納めた(場所は総合情報棟I 3F ロボット工房)。このボードには複数の入出力が搭載されており、学習用に向いているうえに、汎用開発ボードとしても使用可能である。その実証として学部生の実験教材としての使用と、組み込み機器の開発をする場合でも、入退室管理システムの試作に、YM-3664を実際に使用し効果的であることを確認した。

YM-3664を用いた実験の考察として、今回は初めてであるため、教育効果としての蓄積はまだわずかではあるが第1週目のデモプログラムの動作においては楽しそうにスイッチ操作や可変抵抗を回し、どのような動きをするのか確かめている学生が多く見受けられた。また第2週目のオリジナルプログラムの作成では個々の学生が真剣に自分のプログラムを完成させ、確認とデバッグをしていた。その場の実習の雰囲気は良いものであった。使用できるライ

ブラリやシステムコール群を追加し、初学者により親しみやすい改良をしていきたいと考えている。また時間の制約もあり、割り込みの学習も含めたかったが、説明だけにとどめざるを得なかった。次回からは内容を練り直す必要がある。

また知能システム工学実験IIは4時限分の使用であり、簡単な実習のみでYM-3664の機能を十分発揮できなかった。今後、授業15時限分を使う機会があれば、組み込み開発の基礎から実習まで一環した教育が可能であると考えている。今期の実験における反応をまとめ、次期の利用に役立てると共に、より多くの研究を補助するためのマイコン開発環境として、本稿に加え詳細マニュアルを用意するなど、使いやすさの追求を続けていく予定である。

参考文献

- [1] 社団法人 組み込みシステム技術協会：
<http://www.jasa.or.jp/top/index.html>
- [2] 技術者を応援する情報サイト Tech-On：
<http://techon.nikkeibp.co.jp/>
- [3] 組み込みソフトウェア管理者・技術者育成研究会 (SESSAME)：<http://www.sesame.jp/>
- [4] 株式会社ルネサス テクノロジ：H8/3664グループハードウェアマニュアル(2006)
- [5] 株式会社秋月電子通商：AKI-H8/3664Fモジュール単品(通販コードK-159),
<http://akizukidenshi.com>
- [6] 株式会社National Semiconductor：LM35DZ
データシート
- [7] 株式会社National Semiconductor：LM358
データシート
- [8] 株式会社ベストテクノロジー：GCC Developer Lite, <http://besttechnology.co.jp>

付 録

C言語の初学者にも理解しやすい様、さらに組み込みに容易に馴染める様に、入出力デバイスの初期設定と動作は関数化し、入力の判定はマクロ定義した。少ないプログラム行数でデバイス操作が可能である。以下にそれらの一例をあげた。表3が入出力デバイスの設定関数例で、表4がそれを用いたプログラム例である。

表3 入出力デバイス設定関数

	関数名	引数	機能
[1]	in0_init()	引数なし	SW1, 2, 3, 4を使用可能にする
[2]	out0_init()	引数なし	LED1, 2, 3を使用可能にして消灯で待機
[3]	out1_init()	引数なし	ブザーを使用可能にする
[4]	out2_init()	引数なし	LED4を使用可能にして消灯で待機
[5]	wait(int ms)	引数なし	時間待ち
[6]	led1_on() と led1_off() led2_on() と led2_off() led3_on() と led3_off() led4_on() と led4_off()	引数なし	LED1,2,3,4を点灯/消灯する
[7]	buz_on() buz_off()	引数なし	ブザーを鳴動/停止する
[8]	SW1_ON と SW1_OFF SW2_ON と SW2_OFF SW3_ON と SW3_OFF SW4_ON と SW4_OFF		SW1,2,3,4の状態を判断する
[9]	LCD_init(16)	引数なし	液晶を使用可能にする
[10]	LCD_disp(unsigned char* str, int line)	str : 表示させたい内容 line : 液晶の表示行, 1か2	液晶に文字列を表示
[11]	LCD_char(unsigned char dat)	dat : 表示させたい内容	液晶に1文字表示
[12]	LCD_locate(int line, int col)	line : 行を指定, 1か2 col : 列を指定 0~15	液晶の表示位置を指定

表4 プログラム例

	動作	プログラムコード	説明
[1]	もしSW1がonなら LED1を点灯させる	<pre> out0_init(); in0_init(); while(1) { if(SW1_ON) { led1_on(); } } else led1_off(); </pre>	SW1とLED1を使用するため、in0とout0を初期化する。 無限ループで、SW1がONのときだけLED1をONにする。

	動作	プログラムコード	説明
[2]	LED4(out2)を500msおきに点滅させる	<pre>out2_init(); while(1) { LED4_ON; wait(500); LED4_OFF; wait(500); }</pre>	LED4を使用するため、out2を初期化する。無限ループで、LEDのONとOFFを500msごとに行う。
[3]	液晶に文字"A"を表示	<pre>LCD_init(16); unsigned char buf = 'A'; LCD_char(buf);</pre>	液晶を使用するためLCDを初期化する。LCDにAと表示する。
[4]	液晶1行目の3列目からHALと表示	<pre>LCD_init(16); LCD_locate(1, 3); LCD_char('H'); LCD_locate(1, 4); LCD_char('A'); LCD_locate(1, 5); LCD_char('L');</pre>	液晶を使用するためLCDを初期化する。LCDの1行目の3列目からHALと表示する。
[5]	液晶1行目に文字列を表示	<pre>LCD_init(16); LCD_disp("キョウハ10ガツデス, 1");</pre>	液晶を使用するためLCDを初期化する。LCDの1行目から文字列"キョウハ10ガツデス"を表示する。
[6]	液晶に温度センサの値を表示	<pre>unsigned int value; unsigned char temp[] = "0"; LCD_init(16); LCD_disp("Temperature", 1); LCD_disp(" /100°C", 2); while(1) { while((ADCSR & 0x20) == 0x20); value = (ADDRB>>6) * 4.4; itoa(temp, value); LCD_disp(temp, 2); ADCSR = 0x20; }</pre>	液晶を使用するためLCDを初期化する。無限ループで温度センサの値をA/D変換にて取得し続け、液晶に表示する。

