

# AVR マイコンを用いた RFID 制御装置の開発

Development of RFID Control System using an AVR Microcomputer

三屋 恵一郎  
MITSUYA Keiichiro

## 1. はじめに

平成 19 年度、東京硝子製品協同組合の委託により「RFID を用いた購買管理装置」の試作を実施した。平成 20 年度はこの試作機の改良とともに製作方法をマニュアル化する取り組みを行った。

平成 19 年度の開発では、組込み Linux ボードに接続されたリーダ/ライタから IC カードの ID 情報を読み取り、スキャン履歴データとして保存できるシステムを制作した。制御装置が TCP/IP ネットワークに接続されていれば、履歴はブラウザからモニタリングすることができ、必要に応じてデータベースを参照し業務に関連付けられたデータとして活用可能である。しかしながら、このシステムは制御部にレーザーファイブ株式会社（平成 20 年 4 月 1 日ターボソリューションズ株式会社に社名変更）の L-card+ を採用していたため、当初よりコスト高が懸念されていた。事実、普及啓発のためのセミナーや企業教育の実施時にもコストに対する問題が数多く指摘された。さらに、システム構成が有線 LAN を前提としていたためケーブルの引き回しや電源の確保等、設置場所に制限が課せられるという問題もあった。

そこで今年度、装置の「コストダウン」とともに「ロケーションフリー」への取り組みを行うことになった。具体的には制御部にマイコンを使うことで価格を下げ、無線機器の採用とバッテリー駆動を実現することで設置条件を緩和させる。さらに開発・導入方法をマニュアル化することで学生の課題実習や企業内教育セミナーの教材としてもより簡便に利用できるようにした。もちろん、装置開発の最終的な目的とするところは「作業者の労働生産性の向上」つまり作業の効率化を図る道具を製作することである。マニュアルは、その副産物として製作過程

のノウハウをまとめた教育教材であり、新技術の検証と導入を促す一助とするものである。以下、マイコンによる RFID 制御装置製作の技術的方法とマニュアルを教材として展開する際の指導手順の一例について紹介する。

## 2. マイコンによる制御装置の設計と実装

RFID の利用場面（製造・流通現場での部品棚での使用等）を考えたとき、制御装置はコンパクトであるほうがよいことは言うまでもない。この点、昨年度採用した L-Card+（小型の組込み Linux ボード）は PC 制御と比較した際の設置面積の点では十分有効であった。しかしながら、IC カードの ID の読み取りや記録だけのために組込み Linux ボードを用いるのはコストパフォーマンスが悪く、予算の制約から教育や運用のための十分な台数を導入できないという問題があった。そこで、マイコン制御の観点で特にプログラミングの容易性を重視し、検討を重ねた結果、Arduino（Atmel 社の AVR マイコン）を採用することになった。

Arduino を使用した制御装置は、低価格でハードウェアを導入することができ、さらに C 言語に似た構文で処理を簡易に記述できる。オープンソースで使いやすい統合開発環境も有しており、組込みソフトウェアの生産性の向上が大いに期待できるものである。

### 2.1 Arduino とは

Arduino とは、AVR マイコンの統合開発環境のことであり ATmega シリーズマイコンを使ったハードウェアとそれを稼働させるためのブートローダ及びマイコンソフトウェアの開発環境から構成されている。Arduino は PC のシリアルポートや USB ポートと接続して、データの入出力装置として稼働させるという利用形態

を想定して設計されている。Arduino を用いると外界（PC 内の仮想世界に対する現実世界）の物理的な状況を取り込み、判断・制御できるマイコンシステムを簡単に作ることができる。プリント基板の配線図やマイコンのブートローダのソースコードがすべて公開されているので、必要な部品を組み合わせることで自由に Arduino ボードを製作することが可能である。もちろん Arduino 仕様に沿った組み立て済みのマイコンボードを購入することもできる。

Arduino にはマイコン開発環境とともに PC 上で稼動する GUI の開発環境もあるので、対話的なオブジェクトを開発することも可能である。さまざまなスイッチやセンサから入力信号を取り込んで、LED やモーターを制御しながら状況をグラフィカルに PC 上で表現する、といったアプリケーションを作成することが比較的容易に可能となっている。

## 2.2 Arduino の特徴

### 【プログラミングが容易である】

Arduino はハードウェアとして AVR マイコンを搭載したマイコンボードを想定している。よってその開発手順は PC 上でマイコンソフトウェアを開発し、USB またはシリアルポートを介してマイコンのフラッシュメモリ上にプログラムを書き込むことになる。これによってマイコンは単独で処理を行うことが可能になる。これは、Arduino に限らず H8 マイコンや PIC マイコンといった他のマイコンでも同様な手順である。Arduino が特徴的である理由は、マイコンプログラミングにおいて簡略化した C 言語を使って開発することができるという点と、書き込みと実行がシームレスに連携しているという点である（図 1.参照）。これを可能にしたのが Arduino 統合開発環境 (IDE) と Arduino ブートローダ (AVR のブートローダ) の組み合わせにある。Arduino ボードを USB (あるいは RS-232C) ケーブルで PC と接続し、PC 上の統合開発環境のエディタにソースコードを記述して「Upload to I/O Board」ボタン (実行ボタン) を押せば、すぐにマイコンのフラッシュメモ

リ上にプログラムが転送され作成したアプリケーションが実行される。マイコン開発で通常必要とされるような、ホストコンピュータでのソースコード記述、コンパイル、プログラマを使っての Hex ファイルの書き込みといった手順が開発者の目に触れないように構成されている。

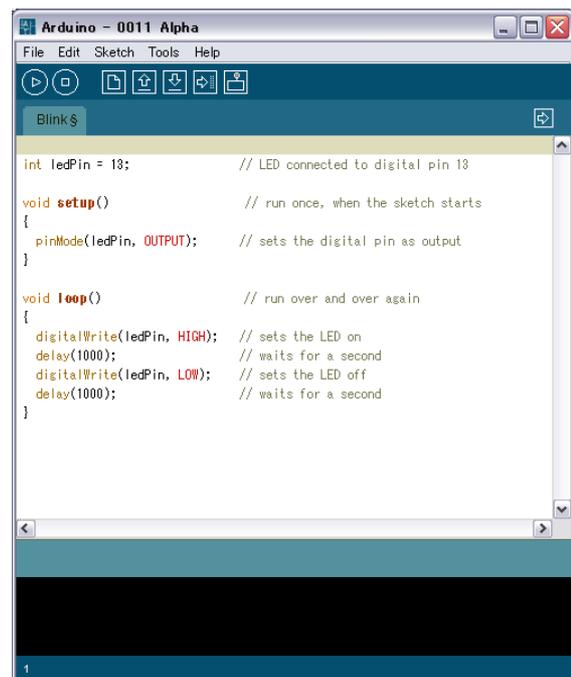


図 1.Arduino 統合開発環境画面

言い換えると Arduino の API がマイコンプログラミングの複雑な部分をパッケージングして、初心者にも容易に利用できるように提供されているということである。

### 【安価である】

あらかじめ組み立てられた Arduino ボードは ¥3,000 程度で購入することができ、他のマイコンプラットホームと比べて比較的安価である。また、部品から Arduino ボードを自作した場合の費用は ¥1,000 程度である。

### 【クロスプラットフォームである】

Arduino 統合開発環境は Windows、マッキントッシュ OS X、Linux OS 等、幅広いプラットフォームに対応している。

【PC のデータ処理環境が提供されている】

Arduino は AVR マイコンを容易にプログラミングできるとともに、PC 上でデータを処理（あるいは表現）するための GUI 環境も提供されている。PC 上のデータ処理プログラム開発環境は Processing と呼ばれている。この環境は、Arduino 統合開発環境と非常に親和性の高いものになっている。また、Adobe 社の Flash を用いてデータ処理を行うことも可能である。

【すべてがオープンソースである】

Arduino 統合開発環境は、ソースコードも含めて広く利用可能なツールとして公開されている。よって、独自の外部コードを追加したい時には、AVR の C 言語ソースコードを直接 Arduino プログラムに追加することができる。

### 3. Arduino ハードウェアの製作

Arduino 仕様のマイコンボードを製作して、RFID 制御回路を実装する方法について述べる。

#### 3.1 ブートローダの書き込み

最新（2008.04.01 時点）の Arduino 仕様に沿ったハードウェアを製作するために AVR マイコンシリーズの中から ATmega168 を選択した。購入したばかりの ATmega168 マイコンには Arduino ブートローダを書き込まなければならない。そのためには完全な仕様に従った Arduino が 1 台必要である。よって Arduino の製品版マイコンボードである Diecimila (USB 版 Arduino、完成品) を購入し書き込み器 (AVR ライタ) として使用した。また、Diecimila は、自作した Arduino ボードの妥当性の検証のためにも利用した。

#### 3.2 RFID 制御回路

公開されている Arduino 回路図 (RS232C シリアル接続用) をもとに図 2. のような RFID 制御回路を作成した。回路は、電源部、処理制御部、レベル変換部、外部インターフェース部、無線通信部の 5 つのモジュールにより構成されている。教材として製作する場合、これをブレ

ッドボード上に試作して (以下 Arduino クローンという。) 動作を確認する。

電源部は RFID リーダ/ライタに対して 9V、処理制御部とレベル変換部に対して 5V、無線通信部に対して 3.3V の直流電源を供給する。今回、主電源として 9V バッテリーを使用したため 3 端子レギュレータを用いて電圧変換を行っている。処理制御部には Arduino ブートローダを書き込んだ ATmega168 を使用している。RFID リーダ/ライタへのコマンド送信と戻り値の確認、ID の受信と EEPROM への書き込みを行うのが主な機能である。レベル変換部では ADM3202 トランシーバ IC を用いて TTL レベルと PC レベルのシリアル通信の電圧変換を行っている。外部インターフェース部はシリアル通信を行うために DSUB9 ピンのコネクタを使用している。RFID リーダ/ライタと接続するための雄コネクタと、PC のシリアルポートと有線で接続した状態でプロトコルをモニタリングするための雌コネクタの 2 つを有している。無線通信部では後述する ZigBee 規格の市販のモジュールを用いて ID 等 IC カードの情報をホストコンピュータへ送信する。RFID リーダ/ライタは株式会社ラステーム・システムズの RMF-1500R を採用している。

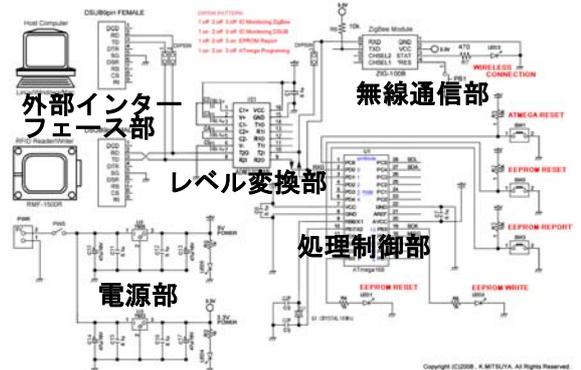


図 2. Arduino を用いた RFID 制御回路

### 4. Arduino のソフトウェアの制作

ブレッドボードに回路を試作後、Arduino 統合環境に用意されている LED 点滅プログラム等を実行し Arduino クローンの動作確認を行う。問題がなければ RFID 制御用ソフトウェアを実装するという手順になる。

#### 4. 1 LED 点滅プログラムの実装

Arduino クローンの動作確認方法は以下のとおりである。

- ① PC と Arduino クローンをシリアルケーブルで接続する。
- ② Arduino 統合開発環境を起動 → File メニュー → Sketchbook → Examples → Digital → Blink をクリックする。
- ③ ソースコードを確認した後 File → Upload to I/O Board をクリックする。直後に Arduino クローンをリセットする。
- ④ Arduino 統合開発環境のモニタ画面内に Binary sketch size 1096 bytes (of a 14336 byte maximum) とだけ表示されればコンパイルと書き込みは完了である。
- ⑤ Arduino クローン上の LED が 1 秒間隔で点滅していればハードウェアは正しく構成されている。

#### 4. 2 RFID 制御プログラムの実装

RFID 制御プログラムを作成する時に、欠かせない機能仕様として以下の 5 点が掲げられる。

- ① ATmega168 マイコンのシリアルポートにレベルコンバータを介して RFID リーダ/ライタを接続し、自動認識コマンドを送信することで IC カードの ID を読み取る。
- ② 自動認識コマンドの結果 (ID 含む) は無線通信でモニタリング用ホストコンピュータに送信する。
- ③ 自動認識成功 (戻り値 0x7D) であれば戻り値 (1 バイト) +ID (4 バイト) をマイコンの EEPROM に書き込む。
- ④ EEPROM に保存された ID はディップスイッチ切り替えにより無線通信でホスト PC に吸い上げる (有線シリアルポートはマイコンプログラムの書き換え時に使用する)。
- ⑤ EEPROM のクリアは任意のタイミングで可能にする。

EEPROM は 512 バイトの容量がある。よって、戻り値 (1 バイト) +ID (4 バイト) の形式でデータを保存した場合約 100 件のスキャン

履歴の蓄積ができる。この履歴はバッチ処理で管理システム側へ送信される。これによりハンディターミナルとしての活用が可能になる。

また、保守作業の効率化のためにオンボードでマイコンプログラムの書き換えが可能なるよう構成する。

#### 5. 無線ネットワーク

システムの設定条件の制限を緩和し運用の柔軟性を確保するために、システム管理用コンピュータと制御装置間の通信には無線機器を使用することにした。

##### 5. 1 ZigBee の利用

RFID は作業者がスキャンした ID の情報を蓄積する処理が大半であるため高速な処理性能は期待されない。また、製造・流通の現場で運用されることを前提としたとき、伝送距離も近距離対応の規格で十分である。こうした短距離通信の代表的な規格には Bluetooth や ZigBee がある。今回、省電力・低コストという点を評価し、無線ネットワークには ZigBee の規格を採用した。

具体的な製品として株式会社ベストテクノロジーの Zig-100B を使用した。当然、送信側と受信側のモジュールが必要である。受信側は待ち受け専用モード (Waiting モード) に、送信側は 1 対 1 通信を行うための Peer-to-Peer モードに設定する。さらに Zig-100B は 3.3V 系の電源電圧で駆動するため、5V で駆動する Arduino と同一基板上に実装する場合は、3 端子レギュレータを利用して電圧変換を行っておく必要がある (図 2.参照)。

#### 6. マニュアルの作成

製作過程はマニュアル化し教材として使用する。その際、業務の流れや背景等「前提条件」が必要になる。これには平成 12 年度から応用課程で利用されている標準課題を参考にした。業務の整理とともに技術的な条件等を付加して、RFID を利用した発注データ入力装置と購買管理システムとを連携させる課題を作成した。

## 6. 1 課題実習の背景

### 【業務の前提条件】

横浜電気産業株式会社(課題実施用仮想会社)の業務上の問題を解決するシステム開発を行う。対象企業の現状調査から業務要求分析を経てシステム設計及び実装・評価に至る流れを、開発会社に見立てたチームにより実践する(グループ学習方式)。業務分析時には、製造部品購買や一般用品購買等、多様な要件が抽出されることになる。ここで特に一般用品購買業務について着目する。用品棚に並べられた品目(図3.参照)の払い出し業務に、古くから現場で実践されている在庫管理手法の一つである、ダブルびん方式を適用させる。



図3.一般用品の用品棚の例

様々な工場で広く受け入れられ今も積極的に活用されている管理方式を、最新技術を使って置き換えることが課題の目的となる(図4.参照)。

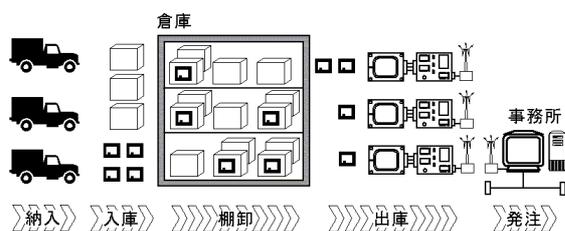


図4.ダブルびん方式情報システムの例

一連の開発過程を通して、生産現場の問題点をより身近に認識し新技術適用の際の業務条件とのすり合わせの難しさを理解する。あわせて、チーム開発におけるコミュニケーションの大切さを学ぶ。

### 【技術の前提条件】

ダブルびん方式と統合生産管理システムを連携する手段としてRFID(ICタグ)を用いる。具体的には、RFIDを利用した発注データ入力装置とそれを制御するソフトウェア及びデータベースと連携した上位システム(購買管理ソフトウェア)を開発する(図5.参照)。

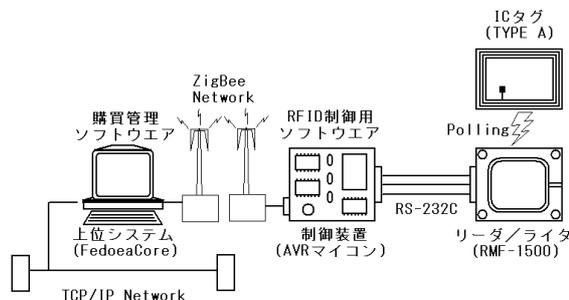


図5.RFIDを用いた購買管理システム

ハードウェア機器の製作、通信プロトコルの解析と実装、制御・管理ソフトウェアの実装はそれぞれ段階的に行う。開発言語はC言語(RFID制御プログラムに使用)とJava言語(データベース連携プログラムに使用)を用いる。特に発注データ入力装置の開発には、組み込みLinuxによる実装とAtmel社のATmega168マイコンを用いた実装の2通りの形態を用意する(図6.参照)。どちらのハードウェア構成を採用するかは、実習の目的に応じて判断する。

この課題実習を通じて、生産管理システムの設計、RFIDを利用した組み込み機器の開発、データベースシステムの実装方法を習得する。

### 【機器・開発環境】

RFIDリーダー/ライター:RMF-1500R(株式会社ラステーム・システムズ)

制御用システム:LinuxPC(Fedora8) or L-Card+(Laser5/Linux) or マイコン(ATmega168)

管理用データベースシステム:LinuxPC(Fedora8),PostgreSQL8.1.3

開発用言語:Java(GUI),gcc(通信),Arduino(マイコン)

ネットワーク:RS-232C(ZigBee),TCP/IP

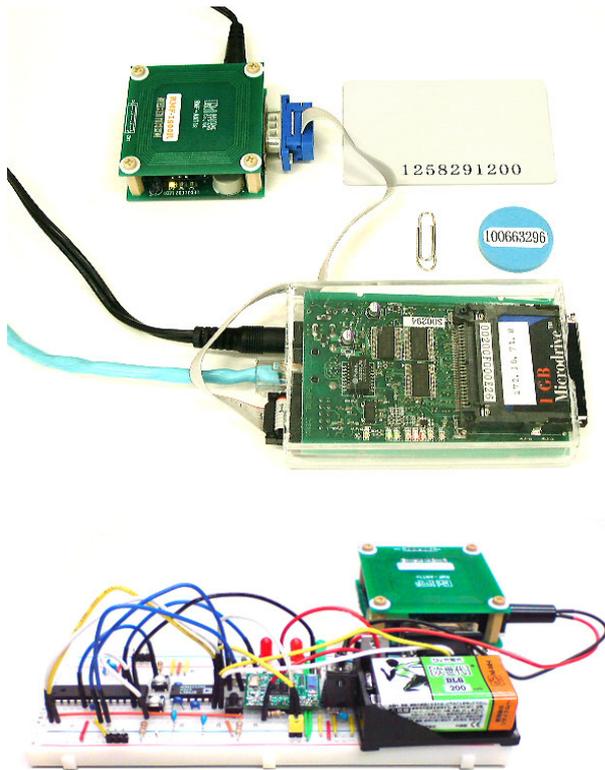


図 6.RFID 制御装置

上：組み込み Linux での実装例  
下：マイコンでの実装例（試作）

## 6. 2 システム構築の手順書

システム開発にあたって「何をどのような順番で実施しなければならないのか」を明確にした手順書を作成した。実習時にはこれをチェックシート形式で提供し、学生自身が進捗状況を評価しながら進めていく。

発注データ入力装置にマイコンを使用した場合の製作手順の一部を紹介する。

### 【ホスト PC ハードウェア構築】

#### ○開発用 PC の準備

Fedora8 DVD よりインストール実行

- ssh 接続確認  
win から Terminal 経由でログイン
- ftp 設定と送受信確認  
win から Terminal 経由で file の get と put (vsftpd,SELinux ポリシー)
- mail 設定と送受信確認  
OutlookExpress 等からメール送受信

dovecot(IMAP),sendmail(SMTP)

#### ○開発環境インストール

jdk6u6 インストールと gcc 確認

- javac 確認  
Terminal から javac コマンド
- java 確認  
Terminal から java コマンド
- gcc 確認  
Terminal から gcc コマンド

#### ○DB インストール

PostgreSQL8.1.3 インストール実行

- pg\_hba.conf 設定  
ネット経由で全ホスト全ユーザが全 DB へ接続可
- postgresql.conf 設定  
TCP/IP での接続可
- pg\_ctl start 起動確認  
Terminal より DB 起動
- pgAdminIII 接続確認  
win 版 pgAdminIII より mrp データベース作成
- アプリケーションデータベース作成  
Win 版 pgAdminIII より SQL 投入、設計済のテーブル+MRP テーブル作成

#### ○アプリケーション実行

MrpExec.jar を Win 側から起動して DB へのアクセス可能か確認

- 運用環境確認  
MrpExec.jar を Fedora 側から起動して DB へのアクセス可能か確認

### 【データ管理 PC 用ソフトウェア制作】

#### ○プロトコル解析

クライアントの WinPC で実施

- ID 読み込み確認  
MSComm サンプルプログラムを実行
- 自動認証コマンドの確認  
コントロール・通信電文一覧を参照  
送受信コマンドの確認

#### ○制御システム開発

基本は Fedora8 上での開発 暫定的に VMWare 上で開発してもよい

- send command 機能実装 (gcc)
  - 自動認証コマンド送信
  - (ネットでソフト検索)
- receive id 機能実装 (gcc)
  - ID 受信 (ネットでソフト検索)
- log write 機能実装 (gcc)
  - 日時+ID をログファイルとして保存
- 管理システム開発 (図 7.参照)
  - データベース&アプリケーション開発
  - DB 設計と実装 (SQL)
    - 品目関連情報の定義と DB 構築
  - log read 機能実装 (javac)
    - ログファイルの読み込み
  - StringTokenizer 機能実装 (javac)
    - ID と日付情報の分割
  - DataBase search 機能実装 (javac)
    - ID をキーに品目関連情報検索
  - item info 機能実装 (javac)
    - 検索結果を Terminal に出力
  - order update 機能実装 (javac)
    - 品目関連情報を発注データとして保存



図 7.発注管理システム例

- 管理システム開発 (Fedora8)
  - コントロール用 GUI 作成 (JavaSwing)
    - データ収集・表示画面等コントロール機能実装
  - order mail or print 機能実装 (JavaSwing)
    - ID 関連の品目情報を発注書として送信 or 印刷
  - end order 機能実装 (JavaSwing)
    - オーダー完了処理 (消しこみ)
  - バッチ処理機能実装 (JavaSwing)
    - EEPROM より一括送信されたデータの反映

- 運用システム開発
  - 運用システム導入 Windows 側 (Cygwin)
    - X 端末ベースシステムのインストール
  - X クライアントの設定 Linux 側 (xdm)
    - Win 側より X 端末を利用する設定
  - X サーバの設定 Windows 側 (CygwinXterminal)
    - Linux 側の X システムの利用確認

#### 【マイコンハードウェア製作】

- Arduino マイコンボード製作
  - Arduino 回路図を参照してブレッドボードに試作
  - 電源回路
    - 3 端子レギュレータによる 9V→5V→3.3V の電圧変換
  - マイコン回路
    - ATmega168 の Arduino 回路の製作
  - レベル変換回路
    - ADM3202 トランシーバ回路の製作
  - 無線モジュール回路
    - ZIG-100B 無線モジュールの製作
- Arduino・PC 間通信確認
  - 無線通信モジュール (ZigBee) の回路図を参照してブレッドボードに試作
  - ZIG-100B の設定
    - 速度、アドレス等設定
  - Arduino・PC 間通信確認
    - RFID 受信データを PC でモニタリング

#### 【マイコンソフトウェア制作】

- Arduino ソフトウェア制作 (図 8.参照)
  - Arduino 統合環境を利用してプログラムをマイコンにアップロード
  - LED blink 確認
    - File → Sketchbook → Examples → Digital→Blink
  - コマンド送信確認
    - RFID リーダ/ライタ認識コマンド送信プログラム及び結果受信プログラム
  - 受信文字列確認
    - RFID 受信データを PC でモニタリング

## ・EEPROM 使用

ID を EEPROM に保存・読み出し・クリアするプログラムを追加

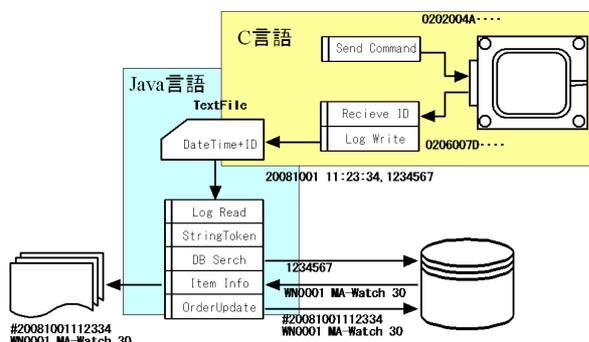


図 8.RFID 制御プログラム流れ図

## 7. まとめと今後の展開

平成 20 年度は、前年度製作した RFID 制御装置をもとに製作費の削減、運用条件の緩和、マニュアル化を実現した。

組込み Linux ボードによる制御からマイコン制御に変更したことで、機能はそのまま装置の全体価格は前年度のほぼ 3 分の 1 にすることができた。さらにコンパクトになり無線機器の使用で取り回しが良くなった。結果としてハンディターミナルとして活用することも可能になり適用業務の拡大が期待されるものとなった。一方で、制御装置全体のコストに占める無線関連部品の価格比率が相対的に大きくなってしまった。適用する業務の形態によっては無線が必要ないものもあると思われる。有線による PC との直結が可能な業務であれば、さらに費用は削減できるだろう。今後、RFID リーダ/ライタ部分の独自製作も視野に入れてさらなるコストダウンを図りたい。同時に、開発したシステムの運用現場でのロードテストを行い、結果を設計にフィードバックしたい。

製作過程をマニュアル化する作業は、実習時間配分の調整等運用上の細かい不備を修正しているところである。そのために、組込み機器開発へとシフトしつつある生産情報システム技術科の生産管理・DB 系の標準課題の教材として、現在試行適用中である。

課題に取り組む学生チームはおおよそ 6 人で 4 チームの構成となっている。システム設計の基盤となる業務モデルは平成 12 年度から標準カリキュラムに採用されている MRP システムを継承している。時代の推移により陳腐化している部分等を修正し、新たな技術条件を付加して業務分析モデルを作成した。このモデルに対してまずは DFD を用いて要件を定義し、ERD を用いてデータベースの設計を行う。ここまでは 6 単位の通常実習の時間内に行った。

次に標準課題における実装の第一段階として以下の作業要件 (PC 制御による課題) を設定した。

- ・ RFID リーダ/ライタと制御 PC 間のプロトコルの解析と実装を行うこと
- ・ PC から RFID リーダ/ライタを活性化し IC カードの ID を読み取ること
- ・ IC カードの ID と付加情報 (日時等) をテキストファイルとして PC に保存すること
- ・ ID をもとにデータベースを検索すること
- ・ 対象品目から発注データを作成すること

この要件の達成に、4 つ全てのグループが 3 単位の実習時間を費やした。標準課題は 10 単位であるため、おおよそ 3 分の 1 の時間であり、ここまでは想定スケジュールの範囲内に収まっている。今後、実習はマイコン制御の実現に移る。RFID 制御プロトコルの実装、無線通信の実装、EEPROM への ID 保存の実装にそれぞれ 1 単位程度の時間をかける予定である。データベースと連携した上位システムの実装は、装置製作と一部並行して 4 単位程度の時間を見積もっている。

製作物を目で見て、手で触って確認できるため、興味をもって実習に取り組む学生の姿が見受けられるのが心強い。最終的にアンケート調査等を実施し教材の評価を行う考えである。

## 謝辞

本装置の製作は東京硝子製品協同組合による平成 20 年度人材確保推進事業をもとに進めたものであり、ご協力いただいた関係各位に深く感謝いたします。

参考文献

- 1) 三屋恵一郎,「RFID を利用した購買管理システムの製作」,職業能力開発総合大学校東京校紀要,第 23 号, 2007 年,P96-P102
- 2) 三屋恵一郎,「平成 19 年度人材確保推進事業 製造・流通業における近接型非接触 IC タグの活用事例研究」,東京硝子製品協同組合,2007 年
- 3) IT 用語辞典 e-Words (ZigBee) :  
<http://e-words.jp/w/ZigBee.html>
- 4) IT 用語辞典 e-Words (Bluetooth) :  
<http://e-words.jp/w/Bluetooth.html>
- 5)Arduino オンラインマニュアル (最初の一步) :  
<http://arduino.cc/en/Guide/Introduction>
- 6) Arduino オンラインマニュアル (リファレンスガイド) :  
<http://arduino.cc/en/Reference/HomePage>
- 7) Arduino オンラインマニュアル (ハードウェア) :  
<http://arduino.cc/en/Main/Hardware>
- 8) Kimio Kosaka,「外付け AVR ライタ無しで Bootloader を書き込む」,  
[http://www.geocities.jp/arduino\\_diecimila/bootloader/index.html](http://www.geocities.jp/arduino_diecimila/bootloader/index.html), 2008.07.27