

中学生のためのマイコン制御学習用プログラム言語の設計と実装 Design and Implementation of a Programming Language for the Study of Microcomputer Control for Junior High School Students

谷岡 政宏 (北海道職業能力開発大学校)

Masahiro Tanioka

平成 24 年度から中学校の技術・家庭科ではプログラムによる計測・制御が必修となった。しかしながら、様々な制約の中で行わなければならないという現実がある。そのために開発された教材に浅田寿展氏の考案によるヒダピオシステムがあるがマイコン制御を学習するには市販のコンパイラをインストールするために OS の管理者権限が必要であり厳格に管理された中学校の教室での使用は困難であった。そこでテキスト処理言語 AWK によって新たにコンパイラを作成した。これは実際に中学校の授業で利用されるほか、小中学生を対象にしたものづくり教室等で電子オルゴール製作に使用される等、その有効性、実用性について実証されている。

キーワード：中学校、計測・制御、プログラミング、教材、言語処理系

1. はじめに

平成 20 年 2 月に告示され平成 24 年度から完全実施となった中学校の技術・家庭（技術分野）では「D 情報に関する技術（3）プログラムによる計測・制御」が必修となった¹⁾。しかし、技術科の教員不足、教具の購入予算がないなどの理由で実施率は十数パーセントに留まっていた²⁾。そのような状況の中、浅田氏によって「JA 制御ヒダピオシステム」が考案された³⁾。しかし、従来のヒダピオシステムでマイコン制御を行うためには市販の BASIC コンパイラを OS の管理者権限を使用してインストールしなければならないが、中学校では教室のパソコンは厳重に管理されており、教員といえども管理者権限を使用することは出来ない。そのことによって必ずしも情報の専門家ではない技術科を担当する先生方にとってマイコン制御を行うにはハードルが高く、マイコン制御の授業を行うことに躊躇し、ヒダピオシステムでパソコン制御の授業を行うまでに留まっている⁴⁾。

そこでインストールに OS の管理者権限が不要で簡単な BASIC のプログラムを機械語に変換するコンパイラをテキスト処理言語 AWK によって実装し、ヒダピオシステムから使用できるようにした。本研究で開発したコンパイラはヒダピオシステムに統合され実際に中学校での授業で使用されている他、小中学生を対象にしたものづくり教室での電子オルゴール作成に使用されており、その有効性、実用性について実証されている。

2. ヒダピオシステム

ヒダピオシステムとは、ハードに HIDaspX を、ソフトに「JA 制御ヒダピオ」を採用し、さらに「被制御器」

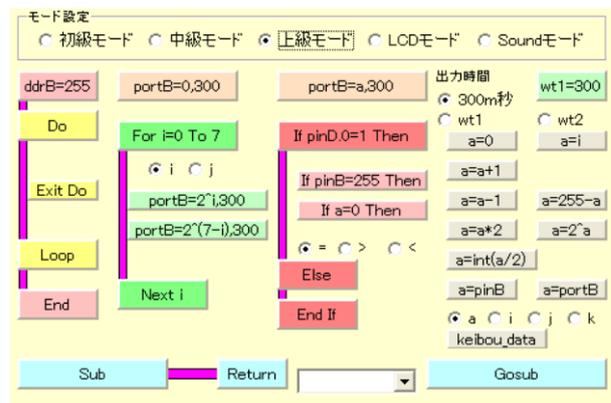


図 1 ヒダピオ簡易プログラミングでの主な命令

や「学習ノート」などの補助教材も準備し、簡単に USB-IO 制御やマイコン制御が楽しめる学習システムの総称である⁴⁾。JA 制御ヒダピオ（以下ヒダピオと呼ぶ）はパソコン制御とマイコン制御を学習するためのソフトウェアである。ここではマイコン制御に関連する簡易プログラミング、電光掲示板、電子オルゴールについて説明する。

2.1 簡易プログラミング

ヒダピオシステム上でプログラムを作成し、マイコンにプログラムを書き込むことでマイコン単独で制御を行うマイコン制御が行える。このプログラミングを行う機能をヒダピオでは簡易プログラミングと呼んでいる。簡易プログラミングはヒダピオでマイコン制御を行うためのプログラムを作成するサブシステムである。プログラムは簡単な BASIC 言語で記述する。使える命令は表 1 の通りである。プログラムの入力はマウスの操作で行える。

命令の選択肢は初級モード、中級モード、上級モード、LCDモード、Soundモードに分類されておりモード設定のグループボックスで選択することで使用できる命令の範囲が変わるようになっている(図1)。

初級モードはパラレルポート B への出力(PortB=)、パラレルポート B の入出力設定(ddrB=)、無限ループ(Do-Loop)、プログラムの終了(End)等の基本的な命令のみが使える。

中級モードでは初級モードの命令に加え指定回数の繰り返し(For)とポートの出力に 2^i 、 $2^{(7-i)}$ 等の制御変数を使った式が使える。制御変数としては、i と j が使える。

上級モードでは初級モード、中級モードの命令に加え、条件分岐(If)とサブルーチン(Gosub-Return)、変数と代入が使える。

LCDモードはLCD表示装置への出力に関する命令(LCD等)が使える。

Soundモードではスピーカ駆動用の矩形波を出力する命令(Sound)が使える。

表2 簡易プログラミングでの主な命令

文	説明	モード
ddrB=255	入出力方向レジスタの設定	初級
portB=0,300	ポートへの出力	初級
Do	無限ループの先頭	初級
Loop	無限ループの末尾	初級
End	プログラムを終了させる	初級
For i=0 To 7	指定された回数の繰り返し	中級
Next i	For による繰り返しの末尾	中級
portB=2 ⁱ ,300	ポートへの出力	中級
portB=2 ⁽⁷⁻ⁱ⁾ ,300	ポートへの出力	中級
Exit Do	Do-Loop のループを抜ける	上級
If 条件式 Then	条件による分岐(表1参照)	上級
Else	条件が成立しない場合の処理	上級
End If	If 文の末尾	上級
代入文	表3参照	上級
Sub ラベル名	サブルーチンのラベル指定	上級
Gosub ラベル名	サブルーチンの呼び出し	上級
Return	サブルーチンからの復帰	上級
Cls	LCD表示器の画面を消去	LCD
Lcd "文字列"	LCD表示器への文字列表示	LCD
Locate 1,1	LCD表示器の出力位置指定	LCD
Waitms 300	待ち時間の指定(ミリ秒単位)	LCD
Sound portB.0,282,189	音を鳴らす	Sound

表1 If文で使用できる条件式と意味

条件式	意味
pinD.0=1	入力ポート D のビット 0 が 1 に等しい
pinB=255	入力ポート B が 255 に等しい
a=0	変数 a が 0 に等しい
pinD.0>1	入力ポート D のビット 0 が 1 より大きい
pinB>255	入力ポート B が 255 より大きい
a>0	変数 a が 0 より大きい
pinD.0<1	入力ポート D のビット 0 が 1 より小さい
pinB<255	入力ポート B が 255 より小さい
a<0	変数 a が 0 より小さい

表3 簡易プログラミングの代入文

代入文	説明
a=0	変数 a に 0 を代入
a=i	変数 a に変数 i の値を代入
a=a+1	変数 a の値を 1 増やす
a=a-1	変数 a の値を 1 減らす
a=255-a	変数 a のビット反転
a=a*2	変数 a を左シフト
a=2 ^a	2 の冪を求める
a=int(a/2)	変数 a を右シフト
a=pinB	ポート B の入力値
a=portB	ポート B の出力値
(以下 j, k も同じ)	
i=0	変数 i に 0 を代入
i=a	変数 i に変数 a の値を代入
i=i+1	変数 i の値を 1 増やす
i=i-1	変数 i の値を 1 減らす
i=255-i	変数 i のビット反転
i=i*2	変数 i を左シフト
i=2 ⁱ	2 の冪を求める
i=pinB	ポート B の入力値
i=portB	ポート B の出力値
i=int(i/2)	変数 i を右シフト
(portB=で変数が使える)	
portB=a,300	出力値を変数で指定
wt1=300	待ち時間用変数
portB=0,wt1	待ち時間を変数で指定
portB=a,wt2	出力値と待ち時間を変数で指定

初級モード、中級モード、上級モードについては、USB-IO 制御装置を使ったパソコン制御とマイコン制御で共通のプログラムが使えるようになっているが、LCDモード、Soundモードではプログラムの作成はヒダピオ

から行うものの、実行にはコンパイルしてマイコンにプログラムする必要がある。尚、簡易プログラミングは最近流行っているブロックの組み合わせでプログラムを作成するビジュアルプログラミングではない。

2.2 電光掲示板

電光掲示板はマイコンに接続した 16 文字×2 行の LCD 表示装置にメッセージを表示させるプログラムを作成するサブシステムである。

登録した複数のメッセージをマイコンのポート入力によって切り替えることができ、スクロールや点滅を組み合わせた 8 種類の表示方法を選ぶことができる。

データの入力はマウスのみで行え、メッセージと表示の仕方を指定しボタンをクリックすることでマイコン用のプログラムが生成される。そのプログラムをコンパイルし、マイコンにプログラムする。

2.3 電子オルゴール

電子オルゴールは楽譜データを入力することで任意の曲を演奏することができる。ボタンをクリックすることでマイコン用のプログラムが生成される。そのプログラムをコンパイルし、マイコンにプログラムする。

3. 言語の設計

3.1 ヒダピオシステムからの出力とコンパイル

簡易プログラミングでは生徒が記述したプログラムを一旦、市販のコンパイラ用のプログラムに変換し、それを生徒がコンパイラを使って HEX ファイルを生成する。

従って、ヒダピオと連携するには、この市販のコンパイラ用に書かれたプログラムをコンパイルできるようにする必要がある。

表 4 簡易プログラミング以外の主な命令文等

文	説明
label:	サブルーチンやデータのラベル
Data 10,20,30,40	プログラム中にデータを置く
Read a	Data 文で置いたデータを変数に読み
Restore label	データを読み出す位置を指示する
a=Lookup(i,label)	指定されたデータ行から、指定された位置の数値データを変数に読み出す
\$regfile=	ターゲット CPU の指定
\$crystal=	CPU の動作周波数を指定
Config LCD=	LCD 表示装置の設定(ダミー)
Dim i As Integer	変数の宣言(i, j, k, x, a, wt1, wt2)
Config PortD=	入出力ポート D の設定(常に Input で入力とする)
PortD=	入出力ポート D への出力(常に &B11111111 でブルアップ)
Config PortB=	入出力ポート B の設定

また、電光掲示板や電子オルゴールからは市販のコンパイラ用のプログラムとして出力されるので、これらで使われている命令もコンパイルする必要がある(表 4)。

従来はそれらを BASCOM-AVR⁵⁾でコンパイルしていた。しかし、市販品であり、無料の Demo 版もあるが、教育用途での使用にはライセンスの購入が必要である。

また、インストールするには管理者権限を必要とするが、教室のパソコンは厳重に管理されており、教員といえども管理者権限を使ってインストールできない。さらに、統合開発環境となっているためコンパイルするには起動、ソースファイルの読み込み、コンパイルボタンのクリックといった操作をする必要がある。

しかし、中学生にとってはメニューやボタンによって余計な操作による混乱を誘発しやすいためコンパイルだけを行う目的には適さない。

3.2 開発目標

つぎの 3 つを開発目標とした。

- (1) インストール (に管理者権限が) 不要である
- (2) ヒダピオシステムの全ての機能が使える
- (3) 中学生でも簡単にコンパイルできる

4. 言語の実装

4.1 記述言語

本コンパイラの実装をするにあたり、処理系の記述言語について検討することにした。コンパイラを最も簡単に実装できると考えられる言語として AWK を選んだ。コンパイラのように複雑な処理をするソフトウェアの記述には通常コンパイラ型言語を使うが、大半が文字列処理であることからスクリプト言語が向いていると考えた。処理時間に関してはそれほど大きなプログラムを作ることには無いので問題にならないと考えている。近年様々なスクリプト言語が発表されているが、高機能になるほどパソコンにインストールするのが面倒である。AWK は実行ファイルをコピーするだけで使える。

AWK は行指向のテキスト処理言語で、あらかじめ記述されたパターンとアクションからなるルールの集合を与えると各行ごとに自動で処理される。各行は指定された区切り文字によって読み込み時にフィールドに分解される。

式の解析には正規表現によるパターンマッチングを使用し、記号表の処理には連想配列を使用した。

今回は AWK の実装の一つである GNU Awk 4.0.0 を使用した。Windows 上で動作するものはいくつか存在するが今回は Klabaster 版⁶⁾を使用した。

4.2 出力コード

本コンパイラからの出力であるオブジェクトコードは ATiny2313 のアセンブリ言語で、Atmel 社のアセンブラと同じ文法を採用した。アセンブリ言語を出力すること

で HEX ファイルの出力をアセンブラに任せることができ開発期間を短縮することができた。

AVRA の最新版である 1.3.0 を使用したが Windows 用のバイナリは配布されていないため MinGW 環境を使って自前でビルドした。

4.3 コンパイラの構成

コンパイラは字句解析部、構文解析部、意味解析部、コード生成部、最適化部で構成されるのが一般的であるが本コンパイラでは最適化は行っておらず、ソース・プログラムを 1 度だけ読み込んでアセンブリ言語を出力するまでを 1 つのプログラムで処理している。

後半の機械語を生成する部分はアセンブラを使用する。対象となる CPU は Atmel 社の ATtiny2313 である。

処理系として Atmel 社の純正アセンブラである AVRASM32.EXE と互換性のある AVRA 1.3.0 を使用した。これは、GNU GPLv2 で配布されているフリーソフトウェアであり、本コンパイラとともに配布することができる。

4.3.1 字句解析

トークンへの分解は AWK のフィールドに対応させる。代入文や式については空白文字を含まないことにし、正規表現により別途解析する。

4.3.2 構文解析

各文は、1 行の中で完結しており、第 1 フィールドからどの文であるかを判別することができる。キーワードの位置関係から構文を識別し、式などを含むフィールドは正規表現を使ったパターンマッチングを行って変数、演算子、定数などに分解する。

Do-Loop、For-Next、If-Else-End If 等の制御文の場合は対応関係の解決にスタックを使用した。

4.3.3 コード生成

文が識別された段階で対応するアセンブリ言語を出力する。

4.3.4 最適化

最適化は行っていない。

4.4 変数の処理

変数管理用の記号表として連想配列を用意し、変数名をキーに、その値として 1 を割り当てている。変数の登録は未登録の変数が表れたときか Dim 文で宣言されたときに行われる。

Dim 文で宣言された変数については型情報も登録する。

プログラムの最後で記号表に登録されているすべての変数を SRAM に割り当てる。

変数 a、i、j、k、x、wt1、wt2 は簡易プログラミング用として特別扱いされる。

4.5 ランタイム・ルーチン

AVRA は直接 HEX ファイルを生成するためソースレベルで 1 本のプログラムでなければならない。

そのためコンパイラからはランタイム・ルーチンも一緒に出力している。AVR 用演算ライブラリは赤松氏が公開されている⁷⁾ので、それを使用した。

4.6 コンパイルモードとコンパイルモードスイッチ

本コンパイラは簡易プログラミングでの文法を受理しアセンブリ言語を出力するネイティブコンパイラであるが、ヒダピオからは従来の BASCOM-AVR 用のソース・プログラムを出力する。また、電光掲示板や電子オルゴールのプログラムは BASCOM-AVR 用に作成されたものである。

従って、いくつかの BASCOM-AVR 用の命令文等も処理することができなければならない。

プログラム 1 に示す簡単な例からはプログラム 2 に示すプログラムが出力される。そこで、表 4 の命令文がコンパイルできるようにした。

プログラム 1 ヒダピオからの出力

```

プログラム 2
生徒の記述
'
ddrB=255
Do
    portB=1,300
    portB=0,300
Loop
End
    
```

```

$regfile = "Attiny2313.dat"
$crystal = 1000000
Config Lcd = 16 * 2
Config PortD=Input
PortD=&B11111111
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim x As Integer
Dim a As Integer
Dim wt1 As Integer
Dim wt2 As Integer
Config PortB=255
Do
    Portb=&B11111110
    Waitms 300
    Portb=&B11111111
    Waitms 300
Loop
portB=255
End
    
```

また、出力されたプログラムから分かる通りポート B に対する出力が反転している。これは、ヒダピオシステムで使用する標準的なハードウェアでは負論理を採用しているが中学生にとって 0 で LED が点灯し 1 で LED が消灯するのは分かりにくいからである。そこで本コンパイラでは HIDapio(ネイテ

イブ)モードと BASCOM-AVR(互換)モードの 2 つのモードを持ちポート B に対する出力を負論理にするか正論理にするかを切り替えられるようにした。また、この切り替えを表 4 の中の \$REGFILE=、Cong PortB=、Dim 等がプログラムの先頭にあることを見て行っている。HIDapio モードではプログラム 2 のプログラム中で行っている初期化処理と同等の処理をプログラムの先頭に挿入する。

4.7 電子掲示板への対応

ヒダピオの簡易プログラミングでは符号無し整数のみを扱う(BASCOM-AVRの Integer型にマッピングされるが符号は考慮されていない)ことからデータ型として 16 ビット符号なし整数のみを考えている。ところが、ヒダピオが出力する掲示板のプログラムでは文字データを Data 文により 1 文字ずつプログラムメモリ上に配置するようになっている。

AVRA にはプログラムメモリ上にはワード単位でしか配置することができない制約があり、このままでは BASCOM-AVR に比べ 2 倍の容量を必要とし、プログラムによっては 2KB しかない ATtiny2313 のプログラム用 Flash メモリが溢れてしまう。

そこで、Data 文ではバイト型データの時は一度バッファに記憶し、End 文または、ラベルが出現することによってデータの終わりと判断しコードを出力する。

4.8 電子オルゴールへの対応

ヒダピオが生成する電子オルゴールのプログラムでは 32bit の除算と文字列処理関数が使われているが本コンパイラはこれらの機能を持っていないのでコンパイルすることはできない。そこで、本コンパイラでコンパイルできるプログラムへ内部で変換することにした。

変換用のプログラムは AWK のスクリプトとして作成した。およそ次の動作をする。

- (1) プログラム中に Sound 文があれば出力ピンを取り出す
- (2) Data 文があれば演奏データであるか確認する
- (3) 演奏データであれば Sound 文の引数として与えるデータに変換する
- (4) Sound 文を含みかつ Data 文に含まれるデータが全て演奏データであれば電子オルゴールのプログラムとみなし、プログラムを生成する(bascomp_i_orz.bas)

尚、(3)で演奏データから Sound 文のパラメータへ変換するために使用したアルゴリズムはヒダピオが出力した BASCOM-AVR 用のプログラムを参考にした。

これにより、ヒダピオおよびヒダピオが出力した BASCOM-AVR 用のプログラムにはほとんど変更を加えずに電子オルゴールの機能を実現することができた。

4.9 コンパイラ・ドライバ

本コンパイラの利用者から直接 AWK が見えないようにし、コンパイルおよびアセンブルを自動的に連続実行して HEX ファイルの出力をするためのコンパイラ・ドライバを Windows の CMD.EXE 用バッチファイルとして作成した。また、ドラッグ&ドロップによって簡単にコンパイルできるようにショートカットを作成した。ショートカットには絶対パスが書き込まれるため利用者が作成しなくてはならないが、WSH(Windows Script Host)の VBScript で書かれたスクリプトによりショートカットを半自動で作成するようにした。

電子オルゴールは AWK の変換スクリプトを書くことによって対応し、コンパイラ・ドライバから自動的に実行するようにした。

4.10 ヒダピオとの統合

本コンパイラはヒダピオの簡易プログラミングの文法を受理しアセンブリ言語のソース・プログラムを出力する。ヒダピオの出力する BASCOM-AVR 用のソース・プログラムもコンパイルすることが出来る。コンパイラ・ドライバは起動時に bascom.bas または bascomp_i.bas という名前のファイルが在れば、それをソース・プログラムとして処理する。bascom.bas に対しては bascom.asm が、bascomp_i.bas に対しては bascomp_i.asm が出力ファイルの名前となる。ヒダピオは本コンパイラでコンパイルするために bascomp_i.bas というファイルを出力するので、これをコンパイルする。さらに、浅田氏により bascomp_i.exe が開発され、ダブルクリックすることで bascomp_i.bat へのショートカットを使って bascomp_i.bas をコンパイルすることができる。これは bascomp_i.bat のショートカットを直接起動するとコンソールが表示され中学生に余計な物を見せることになるのを防いでいる。

4.11 コンパイル法

本コンパイラはヒダピオでの利用を想定して開発したが、ヒダピオとは独立したシステムである。中学生だけでなく高校生以上の生徒に対応した教材を作成する場合には単独で使用することができる。

プログラムの作成には Windows のメモ帳等を使用することができる(図 2)。また、コンパイルはソースファイルをショートカットアイコンにドラッグ&ドロップする

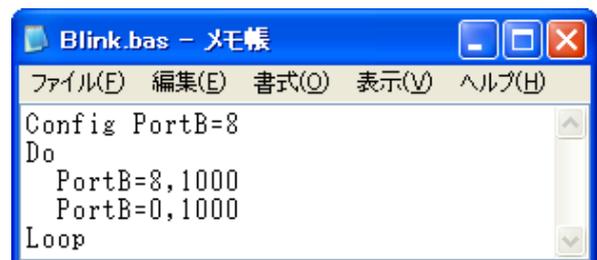


図 2 メモ帳でプログラム作成

だけで簡単に行うことができ、アセンブリ言語のソースである「bascom.asm」(プログラム 3)とアセンブルした結果の機械語ファイルである「bascom.hex」が出力される(図 3)。

5. おわりに

中学生のためのマイコン制御学習用プログラム言語の設計と実装を行った。平成 23 年 6 月の終わりから開発を始め 9 月 3 日にヒダピオに取り込まれるまでに約 2 カ月間かかった。プログラムの行数にして 2300 行程である。

当初は 2 週間から 1 カ月で完成する予定が倍になった。行数も 400 行から 700 行位と見積もっていたが約 3 倍となった。電光掲示板や電子オルゴールに対応したのが原因である。

本コンパイラはヒダピオに同梱する形で配布しており、実際に中学校の授業で利用されるほか、小中学生を対象にしたものづくり教室等で電子オルゴール製作に使用される等、その有効性、実用性について実証されている。

ライセンスは GPLv3 として単体でも配布している⁸⁾。

ヒダピオシステムと本研究の成果がより多くの中学生に利用され計測・制御の理解に資することを期待する。

プログラム 3 出力されたアセンブリ言語(一部略)

```

; Config PortB=8
    ldi R16,0b00001000
    out DDRB,R16
; Do
do_loop_top1:
;   PortB=8,1000
    ldi R16,0b00001000
    out PORTB,R16
    ldi R24,low (1000)
    ldi R25,high (1000)
    rcall waitms
;   PortB=0,1000
    ldi R16,0b00000000
    out PORTB,R16
    ldi R24,low (1000)
    ldi R25,high (1000)
    rcall waitms
; Loop
    rjmp do_loop_top1

```



図 3 ドラッグ&ドロップでコンパイル

本研究に当たりヒダピオの作者である浅田氏に協力いただき本研究の成果であるコンパイラをヒダピオに取り込んでいただいた。赤松武史氏には AVR 用のライブラリの使用を許可していただいた。ここに感謝の意を表する。

参考文献

1. 文部科学省: 新学習指導要領 技術・家庭, http://www.mext.go.jp/a_menu/shotou/new-cs/youryou/c hu/gika.htm
2. 開隆堂出版株式会社: 地域の実態と特性を活かした教科経営の工夫, <http://www.kairyudo.co.jp/general/data/contents/05-data/02-chu/gijutsu/jissen/e-shimane-01.pdf>
3. 浅田寿展: JA 制御 ヒダピオシステム (中学校, 技術・家庭科, コンピュータ制御, USB-IO, プログラミング), <http://hidapio.jp/>
4. 浅田寿展: 初心者にも取り組むことが可能な「プログラムによる計測・制御」の教材開発をめざして, <http://www.kairyudo.co.jp/general/data/contents/05-data/magazine/kgk-journal/vol45-1.pdf>
5. MCS Electronics: Home - MCS Electronics, <http://www.mcselec.com>
6. Klabaster: Freeware Downloads, <http://www.klabaster.com/freeware.htm#dl>
7. 赤松武史: ELM by ChaN, <http://elm-chan.org>
8. 谷岡政宏: 谷岡のページ, <http://t.wiki.coh.jp>

(原稿受付 2014/01/15、受理 2014/03/25)

*谷岡政宏, 修士 (工学)
北海道職業能力開発大学校, 〒047-0292 北海道小樽市銭函
3-190 email: m1tanioka@hokkaido-pc.ac.jp
Masahiro Tanioka, Hokkaido Polytechnic College, 3-190
Zenibako ,Otaru-shi, Hokkaido 047-0292