# シーケンス制御技術者のための パソコン制御プログラム訓練用教材の作成

ポリテクセンター富山　訓練課　電気・電子系　仲野　忠行
（富山職業能力開発促進センター）

## 1．はじめに

従来，プログラマブルコントローラを使ったシーケンス制御を行うには，リレーの接点，コイルを基本としたラダー言語によるプログラム作成技術を習得する必要があった。しかし，近年，パソコンの低価格化と高性能化，ユーザインターフェースの高度化に伴い，プログラム言語，特にVisual Basic言語を使ったパソコン制御が行われることが多くなってきている。

しかし，ラダー言語とプログラム言語は，その記述方法や命令体系が大きく異なることから，ラダー言語を習得している技術者にとってはプログラム言語の習得は容易ではない。

本稿は，ラダー言語プログラム作成技術を習得している技術者が，Visual Basic言語を使ったパソコンによる制御プログラムの教育訓練を受講する際に活用できるよう，ラダー言語と類似の命令体系で記述することができるパソコン制御プログラム訓練用教材を作成したので報告する。

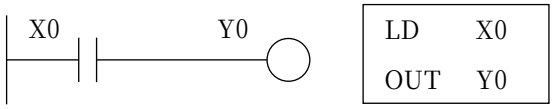
## 2．パソコン制御プログラム開発環境について

今回使用したパソコン制御プログラム開発環境を下記に示す。

パソコン：EPSON　Type-VF　(DOS/V)
OS：Microsoft Windows98
プログラム言語：Microsoft Visual Basic Ver6.0
デジタル入出力ボード：PCI2702C
(㈱インターフェース製)
制御用ソフトウェア：GPC2702C
(㈱インターフェース製)

## 3．従来の制御プログラム例

プログラマブルコントローラを使って，入力ポートに接続されたスイッチX0が押されるとき出力ポートに接続されたランプY0が点灯するプログラムを，ラダー図，ラダー言語で作成すると，図1のようになる。




（a）ラダー図　　（b）ラダー言語

**図1　プログラマブルコントローラを使った制御プログラム例**

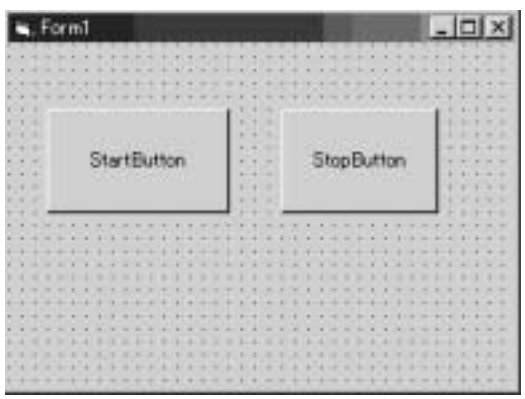



**図2　Visual Basic言語で記述した従来の制御プログラム例（フォーム）**

同じ動作をするプログラムをVisual Basic言語で記述すると，フォームが図２，プログラムコードが表１のようになる。

**表１　Visual Basic言語で記述した制御プログラム例（プログラムコード）**

```
Dim RunFlag As Boolean
Private Sub StartButton_Click()
    RunFlag = True
    Do While RunFlag
        If InPort And 1 Then
            OutPort 1
        Else
            OutPort 0
        End If
        DoEvents
    Loop
End Sub
Private Sub StopButton_Click()
    RunFlag = False
End Sub
```

（注）このプログラムのなかで使われているInPortファンクションプロシージャは，1バイトの入力ポートデータを戻り値とする関数である。またOutPortサブプロシージャは，1バイトの出力ポートに引数のデータを出力する関数である。これらの関数はオペレーティングシステムを介してハードウェアを操作するための特殊な命令で記述されているため，プログラムコードの記述は省略している。

上記に示したように，ラダー言語に比べてパソコンのプログラム言語では，If文などの条件文やInPort，OutPortなどの入出力関数，Do，whileなどの制御文の記述が必要になり，ラダー言語を習得している技術者の訓練にこれらのプログラムを提示，説明する必要がある。

## ４．ラダー言語と類似の命令体系を利用できるプロシージャの作成

表１で示したプログラムのなかで，実際の制御を記述した部分はDo　While～Loopで挟まれた網掛けの部分だけである。

そこで，ラダー言語のみを習得している技術者でも容易にプログラムできるよう，網掛けの部分のプログラムをラダー言語と同様に記述できるプロシージャを作成した。

ラダー言語的仕様を表２に，作成したプログラムのフォームを図３に，実際の制御を記述した部分のプログラムを表３に，作成した各プロシージャを表４に，それぞれ示す。

このプログラムは，図１のプログラムと同じ動作をする。表３の網掛け部分の記述が図１（b）のラダー言語のプログラム例とよく似たものになっている。

タイマリレーはコントロール配列のタイマーコントロールを利用しているため，図３のフォームには８個のタイマーコントロールを配置する必要がある。

**表２　作成したプロシージャのラダー言語的仕様**

| | |
|---|---|
| 入力リレー | Ｘ０～Ｘ７の８点 |
| 出力リレー | Ｙ０～Ｙ７の８点 |
| 補助リレー | Ｍ０～Ｍ７の８点 |
| タイマリレー | Ｔ０～Ｔ７の８点<br>タイマー設定値は0.1秒単位 |
| カウンタリレー | Ｃ０～Ｃ７の８点 |

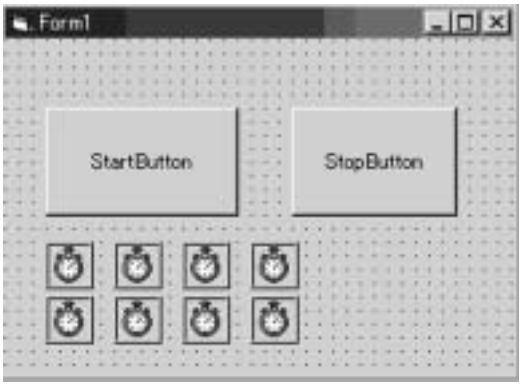



**図３　ラダー言語と類似の命令体系を利用できるプロシージャを使って作成した制御プログラム例（フォーム）**

**表３　実際の制御を記述した部分のプログラム**

```
Do While RunFlag
    InData = InPort
    Ld X(0)
    OutY(0)
    OutPort OutData
    DoEvents
Loop
```

**表４　ラダー言語と類似の命令体系を利用できるプロシージャのプログラムコード**

```
Dim RunFlag As Boolean
Dim InData As Byte, OutData As Byte, MemData As Byte,
TimData As Byte, CntData As Byte
Dim Condition As Boolean
Dim CntCount(0 To 7) As Integer
Dim CntFlag(0 To 7) As Integer
Public Sub Ld(InBool As Boolean)
  If InBool Then
    Condition = True
  Else
    Condition = False
  End If
End Sub
Public Function X(Number As Integer)
  If (Number < 0) Or (Number > 7) Then
    MsgBox ("該当しない入力リレー番号です。")
    Unload Me
  End If
  X = CBool(InData And 2 ^ Number)
End Function
Public Function Y(Number As Integer)
  If (Number < 0) Or (Number > 7) Then
    MsgBox ("該当しない出力リレー番号です。")
    Unload Me
  End If
  Y = CBool(OutData And 2 ^ Number)
End Function
Public Function M(Number As Integer)
  If (Number < 0) Or (Number > 7) Then
    MsgBox ("該当しない補助リレー番号です。")
    Unload Me
  End If
  M = CBool(MemData And 2 ^ Number)
End Function
Public Function T(Number As Integer)
  If (Number < 0) Or (Number > 7) Then
    MsgBox ("該当しないタイマリレー番号です。")
    Unload Me
  End If
  T = CBool(TimData And 2 ^ Number)
End Function
Public Function C(Number As Integer)
  If (Number < 0) Or (Number > 7) Then
    MsgBox ("該当しないカウンタリレー番号です。")
    Unload Me
  End If
  C = CBool(CntData And 2 ^ Number)
End Function
Public Sub OutY(Number As Integer)
  If (Number < 0) Or (Number > 7) Then
    MsgBox ("該当しない出力リレー番号です。")
    Unload Me
  End If
  If Condition Then
    OutData = OutData Or 2 ^ Number
  Else
    OutData = OutData And (Not 2 ^ Number)
  End If
End Sub
Public Sub OutM(Number As Integer)
  If (Number < 0) Or (Number > 7) Then
    MsgBox ("該当しない補助リレー番号です。")
    Unload Me
  End If
  If Condition Then
    MemData = MemData Or 2 ^ Number
  Else
```

```
        MemData = MemData And (Not 2 ^ Number)
    End If
End Sub
Public Sub OutT(Number As Integer, Delay As Integer)
    If (Number < 0) Or (Number > 7) Then
        MsgBox ("該当しないタイマリレー番号です。")
        Unload Me
    End If
    Timer(Number).Interval = Delay * 100
    If Condition Then
        Timer(Number).Enabled = True
    Else
        Timer(Number).Enabled = False
        TimData = TimData And (Not 2 ^ Number)
    End If
End Sub
Public Sub OutC(Number As Integer, MaxCount As
Integer)
    If (Number < 0) Or (Number > 7) Then
        MsgBox ("該当しないカウンタリレー番号です。")
        Unload Me
    End If
    Select Case CntFlag(Number)
        Case 0
            If Condition Then
              CntCount(Number) = CntCount(Number) + 1
                If CntCount(Number) < MaxCount Then
                    CntFlag(Number) = 1
                Else
                    CntData = CntData Or 2 ^ Number
                    CntFlag(Number) = 2
                End If
            End If
        Case 1
            If Not Condition Then CntFlag(Number) = 0
        Case 2
            'リセットを実行するまで何もしない
    End Select
End Sub
Public Sub ResetC(Number As Integer)
    If (Number < 0) Or (Number > 7) Then
        MsgBox ("該当しないカウンタリレー番号です。")
        Unload Me
    End If
    If Condition Then
        CntData = CntData And (Not 2 ^ Number)
        CntCount(Number) = 0
        CntFlag(Number) = 0
    End If
End Sub
Private Sub Timer_Timer(Index As Integer)
    TimData = TimData Or 2 ^ Index
End Sub
```

## ５．ラダー言語と類似の命令体系を利用できるコントロールの作成

Visual Basic言語では，プログラム作成の効率化やプログラム構造の簡単化を図るために，プログラムの一部をコントロールと呼ばれる図形の中に組み込んで利用することができる。コマンドボタンコントロールが，プロパティ，メソッド，イベントを使って，グラフィックプログラムやマウスとのインターフェースプログラムを簡単に記述できるのはこの理由による。

そこで，このコントロールの機能を利用して，先に作成したラダー言語と類似の記述ができるプロシージャやDo While～Loopで挟まれた部分を組み込んだコントロールを作成した。

コントロールの作成には「ActiveXコントロールインターフェースウイザード」と呼ばれるコントロール自動作成ツールを利用した。ラダー言語と類似の記述ができるプロシージャをメソッドとして使用できるようにすること，制御プログラムのDo～Whileループで囲まれた部分を実行するたびにイベント（ExecLadderイベント）を発生させることが今回作成したコントロールの主な仕様である。

コントロールの作成手順については，紙面の都合上省略する。Visual Basicのヘルプファイルや参考

文献１）を参照されたい。

作成したコントロールのメソッド，イベントに関する仕様を表５に，このコントロールを利用した制御プログラム例を図４，表６に示す。表６はシーケンス制御回路でよく利用される自己保持回路とタイマリレーとカウンタリレーを利用した点滅回路である。X（０）を押すとY（０）が１秒周期で５回点滅する。点滅動作中にX（１）を押すと点滅動作を終了する。

このプログラムと同じ動作をするラダー図，ラダー言語を図５，表７に示す。接点の接続命令（AND，OR）がLdと同じ行に記述されてしまうこと，メソッドの前に". "が必要なことなど，いくつかの相違点はあるが，おおむねラダー言語と類似の記述になっている。

**表５　作成したコントロールの仕様**

| | |
|---|---|
| StartLadder メソッド | ExecLadder イベントを発生させるメソッド |
| StopLadder メソッド | ExecLadder イベントを停止させるメソッド |
| ExecLadder イベント | ラダー言語と類似の記述ができるメソッドをこのイベントプロシージャの中に記述する |
| Ld,X,Y,M,T,C,OutY, OutM,OutT,OutC Reset の各メソッド | Ld,X,Y,M,T,C,OutY, OutM,OutT,OutC Reset の各プロシージャと同じ機能を持つ |
| コントロールのフォームの構成 | タイマーコントロール８個とラベル１個で構成 |

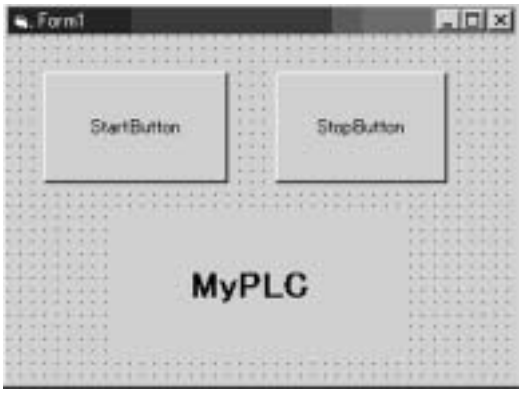



**図４　自作のコントロール（MyPLC）を利用した制御プログラム例（フォーム）**

**表６　自作のコントロール（MyPLC）を利用した制御プログラム例（プログラムコード）**

```
Private Sub StartButton_Click()
    MyPLC.StartLadder
End Sub

Private Sub StopButton_Click()
    MyPLC.StopLadder
End Sub

Private Sub MyPLC_ExecLadder()
    With MyPLC
        .Ld (.X(0) Or .M(0)) And Not .X(1) And Not
        .C(0)
        .OutM (0)
        .Ld .X(1)
        .ResetC (0)
        .Ld .M(0) And Not .T(1)
        .OutT 0, 5
        .Ld .T(0)
        .OutT 1, 5
        .OutC 0, 5
        .Ld Not .T(0) And .M(0)
        .OutY (0)
    End With
End Sub
```

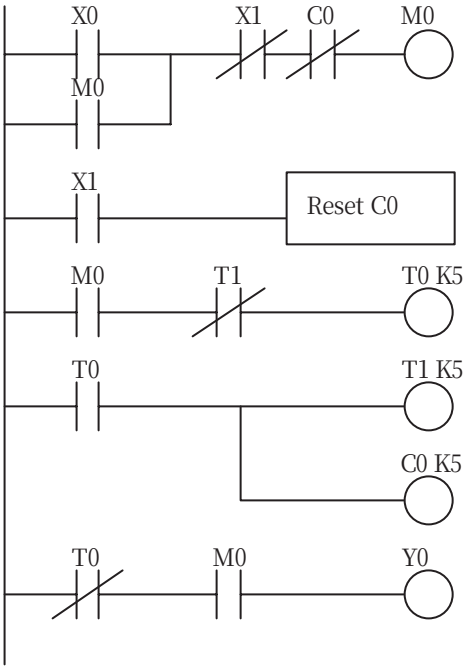



**図５　自作のコントロール（MyPLC）を利用した制御プログラムと同じ動作をするラダー図**

**表７　自作のコントロール（MyPLC）を利用した制御プログラムと同じ動作をするラダー言語**

| | |
|---|---|
| LD | X0 |
| OR | M0 |
| ANI | X1 |
| ANI | C0 |
| OUT | M0 |
| LD | X1 |
| RESET | C0 |
| LD | M0 |
| ANI | T1 |
| OUT | T0 |
| K | 5 |
| LD | T0 |
| OUT | T1 |
| K | 5 |
| OUT | C0 |
| K | 5 |
| LDI | T0 |
| AND | M0 |
| OUT | Y0 |

## ６．まとめ

ラダー言語プログラム作成技術を習得している技術者が，Visual Basic言語を使ったパソコンによる制御プログラムの教育訓練を受講する際に活用できるよう，ラダー言語と類似の命令体系で記述することができるパソコン制御プログラム訓練用教材を作成した。

ラダー言語の記述に精通していれば，短期間でパソコン制御プログラムが作成可能なことや，ラダー言語と同様な記述ができるプロシージャのプログラム構造を理解することで，プログラムコントローラの動作を理解するための教材としても利用できると思われる。

一方，ラダー言語と全く同じ記述にはできないことや，フォームをマウスでドラッグすると制御プログラムが停止してしまう，タイマーの時間があまり正確ではない，といった問題もある。

これらの問題点やこの教材の活用方法については，今後検討していきたい。

**＜参考文献＞**

1）金藤仁：「自動計測システムのためのVB6入門」，技術評論社.