

テーマ	V B A で制御をしよう				
概要	プリンタポートの入出力信号を利用したLEDの点滅・スイッチ入力ボードを作成し、EXCEL や WORD に付属している Visual Basic for Application ( VBA ) を使って制御プログラムを作成することにより、コンピュータ制御の基本的な手法を学ぶ。				
ねらい	身近な家庭電化製品などに関連づけて考えさせるとともに、実際に簡単な制御回路とプログラムを作成することにより、制御の概念について具体的に理解させる。				
関連する主な科目・項目	情報B「情報通信と計測・制御の技術」	難易度	易・ <b>普</b> ・難	想定時間数	約4時間
準備するもの	制御ボード部品・工作工具(後述) EXCEL・WORD等	作成者	山田 富美義		

## 1 学習の展開

### (1) I/Oポートの製作

プリンタポートの入出力信号を利用して、出力をLEDの点滅として表示し、スイッチの状態を入力できるボードを製作します。

#### 回路図

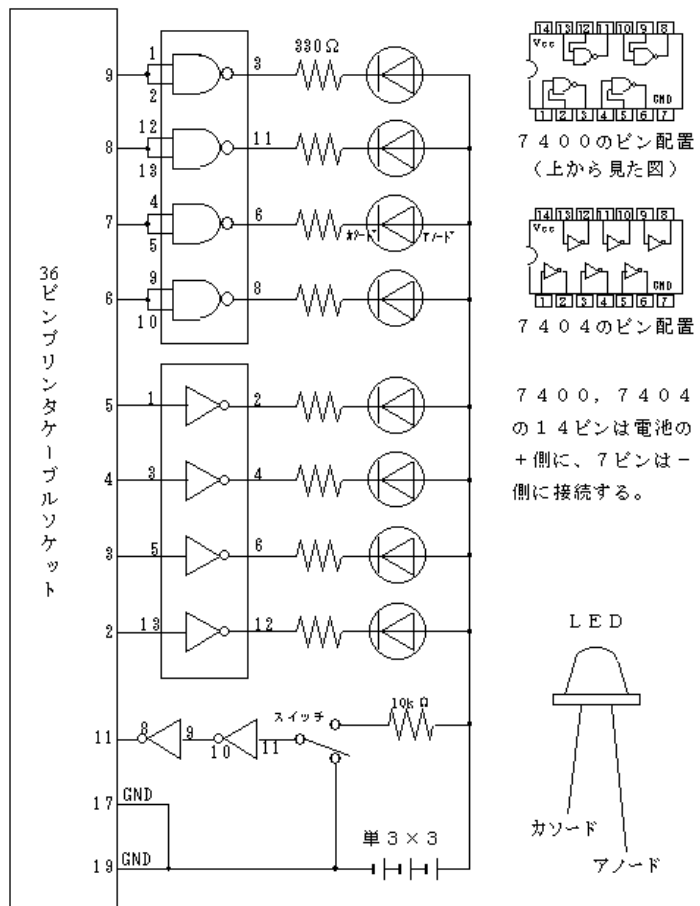
ボードの回路図は、右図のようになります。

出力は、8ビットあり各ビットごとに“1”が出力されるとLEDが点灯し、“0”が出力されるとLEDが消灯します。

入力は、1ビットのみでトグルスイッチの状態により、“1”または“0”が入力されます。

#### 部品一覧

IC (74LS00, 74LS04)	各1個
14ピンICソケット	2個
スペーサ	6本
LED (発光ダイオード)	8個
抵抗 (330 Ω)	8本
抵抗 (10k Ω)	1本
3Pスイッチ	1個
単3×3ソケット	1個
単3ソケットケーブル	1本
プリント基板	1枚
ビニール電線	約50cm
36ピンプリンタケーブルソケット (DDK 57-40360)	1個



## 配線

プリント基盤の裏面から見た実際の配線図は、右図のようになります。

これを参考にして、作成をしてみてください。

## 工作工具

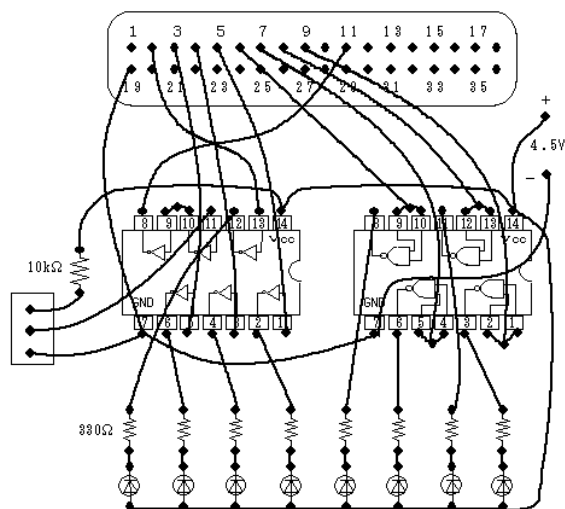
はんだごて、ニッパー、ラジオペンチ

プラスドライバー

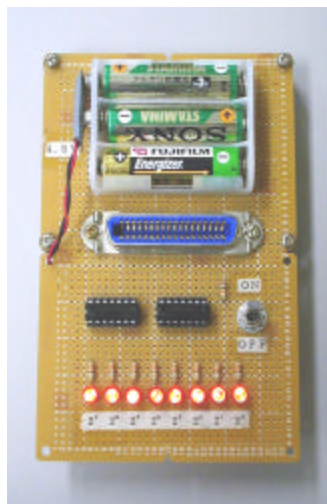
## 留意事項

- ・LEDのアノード、カソードを間違えないこと。抵抗は330 ~ 470 のものであればよい。
- ・はんだ付けが終了したら、乾電池を接続してみます。8個のLEDが全て点灯すれば配線は間違いありません。念のため、乾電池の-端子とコネクタ端子2~9番端子をリード線で接続してみても1個ずつLEDが消灯することを確認します。

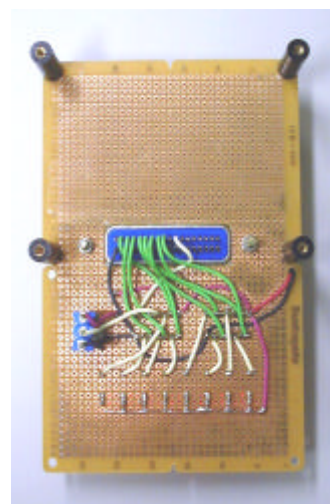
完成写真



全体図



表面



裏面

## (2) VBAによる制御プログラムの作成手順

使用するアプリケーションについては、Office 97以降のVBAが利用できれば、Word・Excel・PowerPoint・Accessなど、どれでも動作します。

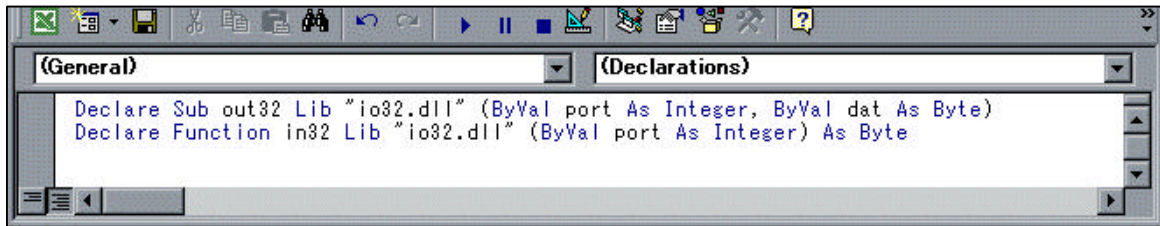
### プログラムウインドウの表示

アプリケーションを起動し、メニューの「ツール」 「マクロ」 「Visual Basic Editor」を選択します。次に、Visual Basic Editor ウィンドウのメニューから「挿入」 「標準モジュール」を選択すると、プログラムウインドウが表示されます。

### 外部ライブラリファイルの使用宣言

VBAを含めてVisual Basicには、外部ポートに直接入出力する命令(VBではメソッドという)が用意されていないので、ポートの入出力命令を使用するために最初に宣言をする必要があります。

そこで、プログラムウインドウの先頭に以下の2行のコードを記述します。



1行目は、“io32.dll”というダイナミックリンクライブラリファイルの中のout32というポートへの出力命令をサブルーチンプロシージャとして参照するように宣言しています。同様に2行目は、in32というポートからの入力命令をファンクションプロシージャとして参照するように宣言しています。

また、プログラムの実行に先立って“io32.dll”ファイルを、VBAプログラムファイルを保存する場所、または“windows\system”フォルダにコピーしておく必要があります。

#### ポートへの出力命令の使用法

ポートへの出力は、以下のように記述して使用します。

```
Call out32 (出力ポートアドレス , 出力データ)
```

出力ポートアドレスは2バイトで表される番地で、一般的なプリンタポートの出力ポートアドレスは、16進数で(0378)です。また、出力データは8ビット分しかありません。

したがって、全部のLEDを点灯したい場合は、次のように記述すればよいことになります。

```
Call out32 (&h0378 , &hFF)
```

#### ポート入力命令の使用法

ポートからの入力は、以下のように記述します。

```
変数名 = in32 (入力ポートアドレス)
```

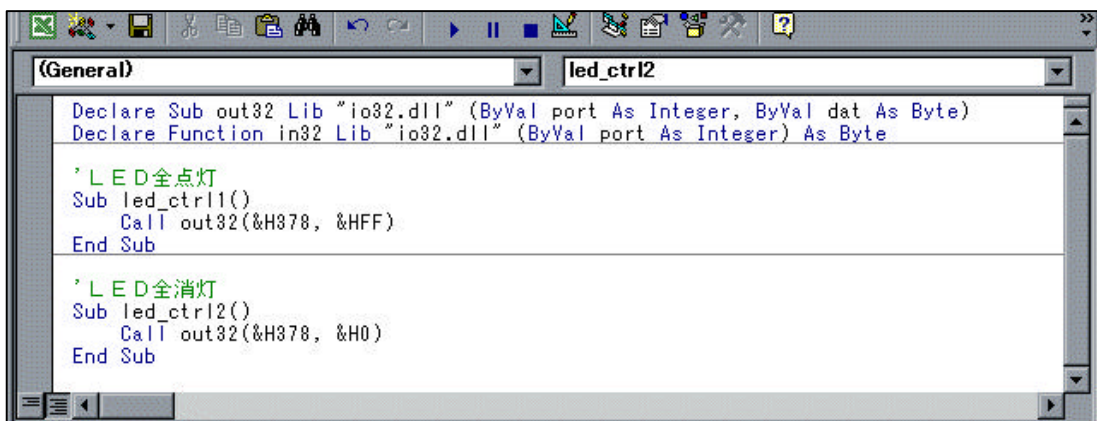
入力ポートアドレスも出力ポートアドレスと同様に2バイトで表される番地で、一般的なプリンタポートの入力ポートアドレスは、16進数で(0379)です。読み込まれたデータは、変数名で示されている場所に記憶されますが、今回のI/Oボードは、スイッチのON・OFFによって最上位ビットのみが“1”または“0”になり、その他のビットについては不定です。したがって、スイッチのON・OFFをチェックするには、入力データと16進数の(80)と論理積をとる必要があります。

```
dat = (in32 (&h0379) And &h80)
```

### (3) 基本的な制御プログラム

#### LEDの全点灯・全消灯プログラム

以下のように記述します。



また、このプログラムをボタンオブジェクトに登録すれば、ボタンのクリックにより実行できるようになります。右図は、Excel の場合です。



スイッチの状態によりLEDの点灯方法を変更するプログラム

以下は、スイッチがONの場合はLEDが右4個のみ、OFFの場合は左4個のみ点灯するプログラム例です。

```

(General) | led_ctrl3
Declare Sub out32 Lib "io32.dll" (ByVal port As Integer, ByVal dat As Byte)
Declare Function in32 Lib "io32.dll" (ByVal port As Integer) As Byte

'スイッチによる切り替え
Sub led_ctrl3()
  If ((in32(&H379) And &H80) > 0) Then 'スイッチの状態チェック
    Call out32(&H378, &HF) 'ONの場合
  Else
    Call out32(&H378, &HF0) 'OFFの場合
  End If
End Sub

```

## 2 発展的な問題

ポートを制御するためには、2進数の概念とビットごとの操作ができる必要があります。そこで、以下のようなプログラム例を示しますので、動作を理解して応用してみてください。

スイッチがONの場合はLEDが右から左へ、OFFの場合は左から右へ順番に点灯するプログラム

```

(General) | led_ctrl
Declare Sub out32 Lib "io32.dll" (ByVal port As Integer, ByVal dat As Byte)
Declare Function in32 Lib "io32.dll" (ByVal port As Integer) As Byte

'スイッチによる切り替え
Sub led_ctrl()
  dat = 1
  While (1)
    If ((in32(&H379) And &H80) > 0) Then 'スイッチの状態チェック
      If (dat > &H80) Then 'ONの場合に
        dat = 1 'LEDを右から左に
        Call out32(&H378, dat) '順番に点灯させる
        dat = dat * 2
      Else
        If (dat < &H1) Then 'OFFの場合に
          dat = &H80 'LEDを左から右に
          Call out32(&H378, dat) '順番に点灯させる
          dat = dat / 2
        End If
      End If
      For i = 1 To 100000: Next i '遅延のためのタイマー
    Wend
  End Sub

```

### 3 参考

#### (1) io32.dll プログラムについて

ダイナミックリンクライブラリは、Visual C++等のようなDLLに対応したコンパイラがあれば作成できます。以下に、VC++で作成したときのio32.dllのソースコードとヘッダコードを示します。

#### ソースコード

io32.cpp

```
#include <conio.h>
#include <windows.h>
BOOL WINAPI DIIMain(HANDLE hModule,DWORD ul_reason_for_call,LPVOID lpReserved){
    return TRUE;
}
void WINAPI out32(unsigned int port,unsigned char dat){
    _outp(port,dat);
}
unsigned char WINAPI in32(unsigned int port){
    return(_inp(port));
}
```

#### ヘッダコード

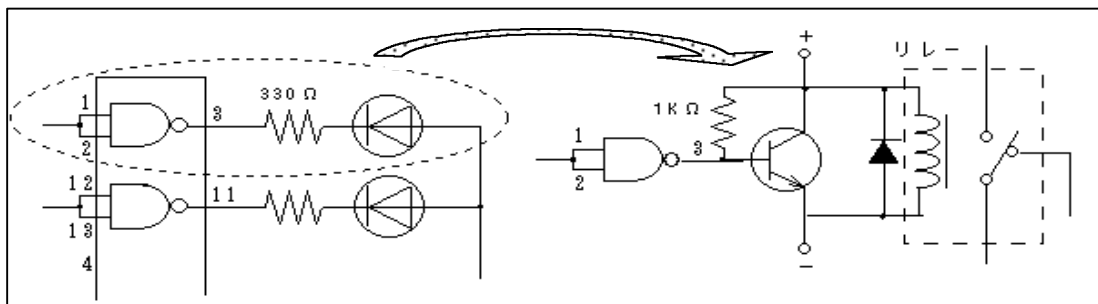
io32.def

```
LIBRARY IO32
EXETYPE WINDOWS
CODE PRELOAD MOVEABLE DISCARDABLE
DATA PRELOAD SINGLE

EXPORTS
    out32
    in32
```

#### (2) LED以外の制御を行いたい場合

制御ボードのLEDの代わりにリレーを付けることにより、色々な機器のON・OFF制御ができるようになります。以下に回路例を示します。



#### (3) Windows95/98でのプリンタインターフェースのI/Oアドレスの調べ方

Windows95/98の「システムのプロパティ(マイコンピュータ コントロール パネル システム)」から「デバイスマネージャタグ」を選び、そこから「ポート」「プリンタポート」をクリックし、そのプロパティのリソースタグより「I/Oの範囲」で示されているアドレスの値を調べます。

「I/Oの範囲」で示されている16進4桁のアドレスのうち、先頭の値が出力ポートのアドレス、次の値が入力ポートのアドレスになります。