専攻演習(一)(農業ロボット工学コース)

第2回:DIO入力

平成 16年 10月 6日

- 1 プログラミング
- 課題1: トグルスイッチの状態を表示する

以下の手順は,実験ボードを用いるプログラムでは共通 になりますので,十分に理解してください.

(1) Visual Basic の起動

デスクトップのアイコンから起動し,新しいプロジェクトをクリックしてください.

しいプロジェクト			
ブロジェクトの種類(<u>P</u>):		テンプレート①:	
	ット ト プロジェクト ノョン	Windows アプリ・ ケーション	「「「」 では、 し ー い し ー い し ー い し ー い し 、 し し 、 し 、 し し し 、 し 、 こ 、 し 、 こ し こ こ し こ こ し こ こ こ こ こ こ こ こ こ こ こ こ こ
		スマート デバイ ス アプリケーシー	ASP.NET Web アプリケーション サービス
Windows ユーザー インターフェイ	スを含むアプリケーションを	作成するためのプロジェクトです。	
ノロンエクト名(M): 04 場所(L): 04	Documents and Setting	gs¥a¥My Documents¥Visual S	Studio F▼ 参照(B)
	LOUIS VIN D		V041000 (-(*;********

プロジェクト(<u>S</u>)	オンライン リソース(N)	マイブロフィール(Y)		
既存のブロジェクトを開く				
名前			更新日時	
新しいプロジェカト(1)	プロジェカトを閉((0)			
*100-20212120	ADATAL SIMUTON			

図 1: Visual Basic 起動時

(2) プロジェクトの作成

プロジェクトの種類およびテンプレートを,それぞれ Visual Basic プロジェクト,Windows アプリケーショ ンを選択し,プロジェクト名に 041006 と入力し,OK をクリックして下さい.

(3) モジュールの追加

実験ボード専用の関数を利用するために,標準モジュー ルを追加します.ソリューションエクスプローラ上で 041006 を右クリックし,追加メニューから既存項目の 追加をクリックします.現在の 041006 フォルダより 一つ階層が上のフォルダ Visual Studio Projects の MT1PortIO.vb をクリックします.すると,ソリュー ションエクスプローラに MT1PortIO.vb が追加された ことが確認できたと思います.





図 3: 標準モジュールの追加

(4) フォーム名の変更

ソリューションエクスプローラの Form1.vb 上で右ク リックし,メニューから名前の変更を選択します.今回の 演習では,フォームモジュールの名前を FormDIO1.vb としましょう.次に,プロパティウィンドウにて,フォー ムの Name および Text を,ともに FormDIO1 に変更 して下さい.

- (5) フォームのデザイン
 - フォーム上に
 - Button \times 3
 - TextBox $\times 2$
 - Label \times 3

を配置してください.

スタート ページ	ormDIO1.vb [5	デザイン]*				
🛃 FormDIO1						
: Label1						
	E		Label2	TextBox1	C	P
;; Button	a (11)	Button2	1	TextBox2	Button3	
::.	t::		Label3		!:	
						<u>-</u>

図 4: フォームのデザイン

ボタンにはそれぞれ初期化,信号入力,終了のイベン トを付けますので,オブジェクト名をそれぞれBtnInit, BtnDI,BtnEndとしてください.また,Textはそれ ぞれ,初期化,読み込み,終了としてください.

一方,テキストボックスのオブジェクト名は TextDId,
TextDIh とし,10進数出力,16進数出力に利用します.

3 つのラベルうちーつは,作成したプログラムを示すタ イトル,例えば,専攻演習 課題 DIO 入力を入力します. また,残りの二つは,TextDId と TextDIh の横に,そ れぞれのテキストボックスが10進数,16進数を示すよう に,Textを変更し配置してください.

(6) コーディング

ボタン (BtnInit) のイベントプロシージャを記述し ましょう.PC のプリンタポートのアドレスを設定す る関数 SetLptAddr() と実験ボードを初期化する関数 InitializeMT1()を実行させます.なお,実験ボードを 利用する際は,必ずこの二つの関数を実行し初期化作業を 行う必要があります.

Private Sub BtnInit_Click SetLptAddr(&H378) InitializeMT1() End Sub

実験ボードを初期化した後にプログラムを終了すると すると,ボタン (BtnEnd) のイベントの内容は次のように なります.

Private Sub BtnEnd_Click InitializeMT1() End End Sub ボタン (BtnDI) のイベントは授業中に説明します.

Private Sub BtnDI_Click

End Sub

課題 2: スイッチの状態にかかわらず下位 4 ビットは常に 0 となる

課題 3: スイッチの状態にかかわらず上位 3 ビットは常に 1 となる

課題 4: 課題 1. の ON/OFF の逆転

2 説明事項

2.1 ポートとアドレス

図 5 は今回の演習用教材の概略図です.図に示すよう に,実験ボードとコンピュータはパラレルポート(プリン タポート)を介して,信号の授受を行います.



図 5: 教材の概略

形式	型	バイト数	値の範囲
Byte	整数型	1	$0 \sim 255$
Boolean	ブール型	2	真 (True) または 偽 (False)
Short	整数型	2	$-32,767 \sim 32,767$
Integer	倍長整数型	4	$-2,147,483,648 \sim 2,147,483,648$
Single	単精度浮動小数点型	4	$-3.40282 E38 \sim -3.40282 E38$
Double	倍精度浮動小数点型	8	$-1.79769 E308 \sim 1.79769 E308$
String	文字列型	可変長	2GB まで

表1:基本的なデータ型

通常,周辺機器とのデータのやりとりは I/O ポートを 介して行われます。また,各ポートにはアドレスが割り当 てられ,そのアドレスを指定することにより周辺機器と データのやりとりが行われます.今回の演習で利用する コンピュータでは,パラレルポートは&H378 に割り当て られています.従って,パラレルポートを利用するために は,プログラム上では&H378 番地を参照することになり ます.

2.2 プロシージャ(関数)

Visual Basic のプロシージャには,サブルーチンプロ シージャ(Sub ~ End Sub)とファンクションプロシー ジャ(Function ~ End Function)があります.大きな違 いは,サブルーチンプロシージャには戻り値がなく,ファ ンクションプロシージャには戻り値があるということで す.また,サブルーチンプロシージャには引数がない場合 があり,ファンクションプロシージャは引数を必要としま す.オブジェクトのイベントプロシージャは一般的にサ ブルーチンプロシージャの形式をとり,ファンクションプ ロシージャは,標準モジュールやジェネラルプロシージャ に記述されます.

2.3 変数

プログラムでは多くの場合に演算を伴い,変数を多用し ます.プログラム中に変数を定義することにより,演算処 理により刻々と変化するデータをメモリに格納すること ができます.

プログラムで変数を用いる場合,変数を宣言することが 不可欠です.基本的はデータ型は表1の通りです.例え ば,プロシージャの中で変数iを整数型,変数aを少数点 型で利用するには,

 $Dim_{\sqcup}i_{\sqcup}as_{\sqcup}Integer$

 $\mathtt{Dim}_{\sqcup}\mathtt{a}_{\sqcup}\mathtt{as}_{\sqcup}\mathtt{Single}$

とプロシージャの最初に宣言します.なお,」は空白を示 します.

2.4 計算式

変数に値を代入するには,

i = 10

のように記述します.また,

i = i + 1

の場合は,変数 A の値に1を加えた値を,新たに変数 A に代入せよという命令になります.ようするに,=イコール記号の意味が数学の等式とは異なり, ← の意味をもつと考えることができます.また,算術演算子としては次のようなものを用い,計算の優先順位が決められています.

表 2: 演算の優先順位

優先順位	演算子	意味
1	括弧で囲まれたもの	
2	関数	
3	^	べき乗
4	-	負符号
5	* /	乗算 除算
6	¥	除算商
7	MOD	除算余
8	+ -	加算 減算

2.5 変数の型変換

前述のように,変数はデータ型を予め決めて宣言し利用 します.プログラム中には,データ型の異なる変数を代入 したい場合が多く発生します.そのような際には,変数の 型変換が必要になります.

Private Sub Hogehoge_Click Dim r as Integer Dim p as Single Dim s as Single r=3 p=3.14 s=2.0*CSng(r)*p End Sub

上記のプロシージャでは,sの算出の際に,変数rを型変 換(CSng)しています.これは,変数sはSingleで宣言 されており,変数sの算出にあたり演算される変数は,少 数点型である必要があるためです.また,TextBox1に算 出したsを表示させる際にも,TextBox1.Textへ代入する のは,文字である必要があるため,変数sを型変換(Cstr) しています.

表 3: 変数の型変換

関数	機能
CInt	式の値を整数型に変換
CSng	式の値を単精度浮動小数点型に変換
CDbl	式の値を倍精度浮動小数点型に変換
CLng	式の値を倍長整数型に変換
CByte	式の値をバイト型に変換
CStr	式の値を文字列型に変換
Hex	数値を 16 進数文字列に変換
Str	数値を 10 進数文字列に変換
Format	書式付き変換

2.6 ビット演算

コンピュータは基本的には0と1のビット単位で動作 するため,細部にわたる制御を行うためにはビット単位で の操作が不可欠になります.そのビット単位の操作のた めにビット演算子が用意されています.ビット論理演算 は対応するビット毎に決められた論理演算を行うことで す.一般的に,プログラムにいて論理演算子は表4のよう な目的で使用されます.

表 4: ビット論理演算子

演算子	意味	演算の目的
AND	ビットの AND	特定のビットのマスク
OR	ビットのOR	特定のビットを1にセット
XOR	ビットの排他的 OR	特定のビットの反転
NOT	ビットのNOT	全ビットの反転

各論理関数の真理値表を表5から表8に示します.

表 5: 論理積の真理値表

А	В	$A \cdot B$
0	0	0
0	1	0
1	1	1
1	0	0

表 6: 論理和の真理値表

А	В	A+B
0	0	0
0	1	1
1	1	1
1	0	1

表 7: 排他的論理和の真理値表

А	В	$A{\oplus}B$
0	0	0
0	1	1
1	1	0
1	0	1

表 8:	否定の真理値表			
	А	Ā	•	
	0	1		
	1	0		

3 関数

3.1 実験ボード固有の関数

Sub SetLptAddr(nADDR As Integer) 機能 プリンタポートのアドレスを設定 引数 nADDR:プリンタポートアドレス 戻り値 なし

Function InitializeMT1(void) As Boolean

__機能 実験ボード MT-1 の初期化

________ 引数 なし

戻り値 ブール型:初期化が成功するとTrue、失敗すると False を返します

Function GetDI_DATA() As Byte

- [—]機能 D/I ポートからデータを読み込む
 - 引数 nAddress:D/I ポートアドレス

戻り値 バイト型