พื้นฐานการอ่านและเขียนข้อมูลระหว่างโปรแกรม visual basic กับ pic16f877 microcontroller

ก่อนที่เราจะใช้ โปรแกรม Visual Basic ติดต่อกับ Plc ผู้เขียนขอเริ่มต้นแนวคิดพื้นในเรื่องของการควบคุม และการจัดการข้อมูลระหว่าง computer กับ ชุดควบคุมที่ใช้ ไมโครคอนโทรลเลอร์เป็นตัวหลักในการควบคุม ระบบอัตโนมัติ และคาดหวังว่าจะก่อให้เกิดแรงจูงใจสำหรับนักพัฒนาโปรแกรมมือใหม่ทั้งหลายหันมาให้ความ สนใจกับระบบควบคุมในโลปัจจุบันและต่อเนื่องไปในอนาคต

เหตุผลที่ผู้เขียนเลือกที่จะใช้ pic16f877 microcontroller เนื่อจากว่าการเขียนโปรแกรม โดยใช้ pic c compiler มีคำสั่งที่ใช้ในการติดต่อสื่อสารผ่านทาง Port RS232 ที่ใช้งานได้ง่าย เนื่องจาก ตัว compiler ได้สร้างฟังชั่นทางด้านการติดต่อสื่อสารผ่านทาง port rs232 ไว้เป็นที่ เรียบร้อยแล้ว

6.1 รูปร่างหน้าตาของ pic c compiler



6

ตัวอย่าง 6.1 การส่งข้อความจาก Microcontoller ให้กับ Computer ขนาด 1 บรรทัด

- เขียนโปรแกรมใน pic c compiler เพื่อกำหนดการทำงานใกับ microcontroller ตามรูป 6.1
- add Mscomm จาก library ซึ่งเป็นเครื่องมือที่ถูกสร้างขึ้นเพื่อใชงานทางด้านการติดต่อสื่อสารทาง serial port
- เขียนโปรแกรมใน visual basic เพื่อรับข้อความที่ส่งมาจาก microcontroller



Components	×
Controls Designers Insertable Objects	
 MicroHelp Key State Control Microsoft ActiveMovie Control Microsoft ActiveX Plugin Microsoft ADO Data Control 6.0 (SP4) (OLEDB) Microsoft Calendar Control 8.0 Microsoft Chart Control 6.0 (OLEDB) Microsoft Common Dialog Control 6.0 (SP6) Microsoft Data Bound Grid Control 5.0 (SP3) Microsoft Data Bound List Controls 6.0 	
Microsoft DataGrid Control 6.0 (SP5) (OLEDB) Microsoft DataList Controls 6.0 (OLEDB) Microsoft DataRepeater Control 6.0 (OLEDB) Microsoft Comm Control 6.0 Location: C:\WINDOWS\SYSTEM\MSCOMM32.OCX	Browse
ОК	Cancel <u>Apply</u>

ฐป 6.3 ขั้นตอนการ เพิ่มเครื่องมือในกล่อง tool box

รูป6.4 check box Microsoft Comm Control6.0

🍾 Project1 - N	/icrosoft Visual Basic [design]		\mathbf{X}
<u>E</u> ile <u>E</u> dit ⊻iev	v <u>P</u> roject F <u>o</u> rmat <u>D</u> ebug <u>R</u> un Q	<u>u</u> ery D <u>i</u> agram <u>T</u> ools <u>A</u> dd-Ins <u>W</u> indow <u>H</u> elp	
🛛 🛃 • 🏷 • 🛅	😅 🖬 👗 🏙 🏙 🎽 🖂 🛛	▶ II = 🧏 🖀 🖶 🎖 🛠 🔍 🐊	~~
Normal -	S Form1		×
		Project - Project1	
	Text1		
N 🔛 A		E Bi Project 1 (scada02 yh	
			ןיי
		Form1 (scada02.	frn
E# 10			
岗 🗆 🗀			
2			
		Properties - MSComm1	<u> </u>
		MSComm1_MSComm	ΨL

รูป 6.5 แสดงหน้าตาของ Mscomm control



- กำหนดคุณสมบัติของ Mscomm control

รูป6.6 กำหนดคุณสมบัติทั่วไป

Property Pages Property Pages		×
General Buffers	Hardware	
InBufferSize:	024 <u>O</u> utBufferSi	ze: 512
<u>R</u> Threshold:	<u>S</u> Threshold	0
Input <u>L</u> en:)	
Г	<u>E</u> OFEnable	
5		
	Cancel	Annly Heln

รูป6.7 กำหนดคุณสมบัติของ Buffers





4	Project1 - Form1 (Code)	
M	1SComm1 💌 OnComm	•
	Private Sub MSComm1_OnComm()	_
	Do Buffer\$ = Buffer\$ ج MSComm1.Input.	
	Loop Until InStr(Buffer\$, ":")	
	Text1.Text = Buffer\$	
	End Sub	-

รูป 6.8 เขียนโปรแกรมที่ sub Mscomm_oncomm()

🐃 Form1		
	temram, samraeng;	<u> </u>
		-
	,	_

รูป 6.9 ผลจากคำสั่ง printf("temram. samraeng:\r\n"); จาก pic c compiler

6.3 การอ่านค่า สัญญาน Analog จาก microcontroller



รูป 6.10 การแสดงค่าสัญญาน analog โดยใช้ textbox และ TechMeter



- ตัวโปรแกรมใน Microcontroller ตามรูป 6.11

รูป6.11 โปรแกรมอ่านสัญญาน analog แล้วส่งออก port rs232

PIC16F877



รูป6.12 วงจรสำหรับอ่านสัญญาน analog

- เครื่องมือใน Visual Basic

tool	property	defind	
Label1	caption	Analog chanel1	
Text1	Fore color	สีเขียว	
	font	Bold/14	
TechMeter	TechMeter maxvalue 255		
	minvalue	0	
	caption	Analog ch1	
Timer1	Time interval	1	
	enable	true	

Components	X
Controls Designers Insertable Objects	
C:\WINDOWS\SYSTEM\refedit.dll C:\WINDOWS\SYSTEM\TDC.OCX Cdlg CiListBox CiMenuBar CiMeterX CITECT Recipe Manager CiText CiTreeView ComponentOne VSFlexGrid 8.0 (Light) CoverEdCtrl ActiveX-Steuerelement-Modul Crystal Report Control 4.6 Desaware Animated Button Control CiMeterX Location: C:\\COMMON FILE5\CITECT\CIMETERX.OCX	
OK Cancel <u>A</u> ppl	у
ฐป 6.13 add tool TechMeter	

รูป 6.14 แสดงเครื่องมือต่าง ๆ บน form1

	🐂 Project1 - Microsoft Visual Basic [design]	<u>- 🗆 ×</u>
	<u>File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window H</u>	<u>H</u> elp
	S→ 🐂 → 🗇 🖬 X 🖬 🖻 A ∽ ♀ → II = 💐 🗳 🗧 🎽 🛠 🧟 🐊	>>
	Normal Form1 Project - Project 1	×
	General General	
🖉 Project1	- Form1 (Code)	□×
		_
(General)	(Declarations)	•
(General)	(Declarations)	<u> </u>
(General)	(Declarations)	
(General) Dim k	(Declarations)	
(General)	(Declarations)	- - -
(General) pim k ≡ <u>≡</u> ↓	(Declarations)	

- การเขียนโปรแกรมใน visual Basic

* ประกาศตัวแปร



	Project1 - Form1 (Code)	_ 🗆 ×
Fo	orm 🔽 Load	v
	Private Sub Form_Load()	•
	MSComm1.PortOpen = True	
	End Sub	

รูป 6.15 แสดงขั้นตอนการทำงาน

6.4 การอ่านค่าสัญาน Analog ที่มากกว่า 1 ช่องสัญาน





- ตัวโปรแกรมใน Microcontroller



รูป 6.16 โปรแกรมอ่านสัญญาน analog 2 Chanel แล้วส่งออก port rs232 * ขั้นตอนใน Visual Basic

🔎 Project1 - Form1 (Code)			_ 🗆 🗙
(General)	-	(Declarations)	•
Dim k, L Dim ANALOG\$			
			🕨 li.

4	Project1 - Form1 (Code)		_ 🗆 ×
<mark>لو</mark> ا	Project1 - Form1 (Code)		_ 🗆 🗙
M	SComm1	OnComm	
	Private Sub MSComm1_On ANALOG\$ = MSComm1.In	Comm() put	
	End Sub		



รูป 6.17 ขั้นตอนและวิธีการเขียนโปรแกรมรับสัญญาน Analog 2 Chanel

6.5 การแสดงสัญญาน Analog บนระนาบ x,y โดยใช้ picture box



รูป6.18 picture box แสดงสัญาญ analog บนระนาบ X,Y

* ขั้นตอนการพัฒนาโปรแกรม

👼 Project1 - Form1	Code)	
(General)	(Declarations)	•
Dim k, L Dim ANALOO Dim xscal,	yscal1, yscal2	

<u>.</u>	Project1 - Form1 (Code)	_ 🗆 🗙
Fo	orm 🔽 Load	-
	Private Sub Form_Load() MSComm1.PortOpen = True Picture1.Scale (0, 255)-(1000, 0)	
F	Picture1.Scale (0, 255)-(1000, 0)	F

🚒 Project1 - Form1 (Code)		_ 🗆 🗙	
M	5Comm1	💽 OnComm	•
	Private Sub MS ANALOG\$ = MS	SComm1_OnComm() SComm1.Input	4
E	End Sub		

🚒 Project1 - Form1 (Code)	_ 🗆 ×
Timer 1 Timer	•
<pre>Private Sub Timer1_Timer() k = Mid\$(ANALOG\$, 1, 3) L = Mid\$(ANALOG\$, 4, 7) Text1 = k Text2 = L yscal1 = VAL(k) yscal2 = VAL(L) Picture1.PSet (xscal, yscal1), QBColor(12) Picture1.PSet (xscal, yscal2), QBColor(4) xscal = xscal + 1 If xscal = 1000 Then Picture1.C1s xscal = 0 End If End Sub</pre>	
	<i>[</i>],

รูป 6.19 แสดงขั้นตอนการเขียนโปรแกรม

6.6 การใช้ เครื่องมือ Micro help gauge control แสดงผลสัญญาน Analog

Components	×
Controls Designers Insertable Objects	
 Kodak Image Scan Control Kodak Image Thumbnail Control LEDMeter ActiveX Control module LHTagBrowser Mabry Volume Control Marquee Control Library MicroHelp Gauge Control MicroHelp Key State Control Microsoft ActiveX Plugin Microsoft ADO Data Control 6.0 (SP4) (OLEDB) 	
Microsoft Calendar Control 8.0 Microsoft Chart Control 6.0 (SP4) (OLEDB)	Browse
MicroHelp Gauge Control Location: C:\WINDOWS\SYSTEM\GAUGE32.OCX	
OK	Cancel <u>A</u> pply



รูป6.20 การเพิ่ม Microhelp gauge control ใน tool box

tool	property	defind
Label1	caption	Analog chanel1
	Fore color	น้ำตาล
Label2	caption	Analog chanel1
	Fore color	น้ำตาล
Gauge1	forecolor	สีส้ม
	Min	0
	max	255
	style	vertical
Gauge2	forecolor	น้ำเงิน
	Min	0
	max	255
	style	vertical

1		v			
91 G 21	การแสดงสถากการเ	Analog @08	Microheln	annen	control
리 U.스 I			IVII OI OI OI OI	yauyu	CONTROL
01	~ ~ ~	0		0 0	

* เปลี่ยนโปรแกรมใน Sub timer



รูป6.22 โปรแกรมเพื่อแสดงสัญญาน Analog ที่ Microhelp gauge

6.7 การเขียนข้อมูลเพื่อควบคุมการทำงานของ Microcontroller



รูป 6.23 แสดงการควบคุมและตรวจสอบการทำงานของ Microcontroller



รูป 6.24 Hardware ที่รองรับการทำงานของโปรแกรมตามรูป 6.23



tool	property	define
Shape1	backstyle	opaque
	name	Shape1
	shape	circle
Shape2	backstyle	opaque
	name	Shape2
	shape	circle
Shape3	backstyle	opaque
	name	Shape3
	shape	circle
Shape4	backstyle	opaque
	name	Shape4
	shape	circle
Command1	caption	Led1_ON
	backcolor	&H0080FF80&
	style	graphic
	name	Command1
Command2	caption	Led1_OFF
	backcolor	&H0080FF80&
	style	graphic
	name	Command2
Command3	caption	Led2_ON
	backcolor	&H0080FF80&
	style	graphic
	name	Command3
Command4	caption	Led2_OFF
	backcolor	&H0080FF80&
	style	graphic
	name	Command4

tool	property	Define
Command5	caption	Led3_ON
	backcolor	&H0080FF80&
	style	Graphic
	name	Command5
Command6	caption	Led1_OFF
	backcolor	&H0080FF80&
	style	Graphic
	name	Command6
Command7	caption	Led4_ON
	backcolor	&H0080FF80&
	style	Graphic
	name	Command7
Command8	caption	Led4_OFF
	backcolor	&H0080FF80&
	style	Graphic
	name	Command8
TEXT1	FORLORE	&H00C00000&
	FONT	BOLD/14
	NAME	TEXT1
	TEXT	<i>i</i> cn
TEXT2	FORLORE	&H00C00000&
	FONT	BOLD/14
	NAME	TEXT2
	TEXT	<i>i</i> cn
TEXT3	FORLORE	&H00C00000&
	FONT	BOLD/14
	NAME	TEXT3
	TEXT	6639
TEXT4	FORLORE	&H00C00000&
	FONT	BOLD/14
	NAME	TEXT4
	TEXT	u 31

- ประกาศตัวแปร

🚒 Project1 - Form1 (Code)		_ 🗆 ×
(General)	(Declarations)	•
Dim code\$		

- โปรแกรมที่ Sub Form_load

<mark>ہے</mark> ا	Project1 - Form1 (Code)	
Fo	orm 🔽 Load	•
	Private Sub Form_Load() MSComm1.PortOpen = True End Sub	

- โปรแกรมที่ sub command_click

<u>.</u>	Project1 - Form1 (Code) 📃 🗖 🗙
C	ommand1 Click 🔽
	Private Sub Command1_Click() MSComm1.Output = Chr\$(1) End Sub
	Private Sub Command2_Click() MSComm1.Output = Chr\$(2) End Sub
	Private Sub Command3_Click() MSComm1.Output = Chr\$(3) End Sub
	Private Sub Command4_Click() MSComm1.Output = Chr\$(5) End Sub
	Private Sub Command5_Click() MSComm1.Output = Chr\$(7) End Sub

```
Private Sub Command6_Click()
MSComm1.Output = Chr$(4)
End Sub
Private Sub Command7_Click()
MSComm1.Output = Chr$(6)
End Sub
Private Sub Command8_Click()
MSComm1.Output = Chr$(8)
End Sub
```

<u>.</u>	Project1 - Form1 (Code)	_ 🗆 ×
Μ	ISComm1 💽 OnComm	-
	<pre>Private Sub MSComm1_OnComm() pode\$ = MSComm1.Input Select Case code\$ Case "led1_ON" Shape1.BackColor = QBColor(12) Text1 = code\$ Case "led1_OFF" Shape1.BackColor = QBColor(0) Text1 = code\$ Case "led2_ON" Shape2.BackColor = QBColor(12) Text2 = code\$ Case "led2_OFF" Shape2.BackColor = QBColor(0) Text2 = code\$ Case "led3_ON" Shape3.BackColor = QBColor(12) Text3 = code\$ Case "led3_OFF" Shape3.BackColor = QBColor(0) Text3 = code\$ Case "led4_ON"</pre>	
	Shape4.BackColor = QBColor(12) Text4 = code\$ Case "led4_OFF" Shape4.BackColor = QBColor(0) Text4 = code\$ End Select End Sub	

#include <16F877.h> #use delay(clock=4000000) #fuses HS,NOWDT #use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7) #define LED1 pin_c0 #define LED2 pin_c1 #define LED3 pin_c2 #define LED4 pin_c3 boolean flg_int; char str; #int_RDA RDA_isr() { str=getc(); flg_int=true; } void main() { output_low(LED1); output_low(LED2); output_low(LED3); output_low(LED4); printf("samraeng:"); flg_int=true; enable_interrupts(INT_RDA); enable_interrupts(global); while(true) { if(flg_int) { if(str==1) printf("led1_ON"); { output_high(LED1); else if(str==2) { output_low(LED1); printf("led1_OFF"); output_high(LED2); else if(str==3) { printf("led2_ON"); else if(str==4) { output_low(LED2); printf("led2_OFF"); else if(str==5) { output_high(LED3); printf("led3_ON"); else if(str==6) { output_low(LED3); printf("led3_OFF");

{

output_high(LED4);

else if(str==7)

}

printf("led4_ON");

}

}

}

}

}

}

```
else if(str==8) { output_low(LED4); printf("led4_OFF"); }
else { }
flg_int=false;
}
}
```