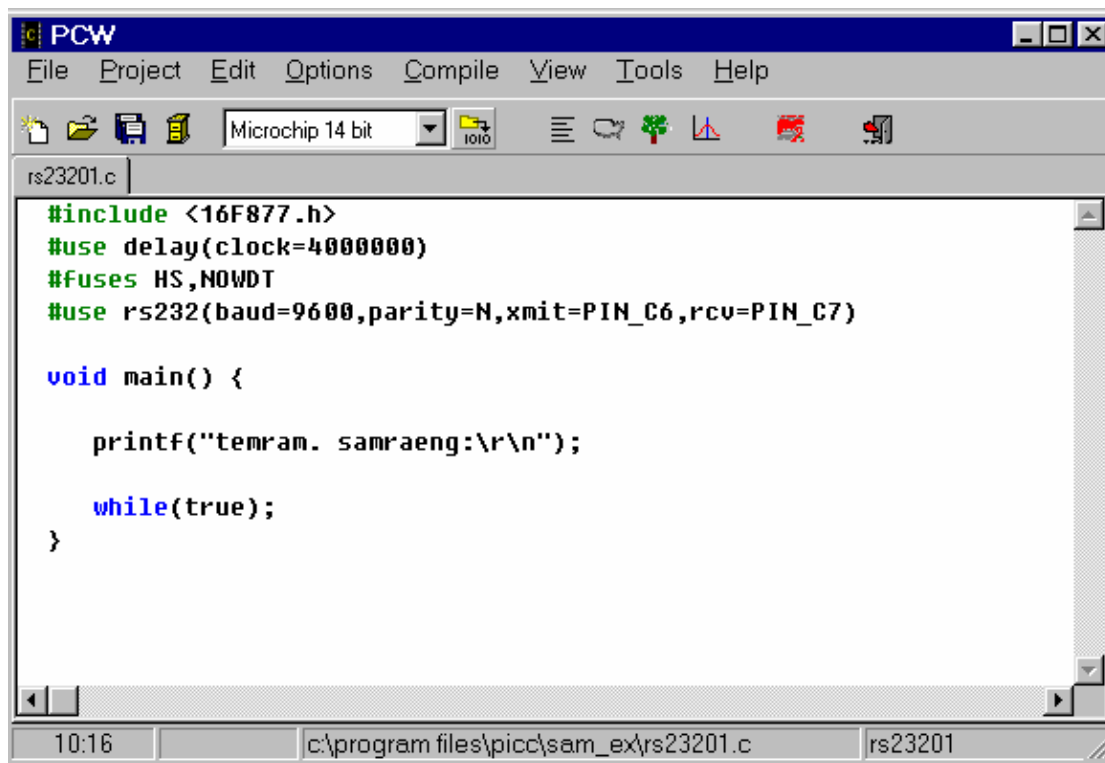


พื้นฐานการอ่านและเขียนข้อมูลระหว่างโปรแกรม visual basic กับ pic16f877 microcontroller

ก่อนที่เราจะใช้ โปรแกรม Visual Basic ติดต่อกับ Plc ผู้เขียนขอเริ่มต้นแนวคิดพื้นฐานในเรื่องของการควบคุมและการจัดการข้อมูลระหว่าง computer กับ ชุดควบคุมที่ใช้ ไมโครคอนโทรลเลอร์เป็นตัวหลักในการควบคุมระบบอัตโนมัติ และคาดหวังว่าจะก่อให้เกิดแรงจูงใจสำหรับนักพัฒนาโปรแกรมมือใหม่ทั้งหลายหันมาให้ความสนใจกับระบบควบคุมในโลปัจจุบันและต่อเนื่องไปในอนาคต

เหตุผลที่ผู้เขียนเลือกที่จะใช้ pic16f877 microcontroller เนื่องจากว่าการเขียนโปรแกรมโดยใช้ pic c compiler มีคำสั่งที่ใช้ในการติดต่อสื่อสารผ่านทาง Port RS232 ที่ใช้งานได้ง่ายเนื่องจาก ตัว compiler ได้สร้างฟังก์ชันทางด้านการติดต่อสื่อสารผ่านทาง port rs232 ไว้เป็นที่เรียบร้อยแล้ว

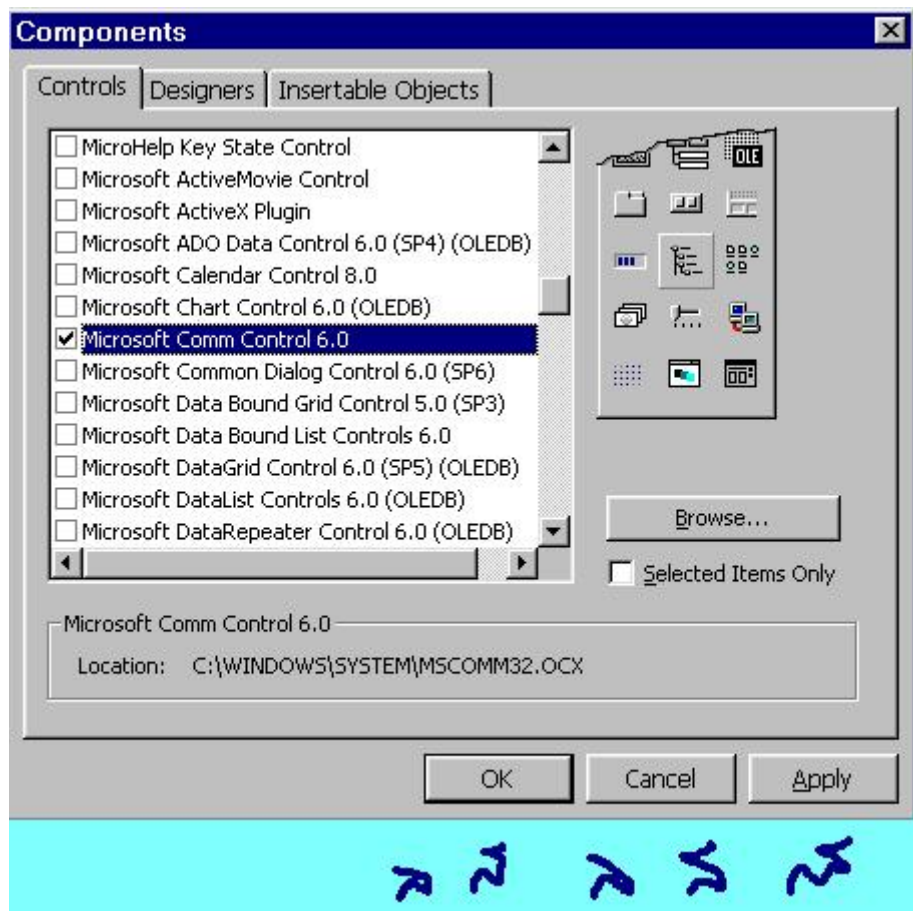
6.1 รูปร่างหน้าต่างของ pic c compiler



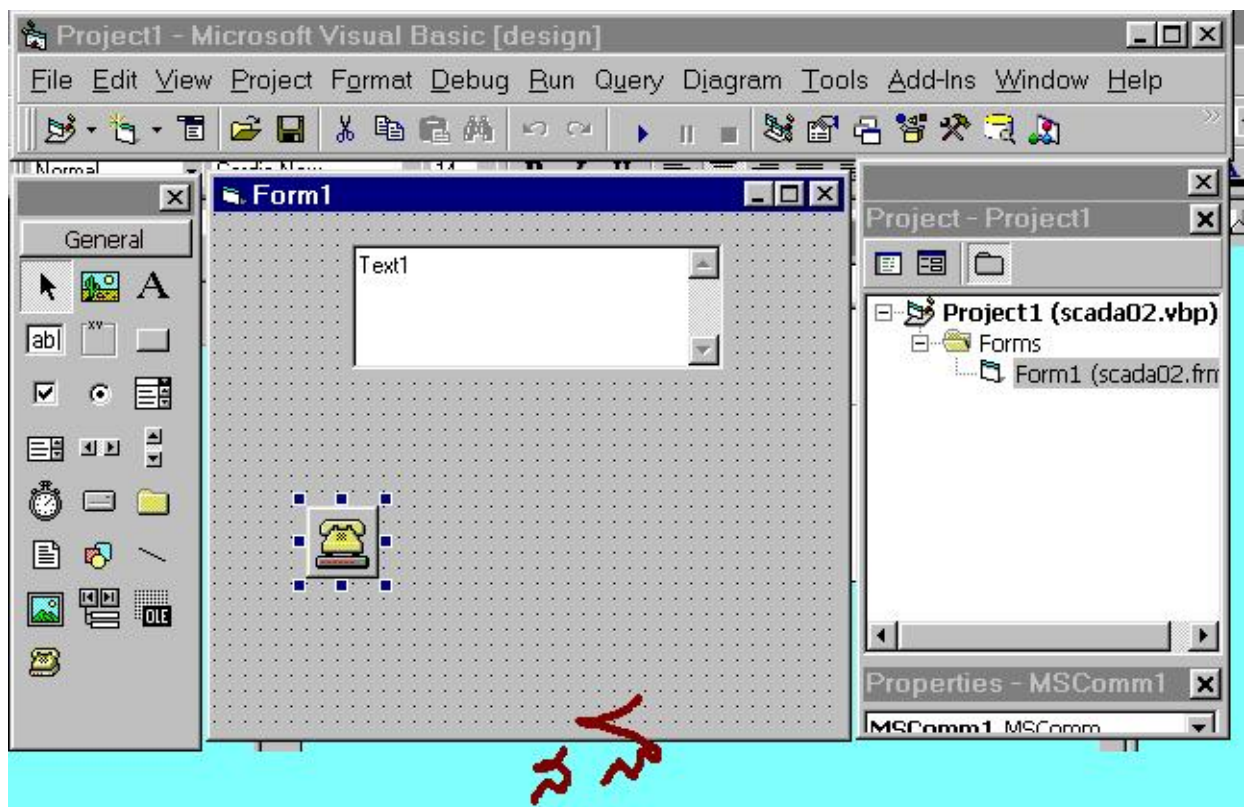
รูป 6.1 editor of pic c compiler

6.2 การใช้ Visual Basic รับข้อความจาก microcontroller ผ่านทาง port rs232

- วงจร Rs232

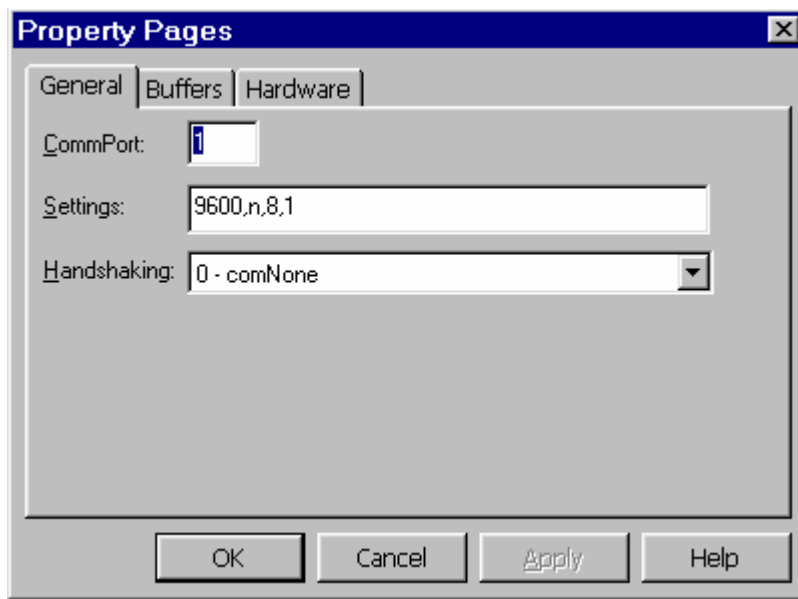


รูป 6.4 check box Microsoft Comm Control 6.0

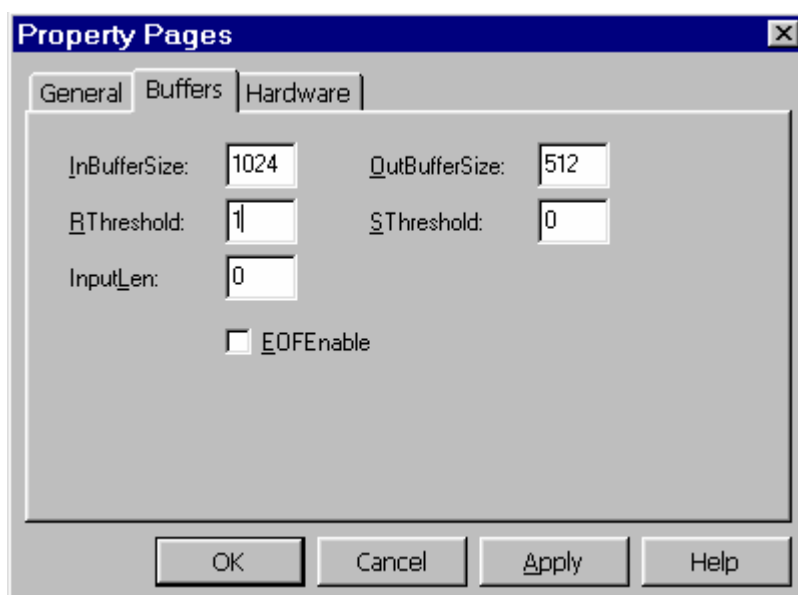


รูป 6.5 แสดงหน้าตาของ Mscomm control

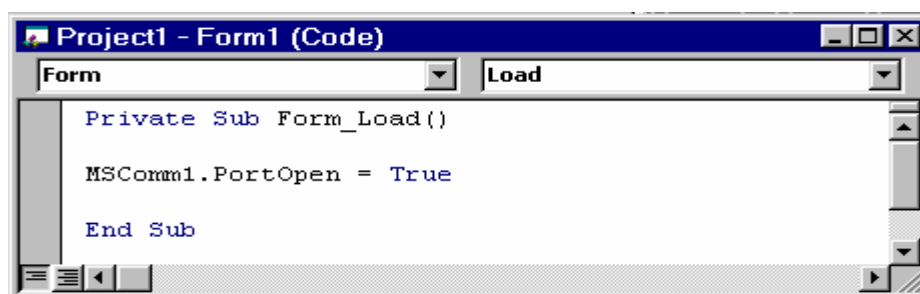
- กำหนดคุณสมบัติของ Mscomm control



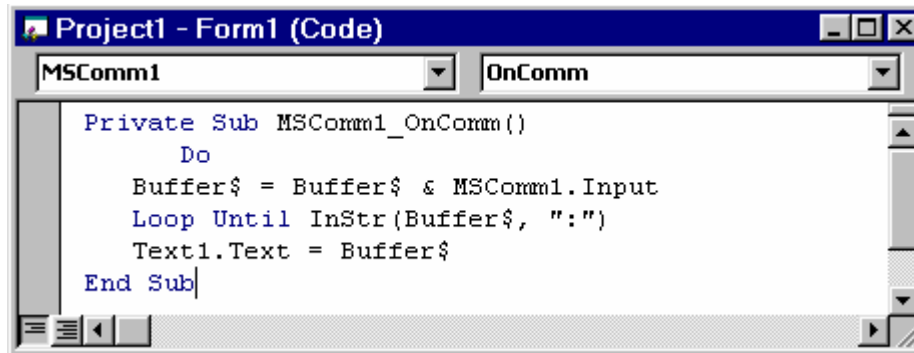
รูป6.6 กำหนดคุณสมบัติทั่วไป



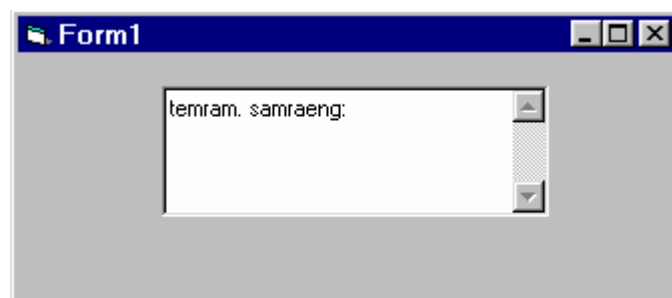
รูป6.7 กำหนดคุณสมบัติของ Buffers



รูป 6.7 เขียนโปรแกรมที่ sub form_load เพื่อเริ่มการทำงานของ com port

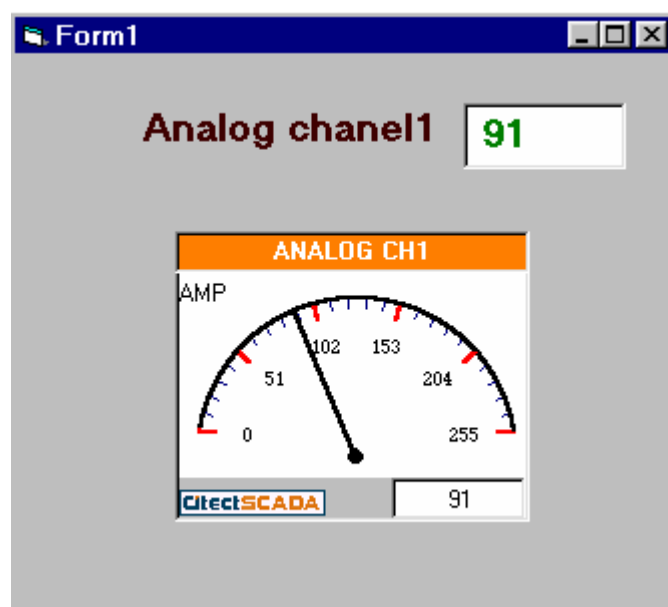


รูป 6.8 เขียนโปรแกรมที่ sub Mscomm_oncomm()



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

6.3 การอ่านค่าสัญญาณ Analog จาก microcontroller



รูป 6.10 การแสดงค่าสัญญาณ analog โดยใช้ textbox และ TechMeter

- ตัวโปรแกรมใน Microcontroller ตามรูป 6.11

```

#include <16F877.h>
#define delay(clock=4000000)
#define fuses HS,NOWDT
#define rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7)
byte value;
void main() {

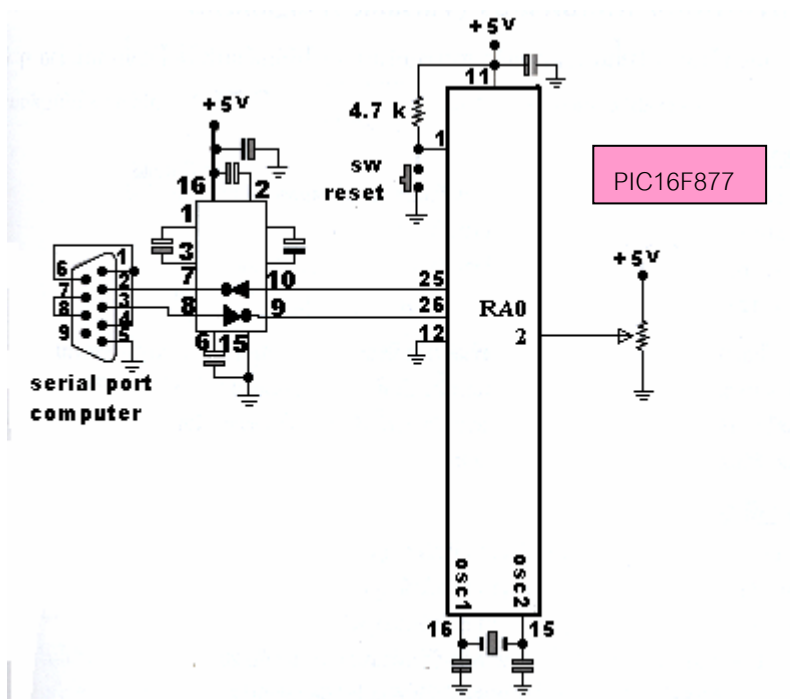
    setup_adc_ports(ALL_ANALOG );
    setup_adc(ADC_CLOCK_DIV_2);
    while(true)
    {
        set_adc_channel( 0 );
        value = read_adc();// read analog chanel0 (ra0)
        printf("%3u ",value);
        delay_ms(100);
    }

}

```

รูป6.11 โปรแกรมอ่านสัญญาณ analog แล้วส่งออก port rs232

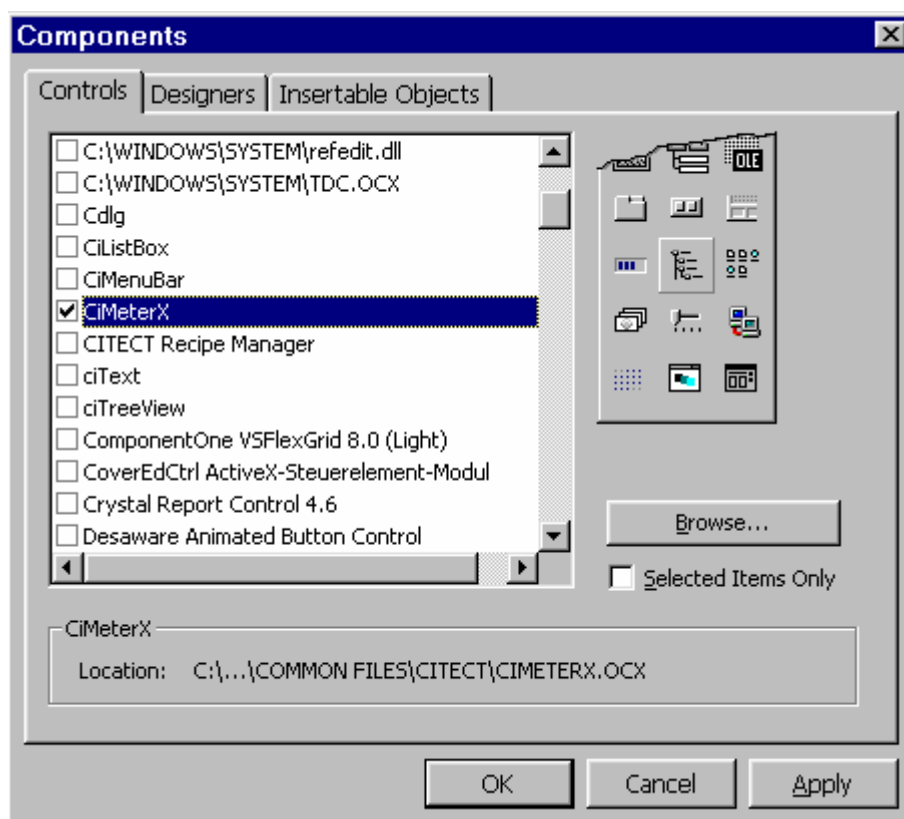
- วงจร



รูป6.12 วงจรสำหรับอ่านสัญญาณ analog

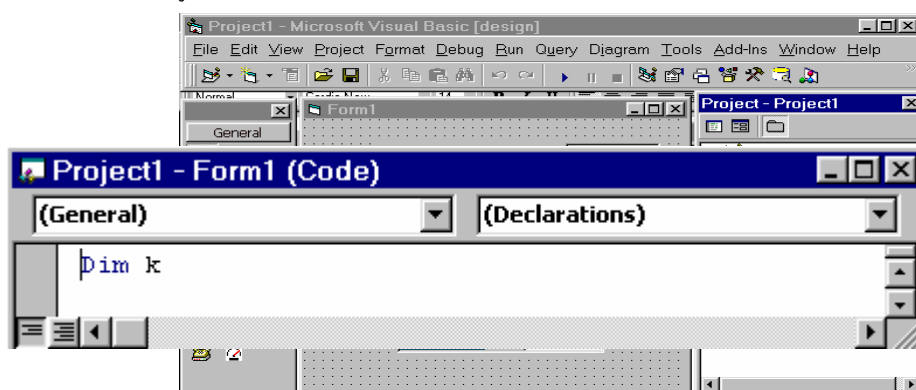
- เครื่องมือใน Visual Basic

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



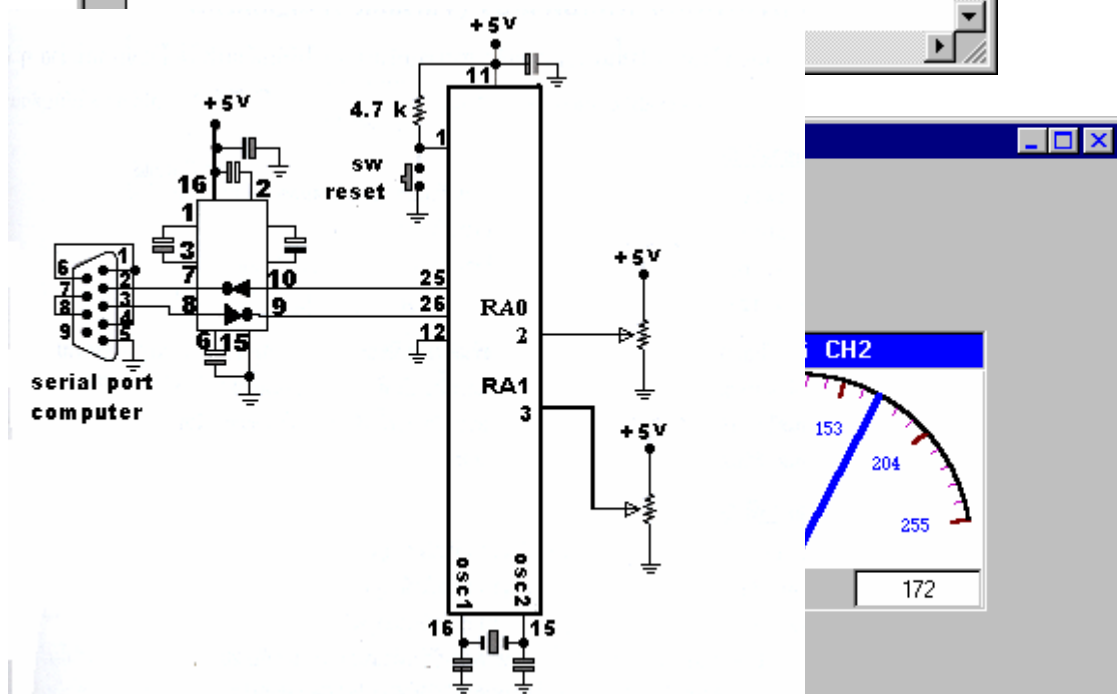
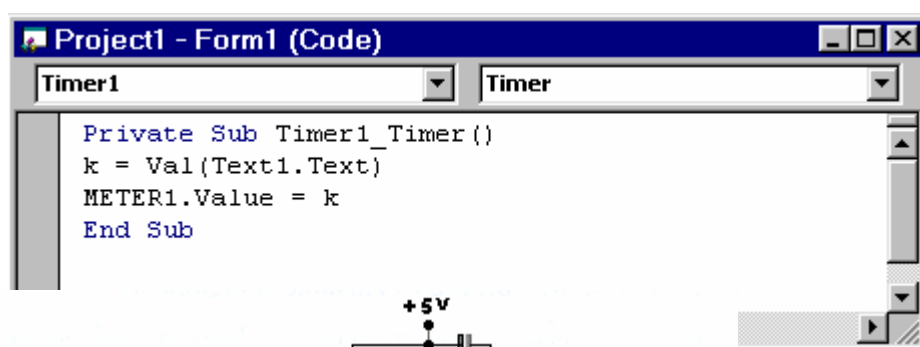
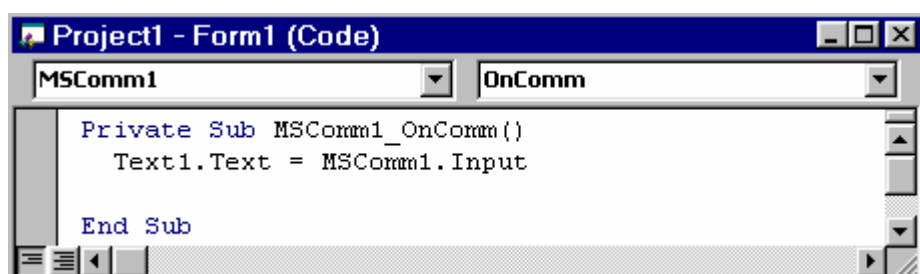
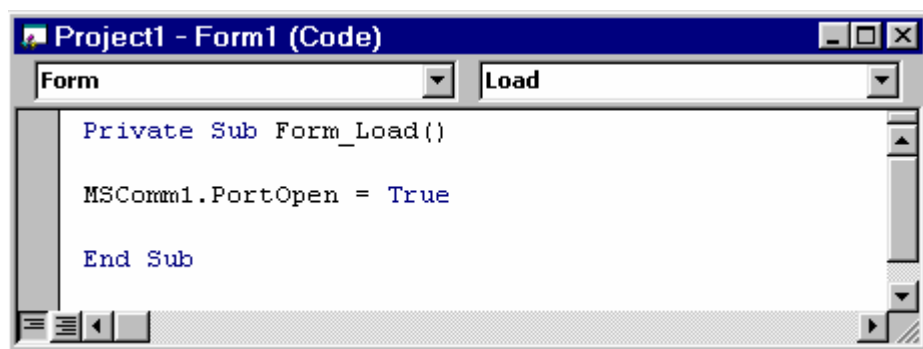
รูป 6.13 add tool TechMeter

รูป 6.14 แสดงเครื่องมือต่างๆ บน form1

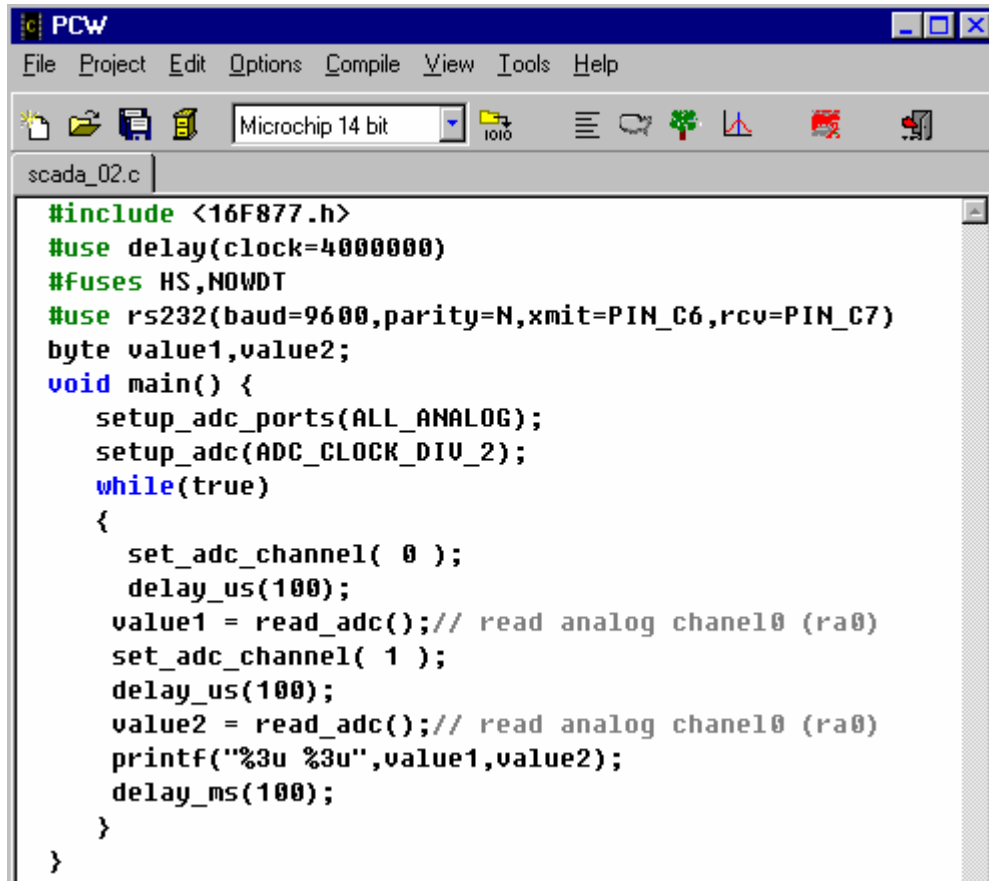


- การเขียนโปรแกรมใน visual Basic

* ประกาศตัวแปร



- ตัวโปรแกรมใน Microcontroller



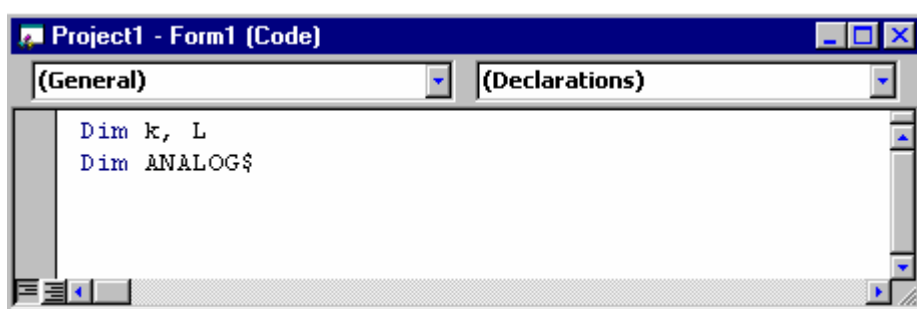
```

#include <16F877.h>
#use delay(clock=4000000)
#fuses HS,NOWDT
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7)
byte value1,value2;
void main() {
    setup_adc_ports(ALL_ANALOG);
    setup_adc(ADC_CLOCK_DIV_2);
    while(true)
    {
        set_adc_channel( 0 );
        delay_us(100);
        value1 = read_adc();// read analog chanel0 (ra0)
        set_adc_channel( 1 );
        delay_us(100);
        value2 = read_adc();// read analog chanel0 (ra0)
        printf("%3u %3u",value1,value2);
        delay_ms(100);
    }
}

```

รูป 6.16 โปรแกรมอ่านสัญญาณ analog 2 Chanel แล้วส่งออก port rs232

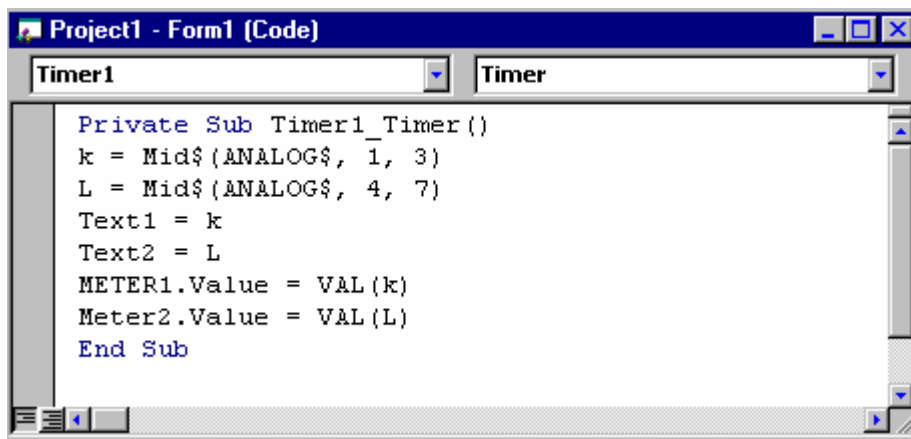
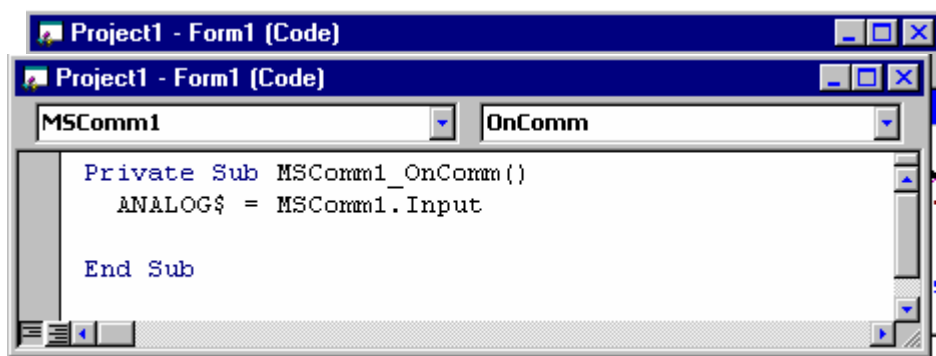
* ขั้นตอนใน Visual Basic



```

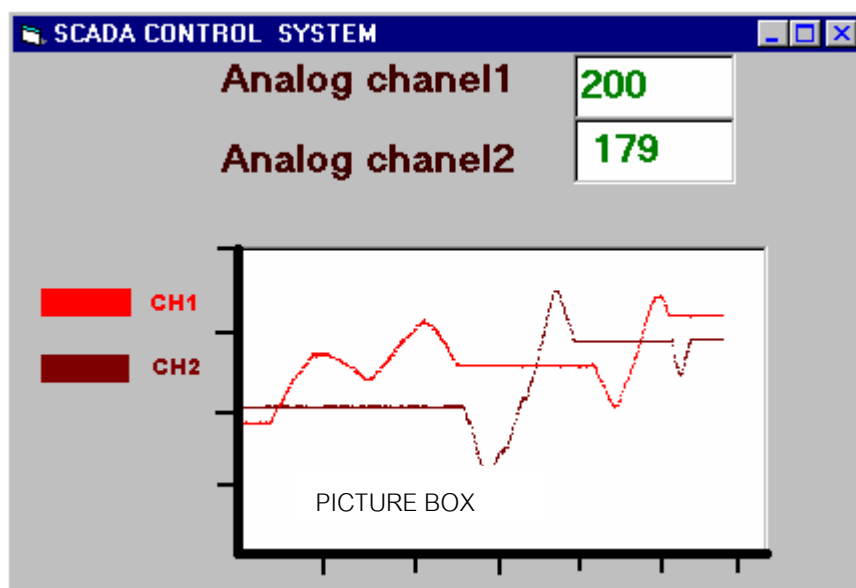
Dim k, L
Dim ANALOG$

```



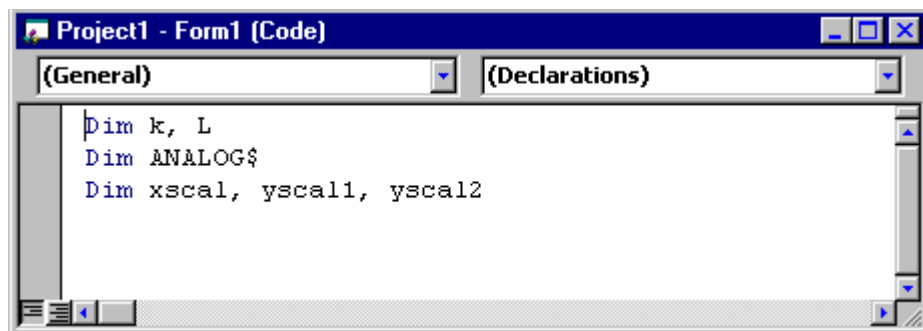
รูป 6.17 ขั้นตอนและวิธีการเขียนโปรแกรมรับสัญญาณ Analog 2 Chanel

6.5 การแสดงสัญญาณ Analog บนระนาบ x,y โดยใช้ picture box



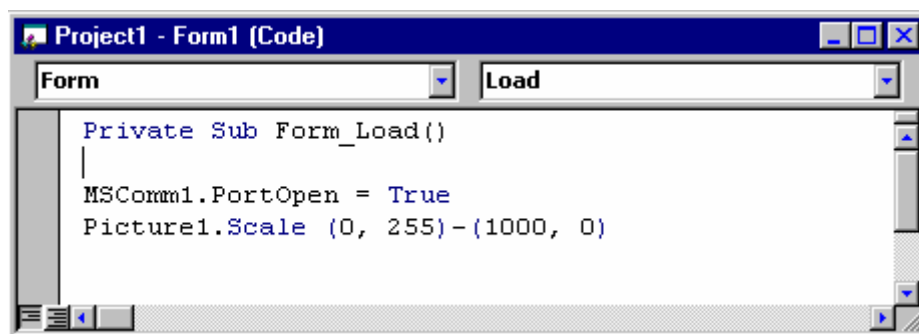
รูป 6.18 picture box แสดงสัญญาณ analog บนระนาบ X,Y

* ขั้นตอนการพัฒนาโปรแกรม



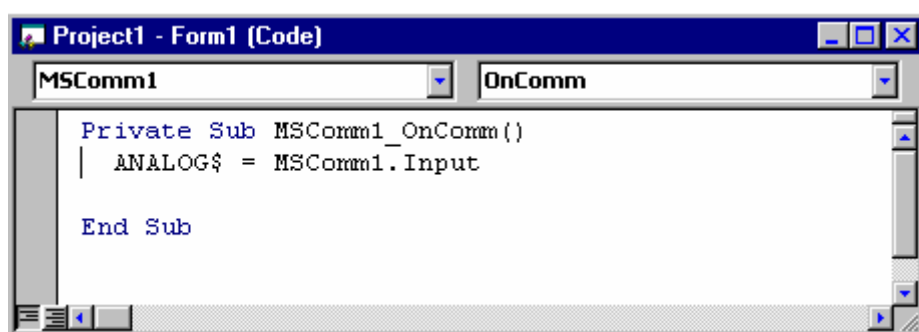
```
Project1 - Form1 (Code)
(General) (Declarations)

Dim k, L
Dim ANALOG$
Dim xscal, yscal1, yscal2
```



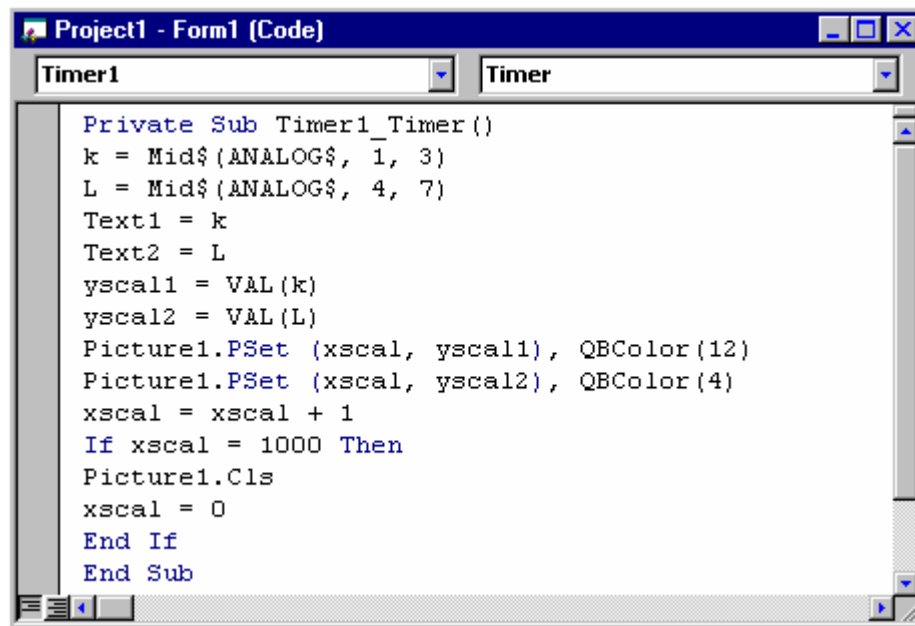
```
Project1 - Form1 (Code)
Form Load

Private Sub Form_Load()
    MSComm1.PortOpen = True
    Picture1.Scale (0, 255)-(1000, 0)
End Sub
```



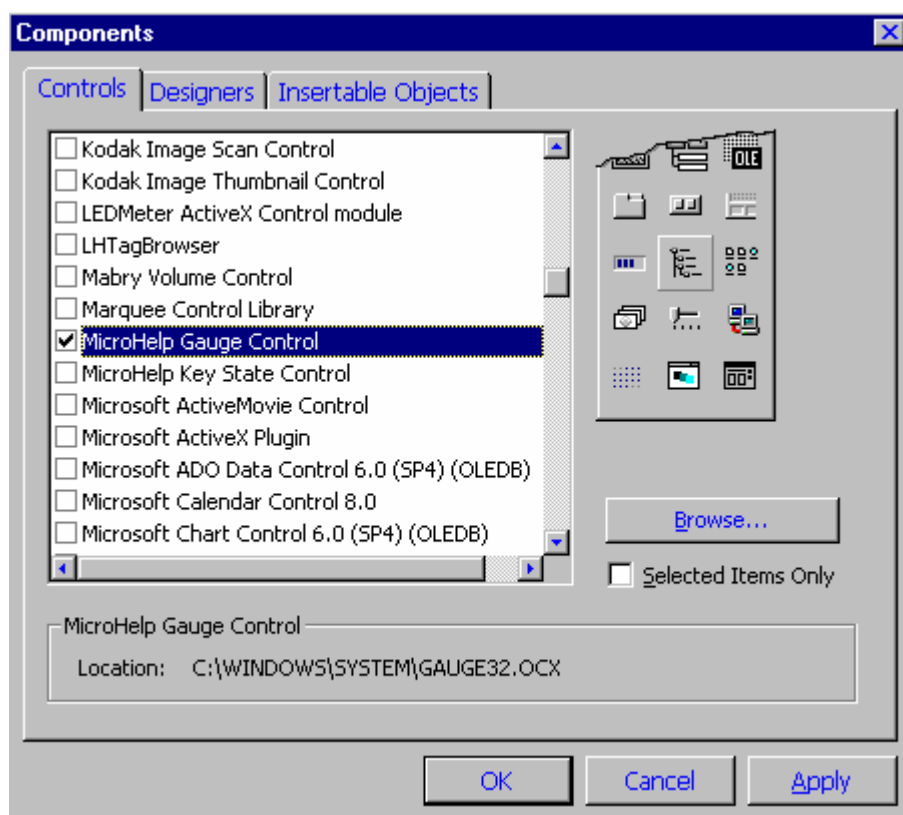
```
Project1 - Form1 (Code)
MSComm1 OnComm

Private Sub MSComm1_OnComm()
    ANALOG$ = MSComm1.Input
End Sub
```

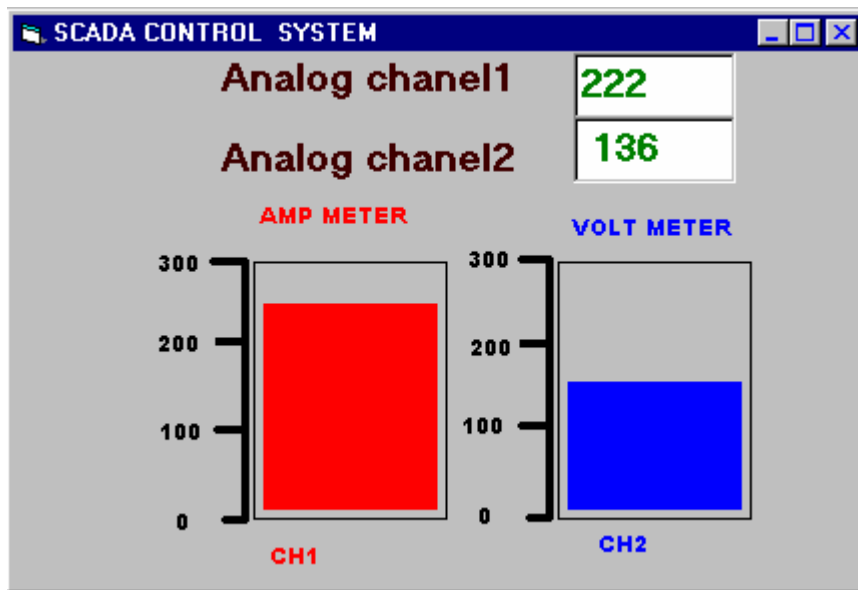


รูป 6.19 แสดงขั้นตอนการเขียนโปรแกรม

6.6 การใช้เครื่องมือ Micro help gauge control แสดงผลสัญญาณ Analog



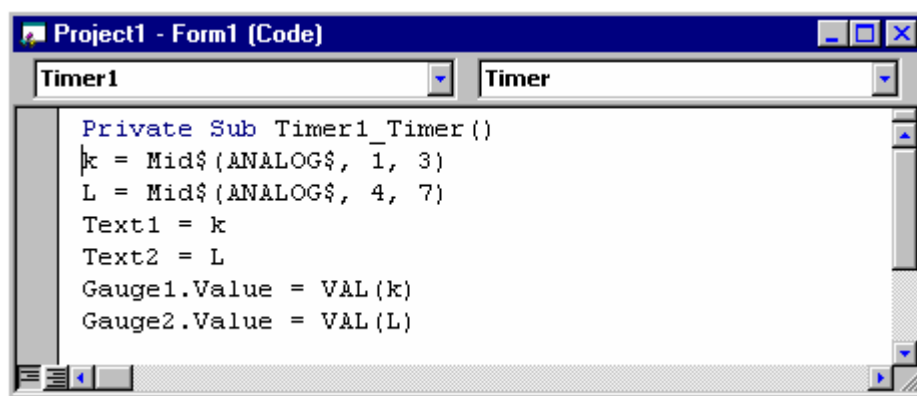
รูป 6.20 การเพิ่ม Microhelp gauge control ใน tool box



รูป 6.21 การแสดงสัญญาณ Analog ด้วย Microhelp gauge control

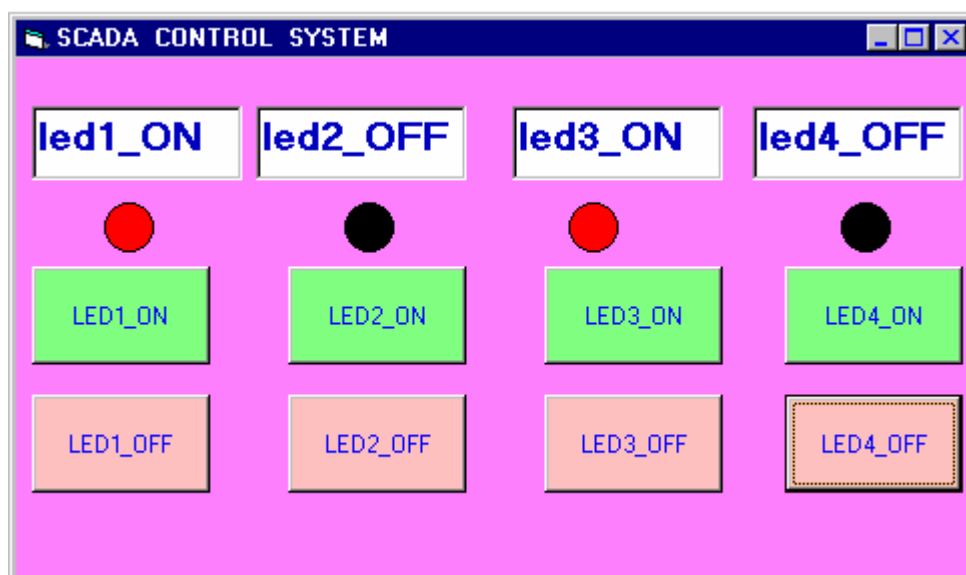
tool	property	defind
Label1	caption	Analog chanel1
	Fore color	น้ำตาล
Label2	caption	Analog chanel1
	Fore color	น้ำตาล
Gauge1	forecolor	สีส้ม
	Min	0
	max	255
	style	vertical
Gauge2	forecolor	น้ำเงิน
	Min	0
	max	255
	style	vertical

* เปลี่ยนโปรแกรมใน Sub timer

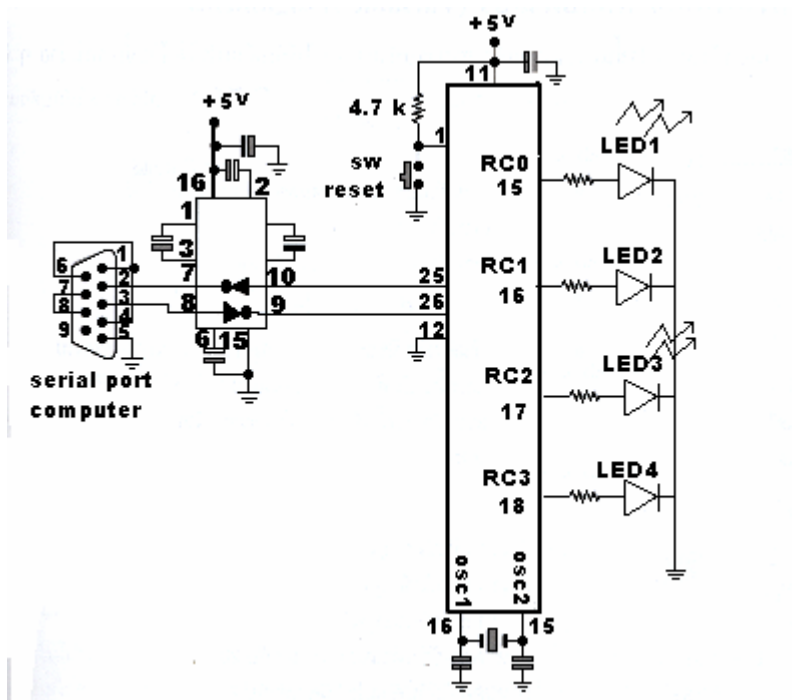


รูป 6.22 โปรแกรมเพื่อแสดงสัญญาณ Analog ที่ Microhelp gauge

6.7 การเขียนข้อมูลเพื่อควบคุมการทำงานของ Microcontroller

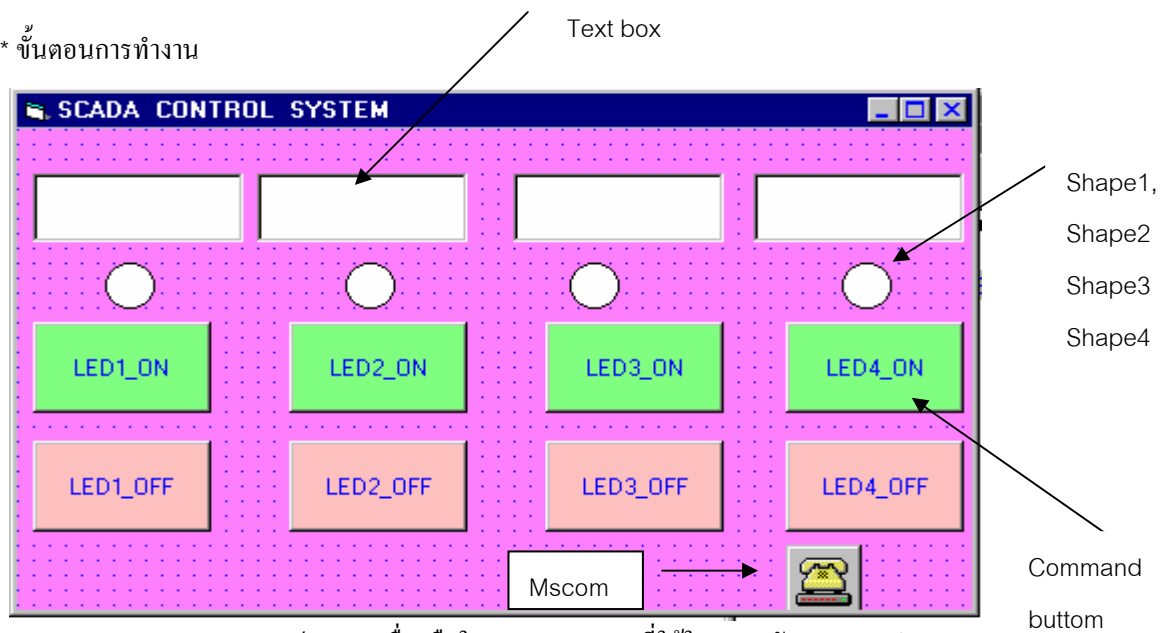


รูป 6.23 แสดงการควบคุมและตรวจสอบการทำงานของ Microcontroller



รูป 6.24 Hardware ที่รองรับการทำงานของโปรแกรมตามรูป 6.23

* ขั้นตอนการทำงาน



รูป 6.25 เครื่องมือใน Visual Basic ที่ใช้ในการสนับสนุนการทำงาน

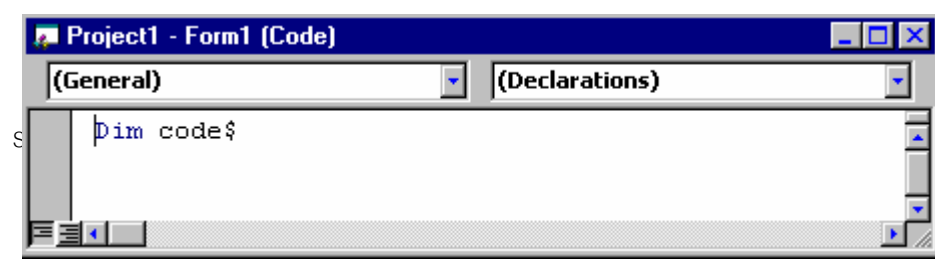
tool	property	define
------	----------	--------

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

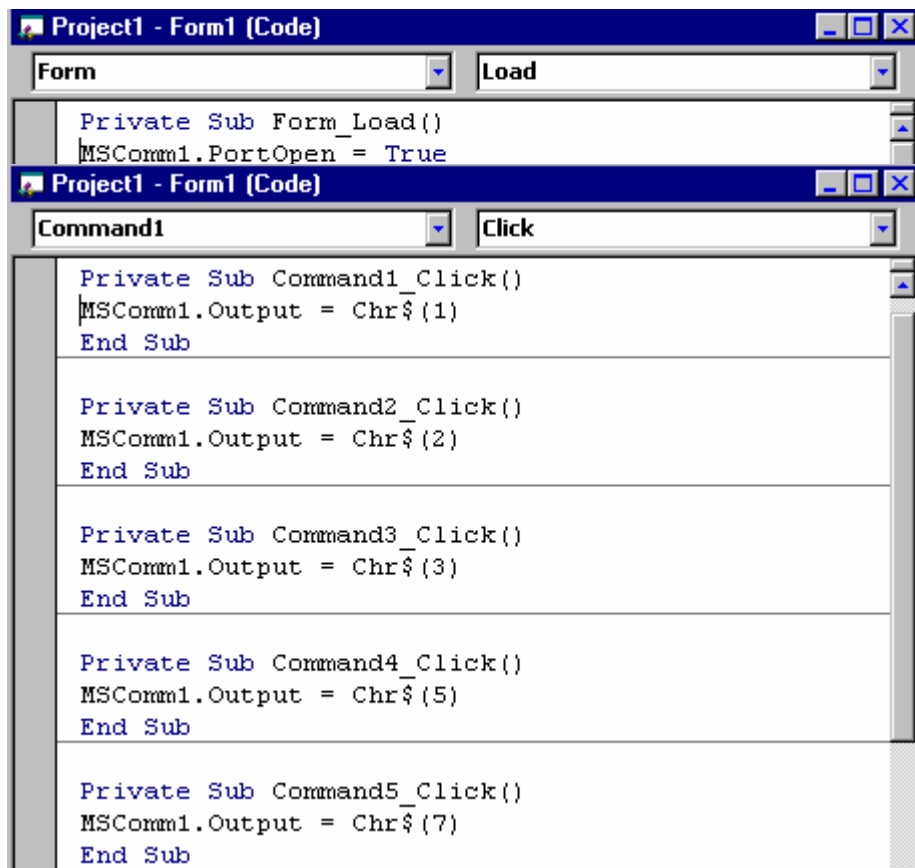
	backcolor	&H0080FF80&
	style	Graphic
	name	Command5
Command6	caption	Led1_OFF
	backcolor	&H0080FF80&
	style	Graphic
	name	Command6
Command7	caption	Led4_ON
	backcolor	&H0080FF80&
	style	Graphic
	name	Command7
Command8	caption	Led4_OFF
	backcolor	&H0080FF80&
	style	Graphic
	name	Command8
TEXT1	FORLORE	&H00C00000&
	FONT	BOLD/14
	NAME	TEXT1
	TEXT	""
TEXT2	FORLORE	&H00C00000&
	FONT	BOLD/14
	NAME	TEXT2
	TEXT	""
TEXT3	FORLORE	&H00C00000&
	FONT	BOLD/14
	NAME	TEXT3
	TEXT	""
TEXT4	FORLORE	&H00C00000&
	FONT	BOLD/14
	NAME	TEXT4
	TEXT	""

- ประกาศตัวแปร



- โปรแกรมที่ Sub Form_load

- โปรแกรมที่ sub command_click

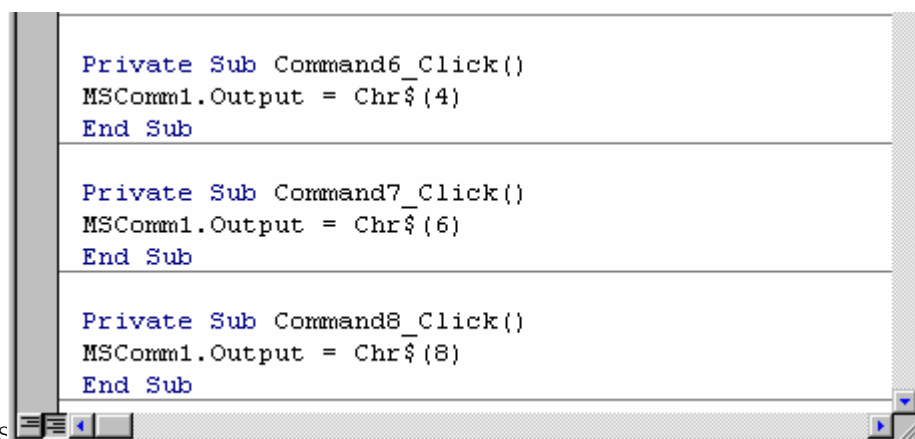


The screenshot shows two overlapping windows from the Visual Basic IDE. The top window is titled "Project1 - Form1 (Code)" and has a dropdown menu set to "Form" and a sub-menu set to "Load". It contains the following code:

```
Private Sub Form_Load()  
MSComm1.PortOpen = True  
End Sub
```

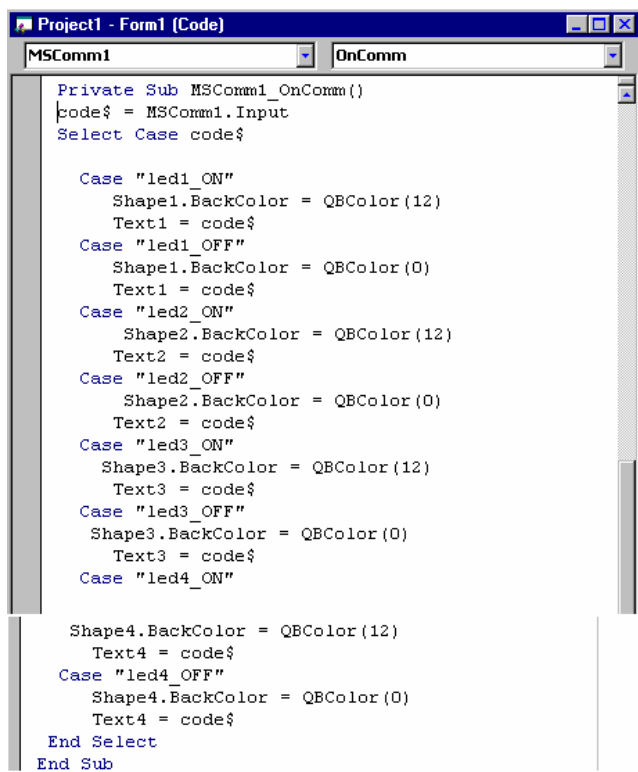
The bottom window is also titled "Project1 - Form1 (Code)" but has a dropdown menu set to "Command1" and a sub-menu set to "Click". It contains five sub-routines for Command1 through Command5:

```
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
```



This block shows the continuation of the code from the previous block, specifically the sub-routines for Command6 through Command8:

```
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
```



```

#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) { output_high(LED1); printf("led1_ON"); }
        else if(str==2) { output_low(LED1); printf("led1_OFF"); }
        else if(str==3) { output_high(LED2); 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"); }
        else if(str==7) { output_high(LED4); printf("led4_ON"); }
        else if(str==8) { output_low(LED4); printf("led4_OFF"); }
        else { }
        flg_int=false;}}}

```