Topic 03: Introduction to Communication by UART

- **UART**: Universal Asynchronous Receiver/Transmitter, 通用非同步收發傳輸器，是電腦硬體的一部分，將資料由串行通信(Serial)與並行通信(Parallel)間作傳輸轉換。
  並行通信的優點是資料傳輸快，因為處理器與周邊裝置間有 8 條以上的資料線連結，處理器能一次輸出或接收 8 個位元以上資料，但缺點是不適合長距離傳輸，因為易受雜訊干擾，且成本較高。
  UART 通常用在與其他通訊介面（如 EIA RS-232）的連結上。(https://zh.wikipedia.org/wiki/UART)
  - 包括了 RS232、RS449、RS423、RS422 和 RS485 等接口(port)標準規範和匯流排標準規範，即 UART 是非同步串行通信口的總稱。
  - RS232、RS449、RS423、RS422 和 RS485 等，是對應各種非同步串行通信埠的接口標準和匯流排標準
    - ✓ 它規定了通信埠的電氣特性、**傳輸速率**、連接特性和**接口的機械特性**等內容。實際上是屬於通信網絡中的實體層（Physical Layer）的概念，**與通信協議沒有直接關係**。
    - ✓ 通信協議，是屬於通信網絡中的資料鏈結層（Data Link Layer）的概念。
    - ✓ COM 是 PC 上，**非同步(異步)**串行通信口的簡寫或稱**串列埠 Serial Port**。由於歷史原因，IBM 的 PC 外部接口配置為 RS232，成為實際上的 PC 界默認標準。所以，現在 PC 機的 COM 均為 RS232。若配有多個**非同步(異步)**串行通信埠，則分別稱為 **COM1**、**COM2**、... 。
- **同步**:是介面中的資料傳輸線路，都依據同一條時脈線路的信號來動作，例如 I^2C 介面、SPI 介面
- **非同步(異步)**: 是每條資料傳輸線路有自己的傳輸步調，不倚賴獨立的時脈線路，或根本沒有獨立的時脈線路，通常是將時脈信號埋藏在自己的傳輸封包中，規律性的每隔一段時間發出。
- 在 USB 介面還沒出現前，個人電腦是用 RS-232 介面連接滑鼠或數據機的。
  - RS-232 的好處是簡單便宜，但傳輸率不高（早期最快為 115.2kbps），且一個 RS-232 介面只能連接一個裝置，不像 RS-422、RS-485 可以同時連接多個裝置。
- 為了讓 PC 能跟 Arduino 通訊傳輸，因此 Arduino 開發板上設置一個 USB-to-UART 晶片，該晶片可以把 USB 傳輸翻譯成 UART 傳輸，同時也能反向翻譯，使 Arduino 晶片能跟 PC 連通。
  - 有了這個介面轉譯晶片後，再搭配上驅動程式，即可讓 Windows 將外接裝置認定成一個串列埠裝置，而 Windows 上原本支援 COM 埠的軟體，能夠用原本對 COM 埠的傳輸方式傳輸，但實際上卻轉成 USB 協定方式傳輸，一路傳到 USB-to-UART 晶片上時，才轉成原本的 UART 傳輸，並送入微控制器晶片內，反之亦然。
  - 有時候 USB-to-UART 也稱為 USB-to-COM，Windows 中的裝置管理員也是顯示 COM 埠，例如 COM5、COM6 等（用軟體模擬出的串列埠，硬體本質是 USB）

- 另外也有人稱為 USB-to-TTL，其實也是類似意思，但 TTL 是指 Transistor-Transistor Logic，是一種 0、1 邏輯信號準位的規範，嚴格而論在此用這個詞不太妥當，但也已約定成俗而慣用。

✧ 串列埠（Serial Port， RS232）有線通訊原理介紹:

串列埠通信實際上是串列埠按位元（bit）發送和接收位元組。儘管比按位元組（byte）的並行通信慢，但是串列埠可以在使用一根線發送資料的同時用另一根線接收資料。它很簡單並且能夠實現遠距離通信。通信使用3根線完成：地線、發送、接收。總之，「把一整排的資料(byte)按照順序排列，逐一(each bit)送到遠方，接收方再用同樣的順序將資料排回去」，這就是串列通訊的原始精神。

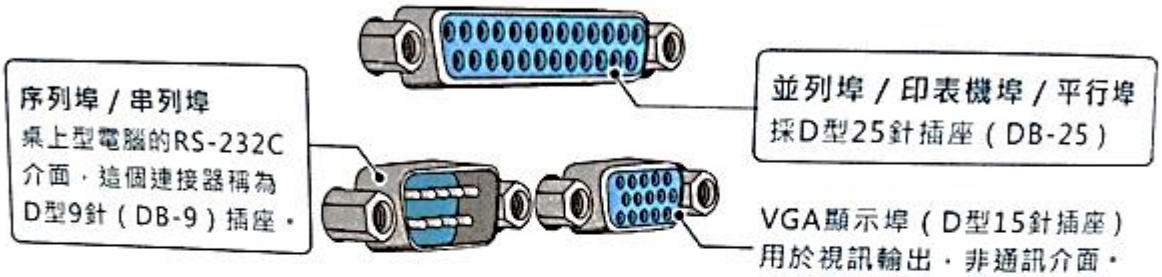
由於串列埠通信是非同步的，埠能夠在一根線上發送資料同時在另一根線上接收資料。其他線用於握手，但是不是必須的。串列埠通信最重要的參數是Baud Rate(波特率)、data bits(資料位元)、stop bit(停止位元)和Parity(奇偶校驗)。**對於兩個進行通信的埠，這些參數必須匹配**：

1. **串列傳輸速率**：這是一個衡量通信速度的參數。它表示每秒鐘傳送的bit 的個數。
   - 例如300 Baud Rate表示每秒鐘發送300 個bit。當我們提到時鐘週期時，我們就是指串列傳輸速率例如如果協定需要4800 串列傳輸速率，那麼時鐘是4800Hz。這意味著串列埠通信在資料線上的採樣率為4800Hz。通常電話線的串列傳輸速率為14400、28800 和36600。
   - 串列傳輸速率可以遠遠大於這些值，但是**串列傳輸速率和距離成反比**。高串列傳輸速率常常用於放置的很近的儀器間的通信，典型的例子就是GPIB 設備的通信。
2. data bits(資料位元)：這是衡量通信中實際資料位元的參數。當電腦發送一個資訊包，實際的資料不會是8 位元的，標準的值是5、7 和8 位。如何設置取決於你想傳送的資訊。
   - 比如，標準的ASCII 碼是0～127（7 位）。擴展的ASCII 碼是0～255（8位）。如果資料使用簡單的文本（標準 ASCII 碼），那麼每個資料包使用7 位元資料。每個包是指一個位元組，包括開始/停止位元，資料位元和奇偶校驗位。
   - 由於實際資料位元取決於通信協定的選取，術語“包”指任何通信的情況。
3. stop bit(停止位元)：用於表示單個包的最後一位。典型的值為1，1.5 和2 位。由於資料是在傳輸線上定時的，並且每一個設備有其自己的時鐘，很可能在通信中兩台設備間出現了小小的不同步。
   - 因此停止位不僅僅是表示傳輸的結束，並且提供電腦校正時鐘同步的機會。適用於停止位的位數越多，不同時鐘同步的容忍程度越大，但是資料傳輸率同時也越慢。
4. Parity(奇偶校驗位元)：在串列埠通信中一種簡單的**檢錯方式**。有四種檢錯方式：偶、奇、高和低。當然沒有校驗位也是可以的。對於偶和奇校驗的情況，串列埠會設置校驗位元（資料位元後面的一位元），用一個值確保傳輸的資料有偶個或者奇個邏輯高位。
   - 例如，如果資料是011，那麼對於偶校驗，校驗位元為0，保證邏輯高的位元數是偶數個。如果是奇校驗，校驗位元為1，這樣就有3 個邏輯高位。高位元和低位元不真正的檢查資料，簡單置位元邏輯高或者邏輯低校驗。這樣使得接收設備能夠知道一個位元的狀態，有機會判斷是否有雜訊干擾了通信或者是否傳輸和接收資料是否不同步。

Ref. “Arduino 互動設計入門” 趙英傑 著 旗標公司

RS-232 序列埠

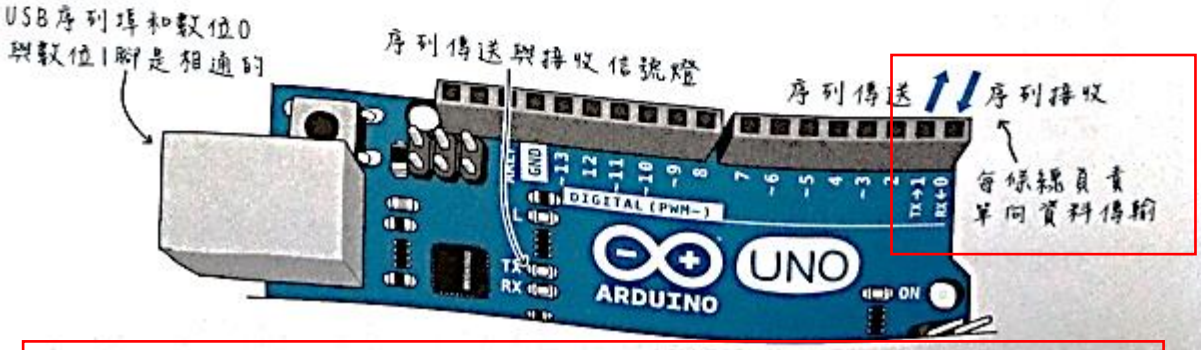
RS-232 是最早廣泛使用的序列埠標準（它其實有不同的版本，目前使用的 RS-232-C 問世於 1969 年，其中的 RS 代表 Recommended Standard），目前許多桌上型電腦仍配備 RS-232C 介面，在 Windows 系統軟體中，**序列介面稱為 COM**，並以 COM1, COM2, ...等編號標示不同的介面，每個 COM 介面同時只能接一個裝置。



在 USB 介面普及之前，許多周邊裝置都採用 RS-232C 介面，例如：滑鼠、條碼掃瞄器、遊戲搖桿、數據機...等等。

完整的 RS-232C 連接器有 25 個腳位，但大多數的裝置不需要複雜的傳輸設定，所以 IBM PC 採用 9 個針腳的 D 型連接器（簡稱 DB9），其中最重要的三個接腳是**數據傳送**（Transmitter，簡稱 **Tx**）、**數據接收**（Receiver，簡稱 **Rx**）和**接地**（Ground，簡稱 **GND**）。

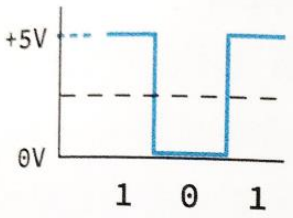
ATmega 微處理器有內建兩個序列連結的腳位，分別連接到 Arduino 板子**第 0 腳（Rx，接收）和第 1 腳（Tx，傳送）**。



然而，ATmega 處理器無法直接和電腦上或其他 RS-232C 設備相連，因為 RS-232C 的訊號電壓跟一般的數位裝置不同。

一般數位 IC 的 0 與 1 訊號的電壓準位，分別是 0V 和 5V（或電源電壓），這種準位又稱為 **TTL 或邏輯準位**。RS-232C 的電壓準位介於 ±3V~±15V，**高於 3V 的準位為 0**，也稱為 **Space（空格）；低於 -3V 的準位為 1**，又稱為 **Mark（標記）**，-3V 和 +3V 之間的訊號則是「不確定值」。

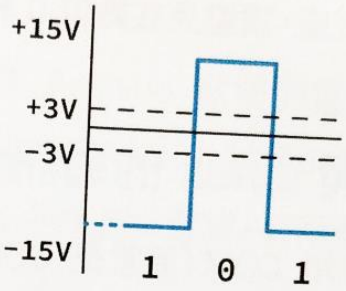
TTL訊號的電壓



高於電源電壓的一半，代表1

低於0.8V，代表0

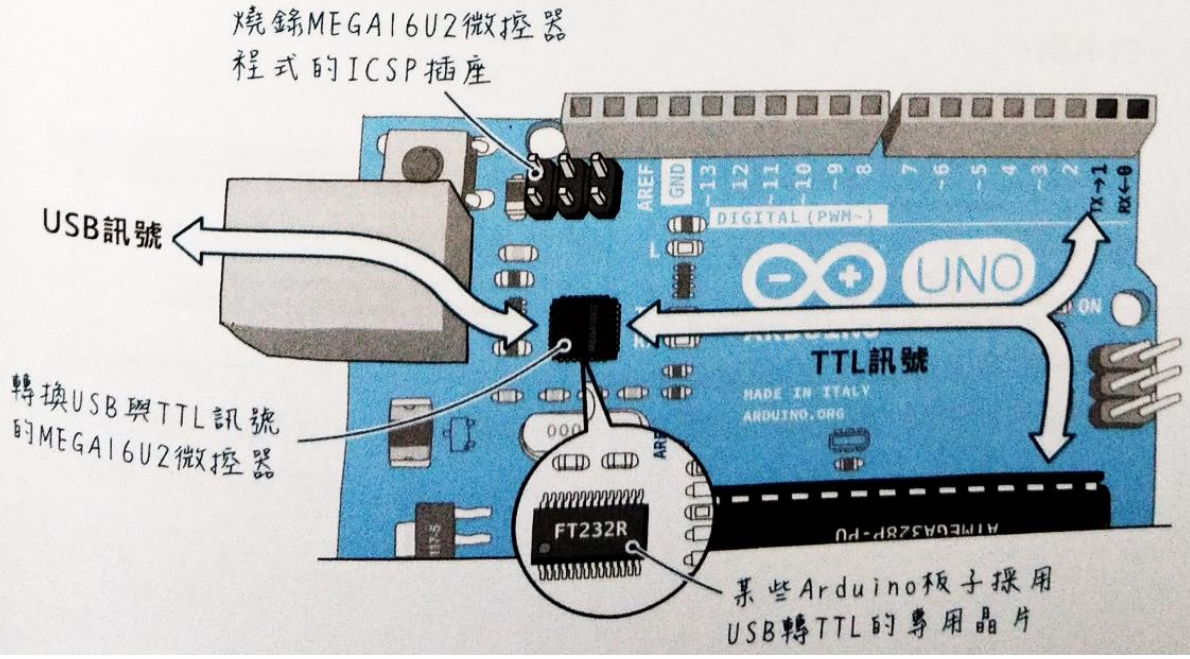
RS-232訊號的電壓



因此，Arduino 和 RS-232C 設備之間，需要加裝一個訊號準位轉換元件（一般稱為 **TTL 轉 RS-232**），才能相連。在實作上，通常採用 MAX232 準位轉換 IC，或者用電晶體電路來轉換。

USB 序列埠

早期的 Arduino 板子採用 RS-232 介面，後來改用 USB；某些 Arduino 控制板採用序列通訊轉換晶片（如：FT232R）來轉換 USB 與 TTL 序列訊號，Uno 板使用 MEGA16U2 微控器。

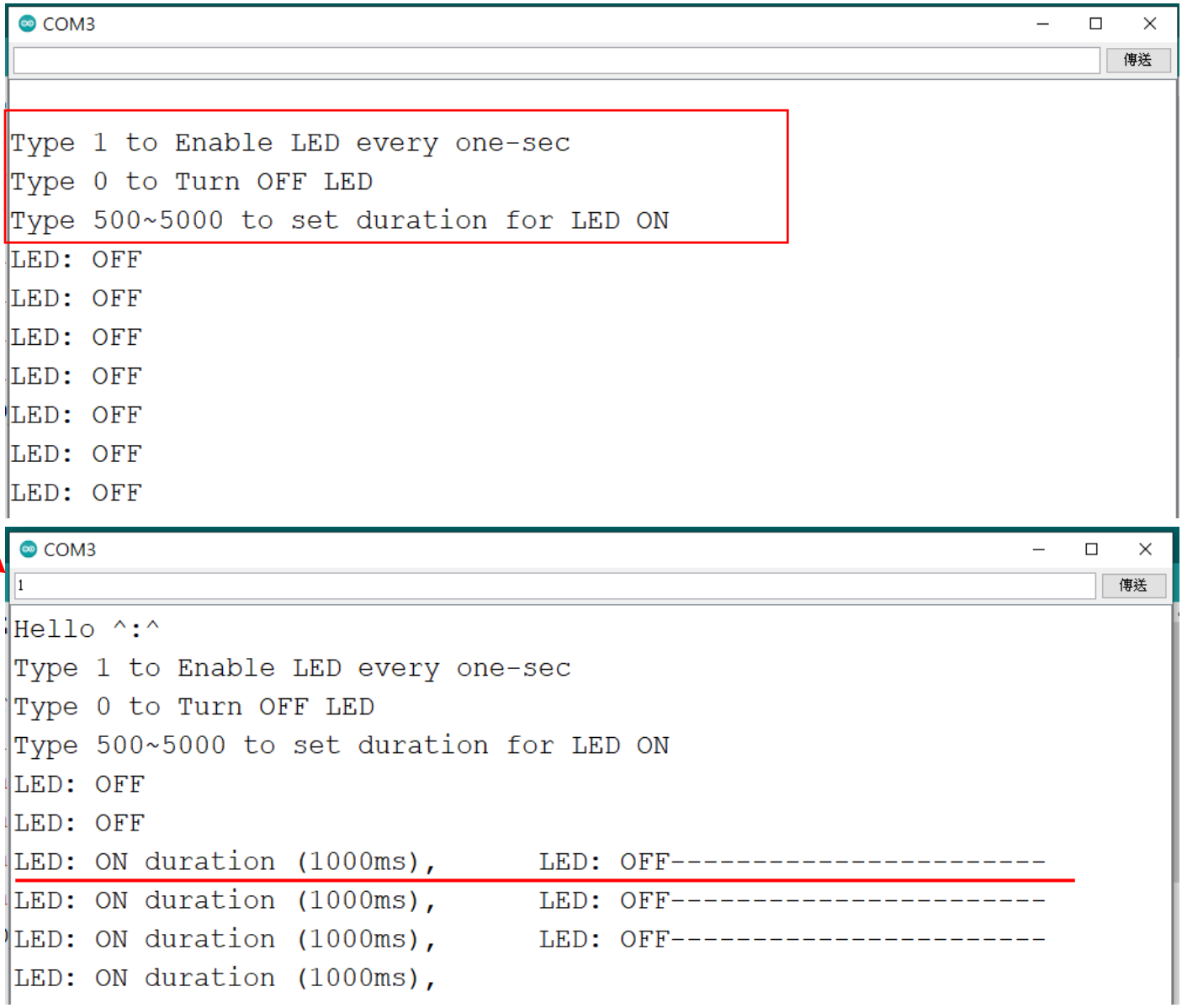


MEGA16U2 微控器在此負責銜接 USB 介面和 Arduino 的數位 0 與 1 腳。如果有需要的話，我們還可以改寫它的程式，讓電腦將此控制板看待成滑鼠、鍵盤、電玩搖桿或 MIDI 數位音樂介面。只不過，Arduino 工具本身並不提供燒錄此微控器程式的功能，要透過另一個叫做 FLIP 的燒錄程式。

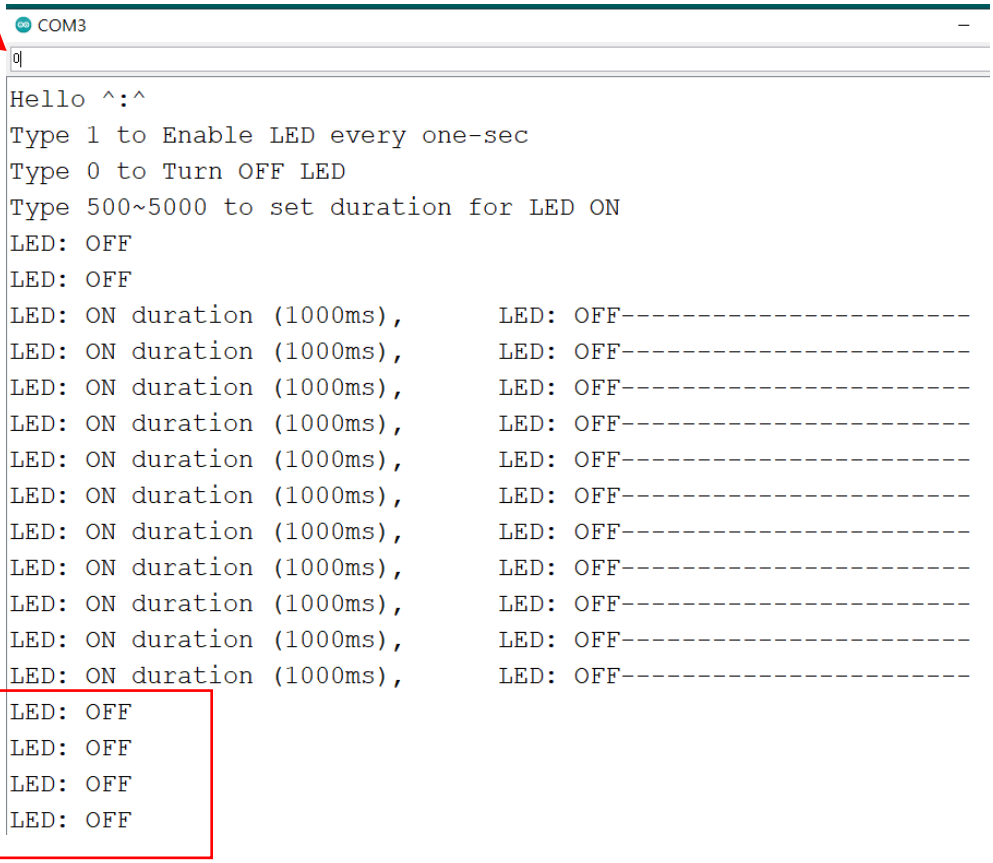
由於 USB 序列埠是 Arduino 程式編輯器傳送程式碼給微處理器（以及下文介紹的「監控」）的管道，**請避免在數位 0 和 1 兩個接腳銜接其他元件**。

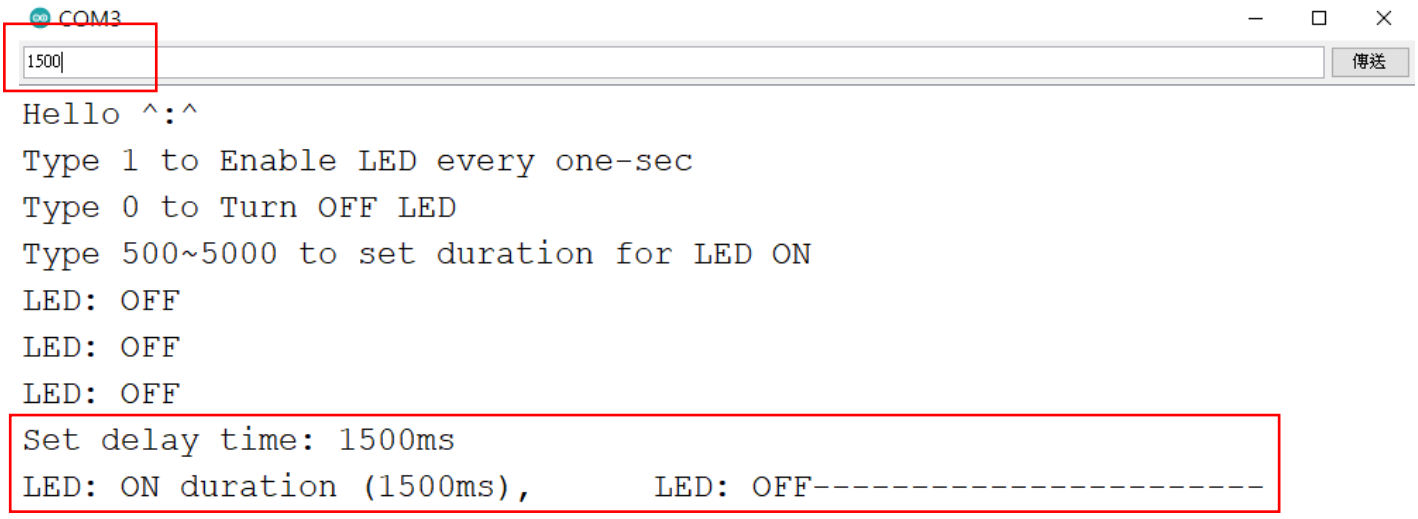
Ex03_1_LED

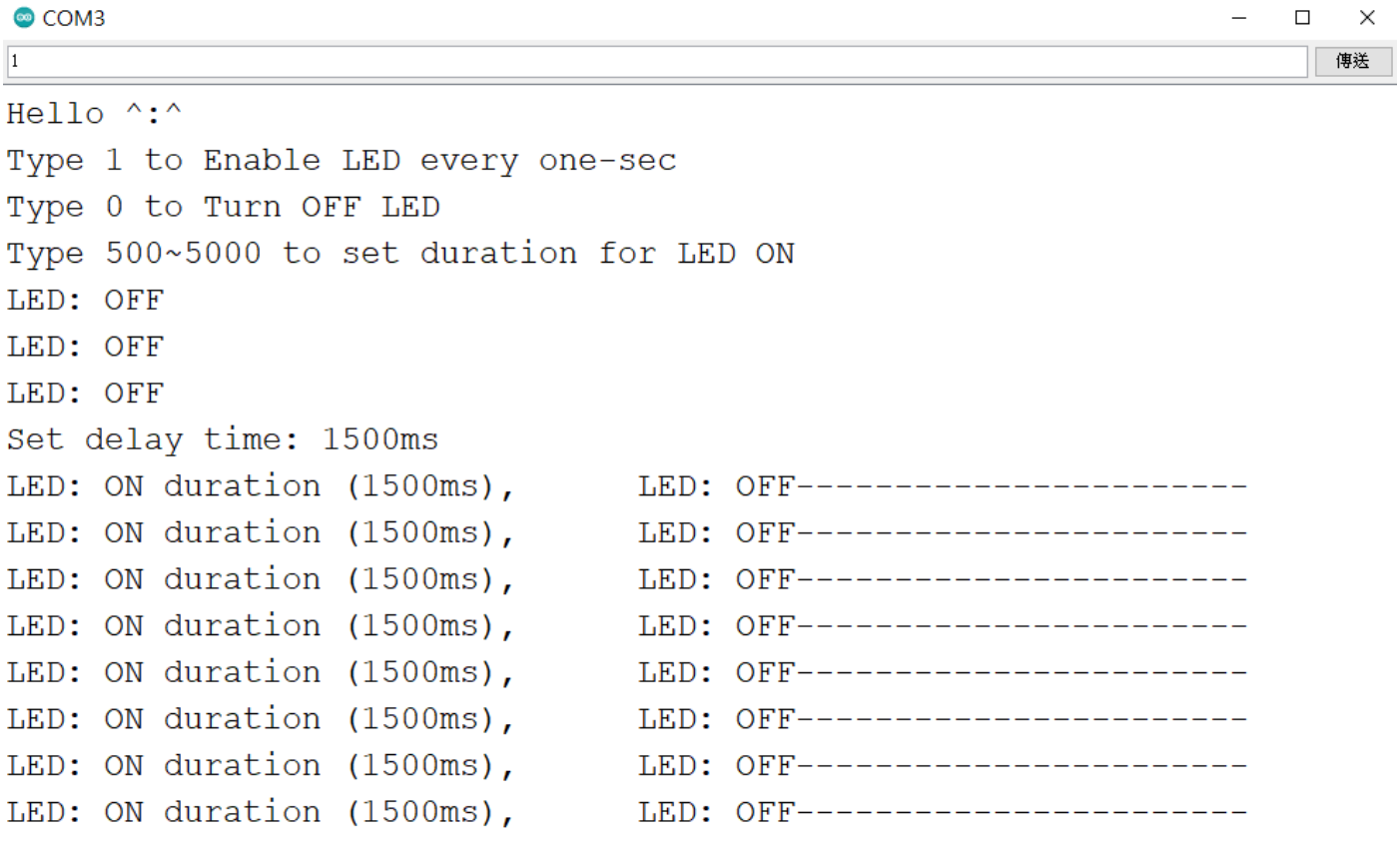
```
const byte LEDpin=13;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Hello ^:^");
  Serial.println("Type 1 to Enable LED every one-sec");
  Serial.println("Type 0 to Turn OFF LED");
  Serial.println("Type 500~5000 to set duration for LED ON");
  pinMode(LEDpin,OUTPUT);
}
char val=0;
String numberStr;
int i0=0,i=0, tDuration=1000;
void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available())
  {
    //val=Serial.read();
    numberStr=Serial.readString();// read the incoming data as string
  }
  if (numberStr.toInt() == 1)
  {
    digitalWrite(LEDpin,HIGH);
    Serial.print("LED: ON duration (");
    Serial.print(tDuration);
    Serial.print("ms),\t");
    delay(tDuration);
    digitalWrite(LEDpin,LOW);
    Serial.print("LED: OFF");
    Serial.println("-----------------------");
    //numberStr ="0";
  }
   else if (numberStr.toInt() == 0)
    {
      digitalWrite(LEDpin,LOW);
      Serial.println("LED: OFF");
//      Serial.print("Delay time: ");
//      Serial.print(tDuration);
//      Serial.println("ms");
      //Serial.println("-----------------------");
    }
    else if (numberStr.toInt() >= 500 && numberStr.toInt() <= 5000)
    {
      tDuration=numberStr.toInt();
      Serial.print("Set delay time: ");
      Serial.print(tDuration);
      Serial.println("ms");
      numberStr ="1";
    }
    delay(tDuration);
}
```


COM3
傳送
Type 1 to Enable LED every one-sec
Type 0 to Turn OFF LED
Type 500~5000 to set duration for LED ON
LED: OFF
LED: OFF
LED: OFF
LED: OFF
LED: OFF
LED: OFF
LED: OFF

COM3
1
傳送
Hello ^:^
Type 1 to Enable LED every one-sec
Type 0 to Turn OFF LED
Type 500~5000 to set duration for LED ON
LED: OFF
LED: OFF
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms),


COM3
0
Hello ^:^
Type 1 to Enable LED every one-sec
Type 0 to Turn OFF LED
Type 500~5000 to set duration for LED ON
LED: OFF
LED: OFF
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: ON duration (1000ms), LED: OFF----------
LED: OFF
LED: OFF
LED: OFF
LED: OFF

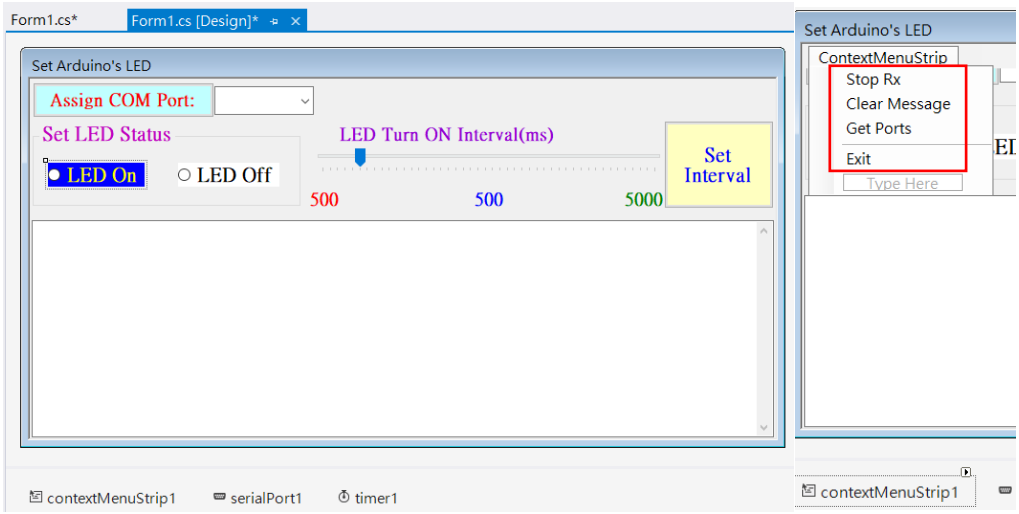

COM3
1500
傳送
Hello ^:^
Type 1 to Enable LED every one-sec
Type 0 to Turn OFF LED
Type 500~5000 to set duration for LED ON
LED: OFF
LED: OFF
LED: OFF
Set delay time: 1500ms
LED: ON duration (1500ms), LED: OFF----------

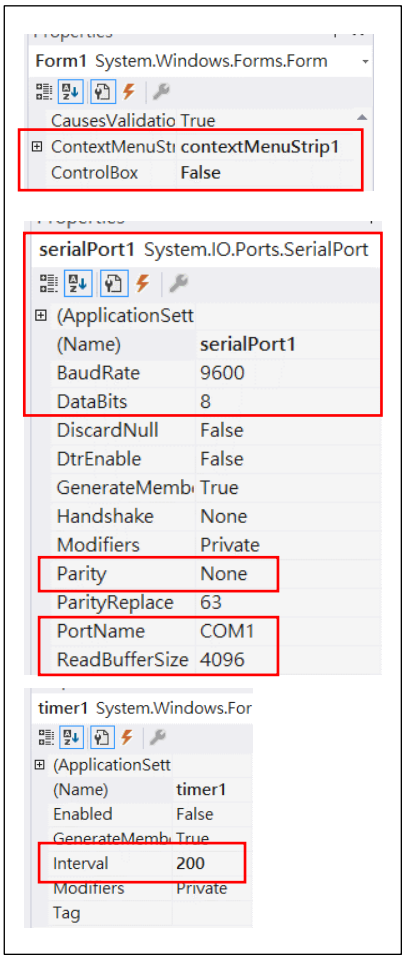

COM3
1
傳送
Hello ^:^
Type 1 to Enable LED every one-sec
Type 0 to Turn OFF LED
Type 500~5000 to set duration for LED ON
LED: OFF
LED: OFF
LED: OFF
Set delay time: 1500ms
LED: ON duration (1500ms), LED: OFF----------
LED: ON duration (1500ms), LED: OFF----------
LED: ON duration (1500ms), LED: OFF----------
LED: ON duration (1500ms), LED: OFF----------
LED: ON duration (1500ms), LED: OFF----------
LED: ON duration (1500ms), LED: OFF----------
LED: ON duration (1500ms), LED: OFF----------
LED: ON duration (1500ms), LED: OFF----------

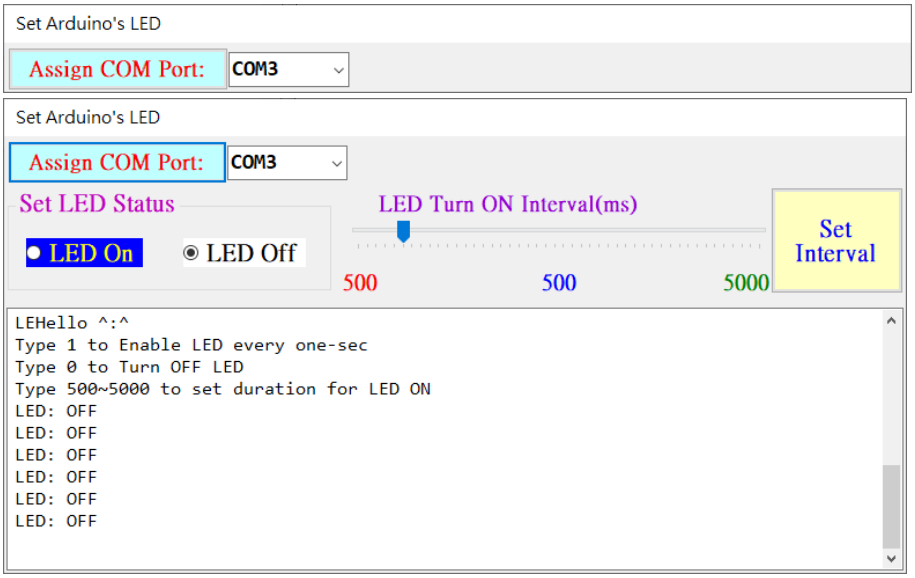
PC 端

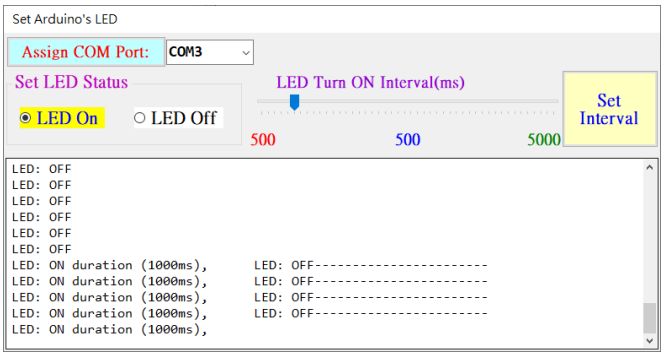
Ex.3-1PC: Set Arduino LED

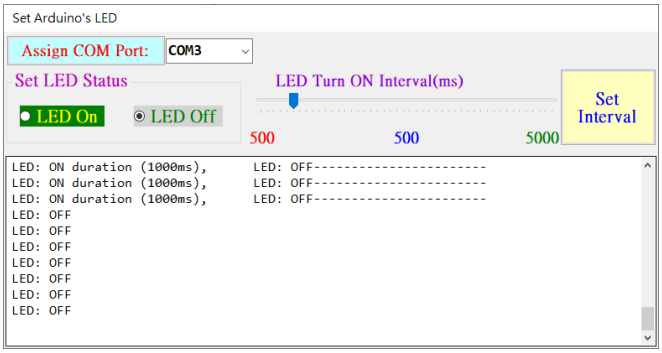
設計階段:

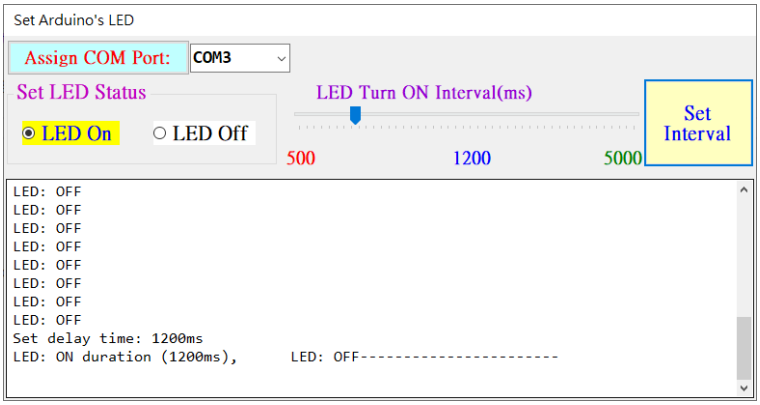


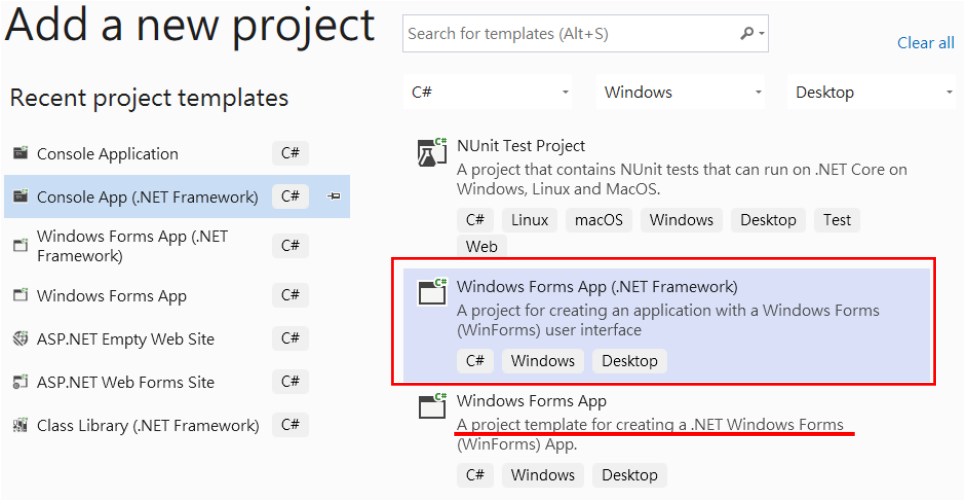













```
9   using System.Windows.Forms;
10  using System.IO.Ports;
11  namespace SetArduinoLED
12  {
13      public partial class Form1 : Form
14      {
15          public Form1()...
19          bool bLED = false;
20          int interval, iNow, len,i;
21          List<byte> raw;
22          byte[] buf;
23          StringBuilder sb;
24          private void getAllPorts()...
33          private void Form1_Load(object sender, EventArgs e)...
40          private void btnAssign_Click(object sender, EventArgs e)...
49          private void getPortsToolStripMenuItem_Click(object sender, EventArgs e)...
53          private void rdBtnOn_Click(object sender, EventArgs e)...
65          private void rdBtnOFF_Click(object sender, EventArgs e)...
77          private void btnSetIntrval_Click(object sender, EventArgs e)...
87          private void exitToolStripMenuItem_Click(object sender, EventArgs e)...
93          private void clearMessageToolStripMenuItem_Click_1(object sender, EventArgs e)...
97          private void trackBar1_Scroll(object sender, EventArgs e)...
102         private void timer1_Tick(object sender, EventArgs e)...
117         private void stopRxToolStripMenuItem_Click(object sender, EventArgs e)...
124         private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)...
134     }
135 }
```

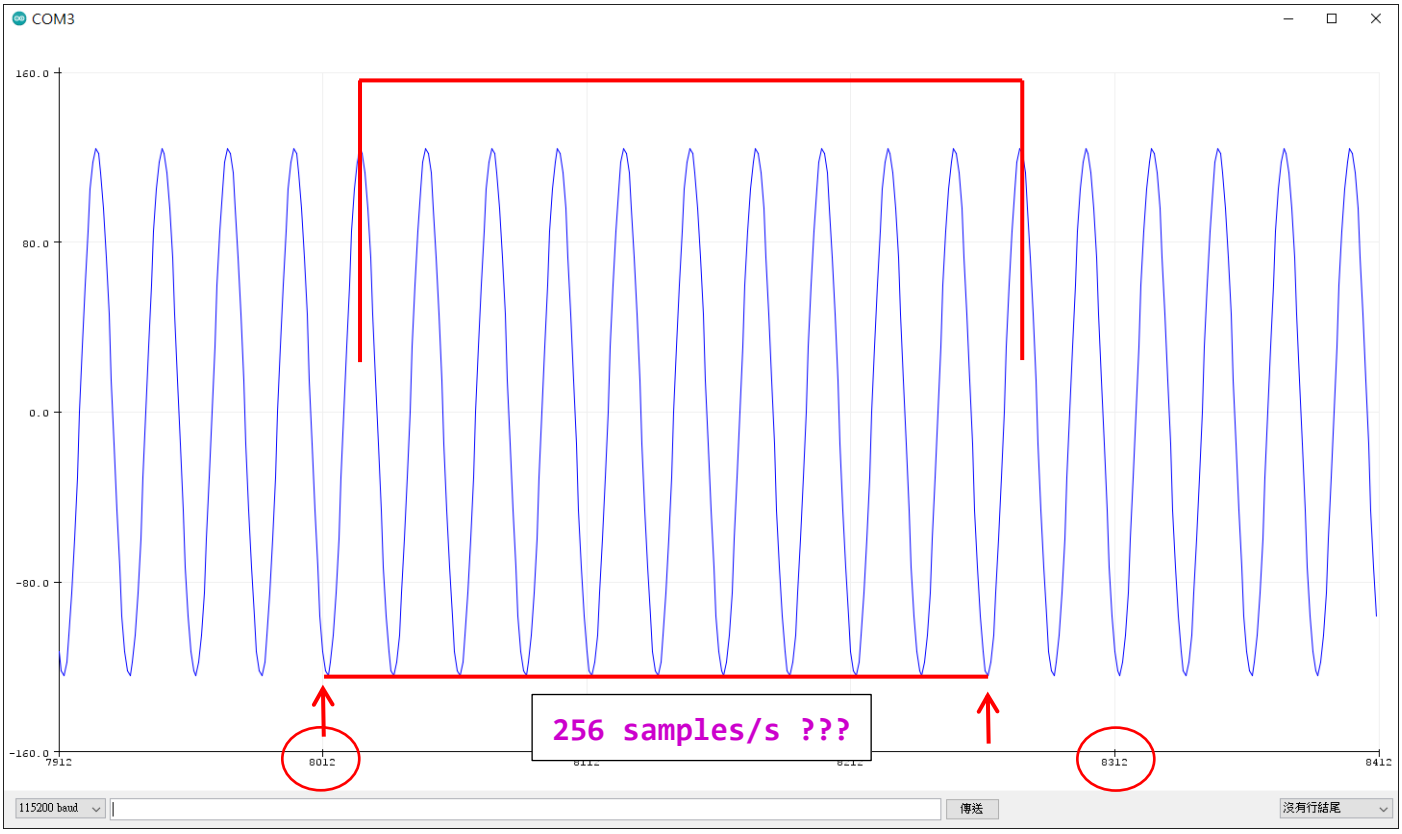
```
24          private void getAllPorts()
25          {
26              cmbBxCOMPort.Items.Clear();
27              string[] ports = SerialPort.GetPortNames();
28              Array.Sort(ports);
29              foreach (string port in ports)
30                  cmbBxCOMPort.Items.Add(port);
31              cmbBxCOMPort.SelectedIndex = cmbBxCOMPort.Items.Count - 1;
32          }
33          private void Form1_Load(object sender, EventArgs e)
34          {
35              getAllPorts();
36              Size = new Size(800, 85);
37              raw = new List<byte>();
38              sb = new StringBuilder();
39          }
```

```
40      private void btnAssign_Click(object sender, EventArgs e)
41      {
42          serialPort1.PortName = (string)cmbBxCOMPort.SelectedItem;
43          Size = new Size(800, 420);
44          iNow = 0;
45          buf = new byte[serialPort1.ReadBufferSize];
46          serialPort1.Open();
47          timer1.Start();
48      }

49      private void getPortsToolStripMenuItem_Click(object sender, EventArgs e)
50      {
51          getAllPorts();
52      }
        1 reference
53      private void rdBtnOn_Click(object sender, EventArgs e)
54      {
55          if (!bLED)
56          {
57              bLED = true;
58              serialPort1.Write("1");
59              rdBtnOn.ForeColor = Color.Blue;
60              rdBtnOn.BackColor = Color.Yellow;
61              rdBtnOFF.ForeColor = Color.Black;
62              rdBtnOFF.BackColor = Color.White;
63          }
64      }
        1 reference
65      private void rdBtnOFF_Click(object sender, EventArgs e)
66      {
67          if (bLED)
68          {
69              bLED = false;
70              serialPort1.Write("0");
71              rdBtnOn.ForeColor = Color.Yellow;
72              rdBtnOn.BackColor = Color.Green;
73              rdBtnOFF.ForeColor = Color.Green;
74              rdBtnOFF.BackColor = Color.LightGray;
75          }
76      }

77      private void btnSetIntrval_Click(object sender, EventArgs e)
78      {
79          serialPort1.Write(interval.ToString());
80          bLED = true;
81          rdBtnOn.Checked = true;
82          rdBtnOn.ForeColor = Color.Blue;
83          rdBtnOn.BackColor = Color.Yellow;
84          rdBtnOFF.ForeColor = Color.Black;
85          rdBtnOFF.BackColor = Color.White;
86      }
        1 reference
87      private void exitToolStripMenuItem_Click(object sender, EventArgs e)
88      {
89          if (serialPort1.IsOpen)
90              serialPort1.Close();
91          Close();
92      }
        1 reference
93      private void clearMessageToolStripMenuItem_Click_1(object sender, EventArgs e)
94      {
95          txtBxRxInfo.Text = "";
96      }
        1 reference
97      private void trackBar1_Scroll(object sender, EventArgs e)
98      {
99          interval = (trackBar1.Value/trackBar1.TickFrequency)* trackBar1.TickFrequency;
100         lblInterval.Text = interval.ToString();
101     }
```

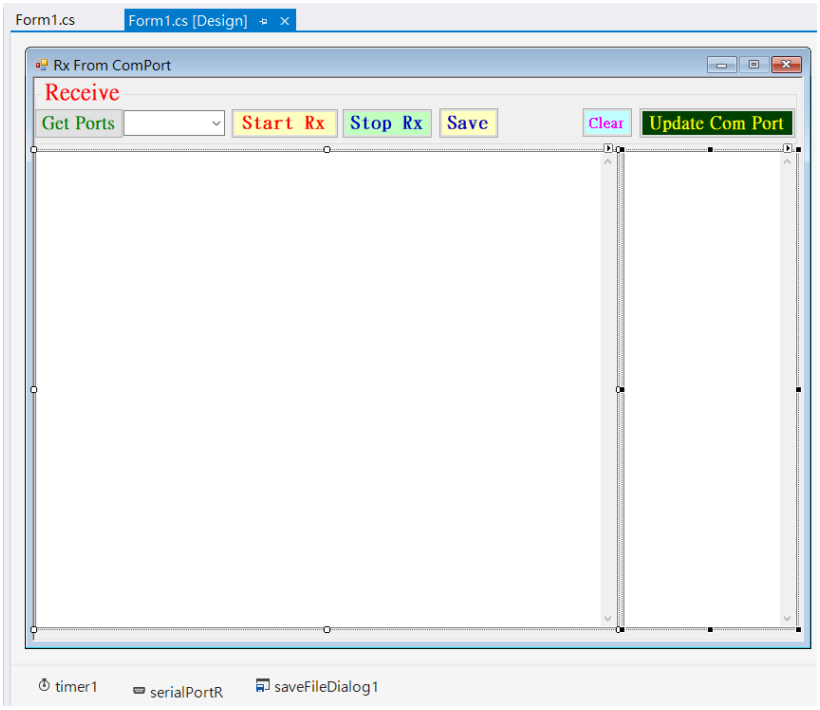
```
102         private void timer1_Tick(object sender, EventArgs e)
103         {
104             if (raw.Count > 0 && iNow < raw.Count)
105             {
106                 //while (iNow < raw.Count)
107                 //    sb.Append((char)raw[iNow++]);
108                 //txtBxRxInfo.Text = sb.ToString();
109                 //txtBxRxInfo.ScrollToCaret();
110                 //txtBxRxInfo.Refresh();
111                 sb.Clear();
112                 while (iNow < raw.Count)
113                     sb.Append((char)raw[iNow++]);
114                 txtBxRxInfo.AppendText(sb.ToString());
115             }
116         }
            1 reference
117         private void stopRxToolStripMenuItem_Click(object sender, EventArgs e)
118         {
119             if (serialPort1.IsOpen)
120                 serialPort1.Close();
121             sb.Clear();
122             timer1.Stop();
123         }
            1 reference
124         private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
125         {
126             if (serialPort1.BytesToRead > 0)
127             {
128                 len = serialPort1.Read(buf, 0, buf.Length);
129                 i = 0;
130                 while (i < len)
131                     raw.Add(buf[i++]);
132             }
133         }
134     }
135 }
```

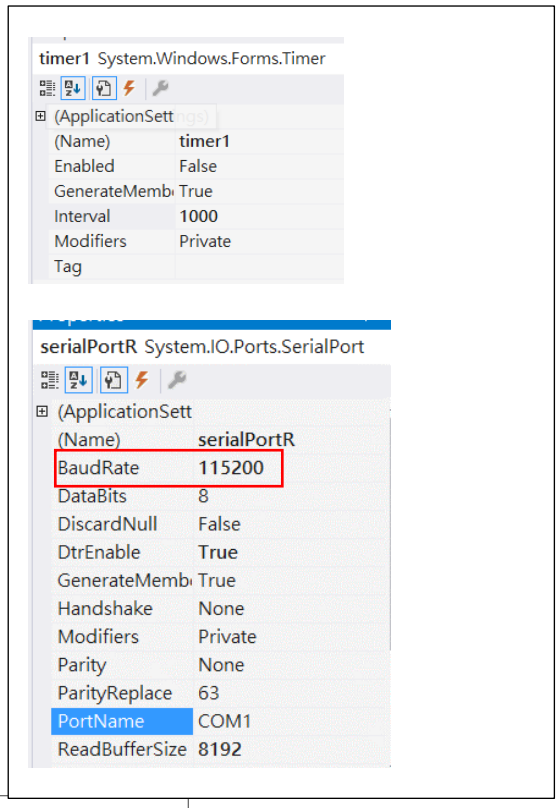
Ex03_02 Generating Sin Wave (Arduino, 0D0A)

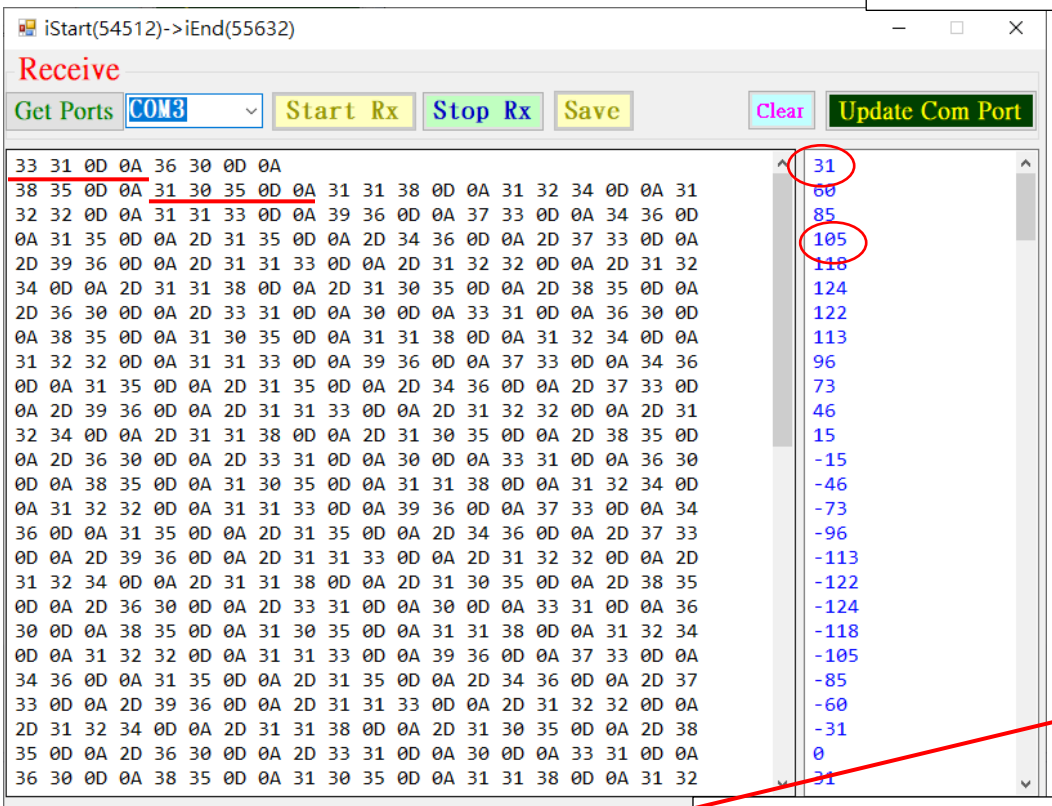
```
const int sampling=250, freq=10;
int i, data;
float dt;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  i=-1;
  dt=1000.0/sampling;
}
void loop() {
  // put your main code here, to run repeatedly:
  i=(i+1)%sampling;
//  data=i;//sawtooth Wave
  data=(int)(125*sin(2*3.14159*freq*i/sampling));
  Serial.println(data);
  delay(dt);
}
```

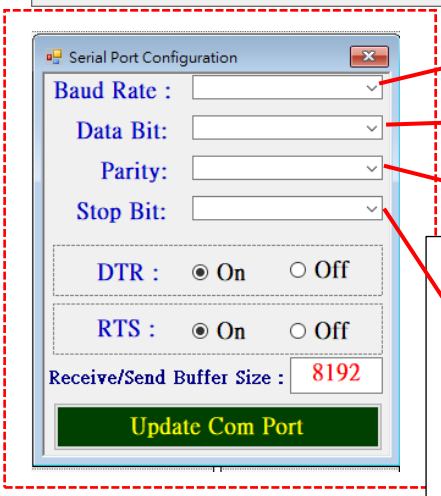


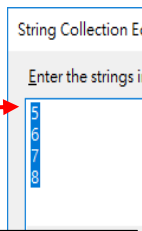
Ex3_3_PC (Rx Char end with 0x0D&0xA)

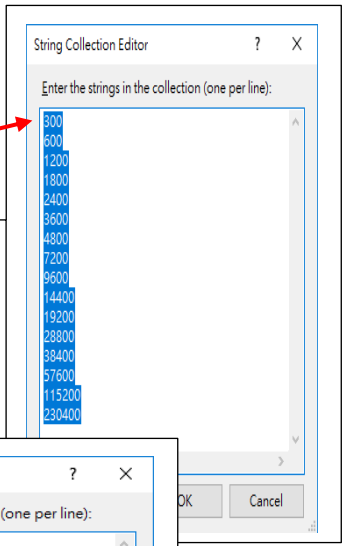


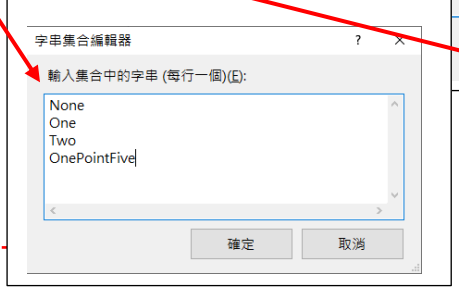


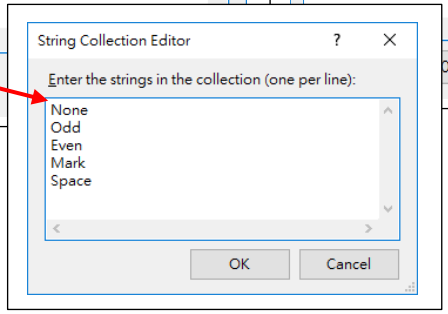


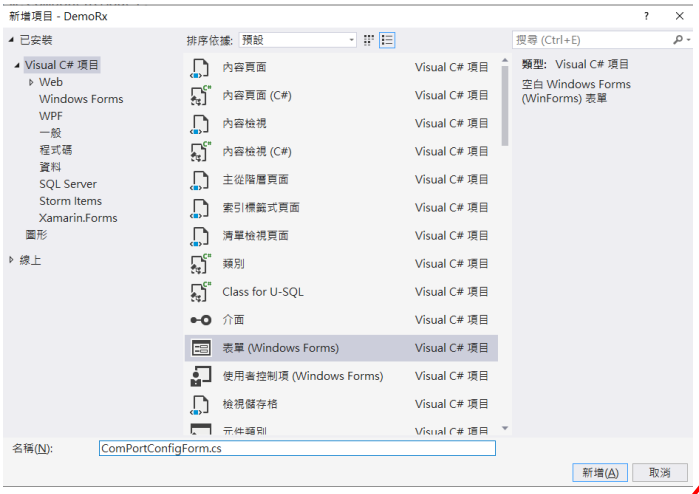


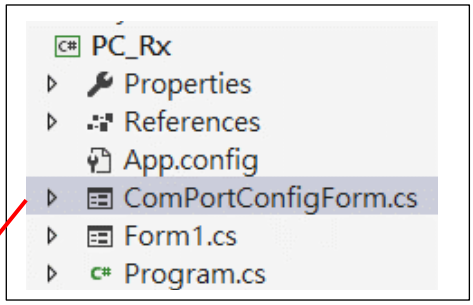













```
9   using System.Windows.Forms;
10  using System.IO.Ports;
11  namespace mySerialPort //Step.1
12  {
        5 個參考
13      public partial class ComPortConfigForm : Form
14      {
15          SerialPort serialPort1;//Step.2
            1 個參考
16          public ComPortConfigForm()
17          {
18              InitializeComponent();
19          }
            0 個參考
20          public ComPortConfigForm(ref SerialPort sp)
21          {//Step 4 customeize constructor
22              InitializeComponent();
23              ComPortConfig(ref sp);
24          }
                void ComPortConfigForm.ComPortConfig(ref SerialPort sp)

25          public void ComPortConfig(ref SerialPort sp)
26          {//Step 3
27              serialPort1 = sp;
28              string str;
29              int i;
30              for (i = 0; i < comboBox1.Items.Count; i++)
31              {
32                  str = comboBox1.Items[i].ToString();
33                  if (sp.BaudRate == int.Parse(str))
34                  {
35                      comboBox1.SelectedIndex = i;
36                      break;
37                  }
38              }
39              for (i = 0; i < comboBox2.Items.Count; i++)
40              {
41                  str = comboBox2.Items[i].ToString();
42                  if (sp.DataBits == int.Parse(str))
43                  {
44                      comboBox2.SelectedIndex = i;
45                      break;
46                  }
47              }
48              for (i = 0; i < comboBox3.Items.Count; i++)
49              {
50                  str = comboBox3.Items[i].ToString();
51                  if (sp.Parity.ToString().Contains(str))
52                  {//(string.Compare(sp.Parity.ToString(), str) == 1):bug, 11/26/2016
53                      comboBox3.SelectedIndex = i;
54                      break;
55                  }
56              }
```

```
            for (i = 0; i < comboBox4.Items.Count; i++)
            {
                str = comboBox4.Items[i].ToString();
                if (sp.StopBits.ToString().Contains(str))
                {
                    comboBox4.SelectedIndex = i;
                    break;
                }
            }
            if (sp.DtrEnable)
                DTROn.Checked = true;
            else
                DTROff.Checked = true;
            if (sp.RtsEnable)
                RTSOn.Checked = true;
            else
                RTSOff.Checked = true;
            txtBx_ReadBufferSize.Text = sp.ReadBufferSize.ToString();
        }
        1 個參考
        private void ComPortConfigForm_Load(object sender, EventArgs e)
        {
            if (serialPort1 == null)
            {
                MessageBox.Show("Bug in app!!");
            }
        }

        private void btnUpdateCom_Click(object sender, EventArgs e)
        {
            serialPort1.Close();
            serialPort1.BaudRate = int.Parse(comboBox1.SelectedItem.ToString());
            serialPort1.DataBits = int.Parse(comboBox2.SelectedItem.ToString());
            serialPort1.Parity = (Parity)Enum.Parse(typeof(Parity), comboBox3.SelectedItem.ToString(), true);
            serialPort1.StopBits = (StopBits)Enum.Parse(typeof(StopBits), comboBox4.SelectedItem.ToString(), true);
            serialPort1.ReadBufferSize = int.Parse(txtBx_ReadBufferSize.Text);
            serialPort1.WriteBufferSize = int.Parse(txtBx_ReadBufferSize.Text);
            serialPort1.DtrEnable = bool.Parse(DTROn.Checked.ToString());
            serialPort1.RtsEnable = bool.Parse(RTSOn.Checked.ToString());
            Close();
        }
    }
}
```

主程式

```
9   using System.Windows.Forms;
10  using System.IO.Ports;
11  using System.IO;
12  using mySerialPort;
13  namespace PC_Rx
14  {
        3 references
15      public partial class Form1 : Form
16      {
17          StringBuilder res, resH;
18          int iStart, iEnd, len, ii;
19          byte[] buf;
20          List<byte> raw;
21          int i;
22          ComPortConfigForm setupComPort;
23          byte val;
24          string s0;
            2 references
25          private void getAllPorts()...
            1 reference
36          private void displayRx()...
            1 reference
55          public Form1()...
            1 reference
60          private void Form1_Load(object sender, EventArgs e)...
            1 reference
70          private void btnGetPorts_Click(object sender, EventArgs e)...
            1 reference
74          private void btnStart_Click(object sender, EventArgs e)...
            1 reference
89          private void timer1_Tick(object sender, EventArgs e)...
            1 reference
94          private void serialPortR_DataReceived(object sender, SerialDataReceivedEventArgs e)...
            1 reference
104         private void btnStop_Click(object sender, EventArgs e)...
            1 reference
113         private void btnSave_Click(object sender, EventArgs e)...
            1 reference
126         private void btnClear_Click(object sender, EventArgs e)...
            1 reference
132         private void btnUpdateComR_Click(object sender, EventArgs e)...
139     }
140 }
```

```
            2 references
25          private void getAllPorts()
26          {
27              cmboBxPortR.Items.Clear();
28              string[] ports = SerialPort.GetPortNames();
29              Array.Sort(ports);
30              foreach (string port in ports)
31                  cmboBxPortR.Items.Add(port);
32              cmboBxPortR.SelectedIndex = cmboBxPortR.Items.Count - 1;
33              btnStart.Enabled = true;
34              serialPortR.PortName = cmboBxPortR.SelectedItem.ToString();
35          }
            1 reference
36          private void displayRx()
37          {
38              res.Clear();
39              resH.Clear();
40              iEnd = raw.Count - 1;
41              Text = string.Format("iStart({0})->iEnd({1})", iStart, iEnd);
42              while (iStart <= iEnd)
43              {
44                  val = raw[iStart++];
45                  s0 = string.Format("{0:X2} ", val);
46                  resH.Append(s0);
47                  if (iStart % 20 == 0)
48                      resH.AppendLine();
49                  res.AppendFormat("{0}", (char)val);
50              }
51              textBoxR.Text = resH.ToString();
52              textBoxChar.Text = res.ToString();
53              Application.DoEvents();
54          }
```

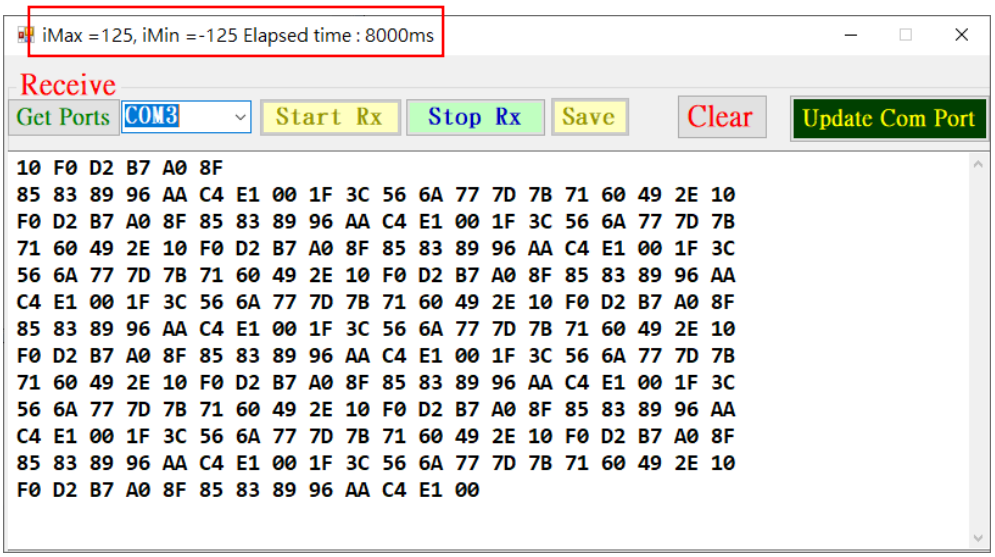
```
55          public Form1()
56          {
57              InitializeComponent();
58              setupComPort = new ComPortConfigForm();
59          }
            1 reference
60          private void Form1_Load(object sender, EventArgs e)
61          {
62              res = new StringBuilder();
63              resH = new StringBuilder();
64              getAllPorts();
65              //setupComPort = new ComPortConfigForm(ref serialPortR);
66              //setupComPort.ShowDialog();
67              raw = new List<byte>();
68              buf = new byte[serialPortR.ReadBufferSize];
69          }
            1 reference
70          private void btnGetPorts_Click(object sender, EventArgs e)
71          {
72              getAllPorts();
73          }
            1 reference
74          private void btnStart_Click(object sender, EventArgs e)
75          {
76              iStart = 0;
77              iEnd = -1;
78              ii = 0;
79              btnStop.Enabled = true;
80              btnSave.Enabled = false;
81              raw.Clear();
82              if (serialPortR.IsOpen)
83                  serialPortR.Close();
84              serialPortR.PortName = cmboBxPortR.SelectedItem.ToString();
85              serialPortR.Open();
86              btnStart.Enabled = false;
87              timer1.Start();
88          }

            1 reference
89          private void timer1_Tick(object sender, EventArgs e)
90          {
91              if (btnStop.Enabled)
92                  displayRx();
93          }
            1 reference
94          private void serialPortR_DataReceived(object sender, SerialDataReceivedEventArgs e)
95          {
96              if (!btnStart.Enabled && serialPortR.BytesToRead > 0)
97              {
98                  len = serialPortR.Read(buf, 0, buf.Length);
99                  i = 0;
100                 while (i < len)
101                     raw.Add(buf[i++]);
102             }
103         }
```

```
1 reference
104         private void btnStop_Click(object sender, EventArgs e)
105         {
106             btnStart.Enabled = true;
107             btnStop.Enabled = false;
108             if (raw.Count > 0)
109                 btnSave.Enabled = true;
110             serialPortR.Close();
111             timer1.Stop();
112         }
113         private void btnSave_Click(object sender, EventArgs e)
114         {
115             serialPortR.Close();
116             saveFileDialog1.FileName = string.Format("Arduino_{0:D4}{1:D2}{2:D2}_{3:D2}{4:D2}{5:D2}.txt",
117                         DateTime.Now.Year, DateTime.Now.Month, DateTime.Now.Day,
118                         DateTime.Now.Hour, DateTime.Now.Minute, DateTime.Now.Second);
119             if (saveFileDialog1.ShowDialog() != System.Windows.Forms.DialogResult.OK)
120                 return;
121             StringBuilder sb = new StringBuilder();
122             for (int i = 0; i < raw.Count; i++)
123                 sb.Append((char)raw[i]);
124             File.AppendAllText(saveFileDialog1.FileName, sb.ToString());
125         }
126         private void btnClear_Click(object sender, EventArgs e)
127         {
128             textBoxR.Text = "";
129             textBoxChar.Text = "";
130             raw.Clear();
131         }
132         private void btnUpdateComR_Click(object sender, EventArgs e)
133         {
134             setupComPort.ComPortConfig(ref serialPortR);
135             setupComPort.ShowDialog();
136             buf = null;
137             buf = new byte[serialPortR.ReadBufferSize];
138         }
139     }
140 }
```

Ex3_4 Generating Sin Wave (byte)

```
const int sampling=250, freq=10;
int i;
byte data;//unsigned
float y;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  i=-1;
  dt=1000.0/sampling;
}
void loop() {
  // put your main code here, to run repeatedly:
  i=(i+1)%sampling;
  y=125*sin(2*3.14159*freq*i/sampling);
  if (y >= 0)
    data=(byte)(y+0.5);
  else
    data=(byte)(y-0.5);
//  Serial.println(data);
  Serial.write(data);
  delay(dt);
}
```

Ex3_4PC2 (Rx byte)

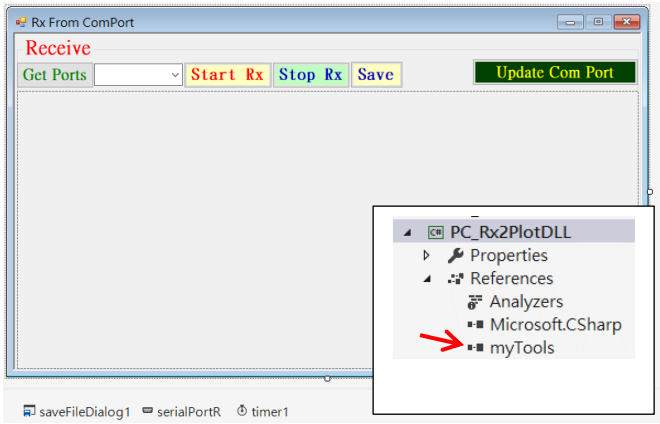


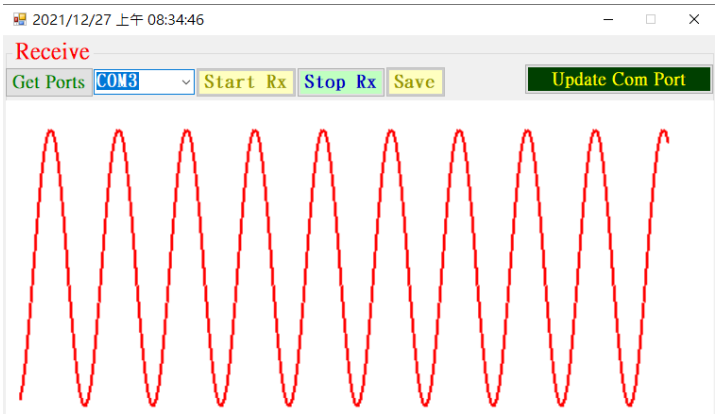

```
9   using System.Windows.Forms;
10  using System.IO.Ports;
11  using System.IO;
12  using mySerialPort;
13  namespace PC_Rx2
14  {
        3 references
15      public partial class RxForm1 : Form
16      {
17          StringBuilder resH;
18          int iStart, iEnd, len, imax, imin,t0;
19          byte[] buf;
20          List<sbyte> raw;
21          PortConfigForm setupComPort;
22          string s0;
23          sbyte val;
            2 references
24          private void getAllPorts()...
            1 reference
35          public RxForm1()...
            1 reference
40          private void serialPortR_DataReceived(object sender, SerialDataReceivedEventArgs e)...
            1 reference
50          private void displayRx()...
            1 reference
69          private void timer1_Tick(object sender, EventArgs e)...
            0 references
78          private void btnUpdateComR_Click(object sender, EventArgs e)...
            1 reference
84          private void RxForm1_Load(object sender, EventArgs e)...
            1 reference
93          private void btnGetPorts_Click(object sender, EventArgs e)...
            1 reference
97          private void btnStart_Click(object sender, EventArgs e)...
            1 reference
114         private void btnStop_Click(object sender, EventArgs e)...
            1 reference
122         private void btnSave_Click(object sender, EventArgs e)...
            1 reference
135         private void btnClear_Click(object sender, EventArgs e)...
140     }
141 }
```

```
24          private void getAllPorts()
25          {
26              cmboBxPortR.Items.Clear();
27              string[] ports = SerialPort.GetPortNames();
28              Array.Sort(ports);
29              foreach (string port in ports)
30                  cmboBxPortR.Items.Add(port);
31              cmboBxPortR.SelectedIndex = cmboBxPortR.Items.Count - 1;
32              btnStart.Enabled = true;
33              serialPortR.PortName = cmboBxPortR.SelectedItem.ToString();
34          }
            1 reference
35          public RxForm1()
36          {
37              InitializeComponent();
38              setupComPort = new PortConfigForm();
39          }
            1 reference
40          private void serialPortR_DataReceived(object sender, SerialDataReceivedEventArgs e)
41          {
42              if (!btnStart.Enabled && serialPortR.BytesToRead > 0)
43              {
44                  len = serialPortR.Read(buf, 0, buf.Length);
45                  for (int i = 0; i < len; i++)
46                      raw.Add((sbyte)buf[i]);
47                  iEnd = raw.Count - 1;
48              }
49          }
            1 reference
50          private void displayRx()
51          {
52              resH.Clear();
53              while (iStart <= iEnd)
54              {
55                  val = raw[iStart++];
56                  if (val > imax)
57                      imax = val;
58                  else if (imin > val)
59                      imin = val;
60                  s0 = string.Format("{0:X2} ", val);
61                  resH.Append(s0);
62                  if (iStart % 20 == 0)
63                      resH.AppendLine();
64              }
65              textBoxR.Text = resH.ToString();
66              Text= string.Format("iMax ={0}, iMin ={1} ", imax, imin);
67              Application.DoEvents();
68          }
            1 reference
69          private void timer1_Tick(object sender, EventArgs e)
70          {
71              t0++;
72              if (btnStop.Enabled)
73              {
74                  displayRx();
75                  Text += string.Format("Elapsed time : {0}ms", t0 * timer1.Interval);
76              }
77          }
```

```
78          private void btnUpdateComR_Click(object sender, EventArgs e)
79          {
80              setupComPort.ComPortConfig(ref serialPortR);
81              setupComPort.ShowDialog();
82              buf = new byte[serialPortR.ReadBufferSize];
83          }
84          private void RxForm1_Load(object sender, EventArgs e)
85          {
86              resH = new StringBuilder();
87              getAllPorts();
88              //setupComPort = new PortConfigForm(ref serialPortR);
89              //setupComPort.ShowDialog();
90              raw = new List<sbyte>();
91              buf = new byte[serialPortR.ReadBufferSize];
92          }
93          private void btnGetPorts_Click(object sender, EventArgs e)
94          {
95              getAllPorts();
96          }

97          private void btnStart_Click(object sender, EventArgs e)
98          {
99              imax = -1000;
100             imin = 1000;
101             iStart = 0;
102             iEnd = -1;
103             btnStop.Enabled = true;
104             btnSave.Enabled = false;
105             raw.Clear();
106             if (serialPortR.IsOpen)
107                 serialPortR.Close();
108             serialPortR.PortName = cmboBxPortR.SelectedItem.ToString();
109             serialPortR.Open();
110             btnStart.Enabled = false;
111             t0 = 0;
112             timer1.Start();
113         }
114         private void btnStop_Click(object sender, EventArgs e)
115         {
116             btnStart.Enabled = true;
117             btnStop.Enabled = false;
118             btnSave.Enabled = true;
119             serialPortR.Close();
120             timer1.Stop();
121         }

122         private void btnSave_Click(object sender, EventArgs e)
123         {
124             serialPortR.Close();
125             saveFileDialog1.FileName = string.Format("Arduino_{0:D4}{1:D2}{2:D2}_{3:D2}{4:D2}{5:D2}.txt",
126                         DateTime.Now.Year, DateTime.Now.Month, DateTime.Now.Day,
127                         DateTime.Now.Hour, DateTime.Now.Minute, DateTime.Now.Second);
128             if (saveFileDialog1.ShowDialog() != System.Windows.Forms.DialogResult.OK)
129                 return;
130             StringBuilder sb = new StringBuilder();
131             for (int i = 0; i < raw.Count; i++)
132                 sb.AppendLine(raw[i].ToString());
133             File.AppendAllText(saveFileDialog1.FileName, sb.ToString());
134         }
135         private void btnClear_Click(object sender, EventArgs e)
136         {
137             textBoxR.Text = "";
138             raw.Clear();
139         }
140     }
141 }
```

Ex3_5 Plotting Wave






```
10  using System.IO.Ports;
11  using System.IO;
12  using mySerialPort;
13  using myTools;
14  namespace PC_Rx2PlotDLL
15  {
        3 references
16      public partial class RxForm1 : Form
17      {
18          delegate void dispg();
19          dispg DispG;
20          int iStart, iEnd, len;
21          byte[] buf;
22          List<sbyte> raw;
23          int i;
24          PortConfigForm setupComPort;
25          sbyte val;
26          myWaveBMP myWave;
27          Image img;
            1 reference
28          private void RxForm1_Load(object sender, EventArgs e)...
            1 reference
35          private void btnStart_Click(object sender, EventArgs e)...
            1 reference
57          private void serialPortR_DataReceived(object sender, SerialDataReceivedEventArgs e)...
            1 reference
67          private void displayG()...
            1 reference
83          private void timer1_Tick(object sender, EventArgs e)...
            2 references
88          private void getAllPorts()...
            1 reference
99          public RxForm1()...
            1 reference
103         private void btnGetPorts_Click(object sender, EventArgs e)...
            1 reference
107         private void btnStop_Click(object sender, EventArgs e)...
            1 reference
115         private void btnSave_Click(object sender, EventArgs e)...
            1 reference
128         private void btnUpdateComR_Click(object sender, EventArgs e)...
135     }
136 }
```

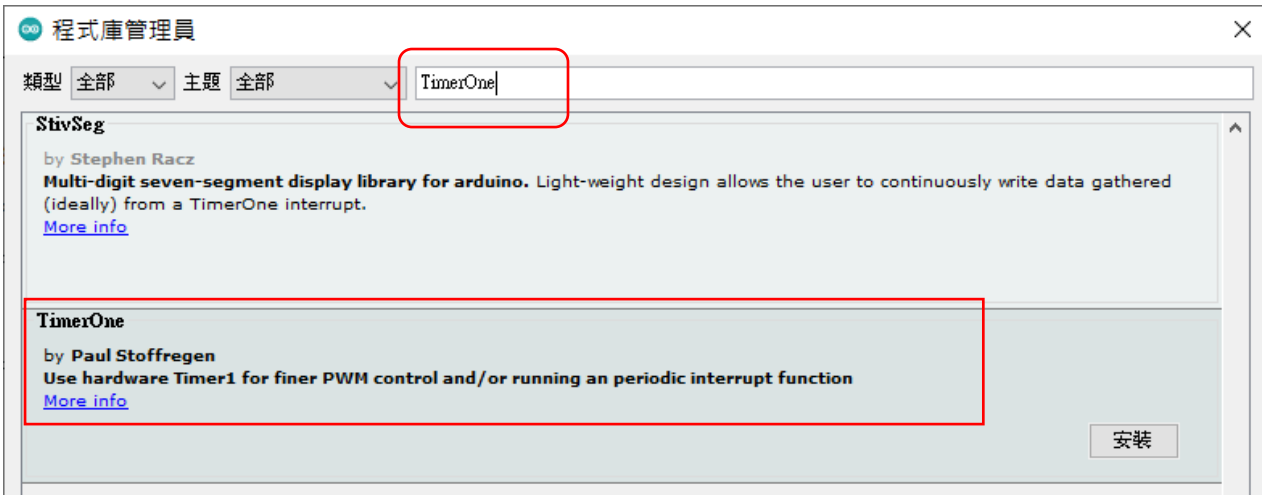
```
28          private void RxForm1_Load(object sender, EventArgs e)
29          {
30              getAllPorts();
31              raw = new List<sbyte>();
32              buf = new byte[serialPortR.ReadBufferSize];
33              DispG = new dispg(displayG);
34          }
```

```
35          private void btnStart_Click(object sender, EventArgs e)
36          {
37              iStart = 0;
38              iEnd = -1;
39              btnStop.Enabled = true;
40              btnSave.Enabled = false;
41              raw.Clear();
42              myWave = new myWaveBMP(250);//250: sampling rate. Display 1-sec signal
43              if (img != null)
44              {
45                  img.Dispose();
46                  img = null;
47              }
48              img = myWave.getBMP();
49              pictureBox1.Image = img;
50              if (serialPortR.IsOpen)
51                  serialPortR.Close();
52              serialPortR.PortName = cmboBxPortR.SelectedItem.ToString();
53              serialPortR.Open();
54              btnStart.Enabled = false;
55              timer1.Start();
56          }
57          private void serialPortR_DataReceived(object sender, SerialDataReceivedEventArgs e)
58          {
59              if (!btnStart.Enabled && serialPortR.BytesToRead > 0)
60              {
61                  len = serialPortR.Read(buf, 0, buf.Length);
62                  for (i = 0; i < len; i++)
63                      raw.Add((sbyte)buf[i]);
64                  BeginInvoke(DispG, new Object[] { });
65              }
66          }
            1 reference
67          private void displayG()
68          {
69              iEnd = raw.Count - 1;
70              while (iStart <= iEnd)
71              {
72                  val = raw[iStart++];
73                  myWave.update(val + 125);
74              }
75              if (img != null)
76              {
77                  img.Dispose();
78                  img = null;
79              }
80              img = myWave.getBMP();
81              pictureBox1.Image = img;
82          }
83          private void timer1_Tick(object sender, EventArgs e)
84          {
85              Text = DateTime.Now.ToString();
86              Application.DoEvents();
87          }
```

```
88          private void getAllPorts()
89          {
90              cmboBxPortR.Items.Clear();
91              string[] ports = SerialPort.GetPortNames();
92              Array.Sort(ports);
93              foreach (string port in ports)
94                  cmboBxPortR.Items.Add(port);
95              cmboBxPortR.SelectedIndex = cmboBxPortR.Items.Count - 1;
96              btnStart.Enabled = true;
97              serialPortR.PortName = cmboBxPortR.SelectedItem.ToString();
98          }
            1 reference
99          public RxForm1()
100         {
101             InitializeComponent();
102         }
            1 reference
103         private void btnGetPorts_Click(object sender, EventArgs e)
104         {
105             getAllPorts();
106         }
            1 reference
107         private void btnStop_Click(object sender, EventArgs e)
108         {
109             btnStart.Enabled = true;
110             btnStop.Enabled = false;
111             btnSave.Enabled = true;
112             serialPortR.Close();
113             timer1.Stop();
114         }
115         private void btnSave_Click(object sender, EventArgs e)
116         {
117             serialPortR.Close();
118             saveFileDialog1.FileName = string.Format("Arduino_{0:D4}{1:D2}{2:D2}_{3:D2}{4:D2}{5:D2}.txt",
119                         DateTime.Now.Year, DateTime.Now.Month, DateTime.Now.Day,
120                         DateTime.Now.Hour, DateTime.Now.Minute, DateTime.Now.Second);
121             if (saveFileDialog1.ShowDialog() != System.Windows.Forms.DialogResult.OK)
122                 return;
123             StringBuilder sb = new StringBuilder();
124             for (int i = 0; i < raw.Count; i++)
125                 sb.AppendLine(raw[i].ToString());
126             File.AppendAllText(saveFileDialog1.FileName, sb.ToString());
127         }
            1 reference
128         private void btnUpdateComR_Click(object sender, EventArgs e)
129         {
130             if (setupComPort != null)
131                 setupComPort = null;
132             setupComPort.ComPortConfig(ref serialPortR);
133             setupComPort.ShowDialog();
134         }
135     }
136 }
```

Ex3_6 Generating Sin Wave (byte, by Timer)

Step.1 Installing Timer Library




```
#include <TimerOne.h>
const int sampling=250, freq=10;
int i;
byte data;
float y;
void callback()
{
  y=125*sin(2*3.14159*freq*i++/sampling);
  if (y >= 0)
    data=(byte)(y+0.5);
  else
    data=(byte)(y-0.5);
  Serial.write(data);
}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  i=0;
// initialize timer1, and set a 4ms second period
  Timer1.initialize(4000); //0.00395*1000000 for comport
  Timer1.attachInterrupt(callback);
  //attaches callback() as a timer overflow interrupt
}
void loop() {
}
```