การเชื่อมต่อพอร์ตอนุกรมด้วยซีชาร์ป

ศุภชัย บุศราทิจ

คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเพชรบุรี

ซีชาร์ป (C#) เป็นภาษาเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming Language) ที่บริษัทไมโครซอฟต์พัฒนาขึ้นมาเพื่อใช้กับสถาปัตยกรรมด็อทเน็ต (.NET Architecture) และเป็นภาษาที่เกิดจากปรับปรุงมาจากภาษาจาวา (Java) และซีพลัสพลัส (C++) ทำให้เป็นภาษาที่รวมข้อดีของทั้งสองภาษาเข้าด้วยกัน พร้อมทั้งยังมีคำสั่งรองรับระบบรักษาความปลอดภัย (security) และเขียนโปรแกรมเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้อย่างง่ายดาย นอกจากนี้ยังซีชาร์ปยังเป็นภาษาที่มีความยืดหยุ่นในการทำงานบนระบบเครื่องเดี่ยว (stand alone) ผ่านระบบเว็บ (web base) และผ่านทางระบบระยะไกลอีกด้วย

ในบทความนี้ผู้เขียนมีจุดประสงค์ที่จะใช้ซีชาร์ปเป็นตัวแสดงผลที่รับมาจากพอร์ตอนุกรมแบบ RS232 ที่ทำการเชื่อมต่อกับบอร์ด ET-ARM7 BASE2103 โดยที่โปรแกรมในบอร์ดนั้นเป็นโปรแกรมที่อ่านนาฬิกาเวลาจริง (RTC) แล้วส่งผลลัพธ์ไปที่ UART0

ฮาร์ดแวร์และซอฟต์แวร์

1. Microsoft Visual C# 2005 Express (www.microsoft.com)
2. Keil-C ARM7 (www.keil.com)
3. Philips LPC21XX Flash-Utilities
4. ET-BASE ARM2103 (http://www.etteam.com/product/ARM/et-base_arm2103.htm)

รูปที่ 1 ET-BASE ARM2103

โปรแกรมสำหรับ ARM7

โปรแกรมสำหรับ ARM7 นั้น ผมเลือกใช้โปรแกรมตัวอย่างของทางอีทีที ที่เขียนโดย คุณเอกชัย มการ ซึ่งโปรแกรมจะทำการอ่านค่านาฬิกาเวลาจริงแล้วส่งผลลัพธ์ไปที่ UART0 ดังนี้

```
/****************************************************/
/* Examples Program For "ET-ARM7 BASE LPC2103" Board */
/* Target MCU  : Philips ARM7-LPC2103  X-TAL : 19.6608 MHz              */
/*             : Run Speed 58.9824MHz (With PLL)     */
/* Keil Editor : uVision3 V3.03a                     */
/* Compiler    : Keil CARM V2.50a                    */
/* Create By   : Eakachai Makarn (WWW.ETT.CO.TH)     */
/* Last Update : 17/April/2006                       */
/* Function    : Example Read RTC + Display on UART0 */
/****************************************************/
// Read/Write Internal RTC of LPC2103
// Display Result on UART0(9600,N,8,1)

#include <LPC2103.H>             // LPC2103 MPU Register
#include <stdio.h>               // นำเข้าเพื่อเรียกใช้คำสั่ง printf


#define MASKSEC 0x3F             // มาส์กของวินาที 00..59   00000000:00000000:00xxxxxx
#define MASKMIN 0x3F00           // มาส์กของนาที 00..59   00000000:00xxxxxx:00000000
#define MASKHR  0x1F0000         // มาส์กของชั่วโมง 00..23   000xxxxx:00000000:00000000

/* pototype  section */
void init_serial0 (void);        // กำหนดค่าเริ่มต้นของ UART-0
int putchar (int ch);            // ส่งอักขระไปที่  UART-0
int getchar (void);              // รับอักระจาก UART-0

int main(void)
{
  unsigned char Hour,Minute,Second,Last_Second;  // ตัวแปรเก็บข้อมูลนาฬิกาเวลาจริง
  init_serial0();                // กำหนดค่าเริ่มต้นการทำงานของ UART0 เป็น 9600,N,8,1
  printf("\n\n\nET-ARM7 STAMP LPC2103...TEST RTC\n");

  // กำหนดฟังก์ชันการทำงานของนาฬิกาเวลาจริง
  CCR &= 0x00; // ล้างค่าทุกบิตให้เป็น 0
  CCR |= 0x10;  // CLKSRC = 1 เพื่อใช้งาน Xtal ภายนอกที่มีความถี่ 32.768 KHz

  CCR |= 0x02;  // Reset Clock     (0000 0010)
  CCR &= 0xFD;// Release Reset  (1111 1101)

  CCR |= 0x01;  // กำหนดให้ RTC Clock เริ่มทำงาน
  CCR  = 0x11;  // เริ่มต้นการทำงานของ RTC Clock โดยใช้สัญญาณนาฬิกาจากภายนอก  32.768 KHz

  // กำหนดค่าเวลาเป็น 00:00:00
  HOUR = 0x00;
  MIN  = 0x00;
  SEC  = 0x00;
  Last_Second = 0x00;
```

```
  // เริ่มดำเนินการอ่านเวลาและส่งไปที่ UART-0
  while(1)   {
    do { // ทำการวนรอบจนกว่าค่าวินาทีจะเปลี่ยน
           Hour   = (CTIME0 & MASKHR)>>16;   // อ่านค่าชั่วโมง
           Minute = (CTIME0 & MASKMIN)>>8;   // อ่านค่านาที
           Second = CTIME0 & MASKSEC;        // อ่านค่าวินาที
    }   while(Last_Second == Second);

    Last_Second = Second;       // จำค่าวินาทีก่อนหน้านี้

    //************************************//
    // แสดงผลเป็นรูปแบบ ชั่วโมง:นาที:วินาที //
    //************************************//
    printf("\rReal Time Clock = ");
    printf(" %2d : %2d : %2d",Hour,Minute,Second);
  }
}

/*******************************/
/* Initial UART0 = 9600,N,8,1 */
/* VPB(pclk) = 29.4912 MHz    */
/*******************************/
void init_serial0 (void)
{
  PINSEL0 &= 0xFFFFFFF0;       // Reset P0.0,P0.1 Pin Config
  PINSEL0 |= 0x00000001;       // Select P0.0 = TxD(UART0)
  PINSEL0 |= 0x00000004;       // Select P0.1 = RxD(UART0)

  U0LCR &= 0xFC;               // Reset Word Select(1:0)
  U0LCR |= 0x03;               // Data Bit = 8 Bit
  U0LCR &= 0xFB;               // Stop Bit = 1 Bit
  U0LCR &= 0xF7;               // Parity = Disable
  U0LCR &= 0xBF;               // Disable Break Control
  U0LCR |= 0x80;               // Enable Programming of Divisor Latches

  // U0DLM:U0DLL = 29.4912MHz / [16 x Baud]
  //             = 29.4912MHz / [16 x 9600]
  //             = 192 = 0x00C0
  U0DLM = 0x00;                // Program Divisor Latch(192) for 9600 Baud
  U0DLL = 0xC0;

  U0LCR &= 0x7F;               // Disable Programming of Divisor Latches
  U0FCR |= 0x01;               // FIF0 Enable
  U0FCR |= 0x02;               // RX FIFO Reset
  U0FCR |= 0x04;               // TX FIFO Reset
  U0FCR &= 0x3F;
}

/*****************************/
/* Write Character To UART0 */
/*****************************/
int putchar (int ch)
{
  if (ch == '\n')  {
    while (!(U0LSR & 0x20));     // รอจนกว่า TXD จะว่าง
    U0THR = 0x0D;                //ส่งรหัส CR (ขึ้นบรรทัดใหม่)
  }
```

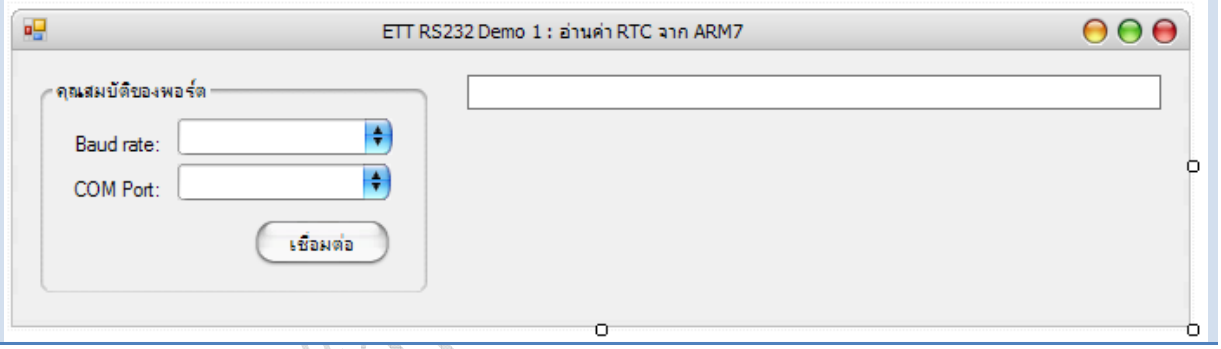
```
  while (!(U0LSR & 0x20));        // รอจนกว่า TXD จะว่าง
  return (U0THR = ch);            // ส่งอักขระที่เก็บในตัวแปร ch
}

/*****************************/
/* Read Character From UART0 */
/*****************************/
int getchar (void)
{
  while (!(U0LSR & 0x01));        // รอจนกว่าจะมีข้อมูลใน RXD
  return (U0RBR);                 // ส่งกลับค่าที่เก็บใน U0RBR
}
```

เขียนโปรแกรมฝั่งซีชาร์ป

ออกแบบหน้าจอ

รูปที่ 2 หน้าจอออกแบบ



คำสั่งเพื่อตอบสนองตอนโหลดฟอร์ม

```
        private void Form1_Load(object sender, EventArgs e)
        {
            // ให้ตัวแปร s เก็บรายการชื่อพอร์ตอนุกรม
            string[] s = SerialPort.GetPortNames();
            int i = 0;

            timer1.Stop();
            // คอมโบสำหรับเลือกความเร็วในการสื่อสาร
            comboBox1.Items.Clear();
            comboBox1.Items.Add("9600");
            comboBox1.Items.Add("19200");
            comboBox1.SelectedIndex = 0;
```

```
        // คอมโบสำหรับเลือกหมายเลขพอร์ต
        comboBox2.Items.Clear();
        foreach (string port in s) // เพิ่มรายชื่อพอร์ตใน comboBox2
        {
            comboBox2.Items.Add(s[i]);
            i++;
        }
        if (i > 0)
        {
            comboBox2.SelectedIndex = 0; // เลือกพอร์ตแรกที่พบเป็นค่าปกติ
        }
        else
        {
            comboBox2.Enabled = false; // ไม่ให้เลือก เนื่องจากไม่พบรายการพอร์ตอนุกรมในเครื่อง
            button1.Enabled = false; // ไม่ให้คลิกปุ่มเชื่อมต่อ
        }
    }
```

คำสั่งตอบสนองเมื่อสั่งเชื่อมต่อ

```
    private void button1_Click(object sender, EventArgs e)
    {
        // จะทำงานถ้ามีรายการพอร์ตติดตั้งในเครื่อง
        if (comboBox2.Items.Count > 0)
        {
            if (rs232.IsOpen)
            {
                rs232.Close(); // ปิดพอร์ต
                timer1.Stop();
                textBox1.Text = "";
                ss = "";
            }
            timer1.Start();
            rs232.BaudRate = int.Parse(comboBox1.Text);
            rs232.DataBits = 8;
            rs232.Parity = (Parity)Enum.Parse(typeof(Parity), "None");
            rs232.StopBits = (StopBits)Enum.Parse(typeof(StopBits), "One");
            rs232.PortName = comboBox2.Text;
            rs232.Open();
        }
    }
```

คำสั่งตอบสนองเมื่อมีข้อมูลผ่านมาทางพอร์ต

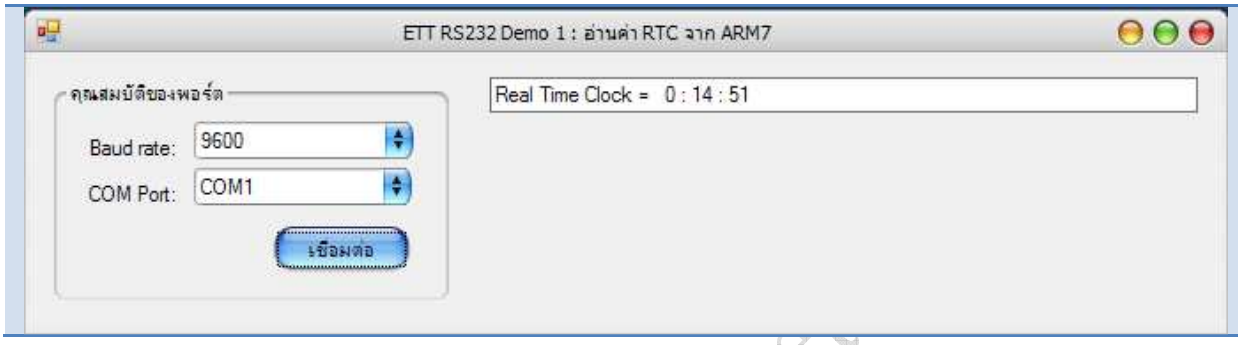
```
    public Form1()
    {
        InitializeComponent();

        // เพิ่มการดักเหตุการณ์รับข้อมูลจากพอร์ต RS232 โดยให้เรียกฟังก์ชัน rs232_DataRecieved
        rs232.DataReceived += new SerialDataReceivedEventHandler(rs232_DataReceived);
    }
```

```
private void rs232_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
   ss += rs232.ReadExisting();
}
```

ผลการทำงานของโปรแกรม

รูปที่ 3 หน้าจอการทำงานของโปรแกรม



สรุป

จากตัวอย่างในบทความนี้คงพอจะเป็นแนวทางสำหรับผู้สนใจได้ศึกษาถึงวิธีการเชื่อมต่อกับพอร์ตอนุกรมแบบ RS232 ด้วยซีชาร์ป ซึ่งรายละเอียดของการใช้งานพอร์ตอนุกรมนั้นสามารถดูได้จาก Help ของ MSDN ที่มากับวิชวลสตูดิโอ 2005 เพราะได้ให้รายละเอียดและวิธีการสั่งงานเอาไว้อย่างมากเพียงพอต่อการทำความเข้าใจ

สุดท้ายนี้ขอขอบคุณครอบครัว ทีมงานอีทีที และคุณกอบกิจ เติมผาติ เป็นอย่างสูงที่ให้การสนับสนุนและเป็นกำลังใจแก่ผู้เขียนบทความเสมอมา ขอบคุณมากครับ