

DTM-319A Protocol of Serial Interface

Baudrate : 9600

Parity: None

Data bits : 8

Stop bits : 1

The command of Digital Output is list below:

RS232 command	Function	Remarks
A(ASC 41H)	Send encoded data	Return encoded 32 byte
C(ASC 43H)	°C°F button	
E(ASC 45H)	REC button	
H(ASC 48H)	HOLD button	
K(ASC 4BH)	Ask for model No.	Return 4 bytes
M(ASC 4DH)	MAX/MIN button	
N(ASC 4EH)	Exit MAX/MIN mode	
T(ASC 54H)	T1-T2 button	
W(ASC 57H)	Back light button	
U(ASC 55H)	Dump all memory	
P(ASC 50H)	Load recorded data	
e(ASC 65H)	Erase memory	
a(ASC 61H)	Send encoded data continusly	Return encoded 10 byte
c(ASC 63H)	Stop sending encoded data	
X(ASC 58H)	Response from PC when receive data	
(ASC 72H,74H,63H)	透過 PC 設定日期時間	
(ASC 77H,45H,AddE,AddH,AddL,Data)	透過 PC 直接對 EEPROM 寫入 data	

Note:you have to send 10 byte to meter, for example, if you want to send A comand, the format will be 0x02 0x41 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x03

- **Command K:**

Return 4 bytes. For example, when sends command "K" to meter, it will return "3","7","4", ASCII(13) .

- **Command M:**

Equivalent to one pushing on the MAX/MIN button and no message is returned.

- **Command N:**

Equivalent to one pushing and hold the MAX/MIN button for two seconds to exit MAX/MIN mode.

- **Command E:**

Equivalent to one pushing on the REC button and no message is returned.

• **Command U:**

Return 32768 bytes .

• **Command P:**

Instead of returning all memory, it only return recorded data .

• **Command A: (回傳 33 Byte)**

1st BYTE:

The first byte is the start byte , it value is **02**.

2nd BYTE:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Low Bat	Hold	T1-T2	Recording	C/F	Avg	MAX/MIN	

bit 1 bit 0

0 0 ->normal mode

0 1 ->MAXIMUM mode

1 0 ->MINIMUM mode

1 1 -> calculate MAX/MIN in background and lcd "MAX""MIN" will flash.

bit3: 1->C 0->F

bit4: 1-> now is recording , 0->not recording

bit5: 1 -> LCD now is displaying T1-T2 .

bit6: 1- HOLD 0 -> Not HOLD

bit7: 1->LOW BATTERY 0 ->BATTERY NORMAL

3rd BYTE:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Auto Off	MemFull	K	J	E	T	bat bit 1	Bat bit 0

Bit7:1->Auto power off enabled.

Bit6:1->Memory full

bit 1 bit 0 電池電力容量

0 0 -> 0 格

0 1 -> 1 格

1 0 -> 2 格

1 1 -> 3 格

For example: 8nd and 9nd byte are 0x01 0x02 then T1 will be 0x0102 that is 258 in decimal , then divided by 10 , that is 25.8 degree

4th BYTE and 5th BYTE: channel 1 讀值

6nd BYTE and 7nd BYTE: channel 2 讀值
8nd BYTE and 9nd BYTE: channel 3 讀值
10nd BYTE and 11nd BYTE: channel 4 讀值 ;
12nd BYTE and 13nd BYTE: T1-T2 讀值
14nd BYTE and 15nd BYTE: 當按下 MAX/MIN 鍵後的 channel 1 的最大值,
16nd BYTE and 17nd BYTE: 當按下 MAX/MIN 鍵後的 channel 1 的最小值,
18nd BYTE and 19nd BYTE: 當按下 MAX/MIN 鍵後的 channel 1 的平均值,
20nd BYTE and 21nd BYTE: 日期 (指 Max)
22nd BYTE and 23nd BYTE: 時間 (指 Max)
24nd BYTE and 25nd BYTE: 日期 (指 Min)
26nd BYTE and 27nd BYTE: 時間 (指 Min)
28nd BYTE and 29nd BYTE: 日期 (指 Avg)
30nd BYTE and 31nd BYTE: 時間 (指 Avg)

32nd BYTE:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T4 unplugged	T3 unplugged	T2 unplugged	T1 unplugged	T4 OL	T3 OL	T2 OL	T1 OL

33rd BYTE:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	T3 AVG unplugged	T2 MAX unplugged	T1 Min unplugged		T3 AVG OL	T2 MAX OL	T1 MIN OL

34th checksum BYTE (不包含 start byte 02 , end byte 03)

35th BYTE

The last byte is the end byte , it value is **03**, first and last byte are used to check frame error.

=====以下為範例程式碼=====

```

using System;
using System.IO.Ports;
using System.Threading;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        SerialPort Rs232sp;
    }
}

```

byte[] DtByte = {0x02, 0x41, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03 }; //指令碼 A ，要
求回傳數據資料。

```
string LogData = "";
```

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
Rs232sp.Write(DtByte, 0, DtByte.Length); //發送指令碼 A  
Thread.Sleep(200); //等待資料反饋 500mS
```

```
int LGo = Rs232sp.BytesToRead; //取得反饋資料長度的變數。  
if (LGo != 0) //當反饋資料不為空時，執行取得資料
```

```
{
```

```
byte[] buffer = new byte[LGo]; //接受位元資料變數  
Rs232sp.Read(buffer, 0, LGo); //提取 RS-232 緩衝區資料
```

```
LogData = BitConverter.ToString(buffer); //轉換資料為字串數據
```

```
}
```

```
label1.Text = LogData; //顯示到畫面上 label1 物件，顯示為取的資料位元  
cpre(LogData); //執行副程式，取得量測數值
```

```
}
```

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
```

```
{
```

```
Rs232sp = new SerialPort(comboBox1.Text, 9600, Parity.None, 8, StopBits.One);  
Rs232sp.Open(); //設定並開起 RS232 連接埠
```

```
}
```

```
private void comboBox1_Enter(object sender, EventArgs e)
```

```
{
```

```
comboBox1.Items.AddRange(System.IO.Ports.SerialPort.GetPortNames());
```

```
} //當點下拉選單時，獲得 RS232 連接埠所有可用的通道。
```

```
public void cpre(string data)
```

```
{
```

```
char[] lokd = data.ToCharArray(); //轉換字串數據為 字元陣列，為了好分析資料。
```

```
string TC = new string(lokd,9, 5); //提取陣列 啟始 9 之後 5 個位元碼。
```

```
label2.Text = Convert.ToInt32(TC.Replace("-", ""), 16).ToString("#.'0");
```

```
} //消除 "-" 符號並將 16 位元轉為 10 位元，然後補上小數點在最後一個數值之前！
```

```
}
```

```
}
```

