

数字称重传感器

型号 740D 1560t

用户说明书



目录

| | | |
|----------|----------------------------------|-----------|
| 1 | 数字称重传感器概述 | 2 |
| 2 | 串口 RS-485 (4 线制全双工) | 3 |
| 2.1 | 接口特性 | 3 |
| 2.2 | 电气连接 | 3 |
| 2.3 | 通信 | 4 |
| 3 | 命令 | 5 |
| 3.1 | 命令格式 | 5 |
| 3.2 | 对命令的应答。 | 5 |
| 3.3 | 测量值的输出。 | 6 |
| 3.4 | 命令概述: | 6 |
| 4 | 命令描述 | 7 |
| 4.1 | ADR.设置总线地址。 | 7 |
| 4.2 | FIL.设置滤波器 | 8 |
| 4.3 | VAL.读取重量 | 9 |
| 4.4 | ZER.校准用户零点..... | 10 |
| 4.5 | GAI.校准用户增益..... | 11 |
| 4.6 | NOM.设置标称定标..... | 12 |
| 4.7 | RES.设备复位..... | 13 |
| 4.8 | RDV.重新加载默认值..... | 14 |
| 4.9 | VER.软件版本..... | 15 |
| 4.10 | TRG.重量触发..... | 16 |
| 4.11 | STU.称重传感器状态 | 17 |
| 4.12 | BAU.更改波特率..... | 18 |
| 4.13 | CAP.称重传感器标称称重能力..... | 19 |
| 4.14 | CHK.发送的重量的校验和..... | 20 |
| 5 | 修订历史记录 | 21 |
| 6 | 附录 校验和算法 | 22 |

1 数字称重传感器概述

- 工作电压：8 VDC18 VDC
- 串口 RS-485（4 线制全双工）
- 可升级数字滤波器
- 参数存储在非易失性存储器中
- 内部分辨率：24 位， Σ - Δ AD 转换器， $\pm 3\text{mV/V}$
- 所有设置通过串行接口进行
- 软件可通过串行接口更新
- 雷电防护

2 串口 RS-485 (4 线制全双工)

通过 RS-485 接口可将多达 32 个数字称重传感器连接在一根公共总线上。总线模式设计为主从配置，其中数字称重传感器以从设备实现。因此，所有系统活动都通过主计算机或数字称重显示仪表发起。

每个称重传感器接收自己的通信地址 (01 - 32)。可以使用广播指令 (设备地址 00) 同时向所有从设备发送一些特殊指令。在这种情况下，从设备都不会发送收悉确认。

用 RS-485 总线可实现最长 1000 米的电缆长度。因为电磁兼容的原因，建议使用双绞屏蔽电缆。由于 RS-485 总线采用差分总线接口，信号水平是线间电压差的状态，与地无关。

总线末端需要终端电阻来实现总线系统的电气功能。

2.1 接口特性

| | |
|-------|-----------------|
| 起始位: | 1 |
| 数据长度: | 8 位 |
| 停止位: | 1 |
| 奇偶校验: | 无校验 |
| 握手: | 未实现 |
| 波特率: | 4800 - 38400 波特 |

2.2 电气连接

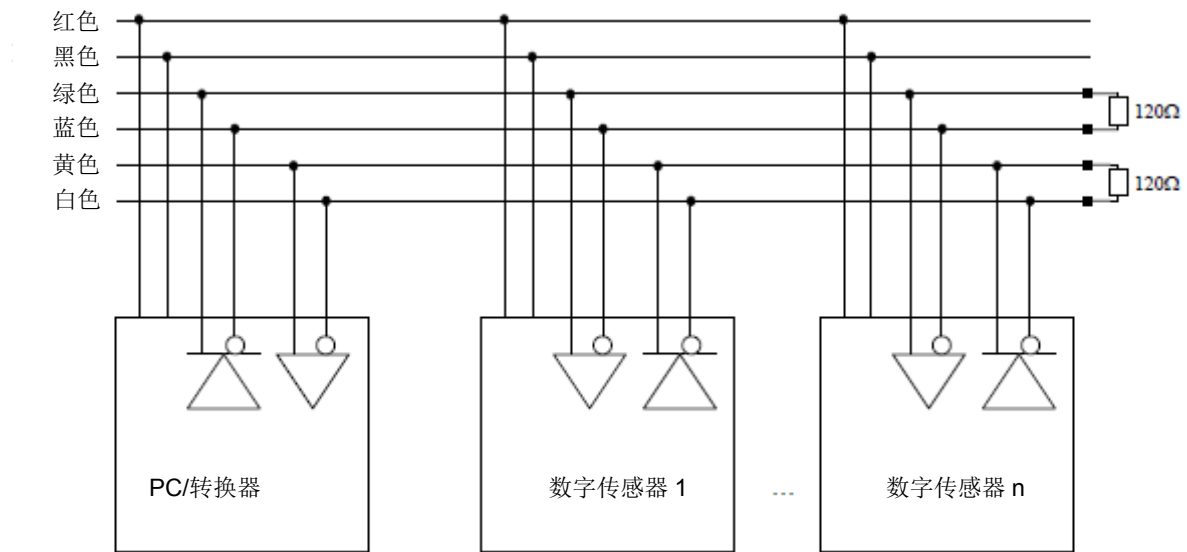
电缆代码和颜色:

| | | |
|-----|--------|------------------------------|
| 红色: | UB | +8V..... +18V DC, 60mA (最大值) |
| 黑色: | GND | |
| 绿色: | RB (+) | |
| 蓝色: | RA (-) | |
| 黄色: | TB (+) | |
| 白色: | TA (-) | |
| 屏蔽: | 扭绞、镀锡 | |

外部触发输入: 未在本版本中实现

2.3 通信

下图是总线连接略图。总线末端需要两只 120Ω 电阻，以匹配线路阻抗。在一根 RS-485 总线上最多可以连接 32 个数字传感器。



3 命令

命令可分类为：

- 接口命令（如：ADR）
- 调整和定标命令（如：ZER、GAI、NOM）
- 测量方式命令（如：FIL、VAL）
- 特殊命令（如：RES、RDV）

3.1 命令格式

命令必须以 ASCII 码和大写字母输入。

每条命令由命令缩写，设备地址、参数和终止符（回车 0Dh）组成。只有发送了终止字符时，从设备才会清除输入缓冲区，并将能够接受下一个命令。

命令描述中用圆括弧指定的参数“（参数）”是必须的，应当输入。方括弧中的参数“[参数]”是可选的。不用输入括弧本身。

被“<”和“>”符号包围的任何参数“<参数>”是 ASCII 非打印字符，由其十六进制数表示。

对于数字输入，前导零被禁止发送，数字必须以直接格式输入。对于负值，先输入负号（-），然后输入数字。

例如： ZER24, -452<CR> 将用户零点设置到 -452 计数。

3.2 对命令的应答。

对输入的应答：

| | 应答 | 结束字符 |
|------|------------------|------|
| 正确输入 | ACK (ASCII 0x06) | <CR> |
| 错误输入 | NAK (ASCII 0x15) | <CR> |

例外情况： 命令 VAL 发回测量值。
 广播命令不发回应答。

3.3 测量值的输出。

测量信号的输入取决于称重传感器的校验和配置，将以 ASCII 码发送。

无校验和的输出：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|----|
| S | W | W | W | W | W | W | W | CR |
|---|---|---|---|---|---|---|---|----|

有校验和的输出：

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|----|----|----|
| S | W | W | W | W | W | W | W | CH | CL | CR |
|---|---|---|---|---|---|---|---|----|----|----|

S - 符号：> 0 的测量值的空间；-是< 0 的值的空间

W - 重量：7 位，先发送最高有效位

CH , - 校验和，2 字符 ASCII 码

CL

CR - 终止符（回车 0x0D）

范围： - ± 9999999

3.4 命令概述：

| 命令 | 功能 | 说明章节 |
|-----|---------------|----------|
| ADR | 设置总线地址 | 第 4.1 节 |
| FIL | 设置滤波器 | 第 4.2 节 |
| VAL | 读取重量 | 第 4.3 节 |
| ZER | 校准用户零点 | 第 4.4 节 |
| GAI | 校准用户增益 | 第 4.5 节 |
| NOM | 设置标称定标 | 第 4.6 节 |
| RES | 设备复位 | 第 4.7 节 |
| RDV | 重新加载默认值 | 第 4.8 节 |
| VER | 软件版本 | 第 4.9 节 |
| TRG | 重量触发 | 第 4.10 节 |
| STU | 称重传感器状态 | 第 4.11 节 |
| BAU | 更改波特率 | 第 4.12 节 |
| CAP | 读取称重传感器标称称重能力 | 第 4.13 节 |
| CHK | 检验和 | 第 4.14 节 |

4 命令描述

4.1 **ADR.**设置总线地址。

范围: 00 – 32, 99
出厂设置: 00
参数: 1/2

输入: **ADR (设备地址), (新地址) [, 序号]<CR>**
输入设备地址, 为十进制数 00 - 32。

序号可以作为第 3 个参数发送。在这种情况下, 只为具有所输入序号的数字称重传感器编程新设备地址。因此, 可以更改具有相同编程地址的几个称重传感器的设备地址 (总线模式的初始化)。使用设备地址 99 会将设备地址为 00 的所有称重传感器更改到新地址, 并导致称重传感器发回 **ACK<CR>**。

例 1: **ADR01, 25<CR>**
设备地址为 01 的称重传感器将更改到设备地址 25。

例 2: **ADR00, 25, 456789<CR>** 序号为 456789 的称重传感器会将其设备地址更改为 25。在这种情况下, 序号为 456789 的称重传感器将发回 **ACK<CR>**。

例 3: 特殊情况设备地址 99:
ADR99, 25<CR>
所有设备地址为 00 (出厂默认) 的称重传感器会将其设备地址更改为 25。

发回: **ACK<CR>**

询问: **ADR (设备地址) ?<CR>**
发回称重传感器十进制数序号及其设备地址。

例如: **ADR25?<CR>**

发回: 11 个字符<CR> (如: 00456789: 25<CR>)

4.2 **FIL.**设置滤波器

范围: 0 - 6
出厂设置: 4
参数: 1

输入: **FIL (设备地址), (滤波器参数) <CR>**
输入具有此设备地址的称重传感器的滤波器参数。

例如: **FIL25, 6<CR>**
将设备地址为 25 的称重传感器的滤波器参数更改为 6

发回: **ACK<CR>**
询问: **FIL (设备地址) ?<CR>**
发回具有此设备地址的称重传感器的滤波器参数。

例如: **FIL25?<CR>**
发回: 11 个字符<CR> (如: 00000006: 25<CR>)

4.3 VAL. 读取重量

范围: ----
出厂设置: ----
参数: ----

输入: VAL (设备地址) <CR>
发回测量值 (格式见第 3.3 节)。

例如: VAL25<CR>
设备地址为 25 的称重传感器向主机发送测量值

发回: 8 个字符<CR> (如: -0052514<CR>)

询问: ----

1.009 版软件的新情况:

在 ADC 故障的情况下, 不会对 VAL 命令发送重量。

4.4 ZER. 校准用户零点

范围: \pm 标称定标 (见 NOM 命令)

出厂设置: 0

参数: 1

输入: ZER (设备地址) [, 零点值]<CR>

称重传感器零点的调整。此功能可通过使用称重传感器电子装置的自动校准完成, 或者通过直接输入零点值完成。

例 1: ZER25<CR>

称重传感器电子装置测量输入信号并将此值存储为用户零点。

例 2: ZER25, -0034567<CR>

不测量, 而是直接输入并存储零点。

发回: ACK<CR>

询问: ZER (设备地址) ?<CR>

发回零点值

例如: ZER25?<CR>

发回: 11 个字符<CR> (如: 00000014: 25<CR>)

4.5 **GAI**.校准用户增益

范围: ± 9.999999 (不允许为零)
出厂设置: 1.000000
参数: 1

输入: **GAI (设备地址), (用户增益) <CR>**
称重传感器用户增益的调整。

格式: 9.999999 (8 个字符) (正数)
+9.999999 (9 个字符) (正数)
<窗格>9.999999 (9 个字符) (正数)
-9.999999 (9 个字符) (负数)

例如: GAI25, 1.000050<CR>
未带符号输入并存储了增益。

发回: ACK<CR>

询问: **GAI (设备地址) ?<CR>**
发回用户增益值。

例如: GAI25?<CR>

发回: 如果为正: 11 个字符<CR>
(如: 1.000000: 25<CR>)
如果为负: 12 个字符<CR>
(如: -1.000000: 25<CR>)

4.6 **NOM.**设置标称定标

范围: 1 - 1000000

出厂设置: 200000

参数: 1

输入: **NOM** (设备地址), (标称定标) <CR>
设置称重传感器上标称负载的输出值。

例如: NOM25, 0250000<CR>
标称定标被输入并存储为 250000。

发回: ACK<CR>

询问: **NOM** (设备地址) ?<CR>
发回标称定标

例如: NOM25?<CR>

发回: 11 个字符<CR> (如: 00200000: 25<CR>)

4.7 RES.设备复位

范围: ----
出厂设置: ----
参数: ----

输入: **RES (设备地址) <CR>**
命令 RES 复位设备。

例 1: **RES25<CR>**
设备地址为 25 的称重传感器将在发送 ACK 后 100ms 内被复位。

例 2: **RES00<CR>**
连接在总线上的所有称重传感器将被复位。

发回: **ACK<CR>**
注意: 地址 00 (广播) 不发回任何信息

询问: **----**

4.8 RDV.重新加载默认值

范围: ----

出厂设置: ----

参数: ----

输入: RDV (设备地址) <CR>

命令 RDV 将所有参数设置并存储为出厂值。此后它会复位设备。

注意: 此功能还会将设备地址设置为 00，因此需要重新对其进行配置。

例 1: RDV00<CR>

连接在总线上的所有称重传感器将执行此命令

例 2: RDV25<CR>

设备地址为 25 的称重传感器将设置并存储出厂默认值，然后将复位。

发回: ACK<CR>

(注意: 地址 00 (广播) 不发回任何信息)

询问: ----

4.9 VER. 软件版本

范围: ----
出厂设置: ----
参数: ----

输入: ----

询问: **VER (设备地址) ?<CR>**
命令 VER 发回具有此特定设备地址的称重传感器上运行的软件的版本。

例如: VER25?<CR>
设备地址为 25 的称重传感器将发送其软件版本。

发回: 9 个字符<CR> (如: 01.003: 25)

4.10 TRG.重量触发

范围: ----
出厂设置: ----
参数: ----

输入: **TRG (设备地址) <CR>**
将当前重量值保存到内部存储空间。当发送了触发询问时将发回此值。

例如: **TRG25<CR>**
设备地址为 25 的称重传感器将测量值保存到一个内部存储器。

发回: **ACK<CR>**
(注意: 地址 00 (广播) 不发回任何信息)

询问: **TRG (设备地址) ?<CR>**
发回存储在内部存储空间中的值。

例如: **TRG25?<CR>**
设备地址为 25 的称重传感器发回存储的重量值。

发回: 8 个字符<CR> (如: -0068377<CR>)

1.009 版软件的新情况:

在 ADC 故障的情况下, 不会对 TRG 命令发送重量。

4.11 STU.称重传感器状态

范围: ----

出厂设置: ----

参数: ----

输入: ----

询问: **STU (设备地址) ?<CR>**

发回几位二进制数 (真 (1) 或假 (0))。

每位的含义如下:

- 第 0 位: 非易失性存储器损坏
- 第 1 位: ADC 故障 (ADC 不响应)
- 第 2 位: 重量读取错误 (ADC 发回一个错误)
- 第 3 位: 保留
- 第 4 位: 保留
- 第 5 位: 保留

(第 0 位是最先发送的字符)

此命令对确保从称重传感器读取的数据正确且测量中不出错是有用的。以固定的时间间隔 (如 5 分钟或 10 分钟) 进行询问比较方便。

从设备在内部故障的情况下不能向主设备发送任何数据, 因此此选项是有用的。主设备必须向从设备询问其内部状态。

例如: **STU25?<CR>**

设备地址为 25 的称重传感器发回其内部状态。

例如: “001000” <CR>表示没有任何信号连接到 ADC 输入。

发回: 6 个字符<CR> (如: 001000<CR>)

1.009 版软件的新情况:

在 ADC 故障的情况下, 不会对 VAL 和 TRG 命令发送重量。

4.12 **BAU.** 更改波特率

范围: 4800 波特、9600 波特、19200 波特、38400 波特
出厂设置: 19200 波特
参数: 1

输入: **BAU (设备地址), (波特率) <CR>**
更改通信波特率。
配置选项为 4800、9600、19200 和 38400。
其他所有波特率不能使用。

注意: 如果称重传感器正确地收到此命令, 它会使用**原来**的波特率发回 **ACK**。此后, 将使用**新的**波特率进行下一次通信。如果称重传感器没有正确地收到此命令, 它会发回一个**<NACK>**, 波特率保持不变。

例如: **BAU25, 38400<CR>**
波特率被更改为 38400bps。

发回: **ACK<CR>**

询问: **BAU (设备地址) ?<CR>**
发回当前波特率。

例如: **BAU25?<CR>**

发回: **11 个字符<CR>** (如: 00038400: 25<CR>)

4.13 CAP. 称重传感器标称称重能力

范围: ----

出厂设置: ----

参数: ----

输入: ----

询问: **CAP (设备地址) ?<CR>**

发回以千克为单位的称重传感器标称称重能力，带一位小数

此命令可以在刻度定义和无测试砝码的校准期间使用。

例如: **CAP25?<CR>**

设备地址为 25 的称重传感器发回其标称称重能力。

例如: "0030000.0: 25"<CR>, 对于 30000 kg 的称重传感器。

发回: 12 个字符<CR> (如: 0030000.0: 25<CR>)

4.14 **CHK.**发送的重量的校验和

范围: 0, 1, 2
出厂设置: 0 (OFF)
参数: 1

输入: **CHK (设备地址), (参数) <CR>**

启用或禁用发送的重量的校验和。

例 1: **CHK25, 1<CR>**

总线地址为 25 的称重传感器以发送带 XOR 校验和的重量响应 VAL 和 TRG 命令。

发回: **ACK<CR>**

例 2: **CHK25, 2<CR>**

总线地址为 25 的称重传感器以发送带 CRC8 校验和的重量响应 VAL 和 TRG 命令。

发回: **ACK<CR>**

询问: **CHK (设备地址) ?<CR>**

发回以千克为单位的称重传感器标称称重能力，带一位小数

例如: **CHK25?<CR>**

发回: 11 个字符<CR> (如: 00000001: 25<CR>)

校验和参数不会保存在 NVM 中，每次复位或上电后 **CHK** 参数将为 0 (OFF)。用户必须在需要时启用此功能。

5 修订历史记录

| 修订 | 日期 | 修改 | 软件版本 |
|----|------------|--|-------|
| 0 | 2004-05-12 | 原始文件 | 1.002 |
| 1 | 2004-05-24 | 增加了标题 2.3 | 1.002 |
| 2 | 2004-06-14 | 增加了命令： BAU 。 修改了命令： GAI （未启用零增益，新范围为 ± 9.999999 ）， ZER （新范围 \pm 标称标定，用户以“用户单位”输入和读取此零点） | 1.003 |
| 3 | 2005-11-30 | 最大工作电压改成了 18V 增加了命令： - CAP ：发送称重传感器的标称称重能力 修改了命令： - ADR ：设备地址'99'迫使所有地址为'00'的称重传感器更改为新地址 - 如果不是广播命令， RES 产生 ACK | 1.007 |
| 4 | 2007-05-03 | 增加了命令： - CHK ：计算重量的校验和 修改了命令： - VAL ：在 ADC 故障的情况下不会发送重量 - TRG ：在 ADC 故障的情况下不会发送重量 | 1.009 |
| 5 | 2009-03-25 | 最小工作电压： 从 7.5V 改成了 8.0V DC 功耗： 从 80mA 改成了 60mA | |

6 附录 校验和算法

1. XOR - 校验和:

我们提出的第一各校验和是一种使用 XOR（异或）的简单校验和算法。

例如，如果发送不带校验和的重量为：

| | | | | | | | | |
|--|---|---|---|---|---|---|---|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | CR |
|--|---|---|---|---|---|---|---|----|

| | 十六进制 | 二进制 | XOR |
|----|------|-----------|-----------|
| 空间 | 20 | 0010 0000 | |
| 1 | 31 | 0011 0001 | 0001 0001 |
| 2 | 32 | 0011 0010 | 0010 0011 |
| 3 | 33 | 0011 0011 | 0001 0000 |
| 4 | 34 | 0011 0100 | 0010 0100 |
| 5 | 35 | 0011 0101 | 0001 0001 |
| 6 | 36 | 0011 0110 | 0010 0111 |
| 7 | 37 | 0011 0111 | 0001 0000 |

因此所有 XOR 的和的结果为 0001 0000 = 10hex。

此校验和被加在值的末尾，因此

发送的带校验和的重量为：

| | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 1 | 0 | CR |
|--|---|---|---|---|---|---|---|---|---|----|

- Visual Basic 中编程的 XOR 校验和函数 -

```
Private Function CalcXor (ByRef data () As Byte, nbytes As Integer) As Byte
'-----
' Function:      (Private) CalcXor
' Description:   computes a 1 byte checksum using bytes 1 to nbytes of the data to be transmitted
'
' Parameter:     Data () As Byte      | data
'               nbytes as interger    | number of bytes in data
' Return Value:  ClacXOR as Byte      | XOR value
'-----
Dim x As Integer
Dim Checksum As Byte

Checksum = 0
For x = 1 To nbytes
    Checksum = data (x) Xor Checksum
Next x
CalcXor = Checksum

End Function
```

2. CRC8 - 校验和:

CRC8 基于多项式: $CRC8 = X^8 + X^5 + X^4 + 1$, 是比简单的 XOR 校验和更加可靠的众所周知并经常使用的生成数据传输校验和的标准。CRC8 也给出了校正数据块中位错误的可能性。我们不想将 CRC8 校验和计算的数学基础讲得太深, 但我们可以给出 VisualBasic 中编程的计算 CRC8 校验和的函数的示例。

- VisualBasic 中的 CRC8 校验和函数 -

```
Private Function CalcCRC8 (ByRef data () As Byte, nbytes As Integer) As Byte
'-----
' Function:      (Private) CRC8
' Description:   computes a 1 byte checksum using bytes 1 to nbytes of the data to be transmitted
'
' Parameter:     Data () As Byte      | data
'               nbytes as interger    | number of bytes in data
' Return Value:  CalcCRC8 as Byte     | CRC value
'-----

Dim crc, Index As Byte
Dim CRCTable1 () As Variant
Dim x As Integer

crc = 0
CRCTable1 = Array ( _
&H00, &H07, &H0E, &H09, &H1C, &H1B, &H12, &H15, _
&H38, &H3F, &H36, &H31, &H24, &H23, &H2A, &H2D, _
&H70, &H77, &H7E, &H79, &H6C, &H6B, &H62, &H65, _
&H48, &H4F, &H46, &H41, &H54, &H53, &H5A, &H5D, _
&HE0, &HE7, &HEE, &HE9, &HFC, &HFB, &HF2, &HF5, _
&HD8, &HDF, &HD6, &HD1, &HC4, &HC3, &HCA, &HCD, _
&H90, &H97, &H9E, &H99, &H8C, &H8B, &H82, &H85, _
&HA8, &HAF, &HAE, &HA1, &HB4, &HB3, &HBA, &HBD, _
&HC7, &HC0, &HC9, &HCE, &HDB, &HDC, &HD5, &HD2, _
&HFF, &HF8, &HF1, &HF6, &HE3, &HE4, &HED, &HEA, _
&HB7, &HB0, &HB9, &HBE, &HAB, &HAC, &HA5, &HA2, _
&H8F, &H88, &H81, &H86, &H93, &H94, &H9D, &H9A, _
&H27, &H20, &H29, &H2E, &H3B, &H3C, &H35, &H32, _
&H1F, &H18, &H11, &H16, &H03, &H04, &H0D, &H0A, _
&H57, &H50, &H59, &H5E, &H4B, &H4C, &H45, &H42, _
&H6F, &H68, &H61, &H66, &H73, &H74, &H7D, &H7A, _
&H89, &H8E, &H87, &H80, &H95, &H92, &H9B, &H9C, _
&HB1, &HB6, &HBF, &HB8, &HAD, &HAA, &HA3, &HA4, _
&HF9, &HFE, &HF7, &HF0, &HE5, &HE2, &HEB, &HEC, _
&HC1, &HC6, &HCF, &HC8, &HDD, &HDA, &HD3, &HD4, _
&H69, &H6E, &H67, &H60, &H75, &H72, &H7B, &H7C, _
&H51, &H56, &H5F, &H58, &H4D, &H4A, &H43, &H44, _
&H19, &H1E, &H17, &H10, &H05, &H02, &H0B, &H0C, _
&H21, &H26, &H2F, &H28, &H3D, &H3A, &H33, &H34, _
&H4E, &H49, &H40, &H47, &H52, &H55, &H5C, &H5B, _
&H76, &H71, &H78, &H7F, &H6A, &H6D, &H64, &H63, _
&H3E, &H39, &H30, &H37, &H22, &H25, &H2C, &H2B, _
&H06, &H01, &H08, &H0F, &H1A, &H1D, &H14, &H13, _
&HAE, &HA9, &HA0, &HA7, &HB2, &HB5, &HBC, &HBB, _
&H96, &H91, &H98, &H9F, &H8A, &H8D, &H84, &H83, _
&HDE, &HD9, &HD0, &HD7, &HC2, &HC5, &HCC, &HCB, _
&HE6, &HE1, &HE8, &HEF, &HFA, &HFD, &HF4, &HF3 )

For x = 1 To nbytes
Index = crc Xor data (x)
crc = CRCTable1 (Index)
Next x
CalcCRC8 = crc

End Function
```