

长英®

# LTM 9950D 液晶显示以太网 (Ethernet) 温湿度传感器 使用说明书

北京长英新业数码科技有限公司

2009-6

*不同的传感器!*

## 目 录

一 简介.....	2
二 技术指标.....	2
三 接线示意图和安装方向.....	2
四. 初始化网络参数.....	4
五. 开关量数据LCD显示 (LTM9950D) .....	5
六. 安装方法.....	5
七. TCP/UDP通讯协议 .....	6
八. SNMP 数据格式.....	10
九. 编程访问流程.....	13

## 一 简介

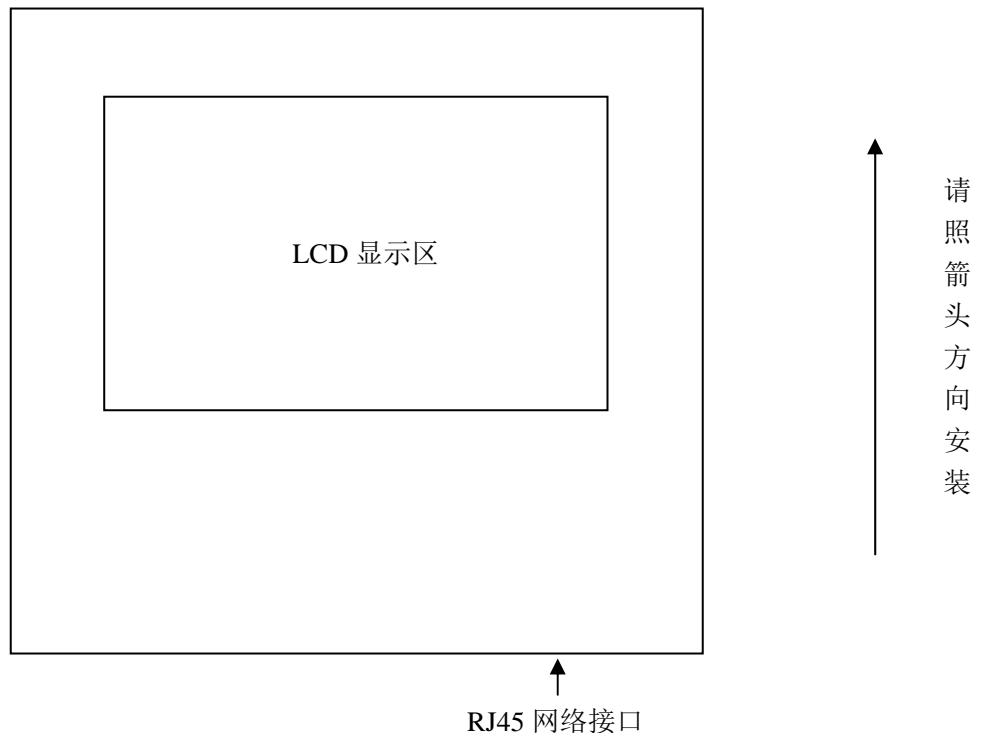
LTM9950D 为带液晶显示的以太网接口的温湿度传感器，同时还可监测 2 路开关量输入信号。随着网络的普及，LTM9950D 可以很方便的把测量点的温度，湿度，及现场开关量数据传到网络上，可以组建多种大小不同的现场监测应用系统。

## 二 技术指标

1. 温度测量范围:  $-25^{\circ}\text{C} \sim +60^{\circ}\text{C}$
2. 温度测量分辨率:  $0.0625^{\circ}\text{C}$  ( LCD 显示分辨率:  $0.1^{\circ}\text{C}$ )
3. 温度测量精度:  $\pm 0.5^{\circ}\text{C}$  ( $25^{\circ}\text{C}$  典型值),  $\pm 1.5^{\circ}\text{C}$  ( $-10^{\circ}\text{C} \sim +60^{\circ}\text{C}$ )
4. 湿度测量范围:  $0\% \sim 99\%$  ( 非凝结 )
5. 湿度测量分辨率:  $0.1\%\text{RH}$  ( LCD 显示分辨率:  $0.1\%\text{RH}$ )
6. 湿度测量精度:  $\pm 3\%\text{RH}$  ( 典型值 )
7. 开关量通道数: 2 路光电隔离独立输入
8. 开关量工作电压: 9-18VDC 输入时为: ON; 无电压时为: OFF;
9. 网络接口: RJ45 , 10/100M 自动识别
10. 支持协议: ARP, UDP, TCP, SNMP, Telnet, DHCP, AutoIP 等
11. TCP/UDP 端口: 10050 ( 缺省 )
12. SNMP 端口: 161
13. 供电电源:  $+6 \sim 36\text{V DC}$ , 300mA 以上 (推荐电压  $9 \rightarrow 12\text{VDC}$ )
14. 外形尺寸:  $98 \times 85 \times 45$  (mm)

## 三 接线示意图和安装方向

### 3.1 外观



### 3.2 内部接线端子定义



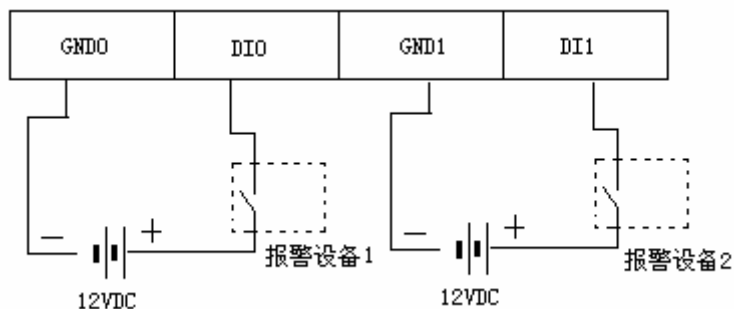
GND0	DI0	GND1	DI1
CN3: DI0 接口		CN4: DI1 接口;	

注 1: DIx ---- 接开关量输入信号 + 端 (信号+)

GNDx ---- 接开关量输入信号 - 端 (信号-)

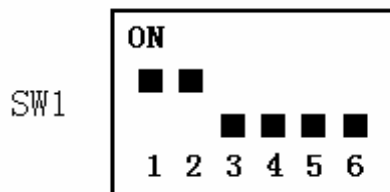
注 2: 不要采用与传感器相同的电源, 否则失去隔离作用

### 3.3 接线示意图

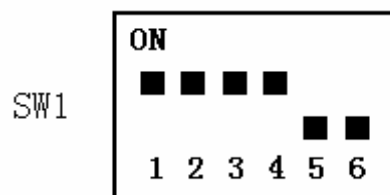


### 3.4 内部拨码定义

参见下图所示:



图A- 正常工作模式设置



图B- 网络参数初始化设置

注 1: 只有需要初始化网络参数时, 才会有图 B 所示设定。正常工作时, 请注意检查拨码设定, 按图 A 所示设定 SW1。如何初始化网络参数, 请参见 4.2 节。

注 2: SW1-6 保留, 正常工作时, 请保证其在 OFF 位置。

### 3.5 LCD 数据自动更新时间设定

SW1-5: --- OFF: 3.5 分钟 (缺省值)

SW1-5: --- ON : 5 分钟

注: 此时间为 LTM9950D 正常工作时, 如持续超过 3.5/5 分钟无网络数据读取访问, LTM9950 D 会自动进行一次数据采集, 更新数据显示。

## 四. 初始化网络参数

LTM9950 模块使用过程中, 用户可能多次更改其网络参数, 如: IP, MASK 等。在某些情况下, 用户可能忘记其参数设定, 或把模块从一个子网环境移动到另一子网环境中使用, 软件有可能搜索不到 LTM9950 模块。此时可利用 LTM9950 的网络参数初始化功能, 使其 IP 地址及其它参数回复到出厂设定, 便于正常使用。

### 4.1 LTM9950 模块的网络初始化参数:

- IP = 0.0.0.0
- MASK = 0.0.0.0
- GATWAY = 0.0.0.0
- 工作模式 = TCP 模式

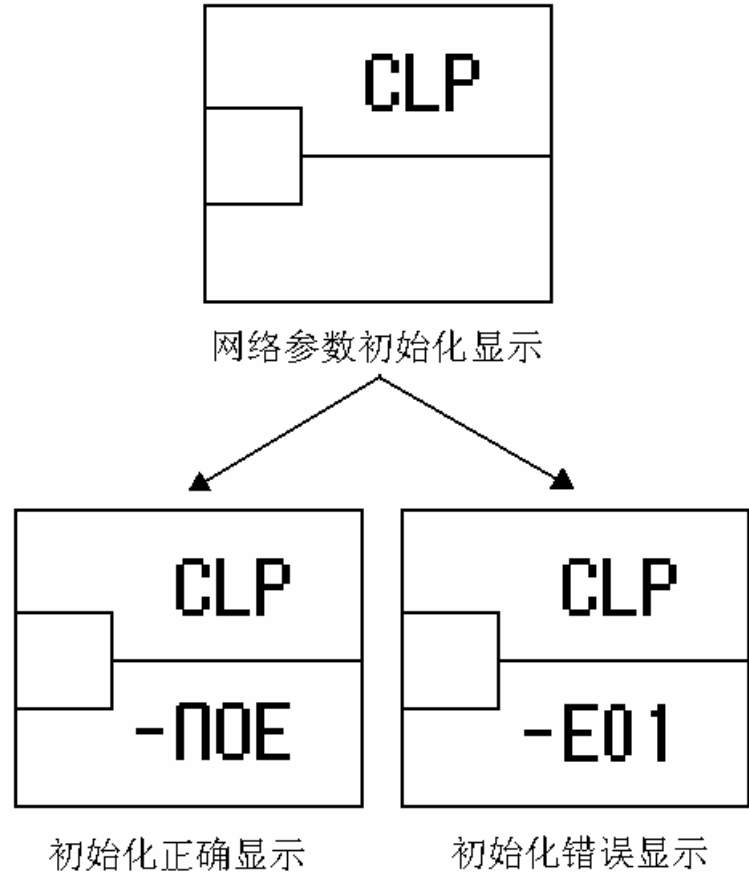
注: LTM9950 模块网络参数初始化后, 如子网中有 DHCP 服务器 (一般的有线/无线路由器及部分交换机均有此功能), LTM9950 模块会自动获取一个 IP 地址。如 LTM9950 模块搜索不到 DHCP 服务器, LTM9950 模块会自动给自己分配一个 IP 地址 (AutoIP), 其范围为:

169.254.0.1 → 169.254.255.1.

### 4.2 初始化网络参数步骤

请按如下步骤操作:

- 第一步: 模块断电, 按 3.3 节 图 B 所示设定 SW1。
- 第二步: 模块上电, 初始化约需 10-15 秒钟。此过程中 LTM9950D 模块会有如下显示:



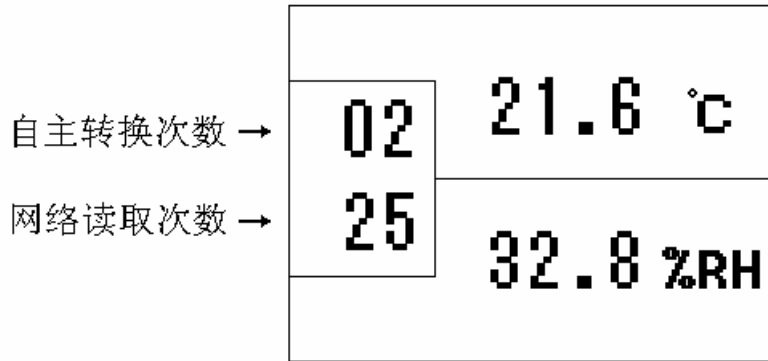
之后, LTM9950D 会自动显示一下模块名称及版本号后, 进入正常工作模式。

第三步: 模块断电, 按 3.3 节将 SW1 由 图 B →图 A (B 拨至 A), 回复到正常工作模式。

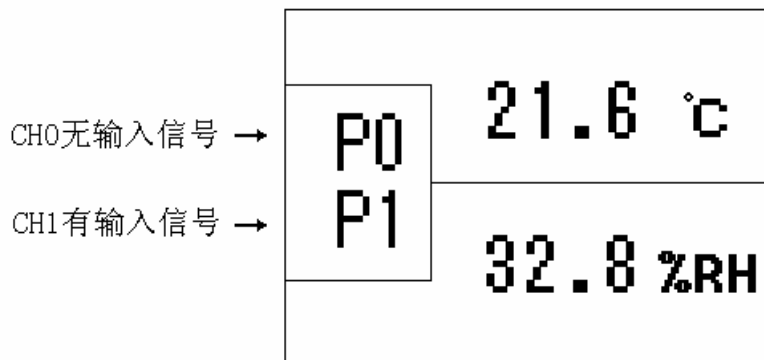
第四步: 模块重新上电, 至此初始化工作全部完成。

## 五. 开关量数据 LCD 显示 (LTM9950D)

仅有 LTM9950D 可以显示 2 路开关量数据 (LTM9950 不可以)。显示界面在如下两个界面间交替显示, 利于观察各个数据。切换时间为 2 秒钟。



正常工作显示



开关量输入显示

注 1: 两个计数值 (自主转换次数/网络读取次数) 各自累计自己的计数值并显示出来, 当数值  $\geq 100$  时, 会自动环回到 00, 重新开始计数。

注 2: 开关量显示值为: P0 时, 表示此通道无信号输入; P1 时, 表示此通道有信号输入。  
如上图所示, 则表示: CH0 无信号输入, CH1 有信号输入。

## 六. 安装方法

按接线示意图连接好网线及电源线即可。注意: 需垂直竖向安装。每个 LTM9950D 出厂时有缺省 IP 地址, 一般为: 0.0.0.0 (子网掩码: 0.0.0.0), 接入网络时, 可通过 DHCP 自动获取 IP。如要设定, 要注意不要与网络上其他设备的 IP 地址相冲突。连接完毕后, 可在网络上的某台 PC 机上运行 NET 程序搜索和显示温湿度数据。

## 七. TCP/UDP 通讯协议

LTM9950D 的数据通信协议采用 Modbus 协议的 ASCII 模式，数据读取指令可通过 TCP/UDP 传送至模块。以下是详细说明。

数据读取指令有两条，一条为与 8950X 兼容的老指令，称为 CMD1。另一条为新增指令，称为 CMD2。

CMD1 命令主要为老用户的硬件与 LTM9950 的兼容性而保留，使用上与 8950X 完全一致。但 CMD1 只可读取原有的数据单元，不可读取新增的开关量数据。

CMD2 命令，为新增指令，除可读取 CMD1 的所有数据，还可读取新增的开关量数据。因此，建议：新用户尽量使用新增命令，即 CMD2。

### 7.1 CMD1 数据读取指令 (8950x/LTM9950x):

(均为 ASCII 字符表示的十六进制数据，共 17 个字符)

**:000450000008 S<sub>1</sub> S<sub>2</sub> CR LF**

依照数据的顺序所代表的含义依次时

**:** 帧起始符

**00** LTM9950 地址(2 个字符)

**04** 读取命令(2 个字符)

**50** LTM9950 内存起点高位(2 个字符)

**00** LTM9950 内存起点低位(2 个字符)

**00** 读取数据数量高位(2 个字符)

**08** 读取数据数量低位(2 个字符)

**S<sub>1</sub> S<sub>2</sub>** 校验和(2 个字符)

**CR LF** 数据帧结束符

注:

① 编成时可以直接用指令“:000450000008A4(CR)(LF)”来读取数据，无需改动。

② 其中的 CR 及 LF 的 16 进制表示方式为: 0X0D, 0X0A

③ 校验和的算法:

起始符“:”不算，从 LTM9950 地址开始等加至数据数量低位字节结束，然后求其补码即可。

例: 如上述命令 (以下数据为 16 进制数据)

$$\begin{aligned} \text{SUM} &= 00 + 04 + 50 + 00 + 00 + 08 \\ &= 54 + 08 \\ &= 5C(H) \end{aligned}$$

$$\text{SUM} = 100(H) - 5C(H) = A4(H)$$

即 S<sub>1</sub> = 'A' S<sub>2</sub> = '4'

④ 对于 LTM9950x，此命令中的地址，应为 00 不变。

### 7.2 CMD1 数据应答: (均为 ASCII 字符表示的 16 进制数据，共 27 个字符)

(用 X1, X2 表示一个数据的高 4 位字节字符及低 4 位字节字符)

**:A<sub>1</sub> A<sub>2</sub> 0 4 0 8 5 0 V<sub>1</sub> V<sub>2</sub> TH<sub>1</sub> TH<sub>2</sub> TL<sub>1</sub> TL<sub>2</sub> HH<sub>1</sub> HH<sub>2</sub> HL<sub>1</sub> HL<sub>2</sub> R<sub>1</sub> R<sub>2</sub> R<sub>3</sub> R<sub>4</sub> S<sub>1</sub> S<sub>2</sub> CR LF**

**:** 帧起始符

**A1 A2** LTM9950 地址(2 个字符) (注: 此地址无具体意义, 仅为协议保留, 区分不同 LTM9950x 模块可根据其 IP 地址或 MAC 码)。

**04** 命令符(2 个字符)

**08** 数据数量(2 个字符)

**50** 类型码(2 个字符)

**V1 V2** 传感器版本号(2 个字符)

**TH1 TH2** 温度数据高位(2 个字符)

**TL1 TL2** 温度数据低位(2 个字符)

**HH1 HH2** 湿度数据高位(2 个字符)

**HL1 HL2** 湿度数据低位(2 个字符)

**R1 - R4** 保留(4 个字符)

**S1 S2** 校验和(2 个字符)

**CR LF** 帧结束符(2 个字符)

注: 校验和算法与 7.1 注 2 相同

示例: 如 LTM9950 应答帧如下:

**:0504085063217C00A00310ECR LF**

即: LTM9950 的传感器地址为: 05H

LTM9950 的传感器版本号为: 63H

LTM9950 的温度数据为: 21, 7CH

LTM9950 的湿度数据为: 00, A0H

校验和 (以下数据为 16 进制数)

$$\begin{aligned} \text{SUM} &= 05 + 04 + 08 + 50 + 63 + 21 + 7C + 00 + A0 + 03 + 10 \\ &= 14(\text{H}) \end{aligned}$$

$$\text{SUM} = 100\text{H} - 14\text{H} = \text{ECH} \quad \text{与应答帧中 } \underline{\text{E}} \underline{\text{C}} \text{ 相同。}$$

### 7.3 CMD2 数据读取指令 (LTM9950x):

( 均为 ASCII 字符表示的十六进制数据, 共 17 个字符 )

**:000450800008 S1 S2 CR LF**

依照数据的顺序所代表的含义依次时

**:** 帧起始符

**00** LTM9950 地址(2 个字符)

**04** 读取命令(2 个字符)

**50** LTM9950 内存起点高位(2 个字符)

**80** LTM9950 内存起点低位(2 个字符)

**00** 读取数据数量高位(2 个字符)

**08** 读取数据数量低位(2 个字符)

**S1 S2** 校验和(2 个字符)

**CR LF** 数据帧结束符

注:

其中的 CR 及 LF 的 16 进制表示方式为: 0X0D, 0X0A

编成时可以直接用指令 “:00045080000824(CR)(LF)” 来读取数据, 无需改动。



校验和的算法:

起始符“:”不算, 从 LTM9950 地址开始等加至数据数量低位字节结束, 然后求其补码即可。

例: 如上述命令 (以下数据为 16 进制数据)

$$\begin{aligned} \text{SUM} &= 00 + 04 + 50 + 80 + 00 + 08 \\ &= 54 + 88 \\ &= \text{DC (H)} \end{aligned}$$

$$\text{SUM} = 100 (\text{H}) - \text{DC} (\text{H}) = 24 (\text{H})$$

即 S<sub>1</sub> = ‘2’ S<sub>2</sub> = ‘4’

对于 LTM9950x, 此命令中的地址, 应为 00 不变。

#### 7.4 CMD2 数据应答: (均为 ASCII 字符表示的 16 进制数据, 共 43 个字符)

(用 X<sub>1</sub>, X<sub>2</sub> 表示一个数据的高 4 位字节字符及低 4 位字节字符)

**:** **A<sub>1</sub>A<sub>2</sub>** **041050** **V<sub>1</sub>V<sub>2</sub>** **TH<sub>1</sub>TH<sub>2</sub>** **TL<sub>1</sub>TL<sub>2</sub>** **HH<sub>1</sub>HH<sub>2</sub>** **HL<sub>1</sub>HL<sub>2</sub>** **R<sub>1</sub>R<sub>2</sub>R<sub>3</sub>R<sub>4</sub>**  
**DI<sub>1</sub>DI<sub>2</sub>** **R<sub>5</sub>R<sub>6</sub>** **R<sub>7</sub>R<sub>8</sub>R<sub>9</sub>R<sub>10</sub>** **R<sub>11</sub>R<sub>12</sub>R<sub>13</sub>R<sub>14</sub>** **R<sub>15</sub>R<sub>16</sub>R<sub>17</sub>R<sub>18</sub>** **S<sub>1</sub>S<sub>2</sub>** **CR LF**

: 帧起始符

**A<sub>1</sub> A<sub>2</sub>** LTM9950 地址(2 个字符) (注: 此地址无具体意义, 仅为协议保留, 区分不同 LTM9950x 模块可根据其 IP 地址或 MAC 码)。

**04** 命令符(2 个字符)

**10** 数据数量(2 个字符) (注: 10h, 即 16 个字节, 32 个字符)

**50** 类型码(2 个字符)

**V<sub>1</sub> V<sub>2</sub>** 传感器版本号(2 个字符)

**TH<sub>1</sub> TH<sub>2</sub>** 温度数据高位(2 个字符)

**TL<sub>1</sub> TL<sub>2</sub>** 温度数据低位(2 个字符)

**HH<sub>1</sub> HH<sub>2</sub>** 湿度数据高位(2 个字符)

**HL<sub>1</sub> HL<sub>2</sub>** 湿度数据低位(2 个字符)

**R<sub>1</sub> - R<sub>4</sub>** 保留(4 个字符)

**DI<sub>1</sub> DI<sub>2</sub>** 开关量数据(2 个字符)

**R<sub>5</sub> - R<sub>18</sub>** 保留(14 个字符) (注: 此处为保留备用数据位置)

**S<sub>1</sub> S<sub>2</sub>** 校验和(2 个字符)

**CR LF** 帧结束符(2 个字符)

注: 校验和算法与 7.1 注 2 相同

示例: 如 LTM9950 应答帧如下:

**:0504105070217C00A00310 0200 0000 0000 0000 DD CR LF**

即: LTM9950 的传感器地址为: 05H

LTM9950 的传感器版本号为: 70H

LTM9950 的温度数据为: 21, 7CH

LTM9950 的湿度数据为: 00, A0H

LTM9950 的开关量数据为: 02H (即: CH<sub>0</sub>-无信号输入, CH<sub>1</sub>-有信号输入, 可参见 7.7 定义)

校验和 (以下数据为 16 进制数)

$$\begin{aligned} \text{SUM} &= 05 + 04 + 08 + 50 + 70 + 21 + 7C + 00 + A0 + 03 + 10 + 02 \\ &= 23(\text{H}) \end{aligned}$$

SUM= 100H - 23H = DDH 与应答帧中 **DD** 相同。

### 7.5 温度数据定义:

(温度) 应答数据格式 :

TH							
7	6	5	4	3	2	1	0
001—温度数据		0		0	X	X	X
000—湿度数据		0		0	X	X	X
数据处理类型		恒为零		FLG	TMP—H		
×	×	×			2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>

TL							
7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X
TMP—L							
2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>

温度数据:

TH— BIT3 为符号位, BIT3=1, 为负温, BIT3=0 为正温。

TH 中, BIT7~5 为数据类型主义, BIT4 恒为 0

TH 中 BIT2~0 及 TL 为温度数据

温度值分辨率为: 0.0625°C (2<sup>-4</sup>位)

温度计算公式为:

正温:  $\text{TMP} = ((\text{TH} \& 07\text{H}) * 256 + \text{TL}) * 0.0625$

负温:  $T_1 = \text{TH} \& 07\text{H}$

$\text{TMP} = - (T_1 * 256 + \text{TL}) * 0.0625$

**注: 若 LTM9950 的温度传感器有故障, HH, HL 均为 FFH**

### 7.6 湿度数据定义:

湿度算法:

湿度值 = ( HH \* 256 + HL ) / 10

HH 湿度数据高位

HL 湿度数据低位

**注: 若 LTM9950 的湿度传感器有故障, HH, HL 均为 FFH**

把上例数据带入算法, HH = 00H HL = A0H

则: 湿度值 = ( 0 \* 256 + 160 ) / 10 = 16%RH

即: 16.0 %RH

### 7.7 开关量数据定义:

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
---	---	---	---	---	---	CH1	CH0

BIT7 --- BIT2: 保留备用;

CH1: = 1, 通道 1 有信号输入; =0, 通道 1 无信号输入;

CH0: = 1, 通道 0 有信号输入; =0, 通道 0 无信号输入;

## 7. 8 访问时间间隔

注: TCP/UDP访问时间间隔应  $\geq 1S$ , 以便LTM9950D模块更新数据。

# 八. SNMP 数据格式

## 8. 1 简介

LTM9950D模块支持 SNMP v1 协议。OID实例(object identifier)为以下5个:

- 1— **1.3.6.1.4.1.7777.1.3.1.1.0** --- 温度数据 (整数, 以0.1℃为分辨率)
- 2— **1.3.6.1.4.1.7777.1.3.1.2.0** --- 湿度数据 (整数, 以0.1%RH为分辨率)
- 3— **1.3.6.1.4.1.7777.1.3.1.3.0** --- 开关量数据 (整数, 详见8.3说明)
- 4— **1.3.6.1.4.1.7777.1.3.1.4.0** --- 整合数据 (字符串, 详见8.3说明)
- 5— **1.3.6.1.4.1.7777.1.3.1.5.0** --- 整合数据 (整数, 详见8.3说明)

注: SNMP访问时间间隔应  $\geq 1S$ , 以便LTM9950D模块更新数据。

## 8. 2 9950MIB 定义

以下为LTM9950D模块支持的 MIB文件定义, 可将以下内容存为一个名称为: LTM9950.MIB 的文本文件, 客户可在自己所用的SNMP(或MIB)管理软件中, 添加至其MIB数据库中, 便可访问以上所列5个OID的数据内容了。

```
M9950-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    enterprises, IpAddress, Counter, TimeTicks
```

```
    FROM RFC1155-SMI
```

```
OBJECT-TYPE
```

```
    FROM RFC-1212
```

```
    DisplayString
```

```
    FROM RFC-1213;
```

```
magic7          OBJECT IDENTIFIER ::= { enterprises 7777 }
```

```
products        OBJECT IDENTIFIER ::= { magic7 1 }
```

```
temphumio       OBJECT IDENTIFIER ::= { products 3 }
```

```
readings        OBJECT IDENTIFIER ::= { temphumio 1 }
```

```
tempv OBJECT-TYPE
```

```
    SYNTAX      INTEGER(-400..1250)
```

```
    ACCESS      read-only
```

```
    STATUS      mandatory
```

```

DESCRIPTION    "Temp data in 0.1c resolution.(e.g. 356 => 35.6c)"
::= { readings 1 }

humv OBJECT-TYPE
    SYNTAX      INTEGER(0..1000)
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION  "Hum data in 0.1%RH resolution.(e.g. 258 => 25.8%RH)"
::= { readings 2 }

ioinv OBJECT-TYPE
    SYNTAX      INTEGER(0..3)
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION  "IO in ch1 & ch0; ch0 on=1,ch1 on=2"
::= { readings 3 }

stringv OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..31))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION  "temp-hum-io <T=xxx°C; H=xxx%RH; IO=xx> value in string format."
::= { readings 4 }

allv OBJECT-TYPE
    SYNTAX      INTEGER(-9876543210..987654321)
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION  "all data value in one integer 32 bit"
::= { readings 5 }

END

```

### **8.3 OID 说明**

#### **8.3.1 温度数据:**

为: -400 → 1250 之间的一个整数, 是以 0.1°C 为分辨率表示的。

实际温度值计算方法:  $T = N * 0.1$  (°C)

例如:  $N = 285$ , 则  $T = 285 * 0.1 = 28.5$ °C

$N = -39$ , 则  $T = -39 * 0.1 = -3.9$ °C

<注: 当此数值 = 3822 (EEEh, 16 进制数) 时, 表示温度传感器有故障 >

#### **8.3.2 湿度数据:**

为: 0 → 1000 之间的一个整数, 是以 0.1%RH 为分辨率表示的。

实际湿度值计算方法:  $H = N * 0.1$  (%RH)

例如:  $N = 386$ , 则  $H = 386 * 0.1 = 38.6\%RH$

<注: 当此数值 = 3822 (EEh, 16 进制数) 时, 表示湿度传感器有故障 >

**8.3.3 开关量数据:**

数据为单字节。其位定义如下:

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
---	---	---	---	---	---	CH1	CH0

BIT7 --- BIT2: 保留备用;

CH1: = 1, 通道 1 有信号输入; =0, 通道 1 无信号输入;

CH0: = 1, 通道 0 有信号输入; =0, 通道 0 无信号输入;

**8.3.4 字符串—整合数据**

此项整合了所有的温度, 湿度, 开关量数据, 用字符串的形式表述出来。此项目的目的在于显示直观明了, 且一项包括了所有的数据。其数据与前面的单项数据是同时更新的。

例如:

字符串 S = "T=035.6C; H=025.6%RH; IO=1-0;"

表示: 温度值=35.6C, 湿度值=25.6%RH, 开关量= 1-0 (通道 1 有信号);

字符串 S = "T=-03.5C; H=056.7%RH; IO=1-1;"

表示: 温度值= -3.5C, 湿度值=56.7%RH, 开关量= 1-1 (通道 0,1 均有信号);

**8.3.5 整数—整合数据**

此项整合了所有的温度, 湿度, 开关量数据, 用一个 32 位的整数 (4 字节) 的形式表述出来。此项目的目的在于使一项包括所有的数据, 便于特定用户 (如自主开发程序者) 使用一条 SNMP 读取命令即可读取所有数据。使用 SNMP (或 MIB) 管理软件的用户可考虑不使用此 OID。其数据与前面的单项数据是同时更新的。

以下为数据定义 (4 字节, 按读取顺序):

字节 1	字节 2	字节 3	字节 4
TM	HM	TL-HL	IO

其中<B7,B6,B4,B3,B0 是指字节(8 位)数据的第 7, 6, 4, 3, 0 位>:

TM ---- B6→B0 为温度数据的整数部分, 最高位 B7 为符号位; B7=1, 为负值;

HM ---- 为湿度数据的整数部分;

TL-HL ---- TL: 高 4 位(B7→B4) 为温度数据的小数部分; 表示值为: 0.0—0.9;

HL: 低 4 位(B3→B0) 为湿度数据的小数部分; 表示值为: 0.0—0.9;

IO ---- 为开关量数据; 定义参见: 8.3.3 节。

**注 1: 当TM = EEh, TL = Eh 时, 表示温度传感器有故障;**

**注 2: 当HM = EEh, HL = Eh 时, 表示湿度传感器有故障;**

例 1:

4 字节数据为: 18 25 56 02 (16 进制数据)

则:  $TM = 18h = 24\text{ C}$  (最高位为 0, 为正值)

$HM = 25h = 37\%RH$

$TL-HL = 56h$ ;  $TL = 5$ , 小数部分为 0.5C;  $HL = 6$ , 小数部分为 0.6%RH

温度值  $T = 24.5C$ ; 湿度值  $H = 37.6\%RH$

开关量数据  $IO = 02h$ , 即表示通道 1 有信号, 通道 0 无信号。

例 2:

4 字节数据为: 87 12 39 01 (16 进制数据)

则:  $TM = 87h$ , 最高位为 1, 为负值,  $TM = -7\text{ C}$

$HM = 12h = 18\%RH$

$TL-HL = 39h$ ;  $TL = 3$ , 小数部分为 0.3C;  $HL = 9$ , 小数部分为 0.9%RH

温度值  $T = -7.3C$ ; 湿度值  $H = 18.9\%RH$

开关量数据  $IO = 01h$ , 即表示通道 1 无信号, 通道 0 有信号。

## 九. 编程访问流程

- ① 与 LTM9950 模块建立 TCP/UDP/SNMP 连接 (使用 IP 地址, 端口号)
- ② 发送数据读取命令帧
- ③ 接受数据 (LTM9950 应答帧)
- ④ 数据换算, 计算出温/湿度数据
- ⑤ 若重复访问 LTM9950, 间隔需  $\geq 1$  秒钟, 回到第二步.
- ⑥ 若不再访问 LTM9950, 断开连接, 关闭网络接口。

附录:

V1.0 ---2009.02.26

初版。

V1.0.1 ---2009.03.30

部分文字修改。

V1.1.0 ---2009.06.11

增加 SNMP 部分说明。