

## DL11-MC-Xx 说明书

文件版本: V1.0.0



## 功能特点

- ◆ 支持 Modbus RTU 协议读写寄存器
- ◆ 支持 USB 转串口波特率 1200bps~115200bps
- ◆ 支持 NTC 或 DS18B20 温度传感器
- ◆ 支持 1 路、2 路、3 路或 6 路高精度温度传感器
- ◆ 温度传感器防尘防水
- ◆ 温度传感器温度范围为-40~125℃ (可定制高温传感器)
- ◆ 温度传感器采用不锈钢外壳 (可定制外壳)
- ◆ 温度传感器线长可定制

## 目录

1. 快速入门 .....	4
1.1. 硬件测试环境 .....	4
1.1.1 硬件准备 .....	4
1.1.2 硬件连接 .....	4
1.1.3 默认参数测试 .....	5
2. 产品概述 .....	9
2.1. 产品简介 .....	9
2.2. 基本参数 .....	9
2.3. 型号定义 .....	10
3. 通信协议 .....	12
3.1 命令帧格式 .....	12
3.2 Modbus 寄存器 .....	12
3.3 Modbus 寄存器参数说明: .....	13
3.4 Modbus 命令格式 .....	15
4. 联系方式 .....	22
5. 免责声明 .....	22
6. 更新历史 .....	22

## 1. 快速入门

DL11-MC-Xx 系列 USB 温度传感器通过 USB 接口采用标准 MODBUS RTU 协议通信,采用高精度的测量单元,测量精准;可以现场通过软件设置内部参数,一次设置永久保存;广泛适用于冷藏车、通讯机房、仓库楼宇以及各种自控需要温度检测的场所;

本章是针对 DL11-MC-Xx 系列产品的快速入门介绍,建议用户系统的阅读本章并按照指示操作一遍,便于对本产品有一个系统的认识,用户也可以根据需求选择感兴趣的章节阅读。针对特定的细节和说明,请参考后续章节。

除此说明书外,我们同时提供了基于本产品的软件设计手册、数据采集软件等资料,方便用户设计参考。

USB转串口驱动:  CH341SER.EXE

下载地址: <http://dalintech.cn>

### 1.1. 硬件测试环境

#### 1.1.1 硬件准备

快速测试所需如下:

- DL11-MC-D1 一个
- 电脑一台



图 1 硬件准备材料

##### 1. 测试目的

- 快速了解 DL11-MC-Xx
- 测试产品基础功能

#### 1.1.2 硬件连接



图 2 硬件连接示意图

### 1.1.3 默认参数测试

项目	内容
设备地址	1
串口波特率	9600
串口参数	None/8/1
温度寄存器地址	0400 H

表 1 设备默认参数

在插入设备前需要安装 USB 转串口驱动，安装完成后会生成 COM 口，按对应的 COM 口配置软件；我们只需要按照以下需要设置相应的通信参数，就可以进行 USB 转串口通信了，测试可选如下三种方式：  
测试方式一：

- 1) 打开 DL-MDM 采集监控平台

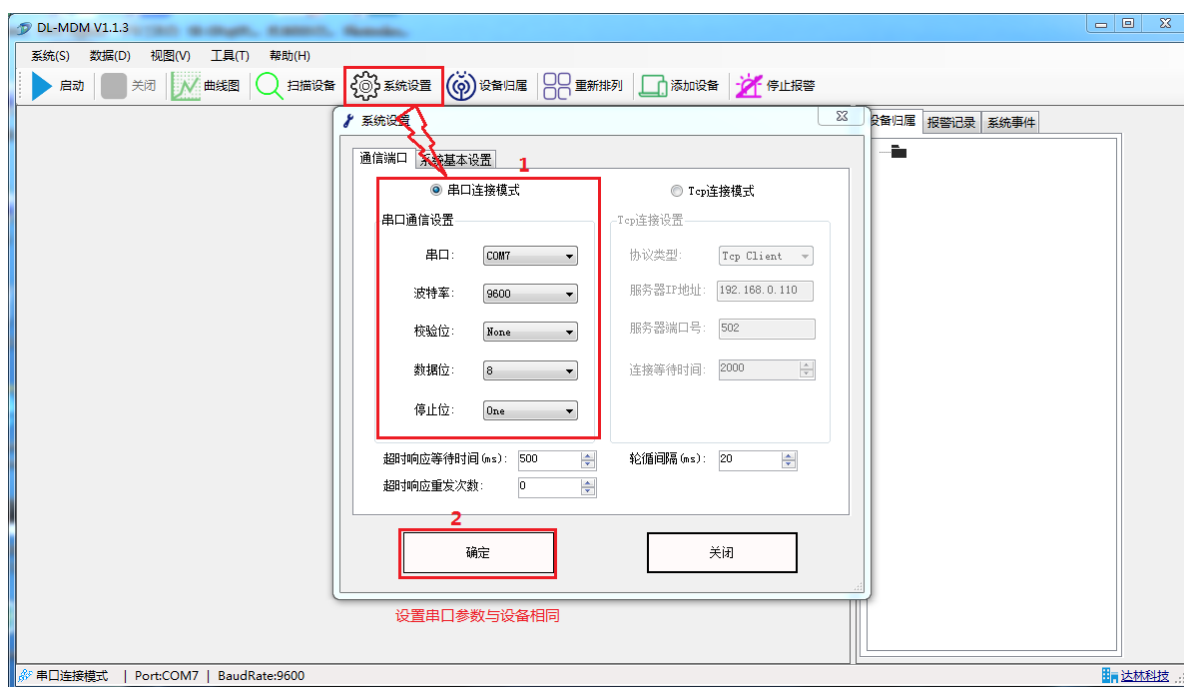


图 3 DL-MDM 参数设置

## 2) 扫描查找设备

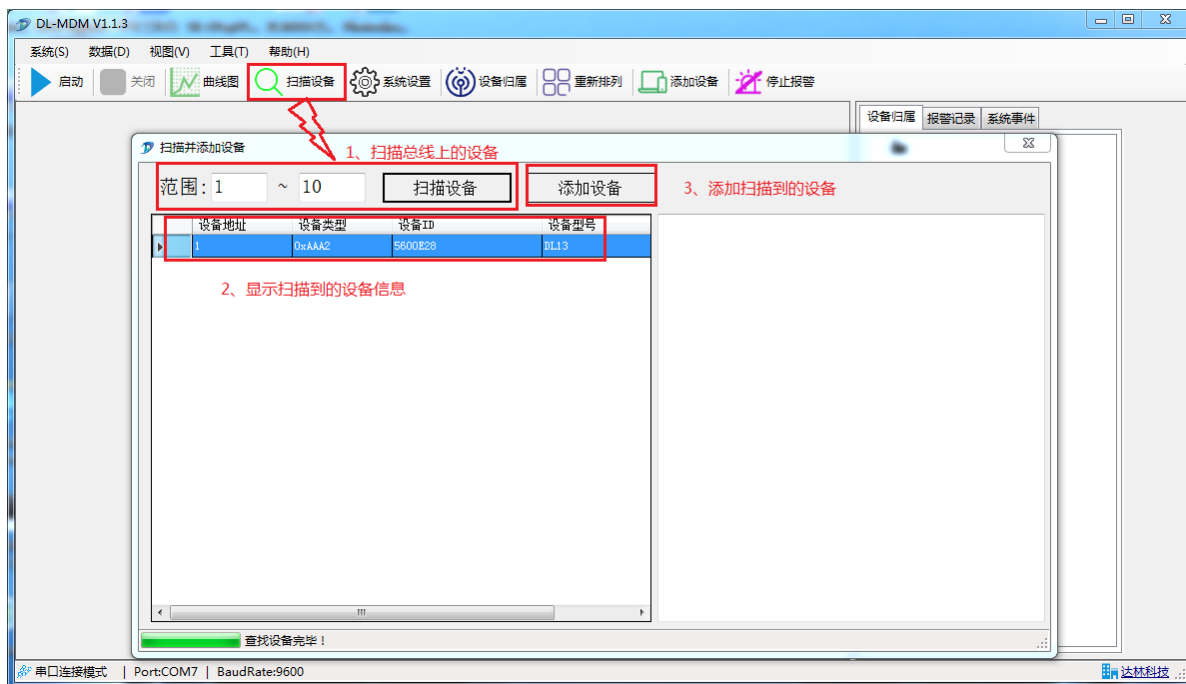


图 4 DL-MDM 扫描查找设备

## 3) 启动读数据

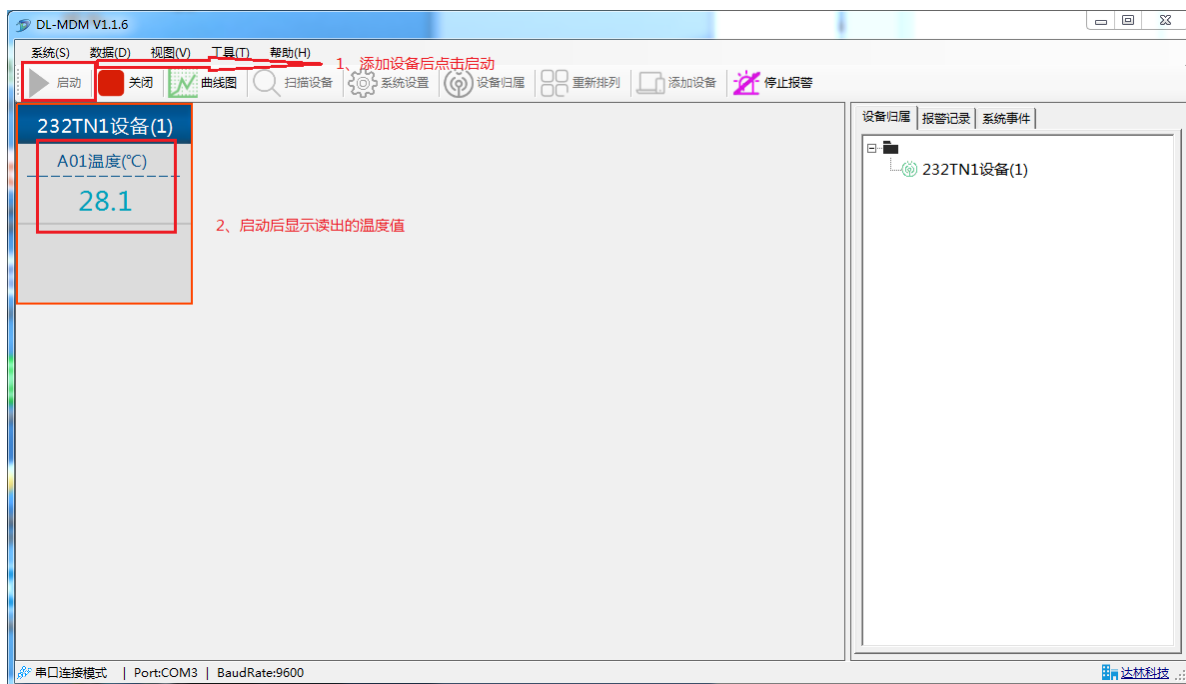


图 5 DL-MDM 启动读数据

## 测试方式二：

- 1) 打开 Modbus Poll 测试软件

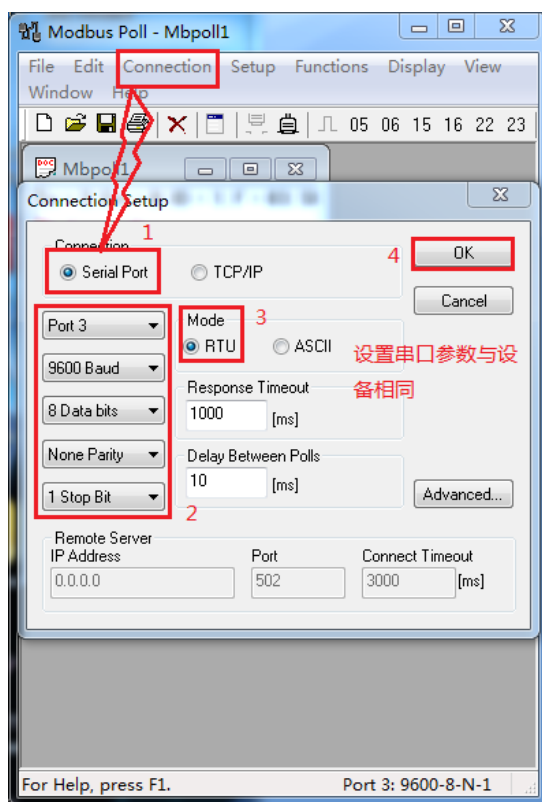


图 6 串口参数设置

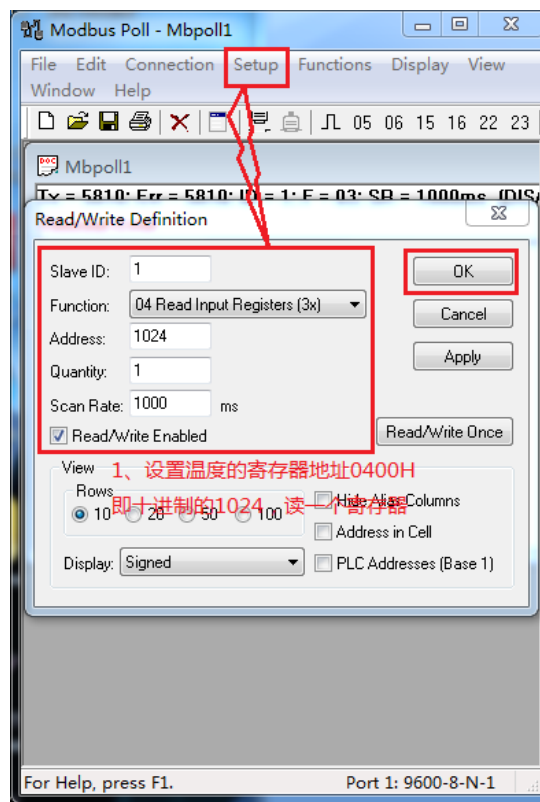


图 7 寄存器设置

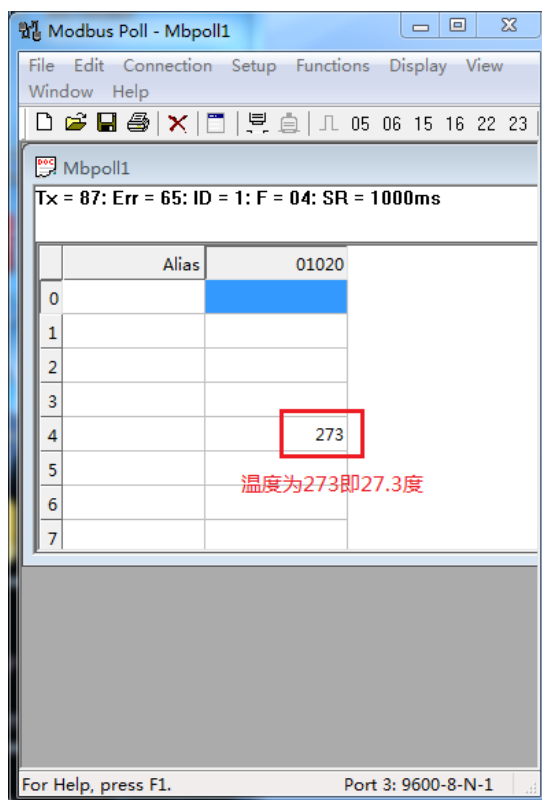


图 8 温度值

测试方式三:

- 1) 用固定命令读出温度值(十六进制)

温度个数	命令	应答
1 路	01 04 04 00 00 01 30 FA	01 04 02 01 17 F8 AE
2 路	01 04 04 00 00 02 70 FB	01 04 04 01 17 7F FF 2A 0C
3 路	01 04 04 00 00 03 B1 3B	01 04 06 01 17 7F FF 7F FF 5C D5
6 路	01 04 04 00 00 06 71 38	01 04 0C 01 1A 7F FF 7F FF 7F FF 7F FF 7F FF 74 A7

红色和蓝色分开即为对应的温度值为 2byte: 01 17 → 279 即 27.9℃

7F FF 表示该路温度传感器不存在或损坏

注意:设备对应几路传感器即用对应几路的命令,否则超出范围有可能读不出数据。

表 2 读温度命令

- 2) 打开串口调试助手 sscom32.exe

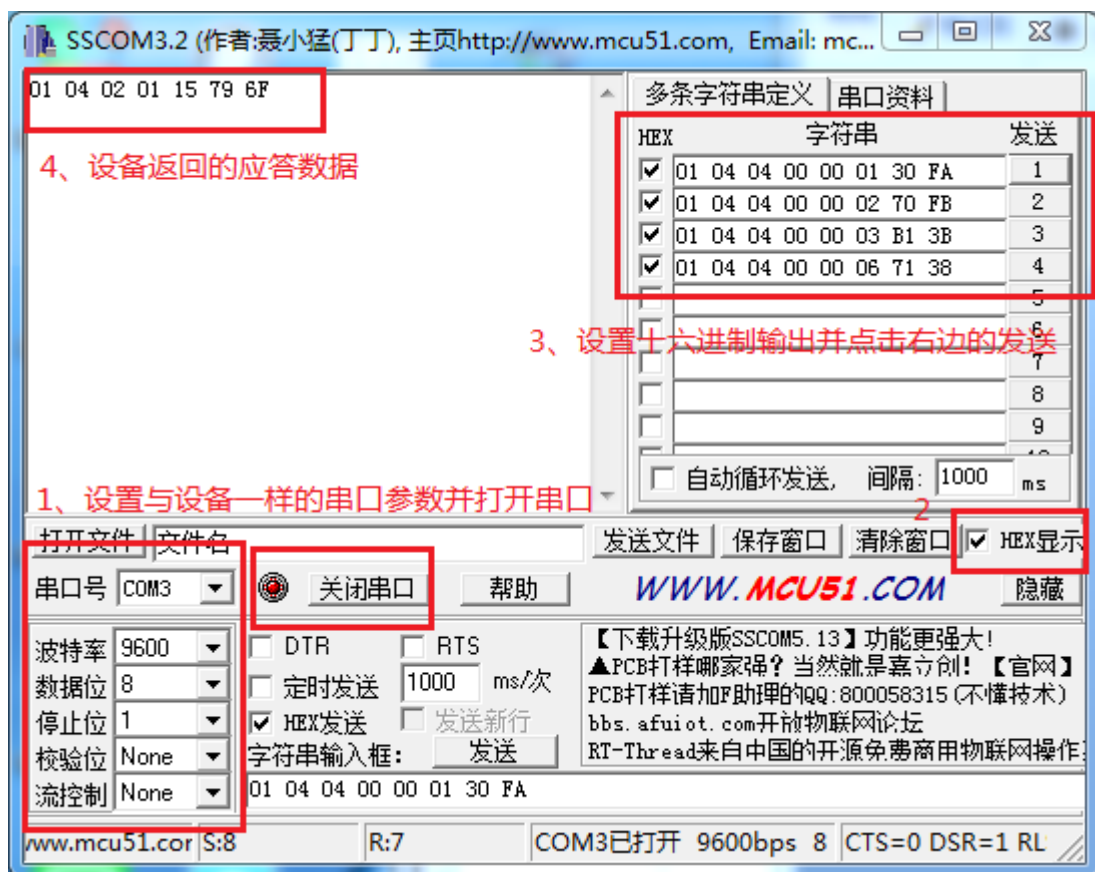


图 9 读温度值



## 2. 产品概述

### 2.1. 产品简介

USB 温度传感器 DL11-MC-Xx，是用来采集当前的环境温度数据通过 USB 传送到数据采集中心进行现场监控的设备，设备本身可以选配多路温度传感器输入，可设置波特率、数据显示精度、温度误差校正值等。

可通过软件 DL-MDM 或其它通用 MODBUS 软件进行参数设置简单方便。

### 2.2. 基本参数

直流供电（默认）	DC 3.3~5V
最大功耗	0.1W
变送器波特率	默认：9600 (1200bps~115200bps)
NTC 温度传感器	误差：±0.5℃ 范围：-40℃~200℃
DS18B20 温度传感器	误差：±0.5℃ 范围：-40℃~125℃
传感器线长	默认：1000mm (可定制)
探头材质	不锈钢 (可定制)
探头工作温度	-40℃~+200℃，默认 -40℃~+80℃
温度显示分辨率	0.1℃/1℃
温湿度刷新时间	1s
长期稳定性	≤0.1℃/y
响应时间	≤15s (1m/s 风速)
输出信号	USB (Modbus RTU 协议)
安装方式	USB 接口

## 2.3. 型号定义

DL11 USB 温湿度变送器选型表				
DL11	USB 温湿度变送器(USB 外壳)			
	代码	输出信号		
	MC	USB 转串口, 标准 MODBUS 通讯协议		
	MU	纯 USB 接口, 标准 MODBUS 通讯协议		
	LC	USB 转串口, 自定义通讯协议		
	LU	纯 USB 接口, 自定义通讯协议		
	DZ	定做		
		代码	传感器类型	
		D	数字温度传感器	
		S	温湿度传感器	
		N	模拟温度传感器	
			代码	传感器个数
			1	1 个传感器
			2	2 个传感器
			N	N 个传感器
DL11	M232	N	1	选型举例
例如: DL11-MC-N1      USB 温湿度变送器、USB 转口串口、MODBUS 通信协议、1 条 NTC 温度传感器				

### 1) DL11-MC-N1 : 1 路 NTC 温度传感器



2) DL11-MC-D1 : 1 路 DS18B20 温度传感器



3) DL11-MC-D2 : 2 路 DS18B20 温度传感器



4) DL11-MC-D3 : 3 路 DS18B20 温度传感器



5) DL11-MC-D6 : 6 路 DS18B20 温度传感器



### 3. 通信协议

#### 3.1 命令帧格式

通过 USB 接口 Modbus RTU 协议下传命令到模块，模块响应命令后回传结果；

主机（上位机）请求命令：

设备地址	功能码	寄存器地址	寄存器数量	CRC 校验
1byte	1byte	2byte	2byte	2byte

从机设备应答：

设备地址	功能码	字节数量	信息数据	CRC 校验
1byte	1byte	1byte	2*Nbyte	2byte

#### 3.2 Modbus 寄存器

寄存器名称	寄存器地址 (十六进制)	功能码 (十六进制)	寄存器说明	读写 属性
模块配置信息 0x01 区	0100H	03/06/10	模块设备地址(默认 0x01)	读/写
	0101H	03/06/10	波特率(默认 9600) 0: 9600                      1: 1200 2: 2400                      3: 4800 4: 9600                      5: 19200 6: 38400                     7: 57600 8: 115200	读/写
	0102H	03/06/10	温度显示分辨率 0:0.1°C 1:1°C	读/写
	0103H	03/06/10	刷新当前总线上的所有传感器 ID 号 写 '0x01' 时自动刷新后硬件清零(只 适于 DS18B20 温度传感器)	读/写

温度误差补偿 0x02 区	0200H	03/06/10	A01 温度传感器误差补偿值	读/写
	0201H	03/06/10	A02 温度传感器误差补偿值	读/写
	0202H	03/06/10	A03 温度传感器误差补偿值	读/写
	0203H	03/06/10	A04 温度传感器误差补偿值	读/写
	0204H	03/06/10	A05 温度传感器误差补偿值	读/写
	0205H	03/06/10	A06 温度传感器误差补偿值	读/写
温度传感器 ID 号 0x03 区 (只适于 DS18B20)	0300H	03/10	A01 温度传感器 ID 号 (8 Byte)	读/写
	0304H	03/10	A02 温度传感器 ID 号 (8 Byte)	读/写
	0308H	03/10	A03 温度传感器 ID 号 (8 Byte)	读/写
	030CH	03/10	A04 温度传感器 ID 号 (8 Byte)	读/写
	0310H	03/10	A05 温度传感器 ID 号 (8 Byte)	读/写
	0314H	03/10	A06 温度传感器 ID 号 (8 Byte)	读/写
当前温度测量 值 0x04 区	0400H	04	A01 温度传感器测量值	读
	0401H	04	A02 温度传感器测量值	读
	0402H	04	A03 温度传感器测量值	读
	0403H	04	A04 温度传感器测量值	读
	0404H	04	A05 温度传感器测量值	读
	0405H	04	A06 温度传感器测量值	读

寄存器详细介绍请参见下节。

### 3.3 Modbus 寄存器参数说明:

命令长度不能超过 250BYTE，寄存器访问不能跨区操作，不能超出本寄存器表地址，每个寄存器将数据分成两字节。

#### 1、模块配置信息 0x01 区:

**0100H:** 模块设备地址 (1-247) 默认 0x01

0100H	
High	low
00	01

高位字节无意义，读/写时总为 0；低位字节为 Modbus 设备地址

**0101H:** 波特率 (默认 9600)

0101H	
High	low
00	00

高位字节无意义，读/写时总为 0；低位字节定义如下，有效范围 0-8:

0: 9600	1: 1200	2: 2400
3: 4800	4: 9600	5: 19200
6: 38400	7: 57600	8: 115200

### 0102H: 温度显示分辨率

0102H	
High	low
00	00

0: 0.1°C

1: 1°C

### 0103H: 刷新当前总线上的所有传感器 ID 号

0103H	
High	low
00	01

写‘1’时自动刷新后硬件清零，高位字节无意义，读/写时总为0；低位字节为‘1’时会刷新一次传感器 ID 号；

## 2、温湿度误差补偿值 0x02 区:

**0200H~0205H:** A01~A6 温度传感器温度误差补偿值(补偿范围: -10.0~+10.0 度)

0200H		0201H		0202H		0203H		0204H		0205H	
High	low	High	low	High	low	High	low	High	low	High	low
00	00	00	64	FF	9C	00	41	00	01	FF	F8
A01 0.0 度		A02 10.0 度		A03 -10.0 度		A04 6.5 度		A05 0.1 度		A06 -0.8 度	

温度误差补偿值为: (高位字节\*256 + 低位字节) / 10; 负数用补码表示; 此数值修改后存于设备内部;

举例: 如果返回的值是0041H (16进制, 原码), 则高字节为00H, 低字节为41H, 那么温度补偿值为 (00H\*256+41H) / 10=6.5摄氏度。

如果返回的值是 FFF8H (16 进制, 补码), 则高字节为 FFH, 低字节为 F8H, 那么温度补偿值为 ((FFH\*256+F8H) - FFFFH - 1H) / 10 = (FFF8H - FFFFH - 1H) / 10 = -0.8 摄氏度。

也可以 Val = ~(TempOffsetValue[i]-1)/10; //负温度转成正数 即 Val = ~(0xFFF8-1)/10;

## 3、温度传感器 ID 号 0x03 区:

**0300H~0304H:** A01 温度传感器 DS18B20 唯一 ID 号 (8Byte, 高位在前, 低位在后)

0300H		0301H		0302H		0303H	
High	low	High	low	High	low	High	low
28	64	67	CF	05	00	00	D7
家族码		ID 码				CRC	

**0304H~0307H:** A02 温度传感器 DS18B20 唯一 ID 号 (8Byte, 高位在前, 低位在后)

0304H		0305H		0306H		0307H	
High	low	High	low	High	low	High	low
28	28	36	D0	05	00	00	25
家族码		ID 码				CRC	

**0308H~0317H:** A03~A6 温度传感器 ID 号 (同上);



#### 4、当前温度测量值 0x04 区：

0400H~0405H: A01~A6 温度传感器当前测量值(温度分辨率 0.1℃)

0400H		0401H		0402H		0403H		0404H		0405H	
High	low	High	low	High	low	High	low	High	low	High	low
00	00	00	FA	02	A6	03	E8	FF	97	FF	97
A01 0.0 度		A02 25.0 度		A03 67.8 度		A04 100.0 度		A05 -10.5 度		A06 -10.5 度	

温度测量值为：（高位字节\*256 + 低位字节）/10；负数用补码表示；当对应通道没有检测到温度传感器或该条温度传感器已经损坏时则用 0x7FFF 表示；

举例：如果返回的值是02A6H（16进制，原码），则高字节为02H，低字节为A6H，那么温度测量值为（02H\*256+A6H）/10=67.8摄氏度。

如果返回的值是 FF02H（16 进制，补码），则高字节为 FFH，低字节为 02H，那么温度测量值为（（FFH\*256+02H）-FFFFH-1H）/10 =（FF02H-FFFFH-1H）/10=-2.54 摄氏度。

### 3.4 Modbus 命令格式

模块支持 Modbus 的 3/4/6/16 号功能，具体功能号对应的命令格式参见以下内容，命令格式中所述的寄存器地址请参见“Modbus 寄存器”一节。

#### 4号功能：读输入寄存器

请求：

设备地址	1字节	0xHH
功能号	1字节	0x04
起始寄存器地址	2字节	0xHHHH
输入寄存器数量	2字节	0xHHHH
CRC校验	2字节	0xHHHH

响应：

设备地址	1字节	0xHH
功能号	1字节	0x04
有效字节数	1字节	2*N
数据	2*N字节	
校验	2字节	0xHHHH

注：N是要读取的寄存器数量

举例1：读温度测量值寄存器，读0x0300H-0x030BH

请求：01 04 04 00 00 0C CRCH CRCL

设备地址	1字节	0x00
功能号	1字节	0x04
起始寄存器地址	2字节	0x0400
输入寄存器数量	2字节	0x000C
CRC校验	2字节	CRCH CRCL

响应: 01 04 18 01 31 01 32 00 FF 00 FE 01 33 01 34 01 35 01 36 01 37 01 38 01 37 01 38 CRCH CRCL

设备地址	1字节	0x01	
功能号	1字节	0x04	
有效字节数	1字节	0x18	
数据	20字节	0x01（高字节）	A01温度传感器的 测量值
		0x31（低字节）	
		0x01（高字节）	A02温度传感器的 测量值
		0x32（低字节）	
		0x00（高字节）	A03温度传感器的 测量值
		0xFF（低字节）	
		0x00（高字节）	A04温度传感器的 测量值
		0xFE（低字节）	
		0x01（高字节）	A05温度传感器的 测量值
		0x33（低字节）	
		0x01（高字节）	A06温度传感器的 测量值
		0x34（低字节）	
		0x01（高字节）	A07温度传感器的 测量值
		0x35（低字节）	
		0x01（高字节）	A08温度传感器的 测量值
		0x36（低字节）	
		0x01（高字节）	A09温度传感器的 测量值
		0x37（低字节）	
		0x01（高字节）	A10温度传感器的 测量值
		0x38（低字节）	
		0x01（高字节）	A11温度传感器的 测量值
		0x37（低字节）	
		0x01（高字节）	A12温度传感器的 测量值
CRC校验	2字节	CRCH CRCL	

注: 对于测量值寄存器, 既可读单寄存器, 也可连续读多个寄存器

### 3号功能: 读保持寄存器

请求:

设备地址	1字节	0xHH
功能号	1字节	0x03
起始寄存器地址	2字节	0xHHHH
寄存器数量	2字节	0xHHHH
CRC校验	2字节	0xHHHH



响应:

设备地址	1字节	0xHH
功能号	1字节	0x03
有效字节数	1字节	2*N
数据	2*N字节	
CRC校验	2字节	0xHHHH

注: N是要读取的寄存器数量

举例1: 读模块唯一ID号, 读0x0001H-0x0002H

请求: 01 03 00 01 00 02 CRCH CRCL

设备地址	1字节	0x01
功能号	1字节	0x03
起始寄存器地址	2字节	0x0001
寄存器数量	2字节	0x0002
CRC校验	2字节	CRCH CRCL

响应: 01 03 04 00 01 02 03 CRCH CRCL

设备地址	1字节	0x01	
功能号	1字节	0x03	
有效字节数	1字节	0x04	
数据	4字节	0x00（高字节）	模块唯一ID号 0x00010203
		0x01（低字节）	
		0x02（高字节）	
		0x03（低字节）	
校验	2字节	CRCH CRCL	

举例 2: 读温度误差补偿, 读 0x0103H-0x010EH

请求: 01 03 01 03 00 0C CRCH CRCL

设备地址	1字节	0x01
功能号	1字节	0x03
起始寄存器地址	2字节	0x0103
寄存器数量	2字节	0x000C
CRC校验	2字节	CRCH CRCL

响应: 01 03 18 00 00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 0A 00 0B CRCH CRCL

设备地址	1字节	0x01	
功能号	1字节	0x03	
有效字节数	1字节	0x18	
数据	20字节	0x00（高字节）	A01温度误差补偿 值值
		0x00（低字节）	

		0x00（高字节）	A02温度误差补偿 值
		0x01（低字节）	
		0x00（高字节）	A03温度误差补偿 值
		0x02（低字节）	
		0x00（高字节）	A04温度误差补偿 值
		0x03（低字节）	
		0x00（高字节）	A05温度误差补偿 值
		0x04（低字节）	
		0x00（高字节）	A06温度误差补偿 值
		0x05（低字节）	
		0x00（高字节）	A07温度误差补偿 值
		0x06（低字节）	
		0x00（高字节）	A08温度误差补偿 值
		0x07（低字节）	
		0x00（高字节）	A09温度误差补偿 值
		0x08（低字节）	
		0x00（高字节）	A10温度误差补偿 值
		0x09（低字节）	
		0x00（高字节）	A11温度误差补偿 值
		0x0A（低字节）	
		0x00（高字节）	A12温度误差补偿 值
		0x0B（低字节）	
CRC校验	2字节	CRCH CRCL	

注：对于同一区内的寄存器，既可读单寄存器，也可连续读多个寄存器

#### 6号功能：写单个保持寄存器

请求：

设备地址	1字节	0xHH
功能号	1字节	0x06
寄存器地址	2字节	0xHHHH
数据	2字节	0xHHHH
CRC校验	2字节	0xHHHH

响应：

设备地址	1字节	0xHH
功能号	1字节	0x06
寄存器地址	2字节	0xHHHH
数据	2字节	0xHHHH
CRC校验	2字节	0xHHHH

举例 1：写模块配置信息，模块设备地址 0x0100H

请求：01 06 01 00 00 02 CRCH CRCL

设备地址	1字节	0x01
功能号	1字节	0x06
寄存器地址	2字节	0x0100
数据	2字节	0x0002
CRC校验	2字节	CRCH CRCL

响应：01 06 01 00 00 02 CRCH CRCL

设备地址	1字节	0x01
功能号	1字节	0x06
寄存器地址	2字节	0x0100
数据	2字节	0x0002
CRC校验	2字节	CRCH CRCL

#### 16号 (0x10) 功能：写多个保持寄存器

在一个远程设备中，使用该功能码写连续寄存器块(1 至约120 个寄存器)。

在请求数据域中说明了请求写入的值。每个寄存器将数据分成两字节。

正常响应返回功能码、起始地址和被写入寄存器的数量。

请求：

设备地址	1字节	0xHH
功能号	1字节	0x10
起始寄存器地址	2字节	0xHHHH
寄存器个数	2字节	0xHHHH
有效字节数	1字节	2*N
数据	2*N字节	
CRC校验	2字节	0xHHHH

注：N为寄存器个数

响应：

设备地址	1字节	0xHH
功能号	1字节	0x10
起始寄存器地址	2字节	0xHHHH
寄存器个数	2字节	0xHHHH
CRC校验	2字节	0xHHHH

举例 1：写温度误差补偿值 0x0103H-0x010EH

请求：01 10 01 03 00 0C 18 00 00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 0A  
00 0B CRCH CRCL

设备地址	1字节	0x01	
功能号	1字节	0x10	
起始寄存器地址	2字节	0x0103H	
寄存器个数	2字节	0x000CH	
有效字节数	1字节	0x18	
数据	20字节	0x00（高字节）	A01温度误差补偿 值
		0x00（低字节）	
		0x00（高字节）	A02温度误差补偿 值
		0x01（低字节）	
		0x00（高字节）	A03温度误差补偿 值
		0x02（低字节）	
		0x00（高字节）	A04温度误差补偿 值
		0x03（低字节）	
		0x00（高字节）	A05温度误差补偿 值
		0x04（低字节）	
		0x00（高字节）	A06温度误差补偿 值
		0x05（低字节）	
		0x00（高字节）	A07温度误差补偿 值
		0x06（低字节）	
		0x00（高字节）	A08温度误差补偿 值
		0x07（低字节）	
		0x00（高字节）	A09温度误差补偿 值
		0x08（低字节）	
		0x00（高字节）	A10温度误差补偿 值
		0x09（低字节）	
		0x00（高字节）	A11温度误差补偿 值
		0x0A（低字节）	
		0x00（高字节）	A12温度误差补偿 值
		0x0B（低字节）	
CRC校验	2字节	CRCH CRCL	

注：对于同一区内的寄存器，既可写单寄存器，也可连续读多个寄存器

响应：01 10 01 03 00 0C CRCH CRCL

设备地址	1字节	0x01
功能号	1字节	0x10
起始寄存器地址	2字节	0x0103
寄存器个数	2字节	0x000C
CRC校验	2字节	CRCH CRCL

**CRC16校验说明**

```
//-----  
//CRC计算C51语言函数如下  
//输入参数1: snd, 待校验的字节数组名  
//输入参数2: num, 待校验的字节总数（包括CRC校验的2个字节）  
//函数返回值: 校验失败时返回非0值。校验成功返回0。  
//-----
```

```
unsigned int calc_crc16 (unsigned char *snd, unsigned char num)  
{  
    unsigned char i, j;  
    unsigned int c, crc=0xFFFF;  
    for(i = 0; i < num; i ++)  
    {  
        c = snd[i] & 0x00FF;  
        crc ^= c;  
        for(j = 0; j < 8; j ++)  
        {  
            if (crc & 0x0001)  
            {  
                crc>>=1;  
                crc^=0xA001;  
            }  
            else  
            {  
                crc>>=1;  
            }  
        }  
    }  
    return(crc);  
}
```

校验成功后，按公式计算温度（负数以补码表示）

如果校验不成功，说明传输过程发生错误，应放弃此次采集到的数据

## 4. 联系方式

公司：广州达林电子科技有限公司

研发地址：广州市荔湾区浣花路 109 号 8085 (东鹏德宝商务中心)

工厂地址：广州市白云区永利东街 17 号三楼

网址：<http://dalintech.cn>

邮箱：[binglin@dalintech.cn](mailto:binglin@dalintech.cn)

电话：020-81522619



欢迎关注微信公众平台

## 5. 免责声明

本文档提供有关 DL11-MC-Xx 系列产品的信息，本文档未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除在其产品的销售条款和条件声明的责任之外，我公司概不承担任何其它责任。并且，我公司对本产品的销售和/或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性，适销性或对任何专利权，版权或其它知识产权的侵权责任等均不作担保。本公司可能随时对产品规格及产品描述做出修改，恕不另行通知。

## 6. 更新历史

2019-01-23 第一个版本 V1.0.0