



# E810-TTL-CAN01 产品规格书

TTL 转 CAN 智能协议转换模块



## 目录

第一章 产品概述.....	3
1.1 产品简介.....	3
1.2 特点功能.....	3
1.3 应用场景.....	3
第二章 规格参数.....	4
2.1 工作参数.....	4
2.2 出厂默认参数.....	4
第三章 机械尺寸与引脚定义.....	5
第四章 模式说明.....	6
4.1 工作模式.....	6
4.2 串口连接.....	6
4.3 数据转换方式.....	6
4.3 透明转换模式.....	7
4.3.1 串行帧转 CAN 报文.....	7
4.3.2 CAN 报文转串行帧.....	8
4.4 转换示例（透明转换）.....	8
4.4.1 串行帧转 CAN 报文.....	8
4.4.2 CAN 报文转串行帧.....	9
4.5 透传带信息模式.....	9
4.5.1 串行帧转 CAN 报文.....	9
4.5.2 CAN 报文转串行帧.....	10
4.6 转换示例（透传带信息）.....	11
4.6.1 串行帧转 CAN 报文.....	11
4.6.2 CAN 报文转串行帧.....	11
4.7 协议模式.....	12
4.8 转换示例（协议模式）.....	13
4.8.1 串行帧转 CAN 报文.....	13
4.8.2 CAN 报文转串行帧.....	14
4.9 ModBus 模式.....	15
第五章 操作指令.....	18
5.1 进入指令配置说明.....	18
5.2 指令概述.....	18
5.3 指令错误码.....	19
5.4 指令列表.....	19
5.4 指令详情.....	20
5.4.1 AT 测试指令.....	20
5.4.2 AT+CANFLT 查询/设置 CAN 过滤器信息.....	20
5.4.3 AT+CAN 查询/设置 CAN 发送参数信息.....	21
5.4.4 AT+EXAT 退出 AT 指令.....	21
5.4.5 AT+E 查询/设置指令回显模式.....	21
5.4.6 AT+MODBUSID 查询/设置模块 MODBUS 地址.....	22
5.4.7 AT+MODE 查询/设置模块工作模式.....	22

5.4.8 AT+MID 查询模块名称.....	22
5.4.9 AT+RESTORE 恢复出厂设置.....	23
5.4.10 AT+REBT 重启模块.....	23
5.4.11 AT+UARTPKT 查询/设置串口分包信息.....	23
5.4.12 AT+UART 查询/设置串口参数.....	24
5.4.13 AT+VER 查询模块版本信息.....	24
第六章 硬件设计.....	25
第七章 常见问题.....	25
7.1 模块易损坏.....	25
7.2 无法成功指令设置.....	25
7.3 参数更改后使用失败.....	25
第八章 焊接作业指导.....	26
8.1 回流焊温度.....	26
8.2 回流焊曲线图.....	26
第九章 批量包装方式（编带）.....	27
修订历史.....	27
关于我们.....	27

# 第一章 产品概述

## 1.1 产品简介

E810-TTL-CAN01是一款超高性价比的CAN-BUS开发产品，拥有强大数据分析能力的工具。同时，CAN-TTL-01型智能协议转换模块，体积小、方便安装，是工程应用、项目调试及产品开发的可靠助手。

E810-TTL-CAN01是成都亿佰特电子科技有限公司自主研发的一款小型智能协议转换模块，该模块集成了透传功能、主从一体，即拿即用。支持串口指令配置模块参数和功能，转换模式支持透明转换、透明带标识转换、协议模式转换，并支持Modbus ASCII/RTU 协议转换。

E810-TTL-CAN01模块集成1路CAN-BUS接口和1路UART TTL接口，可实现UART TTL信号与CAN-BUS之间互相透传或Modbus协议转换。CAN-TTL-01体积小，可3.3V或5V任意一种电源供电，方便集成到各种电路板中。半孔工艺，并同时带有插针焊孔，可贴片焊接，也可用插针方式插拔连接。



## 1.2 特点功能

- 数据转换方向支持UART与CAN之间双向转换；
- 转换方式支持透明转换、透明带标识转换和协议模式转换；
- 支持Modbus RTU 协议转换；
- 支持两种CAN帧数据类型发送方式：通过固定式配置或通过串行帧数据指定；
- 支持UART接口进行参数配置；
- 支持 300~921600范围内UART波特率设置
- 支持软件和硬件进入参数设置模式；
- 支持软件和硬件恢复出厂参数；
- 拥有电源指示灯，状态指示灯，模式指示灯多种指示；

## 1.3 应用场景

- 工业控制等CAN-BUS网络；
- 汽车、铁路设备联网，现场网络数据监控；
- 现有 RS-232 设备连接 CAN-bus 网络；
- 安防、消防网络；
- 地下远程通讯；
- 智能楼宇控制数据广播系统等 CAN-bus 应用系统；
- 停车场设备控制；
- 智能家居。

## 第二章 规格参数

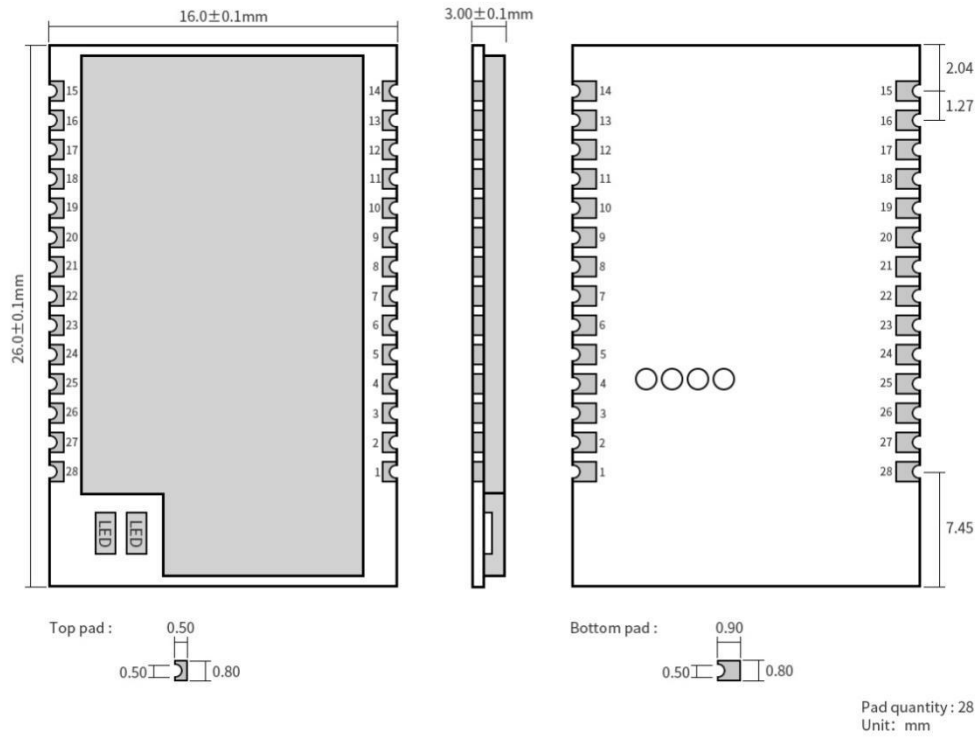
### 2.1 工作参数

主要参数	性能		备注
	最小值	最大值	
电源电压 (V)	2.3	5.5	超过 5.5V 永久烧毁模块, 推荐 5V 或者 3.3V 供电
工作电流 (mA)	19.5	70	平均电流 42mA
工作温度 (°C)	-40	+85	工业级

### 2.2 出厂默认参数

参数类型	参数名称	数值
UART	串口波特率	115200 bps
	奇偶校验	无
	数据位	8
	停止位	1
	流控	关
CAN	CAN 波特率	100kbps
	CAN ID 号	0x00000000
	流控	关
默认工作模式	透传模式	接收所有数据类型
默认设备地址	Modbus 设备地址	默认地址为 1
帧分包参数	时间	10ms
	字节数	1000byte

## 第三章 机械尺寸与引脚定义



引脚序号	引脚名称	功能	说明
1	CANH	CAN 总线高	CAN 总线高
2	CANL	CAN 总线低	CAN 总线低
3	GND	电源地	电源参考地。
4	TXD	串口发送	模块对外输出串口数据。
5	RXD	串口接收	模块接收外部串口数据。
6~12	NC		空脚
13	GND	电源地	电源参考地。
14	VCC	电源输入	电源输入引脚，推荐标准 5V 或者 3.3V 供电。
15	GND	电源地	电源参考地。
16	RESET	模块复位引脚	输入低电平模块进入硬件复位状态，输入高电平模块回复正常工作状态，该功能用于紧急情况下复位操作。
17	CFG	硬件参数配置引脚	模块参数配置时，需将其与 GND 短接。
18	Restore	恢复出厂引脚	模块参数恢复出厂时，需将其与 GND 短接 5s 以上，并在复位或者重新上电后生效。
19~20	NC		空脚
21	STE	状态指示引脚	有数据传输时，STE 引脚为低电平。
22~25	NC		空脚
26	RTS	硬件控制流引脚	硬件控制流 RTS
27	CTS	硬件控制流引脚	硬件控制流 CTS
28	GND	电源地	电源参考地。

## 第四章 模式说明

### 4.1 工作模式

E810-TTL-CAN01 模块有两种工作模式，正常模式和配置模式。

模式	功能
正常模式	模块出厂的通用模式，上电即正常工作。
配置模式	模块可以进行配置的模式。进入配置模式的方法详情见第五章：进入指令配置说明

### 4.2 串口连接

E810-TTL-CAN01 模块使用标准UART电平（3.3V），因此该模块可以直接与单片机的UART接口连接。

### 4.3 数据转换方式

E810-TTL-CAN01 支持四种数据转换方式：透明转换、透明带信息转换、协议转换和MODBUS转换；  
支持CAN与UART 双向转换。

数据转换方式	转换方向
透明转换	CAN与UART 双向转换
透明带信息转换	CAN与UART 双向转换
协议转换	CAN与UART 双向转换
MODBUS转换	CAN与UART 双向转换

### 4.3 透明转换模式

透明转换：转换器将一种格式的总线数据原样转换成另一种总线的数据格式，不附加数据也不对数据做修改。这样既实现了数据格式的交换又没有改变数据内容，对于两端的总线来说转换器如同“透明”的一样，故为透明转换。

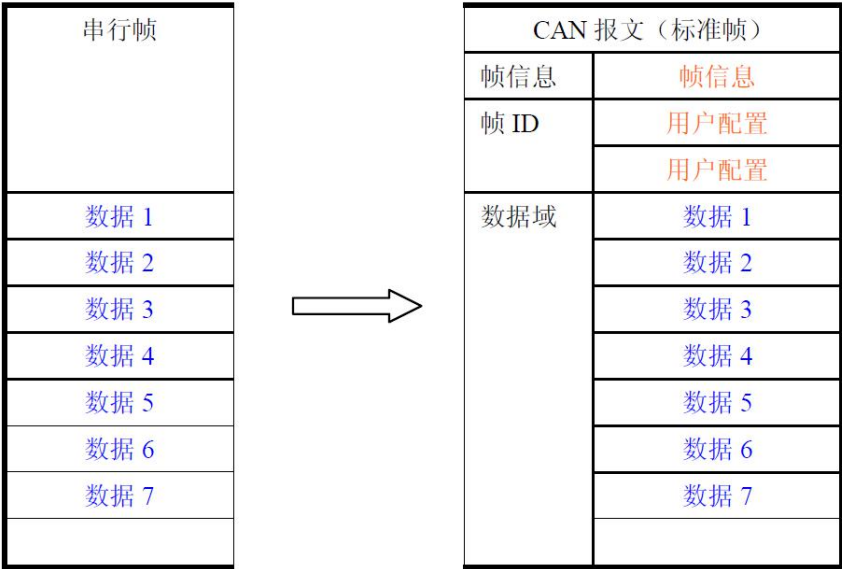
E810-TTL-CAN01 模块可以将CAN总线收到的有效数据原封不动的转换到UART，UART输出同样的数据。同样的，模块也可以将UART收到的数据原封不动的转换到CAN总线输出，实现UART与CAN的透明转换。

#### 4.3.1 串行帧转 CAN 报文

串行帧的全部数据依序填充到 CAN 报文帧的数据场里。转换器一检测到串行总线上有数据后就立即接收并转换。转换成的 CAN 报文的帧类型和帧 ID 来自用户事先的配置，并且在转换过程中帧类型和帧 ID 一直保持不变。数据转换对应格式如图所示。

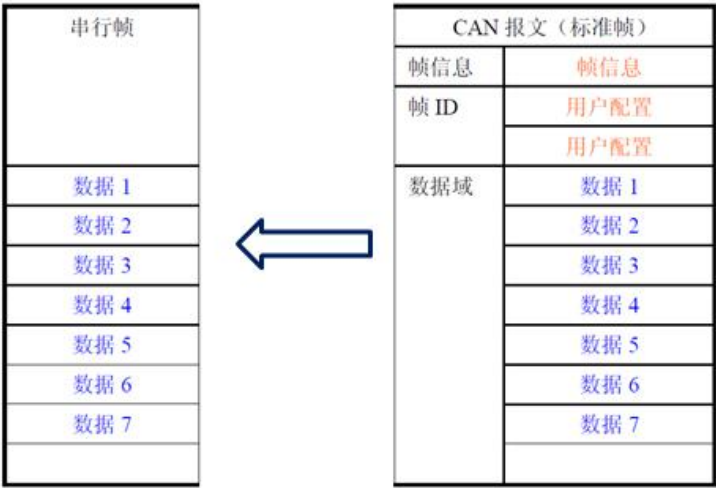
如果收到的串行帧长度小于等于 8 字节，依序将字符 1 到 n (n 为串行帧长度) 填充到 CAN 报文的数据域的 1 到 n 个字节位置（如图中 n 为 7）。

如果串行帧的字节数大于 8，那么处理器从串行帧首个字符开始，第一次取 8 个字符依次填充到 CAN 报文的数据域；将数据发至 CAN 总线后，再转换余下的串行帧数据填充到 CAN 报文的数据域，直到其数据被转换完。



4.3.2 CAN 报文转串行帧

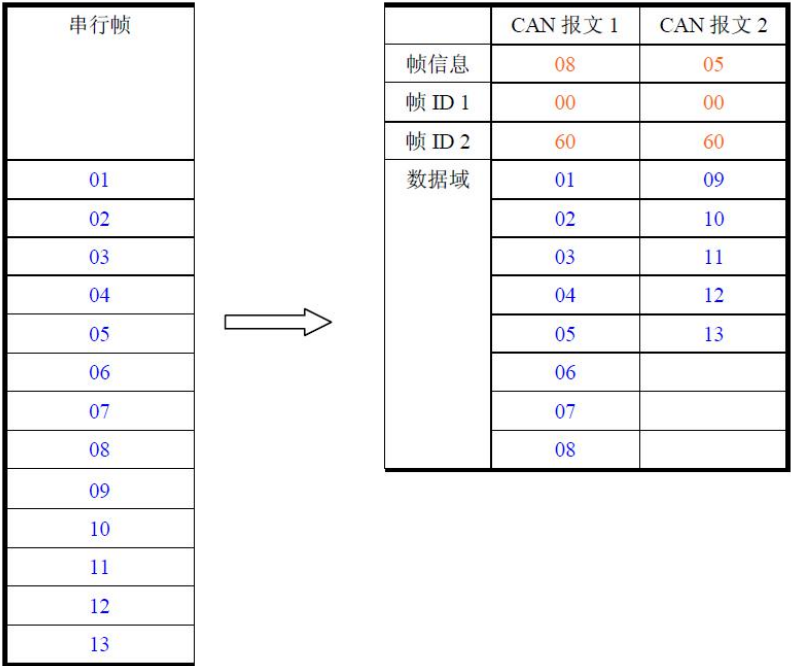
CAN报文帧的数据场里全部数据依序填充到串行帧数据里面；转换器一检测到CAN总线上有数据后就立即接收并转换。



4.4 转换示例（透明转换）

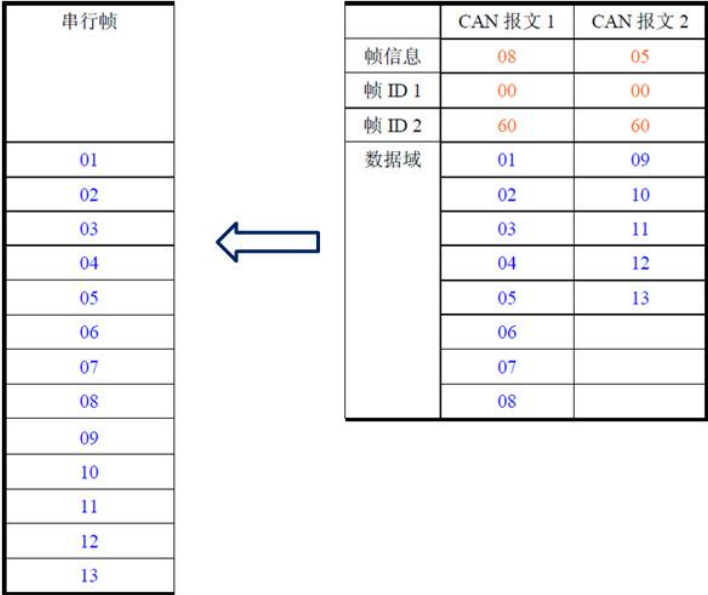
4.4.1 串行帧转 CAN 报文

假设置转换成CAN 报文帧信息为“标准帧”，帧 ID（ID1，ID2）设置为 0060，那么转换格式如图所示。



4.4.2 CAN 报文转串行帧

假设CAN 报文帧信息为“标准帧”，帧ID（ID1，ID2）设置为0060，那么转换格式如图所示。



4.5 透传带信息模式

透传带信息转换是透明转换的一种特殊的用法，也不附加协议。这种转换方式是根据通常的串行帧和 CAN 报文的共有特性，使这两种不同的总线类型也能轻松的组建同一个通信网络。

该方式下，CAN总线接收方能将收到的CAN 报文的帧信息与帧ID，添加到转换后的串行帧中，在这种方式下，接收方能够清晰看到发送方的CAN 报文信息，从而可以根据这些信息作出更灵活的运用。

4.5.1 串行帧转 CAN 报文

同透明转换，串行帧的全部数据依序填充到 CAN 报文帧的数据场里。转换器一检测到串行总线上有数据后就立即接收并转换。

转换成的 CAN 报文的帧类型和帧 ID 来自用户事先的配置，并且在转换过程中帧类型和帧 ID 一直保持不变。数据转换对应格式如图4.1 所示。

如果收到的串行帧长度小于等于 8 字节，依序将字符 1 到 n（n 为串行帧长度）填充到CAN 报文的数据域的1 到n 个字节位置（如图中n 为8）

如果串行帧的字节数大于8，那么处理器从串行帧首个字符开始，第一次取8 个字符依次填充到 CAN 报文的数据域；将数据发至 CAN 总线后，再转换余下的串行帧数据填充到 CAN 报文的数据域，直到其数据被转换完。



4.5.2 CAN 报文转串行帧

转换器一检测到CAN总线上有数据后就立即接收并转换，收到一帧CAN 报文就立即转换一帧，每次转换的时候把CAN 帧信息与帧ID添加在串行帧中（此转换与下面协议模式的CAN 报文转串行帧一致，详情请看协议模式），如下图所示。

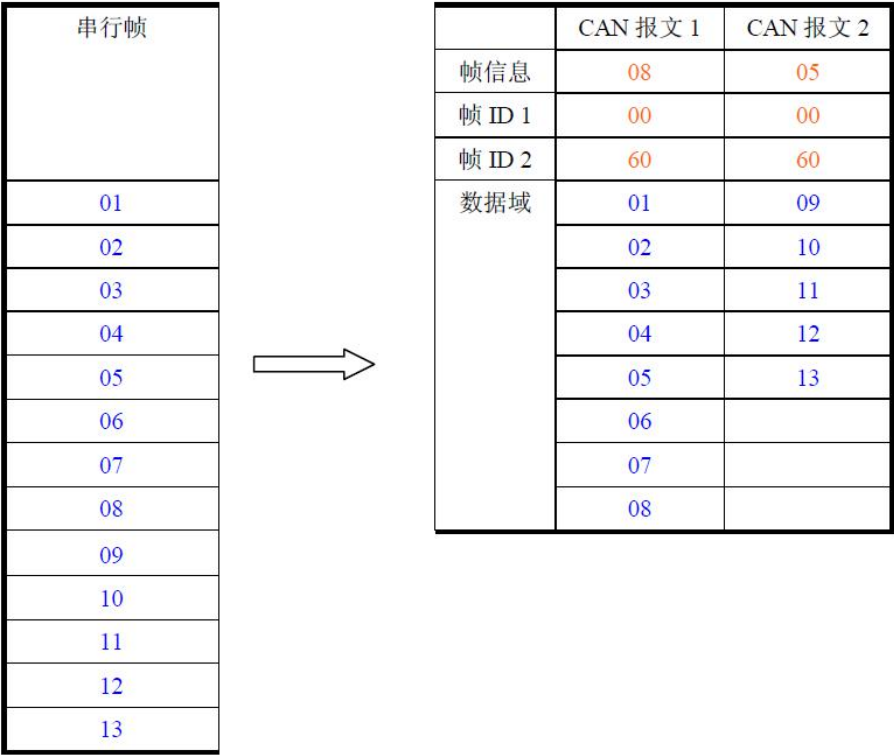
请注意：无论是串行帧还是 CAN 报文在应用的时候其帧格式（标准帧还是扩展帧）应该符合事先配置的帧格式要求，否则可能导致通讯不正常。



## 4.6 转换示例（透传带信息）

### 4.6.1 串行帧转 CAN 报文

假设配置的转换成CAN 报文的帧信息为“标准帧”，帧ID（ID1，ID2）设置为0060，那么转换格式如图所示。



### 4.6.2 CAN 报文转串行帧

CAN 发送:

帧格式: 扩展帧

帧类型: 数据帧

ID : 0x12345678

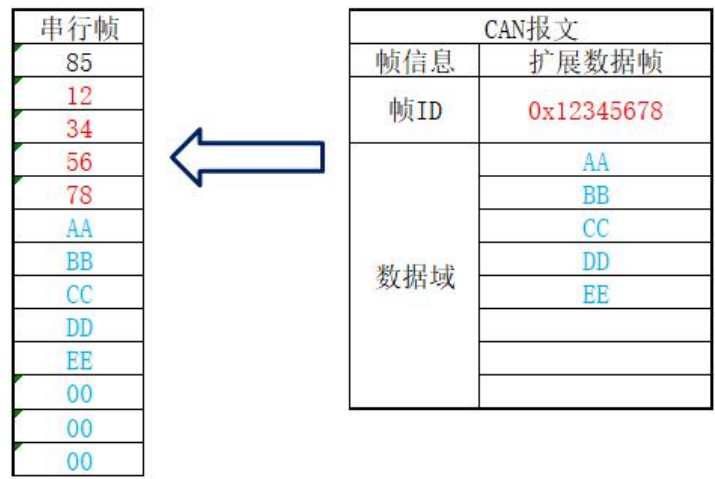
数据 : AAh BBh CCh DDh EEh

串行帧接收: 85 12 34 56 78 AA BB CC DD EE 00 00 00

0x85 表示帧格式为扩展帧，帧类型为数据帧，数据长度为 5

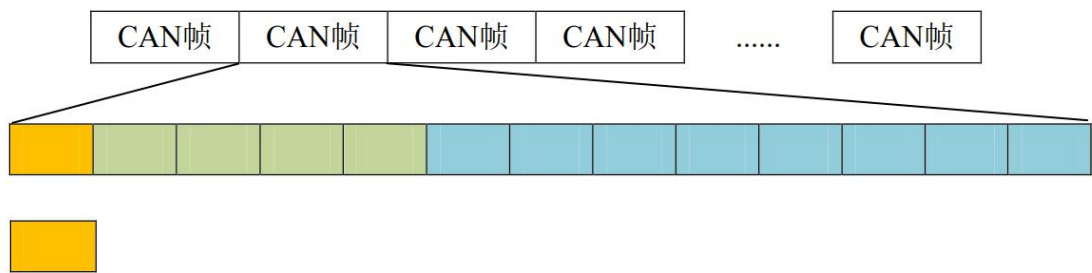
后四位表示 CAN ID 为 12345678

最后 8 位为数据区，有效长度为 5，其余位补齐 0



## 4.7 协议模式

E810-TTL-CAN01模块数据转换格式如下所示，每一个CAN帧包含13个字节，13个字节的内容包括CAN帧信息+帧ID+帧数据。



**帧信息：**长度 1 个字节，用于标识 CAN 帧的一些信息，如类型、长度等。



- FF：**标准帧和扩展帧的标识位，1 为扩展帧，0 为标准帧。
- RTR：**远程帧和数据帧的标识位，1 为远程帧，0 为数据帧。
- 保留：**保留值为 0，不可写入 1。
- D3～D0：**数据长度位，标识该 CAN 帧的数据长度。

**帧 ID:** 长度 4 个字节，标准帧有效位 11 位，扩展帧有效位 29 位。

高字节				低字节			
12h	34h	56h	78h	00h	00h	01h	23h

如上为扩展帧 ID 号

0x12345678 的表示方式

如上为标准帧 ID 号

0x123 的表示方式

**帧数据:** 长度 8 个字节，有效长度由帧信息的 D3~D0 的值决定。

DATA1					DATA8		
11h	22h	33h	44h	55h	66h	77h	88h

如上为 8 个字节有效数据的表示方式。

DATA1					DATA8		
11h	22h	33h	44h	55h	00h	00h	00h

如上为 5 个字节有效数据的表示方式。

#### 举例说明:

以下例子是一个扩展数据帧，帧 ID 为 0x12345678，包含 8 个字节有效数据（11h,22h,33h,44h,55h,66h,77h,88h）的表示方式。

88h	12h	34h	56h	78h	11h	22h	33h	44h	55h	66h	77h	88h
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

以下例子是一个标准数据帧，帧 ID 为 0x123，包含 5 个字节有效数据（11h,22h,33h,44h,55h）的表示方式。

05h	00h	00h	01h	23h	11h	22h	33h	44h	55h	00h	00h	00h
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**请注意:** 每一帧固定是 13 个字节，不足的必须补 0，否则将导致通信错误。

## 4.8 转换示例（协议模式）

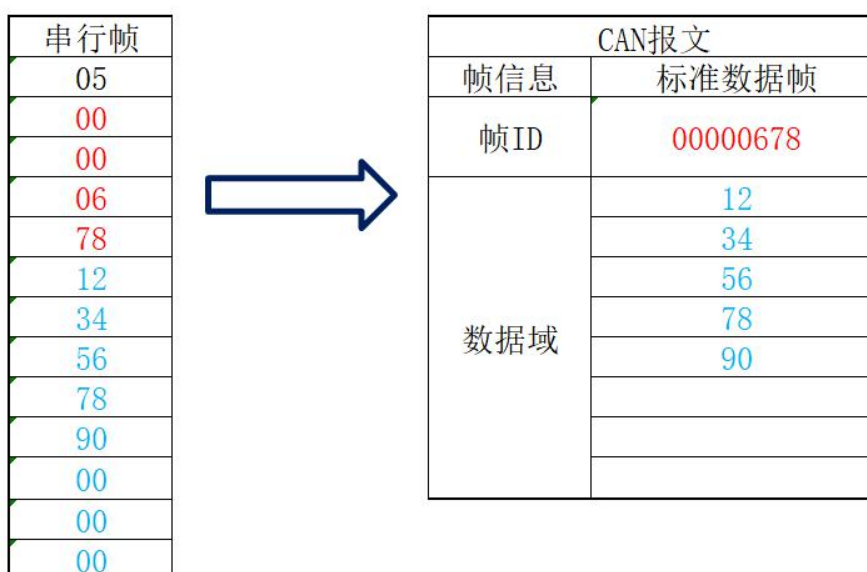
### 4.8.1 串行帧转 CAN 报文

串行帧发送: 05 00 00 06 78 12 34 56 78 90 00 00 00

0x05 表示帧格式为标准帧，帧类型为数据帧，数据长度为 5

00 00 06 78 表示 ID 为 0678

12 34 56 78 90 00 00 00 为数据区，有效长度为 5，如下图所示：



#### 4.8.2 CAN 报文转串行帧

CAN 发送:

帧格式: 扩展帧

帧类型: 数据帧

ID : 0x12345678

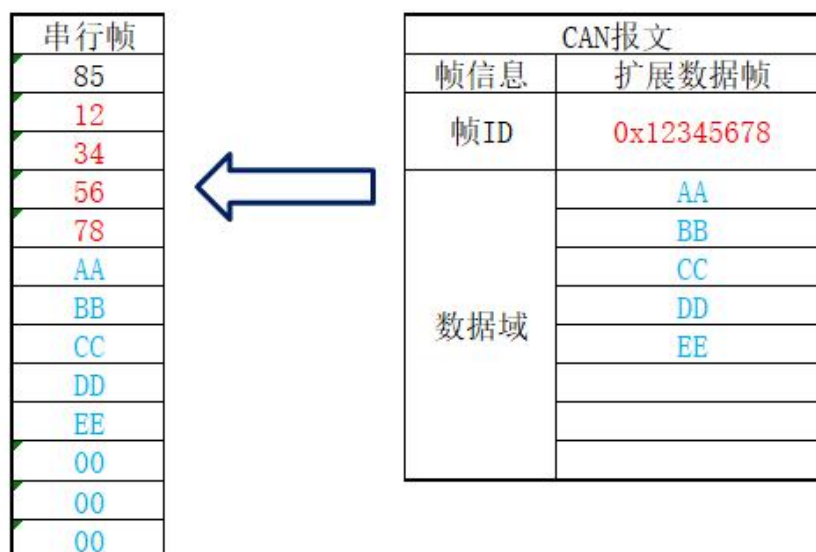
数据 : AAh BBh CCh DDh EEh

串行帧接收: 85 12 34 56 78 AA BB CC DD EE 00 00 00

0x85 表示帧格式为扩展帧, 帧类型为数据帧, 数据长度为 5

后四位表示 CAN ID 为 12345678

最后 8 位为数据区, 有效长度为 5, 其余位补齐 0, 如下图所示:



## 4.9 ModBus 模式

Modbus 转换模式，支持 RTU转换模式。E810-TTL-CAN01模块是作为从设备来使用的，接收并响应主机（通过UART）发送过来的命令。

E810-TTL-CAN01 转换模块支持两种 Modbus 命令：读取寄存器（功能码 03）、写多个寄存器（功能码 16）。

转换模块内部有一缓存器用于缓存接收到的 CAN 帧数据，缓存器按照地址 0~63 总共有 64 级缓存。缓存地址从 0 开始，到地址 63，可连续缓存8帧CAN 数据(每帧数据8字节，共64字节)。当接收到第一帧 CAN 数据时，该 CAN 帧数据存储的地址0中，以后接收到的 CAN 帧数据按照地址逐级递增顺序存放。如果 64 级缓存都存储满了，则新接收到的 CAN 帧数据将存放到地址0中并覆盖原来的数，遵循先进先出。

读取寄存器（功能码 03）：

发送命令：

[设备地址] [命令号03(0x03)] [起始寄存器地址高8位] [低8位] [读取的寄存器数高8位] [低8位]

[CRC校验的高8位] [CRC校验的低8位]

读取格式只允许是从地址00 00 开始 读取 00 08 （一次性读取8个字节即一帧数据，即读取了00 00 - 00 07 地址的数据），读取成功后，该8个字节数据会被清空，它地址之后的数据会往前移8个数据。

比如：

模块在MODBUS模式下，CAN总线收到了4帧数据：

第一帧： 0x01 0x02 0x03 0x04 共4字节数据

第二帧： 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 共6字节数据

第三帧： 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 共8字节数据

第四帧： 0xAA 0xBB 0xCC 共3字节数据

它们在MODBUS缓存地址中是存放的情况：

（没有数据的地址为0x00）

缓存地址	数据	缓存地址	数据
0x31	0x00	0x63	0x00
0x30	0x00	0x62	0x00
0x29	0x00	0x61	0x00
0x28	0x00	0x60	0x00
0x27	0x00	0x59	0x00
0x26	0xCC	0x58	0x00
0x25	0xBB	0x57	0x00
0x24	0xAA	0x56	0x00
0x23	0x88	0x55	0x00
0x22	0x77	0x54	0x00
0x21	0x66	0x53	0x00
0x20	0x55	0x52	0x00
0x19	0x44	0x51	0x00
0x18	0x33	0x50	0x00
0x17	0x22	0x49	0x00
0x16	0x11	0x48	0x00
0x15	0x00	0x47	0x00
0x14	0x00	0x46	0x00
0x13	0x0F	0x45	0x00
0x12	0x0E	0x44	0x00
0x11	0x0D	0x43	0x00
0x10	0x0C	0x42	0x00
0x09	0x0B	0x41	0x00
0x08	0x0A	0x40	0x00
0x07	0x00	0x39	0x00
0x06	0x00	0x38	0x00
0x05	0x00	0x37	0x00
0x04	0x00	0x36	0x00
0x03	0x04	0x35	0x00
0x02	0x03	0x34	0x00
0x01	0x02	0x33	0x00
0x00	0x01	0x32	0x00

用指令：01 03 00 00 00 08（在地址为01的从设备中，读取从地址0000开始的8条数据）将此指令写到专用工具（Modbus CRC 16计算器），计算出CRC校验值并添加到指令后面。通过UART发出该指令：01 03 00 00 00 08 44 0C，从机接收到该指令后，会返回从机转换模块内部的缓存值。（当没有接收到新的 CAN 帧数据时，主机读到的缓存值则全为 0）

返回指令：

[设备地址] [命令号03] [返回的字节个数] [数据1] [数据2]...[数据n] [CRC校验的高8位] [CRC校验的低8位]

如：

从机收到指令01 03 00 00 00 08 44 0C，，则返回：

01 03 10 00 01 00 02 00 03 00 04 00 00 00 00 00 00 00 00 00 1F 9F

读取完成后，该8个字节数据被清空，它地址之后的数据往前移8个数据。如下图所示：

缓存地址	数据	缓存地址	数据
0x31	0x00	0x63	0x00
0x30	0x00	0x62	0x00
0x29	0x00	0x61	0x00
0x28	0x00	0x60	0x00
0x27	0x00	0x59	0x00
0x26	0x00	0x58	0x00
0x25	0x00	0x57	0x00
0x24	0x00	0x56	0x00
0x23	0x00	0x55	0x00
0x22	0x00	0x54	0x00
0x21	0x00	0x53	0x00
0x20	0x00	0x52	0x00
0x19	0x00	0x51	0x00
0x18	0xCC	0x50	0x00
0x17	0xEB	0x49	0x00
0x16	0xAA	0x48	0x00
0x15	0x88	0x47	0x00
0x14	0x77	0x46	0x00
0x13	0x66	0x45	0x00
0x12	0x55	0x44	0x00
0x11	0x44	0x43	0x00
0x10	0x33	0x42	0x00
0x09	0x22	0x41	0x00
0x08	0x11	0x40	0x00
0x07	0x00	0x39	0x00
0x06	0x00	0x38	0x00
0x05	0x0F	0x37	0x00
0x04	0x0E	0x36	0x00
0x03	0x0D	0x35	0x00
0x02	0x0C	0x34	0x00
0x01	0x0B	0x33	0x00
0x00	0x0A	0x32	0x00

由此可进行下一次读取寄存器命令

写多个寄存器（功能码 16）：成功写入时，会将写入的数据发到CAN总线。

发送命令：

[设备地址] [命令号16 (0x10)] [需下置的寄存器地址高8位] [低8位] [数据数量高8位] [数据数量低8位] [下置的数据高8位] [低8位] [……] [……] [CRC校验的高8位] [CRC校验的低8位]

如：

通过写命令写5字节数据发送出去： 01 10 00 00 00 05 0A 00 11 00 22 00 33 00 44 00 55，将此指令写到专用工具（Modbus CRC 16计算器），计算出CRC校验值并添加到指令后面：01 10 00 00 00 05 0A 00 11 00 22 00 33 00 44 00 55 47 84，以HEX格式发送，即可写入5字节数据，成功写入时，会将写入的数据发到CAN总线，即此时CAN总线会发送5个字节的数据：11, 22, 33, 44, 55。CAN报文中，帧信息，帧ID为用户事先配置好的。

注意：此命令的寄存器地址只允许是 00 00，不允许发如： 01 10 00 01 00 05 0A 00 11 00 22 00 33 00 44 00 55 这样的指令格式。

且最多一次性发8字节数据：

通过写命令写8字节数据发送出去： 01 10 00 00 00 08 10 00 11 00 22 00 33 00 44 00 55 00 66 00 77 00 88，将此指令写到专用工具（Modbus CRC 16计算器），计算出CRC校验值并添加到指令后面：01 10 00 00 00 08 10 00 11 00 22 00 33 00 44 00 55 00 66 00 77 00 88 FE D5，以HEX格式发送，即可写入8字节数据，成功写入时，会将写入的数据发到CAN总线，即此时CAN总线会发送8个字节的数据：

0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88。CAN报文中，帧信息，帧ID为用户事先配置好的。

不允许发如： 01 10 00 00 00 09 12 00 11 00 22 00 33 00 44 00 55 00 66 00 77 00 88 00 99 这样的指令格式。

## 第五章 操作指令

### 5.1 进入指令配置说明

注：一般工作模式下模式指示灯以1Hz频率闪烁；当进入指令配置模式时，指示灯以5Hz频率闪烁，无论是软件进入还是硬件进入，配置的参数皆是在复位后生效。

进入指令配置模式有以下两种方式：

①从其他模式切换至指令模式的时序（软件进入）：

串口设备给模块连续发送”+++”，模块收到“+++”后，3秒计时超时开始启动，如果在超时时间内接收到任意的AT指令则成功切换到配置模式(进入后，模块指示灯闪烁频率变为 5Hz)。

此时可以进行参数的配置了，参数列表及参数说明请看下面介绍。

参数配置成功后，串口设备给模块发送指令”AT+ EXAT”，模块在接收到指令后，返回” +OK”同时退出配置模式回到配置前的模式。然后短接复位RESET引脚，参数生效。也可以在用AT指令参数配置成功后，用指令“AT+REBT”复位模块

②从其他模式切换至指令模式的时序（硬件进入）：

将CFG引脚与 GND 短接，可直接进入配置模式(进入后，模块指示灯闪烁频率变为 5Hz，配置过程中，CFG引脚需一直保持低电平)，直接发送AT指令进行配置， 断开短接后，短接复位RESET引脚，参数生效。

备注：硬件进入指令模式与退出指令模式的优先级都比软件指令进入指令模式的优先级高。

如：先用软件指令进入指令模式，再将CFG引脚与GND短接，由于硬件优先级高，所以现在被视为硬件进入指令模式，当CFG引脚变高电平后，直接是硬件退出指令模式。

### 5.2 指令概述

AT指令是指，在命令模式下用户通过UART与模块进行命令传递的指令集，后面将详细讲解AT指令的使用格式。上电成功后切换到配置模式，可以通过UART对模块进行设置。

从透传模式切换至指令模式的时序：

串口设备给模块连续发送”+++”，模块收到“+++”后，3秒计时超时开始启动，如果在超时时间内接收到任意的AT指令则成功切换到配置模式

最后，串口设备给模块发送指令”AT+ EXAT”，模块在接收到指令后，返回” +OK”同时退出配置。然后短接复位RESET引脚，参数生效。也可以在用AT指令参数配置成功后，用指令“AT+REBT”复位模块。

注：配置参数后，一定是复位后参数才生效的。

说明： <CR>: ASCII 码 0x0D

<LF>: ASCII 码 0x0A

### 5.3 指令错误码

错误码	说明
-1	无效的命令格式
-2	无效的命令
-3	无效的操作符
-4	无效的参数
-5	操作不允许

### 5.4 指令列表

AT	测试指令
CANFLT	查询/设置CAN过滤器信息
CAN	查询/设置CAN发送参数信息
EXAT	退出AT指令
E	查询/设置AT指令回显
MODBUSID	查询/设置模块MODBUS地址
MODE	查询/设置模块工作模式
MID	查询模块设备信息
RESTORE	模块参数恢复出厂设置指令
REBT	模块复位指令
UARTPKT	查询/设置模块串口分包参数
UART	查询/设置模块串口参数
VER	查询模块版本信息

## 5.4 指令详情

### 5.4.1 AT 测试指令

**功能：**测试指令

**格式：**设置

**发送：**AT<CR>

**返回：**<CR><LF>+OK<CR><LF>

**示例：**AT<CR>

### 5.4.2 AT+CANFLT 查询/设置 CAN 过滤器信息

**功能：**查询/设置CAN过滤器信息

**格式：**查询

**发送：**AT+CANFLT<CR>

**返回：**<CR><LF>+OK=<id,mask id,mode><CR><LF>

**设置**

**发送：**AT+ CANFLT =<id,mask id,mode><CR>

**返回：**<CR><LF>+OK<CR><LF>

**参数：**

**id：** 过滤id,与过滤mask id配合使用，该参数只在用户自定义过滤模式生效

**mask id：** 过滤mask id, 与 id配合使用，该参数只在用户自定义模式过滤生效

**mode：** 过滤模式 共8种模式： OFF 接收所有类型的帧

EDTF 只接收扩展数据帧

ERTF 只接收扩展远程帧

NRTF 只接收标准远程帧

NDTF 只接收标准数据帧

ETF 只接收扩展帧

NTF 只接收标准帧

USRF 用户自定义过滤模式（即标准CAN标识符屏蔽位模式，请参考CAN相关资料查阅使用，需自行设置标识符匹配值（即参数中的id），与屏蔽码（即参数中的mask id））

**示例：**AT+CANFLT=0,0,OFF <CR>

### 5.4.3 AT+CAN 查询/设置 CAN 发送参数信息

**功能：** 查询/设置CAN发送参数信息。

**格式：** 查询

**发送：** AT+CAN<CR>

**返回：** <CR><LF>+OK=<baud, id, mode><CR><LF>

**设置**

**发送：** AT+ CAN

**返回：** <CR><LF>+OK<CR><LF>

**参数：**

**baud：** CAN波特率 单位： kbps 范围： 6, 10, 20, 50, 100, 120, 125, 150, 200, 250, 400, 500, 600, 750, 1000

**id：** 发送帧ID（标识符） NDTF模式： 0-7FF EDTF模式： 0-1FFFFFFF

**mode：** 发送模式 共2种模式： NDTF 发送标准数据帧

EDTF 发送扩展数据帧

**示例：** AT+CAN=100, 70, NDTF <CR>

### 5.4.4 AT+EXAT 退出 AT 指令

**功能：** 退出AT指令

**格式：** 设置

**发送：** AT+EXAT<CR>

**返回：** <CR><LF>+OK<CR><LF>

**示例：** AT+EXAT<CR>

### 5.4.5 AT+E 查询/设置指令回显模式

**功能：** 查询/设置指令回显模式。

**格式：** 查询

**发送：** AT+E<CR>

**返回：** <CR><LF>+OK=<sw><CR><LF>

**设置**

**发送：** AT+E=<sw><CR>

**返回：** <CR><LF>+OK<CR><LF>

**参数：**

**sw** AT指令回显开关 共两种状态：

ON 打开回显，回显AT指令下输入的指令

OFF 关闭回显，AT指令模式下，输入指令不回显

**示例：** AT+E=ON<CR>

#### 5.4.6 AT+MODBUSID 查询/设置模块 MODBUS 地址

**功能：**查询/设置模块MODBUS地址。

**格式：**查询

**发送：**AT+ AT+MODBUSID <CR>

**返回：**<CR><LF>+OK=<id><CR><LF>

**设置**

**发送：**AT+ AT+MODBUSID =<id><CR>

**返回：**<CR><LF>+OK<CR><LF>

**参数：**

id: MODBUS ID 范围：0-255

**示例：**AT+MODBUSID=9 <CR>

#### 5.4.7 AT+MODE 查询/设置模块工作模式

**功能：**查询/设置模块工作模式。

**格式：**查询

**发送：**AT+MODE <CR>

**返回：**<CR><LF>+OK=<mode><CR><LF>

**设置**

**发送：**AT+MODE =<mode><CR>

**返回：**<CR><LF>+OK<CR><LF>

**参数：**

Mode: 模块工作模式 共4种模式：

TRANS 透明传输模式

TPRTL 透明带信息模式

MODBUS MODBUS模式

PROTOL 协议模式

**示例：**AT+MODE=TRANS <CR>

#### 5.4.8 AT+MID 查询模块名称

**功能：**查询模块名称。

**格式：**查询

**发送：**AT+MID<CR>

**返回：**<CR><LF>+OK=<name><CR><LF>

**参数：**name 模块名称

**示例：**AT+MID=E810-TTL<CR>

#### 5.4.9 AT+RESTORE 恢复出厂设置

**功能：**恢复出厂设置

**格式：**设置

**发送：**AT+RESTORE<CR>

**返回：**<CR><LF>+OK<CR><LF>

**参数：**无

**示例：**AT+ RESTORE <CR>

**〈注意〉：**该命令正确执行后，请使用AT+REBT指令重启模块。

#### 5.4.10 AT+REBT 重启模块

**功能：**重启模块

**格式：**设置

**发送：**AT+REBT<CR>

**返回：**<CR><LF>+OK<CR><LF>

**参数：**无

**示例：**AT+ REBT <CR>

**〈注意〉：**该命令正确执行后，模块重新启动，将退出 AT 模式。

#### 5.4.11 AT+UARTPKT 查询/设置串口分包信息

**功能：**查询/设置串口分包信息。

**格式：**查询

**发送：**AT+UARTPKT<CR>

**返回：**<CR><LF>+OK=<size,time><CR><LF>

**设置**

**发送：**AT+ UARTPKT=<size,time><CR>

**返回：**<CR><LF>+OK<CR><LF>

**参数：**

**size** 打包长度，0，4-1000 字节，0为关闭，两个都为0则是默认打包参数

**time** 打包时间，0，10~1000 毫秒，0为关闭，两个都为0则是默认打包参数

**示例：**AT+UARTPKT=1000,10<CR>

#### 5.4.12 AT+UART 查询/设置串口参数

**功能：** 查询/设置串口参数。

**格式：** 查询

**发送：** AT+UART<CR>

**返回：** <CR><LF>+OK=<baud, data, stop, parity, flowctrl><CR><LF>

**设置**

**发送：** AT+UART=<baud, data, stop, parity, flowctrl><CR>

**返回：** <CR><LF>+OK<CR><LF>

**参数：**

baud 波特率，支持300-921600任意波特率

data 数据位8位

stop 停止位1、2位

parity 校验位ODD（奇校验）、EVEN（偶校验）、NONE（无校验）

flowctrl 流控位NFC（无硬件流控）、FC（有硬件流控）

**示例：** AT+UART=9600, 8, 1, NONE, NFC<CR>

#### 5.4.13 AT+VER 查询模块版本信息

**功能：** 查询模块版本信息

**格式：** 查询

**发送：** AT+VER<CR>

**返回：** <CR><LF>+OK=<ver><CR><LF>

**参数：**

ver 模块版本号

**示例：** AT+VER<CR>

## 第六章 硬件设计

- 推荐使用直流稳压电源对该模块进行供电，电源纹波系数尽量小，模块需可靠接地；
- 请注意电源正负极的正确连接，如反接可能会导致模块永久性损坏；
- 请检查供电电源，确保在推荐供电电压之间，如超过最大值会造成模块永久性损坏；
- 请检查电源稳定性，电压不能大幅频繁波动；
- 在针对模块设计供电电路时，往往推荐保留30%以上余量，有整机利于长期稳定地工作；
- 高频数字走线、高频模拟走线、电源走线必须避开模块下方，若实在不得已需要经过模块下方，假设模块焊接在Top Layer，在模块接触部分的Top Layer 铺地铜（全部铺铜并良好接地），必须靠近模块数字部分并走线在Bottom Layer；
- 假设模块焊接或放置在 Top Layer，在 Bottom Layer 或者其他层随意走线也是错误的，会在不同程度影响模块的杂散以及接收灵敏度；
- 假设模块周围有存在较大电磁干扰的器件也会极大影响模块的性能，跟据干扰的强度建议适当远离模块，若情况允许可以做适当的隔离与屏蔽；
- 假设模块周围有存在较大电磁干扰的走线（高频数字、高频模拟、电源走线）也会极大影响模块的性能，跟据干扰的强度建议适当远离模块，若情况允许可以做适当的隔离与屏蔽；

## 第七章 常见问题

### 7.1 模块易损坏

- 请检查供电电源，确保在推荐供电电压之间，如超过最大值会造成模块永久性损坏。
- 请检查电源稳定性，电压不能大幅频繁波动。
- 请确保安装使用过程防静电操作。
- 请确保安装使用过程湿度不宜过高，部分元件为湿度敏感器件。
- 如果没有特殊需求不建议在过高、过低温度下使用。

### 7.2 无法成功指令设置

- 请检查指令格式是否正确。
- 请检查指示灯闪烁频率，指令配置状态下，指示灯闪烁频率为5Hz。
- 请检查指令所使用的模式。

### 7.3 参数更改后使用失败

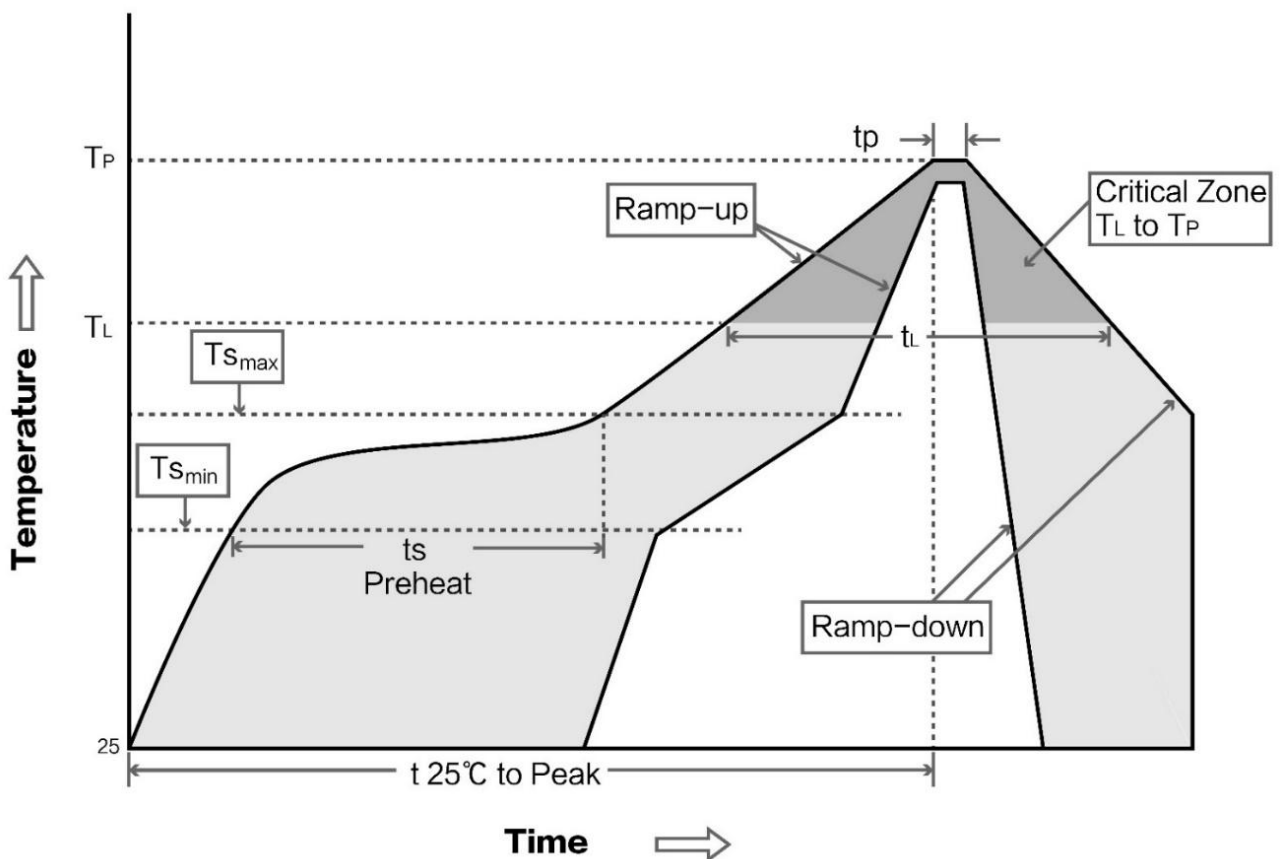
- 请检查串口波特率和CAN波特率是否设置正确。
- 请检查数据格式是否符合规定。
- 请检查电脑串口是否死机。
- 请仔细阅读指令使用说明。

## 第八章 焊接作业指导

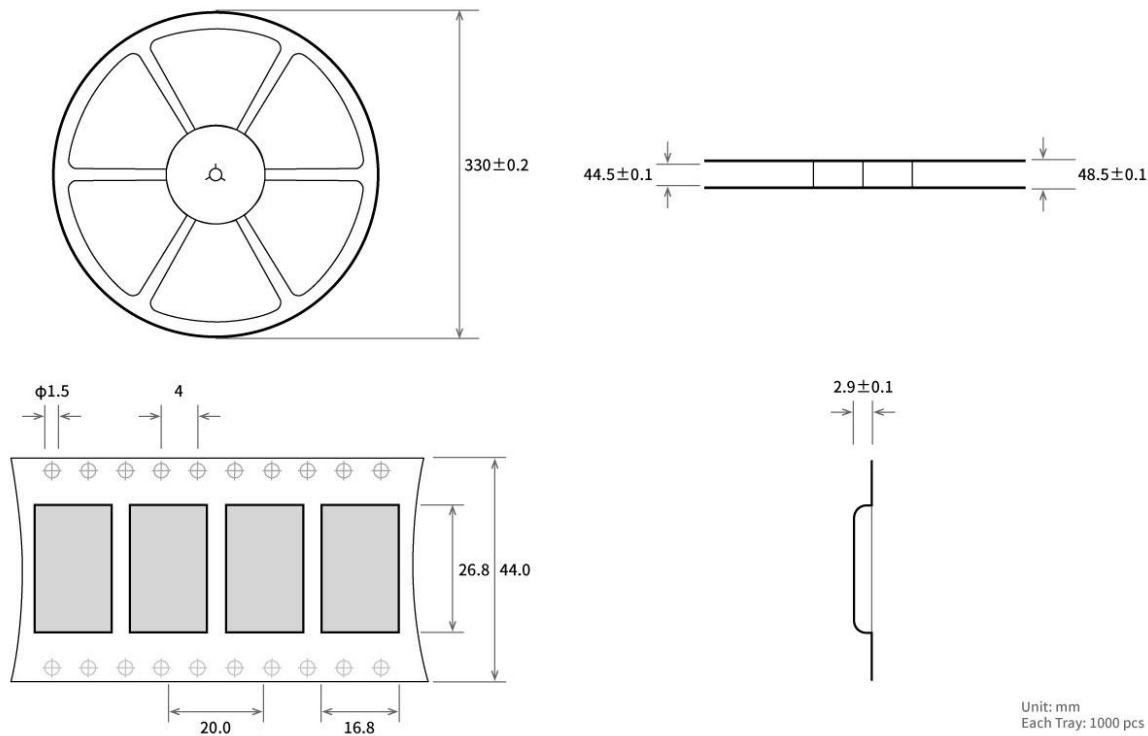
### 8.1 回流焊温度

Profile Feature	曲线特征	Sn-Pb Assembly	Pb-Free Assembly
Solder Paste	锡膏	Sn63/Pb37	Sn96.5/Ag3/Cu0.5
Preheat Temperature min (T <sub>smin</sub> )	最小预热温度	100°C	150°C
Preheat temperature max (T <sub>smax</sub> )	最大预热温度	150°C	200°C
Preheat Time (T <sub>smin</sub> to T <sub>smax</sub> )(t <sub>s</sub> )	预热时间	60-120 sec	60-120 sec
Average ramp-up rate(T <sub>smax</sub> to T <sub>p</sub> )	平均上升速率	3°C/second max	3°C/second max
Liquidous Temperature (T <sub>L</sub> )	液相温度	183°C	217°C
Time (t <sub>L</sub> ) Maintained Above (T <sub>L</sub> )	液相线以上的时间	60-90 sec	30-90 sec
Peak temperature (T <sub>p</sub> )	峰值温度	220-235°C	230-250°C
Aveage ramp-down rate (T <sub>p</sub> to T <sub>smax</sub> )	平均下降速率	6°C/second max	6°C/second max
Time 25°C to peak temperature	25°C到峰值温度的时间	6 minutes max	8 minutes max

### 8.2 回流焊曲线图



# 第九章 批量包装方式（编带）



# 修订历史

版本	修订日期	修订说明	维护人
1.0	2019-3-12	初始版本	Ray
1.1	2019-7-15	添加内容	Blue

# 关于我们



销售热线：4000-330-990

技术支持：[support@cdebyte.com](mailto:support@cdebyte.com)

公司地址：四川省成都市高新西区西芯大道 4 号创新中心 B333-D347

公司电话：028-61399028

官方网站：[www.ebyte.com](http://www.ebyte.com)


**成都亿佰特电子科技有限公司**  
 Chengdu Ebyte Electronic Technology Co.,Ltd.