TTCANopen

CAN(FD)转USB

使用说明书(v4.0)

TTCANopen 工作室

目　录

产品升级说明

1）自2021年1月18日，本产品可由v3.1，升级到v3.2

本次升级主要改善本说明书中第8节“使用指令配置模块”的便捷性，升级后可能在使用指令配置模块时与v3.1略有不兼容，其它内容没有改变，如果您对此不敢兴趣，可以不进行升级。另外，升级后模块添加了CAN(FD)转TTL串口的功能。

2）自2021年8月16日，本产品升级到v3.3

本次升级新增了采样点的选择项。分别是75%、80%和85%左右。用户可以根据情况进行更合理的配置。

3）自2022年1月16日，本产品升级到v3.5

本次升级，应用户要求提供了UID显示，具体当模块CAN速率选择为10K时，帮助界面会有96位的UID显示。

进一步完善了对LINUX的支持。

4）自2024年8月1日，本产品升级到v4.0

本次升级，主要新增了四项功能。分别是新增UID读取指令；模块时间配置指令；CANFD数据域不加速发送；6个ID选择过滤功能。具体描述详见附录3

1. 产品介绍

本品支持CAN和CANFD应用，可做为CAN(FD)转USB、CAN(FD)转TTL模块使用，模块状态切换自如，可更好的适用用户的应用场景。

CAN(FD)转USB模块主要用于将CAN或CANFD网络接口通过USB端口转换使其能够实现“近现场”接入上位监控和管理系统，如：工业现场数据监控等。

其广泛应用于汽车电子，工业自动化监控，医疗仪器，机器人，纺织机械，安防系统，水文气象等领域的数据传输和监控。

CAN(FD)转TTL主要用于将CAN或CANFD网络接口通过TTL串口转换使其能够实现“微现场”接入到其它不具备CAN总线能力的模块中，由于TTL串口的通讯能力通常会低于CANFD，因此其一般只应用于较低信息密度的CANFD或CAN现场总线。

（本说明书以介绍CAN(FD)转USB为主线，在适当的地方插入CAN(FD)转TTL的相关内容。）

TTCANopen CAN(FD)转USB产品分为三种类型。

第一类：学习型，主要面向青年学生和初学者，模块主

控芯片为GD32E103或STM32G431,CAN端口为非隔离方式，集成多种功能，以裸板形式提供，可极大降低学习成本。

第二类：安全型，模块主控芯片为STM32H750,CAN端口采用集成磁隔离芯片进行隔离，可有效的保护PC端设备。模块采用黑塑封装，功能同学习型。

第三类：工程型，主要面向工程应用，采用白塑封装，以单一定制功能模块提供，其它同安全型模块。

模块遵循CAN总线协议 2.0A、2.0B和ISO11891-1:2015 CANFD规范，与上位机通讯形式为(USB)虚拟串口，可以在支持CDC类的操作系统上使用，如：WIN、LINUX和MAC等（本模块测试是在WIN10上进行的）。

（CAN(FD)转TTL串口为 3V TTL三线制串口。）

本品可作为普通CAN转USB模块使用，也可作为CANFD转USB使用，切换自如，功能强大。

本品USB端口能够监听5M CANFD总线100%满载，不丢帧。

本着“开放，共享”的应用理念，本品准许用户非商业性DIY，提供资料中包含设备的“电路逻辑图”“PCB印制板图”和“ISP装载固件”，用户可以根据这些文件非常方便的复制出一模一样的作品，从而解除了用户应用的后顾之忧。

2. 接口和主要指标

学习型: USB端口链接器：A型链接器

USB端口通讯：USB2.0全速， 虚拟串口。

供电：采用USB端口+5V供电

TTL端口：三线TTL串口（3V）

TTL串口配置：　波特率：256000

奇偶校验：无

数据位：8

停止位：1

CAN端口链接器：单排3P插针（2.54mm）

CAN收发器：TJA1057GT/3J（最大支持5M）

终端电阻：平衡式终端电阻

CAN通讯速率kbps：10至1000

CANFD数据速率kbps：10至5000

核心ARM芯片：GD32E103或STM32G431

外观尺寸：60×28×13 mm

模块集成：集成两类转换器，支持CAN和CANFD

1. CAN(FD)转USB

2. CAN(FD)转TTL

在两类转换器中，都集成有五种子模块

1).普通调试模块、

2).普通工作模块、

3).TTCANopen调试模块、

4).TTCANopen工作模块、

5).TTCANopen心跳器模块

安全型: USB端口链接器: A型链接器

USB端口通讯: USB2.0全速, 虚拟串口

供电: 采用USB端口+5V供电

TTL端口：三线TTL串口（3V）

TTL串口配置：　波特率：256000

奇偶校验：无

数据位：8

停止位：1

CAN端口链接器: 3P 3.81mm直脚接线端子

CAN收发器: ADM3057或MORNSUN

终端电阻：无

CAN通讯速率kbps：10至1000

CANFD数据速率kbps：10至5000

核心ARM芯片：STM32H750或GD32

外观尺寸：62×30×13 mm

封装：黑塑封装

模块集成：集成两类转换器，支持CAN和CANFD

1. CAN(FD)转USB

2. CAN(FD)转TTL

在两类转换器中，都集成有五种子模块

1).普通调试模块、

2).普通工作模块、

3).TTCANopen调试模块、

4).TTCANopen工作模块、

5).TTCANopen心跳器模块

工程型: 封装：白塑封装

模块集成：用户定制

其它：同安全型

3. 设备接口说明

1）ISP固件加载更新接口

本品为用户提供了ISP固件加载更新接口，见图1所示，用户可以从www.ttcanopen.com网站获取产品的最新升级版本，使用ISP下载器将最新的固件安装到设备上（当然用户也可以将自己研发的固件安装到设备上）。

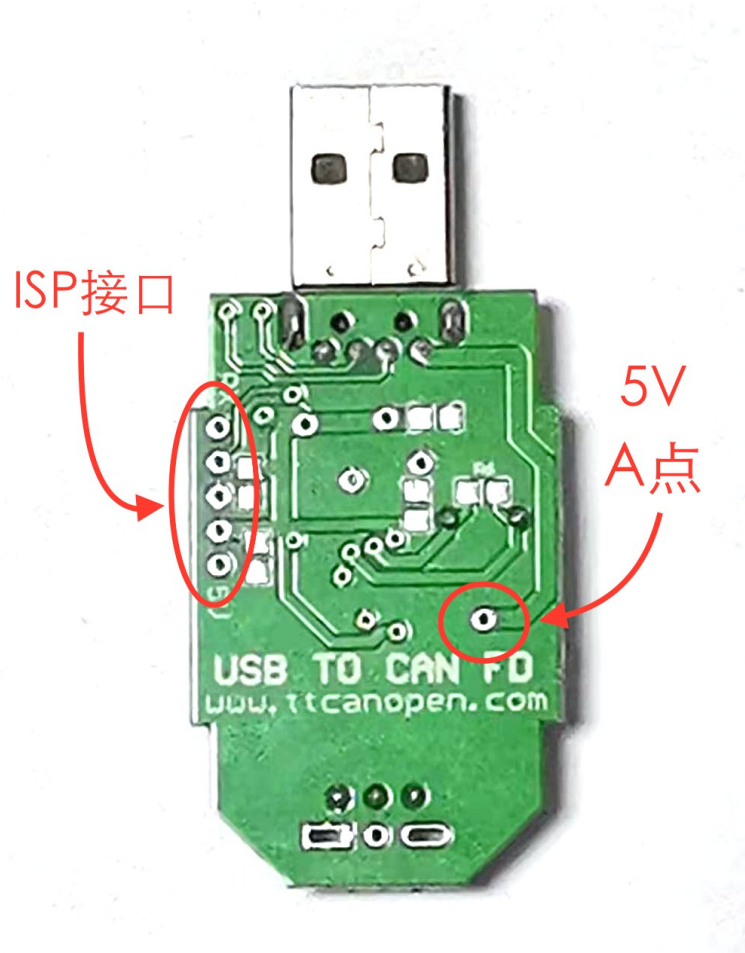


图1　设备ISP固件更新接口

ISP接点从上至下分别为：TX,RX,GND,3.3V,5V。

下载资料中包含ISP运行程序，解压后可以直接运行，请不要擅自更改ISP串口配置参数。

用户可以使用市面上多数ISP下载器安装设备固件，使用时请注意插针的排序，连接不当可能会损毁设备。

用户也可以到TTCANopen淘宝网店购买设备专用的ISP下载器，其使用非常方便，只需将专用ISP下载器的5根插针直接从背面插入到设备的对应ISP接口上即可，见图2所示。

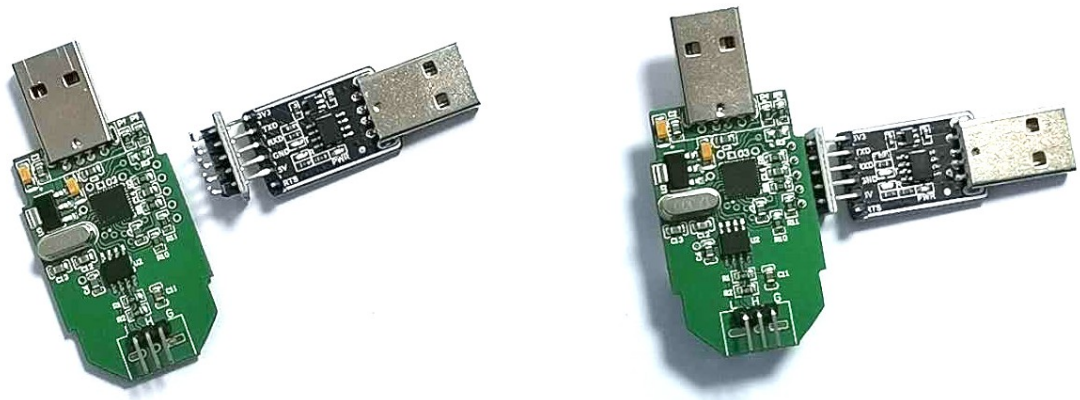


图2 专用ISP下载器的连接

当产品工作在CAN(FD)转TTL串口时（即：不连接USB接口），设备复用了ISP接口的TX,RX,GND做为TTL串行通讯接口；A点为设备5V供电电源接口。

2）CAN(FD)端口

模块CAN(FD)接口有三个接线端子，见图3所示，分别是CAN_L（左）、CAN_H（中）总线信号和GND（右）地线，通常总线信号线为双绞线，地线可选（多为屏蔽地）。

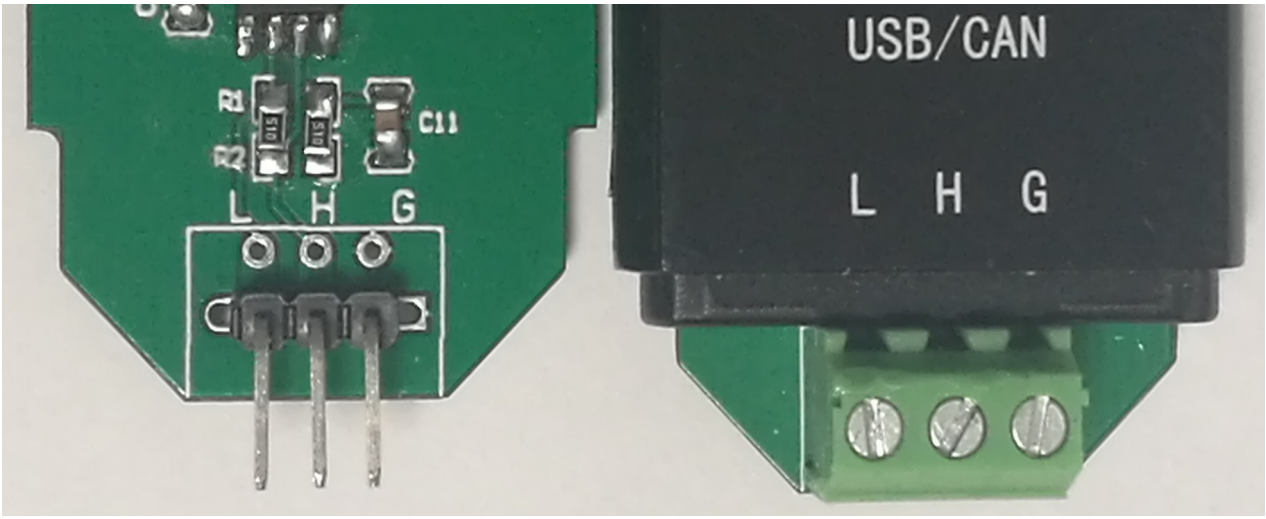


图3 CAN端口接线端子

4. 安装驱动（windows）

WIN10免驱动（linux和MAC免驱），在WIN7和XP上，PC机需安装由GD（或ST）公司提供的虚拟串口驱动。

5. 模块串口基本配置

除了“串口号”外，模块对虚拟串口的其它配置项所属内容都进行了重新定义，打开串口时，在串口接收窗口会显示配置帮助信息和当前模块配置状态信息,见图4所示。

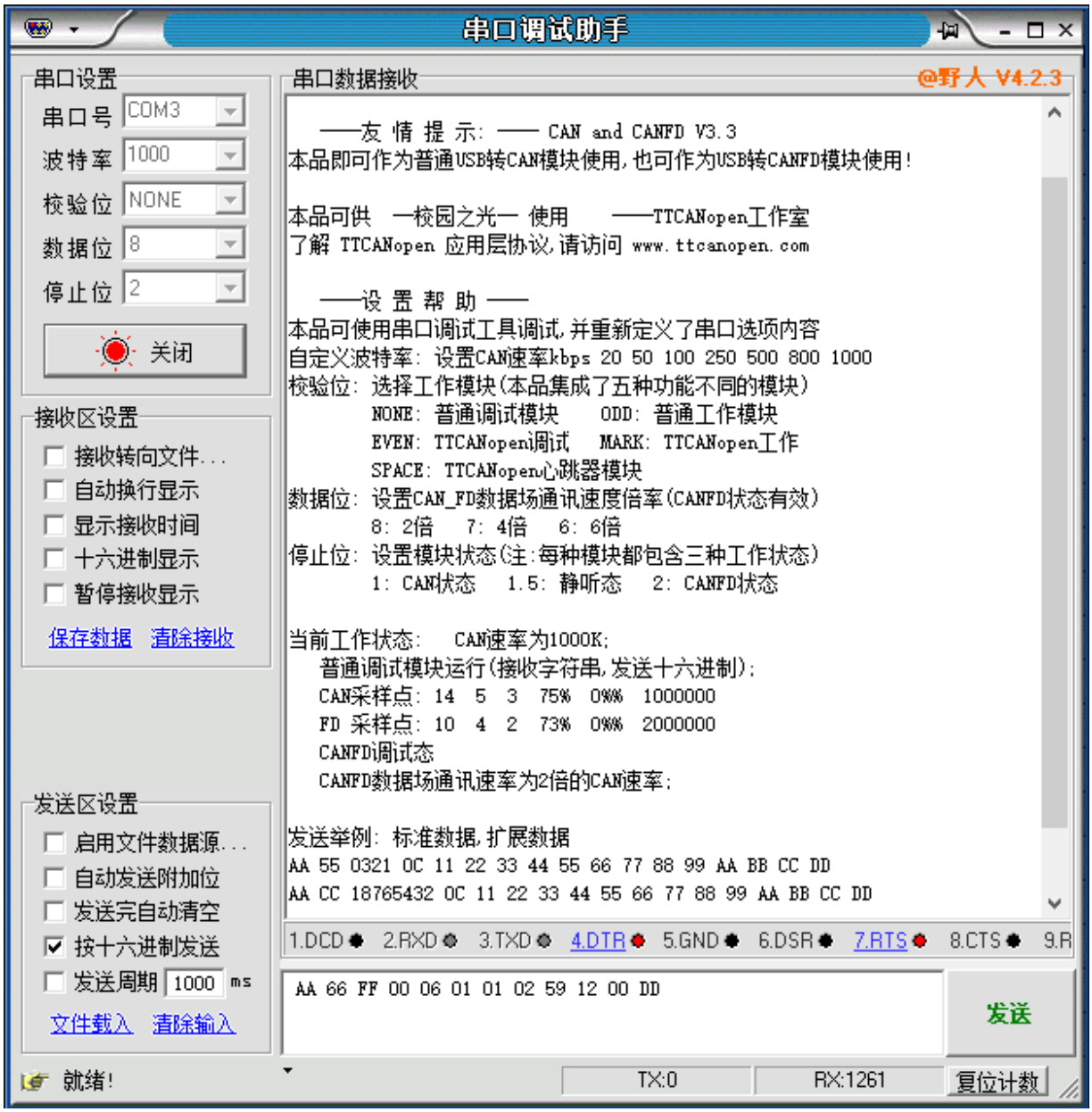


图4　模块配置帮助信息和当前状态信息

下面逐一介绍各项配置：

波特率: （通常使用自定义波特率方式）设置CAN总线通讯速率（单位bps），可输入10000、20000、50000、100000、125000、250000、500000、800000、1000000。

也可使用kbps单位输入10、20、50、100、125、250、500、800、和1000。

对于没有自定义波特率输入方式的串口调试工具，可以使用表1对应关系输入CAN总线通讯速率。

表1 串口波特率与CAN端口通讯速率的对应关系

序号	串口波特率	CAN端口速率kbps
1	9600	10
2	14400	20
3	19200	50
4	38400	100
5	56000	125
6	57600	250
7	115200	500
8	128000	800
9	256000	1000

校验位: 选择功能模块（注：产品集成了5种功能模块）

NONE：　普通调试模块

ODD：　普通工作模块

EVEN：　TTCANopen调试模块

MARK：　TTCANopen工作模块

SPACE：TTCANopen心跳器模块

数据位: 选择CANFD数据域发送倍率（CANFD状态有效）

8：2倍

7：4倍

6：6倍

停止位: 设置模块状态（每种模块都包含三种工作状态）

1：　CAN状态

1.5：　静听状态

2：　CANFD状态

6. 普通调试模块

当串口**校验位**配置为NONE时，模块被配置为普通调试模块。

普通调试模块，是用户在调试已有CAN或CANFD网络时，临时加入的调试设备，待调试完成后脱离原网络。其不是原CAN网络的组成部分。

用户可以通过配置串口的**停止位**，将模块配置为CAN状态或CANFD状态，使其与被调试的CAN网络状态相匹配，当用户只需对原网络进行监听时，可将模块配置为静听态，这样对原网络的影响最小，并能同时静听原网络中的CAN指令和CANFD指令。

在CANFD状态，用户可以通过设置串口的**数据位**，调整CANFD数据场的收发速率，通常为CAN速率的2倍、4倍或6倍

用户也可以使用本说明书第8节所介绍的指令方式设置模块参数（详见第8节）。

为了方便用户观察原网络中指令的运行情况，普通调试模块虚拟串口输出为字符串格式，每帧指令独占一行，用户可使用各种串口调试工具显示调试。

普通调试模块支持4种指令帧的显示，其格式如下：

1）显示标准遥控帧指令格式：

帧计数	接收时标	帧标识	ID字	DLC	帧标识
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	<	0000至07EF	0至8	>

2）显示标准数据帧指令格式：

帧计数	接收时标	帧标识	ID字	DLC	帧标识	数据场DLC个字节
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	[	0000至07EF	0至8	]	xx ～ xx

4）显示扩展遥控帧指令格式：

帧计数	接收时标	帧标识	ID字	DLC	帧标识
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	<	00000000至1FBFFFFF	0至8	>

4）显示扩展数据帧指令格式：

帧计数	接收时标	帧标识	ID字	DLC	帧标识	数据场DLC个字节
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	[	00000000至1FBFFFFF	0至8	]	xx ～ xx

注：上述格式是按普通CAN状态表述的，CANFD状态不支持遥控帧，CANFD指令中DLC字段和其所对应的数据场长度，遵守CANFD协议相关规定。DLC字段用10进制显示。

普通调试模块接收指令显示举例：

1）标准遥控帧：

00000001 00:03:07:671.200 < 0123 8 >

2）标准数据帧：

00000002 00:03:07:671.346 [0321 4] 11 22 33 44

3）扩展遥控帧：

00000003 00:03:07:671.443 < 12345678 6 >

4）扩展数据帧：

00000004 00:03:07:671.605 [18765432 3] 11 22 33

上例中:（举例:扩展数据帧）

第一个字符串“00000004”：

为接收帧计数，计数范围 0～0xFFFFFFFF;

第二个字符串“00:03:07:671.605”：

为接收时标，时：分：秒：毫秒.微妙，分辨率为1微秒;

第三个[扩起来]的字符串“[18765432 3]”：

为帧ID和DLC，(ID为32位十六进制数,低29位有效。DLC为十进制数，CAN状态为0至8，CANFD状态为0至64，并遵守CANFD协议相关规定)；<尖括号>则表述遥控帧;

第四个字符串“11 22 33 ”：

为数据帧中DLC个字节数据内容。

（注：对于标准帧，ID为16位十六进制数，低11位有效）

图5 为在串口调试工具接收窗口观察到的四种帧举例，分别为：标准遥控帧、标准数据帧、扩展遥控帧和扩展数据帧。

图5 普通调试模块接收显示举例

普通调试模块支持4种指令帧的发送，并对发送帧进行“回显”，发送指令使用十六进制方式，其格式如下：

1）发送标准遥控帧指令格式：

帧头	帧标识	ID字	DLC	帧尾
AA	77	0000 至 07EF	00 至 08	DD

2）发送标准数据帧指令格式：

帧头	帧标识	ID字	DLC	数据场DLC个字节	帧尾
AA	55	0000 至 07EF	00 至 08	xx ～ xx	DD

3）发送扩展遥控帧指令格式：

帧头	帧标识	ID字	DLC	帧尾
AA	FF	00000000 至 1FBFFFFF	00 至 08	DD

4）发送扩展数据帧指令格式：

帧头	帧标识	ID字	DLC	数据场DLC个字节	帧尾
AA	CC	00000000 至 1FBFFFFF	00 至 08	xx ～ xx	DD

注：CANFD不支持遥控帧，CANFD指令中DLC字段和其所对应的数据场长度，遵守CANFD协议相关规定。

普通调试模块发送指令举例（十六进制方式）：

1）发送标准遥控帧：

AA 77 0123 08 DD

2）发送标准数据帧：

AA 55 0321 08 11 22 33 44 55 66 77 88 DD

3）发送扩展遥控帧：

AA FF 12345678 06 DD

4）发送扩展数据帧：

AA CC 18765432 08 11 22 33 44 55 66 77 88 DD

图6 为串口调试工具中普通调试模块发送指令和回显示例，分别为：标准遥控帧、标准数据帧、扩展遥控帧和扩展数据帧。

图6 普通调试模块发送指令和回显示例

7. 普通工作模块

当串口**校验位**配置为ODD时，模块被配置为普通工作模块。

普通工作模块，是已有CAN或CANFD网络原生的组成部分，其主要承担CAN网络与系统上位机的通讯连接和协议转换，是中心监控系统的交通枢纽。

用户可以通过配置串口的**停止位**，将模块配置为CAN状态或CANFD状态，使其与固有CAN网络状态保持一致。

在CANFD状态，用户可以通过设置串口的**数据位**，调整CANFD数据场的收发速率，通常为CAN速率的2倍、4倍或6倍

用户也可以使用本说明书第8节所介绍的指令方式设置模块参数（详见第8节）。

为了提高指令效率，模块虚拟串口输入输出皆为16进制格式，帧与帧之间使用帧头帧尾标识。

普通工作模块支持4种指令帧的输入输出，其中，发令格式与普通调试模块相同，但普通工作模块不“回显”发送指令。

普通工作模块接收指令格式如下：

1）接收标准遥控帧指令格式：

帧头	接收时标	帧标识	ID字	DLC	帧尾
BB	tt tt tt tt	77	0000 至 07EF	00 至 08	EE

2）接收标准数据帧指令格式：

帧头	接收时标	帧标识	ID字	DLC	数据场 DLC个字节	帧尾
BB	tt tt tt tt	55	0000 至 07EF	00 至 08	xx ～ xx	EE

3）接收扩展遥控帧指令格式：

帧头	接收时标	帧标识	ID字	DLC	帧尾
BB	tt tt tt tt	FF	00000000 至 1FBFFFFF	00 至 08	EE

4）接收扩展数据帧指令格式：

帧头	接收时标	帧标识	ID字	DLC	数据场 DLC个字节	帧尾
BB	tt tt tt tt	CC	00000000 至 1FBFFFFF	00 至 08	xx ～ xx	EE

注1：tt tt tt tt为32位接收时间计数（计数单位为0.1ms），其低字节在前，高字节在后；而ID字是高字节在前，低字节在后，解帧时需特别留意。

注2：CANFD不支持遥控帧，CANFD指令中DLC字段和其所对应的数据场长度，遵守CANFD协议相关规定。

普通工作模块接收指令举例：

1）标准遥控帧：

BB B5 3C 02 00 77 01 23 08 EE

2）标准数据帧：

BB B6 3C 02 00 55 03 21 08 11 22 33 44 55 66 77 88 EE

3）扩展遥控帧：

BB B7 3C 02 00 FF 12 34 56 78 06 EE

4）扩展数据帧：

BB B9 3C 02 00 CC 18 76 54 32 08 11 22 33 44 55 66 77 88 EE

图7 为普通工作模块，指令接收示例，分别为：标准遥控帧、标准数据帧、扩展遥控帧和扩展数据帧，在这里每帧指令并没有独立成行，而是用帧头BB和帧尾EE进行分割的。

图7 普通工作模块接收指令示例

8. 使用“指令”配置模块参数

模块除了可以使用串口配置项定义模块的配置参数和工作状态，也可以使用专用指令进行上述操作。

（请注意：在本章v3.2版本对V3.1版本进行了较大的修改，配置参数更加方便，但略有不兼容。如果用户不使用“指令”配置模块参数则可以不用升级。在这里本章只介绍v3.2版本的内容。）

（请注意：v3.3版新增了采样点选择项。）

实际上，在模块内部分配了一个固有的寄存器空间来存放模块的配置参数，用户可以通过修改该寄存器中的内容，完成配置参数和设置工作状态。表2为该寄存器中相关参数的分配状况。

表2 模块固有的配置参数寄存器空间的分配

序号	寄存器地址	字节数	参数描述
1	FF00	1	当该值置1时，启动配置，然后自动归0.
2	FF01	1	选择工作模块： 0:为普通调试模块 1:为普通工作模块 2:为TTCANopen调试模块 3:为TTCANopen工作模块 4:为TTCANopen心跳模块
3	FF02	1	配置工作状态： 0:为CAN状态 1:为静听态 2:为CANFD态
4	FF03	1	配置CAN和CANFD速率： 低半字节配置CAN速率 1～9分别对应CAN速率的 10、20、50、100、125、250、500、800、1000 kbps 高半字节配置CANFD的倍率 1～A分别对应CANFD为 CAN速率的1～10倍
5	FF04	2	设置两帧发送间隔（微妙） 高半字节为发送间隔首数据， 低半字节为0的个数。 注意：当该字节被设置为 0xAA时 将由模块根据通讯速率自动生成发送间隔

6	FF05	1	选择配置采样点 低半字节配置CAN采样点 高半字节配置FD采样点 0:为自动配置采样点 1:为75%左右 2:为80%左右 3:为85%左右 4:同v3.2版
7	FF06～FF0F	2	保留

为了能够设置和修改该寄存器内容，设计了专用指令格式如下：

设置模块配置参数寄存器指令格式：

帧头	帧标识	寄存器地址	字节数	数据内容	帧尾
AA	66	FF00 至 FF0F	01 至 10	xx ～ xx	DD

用户配置完参数后，需将0xFF00寄存器置1，新配置方才生效，生效后该寄存器内容自动归零。

配置模块参数举例：

例1；修改CAN发送速率为1000000bps

AA 66 FF03 01 19 DD

AA 66 FF00 01 01 DD

例2:修改CAN工作状态为CANFD

AA 66 FF02 01 02 DD

AA 66 FF00 01 01 DD

当然用户也可以使用一条指令将配置参数一次完成（推荐）。

例3:一次配置全部参数，包括配置立即生效

AA 66 FF00 06 01 01 02 59 12 00 DD

该指令配置模块为“普通工作模块”、“CANFD态”、“CAN速率1000000bps”、“CAN FD速率5000000bps”、“发送间隔100微妙”、“自动配置采样点”配置立即生效。

采样点信息描述：（在配置指令反馈信息中）

CAN采样：14　5　3　75%　0%% 1000000

FD 采样：　8　3　1　75%　0%% 5000000

上述信息依次为：

BS1　BS2　BRP　SamplePoint　Error　BaudRate

（注：本模块CAN时钟为60M，在配置FD波特率和采样点时，应尽可能的使Error=0）

当配置成功后，虚拟串口接收窗口会有信息显现见图8.

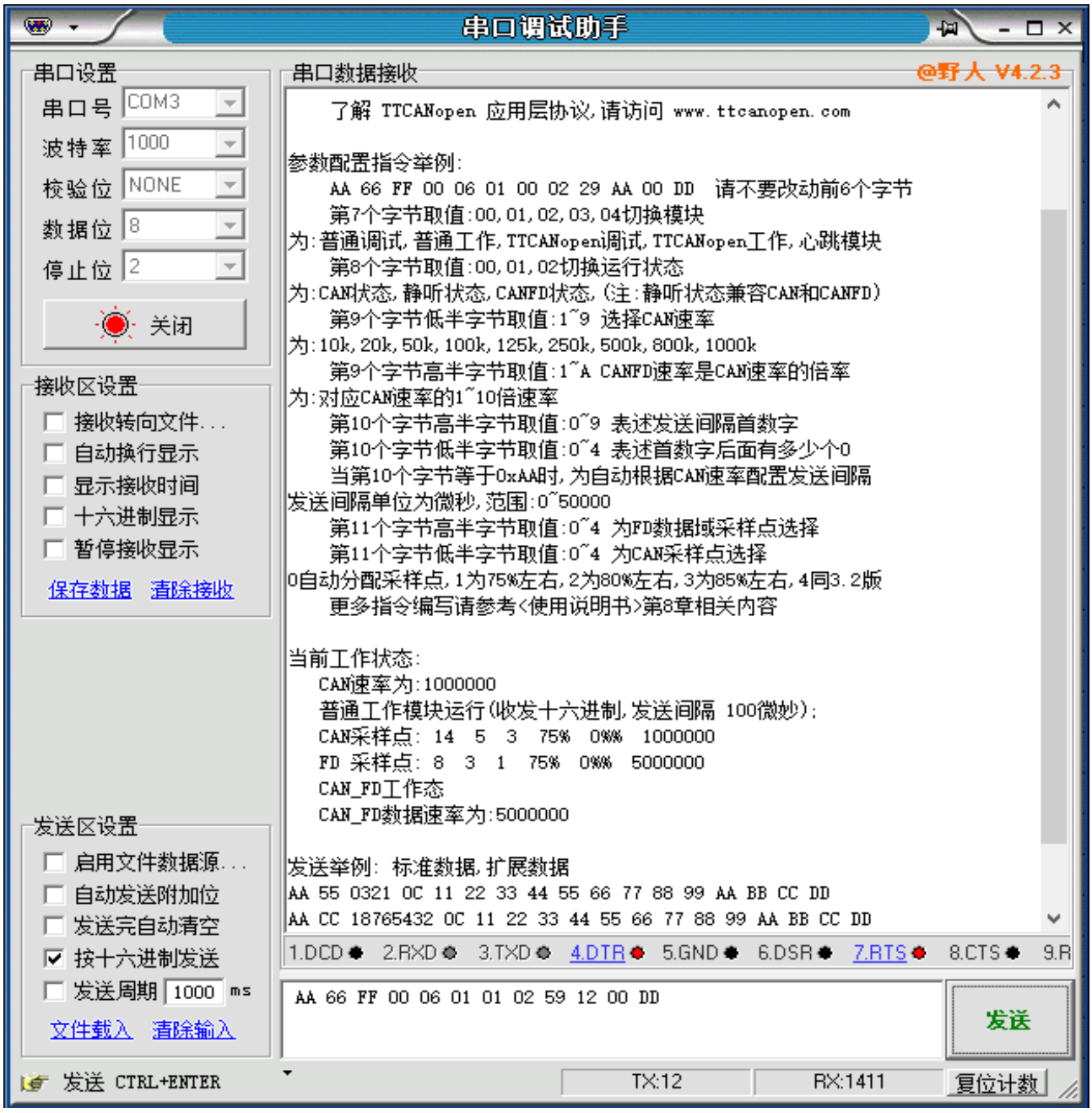


图8 使用配置指令成功反馈显示信息举例

用户在配置模块参数时，需要注意可配置参数的取值范围以其适用性，如：模块的通讯速率除了受主控MCU芯片的限制，同时也受CAN接口芯片指标的限制，以及终端匹配电阻的连接形式的影响。用户在使用模块时应当按照所用模块的硬件配置和应用环境，进行合理设置。 表3为主控芯片和接口芯片CAN通讯速率相关参数：

表3 模块所用各芯片的CAN通讯速率范围

芯片代号	CAN速率kbps	FD速率kbps
GD32E103	10至1000	10至6000
STM32H750	10至1000	10至5000
TJA1057	40至1000	40至5000
ADM3057	10至1000	10至12000
QBD1044	10至1000	10至6000

另外，需要提醒用户的是对于定制“工程模块”，其配置参数和工作状态是用户定制的，当模块上电后，打开串口，定制参数立即生效。

注：CAN(FD)转TTL所有收发和配置指令以及模块所支持的工作状态与CAN(FD)转USB相同，不再赘述。由于TTL串口为真实串口，需按通讯要求配置，不能利用它配置CAN端口参数。

说明书以下部分与TTCANopen协议有关，有需求的用户可以继续阅读。

本品为业界首个直接支持TTCANopen应用层协议的USB转CAN_FD模块，用户可以访问www.ttcanopen.com网站获得更多的TTCANopen相关知识。

TTCANopen是由国人打造的一款拥有自主知识产权的CAN应用层协议，与以往的应用层协议不同， TTCANopen 实际上是一个开放的协议框架。它是由众多的“基本子协议”和“应用子协议”构成，用户可以根据自己的专业应用特点，去选择“应用子协议”;也可以使用“基本子协议”去生成新的“应用子协议”; TTCANopen协议在应用层界面直接兼容最新的CAN_FD。

TTCANopen 应用层协议以连续寄存器空间为信息载体，使用了两个不同的CAN标识符划分的方式分别传递过程变量和非过程变量，巧妙的解决了寄存器空间分层管理与其指令优先级分配不平衡的矛盾。

TTCANopen还特别规范了设备内部多条指令的发送原则，这是其它应用层协议所不具备的。

TTCANopen引入了系统心跳的概念，用以同步总线上的设备，使系统拥有统一的时钟，并使用时间相位矩阵取代了传统的指令查询矩阵，提高了总线带宽的利用率，大大的减小了浪涌出现的概率，使应用系统实现了由传统的指令驱动到现代的时间驱动的跨越。

TTCANopen应用层协议对CAN标识符的划分及其物理含义的分配，如:优先级段、寄存器基地址段、设备编号段、功能码段都具有明确的应用层协议指令含义，据此设计的应用层协议具有很好的直读性，易于解析，大大降低了学习入门的难度。是一款非常适合推广和普及的应用层协议。

另外，TTCANopen有着类modbus特性，使其能够非常容易的和市面上大多数组态软件完美对接。为用户构建应用系统提供了极大的便利。

9. TTCANopen调试模块

当串口**校验位**配置为**EVEN**时，模块被配置为TTCANopen调试模块。

TTCANopen调试模块，是用户在调试已有TTCANopen网络时，临时加入的调试设备，待调试完成后脱离原网络。其不是原TTCANopen网络的组成部分。

用户可以通过配置串口的**停止位**，将模块配置为CAN状态或CANFD状态，使其与被调试的TTCANopen网络状态相匹配，当用户只需对原网络进行监听时，可将模块配置为静听态，并能同时静听原网络中的CAN指令和CANFD指令。

在CANFD状态，用户可以通过设置串口的**数据位**，调整CANFD数据场的收发速率，通常为CAN速率的2倍、4倍或6倍

用户也可以使用本说明书第8节所介绍的指令方式设置模块参数（详见第8节）。

为了方便用户观察原网络中指令的运行情况，TTCANopen调试模块虚拟串口输出为字符串格式，每帧指令独占一行，用户可使用各种串口调试工具显示调试。

与普通模块不同，TTCANopen模块将29位ID字拆分为四个具有实际物理含义的子段，分别为：优先级段、寄存器基地址段、设备编号段和功能码段，使其能够贴近实际应用构成一套完整的应用层指令系统和管理系统。

TTCANopen调试模块支持4种TTCANopen指令帧的显示，其格式如下：

1）显示TTCANopen扩展数据帧指令格式：（在用）

帧计数	接收时标	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	帧标识
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	[	0或1	0000至FBFF	00至7F	00至1F	00至08	]

数据场DLC个字节
xx ～ xx

2）显示TTCANopen扩展遥控帧指令格式：（未用）

帧计数	接收时标	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	帧标识
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	<	0或1	0000至FBFF	00至7F	00至1F	00至08	>

3）显示miniTTCANopen标准数据帧指令格式：（未用）

帧计数	接收时标	帧标识	寄存器地址	段地址	DLC	帧标识	数据场DLC字节
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	[	00至FD	00至07	00至08	]	xx ～ xx

4）显示miniTTCANopen标准遥控帧指令格式：（未用）

帧计数	接收时标	帧标识	寄存器地址	段地址	DLC	帧标识
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	<	00至FD	00至07	00至08	>

注1：上述格式是按普通CAN状态表述的，CANFD状态不支持遥控帧，CANFD指令中DLC字段和其所对应的数据场长度，遵守CANFD协议相关规定。DLC字段用10进制显示。

注2：对于标准帧，支持miniTTCANopen协议，尚未推出。

注3：TTCANopen应用层协议只使用了29位扩展数据帧。

TTCANopen 调试模块显示帧举例：

1）TTCANopen 扩展数据帧：（在用）

00000012 23:18:56:566.211 [1 7100 32 19 2] 11 22

2）TTCANopen 扩展遥控帧：（注：未启用）

00000013 23:18:56:566.623 < 1 7100 32 19 8 >

3）miniTTCANopen 标准数据帧：（注：未启用）

00000014 23:18:56:567.135 [35 3 4] 11 22 33 44

4）niniTTCANopen 标准遥控帧：（注：未启用）

00000015 23:18:56:567.376 < 35 3 8 >

上例中:

第一个字符串“00000012”：

为接收帧计数，计数范围 0～0xFFFFFFFF;

第二个字符串“23:18:56:566.211”：

为接收时标，分辨率为1微秒；

第三个[扩起来]的字符串“[1 7100 32 19 2]”：

分别为 1位优先级、16位寄存器基地址、7位设备地址、5位指令功能码和DLC；　（DLC为十进制数，CAN状态其为0至8，CANFD状态为0至64，并遵守CANFD协议相关规定）〈尖括号〉则表述遥控帧；

第四个字符串“11 22”：

为数据帧中DLC个字节数据内容。

图9 为在串口调试工具接收窗口观察到的四种帧举例，分别为：miniTTCANopen标准遥控帧、miniTTCANopen标准数据帧、TTCANopen扩展遥控帧和TTCANopen扩展数据帧。

图9 TTCANopen调试模块接收显示举例

TTCANopen调试模块支持4种指令帧的发送，并对发送帧进行“回显”，发送指令使用十六进制方式，其格式如下：

1）发送TTCANopen扩展数据帧指令格式：（在用）

帧头	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	数据场 DLC个字节	帧尾
AA	CC	00 或 01	0000 至 FBFF	00 至 7F	00 至 1F	00 至 08	xx ～ xx	DD

2）发送TTCANopen扩展遥控帧指令格式：（未用）

帧头	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	帧尾
AA	FF	00 或 01	0000 至 FBFF	00 至 7F	00 至 1F	00 至 08	DD

3）发送miniTTCANopen标准数据帧指令格式：（未用）

帧头	帧标识	寄存器地址	段地址	DLC	数据场 DLC个字节	帧尾
AA	55	00 至 FD	00 至 07	00 至 08	xx ～ xx	DD

4）发送miniTTCANopen标准遥控帧指令格式：（未用）

帧头	帧标识	寄存器地址	段地址	DLC	帧尾
AA	77	00 至 FD	00 至 07	00 至 08	DD

注1：寄存器基地址为16位十六进制数，高地址在前，低地址在后。支持标准帧的miniTTCANopen协议格式尚未推出。

注2：CANFD不支持遥控帧，CANFD指令中DLC字段和其所对应的数据场长度，遵守CANFD协议相关规定。

在串口调试工具中发送指令举例：

1）扩展数据帧：（注：TTCANopen协议只使用扩展数据帧）

AA CC 01 7100 32 19 08 11 22 33 44 55 66 77 88 DD

2）扩展遥控帧：（注：未启用）

AA FF 01 7100 32 19 08 DD

3）标准数据帧：（注：未启用）

AA 55 35 03 08 11 22 33 44 55 66 77 88 DD

4）标准遥控帧：（注：未启用）

AA 77 35 03 08 DD

图10 为串口调试工具中TTCANopen调试模块发送指令和回显示例，分别为：miniTTCANopen标准遥控帧、miniTTCANopen标准数据帧、TTCANopen扩展遥控帧和TTCANopen扩展数据帧。

图10 TTCANopen调试模块发送指令和回显示例

10. TTCANopen工作模块

当串口**校验位**配置为**MARK**时，模块被配置为TTCANopen工作模块。

TTCANopen工作模块，是已有TTCANopen网络原生的组成部分，其主要承担TTCANopen网络与系统上位机的通讯连接和协议转换，是中心监控系统的交通枢纽。利用TTCANopen协议“组态王”驱动，用户可快速建立起自己的中心监控系统。

用户可以通过配置串口的**停止位**，将模块配置为CAN状态或CANFD状态，使其与固有TTCANopen网络状态保持一致。

在CANFD状态，用户可以通过设置串口的**数据位**，调整CAN_FD数据场的收发速率，通常为CAN速率的2倍、4倍或6倍。

用户也可以使用本说明书第8节所介绍的指令方式设置模块参数（详见第8节）。

为了提高指令效率，TTCANopen工作模块虚拟串口输入输出皆为16进制格式，帧与帧之间使用帧头帧尾标识。

TTCANopen工作模块支持4种指令帧的输入输出，其中，发令格式与TTCANopen调试模块相同，但TTCANopen工作模块不“回显”发送指令。

TTCANopen工作模块接收格式如下：

1）接收TTCANopen扩展数据帧指令格式：（在用）

帧头	接收时标	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	数据场 DLC个字节	帧尾
BB	tt tt tt tt	CC	00或01	0000至FBFF	00至7F	00至1F	00至08	xx ～ xx	EE

2）接收TTCANopen扩展遥控帧指令格式：（未用）

帧头	接收时标	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	帧尾
BB	tt tt tt tt	FF	00或01	0000至FBFF	00至3F	00至1F	00至08	EE

3）接收miniTTCANopen标准数据帧指令格式：（未用）

帧头	接收时标	帧标识	寄存器地址	段地址	DLC	数据场 DLC个字节	帧尾
BB	tt tt tt tt	55	00至FD	00至07	00至08	xx ～ xx	EE

4）接收miniTTCANopen标准遥控帧指令格式：（未用）

帧头	接收时标	帧标识	寄存器地址	段地址	DLC	帧尾
BB	tt tt tt tt	77	00 至 FD	00 至 07	00 至 08	EE

注1：tt tt tt tt为32位接收时间计数（0.1ms），其低字节在前，高字节在后；而ID字是高字节在前，低字节在后，解帧时需特别留意。

注2：CANFD不支持遥控帧，CANFD指令中DLC字段和其所对应的数据场长度，遵守CAN_FD协议相关规定。

在串口调试工具中接收指令举例：

1）TTCANopen扩展数据帧：（在用）

BB 15 23 00 00 CC 01 71 00 32 19 03 11 22 33 EE

2）TTCANopen扩展遥控帧：（注：未启用）

BB 16 23 00 00 FF 01 71 00 32 19 08 EE

3）miniTTCANopen标准数据帧：（注：未启用）

BB 18 23 00 00 55 35 03 06 11 22 33 44 55 66 EE

4）miniTTCANopen标准遥控帧：（注：未启用）

BB 19 23 00 00 77 35 03 08 EE

图11 为TTCANopen工作模块，指令接收示例，在这里每帧指令并没有独立成行，而是用帧头BB和帧尾EE进行分割的。

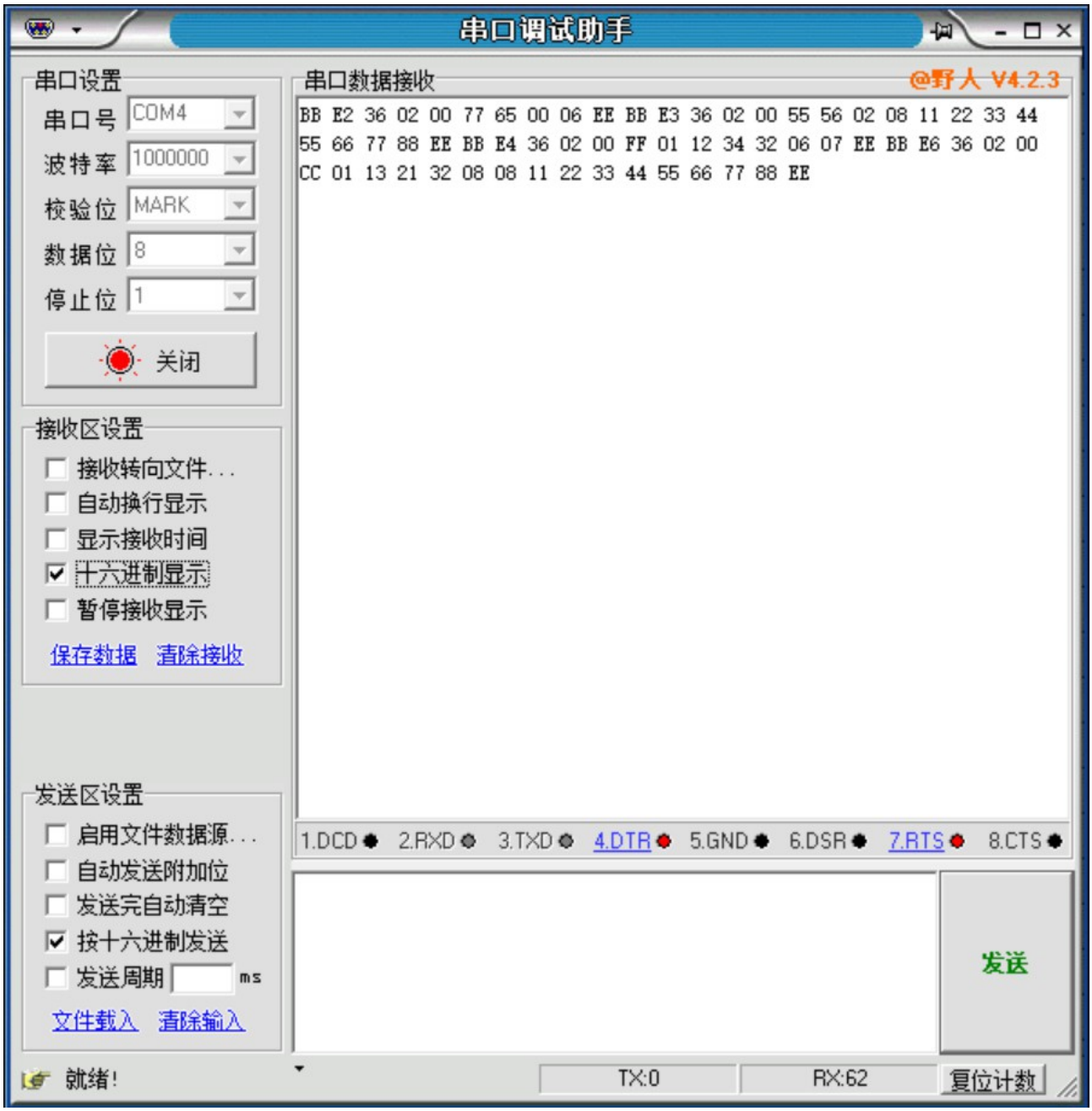


图11 TTCANopen工作模块接收指令演示

11. TTCANopen心跳器模块

当串口**校验位**配置为SPACE时，模块被配置为TTCANopen心跳器模块。

心跳器模块，是为TTCANopen网络提供系统心跳的设备，网络上的设备收到系统心跳会将其本地时间与心跳时间同步，完成总线时间统一。

用户可以通过配置串口的**停止位**，将模块配置为CAN状态或CAN_FD状态，使其与TTCANopen网络状态保持一致。

在CANFD状态，用户可以通过设置串口的**数据位**，调整CANFD数据场的发送速率，通常为CAN速率的2倍、4倍或6倍

用户也可以使用本说明书第8节所介绍的指令方式设置模块参数（详见第8节）。

心跳器，完成一次心跳分为两帧发送，第一帧为时间预报帧，第二帧为心跳同步帧，两帧通常间隔4～5帧时间。

预报帧：00 0A02 01 19 08 tt tt tt tt re te TT TT

心跳帧：00 0A03 01 19 00

其中：tt tt tt tt 为“预报时间”是紧跟着的“心跳同步帧”的结束时刻，以0.1ms为记时单位，0～0x337F97FF是一天中0.1ms的计数个数（最大863999999），即：所谓

TTCANopen时间。

re和te是“CAN接收错误计数”和“CAN发送错误计数”是CAN端口收发错误寄存器中的计数值，是心跳器CAN端口的通讯质量的表征。

TT TT是“心跳周期”以ms为计数单位。默认为1（E803）秒。

心跳同步帧为固定帧，帧发送结束时刻为时标，网络上的设备，接收此帧后立即将其本地时间值同步到预报时间上，完成与系统时间的同步。

在同一网络中只能有一个心跳器工作，其它心跳器处于备机状态，并将本地时间自动同步到主心跳器上，当检查到主心跳器十次没有发出正常心跳时，备机会自动启动替换主心跳器。

心跳器每发送一次心跳都会在USB端仿真串口接收窗口显示当前发送的时间。见图12所示。

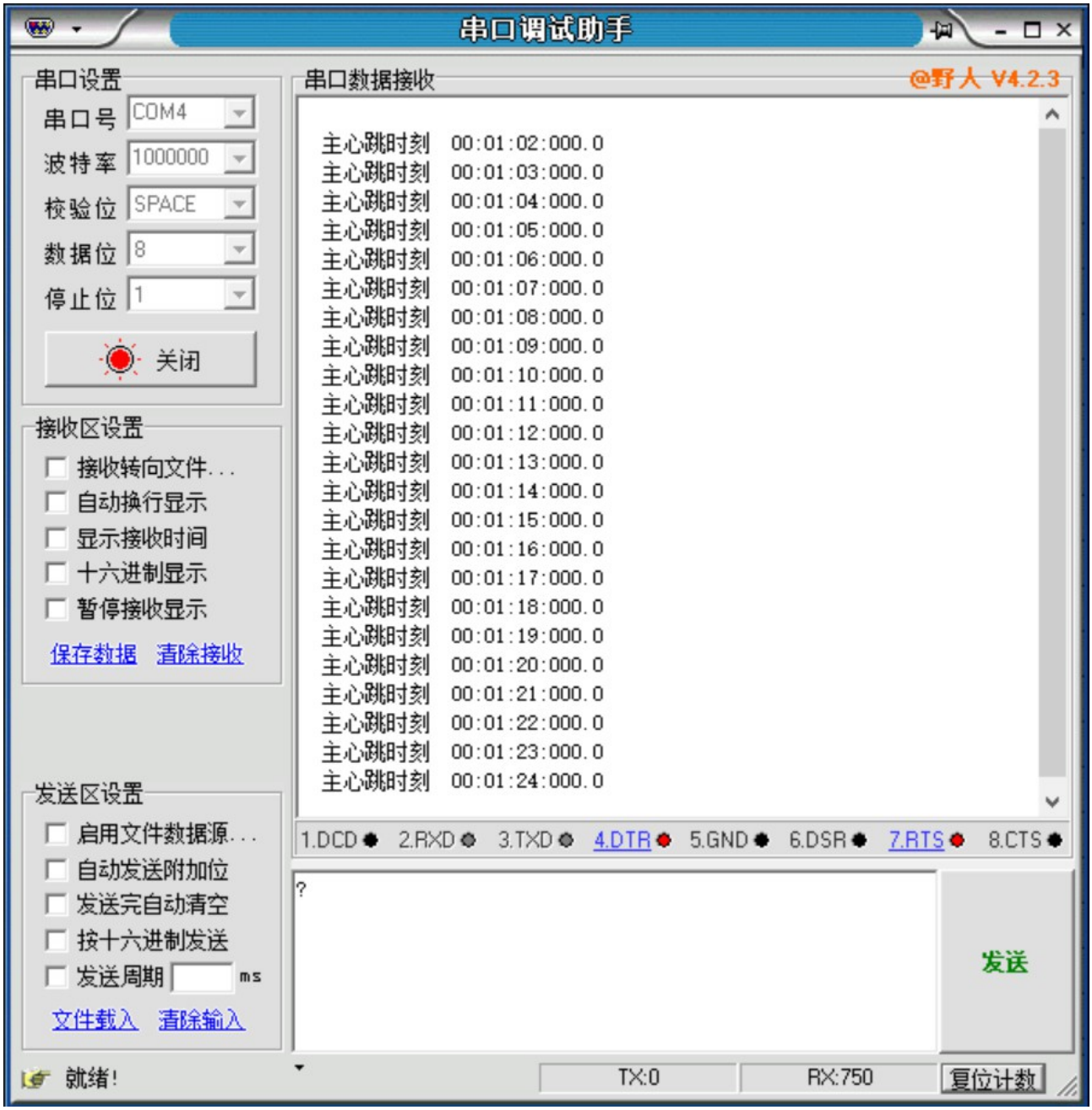


图 12 心跳器显示窗口

用户可以通过USB仿真串口的发送端，控制心跳器的启动和停止，也可以设置心跳器的起始时间和心跳周期。

设置串口调试工具的发送窗口为字符方式，然后输入？问号并发送，这时在接收窗口会出现帮助信息如图13所示。

图13 心跳器帮助信息

用户可以根据该提示在发送窗口进行所需操作，出错时，会有相关提示。

如果用户需要将心跳周期改为2秒，就可以在发送窗口输入：L02000回车，然后点“发送键”；

如果用户需要将起始时间改为12:12:12:000.0，　可输入：T12:12:12:000.0回车，然后发送。

如果用户需求启动或停止心跳，可输入：Q，　然后发送。

用户也可以发送：N　来结束心跳器工作，恢复设备到调试模块。

12. 模块TTCANopen特性（只针对TTCANopen状态）

1）模块遵从“TTCANopen应用层协议CAN标识的分配方法”，见附录1；

2）模块遵从“TTCANopen应用层协议设备内部多条指令的发送规则”，见附录2；

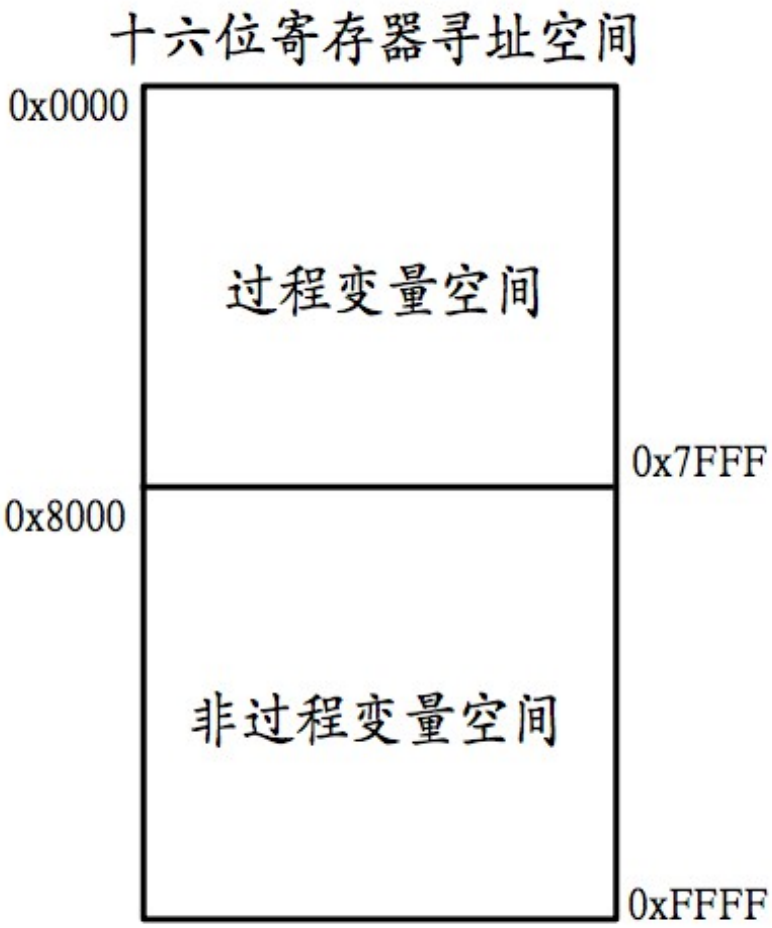
3）模块接受系统心跳，并与之同步，图14为TTCANopen调试模块的时间同步过程。

图14 TTCANopen调试模块与系统时间同步

附录1:　　TTCANopen应用层协议

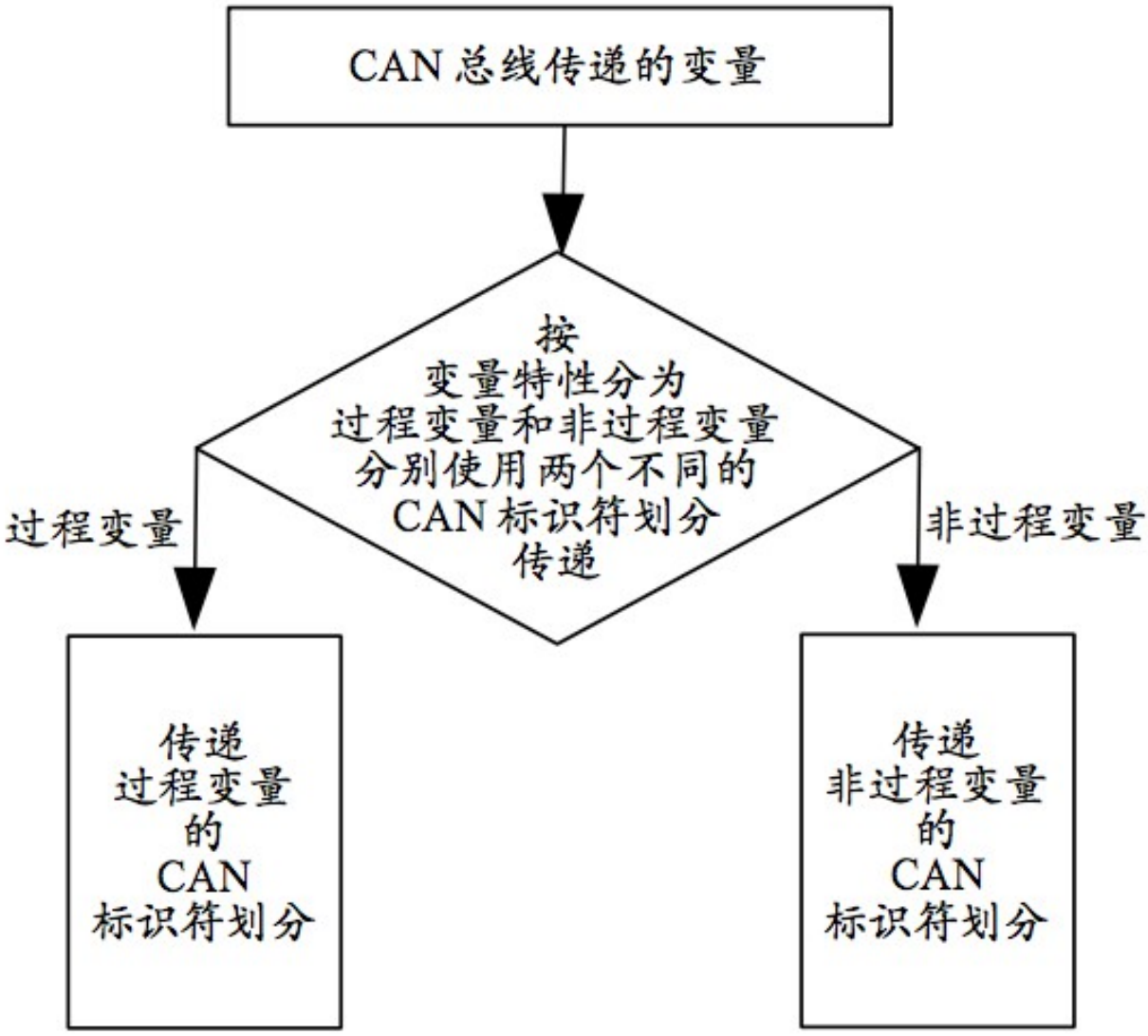
CAN标识的分配方法

TTCANopen应用层协议以连续寄存器空间为信息载体，并将其十六位寄存器寻址空间平分为两部分，见附图1所示。其中：0x0000～0x7FFF用于存放、传递过程变量；0x8000～0xFFFF用于存放、传递非过程变量。



附图1 寄存器空间的划分

TTCANopen 应用层协议根据CAN总线传递变量特性的不同，即过程变量或非过程变量，使用两个不同的CAN标识符划分分别传递上述两种变量，见附图2所示，从而为两种变量提供了不同的总线竞争途径，并为应用层协议实施生产者-消费者模型和主机-从机模型提供了明确的对象指向。



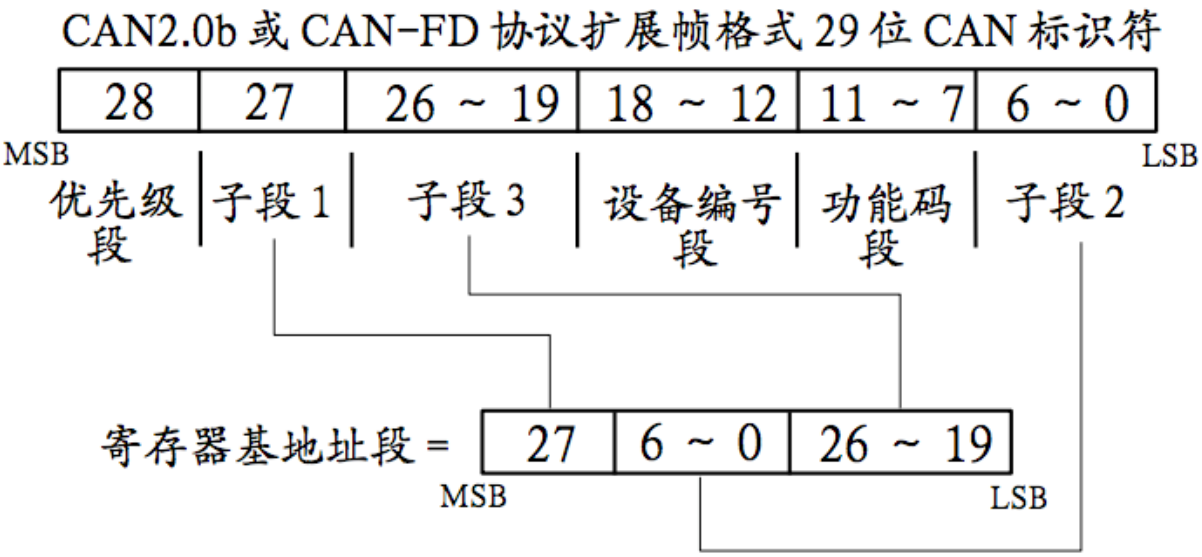
附图2 使用不同的CAN标识符划分传递两种不同的变量

传递过程变量的CAN标识符划分，采用了在CAN标识符中抽取多个位片段，并重新排列拼接组合的方式，构成应用层传递过程变量的寄存器基地址段，从而可以在过程变量寄存器基地址空间构成多个等价竞争层，很好的解决了变量分层管理与总线竞争分配不平衡的矛盾。

传递过程变量的CAN标识符划分，见附图3所示，其将CAN2.0b或CAN_FD协议扩展帧格式中29位CAN标识符划分为六段，即:优先级段、子段1、子段3、设备编号段、功能码段和子段2，其中:子段1、子段2和子段3排列拼接组合构成寄存器基地址段，各段组成说明如下:

优先级段:　由CAN标识符的最高位第28位组成;
子段1:　由CAN标识符的第27位组成;
子段3:　由CAN标识符的第26～19位组成;
设备编号段:由CAN标识符的第18～12位组成;
功能码段:　由CAN标识符的第11～7位组成;
子段 2:　由CAN标识符的第6～0位组成;

寄存器基地址段:由子段1、子段2和子段3排列拼接组合而成，即:由CAN标识符的第27位、6～0 位和 26～19位组成十六位的寄存器基地址段。



附图 3 传递过程变量的 CAN 标识符划分

当 CAN 标识符的第 27 位(即:寄存器基地址段的最高位)为 0 时，寄存器基地址空间指向过程变量空间(即:0x0000～0x7FFF)，由于十六位寄存器基地址段的低八位(26～19 位)处于 CAN 标识符的较高端 MSB，而其次高七位(6～0 位)处于 CAN 标识符的低端 LSB，由此过程变量寄存器空间变量的竞争能力由其十六位寄存器基地址段的低八位决定，从而在过程变量寄存器地址空间形成了 128 个等价竞争层，每层包含 256 个地址空间，在每层相同位置上的过程变量具有相同的总线竞争能力，在同层中，低地址变量的竞争能力优于高地址变量。

传递非过程变量的 CAN 标识符划分，见附图 4 所示，其将 CAN2.0b 或 CAN_FD 协议扩展帧格式中 29 位 CAN 标识符划分

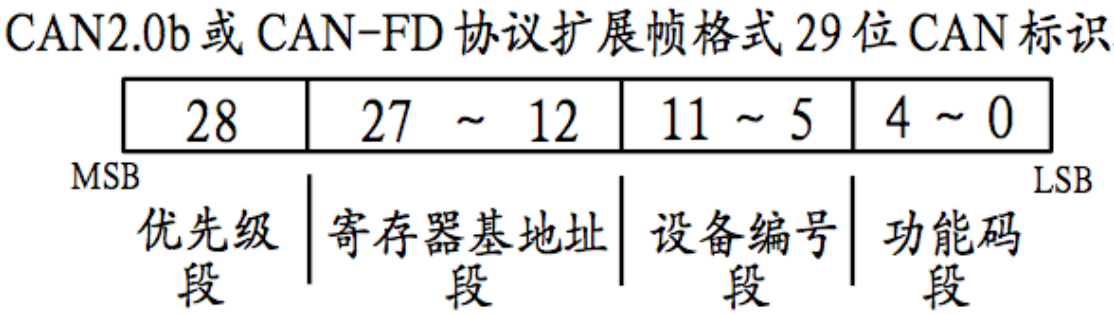
为四段，即:优先级段、寄存器基地址段、设备编号段和功能码段，各段组成说明如下:

优先级段:　　由CAN标识符的最高位第28位组成;

寄存器基地址段:由CAN标识符第27～12位组成;

设备编号段:　　由CAN标识符的第11～5位组成;

功能码段:　　由CAN标识符的第4～0位组成;



附图4 传递非过程变量的 CAN 标识符划分

一般非过程变量的实时性要求不高，在应用层通常适用主机-从机模型，主机和从机间采用问答方式通讯，变量分段管理与其总线竞争能力没有突出的矛盾，因此从CAN标识符的高端直接截取十六位(27～12 位)构成寄存器基地址段，当CAN标识符的第27位(即: 寄存器基地址段的最高位)为1时，寄存器基地址空间指向非过程变量空间(即:0x8000～0xFFFF)，

非过程变量的总线竞争能力整体低于过程变量，这是应用层所期望的，由于CAN标识符的高七位不能连续出现高电平1，当优先级位(28位)为1时，非过程变量实际有效的地址空间为0x8000～0xFBFF，也就是在0xF000段的尾部集中有1024个地址空间(0xFC00～0xFFFF)不能使用，如果对非过程变量采用和过程变量一样的寄存器拼接组合的方法构成寄存器基地址段，势必将这1024个不可用地址空间分散到各等价层中，这对非过程变量寄存器空间的使用和管理无益，因此应用层对过程变量和非过程变量分别采用不同的CAN标识符划分进行总线传递是非常必要的。

TTCANopen应用层协议对CAN标识符的划分及其物理含义的分配，如:优先级段、寄存器基地址段、设备编号段、功能码段都具有明确的应用层协议指令含义，据此设计的应用层协议具有很好的直读性和开放性，使应用层协议易于解析，便于用户在框架内构建适合其使用环境的应用层子协议。

TTCANopen应用层协议以连续寄存器空间为信息载体，其应用层协议指令是对该连续寄存器空间的读写访问，由此使其能够天然的兼容CAN-FD协议对数据场长度的扩充。

附录2: TTCANopen应用层协议
设备内部多条指令的发送规则

通常CAN通讯协议只规范了多个设备在总线上同时发送指令时的竞争顺序，而对于单个设备内部产生的多条指令的发送顺序却很少涉及。

在纯主从模式下，设备内部产生多条指令的可能性并不大，相关问题很难暴露，而当系统发展演绎到全触发方式后情况就不同了，一个设备可能具有多个触发变量，它们可以是周期触发、变化触发、越界触发等等，而总线的发送速率是有限的，且同时还存在其他设备对总线的竞争，因此在同一设备中就有可能积累多条指令等待发送。

通常设备会默认以“先生优势”规则处理这些指令，即先产生的指令优先发送，这样处理的最大好处是指令动作先后逻辑规范，这对那些具有先后关联的系列指令尤为重要，但事物总是两方面的，这种“先生优势”规则，会延迟和阻碍后产生的高优先级指令的发送，如：告警指令，尤其是当总线上存在竞争时，情况尤为严重，如:当设备指令发送端口被一条先生成的低优先级指令占据时，其在总线上的竞争处于劣势，从而使该设备中后产生的告警指令无法参与竞争而

被推迟发送，这种延迟有时可能是致命的。 因此我们需要重新规范设备内部多条指令的发送规则，最粗暴简单的方法就是对这多条指令进行优先级排队，将ID字小的指令排在前面，每当设备产生一条新指令时都要重新进行一次指令排序(注意:这种重新排序应当包括那条已经处在发送端口的未发指令)，这样设备便具有了最优的发送竞争力，但同时可能会破坏前面所说的某些指令的先后逻辑关联，可能使系统运转出现偏差。

为此TTCANopen应用层协议做出如下规范，当一设备同时具有多条指令等待发送时：

1)优先级位为0的指令优先于优先级位为1的指令发送;

2)优先级位相同的指令按其产生的先后顺序发送。

虽然只有短短的两条规定，但它却完成了其它应用层协议很难规范完成的内容，而这对于 TTCANopen协议来说却是自然天成的，它的第一条规定确保了低优先级(1)的指令不能阻碍高优先级(0)指令的发送，使设备中诸如告警指令得以顺利抢先通过总线发送；第二条规定保障了具有先后逻辑关系的系列指令的顺利完成，因为我们一般不会将这些相关指令定义在不同的优先级位上。

更多内容请关注：www.ttcanopen.com

免责声明

本着对用户负责的精神，说明书尽可能翔实正确的对产品功能和指标进行描述，但不能保证没有遗漏和错误。另外，随着产品不断升级和完善，说明书内容也具有一定的时效性和适用性，可能会在没有特别通知的情况下，进行修改和补充，请用户及时关注www.ttcanopen.com网站中相关信息。感谢您的包容与支持。

补充说明：

1，GD公司现已经将GD32E103系列，更改为GD32C103系列。

附录3:　升级到v4.0新增的四项功能

1，**UID读取指令**，读取到单片机序列号中的一个32位数，用户可以通过这个UID区分单片机。

指令格式：AA 66 FF 00 01 02 DD

发送该指令后，模块会以字符串形式返回一个32位数，

如：26533302

2，配置模块时间指令，在TTCANopen应用层协议中，模块时间是由心跳器每秒发送一次主心跳完成总线时间同步的。但对于没有使用TTCANopen应用层协议的用户模块，时间是从上电时刻0开始计数的。有用户提出需求希望模块时间能够与PC机基本同步，为此新增了该指令。

指令格式：AA 66 FF 07 05 01 tt tt tt tt DD

其中， tt tt tt tt 是一个32位无符号数，低字节在前高字节在后，是一天中以毫秒为计时单位的时刻值。

用户可以先读取PC机当前的时间，并将其转换为毫秒计数，发送该指令，这样就可以将模块的时间配置为与PC机基本同步。用户可以通过周期性的发送该指令，去修正同步时间的漂移。

3， **CANFD数据域不加速发送**，在v3.7版本，默认CANFD发送数据域是加速的。但在实际应用中，有少数用户在使用数据域不加速的方式发送CANFD帧，为了使模块功能更加完备，满足更多用户的应用需求，在v4.0版中添加了该功能。

发送方法：在发送指令序列中，将表述CANFD ID字的字节序列的最高位置1.

举例：常规发送指令如：（默认数据域加速）

AA 55 01 23 08 11 22 33 44 55 66 77 88 DD

指令中，01 23 是ID字，将ID字最高位置1后指令为：

AA 55 81 23 08 11 22 33 44 55 66 77 88 DD

再例：

AA CC 12 34 56 78 04 11 22 33 44 DD

指令中，12 34 56 78 是ID字，将ID字最高位置1后指令为：

AA CC 92 34 56 78 04 11 22 33 44 DD

实际发送的CANFD ID字，标准帧是11位，扩展帧是29位的，都没有用到16位或32位字节ID字的最高位，这里最高位置1，只是用于标识发送不加速，不会影响真实发送的ID字。

另外，鉴于不加速帧传输的可靠性更好，模块作为心跳器时，发送心跳帧，使用CANFD数据域不加速方式发送。

4，新增6个CAN ID过滤，最初设计USB转CANFD时，由于认为USB端连接的是PC机，将ID过滤留给功能强大的PC机上位机软件去完成比较妥当，后来模块加入了TTL串口、SPI和网口，连接的可能是另一个单片机，尤其是TTL串口，通信速率过低，因此在v4.0版本模块新添加了6个ID过滤。

这6个ID过滤没有设计屏蔽掩码，只是单纯的选则6个ID，具体指令如下：

AA 66 FF 10 18 xx xx xx xx yy yy yy yy zz zz zz zz mm mm mm mm nn nn nn nn kk kk kk kk DD

当然您也可以配置少于6个ID过滤，如：只配置2个ID过滤，

AA 66 FF 10 08 xx xx xx xx yy yy yy yy DD

其中，xx xx xx xx 和 yy yy yy yy 等，都是低字节在前高字节在后的无符号32位数，代表需要选取的指定ID值，这样模块就会只传递这些指定ID的帧，其它被忽略。

如：您只想看ID字等于0123的帧，就需要发送配置指令：

AA 66 FF 10 04 23 01 00 00 DD

注意：无论是标准帧还是扩展帧都使用32位过滤字，在上面的指令发出后，您可以看到ID字是0123的标准帧，也可以看到ID字是00000123的扩展帧。

另外，需要注意的是过滤ID字必须从地址FF 10开始配置起，而且第一个过滤ID字不能是0（即xx xx xx xx不能是0）。

如果第一个过滤ID字是0，模块将忽略ID过滤功能，将显示全部所收到的帧。