

UARTCAN 用户手册

UARTCAN 是一款 CAN 接口调试工具。可以使 CAN 接口调试与串口调试一样方便。配合低成本的 USBTTL 数据线，可以将 UARTCAN 转化成一款 USBCAN 设备。UARTCAN 同样提供 USB 接口版本。

UARTCAN 具备独特的 TERM 工作模式，借助超级终端等终端模拟软件，可以在 PC 上轻松进行 CAN 接口调试。用户只需熟悉少数几个命令，便可以轻松上手。

UARTCAN 同时具备 PKT 工作模式，该工作模式下，所有通讯数据按照 16 字节定长数据包来组织，CRC 数据校验保证了数据完整性，特别适合用户编程处理。该模式下用户还可以使用 PC 上常见的 CANTest 软件来进行调试。

UARTCAN 具备 BRIDGE 工作模式，该模式下使用简单的封包结构，便用用户编程处理。

UARTCAN 主要特性如下：

- 32 位 ARM 处理器 72M 主频处理能力强大
- TERM 工作模式无需专用上位机软件
- PKT 工作模式便于用户二次开发
- CRC 校验保证数据完整性
- BRIDGE 模式支持透明转发
- 同时提供 TTL 版本和 USB 版本满足不同需求
- 体积小便携使用方便
- 支持固件升级
- 成本低廉

1 外观与接口

1.1 端子布局

主板端子布局如图 1 所示。USB 版本功能与 TTL 版本完全相同，见图 2，固件可通刷，唯一差别是 TTL 接口更换为 USB 接口。

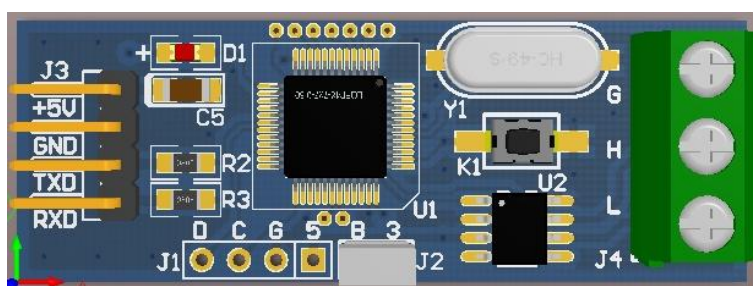


图 1 TTL 版本主板端子布局

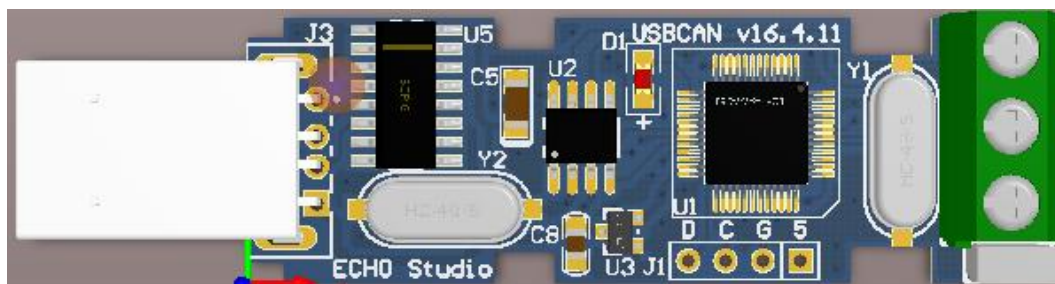


图 2 USB 版本主板端子布局

1.2 端子说明

J3 为 TTL 接口，四个信号分别为+5V、GND、TXD、RXD。

+5V 和 GND 两个网络同时负责给 UARTCAN 模块进行供电。

J4 为 CAN 接口，三个信号分别为 GND、CANH、CANL。

K1 为用户按键。

D1 为指示 LED。

2 快速入门

- 1) 将 UARTCAN 与 USB 转 TTL 线连接，注意 RXD 与 TXD 交叉连接。（USBCAN 直接将 USB 插入电脑，转到第 3 步）
- 2) 将 USB 转 TTL 连接电脑，安装驱动程序，到设备管理器记下对应串口号。
- 3) 打开“超级终端”软件，新建连接见图 3，点击确定，使用 USB 转 TTL 对应的 COM 口，见图 4，按照图 5 设置 COM 口参数，点击“确定”按钮。
- 4) 输入 help 回车，查看在线帮助。
- 5) 输入“std 44 11223344”回车，发送 ID 为 0x44，四个字节 0x11、0x22、0x33、0x44 标准数据帧。
- 6) 输入“ext 44 11223344”回车，发送 ID 为 0x44，四个字节 0x11、0x22、0x33、0x44 扩展数据帧。
- 7) CAN 接口收到的数据会在“超级终端”中实时显示。
- 8) 更多功能请参考本手册其它章节。



图 3 超级终端新建连接



图 4 连接使用 USB 转 TTL 对应 COM 口

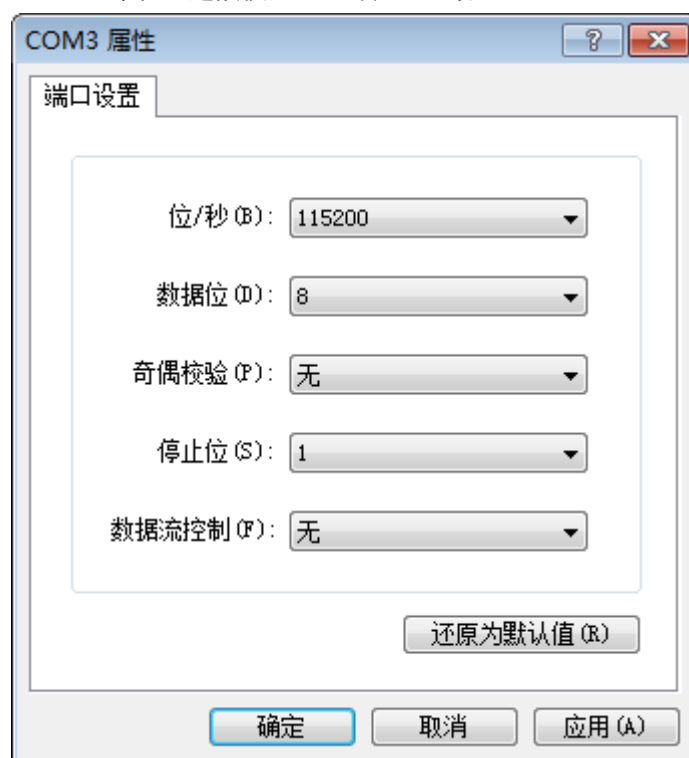


图 5 COM 口参数

3 性能指标

3.1 CAN 接口指标

波特率最低 40kbps，最高 1Mbps，支持的波特率如下：40k、50k、60k、80k、90k、100k、125k、150k、200k、250k、300k、400k、500k、600k、800k、900k、1M。

3.2 UART 接口参数

UART 接口为标准 TTL 接口。

VCC 供电为 5V，TXD 和 RXD 可以接受 5V 或者 3.3V 电平。

UART 接口支持常见波特率，波特率范围 2400bps-921600bps，默认波特率 115200。1 位起始位，8 位数据位，1 位停止位，无校验，参考图 5。

4 工作模式

UARTCAN 模块支持三种工作模式：TERM 模式，PKT 模式，BRIDGE 模式，默认为 TERM 模式，用户可修改默认工作模式。见图 6。

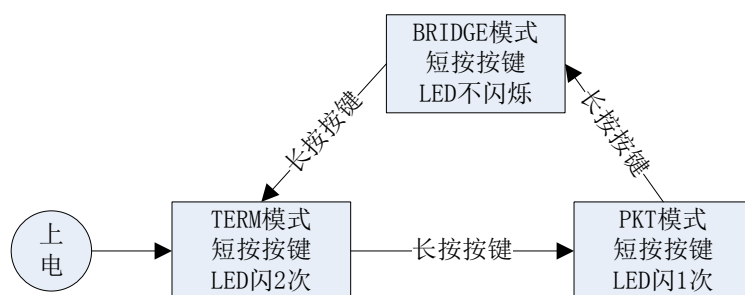


图 6 四种工作模式切换过程

短按按键以后 LED 闪烁指示工作模式。无闪烁表示 BRIDGE 模式，闪烁一次表示 PKT 模式，闪烁两次表示 TERM 模式。上电后 LED 会通过闪烁指示当前工作模式。

通过长按按键可以在 TERM、PKT、BRIDGE 三种工作模式之间进行切换，切换以后自动记忆当前工作模式。

4.1 TERM 模式

在 TERM 模式下，用户可以使用命令行终端方式收发 CAN 数据。UARTCAN 提供了一系列命令操作 CAN 接口数据。

该模式下，PC 端使用“超级终端”软件控制 UARTCAN，无需专门上位机软件。

相关命令可参考文档：《UARTCAN_CmdRef.pdf》

4.2 PKT 模式

PKT 模式下，CAN 接口数据采用 16 字节固定的长度进行封包，然后通过异步串口传输。固定长度的封包便于软件编程处理。串口网络传输数据带 CRC 校验，提高了数据完整性。

PKT 模式下可以借助各种串口调试助手软件调试 CAN 网络。

通过使用专用的驱动，PKT 模式下也可以使用 CANTest 等常见 CAN 网络调试软件。

详情可参考 6。

4.3 BRIDGE 模式

BRIDGE 模式下，UARTCAN 将 UART 接口上收到的字节封装为固定 ID 的单字节标准帧在 CAN 接口上传输。同时将 CAN 接口上收到的固定帧 ID 的单字节标准帧解包以后转发到 UART 接口。

BRIDGE 模式下 CAN 接口统一使用标准帧，帧 ID 固定为 0x87，数据长度为 1 个字节。

BRIDGE 模式可以理解为桥接、透明转发。

5 其它常用操作

5.1 固件升级

使用超级终端连接串口命令行以后，执行 `reboot 900` 命令，延时 900ms 重启，然后马上按住键盘上的字母 ‘e’，设备重启进入 XBOOT，执行 `ymodem` 命令，自动擦除原固件，然后选择要升级的固件，协议选择 ‘ymodem’ 等待固件下载完毕即可。

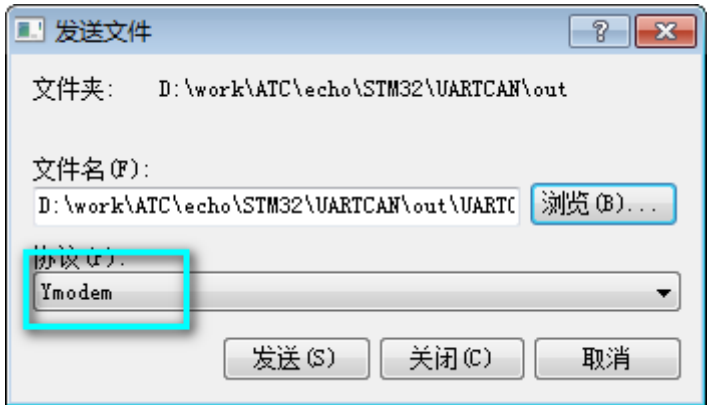


图 7 选择固件文件

升级过程见**错误！未找到引用源。**，`ymodem` 命令会尝试先擦除旧固件，然后写入新固件，升级成功以后给出提示信息。

执行 `ymodem` 命令显示字符 “C” 时，可以按 “A” 键取消当前升级过程。

5.2 记录数据

利用超级终端的 “捕获文字” 功能，菜单 “传送(T)->捕获文字(C)”。

5.3 恢复出厂设置

按住按键上电，UARTCAN 恢复出厂设置。

TERM 模式下，执行 `param restore` 命令可以加载出厂设置，继续执行 `param save` 命令保存出厂设置。

6 PKT 模式通讯协议

PKT 模式下使用固定长度的数据包进行通信，适用于上位机软件和 MCU 处理。

6.1 封包格式

数据包固定 16 字节，按字节依次编号为 0-15，分为 5 个区域：设备地址、功能码、ID 区、DATA 区、校验区。见图 8。

设备地址 一字节	功能码 一字节	ID区 四字节	DATA区 八字节	校验区 二字节
-------------	------------	------------	--------------	------------

图 8 PKT 模式 16 字节封包组成

6.1.1 设备地址

设备地址为 BYTE0，一个字节，默认地址 0x01，有效范围 1-247，需要和 UARTCAN 实际地址对应。建议使用默认地址 0x01。

UARTCAN 与上位机通讯采用“请求—应答”方式，上位机发送一个请求数据包，UARTCAN 根据请求数据包内容进行响应操作，然后进行应答，请求与应答数据包采用相同的固定 16 字节封包格式。

6.1.2 功能码

功能码为 BYTE1，一个字节。功能码定义了一个 PKT 模式数据包的功能，ID 区和 DATA 区数据的意义与功能码密切相关。PKT 模式下支持的功能码见表 1。

表 1 功能码定义

功能码	名称	描述	ID 区与 DATA 区
0x00	IDLE	UARTCAN 不做任何响应	无关，建议填 0
0x01	ECHO	数据包回显	收到数据原样返回
0x02	RDEE	保留	
0x03	WREE	保留	
0x04	LDEE	保留	
0x05	SAEE	保留	
0x06	REBOOT	重启 UARTCAN	无关，建议填 0
0x07	DEVINFO	获取 UARTCAN 设备信息	返回数据在 DATA 区，DATA0 和 DATA1 为 MCU 固件版本，DATA2 和 DATA3 为 CAN 波特率
0x08	DEVSN	获取 UARTCAN 设备序列号	返回 12 字节序列号，占据 ID 区和 DATA 区
0x09	LED	控制主板 LED	DATA 区首字节非零点亮，零熄灭
0x0A	INITCAN	初始化 CAN	发送数据占用 ID 区，ID0 和 ID1 为 CAN 接口波特率，ID2 为 CAN 工作模式
0x30-0x38	STD	发送标准帧，低 4 位为数据长度	ID 区共 32 位，靠右对齐；DATA 区为实际数据
0x40	STDRMT	发送标准远程帧	ID 区共 32 位，靠右对齐
0x50-0x58	EXT	发送扩展帧，低 4 位为数据长度	ID 区共 32 位，靠右对齐；DATA 区为实际数据
0x60	EXTRMT	发送扩展远程帧	ID 区共 32 位，靠右对齐

6.1.3 ID 区

ID 区共 4 个字节 32 位，数据采用 Big Endian，高字节在前，低字节在后。
无论 11 位的 标准帧还是 29 位扩展帧，数据均靠右对齐。
非 CAN 数据帧的情况下，ID 区也可以用来存放数据。

6.1.4 DATA 区

DATA 区共 8 个字节，用来存储 CAN 数据帧最大 8 个字节。如果数据帧不够 8 个字节，优先使用编号靠

前字节，剩余字节用 0x00 补齐。

6.1.5 校验区

校验区共 2 个字节，16 位，采用 Little Endian，即低字节在前，高字节在后。校验算法采用 CRC 校验，具体请参考 MODBUS 协议 CRC 校验码算法。

一种具体的实现请参考图 9。

```
uint16_t ModbusCRC(const uint8_t *data, uint16_t dataLen)
{
    uint16_t i = 0, j = 0, crc=0xFFFF, flag=0;

    for (i = 0; i < dataLen; i++)
    {
        crc ^= data[i];
        for (j = 0; j < 8; j++)
        {
            flag = crc & 0x0001;
            crc >>= 1;
            if (flag)
            {
                crc ^= 0xA001;
            }
        }
    }
    return crc;
}
```

图 9 ModbusCRC 参考算法实现

6.2 通讯举例

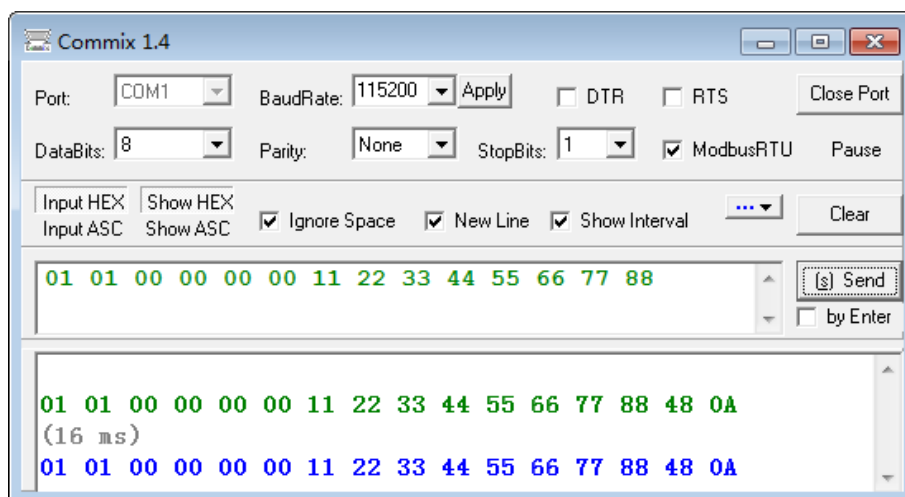


图 10 ECHO 功能码通讯测试

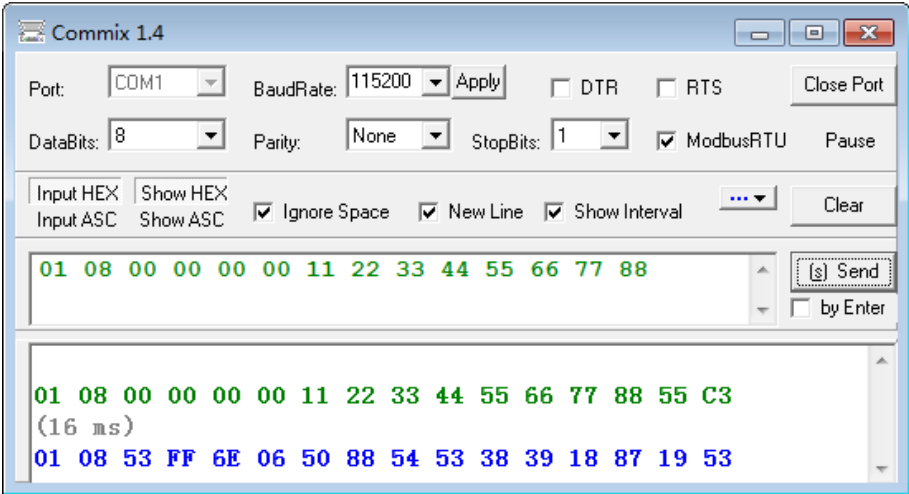


图 11 DEVSN 功能码通讯测试

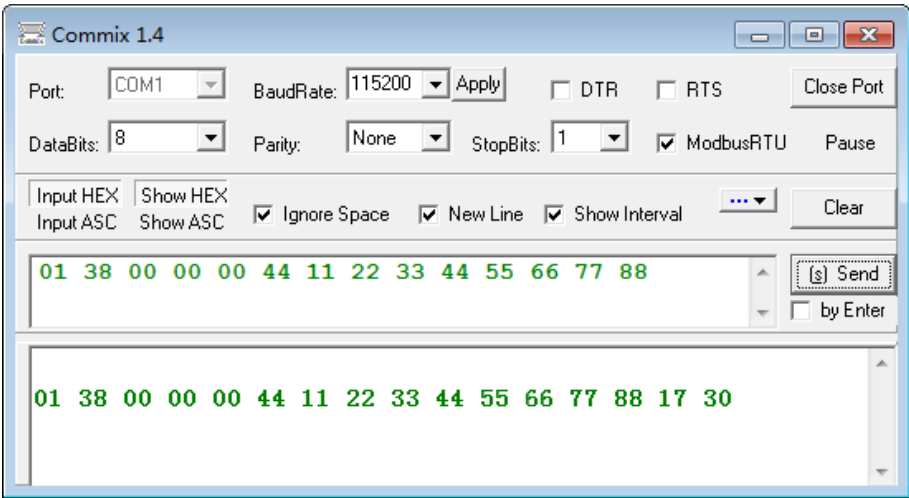


图 12 发送 ID 为 0x44 8 个数据字节的标准帧

6.3 CANTest 使用说明

CANTest 软件是广州致远电子开发的一款 CAN 调试软件，使用广泛。

通过替换 DLL 文件，UARTCAN 也可以用 CANTest 软件作为上位机软件。将 CANTest 安装目录下的 ControlCAN.dll 文件重命名为 ControlCAN_zlg.dll，然后将 UARTCAN 提供的 ControlCAN.dll 文件拷贝到 CANTest 目录下，启动时设备选择 USBCAN1。

kernelDlls	2016/4/1 18:44
language	2016/4/1 18:44
plugin	2016/4/1 18:44
update	2016/4/1 18:44
cantest.exe	2013/4/7 15:24
cantest.ini	2016/4/6 20:46
cfg.ini	2013/3/5 16:38
ControlCAN.dll	2016/1/7 22:50
ControlCAN_zlg.dll	2012/7/12 17:09
libcurl.dll	2011/5/17 14:47
mfc80.dll	2010/7/27 18:44

图 13 CANTest 软件 DLL 替换

CANTest 软件仅供学习研究使用，波特率最低 50k，不支持滤波器设置，只支持正常工作模式。使用 UARTCAN 完整功能请使用 TERM 模式。

7 更新记录

更新日期	更新类型	更新人	更新内容
2015-6-19	A	Echo	新建文档
2016-5-2	A	Echo	完成文档初稿
2016-7-2	A	Echo	增加 BRIDGE 模式说明
2020-8-1	M	Echo	根据 v20.8.1 固件更新，拆分 TERM 模式命令文档

注:

M-->修改

A -->添加

项目主页: <https://github.com/xjtuecho/UARTCAN>

国内镜像: <https://gitee.com/xjtuecho/UARTCAN>