修改记录

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

注:

M-->修改

A -->添加

请使用 PDF 书签阅读本文档，快速定位所需内容！

更多信息请关注

作者博客：http://blog.sina.com.cn/xjtuecho

作者微博：http://weibo.com/eth0

作者淘宝：http://shop114445313.taobao.com/

作者 github 主页：https://github.com/xjtuecho/

最新文档和设备固件请访问 github 项目主页：https://github.com/xjtuecho/UARTCAN/

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 USB 转 TTL 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。用户只需要学习少数几个命令，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 主要特性如下：

- 32 位 ARM 处理器 72M 主频处理能力强大
- 独特的 TERM 工作模式无需专用上位机软件
- 精心设计的 PKT 工作模式便于用户二次开发
- CRC 校验技术避免数据出错
- 兼容周立功 CANTest 软件
- 同时提供 TTL 版本和 USB 版本满足不同需求
- 体积小巧便携使用方便
- 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，固件可通刷，唯一差别是 TTL 接口更换为 USB 接口。

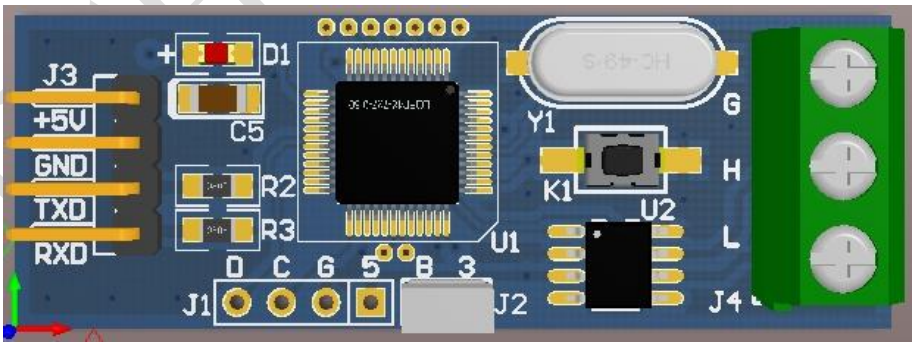

图 1 TTL 版本主板端子布局

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。

短按 K1，D1 闪烁指示通讯协议。闪烁一次表示 PKT 模式，闪烁两次表示 TERM 模式。长按 K1 可以在两种模式之前切换，切换结果立即生效并保存。

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 2，点击确定，使用 USB 转 TTL 对应的 COM 口，见图 3，按照图 4 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） 更多功能请仔细阅读本手册。

图 2 超级终端新建连接

图 3 连接使用 USB 转 TTL 对应 COM 口

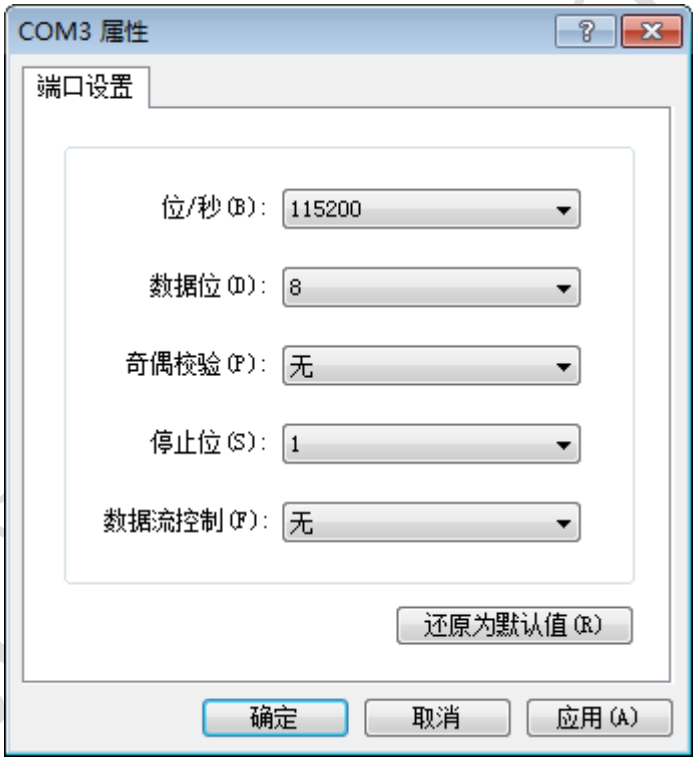

图 4 COM 口参数

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 4。

4 工作模式

UARTCAN 模块支持三种工作模式：TERM 模式，PKT 模式，BOOT 模式，默认上电后自动进入 TERM 模式。见图 5。

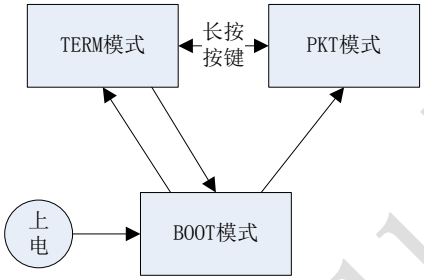

图 5 三种工作模式

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 5 见下方。

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 见下方。

4.3 BOOT 模式

BOOT 模式用来升级用户程序，或者修改内部 EEPROM 的内容，控制设备运行参数。

进入 BOOT 方式有两种，分别如下：

1） 设置好超级终端参数连接 UARTCAN，按住键盘上的‘e’按键，给 UARTCAN 上电。进入 BOOT 模式。

2） TERM 模式下，执行 reboot 1000 命令，延时 1 秒后重启，马上按住键盘上的‘e’按键，重启后进入 BOOT 模式，见图 6。

```
version
 UARTCAN v16.4.7 Flash:64kB SN:53FF6E065088545338391887
 ECHO Studio <echo.xjtu@gmail.com>. All Rights Reserved.
reboot 1000
按eeeeeeeeeeeeeeeeeeee reboot UARTCAN shell ...
 XBOOT for STM32F10x v16.3.30 Flash:64kB SN:53FF6E065088545338391887
 ECHO Studio <echo.xjtu@gmail.com>. All Rights Reserved.
eeeeeee
version

 XBOOT for STM32F10x v16.3.30 Flash:64kB SN:53FF6E065088545338391887
 ECHO Studio <echo.xjtu@gmail.com>. All Rights Reserved.
```

图 6 TERM 模式下进入 BOOT 模式操作流程

BOOT 模式下运行 UARTCAN 的 bootloader 程序，可更新 UARTCAN 设备固件。

4.4 模式间切换

TERM 模式和 PKT 模式可以通过长按按键进行切换，切换以后自动记忆当前工作模式。通过短按按键查看当前工作模式。

BOOT 模式进入方式见 4.3 节。

5 TERM 模式控制命令

5.1 can

显示、设置 CAN 接口参数。

命令格式：can [baud|mode|init] CAN interface params setup.

不带参数的 can 命令显示当前 CAN 接口状态，命令输出见图 7。

```
can
 can [baud|mode|init] CAN interface params setup.
 CAN  baud = 100kbps mode = 0 Normal
 RX   lost = 0 success = 0 show = 1
 TX   lost = 0 success = 0 repeat = 0 interval = 0ms
```

图 7 不带参数 can 命令输出

输出结果分四行：

第一行为命令格式。

第二行为 CAN 接口波特率与工作模式设置，可选的工作模式见表 1。

表 1 CAN 接口工作模式

工作模式	取值	备注
Normal	0	正常模式（常用）
LoopBack	1	环回模式
Silent	2	静默模式
Silent_LoopBack	3	环回静默模式

第三行为接收 CAN 数据帧信息，lost 和 success 分别表示接收失败和成功的 CAN 数据帧数量，show=1 表示显示接收到的数据帧，show=0 表示不显示接收到的数据帧，可以使用 ctrl+p 按键切换。

第四行为发送的 CAN 数据帧信息，lost 和 success 分别表示发送失败和成功的 CAN 数据帧数量，repeat 表示重复发送数量，interval 表示发送间隔，repeat 和 interval 可以用过 repeat 命令设置，用来重复发送相同的数据帧，详见 repeat 命令。

5.1.1 can baud

baud 子命令可以设置 CAN 接口波特率，见图 8，使用“can baud”命令查看当前 CAN 接口波特率，使用“can baud 500”将 CAN 接口波特率设置为 500kbps。

```
can baud
 can baud [baud in kbps] set CAN baudrate.
 CAN baud = 100kbps
can baud 500
 set CAN baud to 500kbps, save and init to apply new baud.
```

图 8 can baud 设置 CAN 接口波特率

修改 CAN 接口波特率以后，需要执行“param save”命令保存参数，并且执行“can init”命令重新初始化 CAN 接口使能新波特率。

5.1.2 can mode

mode 子命令可以设置 CAN 接口工作模式，见图 9。使用“can mode”命令查看当前 CAN 接口工作模式，使用“can mode 1”将 CAN 接口工作模式设置为环回模式。

```
can mode
 can mode [0|1|2|3] set CAN work mode.
 CAN mode = 0 Normal
can mode 1
 set CAN mode to 1 LoopBack, save and init to apply new mode.
```

图 9 can mode 设置 CAN 接口工作模式

修改 CAN 接口工作模式以后，需要执行“param save”命令保存参数，并且执行“can init”命令重新初始化 CAN 接口使能新工作模式。

5.1.3 can init

修改 CAN 接口参数以后，需要执行“can init”命令重新初始化 CAN 接口，命令输出见图 10。

```
can init
 init CAN interface according to user config...
```

图 10 can init 命令输出

5.2 filter

显示、设置 CAN 接口滤波参数。(该命令使用复杂，如无特殊需要，建议使用默认设置)。

命令格式：filter [0-6] [mode|scale|act|r0|r1] [param] CAN filter setup.

UARTCAN 支持最多 7 个滤波器设置，编号 0-6，不带参数的 filter 命令显示当前 CAN 接口滤波器设置，见图 11。

mode 取值为 idlist 或者 idmask；

scale 取值 32 或者 16；

act 取值 1 或者 0；

R0 和 R1 为两个寄存器，具体意义由 mode 和 scale 取值决定。

```
filter
 filter [0-6] [mode|scale|act|r0|r1] [param] CAN filter setup.
 filter 0 MODE=idmask SCALE=32 ACT=1 R0=0x00000000 R1=0x00000000
 filter 1 MODE=idmask SCALE=32 ACT=0 R0=0x00000000 R1=0x00000000
 filter 2 MODE=idmask SCALE=32 ACT=0 R0=0x00000000 R1=0x00000000
 filter 3 MODE=idmask SCALE=32 ACT=0 R0=0x00000000 R1=0x00000000
 filter 4 MODE=idmask SCALE=32 ACT=0 R0=0x00000000 R1=0x00000000
 filter 5 MODE=idmask SCALE=32 ACT=0 R0=0x00000000 R1=0x00000000
 filter 6 MODE=idmask SCALE=32 ACT=0 R0=0x00000000 R1=0x00000000
```

图 11 不带参数的 filter 命令输出

第一个参数为滤波器编号，带一个参数的 filter 命令输出当前滤波器设置，如:“filter 0”命令输出当前滤波器 0 的设置；“filter 1”命令输出当前滤波器 1 的设置，见图 12。可以使用该命令依次查看 7 个滤波器设置。

```
filter 0
 filter 0 MODE=idmask SCALE=32 ACT=1 R0=0x00000000 R1=0x00000000
filter 1
 filter 1 MODE=idmask SCALE=32 ACT=0 R0=0x00000000 R1=0x00000000
```

图 12 带一个参数查看当前滤波器信息

所有命令第一个参数为滤波器编号，UARTCAN 支持编号 0-6 共 7 个滤波器，为了便于叙述，本节以 0 号滤波器为例。

5.2.1 filter 0 mode

设置滤波器滤波模式。

“filter 0 mode 0”设置为 0 号滤波器为 idmask 模式，“filter 0 mode 1”设置 0 号滤波器为 idlist 模式。具体过程见图 13。滤波模式会影响其它参数的设置。

```
filter 0
 filter 0 MODE=idmask SCALE=32 ACT=1 R0=0x00000000 R1=0x00000000
filter 0 mode 1
 filter 0 MODE=idlist SCALE=32 ACT=1 R0=0x00000000 R1=0x00000000
filter 0 mode 0
 filter 0 MODE=idmask SCALE=32 ACT=1 R0=0x00000000 R1=0x00000000
```

图 13 filter 0 mode 命令使用举例

5.2.2 filter 0 scale

设置滤波器位宽(16 位或 32 位)。

“filter 0 scale 0”设置 0 号滤波器为 16 位，“filter 0 scale 1”设置 0 号滤波器为 32 位。见图 14。

```
filter 0 scale 0
 filter 0 MODE=idmask SCALE=16 ACT=1 R0=0x00000000 R1=0x00000000
filter 0 scale 1
 filter 0 MODE=idmask SCALE=32 ACT=1 R0=0x00000000 R1=0x00000000
filter 0 scale 0
 filter 0 MODE=idmask SCALE=16 ACT=1 R0=0x00000000 R1=0x00000000
```

图 14 filter 0 scale 命令使用举例

5.2.3 filter 0 act

使能或者禁止滤波器。

“filter 0 act 0”禁止 0 号滤波器，“filter 0 act 1”使能 0 号滤波器。见图 15。

注意，CAN 接口必须使能至少一个滤波器才能顺利收到数据。

```
filter 0 act 0
 filter 0 MODE=idmask SCALE=32 ACT=0 R0=0x00000000 R1=0x00000000
filter 0 act 1
 filter 0 MODE=idmask SCALE=32 ACT=1 R0=0x00000000 R1=0x00000000
```

图 15 filter 0 act 命令使用举例

5.2.4 filter 0 r0

每一组滤波器工作模式分为 idmask 和 idlist 两种，每一组滤波器位宽可以为 32 位或者 16 位，因此存在 4 种组合方式：

1）1 组 32 位 idmask 滤波器

2）2 组 32 位 idlist 滤波器

3）2 组 16 位 idmask 滤波器

4）4 组 16 位 idlist 滤波器

四种组合方式及其 ID 映射见图 16。

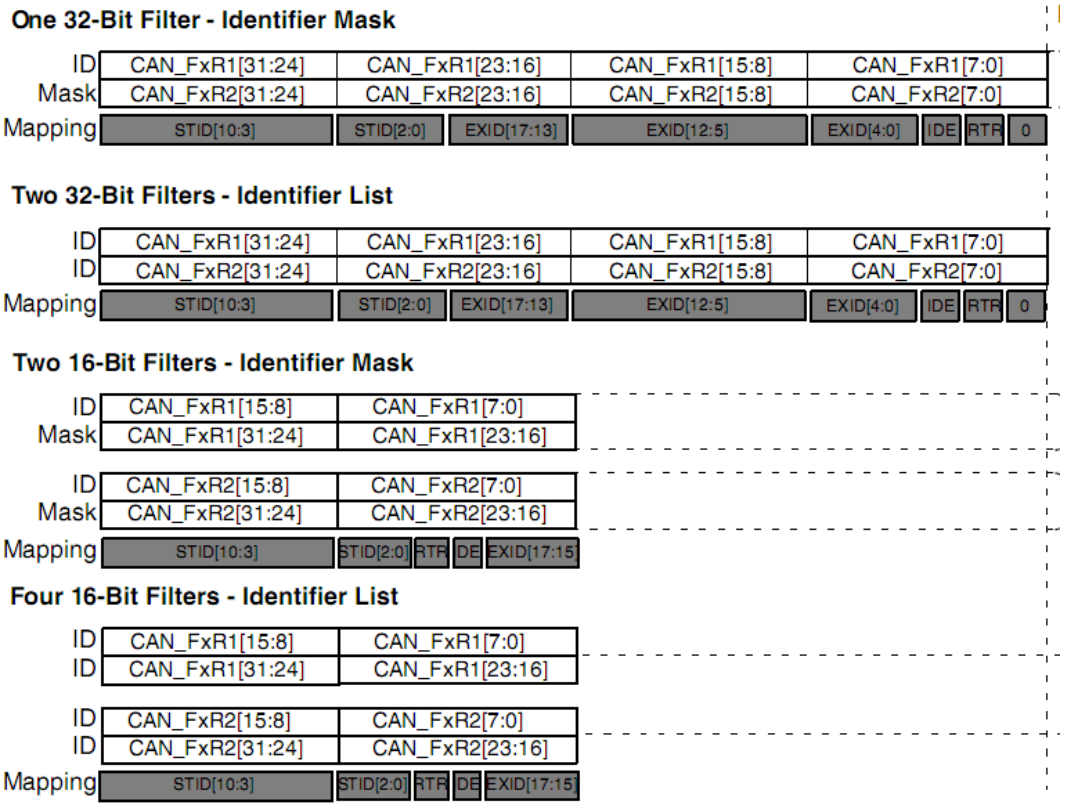

图 16 不同组合方式下滤波器 ID 映射

用户需要按照实际滤波需求计算出 r0，然后使用“filter 0 r0”命令设置 r0 寄存器。

```
filter 0 r0 11
 filter 0 MODE=idmask SCALE=32 ACT=1 R0=0x00000011 R1=0x00000000
filter 0 r1 22
 filter 0 MODE=idmask SCALE=32 ACT=1 R0=0x00000011 R1=0x00000022
```

图 17 r0 与 r1 设置命令

5.2.5 filter 0 r1

设置 r1 寄存器，与“filter 0 r0”配合使用，参考图 17。

以只接收 ID 为 0xAA 的扩展数据帧为例，设置如下：

使用 idlist 方式，32 位滤波器，r1 设置为(0xAA<<3)+4=0x554，设置过程与结果见图 18。

```
filter 0 r1 554
 filter 0 MODE=idlist SCALE=32 ACT=1 R0=0x00000000 R1=0x00000554
param save
 Save parameters to EEPROM...
can init
 init CAN interface according to user config...
 RX:00000008 DATA MSG ID=0x000000AA LEN=8  DATA=11 22 33 44 55 66 77 88
 RX:00000009 DATA MSG ID=0x000000AA LEN=8  DATA=11 22 33 44 55 66 77 88
```

图 18 滤波器设置实例

更多滤波器设置方法请参考 STM32 用户手册 CAN 接口部分。

5.3 std

通过 CAN 接口发送标准帧。

命令格式：std [hex ID] [hex DATA | remote].

```
std 44 12345678
 TX:00000002 DATA MSG ID=0x0044 DATA=12 34 56 78
std 44 remote
 TX:00000003 REMOTE MSG ID=0x0044
std 33 00112233445566788
 TX:00000004 DATA MSG ID=0x0033 DATA=00 11 22 33 44 55 66 77
```

图 19 std 命令发送数据

命令实例见图 19。

第一个参数为帧 ID，十六进制，十六位。

第二个参数如果为 remote，表示远程帧，如果为 hex 字符串，表示实际数据，最大 8 个字节。

5.4 ext

通过 CAN 接口发送扩展帧。

命令格式：ext [ID] [hex DATA | remote].

```
ext 44 12345678
 TX:00000005 DATA MSG ID=0x00000044 DATA=12 34 56 78
ext 44 remote
 TX:00000006 REMOTE MSG ID=0x00000044
ext 33 00112233445566788
 TX:00000007 DATA MSG ID=0x00000033 DATA=00 11 22 33 44 55 66 77
```

图 20 ext 命令发送数据

命令实例见图 20。

第一个参数为帧 ID，十六进制，三十二位。

第二个参数如果为 remote，表示远程帧，如果为 hex 字符串，表示实际数据，最大 8 个字节。

5.5 uart

显示、设置 UART 异步串口参数。

命令格式：uart [baud|mode|addr] UART interface params setup.

不带参数的 uart 命令显示当前 UART 接口状态，执行结果见图 21。

```
uart
 uart [baud|mode|addr] UART interface params setup.
 UART baud = 115200bps  mode = 0 TERM  addr = 1
```

图 21 不带参数的 uart 命令输出

5.5.1 uart baud

baud 子命令可以设置 UART 接口波特率，见图 22，使用“uart baud”命令查看当前 UART 接口波特率，使用“uart baud 57600”将 UART 接口波特率设置为 57600bps。

```
uart baud
 UART baud = 115200bps
uart baud 57600
 set UART baud to 57600bps, save & reboot to apply new baud.
```

图 22 uart baud 命令设置 UART 接口波特率

修改 UART 接口波特率以后，需要执行“param save”命令保存参数，并且执行“reboot”命令重启系统使能新波特率。

5.5.2 uart mode

mode 子命令可以设置 UART 接口工作模式，见图 23，使用“uart mode”命令查看当前 UART 接口工作模式，使用“uart mode 1”将 UART 接口工作模式设置为 PKT 模式。

```
uart mode
 UART mode = 0 TERM
uart mode 1
 set UART mode to 1 PKT, save & reboot to apply new mode.
uart mode 0
 set UART mode to 0 TERM, save & reboot to apply new mode.
```

图 23 uart mode 命令 UART 接口工作模式

修改 UART 接口工作模式以后，需要执行“param save”命令保存参数，并且执行“reboot”命令重启系统使能新工作模式。

5.5.3 uart addr

addr 子命令可以设置 UART 接口地址，见图 24，使用“uart addr”命令查看当前 UART 接口地址，使用“uart addr 2”将 UART 接口地址设置为 2。

```
uart addr
 UART addr = 1
uart addr 2
 set UART addr to 2, save & reboot to apply new addr.
```

图 24 uart addr 命令设置 UART 接口地址

5.6 clear

清理 CAN 发送接收计数器。

命令格式：clear

5.7 param

操作设备参数。

命令格式：param [load|save|restore] Operate parameters.

param save：保存参数到内置 EEPROM。

param load：从内置 EEPROM 加载参数。

param restore：恢复默认参数。

5.8 repeat

设置 CAN 数据帧发送重复次数和时间间隔。

命令格式：repeat [num] [int(ms)] CAN TX repeat times and interval.

该命令用于连续发送数据帧，执行情况参考图 25，每隔 100ms 重复发送一次数据帧，总共发送了 6 个数据帧。

clear 命令可以清除 repeat 和 interval 设置。

```
repeat 5 100
 Set TX repeat = 5 interval = 100ms
std 44 11223344
 TX:00000000 DATA MSG ID=0x0044 DATA=11 22 33 44
 TX:00000001 DATA MSG ID=0x0044 DATA=11 22 33 44
 TX:00000002 DATA MSG ID=0x0044 DATA=11 22 33 44
 TX:00000003 DATA MSG ID=0x0044 DATA=11 22 33 44
 TX:00000004 DATA MSG ID=0x0044 DATA=11 22 33 44
 TX:00000005 DATA MSG ID=0x0044 DATA=11 22 33 44
can
 can [baud|mode|init] CAN interface params setup.
 CAN  baud = 100kbps mode = 0 Normal
 RX   lost = 0 success = 0 show = 1
 TX   lost = 0 success = 6 repeat = 5 interval = 100ms
```

图 25 使用 repeat 命令连续发送数据帧

5.9 reboot

重启 UARTCAN，带一个延时参数，延时单位毫秒，不带参数立即重启。

命令格式：reboot [delay ms] Restart system.

延时 1 秒以后重启：reboot 1000。延时重启可以用来进入 BOOT 模式，具体方法请参考 4.3。

修改 UART 接口参数以后，需要重启生效。

5.10 help

显示在线帮助。

命令格式：help

命令输出见图 26。

```
help
 can -> can [baud|mode|init] CAN interface params setup.
 filter -> filter [0-6] [mode|scale|act|r0|r1] [param] CAN filter setup.
 std -> std [hex ID] [hex DATA | remote] send standard CAN data/remote message.
 ext -> ext [hex ID] [hex DATA | remote] send extended CAN data/remote message.
 uart -> uart [baud|mode|addr] UART interface params setup.
 clear -> clear RX and TX frame counter.
 param -> param [load|save|restore] Operate parameters.
 repeat -> repeat [num] [int(ms)] CAN TX repeat times and interval.
 reboot -> reboot [delay ms] Restart system.
 help -> help Info.
 version -> display SW version and SN.
```

图 26 help 命令输出

5.11 version

显示设备软硬件版本信息。

命令格式：version

命令输出见图 27。

```
version
 UARTCAN v16.5.2 Flash:64kB SN:54FF6D064971535716460687
 ECHO Studio <echo.xjtu@gmail.com>. All Rights Reserved.
```

图 27 version 命令输出

5.12 ctrl+p

```
RX:00000032 DATA MSG ID=0x0000 LEN=8  DATA=00 01 02 03 04 05 06 07
RX:00000033 REMOTE MSG ID=0x0000
RX:00000034 DATA MSG ID=0x00000000 LEN=8  DATA=00 01 02 03 04 05 06 07
RX:00000035 REMOTE MSG ID=0x00000000
```

图 28 超级终端接收到的 CAN 数据帧

TERM 模式下，超级终端会实时显示 CAN 接口收到的数据，见图 28。通过帧 ID 长度区分标准帧和扩展帧，标准帧 ID 为 16 位，扩展帧为 32 位。

按下 ctrl+p 组合键，可暂停 CAN 帧接收显示，再次按下 ctrl+p 继续显示接收到的数据包。

暂停显示以后，设备在后台继续接收 CAN 帧，可以使用 can 命令查看接收到的数据帧数量。

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 29。

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 29 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 2。

表 2 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 30。

```c
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 30 ModbusCRC 参考算法实现

6.2 通讯举例

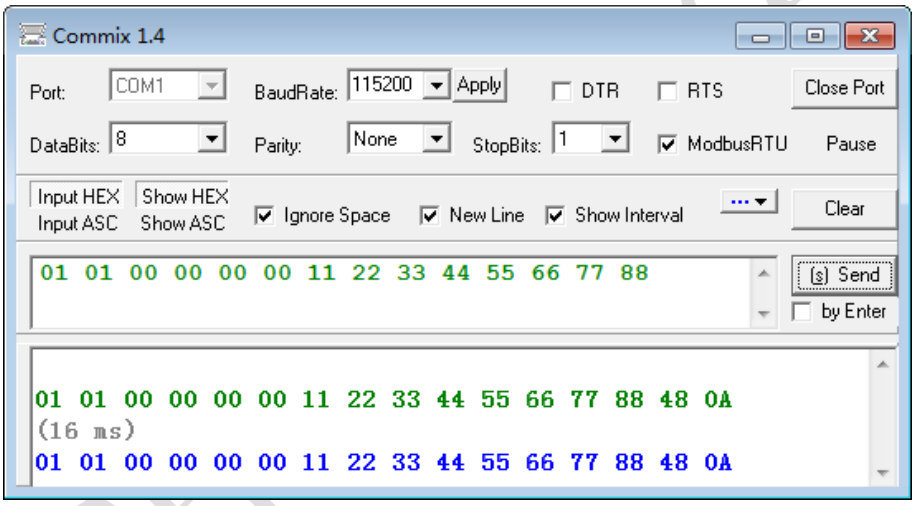

图 31 ECHO 功能码通讯测试

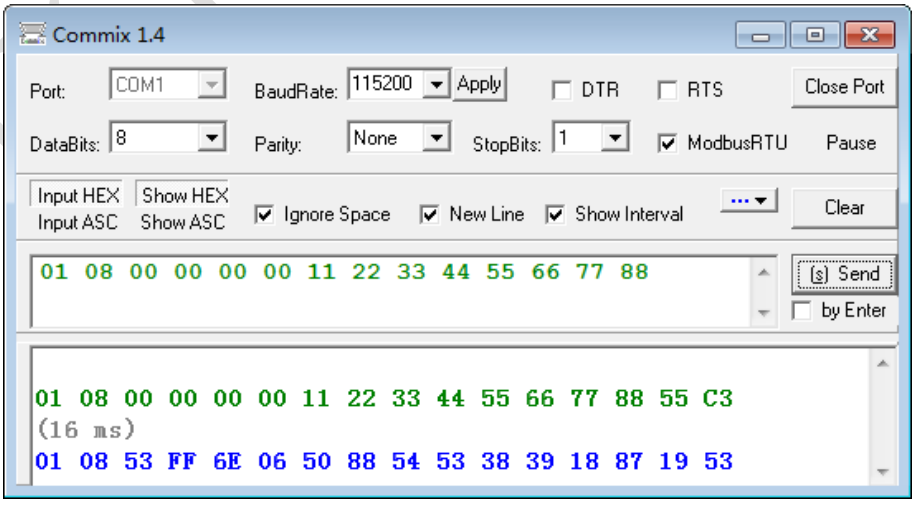

图 32 DEVSN 功能码通讯测试

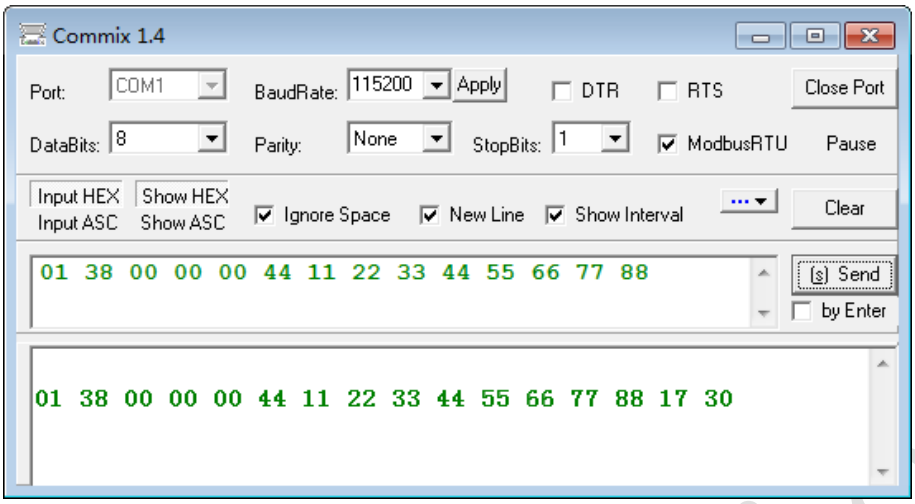

图 33 发送 ID 为 0x44 8 个数据字节的标准帧

6.3 CANTest 软件使用说明

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.

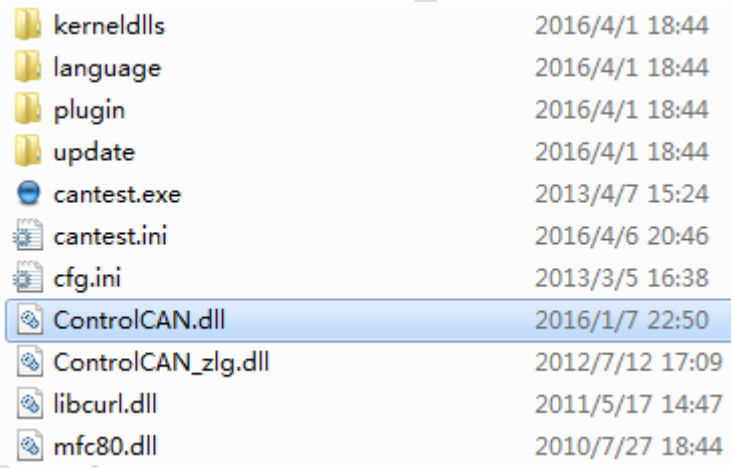

图 34 CANTest 软件 DLL 替换

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 其它常用操作

7.1 固件升级

进入 BOOT 模式，方法请参考 4.3。

执行 ymodem 命令。看到输出字符“C”以后，选择超级终端菜单“传送(T) ->发送文件(S)”，弹出图 35 界面，点击“浏览(B)”按钮，选择待升级的固件文件，后缀名为.hex，协议选择“Ymodem”，点击发送按钮，等待升级完成。

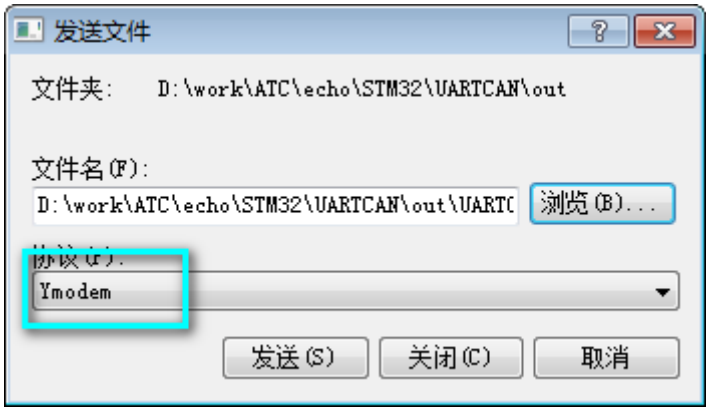

图 35 选择固件文件

升级过程见图 36，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。

```
ymodem
 Page  16 @ 0x08004000 is not empty! Erasing ... OK!
 Page  17 @ 0x08004400 is not empty! Erasing ... OK!
 Page  18 @ 0x08004800 is not empty! Erasing ... OK!
 Page  19 @ 0x08004C00 is not empty! Erasing ... OK!
 Page  20 @ 0x08005000 is not empty! Erasing ... OK!
 Page  21 @ 0x08005400 is not empty! Erasing ... OK!
 Page  22 @ 0x08005800 is not empty! Erasing ... OK!
 Page  23 @ 0x08005C00 is not empty! Erasing ... OK!
 Page  24 @ 0x08006000 is not empty! Erasing ... OK!
 Page  25 @ 0x08006400 is not empty! Erasing ... OK!
 Page  26 @ 0x08006800 is not empty! Erasing ... OK!
 Page  27 @ 0x08006C00 is not empty! Erasing ... OK!
 Page  28 @ 0x08007000 is not empty! Erasing ... OK!
 Page  29 @ 0x08007400 is not empty! Erasing ... OK!
 Page  30 @ 0x08007800 is not empty! Erasing ... OK!
 Page  31 @ 0x08007C00 is not empty! Erasing ... OK!
 ymodem update firmware, press A to abort ...
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
 Update firmware OK!
 File Name: UARTCAN.hex
 File Size: 44998
 End Addr: 0x08007E60
```

图 36 固件升级过程

7.2 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。

7.3 查看与切换工作模式

按下按键，LED 闪烁一次表示 PKT 模式，LED 闪烁两次表示 TERM 模式。长按按键可在两种模式之间切换。切换结果立即生效并保存。

8 FAQ

8.1 串口速度能比上 CAN 接口吗？

能。

CAN 最高波特率是 1Mbps，STM32 的 UART1 最高波特率 4.5Mbps，USB 转串口芯片

CH340G 最高波特率 2Mbps，PL2303HXD 最高波特率 12Mbps。串口最高波特率要高于 CAN。

8.2 如何记录数据？

利用超级终端的“捕获文字”功能，菜单“传送(T)->捕获文字(C)”。