

汽车车载影音系统CAN总线系统解决案例-Kvaser总线分析仪

关键词：CAN 总线 汽车音响 车载音响 车载影音

单元:方向盘-车载音响系统

测试工具:Kvaser CAN 总线分析仪

测试单位:广州智维电子科技有限公司

技术支持: 广州智维电子科技有限公司

地址: 广州市天河区五山华南理工大学国家科技园 2 号楼 223-224 室

邮编: 510640

电话: 020-3874 4186; 3874 4187

传真: 020-3874 4189

E-mail:

销售: sales@triv.cn

一般信息和技术支持: support@triv.cn

一. 方向盘音响控制按键的改装经验分享

第三代万能方向盘控制器的面市给汽车音响行业改装带来了方便,但是在一些特殊车型的改装中,还是需要一定的电子知识,特别是上海大众帕萨特领域刚出来不久就已经有了不少的车主喜欢,而且很多车主都会把原车的音响改装升级,广州智维为多家汽车音响开发生产厂家和 4S店提供大众领域、奔腾、逸朗等车型方向盘音响控制的改装测试分析与开发解决方案。

上海大众帕萨特领域 1.8T

原车方向盘上的右边上分别有 VOL+、VOL-音量加减和上下曲选择 4 个音响控制按键, 左边是巡航按键。

大众车厂也太省了 20 多万的车也就配台这样的单碟 CD 机,怪不得这么多车主都要升级音响换主机。新款的大众主机背后的插座已经不再是原来老款的接口了,右下角最底下的两根线是蓝牙免提电话的音频输入线。

从上边的接线图可以看得出,新款的主机连 ACC 控制信号都没有了,只有 CAN+和 CAN-两根控制线,通过总线的形式让车内的电脑与主机进行操探。

此车是特殊车型,改装的时候需要拆原车的方向盘下来改装里边的电路才行,要拆方向盘首先要后边找到两个孔,然后用螺丝刀去顶开里边的小钢条,和拆原来的老款的帕萨特一样。

两边都松开后轻轻把气囊扶好。注意不要把气囊线搞掉了。把左右两边的控制按键模块拆下来。右边的音响控制按键用的是一个 7 脚的连接器连接到左边去,只用了其中的 4 根线,白色的是上下选曲的控制线输出,黑色的是音量大小的控制线输出,棕色的是地线,灰/蓝色的是控制按键的使用背光源电源线。巡航控制按键模块的左边的是与外边连接的 5 脚插座,右边的 7 脚插座是与音响控制按键模块连接的插座方向盘上的所有按键模块都拆了下来。把音响控制按键的模块拆下来,共有 4 根内六角螺丝。音响控制按键模块的内部就是这么一块电路板,而且是单层的电路板,做得比较差。上边的控制按键和背光灯。这块电路板上没有几个零件,主要是音响控制的 4 个按键和背光灯和电路板上的几个电阻。右边那 4 个电阻就是我们要改的地方了。把那 4 个电阻拆除,旁边刚好有几个孔也是空的。把万能方向盘控制器上配的几个电阻中取出 820 欧、1.5K、2.2K 和 3.6K 出来装不用按顺序,任意装在上边的四个位置上。把电阻串出来的那边剪掉,完成音响控制按

键电阻的改装工作。随便把左边的巡航控制按键的模块也拆下来看看，里边好像比音响控制按键模块要用，用的是四层电路板技术，成本都要贵好多。上边也布满了元件，其中最大的那块是美国半导体公司的 XH51AB 芯片，所有的控制信号都是先输入到它再传送到外边去的。把连接到左边巡航控制按键模块的 7 脚连接器上的白色线剪断，因为它是我们把右边音响控制按键模块改电阻后变成的音响按键控制输出线了。

然后把另外一根 5 脚的棕色线也剪断，这根就是整个方向盘控制电路的地线，我们打算把它偷出来当音响按键的控制输出线。最好把左边的汽车喇叭按键拆下好接线和走线。两个地方都剪断了。

把方向盘底下的盖子拆下，拆的时候要注意除了底下有螺丝以外，在方向盘的正面旋转方向盘后可以看到左右两边还有两颗十字小螺丝在里边。拆下后可以方向盘底下找到上边这个橙色的插座，这个插座上的线就是方向盘上控制按键和气囊的输出线。把外边包的东西剪开一些，把里边棕色的线剪断。

把第三代方向盘控制器的黑色线（可以只接最粗的，也可以两根黑色线都接上）接到左边连接到车上的刚才剪断的棕色线上。把第三代方向盘控制器的绿色线或者蓝色接到棕色线连接到橙色插座的那边。（改装的时候要看加装的电阻的大小，建议如果加装按我上边的电阻值，还是不要用白色的）

然后再把红色的电源线加延长线，把其它线包好，并把发射管延长线的插座也插上。大家看到那个标有 75X 的大螺栓没有，那个就是很多人在改装帕萨特领域这些总线控制的车找不到 ACC 线的时候所需要的 ACC 线了。把第三代万能方向盘控制器的电源线也接到这个开机线上。

让我们再回到方向盘上，把方向盘上刚才剪断的 7 脚连接器上的白色线和 5 脚连接器上的棕色接起来并包好，完成偷线工作，使原来方向盘上控制按键的地线变成了音响控制按键专用的控制输出线，连接到底下的万能方向盘控制器上。然后再把 5 脚连接器上剪断的棕色线接到旁边的螺丝上，让原来的地线搭铁，使电源有回路。

如果可能的话，最好是再多加一根延长线除了固定在刚才的螺丝上以外再接多一根到汽车喇叭按键的大螺丝上。把健伍的遥控器拿来进行学习一下。装的可是健伍 DDX7067 的 2DIN DVD 主机哟。我们把延长的发射管装在了旁边一个空的没用的按键上。

试试操作，还真不错，和原装的一样！

由于原来的音响控制按键只有音量大小和上下选曲 4 个，为了更加人性化，我们把下一曲按键学习成了 SRC 音源选择键，把上一曲按键学习成下一曲按键。这样就可以直接在方向盘上切换音源，功能比原来更加强，而且选曲功能也还有。

二. 总线分析硬件

先进的 CAN 总线解决方案-工程师的第一选择

Advanced CAN Solutions

The Engineer's No. 1 Choice



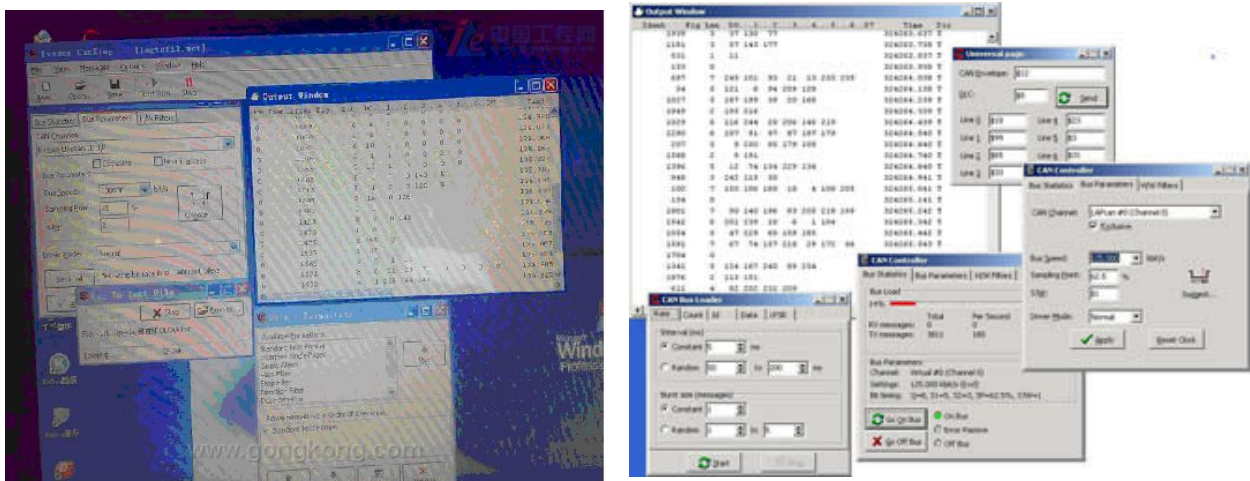
瑞典 Kvaser 公司专注于汽车、卡车、工业自动化和医学设备等现场测控领域。公司创始人 Lars-Bernro Fredriks-son 是世界上顶尖的 CAN 总线协议人物，不但作为 CAN 总线的起草人和执行者，并制定出世界上首个高层协议 CANKingdom，另外还有多项世界专利,如 Magi Sync 和 Silent mode 技术。

Kvaser 在多个平台 (USB、PCMCIA、PCI、ISA、Bluetooth 等)上拥有全方位的产品,可为相同的 API CANlib 开发并且相互兼容。MagiSync————Kvaser 的核心技术之一，使用创新的方法，不需要外部线缆就可以实现多个不同 Kvaser 接口的同步。

局域控制网 CAN，是一种有效支持分布实时控制的串行通讯总线。它是国际上应用最广泛的现场总线之一，在汽车工业、自动控制、航空航天、机械工业、医疗器械等领域中得到了极大的发展。CAN 已形成国际标准，被公认为几种最有前途的现场总线之一。其典型的应用协议有：CANKingdom、SAE J1939、CANOpen、DeviceNet、CANaerospace、NMEA-2000 等。

三. 测试分析软件

图 1:CANKing 分析软件界面:



Kvaser CanKing 是一个免费的 CAN 总线监测器和通用诊断工具。它特别适合于交互式开发工作。您可以容易地发送 CAN 消息并研究它们在目标模块上的影响。你只需要一个 Kvaser CAN 接口。Kvaser CanKing 完全免费。

特点:

基础版本使你能够在CAN总线上显示消息，并能够随意以几种不同方法格式化这些消息。你也可以通过填写对话框和压下OK来发送CAN消息。有很多消息排列选项。（ CanKingdom的用户将看到支持这里的几乎所有King页）

其它特点包括:

- 为你已经发送的消息列出历史表格
- CAN总线统计（消息的编号、总线负载等等）
- 一个通讯量发生器用于研究总线处于重载下的系统
- 消息进和出的时间标记（时间标记可以按绝对和相对格式显示）

四. 现场测试图片

领驭现场测试图



奔腾现场测试图



五. 单元节点 CAN 总线开发例程

调试好的 can 总线程序（c 语言）

```
/*******程序功能*****  
//节点 1 向 CAN 节点 2 发送数据,数据的头两个字节分别是:节点 1 的 ID 和要求节点 2 回复的数据长度  
//节点 2 收到节点 1 的信息后,立即按照节点 1 的要求回复数据
```

```

#include
#include //SJA 存储器定义头文件
#include //SJA 子程序文件
#include //显示头文件
#include //字符串函数
//*****头文件*****
*****//
void Init_Cpu(void); //初始化单片机
void Sja_1000_Init(void); //初始化 SJA
//*****
*****
bit s; //配置 sja 标志
bit flag_send; //发送命令标志
unsigned char data a[5]={0x05, 0x05, 0x05, 0x05, 0x05}; //显示"5"
unsigned char b[5]={0x12, 0x12, 0x12, 0x12, 0x12}; //显示"p"
unsigned char c[5]={0x01, 0x01, 0x01, 0x01, 0x01}; //显示"1"
unsigned char data send_data[10],rcv_data[10]; //发送和接收数组
unsigned char bdata flag_init; //保存中断寄存器值
unsigned int count_k; //延时记数用
sbit rcv_flag=flag_init^0; //接收中断标志
sbit err_flag=flag_init^2; //错误中断标志
//*****
*****//
void main(void)
{

    s=0; //配置 sja1000 出现错误时,重新初始化
    do{
        Sja_1000_Init();
    }while(s!=0);
    Init_Cpu(); //initialize mcu
    flag_init=0x00; //保存中断寄存器值清零
    while(1)
    {
        if(rcv_flag) //if there is receive interrupt
        {
            rcv_flag=0; //接收标志位清零
            BCAN_DATA_RECEIVE(rcv_data); //接收数据
            BCAN_CMD_PRG(RRB_CMD); //释放接收缓冲区
            flag_send=1; //发送命令置位
        }
    }
}

```

```

        if(flag_send)
        {
            flag_send=0; //发送位清零
            send_data[0]=rcv_data[2]; //接收到的"发送方 ID10~ID3"
            send_data[1]=rcv_data[3]; //接收到的"发送方 ID2~ID0"和
要求的数据长度
            send_data[2]=0x88;
            send_data[3]=0x89;
            BCAN_DATA_WRITE(send_data); //发送数据
            BCAN_CMD_PRG(TR_CMD); //置位发送请求位
            for(count_k=0;count_k<200;count_k++)
            display(a); //延时显示"5"
        }
        if (err_flag) //错误中断
        {
            for(count_k=0;count_k<280;count_k++)
            display(c); //错误显示"1"
            err_flag=0; //错误标志位清零
            Sja_1000_Init(); //初始化 SJA
        }
        display(b); //显示"p"
    }
}

```

```

void ex0_int(void) interrupt 0 using 1 //外部中断 0
{
    SJA_BCANAdr=REG_INTERRUPT; //指针指向中断寄存器
    flag_init=*SJA_BCANAdr; //保持中断寄存器值
}

```

```

void Init_Cpu(void) //单片机初始化,开放外部中断
0
{
    PX0=1;
    EX0=1;
    EA=1;
}

```

```

void Sja_1000_Init(void)
{
    s=BCAN_CREATE_COMMUNATION(); //SJA 自测
}

```

```
s=BCAN_ENTER_RETMODEL(); //进入复位
s=BCAN_SET_BANDRATE(0x04); //设置波特率 100K/S
s=BCAN_SET_OBJECT(0xaa, 0x00); //设置地址 ID:550
s=BCAN_SET_OUTCLK(0xaa, 0x48); //设置输出方式, 禁止 COLOCKOUT 输出
s=BCAN_QUIT_RETMODEL(); //退出复位模式
SJA_BCANAdr=REG_CONTROL; //地址指针指向控制寄存器
*SJA_BCANAdr|=0x1e; //开放错误\接收\发送中断
}
```