

# 目的：根据客户提供的DBC文件，在GB-CPro上添加对应车身协议和应用功能协议，来支持雷达在实验车的工作。

以创维汽车-BE11项目的整车协议为例

## 资料：

- BE11\_CAN.dbc
- BSD\_CH\_CHCAN\_BE11\_CAN\_Matrix\_20221025\_V9.1

## 过程

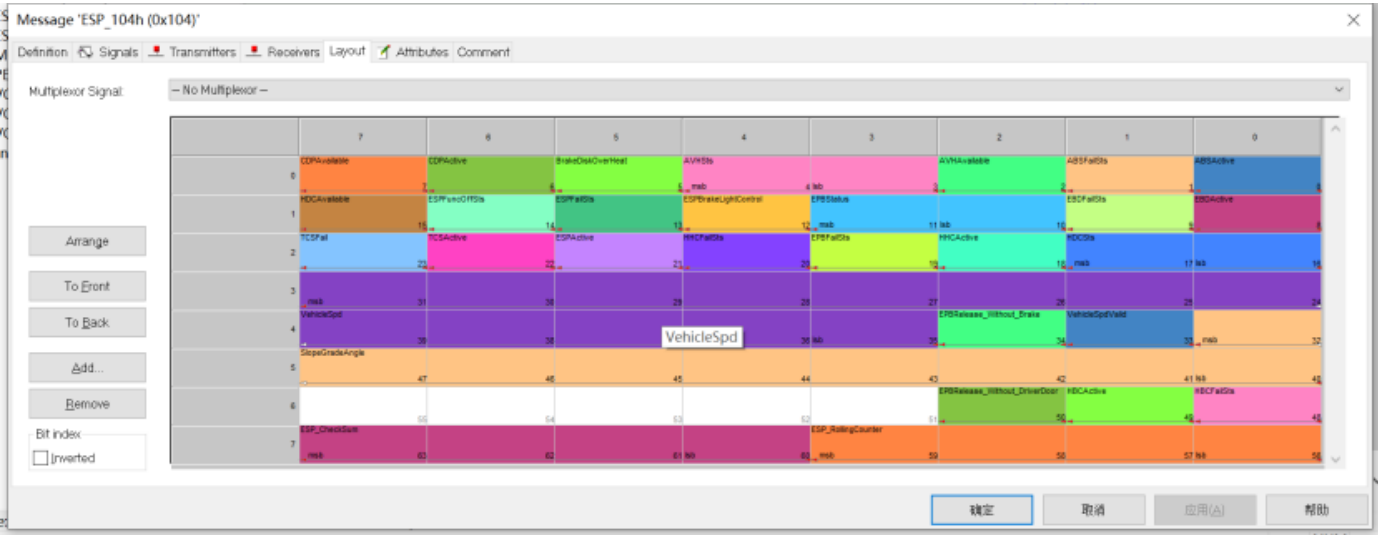
### 步骤一

确定需求：适配车身CAN协议

在BE11\_CAN.dbc找到初步开发需要的三个车身信息，车速speed，转向角速度yawrate和当前档位gear

- 1. 车速speed

0x104



确定起始位，

## Message Signal 'ESP\_104h:VehicleSpd'

Definition	Message	Signal	Receivers	Value Descriptions	Attributes	Comment
Message Name:	ESP_104h					
Signal Name:	VehicleSpd					
Startbit [Bit]:	35					
Multiplexortype	Signal					

确定length, 缩放系数, 最小值, 最大值, 字节序, 值类型, 单位和offset

## Message Signal 'ESP\_104h:VehicleSpd'

Definition	Message	Signal	Receivers	Value Descriptions	Attributes	Comment
Name:	VehicleSpd					
Length [Bit]:	13					
Byte Order:	Motorola	Unit:	km/h			
Value Type:	Unsigned	Init. Value	0			
Factor:	0.05625	Offset:	0			
Minimum:	0	Maximum:	300			
Calculate minimum and maximum						
Value Table:	<none>					

实际: (0 to 300) is 0-300, (300 + 0) / 0.05625 is 5333.3. 只用了0x00 到0x14d5

对应

车速 0x0000~0x14D5: Valid 0x14D6~0x1FFE: Reserved
---

然后, 在can\_hal.c里增加过滤器, 在can\_cli\_commands里收到0x104就开启nvd\_ESP\_104h函数

```

#elif(CAR == BE11)
{
    result = can_cli_register(MESSAGE_ID_VCU_200_BE11,          nvd_VCU_200h);
    result = can_cli_register(MESSAGE_ID_ESP_104_BE11,         nvd_ESP_104h);
    result = can_cli_register(MESSAGE_ID_EPS_117_BE11,         nvd_EPS_117h);
    result = can_cli_register(MESSAGE_ID_VCU_123_BE11,         nvd_VCU_123h);
}

```

nvd\_ESP\_104h按照刚才解读的消息读取未解析的速度，nsdGetARSSpeed得到解析后的速度，供雷达使用

```

nsdGetARSSpeedVehicleInfo.nuhw_Speed int32_t nvd_ESP_104h(uint8_t *data, uint32_t len)
{
    VehicleInfo.nuhw_Speed = nuhlwMtCANanaly_func(35, 13, 8, data);    /*km/h*/

    //TEST
    uint8_t g_datastart[8] = {0};
    g_datastart[1] = ((VehicleInfo.nuhw_Speed & 0xff00) >> 8);
    g_datastart[2] = ((VehicleInfo.nuhw_Speed & 0x00ff) >> 0);
    cansendtest(0,0x15A,g_datastart);
    return 0;
}

float nsdGetARSSpeed(void)
{
    float temp = 0;

    if(VehicleInfo.nuhw_Speed > 0x14D5)
    {
        temp = 0;
    }
    else
    {
        temp = VehicleInfo.nuhw_Speed *0.05625 -0;    //km/h
    }
    return temp;
}

```

2.yawrate 和gear 同上。

## 步骤二

未在网关中的协议，因此要屏蔽掉VehicleSpeed.c的函数

```

/*
 * VehicleSpeed.c
 *
 * Created on: 2021年8月26日
 * Author: Eric
 */
#include "Vehicle_Info.h"
#include "../../ASW/APP/ISO17387/AssistedDriving.h"
#include "BE11.h"

#if (CAR != BE11)
int32_t nvdARSSpeedSave(uint8_t *data, uint32_t len)
{
    VehicleInfo.nub_GearStu = nuhlwMtCANanaly_func(6, 2, 8, data);
    VehicleInfo.nuhw_Speed = nuhlwMtCANanaly_func(8, 13, 8, data);
    stCanTimeOut[2].uhwTimeOutCnt = 0;
    return 0;
}

float nsdGetARSSpeed(void)
{
    return VehicleInfo.nuhw_Speed*0.02;
}

uint8_t nubGetARSGearStu(void)
{
    uint8_t ub_res = 0;

    switch (VehicleInfo.nub_GearStu)
    {
        case 0x2:
            ub_res = Gear_R;
            break;
        default:
            ub_res = 1;
    }
}

```

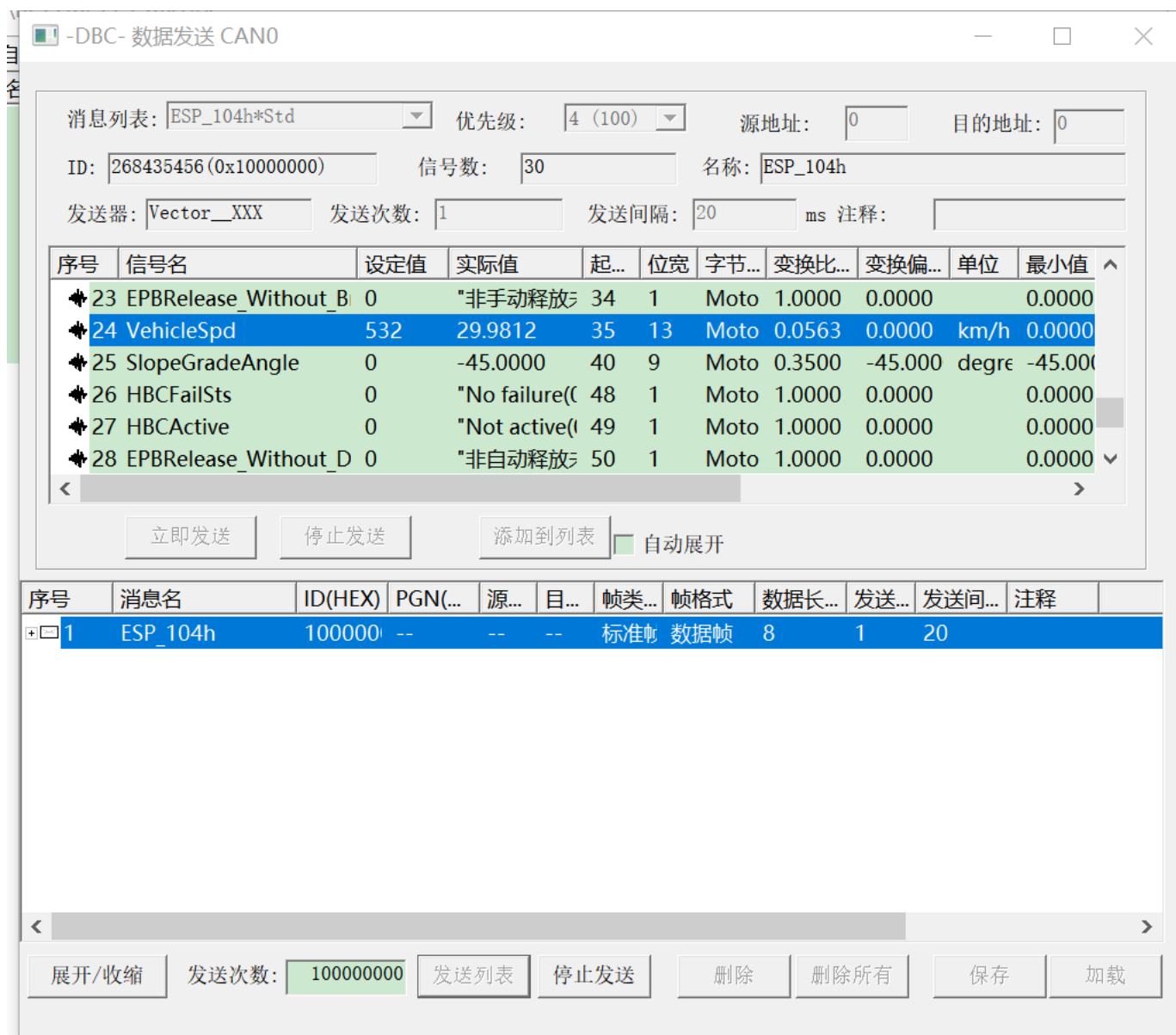
### 步骤三

确定需求：验证车身协议适配开发

烧录后，用canpro往雷达can0发写好的0x104测试30km/h的车速(DBC解析->加载协议->发送)

bug:BE11的协议并未加载下面的帧ID

解决方法：手动发一个0x104,或者让他接收一个0x104



在上面会回一个0x15a的测试CAN消息，他的第1，2为为uint16\_t的车速  
VehicleInfo.nuhw\_Speed

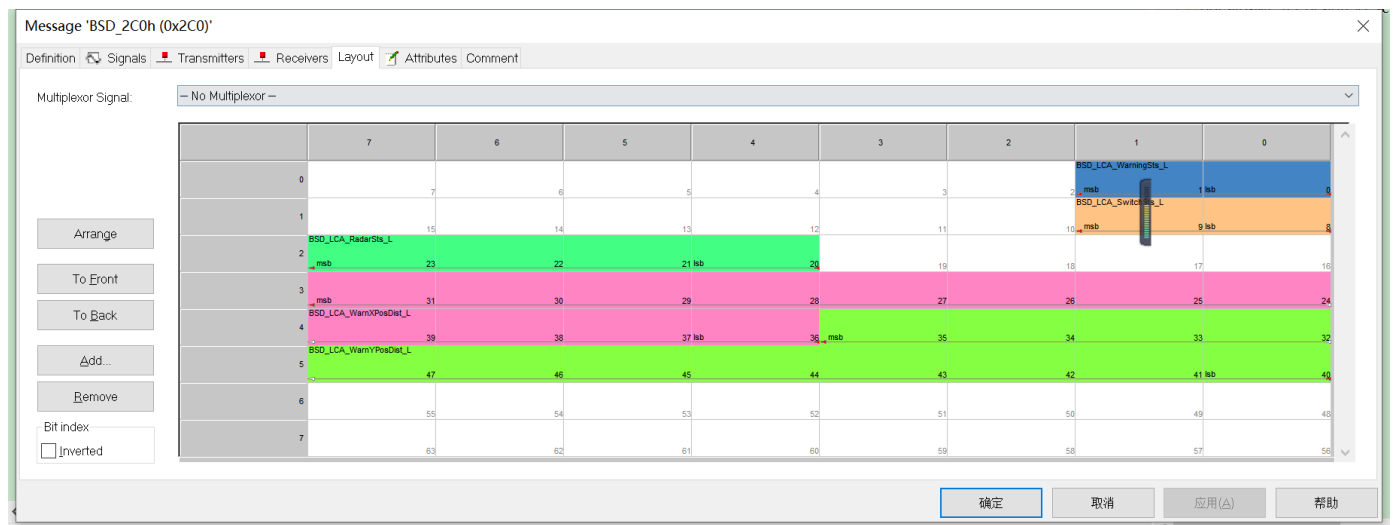


再在nsdGetARSSpeed里， $532 * 0.05625 - 0 = 29.925$

解析正确

### 步骤三

## 确定需求：适配应用功能协议



客户要求左雷达BSD, LCA, RCTA, DOW四个功能, 报警信号共用一个0x2C0 右雷达BSD, LCA, RCTA, DOW四个功能, 报警信号共用一个0x2C1

按实际情况开发

```
/* brief: set warning signal send to Can Bus BE11
***** */
void nvdRadarWarningSend(AppHandle_t *app_Handle)
{
    Type_Can_Msg astMsg;
    uint32_t MsgBuf[2];
    uint8_t LWarningSts = 0;
    uint8_t RWarningSts = 0;
    uint8_t SwitchSts = 0;

    astMsg.CanIdType = FLEXCAN_MSG_ID_STD;
    astMsg.DataLen = eDATA_LEN_8;
    astMsg.Channel = CAN_0_ID;
    astMsg.Data[0] = 0x0;
    astMsg.Data[1] = 0x0;
    astMsg.Data[2] = 0x0;
    astMsg.Data[3] = 0x0;
    astMsg.Data[4] = 0x0;
    astMsg.Data[5] = 0x0;
    astMsg.Data[6] = 0x0;
    astMsg.Data[7] = 0x0;

    /*SwitchSts,be in common use*/
    if(((nstFeeData.stRadarCfg.ubflgBSDSwit) | (nstFeeData.stRadarCfg.ubflgLCASwit) | \
        (nstFeeData.stRadarCfg.ubflgDOWSwit) | \
        (nstFeeData.stRadarCfg.ubflgRCTASwit)) == 1)
    {
        if(((app_Handle->BSD_Handle.active) | (app_Handle->LCA_Handle.active) |\
```

```

1) (app_Handle->DOW_Handle.active) | (app_Handle->RCTA_Handle.active)) ==

{
    SwitchSts = 1; //active
}
else
{
    SwitchSts = 2; //standby
}
}
else
{
    SwitchSts = 0; //OFF
}

if(CURRENT_RADAR == RADAR_REAR_LEFT || CURRENT_RADAR == RADAR_FRONT_LEFT ||
CURRENT_RADAR == RADAR_FRONT_1)
{
    /*WarningSts*/
    if(( (LRadarWarning.BSD) | (LRadarWarning.LCA) | (LRadarWarning.DOW) |
(LRadarWarning.RCTA ) ) >= 2 )
    {
        LWarningSts = 2;
    }
    else if (( (LRadarWarning.BSD) | (LRadarWarning.LCA) | (LRadarWarning.DOW) |
(LRadarWarning.RCTA ) ) == 1)
    {
        LWarningSts = 1;
    }
    else
    {
        LWarningSts = 0;
    }

    astMsg.MsgID = MESSAGE_ID_RADARWARNING_INFO_L;
    astMsg.Data[0] = 0x03 & LWarningSts;
    astMsg.Data[1] = 0x03 & SwitchSts;
    astMsg.Data[2] = 0x10; //todo:BE11 RadarSts_L,radar stu
    astMsg.Data[3] = 0x5A; //todo:5A6 is 1446,-60,read or Write constant
    astMsg.Data[4] = 0X6B;
    astMsg.Data[5] = 0x82; //todo:B82 is 2946,+90 is 0.9m?
}
else if(CURRENT_RADAR == RADAR_REAR_RIGHT || CURRENT_RADAR == RADAR_FRONT_RIGHT ||
CURRENT_RADAR == RADAR_FRONT_2)
{
    /*WarningSts*/
    if(( (RRadarWarning.BSD) | (RRadarWarning.LCA) | (RRadarWarning.DOW) |
(RRadarWarning.RCTA ) ) >= 2 )
    {
        RWarningSts = 2;
    }
    else if (( (RRadarWarning.BSD) | (RRadarWarning.LCA) | (RRadarWarning.DOW) |
(RRadarWarning.RCTA ) ) == 1)

```

```

{
    RWarningSts = 1;
}
else
{
    RWarningSts = 0;
}

astMsg.MsgID = MESSAGE_ID_RADARWARNING_INFO_R;
astMsg.Data[0] = 0x03 & RWarningSts;
astMsg.Data[1] = 0x03 & SwitchSts;
astMsg.Data[2] = 0x10; //todo:BE11 RadarSts_L,radar stu
astMsg.Data[3] = 0x5A; //todo:5A6 is 1446,-60,read or Write constant
astMsg.Data[4] = 0X64;
astMsg.Data[5] = 0x7A; //todo:B82 is 2946,+90 is 0.9m?
}

```

```

MsgBuf[0] = (astMsg.Data[3] << 24) | (astMsg.Data[2] << 16) | (astMsg.Data[1] << 8)
| astMsg.Data[0];
MsgBuf[1] = (astMsg.Data[7] << 24) | (astMsg.Data[6] << 16) | (astMsg.Data[5] << 8)
| astMsg.Data[4];
can_message_send(astMsg.Channel, astMsg.MsgID, MsgBuf, astMsg.DataLen);
}

```

## can解析成功

FrameAnalyzer-C:\Users\Cui\Desktop\BE11\BSD_CH_CHCAN_BE11_CAN_Matrix_20221025_V9.1.dbc													
<span>加载协议</span> <span>暂停</span> <span>分类显示</span> <span>自动滚屏</span> <span>清空列表</span> <span>导出</span> <span>发送</span> <span>协议类型: CAN</span>													
序号	CAN...	传输方...	时间	消息名	ID(HEX)	PGN(H)	源地...	目的...	帧类型	帧格式	数据...	帧数据(HEX)	注释
1	0	接收	3117.9461	BSD_2C1h	000002C1	--	--	--	标准帧	数据帧	08	01 01 10 5A 64 7	
		信号名	实际值	描述	注释	原始值	起始位	位宽	变换比	变换偏			
		BSD_LC	1	1级报警(1)	右告警状态	1	0	2	1	0			
		BSD_LC	1	Active(1)	右开关状态	1	8	2	1	0			
		BSD_LC	1	ON(1)	右角雷达状态	1	20	4	1	0			
		BSD_LC	-60.09999999	--	X向距离_R	5A6	36	12	0.1	-204.7			
		BSD_LC	-90.09999999	--	Y向距离_R	47A	40	12	0.1	-204.7			