



VECTOR 的 GENY IL 接口调用接口生成工具使用说明

CT-ITC-OS



2017-11-12

惠州市和畅五路

目录

| | | |
|----------|------------------------|----------|
| 1 | 工具概况..... | 2 |
| 1.1 | 现状分析..... | 2 |
| 1.2 | 设计思想..... | 3 |
| 2 | 代码生成工具使用步骤..... | 4 |

1 工具概况

RH850 系列芯片的 CAN 部分使用的是 VECTOR 公司的代码包，使用 GENy 工具根据 DBC 生成代码，生成面向信号的 API 接口供应用层使用。VECTOR 代码包中已经实现了大部分工作，但实际调用 API 的时候会存在一些问题，工具致力于解决这些问题（以下论述都是为解决发送的消息，接收的暂时没有完善）。

1.1 现状分析

根据 VECTOR 的帮助文档《TechnicalReference_GENy_InteractionLayer.pdf》的第 26 页说明，已经明确指出了 IL 层只做无符号数（原始值）的值的接口访问，DBC 中的有符号数、浮点数、因子等由应用层做解析。



Caution

All generated signal access only provides **unsigned integer** values. Signed, float and the scaling factors (as adjustable in CANdb++) are not supported and have to be interpreted by the application.

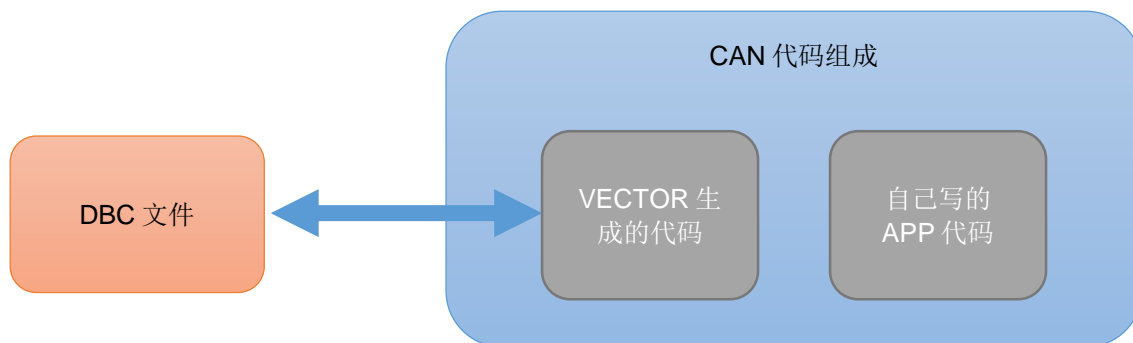
©2013, Vector Informatik GmbH

Version: 2.10.03

26 / 115

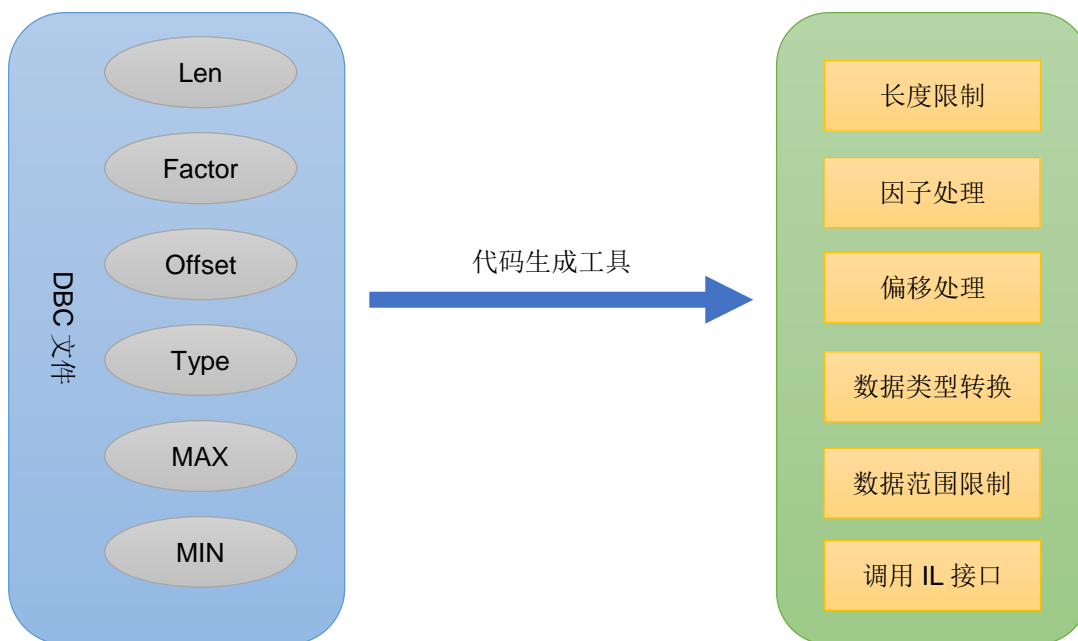
based on template version 3.7

这样如果有成千上万的信号的话，同时有符号（Signed）、无符号（Unsigned）、因子（Factor）、偏移（Offset）混在一起的时候，则需要写很多重复的代码，并且容易出错，更换 DBC 后有需要重复写一遍代码，违背了只维护一份 DBC 的初衷。



1.2 设计思想

认清数据的源头、CAN 的数据变化、报文封装都记录在 DBC 文件中，因此本着只维护 DBC 一份文件，代码中所有跟 DBC 相关的数据更改应该由代码生成工具来完成。只要工具完善，就能做到不容易出错、替换 DBC 方便、减少重复工作、提高开发效率。



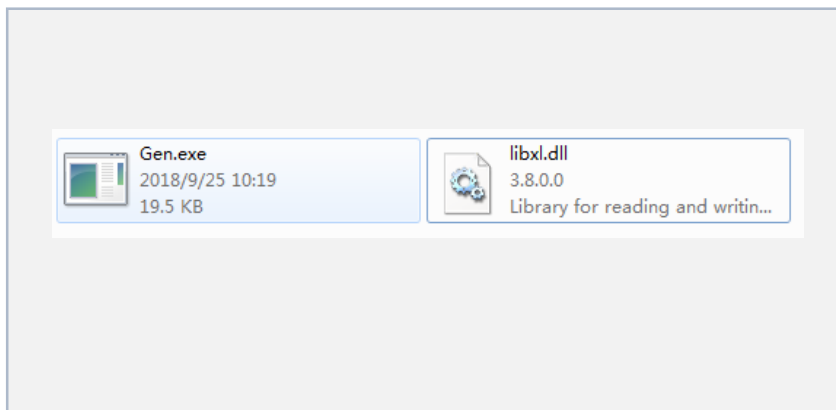
信号长度、因子、偏移等这些信息都记录在 DBC 文件中，并且 GENy 工具没有对这些做解析，需要一种补充的工具来生成这些代码。

2 代码生成工具使用步骤

第一步

1、拷贝公司版本库代码生成工具到 Windows，工具包含一个 EXE 可执行文件和 DLL 动态库。

2、也可从 github 版本库下载源码，路径如下：
<https://github.com/KuangAlbert/GENy-IL-InterfaceGenerator>



第二步

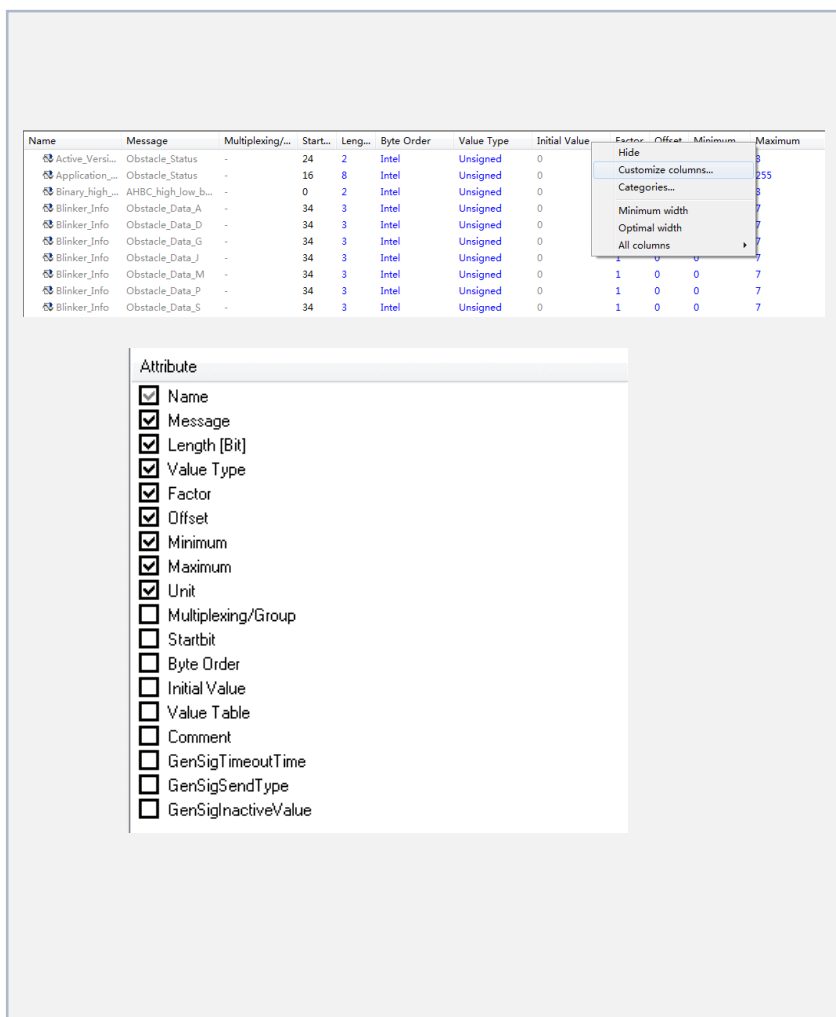
1、DBC 中的 max 和 min 必须设置且设置正确。

2、单击 Name，把信号按名字排序

2、在最上面一栏单击鼠标右键选择 Customize columns

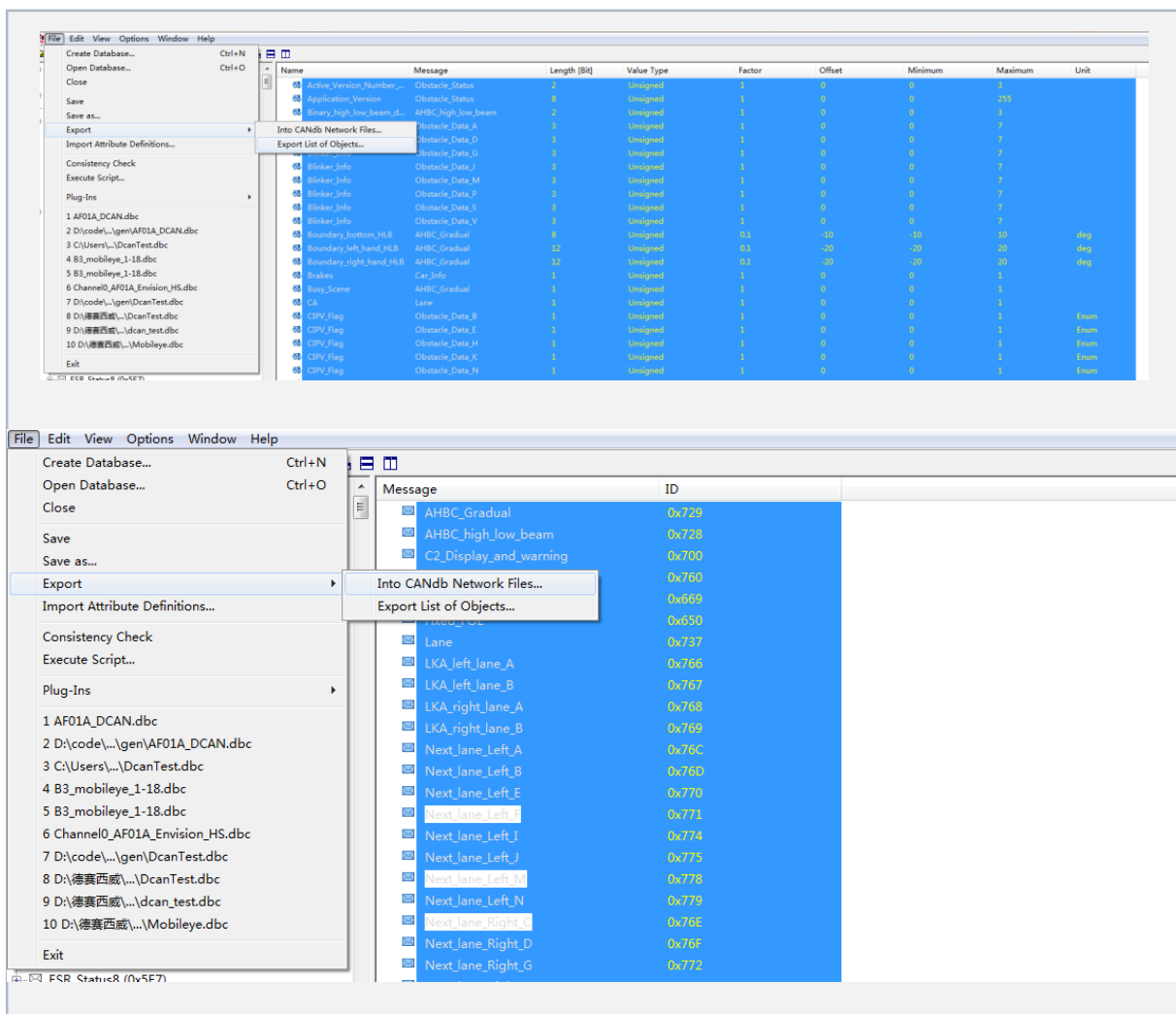
3、筛选值显示

Name
Message
Length
Value Type
Factor
Offset
Min
Max
Unit



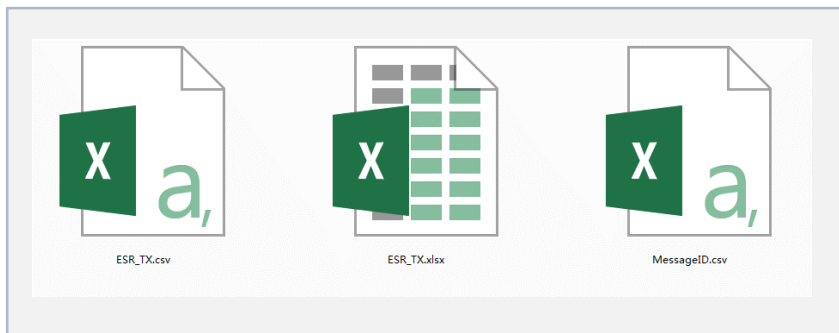
第三步

- 1、将信号全选、导出为 csv 格式的文件
- 2、用同样的方法导出消息和消息 ID 为 CSV 文件



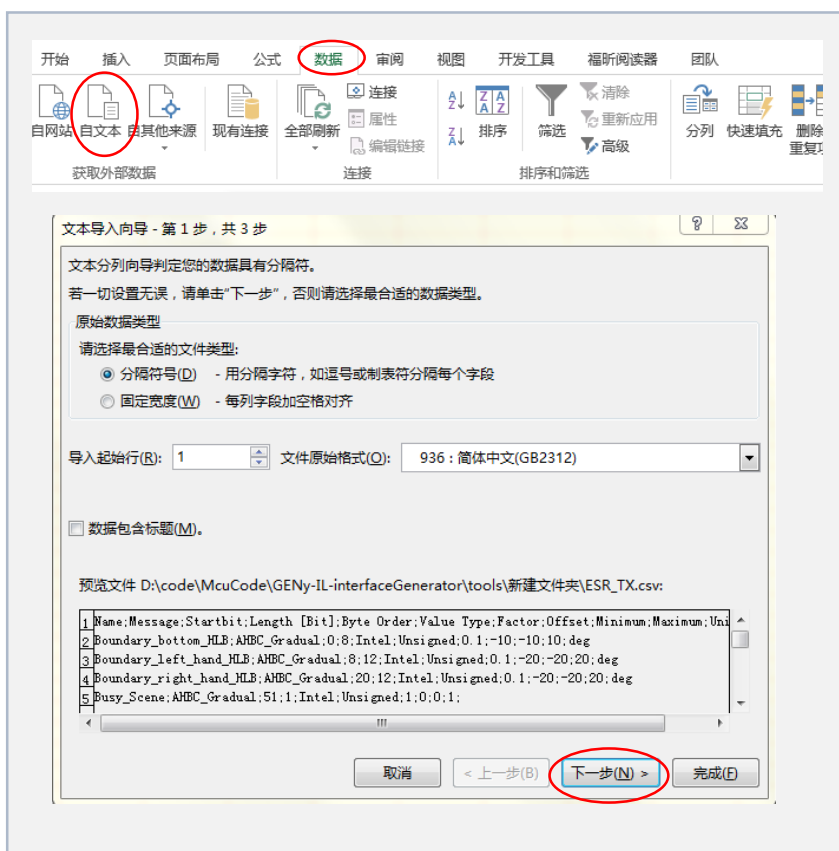
第四步

1、新建一个 excel 空白文档，命名为 ESR_TX.xlsx



第五步

1、选择数据->自文本，选择刚才的 csv 文件



2、单击下一布

第六步

1、选择分号

2、单击下一步

文本导入向导 - 第 2 步, 共 3 步

请设置分列数据所包含的分隔符号。在预览窗口内可看到分列的效果。

分隔符号

☒ Tab 键(T) ☐ 连续分隔符号视为单个处理(R)

☒ 分号(M)

☐ 逗号(C)

☐ 空格(S)

☐ 其他(O):

文本识别符号(Q):

数据预览(P)

| Name | Message | Startbit | Length [Bit] | Byte Order | Value Type | Factor | Offs |
|-------------------------|--------------|----------|--------------|------------|------------|--------|------|
| Boundary_bottom_HLB | AHBC_Gradual | 0 | 8 | Intel | Unsigned | 0.1 | -10 |
| Boundary_left_hand_HLB | AHBC_Gradual | 8 | 12 | Intel | Unsigned | 0.1 | -20 |
| Boundary_right_hand_HLB | AHBC_Gradual | 20 | 12 | Intel | Unsigned | 0.1 | -20 |
| Busy_Scene | AHBC_Gradual | 51 | 1 | Intel | Unsigned | 1 | 0 |

取消 < 上一步(B) 下一步(N) > 完成(F)

第七步

1、把常规都改成文本

2、点击完成

3、用同样的方法把另外一份 csv 导入到 sheet2

使用此屏内容可选择各列，并设置其数据格式。

列数据格式

☒ 常规(G) ☐ 文本(T)

☐ 日期(D): YMD

☐ 不导入此列(跳过)(I)

“常规”数据格式将数值转换成数字，日期值会转换成日期，其余数据则转换成文本。

高级(A)...

数据预览(P)

| 文本 | 常规 | 文本 | 文本 | 文本 | 文本 | 文本 | 文本 |
|-------------------------|--------------|----------|--------------|------------|------------|--------|------|
| Name | Message | Startbit | Length [Bit] | Byte Order | Value Type | Factor | Offs |
| Boundary_bottom_HLB | AHBC_Gradual | 0 | 8 | Intel | Unsigned | 0.1 | -10 |
| Boundary_left_hand_HLB | AHBC_Gradual | 8 | 12 | Intel | Unsigned | 0.1 | -20 |
| Boundary_right_hand_HLB | AHBC_Gradual | 20 | 12 | Intel | Unsigned | 0.1 | -20 |
| Busy_Scene | AHBC_Gradual | 51 | 1 | Intel | Unsigned | 1 | 0 |

取消 < 上一步(B) 下一步(N) > 完成(F)

第八步

1、双击 EXE 即可生成
dcanTx_gen.c 和
dcanTx_gen.h 的代码

```
00001: /*****
00002: This C file is modified according to DBC and is generally not allowed to be changed.
00003: Code generator version: V1.1
00004: Author: KuangSaoFengLei
00005: Date: 2018/09/24
00006: Gen time: 2018-09-24 14:21:35
00007: *****/
00008: #include "dcanTxLink_priv.h"
00009:
00010: sig_attribute attr;
00011:
00012: void Send_signal_IL(_DcanTxDataPacket* pk)
00013: {
00014:     attr.data_type = Unsigned;
00015:     attr.factor = 0.1;
00016:     attr.offset = -10;
00017:     attr.max = 10;
00018:     attr.min = -10;
00019:     attr.nbits = 8;
00020:     attr.value = pk->AHBC_Gradual.Boundary_bottom_HLB;
00021:     can_send_sig(Boundary_bottom_HLB_0x729, data_conv(&attr));
00022:
00023:
00024:     attr.data_type = Unsigned;
00025:     attr.factor = 0.1;
00026:     attr.offset = -20;
00027:     attr.max = 20;
00028:     attr.min = -20;
00029:     attr.nbits = 12;
00030:     attr.value = pk->AHBC_Gradual.Boundary_left_hand_HLB;
00031:     can_send_sig(Boundary_left_hand_HLB_0x729, data_conv(&attr));
00032: }
```

狂暴风雷

2018 年 9 月 25 日 修改