

doi:10.11835/j.issn.1000-582X.2021.109

# 基于正则表达式与多叉树的 DBC 网络协议解析方法

孙 汉<sup>1a,1b</sup>, 杨亚联<sup>1a,1b</sup>, 周 林<sup>2</sup>, 张 昕<sup>3</sup>

(1. 重庆大学 a. 机械与运载工程学院; b. 机械传动国家重点实验室, 重庆 400044; 2. 金康赛力斯新能源汽车设计院, 重庆 401120; 3. 江铃重型汽车有限公司, 太原 030032)

**摘要:**为解决 CAN 网络下 DBC(database CAN)协议数据解析和结构表征问题,通过正则表达式解析了 DBC 文件协议信息,利用多叉树结构,对 CAN 网络协议信息进行结构表征。针对 Intel 和 Motorola 不同格式下的协议解析问题,提出了报文数据场的信号色块表示方法以及起始位转换公式。基于 C# 语言设计和开发了 CAN 网络数据管理软件,并将 CAN 网络协议以 DBC 数据管理文件输出。用实车试验验证了所提出 DBC 协议解析方法的有效性。

**关键词:** CAN 网络; DBC; 数据管理; 正则表达式; 多叉树

**中图分类号:** TP391.1

**文献标志码:** A

**文章编号:** 1000-582X(2022)08-078-09

## A regular-expression and multi-tree based DBC network protocol analysis method

SUN Han<sup>1a,1b</sup>, YANG Yalian<sup>1a,1b</sup>, ZHOU Lin<sup>2</sup>, ZHANG Xin<sup>3</sup>

(1a. College of Mechanical and Vehicle Engineering; 1b. State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, P. R. China; 2. Jinkang Seres New Energy Automobile Design Institute, Chongqing 401120, P. R. China; 3. JMC Heavy Duty Vehicle Co., Ltd., Taiyuan 030032, P. R. China)

**Abstract:** To solve the problem of DBC (Database CAN) protocol data analysis and structure characterization in the CAN network, in this paper, the DBC file protocol is analyzed through regular expressions, and the CAN network protocol is characterized by using the multi-tree structure. To solve the protocol analysis problem in different formats of Intel and Motorola, an expression method of signal color blocks in the data field of messages and a start bit conversion formula are proposed. The CAN network data management software is designed and developed based on the C# language, and the CAN network protocol is output as a DBC data management file. An actual vehicle test is conducted, which verifies the validity of the DBC protocol analysis method proposed in this paper.

**Keywords:** CAN network; DBC; data management; regular expression; multi-tree

收稿日期: 2021-01-07 网络出版日期: 2021-05-19

基金项目: 国家自然科学基金资助项目(51575064)。

Supported by the National Natural Science Foundation of China (51575064).

作者简介: 孙汉(1996—), 男, 重庆大学硕士研究生, 主要从事车联网研究, (E-mail) 2253975649@qq.com。

通信作者: 杨亚联, 男, 教授, 博士生导师, 主要从事动力传动及控制、车联网等研究, (E-mail) yyl@cqu.edu.cn。

CAN 网络在发展的过程中衍生出了多种应用层协议,包括 CANopen、J1939、DeviceNet,目前分别由美国 ODVA 协会、SAE 协会和欧洲 CiA 协会管理。其中 CANopen 协议的核心内容是对象字典,它是一个有序的对象组,定义一个设备的所有信息<sup>[1]</sup>,并记录在 EDS 或 DCF 电子文档中,但是对于那些有很多对象,特别是很多数组的设备,EDS 文档将会变得很大,对嵌入式系统的存储器是一个挑战;J1939 协议主要适用那些以柴油为动力的卡车、公共汽车以及非陆地的交通工具;DeviceNet 协议更多是应用在工业设备上。在之前的研究中,也有采用 XML 作为 CAN 网络协议的描述文件<sup>[2]</sup>,这种方法是将协议内容录入到 XML 文件中,利用了 XML 格式解析的方便性,但是并没有根本解决工作量大且录入容易出错的问题。

DBC 数据文件由德国 Vector 公司发布,它是一种汽车 CAN 网络协议描述文件,里面大量的字符映射了各逻辑节点信息及相互之间的通信规则。相较于 Excel、Word 版的 CAN 协议描述文件,DBC 文件格式结构清楚、设计方便,已经成为 CAN 通信行业的标准,被 CANoe、Matlab、Labview、INCA 等软件和平台广泛采用。

基于 DBC 文件可以进一步开发出 CAN 网络仿真工具软件,用来监视和分析 CAN 网络中所有逻辑节点的参数和状态<sup>[3]</sup>。CANoe 是具有代表性的工具软件之一,它需要借助 DBC 数据文件来对 CAN 网络上的报文帧进行解析<sup>[4]</sup>。基于 CANoe 平台,DBC 文件被应用到很多具体的场景中:车身控制器局域网络仿真<sup>[5]</sup>、基于 CAN 的故障诊断系统<sup>[6-7]</sup>、汽车动力总成的试验台架测试<sup>[8]</sup>、汽车电子 ECU 控制器开发,还被应用于 Simulink 和 CANoe 联合仿真<sup>[9]</sup>以对混合动力汽车的通信和控制策略进行验证<sup>[10-13]</sup>,同时在整车 ECU 和网关等软硬件的开发和测试<sup>[14]</sup>中也得到了应用。

基于 CAN 网络数据管理的需要,并针对上述 CAN 网络协议描述方式存在的弊端,笔者基于 DBC 文件规范,设计了协议解析的正则表达式,根据 CAN 网络协议信息的结构关系,利用二叉树对该网络协议进行表征,针对 Intel 和 Motorola 不同格式下的协议解析问题,提出了报文数据场的信号色块表示方法和起始位转换公式,并基于 C# 语言设计和开发了 DBC 数据管理软件,将 CAN 网络协议描述为一个 dbc 格式的文件。实车试验表明:所设计开发的 DBC 数据管理软件实现了 CAN 网络的协议解析、结构表征和数据管理,进一步验证了所提出 DBC 解析方法的有效性。

## 1 DBC 文件规范

根据《DBC File Format Document》规范,DBC 协议定义了网络节点、报文帧、信号(图 1,其中“L”符号表示一个空格符),它们对 CAN 网络的结构信息和属性信息进行了描述。

### 1.1 结构信息

图中以“BU\_”开头的行定义了节点信息,包括 Engine、Display、Light 3 个节点;以“BO\_”开头的行定义了报文信息,包括 EngineState、LightState2 条报文;每条报文下又定义了所包含的信号信息,以“SG\_”开头,例如,EngineState 报文下定义了 EngineSpeed 和 OnOff 2 条信号。

```

BU_␣Engine␣Display␣Light      → 网络节点定义,用BU_表示网络节点
BO_␣100␣EngineState:␣3␣Engine  → 报文帧定义,用BO_表示报文
␣SG_␣EngineSpeed:␣0|16@1-␣(0.25,0)␣[0|16383.5]␣"rpm"␣Display
␣SG_␣OnOff:␣16|1@0+␣(1,0)␣[0|1]␣"␣"␣Display

BO_␣120␣LightState:␣1␣Light
␣SG_␣FlashLight:␣2|1@1+␣(1,0)␣[0|1]␣"␣"␣Display → 信号定义,用SG_表示信号
␣SG_␣HeadLight:␣0|1@1+␣(1,0)␣[0|1]␣"␣"␣Display

```

图 1 DBC 示例文件局部结构

Fig. 1 Part of a DBC sample file

### 1.2 属性信息

BO\_ MessageId(10 进制数表示) MessageName: MessageSize Transmitter 为报文的格式模板,依据该模板,图中 BO\_ 100 EngineState: 3 Engine 就可以解析成一条由 Engine 节点发送,数据域长度为 3 字节,ID

为 100(0x64),名字为 EngineState 的报文。同样,信号按照该规范对其属性进行了描述。

因此依据上述 DBC 文件规范可以通过适当的解析方法让计算机快速对 CAN 协议进行解析。

### 2 DBC 文件的正则表达式解析

DBC 文件含有“@”“(”“)”“|”“[”“]”、空格等多种字符信息,并且内容信息不固定,不存在通用的解析技术,需要进行专门的解析方法设计。考虑到正则表达式在字符串信息匹配方面的方便性,基于 DBC 文件规范,设计了 DBC 文件中 CAN 协议节点 Node、报文 Message、信号 Signal 对应的正则表达式,分别为 regex1、regex2、regex3。

```
Regex regex1 = new Regex("^ BU_:\\s+([\\w+\\s+]+)");
```

```
Regex regex2 = new Regex("^ (BO_\\s*)(\\d+)(\\s*)(\\w+)(:)(\\s*)(\\d+)(\\s*)(\\w+)$");
```

```
Regex regex3 = new Regex("SG_\\s*(\\w+\\s*\\w*)\\s*:\\s*(\\d+)\\|\\|(\\d+)@[([10]) ([+-])\\s*\\|((- *\\d*\\. *\\d*),(- *\\d*\\. *\\d*))\\s*\\|((- *\\d*\\. *\\d*)\\|((- *\\d*\\. *\\d*)\\|((- *\\d*\\. *\\d*)\\|)\\s*\\|\"(.*)[? \\"]\\s+([\\w+, *]+)$");
```

利用 regex1、regex2、regex3 对 DBC 文件中每行信息进行匹配,判断该行信息描述的是节点、报文还是信号等,利用字符分割技术获取其中描述的具体属性值,从而创建 CAN 网络中对应的节点、报文、信号等实例对象,并建立起各对象之间的结构关系。为了方便计算机对 CAN 网络进行描述,通过多叉树<sup>[15]</sup>对其进行结构表征,最终得到如图 2 所示的多叉树结构模型。

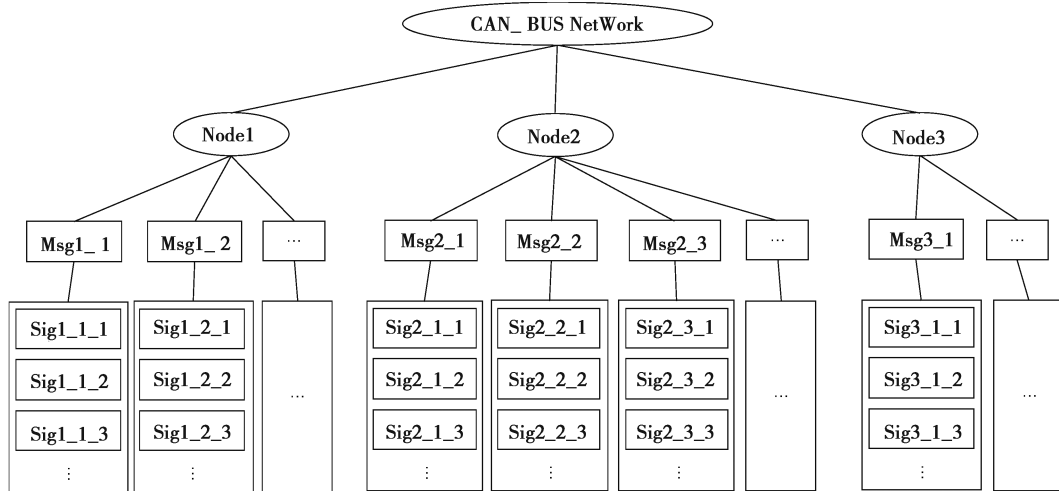


图 2 多叉树结构模型图

Fig. 2 Diagram of multi-tree structure model

该模型描述了一个完整的 CAN 网络工程,其中包括若干个节点(Node1、Node2、Node3 等),每个节点可以发送若干报文,每条报文由若干信号组成。

### 3 DBC 数据管理软件开发

DBC 数据管理软件的目的是对 CAN 网络进行管理,并将 CAN 网络协议以 DBC 数据管理文件的形式输出。通过可视化窗体设计和交互式操作可以直观地对 DBC 文件进行创建、修改和编辑。考虑到 C# 运行在 .NET 框架之下,开发出来的程序可以直接在 Windows 系统下使用,因此基于 C# 语言进行 DBC 数据管理软件开发。

#### 3.1 主界面和功能的开发

窗体 DBCViewer 作为主界面(图 3),左侧的 TreeView 树控件展示了 CAN 网络的结构关系,通过如下代码为 TreeView 添加了点击和拖拽事件监听的响应函数:

```

this.dbcTreeView.ItemDrag += new ItemDragEventHandler(this.DbcTreeView_ItemDrag);
this.dbcTreeView.NodeMouseClick += new TreeNodeMouseClickEventHandler(this.DbcTreeView_NodeMouseClick);
this.dbcTreeView.DragDrop += new DragEventHandler(this.DbcTreeView_DragDrop);
this.dbcTreeView.DragEnter += new DragEventHandler(this.DbcTreeView_DragEnter)。

```

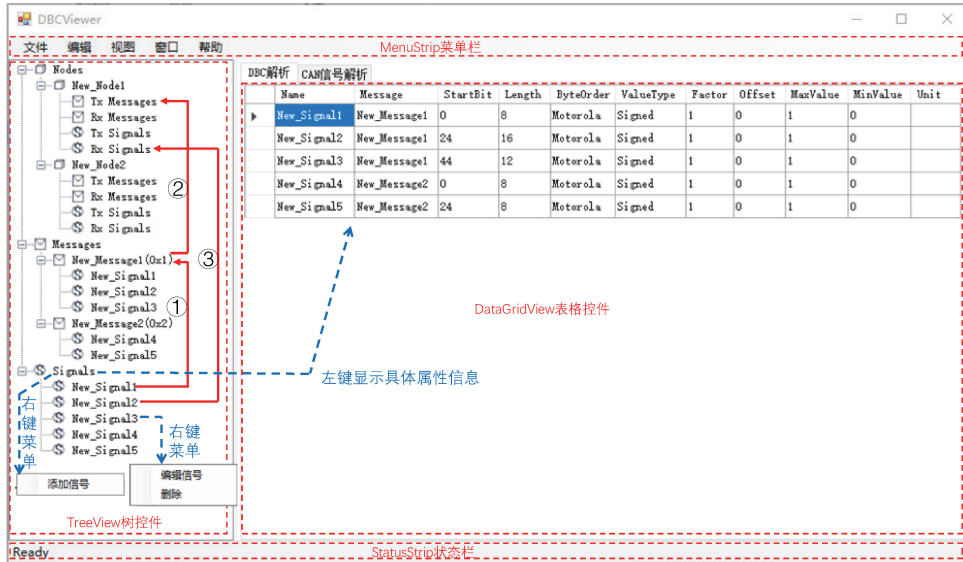


图 3 主功能界面

Fig. 3 Main function interface

TreeView 的拖拽事件用于建立 CAN 网络的结构关系,具体分为图中①②③3 种情况:

- ① Signals 树节点下的信号拖拽到 Messages 树节点下的报文中,确定该报文下的信号;
- ② Messages 树节点下的报文拖拽到 TxMessages 树节点下,确定该报文的发送节点;
- ③ Signals 树节点下的信号拖拽到对应 Node 的 RxSignals 树节点下,确定该信号的接收节点。

### 3.2 编辑功能开发

通过 SignalWindow、MessageWindow、NodeWindow 窗体实现 Signal、Message、Node 对象的创建和修改。图 4 通过设置标签的 BackColor 形成连续色块来模拟连续的信号占位,当左侧表格中信号的起始位变化时,根据 Intel 和 Motorola 的格式,整个数据场将自动调整色块布局。具体的调整程序如图 5 所示。

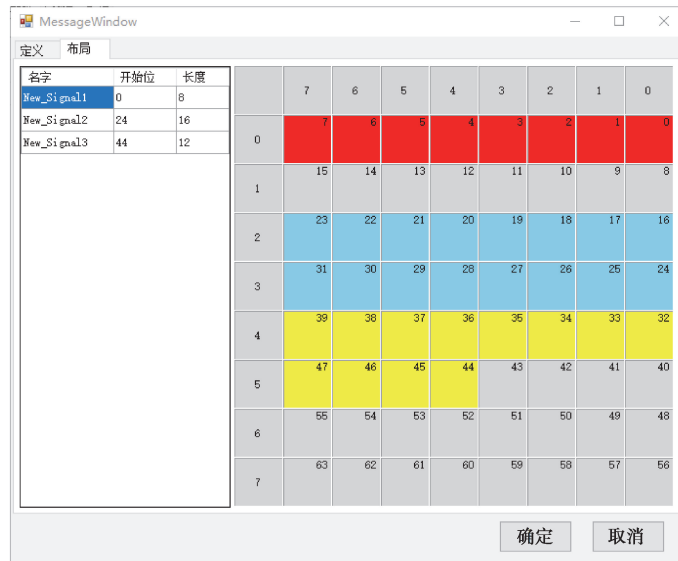


图 4 报文数据场的信号布局界面

Fig. 4 Signal layout interface of the message data field

```

Color[] colors = new Color[] { // 定义颜色选择器
    Color.Red, Color.SkyBlue, Color.Yellow, Color.Chartreuse,
    Color.Brown, Color.Violet, Color.SandyBrown, Color.Pink
};
foreach (DataGridViewRow row in dataGridView1.Rows) { // 对表格中每一行进行遍历
    int startbit = int.Parse(row.Cells[1].Value.ToString());
    int length = int.Parse(row.Cells[2].Value.ToString());
    int position = startbit;
    for (int i = 0; i < length; i++) { // 对当前信号中的每一位对应的Label设置背景色
        Controls.Find("Label" + position, true)[0].BackColor = colors[row.Index % colors.Length];
        if (curByteOrder.Equals("Intel")) { position++; } // 如果为Intel格式
        else if (curByteOrder.Equals("Motorola")) { // 如果为Motorola格式
            if ((position + 1) % 8 == 0) { // 如果达到边界
                position -= 15;
            } else { position++; }
        }
    }
}
}

```

图 5 报文数据场的色块布局调整程序

Fig. 5 Color block layout adjustment program of the message data field

通过上述程序和取余算法,实现了报文数据场的色块布局调整,保证不同信号使用不同的颜色标记,相邻信号之间不会干扰。

### 3.3 Motorola 格式下的起始位转换

DBC 文件中信号矩阵的编码格式分为 Intel 和 Motorola 两种。一般情况下,将 CAN 报文抽象成  $8 \times 8$  的数据场(图 6),每行表示一个字节的 8 个位,从右到左依次是 bit0, bit1, ..., bit7, 左边代表高位;从上到下依次是 byte0, byte1...byte7, 下边代表高字节。根据 CAN 报文的发送规则,通常情况下,先发送低位字节 LSB,后发送高位字节 MSB;而单个字节则是先传输高位 msb,后传输低位 lsb。如图 6 所示,左边采用 Intel 编码格式,右边采用 Motorola 编码格式,通过比较和分析总结出:

Name	Message	Startbit	Length	ByteOrder
New_Signal_1	Ignition_Info	10	8	Intel
New_Signal_2	Ignition_Info	54	8	Intel

Name	Message	Startbit	Length	ByteOrder
New_Signal_1	Ignition_Info	10	8	Motorola
New_Signal_2	Ignition_Info	54	8	Motorola

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
byte0	7	6	5	4	3	2	1	0
byte1	15	14	13	12	11	10	9	8
byte2	23	22	21	20	19	18	17	16
byte3	31	30	29	28	27	26	25	24
byte4	39	38	37	36	35	34	33	32
byte5	47	46	45	44	43	42	41	40
byte6	55	54	53	52	51	50	49	48
byte7	63	62	61	60	59	58	57	56

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
byte0	7	6	5	4	3	2	1	0
byte1	15	14	13	12	11	10	9	8
byte2	23	22	21	20	19	18	17	16
byte3	31	30	29	28	27	26	25	24
byte4	39	38	37	36	35	34	33	32
byte5	47	46	45	44	43	42	41	40
byte6	55	54	53	52	51	50	49	48
byte7	63	62	61	60	59	58	57	56

BO_0 Ignition_Info: 8 Vector_XXX	DBC文件
SG_New_Signal_2: 10 8@1- (1,0) [0 0] ** Vector_XXX	
SG_New_Signal_1: 54 8@1- (1,0) [0 0] ** Vector_XXX	

BO_1 Ignition_Info: 8 Vector_XXX	DBC文件
SG_New_Signal_2: 45 8@0- (1,0) [0 0] ** Vector_XXX	
SG_New_Signal_1: 1 8@0- (1,0) [0 0] ** Vector_XXX	

图 6 跨字节时 Intel 和 Motorola 格式的布局对比

Fig. 6 Comparison of the layouts of Intel and Motorola formats when crossing bytes

- 1) 数据场中,单个信号的起始位均为最低位(即 lsb);
- 2) 数据场中,单个信号的水平填充方向均是从右至左(即 lsb 至 msb 的方向);
- 3) 数据场中,当跨字节时,Intel 是自上而下填充, Motorola 是自下而上填充;
- 4) Motorola 格式下, DBC 文件中信号的起始位与数据场中的起始位不同,是数据场中信号的最高位

msb,即大端方式,因此在对 DBC 文件进行解析和保存时须进行起始位转换。

在进行程序的开发过程中,针对 Motorola 格式进行特殊处理,其中解析 DBC 文件时按照式(1)计算,保存 DBC 文件时按照式(2)计算。式(1)(2)中“/”表示取商运算符,“%”表示取余运算符,length 表示信号的长度,lsb 表示信号的最低位,等于数据场中的起始位,msb 表示信号的最高位,等于文件中的起始位。

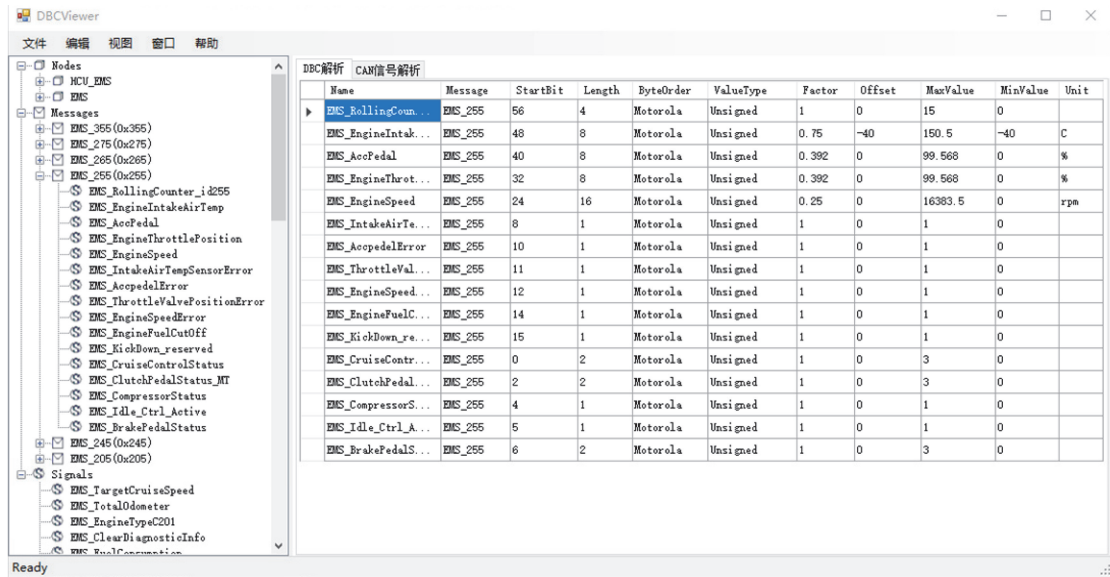
$$\begin{cases} lsb = msb - length + 1, (length \leq msb \% 8 + 1); \\ lsb = 16 \{ [length - msb \% 8 + 6] / 8 \} + msb - length + 1, (length > msb \% 8 + 1). \end{cases} \quad (1)$$

$$\begin{cases} msb = lsb + length - 1, (length \leq 8 - lsb \% 8); \\ msb = lsb + length - 1 - 16 \{ [length + lsb \% 8 - 7] / 8 \}, (length > 8 - lsb \% 8). \end{cases} \quad (2)$$

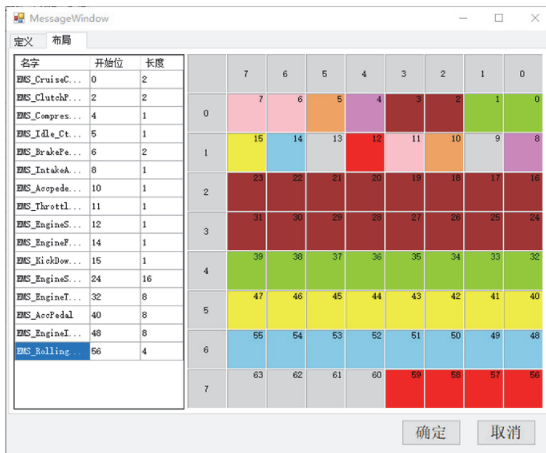
在 Intel 格式下,DBC 文件中信号的起始位就代表信号的最低位,即小端方式,故不存在转换的问题。

### 3.4 DBC 文件输出

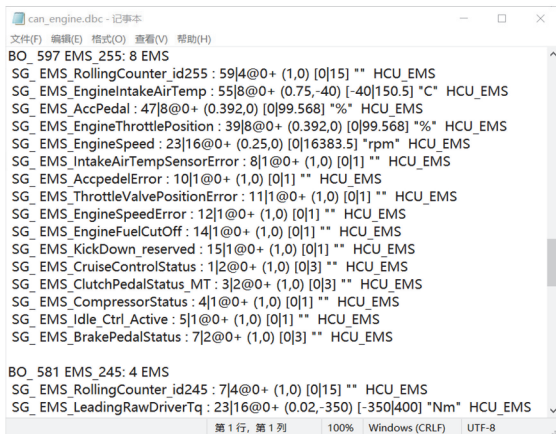
以装车发动机的 CAN 通信矩阵为例,将汽车 CAN 网络协议描述为一个具体的 DBC 文件。如图 7(a)所示,共创建了 HCU\_EMS、EMS 2 个节点,EMS\_205、EMS\_245、EMS\_255、EMS\_265、EMS\_275、EMS\_355 6 条报文,71 个信号。通过 TreeView 树节点拖拽,将 Node、Message、Signal 之间关联起来。以 EMS\_255 报文为例,该报文中添加了 EMS\_RollingCount\_id255、EMS\_EngineIntakeAirTemp 等 16 个信号,在 MessageWindow 窗体的布局中,调整每个信号的起始位来对报文数据场中的信号进行布局排列,调整之后的信号颜色块分布情况如图 7(b)所示。



(a)



(b)



(c)

图 7 DBC 文件制定  
Fig. 7 DBC file production

通过上述 Node、Message、Signal 的创建和关联操作之后,后台程序将这些信息自动同步到 CAN 网络的多叉树结构中。当点击“文件”-“保存”后,程序依据多叉树的广度搜索算法和深度搜索算法对多叉树结构进行遍历,按照 DBC 文件规范将 CAN 网络协议信息写入到 can\_engine.dbc 文件(图 7(c))。

### 4 实车试验

为验证 DBC 协议解析方法的有效性,首先对 DBC 数据管理软件输出的 can\_engine.dbc 文件信息进行验证:1)将文件信息与软件解析界面信息进行比较,相当于图 7(c)与图 7(a)、7(b)中的信息进行比较,发现 DBC 协议属性信息和结构信息均一致,并且 Intel 和 Motorola 格式下的起始位实现了正确转换;2)根据装车发动机的 CAN 通信矩阵,利用 CANoe 输出 can\_engine\_canoe.dbc 文件,并与 can\_engine.dbc 文件比对,节点、报文、信号等信息均一致。初步表明了所提出的正则表达式解析、多叉树结构表征、报文数据场信号色块表示方法和不同格式下起始位转换公式的正确性。

然后将 can\_engine.dbc 在某乘用车上进行了实车试验,同时基于 can\_engine\_canoe.dbc 进行对比,通过汽车 CAN 总线上报文采集和解析结果对比进一步验证。实车 CAN 网络自带报文信息,而且报文数据和种类较多,与仿真环境相比提供了更可靠准确的报文来源。

将 USBCAN-II C 和 Vector 硬件设备同时连接到汽车 OBD 接口,电脑打开对应设备的软件,配置好通道和参数。在试验环境准备好之后,驾驶车辆在重庆大学 A 区校园内(图 8)行驶,副驾驶进行汽车 CAN 报文的采集工作(图 9)。根据采集到的数据报文 ID(图 10,此处为 USBCAN-II C 设备采集到的部分数据截图)发现,除了发动机的 CAN 报文外,还包括其他节点发送的报文 ID。



图 8 汽车驾驶路段

Fig. 8 Car driving section



图 9 汽车 CAN 报文采集

Fig. 9 Car CAN message collection

时间(s)	通道	报文ID	报文类型	报文格式	方向	长度	数据
20.43406	CAN1	0x0237	数据帧	标准帧	接收	0x08	FE FE 63 00 04 69 FF C3
20.4343	CAN1	0x0258	数据帧	标准帧	接收	0x08	FF 50 19 00 29 00 00 9F
20.43456	CAN1	0x0208	数据帧	标准帧	接收	0x08	00 00 00 00 00 00 00 00
20.43479	CAN1	0x0268	数据帧	标准帧	接收	0x08	82 00 09 2E 00 49 46 1E
20.43504	CAN1	0x0338	数据帧	标准帧	接收	0x08	FF FF FF FF 77 FF FF FF
20.43657	CAN1	0x0205	数据帧	标准帧	接收	0x08	10 0B 09 06 FF 0D E0 64
20.43684	CAN1	0x0255	数据帧	标准帧	接收	0x08	64 00 10 97 11 00 61 04
20.43709	CAN1	0x0265	数据帧	标准帧	接收	0x08	4A 00 00 47 00 0D 28 A4
20.43823	CAN1	0x0288	数据帧	标准帧	接收	0x08	00 00 84 00 00 09 80 00
20.44171	CAN1	0x0180	数据帧	标准帧	接收	0x08	00 1B 00 04 00 00 03 B7
20.44194	CAN1	0x0190	数据帧	标准帧	接收	0x08	0A 00 44 5C 75 30 14 43
20.44217	CAN1	0x0210	数据帧	标准帧	接收	0x08	8F 73 FE FE 12 1F 16 69
20.4424	CAN1	0x0220	数据帧	标准帧	接收	0x08	45 AD 45 97 45 AD 28 09
20.44264	CAN1	0x01D0	数据帧	标准帧	接收	0x08	28 00 00 09 44 5C 44 5C
20.44335	CAN1	0x0230	数据帧	标准帧	接收	0x08	02 EA 13 88 FE 80 00 00
20.44361	CAN1	0x02A0	数据帧	标准帧	接收	0x08	01 00 7E 00 00 00 00 00
20.44385	CAN1	0x0218	数据帧	标准帧	接收	0x08	10 3E 83 E8 00 00 20 00
20.44409	CAN1	0x0237	数据帧	标准帧	接收	0x08	FE FE 63 00 04 69 FF C4
20.44433	CAN1	0x0258	数据帧	标准帧	接收	0x08	FF 60 19 00 29 00 00 AF
20.44456	CAN1	0x0268	数据帧	标准帧	接收	0x08	82 00 09 2E 00 49 46 1E
20.44655	CAN1	0x0205	数据帧	标准帧	接收	0x08	10 0B 09 06 FF 0D E0 65
20.44687	CAN1	0x0255	数据帧	标准帧	接收	0x08	64 00 10 97 11 00 61 05
20.44712	CAN1	0x0265	数据帧	标准帧	接收	0x08	4A 00 00 47 00 0D 28 A5
20.44737	CAN1	0x0275	数据帧	标准帧	接收	0x08	60 04 C9 87 69 3E 00 00

图 10 采集到的报文数据

Fig. 10 Collected message data

最后通过 can\_engine.dbc 文件对采集的报文进行解析,其中发动机转速位于 ID 为 0x255 的报文下,采用 Motorola 编码格式,起始位为 24,长度为 16,Factor 因子为 0.25,Offset 偏移量为 0;车速信号位于 ID 为 0x265 的报文下,采用 Motorola 编码格式,起始位为 16,长度为 13,Factor 因子为 0.05625,Offset 偏移量为 0。另外也通过 can\_engine\_canoe.dbc 文件对 Vector 硬件设备采集到的数据进行解析。rpm、v 为本文所开发软件生成 can\_engine.dbc 文件解析后的转速和车速;rpm'、v' 为采用 CANoe 工具软件生成的 can\_engine\_canoe.dbc 文件解析后的转速和车速,对全试验过程结果分别求取方差,结果如下:

$$\sum_{i=1}^{33\,500} (\text{rpm}_i - \text{rpm}'_i)^2 = 0, \quad (3)$$

$$\sum_{i=1}^{33\,500} (v_i - v'_i)^2 = 0. \quad (4)$$

计算结果均为 0,说明两种方式解析后的车速和发动机转速真实值是一致的;两种方式解析之后对应的曲线如图 11 和 12 所示。试验结果证明了本软件所制定的 DBC 文件能够正确解析汽车 CAN 报文。

XML 的解析方式首先将 DBC 格式的协议文件人工转换为 XML 格式的文件,然后计算机通过 XML 格式来识别和解析 CAN 网络协议。本研究中利用正则表达式来直接解析原 DBC 文件协议,利用多叉树的结构表征 DBC 协议,通过减少操作步骤,显著提高了 DBC 数据管理的效率,并且上述的试验结果证明了所提出方法的有效性。

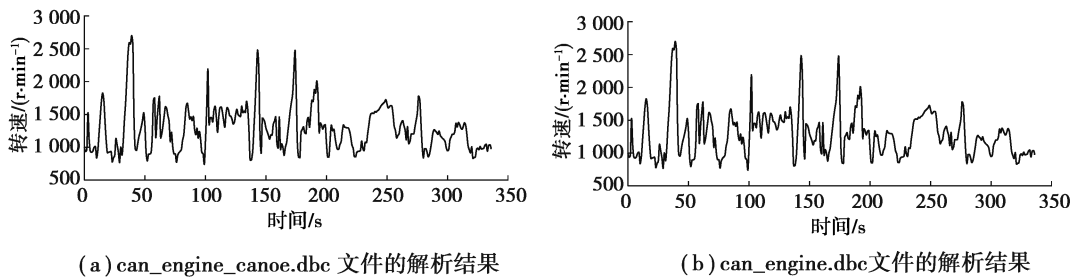


图 11 解析后的发动机转速曲线对比

Fig. 11 Analyzed engine speed curve comparison

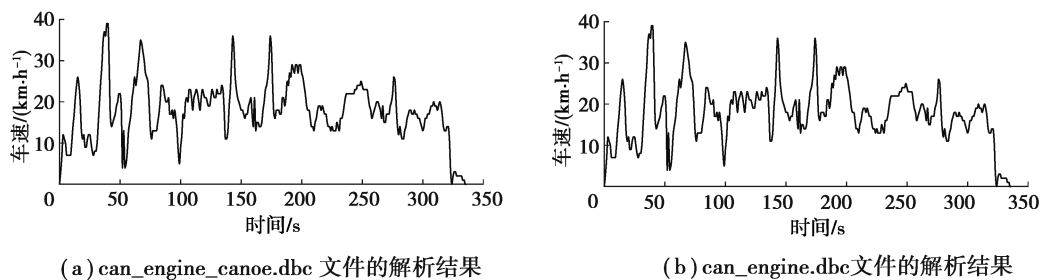


图 12 解析后的车速曲线对比

Fig. 12 Analyzed vehicle speed curve comparison

## 5 结 论

1)通过分析 DBC 文件规范,总结出 Intel、Motorola 编码格式下 CAN 报文的布局规律并推导出 Motorola 格式下起始位转换公式,进一步设计和开发出 DBC 数据管理软件;该软件能够以可视化窗口的交互方式创建和修改 DBC 文件。

2)通过正则表达式的设计实现了对已有 DBC 文件的解析,提出的 CAN 网络多叉树结构模型能够对 CAN 网络信息进行统一管理。

3)通过实车试验,所开发软件根据汽车 CAN 通信矩阵制定出来的 DBC 文件与 CANoe 生成的 DBC 文件进行汽车 CAN 报文的对比试验解析,两者结果一致,验证了所开发软件的实用性。



## 参考文献:

- [1] 孔峰, 张衡, 宋雪桦, 等. 基于 CANopen 协议的汽车控制网络初探[J]. 汽车工程, 2007, 29(7): 594-596, 605.  
Kong F, Zhang H, Song X H, et al. A preliminary investigation into vehicle control network based on CANopen protocol[J]. Automotive Engineering, 2007, 29(7): 594-596, 605. (in Chinese)
- [2] 黄赛杰, 徐敏, 郑小鹿, 等. 动力电池充放电检测系统的设计与实现[J]. 储能科学与技术, 2019, 8(1): 146-154.  
Huang S J, Xu M, Zheng X L, et al. Design and implementation of power battery charging and discharging detection system[J]. Energy Storage Science and Technology, 2019, 8(1): 146-154. (in Chinese)
- [3] Cai Q J, Xu Y, Fu J K, et al. Study on development and testing of vehicle CAN network platform based on CANoe[J]. Applied Mechanics and Materials, 2014, 687/688/689/690/691: 56-59.
- [4] 杨璐激, 陈立涛, 吕妮娜. 基于 CAN 总线的整车电子电气架构的研究[C]//2015 中国汽车工程学会年会论文集. 上海: 中国汽车工程学会, 2015: 280-283.  
Yang L J, Chen L T, Lv N N. Research on vehicle electronic and electrical architecture based on CAN bus[C] // Proceedings of 2015 annual meeting of China Society of Automotive Engineering. Shanghai: China Society of Automotive Engineering, 2015: 280-283. (in Chinese)
- [5] 张新波, 孙泽昌, 罗峰. 使用 CANoe 对车身控制器局域网络仿真的研究[J]. 江苏大学学报(自然科学版), 2003, 24(5): 36-39.  
Zhang X B, Sun Z C, Luo F. Study on CAN body network simulation with CANoe[J]. Journal of Jiangsu University (National Science Edition), 2003, 24(5): 36-39. (in Chinese)
- [6] 丁志华, 罗峰, 孙泽昌. 基于 CANoe 的汽车故障诊断系统研制[J]. 汽车工程, 2007, 29(5): 449-452.  
Ding Z H, Luo F, Sun Z C. Development of vehicle fault diagnostic system based on CANoe[J]. Automotive Engineering, 2007, 29(5): 449-452. (in Chinese)
- [7] Li L J, Ye H J, Gao S. Development of instrument cluster simulation system based on canoe[J]. Advanced Materials Research, 2012, 490/491/492/493/494/495: 1198-1201.
- [8] 陈燕平, 殷承良, 张勇. 混合动力大客车动力总成试验台架的构建及试验研究[J]. 汽车工程, 2011, 33(6): 468-471.  
Chen Y P, Yin C L, Zhang Y. Construction and experimental study of test bench for the powertrain assembly of a hybrid electric bus[J]. Automotive Engineering, 2011, 33(6): 468-471. (in Chinese)
- [9] Zhang X B, Xu L F, Li J Q, et al. Real-time estimation of vehicle mass and road grade based on multi-sensor data fusion[C]// 2013 IEEE Vehicle Power and Propulsion Conference (VPPC), October 15-18, 2013, Beijing, China. IEEE, 2013: 1-7.
- [10] Han J, Liu Z, Cui S, et al. A study on the application of the controller area network communication protocol to hybrid electric vehicle[J]. Automotive Engineering, 2011, 33(12): 1062-982.
- [11] Sharma D, Gupta M. Controller area network for automobile application using ASIC based on PsoC and analysing through vector CANoe[J]. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2013, 2(5): 287-292.
- [12] Peng J K, Fan H, He H W, et al. A rule-based energy management strategy for a plug-in hybrid school bus based on a controller area network bus[J]. Energies, 2015, 8(6): 5122-5142.
- [13] Yi L, He H W, Peng J K. Hardware-in-loop simulation for the energy management system development of a plug-in hybrid electric bus[J]. Energy Procedia, 2016, 88: 950-956.
- [14] 魏学哲, 戴海峰, 孙泽昌. 汽车嵌入式系统开发方法、体系架构和流程[J]. 同济大学学报(自然科学版), 2012, 40(7): 1064-1070.  
Wei X Z, Dai H F, Sun Z C. Methodology, architecture and development flow of automotive embedded systems[J]. Journal of Tongji University (Natural Science), 2012, 40(7): 1064-1070. (in Chinese)
- [15] Zhang L M, Yang F, Lei Y. Tree-based intermittent connection fault diagnosis for controller area network[J]. IEEE Transactions on Vehicular Technology, 2019, 68(9): 9151-9161.

(编辑 罗 敏)