文章编号：2096-1472(2021)-01-39-03
DOI:10.19644/j.cnki.issn2096-1472.2021.01.011

高级驾驶辅助测试可视化系统开发

杨军典，陈凌珊

(上海工程技术大学机械与汽车工程学院，上海 201620)
✉yjd2008@hotmail.com; bechenlsh@163.com

摘　要：为了在高级驾驶辅助系统(ADAS)中对车辆控制算法进行验证，可以利用维克托(Vector)工具链对汽车电子控制器(ECU)内部的变量进行记录，生成MDF(原版光盘映像文件)格式的测试数据文件，然后进行离线分析。利用Qt公司开发的丰富的可视化图形控件Qt(跨平台软件开发框架)和Python(一种计算机程序设计语言)的强大数据处理能力，本文开发了一种可以对MDF文件进行解析并且可视化的软件。该软件可以实现变量的筛选功能，支持变量的树状显示、多个变量测量值的二维时域图形表示。特别地，ADAS感知到的环境目标，本软件可以观察其二维坐标随时间的运动过程，并且支持画面与用户的交互操作。经过测试，本软件运行可靠，在某公司推广使用。

关键词：ADAS；测试；MDF解析；Qt；可视化
中图分类号：TP311.52　**文献标识码**：A

Development of Advanced Driver Assistance Test Visualization System

YANG Jundian, CHEN Lingshan

(*Shanghai University of Engineering Science*, *School of Mechanical and Automotive Engineering*, *Shanghai* 201620, *China*)
✉yjd2008@hotmail.com; bechenlsh@163.com

Abstract: In order to verify the vehicle control algorithm in the Advanced Driver Assistance System (ADAS), Vector's tool chain can be used to record variables inside the automotive Electronic Control Unit (ECU). Then, a test data file in MDF (Measurement Data Format) format is generated, and offline analysis can be performed. This paper proposes to develop a software that can analyze and visualize MDF files by utilizing Qt (the cross-platform software development framework) and Python (a computer programming language) developed by Qt Company. This software can realize the variable filtering function, support tree displays of variables and two-dimensional time-domain graphical representation of measured values of the multiple variables. In particular, the proposed software can observe the movement process of its two-dimensional coordinates over time for the environmental targets that ADAS perceives, and it supports the interactive operation of screen and user. Testing results show that the proposed software runs reliably and is being used by a company.

Keywords: ADAS; testing; MDF analysis; Qt; visualization

1 引言(Introduction)

在ADAS系统车辆ECU软件的开发过程中，控制算法需要经过不断的设计、测试、修改迭代才能完成。在测试这一环节中，包括软件在环测试(SIL)、硬件在环测试(HIL)和实车测试三种。在实车测试过程中，需要提前将软件烧写入汽车ECU，在汽车行驶过程中，通过XCP协议[1]和Vector公司的CAN工具，ECU可以记录内部变量，然后生成符合自动化及测量系统标准协会(ASAM[2])标准的MDF[3]文件。通过将记录的MDF文件回传到算法开发人员手中，开发人员可以离线进行算法的验证工作。因此，MDF文件的解析和可视化对算法的设计开发非常重要。虽然Vector公司已经有对MDF文件进行解析和可视化的商业工具，但是其价格昂贵并且无法进行二次开发。本文的目标是设计一个可视化软件，可以解析MDF文件，并且对文件中的变量进行二维显示和三维的动态播放。

本文首先利用Python丰富的第三方工具包对MDF文件进行解析，然后使用Python的Numpy等数据科学包对MDF中的数据进行处理。由于需要存储的变量较多，查询、筛选频繁，因此选用小型的SQLite3关系数据库来组织文件中的数据。最后使用Qt丰富的控件库，实现软件的整体可视化界面设计。其中，在与用户的交互操作模块上，利用了强大的兼容Qt平台的PyQtGraph[4]图形库，加快了开发过程。

2 MDF文件介绍和解析(The introduction and parse of MDF file)

2.1 历史、版本及用途

MDF文件格式是20世纪90年代由欧洲的Bosch、Vector联合大众等汽车公司专门为汽车行业设计的一种通用文件格式，主要用于汽车电子ECU的开发、标定和测试领域，已经成为该领域事实上的工业标准。由于测量数据的快速增长，在ASAM组织修订下，MDF文件从2.0版本逐渐升级为目前的4.x版本，生成的数据文件后缀名为“.MF4”。MDF文件中记录的典型数据有汽车传感器信号、ECU内部变量状态等。这些存储在MDF文件中的状态信号，可以被算法人员回放，进行算法的验证工作。本文主要专注于最新的MDF 4.x版本文件(即*.MF4文件)的解析和可视化。

2.2 MF4文件介绍

一个MF4文件由若干个二进制模块组合而成，每个模块主要包含三个部分：文件头、链接部分和数据部分，如图1[5]所示。

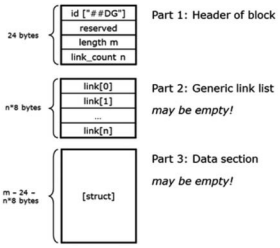

图1 数据块结构

Fig.1 The structure of each block

模块的类型有很多种，每种类型都定义了模块的用途和内容，由二位的大写字母表示。主要的模块类型有：ID(用于标识MDF文件)、HD(MDF文件的通用描述)、TX(记录变量长度的字符容器)、CH(定义通道的逻辑结构)、DG(描述数据分组)、CG(描述通道组)、CN(描述通道)、CC(描述每个通道数据的转换)、DT(每个数据记录的单个值)等。这些模块按照一定的结构存储起来，便可以组成一个MDF文件，图2描述了一个简单模块的物理存储和层次结构[5]。

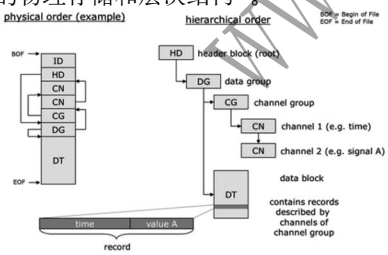

图2 一个简单的MDF文件模块示例

Fig.2 Example of a simple block structure of MDF files

由图2可知，所有的变量在逻辑上是按照树状层级分布的，一个CN相当于一个变量名，描述了测量值的存放位置、字节数等信息。同一个CG是若干个变量名的集合，并且有相同的采样率。DT是真正存储数据值的地方，其内部数据的布局如图3所示[5]。

图3 一个包含两个记录的DT模块示例

Fig.3 Example for a DT block containing two records

2.3 MDF文件解析

Python是一种动态的计算机语言，可以用于数据计算、软件开发等。mdfreader[6]是Python的一个第三方开源工具包，可以用于读取解析MDF文件格式，具体使用方法见文献[6]。

使用mdfreader解析出MDF文件获取的只是最原始的数据，需要按照一定的逻辑层次对数据进行筛选、整理、变换、存储等操作。对于数据处理这部分，可以利用Python中著名的Numpy包进行数学运算。由于解析出来的数据存储在内存中，不能永久保存，因此在下节中使用数据库技术来解决这一问题。

3 技术框架(Technical framework)

3.1 Qt和PyQt简介

Qt是一个著名的面向对象、跨平台的C++图形界面开发框架，可用于开发GUI程序，目前已经更新到5.13版本。由于在上文中使用Python语言解析MDF文件，为了避免混合语言编程，本文选用Python语言开发GUI界面。

PyQt是由Python和Qt库融合而成的一个工具包，允许开发者使用Python调用Qt库中的应用程序接口(API)创建GUI应用程序。在本软件开发中，使用与Qt5对应的PyQt5模块设计本软件。

3.2 SQLite3数据库

SQLite3是一个轻型，支持SQL语法的开源关系数据库。关系数据库使用多张二维表来存储数据，每张表可以通过主关键字唯一确定一行数据。SQL是用于访问和处理数据的标准计算机语言，通过在Python程序中调用SQL语言的接口，可以对SQLite3中的数据库进行表格创建、数据查询、筛选等，完成对MDF中测量数据的永久存储和读取。

3.3 PyQtGraph交互可视化库

前文介绍了使用PyQt5进行GUI应用程序的设计，但是Qt自带的图形库中进行用户交互的函数较少。PyQtGraph是一个基于PyQt/PySide和Numpy的纯Python开源图形GUI函数库，补充了Qt在数据图形与用户交互方面能力不足的缺点，并且提供了帮助快速开发应用程序的工具。因此，在本软件需要与用户交互的模块中，使用PyQtGraph代替Qt进行快速开发。

4 软件架构(Software architecture)

常用的软件架构一般分为三层：界面、服务和数据库。界面负责与用户的直接交互，需要布局合理、美观。服务是软件后台对用户界面中按钮的功能实现，不对用户开放。数据库是用来保存软件中所产生的大量数据，与服务进行交互，提供存储、提取功能。按照这一逻辑，设计本软件的架构层次如图4所示。其中Python调用SQLite3数据库的方法可以在参考文献[7]中进行查询，数据与服务端的交互逻辑在前文中进行了穿插介绍。

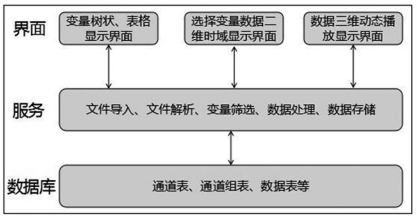

图4 软件层次

Fig.4 The software layers

4.1 界面设计

使用PyQt5写GUI界面可以通过直接手写Python代码实现，但更方便的办法是使用Qt Designer进行按钮的拖拉放置和布局设计。使用Qt Designer可以实现软件界面和逻辑的分离，加速开发速度，生成的*.ui文件可以通过PyQt5自带的pyuic5工具自动转换成对应的Python代码。因此，本文采用这种方法进行如图4所示三个界面的开发任务。

在变量树状、表格显示界面，需要实现文件的导入，后台服务将文件解析之后将文件内部的变量以树状的形式展示给用户。使用Qt Designer设计此界面如图5所示。当用户想要查看某些变量的值时，可以通过在树状图中选择某些变量，然后点击表格显示按钮，实现在右上侧的观察。界面的右下侧显示了被测试车辆的长宽高、传感器安装位置有关的几何静态参数。使用同样的方法设计其余两个界面，具体过程不再赘述。

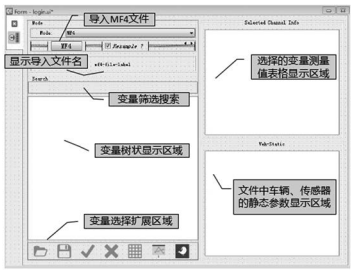

图5 变量树状、表格显示

Fig.5 The dendriform & tabular display of variables

4.2 服务实现

在界面的设计中实现了许多按钮，比如导入文件等操作，这些按钮背后的具体操作需要在服务中实现。对于实现点击按钮打开文件等简单的操作，可以在文献[7]中进行查询，以下着重阐述数据的树状、表格显示，二维时域显示和三维播放功能。

4.2.1 树状、表格显示

在Qt5上面用于界面组件显示与编辑的数据结构是模型/视图结构(Model/View)。视图是用来可视化和编辑数据的界面组件，模型是视图与数据可视化的接口，数据通过代理与编辑，可以把数据和界面分离开来，如图6[8]所示。Qt自带的QSqlQueryModel可直接执行SQL语句，从数据库中读取数据，然后使用QTableView将读取的数据以表格的形式显示出来，详细使用见文献[8]。由于Qt的数据结构只给出了QTreeView，因此树状数据显示的难点在于从QStandardItemModel基础上进行树状模型的定制。

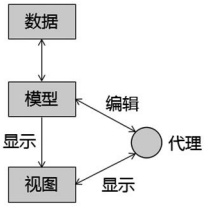

图6 模型/视图结构

Fig.6 The structure of model/view

QStandardItemModel是一种基于数据项的数据结构，由用户组织数据项的结构。数据项的类型是QStandardItem，保存着数据、文字、图标等显示信息，并且可以设置为是否可编辑、选取等。根据图2所示，将DG、CG、CN均表示成一个数据项，然后使用QStandardItemModel的setItem函数对数据项进行排布，形成树状结构。如图7所示为使用上述方法导入一个示例MF4文件后，变量的树状和表格显示结果。

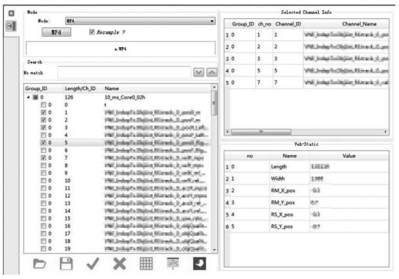

图7 一个MF4文件的树状、表格显示结果

Fig.7 The dendriform & tabular display of a MF4 file

4.2.2 二维时域显示

除了可以使用表格的形式显示变量外，更直观的方法是使用二维图形表示。当用户更新了选择的变量时，二维图形也可以动态地更新。除此之外，当用户点击图形中的某个采样点时，可以将采样点处的数据标注出来，并且可以自由地使用鼠标进行图形的缩放、平移。为了满足这些人机交互功能，这里采用PyQtGraph进行开发。

PyQtGraph工具库自带了支持鼠标交互操作的后台，因此，这一模块开发的重点在于如何动态地更新二维图形的数量和采样点处的数据显示。PyQtGraph工具库和Qt图形框架的本质是一样的，图形显示的原理是首先生成一块画布，所有的图形元素按照一样的布局在画布上放置。当用户选择的变量变化时，首先需要使用布局画布变量GraphicsLayout的clear函数清除画布中已有变量的二维图形，然后再生成新变量的二维图形。当鼠标点击曲线中的某个采样点时，使用PyQtGraph.CurvePoint建立一个锚点，然后以这个锚点为基准新建一个标签显示采样值。为了实现采样数据的动态更新，每次鼠标重新选择采样点时，从画布包含的图形元素中筛选并移除所有CurevePoint类型的锚点元素，最后再生成新采样点处的锚点和数据标签。如图8所示为某一MF4文件中选取三个变量的二维时域显示图形。

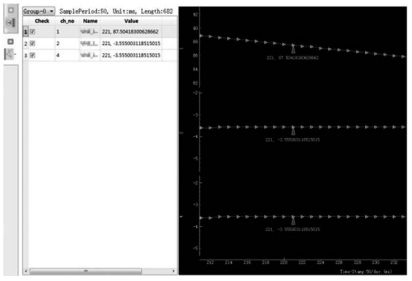

图8 二维时域显示

Fig.8 Display of two-dim time domain

4.2.3 三维播放功能

在ADAS中，汽车的传感器会将车辆周围环境中目标的坐标、速度、加速度等状态信息发送给ECU，然后控制算法会根据感知信息作出相应的动作。随着时间的不断前进，目标的二维坐标值会随着发生变化。因此，对这些信息，需要使用播放功能表达。

实现三维播放功能有两个难点：(1)如何在所有变量中提取环境目标的坐标值；(2)如何设置播放功能。每家公司的变量名都有一定的命名规范，因此通过查找公司的相关文件，对MDF文件中解析出来的变量名称建立符合环境目标命名规

(下转第25页)