**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN TỬ VIỄN THÔNG**

**TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:**

**THIẾT KẾ PHẦN MỀM MÔ PHỎNG CAN**

Sinh viên thực hiện: **NGUYỄN ĐỨC TUẤN – LỚP 12DT3**

**VÕ THANH BAN – LỚP 12DT1**

**TRƯƠNG CÔNG ĐÔNG – LỚP 12DT4**

Người hướng dẫn: **TS. TRẦN THỊ HƯƠNG**

**TS. TRẦN VIỆT THẮNG**

**TS. HỒ PHƯỚC TIẾN**

**Đà Nẵng, …../2017**

# **TÓM TẮT**

Tên đề tài: THIẾT KẾ PHẦN MỀM MÔ PHỎNG CAN

Sinh viên thực hiện: NGUYỄN ĐỨC TUẤN

Số thẻ sinh viên: 106120145 Lớp: 12DT3

Sinh viên thực hiện: VÕ THANH BAN

Số thẻ sinh viên: 106120009 Lớp: 12DT1

Sinh viên thực hiện: TRƯƠNG CÔNG ĐÔNG

Số thẻ sinh viên: 106120162 Lớp: 12DT4

Đề tài “Thiết kế phần mềm mô phỏng CAN” thuộc dự án "Thiết kế và thi công hệ thống mô phỏng CAN BUS” do nhóm sinh viên Khoa Điện Tử - Viễn thông tham gia thực hiện dưới hình thức Capstone Project với công ty FPT Software.

Phần mềm mô phỏng CAN được xây dựng với chức năng mô phỏng và hiển thị thông tin của signal và message qua các hộp thoại giao diện nhằm mô tả hoạt động của hệ thống mô phỏng CAN BUS.

Phần mềm có khả năng tương tác với phần cứng thông qua việc điều chỉnh tốc độ Baud từ các cổng CAN, tuyền tải thông tin từ cơ sở dữ liệu (message/signal) sau đó xử lí và hiển thị chúng thông qua các khối chức năng Interative Generator, Filter, Hardware và mô tả, hiển thị kết quả trên giao diện hộp thoại Trace. Giúp cho người sử dụng có thể dễ dàng kiểm tra, nhận xét thông tin của các dữ liệu được truyền lên từ thiết bị (CAN Device).

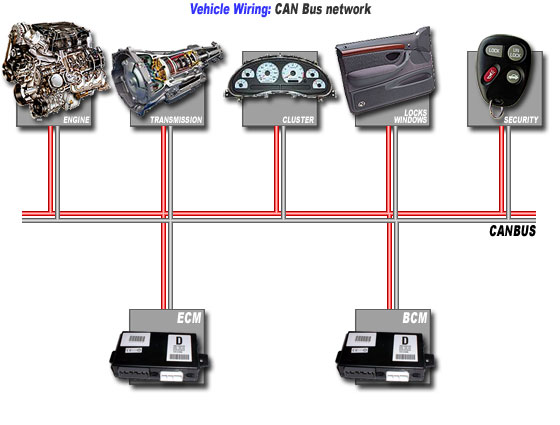
Phần mềm hỗ trợ cho cả 2 nền tảng Windows và Linux, giao diện được viết bằng ngôn ngữ C++ dùng Qt Framework.

# **CHƯƠNG 1: TỔNG QUAN HỆ THỐNG MÔ PHỎNG CAN BUS**

Chương này giới thiệu về đường truyền CAN và hệ thống mô phỏng CAN BUS, bao gồm: Phần mềm mô phỏng CAN, phần mềm cơ sở dữ liệu CAN, CAN Device và PC Driver.

# **Tổng quan về CAN BUS**

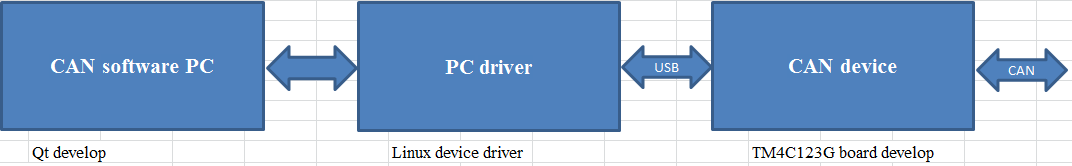
Mạng khu vực điều khiển CAN – Controller Area Network (hay còn gọi là CAN BUS) là mạng lưới đường truyền thông tin các bộ điều khiển độc lập dưới dạng đường truyền BUS. Trong đó các thành phần nối với mạng có quyền ngang nhau trong việc truyền và nhận thông tin (multi master), bất kỳ các thành phần đó đều có thể truyền cũng như nhận thông tin mà chúng cần từ các thành phần khác. Vì vậy, đây là một giao thức truyền thông nối tiếp có hỗ trợ hiệu quả phân phối thời gian thực kiểm soát với mức độ bảo mật rất cao.



HÌNH 1.1: Mô hình CAN BUS trong xe oto

# **Hệ thống mô phỏng CAN BUS**

Hệ thống mô phỏng CAN Bus giao tiếp các ECU trong ô tô sử dụng đường truyền CAN bus, bao gồm CAN Device thực hiện hoạt động truyền nhận thông điệp tương tự như một ECU thực tế,PC Driver có nhiệm vụ giao tiếp giữa Hardware và Software và CAN Simulator là phần mềm mô phỏng, hiển thị định dạng của thông điệp, tín hiệu, đồng thời có thể cấu hình chọn lựa cổng CAN, tốc độ baud…



HÌNH 1. 3 : Sơ đồ khối mô tả giao tiếp hệ thống CAN BUS

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ MÔI TRƯỜNG PHÁT TRIỂN**

Thiết kế phần mềm mô phỏng CAN là nhiệm vụ chính của nhóm trong đề tài “THIẾT KẾ VÀ THI CÔNG HỆ THỐNG MÔ PHỎNG CAN BUS”.

Chương này tập trung trình bày cơ sở lý thuyết cũng như môi trường phát triển của phần mềm mô phỏng CAN, bao gồm Thiết kế chương trình theo hướng đối tượng, lựa chọn hệ điều hành Linux và môi trường phát triển tích hợp IDE - Qt Creater Framework.

Sử dụng kiến trúc phần mềm - Mô hình Model/View cùng với cơ chế Signal và Slot trong QT Framework và các lớp như: QMainWindow, QDialog, QMenu, QAction, QPushbutton,… để thiết kế giao diện và luồng dữ liệu của phần mềm mô phỏng CAN.

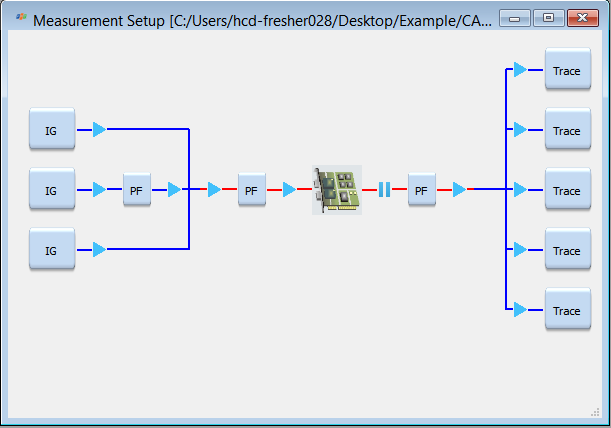
# **Cấu hình dữ liệu:** JSON là một định dạng hoán vị dữ liệu nhanh, chúng dễ dàng cho chúng ta đọc và viết , được xây dựng trên cấu trúc Key- Value, chúng có thể là một Object, Array, Value, String, Number để mô tả dữ liệu của User Interface. Ta sử dụng Qjson trong QT Framework để định dạng cấu hình dữ liệu.

# **CHƯƠNG 3: KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ**

Chương này sẽ trình bày tiến trình thực hiện đề tài “Thiết kế phần mềm mô phỏng CAN”. Gồm các bước thiết kế, phân chia công việc và thi công cũng như đánh giá, nhận xét kết quả được đạt của đề tài.

# **Giao diện chính Simulator software.**

Giao diện chính Measurement Setup phần mềm mô phỏng CAN được thiết kế dựa trên các lớp QPushButton, QLine, QLayout, QPainter…Giao diện chính thể hiện các khối và các nhánh liên kết với nhau

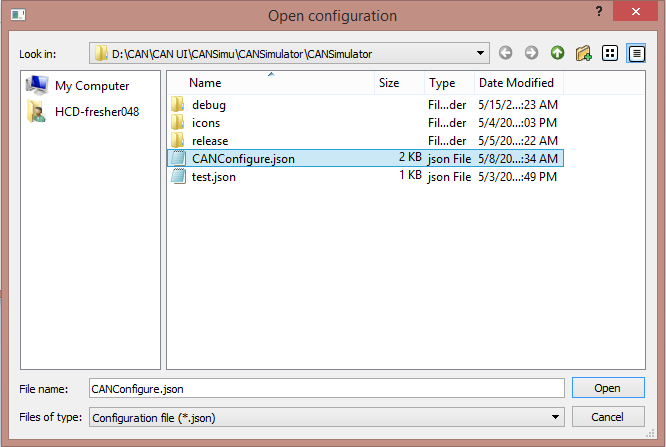


HÌNH 3. 1:Giao diện Measurement Setup phần mềm mô phỏng CAN

# **Hoạt động phần mềm mô phỏng CAN**

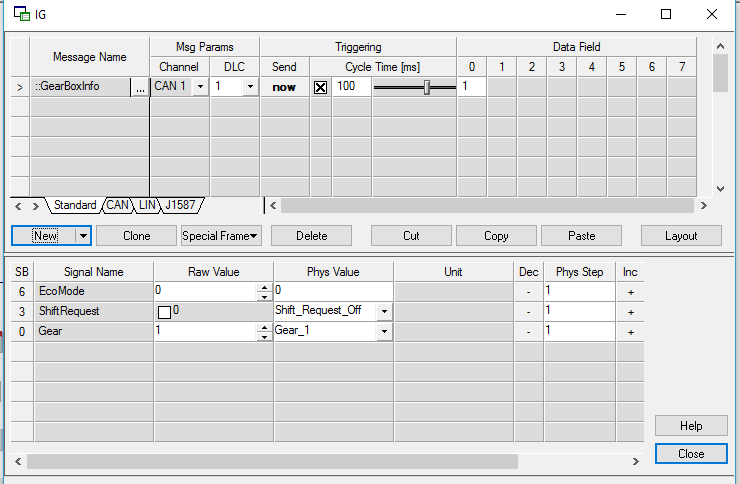
Hoạt động phần mềm mô phỏng CAN theo trình tự sau:

* Để khởi động phần mềm, ta thực hiện click vào File/Open configuration để load file cấu hình với định dạng “json” nhằm hiển thị giao diện Measure setup của hệ thống.



HÌNH 3. 21: Load file cấu hình với định dạng Json

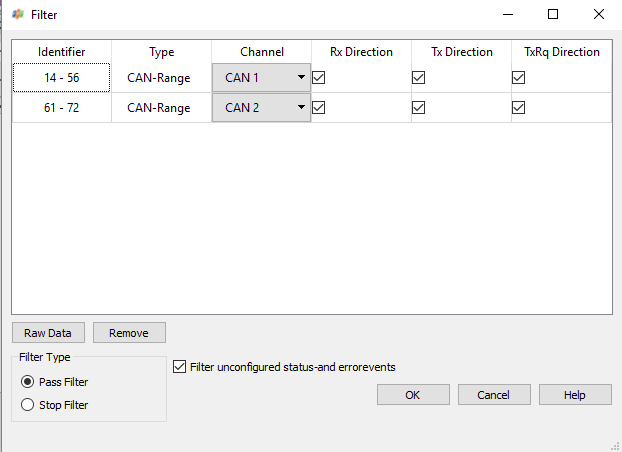
* Phần mềm tương tác với các message và signal từ cơ sở dữ liệu có sẵn (New/Open) thông qua hộp thoại giao diện chức năng IG (interative Generator). Hiển thị các giá trị trường Data, Channel và DLC. Người dùng cũng có thể điều chỉnh các thông số như :Cycle time [ms] của message, Raw Value và Phys Value của signal, đồng thời hiển thị wareform cho phù hợp.



HÌNH 3. 22 Tương tác với các Message/Signal trong hộp thoại IG

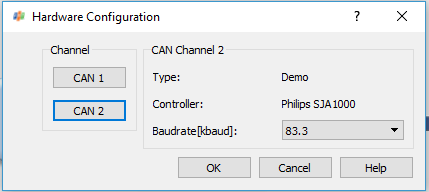
* Dữ liệu (các gói signal) sau khi được xử lý trong hộp thoại IG sẽ được đóng gói và tiếp tục gỡi đến khối Filter nhằm giảm bớt kích thướt của khung frame dữ liệu, rút ngắn thời gian xử lý thông qua việc thiết lập các thông số Start ID và End ID trong hộp thoại Data Raw. Dữ lieu sẽ được so khớp với ID (vừa thiết lập) của khối lọc, nếu trùng khớp chúng sẽ tiếp tục được gửi đi đến trung tâm để sử lý, những message có ID nằm ngoài giá trị ID thiết lập sẽ bị chặn lại (không được xử lí và đóng gói).

Bên cạnh đó, chúng ta cũng có thể lựa chọn kênh (CAN 1 hoặc CAN 2) và các thuộc tính mong muốn cho việc truyền nhận (Rx, Tx, TxRq).



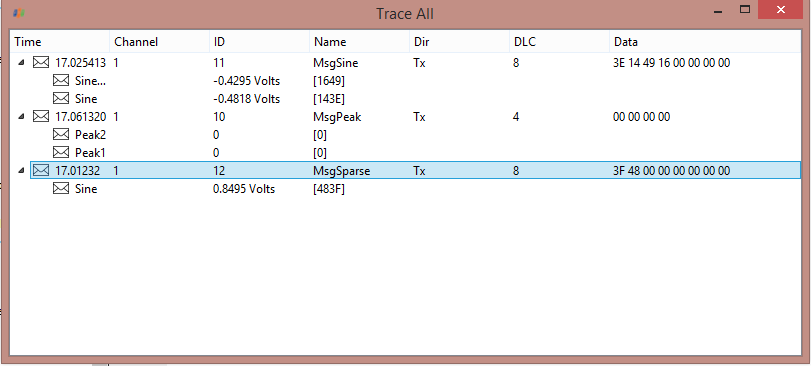
*HÌNH 3. 23 :Dữ liệu được xử lí trong khối Filter*

* Gói signal sau khi được xử lý ở khối IG và Filter sẽ được đóng gói và tiếp tục chuyển đến khối trung tâm (khối Hardware). Tại đây ta có sẽ chọn kênh, thiết lập tốc độ baudrate [kbaud] bằng việc chọn các tốc độ tiêu chuẩn đã giao diện hộp thoại (33.3, 50, 83.3, 100, 125, 500). Hardware sẽ truyền những signal này xuống Board để xử lý, nếu những signal này là hợp lệ board sẽ gỡi trả lại khối Hardware để đóng gói và truyền đến khối Trace, còn nếu signal nào gặp lỗi trên đường truyền thì sẽ được chờ để xử lý. Nếu quá thời gian chờ thì Board sẽ gửi phản hồi trở lại Hardware (No) và tiến hành xử lí những signal tiếp theo.



*HÌNH 3. 24 Thiết lập tốc độ Baudrate của cổng CAN*

* Dữ liệu sau khi được phản hồi lại Hardware sẽ được truyền đến khối Trace để mô tả và hiển thị các thông tin như Time, Channel, ID (Identifier), Tx, Rx, DLC,…của message, signal trong message để giúp người sử dụng có thể kiểm tra, dễ dàng nhận xét, đánh giá kết quả sau những thiết lập và xử lý khi luồng dữ liệu khi qua các khối IG, Filter, Hardware.



HÌNH 3. 25 Cửa sổ Trace hiển thị thông tin của message và signal

Đây cũng là mục đích chính của nhóm khi xây dựng phần mềm này.