

7. SOFTWARE COMPONENTS

Giới thiệu

Lớp ASW (Application Software) trong Autosar được tổ chức thành các đơn vị tự quản gọi là **thành phần phần mềm nguyên tử (atomic software component types)**. Những thành phần này đại diện cho các triển khai chức năng của ứng dụng và được phân loại thành ba loại chính:

Thành phần phần mềm nguyên tử (Atomic Software Component)

Thành phần phần mềm tham số (Parameter Software Component)

Thành phần phần mềm kết hợp (Composition Software Component)

Dưới đây là các loại thành phần phần mềm nguyên tử khác nhau và cách sử dụng của chúng:

Thành phần phần mềm ứng dụng (Application Software Component)

. **Vai trò:** Được sử dụng rộng rãi nhất, chứa các chức năng của phần mềm như tính toán, ra quyết định.

. **Ví dụ:** Thuật toán tính toán và quyết định khi nào dừng và khởi động xe tự động trong chức năng start/stop.

Thành phần phần mềm khối NVM (NV Block Software Component)

. **Vai trò:** Lưu trữ thông tin quan trọng trong bộ nhớ không bay hơi (NVM) để tái sử dụng sau này.

. **Ví dụ:** Lưu trữ tổng số km xe đã đi vào bộ nhớ NVM khi tắt máy và khôi phục lại khi khởi động lại.

Driver thiết bị phức tạp (Complex Device Driver - CDD)

. **Vai trò:** Truy cập phần cứng nhanh hơn khi có các tính năng yêu cầu thời gian xử lý nghiêm ngặt, chẳng hạn như kích hoạt túi khí hoặc kim phun nhiên liệu.

. **Ghi chú:** RTE không bị bỏ qua và tất cả thông tin phải qua RTE ngay cả khi ứng dụng yêu cầu thời gian nghiêm ngặt.

Thành phần dịch vụ (Service Component)

. **Vai trò:** Cung cấp các dịch vụ từ lớp phần mềm cơ bản (BSW) cho lớp ứng dụng, chẳng hạn như trình quản lý trạng thái ECU hoặc trình quản lý sự kiện chẩn đoán.

Thành phần dịch vụ proxy (Service Proxy Component)

. **Vai trò:** Sử dụng các dịch vụ từ một ECU khác khi cần thiết. Được sử dụng khi thành phần ứng dụng có thể được đặt động trên các ECU khác nhau.

Thành phần phần mềm trừu tượng hóa ECU (ECU Abstraction Software Component)

. **Vai trò:** Truy cập và giao diện giữa lớp MCAL và lớp ứng dụng. Được sử dụng để truy cập các thiết bị ngoại vi bên ngoài vi điều khiển.

Thành phần phần mềm cảm biến/điều khiển (Sensor/Actuator Software Component)

. **Vai trò:** Giao diện giữa các cảm biến/điều khiển bên ngoài và các thành phần ứng dụng khác. Chuyển đổi dữ liệu thô từ cảm biến và kiểm soát trình tự hoạt động của các điều khiển bên ngoài dựa trên các lệnh từ ứng dụng.

Thành phần phần mềm tham số (Parameter Software Component)

. **Vai trò:** Cung cấp các tham số hiệu chỉnh cho dự án. Các tham số này có thể được sử dụng trong tất cả các thành phần phần mềm khác.

. **Ghi chú:** Thành phần này không có file C kèm theo và chỉ là cấu hình, không có chức năng thực hiện hoặc đóng góp chức năng cho dự án.

Thành phần phần mềm kết hợp (Composition Software Component)

. **Vai trò:** Đóng vai trò như các container, chứa tập hợp các thành phần phần mềm. Được sử dụng để nhóm các thành phần dựa trên chức năng.

. **Ghi chú:** Thành phần này không có file C thực hiện riêng biệt nhưng có một tập hợp cấu hình quan trọng của các thành phần nó chứa và giao tiếp giữa chúng.

Tóm tắt

. **Thành phần phần mềm nguyên tử:** Chứa các chức năng cụ thể và phổ biến nhất trong lớp ứng dụng.

. **Thành phần phần mềm tham số:** Cung cấp các tham số hiệu chỉnh và không có chức năng thực hiện.

. **Thành phần phần mềm kết hợp:** Nhóm các thành phần phần mềm lại với nhau dựa trên chức năng và không có file thực hiện riêng.

Chúng ta sẽ đi sâu hơn vào chi tiết về từng thành phần này trong các phần tiếp theo của khóa học, bao gồm cả cấu hình các kết nối giữa chúng.

8. SOFTWARE COMPONENTS (EXAMPLE)

Yêu cầu sử dụng:

1. Lấy tốc độ xe từ cảm biến tốc độ bên ngoài.
2. Nếu tốc độ vượt quá giá trị hiệu chỉnh, thực hiện các hành động sau:
 - o Ngắt kim phun ngay lập tức và dừng xe.
 - o Ghi lại lỗi thông qua module chẩn đoán.
3. Tính toán tốc độ tối đa mà xe đã đạt được và lưu dữ liệu này khi tắt máy.
4. Khôi phục lại dữ liệu khi ECU được bật lên trong chu kỳ lái tiếp theo.

Bước 1: Xác định các thành phần phần mềm

1. **Cảm biến tốc độ bên ngoài:**
 - o **Đầu vào:** Đã có sẵn trên chân IO kỹ thuật số của bộ điều khiển.
2. **Thành phần phần mềm trừu tượng hóa ECU (ECU Abstraction Software Component):**
 - o **Vai trò:** Đọc chân IO tương ứng và tính toán thông tin xung từ cảm biến tốc độ.
3. **Thành phần phần mềm cảm biến/điều khiển (Sensor Actuator Software Component):**
 - o **Vai trò:** Tương tác với lớp trừu tượng hóa ECU và nhận các xung từ cảm biến.
 - o **Tính toán tốc độ:** Dựa trên đầu vào từ cảm biến và tính toán tốc độ theo các đơn vị khác nhau như Km/h, m/h.
4. **Thành phần phần mềm ứng dụng (Application Software Component):**
 - o **Vai trò:** Thực hiện logic cốt lõi để ngắt động cơ khi tốc độ vượt quá giá trị hiệu chỉnh.
 - o **Nhận tốc độ:** Từ thành phần phần mềm cảm biến/điều khiển và thực hiện xử lý tiếp theo.
5. **Thành phần phần mềm tham số hiệu chỉnh (Calibration Parameter Software Component):**
 - o **Vai trò:** Cung cấp tham số hiệu chỉnh để quyết định giá trị tốc độ cắt.
6. **Driver thiết bị phức tạp (Complex Device Driver - CDD):**
 - o **Vai trò:** Ngắt kim phun ngay lập tức khi tốc độ vượt quá giá trị hiệu chỉnh.
 - o **Thiết lập chân IO:** Của bộ điều khiển để ngắt kim phun.
7. **Thành phần dịch vụ phần mềm chẩn đoán (BSW Service Software Component for Diagnostics):**
 - o **Vai trò:** Ghi lỗi thông qua module chẩn đoán trong phần mềm cơ bản.
 - o **Kết nối với thành phần ứng dụng:** Để truyền thông tin lỗi khi có sự kiện xảy ra.
8. **Thành phần phần mềm khối NVM (NV Block Software Component):**
 - o **Vai trò:** Tạo khối NVM cho ứng dụng để lưu trữ tốc độ tối đa.
 - o **Lưu trữ dữ liệu:** Gửi dữ liệu đến NVM manager trong phần mềm cơ bản để lưu vào bộ nhớ NVM.

Quy trình thực hiện

1. **Đầu vào từ cảm biến tốc độ:**

- Cảm biến tốc độ cung cấp đầu vào trên chân IO của bộ điều khiển.
- **ECU Abstraction Software Component** đọc và tính toán xung từ cảm biến.
- 2. **Tính toán tốc độ:**
 - **Sensor Actuator Software Component** tính toán tốc độ dựa trên xung cảm biến.
- 3. **Logic cốt lõi:**
 - **Application Software Component** kiểm tra tốc độ với giá trị hiệu chỉnh từ **Calibration Parameter Software Component**.
 - Nếu tốc độ vượt quá giá trị, **Complex Device Driver** thực hiện ngắt kim phun ngay lập tức.
- 4. **Ghi lỗi:**
 - **Application Software Component** truyền thông tin lỗi đến **BSW Service Software Component for Diagnostics** để ghi lại.
- 5. **Lưu trữ tốc độ tối đa:**
 - **Application Software Component** tính toán tốc độ tối đa và gửi đến **NV Block Software Component** để lưu vào bộ nhớ NVM thông qua **NVM manager**.
- 6. **Khôi phục dữ liệu:**
 - Khi khởi động lại, **NVM manager** khôi phục dữ liệu tốc độ từ bộ nhớ NVM và truyền đến lớp ứng dụng thông qua **NV Block Software Component**.

Kết luận

. **Lựa chọn thành phần phần mềm:** Hiểu cách lựa chọn các thành phần phần mềm phù hợp cho ứng dụng Autosar của bạn.

. **Tiếp tục khóa học:** Sử dụng ví dụ này để tìm hiểu sâu hơn về cơ chế giao tiếp mà Autosar cung cấp.

Chúng ta sẽ tiếp tục đào sâu vào ví dụ này và xem xét các cơ chế giao tiếp chi tiết trong các phần tiếp theo của khóa học.