# TỔNG QUAN VỀ TRUYỀN VÀ NHẬN TÍN HIỆU COM

Trong bài viết này, chúng ta sẽ xem xét các hành động diễn ra trong COM khi truyền và nhận tín hiệu/nhóm tín hiệu. Hãy nhớ rằng giao tiếp nội bộ (giao tiếp giữa các Software Components trong cùng một ECU) được xử lý bởi RTE trong Autosar, vì vậy chúng ta không phải lo lắng về điều đó ở đây. Chúng ta sẽ xem toàn bộ quá trình truyền và nhận tín hiệu giữa các ECU khác nhau.

## COM Transmission

Trong hình bên dưới mô tả việc truyền tải messages / signals. Có ba thực thể chịu trách nhiệm kích hoạt các hành động liên quan đến truyền trong COM: các API Com\_SendSignal và Com\_SendSignalGroup và Gateway dựa trên tín hiệu (thực thể này có thể ảnh hưởng đến cả việc truyền và nhận).

Sau bất kỳ hành động nào trong số này, Các bit cập nhật tín hiệu trong cấu trúc dữ liệu được bật lên để đánh dấu rằng có sự thay đổi hoặc cập nhật trong tín hiệu, việc chuyển đổi endianness (*Endianness liên quan đến thứ tự byte của dữ liệu đa byte. Trong một số hệ thống, dữ liệu cần được chuyển đổi giữa big-endian và little-endian để đảm bảo tính tương thích*) và mở rộng dấu (*Mở rộng dấu là quá trình điều chỉnh giá trị của các số âm và số dương trong hệ thống số để đảm bảo rằng dữ liệu được biểu diễn chính xác khi chuyển đổi kích thước dữ liệu*) được xử lý và dữ liệu tín hiệu sẽ được đặt vào I-PDU để sẵn sàng truyền đi sau quá trình xử lý đó. Ở giai đoạn trung gian, khi các tín hiệu thuộc một nhóm tín hiệu, dữ liệu của chúng có thể được lưu tạm thời trong shadow buffer trước khi toàn bộ nhóm tín hiệu được xử lý và chuyển vào I-PDU. (*shadow buffer là một vùng nhớ tạm thời dùng để lưu trữ dữ liệu tín hiệu trước khi dữ liệu này được chuyển vào I-PDU*)

Sau đó, tùy thuộc vào đặc tính truyền tín hiệu và lựa chọn chế độ truyền I-PDU hoặc lệnh gọi API Com\_TriggerTransmit qua PduR, I-PDU sẽ được gửi ra khỏi COM qua PduR, thông qua API PduR\_ComTransmit (sau lệnh gọi I-PDU được kích hoạt), tuân theo giám sát I-PDU MDT (Thời gian trễ tối thiểu).

Sau đó, phản hồi từ PduR bằng API Com\_TxConfirmation sẽ được chờ đợi dưới sự giám sát chặt chẽ bằng việc theo dõi thời hạn truyền. Cuối cùng, việc giám sát thời hạn truyền hoặc ComTxConfirmation sẽ đưa ra thông báo tương ứng là Com\_CbkTxTOut hoặc Com\_CbkTxAck.

A diagram of a computer

Description automatically generated

**Hình 1: Tổng quan về quá trình truyền COM**

### Luồng Mô Tả Trên

**Bước 1:** Bắt đầu Truyền Tín Hiệu

* **Hành động**: Gọi API Com\_SendSignal hoặc Com\_SendSignalGroup hoặc gateway dựa trên tín hiệu.
* **Xử lý**:
  + Các bit cập nhật tín hiệu được thiết lập.
  + Thực hiện chuyển đổi endianness và mở rộng dấu.
  + Điền dữ liệu tín hiệu vào I-PDU.
  + Sử dụng bộ đệm bóng (shadow buffer) nếu là nhóm tín hiệu.

**Bước 2**: Chuẩn Bị Truyền I-PDU

* **Điều kiện**: Dựa trên thuộc tính chuyển tín hiệu và chế độ truyền I-PDU.
* **Hành động**: Gọi API Com\_TriggerTransmit thông qua PduR.
* **Xử lý**:
  + Kích hoạt I-PDU callout nếu có.
  + Tuân thủ giám sát MDT (Minimum Delay Time).

**Bước 3**: Truyền I-PDU

* **Hành động**: Gọi API PduR\_ComTransmit.
* **Kết quả**: I-PDU được truyền từ COM module ra ngoài thông qua PduR.

**Bước 4**: Chờ Phản Hồi

* **Điều kiện**: Phản hồi từ PduR được chờ đợi dưới sự giám sát chặt chẽ.
* **Hành động**: Theo dõi thời hạn truyền (Transmission Deadline Monitoring).

**Bước 5**: Nhận Phản Hồi và Xác Nhận

* **Trường hợp 1**: Nhận phản hồi từ PduR qua API Com\_TxConfirmation trong thời hạn.
  + **Kết quả**: Quá trình truyền thành công.
  + **Thông báo**: Gửi Com\_CbkTxAck.
* **Trường hợp 2**: Không nhận được phản hồi từ PduR trong thời hạn.
  + **Kết quả**: Quá trình truyền thất bại do quá hạn.
  + **Thông báo**: Gửi Com\_CbkTxTOut.

### Sơ Đồ Luồng Mô Tả

Dưới đây là một sơ đồ mô tả luồng quá trình truyền tín hiệu và nhận phản hồi trong hệ thống AUTOSAR:

A screenshot of a computer program

Description automatically generated

## COM Reception

Quá trình nhận tín hiệu được bắt đầu bằng lệnh gọi API Com\_RxIndication. Đồng thời, quá trình giám sát thời gian nhận đang chạy và có thể được khởi động lại thông qua lệnh gọi này hoặc hoặc kích hoạt hàm callback Com\_CbkRxTOut nếu thông điệp không đến trong thời gian mong đợi.

Khi một I-PDU được nhận, hàm I-PDU callout tùy chọn sẽ được gọi. Sau đó, việc chuyển đổi endianness và xử lý dấu được thực hiện cho từng tín hiệu, giá trị được kiểm tra cả cho các bit cập nhật (nếu nó sử dụng chúng và tín hiệu chưa được cập nhật, nó sẽ bị loại bỏ) và để xác minh rằng nó không phải là một giá trị tín hiệu không hợp lệ đã được xác định trước (trong trường hợp đó các hành động bổ sung sẽ được thực hiện, bao gồm khả năng kích hoạt hàm callout Com\_CbkInv) và chúng được lọc theo cấu hình bộ lọc. Chỉ sau khi qua quá trình lọc, việc giám sát thời hạn cho tín hiệu mới có thể được đặt lại.

Hàm callout Com\_CbkRxAck cũng có thể được thực hiện và dữ liệu cũng có thể đi theo gateway truyền tín hiệu dựa trên tín hiệu, để được truyền lại thông qua PDU Router. Cuối cùng, các giá trị tín hiệu được cập nhật, và các bộ đệm bóng cho các nhóm tín hiệu cũng vậy. Khi các tín hiệu được sử dụng bởi ứng dụng, thông qua các API Com\_ReceiveSignal hoặc Com\_ReceiveSignalGroup, các giá trị tín hiệu sẽ được sử dụng.

A screenshot of a computer

Description automatically generated

***Tổng quan về quy trình tiếp nhận COM***

À chính nó đấy. Đây là quá trình gửi và nhận tin nhắn từ module COM. Nếu bài viết này thu hút sự quan tâm của bạn về cách xác định các tín hiệu và nhóm tín hiệu, thì nơi tốt để tiếp theo là bài đăng của chúng tôi về Khám phá các định nghĩa ISignal và ISignalGroup trong Autosar ARXML(mở trong một tab mới). Ở đó, bạn sẽ tìm hiểu cách xác định I-Signals và I-SignalGroups, điều cần thiết cho hành trình Autosar cổ điển của bạn.

Cuối cùng, đừng quên tham gia danh sách chờ nhận sách điện tử sắp ra mắt của chúng tôi!

Tác giả: Micael Coutinho(mở trong tab mới)

Người giới thiệu:

Đặc điểm kỹ thuật truyền thông - Autosar Đặc điểm kỹ thuật

# GIẢI THÍCH

## COM transmisstion

### Com\_SendSignal và Com\_SendSignalGroup

#### ****Com\_SendSignal****:

**Mô tả**: Đây là API được sử dụng để gửi một tín hiệu đơn lẻ từ một Software Component (SWC) đến COM module.

**Quy trình**: Khi API này được gọi, nó sẽ kích hoạt quá trình truyền dữ liệu của tín hiệu từ SWC qua COM module. Quá trình này bao gồm việc thiết lập cập nhật các bit tín hiệu, chuyển đổi endianness (nếu cần), mở rộng dấu, và cuối cùng là điền dữ liệu vào I-PDU.

**Vai trò**: Đảm bảo rằng tín hiệu đơn lẻ được truyền đi một cách chính xác và nhất quán từ SWC đến các hệ thống khác qua COM module.

#### Com\_SendSignalGroup:

**Mô tả**: Đây là API được sử dụng để gửi một nhóm tín hiệu từ SWC đến COM module.

**Quy trình**: Tương tự như Com\_SendSignal, nhưng thay vì gửi một tín hiệu đơn lẻ, API này xử lý một nhóm tín hiệu. Dữ liệu của nhóm tín hiệu được điền vào một bộ đệm bóng (shadow buffer) trước khi được xử lý và truyền đi trong I-PDU.

**Vai trò**: Đảm bảo rằng một nhóm tín hiệu được truyền đi đồng bộ và hiệu quả từ SWC qua COM module đến các hệ thống khác.

### Gateway Dựa trên Tín hiệu

**Mô tả**: Gateway dựa trên tín hiệu là một thực thể trong hệ thống AUTOSAR chịu trách nhiệm định tuyến tín hiệu giữa các ECU khác nhau.

**Chức năng**:

* **Truyền tín hiệu**: Khi một tín hiệu từ một ECU cần được gửi đến một ECU khác, gateway sẽ tiếp nhận tín hiệu này và xác định đường đi (route) để truyền tín hiệu đó đến ECU đích.

**Example**: ECU A muốn gửi tín hiệu đến ECU B. Gateway sẽ nhận tín hiệu từ ECU A, sau đó truyền tín hiệu này đến ECU B theo đường đi đã định sẵn.

* **Nhận tín hiệu**: Gateway sẽ tiếp nhận tín hiệu và thực hiện việc định tuyến tới ECU đích.

**Example**: ECU B gửi tín hiệu đến ECU C. Gateway sẽ nhận tín hiệu từ ECU B, sau đó truyền tín hiệu đến ECU C.

**Ảnh hưởng**: Gateway dựa trên tín hiệu có thể ảnh hưởng đến cả quá trình truyền và nhận tín hiệu, đảm bảo rằng các tín hiệu được định tuyến chính xác và tới đích mong muốn.

### Phân Tích Câu

**Câu gốc**: Sau đó, tùy thuộc vào các thuộc tính chuyển tín hiệu và chế độ truyền I-PDU hoặc cuộc gọi API Com\_TriggerTransmit thông qua PduR, I-PDU sẽ được gửi ra khỏi COM thông qua PduR, thông qua API PduR\_ComTransmit (sau khi cuộc gọi I-PDU callout được kích hoạt), tuân thủ việc giám sát MDT (Minimum Delay Time) của I-PDU.

1. **"Sau đó, tùy thuộc vào các thuộc tính chuyển tín hiệu và chế độ truyền I-PDU hoặc cuộc gọi API Com\_TriggerTransmit thông qua PduR"**:
   * **Thuộc tính chuyển tín hiệu**: Đây là các thiết lập hoặc cài đặt liên quan đến cách thức và điều kiện mà tín hiệu sẽ được truyền. Ví dụ như tốc độ truyền, ưu tiên tín hiệu, hoặc thời gian chờ.
   * **Chế độ truyền I-PDU**: I-PDU có thể được cấu hình với các chế độ truyền khác nhau (như periodic, event-triggered, mixed mode). Chế độ truyền này quyết định khi nào và làm thế nào I-PDU sẽ được gửi.
   * **API Com\_TriggerTransmit**: Đây là một API được gọi khi cần kích hoạt việc truyền I-PDU từ COM module thông qua PduR (Protocol Data Unit Router). API này được gọi để yêu cầu PduR chuẩn bị và gửi I-PDU.
2. **"I-PDU sẽ được gửi ra khỏi COM thông qua PduR, thông qua API PduR\_ComTransmit"**:
   * **COM**: Là mô-đun giao tiếp trong hệ thống AUTOSAR, chịu trách nhiệm quản lý và xử lý các tín hiệu trước khi truyền đi.
   * **PduR (Protocol Data Unit Router)**: Là mô-đun trong hệ thống AUTOSAR chịu trách nhiệm định tuyến các đơn vị dữ liệu giao thức (PDU) giữa các mô-đun khác nhau.
   * **API PduR\_ComTransmit**: Là API trong PduR được sử dụng để truyền I-PDU từ COM module ra bên ngoài. Khi gọi API này, PduR sẽ gửi I-PDU đến đích mong muốn.
3. **"(sau khi cuộc gọi I-PDU callout được kích hoạt)"**:
   * **I-PDU callout**: Đây là một cơ chế trong AUTOSAR cho phép thực hiện các hành động hoặc kiểm tra bổ sung trước khi I-PDU thực sự được truyền đi. Khi cuộc gọi này được kích hoạt, nó sẽ đảm bảo rằng mọi điều kiện tiên quyết hoặc kiểm tra bổ sung được hoàn tất.
4. **"tuân thủ việc giám sát MDT (Minimum Delay Time) của I-PDU"**:
   * **MDT (Minimum Delay Time)**: Là khoảng thời gian tối thiểu giữa hai lần truyền liên tiếp của cùng một I-PDU. Giám sát MDT đảm bảo rằng I-PDU không được truyền quá thường xuyên, tránh tình trạng quá tải hệ thống và đảm bảo hiệu suất truyền dữ liệu.
   * **Tuân thủ MDT**: Nghĩa là việc truyền I-PDU sẽ tuân thủ quy định về thời gian chờ tối thiểu này, đảm bảo rằng các I-PDU được truyền theo đúng chu kỳ được xác định trước.

### Phân Tích Câu

**Câu gốc**: Sau đó, phản hồi từ PduR bằng API Com\_TxConfirmation sẽ được chờ đợi dưới sự giám sát chặt chẽ bằng việc theo dõi thời hạn truyền. Cuối cùng, việc giám sát thời hạn truyền hoặc ComTxConfirmation sẽ đưa ra thông báo tương ứng là Com\_CbkTxTOut hoặc Com\_CbkTxAck.

1. **"Sau đó, phản hồi từ PduR bằng API Com\_TxConfirmation sẽ được chờ đợi"**:
   * **Phản hồi từ PduR**: Sau khi I-PDU được gửi đi, hệ thống cần chờ đợi phản hồi từ PduR để xác nhận rằng I-PDU đã được truyền thành công.
   * **API Com\_TxConfirmation**: Đây là API trong AUTOSAR được gọi bởi PduR để báo cáo rằng quá trình truyền I-PDU đã hoàn thành. Nó xác nhận rằng dữ liệu đã được gửi đến đích mong muốn.
2. **"Dưới sự giám sát chặt chẽ bằng việc theo dõi thời hạn truyền"**:
   * **Giám sát thời hạn truyền**: Đây là quá trình theo dõi và đảm bảo rằng việc truyền I-PDU diễn ra trong khoảng thời gian cho phép. Nếu thời hạn truyền vượt quá giới hạn, hệ thống sẽ xác định rằng việc truyền đã thất bại.
   * **Ý nghĩa**: Việc chờ đợi phản hồi từ PduR không chỉ đơn thuần là đợi một cách thụ động, mà hệ thống còn giám sát chặt chẽ thời gian để đảm bảo rằng quá trình truyền không gặp sự cố hoặc trì hoãn không mong muốn.
3. **"Cuối cùng, việc giám sát thời hạn truyền hoặc ComTxConfirmation sẽ đưa ra thông báo tương ứng"**:
   * **Việc giám sát thời hạn truyền**: Nếu quá trình truyền I-PDU vượt quá thời gian cho phép mà không nhận được phản hồi từ PduR, hệ thống sẽ kết luận rằng việc truyền đã thất bại do quá hạn.
   * **ComTxConfirmation**: Nếu nhận được phản hồi từ PduR trong thời gian cho phép, hệ thống sẽ xác nhận rằng quá trình truyền đã thành công.
4. **"Com\_CbkTxTOut hoặc Com\_CbkTxAck"**:
   * **Com\_CbkTxTOut**: Là thông báo được phát ra khi quá trình truyền vượt quá thời hạn cho phép mà không nhận được phản hồi. "TxTOut" là viết tắt của "Transmission Timeout".
   * **Com\_CbkTxAck**: Là thông báo được phát ra khi quá trình truyền hoàn tất thành công. "TxAck" là viết tắt của "Transmission Acknowledgment".

## COM Reception

### Phân Tích Câu

**Câu gốc**: Việc nhận tín hiệu được bắt đầu bằng cuộc gọi API Com\_RxIndication. Đồng thời, quá trình giám sát thời hạn nhận đang chạy và có thể được khởi động lại thông qua cuộc gọi này hoặc kích hoạt hàm callback Com\_CbkRxTOut nếu thông điệp không đến trong thời gian mong đợi.

Giải Thích Chi Tiết

1. **"Việc nhận tín hiệu được bắt đầu bằng cuộc gọi API Com\_RxIndication"**:
   * **Giải thích**: Khi hệ thống nhận được tín hiệu, API Com\_RxIndication được gọi để khởi động quá trình nhận dữ liệu.
   * **Com\_RxIndication API**: Đây là một giao diện lập trình ứng dụng (API) trong hệ thống AUTOSAR, được sử dụng để thông báo rằng một tín hiệu hoặc một gói dữ liệu đã được nhận.
2. **"Đồng thời, quá trình giám sát thời hạn nhận đang chạy"**:
   * **Giải thích**: Khi tín hiệu bắt đầu được nhận, hệ thống cũng khởi động quá trình giám sát thời gian để đảm bảo tín hiệu được nhận trong khoảng thời gian mong đợi.
   * **Giám sát thời hạn nhận**: Đây là một quá trình tự động theo dõi và kiểm tra xem tín hiệu có được nhận trong thời gian quy định hay không. Nếu tín hiệu không đến trong khoảng thời gian này, hệ thống sẽ xác định có sự cố.
3. **"Và có thể được khởi động lại thông qua cuộc gọi này"**:
   * **Giải thích**: Mỗi lần API Com\_RxIndication được gọi, quá trình giám sát thời hạn nhận có thể được làm mới hoặc khởi động lại. Điều này giúp đảm bảo rằng thời gian giám sát luôn cập nhật theo các lần nhận tín hiệu mới.
4. **"Hoặc kích hoạt hàm callback Com\_CbkRxTOut nếu thông điệp không đến trong thời gian mong đợi"**:
   * **Giải thích**:
     + **Callback Com\_CbkRxTOut**: Đây là một hàm callback trong hệ thống AUTOSAR. Hàm này được gọi khi quá trình giám sát thời hạn nhận phát hiện rằng tín hiệu không đến trong thời gian quy định.
     + **Thông điệp không đến trong thời gian mong đợi**: Nếu tín hiệu không được nhận trong khoảng thời gian đã đặt trước, hệ thống sẽ xác định có lỗi và kích hoạt hàm callback để thông báo về sự cố này.

### Phân tích Câu

**Câu gốc**: Khi một I-PDU được nhận, hàm I-PDU callout tùy chọn sẽ được gọi. Sau đó, việc chuyển đổi endianness và xử lý dấu được thực hiện cho từng tín hiệu, giá trị được kiểm tra cả cho các bit cập nhật (nếu nó sử dụng chúng và tín hiệu chưa được cập nhật, nó sẽ bị loại bỏ) và để xác minh rằng nó không phải là một giá trị tín hiệu không hợp lệ đã được xác định trước (trong trường hợp đó các hành động bổ sung sẽ được thực hiện, bao gồm khả năng kích hoạt hàm callout Com\_CbkInv) và chúng được lọc theo cấu hình bộ lọc. Chỉ sau khi qua quá trình lọc, việc giám sát thời hạn cho tín hiệu mới có thể được đặt lại.

1. **"Khi một I-PDU được nhận"**:
   * **Giải thích**: Khi một gói dữ liệu tương tác (I-PDU) được nhận từ mạng truyền thông của hệ thống AUTOSAR.
2. **"Hàm I-PDU callout tùy chọn sẽ được gọi"**:
   * **Giải thích**: Hàm callout là một hàm tùy chọn có thể được cấu hình để thực hiện các hành động bổ sung khi I-PDU được nhận. Hàm này được gọi để xử lý dữ liệu hoặc thực hiện các kiểm tra bổ sung.
3. **"Việc chuyển đổi endianness và xử lý dấu được thực hiện cho từng tín hiệu"**:
   * **Endianness**: Là thứ tự byte của dữ liệu đa byte. Việc chuyển đổi endianness đảm bảo dữ liệu được định dạng chính xác cho hệ thống đích.
   * **Xử lý dấu**: Điều chỉnh giá trị của các số âm và số dương để đảm bảo dữ liệu được biểu diễn chính xác khi chuyển đổi kích thước dữ liệu.
4. **"Giá trị được kiểm tra cả cho các bit cập nhật"**:
   * **Bit cập nhật**: Là các bit trong cấu trúc dữ liệu chỉ ra rằng một tín hiệu đã được cập nhật. Kiểm tra các bit này để xác định xem tín hiệu có mới hay không.
   * **Giải thích**: Nếu tín hiệu sử dụng các bit cập nhật và các bit này chưa được thiết lập (tín hiệu chưa được cập nhật), tín hiệu sẽ bị loại bỏ và không được xử lý thêm.
5. **"Để xác minh rằng nó không phải là một giá trị tín hiệu không hợp lệ đã được xác định trước"**:
   * **Giải thích**: Kiểm tra giá trị tín hiệu để đảm bảo rằng nó không phải là một giá trị không hợp lệ đã được xác định trước. Các giá trị không hợp lệ có thể là các giá trị đặc biệt được sử dụng để biểu thị lỗi hoặc điều kiện ngoại lệ.
   * **Hành động bổ sung**: Nếu giá trị tín hiệu không hợp lệ, các hành động bổ sung sẽ được thực hiện, bao gồm khả năng kích hoạt hàm callout Com\_CbkInv để xử lý các giá trị không hợp lệ.
6. **"Và chúng được lọc theo cấu hình bộ lọc"**:
   * **Giải thích**: Các tín hiệu sau khi kiểm tra sẽ được lọc dựa trên cấu hình bộ lọc được định nghĩa trước. Bộ lọc giúp loại bỏ các tín hiệu không cần thiết hoặc không hợp lệ trước khi chúng được xử lý thêm.
7. **"Chỉ sau khi qua quá trình lọc, việc giám sát thời hạn cho tín hiệu mới có thể được đặt lại"**:
   * **Giám sát thời hạn**: Là quá trình theo dõi để đảm bảo tín hiệu được nhận trong thời gian cho phép.
   * **Đặt lại giám sát thời hạn**: Sau khi tín hiệu đã qua quá trình lọc và xác nhận hợp lệ, quá trình giám sát thời hạn sẽ được đặt lại để tiếp tục theo dõi các tín hiệu tiếp theo.

### Phân Tích Câu

**Câu gốc**: Hàm callout Com\_CbkRxAck cũng có thể được thực hiện và dữ liệu cũng có thể đi theo gateway truyền tín hiệu dựa trên tín hiệu, để được truyền lại thông qua PDU Router. Cuối cùng, các giá trị tín hiệu được cập nhật, và các bộ đệm bóng cho các nhóm tín hiệu cũng vậy. Khi các tín hiệu được sử dụng bởi ứng dụng, thông qua các API Com\_ReceiveSignal hoặc Com\_ReceiveSignalGroup, các giá trị tín hiệu sẽ được sử dụng.

1. **"Hàm callout Com\_CbkRxAck cũng có thể được thực hiện"**:
   * **Hàm callout Com\_CbkRxAck**: Đây là một hàm callback trong hệ thống AUTOSAR được gọi để xác nhận rằng dữ liệu đã được nhận thành công.
   * **Giải thích**: Khi tín hiệu được nhận, hàm callout này có thể được thực hiện để xác nhận quá trình nhận dữ liệu đã thành công và tiếp tục xử lý các bước tiếp theo.
2. **"Dữ liệu cũng có thể đi theo gateway truyền tín hiệu dựa trên tín hiệu"**:
   * **Gateway truyền tín hiệu dựa trên tín hiệu**: Là một thành phần trong hệ thống AUTOSAR, chịu trách nhiệm định tuyến và truyền lại tín hiệu giữa các ECU khác nhau.
   * **Giải thích**: Dữ liệu tín hiệu có thể được định tuyến thông qua gateway này để tiếp tục truyền đến các thành phần khác trong hệ thống.
3. **"Để được truyền lại thông qua PDU Router"**:
   * **PDU Router**: Là mô-đun trong hệ thống AUTOSAR chịu trách nhiệm định tuyến các đơn vị dữ liệu giao thức (PDU) giữa các mô-đun khác nhau.
   * **Giải thích**: Dữ liệu tín hiệu sau khi đi qua gateway truyền tín hiệu sẽ được truyền lại thông qua PDU Router để đến đích mong muốn.
4. **"Cuối cùng, các giá trị tín hiệu được cập nhật, và các bộ đệm bóng cho các nhóm tín hiệu cũng vậy"**:
   * **Cập nhật giá trị tín hiệu**: Các giá trị tín hiệu được nhận sẽ được cập nhật để phản ánh thông tin mới nhất.
   * **Bộ đệm bóng (shadow buffers)**: Là vùng nhớ tạm thời lưu trữ dữ liệu tín hiệu cho các nhóm tín hiệu.
   * **Giải thích**: Sau khi tín hiệu được nhận và xác nhận, các giá trị tín hiệu sẽ được cập nhật và lưu trữ vào các bộ đệm bóng nếu chúng là một phần của nhóm tín hiệu.
5. **"Khi các tín hiệu được sử dụng bởi ứng dụng, thông qua các API Com\_ReceiveSignal hoặc Com\_ReceiveSignalGroup, các giá trị tín hiệu sẽ được sử dụng"**:
   * **API Com\_ReceiveSignal và Com\_ReceiveSignalGroup**: Đây là các API trong hệ thống AUTOSAR được sử dụng bởi các ứng dụng để nhận dữ liệu tín hiệu.
   * **Giải thích**: Khi các ứng dụng cần sử dụng các giá trị tín hiệu, chúng sẽ gọi các API này để truy cập và sử dụng dữ liệu tín hiệu đã được nhận và cập nhật.

# Exploring ISignal and ISignalGroup Definitions in Autosar ARXML

Hiểu rõ những phức tạp của ISignal (I-SIGNAL) và ISignalGroup (I-SIGNAL-GROUP) trong Autosar là rất quan trọng để trao đổi dữ liệu hiệu quả trong các hệ thống ô tô. Những container này đóng vai trò then chốt trong việc giao tiếp, và khi được định nghĩa đúng cách trong ARXML, chúng có thể tiết kiệm rất nhiều công sức kỹ thuật khi cấu hình mô-đun COM. Trong hướng dẫn toàn diện này, chúng ta sẽ tìm hiểu các yếu tố, cấu hình, và các chuyển đổi định hình nên nền tảng của các yếu tố quan trọng trong Autosar này.

Trong bài viết này, chúng ta sẽ khám phá sâu vào các yếu tố này. Nếu chúng còn mới mẻ đối với bạn, hãy dừng lại một trong các bài viết giới thiệu về giao tiếp của chúng tôi, như Tổng quan về Communication Stack.

## ISignal

Container ISignal đại diện cho một SystemSignal tại lớp tương tác trong Autosar. Nó cho phép cùng một tín hiệu hệ thống được gửi đến nhiều bộ nhận, một khái niệm gọi là signal fan-out. Hầu hết các yếu tố trong ISignal có thể được ánh xạ trực tiếp đến COM, đơn giản hóa cấu hình thủ công thông qua một trình cấu hình tự động. Container ISignal bao gồm các yếu tố sau:

* **ISignalType**: Mô tả nếu tín hiệu là dạng mảng (UINT8\_N hoặc UINT8\_DYN tại COM) hoặc dạng nguyên thủy.
* **Length**: Kích thước của tín hiệu tính bằng bit. Cần xem xét nếu các bộ chuyển đổi thêm dữ liệu bổ sung.
* **InitValue**: Giá trị khởi tạo tùy chọn được đưa ra bởi thuộc tính ValueSpecification.
* **ISignalProps**: Container tùy chọn cung cấp thuộc tính handleOutOfRange, định nghĩa hành vi cho các giá trị ngoài phạm vi.
* **TimeoutSubstitutionValue**: Container ValueSpecification tùy chọn tạo thuộc tính COM ComTimeoutSubstitution.
* **SystemSignal**: Đại diện cho dữ liệu trao đổi giữa các ECU khác nhau.
* **NetworkRepresentationProps**: Container xác định biểu diễn mạng của ISignal.
* **DataTypePolicy**: Liệt kê định nghĩa tập thuộc tính dùng để mô tả SwDataDefProps của một tín hiệu.
* **DataTransformation**: Tham chiếu tùy chọn đến container DataTransformation để áp dụng chuyển đổi dữ liệu cho ISignal.
* **TransformationISignalProps**: Bao gồm các thuộc tính chuyển đổi cụ thể cho tín hiệu này.

Các yếu tố được chứa trong ARXML của một I-SIGNAL

Container ValueSpecification có thể ở dạng một trong các lớp con sau cho một ISignal:

* **NumericalValueSpecification**: Gán cho một phần tử dữ liệu nguyên thủy chứa giá trị số.
* **TextValueSpecification**: Định nghĩa nhãn cho một liệt kê, với các giá trị là chuỗi.
* **ArrayValueSpecification**: Định nghĩa các giá trị cho một mảng, chứa sự kết hợp của các phần tử và một thuộc tính tùy chọn cho các mảng động.

A screenshot of a computer program

Description automatically generated

## ISignalGroup

Container ISignalGroup, được tạo ra vì các lý do tương tự như ISignal, bao gồm các yếu tố sau:

* **SystemSignalGroup**: Tham chiếu đến SystemSignalGroup được truyền bởi ISignalGroup này.
* **ISignal**: Tham chiếu đến từng ISignal được nhóm bởi nhóm tín hiệu.
* **ComBasedSignalGroupTransformation**: Tham chiếu tùy chọn đến container DataTransformation mô tả chuỗi chuyển đổi cho Transformer dựa trên COM (ComXf).
* **TransformationISignalProps**: Một chuỗi chuyển đổi cụ thể cho ISignalGroup này.

Sự kết hợp của các yếu tố ARXML tạo thành một I-SIGNAL-GROUP

Several types of components

Description automatically generated with medium confidence