Some/IP

Introduction

- This protocol specification specifies the format, message sequences and semantics of the AUTOSAR Protocol "Scalable service-Oriented MiddlewarE over IP (SOME/IP)"
- SOME/IP is an automotive/embedded communication protocol which supports remote procedure calls, event notifications and underlying serialization/wire format.

Protocol purpose and Objective

- Yet another RPC-Mechanism (remote procedure call [inter service])
    - Fulfills the hard requirements regarding resource consumption in an embedded world.
    - Compatible with AUTOSAR at least on the wire-format level;
      i.e communicate with PDUs. AUTOSAR can send and receive without modification to the AUTOSAR Standards.
    - Is scalable from tiny to large Platform.

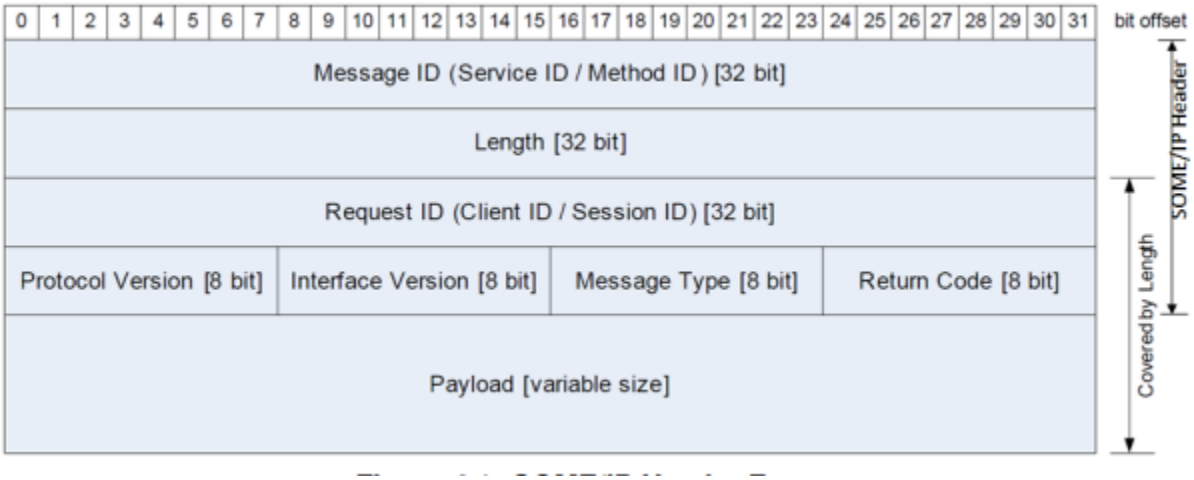
- SOME/IP shall be implemented on different operating system (i.e. AUTOSAR, GENIVI, and OSEK) and even embedded devices without an operating system.
- SOME/IP shall be used for inter-ECU Client/Server Serialization.
- An implementation of SOME/IP allows AUTOSAR to parse the RPC PDUs and transport the signals to the application.

Protocol Specification

- SOME/IP provides service oriented communication over the network.
- It is based on the service definitions that lists the functionality that the service provides.
- A service can contain zero or multiple **events, methods or fields.**
- **Events:** Events provide data that are sent cyclically or on change from the provider to the subscriber.
- **Methods:** Methods provide the possibility to the subscriber to issue remote procedure calls which are executed on the provider side.
- **Fields:** Fields are combinations of one or more of the following three
    - **A notifier** which sends data on change from the provider to the subscribers.
    - **A getter** which can be called by the subscriber to explicitly query the provider for the value.
    - **A setter** which can be called by the subscriber when it wants to change the value on the provider side.

Specification of SOME/IP Message Format (Serialization)

- Serialization describes the way data is represented in protocol data units (PDUs) as payload of either UDP or TCP messages, transported over an IP-based automotive in-vehicle network.



- **Message ID:** Identifier that is used to identify
    - The RPC calls a method of an application.
    - Or to identify an event.

- **Request ID:** The Request ID allows a server and client to differentiate multiple parallel uses of the same method, getter or setter.
- **Message Type:** The message type field is used to differentiate different types of messages and shall contain the following values as shown in the table:

Number	Value	Description
0x00	REQUEST	A request expecting a response (even void)
0x01	REQUEST_NO_RETURN	A fire&forget request
0x02	NOTIFICATION	A request of a notification/event callback expecting no response
0x80	RESPONSE	The response message
0x81	ERROR	The response containing an error
0x20	TP_REQUEST	A TP request expecting a response (even void)
0x21	TP_REQUEST_NO_RETURN	A TP fire&forget request
0x22	TP_NOTIFICATION	A TP request of a notification/event call-back expecting no response
0xa0	TP_RESPONSE	The TP response message
0xa1	TP_ERROR	The TP response containing an error

- **Return Code:** The return code shall be used to signal whether a request was successfully processed.

Message Type	Allowed Return Codes
REQUEST	N/A set to 0x00 (E_OK)
REQUEST_NO_RETURN	N/A set to 0x00 (E_OK)
NOTIFICATION	N/A set to 0x00 (E_OK)
RESPONSE	See Return Codes in [PRS_SOMEIP_00191]
ERROR	See Return Codes in [PRS_SOMEIP_00191]. Shall not be 0x00 (E_OK).

- **Payload:**
  - The size of the SOME/IP payload field depends on the transport protocol used. With UDP the SOME/IP payload shall be between 0 and 1400 Bytes.
  - The limitation to 1400 Bytes is needed in order to allow for future changes to protocol stack (e.g. changing to IPv6 or adding security means).
  - Since TCP supports segmentation of payloads, larger sizes are automatically supported. Payload might consist of data elements for events or parameters for methods.

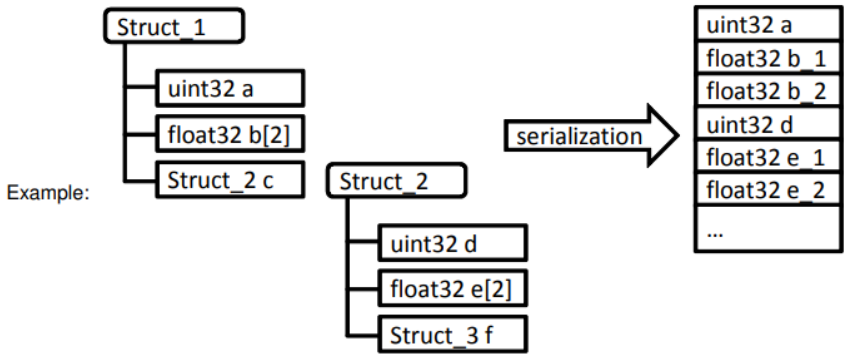
Serialization Of Data Structures

- The serialization is based on the parameter list defined by the interface specification. The interface specification defines the exact position of all data structures in the PDU and has to consider the memory alignment.

- **Basic DataTypes**

Type	Description	Size [bit]	Remark
boolean	TRUE/FALSE value	8	FALSE (0), TRUE (1)
uint8	unsigned Integer	8	
uint16	unsigned Integer	16	
uint32	unsigned Integer	32	
uint64	unsigned Integer	64	
sint8	signed Integer	8	
sint16	signed Integer	16	
sint32	signed Integer	32	
sint64	signed Integer	64	
float32	floating point number	32	IEEE 754 binary32 (Single Precision)
float64	floating point number	64	IEEE 754 binary64 (Double Precision)

  - **Structured DataTypes:** The serialization of a struct shall be close to the in-memory layout. This means, only the parameters shall be serialized sequentially into the buffer.



Strings(Fixed/Dynamic), Arrays (fixed/Dynamic)

Specification Of SOME/IP Protocol

- SOME/IP supports UDP and TCP binding.
- One Service instance can use the following setup for its communication of all the methods, events and notifications.
  - Up to one TCP Connection.
  - Up to one UDP unicast connection
  - Up to one UDP multicast connection.

- Supports
  - **Request/Response Communication (**Client sends request server responds**),**
  - **Fire and Forget Communication (**Requests with no return method**),**

- **Notification Events** (server publishes and client subscribes)
  - Cyclic Update
  - Update on change
  - Epsilon Change
- **Fields (**Represents a status and has a valid value**)**