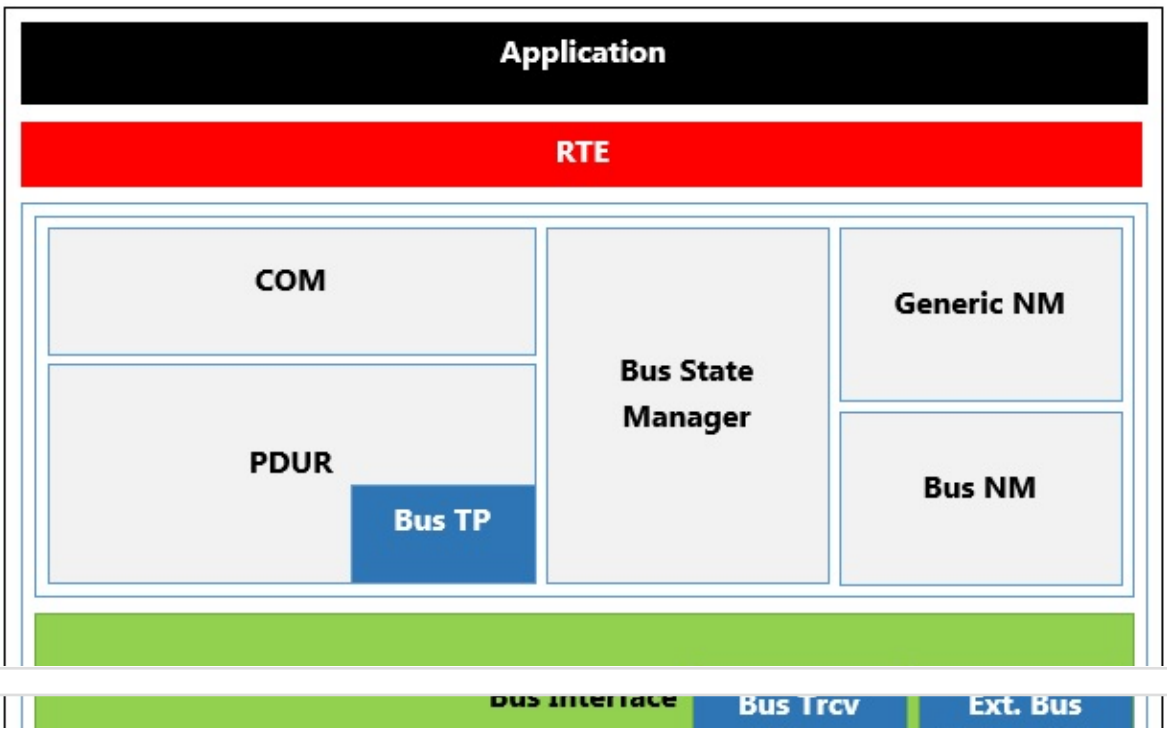
## Automotive & Embedded Info

Never Forget Basics Whether its Life or Anything Else ... Basics are Cores. While seeing a Tree how can we forget Seed...

# Communication Stack

### A generic Communication Stack in AUTOSAR layered architecture is a set of following software modules:

Communication Stack or COM stack facilitates communication. Hence COM stack can be defined as a software stack that provides communication services to the Basic Software Modules and Application Layer or Application Software.




**AUTOSAR COM:**

Part of the services layer. AUTOSAR COM is a module between the RTE and the PDU Router. It is responsible for providing Signal level access to the application layer and PDU level access to the lower layers independent of the protocol.

It packs the signals to a PDU at the transmitter and unpacks the received PDU to provide signal level access to the application at the receiver. At the PDU level, COM is responsible for grouping of the PDUs, starting and stopping of the PDU groups.

**PDUR:**

It stands for Protocol Data Unit Router. Part of the services layer. PDU Router is a module responsible for routing the PDU to the respective Bus Specific Interface modules.

Above the PduR module all the PDUs are protocol independent, and below PduR all the PDUs are routed to the protocol specific modules. PduR is also responsible for PDU level gatewaying i.e. transmitting the received PDU from one Bus Specific Interface module to other Bus Specific Interface module.

Gatewaying can also be done when a PDU is to be routed from one controller to another over the same protocol.

**BUS SPECIFIC TP MODULE:**

It stands for transport protocol. The basic services offered by the Bus TP module are segmentation of messages which have a payload of more than 8 bytes, transmission of the messages with flow control and reassembling the segmented messages at the receiver.

**BUS SPECIFIC INTERFACE MODULE:**

Part of the ECU abstraction layer (Ex: CanIf, LinIf, FrIf, EthIf, FlexRayIf etc). These modules provide interface between hardware abstraction layer and service layer.

Interface is a module in the ECU Abstraction Layer which is responsible for services like Transmit Request, Transmit Confirmation, Reception Indication, Controller mode control and PDU mode control.

**BUS SPECIFIC SM MODULE:**

This module shall implement the control flow for the respective bus. The CAN State Manager (CanSM) is a member of the Communication Service Layer. It interacts with the Communication Hardware Abstraction Layer and the System Service Layer.

**BUS SPECIFIC NM MODULE:**

The AUTOSAR CAN Network Management is a hardware independent protocol tools that can only be used on CAN network.

Its main purpose is to coordinate the transition between normal operation and bus-sleep mode of the network. The Bus Network

Management function provides an adaptation between network Management Interface (NmIf) and Bus Interface module.

**EXTERNAL BUS DRIVER:**

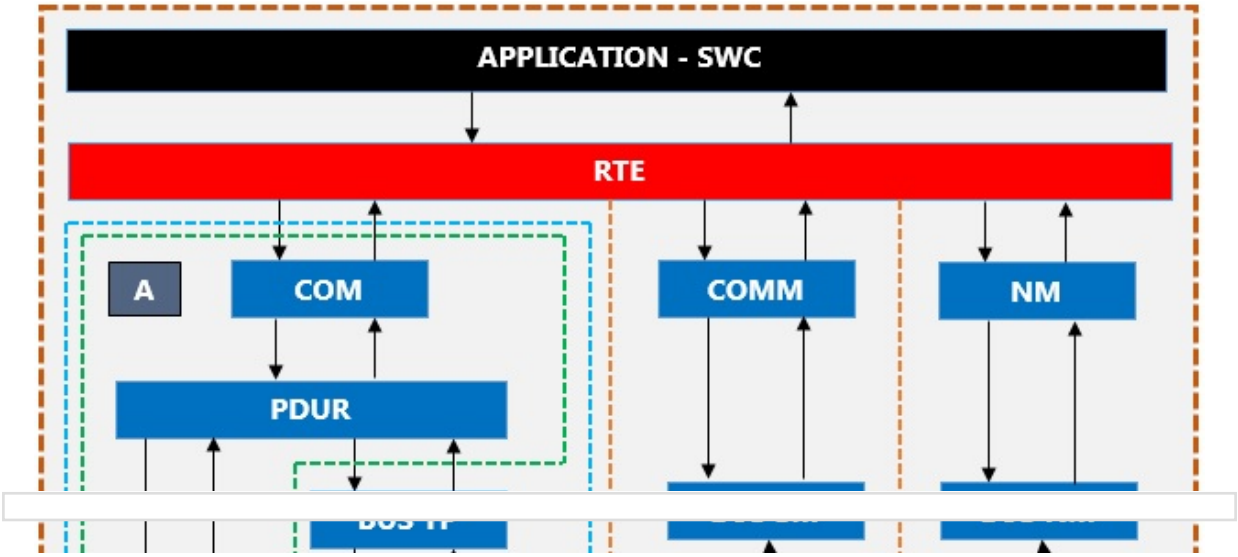
Part of the ECU abstraction layer (Ex: External drivers like CanTrcv, FrTrcv, EthTrcv, etc). This module provides bus specific transceiver access to the upper layer services a hardware-independent interface to the upper layers.

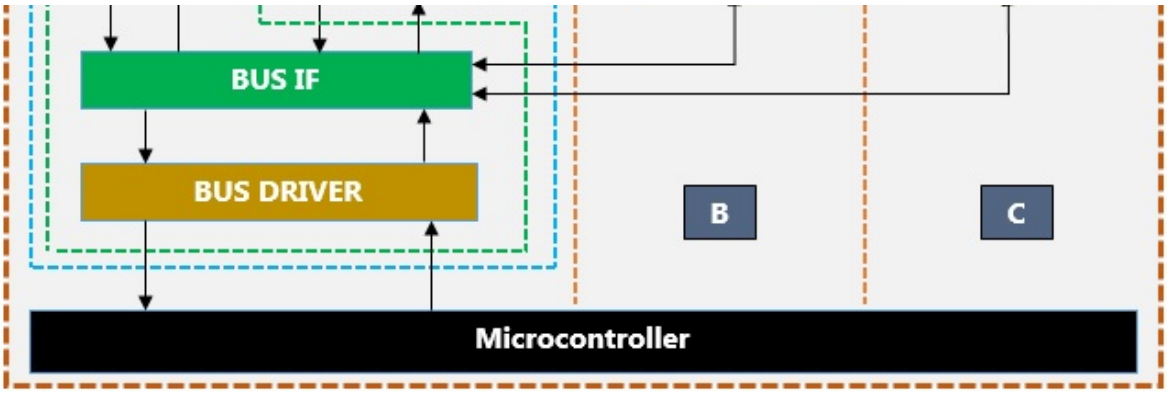
**INTERNAL BUS DRIVER:**

Part of the AUTOSAR MCAL layer (Ex: CanDrv, LinDrv, FrDrv). This module provides hardware access to the upper layer services and a hardware-independent interface to the upper layers. Bus If is the only module that can access the CAN driver.

## I would like to explain tell flow of communication stack with the below diagram:

Above diagram divided into three sections A, B and C.






In AUTOSAR complaints projects applications cannot directly send data directly unless or until it is verified that the hardware is in idle condition to send the data. To verify these idle conditions there are certain parameters such as controller and transceiver are in active state etc.

These parameters are verified by application software component before sending the data. For example please refer below mentioned high level flow of section B:

### EXECUTION FLOW OF FULL_COM REQUEST BY APPLICATION:

Application -> RTE -> COMM -> BUS SM -> BUS IF -> BUS Driver -> Physical layer.

### CALL-BACK MECHANISM TO READ BACK FULL_COM REQUEST :

Physical layer -> BUS Driver -> BUS IF -> BUS SM – > COMM – > RTE -> Application.

### EXECUTION FLOW OF DATA SEND REQUEST BY APPLICATION:

Application -> RTE -> COM -> PDUR- > BUS IF -> BUS Driver -> Physical layer.

### CALL-BACK MECHANISM TO DATA SEND REQUEST :

Physical layer -> BUS Driver -> BUS IF -> BUS SM – > COMM – > RTE -> Application.

### SECTION A: DATA TRANSMISSION AND RECEPTION MANAGEMENT

It is used to send data/signal/Pdu. It is showing the flow of signal from application layer to the physical bus. There are two interface by which application layer can send signal.

1. IF interface is used when length of data is less than or equal to what BUSIF can send.
2. TP interface is used when length of data is more than to what BUSIF can send.

### EX: CAN BUS:

If it is basic CAN it can send only 8 bytes at a time and if it CAN FD then it can send 64 bytes at a time. In this example we can consider it basic CAN so if data/signal is equal to or less than 8 bytes then it will be map to IF interface and if data/signal is more than 8 byte it will be map to TP gateway.

In section A of above figure blue dotted line shows TP interface flow and green dotted line shows IF interface flow of data/signal/Pdu.

### SECTION B: STATE MANAGEMENT

It is used to execute certain condition which is requested by application layer before sending the data. After execution of the request application will be informed via call-back mechanism. If feedback is positive only then application can send data on the bus.

Once use of bus is not require the application can put certain condition like controller and transceiver state inactive for no communication for power saving.

### SECTION C: NETWORK MANAGEMENT

Any communication requires at least two nodes to communicate for any communication. To manage or maintain node activity as per network condition.

This is high level understanding which I am trying to provide. For more details please refer:

http://www.autosar.org/

Share this:

Twitter Facebook

Like

Be the first to like this.