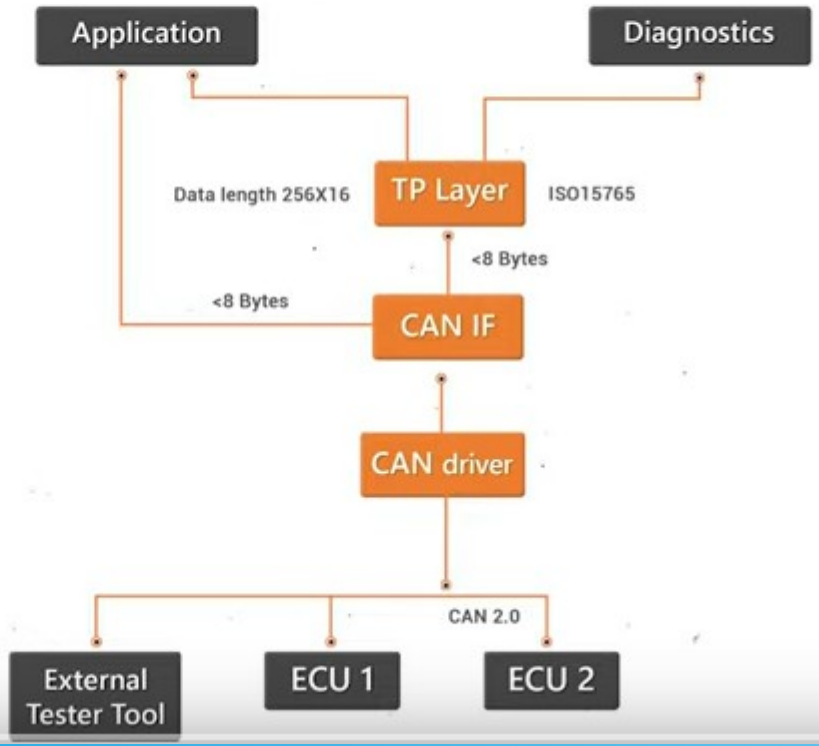
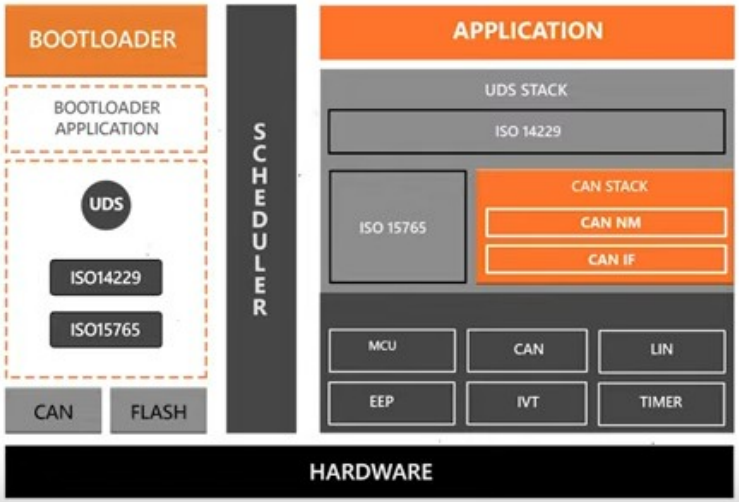
CAN Bus Protocol Stack Part-II: Exploring the Controller Area Network Software Architecture

Understanding the Software Architecture

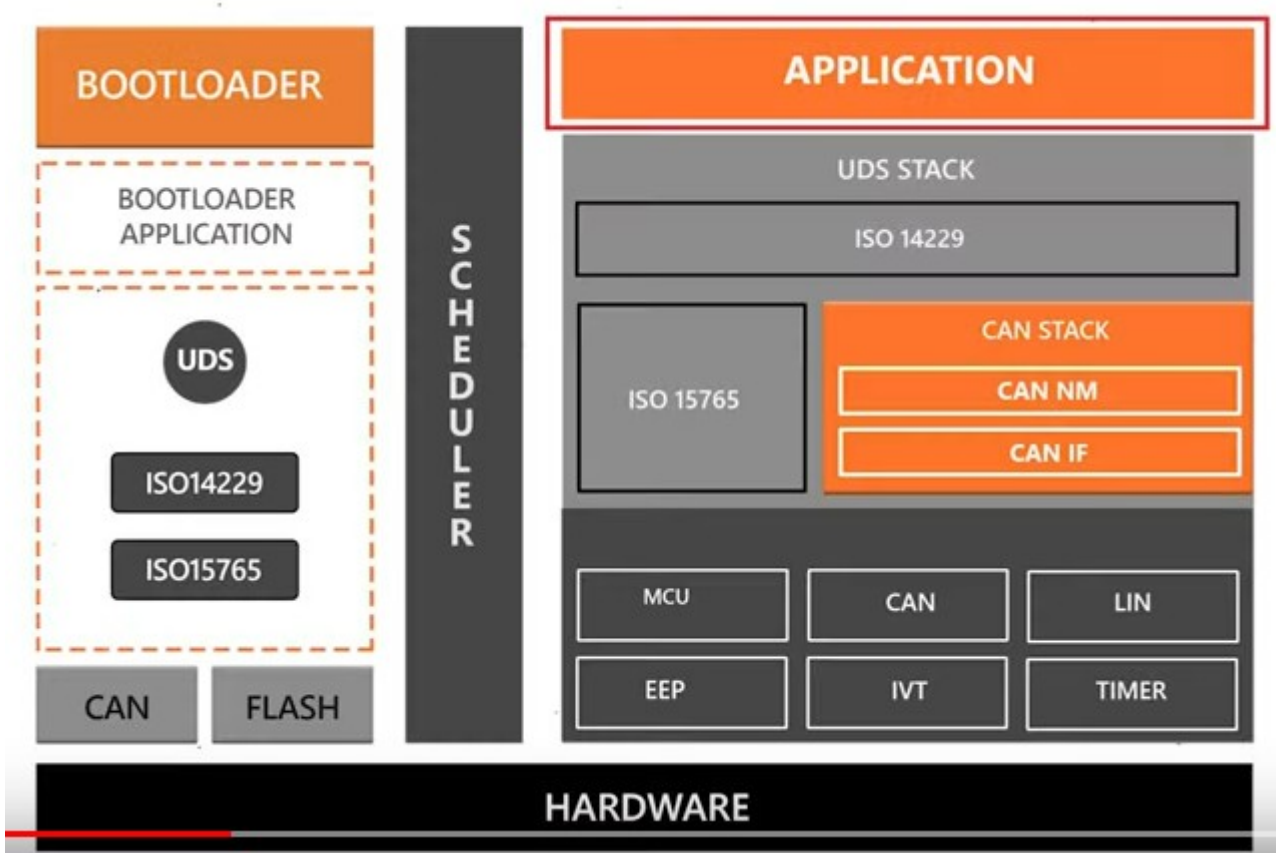


Let's Sneak a Peek of the Software Architecture Diagram



Some Interesting Insights:

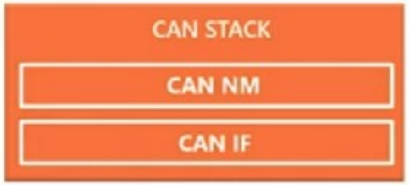
- Software Architecture of the CAN Stack is Layered
- Lower Layers provide Data/Functions/Information required by the Upper Layers
- CAN Stack Layer has services for CAN message sending/receiving
- ISO TP layer aids in sending/receiving larger CAN Messages (>8 Bytes)
- Flash Bootloader is used for ECU Reprogramming


BOOTLOADER
BOOTLOADER APPLICATION
UDS
ISO14229
ISO15765
CAN
FLASH
SCHEDULER
APPLICATION
UDS STACK
ISO 14229
ISO 15765
CAN STACK
CAN NM
CAN IF
MCU
CAN
LIN
EEP
IVT
TIMER
HARDWARE

APPLICATION

- It's the upper most layer of the **CAN** BUS Protocol
- All the code related to Automotive Applications are hosted here.
- All the Applications (like EPS module, Powertrain, etc.) interact with **ECUs'** as per the **CAN** BUS Protocol

Communication over CAN BUS is facilitated by this layer.



CAN Network Management

- Handles Sleep/Wake Up functionality of the CAN Node
- No message in the queue = Activate Sleep Mode of CAN Code
- Message detected = Activate Wake Up Mode

CAN Interface Layer

- Supports Hardware Abstraction
- Facilitates flow of data to and from Application Layer
- Comprises of services like Transmit Request, Controller Mode control etc.
- Configured based on CAN Matrix file

ISO 15765

- When Data Frame Payload > 8 bytes; TP Layer facilitates data transmission
- .This specialized protocol segments the data into multiple frames before transmitting
- Data frames are reassembled by the **ISO TP** layer, at the receiving **CAN** node
- ISO 15765 is also known as the Transport Layer (**ISO TP** Layer)

CAN

- Enables access to hardware resources for upper software layers modules
- Supports the following software services
  - Initiate transmission and **CALLBACK** function to **CAN IF** layer
  - Manage state and behavior of **CAN** Controller
- Handles Hardware based Acceptance Filtering of signals and messages

Flash Bootloader

- Manages **ECU** Reprogramming of Automotive **ECUs**
- A **UDS (ISO 14229)** based Flash Bootloader comprises **UDS** stack as well
- **ISO TP (ISO 15765)** layer helps in transmission of larger **ECU** flashing files