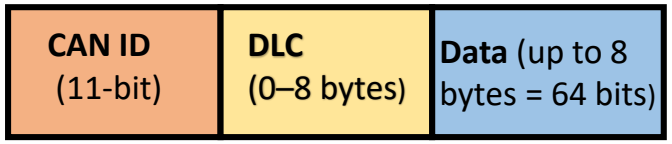
CAN Message and Signal Extraction Using DBC

❑ CAN Basics

- **CAN (Controller Area Network)** is a message-based protocol used for ECU communication in vehicles.
- Communication happens using **CAN frames** identified by **CAN ID**
- Actual vehicle signals are **packed as bits inside CAN data**.

❑ Standard CAN frame:



❑ DLC vs Data Length

- **DLC** = number of bytes sent in CAN frame
- **Data length** = actual bytes used by signals Padding fills the remaining bytes to match DLC

❑ Signals in CAN Data

- CAN data contains multiple **signals**
- **Signal** = a specific set of bits representing physical values
- Signals have:
  - ➢ Start bit
  - ➢ Bit length
  - ➢ Endianness (Little/Big)
  - ➢ Scaling factor & offset

❑ DBC File & Its Need

- DBC (Database CAN) is a **definition file** for CAN communication
- It Contains:
  - ➢ Message ID & name
  - ➢ DLC
  - ➢ Signal names, start bit, length, scaling, units
- Used by software/tools to **decode CAN messages automatically**

Parameter	Standard CAN	Extended CAN
ID Length	11 bits	29 bits
Message Count	Limited	Very Large
Network Size	Small	Large

❑ Bit Extraction:

Selecting specific bits from CAN data using DBC-defined start bit and length to obtain the raw signal value.

❑ Tool & DBC Used

- **Tool:** Vector CANdb++
- **DBC File:** opel_omega_2001.dbc
- **CAN Type:** Classical CAN (8-byte messages)
- **ECUs:** ABS, ESP, ECU, TCU, SAS

❑ Example from DBC File

- **Message Name:** ABS WheelSpeed
- **CAN ID:** 768 (0x300)
- **DLC:** 8 bytes
- **Transmitter:** ABS EC
- **Signal Name:** FrontLeftWheelSpeed
- **Signal Definition (DBC line)**

SG_ FrontLeftWheelSpeed : 5|14@0+ (0.112,0) "km/h"

  - ➢ **Start Bit:** 5
  - ➢ **Length:** 14 bits
  - ➢ **Endianness:** @0 → Big Endian (Motorola format)
  - ➢ **Data Type:** Unsigned (+)
  - ➢ **Factor (Scaling):** 0.112
    (Raw value × 0.112 = physical value)
  - ➢ **Offset:** 0
  - ➢ **Unit:** km/h