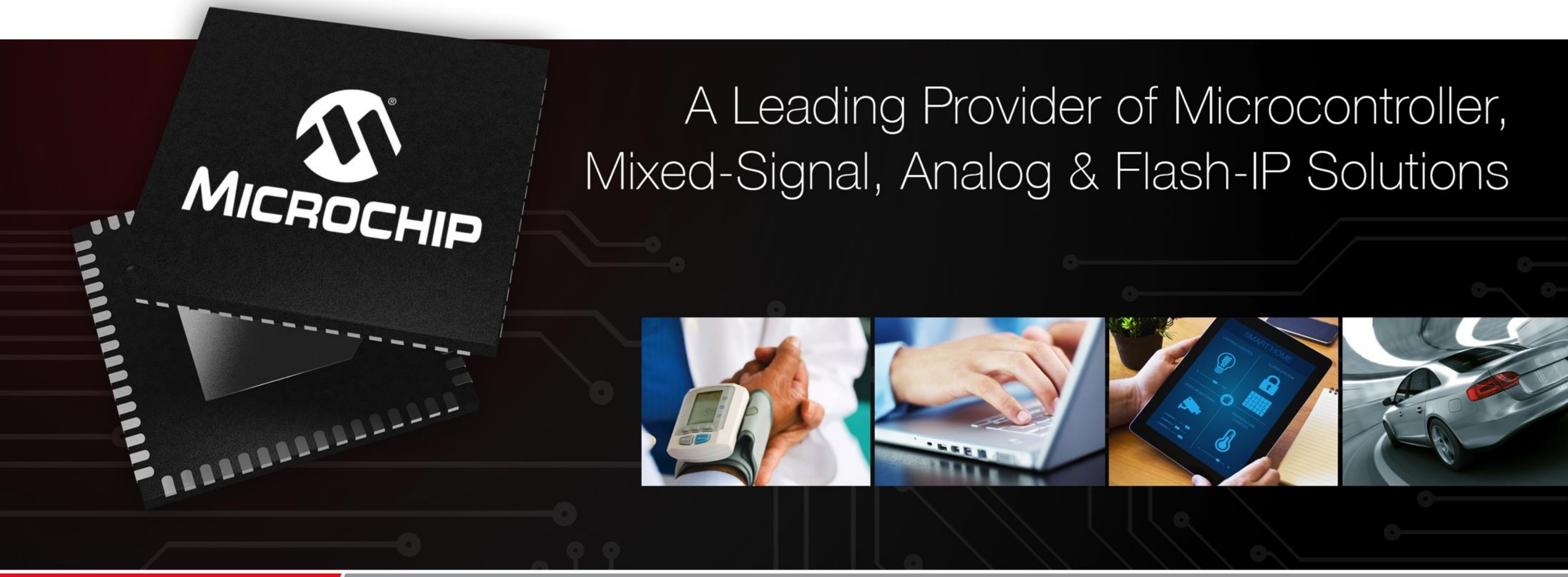
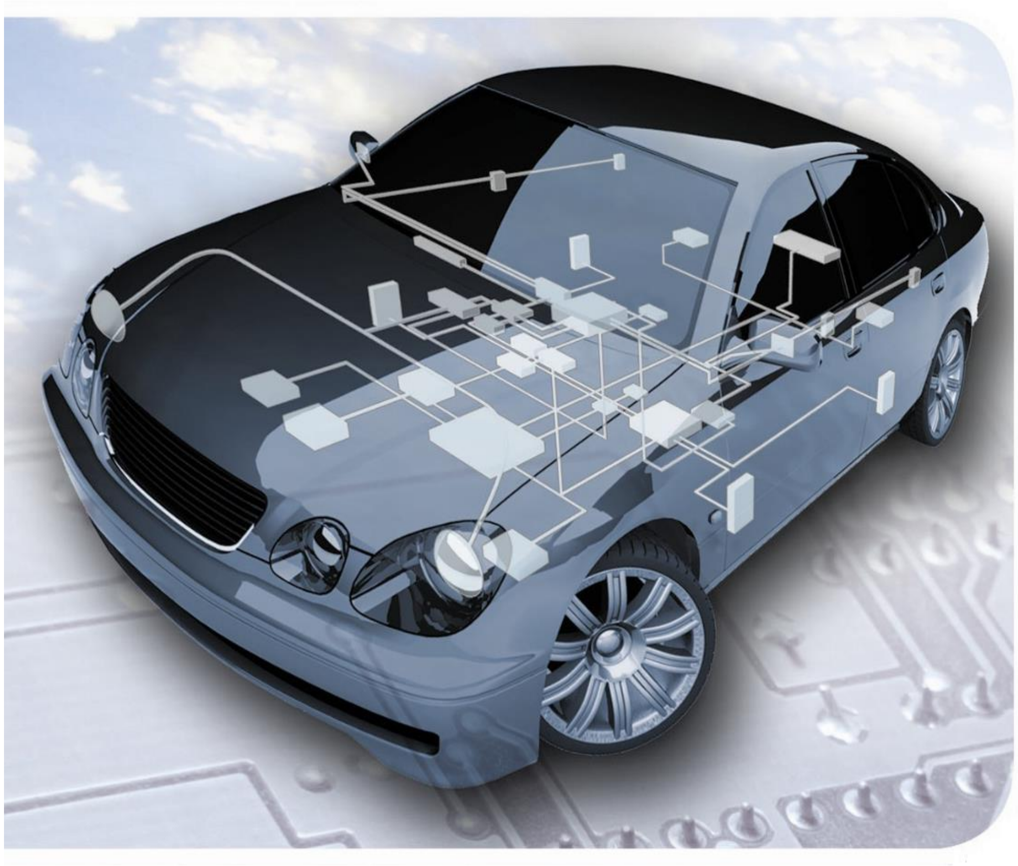
MICROCHIP



Presented by:
Jimmy Yoon, Sr. Embedded Solutions Engineer
March 13, 2019

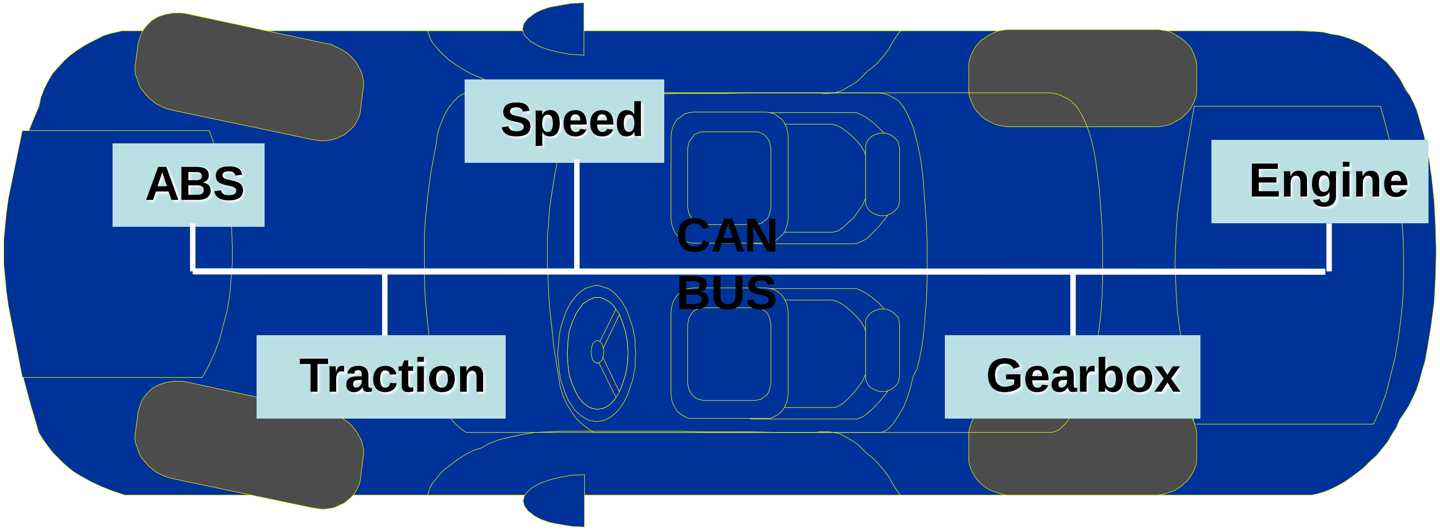
Automotive Network Basic: CAN/LIN Concept and Architecture

CAN (Controller Area Network)



CAN Overview

- CAN is a serial communication protocol
- All nodes attach to common connection
- All nodes must use the same baud rate
- Each node can transmit or receive any message on the bus



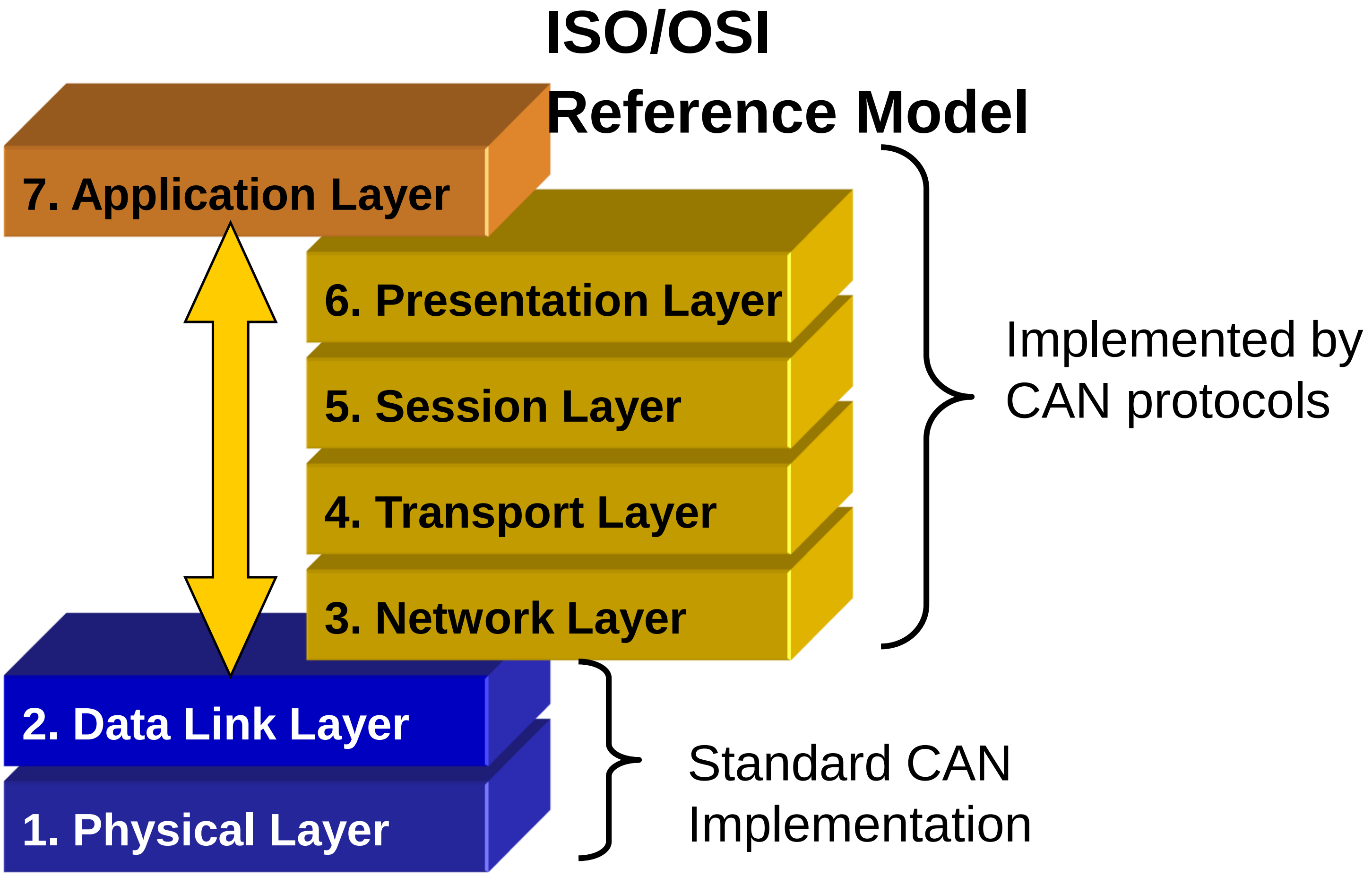
Requirements and Applications

- **CAN requires**
  - Information sharing between vehicle ECUs (engine ECU, mission ECU, ABS, air-bag, ETACS, etc.)
  - Robust communication is required for a noisy environment
  - Network requirements between independent trending ECUs

- **CAN applications**
  - Automotive: sensors and information sharing
  - Industrial facilities: many processors

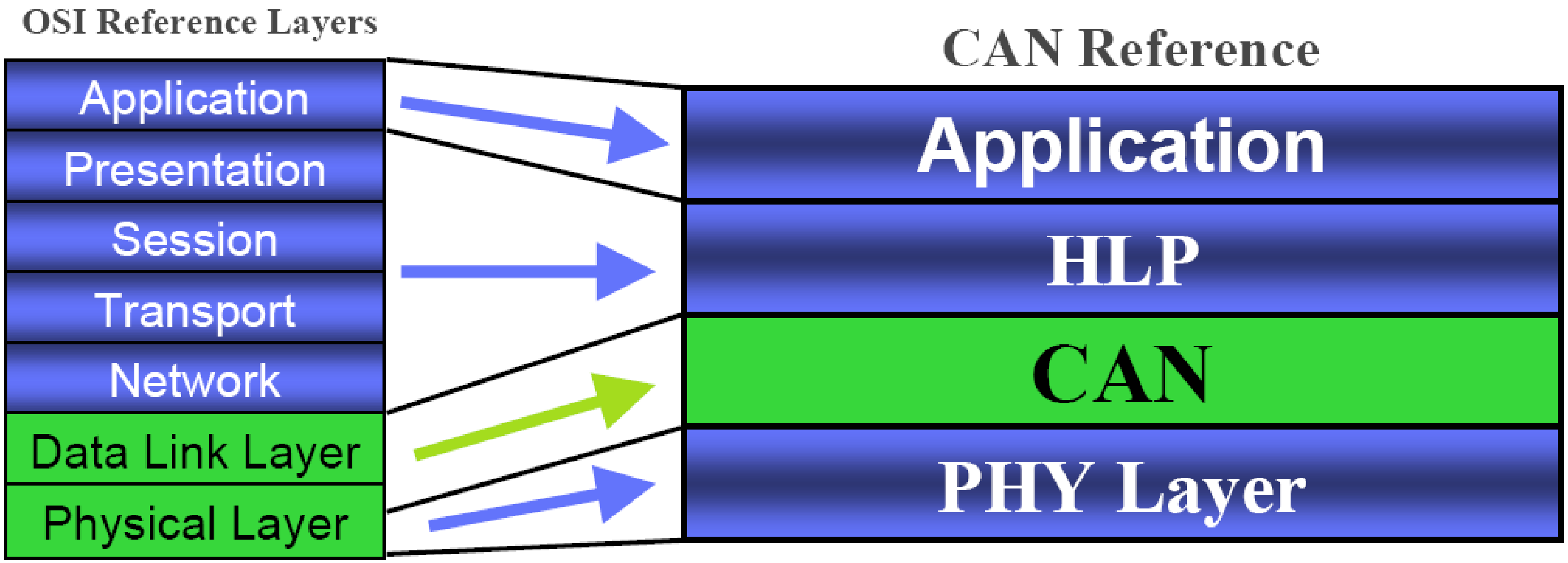
CAN Network Model

- **ISO/OSI Seven Layer Network Reference Model**

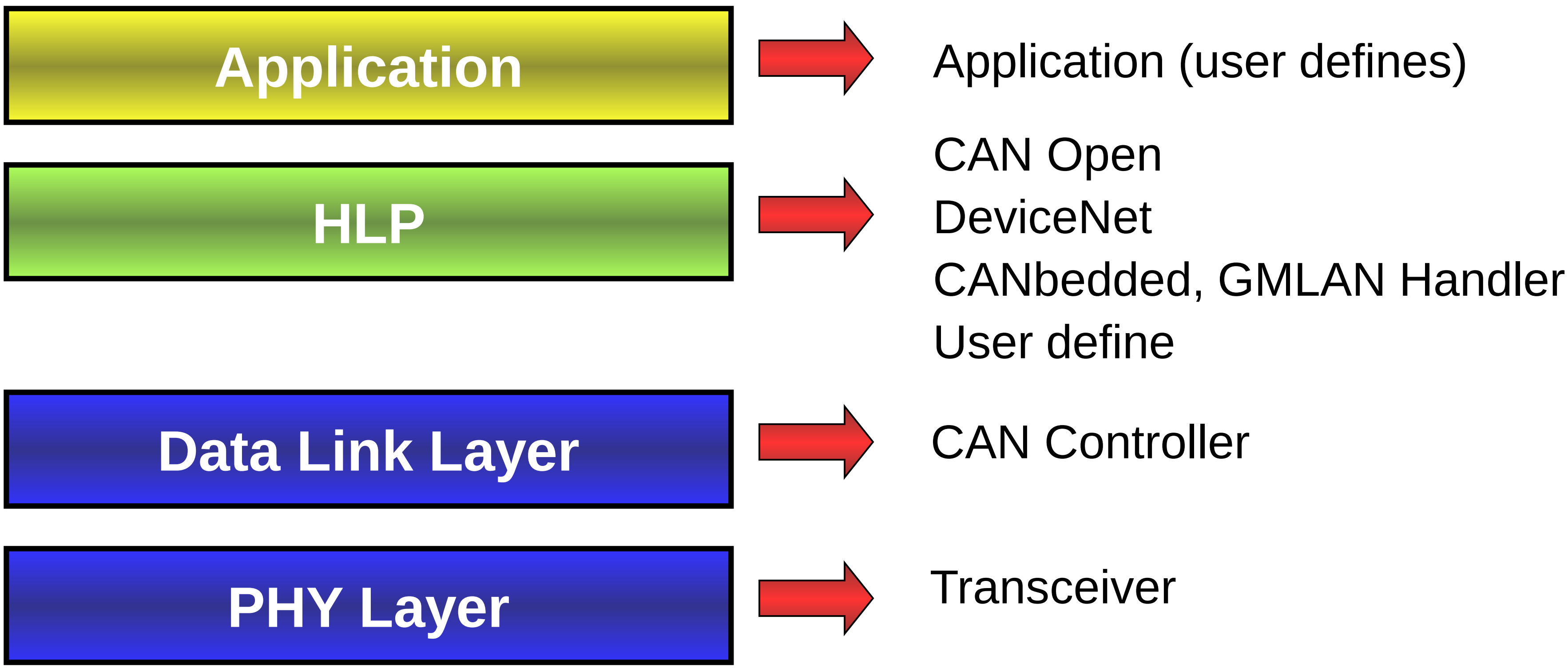


CAN Network Model

- **The CAN specification defines a part of the Data Link Layer and Physical Layer**



CAN Network Model

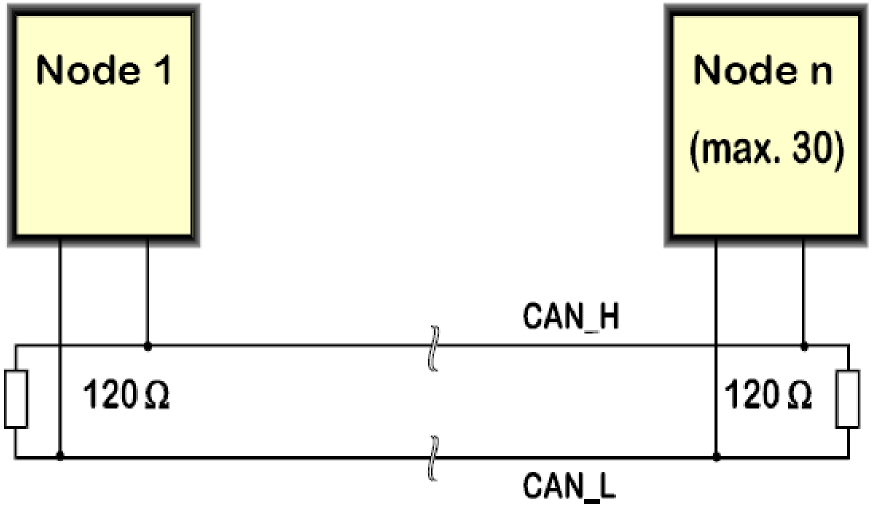


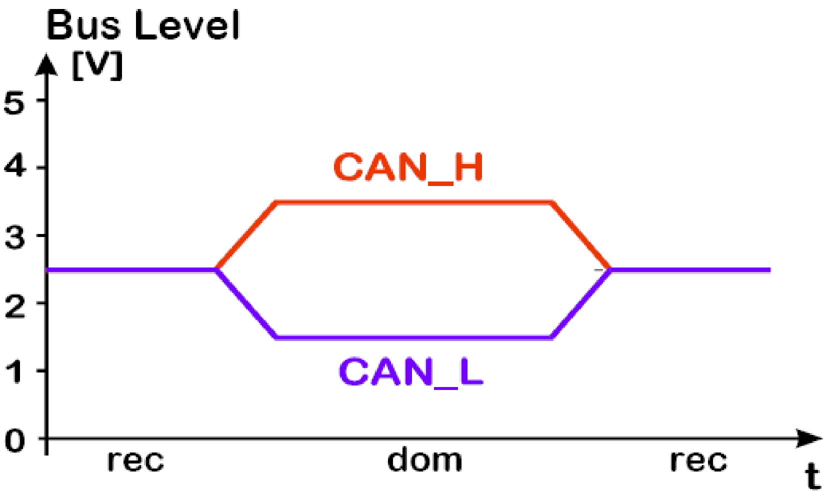
CAN Physical Layer

- **CAN High Speed**
  - ISO11898-2
  - Supports up to 1Mbps bus speed
- **CAN Low Speed (Fault Tolerant)**
  - ISO11898-3
  - Supports up to 125kbps bus speed

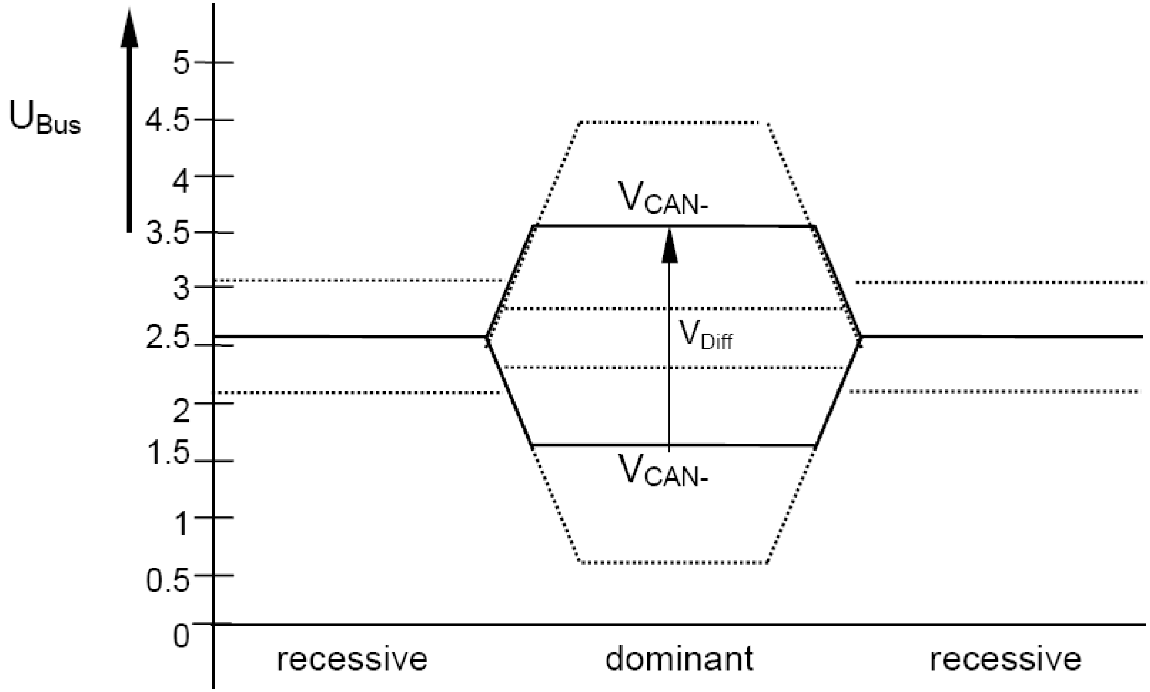
CAN High Speed Physical Layer

- ISO 11898, CAN Class C
- Symmetrical signal transmission
- 40m max cable length @1Mbps
- 1km max cable length @50kbps
- Transmission output current >25 mA



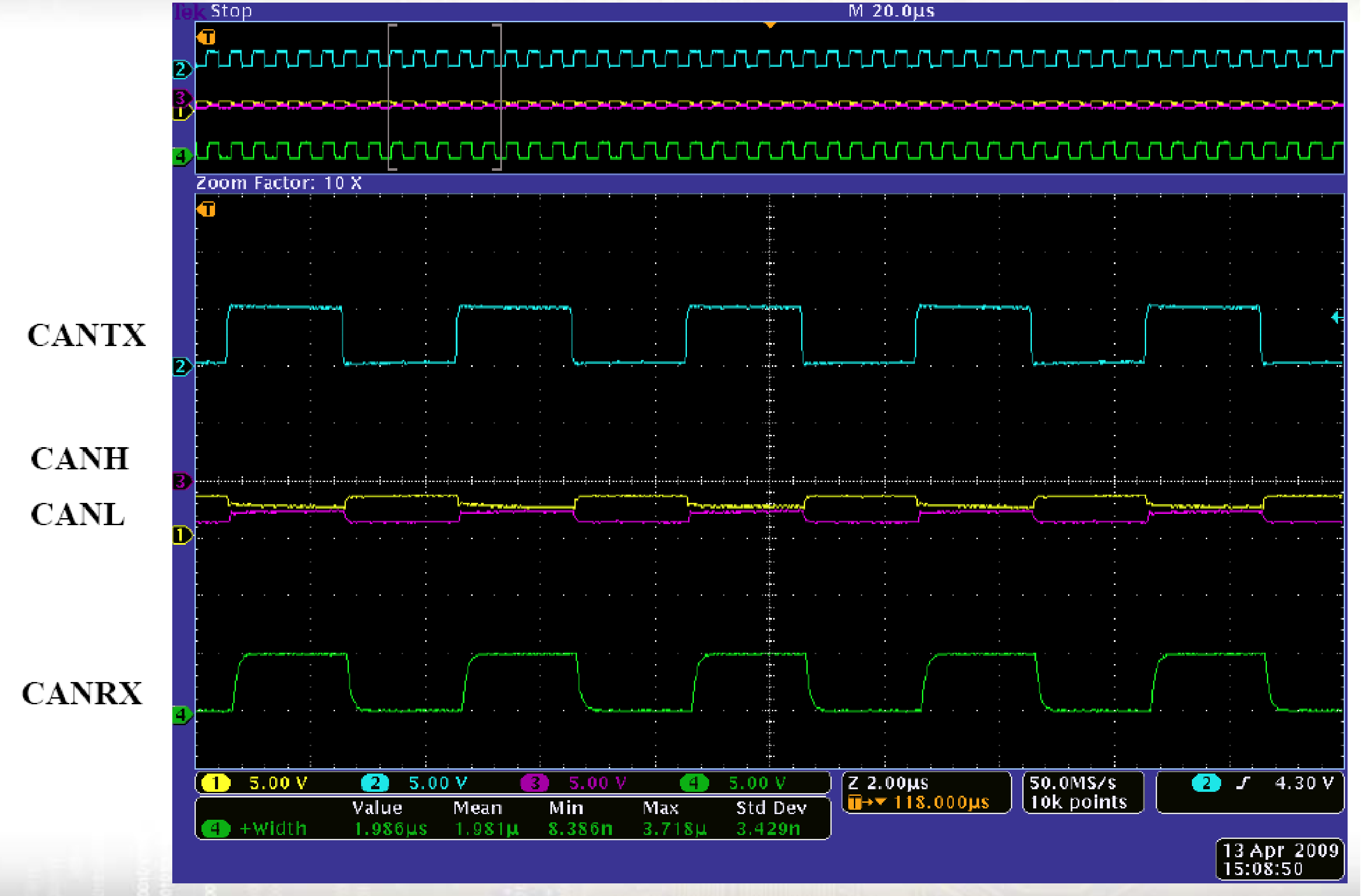


CAN High Speed Physical Layer



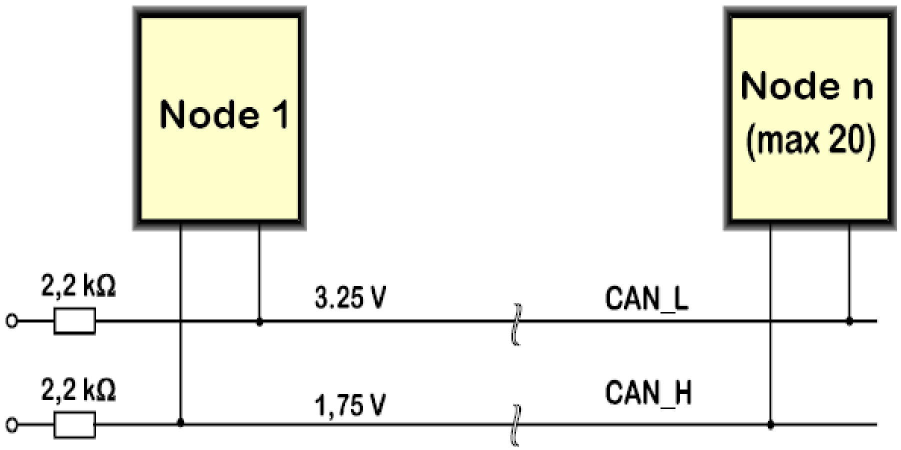
	recessive [V]			dominant [V]		
	min.	typ.	max.	min.	typ.	max.
CAN high	2.00	2.50	3.00	2.75	3.50	4.50
CAN low	2.00	2.50	3.00	0.50	1.50	2.25
Difference	-0.50	0.00	0.05	1.50	2.00	3.00

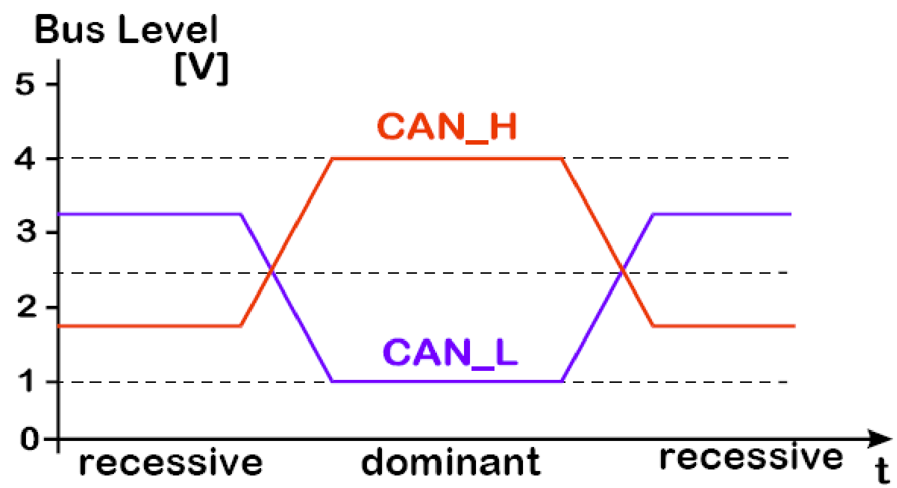
CAN Bus Waveform



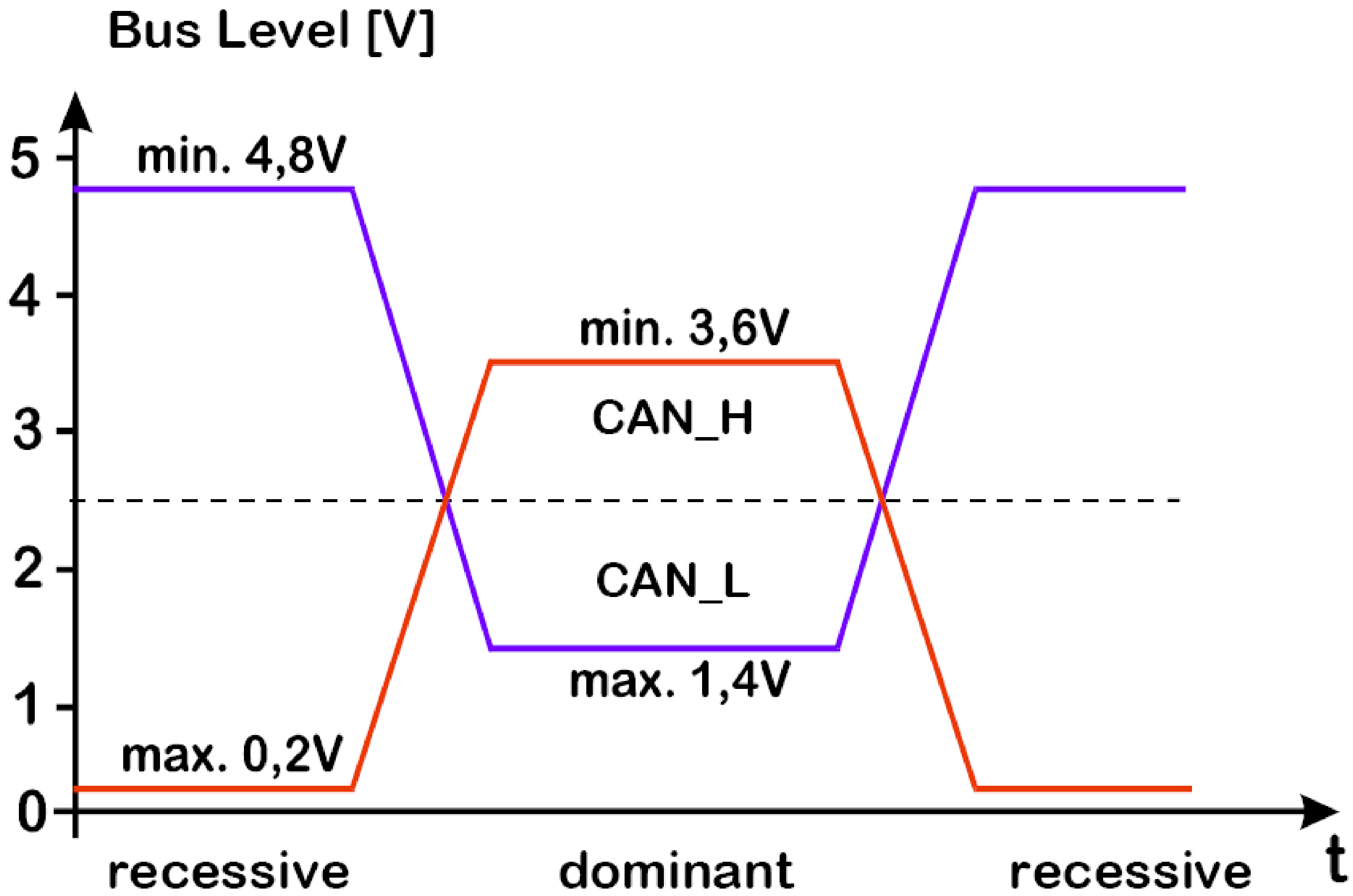
CAN Low Speed Physical Layer

- **ISO 11519, CAN Class B**
- **Symmetrical signal transmission**
- **Data transfer rate up to 125kb/sec**
- **The maximum bus line length depends on the baud rate**
- **Transmission output current <1 mA**

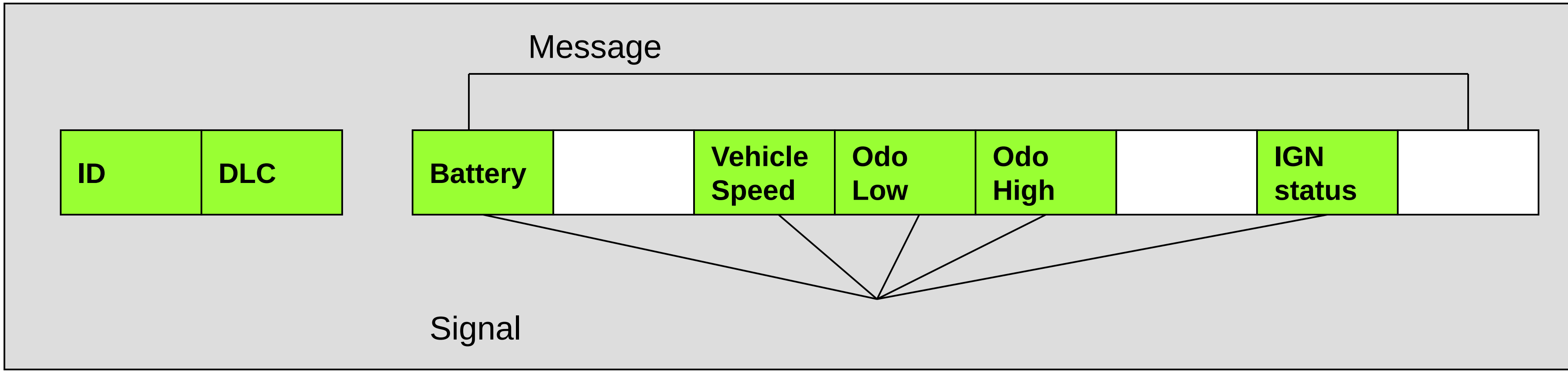




CAN Low Speed Physical Layer



Message and Signal



	80	40	20	10	08	04	02	01
0			Battery					
1								
2			Vehicle Speed					
3			Odo Low					
4			Odo High					
5								
6						IGN		
7								

Key Features of CAN

- **Multi Master**
- **CSMA (Carrier Sense Multiple Access)**
- **CD-CR (Collision Detection with Collision Resolution)**
- **Message based, not address based**
- **Up to 8 byte transfer**
- **Maximum communication speed 1Mbit**
- **Reliability with various error detection**
- **Noise-resistant communication**

CSMA / CD-CR

- **CS** • **Carrier Sense – Every node must monitor bus for a period of no activity before sending a message**
- **MA** • **Multiple Access – Once a period of no activity occurs, every node has an equal opportunity to transmit a message**
- **CD** • **Collision Detection – If 2 nodes transmit at the same time, a collision occurs**
- **CR** • **Collision Resolution – Non-destructive bitwise arbitration**

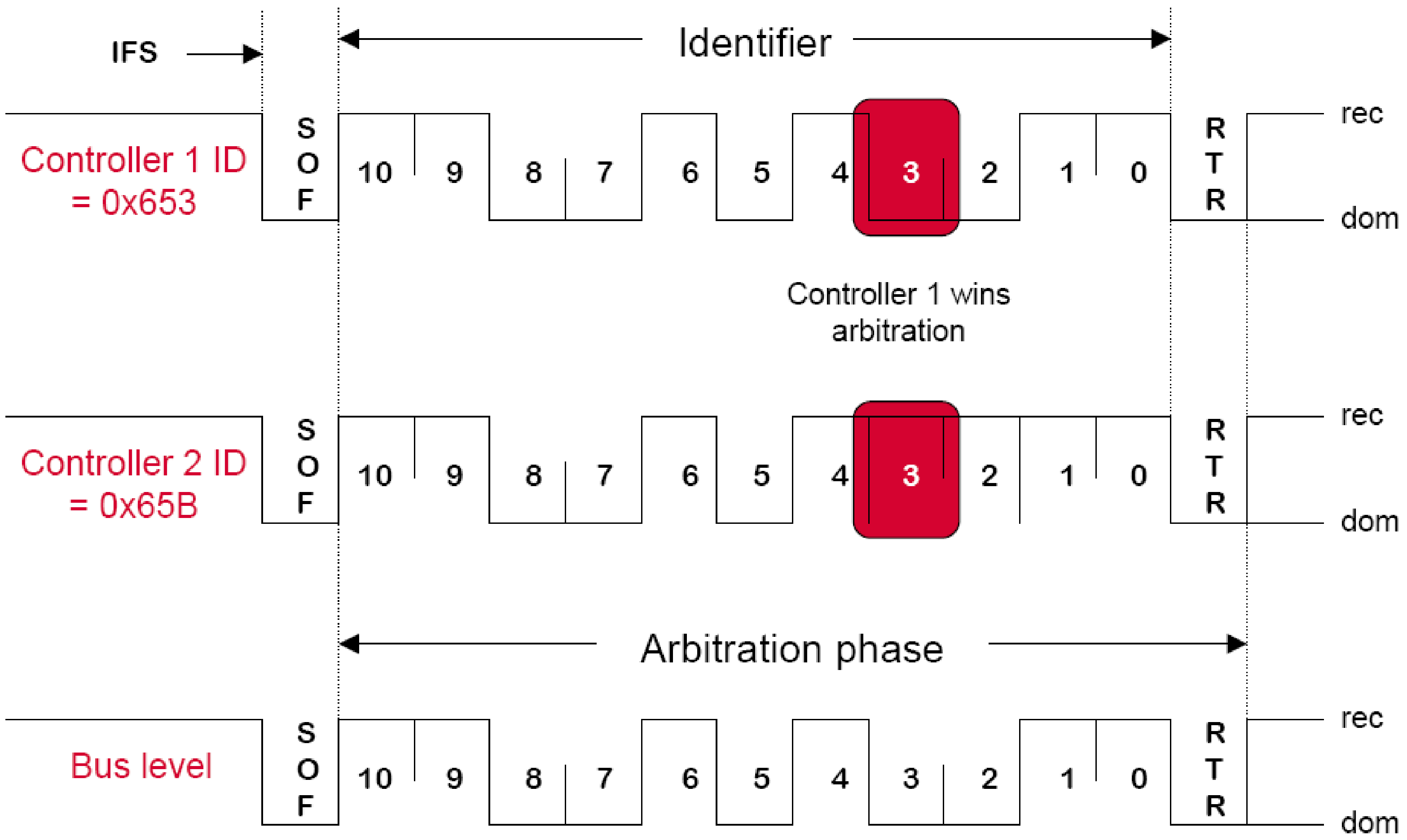
CSMA / CD-CR

CR

- Collision Resolution - Non-destructive bitwise arbitration
  - Messages remain intact even after collision occurs
  - All arbitration takes place without corruption or delay of the highest priority message
  - Any message that loses priority in arbitration is automatically retransmitted at the next available time
  - Requirements
    - Dominant and recessive bit states must be defined (dominant wins arbitration over recessive)
    - Each node monitors bus to see if what was sent actually appears on the bus

CSMA / CD-CR

- **Collision Resolution – Non-destructive bitwise arbitration**

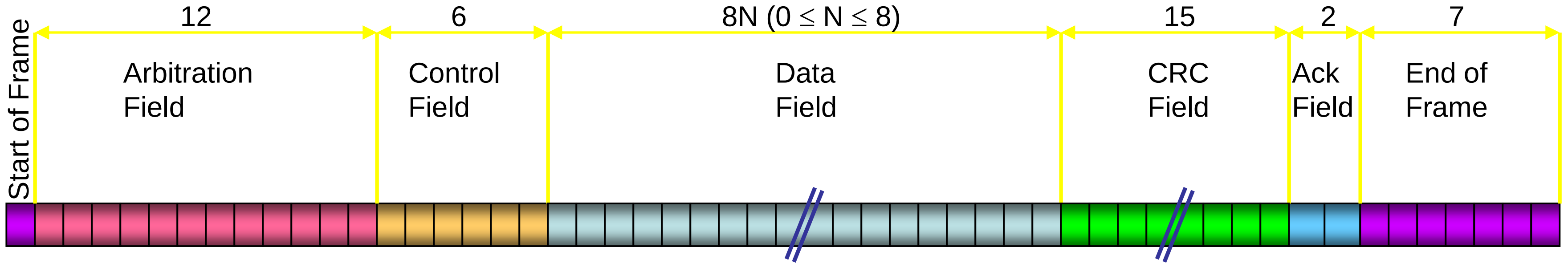


Message Based

- All messages sent between each node are not address-based
- The transmitted message itself is the data with priority
- Each node can receive all messages and send ACK if there is no abnormality in received contents (CAN Peripheral H/W)
- Process with received message with Mask & Filter
- One-to-one or multiple transmission are possible
- Existing nodes do not need to update information about adding new node

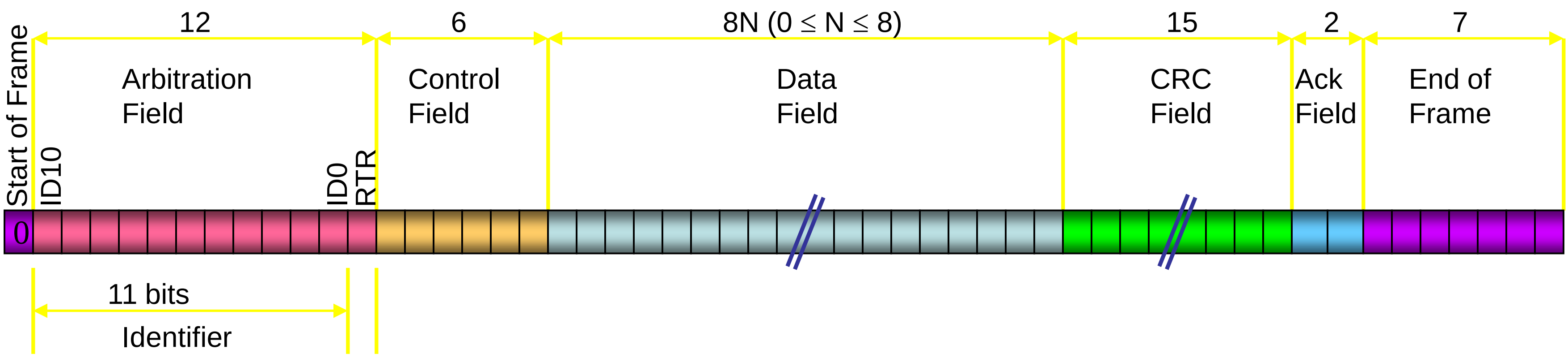
Message Frames (Data Frames)

- CAN 2.0A / CAN 2.0B Format
- Standard Data Frame
  - Versions 1.0 and 2.0A
  - 11-bit Identifier Field



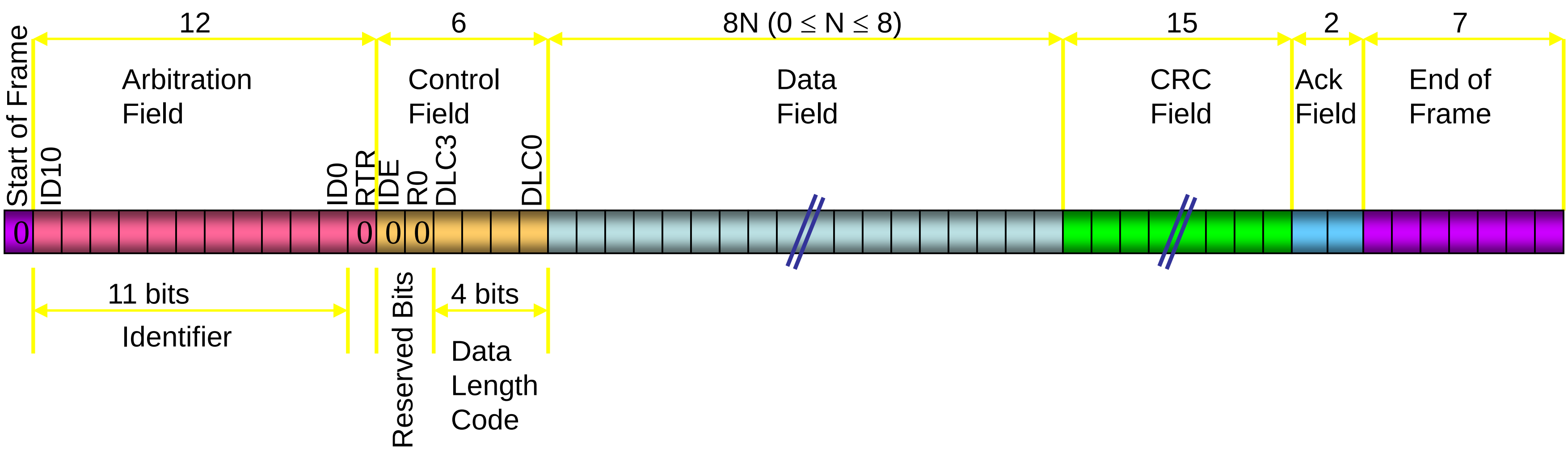
Message Frames (Data Frames)

- Standard Data Frame (Arbitration Field)
  - SOF(Start of Frame): Default - 0
  - Identifier: 11bit
  - RTR: 0 – Data Frame, 1 – Remote Request



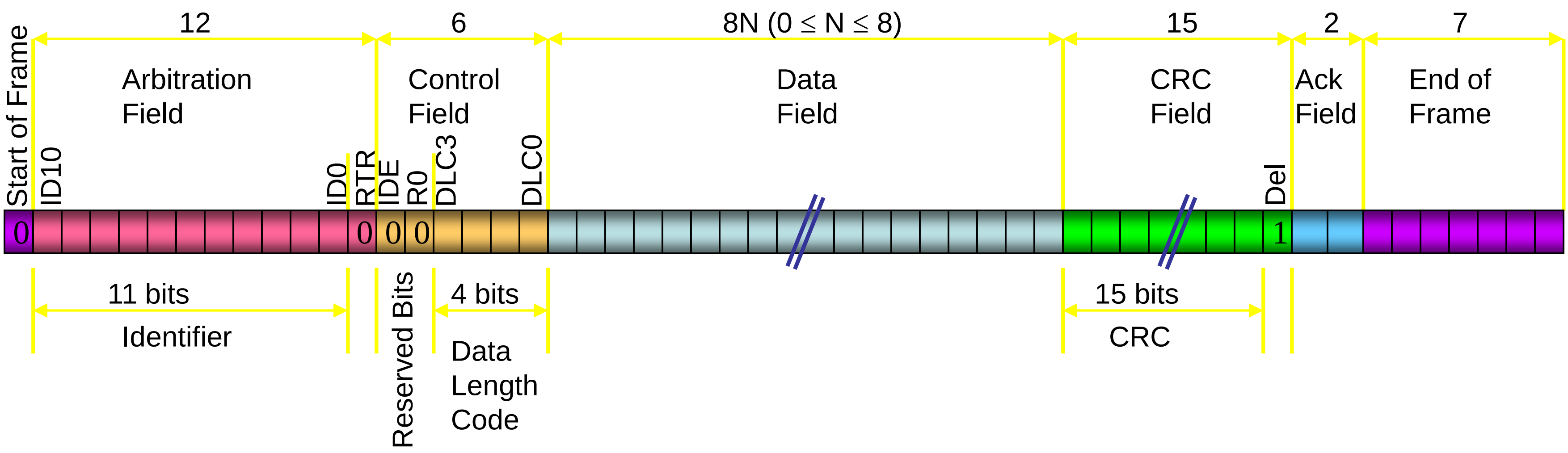
Message Frames (Data Frames)

- Standard Data Frame (Control Filed)
  - IDE & R0: Reserved bit, Default 0
  - DLC: Number of Data Field Byte



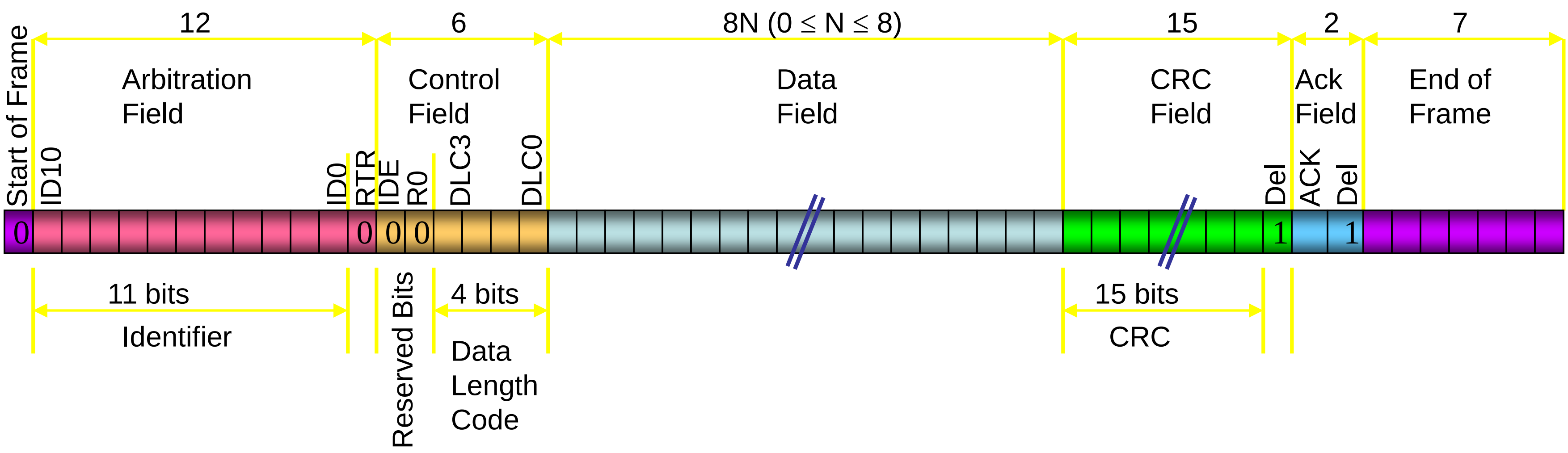
Message Frames (Data Frames)

- Standard Data Frame(CRC Field)
  - 15Bit CRC(Cyclic Redundancy Check)
    - CRC from SOF to Data Field
  - 1Bit CRC Delimiter : Default 1



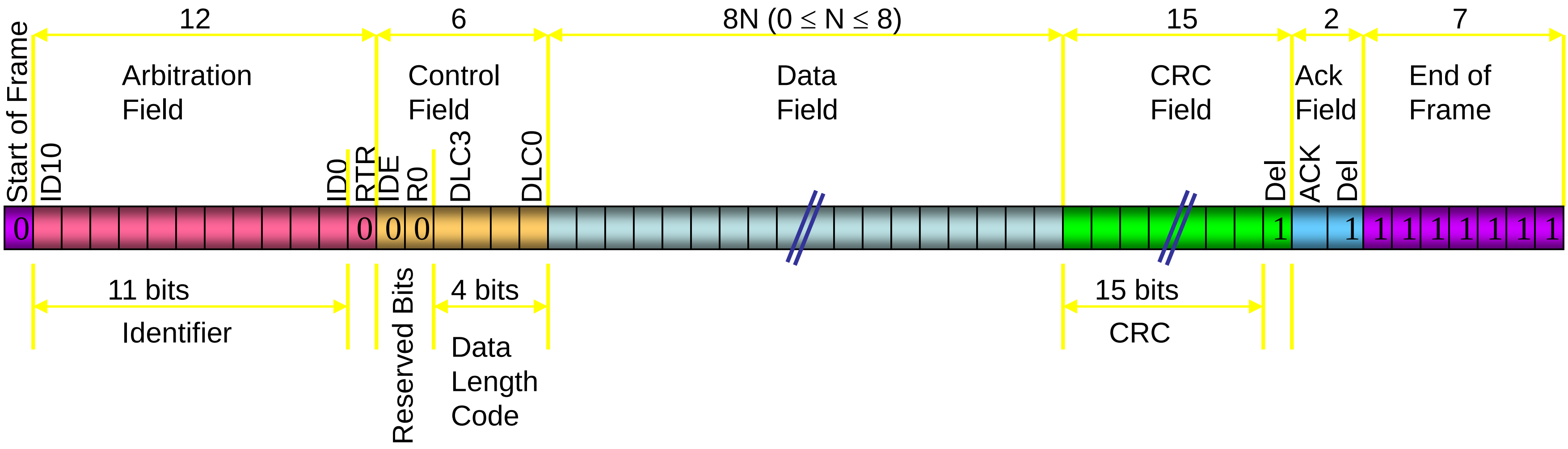
Message Frames (Data Frames)

- Standard Data Frame(ACK Field)
  - ACK Slot:
    - Transmitting node checks the ACK Slot bit, which it has sent as a recessive and checks for a dominant
  - ACK Delimiter: Default 1



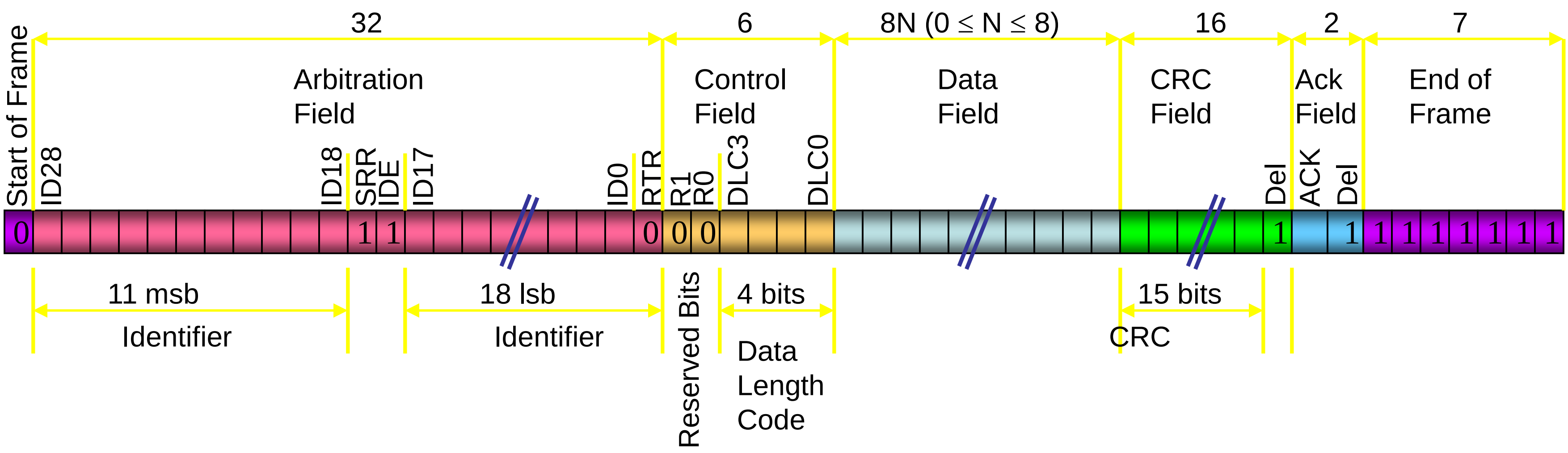
Message Frames (Data Frames)

- Standard Data Frame(End of Frame)
  - 7 Recessive Bit
  - Intermission Filed: Notify of interruption of Frame as 3 Recessive Bits after EOF
  - Bus Idle Time: Arbitrary length after Intermission Filed

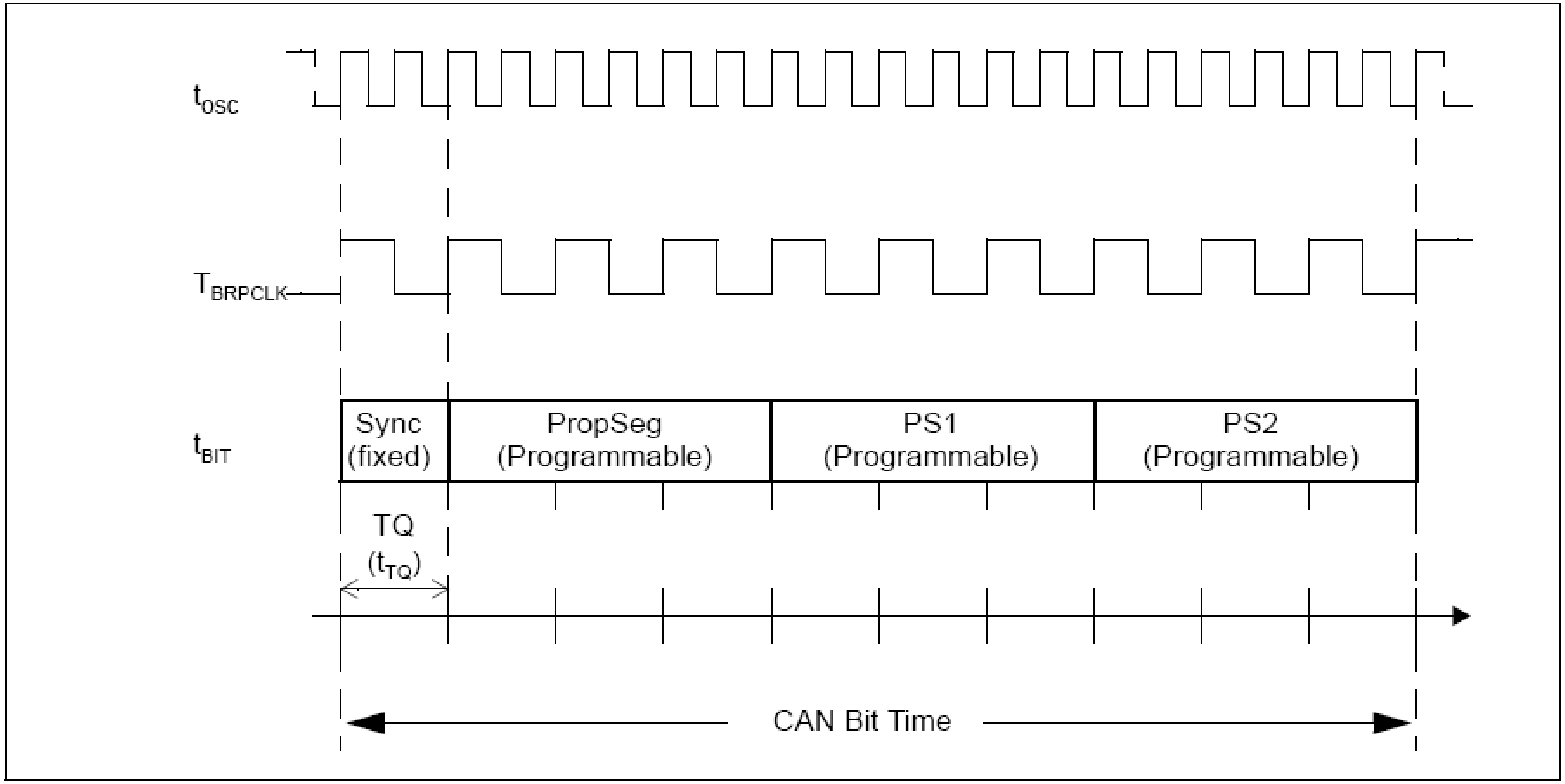


Message Frames (Data Frames)

- **Extended Data Frame**
  - Version 2.0B, 29-bit Identifier field
  - SRR(Substitute Remote Request): RTR in Standard Form
  - IDE(Identifier Extension Bit): Dominant in Standard Form

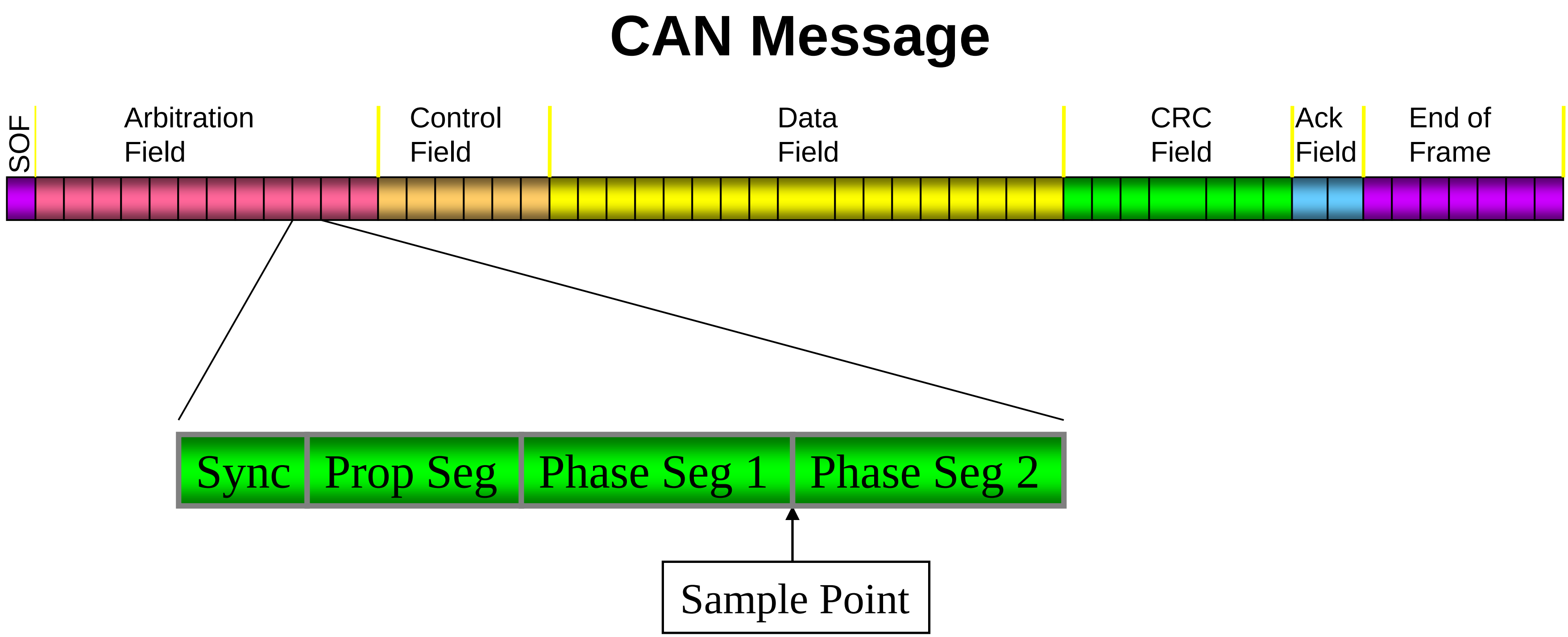


Bit Timing



tOSC
TBRPCLK
tBIT
Sync
(fixed)
PropSeg
(Programmable)
PS1
(Programmable)
PS2
(Programmable)
TQ
(tTQ)
CAN Bit Time

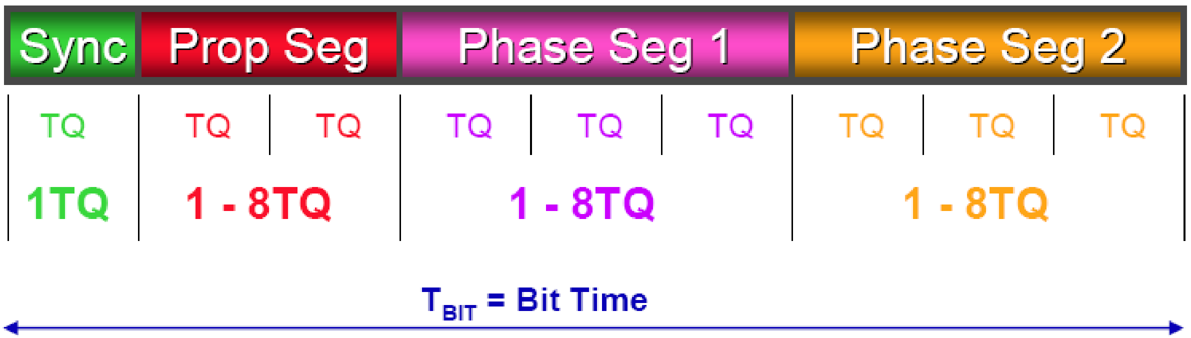
Bit Timing



A CAN message BIT is made up of four segments

Bit Timing

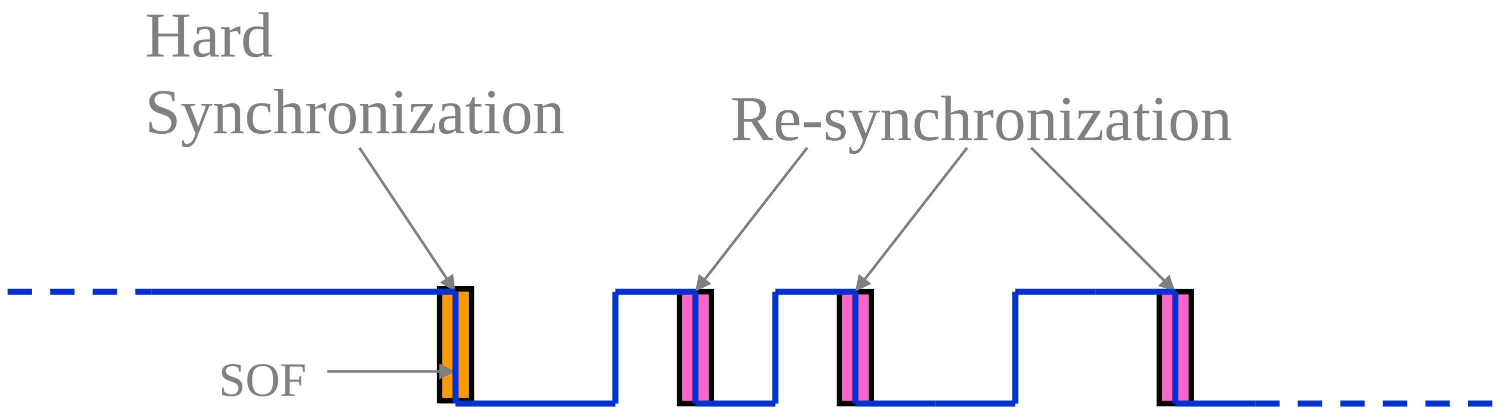
- **Each Bit Timing Segment is made up of integer units of time called Time Quanta (TQ)**



- **Nominal Bit Time : 8TQ ~ 25TQ**
- **TQ = 2(BRP)(Tosc)**
- **The Baud Rate Prescaler (BRP) modifies the TQ time**
  - Min = 1:1, Max = 1:64

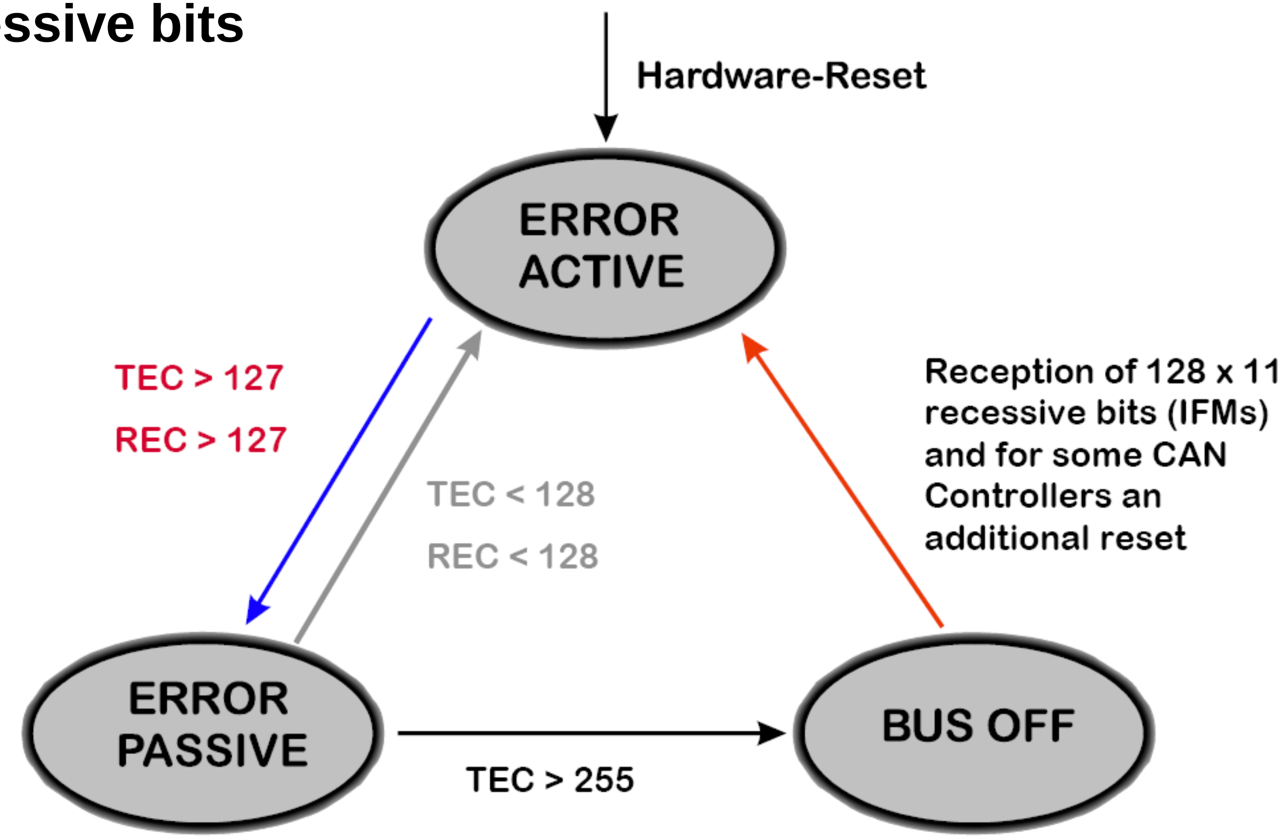
Synchronization

- **No clock in bit stream**
- **Receivers synchronize on recessive to dominant transitions**
  - Hard Synchronization occurs at SOF and resets bit clock
  - Resynchronization occurs at recessive-to-dominant (1-to-0) edges and adjusts the bit clock as necessary



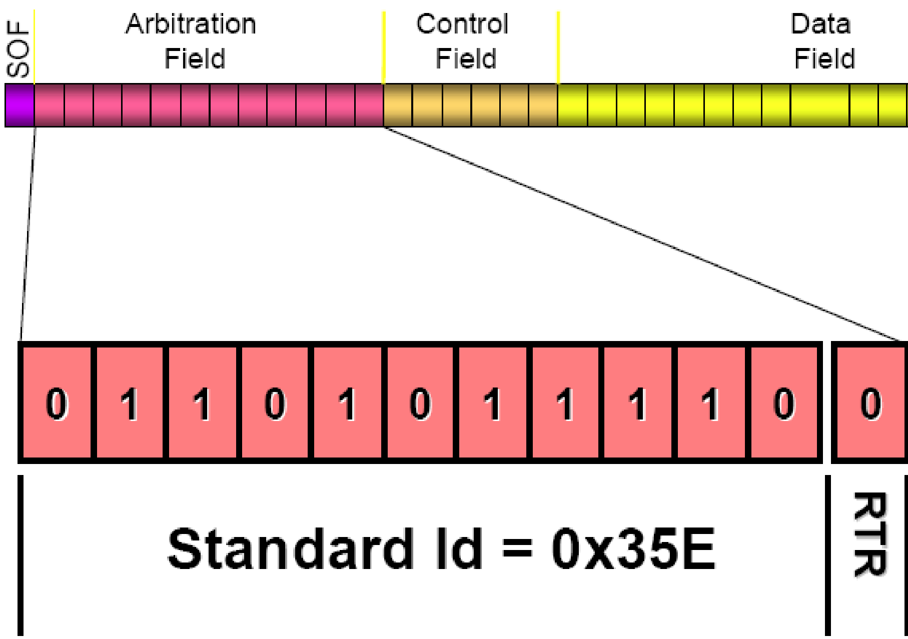
Error Handling

- **To prevent loss of network by Faulty Node**
- **Error Passive: message reception and error frame transmission**
- **Bus off → Error Active: 128 occurrence of 11 consecutive recessive bits**



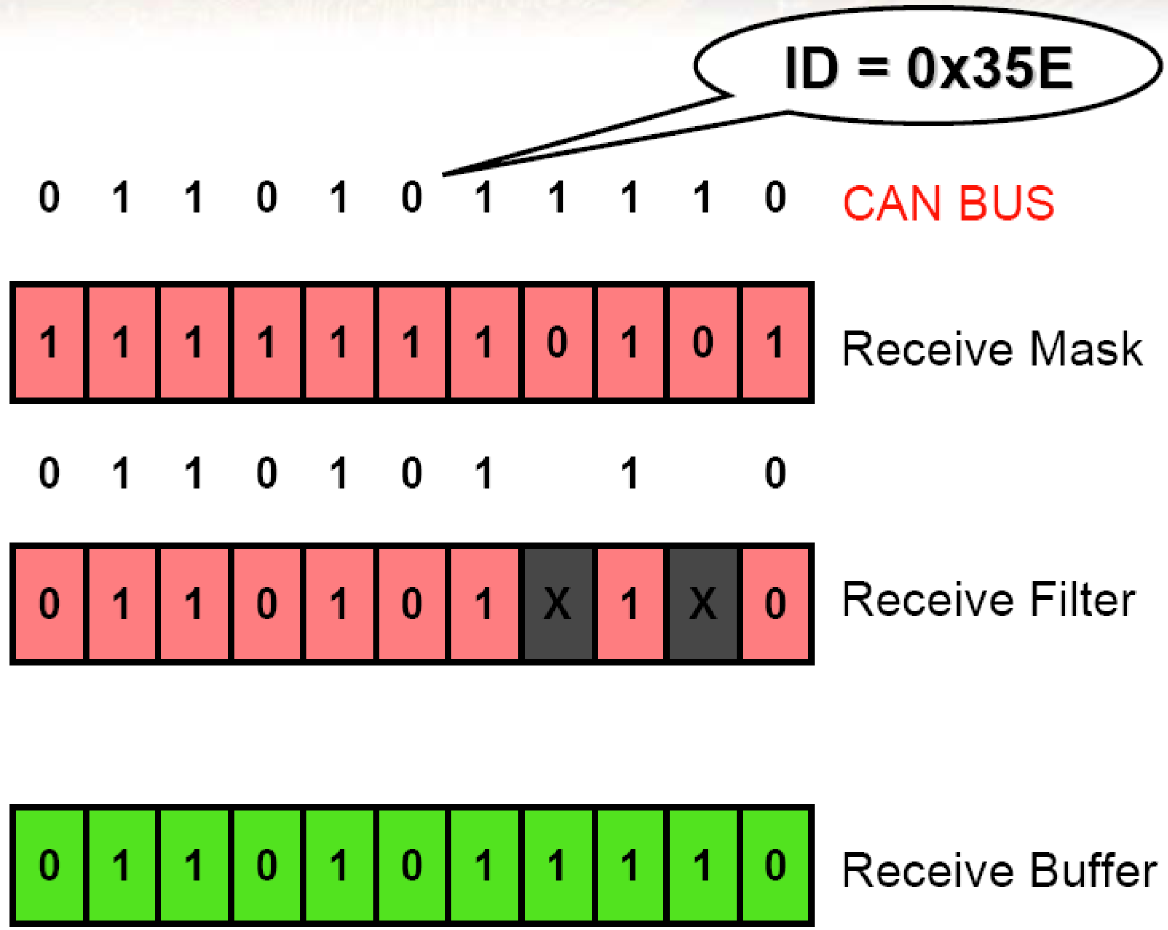
Mask and Filter

CAN Message



FILTER/MASK TRUTH TABLE			
Mask Bit n	Filter Bit n	Message Identifier Bit n	Accept or Reject bit n
0	X	X	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

Mask and Filter

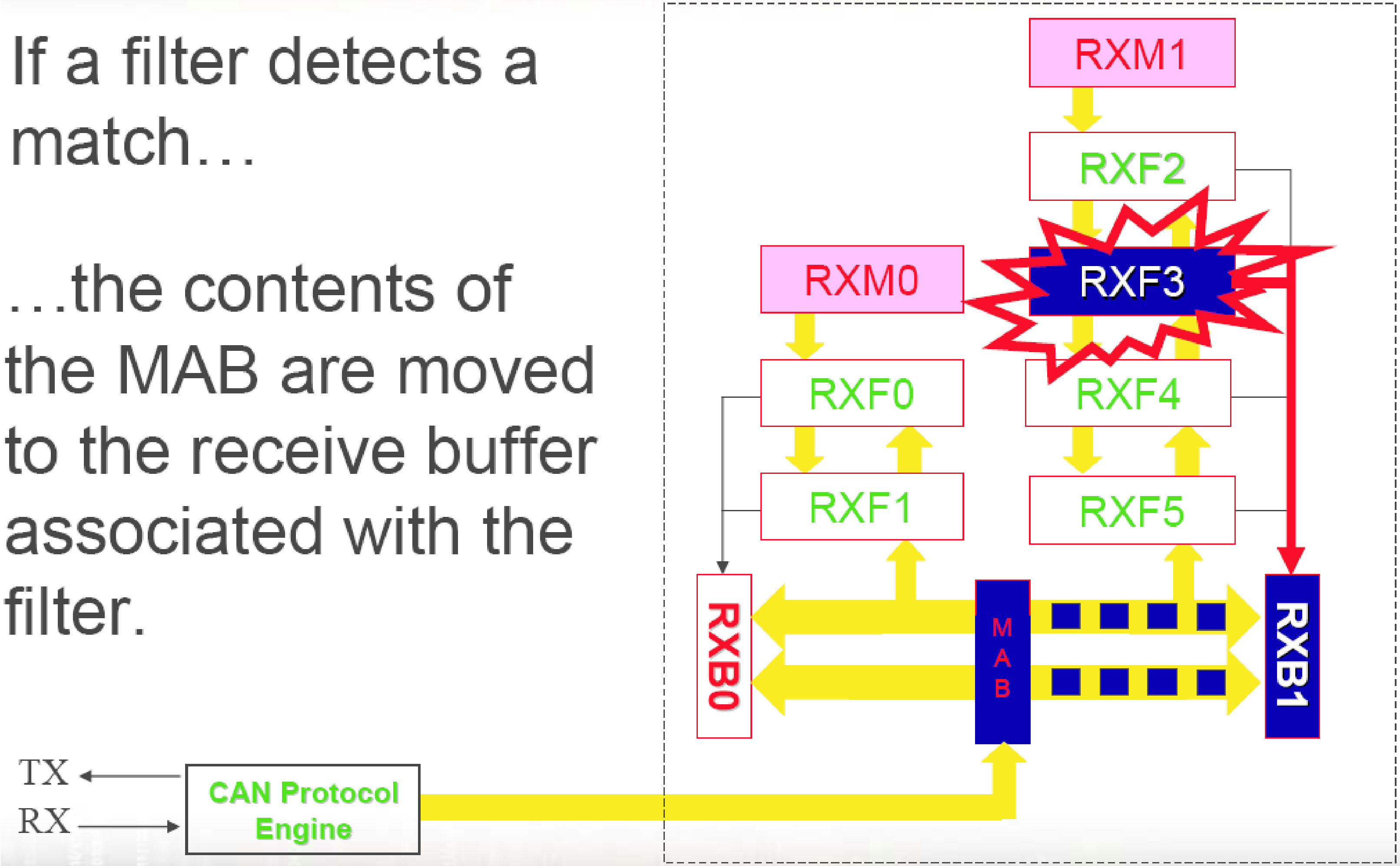


Mask and Filter set for ID's 0x354,0x356,0x35C,0x35E

Mask and Filter

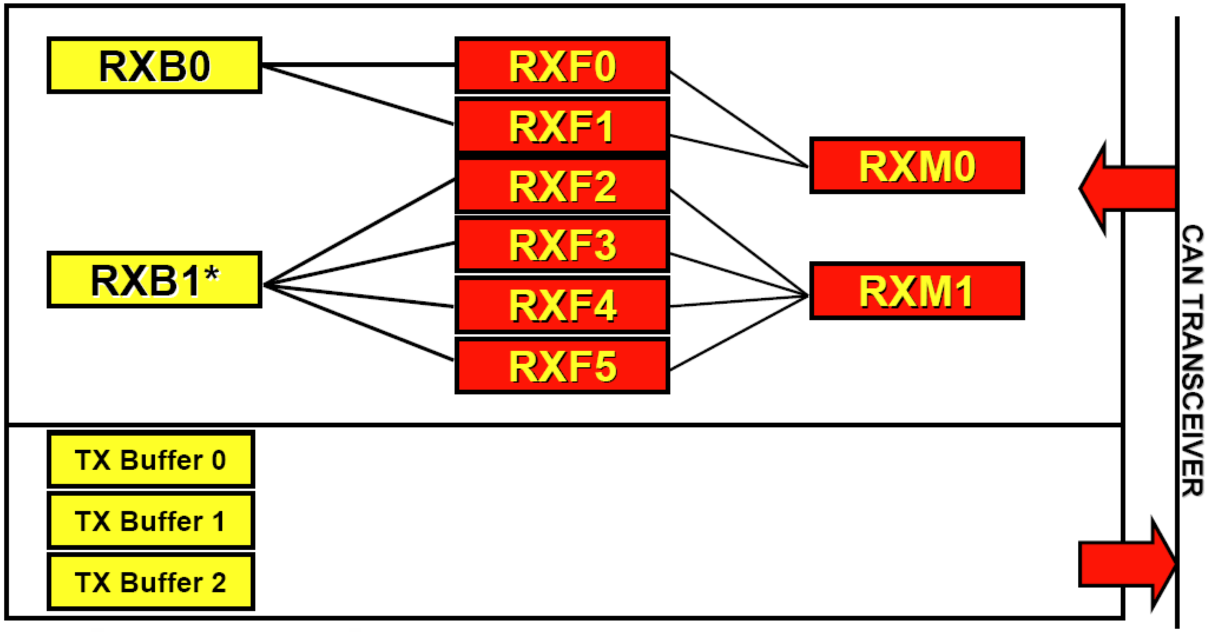
If a filter detects a match…

…the contents of the MAB are moved to the receive buffer associated with the filter.



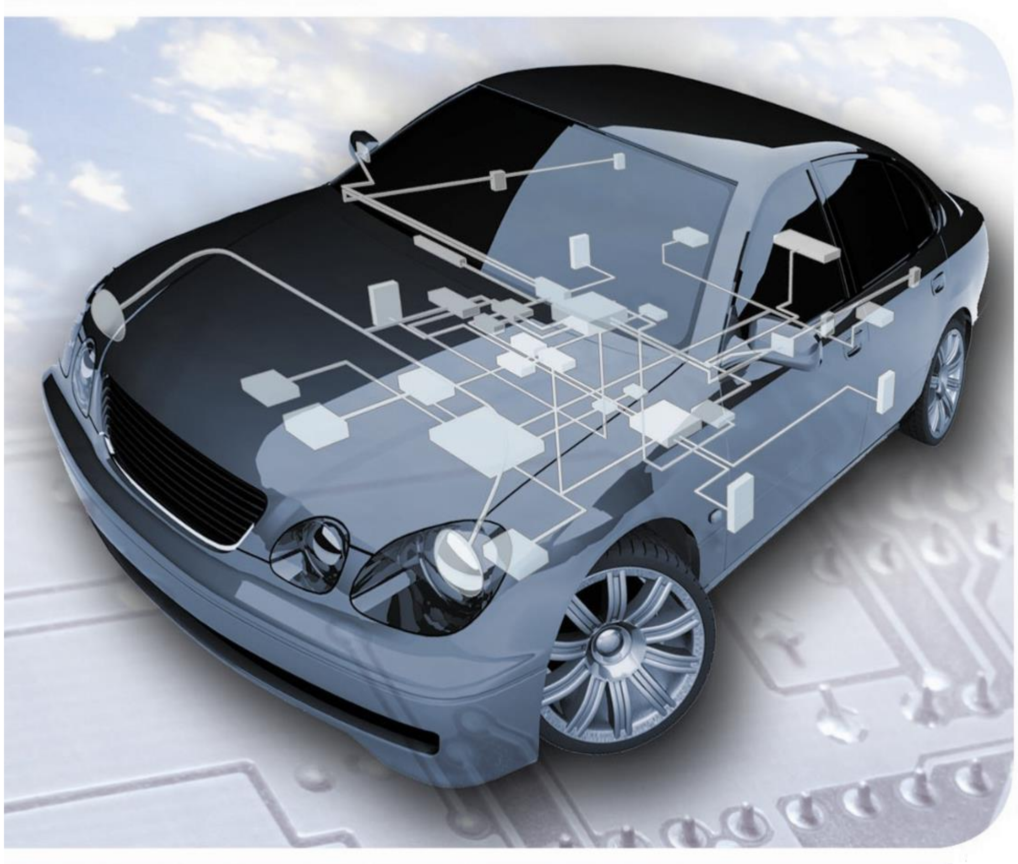
Example: Fliters, Makes and Buffer
PIC18F ECAN™ Module Mode 0

- 3x TX Buffers (Dedicated)
- 2x RX Buffers (Dedicated)
- 1x Message Assembler Buffer
- 6x Full Acceptance Filters
- 2x Full Acceptance Masks



LIN (Local Interconnect Network)

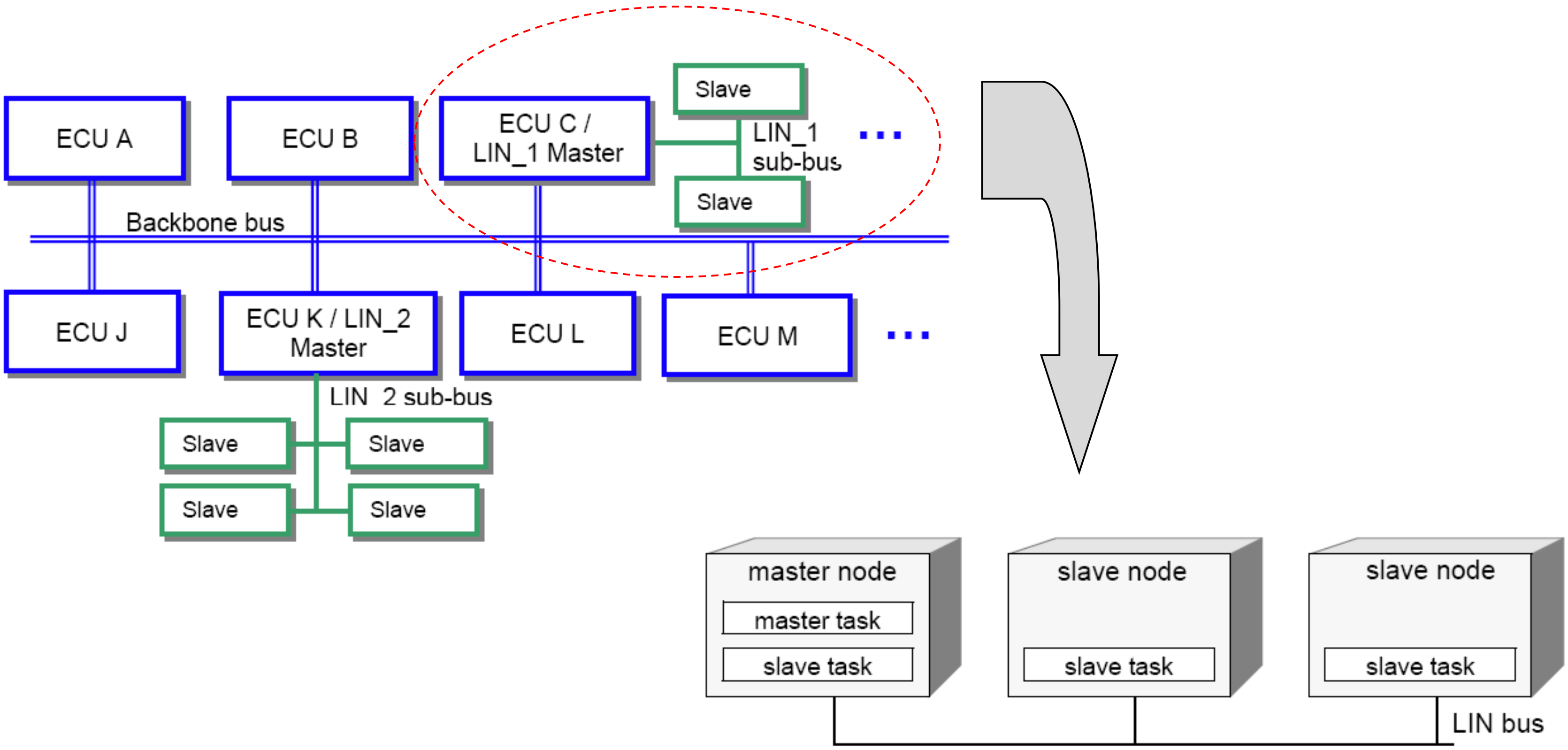
LOCAL INTERCONNECT NETWORK



LIN Overview

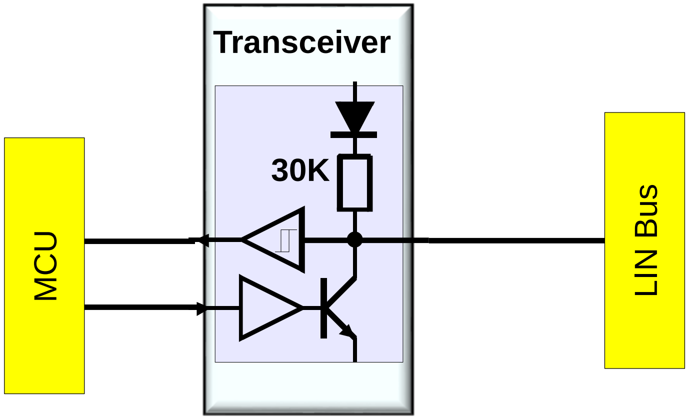
- LIN (Local Interconnect Network) is a low-cost serial communication system for distributed electronic systems in vehicles.
- Used to distribute end point systems of Automotive CAN ECU
- The LIN Bus was defined by major European automobile manufactures like Audi, BMW, DaimlerChrysler, Volvo, VW, VCT and Motorola
- Cost savings between switches, sensors and actuators (sub-systems) that do not require CAN performance and bandwidth
- Low cost silicon implementation based on common UART/SCI interface hardware

Node Concept

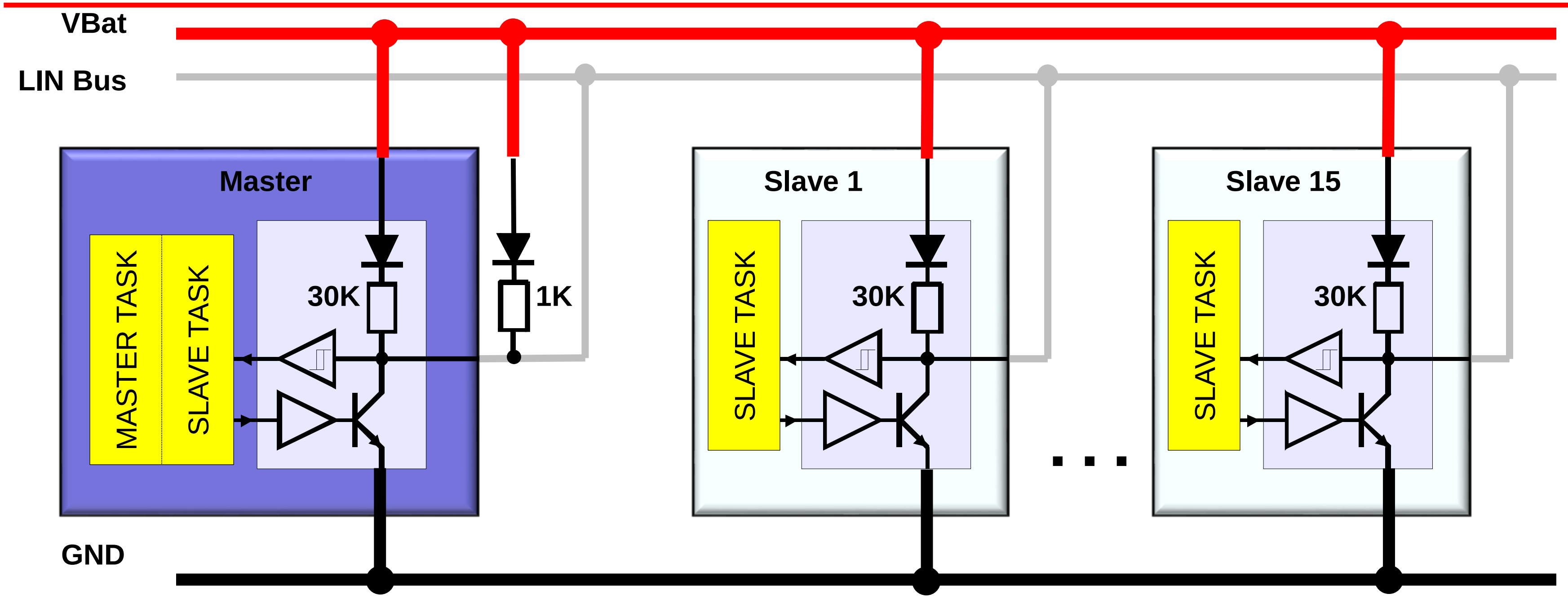


What is a Transceiver?

A transceiver is a device comprising both a transmitter and a receiver which are combined and share common circuitry or a single housing.



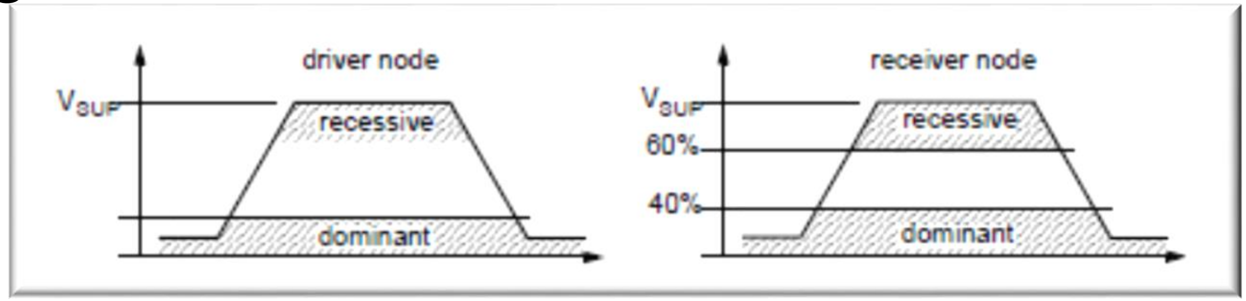
LIN Connectivity
Physical Layer



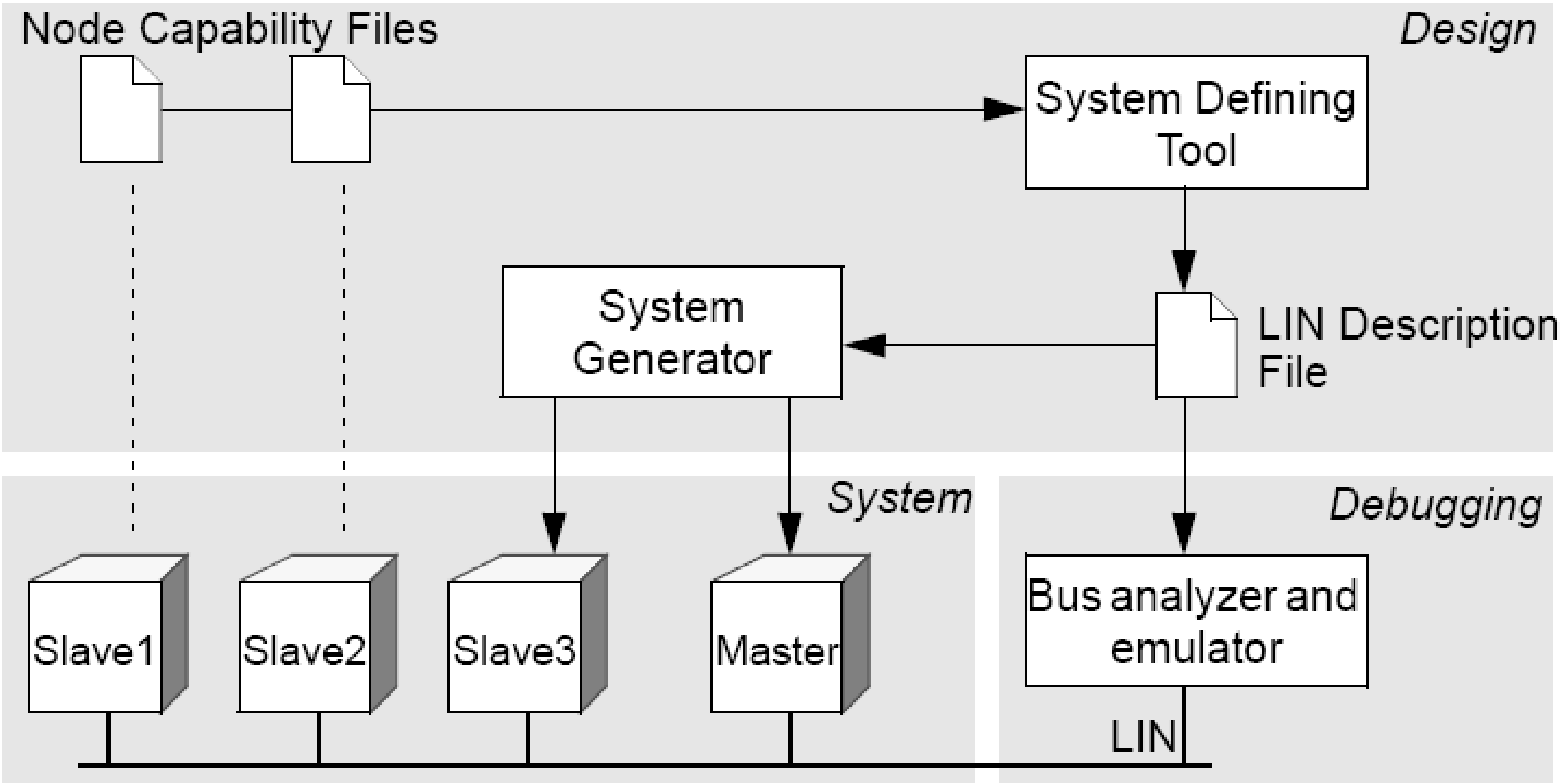
- **Single Master Multiple Slave (up to 15 Slaves)**
- **Single Wire: VBAT, GND, LIN**
- **From 1kbit/s up to 20kbit/s**
  - **10417 biat/s for J2602**
- **Total length of bus line: 40 meters max**
- **Terminations: Master 1k Ω / Slave 30k Ω**

Signal Levels

- **Dominant – Bus LOW – logical 0**
- **Recessive – Bus HIGH – logical 1**
- **Network is Wired AND**
  - All nodes must be HIGH (recessive) in order to transmit a logical 1
  - Only one node LOW (dominant) will transmit a logical 0

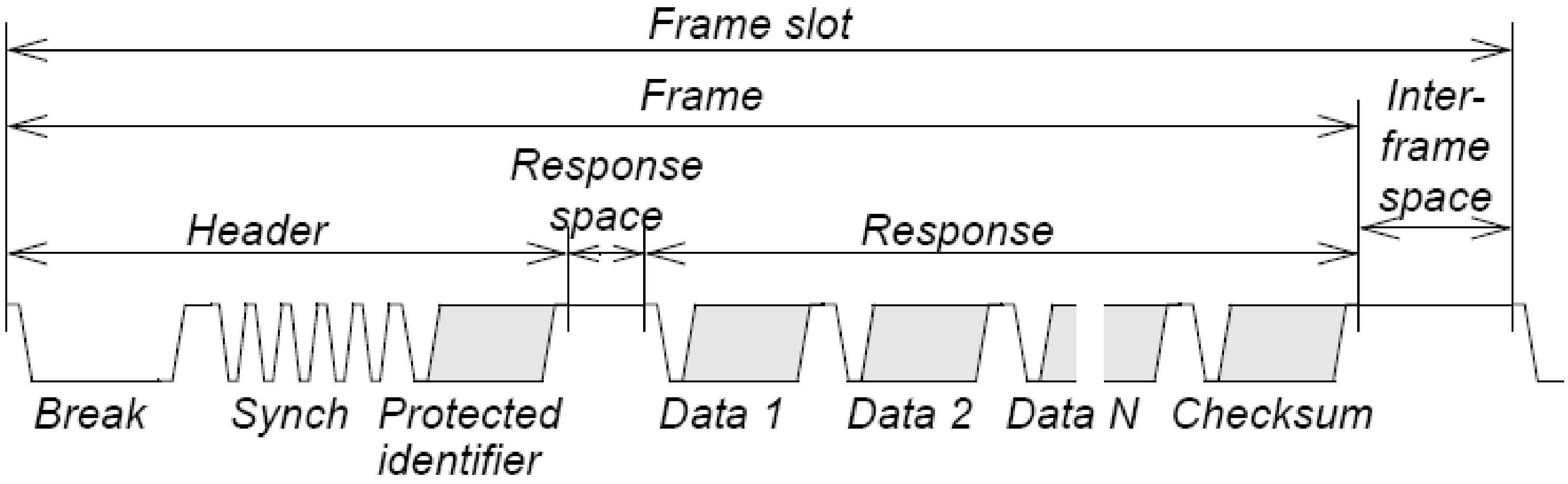


Work Flow Concept



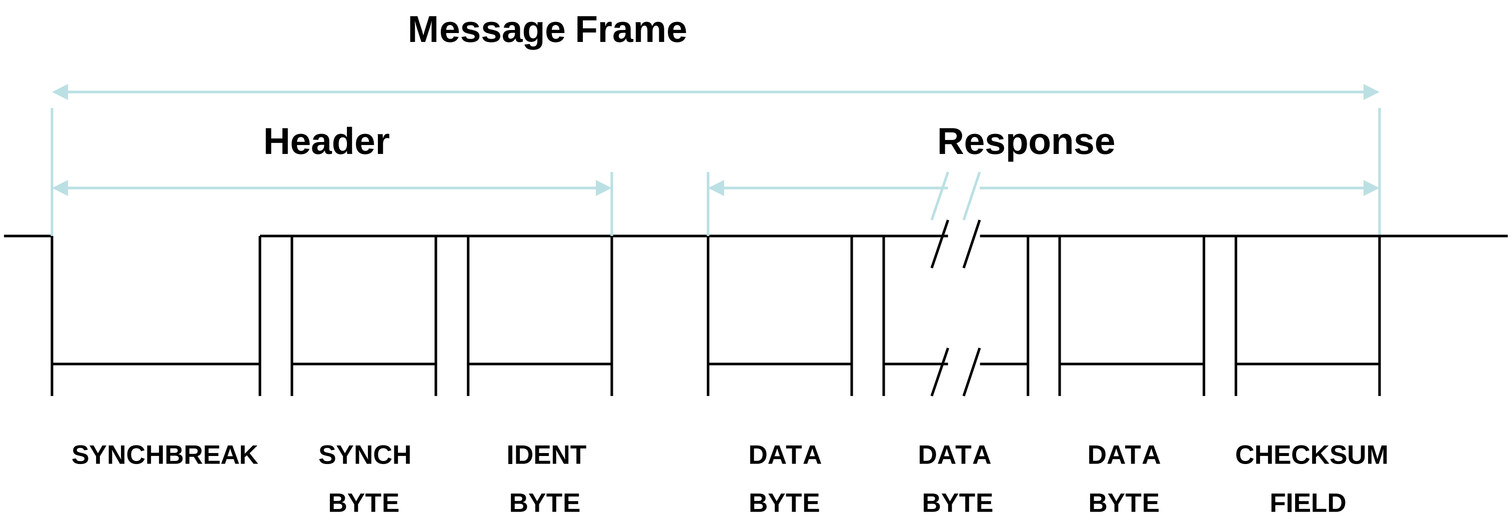
LIN Protocol Overview

- **LIN protocol frame consists of Header and Response as shown below**
- **Header configuration is controlled by Master Node**
- **Master message Header : synch. break, synch. Byte, message identifier**
- **The slave task starts sending a message respond when the ID is correctly identified**
- **Message respond consists of data bytes and checksum byte**

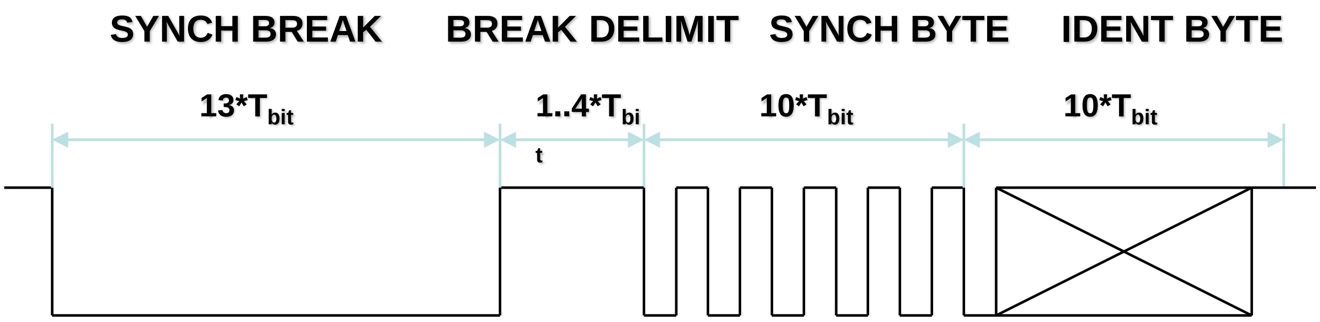


LIN Protocol Overview

- **Message frame: Header and Response**
- **Header: synch. break, synch. byte, message identifier**
- **Respond: Up to 8 data byte and checksum byte**
- **First output bit is LSB**



LIN Protocol Overview



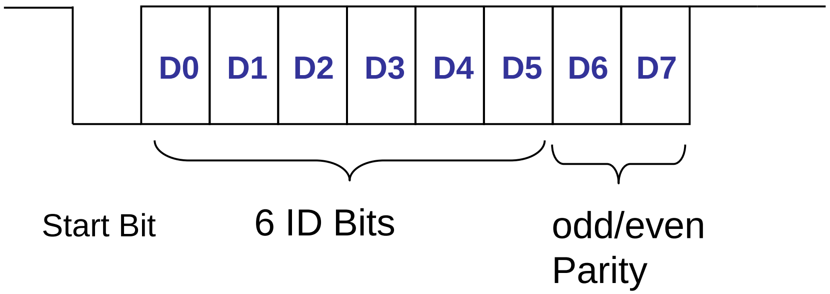
- **SYNCH BREAK**: **low phase(dominant) > databyte length to identify message header**
  New Frame Start

- **BREAK DELIMIT**: **Maintain at least 1 bit of nominal bit**

- **SYNCH BYTE**: **Needed for auto-baud calculation**
  Data field with data value of 0x55

- **IDENT FIELD**: **Parity (2 Bit), ID (6 Bit)**

LIN Protocol Overview



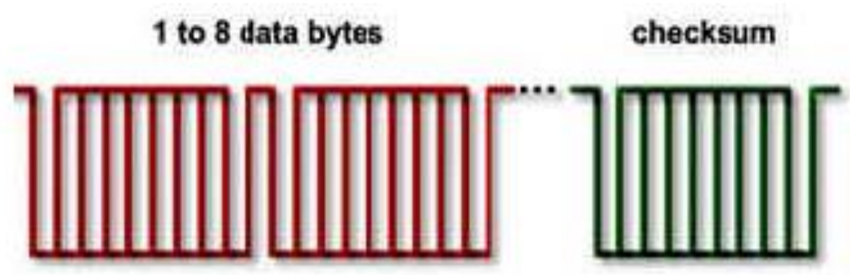
[Identifier]

- **63 IDs using six bits**
- **Frame type**
  - 0 to 59 (0x3b): general frame
  - 60 (0x3c) and 61 (0x3d): diagnostic frame
  - 62 (0x3e) and 63 (0x3f): reserved frame

[Parity (2bits)]

- even (bit6) = D0 xor D1 xor D2 xor D4
- odd (bit7) = ! (D1 xor D3 xor D4 xor D5)

LIN Protocol Overview

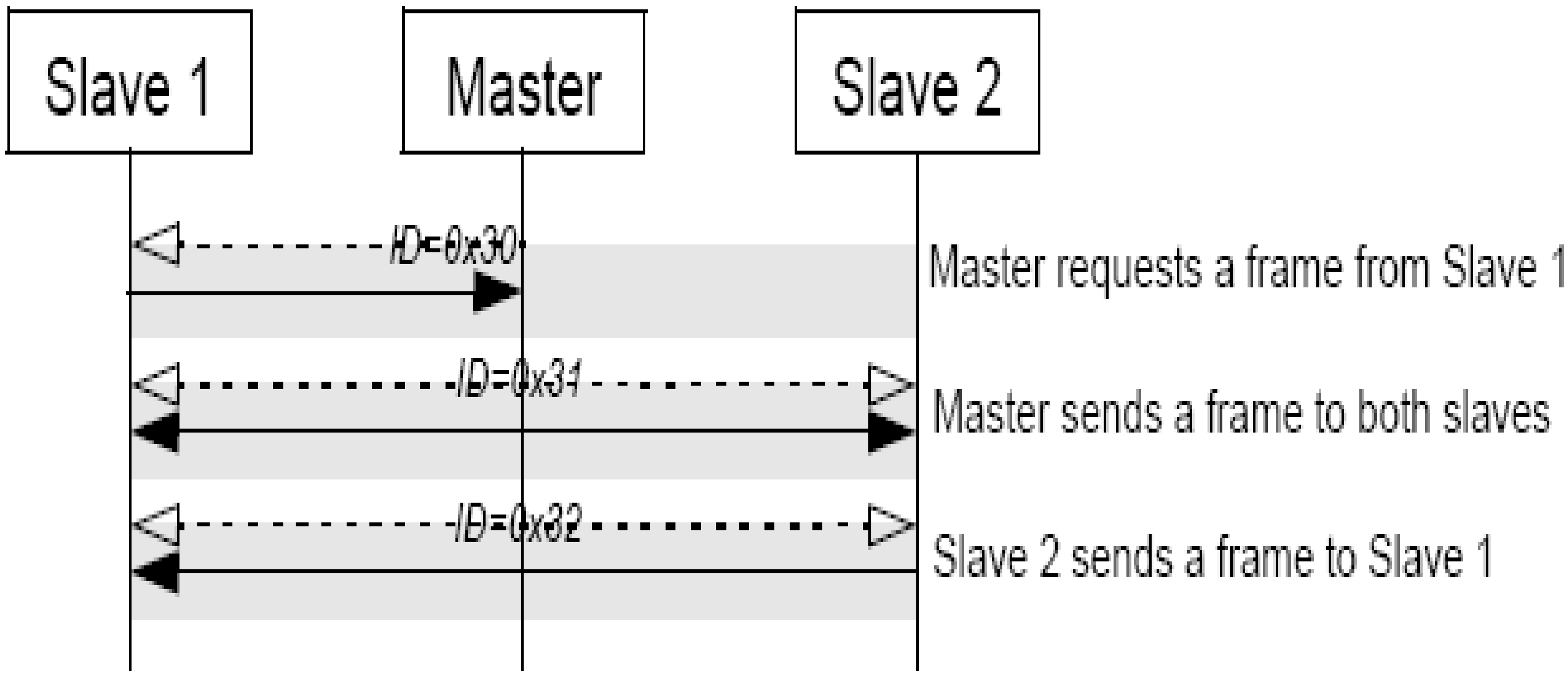


- **Data field**
  - **Up to 8 bytes of data transfer**

- **Checksum**
  - **The last field of the frame**
  - **LIN 1.X: Classic Checksum - Data Bytes Only**
  - **LIN 2.X: Enhanced Checksum - Data Bytes + Protected ID (from Master Header)**

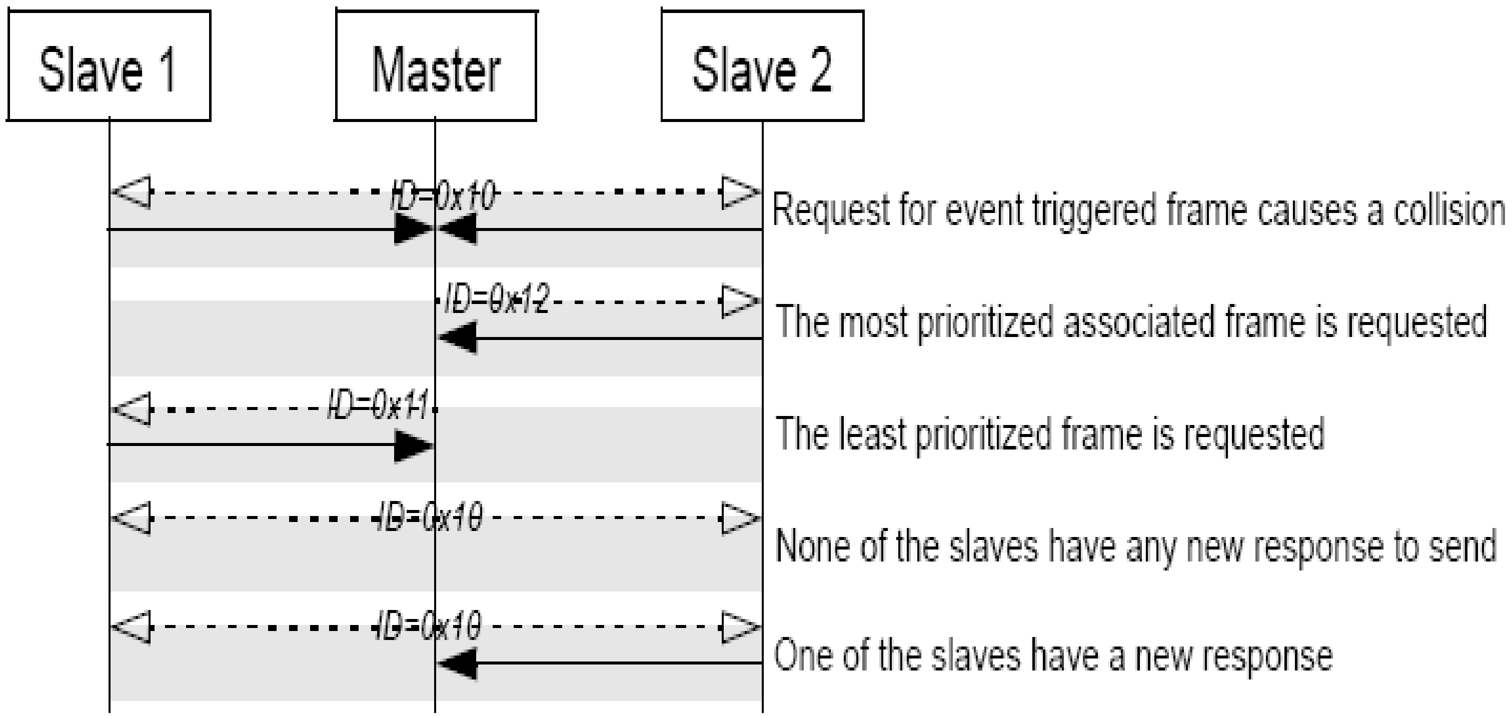
LIN Protocol Overview

Unconditional Frame



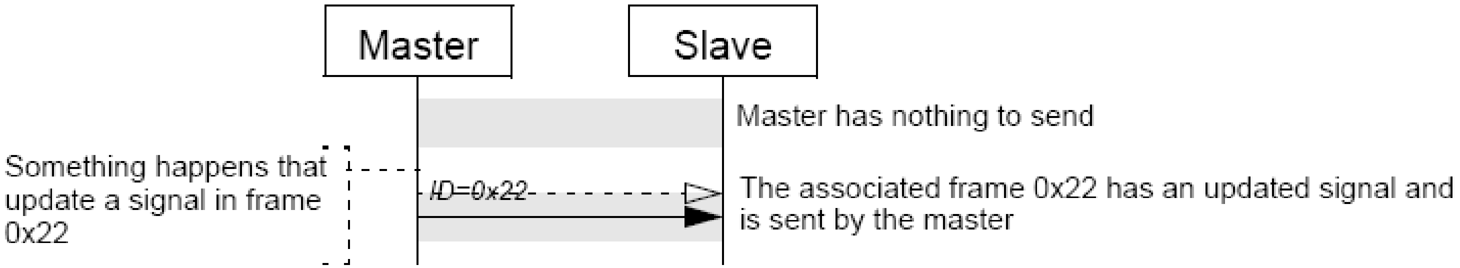
LIN Protocol Overview

Event Triggered Frame



LIN Protocol Overview

- **Sporadic Frames**



- **Diagnostic Frames (0x3c)**
- **User-defined Frames (0x3e)**
- **Reserved Frames (0x3f)**

Thank you!