check the bookmark of CAN for referense :-

CANopen - trademark of CAN automation

(1) Introduction to CAN :-

→ Developed by BOSCH
- multi master
- msg broadcast system
- max signalling rate of 1 mbps (imp)

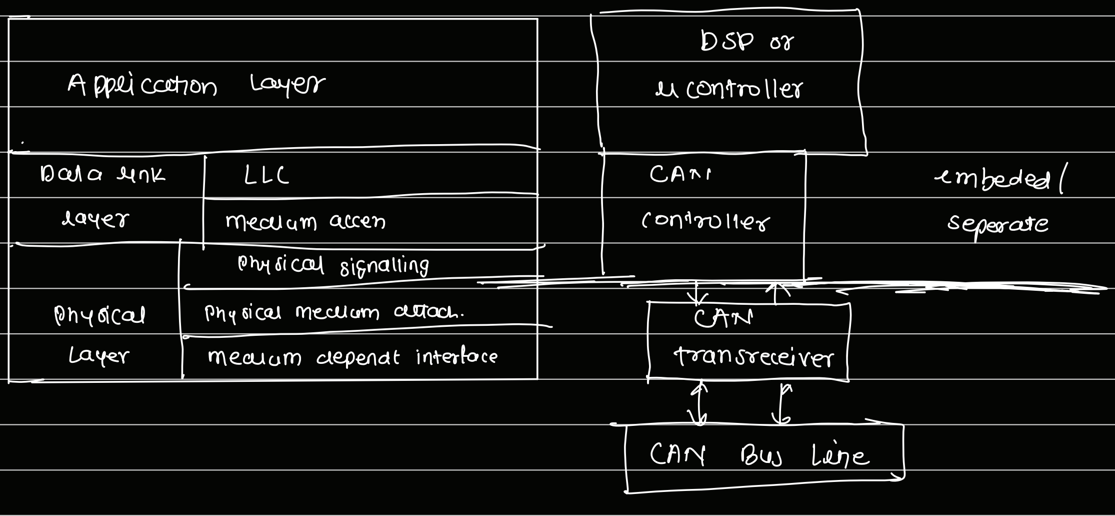
→ CAN doesnt transmit large chunk of data like USB or Ethernet from node A to B
Instead it broadcasts short msgs like temperature, RPM to entire network

(2) ISO Specification for CAN : (ISO 11898 : 2003

→ CAN is ISO standardised $comm^n$ protocol for automotive industry to replace complex wiring harness with two wire bus

→ also used in building automation, medical and manufacturing

ISO 11898 : 2003 explains how information is transmitted between devices on network and that too by adhering to OSI model



[ISO 11898 Architecture]

(3) Standard CAN / Extended CAN

- carrier sense, multiple access with collision detection/ arbitration on message priority (CSMA/CD + AMP)

↖ (arbitration on msg priority)

wait before sending

CSMA => Each node on bus either master/ slave must wait for prescribed period of inactivity before attempting to send a msg

priority through msg id

CD + AMP => collision resolved through bitwise arbitration based on predefined priority of each msg in ID field of sent msg

ISO 11898:2003 standard, with 11 bit ID, provides signalling rate from 125 kbps to 1 mbps

Standard : 11 bit identifier ($2^{11} \cong 2048$ ID's)

Extended : 29 bit identifier ($2^{29} \cong 537$ million ID's)

(3.1) : Bit field of standard CAN :-

Standard CAN

S	11 bit	R	I		D			E	I
O	Identifier	T	D	ro	L	0 .... 8 Byte	CRC	ACK	O	F
F		R	E		C	Data		F	S

SOF : Single dominant SOF (start of frame) bit. Used to sync the nodes on bus after being idle

ID : Priority - Lower binary value ≙ Higher priority

RTR : Remote transmittion Request (RTR) bit is dominant when info is required from another node, All nodes receives this request. but ID determines specific nodes. The responding data is also received by all the nodes and used by ones interested

IDE : Single Identifier Extension bit   0 standard CAN

1 extended CAN

r0 : Reserved

DLC : 4 bit Data length code (No of bytes of data being transmitted

Data : 0:8 bytes of application data

CRC : 16 bit (15 bit + delimiter), checksum of application data for error detection.

ACK : every node receiving accurate msg's overwrites this bit in original msg with a dominant bit
if fails receiving node make it low discards the msg asking sender to send again

EOF : 7 bit field marks end of CAN Frame

imp when 8 bits of same logic level, one opposite is stuffed

IFS : Inter frame space (7 bit) to allow controller to move to another frame

Extended CAN :

S	11 bit	S	I	18 bit	R				0:8	C	A	E	I
O	ID	R	D	ID	T	r1	r0	DLC	byte	R	C	O	F
F		R	E		R				Data	C	K	F	S

substitute remote request ???

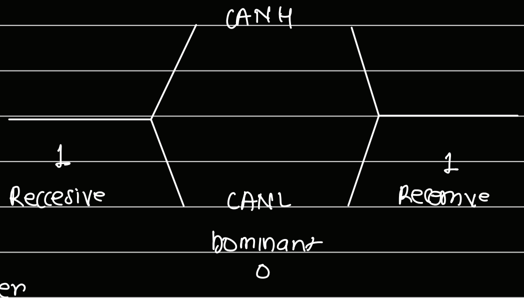
Indicates more ID bits

imp

Fundamental CAN characteristics :

In CAN, logic high is recessive
logic low is dominant

Thats why in many CAN transreceiver
wired are passively pulled high internally so that it 1 when not doing anything



Different message types / Frames

Data Frame

Remote Frame
No data

Error Frame
Frame indicating violation of CAN formatting

Overload Frame
transmitted by node that

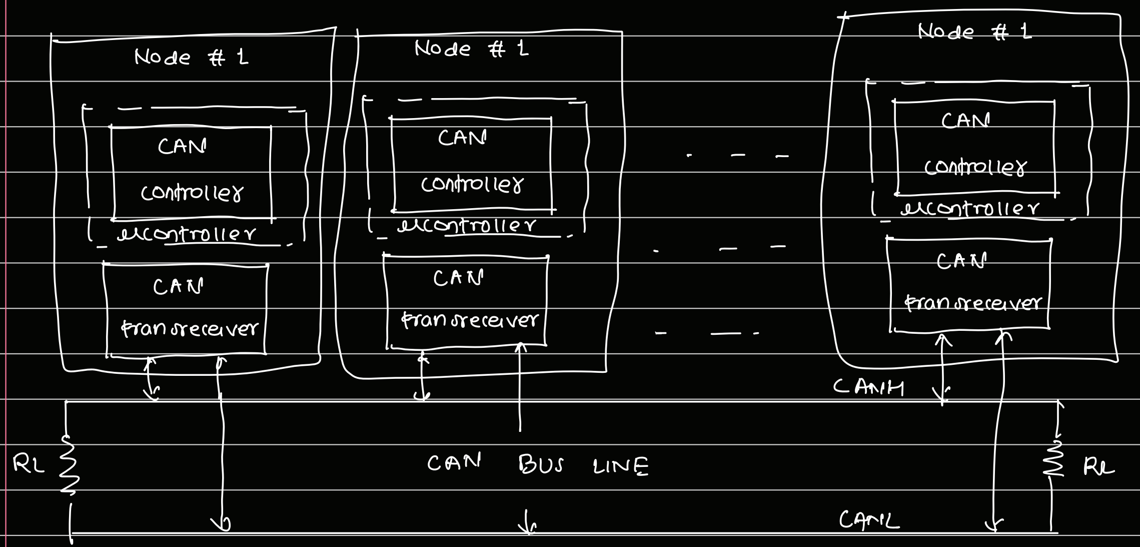
becomes too easy

imp

Q] What is a valid Frame & how to determine if its valid or not ?

By last bit of EOF field of Frame

- if last bit is error free receive state (valid frame)
- —— " —— is dominant state (Repeat)



* Hot Plugging   Nodes can be added even when n/w is operating