CAN (Protocal)

Controller Area Networks

CANoe → Software for programming

eg:- ECU's in CAR

* CAN ISO Number is ISO-11898 → CAN protocal
* it is designed by Robert bosch to exchange data blw Electronics Control units (ECU's) → MC based unit
* Designed for long distance communication.
  eg:- 1km distance & the the speed is 50kbps
* CAN is Message based Protocal → frames

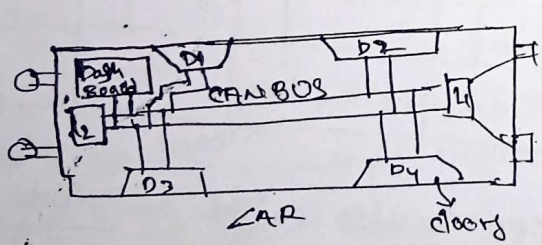
ECU's (Electronics Control units)

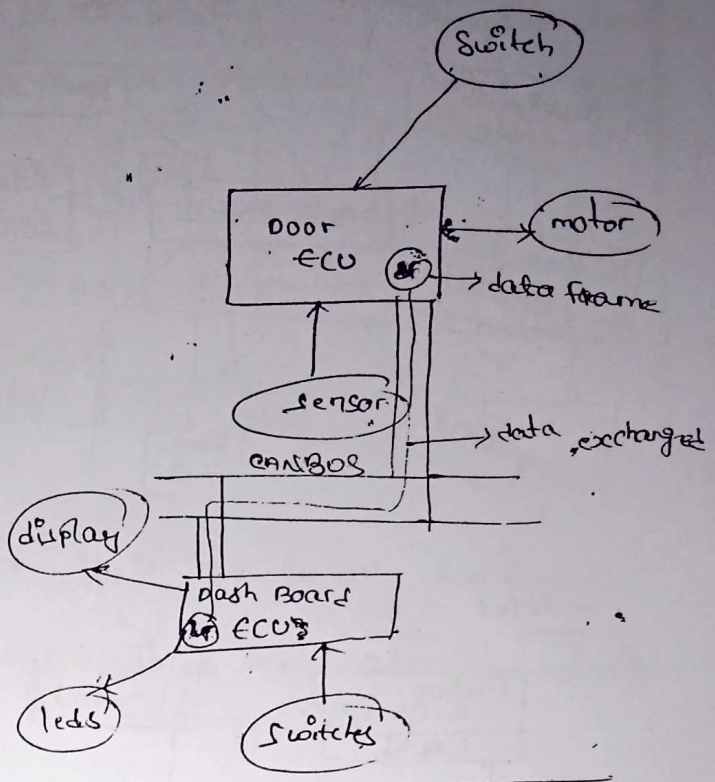
→ It is Micro-control based unit

→ It is used to Control for spectic System

eg:- opening a window in car's

- light control ECU's
- → Steering, → Breaking system
- → Engine control ECU's





Why CAN Protocal

* for reducing the wiring Complexity
* CAN was first created for automotive use, so its most common application is in-Vechicle Electronic Networking

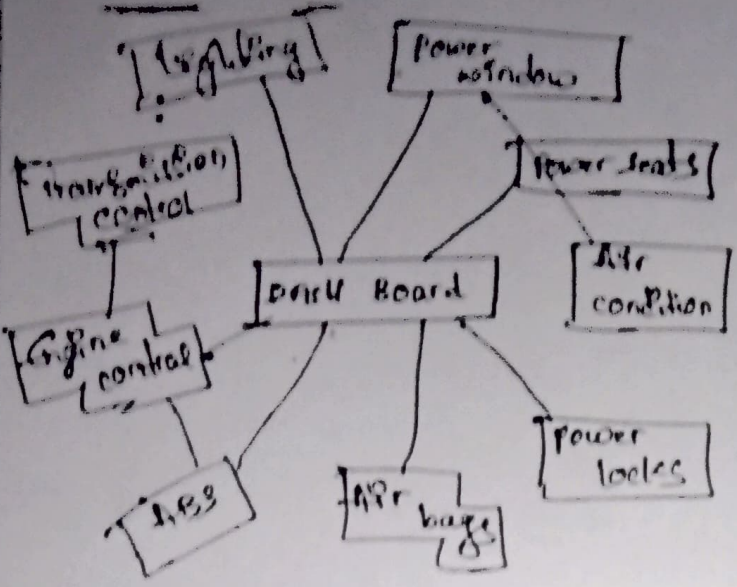
⇒ CAN is also known as In Vechicle Network protocal

* Purpose to replace Complex wiring harness with two-wires-bus

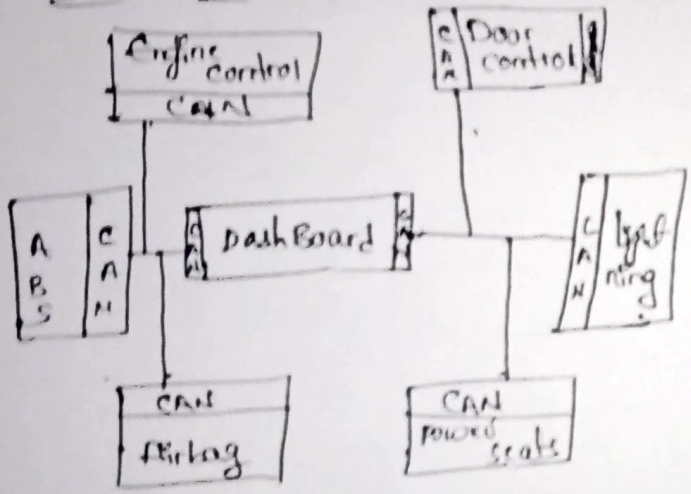
* CAN is a serial Communication technology used especially for reliable data exchange between ECUs in the automobile without using Any Host Computer

* An idea initiated by Robert Bosch GMbH in 1983 to improve the quality of automobiles there by making them more reliable, Safe and fuel efficient

Before CAN



After CAN



Diagnosting

checking the errors in the tools (scanners)

* this type technologies is called on-board Diagnostic (OBD) tech

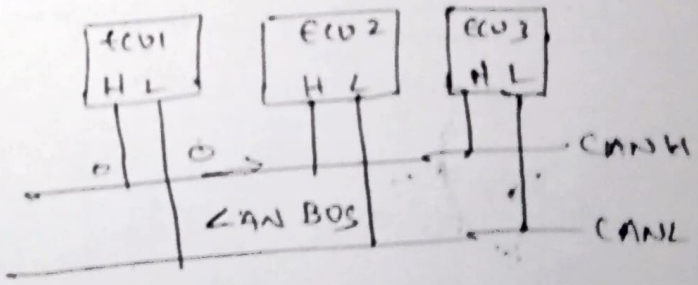
* OBD is vehicles self reporting capability

* for OBD, UDS protocol is used in automotive industry (unified Diagnostic Services)

OBD

* CAN is one of the five protocols used in the on-board diagnostics (OBD) vehicle diagnostic standard

* It is two wired protocol
  -> CAN H (CAN-High)
  -> CAN L (CAN Low)



* ECU is also called as nodes

* these wires are used to send data in one direction

* It is a differential voltage method

* It is Half Duplex.

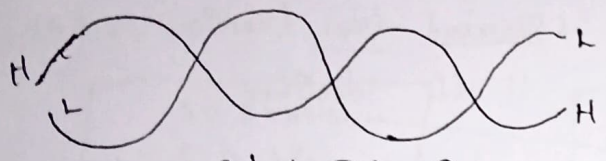
* Differential sig method is less sensitive to external noise

* Why UART not use in cars instead of CAN protocal?

* Because of single wire Method, it is more sensitive than the CAN protocal. (more noise like sound)

---

* In hard-ware straight line Cable, only used twisted pair cable.



twisted Pair Cable

Because of using this twisted Pair Cable reason is reducing electronic magnetic interference (low)

---

3. CAN Protocal is :-

- Broad-cast
- Multi-master (Priority based)
- Half Duplex
- Asynchronous Serial Communication protocal

Every ECU is multi master.

there is, two separate Tx & Rx pin two wires send data in one direction

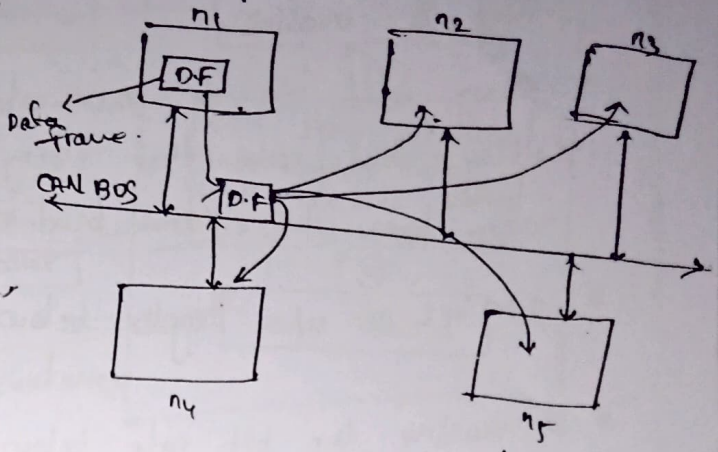
* Every ECU's has one master, so, it is multi-master

* This is no common clock line b/w the wires (Tx & Rx)

* To one Master's not depends one-another

① Broad-cast

* Broadcasting is in built feature of CAN protocal



* It is one to all functions

* Unicast & multi-cast Communication is also possible.

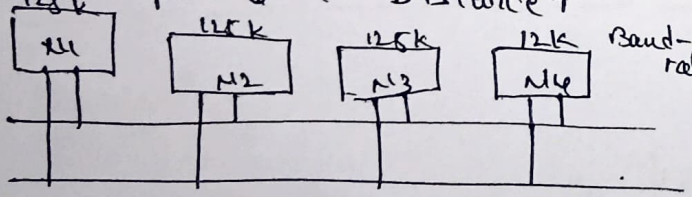
* Message Acceptance Filter Setting Rx nodes

4. CAN Speed & Distance

* In CAN protocal also Baud-rate setting is avaliable.
* Every node to do baud-rate setting but note that every node have the same baud-rate
* 1Mbps at n/w length below 40 meters
* Decreasing the bit rate below, longer network distance.
* In any-one of Node is have diff baud-rate, then that node will not syncronous with the another Nodes

Eg:- 500meters at 125k bps (Kilobit per second).

* Speed ↑ ↔ Distance ↓
* Speed ↓ ↔ Distance ↑



Bus Length Vs Signalling Rate

Bus length (Meters)	signaling Rate (Mbps)
40	1
100	0.5
200	0.25
500	0.10
1000	0.05

5. MAN NODES on CAN

→ Practically up to 30 NODES

(when more than the 30 nodes are used on a bus, it is recommanded that a transceiver with a high bus input impedednce). → (increases (n/w) (bus vtg)

6. CAN is available in two Versions

→ Standard CAN (version 2.0A)
11 bit identifier

→ Extended CAN (version 2.0B)
29 bit identifier

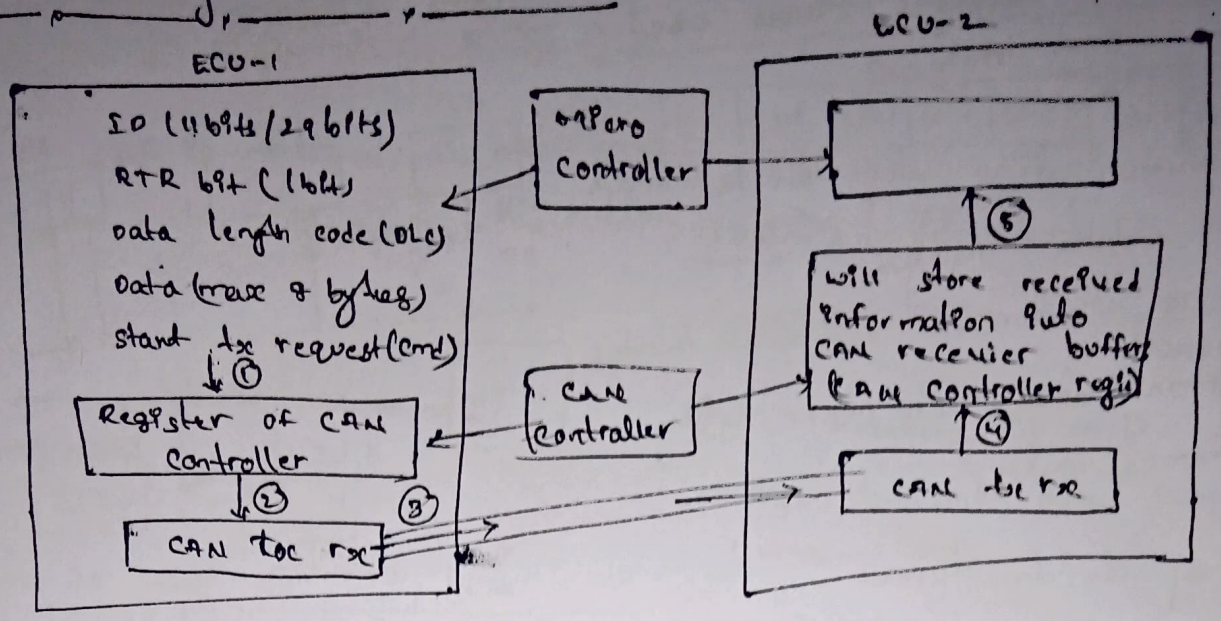
* In the CAN protocal has lot of information, it has data and Identifier (it represents a type of data frame or message)

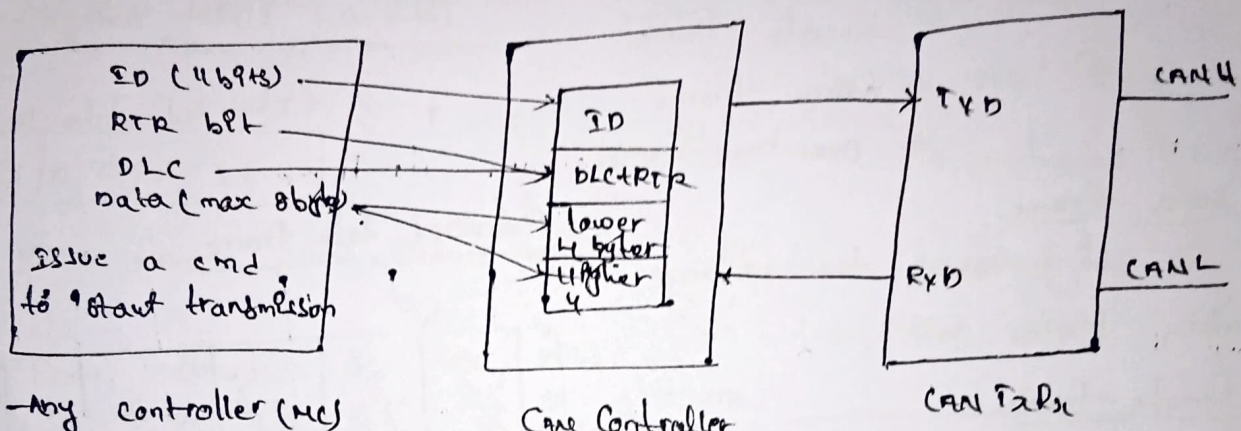
Data frame means:- Data and extra information

7. To achive Design transparency and implementation flexibility CAN has been sub-divided into different layers.

- Data link layer (ISO 11898-1)
- physical layer (ISO 11898-2&3)

Understanding data-flow in the ECU's



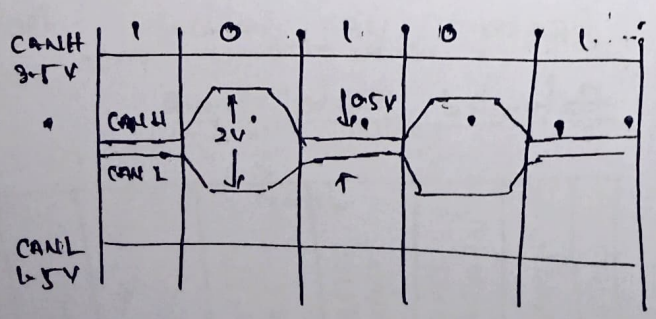


* while giving a cmd to CAN controller it will create the data frame on the CAN Controller.

logic 0: $V_{diff} > 0.9V$ } CAN vtg levels

logic 1: $V_{diff} < 0.5V$ }

$V_{diff} = CANH - CANL$



* logic 0 (Dominant bit) - Differential vtg (V_{diff}) is greater than 0.9V.

* logic 1 (Recessive bit) Differential vtg (V_{diff}) is less than 0.5V.

List of messages / frames used in CAN protocol

- Communicating with ECUs through the data frames.
- * In the CAN n/w ECU's / Nodes Communicates each other using frames only.

They are four types:-

1. Data frame
2. Remote frame
3. Error frame
4. Over load frame.

Data frame

It is used to exchange data b/w's ECU's

Remote frame

It is used to request a data frame from other ECU's

Error frame

It is used to inform error in the n/w.

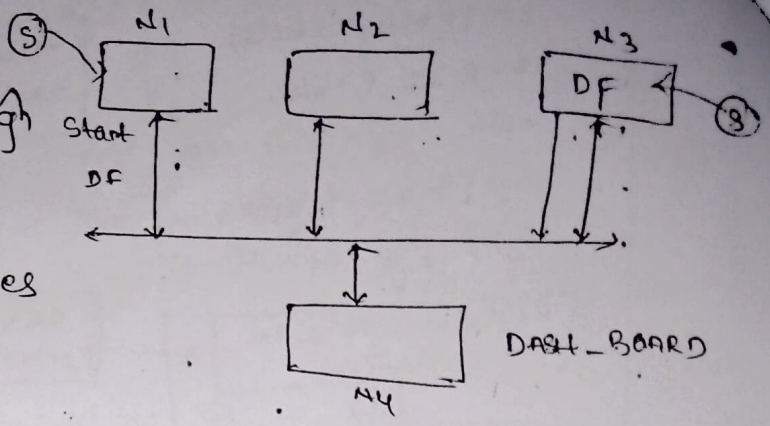
Note:- By default it is board cast frame.

Over load frame

It is generated whenever a node is busy & other node trying to start communication

Note:- This frame is rarely generated by CAN controller

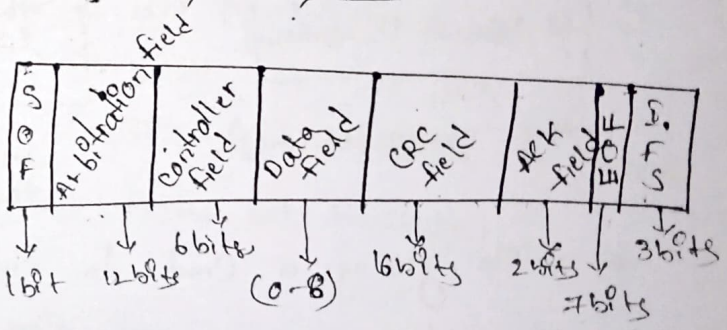
DATA FRAMES



They are two types of data frames 1. Standard Data (U2.CA) 11 bits

2. Extended data (U2.CB) 29 bits

Standard data frame



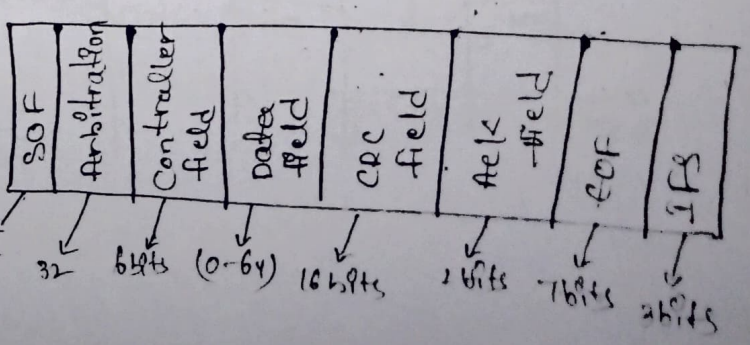
SOF :- Start of frame

EOF :- End of frame

CRC :- Cyclic redundancy check

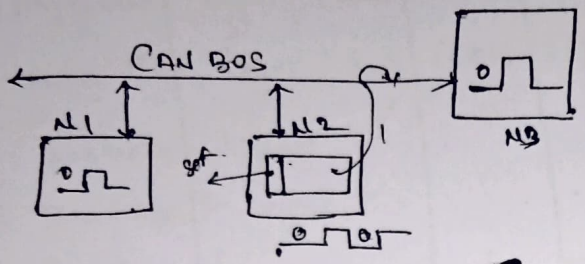
IFS :- Inter-framer Space

Extended 'Data' frame

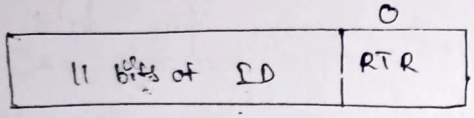


Start of frame (SOF)

* It is a single dominant bit (zero)
* If SOF is detected, all other nodes in the n/w synchronizes their clk.



Arbitration field (12 bits)



RTR – Remote transmission request

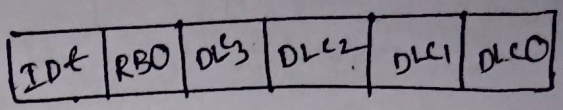
* Data direction is MSB → LSB.
* In the CAN protocol arbitration field in arbitration field only.

RTR:- Remote transmission Request

Identifier represents type of message, light system message, crash detection, ABS messages etc....

RTR → 0 (data frame)
→ 1 (remote frame)

Control field (6 bits)



IDE:- Identifier extension (always dominant in STD data frame)

RB0:- Reserved bit no 0 (always dominant data frame) (MSB → LSB)

DLC Data length code

IDE:- → 0 in STD frame
→ 1 in extended frame.

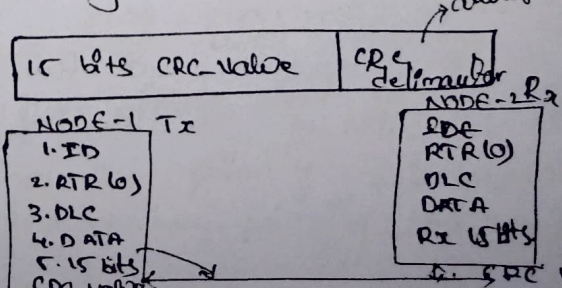
Represents no of bytes of data transmitted within a data frame

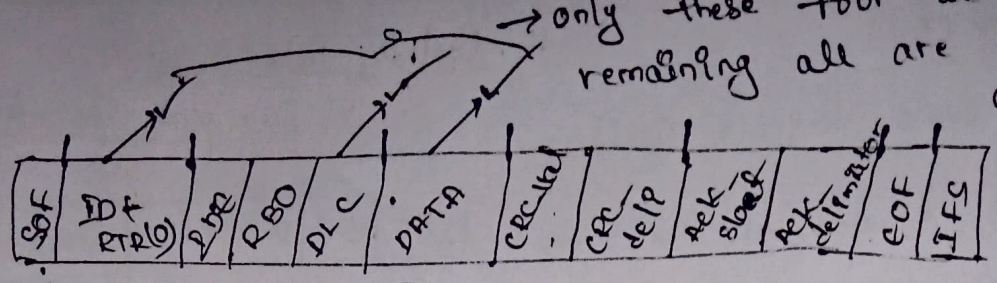
Data Field

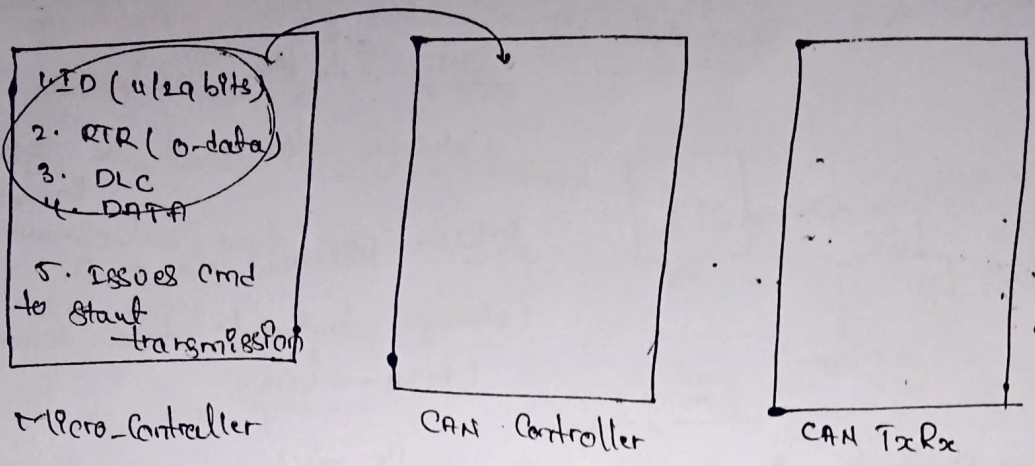
This field consists of data to be transmitted with in a data frame

CRC (16-bits)

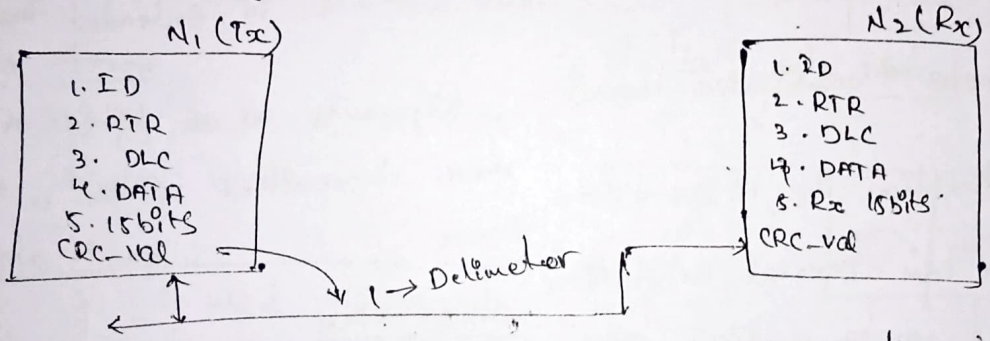
checking is used for error checking CRC delimiter







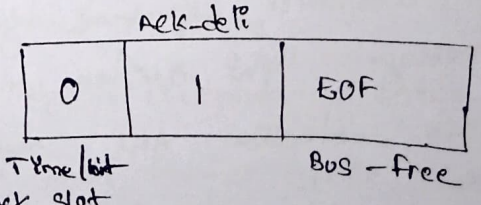
→ there is NO-transmission & Receiver node in the CAN, every node can transmite & Receive the data.



→ the tx node wait for fixed period of time i.e., time / bit (1ms)

→ the waiting period is called as Ack_slot

→ If generatic CRC value is matched with Rx CRC_value then Receiver node generate 0 (Ack_slot)



End of frame (7 bits)

* It is a flag sequence of 7 recessive contiguous bits.
* Tx node, write EOF field to free CAN bus or to finish the transmission

IFS (Inter-frame Space)

* It is sequence of 3 recessive contiguous bits.
* This field is used to provide some delay b/w prev & Next frame.

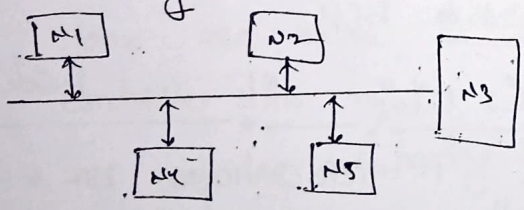
* ISO 11898-2 (High speed CAN) baud >= 125kbps
* ISO 11898-3 (low speed CAN) (baud < 125 kbps)

7 layers of OSI Model

1. physical layer (ISO-11898-1).
2. Data link layer (ISO 11898-2/3)
3. Network layer
4. Transport layer
5. Session layer
6. presentation layer
7. Application layer

8. CAN standard supports several topologies Commonly used topology is line / Bus - topology (methods of connecting computers in nlw that is called topology)

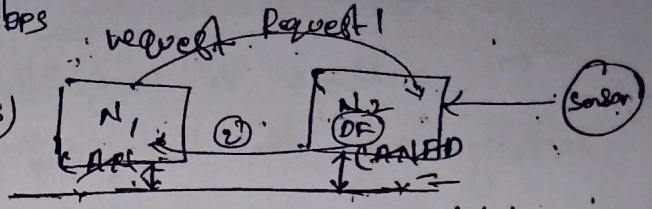
* Commonly used



9. low cost, light weight network.

10. standard CAN protocol supports 8 bytes while CAN FD (flexiable data rate) protocol supports 64 bytes in the data field part.

11. Supports remote data request



12. Supports non-destructive Contention based arbitration (Arbitration is maderitory because it is multi-master).

13. The protocol supports different error detection capabilities such as bit error, ack error, form error, CRC error and stuff error.

14. In CAN protocol any node become transmitter and Receiver

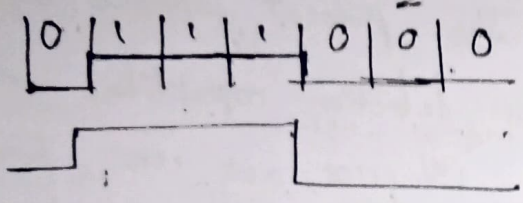
15. Every message has a priority, so, if two nodes try to send messages simultaneously the one with the higher priority gets transmitted and the one with the lower priority gets postponed.

*** In CAN there is node are not have priority Data frame sent by the node has the priority
* Frame has priority (Message)

16. CAN Data -format.

The message is transmitted serially onto the bus using a non-return-to-zero (NRZ) -format.

→ As the name suggests, there is no transition between same polarity bus.



(Manchester formate)

→ there is transion on every bit

* In CAN protocal NRZ formate is used

Applications of CAN Protocol

* Passenger vehicles, trucks, buses (gasoline vechicles and electric vechicles)
* Electronic equipment for aviation and navigation.
* Industrial automation and mechanical control.
* Elevators, escalators
* Building automation (talk back)
* Medical instruments and equipments

Hard-ware Setup of CAN Protocal

Inside ECU

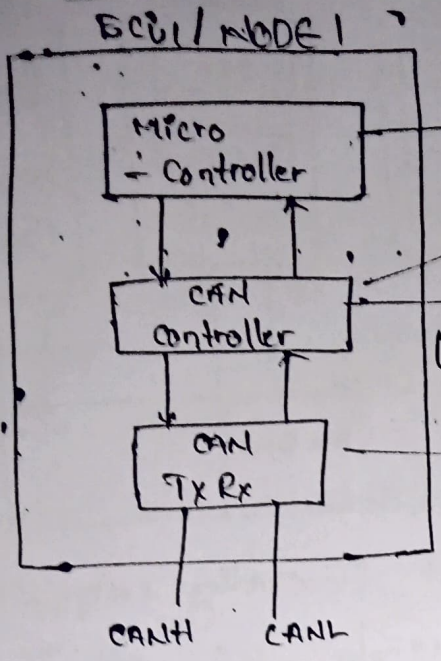
Is Major units used inside ECU

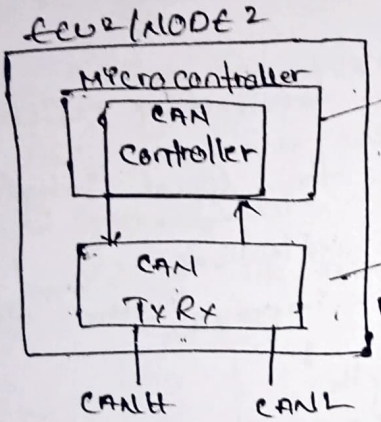
1. Micro Controller
2. CAN Controller
3. CAN Transreceiver

Other units based upon requirement

* LED's / Bulbs
* Motors
* Switchs
* Displays
* Sensors.
* etc.,

Representation of ECU





* Difference b/w (i) & (ii)
  Some MC have in-built CAN controller
* All 8051 MC not having in-built CAN Controller, as well as LPC2148...
  (External CAN Controller MCP2515)

Role of CAN Controller inside the ECU's

1. It is used to create & send messages (data/remote/error/overload frames)
2. Responsible for receiving messages from other nodes
3. error handling
4. re-transmission of lost messages
5. bit stuffing handling
6. bus-off recovery
7. Message acceptance filter setting (which message are frame to be accepted & ignored is decided by this setting)
8. Baud rate setting
9. It can generate interrupt upon receiving and transmitting as well.

ROLE of CAN TxRx

1. It is used to Convert vtg levels while transmitting & receiving messages/frames. It will Convert TTL vtg level to CAN vtg level vice versa.

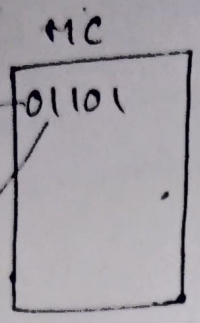
MC vtg ←

2. It is used to protect CAN controller from over voltage.

TTL vtg levels are:

logic 0: 0v to 0.8v

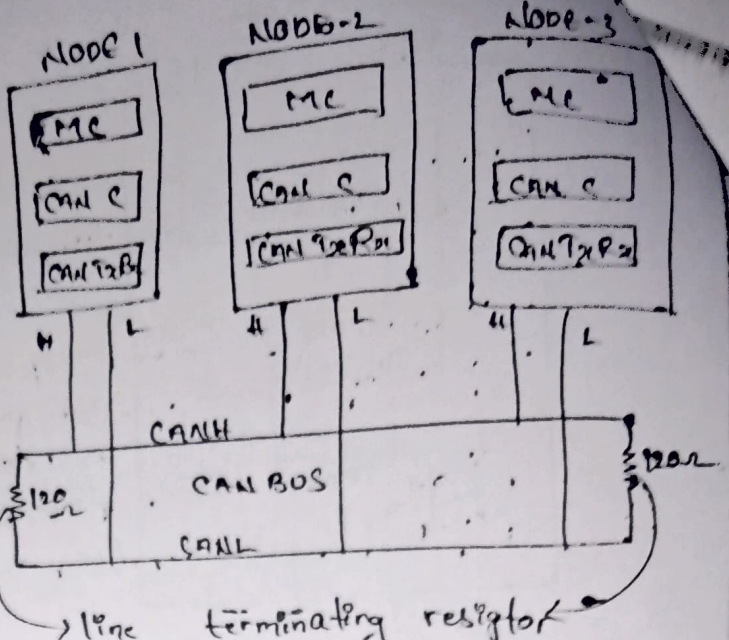
logic 1: 2v to 5.5v



(transmitter logic)

TTL vtg levels == MC vtg levels

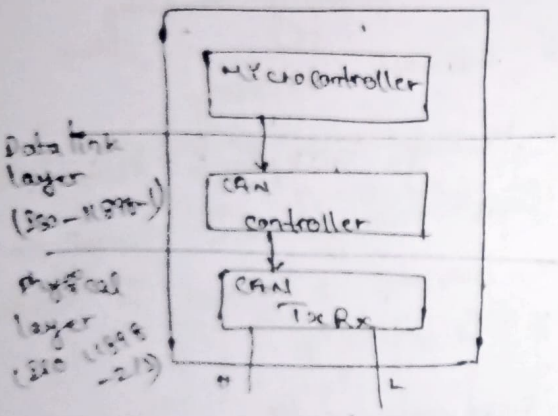
== CAN controllers vtg levels)



→ line terminating resistor

→ for 40meter range we use 120 Ω.

* CAN is following Diff vtg method



* TTL vtg level into CAN vtg is called physical layer

* In which link CAN Controller will work?

→ Data link & physical link (ISO 11898-1) (11898-2/3)

* In which topology

* We are the topologys supported by CAN Controller.

→ Link topology

* line terminatting resistors helps to reduce the signal integrity issues like reflections and improves the signal strength

CAN Bus setup

Idle State of CAN BUS

The idle state of CAN BUS is represented by the recessive level (logical 1).

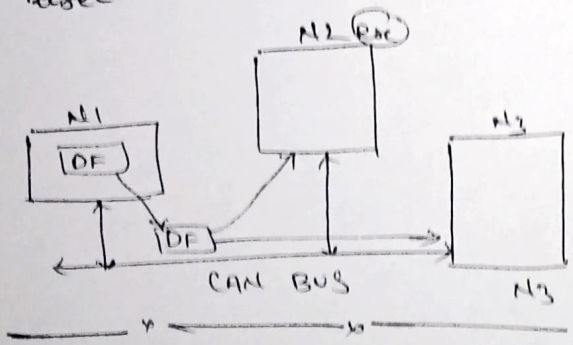
Bit terminologies used in CAN

1. Recessive level / bit
0. Dominant level / bit

→ when any ECU's not sending the data-frame then ECU's not communicating each other is called Idel state of CAN BUS.

Working of CAN Protocol

* Any node in the CAN n/w wants to send data, to another n/w node (or) CAN Bus, they must be create a data frame and write on the bus, that data frame will board. called,
* It is a default feature in the CAN.
* The Receiver node decides the Msg acceptance which has to be ignored data frame based on the identifier (ID)



Role of identifier in CAN Protocol

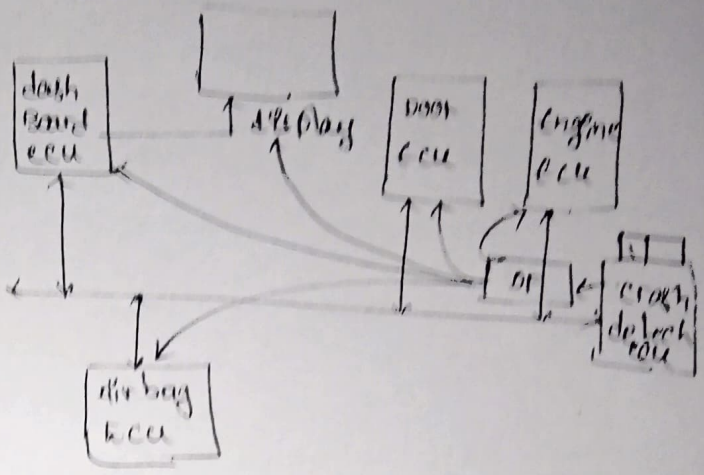
* Represents type of message

Type of message	ID
ABS info	7
Engine temp	56
Door control	32
Wiper control	100
Crash detection	1

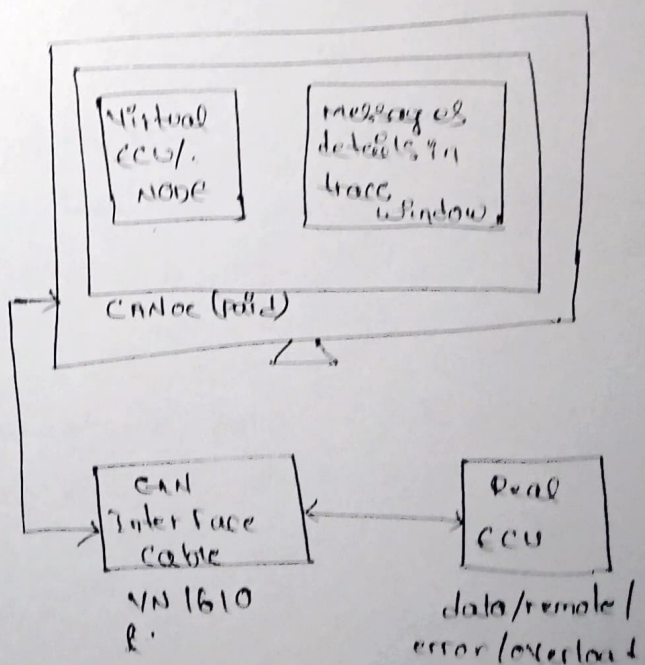
Predefined ID ex: 1, 56, 100, 7, 32

Note:- I.d of data frame represents prio of the data frame

Lower the num of id, higher the prio.



Central lock



Trace window displays received & transmitted frames information or used to display bus traffic

Virtual ECU / Node

We can write a program using CAPL script to do automation with real-ecu's.

* ECU automation (virtually & physically)

How to use CANoe tool?

In CANoe program with **pl.can**

Sending a data frame using interactive generation block (IG

* In the paid version, CANoe can communicate with real ecu not blocks).

In the evaluation version.

*CAN controller also called as CAN channel

CAN Access programming language (CAPL -> kapple)

* It is C-like prog lang.
* CAPL is a scripting lang that is used to access CAN protocol with logical operations.
* with this it is possible to simulate anything on CAN n/w using the script code which is almost like C-lang.
* The CAPL script can be used with Vector CANoe & Vector CAN Analyser tools.

Why CAPL?

* Allows you to quickly develop code.
* The execution of program blocks can be controlled by events (like keyboard, timer, message...)
* Panels for simulations