# * CAN :- ( Controll Area Network)

- CAN (Control Area Network
- Developed by Robert Bosch in 1989.
- It allows microcontrollers and devices to communicate with each other within a vehicle without host computer.
- It is a message-based Protocol.
- Design for automotive applications also use in industrial automation.

- Connect PC to CAN n/w using no. of supported CAN interface h/w.
- XCP protocol for sending and recieving messages
- Filtering CAN messages.
- Generate code from CAN simulink blocks.

* It is broadcast type of bus i.e each node can hear all transmission.
- Here the complexity of node and range from a simple i/p devices sensor & a controller
- The node may also be a gateway allowing a general purpose register to communicate over a USB or ethernet port.
- It is a message based protocol i.e each msg must carry a msg identifier based on with the priority of msg will be decided.
- There is no need of node identification so that we can easily insert & remove the node from the n/w.

- Serial half-duplex Asynchronous Communication
Asynchronous means there is no need of the time clock signal between sender & reciever to synchronize the data.

- CAN is Two wired Communication Protocol each node are connected to each other through two wired bus.
- The wires are twisted pair with 120 Ω ~~cha~~ characteristics impedence.
- one of five protocol used in the onboard diagnostic
* 120 Ω register is used for register for match wire impedence

* Application :-
1. Automotive (passenger vehicles, trucks, buses)
2. Electronic equipment for aviation & Navigation
3. Industrial automation & Mechanical Control.
4. Elevators & escalators.
5. Building automation
6. Medical instruments & equipment
7. Marine, Military Industrial Medical.

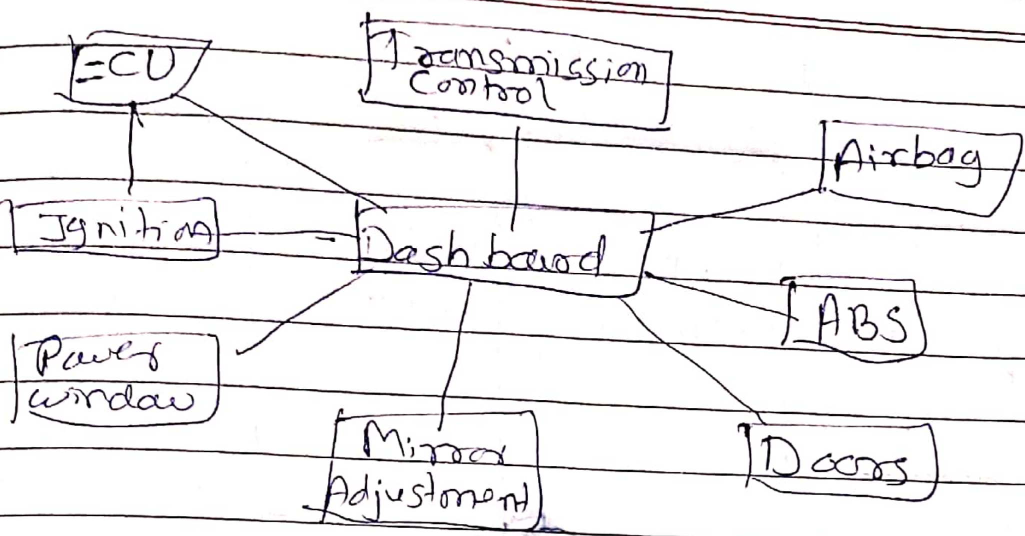
* Max 8 byte can be transmitted in CAN frame
Max data allowed in CAN is 4KB.

* Need of centralized std comm^n ~~proc~~ Protocol.
→ To increasing no. of Electronic Control Unit (ECU)
- In modern vehiche there can be more than 70 ECU for various subsystem

If all node connected to ~~all node th~~ 1 to 1 the speed is of comm^n will be high but n/w complexity, cost of wire is high

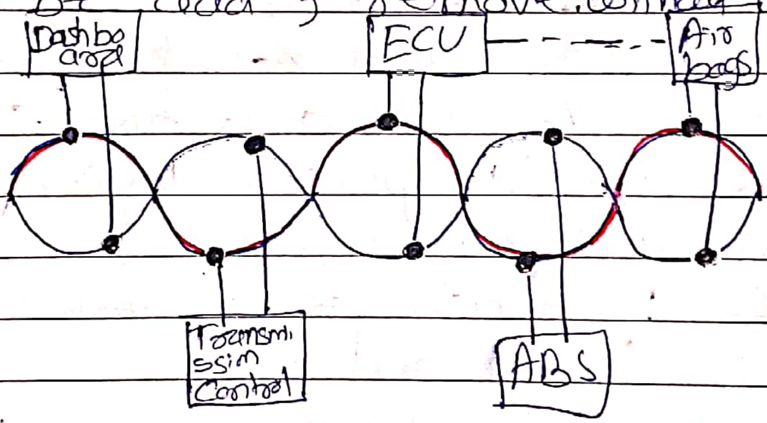
Dashboard need 8 connector

~~Pt~~




To overcome these issues can was introduce as a centralize sol^n which require only 2 wires i.e. CAN_H & CAN_L
can high can low.

- Message prioritization & flexibility as nodes can be add & remove without affecting n/w.




- Centralized - Low cost - flexible - Robust - Efficient

OSI Model -
open System Intersection
It define communication fun^n of each layer.
According to OSI model, CAN defines 2 layer i.e. data link & physical

| |
|---|
| Application |
| Presentation |
| Session |
| Transport |
| Network |
| Data link |
| Physical |

* CAN layered Architecture :-

| ISO/OSI model | CAN in the ISO/OSI model | Standards | implementation |
|---|---|---|---|
| Data Link: LLC, MAC | CAN Protocol | ISO 11898-1 | CAN controller |
| Physical: PLS, PMA, PMS, MDI | CAN Physical layer | ISO 11898-2<br>ISO 11898-3<br>CiA DS-102 | CAN Tx<br>CAN Bus medium<br>CAN Bus connector |

Data link layer is responsible for node to node transmission. It defines the protocol to establish the communication & also responsible for detecting & possibly correcting error that may occure due to physical layer.

it is divided in 2 layer.

MAC :- which devices ~~to~~ in the n/w gains access to the medium

LLC - logical link control is responsible for ~~frame acc~~ acceptance filtering, overload notification and recovery management.

Physical layer is responsible for transmission & reception of raw data, it define the specification for the parameter for voltage level, timing, data rate & ~~collector~~ connector

MAC - Medium Access Control
- Medium Access
- Encapsulation / Decapsulation
- Error Detection
- Signaling
- Signaling.

LLC = Logic link Control
- frame Acceptance filtering
- Overload Notification
- Recovery management.

Physical layer Signaling = PLS
Physical Medium Attachment = PMA
Physical medium specification = PMS
Medium Dependant Interface = MDI

- CAN specification define CAN protocol & Physical layer
- which are defined CAN ISO in CAN standard ISO 11898.

ISO 11898 = 3 parts
↳ ISO 11898-1 $\xrightarrow{\text{covers}}$ specification data link layer & physical signaling

ISO 11898-2 } covers the CAN physical layer for high
ISO 11898-3 } & lowspeed can respectively.

11898-3 → 125 kbps — lowest speed can allows datarate upto 125kbps
— is used where the commⁿ is not critical factor

11898-2 → 1MBPS — A High speed can allow data rate upto 1 MBPS which is primarily used in powertrain & chances area of vehicle.

AND Logic is used at the time of transmission.

CiA – CAN in Automation
defines the specification of CAN connector

It medium } implemented in ~~comp~~ using Window OS

release in 1991

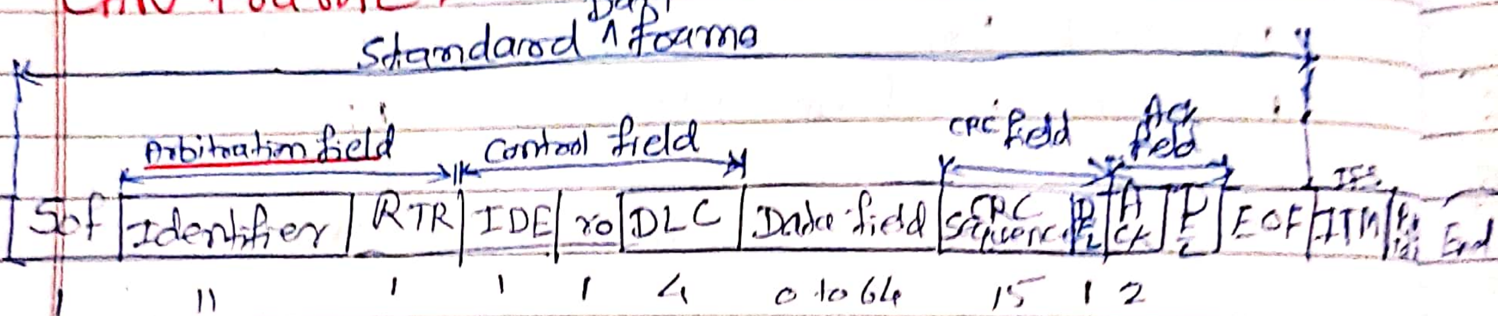
CAN versions:-

CAN 2.0

CAN 2.0A
= 11 bit identifier
Std format

CAN 2.0 B
29 bit identifier
extended format

– CAN frame :-




Arbitration field :- (message identifier)
Control field :- Defines the users defined function
Data field :- where actual data can be capt.
CRC field :- Cyclick Redundancy Check for Error (Data corruption) detection
ACK field :- Recieves Acknowledgement.

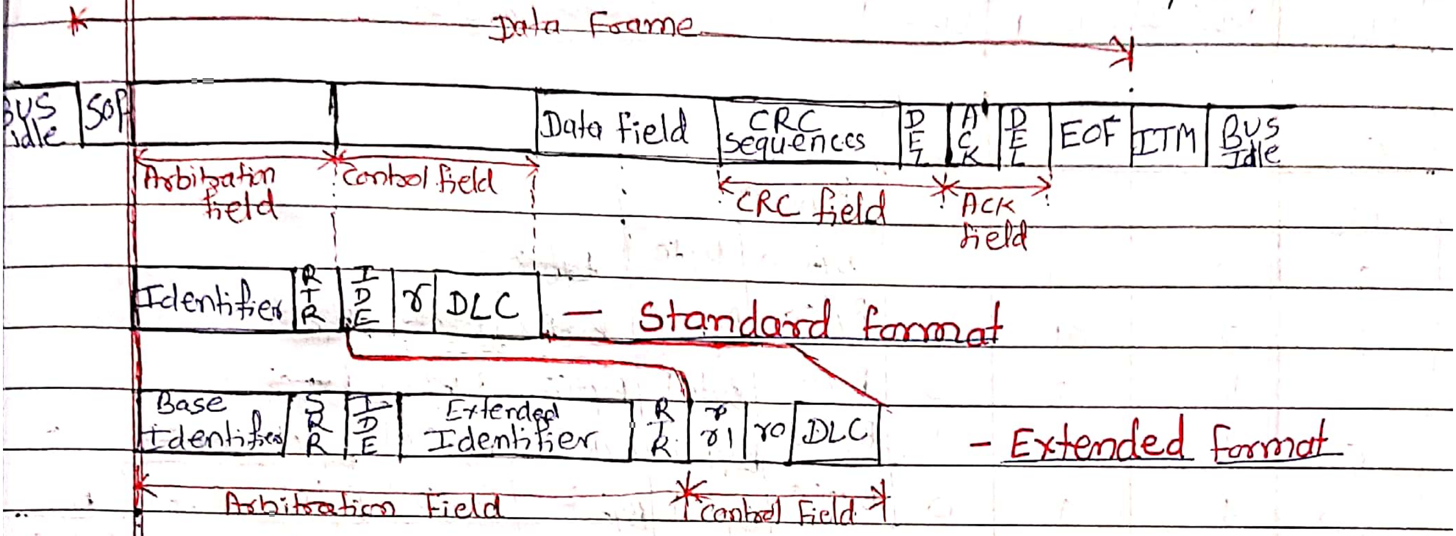
- ~~Start of~~ frame :- every new can frame ea start with it
- SOF – Start of frame bit (1 Bit) → It is dominant to start the frame
- Identifier – A message identifier sets the priority of the data frame.
- RTR – Remote Transmision Request, defines the frame type (data frame or remote frame) (1 Bit)
- It is dominant when information is required from another node. Each node recieves the request but only that node whose identifier matches that of the msg is the required node. Each node recieves the response as well as.

Dominant - 0    Recessive - 1



Identifier - It decides the priority of message. Lower the binary value then higher is the priority. It is 11 bit.

- R0 = Reserve bit - for future application - 1 bit
- Control field :- User defined function.
- DLC - Data Length Code (4 bits)
- Data field - User Defined Data (0 to 8 bytes)




- IDE bit in the control field stands for Identifier extension.
- A dominant(0) IDE bit Indicates 11 bit Standard frame Identifier
- And Recessive(1) IDE bit Indicates 29 bit Extended frame Identifier

- Then in the Control field we have Data Length code bits which define the length of the data in the Data field being sent.

- In data field maximum we can have upto 8 bytes

- 4 bits are available in DLC to configure no. of Bytes required in a perticular msg frame.

- Data upto 64 bit of data can be transmitted.

| Bus Idle | SOF | Identifier | RTR | IDE | r | DLC | Data field | CRC sequence | DEL | ACK | DEL | EOF | ITM | BUS idle |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| DLC-3 | DLC-2 | DLC-1 | DLC-0 | Data field | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 1 | Byte 0 | | | | | | | |
| 0 | 0 | 1 | 0 | Byte 0 | Byte 1 | | | | | | |
| 0 | 0 | 1 | 1 | Byte0 | Byte1 | Byte2 | | | | | |
| 0 | 1 | 0 | 0 | Byte0 | Byte1 | Byte2 | Byte3 | | | | |
| 0 | 1 | 0 | 1 | Byte0 | Byte1 | Byte 2 | Byte3 | Byte4 | | | |
| 0 | 1 | 1 | 0 | Byte0 | Byte1 | Byte 2 | Byte3 | Byte4 | Byte5 | | |
| 0 | 1 | 1 | 1 | Byte0 | Byte1 | Byte 2 | Byte3 | Byte4 | Byte5 | Byte6 | |
| 1 | 0 | 0 | 0 | Byte0 | Byte1 | Byte 2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte 7 |

CRC field

| CRC sequences | DE |
|---|---|

- The frame also contain Cyclic Redundancy check which is of 15 Bit.
- which is use to detect any data corruption during transmission.
- The Sender will compute the CRC before sending the frame. → after recieving the frame each reciever will compute the CRC again and compare with this.
- If the CRC doesn't match then the Reciever will generate the error frame.

DF-value उसिवाय द code पाठवता जात नाही.

CRC - It contains the checksum (no. of Bits transmitted of the preceding application data for error detection.

Date ______
Page ______

CRC - Delimeter bit = 1bit it always recessive

* **ACK** :- Acknowledge — we will see how it handle Ack bit

- In most of the communication protocol a saperate acknowledgement frame is sent by reciever after recieving all the bytes of the from sender.
- but in can there is no saperate packet is sent.

→ **Acknowledgement** - It is 2 bit. It is dominant (0) if an accurate message is recieved.

2 bits → ① tells transmeter / Reciever
② It always recessive.

IFS / ITM - Interframe space
It contains the time required by the controller to move correctly recieved frame to its proper position

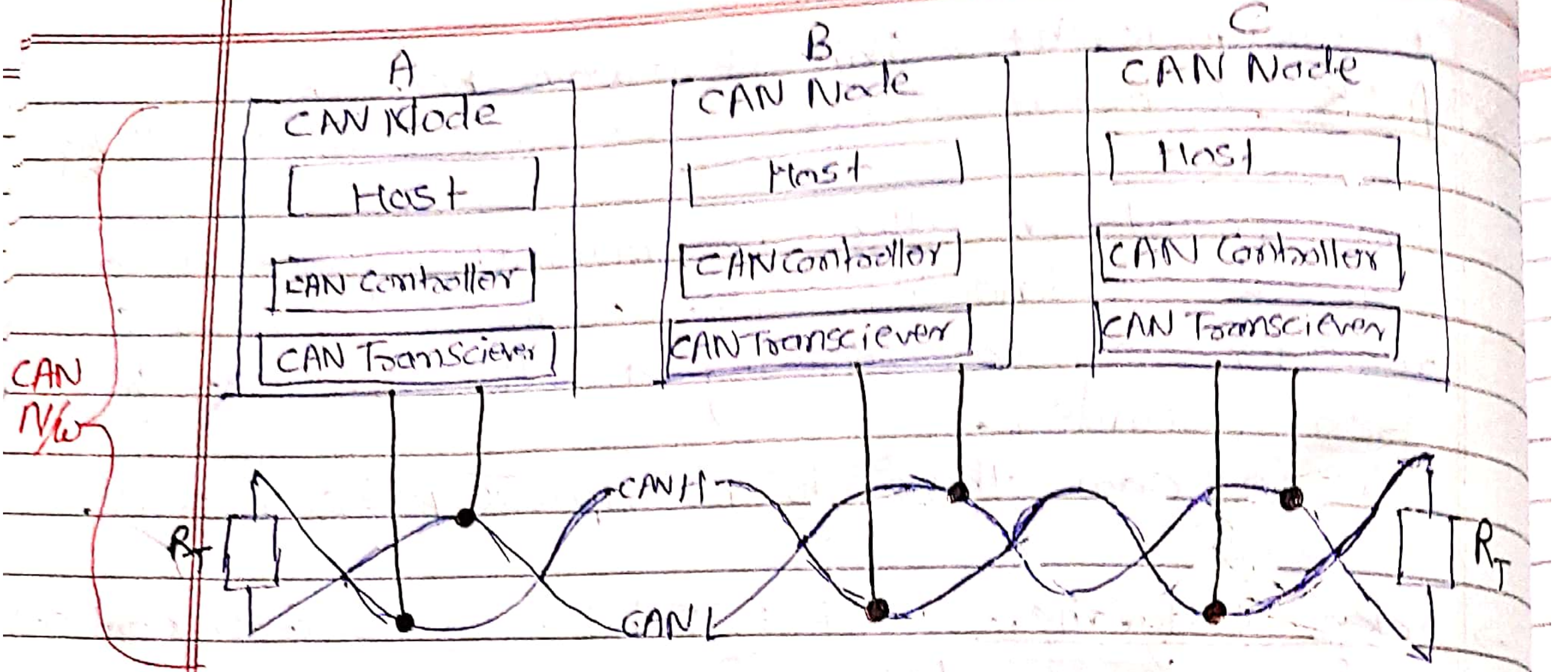
Byte 7.

* **EOF** :- End of Frame
- It marks end of frame & disable bit stuffing.
- It ends with 7 consecutive recessive bit called 7 bit

Qu. How the data in forms of 0's & 1's actually transmitted on can bus ?

→ Can bus electrical characteristics :-
↳ Can n/w consist of a no. of can nodes which are linked via a physical transmission medium.

* for eg. we will se that in below n/w ~~of~~ it have 3 can nodes. that is node A, B & C.
- A modern can node consist of 3 subelements a host it can be a microcontroller & or a ~~m~~ llC. which is actually running core application




- core application to do some specific job.
- A host decides what the recieved msg means & what msg it was to send next
-

CAN controller which deals with the commn function escribe by the can protocol & it also triggereds interrupt upon transmission ~~tt~~ & ~~recieve~~ reception of can messages

-

- Then it has can transciever which is responsible for transmission & reception of data on the CAN Bus.
- It convert can signal collected from the ~~#~~ Bus into stream of data which ~~can~~ CAN controller can understand.

- An unshielded twisted pair (2-wire line) is used & a physical transmission medium to transmit & recieve data it is also called can CAN BUS

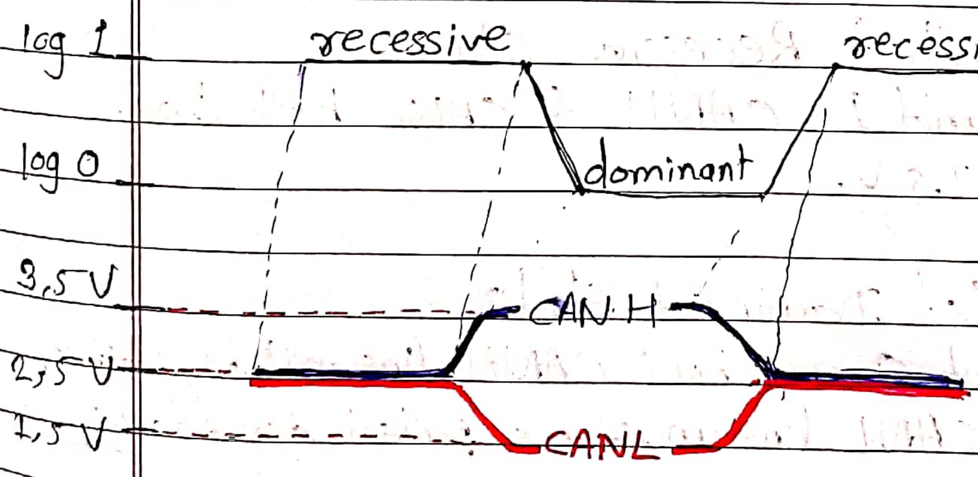
The bus consist of 2 lines CAN-H (CAN-High line) & CAN-L (CAN-Low line).

- The transmission happens with the help of differential voltage applied to the CAN-H & CAN-L

Q4. Why CAN uses twisted pair ~~cable~~ & differential voltage technique for communication?

→ Its bcoz of its environment, in a car including motors, & ignition system there are many devices which can cause data loss & Curreption due to noise.

- And twisting of these 2 lines also reduces the magnetic field. ~~The bus is termi~~
- The bus is terminated with the help of 120Ω resistance at each end.




voltage graph

This is a voltage graph which shows the voltage levels of CANH & CANL line.

In CAN → logic 1 = recessive &
logic 0 = dominant

- When CANH & CANL both are applied with 2.5 volt the actual differential voltage of the bus is 0 volt

- A 0 volt on the CAN bus is read by transciever as a recessive or logic 1.
- 0 V It is the idle state of a Bus.

- when CAN-H line is pulled up to 3.5V & CAN-L line pulled down to 1.5V the actual differential voltage of the 2 line is 2 Volt.
- which is treated as logic 0 or dominant bit by CAN transciever.

- If bus state is dominant or logic 0 by volt a node it is electrically impossible to drive the state to recessive by any other node.

- logic 1 is called Recessive state
→ So to transmit 1 → CANH & CANL both line should ~~be~~ at 2.5V.

- logic 0 is called Dominant State
→ So to transmit 0 on bus CANH line must be pulled up to 3.5V & CANL line must be pulled down to 1.5V.

- Idle state of bus is recessive.

- If a node made the bus state to dominant it cannot be drive back to recessive by any other node

BUS idl

Sta fo

Ech fo

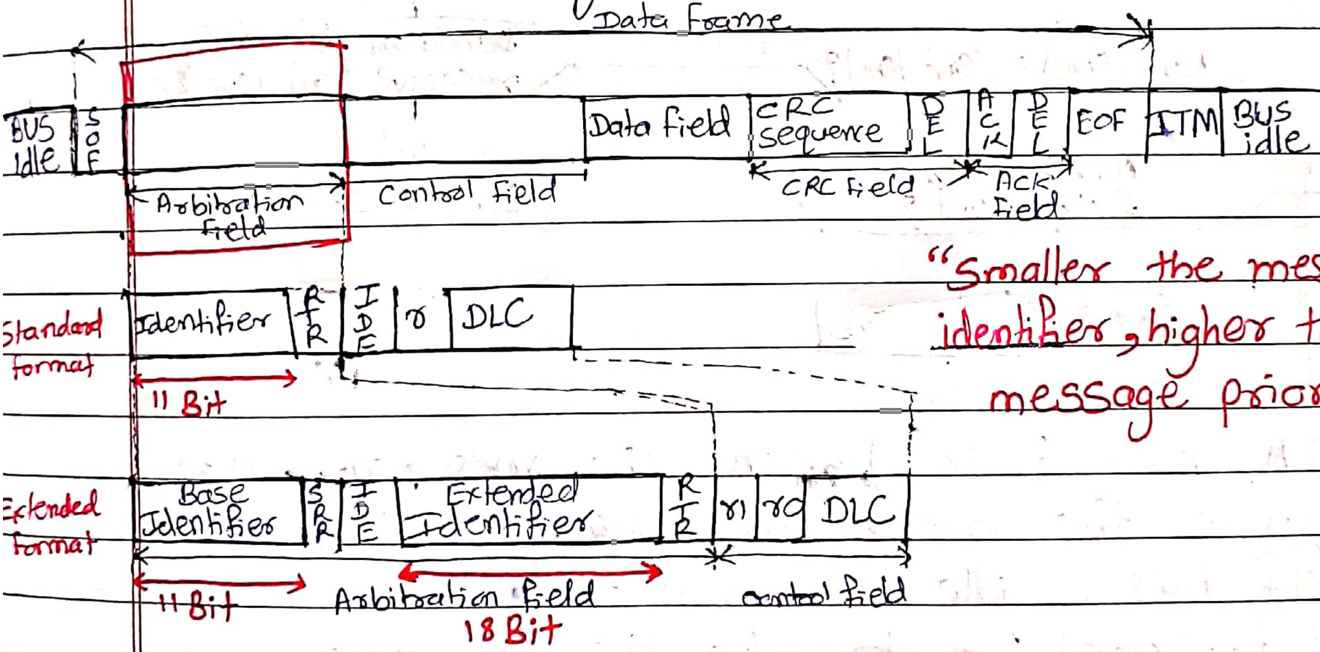
* CAN BUS Logic :-

Wired AND Logic

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sender 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Sender 2 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| Sender 3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Sender 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | Dominant | | | | | | | Recessive |

- Dominant state overrides the recessive state.
- So when different CAN node sends dominand & recessive bus level simultaneously the bus remains at dominant
- & The recessive bus level is accured only when All nodes send recessive state.

→ i.e. this is AND Logic
↳ It is physically implemented by open collector circuit

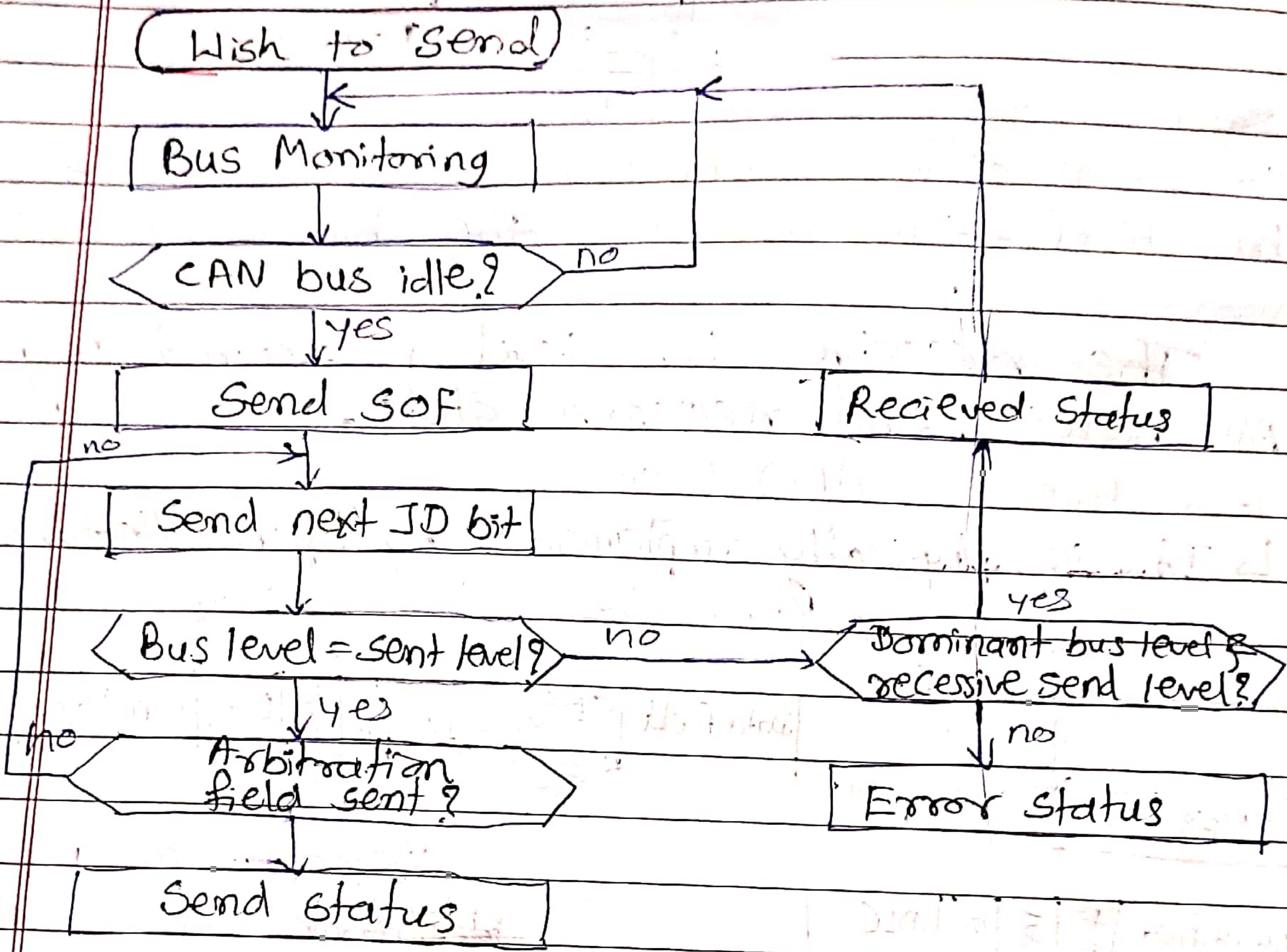



"Smaller the message identifier, higher the message priority"

→ In can communication priority of msg is devendent message identifier.
→ standard frame = 11 bit identifier
Extended frame = 29 bit identifier
- This allows system designer to prioritized the msgs during design time itself.

* Arbitration works :-

CAN communication Principle :-




- A node wish to send the messasge has to wait until the bus become idele.
- If bus is idle it sends the start of frame bit dominant bit to take the bus access
→ Then it sends the message identifier ~~to~~ in MSB (most significant bit) or MSB. 1st session.
→ During this ~~busphase~~ the bus detect the a dominant bit on the bus while it has transmitted a recessive bit it clearly means that the node has lost the arbitration & it stops sending the further bit. It waits & try

Scanned with CamScanner

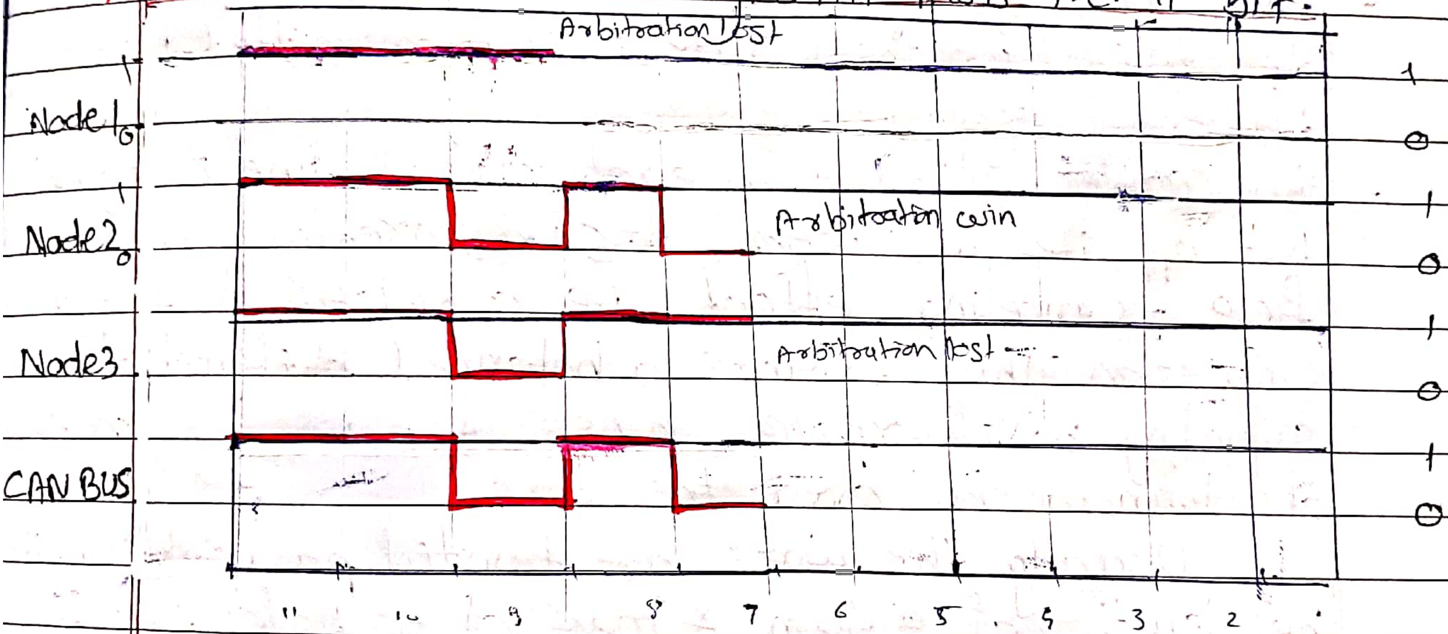
to again once the bus is free again.

for eg:-

| CAN Node | Identifier (Hex) | Identifier (Binary) |
|---|---|---|
| 1 | 0X7F3 | 11111110011 |
| 2 | 0X6B3 | 11010110011 |
| 3 | 0X6D9 | 11011011001 |

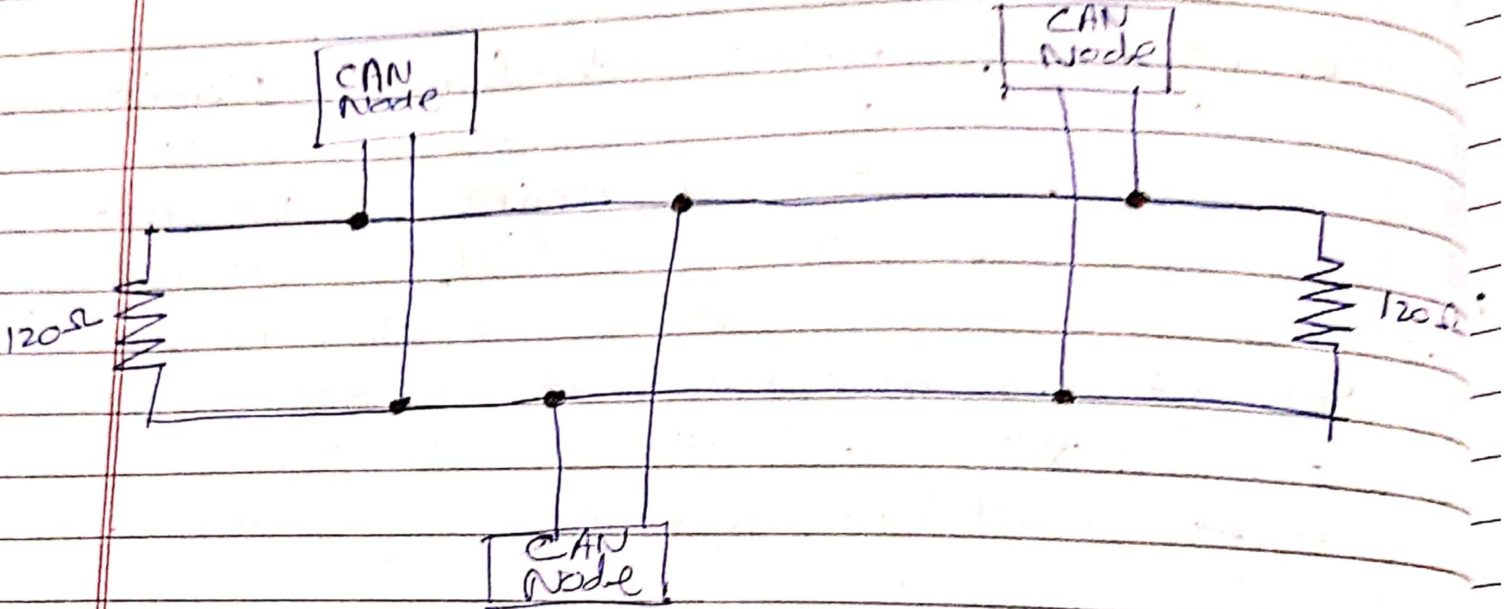
→ There are 3 nodes in n/w & all 3 are ready to transmit their messages node 1 to transmit message identifier 0X7F3 & all other nodes also. & its 11 bit binary representaion is given ~~th~~ here.

★ Lets start transmitting with MSB i.e. $11^{th}$ bit.




→ As firstly ~~at~~ all 3 nodes have $11^{th}$ bit recessive the bus state remain recessive Even $10^{th}$ bit also
→ recessive..
- $9^{th}$ bit node 1 will send a recessive bit but node 2 & 3 send dominant bit the bus state will remain at dominant.
- Then node 1 will lost the arbitration at $8^{th}$ bit the rest 2 & 3 sending recessive bit then CAN bus will be recessive
→ $7^{th}$ bit node 2 sending dominant & 3 sending recessive it clearly means node 3 loss the arbitration & 2 will continuing to transfer msg.

* CAN consists of 2 wires serial link → CAN-H & CAN-L.

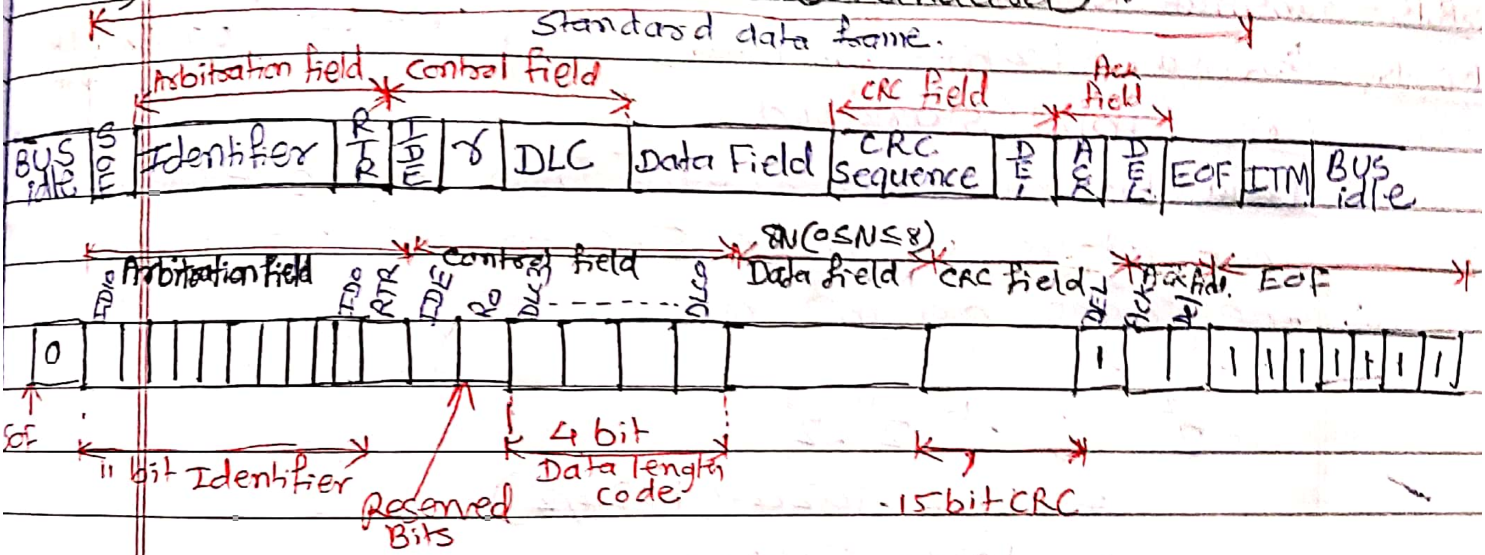



- And their voltage levels relatives to each other determine whether 1 or 0 is transmitted. This is differential signaling.
- The current flowing in each signal line is equal but opposite in direction, ~~resc~~ researching in field-canceling effect i.e. a key to low noise emmission. This ensures a balanced differential signaling which reduces noise & allows high rate of transmission over the wires.

Usually the wires are twisted pair cables with bus length of 40m & $max^m$ of 30 nodes. It is a shielded or unshielded cable with characteristic impedence of 120 Ω.

## CAN Frames :-
1) Data frame (Standard, Extended)
2) Remote frame (Standard, Extended)
3) Error frame (Passive, Active)
4) Overloaded frame

1) Data Frame :-
A) 11 bit identifier field (Standard) :-

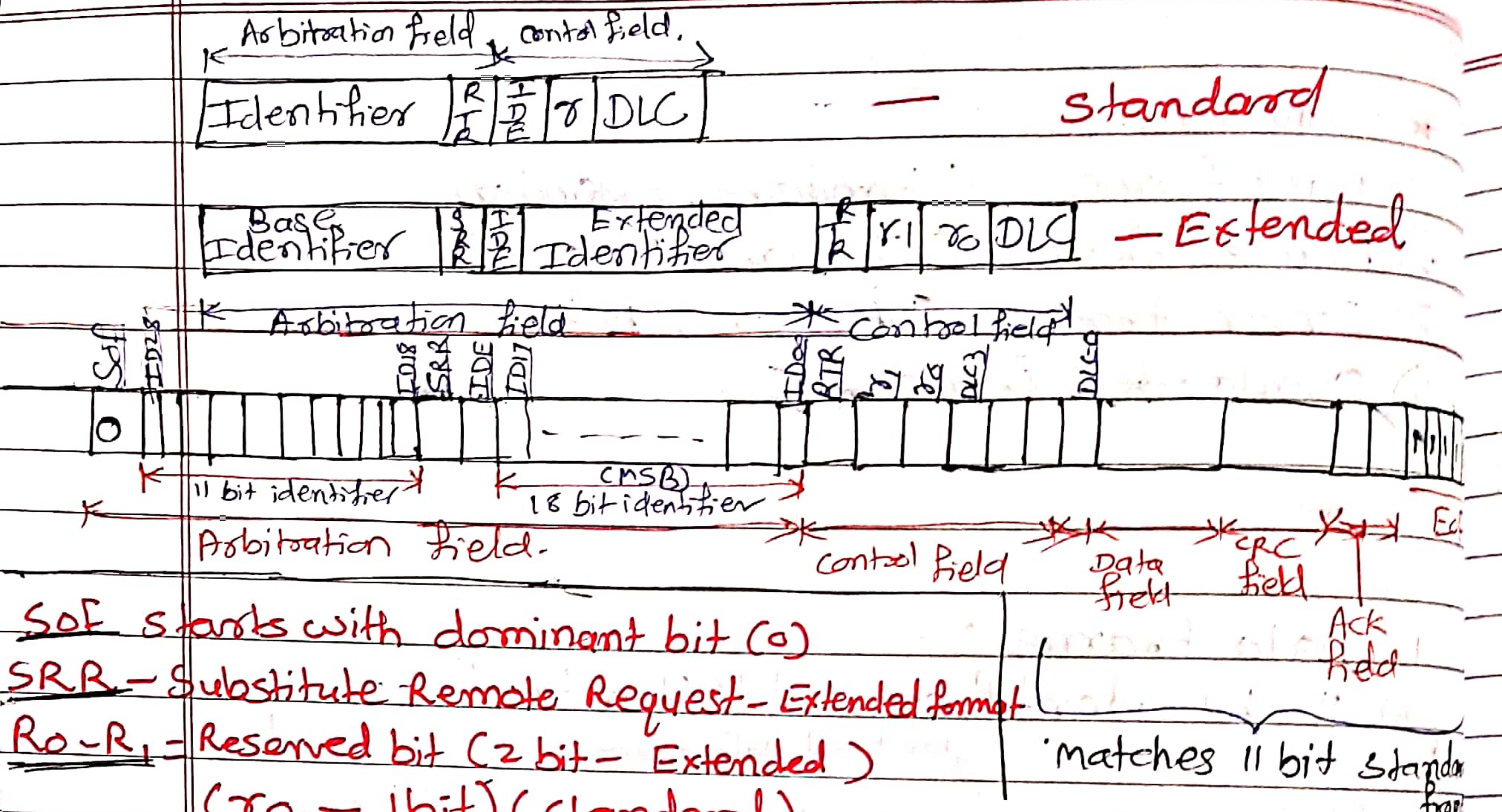



* Data frame start with sof which is dominant (0)

→ This is the data frame
- The data frame consist of the actual payload or data which is sent by the node who is responsible for collecting data.

- The major differences between the two formats is the Arbitration field.
- The Standard frame has 11 bit identifier in the arbitration field. that allows upto 2048 unique messages onto the bus.

$$2^{11} \Rightarrow 2^{10} \cdot 2 \Rightarrow 1024 \times 2 = 2048$$




SOF starts with dominant bit (0)
SRR – Substitute Remote Request – Extended format
$R_0 - R_1$ – Reserved bit (2 bit – Extended)
($r_0$ – 1 bit) (standard)

→ The extended ~~bit~~ frame has 29 bit identifier that allows up over 536 million unique messages on the BUS.

$$2^{29} \Rightarrow 2^{10} \cdot 2^{10} \cdot 2^{9}$$
$$\Rightarrow 1024 \cdot 1024 \cdot (1024/2)$$
$$\Rightarrow 1024 \cdot 524288$$
$$\Rightarrow 536,870,912.$$

Both standard & extended data frames can coexist on the same CAN bus. The standard ~~the~~ 11 bit data frame will always have priority over the same extended 29 bit frame with identical 11 bit base identifier.

Note :- It is recommende that no two nodes transmit the same data frame in terms of ID ... bcoz this would break the arbitration process.

Every node on the bus is responsible to check the message for errors that range from bit stuffing to CRC error.

Control field → Contains → IDE bit & data Length code (DLC)

IDE bit → if message is standard or extended
DLC → if message is bet$^n$ 0 to 8 data bytes.

- CRC is calculated from SoF to the data field.

- At the ack field the transmitter transmit a reciever bit (1) on the bus & listens for the reciever to transmit a dominant (0).
- If upto that point there are no errors with this data frame all recievers should transmit a dominant. This let the transmitting node know that the receiving node recieved the message ok.

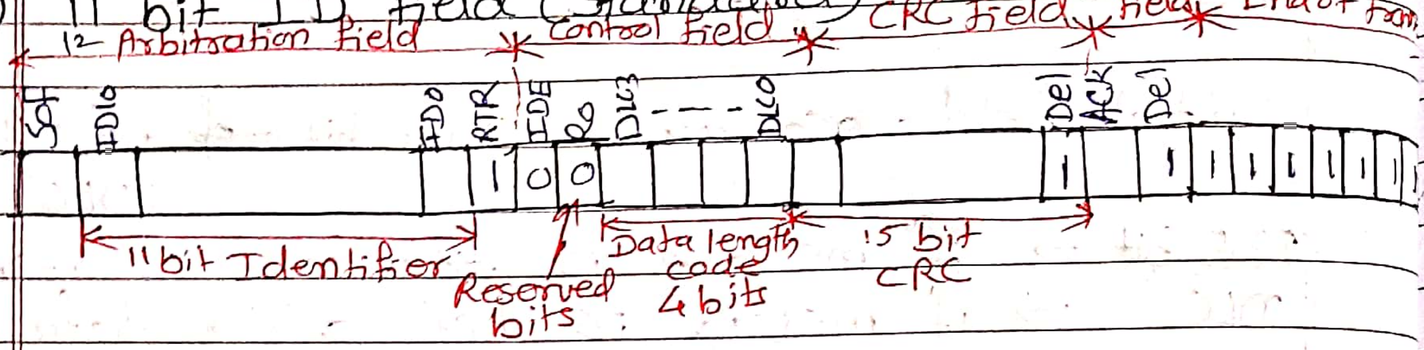
- At the EOF and 3 bits of quiet time; also called inter-frame space (IFS); all nodes including that node that just transmitted will have a chance to transmit again.

- ID of message determines the priority & gives context to information in the data field.

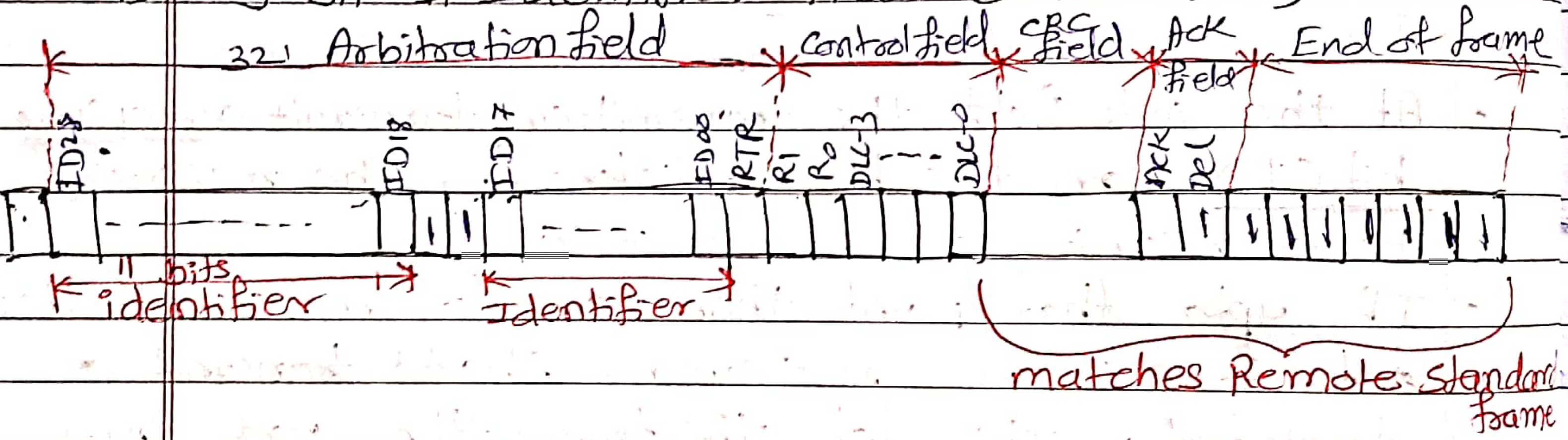
Scanned with CamScanner

## 2) Remote Frame :-

i) 11 bit ID field (Standard)




ii) 29 bit Identifier field (Extended)



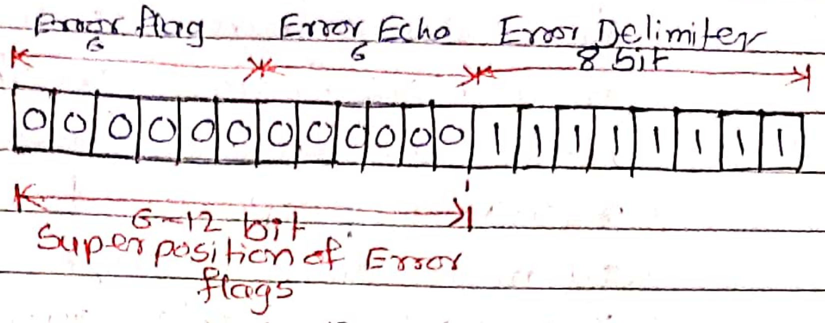

- Remote frames are used when one node needs to request data from another node

Example:- Node A will transmit a remote frame request onto the bus with an identifier that matches a data frame that typically sent by node B.
upon seeing the RTR request node B will transmit the data frame matching ID of data.

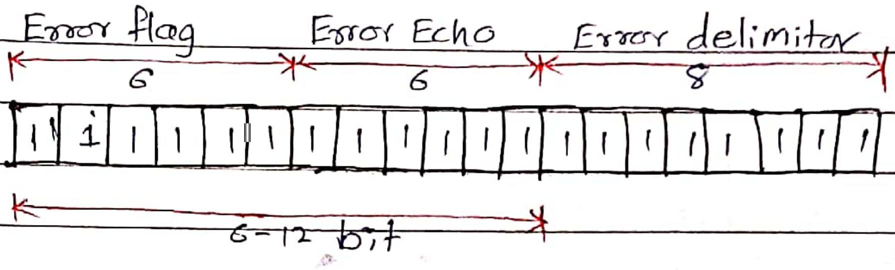
Remote frames are not typically used in Non-priority HLP protocols, but can be found in some priority protocol (High layer protocol.

## 3) Error Frame :-

### A) Active Error frame :-



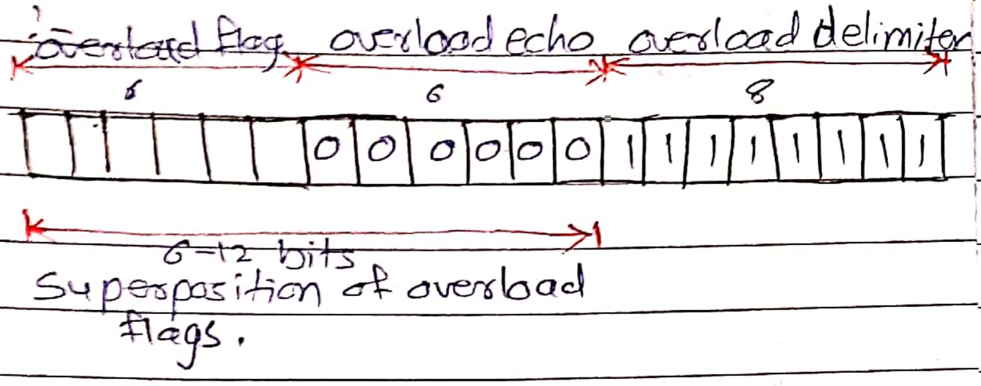

### B) Passive Error Frame :-




- Error frames are transmitted by any nodes when they detect an error with the CAN data or remote frame.
- If the node is in the error active state it will transmit out an active error frame.
- If the node is in the passive error state it will transmit out a passive error frame.
- The error frame can vary in length bcoz it is possible for some noise localized next to a CAN node on the bus will flip a bit in data frame.
- If this occurs that node might see the bit flip while other nodes on the bus will not. Bcoz of this ~~Bi~~ 'bit flip' the CRC calculation should be wrong, & it is possible that the bit stuffing rule may be violated. If that is the case that node will transmit out an error frame, & soon after the other nodes on the bus will start to see the bit stuffing rule being violated and they too will transmit out echoing error flags.

These error frames will "destroy" the current data or remote frame on the bus. The transmitting node will knows that its message wasn't recieved properly by all nodes, and will automatically attempt a retransmission at the next available quiet time on the bus.

4) Overload frame :




- An overload frame is special version of the Error frame that will not ~~b~~ coz a retransmission of the last destroyed message. Instead it is used by a Node to request a delay bet$^n$ data or remote frames during the 'inter-frame' space or quiet time on the CAN bus.