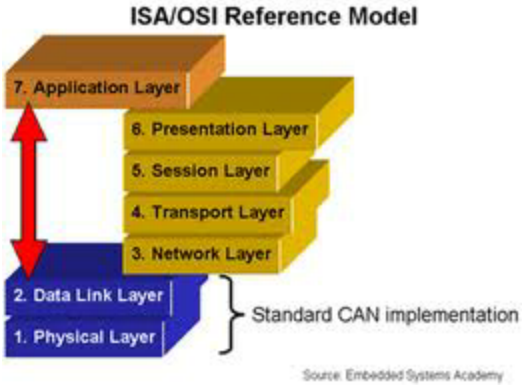
CAN

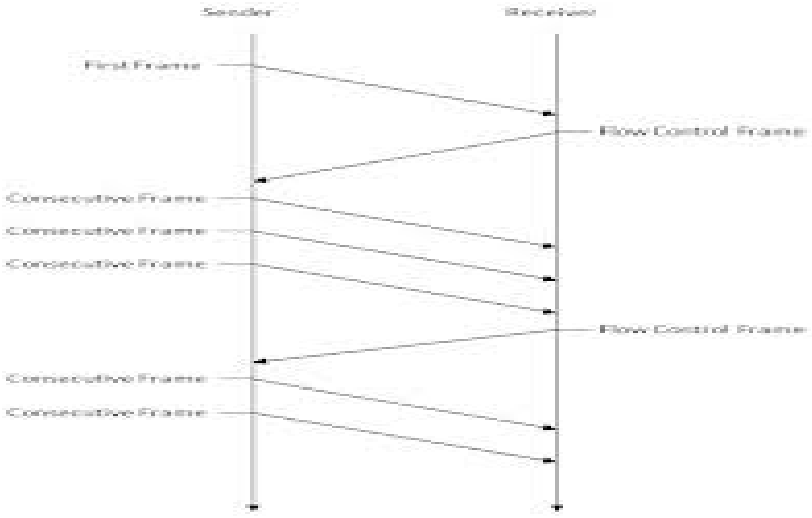
CAN(Controller Area Network): CAN is a Automotive standard protocol. It is a serial asynchronous (In CAN the data transmission is not based on Clock (like in SPI) so CAN is asynchronous. Instead of Clock it is based on Time quanta) protocol. Bit rate is up to 1Mbps and length is up to 40m. whenever speed is 500 kbps then length is 100m, 250kbps-200m……..up to 10kbps-6km.

If it is 1Mbps then 40m, 500Kbps-100m……10Kbps-6km why the reason is Due to RLC in the circuit L (Inductance) doesn't allow sudden changes in the current and C (Capacitance) doesn't allow the sudden changes in the voltages.

CAN is based on 3 layers Physical layer, Data link layer and Application layer. Physical layer is responsible for Bit encoding, Bit decoding and Bit transmission. Data link layer is responsible for Message transmission, reception and error handling.

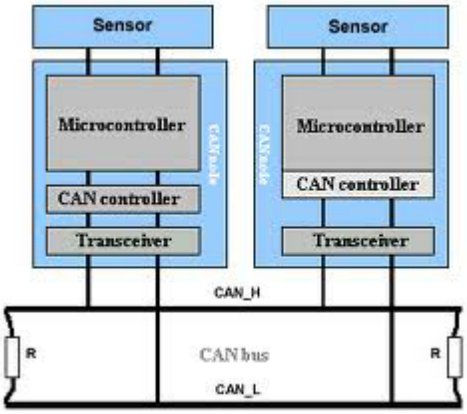


In between we are bypassing these 4 layers. In between these layers in the CAN driver development process CAN_Init, CAN_read, CAN_Write API's done. CAN is limited up to 8 bytes data transmission. If we want to transmit more than 8 bytes then we go for the HLP(Higher layer protocols) like CANopen, Devicenet and ISO15765.



In **ISO15765** we can transmit up to 4096bytes(4KB). The frame format of ISO15765 is single frame, first frame then flow control followed by consecutive frames.

The single frame can carry data up to 7bytes. If the data is more than 7 bytes the first frame can indicate the remaining data along with the carry of the some initial data. The flow control is response from the receiver acknowledging the first frame along with the indication of Block size and minimum separation time between the consecutive frames. The consecutive frame is a frame containing the subsequent data.



CAN is half duplex two wire interface. The both ends are shorted with 120ohm resistor because to oppose the signal reflection from both ends and also to match the impedance. Every CAN Transceiver is having internal resistance of 60 ohm. And the two ends are shorted with 120 ohm those 2 resistors are in parallel so ((120*120)/(120+120))=60 ohm. According to maximum power transform theorem the Source Impedance = Load Impedance.

CAN why it is 2-wire interface: for differential voltage. CAN is two wire interface the data transmission is on 2 wires i.e., CAN_H and CAN_L. If the data is transmitting b/w two wires some amount of noise is added in wire i.e.,CAN_H the same amount will be added on the other side with opposite.

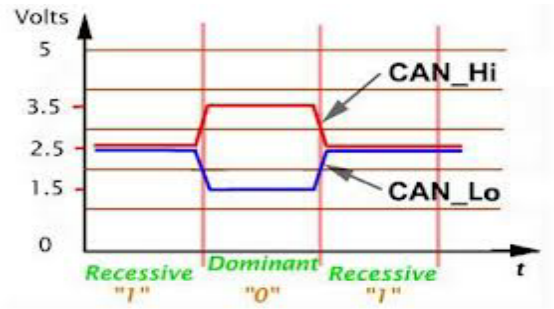
In CAN_H the noise+ data Δx+a

In CAN_L the noise+ data Δx+b

The differential voltage (Δx+a)-(Δx+b)= a-b

Plan data we will get that's why CAN is 2-wire interface.

CAN physical layer voltages:



CAN dominant voltages are from 1.5v to 3.5v.

CAN recessive voltage 2.5v.

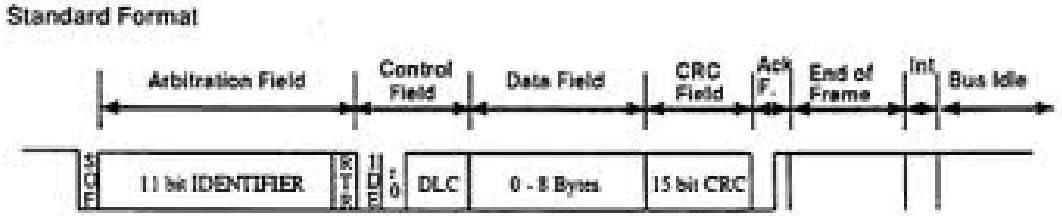
Why like this when it is in dominant 1.5-3.5 when recessive 2.5 and the 2wires are CAN_H and CAN_L are 2.5v when recessive. The reason is every CAN transceiver is having internal transistor the transistors are doesn't allow the voltage/power transmissions when it is in recessive (i.e., 1).It allows only when dominant (i.e., 0). And the voltage difference between CAN_H and CAN_L is whenever >0.9 then only it identifies the bit transmission is dominant otherwise (i.e.,<0.9) it is recessive.

CAN is two formats: Standard format (11-bit Identifier) and Extended format (29-Bit Identifier).

CAN Frame format:

Data Frame:

It carries data from transmitter to the receiver.



SOF: Is always dominant (low(0))., because all nodes are synchronized within this segment only. On edge is expected to lie within this segment only. The Hard synchronization is done in the SOF.

Arbitration: the identifier of the message.

What is mean by arbitration?

In CAN if bus is idle any node can transmit data if two or more nodes transmitting the data at the same time the bus conflict will occur according to the arbitration the highest priority message will be transmitted first. Highest priority means the lowest numeric value will be the highest priority i.e., dominant bit.

RTR: (remote transmission request) is zero(dominant) for data frame 1(recessive) for remote frame.

If we transmitting data and remote frame then data frame will be transmitted first because RTR is 0 for data frame which is the highest priority.

IDE: (Identifier Extension Bit) Is 0 (dominant) for standard format and 1(recessive) for extended format.

If we transmitting Standard and extended data frame then standard frame will be transmitted first because IDE is 0 for standard format which is the highest priority.

Dominant is the standard format and recessive is the extended format.

Control field: Total of 6 bits in this 4 bits are used for DLC(Data length code) from 0000 to 1000 according to 8421 code (possible up to 8bytes). Reserved 2 bits are r1 and r0 in standard format r1 is for IDE and r0 is reserved.

Data field: is up to 8 bytes data transmission. From 0 to 64 bits of data transmission.

CRC: (Cyclic redundancy check) is of 16 bits. 15 bit CRC and 1 delimiter. Polynomial is used for CRC checking.

ACK: 2 bits. ACK Slot and ACK delimiter. In ACK field the transmitting node sends two recessive bits if the receiving node successful reception of message it will indicate to the transmitting node by sending dominant bit to the ACK slot i.e., suppression of the recessive bit.

EOF: (End of frame) 7 recessive bits.

IFS: (Inter frame Space) 3 bits long.

These two fields EOF and IFS are to indicate the BUS Idle condition.

Remote frame: RTR is 1(recessive). In this there is no data field.

To request the transmission of the data frame with the same identifier. Most of them recommended that don't use the remote frame instead of request the data put the response in the bus. If we are putting the data on the bus then any node needs the data then that node can take it easily unnecessary transmission of remote frame.

Error Frame:

It is transmitted by any node by detecting the error on the bus.

If error is detected it can put 6 error flags on the bus. Upto 12 error flags is possible. After 6 dominant bits don't put the error frame if also the error frame is detected wait for the successful transmission of the message.

Error Counters: TEC and REC (transmit error counter and receive error counter) when it is in 96 warning limit and TEC and REC are >127 then it is in error passive state. TEC>256 then it is in bus off state. A node in bus off to come back to error active state wait for the 128 occurrences of 11 consecutive bits.

OR most preferable answer for the interviewer is we can reset by the software i.e., CAN_Reset API in the driver development module.

Types of Errors:

Bit Error: Whenever a node transmitting the data it is also monitors the data until the receiver receives the data. It is expected that the data to be same until the receiver receives. For example if the node is transmitting 0(dominant) to the bus and it is detected 1(recessive) in the bus then it is bit error.

ACK Error: whenever the transmitting node doesn't monitor the dominant bit in the ACK slot then it is ACK error.

Stuff Error: After 5 consecutive bits the 6^{th} bit will be invert (opposite) of the 5^{th} bit otherwise it will be Stuff Error.

Normally the data frame will be 108 bits for standard format and 108+18 for the extended format but the two standard and extended formats are varied (total bits in both formats) based on number of stuff bits inserted.

The stuff bits inserted because for the error detection. After 5 consecutive bits if we are inserting the 6^{th} bit opposite to the 5^{th} bit on edge is occurred the Resynchronization is possible.

The Resynchronization is for the error detection.

Synchronization: It is the co-ordination of events.

CAN is based on NRZ signaling why because the bit time is constant for the 1 successful bit level.

In RZ signaling synchronization possible is very easy but CAN is chosen only NRZ. Because in CAN bit time calculation the entire bit time is constant for one bit level.

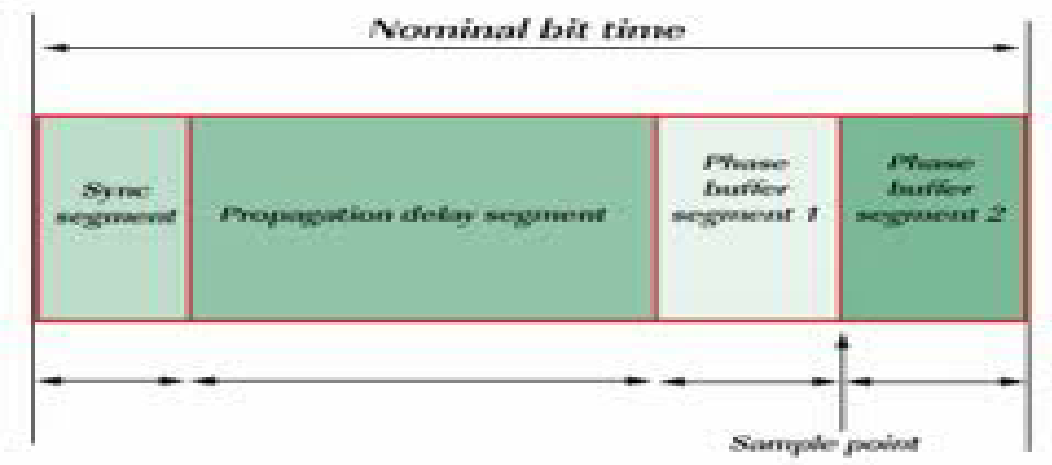
Form Error: In CAN formats the fixed formats bits are CRC Delimiter, ACK Delimiter and EOF these are fixed format fields always recessive for the resynchronization. If any of the bits are dominant then it is form error.

CRC Error: The result of the CRC calculation by transmitting side and receiving side is not same then it is CRC error.

Overload frame:

It is like IFS (inter frame space) to provide the delay between the 2 data frames.

Bit timing Calculation in CAN:



In bit timing calculation the CAN bit timing is Comprised into four segments.

Synchronization segment: this is 1 TQ long. An edge is lie within this segment only. The Hard synchronization is done within this segment only.

TQ: (Time quanta) It is a fixed unit of time derived from the master clock.

1 Time quanta is equal to 1 clock period. It is 1 for the synchronization segment and 1TQ to 8 TQ for Propagation segment and Phase buffer segments.

If the CAN bit rate is 1Mbps then TQ is 12 other than all bit rates (500kbps to 10kbps) the time quanta is 12 and it is fixed.

Propagation Segment: The physical delay adjustments.

Phase Buffer Segment-1 and Phase Buffer Segment-2: The phase error adjustments lengthened or shorthand based on the SJW (synchronization jump width).

Sample Point: It is to indicate where the bus level is read. It is after the phase buffer segment-1 and before the phase buffer segment-2. It is to be adjusted based on the phase buffer segment-1 and phase buffer segment-2.