Lesson-19: SERIAL BUS COMMUNICATION PROTOCOLS – CAN
Serial Communication Distributed Control Area Network (CAN) Bus

Distributed Control Area Network example - a network of embedded systems in automobile
CAN Serial Bus Communication for networking

- CAN-bus line usually interconnects to a CAN controller between line and host at the node. It gives the input and gets output between the physical and data link layers at the host node.
- The CAN controller has a BIU (bus interface unit consisting of buffer and driver), protocol controller, status-cum-control registers, receiver-buffer and message objects. These units connect the host node through the host interface circuit.
Serial IO CAN bus

Serial bus controller for CAN in a Microcontroller

Serial CAN bus

- CAN Device controller
  - Processor of system B
- CAN Device controller
  - Processor of system C
- CAN Device controller
  - Processor of system D
- CAN Device controller
  - Processor of system E
Serial IO CAN bus in automobile

- CAN Controller
- 1 Mbps multicast, multi-master, auto retransmission of erroneous data, 120 Ohm line impedance
- Twisted pair CAN-H and CAN-L wires
- Serial IO bus

Embedded controller system A
CAN controller
Embedded controller system B (engine controller)
- CAN controller
Embedded controller system C (antilock brake controller)
- CAN controller
Embedded controller system D (dash board)
- CAN controller
Embedded controller system E (transmission)
Three standards:
- 33 kbps CAN,
- 110 kbps Fault Tolerant CAN,
- 1 Mbps High Speed CAN
**CAN protocol**

There is a CAN controller between the CAN line and the host node.

- CAN controller—BIU (Bus Interface Unit) consisting of a buffer and driver
- Method for arbitration—CSMA/AMP (Carrier Sense Multiple Access with Arbitration on Message Priority basis)
Each Distributed Node Uses:

- Twisted Pair Connection up to 40 m – for bi-directional data
- Line, which pulls to Logic 1 through a resistor between the line and +4.5V to +12V.
- Line Idle state Logic 1 (Recessive state)
Each Distributed Node:

- Uses a buffer gate between an input pin and the CAN line
- Detects Input Presence at the CAN line pulled down to dominant (active) state logic 0 (ground ~ 0V) by a sender to the CAN line
Each Distributed Node:

- Uses a current driver between the output pin and CAN line and pulls line down to dominant (active) state logic 0 (ground ~ 0V) when sending to the CAN line
Protocol defined start bit followed by six fields of frame bits

- Data frame starts after first detecting that dominant state is not present at the CAN line with logic 1 (R state) to 0 (D state transition) for one serial bit interval
- After start bit, six fields starting from arbitration field and ends with seven logic 0s end-field
- 3-bit minimum inter frame gap before next start bit (R→D transition) occurs
CAN Protocol defined frame bits

- Start bit R to D transition
- 12-bit arbitration field 11-bit identifier, RTR bit
- 6-bit Control field
- 0-64 bits Data length code
- 15-bit CRC, 1-bit delimiter
- 1-bit ACK slot, 1-bit delimiter
- 7 bits 0000000 for frame end
- 3-bit minimum inter frame gap
- Time

Data frame (standard 11 bit identifier case)

State

R (Recessive)
D (Dominant)
Protocol defined First field in frame bits

- First field of 12 bits —'arbitration field.
- 11-bit destination address and RTR bit (Remote Transmission Request)
- Destination device address specified in an 11-bit sub-field and whether the data byte being sent is a data for the device or a request to the device in 1-bit sub-field.
- Maximum $2^{11}$ devices can connect a CAN controller in case of 11-bit address field standard
11-bit address standard CAN

- Identifies the device to which data is being sent or request is being made.
- When RTR bit is at '1', it means this packet is for the device at destination address. If this bit is at '0' (dominant state) it means, this packet is a request for the data from the device.
Protocol defined frame bits Second field

- Second field of 6 bits—control field. The first bit is for the identifier’s extension.
- The second bit is always '1'.
- The last 4 bits specify code for data length
Protocol defined frame bits Third field

- Third field of 0 to 64 bits—Its length depends on the data length code in the control field.
Protocol defined frame bits Fourth field

- Fourth field (third if data field has no bit present) of 16 bits—CRC (Cyclic Redundancy Check) bits.
- The receiver node uses it to detect the errors, if any, during the transmission.
Protocol defined frame bits Fifth field

- Fifth field of 2 bits—First bit 'ACK slot'
- ACK = '1' and receiver sends back '0' in this slot when the receiver detects an error in the reception.
- Sender after sensing '0' in the ACK slot, generally retransmits the data frame.
- Second bit 'ACK delimiter' bit. It signals the end of ACK field.
- If the transmitting node does not receive any acknowledgement of data frame within a specified time slot, it should retransmit.
Protocol defined frame bits Sixth field

- Sixth field of 7-bits — end-of-the-frame specification and has seven '0's
Summary
We learnt

- CAN serial bus for interconnecting a central control network.
- Widely used in automobiles.
- Fields for bus arbitration bits, control bits for address and data length, data bits, CRC check bits, acknowledgement bits and ending bits.
End of Lesson 19 of Chapter 3