Controller Area Network
(CAN)

Topics Covered

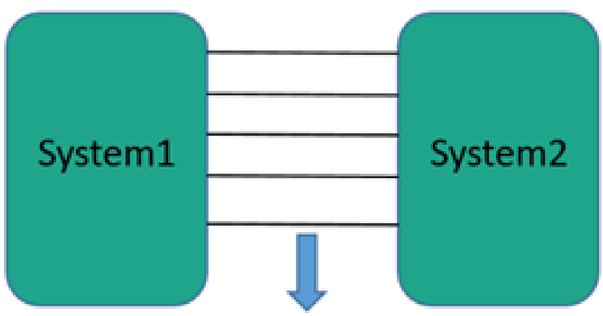
- What do we mean by communication?
- Types of communication
  - Serial and parallel communication
- Classification of serial communication
  - Half duplex and full duplex
  - Synchronus and Asynchronus
- Why CAN bus was invented?
- Advantages of CAN
- CAN protocol concepts
  - Broadcast mechanism
  - Differential bus
  - Bit stuffing
  - Termination resistor
  - Frame format
  - Bus arbitration

Topics Covered

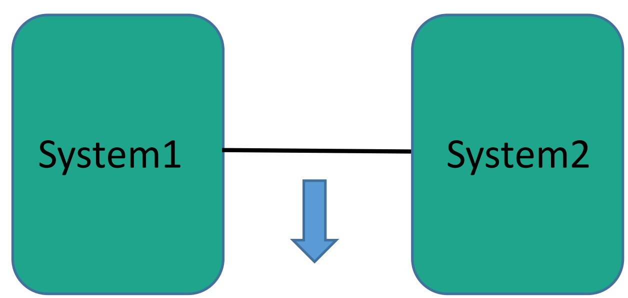
- Types of CAN frames
  - Data frame
  - Remote frame
  - Overload frame
  - Error frame
- Error handling mechanisms
  - Bit error
  - Bit stuffing error
  - Form/Frame error
  - Acknowledgement error
  - CRC error

What is communication?

- Exchange of information between entities
  - Entities can be microcontrollers.
- Types of communication
  - Parallel communication: More than 1 bit transferred at a time
  - Serial communication: One bit transferred at a time



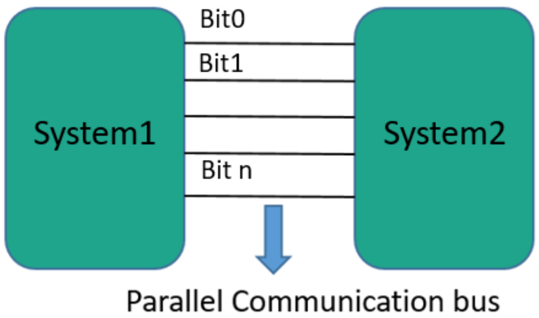
Parallel Communication bus



Serial Communication bus

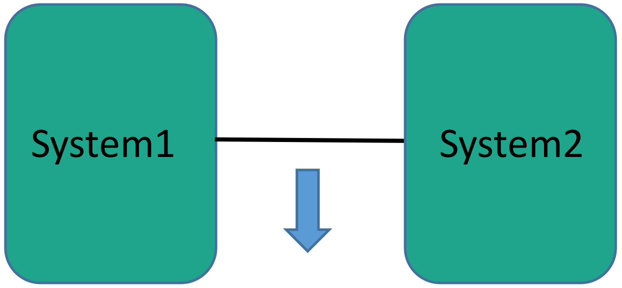
Parallel Communication

- Faster speed
- Wiring complexity is higher
- Space needed for implementation
- Higher cost of implementation at hardware level
- No need for synchronization
- Each signal path affects neighbouring signals – EMI
- Less reliability



Serial Communication

- Single wire connecting devices/systems
  - Data to be transmitted is broken into individual bits and one bit is sent at a time.
  - Data made up of large number of bits is transmitted as a train → one bit at a time
  - Receiver has to reconstruct the bits back into original data form



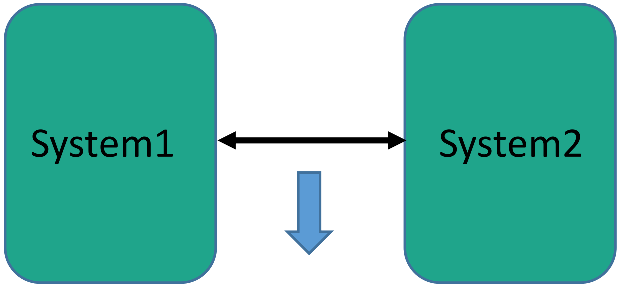
Serial Communication bus

Classification of serial communication

- Classification of serial communication
  - Half duplex and Full duplex
  - Synchronous and Asynchronous

Half duplex Communication

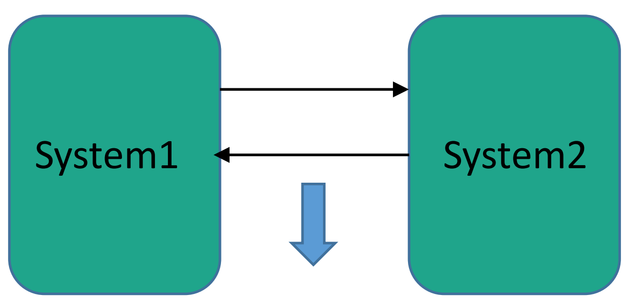
- Half duplex and full duplex communication
  - In Half duplex same wire/path is used for data transfer in both directions
  - Data transfer happens in one direction at a time
  - Transmitting and receiving cannot happen simultaneously



Half duplex

Full duplex Communication

- Full duplex communication
  - In Full duplex dedicated wire / path is used for each direction
  - Data transfer can happen simultaneously in both directions
  - If a system has to both transmit and receive signals, it can happen simultaneously independent of each other



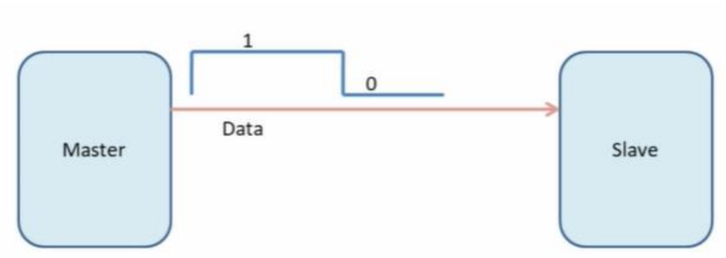
Full duplex

Serial communication

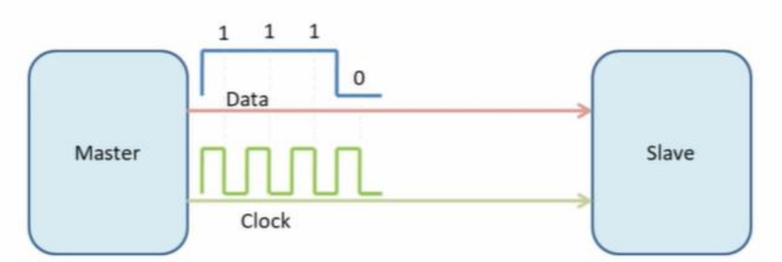
- Data/message to be sent is made up of large number of bits
  - In serial communication, data is broken into individual bits and send sequentially one bit at a time
  - At the end, receiver has to reconstruct the message for which it has to precisely know the start and end of the data bit
  - Possible ways of detecting each bit
    - Synchronous communication
    - Asynchronous communication

Synchronous Communication

- Two wires
  - One wire to carry data bits
  - Second wire carry clock (reference for all transactions)
- One system will become master and other becomes slave
- Master generates clock signal irrespective of data direction
- Suppose master wants to transmit: 1110
  - If no clock, slave see's the value as two bits 1 and 0
  - With reference clock, slave reads the value as 1110



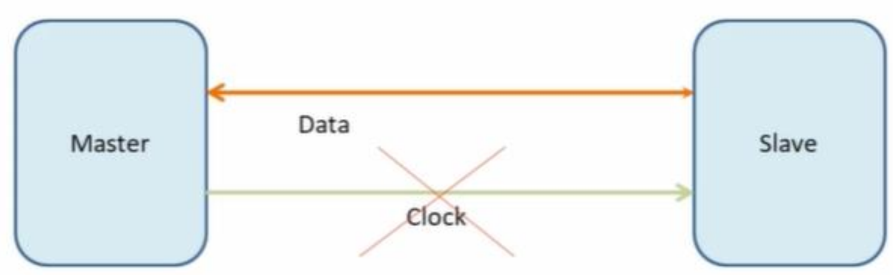
In absence of reference clock



In presence of reference clock

Synchronous Communication

- Limitations of synchronous communication
  - Heavily dependent on reference clock
  - Clock being high frequency signal, can cause data corruption
  - Synchronous communication not suitable for high speed communication
- To overcome this we go for asynchronous communication
  - Clock reference signal wire is removed



Asynchronous Communication

- Asynchronous communication has own way to achieve synchronization (that is to sample data correctly in correct instances)
- Conceptual blocks in asynchronous communication
  - Baud rate
    - Bits transferred per second from transmitter to receiver
    - Baud rate register must be configured with same value for all devices
    - Each device calculates bit duration = 1/Baud rate
    - Receiver will decode bit within bit duration
  - Idle state
    - Default state of data wire when no communication is happening
    - When transmitter is not sending any data, it will maintain logic high as idle state

Asynchronous Communication

- Start bit
  - It signals the end of idle state and beginning of communication cycle
  - Always precedes the data bits
  - Typical value of start bit is logic ‘0’
- Stop bit
  - Brings the communication cycle to end
  - Typical value of stop bit is logic ‘1’
  - After stop bit, data wire goes to idle state

Attributes of asynchronous communication

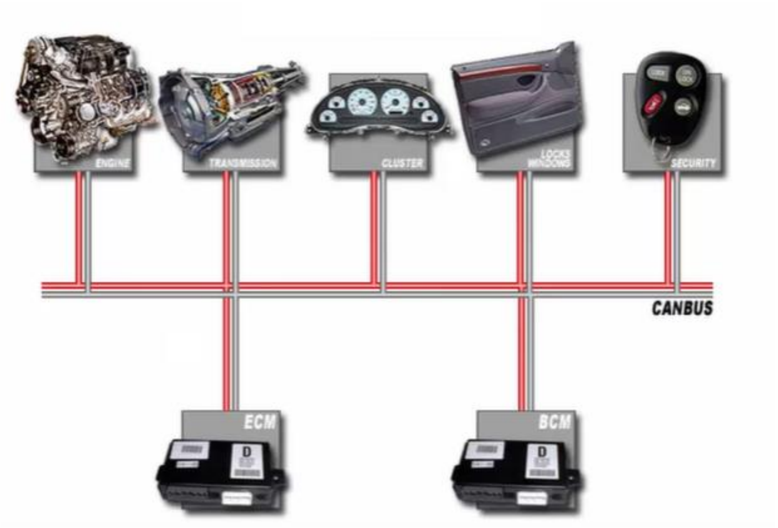
- Higher reliability
- Implementation complexity

Why CAN bus was invented?

- Older cars had very less electronics and sensors
  - Less numbers of wires where needed to connect
  - One wire carries one signal
  - ECU’s relied on increasingly complex point to point wiring
- As electronics increased in vehicle, expansion, maintenance, cost and weight increased
  - As number of wires increases, weight increases, fuel efficiency reduces, troubleshoot is a problem

CAN BUS

- CAN bus replaced large multi wire looms into a pair of wires
- All the sensors, devices are connected to this pair of wires (CAN bus)
  - Weight of wiring reduces



Example of CAN bus Operation

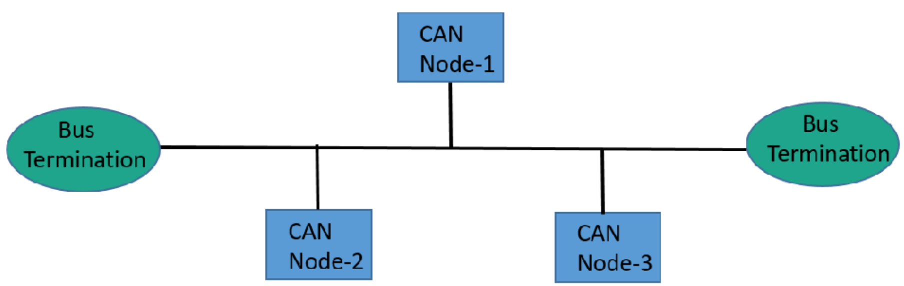
- Example: Power sliding door
  - Door should slide when vehicle is in rest and no obstruction on the way of closure
- When driver presses button on remote to close the door, signal from switch is broadcasted across the CAN bus network
  - BCM (responsible for closing door) checks whether CAR is moving or stationary (Information obtained from ECM)
  - While closing, BCM also checks for any obstruction in the path with the help of a sensor present in DCM
  - If No obstruction detected, and vehicle in rest : Sends a signal to close the door.

Advantages of CAN

- Low cost
  - Serial bus with two wires
- Fast
  - Data rate is 1MBPS @wire length of 40m
- Extremely robust (immune to noise)
  - Uses differential signaling
- Reliability
  - Error handling ability
- Functional addressing
  - Data messages contain identifiers relating to their function and also priority (No source and destination address)

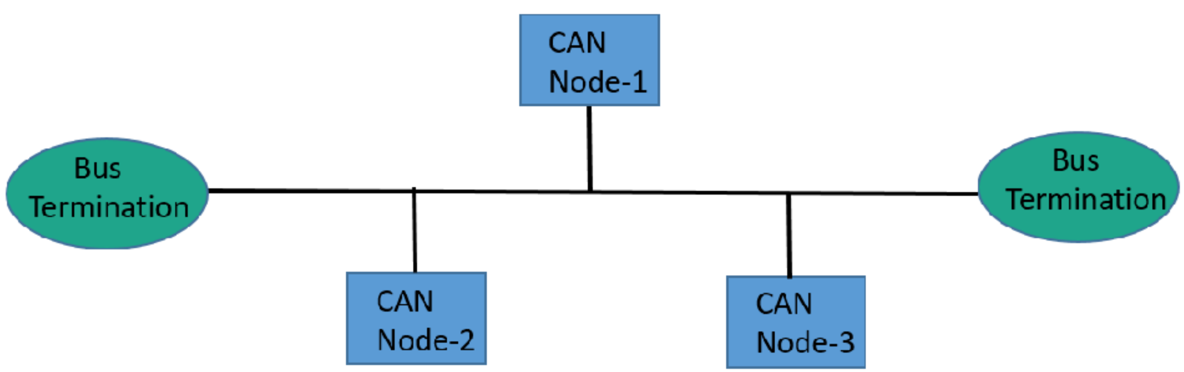
Advantages of CAN

- Broadcast type of bus
  - Unlike traditional network such as Ethernet and USB, CAN does not send point to point (that is specifically to a node)
  - Transmitter sends information with unique message ID onto the bus. This combination is called as CAN message
  - All nodes will default be in receiving mode. They continually monitor bus for messages
  - If they notice any message on the bus, it compares message ID and checks if it needs.
  - If a node, needs the message then it copies to its internal memory else remains silent till cycle is over.



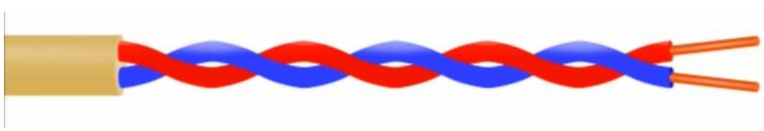
Advantages of CAN

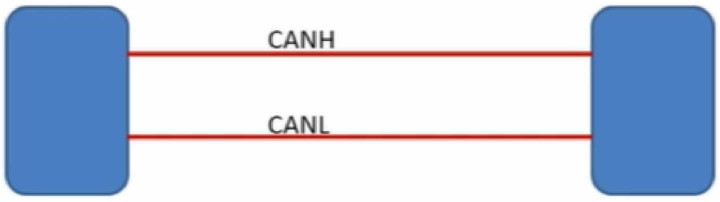
- Nodes may be added/removed at any time even while network is operating (Hot plugging)
- No limit on number of nodes.
- If a message having unique ID 123 and one more message having unique ID 213, message with ID 123 will win the bus → Bus arbitration
  - Collision detection take care in message ID only
- All node receive the message but only the relevant node will acknowledge



CAN Bus

- CAN physical layer has two wires
  - CanH and CanL
  - These two wires are twisted in the CAN cable
  - Transmitter sends same binary information on both CanH and CanL → NRZ code
    - In telecommunication, a non-return-to-zero (NRZ) line code is a binary code in which ones are represented by one significant condition, usually a positive voltage, while zeros are represented by some other significant condition, usually a negative voltage





CAN Bus

- Following are logic levels for logic 0 and logic 1
  - Voltage levels is as per ISO 11898 CAN standard
- Receiver will perform subtraction of the signals to get the logical bit

	CANH	CANL
Binary ‘0’	2.5 Volts	2.5 Volts
Binary ‘1’	3.5 Volts	1.5 Volts

	CANH	CANL	Receiver
Binary ‘0’	2.5 V	2.5 V	2.5 – 2.5 = 0V Binary ‘0’
Binary ‘1’	3.5 V	2.5 V	3.5 – 1.5 = 2 V = Binary ‘1’

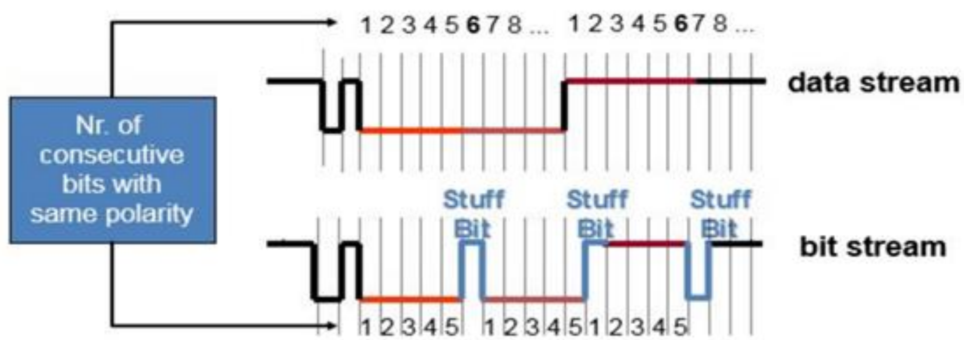
Noise cancellation in CAN Bus

- When external noise gets injected onto the bus, it affects equally on both CanH and CanL.
  - When receiver performs subtraction, undesired change in delta value will get cancelled out and hence originally transmitted logic levels will be decoded

	CANH	CANL	Receiver
Binary '0'	2.5 V + Δ	2.5 V+ Δ	(2.5+ Δ) – (2.5+ Δ) = 0V Binary '0'
Binary '1'	3.5 V+ Δ	2.5 V+ Δ	(3.5 + Δ)– (1.5+ Δ) = 2 V = Binary '1'

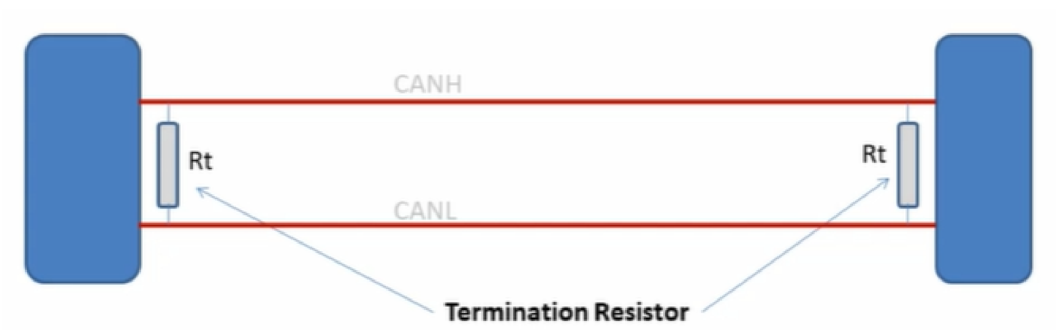
Bit Stuffing

- Bit stuffing ensures enough recessive to dominant edges (Needed due to limitation of NRZ)
- Bit stuffing is the process of inducing inverted bit after five consecutive dominant or recessive bits
  - Stuff bit inserted after 5 consecutive bits of the same state
  - Stuff bit is inverse of previous bit
- De-stuffing done at the receiver



CAN Bus termination resistance

- In every CAN network, there is a resistor connected between CANH and CANL pins of device
  - This resistor is called termination resistor and is placed at the far ends of the network



Purpose of termination resistance

- Purpose of termination resistor
  - According to electrical network theory, the interconnecting cable between source and load exhibits property called characteristic impedance.
    - It is the resistance per unit length of cable
  - When source resistance = load resistance = characteristic impedance, there will be zero losses/No reflections
    - CAN cable characteristic impedance = 120Ohms
    - Therefore Rt = 120Ohms

CAN Frame format

- Two CAN protocol versions available
  - V2.0A (Standard format) → 11 bit message ID → 2048 ID's available
  - V2.0B (Extended format) → 29 bit message ID → 596 million ID's available
- CAN Frame explanation
  - Strat of bit : 1 bit
  - Identifier :11/29 bits
  - Control field
    - Specifies the type of frame
    - Specifies the data length (DLC)

Start of Frame	Identifier 11 bits	Control Field	Data Field (0..8 Bytes)	CRC Field	ACK Field	End of Frame

Start of Frame	Identifier 29 bits	Control Field	Data Field (0..8 Bytes)	CRC Field	ACK Field	End of Frame

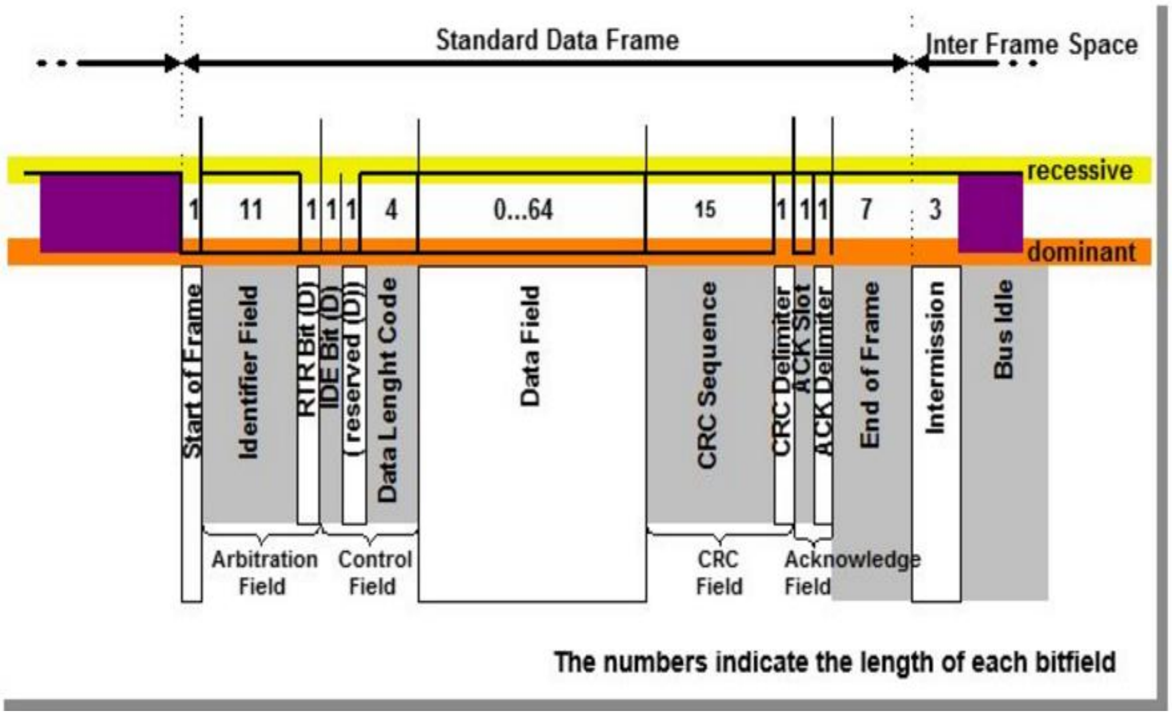
CAN Frame format

- Data field
  - Contains the actual data
- CRC field
  - Will have CRC sequence
  - Will have de-limiter
- ACK field and CRC field is used for error detection purpose
- End of frame
  - Consecutive recessive bits

Start of Frame	Identifier 11 bits	Control Field	Data Field (0..8 Bytes)	CRC Field	ACK Field	End of Frame

Start of Frame	Identifier 29 bits	Control Field	Data Field (0..8 Bytes)	CRC Field	ACK Field	End of Frame

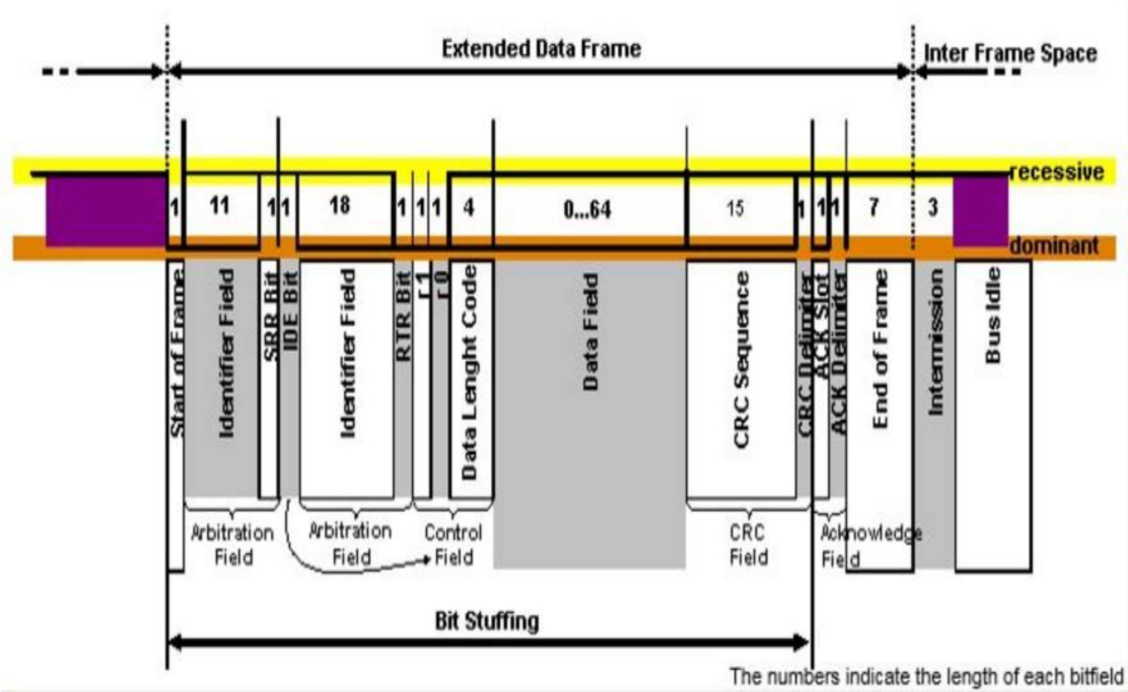
Standard Data Frame



<u>Standard Data Frame</u>

	1	Startbit
+	11	Identifier bits
+	1	RTR bit
+	6	Controlbits
+	64	Databits
+	15	CRC bits
+	1	CRC Delimiter
+	19	Stuffbits*
+	1	ACK Slot
+	1	ACK Delimiter
+	7	EOF bits
+	3	IFS bits
=	130	bits

Extended Data Frame



Extended Data Frame

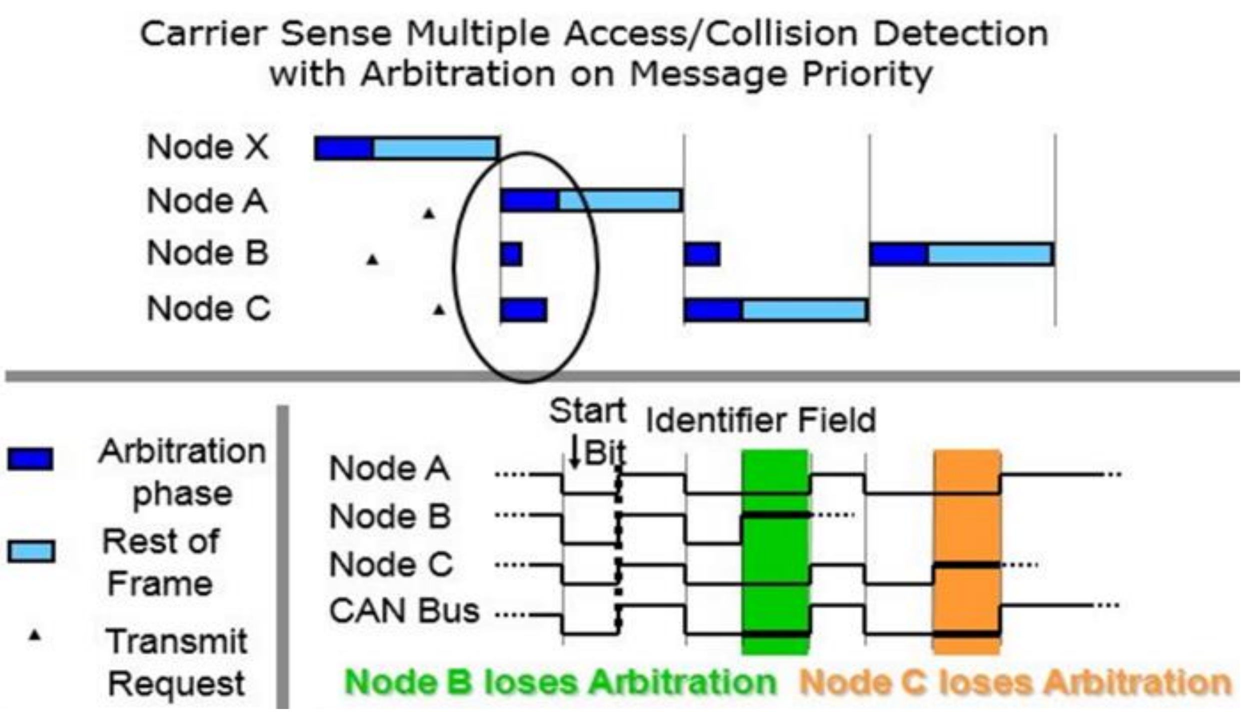
	1	Startbit
+	11	Identifier bits
+	1	SRR bit
+	1	IDE bit
+	18	Identifier bits
+	1	RTR bit
+	6	Controlbits
+	64	Databits
+	15	CRC bits
+	1	CRC Delimiter
+	23	Stuffbits*
+	1	ACK Slot
+	1	ACK Delimiter
+	7	EOF bits
+	3	IFS bits
=	154	bits

NANOCHIP SOLUTIONS

RV-SKILLS CENTER FOR EMERGING TECHNOLOGIES

Message transfer - Bus arbitration

- Carrier Sense Multiple Access/Collision Detection with arbitration on message priority



Example: Bus arbitration

	ID	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
T1	0x20F	0	1	0	0	0	0	0	1	1	1	1
T2	0x29F	0	1	0	1	0	0	1	1	1	1	1
T3	0x20D	0	1	0	0	0	0	0	1	1	0	1
			T2 drops out							T1 drops out		
BUS	0x20D	0	1	0	0	0	0	0	1	1	0	1

T3 wins arbitration and successfully broadcasts its message

Message transfer - Bus arbitration

- Standard and extended frames may exist on the same bus, and even have numerically equivalent identifiers.
  - In this case, the standard frame will have the higher priority.

Types of CAN frames

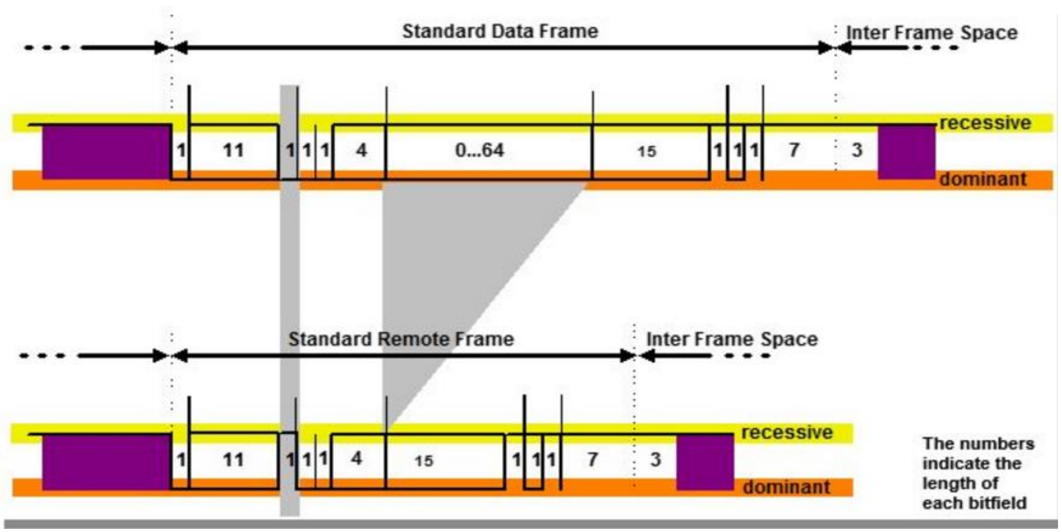
- Data frame
- Remote frame
- Error frame
- Overload frame

Types of CAN frames

- Data frame
  - Data frame is the frame to broadcaste available information on bus
  - RTR bus is dominant (“0”)
- Remote frame
  - Purpose of remote frame is to request desired information from other nodes of network
  - RTR bit is recessive (“1”)
  - No data field
  - When RTR=‘1’, all nodes in the network compare message identifier value to see if they are supposed to respond to the request frame

Types of CAN frames

- Data frame and Remote frame

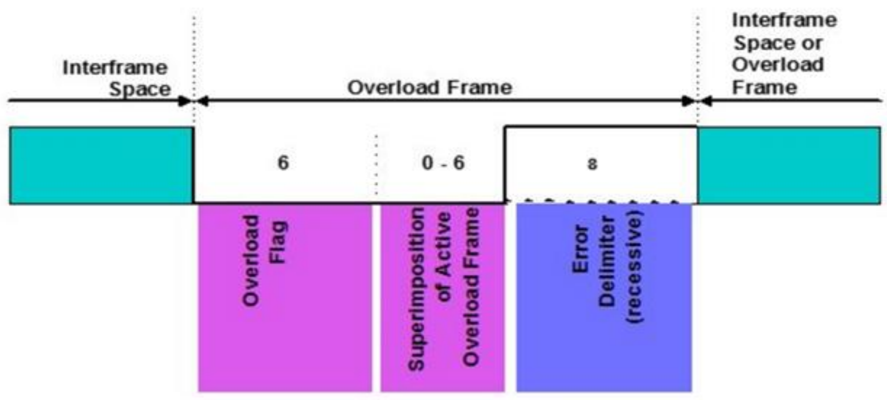


Types of CAN frames

- Error frame
  - When any node in the network detects error in the network, it sends an error frame.
  - Error frame consists of 6 dominant low bits followed by 8 recessive high bits

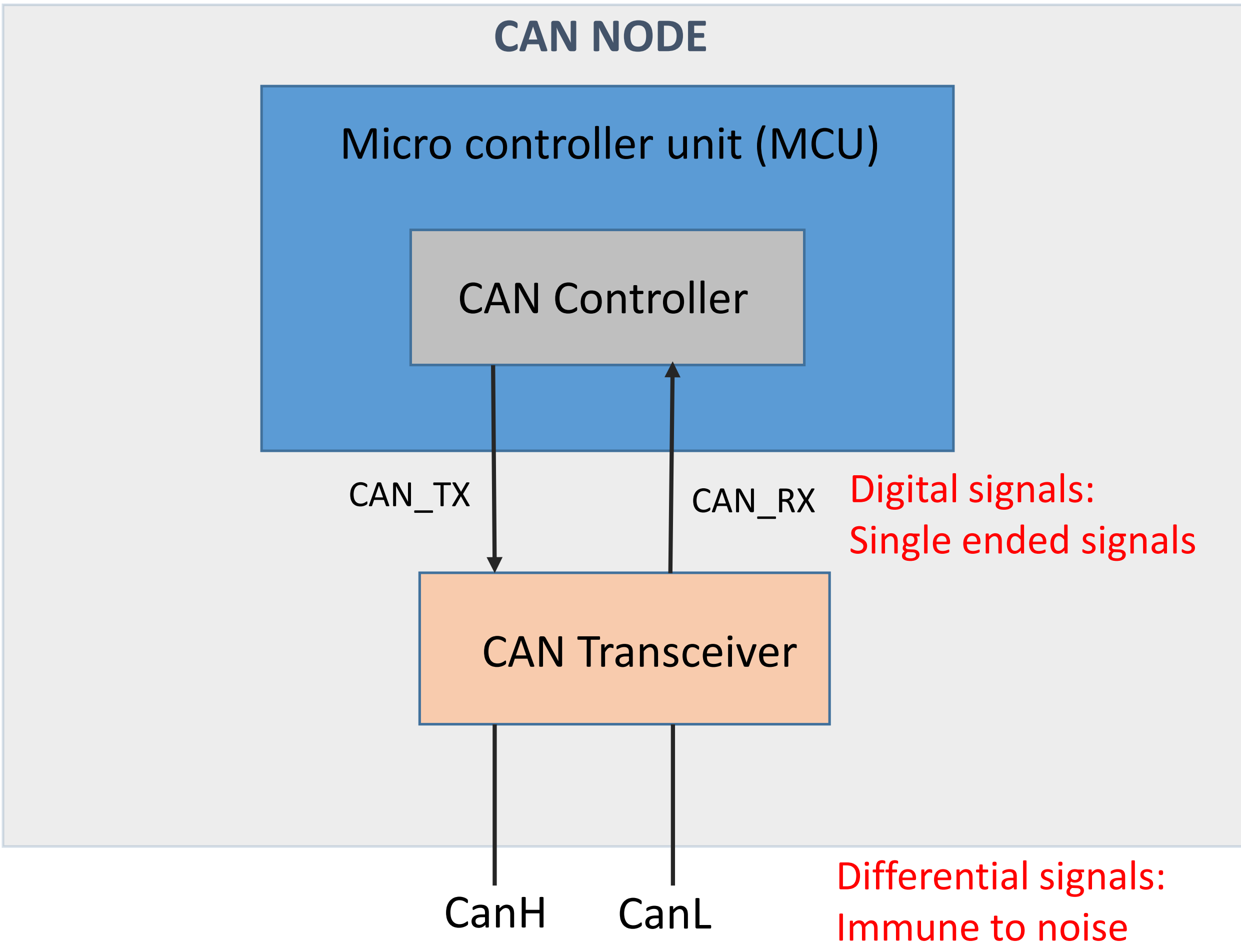
Types of CAN frames

- Overload frame
  - When CAN bus is overload with many messages, to avoid collision and to ensure proper communication, CAN controller will send overload frame



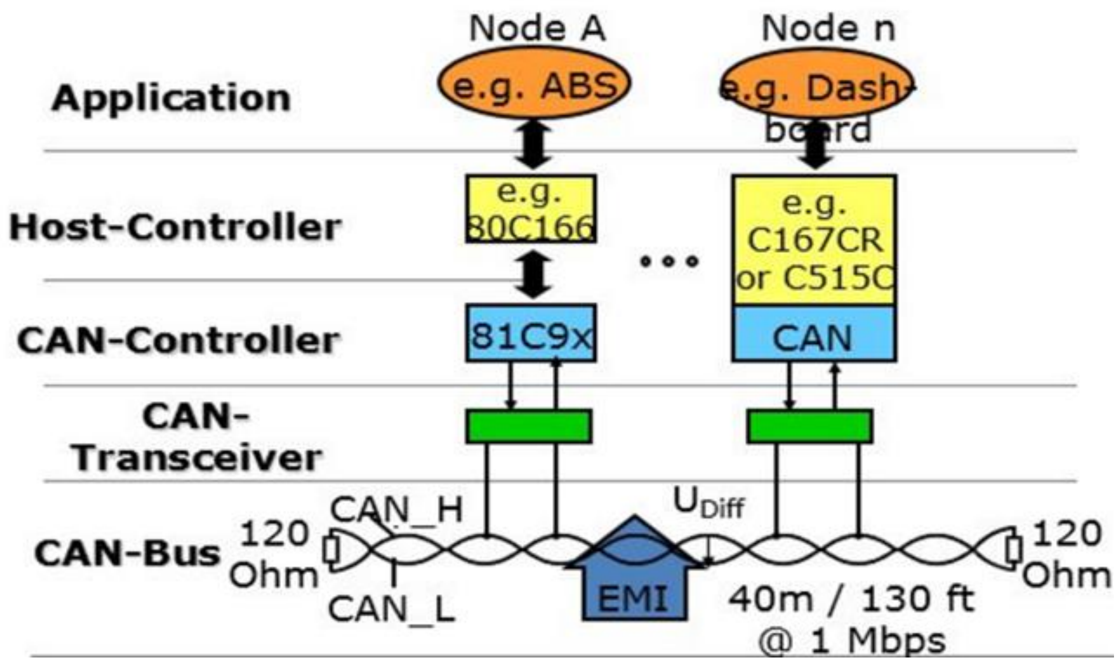
CAN Node

- CAN Node block diagram

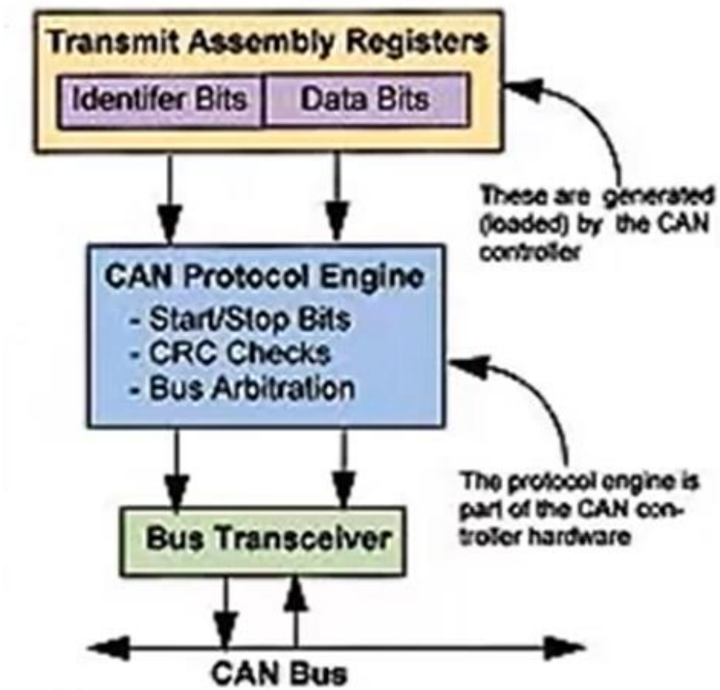


Example of CAN node network

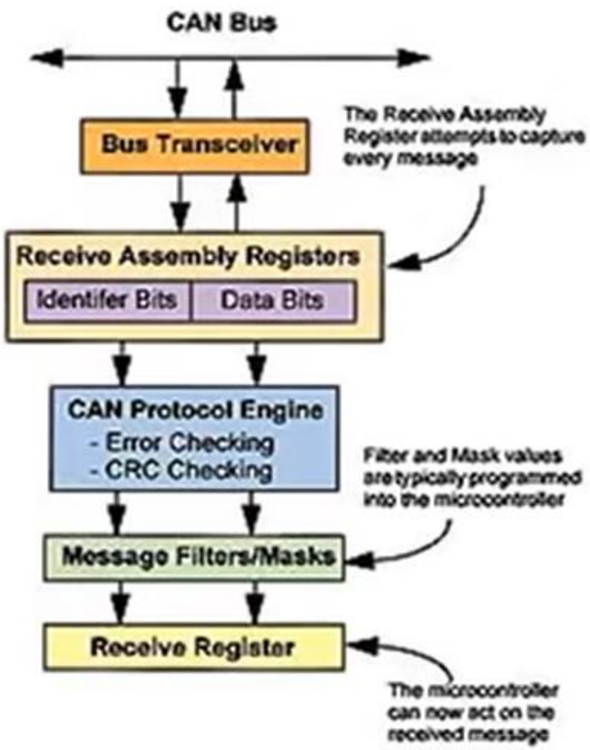
- Example of CAN node network



Message transmit and receive



Message Transmit



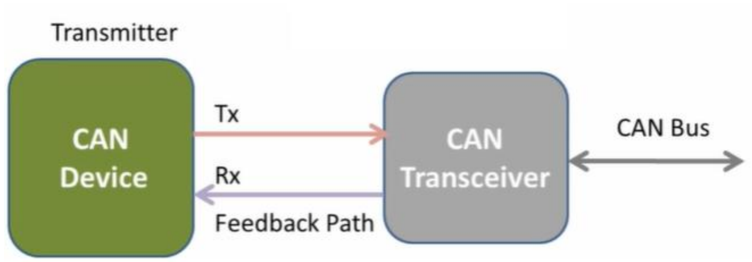
Message Receive

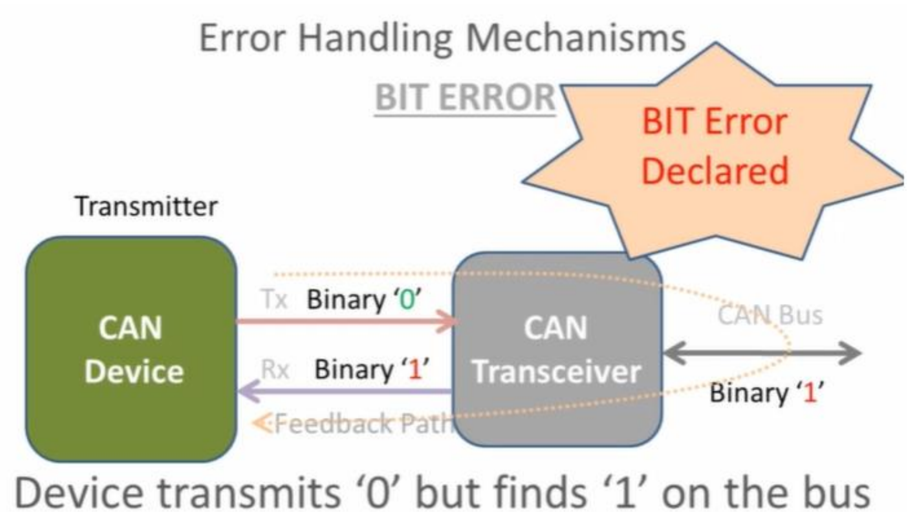
Error handling mechanisms

- CAN protocol has following error handling mechanisms
  - Bit level errors
    - Bit error
    - Bit stuffing error
  - Message or frame level errors
    - Acknowledgement error
    - CRC error
    - Form/Frame error

Bit error

- If transmitted bit is not same as logic level on the bus
  - A transmitted bit must be correctly read back from the CAN bus. Otherwise bit error





Bit stuffing error

- This will help to check if bus is permanently shorted to power supply or ground
- No CAN device can send more than five consecutive bits of same value (Dominant or recessive)
- If receiver finds any violation of bit stuffing rule, then bit stuffing error is declared

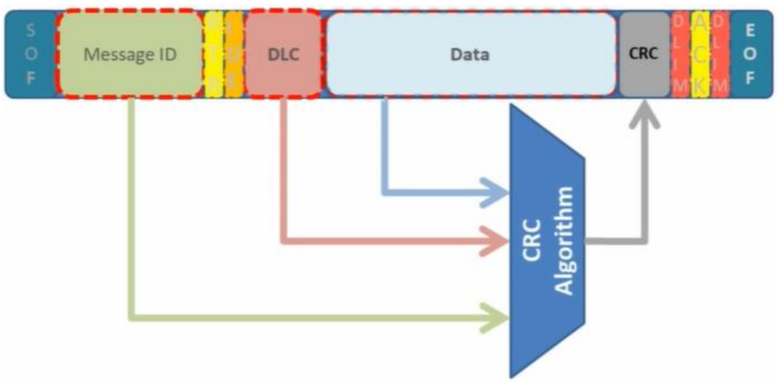
Acknowledgement error

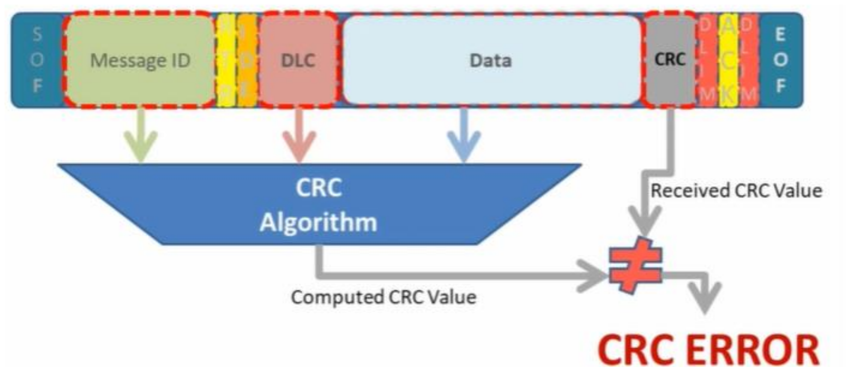
- ACK bit functions as handshake or confirmation signal from receiver to transmitter
- Receiver is expected to pull ACK bit to 0
- If pull down signal not found, transmitter declares acknowledgement error
- This happens in two cases
  - No other node except transmitter in the network
  - Other nodes fail to detect the transmitted message

CRC error

- CRC → Cyclic redundancy Check
- CRC value in CAN frame is 16 bits in length
- CRC is an algorithm
  - Unique number calculated from different fields of CAN message
  - CRC is computed both at transmitter and receiver
  - CRC
- Used to confirm if information carried by CAN message is corrupted

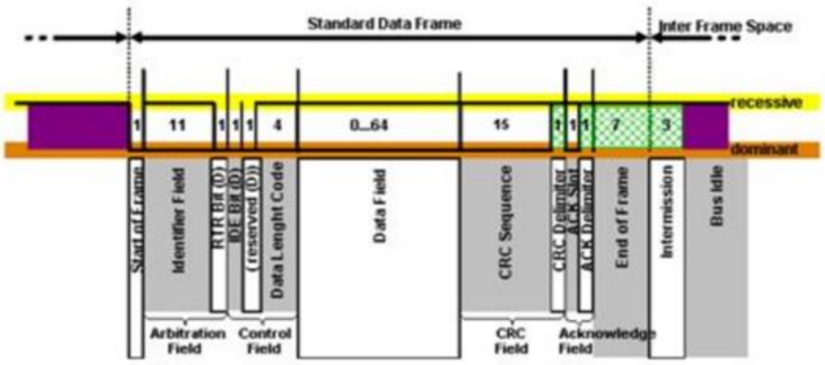
CRC @ Transmitter and Receiver



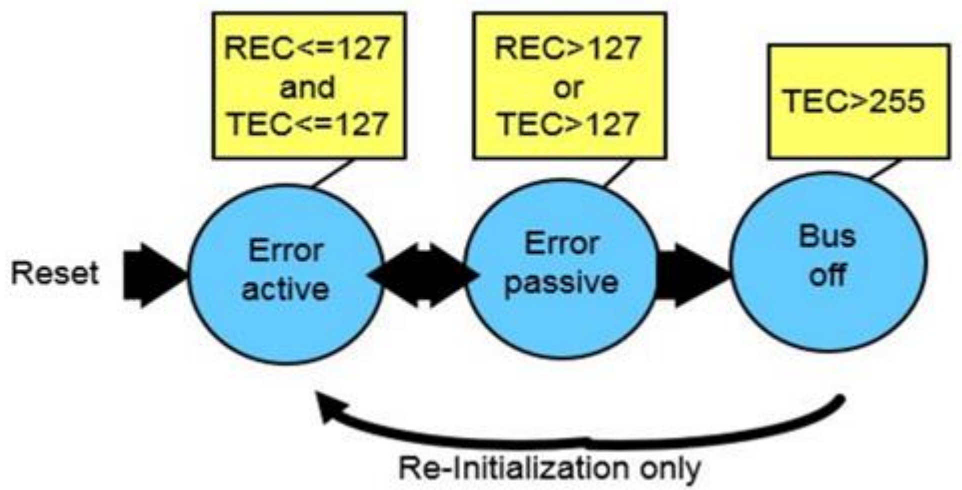


Frame error

- No dominant bits allowed in
  - CRC delimiter
  - ACK delimiter
  - EOF
  - IFS
- If found, then we get frame error



Error Handling



Thank You