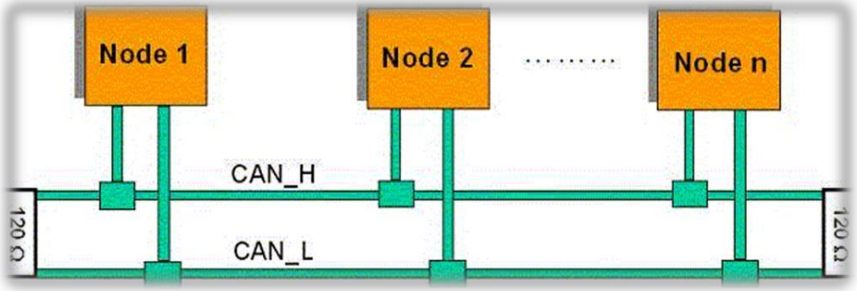
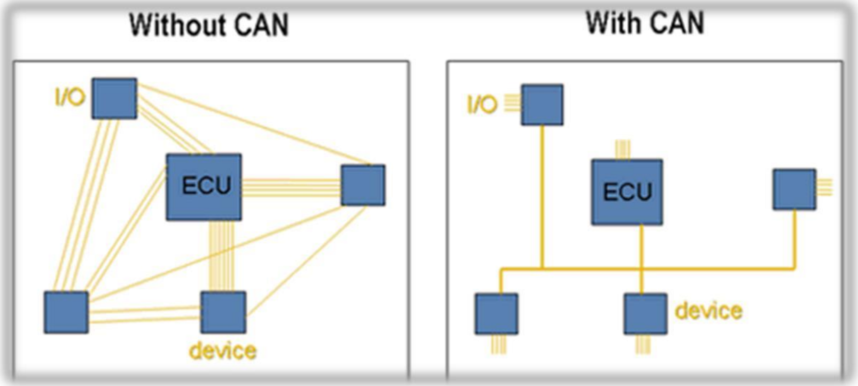
CAN BUS IN TIVA C

MADE BY: AHMED MOHAMED AHMED ABD EL-HAMED

CAN OVERVIEW

- CAN (Controller Area Network) is the most widely used in-vehicle network. It is Multi master Asynchronous Serial Communications Protocol developed by Robert
- CAN is a fast serial bus designed to provide an efficient, reliable and very economical link between sensors and actuators.
- CAN used for reliable data exchange between electronic units(ECUs) in the automobile
- All nodes can send a message at any time, when two nodes are accessing the bus together, arbitration decides who will continue.
- The CAN has several Error Detection Mechanism :
  - Check if the CRC in the frame is identical to CRC computed
  - The structure of frame is valid
  - Each ECU detects an error sends an error flag that notify all ECUs on the bus to be aware of the transmission error.





CAN FEATURES IN TIVA C

- We have to modules of CAN in Tiva c (CAN0, CAN1)
- Each CAN module has two interfaces for read and write message object
- Its better to use one interface for receiving and one for transmitting
- Message Ram with 32 message object
- Bitrate up to 1Mbps
- Test mode
- Programmable loopback mode for self-test operation
- Maskable interrupt
- Programmable FIFO mode enables storage of multiple message objects

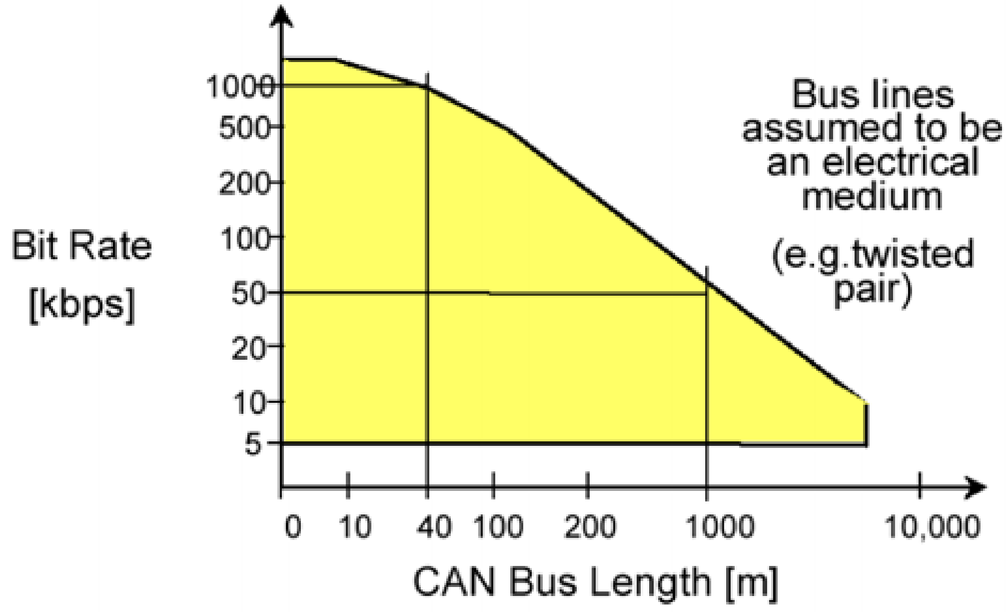
TYPES OF CAN ID

- Standard Identifier:
  - The Standard CAN 11-bit identifier field provides for 2^11, or 2048 different message identifiers.
- Extended Identifier:
  - the Extended CAN 29-bit identifier provides for 229, or 537 million different message identifiers.

NOMENCLATURE	STANDARD	MAX. SIGNALING RATE	IDENTIFIER
Low–Speed CAN	ISO 11519	125 kbps	11-bit
CAN 2.0A	ISO 11898:1993	1 Mbps	11-bit
CAN 2.0B	ISO 11898:1995	1 Mbps	29-bit

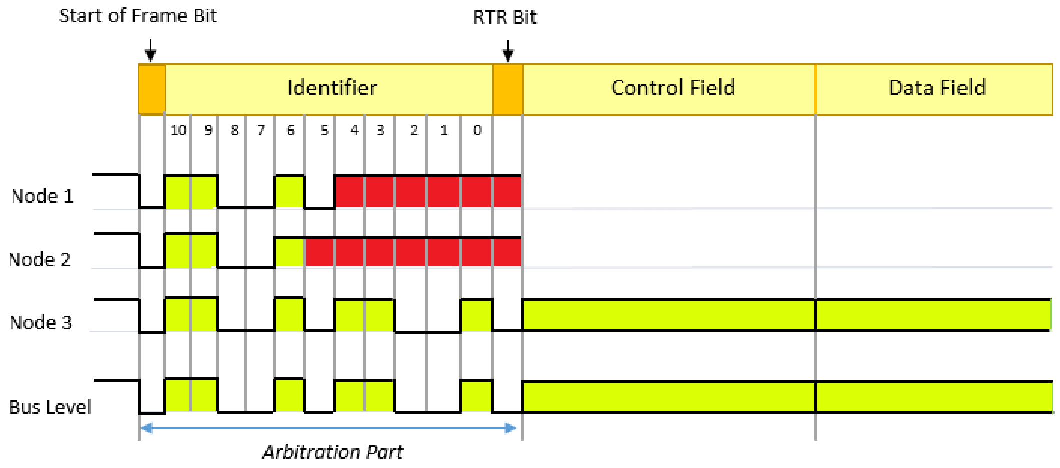
BIT-RATE & BUS LENGTH

- The bit-rate of CAN inversely proportional with the Bus length
    - 1000 kbps -> 40 meters
    - 500 Kbps -> 100 meters
    - 250 Kbps -> 200 meters
    - 125 Kbps -> 500 meters
    - 50 Kbps -> 1000 meters



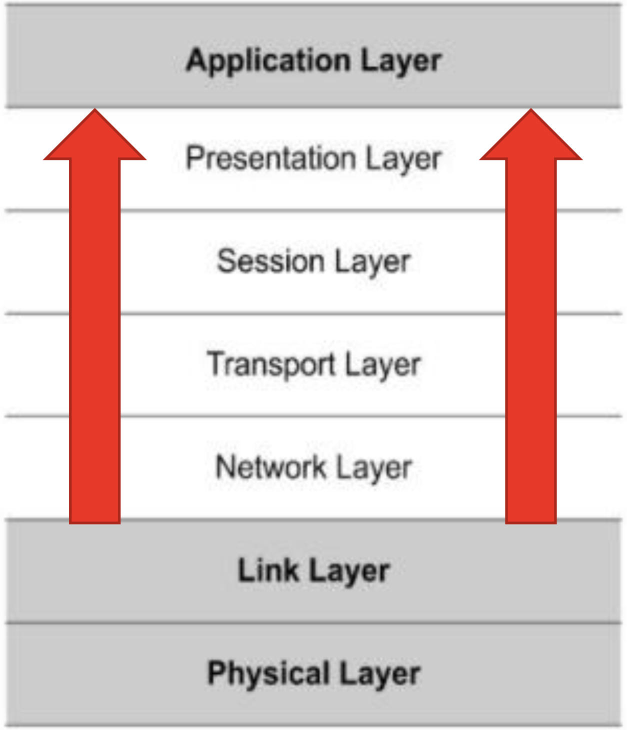
CAN BUS ARBITRATION

- If two messages are sent at the same time over CAN Bus, the Bus take the logical AND of the signals.
    - So, the message identifier with the lowest binary number get the highest priority
    - Every device listens to the bus and backs off when it notice a mismatch the bus's bits and its identifier.



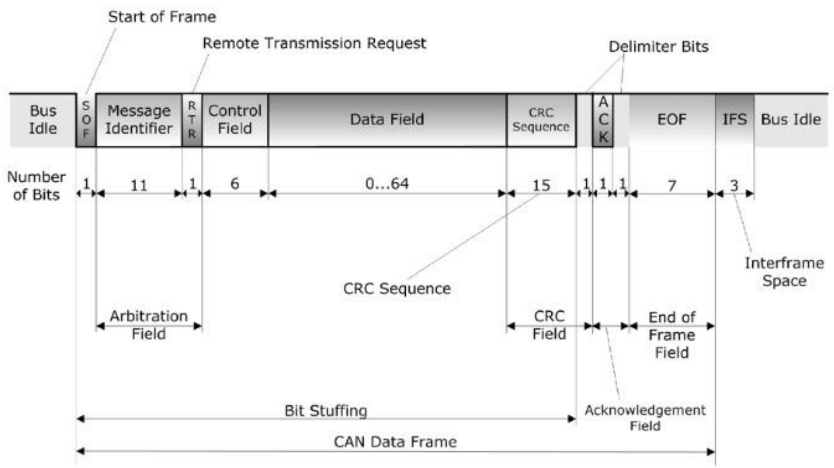
CAN LAYERS

- CAN is a closed network
  - No need for security, sessions or logins
  - No user interface requirements
- Physical Layers:
  - Transmission line parameters, signaling levels, transmission speed…
- Link Layer :
  - Medium access control
  - Frame coding and decoding
  - Addressing
  - Data security
  - Error state behavior



FRAMES TYPES

- Data Frame
  - can transport a maximum payload of eight bytes
- Remote Frame
- Error Frame
  - indicate errors detected during communication
- Overload Frame
  - Sent by a node to request a delay in transmission

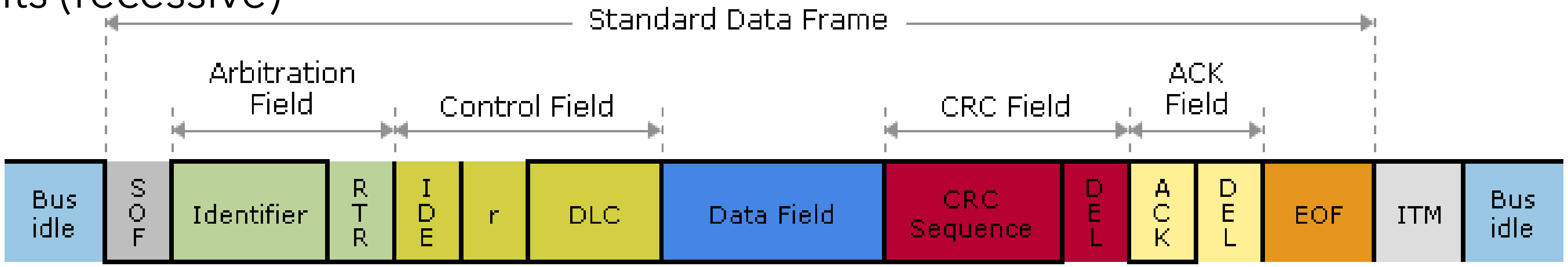


DATA FRAME

- CAN Data frame transfers data from one ECU to others
- Can transport a maximum payload of eight bytes
- A data frame is made up of many different segments. Each individual segment carries out an important task during transmission:
    - Initiate and maintain synchronization
    - Establish communication
    - Transmit and protect user data

DATA FRAME

- Start of Frame – 1-bit
- Arbitration Field – 11-bits/29-bits
- Control Field – 6 bits (2 reserved, 4 representing number of Data Field bytes)
- Data Field – 0 to 8 BYTES
- CRC – 15-bits
- ACK Field – 1-bit/variable
- End of Frame – 7-bits (recessive)

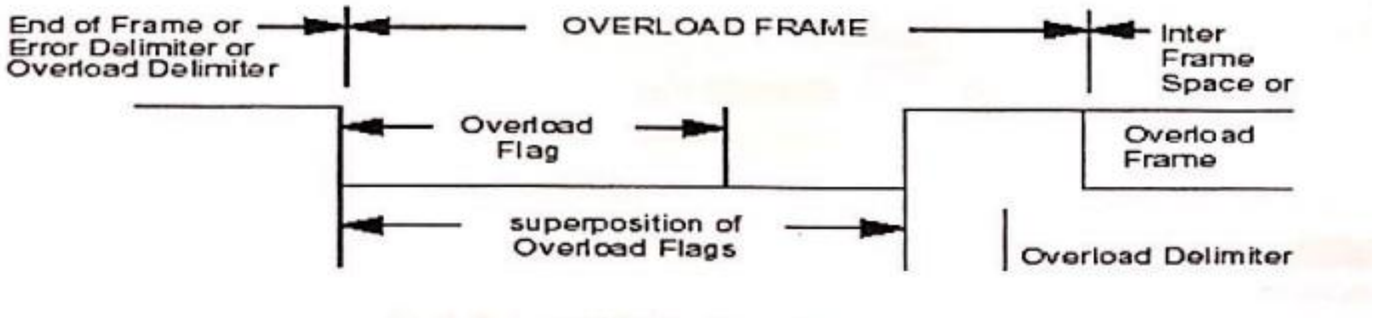


REMOTE FRAME

- Remote frame has the same structure as a data frame but without data
- Data and remote frames are differentiated by the RTR bit (Remote Transmission Request). In the case of data frame, the RTR bit is sent as dominant, but the remote frame is identified by recessive RTR bit.
- A frame type used to request data from any CAN node.
- Remote frames may be defined for all existing data frames in the CAN network. It is only necessary to ensure that the identifiers of the remote frames match those of the associated data frames.
    - CAN controller with object storage, the CAN controller automatically responds to a remote frame.
    - CAN controllers without object storage must let the host know about the remote frame so that it can initiate a response.

OVERLOAD FRAME

- It’s same as error frame
- It sends by the receiver node to request a delay in transmission in case it suffers from data overflow, or it can’t process the received data anymore. It sends the overload frame to delay the next DATA frame or REMOTE frame or when detecting a dominant bit during intermission.



ERROR DETECTION

- Five errors detection implemented in CAN:
  1. Bit monitoring
  2. Frame format checking
  3. CRC
  4. ACK
  5. Bit stuffing

BIT MONITORING

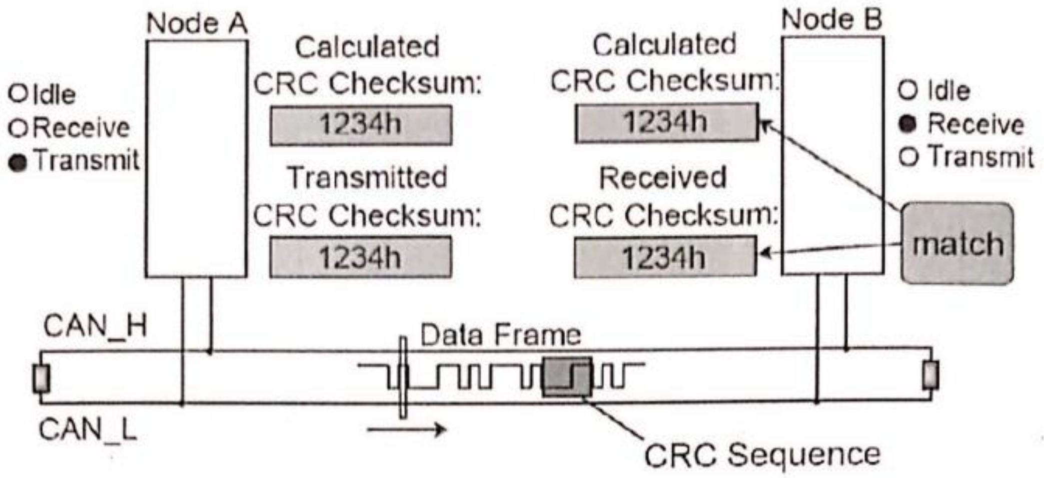
- Responsibility: Sender
- The sender compare the sent bit level with the actual bus level.
- A bit error exists if the sender detects a difference between the two levels

FRAME FORMAT CHECKING

- Responsibility: Receiver
- Comparison of the arriving bit stream with the message format
- Each CAN message always exhibits the same bit sequences at certain positions (CRC delimiter, ACK delimiter, EOF).
- Senders always transmit these message Components recessively. A format error exists if a receiver detects a dominant bus level within one of these message components in the Form Check

CYCLIC REDUNDANCY CHECK (CRC)

- Responsibility: Receiver
- CRC sequence is computed based on the bits to be transmitted.
- Calculated and received CRC checksum must match, otherwise frame was not received correctly (CRC Error).



ACK

- Responsibility: Receiver
- A frame must be acknowledged by at least one other node, otherwise ACK Error
- A receiver acknowledges either positively or negatively. A dominant level in the ACK slot represents a positive acknowledgement, while a recessive level represents a negative acknowledgement. The ACK delimiter is always transmitted recessively.

BIT STUFFING

- Responsibility: Receiver
- Compares every bit placed on the CAN bus with the actual bus level
- Discrepancy indicates a bit monitoring error and results in error handling

FIFO BUFFER

- Received messages with identifiers matching to a FIFO buffer are stored starting with the message object with the lowest message number
- The message object is locked and cannot be written to by the message handler until the CPU has cleared the NEWDAT bit
- Until all the preceding message objects have been released by clearing the NEWDAT bit, all further messages for this FIFO buffer are written into the last message object of the FIFO buffer and therefore overwrite previous messages
- The order of the received messages in the FIFO is not guaranteed

TYPES OF INTERRUPTS

Received a Message Successfully

Transmitted a Message Successfully

Buss off status

Error Passive status

Errors

- Stuff Error
- Format Error
- ACK Error
- Bit Error
- CRC Error
- No Event

Warning Status

TEST MODE

- Silent Mode
  - Silent Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits (Acknowledge Bits, Error Frames).
- Loopback Combined with Silent Mode
  - Loopback Mode and Silent Mode can be combined to allow the CAN Controller to be tested without affecting a running CAN system connected to the CANnTX and CANnRX signals
- Basic Mode
  - Basic Mode allows the CAN Controller to be operated without the Message RAM.
- Loopback Mode
  - Loopback mode is useful for self-test functions. In Loopback Mode, the CAN Controller internally routes the CANnTX signal on to the CANnRX signal and treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into the message buffer.

THANK YOU