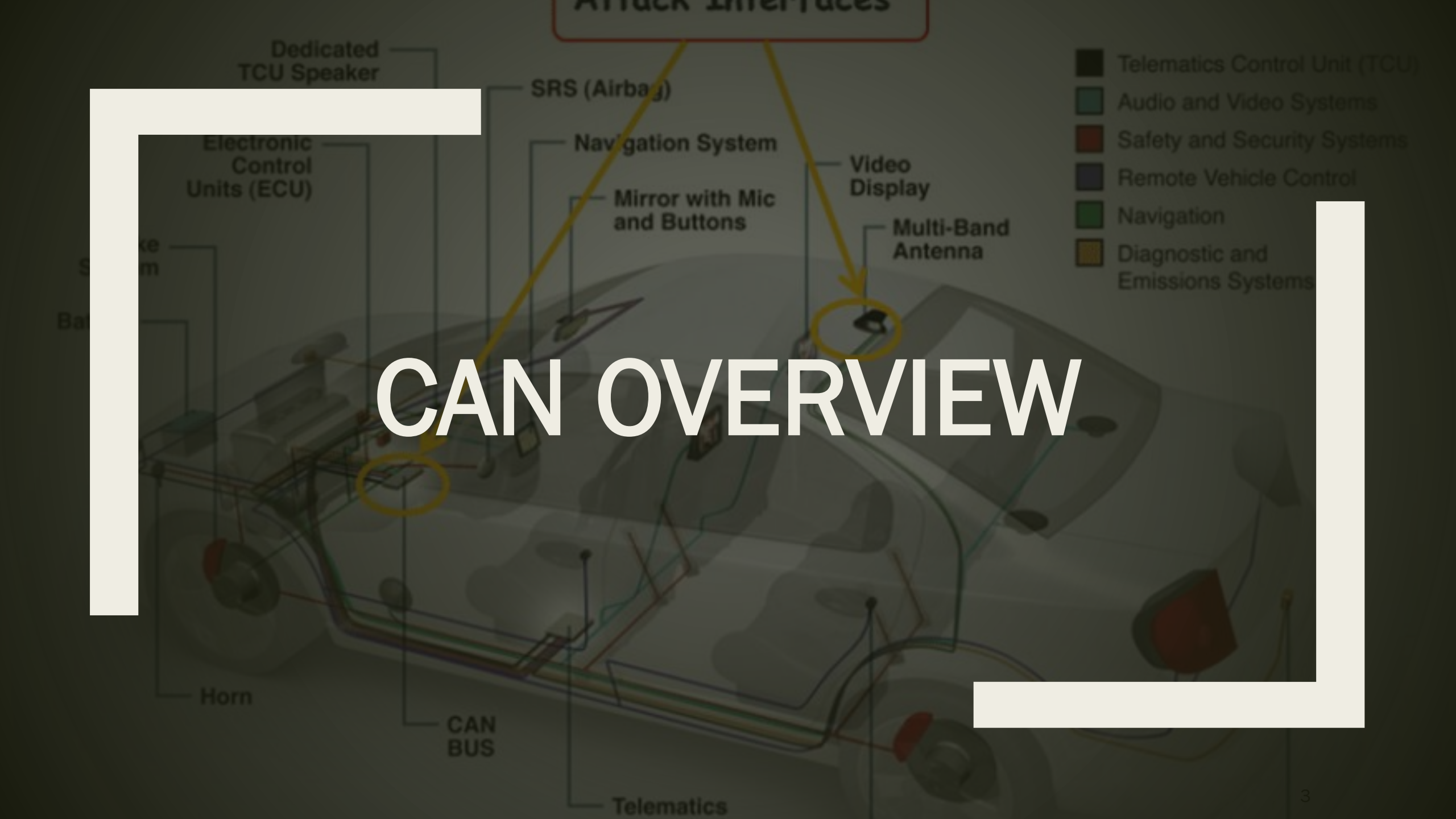
CAN MODULE IN TIVA C

TM4C123GHPM LAUNCHPAD

Prepared by : Basmala Magdy Ali

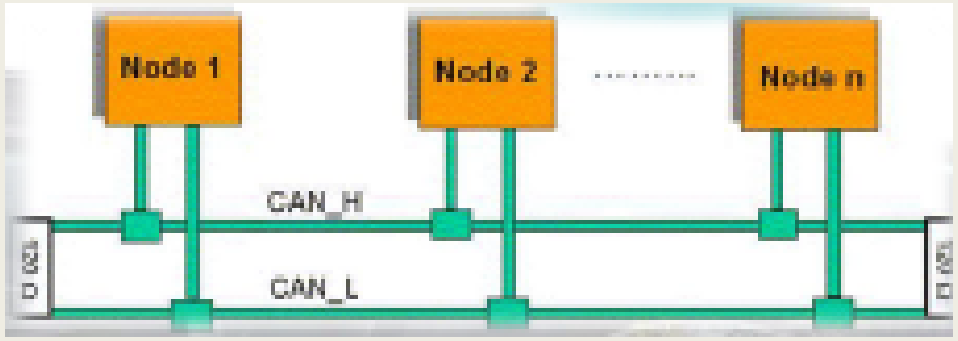
CONTENT

- Can overview
- Can features in Tiva c
- Can layers
- Can block diagram in Tiva c
- Signal description
- Frame types
- Can error detection types
- Test modes types
- Interrupt types
- Can FIFO buffer
- Can driver initialization in Tiva c
- Bit time and Bit Rate



CAN OVERVIEW

CAN (Controller Area Network) : is the most widely used in vehicle network. It is a serial communications protocol developed by Robert.

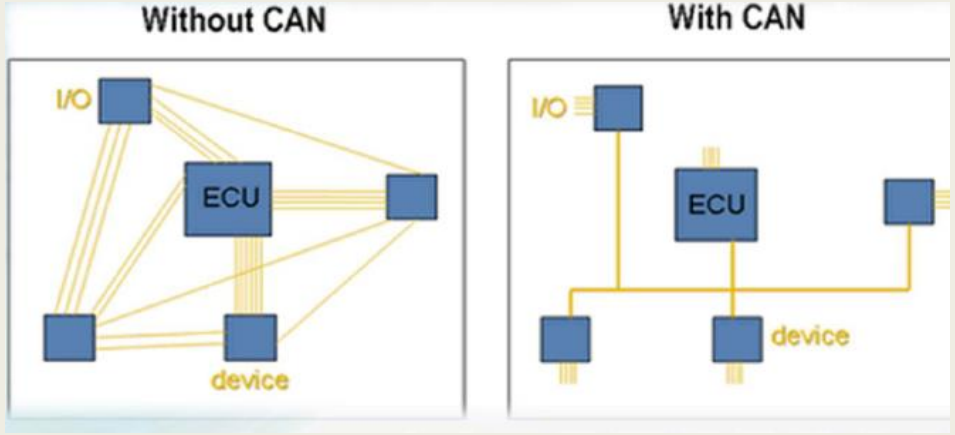


- **Multi-master :** protocol
- **Broadcasting**
- **Asynchronous :** communication (Event Triggered)
- **Serial :** communication technology
- **Priority-based :**bit-wise arbitration
- Variable message priority based on 11-bit
- (or extended 29 bit) packet identifier

In CAN communication, all partners are equal and are able to communicate at any time.

In case of conflicts (two speaking at the same time), arbitrations used to ensure messages are understood.

All nodes can send a message at any time, when two nodes are accessing the bus together, arbitration decides who will continue .



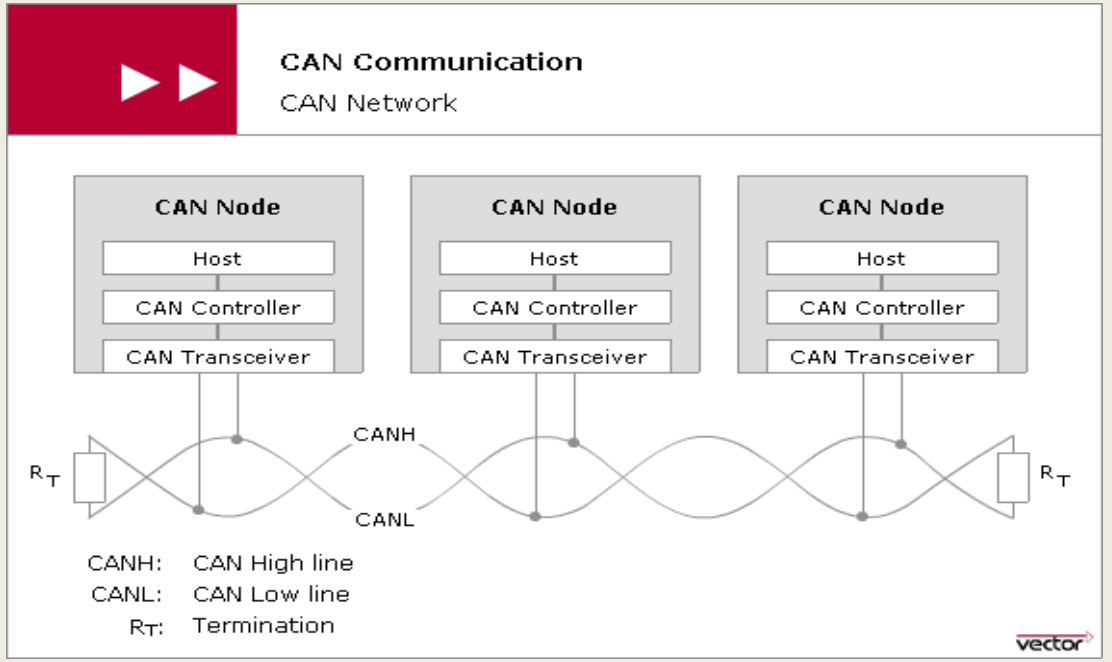
CAN BUS :

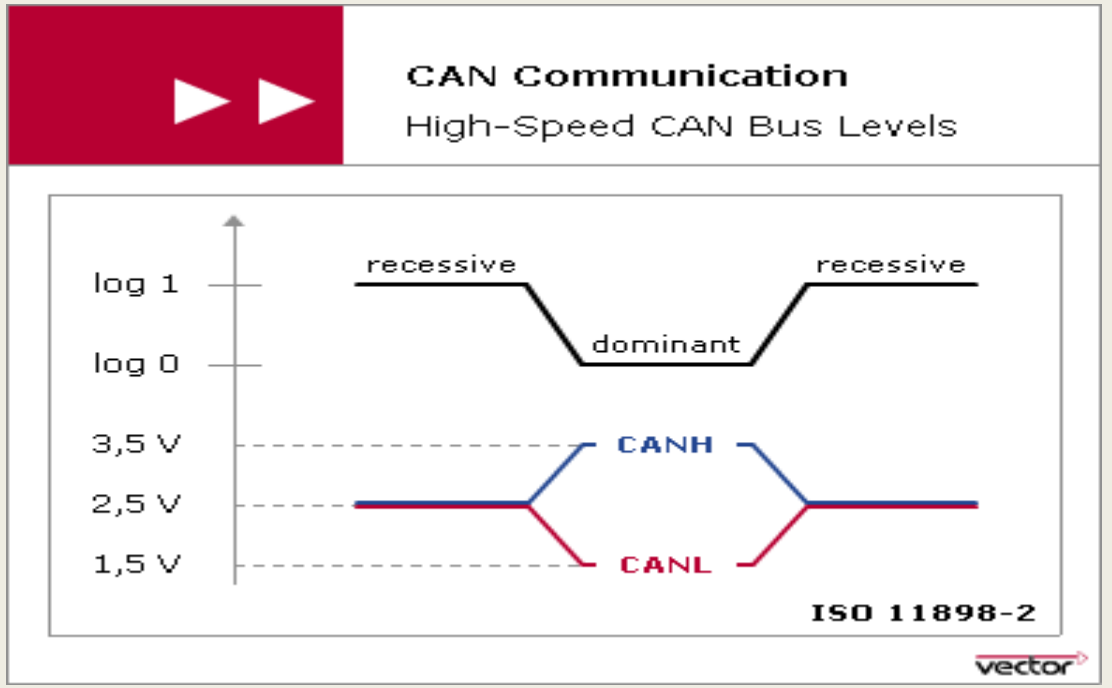
the transmission medium (CAN bus) consists of two lines :

CAN high line (CANH)

CAN low line (CANL).

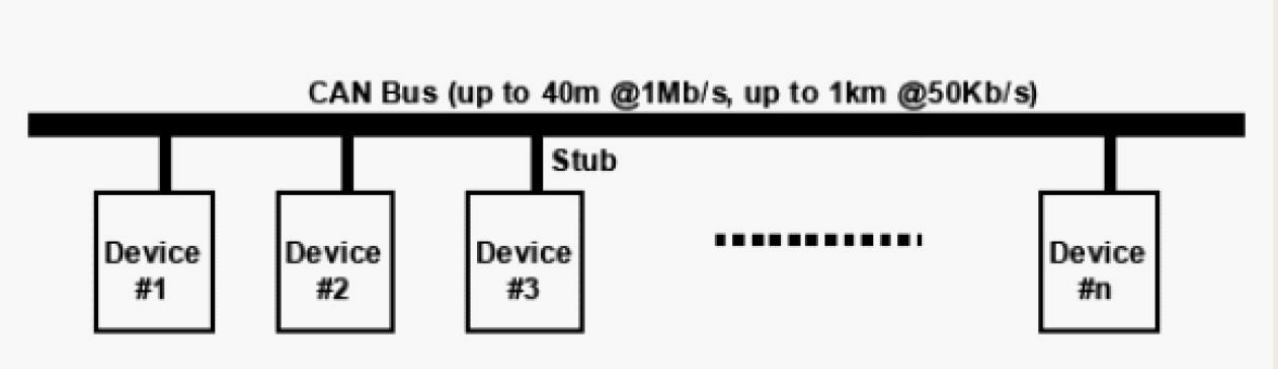
CAN BUS LEVEL





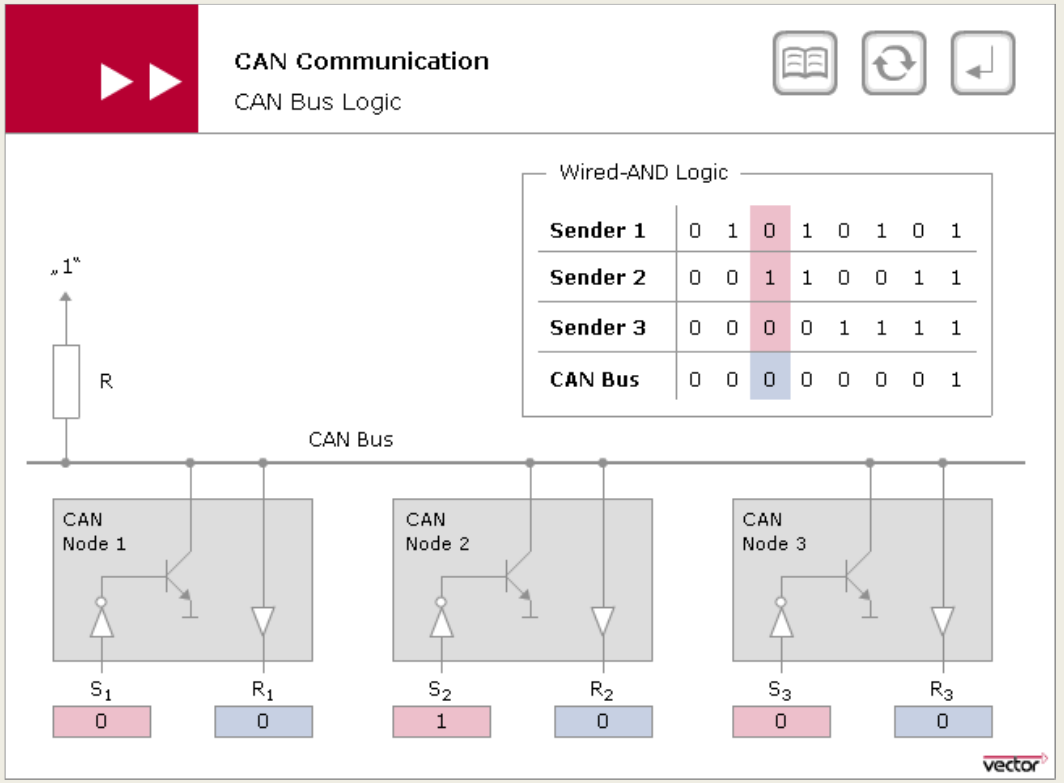
Rate / Bus Length

- ▶1M bit/sec 40 meters (131 feet)
- ▶500K bit/sec 100 meters (328 feet)
- ▶250K bit/sec 200 meters (656 feet)
- ▶125K bit/sec 500 meters (1640 feet)



Bus arbitration

- ❑ The bus takes the "logical AND" of the signal
- ❑ lowest binary number gets the highest priority



CAN FEATURES IN TIVA C

- CAN in Tiva c consist of (CAN0, CAN1)
- CAN protocol version 2.0-part A/B
- 32 message objects with individual identifier masks
- Bit rates up to 1 Mbps
- Maskable interrupt

- Disable Automatic Retransmission mode for Time-Triggered CAN (TTCAN) applications
- Programmable loopback mode for self-test operation
- Programmable FIFO mode enables storage of multiple message objects
- Gluelessly attaches to an external CAN transceiver through the CANnTX and CANnRX signals

CAN LYARES

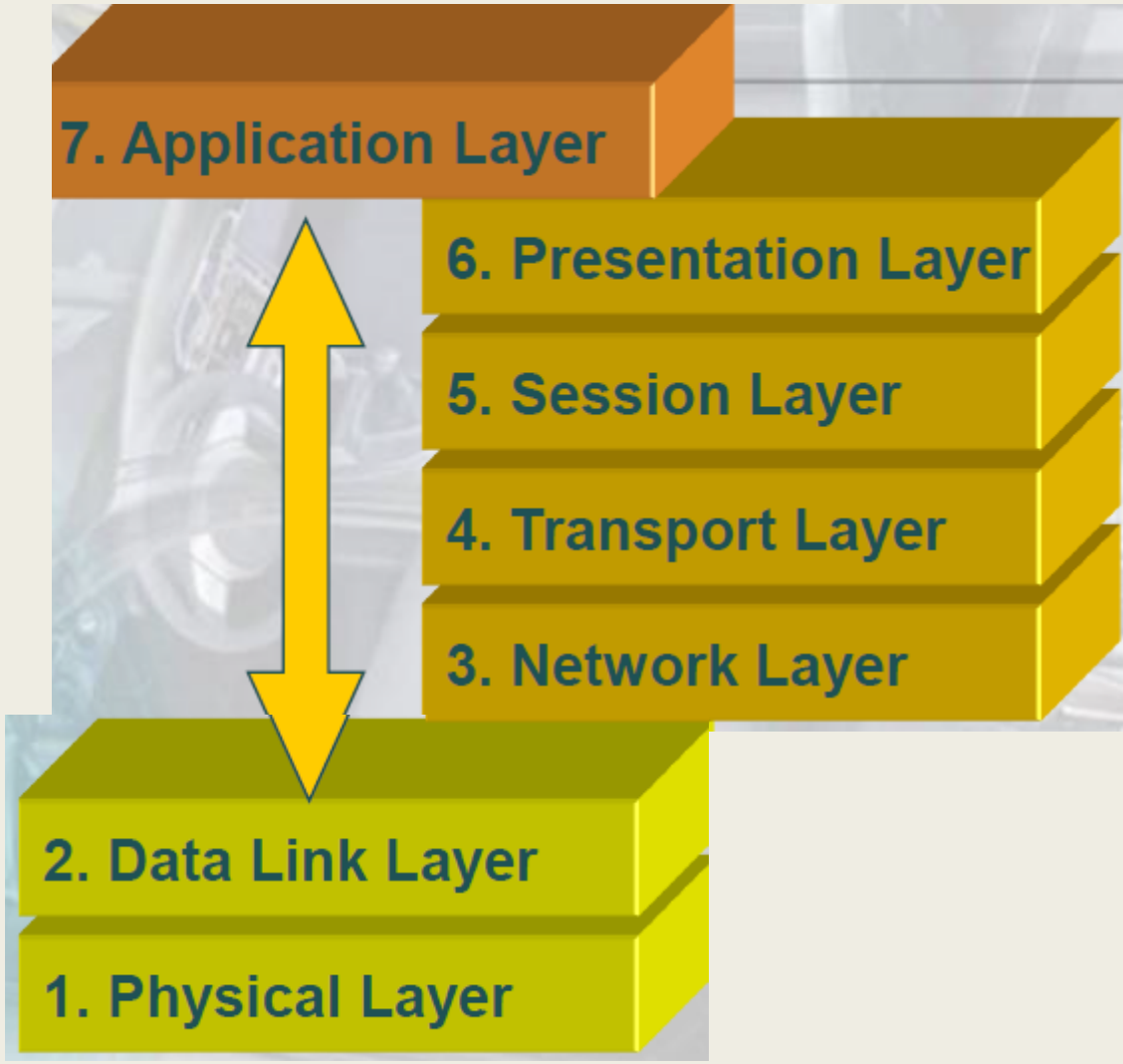
- **The standard CAN implementation bypasses the connection between the Data Link layer and the Application layer.**

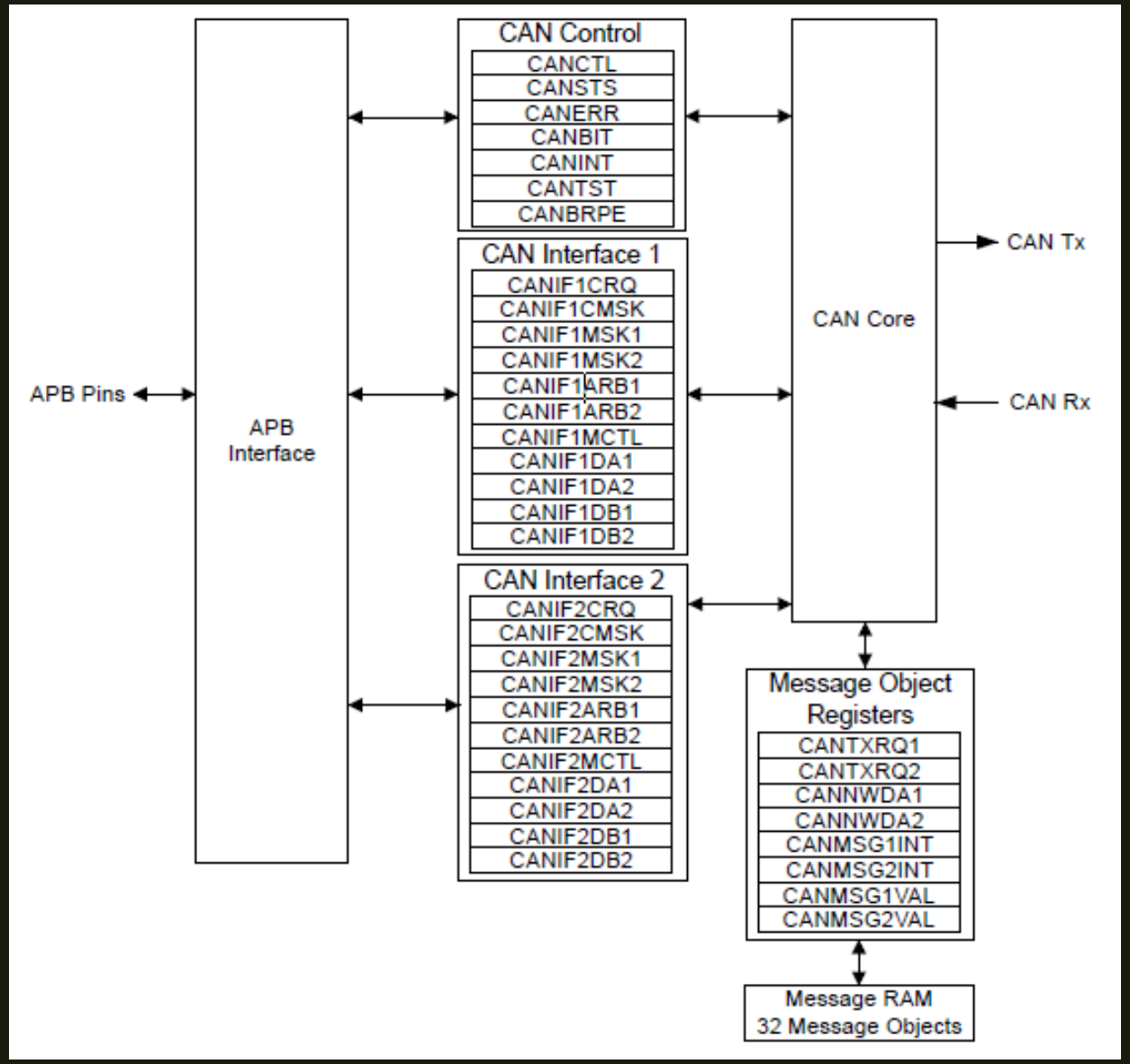
- **The layers above the Data Link Layer are implemented in software which as per definition are called the Higher Layer Protocol**

➢ Physical Layers:
Transmission line parameters, signaling levels, transmission speed…

➢ Link Layer :

- Medium access control
- Frame coding and decoding
- Addressing
- Data security
- Error state behavior



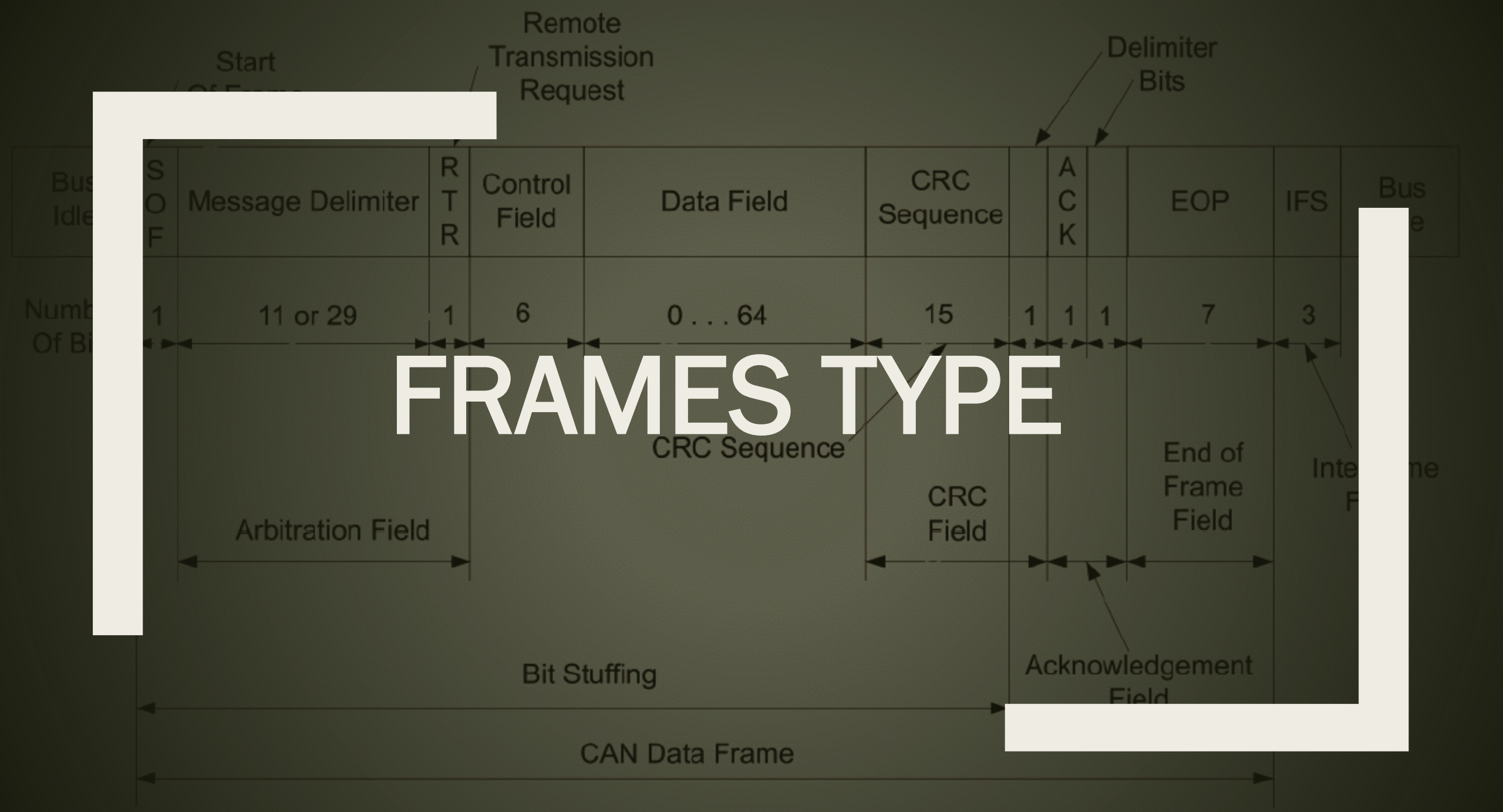


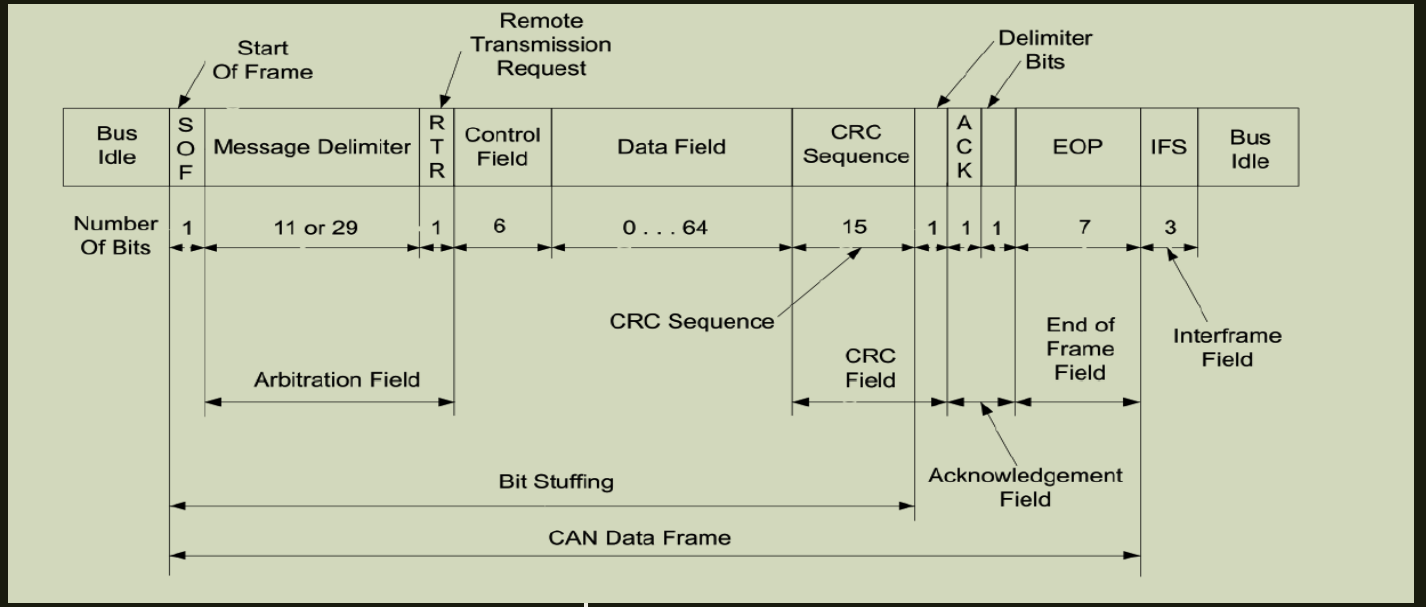
CAN BLOCK DIAGRAM TIVA C

SIGNAL DESCRIPTION

Pin Name	Pin Number	Pin Mux / Pin Assignment	Pin Type	Buffer Type[a]	Description
CAN0Rx	28 58 59	PF0 (3) PB4 (8) PE4 (8)	I	TTL	CAN module 0 receive.
CAN0Tx	31 57 60	PF3 (3) PB5 (8) PE5 (8)	O	TTL	CAN module 0 transmit.
CAN1Rx	17	PA0 (8)	I	TTL	CAN module 1 receive.
CAN1Tx	18	PA1 (8)	O	TTL	CAN module 1 transmit.

FRAMES TYPE


Start
Remote
Transmission
Request
Delimiter
Bits
Bus
Idle
S
O
F
Message Delimiter
R
T
R
Control
Field
Data Field
CRC
Sequence
A
C
K
EOP
IFS
Bus
Numb
Of Bi
1
11 or 29
1
6
0 . . . 64
15
1
1
1
7
3
CRC Sequence
End of
Frame
Field
Inte
F
CRC
Field
Arbitration Field
Bit Stuffing
Acknowledgement
Field
CAN Data Frame



Data Frame

- can transport a maximum payload of eight bytes

Remote Frame

remote frame has the same structure as a data frame.

Error Frame

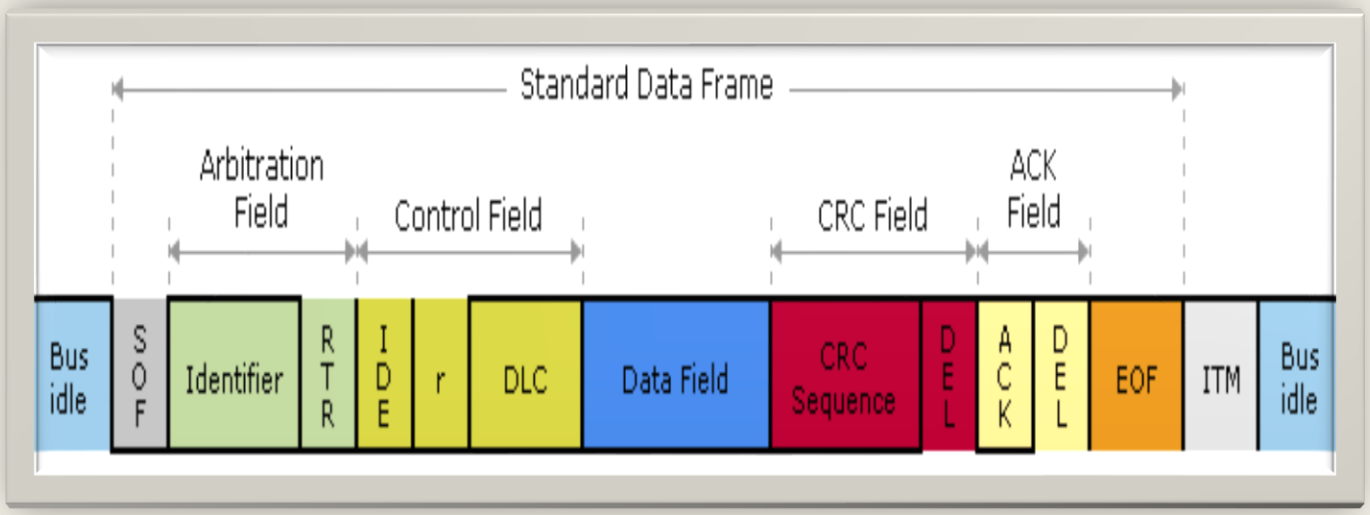
- indicate errors detected during communication

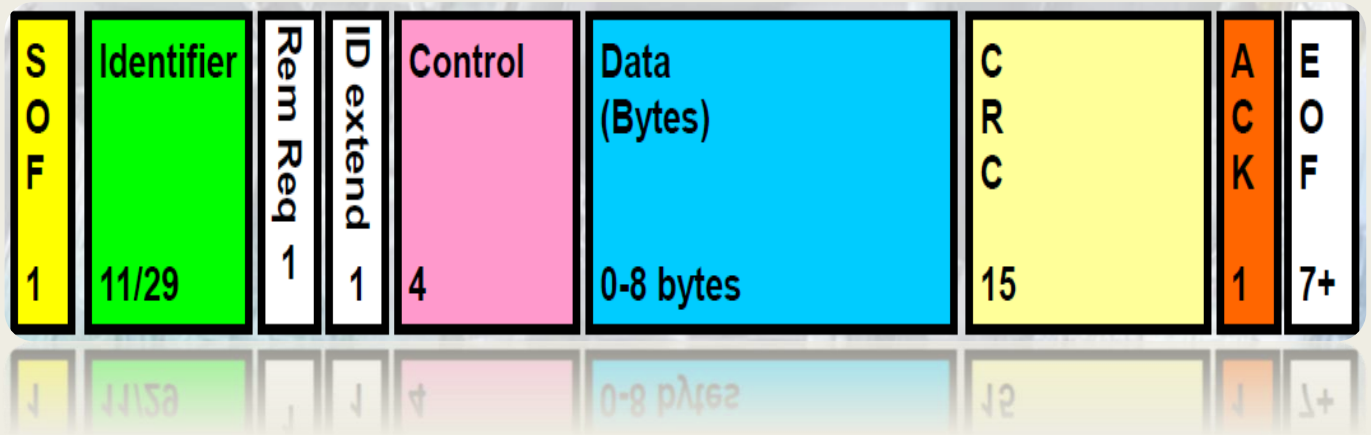
Overload Frame

- Sent by a node to request a delay in transmission

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

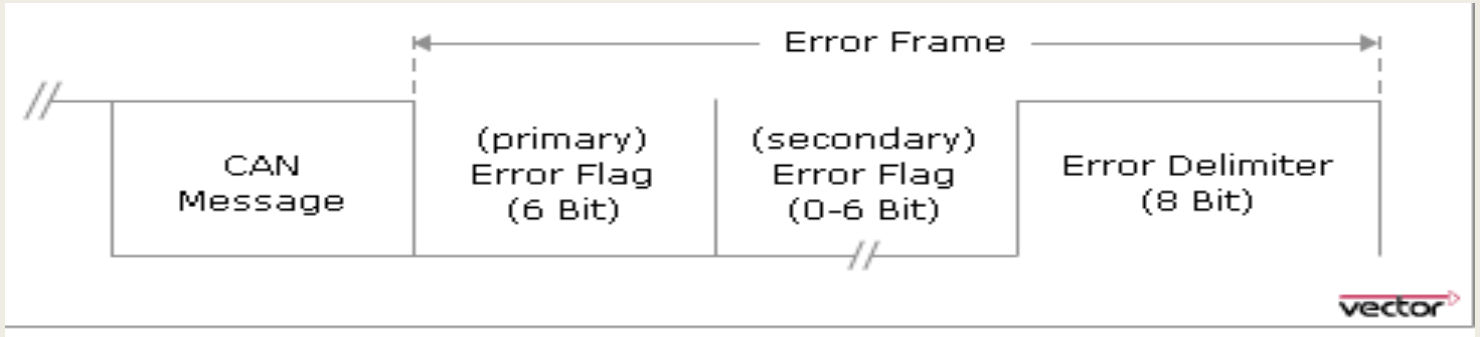
- a frame type used to request data, i.e. data frames, from any CAN node.
- In the case of a **CAN controller with object storage**, the CAN controller automatically responds to a remote frame . **CAN controllers without object storage** must let the host know about the remote frame so that it can initiate a response.

The Error Frame

- the Error Frame is a special message that violates the framing rules of a CAN message.
- It is transmitted when a node detects a fault and will cause all other nodes to detect a fault –so they will send Error Frames, too.
- The transmitter will then automatically try to retransmit the message

The Overload Frame

- It is very similar to the Error Frame with regard to the format and it is transmitted by a node that becomes too busy.



CAN ERROR DETECTION

- Bit monitoring
- Frame format checking
- CRC
- ACK
- Bit stuffing

Bit Monitoring

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

Stuff Check

- Receiver Task
- Compares arriving bit stream for a sequence of six homogeneous bits.
- Detection of a sixth homogeneous bit indicates bit stuffing error and results in error handling

From Check

- Receiver Task
- Comparison of the arriving bit stream with the message format
- Detection of a dominant delimiter bit (CRC delimiter, ACK delimiter) or a dominant bit within EOF indicates a format error and results in error handling

Cyclic Redundancy Check

- Receiver Task
- Utilizes the arriving bit stream and generator polynomial for the Cyclic Redundancy Check defined in ISO 11898-1
- Detection of a CRC error results in error handling

ACK Check

- Sender Task
- Acknowledge error (ACK error) is detected if the recessive level placed by the sender is not overwritten
- Detection of an ACK error results in error handling

TEST MODES TYPES

- **Test Mode**

  A Test Mode is provided which allows various diagnostics to be performed.

- **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 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.

- **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. In this mode, the CANnRX signal is disconnected from the CAN Controller and the CANnTX signal is held recessive.

- **Basic Mode**

  Basic Mode allows the CAN Controller to be operated without the Message RAM. In Basic Mode,The CANIF1 registers are used as the transmit buffer.

Transmitted a Message Successfully

Buss off status

Received a Message Successfully

TYPES OF INTERRUPTS

- Warning Status

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

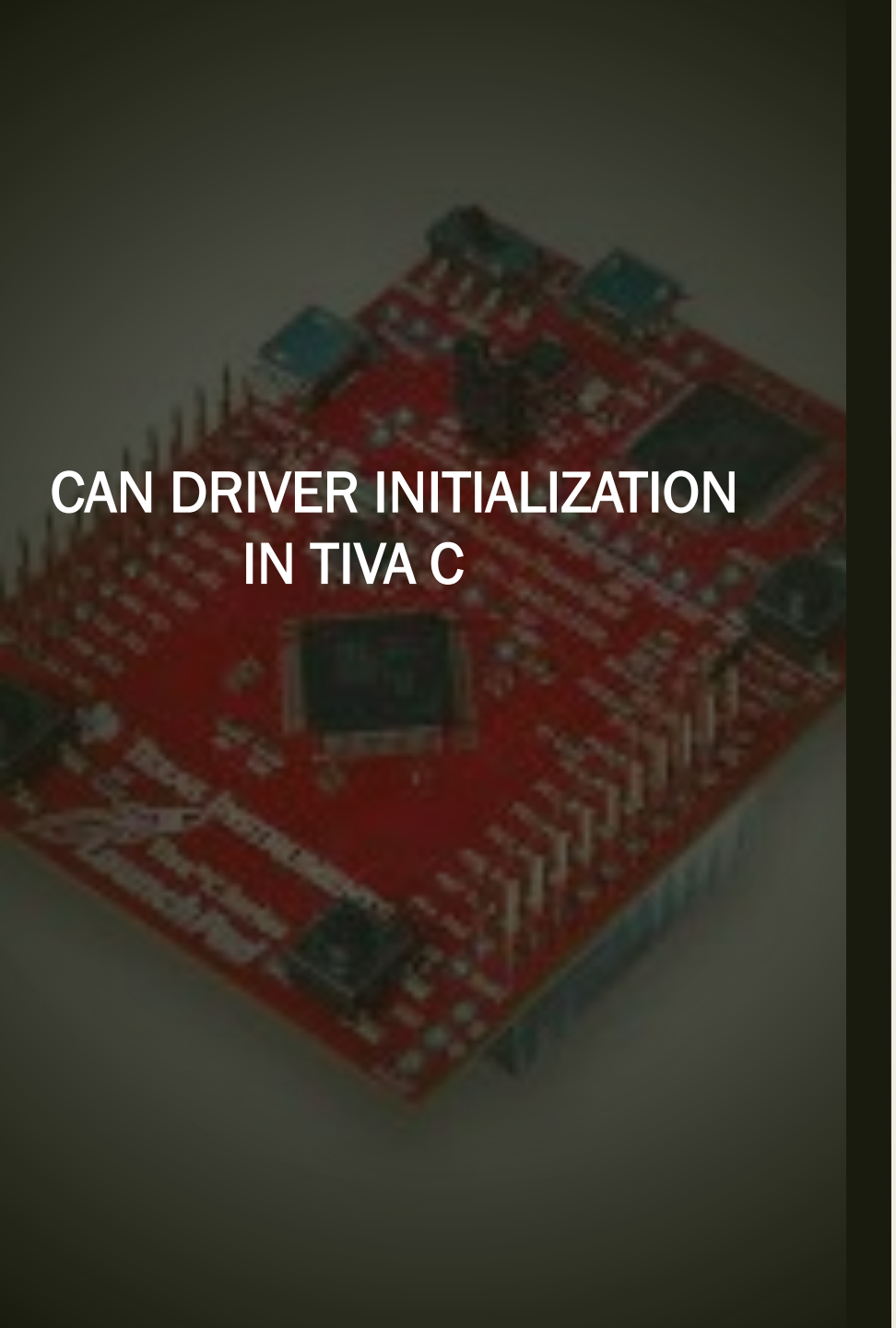
Error Passive status

CAN FIFO BUFFER

- **Configuration of a FIFO Buffer**
  - the configuration of receive message objects belonging to a FIFO buffer is the same as the configuration of a single receive message object
  - the identifiers and masks (if used) of these message objects have to be programmed to matching values

- **Reception of Messages with FIFO Buffers**
  - Received messages with identifiers matching to a FIFO buffer are stored starting with the message object with the lowest message number.
  - When a message is stored into a message object of A FIFO buffer, the `NEWDAT` of the **CANIFnMCTL** register bit of this message object is set.

- **Reading from a FIFO Buffer**
  - To assure the correct function of a FIFO buffer, the CPU should read out the message objects starting with the message object with the lowest message number.



CAN DRIVER INITIALIZATION IN TIVA C

1.the peripheral clock must be enabled using the **RCGC0**register to use the CAN controller.

2.the clock to the appropriate GPIO module must be enabled via the **RCGC2** register.

3.Pins Initialization and configuration (AFSEL, PMCn..)

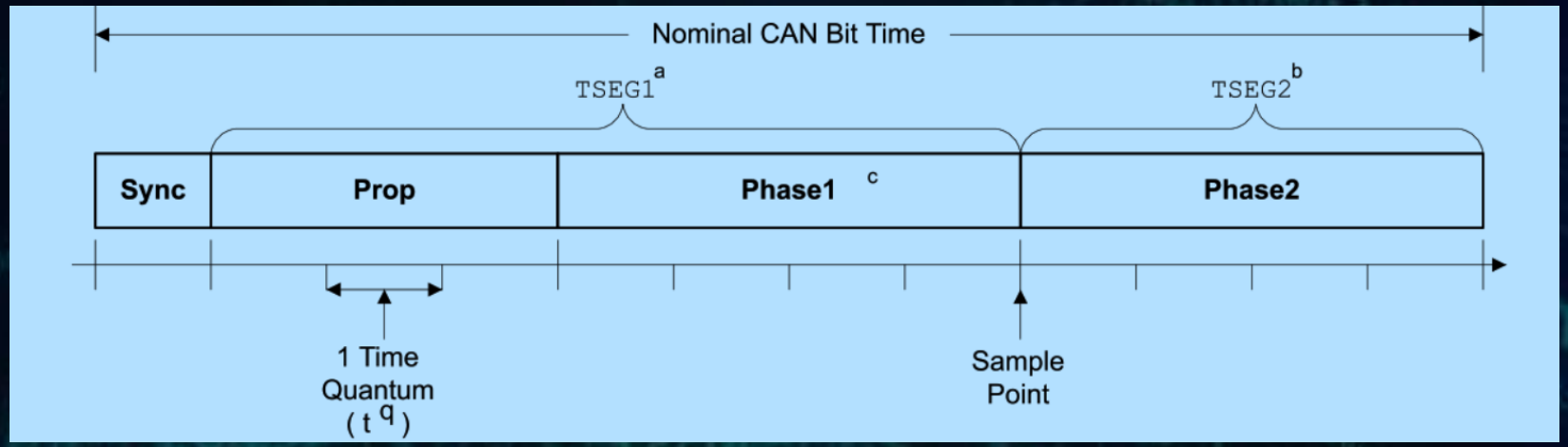
4.Set **INIT**bit in the CAN Control (CANCTL) register

5. SET CCE bit (bit 6) in CANCTL ,where INIT & CCE bits must be set to access CANBIT register.

6. Set the CANBIT register -Bit Time & Bit rate The Can bit rate ranges from 1Kbps up to 1Mbps.
The bit time consists of a number of time quantum (tq).

Bit Time and Bit Rate

Each bit on the CAN bus is, for timing purposes, divided into at least 4 quanta. The quanta are logically divided into four groups or segments Each segment consists of a specific, programmable number of time quanta .The length of the time quantum (tq), which is the basic time unit of the bit time



Parameter	Range
BRP	[1 .. 64]
Sync	1 t_q
Prop	[1 .. 8] t_q
Phase1	[1 .. 8] t_q
Phase2	[1 .. 8] t_q
SJW	[1 .. 4] t_q

CANBIT register Field	Setting
TSEG2	Phase2-1
TSEG1	Prop+Phase1-1
SJW	SJW-1
BRP	BRP-1

CALCULATING BIT RATE

The length of the bit time
= (Sync + Prop + Phase1 + Phase2) x tq
= (1 + TSEG1 + 1 + TSEG2 + 1) x tq

The CAN system supports bit rates in the range of lower than 1 Kbps up to 1000 Kbps. Each member of the CAN network has its own clock generator. The timing parameter of the bit time can be configured individually for each CAN node, creating a commonbit rate even though the CAN nodes' oscillator periods may be different

Synchronization segment
Propagation segment
Phase segment 1
Phase segment 2

Each bit on CAN is 4 divided into 4 segments that needs adjustment properly to get desired bit rate on CAN bus, each segmenttakes Time Quanta; which is the smallest time unit for all configurationsvalues.
Time Quanta = system bus CLK/ CAN Prescaler
Every segment can take X of Time Quanta, calculating X depends on CAN controller type. There is online calculator to calculate X

Thank you

SIEMENS