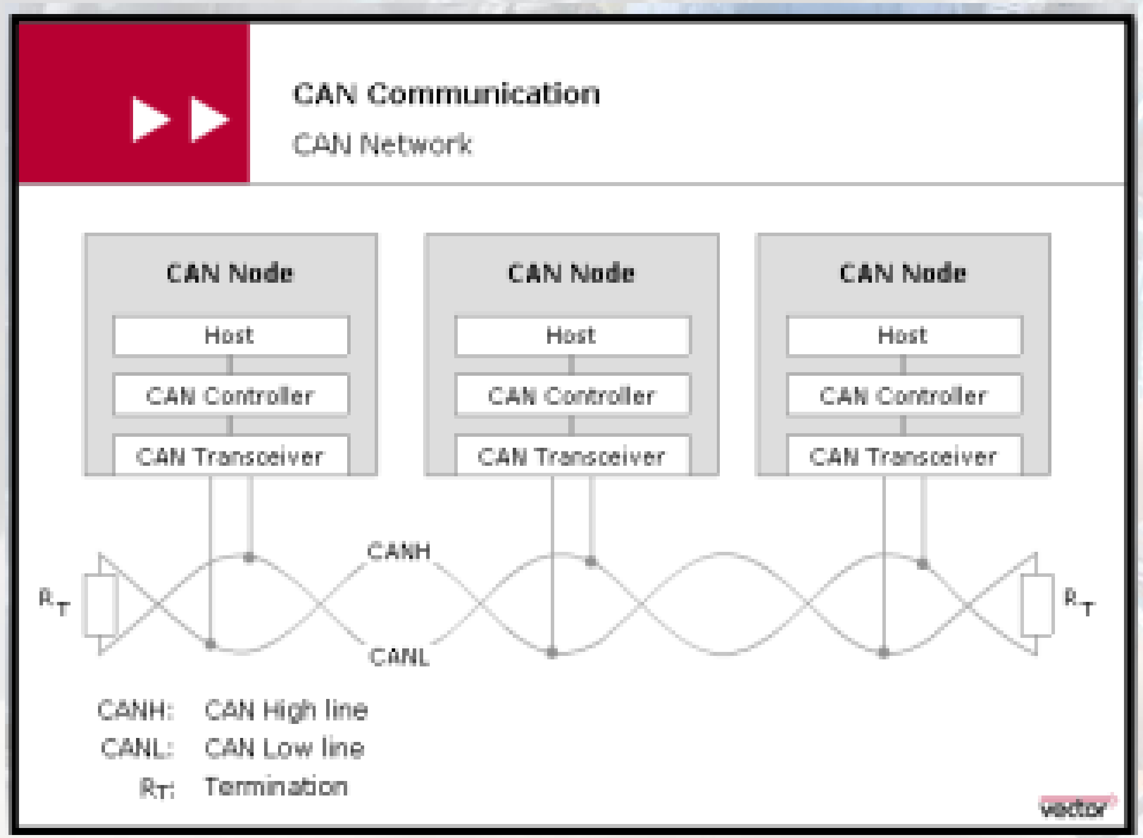
CONTROLLER AREA NETWORK (CAN) MODULE

TIVA™ TM4C123GH6PM MICROCONTROLLER

Bishoy Sabry Salama

Sep. 2021

SIEMENS



CAN Communication

CAN Network

CAN Node

Host

CAN Controller

CAN Transceiver

CAN Node

Host

CAN Controller

CAN Transceiver

CAN Node

Host

CAN Controller

CAN Transceiver

CANH: CAN High line
CANL: CAN Low line
R_T: Termination

SPECIFICATIONS

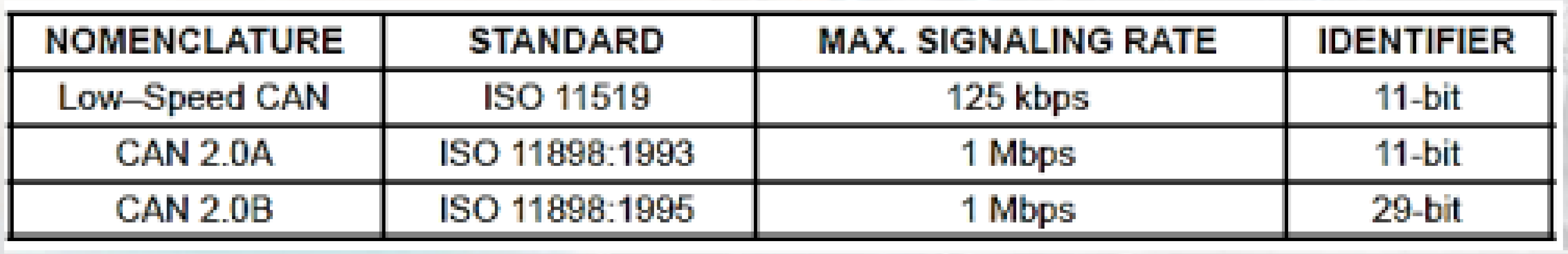
TM4C123GH6PM microcontroller includes *two CAN units* with the following features:

SPECIFICATIONS

Protocol

1- CAN protocol version 2.0 part A/B

Can controller provides classic 11-bit identifier and extended identifier with 29 bit width to accommodate up to 2^29 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

SPECIFICATIONS

Bit Rate

2- it can have bit-rate data transmission up to 1/Mbps:

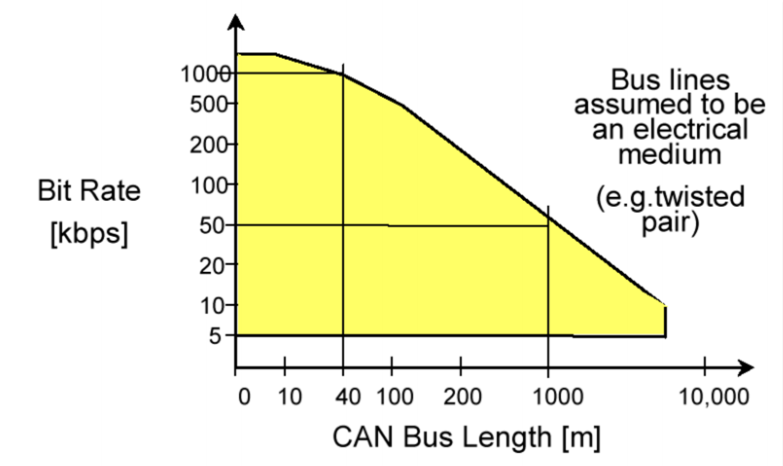
Data transmission has variable bit rate related to the bus length.

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)



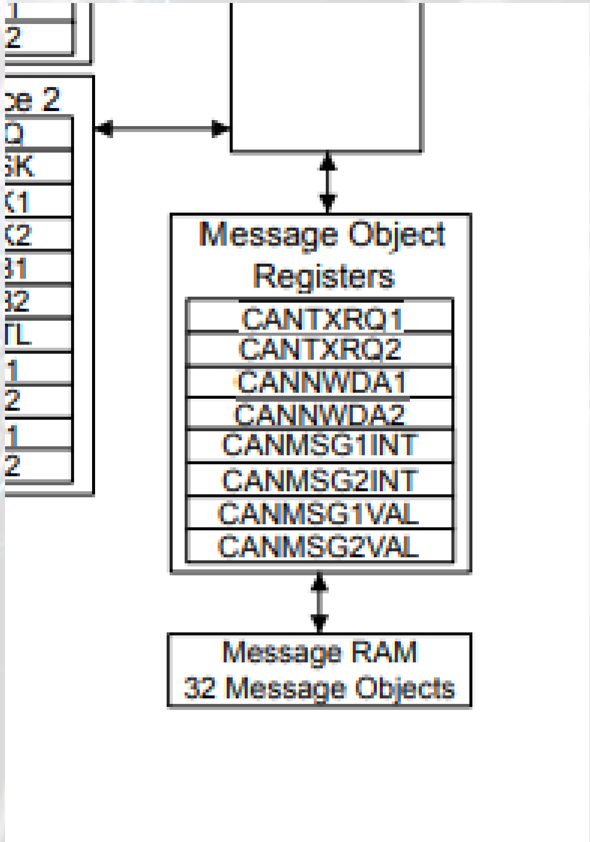
SPECIFICATIONS

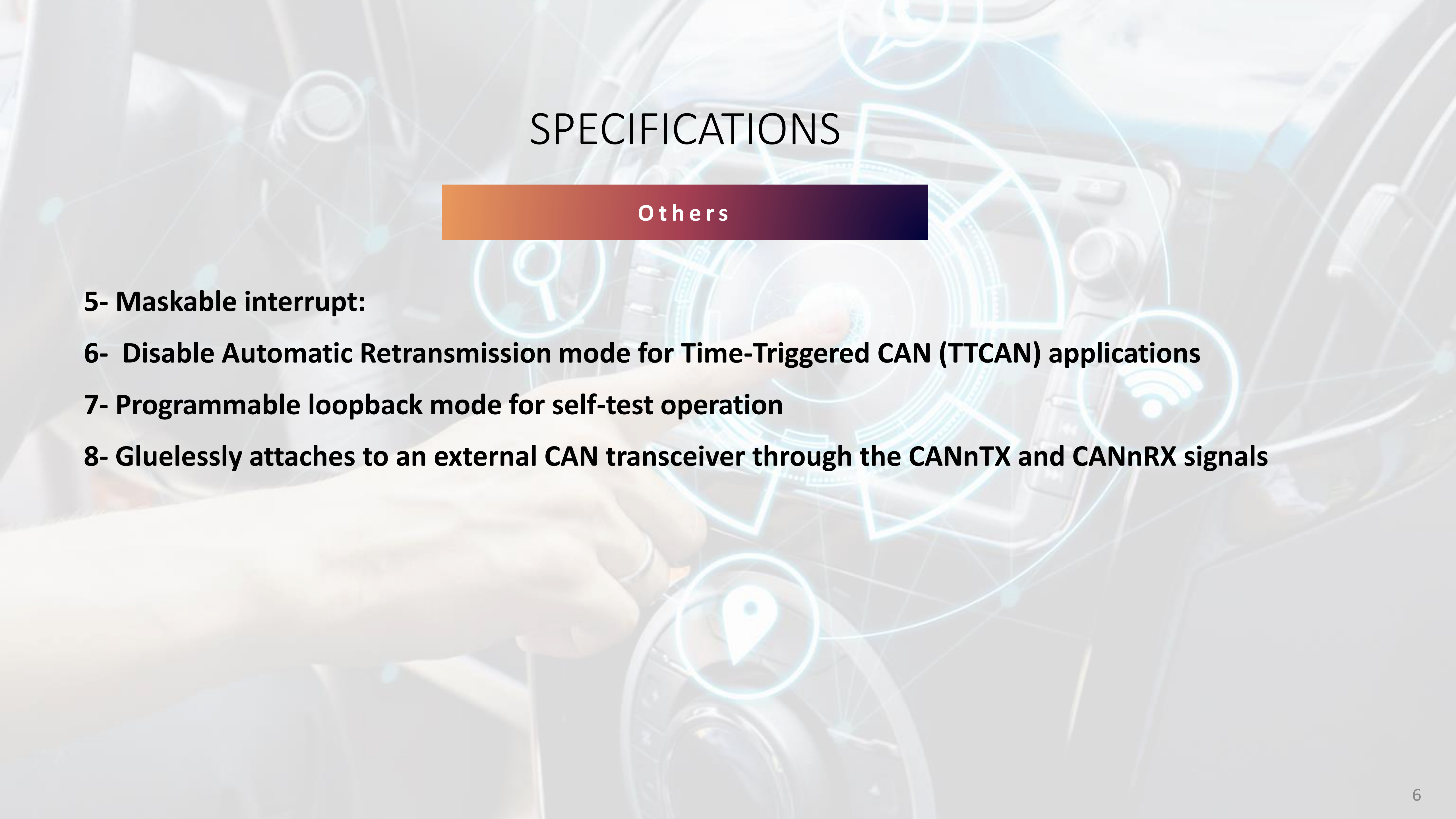
Message Memory

3- It has memory to for 32 message objects with individual identifier masks:

This memory is outside the scope of the tiva-C Microcontroller. So, there is interface to access and edit this memory through the Message Object Registers module inside the can controller

4- Programmable FIFO mode enables storage of multiple message objects





SPECIFICATIONS

Others

5- Maskable interrupt:

6- Disable Automatic Retransmission mode for Time-Triggered CAN (TTCAN) applications

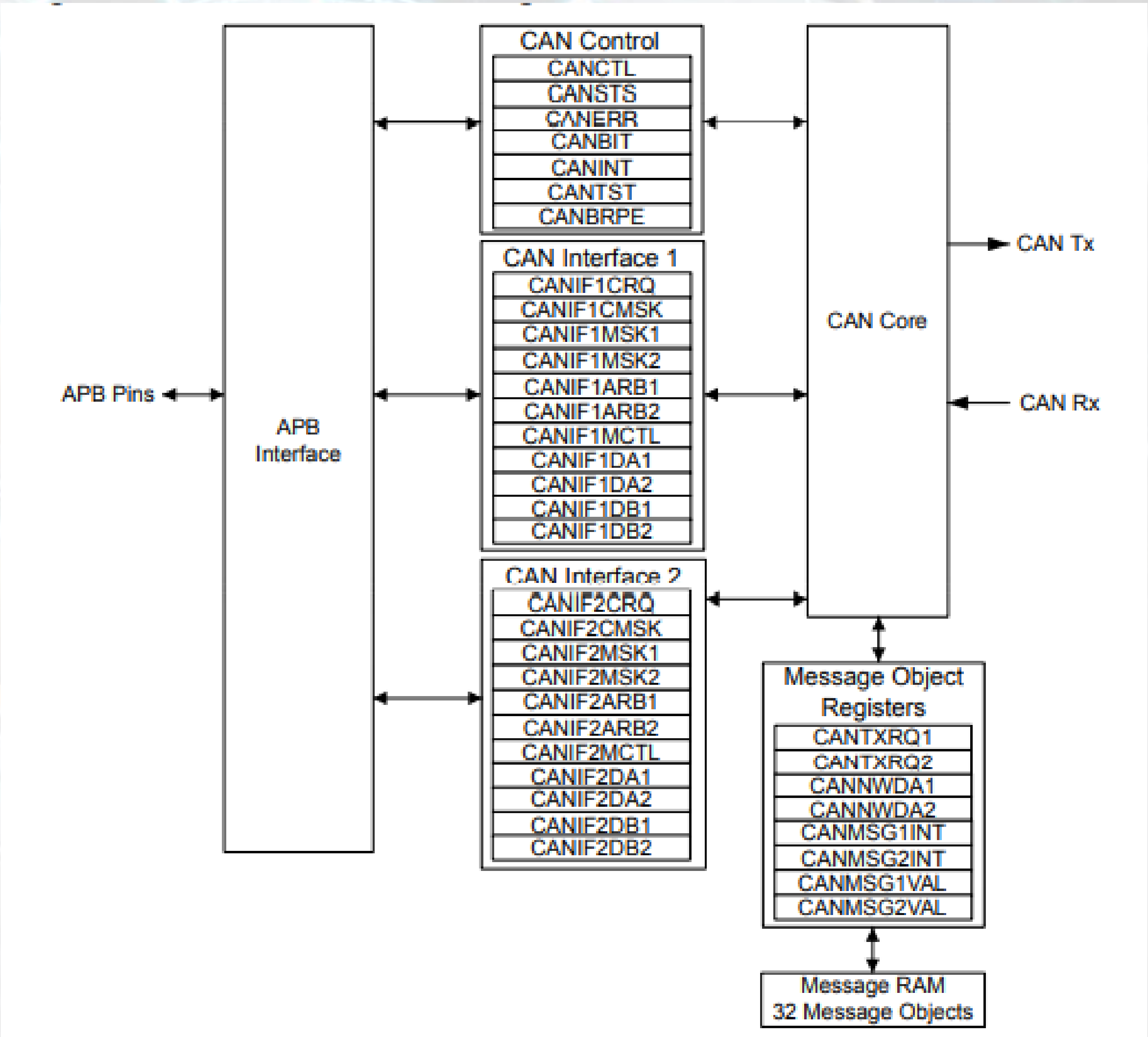
7- Programmable loopback mode for self-test operation

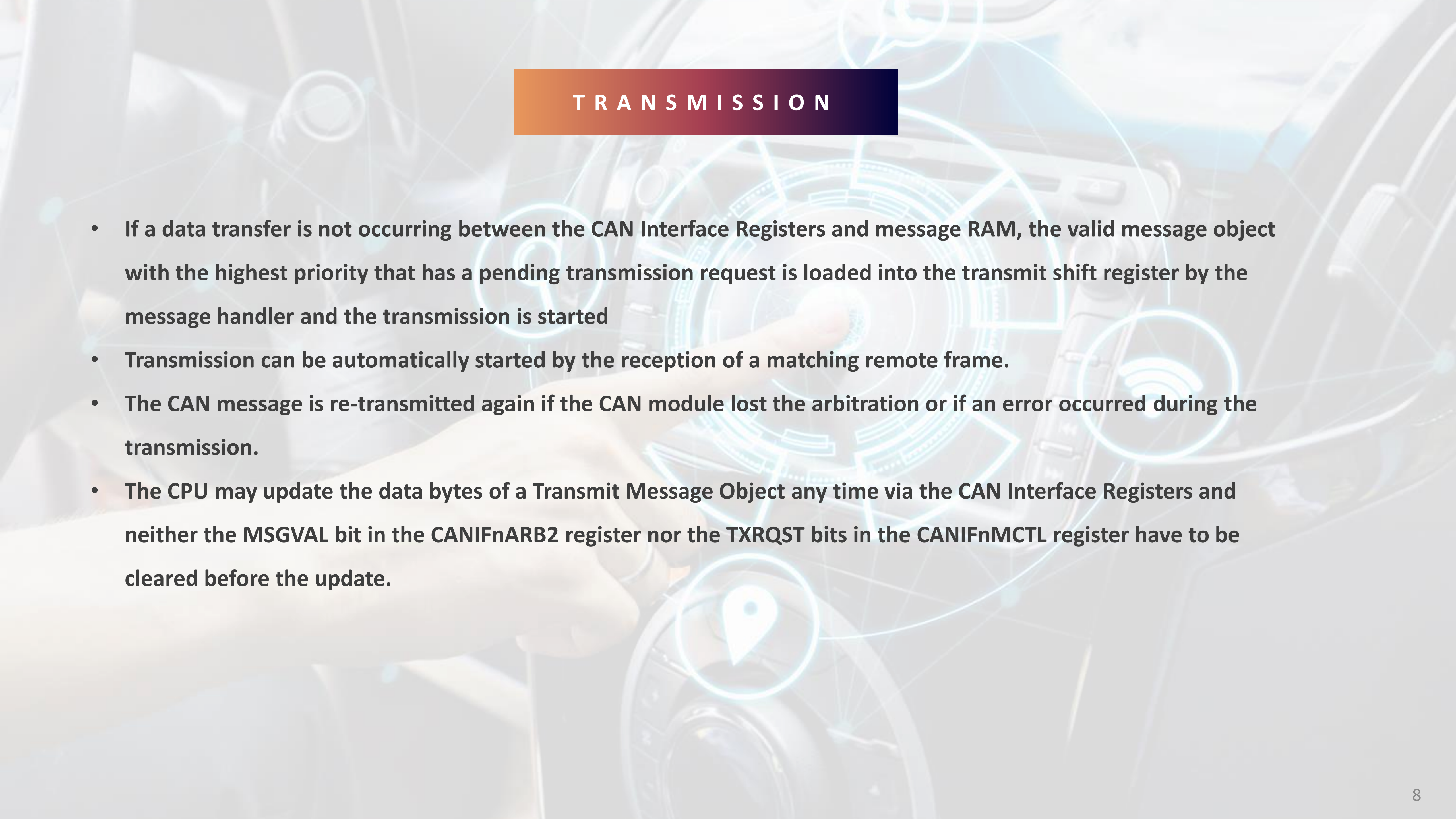
8- Gluelessly attaches to an external CAN transceiver through the CANnTX and CANnRX signals

CAN CONTROLLER BLOCK DIAGRAM

The CAN module consists of three major parts:

- CAN protocol controller and message
- Message memory
- CAN register interface



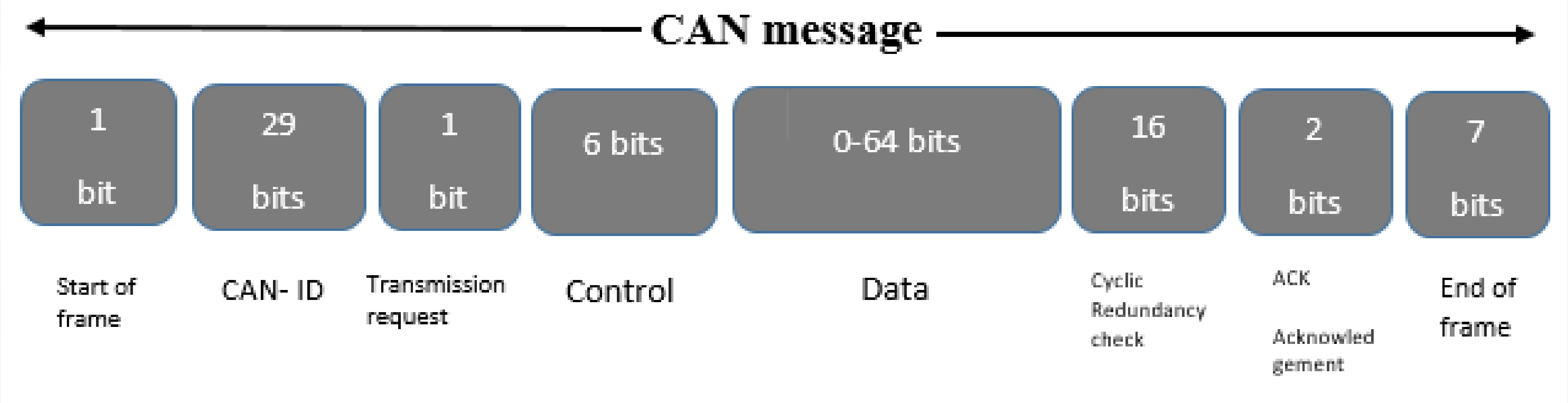


TRANSMISSION

- If a data transfer is not occurring between the CAN Interface Registers and message RAM, the valid message object with the highest priority that has a pending transmission request is loaded into the transmit shift register by the message handler and the transmission is started
- Transmission can be automatically started by the reception of a matching remote frame.
- The CAN message is re-transmitted again if the CAN module lost the arbitration or if an error occurred during the transmission.
- The CPU may update the data bytes of a Transmit Message Object any time via the CAN Interface Registers and neither the MSGVAL bit in the CANIFnARB2 register nor the TXRQST bits in the CANIFnMCTL register have to be cleared before the update.

RECEPTION

- When the arbitration and control field of an incoming message is completely shifted into the CAN controller, the message handling capability of the controller starts scanning the message RAM for a matching valid message object.
- Each Valid message object, starting with object 1, is compared with the incoming message to locate a matching message object in the message RAM. If a match occurs, the scanning is stopped and the message handler proceeds depending on whether it is a data frame or remote frame that was received.

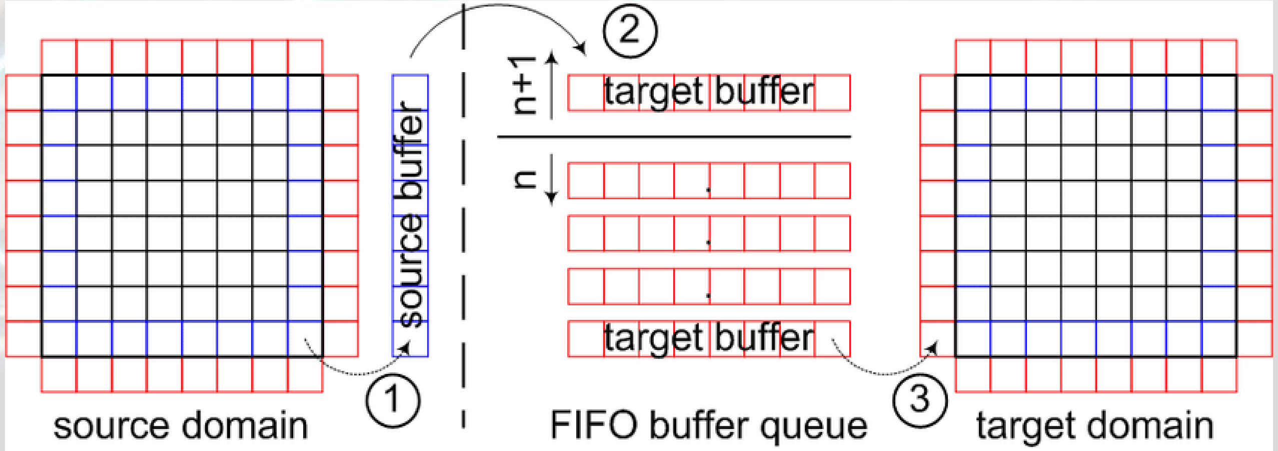


CAN message

1 bit	29 bits	1 bit	6 bits	0-64 bits	16 bits	2 bits	7 bits
Start of frame	CAN- ID	Transmission request	Control	Data	Cyclic Redundancy check	ACK Acknowledgement	End of frame

FIFO BUFFER

- To concatenate two or more message objects into a FIFO buffer, the identifiers and masks (if used) of these message objects have to be programmed to matching values
- Due to the implicit priority of the message objects, the message object with the lowest message object number is the first message object in a FIFO buffer
- 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. By setting
- NEWDAT while EOB is clear, the message object is locked and cannot be written to by the message
- handler until the CPU has cleared the NEWDAT bit.

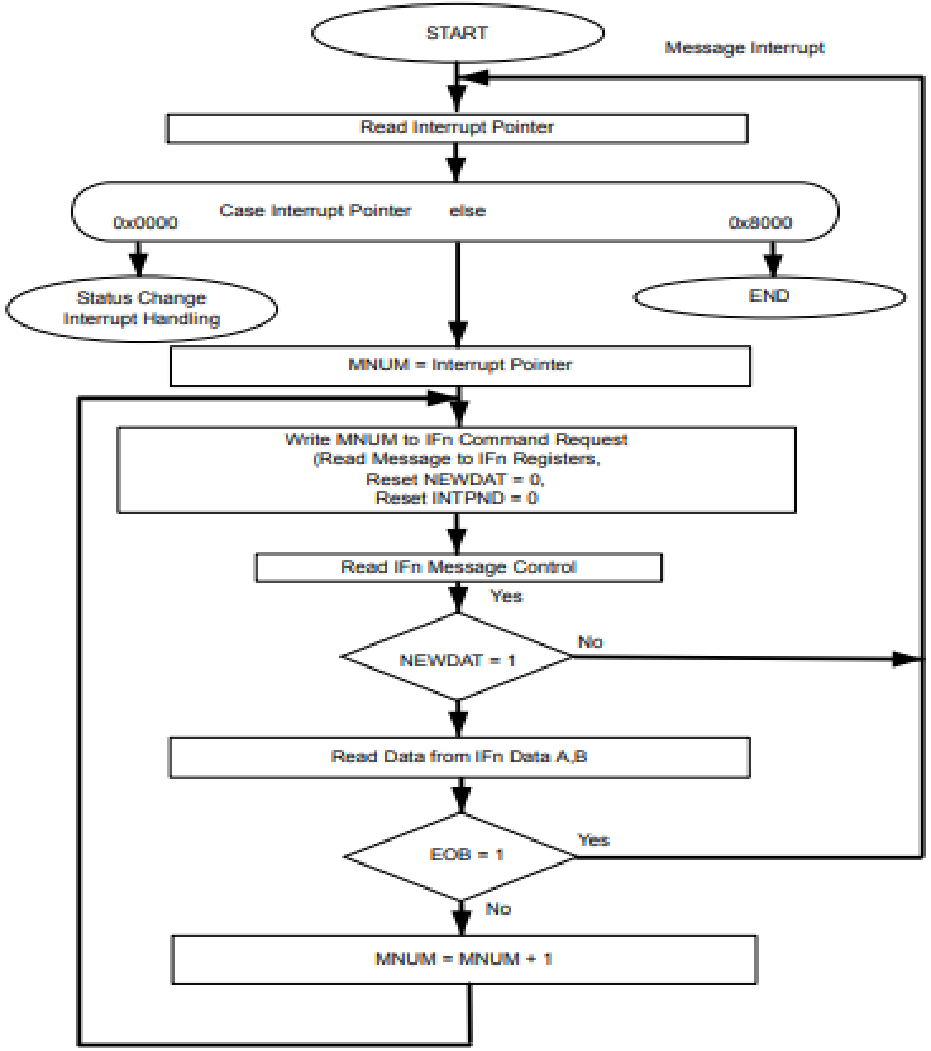


FIFO BUFFER

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.
- When reading from the FIFO buffer, the user should be aware that a new received message is placed in the message object with the lowest message number for which the NEWDAT bit of the CANIFnMCTL register is clear.

Figure 17-3. Message Objects in a FIFO Buffer



INTERRUPTS

Interrupt Type	Description
Bus off status	Occurs when the CAN module is not connected to CAN bus
Warning Status	Occurs when At least one of the error counters has reached the error warning limit of 96.
Error Passive	Occurs when The CAN module is in the Error Passive state, that is, the receive or transmit error count is greater than 127.
Received a Message Successfully	A message has been successfully received, independent of the result of the acceptance filtering
Transmitted a Message Successfully	A message has been successfully transmitted error-free and acknowledged by at least one other node
Last Error Code	Includes (Stuff Error - Format Error - ACK Error - Bit 1 Error - Bit 0 Error - CRC Error)

TEST MODES

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).

The CAN controller is able to receive valid data frames and valid remote frames, but it sends only recessive bits on the CAN bus and cannot start a transmission.

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

n 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.

The CANIF2 Registers are used as a receive buffer

THANK YOU