35. CAN - Control Area Network

35.1. Overview

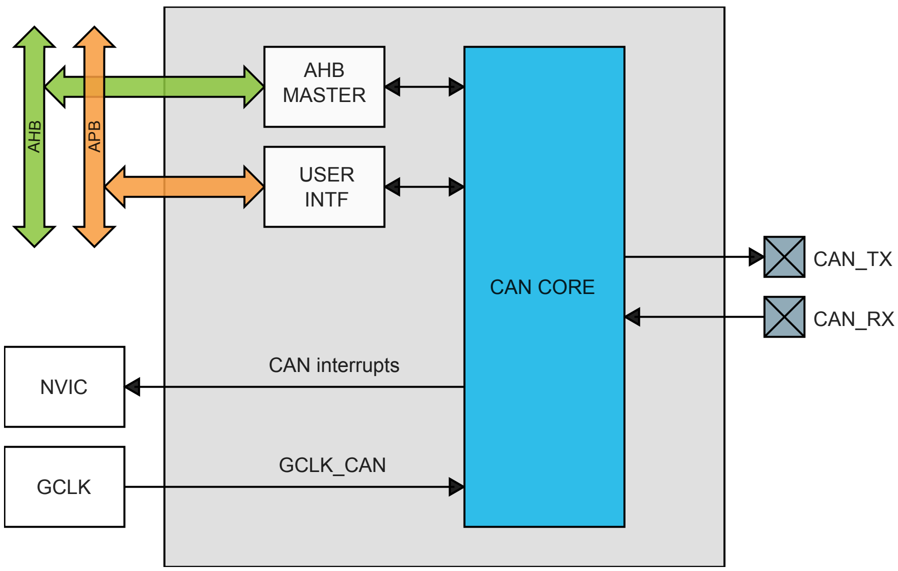
The Control Area Network (CAN) performs communication according to ISO 11898-1 (Bosch CAN specification 2.0 part A,B) and to Bosch CAN FD specification V1.0. The message storage is intended to be a single- or dual-ported Message RAM outside of the module.

35.2. Features

- Conform with CAN protocol version 2.0 part A, B and ISO 11898-1
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- AUTOSAR optimized
- SAE J1939 optimized
- Two configurable Receive FIFOs
- Separate signaling on reception of High Priority Messages
- Up to 64 dedicated Receive Buffers and up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO, Transmit Queue, Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable loop-back test mode
- Maskable module interrupts
- Power-down support; Debug on CAN support

35.3. Block Diagram

Figure 35-1. CAN Block Diagram



35.4. Signal Description

Table 35-1. Signal Description

Signal	Description	Type
CAN_TX	CAN transmit	Digital output
CAN_RX	CAN receive	Digital input

Refer to for details on the pin mapping for this peripheral. One signal can be mapped to one of several pins.

Related Links

I/O Multiplexing and Considerations on page 29

35.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

35.5.1. I/O Lines

Using the CAN's I/O lines requires the I/O pins to be configured.

Related Links

PORT - I/O Pin Controller on page 460

35.5.2. Power Management

The CAN will continue to operate in any sleep mode where the selected source clock is running. The CAN interrupts can be used to wake up the device from sleep modes. Refer to the Power Manager chapter for details on the different sleep modes.

35.5.3. Clocks

The CAN bus clock (CLK_CAN_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_CAN_APB can be found in the Peripheral Clock Masking section.

A generic clock (GCLK_CAN) is required to clock the CAN. This clock must be configured and enabled in the generic clock controller before using the CAN.

This generic clock is asynchronous to the bus clock (CLK_CAN_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

Related Links

Peripheral Clock Masking on page 155

GCLK - Generic Clock Controller on page 130

35.5.4. DMA

The DMA request lines (or line if only one request) are connected to the DMA Controller (DMAC). Using the CAN DMA requests requires the DMA Controller to be configured first.

Related Links

DMAC – Direct Memory Access Controller on page 351

35.5.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the CAN interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 44

35.5.6. Events

Not applicable.

35.5.7. Debug Operation

Not applicable.

35.5.8. Register Access Protection

Not applicable.

35.5.9. Analog Connections

No analog connections.

35.6. Functional Description

35.6.1. Principle of Operation

The CAN performs communication according to ISO 11898-1 (identical to Bosch CAN protocol specification 2.0 part A,B). In addition the CAN supports communication according to CAN FD specification V1.0.

The message storage is intended to be a single- or dual-ported Message RAM outside the module. It is connected to the CAN via AHB.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to the Message RAM as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmit status information.

Acceptance filtering is implemented by a combination of up to 128 filter elements where each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

35.6.2. Operating Modes

35.6.2.1. Software Initialization

Software initialization is started by setting bit CCCR.INIT, either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off. While CCCR.INIT is set, message transfer from and to the CAN bus is stopped, the status of the CAN bus output CAN_TX is ”recessive” (HIGH). The counters of the Error Management Logic EML are unchanged. Setting CCCR.INIT does not change any configuration register. Resetting CCCR.INIT finishes the software initialization. Afterwards the Bit Stream Processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive ”recessive” bits (= Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the CAN configuration registers is only enabled when both bits CCCR.INIT and CCCR.CCE are set (protected write).

CCCR.CCE can only be set/reset while CCCR.INIT = ‘1’. CCCR.CCE is automatically reset when CCCR.INIT is reset.

The following registers are reset when CCCR.CCE is set

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

The Timeout Counter value TOCV.TOC is preset to the value configured by TOCC.TOP when CCCR.CCE is set.

In addition the state machines of the Tx Handler and Rx Handler are held in idle state while CCCR.CCE = ‘1’.

The following registers are only writable while CCCR.CCE = ‘0’

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

CCCR.TEST and CCCR.MON can only be set by the CPU while CCCR.INIT = ‘1’ and CCR.CCE = ‘1’. Both bits may be reset at any time. CCCR.DAR can only be set/reset while CCCR.INIT = ‘1’ and CCCR.CCE = ‘1’.

35.6.2.2. Normal Operation

Once the CAN is initialized and CCCR.INIT is reset to ‘0’, the CAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO0 or Rx FIFO1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

35.6.2.3. CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit. FDF = recessive signifies a CAN FD frame, FDF = dominant signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol. In case the CAN receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit PSR.PXE. When Protocol Exception Handling is enabled (CCCR.PXHD = ‘0’), this causes the operation state to change from Receiver (PSR.ACT = “10”) to Integrating (PSR.ACT = “00”) at the next sample point. In case Protocol Exception Handling is disabled (CCCR.PXHD = ‘1’), the CAN will treat a recessive res bit as a form error and will respond with an error frame.

CAN FD operation is enabled by programming CCCR.FDOE. In case CCCR.FDOE = ‘1’, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element. With CCCR.FDOE = ‘0’, received frames are interpreted as Classic CAN frames, witch leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set. CCCR.FDOE and CCCR.BRSE can only be changed while CCCR.INIT and CCCR.CCE are both set.

With CCCR.FDOE = ‘0’, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format. With CCCR.FDOE = ‘1’ and CCCR.BRSE = ‘0’, only bit FDF of a Tx Buffer element is evaluated. With CCCR.FDOE = ‘1’ and CCCR.BRSE = ‘1’, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN, the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to the table below.

Table 35-2. Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing & Prescaler Register NBTP. In the following CAN FD data phase, the fast CAN bit timing is used as defined by the Data Bit Timing & Prescaler Register DBTP. The bit timing is switched back from the fast timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN clock frequency (GCLK_CAN). Example: with a CAN clock frequency of 20MHz and the shortest configurable bit time of 4 t_q, the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD long and CAN FD fast, the value of the bit ESI (Error Status Indicator) is determined by the transmitter’s error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

35.6.2.4. Transceiver Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CAN_TX the CAN receives the transmitted data from its local CAN transceiver via pin CAN_RX. The received data is delayed by the CAN transceiver’s loop delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transceiver loop delay, the delay compensation is introduced. Without transceiver delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transceivers loop delay.

Description

The CAN’s protocol unit has implemented a delay compensation mechanism to compensate the transmitter delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase of transmitting nodes, the delayed transmit data is compared against the received data at the Secondary Sample Point SSP. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting bit DBTP.TDC.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the CAN’s transmit output CAN_TX through the transceiver to the receive input CAN_RX plus the transmitter delay compensation offset as configured by TDCR.TDCO. The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of mtq.

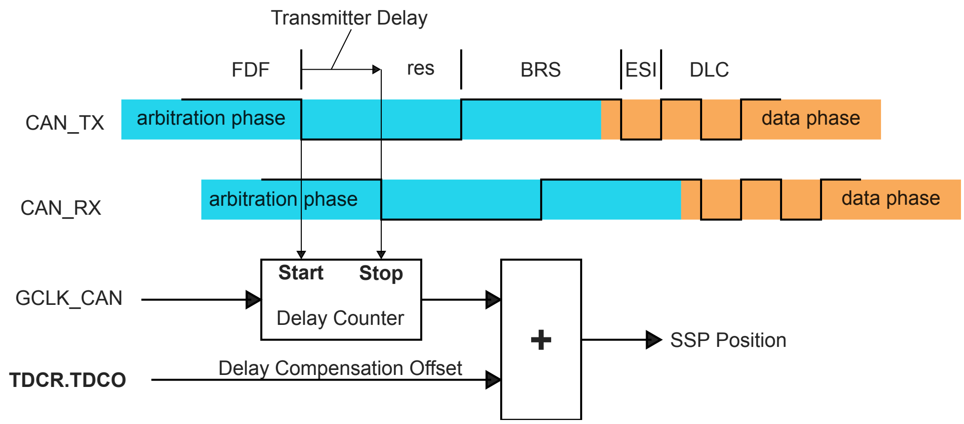
PSR.TDCV shows the actual transmitter delay compensation value. PSR.TDCV is cleared when CCCR.INIT is set and is updated at each transmission of an FD frame while DBTP.TDC is set.

The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN:

- The sum of the measured delay from CAN_TX to CAN_RX and the configured transceiver delay compensation offset FBTP.TDCO has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CAN_TX to CAN_RX and the configured transceiver delay compensation offset FBTP.TDCO has to be less or equal to 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transceiver delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.
  Transmitter Delay Compensation Measurement

If transmitter delay compensation is enabled by programming DBTP.TDC = ‘1’, the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CAN_TX of the transmitter. The resolution of this measurement is one mtq.

Figure 35-2. Transceiver delay measurement



To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a too early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming TDCR.TDCF. This defines a minimum value for the SSP position. Dominant edges of CAN_RX, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least TDCR.TDCF AND CAN _RX is low.

35.6.2.5. Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters (ECR.REC, ECR.TEC) are frozen while Error Logging (ECR.CEL) is active. The CPU can set the CAN into Restricted Operation mode by setting bit CCCR.ASM. The bit can only be set by the CPU when both CCCR.CCE and CCCR.INIT are set to ‘1’. The bit can be reset by the CPU at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the CPU has to reset CCCR.ASM.

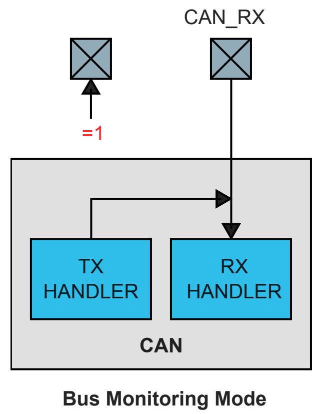
The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

35.6.2.6. Bus Monitoring Mode

The CAN is set in Bus Monitoring Mode by programming CCCR.MON to ‘1’. In Bus Monitoring Mode (see ISO 11898-1, 10.12 Bus monitoring), the CAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the CAN is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode register TXBRP is held in reset state.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. The figure below shows the connection of signals CAN_TX and CAN_RX to the CAN in Bus Monitoring Mode.

Figure 35-3. Pin Control in Bus Monitoring Mode



35.6.2.7. Disabled Automatic Retransmission

According to the CAN Specification (see ISO 11898-1, 6.3.3 Recovery Management), the CAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via CCCR.DAR.

Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending bit TXBRP.TRPx is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx not set
- Successful transmission in spite of cancellation:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx set
- Arbitration lost or frame transmission disturbed:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx not set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = "10" (transmission in spite of cancellation).

35.6.2.8. Power Down (Sleep Mode)

The CAN can be set into power down mode via setting CC Control Register CCCR.CSR = '1'. When all pending transmission requests have completed, the CAN waits until bus idle state is detected. Then the CAN sets then CCCR.INIT to '1' to prevent any further CAN transfers. Now the CAN acknowledges that it is ready for power down by setting CCCR.CSA to '1'. In this state, before the clocks are switched off, further register accesses can be made. A write access to CCCR.INIT will have no effect. Now the module clock inputs CLK_CAN_APB and GCLK_CAN may be switched off.

To leave power down mode, the CPU has to turn on the module clocks before resetting CC Control Register CCCR.CSR = '0'. The CAN will acknowledge this by resetting CCCR.CSA = '0'. Afterwards, the application can restart CAN communication by resetting bit CCCR.INIT.

35.6.2.9. Test Modes

To enable write access to register TEST, bit CCCR.TEST has to be set to ‘1’. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CAN_TX by programming TEST.TX. Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the CAN’s bit timing and it can drive constant dominant or recessive values. The actual value at pin CAN_RX can be read from TEST.RX. Both functions can be used to check the CAN bus’ physical layer.

Due to the synchronization mechanism between GCLK_CAN and GCLK_CAN_APB domains, there may be a delay of several GCLK_CAN_APB periods between writing to TEST.TX until the new configuration is visible at output pin CAN_TX. This applies also when reading input pin CAN_RX via TEST.RX.

Note: Test modes should be used for production tests or self test only. The software control for pin CAN_TX interferes with all CAN protocol functions. It is not recommended to use test modes for application.

External Loop Back Mode

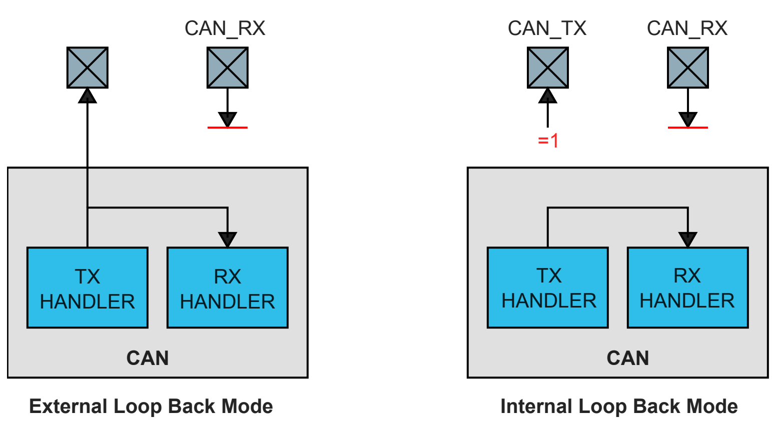
The CAN can be set in External Loop Back Mode by programming TEST.LBCK to ‘1’. In Loop Back Mode, the CAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. The figure below shows the connection of signals CAN_TX and CAN_RX to the CAN in External Loop Back Mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the CAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back Mode. In this mode the CAN performs an internal feedback from its Tx output to its Rx input. The actual value of the CAN_RX input pin is disregarded by the CAN. The transmitted messages can be monitored at the CAN_TX pin.

Internal Loop Back Mode

Internal Loop Back Mode is entered by programming bits TEST.LBCK and CCCR.MON to ‘1’. This mode can be used for a “Hot Selftest”, meaning the CAN can be tested without affecting a running CAN system connected to the pins CAN_TX and CAN_RX. In this mode pin CAN_RX is disconnected from the CAN and pin CAN_TX is held recessive. The figure below shows the connection of CAN_TX and CAN_RX to the CAN in case of Internal Loop Back Mode.

Figure 35-4. Pin Control in Loop Back Modes



35.6.3. Timestamp Generation

For timestamp generation the CAN supplies a 16-bit wrap-around counter. A prescaler TSCC.TCP can be configured to clock the counter in multiples of CAN bit times (1…16). The counter is readable via TSCV.TSC. A write access to register TSCV resets the counter to zero. When the timestamp counter wraps around interrupt flag IR.TSW is set.

On start of frame reception / transmission the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

By programming bit TSCC.TSS an external 16-bit timestamp can be used.

35.6.4. Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO the CAN supplies a 16-bit Timeout Counter. It operates as down-counter and uses the same prescaler controlled by TSCC.TCP as the Timestamp Counter. The Timeout Counter is configured via register TOCC. The actual counter value can be read from TOCV.TOC. The Timeout Counter can only be started while CCCR.INIT = ‘0’. It is stopped when CCCR.INIT = ‘1’, e.g. when the CAN enters Bus_Off state.

The operation mode is selected by TOCC.TOS. When operating in Continuous Mode, the counter starts when CCCR.INIT is reset. A write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored. Writing to TOCV has no effect.

When the counter reaches zero, interrupt flag IR.TOO is set. In Continuous Mode, the counter is immediately restarted at TOCC.TOP.

Note: The clock signal for the Timeout Counter is derived from the CAN Core’s sample point signal. Therefore the point in time where the Timeout Counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

35.6.5. Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO’s Put and Get Indices.

35.6.5.1. Acceptance Filtering

The CAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration GFC
- Standard ID Filter Configuration SIDFC
- Extended ID Filter Configuration XIDFC
- Extended ID AND Mask XIDAM

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM
- Set High Priority Message interrupt flag IR.HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected Rx Buffer or Rx FIFO:

Rx Buffer
New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type see PSR.LEC respectively PSR.FLEC.

Rx FIFO
Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type see PSR.LEC respectively PSR.FLEC. In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in Rx FIFO Overwrite Mode have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

Range Filter

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID for standard frames or EF1ID/EF2ID for extended frames.

There are two possibilities when range filtering is used together with extended frames:

EFT = “00”	The Message ID of received frames is AND’ed with the Extended ID AND Mask (XIDAM) before the range filter is applied
EFT = “11”	The Extended ID AND Mask (XIDAM) is not used for range filtering

Filter for specific IDs

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

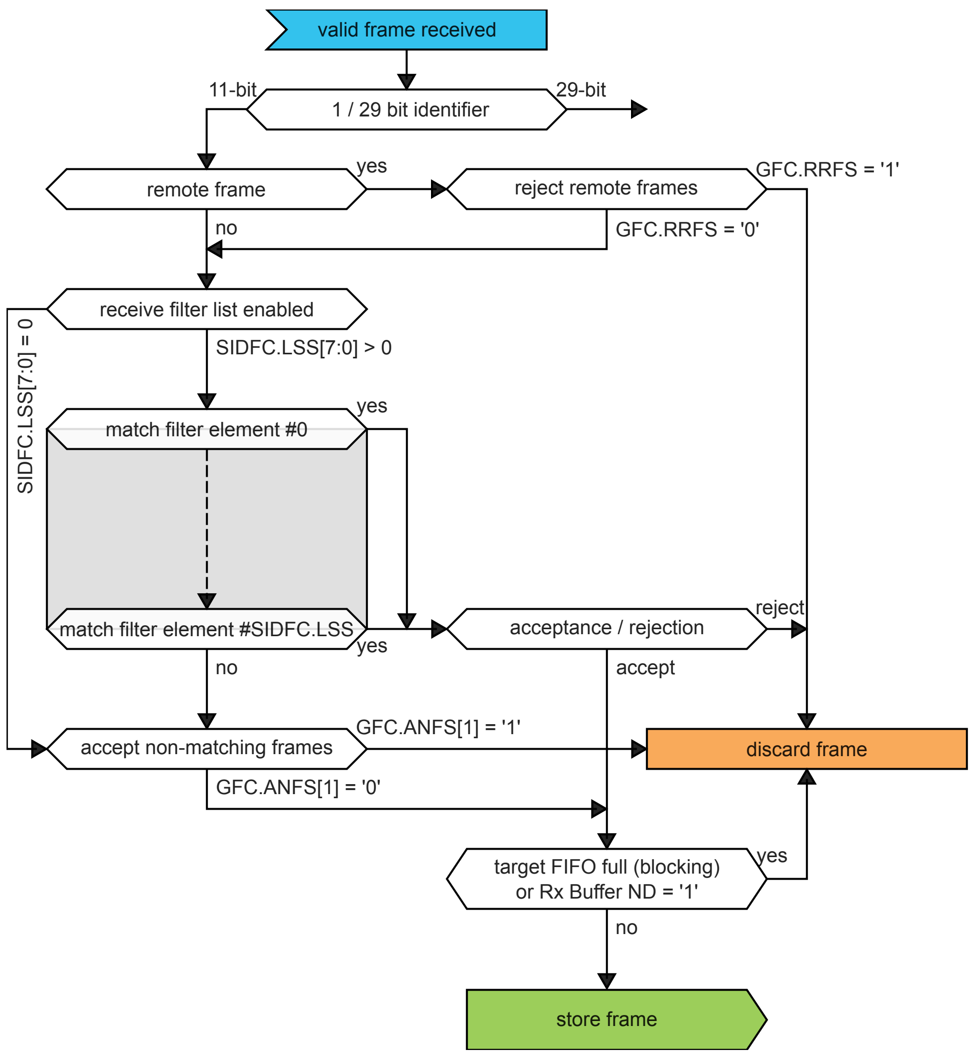
In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

Standard Message ID Filtering

The figure below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in Standard Message ID Filter Element.

Controlled by the Global Filter Configuration GFC and the Standard ID Filter Configuration SIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 35-5. Standard Message ID Filtering



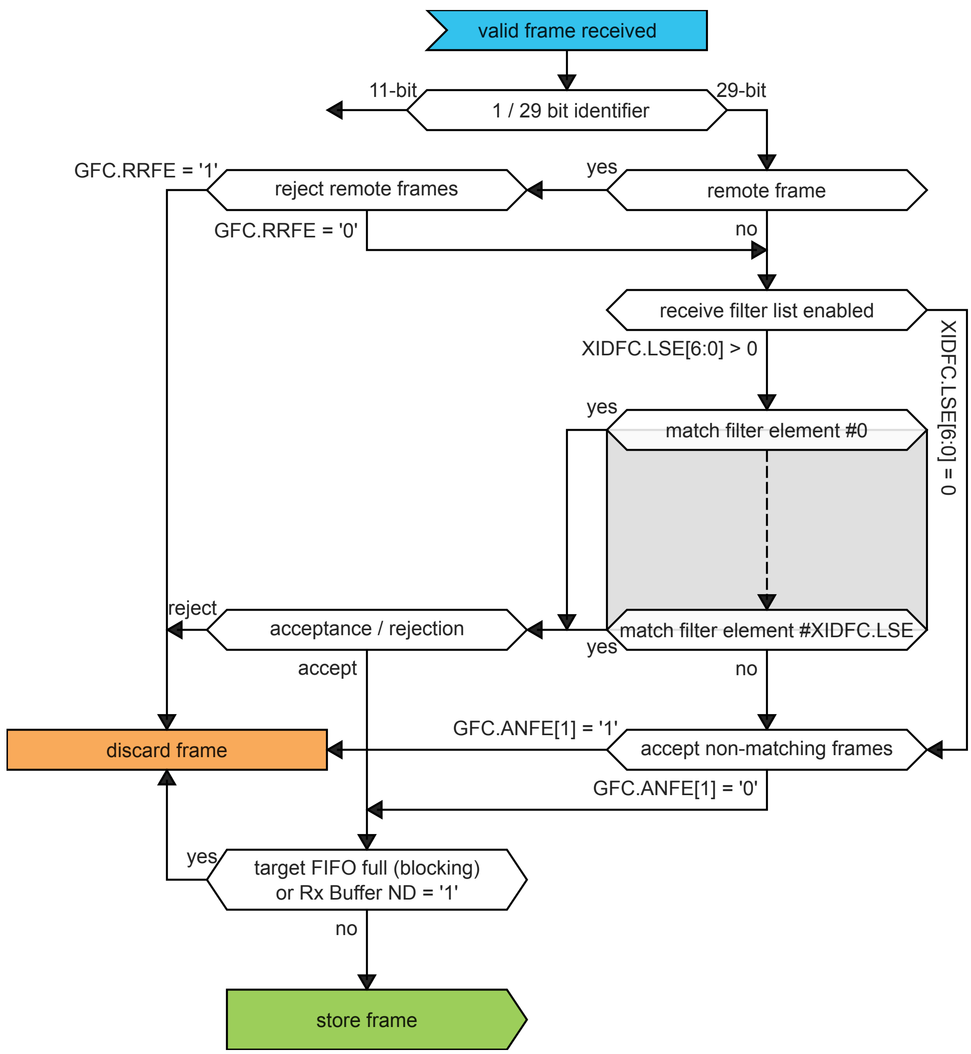
Extended Message ID Filtering

The figure below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in Extended Message ID Filter Element.

Controlled by the Global Filter Configuration GFC and the Extended ID Filter Configuration XIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask XIDAM is AND'ed with the received identifier before the filter list is executed.

Figure 35-6. Extended Message ID Filtering



35.6.5.2. Rx FIFOs

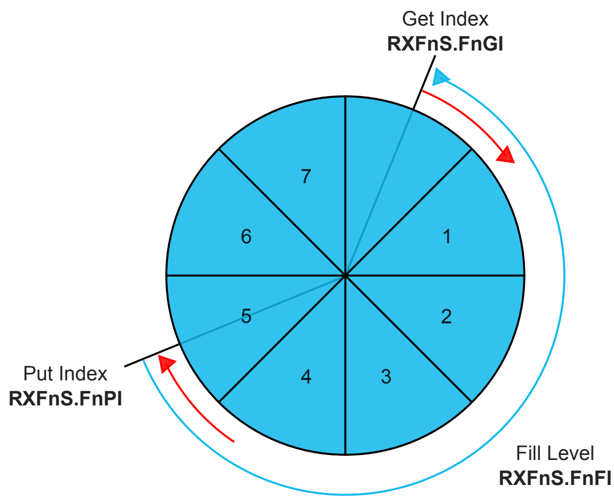
Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via registers RXF0C and RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1 see Acceptance Filtering. The Rx FIFO element is described in Rx Buffer and FIFO Element.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by RXFnC.FnWM, interrupt flag IR.RFnW is set. When the Rx FIFO

Put Index reaches the Rx FIFO Get Index an Rx FIFO Full condition is signalled by RXFnS.FnF. In addition interrupt flag IR.RFnF is set.

Figure 35-7. Rx FIFO Status



When reading from an Rx FIFO, Rx FIFO Get Index RXFnS.FnGI • FIFO Element Size has to be added to the corresponding Rx FIFO start address RXFnC.FnSA.

Table 35-3. Rx Buffer / FIFO Element Size

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by RXFnC.FnOM = ‘0’. This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (RXFnS.FnPI = RXFnS.FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by RXFnS.FnF = ‘1’. In addition interrupt flag IR.RFnF is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by RXFnS.RFnL = ‘1’. In addition interrupt flag IR.RFnL is set.

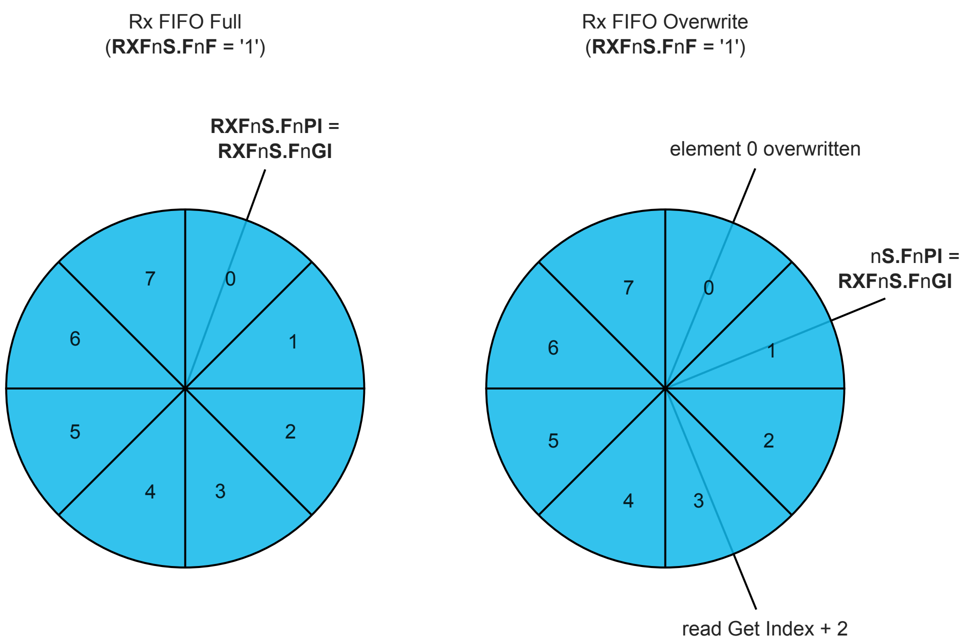
Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by RXFnC.FnOM = ‘1’.

When an Rx FIFO full condition (RXFnS.FnPI = RXFnS.FnGI) is signaled by RXFnS.FnF = ‘1’, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signaled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. The figure below shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 35-8. Rx FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index RXFnA.FnA. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (RXFnS.FnF = ‘0’).

35.6.5.3. Dedicated Rx Buffers

The CAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via RXBC.RBSA.

For each Rx Buffer a Standard or Extended Message ID Filter Element with SFEC / EFEC = “111” and SFID2 / EFID2[10:9] = “00” has to be configured (see Standard Message ID Filter Element and Extended Message ID Filter Element).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag IR.DRX (Message stored in Dedicated Rx Buffer) in the interrupt register is set.

Table 35-4. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] / EFID1[28:0]	SFID2[10:9] / EFID2[10:9]	SFID2[5:0] / EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1, NDAT2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the CPU by writing a '1' to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

Rx Buffer Handling

- Reset interrupt flag IR.DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

35.6.5.4. Debug on CAN Support

Debug messages are stored into Rx Buffers. For debug handling three consecutive Rx buffers (e.g. #61, #62, #63) have to be used for storage of debug messages A, B, and C. The format is the same as for an Rx Buffer or an Rx FIFO element (see Rx Buffer and FIFO Element).

Advantage: Fixed start address for the DMA transfers (relative to RXBC.RBSA), no additional configuration required.

For filtering of debug messages Standard / Extended Filter Elements with SFEC / EFEC = "111" have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by SFID2 / EFID2[5:0].

After message C has been stored, the DMA request output is activated and the three messages can be read from the Message RAM under DMA control. The RAM words holding the debug messages will not be changed by the CAN while DMA request is activated. The behavior is similar to that of an Rx Buffers with its New Data flag set.

After the DMA has completed the DMA unit sets the DMA acknowledge. This resets DMA request. Now the CAN is prepared to receive the next set of debug messages.

Filtering for Debug Messages

Filtering for debug messages is done by configuring one Standard / Extended Message ID Filter Element for each of the three debug messages. To enable a filter element to filter for debug messages SFEC / EFEC has to be programmed to "111". In this case fields SFID1 / SFID2 and EFID1 / EFID2 have a different meaning (see Standard Message ID Filter Element and Extended Message ID Filter Element). While SFID2 / EFID2[10:9] controls the debug message handling state machine, SFID2 / EFID2[5:0] controls the location for storage of a received debug message.

When a debug message is stored, neither the respective New Data flag nor IR.DRX are set. The reception of debug messages can be monitored via RXF1S.DMS.

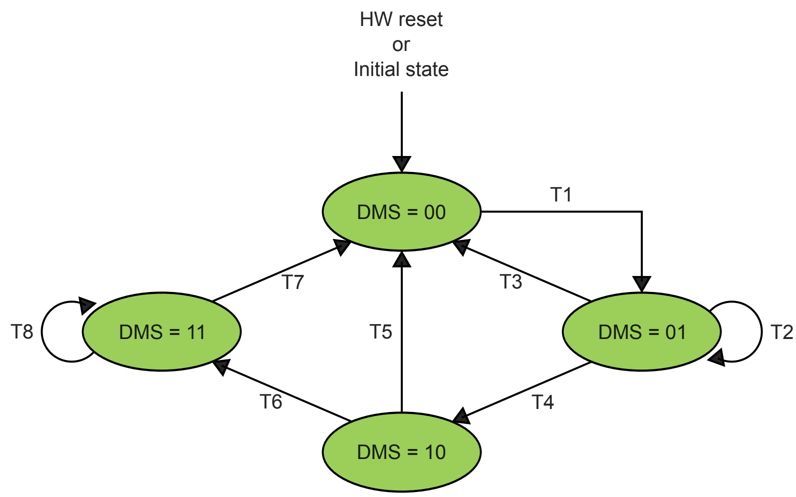
Table 35-5. Example Filter Configuration for Debug Messages

Filter Element	SFID1[10:0] / EFID1[28:0]	SFID2[10:9] / EFID2[10:9]	SFID2[5:0] / EFID2[5:0]
0	ID debug message A	01	11 1101
1	ID debug message B	10	11 1110
2	ID debug message C	11	11 1111

Debug Message Handling

The debug message handling state machine assures that debug messages are stored to three consecutive Rx Buffers in correct order. In case of missing messages the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in correct order.

Figure 35-9. Debug Message Handling State Machine



T0: Reset DMA request output, enable reception of debug message A, B, and C
T1: Reception of debug message A
T2: Reception of debug message A
T3: Reception of debug message C
T4: Reception of debug message B
T5: Reception of debug message A, B
T6: Reception of debug message C
T7: DMA transfer completed
T8: Reception of debug message A, B, C (message rejected)

35.6.6. Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in Tx Buffer Element. The table below describes the possible configurations for frame transmission.

Table 35-6. Possible Configurations for Frame Transmission

CCCR		Tx Buffer Element		Frame Transmission
BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when the Tx Buffer Request Pending register TXBRP is updated, or when a transmission has been started.

35.6.6.1. Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit CCCR.TXP. If the bit is set, the CAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (CCCR.TXP = ‘0’).

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

35.6.6.2. Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (refer to table below). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0…31) • Element Size to the Tx Buffer Start Address TXBC.TBSA.

Table 35-7. Tx Buffer / FIFO / Queue Element Size

TXESC.TBDS[2:0]	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

35.6.6.3. Tx FIFO

Tx FIFO operation is configured by programming TXBC.TFQM to '0'. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index TXFQS.TFGI. After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The CAN calculates the Tx FIFO Free Level TXFQS.TFFL as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index TXFQS.TFQPI. An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS.TFQF = '1') is signaled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a '1' to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is canceled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (refer to Table 35-7 Tx Buffer / FIFO / Queue Element Size). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI (0…31) • Element Size to the Tx Buffer Start Address TXBC.TBSA.

35.6.6.4. Tx Queue

Tx Queue operation is configured by programming TXBC.TFQM to '1'. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index TXFQS.TFQPI. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TXFQS.TFQF = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been canceled.

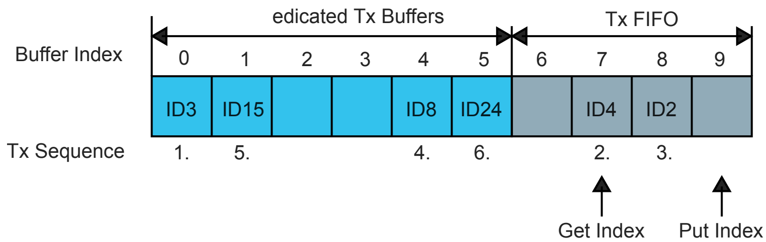
The application may use register TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (refer to Table 35-7 Tx Buffer / FIFO / Queue Element Size). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI (0…31) • Element Size to the Tx Buffer Start Address TXBC.TBSA.

35.6.6.5. Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx FIFO. The number of Dedicated Tx Buffers is configured by TXBC.NDTB. The number of Tx Buffers assigned to the Tx FIFO is configured by TXBC.TFQS. In case TXBC.TFQS is programmed to zero, only Dedicated Tx Buffers are used.

Figure 35-10. Example of mixed Configuration Dedicated Tx Buffers / Tx FIFO



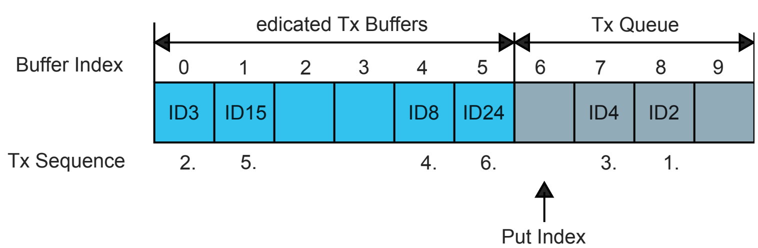
Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by TXFS.TFGI)
- Buffer with lowest Message ID gets highest priority and is transmitted next

35.6.6.6. Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx Queue. The number of Dedicated Tx Buffers is configured by TXBC.NDTB. The number of Tx Queue Buffers is configured by TXBC.TFQS. In case TXBC.TFQS is programmed to zero, only Dedicated Tx Buffers are used.

Figure 35-11. Example of mixed Configuration Dedicated Tx Buffers / Tx Queue



Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

35.6.6.7. Transmit Cancellation

The CAN supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the CPU has to write a '1' to the corresponding bit position (=number of Tx Buffer) of register TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of register TXBCF to '1'.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding TXBTO and TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding TXBCF bit is set.

Note: In case a pending transmission is canceled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

35.6.6.8. Tx Event Handling

To support Tx event handling the CAN has implemented a Tx Event FIFO. After the CAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in Tx Event FIFO Element.

When a Tx Event FIFO full condition is signaled by IR.TEFF, no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by TXEFC.EFWM, interrupt flag IR.TEFW is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index TXEFS.EFGI has to be added to the Tx Event FIFO start address TXEFC.EFSA.

35.6.7. FIFO Acknowledge Handling

The Get Indexes of Rx FIFO 0, Rx FIFO 1 and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (refer to RXF0A, RXF1A and TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the CPU has free access to the CAN's Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The CAN does not check for erroneous values.

35.6.8. Interrupts

The CAN has the following interrupt sources:

- Access to Reserved Address
- Protocol Errors (Data Phase / Arbitration Phase)
- Watchdog Interrupt
- Bus_Off Status
- Error Warning & Passive
- Error Logging Overflow
- Message RAM Bit Errors (Uncorrected / Corrected)
- Message stored to Dedicated Rx Buffer
- Timeout Occurred
- Message RAM Access Failure
- Timestamp Wraparound
- Tx Event FIFO statuses (Element Lost / Full / Watermark Reached / New Entry)
- Tx FIFO Empty
- Transmission Cancellation Finished
- Timestamp Completed
- High Priority Message
- Rx FIFO 1 Statuses (Message Lost / Full / Watermark Reached / New Message)
- Rx FIFO 0 Statuses (Message Lost / Full / Watermark Reached / New Message)

Each interrupt source has an interrupt flag associated with it. The interrupt flag register (IR) is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing '1' or disabled by writing '0' to the corresponding bit in the interrupt enable register (IE). Each interrupt flag can be assigned to one of two interrupt service lines.

An interrupt request is generated when an interrupt flag is set, the corresponding interrupt enable is set, and the corresponding service line enable assigned to the interrupt is set. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the service line is disabled, or the CAN is reset. Refer to IR for details on how to clear interrupt flags. All interrupt requests from the peripheral are sent to the NVIC. The user must read the IR register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller on page 44

35.6.9. Sleep Mode Operation

The CAN can be configured to operate in any sleep mode. To be able to run in standby, register MRCFG.RUNSTDBY must be written to '1'.

To prevent data corruption, it is recommended to allow the CAN to complete all pending transactions before setting the system on standby. This is performed by setting the Clock Stop Request register CCCR.CSR = '1'. Once all transactions are completed, the CAN will automatically set the Clock Stop Acknowledge register CCCR.CSA = '1'. The CAN has reverted back to its initial state and is now safe for the system to go to standby.

When the system wakes up, the CAN cannot be reprogrammed unless the Clock Stop Request register is cleared (CCCR.CSR = ‘0’). When the Clock Stop Acknowledge register CCCR.CSA returns a ‘0’, the CAN is ready to be programmed.

35.6.10. Synchronization

Due to the asynchronicity between the main clock domain (CLK_CAN_APB) and the peripheral clock domain (GCLK_CAN) some registers are synchronized when written. When a write-synchronized register is written, the read back value will not be updated until the register has completed synchronization.

The following bits and registers are write-synchronized:

l Initialization bit in CC Control register (CCCR.INIT)

35.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CREL	7:0								
0x01		15:8								
0x02		23:16	SUBSTEP[3:0]							
0x03		31:24	REL[3:0]				STEP[3:0]			
0x04	ENDN	7:0	ETV[7:0]							
0x05		15:8	ETV[15:8]							
0x06		23:16	ETV[23:16]							
0x07		31:24	ETV[31:24]							
0x08	MRCFG	7:0		RUNSTDBY					DQOS[1:0]	
0x09		15:8								
0x0A		23:16								
0x0B		31:24								
0x0C	DBTP	7:0	DTSEG2[3:0]				DSJW[3:0]			
0x0D		15:8				DTSEG1[4:0]				
0x0E		23:16	TDC			DBRP[4:0]				
0x0F		31:24								
0x10	TEST	7:0	RX	TX[1:0]		LBCK				
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	RWD	7:0	WDC[7:0]							
0x15		15:8	WDV[7:0]							
0x16		23:16								
0x17		31:24								
0x18	CCCR	7:0	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
0x19		15:8		TXP	EFBI	PXHD			BRSE	FDOE
0x1A		23:16								
0x1B		31:24								
0x1C	NBTP	7:0		NTSEG2[6:0]						
0x1D		15:8	NTSEG1[7:0]							
0x1E		23:16	NBRP[7:0]							
0x1F		31:24	NSJW[6:0]							NBRP[8:8]
0x20	TSCC	7:0							TSS[1:0]	
0x21		15:8								
0x22		23:16					TCP[3:0]			
0x23		31:24								
0x24	TSCV	7:0	TSC[7:0]							
0x25		15:8		TSC[14:8]						
0x26		23:16								
0x27		31:24								
0x28	TOCC	7:0						TOS[1:0]		ETOC
0x29		15:8								
0x2A		23:16	TOP[7:0]							
0x2B		31:24	TOP[15:8]							

Offset	Name	Bit Pos.								
0x2C	TOCV	7:0	TOC[7:0]							
0x2D		15:8	TOC[15:8]							
0x2E		23:16								
0x2F		31:24								
0x30 ... 0x3F	Reserved									
0x40	ECR	7:0	TEC[7:0]							
0x41		15:8	RP	REC[6:0]						
0x42		23:16	CEL[7:0]							
0x43		31:24								
0x44	PSR	7:0	BO	EW	EP	ACT[1:0]		LEC[2:0]		
0x45		15:8		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
0x46		23:16		TDCV[6:0]						
0x47		31:24								
0x48	TDCR	7:0		TDCF[6:0]						
0x49		15:8		TDCO[6:0]						
0x4A		23:16								
0x4B		31:24								
0x4C ... 0x4F	Reserved									
0x50	IR	7:0	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
0x51		15:8	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
0x52		23:16	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
0x53		31:24			ARA	PED	PEA	WDI	BO	EW
0x54	IE	7:0	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
0x55		15:8	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
0x56		23:16	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
0x57		31:24			ARAE	PEDE	PEAE	WDIE	BOE	EWE
0x58	ILS	7:0	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
0x59		15:8	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
0x5A		23:16	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
0x5B		31:24			ARAL	PEDL	PEAL	WDIL	BOL	EWL
0x5C	ILE	7:0							EINT1	EINT0
0x5D		15:8								
0x5E		23:16								
0x5F		31:24								
0x60 ... 0x7F	Reserved									
0x80	GFC	7:0			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
0x81		15:8								
0x82		23:16								
0x83		31:24								

Offset	Name	Bit Pos.								
0x84	SIDFC	7:0	FLSSA[7:0]							
0x85		15:8	FLSSA[15:8]							
0x86		23:16	LSS[7:0]							
0x87		31:24								
0x88	XIDFC	7:0	FLESA[7:0]							
0x89		15:8	FLESA[15:8]							
0x8A		23:16		LSE[6:0]						
0x8B		31:24								
0x8C ... 0x8F	Reserved									
0x90	XIDAM	7:0	EIDM[7:0]							
0x91		15:8	EIDM[15:8]							
0x92		23:16	EIDM[23:16]							
0x93		31:24				EIDM[28:24]				
0x94	HPMS	7:0	MSI[1:0]		BIDX[5:0]					
0x95		15:8	FLST	FIDX[6:0]						
0x96		23:16								
0x97		31:24								
0x98	NDAT1	7:0	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
0x99		15:8	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
0x9A		23:16	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
0x9B		31:24	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
0x9C	NDAT2	7:0	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
0x9D		15:8	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
0x9E		23:16	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
0x9F		31:24	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
0xA0	RXF0C	7:0	F0SA[7:0]							
0xA1		15:8	F0SA[15:8]							
0xA2		23:16		F0S[6:0]						
0xA3		31:24	F0OM	F0WM[6:0]						
0xA4	RXF0S	7:0		F0FL[6:0]						
0xA5		15:8			F0GI[5:0]					
0xA6		23:16			F0PI[5:0]					
0xA7		31:24							RF0L	F0F
0xA8	RXF0A	7:0			F0AI[5:0]					
0xA9		15:8								
0xAA		23:16								
0xAB		31:24								
0xAC	RXBC	7:0	RBSA[7:0]							
0xAD		15:8	RBSA[15:8]							
0xAE		23:16								
0xAF		31:24								
0xB0	RXF1C	7:0	F1SA[7:0]							
0xB1		15:8	F1SA[15:8]							
0xB2		23:16		F1S[6:0]						
0xB3		31:24	F1OM	F1WM[6:0]						

Offset	Name	Bit Pos.								
0xB4	RXF1S	7:0		F1FL[6:0]						
0xB5		15:8			F1GI[5:0]					
0xB6		23:16			F1PI[5:0]					
0xB7		31:24	DMS[1:0]						RF1L	F1F
0xB8	RXF1A	7:0			F1AI[5:0]					
0xB9		15:8								
0xBA		23:16								
0xBB		31:24								
0xBC	RXESC	7:0		F1DS[2:0]				F0DS[2:0]		
0xBD		15:8						RBDS[2:0]		
0xBE		23:16								
0xBF		31:24								
0xC0	TXBC	7:0	TBSA[7:0]							
0xC1		15:8	TBSA[15:8]							
0xC2		23:16			NDTB[5:0]					
0xC3		31:24		TFQM	TFQS[5:0]					
0xC4	TXFQS	7:0			TFFL[5:0]					
0xC5		15:8				TFGI[4:0]				
0xC6		23:16			TFQF	TFQPI[4:0]				
0xC7		31:24								
0xC8	TXESC	7:0						TBDS[2:0]		
0xC9		15:8								
0xCA		23:16								
0xCB		31:24								
0xCC	TXBRP	7:0	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
0xCD		15:8	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
0xCE		23:16	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
0xCF		31:24	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
0xD0	TXBAR	7:0	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
0xD1		15:8	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
0xD2		23:16	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
0xD3		31:24	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
0xD4	TXBCR	7:0	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
0xD5		15:8	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
0xD6		23:16	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
0xD7		31:24	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
0xD8	TXBTO	7:0	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
0xD9		15:8	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
0xDA		23:16	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
0xDB		31:24	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
0xDC	TXBCF	7:0	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
0xDD		15:8	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
0xDE		23:16	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
0xDF		31:24	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24

Offset	Name	Bit Pos.								
0xE0	TXBTIE	7:0	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
0xE1		15:8	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
0xE2		23:16	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
0xE3		31:24	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
0xE4	TXBCIE	7:0	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
0xE5		15:8	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
0xE6		23:16	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
0xE7		31:24	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
0xE8 ... 0xEF	Reserved									
0xF0	TXEFC	7:0	EFSA[7:0]							
0xF1		15:8	EFSA[15:8]							
0xF2		23:16			EFS[5:0]					
0xF3		31:24			EFWM[5:0]					
0xF4	TXEFS	7:0				EFFI[4:0]				
0xF5		15:8				EFGI[4:0]				
0xF6		23:16				EFP[4:0]				
0xF7		31:24							TEFL	EFF
0xF8	TXEFA	7:0				EFAI[4:0]				
0xF9		15:8								
0xFA		23:16								
0xFB		31:24								

35.8. Register Description

Registers are 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

35.8.1. Core Release

Name: CREL
Offset: 0x00
Reset: 0x32100000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	REL[3:0]				STEP[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	1	0	0	1	0

Bit	23	22	21	20	19	18	17	16
	SUBSTEP[3:0]							
Access	R	R	R	R				
Reset	0	0	0	1				

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:28 – REL[3:0]: Core Release
One digit, BCD-coded.

Bits 27:24 – STEP[3:0]: Step of Core Release
One digit, BCD-coded.

Bits 23:20 – SUBSTEP[3:0]: Sub-step of Core Release
One digit, BCD-coded.

35.8.2. Endian

Name: ENDN
Offset: 0x04
Reset: 0x87654321
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ETV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	0	0	1	1	1

Bit	23	22	21	20	19	18	17	16
	ETV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	1	0	0	1	0	1

Bit	15	14	13	12	11	10	9	8
	ETV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	0	0	0	0	1	1

Bit	7	6	5	4	3	2	1	0
	ETV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	0	0	0	0	1

Bits 31:0 – ETV[31:0]: Endianness Test Value
The endianness test value is 0x87654321

35.8.3. Message RAM Configuration

Name: MRCFG
Offset: 0x08
Reset: 0x00000002
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					DQOS[1:0]	
Access		R/W					R/W	R/W
Reset		0					1	0

Bit 6 – RUNSTDBY: Run in Standby
This bit controls the behavior of the CAN during standby sleep mode.

Value	Description
0	The CAN GCLK request is always disabled during sleep to conserve power consumption.
1	The CAN GCLK request is disabled during sleep when CCCR.CSA = 1.

Bits 1:0 – DQOS[1:0]: Data Quality of Service
This field defines the memory priority access during the Message RAM read/write data operation.

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive bandwidth
0x2	MEDIUM	Sensitive latency
0x3	HIGH	Critical latency

35.8.4. Data Bit Timing and Prescaler

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

The CAN bit time may be programmed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 GCLK_CAN periods. t_q = (DBRP + 1) mtq.

Note:
With a GCLK_CAN of 8MHz, the reset value 0x00000A33 configures the CAN for a fast bit rate of 500 kBits/s.

The bit rate configured for the CAN FD data phase via DBTP must be higher or equal to the bit rate configured for the arbitration phase via NBTP.

Name: DBTP
Offset: 0x0C
Reset: 0x00000A33
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TDC			DBRP[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DTSEG1[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
	DTSEG2[3:0]				DSJW[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	1

Bit 23 – TDC: Transceiver Delay Compensation

Value	Description
0	Transceiver Delay Compensation disabled.
1	Transceiver Delay Compensation enabled.

Bits 20:16 – DBRP[4:0]: Data Baud Rate Prescaler

Value	Description
0x00 - 0x1F	The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 12:8 – DTSEG1[4:0]: Fast time segment before sample point

Value	Description
0x00 - 0x1F	Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. DTSEG1 is the sum of Prop_Seg and Phase_Seg1.

Bits 7:4 – DTSEG2[3:0]: Data time segment after sample point

Value	Description
0x0 - 0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. DTSEG2 is Phase_Seg2.

Bits 3:0 – DSJW[3:0]: Data (Re)Syncronization Jump Width

Value	Description
0x0 - 0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

35.8.5. Test

Name: TEST
Offset: 0x10
Reset: 0x00000000
Property: Read-only, Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RX	TX[1:0]		LBCK				
Access	R	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit 7 – RX: Receive Pin
Monitors the actual value of pin CAN_RX

Value	Description
0	The CAN bus is dominant (CAN_RX = 0).
1	The CAN bus is recessive (CAN_RX = 1).

Bits 6:5 – TX[1:0]: Control of Transmit Pin
This field defines the control of the transmit pin.

Value	Name	Description
0x0	CORE	Reset value, CAN_TX controlled by CAN core, updated at the end of CAN bit time.
0x1	SAMPLE	Sample Point can be monitored at pin CAN_TX.
0x2	DOMINANT	Dominant (‘0’) level at pin CAN_TX.
0x3	RECESSIVE	Recessive (‘1’) level at pin CAN_TX.

Bit 4 – LBCK: Loop Back Mode

Value	Description
0	Loop Back Mode is disabled.
1	Loop Back Mode is enabled.

35.8.6. RAM Watchdog

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the CAN's AHB Master Interface starts the Message RAM Watchdog Counter with the value configured by RWD.WDC. The counter is reloaded with RWD.WDC when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt IR.WDI is set.

Name: RWD
Offset: 0x14
Reset: 0x00000000
Property: Read-only, Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WDV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WDC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – WDV[7:0]: Watchdog Value
Actual Message RAM Watchdog Counter Value.

Bits 7:0 – WDC[7:0]: Watchdog Configuration
Start value of the Message RAM Watchdog Counter. With the reset value of 0x00 the counter is disabled.

35.8.7. CC Control

Name: CCCR
Offset: 0x18
Reset: 0x00000001
Property: Read-only, Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		TXP	EFBI	PXHD			BRSE	FDOE
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit 14 – TXP: Transmit Pause
This bit field is write-restricted and only writable if bit fields CCE = 1 and INIT = 1.

Value	Description
0	Transmit pause disabled.
1	Transmit pause enabled. The CAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame.

Bit 13 – EFBI: Edge Filtering during Bus Integration

Value	Description
0	Edge filtering is disabled.
1	Two consecutive dominant tq required to detect an edge for hard synchronization.

Bit 12 – PXHD: Protocol Exception Handling Disable
Note: When protocol exception handling is disabled, the CAN will transmit an error frame when it detects a protocol exception condition.

Value	Description
0	Protocol exception handling enabled.
1	Protocol exception handling disabled.

Bit 9 – BRSE: Bit Rate Switch Enable

Note: When CAN FD operation is disabled FDOE = 0, BRSE is not evaluated.

Value	Description
0	Bit rate switching for transmissions disabled.
1	Bit rate switching for transmissions enabled.

Bit 8 – FDOE: FD Operation Enable

Value	Description
0	FD operation disabled.
1	FD operation enabled.

Bit 7 – TEST: Test Mode Enable

This bit field is write-restricted.

Writing a 0 to this field is always allowed.

Writing a 1 to this field is only allowed if bit fields CCE = 1 and INIT = 1.

Value	Description
0	Normal operation. Register TEST holds reset values.
1	Test Mode, write access to register TEST enabled.

Bit 6 – DAR: Disable Automatic Retransmission

This bit field is write-restricted and only writable if bit fields CCE = 1 and INIT = 1.

Value	Description
0	Automatic retransmission of messages not transmitted successfully enabled.
1	Automatic retransmission disabled.

Bit 5 – MON: Bus Monitoring Mode

This bit field is write-restricted.

Writing a 0 to this field is always allowed.

Writing a 1 to this field is only allowed if bit fields CCE = 1 and INIT = 1.

Value	Description
0	Bus Monitoring Mode is disabled.
1	Bus Monitoring Mode is enabled.

Bit 4 – CSR: Clock Stop Request

Value	Description
0	No clock stop is requested.
1	Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.

Bit 3 – CSA: Clock Stop Acknowledge

Value	Description
0	No clock stop acknowledged.
1	CAN may be set in power down by stopping CLK_CAN_APB and GCLK_CAN.

Bit 2 – ASM: Restricted Operation Mode
This bit field is write-restricted.

Writing a 0 to this field is always allowed.

Writing a 1 to this field is only allowed if bit fields CCE = 1 and INIT = 1.

Value	Description
0	Normal CAN operation.
1	Restricted Operation Mode active.

Bit 1 – CCE: Configuration Change Enable
This bit field is write-restricted and only writable if bit field INIT = 1.

Value	Description
0	The CPU has no write access to the protected configuration registers.
1	The CPU has write access to the protected configuration registers (while CCCR.INIT = 1).

Bit 0 – INIT: Initialization
Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. The programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.

Value	Description
0	Normal Operation.
1	Initialization is started.

35.8.8. Nominal Bit Timing and Prescaler

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 GCLK_CAN periods. t_q = (NBRP + 1) mtq.

Note: With a CAN clock (GCLK_CAN) of 8MHz, the reset value 0x06000A03 configures the CAN for a bit rate of 500 kBits/s.

Name: NBTP
Offset: 0x1C
Reset: 0x00000A33
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
	NSJW[6:0]							NBRP[8:8]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

Bit	23	22	21	20	19	18	17	16
	NBRP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NTSEG1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
		NTSEG2[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	1	1

Bits 31:25 – NSJW[6:0]: Nominal (Re)Syncronization Jump Width

Value	Description
0x00 - 0x7F	Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

Bits 24:16 – NBRP[8:0]: Nominal Baud Rate Prescaler

Value	Description
0x000 - 0x1FF	The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 15:8 – NTSEG1[7:0]: Nominal Time segment before sample point

Value	Description
0x00 - 0x7F	Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. NTSEG1 is the sum of Prop_Seg and Phase_Seg1.

Bits 6:0 – NTSEG2[6:0]: Time segment after sample point

Value	Description
0x00 - 0x7F	Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. NTSEG2 is Phase_Seg2.

35.8.9. Timestamp Counter Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Name: TSCC
Offset: 0x20
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					TCP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							TSS[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 19:16 – TCP[3:0]: Timestamp Counter Prescaler

Note: With CAN FD an external counter is required for timestamp generation (TSS = 0x2).

Value	Description
0x0 - 0xF	Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1...16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 1:0 – TSS[1:0]: Timestamp Select

This field defines the timestamp counter selection.

Value	Name	Description
0x0 or 0x3	ZERO	Timestamp counter value always 0x0000.
0x1	INC	Timestamp counter value incremented by TCP.
0x2	EXT	External timestamp counter value used.

35.8.10. Timestamp Counter Value

Note:

1. A write access to TSCV while in internal mode clears the Timestamp Counter value. A write access to TSCV while in external mode has no impact.
2. A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by the write access to TSCV.

Name: TSCV
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		TSC[14:8]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TSC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 14:0 – TSC[14:0]: Timestamp Counter
The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx). When TSCC.TSS = 0x1, the Timestamp Counter is incremented in multiples of CAN bit times [1...16] depending on the configuration of TSCC.TCP. A wrap around sets interrupt flag IR.TSW. When TSCC.TSS = 0x2, TSC reflects the external Timestamp Counter value.

35.8.11. Timeout Counter Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Name: TOCC
Offset: 0x28
Reset: 0xFFFF0000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
	TOP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	TOP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						TOS[1:0]		ETOC
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:16 – TOP[15:0]: Timeout Period
Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

Bits 2:1 – TOS[1:0]: Timeout Select
When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored.

Value	Name	Description
0x0	CONT	Continuous operation.
0x1	TXEF	Timeout controlled by TX Event FIFO.
0x2	RXF0	Timeout controlled by Rx FIFO 0.
0x3	RXF1	Timeout controlled by Rx FIFO 1.

Bit 0 – ETOC: Enable Timeout Counter

Value	Description
0	Timeout Counter disabled.
1	Timeout Counter enabled.

35.8.12. Timeout Counter Value

Note: A write access to TOCV reloads the Timeout Counter with the value of TOCV.TOP.

Name: TOCV
Offset: 0x2C
Reset: 0x0000FFFF
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TOC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	TOC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 15:0 – TOC[15:0]: Timeout Counter

The Timeout Counter is decremented in multiples of CAN bit times [1...16] depending on the configuration of TSCC.TCP. When decremented to zero, interrupt flag IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS.

35.8.13. Error Counter

Note: When CCCR.ASM is set, the CAN protocol controller does not increment TECand REC when a CAN protocol error is detected, but CEL is still incremented.

Name: ECR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RP	REC[6:0]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TEC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – CEL[7:0]: CAN Error Logging
The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR.ELO.

Bit 15 – RP: Receive Error Passive

Bits 14:8 – REC[6:0]: Receive Error Counter
Actual state of the Receive Error Counter, values between 0 and 127.

Bits 7:0 – TEC[7:0]: Transmit Error Counter
Actual state of the Transmit Error Counter, values between 0 and 255.

35.8.14. Protocol Status

Note:

1. When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in FLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.
2. The Bus_Off recovery sequence (see CAN Specification Rev. 2.0 or ISO 11898-1) cannot be shortened by setting or resetting CCCR.INIT. If the device goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0 Error code is written to PSR.LEC, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. ECR.REC is used to count these sequences.

Name: PSR
Offset: 0x44
Reset: 0x00000707
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		TDCV[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	1	1	1

Bit	7	6	5	4	3	2	1	0
	BO	EW	EP	ACT[1:0]		LEC[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	1	1

Bits 22:16 – TDCV[6:0]: Transmitter Delay Compensation Value

Value	Description
0x00 - 0x7F	Position of the secondary sample point, defined by the sum of the measured delay from CAN_TX to CAN_RX and TDCR.TDCO. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.

Bit 14 – PXE: Protocol Exception Event
This field is cleared on read access.

Value	Description
0	No protocol exception event occurred since last read access.
1	Protocol exception event occurred.

Bit 13 – RFDF: Received a CAN FD Message
This field is cleared on read access.

Value	Description
0	Since this bit was reset by the CPU, no CAN FD message has been received.
1	Message in CAN FD format with FDF flag set has been received. This bit is set independent of acceptance filtering.

Bit 12 – RBRS: BRS flag of last received CAN FD Message
This field is cleared on read access.

Value	Description
0	Last received CAN FD message did not have its BRS flag set.
1	Last received CAN FD message had its BRS flag set. This bit is set together with RFDF, independent of acceptance filtering.

Bit 11 – RESI: ESI flag of last received CAN FD Message
This field is cleared on read access.

Value	Description
0	Last received CAN FD message did not have its ESI flag set.
1	Last received CAN FD message had its ESI flag set.

Bits 10:8 – DLEC[2:0]: Data Last Error Code
Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

Bit 7 – BO: Bus_Off Status

Value	Description
0	The CAN is not Bus_Off.
1	The CAN is in Bus_Off state.

Bit 6 – EW: Error Warning Status

Value	Description
0	Both error counters are below the Error_Warning limit of 96.
1	At least one of the error counter has reached the Error_Warning limit of 96.

Bit 5 – EP: Error Passive

Value	Description
0	The CAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected.
1	The CAN is in the Error_Passive state.

Bits 4:3 – ACT[1:0]: Activity
Monitors the module's CAN communication state.

Value	Name	Description
0x0	SYNC	Node is synchronizing on CAN communication.
0x1	IDLE	Node is neither receiver nor transmitter.
0x2	RX	Node is operating as receiver.
0x3	TX	Node is operating as transmitter.

Bits 2:0 – LEC[2:0]: Last Error Code
The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error.

This field is set on read access.

Value	Name	Description
0x0	NONE	No Error: No error occurred since LEC has been reset by successful reception or transmission.
0x1	STUFF	Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
0x2	FORM	Form Error: A fixed format part of a received frame has the wrong format.
0x3	ACK	Ack Error: The message transmitted by the CAN was not acknowledged by another node.
0x4	BIT1	Bit1 Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus was dominant.
0x5	BIT0	Bit0 Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits have been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
0x6	CRC	CRC Error: The CRC checksum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.
0x7	NC	No Change: Any read access to the Protocol Status Register re-initializes the LEC to '7'. When the LEC shows the value '7', no CAN bus event was detected since the last CPU read access to the Protocol Status Register.

35.8.15. Transmitter Delay Compensation

Name: TDCR
Offset: 0x48
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		TDCO[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		TDCF[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 14:8 – TDCO[6:0]: Transmitter Delay Compensation Offset

Value	Description
0x00 - 0x7F	Offset value defining the distance between the measured delay from CAN_TX to CAN_RX and the secondary sample point. Valid values are 0 to 127 mtq.

Bits 6:0 – TDCF[6:0]: Transmitter Delay Compensation Filter Window Length

Value	Description
0x00 - 0x7F	Defines the minimum value for the SSP position, dominant edges on CAN_RX that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq.

35.8.16. Interrupt

The flags are set when one of the listed conditions is detected (edge-sensitive). A flag is cleared by writing a 1 to the corresponding bit field. Writing a 0 has no effect. A hard reset will clear the register.

Name: IR
Offset: 0x50
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			ARA	PED	PEA	WDI	BO	EW
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARA: Access to Reserved Address

Value	Description
0	No access to reserved address occurred.
1	Access to reserved address occurred.

Bit 28 – PED: Protocol Error in Data Phase

Value	Description
0	No protocol error in data phase.
1	Protocol error in data phase detected (PSR.DLEC != 0,7).

Bit 27 – PEA: Protocol Error in Arbitration Phase

Value	Description
0	No protocol error in arbitration phase.
1	Protocol error in arbitration phase detected (PSR.LEC != 0,7).

Bit 26 – WDI: Watchdog Interrupt

Value	Description
0	No Message RAM Watchdog event occurred.
1	Message RAM Watchdog event due to missing READY.

Bit 25 – BO: Bus_Off Status

Value	Description
0	Bus_Off status unchanged.
1	Bus_Off status changed.

Bit 24 – EW: Error Warning Status

Value	Description
0	Error_Warning status unchanged.
1	Error_Warning status changed.

Bit 23 – EP: Error Passive

Value	Description
0	Error_Passive status unchanged.
1	Error_Passive status changed.

Bit 22 – ELO: Error Logging Overflow

Value	Description
0	CAN Error Logging Counter did not overflow.
1	Overflow of CAN Error Logging Counter occurred.

Bit 21 – BEU: Bit Error Uncorrected
Message RAM bit error detected, uncorrected. Generated by an optional external parity / ECC logic attached to the Message RAM. An uncorrected Message RAM bit sets CCCR.INIT to 1. This is done to avoid transmission of corrupted data.

Value	Description
0	Not bit error detected when reading from Message RAM.
1	Bit error detected, uncorrected (e.g. parity logic).

Bit 20 – BEC: Bit Error Corrected
Message RAM bit error detected and corrected. Generated by an optional external parity / ECC logic attached to the Message RAM.

Value	Description
0	Not bit error detected when reading from Message RAM.
1	Bit error detected and corrected (e.g. ECC).

Bit 19 – DRX: Message stored to Dedicated Rx Buffer
The flag is set whenever a received message has been stored into a dedicated Rx Buffer.

Value	Description
0	No Rx Buffer updated.
1	At least one received message stored into a Rx Buffer.

Bit 18 – TOO: Timeout Occurred

Value	Description
0	No timeout.
1	Timeout reached.

Bit 17 – MRAF: Message RAM Access Failure
The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.
- was not able to write a message to the Message RAM. In this case message storage is aborted.

In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.

The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the CAN is switched into Restricted Operation Mode. To leave Restricted Operation Mode, the Host CPU has to reset CCCR.ASM.

Value	Description
0	No Message RAM access failure occurred.
1	Message RAM access failure occurred.

Bit 16 – TSW: Timestamp Wraparound

Value	Description
0	No timestamp counter wrap-around.
1	Timestamp counter wrapped around.

Bit 15 – TEFL: Tx Event FIFO Element Lost

Value	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

Bit 14 – TEFF: Tx Event FIFO Full

Value	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

Bit 13 – TEFW: Tx Event FIFO Watermark Reached

Value	Description
0	Tx Event FIFO fill level below watermark.
1	Tx Event FIFO fill level reached watermark.

Bit 12 – TEFN: Tx Event FIFO New Entry

Value	Description
0	Tx Event FIFO unchanged.
1	Tx Handler wrote Tx Event FIFO element.

Bit 11 – TFE: Tx FIFO Empty

Value	Description
0	Tx FIFO non-empty.
1	Tx FIFO empty.

Bit 10 – TCF: Transmission Cancellation Finished

Value	Description
0	No transmission cancellation finished.
1	Transmission cancellation finished.

Bit 9 – TC: Timestamp Completed

Value	Description
0	No transmission completed.
1	Transmission completed.

Bit 8 – HPM: High Priority Message

Value	Description
0	No high priority message received.
1	High priority message received.

Bit 7 – RF1L: Rx FIFO 1 Message Lost

Value	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. also set after write attempt to Rx FIFO 1 of size zero.

Bit 6 – RF1F: Rx FIFO 1 Full

Value	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

Bit 5 – RF1W: Rx FIFO 1 Watermark Reached

Value	Description
0	Rx FIFO 1 fill level below watermark.
1	Rx FIFO 1 fill level reached watermark.

Bit 4 – RF1N: Rx FIFO 1 New Message

Value	Description
0	No new message written to Rx FIFO 1.
1	New message written to Rx FIFO 1.

Bit 3 – RF0L: Rx FIFO 0 Message Lost

Value	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. also set after write attempt to Rx FIFO 0 of size zero.

Bit 2 – RF0F: Rx FIFO 0 Full

Value	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

Bit 1 – RF0W: Rx FIFO 0 Watermark Reached

Value	Description
0	Rx FIFO 0 fill level below watermark.
1	Rx FIFO 0 fill level reached watermark.

Bit 0 – RF0N: Rx FIFO 0 New Message

Value	Description
0	No new message written to Rx FIFO 0.
1	New message written to Rx FIFO 0.

35.8.17. Interrupt Enable

The settings in the Interrupt Enable register determine which status changes in the Interrupt Register will be signalled on an interrupt line.

Name: IE
Offset: 0x54
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			ARAE	PEDE	PEAE	WDIE	BOE	EWE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAE: Access to Reserved Address Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 28 – PEDE: Protocol Error in Data Phase Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 27 – PEAE: Protocol Error in Arbitration Phase Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 26 – WDIE: Watchdog Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 25 – BOE: Bus_Off Status Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 24 – EWE: Error Warning Status Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 23 – EPE: Error Passive Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 22 – ELOE: Error Logging Overflow Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 21 – BEUE: Bit Error Uncorrected Interrupt Enable.

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 20 – BECE: Bit Error Corrected Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 19 – DRXE: Message stored to Dedicated Rx Buffer Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 18 – TOOE: Timeout Occurred Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 17 – MRAFE: Message RAM Access Failure Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 16 – TSWE: Timestamp Wraparound Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 15 – TEFLE: Tx Event FIFO Event Lost Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 14 – TEFFE: Tx Event FIFO Full Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 13 – TEFWE: Tx Event FIFO Watermark Reached Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 12 – TEFNE: Tx Event FIFO New Entry Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 11 – TFEE: Tx FIFO Empty Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 10 – TCFE: Transmission Cancellation Finished Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 9 – TCE: Transmission Completed Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 8 – HPME: High Priority Message Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 7 – RF1LE: Rx FIFO 1 Message Lost Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 6 – RF1FE: Rx FIFO 1 Full Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 5 – RF1WE: Rx FIFO 1 Watermark Reached Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 4 – RF1NE: Rx FIFO 1 New Message Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 3 – RF0LE: Rx FIFO 0 Message Lost Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 2 – RF0FE: Rx FIFO 0 Full Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 1 – RF0WE: Rx FIFO 0 Watermark Reached Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 0 – RF0NE: Rx FIFO 0 New Message Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

35.8.18. Interrupt Line Select

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from IR to one of the two module interrupt lines.

Name: ILS
Offset: 0x58
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			ARAL	PEDL	PEAL	WDIL	BOL	EWL
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAL: Access to Reserved Address Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 28 – PEDL: Protocol Error in Data Phase Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 27 – PEAL: Protocol Error in Arbitration Phase Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 26 – WDIL: Watchdog Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 25 – BOL: Bus_Off Status Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 24 – EWL: Error Warning Status Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 23 – EPL: Error Passive Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 22 – ELOL: Error Logging Overflow Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 21 – BEUL: Bit Error Uncorrected Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 20 – BECL: Bit Error Corrected Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 19 – DRXL: Message stored to Dedicated Rx Buffer Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 18 – TOOL: Timeout Occurred Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 17 – MRAFL: Message RAM Access Failure Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 16 – TSWL: Timestamp Wraparound Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 15 – TEFLL: Tx Event FIFO Event Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 14 – TEFFL: Tx Event FIFO Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 13 – TEFWL: Tx Event FIFO Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 12 – TEFNL: Tx Event FIFO New Entry Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 11 – TFEL: Tx FIFO Empty Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 10 – TCFL: Transmission Cancellation Finished Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 9 – TCL: Transmission Completed Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 8 – HPML: High Priority Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 7 – RF1LL: Rx FIFO 1 Message Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 6 – RF1FL: Rx FIFO 1 Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 5 – RF1WL: Rx FIFO 1 Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 4 – RF1NL: Rx FIFO 1 New Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 3 – RF0LL: Rx FIFO 0 Message Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 2 – RF0FL: Rx FIFO 0 Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 1 – RF0WL: Rx FIFO 0 Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 0 – RF0NL: Rx FIFO 0 New Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

35.8.19. Interrupt Line Enable

Name: ILE
Offset: 0x5C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							EINT1	EINT0
Access							R/W	R/W
Reset							0	0

Bits 1:0 – EINTn: Enable Interrupt Line n [n = 1,0]

Value	Description
0	CAN interrupt line n disabled.
1	CAN interrupt line n enabled.

35.8.20. Global Filter Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Name: GFC
Offset: 0x80
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:4 – ANFS[1:0]: Accept Non-matching Frames Standard

Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0x0	RXF0	Accept in Rx FIFO 0.
0x1	RXF1	Accept in Rx FIFO 1.
0x2 or 0x3	REJECT	Reject

Bits 3:2 – ANFE[1:0]: Accept Non-matching Frames Extended

Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0x0	RXF0	Accept in Rx FIFO 0.
0x1	RXF1	Accept in Rx FIFO 1.
0x2 or 0x3	REJECT	Reject

Bit 1 – RRFS: Reject Remote Frames Standard

Value	Description
0	Filter remote frames with 11-bit standard IDs.
1	Reject all remote frames with 11-bit standard IDs.

Bit 0 – RRFE: Reject Remote Frames Extended

Value	Description
0	Filter remote frames with 29-bit extended IDs.
1	Reject all remote frames with 29-bit extended IDS.

35.8.21. Standard ID Filter Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Name: SIDFC
Offset: 0x84
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LSS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FLSSA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FLSSA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – LSS[7:0]: List Size Standard

Value	Description
0	No standard Message ID filter.
1 - 128	Number of standard Message ID filter elements.
> 128	Values greater than 128 are interpreted as 128.

Bits 15:0 – FLSSA[15:0]: Filter List Standard Start Address

Start address of standard Message ID filter list. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.8.22. Extended ID Filter Configuration

Name: XIDFC
Offset: 0x88
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		LSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FLESA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FLESA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 22:16 – LSE[6:0]: List Size Extended

Value	Description
0	No extended Message ID filter.
1 - 64	Number of Extended Message ID filter elements.
> 64	Values greater than 64 are interpreted as 64.

Bits 15:0 – FLESA[15:0]: Filter List Extended Start Address
Start address of extended Message ID filter list. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.8.23. Extended ID AND Mask

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Name: XIDAM
Offset: 0x90
Reset: 0x1FFFFFFF
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
				EIDM[28:24]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	EIDM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	EIDM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	EIDM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 28:0 – EIDM[28:0]: Extended ID Mask

For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

35.8.24. High Priority Message Status

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Name: HPMS
Offset: 0x94
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FLST	FIDX[6:0]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSI[1:0]		BIDX[5:0]					
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – FLST: Filter List

Indicates the filter list of the matching filter element.

Value	Description
0	Standard Filter List.
1	Extended Filter List.

Bits 14:8 – FIDX[6:0]: Filter Index

Index of matching filter element. Range is 0 to SIDFC.LSS - 1 (standard) or XIDFC.LSE - 1 (extended).

Bits 7:6 – MSI[1:0]: Message Storage Indicator

This field defines the message storage information to a FIFO.

Value	Name	Description
0x0	NONE	No FIFO selected.
0x1	LOST	FIFO message lost.

Value	Name	Description
0x2	FIFO0	Message stored in FIFO 0.
0x3	FIFO1	Message stored in FIFO 1.

Bits 5:0 – BIDX[5:0]: Buffer Index
Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = 1.

35.8.25. New Data 1

Name: NDAT1
Offset: 0x98
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NDn: New Data n [n = 0..31]
The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.

35.8.26. New Data 2

Name: NDAT2
Offset: 0x9C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NDn: New Data [n = 32..64]
The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.

35.8.27. Rx FIFO 0 Configuration

Name: RXF0C
Offset: 0xA0
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
	F0OM	F0WM[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		F0S[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	F0SA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	F0SA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – F0OM: FIFO 0 Operation Mode
FIFO 0 can be operated in blocking or in overwrite mode.

Value	Description
0	FIFO 0 blocking mode.
1	FIFO 0 overwrite mode.

Bits 30:24 – F0WM[6:0]: Rx FIFO 0 Watermark

Value	Description
0	Watermark interrupt disabled.
1 - 64	Level for Rx FIFO 0 watermark interrupt (IR.RF0W).
>64	Watermark interrupt disabled.

Bits 22:16 – F0S[6:0]: Rx FIFO 0 Size
The Rx FIFO 0 elements are indexed from 0 to F0S - 1.

Value	Description
0	No Rx FIFO 0
1 - 64	Number of Rx FIFO 0 elements.
>64	Values greater than 64 are interpreted as 64.

Bits 15:0 – F0SA[15:0]: Rx FIFO 0 Start Address
Start address of Rx FIFO 0 in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.8.28. Rx FIFO 0 Status

Name: RXF0S
Offset: 0xA4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							RF0L	F0F
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			F0PI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			F0GI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		F0FL[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 25 – RF0L: Rx FIFO 0 Message Lost
This bit is a copy of interrupt flag IR.RF0L. When IR.RF0L is reset, this bit is also reset.

Overwriting the oldest message when RXF0C.F0OM = ‘1’ will not set this flag.

Value	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero.

Bit 24 – F0F: Rx FIFO 0 Full

Value	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

Bits 21:16 – F0PI[5:0]: Rx FIFO 0 Put Index
Rx FIFO 0 write index pointer, range 0 to 63.

Bits 13:8 – F0GI[5:0]: Rx FIFO 0 Get Index
Rx FIFO 0 read index pointer, range 0 to 63.

Bits 6:0 – F0FL[6:0]: Rx FIFO 0 Fill Level
Number of elements stored in Rx FIFO 0, range 0 to 64.

35.8.29. Rx FIFO 0 Acknowledge

Name: RXF0A
Offset: 0xA8
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			F0AI[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – F0AI[5:0]: Rx FIFO 0 Acknowledge Index
After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL.

35.8.30. Rx Buffer Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Name: RXBC
Offset: 0xAC
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RBSA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RBSA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RBSA[15:0]: Rx Buffer Start Address

Configures the start address of the Rx Buffers section in the Message RAM. Also used to reference debug message A,B,C. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.8.31. Rx FIFO 1 Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Name: RXF1C
Offset: 0xB0
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
	F1OM	F1WM[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		F1S[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	F1SA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	F1SA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – F1OM: FIFO 1 Operation Mode

FIFO 1 can be operated in blocking or in overwrite mode.

Value	Description
0	FIFO 1 blocking mode.
1	FIFO 1 overwrite mode.

Bits 30:24 – F1WM[6:0]: Rx FIFO 1 Watermark

Value	Description
0	Watermark interrupt disabled.
1 - 64	Level for Rx FIFO 1 watermark interrupt (IR.RF1W).
>64	Watermark interrupt disabled.

Bits 22:16 – F1S[6:0]: Rx FIFO 1 Size

The Rx FIFO 1 elements are indexed from 0 to F1S - 1.

Value	Description
0	No Rx FIFO 1
1 - 64	Number of Rx FIFO 1 elements.
>64	Values greater than 64 are interpreted as 64.

Bits 15:0 – F1SA[15:0]: Rx FIFO 1 Start Address
Start address of Rx FIFO 1 in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.8.32. Rx FIFO 1 Status

Name: RXF1S
Offset: 0xB4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DMS[1:0]						RF1L	F1F
Access	R	R					R	R
Reset	0	0					0	0

Bit	23	22	21	20	19	18	17	16
			F1PI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			F1GI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		F1FL[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 31:30 – DMS[1:0]: Debug Message Status
This field defines the debug message status.

Value	Name	Description
0x0	IDLE	Idle state, wait for reception of debug messages, DMA request is cleared.
0x1	DBGA	Debug message A received.
0x2	DBGB	Debug message A, B received.
0x3	DBGC	Debug message A, B, C received, DMA request is set.

Bit 25 – RF1L: Rx FIFO 1 Message Lost
This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset.

Overwriting the oldest message when RXF1C.F0OM = ‘1’ will not set this flag.

Value	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero.

Bit 24 – F1F: Rx FIFO 1 Full

Value	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

Bits 21:16 – F1PI[5:0]: Rx FIFO 1 Put Index
Rx FIFO 1 write index pointer, range 0 to 63.

Bits 13:8 – F1GI[5:0]: Rx FIFO 1 Get Index
Rx FIFO 1 read index pointer, range 0 to 63.

Bits 6:0 – F1FL[6:0]: Rx FIFO 1 Fill Level
Number of elements stored in Rx FIFO 1, range 0 to 64.

35.8.33. Rx FIFO 1 Acknowledge

Name: RXF1A
Offset: 0xB8
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			F1AI[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – F1AI[5:0]: Rx FIFO 1 Acknowledge Index
After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F0GI to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL.

35.8.34. Rx Buffer / FIFO Element Size Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Configures the number of data bytes belonging to an Rx Buffer / Rx FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

Name: RXESC
Offset: 0xBC
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						RBDS[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
		F1DS[2:0]				F0DS[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 10:8 – RBDS[2:0]: Rx Buffer Data Field Size

In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer, only the number of bytes as configured by RXESC are stored to the Rx Buffer element. The rest of the frame's data field is ignored.

Value	Name	Description
0x0	DATA8	8 byte data field.
0x1	DATA12	12 byte data field.
0x2	DATA16	16 byte data field.
0x3	DATA20	20 byte data field.
0x4	DATA24	24 byte data field.
0x5	DATA32	32 byte data field.
0x6	DATA48	48 byte data field.
0x7	DATA64	64 byte data field.

Bits 6:4 – F1DS[2:0]: Rx FIFO 1 Data Field Size
In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx FIFO 1, only the number of bytes as configured by RXESC are stored to the Rx FIFO 1 element. The rest of the frame's data field is ignored.

Value	Name	Description
0x0	DATA8	8 byte data field.
0x1	DATA12	12 byte data field.
0x2	DATA16	16 byte data field.
0x3	DATA20	20 byte data field.
0x4	DATA24	24 byte data field.
0x5	DATA32	32 byte data field.
0x6	DATA48	48 byte data field.
0x7	DATA64	64 byte data field.

Bits 2:0 – F0DS[2:0]: Rx FIFO 0 Data Field Size
In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx FIFO 0, only the number of bytes as configured by RXESC are stored to the Rx FIFO 0 element. The rest of the frame's data field is ignored.

Value	Name	Description
0x0	DATA8	8 byte data field.
0x1	DATA12	12 byte data field.
0x2	DATA16	16 byte data field.
0x3	DATA20	20 byte data field.
0x4	DATA24	24 byte data field.
0x5	DATA32	32 byte data field.
0x6	DATA48	48 byte data field.
0x7	DATA64	64 byte data field.

35.8.35. Tx Buffer Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Note: Be aware that the sum of TFQS and NDTB may not be greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

Name: TXBC
Offset: 0xC0
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
		TFQM	TFQS[5:0]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			NDTB[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TBSA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TBSA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 30 – TFQM: Tx FIFO/Queue Mode

Value	Description
0	Tx FIFO operation.
1	Tx Queue operation.

Bits 29:24 – TFQS[5:0]: Transmit FIFO/Queue Size

Value	Description
0	No Tx FIFO/Queue.
1 - 32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

Bits 21:16 – NDTB[5:0]: Number of Dedicated Transmit Buffers

Value	Description
0	No Tx FIFO/Queue.
1 - 32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

Bits 15:0 – TBSA[15:0]: Tx Buffers Start Address
Start address of Tx Buffers section in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.8.36. Tx FIFO/Queue Status

Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indexes indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

Name: TXFQS
Offset: 0xC4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TFQF	TFQPI[4:0]				
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				TFGI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TFFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 21 – TFQF: Tx FIFO/Queue Full

Value	Description
0	Tx FIFO/Queue not full.
1	Tx FIFO/Queue full.

Bits 20:16 – TFQPI[4:0]: Tx FIFO/Queue Put Index
Tx FIFO/Queue write index pointer, range 0 to 31.

Bits 12:8 – TFGI[4:0]: Tx FIFO/Queue Get Index
Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM = ‘1’).

Bits 5:0 – TFFL[5:0]: Tx FIFO Free Level
Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM = ‘1’).

35.8.37. Tx Buffer Element Size Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes >8 bytes are intended for CAN FD operation only.

Name: TXESC
Offset: 0xC8
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						TBDS[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – TBDS[2:0]: Tx Buffer Data Field Size

In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as “0xCC” (padding bytes).

Value	Name	Description
0x0	DATA8	8 byte data field.
0x1	DATA12	12 byte data field.
0x2	DATA16	16 byte data field.
0x3	DATA20	20 byte data field.
0x4	DATA24	24 byte data field.
0x5	DATA32	32 byte data field.
0x6	DATA48	48 byte data field.
0x7	DATA64	64 byte data field.

35.8.38. Tx Buffer Request Pending

Note: TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is canceled immediately, the corresponding TXBRP bit is reset.

Name: TXBRP
Offset: 0xCC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TRPn: Transmission Request Pending
Each Tx Buffer has its own Transmission Request Pending bit.

The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR.

TXBRP bits are set only for those Tx Buffers configured via TXBC. After a TXBRP bit has been set, a Tx scan is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).

A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset.

After a cancellation has been requested, a finished cancellation is signaled via TXBCF

- after successful transmission together with the corresponding TXBTO bit
- when the transmission has not yet been started at the point of cancellation
- when the transmission has been aborted due to lost arbitration
- when an error occurred during frame transmission

In DAR mode all transmissions are automatically canceled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions.

Value	Description
0	No transmission request pending.
1	Transmission request pending.

35.8.39. Tx Buffer Add Request

Note: If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit is already set), this add request is ignored.

Name: TXBAR
Offset: 0xD0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ARn: Add Request
Each Tx Buffer has its own Add Request bit.

Writing a ‘1’ will set the corresponding Add Request bit; writing a ‘0’ has no impact. This enables the Host to set transmission requests for multiple Tx Buffers with one write to TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.

35.8.40. Tx Buffer Cancellation Request

Name: TXBCR
Offset: 0xD4
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRn: Cancellation Request
Each Tx Buffer has its own Cancellation Request bit.

Writing a ‘1’ will set the corresponding Cancellation Request bit; writing a ‘0’ has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset.

Value	Description
0	No cancellation pending.
1	Cancellation pending.

35.8.41. Tx Buffer Transmission Occurred

Name: TXBTO
Offset: 0xD8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TOn: Transmission Occurred
Each Tx Buffer has its own Transmission Occurred bit.

The bits are set when the corresponding TXBRP bit is cleared after a successful transmission.

The bits are reset when a new transmission is requested by writing ‘1’ to the corresponding bit of register TXBAR.

35.8.42. Tx Buffer Cancellation Finished

Name: TXBCF
Offset: 0xDC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CFn: Cancellation Finished
Each Tx Buffer has its own Cancellation Finished bit.

The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately.

The bits are reset when a new transmission is requested by writing '1' to the corresponding bit of register TXBAR.

35.8.43. Tx Buffer Transmission Interrupt Enable

Name: TXBTIE
Offset: 0xE0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TIEn: Transmission Interrupt Enable
Each Tx Buffer has its own Transmission Interrupt Enable bit.

Value	Description
0	Transmission interrupt disabled.
1	Transmission interrupt enabled.

35.8.44. Tx Buffer Cancellation Finished Interrupt Enable

Name: TXBCIE
Offset: 0xE4
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CFIEn: Cancellation Finished Interrupt Enable
Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.

Value	Description
0	Cancellation finished interrupt disabled.
1	Cancellation finished interrupt enabled.

35.8.45. Tx Event FIFO Configuration

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Name: TXEFC
Offset: 0xF0
Reset: 0x00000000
Property: Write-restricted

Bit	31	30	29	28	27	26	25	24
			EFWM[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			EFS[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EFSA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EFSA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – EFWM[5:0]: Event FIFO Watermark

Value	Description
0	Watermark interrupt disabled.
1 - 32	Level for Tx Event FIFO watermark interrupt (IR.TEFW).
>32	Watermark interrupt disabled.

Bits 21:16 – EFS[5:0]: Event FIFO Size

The Tx Event FIFO elements are indexed from 0 to EFS - 1.

Value	Description
0	Tx Event FIFO disabled
1 - 32	Number of Tx Event FIFO elements.
>32	Values greater than 32 are interpreted as 32.

Bits 15:0 – EFSA[15:0]: Event FIFO Start Address

Start address of Tx Event FIFO in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e.

only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.8.46. Tx Event FIFO Status

Name: TXEFS
Offset: 0xF4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							TEFL	EFF
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
				EFP[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				EFGI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				EFFI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 25 – TEFL: Tx Event FIFO Element Lost
This bit is a copy of interrupt flag IR.TEFL. When IR.TEFL is reset, this bit is also reset.

Value	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

Bit 24 – EFF: Event FIFO Full

Value	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

Bits 20:16 – EFP[4:0]: Event FIFO Put Index
Tx Event FIFO write index pointer, range 0 to 31.

Bits 12:8 – EFGI[4:0]: Event FIFO Get Index
Tx Event FIFO read index pointer, range 0 to 31.

Bits 4:0 – EFFI[4:0]: Event FIFO Fill Level
Number of elements stored in Tx Event FIFO, range 0 to 32.

35.8.47. Tx Event FIFO Acknowledge

Name: TXEFA
Offset: 0xF8
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				EFAI[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – EFAI[4:0]: Event FIFO Acknowledge Index
After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI to EFAI + 1 and update the FIFO 0 Fill Level TXEFS.EFFL.

35.9. Message RAM

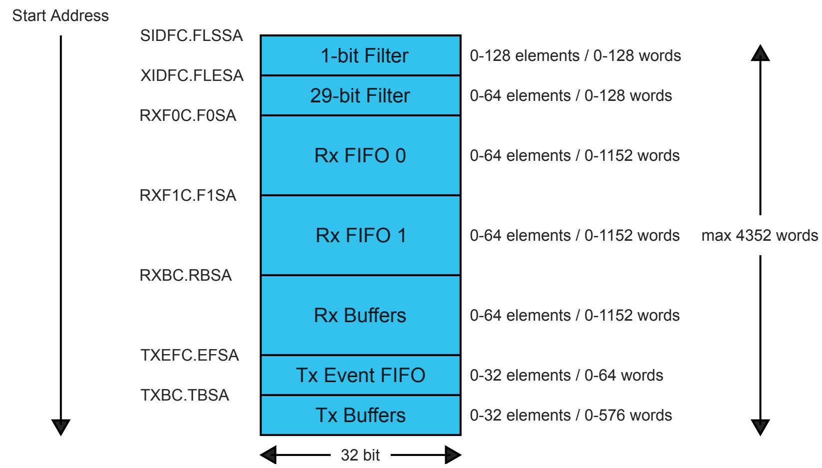
For storage of Rx/Tx messages and for storage of the filter configuration a single- or dual-ported Message RAM has to be connected to the CAN module.

35.9.1. Message RAM Configuration

The Message RAM has a width of 32 bits. In case parity checking or ECC is used a respective number of bits has to be added to each word. The CAN module can be configured to allocate up to 4352 words in the Message RAM. It is not necessary to configure each of the sections listed in the figure below, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode the required Message RAM size strongly depends on the element size configured for Rx FIFO 0, Rx FIFO 1, Rx Buffers, and Tx Buffers via RXESC.F0DS, RXESC.F1DS, RXESC.RBDS, and TXESC.TBDS.

Figure 35-12. Message RAM Configuration



When the CAN addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses (i.e. only bits 15 to 2 are evaluated and the two LSBs are ignored).

Warning: The CAN does not check for erroneous configuration of the Message RAM. Especially the configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully to avoid falsification or loss of data.

35.9.2. Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in the table below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESC.

Table 35-8. Rx Buffer and FIFO Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	ESI	XTD	RTR	ID[28:0]																												
R1	ANMF	FIDX[6:0]									FDF	BRS	DLC[3:0]				RXTS[15:0]															
R2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
R3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
...	...								...								...								...							
Rn	DBm[7:0]								DBm-1[7:0]								DBm-2[7:0]								DBm-3[7:0]							

- R0 Bit 31 - ESI: Error State Indicator

  0 : Transmitting node is error active.

1 : Transmitting node is error passive.

- R0 Bit 30 - XTD: Extended Identifier

  Signals to the Host whether the received frame has a standard or extended identifier.

  0 : 11-bit standard identifier.

  1 : 29-bit extended identifier.

- R0 Bit 29 - RTR: Remote Transmission Request

  Signals to the Host whether the received fram is a data fram or a remote frame.

  0 : Received frame is a data frame.

  1 : Received frame is a remote frame.

  Note: There are no remote frames in CAN FD format. In case a CAN FD frame was received (EDL = ‘1’), bit RTR reflects the state of the reserved bit r1.

- R0 Bits 28:0 - ID[28:0]: Identifier

  Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

- R1 Bit 31 - ANMF: Accepted Non-matching Frame

  Acceptance of non-matching frames may be enabled via GFC.ANFS and GFC.ANFE.

  0 : Received frame matching filter index FIDX.

  1 : Received frame did not match any Rx filter element.

- R1 Bits 30:24 - FIDX[6:0]: Filter Index

  0-127 : Index of matching Rx acceptance filter element (invalid if ANMF = ‘1’).

  Note: Range is 0 to SIDFC.LSS-1 for standard and 0 to XIDFC.LSE-1 for extended.

- R1 Bits 23:22 - Reserved
- R1 Bit 21 - FDF: FD Format

  0 : Standard frame format.

  1 : CAN FD frame format (new DLC-coding and CRC).

- R1 Bit 20 - BRS: Bit Rate Search

  0 : Frame received without bit rate switching.

  1 : Frame received with bit rate switching.

- R1 Bits 19:16 - DLC[3:0]: Data Length Code

  0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

  9-15 : CAN: received frame has 8 data bytes.

  9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

- R1 Bits 15:0 - RXTS[15:0]: Rx Timestamp

  Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP.

- R2 Bits 31:24 - DB3[7:0]: Data Byte 3
- R2 Bits 23:16 - DB2[7:0]: Data Byte 2

- R2 Bits 15:8 - DB1[7:0]: Data Byte 1
- R2 Bits 7:0 - DB0[7:0]: Data Byte 0
- R3 Bits 31:24 - DB7[7:0]: Data Byte 7
- R3 Bits 23:16 - DB6[7:0]: Data Byte 6
- R3 Bits 15:8 - DB5[7:0]: Data Byte 5
- R3 Bits 7:0 - DB4[7:0]: Data Byte 4

...

- Rn Bits 31:24 - DBm[7:0]: Data Byte m
- Rn Bits 23:16 - DBm-1[7:0]: Data Byte m-1
- Rn Bits 15:8 - DBm-2[7:0]: Data Byte m-2
- Rn Bits 7:0 - DBm-3[7:0]: Data Byte m-3

Warning: Depending on the configuration of RXESC, between two and sixteen 32-bit words (Rn = 3 ... 17) are used for storage of a CAN message's data field.

35.9.3. Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS and TXBC.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 35-9. Tx Buffer Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T0	ESI	XTD	RTR	ID[28:0]																												
T1	MM[7:0]								EFC		FDF	BRS	DLC[3:0]																			
T2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
T3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
...	...								...								...								...							
Tn	DBm[7:0]								DBm-1[7:0]								DBm-2[7:0]								DBm-3[7:0]							

- T0 Bit 31 - ESI: Error State Indicator

  0 : ESI bit in CAN FD format depends only on error passive flag.

  1 : ESI bit in CAN FD format transmitted recessive.

  Note: The ESI bit of the transmit buffer is OR'ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.

- T0 Bit 30 - XTD: Extended Identifier

  0 : 11-bit standard identifier.

  1 : 29-bit extended identifier.
- T0 Bit 29 - RTR: Remote Transmission Request

  0 : Transmit data frame.

  1 : Transmit remote frame.

  Note: When RTR = ‘1’, the CAN transmits a remote frame according to ISO 11898-1, even if CCCR.CME enables the transmission in CAN FD format.
- T0 Bits 28:0 - ID[28:0]: Identifier

  Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
- T1 Bits 31:24 - MM[7:0]: Message Marker

  Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.
- T1 Bit 23 - EFC: Event FIFO Control

  0 : Don’t store Tx events.

  1 : Store Tx events.
- T1 Bit 22 - Reserved
- TR1 Bit 21 - FDF: FD Format

  0 : Frame transmitted in Classic CAN format.

  1 : Frame transmitted in CAN FD format.
- T1 Bit 20 - BRS: Bit Rate Search

  0 : CAN FD frames transmitted without bit rate switching.

  1 : CAN FD frames transmitted with bit rate switching.

  Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled CCCR.FDOE = ‘1’. Bit BRS is only evaluated when in addition CCCR.BRSE = ‘1’.
- T1 Bits 19:16 - DLC[3:0]: Data Length Code

  0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

  9-15 : CAN: received frame has 8 data bytes.

  9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.
- T1 Bits 15:0 - Reserved
- T2 Bits 31:24 - DB3[7:0]: Data Byte 3
- T2 Bits 23:16 - DB2[7:0]: Data Byte 2
- T2 Bits 15:8 - DB1[7:0]: Data Byte 1
- T2 Bits 7:0 - DB0[7:0]: Data Byte 0
- T3 Bits 31:24 - DB7[7:0]: Data Byte 7
- T3 Bits 23:16 - DB6[7:0]: Data Byte 6
- T3 Bits 15:8 - DB5[7:0]: Data Byte 5
- T3 Bits 7:0 - DB4[7:0]: Data Byte 4

  ...

- Tn Bits 31:24 - DBm[7:0]: Data Byte m
- Tn Bits 23:16 - DBm-1[7:0]: Data Byte m-1
- Tn Bits 15:8 - DBm-2[7:0]: Data Byte m-2
- Tn Bits 7:0 - DBm-3[7:0]: Data Byte m-3

Note: Depending on the configuration of TXESC, between two and sixteen 32-bit words (Tn = 3 ... 17) are used for storage of a CAN message’s data field.

35.9.4. Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.

Table 35-10. Tx Event FIFO Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E0	ESI	XTD	RTR	ID[28:0]																												
E1	MM[7:0]								ET[1:0]		FDF	BRS	DLC[3:0]				TXTS[15:0]															

- E0 Bit 31 - ESI: Error State Indicator

  0 : Transmitting node is error active.

  1 : Transmitting node is error passive.
- E0 Bit 30 - XTD: Extended Identifier

  0 : 11-bit standard identifier.

  1 : 29-bit extended identifier.
- E0 Bit 29 - RTR: Remote Transmission Request

  0 : Received frame is a data frame.

  1 : Received frame is a remote frame.
- E0 Bits 28:0 - ID[28:0]: Identifier

  Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
- E1 Bits 31:24 - MM[7:0]: Message Marker

  Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.
- E1 Bits 23:22 - ET[1:0]: Event Type

  This field defines the event type.

Table 35-11. Event Type

Value	Name	Description
0x0 or 0x3	RES	Reserved
0x1	TXE	Tx event
0x2	TXC	Transmission in spite of cancellation (always set for transmission in DAR mode)

- E1 Bit 21 - FDF: FD Format

  0 : Standard frame format.

  1 : CAN FD frame format (new DLC-coding and CRC).
- E1 Bit 20 - BRS: Bit Rate Search

  0 : Frame received without bit rate switching.

  1 : Frame received with bit rate switching.
- E1 Bits 19:16 - DLC[3:0]: Data Length Code

  0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

  9-15 : CAN: received frame has 8 data bytes.

  9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.
- E1 Bits 15:0 - TXTS[15:0]: Tx Timestamp

  Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP.

35.9.5. Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address SIDFC.FLSSA plus the index of the filter element (0 ... 127).

Table 35-12. Standard Message ID Filter Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S0	SFT[1:0]		SFEC[2:0]			SFID1[10:0]																SFID2[10:0]										

- Bits 31:30 - SFT[1:0]: Standard Filter Type

  This field defines the standard filter type.

Table 35-13. Standard Filter Type

Value	Name	Description
0x0	RANGE	Range filter from SFID1 to SFID2 (SFID2 >= SFID1)
0x1	DUAL	Dual ID filter for SFID1 or SFID2
0x2	CLASSIC	Classic filter: SFID1 = filter, SFID2 = mask
0x3	RES	Reserved

- Bits 29:27 - SFEC[2:0]: Standard Filter Element Configuration

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = “100”, “101”, or “110” a match sets interrupt flag IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Table 35-14. Standard Filter Element Configuration

Value	Name	Description
0x0	DISABLE	Disable filter element
0x1	STF0M	Store in Rx FIFO 0 if filter matches
0x2	STF1M	Store in Rx FIFO 1 if filter matches
0x3	REJECT	Reject ID if filter matches
0x4	PRIORITY	Set priority if filter matches.
0x5	PRIF0M	Set priority and store in FIFO 0 if filter matches.
0x6	PRIF1M	Set priority and store in FIFO 1 if filter matches.
0x7	STRXBUF	Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored.

- Bits 26:16 - SFID1[10:0]: Standard Filter ID 1

  First ID of standard ID filter element.

  When filtering for Rx Buffers or for debug messages this field defines the ID of a standard mesage to be stored. The received identifiers must match exactly, no masking mechanism is used.
- Bits 15:11 - Reserved
- Bits 10:0 - SFID2[10:0]: Standard Filter ID 2

  This bit field has a different meaning depending on the configuration of SFEC.

  5.1. SFEC = “001” ... “110”: Second ID of standard ID filter element.

  5.2. SFEC = “111”: Filter for Rx Buffers or for debug messages.

  SFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.
  00 = Store message into an Rx Buffer
  01 = Debug Message A
  10 = Debug Message B
  11 = Debug Message C

  SFID2[8:6] is used to control the filter event pins at the Extension Interface. A ‘1’ at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one CLK_CAN_APB period in case the filter matches.

  SFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA for storage of a matching message.

35.9.6. Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address XIDFC.FLESA plus two times the index of the filter element (0…63).

Table 35-15. Extended Message ID Filter Element

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>F0</td><td colspan="3">EFEC[2:0]</td><td colspan="29">EFID1[28:0]</td></tr>
<tr><td>F1</td><td colspan="2">EFT[1:0]</td><td></td><td colspan="29">EFID2[28:0]</td></tr>
</table>

- F0 Bits 31:29 - EFEC[2:0]: Extended Filter Element Configuration

  All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110” a match sets interrupt flag IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Table 35-16. Extended Filter Element Configuration

Value	Name	Description
0x0	DISABLE	Disable filter element.
0x1	STF0M	Store in Rx FIFO 0 if filter matches.
0x2	STF1M	Store in Rx FIFO 1 if filter matches.
0x3	REJECT	Reject ID if filter matches.
0x4	PRIORITY	Set priority if filter matches.
0x5	PRIF0M	Set priority and store in FIFO 0 if filter matches.
0x6	PRIF1M	Set priority and store in FIFO 1 if filter matches.
0x7	STRXBUF	Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored.

- F0 Bits 28:0 - EFID1[28:0]: Extended Filter ID 1

  First ID of extended ID filter element.

  When filtering for Rx Buffers or for debug messages this field defines the ID of a extended mesage to be stored. The received identifiers must match exactly, only XIDAM masking mechanism is used.

- F1 Bits 31:30 - EFT[1:0]: Extended Filter Type

  This field defines the extended filter type.

Table 35-17. Extended Filter Type

Value	Name	Description
0x0	RANGEM	Range filter from EFID1 to EFID2 (EFID2 >= EFID1).
0x1	DUAL	Dual ID filter for EFID1 or EFID2.
0x2	CLASSIC	Classic filter: EFID1 = filter, EFID2 = mask.
0x3	RANGE	Range filter from EFID1 to EFID2 (EFID2 >= EFID1), XIDAM mask not applied.

- F1 Bits 28:0 - EFID2[28:0]: Extended Filter ID 2

This bit field has a different meaning depending on the configuration of EFEC.
1) EFEC = “001” ... “110” Second ID of standard ID filter element.
2) EFEC = “111” Filter for Rx Buffers or for debug messages.

EFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

00 = Store message into an Rx Buffer

01 = Debug Message A

10 = Debug Message B

11 = Debug Message C

EFID2[8:6] is used to control the filter event pins at the Extension Interface. A ‘1’ at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one CLK_CAN_APB period in case the filter matches.

EFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA for storage of a matching message.