Interface IP

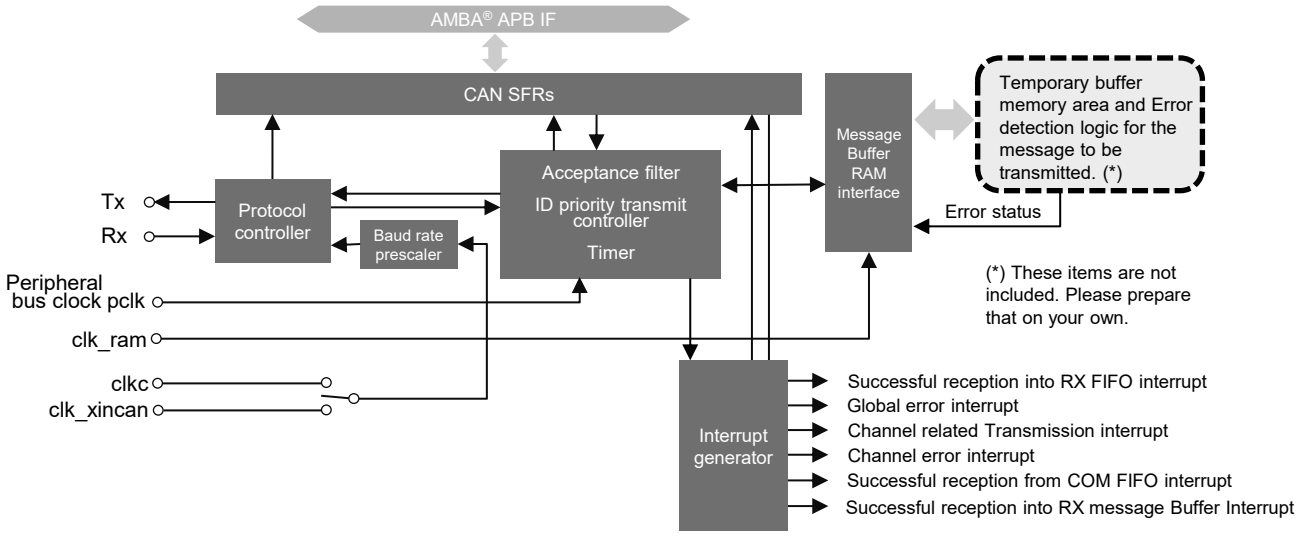
# CAN FD Controller
# Single-channel type

**IMPORTANT: When selling a device, please contact Bosch(© Robert Bosch GmbH) regarding CAN FD Protocol license.**

**CAUTION:**
**1. “CAN FD one channel version” means one channel version of a multi channel Renesas scalable CAN FD IP.**
**2.This IP does not include a temporary buffer memory array and Error detection logic for the message to be transmitted in 10.10, ISO11898. It includes only the Error detection of 10.11, ISO11898.**

## Overview

This IP is CAN FD one channel version controller that is compliant with ISO 11898-1 (2015) Specifications.
This CAN module embeds single CAN channel. This CAN module can be configurable to CAN FD or classic CAN, transmits and receives both ID formats of messages, namely the standard identifier (11 bits) (identifier is hereafter referred to as ID) and extended ID (29 bits).




Tx/Rx:
Input/Output pins of the CAN module

Protocol controller:
Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling, etc.

Acceptance filter:
Performs filtering of received messages. The entries in the Acceptance filter list RAM are used for the filtering process.

Timer:
Two timers
- Reception Timestamp function
- Transmission separation time for FIFO Buffers

Message Buffer RAM interface:
This RAM interface is used to store messages after reception or for transmission using a normal Message Buffer or a FIFO. Each message entry has an individual ID, data length code, data field, message pointer for upper layer application usage and a time stamp.

This RAM interface is also used to store the message acceptance filtering entries. Each acceptance filter entry has an individual ID, data length code, data field, message pointer for upper layer application usage and message direction pointer.

PMO-25-011
R06PF0009EJ0103

## Key Features

| Item | Specification |
|---|---|
| Communication | CAN functionality conform to CAN FD ISO 11898-1 (2015) |
| Protocol engine Version | RS-CANFD_PE V3.0 |
| Data transfer rate | up to 1Mbps for arbitration phase and up to 8Mbps for data phase , individually for CAN channel |
| Proposed min. operation frequency peripheral clock/APB clock | 80MHz |
| Data Link Layer clock (DLLC) | 8MHz ≤ max ≤ Peripheral (APB bus) clock Frequency |
| Input/Output pins | TX/RX |
| CAN channels | 1 |
| Selectable ID type | 11-bit Standard ID |
| | 11-bit Standard ID + 18-bit Extended ID |
| Selectable Frame Type | Data Frame (RTR = 0) (CAN and CAN FD frames) |
| | Remote Frame (RTR = 1) (only CAN frames) |
| Variable Data Byte Count for Data Frames | DLC range: 0 to F |
| Message Buffer RAM interface | Up to 32 reception message buffers |
| | 4 transmit message buffers |
| | 1 transmission queue<br>Automatic message transfer into transmission queues supported |
| FIFO number | 2 Reception FIFO Buffers<br>1 COMMON FIFO individually configurable as<br>- Reception FIFO<br>- Transmission FIFO |
| Automatic delay interval timer for transmission | The delay timer can be applied to:<br>- Transmission FIFO |
| Enhanced reception filtering | support of 11bit and 29bit CAN identifier |
| | programmable 29 bit CAN identifier acceptance filter mask for each entry |
| | programmable routing capability for each FIFO and reception message buffers (up to 2 routing destinations) |
| | RTR and IDE masking |
| | DLC filter |
| | Message buffer payload overload protection |
| | Payload filter |
| | Updating AFL entry during communication |
| General SW Support | Automatic label information added to receive message (for upper SW layer support) |
| Timer | TX and RX Time Stamp function |
| Power down function | Module start stop function for each CAN node (Channel & Global Sleep Mode) |

These items are not included. Please prepare that on your own.

| | |
|---|---|
| Buffer memory area | The size depends on your configuration. Max. is 2328Byte (582 word * 32 bit. The capacity which increases by ECC is not included.) |
| Error detection for 10.10, ISO11898<br>(10.11, ISO11898 is included) | Error detection with Buffer memory.<br>It adds a parity bit in write-data and checks it in read-data. If it detects an error, then it must send an error status to IP. |