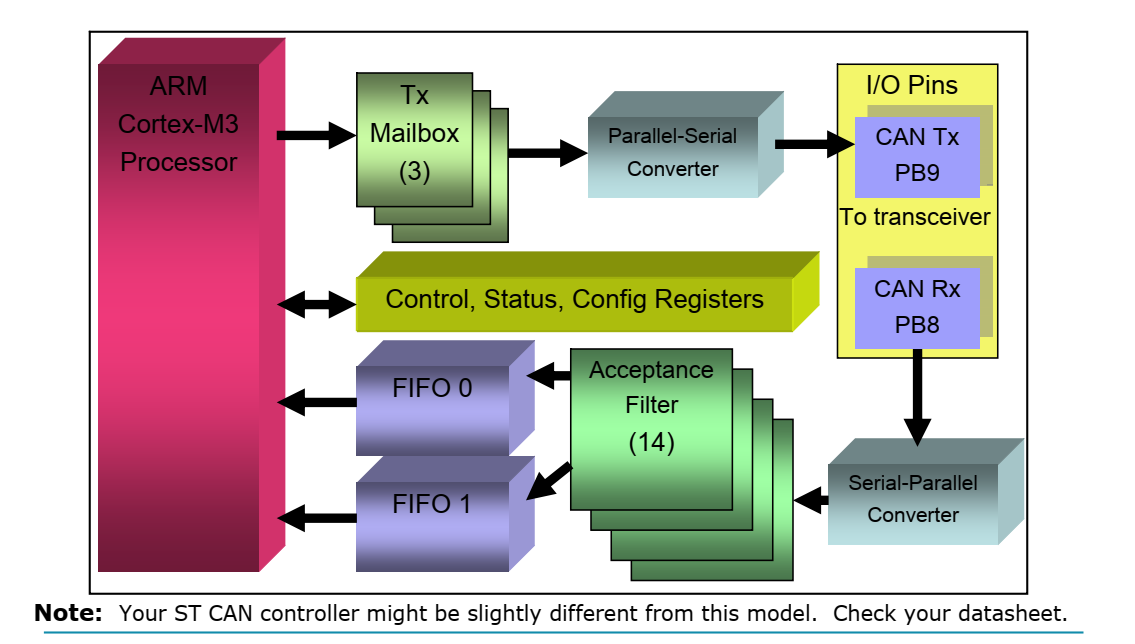
STM32 Cortex-M3 – CAN module.



1. General structure:

* 3 Tx Mailbox: this content message -> Controller will deside which message to be transmited base on ID.
* DATA-IN will pass through 14 Filter (unnecessary data(ID) will be rejected).
* If DATA-IN was accepted -> will be stored in 2 FIFO (each FIFO content 3 message).

1. Operation mode:

* Initialization mode.
* Normal mode.
* Sleep mode (after reset – default).
* Test mode:

+ Silent mode

+ Loop-back mode

* Debug mode.

1. Interrupt

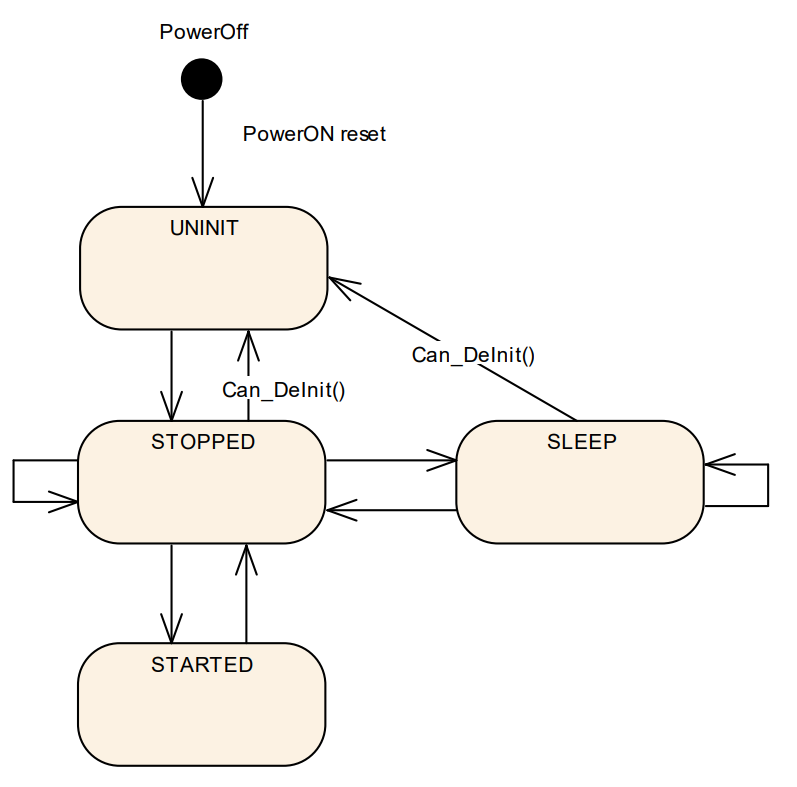
* Transmit interrupt
* FIFO 0 interrupt
* FIFO 1 interrupt
* Error and Status change interrupt

1. Buffer type

* CAN-FD (max 64 bytes). (unsupported)
* Standard CAN (max 8 bytes).
* Standard ID (11 bits ID).
* Extended ID (29 bits ID).

1. CAN controller state in AUTOSAR

* UNINIT: The CAN controller is not initialized
* STOPPED: CAN Controller is initialized but does not participate on the bus
* STARTED: The controller is in a normal operation mode with complete functionality
* SLEEP: CAN hardware that support a sleep mode.



1. Funtions causes state transition

* Can\_Init: UNINIT -> STOPPED
* Can\_SetBaudrate: STOPPED -> STOPPED; SLEEP -> SLEEP; STARTED -> STARTED
* Can\_SetControllerMode:

+ CAN\_CS\_STARTED: STOPPED -> STARTED

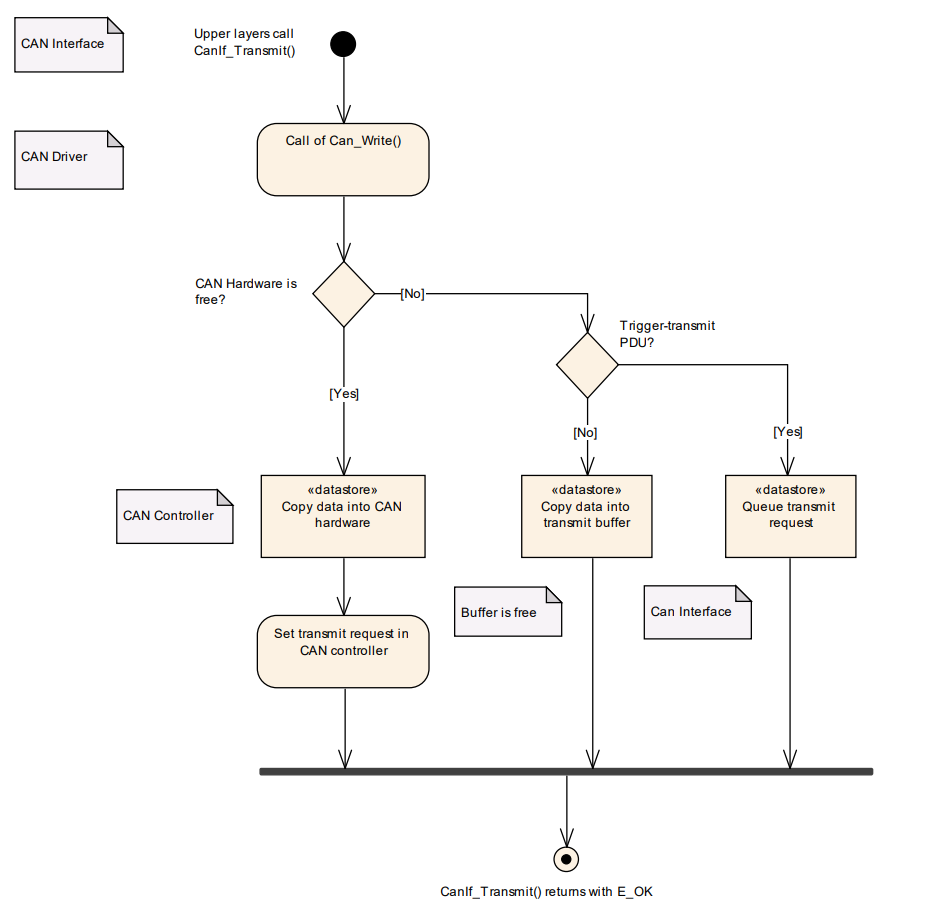
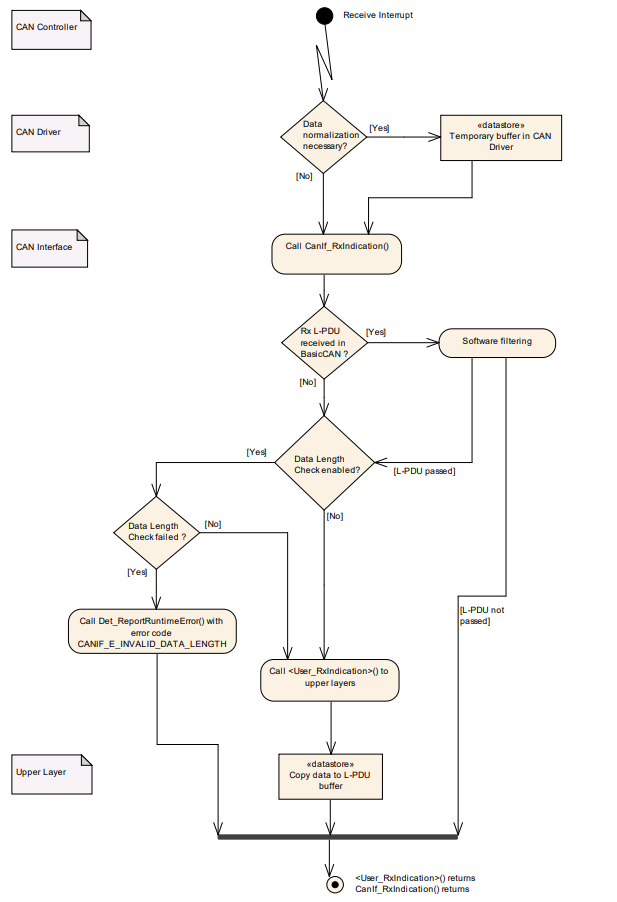
+ CAN\_CS\_STOPPED: STARTED -> STOPPED; SLEEP -> STOPPED

+ CAN\_CS\_SLEEP: STOPPED -> SLEEP

* Can\_DeInit: STOPPED -> UNINIT; SLEEP -> UNINIT (for all controllers in HW unit)

1. State transition caused by Hardware Events

* Hardware Wakeup: SLEEP -> STOPPED
* Bus-Off: STARTED -> STOPPED

1. CAN initilize flow
2. Commands flow for transmit request.
3. Commands flow for receive request