

Exercise – Communication Startup Sequence

Transmission Request

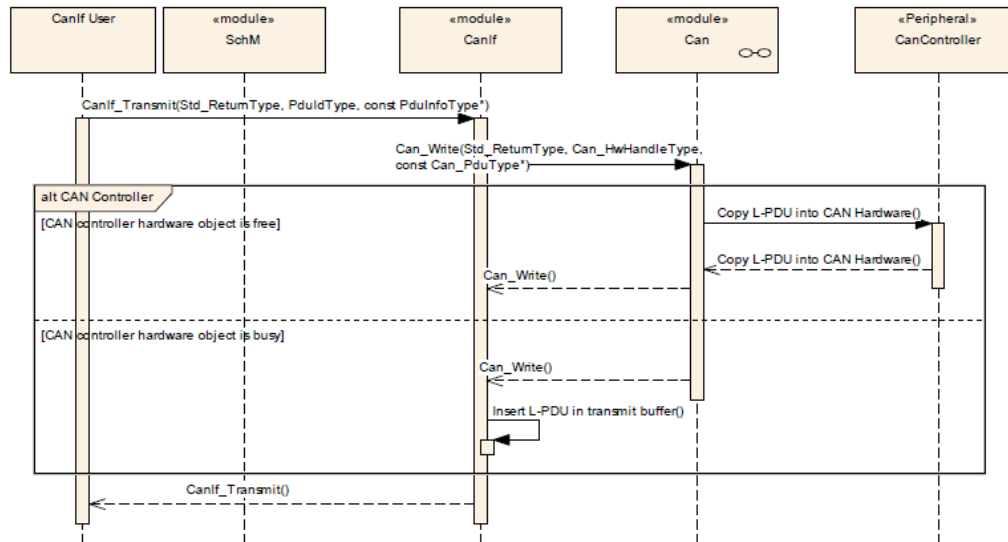


Figure 9.1: Transmission request with a single CAN Driver

Activity	Description
Transmission request	<p>The upper layer initiates a transmit request via the service <code>CanIf_Transmit()</code>. The parameter <code>CanTxPduId</code> identifies the requested L-SDU. The service performs following steps:</p> <ul style="list-style-type: none"> validation of the input parameter definition of the CAN Controller to be used <p>The second parameter <code>*PduInfoPtr</code> is a pointer on the structure with transmit L-SDU related data such as <code>SduLength</code> and <code>*SduDataPtr</code>.</p>
Start transmission	<code>CanIf_Transmit()</code> requests a transmission and calls the <code>CanDrv</code> service <code>Can_Write()</code> with corresponding processing of the HTH.
Hardware request	<code>Can_Write()</code> writes all L-PDU data in the CAN Hardware (if it is free) and sets the hardware request for transmission.
E_OK from Can_Write service	<code>Can_Write()</code> returns E_OK to <code>CanIf_Transmit()</code> .
CAN_BUSY from Can_Write service	If <code>CanDrv</code> detects, there are no free hardware objects available, it returns CAN_BUSY to <code>CanIf</code> .
Copying into the buffer	The L-PDU of the rejected transmit request will be inserted in the transmit buffer of <code>CanIf</code> until the next transmit confirmation.
E_OK from CanIf	<code>CanIf_Transmit()</code> returns E_OK to the upper layer.

Receive indication

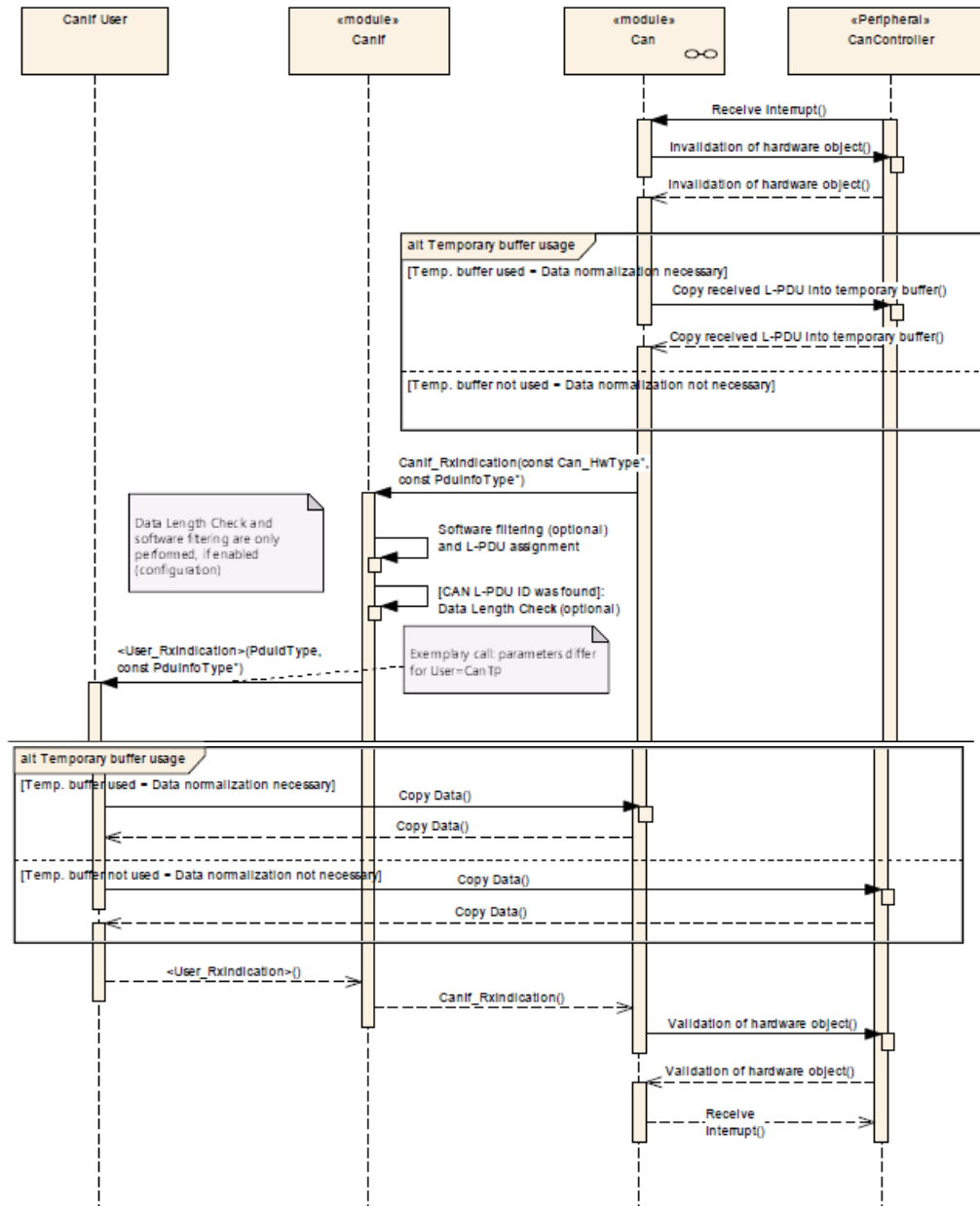


Figure 9.7: Receive indication interrupt driven

Activity	Description
Receive Interrupt	The CAN Controller indicates a successful reception and triggers a receive interrupt.
Invalidation of CAN hardware object, provide CPU access to CAN mailbox	The CPU (CanDrv) get exclusive access rights to the CAN mailbox or at least to the corresponding hardware object, where new data were received.

Buffering, normalizing	The L-PDU is normalized and is buffered in the temporary buffer located in CanDrv. Each CanDrv owns such a temporary buffer for every Physical Channel only if normalizing of the data is necessary.
Indication to CanIf	The reception is indicated to CanIf by calling of CanIf_RxIndication(). The HRH specifies the CAN RAM Hardware Object and the corresponding CAN Controller, which contains the received L-PDU. The temporary buffer is referenced to CanIf by PduInfoPtr->SduDataPtr.
Software Filtering	The Software Filtering checks, whether the received L-PDU will be processed on a local ECU. If not, the received L-PDU is not indicated to upper layers. Further processing is suppressed.
Data Length Check	If the L-PDU is found, the Data Length of the received L-PDU is compared with the expected, statically configured one for the received L-PDU.
Receive Indication to the upper layer	The corresponding receive indication service of the upper layer is called. This signals a successful reception to the target upper layer. The parameter RxPduId specifies the L-SDU, the second parameter is the reference on the temporary buffer within the L-SDU. During is execution of this service the CAN hardware buffers must be unlocked for CPU access/locked for CAN Controller access.
Validation of CAN hardware object, allow access of CAN Controller to CAN mailbox	The CAN Controller get back exclusive access rights to the CAN mailbox or at least to the corresponding hardware object, where new data were already being copied into the upper layer buffer.