

	Sleep	Receive	Transmit			Energy Detection			
	Sleep	WaitingRxFrame	RxHeader	RxFrame	TxAck	CCA	TxFrame	RxAck	ED
API: sleep()	Do nothing	Start sleep procedure in critical section: lock mutex; stop receiver (or transmitter); stop sending ACK procedure, set state Sleep; clear radio events that could have occurred during this procedure.	Not allowed due to mutex: return false.						
API: receive()	Set state WaitingRxFrame; set requested channel; TASK: DISABLE (to enable receiver via DISABLED state)	Do nothing.				Abort transmission procedure in critical section: stop receiver or transmitter (TASK: STOP); disable shorts used during transmission; clear MHRMATCH registers; set state WaitingRxFrame; enable receiver (TASK: DISABLE); clear radio events that could have happen during this procedure.			Not allowed: asserted
API: transmit()	Not allowed: asserted	Start transmission procedure in critical section: lock mutex; stop receiver (or transmitter); set transmission channel and power; stop sending ACK procedure, setting state to CCA; clear radio events that could have occurred during this procedure.	Not allowed due to mutex: return false.			Not allowed: asserted.			
API: energy_detection()	Not allowed: asserted	Start energy detection in critical section: lock mutex; stop receiver (TASK: STOP); set energy detection channel; stop sending ACK procedure; setting state to ED; clear radio events that could have occurred during this procedure.	Not allowed due to mutex: return false.			Not allowed: asserted			
EV: FRAMESTART	Should not happen (asserted)	Started receiving of a frame. Lock mutex; set state RxHeader; if length of frame is shorter of ACK: stop sending ACK procedure setting state WaitingRxFrame.	Should not happen (asserted)		Started transmitting ACK. Do nothing.	This could happen if rx frame started during transmit() critical section. Ignore received frame and do nothing.	Started transmitting frame. Do nothing.	Started receiving a frame. Do nothing.	Should not happen (asserted)

	Sleep	Receive			Transmit			Energy Detection	
	Sleep	WaitingRxFrame	RxHeader	RxFrame	TxAck	CCA	TxFrame	RxAck	ED
EV: BCMATCH	Should not happen (asserted)		<p>First match: MHR Beacon frame: mark frame as broadcast; set state RxFrame. Data or Command frame: set bcc register for second match after destination address received. Other frame type: in promiscuous mode mark as broadcast, set state RxFrame; in other mode stop sending ACK with state WaitingRxFrame.</p> <p>Second match: destination address Broadcast address: mark frame as broadcast; set state RxFrame. Matched unicast: mark frame as unicast; set state RxFrame. Missed unicast: in promiscuous mode mark frame as broadcast, set state RxFrame; in other mode stop sending ACK with state WaitingRxFrame.</p>	Should not happen because FRAMESTART→BCINIT short is enabled only in Receive state (asserted).					
EV: END	Should not happen (asserted)	It could happen if frame length is 0 (in this case FRAMESTART was not triggered) or other procedure took mutex before FRAMESTART handler took it. Check if length of frame is 0 In this case lock mutex and stop sending ACK with state WaitingRxFrame. If mutex cannot be taken or there is other frame length, do nothing: another procedure took mutex and it will prevent sending ACK.	Frame ended before MAC header received: invalid frame. Stop sending ACK procedure with state WaitingRxFrame.	Frame ended correctly. CRC invalid: stop sending ACK procedure with state WaitingRxFrame. CRC valid: set sequence number in ACK; check if received frame should be ACKed. Frame should be ACKed: set state TxAck. Frame should not be ACKed: stop sending ACK procedure with state WaitingRxFrame; notify the higher layer received()	Ended transmission of ACK. Notify the higher layer received() ; set state WaitingRxFrame.	This could happen if rx frame ended during transmit() critical section. Ignore received frame and do nothing.	Ended transmission of data (or beacon, command) frame. Disable radio shorts using during transmission procedure; check if ACK was requested. ACK requested: Set MHRMATCH unit registers; set state RxAck. ACK not requested: notify the higher layer: channel_busy() ; set state WaitingRxFrame.	Ended reception of a frame. Check if received frame was expected ACK. Received expected ACK: Notify the higher layer: transmitted() ; clear MHRMATCH registers; set state WaitingRxFrame; enable shorts used during rx; start receiver (TASK: START); unlock mutex. Received other frame: clear MHRMATCH event; start receiver (TASK: START).	Should not happen (asserted)

	Sleep	Receive			Transmit			Energy Detection	
	Sleep	WaitingRxFrame	RxHeader	RxFrame	TxAck	CCA	TxFrame	RxAck	ED
EV: DISABLED	Ended sleep procedure. Unlock mutex.	Entering Receive state. Enable receiver (TASK RXEN); unlock mutex; clear radio event DISABLED that could have occurred if stopped sending ACK.	Should not happen (asserted)		Disabled receiver due to delay sending ACK. Switch shorts used during receiving to next stage; set pending bit in ACK message.	Disabled radio to enable receiver to perform CCA procedure. Enable receiver (TASK RXEN).	Disabled receiver after CCA to enable transmitter and transmit frame. Do nothing; transmitter is enabled by shorts.	Disabled transmitter after transmitting a frame to receive ACK. Enable receiver (TASK RXEN).	Disabled radio to enable receiver to perform ED procedure. Enable receiver (TASK RXEN).
EV: READY	Should not happen (asserted)	Enabled receiver when entering Receive state. Check if higher layer should be notified about recently received frame. If there is such a frame: do not start receiver yet to prevent receiving next frame (that would overwrite recently received). If there is no such frame: enable shorts used during receive procedure and start receiver (TASK START).	Should not happen (asserted)		Transmitter is ready to send ACK and shorts started to transmit. Disable shorts used in receive state except short disabling transmitter when ACK frame ends.	Receiver is ready to perform CCA procedure. Set state TxFrame; enable shorts used in Transmit procedure; start CCA procedure (TASK CCSTART).	Transmitter is ready to transmit frame. Frame is transmitted by shorts. Do nothing.	Receiver is ready to receive ACK frame. Start receiver (TASK START)	Receiver is ready to perform ED procedure. Start ED procedure (TASK EDSTART).
EV: CCABUSY	Should not happen (asserted)						Channel is busy; frame should not be transmitted. Disable radio shorts used in Transmit state; notify the higher layer channel_busy() ; set state WaitingRxFrame; enable receiver (TASK DISABLE).	Should not happen (asserted)	
EV: EDEND	Should not happen (asserted)							Energy detection procedure ended. Notify the higher layer energy_detected() ; set state WaitingRxFrame; enable receiver (TASK DISABLE).	