UWB Command Interface

Rev. 2.3 — 14 November 2023
UM11539

User manual
RESTRICTED

Document Information

Information	Content
Keywords	UWB UCI CCC
Abstract	Defines UWB Command Interface relevant for CCC based Ranging.

Revision history

Table 1. Revision history

Revision	Date	Description	Author
2.3	23.10.2023	Status codes are updated. • STATUS_INVALID_RESPONDER_SLOT_INDEX is added to 0xA0. • STATUS_SLOT_LEN_NOT_SUPPORTED is changed from 0x1A to 0xA1. • STATUS_INVALID_SLOT_PER_RR is changed from 0x1B to 0xA2. • STATUS_INVALID_STS_IDX is changed from 0x1C to 0xA3. • STATUS_RESPONDER_LISTEN_ONLY_MODE is added to 0xA4. • STATUS_ERROR_UWB_INITIATION_TIME_TOO_OLD is added to 0x1A • STATUS_OK_NEGATIVE_DISTANCE_REPORT is added to 0x1B. • UWB Ranging Session Specific Status Codes are added from 0x20 to 0x26. Application configuration parameter has been changed. • Tag ID for ADAPTIVE_PAYLOAD_POWER is removed. • UWB_INITIATION_TIME definition has been changed from relative time to absolute time.	Boyoung Kim
2.2	13.09.2023	Length of the HOP_MODE_KEY is reduced from 16octets to 4octets.	Boyoung Kim
2.1	27.07.2023	Tag ID for RESPONDER_SLOT_INDEX is changed from 0x1E to 0xA2.	Boyoung Kim
2.0	21.02.2023	Added Session Handle SESSION_INIT_RSP. Replaced “Session ID” by “Session Handle” in all the session specific UCI command/response/notification . Remove STATUS_ERROR_SESSION_DUPLICATED reason code according to SESSION_INIT_CMD/RSP updating. AP(Application Processor) has been replaced with Host.	Boyoung Kim
1.10	29.11.2022	Default value of the CCC_CONFIG_QUIRKS from 0x0 to 0x01: Key Identifier Field formatted as: Source Short Address || Source PAN ID.	Boyoung Kim
1.9	18.11.2022	The SESSION_STATUS_NTF Reason codes has been changed due to conflicting with UCI generic specification. • URSK_EXPIRED from 0x03 to 0xA0 • TERMINATION_ON_MAX_STS from 0x04 to 0xA1	Boyoung Kim
1.8	18.2.2022	LAST_STS_INDEX_USED parameter has been added to SESSION_SET/GET_APP_CONFIG as 0xA8	Boyoung Kim
1.7	13.1.2022	RESPONDER_LISTEN_ONLY parameter has been added to SESSION_SET/GET_APP_CONFIG Add 0xD(Responder is in listen only mode) for RANGE_CCC_DATA_NTF RESPONDER_SLOT_INDEX can be updated while the session is in ACTIVE status. Update UWBD Device State Machine. Update UWB Session State Machine. Extend CORE_GET_DEVICE_INFO_RSP according to UCI Generic Specification. Improve length of SUPPORTED_PROTOCOL_VERSION and SUPPORTED_UWB_CONFIG_ID	Boyoung Kim

Table 1. Revision history*...continued*

Revision	Date	Description	Author
1.6	15.10.2021	Add GID for Proprietary Group	Boyoung Kim
1.5	14.06.2021	Add Application Parameter to support CR 9491 (Key Identifier Field byte order) as an option.	Boyoung Kim
1.4a	08.06.2021	Modify wrong reason code for MAX_RR_RETRY from SESSION_TERMINATION_ON_MAX_RR_RETRY to MAX_RANGING_ROUND_RETRY_COUNT_REACHED. Add MAX_RANGING_BLOCKS_REACHED reason code description for MAX_NUMBER_OF_MEASUREMENTS	Boyoung Kim
1.4	14.04.2021	Changed MAX_BLOCK_NUM name to MAX_NUMBER_OF_MEASUREMENTS and from ID 0xA1 to 0x32. RANGING_RESUME_CMD shall change session status from IDLE to ACTIVE	Gido van Wijk Boyoung Kim
1.3b	19.03.2021	SESSION_PRIORITY parameter for the SESSION_SET_APP_CONFIG_CMD can be changed after a session has been started.	Boyoung Kim
1.3a	13.01.2021	Updated Ranging status in RANGE_CCC_DATA_NTF according to CCC R3 0.0.9 Changed the resolution of UWB_INITIATION_TIME, SLOT_DURATION, and RANGING_INTERVAL as RSTU Increased RANGING_INTERVAL length to 4 Added STATUS_INVALID_STS_IDX status code and TERMINATION_ON_MAX_STS reason code	Anna Jankowska Boyoung Kim
1.3	23.12.2020	Added ADAPTED PAYLOAD POWER for Initiator Modified DST MAC ADDR definition Added Range CCC Data notification chapter for Controller Added GET_POSSIBLE_RAN_MULTIPLIER_VALUE_CMD	Boyoung Kim Niladri Shekhar Paria
1.2b	09.11.2020	Added URSK_TTL for session params in CCC Removed UWB ranging session specific status codes Added URSK_EXPIRED reason for session change	Gido van Wijk
1.2a	30.10.2020	Extend device capabilities for SUPPORTED_ PROTOCOL_ VERSION, SUPPORTED_UWB_ CONFIG_ID and SUPPORTED_ PULSESHAPE_ COMBO	Robert Spreitzer
1.2	12.10.2020	Release 1.2 version	Boyoung Kim
1.1e	08.10.2020	Added RANGE_RESUME_CMD as optional Updated Ranging status field of RANFE_CCC_DATA_NTF according to CCC 0.0.4a release. Change default value for PREAMBLE_ID and SFD_ID as keep value in radio configuration	Boyoung Kim Niladri Shekhar Paria
1.1d	24.09.2020	Marked UWB_INITIATION_TIME as a dynamic session parameter Modified description about state STATUS_ACTIVE Removed DEVICE_MAC_ADDRESS and DST_MAC_ADDRESS from mandatory parameter list	Niladri Shekhar Paria Boyoung Kim
1.1c	09.09.2020	Changed status field of Ranging Data Notification as CCC definition	Boyoung Kim
1.1b	25.08.2020	Removed default value of RANGING_SLOT_LENGTH Added description of Hopping Key usage	Boyoung Kim
1.1a	7.08.2020	Added CCM_TAG field to RANGE_CCC_DATA_NTF Specified Parameter error codes	Niladri Shekhar Paria Boyoung Kim

Table 1. Revision history*...continued*

Revision	Date	Description	Author
		Added FoM field to RANGE_CCC_DATA_NTF	
1.1	26.06.2020	Enabled reconfiguration of an active session Modified session status notification Added Channel_BitMask to CORE_GET_CAPS_INFO_RSP Added CCC related parameters to CORE_GET_DEVICE_INFO_RSP Modified session parameters Added new status codes	Boyoung Kim
1.0a	09.06.2020	Moved SLOTS_PER_RR to UCI Generic parameters Removed SESSION_TERMINATION_ON_MAX_RR_RETRY from SESSION_GET_STATE_RSP	Robert Spreitzer
1.0	18.05.2020	Initial document	Robert Spreitzer

1 Introduction

This document specifies a communication protocol called UWB Subsystem Command Interface (UCI) between UWB Device (UWBD) and Device Host. The packet format is derived from other standard Host controller interface specification and terminologies remains same in some references in this document.

1.1 Objectives

UCI is defined to meet the following requirements:

- Define transport layer agnostic (independent from the physical connection and any associated link protocol).
- Provide most configurable way to program the UWB Subsystem for the needs of application and platform.
- Be extensive, to allow future extensions.
- Allow vendor specific commands by use of reserved fields.

1.2 Scope

The following figure outlines a reference architecture of UWB enabled device. This document is meant for UCI as logical transport layer. This document does not cover physical interface signal handling.

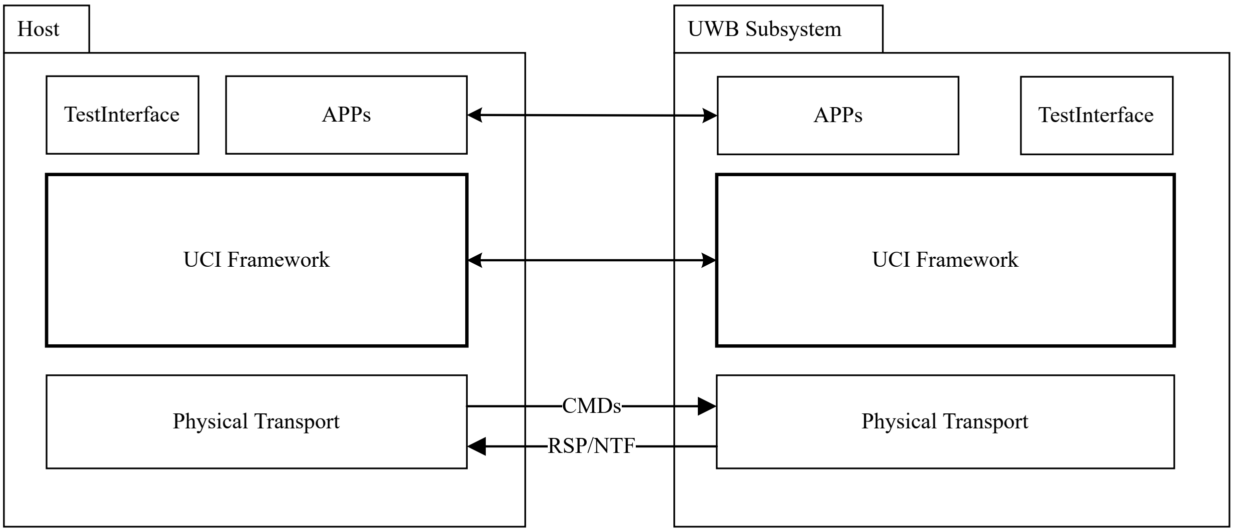

Figure 1. UCI architecture

1.3 Symbols

Across the document are used symbols defined in the table below.

Table 2. Symbols

Symbol	Description
MAX_PACKET_SIZE	Maximum UCI packet size and it is 259
MAX_PAYLOAD_SIZE	Maximum 255 octets in payload of UCI packet

1.4 Coding Conventions

Below coding conventions should be followed:

- Each octet is represented by bits b0 to b7, where b7 is the most significant bit (*msb*) and b0 is the least significant bit (*lsb*). In all representations, the leftmost bit is the *msb*.
- All values greater than 1 Octet are sent and received in Little Endian format means that least significant octet first in time.
- In representations of octet arrays, each Octet is numbered, starting at 0. Octet numbered 0 is sent over the UCI Transport first.
- This document uses the following notations for numbers:
  - Values expressed in hexadecimal form are preceded by ‘0x’,
  - Values expressed in binary form are preceded by ‘0b’.

2 UCI Core Framework

2.1 Overview

The UCI core includes the following required core functionality.

- Packet formats to transmit commands, responses, and notification messages over UCI
- Definitions of the commands, responses, and notifications are used for different operations between the host and UWBD
- A flow control mechanism for command/response message exchange
- Segmentation and reassembly for all UCI messages
- Reset, initialization, and configuration of the UWBD

2.2 UCI control messages

The UCI control messages manage and configure the UWBD device. Control messages consist of commands, responses, and notifications. Commands are only allowed to be sent in the direction from the host to UWBD, and responses and notifications are only allowed to be sent in the other direction.

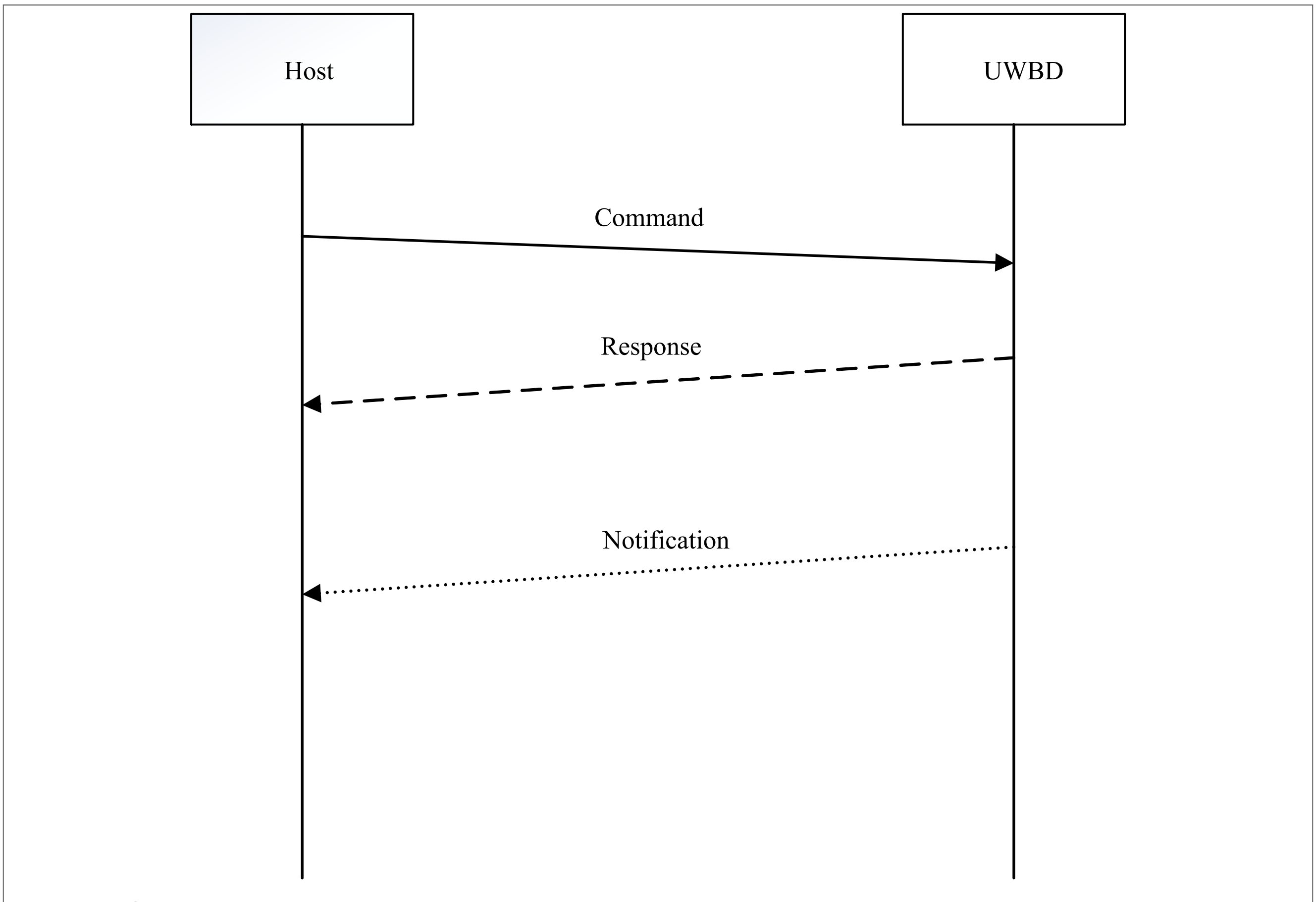

Figure 2. Control Messages exchange

A command can be sent by the host to instruct the UWBD to perform a specific action. For each command received, the UWBD shall answer with a response to acknowledge the reception of the command. The response may also indicate the changes that the command caused at UWBD.

Notifications shall only be sent from the UWBD to the host. A notification may be sent to deliver additional information related to a command. A notification may also be sent independently of any command or response, unless specified otherwise.

The payload of the control messages is sent over the UCI transport layer as a payload of control packets.

A control packet contains either a complete or a segment of a control message payload.

Both the host and UWBD shall be capable of supporting control messages with a payload up to MAX_PAYLOAD_SIZE octets, which is the maximum size of any control message payload.

However, the UWBD may specify a smaller maximum control packet payload size, as defined by the parameter MAX_PAYLOAD_SIZE.

2.2.1 Flow control for control messages

The host and UWBD can send a complete control message over the UCI in as many packets as needed. There is no packet-based flow control for control messages in UCI.

The following flow control rules apply to control messages:

- After sending a command, the host shall not send any further command until it received a response for that command, or until it has taken steps to restore the capability to exchange messages with UWBD if it determines that too much time has elapsed waiting for a response.
- After sending a command, the host shall be able to receive a response.
- After sending a response, the UWBD shall be ready to receive the next command from the host.
- The host shall be able to receive a notification from the UWBD at any time.

2.2.2 Exception handling for control messages

The rules in this section define the exception processing to be performed by a receiver for an erroneous control message.

Any command received by the host shall be ignored.

A control message that is consistent with this specification, apart from the presence of additional bytes at the end, shall not be treated as syntax error, but the additional bytes shall be ignored.

In all other cases of control message with syntax errors, meaning that the coding of the control packet is not consistent with this specification and where the receiver can still determine the type of control message:

- If the control message is a command, the UWBD shall ignore the content of the command and send a response with the same GID and OID field values as in the command and with a status value STATUS_SYNTAX_ERROR. The response shall not contain any additional fields.
- If the control message is a response, the host shall ignore the content of the response and is free to send another command.

The UWBD shall respond to an unknown command (unknown GID or OID) with a response having the same GID and OID field values as the command, followed by a status field with the value of STATUS_UNKNOWN_GID/ STATUS_UNKNOWN_OID respectively and no additional fields.

The host shall ignore any unknown response or notification (unknown GID or OID).

2.3 Packet formats

2.3.1 Common packet header

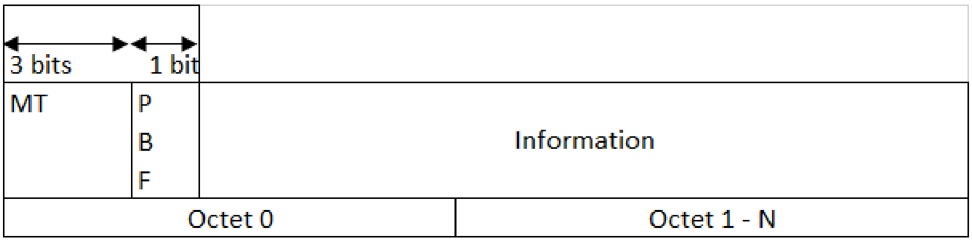

Figure 3. UCI common packet header

All packets have common header, consisting of a Message Type (MT) field and a Packet Boundary Flag (PBF) field as shown in Figure 3

Message Type (MT)

The MT field indicates the contents of the packet and shall be a 3-bit field containing one of the values listed in Table 3. The content of the Information field is dependent on the value of the MT field. The receiver of an MT designated as RFU shall silently discard the packet.

Table 3. Message Types

Message type	Description
000b	RFU
001b	Control packet – Command message as payload
010b	Control packet – Response message as payload
011b	Control packet – Notification message as payload
100b – 111b	RFU

Packet Boundary Flag (PBF)

The Packet Boundary Flag (PBF) is used for segmentation and reassembly and shall be a 1-bit field containing one of the values listed in Table 4.

Table 4. Packet Boundary Flag

PBF	Description
0b0	The packet contains a complete message, or the packet contains the last segment of the segmented message
0b1	The packet contains a segment of a message that is not the last segment

The following rules apply to the PBF flag in packets:

- If the packet contains a complete message, the PBF shall be set to 0b0.
- If the packet contains the last segment of a segmented message, the PBF shall be set to 0b0.
- If the packet does not contain the last segment of a segmented message, the PBF shall be set to 0b1.

2.3.2 Format of control packets

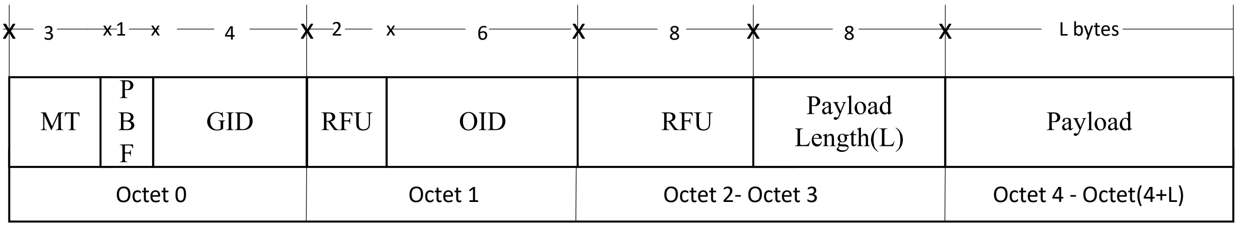

Figure 4. UCI control packets format

Each control packet shall have a 4-octet packet header and may have additional payload for carrying a control message payload or a segment of a control message payload.

NOTE: In the case of an ‘empty’ control message, only the packet header is sent.

Message Type (MT)

Refer to Table 3 for details of the MT field.

Packet Boundary Flag (PBF)

Refer to Table 4 for details of the PBF field.

Group Identifier (GID)

The UCI supports commands, responses are categorized according their individual groups. The Group Identifier (GID) indicates the categorization of the message and shall be a 4-bit field containing one of the values listed in Table 52.

Opcode Identifier (OID)

The Opcode Identifier (OID) indicates the identification of the control message and shall be a 6-bit field that is a unique identification within the group. OID values are defined along with the definition of the respective control messages described in Table 52.

Payload Length (L)

The Payload Length shall indicate the number of octets present in the payload. The Payload Length field shall be a 8-bit field containing a value from 0 to MAX_PAYLOAD_SIZE

Table 5. Symbols

Symbol	Description
MAX_PACKET_SIZE	Maximum UCI packet size
MAX_PAYLOAD_SIZE	Maximum 255 octets in payload of UCI packet

In this specification, Tag-Length-Value (TLV) format is used to configure or get parameters.

Table 6. TLV format

Payload	Field	Length	Description
m+2 Octets	Tag	1 Octets	Parameter ID shall be used as Tag.
	Length	1 Octet	The length of Value (m).
	Value	m Octets	Value of the configuration parameter.

3 UCI core control messages

Following are the descriptions of the commands and responses are part of the UCI core. All the core control messages are accepted only in STATUS_READY.

UWBD shall reject all the core control messages when UWBD in STATUS_ACTIVE state.

3.1 Initialization of UWBD

Initialization procedure is vendor specific and the host shall be notified by DEVICE_STATUS_NTF notification with STATUS_READY state once UWBD powered up and initialized successfully. The host shall not send any commands until the host receives the DEVICE_STATUS_NTF notification with STATUS_READY state.

The host shall send CORE_GET_DEVICE_INFO_CMD to retrieve the device specific information and version of followed specifications.

UWBD shall respond by CORE_GET_DEVICE_INFO_RSP with all device information as shown below. Additional information shall start with CCC related parameters, defined in Table 48, given in TLV form. All other manufacturer specific information (encoded in manufacturer specific mode) shall follow CCC related parameters. Additional information length is a length of both - CCC and manufacturer specific information.

Table 7. CORE_GET_DEVICE_INFO_CMD

Payload field(s)	Length	Value/ Description
Command	0 Octets	Get the device info data from UWBD

Table 8. CORE_GET_DEVICE_INFO_RSP

Payload field(s)	Length	Value/Description
Status	1 Octet	For various status values refer to Table 53. The host shall ignore other parameters if status is not STATUS_OK
UCI Generic version	2 Octets	UCI Generic Specification version. Octet [0]: Major number Octet [1]: Minor number *Example: Major number 1 and Minor number 0 is reported as 0x01 0x00 at UCI transport*
FiRa MAC Version	2 Octets	FiRa MAC specification version. No relavance for CCC Octet [0]: Major number Octet [1]: Minor number *Example: Major number 1 and Minor number 0 is reported as 0x01 0x00 at UCI transport*
FiRa PHY Version	2 Octets	FiRa PHY specification version. No relavance for CCC Octet [0]: Major number Octet [1]: Minor number *Example: Major number 1 and Minor number 0 is reported as 0x01 0x00 at UCI transport*
FiRa UCI Test Version	2 Octets	UCI Generic Specification version. No relavance for CCC Octet [0]: Major number Octet [1]: Minor number *Example: Major number 1 and Minor number 0 is reported as 0x01 0x00 at UCI transport*

Table 8. CORE_GET_DEVICE_INFO_RSP*...continued*

Payload field(s)	Length	Value/Description
Vendor Specific Information Length	1 Octet	The length of the following vendor specific information.
Vendor Specific Information	n Octets	CCC related information (Table 48) in TLV form and UWBD manufacturer specific information like chip version, chip variant encoded in a manufacturer-specific mode.

3.2 UWBD capability information

The host shall use CORE_GET_CAPS_INFO_CMD command to get the capability of the UWBD. The capability information is vendor-specific and the host shall use capability information to communicate with UWBD. Table 49 contains a list of device capability parameters.

Table 9. CORE_GET_CAPS_INFO_CMD

Payload field(s)	Length	Value/ Description
Command	0 Octets	Get the capability data from UWBD

Table 10. CORE_GET_CAPS_INFO_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	For various status values, refer to Table 53.
Number of parameters	1 Octet	Number of TLV structures to follow
Capability data	n Octets	Capability data in TLV form. Type (T) = 1 octet Length (L) = 1 octet (m) Value (V) = parameter value written in m octets

3.3 UWBD Configuration

3.3.1 Setting device configurations

These control messages are used to set the configuration parameters on the UWBD.

Table 11. CORE_SET_CONFIG_CMD

Payload field(s)	Length		Value/ Description	
Number of Parameters	1 Octet		The number of Parameter fields to follow (n).	
Parameter [1..n]	m+2 Octets	ID	1 Octet	The identifier of the configuration parameter. See Table 50 for list of IDs.
		Len	1 Octet	The length of Val (m). If Len is equal to 0x00, then the Val field is omitted, and the UWBD shall set the configuration parameter to its default value.
		Val	m Octets	The value of the configuration parameter.

Table 12. CORE_SET_CONFIG_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	For various status values refer to Table 53.
Number of parameters	1 Octet	The number of Parameter ID fields to follow (n). The value is 0x00 and no Parameter ID is listed if Status is STATUS_OK.
Parameter ID [0..n]	1 Octet	The identifier of the invalid configuration parameter. See Table 50 for list of IDs.
Parameter status	1 Octet	STATUS_SYNTAX_ERROR / STATUS_INVALID_PARAM / STATUS_INVALID_RANGE

All device configuration parameters within the UWBD are set to default values at Table 50, but the host may use CORE_SET_CONFIG_CMD to change these values. The UWBD responds with CORE_SET_CONFIG_RSP, and the Status indicates whether the setting of these configuration parameters was successful or not. A status of STATUS_OK shall indicate that all configuration parameters have been set to the requested values within the UWBD. Any other status code means, that no parameter has been changed.

A control message that is not consistent with specification, apart from the presence of additional bytes at the end, shall be treated as syntax error, but the additional bytes shall be ignored. Examples: Message type is not correct in packet header, number of parameters is not matching the number of TLVs in the payload. In such cases UWBD shall respond with CORE_SET_CONFIG_RSP with status field STATUS_SYNTAX_ERROR.

If the host tries to set invalid parameters (not available on the UWBD, length not as per specification or invalid value), UWBD shall respond with CORE_SET_CONFIG_RSP with status field STATUS_FAILED and including all the invalid parameter ID(s) with an appropriate parameter status code, as defined below:

- STATUS_SYNTAX_ERROR - the parameter length is not as per specification,
- STATUS_INVALID_PARAM - the parameter ID is not available on the UWBD,
- STATUS_INVALID_RANGE - the parameter value is invalid.

If the overall command length is more than MAX_PACKET_SIZE, then the host shall split the command to ensure that TLV is sent at the beginning of the next fragmented packet with PBF is set to 1.

3.3.2 Retrieve device configurations

These control messages are used by the host to retrieve current configuration parameters of the UWBD.

Table 13. CORE_GET_CONFIG_CMD

Payload field(s)	Length	Value/ Description
Number of Parameters	1 Octet	The number of parameter ID fields to follow (n).
Parameter ID [1..n]	1 Octet	The identifier of the configuration parameter. See Table 50 for a list of IDs.

Table 14. CORE_GET_CONFIG_RSP

Payload field(s)	Length	Value/ Description		
Status	1 Octet	For various status values refer to Table 53.		
Number of parameters	1 Octet	The number of TLV(n) follow.		
Parameter [1..n]	m+2 Octets	Type	1 Octet	The identifier of the configuration parameter.

Table 14. CORE_GET_CONFIG_RSP*...continued*

Payload field(s)	Length	Value/ Description		
				See Table 50 for a list of IDs.
		Length	1 Octet	The length of value (m).
		Value	m Octets	The value of the configuration parameter.

The host may use CORE_GET_CONFIG_CMD to retrieve the current configuration parameters of the UWBD. If the UWBD able to respond with all requested parameters, the UWBD shall respond with the CORE_GET_CONFIG_RSP with a status of STATUS_OK.

If the host tries to retrieve any parameter(s) that are not available in the UWBD, the UWBD shall respond with a CORE_GET_CONFIG_RSP with a status field of STATUS_INVALID_PARAM, containing each unavailable parameter ID with a parameter length field of value zero. In this case, the CORE_GET_CONFIG_RSP shall not include any parameter(s) that are available on the UWBD.

After receiving the list of unavailable parameters, the host can assume that the other parameters requested in the CORE_GET_CONFIG_CMD are available, and the host may initiate another CORE_GET_CONFIG_CMD to retrieve those parameters.

3.3.3 Device configuration parameters

These are the device specific configuration parameters related to UWBD and these parameters shall be set by the host before creating first session on the device. Table 50 contains a list of all parameters and all parameters have default values. The CORE_SET_CONFIG_CMD/ CORE_GET_CONFIG_CMD commands are used to set and get the device configuration parameters.

3.4 Generic error

This notification is used to inform the host about a generic error situation.

This notification is used in error situations when the error cannot be notified using an error status in a response. This notification shall not be used to report error scenarios related to UWB session.

To notify a generic error situation, the UWBD shall send CORE_GENERIC_ERROR_NTF to the host with the status code identifying the error case.

The host shall retransmit the same command on receipt of generic notification with STATUS_COMMAND_RETRY status.

Table 15. CORE_GENERIC_ERROR_NTF

Payload field(s)	Length	Value/Description
Status	1 Octet	See Table 53

4 UWBD state machine

The host shall be aware of UWBD states. Each transition from one state to another is accompanied by an appropriate UCI command, response, or notification, so both the host and UWBD can always unambiguously know the current UWBD device state. The device state moves from STATUS_ACTIVE to STATUS_READY if there are no active sessions. RANGE_STOP_CMD should stop notification on a specific session and host shall keep receiving the ranging data notifications from other active sessions. UWBD moves from STATUS_READY to STATUS_ACTIVE on activation of at least one ranging session.

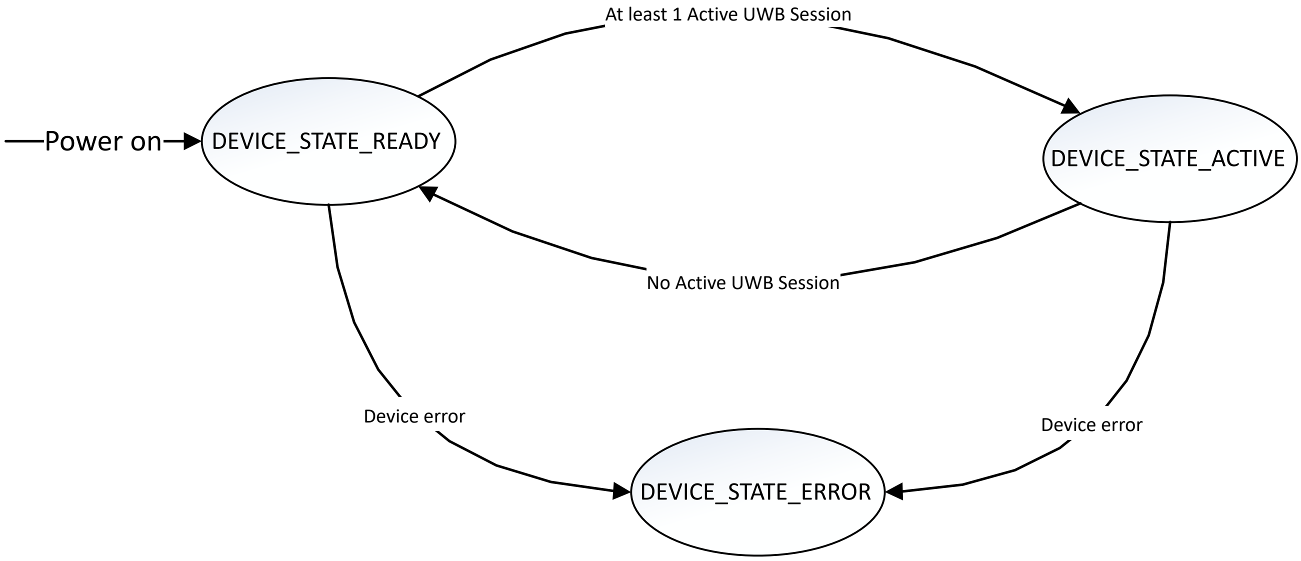

Figure 5. UWBD state transition

4.1 State DEVICE_STATE_READY

This state is the first state of UWBD after power-on or on receipt of DEVICE_RESET_CMD from the host. In this state UWBD can accept all the UCI commands. The UWBD can move to STATUS_ACTIVE state as soon as at least one UWB session is moved to active state i.e. ranging is started. In STATUS_READY state UWBD may enter into low-power mode and it is implementation specific to enter UWBD into low-power mode on its own when LOW_POWER_MODE config is enabled. In STATUS_READY state, the host can query for number of UWB sessions created in UWBD using SESSION_GET_COUNT_CMD and UWBD responds by SESSION_GET_COUNT_RSP with UWB sessions count. It is recommended for UWBD is set to DEVICE_STATE_READY state after all the sessions are moved from active to idle state.

4.2 State DEVICE_STATE_ACTIVE

The UWBD can move from STATUS_READY to STATUS_ACTIVE state if there is at least one successful activation of UWB session. In this state, the host can query for number of UWB sessions created in UWBD using SESSION_GET_COUNT_CMD and UWBD responds by SESSION_GET_COUNT_RSP with UWB sessions count. In this state UWBD can accept an UWB session configuration commands. If there are no active UWB sessions, meaning that all UWB sessions are idle then UWBD shall move to DEVICE_STATE_READY state. It is recommended that UWBD is set to DEVICE_STATE_ACTIVE state when the first session is activated before sending session status notification.

4.3 State DEVICE_STATE_ERROR

The UWBD can move to the DEVICE_STATE_ERROR device state if there are any unrecoverable error is encountered. The Host shall issue DEVICE_RESET_CMD command to reset the UWBD or host shall perform power cycling of UWBD if DEVICE_RESET_CMD command is not responded by UWBD

4.4 Device status notification

UWBD notifies the host whenever there is an update in UWBD state. UWBD shall sends notification to the host and indicates the current device state via DEVICE_STATUS_NTF.

Various state values are given in the table below.

Table 16. DEVICE_STATUS_NTF

Payload field(s)	Length	Value/ Description
Status	1 Octet	UWBD device status. Possible values listed in the Table 17

Table 17. Device Status Values

Status	Value/ Description
0x00	RFU
0x01	STATUS_READY - UWBD is initialized and ready UWB session
0x02	STATUS_ACTIVE - UWBD is busy with UWB session
0xFF	STATUS_ERROR – Error occurred within UWBD

4.5 Reset of UWBD

These control messages are used to reset the UWBD. The DEVICE_RESET_CMD is issued by the host to reset the UWBD where all the sessions, internal states are re-initialized.

This command may be issued anytime following power-up of the UWBD. If the host sends DEVICE_RESET_CMD, it shall ignore all messages except DEVICE_RESET_RSP. Once the host has received DEVICE_RESET_RSP, it shall not send any other command till it receives DEVICE_STATUS_NTF with STATUS_READY.

DEVICE_RESET_CMD command with reset config “UWBD reset” indicates that all the ongoing sessions are invalidated followed by device specific reset operation.

On receipt of DEVICE_RESET_CMD, the UWBD SHALL respond with DEVICE_RESET_RSP with status set to STATUS_OK and begin its reset procedure. On completion of the reset procedure, the UWBD SHALL send DEVICE_STATUS_NTF notification with STATUS_READY state toAP indicating that the UWBD has been reset.

Table 18. DEVICE_RESET_CMD

Payload field(s)	Length	Value/ Description
Reset configuration	1 Octet	0x00: UWBD reset 0x01-0xFF: RFU

Table 19. DEVICE_RESET_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	STATUS_OK

Table 19. DEVICE_RESET_RSP

Payload field(s)	Length	Value/ Description
		For various status values refer to Table 53.

5 UWB session management

The host shall maintain the UWB session states. Transition from one state to another is accompanied by an appropriate UCI command, response, or notification.

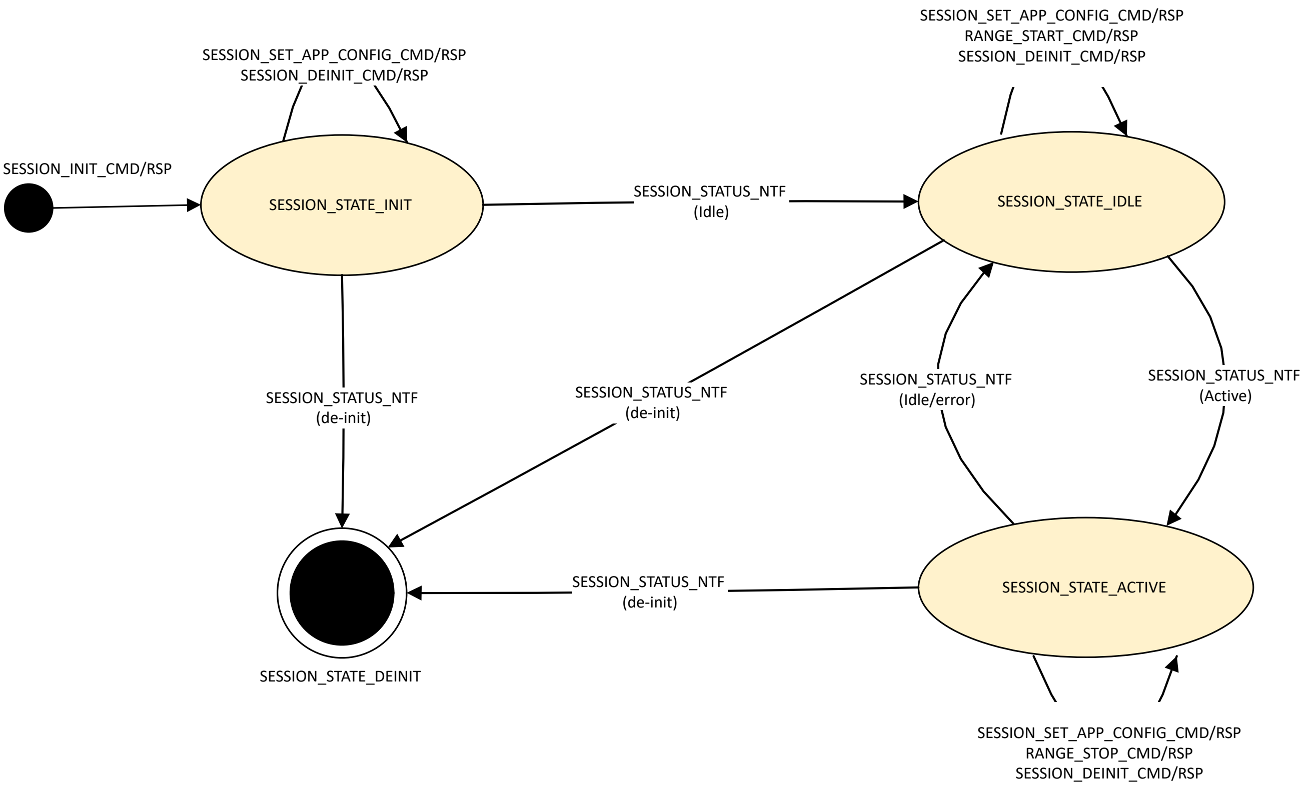

Figure 6. Session state transition - host view

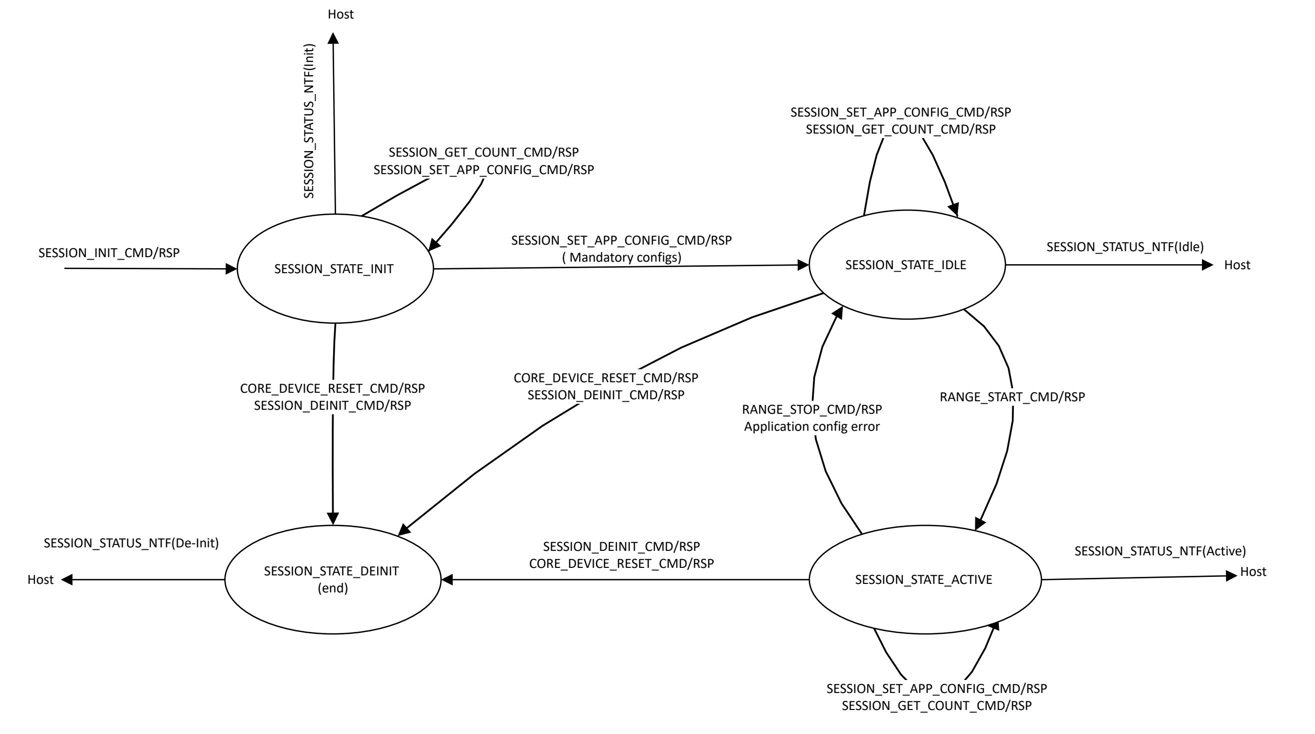
Host
SESSION_STATUS_NTF(Init)
SESSION_GET_COUNT_CMD/RSP
SESSION_SET_APP_CONFIG_CMD/RSP
SESSION_SET_APP_CONFIG_CMD/RSP
SESSION_GET_COUNT_CMD/RSP
SESSION_INIT_CMD/RSP
SESSION_STATE_INIT
SESSION_SET_APP_CONFIG_CMD/RSP
(Mandatory configs)
SESSION_STATE_IDLE
SESSION_STATUS_NTF(Idle)
Host
CORE_DEVICE_RESET_CMD/RSP
SESSION_DEINIT_CMD/RSP
CORE_DEVICE_RESET_CMD/RSP
SESSION_DEINIT_CMD/RSP
RANGE_STOP_CMD/RSP
Application config error
RANGE_START_CMD/RSP
SESSION_STATUS_NTF(De-Init)
Host
SESSION_STATE_DEINIT
(end)
SESSION_DEINIT_CMD/RSP
CORE_DEVICE_RESET_CMD/RSP
SESSION_STATE_ACTIVE
SESSION_STATUS_NTF(Active)
Host
SESSION_SET_APP_CONFIG_CMD/RSP
SESSION_GET_COUNT_CMD/RSP

Figure 7. Session state transition - UWBD view

Note that when the UWB session is de-initialized meaning that the UWB session is removed in the UWBD and corresponding data structure is destroyed.

5.1 UWB session initiation

The host shall establish session by sending the SESSION_INIT_CMD command to create the new UWB session with parameter session ID and type of the session. The UWBD shall respond SESSION_INIT_RSP with status set to STATUS_OK and the session handle field if the command is successfully accepted by the UWBD, otherwise the UWBD shall send the SESSION_INIT_RSP with proper status code which indicates a failure. The UWBD shall notify by sending SESSION_STATUS_NTF with SESSION_STATE_INIT session state as shown in SESSION_STATUS_NTF notification on reception of SESSION_INIT_RSP with STATUS_OK and a session handle. The UWBD and host shall use the session handle as a reference for a UWB session and use the Session handle in any session specific UCI Messages during communication. The Host shall receive the SESSION_STATUS_NTF notification on every UWB session state change. The host shall wait for SESSION_STATUS_NTF notification with session state of SESSION_STATE_INIT before applying application configurations for a UWB session.

The UWBD shall respond with STATUS_ERROR_SESSION_NOT_EXIST error status when the host tries to Configure/Start/Stop the UWB session without initializing the UWB session.

Table 20. SESSION_INIT_CMD

Payload field(s)	Length	Value/Description
Session ID	4 Octets	A unique random number generated by the application. ***Note:*** *Session ID = 0x0000000: Reserved for test mode session*
Session type	1 Octet	Type of session 0x00: Ranging session (not supported) 0x01-0x9F: RFU 0xA0: CCC Ranging session

Table 20. SESSION_INIT_CMD*...continued*

Payload field(s)	Length	Value/Description
		0xA1-0xCF: RFU 0xD0: Device test mode 0xD1-0xDF: RFU 0xE0-0xFF: Proprietary extension

Table 21. SESSION_INIT_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	Status code as per Table 53.
Session handle	4 Octet	Session handle is a 32bit unique number generated by the UWBD and it shall be used to uniquely identify the UWB Session. The host and the UWBD shall use a session handle as a reference for a UWB session in any session specific command/response/notification.

The host shall use SESSION_DEINIT_CMD command to de-initialize the session and the UWBD shall respond by SESSION_DEINIT_RSP with a Status indicating that command is accepted or not. The UWBD shall cleanup the UWB session data associated with Session Handle on reception of de-Init command and the UWBD shall notify SESSION_STATUS_NTF with a Session State of SESSION_STATE_DEINIT to indicate that the session is de-initialized.

Table 22. SESSION_DEINIT_CMD

Payload field(s)	Length	Value/Description
Session handle	4 Octets	Session handle of session to be de-initialized.

Table 23. SESSION_DEINIT_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	For various status values refer to Table 53.

Table 24. SESSION_STATUS_NTF

Payload field(s)	Length	Value/ Description
Session handle	4 Octets	Session handle of UWB session this notification belongs to.
Session status	1 Octet	0x00: SESSION_STATE_INIT 0x01: SESSION_STATE_DEINIT 0x02: SESSION_STATE_ACTIVE 0x03: SESSION_STATE_IDLE 0x04-0xFE: RFU 0xFF: SESSION_ERROR
Reason code	1 Octet	Refer to Table 25 for reason codes.

Table 25. Reason codes for session status change

Code	Description
Reason codes defined in the Generic UCI Specification	
0x00	STATE_CHANGE_WITH_SESSION_MANAGEMENT_COMMANDS
0x01	MAX_RANGING_ROUND_RETRY_COUNT_REACHED
0x02	MAX_RANGING_BLOCKS_REACHED
0x03 - 0x7F	Reserved for future use
Reserved for proprietary use	
0x80-0x9F	Proprietary codes
Reason codes valid only for the CCC session	
0xA0	URSK_EXPIRED
0xA1	TERMINATION_ON_MAX_STS
0xA2 - 0xDF	Reserved for future use
Reserved for proprietary use	
0xE0-0xFF	Proprietary codes

5.2 UWB session configurations

The host shall configure the session when the UWB session is in the SESSION_STATE_INIT, SESSION_STATE_IDLE or SESSION_STATE_ACTIVE session states. The Section 8.4 defines list of application configurations with default values and some of the configurations are mandatory to be set by the Host. When UWB session is in SESSION_STATE_INIT, the UWBD shall notify with SESSION_STATUS_NTF setting Session State to SESSION_STATE_IDLE after all mandatory application configurations are set by host, and this ensures that UWB session is ready for ranging.

The host may retrieve current application configurations for given UWB session at any session states.

The host may establish new UWB sessions and configure UWB sessions during the DEVICE_STATE_READY/ DEVICE_STATE_ACTIVE device state of the UWBD.

The UWBD responds with SESSION_SET_APP_CONFIG_RSP, and the Status indicates whether the setting of these configuration parameters was successful or not. A Status of STATUS_OK shall indicate that all configuration parameters have been set to the requested values within the UWBD. Any other Status code means, that no Parameter has been changed.

A control message that is not consistent with specification, apart from the presence of additional bytes at the end, shall be treated as syntax error, but the additional bytes shall be ignored. Examples: Message type is not correct in packet Header, number of parameters is not matching the number of TLVs in the payload. In such cases UWBD shall respond with SESSION_SET_APP_CONFIG_RSP with status field STATUS_SYNTAX_ERROR.

If the host tries to set invalid parameters (not available on the UWBD, length not as per specification or invalid value), UWBD shall respond with SESSION_SET_APP_CONFIG_RSP with status field STATUS_FAILED and including all the invalid parameter ID(s) with an appropriate parameter Status code, as defined below:

- STATUS_SYNTAX_ERROR - the parameter length is not as per specification,
- STATUS_INVALID_PARAM - the parameter ID is not available on the UWBD,
- STATUS_INVALID_RANGE - the parameter value is invalid.

If the overall command length is more than MAX_PACKET_SIZE, then the host shall split the command to ensure that TLV is sent at the beginning of the next fragmented packet with PBF is set to 1.

5.3 Setting application configurations

The host shall use the SESSION_SET_APP_CONFIG_CMD command to set APP configuration parameters for the requested UWB session. The UWBD shall respond with the SESSION_SET_APP_CONFIG_RSP with status indicating that APP configuration parameters are applied or not. Section 8.4 contains a list of APP configuration parameters. The commands used to set these parameters are specified in Table 26.

The host can set some of the APP configuration parameters while in the SESSION_STATE_ACTIVE state as well. The UWBD shall respond with a SESSION_SET_APP_CONFIG_RSP setting status to STATUS_ERROR_SESSION_ACTIVE if the Host tries to apply configurations which are not allowed to set during the SESSION_STATE_ACTIVE session state. Host shall use RANGE_STOP_CMD to stop the active UWB session before applying the new set of APP configuration parameters which are not permitted to change during SESSION_STATE_ACTIVE.

The host must wait for SESSION_STATUS_NTF notification with SESSION_STATE_INIT status before applying any APP configurations.

The host can apply some of the APP configuration in SESSION_STATE_ACTIVE as well. Parameters, which can be reconfigured when a session is active, are indicated by (*) in Table 51. When the host try to apply configurations, which are not supposed to change in ongoing session, UWBD responds by SESSION_SET_APP_CONFIG_RSP listing invalid parameters with status STATUS_ERROR_SESSION_ACTIVE.

Table 26. SESSION_SET_APP_CONFIG_CMD

<table>
<tr><th>Payload field(s)</th><th>Length</th><th colspan="3">Value/Description</th></tr>
<tr><td>Session handle</td><td>4 Octets</td><td colspan="3">All APP configurations belonging to this session handle.</td></tr>
<tr><td>Number of APP configurations</td><td>1 Octet</td><td colspan="3">The number of APP configuration fields to follow (n).</td></tr>
<tr><td rowspan="3">APP configuration [1..n]</td><td rowspan="3">(2+m) *n Octets</td><td>APP config ID</td><td>1 Octet</td><td>See Table 51</td></tr>
<tr><td>Config length</td><td>1 Octet</td><td>See Table 51</td></tr>
<tr><td>Config values</td><td>m Octets</td><td>See Table 51</td></tr>
</table>

Table 27. SESSION_SET_APP_CONFIG_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	For various status values, refer to Table 53.
Number of parameters	1 Octet	The number of parameter ID fields to follow (n) The value is 0x00 and no parameter ID is listed if status is STATUS_OK.
Parameter ID [0..n]	1 Octet	The identifier of the invalid parameter
Parameter status		STATUS_SYNTAX_ERROR / STATUS_INVALID_PARAM / STATUS_INVALID_RANGE / STATUS_ERROR_SESSION_ACTIVE

5.4 Retrieve application configurations

The host may use SESSION_GET_APP_CONFIG_CMD to retrieve the current APP configuration parameters of the requested UWB session.

If the host tries to retrieve any APP configuration(s) that are not available in the session configuration, the UWBD shall respond with a SESSION_GET_APP_CONFIG_RSP with a Status field of STATUS_INVALID_PARAM, containing each unavailable APP config ID with a config length field of value zero. In this case, the SESSION_GET_APP_CONFIG_RSP shall not include any configuration(s) that are available on the session configuration.

After receiving the list of unavailable configurations, the host can assume that the other APP configurations requested in the SESSION_GET_APP_CONFIG_RSP are available, and the host may initiate another SESSION_GET_APP_CONFIG_RSP to retrieve those configurations.

Table 28. SESSION_GET_APP_CONFIG_CMD

Payload field(s)	Length	Value/Description	
Session handle	4 Octets	APP configurations belonging to this session handle	
Number of APP configurations	1 Octet	The number of APP configuration fields to follow (n) If value is set to 0, then UWBD should respond with all the APP configurations IDs	
APP configuration [1..n]	n Octets	APP config IDs	See Table 51

Table 29. SESSION_GET_APP_CONFIG_RSP

Payload field(s)	Length	Value/Description		
Status	1 Octet	For various status values, refer to Table 53.		
Number of APP Configurations	1 Octet	The number of APP Configuration fields to follow (n) If value is set to 0, then payload contains all the APP configurations IDs		
APP Configuration [1..n]	(2+m) *n Octets	APP config ID	1 Octet	See Table 51
		Config Length	1 Octet	See Table 51
		Config Values	m Octets	See Table 51

5.5 UWB session state query

The host shall use SESSION_GET_STATE_CMD to query the current state of a UWB session. UWBD shall respond SESSION_GET_STATE_RSP response with the current state of the requested UWB session.

The host shall ignore the session state if SESSION_GET_STATE_RSP has status other than STATUS_OK.

Table 30. SESSION_GET_STATE_CMD

Payload field(s)	Length	Value/Description
Session handle	4 Octets	Session handle of the UWB session, for which current state shall be queried

Table 31. SESSION_GET_STATE_RSP

Payload field(s)	Length	Value/Description
Status	1 Octet	For various status values, refer to Table 53.

Table 31. SESSION_GET_STATE_RSP*...continued*

Payload field(s)	Length	Value/Description
		Session state is valid only if status is STATUS_OK.
Session state	1 Octet	0x00: SESSION_STATE_INIT 0x01: SESSION_STATE_DEINIT 0x02: SESSION_STATE_ACTIVE 0x03: SESSION_STATE_IDLE 0x04-0xFF: RFU

5.6 Get number of UWB sessions

The host may use SESSION_GET_COUNT_CMD to retrieve the number of UWB sessions in UWBD.

The UWBD responds by SESSION_GET_COUNT_RSP with current number of UWB sessions.

Table 32. SESSION_GET_COUNT_CMD

Payload field(s)	Length	Value/ Description
Command	0 Octet	Retrieve the current number of UWB sessions

Table 33. SESSION_GET_COUNT_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	For various status values, refer to Table 53.
Session count	1 Octet	Current UWB session count if status is STATUS_OK

5.7 UWB session load query

The host shall use SESSION_GET_POSSIBLE_RAN_MULTIPLIER_VALUE_CMD to check possible RAN multiplier value for a new session. UWBD responds possible RAN multiplier value according to below calculation with SESSION_GET_POSSIBLE_RAN_MULTIPLIER_VALUE_RSP. Runtime for active session shall be calculated with actual slot numbers and additional time which must prepare to range rounds.

- Load = (Total runtime of active sessions) / (Average ranging interval)
- – Total runtime = Sum of runtime of active session
  – Average ranging interval = (Sum of ranging interval of active session) / (Number of active sessions)
- RAN multiplier value = (Load / 25) + 1
- – If load is below 25 %, minimum RAN multiplier value is 1
  – If load is between 25 % to 50 %, minimum RAN multiplier value is 2
  – If load is between 50 % to 75 %, minimum RAN multiplier value is 3
  – If load is between 75 % to 100 %, minimum RAN multiplier value is 4
  – If load is 100 %, then minimum RAN multiplier value is 5

Table 34. SESSION_GET_POSSIBLE_RAN_MULTIPLIER_VALUE_CMD

Payload field(s)	Length	Value/Description
Command	0 Octets	Get possible RAN multiplier value

Table 35. SESSION_GET_POSSIBLE_RAN_MULTIPLIER_VALUE_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	Status code as per Table 53
RAN multiplier value	1 Octet	[1..5] 1: Minimum possible RAN multiplier Value is 1. 2: Minimum possible RAN multiplier Value is 2. 3: Minimum possible RAN multiplier Value is 3. 4: Minimum possible RAN multiplier Value is 4. 5: Minimum possible RAN multiplier Value is 5. Note: The values are indication of system load, the application can set any RAN value using RANGING_INTERVAL parameter.

6 Ranging session management

This section describes the control messages required for starting or stopping the UWB session.

Session must be in SESSION_STATE_IDLE state before starting the session.

UWBD responds RANGE_START_RSP with STATUS_ERROR_SESSION_NOT_CONFIGURED status if session is not in SESSION_STATE_IDLE state on reception of RANGE_START_CMD command.

UWBD shall notify the host by SESSION_STATUS_NTF with status says that UWB session is active or idle on reception of start/stop ranging commands for every UWB session.

UWBD keeps generating the ranging result (RANGE_CCC_DATA_NTF) notification when session is in state SESSION_STATE_ACTIVE.

6.1 Configuration of a ranging session

Below section defines the relevant parameters for each ranging method.

The host must set mandatory APP configurations defined in Table 51 for ranging session. And the host must wait for SESSION_STATUS_NTF with status SESSION_STATE_IDLE to make sure that session is ready for ranging.

Table 36. Mandatory APP configurations for secure car access session

Parameter name
DEVICE_ROLE
NUMBER_OF_ANCHORS
RANGING_SLOT_LENGTH
RESPONDER_SLOT_INDEX

Note: RESPONDER_SLOT_INDEX is not mandatory when DEVICE_ROLE is set 1 (Initiator).

6.2 Ranging start command

The host can request UWBD to start a UWB session by sending RANGE_START_CMD.

After receiving RANGE_START_CMD, UWBD responds by RANGE_START_RSP with status value indicates that ranging command is accepted or not. UWBD shall send the DEVICE_STATUS_NTF notification with status STATUS_ACTIVE when at least one UWB session started successfully.

The host shall be notified with SESSION_STATUS_NTF notification with SESSION_STATE_ACTIVE state on successful activation of the session.

Table 37. RANGE_START_CMD

Payload field(s)	Length	Value/ Description
Session handle	4 Octets	Session handle for which ranging shall start

Table 38. RANGE_START_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	STATUS_OK For various status values, refer to Table 53.

6.3 Ranging resume command (Optional)

RANGE_RESUME_CMD will resume the stopped session from the given STS_INDEX.

After stopping the session, resuming is possible only if all of the static session parameters (which are not marked with (*) in Table 51) remain unchanged. Otherwise, request to resume is rejected.

The response is followed by SESSION_STATUS_NTF, indicating the state transition.

For CCC session, same STS index with same URSK cannot be used again. STS_INDEX in RANGE_RESUME_CMD shall be greater than last STS index used for ranging.

Table 39. RANGE_RESUME_CMD

Payload field(s)	Length	Value/Description
Session handle	4 Octets	Session handle of the session, which shall be resumed
STS index	4 Octets	Resumed STS index

Table 40. RANGE_RESUME_RSP

Payload field(s)	Length	Value/Description
Status	1 Octet	Status code Status REJECTED is returned in the following cases: - static session configuration changed, - session has not been started yet, - resuming is not possible in the current device state.

6.4 Range CCC data notification (CCC controllee)

While the UWB session is ongoing, UWBD shall send ranging result via RANGE_CCC_DATA_NTF notification to the host.

Ranging data payload is shown in below table.

Table 41. RANGE_CCC_DATA_NTF

Payload field(s)	Length	Value/ Description
Session handle	4 Octet	Session handle of the currently active session
Ranging status	1 Octet	STATUS_OK For various status values, refer to Table 53.
STS index	4 Octet	STS index received in final data message
RR index	2 Octet	Next ranging round index
Distance	2 Octet	Distance between initiator and anchor in cm
Uncertainty anchor (FoM)	1 Octet	Ranging timestamp uncertainty of controllee
Uncertainty initiator (FoM)	1 Octet	Ranging timestamp uncertainty of controller
CCM TAG	8 Octets	CCM* TAG calculated over all payload fields. CCM* TAG can be set to all 0xFF if not used.

6.5 Range CCC data notification (CCC controller)

While the UWB session is ongoing, UWBD shall send ranging result via RANGE_CCC_DATA_NTF notification to the host.

Ranging data payload is shown in below table.

Table 42. RANGE_CCC_DATA_NTF

Payload field(s)	Length	Value/ Description
Session handle	4 Octets	Session handle of the currently active session
Ranging status	1 Octet	STATUS_OK For various status values, refer to Table 53.
STS index	4 Octets	STS index of final frame
RR index	2 Octets	Next ranging round index
Response status	4 Octets	Bit mask of responder status. Example: If initiator received response from 0th and 2nd responder, then the response status should be 0x00000005.
CCM TAG	8 Octets	CCM* TAG calculated over all payload fields. CCM* TAG can be set to all 0xFF if not used.

6.6 Ranging stop command

The host shall send RANGE_STOP_CMD to stop the ongoing UWB session. UWBD shall respond the status in RANGE_STOP_RSP command. The host must wait for SESSION_STATUS_NTF notification with status says that UWB session is in idle to make sure that ranging session is stopped.

UWBD shall send the DEVICE_STATUS_NTF notification with status STATUS_READY if all the UWB sessions stopped successfully.

For CCC session, same STS index with same URSK shall not be used again. Therefore after RANGE_STOP_CMD processed, RANGE_START_CMD shall work with new URSK. User shall change URSK via SESSION_SET_APP_CONFIG_CMD. To continue stopped session with same URSK, RANGE_RESUME_CMD shall be used.

Table 43. RANGE_STOP_CMD

Payload field(s)	Length	Value/ Description
Session handle	4 Octet	Stop the requested ranging session

Table 44. RANGE_STOP_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	For various status values refer to Table 53.

6.7 Get ranging count

The host shall use below command to get the number of times ranging has been attempted during ranging session.

Table 45. RANGE_GET_RANGING_COUNT_CMD

Payload field(s)	Length	Value/ Description
Session handle	4 Octets	Session for which ranging count is requested

Table 46. RANGE_GET_RANGING_COUNT_RSP

Payload field(s)	Length	Value/ Description
Status	1 Octet	Count is valid when status STATUS_OK. For various status values, refer to Table 53
Count	4 Octets	Number of times ranging has been attempted

7 Abbreviations

Table 47. Abbreviations

Abbreviations	Description
CMD	Command message
FCS	Frame Check Sequence
GID	Group Identifier
LSB	Least Significant Byte
ms	millisecond
MSB	Most Significant Byte
MT	Message Type
ns	nanosecond
NTF	Notification message
OID	Opcode Identifier
PBF	Packet Boundary Flag
PPM	Part Per Million
RSP	Response message
RSTU	Ranging Scheduling Time Unit
SFD	Start of Frame Delimiter
TCXO	Temperature Compensated Crystol Oscillator
TLV	Type (Tag) Length Value
UCI	Ultrawide Band Command Interface
us	microsecond
UWB	Ultrawide Band
UWBD	UWB Device
XTAL	Crystal Oscillator

8 Appendix A

8.1 Vendor-specific information parameters

Vendor-specific information returned as additional information in CORE_GET_DEVICE_INFO_RSP.

Table 48. Vendor specific information parameters

Name	Length	Tag	Value
CCC specific information			
UCI_CCC_VERSION	2 Octets	0xA0	Version of the UCI CCC specification followed.
CCC_VERSION	8 Octets	0xA1	Version of the CCC specification followed. Octet [0]..Octet[7]: CCC version written in ASCII format
Reserved for UCI CCC use			
RFU		0xA2-0xDF	Reserved for UCI CCC use

8.2 Device capability parameters

Device capability information returned in CORE_GET_CAPS_INFO_RSP.

Table 49. Device capability parameters

Name	Tag	Length	Value	Description
SLOT_BITMASK	0xA0	1 Octet	A bitmap equaling the bit-wise “OR” combination of 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80.	Bitmap of supported values of Slot durations as a multiple of TChap, NChap_per_Slot as defined in CCC Specification. Each “1” in this bit map corresponds to a specific value of NChap_per_Slot where: 0x01 = “3”, 0x02 = “4”, 0x04 = “6”, 0x08 =“8”, 0x10 =“9”, 0x20 = “12”, 0x40 = “24”, 0x80 is reserved. **Note**: 0x40 is used for testing only.
SYNC_CODE_ INDEX_BITMASK	0xA1	4 Octets	A bitmap equaling the bit-wise “OR” combination of 0x00000001, 0x00000002, 0x00000004, … 0x40000000, 0x80000000.	Bitmap of SYNC code indices that can be used. The position of each “1” in this bit pattern corresponds to the index of a SYNC code that can be used, where: 0x00000001 = “1”, 0x00000002 = “2”, 0x00000004 = “3”, 0x00000008 = “4”, …, 0x40000000 = “31”, 0x80000000 = “32”

Table 49. Device capability parameters*...continued*

Name	Tag	Length	Value	Description
				Refer to IEEE 802.15.4-2015 and CCC Specification for SYNC code index definition.
HOPPING_CONFIG_ BITMASK	0xA2	1 Octet	0 - 255	[b7 b6 b5] : bitmask of hopping modes the device offers to use in the ranging session 100 - No Hopping 010 - Continuous Hopping 001 - Adaptive Hopping [b4 b3 b2 b1 b0] : bit mask of hopping sequences the device offers to use in the ranging session b4 = 1 is always set because of the default hopping sequence. Support for it is mandatory. b3 = 1 is set when the optional AES based hopping sequence is supported. Rest of the bits are reserved for future hopping sequences.
CHANNEL_BITMASK	0xA3	1 Octet	A bitmap equaling the bit-wise “OR” combination of 0x01, 0x02 = 0x03	Bitmap of supported UWB channels. Each “1” in this bit map corresponds to a specific value of UWB channel where: 0x01 = "Channel 5" 0x02 = "Channel 9"
SUPPORTED_ PROTOCOL_ VERSION	0xA4	2*n Octets	A list a supported protocol versions where the fields of the protocol versions are 2 bytes long.	Protocol version as defined in the CCC specification is 0x0100. If another futute protocol version 0x0101 would be supported, response has to be: 0x0100, 0x0101. Digital Key applet Protocol Version 1.0 is coded as 0x0100h
SUPPORTED_UWB_ CONFIG_ID	0xA5	2*m Octets	A list of supported UWB configurations where the fields of the UWB configurations are 2 bytes long.	UWB configurations are define in chapter "21.4 UWB Frame Elements" of the CCC specification. Configuration 0x0000 is mandatory for device and vehicle, configuration 0x0001 is mandatory for the device, optional for the vehicle.
SUPPORTED_ PULSESHAPE_ COMBO	0xA6	1*o Octets	A list of supported PulseShape combinations where each PulseShape combination is 1 byte long.	PulseShape combinations are defined in chapter "21.5.3 PulseShape Combinations" of the CCC specification. All possible values defined in CCC specification: 0x00, 0x01, 0x02, 0x10, 0x11, 0x12, 0x20, 0x21, 0x22 ***Note****: A CCC compliant UWB device shall be able to receive all PulseShapes and shall be able to at least transmit one PulseShape. Therefore, an UWB device can support only a subset of all possible CCC PulseShape combinations. Please check the response of CORE_GET_CAPS_INFO_RSP to find out the supported PulseShape combinations of the device.*
Proprietary Use				

Table 49. Device capability parameters*...continued*

Name	Tag	Length	Value	Description
Reserved for proprietary App Configs	0xE3-0xFF			Proprietary ID space

8.3 Device configuration parameters

Table 50. Device configuration parameters

Parameter name	Length (Octets)	Tag (IDs)	Description
DEVICE_STATE	1	0x00	Refer to Table 17 for device status values. Device state will also be notified using DEVICE_STATUS_NTF (default = STATUS_READY) Note: read-only parameter
LOW_POWER_MODE	1	0x01	This config is used to enable/disable the low-power mode 0x00 = Disable low-power mode (default) 0x01 = Enable low-power mode
RFU		0x02-0xDF	Reserved for future use
Reserved for extension of IDs			
ID is 2 Octets in length		0xE0-0xE2	Reserved for future use
Reserved for Proprietary Use			
Proprietary parameters		0xE3-0xFF	Proprietary use

8.4 Application configuration parameters

Parameters marked with (*) are reconfigurable when a session is active.

Parameters marked with (-) cannot be retrieved with SESSION_GET_APP_CONFIG_CMD.

Table 51. Session configuration parameters relevant for CCC session

Parameter name	Length (Octets)	Tag (IDs)	Description
Parameters defined in the Generic UCI Specification			
DEVICE_TYPE	1	0x00	0x00: Controlee 0x01: Controller Default value of this parameter depends on DEVICE_ROLE: Controlee - if the DEVICE_ROLE is responder Controller - if the DEVICE_ROLE is initiator
STS_CONFIG	1	0x02	This parameter indicates how system shall generate the STS. 0: Static STS 1: Dynamic STS (default)
CHANNEL_ID	1	0x04	[5, 6, 8, 9]

Table 51. Session configuration parameters relevant for CCC session*...continued*

Parameter name	Length (Octets)	Tag (IDs)	Description
			CCC specifies support for CH5 and CH9 only. Out of specification usage of other channels is possible (CH6 and CH8)
NUMBER_OF_ANCHORS	1	0x05	Number of anchors (N) 1<=N<=24
DEVICE_MAC_ADDRESS	2	0x06	Mac address of the UWBD. For responder, UWBD can override the device MAC Address. (default = 0)
DST_MAC_ADDRESS	2	0x07	Destination MAC address for broadcast ranging control messages. UWBD can override the destination MAC address (default = 0) Note: This parameter is not supported if DEVICE_ROLE is responder
RANGING_SLOT_LENGTH	2	0x08	Unsigned integer that specifies duration of a ranging slot in the unit of RSTU Possible slot length in terms of RSTU 1200, 1600, 2400, 3200, 3600, 4800, 9600
RANGING_INTERVAL	4	0x09	Ranging interval is time in the unit of 1200 RSTU (=1 ms) between beginning of one ranging round to the beginning of the next. Minimum Ranging interval is 96 which is 115200 RSTU [96 - 24480] RAN_Multiplier = RANGING_INTERVAL / 96 (default = 96)
STS_INDEX0	4	0x0A	STS index initialization value [0 - (2^{30}-1)] (default = 0)
MAC_FCS_TYPE	1	0x0B	CRC type in MAC footer can be set as below: 0x00: CRC 16 (default) 0x01: CRC 32
RNG_DATA_NTF (*)	1	0x0E	0x00: Disable range data notification 0x01: Enable range data notification (default) 0x02: Enable range data notification while in proximity range (Only valid for responder)
RNG_DATA_NTF_PROXIMITY_NEAR (*)	2	0x0F	This parameter sets the lower bound in cm above which the ranging notifications should automatically be enabled if RNG_DATA_NTF is set to 0x02. Should be less than or equal to RNG_DATA_NTF_ PROXIMITY_FAR value. (default = 0) Note: This parameter is not supported if DEVICE_ROLE is initiator
RNG_DATA_NTF_PROXIMITY_FAR (*)	2	0x10	This parameter sets the upper bound in cm above which the ranging notifications should automatically be disabled if RNG_DATA_NTF is set to 0x02.

Table 51. Session configuration parameters relevant for CCC session*...continued*

Parameter name	Length (Octets)	Tag (IDs)	Description
			Should be greater than or equal to RNG_DATA_NTF_PROXIMITY_NEAR value. (default = 20000) Note: This parameter is not supported if DEVICE_ROLE is initiator
DEVICE_ROLE	1	0x11	0x00: Responder 0x01: Initiator
PREAMBLE_ID	1	0x14	Ci Code index If this value is configured as 0xFF(undefined) then Preamble ID in radio configuration will be used. [9 - 12] - support is mandatory [13 - 16], [21 - 24] - optional support (default = 0xFF: Use Preamble ID as in Radio configuration)
SFD ID	1	0x15	Identifier for SFD sequence. If this value is configured as 0xFF(undefined) then preamble ID in radio configuration will be used. [0, 2] (default = 0xFF: Use SFD ID as in radio configuration)
SLOTS_PER_RR	1	0x1B	Number of slots per ranging round. This parameter depends on number of anchors and it overrides the calculated possible minimum. [6, 8, 9, 12, 16, 18, 24, 32]
KEY_ROTATION	1	0x23	This configuration is used to enable/disable STS and Final-Data protection key (dUDSK) rotation. STS key rotation configuration is applied only if STS_CONFIG is dynamic (0x01). 0x00: No rotation 0x01: STS key rotation (only if STS_CONFIG = 0x01) 0x02: dUDSK rotation 0x03: STS key (only if STS_CONFIG = 0x01) and dUDSK rotation (default = 0x03)
SESSION_PRIORITY (*)	1	0x25	Priority of a session [1-100] where 100 is the highest priority (default = 50)
MAX_RR_RETRY	2	0x2A	Number of failed Ranging Round (RR) attempts before terminating the session. The host shall receive SESSION_STATUS_NTF with MAX_RANGING_ROUND_RETRY_COUNT_REACHED status when consecutive ranging is not succeeded for maximum ranging round attempts. [1 - 65535] [0]: Termination is disabled and ranging round attempt number is infinite (default = 0)

Table 51. Session configuration parameters relevant for CCC session*...continued*

Parameter name	Length (Octets)	Tag (IDs)	Description
UWB_INITIATION_TIME	8	0x2B	The UWB initiation time is the absolute time in the UWBS time domain in which the first message in the ranging round shall be transmitted after the session is started. The value shall be specified in the units of microseconds. If the value is set to 0 then UWB_INITIATION_TIME shall not be applicable. The value ranges from 1 to (2^{64} - 1) to specify the UWB_INITIATION_TIME. (default = 0)
HOPPING_MODE	1	0x2C	0: no hopping 2: adaptive hopping using MODULO 3: continuous hopping using MODULO 4: adaptive hopping using AES 5: continuous hopping using AES (default = 0)
MAX_NUMBER_OF_MEASUREMENTS	2	0x32	Numbers of blocks to be executed in a session The host shall receive SESSION_STATUS_NTF with MAX_RANGING_BLOCKS_REACHED status when executed number of blocks are reached to MAX_NUMBER_OF_MEASUREMENTS. [0x0001 – 0xFFFF] (default = 0xFFFF)
RFU		0x33-0x9F	Reserved for Future Use
Parameters valid only for the CCC Session			
HOP_MODE_KEY	4	0xA0	Key to generate hopping sequence This value is used for both AES and MODULO hopping formula. For MODULO hopping, only first 4 bytes are used as converted to 4byte integer. Default key for AES hopping formula = {0x4c,0x57,0x72,0xbc} Default key for MODULO hopping formula = 0xbc72574c
CCC_CONFIG_QUIRKS	1	0xA1	0x00: Key identifier field of auxiliary security header of the MAC header as per CCC Digital Key Release 3, Version 1.0.0 0x01: Key identifier field formatted as: Source short address || Source PAN ID (default = 0x01)
RESPONDER_SLOT_INDEX (*)	1	0xA2	This parameter is used to choose responder index in two way ranging. It is not applicable to controller. N is a number of anchors. 0 – Responder 1 1 – Responder 2 … N-1 – Responder N

Table 51. Session configuration parameters relevant for CCC session*...continued*

Parameter name	Length (Octets)	Tag (IDs)	Description
			Note: This parameter is not supported if DEVICE_ROLE is initiator
RANGING_PROTOCOL_VER	2	0xA3	Version of the ranging protocol (defined by CCC) [0x0000 – 0xFFFF] (default = 0x0100)
UWB_CONFIG_ID	2	0xA4	UWB Configuration ID [0x0000 – 0xFFFF] (default = 0x0001)
PULSESHAPE_COMBO	1	0xA5	Possible combinations are written in format: Pulse shape combo value - Initiator transmit pulse shape - Responder transmit pulse shape 0x00 - 0x0 - 0x0 0x01 - 0x0 - 0x1 0x02 - 0x0 - 0x2 0x10 - 0x1 - 0x0 0x11 - 0x1 - 0x1 0x12 - 0x1 - 0x2 0x20 - 0x2 - 0x0 0x21 - 0x2 - 0x1 0x22 - 0x2 - 0x2 Support for value 0x00 is mandatory. (default = 0x00)
URSK_TTL	2	0xA6	URSK expiration time, in minutes (max 12 hours). After this time from setting URSK, the session will go to idle. [0x001 - 0x2D0] (default = 0x2D0)
RESPONDER_LISTEN_ONLY (*)	1	0xA7	Responder is in listen only mode. If it is enabled, then the responder will not send range frame. In this mode, the responder will report 0xD (responder is in listen only mode) as higher nibble of ranging status in RANGE_CCC_DATA_NTF 0x00: Responder is in normal mode. 0x01: Responder is in listen only mode (default = 0x00)
LAST_STS_INDEX_USED	4	0xA8	Parameter used to get the STS index of the UWB session. When SESSION_GET_APP_CONFIG_CMD issued for this config during SESSION_STATE_ACTIVE the UWBD shall return the last STS Index of the latest completed ranging block. ***Note:*** *The value of this parameter shall be ignored by UWBD when SESSION_SET_APP_CONFIG_CMD with this config is issued in SESSION_STATE_INIT/SESSION_STATE_IDLE state*
RFU		0xA8-0xDF	Reserved for future use
Reserved for extension of IDs			

Table 51. Session configuration parameters relevant for CCC session*...continued*

Parameter name	Length (Octets)	Tag (IDs)	Description
ID is 2 Octets in length		0xE0-0x E2	ID is 2 Octets in length
Reserved for Proprietary Use			
Proprietary App Configs		0xE3-0x FF	Proprietary ID space

8.5 GID and OID definitions

Table 52. GID and OID Definitions

GID	OID	Control message name	Description
UCI core group 0000b (0x0)	000000b (0x00)	DEVICE_RESET_CMD DEVICE_RESET_RSP	To reset the UWBD
	000001b (0x01)	DEVICE_STATUS_NTF	To notify about the current state of the UWBD
	000010b (0x02)	CORE_GET_DEVICE_INFO_CMD CORE_GET_DEVICE_INFO_RSP	To get the UWBD device information
	000011b (0x03)	CORE_GET_CAPS_INFO_CMD CORE_GET_CAPS_INFO_RSP	To get the UWBD capability information
	000100b (0x04)	CORE_SET_CONFIG_CMD CORE_SET_CONFIG_RSP	To set the UWBD device configurations
	000101b (0x05)	CORE_GET_CONFIG_CMD CORE_GET_CONFIG_RSP	To get the UWBD device configurations
	000110b (0x06)	RFU	Reserved for Future Use
	000111b (0x07)	CORE_GENERIC_ERROR_NTF	To notify about any generic errors in UWBD
UWB session config group 0001b (0x1)	000000b (0x00)	SESSION_INIT_CMD SESSION_INIT_RSP	To create a UWB session and initialize it with the default values
	000001b (0x01)	SESSION_DEINIT_CMD SESSION_DEINIT_RSP	To delete a UWB session
	000010b (0x02)	SESSION_STATUS_NTF	To notify about the current UWB session state
	000011b (0x03)	SESSION_SET_APP_CONFIG_CMD SESSION_SET_APP_CONFIG_RSP	To configure a UWB session with specified configurations
	000100b (0x04)	SESSION_GET_APP_CONFIG_CMD SESSION_GET_APP_CONFIG_RSP	To get the current configuration for specified UWB session
	000101b (0x05)	SESSION_GET_COUNT_CMD SESSION_GET_COUNT_RSP	To get the number of the UWB sessions in UWBD device
	000110b (0x06)	SESSION_GET_STATE_CMD SESSION_GET_STATE_RSP	To get the current state of the UWB session

Table 52. GID and OID Definitions*...continued*

GID	OID	Control message name	Description
	100000b (0x20)	SESSION_GET_POSSIBLE_RAN_MULTIPLIER_VALUE_CMD SESSION_GET_POSSIBLE_RAN_MULTIPLIER_VALUE_RSP	To check possible RAN multiplier value for a new session.
UWB ranging session control group 0010b (0x2)	000000b (0x00)	RANGE_START_CMD RANGE_START_RSP	To activate a UWB ranging session
	000001b (0x01)	RANGE_STOP_CMD RANGE_STOP_RSP	To deactivate a ranging session
	000011b (0x03)	RANGE_GET_RANGING_COUNT_CMD RANGE_GET_RANGING_COUNT_RSP	To get the number of ranging rounds attempted
	100000b (0x20)	RANGE_CCC_DATA_NTF	To notify about the ranging results from UWBD device after successful activation of a ranging session
	100001b (0x21)	RANGE_RESUME_CMD (Optional) RANGE_RESUME_RSP (Optional)	To resume a session, which has been stopped
0100b (0x4) – 1000b (0x8)	000000b (0x00) – 111111b (0x3F)	RFU	Reserved for Future Use
Vendor-specific group 1001b (0x9) - 1100b (0xB)	000000b (0x00) – 111111b (0x3F)	For vendor-specific use	For vendor-specific use
Test group 1101b (0xC)	000000b (0x00) – 111111b (0x3F)	Test group commands	Defined in [FiRa UWB UCI Test Specification]
Vendor-specific group 1110b (0xE) – 1111b (0xF)	000000b – 111111b	For Vendor Specific use	Vendor Specific use

8.6 Status codes

Table 53. Status codes

Status codes	Name	Description
	Generic status codes	
0x00	STATUS_OK	Success.
0x01	STATUS_REJECTED	Intended operation is not supported in the current state.
0x02	STATUS_FAILED	Intended operation failed to complete.

Table 53. Status codes*...continued*

Status codes	Name	Description
0x03	STATUS_SYNTAX_ERROR	UCI packet structure is not per spec.
0x04	STATUS_INVALID_PARAM	Config ID is correct, and value is not specified
0x05	STATUS_INVALID_RANGE	Config ID is correct, and value is not in proper range
0x06	STATUS_INVALID_MESSAGE_SIZE	UCI packet payload size is not as per spec.
0x07	STATUS_UNKNOWN_GID	UCI Group ID is not per spec.
0x08	STATUS_UNKNOWN_OID	UCI Opcode ID is not per spec.
0x09	STATUS_READ_ONLY	Config ID is read-only.
0x0A	STATUS_COMMAND_RETRY	UWBD request retransmission from the host
0x0B	STATUS_UNKNOWN	It is not known whether the intended operation was failed or successful.
0x0C	STATUS_NOT_APPLICABLE	The parameter ID is not applicable for the selected operation on the requested session.
0x0B – 0x0F	RFU	Reserved for future use.
UWB session specific status codes		
0x11	STATUS_ERROR_SESSION_NOT_EXIST	Session does not exist (is not created)
0x12	RFU	Reserved for future use.
0x13	STATUS_ERROR_SESSION_ACTIVE	Session is active.
0x14	STATUS_ERROR_MAX_SESSIONS_EXCEEDED	Max. number of sessions already created.
0x15	STATUS_ERROR_SESSION_NOT_CONFIGURED	Session is not configured with required app configurations.
0x16	STATUS_ERROR_ACTIVE_SESSIONS_ONGOING	Sessions are actively running in UWBD
0x17	RFU	Reserved for future use.
0x18	RFU	Reserved for future use.
0x19	RFU	Reserved for future use.
0x1A	STATUS_ERROR_UWB_INITIATION_TIME_TOO_OLD	Session is not configured with required app configurations.
0x1B	STATUS_OK_NEGATIVE_DISTANCE_REPORT	Success. A negative distance was measured: Distance in Table 41is the absolute value of the measurement.
0x1C-0x1F	RFU	Reserved for future use.

Table 53. Status codes*...continued*

Status codes	Name	Description
UWB ranging session specific status codes		
0x20	STATUS_RANGING_TX_FAILED	Failed to transmit UWB packet.
0x21	STATUS_RANGING_RX_TIMEOUT	No UWB packet detected by the receiver.
0x22	STATUS_RANGING_RX_PHY_DEC_FAILED	UWB packet channel decoding error.
0x23	STATUS_RANGING_RX_PHY_TOA_FAILED	Failed to detect time of arrival of the UWB packet from CIR samples.
0x24	STATUS_RANGING_RX_PHY_STS_FAILED	UWB packet STS segment mismatch.
0x25	STATUS_RANGING_RX_MAC_DEC_FAILED	MAC CRC or syntax error.
0x26	STATUS_RANGING_RX_MAC_IE_DEC_FAILED	IE syntax error.
0x27	STATUS_RANGING_RX_MAC_IE_MISSING	Expected IE missing in the packet.
0x28-0x4F	RFU	Reserved for future use.
Vendor specific status codes		
0xA0	STATUS_INVALID_RESPONDER_SLOT_INDEX	Responder slot index is invalid in the requested configuration.
0xA1	STATUS_SLOT_LEN_NOT_SUPPORTED	Slot length is not supported in the requested configuration
0xA2	STATUS_INVALID_SLOT_PER_RR	Number of slots is invalid in the requested configuration
0xA3	STATUS_INVALID_STS_IDX	STS index is not valid.
0xA4	STATUS_RESPONDER_LISTEN_ONLY_MODE	Responder is in the listen only mode. Responder does not send a range frame.
Proprietary status codes		
0x50 – 0xFF	Proprietary status code	Proprietary status codes

9 Legal information

9.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

9.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

9.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

All information provided in this document is subject to legal disclaimers.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com

Date of release: 14 November 2023
Document identifier: NXP UCI CCC Specification Confidential
Document number: UM11539