AT Commands Manual
for VULKANO UWB Development Kit

Rev: 0.7
Date: Aug. 9 2023

Introduction

Ultrawideband Ranging (UWB), a featured function embedded in TrueSense DCU040 and DCU150 modules, can help customers get the current accurate distance and directions (in terms of both azimuth and elevation), between IoT devices equipped with the technology.

The Vulkano Development Kit (VDK) is a UWB-oriented prototyping platform enabling users to quickly experience the technology and anticipate the software integration effort before the UWB modules get embedded directly into their mass production devices.

The VDK hardware is based on modular approach and can host either a DCU150 UWB Anchor or a DCU040 UWB Tag so as to implement all possible UWB Ranging configurations, including fully secured ones thanks to the presence of a Secure Element chip onboard of the Kit.

VDK has 2 modes of operation:

1- **AT mode**:
In this mode the VDK runs a dedicated firmware onboard of the QN9090 BLE5.0 MCU that exposes a AT-based API through a UART port. By hiding all the complexity of the UWB command interface protocol, this approach allows for a significantly simpler integration by means of a well-known programming pattern using AT commands.

2- **Passthrough mode**:
In this mode the control of the DCU040/DCU150 modules onboard of the VDK is totally delegated to a 3rd-party host MCU/CPU board via the SPI interface available on the VDK's Arduino Shield pinout.

This document covers the AT-based interface supported for operation mode #1 .

Please note that some advanced usages of the AT API do require partial knowledge of the UCI Specification developed by the FiRa Consortium (https://www.firaconsortium.org).

Providing full details about the spec is out of the scope of this document and you should refer to FiRa in order to get access to it.

AT Commands for Vulkano Development Kit

Overview of AT Commands

The commands below are only valid for the **AT Mode** solution:

Command	Description
AT+UINIT	Init the DCU module
AT+URESET	Reset the DCU module
AT+USUSPEND	Suspend the DCU Module
AT+UGETCAPS	Get DCU device capabilities
AT+UDEVSTATE	Get device information
AT+UCFG	Set/Get session configuration params
AT+UCFGARRAY	Set/Get session config params for array values
AT+URNGPARAMS	Config ranging params
AT+URNGSTART	Start ranging session
AT+USESSIONDEINIT	Deinitialize a session
AT+USESSIONINIT	Initialize a session
AT+UAPPCOMMONCFG	Set common configuration parameters
AT+URXDATA	Receive data in a ranging session
AT+UTXDATA	Send data in a ranging session
AT+UIMODE	Enter iOS/Android peer2peer ranging mode
AT#HELP	Provide support about available AT Commands
AT#QUIT	Reboot the system

Functions

AT+UINIT Initialize the UWB module

AT+UINIT Initialize the UWB module	
Test Command AT+UINIT=?	Response +UINIT=<INIT:UINT8[RW]> OK
Read Command AT+UINIT?	Response +UINIT=0 or 1 (0 means not initialized) OK
Write Command AT+UINIT=<init>	Response OK If there is an error: +UERR: <err>

Parameter

<init>	0 deinit ,1 init

Error codes

STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+URESET Reset the UWB module+UR

AT+URESET Reset the UWB module	
Execution Command AT+URESET	Response OK If there is an error: +UERR: <err>

Error codes

STATUS_TIMEOUT – if command is timeout
STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_FAILED – otherwise

AT+UGETCAPS get the capability of the UWB module

AT+UGETCAPS get the capability of the UWB module	
Execute Command AT+UGETCAPS	Response (example, the numbers may be different) +UGETCAPS: FW 1.2.3 MAC 1.0 MW 1.2.3 UCI 1.0.0 PHY 1.0 OK If there is an error: +UERR: <err>

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+UDEVSTATE retrieve the device status of the UWB module

AT+USUSPEND retrieve the device information of the UWB module	
Read Command AT+UDEVSTATE	Response +UDEVSTATE: xx (where xx is the state) OK If there is an error: +UERR: <err>

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+UCFG set/get configuration parameters

AT+UCFG set configuration parameters	
Test Command AT+UCFG=?	Response OK
Write Command AT+UCFG=<session_id>,<config_id>,<value>	Response OK If there is an error: +UERR: <err>
Read Command AT+UCFG=<session_id>,<config_id>	Response +UCFG: <config_id>,<value> OK If there is an error: +UERR: <err>
Run Command AT+UCFG	This command applies all configs set by the Write command of AT+UCFG and AT+UCFGARRAY Response OK If there is an error: +UERR: <err>

Parameter

<session_id>	ID of the UWB session
<config_id>	ID of the parameter to set (ARRAY values NOT supported)
<value>	new value for the parameter

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_SESSION_NOT_EXIST – if session is not initialized
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+UCFGARRAY set configuration parameters (for array values)

AT+USETCFG set configuration parameters	
Test Command AT+UCFGARRAY=?	Response OK
Write Command AT+UCFGARRAY=<session_id>,<config_id>,<value>	Response OK If there is an error: +UERR: <err>

Parameter

<session_id>	ID of the UWB session
<config_id>	D of the parameter to set (only ARRAY values ARE supported)
<value>	new value for the parameter

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_SESSION_NOT_EXIST – if session is not initialized
STATUS_TIMEOUT – if command is timeout

STATUS_FAILED – otherwise

AT+URNGPARAMS Set params for ranging session

AT+URNGPARAMS start a ranging session	
Test Command AT+URNGPARAMS=?	Response OK
Write Command AT+URNGPARAMS=<session_id>, <multiNode_mode>, <device_role>, <device_type>, <num_controlees>, <mac_addr_mode>, <dev_mac_addr>, <dst_mac_addr>,	Response OK If there is an error: +UERR: <err>

Parameter

<session_id>	ID of the UWB session
<multinode_mode>	UniCast = 0, OnetoMany = 1, ManytoMany = 2
<device_role>	Responder = 0, Initiator = 1, Master_Anchor = 2, Initiator_And_Responder = 3, Receiver = 4, Advertiser= 5, Observer = 6, DlTDoA_Anchor = 7, DlTDoA_Tag= 8
<device_type>	Controlee = 0, Controller = 1
<num_controlees>	integer
<mac_addr_mode>	0: short 2 byte, 1: extended 8 byte mode
<dev_mac_addr>	hex array (2/8 bytes, depending on mac_addr_mode>
<dst_mac_addr>	hex array

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_SESSION_NOT_EXIST – if session is not initialized
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+URNGSTART Start a ranging session

AT+URNGSTART start a ranging session	
Test Command AT+URNGSTART=?	Response OK
Write Command AT+URNGSTART=<session_id>, <start>	Response OK If there is an error: +UERR: <err>

Parameter

<session_id>	ID of the UWB session
<start>	1: start session, 0: stop session

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_SESSION_NOT_EXIST – if session is not initialized
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+USESSIONDEINIT Deinit a ranging session

AT+USETCFG set configuration parameters	
Test Command AT+USESSIONDEINIT =?	Response OK
Write Command AT+USESSIONDEINIT =<session_id>	Response OK If there is an error: +UERR: <err> OK

Parameter

<session_id>	ID of the UWB session

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_SESSION_NOT_EXISTS – if session with the session ID passed doesn't exist
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+USESSIONINIT Deinit a ranging session

AT+USESSIONINIT initialize	
Test Command AT+USESSIONINIT =?	Response OK
Write Command AT+USESSIONINIT =<session_id>,<session_type>	Response OK If there is an error: +UERR: <err>

Parameter

<session_id>	ID of the UWB session
<session_type>	session_type:0 (SESSION_RANGING), 176 (SESSION_DATA_TRANSFER)

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_SESSION_DUPLICATE – if session with the session ID passed already exists
STATUS_MAX_SESSIONS_EXCEEDED – if more than 5 sessions are exceeded
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+UAPPCOMMONCFG set common configuration parameters (only for DCU150)

AT+UAPPCOMMONCFG set common application configuration parameters	
Test Command AT+UAPPCOMMONCFG=?	Response OK
Write Command AT+UAPPCOMMONCFG=<session_id>,<sts_config>	Response OK If there is an error: +UERR: <err>

Parameter

<session_id>	ID of the UWB session
<sts_config>	0 (CONTROLLER), 1 (CONTROLEE)

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_SESSION_NOT_EXIST – if session is not initialized with sessionId
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

This API will set the following default values
UWB_SET_APP_PARAM_VALUE(*SFD_ID*, 2),
UWB_SET_APP_PARAM_VALUE(*SLOTS_PER_RR*, 25),
UWB_SET_APP_PARAM_VALUE(*RANGING_INTERVAL*, 200),
UWB_SET_APP_PARAM_VALUE(*NUMBER_OF_STS_SEGMENTS*, 1)
UWB_SET_APP_PARAM_VALUE(*RFRAME_CONFIG*, 3),

AT+USTSCFG set the STS configuration

AT+USTSCFG set the STS configuration	
Test Command AT+USTSCFG=?	Response OK
Write Command AT+USTSCFG=<session_id>,<vendor_id><sts_iv>	Response OK If there is an error: +UERR: <err>

Parameter

<session_id>	ID of the UWB session
<vendor_id>	ID of vendor
<stsIV>	sts key IV

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_SESSION_NOT_EXIST – if session is not initialized with sessionId
STATUS_TIMEOUT – if command is timeout
STATUS_FAILED – otherwise

AT+UTXDATA transfer data array over UWB session

AT+UDATATX transfer data array over UWB	
Test Command AT+UTXDATA=?	Response OK
Write Command AT+UTXDATATX=<session_id>,<dst_addr>,<data>	Response OK If there is an error: +UERR: <err>

Parameter

<session_id>	ID of the UWB session
<dst_addr>	MAC address of destination (hex array)
<data>	Data packet (hex array)

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_TIMEOUT – if command is timeout
STATUS_REJECTED – if session is not established when data packet sent
STATUS_NO_CREDIT_AVAILABLE – if buffer is not available to accept data
STATUS_DATA_TRANSFER_ERROR – if data is not sent due to an unrecoverable error
STATUS_FAILED – otherwise

AT+URXDATA receive data array over UWB session

AT+UDATATX transfer data array over UWB	
Test Command AT+URXDATA=?	Response OK
Write Command AT+URXDATA=<delay>	Response OK If there is an error: +UERR: <err>

Parameter

<delay>	Time (in ms) to wait for data

Error codes

STATUS_NOT_INITIALIZED – if UWB stack is not initialized
STATUS_INVALID_PARAM – if invalid parameters are passed
STATUS_TIMEOUT – if command is timeout
STATUS_REJECTED – if session is not established when data packet sent
STATUS_NO_CREDIT_AVAILABLE – if buffer is not available to accept data
STATUS_DATA_TRANSFER_ERROR – if data is not sent due to an unrecoverable error
STATUS_FAILED – otherwise

AT+UIMODE Enter iOS/Android Peer2Peer ranging mode (aka iMode)

AT+UIMODE Enter iOS/Android Peer2Peer ranging mode	
Test Command AT+UIMODE=?	Response OK
Run Command AT+UIMODE	Response OK If there is an error: +UERR: Could not create iMode task

Please note that entering the iOS/Android ranging mode will override all configurations and ongoing sessions previously issued by using the other AT Commands detailed above.

In iMode the device will enter a state where it will wait for a BLE connection to be established by a mobile App for iOS or Android. App and Vulkano Dev Kit will use this connection to exchange the parameters needed to establish a new UWB ranging session and eventually start it.

TRUESENSE provide reference source code for both iOS and Android, respectively using the Apple Nearby Interaction Framework (https://developer.apple.com/documentation/nearbyinteraction/) and Android Core UWB API via the androidx.core.uwb jetpack library (https://developer.android.com/jetpack/androidx/releases/core-uwb)

Please refer to https://ultrawideband.truesense.it/support for access to source code of the apps and related documentation

Notifications

When a session is active and started the UWB module will print to console a series of notifications, depending on the operation being performed.
In this chapter we detail the notifications that can be used to derive useful information for users of the AT Commands interface.

Please note, C printf() format specifiers are used in the following sections in order to provide a clear indication of which type of dynamic data will be printed to console. Each parameter is described on a dedicated line, reporting it's meaning and the data type.

Session Status notification

+USESSSTATUS: session_id	: %" PRIu32 "
,state	: %hu
,reason_code	: %hu

Session Data notification

+USESSDATA: Status,	: %" PRIu32 "
, Session Counter	:%d "
:For each session:	
+USESSDATA:Session_num	:%d
,Session ID	:, %" PRIu32 "
,type	:%hu
state	:%hu

Multicast session list notification

+UMCASTSTATUS:session_id,	: %" PRIu32 "
remaining_list,	:%hu
no_of_controlees	: %hu
For each controlee:	
+UMCASTSTATUS:mac_address	: %" PRIu16 "
, subsession_id	:%" PRIu32 "
,status,	: %hu

Ranging params notification

+URNGPARAMS:deviceRole :%hu,
,multiNodeMode, :%hu
,macAddrMode :%hu
,noOfControlees :%hu
,deviceMacAddr :<HEX ARRAY>
For each controlee:
+URNGPARAMS: deviceMacAddr, :<HEX ARRAY>,
deviceType :%hu

Ranging data notification

Ranging data notifications are the most complex and their output depends on the type of session currently being active.
All session types share an initial part as per below:

+URNGDATA:seq_ctr :%" PRIu32 "
,sessionId :,0x%x
,rcr_indication, :%hu
,curr_range_interval, :%hu
,ranging_measure_type, :%hu
,mac_addr_mode_indicator :%hu
,no_of_measurements :%hu :

For sessions where measure_type is MEASUREMENT_TYPE_TWOWAY:
For each measurement i:
+URNGDATA:mac_addr : <HEX ARRAY>

,status : %x "
If status is OK:
nLos : %hu //1 if not line of sight
,distance : %" PRIu16 " //in centimeters
aoa_azimuth : %d.%d //in degrees
aoa_azimuth_FOM : %hu //FOM is reliability index
aoa_elevation : %d.%d
aoa_elevation_FOM : %hu
aoa_dest_azimuth : %" PRIi16 "
aoa_dest_azimuth_FOM : %hu
aoa_dest_elevation : %" PRIi16 "
aoa_dest_elevation_FOM : %hu
,slot_index : %hu

For sessions where measure_type is MEASUREMENT_TYPE_ONEWAY (UL TDoA Anchor)

For each measurement i:

+URNGDATA:mac_addr : <HEX ARRAY>
,frame_type : %hu
,nLos : %hu",
,aoa_azimuth : %" PRIi16 "
,aoa_azimuth_FOM : %hu
,aoa_elevation : %" PRIi16 "
,aoa_elevation_FOM : %hu
,timestamp : %" PRIu64 "
,blink_frame_number : %" PRIu32 "
,device_info_size : %hu
,device_info : <HEX ARRAY>
.blink_payload_size : %hu
blink_payload : <HEX ARRAY>

//ONLY FOR DCU150
,rssi_rx1 : %" PRIi16 "
,rssi_rx2 : %" PRIi16 "
,pdoaFirst :%" PRIi16 “
,pdoaFirstIndex : %" PRIu16 "
,pdoaSecond : %" PRIi16 "
pdoaSecondIndex : %" PRIu16 "
//end of SR150 specific

For sessions where measure_type is MEASUREMENT_TYPE_DLTDOA

For each measurement i:

If status is OK

+URNGDATA:mac_addr : <HEX ARRAY>
,status : %x
,message_type : %x
,message_control : %x
,block_index : %x
,round_index : %x
,nLoS : %x
,aoa_azimuth : %x
,aoa_azimuth_fom : %x
,aoa_elevation : %x
,aoa_elevation_fom : %x
,rssi : %x
,tx_timestamp : <HEX ARRAY>

,rx_timestamp :<HEX ARRAY>
.cfo_anchor : %x
cfo : %x
reply_time_initiator : %x
,reply_time_responder : %x
,initiator_responder_tof : %x

For sessions where measure_type is MEASUREMENT_TYPE_OWR_WITH_AOA
For each measurement i:
+URNGDATA:mac_addr : <HEX ARRAY>
,status : %x
,nLos : %x
,frame_seq_num : %x
,block_index : %d
,aoa_azimuth : %x
,aoa_azimuth_FOM : %x
,aoa_elevation : %d
,aoa_elevation_FOM : %x

Data Transmission status notification

+UTXSTATUS:session_id : %x
,sequence_number : %d
,status : %d

Data reception notification

+URXSTATUS:session_id : %x
,status : %d
,sequence_number : %x
,src_address : <HEX ARRAY>
,src_endpoint : %x
,dst_endpoint : %d
,data_size : %x
,data : <HEX ARRAY>

Examples

Please see Truesense's AT commands examples on Github:

https://github.com/Truesense-it/UWB_AT_Examples

Appendix 1:
Error codes

Command succeeded
STATUS_OK 0x00

Request is rejected
STATUS_REJECTED 0x01

Command Failed
STATUS_FAILED 0x02

API called without UCI being initialized
STATUS_NOT_INITIALIZED 0x03

Invalid parameter provided
STATUS_INVALID_PARAM 0x04

Invalid value range provided
STATUS_INVALID_RANGE 0x05

Session wrt SESSION ID Does not exist
STATUS_SESSION_NOT_EXIST 0x11

Session wrt SESSION ID Already Present
STATUS_SESSION_DUPLICATE 0x12

Session active
STATUS_SESSION_ACTIVE 0x13

MAX Sessions exceeded
STATUS_MAX_SESSIONS_EXCEEDED 0x14

Operation is started with out configuring required parameters for Session
STATUS_SESSION_NOT_CONFIGURED 0x15

sessions ongoing
STATUS_SESSIONS_ONGOING 0X16

Indicates when multicast list is full during one to many ranging

STATUS_MULTICAST_LIST_FULL 0x17

ESE Rest happened during command processing.
STATUS_ESE_RESET 0x71

Unrecoverable data transfer error
STATUS_DATA_TRANSFER_ERROR 0x30

Credit not available for Data Packet
STATUS_NO_CREDIT_AVAILABLE 0x31

given round index couldn't be activated
STATUS_ERROR_ROUND_INDEX_NOT_ACTIVATED 0x28

given round exceeds the possible number of ranging rounds
STATUS_ERROR_NUMBER_OF_ACTIVE_RANGING_ROUNDS_EXCEEDED 0x29

The role for the configured ranging round index is not Initiator and therefore RDM list cannot be set
STATUS_ERROR_ROUND_INDEX_NOT_SET_AS_INITIATOR 0x2A

device address not matching
STATUS_DLTDOA_DEVICE_ADDRESS_NOT_MATCHING_IN_REPLY_TIMELIST 0x30

Buffer overflow
STATUS_BUFFER_OVERFLOW 0xFA

Status PBF=1 CMD SENT
STATUS_PBF_PKT_SENT 0xFB

Device is woken up from HPD
STATUS_HPD_WAKEUP 0xFC

Command failed with time out
STATUS_TIMEOUT 0xFD

SE Error
STATUS_ESE_ERROR 0xFF

Ranging suspended

STATUS_SUSPEND 0x8B