RangeNet Application Programming Interface (API) Specification

TDSR UWB Radios

TDSR Headquarters
810 Tight Bark Hollow Road
Petersburg, TN 37144 USA
www.tdsr-uwb.com
Tel: +1 256.617.3132
+1 256.990.4217

TDSR

320-0313I
October 2020

Copyright

Trademarks

Rights

Table of Contents

1 Introduction

This manual specifies the programmer's interface between the user's Host processor and any member of the P400 series of single-board Ultra Wideband (UWB) platforms. As of this date, this family includes the P400, P410, P412 and P440, all of which can interoperate. These units (shown in **Figure 1)** and are collectively referred to as P4xx platforms.

The P4xx platforms use Two-Way Time-of-Flight (TW-TOF) ranging to accurately measure the distance between two units. TW-TOF distance measurements are referred to as Precision Range Measurements (PRMs). The P4xx platforms also measure the signal strength of the first arriving energy to provide a Coarse Range Estimate (CRE). Since a transmission from one unit can be heard by many units, CREs are effectively a broadcast in that every transmission will result in a CRE being generated at every receiving unit. To maintain accuracy, the CREs are periodically updated with PRMs. The recalibrated, filtered CREs are provided in the form of Filtered Range Estimates (FREs).

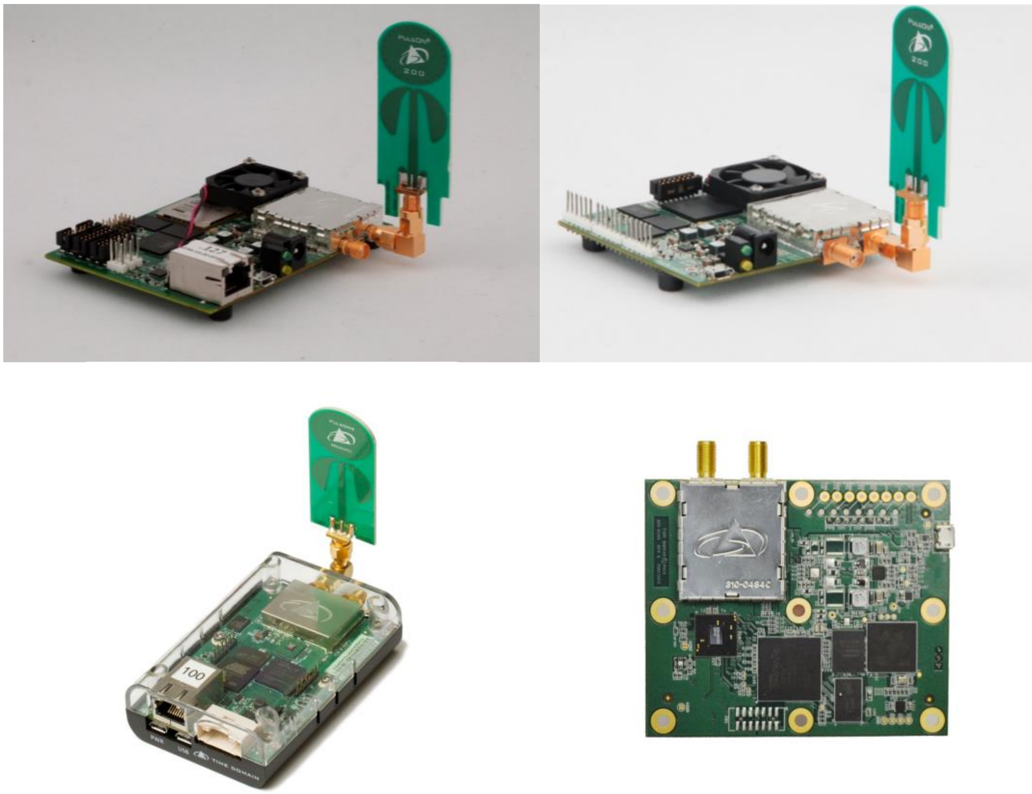


Fig. 1: Clockwise from upper left P400, P410, P412, and P440 (shown with attached Broadspec Antenna)

The P4xx units can be operated as standalone devices, in that they do not need to be connected to a Host computer. In such cases they will operate in whatever state they were last used.

The units can be operated as peer-to-peer ranging devices (Ranging Mode) or as part of the RangeNet network. RangeNet is TDSR's networking protocol and is specifically designed to address the needs of TW-TOF as well as accommodate CREs/FREs. RangeNet is resident on the P4xx and supports either the TDMA or ALOHA protocols.

When operating in RangeNet Mode, the user has the option of engaging TDSR's Location Engine. This localizer is resident on the P4xx and computes the current location of the P4xx using either a Geometric (Least Squares Fit) algorithm or a Kalman Filter (and, of course, inter-unit TW-TOF range measurements). The Location Engine will report either 2D (XY) or 3D (XYZ) positions based on a user-defined setting. If the user has selected 2D positioning, then Z is assumed to be fixed and is user-definable.

RangeNet Lite is available on all P4xx platforms. This version is node-locked such that operation is restricted to a maximum of 10 nodes. Users interested in a obtaining an unrestricted license should contact TDSR directly.

A more detailed description of RangeNet network and localization operations will be presented later in this document. Additional information can be found in **Section 7 - Network Principles** and **Section 10 - Localization Principles** of the *320-0320 RangeNet User Guide*. The User Guide describes the operation of TDSR's Host-based Graphical User Interface (GUI). The RangeNet GUI is provided with the TDSR Ranging and Localization Kits and TDSR Labs. In particular, the RangeNet GUI enables the user to:

- Operate the P4xx in any of the various modes
- Modify settings, parameters, and configurations
- Display range information, captured waveform scans, network activity, and locations
- Log any information passed between the Host and a P4xx

The RangeNet GUI is an excellent tool for learning how the system works, demonstrating P4xx capabilities to others, and measuring performance. This MS Windows PC application provides a graphical representation of the interface data structures and allows the user to quickly become familiar with Host and P4xx behaviors. The RangeNet GUI is provided with a Quick Start Guide. The *320-0314 RangeNet Quick Start Guide* walks the user through initial set-up and demonstrates each mode of operation.

Programmers are strongly encouraged to review and make use of the sample C and MATLAB code provided with the software release. Doing so will accelerate your development efforts.

Section 3 - RCM API Messages describes all of the instructions relative to operating in Ranging Mode. **Section 4 - RangeNet API Messages** describes all of the instructions relative to operating in RangeNet Mode. **Section 5 - Location API Messages** describes all of the instructions relative to operating in the Location Modes. Changing from one mode to the other is described in **Section 3.17**.

A separate application note, *320-0287 Using the USB and Serial Interfaces*, describes the extended header bytes and protocol required to support the USB and 3.3V TTL Serial UART interfaces. This document provides a reference of the message structures and bit patterns in an Ethernet UDP/IP programming interface.

Additionally, RangeNet also allows the user to control and monitor the GPIOs associated with P440s. (Note that this support is for P440s only.)

1.1 Usage Notes: Interface and Ranging

This section provides a short overview of key facts relative to P4xx behavior and interfaces.

The P400s have both an Ethernet and USB interface. The P410s support both USB and Serial UART interfaces. The P440s support Ethernet, USB, Serial, CAN and SPI.

Interfacing with USB is trivial. One just connects the Host to the P4xx with a Micro-USB cable and connection will be established. Connecting via Serial is similarly easy; however the location of the jack varies depending on board types. For details, read the document *320-0287 Using the USB and Serial Interfaces*.

Connection with CAN or SPI is described in the *320-0317 P440 Data Sheet*.

Interface with Ethernet is more involved because one must assure that the Host PC's TCP/IP and Subnet mask are configured properly. Depending on how the P4xx was purchased, the unit might be configured for DHCP. For details on connecting with Ethernet, refer to the *320-0320 RangeNet User Guide*, **Appendix A: Connecting with Ethernet** and **Section 3.1 - Connecting with Ethernet**.

Upon power-up, with or without a Host connection, the P4xx will operate in accordance with the last saved configuration. If the last saved configuration was Ranging Mode, then the P4xx will automatically respond to a "range request" packet with its Node ID as the range target. If the last saved configuration was either of the Networking Modes, then it will still respond to a range requests but it will also initiate range requests and/or send data packets as defined by the RangeNet configuration settings. The factory default is Ranging Mode.

All P4xx devices operating in Ranging Mode are slaved to their respective Hosts. There is no mechanism built into the P4xx to coordinate transmissions other than automatic response to targeted range requests. In Network Mode, the P4xx controls transmissions based on parameters set by the Host through the API.

The P4xx measures distance through a TW-TOF radio frequency (RF) ranging technique. If the antennas are altered or extra SMA cables are introduced, then the range measurement will exhibit a bias based on the longer TOF of the RF pulse. This bias can be calibrated and removed by the radio firmware using the Calibration Tab of the RangeNet GUI. The calibration procedure can be found in the *320-0320 RangeNet User Guide*.

1.2 RangeNet

This section discusses RangeNet terms, operation and network modes.

1.2.1 RangeNet Terms

Definitions of the terms used by RangeNet.

- RangeNet Node
  - A P4xx operating in RangeNet Networking Mode.
- Target
  - The object of a RangeNet Node's range request.
- Neighbor
  - All RangeNet Nodes that the local RangeNet Node can hear are considered its neighbors.
- Neighbor Database (NDB)
  - A database which contains the Node IDs and associated range information of every node that it has heard in the recent past. The "age-out" period is configurable by the user.
- Beacons

    - RangeNet Nodes which will respond to range requests but which will not initiate a range request.   They will periodically transmit a beacon message at a rate configured by the user, if they have not recently responded, in order to inform neighbors of their presence.
- Anchor (also called a Reference)
    - These nodes are normally located at fixed points and are used in localization as reference points from which other units can compute their own locations. In Location Mode, an Anchor operates as a Beacon.
- Aging Out
    - If a node has not heard from a Neighbor in a preconfigured amount of time, then the Neighbor will be deleted from the node's Neighbor Database. ("Not heard from" implies the node has received neither a range response nor a beacon packet recently.)
- Slot Map
    - The Slot Map is a construct which defines (1) the exact time when units are allowed to initiate conversations, (2) which units they are communicating with, and (3) the defining parameters (PII, code channel, data bits, etc.) for those communications. The *320-0320 RangeNet User Guide* provides an excellent visual description of the Slot Map.

1.2.2 RangeNet Overview

RangeNet is a mode of operation in which multiple P4xx devices form a simple ranging network. P4xx units operating in RangeNet Mode are known as RangeNet Nodes. RangeNet Nodes automatically discover and range to each other and report ranges, user data, and other information.

If connected to a Host PC or microcontroller, then this information is sent to the Host. The user can also define a default communication port over which the information will be reported when the Host is not connected. This is useful on boot-up of the Host or when the system is recovering from a power failure (the user can also select "none" as the default).   In any event, once the Host has successfully connected with the P4xx then all information will be reported over the communications port through which the Host connected. For example, if the User set the default communications port to serial and then connected with a Host through USB, then the P4xx would initially send information through the serial connection and then switch to USB when the connection was made.

RangeNet over-the-air range request and response packets have a slightly different internal data format from standard Ranging range and response packets. P4xx units in the standard Ranging Mode will not respond to RangeNet range requests. However, RangeNet Nodes will respond to standard Ranging requests, and both modes can make use of Coarse Range Estimation and Filtered Range Estimates (the combination of Precision and Coarse) from any received over-the-air packet.

RangeNet Nodes announce their presence by sending RangeNet requests or by transmitting data packets (also known as Beacon packets). Targets of the range requests are chosen from the Neighbor Database. In ALOHA, targets will be identified and ranged to on a round-robin basis. In TDMA, units will be ranged to as defined by the Slot Map.   If no neighbor Node IDs are known, as when first starting up or if this RangeNet node has been out of contact with the rest of the network for a while, a short beacon packet is transmitted. Otherwise, specific neighbor Node IDs are chosen as targets by the Target Prioritizer.

Both protocols support the discovery of new nodes and the exit of existing nodes.

1.2.3 RangeNet Network Modes

ALOHA Network Protocol: In this concept of operation, network nodes do not own any particular time slot and transmit range requests at random times between user-configurable minimum and maximum intervals. Setting minimum and maximum times is described in **Appendix C**. RangeNet also supports Automatic Congestion Control (ACC). When this mode is selected, the minimum and maximum intervals will be overridden and the ACC algorithm will operate the network as fast as possible. Furthermore, RangeNet will throttle this rate as the number of nodes in the network increase or decrease. For more information on how the network can be best operated manually or automatically, see the white paper entitled "RangeNet/ALOHA Guide to Optimal Performance."

All modes support a Virtual Carrier Sense feature, which prevents sending a range request when it is known that another P4xx (whether in standard Ranging or RangeNet Mode) is responding to a range request. This avoids knowingly interfering with an ongoing range conversation.

The ALOHA protocol as implemented in RangeNet has an additional benefit in that the network will behave as a Slotted ALOHA system. This is advantageous because the capacity of a Slotted ALOHA system is twice the capacity of a Pure ALOHA system (36% vs. 18%).

TDMA (Time Division Multiple Access) Network Protocol: In this concept of operation, the network nodes do own a particular time slot. The time at which they can transmit, the radios they communicate with, the duration of the transmission, the order in which they transmit, and the parameters associated with the range conversation or data transmission are all predefined by the user. This definition is captured as a Slot Map.

The transmitter in the first slot (Slot #0) maintains the master clock. All units in the system will synchronize to this transmitter. If this transmitter should lose power or otherwise exit the system, then once the departure has been recognized, all of the units in the system will stop transmitting and the system will halt. The amount of time between actual departure and recognition of departure is normally on the order of seconds. This time is actually defined by the user as "Max Neighbor Age" and the default value for this is 10 seconds. The user can select different values but should take care not to make the value "too large." During the time between exit and Max Neighbor Age, the network will continue to function for as long as the clocks in the radios are reasonably synchronized. Because each unit has a high precision oscillator, the system will operate for at least 10 and maybe many more seconds. For example, setting the Max Neighbor Age to a very high value, like 10,000 seconds, is asking for trouble. With such a setting, the network will sink into chaos when the oscillators finally lose synchronization.

Different units may have different Slot Maps but all units need to have cooperative and consistent Slot Maps. In other words, all slots in all Slot Maps must have equal duration and be arranged so there no conflict. The user not only has the ability to define a Slot Map, but it is also the user's responsibility to ensure that the Slot Map is organized such that there is no possibility for conflicting messages. For example, the user could define that in time slot 2, Units 1 and 2 could communicate on code channel 1 and Units 3 and 4 could communicate on code channel 2. However, having all 4 units communicate in the same slot, or requiring one unit to range to more than one unit, or other such nonsense is incompatible with a well-functioning network and will cause conflict.

Once the Slot Map has been defined, there is no automatic mechanism for allowing new units to be added to the system. However, a higher level protocol could allow the Host to monitor the network and redefine the Slot Map on the fly such that new units could be added. However, this Host-based code would be the responsibility of the user to develop.

Where ALOHA handles conflict automatically based on infrequent and random transmissions as well as retransmissions, TDMA systems require that the user predefine an operating order such that conflict is avoided. While ALOHA is easier to operate (especially in complex environments) it is less efficient and has lower throughput. A TDMA network operating on one channel has more than twice the capacity of an ALOHA system.

1.2.4 Localization Modes

Localization is resident in the P4xx and is implemented as an additional layer to the RangeNet network and is supported in either the ALOHA or TDMA network protocols. It is capable of handling systems consisting of up to 60 nodes.

The Location Engine combines the following elements to compute the P4xx's location:

- Information on the location of the Anchors
- Range measurements from the P4xx to the Anchors
- A Kalman Filter
- A motion model
- A calculation of the Geometric Dilution of Precision (GDOP)
- Various other tuning parameters

Locations are reported in the X, Y, and Z dimensions. Localization can be operated either in 2D (in which case Z is assumed to be fixed at a user selectable height) or in 3D. In addition, the Location Engine also reports the Variance and Covariance values associated with the range calculation.

The Location layer takes advantage of the UWB network to transfer configuration and location information throughout the network. Because of this, the user can connect to any node and monitor or control every unit in the system.

Configuration information is largely captured in the Location Map, or LMAP. This construct contains:

- A list of all nodes active in the system and whether they are Anchors or Mobiles
- The rate at which nodes will beacon or range
- Instructions as to whether or not they will transmit ELR (Echo Last Range) and ELL (Echo Last Location) information as data when they issue range requests
- Location (in X, Y, Z dimensions) of all Anchors

The Location Engine will drive the network, so there is little need for the user to deal with the various network mode commands. The Location Engine will operate using the communications channel, antenna definition, transmit power, and Pulse Integration Index (PII) defined through the RCM_SET_CONFIG_REQUEST message.

Since the Location Engine is resident in the P4xx, the Host computer's processing load is significantly reduced. In fact, the role of the Host processor is largely limited to:

- Providing configuration information
- Commanding the system to transition from one mode to another

- Serving as a port to which the connected P4xx will provide a torrent of location information for all of the units in the system

Be advised that it is surprisingly easy to underestimate how much data can be sent by the P4xx. For example, if the user requests that all location, ranging, and scan data be reported, then the P4xx can easily overwhelm the capacity of the Host computer to handle the volume of messages that are sent.

The nodes behave in the following manner.

Anchor P4xxs are fixed in location and have only two functions:

- Anchors occasionally send Beacon messages to announce their presence. This behavior acts as a seed to start communications between units in the ALOHA network.
- Anchors can, if so directed by the user, report any range or location information which it may have received from a Mobile to a Host computer.

Mobile P4xx units, as the name suggests, do not have a fixed location but rather move either continuously or intermittently. They have four functions:

- Mobiles will issue range requests to any Anchor whose Beacon message they might hear.
- During initialization, once a Mobile has ranged to a sufficient number of Beacons, it will compute its location using the Non-linear Least Squares (NLS) or Geometric solver. Once the Mobile's position is initialized with the NLS solver, the Mobile will, at the user's option, either continue using the Geometric solver or use the Kalman solver. Positions are reported after each subsequent successful range measurement.
- As part of each range request message a Mobile transmits it will, if so directed by the user, also transmit as data their last calculated location.
- Mobiles can, if so directed by the user, report any range and/or location information which it may have received to a Host computer.

Localization has three operational modes: Idle, Autosurvey, and Tracking.

While in the Idle Mode, the X, Y, and Z locations of the Anchors and the Z location of the Mobiles (as well as other configuration information) can be broadcast to the other units in the network. This broadcast uses a flooding algorithm to distribute the information. The algorithm incorporates acknowledgements so it is easy to confirm that all units in the system have received the messages.

Once this information has been distributed, the system can be transitioned to Tracking Mode. In this mode, the location of Mobiles will be calculated and that information distributed through the network as configured by the user.

Autosurvey is an optional mode. When engaged, it will direct the Anchors to range to each other while the Mobiles stay idle. The resultant information will flow back to the Host. It will then be the responsibility of the Host to decide when enough information has been collected and to load this information into the LMAP.

TDSR's RangeNet GUI is an excellent tool for illustrating operation of ranging, networking and location. Exercising the GUI will provide the user with insight into how the API commands are used and how the system works. For that reason we strongly recommend that the user begin any development program by first mastering the RangeNet GUI. Mastery can be accomplished in a few days, perhaps weeks, and will greatly reduce the time understanding how the API works.

If properly set up and operated in a benign environment, the system will provide location accuracy of the Mobiles on the order of a centimeter. In some cases, accuracies of 1 or 2 millimeters have been achieved. Guidance on how to achieve the best accuracy possible is provided in **Appendix G: Maximizing Location Performance** of the *320-0320 RangeNet User Guide*.

1.3 Ranging, RangeNet, and Localization Configuration Control

There are three types of configuration data: (1) information relating to individual transmissions (such as integration rate, transmit power, communications channel, antenna selection, etc.) (2) information relating to the network operation (such as polling frequency, node status, etc.) and (3) information relating to localization. The first type of configuration data is set with the RCM configuration command (RCM_SET_CONFIG_REQUEST), the second with the RangeNet configuration command (RN_SET_CONFIG_REQUEST) and the third with Location configuration command LOC_SET_CONFIG_REQUEST) Configurations can be changed dynamically without having to stop or reboot the radio.

Setting either the Ranging or RangeNet parameters will result in RangeNet resetting statistics and the Neighbor Database.

1.4 Sending and Receiving Data

Data can be transmitted as a simple data packet in any mode. Such data is sent using the RCM_SEND_DATA_REQUEST. Data received will be sent from the P4xx to the Host using the RCM_DATA_INFO packet.

However, data can also be sent as part of a range request or response. In Ranging Mode, the user has two options. Data can be sent as part of the range request using the RCM_SEND_RANGE_REQUEST command or the Host can transfer a specific message whenever it responds to a range request. The latter is accomplished by loading the response buffer using the RCM_SET_RESPONSE_DATA_REQUEST command.

When operating in RangeNet Mode, the P4xx is responsible for scheduling when and to which unit a range request will be sent. The Host always has the option of transmitting RCM data packets, but the Host can also preload the transmit and receive buffers with data using the RN_SET_REQUEST_DATA and RN_SET_RESPONSE_DATA commands. This enables the Host to send small amounts of data automatically.

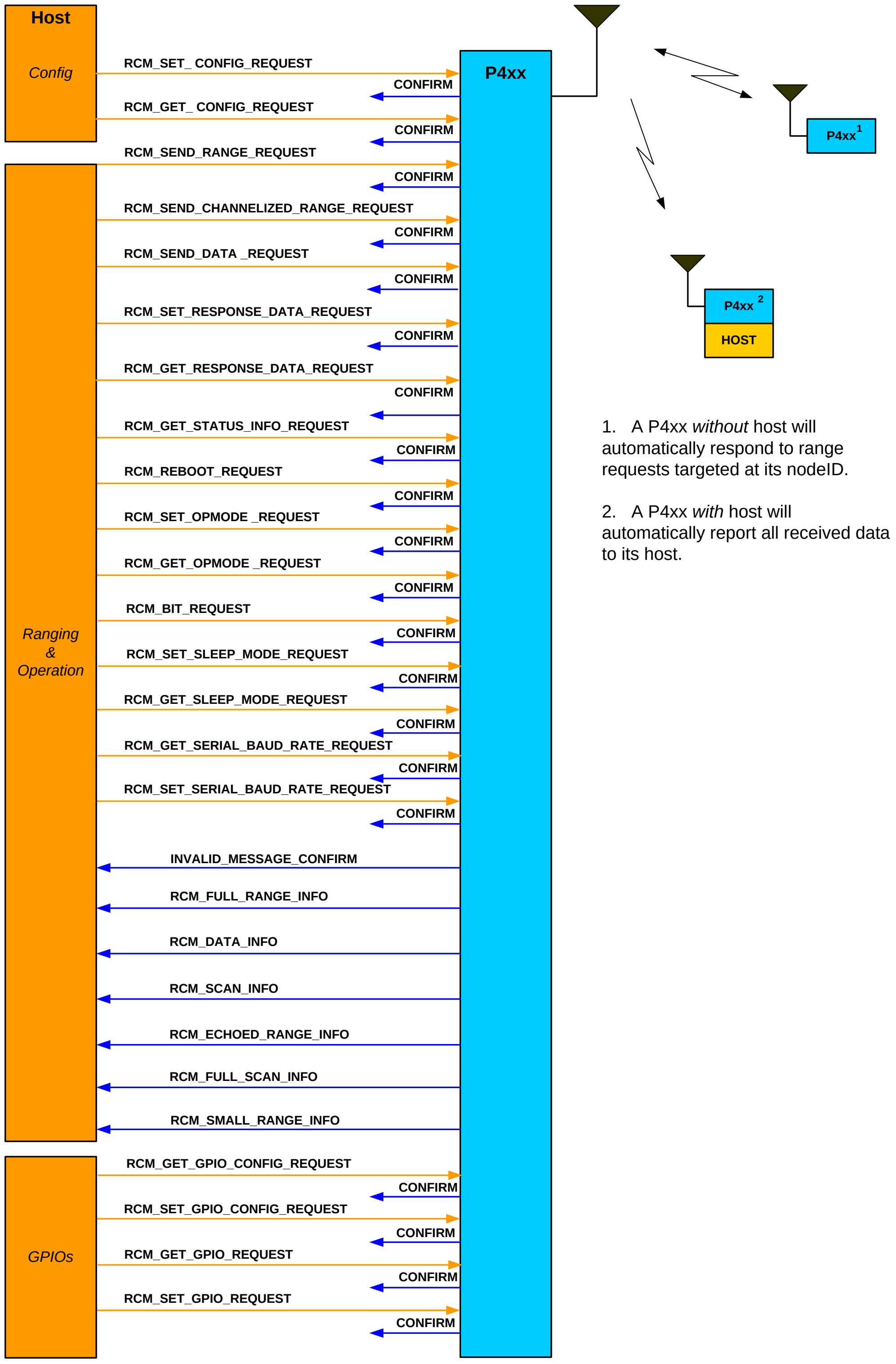
2 The Ranging, RangeNet, and Localization Interface

This is a high-level description of the data passed between a Host processor and the P4xx.

Figures 2-A, 2-B, and 2-C illustrate a system of three P4xx platforms, two with Hosts and one without a Host. A high-level data flow interface is graphically depicted between one of the Hosts and its co-located P4xx. Ranging Mode messages have the prefix RCM and are shown in **Figure 2-A**. RangeNet (also called Networking) Mode messages have the prefix RN and are shown in **Figure 2-B**. Location Mode messages have the prefix LOC and are shown in **Figure 2-C**.

The messages relating to control and monitoring of the GPIOs are included with the RCM commands. For an extensive discussion of the GPIOs, please see Section 5.6 of the document *320-0320 RangeNet User Guide.*

In Ranging Mode, the HOST<->P4xx interface consists of twenty REQUEST messages from Host to P4xx with their associated CONFIRM messages. In addition, there are seven INFO messages that are sent to the Host upon receiving UWB packets from other P4xxs.



1. A P4xx *without* host will automatically respond to range requests targeted at its nodeID.

2. A P4xx *with* host will automatically report all received data to its host.

Fig. 2-A: Ranging Mode RCM Message flow - Host to P4xx messages and P4xx to P4xx RF packets

In RangeNet (Networking) Mode, the HOST<->P4xx interface consists of all of the Ranging Mode commands plus an additional twenty networking-specific REQUEST messages from Host to P4xx with their associated CONFIRM messages as well as two INFO message that are sent to the Host.

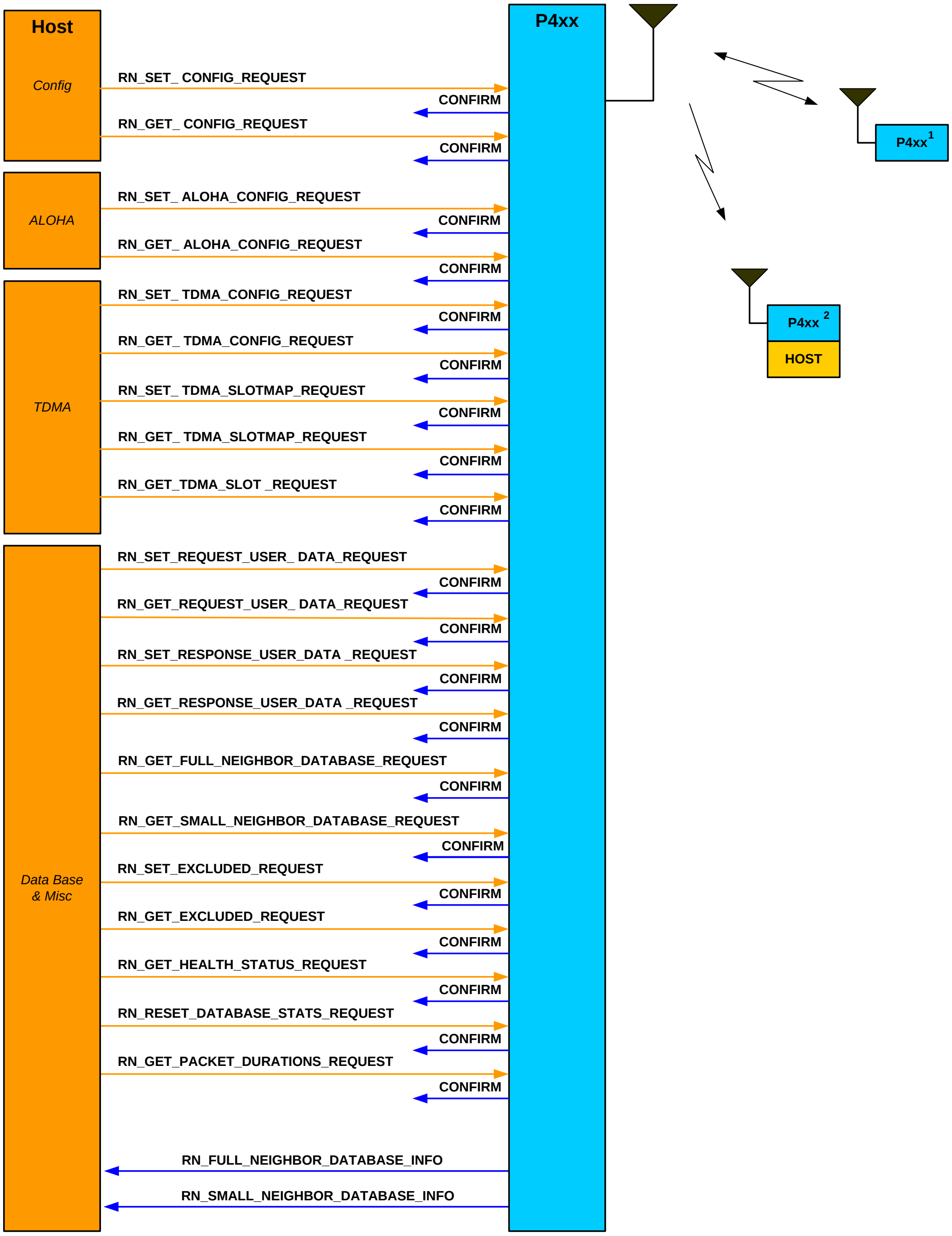

Host
Config
RN_SET_ CONFIG_REQUEST
CONFIRM
RN_GET_ CONFIG_REQUEST
CONFIRM
ALOHA
RN_SET_ ALOHA_CONFIG_REQUEST
CONFIRM
RN_GET_ ALOHA_CONFIG_REQUEST
CONFIRM
TDMA
RN_SET_ TDMA_CONFIG_REQUEST
CONFIRM
RN_GET_ TDMA_CONFIG_REQUEST
CONFIRM
RN_SET_ TDMA_SLOTMAP_REQUEST
CONFIRM
RN_GET_ TDMA_SLOTMAP_REQUEST
CONFIRM
RN_GET_TDMA_SLOT _REQUEST
CONFIRM
Data Base & Misc
RN_SET_REQUEST_USER_ DATA_REQUEST
CONFIRM
RN_GET_REQUEST_USER_ DATA_REQUEST
CONFIRM
RN_SET_RESPONSE_USER_DATA _REQUEST
CONFIRM
RN_GET_RESPONSE_USER_DATA _REQUEST
CONFIRM
RN_GET_FULL_NEIGHBOR_DATABASE_REQUEST
CONFIRM
RN_GET_SMALL_NEIGHBOR_DATABASE_REQUEST
CONFIRM
RN_SET_EXCLUDED_REQUEST
CONFIRM
RN_GET_EXCLUDED_REQUEST
CONFIRM
RN_GET_HEALTH_STATUS_REQUEST
CONFIRM
RN_RESET_DATABASE_STATS_REQUEST
CONFIRM
RN_GET_PACKET_DURATIONS_REQUEST
CONFIRM
RN_FULL_NEIGHBOR_DATABASE_INFO
RN_SMALL_NEIGHBOR_DATABASE_INFO
P4xx
P4xx 1
P4xx 2
HOST

Fig. 2-B: RangeNet (Networking) RN Message flow - Host to P4xx messages and P4xx to P4xxRF packets

In Location Mode, the HOST<->P4xx interface consists of all of the Ranging and RangeNet Mode commands plus an additional six specific REQUEST messages from Host to P4xx with their associated CONFIRM messages as well as six INFO messages that are sent to the Host.

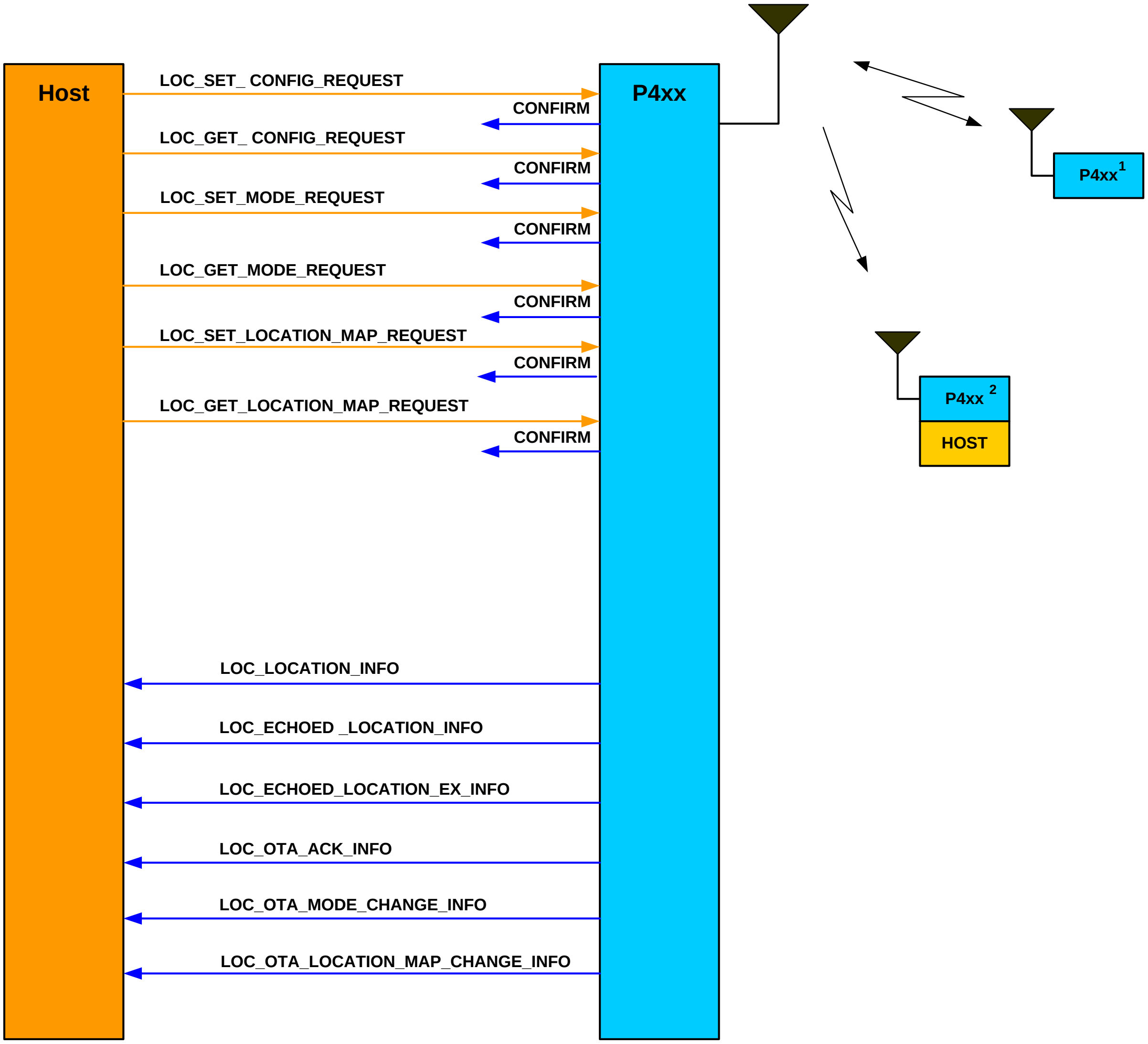


Fig. 2-C: Localization LOC Message flow - Host to P4xx messages and P4xx to P4xxRF packets

All P4xx devices, including those without Hosts, respond automatically to range request packets that are targeted at them. They will append user data stored in their RESPONSE DATA buffer. P4xx units with connected Hosts will automatically send data stored in their REQUEST DATA buffer and (optionally) provide scan information to the Host in separate UDP or USB packets. The format of these messages is described in this document.

The REQUEST and INFO messages are described in the next subsection. **Appendix A** contains an illustration of the data flow for a complete range/data conversation. **Appendix B** contains extra descriptions of many of the Ranging Mode parameters. **Appendix C** contains extra descriptions of several of the RangeNet Mode parameters.

3 RCM API Messages

3.1 RCM_SET_CONFIG_REQUEST (0x0001)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SET_CONFIG_CONFIRM (Radio)

Purpose: This message configures the basic parameters in the P4xx, thereby defining radio operation.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_CONFIG_REQUEST (0x0001)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Node ID	UINT32	UWB ID of this radio (used for UWB range targeting.) Valid values are 1 - 2^32-2. By default this value will be 100 - 10X. For P400's, this matches the last byte of the IP address. Modifying Node ID of a P400 does NOT change the IP address. P410s are initially set to Node ID 100 except when they are shipped as part of a kit, in which case they are numbered sequentially from 100. P412 Node IDs are set to match the board serial number. Avoid using IDs 0 and 2^32-1. They are reserved for network and broadcast functions.
3	Pulse Integration Index (PII)	UINT16	Specifies an index, which configures the number of pulses per data symbol. Valid values are [4-9]. The default is 7 meaning 2^7 = 128 pulses per symbol.
4	Antenna Mode	UINT8	Specifies the default antenna for transmission and reception. Valid values are [0 = A, 1 = B, 2 = TXA/RXB, 3 = TXB/RXA]. In addition, setting the high order bit (0x80) of this byte enables automatic toggling of the antenna after each response by this radio. The default value is 0x00.

#	Parameter	Type	Definition
5	Code Channel	UINT8	Specifies the active UWB channel. Multiple ranging conversations can occur simultaneously if multiple code channels are used. Both the requester and responder radios must be configured to the same code channel for successful communication. Possible values are [0-10]. The default value is 0.
6	Antenna Delay A	INT32	Specifies the approximate time delay, in picoseconds, of the SMA cable connecting antenna A. The default value of 0 ps corresponds to a Broadspec antenna connected via 90º SMA elbow.
7	Antenna Delay B	INT32	Specifies the approximate time delay, in picoseconds, of the SMA cable connecting antenna B. The default value of 0 ps corresponds to a Broadspec antenna connected via 90º SMA elbow.
8	Flags	UINT16	Bits 0-1 are the Send Scan bits. If enabled, received UWB waveform scans are sent to the host (e.g. upon UWB reception of range a range request or response). Set to 01b (bit 0 set) to enable sending SCAN_INFOs. Set to 10b (bit 1 set) to enable sending FULL_SCAN_INFOs. Setting 11b (both bits set) is reserved for a future, enhanced scan message. Default is 00b (no scans are sent to the host). Bit 2 is the "FAN_OFF" bit. It disables power to the fan. [0=on, 1=off]. By default this bit is 0 (fan is ON). Bit 3 is the "SEND_SCAN_WITH_DATA" bit. It enables sending SCAN_INFO with each data-only (not range) packet. Default is 0 (do not send scan with data-only packets). Bit 4 is the "DISABLE_CRE_RANGE_MSGS" bit. It disables sending promiscuous RANGE_INFO packets with Coarse Range Estimated values with each packet received. Default is 0 (send RANGE_INFO messages with each packet received). Bit 5 is the "Both Sides Delta Check" flag. When set, additional information is sent in the range response UWB packet that allows the requester to flag possibly slightly inaccurate ranges. This setting is highly situational and is intended for use only in specific RF environments to attain maximum accuracy and precision. To be enabled, it must be enabled on both requester and responder. Default is 0 (disabled). Bit 6 is the "Flag LOS/NLOS Mismatch" flag. When set, ranges where one side's LED reported non-

#	Parameter	Type	Definition
			line-of-sight and the other side reported line-of-sight are flagged with an error status code (11, RCM_RANGE_STATUS_LOS_NLOS_MISMATCH) Bit 7 is the “ENABLE_ECHO_LAST_RANGE” bit. When enabled, it sends a message containing the ranges it hears from other nodes’ range measurements. It also enables the sending of its own ranges in range requests. [0=disabled, 1=enabled]. Bit 8 is the send “RCM_SMALL_RANGE_INFO” flag. When set, the much smaller SMALL_RANGE_INFO message is sent instead of the normal, RCM_FULL_RANGE_INFO message. (Default is 0, send FULL_RANGE_INFO messages.) Bits 10-11 are the Antenna Orientation flags. Bit 10 is for Antenna A, bit 11 is for Antenna B. Set the appropriate bit if the antenna is upside down compared to the standard orientation for your application. This will allow the radio to handle the inverted waveform produced by mismatched antenna orientations, providing slightly more accurate ranges. Bit 12 is reserved. Bit 13 is the Disable FFT Brickwall Filter flag and applies to P440 radios only. Default is 0 (leave FFT Brickwall Filter enabled).
9	Transmit Gain	UINT8	Specifies the active UWB transmit power from 0 (lowest) to 63 (highest). The actual transmit power depends on the radio model. See the model's data sheet for additional information.

#	Parameter	Type	Definition
10	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.

3.2 RCM_SET_CONFIG_CONFIRM (0x0101)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SET_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RCM_SET_CONFIG_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of the RCM_SET_CONFIG_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_CONFIG_CONFIRM (0x0101)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.3 RCM_GET_CONFIG_REQUEST (0x0002)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_GET_CONFIG_CONFIRM (Radio)

Purpose: This is a request message sent by the Host to P4xx for the current radio configuration.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_CONFIG_REQUEST (0x0002)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

3.4 RCM_GET_CONFIG_CONFIRM (0x0102)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_GET_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx in response to a RCM_GET_CONFIG_REQUEST from the Host. It provides the current P4xx configuration information.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_CONFIG_CONFIRM (0x0102)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Node ID	UINT32	UWB ID of this radio (used for UWB range targeting.) Valid values are 1 - 2^32-2. By default this value will be 100 - 10X, matching the last byte of the IP address. Modifying Node ID does NOT change the IP address.
3	Pulse Integration Index	UINT16	Specifies an index, which configures the number of pulses per data symbol. Valid values are [4-9]. The default is 7 meaning 2^7 = 128 pulses per symbol.
4	Antenna Mode	UINT8	Specifies the default antenna for transmission and reception. Valid values are [0=A, 1=B, 2=TXA,RXB, 3=TXB,RXA]. In addition, setting the high order bit (0x80) of this byte enables automatic toggling of the antenna after each response. The default value is 0.
5	Code Channel	UINT8	Specifies the active UWB channel. Multiple ranging conversations can occur simultaneously if multiple code channels are used. Both the requester and responder radios must be configured to the same code channel for successful communication. Possible values are [0-10]. The default value is 0.
6	Antenna Delay A	INT32	Specifies the approximate time delay, in picoseconds, of the SMA cable connecting antenna A. The default is that of the default Broadspec antenna connected via 90° SMA elbow.

#	Parameter	Type	Definition
7	Antenna Delay B	INT32	Specifies the approximate time delay, in picoseconds, of the SMA cable connecting antenna B. The default is that of the default Broadspec antenna connected via 90º SMA elbow.
8	Flags	UINT16	Bits 0-1 are the Send Scan bits. If enabled, received UWB waveform scans are sent to the host (e.g. upon UWB reception of range a range request or response). Set to 01b (bit 0 set) to enable sending SCAN_INFOs. Set to 10b (bit 1 set) to enable sending FULL_SCAN_INFOs. Setting 11b (both bits set) is reserved for a future, enhanced scan message. Default is 00b (no scans are sent to the host). Bit 2 is the "FAN_OFF" bit. It disables power to the fan. [0=on, 1=off]. By default this bit is 0 (fan is ON). Bit 3 is the "SEND_SCAN_WITH_DATA" bit. It enables sending SCAN_INFO with each data-only (not range) packet. Default is 0 (do not send scan with data-only packets). Bit 4 is the "DISABLE_CRE_RANGE_MSGS" bit. It disables sending promiscuous RANGE_INFO packets with Coarse Range Estimated values with each packet received. Default is 0 (send RANGE_INFO messages with each packet received). Bit 5 is the "Both Sides Delta Check" flag. When set, additional information is sent in the range response UWB packet that allows the requester to flag possibly slightly inaccurate ranges. This setting is highly situational and is intended for use only in specific RF environments to attain maximum accuracy and precision. To be enabled, it must be enabled on both requester and responder. Default is 0 (disabled). Bit 6 is the "Flag LOS/NLOS Mismatch" flag. When set, ranges where one side's LED reported non-line-of-sight and the other side reported line-of-sight are flagged with an error status code (11, RCM_RANGE_STATUS_LOS_NLOS_MISMATCH). Bit 7 is the "ENABLE_ECHO_LAST_RANGE" bit. When enabled, it sends a message containing the ranges it hears from other nodes' range measurements. It also enables the sending of its own ranges in range requests. [0=disabled, 1=enabled]. Bit 8 is the send "RCM_SMALL_RANGE_INFO"

#	Parameter	Type	Definition
			flag. When set, the much smaller SMALL_RANGE_INFO message is sent instead of the normal, RCM_FULL_RANGE_INFO message. (Default is 0, send FULL_RANGE_INFO messages.) Bits 10-11 are the Antenna Orientation flags. Bit 10 is for Antenna A, bit 11 is for Antenna B. Set the appropriate bit if the antenna is upside down compared to the standard orientation for your application. This will allow the radio to handle the inverted waveform produced by mismatched antenna orientations, providing slightly more accurate ranges. Bit 12 is reserved. Bit 13 is the Disable FFT Brickwall Filter flag. P440 radios only. Default is 0 (leave FFT Brickwall Filter enabled).
9	Transmit Gain	UINT8	Specifies the active UWB transmit power from 0 (lowest) to 63 (highest). The actual transmit power depends on the radio model. See the model's data sheet for additional information.
10	Unused	UINT8	Reserved
11	Timestamp	UINT32	Milliseconds since P4xx power-up. Note this implies a rollover approximately every 50 days.
12	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.5 RCM_SEND_RANGE_REQUEST (0x0003)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SEND_RANGE_CONFIRM (Radio)

Purpose: This message commands the P4xx to send a UWB range request packet (with optional data) to a targeted RCM node. Note the (optional) data sent in this packet is typically received by ALL RCM nodes within range, not just the targeted node. All P4xxs that receive this data will promiscuously send this data to their respective Host. The targeted node will, however, be the only node that responds with a range response.

Warning: The user should avoid sending RCM range requests when operating in Network or Location mode as these messages will conflict/interfere with proper operation of the Network.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SEND_RANGE_REQUEST (0x0003)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Responder ID	UINT32	Node ID of the range request target. A value of 2^32-1 (4294967295) indicates broadcast.
3	Antenna Mode	UINT8	Specifies the active antenna for transmission and reception. Valid values are [0 = A, 1 = B, 2 = TXA/RXB, 3 = TXB/RXA]. The default value is 0 (use antenna A for both transmissions and receptions.)
4	Reserved	UINT8	Reserved
5	Data Size	UINT16	Number of bytes to include in the range request packet. These bytes follow. Maximum = 1000. **Note:** the P4xx transmits 32bit (4byte) words. Any partial words will be zero-filled over the air but these bits will be removed upon reception. **Free Data:** Due to the waveform scan requiring a minimum number of pulses, a small amount of user data may be added without any packet time penalty. For RCM range requests, up to 20 bytes of user data may be added without increasing packet TX time (or 3 bytes if ELR is enabled).

#	Parameter	Type	Definition
6	Data	N*UINT8	Data to be sent with the range request packet. Max Number of bytes is 1000.

3.6 RCM_SEND_RANGE_REQUEST_CONFIRM (0x0103)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SEND_RANGE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RCM_SEND_RANGE_REQUEST command. This response confirms the UWB range request packet was successfully sent by the P4xx.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SEND_RANGE_REQUEST_CONFIRM (0x0103)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.7 RCM_SEND_CHANNELIZED_RANGE_REQUEST (0x0006)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SEND_CHANNELIZED_RANGE_CONFIRM (Radio)

Purpose: This message commands the P4xx to send a UWB range request packet (with optional data) to a targeted RCM node on a specific channel that can be different than the channel specified in the RCM_SET_CONFIGURATION_REQUEST message. Note the (optional) data sent in this packet is typically received by ALL RCM nodes within range, not just the targeted node. All P4xxs that receive this data will promiscuously send this data to their respective Host. The targeted node will, however, be the only node that responds with a range response.**Warning:** The user should avoid sending RCM range requests when operating in Network or Location mode as these messages will conflict/interfere with proper operation of the Network.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SEND_CHANNELIZED_RANGE_REQUEST (0x0006)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Responder ID	UINT32	Node ID of the range request target. A value of 2^32-1 (4294967295) indicates broadcast.
3	Antenna Mode	UINT8	Specifies the active antenna for transmission and reception. Valid values are [0 = A, 1 = B, 2 = TXA/RXB, 3 = TXB/RXA]. The default value is 0 (use antenna A for both transmissions and receptions.)
4	Code Channel	UINT8	Possible values are 0-10. Default is 0.

#	Parameter	Type	Definition
5	Data Size	UINT16	Number of bytes to include in the range request packet. These bytes follow. Maximum = 1000. **Note:** the P4xx transmits 32bit (4byte) words. Any partial words will be zero-filled over the air but these bits will be removed upon reception. **Free Data:** Due to the waveform scan requiring a minimum number of pulses, a small amount of user data may be added without any packet time penalty. For RCM range requests, up to 20 bytes of user data may be added without increasing packet TX time (or 3 bytes if ELR is enabled).
6	Data	N*UINT8	Data to be sent with the range request packet. Max Number of bytes is 1000.

3.8 RCM_SEND_CHANNELIZED_RANGE_REQUEST_CONFIRM (0x0106)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SEND_CHANNELIZED_RANGE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the HOST in response to a RCM_SEND_CHANNELIZED_RANGE_REQUEST command. This response confirms the UWB channelized range request packet was successfully sent by the P4xx.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SEND_CHANNELIZED_RANGE_REQUEST_CONFIRM (0x0106)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.9 RCM_SEND_DATA_REQUEST (0x0004)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SEND_DATA_CONFIRM (Radio)

Purpose: This message commands the P4xx to send a UWB data-only packet (without range request). All P4xxs within range will receive this data and promiscuously send this data to their respective Host (if attached). No automatic acknowledgement is provided for data.

Warning: The user should avoid sending RCM data requests when operating in Network or Location mode as these messages will conflict/interfere with proper operation of the Network.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SEND_DATA_REQUEST (0x0004)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Antenna Mode	UINT8	Specifies the active antenna for transmission and reception. Valid values are [0=A, 1=B, 2=TXA,RXB, 3=TXB,RXA]. The default value is 0 (use antenna A for both transmissions and receptions.)
3	Reserved	UINT8	Not used
4	Data Size	UINT16	Number of bytes to include in the data-only packet. These bytes follow. Maximum = 1000. **Note**: the P4xx transmits 32bit (4byte) words. Any partial words will be zero-filled over the air but these bits will be removed upon reception
5	Data	N*UINT8	Data to be sent with the range request packet. Max Number of bytes is 1000.

3.10 RCM_SEND_DATA_CONFIRM (0x0104)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SEND_DATA_REQUEST (Host)

Purpose: This message is sent from the P4xx to the Host in immediate response to a RCM_SEND_DATA_REQUEST command. This response confirms the data-only packet was successfully sent by the P4xx.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SEND_DATA_CONFIRM (0x0104)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.11 RCM_SET_RESPONSE_DATA_REQUEST (0x0005)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SET_RESPONSE_DATA_CONFIRM (Radio)

Purpose: This message allows the Host to set the data buffer in the RCM range response packet. This data will be transmitted by the P4xx whenever it sends a range response packet. This data buffer will remain in effect until changed by the Host. Upon boot this buffer will be empty.

Warning: The user should avoid sending RCM data requests when operating in Network or Location mode as these messages will conflict/interfere with proper operation of the Network.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_RESPONSE_DATA_REQUEST (0x0005)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Reserved	UINT16	Reserved

#	Parameter	Type	Definition
3	Data Size	UINT16	Number of bytes to include in each range response packet. These actual bytes follow. Maximum = 1000. **Note:** the P4xx transmits 32bit (4byte) words. Any partial words will be zero-filled over the air but these bits will be removed upon reception. **Free Data:** Due to the waveform scan requiring a minimum number of pulses, a small amount of user data may be added without any packet time penalty. For RCM range responses, up to 4 bytes of user data may be added without increasing packet TX time.
4	Data	N*UINT8	Data to be sent with the range request packet. Extra bytes above <Data Size> will be ignored.

3.12 RCM_ SET_RESPONSE_DATA_CONFIRM (0x0105)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SET_RESPONSE_DATA_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_SET_RESPONSE_DATA_REQUEST command. This response confirms the buffer was successfully written.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_RESPONSE_DATA_CONFIRM (0x0105)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.13 RCM_GET_RESPONSE_DATA_REQUEST (0x0007)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_GET_RESPONSE_DATA_CONFIRM (Radio)

Purpose: This message allows the Host to read the data buffer in the RCM range response packet. This data will be transmitted by the P4xx whenever it sends a range response packet. This data buffer will remain in effect until changed by the Host. Upon boot this buffer will be empty.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_RESPONSE_DATA_REQUEST (0x0107)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

3.14 RCM_ GET_RESPONSE_DATA_CONFIRM (0x0107)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_GET_RESPONSE_DATA_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_GET_RESPONSE_DATA_REQUEST command. This response contains the RCM response data buffer.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_RESPONSE_DATA_CONFIRM (0x0107)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Reserved	UINT16	Reserved

#	Parameter	Type	Definition
3	Data Size	UINT16	Number of bytes to include in each range response packet. These actual bytes follow. Maximum = 1000. **Note:** the P4xx transmits 32bit (4byte) words. Any partial words will be zero-filled over the air but these bits will be removed upon reception. **Free Data:** Due to the waveform scan requiring a minimum number of pulses, a small amount of user data may be added without any packet time penalty. For RCM range responses, up to 4 bytes of user data may be added without increasing packet TX time.
4	Data	N*UINT8	Data to be sent with the range response packet.

3.15 RCM_GET_STATUS_INFO_REQUEST (0xF001)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_GET_STATUS_INFO_CONFIRM (Radio)

Purpose: This message prompts the P4xx to send the Host a data structure describing the hardware and software version numbers as well as other P4xx status information.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_STATUS_INFO_REQUEST (0xF001)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

3.16 RCM_GET_ STATUS_INFO_CONFIRM (0xF101)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_GET_STATUS_INFO_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_GET_VERSION_REQUEST command. This response provides a list of the hardware and software version numbers as well as other P4xx status information.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_STATUS_INFO_CONFIRM (0xF101)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	RCM Version Major	UINT8	RCM embedded major version number
3	RCM Version Minor	UINT8	RCM embedded minor version number
4	RCM Version Build	UINT16	RCM embedded build version number
5	UWB Kernel Major	UINT8	Kernel code major version number
6	UWB Kernel Minor	UINT8	Kernel code minor version number
7	UWB Kernel Build	UINT16	Kernel code build version number
8	FPGA Firmware Version	UINT8	Firmware version number represented in hexadecimal
9	FPGA Firmware Year	UINT8	Firmware year encoded. Use (year >> 4) * 10 + (year % 16) to get decimal value.
10	FPGA Firmware Month	UINT8	Firmware month encoded. Use (month >> 4) * 10 + (month % 16) to get decimal value.
11	FPGA Firmware Day	UINT8	Firmware day encoded. Use (day >> 4) * 10 + (day % 16) to get decimal value.
12	Serial Number	UINT32	Device serial number represented in Hexadecimal
13	Board Revision	UINT8	PCB revision – a single ASCII character
14	Power-On BIT Test Result	UINT8	Built-in Test Results, non-zero indicates BIT failure. See RCM_BIT_REQUEST and RCM_BIT_CONFIRM for more information.
15	Board Type	UINT8	Specifies the PCB type (or rather the radio type). 1=P400, 2=P410, 3=P412, 4=P440.
16	Pulser Configuration	UINT8	Pulser type on the radio. 0=FCC, 1=High Power, 2=EU.
17	Temperature	INT32	Board temp in 0.25^{o}C (divide this number by 4 to produce floating point ^{o}C).

#	Parameter	Type	Definition
18	Package Version	CHAR[32]	Embedded software package version number.
19	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.17 RCM_REBOOT_REQUEST (0xF002)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_REBOOT_CONFIRM (Radio)

Purpose: This message causes the P4xx to reboot, adopting configuration parameters saved to flash.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_REBOOT_REQUEST (0xF002)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

3.18 RCM_REBOOT_CONFIRM (0xF102)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_REBOOT_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_REBOOT_REQUEST command. Immediately after sending this message to the Host, the P4xx will reboot.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_REBOOT_CONFIRM (0xF102)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

3.19 RCM_SET_OPMODE_REQUEST (0xF003)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SET_OPMODE_CONFIRM (Radio)

Purpose: This message can be used to transition the P4xx between Ranging, RangeNet, and Location Modes. The new OpMode setting is not automatically saved to flash. After a reboot the radio will revert to its previous OpMode. However, saving the radio configuration to flash (e.g. through RCM_SET_CONFIG_REQUEST with the Persist flag set) will update the OpMode saved in flash, so the radio will be in the new OpMode after the next reboot.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_OPMODE_REQUEST (0xF003)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Operational Mode	UINT32	0: Ranging 4: RangeNet (Networking) 6: Location

3.20 RCM_ SET_OPMODE _CONFIRM (0xF103)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SET_OPMODE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RCM_SET_OPMODE_REQUEST command indicating the status of the request.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_OPMODE_CONFIRM (0xF103)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Operational Mode	UINT32	New Operational Mode
3	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.21 RCM_GET_OPMODE_REQUEST (0xF004)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_GET_OPMODE_CONFIRM (Radio)

Purpose: This message is used to request the P4xx Operating Mode.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_OPMODE_REQUEST (0xF004)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

3.22 RCM_ GET_OPMODE _CONFIRM (0xF104)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_GET_OPMODE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RCM_GET_OPMODE_REQUEST command indicating the status of the request. It contains the current operating mode.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_OPMODE_CONFIRM (0xF104)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Operational Mode	UINT32	0: RCM (Ranging) 4: RangeNet (Networking) 6: Location

3.23 RCM_BIT_REQUEST (0xF008)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_BIT_CONFIRM (Radio)

Purpose: This message prompts the P4xx to perform a BIT (Built-In-Test), returning results in the RCM_BIT_CONFIRM message.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_BIT_REQUEST (0xF008)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

3.24 RCM_ BIT_CONFIRM (0xF108)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_BIT_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RCM_BIT_REQUEST command. This response provides the status of the BIT (Built-In-Test).

Packet Definition:

#	Parameter	Type	Definition
0	RCM_BIT_CONFIRM (0xF108)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	BIT Status	UINT32	Return status of the Built-In-Test. Zero indicates no errors detected.

3.25 RCM_SET_SLEEP_MODE_REQUEST (0xF005)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SET_SLEEP_MODE_CONFIRM (Radio)

Purpose: This message causes the P4xx to transition to a low power mode.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_SLEEP_MODE_REQUEST (0xF005)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	Sleep Mode	UINT32	Specifies the transition state. 0: ACTIVE is required for transition out of any low-power state. 1: IDLE turns off UWB acquisition. Leaves all interfaces active. 2: ETHERNET powers down most components but leaves the Ethernet interface enabled. 3: SERIAL powers down most components including the Ethernet interface.

3.26 RCM_SET_SLEEP_MODE_CONFIRM (0xF105)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SET_SLEEP_MODE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_SLEEP_MODE_REQUEST command. This response verifies the P4xx received the request.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_SLEEP_MODE_CONFIRM (0xF105)	UINT16	Message type

#	Parameter	Type	Definition
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.27 RCM_GET_SLEEP_MODE_REQUEST (0xF006)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_GET_SLEEP_MODE_CONFIRM (Radio)

Purpose: This message queries the P4xx for the current sleep mode.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_SLEEP_MODE_REQUEST (0xF006)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.

3.28 RCM_GET_SLEEP_MODE_CONFIRM (0xF106)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_GET_SLEEP_MODE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_GET_SLEEP_MODE_REQUEST command. This response verifies the P4xx received the request and contains the current sleep mode.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_SLEEP_MODE_CONFIRM (0xF106)	UINT16	Message type

#	Parameter	Type	Definition
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	Sleep Mode	UINT32	Specifies the transition state. 0: ACTIVE is required for transition out of any low-power state. 1: IDLE turns off UWB acquisition. Leaves all interfaces active. 2: ETHERNET powers down most components but leaves the Ethernet interface enabled. 3: SERIAL powers down most components including the Ethernet interface.

3.29 RCM_GET_SERIAL_BAUD_RATE_REQUEST (0xF00A)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_GET_SERIAL_BAUD_RATE_CONFIRM (Radio)

Purpose: This message queries the P4xx for the current serial baud rate.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_SERIAL_BAUD_RATE_REQUEST (0xF00A)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.

3.30 RCM_GET_SERIAL_BAUD_RATE_CONFIRM (0xF10A)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_GET_SERIAL_BAUD_RATE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_GET_SERIAL_BAUD_RATE_REQUEST command. This response verifies the P4xx received the request and contains the current serial baud rate.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_SERIAL_BAUD_RATE _CONFIRM (0xF10A)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	Baud Rate	UINT32	Specifies the serial baud rate. Legitimate values are: 9600, 19200, 38400, 57600, 115200 (default), 230400,460800 and 921600. Operation at rates higher than the default value are not recommended but are possible in some cases. See document *320-0287 Using the USB and Serial Interfaces* for details.

3.31 RCM_SET_SERIAL_BAUD_RATE_REQUEST (0xF00B)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SET_SERIAL_BAUD_RATE_CONFIRM (Radio)

Purpose: This message sets the serial baud rate on the P4xx.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_SERIAL_BAUD_RATE _REQUEST (0xF00B)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.

#	Parameter	Type	Definition
2	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.
3	Unused	UINT8	Reserved
4	Unused	UINT16	Reserved
5	Baud Rate		Specifies the serial baud rate. Legitimate values are: 9600, 19200, 38400, 57600, 115200 (default), 230400,460800 and 921600. Operation at rates higher than the default value are not recommended but are possible in some cases. See document *320-0287 Using the USB and Serial Interfaces* for details.

3.32 RCM_SET_SERIAL_BAUD_RATE_CONFIRM (0xF10B)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SET_SERIAL_BAUD_RATE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_SET_SERIAL_BAUD_RATE_REQUEST command. This response verifies the P4xx received the request and contains a status code indicating success or failure of setting the new baud rate.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_SERIAL_BAUD_RATE _CONFIRM (0xF10B)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.33 RCM_INVALID_MESSAGE_CONFIRM (0xF10C)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: N/A

Purpose: This message is sent by the P4xx to the Host in immediate response to an unknown command. The response includes the unknown message type as well as the corresponding Message ID.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_INVALID_MESSAGE_CONFIRM (0xF10C)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	Invalid Message Type	UINT16	The unknown message type sent by the Host.
3	Invalid Message ID	UINT16	The tracking number used to associate the invalid Host REQUEST message to this CONFIRM message.
4	Status	UINT32	Status = 5 indicates this message is in response to an incorrect message size. Status = 8 indicates this is in response to an unrecognized message.

3.34 RCM_FULL_RANGE_INFO (0x0201)

API: RCM API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent by the local requesting P4xx to its Host at the end of a full UWB ranging conversation or timeout. Timeouts, indicated in the status field, can occur if the targeted responder failed to receive, or the requester failed to acquire the response packet.

Note: this structure was modified and extended in RCM v2.0 to include Coarse Range Estimates (CRE). If the P4xx was able to generate a CRE from any received UWB packet, and the P4xx is connected to a Host, it will send an RCM_FULL_RANGE_INFO message to the Host. Setting the DISABLE_CRE_RANGE_MSGS bit in the configuration flags (RCM_SET_CONFIG_REQUEST) will disable this behavior.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_FULL_RANGE_INFO (0x0201)	UINT16	Message type
1	Message ID	UINT16	Identifier to correlate range requests with info messages
2	Responder ID	UINT32	Node ID of the UWB module that sent the range response.
3	Range Status	UINT8	See Table 3-2 for a detailed description.
4	Antenna Mode	UINT8	Specifies the antenna ports used during this range conversation. The lower nibble describes the antenna configuration used by the requester, and the upper nibble the antenna configuration used by the responder. Valid values for each nibble are: 0 = Transmit and Receive on the A port, 1 = Transmit and Receive on the B port, 2 = TX on A, RX on B, 3 = TX on B, RX on A.
5	Stopwatch Time	UINT16	Duration of the range conversation in milliseconds.
6	Precision Range Measurement (PRM)	UINT32	Raw precise distance in millimeters between UWB modules based on a Two-Way Time-of-Flight (TW-TOF) measurement.

#	Parameter	Type	Definition
7	Coarse Range Estimate (CRE)	UINT32	Raw coarse distance in millimeters based on direct path signal strength. Note this value is recalibrated to the PRM value each time a PRM of the link is measured.
8	Filtered Range Estimate (FRE)	UINT32	Filtered distance in millimeters based on the combination of PRM and CRE values passed through a recursive optimal Kalman estimator with two state variables (r & rdot). If CREs are not used, the FRE will still be generated and will use the Kalman estimator and the PRM. FREs can provide higher update rates when used in a network. The increase in update rate comes at the cost of higher range error and increased latency. This is an advanced feature may benefit some networks. However, we strongly recommend that users focus on using PRMs and avoid FREs. Those interested in advanced networks should contact TDSR to discuss whether or not consider using FREs.
9	PRM Error (PRME)	UINT16	Estimated standard deviation error, in millimeters, of associated PRM value. Estimated from pulse waveform signature.
10	CRE Error (CREE)	UINT16	Estimated standard deviation error, in millimeters, of associated CRE value.
11	FRE Error (FREE)	UINT16	Estimated standard deviation error, in millimeters, of associated FRE value. Derived by the filter.
12	Filtered Range Velocity (FRV)	INT16	Estimated radial velocity in millimeters per second.
13	FRV Error (FRVE)	UINT16	Estimated standard deviation, in millimeters per second, of the FRV value.
14	Range Measurement Type	UINT8	Specifies the valid components of this message. 1 = Precision Range Measurement (PRM) 2 = Coarse Range Estimate (CRE) 4 = Filtered Range Estimate (FRE)

#	Parameter	Type	Definition
15	reserved	UINT8	SNR Reported by the responder, in integer dB
16	Requester LED Flags	UINT16	These characteristics refer to the requester's received scan: 1 (Bit 0) = SATURATED 2 (Bit 1) = SCAN WINDOW TOO SHORT4 (Bit 2) = SNR TOO LOW 8 (Bit 3) = LINE OF SIGHT (LOS)16 (Bit 4) = NON-LINE OF SIGHT (NLOS) 32 (Bit 5) = LINE-OF-SIGHT PEAK MISMATCH 128 (Bit 7) = RMS Delay Spread Squared and Noise Ratio Thresholds Exceeded (internal LED metric) 256 (Bit 8) = Norm Squared Residual Threshold Exceeded (internal LED metric)
17	Responder LED Flags	UINT16	These characteristics refer to the responder's received scan, or, for a CRE-only range, the local P4xx's received scan: 1 (Bit 0) = SATURATED 2 (Bit 1) = SCAN WINDOW TOO SHORT4 (Bit 2) = SNR TOO LOW 8 (Bit 3) = LINE OF SIGHT (LOS)16 (Bit 4) = NON-LINE OF SIGHT (NLOS) 32 (Bit 5) = LINE-OF-SIGHT PEAK MISMATCH 128 (Bit 7) = RMS Delay Spread Squared and Noise Ratio Thresholds Exceeded (internal LED metric) 256 (Bit 8) = Norm Squared Residual Threshold Exceeded (internal LED metric)
18	Noise	UINT16	Noise represents the noise generated in the receiver from the received waveform scan. Along with Vpeak, Noise can be used to compute SNR using 20 * Log_{10} (Vpeak / Noise).
19	Vpeak	UINT16	The absolute maximum value in the leading edge window of the received waveform.
20	Coarse TOF	INT32	The range measurement before applying leading edge offsets.

#	Parameter	Type	Definition
21	Timestamp	UINT32	If in RCM Mode, or Pure ALOHA RangeNet Mode (the default mode for RangeNet), this is a snapshot of milliseconds from boot to time of range conversation completion. If in other RangeNet Modes, this is a snapshot of the RangeNet microsecond network clock.

3.35 RCM_DATA_INFO (0x0202)

API: RCM API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent by the local connected P4xx to its Host whenever the P4xx receives a UWB message. Note that a promiscuous P4xx will “overhear” two packets per range conversation: one is the request packet, and the other is the response packet. If these packets contain user data then the promiscuous P4xx will send two of RCM_DATA_INFO packets to its Host (if attached). This enables the P4xx to support “sideways” data transmission outside the range request/response pair.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_DATA_INFO (0x0202)	UINT16	Message type
1	Message ID	UINT16	Identifier to correlate data packets across radios
2	Source ID	UINT32	Node ID of the UWB module that sent the data.
3	Noise	UINT16	Noise represents the noise generated in the receiver from the receive waveform scan. Along with Vpeak, Noise can be used to compute SNR using 20 * Log_{10} (Vpeak / Noise).
4	Vpeak	UINT16	The absolute maximum value in the leading edge window of the received waveform. This value is used to determine the Coarse Range Estimate.
5	Timestamp	UINT32	Milliseconds from boot to time of data reception.

#	Parameter	Type	Definition
6	Antenna ID	UINT8	Indicates which antenna was used to receive this data (0=A, 1=B).
7	Reserved	UINT8	Reserved
8	Data Size	UINT16	Number of bytes received. These bytes follow.
9	Data	N*UINT8	Data bytes received. Max will be 1000.

3.36 RCM_SCAN_INFO (0x0203)

API: RCM API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is optionally sent by the P4xx to the Host whenever it receives an RF packet. This is debug information only and is not sent by default. The default state is defined by the RCM_SET_CONFIG_REQUEST message.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SCAN_INFO (0x0203)	UINT16	Message type
1	Message ID	UINT16	Identifier used to correlate transmissions with info messages
2	Source ID	UINT32	Node ID of the transmitting radio
3	Antenna ID	UINT8	Designator of receiving antenna (0=A, 1=B)
4	Reserved	UINT8	Reserved

#	Parameter	Type	Definition
5	LED Flags	UINT16	These characteristics refer to the received scan: 1 (Bit 0) = SATURATED 2 (Bit 1) = SCAN WINDOW TOO SHORT 4 (Bit 2) = SNR TOO LOW 8 (Bit 3) = LINE OF SIGHT (LOS) 16 (Bit 4) = NON-LINE OF SIGHT (NLOS) 32 (Bit 5) = LINE-OF-SIGHT PEAK MISMATCH 128 (Bit 7) = RMS Delay Spread Squared Threshold Exceeded (internal LED metric) 256 (Bit 8) = NoiseRatio Threshold Exceeded (internal LED metric)
6	Noise	UINT16	Noise represents the noise generated in the receiver from the receive waveform scan. Along with Vpeak, Noise can be used to compute SNR using 20 * Log_{10} (Vpeak / Noise).
7	Vpeak	UINT16	The absolute maximum value in the leading edge window of the received waveform. This value is used to determine the Coarse Range Estimate.
8	Timestamp	UINT32	Milliseconds from boot to time of data reception.
9	Leading Edge Offset	INT32	Offset from first sample where radio found leading edge of pulse.
10	Lockspot Offset	INT32	Offset from first sample where radio locked on the pulse.
11	Number of Scan Samples	UINT32	The number of data points (UINT32) that follow. Maximum number of points is 350.
12	Scan Data	INT32	Scan values collected by the radio.

3.37 RCM_ECHOED_RANGE_INFO (0x0204)

API: RCM API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent when Echo Last Range (ELR) is enabled via the

RCM_SET_CONFIG_REQUEST message. The ELR feature allows the current node to receive the PRMs from other radio pairs. When this feature is enabled, the radio will place the last successful PRM in its request packet. This allows other radios that hear the request (but are not the target of the request) to know the ranges of the requesting radio. Because ranging occurs in a round-robin fashion, this allows all radios to know all ranges that occur between the radios in the network.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_ECHOED_RANGE_INFO (0x0204)	UINT16	Message type
1	Message ID	UINT16	Identifier used to correlate transmissions with info messages
2	Requester ID	UINT32	Node ID of the transmitting radio
3	Responder ID	UINT32	Node ID of the responding radio
4	Precision Range Measurement (PRM)	UINT32	PRM is the raw precise distance in millimeters between UWB modules based on a Two-Way Time-of-Flight (TW-TOF) measurement.
5	PRM Error Estimate	UINT16	PRME is the estimated standard deviation error, in millimeters, of the associated PRM value. Estimated from the pulse waveform signature.
6	Requester LED Flags OR'd with Responder LED Flags	UINT16	These characteristics refer to the requester's and responder's received LED scan: 1 (Bit 0) = SATURATED 8 (Bit 3) = LINE OF SIGHT (LOS) 16 (Bit 4) = NON-LINE OF SIGHT (NLOS) 32 (Bit 5) = LINE-OF-SIGHT PEAK MISMATCH 128 (Bit 7) = RMS Delay Spread Squared Threshold Exceeded (internal LED metric) 256 (Bit 8) = NoiseRatio Threshold Exceeded (internal LED metric)
7	Timestamp	UINT32	Milliseconds from boot to time of range conversation completion, on the original requesting radio. This can be used to identify duplicate echoed range reports.

3.38 RCM_SMALL_RANGE_INFO (0x0205)

API: RCM API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent by the local requesting P4xx to its Host at the end of a full UWB ranging conversation or timeout much like the RCM_RANGE_INFO message. However, the RCM_SMALL_RANGE_INFO message contains a subset of the RCM_RANGE_INFO. This allows for obtaining necessary range information in bandwidth limited configurations.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SMALL_RANGE_INFO (0x0205)	UINT16	Message type
1	Message ID	UINT16	Identifier to correlate range requests with info messages
2	Responder ID	UINT32	Node ID of the UWB module that sent the range response.
3	Range	UINT16	Range measurement or estimate in centimeters based on Range Measurement Type setting (see Item 5 below).
4	Range Error Estimate	UINT8	Estimated standard deviation error, in centimeters, of associated range value. Based on Range Measurement Type (see Item 5 below).
5	Range Measurement Type	UINT8	Specifies the valid components of this message. 1 = Precision Range Measurement (PRM) 2 = Coarse Range Estimate (CRE) 4 = Filtered Range Estimate (FRE)
6	Range Status	UINT8	See Table 3-2 for a detailed description.
7	Reserved	UINT8	Reserved

3.39 RCM_FULL_SCAN_INFO (0xF201)

API: RCM API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is optionally sent by the P4xx to the Host whenever it receives an

RF packet. This is debug information only and is not sent by default. The default state is defined by the RCM_SET_CONFIG_REQUEST message. Due to the number of samples in a full waveform scan, the waveform scan data is split into many RCM_FULL_SCAN_INFO messages.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_FULL_SCAN_INFO (0xF201)	UINT16	Message type
1	Message ID	UINT16	Identifier used to correlate transmissions with info messages
2	Source ID	UINT32	Node ID of the transmitting radio
3	Timestamp	UINT32	Milliseconds from boot to time of data reception.
4	Noise	UINT16	Noise represents the noise generated in the receiver from the receive waveform scan. Along with Vpeak, Noise can be used to compute SNR using 20 * Log_{10} (Vpeak / Noise).
5	Vpeak	UINT16	The absolute maximum value in the leading edge window of the received waveform. This value is used to determine the Coarse Range Estimate.
6	Reserved	UINT32	Reserved
7	Leading Edge Offset	INT32	Offset from first sample where radio found leading edge of pulse.
8	Lockspot Offset	INT32	Offset from first sample where radio locked on the pulse.
9	Scan Start	INT32	The start position of the waveform scan relative to the lockspot in picoseconds.
10	Scan Stop	INT32	The stop position of the waveform scan relative to the lockspot in picoseconds.
11	Scan Step	UINT16	The amount of time between each sample in bins. By default this is 32 bins equating to about 61 picoseconds between samples.
12	Reserved	UINT8	Reserved
13	Reserved	UINT8	Reserved
14	Antenna ID	UINT8	Designator of receiving antenna (0=A, 1=B)

#	Parameter	Type	Definition
15	Operational Mode	UINT8	Specifies the mode of the radio when this scan was collected. [RCM=0, RangeNet=4]
16	Number of Samples in this Message	UINT16	The number of samples that follow in this message (UDP or Serial). Maximum is 350 samples in a single RCM_FULL_SCAN_INFO message.
17	Total Number of Scan Samples	UINT32	The total number of samples in the full scan.
18	Message Index	UINT16	The index of this message within the sequence of RCM_FULL_SCAN_INFO messages.
19	Total Number of Messages	UINT16	The total number of RCM_FULL_SCAN_INFO messages in the entire scan.
20	Scan Data	350*INT32	Scan values collected by the radio.

3.40 RCM_GET_GPIO_CONFIG_REQUEST (0xF012)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_GET_GPIO_CONFIG_CONFIRM (Radio)

Purpose: This message queries the P4xx for the current GPIO configuration.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_GPIO_CONFIG_REQUEST (0xF012)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.

3.41 RCM_GET_GPIO_CONFIG_CONFIRM (0xF112)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_GET_GPIO_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_GET_GPIO_CONFIG_REQUEST command. This response verifies the P4xx received the request and contains the current GPIO configuration.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_GPIO_CONFIG_CONFIRM (0xF112)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	GPIO Mode	UINT32	Specifies the mode for each GPIO, 2 bits per GPIO. A mode of 00b selects I/O while other values are specific to the GPIO. See *320-0320 RangeNet User Guide* for details on GPIOs and their various modes. The mode for GPIO 0 is mapped to GPIO Mode bits [0:1]

#	Parameter	Type	Definition
3	GPIO Direction	UINT16	If a GPIO mode is 00b, the GPIO Direction bit for that mode sets the GPIO as either an input (value of 0) or an output (value of 1). GPIO 0 is mapped to GPIO Direction bit 0
4	Reserved	UINT16	

3.42 RCM_SET_GPIO_CONFIG_REQUEST (0xF013)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SET_GPIO_CONFIG_CONFIRM (Radio)

Purpose: This message sets the GPIO configuration on the P4xx.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_GPIO_CONFIG_REQUEST (0xF013)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	GPIO Mode	UINT32	Specifies the mode for each GPIO, 2 bits per GPIO. A mode of 00b selects I/O while other values are specific to the GPIO. See *320-0320 RangeNet User Guide* for details on GPIOs and their various modes. The mode for GPIO 0 is mapped to GPIO Mode bits [0:1]
3	GPIO Direction	UINT16	If a GPIO mode is 00b, the GPIO Direction bit for that mode sets the GPIO as either an input (value of 0) or an output (value of 1). GPIO 0 is mapped to GPIO Direction bit 0
4	Reserved	UINT8	

#	Parameter	Type	Definition
5	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.

3.43 RCM_SET_GPIO_CONFIG_CONFIRM (0xF113)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SET_GPIO_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_GET_GPIO_CONFIG_REQUEST command. This response verifies the P4xx received the request and contains a status code indicating success or failure in setting the GPIO configuration.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_GPIO_CONFIG_CONFIRM (0xF113)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.

#	Parameter	Type	Definition
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

3.44 RCM_GET_GPIO_ REQUEST (0xF014)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_GET_GPIO_CONFIRM (Radio)

Purpose: This message queries the P4xx for the current state of GPIOs.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_GPIO_REQUEST (0xF014)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.

3.45 RCM_GET_GPIO_CONFIRM (0xF114)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_GET_GPIO_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_GET_GPIO_REQUEST command. This response verifies the P4xx received the request and contains the current state of the GPIOs.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_GET_GPIO_CONFIRM (0xF114)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	GPIO State	UINT16	Specifies the state for each GPIO. Bit 0 is mapped to GPIO 0

#	Parameter	Type	Definition
3	GPIO I/O Value	UINT16	Specifies the state of each GPIO set to I/O as an output. Used to verify user settings

3.46 RCM_SET_GPIO_REQUEST (0xF015)

API: RCM API
Message type: REQUEST (Host)
Corresponding Message type: RCM_SET_GPIO_CONFIRM (Radio)

Purpose: This message sets the state of the GPIOs on the P4xx.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_GPIO_ REQUEST (0xF015)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.
2	GPIO	UINT16	Drives a 0 or 1 onto a GPIO configured as output. Ignored for others. Bit 0 is mapped to GPIO 0.
3	Mask	UINT16	Masks the bits in GPIO so that only GPIOs set in the Mask are changed.
4	Reserved	UINT16	
5	Reserved	UINT8	

#	Parameter	Type	Definition
6	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.

3.47 RCM_SET_GPIO_CONFIRM (0xF115)

API: RCM API
Message type: CONFIRM (Radio)
Corresponding Message type: RCM_SET_GPIO_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RCM_GET_GPIO_CONFIG_REQUEST command. This response verifies the P4xx received the request and contains a status code indicating success or failure in setting the state of the GPIOs.

Packet Definition:

#	Parameter	Type	Definition
0	RCM_SET_GPIO_CONFIRM (0xF115)	UINT16	Message type
1	Message ID	UINT16	A tracking number used to associate Host REQUEST messages to RCM CONFIRM messages.

#	Parameter	Type	Definition
2	Status	UINT32	0 = Successful. For error codes see Table 3-1 at the end of this section.

Table 3-1: CONFIRM Message Status Codes

Code	Status	Description
0	Success	The REQUEST message was processed successfully.
1	Generic Failure	Catch-all for uncategorized failures.
2	Wrong OpMode	The REQUEST message cannot be acted upon in the current OpMode.
3	Unsupported Value	The REQUEST message contained an unsupported value in one or more of its fields.
4	Invalid During Sleep	The REQUEST message cannot be acted upon in the current sleep mode.
5	Wrong Message Size	The number of bytes in the REQUEST message did not match the expected number of bytes for the message type.
6	Not Enabled	The feature used by the REQUEST message is currently disabled.
7	Wrong Buffer Size	The specified size of a buffer in the REQUEST message, or the size of the buffer itself, did not match the expected number of bytes for the message type.
8	Unrecognized Message Type	The REQUEST Message Type was not recognized.
0x80000000	Internal Error Code	An internal error code was generated. This status is OR'ed with the internal error code itself and should be used in communication with TDSR’s support team.

Table 3-2: Range Info Message Status Codes

Code	Status	Description
0	Success	The requester successfully completed a range measurement.

1	Timeout Failure	The requester initiated a range request but the intended responder never responded. For example, the responder may be out of range, operating on a different channel or at a different PII.
2	Requester LED Failure	The requester was unable to properly detect the leading edge of the waveform. (However the responder was able to properly detect the leading edge of the waveform it received.) This range reading is probably incorrect and should be ignored.
3	Virtual Carrier Sense	The Host computer requested that the unit initiate a range request. However, when the Host message was received, the P4xx was either responding to a range request from another unit or had detected that a range request between two other units was currently in process on this communications code channel. Because the P4xx was busy, the requested range request was cancelled. In other words, the P4xx or channel is busy, please try again later.
4	Responder LED Failure	The responder was unable to properly detect the leading edge of the waveform. (However the requester was able to properly detect the leading edge of the waveform it received.) This range reading is probably incorrect and should be ignored.
5	Requester RX weak	The requester has determined that the waveform it received is >10 dB weaker than the waveform which the responder received. This measurement is based on SNR, not Vpeak. If the RF channel is the same, then the two units should be measuring waveforms of similar magnitudes. That there is a difference probably indicates that either: a) the responder is transmitting less power than the requester b) the requester has a weak receiver c) the RF channel is not symmetric, or d) an interference source close to the requester may be increasing the noise floor of the requester, but is too distant to affect the responder. The most likely cause is the units have been set up to operate at different transmit powers. If the units are operating at the same transmit power and the receivers are both functioning properly then the reading should be discarded.
6	Req/Res LED Failure	Both the requester and responder were unable to properly detect the leading edge of their respective waveforms. This range reading is probably incorrect and should be ignored.
7	Responder RX Weak	The requester has determined that the waveform it received is >10 dB stronger than the waveform which

		the responder received. This measurement is based on SNR not Vpeak. If the RF channel is the same, then the two units should be measuring waveforms of similar magnitudes. That there is a difference probably indicates that either: a) the responder is transmitting more power than the requester b) the responder has a weak receiver c) the RF channel is not symmetric, or d) an interference source close to the responder may be increasing the noise floor of the responder, but is too distant to affect the requester. The most likely cause is the units have been set up to operate at different transmit powers. If the units are operating at the same transmit power and the receivers are both functioning properly then the reading should be discarded.
11	LOS/NLOS Mismatch	The requester and responder disagree as to the nature of the channel. One believes that the channel is LOS (line of sight), the other thinks the channel is NLOS (non-line of sight). There are several possible causes. The units may be operating in different RF channels. The link may be very weak and the LOS/NLOS detection algorithm is confused. The safest course of action is to ignore the reading. If this happens continuously in your application, please contact technical support at TDSR.
15	Both-Sides-Delta-Check Failed	The requester detected a mismatch between the LED on the responder and the requester, indicating a possible incorrect range.
32	Range out of bounds	The time of flight range calculation process has resulted in a range that is out of bounds. This reading should be ignored.
64	Coarse Range Estimate	This information was not generated as the result of a precision range request. This information was generated by a coarse range measurement.

4 RangeNet API Messages

4.1 RN_SET_CONFIG_REQUEST (0x3001)

API: RangeNet API

Message type: REQUEST (Host)
Corresponding Message type: RN_SET_CONFIG_CONFIRM (Radio)

Purpose: This message configures RangeNet parameters and network behavior.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_CONFIG_REQUEST (0x3001)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Max Neighbor Age	UINT32	Specifies the maximum age, in milliseconds, of a neighbor node in this node's local database. Database entries older than this are deleted. Default value is 10000.
3	Autosend Update Interval for Neighbor Database	UINT16	Specifies the milliseconds between each automatic pushing of the neighbor database to the Host. Valid values are 0-65535. For performance reasons, the interval is limited to a minimum of 100 ms (mz rate of 10Hz). Default is 300 ms (3.33 Hz).
4	Configuration Flags	UINT16	Bit flags for various configuration parameters: • Bit 0 – Reserved • Bit 1 – Do not range to me; 0 = Normal mode, 1 = Do not range to me (Default = 0) • Bit 2 – Reserved; set to 0 • Bit 3 – Echo Last Range; 0 = Disabled, 1 = Enabled (Default = 0) • Bits 4:15 – Reserved
5	Network Sync Mode	UINT8	Network synchronization mode [0=ALOHA (default), 1=TDMA]

#	Parameter	Type	Definition
6	Autosend Flags	UINT8	Specifies what messages to automatically send to the Host. Lower 2 bits (bits 0-1) control sending of RANGE_INFO messages: 0: Do not automatically send RCM_RANGE_INFO messages. 1: Send RCM_RANGE_INFO messages for only successful range calculations. 2: Send RCM_RANGE_INFO messages for all range calculations. Upper 2 bits of the lower nibble (bits 2-3) control sending of the Neighbor Database: 0: Do not automatically send the Neighbor Database. 1: Automatically send the full Neighbor Database format periodically. 2: Automatically send the small Neighbor Database format periodically. Lower 2 bits of the upper nibble (bits 4-5) control sorting of the Neighbor Database: 0: Sort by Node ID 1: Sort by range, closest first 2: Sort by age, newest first Default is 0x04: Autosend the full Neighbor Database.
7	Reserved	UINT8	Reserved. Set to 0.
8	Default Interface	UINT8	Default interface for INFO messages. When radio boots, it will automatically send INFO messages to the selected default interface. [0=None, 1=Ethernet, 2=USB, 3=Serial, 4=CAN]. Note that some interfaces are radio-specific.
9	Default Interface Address1	UINT32	Default address for default interface. When Ethernet is selected as the default interface, this value is the Ethernet IP address. If CAN is chosen as the interface, this value is the corresponding CAN address.

#	Parameter	Type	Definition
10	Default Interface Address2	UINT32	Default address for default interface. If Ethernet is chosen as the default interface, this value will be the Ethernet Port number.
11	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.
12	Reserved	UINT8	Reserved. Set to 0.
13	Reserved	UINT16	Reserved. Set to 0.

4.2 RN_SET_CONFIG_CONFIRM (0x3101)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_SET_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RN_SET_CONFIG_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of the RN_SET_CONFIG_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_CONFIG_CONFIRM (0x3101)	UINT16	Message type

#	Parameter	Type	Definition
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.3 RN_GET_CONFIG_REQUEST (0x3002)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_CONFIG_CONFIRM (Radio)

Purpose: This is a request message sent by the Host to P4xx to request a copy of the current RangeNet configuration.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_CONFIG_REQUEST (0x3002)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

4.4 RN_GET_CONFIG_CONFIRM (0x3102)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_GET_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx in response to a RN_GET_CONFIG_REQUEST from the Host. It provides the current RangeNet configuration information.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_CONFIG_CONFIRM (0x3102)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages

#	Parameter	Type	Definition
2	Max Neighbor Age	UINT32	Specifies the maximum age, in milliseconds, of a neighbor node in this node's local database. Database entries older than this are deleted. Default value is 10000.
3	Autosend Update Interval for Neighbor Database	UINT16	Specifies the milliseconds between each automatic pushing of the neighbor database to the Host. Valid values are 0-65535. Default is 300 ms (3.33 Hz).
4	Configuration Flags	UINT16	Bit flags for various configuration parameters; • Bit 0 – Reserved • Bit 1 – Do not range to me; 0 = Normal mode, 1 = Do not range to me (Default = 0) • Bit 2 – Reserved; set to 0 • Bit 3 – Echo Last Range; 0 = Disabled, 1 = Enabled (Default = 0) Bits 4:15 – Reserved
5	Network Sync Mode	UINT8	Network synchronization mode [0=ALOHA (default), 1=TDMA]

#	Parameter	Type	Definition
6	Autosend Flags	UINT8	Specifies what messages to automatically send to the Host. Lower 2 bits (bits 0-1) control sending of RANGE_INFO messages: 0: Do not automatically send RCM_RANGE_INFO messages. 1: Send RCM_RANGE_INFO messages for only successful range calculations. 2: Send RCM_RANGE_INFO messages for all range calculations. Upper 2 bits of the lower nibble (bits 2-3) control sending of the Neighbor Database: 0: Do not automatically send the Neighbor Database. 1: Automatically send the full Neighbor Database format periodically. 2: Automatically send the small Neighbor Database format periodically. Lower 2 bits of the upper nibble (bits 4-5) control sorting of the Neighbor Database: 0: Sort by Node ID 1: Sort by range, closest first 2: Sort by age, newest first Default is 0x04: Autosend the full Neighbor Database.
7	Reserved	UINT8	Reserved
8	Default Interface	UINT8	Default interface for INFO messages. When radio boots, it will automatically send INFO messages to the selected default interface. [0=None, 1=Ethernet, 2=USB, 3=Serial, 4=CAN]. Note that some interfaces are radio-specific.

#	Parameter	Type	Definition
9	Default Interface Address1	UINT32	Default address for default interface. When Ethernet is selected as the default interface, this value is the Ethernet IP address. If CAN is chosen as the interface, this value is the corresponding CAN address.
10	Default Interface Address2	UINT32	Default address for default interface. If Ethernet is chosen as the default interface, this value will be the Ethernet Port number.
11	Timestamp	UINT32	Milliseconds since radio boot and this message sent.
12	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.5 RN_SET_ALOHA_CONFIG_REQUEST (0x300D)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_SET_ALOHA_CONFIG_CONFIRM (Radio)

Purpose: This message configures the RangeNet parameters for the ALOHA network mode.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_ALOHA_CONFIG_REQUEST (0x300D)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Min Time Between TX	UINT16	Specifies the minimum time, in milliseconds, between start of one range request to start of the next range request from this node.
3	Max Time Between TX	UINT16	Specifies the maximum time, in milliseconds, between start of one range request to start of the next range request from this node.

#	Parameter	Type	Definition
4	Max Request Data Size	UINT16	Specifies the maximum amount of user data in a range request the Host application will use. If the Host application knows it will never use the maximum 1000 bytes, this can be set to a lower value. This enables RangeNet to reduce slot duration in time-slotted networks in order to increase update rate. Default value is 10. Valid values are 0 through 1000. When operating in Location mode the max value is 900.
5	Max Response Data Size	UINT16	Specifies the maximum amount of user data in a range response the Host application will use. If the Host application knows it will never use the maximum 1000 bytes, this can be set to a lower value. This enables RangeNet to reduce slot duration in time-slotted networks in order to increase update rate. Default value is 10. Valid values are 0 through 1000. When operating in Location mode the max value is 900.
6	Configuration Flags	UINT16	Bit flags for various configuration parameters; • Bit 0 – Beacon mode. 0 = Normal mode, 1 = Beacon mode (Default = 0) • Bit 1 – Reserved. Set to 0. • Bit 2 – Automatic Congestion Control (ACC), 0 = Disabled, 1 = Enabled (Default = 0) • Bits 3:15 – Reserved
7	Reserved	UINT16	Reserved. Set to 0.

#	Parameter	Type	Definition
8	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.
9	Reserved	UINT8	Reserved. Set to 0.
10	Reserved	UINT16	Reserved. Set to 0.

4.6 RN_SET_ALOHA_CONFIG_CONFIRM (0x310D)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_SET_ALOHA_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RN_SET_ALOHA_CONFIG_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of RN_SET_ALOHA_CONFIG_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_ALOHA_CONFIG_CONFIRM (0x310D)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

#	Parameter	Type	Definition
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.7 RN_GET_ALOHA_CONFIG_REQUEST (0x300E)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_ALOHA_CONFIG_CONFIRM (Radio)

Purpose: This is a request message sent by the Host to P4xx to request a copy of the current RangeNet ALOHA configuration.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_ALOHA_CONFIG_REQUEST (0x300E)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

4.8 RN_GET_ALOHA_CONFIG_CONFIRM (0x310E)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_GET_ALOHA_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx in response to a RN_GET_ALOHA_CONFIG_REQUEST from the Host. It provides the current RangeNet ALOHA configuration information.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_ALOHA_CONFIG_CONFIRM (0x310E)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Min Time Between TX	UINT16	Specifies the minimum time, in milliseconds, between start of one range request to start of the next range request from this node.

#	Parameter	Type	Definition
3	Max Time Between TX	UINT16	Specifies the maximum time, in milliseconds, between start of one range request to start of the next range request from this node.
4	Max Request Data Size	UINT16	Specifies the maximum amount of user data in a range request the Host application will use. If the Host application knows it will never use the maximum 1000 bytes, this can be set to a lower value. This enables RangeNet to reduce slot duration in time-slotted networks in order to increase update rate. Default value is 10. Valid values are 0 through 1000. When operating in Location mode the max value is 900.
5	Max Response Data Size	UINT16	Specifies the maximum amount of user data in a range response the Host application will use. If the Host application knows it will never use the maximum 1000 bytes, this can be set to a lower value. This enables RangeNet to reduce slot duration in time-slotted networks in order to increase update rate. Default value is 10. Valid values are 0 through 1000. When operating in Location mode the max value is 900.
6	Configuration Flags	UINT16	Bit flags for various configuration parameters: • Bit 0 – Beacon mode. 0 = Normal mode, 1 = Beacon mode (Default = 0) • Bit 1 – Reserved • Bit 2 – Automatic Congestion Control (ACC), 0 = Disabled, 1 = Enabled (Default = 0) • Bits 3:15 – Reserved
7	Reserved	UINT16	Reserved
8	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.9 RN_SET_TDMA_CONFIG_REQUEST (0x3013)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_SET_TDMA_CONFIG_CONFIRM (Radio)

Purpose: This message configures the RangeNet parameters for the TDMA network mode.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_TDMA_CONFIG_REQUEST (0x3013)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Max Request Data Size	UINT16	Specifies the maximum amount of user data in a range request the Host application will use. If the Host application knows it will never use the maximum 1000 bytes, this can be set to a lower value. This enables RangeNet to reduce slot duration in TDMA networks in order to increase update rate. Default value is 10. Valid values are 0 through 1000. When operating in Location mode the max value is 900.
3	Max Response Data Size	UINT16	Specifies the maximum amount of user data in a range response the Host application will use. If the Host application knows it will never use the maximum 1000 bytes, this can be set to a lower value. This enables RangeNet to reduce slot duration in TDMA networks in order to increase update rate. Default value is 10. Valid values are 0 through 1000. When operating in Location mode the max value is 900.
4	Reserved	UINT16	Reserved. Set to 0.
5	Reserved	UINT16	Reserved. Set to 0.

#	Parameter	Type	Definition
6	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.
7	Reserved	UINT8	Reserved. Set to 0.
8	Reserved	UINT16	Reserved. Set to 0.

4.10 RN_SET_TDMA_CONFIG_CONFIRM (0x3113)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_SET_TDMA_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RN_SET_TDMA_CONFIG_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of RN_SET_TDMA_CONFIG_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_TDMA_CONFIG_CONFIRM (0x3113)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

#	Parameter	Type	Definition
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.11 RN_GET_TDMA_CONFIG_REQUEST (0x3014)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_TDMA_CONFIG_CONFIRM (Radio)

Purpose: This is a request message sent by the Host to P4xx to request a copy of the current RangeNet TDMA configuration.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_TDMA_CONFIG_REQUEST (0x3014)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

4.12 RN_GET_TDMA_CONFIG_CONFIRM (0x3114)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_GET_TDMA_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx in response to a RN_GET_TDMA_CONFIG_REQUEST from the Host. It provides the current RangeNet TDMA configuration information.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_TDMA_CONFIG_CONFIRM (0x3114)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages

#	Parameter	Type	Definition
2	Max Request Data Size	UINT16	Specifies the maximum amount of user data in a range request the Host application will use. If the Host application knows it will never use the maximum 1000 bytes, this can be set to a lower value. This enables RangeNet to reduce slot duration in TDMA networks in order to increase update rate. Default value is 10. Valid values are 0 through 1000. When operating in Location mode the max value is 900.
3	Max Response Data Size	UINT16	Specifies the maximum amount of user data in a range response the Host application will use. If the Host application knows it will never use the maximum 1000 bytes, this can be set to a lower value. This enables RangeNet to reduce slot duration in TDMA networks in order to increase update rate. Default value is 10. Valid values are 0 through 1000. When operating in Location mode the max value is 900.
4	Reserved	UINT16	Reserved
5	Reserved	UINT16	Reserved
6	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.13 RN_SET_TDMA_SLOTMAP_REQUEST (0x3010)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_SET_TDMA_SLOTMAP_CONFIRM (Radio)

Purpose: This message configures the RangeNet TDMA Slot Map. The message can be used to overwrite an existing Slot Map as well as update one or many slots.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_TDMA_SLOTMAP_REQUEST (0x3010)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages

<table>
<tr><th>#</th><th>Parameter</th><th>Type</th><th>Definition</th></tr>
<tr><td>2</td><td>Number of Slots</td><td>UINT8</td><td>Indicates the number of slots included in the slot definitions below. Maximum number of slots is 32.</td></tr>
<tr><td>3</td><td>Slotmap Flags</td><td>UINT8</td><td>These flags are defined as follows;
Bit 0: Modify Slotmap – when this flag is set, this message will update the Slot Map instead of overwriting it.</td></tr>
<tr><td>4</td><td>Reserved</td><td>UINT8</td><td>Reserved. Set to 0.</td></tr>
<tr><td>5</td><td>Persist Flag</td><td>UINT8</td><td>Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling).
Regardless of the Persist Flag’s setting, the active radio settings are immediately updated from this message.
Possible Persist Flag values are:
0: Do not write anything to flash; only update the active radio settings.
1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages.
2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash.
For more details, see Appendix D: Persist Flag Details and Usage.</td></tr>
<tr><td colspan="4">BEGIN SLOT DEFINITIONS
The maximum number of slots is 32 and only the number of slots indicated in the ‘Number of Slots’ field above need to be included.</td></tr>
<tr><td>0</td><td>Slot Type</td><td>UINT8</td><td>Used to indicate the type of slot. Valid values are:
0: Indicates an invalid slot such as the last slot in a Slot Map.
1: Indicates a Range slot used for a normal range request – response transaction.
2: Indicates a Data slot where the radio indicated by the Requester ID sends out a data packet.</td></tr>
<tr><td>1</td><td>Slot Number</td><td>UINT8</td><td>Indicates the slot number of this particular slot.</td></tr>
</table>

#	Parameter	Type	Definition
2	Flags	UINT16	The flags for a given slot are defined as below: Bit 0: Sleep – when this flag is set, the radio will go into sleep mode if it is neither the requester nor the responder. Bit 1: Requester Data – when this flag is set, data from the requester data buffer will be included in the range request for this slot. Bit 2: Responder Data – when this flag is set, data from the responder data buffer will be included in the range response for this slot.
3	Pulse Integration Index	UINT8	Specifies an index, which configures the number of pulses per data symbol. Valid values are [4-9]. The default is 7, meaning 2^7 = 128 pulses per symbol.
4	Antenna Mode	UINT8	Specifies the default antenna for transmission and reception. Valid values are [0 = A, 1 = B, 2 = TXA/RXB, 3 = TXB/RXA]. In addition, setting the high order bit (0x80) of this byte enables automatic toggling of the antenna after each response by this radio. The default value is 0.
5	Code Channel	UINT8	Specifies the active UWB channel. Multiple ranging conversations can occur simultaneously if multiple code channels are used. Both the requester and responder radios must be configured to the same code channel for successful communication. Possible values are [0-10]. The default value is 0.
6	Reserved	UINT8	Reserved. Set to 0.
7	Requester ID	UINT32	The Node ID of the radio sending the range request.
8	Responder ID	UINT32	The Node ID of the radio to send the range response.
9	Manual Slot Duration (µs)	UINT32	Indicates the slot duration for this particular slot. A value of 0 will cause the radio to use the computed time for the slot. An error is returned when 0 < Manual Slot Time < Computed Slot Time.
END SLOT DEFINITIONS			

4.14 RN_SET_TDMA_SLOTMAP_CONFIRM (0x3110)

API: RangeNet API

Message type: CONFIRM (Radio)
Corresponding Message type: RN_SET_TDMA_SLOTMAP_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RN_SET_TDMA_SLOTMAP_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of RN_SET_TDMA_SLOTMAP_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_TDMA_SLOTMAP_CONFIRM (0x3110)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.15 RN_GET_TDMA_SLOTMAP_REQUEST (0x3011)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_TDMA_SLOTMAP_CONFIRM (Radio)

Purpose: This is a request message sent by the Host to P4xx to request a copy of the current RangeNet TDMA Slot Map.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_TDMA_SLOTMAP_REQUEST (0x3011)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

4.16 RN_GET_TDMA_SLOTMAP_CONFIRM (0x3111)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_GET_TDMA_SLOTMAP_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RN_GET_TDMA_SLOTMAP_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of RN_GET_TDMA_SLOTMAP_REQUEST.

Packet Definition:

<table>
<tr><th>#</th><th>Parameter</th><th>Type</th><th>Definition</th></tr>
<tr><td>0</td><td>RN_GET_TDMA_SLOTMAP_ CONFIRM (0x3111)</td><td>UINT16</td><td>Message type</td></tr>
<tr><td>1</td><td>Message ID</td><td>UINT16</td><td>Associates request to confirm and info messages</td></tr>
<tr><td>2</td><td>Number of Slots</td><td>UINT8</td><td>Indicates the number of slots included in the slot definitions below. Maximum number of slots is 32.</td></tr>
<tr><td>3</td><td>Reserved</td><td>UINT8</td><td>Reserved.</td></tr>
<tr><td>4</td><td>Reserved</td><td>UINT16</td><td>Reserved.</td></tr>
<tr><td>5</td><td>Status</td><td>UINT32</td><td>0 = Successful. For error codes see Table 4-1 at the end of this section.</td></tr>
<tr><td colspan="4">BEGIN SLOT DEFINITIONS
The maximum number of slots is 32 and only the number of slots indicated in the ‘Number of Slots’ field above need to be included.</td></tr>
<tr><td>0</td><td>Slot Type</td><td>UINT8</td><td>Used to indicate the type of slot. Valid values are:
0: Indicates an invalid slot such as the last slot in a Slot Map.
1: Indicates a Range slot used for a normal range request – response transaction.
2: Indicates a Data slot where the radio indicated by the Requester ID sends out a data packet.</td></tr>
<tr><td>1</td><td>Slot Number</td><td>UINT8</td><td>Indicates the slot number of this particular slot.</td></tr>
</table>

#	Parameter	Type	Definition
2	Flags	UINT16	The flags for a given slot are defined as below: Bit 0: Sleep – when this flag is set, the radio will go into sleep mode if it is neither the requester nor the responder. Bit 1: Requester Data – when this flag is set, data from the requester data buffer will be included in the range request for this slot. Bit 2: Responder Data – when this flag is set, data from the responder data buffer will be included in the range response for this slot.
3	Pulse Integration Index	UINT8	Specifies an index, which configures the number of pulses per data symbol. Valid values are [4-9]. The default is 7 meaning 2^7 = 128 pulses per symbol.
4	Antenna Mode	UINT8	Specifies the default antenna for transmission and reception. Valid values are [0 = A, 1 = B, 2 = TXA/RXB, 3 = TXB/RXA]. In addition, setting the high order bit (0x80) of this byte enables automatic toggling of the antenna after each response by this radio. The default value is 0.
5	Code Channel	UINT8	Specifies the active UWB channel. Multiple ranging conversations can occur simultaneously if multiple code channels are used. Both the requester and responder radios must be configured to the same code channel for successful communication. Possible values are [0-10]. The default value is 0.
6	Reserved	UINT8	Reserved. Set to 0.
7	Requester ID	UINT32	The Node ID of the radio sending the range request.
8	Responder ID	UINT32	The Node ID of the radio to send the range response.
9	Manual Slot Duration (µs)	UINT32	Indicates the slot duration for this particular slot. A value of 0 will cause the radio to use the computed time for the slot. An error is returned when 0 < Manual Slot Time < Computed Slot Time.
10	Computed Slot Duration (µs)	UINT32	Slot duration as computed by the radio. This time will be used if Manual Slot Time is set to 0.
END SLOT DEFINITIONS			

4.17 RN_GET_TDMA_SLOT_REQUEST (0x3012)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_TDMA_SLOT_CONFIRM (Radio)

Purpose: This is a request message sent by the Host to P4xx to request a single slot from the current RangeNet TDMA Slot Map.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_TDMA_SLOT_REQUEST (0x3012)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Slot Number	UINT8	The slot number being requested
3	Reserved	UINT8	Reserved. Set to 0.
4	Reserved	UINT16	Reserved. Set to 0.

4.18 RN_GET_TDMA_SLOT_CONFIRM (0x3112)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_GET_TDMA_SLOT_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a RN_GET_TDMA_SLOT_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of RN_GET_TDMA_SLOT_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_TDMA_SLOT_CONFIRM (0x3112)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

#	Parameter	Type	Definition
3	Slot Type	UINT8	Used to indicate the type of slot. Valid values are: 0: Indicates an invalid slot such as the last slot in a Slot Map. 1: Indicates a Range slot used for a normal range request – response transaction. 2: Indicates a Data slot where the radio indicated by the Requester ID sends out a data packet.
4	Slot Number	UINT8	Indicates the slot number of this particular slot.
5	Flags	UINT16	The flags for a given slot are defined as below: Bit 0: Sleep – when this flag is set, the radio will go into sleep mode if it is neither the requester nor the responder. Bit 1: Requester Data – when this flag is set, data from the requester data buffer will be included in the range request for this slot. Bit 2: Responder Data – when this flag is set, data from the responder data buffer will be included in the range response for this slot.
6	Pulse Integration Index	UINT8	Specifies an index, which configures the number of pulses per data symbol. Valid values are [4-9]. The default is 7 meaning 2^7 = 128 pulses per symbol.
7	Antenna Mode	UINT8	Specifies the default antenna for transmission and reception. Valid values are [0 = A, 1 = B, 2 = TXA/RXB, 3 = TXB/RXA]. In addition, setting the high order bit (0x80) of this byte enables automatic toggling of the antenna after each response by this radio. The default value is 0.
8	Code Channel	UINT8	Specifies the active UWB channel. Multiple ranging conversations can occur simultaneously if multiple code channels are used. Both the requester and responder radios must be configured to the same code channel for successful communication. Possible values are [0-10]. The default value is 0.
9	Reserved	UINT8	Reserved. Set to 0.
10	Requester ID	UINT32	The Node ID of the radio sending the range request.
11	Responder ID	UINT32	The Node ID of the radio to send the range response.

#	Parameter	Type	Definition
12	Manual Slot Duration (µs)	UINT32	Indicates the slot duration for this particular slot. A value of 0 will cause the radio to use the computed time for the slot. An error is returned when 0 < Manual Slot Time < Computed Slot Time.
13	Computed Slot Duration (µs)	UINT32	Slot duration as computed by the radio. This time will be used if Manual Slot Time is set to 0.

4.19 RN_SET_REQUEST_USER_DATA_REQUEST (0x3003)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_SET_REQUEST_USER_DATA_CONFIRM (Radio)

Purpose: This message allows the Host to set the data buffer in the RangeNet range request packet. This data will be transmitted by the unit whenever it sends a range request packet. The buffer will remain in effect until changed by the Host. Upon boot this buffer will be empty. If the amount of data in the buffer is greater than the Max Request Data Size value in the RangeNet configuration, the buffer will be truncated.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_REQUEST_USER_DATA_REQUEST (0x3003)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Reserved	UINT16	Reserved
3	Data Size	UINT16	Number of bytes to include in each range response packet. These actual bytes follow. Maximum = 1000. When operating in Location mode the max value is 900. **Note:** the P4xx transmits 32bit (4byte) words. Any partial words will be zero-filled over the air but these bits will be removed upon reception. **Free Data:** Due to the waveform scan requiring a minimum number of pulses, a small amount of user data may be added without any packet time penalty. For RangeNet range requests, up to 12 bytes of user data may be added without increasing packet TX time (or 3 bytes if ELR is enabled).

#	Parameter	Type	Definition
4	Data	N*UINT8	Data to be sent with the range request packet. Extra bytes above <Data Size>, or <Max Request Data Size> from the RangeNet ALOHA configuration, whichever is less, will be ignored.

4.20 RN_SET_REQUEST_USER_DATA_CONFIRM (0x3103)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_SET_REQUEST_USER_DATA_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_SET_REQUEST_USER_DATA_REQUEST command. This response confirms the buffer was successfully written.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_REQUEST_USER_DATA_CONFIRM (0x3103)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.21 RN_GET_REQUEST_USER_DATA_REQUEST (0x300B)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_REQUEST_USER_DATA_CONFIRM (Radio)

Purpose: This message allows the Host to retrieve the data from the RangeNet range request data buffer. This data will be transmitted by the unit whenever it sends a range request packet. The buffer will remain in effect until changed by the Host. Upon boot this buffer will be empty.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_REQUEST_USER_DATA_REQUEST (0x300B)	UINT16	Message type

#	Parameter	Type	Definition
1	Message ID	UINT16	Associates request to confirm packets

4.22 RN_GET_REQUEST_USER_DATA_CONFIRM (0x310B)

API: RangeNet API
Message type: CONFIRM (Host)
Corresponding Message type: RN_GET_REQUEST_USER_DATA_REQUEST (Radio)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_GET_REQUEST_USER_DATA_REQUEST command. This response confirms the buffer was successfully retrieved.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_REQUEST_USER_DATA_CONFIRM (0x310B)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Reserved	UINT16	Reserved
3	Data Size	UINT16	Number of bytes in the range request buffer. Maximum = 1000. When operating in Location mode the max value is 900.
4	Data	(Data Size) * UINT8	Data from the range request buffer.

4.23 RN_SET_RESPONSE_USER_DATA_REQUEST (0x3004)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_SET_RESPONSE_USER_DATA_CONFIRM (Radio)

Purpose: This message allows the Host to set the data buffer in the RangeNet range response. This data will be transmitted by the unit whenever it sends a range response. The buffer will remain in effect until changed by the Host. Upon boot this buffer will be empty. If the amount of data in the buffer is greater than the Max Response Data Size value in the RangeNet configuration, the buffer will be truncated.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_RESPONSE_USER_DATA_REQUEST (0x3004)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Reserved	UINT16	Reserved
3	Data Size	UINT16	Number of bytes to include in each range response packet. These actual bytes follow. Maximum = 1000. When operating in Location mode the max value is 900. **Note:** the P4xx transmits 32bit (4byte) words. Any partial words will be zero-filled over the air but these bits will be removed upon reception.
4	Data	N*UINT8	Data to be sent with the range request packet. Extra bytes above <Data Size>, or <Max Response Data Size> from the RangeNet configuration, whichever is less, will be ignored.

4.24 RN_SET_RESPONSE_USER_DATA_CONFIRM (0x3104)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_SET_RESPONSE_USER_DATA_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_SET_RESPONSE_USER_DATA_REQUEST command. This response confirms the buffer was successfully written.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_RESPONSE_USER_DATA_CONFIRM (0x3104)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.25 RN_GET_RESPONSE_USER_DATA_REQUEST (0x300C)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_RESPONSE_USER_DATA_CONFIRM (Radio)

Purpose: This message allows the Host to retrieve the data from the RangeNet range response data buffer. This data will be transmitted by the unit whenever it sends a range response packet. The buffer will remain in effect until changed by the Host. Upon boot this buffer will be empty.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_RESPONSE_USER_DATA_REQUEST (0x300C)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

4.26 RN_GET_RESPONSE_USER_DATA_CONFIRM (0x310C)

API: RangeNet API
Message type: CONFIRM (Host)
Corresponding Message type: RN_GET_RESPONSE_USER_DATA_REQUEST

(Radio)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_GET_RESPONSE_USER_DATA_REQUEST command. This response confirms the buffer was successfully retrieved.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_RESPONSE_USER_DATA_CONFIRM (0x310C)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Reserved	UINT16	Reserved
3	Data Size	UINT16	Number of bytes in the range response buffer. Maximum = 1000. When operating in Location mode the max value is 900.
4	Data	(Data Size) * UINT8	Data from the range response buffer.

4.27 RN_GET_FULL_NEIGHBOR_DATABASE_REQUEST (0x3005)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_FULL_NEIGHBOR_DATABASE_CONFIRM (Radio)

Purpose: This message prompts the P4xx to send the Host a list of all known neighboring RangeNet nodes and various associated data. While the number of neighbors a unit might see has no limit, the number of neighbors which can be reported with this message is limited to 32. This limit is set by a buffer size constraint. While this constraint is arbitrary, it is currently a hard limit. If this becomes an issue, the user can either change the sorting arrangement to identify the closest 32 or the most recent 32.

If this is not satisfactory, then the user should consider using the RN_GET_SMALL_NEIGHBOR_DATABASE command. This command reports less data per neighbor and can therefore accommodate up to 80 neighbors. With this command it is also possible to sort by Node ID, range, or age.

While these are limits, they are rather generous limits and should satisfy most if not all applications. Users needing additional capacity should contact TDSR directly to discuss other options.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_FULL_NEIGHBOR_DATABASE_REQUEST (0x3005)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Number of Entries	UINT8	The maximum number of entries to be returned by the radio. Set to maximum of 32 to get all the entries.
3	Sort Type	UINT8	Order in which the entries will be listed; 0 = Sort by Node ID, lowest first 1 = Sort by Range, closest first 2 = Sort by Range Age, most recent first
4	Reserved	UINT16	Reserved; set to 0.

4.28 RN_GET_FULL_NEIGHBOR_DATABASE_CONFIRM (0x3105) / RN_FULL_NEIGHBOR_DATABASE_INFO (0x3203)

API: RangeNet API
Message type: CONFIRM (Radio) / INFO (Radio)
Corresponding Message type: RN_GET_FULL_NEIGHBOR_DATABASE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_GET_FULL_NEIGHBOR_DATABASE_REQUEST command. This message may also be sent automatically (with message type RN_FULL_NEIGHBOR_DATABASE_INFO) if the Autosend flags and interval are appropriately configured in the RangeNet configuration. This response provides a list of the known neighboring RangeNet nodes as well as various associated data. This message contains entries for 32 neighbors. The first <NumNodes> entries are filled and the rest zeroed out.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_FULL_NEIGHBOR_DATABASE_CONFIRM (0x3105) / RN_FULL_NEIGHBOR_DATABASE_INFO (0x3203)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Number of Nodes	UINT8	Number of filled entries in the neighbor list.
3	Sort Type	UINT8	Order in which the entries are listed; 0 = Sorted by Node ID, lowest first 1 = Sorted by Range, closest first 2 = Sorted by Range Age, most recent first
4	Reserved	UINT16	Reserved
5	Timestamp	UINT32	This is a snapshot of milliseconds from boot to time the Neighbor Database is sent.
6	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.
	BEGINNING OF NEIGHBOR LIST	32X	The following section is repeated 32 times to support a list of up to 32 neighbor nodes. Only the first <NumNodes> entries are filled; the remaining entries are zeroed out.
0	Responder ID	UINT32	Node ID of the UWB module that sent the range response.

#	Parameter	Type	Definition
1	Range Status	UINT8	0 – PRM Range Success 64 – Coarse Range Estimate (CRE) Other values – Reserved
2	Antenna Mode	UINT8	Specifies the antenna ports used during this range conversation. The lower nibble describes the antenna configuration used by the requester, and the upper nibble the antenna configuration used by the responder. Valid values for each nibble are: 0 = Transmit and Receive on the A port 1 = Transmit and Receive on the B port 2 = TX on A, RX on B 3 = TX on B, RX on A.
3	Stopwatch Time	UINT16	Duration of the range conversation in milliseconds.
4	Precision Range Measurement (PRM) / Filtered Range Estimate (FRE) (See Range Measurement Type)	UINT32	PRM is the unfiltered, precise distance in millimeters between UWB modules based on a Two-Way Time-of-Flight (TW-TOF) measurement. FRE is the filtered distance in millimeters based on the combination of PRM and CRE values passed through a recursive optimal Kalman estimator with two state variables (r & rdot). If CREs are not used, the FRE will still be generated and will use the Kalman estimator and the PRM. FREs can provide higher update rates when used in a network. The increase in update rate comes at the cost of higher range error and increased latency. This is an advanced feature may benefit some networks. However, we strongly recommend that users focus on using PRMs and avoid FREs. Those interested in advanced networks should contact TDSR to discuss whether or not consider using FREs.

#	Parameter	Type	Definition
5	PRM Error (PRME) / FRE Error (FREE) (See Range Measurement Type)	UINT16	PRME is the estimated standard deviation error, in millimeters, of associated PRM value. Estimated from pulsed waveform signature. FREE is the estimated standard deviation error, in millimeters, of associated FRE value. Derived by the filter.
6	Filtered Range Velocity (FRV)	INT16	Estimated radial velocity in millimeters per second. Only valid if Range Measurement Type = 4, otherwise it is 0.
7	Range Measurement Type	UINT8	Specifies the type of the range measurement for this responder. 1 = Precision Range Measurement (PRM) 2 = 0 (Reserved) 4 = Filtered Range Estimate (FRE)
8	Flags	UINT8	Bit flags for various parameters: • Bit 0 – Beacon mode: 0 = Normal mode, 1 = Beacon mode • Bit 1 – Do not range to me: 0 = Normal, 1 = Do not range to me • Bit 2 – In Exclusion List: 0 = Not excluded, 1 = Excluded • Bit 3 – Range Uncalibrated: 0 = Normal, 1 = Uncalibrated • Bits 4:7 – Reserved
9	Requester LED Flags OR'd with Responder LED Flags	UINT16	These characteristics refer to the requester's and responder's received scan: 1 = SATURATED 8 = LINE OF SIGHT (LOS) 16 = NON-LINE OF SIGHT (NLOS)
10	Noise	UINT16	The noise measured by the radio. SNR can be computed using; 20 * Log_{10} (Vpeak / Noise).
11	Vpeak	UINT16	The absolute maximum value in the leading edge window of the received waveform. This value is used to determine the Coarse Range Estimate.

#	Parameter	Type	Definition
12	Num Range Attempts	UINT16	A count of the number of times this radio has attempted to perform a precision range to this neighbor.
13	Num Precision Range Successes	UINT16	A count of the number of successful precision ranges to this neighbor.
14	Statistics Time	UINT32	The duration in milliseconds over which the previous two fields were calculated.
15	Range Update Timestamp	UINT32	Time since radio boot, in milliseconds, since this neighbor list entry's range was updated.
16	Last Heard Timestamp	UINT32	Time since radio boot, in milliseconds, since this neighbor was last heard over UWB.
17	Added to NDB Timestamp	UINT32	Time since radio boot, in milliseconds, when the neighbor was added to the NDB. Used to determine which uncalibrated node to range to next.
	END OF NEIGHBOR LIST		The preceding section is repeated 32 times to support a list of up to 32 neighbor nodes. Only the first <NumNodes> entries are filled; the remaining entries are zeroed out.

4.29 RN_GET_SMALL_NEIGHBOR_DATABASE_REQUEST (0x3006)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_SMALL_NEIGHBOR_DATABASE_CONFIRM (Radio)

Purpose: This message prompts the P4xx to send the Host a list of all known neighboring RangeNet nodes and various associated data up to a limit of 80 units. See the RN_GET_FULL_NEIGHBOR_DATABASE_REQUEST for additional discussion on limits and options.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_SMALL_NEIGHBOR_DATABASE_REQUEST (0x3006)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

#	Parameter	Type	Definition
2	Number of Entries	UINT8	The maximum number of entries to be returned by the radio. Set to maximum of 80 to get all the entries.
3	Sort Type	UINT8	Order in which the entries will be listed: 0 = Sort by Node ID, lowest first 1 = Sort by Range, closest first 2 = Sort by Range Age, most recent first
4	Reserved	UINT16	Reserved; set to 0.

4.30 RN_GET_SMALL_NEIGHBOR_DATABASE_CONFIRM (0x3106) / RN_SMALL_NEIGHBOR_DATABASE_INFO (0x3204)

API: RangeNet API
Message type: CONFIRM (Radio) / INFO (Radio)
Corresponding Message type: RN_GET_SMALL_NEIGHBOR_DATABASE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_GET_SMALL_NEIGHBOR_DATABASE_REQUEST command. This message may also be sent automatically (with message type RN_SMALL_NEIGHBOR_DATABASE_INFO) if the Autosend flags and interval are appropriately configured in the RangeNet configuration.

This message is designed to be a minimal form of the Neighbor Database, especially for transferring over serial data lines where bandwidth may be precious. As such, note that unlike other RCM/RangeNet messages which are designed to be friendly to 4-byte alignments, this message uses 1-byte alignment (i.e., a 32-bit field might begin on a boundary that is not a 32-bit boundary). Also, this is a variable-length packet whose length is based on the number of neighbors listed.

This response provides a list of the known neighboring RangeNet nodes as well as various associated data. This message contains entries for up to 80 neighbors.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_SMALL_NEIGHBOR_DATABASE_CONFIRM (0x3106) / RN_SMALL_NEIGHBOR_DATABASE_INFO (0x3204)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

#	Parameter	Type	Definition
2	Number of Nodes	UINT8	Number of filled entries in the neighbor list.
3	Sort Type	UINT8	Order in which the entries are listed: 0 = Sorted by Node ID, lowest first 1 = Sorted by Range, closest first 2 = Sorted by Range Age, most recent first
4	Reserved	UINT16	Reserved
	BEGINNING OF NEIGHBOR LIST	Up to 80X	The following section is repeated up to 80 times to support a list of up to 80 neighbor nodes.
0	Node ID	UINT32	Node ID of this RangeNet neighbor.
1	Range	UINT16	PRM or FRE, in centimeters, to this RangeNet neighbor. Type depends on setting in RangeNet Configuration.
2	PRM Error (PRME) / FRE Error (FREE) (See Range Measurement Type)	UINT8	PRME is the estimated standard deviation error, in millimeters, of associated PRM value. Estimated from pulsed waveform signature. FREE is the estimated standard deviation error, in millimeters, of associated FRE value. Derived by the filter.
3	Reserved	UINT8	Reserved
4	Age	UINT16	Age, in milliseconds, since this neighbor list entry’s range was updated.
5	Range Measurement Type	UINT8	Specifies the valid components of this message. 1 = Precision Range Measurement (PRM) 2 = 0 (Reserved) 4 = Filtered Range Estimate (FRE)

#	Parameter	Type	Definition
6	Flags	UINT8	Bit flags for various parameters: • Bit 0 – Beacon mode: 0 = Normal mode, 1 = Beacon mode • Bit 1 – Do not range to me: 0 = Normal, 1 = Do not range to me • Bit 2 – In Exclusion List: 0 = Not excluded, 1 = Excluded • Bit 3 – Range Uncalibrated: 0 = Normal, 1 = Uncalibrated • Bits 4:7 – Reserved
	END OF NEIGHBOR LIST		The preceding section is repeated up to 80 times to support a list of up to 80 neighbor nodes.

4.31 RN_SET_EXCLUDED_REQUEST (0x3007)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_SET_EXCLUDED_CONFIRM (Radio)

Purpose: This message allows the Host to configure the P4xx's exclusion list. The exclusion list is a user defined list of Node IDs that the P4xx will not range to and exclude from its Neighbor Database. Note that this is a variable length packet dependent on the number of nodes in the exclusion list. The maximum of number of nodes that can be in the exclusion list is 256.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_EXCLUDED_REQUEST (0x3007)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Number of Nodes	UINT8	Number of Node IDs in the exclusion list. Maximum is 256.
3	Reserved	UINT8	Reserved
4	Reserved	UINT16	Reserved

#	Parameter	Type	Definition
5	Exclusion List	[UINT32]	This is a variable length array with each element containing a Node ID. The array only needs to be as long as the indicated number of Nodes with a maximum number of elements set at 256.

4.32 RN_SET_EXCLUDED_CONFIRM (0x3107)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_SET_EXCLUDED_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_SET_EXCLUDED_REQUEST command. This response confirms the exclusion list was successfully written.

Packet Definition:

#	Parameter	Type	Definition
0	RN_SET_EXCLUDED_CONFIRM (0x3107)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.33 RN_GET_EXCLUDED_REQUEST (0x3008)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_EXCLUDED_CONFIRM (Radio)

Purpose: This message is sent by the Host to the P4xx to request the exclusion list. The list will be sent back by the P4xx in the corresponding RN_GET_EXCLUDED_CONFIRM message.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_EXCLUDED_REQUEST (0x3008)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

4.34 RN_GET_EXCLUDED_CONFIRM (0x3108)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_GET_EXCLUDED_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_GET_EXCLUDED_REQUEST command. The exclusion list is a user-defined list of Node IDs that the P4xx will not range to and exclude from its Neighbor Database. Note that this is a variable length packet dependent on the number of nodes in the exclusion list. The maximum of number of nodes that can be in the exclusion list is 256.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_EXCLUDED_CONFIRM (0x3108)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Number of Nodes	UINT8	Number of Node IDs in the exclusion list. Maximum is 256.
3	Reserved	UINT8	Reserved
4	Reserved	UINT16	Reserved
5	Exclusion List	[UINT32]	This is a variable length array with each element containing a single Node ID. The number of elements is dictated by the Number of Nodes included in the packet.

4.35 RN_GET_HEALTH_STATUS_REQUEST (0x3009)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_HEALTH_STATUS_CONFIRM (Radio)

Purpose: This message is sent by the Host to the P4xx to request the RangeNet health statistics. The statistics will be sent back by the P4xx in the corresponding RN_GET_HEALTH_STATUS_CONFIRM message.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_HEALTH_STATUS_REQUEST (0x3009)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

4.36 RN_GET_HEALTH_STATUS_CONFIRM (0x3109)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_GET_HEALTH_STATUS_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_GET_HEALTH_STATUS_REQUEST command. The health statistics include metrics from the Neighbor Database as well as ranging transactions.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_HEALTH_STATUS_CONFIRM (0x3109)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages.
2	Temperature	UINT32	Temperature reading from PCB sensor. Divide by 4 to get degrees Celsius.
3	Number of Neighbors	UINT32	Number of neighbors listed in the Neighbor Database.
4	Statistics Time	UINT32	The time is milliseconds the statistics represent.
5	Number of Range Attempts	UINT32	Number of range attempts.
6	Number of PRMs	UINT32	Number of successful Precision Range Measurements.
7	Number of CREs	UINT32	Number of successful Coarse Range Estimates.
8	Number of Timeouts	UINT32	Number of timeout errors.
9	Number of VCSs	UINT32	Number of Virtual Carrier Sense detections.
10	Number of LED Failures	UINT32	Number of LED failures.

#	Parameter	Type	Definition
11	Number of CCI Failures	UINT32	Number of failures due to Co-channel Interference.

4.37 RN_RESET_DATABASE_AND_STATS_REQUEST (0x300A)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_RESET_DATABASE_AND_STATS_CONFIRM (Radio)

Purpose: This message is sent by the Host to the P4xx to reset various neighbor database quantities.

Packet Definition:

#	Parameter	Type	Definition
0	RN_RESET_DATABASE_AND_STATS_REQUEST (0x300A)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Reset Flags	UINT32	Flags to control what information is reset in the neighbor database: Bit 0 – Reset Database: 0 = No behavior, 1 = removes the specified neighbors from the database. Bit 1 – Clears all health statistics. Specifying Node ID has no relevance. Bit 2 – Reset Statistics: 0 = No behavior, 1 = Clears the statistics for the specified nodes in the database. Bits 3:31 – Reserved. Set to 0.
3	Node ID	UINT32	Node ID of the neighbor to reset, or 0 to reset all nodes in the Neighbor Database.

4.38 RN_RESET_DATABASE_AND_STATS_CONFIRM (0x310A)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_RESET_DATABASE_AND_STATS_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_RESET_DATABASE_AND_STATS_REQUEST command.

Packet Definition:

#	Parameter	Type	Definition
0	RN_RESET_DATABASE_AND _STATS_CONFIRM (0x310A)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Status	UINT32	0 = Successful. For error codes see Table 4-1 at the end of this section.

4.39 RN_GET_PACKET_DURATIONS_REQUEST (0x300F)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: RN_GET_PACKET_DURATIONS_CONFIRM (Radio)

Purpose: This message is sent by the Host to the P4xx to request the packet timing associated with a range transaction. The values will be sent back by the P4xx in the corresponding RN_GET_PACKET_DURATIONS_CONFIRM message.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_PACKET_DURATIONS_ REQUEST (0x300F)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

4.40 RN_GET_PACKET_DURATIONS_CONFIRM (0x310F)

API: RangeNet API
Message type: CONFIRM (Radio)
Corresponding Message type: RN_GET_PACKET_DURATIONS_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in immediate response to a RN_GET_PACKET_DURATIONS_REQUEST command. Several factors impact packet duration including the Pulse Integration Index (PII), request and response data, the ELR flag, and processing time.

Packet Definition:

#	Parameter	Type	Definition
0	RN_GET_PACKET_DURATIONS_CONFIRM (0x310F)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages.
2	Request with no Data	UINT32	The amount of time in microseconds for the request packet with no data included.
3	Response with no Data	UINT32	The amount of time in microseconds for the response packet with no data included.
4	Request with User Data	UINT32	The amount of time in microseconds for the request packet with user data included.
5	Response with User Data	UINT32	The amount of time in microseconds for the response packet with user data included.
6	Conversation Time with no Data	UINT32	The amount of time in microseconds for the entire range conversation with no data included.
7	Conversation Time with User Data	UINT32	The amount of time in microseconds for the entire range conversation with user data included.
8	Data Packet with no Data	UINT32	The amount of time in microseconds to send a data packet with no user data included.
9	Data Packet with User Data	UINT32	The amount of time in microseconds to send a data packet with user data included.

Table 4-1: CONFIRM Message Status Codes

Code	Status	Description
0	Success	The REQUEST message was processed successfully
1	Generic Failure	Catch-all for uncategorized failures
2	Wrong Op Mode	The REQUEST message cannot be acted upon in the current op mode.
3	Unsupported Value	The REQUEST message contained an unsupported value in one or more of its fields.
4	Invalid During Sleep	The REQUEST message cannot be acted upon in the current sleep mode.
5	Wrong Message Size	The number of bytes in the REQUEST message did not match the expected number of bytes for the message type.
6	Not Enabled	The feature used by the REQUEST message is currently disabled.
7	Wrong Buffer Size	The specified size of a buffer in the REQUEST message, or the size of the buffer itself, did not match the expected number of bytes for the message type.
8	Unrecognized Message Type	The REQUEST Message Type was not recognized.
0x80000000	Internal Error Code	An internal error code was generated. This status is OR'ed with the internal error code itself and should be used in communication with TDSR’s support team.

5 Location API Messages

5.1 LOC_SET_CONFIG_REQUEST (0x5001)

API: Location API
Message type: REQUEST (Host)
Corresponding Message type: LOC_SET_CONFIG_CONFIRM (Radio)

Purpose: This message configures Location parameters and advanced solver settings.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_SET_CONFIG_REQUEST (0x5001)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Flags	UINT16	Bits 0 – 1 : Controls how the Location Info message is sent from the radio to the Host 0 : None – disables the sending of Location Info messages to the Host 1 : Successful – only sends those messages with a Solver Error Code of 0 (successful) 2 : All – sends all Location Info messages regardless of Solver Error Code Bits 2 – 3 : Controls how the Range Info message is sent from the radio to the Host 0 : None – disables the sending of the Range Info message to the Host 1 : Successful – only sends those message with a Status of 0 (successful) 2 : All – sends all Range Info messages regardless of the Status Bit 4 : Report ELLs – enabling this flag will cause the radio to send received ELL or ELLEX messages to the Host Bit 5 : Report ELRs – enabling this flag will cause the radio to send received ELR messages to the Host Bits 6-8 : Reserved; set to 0 Bit 9 : Allow non-line-of-sight ranges into the solver 0 : NLOS ranges are not passed to the solver (line-of-sight ranges only) (default)

#	Parameter	Type	Definition
			1 : NLOS ranges are passed to the solver. Bits 10 – 11 : Selects the network timing mode 0 : ALOHA – the RangeNet ALOHA transmit scheduler is used (default) 1 : TDMA – the RangeNet TDMA slotmap set up by the user is used for transmit timing. The user must take care to set up the slotmap for the needs of the Location activity (e.g. slots may need to be longer to accommodate ELL, ELR, etc.). For a fixed installation with a well-designed slotmap, this mode can offer the most optimized network timing. 2 : TDMA Auto – the RangeNet TDMA timing infrastructure is used, but with an internal slotmap that's automatically generated from the Location Map. This mode is generally higher performance than ALOHA, but without the need for manual configuration in vanilla TDMA. Bits 12 – 13 : Selects the Solver mode used for tracking mobiles. For 2D tracking, the system needs the minimum of 3 anchors. For 3D tracking, the system needs a minimum of 4 anchors. 0 : Kalman 2D – the 2D geometric solver is used for initialization, and afterwards a 2D Kalman filter is used for tracking. (default) 1 : Geometric 2D – the 2D geometric solver is used for tracking. 2 : Kalman 3D – the 3D geometric solver is used for initialization, and afterwards a 3D Kalman filter is used for tracking. 3 : Geometric 3D – the 3D geometric solver is used for tracking. Bits 14-15 : Reserved; set to 0

#	Parameter	Type	Definition
3	Boot Mode	UINT8	Sets the Location Operation Mode the radio will start in on power-up or reboot. Note that the radio will ignore this setting if its boot-up mode is already set to Ranging or Networking. 0 : Idle – the radio will start in the Idle Mode. 1 : Autosurvey – the radio will start in the Autosurvey Mode. In this mode radios defined as Anchors in the Location Map will range to each other and Mobile radios will remain idle. 2 : Tracking – the radio will start in the Tracking Mode. This is the operational tracking mode where radios defined as Mobiles in the Location Map will range to Anchors.
4	Reserved	UINT8	Set to 0
5	Reserved	UINT16	Set to 0
6	Solver Max REE	UINT16	Maximum Range Error Estimate (REE) in millimeters before the range is rejected. Set to 0 to disable filter. Default is 100 mm.
7	Solver Max GDOP	UINT16	Maximum GDOP of the location solution before the system triggers rebirth. The value passed to and from the radio is a scaled integer and should be divided by 100 to relate to notional GDOP values. Default is 4.0.
8	GDOP Anchor History Depth	UINT8	History depth for Anchors used in determining GDOP. For 2D tracking, in general it should be set to 6 for systems with 3 Anchors and 4 for systems with 4 or more Anchors. For 3D tracking, which requires at least 4 anchors, in general it should be set to 8 or more. Valid values are 4 (min) to 32 (max). Default is 4.
9	Solver NLS to Kalman Updates	UINT8	The number of successful nonlinear-least-squares solver (geometric solver) updates that must occur before transitioning to the Kalman solver. Default is 4.

#	Parameter	Type	Definition
10	Kalman Sigma Accel	UINT16	Kalman model confidence weight relative to measurement weight. Set to 100 to give equal weight to both the Kalman model and to range measurements. A smaller value indicates more confidence in the Kalman model (which results in a greater filtering effect of the positions); a larger value indicates more confidence in the range measurements (which results in slightly faster convergence to new positions). Set to 0 to use default value of 100.
11	NLS Solver Output Boxcar Filter Depth	UINT8	Moving average boxcar filter depth for Geometric solver modes. Use 0 (or 1) to disable the boxcar filter. Valid values are 0 to 64. Default is 4.
12	Reserved	UINT8	Set to 0
13	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.
14	Reserved	UINT8	Set to 0
15	Reserved	UINT16	Set to 0

5.2 LOC_SET_CONFIG_CONFIRM (0x5101)

API: Location API

Message type: CONFIRM (Radio)
Corresponding Message type: LOC_SET_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a LOC_SET_CONFIG_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of the LOC_SET_CONFIG_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_SET_CONFIG_CONFIRM (0x5101)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 5-1 at the end of this section.

5.3 LOC_GET_CONFIG_REQUEST (0x5002)

API: RangeNet API
Message type: REQUEST (Host)
Corresponding Message type: LOC_GET_CONFIG_CONFIRM (Radio)

Purpose: This is a request message sent by the Host to P4xx to request a copy of the current Location configuration.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_GET_CONFIG_REQUEST (0x5002)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

5.4 LOC_GET_CONFIG_CONFIRM (0x5102)

API: Location API
Message type: CONFIRM (Radio)
Corresponding Message type: LOC_GET_CONFIG_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a LOC_GET_CONFIG_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of the LOC_GET_CONFIG_REQUEST and return the Location configuration.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_GET_CONFIG_CONFIRM (0x5102)	UINT16	Message type
1	Message ID	UINT16	Associates request to request and info messages
2	Flags	UINT16	Bits 0 – 1 : Controls how the Location Info message is sent from the radio to the Host 0 : None – disables the sending of Location Info messages to the Host 1 : Successful – only sends those messages with a Solver Error Code of 0 (successful) 2 : All – sends all Location Info messages regardless of Solver Error Code Bits 2 – 3 : Controls how the Range Info message is sent from the radio to the Host 0 : None – disables the sending of the Range Info message to the Host 1 : Successful – only sends those message with a Status of 0 (successful) 2 : All – sends all Range Info messages regardless of the Status Bit 4 : Report ELLs – enabling this flag will cause the radio to send received ELL or ELLEX messages to the Host Bit 5 : Report ELRs – enabling this flag will cause the radio to send received ELR messages to the Host Bits 6-8 : Reserved; set to 0 Bit 9 : Allow non-line-of-sight ranges into the solver 0 : NLOS ranges are not passed to the solver (line-of-sight ranges only) (default) 1 : NLOS ranges are passed to the solver.

#	Parameter	Type	Definition
			Bits 10 – 11 : Selects the network timing mode 0 : ALOHA – the RangeNet ALOHA transmit scheduler is used (default) 1 : TDMA – the RangeNet TDMA slotmap set up by the user is used for transmit timing. The user must take care to set up the slotmap for the needs of the Location activity (e.g. slots may need to be longer to accommodate ELL, ELR, etc.). For a fixed installation with a well-designed slotmap, this mode can offer the most optimized network timing. 2 : TDMA Auto – the RangeNet TDMA timing infrastructure is used, but with an internal slotmap that's automatically generated from the Location Map. This mode is generally higher performance than ALOHA, but without the need for manual configuration in vanilla TDMA. Bits 12 – 13 : Selects the Solver mode used for tracking mobiles. For 2D tracking, the system needs the minimum of 3 anchors. For 3D tracking, the system needs a minimum of 4 anchors. 0 : Kalman 2D – the 2D geometric solver is used for initialization, and afterwards a 2D Kalman filter is used for tracking. (default) 1 : Geometric 2D – the 2D geometric solver is used for tracking. 2 : Kalman 3D – the 3D geometric solver is used for initialization, and afterwards a 3D Kalman filter is used for tracking. 3 : Geometric 3D – the 3D geometric solver is used for tracking. Bits 14-15 : Reserved; set to 0

#	Parameter	Type	Definition
3	Boot Mode	UINT8	Sets the Location Operation Mode the radio will start in on power-up or reboot. Note that the radio will ignore this setting if its boot-up mode is already set to Ranging or Networking. 0 : Idle – the radio will start in the Idle Mode. 1 : Autosurvey – the radio will start in the Autosurvey Mode. In this mode radios defined as Anchors in the Location Map will range to each other and Mobile radios will remain idle. 2 : Tracking – the radio will start in the Tracking Mode. This is the operational tracking mode where radios defined as Mobiles in the Location Map will range to Anchors.
4	Reserved	UINT8	Reserved
5	Reserved	UINT16	Reserved
6	Solver Max REE	UINT16	Maximum Range Error Estimate (REE) in millimeters before the range is rejected. Set to 0 to disable filter. Default is 100 mm.
7	Solver Max GDOP	UINT16	Maximum GDOP of the location solution before the system triggers rebirth. The value passed to and from the radio is a scaled integer and should be divided by 100 to relate to notional GDOP values. Default is 4.0.
8	GDOP Anchor History Depth	UINT8	History depth for Anchors used in determining GDOP. For 2D tracking, in general it should be set to 6 for systems with 3 Anchors and 4 for systems with 4 or more Anchors. For 3D tracking, which requires at least 4 anchors, in general it should be set to 8 or more. Valid values are 4 (min) to 32 (max). Default is 4.
9	Solver NLS to Kalman Updates	UINT8	The number of successful nonlinear-least-squares solver (geometric solver) updates that must occur before transitioning to the Kalman solver. Default is 4.

#	Parameter	Type	Definition
10	Kalman Sigma Accel	UINT16	Kalman model confidence weight relative to measurement weight. Set to 100 to give equal weight to both the Kalman model and to range measurements. A smaller value indicates more confidence in the Kalman model (which results in a greater filtering effect of the positions); a larger value indicates more confidence in the range measurements (which results in slightly faster convergence to new positions). Set to 0 to use default value of 100.
11	NLS Solver Output Boxcar Filter Depth	UINT8	Moving average boxcar filter depth for Geometric solver modes. Use 0 (or 1) to disable the boxcar filter. Valid values are 0 to 64. Default is 4.
12	Reserved	UINT8	Reserved
13	Timestamp	UINT32	Milliseconds from boot to time of message sent
14	Status	UINT32	0 = Successful. For error codes see Table 5-1 at the end of this section.

5.5 LOC_SET_MODE_REQUEST (0x5003)

API: Location API
Message type: REQUEST (Host)
Corresponding Message type: LOC_SET_MODE_CONFIRM (Radio)

Purpose: This message sets the Location Operation Mode.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_SET_MODE_REQUEST (0x5003)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages

#	Parameter	Type	Definition
2	Mode	UINT8	Sets the current Location Operation Mode 0 : Idle – the purpose of the Idle mode is minimize RF and interface traffic so there is minimal chance of interference when sending configuration messages down to the radio and over the air via the Broadcast flag. In this mode Anchors will still beacon so the Mobiles will keep them in their Neighbor Database but will not pass Info messages to the Host. Mobiles sit idle and will not send range requests or Info messages to the Host. Radios will still report Info messages pertaining to Location configuration changes 1 : Autosurvey – in this mode radios defined as Anchors in the Location Map will range to each other and Mobile radios will remain idle. This mode is primarily for the RangeNet GUI to collect the necessary information for Autosurvey (or for the user to implement their own Autosurvey). 2 : Tracking – this is the operational tracking mode where radios defined as Mobiles in the Location Map will range to Anchors.
3	Broadcast Flag	UINT8	If the Broadcast Flag is set, the Mode will be broadcast so that all radios in the system will change Modes as well. A flood routing algorithm is used so that radios that can't hear the originator but hear a radio who heard the broadcast will also change their Mode. Since Mode changes affect which radios transmit, the units synchronize the Mode change to occur simultaneously 2 to 4 seconds after this message is first sent.
5	Reserved	UINT16	Set to 0

5.6 LOC_SET_MODE_CONFIRM (0x5103)

API: Location API
Message type: CONFIRM (Radio)
Corresponding Message type: LOC_SET_MODE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a LOC_SET_MODE_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of the LOC_SET_MODE_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_SET_MODE_CONFIRM (0x5103)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Mode	UINT8	Used to confirm the mode change 0 : Idle 1 : Autosurvey 2 : Tracking
3	Reserved	UINT8	Reserved
4	Reserved	UINT16	Reserved
5	Status	UINT32	0 = Successful. For error codes see Table 5-1 at the end of this section.

5.7 LOC_GET_MODE_REQUEST (0x5004)

API: Location API
Message type: REQUEST (Host)
Corresponding Message type: LOC_GET_MODE_CONFIRM (Radio)

Purpose: This message gets the Location Operation Mode.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_GET_MODE_REQUEST (0x5004)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages

5.8 LOC_GET_MODE_CONFIRM (0x5104)

API: Location API
Message type: CONFIRM (Radio)
Corresponding Message type: LOC_GET_MODE_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a LOC_GET_MODE_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of the LOC_GET_MODE_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_GET_MODE_CONFIRM (0x5104)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

#	Parameter	Type	Definition
2	Mode	UINT8	The current Location Operation Mode 0 : Idle – the purpose of the Idle Mode is minimize RF and interface traffic so there is minimal chance of interference when sending configuration messages down to the radio and over the air via the Broadcast flag. In this mode Anchors will still beacon so the Mobiles will keep them in their Neighbor Database but will not pass Info messages to the Host. Mobiles sit idle and will not send range requests or Info messages to the Host. Radios will still report Info messages pertaining to Location configuration changes. 1 : Autosurvey – in this mode radios defined as Anchors in the Location Map will range to each other and Mobile radios will remain idle. This mode is primarily for the RangeNet GUI to collect the necessary information for Autosurvey (or for the user to implement their own Autosurvey function). 2 : Tracking – this is the operational tracking mode where radios defined as Mobiles in the Location Map will range to Anchors.
3	Reserved	UINT8	Reserved
4	Reserved	UINT16	Reserved

5.9 LOC_SET_LOCATION_MAP_REQUEST (0x5005)

API: Location API
Message type: REQUEST (Host)
Corresponding Message type: LOC_SET_LOCATION_MAP_CONFIRM (Radio)

Purpose: This message sets the Location Map in the P4xx. The Location Map defines the roles and message parameters for radios in the location system. It also has an option for broadcasting the Location Map to other P4xx's.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_SET_LOCATION_MAP_REQUEST (0x5005)	UINT16	Message type

#	Parameter	Type	Definition
1	Message ID	UINT16	Associates request to confirm and info messages
2	Reserved	UINT8	Set to 0
3	Broadcast Flag	UINT8	If the Broadcast Flag is set, the Location Map will be broadcast so that all radios in the system will have the updated map. A flood routing algorithm is used so that radios that can’t hear the originator but hear a radio who heard the broadcast will also be able to receive the map.
4	Number of Entries	UINT8	Number of entries in the Location Map. This is a variable-length message, so only the populated entries need be included. Maximum of 60.
5	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag’s setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.
BEGIN LOCATION MAP ENTRIES The maximum number of Location Map Entries is 60			
0	Node ID	UINT32	The Node ID of the radio for this entry

#	Parameter	Type	Definition
1	Node Type	UINT8	Specifies the role of the radio in the system. 0 : Mobile – the radio that ranges to Anchors and computes its own position. 1 : Anchor – the radio with known, fixed location. If not being actively ranged to, Anchors will send out a beacon so that Mobiles know they are available. 2 : Origin – if using Autosurvey, this Anchor type specifies that the Anchor is the origin in the relative coordinate system. 3 : +X Axis – if using Autosurvey, this Anchor type specifies that the line from the origin to this Anchor represents the +X axis. 4 : -X Axis – if using Autosurvey, this Anchor type specifies that the line from the origin to this Anchor represents the –X axis. 5 : +Y Axis – if using Autosurvey, this Anchor type specifies that the Y axis is in the positive direction relative to the X axis and that this Anchor will be placed in the +Y plane. 6 : -Y Axis – if using Autosurvey, this Anchor type specifies that the Y axis is in the negative direction relative to the X axis and that this Anchor will be placed in the -Y plane. 7 : Z Axis – if using Autosurvey, this Anchor type specifies an additional Anchor to be surveyed and is not currently used to designate any part of the coordinate system. Its name as the Z Axis is a place holder for 3D implementation planned in a later software release.
2	Reserved	UINT8	Set to 0

#	Parameter	Type	Definition
3	Flags	UINT8	Bits 0 – 1 : Sets sending the ELL or ELLEX information. Only relevant for Mobiles since Anchors do not compute positions. 0 : None – neither the ELL or ELLEX information is sent 1 : ELL – the Mobile will send the last position computed on its next range request 2 : ELLEX – the Mobile will send the last position along with the variance and covariance information in its next range request. Essentially includes the same information as the Location Info message Bit 2 : Enables ELR where the Mobile will include its last range measurement in its next range request. Only relevant for Mobiles in Tracking Mode. Anchors in Tracking Mode do not range at all, and in Autosurvey Mode, Anchors automatically echo their last range measurements, while Mobiles do not range at all.
4	Reserved	UINT8	Set to 0
5	Reserved	UINT16	Set to 0
6	Beacon Interval	UINT16	The average ALOHA transmit time in milliseconds. Set to 0 to enable Automatic Congestion Control (ACC). For Anchors it represents the average beacon interval and for Mobiles it represents the average ranging time. Note that with ALOHA the radio will generate a random hold-off such that the minimum time is the amount of time it takes to measure a range, and the longest time is ((2 * Beacon Interval) – Ranging Time). Ignored in TDMA modes.
7	X Location	INT32	The X location of the radio in millimeters
8	Y Location	INT32	The Y location of the radio in millimeters
9	Z Location	INT32	The Z location of the radio in millimeters
END LOCATION MAP ENTRIES			

5.10 LOC_SET_LOCATION_MAP_CONFIRM (0x5105)

API: Location API
Message type: CONFIRM (Radio)

Corresponding Message type: LOC_SET_LOCATION_MAP_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a LOC_SET_LOCATION_MAP_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of the LOC_SET_LOCATION_MAP_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_SET_LOCATION_MAP_CONFIRM (0x5105)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets
2	Status	UINT32	0 = Successful. For error codes see Table 5-1 at the end of this section.

5.11 LOC_GET_LOCATION_MAP_REQUEST (0x5006)

API: Location API
Message type: REQUEST (Host)
Corresponding Message type: LOC_GET_LOCATION_MAP_CONFIRM (Radio)

Purpose: This message gets the Location Map from the P4xx. The Location Map defines the roles and message parameters for radios in the location system.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_GET_LOCATION_MAP_REQUEST (0x5006)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm packets

5.12 LOC_GET_LOCATION_MAP_CONFIRM (0x5106)

API: Location API
Message type: CONFIRM (Radio)
Corresponding Message type: LOC_GET_LOCATION_MAP_REQUEST (Host)

Purpose: This message is sent by the P4xx to the Host in response to a LOC_GET_LOCATION_MAP_REQUEST message previously received by the P4xx from the Host. Its purpose is to confirm successful operation of the LOC_GET_LOCATION_MAP_REQUEST.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_GET_LOCATION_MAP_ CONFIRM (0x5106)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Number Locations	UINT8	Number of entries in the Location Map. Only the populated entries are sent in this variable-length message.
3	Reserved	UINT8	Reserved
4	Reserved	UINT16	Reserved
5	Status	UINT32	0 = Successful. For error codes see Table 5-1 at the end of this section.
BEGIN LOCATION MAP ENTRIES The maximum number of Location Map Entries is 60			
0	Node ID	UINT32	The Node ID of the radio for this entry

#	Parameter	Type	Definition
1	Node Type	UINT8	Specifies the role of the radio in the system. 0 : Mobile – the radio that ranges to Anchors and computes its own position. 1 : Anchor – the radio with known, fixed location. If not being actively ranged to, Anchors will send out a beacon so that Mobiles know they are available. 2 : Origin – if using Autosurvey, this Anchor type specifies that the Anchor is the origin in the relative coordinate system. 3 : +X Axis – if using Autosurvey, this Anchor type specifies that the line from the origin to this Anchor represents the +X axis. 4 : -X Axis – if using Autosurvey, this Anchor type specifies that the line from the origin to this Anchor represents the –X axis. 5 : +Y Axis – if using Autosurvey, this Anchor type specifies that the Y axis is in the positive direction relative to the X axis and that this Anchor will be placed in the +Y plane. 6 : -Y Axis – if using Autosurvey, this Anchor type specifies that the Y axis is in the negative direction relative to the X axis and that this Anchor will be placed in the -Y plane. 7 : Z Axis – if using Autosurvey, this Anchor type specifies an additional Anchor to be surveyed and is not currently used to designate any part of the coordinate system. Its name as the Z Axis is a place holder for 3D implementation planned in a later software release.
2	Reserved	UINT8	Reserved

#	Parameter	Type	Definition
3	Flags	UINT8	Bits 0 – 1 : Sets sending the ELL or ELLEX information. Only relevant for Mobiles since Anchors do not compute positions. 0 : None – neither the ELL or ELLEX information is sent 1 : ELL – the Mobile will send the last position computed on its next range request 2 : ELLEX – the Mobile will send the last position along with the variance and covariance information in its next range request. Essentially includes the same information as the Location Info message. Bit 2 : Enables ELR where the Mobile will include its last range measurement in its next range request. Only relevant for Mobiles in Tracking Mode. Anchors in Tracking Mode do not range at all, and in Autosurvey Mode, Anchors automatically echo their last range measurements, while Mobiles do not range at all.
4	Reserved	UINT8	Reserved
5	Reserved	UINT16	Reserved
6	Beacon Interval	UINT16	The average ALOHA transmit time in milliseconds. Set to 0 to enable Automatic Congestion Control (ACC). For Anchors it represents the average beacon interval and for Mobiles it represents the average ranging time. Note that with ALOHA the radio will generate a random hold-off such that the minimum time is the amount of time it takes to measure a range, and the longest time is ((2 * Beacon Interval) – Ranging Time). Ignored in TDMA modes.
7	X Location	INT32	The X location of the radio in millimeters
8	Y Location	INT32	The Y location of the radio in millimeters
9	Z Location	INT32	The Z location of the radio in millimeters
END LOCATION MAP ENTRIES			

5.13 LOC_LOCATION_INFO (0x5201)

API: Location API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is optionally sent by the P4xx configured as a Mobile in the Location Map to the Host whenever a new location computation is attempted. It reports the new location as well as several associated metrics. If the computation fails, an error code is sent in the message. The sending of this message to the Host is defined by the Flags field in the LOC_SET_CONFIG_REQUEST message.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_LOCATION_INFO (0x5201)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Timestamp	UINT32	Milliseconds from boot to time of message sent
3	Node ID	UINT32	Node ID of radio sending this message

#	Parameter	Type	Definition
4	Node Type	UINT8	Specifies the role of the radio in the system. 0 : Mobile – the radio that ranges to Anchors and computes its own position. 1 : Anchor – the radio with known, fixed location. If not being actively ranged to, Anchors will send out a beacon so that Mobiles know they are available. 2 : Origin – if using Autosurvey, this Anchor type specifies that the Anchor is the origin in the relative coordinate system. 3 : +X Axis – if using Autosurvey, this Anchor type specifies that the line from the origin to this Anchor represents the +X axis. 4 : -X Axis – if using Autosurvey, this Anchor type specifies that the line from the origin to this Anchor represents the –X axis. 5 : +Y Axis – if using Autosurvey, this Anchor type specifies that the Y axis is in the positive direction relative to the X axis and that this Anchor will be placed in the +Y plane. 6 : -Y Axis – if using Autosurvey, this Anchor type specifies that the Y axis is in the negative direction relative to the X axis and that this Anchor will be placed in the -Y plane. 7 : Z Axis – if using Autosurvey, this Anchor type specifies an additional Anchor to be surveyed and is not currently used to designate any part of the coordinate system. Its name as the Z Axis is a place holder for 3D implementation planned in a later software release.

#	Parameter	Type	Definition
5	Solver Stage	UINT8	Solver stage – indicates how far along the initialization-to-steady-running process the solver has progressed. Note that the final stage of the solver depends on the configured solver mode (Kalman vs. Geometric, 2D vs 3D). 0 : Initialized from Location Map – the first stage of the solver that indicates the coordinates from the Location Map will be passed on to the next stage. 1 : Nonlinear Least Squares (2D) – the geometric solver that is used to provide an initial condition to the Kalman stage, or is the standard solver when the solver mode is Geometric 2D. This is the stage the solver will return to for rebirth (when the GDOP threshold is exceeded). 2 : Kalman (2D) – this is the normal, steady-state stage of the solver. With each new range, the solver will fold that new range into its motion model and compute a new location. 3 : Nonlinear Least Squares (3D) – the geometric solver that is used to provide an initial condition to the Kalman stage, or is the standard solver when the solver mode is Geometric 3D. This is the stage the solver will return to for rebirth (when the GDOP threshold is exceeded). 2 : Kalman (3D) – this is the normal, steady-state stage of the solver. With each new range, the solver will fold that new range into its motion model and compute a new location.
6	Solver Error Code	UINT8	See Table 5-2 at the end of this section for a list of the Solver Error Codes.
7	Reserved	UINT8	Reserved
8	Reserved	UINT16	Reserved
9	GDOP	UINT16	Bits 0 – 11 (lower 3 nibbles): Geometric Dilution of Precision – taken from the GPS community, its value represents the quality of the geometry of the mobile location relative to the Anchors. See the Wikipedia article on GDOP for details on how it is computed. Note that the value is scaled by 100. Bits 12 – 15 (uppermost nibble): The number of Anchors used to compute GDOP.
10	Timestamp	UINT32	Milliseconds from boot to time of location computed
11	X Location	INT32	The X component of the Mobile location in millimeters

#	Parameter	Type	Definition
12	Y Location	INT32	The Y component of the Mobile location in millimeters
13	Z Location	INT32	The Z component of the Mobile location in millimeters
14	X Variance	UINT16	The X variance in tenths of millimeters
15	Y Variance	UINT16	The Y variance in tenths of millimeters
16	Z Variance	UINT16	The Z variance in tenths of millimeters
17	XY Covariance	INT16	The XY covariance in tenths of millimeters
18	XZ Covariance	INT16	The XZ covariance in tenths of millimeters
19	YZ Covariance	INT16	The YZ covariance in tenths of millimeters

5.14 LOC_ECHOED_LOCATION_INFO (0x5202)

API: Location API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent when Echo Last Location (ELL) is enabled via the Report ELLs flag in the LOC_SET_CONFIG_REQUEST message. ELLs work analogous to ELRs in that when a Mobile computes a new location, when enabled via the Send ELL flag in the Location Map, it will include that location in its next range request packet. This allows other radios that hear the request (but are not the target of the request) to know the location of the other Mobiles in the system. Note that this message includes a subset of the information available from the LOC_LOCATION_INFO message. For the full set of information available, see the LOC_ECHOED_LOCATION_EX_INFO message.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_ECHOED_LOCATION_INFO (0x5202)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Node ID	UINT32	Node ID of radio sending this message
3	Remote Timestamp	UINT32	Milliseconds from boot to time of location computed
4	X Location	INT32	The X component of the Mobile location in millimeters

#	Parameter	Type	Definition
5	Y Location	INT32	The Y component of the Mobile location in millimeters
6	Z Location	INT32	The Z component of the Mobile location in millimeters
7	GDOP	UINT16	Bits 0 – 11 (lower 3 nibbles): Geometric Dilution of Precision – taken from the GPS community, its value represents the quality of the geometry of the Mobile location relative to the Anchors. See the Wikipedia article on GDOP for details on how it is computed. Note that the value is scaled by 100, Bits 12 – 15 (uppermost nibble): The number of Anchors used to compute GDOP
8	Solver Stage	UINT8	Solver stage – indicates how far along the initialization-to-steady-running process the solver has progressed. Note that the final, steady-running solver stage depends on the configured solver mode (Kalman vs. Geometric, 2D vs 3D). 0 : Initialized from Location Map – the first stage of the solver that indicates the coordinates from the Location Map will be passed on to the next stage. 1 : Nonlinear Least Squares (2D) – the geometric solver that is used to provide an initial condition to the Kalman stage, or is the standard solver when the solver mode is Geometric 2D. This is the stage the solver will return to for rebirth (when the GDOP threshold is exceeded). 2 : Kalman (2D) – this is the normal, steady-state stage of the solver. With each new range, the solver will fold that new range into its motion model and compute a new location. 3 : Nonlinear Least Squares (3D) – the geometric solver that is used to provide an initial condition to the Kalman stage, or is the standard solver when the solver mode is Geometric 3D. This is the stage the solver will return to for rebirth (when the GDOP threshold is exceeded). 2 : Kalman (3D) – this is the normal, steady-state stage of the solver. With each new range, the solver will fold that new range into its motion model and compute a new location.
9	Solver Error Code	UINT8	See Table 5-2 at the end of this section for a list of the Solver Error Codes.

5.15 LOC_ECHOED_LOCATION_EX_INFO (0x5203)

API: Location API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent when Echo Last Location (ELL) is enabled via the Report ELLs flag in the LOC_SET_CONFIG_REQUEST message. ELLs work analogous to ELRs in that when a Mobile computes a new location, when enabled via the Send ELLEX flag in the Location Map it will include that location in its next range request packet. This allows other radios that hear the request (but are not the target of the request) to know the location of the other Mobiles in the system. Note that this message includes the same information as the LOC_LOCATION_INFO message. If not all of the information is needed, use the smaller LOC_ECHOED_LOCATION_INFO message.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_ECHOED_LOCATION_EX_INFO (0x5203)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Node ID	UINT32	Node ID of radio sending this message

#	Parameter	Type	Definition
3	Node Type	UINT8	Specifies the role of the radio in the system. 0: Mobile – the radio that ranges to Anchors and computes its own position. 1 : Anchor – the radio with known, fixed location. If not being actively ranged to, Anchors will send out a beacon so that Mobiles know they are available. 2 : Origin – if using Autosurvey, this Anchor type specifies that the Anchor is the origin in the relative coordinate system. 3 : +X Axis – if using Autosurvey, this Anchor type specifies that the line from the origin to this Anchor represents the +X axis. 4 : -X Axis – if using Autosurvey, this Anchor type specifies that the line from the origin to this Anchor represents the –X axis. 5 : +Y Axis – if using Autosurvey, this Anchor type specifies that the Y axis is in the positive direction relative to the X axis and that this Anchor will be placed in the +Y plane. 6 : -Y Axis – if using Autosurvey, this Anchor type specifies that the Y axis is in the negative direction relative to the X axis and that this Anchor will be placed in the -Y plane. 7 : Z Axis – if using Autosurvey, this Anchor type specifies an additional Anchor to be surveyed and is not currently used to designate any part of the coordinate system. Its name as the Z Axis is a place holder for 3D implementation planned in a later software release.

#	Parameter	Type	Definition
4	Solver Stage	UINT8	Solver stage – indicates how far along the initialization-to-steady-running process the solver has progressed. Note that the final, steady-running solver stage depends on the configured solver mode (Kalman vs. Geometric, 2D vs 3D). 0 : Initialized from Location Map – the first stage of the solver that indicates the coordinates from the Location Map will be passed on to the next stage. 1 : Nonlinear Least Squares (2D) – the geometric solver that is used to provide an initial condition to the Kalman stage, or is the standard solver when the solver mode is Geometric 2D. This is the stage the solver will return to for rebirth (when the GDOP threshold is exceeded). 2 : Kalman (2D) – this is the normal, steady-state stage of the solver. With each new range, the solver will fold that new range into its motion model and compute a new location. 3 : Nonlinear Least Squares (3D) – the geometric solver that is used to provide an initial condition to the Kalman stage, or is the standard solver when the solver mode is Geometric 3D. This is the stage the solver will return to for rebirth (when the GDOP threshold is exceeded). 2 : Kalman (3D) – this is the normal, steady-state stage of the solver. With each new range, the solver will fold that new range into its motion model and compute a new location.
5	Solver Error Code	UINT8	See Table 5-2 at the end of this section for a list of the Solver Error Codes.
6	Reserved	UINT8	Reserved
7	Reserved	UINT16	Reserved
8	GDOP	UINT16	Bits 0 – 11 (lower 3 nibbles): Geometric Dilution of Precision – taken from the GPS community, its value represents the quality of the geometry of the Mobile location relative to the Anchors. See the Wikipedia article on GDOP for details on how it is computed. Note that the value is scaled by 100. Bits 12 – 15 (uppermost nibble): The number of Anchors used to compute GDOP.
9	Timestamp	UINT32	Milliseconds from boot to time of location computed
10	X Location	INT32	The X component of the Mobile location in millimeters

#	Parameter	Type	Definition
11	Y Location	INT32	The Y component of the Mobile location in millimeters
12	Z Location	INT32	The Z component of the Mobile location in millimeters
13	X Variance	UINT16	The X variance in tenths of millimeters
14	Y Variance	UINT16	The Y variance in tenths of millimeters
15	Z Variance	UINT16	The Z variance in tenths of millimeters
16	XY Covariance	INT16	The XY covariance in tenths of millimeters
17	XZ Covariance	INT16	The XZ covariance in tenths of millimeters
18	YZ Covariance	INT16	The YZ covariance in tenths of millimeters

5.16 LOC_OTA_ACK_INFO (0x5206)

API: Location API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent from the P4xx to the Host in response to a Location Mode or Location Map change that has been sent with the Broadcast flag enabled. The purpose of this message is to let the Host know which radios heard the broadcast message.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_OTA_ACK_INFO (0x5206)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Node ID	UINT32	Node ID of the acknowledging neighbor
3	Message Type	UINT16	Either LOC_SET_MODE_REQUEST (0x5001) or LOC_SET_LOCATION_MAP_REQUEST (0x5005)
4	Reserved	UINT16	Reserved

5.17 LOC_OTA_MODE_CHANGE_INFO (0x5207)

API: Location API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent from the P4xx to the Host when a broadcast Location Mode change has been received.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_OTA_MODE_CHANGE_INFO (0x5207)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	New Mode	UINT8	The new Location Mode 0 : Idle 1 : Autosurvey 2 : Tracking
3	Time Until Mode Change	UINT8	Time until the mode change takes effect, in units of 50ms. This will be 0 to report that the mode change is occurring immediately.
4	Reserved	UINT16	Reserved
5	Source Node ID	UINT32	Node ID of the radio that issued the change command
6	Route Node ID	UINT32	Node ID of the radio the change request was heard from

5.18 LOC_OTA_LOCATION_MAP_CHANGE_INFO (0x5208)

API: Location API
Message type: INFO (Radio)
Corresponding Message type: none

Purpose: This message is sent from the P4xx to the Host when a broadcast Location Map change has been received.

Packet Definition:

#	Parameter	Type	Definition
0	LOC_OTA_LOCATION_MAP_ CHANGE_INFO (0x5208)	UINT16	Message type
1	Message ID	UINT16	Associates request to confirm and info messages
2	Persist Flag	UINT8	Specifies how the radio settings in this message are written to flash memory (so changed settings persist through power cycling). Regardless of the Persist Flag's setting, the active radio settings are immediately updated from this message. Possible Persist Flag values are: 0: Do not write anything to flash; only update the active radio settings. 1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages. 2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. For more details, see Appendix D: Persist Flag Details and Usage.
3	Reserved	UINT8	Reserved
4	Reserved	UINT16	Reserved
5	Source Node ID	UINT32	Node ID of the radio that issued the change command
6	Route Node ID	UINT32	Node ID of the radio the change request was heard from

Table 5-1: CONFIRM Message Status Codes

Code	Status	Description
0	Success	The REQUEST message was processed successfully
1	Generic Failure	Catch-all for uncategorized failures
2	Wrong Op Mode	The REQUEST message cannot be acted upon in the current OpMode
3	Unsupported Value	The REQUEST message contained an unsupported value in one or more of its fields
4	Invalid During Sleep	The REQUEST message cannot be acted upon in the current sleep mode
5	Wrong Message Size	The number of bytes in the REQUEST message did not match the expected number of bytes for the message type
6	Not Enabled	The feature used by the REQUEST message is currently disabled
7	Wrong Buffer Size	The specified size of a buffer in the REQUEST message, or the size of the buffer itself, did not match the expected number of bytes for the message type
8	Unrecognized Message Type	The REQUEST Message Type was not recognized
0x80000000	Internal Error Code	An internal error code was generated. This status is OR'ed with the internal error code itself and should be used in communication with TDSR's support team.

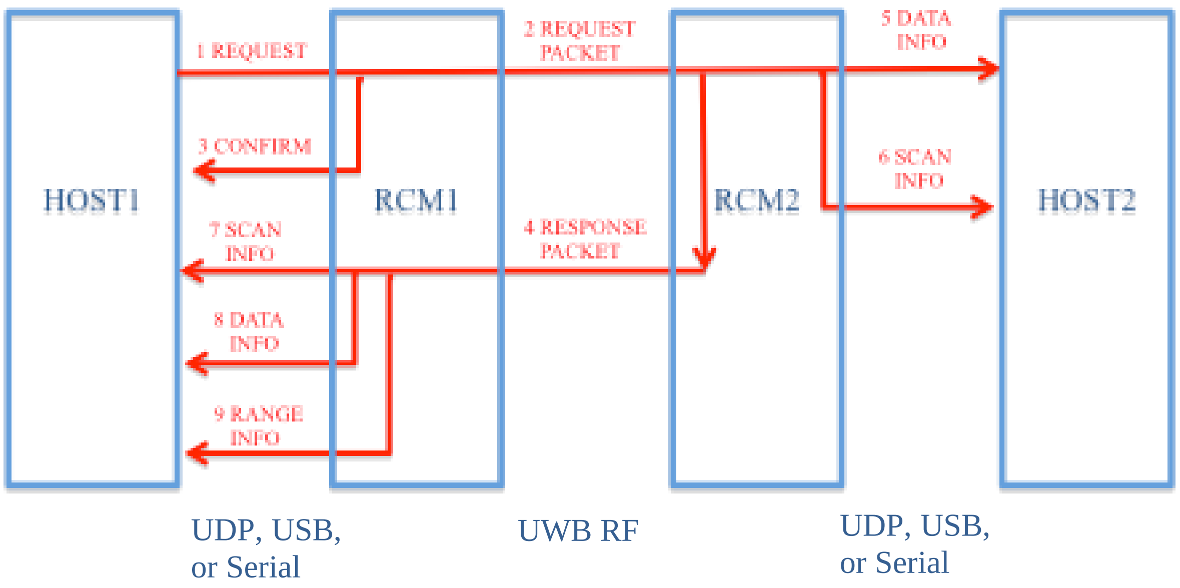
Table 5-2: Solver Error Codes

Code	Status	Description
0	Success	The solver computed a new location successfully
128	Range Error	Range did not pass threshold filter
129	Not Enough Ranges	There were not enough ranges collected for NLS to complete. For 2D, NLS requires at least three ranges from three different Anchors, within the last 3 seconds. For 3D, four ranges from four different Anchors are required.
130	Bad Geometry	The geometry of the Anchors to the Mobile was too poor to support a NLS solution
131	NLS Failed	NLS failed to converge. A likely cause would be a bad range

Code	Status	Description
		that passed the range filter.
132	Kalman Error	The Kalman error internally computed was too large
133	Kalman Fit Error	The Kalman solver produced a position that did not fit well with the measured ranges.
255	Generic Error	Contact TDSR if this error is reoccurring

Appendix A: Anatomy of a Complete Range Conversation

This diagram illustrates the process flow when measuring a single PRM from the point of view of (user) Host1 connected to a P4xx operating as an RCM Node.



1 REQUEST: HOST1 issues a RCM_SEND_RANGE_REQUEST message to RCM1, which is connected via Ethernet, USB, or Serial.

2 REQUEST PACKET: RCM1 emits a UWB packet in range request form targeted at RCM2.

3 CONFIRM: RCM1 responds with a RCM_SEND_RANGE_CONFIRM message.

4 RESPONSE PACKET: RCM2, if targeted by HOST1 as the RESPONDER, immediately responds with a UWB response packet.

5 DATA INFO: Any user data in the request packet is reported to HOST2 (if connected).

6 SCAN INFO: If configured to send scan info, RCM2 will send SCAN_INFO data to HOST2.

7 SCAN INFO: Upon reception of the response from RCM2 and if configured to send scan info, RCM1 will report scan data to HOST1.

8 DATA INFO: Any RESPONSE_DATA in the response packet will be reported to HOST1.

9 RANGE INFO: Finally RCM1 will compute precision and filtered distance measurements and report a RCM_RANGE_INFO message to HOST1.

NOTE 1: DATA INFO is reported to ANY Host connected to ANY RCM that "overhears" a packet (promiscuous operation.)
NOTE 2: RCM_SCAN_INFO is only sent to the Host if the Host has pre-configured the RCM to send it.
NOTE 3: A single MESSAGE_ID in HOST1's originating REQUEST message will be echoed at all end-points of this information flow. All RANGE, DATA, and SCAN INFO messages, at source, sink, and promiscuous Hosts, can be tied together using this single MESSAGE_ID. Thus MESSAGE_ID becomes an important tool when logging data and post-processing for full system connectivity and range analysis. In fact, the Host should rely on MESSAGE_ID for correlating messages, rather than any particular order or timing of RANGE, DATA, and SCAN INFO messages.

Appendix B: RCM Mode Parameter Descriptions

The following sections provide additional UWB or system-level detail for API parameters described in **Section 3**.

B.1 Message ID

The Message ID parameter is a convenient way for the system integrator to keep track of messages and associated responses.

The user typically specifies a unique Message ID for each REQUEST command sent to one or more local P4xx units. The P4xx will echo this Message ID in each of its responses. In addition, upon receipt of RANGE_INFO, DATA_INFO, and SCAN_INFO messages the Message ID in each of these messages will uniquely match the RCM_SEND_RANGE_REQUEST message used to generate these INFO packets.

Typical Host software for data collection will increment Message ID between each RANGE_REQUEST and use the Message ID to help relate RANGE, DATA, and SCAN info to a particular RANGE_REQUEST, as well as isolate and debug dropped Ethernet/UDP messages.

In RangeNet Mode, the P4xx is responsible for transmitting messages and will use the same technique of sending a unique Message ID such that it is possible to track message flow through the network. However, in Network Mode this message counter is separate and independent from the Message ID used during the Ranging Mode.

B.2 Pulse Integration Index (PII)

This value determines the number of pulses used in each radio symbol or scan point. Larger PII values result in a higher signal-to-noise ratio (SNR) with longer distance operation at the expense of slower ranging and data rates. All RCM nodes must be pre-configured with identical PII values in order to establish communication and ranging.

The user configures the “power of 2” of the pulse integration value. For example, a configured value of PII=7 results in the transmitting node sending 128 pulses per symbol and the receiving node expecting 128 pulses per symbol. If the user needs distance more than speed, then the PIIs of both transmitter and receiver can be reconfigured to PII=8, providing 2x the SNR in each symbol resulting in approximately 40% more distance in line-of-sight conditions. The ranging conversation time will roughly double with each increment of PII.

The entire range of PII values with consequential distances, data rates, and ranging measurement update times are provided in the *320-0317 P440 Data Sheet*.

Range measurement duration is the stopwatch time from request packet start to response packet received. This does not include overhead due to communication and processing by the user Host computer between successive packet reception and transmissions.

B.3 Antenna Mode

The P4xx supports two antenna ports. The port nearest the corner of the board is typically designated as the "A" port. By default the P4xx uses the A port. The user can define which port is to be used for transmit and which is to be used for receive. It is possible to use the same antenna for both transmission and reception.

A special antenna mode, Mode 128, is available to support automatic toggling of the default antenna port after each range response. This allows one or more Host-controlled Anchor P4xx nodes to iteratively measure distances to two antenna points connected to a responder, without requiring Host support for the responder.

This definition is specified in the following RCM Mode commands:

- RCM_SET_CONFIG_REQUEST
- RCM_SEND_RANGE_REQUEST
- RCM_SEND_DATA_REQUEST

The following table illustrates the values:

Mode Value	Transmit Port	Receive Port
0	A	A
1	B	B
2	A	B
3	B	A
128	Auto-Toggle after Response	

Table B-1: Antenna mode configuration settings for the P4xx

Note that there is also a parameter which allows the user to define the orientation of the Broadspec antenna (i.e., define whether the antenna is pointed up or pointed down). This capability is of importance for those interested in centimeter-level accuracy, since a mismatched antenna pair will result in an error in bias of 4 cm. A pair of antennas is considered mismatched when one antenna is pointing up and the other is pointing down. Defining which way the antenna is pointing allows the ranging algorithm to compensate for the difference and thereby avoid a 4 cm error.

B.4 Code Channel

Eleven separate and independent communications channels have been provided. P4xx devices only support one channel at a time. P4xxs receiving on a particular code channel will not hear those transmitting on a different channel. The channels are designated as channels 0 through 10. The default channel is channel 0. These channels allow the user to implement a code-division multiple access (CDMA) network supporting up to 10 different "cells," or a unique beacon code used for coordination which does not interfere with sub-cells.

The user is responsible for coordinating these code channels. Many more code channels are possible. The user should contact TDSR for additional or unique code channel enhancements.

B.5 Antenna Delay A & B

The P4xx platforms implement a Two-Way Time-of-Flight (TW-TOF) ranging technique. The result of a ranging conversation is the time it takes a pulse to travel from the pulser circuit of the requester to the sampling circuit of the responder and back. This value is divided by two and multiplied by the speed of light to return distance.

Typically the user wants a distance measurement relative to the antennas of the radios. P440 units have been calibrated by default to presume a zero Antenna Delay when using the default Broadspec antennas with a simple 90-degree elbow SMA connector. If the SMA connector is not used (i.e., the Broadspec antenna is attached directly to the P4xx antenna) then the user should substitute the value of 0 with a value of -91. In any event this effect is minor as the SMA connector increases the bias by about 25 mm or 1 inch.

If the system integrator changes to a new UWB antenna or uses a SMA coaxial cable extension, (for instance to keep the P4xx inside an electronics box with the antenna outside) then the range value reported would increase proportional to this extended TOF. The user can either consistently subtract the extra bias in his software or reconfigure the P4xx (using the RCM_SET_CONFIG_REQUEST) to store this bias using these registers.

B.6 Timestamp

Time stamp is the number of milliseconds that have elapsed since the latest P4xx power-up. This parameter is not used by the P4xx, but is provided to enable the system integrator to establish when the range was collected or data was received.

Note this is an interrupt-driven CMOS timestamp with millisecond accuracy. It is not based on picosecond timing triggers in the RF front end.

B.7 Vpeak, Noise, and SNR

Vpeak and Noise are signal quality metrics that can be used to compute the received Signal-to-Noise Ratio (SNR). These measurements do not require a full range conversation and are produced whenever the P4xx receives a range, data, or scan message packet.

Vpeak is also used as the basis for the Coarse Range Estimate (CRE). Vpeak is a measure of the maximum absolute value of the signal amplitude measured just after the leading edge offset. This peak value is an improved form of Relative Signal Strength (RSS) and rarely suffers from multipath fading or construction. It can be used to estimate distance from a receive-only signal, increasing the capacity and scalability of networks of RCM nodes. This is the basis for the Coarse Range Estimate (CRE).

Vpeak is a scaled value of the absolute value of the largest magnitude signal in the leading edge of the waveform scan. This scaling is based on the PII as follows: Vpeak*(2^PII)/512.

Noise is a scaled estimate of the energy present prior to the leading edge. It is scaled by the same factor: Noise * (2^PII)/512.

SNR can be computed from these values using the following equation,

$$SNR= k*20*Log10(Vpeak/Noise)$$

where “k” is a proportionality constant equal to 1.25 which is used to compensate for a bias in the Noise estimation process.

Signal and Noise are imperfect but adequate estimates. While both of these estimates are close, neither estimate is exact. Consequently the measure of SNR is close, but not exact.

Signal and Noise are imperfect estimates in the following senses.

First, SNR is actually computed from the scan measured during waveform generation after the receiver has acquired lock. Given that, the SNR reported is not the SNR the radio sees when it acquires, but rather the SNR it sees during waveform scan. Also, the peak is the largest measured signal withing a few nanoseconds of the Leading edge rather than the largest signal present. These two values are different \ depending on the radio lock point. (To illustrate this, set up a radio link in a multipath environment and observe the maximum signal as a function of lock point.)

Rangenet 2.1 (release 170330-final) includes the ability to use an FFT to reduce the noise in a measured scan and thereby increase the SNR by about 6 dB. Doing so will improve the accuracy and robustness of the reported range measurement. Therefore, the default setting is FFT on. However, the user has the ability to override this setting and not use the FFT. Note that the FFT is performed after the scan has been captured and does not have any effect on acquisition. Also, because of limitation in FPGA sizes, this capability is only available on P440s and not on the preceding P400, P410, and P412 platforms.

While this capability is an improvement for range measurement, SNR is also used to measure the operational range of the system. Since the SNR improvement applies only to the scan and not operational range, this could lead to confusion. An imperfect solution has been provided to avoid this confusion. While the waveforms (scans) and Vpeak are reported with the benefit of the FFT, the noise value reported has been reduced by an estimate of the benefit normally seen by the FFT. As a result, the value of SNR calculated from the Vpeak and Noise values will be approximately the same (within a dB) regardless of whether or not the FFT is used.

Noise is also imperfect in that it has been estimated based on an approximation technique rather than on the computation of standard deviation. For example, the “proper” way of calculating noise might be based on computing the standard deviation of the 1300 readings which occur prior to the leading edge. For processor computation reasons a much simpler estimation process was used. This process has an inherent bias of 1.25. Once this factor has been applied then the noise estimate has proven to be very close to the more “proper” technique.

As a practical matter SNR computed from Signal and Noise has proved to be a useful and repeatable, if slightly inaccurate, tool for describing radio performance.

Users wishing more exact estimates of SNR can turn FFT off, log scan waveforms, and develop custom algorithms that yield more accurate results. An example is offered below. It is more computationally intensive but still produces results within 1 dB of those produced by the P4xx.

1) Compute the standard deviation of the first 1337 entries in a Full Scan. This will produce statistics on the quiet, signal free period that precedes the first arriving energy in a scan. This is a measure of noise.
2) Noise = 2*(Standard Deviation of 1337)^2
3) Unscaled Vpeak = Vpeak*(2^PII)/256
4) SNR = 10*log10(Vpeak^2/Noise)

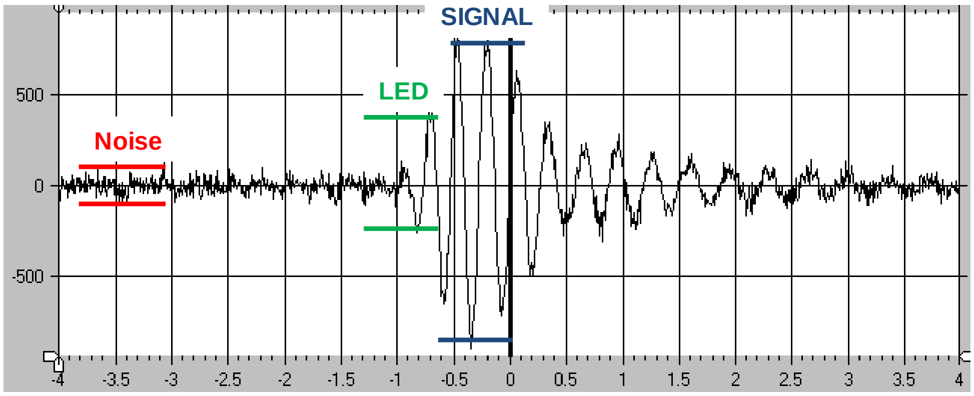


Fig. B-1: Illustration of Leading Edge, Signal, and Noise values in the pulse scan

B.8 Precision Range Measurement (PRM) and PRM Error Estimate (PRME)

A Precision Range Measurement (PRM) is the TW-TOF distance between the requesting and responding P4xxs. More precisely, it is a measure of the most direct path of a RF pulse from pulser to sampler, after subtraction of the antenna delay offsets.

After calibration with respect to antenna delay, the measurement should be considered the distance from antenna phase centers. The phase center point for the standard BroadSpec™ antenna is shown in the following illustration. PRM is reported in millimeters.

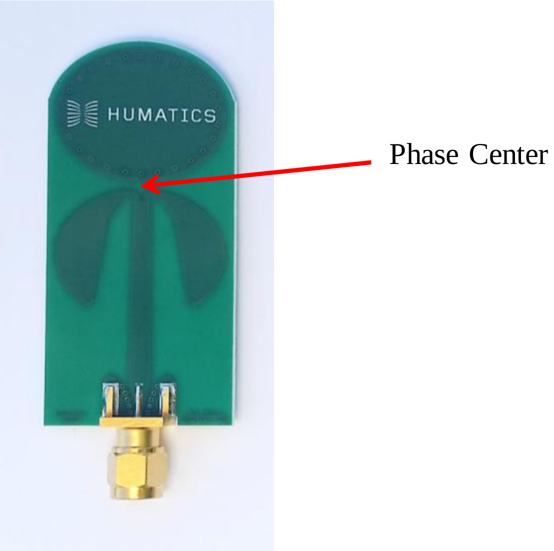


Fig. B-2: Broadspec UWB antenna with radiating point indicated

A Precision Range Measurement Error Estimate (PRME), returned with each PRM, is an estimate of the error of the associated PRM. This metric is pulled from a look-up table indexed by features of the incident pulse such as Channel Rise and Vpeak. The correlation analysis was performed in an indoor/outdoor propagation campaign in and around a standard office environment.

B.9 Coarse Range Estimate (CRE) and Coarse Range Error Estimate (CREE)

The firmware features a number of advanced measurements in addition to PRM. Any time an RCM receives a packet, the waveform scan is analyzed to provide a Coarse Range Estimate (CRE) through relative signal strength measure of the earliest, most direct path pulse. Even though pulsed-RF has the advantage of multipath separation, this technique can suffer from errors due to channel and antenna pattern changes. These changes are inherent in distributed dynamic localization systems. This problem can be overcome through occasional calibration through PRM ranging. CRE is only valid in Line-of-Sight (LOS), non-saturated conditions (saturation occurs even if the transmitting radio is at the lowest transmit gain and the receiver is LOS and within 16 feet (4.9 m). Specifically CRE values will be automatically generated only if:

1. The SEND_CRE flag is enabled in the RCM configuration
2. The CRE has been calibrated through at least one PRM measurement to the target Node ID
3. The RF channel is not SATURATED as determined by the radio firmware (and reported in the LEDflags field)
4. The RF channel is Line-of-Sight as determined by the radio firmware (and reported in the LEDflags field)

CRE Error (CREE) is also reported with each CRE to support the weighted combination of CRE with other distance measurements (such as PRM). These values are generated by a lookup table informed by the correlation between SNR and CRE Error.

B.10 Range Filter and Filtered Range Estimate (FRE)

The Filtered Range Estimate (FRE) and the Filtered Range Velocity (FRV), as well as their associated error estimates FRE Error (FREE) and FRV Error (FRVE), are the result of a recursive optimal (Kalman) estimator in the firmware. This Kalman Filter is based on two state variables, range and range velocity. As shown in the figure below, a unique Kalman Filter instance is created and maintained for each unique PRM range target. Whenever a new PRM or valid CRE measurement is generated this filter performs a weighted combination of these measurements with an internal linear motion model of distance. FREE and FRVE are generated and reported as a byproduct of this filter.

The Kalman Filter configuration parameters have been internally adjusted to optimize FRE and FRV operation on a group of vehicles moving 25 km/h or slower. When occasional PRMs are combined with frequent CRE measurements (which result from promiscuous listening) through the embedded filter, the PRM rate requirement is greatly reduced. See *B. Dewberry and W. Beeler*, "Increased Ranging Capacity using Ultra-Wideband Direct-path Pulse Signal Strength with Dynamic Recalibration", ION PLANS April 2012.

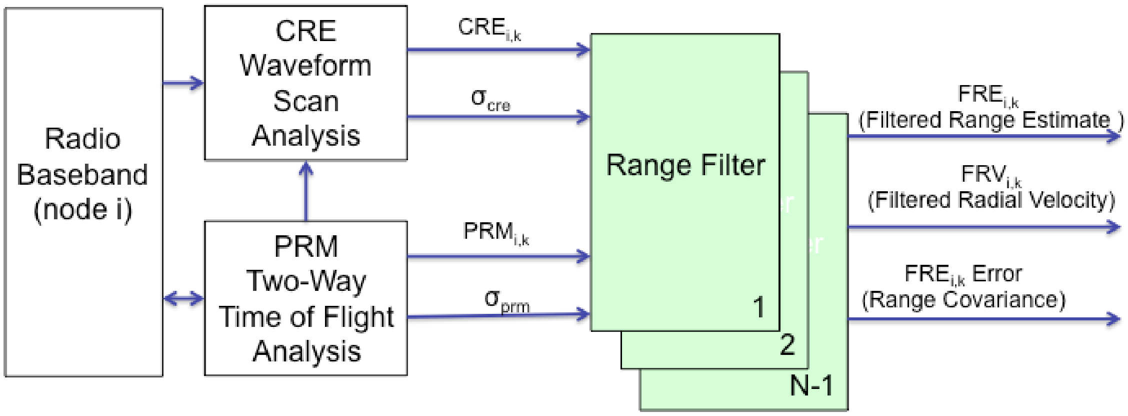

Radio Baseband (node i)
CRE Waveform Scan Analysis
PRM Two-Way Time of Flight Analysis
CRE_i,k
σ_cre
PRM_i,k
σ_prm
Range Filter
1
2
N-1
FRE_i,k (Filtered Range Estimate)
FRV_i,k (Filtered Radial Velocity)
FRE_i,k Error (Range Covariance)

Fig. B-3: Range filtering is used to generate FRE and FRV

FREs can provide higher update rates when used in a network. The increase in update rate comes at the cost of higher range error and increased latency. This is an advanced feature may benefit some networks. However, we strongly recommend that users focus on using PRMs and avoid FREs. Those interested in advanced networks should contact TDSR to discuss whether or not consider using FREs.

B.11 Leading Edge Offset, Lockspot Offset, Number of Scan Samples, and Scan Data

These parameters are all associated with the direct sequential scan of the pulse waveform and subsequent computation of the direct path. In order for the P4xx to compute an accurate TOF it must measure the offset from the lockspot, which is often on a multipath reflection, to the most direct pulse. The P4xx performs a scan relative to the lockspot, then measures the offset and updates the range estimate.

The RCM_SCAN_INFO message contains either 350 or 1632 points of the scan data depending on the "Flags" field in the RCM_SET_CONFIG_REQUEST command. Setting Bit1 to a 0 will produce scans of 350 points whereas setting Bit1 to 1 will produce scans of 1632 points. A short scan will be centered on the leading edge. A full scan will show 90 ns before the leading edge and 10 ns after the leading edge. The resolution of the scan waveform is 61 ps. The leading edge offset is the index in this scan where the system determined the time-of-arrival (TOA) of this pulse. The lockspot offset is the relative index where the radio acquired.

The RCM_SCAN_INFO is not required to use the P4xx. It is provided to allow the user to investigate (optionally) the channel impulse response around the direct path and make conclusions about the range or channel multipath content.

These parameters are illustrated in the following figure.

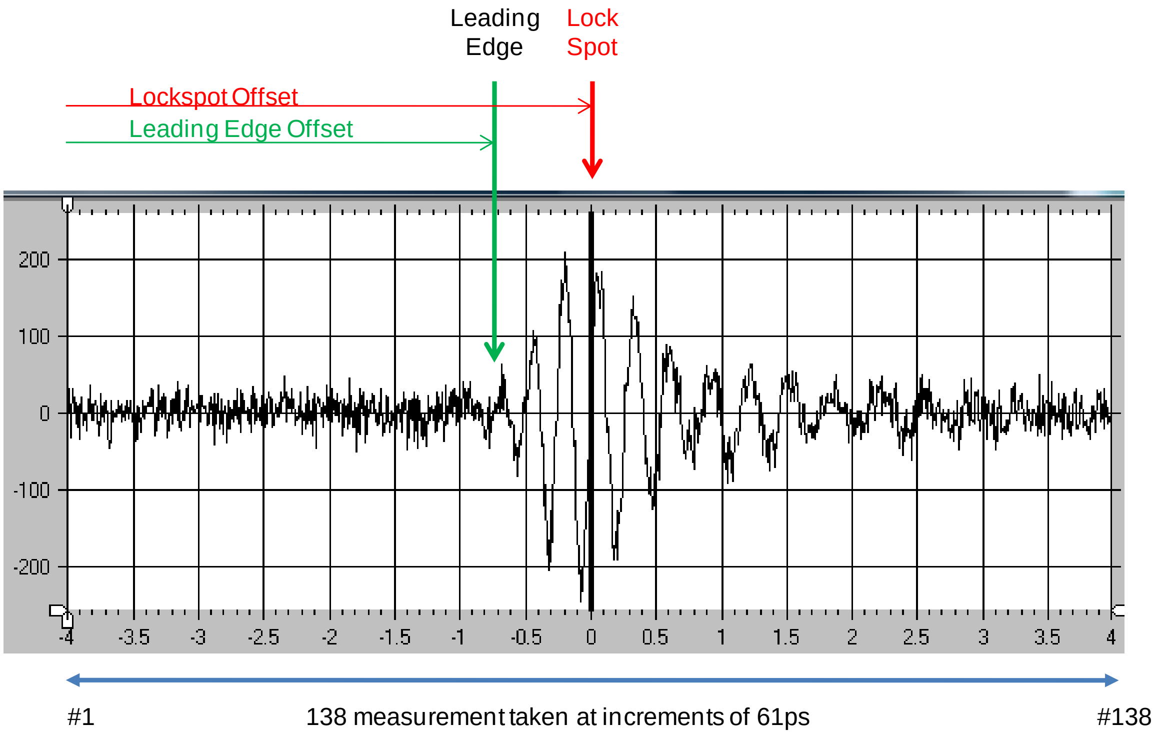


Fig. B-4: Waveform scan showing Lockspot, Lockspot Offset, Leading Edge and Leading Edge Offset

B.12 LED Flags

In pulsed RF radios the short window just after the Leading Edge offset contains much of the information required to characterize the RF channel. Four such characterizations are illustrated in **Figure B-5** below. Flags indicating SATURATION, NLOS, and LOS conditions are generated by the radio firmware any time a new packet is received and optionally reported to the Host in RCM_RANGE_INFO and RCM_SCAN_INFO messages.

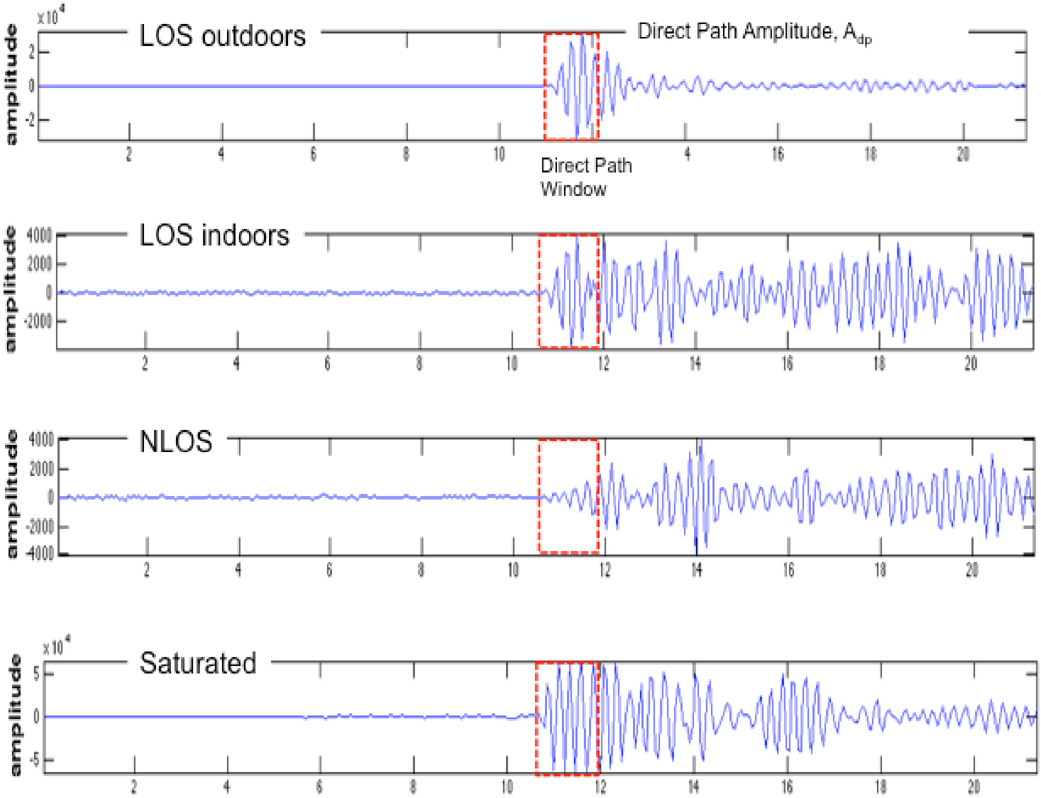


Fig. B-5: Using the Direct Path Window to characterize the RF channel

Appendix C: RangeNet Mode Parameter Descriptions

The following sections provide additional UWB or system-level detail for API parameters described in **Section 4**.

C.1 Selecting Min and Max Intervals

The amount of time required to complete a range request is dependent on the integration rate selected and the amount of data to be transferred. For example, a range request sent with PII=7 and no data will require 21 ms to complete. To determine how much time a given range request and data transfer will take to complete, switch to RCM Mode, load the data buffer with the required amount of data, set the PII to the desired value and range to another unit. (If the responder is also expected to transfer additional data then load its response buffer as well.) The time required for the range measurement is shown as "Stopwatch Time" on the Receive Tab.

Minimum transmit time can be set to be equal to the maximum interval. In which case the unit will initiate range requests at a fixed rate. Setting the minimum interval less than the time to complete is also valid. However, since doing so will occasionally result in range requests being requested before the previous request is complete, such requests will be reported as busy, indicated by a VCS (Virtual Carrier Sense) flag in the RANGE_INFO message, and a new random time interval will be generated.

Finally, while the RangeNet GUI will not allow the user to set a minimum interval greater than the maximum time, the API performs no bounds checks and will generate a random time between the minimum and maximum values. The difference being that the random time generated will always be more the maximum time and less than the minimum interval.

The User has the option of either setting the Min and Max Intervals manually or allowing RangeNet to independently set the ranging rate. This option is controlled by the ACC (Automatic Congestion Control) Flag. Setting the flag to off or zero will enable manual control. Setting the flag will enable ACC. When ACC is set, RangeNet will throttle the range request rate based on the number of nodes in the system such that all nodes operate at the fastest possible rate for a given number of nodes. For more information see the white paper "RangeNet/ALOHA Guide to Optimal Performance."

C.2 Using Beacons and Excluding Nodes

These are convenient tools for conserving network capacity by preventing unnecessary range requests.

Using Beacons. Beacons are defined as units which do not initiate range requests, but which periodically transmit a data packet or "beacon" and respond to range requests from other units. Beacons are useful in several instances. For example, consider a system that contains stationary Anchor Nodes and Mobile Nodes wherein the objective is to have the Mobiles know their own locations. In such a system, the Mobiles need to measure the distance to the Anchors and the Anchors, being stationary, have little need for range information. In this situation, the Anchors should be set up as Beacons. This will conserve airtime for range requests from Mobiles. However, it is important that Beacons occasionally transmit so that they can announce their presence to units which might enter the network. Beacons will initially "beacon" their name at the rate configured by the Min and Max intervals, unless otherwise queried to send a range response. Thus, in the example above, the Beacon could be configured for a Mean transmit rate around 5 seconds (this rate is a trade-off between airtime usage and on quickly a Mobile Node enters the area.) If they are queried, then the Beacons will not initiate beacon messages. Once they stop being queried they will resume transmitting beacon messages.

Excluding Nodes. The ability to exclude nodes is also useful in cases where some of the Anchors are fixed relative to each other but are moving relative to a Mobile. An example would be a robot vehicle using ranging to maintain the distance between the vehicle and a leader walking ahead of the vehicle. In this case, the walker would carry one node and the vehicle could have one node mounted on each of the vehicle corners. Ranges from the vehicle nodes to the walker would be used to determine range and bearing. Since the vehicle needs to know range and bearing, the nodes on the vehicle should initiate range measurements. In which case, all of the nodes on the vehicle would source range requests but there is no need for them to range to other vehicle mounted nodes. The Exclusion capability would be used to prevent any given vehicle mounted unit from communicating to other units mounted on the same vehicle.

Beaconing and Exclusion does not have to be a permanent assignment. Consider the first example of fixed Anchor Beacons and Mobile devices. When the Beacons are initially installed it may be necessary for them to compute their location relative to each other. Given that, it might be advisable for a unit to initially act as a Mobile so that the unit can compute its own location relative to other devices. Once that location has been determined, it could then convert to operation as a Beacon.

None of these examples are meant to suggest a preferred architecture. Depending on the needs of the end application, one might decide not to use Beacons or Exclusion in either of these examples. However, this capability is handy and can be very useful in limiting unnecessary range measurements. Limiting unnecessary measurements is valuable in that it maximizes overall system capacity.

C.3 Difference between Message ID and Host Message ID

In RCM Mode, Message ID numbers are controlled and incremented by the Host. This is the responsibility of the Host because the Host initiates the launch of all RF packets. In a complex system where many units are initiating messages, Message ID can be a useful tool for monitoring message flow.

In RangeNet Mode, Message IDs are controlled by the P4xx Network node and the Host has no ability to modify the Message ID number. However, there are a few RangeNet messages that travel from the Host to P4xx that do not initiate RF packets. Examples include setting the configuration and loading Requester or Response Data buffers. These messages are Host messages and each of these messages has an associated Host Message ID number which the Host is responsible for setting. Confirmation messages sent by the P4xx will use this same Host Message ID number.

Message flow in a network is even more complicated to track than the simple node to node messaging typical of RCM operation. This additional complexity can be handled by tracking both the Message ID and source Node ID of messages.

Appendix D: Persist Flag Details and Usage

Most radio settings can be saved to flash memory so that they can be applied at radio boot time. Generally, API messages for configuring the radio have a Persist Flag that can tell the radio to save the new settings to flash. This section provides more details on the Persist Flag.

D.1 Persist Flag Values

In an API message with a Persist Flag, the Persist Flag may be either 0 (Do not save to flash), 1 (Save all settings to flash), or 2 (Save settings from this specific message to flash). In more detail:

0: Do not write anything to flash; only update the active radio settings.

Rebooting the radio at this point would restore the radio to the last saved configuration.

1: Save all active radio settings to flash (including the settings updated by this message). This also includes any settings updated by any other messages, even if the persist flag was set to 0 in the other messages.

At this point, since the entire radio configuration is now saved to flash, rebooting the radio would cause it to boot up with the same configuration it is now using.

2: Save only the settings updated by this message to flash. Unlike when the Persist Flag is set to option 1, changed settings from other messages with the Persist Flag set to 0 will not be saved to flash. Note that this still involves the same overhead of flash writing as Persist Flag 1; the only difference between Persist Flags 1 and 2 is that 2 effectively leaves settings that are already in flash unchanged if they aren’t directly affected by this message.

Rebooting the radio at this point would restore the radio to the last saved configuration in general, but with the settings from this specific message included.

D.2 Persist Flag Usage

Host applications will typically only use Persist Flags 0 and 1. 0 is used when possible to avoid unnecessary flash writes (which take up enough time to matter for time-critical applications). When the application specifically needs to store the radio configuration in flash, it uses Persist Flag 1. Persist Flag 2 is fairly situational and not commonly used.

A configuration change commonly involves multiple API messages (e.g. RCM_SET_CONFIG_REQUEST with RN_SET_CONFIG_REQUEST, etc.). If this is the case, only the last message of a series of messages needs to have its Persist Flag set to 1, and the other messages can use Persist Flag 0. Setting all messages’ Persist Flag to 1 is ultimately harmless, but wastes resources on writing the radio configuration to flash for every message.

D.3 Saving the Opmode to Flash

The Radio Operational Mode / Opmode (RCM, RangeNet, Localization) can be saved to flash so that the radio boots up in the desired Opmode. However, for backwards compatibility reasons the API message (RCM_SET_OPMODE_REQUEST) does not have a Persist flag. To save the Opmode to flash, an API message with a Persist Flag set to 1 needs to be sent while the radio is in the desired Opmode. (Any message with a Persist Flag will work.)