V7 External Antenna RTLS Module User Manual

Version 1.6

Total 26 pages

Table of Contents

1. Introduction

The Redpoint RTLS Module is a full-function Real-Time Location System (RTLS) subsystem in a compact form factor. The self-contained module delivers all RTLS functions including high-accuracy positioning, data communication over Redpoint UWB network, and Bluetooth without the need of additional hardware.

The module is designed to allow our partners and customers to design their own location tags or add real-time location and two-way data communication capability to their existing products. The module can be used as it is, or as part of a larger system. It can be connected to external processors and peripherals through serial and GPIO interfaces. The single-sided design with castellation pins is compatible with the standard SMT assembly process and therefore allows the module to be directly installed on custom-designed carrier PCBs. The firmware is preloaded in the modules before they are shipped and therefore no programming or configuration is needed by the designers. The module will start operating when the power is supplied, and the reset is de-asserted.

FIGURE 1. REDPOINT RTLS MODULE

Figure 2 shows the functional blocks in the RTLS Module. The module is a single PCB design consisting of a micro-controller, an UWB radio transceiver with an external antenna, a 6-axis Inertial Measurement Unit (IMU), and power management circuitry.

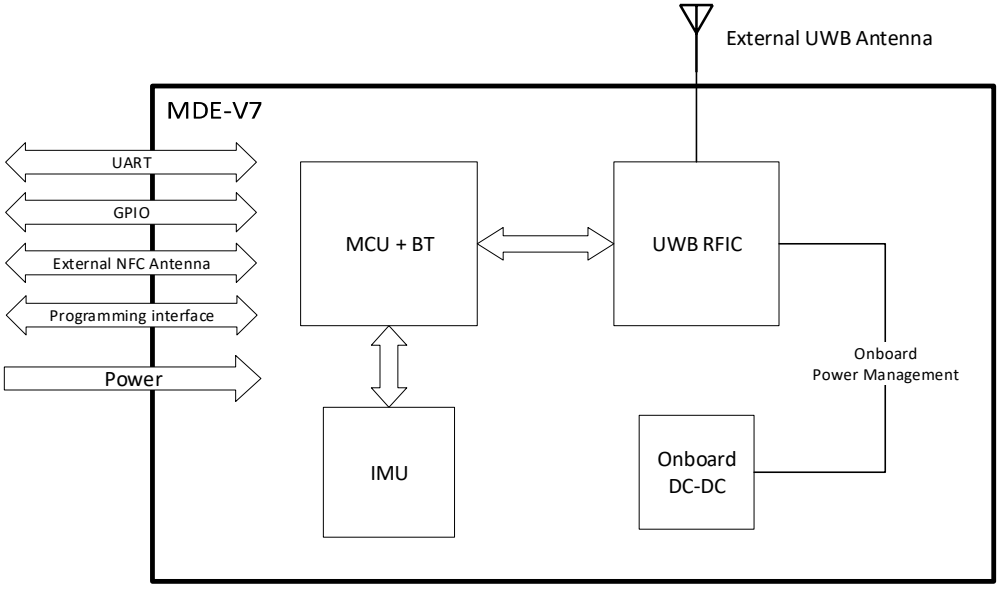

FIGURE 2. FUNCTIONAL BLOCK DIAGRAM

2. Partner Branding Guidelines

Redpoint Positioning Corp. requires all partner companies to display the 'Positioned by Redpoint' logo:

- On your software UI if you access the Redpoint server API
- On the outside of your tag if you use the Redpoint RTLS module

For questions about partner branding, or to obtain the logo, please contact support@redpointpositioning.com.

3. Hardware Design

The module consists of an Ultra-Wideband (UWB) transceiver (Decawave DW1000 ASIC), a Nordic nRF52 SoC, a 6-axis IMU and discrete components including UWB and Bluetooth antennas, RF circuitry and power management circuitry. A block diagram of the high-level architecture for the RTLS module is shown in Figure 3.

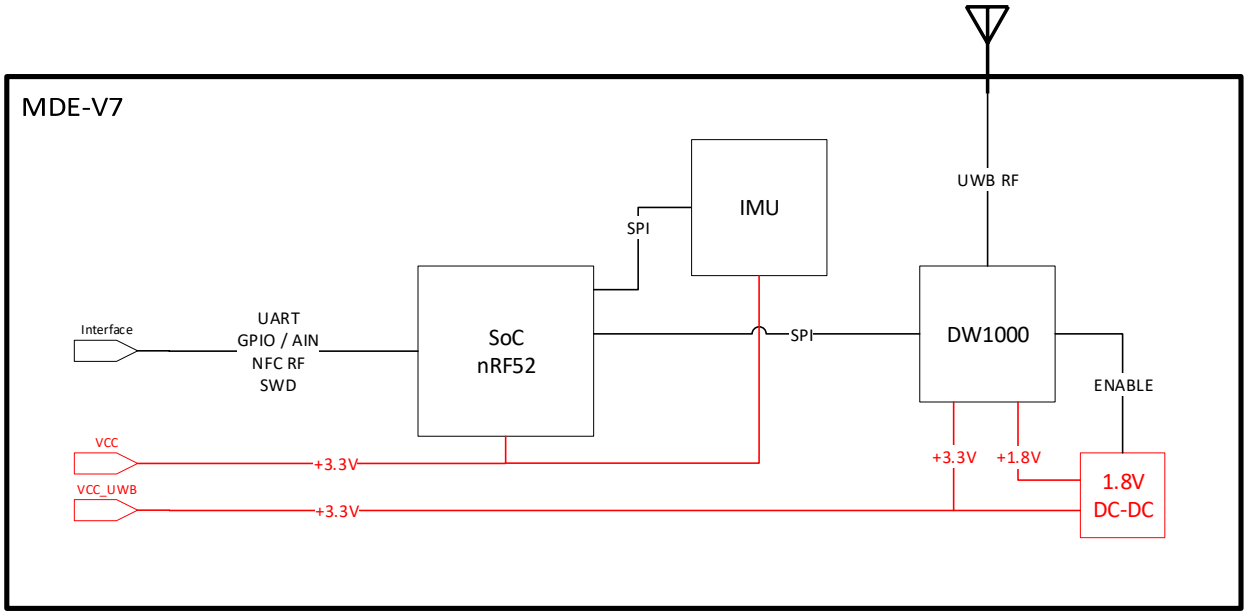

FIGURE 3. HARDWARE BLOCK DIAGRAM

3.1. Storage and Operating Conditions

The modules should be stored indoors with relative humidity no greater than 95%, and temperature between -40 °C and 85 °C.

Parameter	Maximum Rating
Temperature (°C)	-40 to 85
Humidity (%)	< 95

TABLE 1. STORAGE CONDITIONS

Parameter	Min	Max
Temperature (°C)	-40	85
Humidity (%)	N/A	<90
Supply Voltage (V)	2.8	3.6

TABLE 2. OPERATING CONDITIONS

3.2. Pin Definition

The RTLS module has a total of 24 pins, including:

- 10 for power and ground
- 5 dedicated pins (reset, programming, and etc.)
- 2 for NFC Antenna
- 2 for UART
- 5 configurable IOs as special-purpose or general-purpose IOs (GPIOs)

The pin assignment and layout are shown in Figure 4.

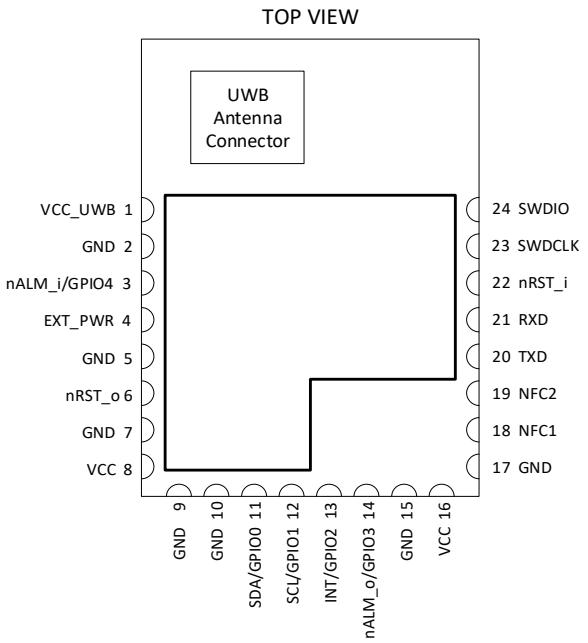

FIGURE 4. PIN ASSIGNMENT AND LAYOUT

The external IOs can be grouped functionally as:

- Dedicated pins
- One (1) SWD programming interface
- One (1) UART interface
- Five (5) configurable IOs
- One (1) NFC antenna port

The pin definitions are given in Table **3**.

Category	Pin name	Type	Pin	Description
Power	VCC	power input	8, 16	2.8V-3.6V
	VCC_UWB	power input	1	Power for UWB RF, must be the same voltage with VCC
	nRST_o	output	6	Open drain reset output, needs external pull-up, active low.
	EXT_PWR	input	4	External power source mode, active high
	GND	ground	2,5,7,9,10, 15,17	Power ground
Programming Interface	SWDCLK	input	23	SWD programming interface with internal pull-down resistor
	SWDIO	input / output	24	SWD programming interface with internal pull-up resistor
	nRST_i	input	22	Reset input with 13KOhm internal pull-up resistor, active low
UART	TXD	output	20	UART interface
	RXD	input	21	UART interface
NFC RF	NFC1	RF	18	NFC antenna port
	NFC2	RF	19	NFC antenna port
Configurable multipurpose IOs	SDA / GPIO(0)	digital IO / analog in	11	I2C bus with 13KOhm internal pull-up resistor / GPIO
	SCL / GPIO(1)	digital IO / analog in	12	I2C bus with 13KOhm internal pull-up resistor / GPIO
	INT / GPIO(2)	digital IO / analog in	13	IRQ input with internal pull-up resistor / GPIO
	nALM_o / GPIO(3)	digital IO / analog in	14	Open drain alarm output, tri-state in inactive mode, active low / GPIO
	nALM_i / GPIO(4)	digital IO / analog in	3	User alarm input with internal pull-up resistor, active low / GPIO

TABLE 3. PIN DEFINITION AND DESCRIPTION

3.2.1. Reset

The module can be reset externally by driving the *nRST_i* pin low for at least 0.5us. *nRST_i* is internally pulled up and no external pull-up is necessary.

During bootup, the module will drive *nRST_o* low for at least 10ms. Once the module is operational, the *nRST_o* will be tri-stated. Note that *nRST_o* is only asserted after *nRST_i* is de-asserted.

3.2.2. External Power Source Mode

The module reads *EXT_PWR* to determine if there is an unlimited external power source. If *EXT_PWR* is pulled up, the device is considered externally powered and operates in a high-current consumption mode. If *EXT_PWR* is not connected or pulled down, the device is considered battery-powered. *EXT_PWR* is connected to an internal 13K pull-down resistor, so the external driver should provide at least 160uA to pull up the *EXT_PWR* reliability.

3.2.3. UART

The UART is the full duplex bi-directional serial interface and the primary communication channel between the module and the external host or slave devices (e.g., host processor, display controller, etc.). To communicate to the module through UART, the interface shall be configured as follows:

Baud Rate	115,200
Flow Control	None
Data format	8 bit
Parity	None
Stop	1 bit

3.2.4. Configurable IOs

There are 5 multi-purpose pins that can be configured as GPIOs or special purpose pins. The configurations of these IOs are defined by the CFG register (see details in 3.2.4.2). When configured as special purpose IOs, these pins can be used as:

- I^2C bus: only supports the peripherals verified by RPP
- Alarm output: user alarm output, active low

3.3. Power Supply Requirements

The power supplies for the UWB radio are separated from the rest of the circuit on the module to improve the performance.

Power Rail	Parameter	Condition	Min	Typ	Max	Unit
VCC_UWB	Voltage		2.8	3.3	3.6	V
	Current	@3.3V		-	145	mA
VCC	Voltage		2.8	3.3	3.6	V
	Current	@3.3V		-	12	mA

TABLE 4. POWER INPUT REQUIREMENTS

3.4. RF Transceiver

The module has two complete radios, the UWB and the Bluetooth® radio.

The module contains a UWB radio operating in the unlicensed UWB band. The transmitted signal bandwidth is 500 MHz or greater. The unit is calibrated such that the maximum radiated spectrum density does not exceed -41 dBm/MHz and is fully compliant to the spectrum mask defined in FCC part 15. It is also compliant with the similar spectrum regulation for regions and countries including EU and China.

Parameter	Min	Max
Frequency Range (MHz)	3244	6999
10dB Bandwidth (MHz)	500	
Center Frequency[1] (MHz)	3494.4	6489.6
Spectrum Density (dBm/MHz)	N/A	-41

TABLE 5. UWB RF PARAMETERS

Based on configuration via software, the module can operate on one of the four 500MHz wide UWB channels specified in the table below.

Channel Number	F_{center}[1] (MHz)	F_{min}[2] (MHz)	F_{max} (MHz)
1[3]	3494.4	3244.8	3744
2	3993.6	3774	4243.2
3	4492.8	4243.2	4742.4
5	6489.6	6240	6739.2

TABLE 6. UWB CHANNELS AND FREQUENCY

3.5. Layout and Footprint

The module can be installed on a carrier board using a standard SMT process. Figure **5** shows the recommended land pattern on the carrier board.

To achieve good positioning performance, the antenna patterns must be preserved. Therefore, it is essential that there is no metal on any layer in the KEEP OUT AREA underneath the antennas. The orientation of the module also affects the module performance as the UWB antenna is linear polarized. In most cases, the optimal performance is achieved when the module is in vertical position.

[1] The center frequency is software configurable. See table 6 for details.
[2] F_{min} and F_{max} are the 10 dB lower and upper band edges, i.e., the frequency at which the power density drops 10 dB below the peak power density.
[3] Detection and Avoidance (DAA) is required in some countries for the channel(s). Redpoint's system does NOT implement DAA and therefore the user shall not operate the system in these channel(s).

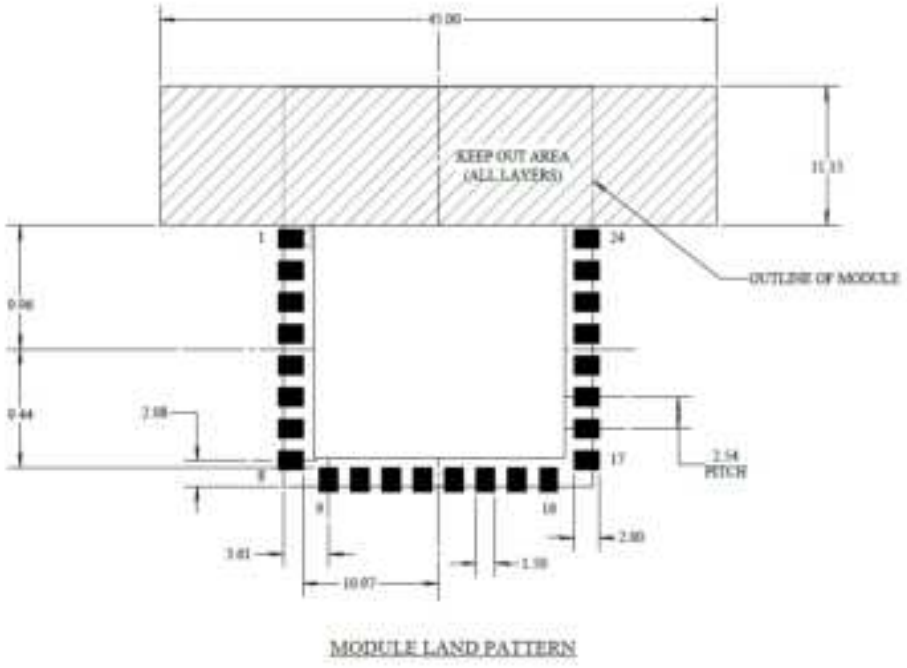

FIGURE 5. MODULE LAND PATTERN

3.6. Reference Design

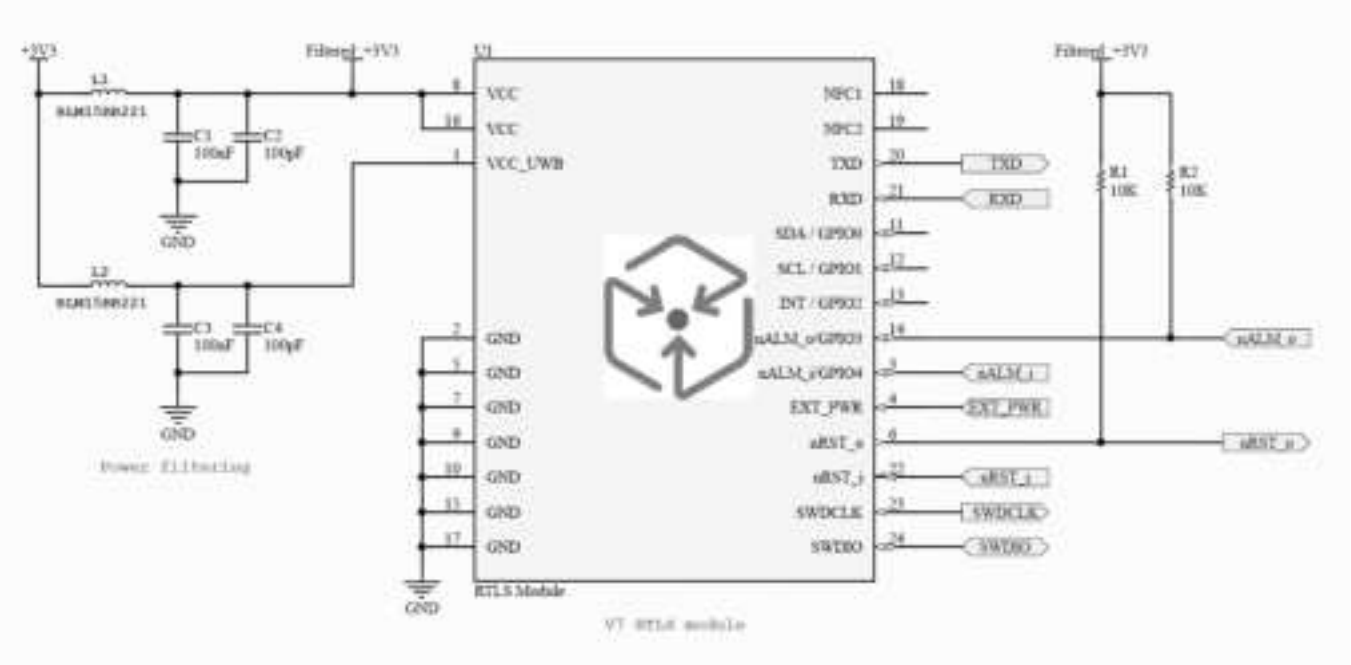

Figure 6. Reference Design

3.7. Packaging and Handling

The dimensions of the module are 32mm(L) x 25mm(W) x 3.11mm(T). There are 24 castellation hole pin-outs around the module. The pitch is 100mil (2.54mm).

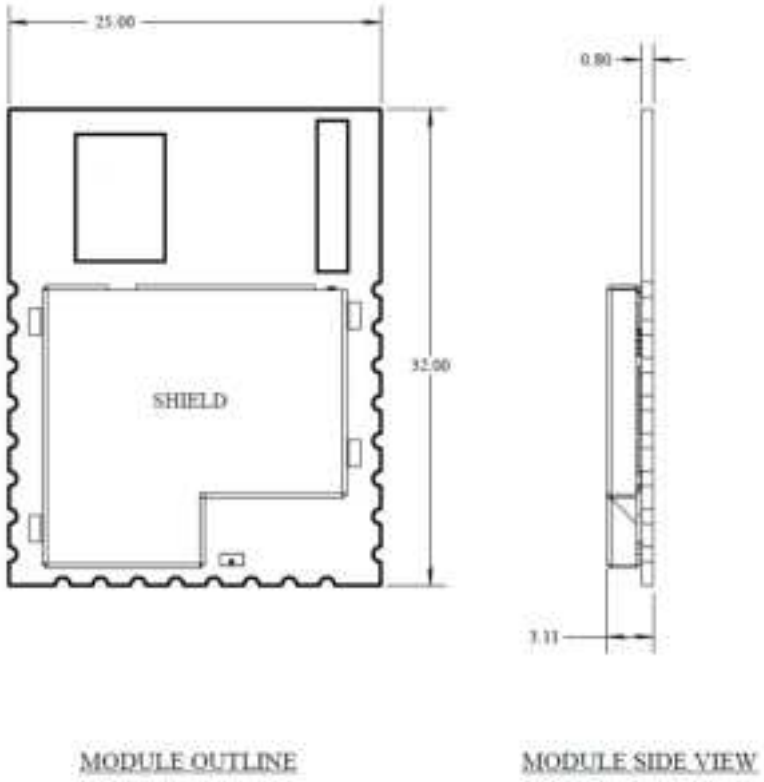

FIGURE 7. DIMENSION OF RTLS MODULE

3.8. Recommend Reflow Soldering Profile

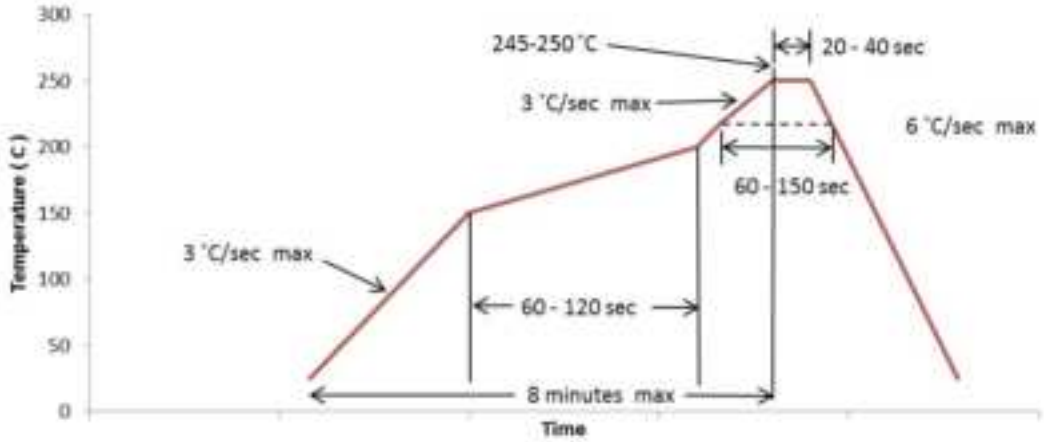

FIGURE 8. RECOMMEND PB-FREE REFLOW SOLDERING PROFILE

3.9. Certification and Marking

The RTLS module is certified for the following countries and regions:

Country or Region (Certification Authority)	ID or Marking
USA (FCC)	2ADX4-MODV7
Canada (IC)	12677-MODV7
European Union	CE marking for emission, safety and environmental
China, including Hong Kong (CMIIT)	

TABLE 7. REGULATORY CERTIFICATIONS

Products containing Redpoint RTLS modules are required to display the following on the product label when sold in the US: “containing FCC ID 2ADX4-MOD7V.”

Products containing Redpoint RTLS modules are required to display the following on the product label when sold in Canada: “containing IC: 12667-MOD7V.”

4. Software Design

4.1. Data Types and Sizes

Table 8 specifies data types and sizes used in this document.

Type	Range	Description
int8	-128 to 127	Signed 8-bit integer
int16	-32,768 to 32,767	Signed 16-bit integer
int32	-2,147,483,648 to 2,147,483,647	Signed 32-bit integer
uint8	0 to 255	Unsigned 8-bit integer
uint16	0 to 65,535	Unsigned 16-bit integer
uint32	0 to 4,294,967,295	Unsigned 32-bit integer
string	ASCII	ASCII character string
hex[N]	0-9,A-F, a-f	Hexadecimal string representing N/2 bytes

TABLE 8 DATA TYPES

4.2. Module Communication Interfaces

The RTLS module outputs various application data, such as tags’ positions, application alarms and messages. The module accepts input messages including configuration, commands and user data to be sent directly to the server.

For the rest of this document, the direction of the message is defined with respect to the module.

Messages shall be ASCII strings terminated with the LF (0x0A). Output messages are preceded with the # (pound) character followed by the message type. Optionally, a message can have content fields. Content of the message is generally in the form of variable=*value* separated by spaces.

The RTLS module supports communication over UART interfaces.

4.2.1. UART Interface

The UART is the full duplex bi-directional interface between the module and the external host device.

The following is the physical configuration of the UART:

- Baud rate: 115200.
- No hardware flow control.
- 8N1 byte format.

4.2.1.1. UART Modes of Operation

The module's UART has two modes of operation, ***command*** and ***binary***.

Command mode is used to provide the communication interface described in this document.

Binary mode is used to provide the sensor interface communication channel described in "[3] Sensor Interface User Guide". When UART is set to operate in binary mode, the module does not process incoming data and sends its payload as a binary sensor data packet directly to the server.

During boot-up, the UART is configured to command mode for 30 seconds. After 30 seconds, the module checks the "Keep CMD on" flag. If the flag is set, the UART stays in command mode until it receives a different configuration from the server or is turned off using the command line. If the flag is not set, the module turns off UART after 30 seconds. After it turns off, it can be turned on again only after rebooting. The flag can be configured using the CFG UART command described below.

4.2.2. Outgoing Messages

The available message types are:

- **#POS**—calculated position
- **#ALM**—alarm message
- **#MSG**—text message received from the server
- **#IMU**—raw output of the IMU
- **#VER**—printing hardware and firmware versions
- **#CFG**—module configuration

4.2.2.1. #POS - Position update

Position update messages start with a #POS preamble and have the following content fields:

Name	Type	Values	Description
x	int32		Position X in mm.
y	int32		Position Y in mm.

Name	Type	Values	Description
z	int32		Position Z in mm.
sl	uint16		Area ID
fom	uint8	50-255	**Figure of merit**—confidence level of the position estimate. A smaller value is better. **Note**: Not available for 1D, 0D and PD positioning mode.
er	uint32		0D mode - distance to 0D anchor in mm. 1D mode – distance to the line connecting anchor pair(mm). **Note**: Only available in 0D and 1D modes.
op	string	UWB|IMU, 0D|1D|2D|3D|PD, E,I,A,S	Comma separated flags indicating various position estimation options. Current values are: • **UWB|IMU -** position estimated from UWB or IMU • **0D|1D|2D|3D|PD -** positioning mode • **E -** edge correction applied • **I -** inactive state, IMU detected device not moving • **A -** ranging in non-synchronous mode (TW-TOA) • **S -** indicate tag being synchronized in network
ts	uint32		Local timestamp. Number of milliseconds elapsed since node boot up.

TABLE 9 **POSITION UPDATE CONTENT FIELDS**

The following is an example of a position update message:

```
#POS x=10352 y=2768 z=1008 sl=7 fom=50 op=UWB,2D,E ts=45376
```

4.2.2.2. #IMU – IMU output

The raw reading of the on-board 6-axis IMU can be output directly to the UART. The sampling rate of the IMU is 52Hz. The following table lists the content fields of the #IMU message:

Name	Type	Description
ax	int32	Linear acceleration axis X in mg/s^2
ay	int32	Linear acceleration axis Y in mg/s^2
az	int32	Linear acceleration axis Z in mg/s^2
rx	int32	Angular rate axis X in $rad*10^{-3}/s$
ry	int32	Angular rate axis Y in $rad*10^{-3}/s$
rz	int32	Angular rate axis Z in $rad*10^{-3}/s$
ts	uint32	Local timestamp of the number of milliseconds elapsed since node boot up

Name	Type	Description
md	uint8	0 - no motion detected / 1 - motion detected

TABLE 10 **CONTENT FIELDS OF THE IMU MESSAGE.**

The following is an example of the IMU raw readings:

```
#IMU ax=ddddd ay=ddddd az=ddddd rx=bbbbb ry=bbbbb rz=bbbb ts=45376 md=1
```

4.2.2.3. #ALM – Alarm

Alarm messages start with the preamble #ALM and have the following content fields:

Name	Type	Values	Description
t	string	SA|CA	Alarm type. Supported types: • **SA**—safety zone violation alarm • **CA**—collision avoidance alarm.
zal	uint8	0-5	Alarm level. **Note**: Only for SA alarms. • Non-0 indicates there is an alarm state. • 0 level means the alarm cleared and there is no alarm state.
z_uid	string	22 characters	The UID of the zone associated with the alarm. **Note**: Only for SA alarms.
d	uint16		Distance to monitor/announcer. **Note**: Only for CA alarms.
a16	hex[4]		16-bit short address of the monitor/announcer. **Note**: Only for CA alarms.
op	string	A|V	Comma-separated flags indicating various options: • **A**—Alarm is triggered by T2T LLL announcer with address a16. • **V**—Alarm is triggered by T2T LLL monitor with address a16.

TABLE 11 **ALARM MESSAGE CONTENT FIELDS**

The following is an example of a safety zone violation alarm:

```
#ALM t=SA zal=5 z_uid=auEbCGexShi14TfSmGs1kw
```

The following is an example of a collision avoidance violation alarm:

```
#ALM t=CA d=1234 a16=A4BF op=A
```

Note: When the alarm message is sent, the *nALM_o* is also driven low if special purpose *nALM_o* is configured using CFG command.

4.2.2.4. #MSG – Server message

ASCII messages received from the server are issued from the module as #MSG messages. This message type is indicated with the preamble #MSG, followed by the actual ASCII strings of the message. The preamble and the body of the server message is separated by a space. However, the message body may contain spaces and shall be preserved. Note that the format of the content field is different from other outgoing messages.

The following is an example of the message received from the server and output to the interface:

```
#MSG txt=”test message from the server”
```

The following is an example if binary data is sent from the server to the module:

```
#MSG hex=hex_string
```

4.2.2.5. #ERR – Hardware fault code

A message is printed when the module boots up if there is any hardware failure.

All faults are aggregated into one 32bit value. Every fault event is mapped to a particular bit.

Bit #	Description
0	Application image is corrupted.
1	Radio chip not found (initialization failed).
2	IMU chip not found (initialization failed).

TABLE 12 **VERSION MESSAGE CONTENT FIELDS**

The following is an example of the message when the radio chip has failed:

```
#ERR hw=00000002
```

4.2.2.6. #VER – Hardware and Firmware versions

Version messages start with the preamble #VER and have the following content fields:

Name	Type	Description
sn	string	MAC address of the module.
hw	string	Hardware revision of the module.
cbid	string	Carrier board ID.
fw	string	Firmware version.

TABLE 13 VERSION MESSAGE CONTENT FIELDS

The following is an example of the #VER message:

```
#VER sn=E4956EAE01BB hw=7.0 cbid=3E61 fw=5.7.1
```

The module will print version messages at bootup automatically if the “Keep CMD on” flag is set.

4.2.3. Incoming Messages and Commands

Through the UART, commands and messages can be sent to the module. Similar to output messages, the incoming messages are ASCII strings with a predefined 3-letter preamble.

- Configuration
- User Alarms
- User Messages
- External battery remaining capacity
- Others

4.2.3.1. CFG – Configuration

Many functions and IOs of the RTLS module are configurable. The module is typically configured by the server with over-the-air messages. Some configurations can also be set locally via the UART. If the CFG command is sent without the value field, it is treated as a read command and the module will output the current configuration value for the corresponding unit.

The following is the list of the configuration messages supported by the module:

- CFG config_word – set/get configuration register value
- CFG OUT – configure module output messages
- CFG IMU – to configure IMU module
- CFG UART – to configure UART mode
- CFG OTA – to request OTA control by host
- CFG CBD – to set/get CBID value

4.2.3.2. CFG config_word

The configuration word is set using the following command:

```
CFG config_word
```

where *config_word* is a 32bit hexadecimal value (reset value is 0xA0000000). The '0x' prefix is not required. If the command is sent without the value field, it is treated as a read command and the module will output the configuration value.

Bits	Field name	Type	Description
31:30	MODE	R/O	Tag operation mode, 00=navigation (DL-TDOA), 01=asset (UL-TDOA), 11=debug (TW-TOA), 10=reserved (not configured)
29	IMU_SLEEP_EN	R/W	0=node does not switch to inactive mode when it stops moving; 1= node switches to inactive mode when it stops moving;
27:18			Reserved
17	ALM_OUT_EN	R/W	GPIO(3) is configured as Alarm output when this bit is set.
16:0			Reserved

TABLE 15 **CFG MESSAGE CONTENT FIELDS**

4.2.3.3. CFG OUT

The CFG OUT message is used to configure which outgoing messages will be output by the module and has the following content fields:

Name	Type	Values	Description
en	string	P,I,M,A,H	Enabling output from following: P – positioning #POS I – enable #IMU output M – enable #MSG A – enable alarms #ALM H – enable high frequency positioning output if available

The following is an example of a CFG OUT message, enabling output for positions and alarms only:

```
CFG OUT en=P,A
```

A CFG OUT message without any option will disable the output.

```
CFG OUT en=
```

Note: The local pos output at tag UART (CFG OUT en=P) may not be matched with the pos updates sent to server when AF (Advance Filtering) is enabled on tag. Please refer to 3.2.4.4 CFG IMU for more details.

When AF is enabled, the pos update could be sent from AF with IMU data. The local pos output (CFG OUT en=P) could be mismatched with the pos data sent to server. It is a requirement to check both the pos and high frequency output (CFG OUT en=P,H) to match the pos data sent to server.

4.2.3.4. CFG IMU

The CFG IMU message is used to configure the IMU's mode of operation and has the following fields:

Name	Type	Values	Description
m	string	off md af	Configuring IMU mode: **off**—turn IMU off **md**—motion detection mode **af**—advance filtering

The following is an example of a CFG IMU message, turning IMU off:

```
CFG IMU m=off
```

4.2.3.5. CFG UART

The CFG UART message is used to set/erase ***command*** mode of the operation flag and has the following fields:

Name	Type	Values	Description
cmd	string	off on	Configuring UART mode: • **off** - turn UART off and clear "Keep CMD on" flag • **on** – set UART to command mode and set "Keep CMD on" flag

The following is an example of a CFG UART message, setting command mode and flag:

```
CFG UART cmd=on
```

4.2.3.6. CFG CBD

The CFG CBD message is used to check and set the CBID and has the following fields:

Name	Type	Values	Description
id	string	null <4 character string value>	The ID can only be set once. Redpoint will assign a CBID based on the carrier board with which the RTLS module is embedded.

The following is an example of a CFG CBD message, enabling output for positions and alarms only:

- To see the current value:

```
CFG CBD id
```

- To set a CBID value:

```
CFG CBD id=3E58
```

4.2.3.7. MSG – User Message

The MSG command allows the user to send data from the tag to the server. User data received by the server are sent out via the WebSocket. The RTLS server does NOT store user messages.

```
MSG hex=hex_string
```

The maximum payload size is 256 binary bytes. The payload is a string of hexadecimal digits (ASCII string consists of only characters ‘0’–‘9’ and ‘A’-‘F’).

Note: The delivery of the user data is not guaranteed. Users shall implement QoS mechanism if guaranteed delivery if required.

4.2.3.8. Hardware fault code

Upon receiving the following command, the module will print out the hardware fault code.

```
ERR
```

See 4.2.2.5 for details.

4.2.3.9. Version

Upon receiving the following command, the module will print out the HW and SW versions.

```
VER
```

See 4.2.2.5 for details.

4.2.3.10. BAT - External battery remaining capacity

The BAT command provides an interface to set the value of the remaining external battery capacity. It reports to the server as a battery level. If it is issued without arguments, the module will print the current value. The default reset value is "-1", meaning the battery is not attached or undefined.

If the value of the 'cap' argument is outside of allowed range or cannot be parsed, module returns the string 'error'.

The command has the following field:

Name	Type	Values	Description
cap	uint8	0-100 -1	Remaining battery capacity in percent To reset value to 'UNDEFINED'/No battery

The following are examples of a BAT command, setting capacity to different level:

```
BAT cap=57 // 57%
#BAT cap=57

BAT // empty to read current value
#BAT cap=57

BAT cap=125 // wrong value
#BAT error
```

4.3. Firmware Update

The module is pre-loaded with an application image. After it is powered on, the module can automatically start the application. The application firmware update can be carried out using the Over-the-Air (OTA) method or via the Serial Wire Debug (SWD) interface.

4.3.1. Over the Air (OTA)

An OTA firmware upgrade is performed using Redpoint's utility software. Please refer to *Redpoint SitePlan User Guide* for more information.

4.3.1.1. CFG OTA

The CFG OTA message is used by host to request OTA control and has the following fields:

Name	Type	Values	Description
host	string	• off • on	Requesting OTA control: • **on** – host requesting OTA control • **off** – host cancels OTA control

In the following examples, *H* is the host MCU and *M* is the RPP module.

The following is an example of a CFG OTA message. The Host is requesting OTA control and the module accepts the request:

1. H → M

```
CFG OTA host=on
```

2. M → H

```
#CFG OTA host=on
```

The following is an example of a CFG OTA message. The host is requesting OTA control and the module rejects the request:

1. H → M

```
CFG OTA host=on
```

2. M → H

```
#CFG OTA host=off
```

4.3.1.2. #OTA – OTA Output messages

The OTA output messages are used by the module to report the status of the OTA process to the host. An update is reported every 30 seconds while the OTA update is in process. OTA output messages start with an #OTA preamble and have the following content fields:

Name	Type	Values	Description
status	string	on off	• **on** – OTA process is on-going, variables "completed" and "size" indicates completion percentage and size of the FW image • **off** –OTA process is terminated/completed
completed	uint8	0-100	Variable indicates completion of the OTA process in percent. • Only available when ***status=on***
size	uint32		Size of the FW image in bytes that needed to be downloaded to the module • Only available when ***status=on***

The following is an example of the message reporting 75% completion, followed by the message, reporting that OTA is off:

```
#OTA status=on completed=75 size=128000
#OTA status=off
```

4.3.1.3. OTA Command

The OTA host-to-module message is used by the host MCU to control the OTA process. The control must be requested by the host using the CFG OTA command and accepted by the module. The OTA message has the following fields:

Name	Type	Values	Description
cmd	string	status cancel	• **status** – request to report status of the OTA process • **cancel** – request to cancel on-going OTA process

The following is the example of commands requesting status, followed by a request to cancel the OTA process:

```
OTA cmd=status
OTA cmd=cancel
```

4.3.1.4. OTA Examples

In the following examples, *H* is the host MCU and *M* is the RPP module.

OTA started, host allowed to proceed:

M → H

```
#OTA status=on completed=1 size=256000
```

OTA started, host canceled operation:

1. M → H

   ```
   #OTA status=on completed=1 size=256000
   ```

2. H → M

   ```
   OTA cmd=cancel
   ```

3. M → H

   ```
   #OTA status=off
   ```

OTA periodic reporting:

M → H

```
#OTA status=on completed=57 size=256000
```

OTA status check requested by host:

1. H → M

   ```
   OTA cmd=status
   ```

2. M → H:

   ```
   #OTA status=on completed=57 size=256000
   ```

or

1. H → M :

```
OTA cmd=status
```

2. M → H:

```
#OTA status=off
```

4.3.2. SWD interface

In the case that the OTA firmware update is impractical, the update can be performed via the SWD (Serial Wire Debug) programming interface.

SWD is a two-wire protocol for accessing the ARM debug interface. It is part of the ARM Debug Interface Specification v5 and is an alternative to JTAG. Please refer to reference [2] for more details.

5. Regulatory Information

5.1. Indoor use only

This equipment may only be operated indoors. Operation outdoors is in violation of 47 U.S.C. 301 and could subject the operator to serious legal penalties.

5.2. FCC Interference Statement (Part 15.105 (b))

According to FCC Part 15, Class A devices must contain the following text (or something similar) in the manual:

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

5.3. FCC Part 15 Clause 15.21:

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment

5.4. FCC Part 15.19(a):

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

1) This device may not cause harmful interference, and

2) this device must accept any interference received, including interference that may cause undesired operation.

5.5. ISED RSS-Gen Notice (in English and French):

This device complies with Industry Canada's licence-exempt RSSs. Operation is subject to the following two conditions:

1) This device may not cause interference; and
2) This device must accept any interference, including interference that may cause undesired operation of the device."

"Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

1) l'appareil ne doit pas produire de brouillage;
2) l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

5.6. ISED Canada ICES-003 Compliance Label

CAN ICES-3 (B)/NMB-3(B)

5.7. RF Exposure Guidance

In order to comply with FCC / ISED RF Exposure requirements, this device must be installed to provide at least 20 cm separation from the human body at all times.

Afin de se conformer aux exigences d'exposition RF FCC / ISED, cet appareil doit être installé pour fournir au moins 20 cm de séparation du corps humain en tout temps.

5.8. Module integration instructions for the End-Product Manufacturer

The End-Product Manufacturer shall include the applicable items as appropriate for the radio module:

1) List the FCC rules that are applicable to the modular transmitter (not part 15B)
2) Summarize any specific operational use conditions (power reduction requirements/compensation for cable loss for point to point antennas, peak and min gain per frequency band for 5GHz DFS master, professional use limitation-extends to host manufacturers instruction manual)
3) RF exposure considerations: (1) for the host product manufacturer and (2) for the end user
4) Antennas - A list of antennas included; identify types
5) Label and compliance information – advise host manufacturers to provide a physical / e-label stating, "Contains FCC ID: 2ADX4-MDEV7". "Contains IC: 12677A-MDEV7" with their finished product
6) Information on test modes and additional testing requirements – how to configure test modes for host product testing

7) Additional testing, Part 15 Subpart B disclaimer - include a statement that the final host product still requires Part 15B compliance testing with the modular transmitter installed

Disclaimer: The information provided in this document is subject to change without notice.