APS016 APPLICATION NOTE

MOVING FROM TREK1000
TO A PRODUCT

A discussion of the elements to consider when developing a commercial product based on the TREK1000 Two-Way-Ranging (TWR) RTLS IC Evaluation Kit

Version 2.3

This document is subject to change without notice

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

This document aims to give an appreciation and overview of the steps involved in developing a commercial RTLS product starting from Decawave's TREK1000 Two-Way-Ranging (TWR) RTLS IC Evaluation Kit. This kit employs Decawave's DW1000 ultra-wideband (UWB) transceiver IC.

In RTLS based on two-way-ranging a tag needs to perform two-way ranging based distance measurements with a minimum of three fixed anchors in order to be located. This is the RTLS scheme used in TREK1000 and the only RTLS scheme covered in this application note discussing the design elements a developer needs to consider in moving from the TREK1000 to a commercial TWR RTLS product offering.

Decawave has implemented another RTLS scheme, time difference of arrival (TDOA) RTLS, where a single message transmitted by a mobile tag and received by time-synchronised anchors is used to locate the tag. That RTLS technique is not discussed here. Please contact Decawave for more details of it.

1.1 Overview of TREK1000 and its RTLS operation

A simple approach was adopted for the TREK1000 design and operation: All nodes, whether mobile tags or fixed anchors, use the same EVB1000 boards with the same *DecaRangeRTLS* ARM cortex software image installed on all of them. The choice between anchor or tag modes of operation and the assignment of up to eight unique tag addresses and four unique anchor addresses is made via physical configuration switches. The tag operates to sleep for a period and then wake up and range to the anchors and then return to sleep for another period. After the ranging message exchange, each anchor computes a time-of-flight (TOF) estimate for the tag. The TREK1000 employs an optimised message exchange scheme whereby the tag sends a single poll message, listens for one response from each anchor, and then sends one final message to complete the ranging exchange with all anchors. At this point each anchor has sufficient information to calculate the TOF between itself and the tag. This scheme is described in detail in § 2.1 below. In an RTLS system, the TOF results would typically be sent to a location solver, (often over an Ethernet LAN), where the location of the tag would be computed. In the TREK1000, where the tag is continually ranging, the resultant TOF value calculated by each anchor is returned to the tag, during the next ranging exchange, embedded in the anchor's response to the tag. This is an efficient use of air time but means that the results incur a natural lag equal to the period of tag ranging attempts, i.e. the location rate.

Anchor #0 has additional special functionality in two respects:

- Firstly, Anchor #0 operates to assign activity slots to each of the tags so that their ranging exchanges do not mutually interfere. Anchor #0 does this by including a sleep time adjustment value every time it responds to a tag's ranging attempt. This sleep time adjustment is calculated to position each of the tags to wake-up in a separate slot, based on their address (#0 to #7). For the purposes of future discussion we shall call the full period, scheduling the interactions of each tag into separate slots, the ***scheduling frame***.

- Secondly, Anchor #0 reports the results of all ranging exchanges via its USB port. It does this by listening for and receiving time-of-flight results that the other anchors embed in their response messages, and gathering these along with its own calculated TOF results, before sending the a full set of ranging results for each tag to and attached PC via its USB port.

The *DecaRangeRTLS* PC software application connects to the USB port on Anchor #0 and gathers the ranging results for each tag ranging to each anchor. The *DecaRangeRTLS* PC software, knowing the coordinates of the anchor nodes (via configuration), solves the ranging distance results to give a location estimate for each tag, which it then displays on a floor plan graphic. The application can also log the results to a file for later analysis if this is desired.

1.2 TREK1000 system expansion and commercialisation

The TREK1000 was designed to allow evaluation of RTLS based on two-way ranging, (using the existing EVB1000 evaluation board for the DW1000 IC), and supporting a few tags being located by just 3 anchors.

The TREK1000 can be extended by adding a fourth anchor, say an additional EVB1000. Details on how to do this are outlined in the document "TREK1000 Expansion Options Instructions". As explained in the TREK1000 user manual this gives the TREK1000 more resolution in the Z or vertical axes. It does not increase the area covered by the TREK1000 kit.

Some optimisations can be achieved with the existing TREK1000 architecture to increase the number of anchors and tags. Some details on this, are covered later in this application note. However, the algorithm used in the DecaRangeRTLS software for location will require significant changes to take its input from more than four anchors. All of these changes require a detailed understanding of the existing TREK1000 system architecture and an amount of software development.

Extending the TREK1000 to cover physically larger areas requires system development, principally in the area of software architecture design and software development. This application note discusses a number of approaches to this type of development but does not provide details of the development required. Development of an expanded system such as this is left to the customer's system development and software teams.

The discussion in this application note assumes that the reader is considering developing a two-way ranging based RTLS solution. This application note aims to guide the development decision by giving an overview appreciation of the development work required. The elements for consideration are:

- NUMBER OF TAGS

  How many tags are needed and what is the required location frequency? Increasing the number of tag locations per second increases the air-utilisation which has physical limitations that must be considered. This is discussed in § 2 below.

- COVERAGE

  What area needs to be covered? The operational range of UWB is at most a couple of hundred meters in line-of-sight conditions, and considerably less in situations where walls and other obstructions, (including the object being located), act to attenuate the radio signals. [Please see the DW1000 Datasheet [1] for receiver sensitivity]. To provide RTLS coverage over larger areas, additional anchors are needed. What does this mean for the system and its expansion, and how can a system manage resources when there are lots of tags which can move from one area of coverage into another? This question and its related issues are discussed in § 3 below.

- ADDRESSING

In real world systems individual tags and anchors each need to have a unique address in order to refer to them and so that they can communicate correctly with one another. How are these addresses assigned and handled? This is discussed in § 4 below.

- HARDWARE DESIGN

  The PCB employed in the TREK1000 is the same as the board employed for the EVK1000, which is designed for evaluation of the DW1000 IC. This board is not suitable to act as a commercial RTLS tag or anchor. The typical design elements and considerations are discussed in § 5 below dealing with the design of anchor and tag hardware, and other considerations of production.

2 SCALING UP TWO-WAY RANGING – TAG LOCATIONS PER SECOND

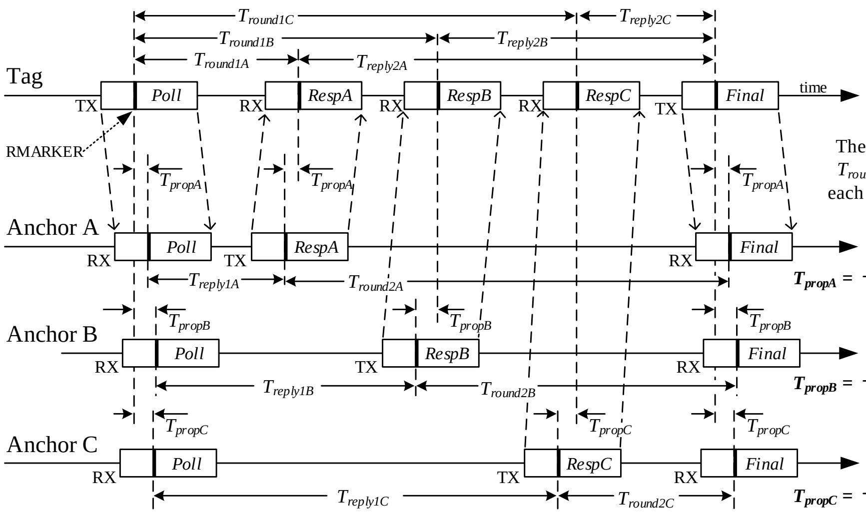
This section discusses the design issues of air-utilisation that need to be considered when increasing the number of tags being located and/or the frequency of locations.

2.1 Details of current TREK1000 implementation

The TREK1000 is a limited system designed to provide a simple Two-Way-Ranging based RTLS evaluation system comprised of just three anchors and up to eight tags.

With three anchors, the trilateration solver actually gives two solutions equidistant each side of the plane of the anchors, which is assumed to be all horizontal. A fourth anchor can be used to separately distinguish between the two solutions. With three anchors the lower of the solution pair is chosen, on the assumption that the anchors are all mounted at the same height above the area of interest for the localisation.

Instead of performing individual two-way ranging exchanges between the tag and each anchor, the TREK1000 employs an optimised message exchange scheme where the tag sends (broadcasts) a single *Poll* message received by all anchors, to which each anchor in turn sends a *Response* message, after which the tag completes the ranging exchange by sending a *Final* message received by all anchors. Each device precisely timestamps the transmission and reception times of the messages. Figure 1 shows the messaging for a three anchor case, labelling the various reply times and round-trip times observed by the tag and the anchors. In the *Final* message payload, the tag sends the actual transmission time of the *Final* message itself, along with the send time of the *Poll* and the times at which it received the responses from each of the anchors. Each anchor receiving the *Final* message then has sufficient information to calculate the time-of-flight (TOF) between itself and the tag, (assuming that the anchor correctly received the *Poll* and that the tag correctly received the anchor's *Response*). In Figure 1 the TOF values are labelled "T_{prop}" meaning the propagation time of the message between the tag and each of the anchors. The RMARKER is the part of the frame that is notionally time-stamped at the device antenna. The RMARKER is the beginning of the first symbol of the PHY header. Please refer to IEEE 802.15.4 [10] for more details of the UWB frame and its modulation. Figure 1 gives the formulae to calculate T_{prop} at each of the anchors.



$$T_{propA} = \frac{T_{round1A} \times T_{round2A} - T_{reply1A} \times T_{reply2A}}{T_{round1A} + T_{round2A} + T_{reply1A} + T_{reply2A}}$$

$$T_{propB} = \frac{T_{round1B} \times T_{round2B} - T_{reply1B} \times T_{reply2B}}{T_{round1B} + T_{round2B} + T_{reply1B} + T_{reply2B}}$$

$$T_{propC} = \frac{T_{round1C} \times T_{round2C} - T_{reply1C} \times T_{reply2C}}{T_{round1C} + T_{round2C} + T_{reply1C} + T_{reply2C}}$$

Figure 1: two-way ranging to three anchors with five messages

Note that the TREK1000 ranging exchange is actually designed to allow for four anchors, and so Figure 1 should really be extended to add an additional line with a response message, *RespD*, from the fourth anchor, and lengthening the time between *RespC* and the *Final* message to allow for the *RespD* message. When there is no fourth anchor present the TREK1000 tag still tries to receive this response so is wasting time and power turning on its receiver to await a response that will not come.

This (Figure 1) ranging scheme has very good performance because it is double-sided (in that there are round trip times measured from both sides) and the range calculation formulae use this double-sidedness to compensate for the clock offset between anchor and tag.

This scheme is also power and time efficient (compared to individual two-way ranging exchanges) as it reduces the number of messages to complete ranging to multiple anchors. The DW1000 implementation also saves power in the Tag by only turning on the receiver precisely when the messages from the anchors are expected. This is achieved using the *Delayed Transmission* and *Delayed Reception* features of the DW1000, to precisely coordinate the anchor transmission and tag reception times. The *Delayed Transmission* also allows the Tag to predict and embed the transmission time of the *Final* message within the message payload of the *Final* message itself.

In the TREK1000, in order to gather the TOF data for solving, and rather than employing any backhaul infrastructure, each anchor sends the TOF that it calculates for a tag, back to the tag in the next *Response* message to that tag. So, as each tag periodically ranges to the anchors, it gets the results from one period during the very next period. The tag in the TREK1000, (or rather the attached PC that picks up the results delivered through the USB port), thus has the data necessary to solve its location. In the TREK1000, Anchor #0 also listens for and receives all the other anchors' *Response* messages so that the PC attached to Anchor #0 can be given the TOF from all anchors to all tags and is then able solve and display the locations of all the tags. (Solving the location, of course, requires that the position of each anchor is known to the solver).

The lengths of the messages employed in this scheme including all MAC framing are: *Poll* = 13 octets, *Response* = 19 octets and *Final* = 44 octets.

The TREK1000 has two modes of operation. A *short frame* mode where the data rate is 6.81 Mbps, and the preamble length is just 128 symbols; and a *long range* mode where the data rate is 110 kbps, and the preamble is 1024 symbols long. In the *long range* mode the 44-octet *Final* message has a total frame duration of 4.95 ms. [In the short frame mode this is just 214 µs].

The TREK1000 employs a *scheduling frame*[1] structure of slots, where each slot is dedicated to one tag and is long enough for the tag to complete its ranging exchange to the four anchors. A fixed scheduling frame of 10 slots is used in the 110 kbps *long range* mode giving an update rate of approximately 3.5 Hz, while in the 6.81 Mbps *short frame* mode the scheduling frame is 10 slots long giving an update rate of 10 Hz. [In each case, in the TREK1000, two slots are reserved for anchor-to-anchor ranging which is used to automate the positioning of TREK1000 anchors].

For more details of the TREK1000 implementation the reader is referred to the *DecaRangeRTLS ARM Source Code Guide* [8].

[1] See § 1.1 for definition/discussion of *scheduling frame*. Note the term *superframe* may also be used for this.

2.2 Optimising the two-way ranging exchange

The interactions of the TREK1000 ranging exchange are fairly efficient. However there is some scope for increasing the performance to handle more tags or more updates per second.

The TREK10000 allows 100 µs between messages, to account for delays in the microprocessor in processing and responding to the message, and it further divides time into slots which include 3 ms to 6 ms of guard time between tag transmissions, (where the 1 ms microprocessor tick timer is used to control transmission).

To optimise for locating with only three anchors we can remove the time for the fourth anchor response and shorten the *Final* message accordingly. We can also reduce the time between messages to 80 µs, (which is achievable in the ARM), and employ a dedicated slot timing timer to give more precise timing that allows the guard time be reduced to ¼ of a millisecond, 250 µs. With these optimisations we can have 669 slots per second in the 6.81 Mbps *short frame* mode, (or 63 slots per second in the 110 kbps *long range* mode), where each slot can be used to complete a ranging exchange. The data for this calculation is given in Table 1 below.

This slot frequency defines the maximum number of locations that can be done per second, i.e. the product of the number of tags and the number of updates per second cannot be greater than the number of slots available. [This of course assumes that the system is employing a TDM (time-division-multiplexing) scheme so that tag interactions are controlled to all occur in the correct slot. Random access with ALOHA is only useful when the air-utilisation is low].

Table 1: Message sizes and slot timings

Mode: / Activity:	6.81 Mbps 128 PSR: Timings (s)	110 kbps 1024 PSR: Timings (s)	Description
POLL (13)	176.67E-6	2.53E-3	Tag sends Poll Message (13 octets)
µP processing	80.00E-6	80.00E-6	Anchor RX to response Delay
RESP (19)	182.82E-6	2.93E-3	Anchor A response, (19 octets), includes sleep time update (2) and previous TOF (4)
µP processing	80.00E-6	80.00E-6	Delay to allow Tag RX processing
RESP (19)	182.82E-6	2.93E-3	Anchor B response
µP processing	80.00E-6	80.00E-6	Delay to allow Tag RX processing
RESP (19)	182.82E-6	2.93E-3	Anchor C response
µP processing	80.00E-6	80.00E-6	Delay to allow Tag RX processing
FINAL (39)	202.82E-6	4.21E-3	Tag sends Final message, with Poll and Final TX times and three Response RX times, (each time is 5 octets)
Subtotal:	1.243E-3	15.55E-3	Time for complete interaction
Guard time	250.00E-6	250.00E-6	Keeps tags apart. Anchors calculate the range here too.
Total:	1.493E-3	15.797E-3	Slot duration = Interaction time + Guard time
Rate (Hz)	669.6	63.3	Number of slots per second.

The practical factors that reduce the numbers in Table 1 are discussed in § 2.3 below.

2.3 Choice of operating mode – channel and data rate

Table 1 shows that there is a large difference between the two TREK1000 operating modes in terms of the number of locations per second. There are also differences in power consumption. The longer messages at 110 kbps require that the transmitter and receiver are turned on for longer periods during each interaction which means that they consume more power. This could have a significant impact on tag battery life.

In general the operating range of the 110 kbps data rate is the main reason to choose it, however the 6.81 Mbps mode can achieve a performance boost if the messages are kept short and spaced appropriately to allow only one transmission per millisecond. In this case, since the regulations stipulate maximum power per millisecond, it is possible to boost the transmission power to account for the fraction of a millisecond that the transmissions takes. For example a message of half a millisecond duration can be sent at twice the nominal power, giving a 40% increase in line-of-sight operational range[2]. From Table 1 we can see that: the largest message at the 6.81 Mbps data is less than 250 µs; the spacing between *Poll* and *Final* is > 1 ms; and, the spacing between and individual anchor's response transmissions is also > 1 ms. So it is possible then to send these messages, from both tags and anchors, at four times the nominal power, which will give a doubling of the line-of-sight operational range that would otherwise be expected.

Another consideration is the choice of operating channel centre frequency. Table 2 presents the centre frequencies supported by the DW1000. In general the lower the channel centre frequency the longer the range. Regional regulations may however favour Channel 5 depending on the use case. For instance the low-band channels are only usable in Europe if low duty cycle (LDC) rules are complied with.

Table 2: UWB centre frequencies supported by DW1000

Channel #	Centre frequency	Band group
1	3494.4 MHz	Low-band
2	3993.6 MHz	Low-band
3	4492.8 MHz	Low-band
5	6489.6 MHz	High-band

The low duty cycle rules for low-band channels in Europe say that each device is allowed to transmit for a maximum of 5% of any second (i.e. 50ms) and for maximum of 0.5% of any hour (i.e. 18 seconds). These limits are quite restrictive for a two-way ranging based RTLS system given the amount of transmissions and receptions that are needed to estimate a single location. This may make Channel 5 more attractive, as is explained below:

Assuming that tags and anchors are active all the time, the most restrictive rule is the 18 seconds per hour limit. This is just 300 ms per minute. Exploring this, we can see from Figure 1 that each location exchange requires that tag sends two messages the *Poll* and the *Final*, while the anchor has to send one *Response* message.

Table 3 below presents the transmission times in milliseconds for tag and anchor for a single location

[2] Please refer to the DW1000 Datasheet [1] for details of the DW1000 receiver sensitivity.

using the frame timings from Table 1 relating to the optimised scheme shown in Figure 1.

Using the values from Table 3 with the 300 ms per minute limit for LDC rules (applying to low-band channels in Europe) and ignoring any other messaging overheads (e.g. to control the tag, etc.), we can calculate that a tag in the 6.81 Mbps mode would be allowed to perform 300/0.3754 = 790 locations per minute or 13.18 per second, while a tag operating in the 110 kbps mode would be restricted to 44.51 locations per minute, which is one every 1.35 seconds.

Table 3: Ranging TX air-time & Tag air-time for location

Mode	Tag Poll	Tag Final	Tag TX (*Poll + Final*) for 1 location	Anchor TX (*Response*) for 1 location
6.81 Mbps/128 PSR	0.177 ms	0.203 ms	0.3795 ms	0.1828 ms
110 kbps/1024 PSR	2.53 ms	4.21 ms	6.741 ms	2.927 ms

Considering the anchor then and using the Table 3 air-time numbers for the anchor responses: With the 300 ms per minute LDC limit, and ignoring any other overheads, each anchor operating in the 6.81 Mbps mode would be allowed to respond to 1640 tags per minute or 27.3 per second, but operating in the 110 kbps mode each anchor could only to respond to 102 tags per minute or just 1.71 per second.

As can be seen from this discussion the low duty cycle rules that apply to low-band channels in Europe can be quite restrictive for TWR RTLS, and this could be reason enough to use a higher data rate mode or to use Channel 5 for the European market.

Note also that in a real system there may be a need for additional messages (or longer messages) to control tag operation. This is discussed in § 3 below.

To get maximum use of the available air-time, the ranging attempts of individual tags need to be organised so they do not collide with each other. Random transmission with Aloha channel access is a good approach for a TDOA RTLS system but it is not suitable for a TWR based RTLS where a number of carefully timed interactions have to succeed in order to get a single location result. Controlling the timing of tags interactions with the anchors, via a time-division slot based approach is necessary to efficiently use the available air-time. The TREK1000 does this (as described in § 1.1) to support its eight tags. The control and scheduling of tag-to-anchor interactions is discussed further in § 3 below.

Note that the accuracy of the crystal oscillator timing reference may also be a factor in the operating communications range of the DW1000. With a low clock offset between units it is possible to achieve a longer operational range at the 110 kbps data rate, (see receiver sensitivity in the DW1000 Datasheet [1]). This requires the DW1000's *Receiver Operating Parameter Set* configured to the "*Tight*" selection (see the DW1000 User Manual [2] for details) and requires the use of a TCXO with the required 1 ppm specification.

Additional operating range at the 6.81 Mbps data rate might be achieved by employing a 64-symbol preamble at the 6.81 Mbps data rate to reduce the frame length, (selecting the *Length64* receiver operating parameter set), which allows an increase in the transmit power (giving better range) while still keeping the average power per millisecond within the regulatory limits. This mode of operation requires that the total clock offset is < 15 ppm, or +/- 7 ppm at each end of the link, which can be achieved with a moderate crystal tolerance specification and by employing the crystal trimming feature of the DW1000.

3 COVERING A LARGER AREA IN A BIGGER SYSTEM

In a building, or wider campus, where the area to be covered by the RTLS system is larger than the range of UWB, additional anchors are needed. This section discusses the ramifications of expanding the system to have many anchors.

The maximum number of tag locations per second was discussed in § 2 above. These air-occupancy limits apply to small and large systems alike, however over a wider campus there is scope for reusing the same air-time for separate tags' ranging interactions providing that they are sufficiently remote from each other that they do not interfere.

The two main system design questions dealt with in this section are:

- Where are the ranging results gathered, solved, and used? This dictates the system architecture.
- How does the mobile (tag) unit know what fixed (anchor) units are in the vicinity and which of those anchors it should range to, and when? This dictates the system scheduling and control.

These two questions and the approaches to their solution are discussed below along with how these choices affect the resulting system design.

3.1 System architecture for large area deployment

In the TREK1000 the location solver is part of the DecaRangeRTLS PC (GUI) application, and this can operate in two ways: (a) When the PC is connected to Anchor #0 the DecaRangeRTLS application gets range estimates for all tags talking to Anchor #0 and also receives the ranging results from Anchor #1 and Anchor #2 (embedded in their response messages to the tags), and so can solve and display the location of all the mobile tags. This is the RTLS system (tracking) mode where all assets are located by a central location engine (CLE) that monitors and reports as required. (b) When the PC is connected to a mobile tag the DecaRangeRTLS application receives the distance to all anchors that the tag performs ranging with and solves to find the location for the tag. This is a navigation mode – the tag and PC solver are the mobile system that figures out its own location as it moves around the environment

For a navigation application over a larger area a mobile device can also incorporate the solver, so that when it gets its distance from three known position anchors it can calculate its location. The anchors themselves do not have to know where they are, although if they did they could perhaps pass this information in the ranging response message. The solver has to know the anchor locations to solve the trilateration and locate the mobile unit. The anchor position information could be distributed via the anchors or another system depending on the capabilities of the mobile device.

For an asset tracking RTLS system it is usual for the anchors to report the tag ranges to a CLE for solving. This *backhaul* from anchors to CLE may be via wired Ethernet say, or using a Wi-Fi transport, or maybe via network implemented using the anchors' UWB. Obviously, there is a cost in the anchors to implement any of these *backhaul* mechanisms, and, if using UWB *backhaul* there is the additional cost is the air-time for the *backhaul* messages which will give less time for ranging interactions. The system designer needs to consider the merit of each possible choice.

In reality the solver could be in a CLE on the network or in the mobile unit, and still support either navigation asset tracking use cases. The network CLE could transmit the RTLS solution to the mobile device if it needed it for navigation, or, the mobile solver could send the result to the central monitoring station for logging and reporting as required. One challenge is selecting the anchors to range to and how this activity is coordinated. This is discussed in § 3.2 below.

3.2 System scheduling and control for large area deployment

In an RTLS where tags randomly wake and try to communicate or perform ranging with multiple anchors, the system performance degrades rapidly as air-occupancy increases, and so, such systems are only practical when the number of tags and the number of location updates per second is low, e.g. using < 15 % of the available air-time.

In general then, for good performance and to maximise the number of location updates per second, the two-way ranging based RTLS system needs to coordinate its tags so that their ranging exchanges with the anchors do not conflict with each other. This really means that the tags need to keep track of time while idle (or sleeping in a low power mode) and wakeup in their assigned 'slot', when it is their turn to interact with the anchors around them. The RTLS system thus needs to communicate with each tag to tell it when to perform ranging exchanges and which anchors to range to.

A typical way to achieve this is for a new tag to send a periodic "blink" and listen for a reply until it finds (or is *discovered* by) an RTLS that tells it what to do, assigning it a dedicated ranging slot. The tag blink period in this discovery phase would generally be substantially less than the normal RTLS tag scheduling interval, and not aligned with it (i.e. to avoid staying in lock-step with any particular tag's interactions that either hide the new tag and/or corrupt those interactions).

Blinking and listening directly afterwards for a response is a good approach for a tag as it is reasonably power efficient, and it is not too invasive of system operation providing that this is a relatively infrequent occurrence. A tag that is newly initialised/powered-on or loses contact with an RTLS (e.g. if it gets no response from any anchor after some number of retries) could enter this blinking/listening state until it is *discovered* or re-*discovered*.

An alternative approach is for selected anchors in the RTLS infrastructure to send beacon messages defining the *scheduling frame* where a particular slot might be allocated for tags to perform the joining mechanism. This approach means that a tag that has not joined an RTLS may have to listen for the beacon for a protracted period which is a very power costly thing to do if it has to be done for a long period. A modification of this technique to make it more efficient is to provide some scheduling information in all anchors' responses so an un-*discovered* tag just has to listen for just one ranging slot period to get the scheduling information. Ranging slots unassigned to any tag would need to be occupied by place holder anchor transmissions with the scheduling information. This listening strategy is okay provided that the tag will always be in range of the RTLS network elements (i.e. at least one anchor) but if the tag can go out of range then listening for prolonged periods can still be quite costly in terms of tag power drain.

In a system where the tag individually performs ranging with selected anchors, part of controlling the tag transmission schedule would be telling the tag which anchors to perform ranging exchanges with. In a deployment where a tag can roam among many anchors within a large facility, it is clearly inefficient for the tag to try to perform ranging with each of the anchors in the facility since most anchors will be out of range and also because only three ranges to three anchors are needed to locate the tag, and, in general it is desirable to range to the nearest anchors to have a better chance of success and better location results. The RTLS system then might tell the tag which anchors are around it that it should range to. This information could be included in the response to the blink as part of discovery, and/or updated/communicated during the ranging exchanges depending on which anchors hear the tags' transmissions.

A quite different approach can be adopted when employing the optimised ranging scheme of Figure 1 because, as is done in the TREK1000, the tag can send the *Poll* and *Final* messages using broadcast addressing. The activity of controlling the ranging then for the infrastructure to nominate which of its anchors should respond to the tag in what sequence, and for one of these responses include the control data to maintain the tag's transmission slot.

3.3 Example operating methodology for larger system

This section outlines a specific operational scenario based on the ideas introduced above as an example of a possible implementation approach for an RTLS, and in particular using the optimised three-anchor messaging scheme of Figure 1. An implementer may follow this but is of course also free to consider other implementation options to suit the target application use cases.

Starting with a set of installed anchors and a central controller. The term "*central location engine"* abbreviated to "CLE" will be used here to mean both the controller function and the location solving function. It is assumed that the CLE has good two-way communication with anchors to get updates from them and to be able to control them. Typically this will be TCP/IP over Ethernet or Wi-Fi, (using UWB for this communications is possible but that will eat into the air-time available for localisation).

Figure 2 shows a typical office floorplan with anchors in position and connected to the CLE via an Ethernet LAN. Note that this is not advocating any particular anchor topology or installation density. The number of anchors required to cover an area is a function of the anchor design, the operating mode, the materials within the space being covered, and the performance required.

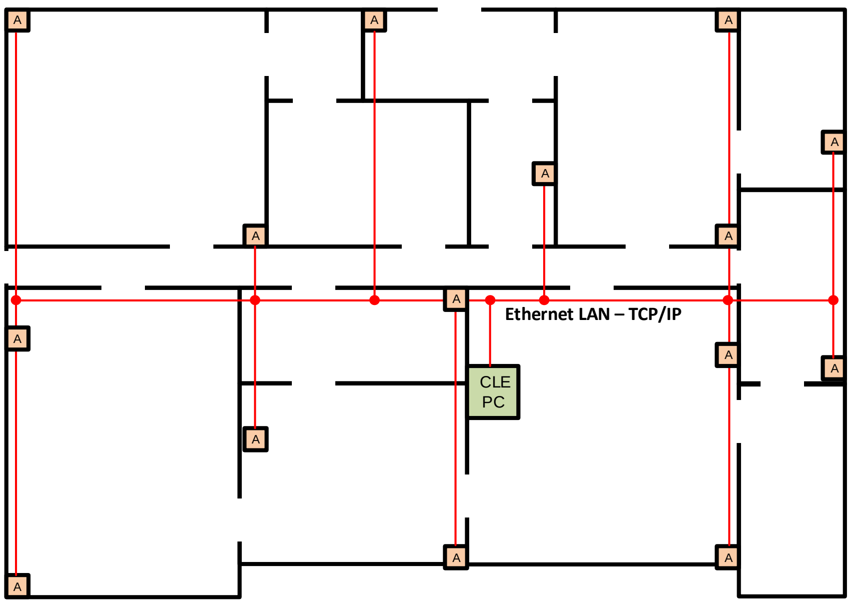


Figure 2: typical expanded RTLS system with central location engine (CLE)

Let's say the ranging exchange slot is of time T_S including the guard time (G) and the system is required to handle N tags. The scheduling frame then consists of at least N slots of time T_S so the minimum scheduling frame is then N x T_S in duration, which defines the fastest update rate for each tag in the system. Where a slower update rate is chosen, e.g. to save tag power, then either the slots can be made larger or the number of slots can be increased with the extra slots unused and available for expansion.

A new/undiscovered tag sleeps and awakens to "blink" with some (long) periodicity and waits for a response. The blink is essentially a message identifying the tag and its availability for ranging interaction with the RTLS. When the RTLS infrastructure sees the tag it sends a reply telling the tag its assigned slot, (e.g. this could be by telling it how long to sleep before it should awaken and perform the ranging exchange). The infrastructure also configures its anchors, based on which heard the blink, to select which anchors should respond to this tag when it sends its *Poll* and in what order, i.e. which are the first, second and third responder anchors.

In implementing this, all anchors could report to the CLE whenever they hear a tag blink or any tag message and include an estimate of the received power of each message which can serve as a crude measure of how close the tag is to that anchor. The CLE can use this information and/or the last calculated position for the tag to decide which anchors should participate in the next ranging exchange. Given that the interactions from anchor to CLE and back to anchors may take some appreciable time (perhaps 100s of milliseconds) the decision of which anchor should respond to the tag and which anchors the tag is told to range to will typically be based on an earlier blink or scheduling frame interval.

The CLE then can collect data from all anchors over a scheduling frame interval, and inform the anchors which should participate in the next ranging exchange for each tag. The CLE can also supply slot timing update information to a selected anchor (probably the first one to respond to the tag) which the anchor includes in its response to the tag, so it is updated to set/keep its slot aligned.

A tag blink is then responded to by an anchor that the CLE has designated (probably the closest one) to assign the tag a slot, (i.e. update the tag's sleep to set its activity period to its designated slot in the scheduling frame), and allow the tag to begin periodic ranging in its assigned slot.

The scheduling algorithm is aiming to keep the tags' ranging exchanges each in their own scheduling frame slot. Across a larger area different anchors will be participating at different times in ranging with different tags, and in the process also updating and maintaining their slot positions. To successfully maintain the tag slots, the anchors must base the scheduling updates on a common view of time. Since anchors are reporting the reception of tags messages to the CLE (as described above for the purpose of deciding which anchors should range with the tag) it is a small extension to also include the message arrival time data in the reports to the CLE. The CLE can then monitor the anchor specific local arrival times of the same tag message at adjacent anchors and can periodically send offset adjustments to individual anchors to adjust the slot timing updates they send to the tags. This method is assuming that the timing precision required for slot synchronisation is much less than the TOF between the anchors (which is typically less than 500 ns).

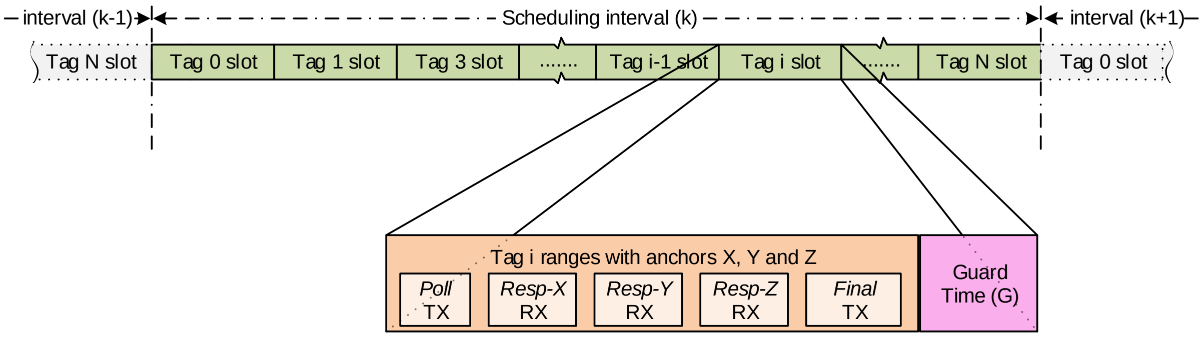


Figure 3: example of a scheduling (super-frame) interval

Figure 3 shows an example of a slotted system. Each tag has a slot where it wakes up and ranges with three anchors, after which the tag sleeps for another complete scheduling interval period. The guard time (G) should be set to account for all imprecisions in the sleep timing and any network scheduling precision being achieved across the larger area implementation.

The RTLS monitors each tag's active period and updates its sleep time to keep it from drifting out of its assigned slot.

In a larger system, spread over a campus of sufficient size that tags and anchors in one area are not in communications range of tags and anchors in another area, it would be possible for the CLE to manage the scheduling intervals for those separate areas to assign/reuse the same intervals of air-time in these separate areas without risking mutual interference between them. This strategy would be aiming to give a higher overall update rate, but it does add a complication to the system control since, as tags roam around, it would be necessary to dynamically increase/decrease the scheduling frame period to incorporate extra slots when tags congregate in different areas. This would naturally reduce the update rate for each tag in the lengthened superframe as their slots are occurring less frequently.

Individually tags need to have unique IDs so that they can be identified (§ 4 below discusses the allocation of unique IEEE EUI-64) however for TWR RTLS is desirable to use 16-bit addressing to make the messages shorter (saving air-time and power), so while an initial blink identifying a tag must use the full 64-bit address it may be desirable that a 16-bit address is assigned for use in the ranging exchanges and subsequent communications. In this case the response to a blink that tells the tag to being ranging it its slot might also assign a 16-bit address to the tag. The CLE would then employ a "database" that relates the tag's temporary 16-bit address to the tag's actual unique ID and the associated the person/asset.

3.4 International standard ISO/IEC 24730-62:2013

The international standard ISO/IEC 24730-62:2013 [11] is an air-interface standard for RTLS systems. While this standard's mandatory operation is a simple blink aimed at TDOA RTLS use, the standard also defines (optional) two-way communications messages and procedures, including support for a two-way ranging exchange. This could be used to implement a two-way ranging RTLS. In particular when a tag blinks in 24730-62 the defined response can assign a temporary 16-bit address to the tag and tell it to perform ranging to the responding anchor, and each ranging exchange is completed with a message either ending the tag's ranging activity (optionally giving it a new blink period, i.e. defining its scheduling interval) or telling the tag what anchor to range to next (supplying the 16-bit address of that anchor). The 24730-62 standard thus can be used to implement a network of tags that blink in their defined slots and in response are told which anchors around them to range to before finally being updated with a new sleep time (blink period) to wait until their next interaction (i.e. defining the scheduling frame).

Decawave participated in the working group that defined the ISO/IEC 24730-62 standard. We would be delighted to hear of customers implementing solutions following this standard. Decawave advocates a standards based approach to RTLS because this helps with marketing RTLS products to large organisations who often mandate standardised solutions. We especially <u>request customers</u>, who implement solutions based on this standard, whether TDOA or TWR, to <u>please let your national standards organisation know that you are using ISO/IEC 24730-62:2013</u> in your products, as national bodies vote every 5 years on each standard and if it is not being used in sufficient (5) countries it will be withdrawn as an international standard.

4 UNIQUE ADDRESSES – ASSIGNMENT AND HANDLING

In the TREK1000 the addresses assigned to the anchor and tag nodes are determined by the setting of physical switches. This was done for expediency and simplicity of operation. While this approach is fine for the purposes of evaluation, it is not suitable for wider deployment where to avoid confusion every radio node should have its own unique address assigned. There are two questions then: firstly, how to get and assign addresses to the nodes, and secondly, how to learn and use the addresses in the context of the RTLS system. These questions are addressed below.

4.1 Unique 64-bit address via IEEE registration

The DW1000 is designed to follow the IEEE Std 802.15.4™-2011 standard, in particular implementing the physical layer defined by the UWB PHY clause of the standard. For the 802.15.4 MAC layer, devices should have a unique 64-bit universal address, also known as a 64-bit Extended Unique Identifier (EUI-64). The IEEE has a Registration Authority service, which may be found at the link http://standards.ieee.org/develop/regauth), that is responsible for allocating blocks of addresses (Small, Medium and Large) for a fee. The options here are given in Table 4 below.

Table 4: IEEE MAC address registration sizes

MAC Address Block	Size of identifier allocated (bits)	N bits to complete the 64-bit address	Number of addresses in block ($=2^N$)	Cost of registration
Small	36	30	> 1 billion	US $645
Medium	28	36	> 68 billion	US $1,545
Large	24	40	> 1 trillion	US $2,575

The DW1000 has a one-time-programmable memory area that may be programmed with the EUI-64 during customer's product manufacturing process, (please refer to the DW1000 user manual [2] for details), or manufacturers may choose to store the EUI externally to the DW1000.

For development purposes it is possible to generate a DW1000 unique number by combining the *Lot ID* & *Part Number* values programmed into the DW1000 during its manufacture. The *DW1000 device driver API guide* [2] details the *dwt_getpartid( )* and *dwt_getlotid( )* API calls that may be used to access these parameters. However please note that there is no guarantee that an address generated in this way will not conflict with someone else's implementation and for that reason it is recommended that customers buy a block of addresses from the IEEE Registration Authority for their production items.

5 HARDWARE – DESIGN OF TAGS AND ANCHORS

The board employed in the TREK1000 is the same as the board employed on the EVK1000, and is designed for evaluation of the DW1000 IC. It is not a suitable to directly act as a commercial RTLS tag or anchor. Development of a commercial product will require the design of new hardware and software depending on the capabilities needed in the final system. As an introduction to this the following lists some of the main considerations for these developments.

Anchor design:

- The anchors are the fixed elements of the system.
- Anchors need to be boxed units packaged for mounting on wall, ceilings or poles depending on the application. They may need to be ruggedised or waterproofed depending on the target application/market
- Anchors are typically powered on all the time and mostly in receive mode ready and waiting to interact with tags. This generally means that the anchor power source has to be based on the mains electricity supply.
- Anchors will typically need a "*backhaul*" to communicate their ranging results to a central point for solving the RTLS results. This *backhaul* to the *central location engine* (CLE) could be implemented using a wired communications means like Ethernet, or Wi-Fi, or using UWB. Depending on the complexity of the system this communications link will probably also have a role in controlling the anchors and via them controlling the tags.
- In case of using UWB for the *backhaul* (and control) this traffic needs to be considered as part of the air-utilisation which will significantly reduce the number of tag locations per second that the system can perform, (see discussion of air utilisation in § 2.2).
- If Ethernet is used for the *backhaul* it may be worth considering using Power-over-Ethernet (PoE) to supply power and communications via one cable.
- The UWB antenna is a key part of the anchor. The design of UWB antennas can be challenging since their performance is affected by the surroundings including the anchor hardware and enclosure and the surfaces on which the anchor is mounted. Antenna selection/design is discussed further in Decawave application note APH007 [5].

Tag design:

- The tags are the mobile elements of the system that are tracked and located.
- The tag could be a subsystem incorporated into some larger mobile platform design or it may be a standalone unit designed for mounting on the item to be located.
- Tags generally need to have a small form factor, and their enclosure may need to be ruggedised to suit the target application.
- Tags are mobile so are battery powered. The battery system needs to cope with the peak demands of the ranging transaction, and the burst nature of the power demands. Clearly the whole tag system including the microprocessor and other active components needs to be considered, and in the case of infrequent ranging activity, the whole system sleep current is also significant. A rechargeable battery may be appropriate. Battery and power source selection considerations are discussed further in Decawave application note APH005 [4].
- The antenna is also a key part of the tag and generally for a tag there is much less freedom of size and position than in the anchor so this can be especially challenging. The whole tag design, its enclosure and the items it is intended to be mounted on, need to be considered as part of the tag antenna design. Antenna selection/design is discussed further in Decawave application note APH007 [5].

Production:

- Depending on the operating range requirements and on the ranging accuracy demanded of the final system. It may be necessary to individually calibrate tags and anchors during their manufacture for both output power and antenna delays. Decawave application note APS012 [7] describes the production testing of DW1000 based product including the calibration of transmit power and a basic antenna delay calibration, while application note APS014 [9] deals specifically with antenna delay calibration.

UWB Regulations:

- Depending on the target market and application, different UWB regulations apply. These typically may determine the operating channel to be used (a potential factor in the UWB antenna design/selection), what transmit power is allowed, and possibly how frequently a device can transmit. There is a discussion of the low-duty cycle requirement in § 2.3 above, but please consult with your Decawave representative for more details of UWB regulations.
- In many markets (e.g. USA, Europe, etc.) radio equipment has to be certified before it can be sold commercially. Decawave application note APR002 [6] discusses the certification process in the USA. For additional markets please consult your Decawave representative.

6 CONCLUSIONS

This application note presented an overview of the main issues a developer will face in moving from the TREK1000 to a commercial RTLS product offering. The reader who gets this far having gone through everything above should have a very good appreciation of what is involved in developing a commercial Two-Way-Ranging RTLS product.

Decawave would be delighted to discuss these matters further with any customers who plan a product development based on DW1000. Our sales representatives (whose contact details can be found in § **Error! Reference source not found.**) are available to help you with any additional questions.

We also welcome feedback on the subject matter of this application note, and any suggestions for its improvement.

7 Abbreviations and Acronyms

Abbreviation	Full Name	Explanation
DW1000		Decawave's UWB transceiver IC.
IC	integrated circuit	An electronic circuit integrated onto a silicon chip.
PHY	physical layer	The term PHY relates to the physical layer implementation including the modulation, encoding and transmission of data.
PSR	preamble symbol repetitions	The number preamble symbol repetitions, defines the length of the preamble sequence. Each symbol is approximately 1 microsecond long. Typically, for long range one would use a longer preamble and a lower data rate, and for lower power higher density applications a short preamble and a higher data rate would be used.
RTLS	real time location system	A system able to locate mobile tags with respect to an infrastructure of fixed anchor nodes of known position.
RX	Receive	This may refer to the act of receiving, the data that is received, or some related parameter.
	Scheduling frame	This is name we give to the period of time that we divide into slots in which individual ranging exchanges take place.
TCXO	temperature compensated crystal oscillator	A crystal oscillator generally of high specification in terms of its PPM tolerance guaranteed over an operating temperature range.
TDOA	time difference of arrival	The difference in time of arrival of a message at a pair of synchronised nodes, or the general mechanism of location given the TOA of a tag blink at multiple synchronised anchor nodes.
TOA	time of arrival	The time of arrival of a message or its report.
TOF	time of flight	The time it takes a message to go through the air between two nodes. Typically a TOF will be the result of a TWR measurement.
TX	transmit	This may refer to the act of transmitting, the data that is transmitted, or some related parameter.
TWR	two-way ranging	A scheme that estimates the round trip delay (halving it to get the one way trip time) between two nodes to give an estimate of their separation distance.
UWB	ultra-wide band	A radio scheme employing channel bandwidths typically from 500 MHz upwards. The DW1000 is an impulse radio (IR) employing pulses of 2 ns for its 500 MHz bandwidth IR-UWB transmission.

8 References

Reference is made to the following documents in the course of this Application Note:

Table 5: Table of References

Ref	Author	Date	Version	Title
[1]	Decawave		Current	DW1000 DATASHEET
[2]	Decawave		Current	DW1000 USER MANUAL
[3]	Decawave		Current	DW1000 DEVICE DRIVER APPLICATION PROGRAMMING INTERFACE (API) GUIDE
[4]	Decawave		Current	APH005 APPLICATION NOTE – DW1000 POWER SOURCE SELECTION GUIDE Selecting the power source for your DW1000 based product
[5]	Decawave		Current	APH007 APPLICATION NOTE – Antenna Selection / Design Guide for DW1000
[6]	Decawave		Current	APR002 APPLICATION NOTE – UWB PRODUCT CERTIFICATION PROCESS IN THE USA How to take your Decawave DW1000 / DWM1000 based product through the certification process with the FCC
[7]	Decawave		Current	APS012 APPLICATION NOTE – PRODUCTION TESTS FOR DW1000-BASED PRODUCTS Typical production tests that should be applied to DW1000-based products
[8]	Decawave		Current	*DecaRangeRTLS ARM Source Code Guide* A description of the TREK1000 (ARM) code
[9]	Decawave		Current	APS014 APPLICATION NOTE – ANTENNA DELAY CALIBRATION OF DW1000 BASED PRODUCTS AND SYSTEMS
[10]	IEEE Computer Society		2011	IEEE 802.15.4 Standard for local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks. Available from http://standards.ieee.org/
[11]	ISO/IEC JTC1	Sept 2013	First edition	International Standard: ISO/IEC 24730-62 *Information technology — Real time locating systems (RTLS) — Part 62: High rate pulse repetition frequency Ultra Wide Band (UWB) air interface.* Available from: http://www.iso.org/

9 Document History

Table 6: Document History

Revision	Date	Description
1.00	31st March 2015	Initial release for TREK1000 market introduction.
2.00	2st October 2015	Substantial re-write as a result of TREK1000 optimisation that moved from individual two-way ranging exchanges to a scheme where ranging to three anchors can be achieved with just five messages.
2.01	23rd October 2015	Changed the 6.81 mode number of slots to 10 – to match the TREK ARM rev 2.10 Code release.
2.2	12th April 2017	Additional clarification in section 1.2 on requirements when expanding the TREK1000 system.
2.3	15th July 2018	Updated with new LOGO

10 FURTHER INFORMATION

Decawave develops semiconductors solutions, software, modules, reference designs - that enable real-time, ultra-accurate, ultra-reliable local area micro-location services. Decawave's technology enables an entirely new class of easy to implement, highly secure, intelligent location functionality and services for IoT and smart consumer products and applications.

For further information on this or any other Decawave product, please refer to our website www.decawave.com.