TTK1000 RTLS System Architecture

Real Time Location System Scalable Software Architecture

Version 1.0

This document is subject to change without notice

TABLE OF CONTENTS

LIST OF FIGURES

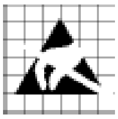
DOCUMENT INFORMATION

Disclaimer

Decawave reserves the right to change product specifications without notice. As far as possible changes to functionality and specifications will be issued in product specific errata sheets or in new versions of this document. Customers are advised to check the Decawave website for the most recent updates on this product

LIFE SUPPORT POLICY

Decawave products are not authorized for use in safety-critical applications (such as life support) where a failure of the Decawave product would reasonably be expected to cause severe personal injury or death. Decawave customers using or selling Decawave products in such a manner do so entirely at their own risk and agree to fully indemnify Decawave and its representatives against any damages arising out of the use of Decawave products in such safety-critical applications.



Caution! ESD sensitive device.

Precaution should be used when handling the device in order to prevent permanent damage

1 DECAWAVE TDOA RTLS OVERVIEW

Decawave's RTLS offering is based on the Decawave's DW1000 IEEE 802.15.4 UWB (Ultra Wide Band) wireless transceiver IC. The RTLS's function is to locate mobile *tags* with respect to fixed position *anchor* nodes. The tags send periodic *blink* messages, which are received and time-stamped at the anchor nodes. Each anchor node then sends the time-of-arrival reports (for the blink messages it receives) to a Central Location Engine (CLE) where, assuming sufficient anchors report the TOA of the blink message, the CLE uses the Time Difference of Arrival (TDOA) of the blink message at the anchors to estimate the sending tag's location. This solving of the TDOA data is called multilateration.

Figure 1 shows a deployment example of anchors in an office environment.

Tags are intended to be low power, low complexity, low cost and usually battery powered mobile devices, while anchors act as fixed location reference points and are generally mains powered. Tags periodically send "blink" messages which are received by all anchors in range.

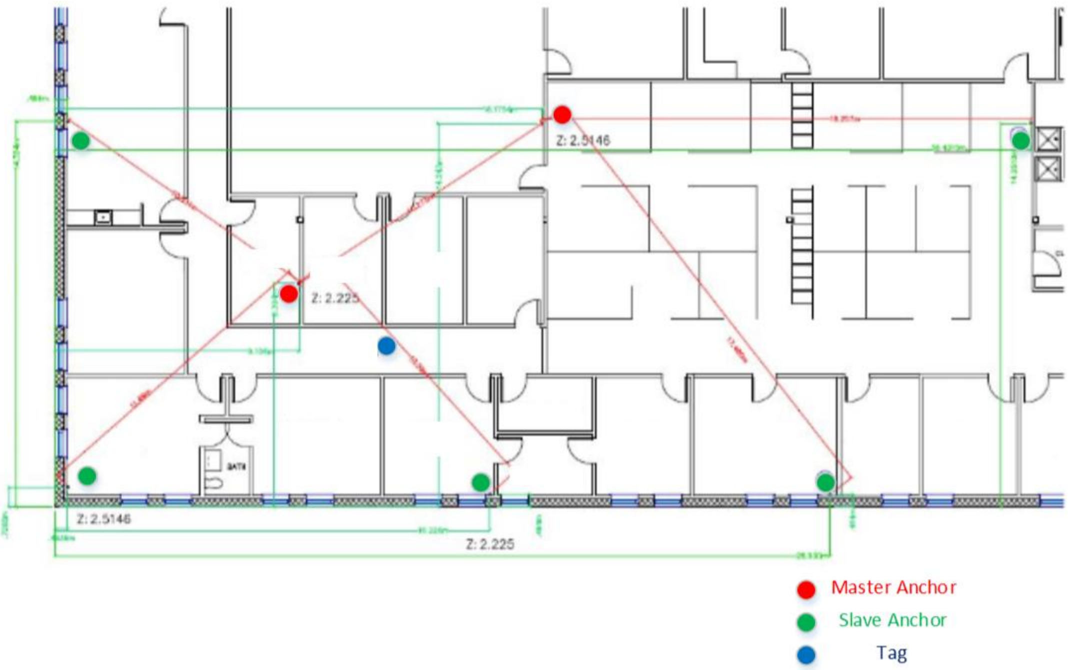


Figure 1 – Example office configuration

For TDOA to work the blink arrival times at all anchors need to be using the same absolute time so that the time differences are meaningful. This is called time or clock synchronisation of the anchor nodes and is typically achieved via wired clock distribution to the anchor nodes. As an alternative to using a wired clock the Decawave TDOA RTLS software includes a wireless clock synchronisation algorithm that employs UWB messages sent between anchors and runs a software-based clock tracking and blink timestamp correction functionality. Anchors configured as *Master* anchors transmit Clock Calibration Packets (CCP) periodically, while *Slave* anchors receive these CCP and report their reception to the CLE to track the relative clock drift between the sending master anchors and the receiving slave anchors. If an RTLS system has more than one master, we call this a multi-master system and to prevent

CCP collisions between CCP transmissions, a “secondary” master can delay the sending of its CCP by a configured lag time after the reception of a CCP from a “primary” master anchor which they ”follow”.

Every anchor reports the reception of blinks and CCP’s to the CLE via TCP/IP. The (synchronization) tracking of the anchors clocks and (synchronization) correction of the blink message timestamps to a common time base is performed by the CLE.

The CLE also performs the multilateration to estimate the tags’ locations.

Figure 2 below shows a minimal system the UWB messages (blinks and CCP) and Ethernet communication (reports) for a system with 4 Anchors, 1 Tag, CLE and Controller. A2 is the master anchor and A1, A3, A4 are slave anchors. T1 is the tag and CLE runs on the PC.

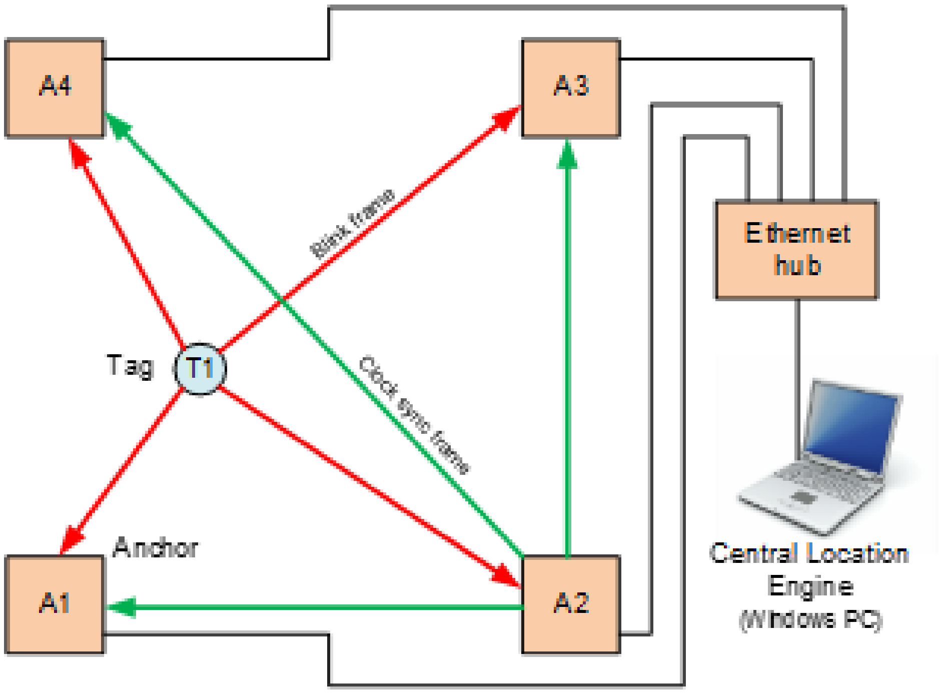


Figure 2 – Minimal RTLS system (wireless synchronisation)

2 RTLS Operation

This briefly describes the salient features and functionality of the anchor, tag, controller and CLE. Please see the Bibliography in section 3 below for the individual documents relating to these components which the reader can refer to for more detailed information.

2.1 Tag

The Tag transmits a blink frame, which is a 12 byte frame as defined in IEEE 802.15.4e-2012 standard and used in the ISO/IEC 24730-62 international standard.

1 octet FC	1 octet	8 octets	2 octets
0x56	Seq. Num	64-bit Tag ID	FCS

Figure 3 – Minimal Tag IEEE blink frame

Blinks are transmitted periodically. These blinks are received by all the anchors in the vicinity of the tag. Upon reception of the blink packet from a tag, the anchor will create a *Blink Report* that is sent to the CLE.

2.2 Anchor

Anchors spend most of the time in the receive mode. Master anchors periodically stop receiving to transmit a CCP message, which are received by the slave anchors and reported to the CLE (via the **CCP Rx Reports**). The master's CCP transmission time is reported to CLE via the **CCP Tx Report**. The CLE uses the CCP TX and RX reports to track the timing drift between the anchors, and this information is used by the CLE to correct (synchronise) blink message timestamps to a common time base.

When an anchor receives a tag blink message, it reports the blink reception time via a **Blink Report** to the CLE.

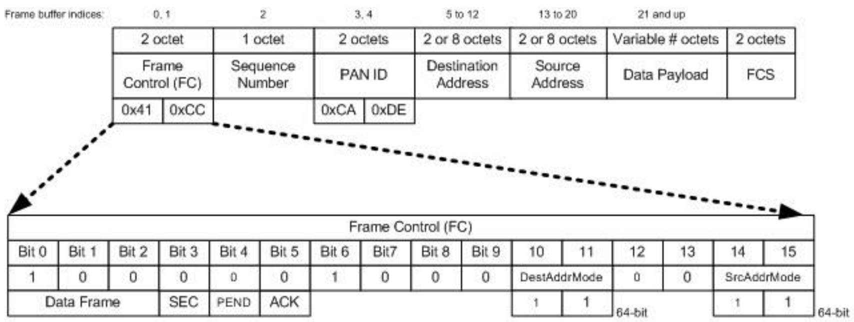


Figure 4 – General UWB message format

There may be more than one master anchor in a system to cover a large area of operation, as shown in Figure 5 and Figure 6. In this case, the CCP transmissions of various masters are staggered in a synchronized manner to avoid collision between the CCP messages. The paragraphs below describe CCP master sync scheme in general, please review the *Anchor Software Guide* for more details.

2.3 Clock Synchronization Scheme

For correct operation of TDOA RTLS systems all the anchors' blink timestamps need to be synchronized to a common time base to enable the TDOA multilateration compute sensible tag location estimates. In order to synchronize the times, anchors designated as master anchors periodically transmit Clock Calibration Packet (CCP) UWB Frames. Currently this is done every 150 milliseconds. Using the CCP transmission and reception time-stamps, the CLE runs a Kalman filter to continually track and estimate the time difference between the master and slave anchors. This information is then used to correct the blink timestamps.

To cover a large area, it is possible and necessary to employ more than one master (Figure 5), where each master is used as reference to track the clock drift of its neighbouring slave anchors and correct their blink timestamp reports. Within the CLE each anchor may be designated to track more than one master, (i.e. for any particular anchor the CLE can be processing the CCPs it receives from more than one master and the CLE will run a Kalman filter to track the relative clock drift for each resulting master-slave pair).

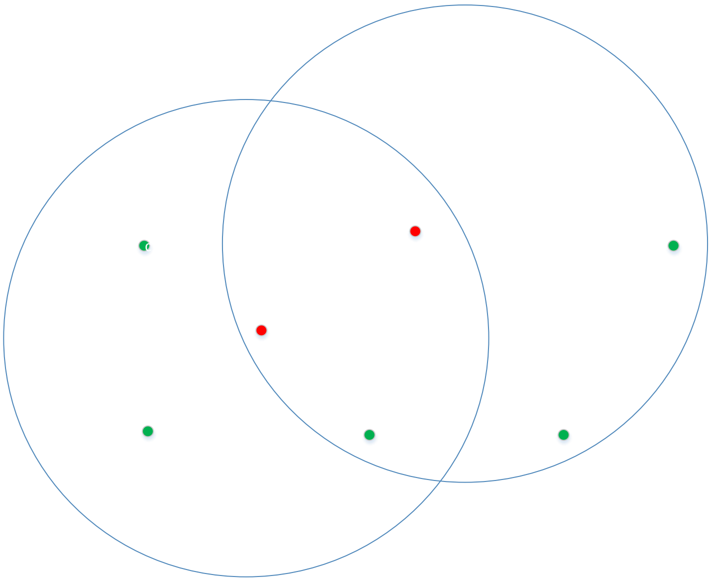


Figure 5 – Example of two master (in red) configuration

In order to avoid collisions of CCP packets, the masters are designated as primary and secondary masters. A secondary master follows its designated primary master and transmit its CCP at a configured lag time after the reception of the CCP from its primary master. There may be more than one secondary master, in which case they should be configured so that their CCP transmit times are staggered to avoid collision.

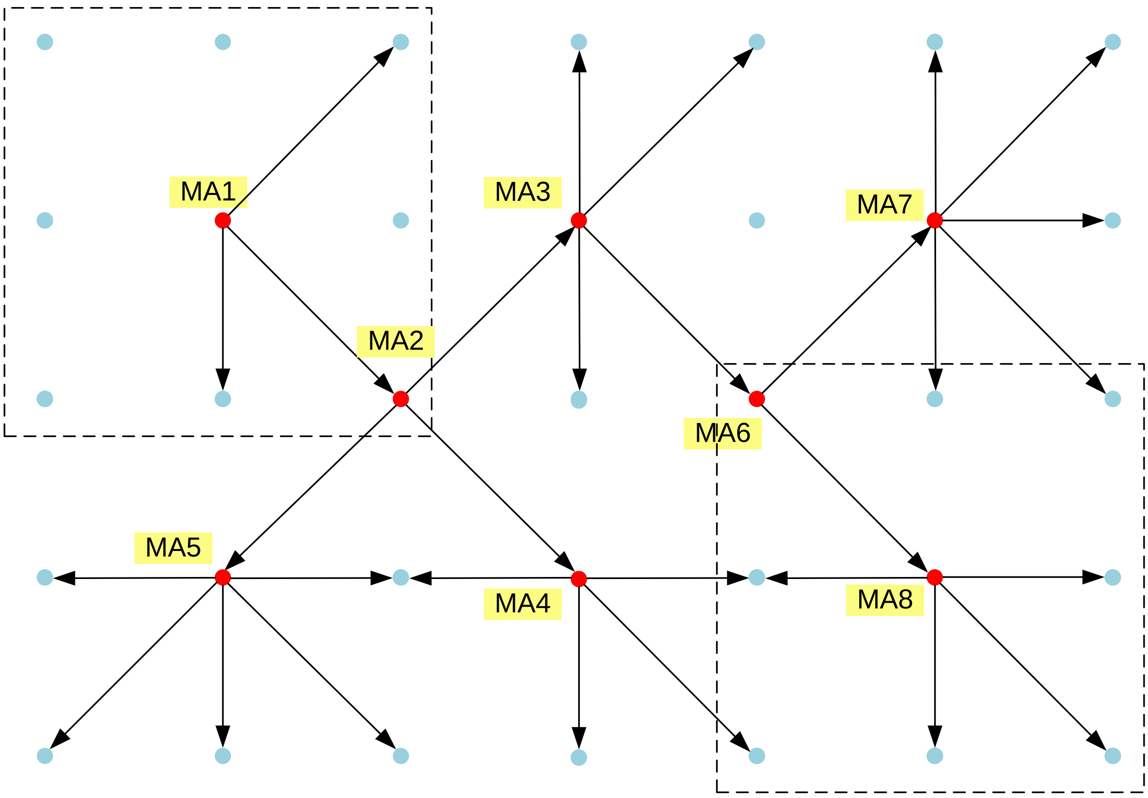


Figure 6 - example of a master anchor arrangement in large system

Figure 6 shows a possible anchor arrangement, master anchors sending CCPs are red, and slave anchors only receiving CCPs are blue.

The selection of master anchors in Figure 6 has been made to ensure each pair of slave anchors is in the range of a master, and that each secondary master is in range of another primary master from which it can time its CCP transmission.

The operational summary of this example system is as follows:

- Every master anchor MA1-MA8 sends a master CCP packets at the configured period.
- M1CCP (CCP from MA1) is received by 8 surrounding anchors, including MA2. Each slave (including MA2) reports the M1CCP Time-of-Arrival (TOA) to the Central Location Engine (CLE) which runs 8 separate Kalman filters (one for each of these slave anchors).
    - These Kalman filters will allow tag blink TOA from any pair of these 9 anchors to be converted into a single time base to yield a TDOA that may be used for multilateration.
- Anchor MA2 is configured to also act as a secondary master to send CCP (M2CCP) at a Time Lag interval (TL) following the receipt of the primary master's M1CCP. The M2CCP is similarly received by the 8 anchors surrounding MA2 (including MA3, MA4 and MA5) and the M2CCP TOA reports for these give the CLE eight additional Kalman filters it could run.

- MA3, MA4 and MA5 are similarly also configured to also act as secondary master anchors and to send their CCP in staggered fashion following on from their receipt of their primary master's M2CCP. For example, MA3 is set to send its M3CCP after time TL, MA4 is set to send its M4CCP after time 2 x TL, and MA5 is set to send its M5CCP after 3 x TL.
- Similarly, anchor MA6 send its M6CCP at a specified time lag from the arrival of the M3CPP, and MA7 and MA8 send their M7CPP and M8CPP at specified time offsets from the arrival of the M6CPP. Each CCP transmission is received by 8 (potential slave) anchor nodes who report its TOA, enabling the CLE to run an additional Kalman filters.
- Other than for the initial master, MA1, each CCP transmission is synchronized to be timed from the arrival of another master anchor's CCP, to give a staggering sending (configured under control of the CLE) so that transmissions do not collide at any slave anchor receivers.

Figure 7 shows a possible staggering of CCP based on the example on the **Figure 6**.

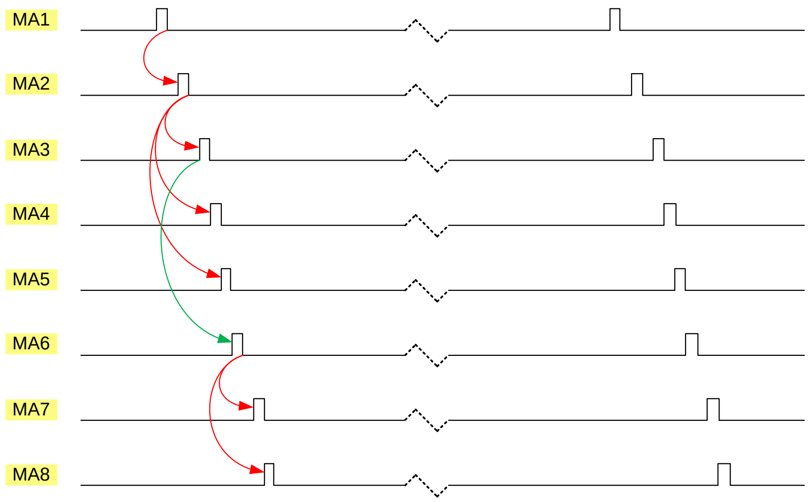


Figure 7 – Staggered CCP transmission times for example under discussion here

2.4 Central Location Engine

The Central Location Engine (CLE) is a PC application running on Windows. The CLE controls the anchors and processes the reports from the anchors. The RTLS system (anchors and CLE) is configured by the external control client application (Controller), which could be running on the same PC as the CLE or on a separate PC networked to it. The Controller sends a configuration commands to the CLE via a TCP/IP socket connection to configure the anchor positions and modes, the Kalman filters, selector algorithm, motion filters etc. (Please refer to CLE software documentation for more details). When anchor configuration (or reconfiguration) is necessary the CLE will send the appropriate commands to the anchors via TCP/IP also.

The clock tracking (Kalman), blink timestamp correction and the multilateration algorithms are run by the CLE. When the CLE solves the multilateration to estimate the tag's location, the tag location is available to client applications via a socket connection, which is then used by the

Controller to display the results. A separate Display client implementation, which has a displaying functionality only, is also available as an example of a stand-alone client.

The main blocks of the CLE design are described below. For more details, please refer to the CLE Architecture guide.

- Anchor "Static IP" configuration service
    - Anchors connected to the Ethernet network and they awaiting of connection from CLE (via POSIX-compatible socket).
    - Before connecting to the anchor, CLE should know its IP address. This can be done either via "Static IP" configuration or via Zero-network configuration protocol, such as mDNS.
    - When a configuration file "dns_cle.cfg" is loaded by CLE, it is used to resolve anchors IP addresses with respect to their unique ID and the Zero-network configuration service, i.e. mDNS will not be used.
- Anchors discovery via mDNS service
    - If "Static IP" configuration is off (no "dns_cle.cfg" file), then the multicast Domain Name System (**mDNS**) client is connects to the external Bonjour service, which shall be running in the system.
    - The Bonjour service performs a periodic transmission of mDNS Request packets to the network. If anchors are not yet connected to the CLE, they will respond to the request, this will enable to the CLE to find the IP addresses of anchors and further connect to them if they belong to its local RTLS network.
- Anchor Connection Handler
    - Handles the **CCP Tx, CCP Rx** and **Blink Reports** from the anchors.
    - Clock interconnection algorithm. This aligns (pairs up) received CCP RX reports (from slave anchors) with their respective master CCP TX timestamp reports, using the frame sequence numbers of the CCPs.
    - Queues the reports for Kalman filter updates or Multilateration.
- Anchor Reconnection service
    - This performs periodic keep alive service to the connected anchors.
    - It is testing the connection by sending a short keep-alive message to every connected anchor in the network. If TCP/IP stream is broken (e.g. anchor is switched off), then socket will be closed and reconnection service will perform a periodical ICMP ping to desired IP address. On successful reception of ICMP pong, it will attempt to connect to the anchor using standard connection method, which will re-construct the anchors network.
- Wireless Clock Sync Tracking Algorithm
    - Kalman filters are used to track each slave anchor's clock offset with respect to their designated master anchors' clocks.
- Selector

    - Manages the blink TOA Reports from the anchor and queues them for processing
    - Finds the best possible Zeroth anchor for multilateration
    - Requests multilateration
    - Maintains tag statistics
- Data Server
    - Accepts connection from clients requesting tag location data
    - Reports the results to the connected clients
- Solver
    - Solves the tag position using the multilateration algorithm based on the tag blink TDOA data
- Filters
    - Motion filter, etc.

2.5 Controller

The Controller is used to configure the RTLS System. The Controller connects to the CLE (via TCP/IP "Control" POSIX-compatible socket) and configures the RTLS System in the CLE. The GUI provides options to configure each anchor and the CLE.

The Controller interface may be used for initial configuration of the system, after starting up of running of RTLS system, the Control application does not need to be active.

The Controller to CLE interface is an xml based command and response exchange sent over TCP/IP. Developers may implement this interface in their existing solutions.

The Controller-CLE interface is described in more detail in the document *TTK1000 RTLS CLE Controller API Spec*. For details, please refer to the *Controller Software Guide*.

2.6 Displayer

The Displayer application is used to display the results of tag locations. The Display application connects to the CLE (via TCP/IP "Display" POSIX-compatible socket) and receives estimated tag location from the CLE.

On reception of every Tag location, the CLE is also providing the set of anchors, which were chosen for current Tag location, such the customer's application can do feature filtering and improvements, based on this information.

It is up to developers to design the frontend application. For details, please refer to the *Display Software Guide*.

3 BIBLIOGRAPHY

Table 1 Table of References

Ref	Title
1	IEEE 802.15.4-2011 or "IEEE Std 802.15.4™-2011" (Revision of IEEE Std 802.15.4-2006). IEEE Standard for Local and metropolitan area networks— Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Computer Society Sponsored by the LAN/MAN Standards Committee. Available from http://standards.ieee.org/
2	TTK1000 _RTLS_CLE_Controller_API_Spec.pdf
3	TTK1000 _Controller_Software_Guide.pdf
4	TTK1000 _Display_Software_Guide.pdf
5	INTERNATIONAL STANDARD ISO/IEC 24730-62:2013 Information technology — Real time locating systems (RTLS) — Part 62: High rate pulse repetition frequency Ultra Wide Band (UWB) air interface

4 Document History

Table 2 Revision history

Revision	Date	Description
1.0	06-Sep-2018	Initial release, matches RC_06 release

5 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.