SOURCE CODE GUIDE

DECARANGERTLS PC SOURCE CODE

Understanding and using the DecaRangeRTLS PC source code

Version 2.1

This document is subject to change without notice.

TABLE OF CONTENTS

1 INTRODUCTION 6

2 DECARANGERTLS PC OVERVIEW 9

2.1 RTLSDISPLAY APPLICATION 9
2.2 SERIAL/COM PORT CONNECTION 9
2.3 RTLS CLIENT 10
2.3.1 Trilateration Function 10
2.3.2 Range Report Format 11
2.4 MAIN DISPLAY WINDOW 11
2.4.1 Graphics 11
2.4.2 Connection Widget 13
2.4.3 Status Bar 13
2.4.4 View Settings 13
2.4.5 Minimap View 13
2.4.6 Drop Down Menus 14
2.5 WIDGETS AND VIEWS 16
2.5.1 Graphics View 16
2.5.2 Graphics 16
2.5.3 Minimap 17
2.5.4 View Settings 17
2.6 TOOLS AND UTILITIES 17
2.6.1 Abstract Tool 17
2.6.2 Origin Tool 17
2.6.3 Rubber Band Tool 17
2.6.4 Scale Tool 18
2.6.5 QProperty Model 18

3 QT SDK INSTALLATION GUIDE 19

4 BIBLIOGRAPHY 21

5 REVISION HISTORY 22

6 FURTHER INFORMATION 22

LIST OF TABLES

TABLE 1: LIST OF SOURCE FILES IN THE DECARANGERTLS ARM APPLICATION 7
TABLE 2: TABLE OF REFERENCES 21

LIST OF FIGURES

FIGURE 1: TREK1000 SYSTEM COMPONENTS (WITH 2 TAGS) 6
FIGURE 2: DECARANGERTLS PC/RTLSDISPLAY APPLICATION - MAIN FUNCTIONAL BLOCKS 7
FIGURE 3: DECARANGERTLS APPLICATION – UI COMPONENTS 9
FIGURE 4: TRILATERATION FUNCTION - INTERSECTION OF 3 SPHERES 10
FIGURE 5: TAG TABLE 12
FIGURE 6: ANCHOR TABLE 12
FIGURE 7: ANCHOR'S TAG CORRECTION TABLE 12
FIGURE 8: VIEW SETTINGS WIDGET 14
FIGURE 9: ABOUT WINDOW 16

DOCUMENT INFORMATION

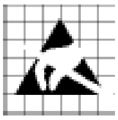
Disclaimer

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

Copyright © 2016 Decawave Ltd

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

DISCLAIMER

This Disclaimer applies to the DecaRangeRTLS PC source code (“Decawave Software”) provided by Decawave Ltd. (“Decawave”).

Downloading, accepting delivery of or using the Decawave Software indicates your agreement to the terms of this Disclaimer. If you do not agree with the terms of this Disclaimer do not download, accept delivery of or use the Decawave Software.

Decawave Software is solely intended to assist you in developing systems that incorporate Decawave semiconductor products. You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your systems and products. THE DECISION TO USE DECAWAVE SOFTWARE IN WHOLE OR IN PART IN YOUR SYSTEMS AND PRODUCTS RESTS ENTIRELY WITH YOU.

DECAWAVE SOFTWARE IS PROVIDED "AS IS". DECAWAVE MAKES NO WARRANTIES OR REPRESENTATIONS WITH REGARD TO THE DECAWAVE SOFTWARE OR USE OF THE DECAWAVE SOFTWARE, EXPRESS, IMPLIED OR STATUTORY, INCLUDING ACCURACY OR COMPLETENESS. DECAWAVE DISCLAIMS ANY WARRANTY OF TITLE AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO DECAWAVE SOFTWARE OR THE USE THEREOF.

DECAWAVE SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY THIRD PARTY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON THE DECAWAVE SOFTWARE OR THE USE OF THE DECAWAVE SOFTWARE WITH DECAWAVE SEMICONDUCTOR TECHNOLOGY. IN NO EVENT SHALL DECAWAVE BE LIABLE FOR ANY ACTUAL, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, INCLUDING WITHOUT LIMITATION TO THE GENERALITY OF THE FOREGOING, LOSS OF ANTICIPATED PROFITS, GOODWILL, REPUTATION, BUSINESS RECEIPTS OR CONTRACTS, COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION), LOSSES OR EXPENSES RESULTING FROM THIRD PARTY CLAIMS. THESE LIMITATIONS WILL APPLY REGARDLESS OF THE FORM OF ACTION, WHETHER UNDER STATUTE, IN CONTRACT OR TORT INCLUDING NEGLIGENCE OR ANY OTHER FORM OF ACTION AND WHETHER OR NOT DECAWAVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT OF DECAWAVE SOFTWARE OR THE USE OF DECAWAVE SOFTWARE.

You are authorized to use Decawave Software in your end products and to modify the Decawave Software in the development of your end products. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER DECAWAVE INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY THIRD PARTY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT, IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Decawave semiconductor products or Decawave Software are used.

You acknowledge and agree that you are solely responsible for compliance with all legal, regulatory and safety-related requirements concerning your products, and any use of Decawave Software in your applications, notwithstanding any applications-related information or support that may be provided by Decawave.

Decawave reserves the right to make corrections, enhancements, improvements and other changes to its software at any time.

Mailing address: -

Decawave Ltd.,
Adelaide Chambers,

Peter Street,
Dublin 8

Copyright (c) 01/04/2015 by Decawave Limited. All rights reserved.

1 INTRODUCTION

This document describes the DecaRangeRTLS PC application and its source code. The application connects to TREK1000 anchors or tags (running DecaRangeRTLS ARM application), and consumes the Time of Flight (TOF) reports coming from the anchors/tags and produces tag's location estimate based on the anchors' location.

The reader is encouraged to read the accompanying TREK documents, in particular the DecaRangeRTLS ARM Source Guide, and TREK1000 User Manual.

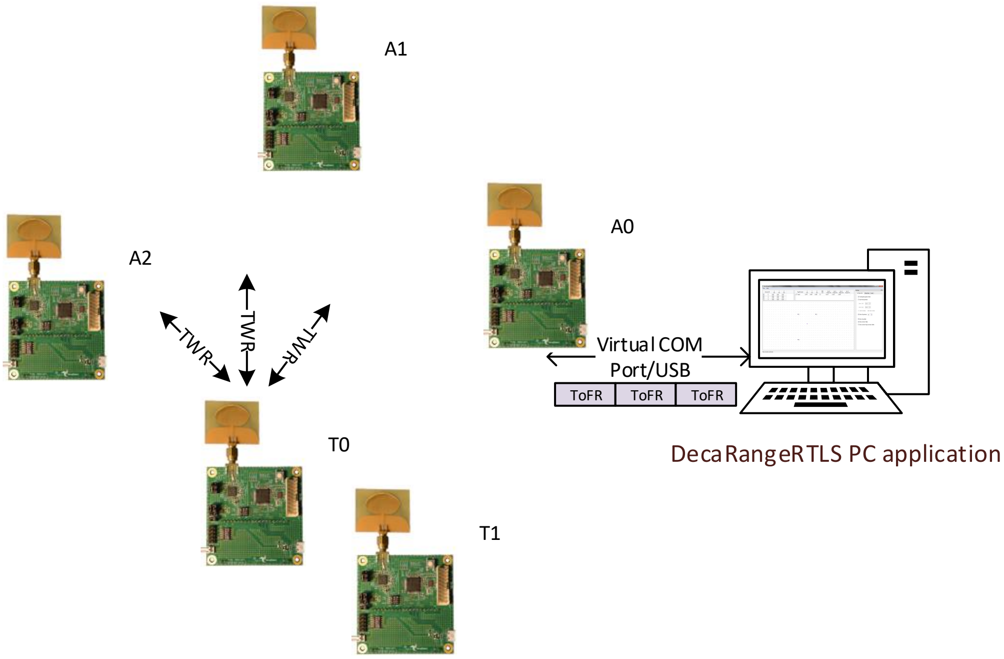


Figure 1: TREK1000 system components (with 2 tags)

The DecaRangeRTLS PC application is built with the Qt framework. Qt SDK Installation Guide describes how to set up and build the project. The main parts of the DecaRangeRTLS PC application are:

- Main window – this is the main user interface, it contains:
  - Drop down menus: View and Help
  - Table of known tags, anchors and the main display area, which shows position of tags and anchors on a floor plan.
- Serial connection – this is used to establish and monitor Virtual COM port connection with a TREK1000 Tag or Anchor over USB.
- RTLS client/Display client – this processes the TOF report messages from Virtual COM port connection, and sends location estimate to the main display.
- Anchor list – this is a list of known tags and it stores their properties (e.g. x, y, z coordinates etc.)
- Tag list – this is a list of known tags and it stores their properties (e.g. x, y, z coordinates etc.)
- Configuration – used for the use case settings and floorplan configurations
- Loading and Saving of anchor, tag parameters and view configuration

These are described in more detail in the following chapters.

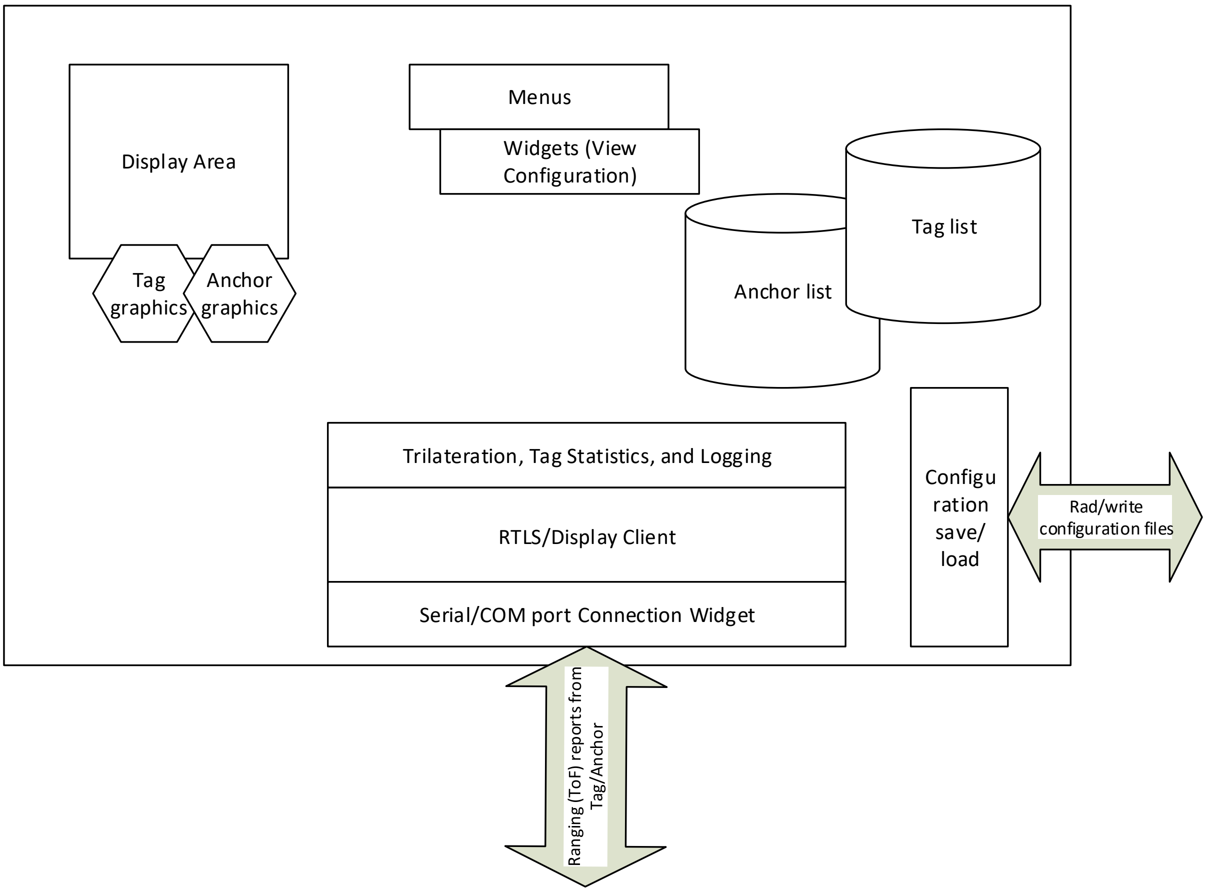

Display Area
Tag graphics
Anchor graphics
Menus
Widgets (View Configuration)
Anchor list
Tag list
Trilateration, Tag Statistics, and Logging
RTLS/Display Client
Serial/COM port Connection Widget
Configuration save/load
Rad/write configuration files
Ranging (ToF) reports from Tag/Anchor

Figure 2: DecaRangeRTLS PC/RTLSDisplay application - main functional blocks

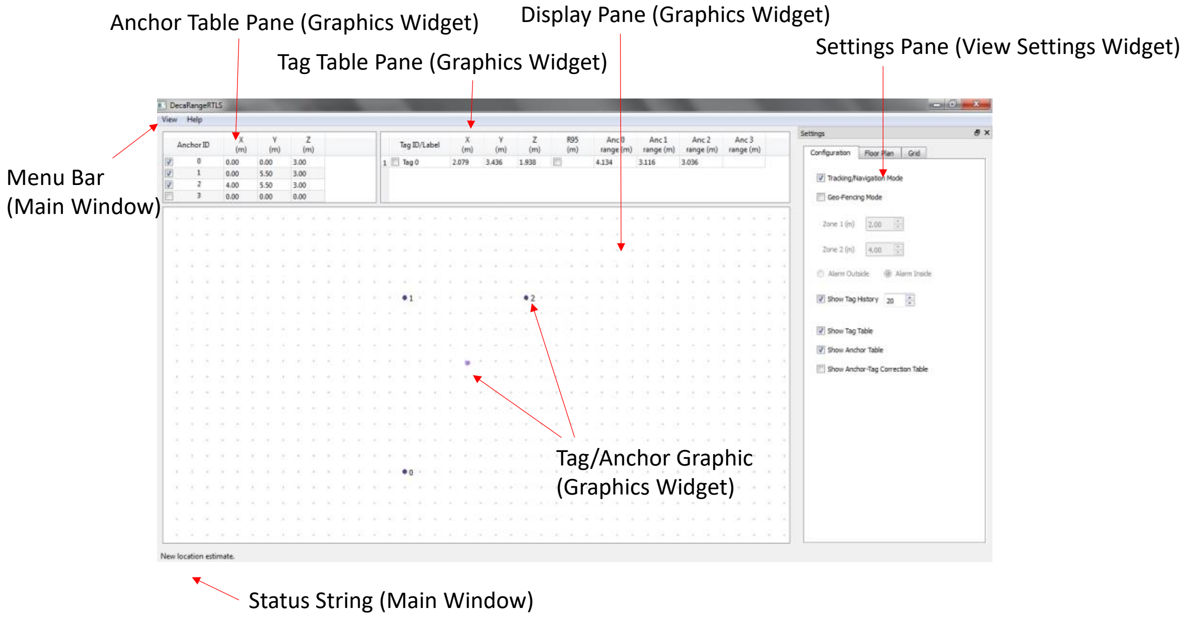
Table 1: List of source files in the DecaRangeRTLS ARM application

Filename	Brief description
main.cpp	This is the application main entry point
RTLSDisplayApplication.cpp	The DecaRangeRTLS application contructor – source file
RTLSDisplayApplication.cpp	The DecaRangeRTLS application contructor – header file
ViewSettings.cpp	View configuration class – source code
ViewSettings.h	View configuration class – header
RTLSClient.cpp	RTLSClient (consumes the TOF reports) class – source code
RTLSClient.h	RTLSClient (consumes the TOF reports) class – header
SerialConnection.cpp	Serial (COM port) connection management class – source code
SerialConnection.h	Serial (COM port) connection management class – header
AbstractTool.h	Abstract tool – header
OriginTool.cpp	Origin manipulation tool – source file
OriginTool.h	Origin manipulation tool – header file
RubberBandTool.cpp	Rubberband tool – source file
RubberBandTool.h	Rubberband tool – header file

Filename	Brief description
ScaleTool.cpp	Scale manipulation tool – source file
ScaleTool.h	Scale manipulation tool – header file
trilateration.cpp	Trilateration functions – source code
trilateration.h	Trilateration functions – header file
QPropertyModel.cpp	QProperty Model – source code
QPropertyModel.h	QProperty Model – header file
connectionwidget.cpp	Connection widget - source
connectionwidget.h	Connection widget - header
connectionwidget.ui	Connection widget - UI
GraphicsView.cpp	GraphicsView widget - source
GraphicsView.h	GraphicsView widget - header
GraphicsWidget.cpp	GraphicsWidget widget - source
GraphicsWidget.h	GraphicsWidget widget - header
GraphicsWidget.ui	GraphicsWidget widget - UI
mainwindow.cpp	mainwindow widget - source
mainwindow.h	mainwindow widget - header
mainwindow.ui	mainwindow widget - UI
MinimapView.cpp	MinimapView widget - source
MinimapView.h	MinimapView widget - header
ViewSettingsWidget.cpp	ViewSettingsWidget widget - source
ViewSettingsWidget.h	ViewSettingsWidget widget - header
ViewSettingsWidget.ui	ViewSettingsWidget widget - UI

2 DECARANGERTLS PC OVERVIEW

This chapter gives an overview about each of the DecaRangeRTLS PC application functional blocks and UI components as shown in Figure 2 and Figure 3:


Anchor Table Pane (Graphics Widget)
Display Pane (Graphics Widget)
Tag Table Pane (Graphics Widget)
Settings Pane (View Settings Widget)
Menu Bar
(Main Window)
DecaRangeRTLS
View
Help
Settings
Configuration
Floor Plan
Grid
1
2
0
Tag/Anchor Graphic
(Graphics Widget)
Status String (Main Window)

	Anchor ID	X (m)	Y (m)	Z (m)
☑	0	0.00	0.00	3.00
☑	1	0.00	5.50	3.00
☑	2	4.00	5.50	3.00
☐	3	0.00	0.00	0.00

	Tag ID/Label	X (m)	Y (m)	Z (m)	R95 (m)	Anc 0 range (m)	Anc 1 range (m)	Anc 2 range (m)	Anc 3 range (m)
1	☐ Tag 0	2.079	3.436	1.938	☐	4.134	3.116	3.036	

☑ Tracking/Navigation Mode
☐ Geo-Fencing Mode
Zone 1 (m) 2.00
Zone 2 (m) 4.00
○ Alarm Outside ● Alarm Inside
☑ Show Tag History 20
☑ Show Tag Table
☑ Show Anchor Table
☐ Show Anchor-Tag Correction Table

New location estimate.

Figure 3: DecaRangeRTLS application – UI components

2.1 RTLSDisplay Application

The *RTLSDisplay Application* class initialises the application and its parts: the _serialConnection is used for managing the COM port connection, the _client consumes the data received over the COM port connection and sends the processed data to the graphical display (graphicsWidget) part of the _mainWindow which holds the various GUI parts (connection widget (_cWidget), statusBar, viewSettings_w (configuration) and dockable widgets: viewSettings_dw and minimap_dw) and menu items (viewMenu, helpMenu), the _viewSettings is used to hold many properties about the view, such as the grid and viewplan settings for configuration of the graphical display.

The application will automatically try to connect to a Tag/Anchor. If the COM port connection fails the user will be prompted to close the application or try again. If the COM port connection is successful, the application will then open the *Main display window* and finish configuration / initialisation of all of its parts. As part of initialisation the configuration files, if present, will also be loaded.

2.2 Serial/COM Port Connection

The *Serial Connection* class is used to establish and monitor the COM port connection to the Tag/Anchor. When the application is started it will scan all of the COM ports available in the system (PC) and check which ones have “STMicroelectronics Virtual COM Port” description (findSerialDevices()). It will then try and connect to the 1st COM port in the list which matches the “STMicroelectronics Virtual COM Port” description (openConnection()).

For the connection to be successful the TREK1000 Tag/Anchor needs to be already plugged into the

PC. (The PC has to have “STMicroelectronics Virtual COM Port” driver installed, so that there is COM port associated with the connected Tag/Anchor.) After opening of the COM port connection the application will send “deca$” string to the Tag/Anchor to find out what is the Tag/Anchor SW version and configuration. If the returned string matches “nVer. v.st TREK” then the application will assume it has connected successfully.

The *Connection state* can be: Disconnected, Connecting, Connected or Connection Failed.

2.3 RTLS client

If the COM port connection as part of initialisation of *Serial Connection* class was successful the COM port data handler will be assigned to the *RTLS client.*

The *RTLS client* will open a log file (./Logs/yyyyMMdd_hhmmssRTLS_log.txt) and initialise its data structures. Then it will handle any the incoming TOF report messages from the Tag/Anchor via the Serial/COM port connection. The reports are firstly grouped into sets of 4 (max 4 TOF reports – one form each anchor (A0, A1, A2 and A3) for each range sequence (RSN)). Once all the range reports with RSN are received, the trilateration function will be called to try and calculate location of the Tag. If there is a solution, Tag’s location estimate will be sent to the the graphical display (graphicsWidget) to update Tag’s position information on the application display window.

There is no option to close or open new log files. If new log file is required the application will need to be restarted.

2.3.1 Trilateration Function

The trilateration function (GetLocation()) which is used to estimate Tag’s location is based on spherical intersection. It can be called with 3 or 4 range measurements. The user should review the function in trilateration.cpp to familiarise themselves with its operation.

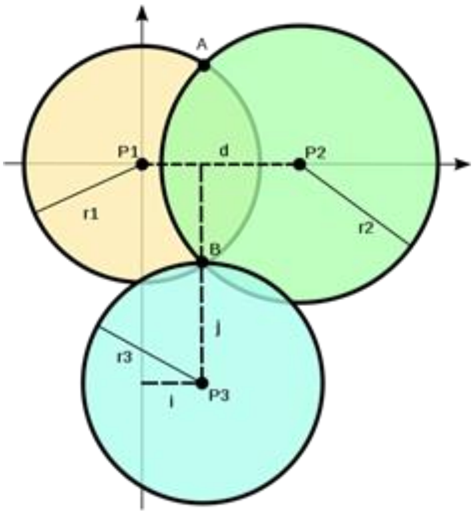


Figure 4: Trilateration function - intersection of 3 spheres

There should be only one solution ideally when three spheres are intersecting. From the figure above, assuming that all of spheres are on the same plane, point B is the only solution which has exactly r1 distance to P1, r2 distance to P2, and r3 distance to P3. However because of range measurement error the intersection B will be given by 2 points equidistant from the plane of the 3 anchors (Ps). The algorithm used in DecaRangeRTLS application will automatically pick the solution which is below the plane. However if there are 4 TOF reports, i.e. if there are 4 anchors present in the system, the solution that is picked is the one closer to anchor 4 (anchor with ID 3).

2.3.2 Range Report Format

There are three ranging report messages sent over the USB port:

```
1. mr 0f 000005a4 000004c8 00000436 000003f9 0958 c0 40424042 a0:0
2. ma 07 00000000 0000085c 00000659 000006b7 095b 26 00024bed a0:0
3. mc 0f 00000663 000005a3 00000512 000004cb 095f c1 00024c24 a0:0
```

The “mr” message consists of tag to anchor raw ranges, “mc” tag to anchor range bias corrected ranges – used for tag location and “ma” anchor to anchor range bias corrected ranges – used for anchor auto-positioning.

MID MASK RANGE0 RANGE1 RANGE2 RANGE3 NRANGES RSEQ DEBUG aT:A

MID	this is the message ID, as described above: mr, mc and ma
MASK	this states which RANGEs are valid, if MASK=7 then only RANGE0, RANGE1 and RANGE2 are valid (in hex, 8-bit number)
RANGE0	this is tag to anchor ID 0 range if MID = mc/mr (in mm, 32-bit hex number)
RANGE1	this is tag to anchor ID 1 range if MID = mc/mr or anchor 0 to anchor 1 range if MID = ma (in mm, 32-bit hex number)
RANGE2	this is tag to anchor ID 2 range if MID = mc/mr or anchor 0 to anchor 2 range if MID = ma (in mm, 32-bit hex number)
RANGE3	this is tag to anchor ID 3 range if MID = mc/mr or anchor 1 to anchor 2 range if MID = ma (in mm, 32-bit hex number)
NRANGES	this is a number of ranges completed by reporting unit raw range (16-bit hex number)
RSEQ	this is the range sequence number (8-bit hex number)
DEBUG	this is the TX/RX antenna delays (if MID = ma) – two 16-bit numbers or time of last range reported – if MID = mc/mr (32 bit hex number)
aT:A	the T is the tag ID and A id the anchor ID

2.4 Main display window

The *Main display window* is a class that contains the drop down menus, main display area, tag information table and other information and control widgets as described in the sections below:

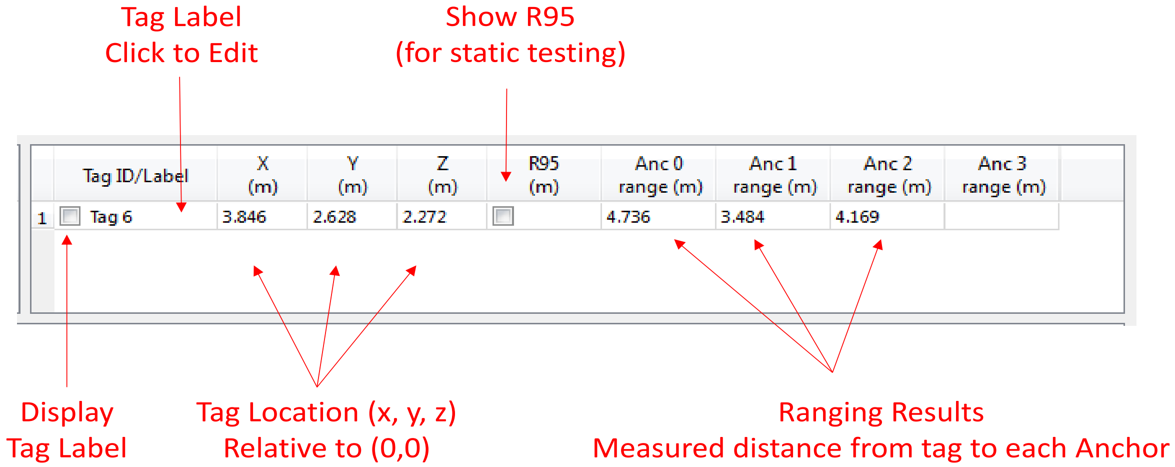
2.4.1 Graphics

The graphical display (graphicsWidget) part of the _mainWindow which holds the various GUI parts namely these are Tag and Anchor table pane, the display pane. The display pane consists of *Graphics View* which has visible rectangle to keep track of the visible rectangle 2.5.1 and tools 2.6 which operate on it and the *Grid Layout*.

As part of initialisation the graphical display will configure tag settings based on the setting in TREKtag_config.xml configuration file.

2.4.1.1 Tag and Anchor Tables

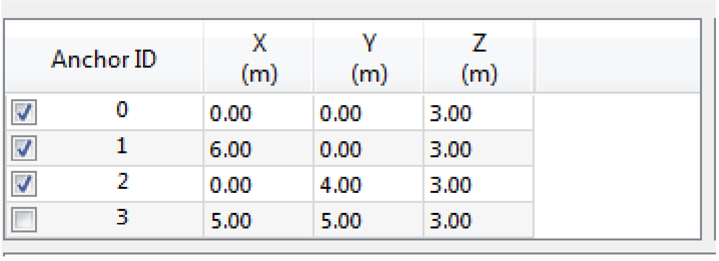
Tag and Anchor tables are Qt Table widgets, the Tag table contains Tag’s ID, position information and range measurements, as shown in Figure 5.


Tag Label
Click to Edit
Show R95
(for static testing)
Display
Tag Label
Tag Location (x, y, z)
Relative to (0,0)
Ranging Results
Measured distance from tag to each Anchor

	Tag ID/Label	X (m)	Y (m)	Z (m)	R95 (m)	Anc 0 range (m)	Anc 1 range (m)	Anc 2 range (m)	Anc 3 range (m)
1	☐ Tag 6	3.846	2.628	2.272	☐	4.736	3.484	4.169	

Figure 5: Tag table

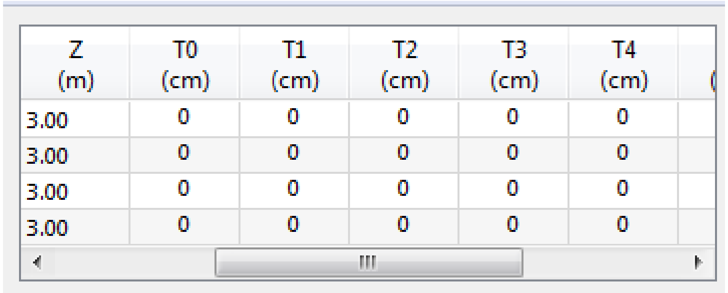
The Anchor table contains Anchor's ID, position information and optional Tag correction table as shown in Figure 6 and Figure 7.



	Anchor ID	X (m)	Y (m)	Z (m)
☑	0	0.00	0.00	3.00
☑	1	6.00	0.00	3.00
☑	2	0.00	4.00	3.00
☐	3	5.00	5.00	3.00

Figure 6: Anchor table

The Tag correction table is used to apply correction (in cm) to Tag-Anchor range measurement. These correction values can be used when TREK1000 SW is used on non-TREK calibrated EVB1000s (e.g. on EVB1000 from and EVK1000).



Z (m)	T0 (cm)	T1 (cm)	T2 (cm)	T3 (cm)	T4 (cm)
3.00	0	0	0	0	0
3.00	0	0	0	0	0
3.00	0	0	0	0	0
3.00	0	0	0	0	0

Figure 7: Anchor's tag correction table

2.4.1.2 Display pane

The display pane has a _scene (QGraphicsScene) element, it provides a surface for managing a large number of 2D graphical items. It is the _scene part of *Graphics View* 2.5.1. When the graphical display widget receives data from the *RTLS client* it will:

- Add/update tag's new visual position
- It will remove (gray out) tag's old positions
- Update labels
- Update R95 circle if shown

The graphical display will also change the view from Tracking/Navigational use case to Geo-Fencing use case depending on user TREK use case configuration.

2.4.2 Connection Widget

The *Connection widget* contains the *Serial Connection* (COM port) configuration options. This widget is hidden from the user as the RTLS application connects automatically.

2.4.3 Status Bar

The *Status bar* is used to display status messages/strings:

- "DecaRangeRTLS Anchor/Tag ID Mode" – on COM port open it will display which Tag/Anchor the PC application is connected to.
- "Connected to Anchor/Tag/Listener ID" – on reception of TOF it will display which Tag/Anchor the PC application is connected to.
- "No location solution" – if the trilateration function cannot calculate Tag's position from 5 or more consecutive TOF report sets.
- "Open error" – if the COM port did not open successfully.

2.4.4 View Settings

The *View settings* widget contains three Tabs: configuration, floorplan and grid.

2.4.4.1 Configuration tab

The Configuration tab contains the general RTLS application settings for use case modes and showing and hiding of Tag and Anchor tables and Tag history function. Here we can switch between 'Tracking/Navigation' mode and 'Geo-Fencing' mode using the checkboxes.

In Geo-Fencing mode:

- Set 2 zone perimeters on the Display Pane – enter the desired perimeters in metres
- Select 'Alarm Outside' or 'Alarm Inside' depending on whether the no-go area is far from or near to the Anchor

4 User Checkboxes:

- Show / Hide the Tag history
- Show / Hide the Anchor Table
- Show / Hide the Tag Table
- Show / Hide the Anchor-Tag Correction Table

2.4.5 Minimap View

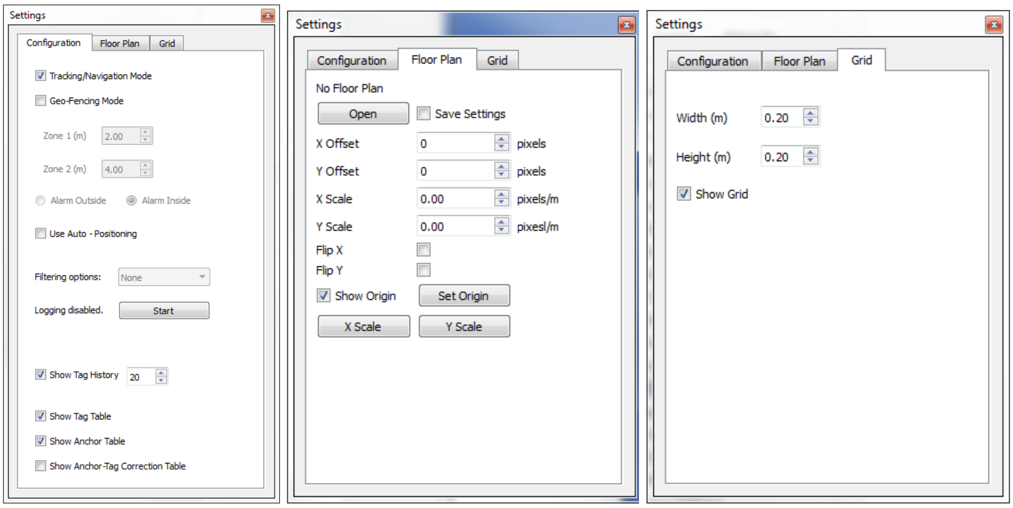
The *Minimap view* widget allows the user, by using the mouse, to select different regions of the floorplan to be displayed in the main scene. It is only operational if a floorplan has been loaded.

2.4.6 Drop Down Menus

2.4.6.1 VIEW

View menu contains the following items: -

- ***View Settings*** – selecting this will open a view configuration widget. This allows the user to upload a floor plan and specify the grid, X and Y axis scale and origin positions: -



Settings

Configuration | Floor Plan | Grid

☑ Tracking/Navigation Mode
☐ Geo-Fencing Mode
Zone 1 (m) 2.00
Zone 2 (m) 4.00
○ Alarm Outside ● Alarm Inside
☐ Use Auto - Positioning
Filtering options: None
Logging disabled. Start
☑ Show Tag History 20
☑ Show Tag Table
☑ Show Anchor Table
☐ Show Anchor-Tag Correction Table

Settings

Configuration | Floor Plan | Grid

No Floor Plan
Open ☐ Save Settings
X Offset 0 pixels
Y Offset 0 pixels
X Scale 0.00 pixels/m
Y Scale 0.00 pixesl/m
Flip X ☐
Flip Y ☐
☑ Show Origin Set Origin
X Scale Y Scale

Settings

Configuration | Floor Plan | Grid

Width (m) 0.20
Height (m) 0.20
☑ Show Grid

Figure 8: View Settings widget

  - *Configuration* tab:

Field Name	**Description**
Tracking/Navigation Mode	Selects Tracking/Navigation as the RTLS application use case
Geo-Fencing Mode	Selects Geo-Fencing as the RTLS application use case
Zone1	Configures Zone 1 radius (only in Geo-Fencing use case)
Zone2	Configures Zone 2 radius (only in Geo-Fencing use case)
Alarm Outside/Inside	Configures if the Alarm is triggered when Tag enters into the minimum zone or leaves the maximum zone. (only in Geo-Fencing use case)
Show Tag History (N)	User can configure if to show the last N Tag positions
Show Tag Table	Shows/hides Tag table
Show Anchor Table	Shows/hides Anchor table
Show Anchor-Tag Correction Table	Shows/hides Anchor-Tag correction table
Auto Positioning	If enabled, the anchors will be placed on screen according to the anchor-to-anchor ranging reports and their coordinates updated
Filtering	If turned on, then filter will be applied to the tag's location results

Field Name	Description
Logging	Logging can be enabled or disabled

- *Grid* tab:

Field Name	Description
Width	Sets the horizontal grid spacing, unit is meters
Height	Sets the vertical grid spacing, unit is meters
show	Shows or hides the grid

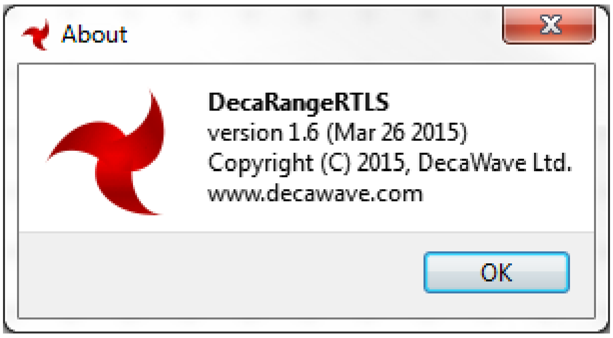
- *Floor Plan* tab:

Field Name	Description
Open/Clear	This lets the user upload an image of the floor plan of the area where the anchors are installed. The button will display “Clear” when the floorplan has been loaded and can be then used to remove/clear the floorplan and associated settings.
Save Settings	When checked then the floor plan settings will be saved in the view settings file.
X offset	This is the origin offset in the X direction from the 0, 0 point on the floorplan (in pixels).
Y offset	This is the origin offset in the Y direction from the 0, 0 point on the floorplan (in pixels)
X scale	This is the scale (pixels per meter) of the x axis (in pixels per meter).
Y scale	This is the scale (pixels per meter) of the y axis (in pixels per meter)
Flip X	This flips the image in the x-axis.
Flip Y	This flips the image in the y-axis.
show	This shows or hides the origin in the map.
Set Origin	This button lets the user click and set the origin coordinates.
X Scale button	Pressing this button produces a measuring tool with which the user can firstly select a distance on the map and then enter the actual distance (in meters) that range corresponds to, this sets the *X scale* value.
Y Scale button	Pressing this button produces a measuring tool with which the user can firstly select a distance on the map and then enter the actual distance (in meters) that range corresponds to, this sets the *Y scale* value.

- ***Minimap*** – selecting this opens a Minimap widget, which shows the loaded image and the zoomed in area (which is shown in the Main Display Area window).

2.4.6.2 HELP

- ***About*** – this opens the “About” window which provides information on the revision of the client.



DecaRangeRTLS
version 1.6 (Mar 26 2015)
Copyright (C) 2015, DecaWave Ltd.
www.decawave.com

Figure 9: About window

2.5 *Widgets and Views*

The DecaRangeRTLS application has a number of Widgets and View classes they are used to display the data, and allow user to change/configure the various parameters.

2.5.1 Graphics View

The *Graphics View* class draws the scene and provides user interaction using the mouse. User interaction can be complex as we have to handle many different interaction based on very little mouse events:

- Scene interaction (Selecting anchors, moving them,) - (this is handled by QGraphicsScene and GraphicsDataItem)
- Unselecting all anchors by clicking on the background.
- Zooming using the scroll wheel
- Panning by dragging
- Contextual menu by right-click
- Cancelling the current tool by right-click
- Starting a *Rubber Band Tool* on Ctrl+Drag (2.6.3)
- Tools listening to AbstractTool::clicked() events (2.6.1)
- Tools listening to AbstractTool::mousePressEvent() events (2.6.1)

Most of them are handled by the mousePressEvent()/mouseMoveEvent()/mouseReleaseEvent() cycle. During mousePressEvent(), we find the suited action, and decide of a MouseContext based on that. The MouseContext is kept until mouseReleaseEvent()

The GraphicsView keeps track of the visible rectangle, in scene coordinates. This rectangle will always be visible on the screen. Initially, the visible rectangle is the square from (0, 0) to (1, 1).
The visible rectangle can be transformed using translateView() or scaleView() or changed using setVisibleRect(). Whenever the visible rectangle changes, for any reason, the visibleRectChanged() signal is called.
Tools allow simple interaction inside the scene. A new tool can be set using setTool(). The tool then remains active until it's AbstractTool::done() signal is emitted. When ESC button or right click is pressed, the view attempts to cancel the tool by calling AbstractTool::cancel().

2.5.2 Graphics

The *Graphics* widget is used to display tags and anchors on the main display. It adds tag and anchor items to the scene, and updates the x, y, z co-ordinates as it get new data from the CLE.

The list of tags is also shown, here the user can add tag label, and view multilateration, and blink reception rates as well as turning on/off the R95 indicator.

2.5.3 Minimap

The *Minimap* view shows a zoomed out view of the whole floorplan. It allows the user to quickly select and zoom into a particular place on the floor plan loaded.

2.5.4 View Settings

The *View Settings* widget allows the user to configure the grid size, floorplan settings etc. This is explained in detail in section 2.4.6.1 View Settings.

2.6 Tools and Utilities

2.6.1 Abstract Tool

The Abstract Tool class is a base class any tool implementations should inherit. Tools allow simple, temporary user interaction in the graphics view, based on mouse events. They are used to set the scale (*Scale Tool 2.6.4*) and origin (*Origin Tool 2.6.2*) of the bitmap, and select anchors using a click and drag style rubber band (*Rubber Band Tool 2.6.3*).

Once a tool is enabled, using GraphicsView::setTool(), the cursor changes to the return value of cursor(), the draw() function gets called when drawing the foreground of the scene, and the tool starts receiving mouse events.

The tool can receive the mouse events through two means, pressing and releasing of the left mouse button. The clicked() function gets called when the left button is pressed and released. For more complex interaction, the mousePressEvent()/mouseMoveEvent()/mouseReleaseEvent() functions can be overridden. When a mouse button is pressed, mousePressEvent is called. If it returns true, then tool grabs the mouse interaction, and will receive mouse events until the button is released. Otherwise the scene will be handling mouse events, and mouseMoveEvent()/mouseReleaseEvent() won't be called.

Note that the two mouse event mechanisms are incompatible. If mousePressEvent returns true, clicked() will not be called for this mouse click.

2.6.2 Origin Tool

The *Origin Tool* class is a tool used for setting the floorplan's origin point. It reacts to the clicked() event. When clicked() is called, the origin is calculated based on the click position, and the tool finishes right away.

2.6.3 Rubber Band Tool

The *Rubber Band Tool* class is a tool used to select anchors using a click and drag to draw a selection box. Once the user has started the drag, a rectangular selection box is drawn, and all items within the box get selected. In order to differentiate this tool's click and drag with the one used to move the scene, the tool is started if the Control button is held during the initial mouse press event.

This is handled by the GraphicsView::mousePressEvent(), and is outside the scope of this class.

2.6.4 Scale Tool

The *Scale Tool* class is a tool used to change the floorplan's scale. It allows the user to select two points by waiting for two consecutive clicked() events. Once the user has clicked on two points, the tool shows a popup to enter the distance between the two points. The scale is recalculated and stored.

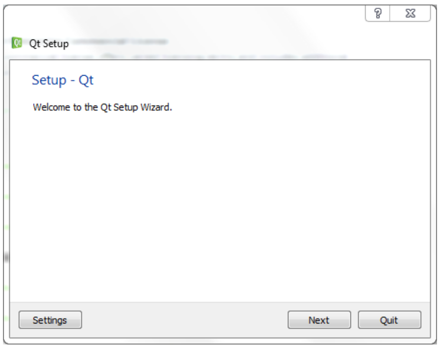
2.6.5 QProperty Model

The *QProperty Model* is a class for turning any QObject-derived subclass with properties into a one-row model. (Copyright 2013 - Harvey Chapman hchapman@3gfp.com Source: https://gist.github.com/sr105/7955969). QPropertyModel creates a single row data model consisting of columns mapping to properties in a QObject. The column list can be retrieved as a QStringList, and a method exists to convert the property names to column numbers.

3 QT SDK INSTALLATION GUIDE

The DecaRangeRTLS application is built under the Qt Framework, the released binary is compiled and built with Qt Creator (3.1.2 opensource). The Qt installer for windows can be found online at http://www.qt.io/download-open-source/ . Select *Qt Online Installer* for your version of Windows platform:

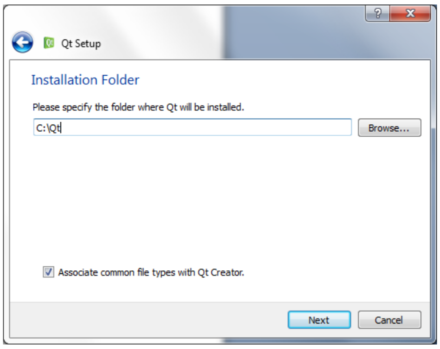
Once the installer is downloaded, run it and press *Next*:



Setup - Qt

Welcome to the Qt Setup Wizard.

Specify the installation path, and press *Next*:



Installation Folder

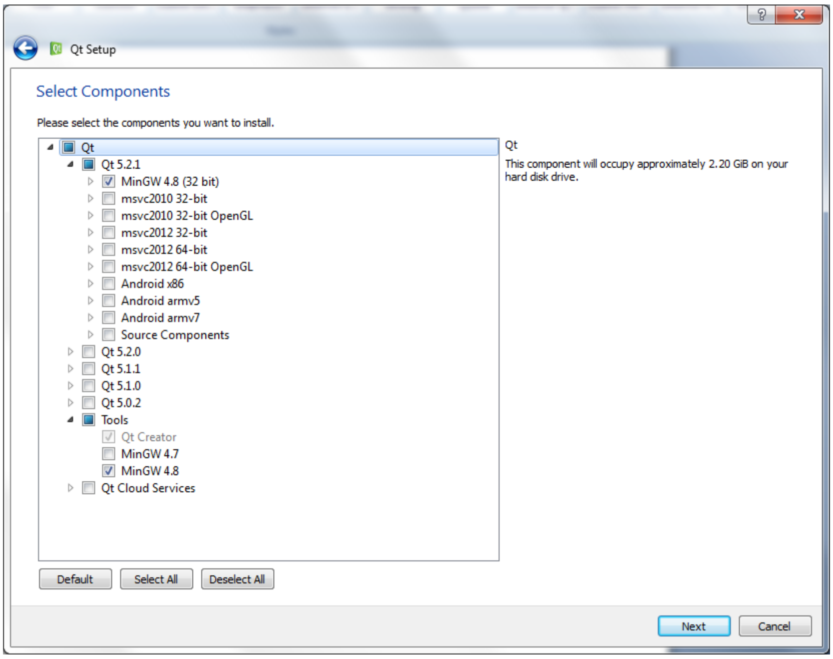
Please specify the folder where Qt will be installed.

C:\Qt

☑ Associate common file types with Qt Creator.

Select the following components, and press *Next*:

- *Qt -> Qt 5.2.1 -> MinGW 4.8*
- *Qt -> Tools -> Qt Creator*
- *Qt -> Tools -> MinGW 4.8*


Qt Setup
Select Components
Default
Select All
Deselect All
Next
Cancel

Please select the components you want to install.

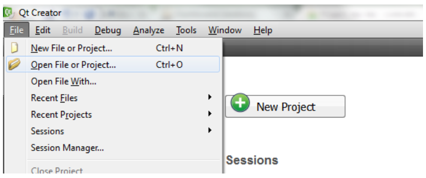
- ■ Qt
  - ■ Qt 5.2.1
    - ☑ MinGW 4.8 (32 bit)
    - ☐ msvc2010 32-bit
    - ☐ msvc2010 32-bit OpenGL
    - ☐ msvc2012 32-bit
    - ☐ msvc2012 64-bit
    - ☐ msvc2012 64-bit OpenGL
    - ☐ Android x86
    - ☐ Android armv5
    - ☐ Android armv7
    - ☐ Source Components
  - ☐ Qt 5.2.0
  - ☐ Qt 5.1.1
  - ☐ Qt 5.1.0
  - ☐ Qt 5.0.2
  - ■ Tools
    - ☑ Qt Creator
    - ☐ MinGW 4.7
    - ☑ MinGW 4.8
  - ☐ Qt Cloud Services

Qt

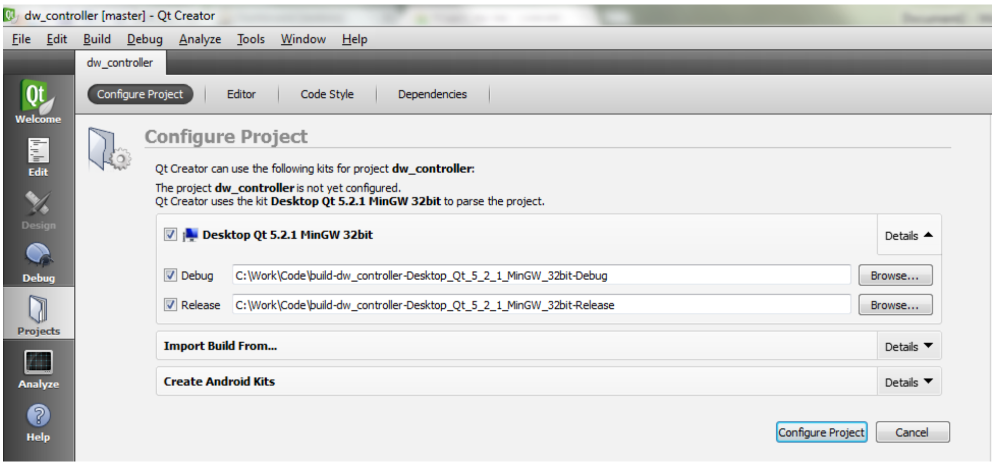
This component will occupy approximately 2.20 GiB on your hard disk drive.

Choose the default options until the end of the wizard.

Run the Qt Creator software. Under the *File* menu, select *Open File or Project*


Qt Creator
File
Edit
Build
Debug
Analyze
Tools
Window
Help
New File or Project... Ctrl+N
Open File or Project... Ctrl+O
Open File With...
Recent Files
Recent Projects
Sessions
Session Manager...
Close Project
New Project
Sessions

Select the *.pro* file inside the Project's directory. The first time the project is opened, it must be configured. Leave the default options, and click *Configure Project*


dw_controller [master] - Qt Creator
File
Edit
Build
Debug
Analyze
Tools
Window
Help
dw_controller
Welcome
Edit
Design
Debug
Projects
Analyze
Help
Configure Project
Editor
Code Style
Dependencies
Configure Project
Details
Browse...
Browse...
Import Build From...
Details
Create Android Kits
Details
Configure Project
Cancel

Qt Creator can use the following kits for project **dw_controller**:

The project **dw_controller** is not yet configured.
Qt Creator uses the kit **Desktop Qt 5.2.1 MinGW 32bit** to parse the project.

☑ **Desktop Qt 5.2.1 MinGW 32bit**

☑ Debug C:\Work\Code\build-dw_controller-Desktop_Qt_5_2_1_MinGW_32bit-Debug

☑ Release C:\Work\Code\build-dw_controller-Desktop_Qt_5_2_1_MinGW_32bit-Release

4 BIBLIOGRAPHY

Table 2: Table of References

Ref	Author	Date	Version	Title
[1]	Decawave		Current	APS003 Real Time Location Systems
[2]	Decawave		Current	DW1000 User Manual
[3]	Decawave		Current	TREK1000 User Manual
[5]	IEEE		2011	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/

5 REVISION HISTORY

Table 3: Document History

Revision	Date	Description
2.0	31st December 2015	Internal release for review.
2.1	20-Augustus-2018	Logo update

6 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.