



DecaWave: UWB-IOT tools Infrastructure RIOT-DW1000 User Guide

The authorized version of this document is an electronic master stored in the Document Repository (PP Intranet Portal / PAL). It is advised that the version of the document in the repository be matched with the hard copy before using it. The information contained in this document is proprietary to PathPartner Technology Consulting Pvt Ltd.

Copyright © PathPartner Technology Consulting Pvt Ltd. This material, including documentation and any related computer programs, is protected by copyright controlled by PathPartner Technology. All rights are reserved. Copying, including reproducing, storing, adapting or translating, any or all of this material requires the prior written consent of the authorized personnel of PathPartner Technology.

This material contains confidential information, which shall not be disclosed to others without the prior written consent of the authorized personnel of PathPartner Technology.

Table of Contents

1.	About this Manual.....	3
2.	Getting Started	3
2.1.	System Requirements.....	3
2.1.1.	Installing Jlink.....	3
2.2.	Running DecaRange RTLS ARM Application on Nordic NRF52840 and STM32F401RE Nucleo platforms and Deca Multi-Threaded Application on Nordic NRF52840 platform.....	3
2.2.1.	Common Steps for all examples	3
2.2.2.	Connecting Hardware Platform to EVB1000 with SPI Interface	5
2.2.2.1	NRF52840 Nordic Platform	5
2.2.2.2	STM32F401RE Nucleo Platform	6
2.2.3.	Running DecaRange RTLS ARM Application.....	7
2.2.4.	Running Deca Multi-Threaded Application on Nordic platform	7
2.3.	BSP Overview	7
2.3.1.	core	8
2.3.2.	boards.....	8
2.3.3.	cpu	8
2.3.4.	cpu/periph.....	8
2.3.5.	drivers.....	8
2.3.6.	sys	9
2.3.7.	sys/net.....	9
2.3.8.	pkg	9
2.3.9.	examples.....	9
2.3.10.	tests.....	9
2.3.11.	dist and doc.....	9
2.4.	RIOT-DW1000 Code / Directory Organization	9
2.4.1.	Top Level	9
2.4.2.	Decawave Board Module for Nordic NRF52840.....	10
2.4.3.	Decawave Board Module for STM32F401RE Nucleo	10
2.4.4.	Docs	10
2.4.5.	Examples.....	11

1. About this Manual

This document describes how to build and work with the BSP for Decawave on RIOT for Nordic NRF52840 and STM32F401RE Nucleo platforms. The BSP package for Decawave on RIOT serves to provide a software platform for development, deployment and execution of DecaRange RTLS ARM and Deca Multi-Threaded application.

In this context, the document contains instructions for:

- Setting up the Environment
- Build the DecaRange RTLS ARM application for defined platform
- Flash the DecaRange RTLS ARM application for defined platform
- Build the Deca Multi-Threaded Application for Nordic platform
- Flash the Deca Multi-Threaded Application for Nordic platform

2. Getting Started

2.1. System Requirements

Refer to 'Dependencies' section of release notes

2.1.1. Installing Jlink

- To flash the binary JLink is used, please install the tool manually from <https://www.segger.com/downloads/jlink>.
- Expand J-Link Software and Documentation Pack and select the suitable package and click download.
- Double click on the downloaded package to install.

2.2. Running DecaRange RTLS ARM Application on Nordic NRF52840 and STM32F401RE Nucleo platforms and Deca Multi-Threaded Application on Nordic NRF52840 platform

This section describes how to run the defined application for DW1000 on RIOT.

2.2.1. Common Steps for all examples

- To initialize the Environment Variables, installing Dependent Libraries (First time), setup (clone the RIOT repo and sym link creation for dw1000) and building the application
 - Inside the cloned repo, run the script – **"source ./build_setup.sh INITIAL"**
- To initialize the Environment Variables, setup (clone the RIOT repo and sym link creation for dw1000) and building the application (Next Time on Wards)
 - Inside the cloned repo, run the script – **"source ./build_setup.sh UPDATE"**
- To build the DecaRange RTLS ARM application and Deca Multi-Threaded Application for Nordic nrf52840 platform.
 - Inside the cloned repo, run the script -- **"source ./build_setup.sh"**
- To build the DecaRange RTLS ARM application for STM32F401RE Nucleo platform.
 - Modify the setenv.sh as **"DW_HW_PLATFORM=nucleo-f401"**
 - Inside the cloned repo, run the script -- **"source ./build_setup.sh"**
- To connect EVB1000 board to Nordic NRF52840 or STM32F401RE Nucleo External Microcontroller
 - Turn off S2 DIP switches
 - Turn off S3-1 DIP switch
 - Remove the jumper J10
 - Make sure the pins of Nordic NRF52840 or STM32F401RE Nucleo Platforms and EVB1000 Device are wired as defined below

EVB 1000 to Nordic NRF52840

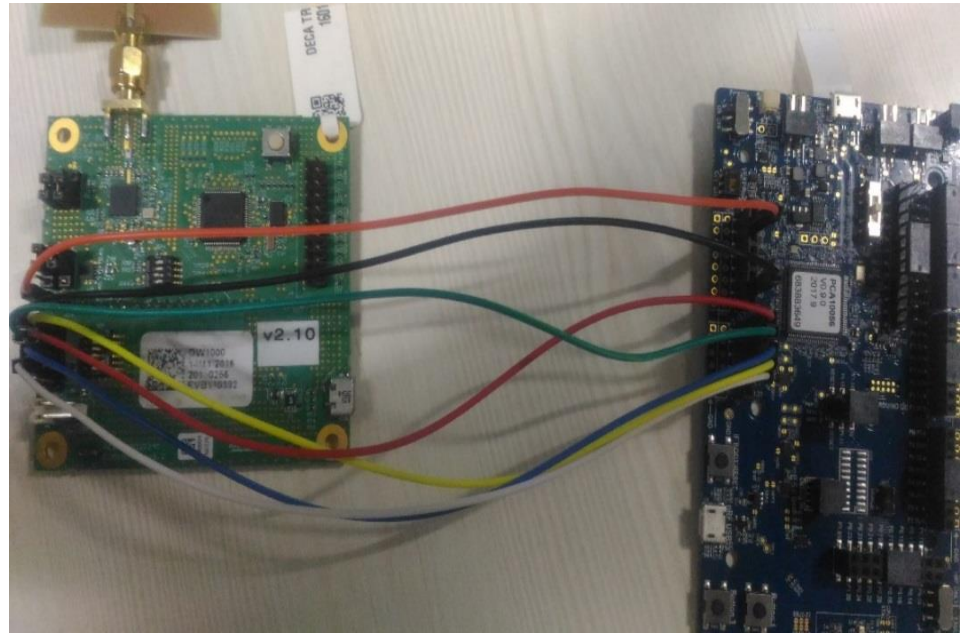
Pin	EVB1000	Nordic NRF52840
IRQ	J6: Pin4	P0.30
MISO	J6: Pin5	P0.28
MOSI	J6: Pin8	P0.04
SCLK	J6: Pin7	P0.03
CS	J6: Pin9	P0.29
GND	J6: Pin10	GND
VDD	J10: Pin2	VDD

EVB 1000 to STM32F401RE Nucleo

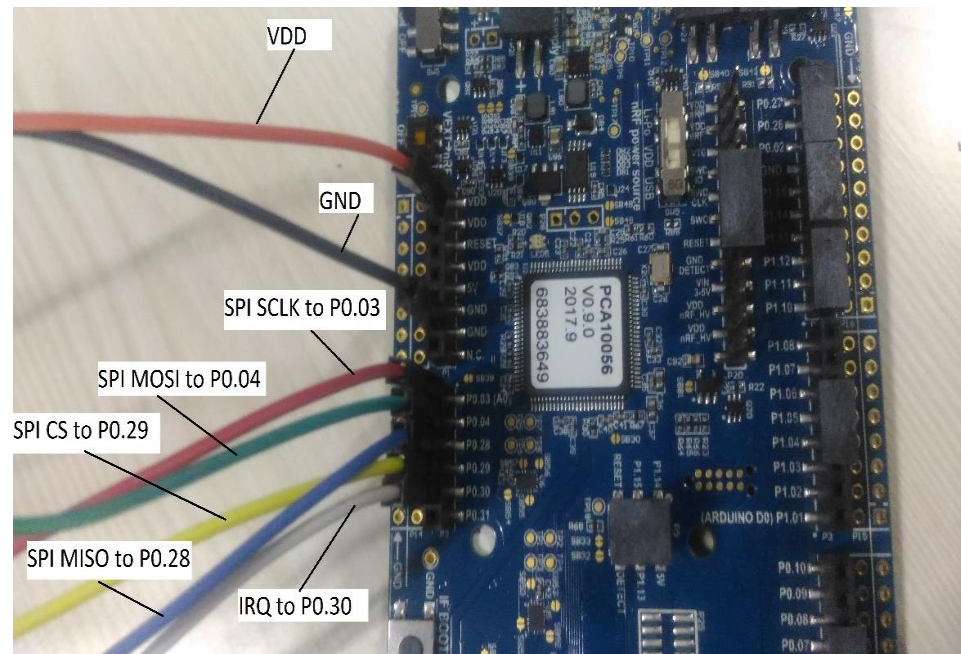
Pin	EVB1000	STM32F401RE Nucleo
IRQ	J6: Pin4	PA06
MISO	J6: Pin5	PB04
MOSI	J6: Pin8	PB05
SCLK	J6: Pin7	PB03
CS	J6: Pin9	PA04
GND	J6: Pin10	GND
VDD	J10: Pin2	3V3

2.2.2. Connecting Hardware Platform to EVB1000 with SPI Interface

2.2.2.1 NRF52840 Nordic Platform

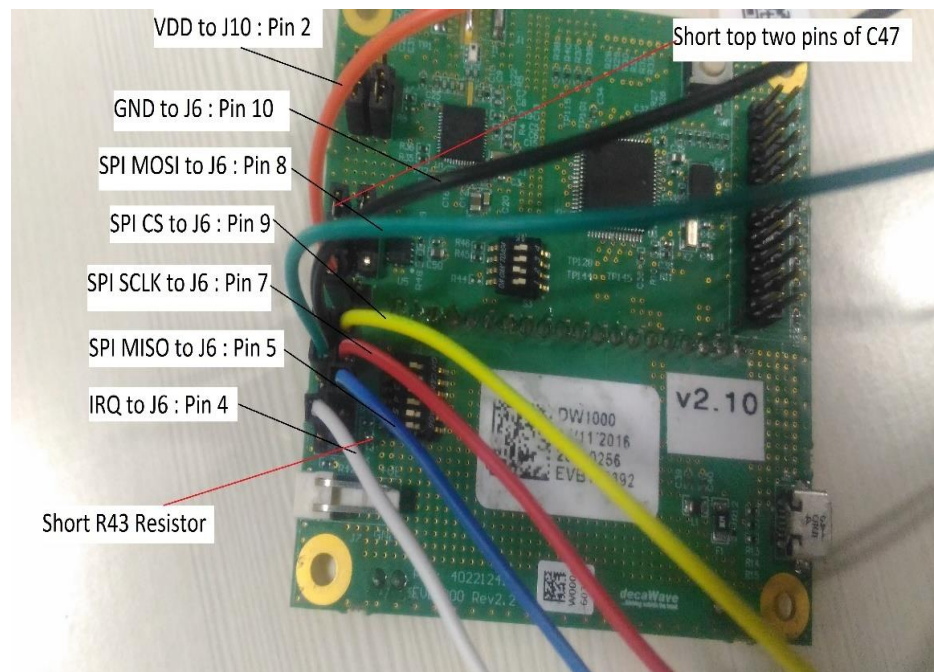


Connectors for SPI Interface on NRF52840 Nordic Platform to EVB1000



Connection Details of SPI Interface on NRF52840 Nordic Platform

RIOT-DW1000 UserGuide

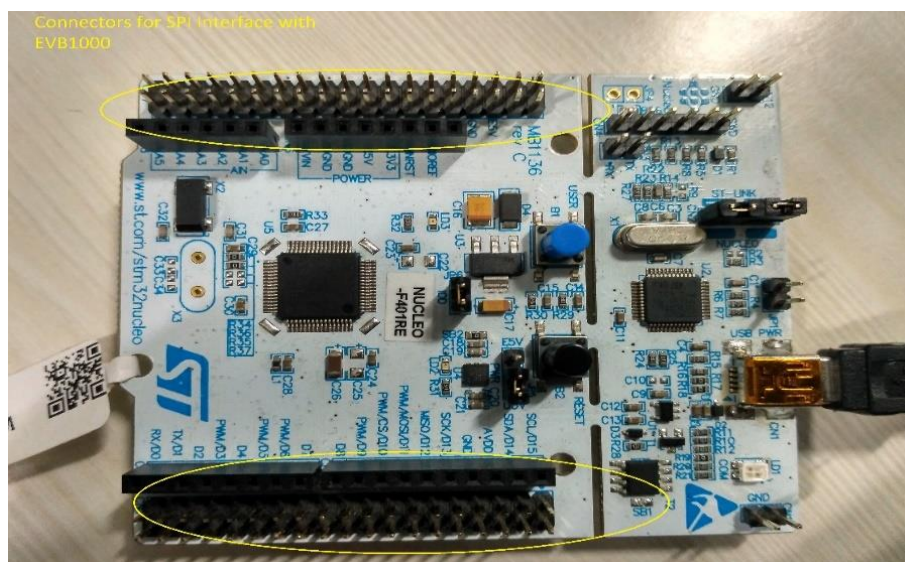


Connection Details of SPI Interface on EVB1000

2.2.2.2 STM32F401RE Nucleo Platform

Interfaces

SPI interface with EVB1000 is highlighted below



Connectors for SPI Interface on STM32F401RE Nucleo Platform to EVB1000

2.2.3. Running DecaRange RTLS ARM Application

- Inside Ubuntu, open a terminal and input –
- To Flash –
`"sudo make -C $RIOT_ROOT/examples/dw1000_rtls_arm BOARD=$DW_HW PLATFORM flash"`
- open a new terminal and input –
`"cd $RIOT_ROOT/RIOT/dist/tools/pyterm"`
`"sudo ./pyterm -p /dev/ttyACM0"`
- Select the EVB1000 device configuration such as Mode(Mode1/Mode2/Mode3/Mode4), UNIT(Tag/Anchor) and UNIT ID (0 to 2)
 For Example:
 - If the user selects Mode as 'Mode1', UNIT as 'Tag' and UNIT ID as '0' then EVB1000 with Nordic NRF52840 or STM32F401RE Nucleo is configured for Tag0 with Mode1 Configuration
- In case of Nordic NRF52840, Configure A0, A1 and A2 for Mode 1 configuration as defined above
- In case of STM32F401RE Nucleo, Select the EVB1000 with STM32F10x (internal controller) as A0, A1 and A2 for Mode 1 configuration
- Now calculated range distance is printed on the terminal

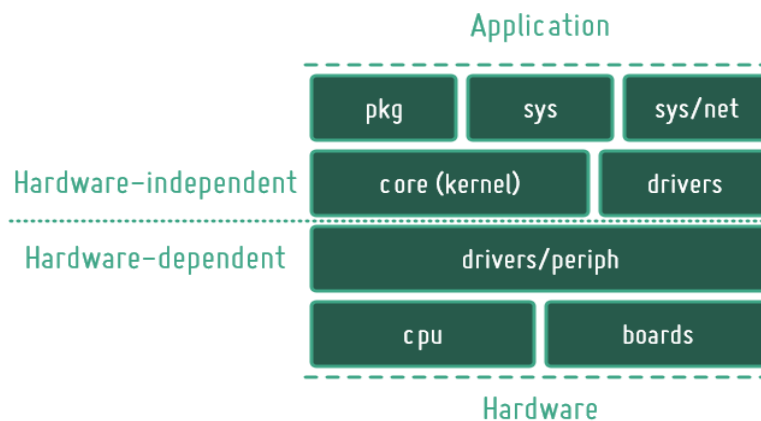
2.2.4. Running Deca Multi-Threaded Application on Nordic platform

- Inside Ubuntu, open a terminal and input –
- To Flash –
`"sudo make -C $RIOT_ROOT/examples/dw1000_multithreaded_app BOARD=$DW_HW PLATFORM flash"`
- open a new terminal and input –
`"cd $RIOT_ROOT/RIOT/dist/tools/pyterm"`
`"sudo ./pyterm -p /dev/ttyACM0"`
- Select the EVB1000 device configuration such as Mode(Mode1/Mode2/Mode3/Mode4), UNIT(Tag/Anchor) and UNIT ID (0 to 2)
 For Example:
 - If the user selects Mode as 'Mode1', UNIT as 'Tag' and UNIT ID as '0' then EVB1000 with Nordic NRF52840 is configured for Tag0 with Mode1 Configuration
- For Nordic RF52840, Configure A0, A1 and A2 for Mode 1 configuration as defined above.
- Now, Deca Multi-Threaded Application spawns three threads and then DecaRange RTLS ARM Application (defined as Thread1 with High Priority) prints calculated range distance on the terminal. The Soft timer Application (defined as Thread2 with Medium Priority) wakeup on interrupt and updated Timer data when Thread1 goes to sleep. The Solver Application (defined as Thread 3 with low priority) prints the Timer data and Range data values collected from both Deca Range RTLS ARM application and soft timer application on to the console when Thead1 and Thread 2 goes to sleep.

2.3. BSP Overview

RIOT Board support package supports generic BSP module for different H/W Platforms which helps to integrate an application easily for different H/W Platforms. Based on RIOT, we have come up with BSP module-specific design for Decawave peripheral.

Software layer structure of the DecaRange RTLS ARM application based on RIOT is shown below.



Software layer for DecaRange RTLS ARM application

RIOT Board support package is structured into five groups.

- The kernel (core)
- Platform specific code (cpu; boards)
- Device drivers (drivers)
- Libraries and network code (sys; pkg)
- Applications for demonstrating features and for testing (examples; tests)

The following list gives a more detailed description of BSP top-level directories

2.3.1. core

It contains the actual kernel. The kernel consists of the scheduler, inter-process-communication (messaging), threading, thread synchronization, and supporting data-structures and type definitions

2.3.2. boards

The platform dependent code is split into two logic elements: CPUs and boards, a board has exactly one CPU, while a CPU can be part of multiple boards. This configuration mainly includes the peripheral configuration and pin-mapping, the configuration of on-board devices, and the CPU's clock configuration

2.3.3. cpu

It contains all CPU specific configurations, such as implementations of power management, interrupt handling and vectors, startup code, clock initialization code and thread handling code

2.3.4. cpu/periph

It contains CPU's peripheral drivers like SPI, UART, GPIO, etc.,

2.3.5. drivers

It contains the drivers for external devices such as network interfaces, sensors and actuators. All of device drivers are based on the peripheral driver API (e.g. SPI, GPIO, etc.) and other

modules like the xtimer. This way the drivers are completely platform agnostic and they don't have any dependencies into the CPU and board code

2.3.6. sys

It contains modules that are not part of the hardware abstraction nor device. The libraries include data structures (e.g. bloom, color), crypto libraries (e.g. hashes, AES), high-level APIs (e.g. Posix implementations), memory management (e.g. malloc), the RIOT shell etc.,

2.3.7. sys/net

The sys/net contains networking code

2.3.8. pkg

It comes with support for a number of external libraries (e.g. ccn-lite, microcoap)

2.3.9. examples

It contains example applications that demonstrate certain features

- dw1000_rtls_arm
DecaRange RTLS ARM Application prints calculated range distance between the Tag and Anchor nodes based on TWR Ranging Services on the terminal.
- dw1000_multithreaded_app
Deca Multi-Threaded Application spawns three threads and then DecaRange RTLS ARM Application (defined as Thread1 with High Priority) prints calculated range distance on the terminal. The Soft timer Application (defined as Thread2 with Medium Priority) wakeup on interrupt and updated Timer data when Thread1 goes to sleep. The Solver Application (defined as Thread 3 with low priority) prints the Timer data and Range data values collected from both Deca Range RTLS ARM application and soft timer application on to the console when Thead1 and Thread 2 goes to sleep.

2.3.10. tests

It contains test application of many features/modules in RIOT

2.3.11. dist and doc

It contains all the tooling around RIOT

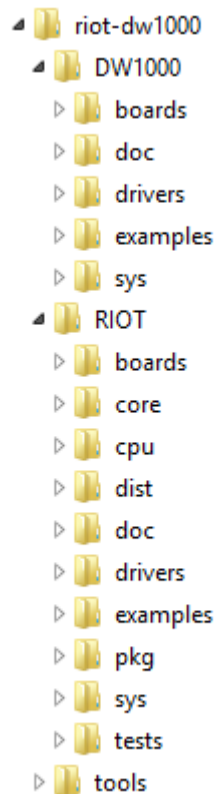
"doc" contains the doxygen configuration and also contains the compiled doxygen output after running "make doc"

"dist" contains tools for help with RIOT. These include the serial terminal application pyterm, generic scripts for flashing, debugging, resetting (e.g. support for OpenOCD, Jlink), as well as code for enabling easy integration to open testbeds such as the IoT-LAB

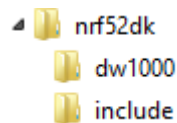
2.4. RIOT-DW1000 Code / Directory Organization

2.4.1. Top Level

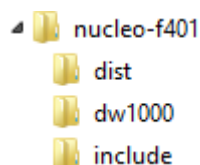
RIOT-DW1000 source code directory will have following folders. Let's consider 01_00_04 version release as an example

RIOT-DW1000 UserGuide**2.4.2. Decawave Board Module for Nordic NRF52840**

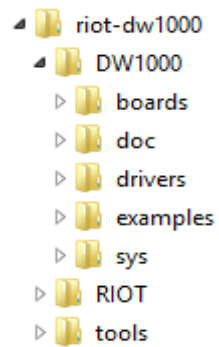
The boards folder contains platform specific folder such as nrf52dk etc., The EVB1000 related changes for Nordic NRF52840 has been defined in "boards/nrf52dk/dw1000" folder. This configures peripheral, interrupt configuration and pin-mapping specific to EVB1000. Any special programming needed by the EVB1000 board is also added here.

**2.4.3. Decawave Board Module for STM32F401RE Nucleo**

The boards folder contains platform specific folder such as nucleo-f401 etc., The EVB1000 related changes for STM32F401-Nucleo has been defined in "boards/ nucleo-f401/dw1000" folder. This configures peripheral, interrupt configuration and pin-mapping specific to EVB1000. Any special programming needed by the EVB1000 board is also added here.

**2.4.4. Docs**

The doc folder contains doxygen API guide, release notes, user guide and test Reports for Nordic NRF52840 and STM32F401RE Nucleo platforms. The relnotes_archive folder contains the user guides and release notes for the previous releases.

RIOT-DW1000 UserGuide**2.4.5. Examples**

The examples folder is the root folder for all the DecaRange RTLS ARM Application for various supported platforms and Deca Multi-Threaded Application for Nordic platform.

