Porting EVK1000 to ST Workbench

Upgrading EVB1000 to latest software IDE

Offset Compensation

Version 1.0

This document is subject to change without notice

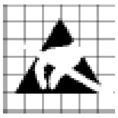
DOCUMENT INFORMATION

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 with Decawave 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.

REGULATORY APPROVALS

This EVK1000 evaluation kit based on Decawave DW1000 IC is intended solely for use by competent engineering personnel for the purposes of evaluating the use of Decawave DW1000 IC in wireless location and communications systems.

The EVK1000, as supplied from Decawave, has not been certified for use in any particular geographic region by any regulatory body governing radio emissions in such regions.

The EVK1000 is supplied under the following conditions: -

- The distribution and sale of the EVK1000 is intended solely for use in future development of devices which may be subject to regulations or other authority governing radio emission.
- This EVK1000 may not be resold by users for any purpose.
- The EVK1000 as supplied by Decawave may not be incorporated directly into user devices or products unless such products undergo the appropriate certification.
- Operation of the EVK1000 in the development of future devices is at the discretion of the user and the user bears all responsibility for any compliance with regulations laid down by the authority governing radio emissions in the user's jurisdiction.

All products developed by the user incorporating the DW1000 must be approved by the relevant authority governing radio emissions in a jurisdiction prior to the marketing or sale of such products in that jurisdiction. User bears all responsibility for obtaining such approval.

If the user has obtained the EVK1000 for any purpose other than those listed above the user should return the EVK1000 to the supplier immediately.

FCC NOTICE: This kit is designed to allow (i) product developers to evaluate electronic components, circuitry, or software associated with the kit to determine whether to incorporate such items in a finished product and (ii) software developers to write software applications for use with the end product. This kit is not a finished product and when assembled may not be resold or otherwise marketed unless all required FCC equipment authorizations are first obtained. Operation is subject to the conditions that this device not cause harmful interference to licensed radio stations and that this device accept harmful interference. Unless the assembled kit is designed to operate under Part 15, Part 18 or Part 95 of the FCC Rules, the operator of the kit must operate under the authority of an FCC license holder or must secure an experimental authorization under Part 5 of the FCC Rules.

DISCLAIMER

This Disclaimer applies to the DW1000 API source code and the “DecaRanging” sample application source code (collectively “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

TABLE OF CONTENTS

LIST OF TABLES

1 INTRODUCTION

This document is **only** intended for EVK1000 customers who wish to change from default Coocox project and use ST's CubeMX tools and ST System Workbench IDE.

It describes the steps to achieve this port.

Note: It is assumed that user has a knowledge of CubeMX tool and ST System Workbench IDE. For support on these tools ST should be contacted. The versions used in this document:

- STM32Cube MX version 4.18.0
- ST System Workbench - Version: Mars.2 Release (4.5.2)

2 CUBEMX CODE GENERATION PROCEDURE

2.1 Installation of tools and drivers

Follow ST's document which describes installation of required tools and drivers [1]. The CubeMX version used for generation of this document is 4.18.0. As a prerequisite STM32F1 firmware package libraries have been installed (version 1.4.0, and all the preceding ones).

2.2 Get EVB1000 schematic

Get EVB1000 schematic from Decawave, this will show you which GPIOs/peripherals are used and how they need to be connected to DW1000 and other components on the EVB1000 board.

2.3 Create CubeMX project

Open STM32CubeMX and start a new project:

a. Firstly select microcontroller (STM32105RCTx is used on EVB1000)
b. Then follow steps below to configure various peripherals and GPIOs, as described in 2.4
c. Then configure the MCU clocks, as described in 2.5
d. Lastly check the overall configuration, as described in 2.6

2.4 Configure microcontroller peripherals and GPIOs

In the Pinout tab, label and configure the needed GPIOs and their peripheral functions, see Figure 1 below.

a. For the RCC peripheral select the HSE source as crystal
b. For the SPI1 peripheral select mode as Full-Duplex master, disable NSS, will use SW controlled CS, thus set GPIO PA4 as GPIO_Output, label as DW_NSS, and the other SPI signals as DW_MOSI, DW_MISO and DW_SCK
c. For the SPI2 peripheral select mode as Full-Duplex master, disable NSS, will use SW controlled CS, thus set GPIO PB12 as GPIO_Output, label as LCD_NSS, and the other SPI signals as LCD_MOSI, LCD_MISO and LCD_SCK
d. Configure PB10 and PB11 as GPIO_Output and label as LCD_RW and LCD_RS.
e. Set GPIOs PC6, PC7, PC8 and PC9 as GPIO_Output these connect to LEDs (LED5, LED6, LED7 and LED8 respectively), label the GPIOs as LED5, LED6, LED7 and LED8.
f. Configure GPIO PA0 as EXTI0, and label as DW_RESET
g. Configure GPIO PB5 as EXTI5, and label as DW_IRQn
h. Configure GPIO PB0 as GPIO_Output and label as DW_WUP
i. In the ***MiddelWares*** section enable USB_DEVICE and select Communications Device Class (Virtual Port Com) as Class for FS IP, this will configure PA12, PA11 and PA9 GPIOs, note in the USB_OTG_FS peripheral select Device_Only and activate VBUS. Label PA9, PA11 and PA12 as USB_VBUS, USB_DM and USB_DP, also select PA10 as GPIO_Input and label as USB_ID.
j. Configure PB3 as JTDO, PB4 as JTRST, PA13 as JTMS, PA14 as JTCK and PA15 as JTMS. Label the pins as J_TDO, J_TRST, J_TMS, _JTCK and J_TMS respectively.
k. Configure PB2 as GPIO_Input and label as PB2_BOOT1
l. Configure PC0, PC1, PC2, PC3, PC4 and PC5 and GPIO_Input and label as PC0, PC1, PC2, PC3, PC4 and PC5 respectively.

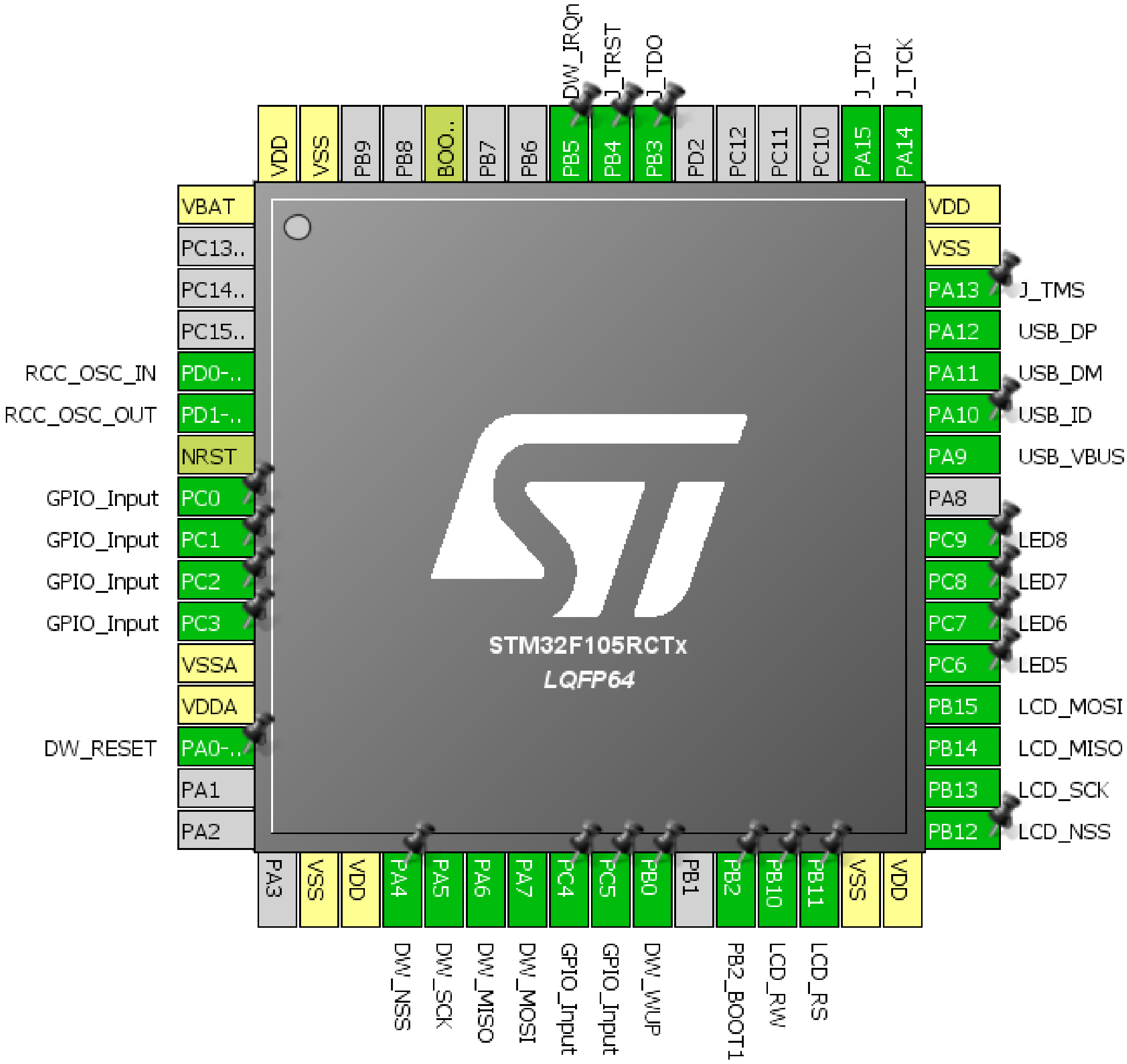


Figure 1: STM32 pinout after configuration

2.5 Check clock configuration

We would like the HCLK to be at 72 MHz and the USB to be at 48 MHz. To achieve this firstly set Input frequency to match the oscillator on the EVB1000, set this to 12 MHz, see Figure 2. Then Set USB to 48 MHz, and finally set APB2 clock to 72 MHz.

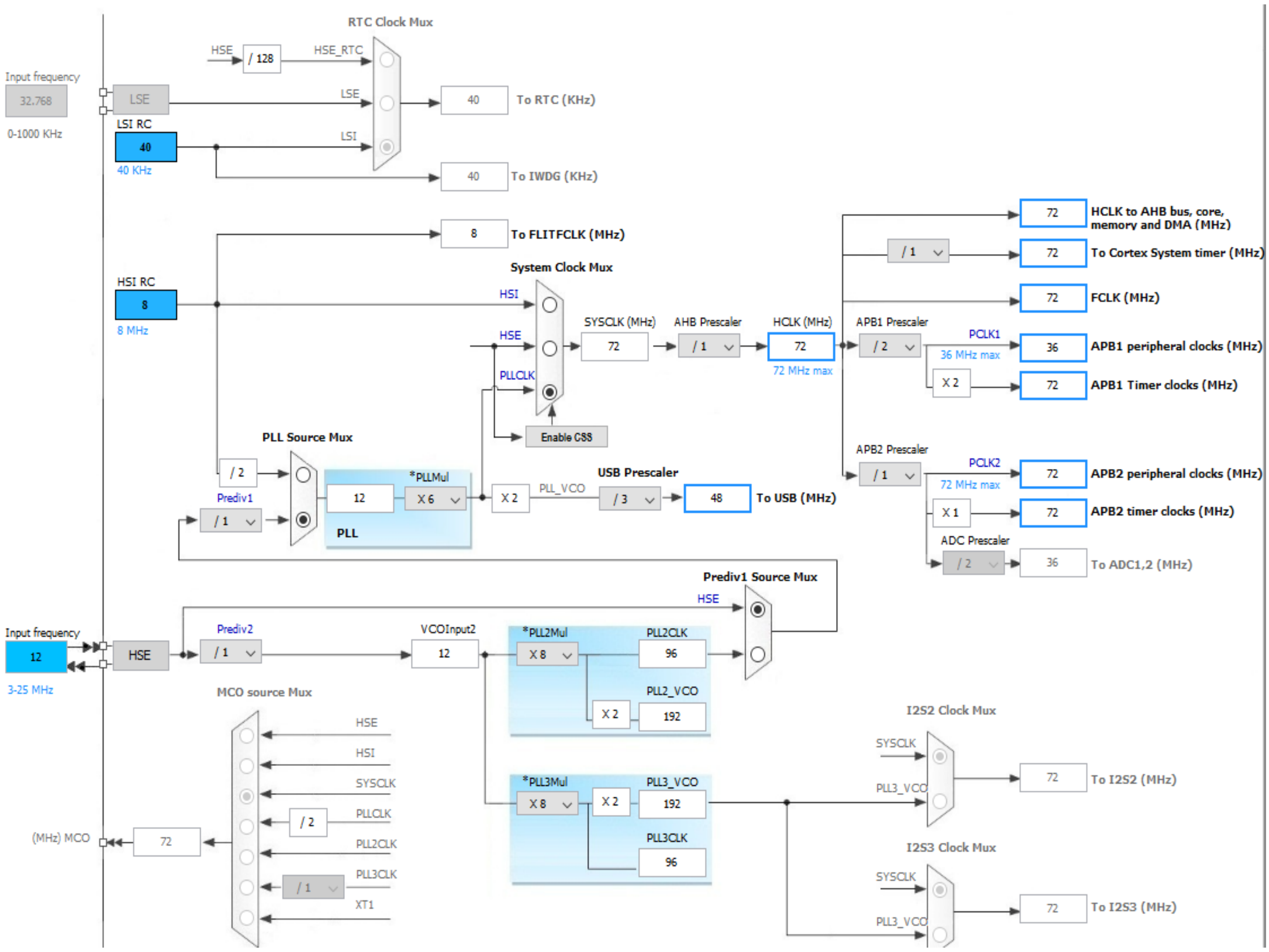


Figure 2: Clock configuration

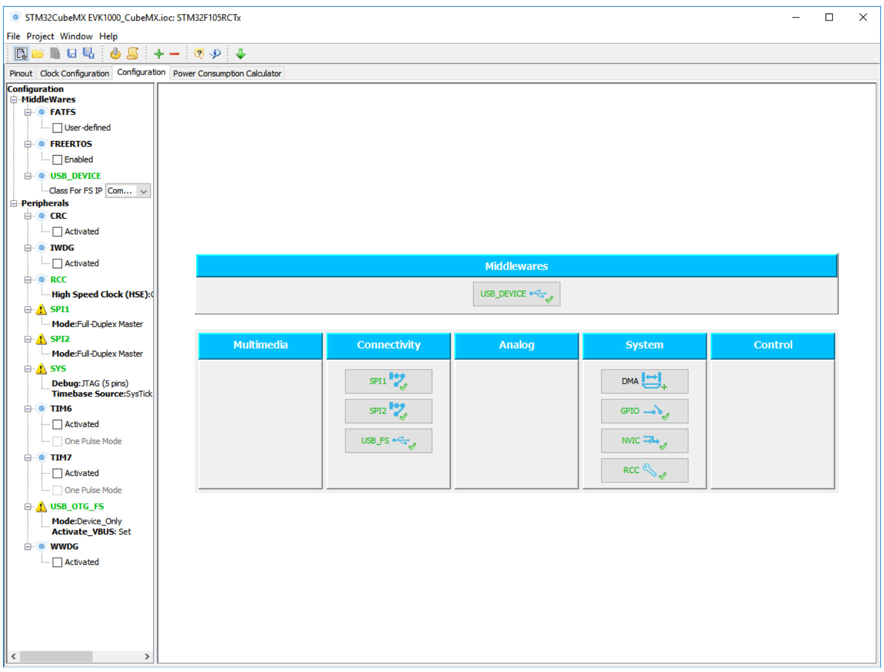


Figure 3: Configuration tab

2.6 Configuration Tab

a. SPI1 change the prescaler 32 to give 2.25 MBits/s, the polarity should be low and phase set to 1st edge. In the GPIO settings set MISO to Pull-down. Other settings can be left at defaults.

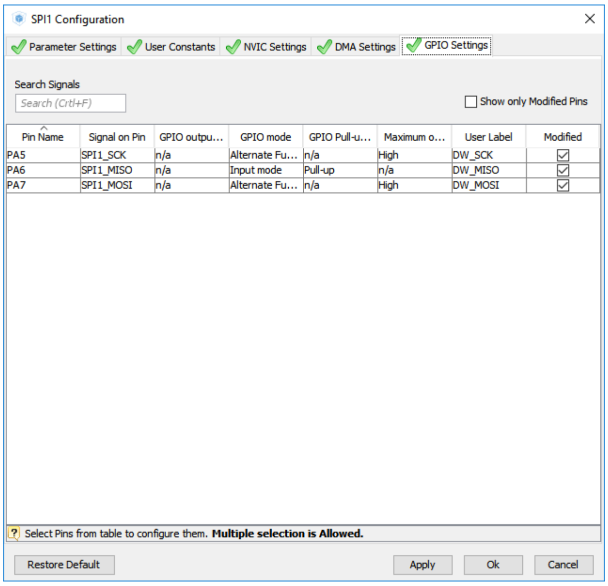


Figure 4: SPI1 configuration

b. SPI2 change the prescaler to 128 to give 281.25 Kbits/s, the polarity should be high and phase set to 2nd edge.

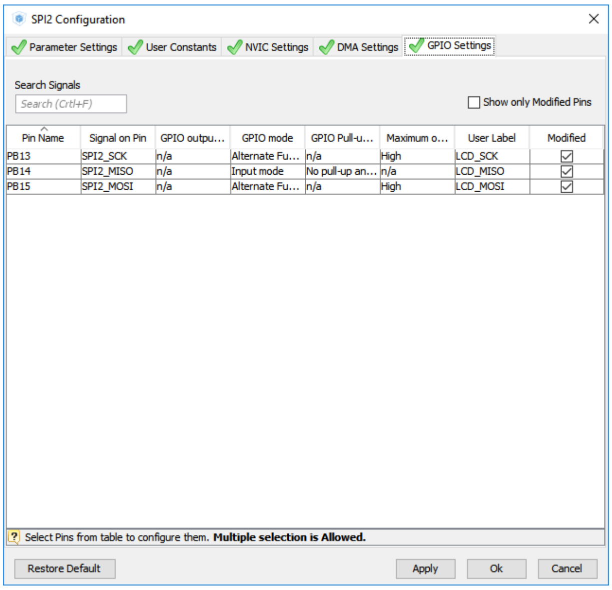


Figure 5: SPI2 configuration

c. GPIOs:
- Set PB5 as Pull-down
- Set PB0 as Open drain
- Set all Output GPIOs to have high max output speed

- Set PB0, PA4 and PB12 output to High

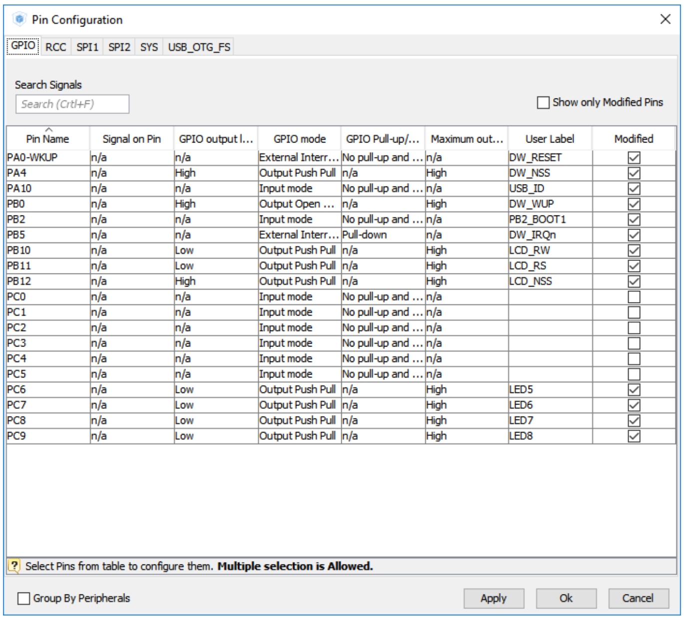


Figure 6: GPIO configuration

d. NVIC:
- Select EXTI line0 interrupt, and set priority to 3
- Select EXTI line[9:5] interrupts and set priority to 4
- Set USB interrupt priority to 7
- Set tick timer priority to 5
- Select EXTI0, EXTI[9:5] and USB for init seq. ordering, the rank does not matter.

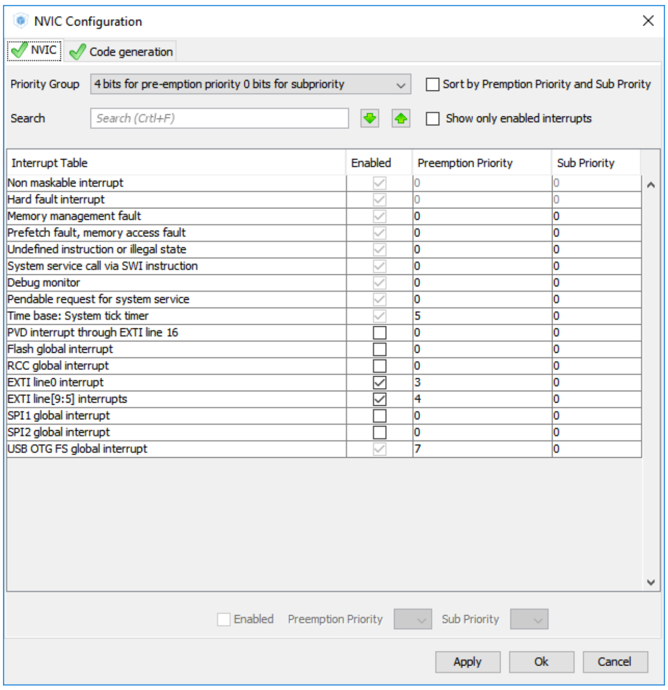


Figure 7: NVIC configuration

2.7 *Project settings*

In the Project Settings configuration:

a. Assign project name e.g. EVK1000_CubeMX, and select location, i.e. where the code will be generated
b. Select Toolchain / IDE – set as SW4STM32
c. Select “Generate Under Root”
d. In the Code Generator tab: HAL Settings set all free pins as analog

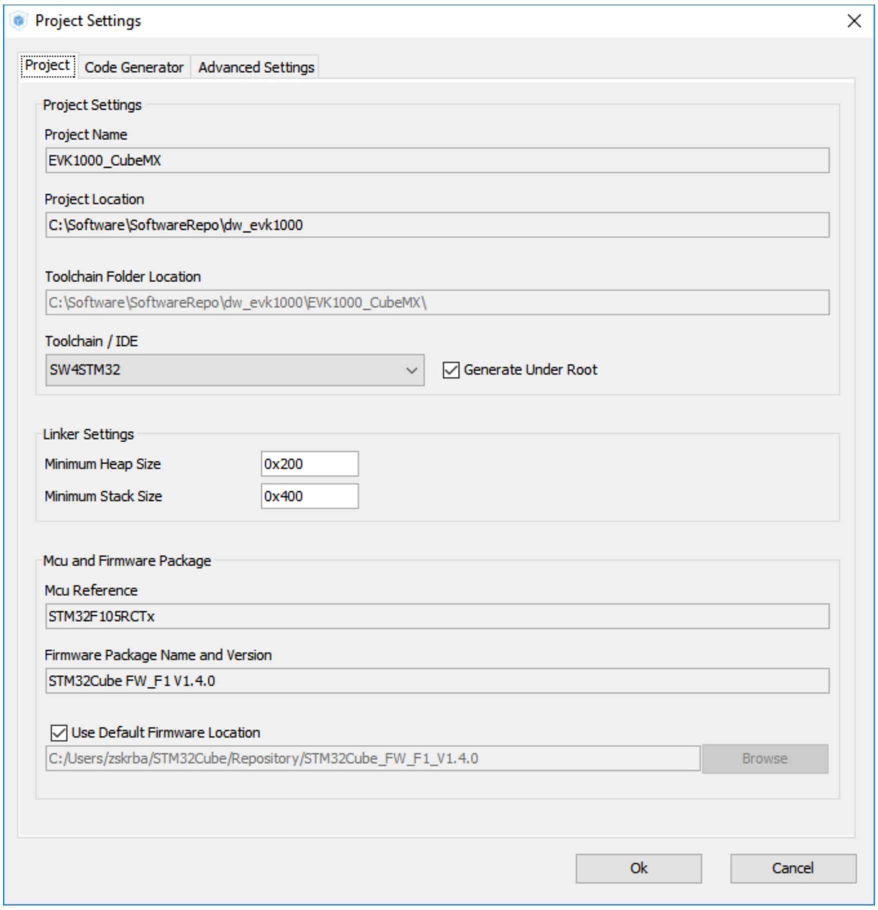


Figure 8: Project Settings

2.8 *Generate Code*

Once you reach this step all the project settings should be configured. Save the project and then generate the code. Figure 9 shows the generated folder structure and files.

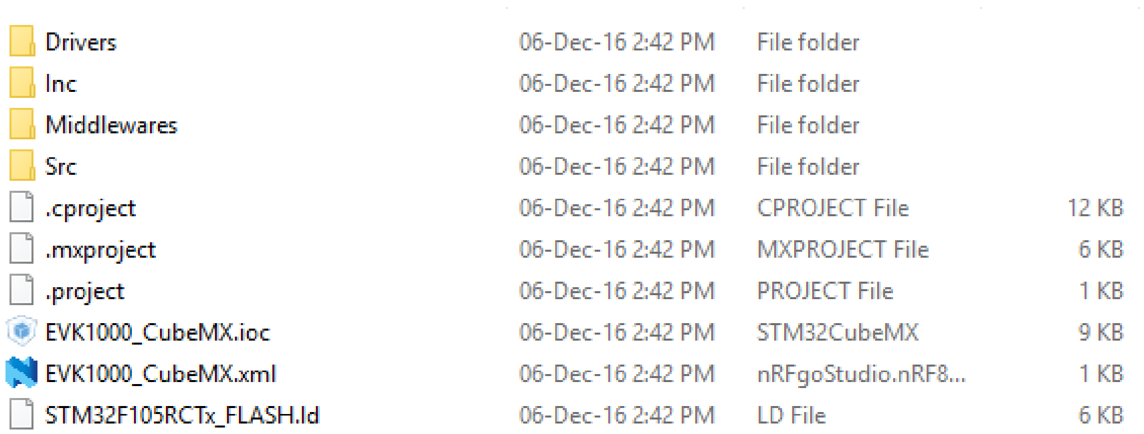


Figure 9: Folder containing the generated project files and code

3 Building Simple Application in ST Workbench

Import newly generated project into ST system Workbench IDE. Go to File->Import-> and select the folder where the project has been generated.

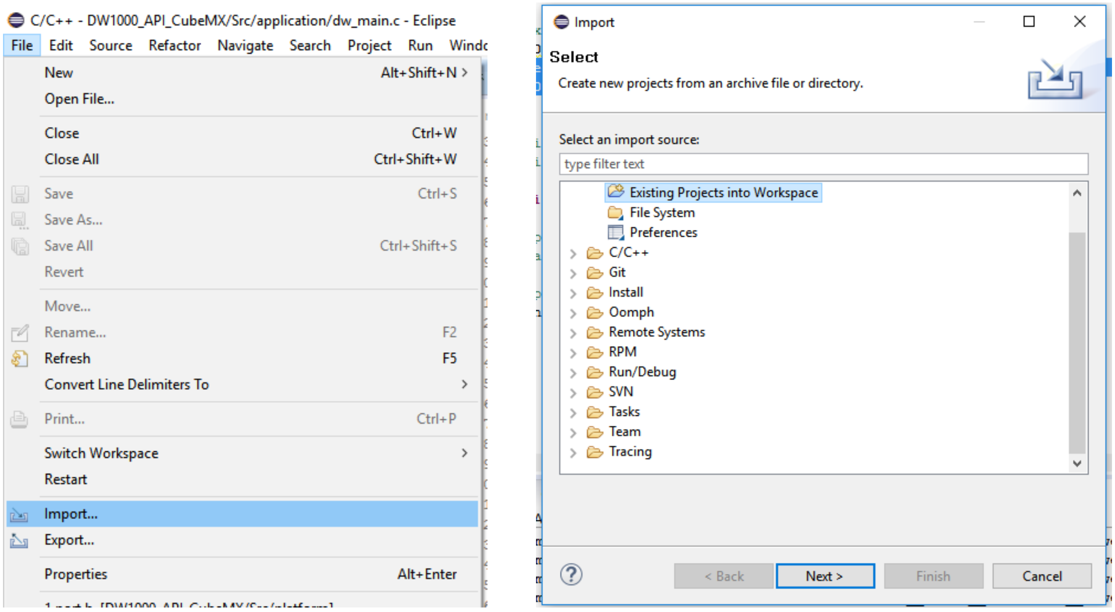


Figure 10: Importing newly generated project

Then in the project settings select internal builder, and build the project by clicking the build icon 

To download to the EVB1000, press , and select Ac6 STM32 Application, then select Debug device (e.g. ST-LinkV2) and download to the EVB1000.

Once the project can be built successfully and downloaded to the EVB1000, you can also visually check that the code is running, by adding following two lines into main while (1) loop:

```
while (1)
{
/* USER CODE END WHILE */
      HAL_GPIO_TogglePin(LED5_GPIO_Port, LED5_Pin);

      HAL_Delay(1000);
/* USER CODE BEGIN 3 */

}
```

This will flash LED5 on the EVB1000.

4 Porting EVK application into System Workbench IDE

Section above showed how to build and download simple application into EVB1000 HW. Next the user can port Decawave's EVK application.

1. Copy application files from EVK1000 Src\application folder into Src\application
2. Copy compiler file from EVK1000 Src\compiler folder into Src\compiler
3. Copy decadriver files from EVK1000 Src\decadriver folder into Src\decadriver
4. Copy deca_usb.h and deca_usb.c files from EVK1000 Src\usb folder into Src\usb
   Some of the CubeMX generated files which contain the library interface functions need to be modified:
   usbd_cdc_if.c, also deca_usb.h and deca_usb.c need to be modified to work with new HAL API. Please refer to the provided EVK1000 ST Workbench project.
5. Similarly the platform dependant files found in EVK1000 Src\platform need to be adapted for the new HAL. Especially SPI and other GPIO functions in port.h For this aging please refer to the provided EVK1000 ST Workbench project.
6. Add a call to dw_main() in the main.c

```
/* USER CODE BEGIN 0 */
extern void dw_main(void);
/* USER CODE END 0 */

while (1)
{
/* USER CODE END WHILE */
      dw_main();
/* USER CODE BEGIN 3 */

}
```

7. You are now ready to compile and build Decaranging EVK1000 code.
8. Note: In the project properties, C/C++ Build settings, in GCC Linker a –specs=nano.specs option should be removed to allow floating point printing to e.g. LCD.

4.1 Conclusion

This document gives the basic overview of how to use ST's tools (Cuble MX and System Workbench) and Decawave's EVB1000 HW to create applications on the EVB1000. This is a good starting point on which further application scan be developed.

5 References

5.1 Listing

References are made to the following documents in the course of this document:

Table 1: Table of References

Ref	Author	Version	Title
[1]	Decawave	Current	EVK1000 User Manual

6 Document History

6.1 Revision History

Table 2: Document History

Revision	Date	Description
1.0	15-Feb-2017	Initial release

7 Major Changes

Revision 1.0

Page	Change Description
All	Initial Release

8 About Decawave

Decawave is a pioneering fabless semiconductor company whose flagship product, the DW1000, is a complete, single chip CMOS Ultra-Wideband IC based on the IEEE 802.15.4-2011 UWB standard. This device is the first in a family of parts that will operate at data rates of 110 kbps, 850 kbps and 6.8 Mbps.

The resulting silicon has a wide range of standards-based applications for both Real Time Location Systems (RTLS) and Ultra Low Power Wireless Transceivers in areas as diverse as manufacturing, healthcare, lighting, security, transport, inventory & supply chain management.

Further Information

For further information on this or any other Decawave product contact a sales representative as follows: -

Decawave Ltd
Adelaide Chambers
Peter Street
Dublin 8
t: +353 1 6975030
e: sales@Decawave.com
w: www.Decawave.com