

# Taiyo Yuden Quick Start Guide for EYSHxN Module

Keil + No Packs Version

---

## Contents

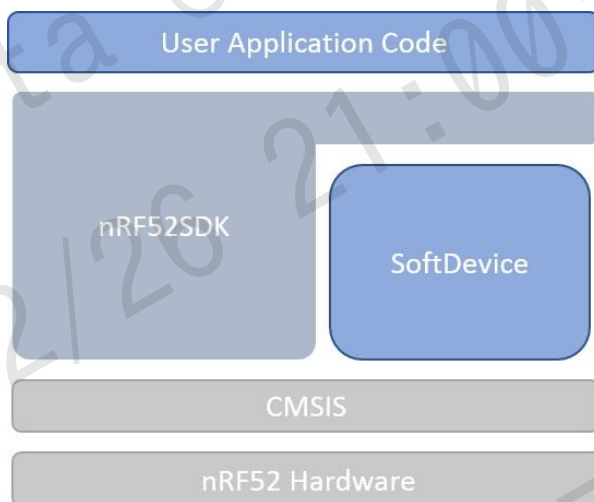
1. Introduction.....	3
2. About this Quick Start Guide .....	4
3. Software Needed.....	4
4. Installing the IDE.....	5
5. Installing nRFGo Studio.....	7
6. Installing the nRF52 SDK.....	7
7. Programming the SoftDevice .....	8
8. Setting up the BLE UART project.....	9
8.1. Setting up the EYSHxN target.....	9
8.2. Settings for Target Device.....	9
8.3. Setting up the target options .....	10
8.4. Setting up the debugger.....	11
9. Overview of BSP .....	12
9.1. Setting up the custom ports for each project .....	12
9.2. Adding the custom board for the SDK.....	14
10. Executing the Sample.....	15

## List of Figures

Figure 1 Software layout of nRF52832.....	3
Figure 2 Typical Memory Layout of nRF52832 .....	3
Figure 3 Keil Pack Manager.....	6
Figure 4 Installing the Keil Pack.....	6
Figure 5 Nordic's Developers Website.....	7
Figure 6 Using nRFGo Studio to program the SoftDevice.....	8
Figure 7 Setting the new Target Setting up the target device.....	9
Figure 8 Target Device Options .....	10
Figure 9 Target Memory Options.....	11
Figure 10 Target Debugger Options.....	11
Figure 11 Create Board Support.....	12
Figure 12 Defining pins for the board .....	13
Figure 13 Defining the Global Board Settings.....	14
Figure 14 Confirming the EYSHxN COM Board.....	15
Figure 15 Connecting the BLE Module.....	16

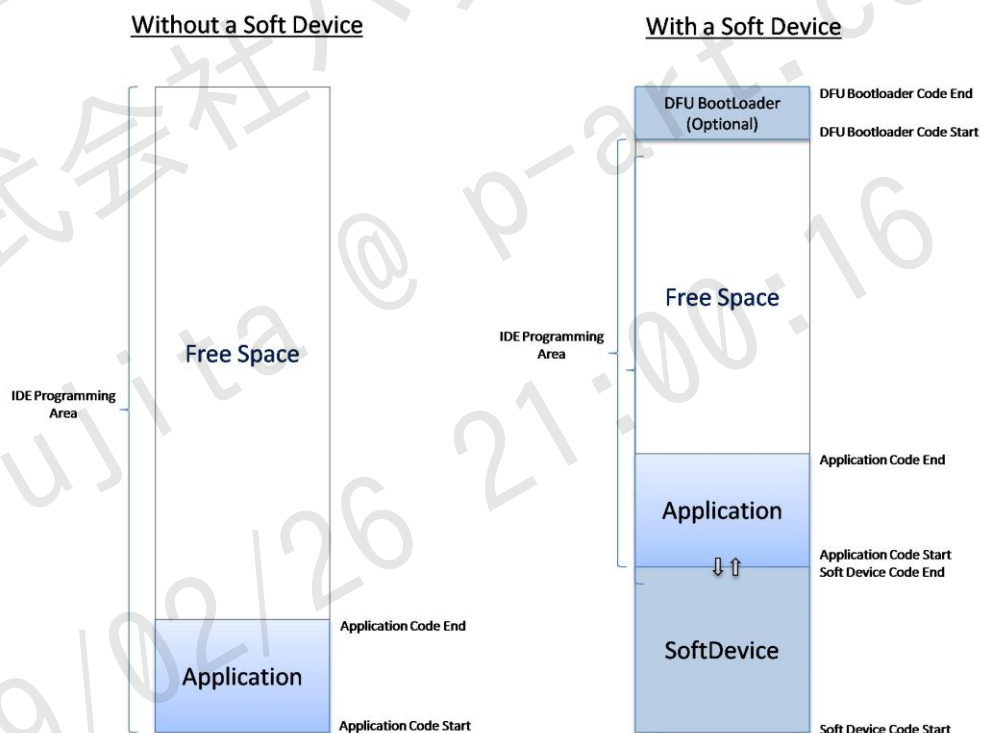
## 1. Introduction

The EYSHxN module comes with the Nordic's nRF52832 Bluetooth Low Energy 5(BLE5) module. For ease of programming this module, Nordic has released its SDK (Software Development Kit). To develop Bluetooth Low Energy, there are self-contained BLE static library stacks for ease connection to the API, these are called SoftDevices.



**Figure 1** Software layout of nRF52832

To use the “SoftDevice”, in the place of the device static memory, the static library is used. The SoftDevice is programmed to the module’s flash memory before the user application. If the Softdevice is being used it will be shown in the memory layout as shown below.



**Figure 2** Typical Memory Layout of nRF52832

## 2. About this Quick Start Guide

This quick start guide will show how to install the developing environment used to program the BLE UART application on the Taiyo Yuden EYSHxN module, using the Nordic's nRF SDK.

On this guide, we will be focusing on using the free version of the Keil IDE with the "No-Packs" (zip) version. Using the "No-packs" version of the SDK makes it easier to understand the SDK layout, as well as to change between the SDK versions, since it keeps the whole SDK in a single folder.

The needed things are as follows:

- Internet Connection
- Windows PC
- EYSHxN Board
- ARM Cortex M-9 Compatible Programmer (For example: J-Link Lite CortexM-9 JTAG/SWD Emulator)
- An Android or iOS device for testing
- Serial Terminal Software (For example: Termit, TerraTerm, RealTerm)

In the next steps, we will focus on explaining the different steps for developing the EYSHxN board:

- Installing the IDE
- Installing the NRF stack
- Setting the BLE UART Project
- Programming the SoftDevice
- Executing the samples

## 3. Software Needed

Including IAR, GCC and Keil, there are many developing environments that can be used to set up the EYSHxN module.

The recommended tools for testing the developed firmware on the EYSHxN module are as follows.

Tool Name	Used For	Download From
Keil uVision IDE (MDK-ARM)	Developing and programming the firmware application	Keil's Homepage( <a href="http://www.keil.com">www.keil.com</a> )
nRFGo Studio	To control the flash memory. To program the Softdevices and Bootloaders	Nordic's Homepage ( <a href="http://www.nordicsemi.com">www.nordicsemi.com</a> )
Nordic nRF52 SDK (For BLE5, it needs to be 14 or over)	It offers easier ways to control the hardware abstraction and support of the libraries for the nRF52832 module.	Nordic's Developers Homepage ( <a href="http://developer.nordicsemi.com">http://developer.nordicsemi.com</a> )

Table 1 Software Needed

## 4. Installing the IDE


Keil uVision IDE (MDK-ARM) is available to download in the Keil's Homepage (<http://www.keil.com>). From the time that this guide is written, the download is available here:

Downloads->Product Downloads->MDK-ARM

Registration is needed.

Start Keil uVision installer. The first time, the window showed on Figure 3 "Pack Installer" will appear.

Caution: The Pack Installer will automatically open as soon as the MDK-ARM installation is finished.

Caution: If after the IDE starts and the Pack Installer is not automatically started, go to Project -> Manage -> Pack Installer or press the  icon on the main menu to open.

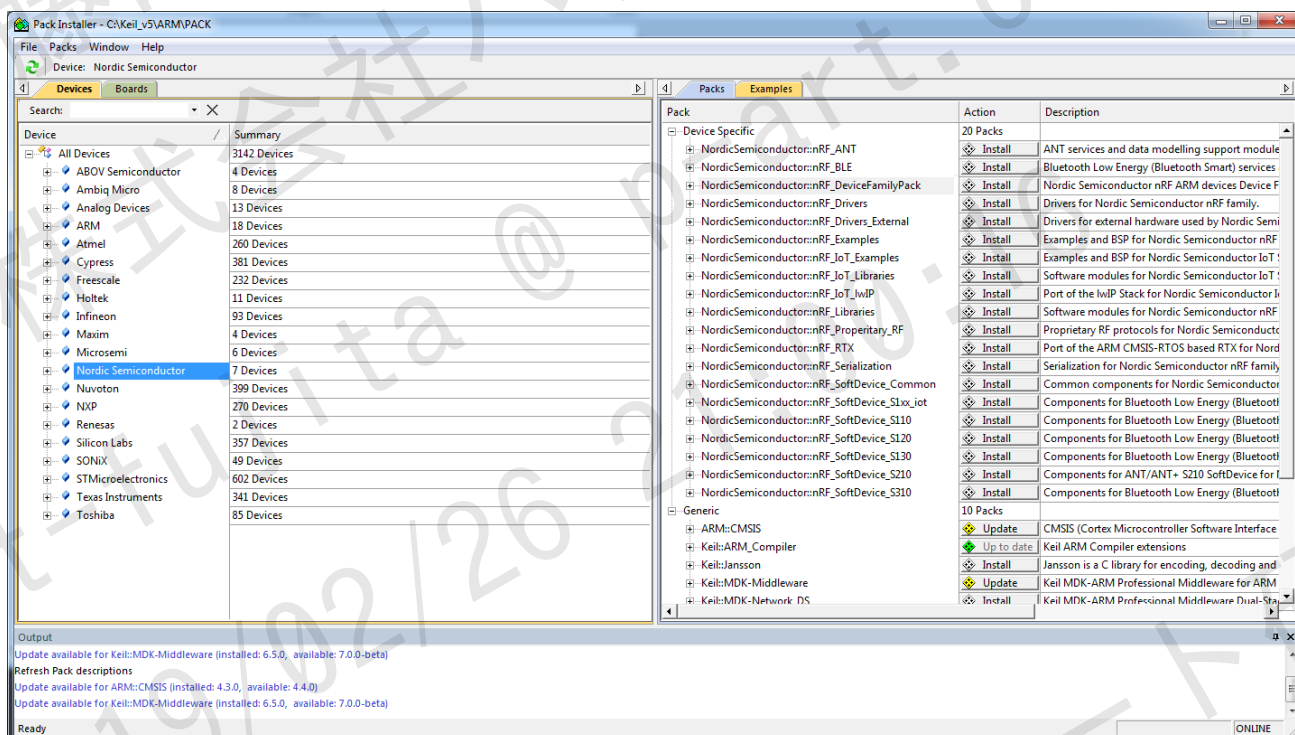



Figure 3 Keil Pack Manager

By pressing  or in the menu File -> Refresh Packs, the selected available Packs will update. If Refresh Packs is not available for selection, it might be possible that the update is already in progress or the system is offline. Confirmation can be seen on the bottom left corner of the window.

For the “No Packs” Nordic SDK Version in the settings, the only pack needed to be downloaded is the “Device Family Pack” which adds the nRF series devices to the Keils target database. (There are other ways to download the “Device Family Pack” but this is the easiest and safest way to obtain the latest target information.)

In the contents of the “Devices”, click on the “Nordic Semiconductor” and select install on the “NordicSemiconductor::nRF\_DeviceFamilyPack”.

After the installation is complete, the “Install” icon will change to “Up to date” as shown in the next figure.

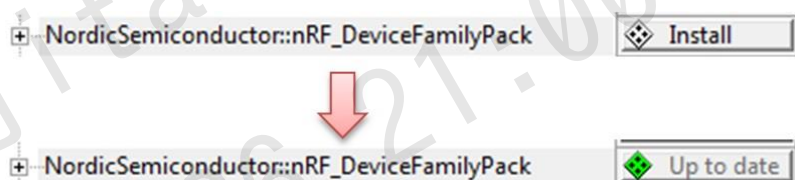


Figure 4 Installing the Keil Pack

Please do not install other packs. The other files are all included in the downloaded “No Packs” SDK.

## 5. Installing nRFGo Studio

In the Nordic Semiconductor homepage(<http://www.nordicsemi.com>), download the nRFGo-Studio.

The nRFGo-Studio download can be found in the” Downloads” in RF52832 product page. The installer comes with nRF-Tools, which includes additional support for other debuggers.

## 6. Installing the nRF52 SDK

The latest version of the nRF52 SDK can be directly downloaded from Nordic Developer’s Website (<http://developer.nordicsemi.com>). The SDK are shown on top of the page.

**For BLE5, please download SDK14 and above (Latest)**

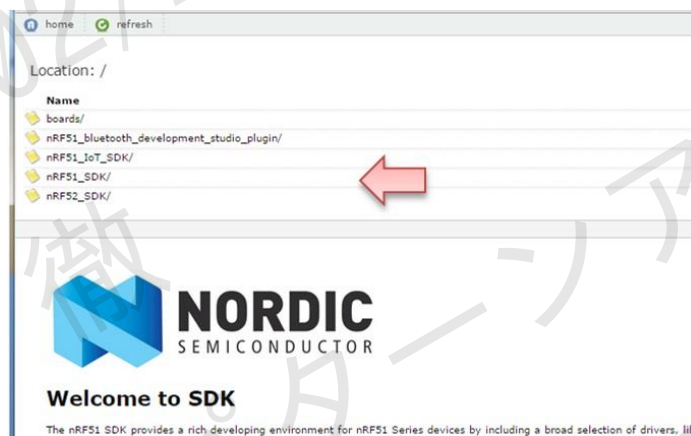


Figure 5 Nordic’s Developers Website

On the time this guide was created, the latest version of the” no packs” SDK can be downloaded from here:

nRF5\_SDK -> nRF5\_SDK\_vx.x.x -> nRF5\_SDK\_X.X.X\_yyyyyyy.zip

The x is the SDK version.

**Caution: This guide is focused on” no packs”. Please do not download the” packs” SDK version.**

Unzip the folder in your working directory. If a Japanese PC is being used, please avoid using Japanese named paths. (For example: C:\Nordic52SDK)

**Caution: The SDK installer inside the unzipped folder is not needed for the environment so it can be removed without any problem. Please do not execute it.**

After this point in the guide, the location of the SDK is “sdkroot”.



## 7. Programming the SoftDevice

The BLE UART Demo uses the S132 BLE SoftDevice. This needs to be programmed to the module before the actual application.

To start, connect the EYSHxN Board to the PC.

**Caution:** The programmer doesn't power the EYSHxN board. To power the board, please connect it through Mini-USB.

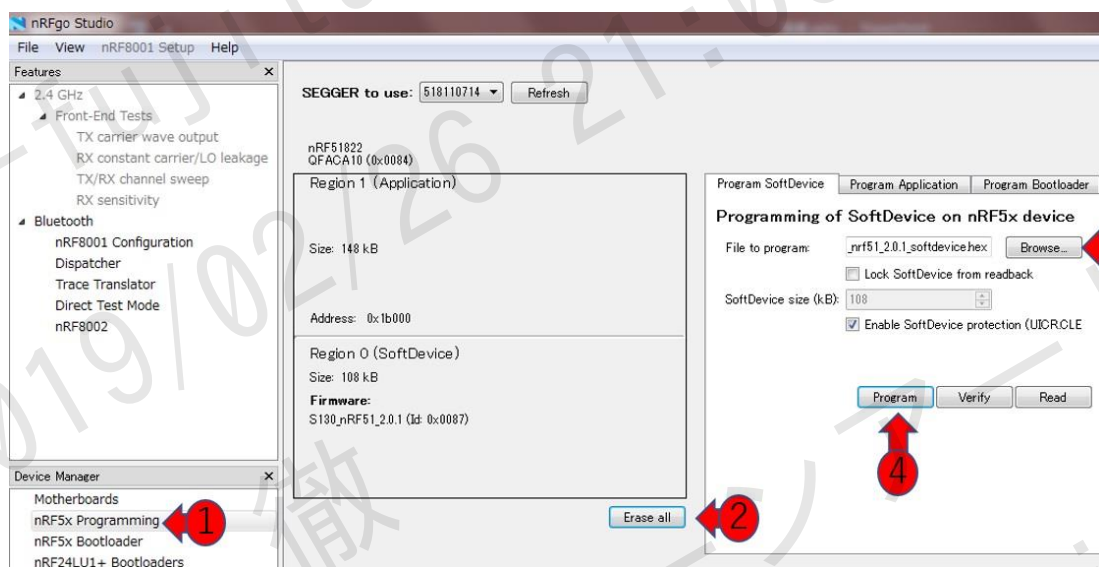


Figure 6 Using nRGo Studio to program the SoftDevice

1. Start nRGo Studio, on the left menu select "nRF5x Programming".
2. Delete the device's whole memory.
3. Make sure that the "Program SoftDevice" tab is selected, in "Browse" please refer to the SoftDevice. Use S132 SoftDevice as shown below:

*sdkroot/components/softdevice/s132/hex/s132\_nrf52\_XX.X\_softdevice.hex*

Or from the Nordic Semiconductor nRF52832 product page, the latest S132 SoftDevice is available for download.

4. Click "Program" to program the SoftDevice into the module.

nRGo studio displays the current memory map of the SoftDevice installed at the bottom and the application area directly above it.




## 8. Setting up the BLE UART project

All the nRF51 SDK project files are set up for PCA10040 developer board. The easiest way to use the sample for the EYSHxN board is using the PCA10040 project and modify it to be used on the EYSHxN.

In the place below, open the SoftDevice S132 no packs BLE UART sample project:

*sdkroot/examples/ble\_peripheral/ble\_app\_uart/pca10040/s132/arm5\_no\_packs*

### 8.1. Setting up the EYSHxN target

In the main window click the  icon and as shown below, create a new target name: Project->Manage->Project Items


Pressing on  adds a new target name. Call it “taiyoyuden\_EYSHxN\_S132” and click on “Set as Current Target”.

As Figure 7 shows, the target folder and the drop-down will show “taiyoyuden\_EYSHxN\_S132”. If it doesn’t show, please select it on the drop-down menu.



Figure 7 Setting the new Target Setting up the target device

### 8.2. Settings for Target Device

As explained in 8.1, after confirming that “taiyoyuden\_EYSHxN\_S132” settings are selected, in the main window click the  icon or open the target options as shown:

Project -> Options for target taiyoyuden\_EYSHxN\_S132

In the “Device” tab select nRF52832xxAA.

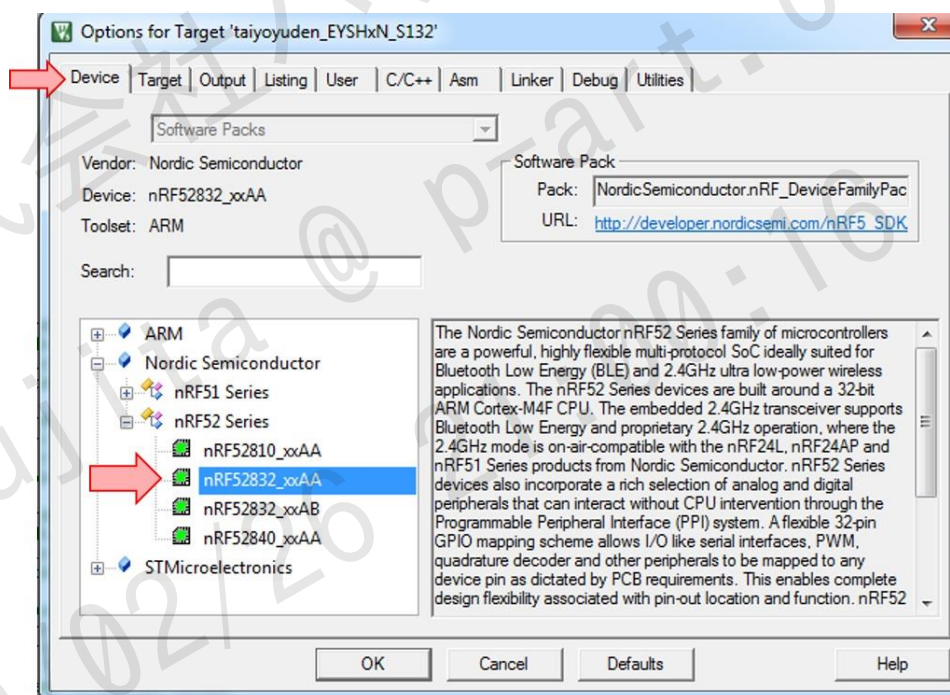


Figure 8 Target Device Options

### 8.3. Setting up the target options

In the “Target” tab change the Xtal(MHz) to 64.0.

Next, it's necessary to restrict the IDE's chip code memory(flash) and RAM (See Figure 2) for the application. The exact settings depend on application's environment (SoftDevice Version, Bootloader, and other static libraries). In this case, as S132 SoftDevice is used, the chip's remaining memory is freely used by the application.

The SoftDevice ROM and RAM size usage, it's described on the specifications depending on the version of SoftDevice. The start of the application area is indicated as APP\_CODE\_BASE and APP\_RAM\_BASE.

nRF52832's flash code memory address starts at 0x00000 and the size is 512kb(0x80000)/ The RAM starts in the address 0x2000000 and the size can be up 64kb(0x4000).

Figure 9 shows memory settings of the application using 64kb RAM module (EYSHxNZWZ) and SoftDevice S132 v5.0.0. This SoftDevice uses 0x2300 of flash memory and 0x20002800 of RAM.

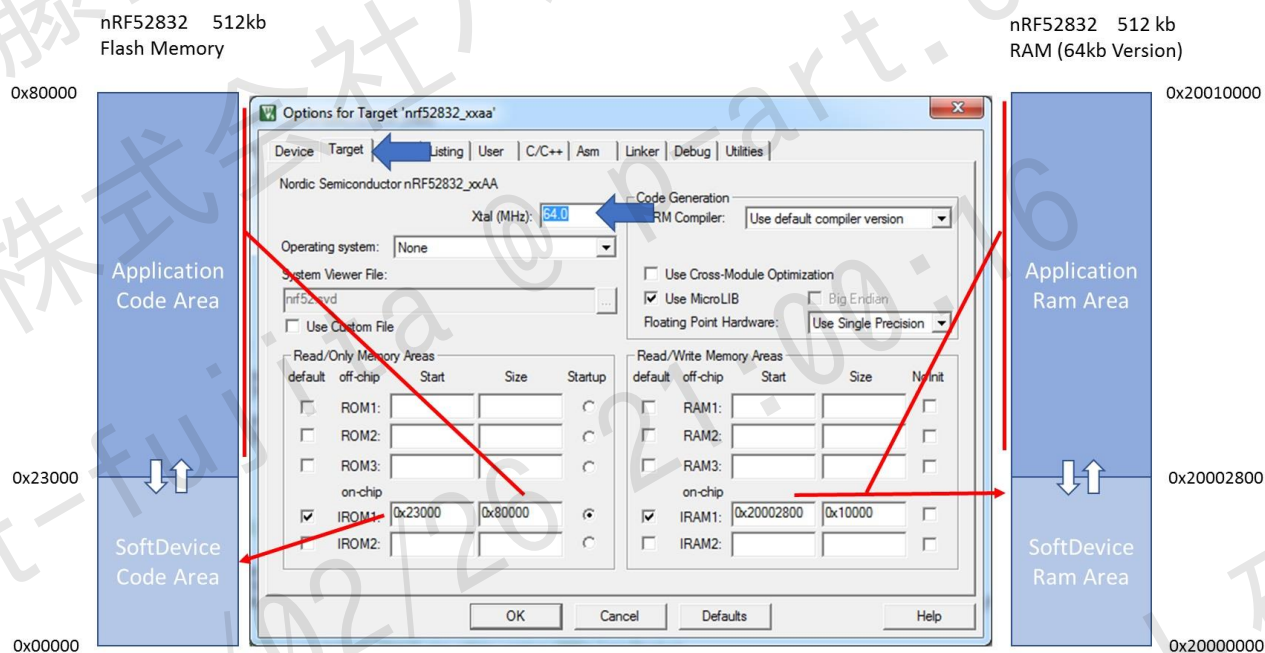


Figure 9 Target Memory Options

**Caution:** If memory settings are incorrect the application will not run correctly. If using different SoftDevice versions, the ROM and RAM values will be different. Always check the memory setting of the SoftDevice from the SoftDevice specification and the appropriate IDE memory area.

#### 8.4. Setting up the debugger

In the "Debug" tab, change the debugger using the drop-down menu to the debugger currently in use.

As Figure 10 shows, "J-Link Lite" debugger has been selected.

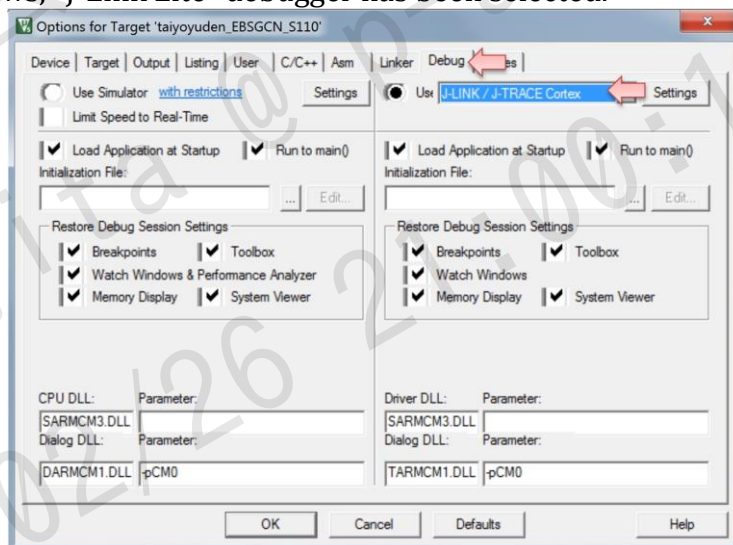


Figure 10 Target Debugger Options

## 9. Overview of BSP

When using IO or peripherals, the nRF52 SDK uses preprocessor pin definitions so transferring to different modules becomes an easier task.

All the pin configurations are contained in the "Board Support File" (bsp), so it can be easily adjusted depending on the hardware. The starting SDK sample is set up for the PCA10040 board. To use the EYSHxN module pin configurations need to be adjusted.

Before executing most of the SDK peripheral samples, pins for the EYSHxN need to be adjusted.

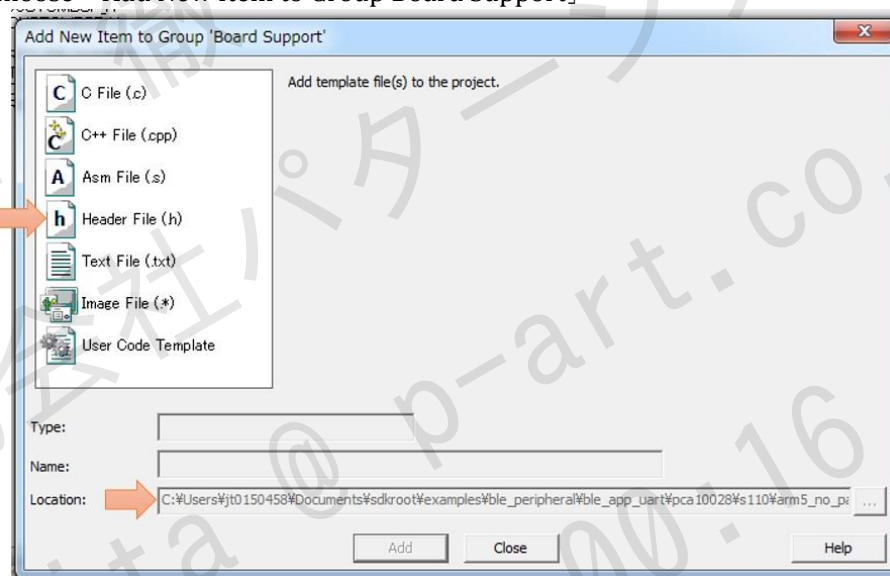
### 9.1. Setting up the custom ports for each project

A file called "custom\_board.h" can be created in the project directory for your custom board. In this example, we will create the "custom\_board.h" file in following location:

*sdkroot /examples/ble\_peripheral/ble\_app\_uart/pca10040/s132/arm5\_no\_packs/RTE*

Using Keil IDE to add the file, use the following procedure.

- ① Right click on the 「Board Support」 folder
- ② Choose 「Add New Item to Group Board Support」



**Figure 11 Create Board Support**

After creating the Board Support File, the pins used on the BLE UART Sample need to be defined.

The Mini-USB connector on the EYSHxN development board, includes TTL-UART. To use the UART through the connector, pins need to be set as shown in Table 2.

Caution: If other functions (LED、SPI) are needed, pins need to be defined per design.



Hint: EYSHxN's through hole pitch, it's compatible with the standard breadboard's solderless prototyping.

Define Name	Function	Pin
RX_PIN_NUMBER	UART Receive	8
TX_PIN_NUMBER	UART Transmit	6
CTS_PIN_NUMBER	UART CTS	7
RTS_PIN_NUMBER	UART RTS	5

**Table 2 Settings for BLE UART demo**

The header file of the "custom\_board.h" is as shown.

In addition, pin 25 is defined as the LED and pin 24 as the button. The board itself does not contain a button or and LED, to use these, please connect LED or button respectively.

```

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
#ifndef CUSTOMBOARD_H
#define CUSTOMBOARD_H

#ifdef __cplusplus
extern "C" {
#endif

#include "nrf_gpio.h"

// LEDs definitions for PCA10040
#define LEDS_NUMBER 0

#define LED_START 25
#define LED_1 25
#define LED_STOP 25

#define LEDS_ACTIVE_STATE 0

#define LEDS_INV_MASK LEDS_MASK

#define LEDS_LIST {LED_1}

#define BSP_LED_0 LED_1

#define BUTTONS_NUMBER 1

#define BUTTON_START 24
#define BUTTON_1 24
#define BUTTON_STOP 24
#define BUTTON_PULL NRF_GPIO_PIN_PULLUP

#define BUTTONS_ACTIVE_STATE 0

#define BUTTONS_LIST {BUTTON_1}

#define BSP_BUTTON_0 BUTTON_1

#define RX_PIN_NUMBER 8
#define TX_PIN_NUMBER 6
#define CTS_PIN_NUMBER 7
#define RTS_PIN_NUMBER 5


// Low frequency clock source to be used by the SoftDevice
#define NRF_CLOCK_LFCLKSRC {source = NRF_CLOCK_LF_SRC_XTAL, \
rc_ctiv = 0, \
rc_temp_ctiv = 0, \
xtal_accuracy = NRF_CLOCK_LF_XTAL_ACCURACY_20_PPM}

#ifdef __cplusplus
}
#endif

```

**Figure 12 Defining pins for the board**

Once the custom board file is created, the global preprocessor definition used by the board support system needs to be changed to "BOARD\_CUSTOM".

As 8.1 explained, ensure that the target is set to "taiyoyuden\_EYSHxN\_S132", click the  icon in the main window and open the target options:

Project -> Options for target taiyoyuden\_EYSHxN\_S132

On the "C/C++" tab, as shown in [Figure 13](#) change the "BOARD\_PCA10040" to "BOARD\_CUSTOM".

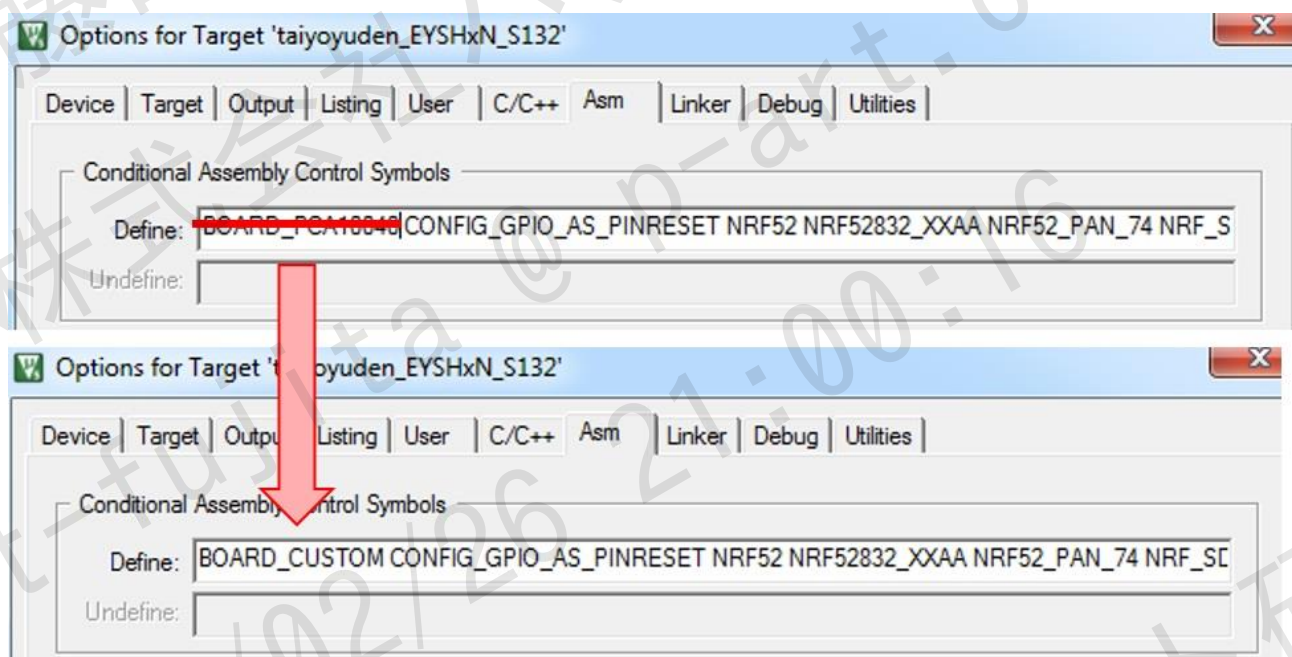


Figure 13 Defining the Global Board Settings

For more information on how to use board definitions see *sdkroot/ components/boards /boards.h* file and the "BLE Development Manual".

## 9.2. Adding the custom board for the SDK

Instead of creating a "custom\_board.h" for each project, you can permanently add your own board to the future SDK projects. Details will not be explained in this Start Up Guide. Please refer to the "BLE development manual" for more information.



## 10. Executing the Sample

Connect the programmer to both the PC and the EYSHxN board, and at the same time connect the EYSHxN board to the PC through the Micro-USB.

Connect the GCN board and the Micro USB connector of the EYSHxN board to the PC.

When connected, the EYSHxN board will appear in the “Device Manager” as “USB Serial Port”. The way to open the device manager window depends on the Windows version and will not be explained on this Quick Start Guide.

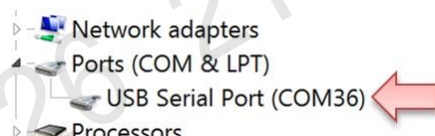




Figure 14 Confirming the EYSHxN COM Board

**Caution:** If the EYSHxN is not shown correctly, please download the FT232RG driver from the FTDI website.  
(<http://www.ftdichip.com/Drivers/D2XX.htm>)

The project will be compiled if this  icon is pressed or and following the bottom menus are followed.

Project -> Rebuild all target files

To download the compiled application into the module, press this  icon or follow the steps below through the menus.

Flash -> Download

The code is downloaded to the module and executed.

**Caution:** If the error message “Error: Flash download failed – Cortex-M4” appears, the SoftDevice memory options are not set correctly. Please refer to chapter 7.

To test the program, connect the terminal software (TerraTerm、RealTerm、Termite) to the EYSHxN COM Port with the settings shown below.

Specification	Value
Baudrate	115200
Parity	None
Data Bits	8
Stop Bits	1

Table 3 Serial Settings for BLE UART demo

After connecting to the terminal software, download "nRF Toolbox" application from the App Store or Play Store (iOS or Android) to the tester mobile device.

Open "nRF Toolbox" application, select "UART" demo and click "CONNECT" to discover the device. Select "Nordic\_UART" and once connected all button operations will send signals to the EYSHxN module through BLE and will be displayed as text in the terminal software.

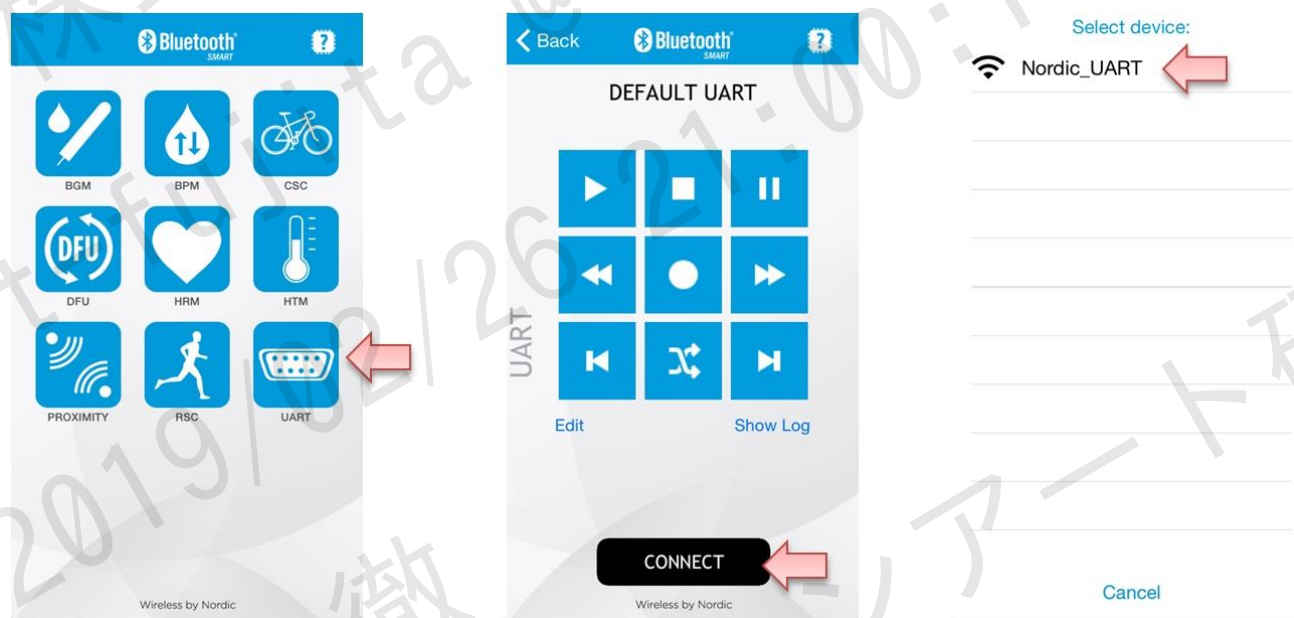


Figure 15 Connecting the BLE Module