

INSTALLATION GUIDE

Eclipse IDE in firmware development with IOsonata

Version 1.3



Revision history

Version	Date	Note	Contributor(s)	Approver
1.0	12 Dec 2018	Initial version	Nguyen Hoang Hoan	Nguyen Hoang Hoan
1.1	2019		Nguyen Hoang Hoan	Nguyen Hoang Hoan
1.2	2020		Nguyen Hoang Hoan	Nguyen Hoang Hoan
1.3	2021		Nguyen Hoang Hoan	Nguyen Hoang Hoan



Copyright © 2019 I-SYST, all rights reserved.

3514, 1re Rue, Saint-Hubert, QC., Canada J3Y 8Y5

This document may not be reproduced in any form without, express written consent from I-SYST.



Contents

1. Introduction	5
1.1 Required components	5
2. Installation	5
2.1 Installing ARM GCC compiler	5
2.2 Installing Build Tools for Windows	5
2.3 Installing OpenOCD for Source Debugging	5
2.3.1 For OSX use	5
2.3.2 For Windows and Linux use	5
2.4 Installing IOsonata and its dependencies	5
2.5 Installing Eclipse	7



1. Introduction

This document shows step-by-step how easy it is to install the Eclipse IDE in firmware development with IOsonata.

1.1 Required components

The following are needed for a full development environment for IOsonata and Nordic SDK:

- Eclipse CDT (for C/C++ Development) with GNU MCU plugins
- ARM GCC compiler
- OpenOCD for debugging
- IDAPnRFProg command line utility for flashing
- The SDK and all the external libraries

2. Installation

2.1 Installing ARM GCC compiler

Download the ARM GCC compiler package for your OS

<u>GNU Toolchain | GNU Arm Embedded Toolchain Downloads – Arm Developer</u>

You can either select an installer or a tar/zip package. Once installation is completed, note where you have installed it. You'll need this to set the full path to the compiler later in Eclipse settings.

The current installer version is GNU Arm Embedded Toolchain: 10-2020-q4-major December 11, 2020

2.2 Installing Build Tools for Windows

Follow these instruction to install the xPack Windows Build Tools binaries (not required on MacOS and GNU/Linux, use the system tools)

How to install the xPack Windows Build Tools binaries | The xPack Project

2.3 Installing OpenOCD for Source Debugging

In order to do source level debugging in Eclipse, OpenOCD is required. Installing OpenOCD differs depending on which OS your PC is running.

2.3.1 For OSX use

Use this command in the CLI: brew install openocd --HEAD

2.3.2 For Windows and Linux use

Follow these instructions on GNU MCU

How to install the xPack OpenOCD binaries | The xPack Project

Again, remember the path location where OpenOCD was installed. This path will be set in Eclipse settings later

2.4 Installing IOsonata and its dependencies

IOsonata is an open source, multi-architecture, highly optimized, hardware abstraction library. Compiling the IOsonata target libraries requires external SDK & libraries.

Follow the instructions below to download and install with appropriate locations and naming:

nRF5_SDK: Nordic nRF5x Bluetooth Low Energy. Select the latest nRF5_SDK. Unzip it and rename the folder to nRF5_SDK



nrf5_SDK_Mesh: Nordic nRF5 SDK for Bluetooth Mesh. Unzip it & rename the folder to nrf5_SDK_Mesh.

ICM-20948 Motion_Driver: First, create a user. In the "Development Kits" block, download "DK-20948 SmartMotion eMD 1.1.0". Unzip the downloaded file and navigate to EMD-Core/sources. Copy the folder Invn to external/Invn as indicated in the folder tree below.

BSEC: Bosch Sensortec Environmental Cluster (BSEC) Software for #BME680 environmental sensor. BSEC is needed for calculating Air Quality Index.

Go to https://www.bosch-sensortec.com/bst/products/all_products/bsec. At the end of the page select the checkbox to accept license terms and download. Unzip the downloaded file. Rename the extracted folder BSEC, then copy the whole folder to external as indicated in the folder tree below.

LWIP: A Lightweight TCP/IP stack. This library is required for IoT network connectivity over Ethernet, Wifi, LTE etc. Download it via this link. Rename the extracted folder as lwip and copy it to external.

The way the IOsonata folder is structured is simple. The deeper you go inside, the more specific it is to the architecture or platform. The parent folder contains everything commonly available to the child folder. This means source files from the child folder can access any source in the upper parent folder, but not the other way around. This keeps the abstraction separated from implementation and makes it easier to keep track of things.

```
/your_root
             - Development root directory
 -- external
                   - Contains downloaded SDKs from silicon vendors
    -- nRF5_SDK

    Latest Nordic SDK (https://developer.nordicsemi.com)

    | | |-- components
    | | |-- examples
 T
    |...
 -- nRF5_SDK_12
                      - Last version of Nordick SDK12 for nRF51 series
    | | -- components
 | | |-- examples
    1...
 |-- nrf5_SDK_Mesh - Latest Nordic SDK for Mesh
    | | |-- Mesh
    | | |-- Models
 |...
    -- BSEC
                      - Bosch Sensortec Environmental Cluster (BSEC) Software (https://www.bosch-
sensortec.com/bst/products/all_products/bsec) for #BME680
 -- Invn
                      - Invensense SmartMotion Driver (download
https://www.invensense.com/developers)
   | |-- Devices
 | |...
 |-- lwip
                      - Lightweight TCP/IP stack (download
https://download.savannah.nongnu.org/releases/lwip/)
   |-- Others as require
 | |...
   -- IOsonata
                - Put the IOsonata here
    -- include - Generic include common to all platforms
      -- bluetooth - Generic definition for Bluetooth
```



|-- converters - Generic definition for ADV, DAC, etc... -- coredev - Generic definition MCU builtin devices such as i2c, uart, spi, timer, etc... |-- miscdev - Generic definition for other non categorized devices -- sensors - Generic definition for al sort of sensors (environmental, motion, etc...) |-- usb - Generic definition for USB |... - Generic implementation source common to all platforms -- src -- bluetooth - Generic source for Bluetooth |-- converters - Generic source for ADV, DAC, etc... -- coredev - Generic source for MCU builtin devices such as i2c, uart, spi, timer, etc... -- miscdev - Generic source for other non categorized devices -- sensors - Generic source for al sort of sensors (environmental, motion, etc...) -- usb - Generic source for USB |... -- ARM - ARM series based MCU -- include - Common include for all ARM platform -- src - Common source for all ARM platform |-- DbgConfig - Debugger configuration files. - Linker script files -- ldscript -- Nordic - Nordic Semiconductor based MCU -- nRF52 - nRF52 serie MCU -- include - Common include for this target series -- src - Common source for this target series -- nRF52832 - Target MCU - IOsonata library for this target |-- lib |-- Eclipse - Eclipse project for this lib -- IAR - IAR project for this lib |-- CrossWorks- CrossWorks project for this lib |... |-- exemples - Example projects for this target -- Blink - Blink example Т -- src - Source code for this exaple |-- Eclipse - Eclipse project for this example -- IAR - IAR project for this example |-- CrossWorks- CrossWorks project for this example |... |-- Many other examples same -- nRF52840 - Target MCU |-- lib - IOsonata library for this target -- Eclipse - Eclipse project for this lib -- IAR - IAR project for this lib -- CrossWorks- CrossWorks project for this lib |...

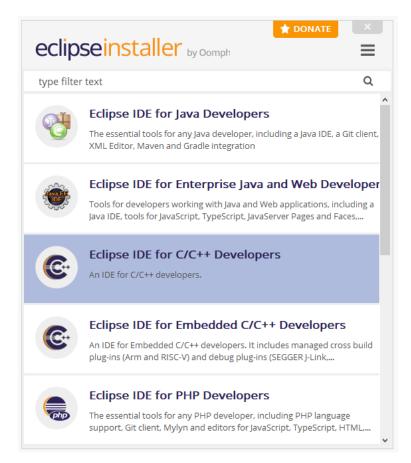


I					exemples - Example projects for this target
I					Blink - Blink example
I					src - Source code for this exaple
				Ι	Eclipse - Eclipse project for this example
				Ι	IAR - IAR project for this example
				Ι	CrossWorks- CrossWorks project for this example
				Ι	
				Ι	Many other examples same
I	•••				

2.5 Installing Eclipse

Start by downloading Eclipse IDE for C/C++ Developers here: https://www.eclipse.org/downloads/.

- 1. Start the Eclipse installer.
- 2. Select "Eclipse IDE for C/C++ Developers".



3. Select the install directory



eclipseins	taller by Comph	×
	DE for C/C++ Developers de	etails
Java 11+ VM	J.eclipse.org/justj/jres/15/updates/release/15.0.2 V	
Installation Folder	D:\i_syst\Development\eclipse\cpp-2021-03	
	✓ create start menu entry✓ create desktop shortcut	
	🕹 INSTALL	
& BACK		

4. Click "Install". Installation will start with a pop-up asking you to agree to the license. Accept and continue.

5. Now that is installed, start Eclipse and select where you want your workspace location.

Eclipse IDE Launcher		×
Select a directory as workspace		
Eclipse IDE uses the workspace directory to store its preferences and development artifacts.		
		_
Workspace: Vorkspace: D:\i_syst\Development\eclipse-workspace	Browse	
Use this as the default and do not ask again		
Launch	Cancel	
		_

6. Be patient, Eclipse is a bit slow to start. A welcome screen will show up. On the top right, select Open Workbench perspective. Select from the menu 'Help/Eclipse Marketplace...'. A pop-up will appear. Type 'arm' in the search box and install the 'GNU MCU Eclipse ...'. Again, say "yes" to all the licenses.



🖨 Eclipse Marl	ketplace —		\times
Eclipse Market	tplace	5	
	: to install. Press Install Now to proceed with installation. : info" link to learn more about a solution.	Ľ	5
Search Recent	Popular Favorites Installed 🖓 Giving IoT an Edge		
Find: P arm	× All Markets ~ All Categories	~	Go
	Eclipse Embedded C/C++ 6.1.2		^
GNU	The Eclipse Embedded CDT (C/C++ Development Tools, former MCU/ARM Eclipse) is an open source project that includes a fa Eclipse plug-ins and tools <u>more info</u>	-	
MCU	by <u>Liviu Ionescu</u> , EPL <u>GNU Arm Cortex-M RISC-V J-Link</u>		
* 172	Installs: 121K (3,985 last month)	Install	
	EmbSysRegView 0.2.6		
120	EMBedded SYStems REGister VIEW is an Eclipse Plugin which is for monitoring and modifying memory values of embedded de Therefore it offers a <u>more info</u>	-	
	by <u>Missing name Mising name</u> , EPL <u>EmbSysRegView memory memory view memory values memory</u> <u>monitoring</u>	<u>)ry</u>	
★ 12	Installs: 6.80K (91 last month)	Install	
	impulse Embedded Extension 2.1.4		
	This solution listing is an extension for impulse. The main impu- solution listing can be found here (with the option to install me extensions). impulse <u>more info</u>		
	by <u>toem GmbH</u> , Free for non-commerical use impulse Waveform <u>Plot signal analysis</u>		~
Marketpla	aces		
e 🔧	<u>, ()</u>		
?	< Back Install Now > Finish	Cancel	

7. Next step is to set the path to the toolchains. Open Eclipse preferences. For Linux & Windows, look in Help menu list. For OSX, prefs are in the usual place. A pop-up will appear. Find 'MCU' from the list on the left side and open it. Inside, set the path for both GCC and OpenOCD in the global section.



Preferences			
type filter text		Global Arm Toolchains Paths 🗢	▼ □> ▼ 8
> C/C++ ChangeLog	^	Configure the locations where various GNU Arm toolchains are installed. The values are stored within Eclipse. Unless rec specifically, they are used for all projects in all workspaces.	defined more
> Docker		Default toolchain: xPack GNU Arm Embedded GCC	
> Help			*
> Install/Update		Toolchain name: xPack GNU Arm Embedded GCC	
> Library Hover V MCU		Toolchain folder: D:\i_syst\Development\GNU Arm Embedded Toolchain\10 2020-q4-major Browse	xPack
Global Arm Toolchains Paths			
Global Build Tools Path			
Global OpenOCD Path			
Global pyOCD Path			
Global QEMU Path			
Global RISC-V Toolchains Paths Global SEGGER J-Link Path			
Workspace Arm Toolchains Paths			
Workspace Build Tools Path			
Workspace OpenOCD Path			
Workspace pyOCD Path			
Workspace QEMU Path			
Workspace RISC-V Toolchains Paths			
Workspace SEGGER J-Link Path			
> Mylyn			
> Oomph > Remote Development		Restore Defaults	Apply
Preferences			□ × •⇔• ŝ
type filter text	_	Global OpenOCD Path	• -> • 8
ChangeLog	^	Configure the location where xPack OpenOCD is installed. The values are stored within Eclipse. Unless redefined more sp	pecifically,
> Docker		they are used for all projects in all workspaces.	
> Help		After installing OpenOCD updates, restart Eclipse for the defaults to be re-evaluated and use the Restore Defaults button	to configure
> Install/Update		the new location.	
> Library Hover		Executable: openocd.exe	
 MCU Global Arm Toolchains Paths 		Folder: C:/Users/TAIHM/AppData/Roaming/xPacks/openocd/0.11.0-1/bin Browse	xPack
Global Build Tools Path			
Global OpenOCD Path			
Global pyOCD Path			
Global QEMU Path			
Global RISC-V Toolchains Paths			
Global SEGGER J-Link Path Workspace Arm Toolchains Paths			
Workspace Build Tools Path			
Workspace OpenOCD Path			
Workspace pyOCD Path			
Workspace QEMU Path			
Workspace RISC-V Toolchains Paths			
Workspace SEGGER J-Link Path			
> Mylyn > Oomph			
 Compn Remote Development 	~	Restore Defaults	Apply
? 🔤 🗹 🔘		Apply and Close	Cancel

That is all that's needed for Eclipse and toolchain installations. This Eclipse installation is not limited to Nordic based development. It is a generic installation that allows you to work with any ARM Cortex MCU from any vendor. It works for RISC-V as well. You will need to install toolchains for RISC-V if you want to work with that in Eclipse.

Important notes for OSX users

Since the Catalina update, there is a new security measure that blocks the execution of command line tools such as the GCC compiler and OpenOCD and other downloaded executables. First thing,



open System Preferences/Security & Privacy/Privacy. Select 'Developer Tools'. Then add Eclipse to the list.

Now that Eclipse and all the toolchains are fully installed, lets start compiling. Select menu 'File/Open Projects from File System...'.

		avigate Search Project									
	New	Alt+Shift+N >	h Configurations	✓ on:			- Y 🌣	📑 🗕 🗌		🏷 🔻 📮	•
~	Open File		Q 😕 📁 🛷	- R I T 2 -		****		4		Q .	9 6
	Open Projects from File Syste Recent Files	m	0			₽0 🛛 🔭			(2) Welcom		
		>					5	Moneth w			
	Close Editor	Ctrl+W				There is no acti	- •	Contents			~
	Close All Editors	Ctrl+Shift+W				that provides a	n outline.	Related 1			
	Save	Ctrl+S							opics up bo	OKMARKS	
	Save As										
ß	Save All	Ctrl+Shift+S						Search ex	pression:		
	Revert									`	G
	Move							Scope De	fault		
P	Rename	F2									
8	Refresh	F5									
	Convert Line Delimiters To	>									
Ð	Print	Ctrl+P									
2n	Import										
4	Export										
	Properties	Alt+Enter									
	Switch Workspace	>									
	Restart										
	Exit										
		🖹 Problems 🛛 🧔 Tas 0 items	ks 📃 Console 🔲 P	roperties		78					
		Description	^	Resource	Path	I	Location				
		<					>				

A pop-up will open. Click on the "Directory" button, navigate to, and select the 'nRF52832' folder in the IOsanota/ARM/Nordic/nRF52/ location. Eclipse will search and list all projects available within that folder. Deselect the first checkbox 'nRF52832' and keep all the others. For the BLYST840, use the 'nRF52840' instead.



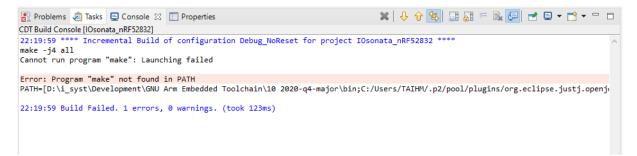
Import Projects from File System or Archive		– 🗆 X
Import Projects from File System or Archive		
This wizard analyzes the content of your folder or archive file to find projects and import the	iem in the IDE.	
Import source: D:\i_syst\Development\IOsonata\ARM\Nordic\nRF52\nRF52840		V Directory Archive
type filter text		Select All
Folder	Import as	Deselect All
✓ nRF52840 ✓ nRF52840\exemples\AdcDemo\Eclipse ✓ nRF52840\exemples\AnalogCompDemo\Eclipse ✓ nRF52840\exemples\Blackdvertiser\Eclipse ✓ lose newly imported projects upon completion Use installed project configurators to: ✓ Search for nested projects ✓ Detect and configure project natures	Eclipse project Eclipse project Eclipse project Eclipse project	39 of 39 selected ✓ ☐ Hide already open projects
Working sets Add project to working sets Working sets:		New Select Show other specialized import wizards
?	< Back Next >	Finish Cancel

Click 'Finish'. Eclipse will load all projects into the project explorer on the left pane. Select & rightclick on the 'IOsonata_nRF52832' project. Then select 'Build Configuration/Build All' to build all variants of the IOsonata library for the nRF52832.



le Edit So	urce	Refactor Navigate Search	Project Run Wind	dow Help						
s ()		~	No Launch Configurat	ions 🗸 🗸	on:			× 🔅	📑 🗝 🗐 👘 👻 🗸	≪ - 🗟 📮
• × &		New	> -	🥭 🗀 🔗 🛨 🗟	π : £] - *> d≯ ¢		-	۹ 🖻
Project Expl		Go Into					₽0 🛛 🔭		👘 Help 🔀 🚯 Welcon	ne 🗆 [
		Open in New Window						5° 8		
💕 dfu_uart		Show In	Alt+Shift+W >				There is no active		🗎 Contents 🔗 Searc	h
> 🗊 Inclu		Show in Local Terminal	>				that provides an	outline.	📽 Related Topics 🛄 B	ookmarks
> 🛵 src 🕰 dfu_uart		Сору	Ctrl+C						🛗 Index	
💕 Eeproml	Ē	Paste	Ctrl+V						Search expression:	
💕 IOsonata	×	Delete	Delete							~ G
> 🔊 Inclu > 🚰 inclu		Source	>						Scope Default	
> 🔁 src		Move							v Scope benduit	
🗃 nRF5284		Rename	F2							
> 🔂 exem > 📂 Idscri	2	Import								
> 🔂 lib	4	Export								
-		Build Project								
		Clean Project								
	8	Refresh	F5							
		Close Project								
		Close Unrelated Projects								
		Build Targets	>							
		Index	>							
		Build Configurations	>	Set Active	>					
		Profiling Tools	>	Manage			7 8			
	0	Run As	>	Build All						
	*	Debug As	>	Clean All		Path		Location		
		Profile As	>	Build Selected						
		Restore from Local History	Γ							
	*	Run C/C++ Code Analysis								
lOsonata_nl		Team	>					>		

You may encounter the following failure



If you are using GNU MCU Eclipse on Windows, make sure <u>Windows Build Tools</u> are installed, then check the installation path and fill the "Global Build Tools Path" inside Eclipse Window/Preferences... :



Preferences		$\Box \rightarrow$
type filter text	Global Build Tools Path	;
> General	The locations where various Eclipse Embedded CDT build tools are installed. Unless defined more specifically, they are used for all projects in all workspa	CAS
> C/C++		
ChangeLog	Build tools folder: C:\Users\AppData\Roaming\xPacks\windows-build-tools\xpack-windows-build-tools-4.2.1-2\bin Browse	xPack
Docker		
Help		
Install/Update		
Library Hover		
MCU		
Global Arm Toolchains Paths		
Global Build Tools Path		
Global OpenOCD Path		
Global pyOCD Path		
Global QEMU Path		
Global RISC-V Toolchains Paths		
Global SEGGER J-Link Path		
Workspace Arm Toolchains Paths		
Workspace Build Tools Path		
Workspace OpenOCD Path		
Workspace pyOCD Path		
Workspace QEMU Path		
Workspace RISC-V Toolchains Paths		
Workspace SEGGER J-Link Path		
Mylyn	Restore Defaults	Anal
Oomph	Restore Defaults	Apply
? 🖻 🗹 🖲	Apply and Close	Cancel

It will take a while to compile all the libraries. There is a lot of source code. Look at the bottom pane in the 'Console' tab for the compilation results.



Once the library compilations are complete, you can build any example project listed. To start, let's build the Blinky example. Select the Blinky project to highlight it. Find the hammer in the middle of the toolbar and click on it to build the highlighted project.



🐔 🚺 🔳 🗸 Ni	lo Launch Configurations 🗸 on: 🚥 🗸 😵 🗸 🔅 😴 🗸 🔞 🗸 🔞 🗸 🔞 🗸 🔞 🗸 🔞 🗸 🔞 🗸 🔞 🗸 🔞 🗸 🔞 🗸 🔞 🗸 🔞	* * * • • • 8
·····································		
Project Explorer 🛛 📄 🐄 🍸 🖇 🗖 🗖	ic) blinky.c	- 8
AdcDemo (in Eclipse)	71 #detime LED3 PIN BLUEIO LED3 PIN	_
AnalogCompDemo (in Eclipse)	72 #define LED3_PINOP BLUEI0_LED3_PINOP	
BleAdvertiser (in Eclipse)	73	
Blinky (in Eclipse)	74 #define LED4_PORT BLUEIO_LED4_PORT 75 #define LED4 PIN BLUEIO LED4 PIN	
> & Binaries	76 #define Lev4_rin BLUEID_LEV4_rin 76 #define LEV4_PIND BLUEID_LEV4_FINDP	
> 🔊 Includes		
> 🗁 Debug	780#define BUTTON_PINS_MAP {\	
> > Release	79 {BUT1_PORT, BUT1_PIN, BUT1_PINOP, IOPINDIR_INPUT, IOPINRES_PULLDOWN, IOPINTYPE_NORMAL}, \	
> > Release_mbr	<pre>80 {BUT2_PORT, BUT2_PIN, BUT2_PINOP, IOPINDIR_INPUT, IOPINRES_PULLDOWN, IOPINTYPE_NORMAL}, \ 81 }</pre>	
	01 J 82	
✓ math binky.c	830 #define LED PINS MAP { \	
> 🙀 blinky.c > 🙀 board.h	84 {LED1_PORT, LED1_PIN, LED1_PINOP, IOPINDIR_OUTPUT, IOPINRES_NONE, IOPINTYPE_NORMAL}, \	
BluelOThingy (in Eclipse)	<pre>85 {LED2_PORT, LED2_PIN, LED2_PINOP, IOPINDIR_OUTPUT, IOPINRES_NONE, IOPINTYPE_NORMAL}, \</pre>	
Cli (in Eclipse)	<pre>86 {LED3_PORT, LED3_PIN, LED3_PINOP, IOPINDIR_OUTPUT, IOPINRES_NONE, IOPINTYPE_NORMAL}, \ 87 {LED4 PORT, LED4 PIN, LED4 PINOP, IOPINDIR OUTPUT, IOPINRES NONE, IOPINTYPE NORMAL}, \</pre>	
Gruble (in Eclipse)	8/ {LEU4_PORT, LEU4_PIN, LEU4_PINOP, IOPINDIK_OUTPUT, IOPINTRES_NONE, IOPINTYPE_NORMAL}, \	
🚰 dfu_usb (in Eclipse)	89	
dfu_usb_open (in Eclipse)	900 #define PULSE_TRAIN_PINS_MAP { \	
ElnkDiisplayDemo (in Eclipse)	91 {0, 2, 0}, {0, 3, 0}, {0, 4, 0}, {0, 5, 0}, {0, 6, 0}, {0, 7, 0}, \	
ESBprx (in Eclipse)	92 {0, 8, 0}, {0, 9, 0}, {0, 10, 0}, {0, 11, 0}, {0, 12, 0}, {0, 13, 0}, {0, 14, 0}, {0, 15, 0}, \ 93 {0, 16, 0}, {0, 17, 0}, {0, 18, 0}, {0, 19, 0}, {0, 20, 0}, {0, 21, 0}, {0, 22, 0}, {0, 23, 0}, \	
ESBptx (in Eclipse)	$\begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} $	
FlashMemoryDemo (in Eclipse)	95 {1, 0, 0}, {1, 1, 0}, {1, 2, 0}, {1, 3, 0}, {1, 4, 0}, {1, 5, 0}, {1, 6, 0}, {1, 7, 0}, \	
I2CMasterSlave (in Eclipse)	96 {1, 8, 0}, {1, 9, 0}, {1, 10, 0}, {1, 11, 0}, {1, 12, 0}, {1, 13, 0}, {1, 14, 0}, {1, 15, 0}, \	
IOsonata_nRF52840 (in Eclipse)	97 }	
LedDemo (in Eclipse)	98 99 #endif //BOARD_H	
MeshLgtSwClient (in Eclipse)	100	~
MeshLqtSwServer (in Eclipse)	<	>
MotionSensorDemo (in Eclipse)	👔 Problems 🧔 Tasks 📮 Console 🔅 🔲 Properties 📲 Include Browser	×
PulseTrain (in Eclipse)		~
PwmDemo (in Eclipse)	CDT Build Console (Blinky) arm-none-easi-gcc -mcpu=cortex-m4 -mtnumo -mtioat-aoi=nara -mtpu=tpv4-sp-aio -us -tmessage-iengtn=0 -tsigned-cnar -ttunction-sec	TIONS -TOATA-SECT
💕 radio_test (in Eclipse)	Finished building: D:/i_syst/Development/IOsonata/exemples/misc/blinky.c	
SPIMasterSlave (in Eclipse)		
FimerDemo (in Eclipse)	Building target: Blinky.elf Invoking: Cross ARM C Linker	
💕 TPHDemo (in Eclipse)	Invoking: cross ARM C Linker amm-none-eabi-gcc =mcpu=cortex-m4 -mthumb -mfloat-abi=hard -mfpu=fpv4-sp-d16 -Os -fmessage-length=0 -fsigned-char -ffunction-sec	tions -fdata-sect
FTPHSensorTag (in Eclipse)	Finished building target: Blinky.elf	
removersoriad (in collpse)		
🚰 UartBleBridge (in Eclipse)	Invoking: Cross ARM GNU Create Flash Image	
🚰 UartBleBridge (in Eclipse) 🚰 UartBleCentralDemo (in Eclipse)	arm-none-eabi-objcopy -0 ihex "Blinky.elf" "Blinky.hex"	
 UartBleBridge (in Eclipse) UartBleCentralDemo (in Eclipse) UartBleDemo (in Eclipse) 		
 UartBleBridge (in Eclipse) UartBleCentralDemo (in Eclipse) UartBleDemo (in Eclipse) UartBleDemo (in Eclipse) UartBleFreeRTOS (in Eclipse) 	arm-none-eabi-objcopy -0 ihex "Blinky.elf" "Blinky.hex"	
UartBleBridge (in Eclipse) UartBleCentralDemo (in Eclipse) UartBleDemo (in Eclipse) UartBleDemo (in Eclipse) UartBleFreeRTOS (in Eclipse) UartBrFreeRTOS (in Eclipse)	arm-none-eabi-objcopy -O inex "Blinky.elf" "Blinky.hex" Finished building: Blinky.hex Invoking: Cross ARM GNU Print Size arm-none-eabi-sizeformat-berkeley "Blinky.elf"	
UartBleBridge (in Eclipse) UartBleCentralDemo (in Eclipse) UartBleDemo (in Eclipse) UartBleFreeRTOS (in Eclipse) UartInRFLogTest (in Eclipse) UartInSRATert (in Eclipse)	arm-none-eabi-objcopy -0 ihex "Plinky.elf" "Blinky.hex" Finished building: Blinky.hex Invoking: Cross ARM GNU Print Size arm-none-eabi-sizeformat-berkeley "Blinky.elf" text data bss dec hex filename	
Gurthersteiner (in Eclipse)	arm-none-eabi-objcopy -0 inex "Blinky.elf" "Blinky.hex" Finished building: Blinky.hex Invoking: Cross ARM GNU Print Size arm-none-eabi-sizeformat-berkeley "Blinky.elf" text data bss dec hex filename 2668 768 136 3572 df4 Blinky.elf	
SurtBleBridge (in Eclipse) SurtBleCentralDemo (in Eclipse) SurtBleCentralDemo (in Eclipse) SurtBleFreeRTOS (in Eclipse) SurtBleFreeRTOS (in Eclipse) SurtProskVTett (in Eclipse) SurtProskVTett (in Eclipse) SurtProskVTett (in Eclipse)	arm-none-eabi-objcopy -0 ihex "Plinky.elf" "Blinky.hex" Finished building: Blinky.hex Invoking: Cross ARM GNU Print Size arm-none-eabi-sizeformat-berkeley "Blinky.elf" text data bss dec hex filename	
UatBleBridge (in Eclipse) UatBleCentralDemo (in Eclipse) UatBleGeno (in Eclipse) UatBleFreeRTOS (in Eclipse) UatBleFreeRTOS (in Eclipse) UathRFLogTest (in Eclipse)	arm-nonē-eabi-objcoy -0 ihex "Plinky.elf" "Blinky.hex" Finished building: Blinky.hex Invoking: Cross ARH GMD Print Size arm-none-eabi-sizeformat-berkeley "Blinky.elf" text data bis dec hex filename 2668 768 136 3572 df4 Blinky.elf Finished building: Blinky.siz	
 UartBleBridge (in Eclipse) UartBleCentralDemo (in Eclipse) UartBleGene (in Eclipse) UartBleFreeRTOS (in Eclipse) UartBleFreeRTOS (in Eclipse) UarthrbsTret (in Eclipse) 	arm-none-eabi-objcopy -0 inex "Blinky.elf" "Blinky.hex" Finished building: Blinky.hex Invoking: Cross ARM GNU Print Size arm-none-eabi-sizeformat-berkeley "Blinky.elf" text data bss dec hex filename 2668 768 136 3572 df4 Blinky.elf	
UatBileEndge (in Eclipse) UatBileEndraftDemo (in Eclipse) UatBileCentralDemo (in Eclipse) UatBileTextRTDS (in Eclipse) UathRietOrate (in Eclipse) UathRietOrate (in Eclipse) UathProStNett (in Eclipse) UathProStNett (in Eclipse) UathSdkPhoElet (in Eclipse)	arm-nonē-eabi-objcoy -0 ihex "Plinky.elf" "Blinky.hex" Finished building: Blinky.hex Invoking: Cross ARH GMD Print Size arm-none-eabi-sizeformat-berkeley "Blinky.elf" text data bis dec hex filename 2668 768 136 3572 df4 Blinky.elf Finished building: Blinky.siz	
UartBleRidge (in Eclipse) UartBleRidge (in Eclipse) UartBleCentralDemo (in Eclipse) UartBleTeeRTOS (in Eclipse) UartBleFeeRTOS (in Eclipse) UartBleFeeRTOS (in Eclipse) UartBrbsRvTest (in Eclipse) UartBletstertemo (in Eclipse)	arm-nonē-eabi-objcoy -0 ihex "Plinky.elf" "Blinky.hex" Finished building: Blinky.hex Invoking: Cross ARH GMD Print Size arm-none-eabi-sizeformat-berkeley "Blinky.elf" text data bis dec hex filename 2668 768 136 3572 df4 Blinky.elf Finished building: Blinky.siz	