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

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

Contents

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
GNU Toolchain | GNU Arm Embedded Toolchain Downloads – Arm Developer
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
 |   |...
 |   |-- nRF5_SDK_12     - Last version of Nordick SDK12 for nRF51 series
 |   |   |   |-- components
 |   |   |   |-- examples
 |   |...
 |   |-- 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
```

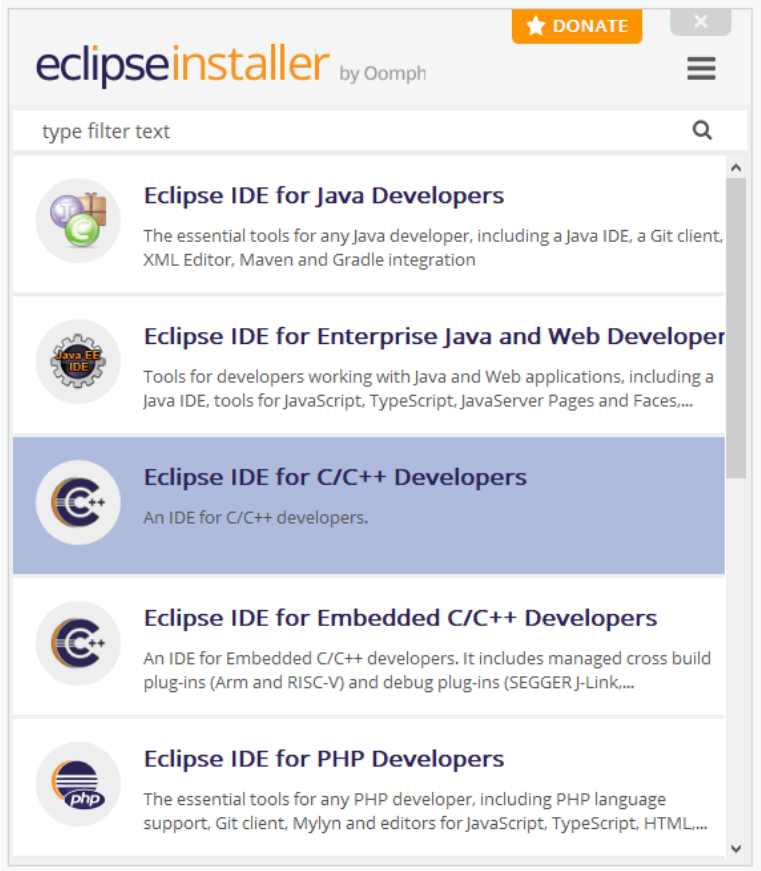
```
|   |   |-- converters  - Generic definition for ADV, DAC, etc...
|   |   |-- coredev     - Generic definition MCU builtin devices such as i2c, uart, spi, timer, etc...
|   |   |-- miscdev     - Generic definition for other non categorized devices
|   |   |-- sensors     - Generic definition for al sort of sensors (environmental, motion, etc...)
|   |   |-- usb         - Generic definition for USB
|   |   |...
|   |-- src         - Generic implementation source common to all platforms
|   |   |-- 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.
|   |   |-- ldscript    - Linker script files
|   |   |
|   |   |-- 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
|   |   |   |   |   |-- lib        - IOsonata library for this target
|   |   |   |   |   |   |-- 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
|   |   |   |   |   |   |   |-- 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
|   |   |   |   |   |   |...
|   |   |   |   |   |
```

```
|   |   |   |   |   |-- exemples   - Example projects for this target
|   |   |   |   |   |   |-- Blink     - Blink example
|   |   |   |   |   |   |   |-- 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
| ...
```

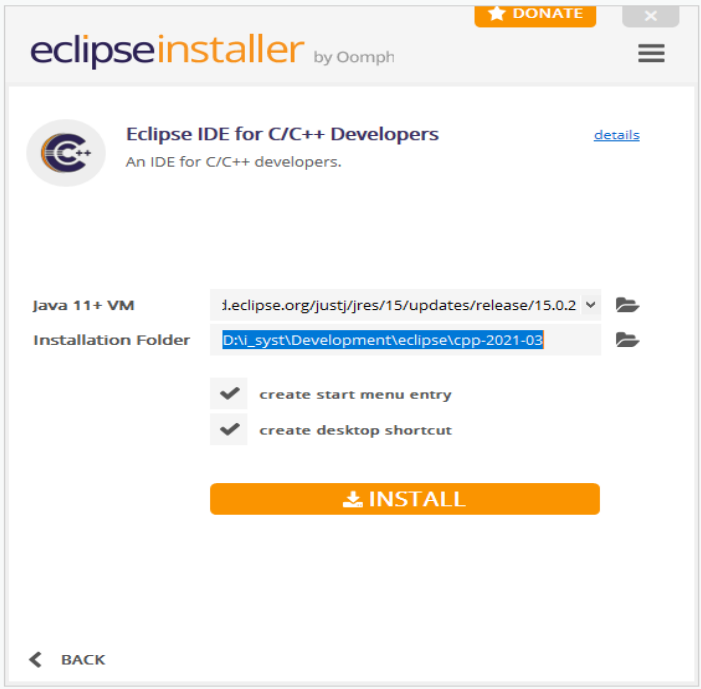
2.5 Installing Eclipse

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".

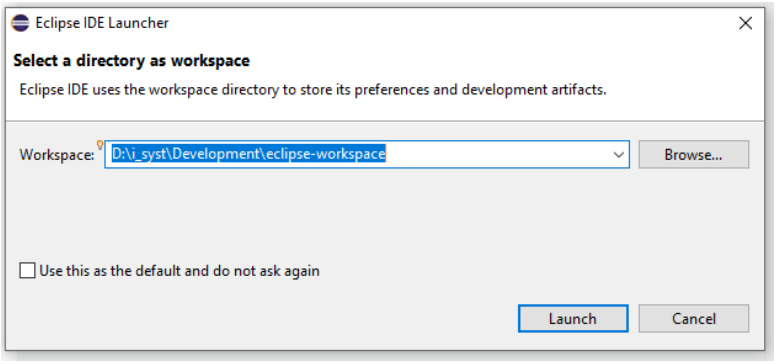


3. Select the install directory



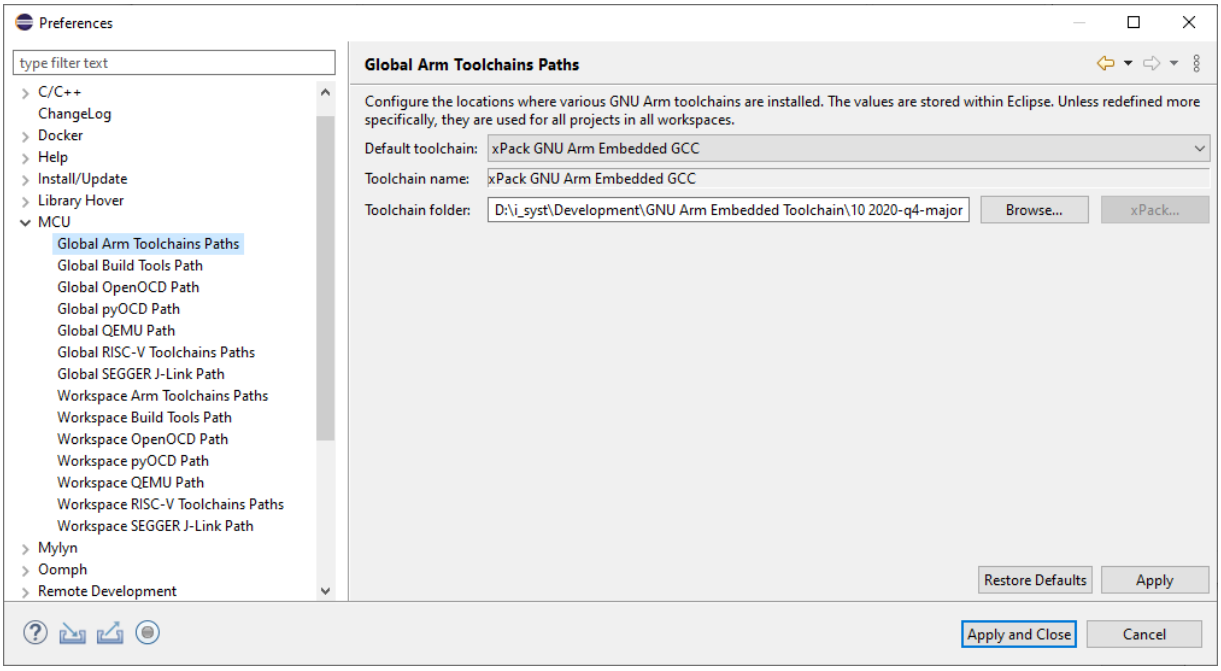
4. Click "Install". Installation will start with a pop-up asking you to agree to the license. Accept and continue.

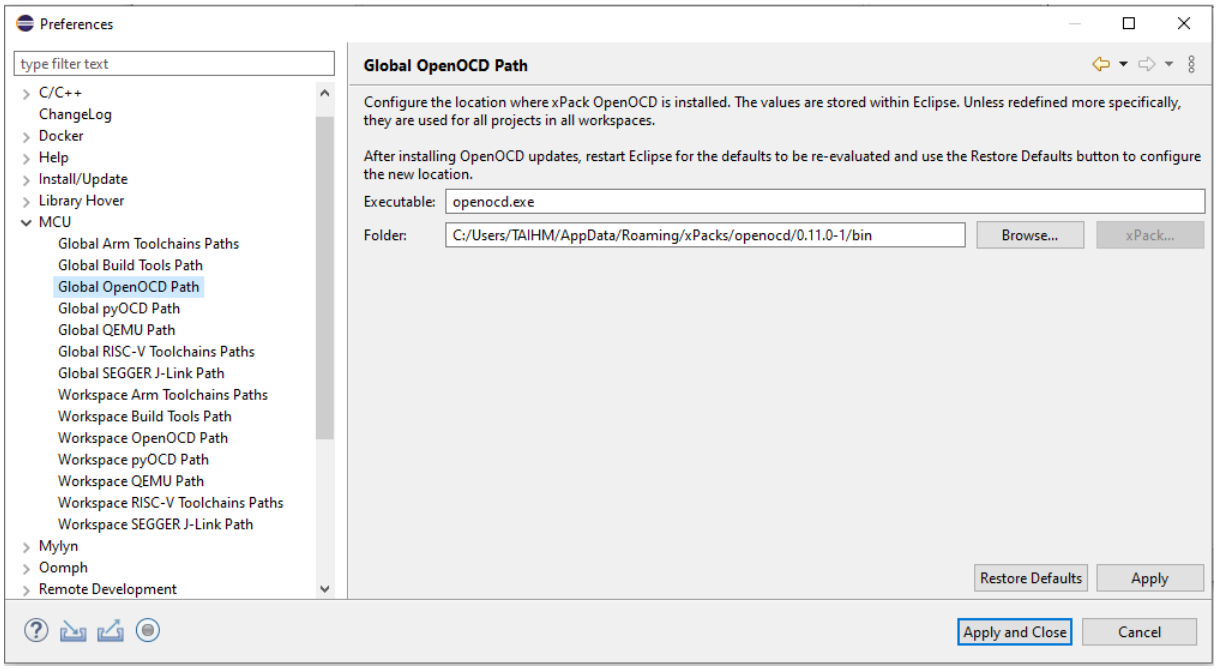
5. Now that is installed, start Eclipse and select where you want your workspace location.



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.

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.



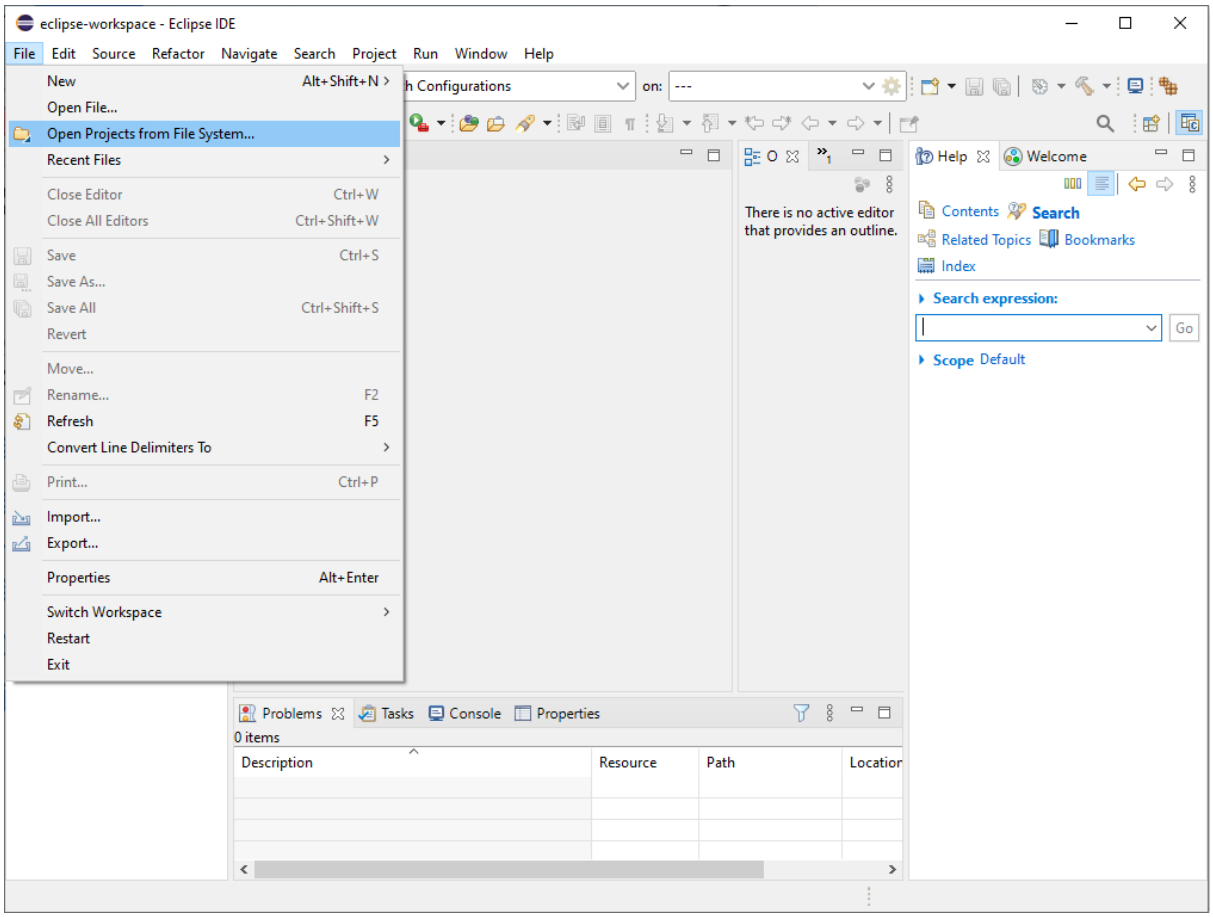


That is all that's needed for Eclipse and toolchain installations. This Eclipse installation is not limited to Nordic based development. It is a generic installation that allows you to work with any ARM Cortex MCU from any vendor. It works for RISC-V as well. You will need to install toolchains for RISC-V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

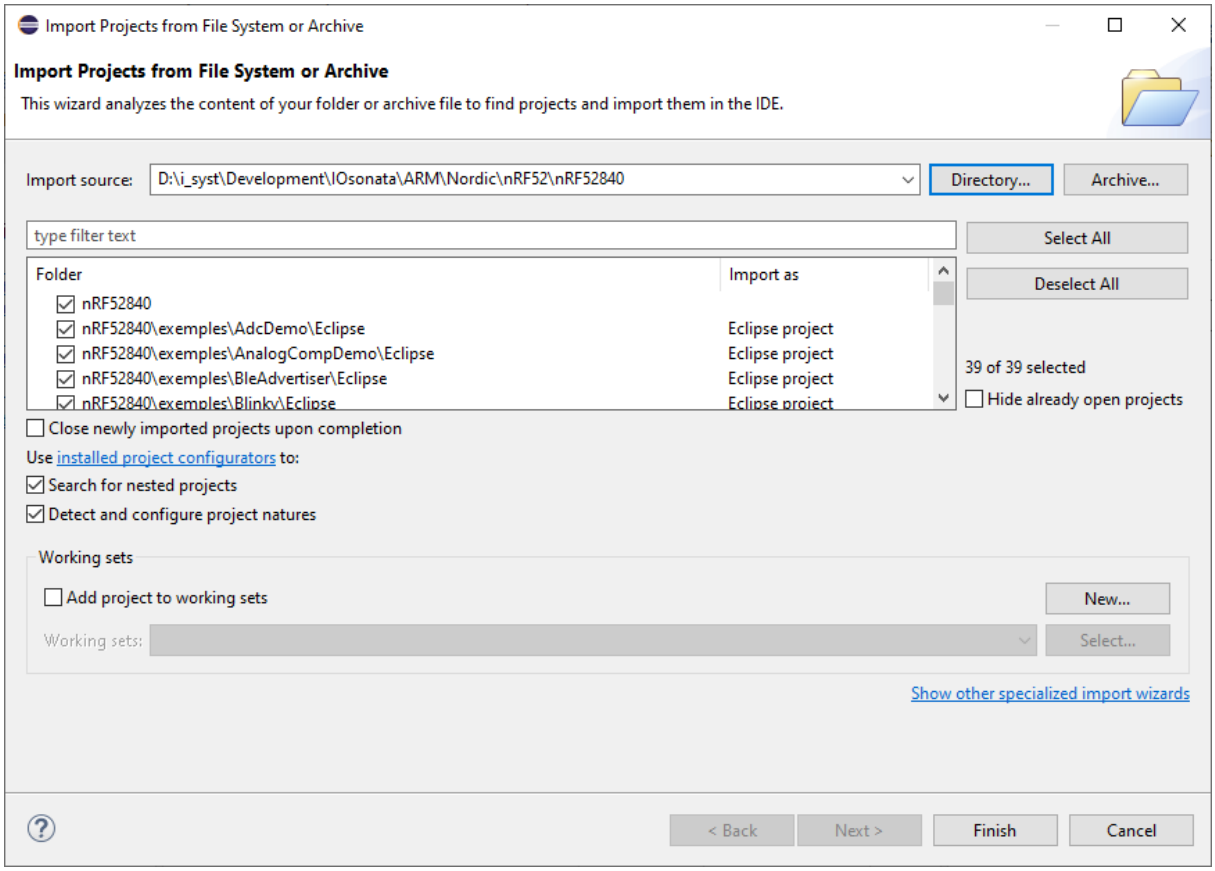
Important notes for OSX users

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.

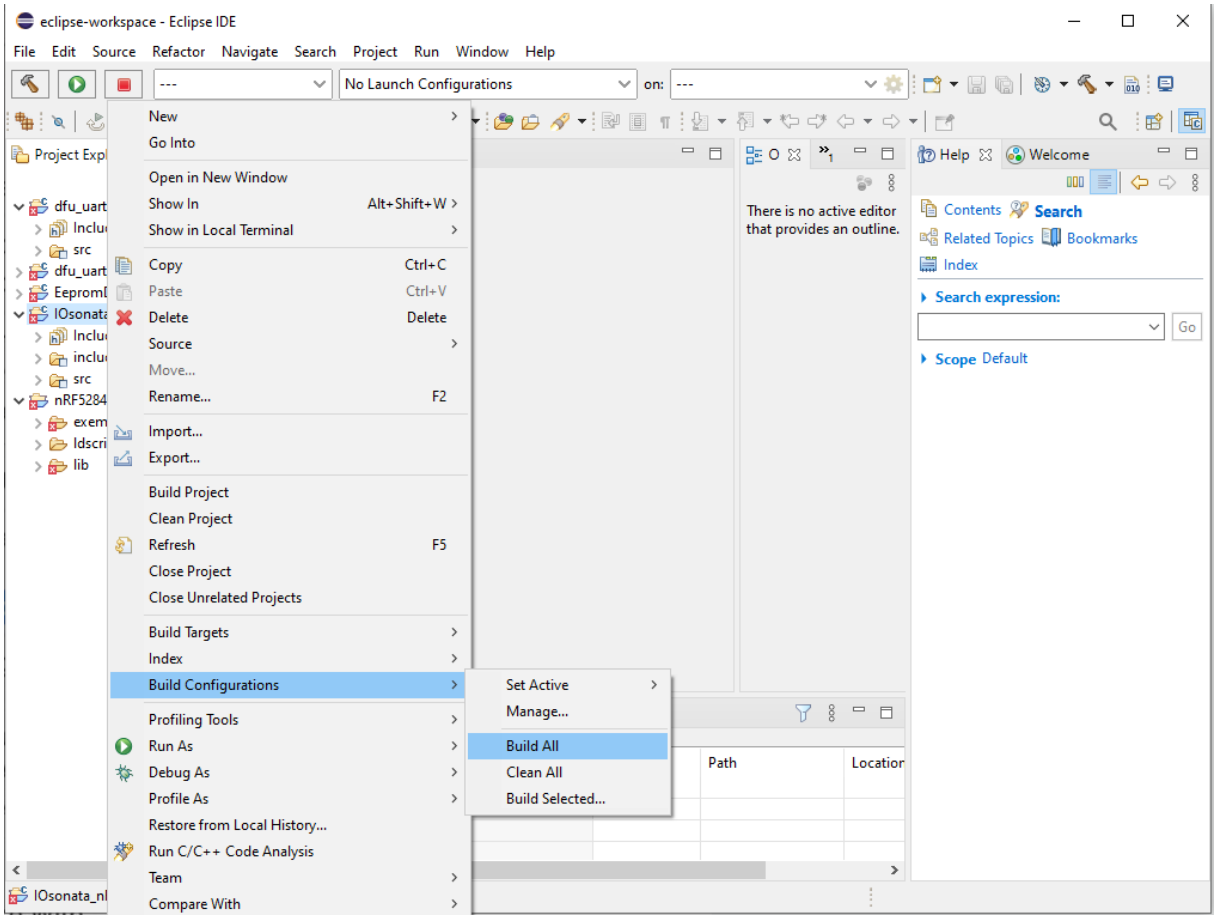
Now that Eclipse and all the toolchains are fully installed, lets start compiling. Select menu ‘File/Open Projects from File System…’.



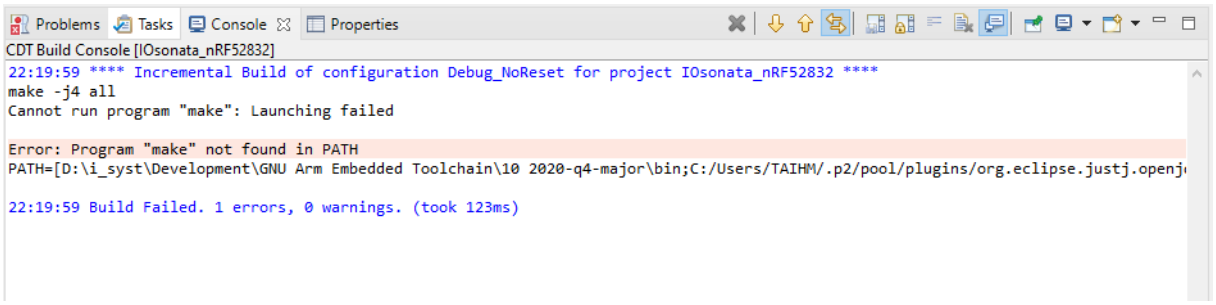
A pop-up will open. Click on the "Directory" button, navigate to, and select the ‘nRF52832’ folder in the IOsanota/ARM/Nordic/nRF52/ location. Eclipse will search and list all projects available within that folder. Deselect the first checkbox ‘nRF52832’ and keep all the others. For the BLYST840, use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



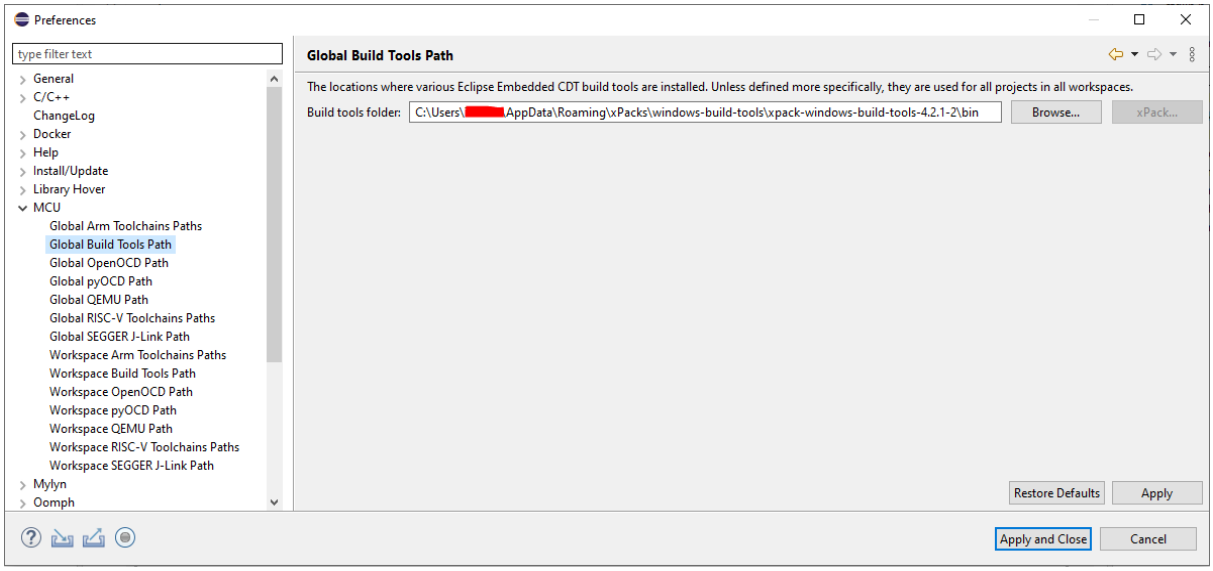
Click ‘Finish’. Eclipse will load all projects into the project explorer on the left pane. Select & right-click on the ‘IOsonata_nRF52832’ project. Then select ‘Build Configuration/Build All’ to build all variants of the IOsonata library for the nRF52832.



You may encounter the following failure



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



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.