

OPTIGA™ Trust M

Product Version: V1

About this document

Scope and purpose

The purpose of this document is to guide a beginner to use the OPTIGA™ Trust M XMC4800 IoT Connectivity kit. The scope is limited to OPTIGA™ Trust M XMC4800 IoT Connectivity kit and its hardware and software components.

Intended audience

This document addresses: customers, solution providers and system integrators.

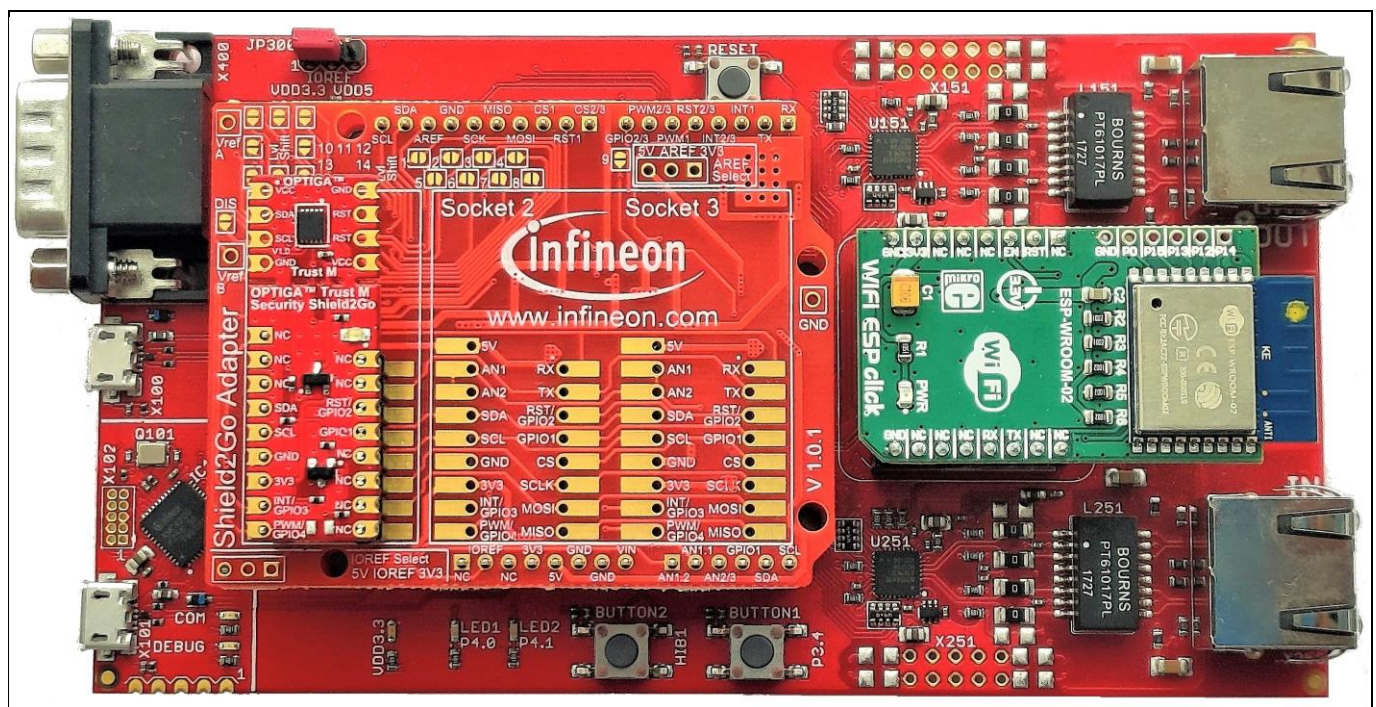


Table of Contents

About this document.....	1
Table of Contents	2
1 Introduction	3
1.1 References	3
1.2 Abbreviations	3
2 OPTIGA™ Trust M	5
2.1 OPTIGA™ Trust M XMC4800 IoT Connectivity kit	5
2.1.1 XMC4800 IoT Connectivity kit Components	5
2.2 Installed Software Components	6
3 System Setup.....	7
3.1 System Overview	7
3.2 Hardware Setup.....	8
3.2.1 XMC4800 IoT Connectivity Kit	8
3.2.2 My IoT Adapter	9
3.2.3 Shield2Go Security OPTIGA™ Trust M.....	10
3.3 Software Setup.....	10
3.3.1 Software Components	10
3.3.1.1 XMC4800 IoT Connectivity Kit.....	10
3.3.2 PC Requirements and Configurations.....	11
3.3.2.1 PC Requirement	11
4 Using OPTIGA™ Trust M	12
4.1 Quick Setup	12
4.1.1 Running OPTIGA™ Trust M Example Application	12
4.1.2 Steps to download example hex file to XMC4800 Connectivity IoT Kit.....	12
4.1.2.1 Using JFlashLite tool.....	12
4.1.3 Logger.....	14
4.1.3.1 Logger setup.....	14
4.1.3.2 Logger control	16
4.2 Advanced Setup	17
4.2.1 Setting up DAVE™ IDE on PC	17
4.2.2 Running OPTIGA™ Trust M Example Application Project with DAVE™	18
5 Troubleshooting	22
Revision History	23

Introduction

1 Introduction

This document describes how to setup the environment to run OPTIGA™ Trust M application and use the provided binaries.

1.1 References

Table 1 **References**

Definition	Source
[1] xmc4800_IOTkit_usermanual	Infineon
[2] Infineon_I2C_Protocol	Infineon

1.2 Abbreviations

Table 2 **Abbreviations**

Abbreviation	Definition
API	Application Programming Interface
CA	Certificate Authority
CHM	Microsoft Compiled HTML Help
CMOS	Complementary Metal Oxide Semiconductor
DAVE	Digital Application Virtual Engineer
ECC	Elliptic Curve Cryptography
HTML	Hyper Text Markup Language
HW	Hardware
I2C	Inter Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
NIST	National Institute of Standards and Technology
OS	Operating System
PAL	Platform Abstraction Layer
PC	Personal Computer
RST	Reset
SCL	Serial Clock
SDA	Serial Data
SW	Software
TTL	Transistor Transistor Logic
USB	Universal Serial Bus
XMC	XMC4800 IoT Amazon FreeRTOS

Introduction

Abbreviation	Definition
	Connectivity Kit WIFI with EtherCAT Kit-V1.1

2 OPTIGA™ Trust M

OPTIGA™ Trust M is a security solution with a pre-programmed security controller built on Elliptic Curve Cryptography (ECC) with 256 and 384 bit curve length, RSA PKCS V1.5, SHA-256.

It supports secure data object update, hibernate and toolbox functionalities, which is used for secure communication, platform integrity, secure updates, data store protection and lifecycle management for Connected Device Security.

2.1 OPTIGA™ Trust M XMC4800 IoT Connectivity kit

OPTIGA™ Trust M XMC4800 IoT Connectivity kit is designed to provide all the components required to setup the environment to demonstrate the features of the OPTIGA™ Trust M.

2.1.1 XMC4800 IoT Connectivity kit Components

Table 3 XMC4800 IoT Connectivity kit contents

No.	Item	Description
1	XMC4800 board	Hardware Evaluation board for XMC4800 microcontroller from Infineon. More details can be found on Infineon website .
2	WIFI ESP Click	WIFI hardware module, which can be integrated with XMC4800 microcontroller.
3	Micro USB to USB cable	The cable provides DC supply to XMC4800 IoT Connectivity Kit and to flash software.

2.2 Installed Software Components

The installed directory structure of OPTIGA™ Trust M setup software is shown below:

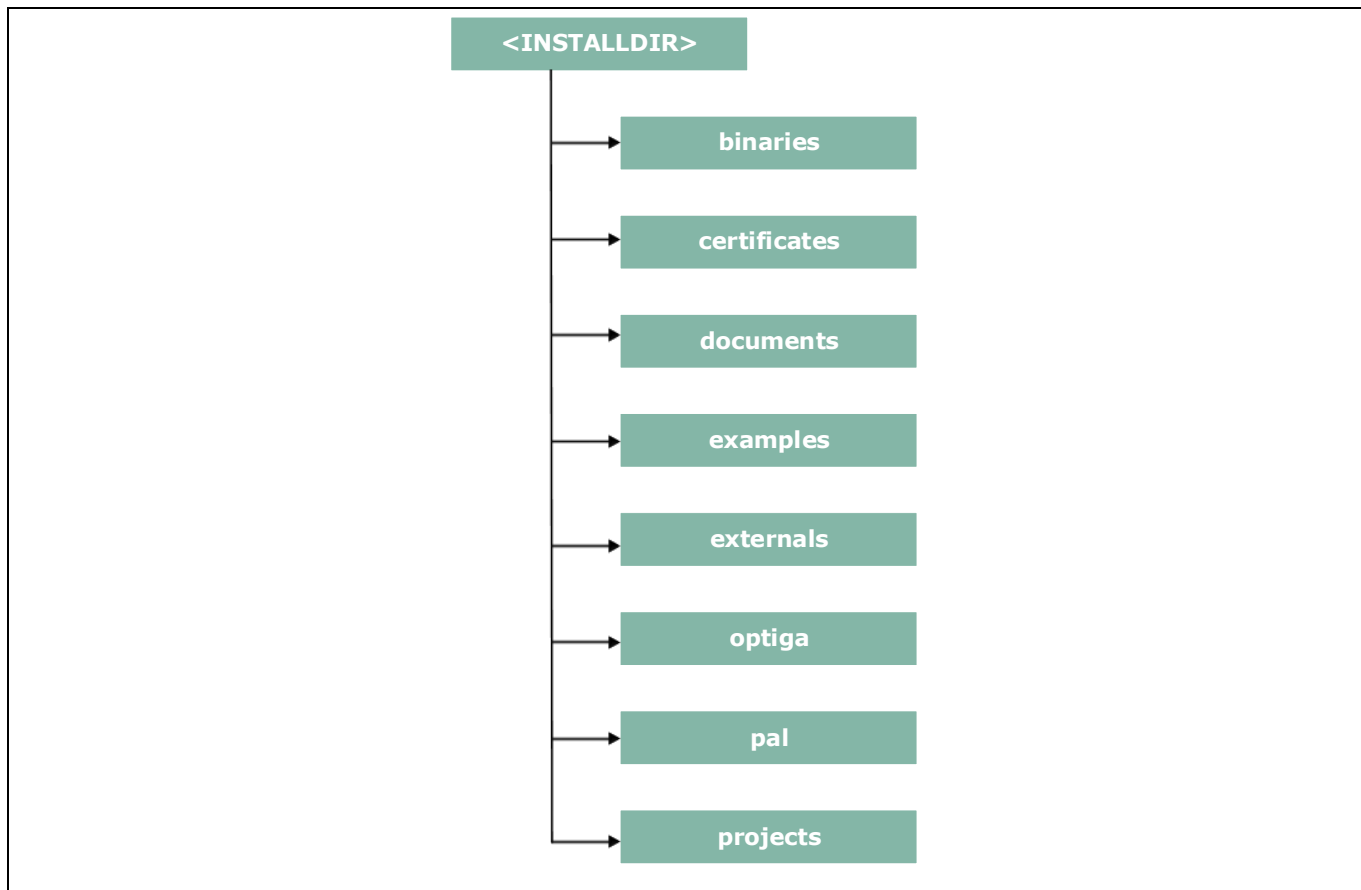


Figure 1 Installed directory structure

<INSTALLDIR> is the root directory to which the release package contents are extracted. The following section explains the contents of each subdirectory under installed directory:

1. **binaries** -- binaries for OPTIGA™ Trust M example application.
2. **certificates** -- OPTIGA™ Trust M Test CA and Productive CA certificates.
3. **documents** -- Relevant OPTIGA™ Trust M documentation.
4. **examples** -- Example usecases for Toolbox features and a tool for generation of manifest for secure data object feature.
5. **externals** -- mbedtls software crypto library.
6. **optiga** -- OPTIGA™ Trust M libraries.
7. **pal** -- PAL for XMC4800 device and PAL for mbedtls software crypto library.
8. **projects** -- XMC4800 device example project in DAVE workspace.

System Setup

3 System Setup

This section explains the basic components required for system setup.

3.1 System Overview

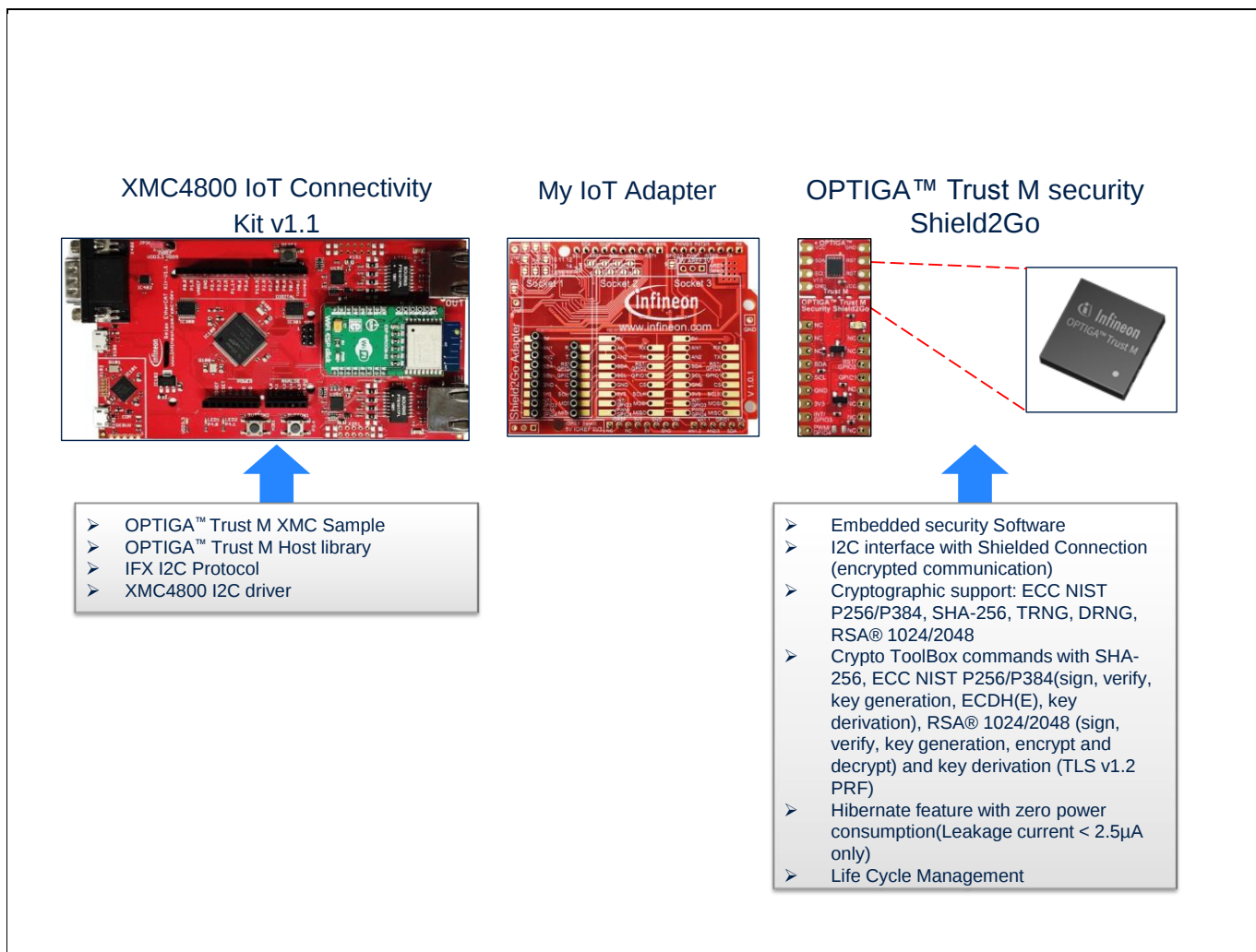


Figure 2 System Overview

This system consists of the following components:

1. XMC4800 IoT Connectivity Kit v1.0 from Infineon
 - The XMC4800 IoT Connectivity Kit is an evaluation board with XMC4800 Microcontroller from Infineon. For more information refer document [\[1\]](#).
 - It can connect to a WiFi access point using WiFi ESP click module.
 - It is used as a reference platform to simulate the Host.
 - It interacts with secure element via I2C.
2. My IoT Adapter
 - It acts as a gateway to add Shield2Go boards onto XMC4800 IoT Connectivity Kit V1.0.
3. OPTIGA™ Trust M Security Shield2Go

System Setup

- Shield2Go board contains OPTIGA™ Trust M chip. It is compatible with Arduino Uno along with Infineon's My IoT adapter.

The following interface/connection is done among the above components:

- Micro USB data cable (with Data line) from PC is connected to XMC to supply power.

3.2 Hardware Setup

The hardware required to run OPTIGA™ Trust M setup is described in this section.

3.2.1 XMC4800 IoT Connectivity Kit

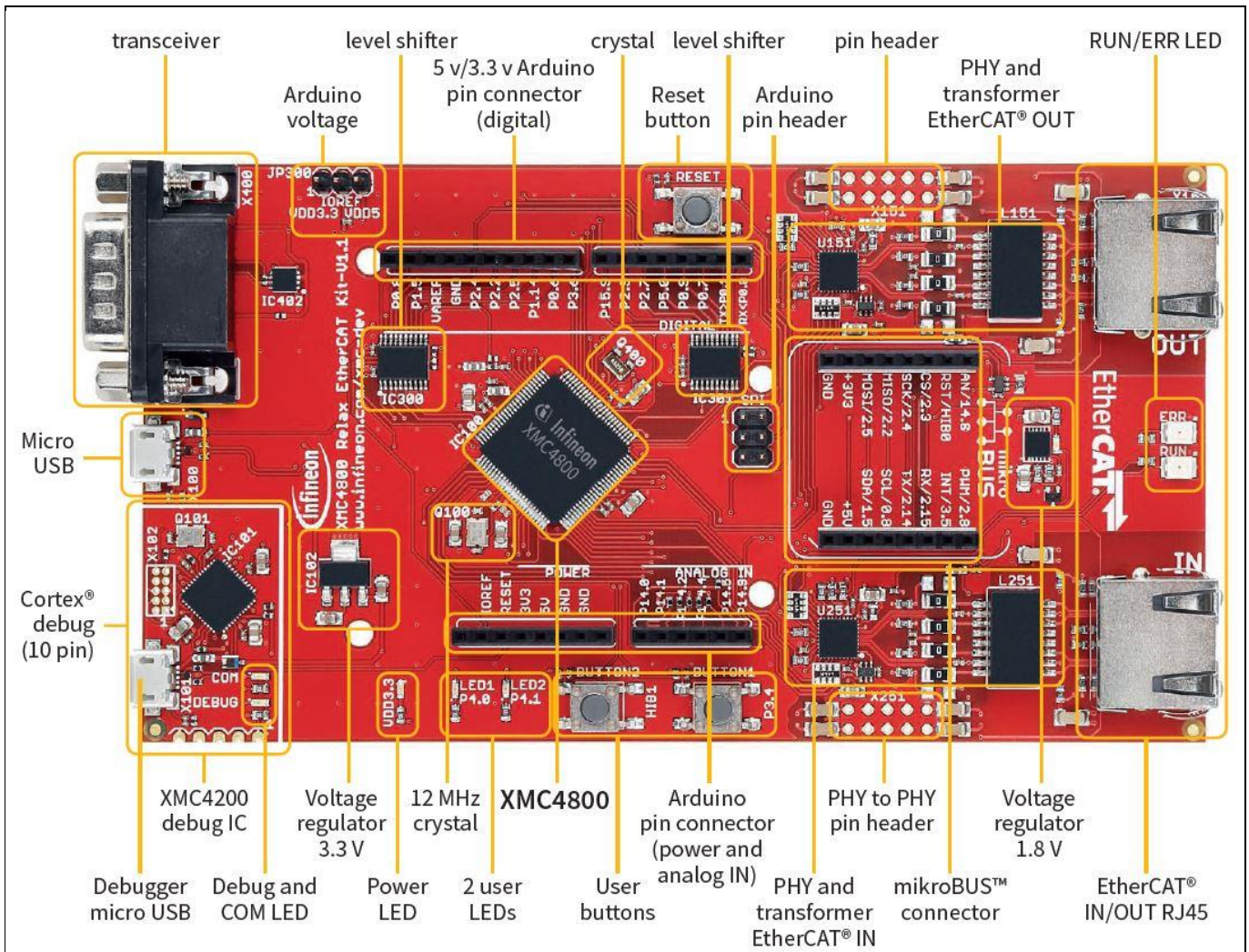


Figure 3 XMC4800 IoT Connectivity Kit

Table 4 XMC4800 IoT Connectivity kit Components

No.	Item	Description
1	DC Supply	Power supply of 5V is provided by connecting to Micro USB connector.
2	Arduino compatible connector	External interface to connect to Arduino Shields.
3	mikroBUS socket	Socket to connect to WiFi ESP click module from MikroElektronika.
4	On-board debug probe	Supports Serial Wire Debug and UART communication for debugging and logging purposes.

System Setup

The pin headers for Arduino shields can be used for GPIOs or signal interface as well. Arduino compatible connector supports I2C, UART and SPI interfaces among others.

Table 5 XMC4800 IoT Connectivity kit I2C Pin Information

No.	Description	Pin
1	I2C SCL	P0.8
2	I2C SDA	P1.5
3	RST	P0.6

For more information about pin details of Arduino shield, refer document [\[1\]](#).

For more information about the XMC Specification, Architecture and Design/Schematic, refer document [\[1\]](#)

3.2.2 My IoT Adapter

The My IoT adapter is an evaluation board that allows users to easily combine different Shield2Go boards to Arduino compliant ecosystem, for fast evaluation of IoT systems. With its solderless connectors, it allows users to easily stack Shield2Go boards instead of soldering it. The shield design is derived from XMC2Go evaluation board.

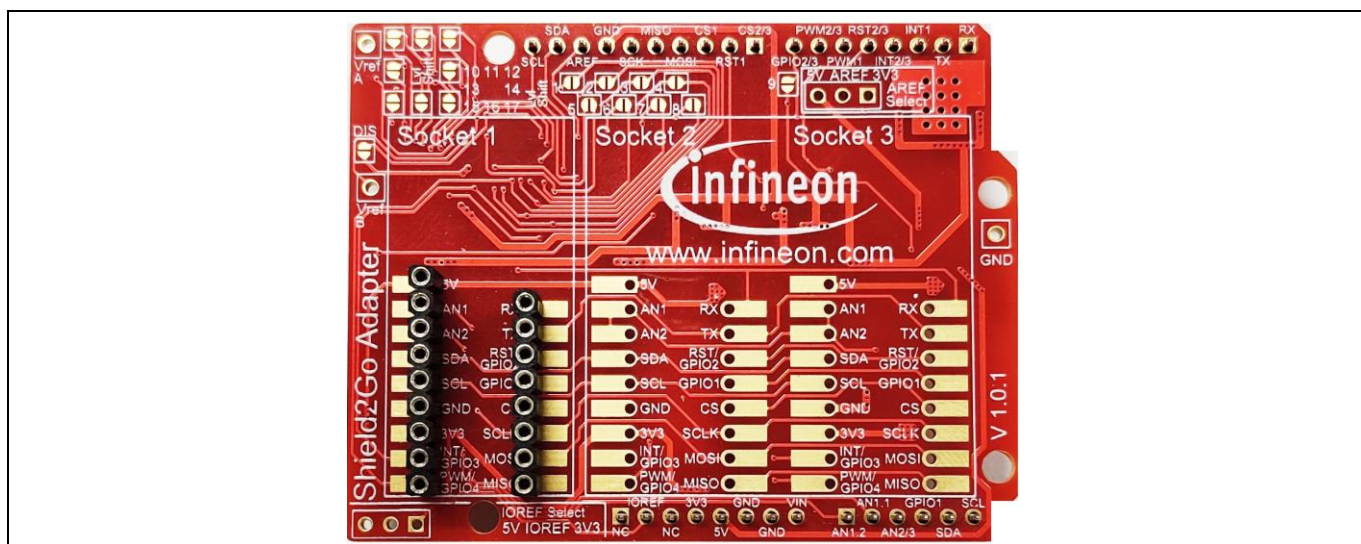


Figure 4 My IoT adapter

My IoT adapter features are as follows:

- Provide power supply and connectivity for Shield2Go boards.
- Level shifting handling capabilities between CMOS 3.3V and TTL 5V.
 - Solder bridges to selectively deactivate level shifting.
 - Additional pins enable setting the reference voltages for level shifting.
- Separate power control switches for Socket 1 and Socket 2. Socket 1 is independently controllable while Socket 2 and 3 share pins to underlying control board.

More information is available at [Infineon website](#).

System Setup

3.2.3 Shield2Go Security OPTIGA™ Trust M

Shield2Go boards are equipped with featured Infineon ICs and provide a standardized form factor and pin layout, allowing a ‘plug and play’ approach for easy prototyping.

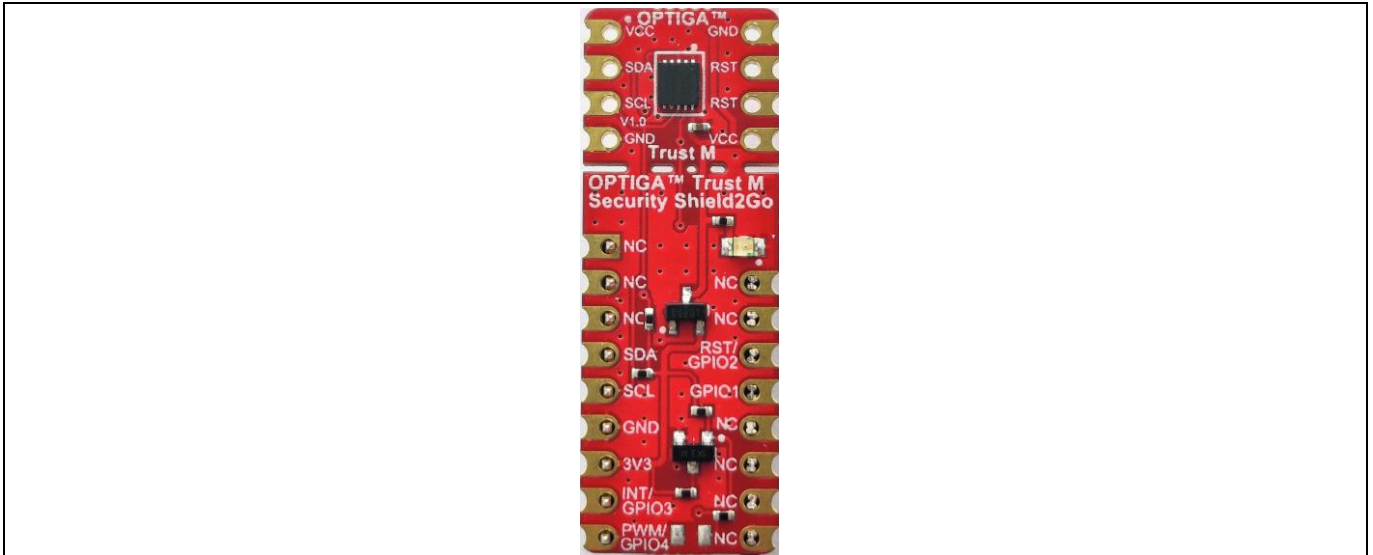


Figure 5 OPTIGA™ Trust M Shield2Go

The OPTIGA™ Trust M Shield2Go is equipped with OPTIGA™ Trust M security chip. It allows users to develop system solutions by combining Shield2Go with My IoT adapter and XMC.

Note: Ensure no voltage supplied to any of the pins exceeds the absolute maximum rating of $V_{cc} + 0.3\text{ V}$.

3.3 Software Setup

This section describes the software used in XMC to run the OPTIGA™ Trust M setup.

3.3.1 Software Components

All the software components required on XMC are explained in the following sections.

3.3.1.1 XMC4800 IoT Connectivity Kit

1. OPTIGA™ Trust M Host Library consists of the following:

- Service Layer
The layers (Util and Crypt) provide APIs to interact with OPTIGA™ for various use-case functionalities.
- Access Layer
This layer manages the access to the command interface of OPTIGA™ security chip. It also provides the communication interface to the OPTIGA™.
- Platform Abstraction Layer
This layer provides platform agnostic interfaces for the underlying HW and SW platform functionalities used by OPTIGA™ libraries.
- Platform Layer
This layer provides the platform specific components and libraries for the supported platforms.

System Setup

2. I2C Protocol

This is an implementation as per document [\[2\]](#).

3. XMC4800 I2C Driver

These are low level I2C device driver for I2C communication from XMC to OPTIGA™ Trust M Security chip.

4. OPTIGA™ Trust M XMC Example

This Example Application demonstrates Secure Data Object, Hibernate feature, Cryptographic ToolBox Functionalities and Read/Write General Purpose Data use cases.

Note: The binaries and the example application provided with the application note are meant for the XMC4800 IoT ConnectivityKit v1. These binaries may not work as expected if executed on a different platform.

3.3.2 PC Requirements and Configurations

3.3.2.1 PC Requirement

A 32-bit or 64-bit PC with Windows 7/10 Operating System with the below requirements need to be used for setting up the OPTIGA™ Trust M setup:

1. One USB port.
2. DAVE 4.4.2 and device feature 2.2.4, which can be downloaded from Infineon website.

Link to download DAVE 4.4.2: [Dave Download](#)

3. Segger J-Link tool v6.00 or greater for flashing software on XMC.

Link to download Segger: [J-Link tool Download](#)

Link to download manual: [J-Link manual Download](#)

Note: The path where DAVE tool is extracted is henceforth referred to as <DAVE_PATH> in the document.

Note: All the tools mentioned in the above list are intended to be used with the binaries or source code given in the release package.

Using OPTIGA™ Trust M

4 Using OPTIGA™ Trust M

4.1 Quick Setup

This section explains the steps to run OPTIGA™ Trust M example application.

4.1.1 Running OPTIGA™ Trust M Example Application

1. Make the connections among XMC4800 IoT Connectivity Kit, My IoT Adapter and OPTIGA™ Trust M Shield2Go as shown below

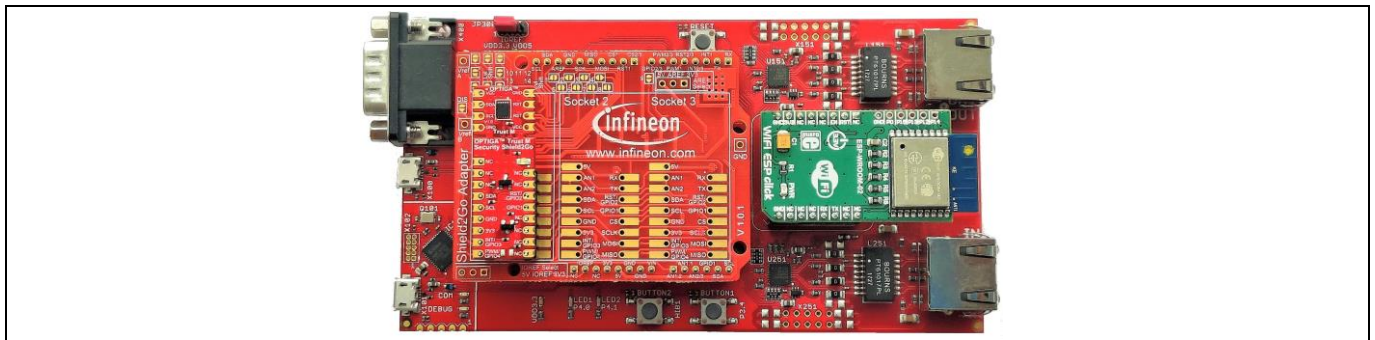


Figure 6 XMC4800 IoT Connectivity Kit, My IoT Adapter and OPTIGA™ Trust M Shield2Go connection

2. Power up the kit by connecting Micro USB cable between PC and Debugger micro USB. For placement of Debugger micro USB refer Figure 3.
3. Download the OPTIGA™ Trust M example application using JFlashLite tool as described in section 4.1.2.1. Hex file location is <INSTALLDIR>\binaries\xmc4800_iot_kit\dave4\xmc4800_optiga_example.hex.
4. OPTIGA™ Trust M example application uses USBD_VCOM for logging, refer section 4.1.3 for logging details.

4.1.2 Steps to download example hex file to XMC4800 Connectivity IoT Kit

4.1.2.1 Using JFlashLite tool

1. Run JFlashLite.exe from JLink installation folder. It shows a notice window. Click OK.

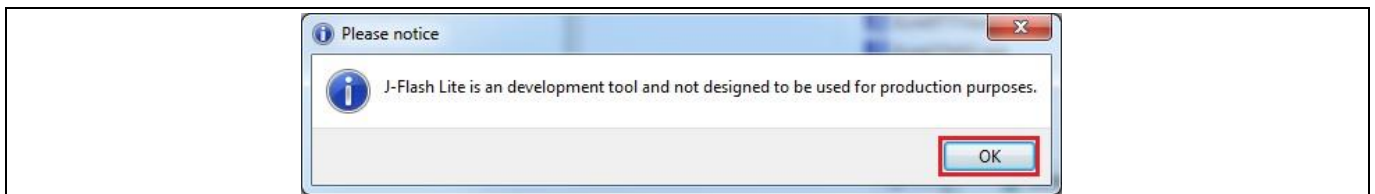


Figure 7 JFlashLite launch window

2. Click on Device to select a target device.

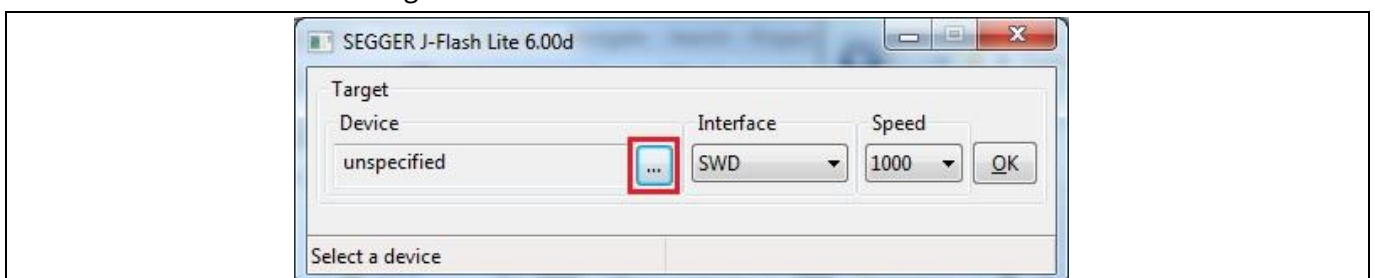


Figure 8 JFlashLite select a device

3. Select Infineon as Manufacturer and Device as XMC4800-2048, and then click OK.

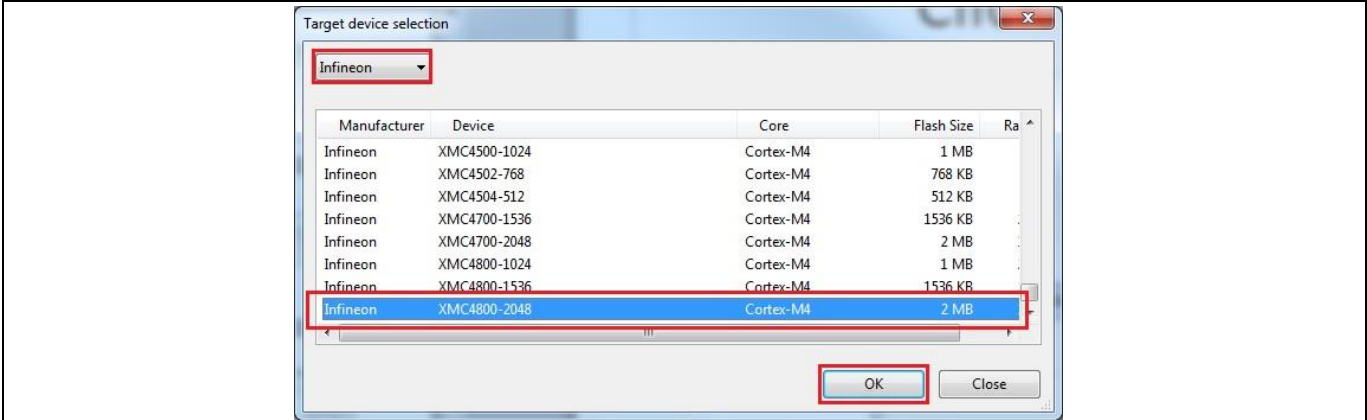


Figure 9 JFlashLite Target device selection

4. After target device selection, click OK on window shown in Figure 9.
5. Select hex file to be flashed under Data File and click on Program Device. It then shows the programming progress window.

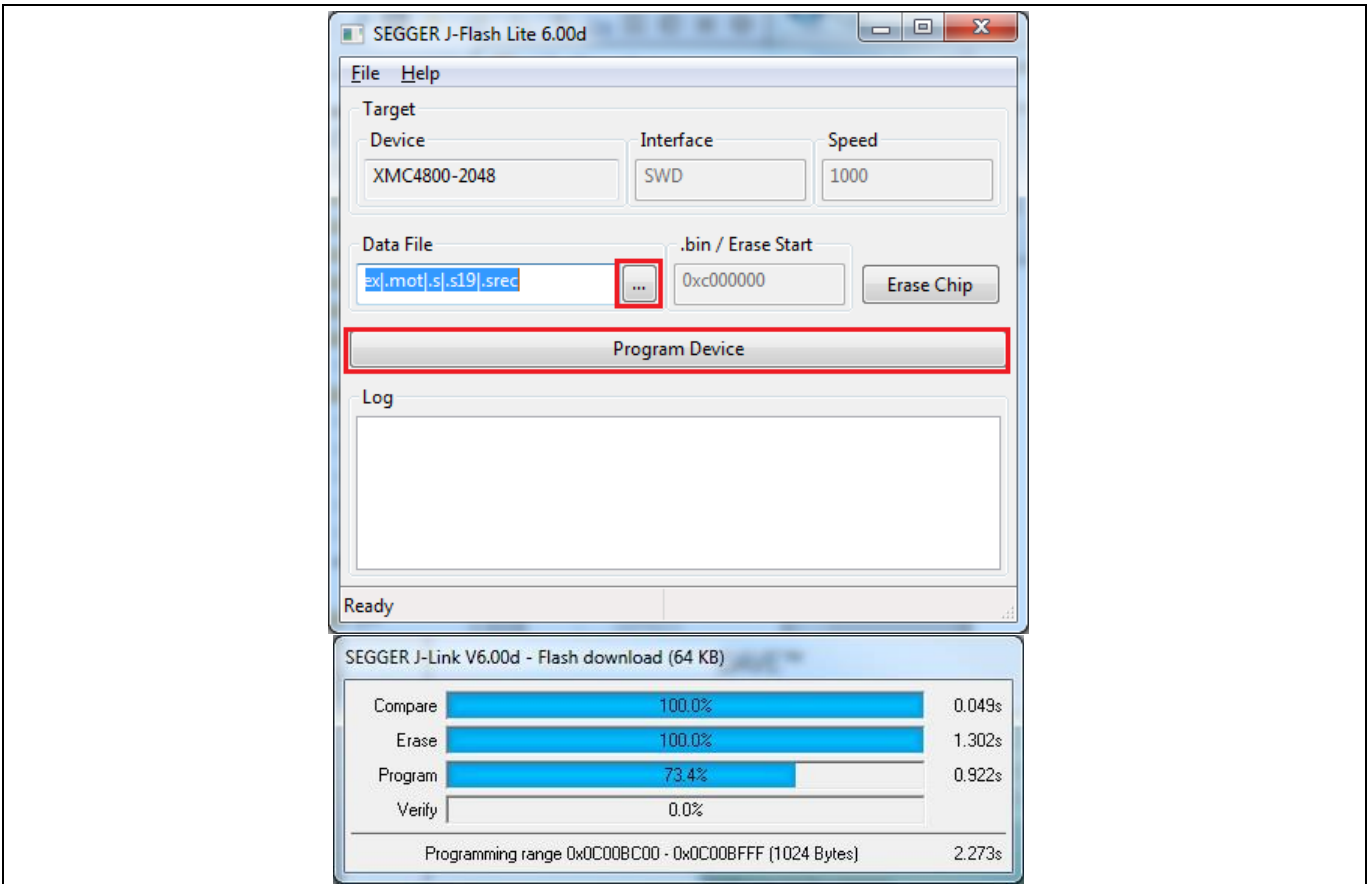


Figure 10 JFlashLite Hex file selection and programming progress window

6. Flash download completed.

Using OPTIGA™ Trust M

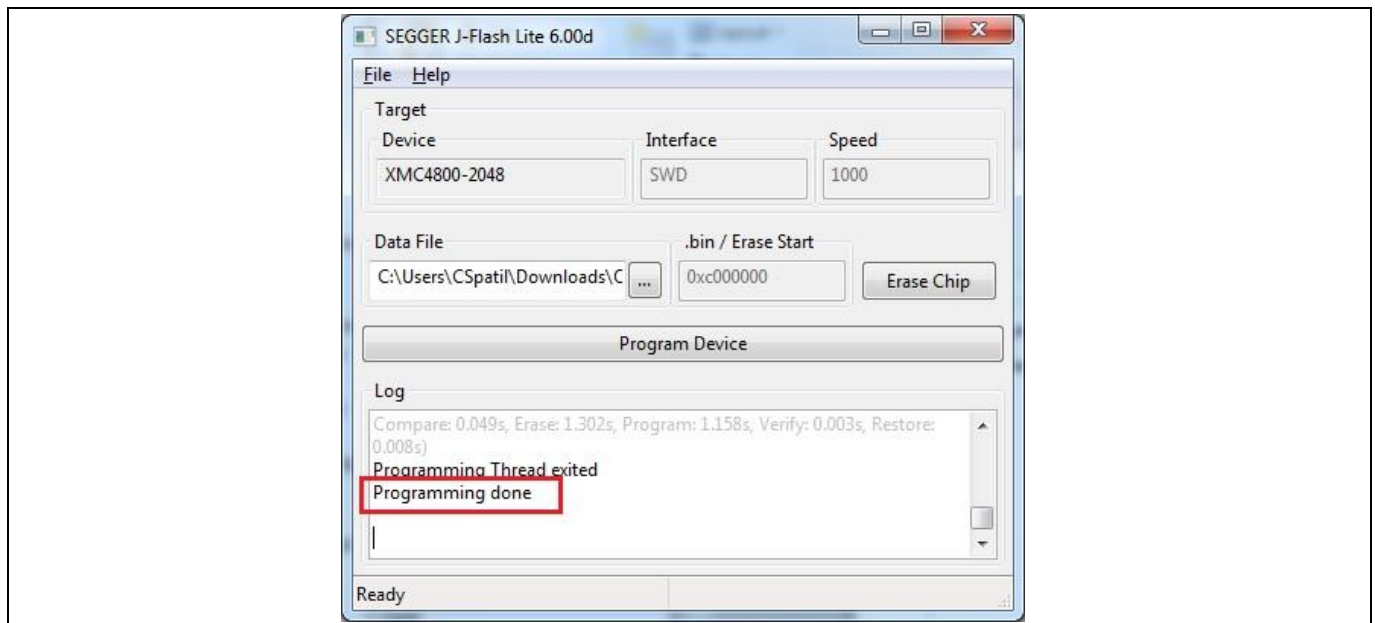


Figure 11 JFlashLite programming completion window

4.1.3 Logger

4.1.3.1 Logger setup

1. Connect the micro USB cable between PC and micro USB. For placement of micro USB refer Figure 3.
2. Reset the XMC4800 by pressing the reset button.
3. Select the COM port with name "Communications Port".

Note: For binding the Windows serial driver(usbser.sys) with USBD_VCOM device user has to point to the driver.inf file in the folder path:
 <INSTALLDIR>\projects\xmc4800_iot_kit\common\Dave\Generated\USBD_VCOM\inf\

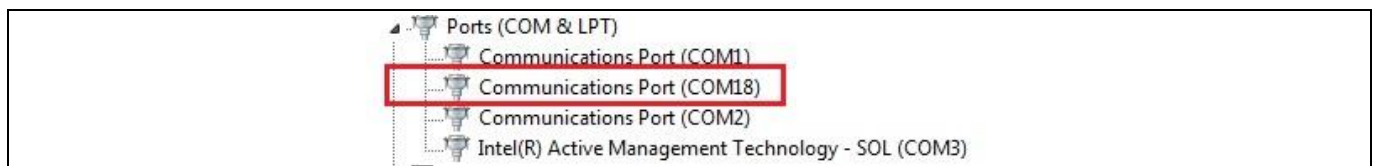


Figure 12 Discovery of USB Serial Device COM port

4. Configure COM port with 9600 8N1.

Using OPTIGA™ Trust M

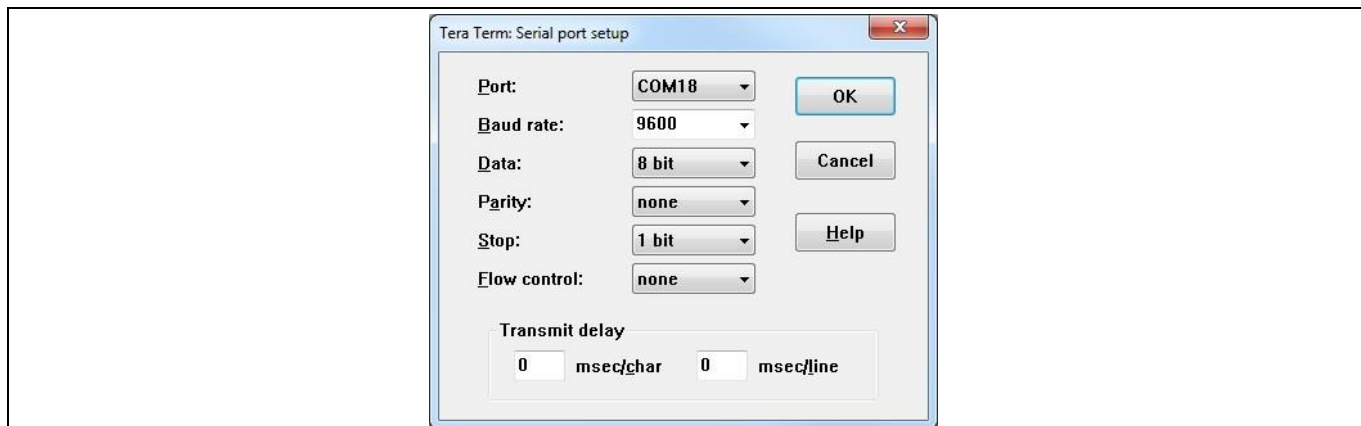


Figure 13 TeraTerm terminal serial configuration

5. Once connected, the terminal displays the text “Press any key to start example demonstration”.
6. The logs of the example execution are displayed along with status of each example as Passed or Failed.

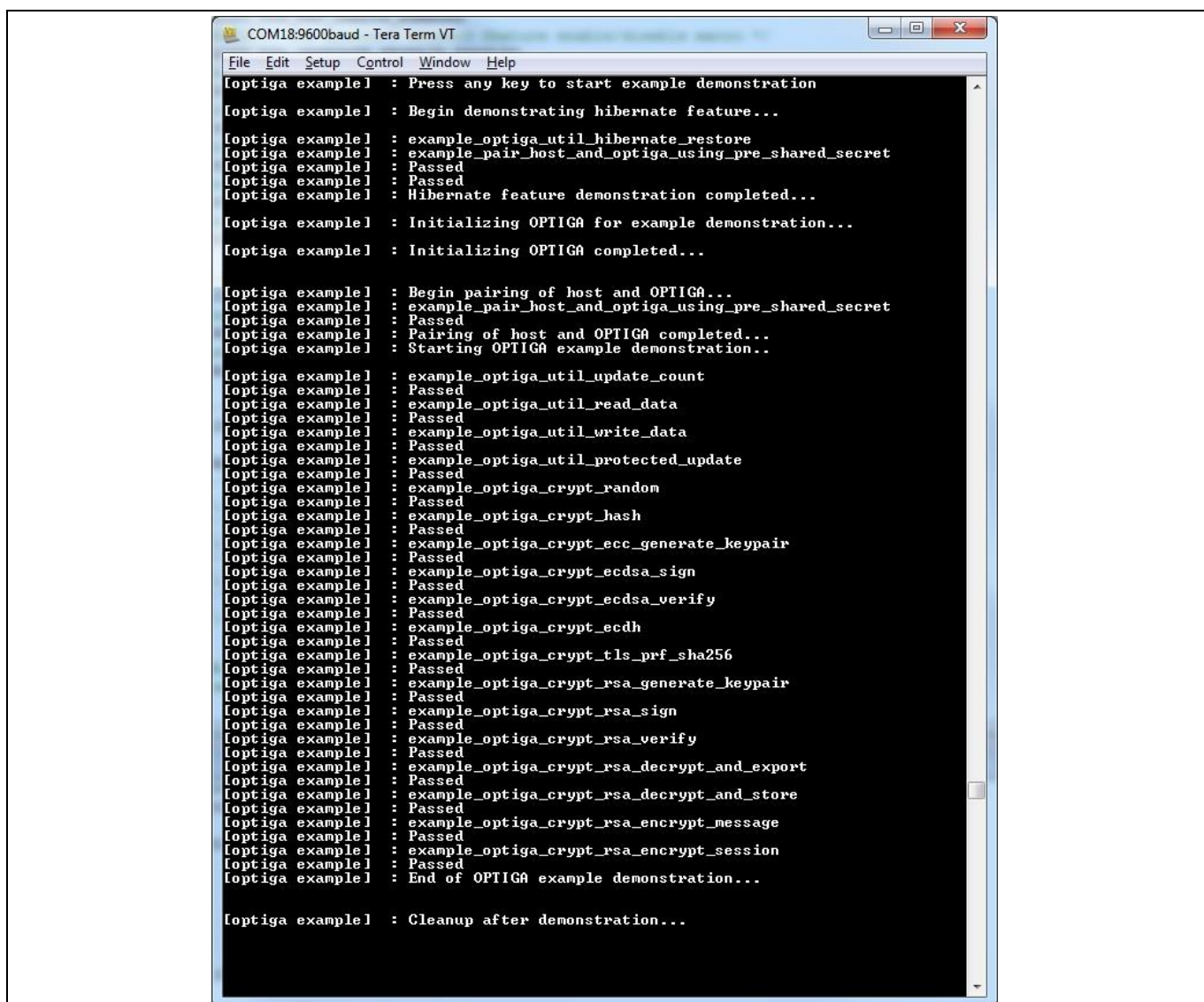


Figure 14 TeraTerm log of example application

Using OPTIGA™ Trust M

4.1.3.2 Logger control

By default only logging from example is enabled in the release package.

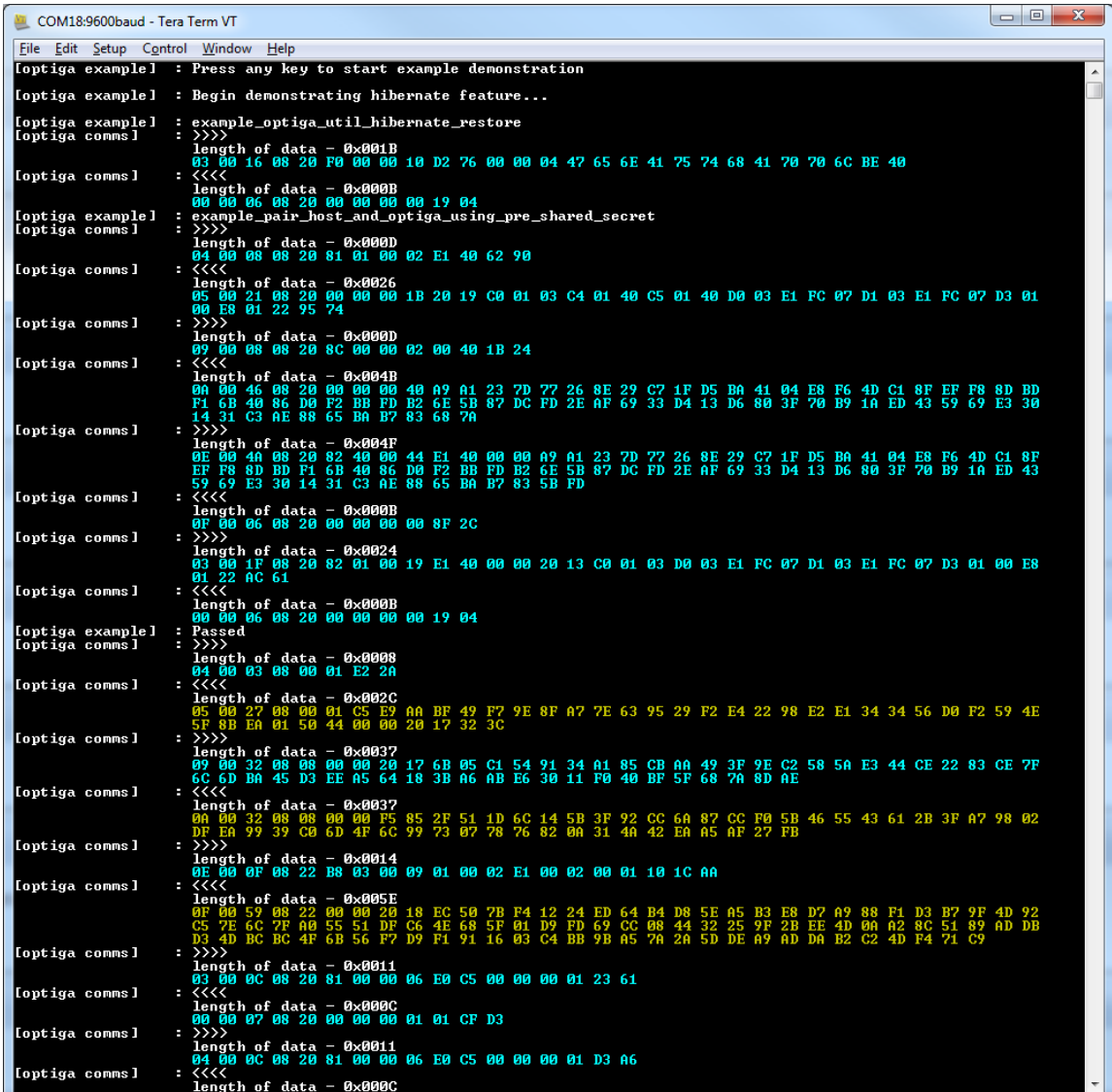
Further control for OPTIGA™ Trust M host code logging is available in `optiga_lib_config.h`.

The macro `OPTIGA_LIB_ENABLE_LOGGING` provides complete control to enable/disable logging at host code. In addition, logging at UTIL, CRYPT, CMD and COMMS layer can be controlled using the following macros,

- `OPTIGA_LIB_ENABLE_UTIL_LOGGING`
- `OPTIGA_LIB_ENABLE_CRYPT_LOGGING`
- `OPTIGA_LIB_ENABLE_CMD_LOGGING`
- `OPTIGA_LIB_ENABLE_COMMS_LOGGING`

For Example,

1. To enable logging for only COMMS layer, enable `OPTIGA_LIB_ENABLE_COMMS_LOGGING` and disable rest all layer macros.
2. Build the project and run the project as defined in 4.1.1



```

COM18:9600baud - Tera Term VT
File Edit Setup Control Window Help
[optiga example] : Press any key to start example demonstration
[optiga example] : Begin demonstrating hibernate feature...
[optiga example] : example_optiga_util_hibernate_restore
[optiga comms] : >>>>
length of data - 0x001B
03 00 16 08 20 F0 00 00 10 D2 76 00 00 04 47 65 6E 41 75 74 68 41 70 70 6C BE 40
[optiga comms] : <<<<
length of data - 0x000B
00 00 06 08 20 00 00 00 00 19 04
[optiga example] : example_pair_host_and_optiga_using_pre_shared_secret
[optiga comms] : >>>>
length of data - 0x000D
04 00 08 08 20 81 01 00 02 E1 40 62 90
[optiga comms] : <<<<
length of data - 0x0026
05 00 21 08 20 00 00 00 1B 20 19 C0 01 03 C4 01 40 C5 01 40 D0 03 E1 FC 07 D1 03 E1 FC 07 D3 01
00 E8 01 22 95 74
[optiga comms] : >>>>
length of data - 0x000D
09 00 08 08 20 8C 00 00 02 00 40 1B 24
[optiga comms] : <<<<
length of data - 0x004B
00 00 46 08 20 00 00 00 40 A9 A1 23 7D 77 26 8E 29 C7 1F D5 BA 41 04 E8 F6 4D C1 8F EF F8 8D BD
F1 6B 40 86 D0 F2 BB FD B2 6E 5B 87 DC FD 2E AF 69 33 D4 13 D6 80 3F 70 B9 1A ED 43 59 69 E3 30
33 31 C3 AE 88 65 BA B7 83 68 7A
[optiga comms] : >>>>
length of data - 0x004F
0E 00 4A 08 20 82 40 00 44 E1 40 00 00 A9 A1 23 7D 77 26 8E 29 C7 1F D5 BA 41 04 E8 F6 4D C1 8F
EF F8 8D BD F1 6B 40 86 D0 F2 BB FD B2 6E 5B 87 DC FD 2E AF 69 33 D4 13 D6 80 3F 70 B9 1A ED 43
59 69 E3 30 14 31 C3 AE 88 65 BA B7 83 5B FD
[optiga comms] : <<<<
length of data - 0x000B
0F 00 06 08 20 00 00 00 00 8F 2C
[optiga comms] : >>>>
length of data - 0x0024
03 00 1F 08 20 82 01 00 19 E1 40 00 00 20 13 C0 01 03 D0 03 E1 FC 07 D1 03 E1 FC 07 D3 01 00 E8
01 22 AC 61
[optiga comms] : <<<<
length of data - 0x000B
00 00 06 08 20 00 00 00 00 19 04
[optiga example] : Passed
[optiga comms] : >>>>
length of data - 0x0008
04 00 03 08 00 01 E2 20
[optiga comms] : <<<<
length of data - 0x002C
05 00 27 08 00 01 C5 E9 AA BF 49 F7 9E 8F A7 7E 63 95 29 F2 E4 22 98 E2 E1 34 34 56 D0 F2 59 4E
5F 8B EA 01 50 44 00 00 20 17 32 3C
[optiga comms] : >>>>
length of data - 0x0037
09 00 32 08 08 00 00 20 17 6B 05 C1 54 91 34 A1 85 CB AA 49 3F 9E C2 58 5A E3 44 CE 22 83 CE 7F
6C 6D BA 45 D3 EE A5 64 18 3B A6 AB E6 30 11 F0 40 BF 5F 68 7A 8D AE
[optiga comms] : <<<<
length of data - 0x0037
00 00 32 08 08 00 00 F5 85 2F 51 1D 6C 14 5B 3F 92 CC 6A 87 CC F0 5B 46 55 43 61 2B 3F A7 98 02
DF EA 99 39 C0 6D 4F 6C 99 73 07 78 76 82 0A 31 4A 42 EA A5 AF 27 FB
[optiga comms] : >>>>
length of data - 0x0014
0E 00 0F 08 22 B8 03 00 09 01 00 02 E1 00 02 00 01 10 1C AA
[optiga comms] : <<<<
length of data - 0x005E
0F 00 59 08 22 00 00 20 18 EC 50 7B F4 12 24 ED 64 B4 D8 5E A5 B3 E8 D7 A9 88 F1 D3 B7 9F 4D 92
C5 7E 6C 7F A0 55 51 DF C6 4E 68 5F 01 D9 FD 69 CC 08 44 32 25 9F 2B EE 4D 0A A2 8C 51 89 AD DB
D3 4D BC BC 4F 6B 56 F7 D9 F1 91 16 03 C4 BB 9B A5 7A 2A 5D DE A9 AD DA B2 C2 4D F4 71 C9
[optiga comms] : >>>>
length of data - 0x0011
03 00 0C 08 20 81 00 00 06 E0 C5 00 00 00 01 23 61
[optiga comms] : <<<<
length of data - 0x000C
00 00 07 08 20 00 00 00 01 01 CF D3
[optiga comms] : >>>>
length of data - 0x0011
04 00 0C 08 20 81 00 00 06 E0 C5 00 00 00 01 D3 A6
[optiga comms] : <<<<
length of data - 0x000C

```

Figure 15 Logging data with only COMMS layer enabled

Using OPTIGA™ Trust M

Note: Execution time of example increase if more logging information is printed.

4.2 Advanced Setup

This section explains the steps to build and run OPTIGA™ Trust M example application.

4.2.1 Setting up DAVE™ IDE on PC

1. Refer to the installation guide in <DAVE_PATH> to install DAVE™ on your PC.
2. Start DAVE™ from <DAVE_PATH>\eclipse\DAVE.exe. The following splash screen will appear:

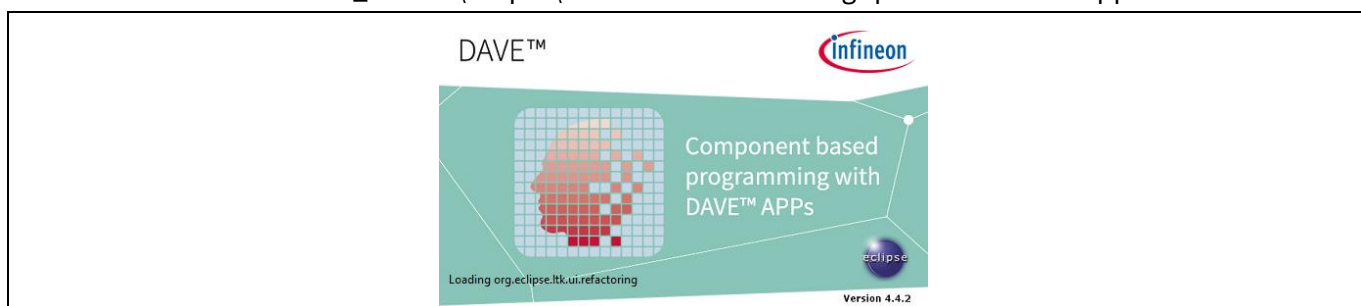


Figure 16 Opening DAVE™

3. Eclipse Launcher will pop-up. Select the workspace for DAVE™.

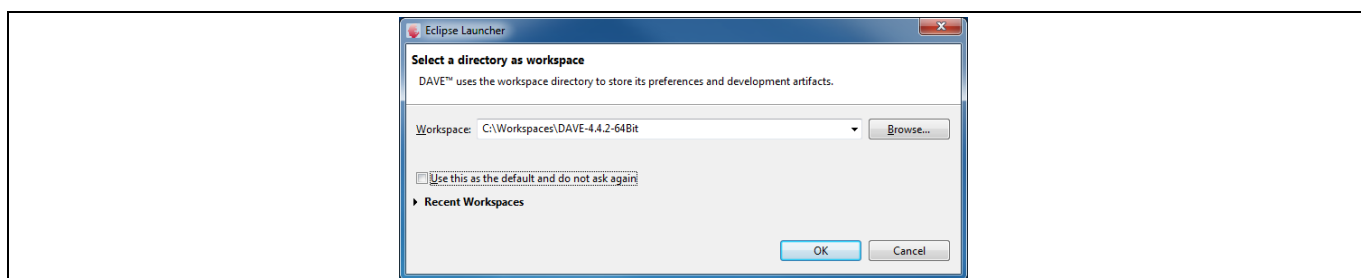


Figure 17 Select workspace

4. DAVE IDE enabled window.

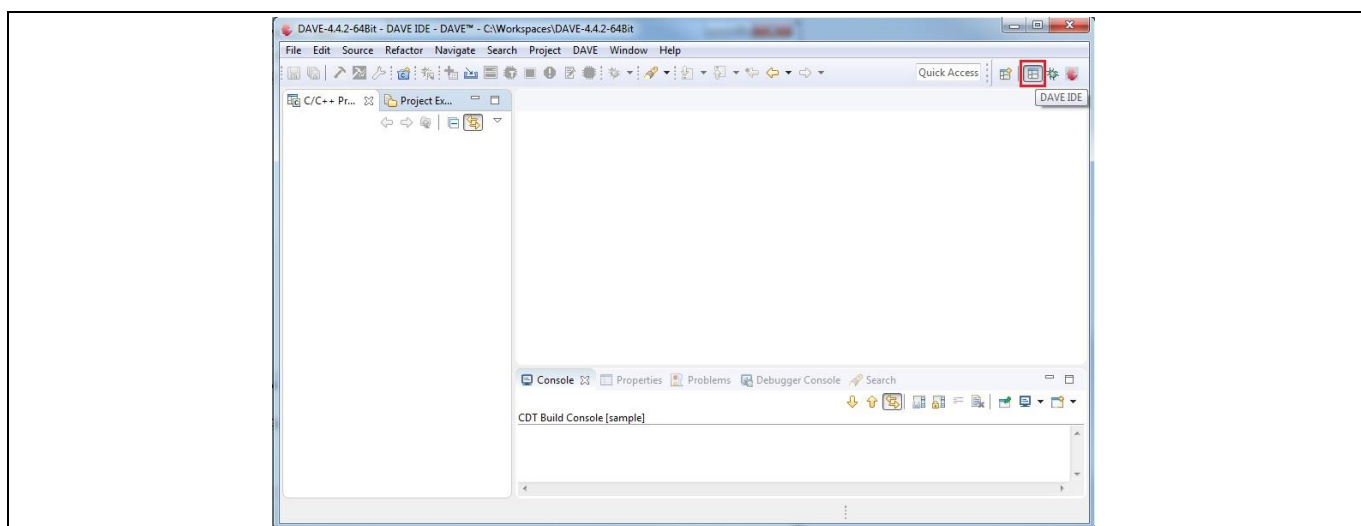


Figure 18 DAVE IDE Perspective window

Using OPTIGA™ Trust M

4.2.2 Running OPTIGA™ Trust M Example Application Project with DAVE™

1. Make the connections among XMC4800 IoT Connectivity Kit, My IoT Adapter and OPTIGA™ Shield2Go.
2. Power up the kit by connecting Micro USB cable between PC and Debugger micro USB. For placement of Debugger micro USB refer Figure 3.
3. Import example application project into DAVE IDE, by navigating through **File -> Import**. In Import pop-up, select Existing Projects into Workspace under General and then click Next.

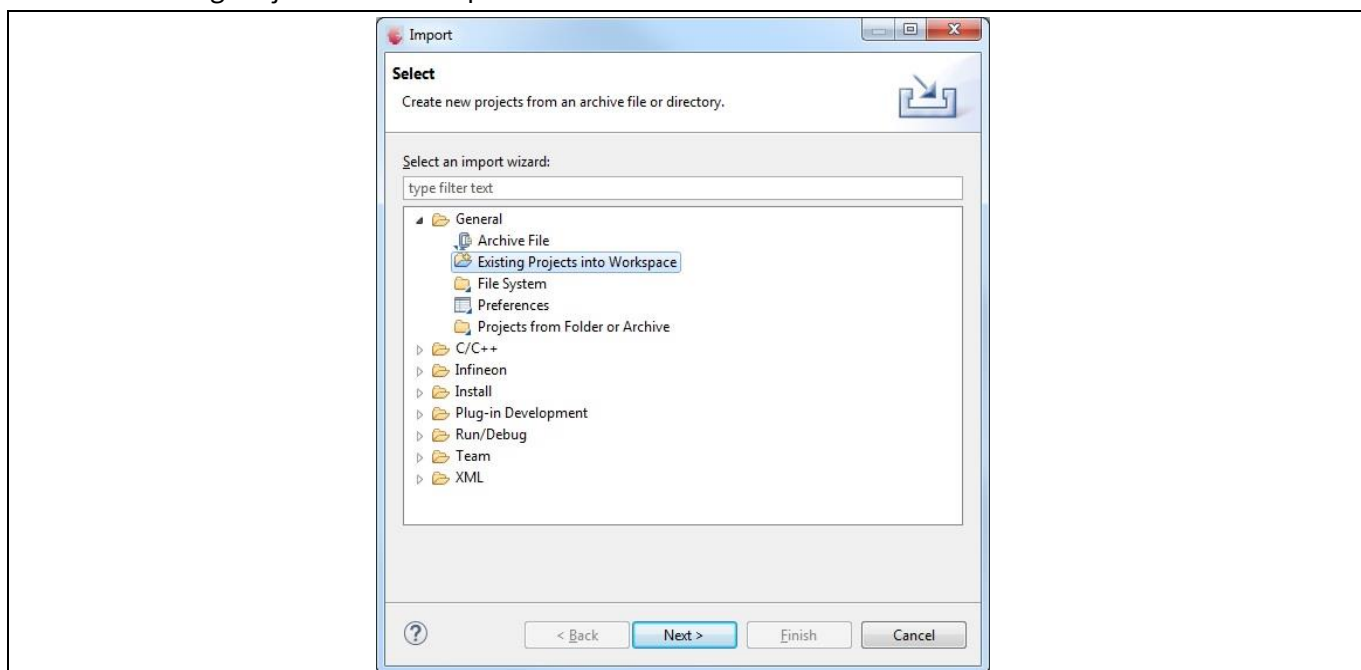


Figure 19 Import DAVE project window

4. Browse to <INSTALLDIR>\projects\xmc4800_iot_kit\dave4 for Select root directory, select xmc4800_optiga_example and then click Finish.

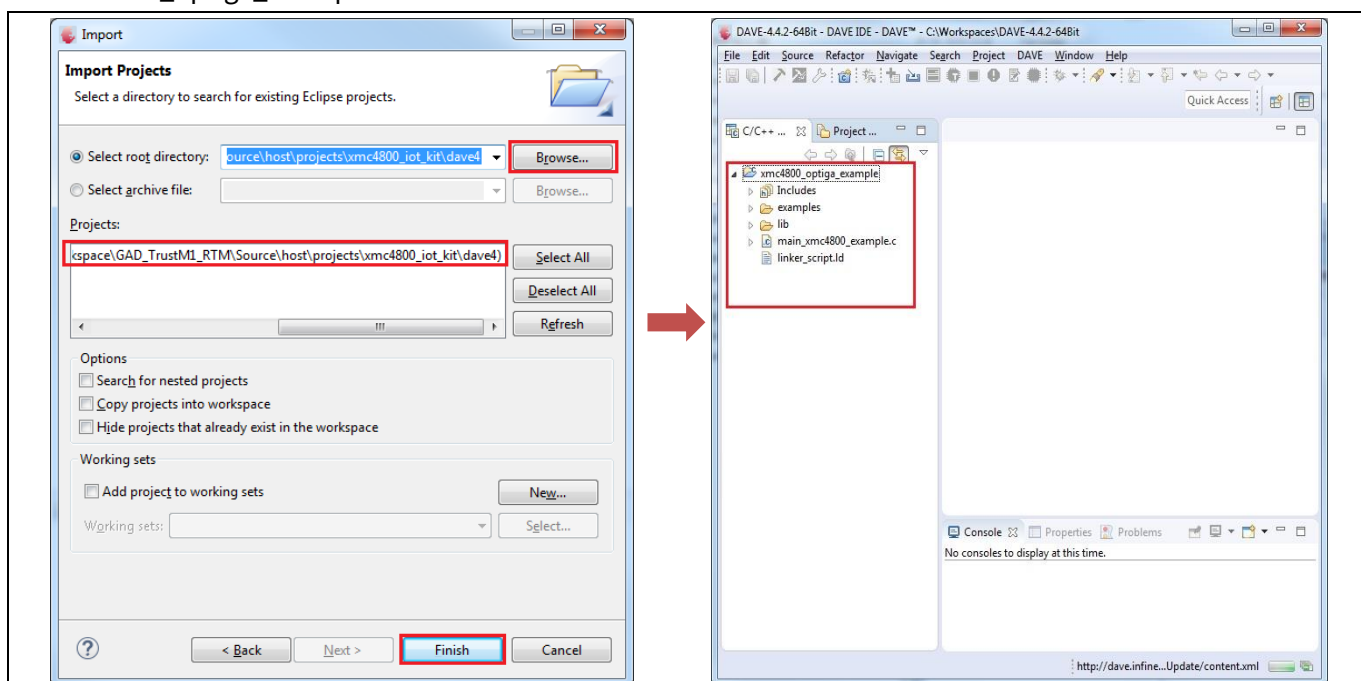


Figure 20 Import a example project

Using OPTIGA™ Trust M

5. Set example project as an active project by right-click on project and select **Set Active Project**.

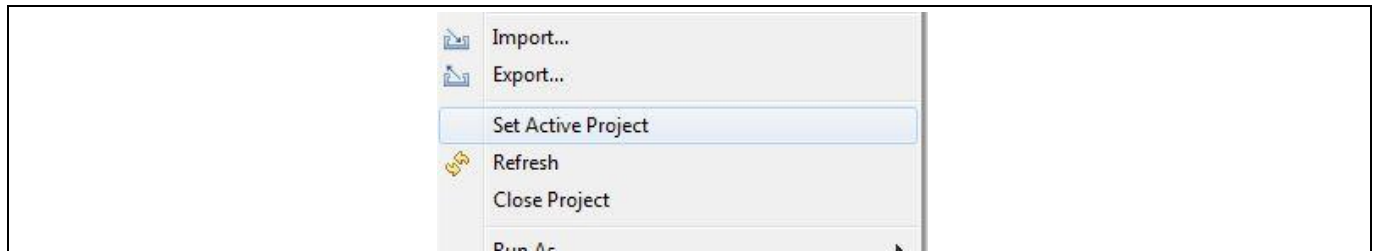


Figure 21 Example project set as active project

6. Select the build configuration by right-click on example project and then select **Build Configurations -> Set Active -> Debug**.

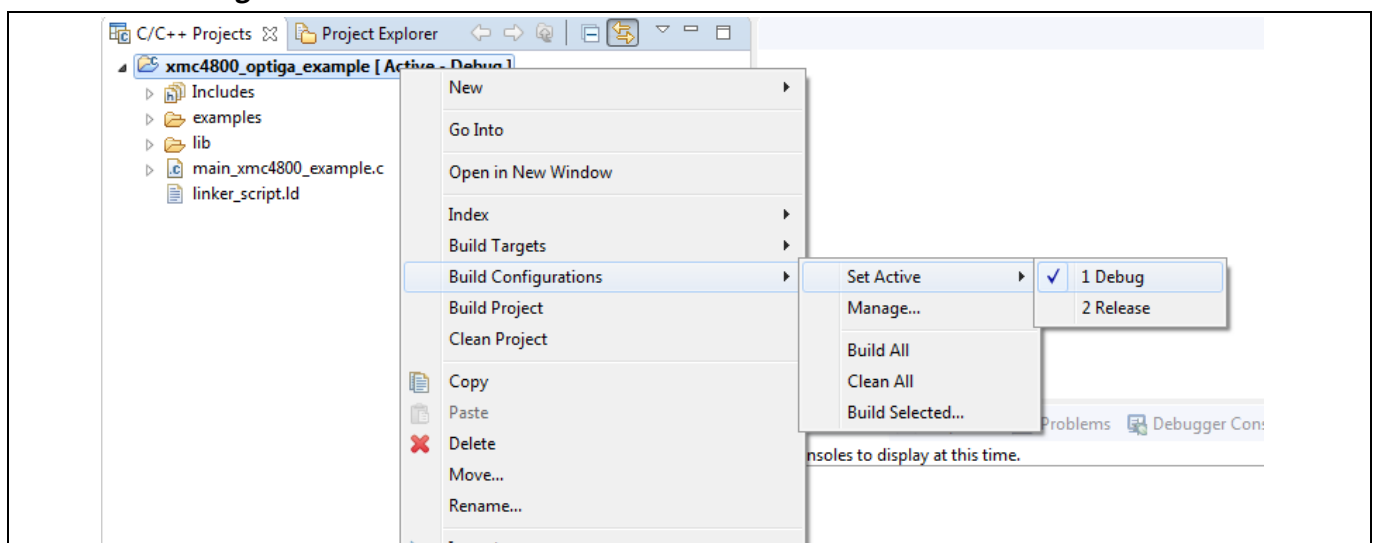


Figure 22 Build configuration selection

7. Build the project in debug configuration. It should be error free.

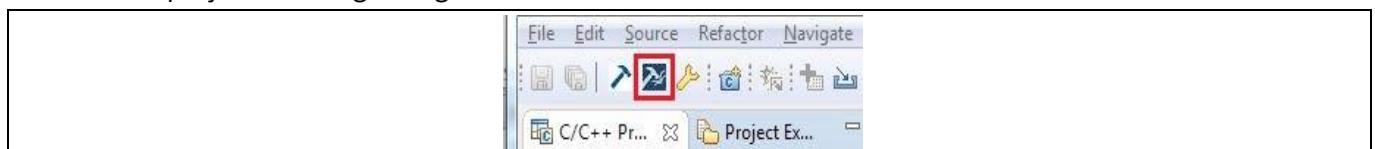


Figure 23 Example build in debug

8. Before launching the debugger, ensure that values are properly updated for variables like jlink_gdbserver and jlink_path. Navigate through **Window -> Preferences -> Run/Debug -> String Substitution** and update values as shown in the figure below:

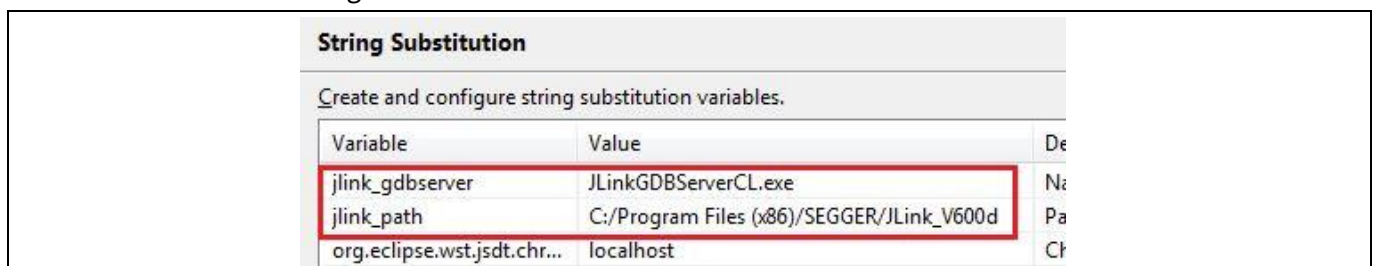


Figure 24 J-Link variable mapping

9. Launch debugger for debug of example application by clicking on bug symbol.

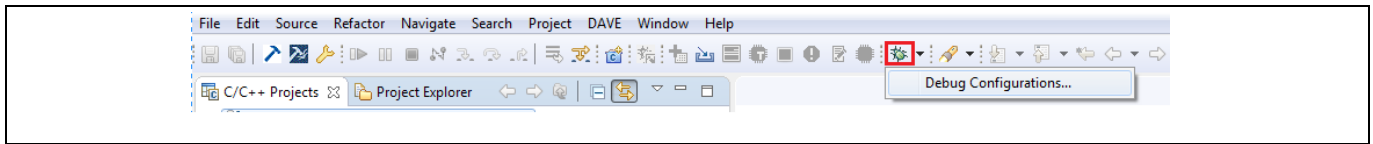


Figure 25 Debugger launch

10. Create a debug configuration and then click on Debug.

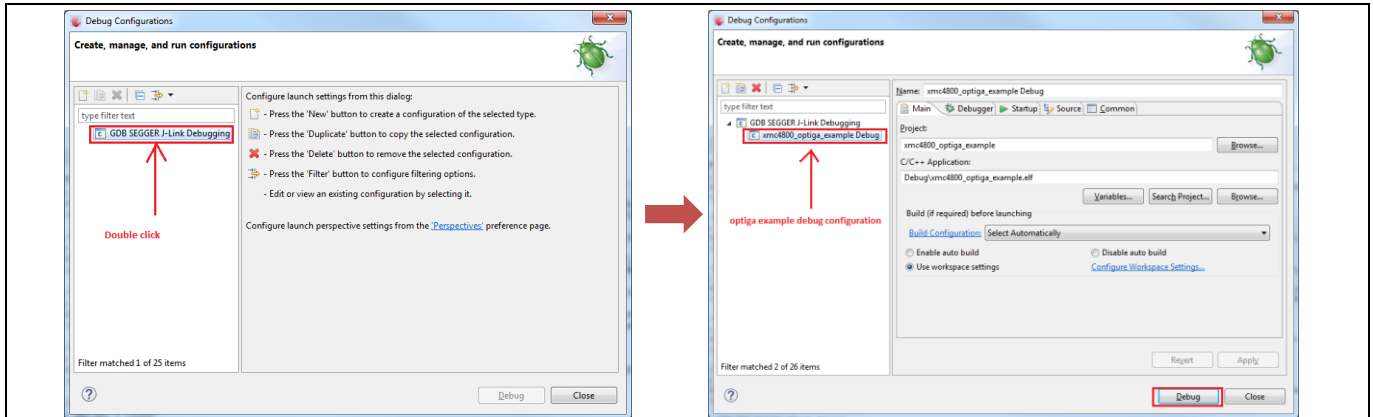


Figure 26 Creating a example debug configuration

11. If a window prompts to confirm the perspective switch, check the Remember my decision, and click yes.

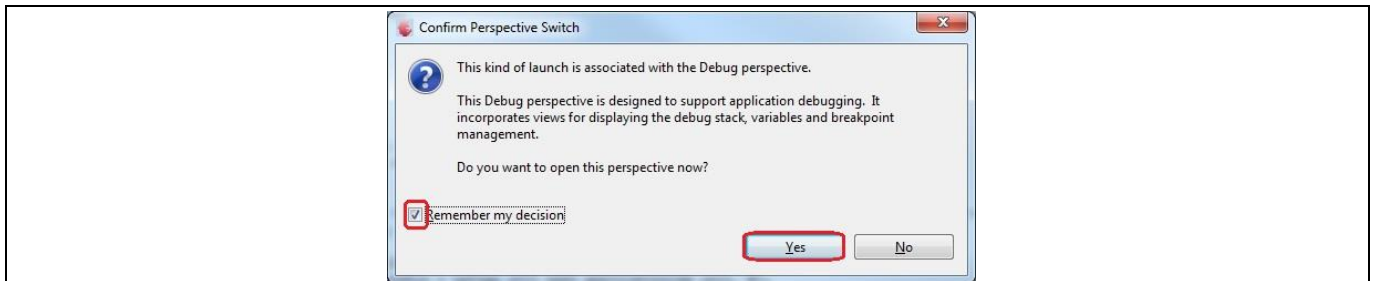


Figure 27 Confirm perspective switch

12. Start the debugger.

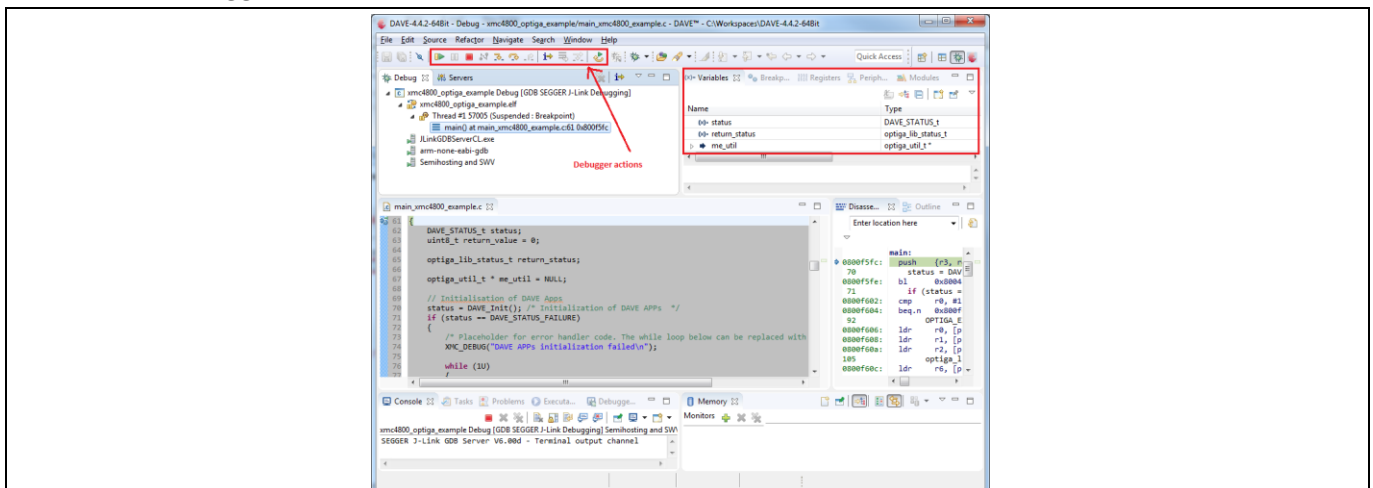


Figure 28 Starting a debug session

Using OPTIGA™ Trust M

13. Refer section 4.1.3.1 to setup USB_D_VCOM for logger.
14. Example logs can be seen on terminal as shown in Figure 14.
15. To build project in release configuration, select the build configuration as **Release** as shown in Figure 22 and build the project again.
16. Create a new configuration by right-click on example project and **Run As -> Run Configurations**. Double-click on GDB SEGGER J-Link Debugging and select Release configuration then click on Run. The logs of the executing example can be seen on the terminal as shown in Figure 14.

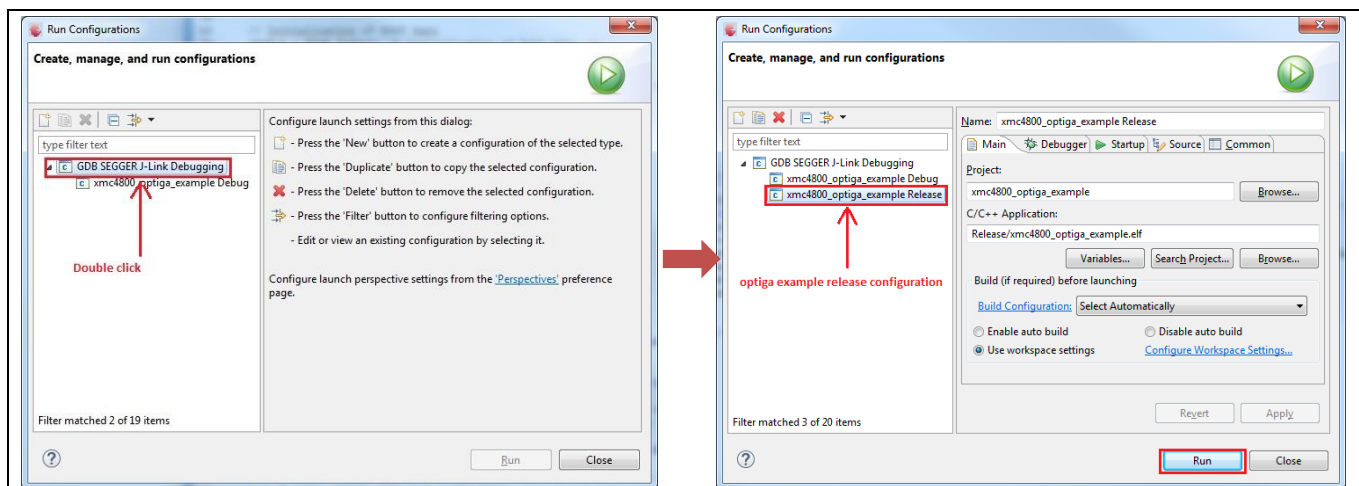


Figure 29 Run example with release configuration

17. To execute the example without shielded connection, disable the macro OPTIGA_COMMS_SHIELDED_CONNECTION in file optiga_lib_config.h.

```

/** @brief OPTIGA COMMS shielded connection feature.
 *      To disable the feature, undefine the macro
 *
 */
// #define OPTIGA_COMMS_SHIELDED_CONNECTION

```

Figure 30 OPTIGA_COMMS_SHIELDED_CONNECTION disable

18. To run example application with logging for different layers refer to section 4.1.3.2.

Troubleshooting

5 Troubleshooting

Table 6 Troubleshooting

No	Problem	Reason	Solution
1	The Green LED light is “Not on” on XMC4800 IoT Connectivity kit	No power supply	Verify that power supply is connected to XMC4800 IoT Connectivity kit.
2	CDC port not detected	SW not correctly installed	In device manager, click on the malfunctioning CDC port and select to manually install the driver. Provide directory as C:\ for path to install the driver.
3	Problem occurred during debugger launch	Debug session is not terminated	Go to Debug perspective and remove all terminated launches.

Revision History

Revision History

Table 7

Document version	Date of release	Description of changes
1.00	01-08-2019	Release to production release

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2019-08-01

Published by

Infineon Technologies AG
81726 Munich, Germany

© 2019 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about this document?

Email:
DSSCustomerService@infineon.com

Document reference

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.