Visual Studio and Eclipse Debugging for i.MX 8M FamilyRev. 0 — 11 November 2022Application note

Document information

Information	Content		
Keywords	AN13739, i.MX 8M, Yocto, Eclipse IDE, Microsoft Visual Studio IDE		
Abstract	This document provides multiple solutions for cross-compilation, deployment, and debugging of applications that target the i.MX 8M family.		



1 Introduction

This document describes the processes to set up the Windows software development environment for the i.MX 8M family devices in various development environments.

In addition, it helps the user to cross-compile, deploy, and debug the code for an i.MX 8M device in the following two Integrated Development Environments (IDEs):

- Microsoft Visual Studio
- Eclipse

The programming languages covered in this document are:

- C/C++
- Python

The main objectives of this document are:

- Describing the board setup for both hardware and software. The setup process includes the steps to build the image from scratch using the Yocto project.
- Setting up the connection from PC to board.
- Setting up IDEs for the programming languages, such as C and Python. The IDEs setup configurations include code and project examples.

2 **Prerequisites**

This section provides the required list of hardware and software components that should be installed in prior to setting up the Windows software development environment.

2.1 Hardware

<u>Table 1</u> lists the required hardware components:

 Table 1. Hardware Components

Component	Description
NXP i.MX 8M	Mini EVK board
SanDisk Ultra 32 GB Micro SDHC	Micro SD card
USB Type-A to Type-C, or USB Type-C	Power cable for power port
Ethernet cable	Cable to connect the devices of the network

2.2 Software

The configuration of Eclipse and Visual Studio environments require specific software tools.

Table 2 lists the required software components:

Table 2.	Software	components
----------	----------	------------

Environment	Software component	Description
Linux	Linux BSP Release <u>5.10.72_2.2.0</u>	Build Linux environment and customized with Yocto.
Eclipse	C/C++	Develop applications in C or C++ using Eclipse IDE.

© 2022 NXP B.V. All rights reserved.

Table 2. Software components...continued

Environment	nvironment Software component Description	
	Download link for Eclipse: <u>https://</u> www.eclipse.org/downloads/ packages/ Toolchain installer file for Windows: gcc-arm-10.3-2021.07-mingw-w64-i68 <u>6-aarch64-none-linux-gnu.tar.xz</u>	Develop applications in C or C++ using Eclipse IDE.
	Python for Java Java version used in this context is 2022-03 R	Develop applications in Python using Eclipse IDE.
Visual Studio	Microsoft Visual Studio 2022	Develop applications in C++ using Microsoft Visual Studio IDE.

3 Yocto project

This section describes the Linux image customizations and Yocto file modifications that are required for the Yocto project setup.

The Yocto project is an open source collaboration project that helps developers create a customized Linux-based system regardless of the hardware architecture.

3.1 Customizing Linux image

Eclipse uses specific ciphers that are not available in the default SSH server manager on the Dropbear board. The following specific customizations of the image are required for connecting to the board through SSH:

- Changing the default SSH server manager on the board (Dropbear) to the server manager compatible with the presented IDEs (OpenSSH).
- Including the SSH File Transfer Protocol (SFTP) tools to the image. These SFTP tools are required for remote file transfer between the board and the presented IDEs.

The following list of tools is required for remote debugging:

- /usr/libexec/sftp-server
- /usr/bin/sftp
- /usr/bin/gdbserver
- /usr/lib/libstdc++.so.6

Note: The tools listed above are included in the Linux image for the build.

3.2 Modifying Yocto file

To remove Dropbear from the board, the Yocto file local.conf parameters are modified. In addition, the OpenSSH package is customized for compilation.

To modify the Yocto recipe parameters, perform the following steps on a Linux host or VM:

1. Run the following commands:

```
$ mkdir ~/bin ; (This step is may not required, if the bin
folder already exists.)
```

© 2022 NXP B.V. All rights reserved.

Visual Studio and Eclipse Debugging for i.MX 8M Family

```
$ curl https://storage.googleapis.com/git-repo-downloads/repo
> ~/bin/repo
$ chmod a+x ~/bin/repo
$ git config --global user.name "Your Name"
$ git config --global user.email "Your Email"
$ git config --list
$ mkdir imx-yocto-bsp
$ cd imx-yocto-bsp
$ repo init -u https://source.codeaurora.org/external/imx/
imx-manifest -b imx-linux-hardknott -m imx-5.10.72-2.2.0.xml
$ repo sync
$ DISTRO=fsl-imx-xwayland MACHINE=imx8mm-lpddr4-evk source
imx-setup-release.sh -b build-comp
```

- 2. To remove Dropbear, perform the following steps:
 - a. Open the recipe imx-yocto-bsp/sources/meta-imx/meta-sdk/ recipes-fsl/images/imx-image-core.bb
 - b. Locate the following lines in the code:

- Remove the line ssh-server-dropbear \ from the yocto recipe imx-yoctobsp/sources/meta-imx/meta-sdk/recipes-fsl/images/imx-imagecore.bb.
- Open the yocto recipe imx-yocto-bsp/sources/meta-imx/meta-sdk/ recipes-fsl/images/imx-image-multimedia.bb
- 5. Remove the line ssh-server-dropbear \ (same line as in the yocto recipe imximage-core.bb).
- 6. Modify the line imx-yocto-bsp/build-comp/conf/local.conf by adding the following lines at the end of it:

```
#SSH
PACKAGE_EXCLUDE += " packagegroup-core-ssh-dropbear"
CORE_IMAGE_EXTRA_INSTALL += "openssh"
```

7. From the directory path imx-yocto-bsp/build-comp, build the image image imx-image-multimedia using the following command:

```
$ bitbake imx-image-multimedia
```

For more information regarding the Yocto image build, refer to *i.MX Yocto Project User's Guide* (document <u>IMXLXYOCTOUG</u>).

The built image name and its location are given below:

• Location: imx-yocto-bsp/build-comp/tmp/deploy/images/imx8mmevk

- **Image name**: imx-image-multimedia-imx8mmevk.wic.bz2.
- 8. After completion of the build, connect the board through Putty using the following steps:
 - a. Flash the built image on the SD card.
 - Insert the SD card in the board and then connect the board through Putty.
 For more information on image flashing and board connection process, refer to *i.MX Linux User's Guide* (document <u>IMXLUG</u>).

4 Configuring LAN

This section helps the user to create Local Area Network (LAN) between the host PC and the board using an Ethernet cable. In addition, it provides the steps to assign the static IP addresses to the interfaces on each end of the cable.





Figure 1. Ethernet cable

4.1 Host PC configuration

To configure the IP address for the host PC, perform the following steps:

- 1. On the host PC, navigate through Control Panel > Network and Internet > Network and Sharing Center.
- 2. Click Network Status and Tasks to view the network connections.
- 3. From **Connections**, click the connected network. For example, Ethernet is the connected network in the Figure 2.

NXP Semiconductors

AN13739

Visual Studio and Eclipse Debugging for i.MX 8M Family

Setwork and Sharing Center		– 🗆 X
	and Internet > Network and Sharing Center	✓ U Search Control Panel
	View your basic network information	
Control Panel Home	View your active networks	
Change adapter settings		
Change advanced sharing settings	wbi.nxp.com Domain network	Access type: Internet Connections: & Ethernet 3
Media streaming options		
	Unidentified network	Access type: No network access
	Public network	Connections: 🥙 Ethernet
	Change your networking settings	
	Set up a new connection or network Set up a broadband, dial-up, or VPN cor	nnection; or set up a router or access point.
	Troubleshoot problems	
	Diagnose and repair network problems,	or get troubleshooting information.
See also		
Internet Options		
Windows Defender Firewall		
Figure 2. Network and sh	aring center	
4. Click Properties. The	Ethernet Properties dialog	box appears.
	Ethernet Properties	×
	Networking Authentication Sharing	
	Connect using: Intel(R) Ethemet Connection (10) I219-LM	
	This connection uses the following items:	onfigure
	Internet Protocol Version 4 (TCP/IPv4)	^
	Image: Microsoft Network Adapter Multiplexor Protoc Image: Microsoft LLDP Protocol Driver	loi
	Internet Protocol Version 6 (TCP/IPv6)	
	Link-Layer Topology Discovery Responder	hiver
	Link-Layer Topology Discovery Mapper I/O D L Hyper-V Extensible Virtual Switch	viver v
	<	>
		roperties
	Description Transmission Control Protocol/Internet Protocol. The	e default
	wide area network protocol that provides communic across diverse interconnected networks.	
	ОК	Cancel
Einen O. Ethomat		
Figure 3. Ethernet proper	rties	

5. In the Ethernet Properties dialog box, select Internet Protocol Version 4 (TCP/ IPv4) and click OK.

The Internet Protocol Version 4 (TCP/IPv4) Properties dialog box appears.

 In the Internet Protocol Version 4 (TCP/IPv4) Properties dialog box, configure the static IPv4 address for the interface assigned to the PC (from the Ethernet cable). Use the IP address details provided in the Figure 4.

Internet F	Protocol Version 4 (TCP/IPv4) Properties	\times
General			
this cap		matically if your network supports o ask your network administrator	
Oot	otain an IP address automatica	lly	
Us	e the following IP address: —		
IP ad	ldress:	18 . 125 . 5 . 251	
Subn	et mask:	255.255.255.0	
Defa	ult gateway:		
O	otain DNS server address auto	matically	
O Us	e the following DNS server add	dresses:	
Prefe	erred DNS server:		
Alter	nate DNS server:		
v	alidate settings upon exit	Advanced	
		OK Cance	ł

Figure 4. IPv4 properties configuration

- 7. Enter the IP address and its Subnet mask. For example, the IP (IPv4) address is 18.125.5.251 and the subnet mask is 255.255.255.0.
- 8. Click **OK**.

4.2 Board configuration

To connect the board by assigning an IP address, perform the following steps:

- On the board, use the following command to create a new Ethernet wired network: \$ cat <<"EOF" >> /lib/systemd/network/80-wired.network The newly created prompt (Ethernet network) is opened.
- 2. Navigate to the newly opened prompt for configuring the IP address.
- 3. Copy the IP address provided in the below example or use a different IP address: **Note:** The IP addresses of the PC and the board are configured within the same IP network. Ensure that the Ethernet cable is connected between the board and PC.

```
[Match]
Name=eth0
KernelCommandLine=!ip
[Network]
Address=18.125.5.252/16
EOF
```

4. Reboot the board.

Note: If the Ethernet cable is connected, the IP address is automatically assigned.

- 5. To verify whether the board is configured with the IP address or to retrieve the board IP address, perform the following steps:
 - a. Run the command if command.
 - b. Search for eth0 configuration.
 For example, eth0 is configured with IP address 18.125.5.252 in this context.

5 Configuring Eclipse environments

This section describes installation, setting up, and debugging using the Eclipse IDE.

In addition, this section describes the usage of Eclipse IDE developed using multiple programming languages, such as C and Python. The programming languages include specific installations and usage examples.

5.1 Configuring IDE using C language

This section describes the steps to download and install Toolchain, and Eclipse IDE packages. In addition, it describes the process to set up the Eclipse IDE using C as the programming language.

5.1.1 Installing Toolchain

C is a compiled language and its applications for the i.MX 8M board architecture involves cross-compilation. Toolchain is used to build the projects in C language.

Toolchain contains a set of tools, such as a compiler, linker, and assembler. These tools are required to build the projects in C/C++.

To install Toolchain, perform the following steps:

1. Download the Toolchain installer package for Windows by using the link given below: gcc-arm-10.3-2021.07-mingw-w64-i686-aarch64-none-linux-gnu.tar. xz

Note: The Windows version 10.3-2021.07 is used in this context.

- 2. After downloading the package, extract the files of the installer package. For example, extract the files using the application <u>7Zip</u>.
- 3. Launch the setup and install the Toolchain. *Note: Prefer to download in the default location.*

5.1.2 Installing Eclipse IDE

This section explains the steps to download and install the Eclipse IDE and configure the workspace.

To install the Eclipse IDE, perform the following steps:

- Download the Eclipse package using the link given below: <u>Eclipse IDE for Embedded C/C++ Developers</u> For example, the Eclipse version 12.2021 is used in this context.
- 2. Open the downloaded folder.
- 3. Create a workspace or use a desired workspace. For example, the workspace imx8-C-eclipse-workspace is used this context.

	Eclipse IDE Launcher	×
	Select a directory as workspace	
	Eclipse IDE uses the workspace directory to store its preferences and development artifacts.	
	Workspace: C:\Users\ <username>\imx8-C-eclipse-workspace Browse</username>	
	Use this as the default and do not ask again	
	Launch Cancel	
Figure 5. Eclipse	workspace	

5.1.2.1 Workspace selection

To create a C/C++ project, perform the following steps:

- 1. Open the Eclipse IDE UI.
- 2. Navigate File > New.
- 3. Select C/C++ Project.
 - A new project window C/C++ Project opens.

imx8-C-eclipse-workspace - Ecli	pse IDE		
File Edit Source Refactor Navig	jate Search Project R	un Window Help	
New	Alt+Shift+N >	Makefile Project with Existing Code	
Open File		C/C++ Project	
Open Projects from File System	n	Project	
Recent Files	>	Con Create a new C or C++ project ct	
Close Editor	Ctrl+W	Convert to a C/C++ Project (Adds C/C++ Nature)	re)
Close All Editors	Ctrl+Shift+W	Source Folder	
		🐸 Folder	
Save	Ctrl+S	Source File	
Save As	Ci 1 Ci 16 C	Header File	
Save All	Ctrl+Shift+S	C File from Template Class	
Revert			
Move		Example	
🖄 Rename	F2	Cother	Ctrl+N
Refresh	F5		
re 6. New C/C++ Project			

5.1.3 Sample program

To create a sample Hello, World! program in C or C++, perform the following steps:

- 1. Navigate to C/C++ Project > New Templates for New C/C++ Project.
- 2. In the left pane, click All.
- 3. Select **C Managed Build** or **C++ Managed Build** depending on your build requirement.

Visual Studio and Eclipse Debugging for i.MX 8M Family

New C/C++ Project	- 🗆 X
Templates for New C/C++ Project	
All CMake C Managed Build CMake A C Project build using the CDT's managed build	system.
Meson C++ Managed Build C++ Project build using the CDT's managed built	ild system.
CMake Project A CMake project with a Hello World executable t	o get started.
Empty or Existing CMake Project Create a CMake project with no files. Can be use	d to create
one over existing content. Empty or Existing Meson Project Create a Meson project with no files. Can be used	l to create
Makefile Project (Experimental) Create a new project that builds w	ith
the'make' build tool using CDT's new Core Build Meson Project	System.
A Meson project with a Hello World executable to	i get started.
(2) < Back Next > Finish	Cancel
Figure 7. C menaged build	
Figure 7. C managed build	at dialage bay
4. Click Next and navigate to the corresponding proje	ect dialog box.
In this context, C Project window appears.	
	- • ×
C Project	
Create C project of selected type	
Project name: hello_world_imx8	
Use default location	
Location: C\Users\NXF68462\imx8-C-eclipse-workspace\hello_world_imx Choose file system: default	8 Browse
Project type: Toolchains:	
Empty Project Arm Cross GCC	
Hello World ANSI C Project Cross GCC Hello World Arm C Project MinGW GCC	
Hello World RISC-V C Project RISC-V Cross GCC ADuCM36x C/C++ Project	
 Hello World Arm Cortex-M C/C++ Project 	
Freescale Processor Expert C/C++ Project Freescale Kinetis KLxx C/C++ Project	
SiFive RISC-V C/C++ Project STM32F0xx C/C++ Project	
 STM32F1xx C/C++ Project 	
STM32F2xx C/C++ Project STM32F3xx C/C++ Project	
STM32F4xx C/C++ Project STM32F7xx C/C++ Project	
<pre>SIM32F/xx C/C++ Project </pre>	
Show project types and toolchains only if they are supported on the pl	atform
⑦ < Back Next > Fini	sh Cancel
(2) < Back Next > Fini	Cancer
Figure 8. Project type and toolchains	

Figure 6. Project type and toolchains

- 5. Assign a name for the project.
- 6. In the Project Type section, select the Empty Project option.
- 7. In the Toolchains section, select the Cross GCC option.
- 8. Click Next.
- The Select Configurations window appears.
- 9. In the Select Configurations window, do not change the default settings.
- 10. Click Next.

Visual Studio and Eclipse Debugging for i.MX 8M Family

		– 🗆 X	
	Select Configurations		
	Select platforms and configurations you wish to deploy on		
	Project type: Executable		
	Toolchains: Arm Cross GCC Configurations:		
	☑ [©] [©] [©] [©] [©]	Select all	
	🖾 🛞 Release	Deselect all	
		Advanced settings	
		Advanced settings	
	Use "Advanced settings" button to edit project's properties.		
	Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on property	/ partes	
	ose manage configurations battons ether on tooloar of on property	pages.	
	② < Back Next >	Finish Cancel	
	ss GCC Command tab, the Toolchain a		vt
lote: Th	ess GCC Command tab, the Toolchain a e Toolchain aarch64-none-linux-gr a in the Cross compiler prefix as shown	nu-gcc displays in this conte n in <u>Figure 10</u> .	
lote: Th	e Toolchain aarch64-none-linux-g	nu-gcc displays in this conte	xt.
lote: Th add prefix	e Toolchain aarch64-none-linux-gi in the Cross compiler prefix as shown	nu-gcc displays in this conte n in <u>Figure 10</u> .	
lote: Th .dd prefix ss GCC Comma	e Toolchain aarch64-none-linux-gr in the Cross compiler prefix as shown	nu-gcc displays in this conte n in <u>Figure 10</u> .	
lote: Th add prefix as GCC Comma nfigure the Cross	e Toolchain aarch64-none-linux-gi in the Cross compiler prefix as shown	nu-gcc displays in this conte n in <u>Figure 10</u> .	

Visual Studio and Eclipse Debugging for i.MX 8M Family

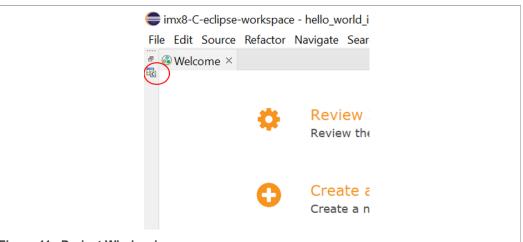


Figure 11. Project Window icon

16. Open the **Project Explorer** tab and create a source folder **src** for your project.

17. Click Finish.

imx8-C-eclipse-works	New	>	🗅 Project		
File Edit Source Refac	Go Into		🗅 File		
	Open in New Window		File from Template		
🖉 눱 Project Explorer 🗵	Show In	Alt+Shift+W >	🗳 Folder		
► world_imx	Show in Local Terminal		Create a new folder resource		
> 🔊 Includes			Header File		
	Сору	Ctrl+C	Source File		
1	Paste	Ctrl+V	Source Folder		
2	Delete	Delete			

Figure 12. New folder

18. Add the new file to the src folder.

	CNew Folder		- 🗆 ×
	Folder Create a new folder resource.		
	Enter or select the parent folder:		
	hello_world_imx8		
	Folder name: src Advanced >>]
	0	Finish	Cancel
Figure 13. src folder			

Figure 13. src folder

This new file is the source of Hello, World!. Ensure that the extension *.c is added for the C projects and *.cpp for the C++ projects.

Visual Studio and Eclipse Debugging for i.MX 8M Family

	Create New File - X
	File Create a new file resource.
	Enter or select the parent folder: hello world imx8/src
	neilo_wond_imx8/src
	 ✓ Settings
	Src
	File name: hello_world.c
	Advanced >>
	⑦ Finish Cancel
	Figure 14. New file
	19. Write a simple program for Hello, World! using the steps given below:
	a. In the Notepad application, open a new blank file.
	b. Write the steps as shown in <u>Figure 15</u> and <u>Figure 16</u> .
	c. Save it with a new filename hello world by adding an extension *.c.
	<pre> Bello_world.c × 1 #include <stdio.h> </stdio.h></pre>
	2®int main() { 3 // printf() displays the string inside quotation
	<pre>4 printf("Hello, World!");</pre>
	5 return 0; 6 }
	7
	Figure 15. "Hello, World!" example program in C
	For C++ project, save the file by adding an extension *.cpp.
	<pre>#include <iostream></iostream></pre>
	using namespace std;
	<pre>int main() {</pre>
	cout << "Hello, Worldplusplus!" << endl;
	}
	Figure 16. "Hello, World!" example program in C++
	d. To build the debug version of the executable file, click the hammer icon. Note: Do not receive any errors on cross-compilation.
3.1	Debugging image configurations
	This section provides the process to debug the configurations created in C/C++.
	To debug the Linux image configurations, perform the following steps:

1. From the menu bar, navigate to **Run > Debug Configurations** to debug the configurations.

Note: A similar Debug Configurations window appears on the screen.

© 2022 NXP B.V. All rights reserved.

Visual Studio and Eclipse Debugging for i.MX 8M Family

Debug Configurations					is.
reate, manage, and run configu	ations			, Contraction of the second seco	0
Program not specified					5
1 🖻 🕫 🗎 🗮 🖻 🏹 🔻	Name: hello_world_imx8 Debug				_
type filter text	🖫 Main 🕪 Arguments 🎘 Debugger 🦻 Source 🗉	Common			
C/C++ Application	Project:				
C/C++ Attach to Application	hello_world_imx8			Browse	
 C/C++ Container Launcher C/C++ Postmortem Debugger 	C/C++ Application:			browse	
C/C++ Remote Application 1	C/C++ Application:				
hello_world_imx8 Debug					-
cü C/C++ Unit		Variables	Search Project	Browse	
GDB Hardware Debugging	Build (if required) before launching				
GDB OpenOCD Debugging	Build Configuration: Select Automatically			~	
GDB PyOCD Debugging	O Enable auto build	O Disable auto b	uild		
GDB QEMU Debugging GDB SEGGER J-Link Debugging	Use workspace settings	Configure Worksp	ace Settings	3	
Launch Group			_		1
•	Connection: Local	V New	Edit	Properties	l
	Remote Absolute File Path for C/C++ Application	n:			Π
	C:/Users/ <username>/imx8-C-eclipse-workspa</username>	ce/hello_world_imx8/Debug/hello	o_world_imx8.elf	Browse	
	Commands to execute before application				
					1
	Skip download to target path.				1
	Using GDB (DSF) Automatic Remote Debugging other	g Launcher - <u>Select</u>	Revert	Apply	
ter matched 13 of 13 items	<u>ourer</u>				

- Figure 17. Debug configurations window
- In the Debug Configurations window, select C/C++ Remote Application as shown in Figure 17.
- 3. Under C/C++ Application, click Search Project.
- 4. Search for the project binary file hello world imx8, select it, and then click OK.
- 5. To connect to the board, perform the following steps:
 - a. From the **Connection** section, click **New**.
 - b. Select SSH.
 - A sample tab New Connection appears as shown in Figure 18.

-		— 🗆 X
New Conne	ection	
Specify prop	perties of a new connection	
Connection n	ame: debug_SSH	
Host inform		
Host:	18.125.5.252	
User:	root	
O Public ke	ey based authentication	Keys are set at <u>Network Connections, SSH2</u>
Passphrase:		
Passwore	d based authentication	
Password:		
 Advanced 		

Figure 18. New connection

- 6. In the New Connection tab, provide the details in the fields as given below:
 - Connection: Enter the desired connection name.
 - Host: Enter the host address. For example, 18.125.5.252

- User: Enter the user name. For example, root.
- Note: Select Password based authentication.
- Password: Enter password for the user name root.
- 7. Click Finish.

5.1.3.2 Debugging SSH configurations

To debug the SSH configurations, perform the following steps:

- 1. Navigate to the Debug Configuration window.
- 2. In the menu bar, navigate **Run > Debug Configurations**.
- In the Remote Absolute File Path for C/C++ Application text box, choose a name for the debug executable file to upload on the board. For example, debug_eclipse. <u>Figure 19</u> shows the Remote Absolute File Path section of the Debug Configurations window.

Connection: debug_SSH		~	New	Edit	Properties
Remote Absolute File Path for (C/C++ Application	:			
/home/root/debug_eclipse/he					Browse
Figure 19. Remote Ab	solute File I	Path			
 On the Debugger From the GDB de The debugger file see <u>Section 5.1.1</u>. 	bugger , clio	ck Browse to se	elect the deb	ugger file.	information,
Debug Configurations Create, manage, and run configuration	ations				×
▼ ● ● ● ▼ type filter text © C/C++ Application © C/C++ Attach to Application © C/C++ Attach to Application © C/C++ Perstoret Publication © Hello world imx8 Debug © C/C++ Nemote Application © Hello world imx8 Debug © C/C++ Nemote Application © Hello world imx8 Debug © C/C++ Unit © GDB Hardware Debugging © GDB PyenOCD Debugging © GDB SEGGER J-Link Debugging © GDB SEGGER J-Link Debugging © Launch Group	Stop on startup a Debugger Options Main Shared Lib GDB debugger: GDB command file (Warning: Some co Non-stop mode Enable Reverse l Force thread list Automatically d		Arm Embedded Toolchain e with the startup operatio ftware Reverse Debugging equires Multi Process GDB)	n of the debugger, for exar (detailed but slower)	Browse Browse 2 nple "run".)
Filter matched 13 of 13 items	Using GDB (DSF) Aut other	omatic Remote Debugging Laur	ncher - <u>Select</u>	Revert	Apply
0				Debug	Close

Figure 20. Debugger configuration

Note: In the above example, the full path is given as C: \Program Files

(*x*86) \GNU Arm Embedded Toolchain \10 2021.10 \bin \arm-none-eabiqdb.exe. A similar path is available if the toolchain is installed in its default location.

- 6. Remove the contents from the field GDB command file.
- 7. Click Debug.

Note: If you are connecting for the first time to the board, then a pop-up message regarding the authenticity of the IP host appears.

- 8. To continue connecting, click **Yes**. A file debug_eclipse appears on the board console. An error shown in <u>Figure 21</u> displays on the Eclipse UI. **GDB version error information**:
 - Occurs due to the GDB cross-compiler and does not affect the running and debugging of the program.
 - Does not have many functionalities and it is similar to the error which occurs during compilation on Windows.
 - One of the lacking functionalities of this error is its version which is reported in the Windows compiler command.

	Pro	blem Occurred		-		×			
	8	'Launching hello_wo Could not determir C:\Users\ <username> arm-10.3-2021.07- in\aarch64-none-lir</username>	e GDB version usin \Documents\AN_E mingw-w64-i686-a	ig commi clipse_VS arch64-n version	and: \Toolcha	in\gcc-			
			ON		couns //				
Figure 21	. GDB	version error							
the Ec	tipse co the Rur	ror and click OK to console. It depicts the button to run the d ninated > hello_world	e board console thr ebug executable fil Debug [C/C++ Kemo	rough SS le. Ite Applica	SH. ationj C:\U				
	root@imx8mmevk:~# /home/root/debug_eclipse;exit Hello, World! logout								
•		ing output							
	eck whe lowing	ether the file is succe ways:	essfully displayed o	on the bo	ard cons	ole, use any of			
	-	bug version of the e	xecutable file on th	ne board.					
	 Check whether the file is cross-compiled for the i.MX 8M platform using the following commands: 								
\$		debug_eclipse ug eclipse							

GNU/Linux 3.7.0, with debug_info x8mmevk:∼# ./debug_eclipse World! x8mmevk:∼# ∎	e, ARM aarch64, version 1 (SYSV), dynamically , not stripped on check on the board console	linked, interpreter /lib/ld-linux-aarch6
	tered upon running from Eclip igs tab as shown in <u>Figure 24</u>	
Create, manage, and run configura	tions	- L ^
♥ ● ● type filter text © C/C++ Attach to Application © C/C++ Attach to Application © C/C++ Attach to Application © C/C++ Postmortern Debugger © C/C++ Postmortern Debugger © C/C++ Notice Application © DBA gendro Debugging © GDB Tandware Debugging © GDB SteeGRA Link Debugging © GDB SteGRA Link Debugging © GDB SteGRA Link Debugging © GDB SteGRA Link Debugging	Name: hello world jms8 Debug ■ Main in Arguments © Folunger IV Source © Common Stop on startup at main Debugger Options Main Shared Libraries Gdbserver Settings Gdbserver path: C:\MinGWbingdbserver Port number: 2345 Gdbserver options: ☐ Remote timeout (seconds):	
Filter matched 13 of 13 items	Using GDB (DSF) Automatic Remote Debugging Launcher - <u>Select</u> other	Revert Apply
∞ e 24. Full gdbserver pa		Debug Close

5.2 Configuring IDE using Python language

This section describes the steps to install Eclipse IDE and set up Python. In addition, it describes the steps to transfer a file directly from the Eclipse IDE to a remote system.

5.2.1 Installing Eclipse for Java and Python setup

To install Eclipse and configure Python, perform the following steps:

- 1. Download Eclipse for Java and open it. For example, the Eclipse version 10.2021 is used in this context.
- Create a workspace or use a specific workspace. For example, the workspace imx8-python-eclipse-workspace is used in this context.

Visual Studio and Eclipse Debugging for i.MX 8M Family

		Eclipse IDE Launcher	×	
		Select a directory as workspace		
		Eclipse IDE uses the workspace directory to store its preferences and c	levelopment artifacts.	
				-
		Workspace: C:\Users\myuser\eclipse-workspace ~	Browse	
		Recent Workspaces		
		Copy Settings		
		0	aunch Cancel	
Fig	gure 25	5. Workspace selection		1
	-	the Menus bar of Eclipse, navigate through He	p > Eclipse Marketpl	ace.
	•	olug-ins for Python development and Remote Sy ocation.	stems Explorer are ad	ded from
4.		gate to the Search tab.		
	•	Find search bar, enter pydev , and click Go .		
6.	Install	Il the pydev package with all of its components.		
	Figure	e 26 shows the installed Python IDE for Eclipse		
	Clipse	se Marketplace	—	
	Eclipse N	Marketplace		
		olutions to install. Press Install Now to proceed with installation. ne "more info" link to learn more about a solution.		-
	Search Rec	ecent Popular Favorites Installed 💡 Giving IoT an Edge		
	Find: 🎤 p	pydev ×	All Markets V All Categories	∽ Go
		PyDev - Python IDE for Eclipse 9.3.0		^
	PyDev	$^{\rm P}$ PyDev is a plugin that enables Eclipse to be used as a Python IDE (supporting also type inference techniques which allow $\underline{more\ info}$	y Jython and IronPython). It uses advand	ced
		by <u>Brainwy Software</u> , EPL <u>IDE Python Aptana Pydev Django</u>		
	* 2082		Ins	talled
		Live Coding in Python 2.25.0		
		Run your Python code while you type. Displays variable state or turtle graphics th PyDev, so install PyDev first. <u>more info</u>	at are updated as you type your code. F	Requires
		by <u>Don Kirkby</u> , MIT		
	* 42	Python Minstalls: 19.3K (111 last month)		Install
		Python :: CodeMix 3 Cl 2020.7.22		、
	Marke	tetplaces		
)	•		
	?	< Back Install I	Now > Finish Ca	incel
Fig	gure 26	6. Eclipse marketplace for PyDev		

- 7. After installation, restart Eclipse if prompted and navigate to the Eclipse UI.
- 8. For Remote Systems Explorer, navigate through **Help > Install New Software**.

- 9. From the Work with field, select All Available Sites.
- 10. Type remote system as the filter text to narrow down the list of sites.
- 11. Under the **General Purpose Tools** options, select **Remote System Explorer User Actions** to install.
- 12. Click Next.

Vork with:All Available Sites	~	Add	Manage
remote system			× Select All
Name	Version		Deselect All
 General Purpose Tools 			
🗌 🕸 Remote System Explorer End-User Runtime	4.5.300.202101081641		
🗹 🕸 Remote System Explorer User Actions	4.5.300.202101081641		
Linux Tools			
Mobile and Device Development			
Programming Languages			

Figure 27. Remote System Explorer

- 13. After completion of the installation, restart Eclipse if prompted.
 - To proceed further, ensure that the following requirements are met for the board:
 - The board is connected to the PC through an Ethernet cable.
 - The board has an IP address as described in the <u>PC board connection</u> section of this document.
- 14. Click the Eclipse menu bar.
- 15. Navigate to **Window > Open Perspective > Other... > Remote System Explorer** to open the RSE perspective.
- 16. Right-click the **Remote Systems** tab and select **New Connection** as shown in Figure 28.

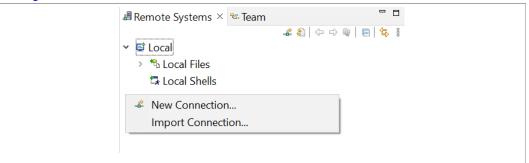


Figure 28. RSE new connection

- 17. Select SSH Only and click Next.
- Complete the newly opened window by providing the board IP address. In <u>Figure 29</u>, the IP address 18.125.2.252 is provided in the **Host Name** field.
- 19. Provide a desired name for the newly created connection in the **Connection name** field as shown in <u>Figure 29</u>.

Visual Studio and Eclipse Debugging for i.MX 8M Family

New Connection				×
Remote SSH Only	ystem Connection			
Define connection inf	ormation			
Parent profile:				×
Host name:	18.125.2.252			\sim
Connection name:	imx8mmini			
Description:				
✓ Verify host name				
Configure proxy settin	gs			
(?)	De de Martes	Charles I.	C	
Ø	< Back Next >	Finish	Cancel	

Figure 29. New connection example

- 20. Click **Finish** and proceed with the next steps.
- 21. From the **Remote Systems** tab, navigate to **Sftp Files > Root**.
- 22. In the pop-up window, provide the login credentials for the board. The default credentials are:
 - User ID: root
 - Password: There is no password.

¢	Enter Password	×
H C		SSH Only 18.125.5.252 18.125.5.252 root
P	Password (optional):	Save user ID
Figure 30. Pop-up login win	OK	Cancel

23. Click **OK**.

You can view that the board is connected successfully from Eclipse using SSH. The file system is available from Eclipse through SFTP.

- 24. To create a folder on the board from Eclipse console, navigate to home > root.
- 25. Right-click the **root** folder.

v 🍰				
	🗅 bin			
>	🗅 boot			
>	🗀 dev			
>	🗀 etc			
~	🖻 home	New	>	File
	> 🗀 root	Go To	>	, 🗳 Folder
Figure 31. Ne				
-		to the folder and click	-	
27. In the new	working all	ectory, create your fire	St Python file he	ello.py.
		Y ≥ home		
		∽ 🗁 root		
		Y 🗁 pyth		
		₽ he	ello.py	
Figure 32. Ne	w *.py file			
-	-	•	lanual Config	as shown in <u>Figure 33</u> .
🛷 🔻 🕌 🔻 🍋				
≧ hello × 1	🔵 Default Ec	ipse preferences for PyDev		
1	Uncheck setti	ngs that should not be change	he	
		ngoing build before launching		
		ason: To launch a file in PyDev		4
	neo			
		te that JDT/CDT may require a leave it unchecked if you use		_
Python	not configured			×
D lt	seems that the	Python interpreter is not curre	ntly configured.	
н	low do you wan	t to proceed?		
	Manua	Config Config first in PA	TH Choose from li	st Don't ask again
	✓ Re-check	whenever a PyDev editor is op	ened?	
<				
Remote System	1		ОК	Cancel
Figure 33. Ma	anual config			

Note: Configure the Python interpreter to debug the Python scripts in Eclipse.

type filter text	Python Interpre	ters		<p 8<="" th="" ⇒="" ▼="" ♥=""></p>	
> Help ^	Python interpreters	(e.g.: python.exe, pypy.exe). Double-clic	ck to rename.		
> Install/Update > Java	Name	Location		New	
> Language Sen				⊕ Up	
> Maven > Oomph				Down	
✓ PyDev				Set as Default	
Builders				× Remove	
> Debug > Editor				Config Conda	
 > Interactive ✓ Interpreters 	🖶 Packages 🛋 Li	braries Forced Builtins Predefined 🏾 Er	vironment • String Substitu		
IronPytł	Library	Version	Manag	Manage with pip	
Jython I Python			Manage	with conda	
Logging			Manage	Manage with pipenv	
PyUnit Run Scripting P Task Tags > Remote Syster > Run/Debug > Run/Debug > Terminal > TextMate > Version Contro			☐ Load conda e	nv vars before run?	
> XML (Wild We 🗸			Restore Defaults	Apply	
			Nestore Delauits	Арріу	

Figure 34. Python Interpreter main window

- Install the Python interpreter on your host, if it is not installed.
 To check whether the Python interpreter is already installed on your host, perform the following steps:
 - a. Open Windows PowerShell
 - b. In the cmd window, type python as shown in Figure 35.

Figure 35. Python version check

Note: If Python is not found, then install it.

- 30. If Python is installed on your host, continue the below steps.
- 31. Locate where Python is installed using the following command in the python prompt:

>>> import sys; print(sys.exec_prefix)

The output should contain a path as shown in Figure 36.

>>> import sys;	<pre>print(sys.exec_prefix)</pre>
C:\Users\	\Anaconda3

Figure 36. Python path

- 32. Navigate to the Eclipse console.
- 33. From the Preferences menu, navigate through New > Choose from Conda. Note: For other Python interpreters, select the appropriate Choose from... setting. In this example Figure 37, the interpreter used is Anaconda3.

Visual Studio and Eclipse Debugging for i.MX 8M Family

	\blacksquare Conda interpreter selection $ \Box$ \times
	Select an intepreter from the list.
	Anaconda3 C:\Users\NX
	Cancel
gure 37. Interp	reter selection
Click OK .	the default DuDa y preferences as shown in Figure 29
Do not change	e the default PyDev preferences as shown in <u>Figure 38</u> .
	Default Eclipse preferences for PyDev
	Uncheck settings that should not be changed.
	☑ Wait for ongoing build before launching: never
	Reason: To launch a file in PyDev a build is not required.
	Note that JDT/CDT may require a build to compile before launching, so, leave it unchecked if you use one of those plugins and want to automatically build before launching.
	Build (if required) before launching: no
	Reason: To launch a file in PyDev a build is not required.
	Note that JDT/CDT may require a build to compile before launching, so, leave it unchecked if you use one of those plugins and want to automatically build before launching.
	Re-check whenever a PyDev editor is opened?
	OK Cancel
Eigure 38 Defau	It PyDey preferences

36. Click **OK**.

5.2.2 Sample program

This section describes how to write a sample Hello, World! program, which includes multiple prints, and demonstrate debugging. For more details, refer to <u>Section 5.2.2.2</u>.

To write a sample Hello, World! program, perform the following steps:

1. Navigate to the Remote System Explorer perspective.

If the perspective is not already set, then set the perspective by navigating from the menu bar through Window > Perspective > Open Perspective > Other... > Remote System Explorer.

		🚚 Remote Systems × 📽 Team 🖓 🗖			
		🚜 🌒 (()) 🚳 🚍 😫 💈			
		✓ □ 18.125.5.252			
		✓ [®] Sftp Files			
		> ಾ My Home ∽ ಾ Root			
		 ✓ ♣ / 			
		> 🗀 bin			
		> 🗅 boot			
		> 🗀 dev			
		> 🗀 etc			
		Y 🗁 home			
		✓ ➢ root			
		😂 python_eclipse			
		> 🗀 weston			
		> 🗀 lib			
		> 🗅 lost+found			
		> 🗀 media			
		> Commt			
		> 🗅 opt			
		> C proc			
		> 🗀 run > 🗅 sbin			
		> 🖆 STV 🗸			
Figure 3	9. Remote S	vstems tab			
		-			
-		ated directory and navigate to New > File .			
The	below window	v appears:			
	A 11 EI				
	le New File	$ \Box$ X			
	Remote File				
	Create a New File				
	Connection name				
	Connection name	: 18.125.5.252			
	Parent folder: /home/root/python_eclipse				
	New file name:	hello_world.py			
	?				
		Finish Cancel			
Figure 4	0. Remote file	e creation			
L		the example file.			
		. py extension is included at the end of the filename.			
An e	xample code	is shown below:			
	•				
		🖻 hello_world ×			
		1 print("Hello, World!")			
		2 print("Hi, Python!")			
Figure 4	1. Hello, Wor	ld! example			
		ble with multiple prints suffices.			
Note	: Anvexame				

In the RSE tab, the python eclipse directory is created.

5.2.2.1 Running the Eclipse configurations

This section explains the process to run the configurations.

To set a run configuration, perform the following steps:

 Click the Eclipse menu bar and navigate to Run > Run Configurations. A sample running configuration is shown in <u>Figure 42</u>.

ype filter text	🗟 Main 🏁 Arguments 🕏 Interpreter 🤣 Refresh 📼 Environment 💷 Common				
🗬 Gradle Task	Project				
	RemoteSystemsTempFiles	Browse			
e IronPython uni	Main Module				
🜌 Java Applet 亚 Java Applicatio	\${workspace_loc:RemoteSystemsTempFiles/18.125.5.252/home/root/python_eclipse/hello_world.py}	Browse			
	PYTHONPATH that will be used in the run:				

Figure 42. Setting a run configuration

- 2. Enter a desired name for the configuration. For example, conf-hello (outline 1).
- 3. In the **Project** section, browse for RemoteSystemsTempFiles (outline 2).
- In the Main Module section, browse for the location of the hello_world.py file (outline 3).
- 5. Click Apply and Run.

The file runs and prints the outputs in the **Console** tab as shown in Figure 43.

```
Tasks Remote Shell Console ×
<terminated> hello_world.py [C:\Users\<Username>\Anaconda3\python.exe]
Hello, World!
Hi, Python!
```

Figure 43. Output example

5.2.2.2 Debugging

This section describes how to debug the configurations.

To debug the configurations, perform the following steps:

1. To set a breakpoint, double click the edge of the code window which is in front of the code line.

A green mark appears as shown in Figure 44.

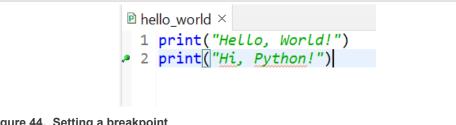


Figure 44. Setting a breakpoint

2. To start debugging, click the **Debug** icon.

Note: The same configuration that was used on running the program launches for debugging.

The following message appears which switches the perspective automatically to **Debug Perspective**.

	Confirm Perspective Switch					
	This kind of launch is configured to open the Debug perspective when it suspends.					
	This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.					
	Switch to this perspective?					
	Remember my decision					
	Switch No					
Figure 45.	Perspective switch					
3. Select	Switch.					
If the m	nessage is not displayed, then change the perspective manually from	Window				
		Willaow				
> Fers	pective > Open Perspective > > Other > Debug.					
1 pr	world × Constraints and the second se					
<						
	ole × 🗈 Problems 🗵 Debug Shell 💁 Remote Shell					
pydev	orld.py [debug] [C:\Users\ <username> \Anaconda3\python.exe] debugger: starting (pid: 19884) , World!</username>					
Figure 46.	Debugging steps					

4. To step over to the next breakpoint, press F6.

6 Configuring Visual Studio

This section describes the steps to download and install Visual Studio from <u>Visual Studio</u> <u>2022 Community</u>. In addition, this section explains the steps to configure the remote SSH connection to the board.

6.1 Installing Visual Studio

- 1. Install Visual Studio using the link Visual Studio 2022 Community.
- 2. Navigate to the Visual Studio UI and select the hyperlink Continue without code.

Visual Studio and Eclipse Debugging for i.MX 8M Family

Visual St	tudio 2022			
Open recent		Get st	arted	
/ Today	٩	→	Clone a repository Get code from an online repository like or Azure DevOps	GitHub
Solution1	Lsin 3/28/2022 12:04 Username \source/repos\Solution1	м 🗗	Open a project or solution Open a local Visual Studio project or .sli	
		đ	Open a local folder Navigate and edit code within any folde	я
		ئ	Create a new project Choose a project template with code so to get started	affolding
			Continue without code \rightarrow	
Figure 47. Visual S				
	he menu bar to Tools > (
	and Mobile, select the D			
	lsets, select Linux and (levelopment wit	h C++.
	o install the selected pac s to apply the changes.	kages.		
	esktop & Mobile (5)			
	Mobile development with .NET Build cross-platform applications for iOS, Android or Windows using Xamarin. This includes a preview of the	NET desktop developm Build WPF, Windows For using C#, Visual Basic, ar	ent	
	Desktop development with C++ Build modern C++ apps for Windows using tools of your choice, including MSVC, Cleng, CMeke, or MSBuild.	Universal Windows Plat Create applications for t with C#, VB, or optional	he Universal Windows Platform	
	*+ Mobile development with C++ Build cross-platform applications for IOS, Android or Windows using C++.			
Figure 48. Toolsets	selection			
Other Te	oolsets (5)			
	Data storage and processing Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.	Data science and and Languages and toolir applications, includin	ng for creating data science	
œ	Visual Studio extension development	Office/SharePoint de Create Office and Sha solutions, and VSTO a	velopment arePoint add-ins, SharePoint add-ins using C#, VB, and JavaScript.	
Δ	Linux and embedded development with C++ Create and debug applications running in a Linux environment or on an embedded device.			
-				

Figure 49. Toolsets selection

- 6. After restart, navigate from the menu bar to **Tools > Options**.
- 7. From the new dialog box, select **Cross Platform > Connection Manager**.
- In the Connection Manager dialog box, choose the Add button as shown in Figure 50.

Visual Studio and Eclipse Debugging for i.MX 8M Family

Options		? ×
Search Options (Ctrl+E)	Add or Remove SSH connections to Remote Systems like Linux, Ma	c or
▷ Environment	Windows:	0.01
▷ Projects and Solutions		
▷ Source Control	Default Host Name Port User Name OS	Add
▷ Work Items		
▷ Text Editor		Edit
▷ Debugging) (::f
▷ CMake		Verify
▲ Cross Platform		Remove
 Connection Manager 		
Remote Headers IntelliSense Manaç		
Logging and Diagnostics		
▷ Database Tools		
▷ IntelliCode		
▷ Live Share		
NuGet Package Manager		
Web Forms Designer		
Windows Forms Designer	The connections added here can be used later for build or debuggir	ng, or in
< >	projects that use remote builds.	
	ОК	Cancel
	- OK	currect
e 50. Connection manager		
the following window, enter	the prompted details:	
-		5 5 252
	Idress of the board. For example, 18.12	5.5.252.
Port: Enter the port number	r. For example, 22.	
Iser name: Enter the user	name. For example, root.	

- Authentication type: Select the type of authentication. For example, Password.
- **Password**: Enter the password for the user name root.

An example is shown in Figure 51.

Connect to	Remote System	
Connect to	Kemole System	
	ect to Linux, Mac, Windows, or other system Ided here can be used later for build or deb e builds.	
Host name:	18.125.5.252	
Port:	22	
User name:	root	
Authentication type:	Password	
Password:		
	Connect	Cancel

Figure 51. Connect to remote system

- 10. Select **Connect**. Now you have a remote SSH connection to the board.
- 11. Optionally, set the board connection as **Default** by selecting the radio button as shown in <u>Figure 52</u>.

Defau	t Host Name	Port	User Name	OS
 Denue 		22	root	Unknown (A
<				>
52. Default board co	nnection			

6.2 Sample program

This section explains how to create a sample Hello, World! program in C/C++.

To create a project in C, perform the following steps:

- Navigate to File > New > Project. Ensure that you select Linux instead of All Platforms.
- 2. Select Empty Project.
- 3. Enter a desired name to the project. For example, my_first_project.

Create a new	v project		- م	Clear all
Recent project templa	tes	All languages 🔹 Linux	 All project 	ct types 🔹
🕌 Empty Project			apps that don't depend on .sln or .v	rcxproj files.
Ja Blank Solution		Console Application Run code in a Linux terminal. Prints C++ Linux Console		
		Empty Project Start from scratch with C++ for Lin C++ Linux Console	ux. Provides no starting files.	
		A blinking LED app using WiringPi C++ Linux IoT Consol		
		Makefile Project Bring your own build system to co C++ Linux Library Cor	ompile C++ for Linux. nsole Desktop	
				Next

Figure 53. New C/C++ Project

4. To add files to the new project, navigate to the **Solution Explorer** window as shown in Figure 54.

Visual Studio and Eclipse Debugging for i.MX 8M Family

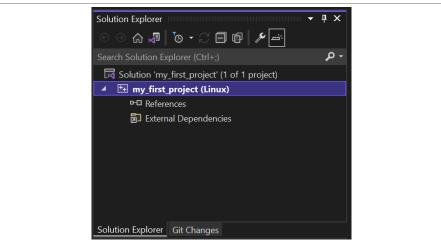
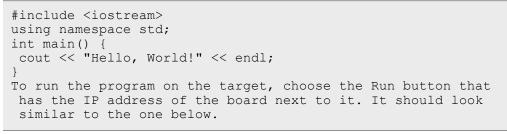


Figure 54. Solution explorer window

- Right-click the new project. For example, my_first_project is used in this context.
- 6. Write a simple Hello, World! program in C++ as shown below:



 To debug on the board, include breakpoints through the program. For example, a breakpoint is included before the message output as shown in <u>Figure 55</u>.

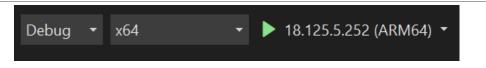


Figure 55. Platform-specific Run

 Click the **Run** button again. The debug pop-up window appears as shown in <u>Figure 56</u>.

Visual Studio and Eclipse Debugging for i.MX 8M Family

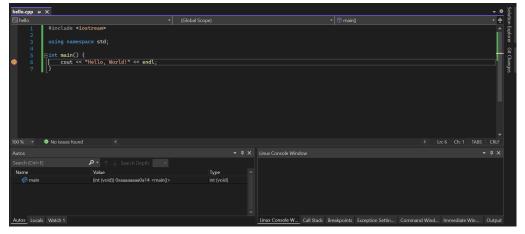


Figure 56. VS debugging view

For C development, use the steps from Step 1 to Step 8.
 Ensure that the source file extensions are changed to *.c.
 A sample C program Hello, World! is given below:

```
#include <stdio.h>
int main()
{
    printf("Hello, World!");
    return 0;
}
```

7 References

For more information on I.MX Devices, Python scripting, and Eclipse setup, refer to the following list of documents:

- i.MX Yocto Project User's Guide
- •
- i.MX Linux User's Guide
- Developing an Application for the i.MX Devices on the Linux Platform
- Raspberry pi development setup on Eclipse [Linux] and Debugging via SSH
- Remote development of Python scripts on Raspberry Pi with Eclipse

Revision history

<u>Table 3</u> summarizes the changes done in this document from the initial release.

Table 3. Revision History

Revision number	Date	Substantive changes	
0	11 November 2022	Initial release	

AN13739 Application note

Visual Studio and Eclipse Debugging for i.MX 8M Family

8 Legal information

8.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

8.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at <u>PSIRT@nxp.com</u>) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

8.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

i.MX — is a trademark of NXP B.V.

Visual Studio and Eclipse Debugging for i.MX 8M Family

Contents

1	Introduction	2
2	Prerequisites	2
2.1	Hardware	
2.2	Software	2
3	Yocto project	3
3.1	Customizing Linux image	3
3.2	Modifying Yocto file	3
4	Configuring LAN	5
4.1	Host PC configuration	5
4.2	Board configuration	7
5	Configuring Eclipse environments	8
5.1	Configuring IDE using C language	8
5.1.1	Installing Toolchain	
5.1.2	Installing Eclipse IDE	8
5.1.2.1	Workspace selection	9
5.1.3	Sample program	
5.1.3.1	Debugging image configurations	13
5.1.3.2	Debugging SSH configurations	
5.2	Configuring IDE using Python language	
5.2.1	Installing Eclipse for Java and Python setup .	
5.2.2	Sample program	
5.2.2.1	Running the Eclipse configurations	
5.2.2.2	Debugging	25
6	Configuring Visual Studio	26
6.1	Installing Visual Studio	26
6.2	Sample program	
7	References	31
8	Legal information	32

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2022 NXP B.V.

All rights reserved.

For more information, please visit: http://www.nxp.com