

## 1 Introduction

The i.MX RT10xx series are processors featuring NXP's advanced implementation of the Arm® Cortex®-M7 core, which can provide good CPU performance and real-time response. One of the advantage of the Cortex-M7 core is that it contains Instrumentation Trace Macrocell (ITM) in the CoreSight architecture. ITM is a novel trace method for MCU debugging and it supports the Serial Wire Output (SWO) trace.

This application note introduces the principle and advantages of SWO trace. It provides guidance on how to enable the SWO trace with i.MX RT10xx Series, including the hardware and software configurations. Also this document describes the steps for using the basic SWO trace feature (ITM debug output) based on different IDEs.

## 2 SWO trace overview

### 2.1 CoreSight components

CoreSight is an architecture used to debug and trace by Arm. The architecture consists of several useful components. [Figure 1](#) shows a typical CoreSight architecture, which includes two Arm processors, one DSP module and a number of other components. There are three paths in the below environment: trace path, debug path and trigger path. This document focuses on the debug path, which includes ITM module and SWO module.

### Contents

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
<b>2</b>	<b>SWO trace overview.....</b>	<b>1</b>
2.1	CoreSight components.....	1
2.2	Serial Wire Output (SWO).....	2
2.3	Instrumentation Trace Macrocell (ITM).....	4
2.4	Debug probes.....	4
<b>3</b>	<b>Project and hardware modifications required to enable SWO Trace.....</b>	<b>5</b>
3.1	MIMXRT1010-EVK.....	5
3.2	MIMXRT1020-EVK.....	6
3.3	MIMXRT1050-EVK.....	7
3.4	MIMXRT1060-EVK.....	7
3.5	MIMXRT1060-EVKB.....	7
<b>4</b>	<b>Using SWO with IAR.....</b>	<b>8</b>
<b>5</b>	<b>Using SWO with Keil.....</b>	<b>11</b>
<b>6</b>	<b>Conclusion.....</b>	<b>15</b>
<b>7</b>	<b>References.....</b>	<b>16</b>
<b>8</b>	<b>Revision history.....</b>	<b>16</b>
	Legal information.....	17



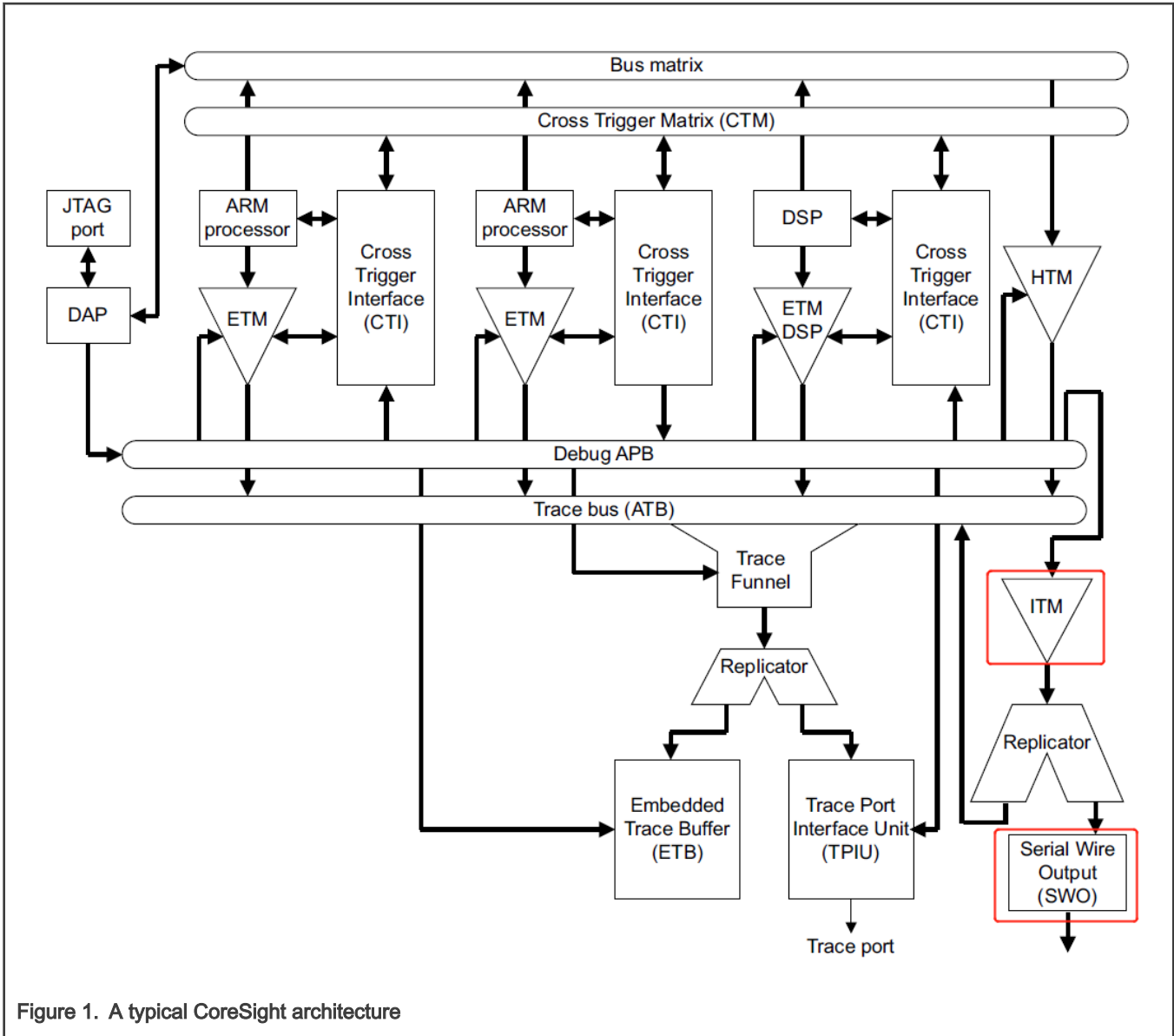


Figure 1. A typical CoreSight architecture

## 2.2 Serial Wire Output (SWO)

Serial Wire Output (SWO) is a single pin trace interface that is part of the Cortex-M CoreSight architecture. It acts as a bridge between the on-chip trace data to Trace Port Analyzer (TPA).

The SWO trace provides access to the memory of a running target without interrupting the target. It is available on Cortex-M3/M4/M7 based MCUs. Figure 2 shows the block diagram of SWO. In addition to the standard Serial Wire Debug (SWD) connection, SWO trace requires just one extra pin.

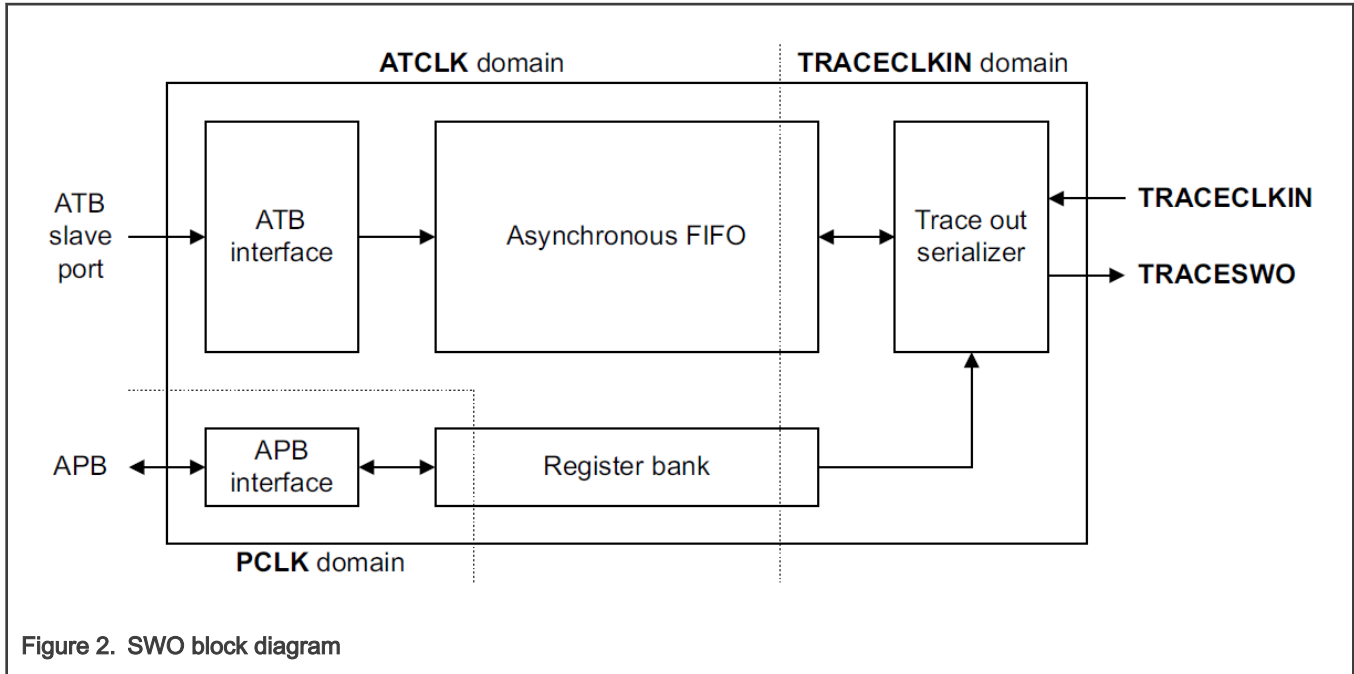


Figure 2. SWO block diagram

The SWO trace interface is a TPIU-like device that supports a limited subset of the full TPIU functionality. SWO trace features are described as below:

- Sending debug messages as strings
- Recording interrupt entry/exit
- Recording function entry/exit
- Periodic PC (Program Counter) value sampling
- Event notification
- Variable or memory cell change over time

One of the most common usage of SWO trace is using SWO to print debug messages from the target, as described in [Using SWO with IAR](#) and [Using SWO with Keil](#).

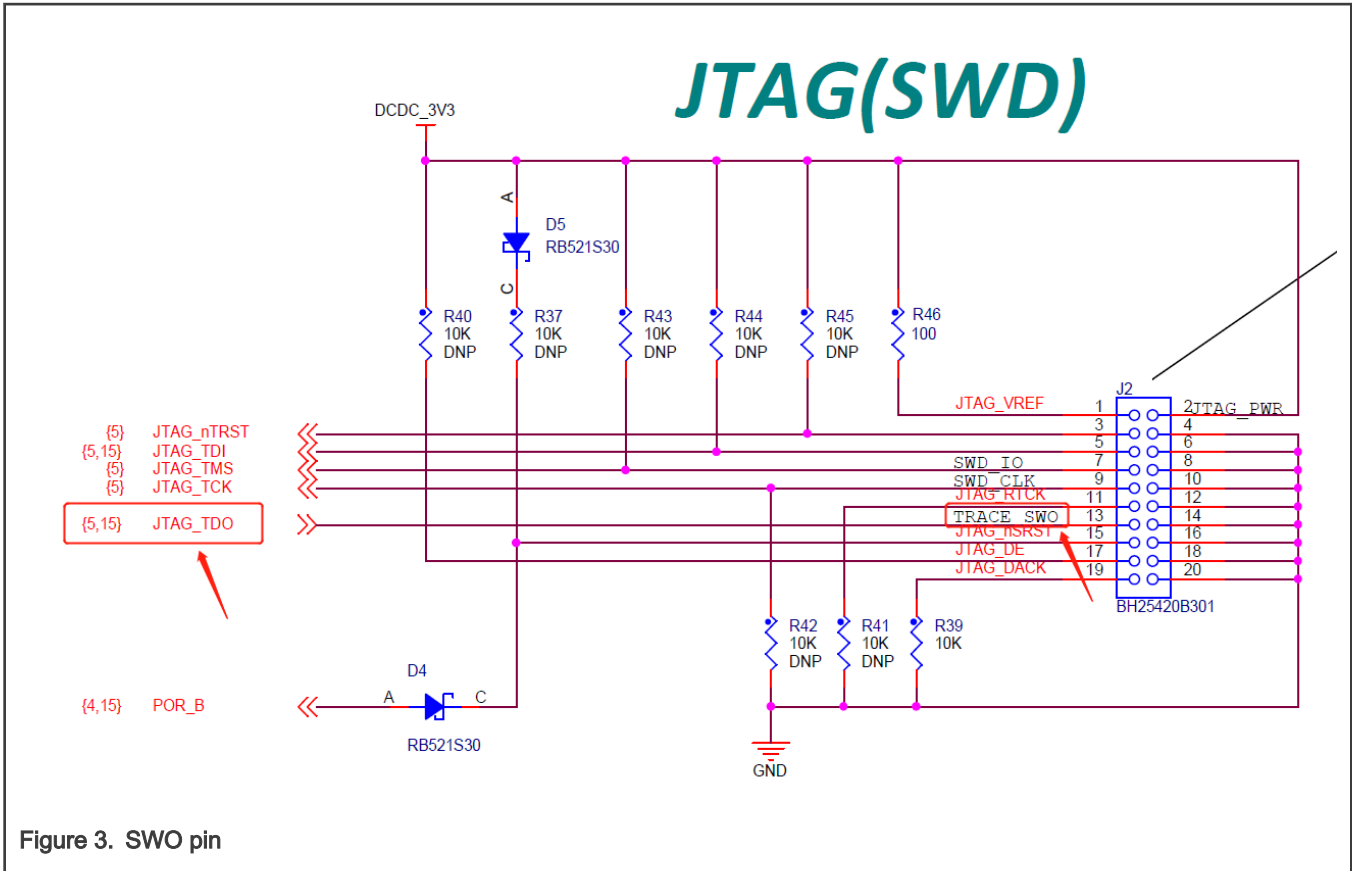


Figure 3. SWO pin

**NOTE**

SWO pin is multiplexed with JTAG\_TDO pin, as shown in Figure 3. This means the Serial Wire Debug (SWD) mode must be used, not the JTAG mode.

### 2.3 Instrumentation Trace Macrocell (ITM)

ITM generates trace information as packets and it provides a mechanism for sending data from the target to debugger via the SWO stream. Trace data from ITM is forwarded to the SWO interface and then SWO transports the data streams to the Trace Port Analyzer.

ITM supports 32 stimulus channels which allow different software to output data to different ports, so that the debugging target is able to separate different packets. Among the 32 ports, **Port0** is designated for outputting debug information. Different from the UART-based output, ITM output does not cause significant latency to the application, because there is a FIFO that buffers the output data inside ITM block.

ITM is used to:

- support printf style debugging
- trace OS and application events
- emit diagnostic system information

### 2.4 Debug probes

There are a number of debug solutions for i.MX RT10xx series MCUs, such as CMSIS DAP, J-Link and PE micro. However, not every debug solution supports SWO trace. The SWO trace function can be used with several common compatible debug probe such as:

- LinkServer LPC-Link2

LPC-Link2 is an extensible, stand-alone debug probe that can be configured to support various development tools and IDEs using a variety of different downloadable firmware images. It has also been built into many LPCXpresso development boards and some MIMXRT-EVK boards.

LinkServer debug connections allow SWO trace to be started and stopped while the target is running. That means the SWO trace using LPC-Link2 debug solution has no effect on the MCU performance. The powerful features of SWO trace have been tested with LPC-Link2 firmware based on MCUXpresso IDE. For more information, see [MCUXpresso IDE SWO Trace Guide](#).

- **SEGGER J-Link**

J-Link is a debug probe supporting a large number of MCU cores. Regarding to the SWO trace with J-Link debug solution, the target must be halted when starting or stopping the SWO trace.

The correct target clock speed must be sampled to allow the internal sampling rates to be calculated correctly. The steps for using SWO trace with J-Link debug probe based on different toolchains will be clarified explicitly in [Using SWO with IAR](#) and [Using SWO with Keil](#).

### 3 Project and hardware modifications required to enable SWO Trace

The default project examples in SDK and some original EVK boards do not enable SWO trace features. To enable the SWO trace, several project changes and hardware modifications are required. This chapter lists the modifications for enabling SWO trace based on the following i.MX RT platforms:

- MIMXRT1010-EVK
- MIMXRT1020-EVK
- MIMXRT1050-EVK
- MIMXRT1060-EVK
- MIMXRT1060-EVKB

#### NOTE

The source code changes are performed based on the latest SDK (version 2.9.1).

#### 3.1 MIMXRT1010-EVK

##### 1. Modifying `pin_mux.c`

Within a project, select the **board** subfolder, locate the `pin_mux.c` file and add the below lines in the `BOARD_InitPins` function:

```
IOMUXC_SetPinMux(          /* Add these lines*/
    IOMUXC_GPIO_AD_09_ARM_TRACE_SWO,
    0U);
IOMUXC_SetPinConfig(      /* Add these lines*/
    IOMUXC_GPIO_AD_09_ARM_TRACE_SWO,
    0x00F9U);
```

##### 2. Modifying `clock_config.c`

Within a project, select the **board** subfolder, locate the `clock_config.c` file and modify the lines in the `BOARD_BootClockRUN` function as below:

```
/* Disable TRACE clock gate. */          /* Locate these entries*/
CLOCK_DisableClock(kCLOCK_Trace);
/* Set TRACE_PODF. */
CLOCK_SetDiv(kCLOCK_TraceDiv, 3);
/* Set Trace clock source. */
CLOCK_SetMux(kCLOCK_TraceMux, 0);
```

```
CLOCK_EnableClock(kCLOCK_Trace); /* Make these edits*/
/* Set TRACE_PODF. */
CLOCK_SetDiv(kCLOCK_TraceDiv, 0); /* Make these edits*/
/* Set Trace clock source. */
CLOCK_SetMux(kCLOCK_TraceMux, 0); /* Make these edits*/
```

### 3.2 MIMXRT1020-EVK

1. Connect SWO pin and JTAG-TDO pin

Connect SWO (J19-3) and JTAG-TDO (J16-13).

2. DNP R116

JTAG-TDO pin is reused as a ENET-RESET signal. There is a capacitor (C137) that affects the signal, so remove R116, as shown in [Figure 4](#).

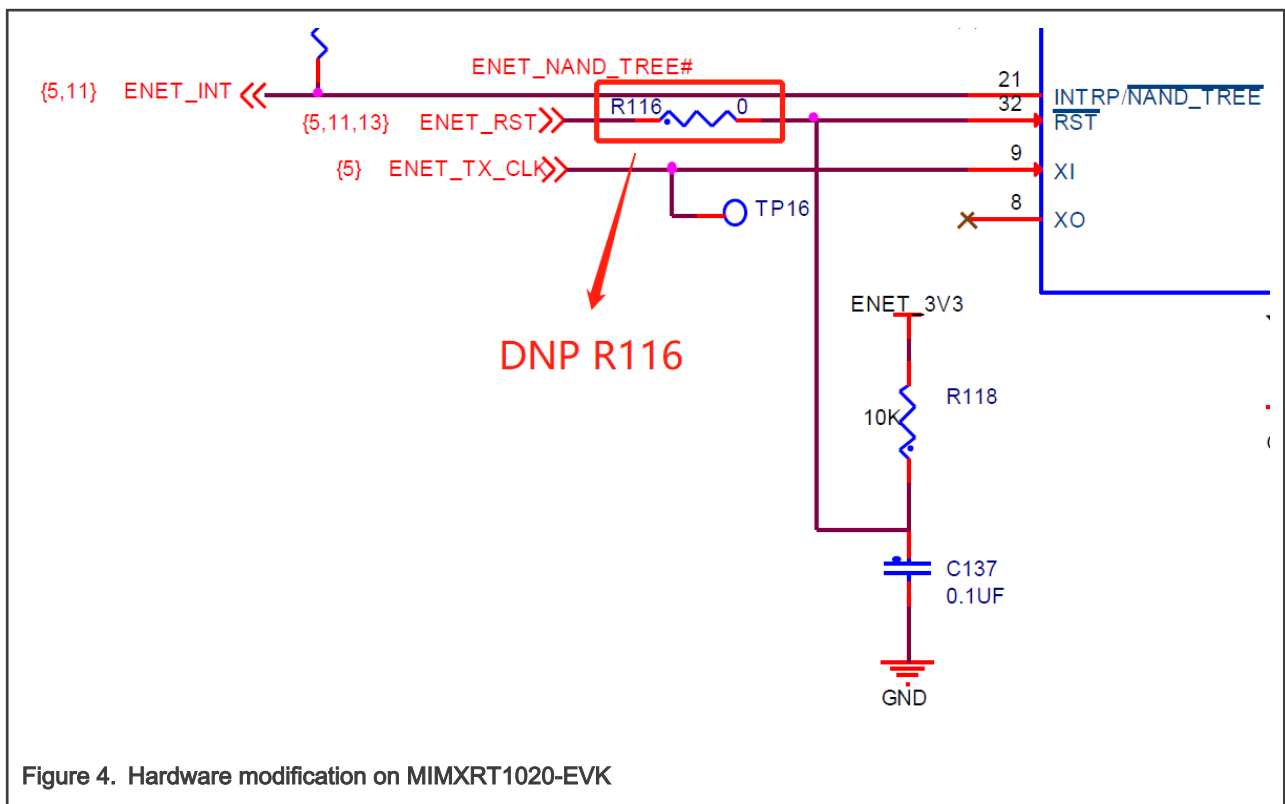


Figure 4. Hardware modification on MIMXRT1020-EVK

3. Modifying pin\_mux.c

Within a project, select the **board** subfolder, locate the `pin_mux.c` file and add the below lines in the `BOARD_InitPins` function:

```
IOMUXC_SetPinMux( /* Add these lines*/
    IOMUXC_GPIO_AD_B0_11_ARM_CM7_TRACE_SWO,
    0U);
IOMUXC_SetPinConfig( /* Add these lines*/
    IOMUXC_GPIO_AD_B0_11_ARM_CM7_TRACE_SWO,
    0x00F9U);
```

4. Modifying clock\_config.c

Same as the clock modifications of MIMXRT1010-EVK in [MIMXRT1010-EVK](#).

### 3.3 MIMXRT1050-EVK

1. Patch wire between R256 and pin 13 of J21.
2. Modifying `pin_mux.c`

Within a project, select the **board** subfolder, locate the `pin_mux.c` file and add the below lines in the `BOARD_InitPins` function:

```
IOMUXC_SetPinMux(          /* Add these lines*/
    IOMUXC_GPIO_B0_13_ARM_CM7_TRACE_SWO,
    0U);
IOMUXC_SetPinConfig(      /* Add these lines*/
    IOMUXC_GPIO_B0_13_ARM_CM7_TRACE_SWO,
    0x00F9U);
```

3. Modifying `clock_config.c`

Within a project, select the **board** subfolder, locate the `clock_config.c` file and modify the lines in `BOARD_BootClockRUN` function as below:

```
/* Disable TRACE clock gate. */          /* Locate these entries*/
CLOCK_DisableClock(kCLOCK_Trace);
/* Set TRACE_PODF. */
CLOCK_SetDiv(kCLOCK_TraceDiv, 3);
/* Set Trace clock source. */
CLOCK_SetMux(kCLOCK_TraceMux, 0);

CLOCK_EnableClock(kCLOCK_Trace);        /* Make these edits*/
/* Set TRACE_PODF. */
CLOCK_SetDiv(kCLOCK_TraceDiv, 0);        /* Make these edits*/
/* Set Trace clock source. */
CLOCK_SetMux(kCLOCK_TraceMux, 3);        /* Make these edits*/
```

### 3.4 MIMXRT1060-EVK

1. Modifying `pin_mux.c`

Within a project, select the **board** subfolder, locate the `pin_mux.c` file and add the below lines in the `BOARD_InitPins` function:

```
IOMUXC_SetPinMux(IOMUXC_GPIO_AD_B0_10_ARM_TRACE_SWO, 0U);
IOMUXC_SetPinConfig(IOMUXC_GPIO_AD_B0_10_ARM_TRACE_SWO,0x00F9U);
```

2. Modifying `clock_config.c`

Same as the clock modifications of MIMXRT1050-EVK in [MIMXRT1050-EVK](#).

### 3.5 MIMXRT1060-EVKB

1. DNP R173
2. Modifying `pin_mux.c`

Same as the pin modifications of MIMXRT1060-EVK in [MIMXRT1060-EVK](#).

3. Modifying `clock_config.c`

Same as the clock modifications of MIMXRT1060-EVK in [MIMXRT1060-EVK](#).

**NOTE**

The frequency of `TRACE_ROOT_CLK` should be close to `CORE_CLK`. Pay attention to the differences of clock configuration between different MCUs.

## 4 Using SWO with IAR

After making the hardware and project changes as described in [Project and hardware modifications required to enable SWO Trace](#), the SWO trace can be captured. However, to support the SWO trace function in different development environment, the IDE configurations are different. This chapter focuses on performing the basic SWO trace feature (ITM debug output) in IAR with J-Link debug probe. The configurations steps are as follows:

### 1. Preparations:

- Software: MIMXRT1060-EVK SDK release (version: 2.9.1)
- Hardware: MIMXRT1060-EVK board
- Debug probe: J-Link Plus
- IDE: IAR Embedded Workbench for Arm 8.50.9
- ITM debug output code:

```
while (1)
{
    ITM_SendChar('A'); /* Send character A via ITM to designate port*/
    ITM_SendChar('B');
    ITM_SendChar('C');
}
```

The default output port is **port0**. For more information, see the `ITM_SendChar()` function in `core_cm7.h`.

2. Click **Project -> Options -> General Options -> Library Configuration**. Set the library to **Full** and in **stdout/stderr**, choose **Via SWO**, as shown in [Figure 5](#).

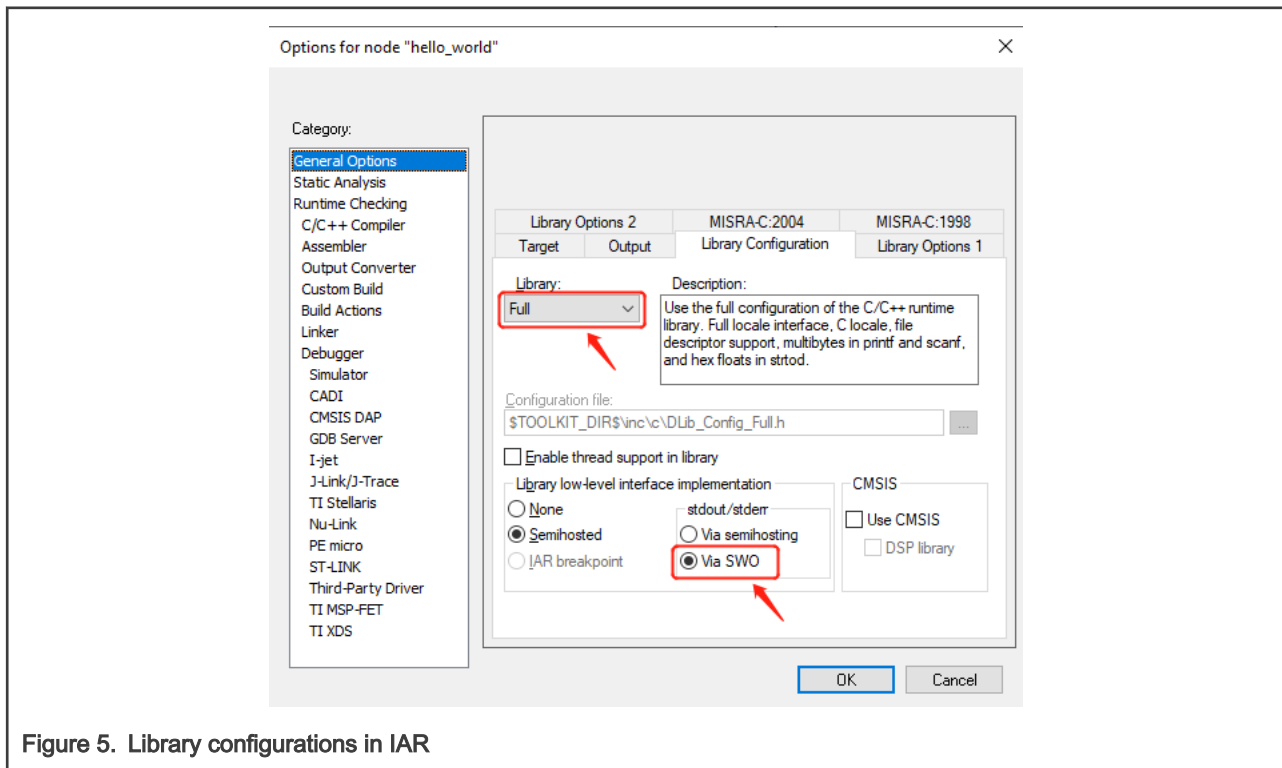


Figure 5. Library configurations in IAR



- Click **Project -> Options -> Debugger**. Set the driver to **J-Link/J-Trace**, as shown in [Figure 6](#).

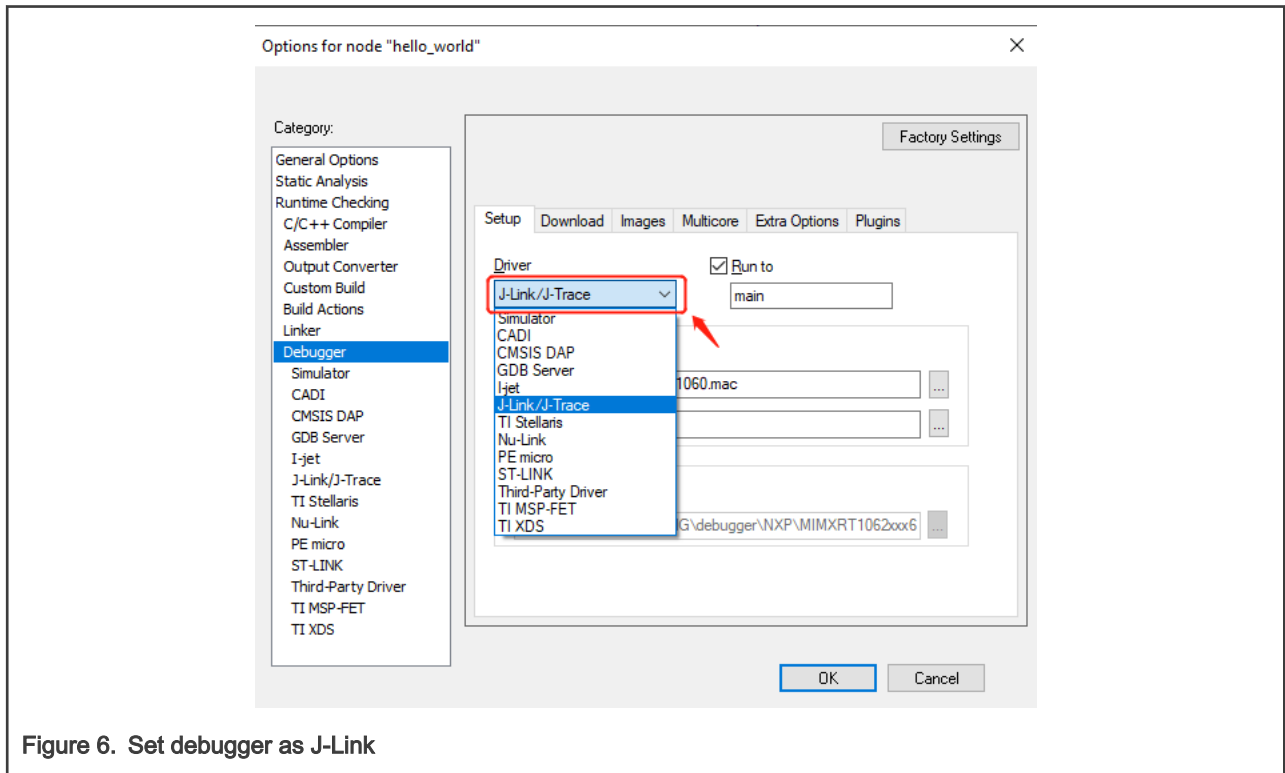


Figure 6. Set debugger as J-Link

- Click **Project -> Options -> Debugger -> J-Link/J-Trace**. In the **Setup** submenu, set the reset method to **Normal** and CPU clock to **600 MHz**. In the **Connection** submenu, set the interface to **SWD**, as shown in [Figure 7](#).

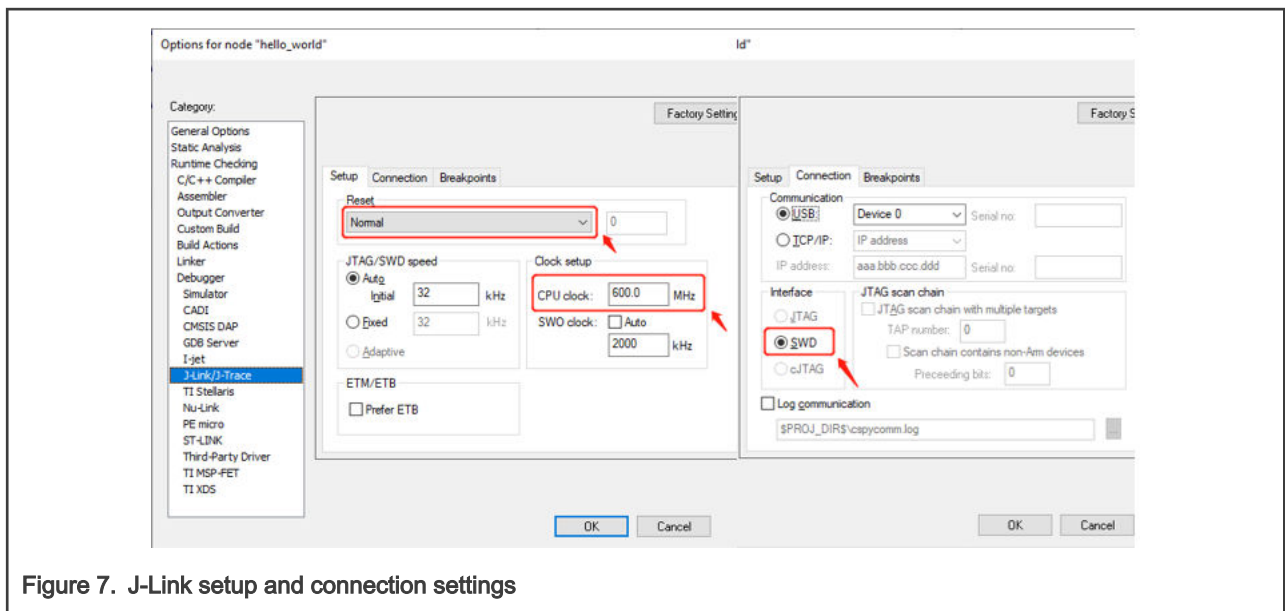


Figure 7. J-Link setup and connection settings

- Download and debug the project. Click **File Menu -> J-Link -> SWO Configuration**. Make configurations on the **SWO Trace Window Settings** window, as shown in [Figure 8](#) and [Figure 9](#).

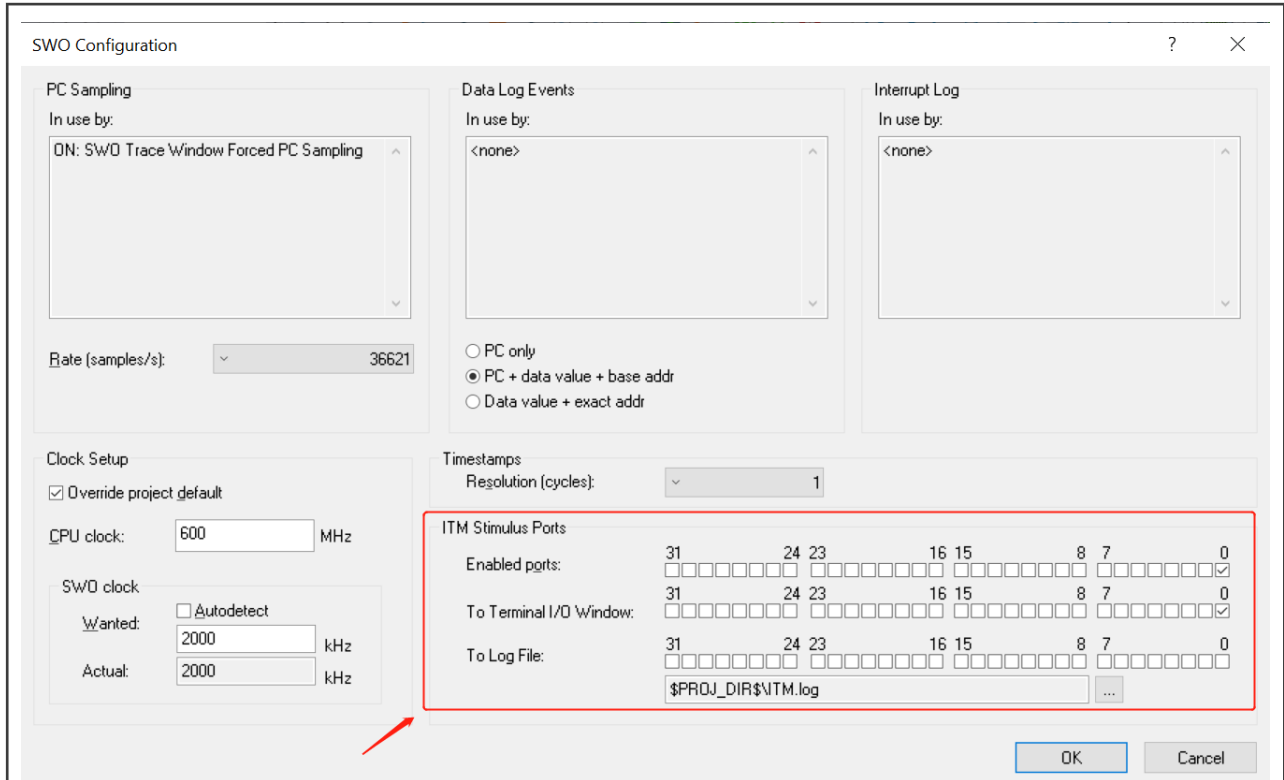


Figure 8. Set the ITM stimulus port to port0

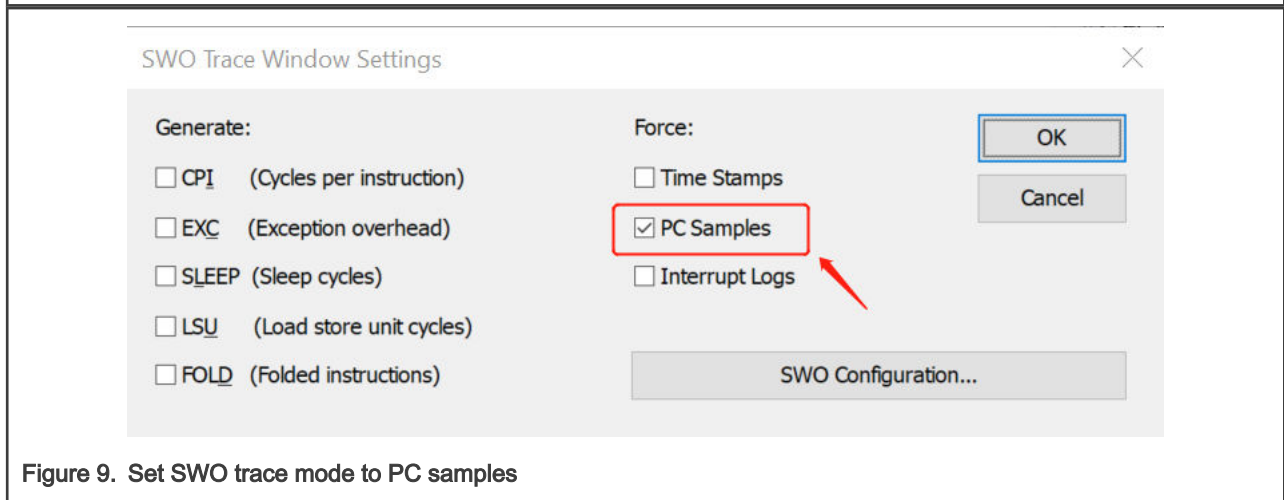



Figure 9. Set SWO trace mode to PC samples

6. Click **File Menu** -> **J-Link** -> **SWO trace** to open the SWO Trace panel. Click the  button to enable the trace.
7. Click **View** -> **Terminal I/O** to open the Terminal I/O window. Press **F5** to run the project. The output message is as shown in [Figure 10](#).



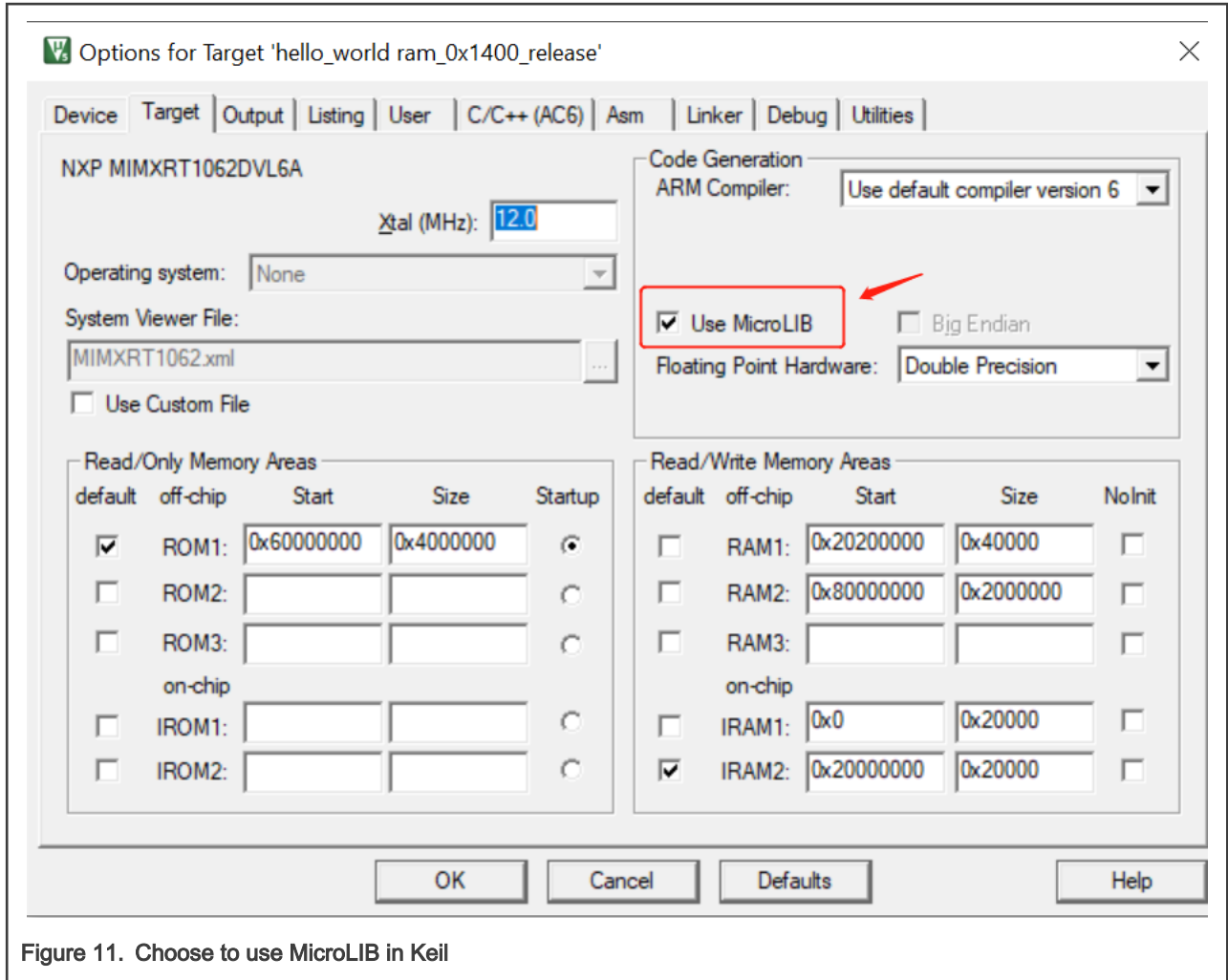


Figure 11. Choose to use MicroLIB in Keil

3. In the **Debug** submenu, set the debugger to **J-Link/J-TRACE Cortex**, as shown in [Figure 12](#).

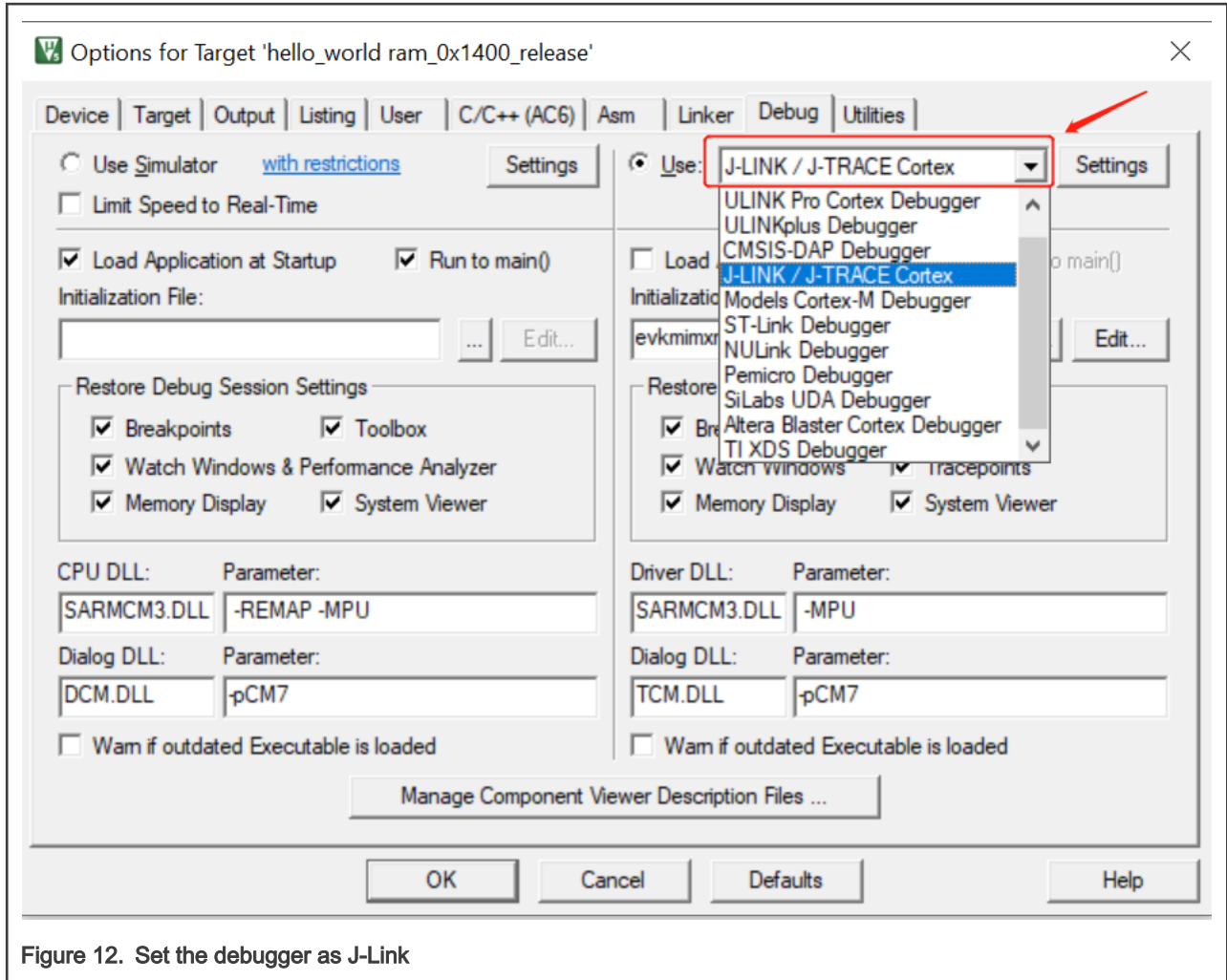


Figure 12. Set the debugger as J-Link

4. Click the **Settings** button. On the J-Link setup menu, set the port to **SW** and the reset method to **Normal**, as shown in [Figure 13](#).

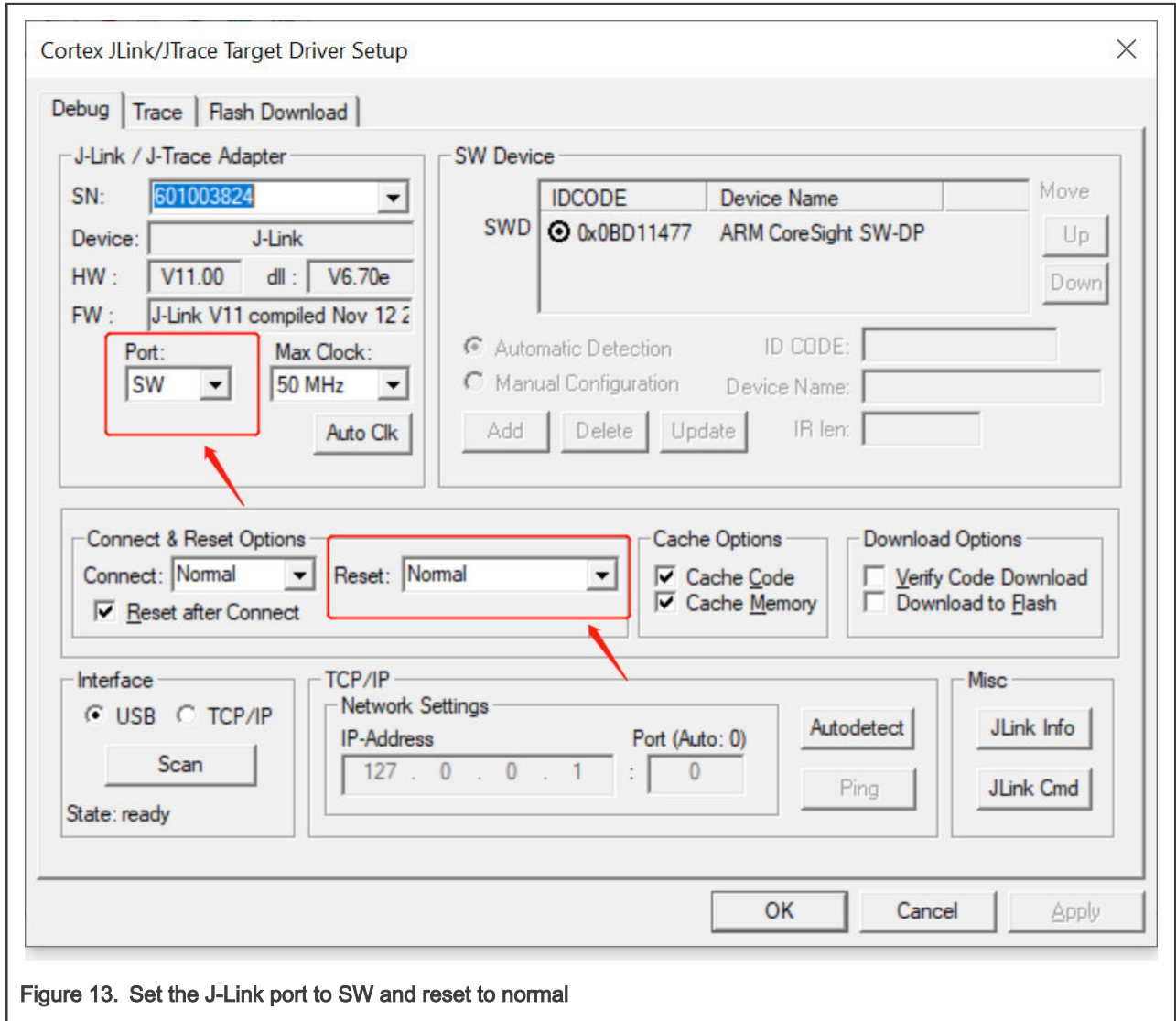


Figure 13. Set the J-Link port to SW and reset to normal

5. Click the trace submenu and make the configurations as shown in Figure 14.

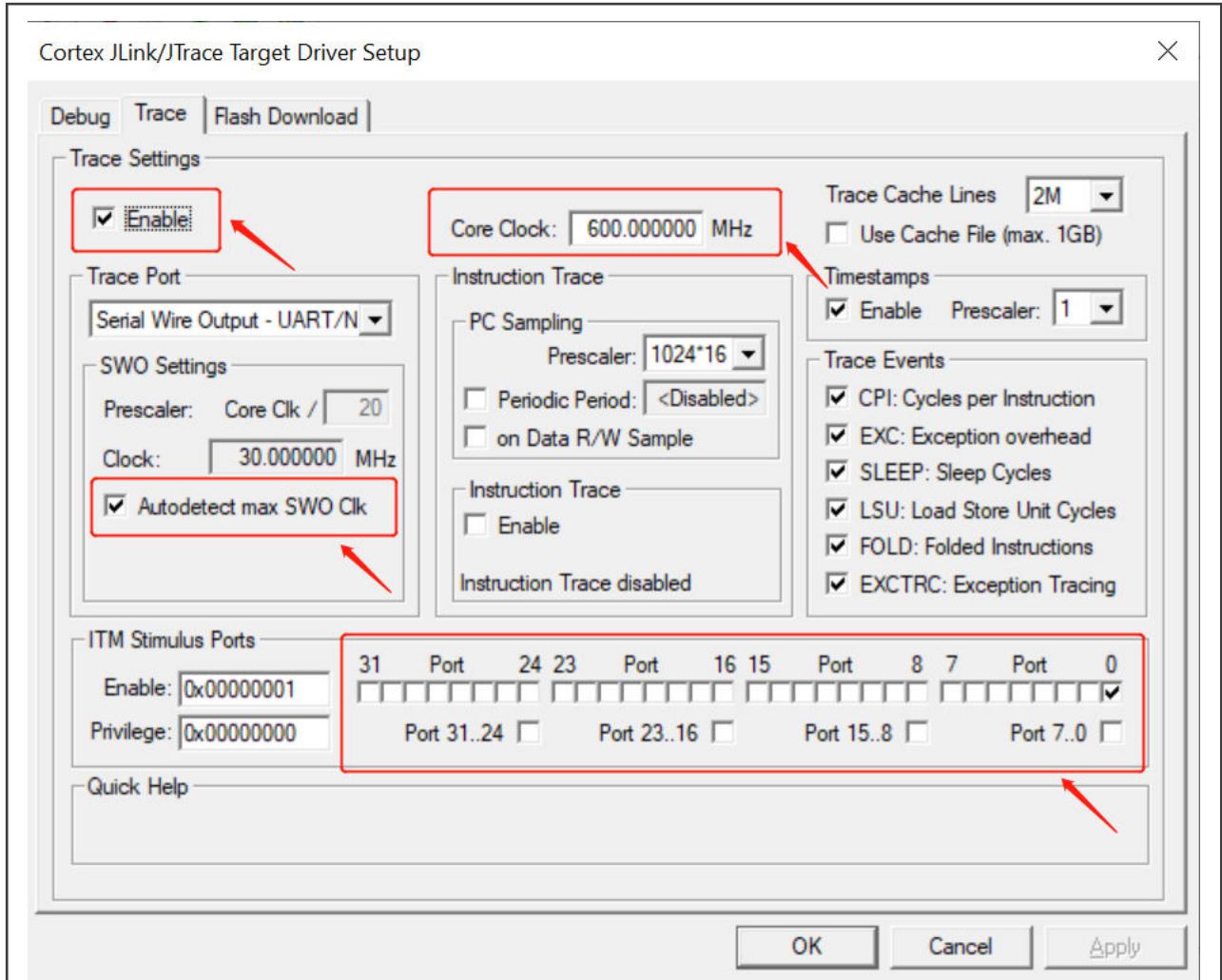


Figure 14. J-Link trace configurations

- Start debug session and click **View -> Serial Window -> Debug(printf) Viewer**. Run the project and the output message is as shown in Figure 15.



Figure 15. Debug(printf) Viewer in Keil

## 6 Conclusion

One of the basic requirements of embedded development is to output debugging information through the terminal, which can be achieved in either the following ways:

- Use the serial line to connect the UART on the board with the COM port on the PC



- Use the semi-host mechanism.

The CoreSight architecture in Cortex-M7 core breaks this situation. The Cortex-M7 core provides the ITM interface that supports output debug information via SWO pin. With this implementation, the debug information can be output without configuring serial ports and using terminal debugging software. Compared with the common serial port print, the advantages of SWO are as follows.

- The serial port is the on-chip peripheral of the MCU and occupies a peripheral resource, while the SWO does not occupy a peripheral.
- It requires to rewrite the driver for the serial port when using different MCUs, whereas the driver for SWO can be used with all MCUs with supported Cortex-M core.
- Serial port output is typically sent in an interrupt mode, and there may be some problems for observing some information in the interrupt while debugging, whereas SWO trace does not have interrupt problems.
- SWO is much faster than the serial port and has less impact on the code because the latency of SWO trace is low.

In addition to the debug information output feature that is most commonly used, there are other powerful features of SWO trace, such as, to display real-time variable changes, to measure code execution time, to monitor the interrupt activity, and so on.

## 7 References

- *i.MX RT1010 Processor Reference Manual* (document [IMXRT010RM](#))
- *i.MX RT1020 Processor Reference Manual* (document [IMXRT1020RM](#))
- *i.MX RT1050 Processor Reference Manual* (document [IMXRT1050RM](#))
- *i.MX RT1060 Processor Reference Manual* (document [IMXRT1060RM](#))
- *Using the MIMXRT1060/4-EVK with MCUXpresso IDE*
- *MCUXpresso IDE SWO Trace Guide* (document [MCUXPRESSO-SWO-TRACE](#))
- [CoreSight Components Technical Reference Manual](#)

## 8 Revision history

Rev.	Date	Substantive changes
1	06 July 2022	Updated code in <a href="#">MIMXRT1060-EVK</a>
0	April 2021	Initial release



## Legal information

### 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.

### 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 [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

## 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,  $\mu$ Vision, 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.

**Airfast** — is a trademark of NXP B.V.

**Bluetooth** — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

**Cadence** — the Cadence logo, and the other Cadence marks found at [www.cadence.com/go/trademarks](http://www.cadence.com/go/trademarks) are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide.

**CodeWarrior** — is a trademark of NXP B.V.

**ColdFire** — is a trademark of NXP B.V.

**ColdFire+** — is a trademark of NXP B.V.

**EdgeLock** — is a trademark of NXP B.V.

**EdgeScale** — is a trademark of NXP B.V.

**EdgeVerse** — is a trademark of NXP B.V.

**eIQ** — is a trademark of NXP B.V.

**FeliCa** — is a trademark of Sony Corporation.

**Freescale** — is a trademark of NXP B.V.

**HITAG** — is a trademark of NXP B.V.

**ICODE and I-CODE** — are trademarks of NXP B.V.

**Immersiv3D** — is a trademark of NXP B.V.

**I2C-bus** — logo is a trademark of NXP B.V.

**Kinetis** — is a trademark of NXP B.V.

**Layerscape** — is a trademark of NXP B.V.

**Mantis** — is a trademark of NXP B.V.

**MIFARE** — is a trademark of NXP B.V.

**NTAG** — is a trademark of NXP B.V.

**Processor Expert** — is a trademark of NXP B.V.

**QorIQ** — is a trademark of NXP B.V.

**SafeAssure** — is a trademark of NXP B.V.

**SafeAssure** — logo is a trademark of NXP B.V.

**Synopsys** — Portions Copyright © 2021 Synopsys, Inc. Used with permission. All rights reserved.

**Tower** — is a trademark of NXP B.V.

**UCODE** — is a trademark of NXP B.V.

**VortiQa** — is a trademark of NXP B.V.

arm

---

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

---

© NXP B.V. 2021-2022.

All rights reserved.

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

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 06 July 2022

Document identifier: AN13234