

# FM UNIVERSAL PERIPHERAL DRIVER LIBRARY QUICK START GUIDE

32-BIT MICROCONTROLLER  
FM Family

*User Manual*

---



## Revision History

Date	Version	Author	Issue
2014-12-11	V0.1	Edison Zhang	1 <sup>st</sup> preliminary version
2014-12-19	V0.2	Edison Zhang	Add RTC, VBAT, LPM driver. Add examples of RTC, WDG, DT, MFT. Support S62CDHx series.
2015-01-08	V0.3	Edison Zhang	Add I2S-Lite, VBAT, Reset, LVD drivers Add I2S-Lite, VBAT, LPM, LVD examples Support S6E100xF series
2015-02-13	V0.4	Edison Zhang	Add Program-CRC, DSTC, LCD, I2S, HyperBus, Flash, External Bus I/F drivers and examples. Add utility drivers for HyperFlash, Nand Flash, SDRAM, PSRAM Seg-LCD Support S6E1B8x/ S6E1B7x series
2015-03-30	V0.5	Edison Zhang	Add HSSPI, CAN, CANFD, SDIF, HDMI-CEC driver and examples.
2015-04-01	V1.0	Edison Zhang	First formal release
2015-08-17	V1.0.1	Edison Zhang	Support S6E2GMx series
2015-09-01	V1.1.0	Edison Zhang	Support S6E1C3x, S6E2HGx series
2015-09-29	V2.0.0	Edison Zhang	Modify location of pdl_device.h

This document contains 31 pages.



## Target products

This application note describes these products;

Series	Product Number (not including package suffix)
All FMx Series	-

## Table of Contents

### Contents

<b>1. Introduction.....</b>	<b>5</b>
1.1 About this Document .....	5
1.2 About PDL .....	5
1.3 Development Environment .....	5
1.4 Devices Supported .....	5
1.5 Peripheral Drivers and Examples Supported.....	6
<b>2. How to use the PDL in IAR Embedded Workbench .....</b>	<b>8</b>
<b>3. How to use the PDL in Keil MDK .....</b>	<b>18</b>
<b>4. How to Use Doxygen Document.....</b>	<b>27</b>
<b>5. Troubleshooting Tips .....</b>	<b>28</b>

# 1. Introduction

## 1.1 About this Document

This application notes records the release history of universal PDL and shows how to use the examples of universal PDL in both IAR Embedded Workbench and Keil MDK.

## 1.2 About PDL

The PDL is designed to ease the use of the peripherals of the FM MCU series. You do not need to know about the register and bit structure of the peripherals. You need only to set up a configuration and make use of the API functions, such as initialization of peripheral.

For most of the peripherals, several example codes are provided. You can study how to use the peripheral quickly by running the according examples.

## 1.3 Development Environment

- IAR Embedded Workbench V6.50
- Keil MDK 4.70

## 1.4 Devices Supported

Core	Deivce Type	Device Series	PDL v0.1	PDL v0.2	PDL v0.3	PDL v0.4	PDL v1.0	PDL v1.0.1	PDL v1.1.0	PDL V2.0.0
FM3	All	All	O	O	O	O	O	O	O	O
FM4	TYPE1	MB9BF56x	O	O	O	O	O	O	O	O
	TYPE2		O	O	O	O	O	O	O	O
	TYPE3	S6E2CCx	O	O	O	O	O	O	O	O
	TYPE4	S6E2DHx		O	O	O	O	O	O	O
	TYPE5	S6E2GMx						O	O	O
	TYPE6	S6E2HGx							O	O
FM0+	TYPE1	S6E1A1x	O	O	O	O	O	O	O	O
	TYPE2	S6E100x			O	O	O	O	O	O
		S6E1B8x				O	O	O	O	O
		S6E1B7x								
	TYPE3	S6E1C3x							O	O

O: Supported

## 1.5 Peripheral Drivers and Examples Supported

	PDL v0.1		PDL v0.2		PDL v0.3		PDL v0.4	
Peripheral	Driver	Example	Driver	Example	Driver	Example	Driver	Example
ADC	O	O	O	O	O	O	O	O
BT	O	O	O	O	O	O	O	O
CAN								
CAN-FD								
CLK	O		O		O		O	
CRC	O	O	O	O	O	O	O	O
CR	O		O		O		O	O
CSV	O	O	O	O	O	O	O	O
DAC	O	O	O	O	O	O	O	O
DMA	O	O	O	O	O	O	O	O
DT	O	O	O	O	O	O	O	O
DSTC							O	O
EXINT	O	O	O	O	O	O	O	O
EXTIF							O	O
HBIF							O	O
HSSPI								
I <sup>2</sup> S-Lite (M0+)					O	O	O	O
I <sup>2</sup> S (M4)							O	O
Flash							O	O
GPIO	O	O	O	O	O	O	O	O
LPM			O		O	O	O	O
LVD	O		O		O	O	O	O
MFS	O	O	O	O	O	O	O	O
MFT	O	O	O	O	O	O	O	O
PCRC							O	O
QPRC	O	O	O	O	O	O	O	O
RESET					O		O	
RTC			O	O	O	O	O	O
SD								
UID	O		O		O	O	O	O
USB								
VBAT			O		O	O	O	O
WC	O		O		O	O	O	O
WDG	O		O	O	O	O	O	O

O: Supported

Peripheral	PDL v1.0		PDL v1.0.1		PDL v1.1.0		PDL v2.0.0	
	Driver	Example	Driver	Example	Driver	Example	Driver	Example
ADC	O	O	O	O	O	O	O	O
BT	O	O	O	O	O	O	O	O
CAN	O	O	O	O	O	O	O	O
CAN-FD	O	O	O	O	O	O	O	O
CLK	O	O	O	O	O	O	O	O
CRC	O	O	O	O	O	O	O	O
CR	O	O	O	O	O	O	O	O
CSV	O	O	O	O	O	O	O	O
DAC	O	O	O	O	O	O	O	O
DMA	O	O	O	O	O	O	O	O
DT	O	O	O	O	O	O	O	O
DSTC	O	O	O	O	O	O	O	O
EXINT	O	O	O	O	O	O	O	O
EXTIF	O	O	O	O	O	O	O	O
HBIF	O	O	O	O	O	O	O	O
HSSPI	O	O	O	O	O	O	O	O
I <sup>2</sup> CSLAVE					O	O	O	O
I <sup>2</sup> S-Lite	O	O	O	O	O	O	O	O
I <sup>2</sup> S	O	O	O	O	O	O	O	O
Flash	O	O	O	O	O	O	O	O
GPIO	O	O	O	O	O	O	O	O
LPM	O	O	O	O	O	O	O	O
LVD	O	O	O	O	O	O	O	O
MFS	O	O	O	O	O	O	O	O
MFT	O	O	O	O	O	O	O	O
PCRC	O	O	O	O	O	O	O	O
QPRC	O	O	O	O	O	O	O	O
RESET	O		O		O		O	
RTC	O	O	O	O	O	O	O	O
SD	O	O	O	O	O	O	O	O
UID	O	O	O	O	O	O	O	O
USB								
VBAT	O	O	O	O	O	O	O	O
WC	O	O	O	O	O	O	O	O
WDG	O	O	O	O	O	O	O	O

O: Supported

## 2. How to use the PDL in IAR Embedded Workbench

- 1) Open IAR Embedded Workbench and select the MCU series and package of the MCU product you use. For example, you are using MB9AF314L. So make following setting in the `pdl_device.h` file.

- Set the MCU series

```
/**
*****
** \brief Global device series definition
**
** See pdl.h line 102 for choice list.
**
** \note This definition is used for GPIO settings, interrupt vector,
**       enumeration handling, and Type(0, 1, ...) Device handling.
*****/
#define PDL_MCU_SERIES PDL_DEVICE_SERIES_MB9AF31X
```

- Set the MCU package

```
/**
*****
** \brief Global package definition
**
** See pdl.h line 108 for choice list.
**
** \note This definition is used for device package settings
*****/
#define PDL_MCU_PACKAGE PDL_DEVICE_PACKAGE_MB_L
```

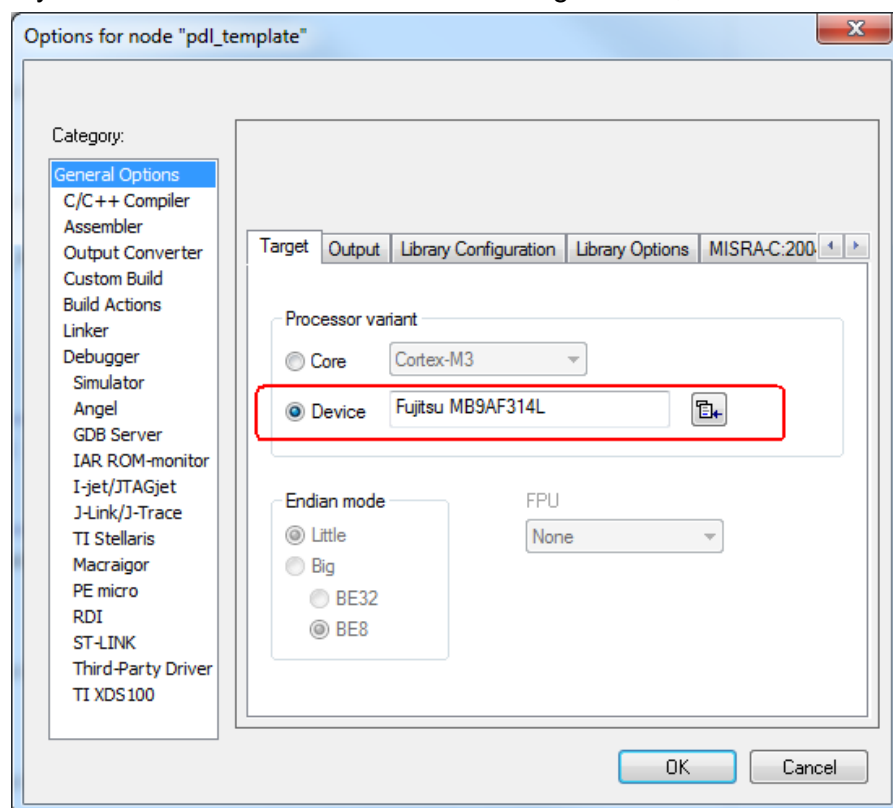
The original Fujitsu product starts with “MB” and should use the package definition “PDL\_DEVICE\_PACKAGE\_MB\_X”.

The current Spansion product starts with “S6” and should use the package definition “PDL\_DEVICE\_PACKAGE\_S6\_X”.

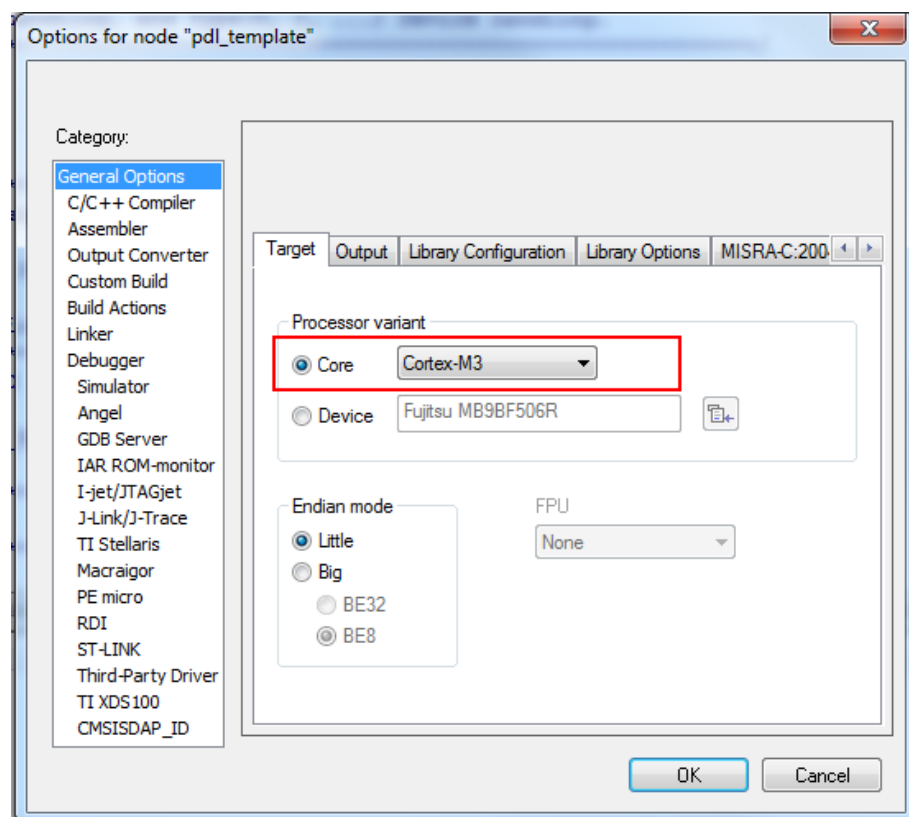


2) Select the MCU device or core of your MCU.

If you can find the exact device in the “Target” table, select the device:

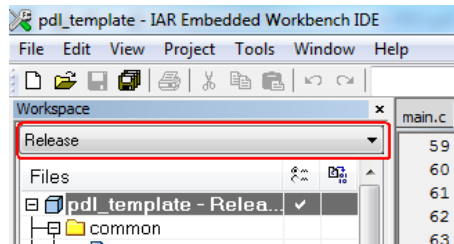


If you can't find the exact device in the “Target” table, select the core of your device:

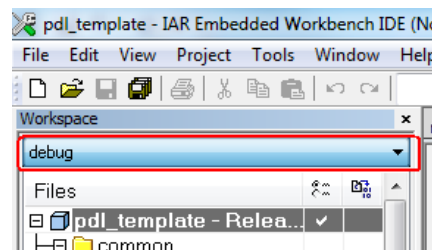


3) Both flash and RAM debug are supported.

- **Select “release” item as in the following picture and then see flash debug setting at step 4), 5), 6)**



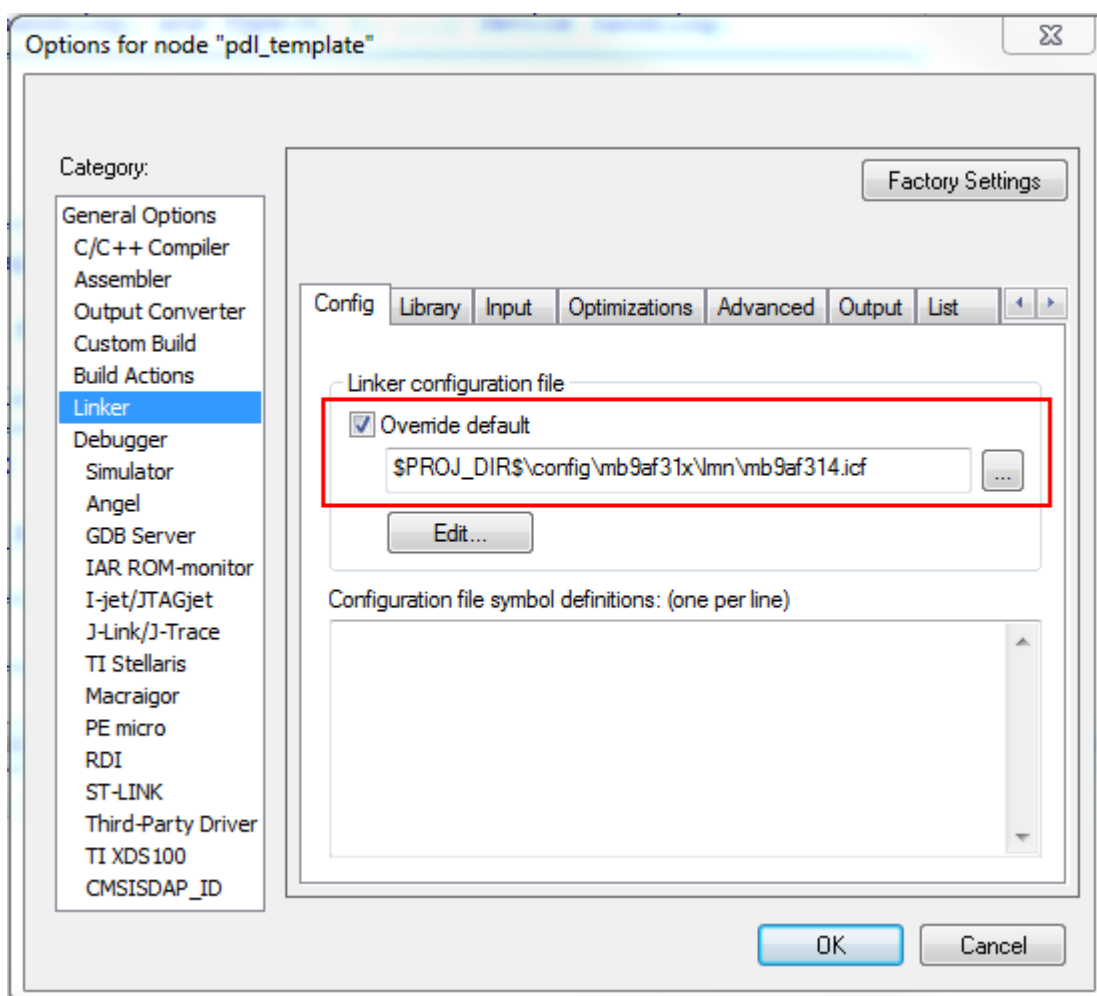
- **Select “debug” item as in the following picture and see RAM debug setting at step 7), 8), 9)**



## Flash Debug

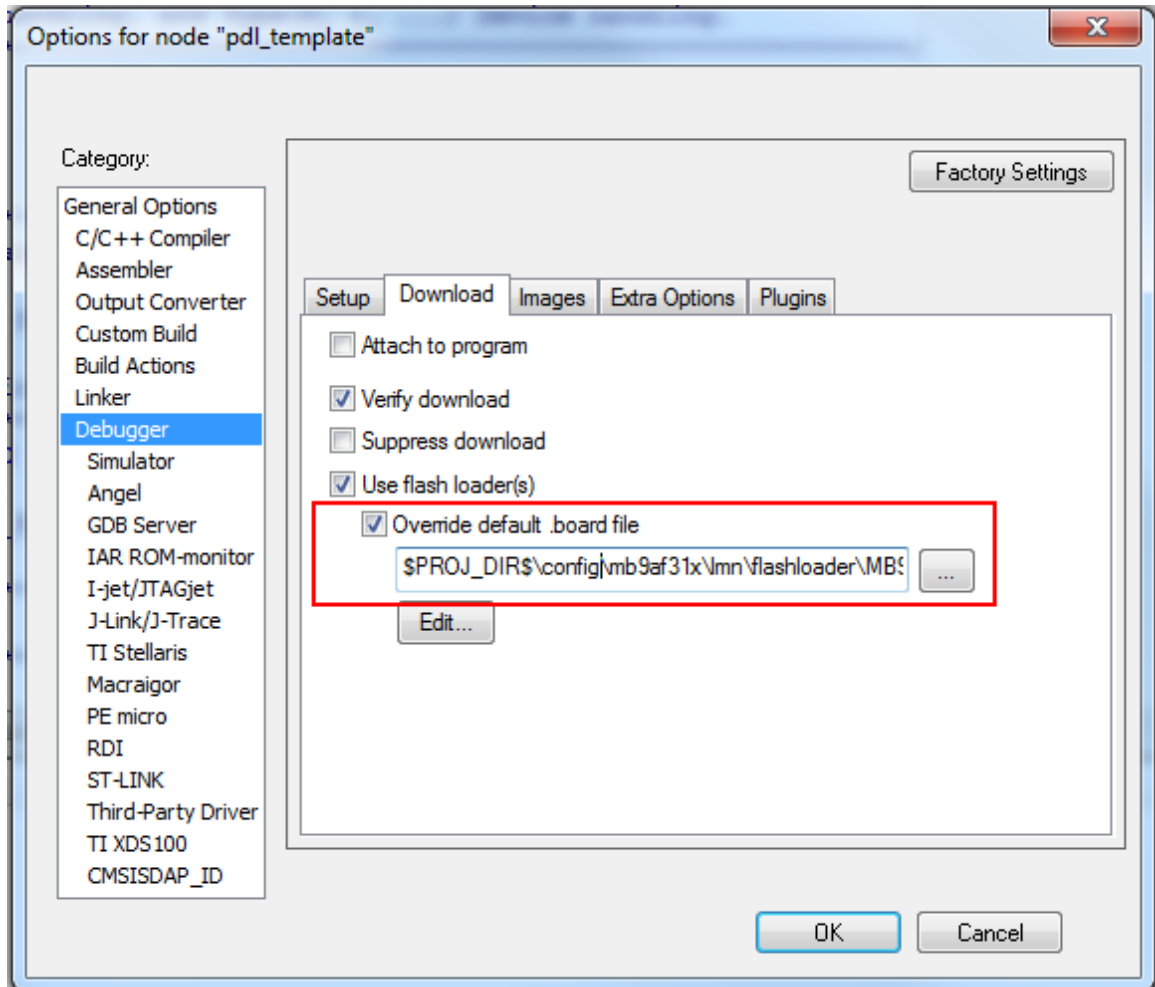
- 4) Select ICF file at the path: `..\template\IAR\config\MCU series\MCU package group\ MCU part number.icf`

Here, MCU series: mb9af31x, MCU part number: mb9af314, MCU package: L, which belongs to lmn



- 5) Select flash loader file at the path: `..\template\IAR\config\MCU series\MCU package group\flashloader\xxx.board`

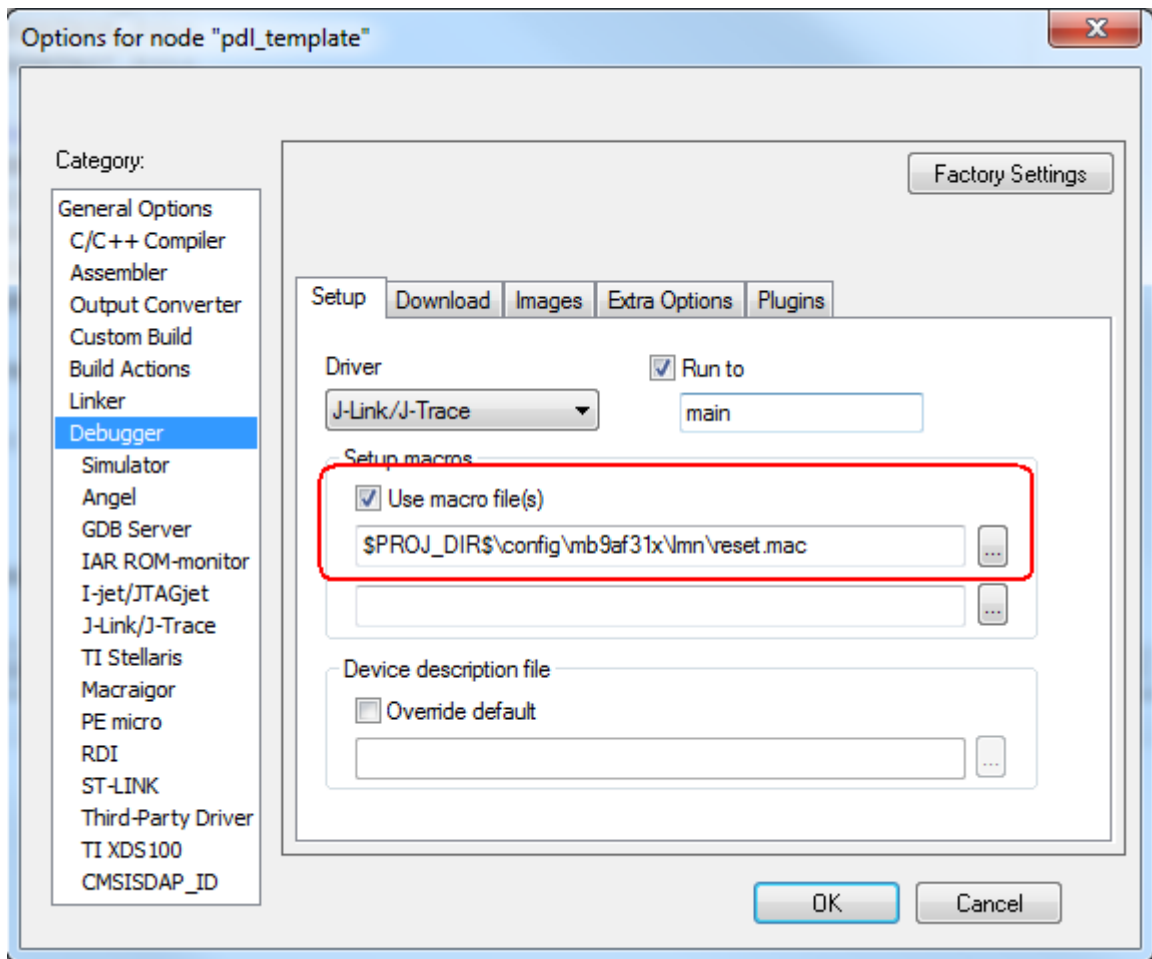
Here, `MCU series: mb9af31x`, `MCU package: L`, which belongs to `lmn`



- 6) Select macro file at the path: `..\template\IAR\config\MCU series\MCU package group\reset.mac`.

Here, `MCU series: mb9af31x`, `MCU package: L`, which belongs to `lmn`.

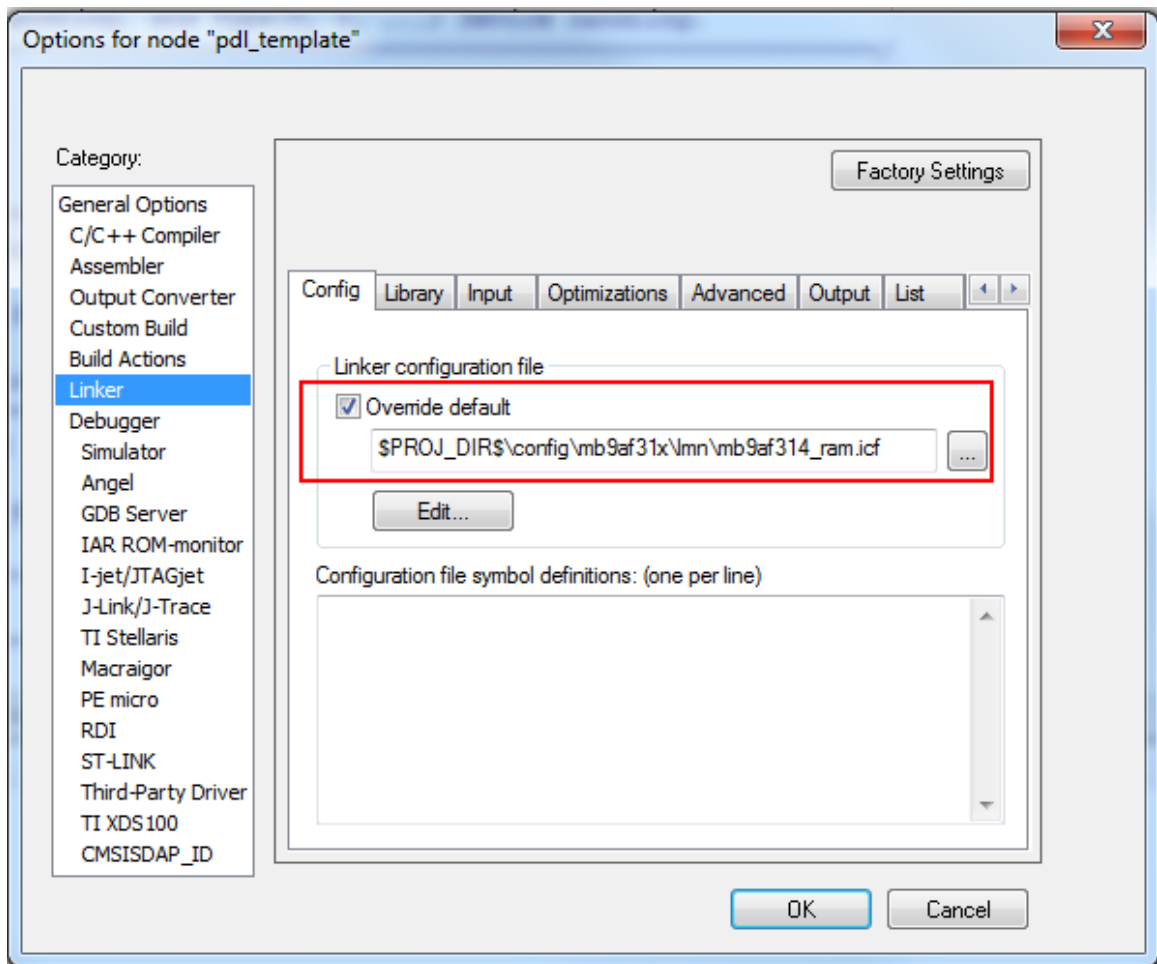
After this step, Jump to see step 10)



## RAM Debug

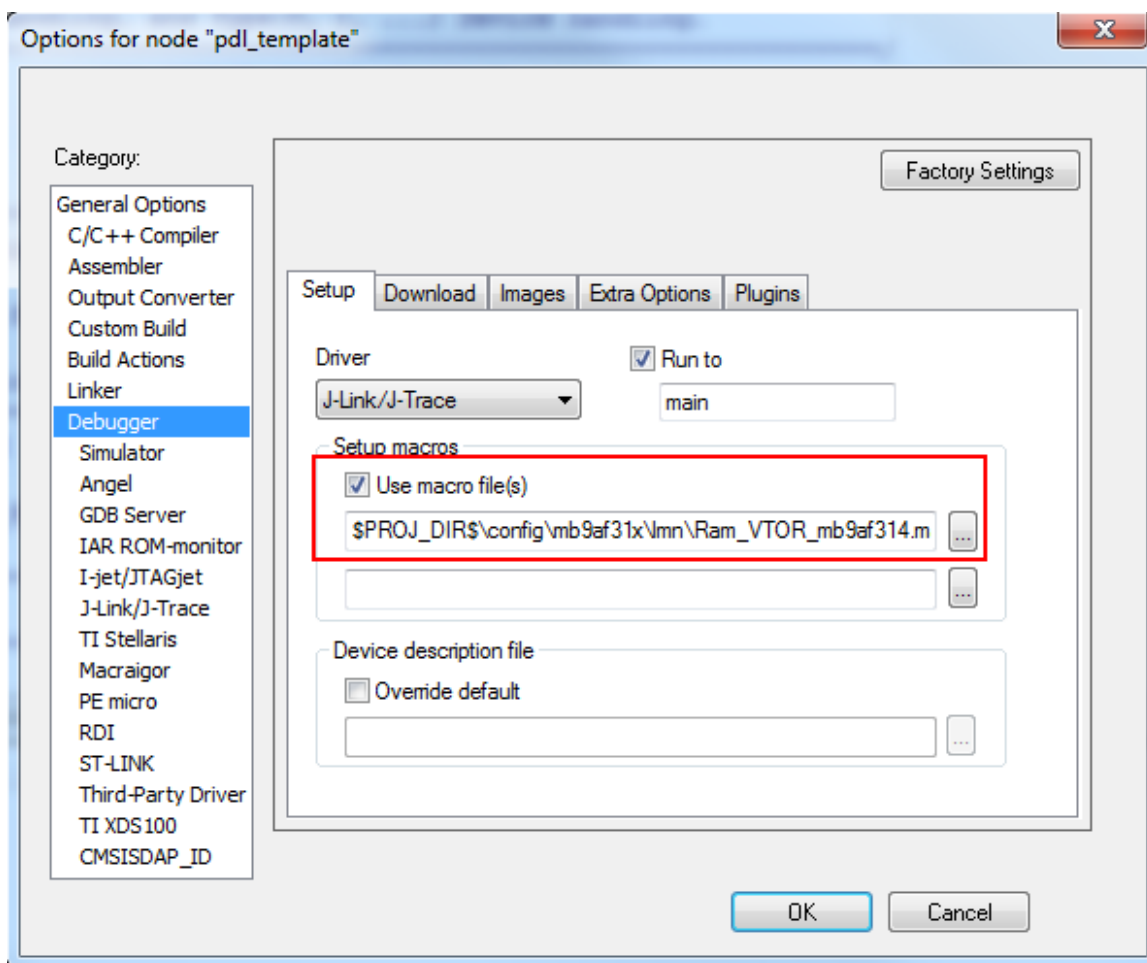
- 7) Select ICF file at the path: `..\template\IAR\config\MCU series\MCU package group\ MCU part number_ram.icf`

Here, **MCU series**: mb9af31x, **MCU part number**: mb9af314, **MCU package**: L, which belongs to lmn

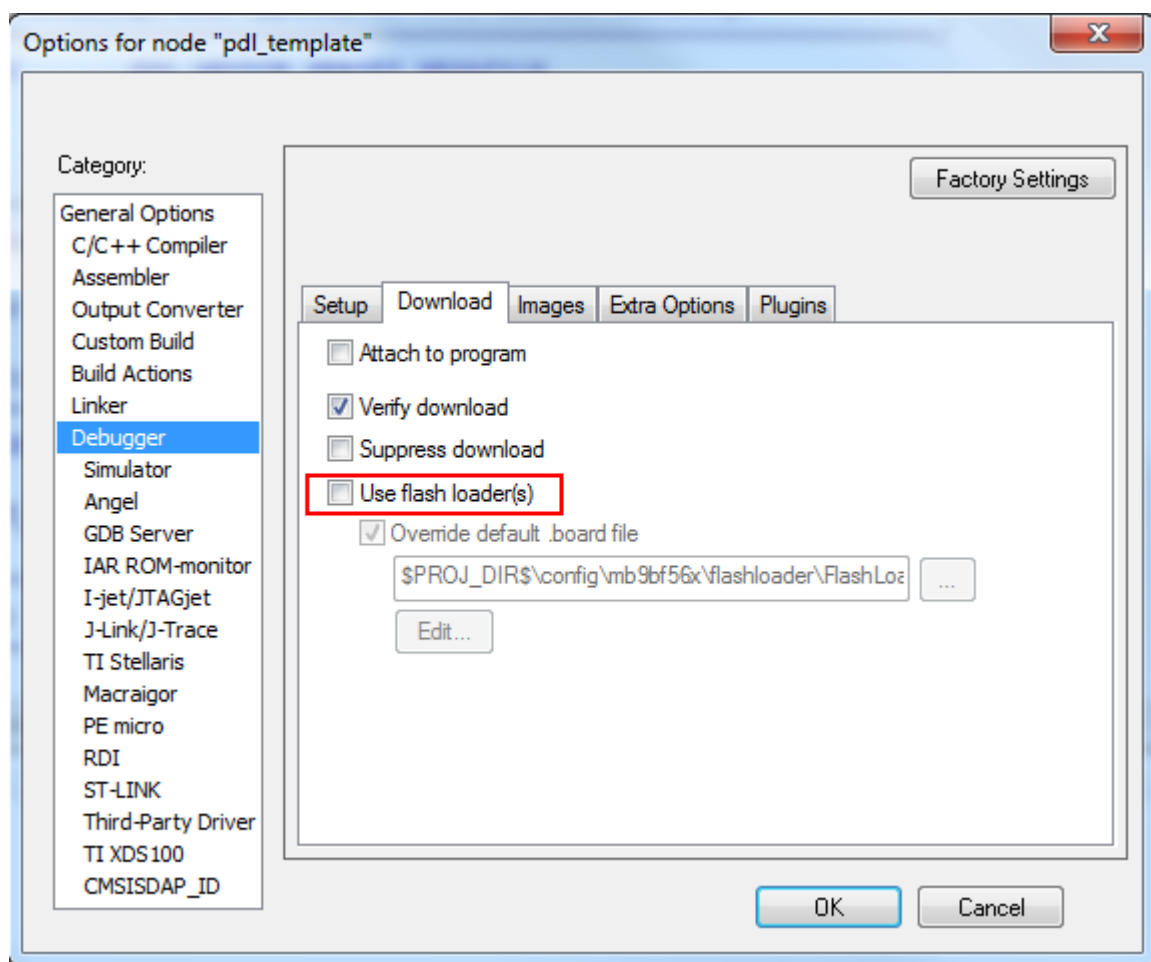


- 8) Select macro file at the path: `..\template\IAR\config\MCU series\MCU package group\ Ram_VTOR_mcu part number.mac`

Here, `MCU series: mb9af31x`, `MCU part number: mb9af314`, `MCU package: L`, which belongs to `lmn`

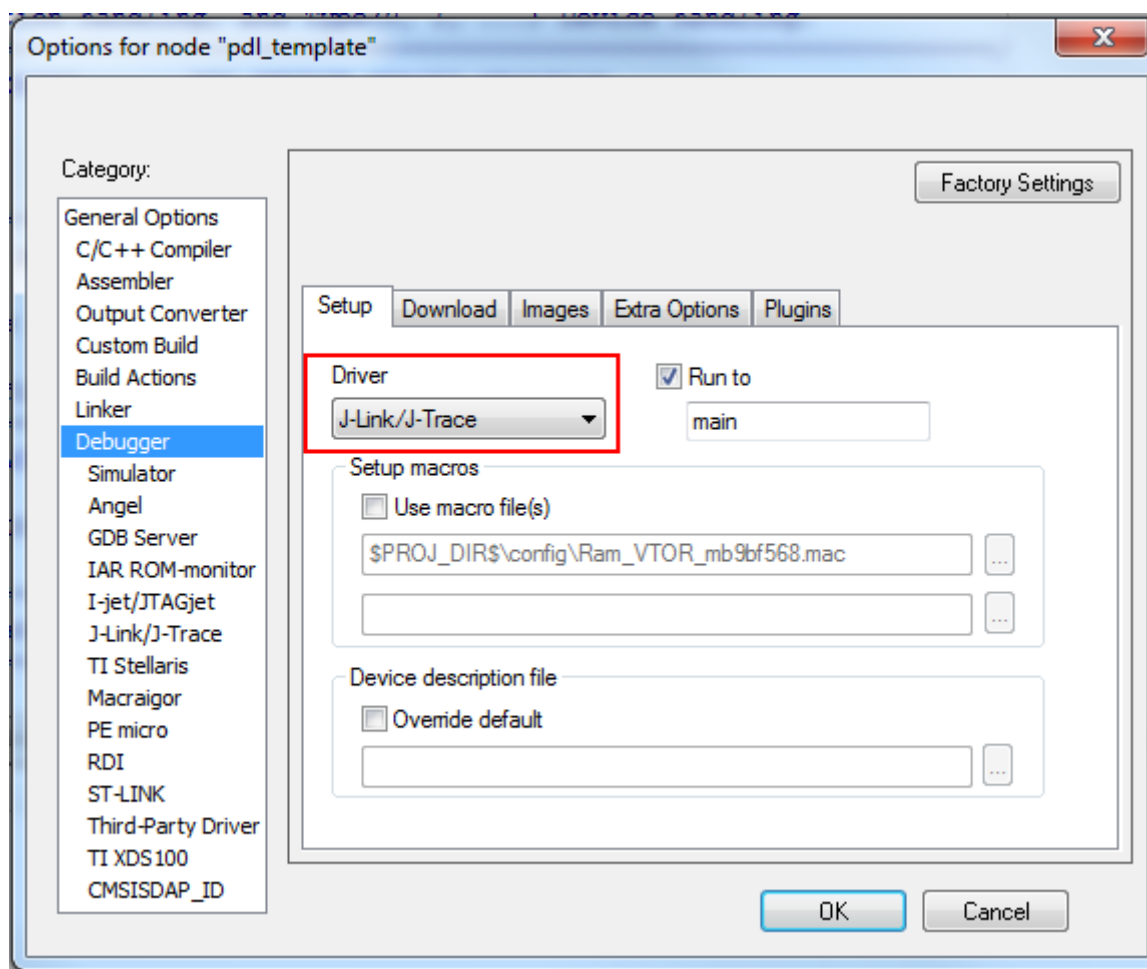


9) Make sure the flash loader file is unchecked.





10) Select a debugger.



FM3/FM4 can support both JTAG and SWD interfaces.

FM0+ can support only the SWD interface.

11) Copy the source files (`main.c` and `pdl_user.h`) under each example folder to the source folder of template folder.

12) Rebuild the project and start to debug.

### 3. How to use the PDL in Keil MDK

- 1) Select the MCU series and package of the MCU product you use. For example, you are using MB9AF314L. So make following setting in the `pdl_device.h` file.

- Set the MCU series

```
/**
*****
** \brief Global device series definition
**
** See pdl.h line 102 for choice list.
**
** \note This definition is used for GPIO settings, interrupt vector,
**       enumeration handling, and Type(0, 1, ...) Device handling.
*****/
#define PDL_MCU_SERIES PDL_DEVICE_SERIES_MB9AF31X
```

- Set the MCU package

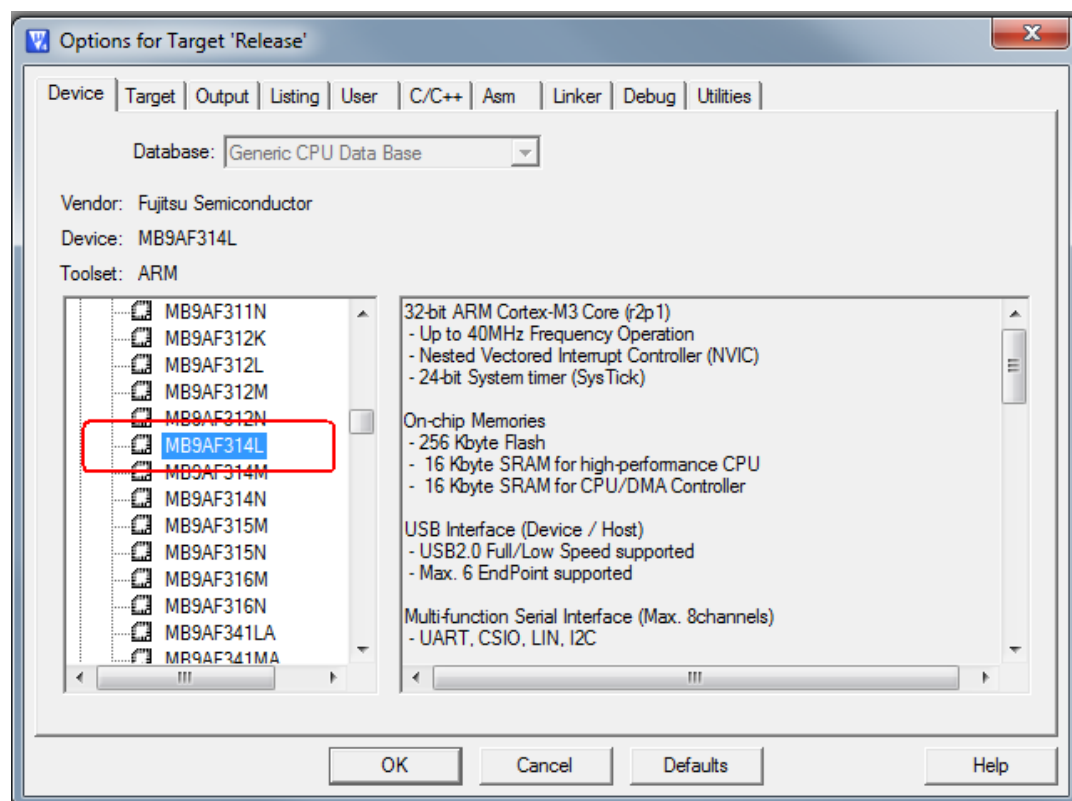
```
/**
*****
** \brief Global package definition
**
** See pdl.h line 108 for choice list.
**
** \note This definition is used for device package settings
*****/
#define PDL_MCU_PACKAGE PDL_DEVICE_PACKAGE_MB_L
```

The original Fujitsu product starts with “MB” and should use the package definition “PDL\_DEVICE\_PACKAGE\_MB\_X”.

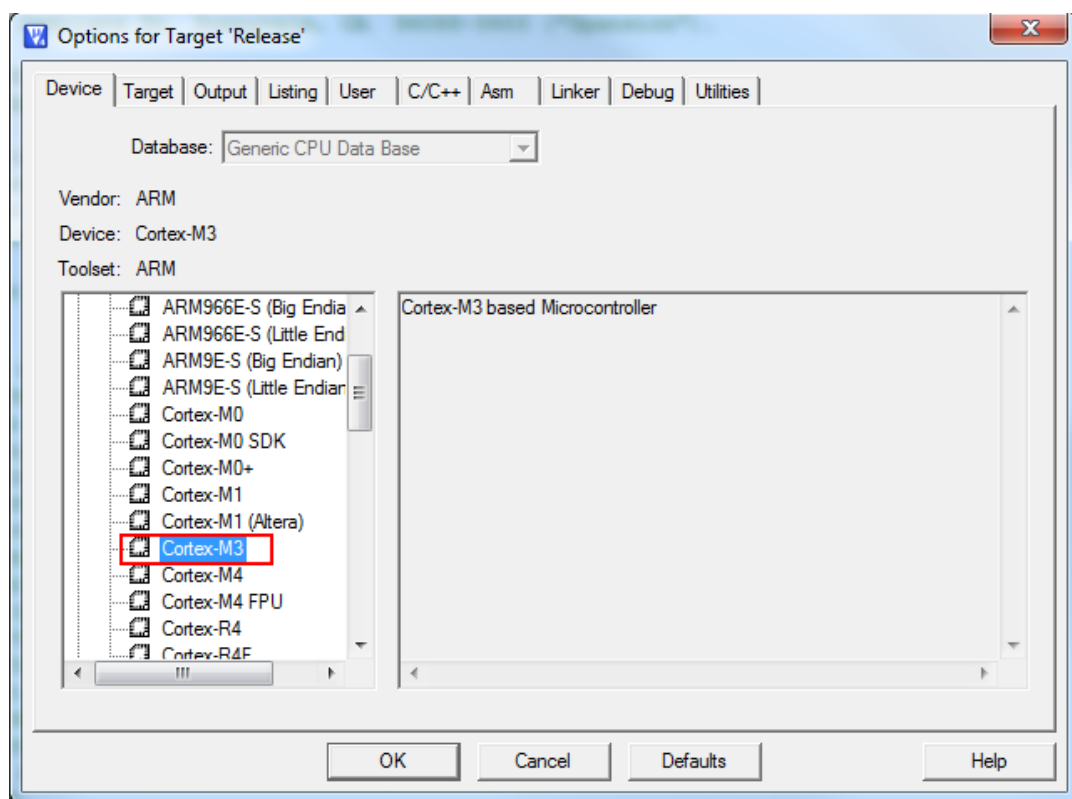
The current Spansion product starts with “S6” and should use the package definition “PDL\_DEVICE\_PACKAGE\_S6\_X”.

2) Select the exact device or core of the product you use.

If you can find the exact device in the “Device” table, select the device:

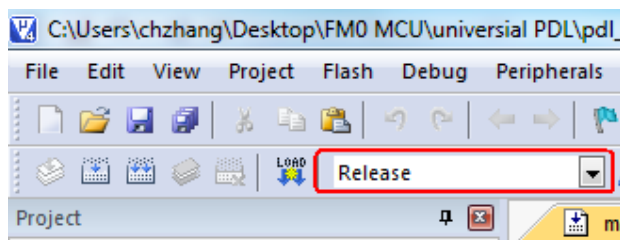


If you can't find the exact device in the “Device” table, select the core of your device:

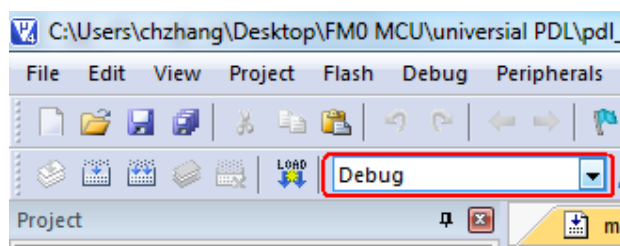


3) Both flash and RAM debug are supported.

- **Select “Release” item as in the following picture and see flash debug setting at step 4), 5), 6)**

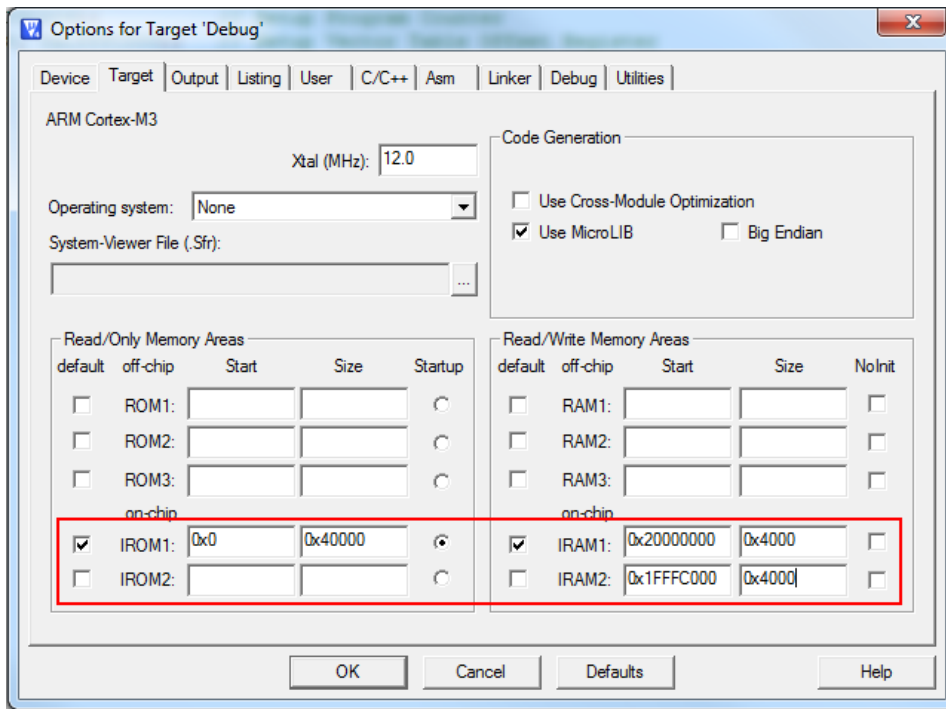


- **Select “Debug” item as in the following picture and see RAM debug setting at step 7), 8), 9)**

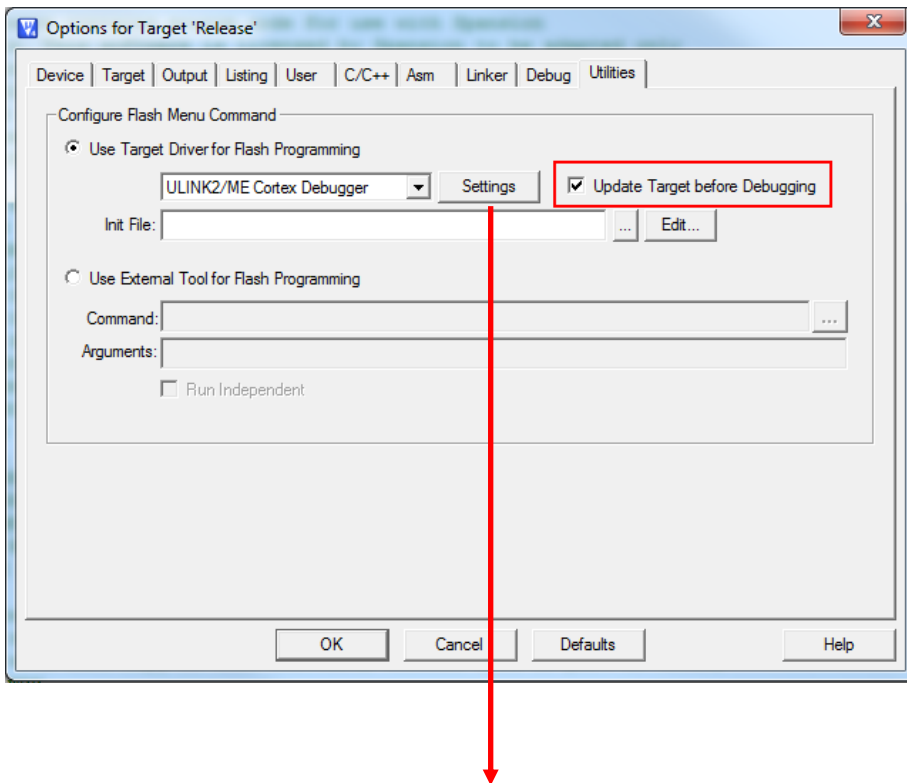


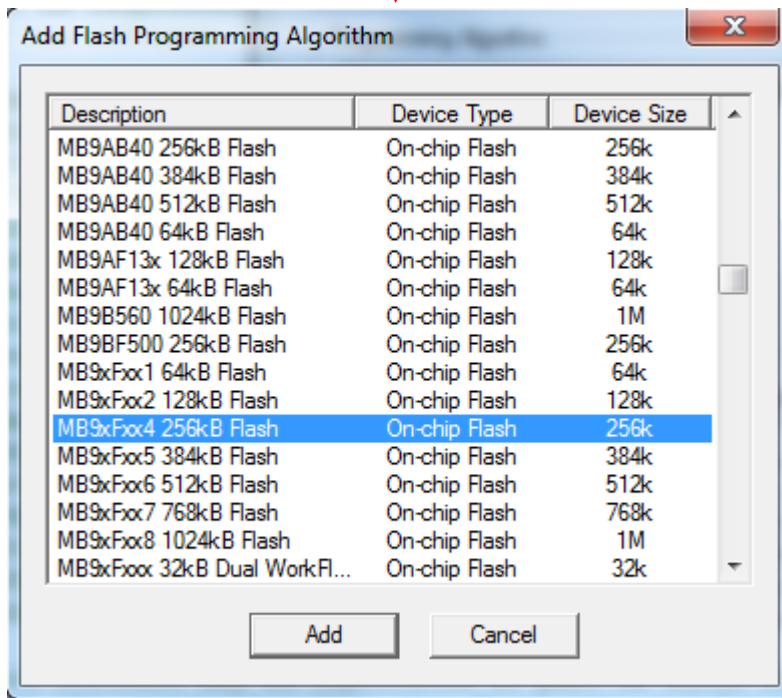
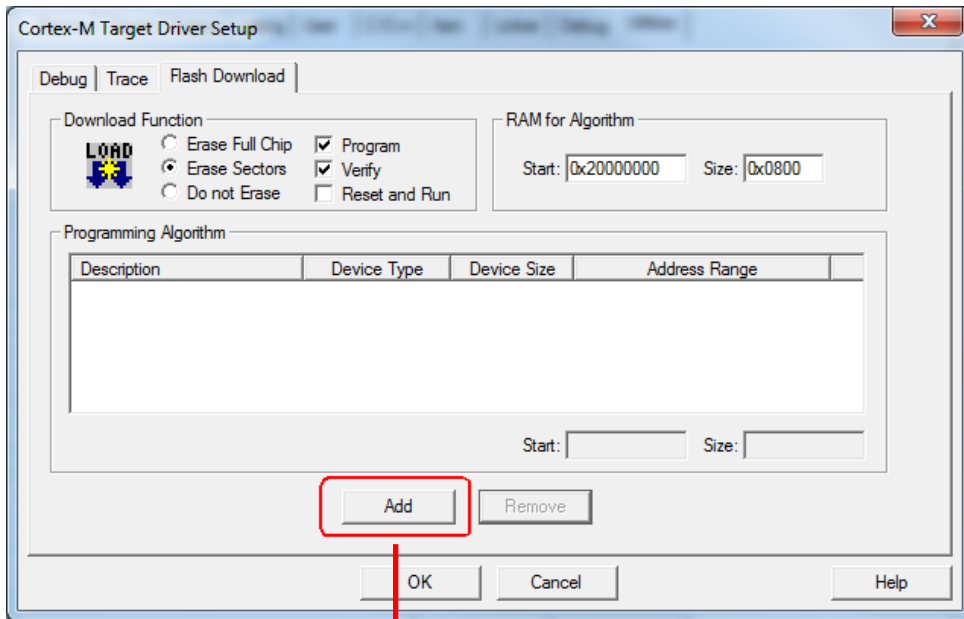
## Flash Debug

- 4) Set the ROM and RAM area of your product. (You can check this information in the product data sheet.)



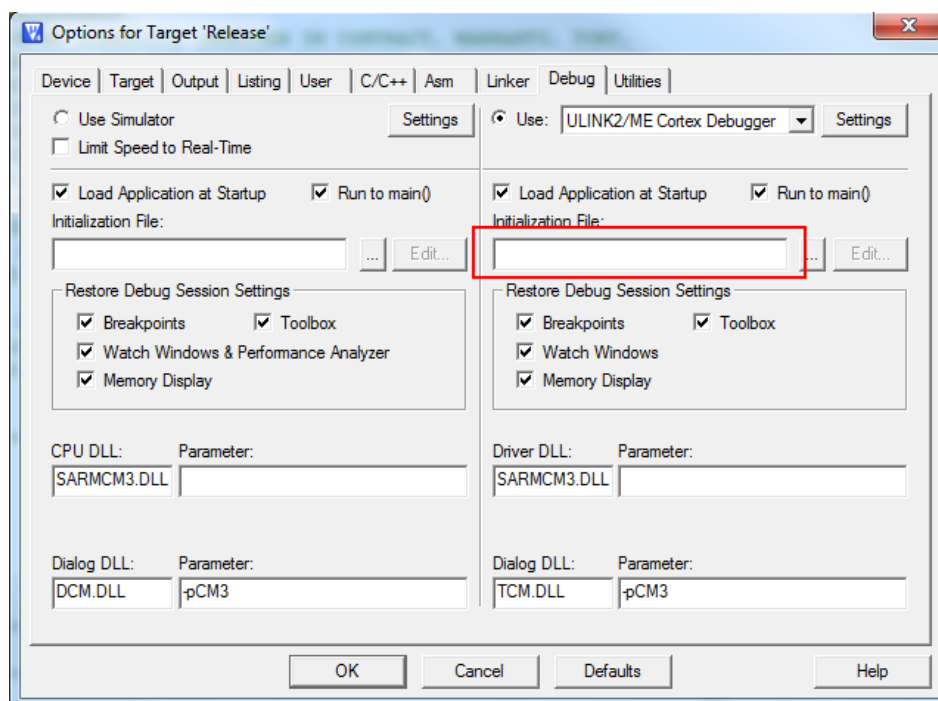
- 5) Check "Update Target before Debugging" and Select the flash loader file.





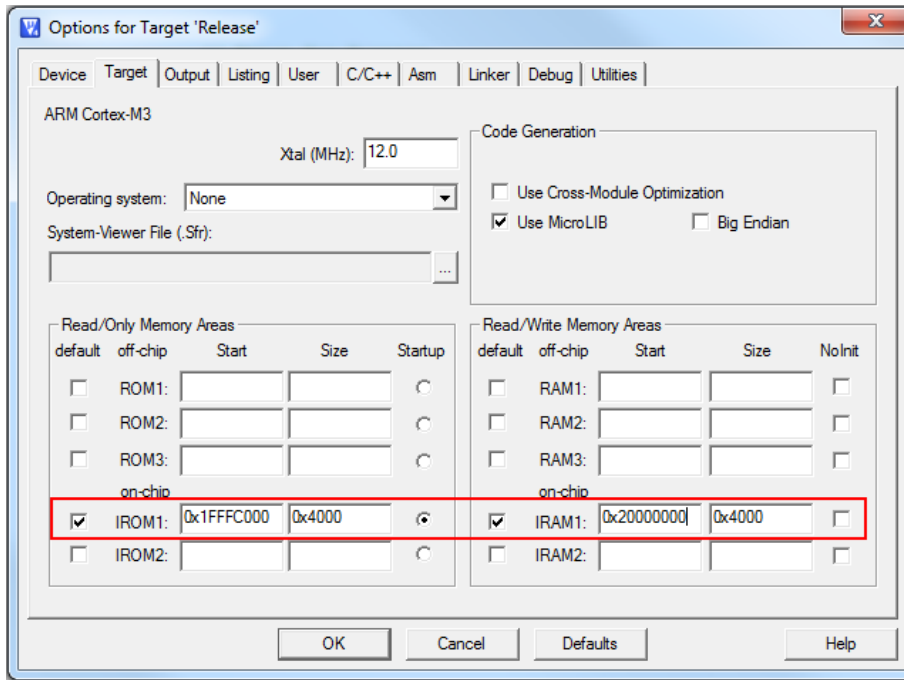
If you can't find the flash loader file for your device in the above pop up window, copy the flash loader files from the path: `\template\ARM\flashloader` to the path `..\Keil\ARM\Flash` and try again.

- 6) You do not need the init file for Flash debug. Delete the init file.  
After this step, jump to see step 10).

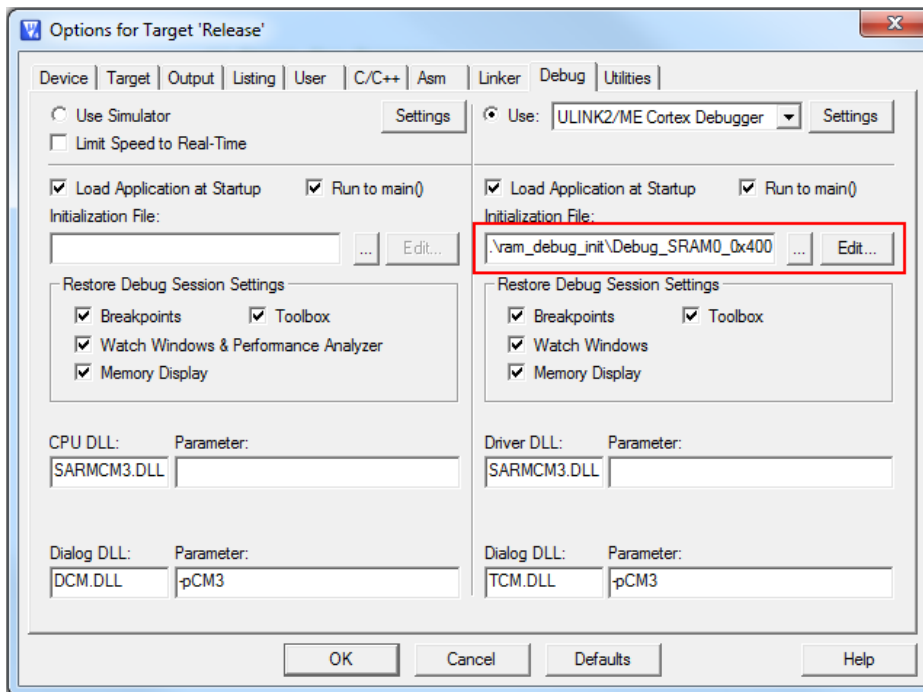


## RAM Debug

- 7) Set the ROM and RAM area of your product. (You can check this information in the product data sheet.) Note that ROM must be set to RAM space, as the code is debugged in RAM area and the IROM area should not overlap with IRAM area.

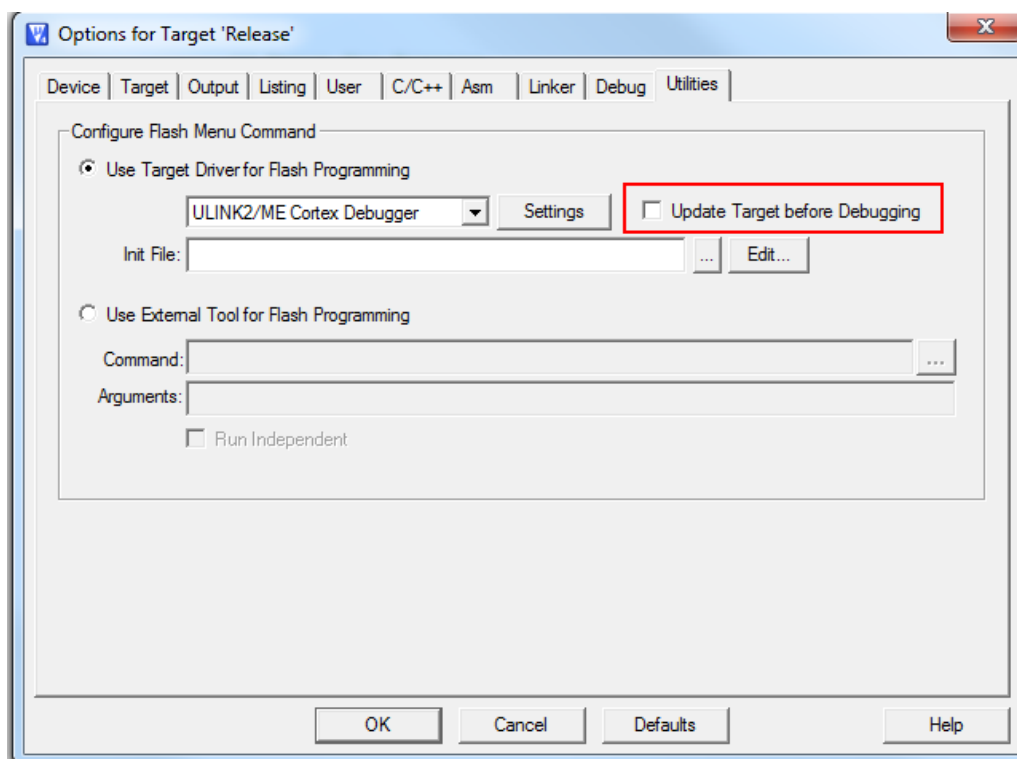


- 8) Select an init file at the path `..\template\ARM\ram_debug_init\Debug_SRAM0_offset.ini`. Here, `offset`: the offset of `0x2000 0000`. For example, `0x4000` means the RAM area of the device is `0x1FFF C000 – 0x2000 4000`. And for this device, the IROM and IRAM must be set to `0x1FFF C000 - 0x2000 0000` and `0x2000 0000 - 0x2000 4000`.

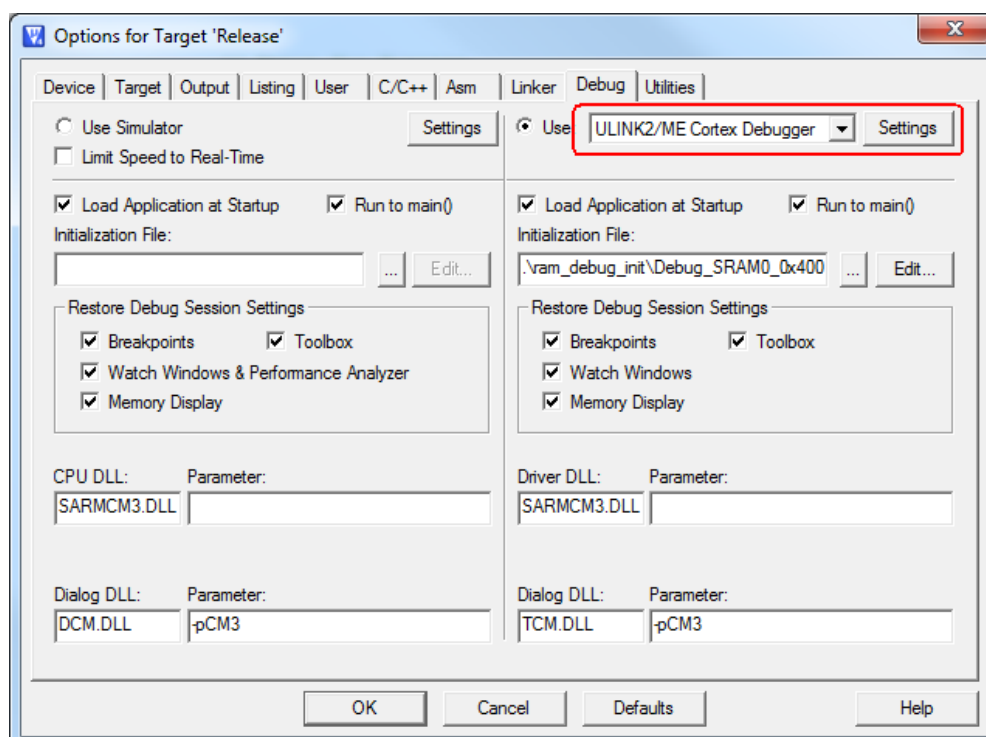




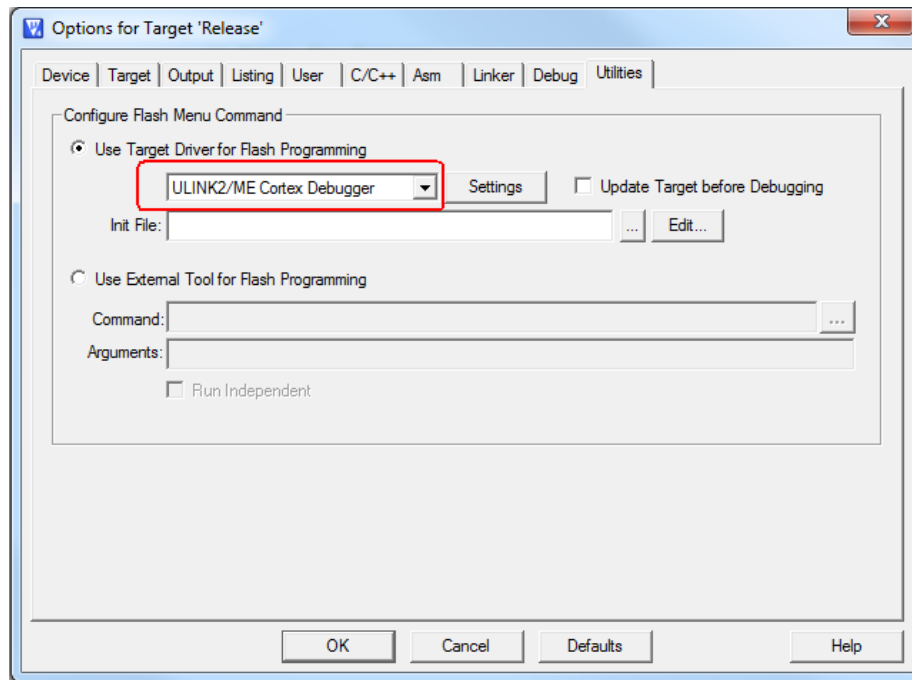
9) Uncheck the “Update Target before Debugging”



10) Select a debugger



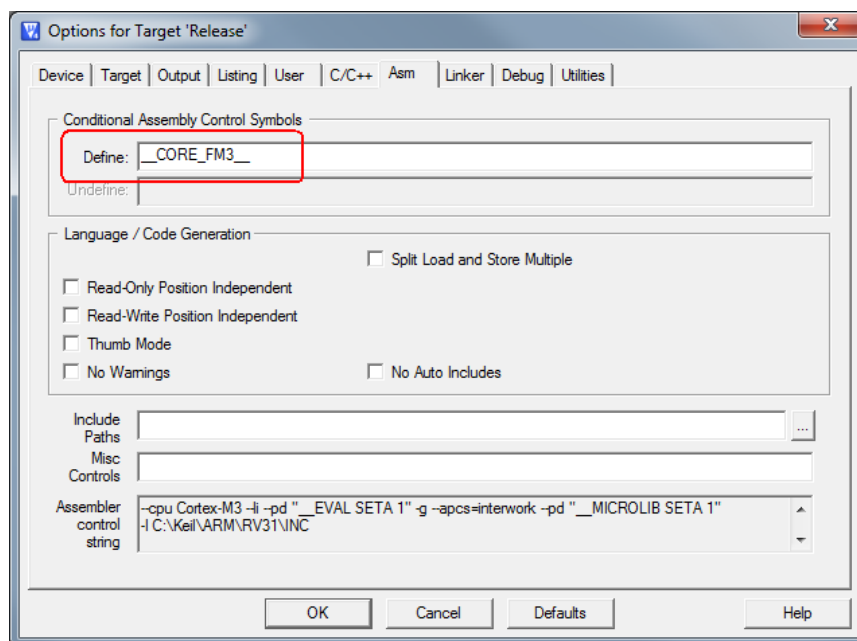
Also check the setting in the following window.



FM3/FM4 can support both JTAG and SWD interfaces.  
 FM0+ can support only the SWD interface.

11) Define MCU core in the ASM setting table.

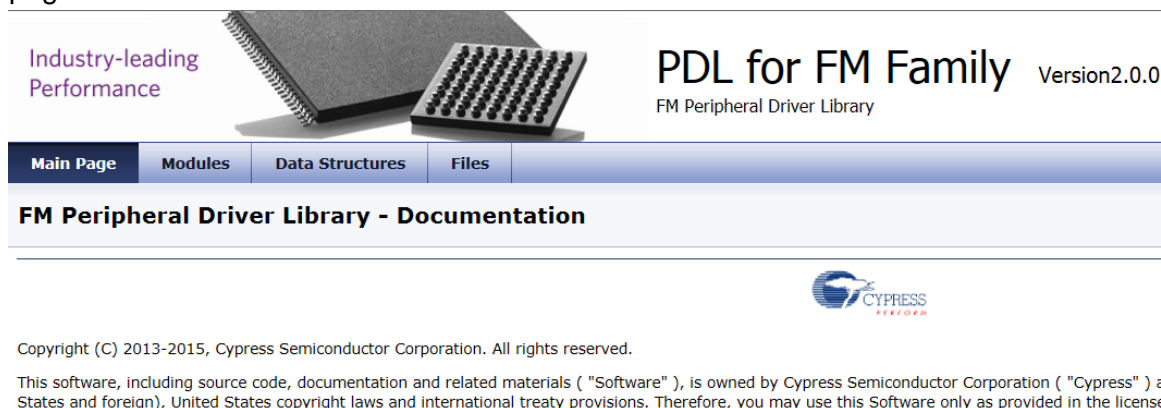
- FM3: Use “\_\_CORE\_FM3\_\_”
- FM0+: Use “\_\_CORE\_FM0P\_\_”
- FM4: Use “\_\_CORE\_FM4\_\_”



- 12) Copy the source files (`main.c` and `pdl_user.h`) under each example folder to the source folder of the template folder.
- 13) Rebuild the project and start to debug.

## 4. How to Use Doxygen Document

- 1) Double-click the file “\_description\_8h.html” at the path: `\doc\doxy\html`. The following web page can be seen:



- 2) There are four tables:
  - a. Main Page: record project history, driver modules, and examples of each module.
  - b. Modules: List all the driver modules. The API of each module can be scanned by clicking the name of each module.
  - c. Data structure: List all the global structure of the library. The elements of each structure can be scanned by clicking the name of each structure.
  - d. Files: List all the source files in the library. The source code of each file can be scanned by clicking the name of each file.

## 5. Troubleshooting Tips

### 1) How can I switch to another channel of peripheral when using the example code?

A definition or variable is always provided to record the channel information of the peripheral in each example project. Usually, the following steps must be done to switch to another channel:

- Modify the channel definition or variable
- Change I/O port (if GPIO or function pins are used)
- Set the original channels to "PDL\_OFF" and new channels to "PDL\_ON" in the `pdl_user.h` file.

### 2) How do I deal with an error like "SetPinFunc\_xxx is not defined"?

First, check whether the peripheral you used is available in this product and then check whether this pin is available in this product.

If this pin is not available in this product, try to use another location for the pin or other channels.

### 3) Can I make use of "printf" function to dump the information via UART?

Yes, this function is supported in the PDL.

Set "PDL\_PERIPHERAL\_ENABLE\_MFS0" and "PDL\_UTILITY\_ENABLE\_UART\_PRINTF" to PDL\_ON in `pdl.user.h`

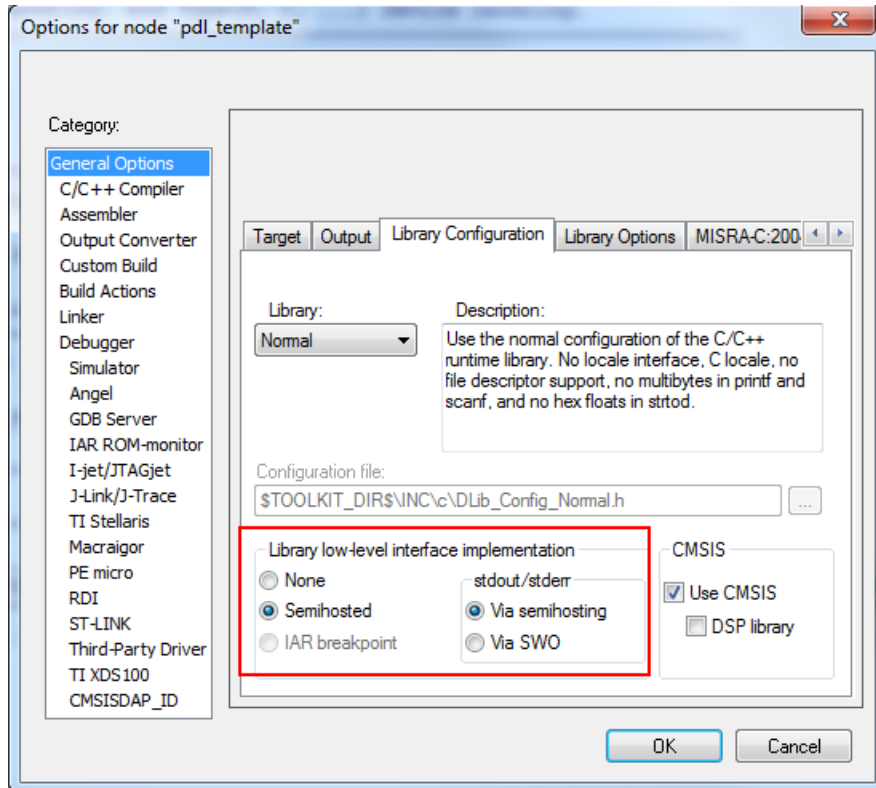
Call "Uart\_Io\_Init()" at the beginning of main function.

Open Hyper Terminal and set the configuration:

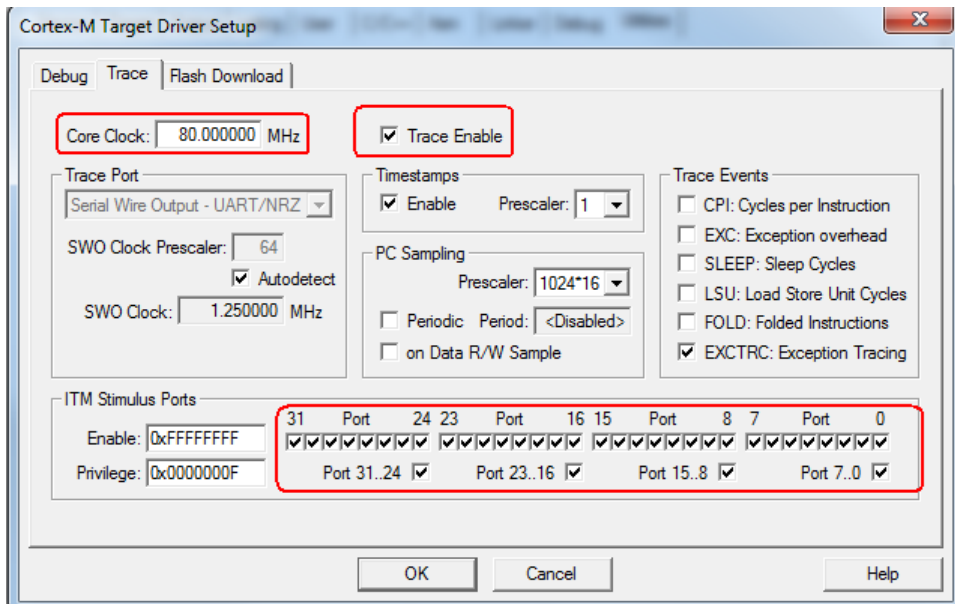
- Baudrate: 115200
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None

#### 4) Can I make use of “printf” fucniton to dump the information to terminal window?

In the IAR workbench, make following setting:



In the Keil MDK, make the following setting that enables trace, sets MCU core clock, and enables the ITM ports.



### 5) Can I use “printf” to dump information to terminal window with the CMSIS-DAP debugger?

The J-Link and U-Link can support the function of “printf” to terminal window well, but CMSIS-DAP may have some problems when using this function in some situations. We do not recommend using this function with CMSIS-DAP debugger.

Comment out the definition “DEBUG\_PRINT” in `pdl_user.h` to disable this function.

---

**FMx\_UM\_PDL**

---

**Cypress • Controller Manual**

FM Family  
32-BIT MICROCONTROLLER  
All FM Series  
PDL USER MANUAL

---

April, 2015 Rev. 1.0

Published: Cypress Inc.  
Edited: MCU Industrial Application Team

---

### ***Colophon***

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

### ***Trademarks and Notice***

The contents of this document are subject to change without notice. This document may contain information on a Spansion product under development by Spansion. Spansion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spansion assumes no liability for any damages of any kind arising out of the use of the information in this document.

Copyright © 2013 Spansion Inc. All rights reserved. Spansion®, the Spansion logo, MirrorBit®, MirrorBit® Eclipse™, ORNAND™ and combinations thereof, are trademarks and registered trademarks of Spansion LLC in the United States and other countries. Other names used are for informational purposes only and may be trademarks of their respective owners.