



# **Visual Studio Code User Setup**

## **Readme**

### **V1.3**

## **About this Document**

This document only supports “**VSCodeUserSetup-x64-1.62.3**”. For any other version than v1.62.3, there is no guaranteed it can work properly.

## **Table of Contents**

About this Document .....	1
1. Introduction .....	2
2. Visual Studio Code Development Environment .....	3
2.1 Debugger .....	3
2.2 Development Board .....	3
2.3 Visual Studio Code .....	4
3. Visual Studio Code Setup .....	5
3.1 GNU ARM Toolchain Configuration .....	5
3.2 JLink Extension Configuration .....	5
3.3 JLink Edit JLinkDevices.xml .....	6
3.4 VS Code Extension Configuration .....	8
3.5 VS Code Configuration .....	8
3.6 VS Code Disable Update .....	10
3.7 Project Builder Configurations .....	12
Revision History .....	13

## 1. Introduction

The purpose of this document is to teach users about the installation and configuration of Visual Studio Code, Embedded IDE, and [GNU ARM Embedded Toolchain](#). Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging, so users can develop applications on the Visual Studio Code development kit.

## 2. Visual Studio Code Development Environment



Rafael RT58x EVK



ARM J-Link Adapter



Visual Studio Code/EIDE

### 2.1 Debugger

- Install the ARM J-Link driver.
- Connect J-Link Adapter to RT58x EVK board via JTAG/SWD bus.
- Connect J-Link Adapter to PC via USB.

### 2.2 Development Board

The RT58x EVK provides an SWD interface with connector for use with an ICE debugger (J-Link Adapter) via 20pin IDE cable.



For detailed description of RT58x EVK board, please refer to RT58X EVK User Guide.

## 2.3 Visual Studio Code

Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute command in the console. Visual Studio Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster.

VS Code is available and free for download. It does not require a serial number or license key. For detailed introduction and download, please click the following link:

<https://code.visualstudio.com>

Users can also find the VS Code setup kit in the Rafael Micro download center.

<https://support.rafaelmicro.com:8088/projects/download-center/files>

The VS Code setup kit contains the following:

VS Code development kit:

[VS Code setup kit]\VSCode\VSCodeUserSetup-x64-1.62.3.exe

VS Code extension kit:

[VS Code setup kit]\VSCode\extensions.rar

GNU ARM Toolchain:

[VS Code setup kit]\ToolChain\GNU Arm Embedded Toolchain.rar

JLink extension kit:

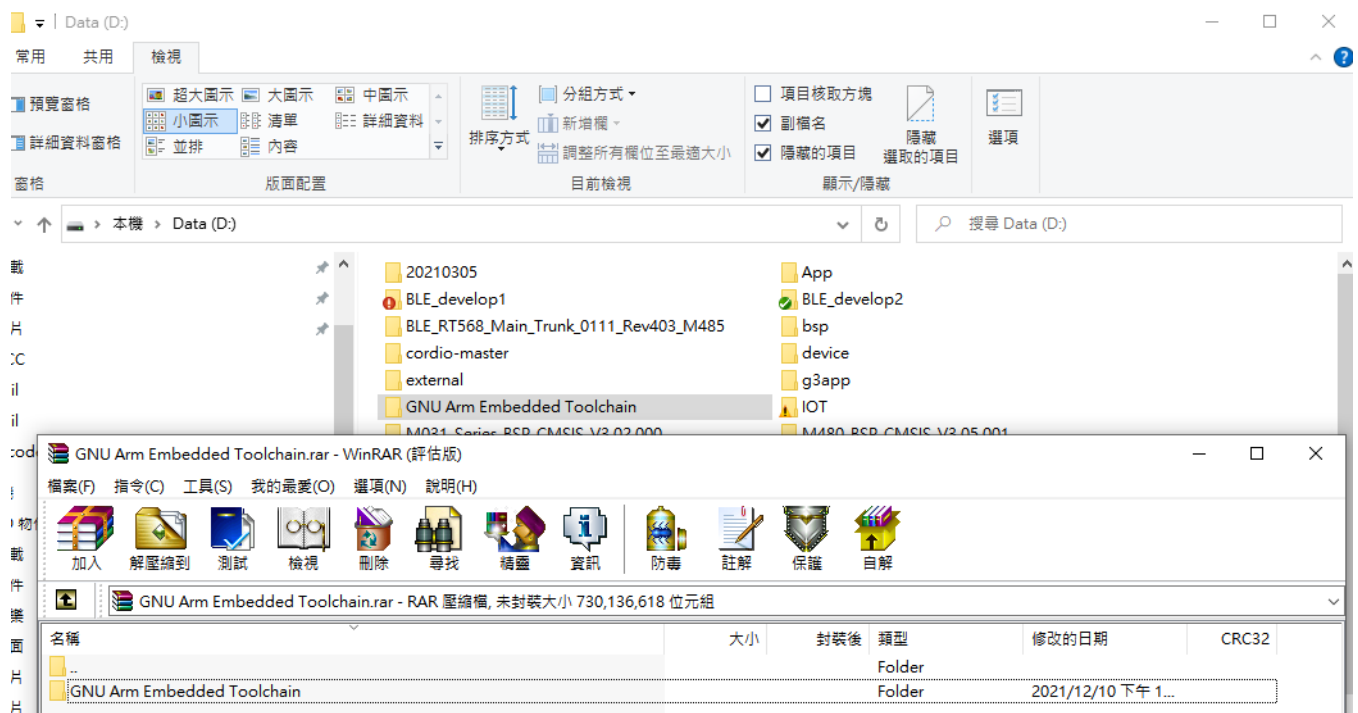
[VS Code setup kit]\JLink.rar

### 3. Visual Studio Code Setup

Getting up and running with Visual Studio Code is quick and easy. The Visual Studio Code installer is a small download so users can run the installer (VSCodeUserSetup-{version}.exe) to easily install in a few minutes by following the installation prompts.

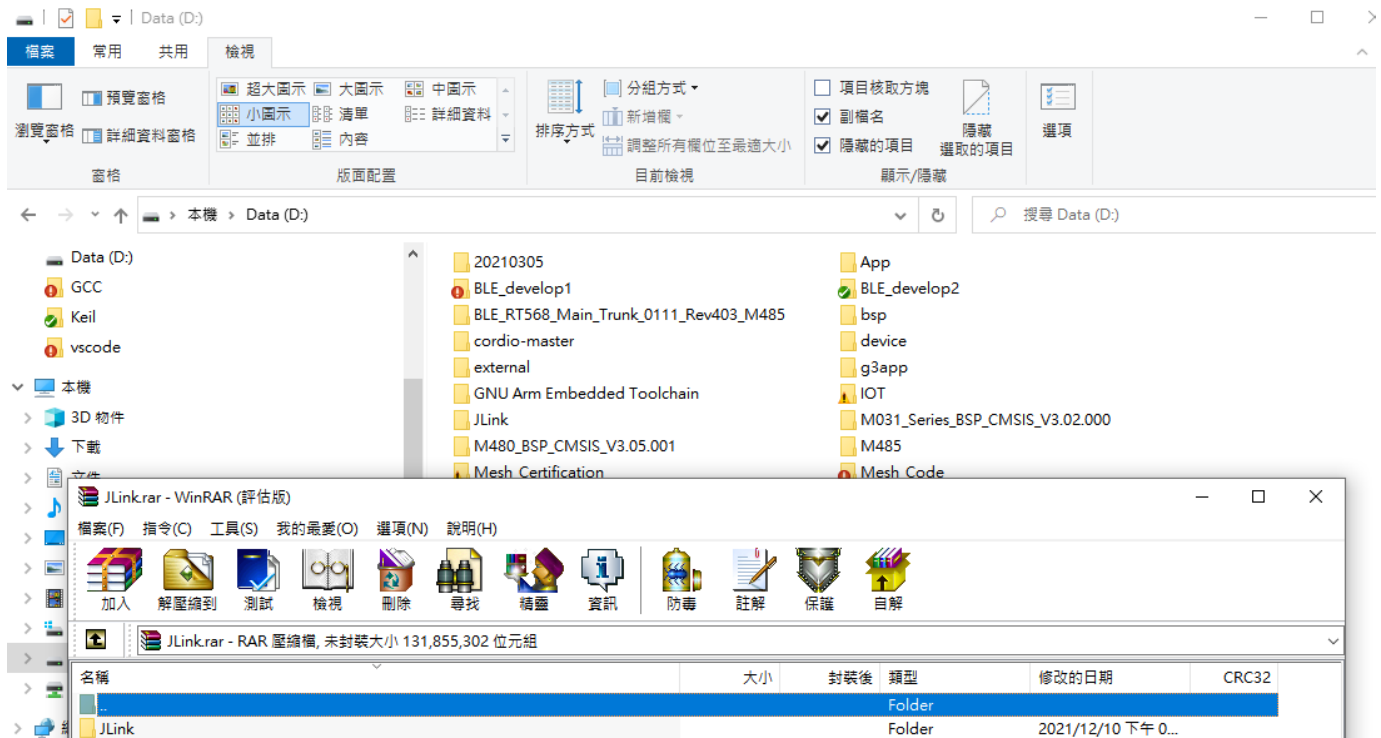
#### 3.1 GNU ARM Toolchain Configuration

Open the compressed file GNU Arm Embedded Toolchain.rar and extract the GNU Arm Embedded Toolchain folder to D disk.



#### 3.2 JLink Extension Configuration

Open the compressed file JLink.rar and extract the JLink folder to D disk.



### 3.3 JLink Edit JLinkDevices.xml

Edit Jlinkdevices.xml that the JLink for support different flash size MCU.

1. Copy Flash algorithm (RT58X\_1MB.FLM/RT58x\_512KB.FLM/RT58x\_2MB.FLM) into "D:\JLink"
2. Edit D:\JLink\JlinkDevices.xml file add new device

```

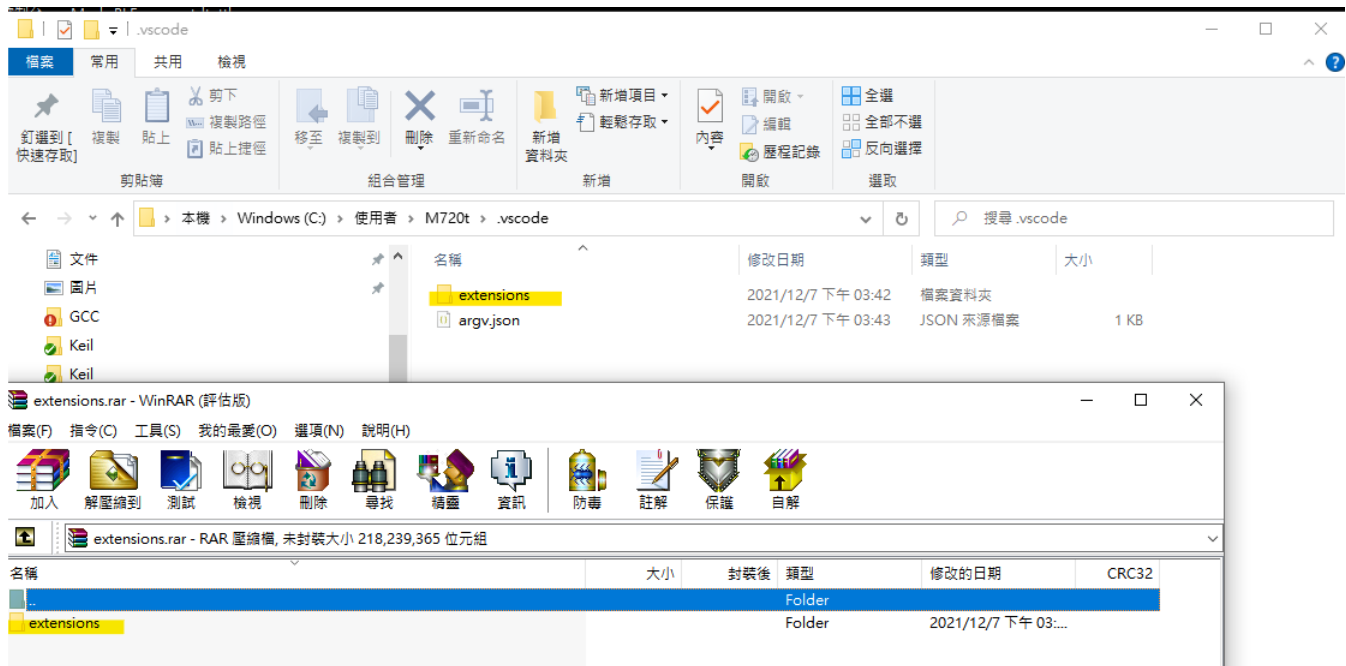
<DataBase>
  <!-- -->
  <!-- Rafael -->
  <!-- -->
  <Device>
    <ChipInfo Vendor="Rafael"
      Name="ARMCM3"
      Core="JLINK_CORE_CORTEX_M3"
      WorkRAMAddr="0x20000000"
      WorkRAMSize="0x1000" />
    <FlashBankInfo Name="Internal Flash"
      BaseAddr="0x00007000"
      MaxSize="0x000F7000"
      Loader="RT58x_1MB.FLM"
      LoaderType="FLASH_ALGO_TYPE_CMSIS"
      AlwaysPresent="1" />
  </Device>
  <Device>
    <ChipInfo Vendor="Rafael"
      Name="RT58x_512KB"
      Core="JLINK_CORE_CORTEX_M3"
      WorkRAMAddr="0x20000000"
      WorkRAMSize="0x1000" />
    <FlashBankInfo Name="Internal Flash"
      BaseAddr="0x00007000"
      MaxSize="0x00077000"
      Loader="RT58x_512KB.FLM"
      LoaderType="FLASH_ALGO_TYPE_CMSIS"
      AlwaysPresent="1" />
  </Device>
  <Device>
    <ChipInfo Vendor="Rafael"
      Name="RT58x_2MB"
      Core="JLINK_CORE_CORTEX_M3"
      WorkRAMAddr="0x20000000"
      WorkRAMSize="0x1000" />
    <FlashBankInfo Name="Internal Flash"
      BaseAddr="0x00007000"
      MaxSize="0x001F7000"
      Loader="RT58x_2MB.FLM"
      LoaderType="FLASH_ALGO_TYPE_CMSIS"
      AlwaysPresent="1" />
  </Device>
</DataBase>

```



### 3.4 VS Code Extension Configuration

Open the compressed file extensions.rar, extract the extensions folder and overwrite it to the location of the VS Code extension kit. By default, VS Code extension kit is installed under C:\Users\your\_PC\_name\.vscode.



### 3.5 VS Code Configuration

Execute VSCode, press the key “Ctrl + Shift + P”, type the text “settings”, and select the option “Preferences: Open Settings(JSON)”.





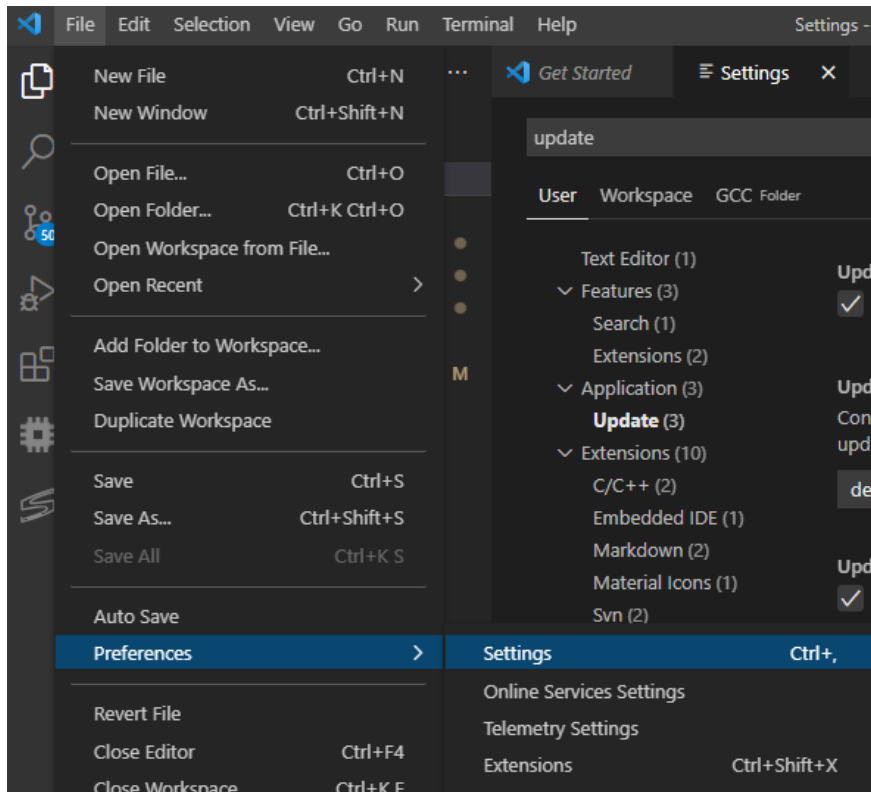
Open `settings.json`, add the following settings at the end, save and close the settings file.

```
"EIDE.JLink.InstallDirectory": "D:\\JLink",  
"EIDE.JLink.DeviceXmlPath": "D:\\JLink",  
"EIDE.ARM.GCC.InstallDirectory": "D:\\GNU Arm Embedded Toolchain\\10 2021.10",  
"cortex-debug.armToolchainPath": "D:\\GNU Arm Embedded Toolchain\\10 2021.10\\bin",  
"cortex-debug.JLinkGDBServerPath": "D:\\JLink\\JLinkGDBServerCL.exe",
```

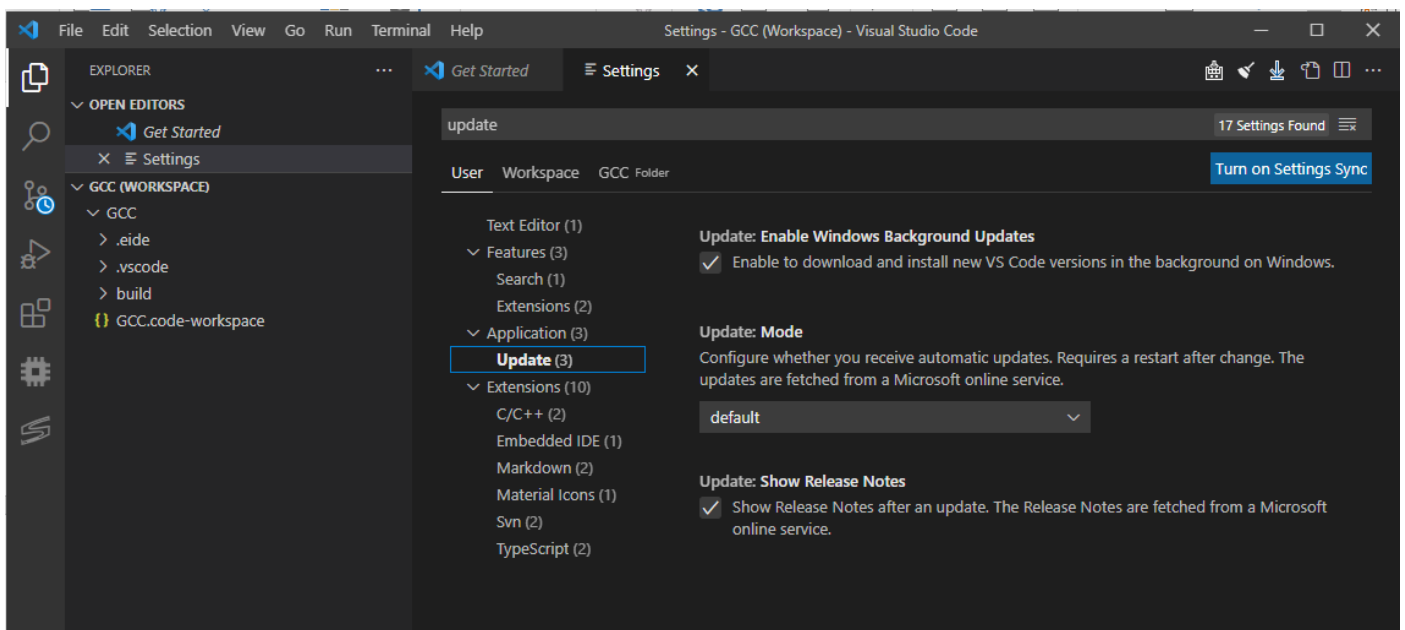
```
25  
26     "EIDE.JLink.InstallDirectory": "D:\\JLink",  
27     "EIDE.JLink.DeviceXmlPath": "D:\\JLink",  
28     "EIDE.ARM.GCC.InstallDirectory": "D:\\GNU Arm Embedded Toolchain\\10 2021.10",  
29     "cortex-debug.armToolchainPath": "D:\\GNU Arm Embedded Toolchain\\10 2021.10\\bin",  
30     "cortex-debug.JLinkGDBServerPath": "D:\\JLink\\JLinkGDBServerCL.exe",  
31 }
```

## 3.6 VS Code Disable Update

File→Preference→Setting



Select Update Option



## Update Mode

Default : Auto Update

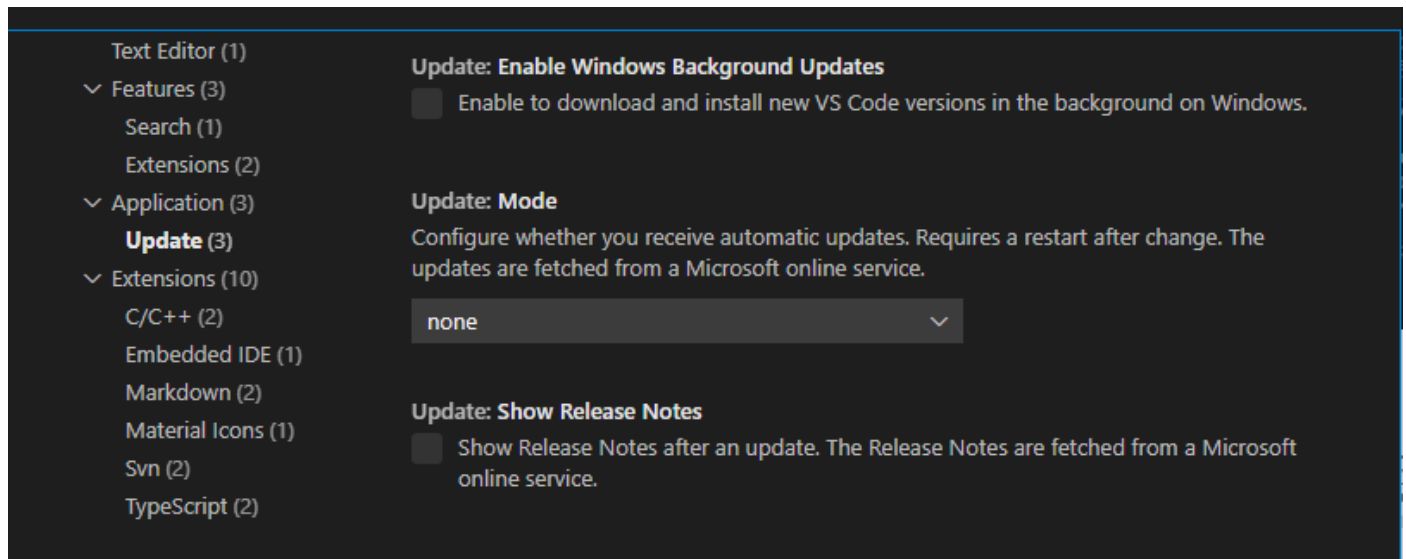
None : Disable Update

Manual : User Update

Start : Check Update

Update : Enable Windows Background Updates

Update : Show Release Notes



### 3.7 Project Builder Configurations

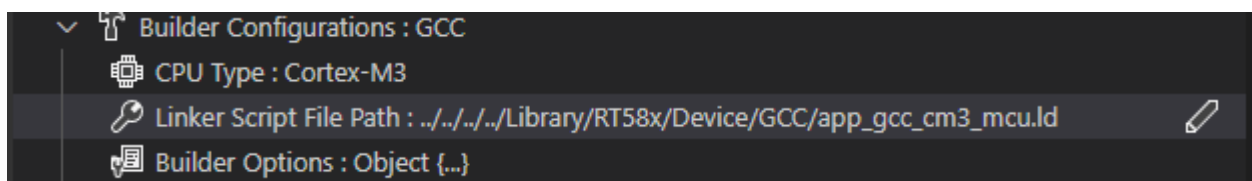
Default Linker Script File Path: Library\RT58x\Device\GCC\app\_gcc\_cm3\_mcu.ld.

According to the RafaelMicro MCU flash size to select the link script file.

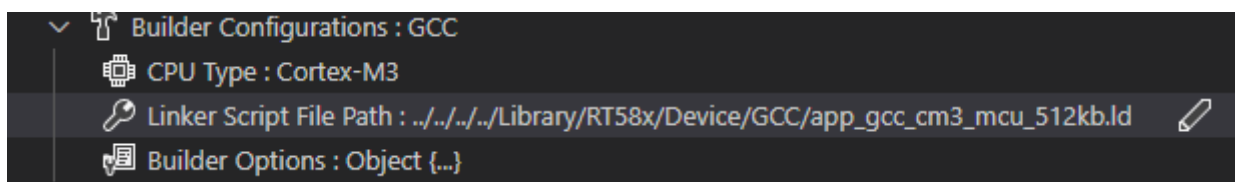
If you are using 512K Flash MCU, modify the path of the link script file to "Library RT58x Device GCC app\_gcc\_cm3\_mcu\_512KB.ld".

Using the 2MB Flash MCU, modify the path of the link script file to "Library RT58x Device GCC app\_gcc\_cm3\_mcu\_2mb.ld".

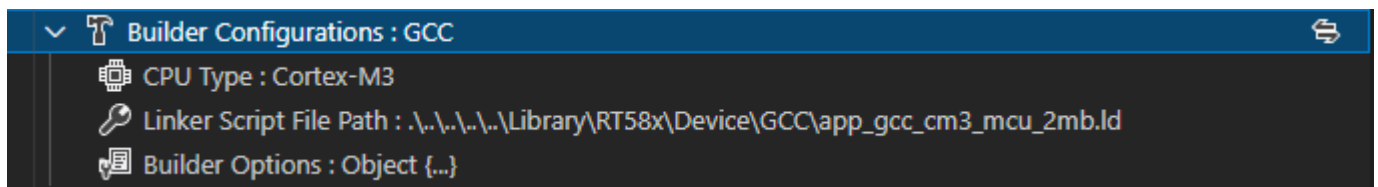
Default Linker Script File Path:



512K Flash MCU Linker Script File Path:



2MB Flash



## Revision History

Revision	Description	Owner	Date
1.0	Initial version.	Chiaho Hu	2022/02/15
1.1	Revise vs code kit download description	Ives Lee	2022/07/15
1.2	Add edit JLinkdevices.xml for download code Add change linker script file for different flash size mcu	Ives Lee	2022/07/18
1.3	Revise JLinkdevices.xml and linker script description Add disable Vs code update	Ives Lee	2022/05/03

© 2021 by Rafael Microelectronics, Inc.

All Rights Reserved.

Information in this document is provided in connection with **Rafael Microelectronics, Inc.** ("**Rafael Micro**") products. These materials are provided by **Rafael Micro** as a service to its customers and may be used for informational purposes only. **Rafael Micro** assumes no responsibility for errors or omissions in these materials. **Rafael Micro** may make changes to this document at any time, without notice. **Rafael Micro** advises all customers to ensure that they have the latest version of this document and to verify, before placing orders, that information being relied on is current and complete. **Rafael Micro** makes no commitment to update the information and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to its specifications and product descriptions.

THESE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, RELATING TO SALE AND/OR USE OF **RAFAEL MICRO** PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, CONSEQUENTIAL OR INCIDENTAL DAMAGES, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. **RAFAEL MICRO** FURTHER DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS. **RAFAEL MICRO** SHALL NOT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUES OR LOST PROFITS, WHICH MAY RESULT FROM THE USE OF THESE MATERIALS.

**Rafael Micro** products are not intended for use in medical, lifesaving or life sustaining applications. **Rafael Micro** customers using or selling **Rafael Micro** products for use in such applications do so at their own risk and agree to fully indemnify **Rafael Micro** for any damages resulting from such improper use or sale. **Rafael Micro**, logos and **RT568** are **Trademarks** of **Rafael Microelectronics, Inc.** Product names or services listed in this publication are for identification purposes only, and may be trademarks of third parties. Third-party brands and names are the property of their respective owners.