GigaDevice Semiconductor Inc.

GD32F1x0
ARM® Cortex™-M3 32-bit MCU

Application Note
AN007

Table of Contents

List of Figures

List of Tables

1. Introduction

GigaDevice.GD32F1x0_DFP.2.0.0.pack is used to support GD32F1x0 series chips.

- Online installation (highly recommended).
- Locally installation.
- Automatically generate GD32F1x0 series MCU and the corresponding feature information list.
- Automatically matching the corresponding flash algorithm to the selected chip.
- This Pack provides a complete set of library and components of GD32F1x0 series chips in the Manage Run-Time Environment (RTE). The users can select the required library files.
- Support checking register status in the Debug mode.
- Support getting documents from the Books Tab.

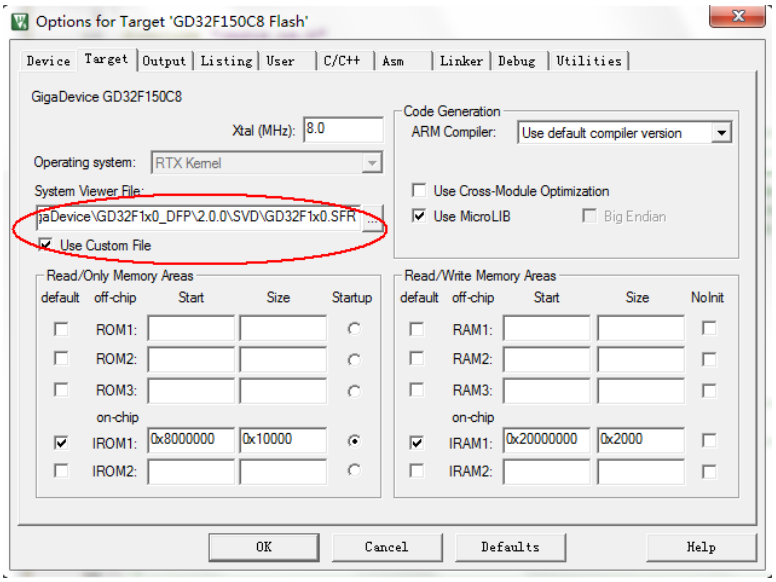
1.1. Versions of Keil

The Pack is suitable for Keil v5.15a and above versions. This note is operated on Keil v5.18a as an example.

Note: For versions Keil v5.13 and Keil v5.14, there are some problems:

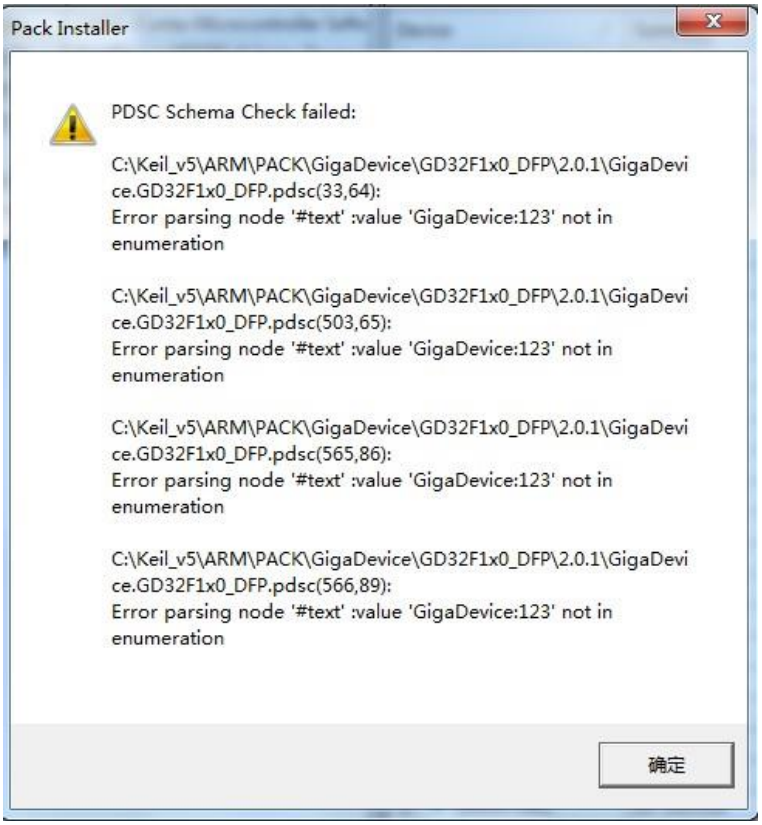
1. Not supported to get documents from the Books Tab.
2. Users can't observe registers in the Debug mode by calling svd file. To solve this problem, the users should change file .svd to .sfr as shown in figure 1-1.

Figure 1-1. Change file .svd to .sfr



3. Schema check to the pack, there will be an error, shows in figure 1-2.

Figure 1-2. Schema check error

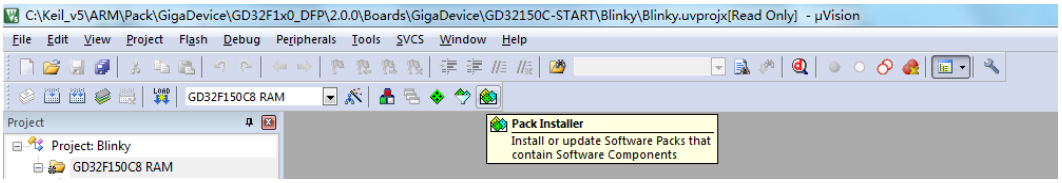


2. How to download and Install Pack by Pack Installer

To create a new Keil5 project, GigaDevice.GD32F1x0_DFP.2.0.0.pack should be downloaded and installed first.

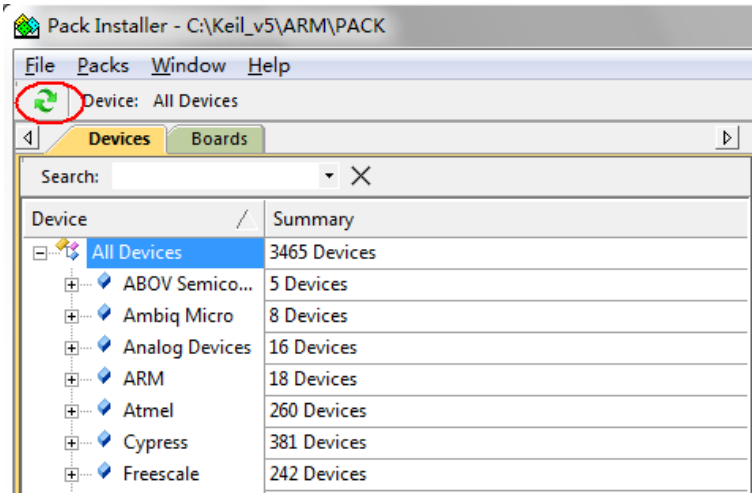
1. Let's open the Keil5, and select the Pack Installer, as shown in figure 2-1.

Figure 2-1. Pack installer step 1



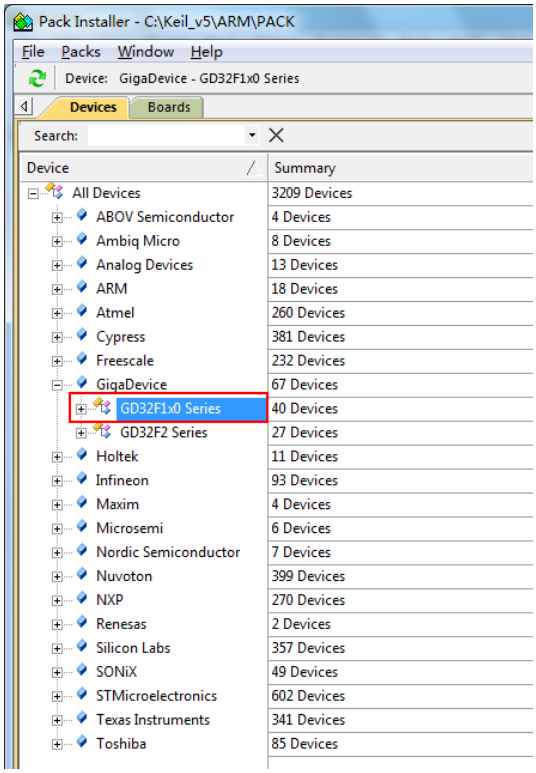
2. Click on the Pack Installer button to enter the interface as shown in figure 2-2. Click the refresh button before downloading the latest Pack.

Figure 2-2. Pack installer step 2



3. As shown in figure 2-3, find out and then choose GD32F1x0 Series in Devices Tab.

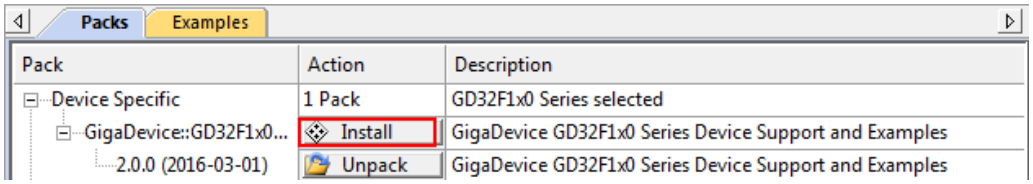
Figure 2-3. Pack selection



4. Correspondingly it will show GigaDevice::GD32F1x0_DFP in drop-down menu of Device Specific in figure 2-4. Click on the Install button behind GigaDevice::GD32F1x0_DFP to download the Pack of GD32F1x0 series.

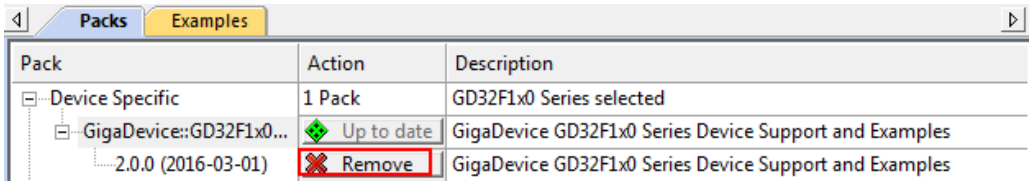
   The Pack will then be downloaded to the directory (.\Keil_v5\ARM\Pack\.Download), and then the installation will be done.

Figure 2-4. Pack download



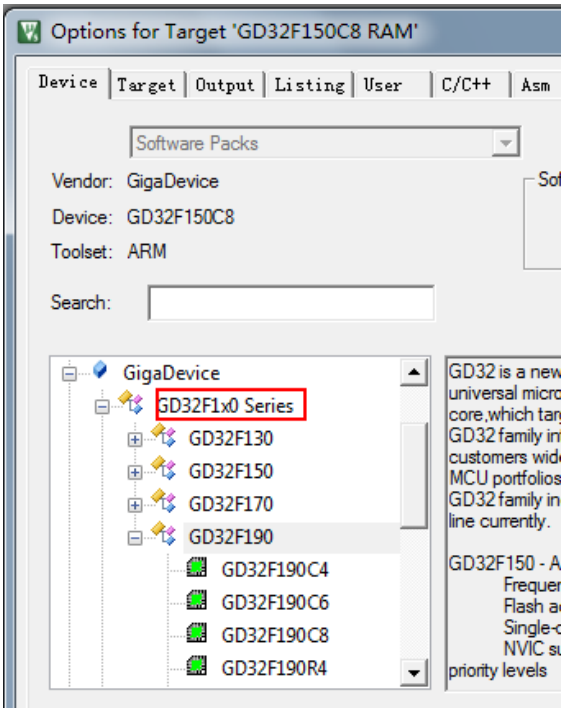
5. If you want to remove the Pack from Keil5, Click on the Remove button, but this operation will not clear the executable file in the directory of Keil5. Thus if you has not manually deleted the executable file, re-install the Pack can be done off-line without the download process.

Figure 2-5. Remove pack



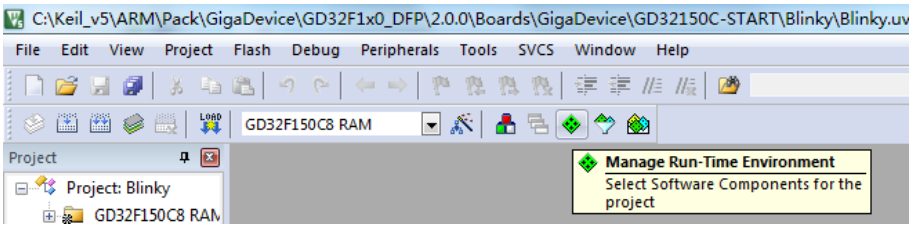
When the Pack installation is completed, the related GD32F1x0 components will be added to the RTE (as shown in figure 2-5) interface, as shown in figure 2-6.

Figure 2-6. MCU lists of GD32F1x0 series



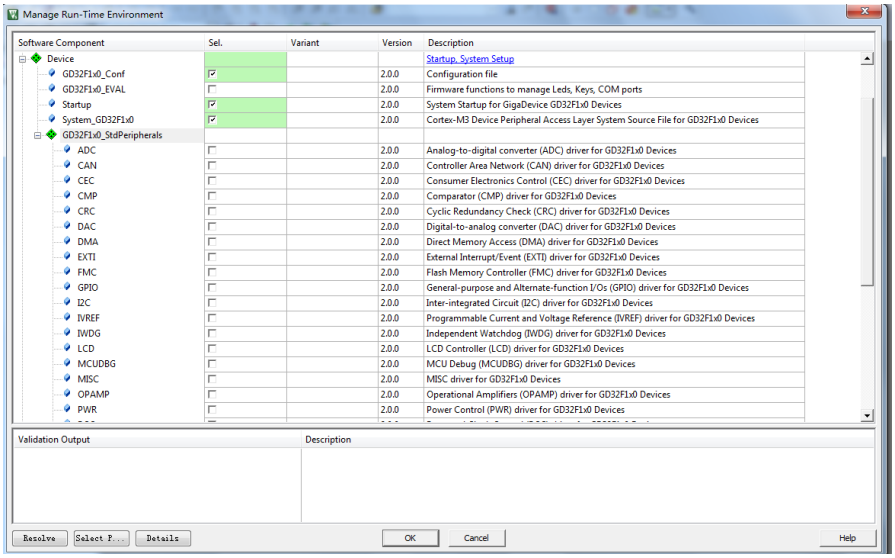
Click the button shows in figure 2-7 to open RTE(Manage Run-Time Environment).

Figure 2-7. Manage run-time environment



In figure 2-8, users can see the related components of GD32F1x0.

Figure 2-8. RTE interface

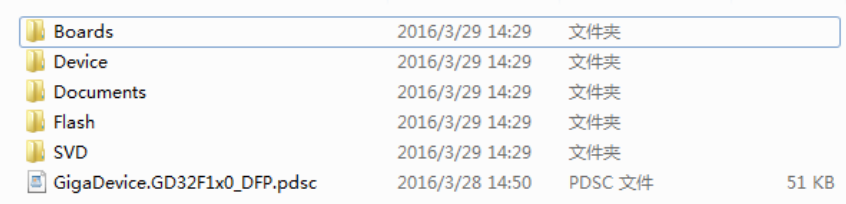


Note: If the corresponding chip selection is still unavailable after the Pack installation, you may reopen the project to loading Pack.

3. Structure of GD32Fxxx pack

After the installation of GigaDevice.GD32F1x0_DFP.2.0.0.pack, the folder directory can be seen in directory of Keil5 (.\Keil_v5\ARM\Pack\GigaDevice\GD32F1x0_DFP\2.0.0), as shown in figure 3-1.

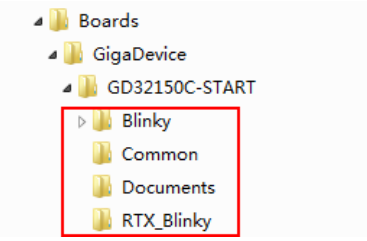
Figure 3-1. Structure of Pack



1. Boards folder includes available evaluation board resources, currently only provide GD32150C-START evaluation board, shows in figure 3-2, the contents of the folder are shows below:

   - Blinky folder contains the Keil general demo.
   - The Common folder contains Button.c and LED.c, which are used to realize operations about flashing lights and keystrokes in demo.
   - The Documents folder provides GD32150C-START evaluation board schematic diagram.
   - RTX_Blinky folder contains the Blinky demo which is based on the operating system.

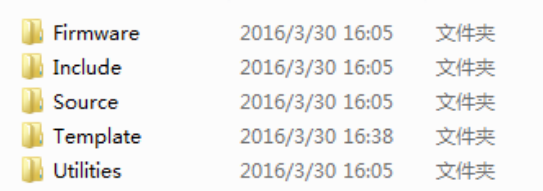
Figure 3-2. Structure of Boards Folder



2. Device folder contains the content as shown in figure 3-3:

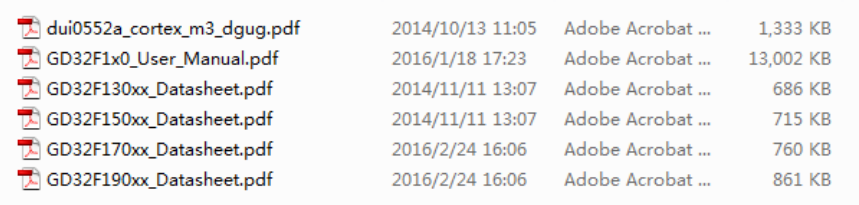
- Firmware folder contains the GD32F1x0 peripherals library.
- Include folder contains the global header file gd32f1x0.h and the system header file system_gd32f1x0.h.
- Source folder contains the system C file system_gd32f1x0.c and the startup file startup_gd32f1x0.s.
- Template folder contains a template project.
- Utilities folder contains gd32f1x0_eval.c and gd32f1x0_eval.h.

Figure 3-3. Structure of Device Folder



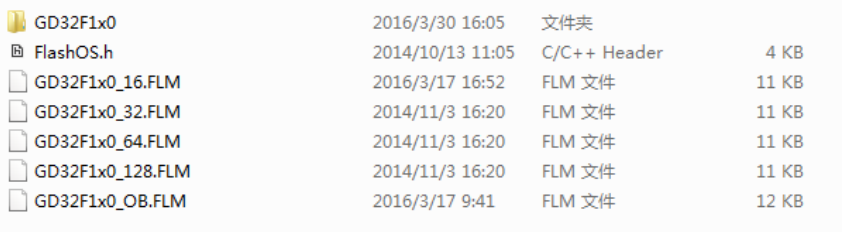
3. Documents folder contains the documents for Books Tab, as shown in figure 3-4.

Figure 3-4. Documents Folder content



4. Flash folder contains algorithms of flash, as shown in figure 3-5.

Figure 3-5. Flash Folder content



5. SVD (System View Description) file.

6. PDSC (Pack Description) file.

4. How to play demo

After installing GigaDevice.GD32F1x0_DFP.2.0.0.pack, the users can use the two Demos based on GD321x0C-START board in the Pack, one is called Blinky without operating system, the other is RTX_Blinky based on RTX Kernel operating system.

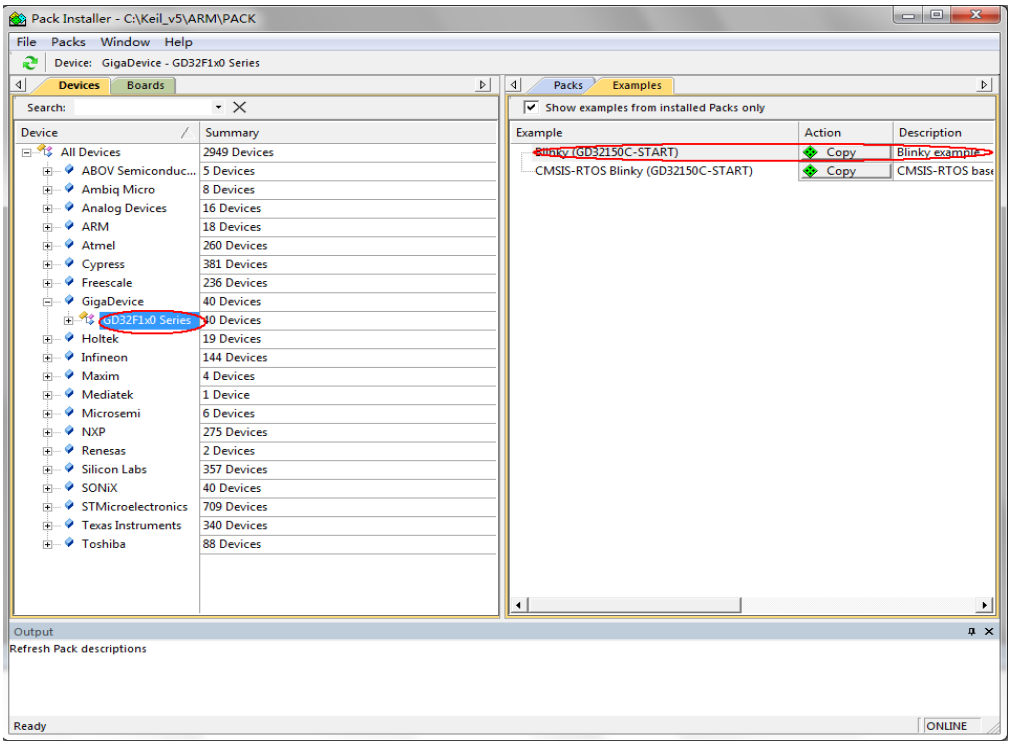
The steps of opening the Demo Blinky:

1. Open Keil5 and then open the Pack Installer.

2. In the open dialog, find GigaDevice in Devices Tab and select GD32F1x0 Series, then the Demos will be shown in the Examples Tab. As shown in figure 4-1, choose Blinky and click Copy.

3. Select store path in the pop-up dialog. The default store path is the new folder named Boards in D:\Documents. Click OK and the Project will open.

Figure 4-1. Blinky select



Open the project, unfold the folders in the Project Tab, as shown in figure 4-2, it includes the following documents:

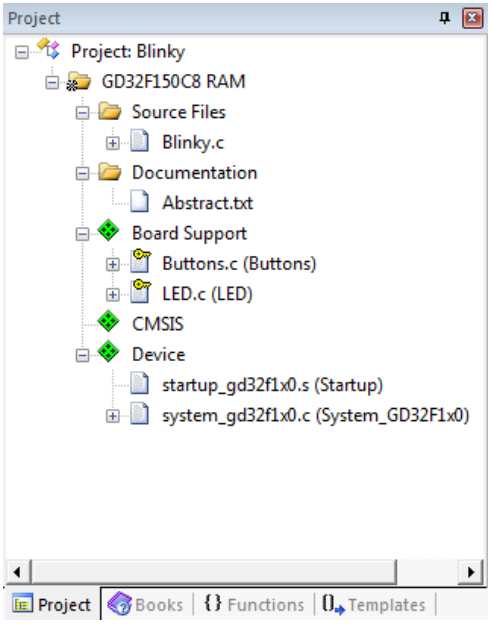
- Source Files
- Documentation
- Board Support Components
- CMSIS Components

- Device Components

Select the components in Manage Run-Time Environment:

- Source Files folder includes the Blinky.c which is the source file of Demo.
- Documentation folder includes the demo instructions.
- Board Support components include two parts: Buttons provides Buttons.c and related header files, LED provides LED.c and related header files, they are used for test on GD321x0C-START board.
- CORE in CMSIS components provides the required functions and variables of core.
- Device components include three parts: GD32F1x0_Conf provides module configuration of project, Startup provides startup_gd32f1x0.s, System_GD32F1x0 provides system_gd32f1x0.c and related header files.

Figure 4-2. Structure of Blinky



To open the Demo CMSIS-RTOS Blinky, the procedure is similar to Blinky.

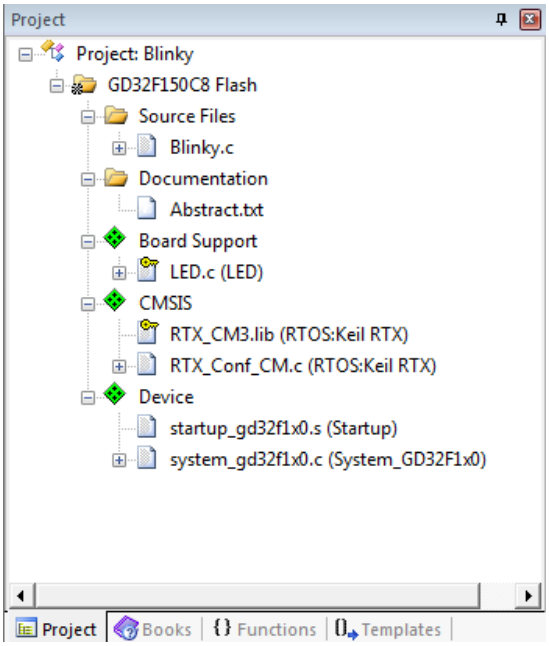
Open the project, unfold the folders in the Project Tab, as shown in figure 4-3, it includes the following documents:

- Source Files
- Documentation
- Board Support components
- CMSIS components
- Device components

Select the components in Manage Run-Time Environment:

- CORE in CMSIS components provides the required functions and variables of core. Components RTOS provides RTX_CM3.lib and RTX_Conf_CM.c.
- The remained steps are similar to Blinky.

Figure 4-3. Structure of CMSIS-RTOS Blinky



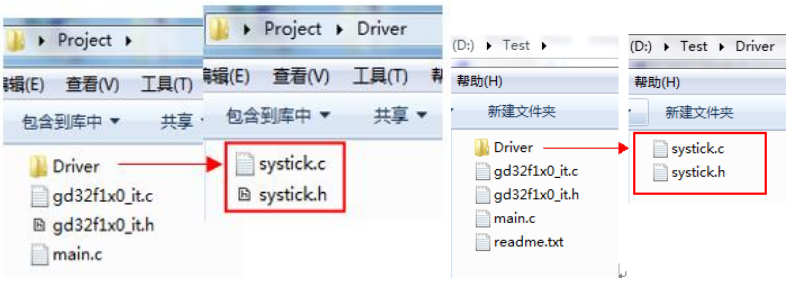
5. How to create a new project

Download the Pack and install it, then you can create a new project.

The detailed steps are below:

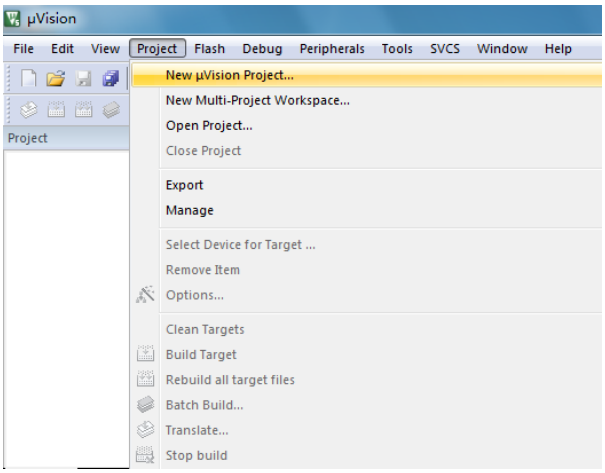
1. Create a new folder, named such as Project, to save the new created project. As shown in figure 5-1, put the application program in the Project folder, and create another new folder named such as Driver, put in the related driver files.

Figure 5-1. Content of new project



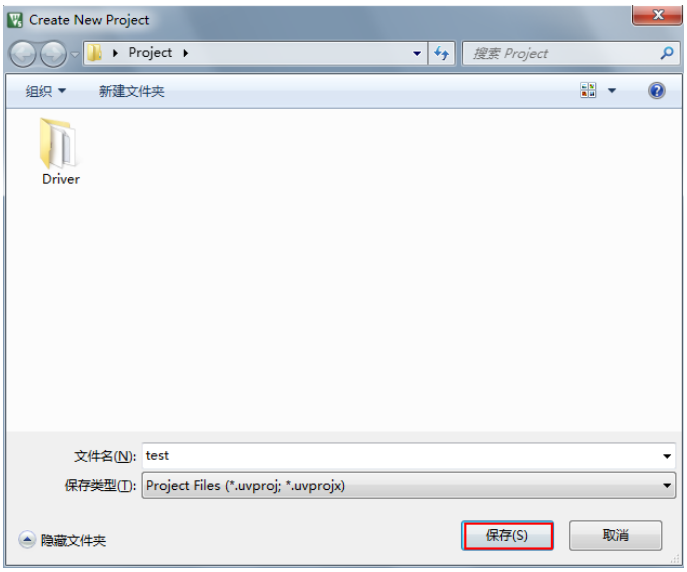
2. Open Keil5, shows in Figure 5-2, select New uVision Project in the Project menu.

Figure 5-2. Menu of create a new project



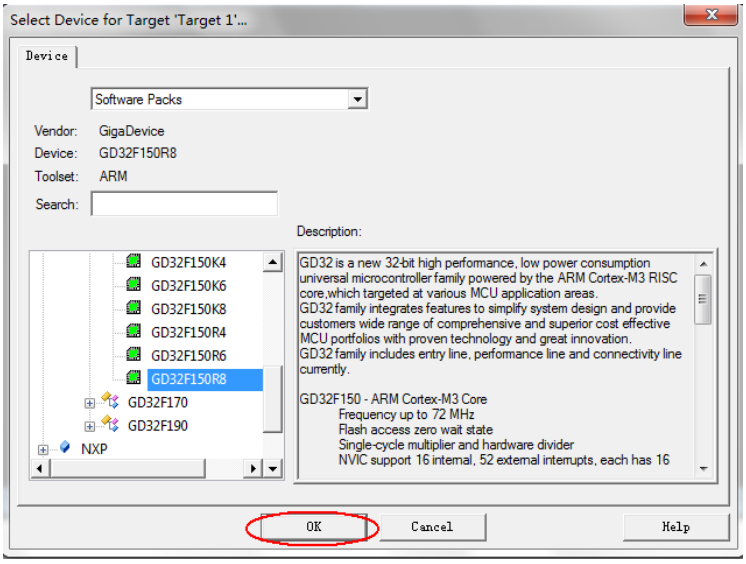
3. Select the new created folder Project as target location in the pop-up dialog, and name it test, as shown in figure 5-3. The driver files are stored in the folder called Driver.

Figure 5-3. Save the new projrect



4. After clicking save, a dialog will pop out for project configuration and chip selection. In this test, we select GD32F150R8, click OK.

Figure 5-4. Selection of Devices

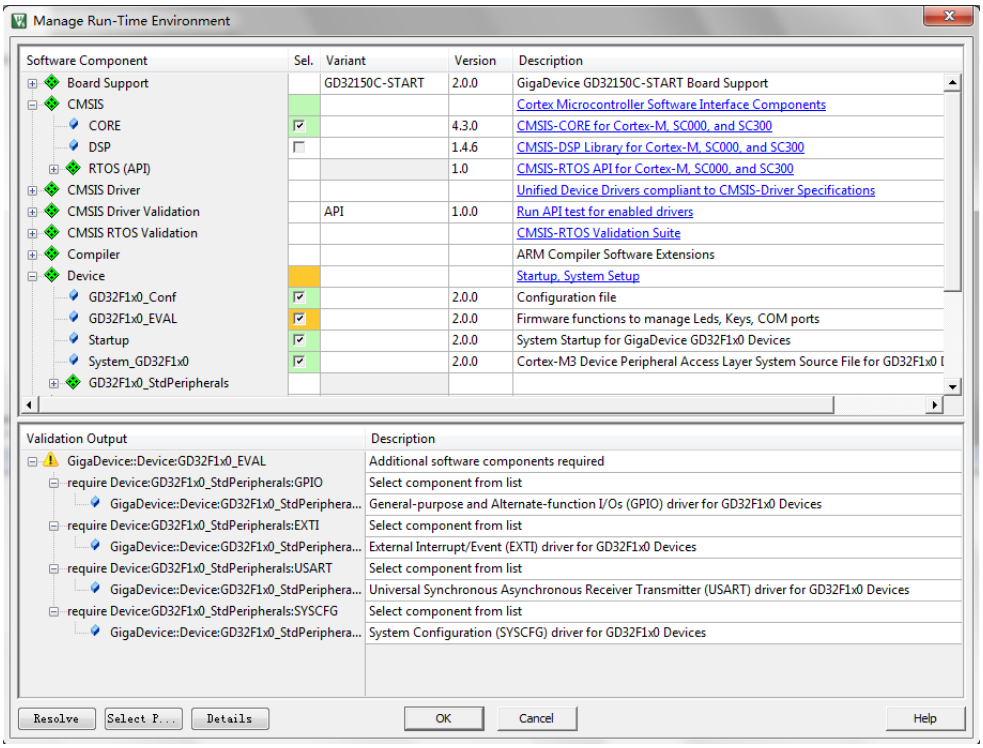


5. Then the RTE will pop out, users can select the required modules.

   The components which must be selected are CORE in CMSIS components, Startup, System_GD32F1x0, GD32F1x0_Conf in Device components.

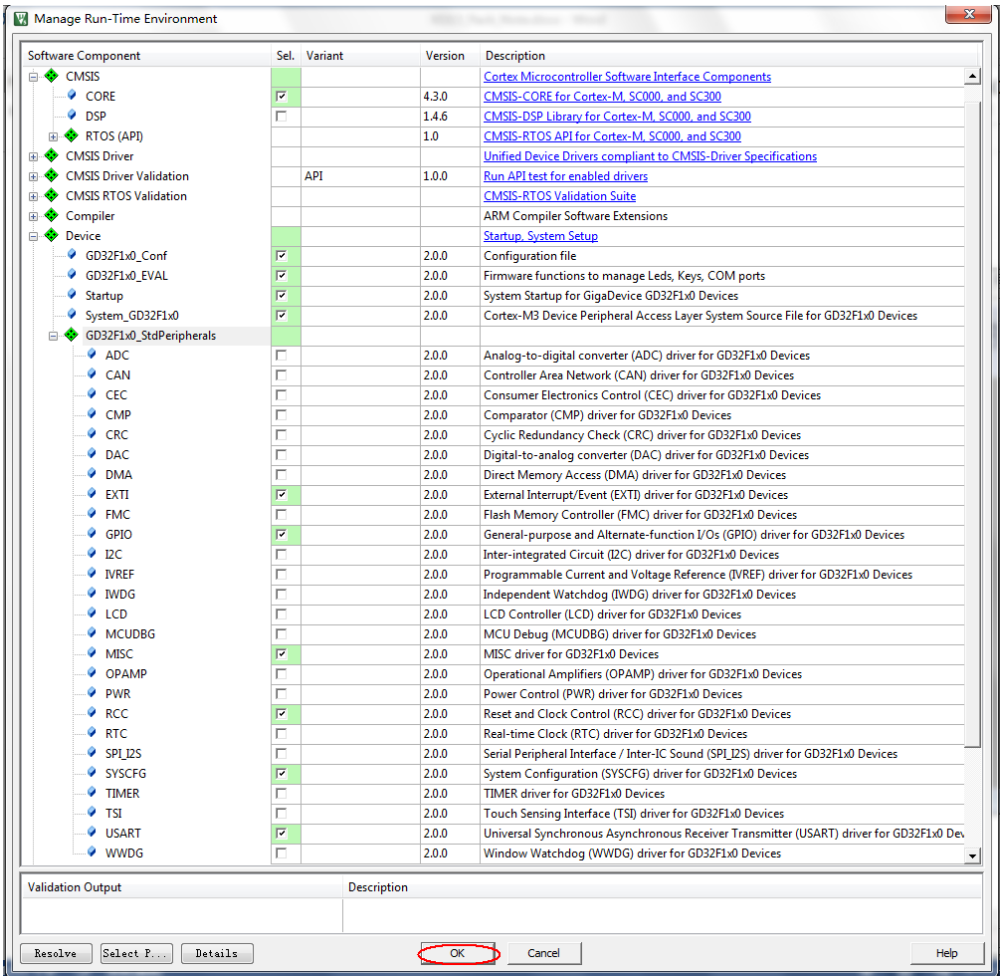
   If the EVAL related functions have been used in the project, then add GD32F1x0_EVAL in Device components. Finally, add the required modules according to the warnings.

Figure 5-5. Warnings



As shown in figure 5-6, when there is no warning, click OK.

Figure 5-6. Selection of components



6. Add the source files, as shown in figure 5-7. In the new created project test, there is no need to add the header file include path, shown as figure 5-8.

Figure 5-7. Add source files

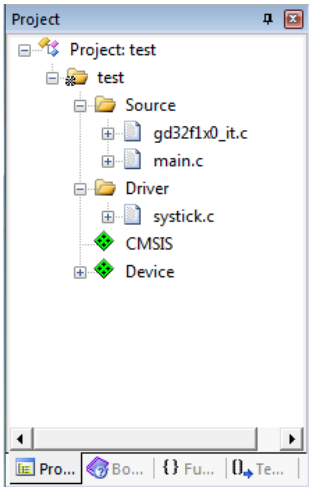
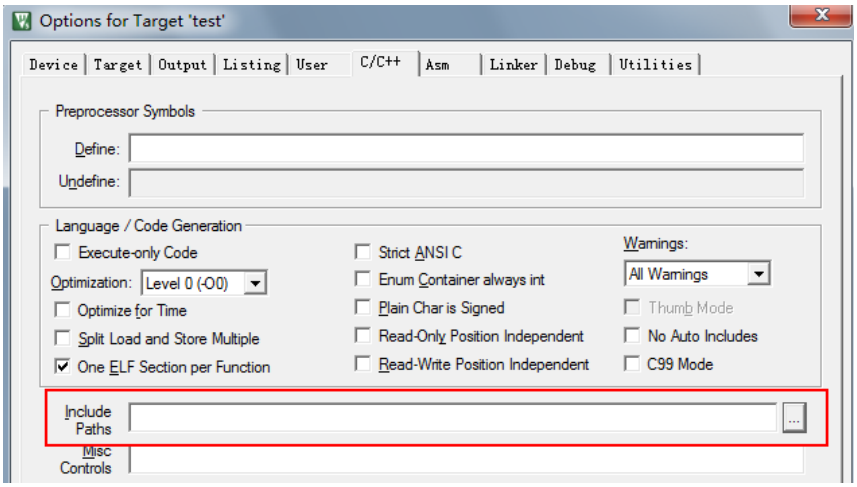
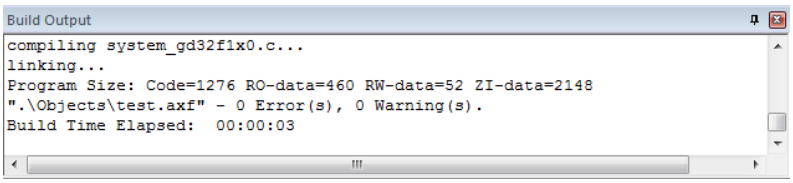


Figure 5-8. Header file include path



7. Compile the project, the result is shown in figure 5-9. If there is any error, maybe you have added modules incompletely. Add the related modules according to the error, then compile the project again.

Figure 5-9. Compling result

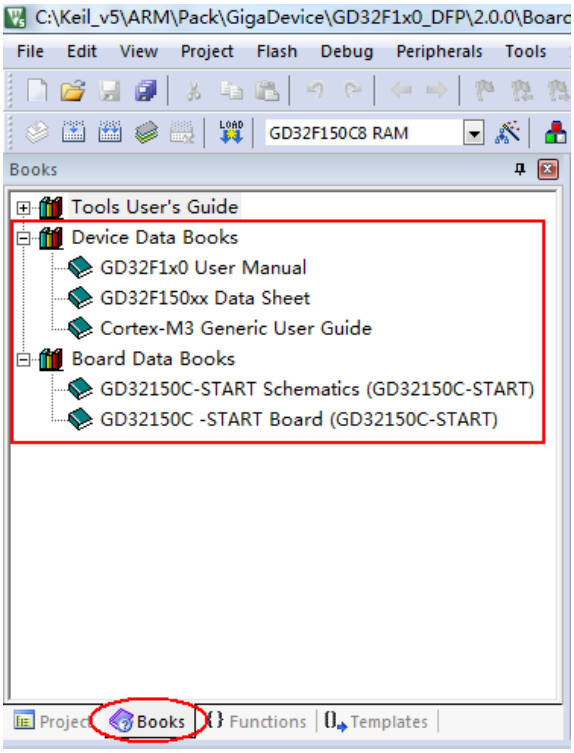


8. Select a debug tool, then you can download program or debug online.

6. How to turn to the related documents by Books Tab

Open the Keil5 project, switch to the Books Tab, as shown in figure 6-1. Click on the book you want to check to turn to the corresponding document or web page.

Figure 6-1. Books Tab



7. How to check register status in Debug mode

Click the debug button to enter the Debug mode, Keil5 supports the use of a graphical interface to view all register status in real time.

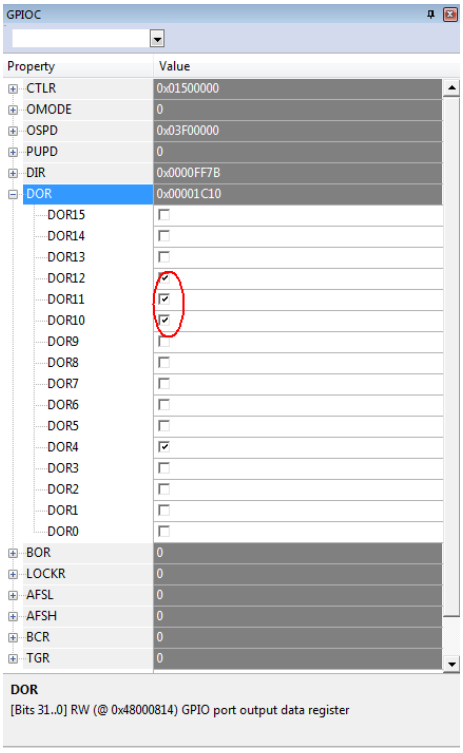
Steps shown as below:

1. As shown in figure 7-1, click the System Viewer Windows button, select the register you want to observe. An example, to observe the DOR of GPIOC register.

Figure 7-1. System Viewer Windows

2. The Periodic Windows Update in View menu is selected, then the corresponding registers will refresh periodically. You can observe changes of the IO ports status and other register status.

Figure 7-2. Register status window



8. How to open Keil4 project in Keil5

The information of devices, peripherals, tools, chains and others which called in Keil5 project are all included in Pack, but the call to related file path in Keil4 is different from Keil5, in this circumstance, it's necessary to do the compatible process and there are two methods to realize the process:

Method 1: Change the project property setting, this method is recommended.

Note: Once the project property setting is changed, it can't be opened and built in Keil4.

Method 2: Install GigaDevice patches Keil4_GD32F1x0_ADDON.exe to load related documents.

Note: The project property setting is not changed, it can still be opened and built in Keil4.

1.2. Compatible method 1

1. Using Keil5 to open a project created in Keil4.
2. Select the Manage menu in the Project menu, click on the Migrate to Version 5 Format.. menu, as shown in figure 8-1, then a prompt information will pop out, as shown in figure 8-2.

Figure 8-1. Change Project Property Setting

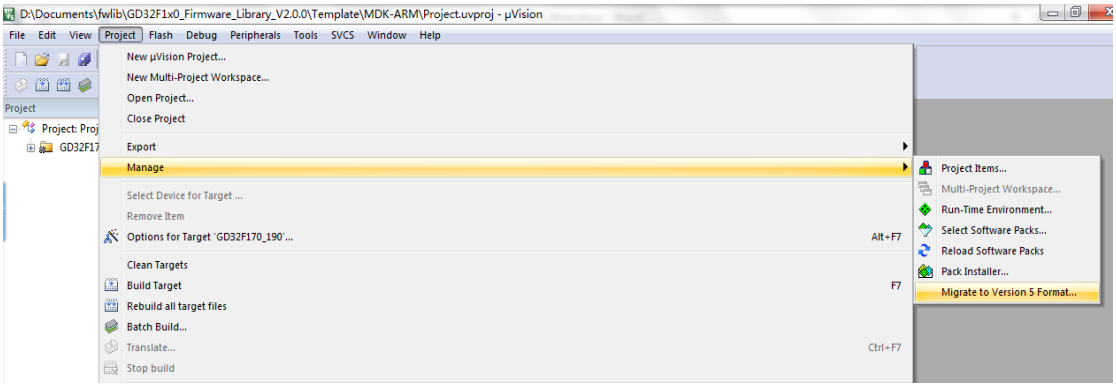
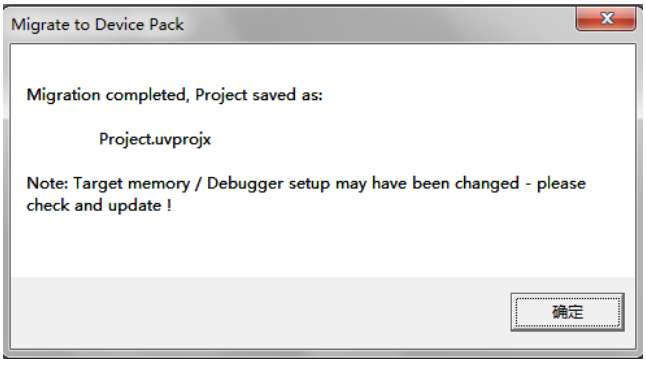
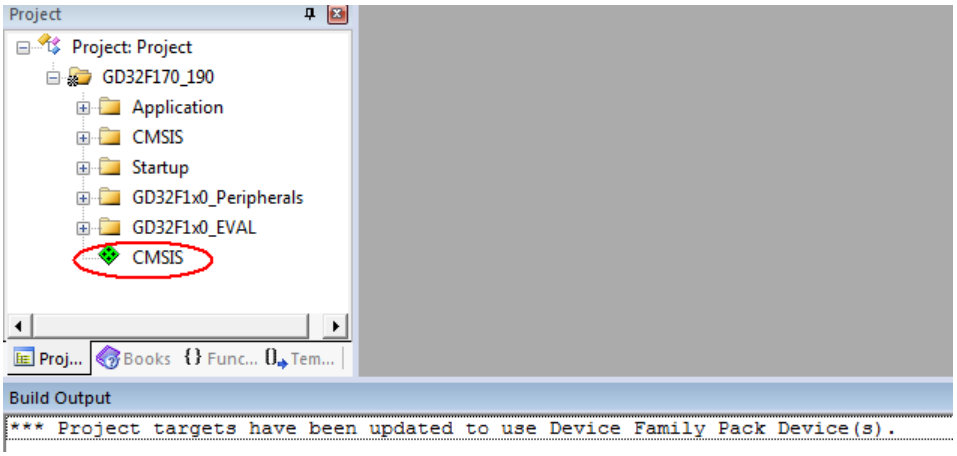


Figure 8-2. Prompt Information



3. Click the OK button, then a CMSIS component will be added to the Project Tab, also the Build Output window will print the information to declare that Keil4 related support of project has been updated to Keil5 Pack support, as shown in figure 8-3.

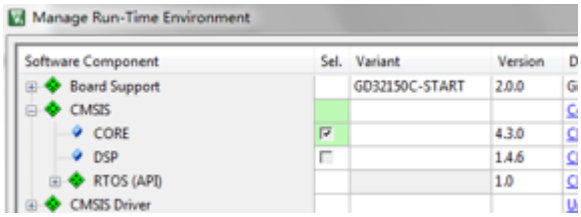
Figure 8-3. Project after Compatibility



4. Compile the project.

Note：For versions of Keil v5.13 and Keil v5.14, after migrate to Version 5 format, it needs to enter RTE environment, to select the CORE in CMSIS components, as figure shows below,

Figure 8-4. Select CMSIS CORE



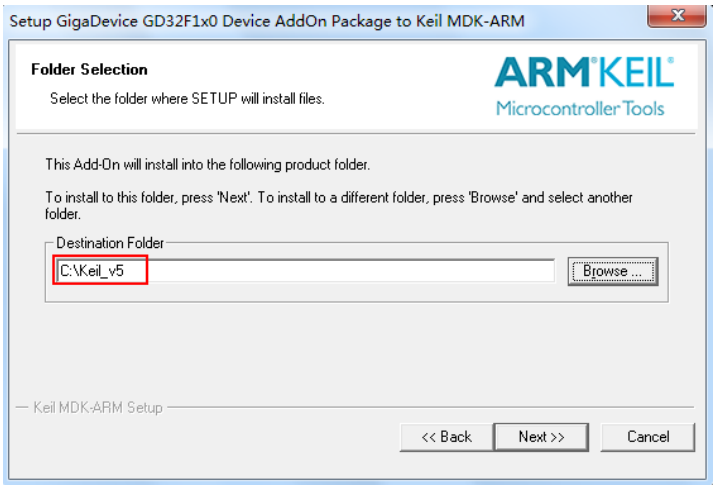
Then compile the project.

1.3. Compatible method 2

1. Download file Keil4_GD32F1x0_ADDON.exe on GigaDevice official website (http://gd32mcu.21ic.com/documents), as shown in figure 8-5. Install the executable file in Keil_v5 installation directory, as shown in figure 8-6.

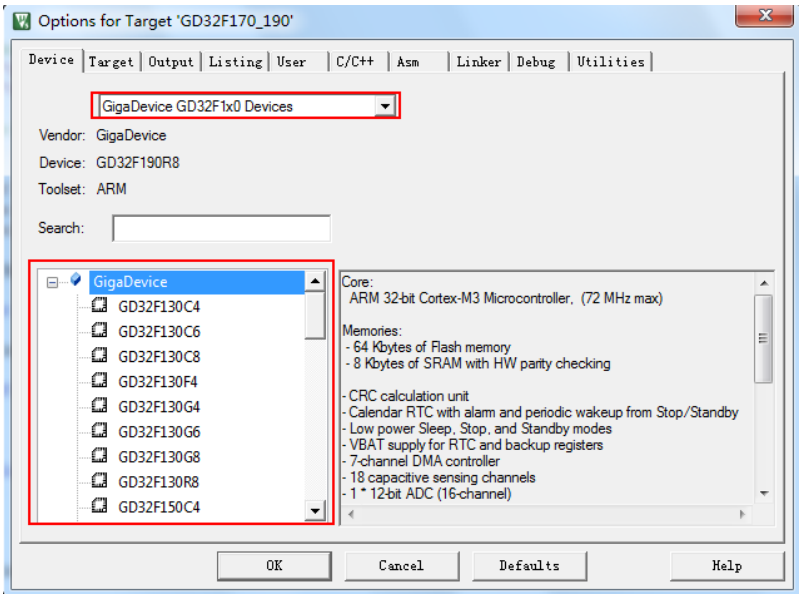
Figure 8-5. GD32F1x0_ADDON executable File

Figure 8-6. GD32F1x0_ADDON installation



2. Using Keil5 to open a project created in Keil4. Select GigaDevice GD32F1x0 Devices in Option for the Target Device, then GigaDevice GD32F1x0 series product selection will be listed in the table, as shown in figure 8-7, select your required device.

Figure 8-7. Product Series Selection

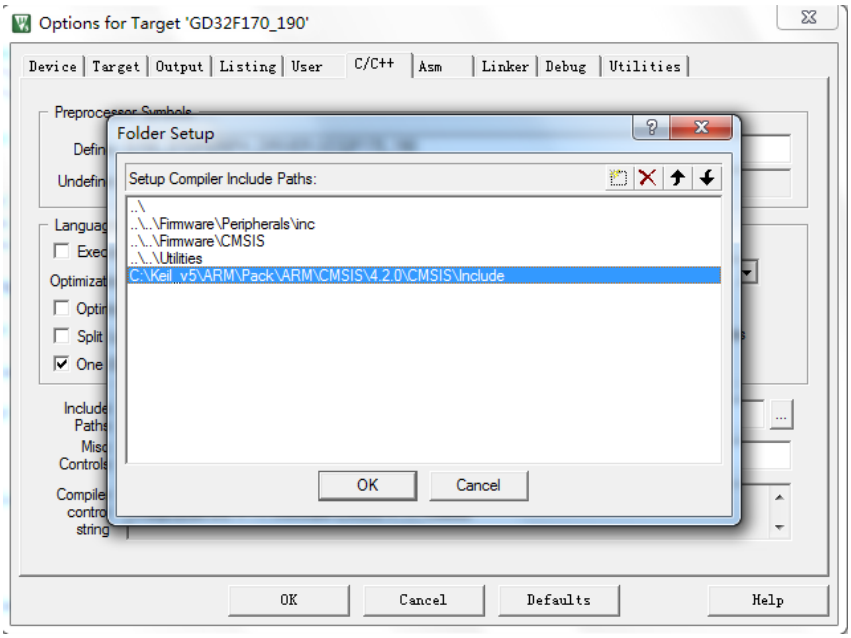


3. Compile the project, there may be a lot of errors, as shown in figure 8-8. The cause is that path of core_cmInstr.h is different in Keil5 and Keil4, so it is necessary to add the file path of core_cmInstr.h to C/C++ in Option for Target, as shown in figure 8-9.

Figure 8-8. Compile Result Error

```
..\..\Firmware\CMSIS\core_cm3.h(147): error:  #5: cannot open source input file "core_cmInstr.h": No such file or directory
  #include <core_cmInstr.h>               /* Core Instruction Access                          */
..\..\Firmware\Peripherals\src\gd32f1x0_dma.c: 0 warnings, 1 error
compiling gd32f1x0_fmc.c...
..\..\Firmware\CMSIS\core_cm3.h(147): error:  #5: cannot open source input file "core_cmInstr.h": No such file or directory
  #include <core_cmInstr.h>               /* Core Instruction Access                          */
..\..\Firmware\Peripherals\src\gd32f1x0_fmc.c: 0 warnings, 1 error
```

Figure 8-9. Add File Path



9. Version history

Table 9-1. Version history

Version number	Description	Date
1.0	Initial Release	Apr.6, 2016