

# HOLTEK HT32 Series

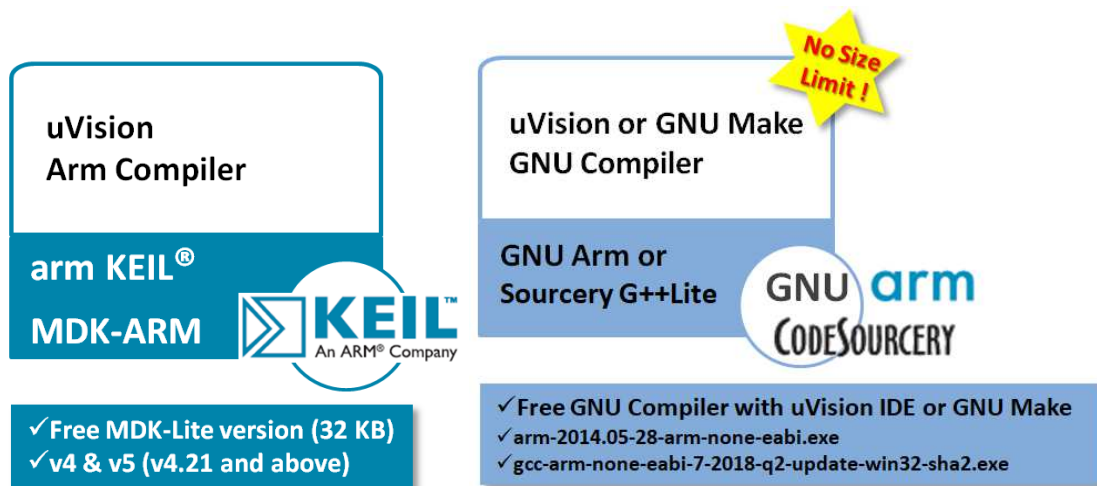
Setup Guide of Keil® MDK-ARM and  
GNU Compiler/Make with Starter Kit

## Revision History

Version	Date	Descriptions
0.01	2016-09-28	✓ Initial version
0.02	2018-07-14	✓ Add a description of SourceryG++Lite Compiler with Keil MDK-ARM
0.03	2018-08-02	✓ Fix link and typing error
		✓ Add a description of GNU Arm Compiler with Keil MDK-ARM
0.04	2018-09-30	✓ Add a description of GNU Arm Compiler with GNU Make tool
0.05	2019-01-30	✓ Add "Add path to environment variable" in the Step 3.

## 1. Overview

This document shows how to install and configure the Keil MDK-ARM with HOLTEK HT32 Starter Kit. It also includes how to use the “SourceryG++Lite” and “GNU Arm” compiler with Keil MDK-ARM or GNU Make tool. For detailed information, please refer to the quick start guide at the HT32 Development Resource, “\\Document\_QuickStartGuide”.



## 2. HT32 Development Resource Download Address

Google	<a href="https://goo.gl/Qg7kHr">https://goo.gl/Qg7kHr</a>
HOLTEK	<a href="http://mcu.holtek.com.tw/ht32/resource/">http://mcu.holtek.com.tw/ht32/resource/</a>

Before starting the operation, please download the HT32 Development Resource zip file from the above link. The file name is as follows.

**“HT32\_M0+\_vxxxxxxxx.zip” or “HT32\_M3\_vxxxxxxxx.zip”**

where xxxxxxxx equal to the release date.

You need following files for the subsequent operations.

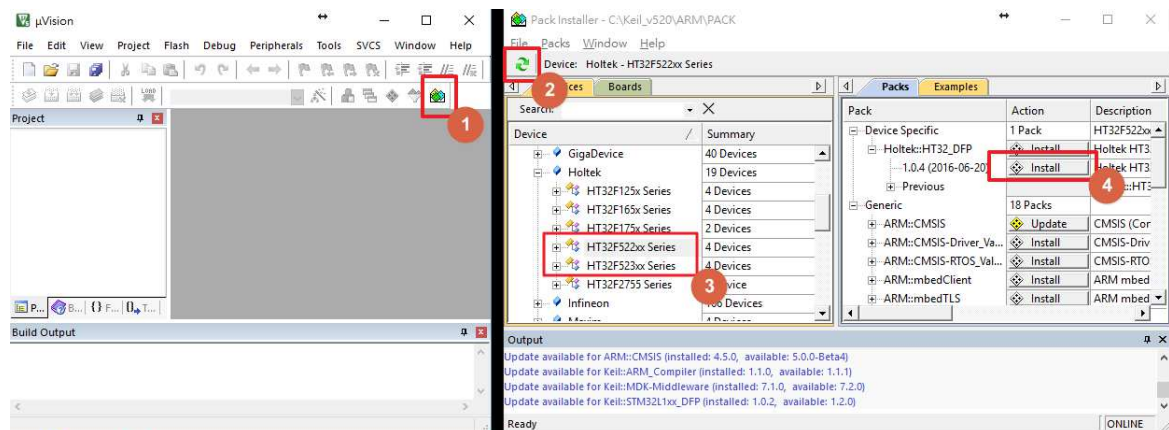
Firmware Library	\\Firmware_Library\HT32Fxxxxx_FWLib_vnnn_nnn.zip
Keil CMSIS PACK file	\\Tools\Holtek.HT32_DFP.n.n.n.pack
Virtual COM Driver	\\Tools\HT32_VCP_Driver_v110.exe

### 3. Setup Keil MDK-ARM with HT32 Starter Kit

**Step1.** Install Keil MDK-Lite, 32K version. You can download the setup EXE file from the Keil web site. <https://www.keil.com/demo/eval/arm.htm>

**Step2.** Install HT32 PACK by online PACK installer as the figure below. If your computer has no internet connection, please install the PACK file in the path below (double-click the “.pack” file).

“\\Tools\\Holtek.HT32\_DFP.n.n.n.pack”.



**Step3.** Unzip HT32 Firmware Library.

“\\Firmware\_Library\\HT32Fxxxxx\_FWLib\_vnnn\_nnn.zip”

**Step4.** Open the project file

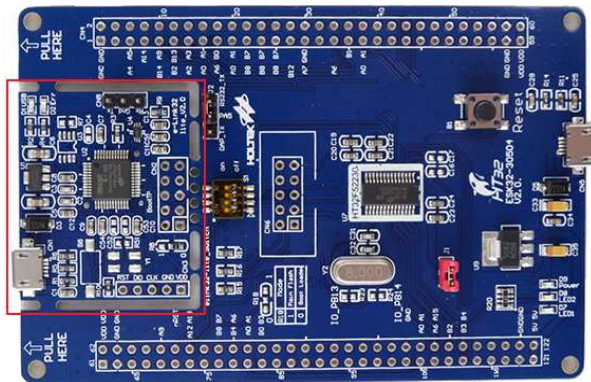
\\HT32Fxxxxx\_FWLib\_vnnn\_nnn\\project\_template\\IP\\Example\\MDK\_ARMv5\\Project\_XXXXX.uvprojx (xxxxx is chip name).

**Step5.** Build the project by press F7.

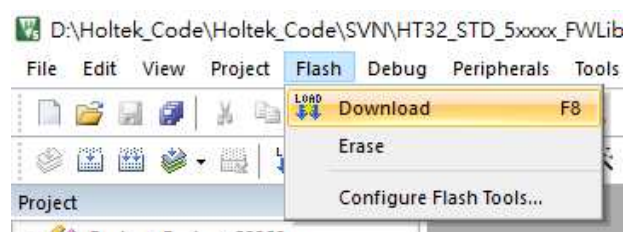
**Step6.** Install the HT32 Series Virtual COM Driver.

\\Tools\\HT32\_VCP\_Driver\_v110.exe

**Step7.** Connect the USB debug adapter of Starter Kit with PC by USB cable, confirm the USB debug adapter is working successfully by the “Windows Device Manager”. You can find a new “Holtek USB Virtual COM Port (COMn)” device. Please remember the COM Port number, COMn for Step10.

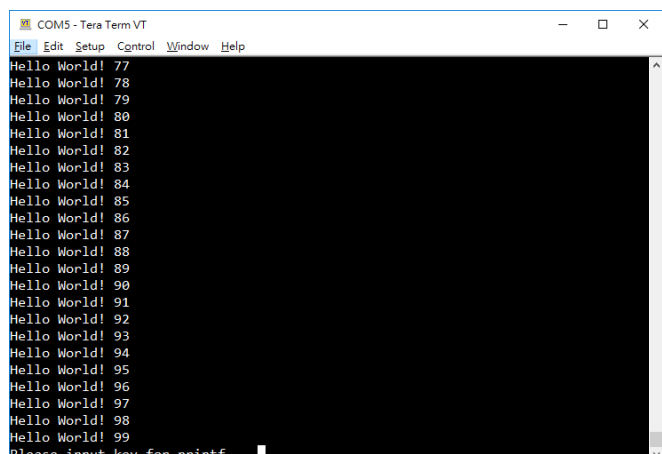
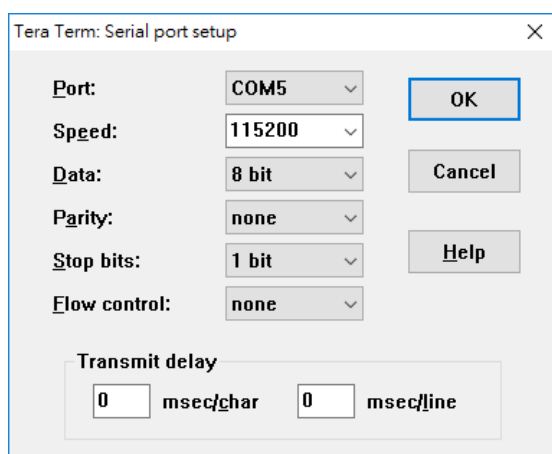


**Step8.** Program the chip by download icon or Keil's menu, Flash->Download.



**Step9.** Press the reset button on the Start Kit. Confirm the LED1 and LED2 is toggling.

**Step10.** You can use the terminal program such as “**Tera Team**” to see the printf message from the UART of target MCU. Please run the terminal program, open the COM port number which you find in the Step7, and then configure the parameter, “Baud Rate: **115200**, Data: **8 bit**, Parity: **none**, Stop bits: **1 bit**, Flow control: **none**”. Press the reset button again, after the LED blinking, you will see the Hello World message in the terminal program.



## 4. SourceryG++Lite Compiler with Keil MDK-ARM

If you want to use “SourceryG++Lite” compiler with Keil, please refer to the steps below. Notice that only the Firmware Library version below or later supports SourceryG++Lite project file.

HT32\_STD\_5xxxx\_FWLib\_v007\_nnnn.zip

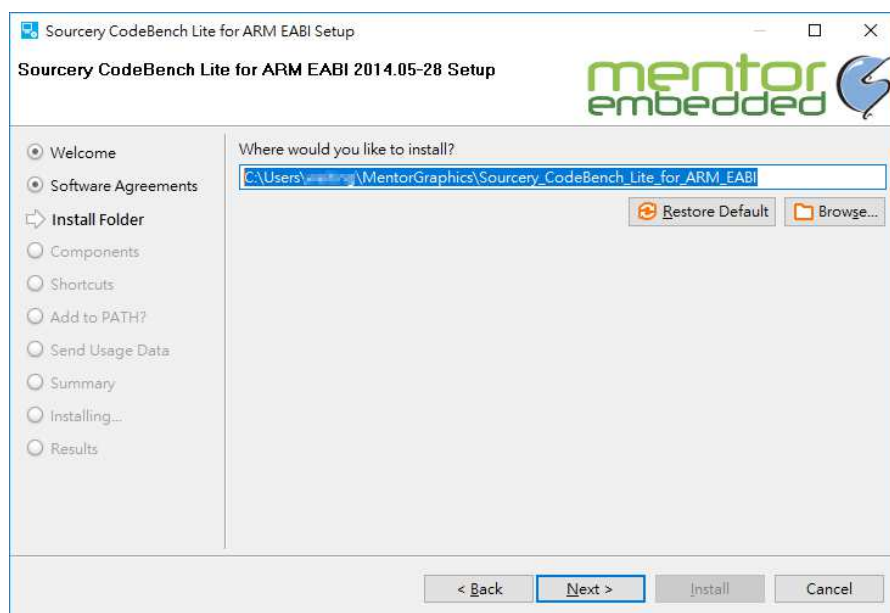
HT32\_STD\_1xxxx\_FWLib\_v002\_nnnn.zip

**Step1.** Download the SourceryG++Lite install EXE file from the link below.

<https://sourcery.mentor.com/GNUToolchain/package11445/public/arm-none-eabi/arm-2014.05-28-arm-none-eabi.exe>

**Step2.** Install SourceryG++Lite compiler. In the “Install Folder” page, copy and save the install path text (You need to configure the path into the Keil later). For example, “C:\Users\account\_name\MentorGraphics\Sourcery\_CodeBench\_Lite\_for\_ARM\_EABI”.

**Notice: We recommend rebooting your computer after installing a new compiler. You can reboot the compiler after Step4.**

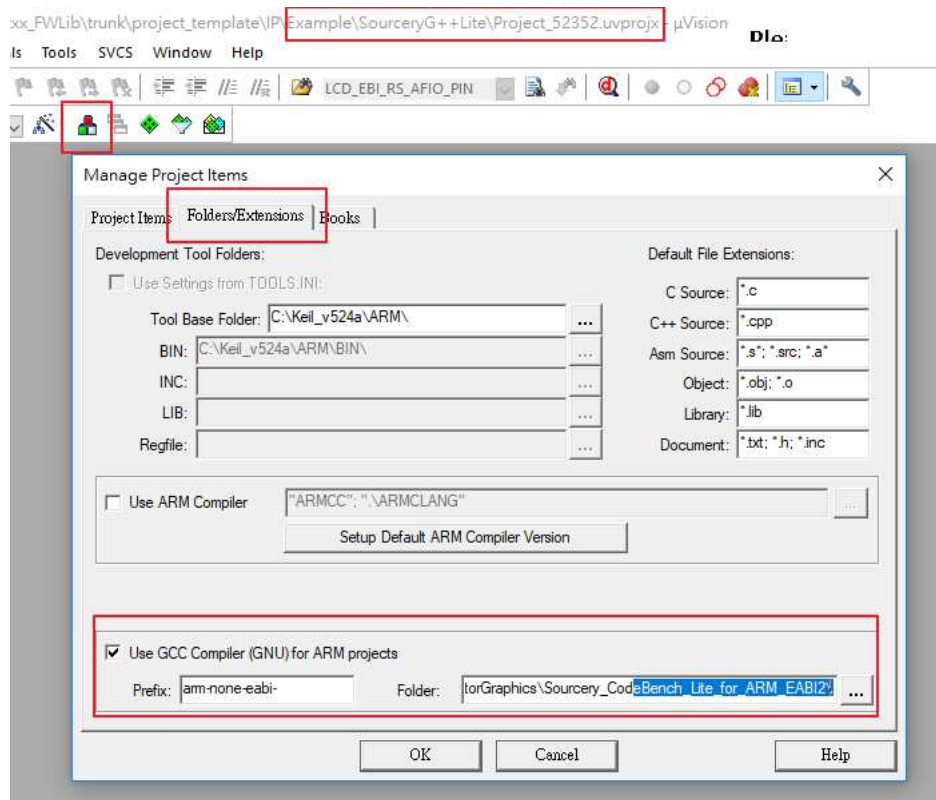


**Step3.** Open the project file

\\HT32Fxxxxx\_FWLib\_vnnn\_nnn\project\_template\IP\Example\SourceryG++Lite  
\Project\_xxxxx.uvprojx (xxxxx is chip name).

**Step4.** Press the “Manage Project Items” icon, click the “Folders/Extensions” tab, enable the checkbox, “Use GCC Compiler (GUN) for ARM projects”, and then

copy the install path of SourceryG++Lite into the “**Folder**” text box.

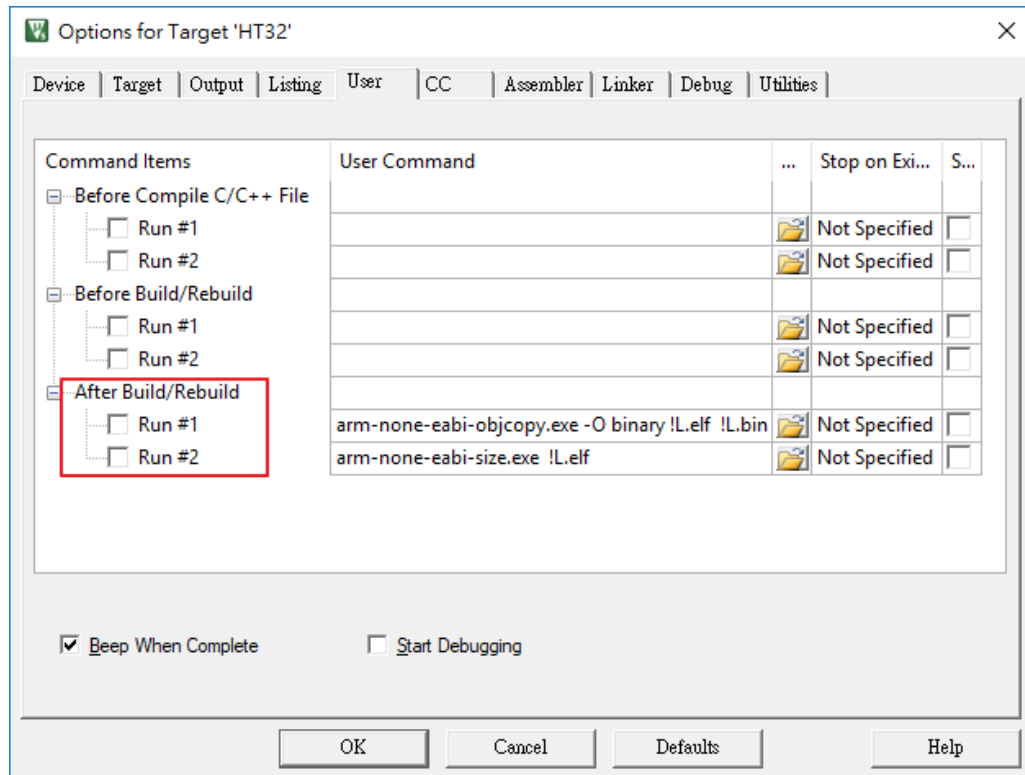


**Step5.** Build the project by press F7 and follow the Step 5 ~ 8 of “**Setup Keil MDK-ARM with HT32 Starter Kit**” section to confirm the compile and download is working.

**Step6.** If you encounter the following error message, please reboot the computer or run the Keil MDK-ARM as administrator privileges to let the “After Build” operation success.

```
compiling ht32f5xxxx_tm.c...
compiling ebi_lcd.c...
linking...
creating hex file...
After Build - User command #1: arm-none-eabi-objcopy.exe -O binary .\HT32\52352\Obj\HT32.elf .\HT32\52352\Obj\HT32.bin
*** Error: CreateProcess failed, Command: 'arm-none-eabi-objcopy.exe -O binary .\HT32\52352\Obj\HT32.elf .\HT32\52352\Obj\HT32.bin'
After Build - User command #2: arm-none-eabi-size.exe .\HT32\52352\Obj\HT32.elf
*** Error: CreateProcess failed, Command: 'arm-none-eabi-size.exe .\HT32\52352\Obj\HT32.elf'
".\HT32\52352\Obj\HT32.elf" - 2 Error(s), 0 Warning(s).
Target not created.
Build Time Elapsed: 00:00:06
```

If the problem cannot be fixed by the step above, you can also simply disable the “After Build/Rebuild” option as below. However, Keil will not output the binary format and code size information anymore.





## 5. GNU Arm Compiler with Keil MDK-ARM

If you want to use “GNU Arm” compiler with Keil, please refer to the steps below. Notice that only the Firmware Library version below or later supports GNU Arm project file.

HT32\_STD\_5xxxx\_FWLib\_v007\_nnnn.zip

HT32\_STD\_1xxxx\_FWLib\_v002\_nnnn.zip

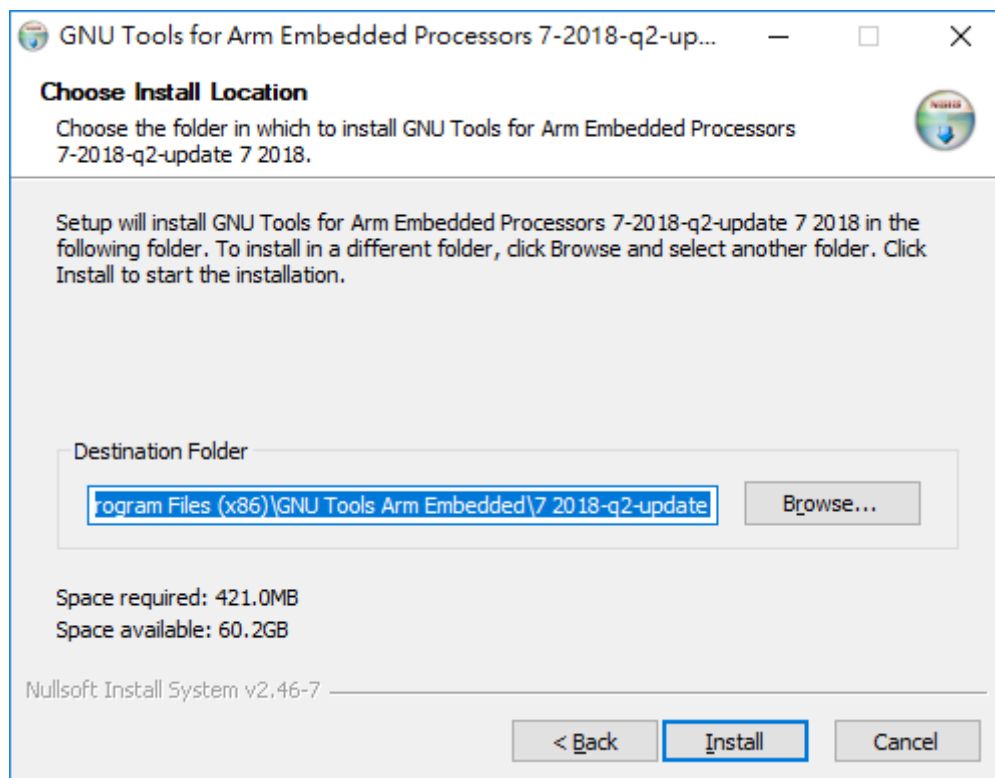
**Step1.** Download the GNU Arm install EXE file from the link below.

<https://developer.arm.com/open-source/gnu-toolchain/gnu-rm>

The filename is similar to

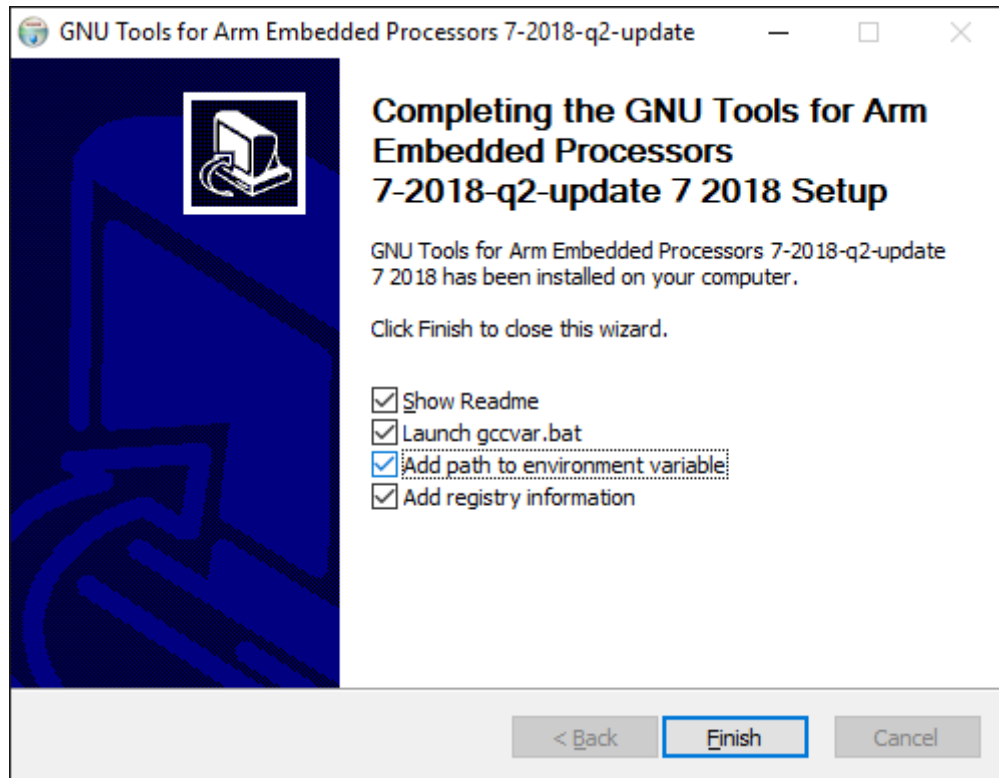
**“gcc-arm-none-eabi-7-2018-q2-update-win32-sha2.exe”.**

**Step2.** Install GNU Arm compiler. In the “Choose Install Location” page, copy and save the install path text (You need to configure the path into the Keil later). For example, “C:\Program Files (x86)\GNU Tools Arm Embedded\7 2018-q2-update”.



**Step3.** In the last page of the setup, tick the “Add path to environment variable” and press “Finish” button.

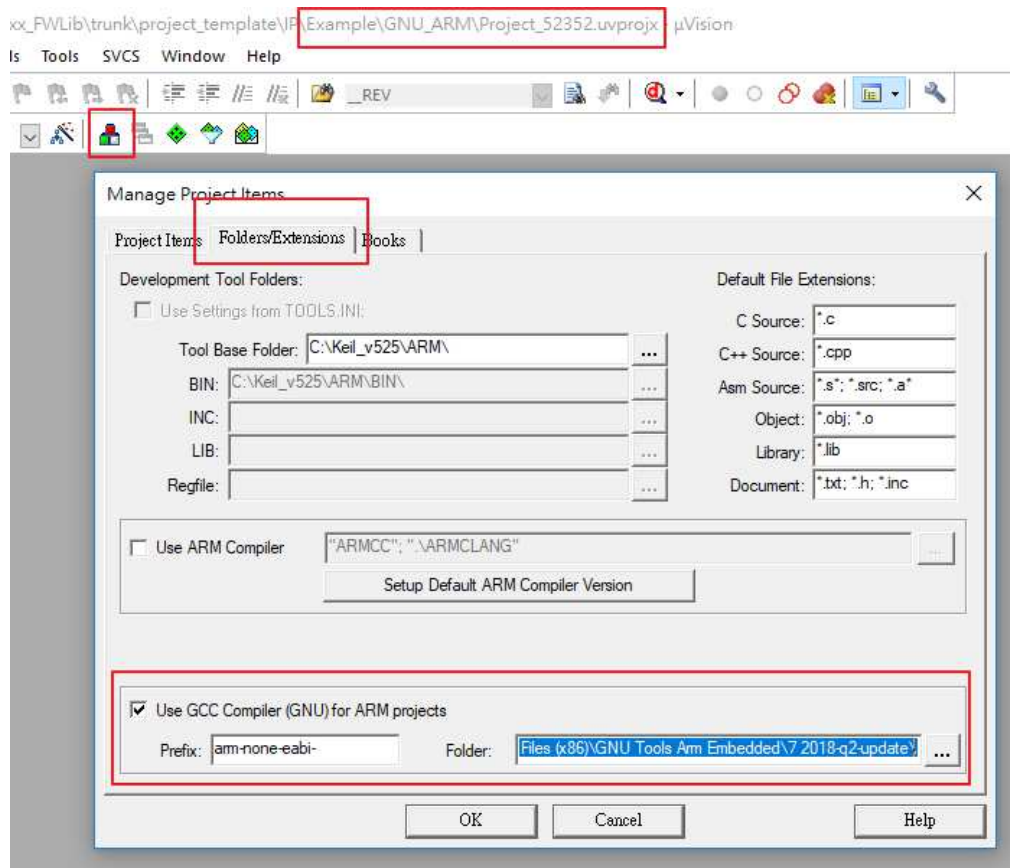
**Notice:** We recommend rebooting your computer after installing a new compiler. You can reboot the compiler after Step5.



**Step4.** Open the project file

\\HT32Fxxxxx\_FWLib\_vnnn\_nnn\project\_template\IP\Example\GNU\_ARM\Project\_xxxxx.uvprojx (xxxxx is chip name).

**Step5.** Press the “Manage Project Items” icon, click the “Folders/Extensions” tab, enable the checkbox, “Use GCC Compiler (GUN) for ARM projects”, and then copy the install path of GNU Arm into the “Folder” text box.

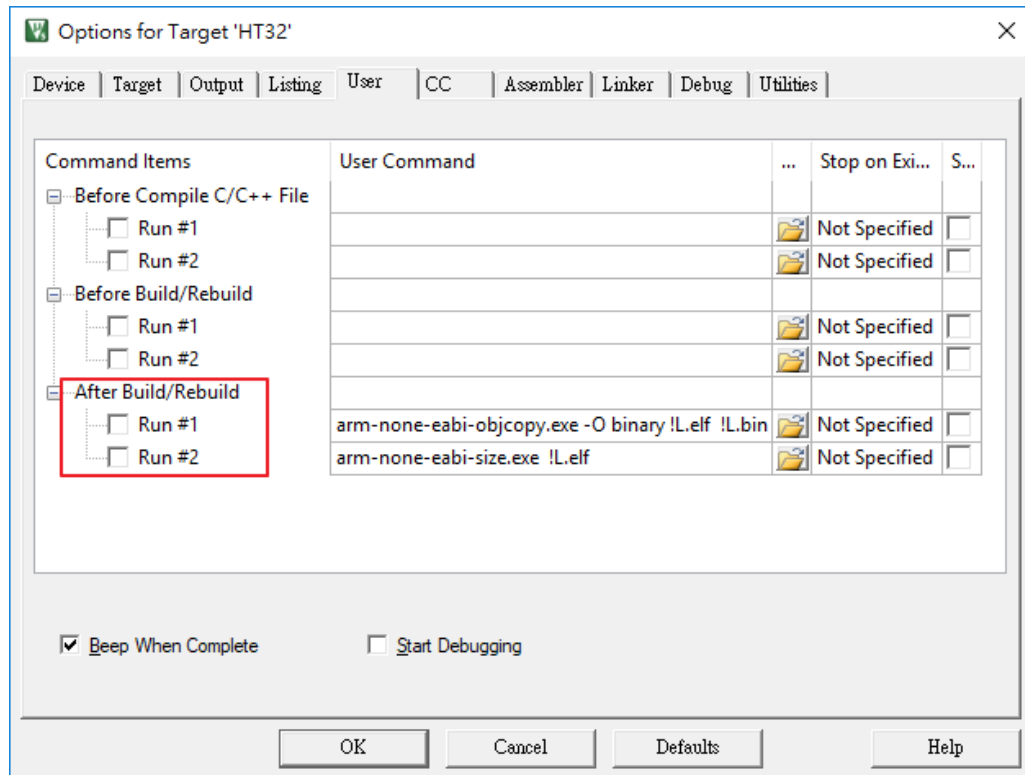


**Step6.** Build the project by press F7 and follow the Step 5 ~ 8 of “**Setup Keil MDK-ARM with HT32 Starter Kit**” section to confirm the compile and download is working.

**Step7.** If you encounter the following error message, please reboot the computer or run the Keil MDK-ARM as administrator privileges to let the “After Build” operation success.

```
|compiling ht32f5xxxx_tm.c...
|compiling ebi_lcd.c...
|linking...
|creating hex file...
After Build - User command #1: arm-none-eabi-objcopy.exe -O binary .\HT32\52352\Obj\HT32.elf .\HT32\52352\Obj\HT32.bin
*** Error: CreateProcess failed, Command: 'arm-none-eabi-objcopy.exe -O binary .\HT32\52352\Obj\HT32.elf .\HT32\52352\Obj\HT32.bin'
After Build - User command #2: arm-none-eabi-size.exe .\HT32\52352\Obj\HT32.elf
*** Error: CreateProcess failed, Command: 'arm-none-eabi-size.exe .\HT32\52352\Obj\HT32.elf'
".\HT32\52352\Obj\HT32.elf" - 2 Error(s), 0 Warning(s).
Target not created.
Build Time Elapsed: 00:00:06
```

If the problem cannot be fixed by the step above, you can also simply disable the “After Build/Rebuild” option as below. However, Keil will not output the binary format and code size information anymore.



## 6. GNU Arm Compiler with GNU Make

If you want to use “GNU Arm” compiler with GNU Make, please refer to the steps below. GNU Make is a command line tool which controls the generation of the executables (means firmware image here). Please get the latest version of HT32 Firmware Library and unzip them first.

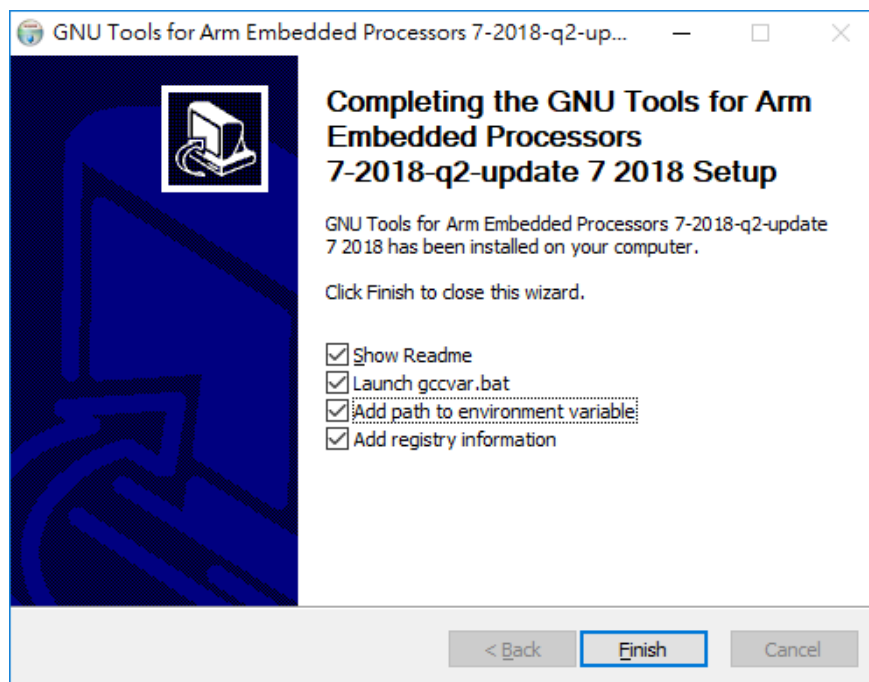
**Step1.** Download the GNU Arm install EXE file from the link below.

<https://developer.arm.com/open-source/gnu-toolchain/gnu-rm>

The filename is similar to

**“gcc-arm-none-eabi-7-2018-q2-update-win32-sha2.exe”.**

**Step2.** Install GNU Arm compiler by executing the file you download from the Step1. In the latest page “Completing the GNU Tools....”, tick the “Add path to environment variable” checkbox and press the “Finish” button. It adds the path of GNU Arm compiler into the “Path” Environment Variable. The path is similar to “C:\Program Files (x86)\GNU Tools Arm Embedded\7 2018-q2-update\bin”. You can also add the path manually.

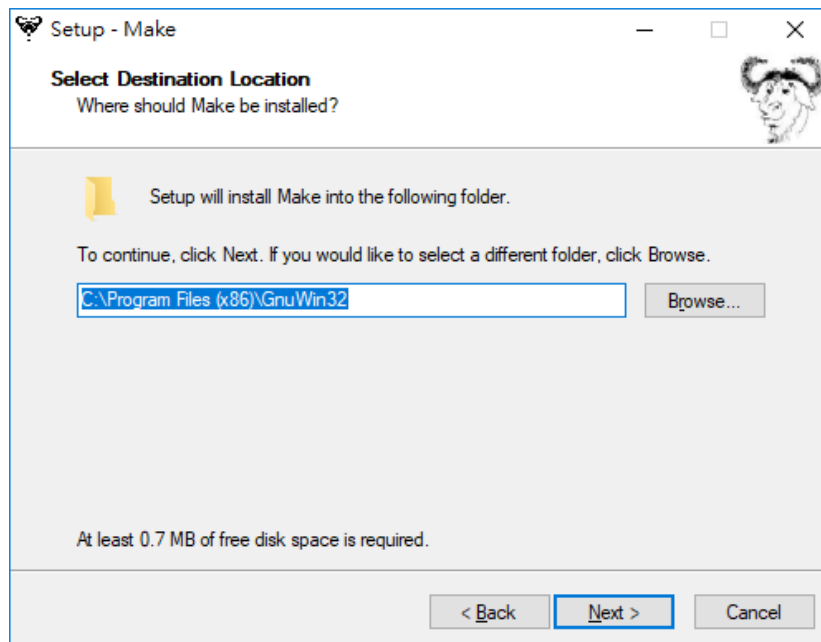


**Step3.** Download the GNU Make install EXE file from the link below.

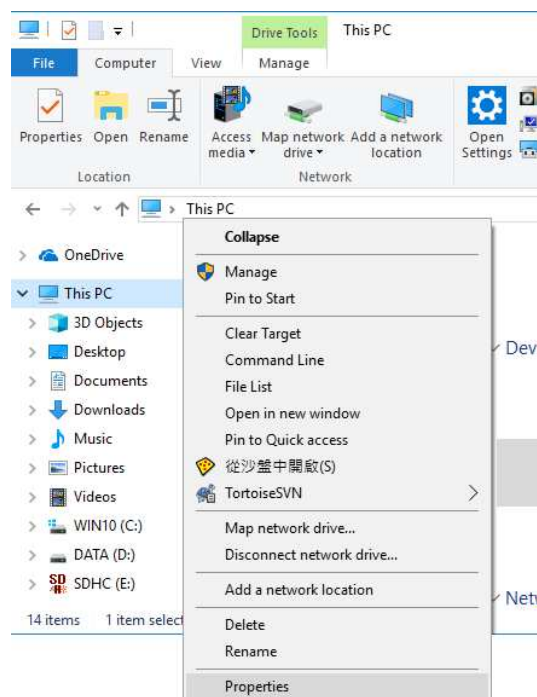
<http://gnuwin32.sourceforge.net/packages/make.htm>

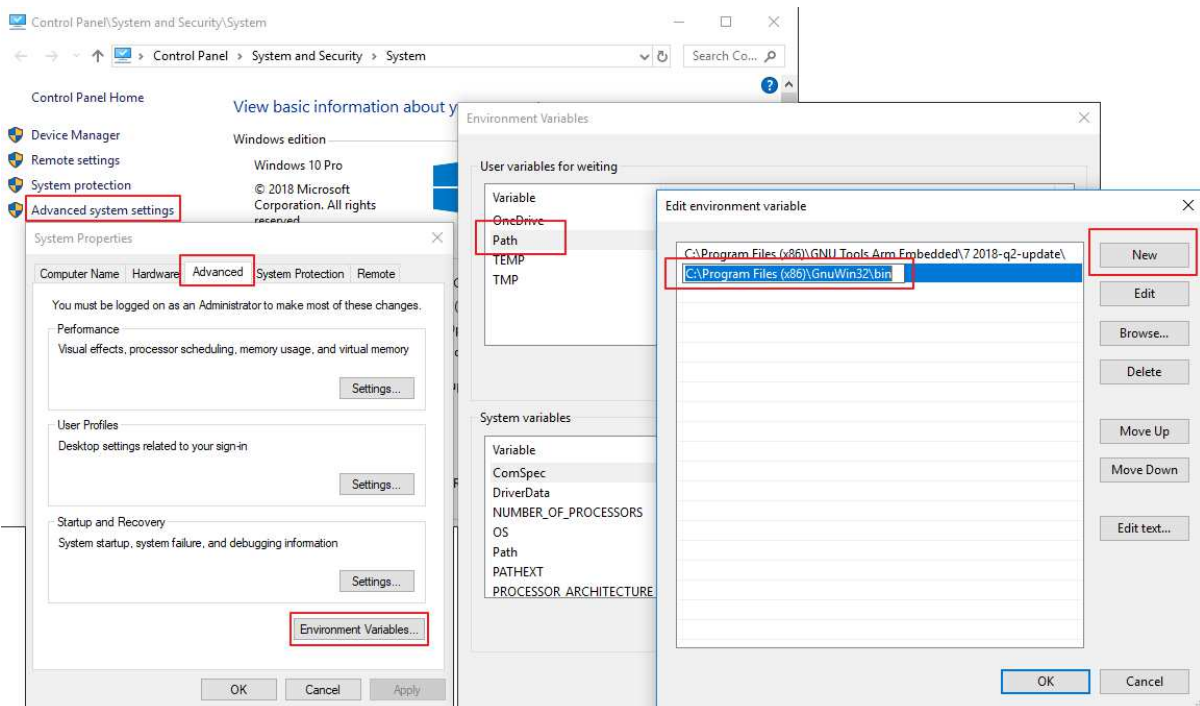
The filename is similar to **“make-3.81.exe”.**

**Step4.** Install GNU Make. In the “Select Destination Location” page, copy and save the install path text. For example, “C:\Program Files (x86)\GnuWin32”.



**Step5.** Please add an extra string “\bin” with the path you copy from the Step4, and configure the “Path” Environment Variable to add the path of the “make” tool. The whole path is similar to “C:\Program Files (x86)\GnuWin32\bin”. You can refer to the images below which shows how to edit the “Path” Environment Variable.





**Step6.** Use the command line and change the working directory into the "GNU\_ARM" folder of the project. And then type "make xxxxx" or "make -f xxxxx.mk" to build the code (where xxxxx is the chip name). After all the build operation finished, you will get the "BUILD SUCCESSFUL" message. For example:

```

Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\weiting>d:

D:\>cd HT32_STD_lxxxx_FWLib_v002_1406

D:\HT32_STD_lxxxx_FWLib_v002_1406>cd project_template\IP\Example\GNU_ARM

D:\HT32_STD_lxxxx_FWLib_v002_1406\project_template\IP\Example\GNU_ARM>make 12366
make[1]: Entering directory `D:/HT32_STD_lxxxx_FWLib_v002_1406/project_template/IP/Example/GNU_ARM'
arm-none-eabi-as -mcpu=cortex-m3 -mthumb -mthumb-interwork --defsym USE_HT32_CHIP=2 startup_ht32flxxxx_gcc_01.s -o startup_ht32flxxxx_gcc_01.o
arm-none-eabi-gcc -mcpu=cortex-m3 -mthumb -mthumb-interwork -gdwarf-2 -MD -Wall -Os -mapcs-frame -ffunction-sections -fdata-sections -DUSE_HT32_DRIVER -DUSE_HT32F12366_SK -DUSE_HT32F12365_66 -DUSE_MEM_HT32F12366 -c -I./.../library/CMSIS/Include -I./.../library/Device/Holtek/HT32F1xxxx/Include -I./.../library/HT32F1xxxx_Driver/inc -I./.../library/HT32_USBD_Library/inc -I./.../utilities ../main.c ../ht32flxxxx_01_it.c ../system_ht
c ./.../library/HT32F1xxxx_Driver/src/ht32f2xxxx_csif.c ./.../library/HT32F1xxxx_Driver/src/ht32_retargct.c
./.../library/HT32F1xxxx_Driver/src/ht32_serial.c ./.../library/HT32F1xxxx_Driver/src/printf.c ./.../library/HT32F1xxxx_Driver/src/syscalls.c ./.../utilities/ht32_board.c ./.../utilities/common/i2c_eeprom.c ./.../utilities/common/spi_flash.c ./.../utilities/common/ebi_lcd.c
arm-none-eabi-gcc -Tlinker.ld -mcpu=cortex-m3 -mthumb -mthumb-interwork -Wl,-Map=HT32.map -Wl,--gc-sections --specs=nano.specs *.o -o HT32.elf
arm-none-eabi-objcopy -O ihex HT32.elf HT32.hex
arm-none-eabi-objcopy -O binary HT32.elf HT32.bin
arm-none-eabi-size.exe HT32M\12366\Obj\HT32.elf
   text    data    bss     dec    hex filename
   3088      8      32    3128    c38 HT32M\12366\Obj\HT32.elf
BUILD SUCCESSFUL
make[1]: Leaving directory `D:/HT32_STD_lxxxx_FWLib_v002_1406/project_template/IP/Example/GNU_ARM'

D:\HT32_STD_lxxxx_FWLib_v002_1406\project_template\IP\Example\GNU_ARM>
  
```

**Step7.** After Step6, you can find the Hex and Binary file in the path below. Now you can use other programming tools such as e-Writer32, HT32 ICP Tool with e-Link32 Pro, or HT32 Flash Programmer with ISP bootloader to program the firmware into the Starter Kit.

"...\GNU\_ARM\HT32M\xxxxx\Obj\HT32.bin"

"...\GNU\_ARM\HT32M\xxxxx\Obj\HT32.hex"

Note: The "xxxxx" represents the chip name.