GigaDevice Semiconductor Inc.

Arm® Cortex®-M3/4/23/33 32-bit MCU

应用笔记

AN033

Table of Contents

List of Figures

List of Tables

1. Introduction

We usually use the IDE under windows to develop MCU project, This article describes how to manage a RTOS projects in a Linux environment with multiple Makefiles, develop an LED flashing feature in RTOS task. This scheme can specify modules or files to compile.

2. Development environment construction

The development environment prepare:

- Hardware development board: GD32F303-Test-V1.1
- Cpmpile environment: ubuntu16.04
- Tool chain: gcc-arm-none-eabi, gcc-arm-none-objcopy
- Programing tools: SEGGR J-FlashVV6.50b

2.1. Install Ubuntu VM

VMware workstation download address:https://www.vmware.com/cn/products/workstation-pro/workstation-pro-evaluation.html.

Double-click the installation package and follow the installation wizard, shown in ***Figure 2-1. Ubuntu vm installation wizard 1*** click"next", Select the default settings.

Figure 2-1. Ubuntu vm installation wizard 1

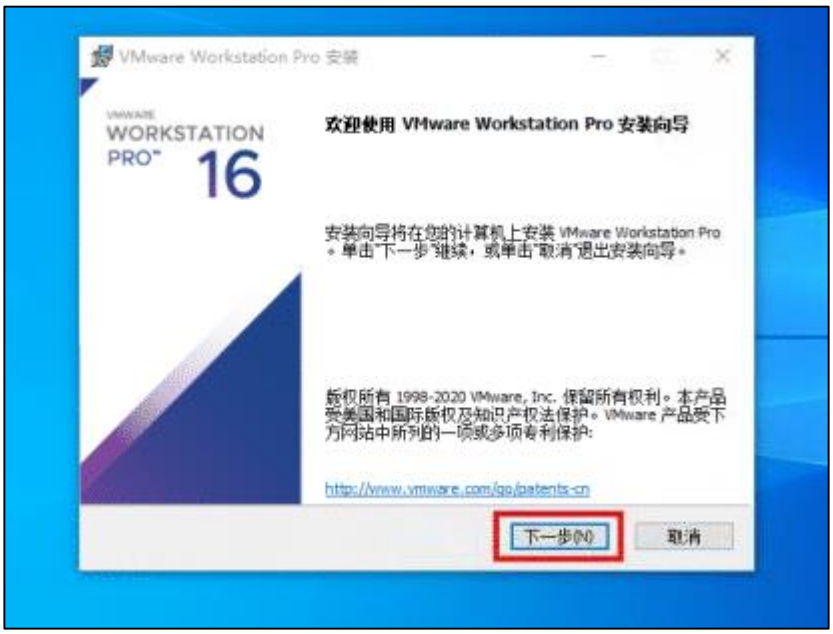


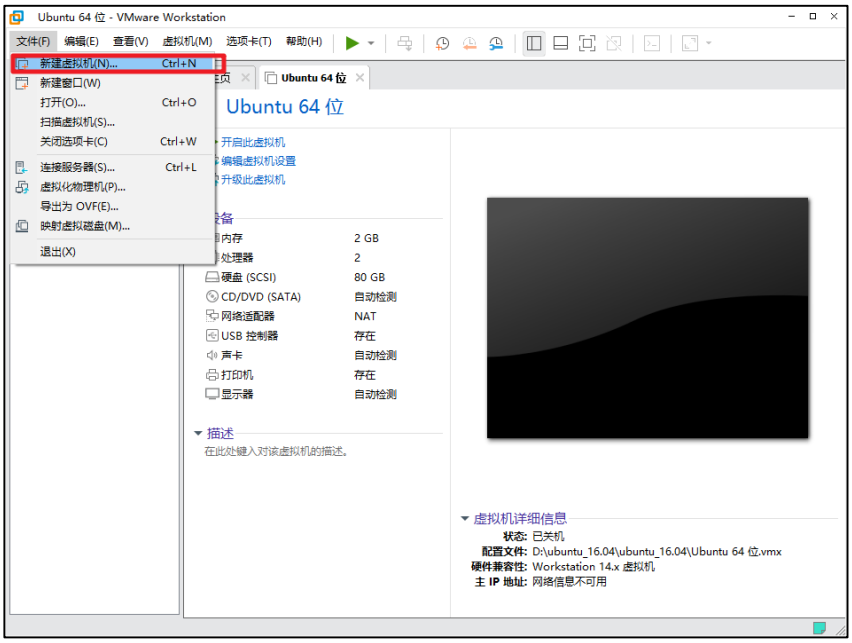
Figure 2-2. Ubuntu vm installation wizard 2

Click ”finish”, finish install.

Figure 2-3. Ubuntu vm installation wizard 3

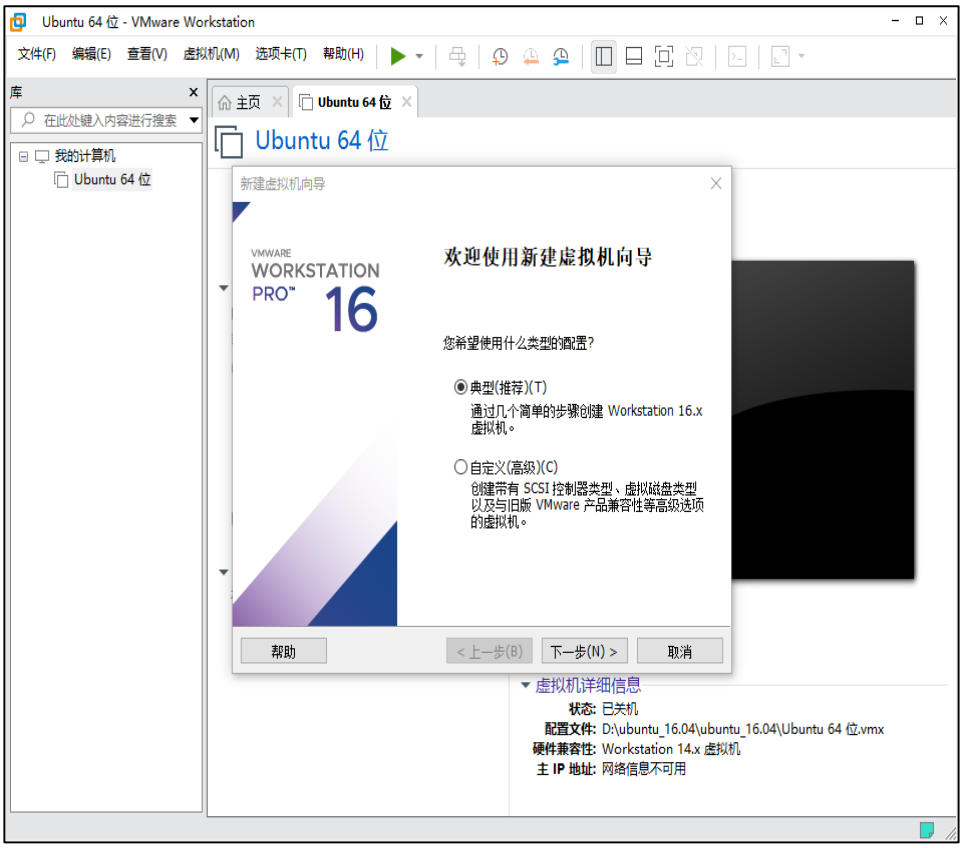
Download ubuntu iso file and install.download address:http://mirrors.aliyun.com/ubuntu-releases/16.04/ After dowmload Ubuntu iso,open VMware, click ”file --> create new virtual machine”.

Figure 2-4. Ubuntu vm installation wizard 4



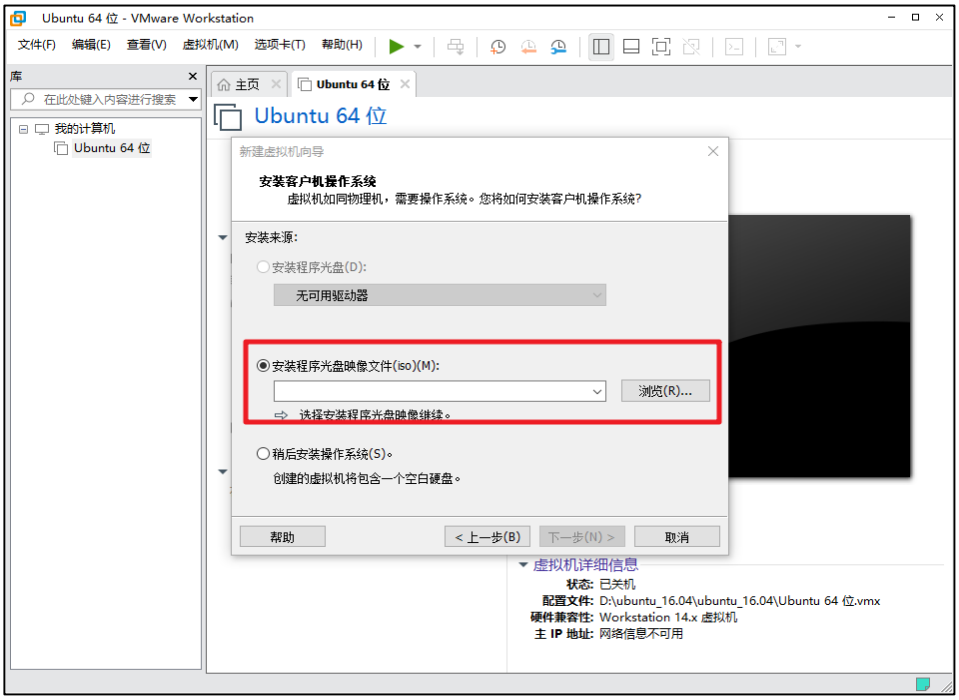
Use default settings, click”next”.

Figure 2-5. Ubuntu vm installation wizard 5



Select the Ubuntu iso file downloaded earlier,use default settings and click “next”.

Figure 2-6. Ubuntu vm installation wizard 6



Click start this virtual machine, the first startup takes a long time, select the default settings until startup is complete.

Figure 2-7. Ubuntu vm installation wizard 7

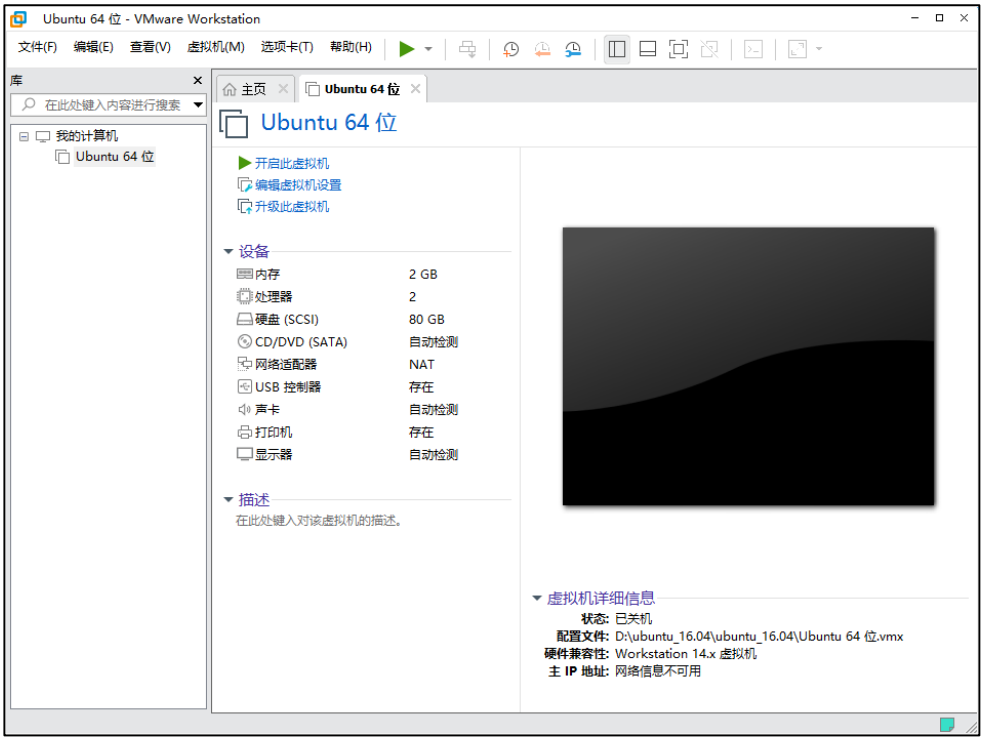
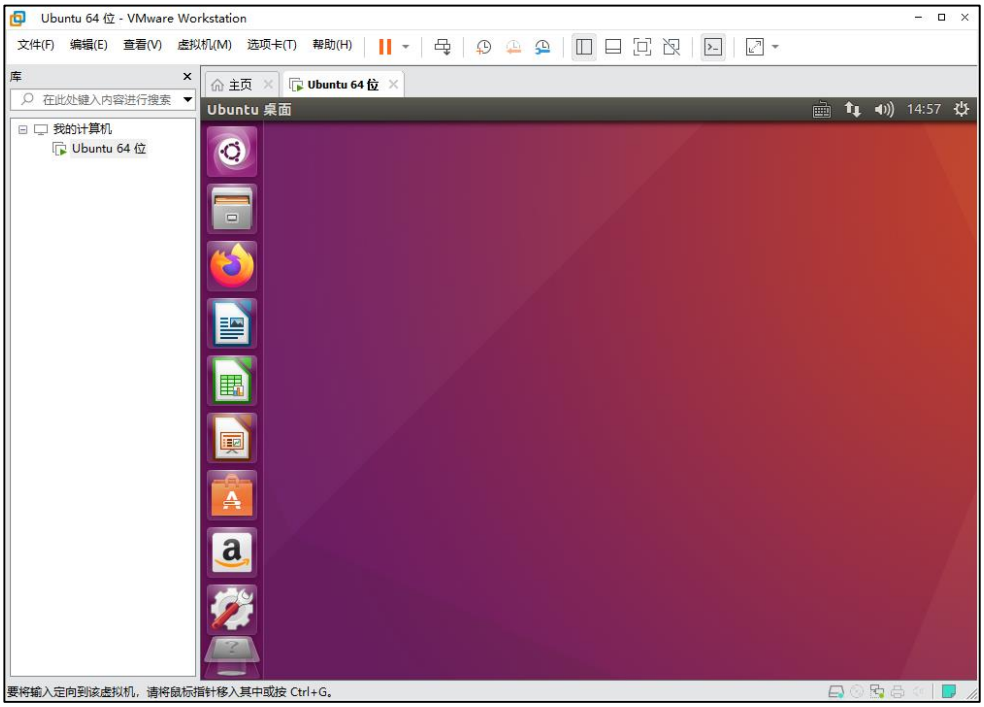


Figure 2-8. Ubuntu vm start done 8 show the page after the Ubuntu VM is started.

Figure 2-8. Ubuntu vm start done 8

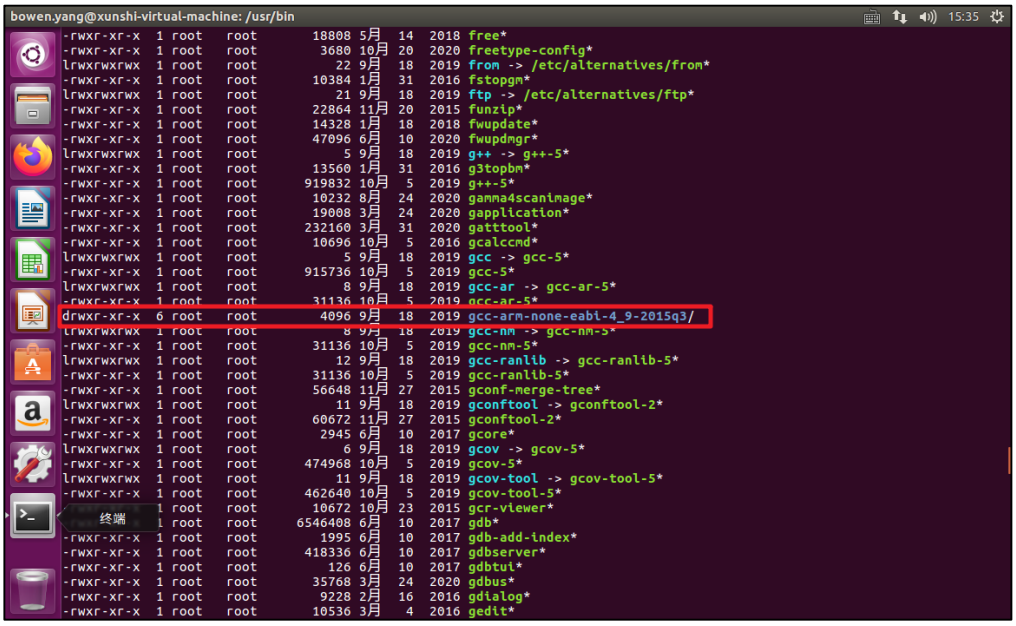


2.2. Install toolchain

Use the compiled toolchain,download address:https://launchpad.net/gcc-arm-embedded/+download.

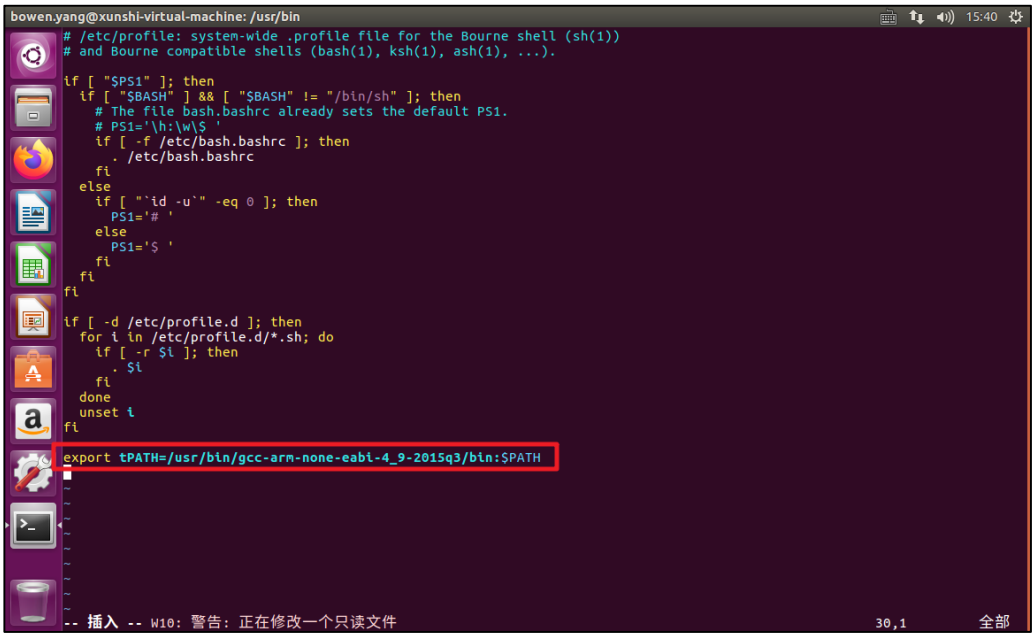
Decompress the downloaded toolchain to the /usr/bin directory of Ubuntu, get a directory gcc-arm-none-eabi-4_9-2015q3/. shown in ***Figure 2-9. directory of GCC toolchain***.

Figure 2-9. directory of GCC toolchain



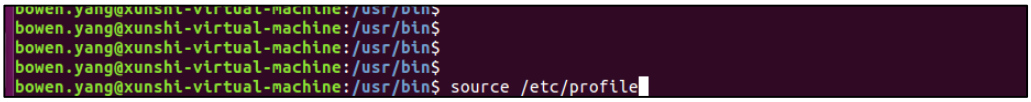
Configure the linux system environment variables. assign GCC toolchain directory in the last of /etc/profile.

Figure 2-10. configuration environment variable



Then input command ”source /etc/profile” make the environment variable effective. The Ubuntu VM dose not need to be restarted.

Figure 2-11. make the environment variable effective



After finishing, input command “arm”with the tab key,is successful if the tool chain list is displaye. shown in ***Figure 2-12. toolchain list*** here is arm-none-eabi-gcc and arm-none-eabi-objcopy we needed.

Figure 2-12. toolchain list

3. Create project

3.1. Create project directory

Place the code that needs to be compiled below the path D:\share\RTX\ubuntu, shown in *Figure 3-1. project code directory*.

Figure 3-1. project code directory

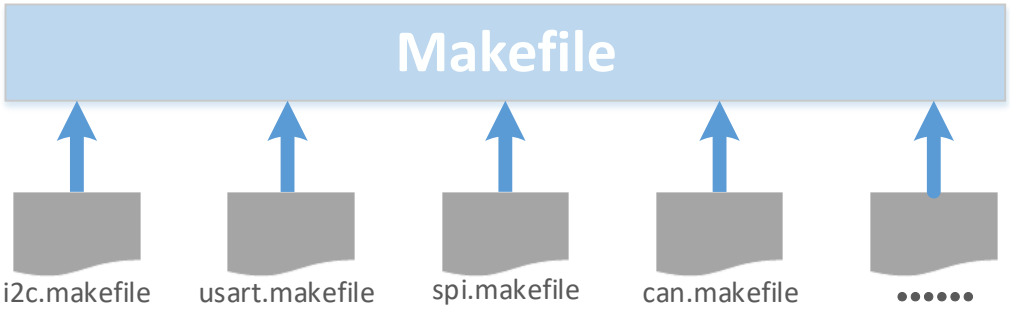
The directory structure is shown below, every directory needed a makefile.

```
├──APP
│     ├──include
│     └──source
├──CMSIS
│     ├──GCC
│     ├──include
│     └──source
├──Core
│     ├──include
│     └──source
├──Device
│     ├──include
│     └──source
└──SourceGroup
      ├──include
      └──source
```

3.2. Makefile writing

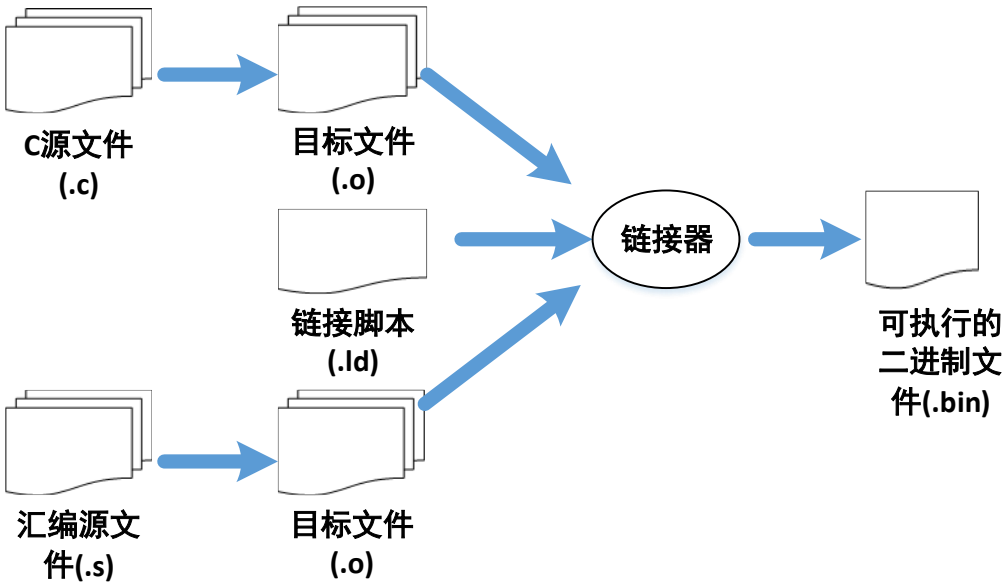
Use top-level makefile to manager makefile in subdirectories, shown in *Figure 3-2. makefile graphical*.

Figure 3-2. makefile graphical



The compilation processs shown in *Figure 3-3. compilation flow chart*.

Figure 3-3. compilation flow chart



Top-level makefile edit an below:

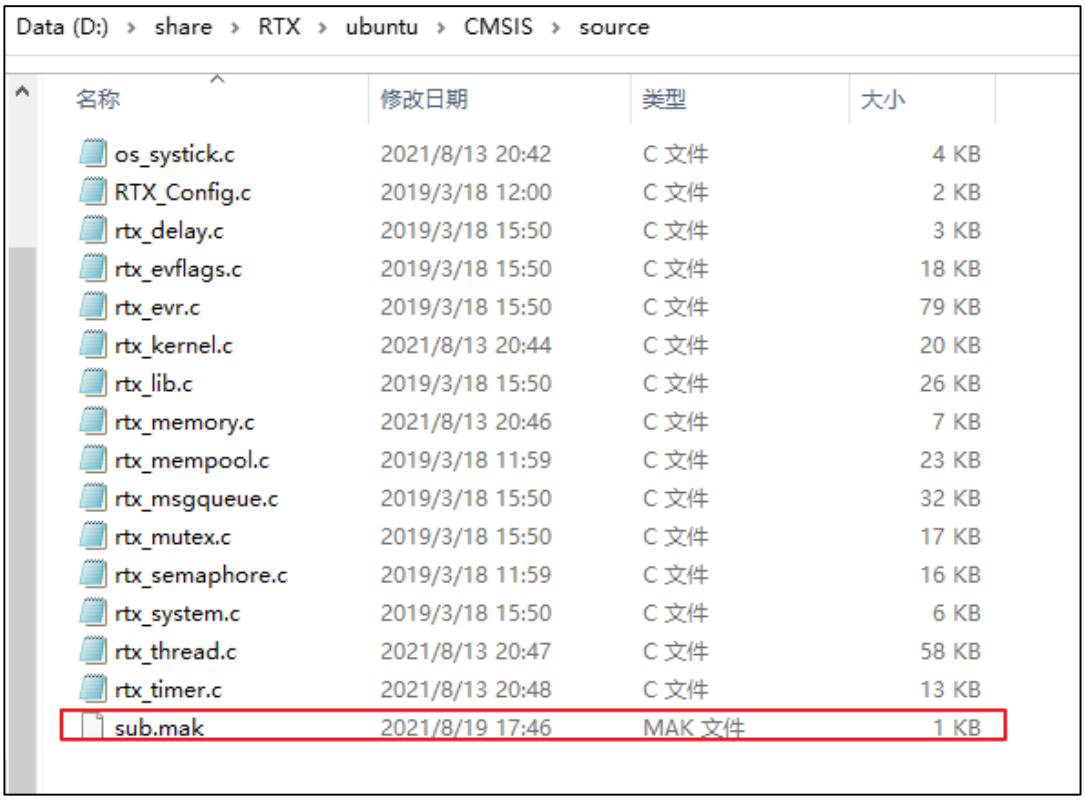
Table 3-1. top-level makefile edit

```
CROSS_COMPILE=arm-none-eabi-
CC                = $(CROSS_COMPILE)gcc
OBJCOPY           = $(CROSS_COMPILE)objcopy

TOP=$(shell pwd)
INC_FLAGS = -I$(TOP)/Device/include \
            -I$(TOP)/Core/include \
            -I$(TOP)/APP/include \
            -I$(TOP)/Stdlib/include \
            -I$(TOP)/CMSIS/include

CC_FLAGS = -W -Wall -g -mcpu=cortex-m4 -mthumb -D GD32F30X_HD -D
USE_STDPERIPH_DRIVER $(INC_FLAGS) -O0 -std=gnu11
```

```
CC_ASM_FLAGS = -mthumb -mcpu=cortex-m4 -g -Wa,--warn

CC_LD_FLAGS += -mthumb -mcpu=cortex-m4
CC_LD_FLAGS += -Wl,--start-group -lc -lm -Wl,--end-group -specs=nosys.specs -static -Wl,-cref,-u,Reset_Handler-Wl,-Map=RTX_Project.map-Wl,--gc-sections-Wl,--defsym=malloc_getpagesize_P=0x80

LD_PATH = -TDevice/source/gd32f30x_flash.ld
#Indicates whether the module participates in compilation
SUPPORT_IIC = yes
SUPPORT_SPI = yes
SUPPORT_CAN = yes
SUPPORT_KEY = no
#Specify the source file to compile
include APP/source/sub.mak
include CMSIS/source/sub.mak
include Device/source/sub.mak
include SourceGroup/source/sub.mak

TARGET = RTX_Project

.PHONY: clean all
#replace with.o
C_OBJ = $(C_SRC:%.c=%.o)
ASM_OBJ = $(ASM_SRC:%.s=%.o)

all:$(C_OBJ) $(ASM_OBJ)
	$(CC) $(C_OBJ) $(ASM_OBJ) $(LD_PATH) -o $(TARGET).elf $(CC_LD_FLAGS)
	$(OBJCOPY) $(TARGET).elf  $(TARGET).bin -Obinary

%.o:%.c
	$(CC) -c $(CC_FLAGS) -o $@ $<
%.o:%.s
	$(CC) -c $(CC_ASM_FLAGS) -o $@ $<

clean:
	rm -rf *.o $(C_OBJ) $(ASM_OBJ) $(TARGET) *.bin *.map *.elf
```

Create sub.mak file in each source directory of APP, CMSIS, Core, Device, SourceGroup directory, all sub.mak files are written an the same way, just introduce the sub.mak of directory CMSIS/source as below:

Figure 3-4. sub.mak folder



Specifies the file in this directory to compile in sub.mak. shown as below:

Table 3-2. Subdirectory sub.mak

```
CMSIS_PATH = CMSIS/source
C_SRC += $(CMSIS_PATH)/os_systick.c \
                $(CMSIS_PATH)/RTX_Config.c \
                $(CMSIS_PATH)/rtx_delay.c \
                $(CMSIS_PATH)/rtx_evflags.c\
                $(CMSIS_PATH)/rtx_evr.c \
                $(CMSIS_PATH)/rtx_kernel.c \
                $(CMSIS_PATH)/rtx_lib.c \
                $(CMSIS_PATH)/rtx_memory.c \
                $(CMSIS_PATH)/rtx_mempool.c \
                $(CMSIS_PATH)/rtx_msgqueue.c \
                $(CMSIS_PATH)/rtx_mutex.c \
                $(CMSIS_PATH)/rtx_semaphore.c \
                $(CMSIS_PATH)/rtx_system.c \
                $(CMSIS_PATH)/rtx_thread.c \
                $(CMSIS_PATH)/rtx_timer.c

ifeq ($(SUPPORT_KEY), yes)
C_SRC += $(AUDIO_PATH)/rtx_keymanager.c
endif
```

```
ASM_SRC += $(CMSIS_PATH)/../GCC/irq_cm3.s
```

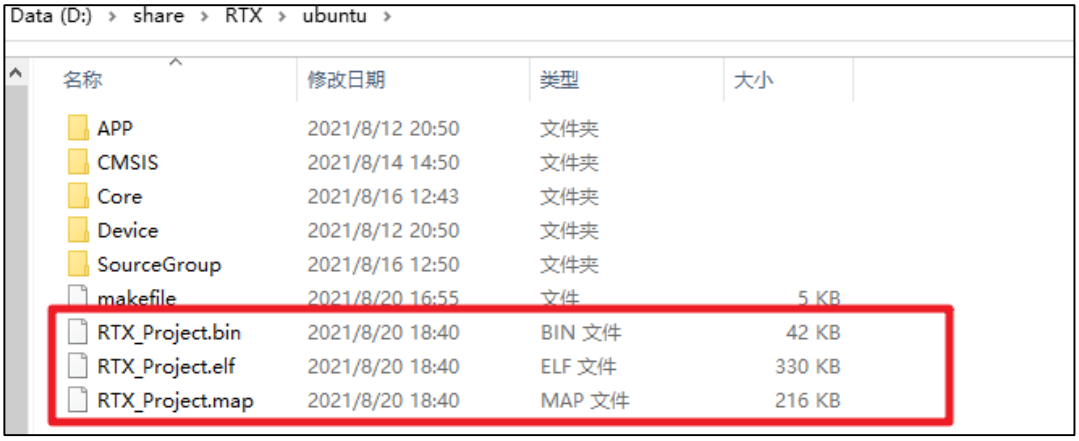
3.3. Compile and test

input command “ make” in the directory where the top-level makefile are.

Figure 3-5. make result

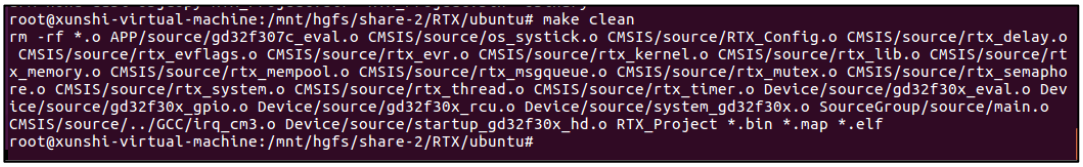
The top-level makefile folder will generate .bin file, .elf file and .map file.

Figure 3-6. top-level makefile folder



Input command “make clean” in top-level makefile folder, the .bin file, .elf file and .map file is deleted.

Figure 3-7. make clean result



Finally, we can use SEGGR J-Flash download the firmware to mcu for test. The LED is working.

4. Revision history

Table 4-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Aug.26, 2021

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.