Firmware Compiling User Guide

Calterah Semiconductor

Date	Version	Description	Author
2020.06.16	V0.1.0	First version	Xudong Ran
2020.06.19	V0.9.0	Reorganize	Yingzhe Zhou

Table of Contents

CHAPTER

ONE

INTRODUCTION

The purpose of this document is to illustrate how to compile Calterah radar Soc firmware configured with two-stage boot and three-stage boot architecture. This document applies to Alps and Rhine series only.

Calterah radar Soc firmware implement two-stage boot and three-stage boot architecture based on different use purpose. In software release package, by default, the firmware is configured to two-stage boot architecture.

There are 2 types of two-stage boot:

1. Normal mode:
   ROM code load firmware image from external flash and copy to RAM and then executes.
2. XIP Mode:
   ROM code enable XIP function and firmware can be executes in XIP mode.

Compared to two-stage boot architecture, in order to implement firmware OTA function, one more boot stage need to be added that the boot sequence shall be changed to three stages (ROM->Boot->Firmware) instead of two stages (ROM->Firmware). Once firmware received OTA command, the OTA request flag will be set accordingly and radar Soc chip will reboot automatically and enter boot stage for OTA purpose.

There are 3 types of three-stage boot:

1. Normal Mode:
   (1) ROM code load boot image from external flash and copy to ARC ICCM area.
   (2) During boot stage, firmware image will be loaded from external flash and copy to RAM and then executes.
2. XIP Mode:
   (1) ROM code load boot image from external flash and copy to ARC ICCM area.
   (2) During boot stage, XIP function is enabled and firmware will be executed under XIP mode.
3. Boot Split Mode:
   (1) Boot is split into Boot0(tiny-boot) image and Boot1 image.
   (2) ROM code load Boot0(tiny-boot) image from external flash and copy to ARC ICCM area.
   (3) During Boot0(tiny-boot) stage:
      a) If OTA request flag has NOT been set before, XIP function will be enabled and

firmware will be executed under XIP mode.

b) If OTA request flag has been set before, Boot0(tiny-boot) will load Boot1 from external flash and copy to RAM, OTA feature will be implemented during Boot1 stage.

(4) Boot split mode mainly used to accelerate the system boot up time.

CHAPTER

TWO

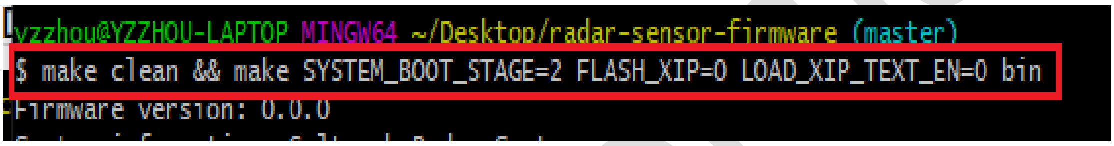
TWO-STAGE BOOT COMILING

There are 2 types of two-stage boot: Normal mode and XIP mode.

1. Normal mode:

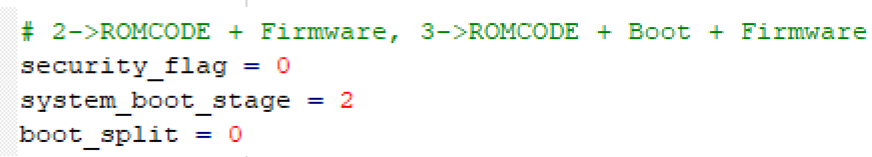
(1) Send below "**make**" command under "**./radar-sensor-firmware**" directory.

make clean && make SYSTEM_BOOT_STAGE=2 FLASH_XIP=0 LOAD_XIP_TEXT_EN=0 bin



```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ make clean && make SYSTEM_BOOT_STAGE=2 FLASH_XIP=0 LOAD_XIP_TEXT_EN=0 bin
Firmware version: 0.0.0
```

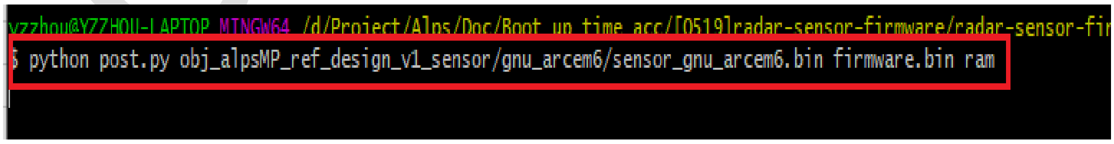
(2) Open "**post.py**" script with Notepad under "**./radar-sensor-firmware**" directory and change the value of "**system_boot_stage"** parameter to 2.



```
# 2->ROMCODE + Firmware, 3->ROMCODE + Boot + Firmware
security_flag = 0
system_boot_stage = 2
boot_split = 0
```

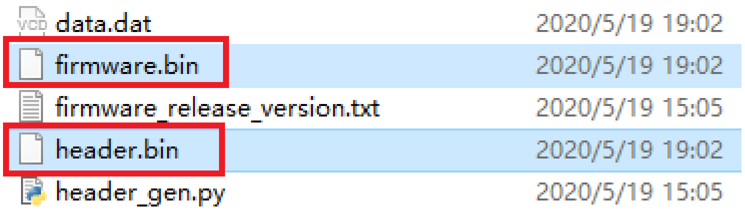
(3) Execute below "**post.py**" script under "**./radar-sensor-firmware**" directory to generate "**header.bin**" and "**firmware.bin**".

python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin ram



```
yzzhou@YZZHOU-LAPTOP MINGW64 /d/Project/Alps/Doc/Boot_up_time_acc/[0519]radar-sensor-firmware/radar-sensor-fir
$ python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin ram
```

(4) After all the above steps, you can find two binary files, "**header.bin**" and "**firmware.bin**" under "**./radar-sensor-firmware**" directory.



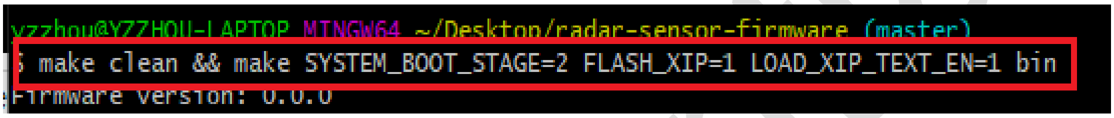
data.dat	2020/5/19 19:02
firmware.bin	2020/5/19 19:02
firmware_release_version.txt	2020/5/19 15:05
header.bin	2020/5/19 19:02
header_gen.py	2020/5/19 15:05

(5) For how to program "**header.bin**" and "**firmware.bin**" into external flash, please refer to the "**Flash Downloader User Guide**" document in software release package under "**Document**" directory.

2. XIP Mode:

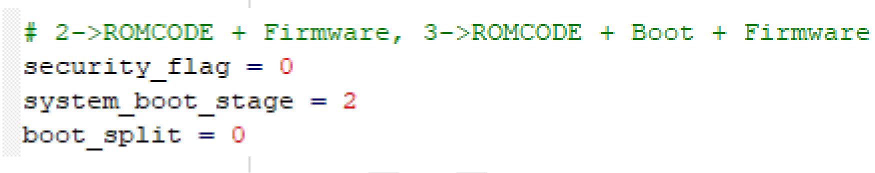
(1) Send below "**make**" command under "**./radar-sensor-firmware**" directory.

make clean && make SYSTEM_BOOT_STAGE=2 FLASH_XIP=1 LOAD_XIP_TEXT_EN=1 bin



```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ make clean && make SYSTEM_BOOT_STAGE=2 FLASH_XIP=1 LOAD_XIP_TEXT_EN=1 bin
Firmware version: 0.0.0
```

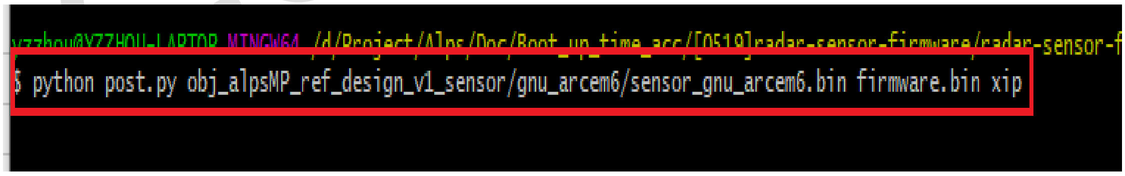
(2) Open "**post.py**" script with Notepad under "**./radar-sensor-firmware**" directory and change the value of "**system_boot_stage"** parameter to 2.



```
# 2->ROMCODE + Firmware, 3->ROMCODE + Boot + Firmware
security_flag = 0
system_boot_stage = 2
boot_split = 0
```

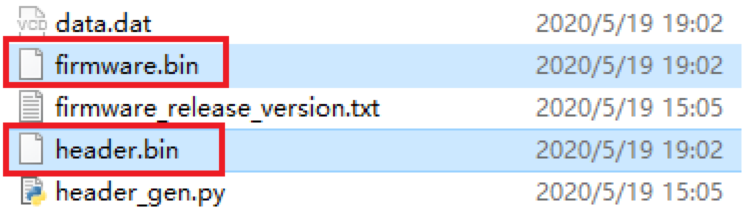
(3) Execute below "**post.py**" script under "**./radar-sensor-firmware**" directory to generate "**header.bin**" and "**firmware.bin**".

python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin xip



```
yzzhou@YZZHOU-LAPTOP MINGW64 /d/Project/Alps/Doc/Boot_up_time_acc/[0519]radar-sensor-firmware/radar-sensor-f
$ python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin xip
```

(4) After all the above steps, you can find two binary files, "**header.bin**" and "**firmware.bin**" under "**./radar-sensor-firmware**" directory.



data.dat	2020/5/19 19:02
firmware.bin	2020/5/19 19:02
firmware_release_version.txt	2020/5/19 15:05
header.bin	2020/5/19 19:02
header_gen.py	2020/5/19 15:05

(5) For how to program "**header.bin**" and "**firmware.bin**" into external flash, please refer to the “**Flash Downloader User Guide**” document in software release package under “**Document**” directory.

CHAPTER

THREE

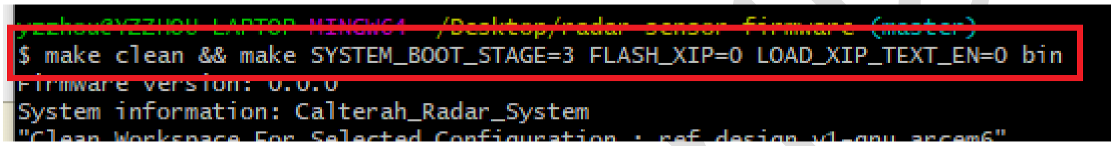
THREE-STAGE BOOT COMPILING

There are 3 types of two-stage boot: Normal mode, XIP mode and Boot split mode.

1. Normal mode:

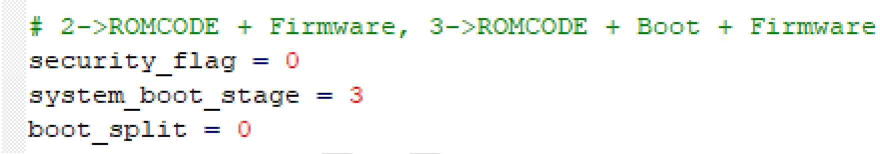
(1) Send below "**make**" command under "**./radar-sensor-firmware**" directory.

make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=0 LOAD_XIP_TEXT_EN=1 bin



```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=0 LOAD_XIP_TEXT_EN=0 bin
Firmware version: 0.0.0
System information: Calterah_Radar_System
"Clean Workspace For Selected Configuration : ref_design_v1-gnu_arcem6"
```

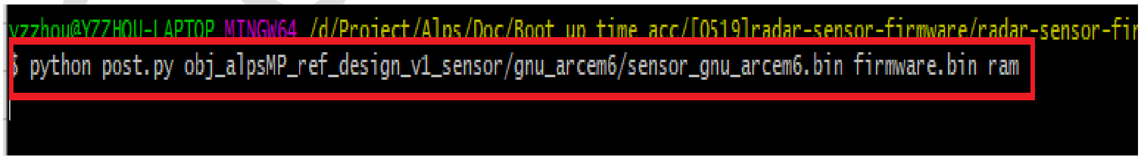
(2) Open "**post.py**" script with Notepad under "**./radar-sensor-firmware**" directory and change the value of "**system_boot_stage"** parameter to 3.



```
# 2->ROMCODE + Firmware, 3->ROMCODE + Boot + Firmware
security_flag = 0
system_boot_stage = 3
boot_split = 0
```

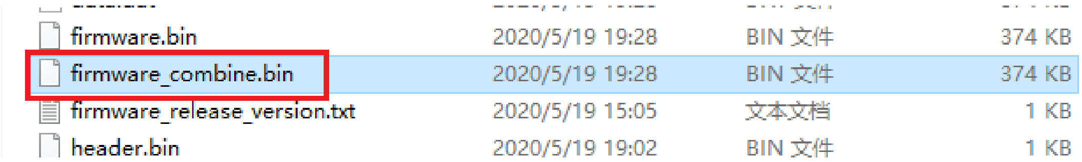
(3) Execute below "**post.py**" script under "**./radar-sensor-firmware**" directory to generate "**firmware_combine.bin**".

python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin ram



```
yzzhou@YZZHOU-LAPTOP MINGW64 /d/Project/Alps/Doc/Boot_up_time_acc/[0519]radar-sensor-firmware/radar-sensor-fir
$ python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin ram
```

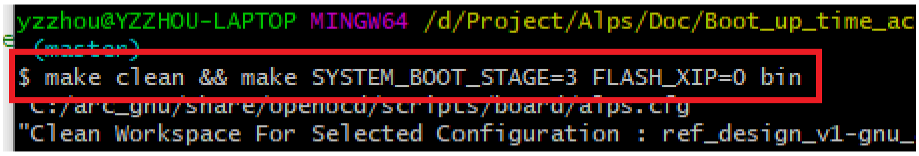
The generated "**firmware_combine.bin**" file is shown as below:



firmware.bin	2020/5/19 19:28	BIN 文件	374 KB
firmware_combine.bin	2020/5/19 19:28	BIN 文件	374 KB
firmware_release_version.txt	2020/5/19 15:05	文本文档	1 KB
header.bin	2020/5/19 19:02	BIN 文件	1 KB

(4) Send below "**make**" command under "**./radar-sensor-firmware/calterah/baremetal**" directory.

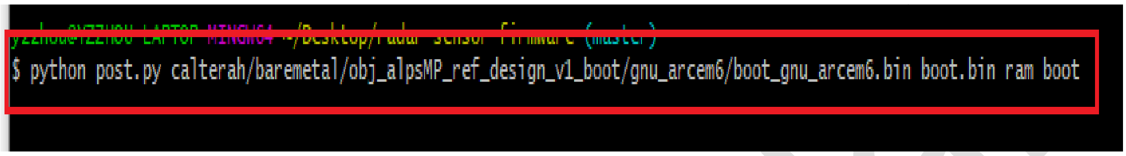
make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=0 bin



```
yzzhou@YZZHOU-LAPTOP MINGW64 /d/Project/Alps/Doc/Boot_up_time_ac
(master)
$ make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=0 bin
C:/arc_gnu/share/openocd/scripts/board/alps.cfg
"Clean Workspace For Selected Configuration : ref_design_v1-gnu_
```

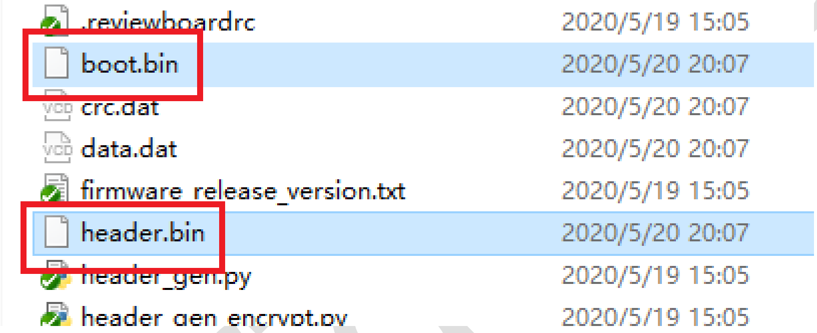
(5) Back to "**./radar-sensor-firmware**" directory, execute below "**post.py**" script to generate "**boot.bin**" and "**header.bin**".

python post.py calterah/baremetal/obj_alpsMP_ref_design_v1_boot/gnu_arcem6/boot_gnu_arcem6.bin boot.bin ram boot



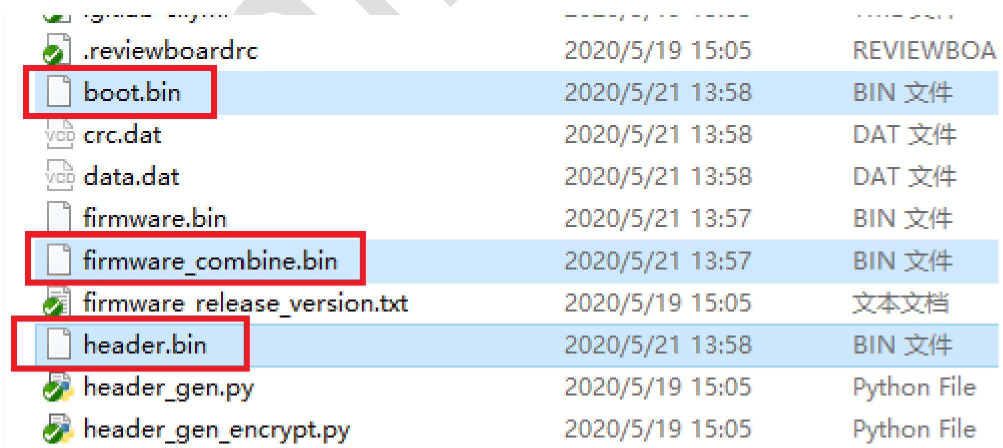
```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ python post.py calterah/baremetal/obj_alpsMP_ref_design_v1_boot/gnu_arcem6/boot_gnu_arcem6.bin boot.bin ram boot
```

The generated "**boot.bin**" and "**header.bin**" file are shown as below:



.reviewboardrc	2020/5/19 15:05
boot.bin	2020/5/20 20:07
crc.dat	2020/5/20 20:07
data.dat	2020/5/20 20:07
firmware_release_version.txt	2020/5/19 15:05
header.bin	2020/5/20 20:07
header_gen.py	2020/5/19 15:05
header_gen_encrypt.py	2020/5/19 15:05

(6) Right now, you will have 3 generated bin files under "**./radar-sensor-firmware**" directory: "**firmware_combine.bin**", "**boot.bin**" and "**header.bin**".



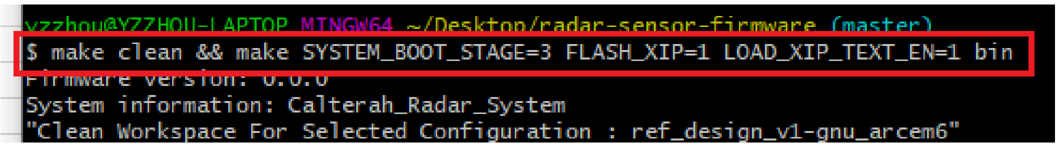
.reviewboardrc	2020/5/19 15:05	REVIEWBOA
boot.bin	2020/5/21 13:58	BIN 文件
crc.dat	2020/5/21 13:58	DAT 文件
data.dat	2020/5/21 13:58	DAT 文件
firmware.bin	2020/5/21 13:57	BIN 文件
firmware_combine.bin	2020/5/21 13:57	BIN 文件
firmware_release_version.txt	2020/5/19 15:05	文本文档
header.bin	2020/5/21 13:58	BIN 文件
header_gen.py	2020/5/19 15:05	Python File
header_gen_encrypt.py	2020/5/19 15:05	Python File

(7) For how to program "**firmware_combine.bin**", "**boot.bin**" and "**header.bin**" into external flash, please refer to the "**Flash Downloader User Guide**" document in software release package under "**Document**" directory.

2. XIP mode:

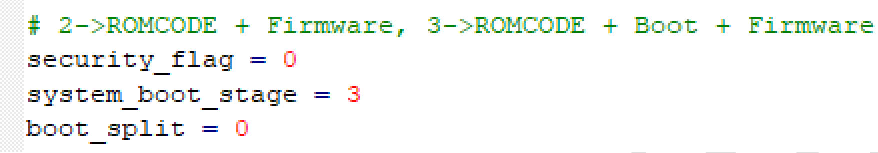
(1) Send below "**make**" command under "**./radar-sensor-firmware**" directory.

```
make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=1 LOAD_XIP_TEXT_EN=1 bin
```



```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=1 LOAD_XIP_TEXT_EN=1 bin
Firmware version: 0.0.0
System information: Calterah_Radar_System
"Clean Workspace For Selected Configuration : ref_design_v1-gnu_arcem6"
```

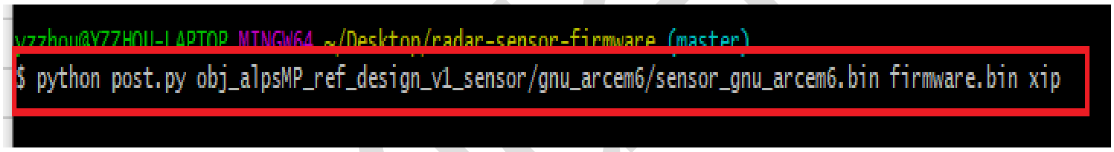
(2) Open "**post.py**" script with Notepad under "**./radar-sensor-firmware**" directory and change the value of "**system_boot_stage"** parameter to 3.



```
# 2->ROMCODE + Firmware, 3->ROMCODE + Boot + Firmware
security_flag = 0
system_boot_stage = 3
boot_split = 0
```

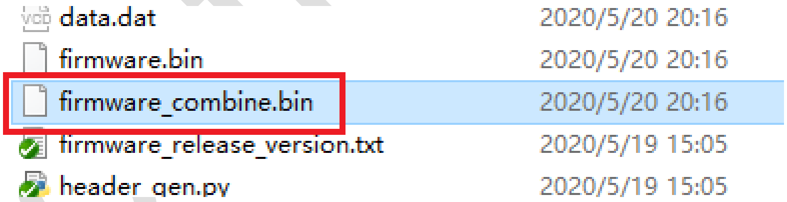
(3) Execute below "**post.py**" script under "**./radar-sensor-firmware**" directory to generate "**firmware_combine.bin**".

```
python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin xip
```



```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin xip
```

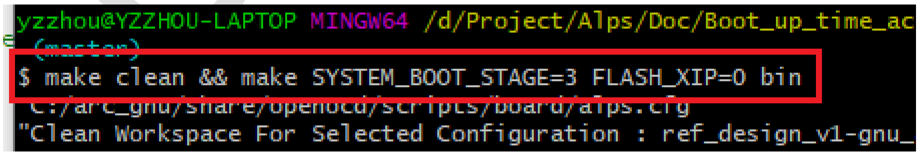
The generated "**firmware_combine.bin**" file is shown as below:



```
data.dat                        2020/5/20 20:16
firmware.bin                    2020/5/20 20:16
firmware_combine.bin            2020/5/20 20:16
firmware_release_version.txt    2020/5/19 15:05
header_gen.py                   2020/5/19 15:05
```

(4) Send below "**make**" command under "**./radar-sensor-firmware/calterah/baremetal**" directory.

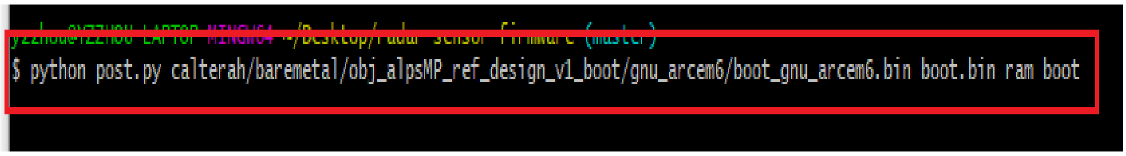
```
make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=0 bin
```



```
yzzhou@YZZHOU-LAPTOP MINGW64 /d/Project/Alps/Doc/Boot_up_time_ac
 (master)
$ make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=0 bin
 C:/arc_gnu/share/openocd/scripts/board/alps.cfg
"Clean Workspace For Selected Configuration : ref_design_v1-gnu_
```

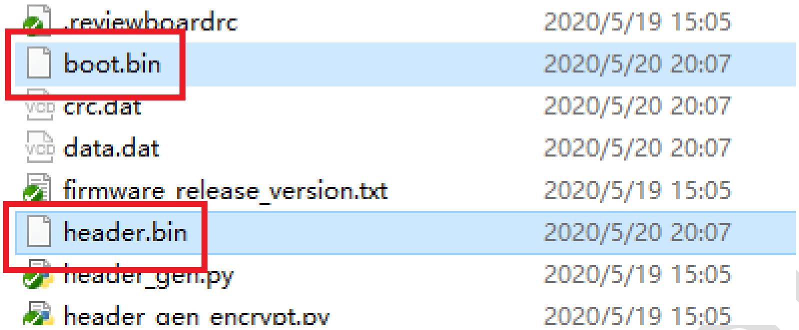
(5) Back to "**./radar-sensor-firmware**" directory, execute below "**post.py**" script to generate "**boot.bin**" and "**header.bin**".

```
python post.py calterah/baremetal/obj_alpsMP_ref_design_v1_boot/gnu_arcem6/boot_gnu_arcem6.bin boot.bin ram boot
```



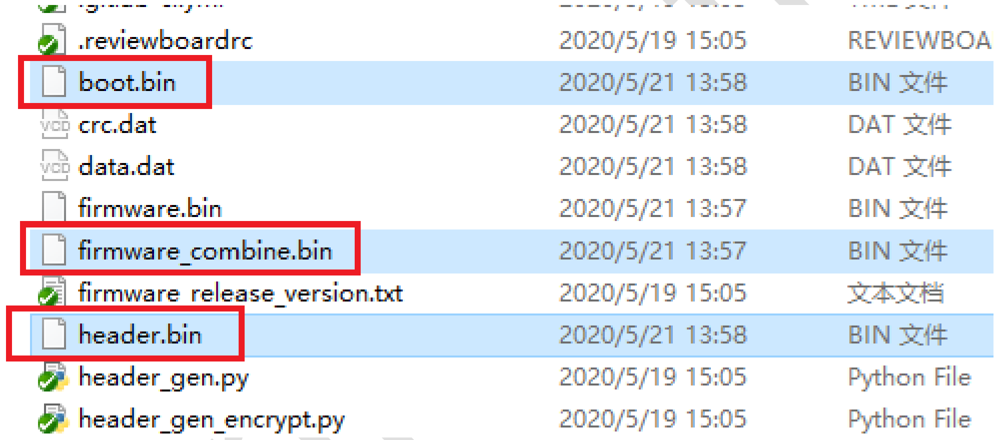
```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ python post.py calterah/baremetal/obj_alpsMP_ref_design_v1_boot/gnu_arcem6/boot_gnu_arcem6.bin boot.bin ram boot
```

The generated "**boot.bin**" and "**header.bin**" file are shown as below:



.reviewboardrc	2020/5/19 15:05
boot.bin	2020/5/20 20:07
crc.dat	2020/5/20 20:07
data.dat	2020/5/20 20:07
firmware_release_version.txt	2020/5/19 15:05
header.bin	2020/5/20 20:07
header_gen.py	2020/5/19 15:05
header_gen_encrypt.py	2020/5/19 15:05

(6) Right now,you will have 3 generated bin files under "**./radar-sensor-firmware**" directory: "**firmware_combine.bin**", "**boot.bin**" and "**header.bin**".



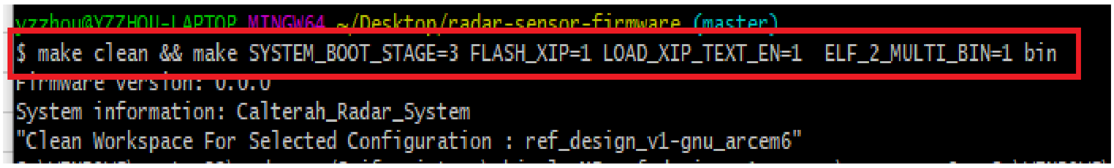
.reviewboardrc	2020/5/19 15:05	REVIEWBOA
boot.bin	2020/5/21 13:58	BIN 文件
crc.dat	2020/5/21 13:58	DAT 文件
data.dat	2020/5/21 13:58	DAT 文件
firmware.bin	2020/5/21 13:57	BIN 文件
firmware_combine.bin	2020/5/21 13:57	BIN 文件
firmware_release_version.txt	2020/5/19 15:05	文本文档
header.bin	2020/5/21 13:58	BIN 文件
header_gen.py	2020/5/19 15:05	Python File
header_gen_encrypt.py	2020/5/19 15:05	Python File

(7) For how to program "**firmware_combine.bin**", "**boot.bin**" and "**header.bin**" into external flash, please refer to the "**Flash Downloader User Guide**" document in software release package under "**Document**" directory.

3. Boot split mode:

(1) Send below "**make**" command under "**./radar-sensor-firmware**" directory.

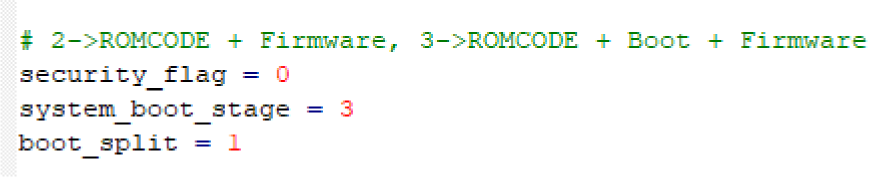
make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=1 LOAD_XIP_TEXT_EN=1 ELF_2_MULTI_BIN=1 bin



```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=1 LOAD_XIP_TEXT_EN=1  ELF_2_MULTI_BIN=1 bin
Firmware version: 0.0.0
System information: Calterah_Radar_System
"Clean Workspace For Selected Configuration : ref_design_v1-gnu_arcem6"
```

(2) Open "**post.py**" script with Notepad under "**./radar-sensor-firmware**" directory and change the value of "**system_boot_stage"** parameter to 3 and "**boot_split**" parameter

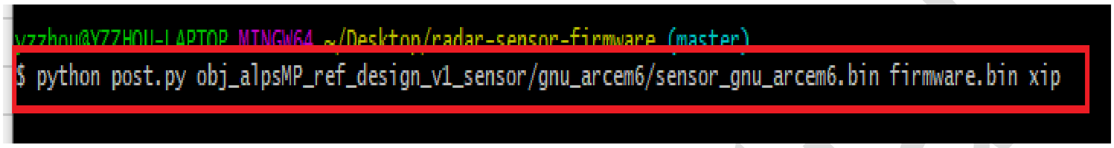
to 1.



```
# 2->ROMCODE + Firmware, 3->ROMCODE + Boot + Firmware
security_flag = 0
system_boot_stage = 3
boot_split = 1
```

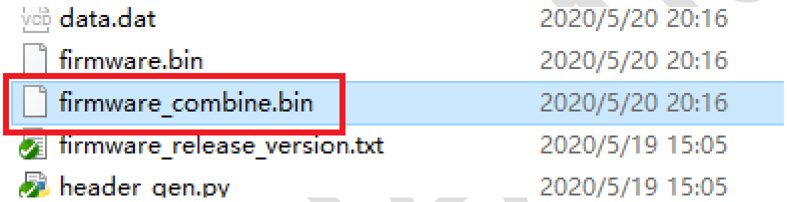
(3) Execute below "**post.py**" script under "**./radar-sensor-firmware**" directory to generate "**firmware_combine.bin**".

python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin xip



```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ python post.py obj_alpsMP_ref_design_v1_sensor/gnu_arcem6/sensor_gnu_arcem6.bin firmware.bin xip
```

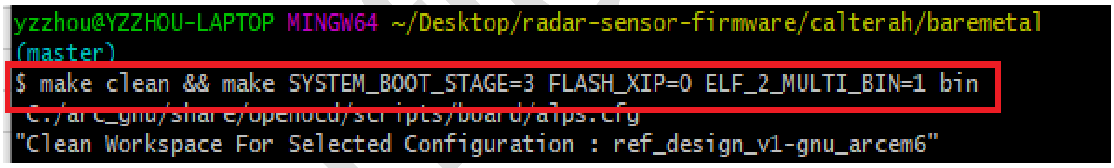
The generated "**firmware_combine.bin**" file is shown as below:



```
data.dat                      2020/5/20 20:16
firmware.bin                  2020/5/20 20:16
firmware_combine.bin          2020/5/20 20:16
firmware_release_version.txt  2020/5/19 15:05
header_gen.py                 2020/5/19 15:05
```

(4) Send below "**make**" command under "**./radar-sensor-firmware/calterah/baremetal**" directory.

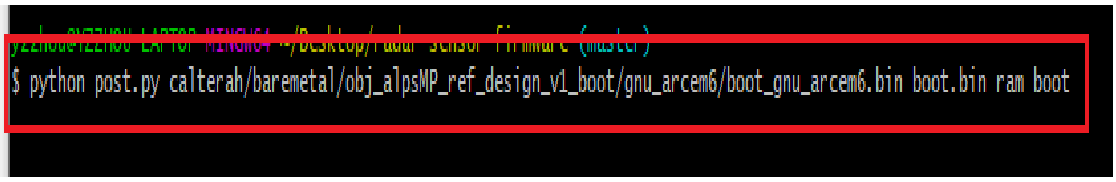
make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=0 ELF_2_MULTI_BIN=1 bin



```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware/calterah/baremetal
(master)
$ make clean && make SYSTEM_BOOT_STAGE=3 FLASH_XIP=0 ELF_2_MULTI_BIN=1 bin
 C:/arc_gnu/share/openocd/scripts/board/alps.cfg
"Clean Workspace For Selected Configuration : ref_design_v1-gnu_arcem6"
```

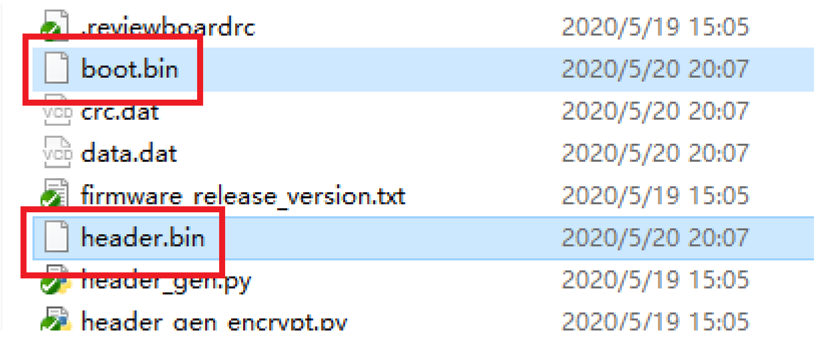
(5) Back to "**./radar-sensor-firmware**" directory, execute below "**post.py**" script to generate "**boot.bin**" and "**header.bin**".

python post.py calterah/baremetal/obj_alpsMP_ref_design_v1_boot/gnu_arcem6/boot_gnu_arcem6.bin boot.bin ram boot



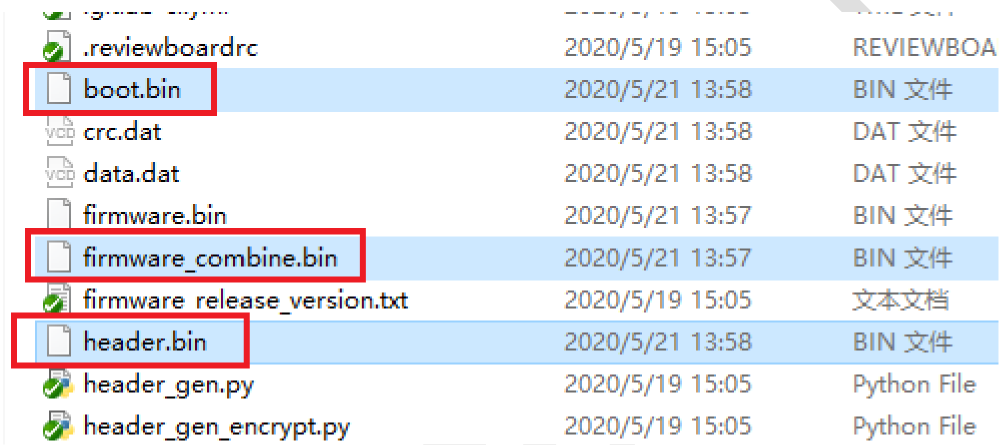
```
yzzhou@YZZHOU-LAPTOP MINGW64 ~/Desktop/radar-sensor-firmware (master)
$ python post.py calterah/baremetal/obj_alpsMP_ref_design_v1_boot/gnu_arcem6/boot_gnu_arcem6.bin boot.bin ram boot
```

The generated "**boot.bin**" and "**header.bin**" file are shown as below:



.reviewboardrc	2020/5/19 15:05
boot.bin	2020/5/20 20:07
crc.dat	2020/5/20 20:07
data.dat	2020/5/20 20:07
firmware_release_version.txt	2020/5/19 15:05
header.bin	2020/5/20 20:07
header_gen.py	2020/5/19 15:05
header_gen_encrypt.py	2020/5/19 15:05

(6) Right now,you will have 3 generated bin files under "**./radar-sensor-firmware**" directory: "**firmware_combine.bin**", "**boot.bin**" and "**header.bin**".



.reviewboardrc	2020/5/19 15:05	REVIEWBOA
boot.bin	2020/5/21 13:58	BIN 文件
crc.dat	2020/5/21 13:58	DAT 文件
data.dat	2020/5/21 13:58	DAT 文件
firmware.bin	2020/5/21 13:57	BIN 文件
firmware_combine.bin	2020/5/21 13:57	BIN 文件
firmware_release_version.txt	2020/5/19 15:05	文本文档
header.bin	2020/5/21 13:58	BIN 文件
header_gen.py	2020/5/19 15:05	Python File
header_gen_encrypt.py	2020/5/19 15:05	Python File

(7) For how to program "**firmware_combine.bin**", "**boot.bin**" and "**header.bin**" into external flash, please refer to the “**Flash Downloader User Guide**” document in software release package under “**Document**” directory.