

# **AAEON KL520 Firmware update from v0.95 to v1.0**

*Edit by Thunder Zhan*

2020/11/23

**STEP1:** Extract KL520\_SDK\_V1.0.zip & fw\_bin\_v095\_patch2\_AAEON.zip

**STEP2:** Open folder KL520\_SDK\_v1.0/host\_lib

**STEP3:** Open the terminal in this folder(right click open in terminal)and then run the following steps ° (following steps can refer to README.md in this folder)

**STEP4:** Run sudo apt install libusb-1.0-0-dev

**STEP5:** Enter user password

**STEP6:** Run mkdir build && cd build

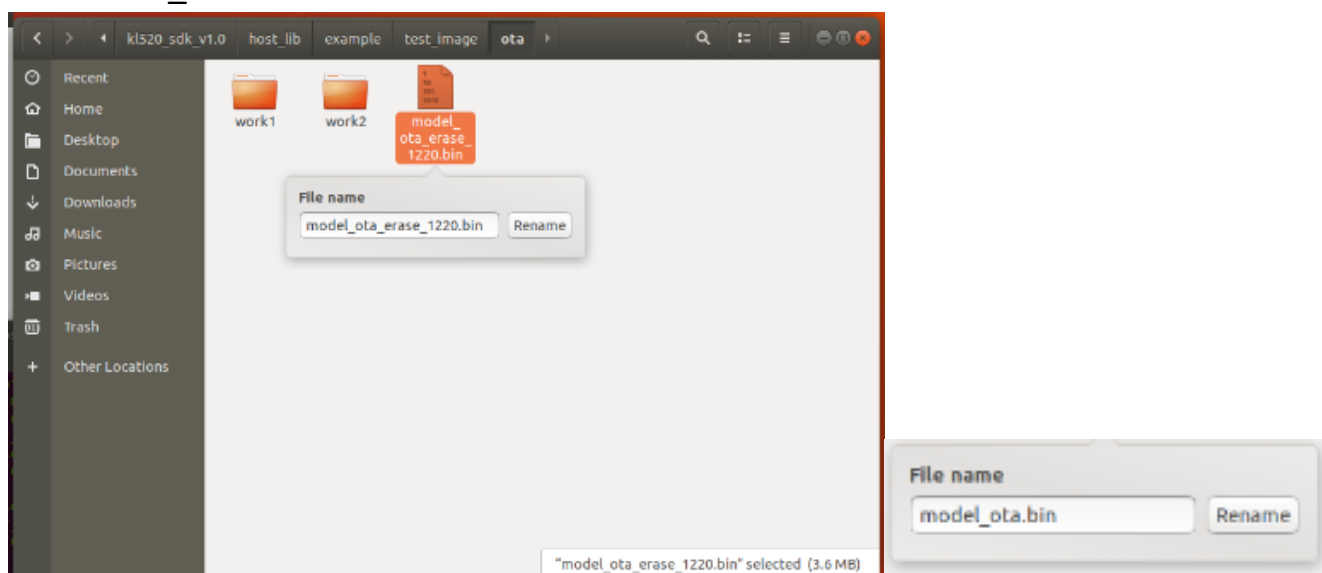
**STEP7:** Run cmake ..

**STEP8:** Run make -j4

**STEP9:**

- Open folder fw\_bin\_v095\_patch2\_AAEON.
- Copy model\_ota\_erase\_1220.bin to kl520\_sdk\_v1.0/host\_lib/example/test\_image/ota then renames the file to model\_ota.bin .

Note: Please remove all the files in this folder and only keep the bin file "model\_ota.bin".



**STEP10:** Go to `kl520_sdk_v1.0/host_lib/build/example`. Run `./udt_md` in terminal

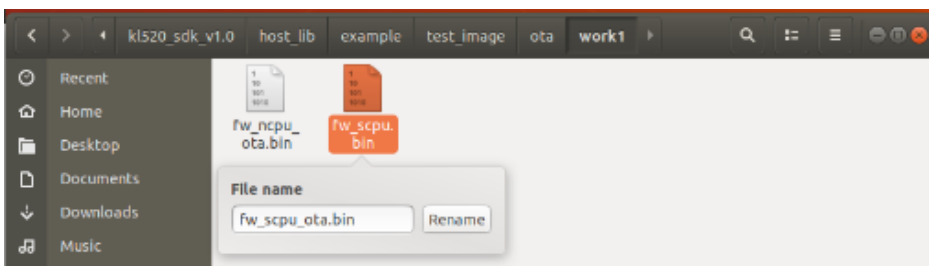
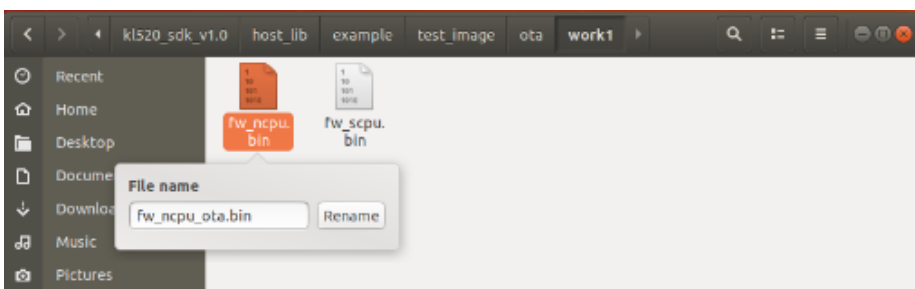
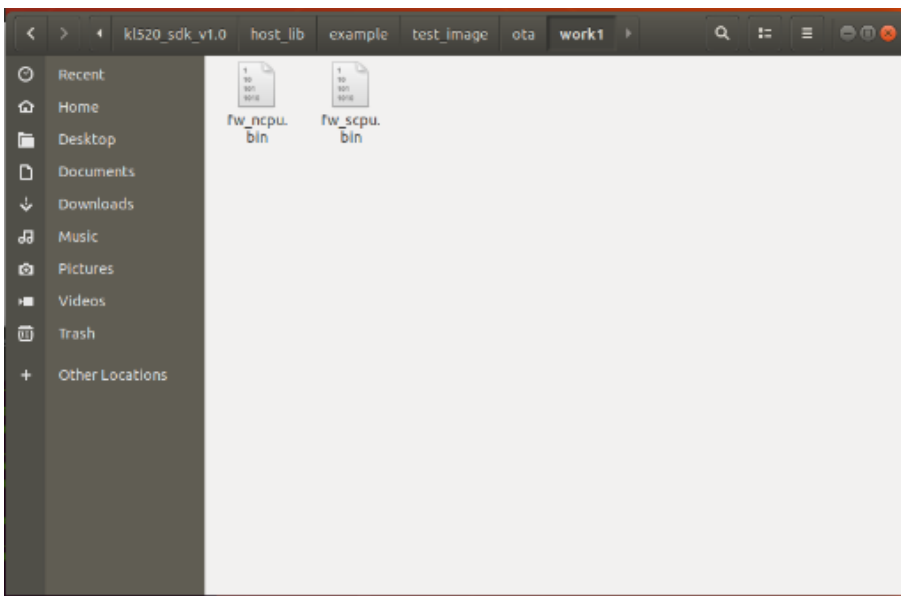
```

user@user-PICO-APL3: ~/Downloads/kl520_sdk_v1.0/host_lib/build/example
File Edit View Search Terminal Help
user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$ ./udt_md
init kdp host lib log...
adding devices...
start kdp host lib...
doing test :0...
starting update model :0...
update model succeeded...
de init kdp host lib...
user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$

```

**STEP11:**

- Put the compiled `fw_ncpu.bin` & `fw_scpu.bin`(★1) to `kl520_sdk_v1.0/host_lib/example/test_image/ota/work1`
  - Rename the files to `fw_ncpu_ota.bin` & `fw_scpu_ota.bin` accordingly
- Note: Please remove all the files in this folder and only keep these two compiled files



**STEP12:** Go to kl520\_sdk\_v1.0/host\_lib/build/example. Run ./udt\_fw 1 firstly then run ./udt fw 2 in terminal

```

user@user-PICO-APL3: ~/Downloads/kl520_sdk_v1.0/host_lib/build/example
File Edit View Search Terminal Help
user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$ ./udt_md
init kdp host lib log...
adding devices...
start kdp host lib...
doing test :0...
starting update model :0...
update model succeeded...
de init kdp host lib...
user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$ ./udt_fw
1
init kdp host lib log...
adding devices...
start kdp host lib...
doing test :0...
starting update fw ...
update firmware succeeded...
de init kdp host lib...
user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$

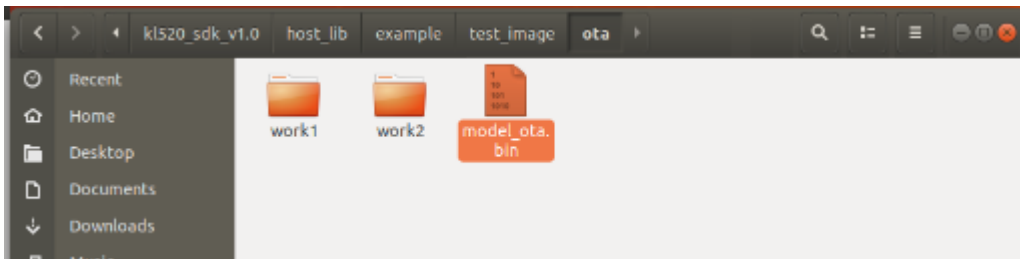
```

```

user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$ ./udt_fw
2
init kdp host lib log...
adding devices...
start kdp host lib...
doing test :0...
starting update fw ...
update firmware succeeded...
de init kdp host lib...
user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$

```

**STEP13:** Put the combined model\_ota.bin(★2) to kl520\_sdk\_v1.0/host\_lib/example/test\_image/ota  
Note: Please remove previous model\_ota.bin first



**STEP14:** Run ./udt\_md(★3)in terminal

```

user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$ ./udt_fw
2
init kdp host lib log...
adding devices...
start kdp host lib...
doing test :0...
starting update fw ...
update firmware succeeded...
de init kdp host lib...
user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$ ./udt_md
init kdp host lib log...
adding devices...
start kdp host lib...
doing test :0...
starting update model :0...
update model succeeded...
de init kdp host lib...
user@user-PICO-APL3:~/Downloads/kl520_sdk_v1.0/host_lib/build/example$

```

**STEP15:** Complete.

=====

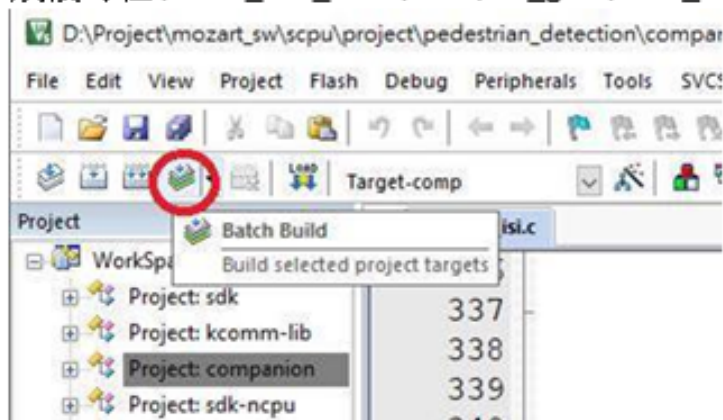
⊙ If needs to update the latest firmware after v1.0 Please refer to the previous document (The instruction update\_fw in host\_lib\_v0.4 .How to obtain the firmware please refer to ★1 .)

=====

## Appendix

### ★1

若要編譯 fw 相關的 binary，需使用有授權的 Keil C (ARM 開發編譯軟體)來開啟 \kl520\_sdk\_v1.0\example\_projects\companion 中的專案檔  
開啟後按下 Batch Build 即開始編譯 (下圖僅為編譯按鈕的示意，並非相同的專案)，編譯完成後可在 \kl520\_sdk\_v1.0\utils\bin\_gen\flash\_bin 找到對應的 fw binaries



If needs to compile the firmware, please compile with the Authorized Keil C to create the project file in kl520\_sdk\_v1.0\example\_projects\companion.

Open the project file and click on the “Batch Build” which starts the compilation. The corresponding compiled firmware could be found in kl520\_sdk\_v1.0\utils\bin\_gen\flash\_bin.

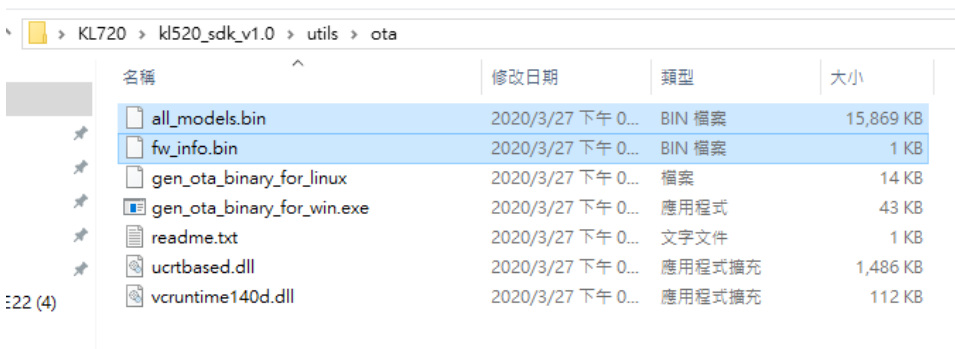
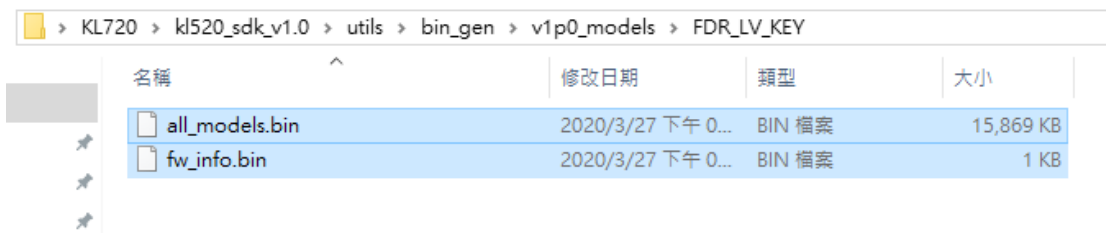
Note: Picture is as example only.

★2

Process of compiling model\_ota.bin

1. Copy the files fw\_info.bin & all\_models.bin in kl520\_sdk\_v1.0\utils\bin\_gen\v1p0\_models\FDR\_LV\_KEY and paste in kl520\_sdk\_v1.0/utils/ota

Note: The step is demonstrated in windows OS



2. Open command window in administrator mode then go to kl520\_sdk\_v1.0/utils/ota
3. Run gen\_ota\_binary\_for\_win.exe -model fw\_info.bin all\_models.bin model\_ota.bin



4. The combined model\_ota.bin will be produced in the following folder.

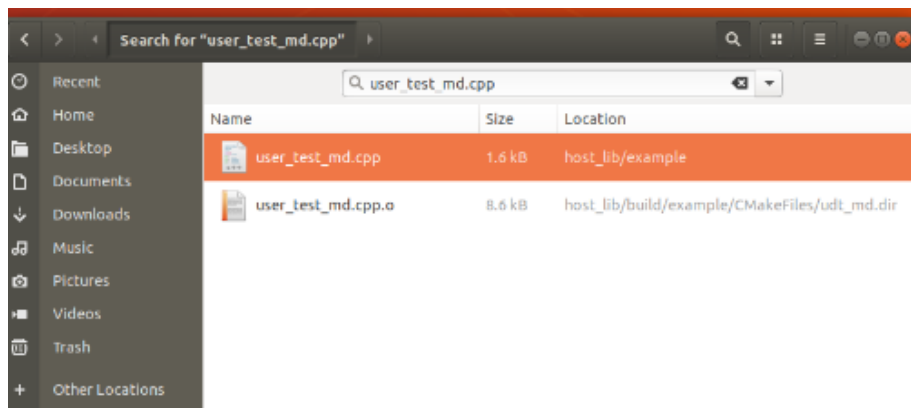
KL720 > kl520\_sdk\_v1.0 > utils > ota

名稱	修改日期	類型	大小
all_models.bin	2020/3/27 下午 0...	BIN 檔案	15,869 KB
fw_info.bin	2020/3/27 下午 0...	BIN 檔案	1 KB
gen_ota_binary_for_linux	2020/3/27 下午 0...	檔案	14 KB
gen_ota_binary_for_win.exe	2020/3/27 下午 0...	應用程式	43 KB
model_ota.bin	2020/11/3 下午 0...	BIN 檔案	15,873 KB
readme.txt	2020/3/27 下午 0...	文字文件	1 KB
ucrtbased.dll	2020/3/27 下午 0...	應用程式擴充	1,486 KB
vcruntime140d.dll	2020/3/27 下午 0...	應用程式擴充	112 KB

★3

Revising the parameter of the program is needed when the size of combined model\_ota.bin is exceeded than the default size of udt\_md.

1. Search user\_test\_md.cpp in kl520\_sdk\_v1.0



- Open the program and then find the code #define MD\_FILE\_SIZE (10\*1024\*1024)

```

Warning: This file is not part of any project. The code model might have issues to parse this file properly.
1 | #include <...>
2 |
3 | #include "errno.h"
4 | #include "kdp_host.h"
5 | #include "stdio.h"
6 |
7 | #include <string.h>
8 | #include <unistd.h>
9 | #include "user_util.h"
10 |
11 | #if defined(__cplusplus) || defined(cplusplus)
12 | extern "C" {
13 | #endif
14 |
15 | #define MODEL_FILE (TEST_OTA_DIR "model_ota.bin")
16 | #define MD_FILE_SIZE (20 * 1024 * 1024)
17 |
18 | //user test update model
19 | int user_test_md(int dev_idx, int user_id)
20 | {
21 |     //udt model
22 |     uint32_t model_id = user_id;
23 |     uint32_t buf_len = 0;
24 |     int buf_len_ret = 0;
25 |     char* p_buf = new char[MD_FILE_SIZE];
26 |     if(!p_buf) {
27 |         printf("memory fail...\n");
28 |         return -1;
29 |     }
30 |
31 |     memset(p_buf, 0, MD_FILE_SIZE);
32 |     printf("starting update model :%d...\n", model_id);
33 |
34 |     buf_len_ret = read_file_to_buf(p_buf, MODEL_FILE, MD_FILE_SIZE);
35 |     if (buf_len_ret <= 0) {
36 |         printf("reading file to buf failed:%d...\n", buf_len_ret);
37 |         return -1;
38 |     }
39 | }

```

- Revise the parameter to (20\*1024\*1024)

```

20 | #define MODEL_FILE (TEST_OTA_DIR "model_ota.bin")
21 | #define MD_FILE_SIZE (20 * 1024 * 1024)
22 |

```