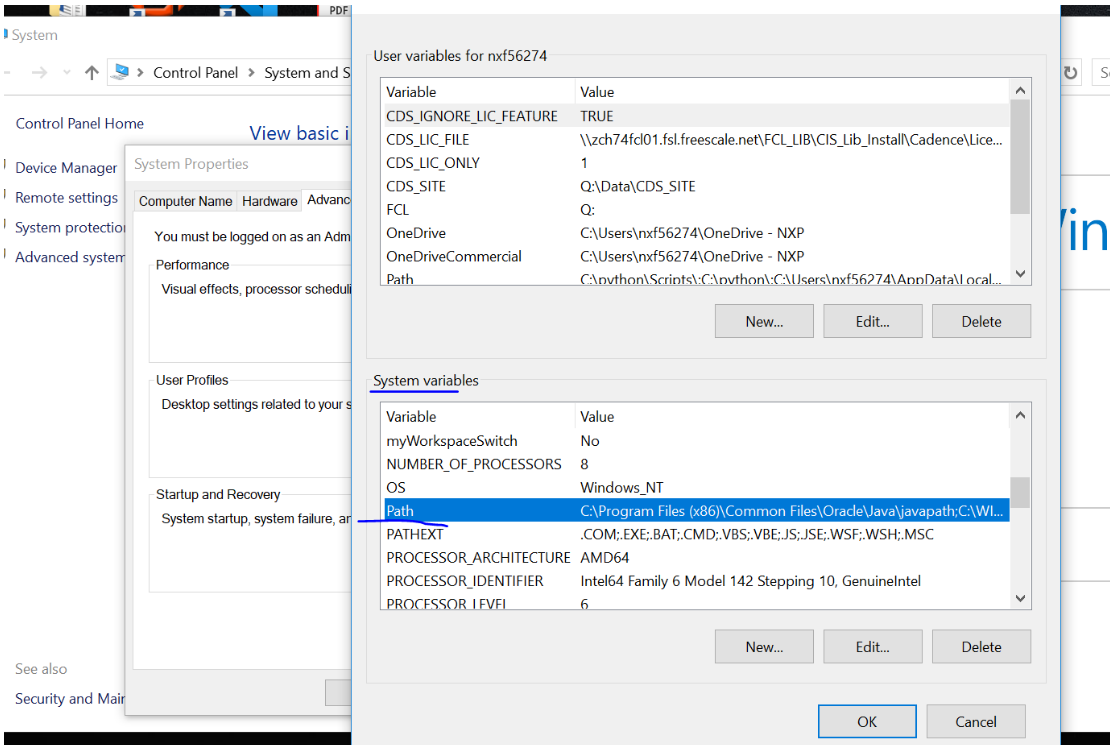
Compile the firmware by yourself

Our debug firmware is generally downloaded from the official website of nxp. nxp.com/opensda. But sometimes we want to modify the source code of bootloader and firmware according to our own requirements. So we introduce the open source project daplink. Arm Mbed DAPLink is an open source software project that can program and debug application software running on the Arm Cortex CPU. DAPLink is usually called interface firmware, and it runs on the auxiliary MCU connected to the SWD or JTAG port of the application MCU.
It provides k20 bootloader and interface firmware and k26 bootloader and interface firmware. Many frdm boards use k20 as a debugger, and a few boards use k26 as a debugger.
board：FRDM-K64
OS：WIN10

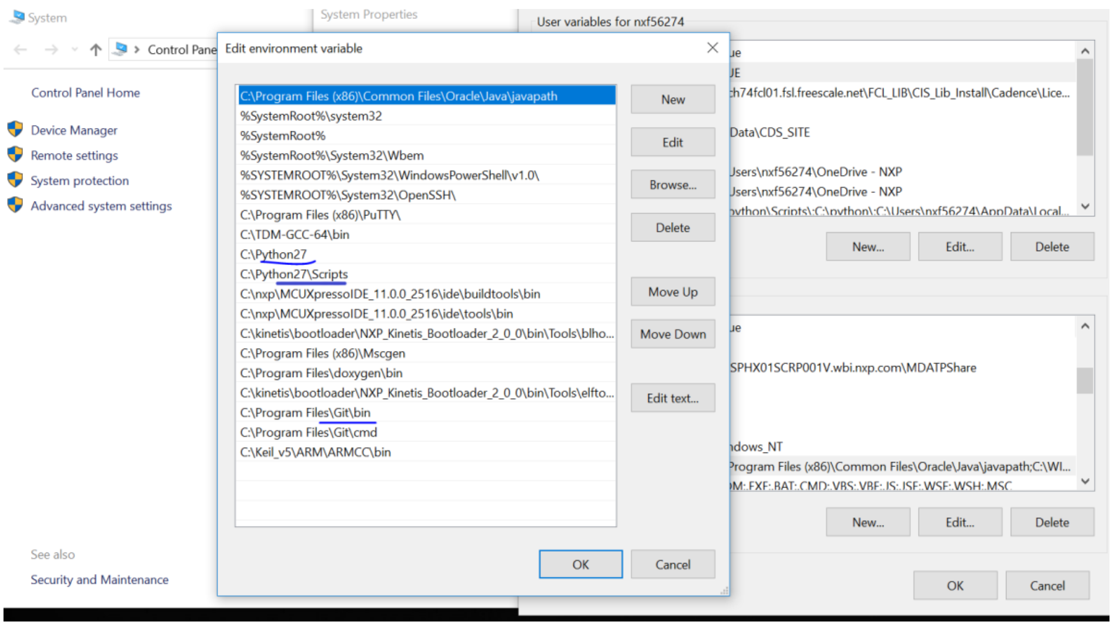
steps：
1.Install git, python2.7.11 or above, add these two software to the computer system environment variables (required), it is best to add the scripts folder under python to the environment variables, and install keil. DAPlink currently only supports IDE keil.


User variables for nxf56274

Variable	Value
CDS_IGNORE_LIC_FEATURE	TRUE
CDS_LIC_FILE	\\zch74fcl01.fsl.freescale.net\FCL_LIB\CIS_Lib_Install\Cadence\Lice...
CDS_LIC_ONLY	1
CDS_SITE	Q:\Data\CDS_SITE
FCL	Q:
OneDrive	C:\Users\nxf56274\OneDrive - NXP
OneDriveCommercial	C:\Users\nxf56274\OneDrive - NXP
Path	C:\python\Scripts\;C:\python\;C:\Users\nxf56274\AppData\Local...

System variables

Variable	Value
myWorkspaceSwitch	No
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\WI...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 10, GenuineIntel
PROCESSOR_LEVEL	6



```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\PuTTY\
C:\TDM-GCC-64\bin
C:\Python27
C:\Python27\Scripts
C:\nxp\MCUXpressoIDE_11.0.0_2516\ide\buildtools\bin
C:\nxp\MCUXpressoIDE_11.0.0_2516\ide\tools\bin
C:\kinetis\bootloader\NXP_Kinetis_Bootloader_2_0_0\bin\Tools\blho...
C:\Program Files (x86)\Mscgen
C:\Program Files\doxygen\bin
C:\kinetis\bootloader\NXP_Kinetis_Bootloader_2_0_0\bin\Tools\elfto...
C:\Program Files\Git\bin
C:\Program Files\Git\cmd
C:\Keil_v5\ARM\ARMCC\bin
```

```
ue
JE
:h74fcl01.fsl.freescale.net\FCL_LIB\CIS_Lib_Install\Cadence\Lice...
Data\CDS_SITE
Jsers\nxf56274\OneDrive - NXP
Jsers\nxf56274\OneDrive - NXP
python\Scripts\;C:\python\;C:\Users\nxf56274\AppData\Local...
ue
SPHX01SCRP001V.wbi.nxp.com\MDATPShare
ndows_NT
Program Files (x86)\Common Files\Oracle\Java\javapath;C:\WI...
OM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
```

2. Use python to install pip, you can search for tutorials online

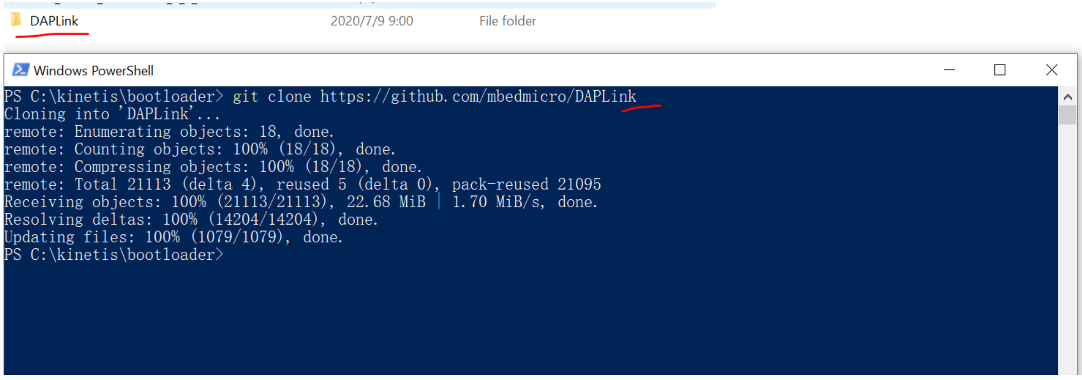
3. Install virtualenv, use powershell (hold down shift and click the right mouse button), Input 'pip install virtualenv'

4. After that, the commands are all completed under powershell. Get the source code.

   Input 'git clone https://github.com/mbedmicro/DAPLink'

   Note: You must use git to download the code, or you will fail at compiling the code.

   It Will generate a DAPLink folder in your current directory



```
DAPLink                                    2020/7/9 9:00         File folder
```

```
PS C:\kinetis\bootloader> git clone https://github.com/mbedmicro/DAPLink
Cloning into 'DAPLink'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 21113 (delta 4), reused 5 (delta 0), pack-reused 21095
Receiving objects: 100% (21113/21113), 22.68 MiB | 1.70 MiB/s, done.
Resolving deltas: 100% (14204/14204), done.
Updating files: 100% (1079/1079), done.
PS C:\kinetis\bootloader>
```

5.Enter the directory. Input 'cd DAPLink', The docs/DEVELOPERS-GUIDE.md under this folder is more detailed how to use this DAPLink

6.Create a virtual environment, Input 'virtualenv venv'

7.Input 'venv/Scripts/activate.bat' to active the virtual environment

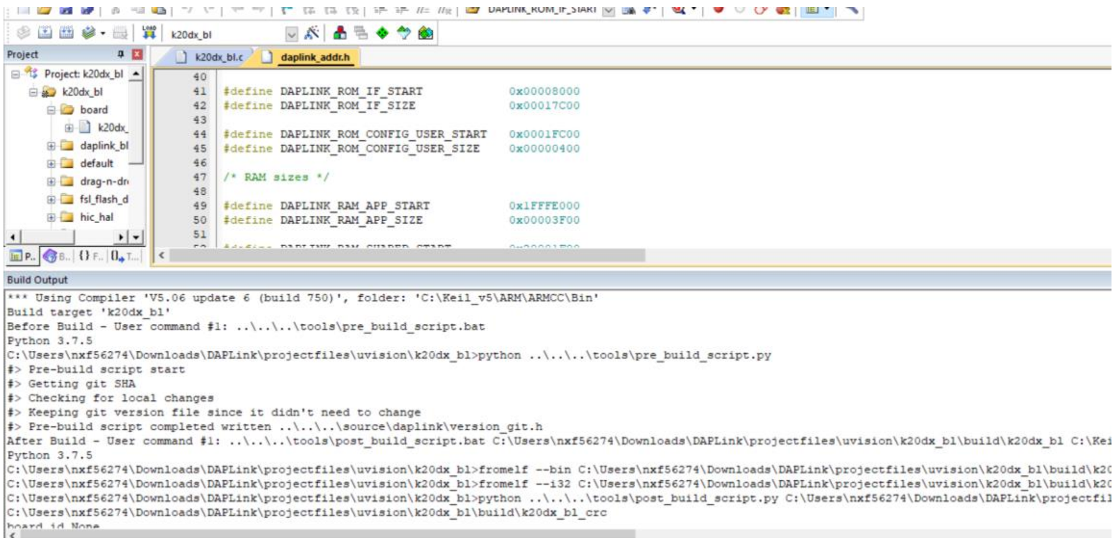
8.Install necessary tools, 'pip install -r requirements.txt'

9.Generate keil project,

   input 'progen generate -t uvision'

   It will generate projectfiles/uvision, enter the folder and you will find various bootloader and firmware. The name with 'bl' is the bootloader, and the name with 'if' is the interface firmware, which is to be dragged into the mcu. Open the first project about k20. After compilation, a bin file will be generated. The bin file with crc is what we want to burn or drag. For the name 'if' is the same. The git command will be called during compilation. If you do not add this command to the environment variable, the compilation will fail.

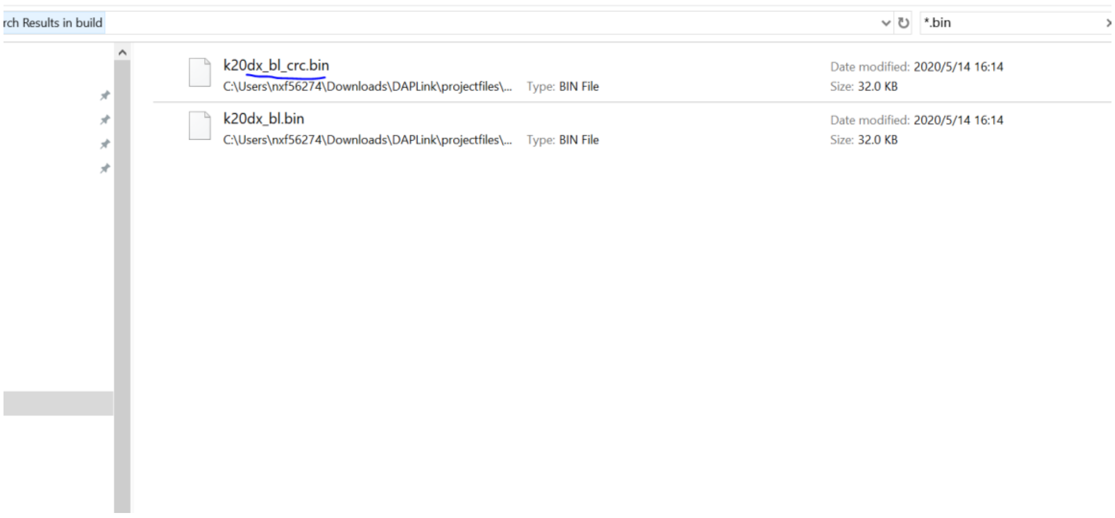
This is the bootloader source code



```
#define DAPLINK_ROM_IF_START            0x00008000
#define DAPLINK_ROM_IF_SIZE             0x00017C00

#define DAPLINK_ROM_CONFIG_USER_START   0x0001FC00
#define DAPLINK_ROM_CONFIG_USER_SIZE    0x00000400

/* RAM sizes */

#define DAPLINK_RAM_APP_START           0x1FFFE000
#define DAPLINK_RAM_APP_SIZE            0x00003F00
```

```
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'k20dx_bl'
Before Build - User command #1: ..\..\..\tools\pre_build_script.bat
Python 3.7.5
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\uvision\k20dx_bl>python ..\..\..\tools\pre_build_script.py
#> Pre-build script start
#> Getting git SHA
#> Checking for local changes
#> Keeping git version file since it didn't need to change
#> Pre-build script completed written ..\..\..\source\daplink\version_git.h
After Build - User command #1: ..\..\..\tools\post_build_script.bat C:\Users\nxf56274\Downloads\DAPLink\projectfiles\uvision\k20dx_bl\build\k20dx_bl C:\Kei
Python 3.7.5
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\uvision\k20dx_bl>fromelf --bin C:\Users\nxf56274\Downloads\DAPLink\projectfiles\uvision\k20dx_bl\build\k20
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\uvision\k20dx_bl>fromelf --i32 C:\Users\nxf56274\Downloads\DAPLink\projectfiles\uvision\k20dx_bl\build\k20
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\uvision\k20dx_bl>python ..\..\..\tools\post_build_script.py C:\Users\nxf56274\Downloads\DAPLink\projectfil
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\uvision\k20dx_bl\build\k20dx_bl_crc
board_id None
```

Bin file

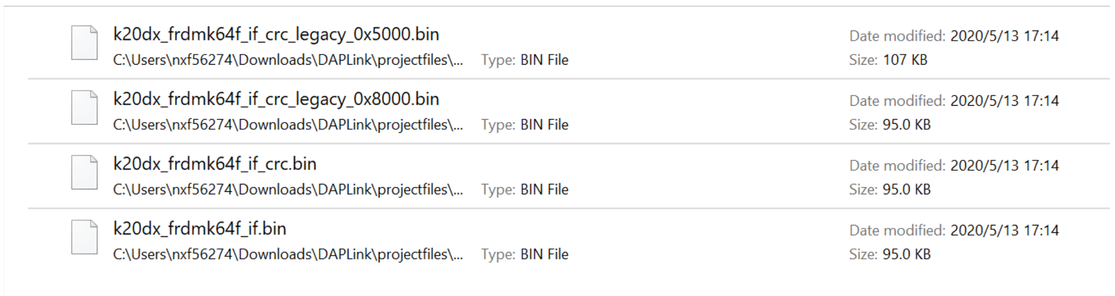

rch Results in build

*.bin

k20dx_bl_crc.bin
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\... Type: BIN File
Date modified: 2020/5/14 16:14
Size: 32.0 KB

k20dx_bl.bin
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\... Type: BIN File
Date modified: 2020/5/14 16:14
Size: 32.0 KB

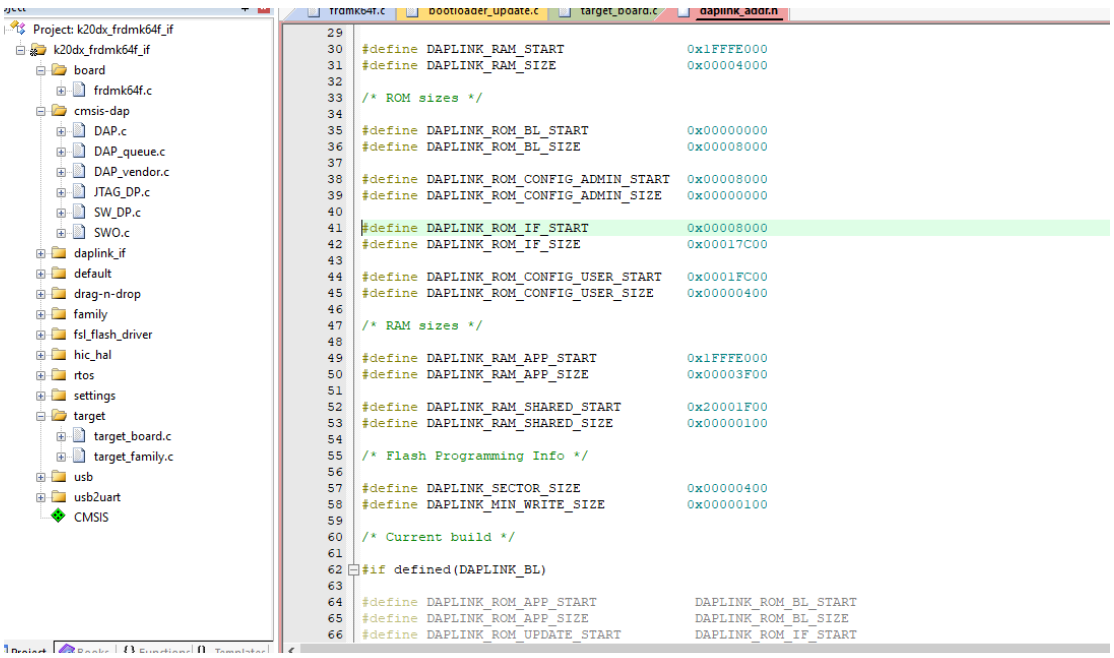
This is interface firmware.


k20dx_frdmk64f_if_crc_legacy_0x5000.bin
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\... Type: BIN File
Date modified: 2020/5/13 17:14
Size: 107 KB

k20dx_frdmk64f_if_crc_legacy_0x8000.bin
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\... Type: BIN File
Date modified: 2020/5/13 17:14
Size: 95.0 KB

k20dx_frdmk64f_if_crc.bin
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\... Type: BIN File
Date modified: 2020/5/13 17:14
Size: 95.0 KB

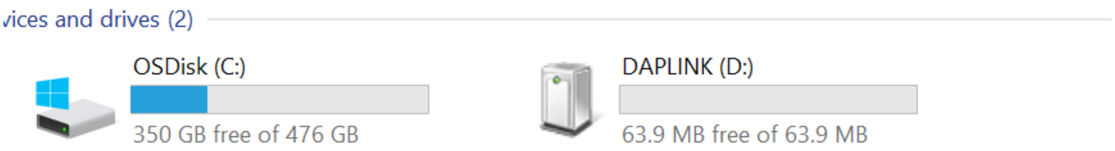
k20dx_frdmk64f_if.bin
C:\Users\nxf56274\Downloads\DAPLink\projectfiles\... Type: BIN File
Date modified: 2020/5/13 17:14
Size: 95.0 KB



```
#define DAPLINK_RAM_START                 0x1FFFE000
#define DAPLINK_RAM_SIZE                  0x00004000

/* ROM sizes */

#define DAPLINK_ROM_BL_START              0x00000000
#define DAPLINK_ROM_BL_SIZE               0x00008000

#define DAPLINK_ROM_CONFIG_ADMIN_START    0x00008000
#define DAPLINK_ROM_CONFIG_ADMIN_SIZE     0x00000000

#define DAPLINK_ROM_IF_START              0x00008000
#define DAPLINK_ROM_IF_SIZE               0x00017C00

#define DAPLINK_ROM_CONFIG_USER_START     0x0001FC00
#define DAPLINK_ROM_CONFIG_USER_SIZE      0x00000400

/* RAM sizes */

#define DAPLINK_RAM_APP_START             0x1FFFE000
#define DAPLINK_RAM_APP_SIZE              0x00003F00

#define DAPLINK_RAM_SHARED_START          0x20001F00
#define DAPLINK_RAM_SHARED_SIZE           0x00000100

/* Flash Programming Info */

#define DAPLINK_SECTOR_SIZE               0x00000400
#define DAPLINK_MIN_WRITE_SIZE            0x00000100

/* Current build */

#if defined(DAPLINK_BL)

#define DAPLINK_ROM_APP_START              DAPLINK_ROM_BL_START
#define DAPLINK_ROM_APP_SIZE               DAPLINK_ROM_BL_SIZE
#define DAPLINK_ROM_UPDATE_START           DAPLINK_ROM_IF_START
```

The generated bin file with '0x' is firmware address. Generally, the default firmware address of the DAPLink bootloader is 0x8000. As you can see from the above figure, this macro defines DAPLINK_ROM_IF_START, so the file we want to drag is the file with the name '0x8000'. If the firmware start address is modified in the bootloader, the interface firmware should also be modified accordingly

Burn the bootloader into k20, then drag the interface firmware into k20 to see this result


vices and drives (2)
OSDisk (C:)
350 GB free of 476 GB
DAPLINK (D:)
63.9 MB free of 63.9 MB