方糖电子

CUBIETECH LIMITED

CubieScreen S500 Transplant Guide

version：1.0

2018-0814

1 Table Of Content

2 Introduction

2.1 Purpose

This article focuses on the code transplantation of the CubieScreen S500. The main control used by CubieBoard6 is Action's S500, so the following is an example of CubieBoard6(CB6 for short), which helps enthusiasts and developers to quickly transplant CubieScreen to CB6 and then compile firmware that can use CubieScreen. At the same time, it can familiarize yourself with the basic architecture, configuration instructions and debugging methods of the S500 LCD and TP through this transplantation. In addition, this article has common problem descriptions to provide reference for the maintenance and modification of LCD and TP modules.

This article is intended for people who have some basic knowledge of the Android framework and the Linux driver architecture.

2.2 Terms And Abbreviations

Terms And Abbreviations	Explains
DE	Display Engine
DSS	Display subsystem
MIPI	Mobile Industry Processor Interface

3 Overall Design

3.1 Requirements

The CubieBoard6 supports the 3.5-inch CubieScreen, which integrates a display and a capacitive touchpad. The default screen resolution is 800*480pix, which fits perfectly with the CB6 pin.

Figure 3-1 CubieScreen On CB6

S500 supports RGB, LVDS, and MIPI interface. CubieScreen uses the RGB interface and it need initial code to initialize the LCD driver IC. The initial code can be added to the u-boot or kernel stage. The strategy used by CB6 is to initialize the IC through the GPIO simulate SPI in the u-boot stage.

In terms of TP, since the TP driver code is mainly modified according to the demo code provided by the TP module factory, different module codes are not the same, and the main work of TP debugging is transplant.

3.2 Overall Introduction

Based on the CB6 release SDK related driver, the Android system patch is based on the v3.0-20170523 Android SDK version and the Linux system patch is based on v1.0-20170210. If the SDK version does not correspond, you can refer to the patch to transplant manually.

CB6 currently supports capacitive touchpad, and it communicate with the main control IC via an I2C interface. The complete workflow is as follows: After the main control is powered on, the power and reset pins of the TP are powered up according to the power-on sequence defined by the TP specification. When the CTP is touched, the CTP generates an interrupt through the interrupt line, and in the interrupt processing (the second half), the main control IC communicates with the TP IC through the I2C interface, reads the coordinate value of the point, and then reports it to the input subsystem.

For LCD, CB6 currently supports the following types of LCD with a maximum resolution of 1920*1200 and supported interface types: RGB, LVDS, MIPI.

- RGB and LVDS screens use the IC LCD controller, and the associated drivers are in kernel\drivers\video\owl\displays\lcdc\
- LCD requires backlight control, the relevant driver is kernel\drivers\video\owl\backlight\
- The LCD display function is also supported in u-boot, and related drivers in u-boot\drivers\video\owl_lcd.c

The driver in u-boot mainly display the boot-logo interface. The driver only needs to define the relevant interface. The u-boot DSS driver calls the defined interface to implement the open screen action. The calling process is shown in Figure 3-2:

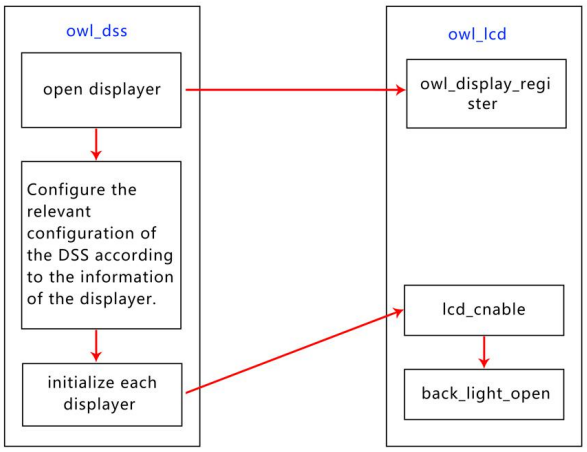

Figure 3-2 LCD Open Flow Chart

"owl_lcd.c" file is mainly responsible for the related operations of LVDS and RGB LCD, such as switching the LCD, initializing the LCD, reading some configuration items of the LCD, and so on. The opening screen action is implemented in the "lcd_enable" function. According to the above figure, after the screen is opened, the driver will call the relevant function to initialize the LCD. The GPIO simulate SPI is used to send the control command to the LCD controller, then the backlight is turned on and the boot-logo is displayed.

For details, refer to the LCD driver in the SDK.

4 Preparation Before Transplant

4.1 Schematic

CB6 and CubieScreen are pin-to-pin. In the following, figure 4-1 shows the extension pin part of CB6 schematic, LCD and TP related pins are on U24 and U23. Figure 4-2 and Figure 4-3 are part of the schematic of CubieScreen. In Figure 4-1 and Figure 4-2, U23 and U24 of CubieBoard6 schematic correspond to CON2 and CON1 of the CubieScreen schematic respectively.

The following schematics need to be combined to confirm the configuration:

- Which I2C is used by TP?

In Figure 4-2, red box 1 corresponds to the arrow 1 of CON2, since U23 in the CB6 schematic corresponds to CON2 of the CubieScreen schematic. Find the 7th pin of U23 in Figure 4-1 (connected to TWI1), so the I2C used by TP is I2C1.

- Which power and reset pin are used by the LCD?

Power:

Combined with the following three figures, and the arrow 2 in figure 4-2 corresponds to the red box 2 in figure 4-3. U24 of the CubieBoard6 schematic corresponds to CON1 of the CubieScreen schematic. Find the 40th pin (LCD0-SCK) of U24 in figure 4-1, in the complete schematic of CubieBoard6, you can find the 40th pin of U24, corresponding to GPIOE2/TWI2_SCLK, so power gpio uses GPIOE2.

Reset:

Use the same method can find the 39th pin(LCD0-SDA) of the U24 in figure 4-1, corresponding to GPIOE3/TWI2_SDATA, so the reset gpio uses GPIOE3.

- Which backlight enable pin and PWM used by the LCD?

In the schematic of the backlight part in figure 4-3, it can be learned that the backlight power is not controlled by GPIO, that is, it is directly enabled when the CB6 is power on, so the backlight enable gpio can be configured without configuration in the software configuration. In addition, in the red box 4 in figure 4-3, the PWM-BL can be found, corresponding to the 31st pin of CON1 in figure 4-2, and then the 31st pin of the U24 in the figure 4-1 is found, so PWM-BL is PWM0.

4.2 Related Information Download

Users can obtain CB6 and CubieScreen related information by MEGA.

CB6 hardware information: Board -> CubieBoard6 -> Hardware

CubieScreen information: Add-On -> CubieScreen

Mega link:

https://mega.nz/#F!ZtwxCCJC!AIYHcTqz-ucjuzKnE9qD7A

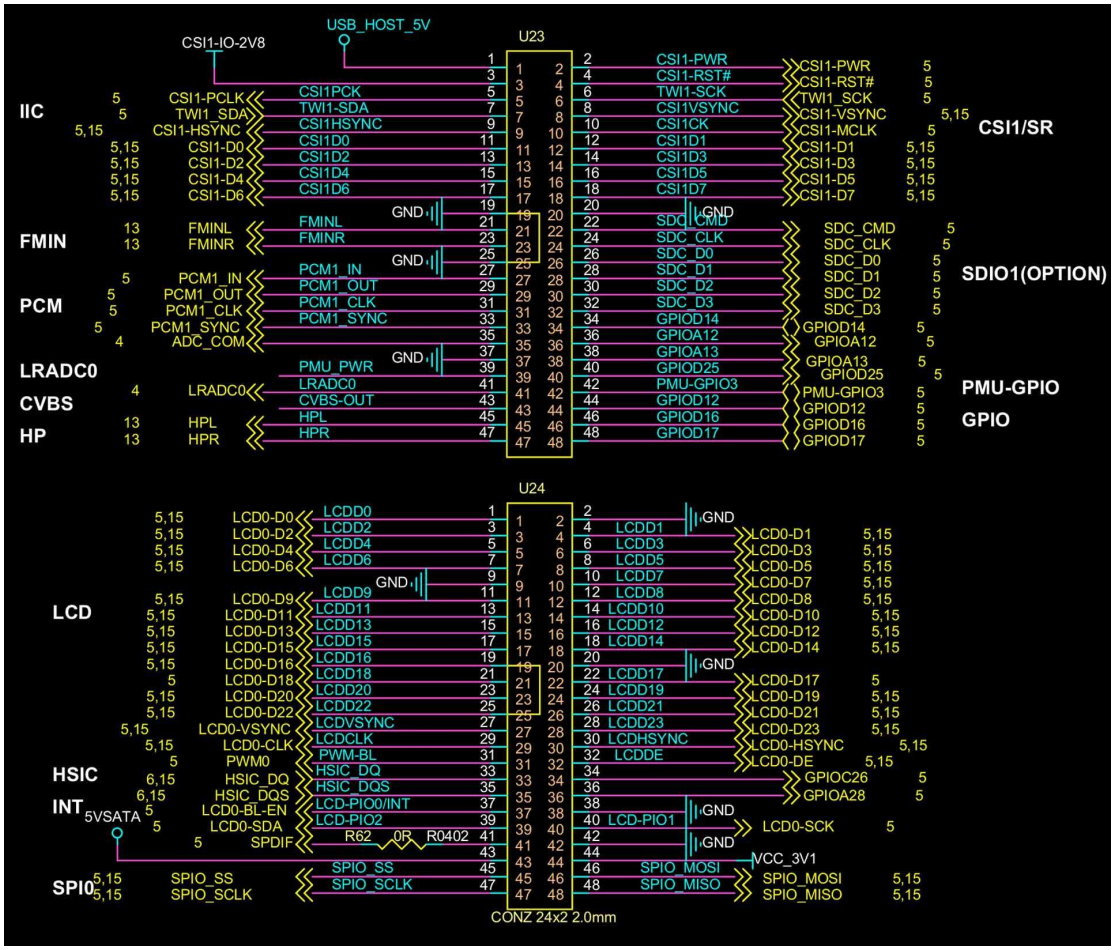

Figure 4-1 Extension Pin Part Of CB6 Schematic

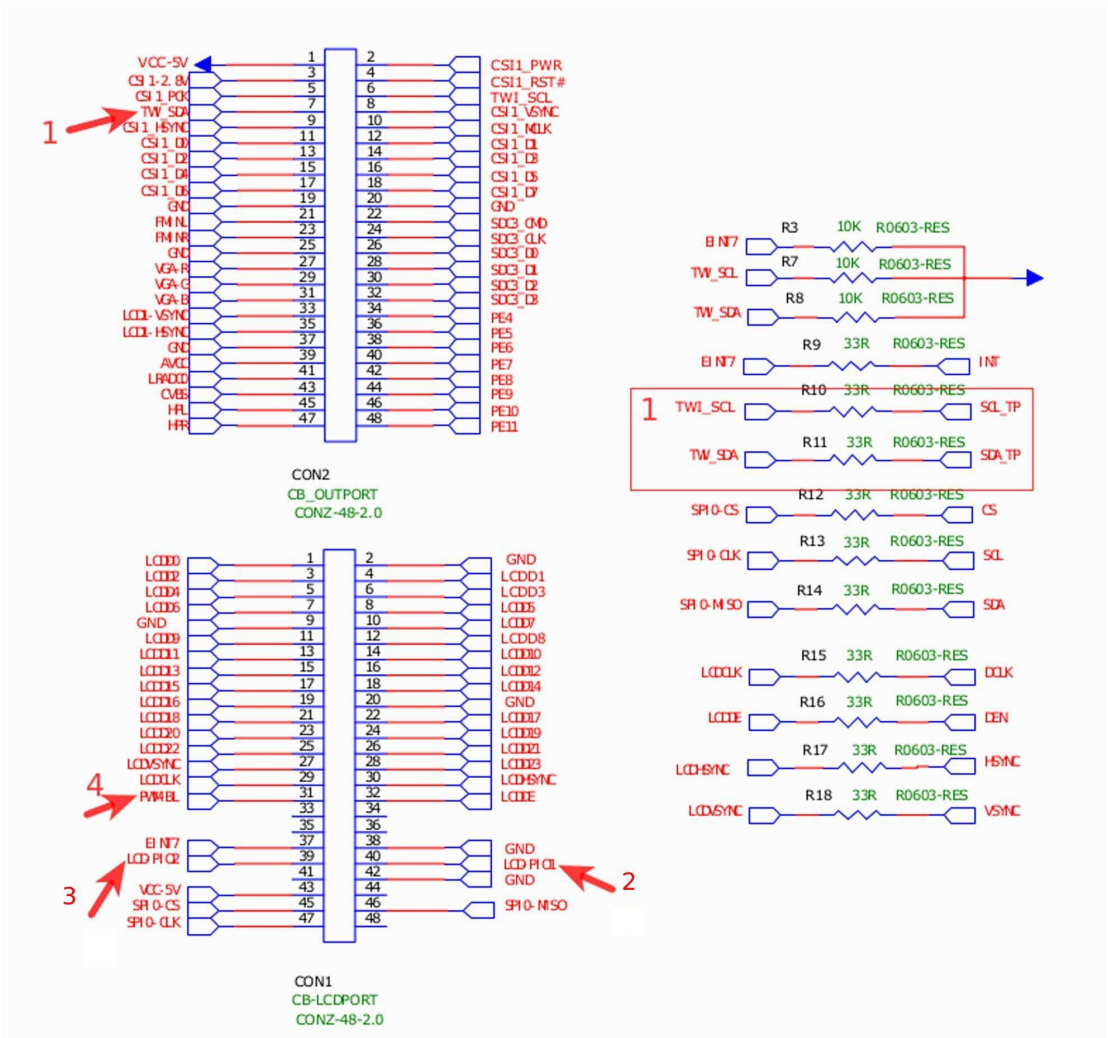

Figure 4-2 Part Of CubieScreen Schematic

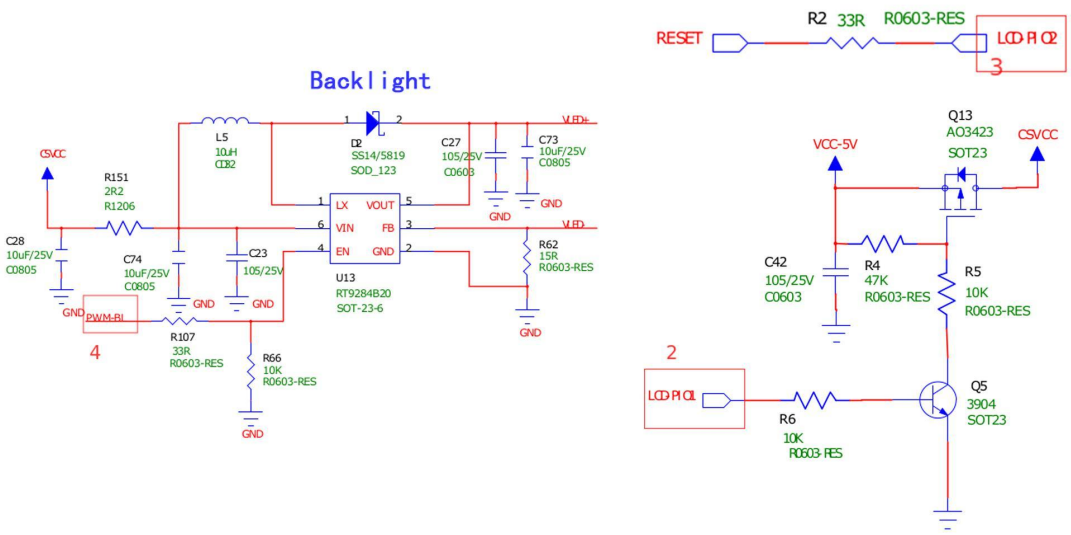

Figure 4-3 Part Of CubieScreen Schematic

4.3 Driver Configuration

Android & Linux systems, the same as the underlying implementation, configure DTS, kernel config, etc.

CubieScreen's transplantation debugging focuses on the two modules of display and touch screen. LCD (RGB) needs to configure DTS in the driver and add initial code, and the TP needs to add a new driver to the original SDK, no need to configure DTS.

4.3.1 LCD Driver Configuration

(1) DTS configuration

In the section 4.1 "Schematics" of this article, the following configurations were learned by combining the schematics of CubieBoard6 and CubieScreen: I2C used by TP, power and reset pins used by LCD, LCD backlight enable pin and PWM.

LCD DTS configuration, kernel and u-boot dts need to be consistent.

```
lcd0: lcd0@b02a0000 {
        status = "okay";
        pinctrl-names = "default";
```

```
        pinctrl-0 = <&rgb_state_default>;          /*select RGB pin configuration*/
        lcd_power_gpios = <&gpio 130 0>;          /*GPIOE(2)    0: high active*/
        lcd_reset_gpios = <&gpio 131 0>;          /*GPIOE(3)    0: low active */
        lcd_spi_csx = <&gpio 73 0>;               /*gpio simulate spi, send initial code in driver*/
        lcd_spi_sdi = <&gpio 71 0>;
        lcd_spi_sck = <&gpio 72 0>;
        port_type = "rgb";
        data_width = <0>;  /*date width，0: 24bit, 1: 18bit, 2: 16bit*/
        videomode-0 = <&lcd_mode0>;
        vsync_inversion = <0>;  /*vsync low active */
        hsync_inversion = <0>; /*vsync low active */
        dclk_inversion = <0>;    /*dclk, effective data acquisition edge of the signal,0: Falling edge
active */
         lde_inversion = <0>; /*DATA ENABLE, 0: high active*/
         lvds_ctl = <0x000a9521>;
         lvds_alg_ctl0 = <0xc141a030>;
         lightness = <125>;
         saturation = <7>;
         contrast = <7>;
         lcd_mode0:mode@800x480p60 {
                refresh = <60>;
                xres = <800>;
               yres = <480>;
               /*in pico second，  0.000 000 000 001s*/
               pixclock = <37037>;
               left_margin = <50>;
               right_margin = <50>;
               upper_margin = <3>;
               lower_margin = <3>;
               hsync_len = <20>;
               vsync_len = <2>;
               /*0: FB_VMODE_NONINTERLACED，  1:FB_VMODE_INTERLACED*/
               vmode = <0>;
         };
}
```

Backlight dts configuration, kernel and u-boot dts need to be consistent:

```
backlight {
        /*1.pwm num; 2. period in ns; */
        /*3.plarity，  0: high active，  1: low active*/
        backlight_en_gpios = <&gpio 124 0>;
        pwms = <&pwm 0 100000 0>;      //*Backlight PWM. 0: PWM0. 100000: PWM period(unit:
ns), range: 0~100000.    0: high active*/
        total_steps = <1024>;
        min_brightness = <524>;
        max_brightness = <1024>;
        dft_brightness = <724>;
        delay_bf_pwm    = <200>;
        delay_af_pwm    = <10>;
};
```

(2) Add initial code

The initial code of the screen can be added in the kernel or in the u-boot stage. Since the initialization in the u-boot stage can make the CubieScreen display faster, the current display strategy is to put the initialization process in the u-boot stage.

The main modification file: SDK/u-boot/drivers/video/owl_lcd.c, the specific initial code can refer to the corresponding files in the directory.

Path: cb6-cubiescreen-patch /new

4.3.2 TP Configuration

(1) Add driver files and header files

As shown below, add ft5x_ts.c, ft5x_ts.h, ft_app.i, ctp.h four files in the specified directory.

```
├── kernel
│   ├── drivers
│   │   └── input
```

```
│   │       └── touchscreen
│   │           ├── ft5x_ts.c
│   │           ├── ft5x_ts.h
│   │           ├── ft_app.i
│   │           └── Makefile
│   └── include
│       └── linux
│           └── ctp.h
```

In the section 4.1 of this article, it has been learned that CubieScreen's TP uses I2C1, which is already configured in the “ctp_get_system_config” function in the ft5x_ts.c driver file.

```
static int ctp_get_system_config(void)
{
      //ctp_print_info(config_info，DEBUG_INIT);
       twi_id = 1;     //config_info.twi_id, use i2c1；
       screen_max_x = 800;    //config_info.screen_max_x;
       screen_max_y = 480;//config_info.screen_max_y;
       ......
}
```

(2) Configure the “Makefile” to compile the new driver. The “Makefile” is as above.

Add “obj-m += ft5x_ts.o” to the file so it can compile the driver directly into a module:

```
...
obj-$(CONFIG_TOUCHSCREEN_DETECT)                    += ctp_detect/
obj-m                                                    += ft5x_ts.o
```

(3) Configure the IDC file in the android

IDC (Input Device Configuration) is an input device configuration file that contains device specific configuration properties that affect the behavior of the input device. For TP devices, an IDC file is always needed to define its behavior. The path to the IDC file of the CubieScreen S500 is as follows:

SDK/android/device/actions/cubieboard6/ft5x_ts.idc

(4) Load the TP driver in the init.extra_modules.rc file of the android.
The path to init.extra_modules.rc is as follows:

SDK/android/device/actions/cubieboard6/config/root/init.extra_modules.rc

```
#tp
insmod /misc/modules/ft5x_ts.ko
```

5 Code Transplantation

5.1 Merge Patch Into The SDK

There are two ways to transplant the CubieScreen related code:
①patch it directly.
②Merge it based on the patch file manually.

Direct patching requires more rigor, and you need to confirm the status of the local code to ensure that there are no conflicts between local files and patches. If there has conflict, it may not be patched successfully, so you need to manually merge code according to the patch file.

5.1.1 Android SDK

The SDK mainly modifies these four directories: android, kernel, u-boot and owl/s500.
It is best to create a new branch to transplant the CubieScreen code.

Merge patches directly

1. First confirm the status of the SDK version and local code
(1) The SDK version must be v3.0-20170523 and use the “git log” command to check if the latest local commit is the same as the initial version. The latest local commit for the SDK is "00c45c8847ad59bfe23d170c84e151eeaf7e151a"

```
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0$ git log
commit 00c45c8847ad59bfe23d170c84e151eeaf7e151a
Author: Aaron <aaron@cubietech.com>
Date:   Tue May 23 16:58:21 2017 +0800

    add .gitignore and README.TXT

commit 4dbfacdb6d6657680f1f3022c6d95f263ba2c150
Author: Aaron <aaron@cubietech.com>
Date:   Tue May 23 16:40:44 2017 +0800

    TAG_170523_v3.0_owl_s500_cb6_android_5110
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0$
```

Figure 5-1

(2) Execute the “git status” command to check the status of the local code. If there has some files are modified, you need to manually restore it to prevent the locally modified file from conflicting with the file in the patch. View the xxx.patch file. If it is determined that the local modification and patch do not conflict, you can’t restore it.

2. Merge the patch

(1) Execute the “git apply” command to merge the patch

```
../SDK$ git apply cb6-cubiescreen.patch
```

(2) Execute “git status” command to check whether the patch is successfully merged.

The following red part of the file is the file that was modified and added after the patch was merged.

```
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0$ git status ./
# On branch cubieboard6_v3.0
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   android/device/actions/common/prebuilt/apk/thirdparty/sofabutler/app/shafaguanjia.apk
#       modified:   android/device/actions/cubieboard6/config/root/init.extra_modules.rc
#       modified:   android/device/actions/cubieboard6/device.mk
#       modified:   android/frameworks/base/packages/SystemUI/res/layout/navigation_bar.xml
#       modified:   kernel/.gitignore
#       modified:   kernel/arch/arm/boot/dts/cubieboard6_s500_android.dts
#       modified:   kernel/drivers/input/touchscreen/Makefile
#       modified:   owl/s500/boards/android/cubieboard6/misc/boot_logo.bmp.gz
#       modified:   u-boot/arch/arm/dts/cubieboard6_s500_android.dts
#       modified:   u-boot/drivers/video/owl_lcd.c
#       modified:   u-boot/include/owl_lcd.h
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       android/device/actions/cubieboard6/ft5x_ts.idc
#       cb6-cubiescreen-patch/
#       kernel/drivers/input/touchscreen/ft5x_ts.c
#       kernel/drivers/input/touchscreen/ft5x_ts.h
#       kernel/drivers/input/touchscreen/ft_app.i
#       kernel/include/linux/ctp.h
no changes added to commit (use "git add" and/or "git commit -a")
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0$
```

Figure 5-2

Merge patches manually

There are patch and new folders in “cb6-cubiescreen-patch” directory, you can manually add files based on these two files. Install the following tools to facilitate code transplantation:

① “meld” tool: Can contrast code file.

② “tree” tool: List the contents of the directory in a tree view.

Please refer to the sixth chapter of the article "Common Commands".

1. View the files that need to be transplanted

Enter the “cb6-cubiescreen-patch/new” directory. Execute the “ls” command to know that you need to modify the android, kernel, owl, and u-boot directories. Then execute the “tree” command to view the all modified files. Some files in the new directory of “cb6-cubiescreen-patch” are not find in local, indicating that the file is new and copy it to the the local directory of SDK.

```
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0/cb6-cubiescreen-patch/new$ ls
android  kernel  owl  u-boot
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0/cb6-cubiescreen-patch/new$ tree
.
├── android
│   ├── device
│   │   └── actions
│   │       ├── common
│   │       │   └── prebuilt
│   │       │       └── apk
│   │       │           └── thirdparty
│   │       │               └── sofabutler
│   │       │                   └── app
│   │       │                       └── shafaguanjia.apk
│   │       └── cubieboard6
│   │           ├── config
│   │           │   └── root
│   │           │       └── init.extra_modules.rc
│   │           ├── device.mk
│   │           └── ft5x_ts.idc
```

```
│   └── frameworks
│       └── base
│           └── packages
│               └── SystemUI
│                   └── res
│                       └── layout
│                           └── navigation_bar.xml
├── kernel
│   ├── arch
│   │   └── arm
│   │       └── boot
│   │           └── dts
│   │               └── cubieboard6_s500_android.dts
│   ├── drivers
│   │   └── input
│   │       └── touchscreen
│   │           ├── ft5x_ts.c
│   │           ├── ft5x_ts.h
│   │           ├── ft_app.i
│   │           └── Makefile
│   └── include
│       └── linux
│           └── ctp.h
├── owl
│   └── s500
│       └── boards
│           └── android
│               └── cubieboard6
│                   └── misc
│                       └── boot_logo.bmp.gz
└── u-boot
    ├── arch
    │   └── arm
    │       └── dts
    │           └── cubieboard6_s500_android.dts
```

```
├── drivers
│   └── video
│       └── owl_lcd.c
└── include
    └── owl_lcd.h
```

2. View the specific changes in the file of the patch

Enter the “cb6-cubiescreen-patch/patch” directory and execute the “vim xxx.patch” command to view the changes in the patch. There are two main types of files modified in patch: binary files (such as pictures and APK files) and normal files (viewable by vim).

①binary file

Files like images, APK, and audio are displayed as binary files in the patch. The modification content of APK in patch is as shown below.

```
diff --git a/android/device/actions/common/prebuilt/apk/thirdparty/sofabutler/app/shafaguanjia.apk
rdparty/sofabutler/app/shafaguanjia.apk
index 51adfe0f78097a6417b87c8eaf65bfc7d3bcbf15..58577f15c488cf8f97ff41f2ae0e80cffea12dca 100644
GIT binary patch
delta 4524115
zcmb@sbzBtT+b+KJ(jZEwbcYg>3IdXf^itB&-3tRK0@Bhgt#mg_cXzXkbc1vs{OWn%
z_xwKRbN>3x&dgqOJu&xO_dWOX%+A(_{zlpD|BgcXP5}k=F$fF>gBndg$~&rL2v1Xh
zK#kwxBZE+=fGRyqceTk`SUWk-2QX>1uXQFZ8NsaHoHc_Vfg~~V*DnO$2zt%<3>`i}
z3Z^Wp_N&whf99wcOzAn?FBVl*R#ijdAw^XhoK;izB@wsq<Hdu46t4}RfkZ=|;r`SY
```

Figure 5-3

② normal file

For example, like “device.mk”. In the figure below, "+" represents the addition of this sentence code, and "-" represents the deletion of this sentence code.

```
diff --git a/android/device/actions/cubieboard6/device.mk b/android/device/actions/cubieboard6/device.mk
index bee406b..0112a71 100755
--- a/android/device/actions/cubieboard6/device.mk
+++ b/android/device/actions/cubieboard6/device.mk
@@ -30,6 +30,7 @@ VERSION_DATE=$(shell date "+%y%m%d")
 PRODUCT_COPY_FILES := \
     device/actions/cubieboard6/kernel:kernel \
        device/actions/cubieboard6/ft5x06-touch.idc:system/usr/idc/ft5x06-touch.idc \
+       device/actions/cubieboard6/ft5x_ts.idc:system/usr/idc/ft5x_ts.idc \
        device/actions/cubieboard6/gl5203-adckey.kl:system/usr/keylayout/gl5203-adckey.kl \
        device/actions/cubieboard6/vold.fstab:system/etc/vold.fstab \
        device/actions/cubieboard6/apns-conf.xml:system/etc/apns-conf.xml \
@@ -429,7 +430,7 @@ ADDITIONAL_BUILD_PROPERTIES += \
        dalvik.vm.heaptargetutilization=0.75 \
        dalvik.vm.heapminfree=2m \
        dalvik.vm.heapmaxfree=8m \
-       ro.sf.lcd_density=160 \
+       ro.sf.lcd_density=240 \
```

Figure 5-4

3. Start manually merge the relevant code

Use the “tree” command in the “cb6-cubiescreen-patch/patch/new” directory to view the list and manually modify the file one by one. The following example is not a manual integration in the

order, but the three representative files are taken as example to merge the modificate content. Developers and enthusiasts should manually follow the order in order to avoid omission.

example:

The first file: “shafaguanjia.apk”, directly copy the apk file in the new directory to the corresponding directory of the SDK.

The second file: “init.extra_modules.rc”, uses the “meld” command.

After executing the command, as shown in the figure below, the left side is the local SDK file, and the right side is the file under “cb6-cubiescreen-patch/patch/new”. And merge the line code by clicking the arrow in the red box in the figure below directly. After merged, click on the blank space on the left side and click "Save" to save.

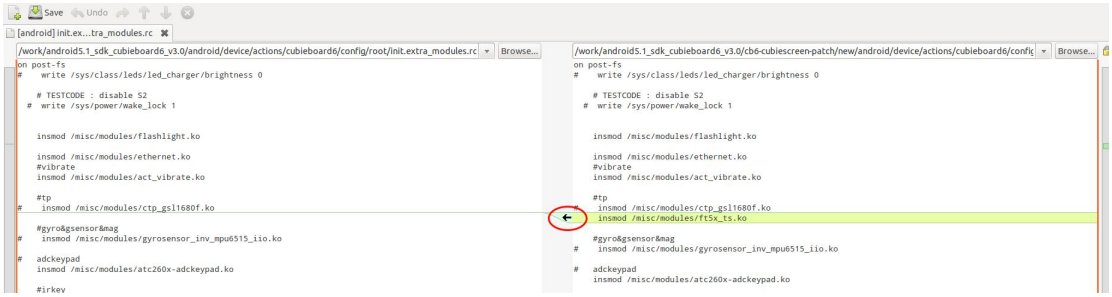

Figure 5-5

The third file: “ft5x_ts.c” Local SDK does not have this file, copy the file in the “new” directory to the corresponding directory of the local SDK directly.

4. Manual merge is completed

Until all the files in the “cb6-cubiescreen-patch/patch/new” directory have been manually modified, execute “git status” to check the changes. After confirming, you can compile the firmware to verify whether the manual transplantation is successful.

```
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0$ git status ./
# On branch cubieboard6_v3.0
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   android/device/actions/common/prebuilt/apk/thirdparty/sofabutler/app/shafaguanjia.apk
#       modified:   android/device/actions/cubieboard6/config/root/init.extra_modules.rc
#       modified:   android/device/actions/cubieboard6/device.mk
#       modified:   android/frameworks/base/packages/SystemUI/res/layout/navigation_bar.xml
#       modified:   kernel/.gitignore
#       modified:   kernel/arch/arm/boot/dts/cubieboard6_s500_android.dts
#       modified:   kernel/drivers/input/touchscreen/Makefile
#       modified:   owl/s500/boards/android/cubieboard6/misc/boot_logo.bmp.gz
#       modified:   u-boot/arch/arm/dts/cubieboard6_s500_android.dts
#       modified:   u-boot/drivers/video/owl_lcd.c
#       modified:   u-boot/include/owl_lcd.h
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       android/device/actions/cubieboard6/ft5x_ts.idc
#       cb6-cubiescreen-patch/
#       kernel/drivers/input/touchscreen/ft5x_ts.c
#       kernel/drivers/input/touchscreen/ft5x_ts.h
#       kernel/drivers/input/touchscreen/ft_app.i
#       kernel/include/linux/ctp.h
no changes added to commit (use "git add" and/or "git commit -a")
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0$
```

Figure 5-6

5.1.2 Linux SDK

Since the Linux SDK is v1.0, so use the “s500_linux_sdk_v1.1.patch” update to v1.1 first, and then update the CubieScreen code with “linux-cubiescreen.patch”.

Merge patches directly

1. First confirm the status of the SDK version and local code.

(1) The SDK version must be v1.0 and use the “git log” command to check if the latest local commit is the same as the initial version. The latest local commit for the SDK is "5553d98f5f03e1d7ea633d1c751a28db79f8b330"

```
ft@ft:~/work/test-SDK/s500-sdk$
ft@ft:~/work/test-SDK/s500-sdk$ git log
commit 5553d98f5f03e1d7ea633d1c751a28db79f8b330
Author: Darren Jiang <darren@cubietech.com>
Date:   Fri Feb 10 16:28:55 2017 +0800

    add sdk build README

commit a5c877f3ac8592e051849c5f88f2a422fcb023d4
Author: Darren Jiang <darren@cubietech.com>
Date:   Fri Feb 10 15:04:57 2017 +0800

    init s500 linux sdk
ft@ft:~/work/test-SDK/s500-sdk$
```

Figure 5-7

(2) Execute the “git status” command to check the status of the local code. If there has some files are modified, you need to manually restore it to prevent the locally modified file from conflicting

with the file in the patch. View the “xxx.patch file”. If it is determined that the local modification and patch do not conflict, you can’t restore it.

2. Merge the patch

(1) Execute the “git apply” command to merge the patch

```
../SDK$ git apply s500_linux_sdk_v1.1.patch
```

```
ft@ft:~/work/test-SDK/s500-sdk$ git apply s500_linux_sdk_v1.1.patch
s500_linux_sdk_v1.1.patch:89: space before tab in indent.
                rear_channel = <0>; /* 0: use channel_0, 1: use channel_1 */
s500_linux_sdk_v1.1.patch:118: trailing whitespace.

s500_linux_sdk_v1.1.patch:142: trailing whitespace.

s500_linux_sdk_v1.1.patch:163: trailing whitespace.

s500_linux_sdk_v1.1.patch:164: space before tab in indent.
                /*rear-reset-gpios = <&gpio 20 1>;*/ /*86v:GPIOA20*/
warning: 抑制下仍有 1035 个空白字符误用
warning: 1040 行有空白字符误用。
ft@ft:~/work/test-SDK/s500-sdk$
```

Figure 5-8

(2) Execute the “git commit” command, then use “git log” to view the commit information.

```
../SDK$ git add    owl/.config      kernel/drivers/input/touchscreen/ft5x_ts.c
../SDK$ git commit -m "update s500 linux SDK v1.1" -a


../SDK$ git log
```

```
ft@ft:~/work/test-SDK/s500-sdk$ git commit -m "update s500 linux SDK v1.1" -a
[cubieboard6 f5bbed5] update s500 linux SDK v1.1
 96 files changed, 3734 insertions(+), 825 deletions(-)
 create mode 100644 kernel/drivers/input/touchscreen/ft5x_ts.c
 create mode 100644 owl/.config
```

Figure 5-9

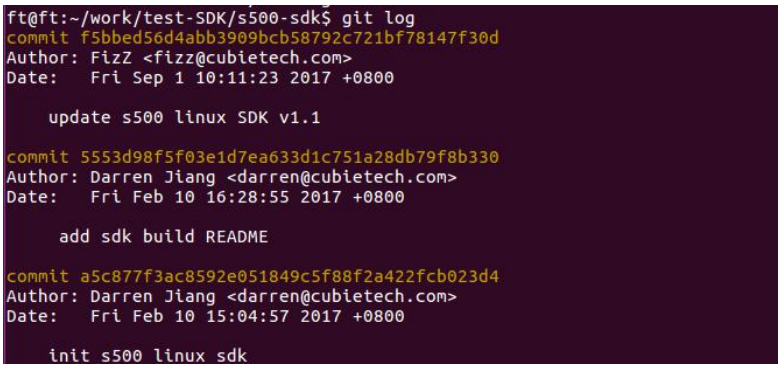

Figure 5-10

(3) Copy the “cubiescreen.patch” file to the SDK directory and execute the “git apply” command to merge the patch.

```
../SDK$ git apply cubiescreen.patch
```

(4) Execute the “git status” command to check whether the patch is successfully merged. The following red part files are the files that have been modified and added after the patch is merged.

Figure 5-11

Merge patches manually

There are patch and new folders in “linux-cubiescreen-patch”, add files based on these two files manually.

1. View the files that need to be transplanted

Enter the “linux-cubiescreen-patch/new” directory. Execute the “ls” command to know that you need to modify the kernel, owl, and u-boot directories. Then execute the “tree” command to view the all modified files. Some files in the new directory of “linux-cubiescreen-patch” are not find in local, indicating that the file is new and copy it to the the local directory of SDK.

```
ft@ft:~/桌面/cb6-cubiescreen/linux-cubiescreen-patch/new$ ls
```

```
kernel    owl    u-boot
ft@ft:~/桌面/cb6-cubiescreen/linux-cubiescreen-patch/new$ tree
├── kernel
│   ├── arch
│   │   └── arm
│   │       ├── boot
│   │       │   └── dts
│   │       │       └── cubieboard6_s500_linux_cubiescreen.dts
│   │       └── configs
│   │           └── cubieboard6_s500_linux_defconfig
│   └── drivers
│       ├── input
│       │   └── touchscreen
│       │       └── ft5x_ts.c
│       └── video
│           └── owl
│               └── displays
│                   └── lcdc
│                       └── lcdchw.c
├── owl
│   └── s500
│       └── boards
│           └── ubuntu
│               └── cubieboard6
│                   ├── config
│                   └── misc
│                       └── boot_logo.bmp.gz
└── u-boot
    ├── configs
    │   └── cubieboard6_s500_defconfig
    ├── drivers
    │   └── video
    │       └── owl_lcd.c
    └── include
        └── owl_lcd.h
```

Since the ".config" in the owl directory is a hidden file, it cannot be displayed using the "tree" command. You still need to modify the .config file in the owl directory when modify it.

3. Start manually merge the relevant code

Use the "tree" command in the "linux-cubiescreen-patch/new" directory to view the list and manually modify the file one by one. The following example is not a manual integration in the order, but the three representative files are taken as example to merge the content. Developers and enthusiasts should manually follow the order in order to avoid omission.

example:

The first file: "boot_logo.bmp.gz", directly copy the file in the new directory to the corresponding directory of the SDK.

The second file: "cubieboard6_s500_linux_defconfig", uses the "meld" command.

```
../SDK$ meld kernel/arch/arm/configs/cubieboard6_s500_linux_defconfig ~/桌面/new/kernel/arch/arm/configs/cubieboard6_s500_linux_defconfig
```

After executing the command, as shown in the figure below, the left side is the local SDK file, and the right side is the file under "linux-cubiescreen-patch/new". And then merge the line code by clicking the arrow in the red box in the figure below directly. After merged, click on the blank space on the left side and click "Save" to save.

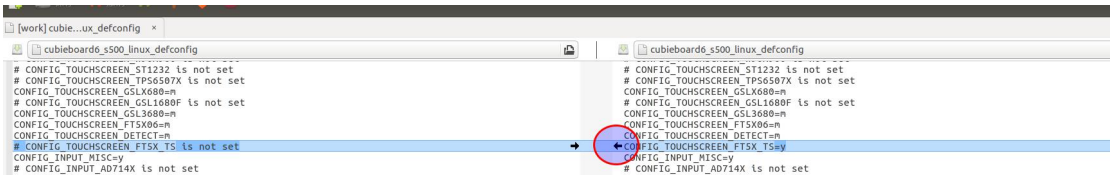

Figure 5-12

4. Manual merge is completed

Until all the files in the "linux-cubiescreen-patch/new" directory have been manually modified, execute "git status" to check the changes. After confirming, then compile the firmware to verify whether the manual transplantation is successful.

Figure 5-13

6 Verify After Transplantation

6.1 Android SDK compile

```
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0/owl$ ./config.sh
Select board type:
        1. cubieboard6

Which would you like? [cubieboard6] 1
s500 android cubieboard6 configured.
bill@bill:/work/android5.1_sdk_cubieboard6_v3.0/owl$ make    //compile firmware
```

Refer to *CubieBoard6-Android-Building-Guide*

6.2 Linux SDK compile

(1) Execute “./config.sh” command.

```
ft@ft:~/work/s500/owl$ ./config.sh
Select board type:
        ……
        3. cubieboard6
```

```
Which would you like? [bubble_gum] 3
s500 ubuntu cubieboard6 configured.
ft@ft:~/work/s500/owl$
```

(2) Execute “make” command

Refer to *CubieBoard6-Linux-SDK-Usage-Guide*

6.3 Device Verification

Connect the CubieScreen pin to the CubieBoard6 and plug it in with 5V DC power.

Figure 6-1 CubieScreen On CB6

7 Common Command

Common command:

① tree

Lists the files and folders of the current directory in a tree form, without any parameters, it automatically lists all depth level files and directories under the current directory. The installation method is as follows.

```
$ sudo apt-get install tree
```

② meld

“Meld” allows users to view changes between files and directories. Directory comparisons, project comparisons, and file content comparisons are possible. The installation method is as follows.

```
$ sudo apt-get install meld
```

③git apply

```
../SDK$ git apply xxx.patch
```

8 Typical Problem Analysis

1. As mentioned in Section 4.1 of this article, CubieScreen does not need to configure the backlight enable gpio port. By remove this configuration item directly, it will cause the log to print a "cut here" error each time it is power on. This error does not affect the normal operation of the system. To avoid this error, you can configure an empty gpio port.

2. Manually transplant the patch, you can open the “xxx.patch” file to view the modified content, and directly copy the code to the corresponding file of the local SDK. It is recommended to use the “meld” tool to compare the local SDK file with the “new” directory file of the patch, and merge the relevant modified to local file of
SDK.

3. After manually patching the patch, If display is abnormal. First, check the configuration of DTS. The LCD and backlight configurations in DTS use some gpio ports, so check if there is gpio or other functions multiplexing in the DTS file.

4. In the LCD configuration of DTS, the reset gpio and lcd gpio configuration values are all 0, but lcd gpio is active high, and reset gpio is active low, so it will be doubtful. In fact, you can't just look at the surface, you need to check the relevant code in the driver. According to the following DTS configuration, reset gpio is set “0” to enable in the driver, so it is active low, and power gpio is set “1” to enable, so it is active high.

```
lcd_power_gpios = <&gpio 130 0>;      /*GPIOE(2)    0 active high*/
lcd_reset_gpios = <&gpio 131 0>;      /*GPIOE(3)    0 active low*/
```

5．After booting the patch, the CubieScreen can be displayed normally and the TP cannot be used successfully. First, enter the serial port debugging to confirm whether the tp driver is loaded.

```
root@AD500A:/ # lsmod
…...
ft5x_ts 37030 0 - Live 0x00000000
…....
```

If ft5x_ts is not found after executing “lsmod”, it means that it is not loaded. You can load the driver by executing the following command manually.

```
root@AD500A:/ # insmod misc/modules/ft5x_ts.ko
```

If the above command is executed, it is find that the driver is not loaded successfully. It may be a problem when transplant the TP driver. There is a problem with the compiled ko file. Please confirm whether the driver code has errors then recompile.

9 Reference Documents

CubieBoard6-Android-Building-Guide
CubieBoard6-Installation-Guide
CubieBoard6-20161223-SCH
CubieBoard6-Linux-SDK-Usage-Guide
A035VL01 V0_SPEC (CubieScreen datasheet)
cubiescreenV1.1_sch

10 History

Date	Version	Annotation	Author
2018.08.14	1.0	Built the first version of CubieScreen S500 transplant guide	bill