

Development of Video Driver System based on Linux Operation System

Yue Pan¹, Desheng Li^{2,*}

¹Bengbu Automobile Non-commissioned Officer Academy, Bengbu, Anhui, China

²Anhui Science and Technology University, Chuzhou, Anhui, China

*Corresponding author

E-mail: ¹940220338@qq.com, ²ldsyy2006@126.com

Abstract

This article mainly introduces the design of the video system driver architecture based on LINUX3.10 kernel. With the development of the times, the video users are increasing year by year, the user is not satisfied with the use of computer for video communication. The development of this system is based on embedded technology as the foundation, will the traditional computer video software is transplanted to the embedded system, from the bulky computers, video software can be used in security monitoring, can also be used for mobile video communication. The design of video system mainly takes the LINUX operating system as the core, and the periphery increases the camera and the display module. In this paper, the structure of the system is clear, and the implementation of the system is discussed in the application layer and the KERNEL layer. The design and development of the system are discussed in the MAKEFILE, SHELL and DRIVER files. The result of the system is stable, and it has certain practical value.

Key words: video drivers; LINUX; peripheral; frame structure.

1. System overview

1.1. Development background

Human beings invented the first electronic computer in the world at a famous university in the United States in 1946. Today, the computer already has a rapid development. The first computer ENIAC from weight 30 tons of development to today's portable laptop, not more than 5 kg in weight. From the 5000 times per second operation to today's computing speed one hundred trillion times per second. The great changes had happened in these short 70 years. What can be concluded from the process of development is that the computer operation speed is slow by gradually upgrade of the rapid computing and large volume by gradually upgraded to portable, costs are high by gradually reduced. With the development of computer and increasingly widely application fields, single-chip microcomputer replace the computer in many fields gradually, because the single chip microcomputer has its unique advantages. Compared with the computer, SCM is small in size, and the manufacturing cost is a lot lower than the computer. Because of these features, SCM gradually entered people's sight, the use of embedded technology field is becoming more and more widely.

The LINUX operating system as the core of the design and development of the Video drive system mainly, adding the IMG, LCM, TP module in the peripheral, at the application layer and the KERNEL layer respectively discusses the software realization principle in system structure. The MAKEFILE,

SHELL, DRIVER files respectively discusses software implementation principle in the system driver file. Based on the embedded technology, designing and developing the video driver software system.

Currently many fields applied to this system , such as: the monitoring equipment in the store, intelligent TV at home, camera used to take pictures , etc., video software is everywhere in life.

1.2. Development goals

Through the development of video software, the fundamental purpose is servicing to the people better in life by using the system, providing users with a good video software development environment. What's more, I can go on continuous learning in the process of development, preparing to build a more secure, stable video software. This system is based on the LINUX kernel, using the LINUX environment to compile the program. The system mainly around the following objectives from development to end:

(1)The target of hardware aspects: in-depth analyzing and understanding the video tools in-depth.

(2)The target of software aspects: understanding android, Linux system architecture and the function of the camera software grossly.

(3)The target of designing driver aspects: analyzing the video driver software in-depth, developing a set of video tools that can work independently.

The video driver software, including two parts of the input and the output. The input part consists of the signal output and the control input mainly, because it is a video software, here this paper choose camera as the input device. When selecting a control input, there are several options: matrix keyboard input control, and touch screen input control .regarding to the first input, using the reverse scan method can realize the function of the input easily, but in Linux operating system, a mount of peripheral devices usually have IIC serial number, connecting the signal lines directly to the outside of the single chip microcomputer on the pin could bring the following disadvantages:

Wasting of the system resources, reduce the system stability and structural.

The keyboard control rarely appears in the market now, the design is not practical.

So here in this paper, choose a touch screen input as control input. The following three parts discussed the development and design of the three components.

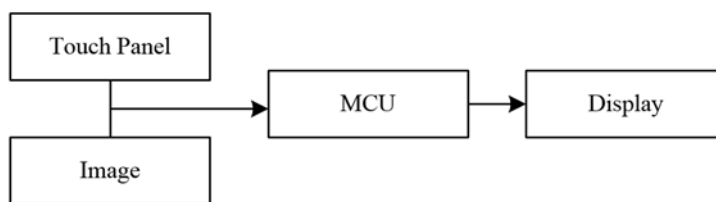


Fig.1. System structure

1.3. Development prospects

With the increase of the number of android users, the type of android software is increasing, the video software is varied on the market, such as: the original camera software provided by the android system, Baidu camera developed by Baidu Company, etc. With the development of the android, the type of the video application software is increasing, developing the video applications again seems to be some "extra" under such a large resource development, so this paper will focus on the development of the video driver software.

The development prospects of the video software can be summed up to the following:

To prepare for the development of new system: with the increase of users needs, the video system can appear in various fields, such as: market monitoring, the camera on a new cell phone, these all need the video driver software system. Only some simple transplant, even application of this system directly can achieve various functions on the basis of this paper. So the system offers help for the development of new system of driver software.

To popular LINUX operating system structure: with the development of the embedded technology, the LINUX system will be understood by the masses of users gradually. This paper based on the LINUX system as the core, to develop on the LINUX system structure, discussing the development process of the video software from the system files and the system structure two aspects. Through the design of this paper, the LINUX operating system can be spread and making it acceptable to more users.

2. Development premise

2.1. Hardware conditions

Any software development needs hardware support. Video-software development also need to have a certain hardware conditions. Summarized as follows:

(1) Computer Configuration: PC configuration software generates a direct impact on the formation rate of the software. Based on the configuration used in the development process are: Core i7 CPU, 16 GB of memory, 1T drive.

(2) Power supply: adjustable power is supply for equipment.

(3) LCM liquid crystal screen: monomer for output image data information. IC model of the two screens monomer respectively is: ILI9806, ILI5647.

(4) TP touch screen monomer: used to input control command.

(5) The core board and camera: used to collect image information.

2.2. Software prerequisites

2.2.1. UBUNTU 12.04 operating system

More common in the market now is a Windows system, it is composed of the United States in 1985, Microsoft released. Windows has its own unique advantages, such as graphical interface particularly clear and intuitive. And under Windows can be compatible with most software, at present most of the hardware manufacturers support the operating system. Because of Windows system is not open source, plenty of compile software under Windows not development, so this paper choose the Linux operating system in the video software development, UBUNTU 12.04 is a based on Linux system, and open source operating system. In UBUNTU 12.04 can easily find the needed to compile software and tools.

UBUNTU 12.04 operating system and Windows memory management in a different way, Windows is the operating system is written in memory, and UBUNTU is the operating system is mounted on the memory. UBUNTU operating system, mainly using shell commands to control the development environment. Development environment is often used to ultraedit, nfs, samba and other tools. One of the most important is to compile tools GCC, GDB debugging tools, project management tools MAKEFILE.

Before developing video software, Operating system to install UBUNTU 12.04, you can download the open source LINUX On the website. Enter the UBUNTU environment configuration after installation is complete.

2.2.2. FLASH TOOLS software

Remember in the process of learning stm32 and 51-MCU development, often using KEIL software to realize the compilation and burning. In this video software development process, main is compiled using UBUNTU 12.04, FLASH TOOLS to burning.

FLASH burning principle of software TOOLS can be roughly summed up as the following steps. Writing FLASH TOOLS software works can be roughly summarized as follows:

- (1) After the device is powered, UBUNTU memory BOOTROM run.
- (2) The BOOTROM initialization software stack.
- (3) The BOOTROM through UART/USB to shake hands with FLASH TOOLS software.
- (4) The BOOTROM through UART tools loading preloader. Not directly downloaded to the micro controller because single-chip DRAM memory part has not been initialized.
- (5) The BOOTROM jump to preloader and began to perform.
- (6) Preloaded will initialize the MCU DRAM memory part, and implement communication through USB FLASH TOOLS and single chip microcomputer.
- (7) FLASH TOOLS remaining download image files such as '.bin' file into the NAND FLASH.

When using this tool, I have met many problems. When using the tool, it may appear permission limit. When this problem occurs need to use the command sudo chmod, such as sudo chmod <access > <Target file>.Change FLASH TOOLS software permissions, ensure that these files can be used without the permission of the root.

Video driver software environment tools here has nearly installation is complete. To this video driver software environment tools have been set up to complete. Then introduced Linux operating system based on the KERNEL 3.02 in this paper.

3. The construction of the software compilation environment

3.1. The environment of UBUNTU system configuration

The first need to do is to upgrade the management system, Use command apt-get update. Apt-get is a command if UBUNTU, used to manage all kinds of software in the system.

After the completion of the Installation operating system, need to install open-jdk software. The way of install the software is Enter apt-get-install-openjdk-7-jdk commands in the terminal. Its is released by the Sun and used to Java platform source compile tools. After the installation of its need to solve the problem in the UBUNTU system dependent. In the UBUNTU operating system in the operation of the software, most software need to rely on the library files in the system, as we write the first hello world depends on the #include<stdio.h>, need to install some library files: apt - get the install git - core GNRPG flax.

In the built environment, met a lot of problem when I install the software, UBUNTU operating system installed on compile software often there will be a lot of dependence, you can use: sudo apt - get <-f> <install>, can quickly solve the problem of all dependent. In Linux system development mount command, is often used when there are many ways to mount the more commonly used is:

```
sudo mount. <file format>//purpose/mount //mount point -o user=<user name>,pass=<password>.
```

3.2. Introduced of Linux system framework

3.2.1. Linux operating system directory tree

The Linux operating system different from the Linux in computer. The LIUNIX operating system in

the computer is proposed to compile a micro operation system, just like a truck with a pocket car contrast. The following article describes the LIUNIX operating system directory tree.

Linux operating system is a multitasking, multithreaded operating system. The company and published in 1991, Linux root directory trees include the following: frameworks, kernel, packages, system, vendor, etc. In this huge with file directory, the most important thing is the kernel folder, here is the kernel version 3.02, folder mainly include the Linux operating system kernel file, in this paper the video driver software development, the driver files are mostly in the kernel directory.

The rest of the folder, such as frameworks folder is the company provides Linux framework, in this article the video driver in the development process of don't need too much change.

3.2.2. Linux operating system structure tree

Discuss the structure of the LIUNIX operating system, mainly including the upper software, the middle driver and the lower level hardware. In fact, if the Linux operating system structure continues to subdivide, from top to bottom can be divided into: application layer, HAL, KERNEL layer, LK, PERLOADER layer and hardware layer. One layer of software, such as our common Java language, is the development of software layer common language. But in the process of software development, often use the lower provide interface, the interface is provided to the upper layers HAL, HAL based on KERNEL layer and application layer, and between the general is mainly written in c ++ language.

PERLOADER is an excessive layer. PERLOADER and BIOS have common ground. BIOS is used to load the operating system of the computer. PERLOADER is used to load the first layer of the LIUNIX operating system.

In this video compiled software development process, met a lot of problems. Video software relates to touch screen, camera, and display. Display and touch screen appears mainly in the KERNEL layer and LK, but cameras use mainly in the KERNEL layer and HAL, so when compile camera module driver needs the overall compilation, compile the KERNEL only camera will not activate. The concrete realization process is discussed in the following paper.

3.3. The MAKEFILE and SHELL script

3.3.1. MAKEFILE

To discuss MAKEFILE files, common is GCC -g -o main.o, make clean. There are many types of MAKEFILE files. If each file from the project as a drop of water, so the MAKEFILE file is a container that will be in all kinds of files will be pooled together, completely on the MAKEFILE to organize connection. MAKEFILE files will be linked to a variety of files together. Completely by MAKEFILE to organize the connection. MAKEFILE also has its own grammar, mainly including:

- (1) Specific rules
- (2) Implicit rules
- (3) Define variables
- (4) Instruction
- (5) The comment lines

This article does not discuss here the MAKEFILE detailed rules of grammar in a short period of MAKEFILE of video software engineering as an example:

```
CUSTOM_HAL_IMGSSENSOR = gc5024_mipi_raw  
LCM_HEIGHT = 854
```

```
LCM_WIDTH = 480
```

These three lines of code is main make document in the configure file, through these three lines definition can tell the system that Hal camera file is gc5024 and screen resolution is 854 * 480. A MAKEFILE improve the compatibility and portability of the system on one hand, on the other hand, greatly improve the system logic. To discuss here, maybe many people will have such doubt, tell the system camera is GC5024, How to find GC5024 drive file in system, How to compile and add the file to the system? A SHELL script file is required to tell the system at this time, where is the file.

3.3.2. SHELL script

Script file is a special kind of descriptive file, referred to as the macro or batch file. The definition and usage in this article, no longer discuss specific script file, with video in software engineering, for example a small script file:

```
PARAMETER_LEN=$#
if [[ $PARAMETER_LEN -ne 2 ]]; then
echo  project name is not set. !! ERROR , exit
exit
fi
```

SHELL script is mainly used to detect the number of parameters.

“\$#” is the number of parameters and the number assigned to the PARAMETER_LEN variable. The function of echo is the same as the fprintf function in the C grammar. Exit function exit, the same as the return function in the C language.

To discuss here, did not solve the problems left by the previous section, how to tell the location of the system files. We can see this line of code:

```
MAKEFILE_NAME=bird/make/$1/$1.mak/
make/
```

In the bird/make / < parameter > / directory, a corresponding parameter assignment for MAKEFILE_NAME. Then run make, namely began to compile the files in the directory. So that you can achieve the purpose of the script and MAKEFILE common management engineering.

The construction of LIUNIX operating system and environment is discussed. In this paper, the development and implementation process of video driver software is discussed.

4. LCM development

LCM means to display the LCD module. LCD means liquid crystal display Common liquid crystal display structure is composed of glass and liquid crystal box. Put two pieces of parallel glass in in a liquid crystal box, here is not the same as the two pieces of parallel glass plate configuration. In the allocation of the substrate under the LCD TFT (thin film transistor). Piece on the liquid crystal display color filter device substrate configuration, and on the base board to join the signal and the voltage of a given size, degree of filter is used to control the first fast substrate, to show different color, this is the simple principle of LCD display.

Followed by a brief introduction of LCM, it is the difference between a and LCD, LCM is about LCD control device and peripheral circuit, such as on a module. Usually called the module LCM.

In this paper, in the design of video software, LCM module from the market to buy a piece of ordinary screen, from another perspective, also reflected the popularity of video software. Here are the parameters

of the LCM table (provided by the manufacturer):

Table 1. LCM configuration table

Parameter	Data
Device name	LCM
Supplier name	TONGXINGDA
Module	TXDTPDKP-T.B.D
IC model	ILI9806E
Size	5.0
Resolution	FWVGA(480*854)
ID pin	Low level
Glass type	BOE

To get the parameters need to do is after analysis, before write driver software is important to note several parameters of the device: mainly include: IC models: ILI9806E_2, resolution: FWVGA (480 * 854) and glass models: BOE. According to the above information, the LCM driver file naming rules: < IC type>_<glass type>_<resolution size>.So the LCM driver file named "ILI9806E_BOE_FWVGA".

4.1. LCM circuit diagram analysis

LCM pins connect diagram is as follows, can learn in the following picture LCM module pin.

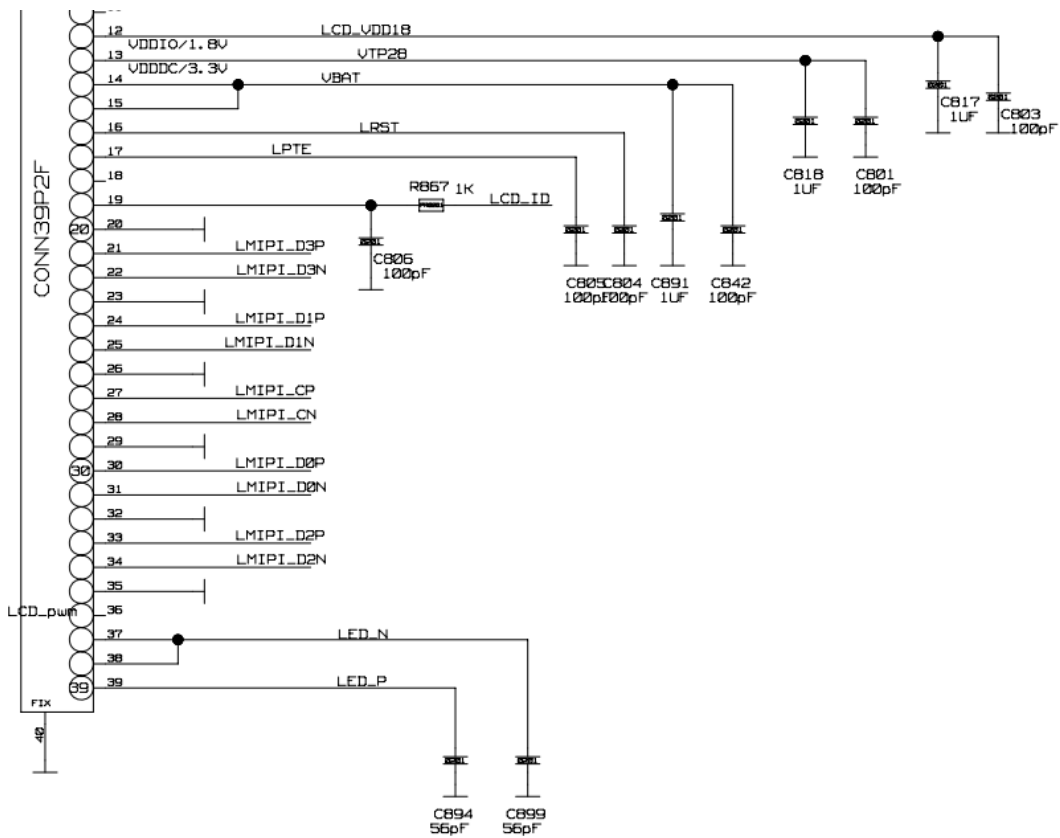


Fig.2. LCM interface figure

From the manufacturer to provide the circuit diagram of can get the following information:

(1) The LCM data transmission, a total of 10 pins, respectively is: LMIPI_D0P, LMIPI_D0N, LMIPI_D1P, LMIPI_D1N, LMIPI_D2P, LMIPI_D2N, LMIPI_D3P, LMIPI_D3N, LMIPI_CP, LMIPI_CN.

(2) TP28 is the power supply pin of LCM.

(3) LRST is LCM reset pin, used to restart the LCM.

(4) LCM_ID can use this pin to read LCM ID model.

4.2. LCM driver writing

4.2.1. LCM file structure

Has been discussed above, a complete project, such as the video software development projects, the project each branch mainly by MAKEFILE connected with the SHELL file. So LCM driver file structure, the first part is the main-MAKE configuration file. Open/bird/make/<proj>/directory, Lord make files in this directory to this, the first thing you need to do is to configure the LCM driver files, tell system: using this screen:

```
CONFIG_CUSTOM_KERNEL_LCM="ILI9806E_BOE_FWVGA"
```

Then need to configure parameters of LCM, including the resolution of the LCM. As well as configuration name, need to add the corresponding parameter to main MAKE file information, the code is as follows:

```
CONFIG_LCM_HEIGHT="854"
```

```
CONFIG_LCM_WIDTH="480"
```

Tell LCM system parameters and their name, must be compiled, make an inappropriate example: as we are looking for a person, just know that the person's name, height, weight is certainly can't find, also need to know the address of the person. Similarly, in the Linux operating system, the MAKEFILE as told us the address, so this article are defined as follows MAKEFILE, tell where system driver file, where to go to compile:

```
include $(srctree)/drivers/misc/mediatek/Makefile.custom
```

```
obj-y += mt65xx_lcm_list.o
```

The above two lines of code is to tell the location of the system needs to be compiled file, \$(srctree) means in the current directory tree, for example, I put the project on Linux/home/root/directory of the operating system, the system will go/home/root/drivers/misc/mediatek/Makefile. To find a Makefile in the custom. Here it is important to note that two different MAKEFILE. The first MAKEFILE to tell system to/home/root/drivers/misc/mediatek/MAKEFILE. Custom directory to find driver files, the second MAKEFILE tell the system which compiled files.

The second line of code obj - y + = mt65xx_lcm_list. O, is to make the system will be mt65xx_lcm_list. C file compiled into mt65xx_lcm_list. O file.

Then this paper discusses mt65xx_lcm_list. C document writing. Mt65xx_lcm_list. C file need to do is to tell the system, to build LCM driver files have been defined.

So this article written like this:

```
#if defined(ILI9806E_BOE_FWVGA)
```

```
&ili9806e_boe_fwvga_lcm_drv,
```

```
#endif
```

If ILI9806E_BOE_FWVGA this macro, defined in the main MAKE will ili9806e_boe_fwvga_drv

join to join in the system, to provide interfaces for the upper, so in mt65xx_lcm_list. C file needs to be defined in the external function:

```
extern LCM_DRIVER ILI9806E_BOE_FWVGA_lcm_drv;
```

Here the LCM file structure has completed the setup, the next step to discuss LCM driver documentation.

4.2.2. LCM driver documentation

LCM driver file with the hello world. C file, you first need to include the corresponding header file: string.h, mt_gpio.h, Linux/string.h, lcm_drv. h.

In the header file, there are common: string. H including the C library, common characters in mt_gpio. H including main board GPIO configuration library, Linux/string. H including the common characters of LinuxC library Asm/arch/mt_gpio. H including motherboard information file, these. H file belongs to the Linux open source information.

For ILI9806E_2_BOE_FWVGA this screen, including screen driver commands, parameters, and registers, it can be defined as a structure LCM_setting_table, the parameters in the structure of the screen:

```
struct LCM_setting_table {
    unsigned char cmd;
    unsigned char count;
    unsigned char para_list[64];
};
```

After each screen factory, merchants can provide the parameters of the initial screen, written after the register is used to activate the LCM, this paper defined array LCM_setting_table_V3lcm_initialization_setting_V3 [], used in the initialization parameter. After initialization parameters to an array, through function push_table function, the parameter to LCM.

It is important to note that the parameters are generally more, while in parameter in order to prevent memory leaks cause system crash, need to join a detection of LCM memory usage function:

```
static void set_LCM(const u8 *temp){
    memcpy((void*)&lcm_util, (void*)util, sizeof(LCM_UTIL_FUNCS));
}
```

This function is used to calculate parameters of memory, ensure system to normal operation. The initialization LCM screen, through the function: lcm_get_params. Basic initialization of LCM driver here is done.

But in the LINUX operating system, drivers need to upper application, namely the HAL layer provides interfaces, so then need to write LCM initialization function:

```
init_lcm_registers
init_lcm_registers_sleep
init_lcm_registers_sleep_cmd
```

Their role is respectively: initialization register of LCM, in sleep status in LCM and LCM sleep wake up, discuss the realization of the function of the three methods below:

```
init_lcm_registers
```

By calling the dsi_set_cmdq function, will reflect data_array write array data to LCM, written after register will automatically LCM related functions, such as initializing the device.

```
init_lcm_registers_sleep
```

```
data_array[0] = 0x00280500;
dsi_set_cmdq(data_array, 1, 1);
```

LCM was achieved by the statement of sleep, 0 x00280500 is LCM standby Command, the Command to the Display Off, tell Command to register by dsi_set_cmdq function again, LCM detects the corresponding register is written to the Display Off Command, realize the LCM dormancy.

```
reset_lcd_module
```

Through init_lcm_registers_sleep dormancy LCM, how to speak, the old LCM sensei, is followed by a need to consider, reset_lcd_module introduced here to achieve the function of LCM awoken, mainly through the LCM GPIO pin connected to the chip to realize the control of LCM wake up.

Mainly through set_gpio_mode this function to achieve function, into several parameters: RES, can ENABLE the feet. After can use set_gpio_out function to achieve the corresponding pin is pulled down. Set_gpio_mode, as well as the STM32 chip first need to initialize the GPIO pin function, the function of this function is: will the LCM RST pin set as output pins, and open outlet, ready to the GPIO up or grounded.

It is important to note here GPIO_OUT_ONE or GPIO_OUT_ZERO questions, contact factory before parameters:

Device name: LCM, supplier name: with xingda, module type: TXDTPDKP - T.B.D, IC models: ILI9806E_2, size: 5.0, resolution: FWVGA (480 * 854) ID foot low, module interface: no, glass models: to indicate.

Through the above information can be known: ID enable pin for low level, So here we use the second code: set_lcm_gpio (GPIO_LCM9806_RST, 0); By using this method to realize the wake of LCM.

4.3. LCM_Debug

In video software development process, will meet many problems. The most intuitionistic performance is the point of the screen is not bright, at this time of processing method, first look at LCM backlight is on, if the backlight doesn't open, you can check the monomer problem, LCM monomer in ensuring no problem after checking driver file with the LCM monomer MAKEFILE configuration file.

If parameters are not configured in the MAKEFILE, LCM may occur in black screen area, basically check the parameters of LCM resolution is the same with the MAKE parameters in the configuration file.

5. TP development

Touch screen is very common in the market, such as the office hall, industrial control, game, mobile phone, the MCU development can see the shadow of the touch screen. Touch screen has gradually become a kind of can be accepted by users using a broader input devices. Touch screen to give now popular "multimedia office" in new ways and methods, users only need to move your fingers, you'll be able to control. Touch screen has the characteristics of simple, human nature is gradually accepted by the masses of users.

Touch screen has the following characteristics, determines its characteristics of widely used.

(1) Transparency: transparent this feature determines the TP and LCM collocation can be used, by LCM intuitive output and TP simple input to achieve control.

(2) Coordinates: TP as well as the mouse coordinates this feature, users will need the following points on the TP, and TP can be read automatically.

There are many kinds of touch screen on the market: transparent touch screen, acoustic wave touch

screen, resistive screen with capacitive screen, etc.

Video driver software, this paper adopts the TP is one of the capacitance screen, around the perimeter of the capacitive touch voltage, the tiny and coated with a transparent conductive layer, so the capacitance screen surface has a small current flow, when touching the screen with your fingers, originally in the screen surface flow to small current collection from the four corners of the screen to another pole, and then through the calculation of small current flowing from around the location of the finger touching the screen touch the relative coordinates, and finally through the IIC communication coordinate information into the single chip microcomputer.

In this paper, in the design of video software, TP module from the market to buy a piece of ordinary screen, from another perspective, also reflected the popularity of video software.

Here are the parameters of the TP (provided by the manufacturer):

Table 2. TP configuration table

Parameter	Data
Device name	TP
Supplier name	DAWOSI
Module	V62-ISP
IC model	MSG2238
Size	5.0
Resolution	FWVGA(480*854)
ID pin	No level
Glass type	None

To get the parameters need to do is after analysis, before write driver software is important to note several parameters of the device: mainly include: IC models: MSG2238, resolution: FWVGA (480 * 854) and ID feet: no.

According to the above information, the TP driver file naming rules: <IC type>_<resolution size>.

So the TP driver file named "MSG2238_FWVGA".

5.1. TP circuit diagram analysis

TP pins connect diagram is as follows, can learn in the following picture LCM module pin.

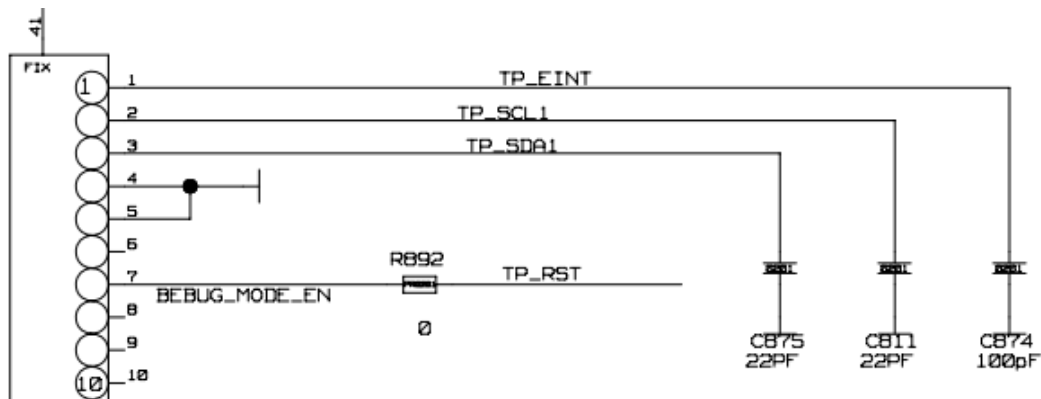


Fig.3. TP pin figure

From the manufacturer to provide the circuit diagram of can get the following information:

- (1) The TP data transmission no single pin.
- (2) With SCL, SDA pin: shows that TP using IIC protocol communication, using IIC protocol implementation data interaction.
- (3) TP_RINT pin: TP can generate the interrupt signal, using interrupt way remind system signal input.
- (4) TP_RST pin: TP reset pin user TP to restart.

5.2. TP driver writing

5.2.1. TP file structure

Has been discussed above, a complete project, such as the video software development projects, the project each branch mainly by MAKEFILE connected with the SHELL file. So TP driver file structure, the first part is the main MAKE.

Open/bird/make/<proj> / directory, Lord make files in this directory to this, the first thing you need to do is to configure the TP driver files, and tell the system: "using the touch screen":

```
CONFIG_TOUCHSCREEN_MSG2238=y
```

Then need to configure parameters of LCM, including the resolution of the TP.

As well as configuration name, need to add the corresponding parameter to main MAKE file information, the code is as follows:

```
CONFIG_TP_HEIGHT="854"
```

```
CONFIG_TP_WIDTH="480"
```

Tell the system the parameters of the TP and its name, it must be compiled but, make an inappropriate example: as we are looking for a person, just know that the person's name, height, weight is certainly can't find, also need to know the address of the person. Similarly, in the Linux operating system, the MAKEFILE as told us the address, so this article are defined as follows MAKEFILE, tell where system driver file, where to go to compile:

```
include $(srctree)/drivers/misc/mediatek/Makefile.custom
```

```
obj-y += tpd_setting.o
```

The above two lines of code is to tell the location of the system needs to be compiled file, \$(srctree) means in the current directory tree, for example, I put the project on Linux/home/root/directory of the operating system, the system will go/home/root/drivers/misc/mediatek/Makefile. To find a Makefile in the custom.

Here it is important to note that two MAKEFILE role is different. Make an inappropriate example: like we go out to travel, from home to the door needs a guide, to then need another guide. Truth is the same, the first MAKEFILE to tell system to/home/root/drivers/misc/mediatek/MAKEFILE. Custom directory to find driver files, the second MAKEFILE tell the system which compiled files. The second line of code obj - y + = tpd_setting. O, is to make the system will be tpd_setting. C file compiled into tpd_setting. O file.

Then this paper discusses tpd_setting. C document writing Tpd_setting. C file need to do is to tell the system, to compile the already defined TP driver file. So this article written like this:

```
ifeq ($(CONFIG_TOUCHSCREEN_MSG2238),y)
```

```
obj-y += MSG2238/
```

```
endif
```

If the Lord MAKE CONFIG_TOUCHSCREEN_MSG2238 this macro, defined will MSG2238 to join in the system, to provide the upper interface. Here TP file structure has completed the setup, the next step to discuss TP driver documentation.

5.2.2. TP driver documentation

TP driver file with the hello world. C file, you first need to include the corresponding header file: `cust_gpio_usage.h`, `Linux/sensors_io.H`. Including `cust_gpio_usage.H` including main board GPIO configuration library, `Linux/sensors_io.H` common peripheral device interface library, including Linux system. H file belongs to the Linux open source information.

For MSG2238 the touch screen, the use of IIC interface, MSG2238 SDA and SCL signal lines carry two Linux IIC bus to realize data exchange, but in the Linux operating system, different from those of single chip microcomputer communication. LINUX design process management, before implement IIC communication need IIC interface to the system application, "approved" before they can wait for system data transmission, this paper discusses application of IIC interface below.

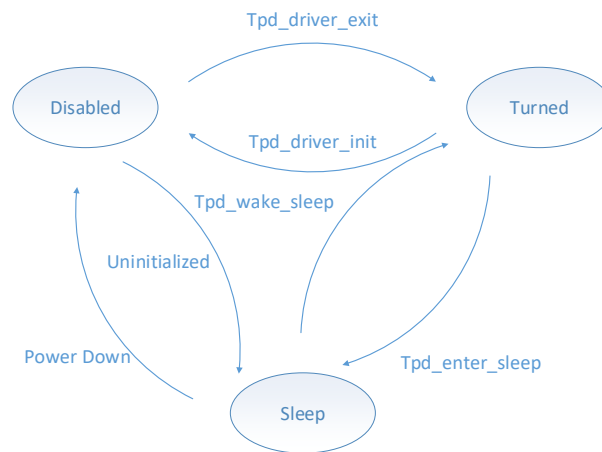


Fig.4. TP state transition diagram

Linux kernel from 2.6 after introducing `pd_device` platform equipment models, in the relevant registration information, have a registered IIC interface function: `iic_register_board_info` (`hw_iic_num`, & `iic_<SENSOR>`, 1), including the `Iic_num` information, then use the following definitions in this TP_Sensor MSG2238 method:

```

static struct i2c_driver tpd_i2c_driver = {
    .probe = tpd_i2c_probe,
    .remove = tpd_i2c_remove,
    .detect = tpd_i2c_detect,
    .driver.name = "MSG2238",
    .id_table = tpd_i2c_id,
    .address_list = (const unsigned short *) forces,
};
  
```

This structure includes TP IIC registration information, including: driver. The name is the device information, `Address_list` is the IIC bus system device list, define the structure, the IIC interface of TP are defined. The next need to do is write IIC protocol communication code. In order not to destroy the Linux system structural, here in this paper, the IIC communication protocol code is written on the outside, in

MSG2238 driver file, define functions: `i2c_read_bytes` `i2c_write_bytes`. The communication can be called directly.

Here this article focuses on the touch screen is how to identify and record the coordinates of the touch location information, first of all be clear, the touch screen itself no coordinates, more do not know what users touch is a place, so, at the time of system initialization, need in advance to tell touch screen, set up a simulation of the coordinate system. Put the coordinate system in TP driven. H file, the code is as follows:

```
#if defined(MSG2238_FWVGA)
#define TOUCH_SCREEN_X_MAX   (540)
#define TOUCH_SCREEN_Y_MAX   (960)
```

Acer and `TOUCH_SCREEN_X_MAX` `TOUCH_SCREEN_Y_MAX` here is to tell the resolution of the system in this paper, using the touch screen size, the back of the parameters is derived from the touch screen when buying manufacturer.

Then we discuss the implementation process of TP driver, before using the touch screen, touch screen need to describe status: closed, open, and sleep. First of all, this article to discuss that the realization of the three states have mutual conversion method.

Closed to open position: Touch screen in the case of no electricity nature is closed, the system boot, through `tpd_driver_init` function, initialization of touch screen, let the touch screen from closed to open, and is to realize the function of the jump function code below:

```
static int tpd_driver_init(void){
    GTP_INFO("touch panel driver init\n");
    i2c_regist_bard_infor(I2C_BUS_NUM, &i2c_tpd, 1);
    i2c_regist_bard_infor(0, &i2c_tpd, 1);
    if (tp_dri_add(&tp_dev) < 0)
        GTP_INFO("add failed\n");
    return 0;
}
```

First line `GTP_INFO` function is a debugging tool, when the system ran `tpd_driver_init` code print on the terminal "touch panel driver init", so it can guarantee the TP driver debugging, when TP is ineffective in the case can be found in time, is a question of initialization function, or the structure of the system. Then continue to write the realization of the IIC communication related functions. Were illustrates the above, in order to register TP IIC address.

Finally the use of a statement, read the IIC TP ID, if read IIC ID less than zero, registered failure, TP driven nature also cannot run in Linux system, then the output on the screen "add generic driver failed" prompt the user registration failed. Registration successful it returns 0. Open to closed:

Open, discussed the TP above that when the system, how to realize the closure of the TP is the focus of discussion. Closed by a `tpd_driver_init` function, touch screen, touch screen by open jump to the closed position, is to realize the function of the jump function code below:

```
static void tpd_driver_exit(void){
    GTP_INFO("touch panel driver exit\n");
```

Sleep to open: In the completed by open to sleep after the conversion of natural need from sleep converted to open. Here mainly through `gtp_wakeup_sleep` function, just as its name implies, the function of the function is "touch screen sleep wake up".

Like to sleep not, wake up function implementation is very simple, only needs to be `INT_PORT` feet

up, can realize functions, but the difficulty is to make the system identification is currently "sleep or closed", read this article USES the TP IIC address here, if you have to return to the address of the IIC, explain the status now is sleep, if you don't return to the IIC address, then the state is now closed.

The following is to realize the function of the jump function code:

```
static s8 gtp_wakeup_sleep(struct i2c_client *client){
    u8 retry = 0;
    s8 ret = -1;
    while (retry++ < 5){
        ret = tpd_power_on(client);
        if (ret < 0){
            GTP_ERROR("I2C Power on ERROR!");
            continue;
        }
        GTP_INFO("Ic wakeup by poweron");
        return 0;
    }
    GTP_GPIO_OUTPUT(GTP_INT_PORT, 1);
    msleep(5);
}
```

5.3. TP_Debug

In video software development process, will meet many problems, TP, and one of the most intuitive performance is insensitive touch screen control, click on the screen when unable to accurately the position of the input current click.

Treatment at this time, first check the following TP monomer, in ensuring that TP monomer no problem after checking driver files, input the reason of inaccurate generally in the drive. H file, check whether the resolution of the TP driver setting is correct.

6. IMG development

Camera is a kind of image input device, it is widely used to various places, such as video conference and safety monitor, camera and so on. Camera according to the different functions can be divided into two types, digital cameras and analog camera. Digital cameras, video acquisition device can be produced by the analog video signal into digital signal, and then store it in the computer. Analog cameras to capture video signals must pass a specific video capture card converts analog signals into digital mode, and compress it can convert to the computer after using. Digital cameras can capture images directly, and then through the series and parallel port or USB interface to the computer.

Computer the cameras on the market basically is given priority to with digital cameras, and digital camera to use a new data transmission interface of USB digital camera is given priority to, visible most of them are the products on the market. In this paper, in the design of video software, IMG module from the market to buy a piece of ordinary cameras, from another perspective, also reflected the popularity of video software.

Here are the parameters of the IMG (provided by the manufacturer):

Table 3.IMG configuration table

Parameter	Data
Device name	IMG
Supplier name	LESU
Module	LS1138-V62 V1.0
IC model	GC5024
Size	8.5*8.5
Resolution	0.3M
ID pin	No level
Glass type	M5101

After getting the parameter needs to do is the parameter analysis. Several parameters of the device should be noted before writing the driver software. Mainly include: IC models: GC5024, per pixel resolution: 0.3 M and module type: LS1138 - V62 V1.0. According to the above information, the IMG driver file naming rules: < IC type > _ < module type > _ < resolution size >.

So the driver file named "GC5024_mipi_raw".

Several important note about CAMERA:

(1) Hardware interface: In brief, the Camera interface is divided into two ways: parallel and serial, and at present our platform mainly supports serial mode Mipi interface, Parallel interface and Mipi interface.

(2) The common basic concepts: three-way voltage: camera contains three way of voltage for the analog voltage (VCAMA), digital voltage (VCAMD), IO port voltage (VCAMIO).

I2C signals: BB with the Sensor through the I2C communication (read/write registers), including SCL (I2C Clock) and SDA (I2C Data) two-way signal.

Mipi several lane: mipi data is pair of differential signal, MIPI_RDN and MIPI_RDP, there are a few of the pin, is that a few lane, the same star sensor due to the register setting is different, the output signal may be lane 2 or 4 lane, etc.

Parallel level eight: the parallel interface general Data has 10 pin, named Data0 ~ Data9, parallel the sensor output Data signal is 8 pin, this is the eight root pin received Data0 ~ Data7 or Data2 ~ Data9, need to be configured correctly, is called received high eight or low eight.

Data Format: Sensor output Data Format, for YUV Sensor Data Fomat typically have YUYV YVYU, UYVY etc., the configuration is wrong may cause the color and brightness in the wrong.

The Raw Sensor, Data Format is First Pixel color, divided into R, Gr, Gb, B, right configuration can cause color error.

MCLK: BB to provide Sensor external clock.

PCLK: the Parallel interface of the Sensor output clock, the clock change time, data update again.

Mipi signal: mipi signal including mipi clock and mipi data, the signal is high speed signal, is used to transmit mipi packets.

6.1 IMG circuit diagram analysis

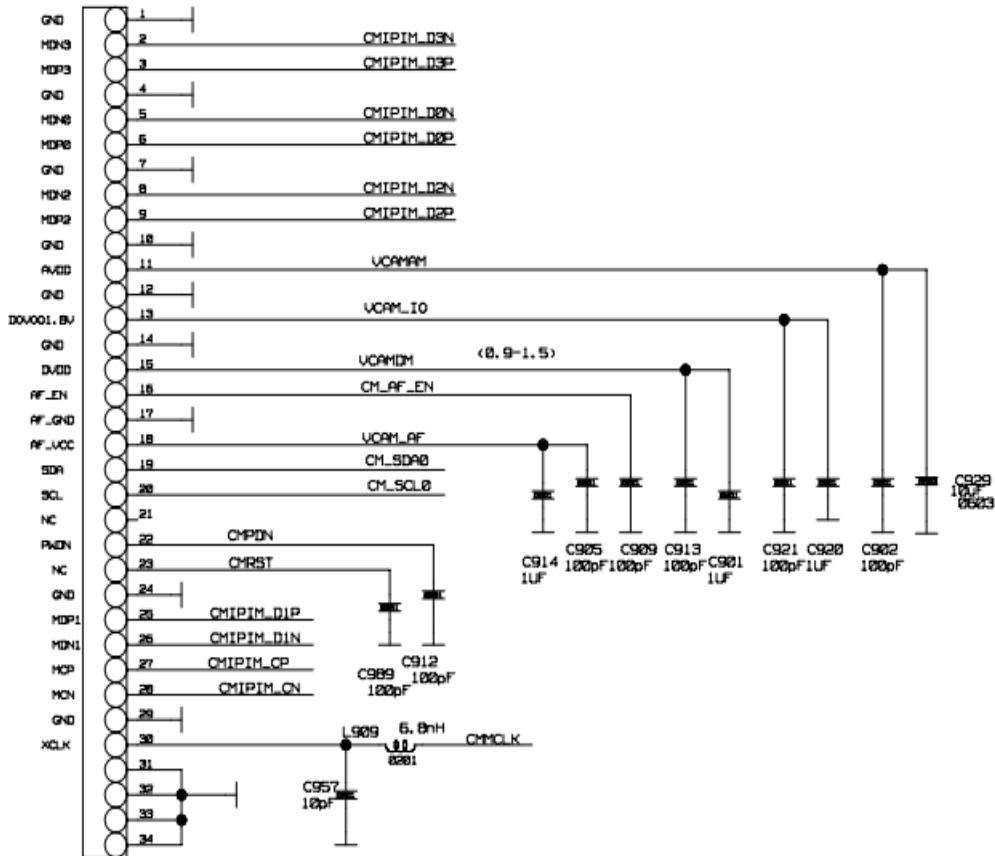


Fig.5. IMG pin figure

From the manufacturer to provide the circuit diagram of can get the following information:

(1) The IMG data transmission, a total of 6 feet CMIPIM_D0N, root canal CMIPIM_D0P, CMIPIM_D1N, CMIPIM_D1P, CMIPIM_D2N, CMIPIM_D2P.

Through this 6 pins for data transmission.

(2) Three way voltage pin: VCAM_IO, VCAM_AF, VCAM_EN. Through this three way voltage control camera work.

(3) The IMG monomer has IIC channels: pin CM_SCL, CM_SDA pins for IIC communication.

(4) Reset the feet of IMG monomer is defined as: CMRST.

6.2. IMG driver writing

IMG according to the architecture of android from top to bottom can be divided into several layers like this:

1) Applications: here is the application of the top, after a LINUX compiled generated Camera APK software.

2) The Application Framework: mainly for IMG software provides the API interface.

3) The JNI: make a layer and layer Libraries realize interactive function.

4) Libraries, including the IMG framework functions and service.

5) HAL: this layer is the hardware abstraction layer, is used to link drive and IMG software service layer;

6) The Kernel: IMG driver layer.

The second to the fourth floor of most part of the android native architecture and code, no longer discuss in this paper. Camera APK software is based on the android native camera APK modify application, also however much discussed here.

And above the LCM and TP is different: LCM and TP driver files relate only to the KERNEL layer, but the IMG drive HAL layers involved, it is to bring a lot of difficulties in the development of video driver software, in order to ensure that in this paper, the structural, here will be IMG drive is divided into two parts: HAL IMG drivers and the KERNEL layer IMG drive, started to discuss below.

6.2.1. IMG drive

Has been discussed above, a complete project, such as the video software development projects, the project each branch mainly by MAKEFILE connected with the SHELL file. IMG driver file structure, the first part is the main MAKE configuration file.

Open/bird/make /<project>/directory, Lord Make files in this directory to this, the first thing you need to do is to configure the IMG driver files, and tell the system "using the camera":

```
CUSTOM_KERNEL_IMGSENSOR = gc5024_mipi_raw
```

Tell LCM system parameters and their name, it must be compiled but, in a Linux operating system, the MAKEFILE is acted as tell us the address, so this article are defined as follows MAKEFILE, tell where system driver file, where to go to compile:

```
include $(srctree)/drivers/misc/mediatek/Makefile.custom
obj-y += kd_sensorlist.o
```

The above two lines of code is to tell the location of the system needs to be compiled file, \$(srctree) means in the current directory tree, for example, I put the project on Linux/home/root/directory of the operating system, the system will go/home/root/drivers/misc/mediatek/Makefile. To find a Makefile in the custom.

The second line of code obj - y + = kd_sensorlist. O, is to make the system will be kd_sensorlist. C file compiled into kd_sensorlist O file. Then this paper discusses kd_sensorlist. C document writing.

Kd_sensorlist. C file need to do is to tell the system, to compile the IMG driver file is already defined. So this article written like this:

```
#if defined(GC5024_MIPI_RAW)
{GC5024MIPI_ID,SENSOR_NAME_GC5024_MIPI_RAW,GC5024MIPI_SensorInit},
#endif
```

If defined in the main MAKE GC5024_MIPI_RAW this macro (above has been defined), the system will go to detect IMG ID model, test read ID with GC5024MIPI_SENSOR_ID are the same, so you need to join GC5024 ID, in Linux system, ID in kernel/drivers/misc/directory mediatek imgsensor/inc kd_imgsensor. H file, the code is as follows:

```
#define SENSOR_DRVNAME_GC5024_MIPI_RAW "gc5024mipi"
#define GC5024MIPI_SENSOR_ID 0x5024
```

Through the definition of the two lines of code, you can know GC5024 ID, system in reading GC5024 ID with kd_imgsensor. H defines the ID of the same, will go to call GC5024MIPI_RAW_SensorInit function, loading initialize camera driver code. Here the IMG file structure has completed the setup, the next step to discuss IMG driver documentation.

6.2.2. IMG driver documentation

IMG driver file with the hello world. C file, you first need to include the corresponding header file: `Linux/i2c.h`, `linux/delay.h`, `kd_camera_hw.h`, `kd_imgsensor.h`, `gc5024mipi_Sensor.h`. In the header file, there are common: `Linux/i2c.H` including the IIC agreement related to the content, including the header files after IIC relevant code can be directly driven file calls `Linux/delay.H` including the time delay function of the common Linux operating system `Asm/system.H` including related to the function of the system. These. H file belongs to the Linux open source information.

IMG hardware while not using IIC protocol for data exchange with Linux systems, but is carrying on the IIC bus device. So need to apply for IIC serial number, need to define an array of IIC structure before `imgsensor_struct` `imgsensor`, then through the IIC bus ID array application. IIC application is completed, you need to do is to use the IIC bus communicate with IMG, communication mainly includes two aspects: reading and writing, the next step in this paper we discuss IMG drive method of reading and writing. This part mainly through `read_cmos_sensor` and `write_cmos_sensor` function realization function. `Read_cmos_sensor` function is the function of reading device data, the implementation method is mainly to invoke `Linux/i2c iReadRegI2C` function of h, through the function return value to u8 bits of data read from a fixed position. Similarly `write_cmos_sensor` function realization method mainly called `Linux/i2c iWriteRegI2C` function of h, 8 through this function will need to write data to the IMG device.

After finished the IMG function to read and write, the first job is to use `read_cmos_sensor` to read the camera ID, will this function named `return_sensor_id`, as long as the call `read_cmos_sensor` function, will read the value of the return and realize the functions as follows:

```
static kal_uint32 return_sensor_id(){
    return ((read_cmos_sensor(0xf0) << 8) | read_cmos_sensor(0xf1));
}
```

Will be completed IMG drive recognition, namely the success after mount on the IIC bus, the first step is mainly do is initialize camera monomer. You first need to define the init function, through the function for a camera initial collect data and prepare the next step work. So how should realize the camera initialization? The answer can be found on the illustrations of the single camera.

Single camera initialization, mainly through the IIC agreement to write command device, in a string of monomer can be found on the manual of initialization parameters: `0xfe`, `0x80`;

Such as call `write_cmos_sensor` function, and add the two parameters `0xfe` and `0x80`, it can be the first parameter `0x80` written to `0xfe` address. By the same token, the camera and preview Settings `preview_setting` capture the `capture_setting` normal `video_setting` normal camera model, and other functions, can be reference to be set by the specification of the parameters of the camera monomer.

6.3. IMG_Debug

In video software development process, will meet many problems, IMG, is the most intuitive performance camera can't use and then processing method, first check the `bird/custom/<Proj>/kernel/camera/camera` path `kd_camera_hw.c` file, the file on the main control the camera single process, electric files on to ensure that the right circumstances, if won't be able to read the ID of cameras, then check if the following IIC channel unobstructed.

In the software layer did not screen out problem after check whether there is a camera monomer.

Acknowledgments

The research was supported by Natural Science Foundation of Anhui Province (1708085MF161), Key

Project of Supporting Program for Outstanding Young Talents in Universities of 2016 (Gxyqzd2016214), Key Project of Natural Science Research of Universities in Anhui (KJ2015A236), and was partially supported by grants of National Science Foundation for Young Scientists of China (No. 61702007), Natural Science Foundation of Anhui province (No. 1508085MC55), Key Project from Education Department of Anhui Province (No. KJ2013A076), Project of Supporting Program for Outstanding Young Talents in Universities of 2016 (Gxyq2017045), Internet and Information Security Anhui Key Laboratory Open Project (ahnis2018002).

References

- [1] Abraham Silberschat, LINUX development Vol. 212. Englewood Cliffs, NJ: Prentice-Hall, 1973.
- [2] Chaudhary, Santosh, Sawai Singh, Susheela Chaudhary. Video software development. *Applied Mathematics* 6(3) (2013): 236.
- [3] Brown, Ann L. "LINUX kernel driver No. 325." (2015).
- [4] Mc Connell, Code Complete. Cengage Learning EMEA, 2016.