HCCHIP SIS9509触摸屏使用说明文档

1. 目录

2. 文档履历

版本号	日期	制/修订人	制/修订记录
1.0	2024.03.02	邱浩佳	新增文档说明

3. 概述

3.1 编写目的

介绍hcchip sis9509触摸屏驱动的使用

3.2 读者对象

软件开发工程师和技术支持工程师。

4. 如何使用

4.1 HCRTOS

4.1.1 设备树配置

在对应的板子设备树文件中，添加响应的节点

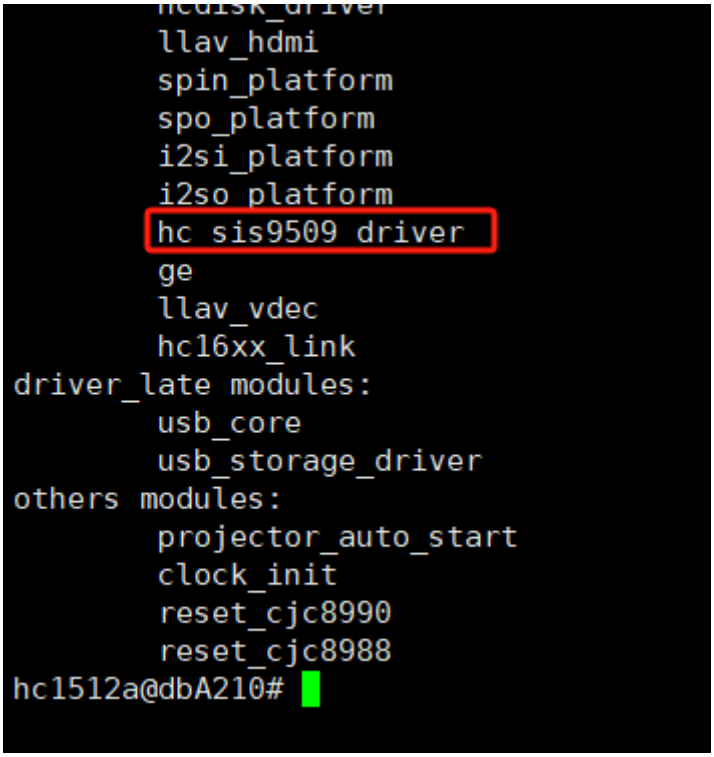
```
i2c@2{
        pinmux-active = <PINPAD_T14 3 PINPAD_T15 3>;
        devpath = "/dev/i2c2";
        baudrate = <100000>;
        mode = "master";
        status = "okay";
};

sis_touchscreen@5c {
        compatible = "sis,sis_touch";
        touch-gpio = <PINPAD_L15 0>;
        i2c-devpath = "/dev/i2c2";
        reg = <0x5c>;
        status = "okay";
};
```

4.1.2 menuconfig配置

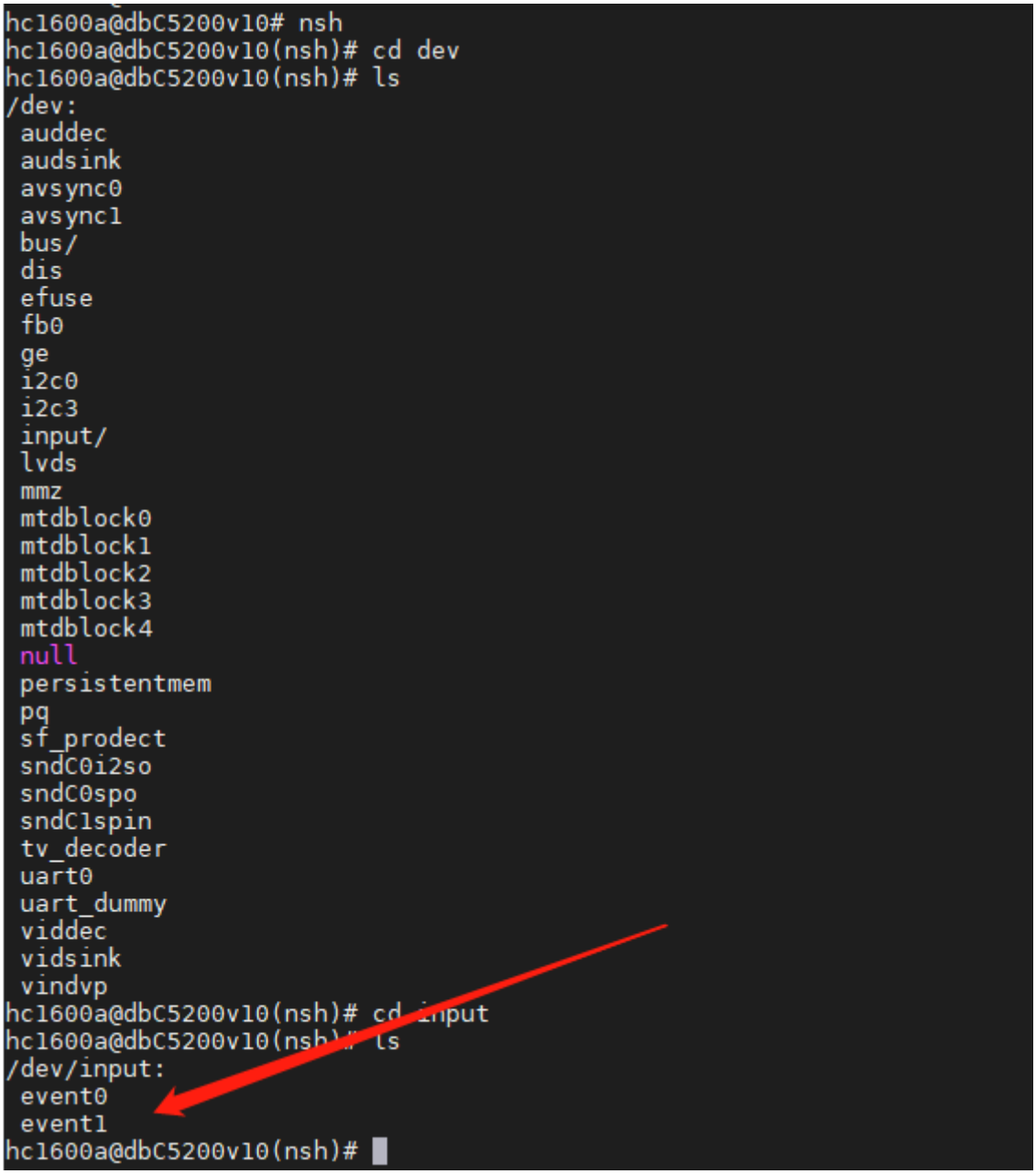
根据下面路径选中ilitek驱动。

```
There is no help available for this option.
Symbol: CONFIG_HC_SIS9509 [=y]
Type  : bool
Prompt: sis9509
  Location:
    -> Components
      -> kernel (BR2_PACKAGE_KERNEL [=y])
        -> Drivers
          -> input event (CONFIG_DRV_INPUT [=y])
            -> tp menu (CONFIG_TP [=y])
  Defined at tp:21
  Depends on: BR2_PACKAGE_KERNEL [=y] && CONFIG_DRV_INPUT [=y] && CONFIG_TP
[=y]
```

配置完成后，在sdk根目录输入：make kernel-rebuild all，进行编译后烧录，在串口控制台终端输入ls命令既可以查看驱动节点。



使用时，需要open对应的input节点。这里open的是/dev/input/event1。具体操作参考Sample Code。



4.1.3 模块接口说明

该模块没有提供额外接口。

4.1.4 模块测试用例与Sample Code

介绍本模块相关的测试用例及相关Sample Code

```
#include <stdlib.h>
#include <poll.h>
#include <unistd.h>
#include <stddef.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <hcuapi/input.h>
#include <kernel/lib/console.h>

```

```c
#define BUF_SIZE 1024

static void print_help(void) {
    printf("**********************************\n");
    printf("input test cmds help\n");
    printf("\tfor example : input_test -i 1\n");
    printf("\t'i'   1 means event1\n");
    printf("**********************************\n");
}

static int input_test(int argc, char *argv[])
{
    int fd;
    struct input_event t;
    struct pollfd pfd;
    char input_buf[BUF_SIZE];
    char *s = "/dev/input/event";

    long tmp;
    int x = 0, y = 0;
    int event_num = -1;
    char ch;
    opterr = 0;
    optind = 0;

    while ((ch = getopt(argc, argv, "hi:")) != EOF) {
        switch (ch) {
        case 'h':
            print_help();
            return 0;
        case 'i':
            tmp = strtoll(optarg, NULL, 10);
            event_num = tmp;
            break;
        default:
            printf("Invalid parameter %c\r\n", ch);
            print_help();
            return -1;
        }
    }
    if (event_num == -1) {
        print_help();
        return -1;
    }

    sprintf(input_buf, "/dev/input/event%d", event_num);

    fd = open(input_buf, O_RDONLY);
    pfd.fd = fd;
    pfd.events = POLLIN | POLLRDNORM;

    if (fd < 0) {
        printf("can't open %s\n", input_buf);
        return -1;
    }

    char name[256] = "Unknown";
    ioctl(fd, EVIOCGNAME(sizeof(name)), name);
```

```
    printf("\nInput Device Name:[%s]. \n", name);

    while (1) {
        if (poll(&pfd, 1, -1) <= 0)
            continue;

        if (read(fd, &t, sizeof(t)) != sizeof(t))
            continue;

        printf("type:%d, code:%d, value:0x%lx\n", t.type, t.code,
               t.value);
    }

    close(fd);

    return 0;
}

CONSOLE_CMD(input, NULL, input_test, CONSOLE_CMD_MODE_SELF, "input test,
press power to exit test")
```

4.1.5 模块调试方法

暂无

4.1.6 常见问题

暂无

4.2 HCLINUX