HCRTOS GT911 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	2023.08.16	邱浩佳	新增文档说明

3. 概述

3.1 编写目的

介绍hcrtos sdk gt911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. 模块介绍

- 该驱动可以直接输出坐标，只需要打开gt911.h文件中的宏GTP_DEBUG_ON。
- 该模块在SDK中的位置：**hcrtos/components/kernel/source/drivers/input/tp/gt911**

4.1 设备树配置

在对应的板子设备树文件中，添加响应的节点

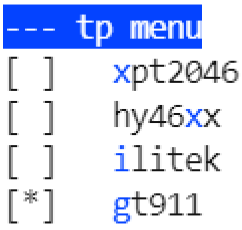
```
i2c@2 {
        pinmux-active = <PINPAD_B19 2 PINPAD_B20 2>;
        devpath = "/dev/i2c2";
        baudrate = <100000>;
        mode = "master";
        status = "okay";
};

goodix_ts@5d {
        compatible = "goodix,gt9xx";
        i2c-devpath = "/dev/i2c2";
        reg = <0x5d>;
        goodix,rst-gpio = <PINPAD_B17 0>;
        goodix,irq-gpio = <PINPAD_B18 0>;
        //goodix,cfg-group0的值根据触摸芯片有所不同，需要向触摸芯片原厂拿取；
        goodix,cfg-group0 = [
        41 20 03 00 05 0A 05 00 01 08 28 05 50
                32 03 05 00 00 00 00 00 00 00 00 00
                00 00 90 30 AA 2E 2A D3 07 00 00 00
                B9 03 24 00 00 00 00 00 00 00 00 00
                00 00 20 4A 94 C5 02 07 00 00 04 98
                22 00 83 29 00 74 30 00 66 39 00 5B
                44 00 5B 00 00 00 00 00 00 00 00 00
                00 00 00 00 00 00 00 00 00 00 00 00
                00 00 00 00 00 00 00 00 00 00 00 00
                00 00 00 19 18 17 16 15 14 11 10 0F
                0E 0D 0C 09 08 07 06 05 04 01 00 00
                00 00 00 FF FF FF FF FF FF 00 02 04
                06 07 08 0A 0C 0D 0E 0F 10 11 12 13
                14 2A 29 28 27 26 25 24 23 22 21 20
                1F 1E 1C 1B 19 FF FF FF FF FF FF FF
                FF FF FF 0A 01
            ];
};
```

4.2 menuconfig配置

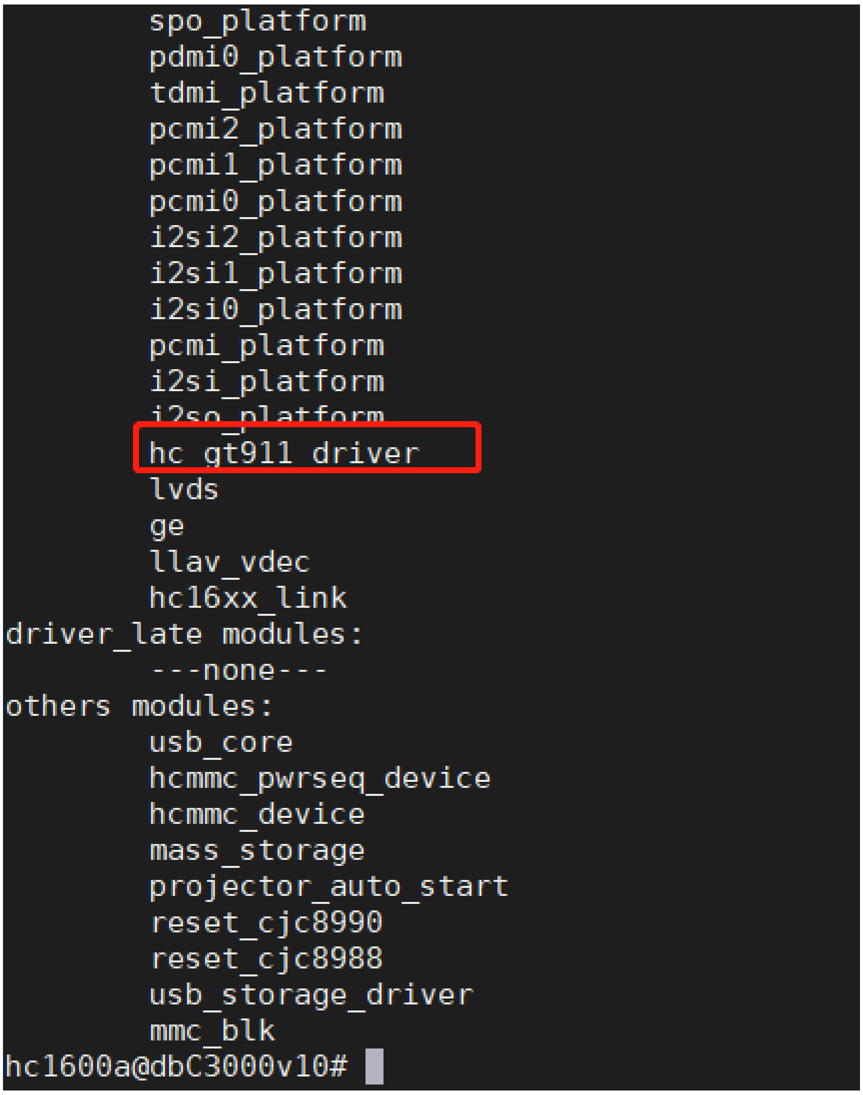
根据下面路径选中ilitek驱动。

```
Location:
  -> Components
    -> kernel (BR2_PACKAGE_KERNEL [=y])
      -> Drivers
        -> input event (CONFIG_DRV_INPUT [=y])
          -> tp menu (CONFIG_TP [=y])
```



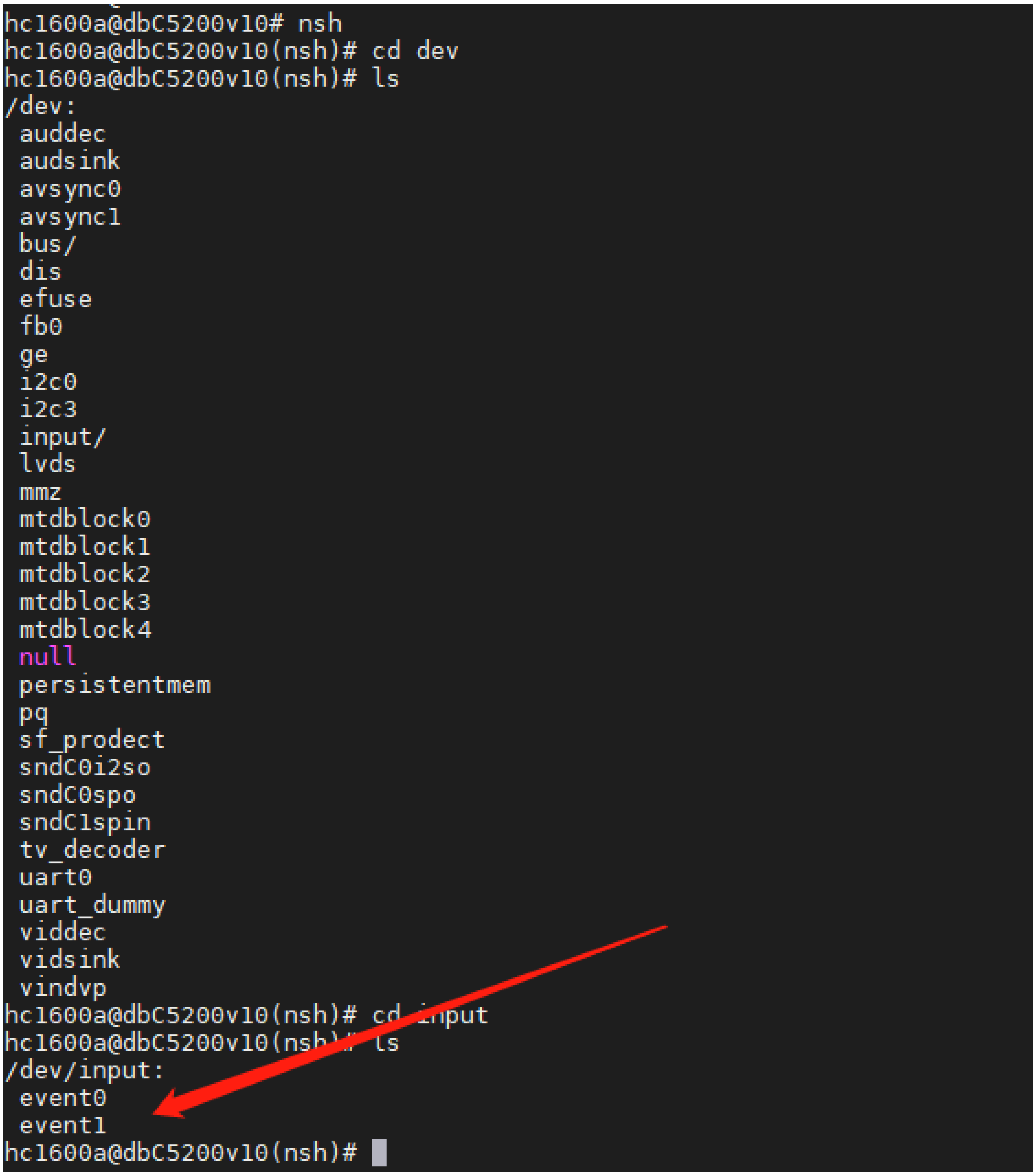
```
--- tp menu
[ ]    xpt2046
[ ]    hy46xx
[ ]    ilitek
[*]    gt911
```

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。



```
        spo_platform
        pdmi0_platform
        tdmi_platform
        pcmi2_platform
        pcmi1_platform
        pcmi0_platform
        i2si2_platform
        i2si1_platform
        i2si0_platform
        pcmi_platform
        i2si_platform
        i2so_platform
        hc_gt911_driver
        lvds
        ge
        llav_vdec
        hc16xx_link
driver_late modules:
        ---none---
others modules:
        usb_core
        hcmmc_pwrseq_device
        hcmmc_device
        mass_storage
        projector_auto_start
        reset_cjc8990
        reset_cjc8988
        usb_storage_driver
        mmc_blk
hc1600a@dbC3000v10#
```

使用时，需要open对应的input节点。这里open的是/dev/input/event0。具体操作参考Sample Code。



```
hc1600a@dbC5200v10# nsh
hc1600a@dbC5200v10(nsh)# cd dev
hc1600a@dbC5200v10(nsh)# ls
/dev:
 auddec
 audsink
 avsync0
 avsync1
 bus/
 dis
 efuse
 fb0
 ge
 i2c0
 i2c3
 input/
 lvds
 mmz
 mtdblock0
 mtdblock1
 mtdblock2
 mtdblock3
 mtdblock4
 null
 persistentmem
 pq
 sf_prodect
 sndC0i2so
 sndC0spo
 sndC1spin
 tv_decoder
 uart0
 uart_dummy
 viddec
 vidsink
 vindvp
hc1600a@dbC5200v10(nsh)# cd input
hc1600a@dbC5200v10(nsh)# ls
/dev/input:
 event0
 event1
hc1600a@dbC5200v10(nsh)#
```

5. 模块接口说明

该模块没有提供额外接口。

6. 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>
```

```c
#include <sys/ioctl.h>
#include <hcuapi/input.h>
#include <kernel/lib/console.h>

#define BUF_SIZE 1024

static void print_help(void) {
        printf("***********************************\n");
        printf("input test cmds help\n");
        printf("\tfor example : input_test -i1\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;
        }

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

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

                printf("type:%d, code:%d, value:%ld\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")
```

7. 模块调试方法

暂无

8. 常见问题

暂无