



# HCRTOS xpt2046 触摸屏使用说明文档

## 1. 目录

### HCRTOS xpt2046 触摸屏使用说明文档

1. 目录
2. 文档履历
3. 概述
  - 3.1 编写目的
  - 3.2 读者对象
4. 模块介绍
5. 模块测试用例与Sample Code
  - 5.1 测试代码
6. 模块调试方法
7. 常见问题

## 2. 文档履历

版本号	日期	制/修订人	制/修订记录
1.0	2023.4.10	邱浩佳	新增文档说明

## 3. 概述

### 3.1 编写目的

介绍xpt2046驱动模块的使用

## 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

- xpt2046触摸屏为电阻触摸屏，
- 使用spi协议进行数据通信。
- 驱动对接了Linux的input子系统，应用层获取数据通过对应的input节点，就可以获取到数据。

### 4.1 设备树配置

```
1  spi-gpio {
2      // pinmux-active = <PINPAD_L28 0 PINPAD_L24 0 PINPAD_L23 0
PINPAD_L25 0>;
3      status = "okay";
4
5      gpio-sck  = <PINPAD_L28>;
6      gpio-mosi = <PINPAD_L24>;
7
8      gpio-miso = <PINPAD_L23>;
9
10     cs-gpios  = <PINPAD_L25>;
11
12     num-chipselects = <1>;
13
14     xpt2046{
15         status = "okay";
16
17         touch-gpio = <PINPAD_L27>;
18
19         devpath = "/dev/xpt2046";
20     };
21 };
```

### 4.2 menuconfig配置

在SDK根目录输入make menuconfig，按照下面的路径进行勾选hy46xx\_ts驱动。

```

1 | There is no help available for this option.
2 | Symbol: CONFIG_HC_XPT2046 [=n]
3 | Type : bool
4 | Prompt: xpt2046
5 |   Location:
6 |     -> Components
7 |       -> kernel (BR2_PACKAGE_KERNEL [=y])
8 |       -> Drivers
9 |         -> input event (CONFIG_DRV_INPUT [=y])
10 |          -> tp menu (CONFIG_TP [=y])
11 |   Defined at tp:1
12 |   Depends on: BR2_PACKAGE_KERNEL [=y] && CONFIG_DRV_INPUT [=y] &&
CONFIG_TP [=y]

```

## 4.3 编译

配置完成后，输入make kernel-rebuild all，进行编译和烧录，在串口控制台输入ls命令就可以看到hy46xx驱动。

## 5. 模块测试用例与Sample Code

在SDK根目录输入make menuconfig，根据下面路径选中测试命令

```

1 | Location:
2 |   -> Components
3 |     -> Cmds (BR2_PACKAGE_CMDS [=y])

```

```

--- Cmds
[*] OS operations --->
[ ] pthread operations
[*] Nsh operations --->
[ ] sound test operations ----
[*] lsmod operations
[ ] adc test operations ----
[ ] fb test operations
[*] input event operations
[ ] mtd operations
[ ] spi operations
[ ] persistent memory operations
[ ] pok test operations
[ ] uart test operations
[ ] watchdog test operations
[ ] Efuse bits dump operations
[ ] hdmi rx test operations
[ ] tv decoder(cvbs in) test operations

```

选中后，输入make cmds-rebuild all，进行编译和烧录。在串口控制台输入下面命令即可以查看测试命令。

```

hc1512a@dbB200#
hc1512a@dbB200# input -h
*****
input test cmds help
    for example : input_test -i1
                  'i'    1 means event1
*****
hc1512a@dbB200# input -i1

```

按压触摸屏就会接收到触摸点的消息。

## 5.1 测试代码

```

1  #include <stdlib.h>
2
3  #include <poll.h>
4  #include <unistd.h>
5  #include <stddef.h>
6  #include <stdio.h>
7  #include <fcntl.h>
8  #include <sys/ioctl.h>
9  #include <hcuapi/input.h>
10 #include <kernel/lib/console.h>
11
12 #define BUF_SIZE 1024
13
14 static void print_help(void) {
15     printf("*****\n");
16     printf("input test cmds help\n");
17     printf("\tfor example : input_test -i1\n");
18     printf("\t'i'    1 means event1\n");
19     printf("*****\n");
20 }
21
22 static int input_test(int argc, char *argv[])
23 {
24     int fd;
25     struct input_event t;
26     struct pollfd pfd;
27     char input_buf[BUF_SIZE];
28     char *s = "/dev/input/event";
29
30     long tmp;
31     int x = 0, y = 0;
32     int event_num = -1;
33     char ch;
34     opterr = 0;
35     optind = 0;
36
37     while((ch = getopt(argc, argv, "hi:")) != EOF){
38         switch (ch) {
39             case 'h':
40                 print_help();
41                 return 0;
42             case 'i':
43                 tmp = strtoll(optarg, NULL, 10);

```

```

43         event_num = tmp;
44         break;
45     default:
46         printf("Invalid parameter %c\r\n", ch);
47         print_help();
48         return -1;
49     }
50 }
51 if(event_num == -1)
52 {
53     print_help();
54     return -1;
55 }
56
57 sprintf(input_buf, "/dev/input/event%d", event_num);
58
59 fd = open(input_buf, O_RDONLY);
60 pfd.fd = fd;
61 pfd.events = POLLIN | POLLRDNORM;
62
63 if(fd < 0){
64     printf("can't open %s\n", input_buf);
65     return -1;
66 }
67
68 while (1) {
69     if (poll(&pfd, 1, -1) <= 0)
70         continue;
71
72     if (read(fd, &t, sizeof(t)) != sizeof(t))
73         continue;
74
75     printf("type:%d, code:%d, value:%ld\n", t.type, t.code,
t.value);
76
77     if (t.type == EV_KEY) {
78         printf("key %d %s\n", t.code,
79             (t.value) ? "Pressed" :
"Released");
80
81         if (t.code == KEY_POWER && !t.value) {
82             while (read(fd, &t, sizeof(t)) ==
sizeof(t))
83                 ;
84             break;
85         }
86     }
87     else{
88         if (t.type == EV_ABS)
89         {
90             if (t.type == EV_ABS&& t.code == ABS_X) {
91                 x = t.value;
92             }
93             if (t.type == EV_ABS && t.code == ABS_Y) {
94                 y = t.value;
95             }
96         }
97         if (t.type == EV_SYN) {
98             printf("(%4d %4d)\n", x, y);

```

```
98         }
99     }
100 }
101
102     close(fd);
103
104     return 0;
105 }
106
107     CONSOLE_CMD(input, NULL, input_test, CONSOLE_CMD_MODE_SELF, "input test,
press power to exit test")
```

## 6. 模块调试方法

---

使用测试命令测试。

## 7. 常见问题

---

暂无