HCRTOS_blt1680多点电容触摸屏使用说明文档

1. 文档履历

版本号	日期	制/修订人	制/修订记录
1.0	2023.11.09	邱浩佳	新增说明

---

2. 概述

2.1 编写目的

介绍和指导如何在hcrtos上使用blt1680的多点电容触摸屏;

2.2 读者对象

hcrtos的开发者和FAE工程师；

3. 模块介绍

1. **blt1680触摸屏需要加载对应的固件才可以产生中断及正常使用，代码目录下的固件并不一定适用，需要联系原厂调试合适的固件；**
2. **由于该触摸芯片出厂并不内置固件，所以在第一次使用时需要一定时间加载固件进去；**
3. hcrtos上blt1680不支持电源管理，即不支持睡眠模式；
4. 触摸芯片支持在一定时间没有触摸会进入低功耗模式，有触摸时会自动唤醒；
5. blt1680触摸芯片最大支持10点触摸，需要固件支持；

3.1 设备树的配置

```
i2c@1{
        pinmux-active = <PINPAD_B02 3 PINPAD_B03 3>;    //i2c所使用的引脚及复用功能
        devpath = "/dev/i2c1";  //i2c节点所生产的路径
        baudrate = <100000>;    //i2c波特率
        mode = "master";        //i2c模式，此处为master
        status = "okay";        //okay代表开启
};

betterlife_ts@2c{
        i2c_devpath = "/dev/i2c1";  //触摸屏所使用的i2c节点
        i2c_addr = <0x2c>;          //触摸屏的7位设备地址
        reset_gpio = <PINPAD_L24 0>;//触摸屏复位所使用的引脚
        irq_gpio = <PINPAD_L27 0>;  //触摸屏中断所使用的引脚
   //   vdd_name = "vdd28";         //触摸屏所使用的电源管理，目前不支持
   //   virtualkeys = <80 900 120 44 240 900 120 44 400 900 120 44>;    //实体触摸按键，需要硬件支持
        TP_MAX_X = <480>;   //触摸屏X坐标分辨率
        TP_MAX_Y = <800>;   //触摸屏Y坐标分辨率
        status = "okay";    //okay代表开启
};
```

3.2 menuconfig的配置

3.2.1 i2c的开启

```
Symbol: CONFIG_I2C_SCB_MASTER [=y]
Type  : bool
Prompt: I2C SCB Master
  Location:
    -> Components
      -> kernel (BR2_PACKAGE_KERNEL [=y])
        -> Drivers
(1)       -> I2C Driver Support (CONFIG_I2C [=y])
  Defined at i2c:18
  Depends on: BR2_PACKAGE_KERNEL [=y] && CONFIG_I2C [=y]
```

3.2.2 blt1680触摸屏的开启

```
There is no help available for this option.
Symbol: CONFIG_HC_BLT1680 [=y]
Type  : bool
Prompt: blt1680
  Location:
    -> Components
      -> kernel (BR2_PACKAGE_KERNEL [=y])
        -> Drivers
          -> input event (CONFIG_DRV_INPUT [=y])
            -> tp menu (CONFIG_TP [=y])
  Defined at tp:13
  Depends on: BR2_PACKAGE_KERNEL [=y] && CONFIG_DRV_INPUT [=y] && CONFIG_TP
[=y]
```

3.2.3 测试命令的开启

```
There is no help available for this option.
Symbol: CONFIG_CMDS_INPUT [=y]
Type  : bool
Prompt: input event operations
  Location:
    -> Components
      -> Cmds (BR2_PACKAGE_CMDS [=y])
  Defined at source:46
  Depends on: BR2_PACKAGE_CMDS [=y] && CONFIG_DRV_INPUT [=y]
```

3.2.4 编译

```
make kernel-rebuild cmds-rebuild all
```

3.2.5 测试命令的使用

在串口控制终端输入：input -i1，这里的1代表event1；

```
hc1512a@dbB200# input -i1
ID:0, X:360, Y:717, W:1
type:3, code:48, value:1
type:3, code:50, value:1
type:3, code:54, value:717
type:1, code:330, value:1
key 330 Pressed
```

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

```c
#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>

#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")
```

5. 模块调试方法

调试log宏BTL_DEBUG_SUPPORT的开启：
components/kernel/source/drivers/input/tp/blt1680/bl_chip_custom.h；

开启后会在初始化以及触摸时打印调试信息；

6. 常见问题

暂无；