



HICHIP SC7A20 gsensor userguide

1. 文档履历

版本号	日期	制/修订人	制/修订记录
1.0	2023/12/01	邱浩佳	新增文档说明
2.0	2023/12/19	邱浩佳	修改数据处理方式

HICHIP SC7A20 gsensor userguide

1. 文档履历
2. 概述
 - 2.1 编写目的
 - 2.2 读者对象
3. 模块介绍
4. linux
 - 4.1 模块接口说明
 - 4.2 模块测试用例与Sample Code
 - 4.2.1 开启模块
 - 4.2.2 设备树的配置
 - 4.2.3 编译
5. rtos
 - 5.1 模块接口说明
 - 5.2 模块测试用例与Sample Code
 - 5.2.1 如何开启测试命令
 - 5.2.2 menuconfig的配置
 - 5.2.3 设备树配置
 - 5.2.4 编译
6. 调试信息

2. 概述

2.1 编写目的

介绍hichip平台下的sc7a20 gsensor的使用;

2.2 读者对象

hichip软件开发工程师和技术支持工程师。

3. 模块介绍

- hichip软件平台下的sc7a20驱动支持xyz三个方向 $\pm 2G$ 范围的加速度检查;
- 目前驱动采用轮询模式读取数据, 应用层可直接read对应的/dev/input/eventx节点, 获取到对应的xyz三个方向数据;
- 数据的解释: 通过驱动获取到的值为对应方向的加速度计值 * 1000000。如: Gsensor x=-8789 y=20507 z=9892187; 此时平放在桌面上, xy轴存在一定的漂移, 这是正常的情况变化; z轴992187的值除以1000000, 既可以得出约 $9.8m/s^2$ 的重力加速度;

4. linux

4.1 模块接口说明

暂无;

4.2 模块测试用例与Sample Code

```
1 static int input_test(int argc, char *argv[])
2 {
3     int fd;
4     struct input_event t;
5     struct pollfd pfd;
6     char input_buf[BUF_SIZE];
7     char *s = "/dev/input/event0";
8
9     long tmp;
10
11     fd = open(input_buf, O_RDONLY);
12     pfd.fd = fd;
13     pfd.events = POLLIN | POLLRDNORM;
14
15     if (fd < 0) {
16         printf("can't open %s\n", input_buf);
17         return -1;
18     }
19 }
```

```

20     while (1) {
21         if (poll(&pfd, 1, -1) <= 0)
22             continue;
23
24         if (read(fd, &t, sizeof(t)) != sizeof(t))
25             continue;
26
27         printf("type:%d, code:%d, value:0x%lx\n", t.type, t.code,
28               t.value);
29     }
30
31     close(fd);
32
33     return 0;
34 }

```

4.2.1 开启模块

在SDK根目录下：输入make linux-menuconfig，根据路径打开测试命令

```

1  There is no help available for this option.
2  Symbol: HC_SC7A20H [=y]
3  Type : boolean
4  Prompt: HC_SC7A20H sensing driver
5  Location:
6      -> Device Drivers
7      -> HC drivers
8          -> input driver (HC_INPUT [=y])
9              -> misc driver (HC_MISC [=y])
10                  -> Gravity sensing driver (HC_GSENSOR [=y])
11  Defined at drivers/hcd/drivers/input/misc/gsensor/Kconfig:5
12  Depends on: HC_INPUT [=y] && HC_MISC [=y] && HC_GSENSOR [=y]

```

4.2.2 设备树的配置

在/board/hichip/hc16xx/common/dts/hc16xx-common.dtsi文件下添加

```

1  i2c_gpio: i2c-gpio {
2      compatible = "i2c-gpio";
3      status = "okay";
4  };

```

在对应的板子文件下，比如：/board/hichip/hc16xx/common/dts/hc16xx-db-d3100-v30.dts添加以下节点：

```

1  &i2c_gpio {
2      compatible = "i2c-gpio";
3      gpios = <&gpio_L 2 GPIO_ACTIVE_HIGH /* sda */
4              &gpio_L 1 GPIO_ACTIVE_HIGH /* scl */
5              >;
6      i2c-gpio,sda-open-drain;
7      i2c-gpio,scl-open-drain;

```

```

8      i2c-gpio,delay-us = <2>; /* ~170 kHz */
9      #address-cells = <1>;
10     #size-cells = <0>;
11     status = "okay";
12     sc7a20@19 {          //或者将该节点添加到对应i2c节点下面;
13         compatible = "Amlogic_gs";
14         reg = <0x18>;
15     };
16 };

```

4.2.3 编译

执行命令：make linux-rebuild all;

5. rtos

5.1 模块接口说明

暂无;

5.2 模块测试用例与Sample Code

```

1  static int input_test(int argc, char *argv[])
2  {
3      int fd;
4      struct input_event t;
5      struct pollfd pfd;
6      char input_buf[BUF_SIZE];
7      char *s = "/dev/input/event0";
8
9      long tmp;
10
11     fd = open(input_buf, O_RDONLY);
12     pfd.fd = fd;
13     pfd.events = POLLIN | POLLRDNORM;
14
15     if (fd < 0) {
16         printf("can't open %s\n", input_buf);
17         return -1;
18     }
19
20     while (1) {
21         if (poll(&pfd, 1, -1) <= 0)
22             continue;
23
24         if (read(fd, &t, sizeof(t)) != sizeof(t))
25             continue;
26
27         printf("type:%d, code:%d, value:0x%lx\n", t.type, t.code,

```

```

28         t.value);
29     }
30
31     close(fd);
32
33     return 0;
34 }

```

测试代码位置: components/cmds/source/input_event/input_test.c;

5.2.1 如何开启测试命令

```

1  There is no help available for this option.
2  Symbol: CONFIG_CMDS_INPUT [=y]
3  Type   : bool
4  Prompt: input event operations
5      Location:
6          -> Components
7              -> Ccmds (BR2_PACKAGE_CMDS [=y])
8      Defined at source:46
9      Depends on: BR2_PACKAGE_CMDS [=y] && CONFIG_DRV_INPUT [=y]

```

编译命令: make cmds-rebuild all

5.2.2 menuconfig的配置

```

1  /* sc7a20驱动的开启 */
2  There is no help available for this option.
3  Symbol: CONFIG_GSENSOR_SC7A20 [=y]
4  Type   : bool
5  Prompt: hc gsensor sc7a20h
6      Location:
7          -> Components
8              -> kernel (BR2_PACKAGE_KERNEL [=y])
9                  -> Drivers
10                      -> input event (CONFIG_DRV_INPUT [=y])
11                          -> misc (CONFIG_MISC [=y])
12                              -> sensor (CONFIG_SENSOR [=y])
13                                  -> gsensor (CONFIG_SENSOR_GSENSOR [=y])
14      Defined at gsensor:1
15      Depends on: BR2_PACKAGE_KERNEL [=y] && CONFIG_DRV_INPUT [=y] &&
CONFIG_MISC [=y] && CONFIG_SENSOR [=y] && CONFIG_SENSOR_GSENSOR [=y]
16
17  /* gpio i2c 驱动的开启 */
18  Prompt: I2C GPIO Master
19      Location:
20          -> Components
21              -> kernel (BR2_PACKAGE_KERNEL [=y])
22                  -> Drivers
23                      -> I2C Driver Support (CONFIG_I2C [=y])

```

5.2.3 设备树配置

```
1  gpio-i2c@0 {
2      device_type = "hichip,hcrtos-setup-setbit";
3      reg_bit = <0xb8800094 16 1>;
4      sda-pinmux = <PINPAD_L02>;
5      scl-pinmux = <PINPAD_L01>;
6      status = "okay";
7      simulate;
8  };
9
10 sc7a20@19 {
11     i2c-devpath = "/dev/gpio-i2c0";           //sc7a20驱动所使用的i2c控制器
12     reg = <0x18>;                             //七位地址
13 };
```

5.2.4 编译

make kernel-rebuild all

6. 调试信息

可以通过下面define开启驱动内部的调试信息。

```
25
26 #if 1
27 #define mmapprintk(x...) printk(x)
28 #else
29 #define mmapprintk(x...)
30 #endif
31
32 #if 0
33 #define mmapprintkd(x...) printk(x)
34 #else
35 #define mmapprintkd(x...)
36 #endif
37
38 #if 1
39 #define mmapprintkf(x...) printk(x)
40 #else
41 #define mmapprintkf(x...)
42 #endif
43
44 #define SC7A20_SPEED 200 * 1000
45 #define SC7A20_DEVID 0x11
46 typedef char status_t;
47 /*status*/
48 #define SC7A20_OPEN 1
NORMAL sc7a20.c
```