

Introduction to Embedded Software

Practice #6

Expose the Nunchuk functionality to user space

Dongkun Shin

Embedded Software Laboratory

Sungkyunkwan University

<http://nyx.skku.ac.kr/>

Term Project

Term Project

- **Implement your own program using device driver and kernel module**
- **Free topic**
- **One team per two persons**
- **At least**
 - One output Sensor
 - Two input sensors including Wii Nunchuk

Evaluation Policy

- **Project Plan Report : 5%**
- **Use of what you have learned : 20%**
- **Completeness : 25%**
- **Difficulty : 30%**
- **Novelty : 15%**
- **Final Report : 5%**

Report

- **Project Plan Report**
 - Topic, Implementation content, Plan, **Required equipment (sensors)**
 - Submit to i-Campus
 - Due ~ 5/19 (Sun) 23:59:59
- **Final Report**
 - Just describe your program briefly so that we can understand it.

WiiNunchuk Device

Driver Review

Module Init & Module Exit

- Register & Delete Device

```
191  static int __init wiinunchuk_init(void) {
192      return i2c_add_driver(&wiinunchuk_driver);
193  }
194
195  static void __exit wiinunchuk_exit(void) {
196      i2c_del_driver(&wiinunchuk_driver);
197  }
```

Device Driver Structure

```
166 static const struct i2c_device_id wiinunchuk_id[] = {
167     { "nunchuk", 0 },
168     { }
169 };
170 MODULE_DEVICE_TABLE(i2c, wiinunchuk_id);
171
172 #ifdef CONFIG_OF
173 static const struct of_device_id wiinunchuk_dt_ids[] = {
174     { .compatible = "nintendo,nunchuk", },
175     { }
176 };
177 MODULE_DEVICE_TABLE(of, wiinunchuk_dt_ids);
178 #endif
179
180 static struct i2c_driver wiinunchuk_driver = {
181     .probe = wiinunchuk_probe,
182     .remove = wiinunchuk_remove,
183     .id_table = wiinunchuk_id,
184     .driver = {
185         .name = "nunchuk",
186         .owner = THIS_MODULE,
187         .of_match_table = of_match_ptr(wiinunchuk_dt_ids),
188     },
189 };
```


Device Initialization

- **Send two bytes to the device twice**

```
70 static void wiinunchuk_open(struct input_polled_dev *polled_input) {
71     struct wiinunchuk_device *wiinunchuk = polled_input->private;
72     struct i2c_client *i2c = wiinunchuk->i2c_client;
73     static uint8_t data1[2] = { 0xf0, 0x55 };
74     static uint8_t data2[2] = { 0xfb, 0x00 };
75     struct i2c_msg msg1 = { .addr = i2c->addr, .len = 2, .buf = data1 };
76     struct i2c_msg msg2 = { .addr = i2c->addr, .len = 2, .buf = data2 };
77
78     i2c_transfer(i2c->adapter, &msg1, 1);
79     i2c_transfer(i2c->adapter, &msg2, 1);
80
81     wiinunchuk->state = 0;
82 }
```

Input Polling

- **State 0: Write 0x00 to the bus**
- **State 1: Read 6 bytes from the device**

```
33 switch (wiinunchuk->state) {
34 case 0:
35     i2c_transfer(i2c->adapter, &cmd_msg, 1);
36     wiinunchuk->state = 1;
37     break;
38
39 case 1:
40     i2c_transfer(i2c->adapter, &data_msg, 1);
41
42     jx = b[0];
43     jy = b[1];
44     ax = (b[2] << 2) | ((b[5] >> 2) & 0x3);
45     ay = (b[2] << 2) | ((b[5] >> 4) & 0x3);
46     az = (b[2] << 2) | ((b[5] >> 6) & 0x3);
47     z = !(b[5] & 1);
48     c = !(b[5] & 2);
49
50     input_report_abs(polled_input->input, ABS_X, jx);
51     input_report_abs(polled_input->input, ABS_Y, jy);
52     input_report_abs(polled_input->input, ABS_RX, ax);
53     input_report_abs(polled_input->input, ABS_RY, ay);
54     input_report_abs(polled_input->input, ABS_RZ, az);
55     input_report_key(polled_input->input, BTN_C, c);
56     input_report_key(polled_input->input, BTN_Z, z);
57     input_sync(polled_input->input);
58
59     wiinunchuk->state = 0;
60     printk("wiinunchuk: j=%.3i,%.3i a=%.3x,%.3x,%.3x %c%c\n",
61           jx, jy, ax, jy, az, c?'C':'c', z?'Z':'z');
62
63     break;
```

**Expose the WiiNunchuk
to user space**

Evttest

```
pi@raspberrypi:~ $ evtest
No device specified, trying to scan all of /dev/input/event*
Not running as root, no devices may be available.
Available devices:
/dev/input/event0:      Logitech USB Keyboard
/dev/input/event1:      Logitech USB Keyboard
/dev/input/event2:      Nintendo WiiNunchuk
Select the device event number [0-2]: █
```

```
Event: time 1557749240.469300, ----- SYN_REPORT -----
Event: time 1557749240.589316, type 3 (EV_ABS), code 3 (ABS_RX), value 444
Event: time 1557749240.589316, type 3 (EV_ABS), code 4 (ABS_RY), value 447
Event: time 1557749240.589316, type 3 (EV_ABS), code 5 (ABS_RZ), value 444
Event: time 1557749240.589316, type 1 (EV_KEY), code 309 (BTN_Z), value 1
Event: time 1557749240.589316, ----- SYN_REPORT -----
Event: time 1557749240.709317, type 3 (EV_ABS), code 3 (ABS_RX), value 393
Event: time 1557749240.709317, type 3 (EV_ABS), code 4 (ABS_RY), value 395
Event: time 1557749240.709317, type 3 (EV_ABS), code 5 (ABS_RZ), value 392
Event: time 1557749240.709317, type 1 (EV_KEY), code 309 (BTN_Z), value 0
```

Exercise

- **Implement your Evtest program**
 - Open /dev/input/event# file
 - Read file with input_event struct

```
struct input_event {  
    struct timeval time;  
    __u16 type;  
    __u16 code;  
    __s32 value;  
};
```

Output

```
pi@raspberrypi:~ $ ./evtest /dev/input/event2
Event: time 1557748430.319313, type 3 (Absolute), code 5 (Rz), value 291
Event: time 1557748430.319313, type 0 (Sync), code 0 (Sync), value 0
Event: time 1557748430.439310, type 3 (Absolute), code 5 (Rz), value 290
Event: time 1557748430.439310, type 0 (Sync), code 0 (Sync), value 0
Event: time 1557748437.519308, type 3 (Absolute), code 4 (Ry), value 293
Event: time 1557748437.519308, type 0 (Sync), code 0 (Sync), value 0
Event: time 1557748437.639302, type 3 (Absolute), code 4 (Ry), value 292
Event: time 1557748437.639302, type 0 (Sync), code 0 (Sync), value 0
Event: time 1557748443.999305, type 3 (Absolute), code 3 (Rx), value 236
Event: time 1557748443.999305, type 3 (Absolute), code 4 (Ry), value 239
Event: time 1557748443.999305, type 3 (Absolute), code 5 (Rz), value 237
Event: time 1557748443.999305, type 0 (Sync), code 0 (Sync), value 0
Event: time 1557748444.119311, type 3 (Absolute), code 3 (Rx), value 277
Event: time 1557748444.119311, type 3 (Absolute), code 4 (Ry), value 278
Event: time 1557748444.119311, type 3 (Absolute), code 5 (Rz), value 279
Event: time 1557748444.119311, type 0 (Sync), code 0 (Sync), value 0
Event: time 1557748444.239306, type 3 (Absolute), code 3 (Rx), value 252
Event: time 1557748444.239306, type 3 (Absolute), code 4 (Ry), value 252
Event: time 1557748444.239306, type 3 (Absolute), code 5 (Rz), value 252
Event: time 1557748444.239306, type 0 (Sync), code 0 (Sync), value 0
```