Device Driver Development of External Human Interaction Device for the Android

Vadigepalli Prathyusha
M.Tech (Embedded systems)
Department of ECE,
B.V. Raju Institute of Technology,
Narsapur, Medak, Telangana State, India.

ABSTRACT:
Smart phones become popular. They have various sensors and connection modules. They employ a mobile operating system for the control of sensors and modules. In this situation, a large amount of data comes to the operating system from sensors and modules. While such data is important for effective operations of smart phones, it is not well known that how it affects the operating system performance. In this paper, we prepare an experiment environment that mimics a smart phone. We selected the Android as an operating system and the Pandaboard-ES as a target board. The Pandaboard-ES equips a I2C controller module. We select Accelerometer sensor to operate the Android via I2C. We implement programs to use Accelerometer as the input device of the Android, and improve this program. We make change profiling software for the Linux and are enable to use it on the Android.

Index-terms: ARM Cortex A9 controller, keyboard, Panda board, Accelerometer.

I. INTRODUCTION:
Recently, smart phones spread all over the world. In 2008, almost all cellular phones are Feature phones while now 70% of cell phones are smart phones. Furthermore, the adoption rate is expected to reach 80% by the end of 2016. The difference of feature phones and smart phones is that the smart phones employ a mobile operating system (OS), such as iOS or Android. Another feather of smart phones is that it is equipped with in number of sensor modules that include GPS, Bluetooth, acceleration sensor, direction sensor and luminance sensor and so on. Because it has many modules and it also can connect kinds of external devices, a large amount of data comes from modules and/or external devices to the smart phones, so smart phones must process these data. These data are too large, so if we don't process these data effectively, the application goes down at worst. There is no study how we can process data from the external device effectively and how external devices can be connected with Android.

The objective of this study is to examine how the data from the external device affect the Android and how time the input event complete. For our objective, we connect the external device with the Android and handle the Android by the external device. We implement a driver to get the data from Accelerometer sensor and use this data to make input event to the Android. We get the processing time of the input event by using this program.

II.RELATED WORK:
2.1 Scope of the project:
• Study of Input subsystem in Linux.
• Study of ARM Cortex A9 controller.
• Study of i2c controller in OMAP4460 processor.
• Linux Porting on OMAP4460 controller.
• Implementing Accelerometer Sensor driver program at kernel level.
• Implementing a get event application programs at user level to communicate accelerometer sensor.
• Testing the Interaction between kernel to user level using application and kernel program.
2.2 Execution Environment:
We set up the following environment to get data from the external device and handle these data.
OS:
We choose Android as an embedded operating system on smartphone because Android is an open source software, and it is Linux based operating system so we can use some software for Linux. The version we selected is 4.0.4(Ice Cream Sandwich).
Board:
We choose Pandaboard-ES[12] for the target board. Pandaboard-ES is a single-board computer based on the OMAP4460 developed by Texas Instruments. The features Pandaboard-ES are a dual-core 1.2GHzCPU and 384MHz GPU, 1GB low power DDR2 RAM, onboard 10/100 Ethernet, 802.11 b/g/n, Bluetooth v2.1, general purpose expansion header, and so on. Some smart phones use the OMAP4460, so this board is similar to the smart phone. Figure1 is the appearance of Pandaboard-ES.

1. Supports multiple functions
2. ARM cortex A9 processor
3. On chip memory (SRAM-56K,ROM-48k)
4. External memory(DRAM-1GB)
5. GPMC interfacing for NAND and NOR
6. Supports 4 UART controllers and 4 I2C
7. Audio Back End Systems, Display Sub System and Imaging Sub System

2.3. Accelerometer sensor (ADXL345):
The ADXL345 is a complete 3axisaccelerationmeasurement system with a selectable measurement range of ±2 g, ±4 g, ±8 g, or ±16 g. The accelerometer is a built in electronic component that measures tilt and motion. It is also capable of detecting rotation and motion gestures such as swinging and shaking.

2.4 DEVICE DRIVER:
We implement device driver for Accelerometer. Our Android prepares insmod and rmmod commands so we can use kernel module on this Android. Device driver for ubuntu already exists, but we cannot use this driver on the Android. We Implement a get event application programs at user level to communicate accelerometer sensor and testing the Interaction between kernel to user level using application and kernel program.
Proposed Block Diagram:

III. RESULTS:

3.1 Graph:

Compare Three Versions:

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Virtual Input Device</th>
<th>Input Command</th>
<th>Device Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

Table-1: Command Result

3.2 LOGFILE:

127|root@android:/ # getevent -t

add device 1: /dev/input/event0
name: "ft5x06_ts"
could not get driver version for /dev/input/mice, Not a typewriter
add device 2: /dev/input/event4
name: "PandaES Headset Jack"
add device 3: /dev/input/event5
name: "gpio-keys"
add device 4: /dev/input/event1
name: "ads7846"
could not get driver version for /dev/input/mouse0, Not a typewriter
add device 5: /dev/input/event2
name: "ADXL34x accelerometer"
could not get driver version for /dev/input/js0, Invalid argument
add device 6: /dev/input/event3
name: "twl6040:vibrator"
287-194183: /dev/input/event2: 0003 0000 000000e7
287-194213: /dev/input/event2: 0003 0001 fffffff8
287-194213: /dev/input/event2: 0003 0002 00000185
287-194213: /dev/input/event2: 0000 0000 00000000
287-237274: /dev/input/event2: 0003 0000 00000007
287-237335: /dev/input/event2: 0003 0001 fffffff8
287-237335: /dev/input/event2: 0003 0002 00000185
287-237335: /dev/input/event2: 0000 0000 00000000
287-280364: /dev/input/event2: 0003 0000 000000ed
287-280395: /dev/input/event2: 0003 0001 0000000e
287-280395: /dev/input/event2: 0000 0000 00000000
287-323516: /dev/input/event2: 0003 0000 000000ef
287-323547: /dev/input/event2: 0003 0001 0000000f
287-323547: /dev/input/event2: 0003 0002 00000183
287-323547: /dev/input/event2: 0000 0000 00000000
287-366516: /dev/input/event2: 0003 0001 0000014c
287-366577: /dev/input/event2: 0003 0002 00000185
287-366577: /dev/input/event2: 0000 0000 00000000
287-409729: /dev/input/event2: 0003 0001 0000000f
287-409729: /dev/input/event2: 0003 0002 00000186
287-409729: /dev/input/event2: 0000 0000 00000000
287-452819: /dev/input/event2: 0003 0000 0000010a
287-452850: /dev/input/event2: 0003 0001 0000002e
287-452850: /dev/input/event2: 0003 0002 0000018c
287-452850: /dev/input/event2: 0000 0000 00000000
IV. CONCLUSION
In this paper, we describe that how the data from an external device is handled. To get the data from the external device, we connect the accelerometer to the Panda board. We implement the driver program for the input device. By using this method it takes short time to complete input event.

V. REFERENCES

