CMPC Debian 7.1
Referenced Development Kit
Design & Implementation

Disclaimer and Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/#/en_US_01

This document is Intel Confidential and must be recorded in an Intel Confidential Information Transmittal Record.

INTEL EMPG SW Release Team

Revision History

Version	Date	Contributor	Comments
0.6	2014-01-09	Zhu Kaiyue	Draft
1.0	2014-02-25	Zhu Kaiyue	Update

Definitions

Term	Definition
CMPC	Classmate PC
ISV	Independent Software Vendor
PDT	Product Development Team
PRD	Product Requirements Document
SAS	Software Architecture Specification
SFS	Software Functional Specification
HLD	High Level Design

Table of Contents

1 Purpose

The purpose of this document is to introduce the design and the implementation of the function key package on Intel-powered classmate PC platform.
OSVs could integrate these two CMPC packages into their Debian 7.1 compliant OS
distro with minimum integration efforts.

2 Scope

This document covers the software architecture, main functions, design and implementation. Anyone interested in function key design is recommended to
read this document.

The evaluated hardware platform of these packages is MarblePoint. The source code and the test report are attached.

3   CMPC Introduction

Intel-Powered Classmate PC

As part of the Intel World Ahead Program, Intel aims to transform and improve education Worldwide by providing individual, complete, and uncompromised education solutions that meet the particular needs of people and governments in the world's developing countries. While the Intel powered classmate PCs are designed for students in emerging Market, Intel also understands and addresses the particular needs, goals, and aspirations of each and every player of education systems worldwide – namely, children, teachers, parents, and policy makers.

Intel powered classmate PCs are small, mobile education-oriented PC to be used in classrooms in emerging markets. The fully-functional PC is designed to provide affordable, collaborative learning environments for students K-12 and their teachers.

Intel-powered classmate PCs are:
· "Rugged" mobile educational solutions
 o Durable, rugged design for day-to-day use by children, almost "kid-proof"
 o Small-form-factor that integrates easily into the classroom

o Easy to carry, light weight, a personal learning assistant that looks like a textbook

o Include theft control

o Operate in wireless ad-hoc and infrastructure networks

o Allow students to collaborate, exchange information and review e-learning material

o Allow teachers to monitor classroom activity, supplement and extend their lectures with interactive material

o Allow parents to track their child's progress in school and facilitates parent-teacher communication

· Personalized learning mobile PCs:

o Education-specific feature set

o Integrated software and hardware solution, allows classroom/learning/content management

o Brings learning through fun, collaboration and interaction

· Easy-to-deploy

o Based on a full x86 processor, runs already available content, applications and operating systems with full compatibility with standard PC ecosystem to prepare students to use technology that is prevalent today

o Minimum IT infrastructure required using WLAN and ad-hoc networks

4 Function Key Package Architecture

4.1 System Architecture
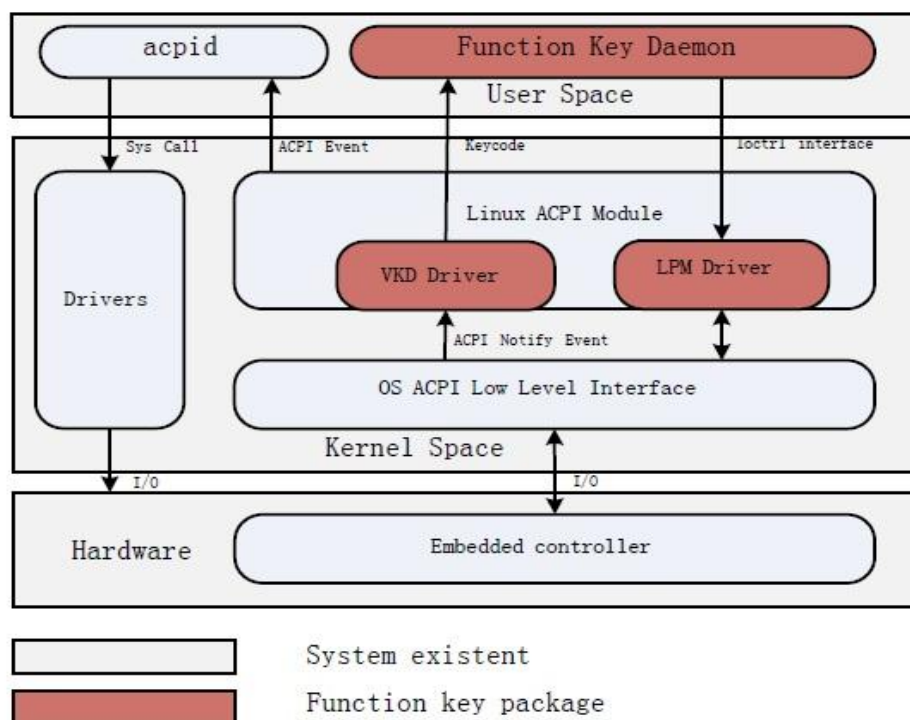
System Diagram

4.2 Function List

Key Combination	Summit Peak	Implemented by
Fn+F1	Turn on/off WLAN	package
Fn+F2	Suspend	OS
Fn+F3	Mute/Unmute volume	OS
Fn+F4	Volume down	OS
Fn+F5	Volume up	OS
Fn+F6	Display switcher	package
Fn+F7	Brightness down	OS
Fn+F8	Brightness up	OS

4.3 Components

User space	Fnkey Daemon
Kernel space	VKD Driver

Module	Location	Description
VKD	/proc/acpi/cmpc_vkd	Fetch ACPI event
Fnkey	/usr/bin/fnkey	Function key daemon

5 VKD Driver

5.1 Design
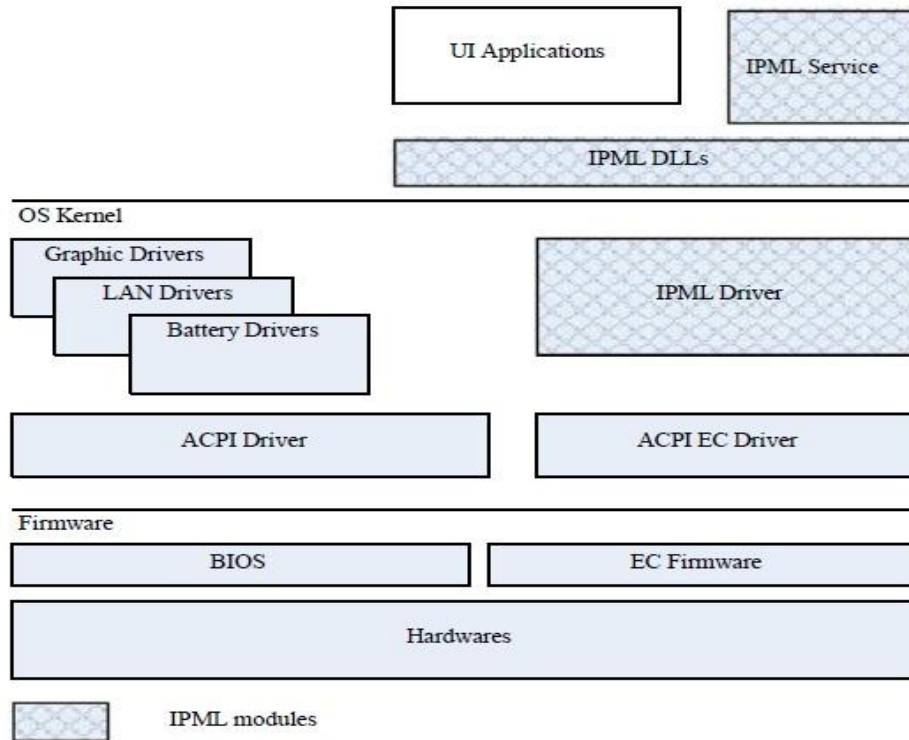
A new device called vkd is introduced to fetch Fn+Fx, bezel buttons clicking events.
Function key daemon could control WLAN on/off, photo taken, according to these events.
A waiting queue is introduced for function key event reader. Once a key event occurs,
reader wakes up from that waiting queue.

Working Process
1. Register a char device called /dev/vkd;

2. Register a handler to forward key events;

3. Store key event into a FIFO and be ready to be fetched;

Policy

 /dev/vkd is +r.

:

5.2 Implementation

The vkd driver has the following main functions:
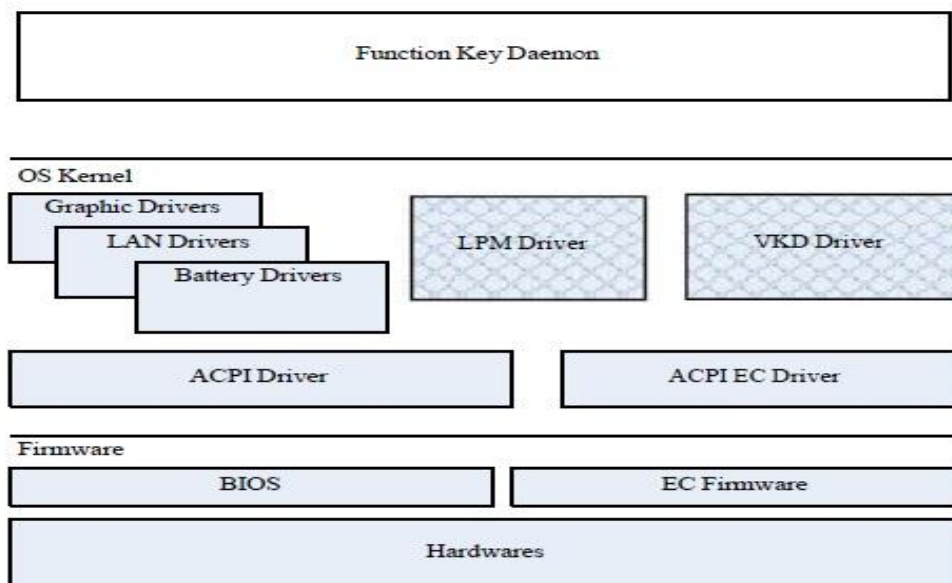
- Read()
  - Fetch key events from a FIFO
  - Return events to user space
- Write()
  - NULL
- Add()
  - Create /proc/vkd entry
  - Register ACPI event handler

6 Function Key Daemon

6.1 Design

A daemon running as a background process is designed to listen to specific devices and function key strokes.

SW Architecture

Working Process

1. Create one thread as a daemon.

2. Block read on function key stroke.

3. Analysis strokes.

   a) OSD drawing;

   b) WLAN on/off by using rfkill system utility;

   c) dual-monitor control.

6.2 Implementation

Main()

- Create one thread as a daemon

- Gtk main loop

VkdEventHandler()

- Block read on vkd device
- Analysis function key stroks
- Do OSD/WLAN/Dual-monitor

7 UI Description

7.1 Wireless On/Off

Wireless on   **wireless off**

7.2 Dual-monitor Display

Display switch screenshot