

# 飞腾平台 Embedded Controller 接口规范

(Version 1.4)

2020 年 12 月

天津飞腾信息技术有限公司

[www.phytium.com.cn](http://www.phytium.com.cn)

版权所有© 天津飞腾信息技术有限公司 2020

此文档用于指导用户的相关应用和开发工作。天津飞腾信息技术有限公司对此文档内容拥有版权，并受法律保护

免责声明© 天津飞腾信息技术有限公司对本文档内容有解释权，且保留持续修改的权利



## 当前版本

文件标识	<b>P-U-SW-EC</b>
当前版本	<b>1.4</b>
作者	
完成日期	2020.12.1

## 版本历史

版本	修订时间	修订章节	修订内容
1.0	2019.12.12		第一个正式发布的版本
1.1	2019.12.31		EC 事件中断源从 LPC 改为 GPIO;
1.2	2020.03.03	4.2.4; 4.2.10; 4.3;	增加对“S3 超时时间”的操作说明; 修正“切换显示屏”键的通码; 增加“键盘灯开关”、“麦克风开关”、“摄像头开关”等键盘按键的扫描码; 增加 4.2.10 一节 CPU 内置温度传感器
1.3	2020.04.10	4.1	修正表 4-1 中 EC_SC、EC_DATA 偏移值错误, 修正表 4-2 中“ec_battery_charge_h8”拼写错误
1.4	2020.12.1	4.1; 4.2.9; 4.2.10; 4.3; 4.5; 4.6; 5	4.1 增加 ec_lcd_event_info; 增加 4.2.9.3 说明 ec_lcd_event_info; 4.2.10 增加内置温度传感器正负值说明; 4.3 增加“锁屏”扫描码, 删除“音量+”、“音量-”, 保持兼容第二套扫描码, 不再单独添加已有的扫描码。删除“触摸板开”和“触摸板关”扫描码, 统一使用“触摸板开关”扫描码。增加按键处理说明; 4.5 合盖事件修改为开合盖时间, 新增两个保留事件; 增加 4.6 章节; 新增 5 章节

## 目录

<b>1</b>	<b>范畴</b> .....	<b>1</b>
<b>2</b>	<b>定义与缩写</b> .....	<b>2</b>
2.1	定义.....	2
2.2	缩写.....	2
<b>3</b>	<b>参考文献</b> .....	<b>3</b>
<b>4</b>	<b>飞腾平台 EC 规范</b> .....	<b>4</b>
4.1	EC 寄存器及资源偏移 .....	4
4.2	EC 资源说明 .....	6
4.2.1	EC 软件版本号 .....	6
4.2.2	系统电源控制 .....	6
4.2.3	系统复位模式.....	7
4.2.4	设置 S3 超时时间.....	7
4.2.5	电源存在状态.....	8
4.2.6	电池.....	8
4.2.7	板载温度传感器.....	9
4.2.8	风扇.....	10
4.2.9	显示屏.....	10
4.2.10	CPU 内置温度传感器 .....	11
4.3	键盘控制器.....	11
4.4	基于 LPC 的 EC 中断 .....	12
4.5	EC 事件 .....	12
4.6	EC 综合服务 .....	13
4.6.1	开关屏服务.....	14
<b>5</b>	<b>EC 的 ACPI 描述</b> .....	<b>15</b>

## 表目录

表 4-1	EC 寄存器 .....	4
表 4-2	EC 资源偏移表 .....	5
表 4-3	自定义键盘扫描码 .....	12
表 4-4	LPC EC 中断源 .....	12
表 4-5	EC 事件 GPIO 中断 .....	13
表 4-6	EC 事件 .....	13
表 4-7	综合服务接口 .....	13
表 4-8	服务功能表 .....	14

## 1 范畴

此文档适用于基于飞腾处理器的系统平台，为 EC 软件设计提供参考。

PHYTIUM

## 2 定义与缩写

### 2.1 定义

### 2.2 缩写

EC            Embedded Controller

PHYTIUM

### 3 参考文献

- [1] UEFI, Advanced Configuration and Power Interface Specification Version 6.3.
- [2] 飞腾, 飞腾系统 ACPI 描述规范, v1. 1.
- [3] Keyboard Scan Codes: Set 2
- [4] 飞腾, FT-2000/4 软件编程手册, v1.4.

PHYTIUM



## 4 飞腾平台 EC 规范

基于飞腾处理器的平台，比如，飞腾笔记本，可能采用嵌入式控制器（Embedded Controller, EC）来实现系统电源控制，以及键盘、触摸板、电池、传感器等外设的管理。EC 连接到处理器的 LPC 总线上，EC 下连接了键盘、触摸板、电池、温度传感器等设备。

### 4.1 EC 寄存器及资源偏移

系统软件，固件或操作系统，通过 EC 寄存器访问 EC 下的资源。EC 包含 3 个寄存器：状态寄存器 EC\_SC、命令寄存器 EC\_SC 和数据寄存器 EC\_DATA。其中，状态寄存器和命令寄存器是同一个寄存器，遵循 ACPI EC 规范<sup>[1]</sup>。ARM64 体系结构没有专门的 IO 指令，不能以端口（port）访问的方式操作 EC 寄存器。EC 寄存器以内存地址映射（memory mapped）的方式访问，访存地址为总线基址（比如，LPC 的基址）+ 偏移。其中，偏移与传统端口号相对应，EC\_SC 和 EC\_DATA 的偏移（端口）如表 4-1 所示。固件可以通过 ACPI 表，告知操作系统这两个寄存器的偏移，对应的 ACPI 属性名为 ec\_command\_reg、ec\_data\_reg 描述。ACPI 描述见 4.5 节。EC 还可能包含键盘控制器，对应的寄存器为 KBC\_SC 和 KBC\_DATA。

EC 寄存器端口分配如表 4-1 所示。

表 4-1 EC 寄存器

寄存器	ACPI 属性名	偏移/端口	R/W	说明
EC_SC	ec_command_reg	0x66	R	EC 状态寄存器
EC_SC	ec_command_reg	0x66	W	EC 命令寄存器
EC_DATA	ec_data_reg	0x62	R/W	EC 数据寄存器
KBC_SC	kbc_command_reg	0x64	R	键盘控制器状态寄存器
KBC_SC	kbc_command_reg	0x64	W	键盘控制器命令寄存器
KBC_DATA	kbc_data_reg	0x60	R/W	键盘控制器数据寄存器

系统按照 ACPI 规范，通过 EC 寄存器，向 EC 发送读(RD\_EC)、写(WR\_EC)

命令，访问 EC 资源。EC 提供 256 字节的可被系统读写的 RAM 空间，EC 资源在该 RAM 空间映射，通过访问对应偏移（0x00~0xFF）即可操作对应资源。资源偏移如表 4-2 所示。

表 4-2 EC 资源偏移表

ACPI 属性名	偏移	方向	描述
EC 软件描述			
ec_version_minor	0x00	读	EC 软件次版本号
ec_version_senior	0x01	读	EC 软件主版本号
系统电源管理			
ec_power_ctrl	0x0A	写	电源控制
ec_reset_mode	0x0B	读	加电原因
ec_s3_timeout_l8	0x0C	写	S3 状态的超时时间，低 8 位
ec_s3_timeout_h8	0x0D	写	S3 状态的超时时间，高 8 位
电池信息			
ec_battery_remain_percent	0x21	读	电池剩余电量百分比
ec_battery_charge_l8	0x24	读	电池充满电量，低 8 位
ec_battery_charge_h8	0x25	读	电池充满电量，高 8 位
ec_battery_remain_l8	0x26	读	电池剩余电量，低 8 位
ec_battery_remain_h8	0x27	读	电池剩余电量，高 8 位
ec_battery_temp_l8	0x28	读	电池温度，低 8 位
ec_battery_temp_h8	0x29	读	电池温度，高 8 位
ec_battery_voltage_l8	0x2A	读	电池电压，低 8 位
ec_battery_voltage_h8	0x2B	读	电池电压，高 8 位
ec_battery_current_l8	0x2C	读	电池电流，低 8 位
ec_battery_current_h8	0x2D	读	电池电流，高 8 位
ec_battery_avg_current_l8	0x2E	读	电池平均电流，低 8 位
ec_battery_avg_current_h8	0x2F	读	电池平均电流，高 8 位
ec_battery_design_charge_l8	0x38	读	电池设计充满电量，低 8 位
ec_battery_design_charge_h8	0x39	读	电池设计充满电量，高 8 位

ec_battery_design_voltage_l8	0x3A	读	电池设计电压, 低 8 位
ec_battery_design_voltage_h8	0x3B	读	电池设计电压, 高 8 位
ec_battery_status	0x3C	读	电池状态
ec_battery_sn_l8	0x3E	读	电池序列号, 低 8 位
ec_battery_sn_h8	0x3F	读	电池序列号, 高 8 位
ec_power_status	0xB0	读	电源存在状态
传感器			
ec_cpu_temp	0x98	读	CPU 温度
ec_gpu_temp	0x9A	读	GPU 温度
ec_soc_temp_0	0xA0	写	CPU 内置温度传感器 0
ec_soc_temp_1	0xA1	写	CPU 内置温度传感器 1
风扇			
ec_cpu_fan_speed_l8	0xF2	读	CPU 风扇转速, 低 8 位
ec_cpu_fan_speed_h8	0xF3	读	CPU 风扇转速, 高 8 位
ec_gpu_fan_speed_l8	0xF4	读	GPU 风扇转速, 低 8 位
ec_gpu_fan_speed_h8	0xF5	读	GPU 风扇转速, 高 8 位
显示屏			
ec_lcd_brightness_ctrl	0x0E	写	设置目标亮度
ec_lcd_brightness_status	0x0F	读	查询当前亮度
ec_lcd_event_info	0x46	读	显示屏事件详细信息

## 4.2 EC 资源说明

### 4.2.1 EC 软件版本号

输入参数	读 EC-RAM 地址: 0x01,0x00	
返回值	16bit	EC 软件的版本号: 偏移 0x01 为主版本号; 偏移 0x00 为次版本号

### 4.2.2 系统电源控制

写该地址，对应位写 1，控制系统电源。

输入参数	写 EC-RAM 地址: 0x0A	
设置值	8bit	Bit3: 重启 Bit2: 关机 Bit1: S4 Bit0: S3 挂起

#### 4.2.3 系统复位模式

读，判断加电原因

输入参数	读 EC-RAM 地址: 0x0B	
返回值	8bit	0x55: 从 S3 正常唤醒; 0xAA: 从 S3 超时唤醒; 其他: 正常上电启动

#### 4.2.4 设置 S3 超时时间

可以设置系统在 S3 的持续时间，以分钟为单位。如果用户设置了这个值，当进入 S3 后，EC 会启动超时定时器，定时器超时后，EC 会自动唤醒系统，并置系统复位模式值为 0xAA。如果要取消定时唤醒机制，需要将超时时间设置为 0。

系统开机（包括从 S4 恢复）、重启时，EC 负责将该域初始化为上电缺省值 0，即永不超时。该域由 CPU 端系统软件（比如，操作系统）负责配置。除了上述的初始化外，EC 不修改该域值，并负责保持 CPU 端系统软件对该域的配置不丢失，包括 S3 挂起、恢复。

输入参数	写 EC-RAM 地址: 0x0D,0x0C	
返回值	16bit	无符号整数，持续处于 S3 状态的超时时间，以分钟为单位。 0: 永不超时。

### 4.2.5 电源存在状态

输入参数	读 EC-RAM 地址: 0xB0	
返回值	8bit	Bit 0: 电源适配器存在状态: 1: 插入, 0: 断开 Bit 1: 电池存在状态: 1: 插入, 0: 断开

### 4.2.6 电池

#### 4.2.6.1 电池状态

输入参数	读 EC-RAM 地址: 0x3C	
返回值	8bit	Bit 6: 是否在放电 0=充电状态; 1=放电状态 Bit 5: 是否充满: 0=电池还没有充满; 1=电池充满 Bit 4: 是否耗完: 0=电池电量没有耗完; 1=电池电量耗完

#### 4.2.6.2 电池剩余百分比电量

输入参数	读 EC-RAM 地址: 0x21	
返回值	8bit	电池剩余百分比电量%

#### 4.2.6.3 电池充满电量

输入参数	读 EC-RAM 地址: 0x25,0x24	
返回值	16bit	电池充满电量, 单位为 mAh

#### 4.2.6.4 剩余电量

输入参数	读 EC-RAM 地址: 0x27,0x26	
返回值	16bit	剩余电量, 单位为 mAh

#### 4.2.6.5 电池温度

输入参数	读 EC-RAM 地址: 0x29,0x28	
返回值	16bit	电池温度, 单位为: 0.1 °C

## 4.2.6.6 电池电压

输入参数	读 EC-RAM 地址: 0x2B,0x2A	
返回值	16bit	电池电压, 单位为 mV

## 4.2.6.7 电池电流

输入参数	读 EC-RAM 地址: 0x2D,0x2C	
返回值	16bit	电池当前电流, 充电时为+, 放电时为-, 单位为 mA, 值为有符号整数。

## 4.2.6.8 电池平均电流

输入参数	读 EC-RAM 地址: 0x2F,0x2E	
返回值	16bit	电池平均电流, 充电时为+, 放电时为-, 单位为 mA, 值为有符号整数。

## 4.2.6.9 电池设计充满电量

输入参数	读 EC-RAM 地址: 0x39,0x38	
返回值	16bit	电池设计充满电量, 单位为 mAh

## 4.2.6.10 电池设计电压

输入参数	读 EC-RAM 地址: 0x3B,0x3A	
返回值	16bit	电池设计电压, 单位为 mV

## 4.2.6.11 电池序列号

输入参数	读 EC-RAM 地址: 0x3F,0x3E	
返回值	16bit	电池的序列号

## 4.2.7 板载温度传感器

## 4.2.7.1 CPU 温度

输入参数	读 EC-RAM 地址: 0x98	
返回值	8bit	CPU 温度, 单位为°C

## 4.2.7.2 GPU 温度

输入参数	读 EC-RAM 地址: 0x9A	
------	-------------------	--

返回值	8bit	GPU 温度，单位°C
-----	------	-------------

## 4.2.8 风扇

### 4.2.8.1 CPU 风扇转速

输入参数	读 EC-RAM 地址: 0xF3,0xF2	
返回值	16bit	CPU 风扇转速，单位为: r/min

### 4.2.8.2 GPU 风扇转速

输入参数	读 EC-RAM 地址: 0xF5,0xF4	
返回值	16bit	GPU 风扇转速，单位为: r/min

## 4.2.9 显示屏

### 4.2.9.1 设置当前亮度

以显示屏支持的最大亮度为基准 100，将亮度可调节范围平均等分，取值范围为 0~100。

输入参数	写 EC-RAM 地址: 0x0E	
设置值	8bit	设置目标亮度，单位为显示屏支持的最高亮度的百分比，取值范围为: 0~100。

### 4.2.9.2 查询屏幕亮度

输入参数	读 EC-RAM 地址: 0x0F	
返回值	8bit	查询当前亮度，单位为显示屏支持的最高亮度的百分比，取值范围为: 0~100。

### 4.2.9.3 查询显示屏事件详细信息

输入参数	读 EC-RAM 地址: 0x46	
返回值	8bit	显示屏相关事件，比如，开合盖事件，产生后，查询事件的详细信息。 Bit 0: 0: 开盖; 1: 合盖。

		Bit 1-7: 保留
--	--	-------------

注：如具体实现与规范不一致，可以在 ACPI 表中，自行修改字段定义。

查询显示屏事件详细信息的 ACPI 字段描述参考如下：

Package (0x02)

```
{
    "ec_lcd_event_info", /* acpi 属性名 */
```

Package (0x04)

```
{
    0x46, /*ec_lcd_event_info 的 偏移地址 */
    0, /* 开合盖状态所处位域, bit0 位。 */
    0, /* 开盖状态值*/
    1 /* 合盖状态值*/
}
}
```

#### 4.2.10 CPU 内置温度传感器

飞腾处理器内置多个温度传感器，处理器带内访问这些温度传感器。处理器端的系统软件（比如，BIOS、操作系统）负责将这些温度传感器中的温度值写入 EC 对应域。EC 从该域读取对应温度传感器的值，其中正温度最高位 bit7 为 1，负温度高位 bit7 为 0。

##### 4.2.10.1 CPU 内置温度 0

输入参数	写 EC-RAM 地址：0xA0	
返回值	8bit	飞腾处理器内置温度传感器 0 的温度，单位为°C

##### 4.2.10.2 CPU 内置温度 1

输入参数	读 EC-RAM 地址：0xA1	
返回值	8bit	飞腾处理器内置温度传感器 1 的温度，单位为°C

### 4.3 键盘控制器

EC 下挂键盘控制器兼容第二套键盘（AT Keyboard）扫描码<sup>[3]</sup>，本文只描述



第二套键盘扫描码未定义的功能扫描码。

表 4-3 自定义键盘扫描码

KEY	ACPI 属性名	Make (通码)	Break (断码)
屏幕背光-	kbc_backlight_down	E0,01	E0,F0,01
屏幕背光+	kbc_backlight_up	E0,02	E0,F0,02
触摸板开关	kbc_touchpad_switch	E0,03	E0,F0,03
wifi 开关	kbc_wifi_switch	E0,04	E0,F0,04
屏幕开关	kbc_lcd_switch	E0,07	E0,F0,07
切换显示屏	kbc_switch_video_mode	E0,08	E0,F0,08
键盘灯开关	kbc_kb_light_switch	E0,09	E0,F0,09
麦克风开关	kbc_mic_switch	E0,0A	E0,F0,0A
摄像头开关	kbc_camara_switch	E0,0B	E0,F0,0B
锁屏	kbc_screen_lock	E0,7F	E0,F0,7F

EC 检测到按键动作后，不作按键的功能性处理，直接将对应的键值报告给 CPU。CPU 端的系统软件根据收到键值，作相应的处理，处理过程可能需要调用 EC 提供的服务。

#### 4.4 基于 LPC 的 EC 中断

如果 EC 挂在处理器的 LPC 总线下，EC 下的设备，比如触摸板、键盘等，通过 LPC 向处理器上报中断。LPC 中断处理器程序查询 LPC 中断状态寄存器来确定中断源，再进一步处理中断。

表 4-4 LPC EC 中断源

中断源	LPC 中断源编码
触摸板	12
键盘	1

#### 4.5 EC 事件

在飞腾平台上，EC 通过 GPIO 中断向处理器汇报 EC 事件。当处理器收到

GPIO 中断后，查询 GPIO 中断状态寄存器<sup>[4]</sup>，确定中断源是 EC 事件。实现 EC 事件的 GPIO 中断与飞腾处理器平台相关<sup>[4]</sup>。

表 4-5 EC 事件 GPIO 中断

平台	GPIO 端口	中断号
FT-2000/4	gpio0_porta_7	42

EC 事件的上报，遵循 ACPI 规范<sup>[1]</sup>。处理器收到 EC 事件中中断后，通过 EC 事件查询命令查询触发 EC 事件的原因。EC 事件原因以及事件编码如下表所示。

表 4-6 EC 事件

事件	事件号	ACPI 属性名
电池属性值更新	0xB3	ec_battery_q_event
电源适配器或电池的插拔	0xB4	ec_power_q_event
短按电源键	0xB5	ec_pwbutton_q_event
开合盖	0xD0	ec_lcd_q_event
保留事件	0xD1	
保留事件	0xD2	

## 4.6 EC 综合服务

EC 可以向 CPU 端的系统软件提供各种综合服务。CPU 端系统软件调用 EC 服务的接口如表 4-7 所示。

CPU 端调用服务前，先根据具体服务要求，填写相关服务调用参数，并将状态（Status）置为 Busy，然后再写服务功能号为对应的服务编码，发起相应的服务。之后，CPU 端查询 Status 域，等待 EC 完成相应的服务。

EC 完成相应的服务后，将相关的执行结果写入返回值，并更新 Status、清除调用参数和服务功能号。

表 4-7 综合服务接口

名称	RAM 地址	描述
Function ID	0x10	服务功能号
Arg1 / Ret1	0x11	调用参数 1 / 返回值 1

Arg2 / Ret3	0x12	调用参数 2 /返回值 2
Arg3 / Ret3	0x13	调用参数 3 / 返回值 3
Status	0x14	状态： 0xFF: Busy; 0x00: 成功; 0x01: 失败; 0x02: 不支持; 0x03: 无效参数; 其它: 保留。

EC 支持的综合服务功能表见表 4-8。

表 4-8 服务功能表

Function ID	说明
0x01	开关屏

#### 4.6.1 开关屏服务

开关屏服务的功能是开启/关闭显示屏。

调用接口如下：

Function ID = 1 (开关屏)		
调用 参数	Arg1	0: 开屏 1: 关屏
	Arg2	无
	Arg3	无
返回 值	Ret1	无
	Ret2	无
	Ret3	无

## 5 EC 的 ACPI 描述

如果 EC 的具体实现与本规范不一致，比如某个 ACPI 属性名对应的偏移与本规范不同，则可以在 ACPI 表中，增加对应的 ACPI \_DSD 属性描述，将属性赋值为实际的实现值。操作系统，可以基于该 ACPI 属性描述，获得 EC 的具体实现。

关于 EC 的 ACPI 描述，参考《飞腾系统 ACPI 描述规范》<sup>[2]</sup>。

PHYTIUM