

OpenHarmonyOS 基础外设开发--EEPROM



目 录

CONTENTS

- [01] 什么是EEPROM
- [02] EEPROM控制原理
- [03] EEPROM相关接口
- [04] 如何控制EEPROM

01

什么是EEPROM

EEPROM是“Electrically Erasable Programmable Read-only”（可擦写可编程只读存储器）的缩写，EEPROM在正常情况下和EPROM一样，**可以在掉电的情况下保存数据**，所不同的是它可以在特定引脚上施加特定电压或使用特定的总线擦写命令就可以在在线的情况下方便完成**数据的擦除和写入**，这使EEPROM被用于广阔的的消费者范围。

02

EEPROM控制原理

在本节，我们以最常见的I2C（I2C也叫IIC）接口的EEPROM芯片K24C02为例进行学习研究。它采用PHILIPS公司开发的两线式串行总线（I2C总线）。

（1）EEPROM型号：K24C02，容量为 $256 * 8\text{bits} = 2\text{K bits} = 256 \text{ Byte}$ 。

（2）I2C从设备地址：0x51

（3）读操作

- ❑ 当前地址读数据（Current Address Read）
- ❑ 随机读数据（Random Read）
- ❑ 连续读数据（Sequential Read）

（4）写操作

02

EEPROM控制原理

(4) 写操作

- 字节写操作 (Byte Write)
- 页写操作 (Page Write)

03

EEPROM相关接口

EEPROM相关接口的头文件

/vendor/lockzhiner/rk2206/samples/b3_eeprom/include/**eeprom.h**

OpenHarmonyOS基础外设开发中，EEPROM相关接口有很多，主要分为几大类：

- (1) 初始化、销毁EEPROM；
- (2) 获取页大小；
- (3) 读操作；
- (4) 写操作。

03

EEPROM 相关接口

功能分类	接口名	功能描述
初始化、销毁EEPROM	eeeprom_init	EEPROM初始化
	eeeprom_deinit	EEPROM注销
获取页大小	eeeprom_get_blocksize	获取EEPROM页大小
读操作	eeeprom_read	控制EEPROM读多个字节
写操作	eeeprom_write	控制EEPROM写多个字节

03

EEPROM 相关接口

```
unsigned int eeprom_init();
```

该函数主要功能是EEPROM初始化。

返回0为成功，其余为失败。

```
unsigned int eeprom_deinit();
```

该函数主要功能是EEPROM注销。

返回0为成功，其余为失败。

03

EEPROM 相关接口

```
unsigned int eeprom_get_blocksize();
```

该函数主要功能是获取EEPROM页大小。

返回EEPROM页大小。

03

EEPROM 相关接口

```
unsigned int eeprom_read(unsigned int addr,  
    unsigned char *data,  
    unsigned int data_len);
```

该函数主要功能是控制EEPROM读多个字节。

- * 参数addr: EEPROM存储地址
- * 参数data: 存放EEPROM的数据指针
- * 参数data_len: 读取EEPROM数据的长度。

返回写入字节的长度，反之为错误。

03

EEPROM 相关接口

```
unsigned int eeprom_write(unsigned int addr,  
                           unsigned char *data,  
                           unsigned int data_len);
```

该函数主要功能是控制EEPROM写多个字节。

- * 参数addr: EEPROM存储地址
- * 参数data: 需要写入EEPROM的数据指针
- * 参数data_len: 需要写入EEPROM数据的长度。

返回写入字节的长度，反之为错误。

04

如何控制EEPROM

1、打开sdk下面路径的文件

```
vendor/lockzhiner/rk2206/samples/b3_eeprom/eeprom_example.c
```

2、创建任务

在eeprom_example函数中，通过LOS_TaskCreate函数创建eeprom_proress任务。

```
task.pfnTaskEntry = (TSK_ENTRY_FUNC)eeprom_proress;
```

```
task.uwStackSize = 2048;
```

```
task.pcName = "eeprom process";
```

```
task.usTaskPrio = 24;
```

```
ret = LOS_TaskCreate(&thread_id, &task);
```

04

如何控制EEPROM

eeeprom_proress函数对EEPROM进行初始化

```
eeeprom_init();
```

eeeprom_proress函数每5秒执行1次EEPROM写、1次EEPROM读。其中，EEPROM地址每次都累加32个字节，写入内容每次都有变化。

```
/* 写EEPROM */
```

```
ret = eeeprom_write(addr_offset, buffer, FOR_CHAR);
```

```
/* 读EEPROM */
```

```
ret = eeeprom_read(addr_offset, buffer, FOR_CHAR);
```

3、修改编译脚本

修改 `vendor/lockzhiner/rk2206/sample` 路径下 `BUILD.gn` 文件，指定 `b3_eeprom` 参与编译。

```
"/b3_eeprom:eeprom_example",
```

修改 `device/lockzhiner/rk2206/sdk_liteos` 路径下 `Makefile` 文件，添加 `-leeprom_example` 参与编译。

```
hardware_LIBS = -lhal_iohardware -lhardware -leeprom_example
```

4、编译固件

```
hb set -root .
```

```
hb set
```

```
hb build -f
```

5、烧写固件

6、通过串口查看结果

运行结果

```
***** Eeprom Process *****
```

```
BlockSize = 0x8
```

```
Write Byte: 3 = !
```

```
Write Byte: 4 = "
```

```
Write Byte: 5 = #
```

```
Write Byte: 6 = $
```

```
.....
```

谢谢聆听

单击此处添加副标题内容