



OPEN AI LAB

EAI610 可编程接口板应用指南

2018-11-19

OPEN AI LAB

变更记录 (Reversion Record)

日期 (Date)	版本 (Rev)	说明 (Change Description)	作者 (Author)
2018-10-24	V0.1	初稿，基于瑞芯微提供材料	路明
2018-11-19	V0.2	根据会议Review意见定稿	路明

目录(catalog)

1 前言	3
1.1 目的	3
1.2 术语	3
2 接口板测试说明	3
2.1 EXTIO_TEST 测试规范	3
2.2 EXTIO_TEST 编译及测试方法	4
2.3 EXTIO_TEST 主程序说明.....	5
2.4 EXTIO_TEST 相关驱动代码说明.....	6

1 前言

1.1 目的

本文档用于描述 EAI610 的嵌入式可编程接口板的应用测试方案

1.2 术语

- **I2S**: Inter IC Sound 总线, 又称集成电路内置音频总线, 是飞利浦公司为数字音频设备之间的音频数据传输而制定的一种总线标准, 该总线专门用于音频设备之间的数据传输, 广泛应用于各种多媒体系统。
- **I2C**: Inter-Integrated Circuit, I2C 总线是由 Philips 公司开发的一种简单、双向二线制同步串行总线。它只需要两根线即可在连接于总线上的器件之间传送信息。
- **SPI**: Serial Peripheral Interface, 串行外设接口, 是一种高速的, 全双工, 同步的通信总线, 占用四根信号线。
- **GPIO**: General Purpose Input Output, 通用目的输入输出, 又被称为总线扩展器。当微控制器或芯片组没有足够的 I/O 端口, 或当系统需要采用远端串行通信或控制时, GPIO 产品能够提供额外的控制和监视功能。
- **DTS**: Device Tree Source, 设备树源码。Device Tree 是一种描述硬件的数据结构, 它起源于 OpenFirmware (OF)。DTS 文件用于描述一个特定平台。

2 接口板测试说明

EAI610 提供针对接口板的测试样例代码 EXTIO_TEST。

2.1 EXTIO_TEST 测试规范

1. SPI_FLASH 数据读写对比, 比对成功, 点亮数码管。
2. 数码管可以显示 0~9, 还可以做定时器显示时间实验。
3. GPIO 按键, 做 GPIO 中断编程实验, 中断函数可以处理开始 SPI_FLASH 实验和电压采集实验。

4. ADC 输入实验，外接按键，对应点亮一个 LED 灯。
5. 3 个 I2C 通道，检测 3 路电压，检测正常，数码管显示。

2.2 EXTIO_TEST 编译及测试方法





1. 在源码目录下执行 make 命令，生成 extio_test 命令。
2. 插上 ExtIO 小板。
3. ADC 输入实验，按下 ADC 按键，LED 灯亮，松开按键，LED 灯灭。
4. 务必用 root 权限运行 extio_test 命令，进行 GPIO 中断编程实验

```
sudo ./extio_test
```

5. GPIO 按键有两个，分别为 GPIO_KEY1 和 GPIO_KEY2:

- 1) GPIO_KEY1 对应处理 SPI-Flash 数据比对实验。
- 2) GPIO_KEY2 对应处理电压采集实验。

6. 实验结果可直接由数码管显示:

- 1) : 按下其中任意一个按键，开始处理对应实验，若成功则显示
- 2) : 再次按下剩下一个按键，开始处理另一个对应实验，若成功则显示
- 3) : SPI-Flash 数据比对实验若失败则显示
- 4) : 电压采集实验若失败则显示

7. 若实验结果成功，串口输出结果如下:

```
## Please press GPIO_KEY1 or GPIO_KEY2 within 10 seconds!
----- Start Flash Test -----
## Flash Data Comparison is OK!!
----- Flash Test is Successful -----
----- Start Voltage Test -----
## Reference 5000mv, Read 4992mv is valid.
## Reference 3300mv, Read 3316mv is valid.
## Reference 12000mv, Read 12056mv is valid.
## Voltage Detect is OK!!
----- Voltage Test is Successful -----
----- ExtIO Test is Passed -----
```

2.3 EXTIO_TEST 主程序说明

1. `main()`: 设置 GPIO 中断触发方式, 当触发中断时, 根据对应 GPIO_KEY, 执行对应实验。

- 设置 GPIO 中断触发模式为下降沿触发, 先读取 GPIO 引脚的值。

```
1) edge_fd[i] = open("/sys/class/gpio/gpio%d/edge", O_RDWR);
2) write(edge_fd[i], "falling", sizeof("falling"));
3) gpio_fd[i] = open("/sys/class/gpio/gpio%d/value", O_RDONLY);
4) read(gpio_fd[i], buff[n], 10);
```

- 死循环内使用 `poll()` 函数监听 GPIO, 当触发中断时, 处理对应函数。

```
1) poll(fds, 2, 10000); //如果超过 10 秒未产生 GPIO 中断, 退出测试, 并返回失败结果
2) 当 fds[i].revents & POLLPRI 为真, 表示有 GPIO_KEY 按下, 产生中断。
3) lseek(gpio_fd[i], 0, SEEK_SET); //重置文件指针指向开头
4) read(gpio_fd[i], buff[n], 10); //重新读取 GPIO 引脚的值
5) 当 GPIO_KEY1 按下, 执行 flash_data_comparison_test 函数
6) 当 GPIO_KEY2 按下, 执行 voltage_detect 函数
7) 任意一个实验失败, 即退出循环, 返回失败结果, 否则两次实验均成功, 退出循环, 返回成功结果。
8) digit_tube_display(); //实验结果由数码管显示。
```

2. `flash_data_comparison_test()`: SPI-Flash 数据比对实验

- 打开 SPI-Flash 设备节点, 读取节点信息, 获取 `erasesize`, 在内存中构建 `erasesize` 空间的比对数据。

```
1) fd = open(mtd_path, O_RDWR); //设备节点路径 Linux 与 Android 是有区别的
   (Linux 是 /dev/mtd0,   Android 是/dev/mtd/mtd0)
2) ioctl(fd, MEMGETINFO, info);
3) memset(compared_data, 0xab, info->erasesize); //对比数据
```

- 擦除 Spi-Nor-Flash 第一页的数据, 再写入对比数据, 然后读出第一页数据, 与对比数据比较, 若相同, 则返回成功结果, 否则返回失败结果。

```

1) non_region_erase(fd, 0 , 1, 0); //擦除第一页数据
2) write(fd, compared_data, info->erasesize);
3) lseek(fd, 0, SEEK_SET); //重置文件指针指向文件头
4) read(fd, read_buf, info->erasesize);
5) memcmp(read_buf, compared_data, info->erasesize);

```

3. voltage_detect(): 电压采集实验

● 分别打开三个电压采集节点，读取对应电压值，与有效参考电压区间比较，若无效电压值，则返回失败，若三个电压值均为有效值，则返回成功。

```

1) fd[i] = open(test_path[i], O_RDONLY);
2) read(fd[i], buf, 16);
3) 3 个 i2c 设备采集三种电压(i2c-2: 5V, i2c-6: 3.3V, i2c-7: 12V)

```

2.4 EXTIO_TEST 相关驱动代码说明

1. ExtIO 驱动

● 代码路径:

drivers/misc/extbrd/extbrd_drv.c

• DTS 配置:

1) DTS 路径:

Linux: arch/arm64/boot/dts/rockchip/rk3399-eaidk-linux.dts

Android: arch/arm64/boot/dts/rockchip/rk3399-eaidk-android.dts

2) 节点信息:

```

extbrd: extbrd {
    compatible = "eaidk-extboard";
    io-channels = <&saradc 0>, <&saradc 3>; //两个 ADC-KEY 通道
};

```

3) Config 配置信息:

CONFIG_EAIDK_EXTBRD=y

2. Spi-Nor-Flash 驱动

- 代码路径:

```
drivers/mtd/devices/m25p80.c
drivers/mtd/spi-nor/spi-nor.c
drivers/mtd/mtdcore.c
drivers/mtd/mtdpart.c
drivers/mtd/mtdsuper.c
```

- DTS 配置:

- 1) DTS 路径:

Linux: arch/arm64/boot/dts/rockchip/rk3399-eaidk-linux.dts

Android: arch/arm64/boot/dts/rockchip/rk3399-eaidk-android.dts

- 2) 节点信息:

```
&spi1 {
    status = "okay";
    flash@0 {
        compatible = "gigadevice,w25q64", "jedec,spi-nor";
        #address-cells = <1>;
        #size-cells = <1>;
        reg = <0>;
        m25p,fast-read;
        // The max SCLK of the flash 104/80 MHZ
        spi-max-frequency = <50000000>;
    };
};
```

- 3) Config 信息配置:

```
CONFIG_MTD=y
CONFIG_OF_MTD=y
CONFIG_MTD_OF_PARTS=y
CONFIG_MTD_MAP_BANK_WIDTH_1=y CONFIG_MTD_MAP_BANK_WIDTH_2=y
CONFIG_MTD_MAP_BANK_WIDTH_4=y
CONFIG_MTD_CFI_I1=y
CONFIG_MTD_CFI_I2=y
CONFIG_MTD_SPI_NOR=y
CONFIG_MTD_SPI_NOR_USE_4K_SECTORS=y
CONFIG_MTD_M25P80=y
```


3. 电压采集驱动

- 代码路径:

drivers/hwmon/ina2xx.c

- DTS 配置:

- 1) DTS 路径:

Linux: arch/arm64/boot/dts/rockchip/rk3399-eaidk-linux.dts

Android: arch/arm64/boot/dts/rockchip/rk3399-eaidk-android.dts

- 2) 节点信息:

```
&i2c2 {
    status = "okay";
    i2c-scl-rising-time-ns = <300>;
    i2c-scl-falling-time-ns = <15>;
    clock-frequency=<400000>;
    ina219x41c2: ina219@41 {
        compatible = "ina219";
        reg = <0x41>;
    };
};

&i2c6 {
    status = "okay";
    i2c-scl-rising-time-ns = <300>;
    i2c-scl-falling-time-ns = <15>;
    clock-frequency=<400000>;
    ina219x47c6: ina219@47 {
        compatible = "ina219";
        reg = <0x47>;
    };
};

&i2c7 {
    status = "okay";
    i2c-scl-rising-time-ns = <300>;
    i2c-scl-falling-time-ns = <15>;
    clock-frequency=<400000>;
```

```
    ina219x40c7: ina219@40 {  
        compatible = "ina219";  
        reg = <0x40>;  
    };  
  
};
```

3) Config 信息配置:

```
CONFIG_SENSORS_INA2XX=y
```