

OpenHarmony 基础设备开发-----ADC



目 录

CONTENTS

- [01] 什么是ADC
- [02] ADC的相应接口
- [03] 如何使用ADC

01

什么是ADC

A/D转换器（ADC）是模拟系统与数字系统接口的关键部件，长期以来一直被广泛应用于雷达、通信、电子对抗、声纳、卫星、导弹、测控系统、地震、医疗、仪器仪表、图像和音频等领域。随着计算机和通信产业的迅猛发展，进一步推动了ADC在便携式设备上的应用并使其有了长足进步，ADC正逐步高速、高精度和低功耗的方向发展。

本文采用RK2206芯片自带的逐次逼近寄存器型模数转换器（Successive-Approximation Analog to Digital Converter），是一种常用的A/D转换结构，其较低的功耗表现，不错的转换速率，在有低功耗要求（可穿戴设备、物联网）的数据采集场景下广泛应用。

02

ADC的相应接口

ADC接口的头文件

/device/rockchip/rk2206/adapter/include/**lz_hardware.h**

OpenHarmony基础外设开发中，ADC接口主要分为几大类：

- (1) 初始化、销毁ADC；
- (2) ADC读操作。

02

任务的接口

功能分类	接口名	功能描述
初始化、释放ADC接口	LzSaradcInit	SARADC设备初始化
	LzSaradcDeinit	SARADC设备释放
读ADC接口	LzSaradcReadValue	SARADC设备读取

02

ADC接口

```
int LzSaradcInit(void);
```

该函数主要功能是SARADC设备初始化。

成功返回LZ_HARDWARE_SUCCESS，其余为失败。

```
int LzSaradcDeinit(void);
```

该函数主要功能是SARADC设备释放。

成功返回LZ_HARDWARE_SUCCESS，其余为失败。

02

ADC接口

```
int LzSaradcReadValue(unsigned int chn, unsigned int *val);
```

该函数主要功能是SARADC设备读取。

- 参数chn：ADC通道id；
- 参数val：读取值。

成功返回LZ_HARDWARE_SUCCESS，其余为失败。

03

如何使用ADC

本课程通过4个按键来让一路ADC采集线路电压发生变化。具体电压变化如下：

- (1) 无按键按下时，电压为2.3V
- (2) Key1按下时，电压为0.01V
- (3) Key2按下时，电压为0.55V
- (4) Key3按下时，电压为1.00V
- (5) Key4按下时，电压为1.65V

03

如何使用ADC

1、打开sdk下面路径的文件

```
vendor/lockzhiner/rk2206/samples/b1_adc/adc_example.c
```

2、创建任务

在adc_example()中，通过LOS_TaskCreate()创建adc_process()任务。

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

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

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

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

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

03

如何使用ADC

3、ADC采集

在adc_process()中，通过LzSaradcInit()将GPIO0_PC5复用为SRADC5，然后每隔1秒通过LzSaradcReadValue()读取ADC当前电压。

```
static unsigned int adc_dev_init()
{
    unsigned int ret = 0;

    uint32_t *pGrfSocCon29 = (uint32_t *) (0x41050000U + 0x274U);
    uint32_t ulValue;

    ret = DevIoInit(m_adcKey);
    if (ret != LZ_HARDWARE_SUCCESS)
    {
        printf("%s, %s, %d: ADC Key IO Init fail\n", __FILE__, __func__,
            __LINE__);
        return __LINE__;
    }
}
```

```
    ret = LzSaradcInit();
    if (ret != LZ_HARDWARE_SUCCESS) {
        printf("%s, %s, %d: ADC Init fail\n", __FILE__, __func__, __LINE__);
        return __LINE__;
    }

    /* 设置saradc的电压信号，选择AVDD */
    ulValue = *pGrfSocCon29;
    ulValue &= ~(0x1 << 4);
    ulValue |= ((0x1 << 4) << 16);
    *pGrfSocCon29 = ulValue;

    return 0;
}
```

```
static float adc_get_voltage()
{
    unsigned int ret = LZ_HARDWARE_SUCCESS;
    unsigned int data = 0;

    ret = LzSaradcReadValue(ADC_CHANNEL, &data);
    if (ret != LZ_HARDWARE_SUCCESS)
    {
        printf("%s, %s, %d: ADC Read Fail\n", __FILE__, __func__, __LINE__);
        return 0.0;
    }

    return (float)(data * 3.3 / 1024.0);
}
```



如何使用ADC

4、修改编译脚本

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

```
"/b0_adc:adc_example",
```

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

```
hardware_LIBS = -lhal_iohardware -lhardware -ladc_example
```

5、编译固件

```
hb set -root .
```

```
hb set
```

```
hb build -f
```



如何使用ADC

6、烧写固件

7、通过串口查看结果

运行结果

```
*****Adc Example*****
```

```
vlt:3.297V
```

```
*****Adc Example*****
```

```
vlt:3.297V
```

```
*****Adc Example*****
```

```
vlt:3.297V
```

```
.....
```



谢谢聆听

单击此处添加副标题内容