**通过jni接口，Java的native方法来调用相关的接口。把libs文件下的内容拷贝添加到工程目录的libs文件夹里面。把src文件拷贝到工程src目录下，注意此文件的包名不能改package** com.yf.humansensor,文件路径也要保持一致;**改了就会调用不到。可以参考测试源码。**

1. **人体感应的获取**

**方法原型是int** get\_humansensor\_value(**int** fd);

**其中参数fd是打开设备节点返回的值，例如**

**int** value;

**int** fd;

fd=humansensor\_manager.*open*();

value=humansensor\_manager.*get\_humansensor\_value*(fd);

**得到的值value=1证明有人靠近，value=0证明没人。**

1. **GPIO1的控制**

**方法原型是void** set\_gpio1\_value(**int** fd, **int** value);

**其中参数fd是打开设备节点返回的值，例如**

**int** fd;

fd=humansensor\_manager.*open*();

humansensor\_manager.set\_gpio1\_value(fd,value);

**参数value是控制IO口输出的电平，value=1代表输出高电平，value=0代表输出低电平。**

**GPIO1的获取**

**方法原型是int** get\_gpio1\_value(**int** fd);

**其中参数fd是打开设备节点返回的值，例如**

**int** value;

**int** fd;

fd=humansensor\_manager.*open*();

Value=humansensor\_manager.*get\_gpio1\_value*(fd);

**得到的值value=1证明这个IO口是高电平，value=0证明这个IO口是高电平。**

1. **GPIO2的控制**

**方法原型是void** set\_gpio2\_value(**int** fd, **int** value);

**其中参数fd是打开设备节点返回的值，例如**

**int** fd;

fd=humansensor\_manager.*open*();

humansensor\_manager.set\_gpio2\_value(fd,value);

**参数value是控制IO口输出的电平，value=1代表输出高电平，value=0代表输出低电平。**

**GPIO2的获取**

**方法原型是int** get\_gpio2\_value(**int** fd);

**其中参数fd是打开设备节点返回的值，例如**

**int** value;

**int** fd;

fd=humansensor\_manager.*open*();

Value=humansensor\_manager.*get\_gpio2\_value*(fd);

**得到的值value=1证明这个IO口是高电平，value=0证明这个IO口是高电平。**

1. **GPIO3的控制**

**方法原型是void** set\_gpio3\_value(**int** fd, **int** value);

**其中参数fd是打开设备节点返回的值，例如**

**int** fd;

fd=humansensor\_manager.*open*();

humansensor\_manager.set\_gpio3\_value(fd,value);

**参数value是控制IO口输出的电平，value=1代表输出高电平，value=0代表输出低电平。**

**GPIO3的获取**

**方法原型是int** get\_gpio3\_value(**int** fd);

**其中参数fd是打开设备节点返回的值，例如**

**int** value;

**int** fd;

fd=humansensor\_manager.*open*();

Value=humansensor\_manager.*get\_gpio3\_value*(fd);

**得到的值value=1证明这个IO口是高电平，value=0证明这个IO口是高电平。**

1. **GPIO4的控制**

**方法原型是void** set\_gpio4\_value(**int** fd, **int** value);

**其中参数fd是打开设备节点返回的值，例如**

**int** fd;

fd=humansensor\_manager.*open*();

humansensor\_manager.set\_gpio4\_value(fd,value);

**参数value是控制IO口输出的电平，value=1代表输出高电平，value=0代表输出低电平。**

**GPIO4的获取**

**方法原型是int** get\_gpio4\_value(**int** fd);

**其中参数fd是打开设备节点返回的值，例如**

**int** value;

**int** fd;

fd=humansensor\_manager.*open*();

Value=humansensor\_manager.*get\_gpio4\_value*(fd);

**得到的值value=1证明这个IO口是高电平，value=0证明这个IO口是高电平。**