

本压缩包由一下几个目录组成

1. **include**: 使用本 sdk 接口需要引入的头文件。

2. **libs**: 使用本 sdk 需要引入的 so 库文件。

除了依赖这个目录下的 so 库外, 编译时还需要引入 **-pthread** 库。

如果使用 **cmake** 构建项目, 可以参考 **samples** 目录下的 **sample_**开头的目录下的 **CMakeLists.txt** 文件。

类似这种:

```
cmake_minimum_required(VERSION 2.8)
project(sample_semantic)
include_directories(include)
set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++11 -g -Wall -fexceptions -fpermissive -fPIC")
# 编译测试文件
add_executable(${PROJECT_NAME} testsemantic.cpp)
target_link_libraries(${PROJECT_NAME} -L./lib taisdk -pthread)
```

引入头文件

引入taisdk和pthread

编译好的程序, 在运行时可能出现如下错误:

```
/sample_wakeup: error while loading shared libraries: libtaisdk.so: cannot open shared object file: No such file or directory
```

这是因为 **libs** 下的 so 库并不能被系统找到。 有三种解决办法:

- 将 **libs** 下的 so 库放入系统库目录下, 如 **/lib**, 这样能简单粗暴解决。
- 开发者采用 **dlopen** 灵活加载 so 库。
- 修改 **LD_LIBRARY_PATH**

执行以下命令:

```
export LD_LIBRARY_PATH=./libs:$LD_LIBRARY_PATH
```

```
sudo ldconfig
```

3. **samples**: 示例代码, 结构如下

- out**: 程序输出目录, 内部有脚本文件, 方便执行程序, 可以阅读.sh 代码了解如何执行程序的。

b. `sample_`开头的目录:示例程序源代码工程, `cmake` 构建

`sample_wakeup`:唤醒示例, 非必有文件夹

`sample_voice`: 语音识别示例, 非必有文件夹

`sample_tts`:语音合成示例, 非必有文件夹

`sample_semantic`:语义示例, 非必有文件夹

编译运行步骤如下:

1. 环境准备: 需要先在系统上装 `cmake`。

```
sudo apt-get install cmake
```

2. 编译: 在各个 `sample_`开头的目录里面执行

```
cmake .
```

```
make
```

3. 可执行程序输出到`./out`目录下。

4. `sudo` 执行 `out` 目录下的 `run 某某.sh` 脚本程序 (会自动加载 `so` 库并执行相应的示例程序)

```
sudo ./runsemantic.sh 调用语义示例程序 (非必有文件)
```

```
sudo ./runwakeup.sh 调用唤醒示例程序 (非必有文件)
```

```
sudo ./runvoice.sh 调用语音识别示例程序 (非必有文件)
```

```
sudo ./run tts.sh 调用 TTS 示例程序 (非必有文件)
```

4. `res`: 资源文件+配置文件。

配置文件有 `config.conf`

4. 各类 pdf 文档: 顾名思义