



# 旷明 XOS 开源项目添加方法

部 门	
文 档 编 号	
版 本 号	V1.0.1
作 者	QUA

版权所有

旷明智能科技（无锡）有限公司

本资料及其包含的所有内容为旷明智能科技（无锡）有限公司所有,受中国法律及适用之国际公约中有关著作权法律的保护。未经旷明智能科技（无锡）有限公司书面授权，任何人不得以任何形式复制、传播、散布、改动或以其它方式使用本资料的部分或全部内容，违者将被依法追究责任。

**更新记录**

日期	更新人	版本	备注
2025/09/01	QUA	V1.0.1	初稿

## 目录

<b>1、 引言 .....</b>	<b>5</b>
1.1 编写目的 .....	5
1.2 预期读者和阅读建议 .....	5
<b>2、 XOS 第三方应用与库的导入 .....</b>	<b>错误！未定义书签。</b>
2.1 源码编译 （以 SPNG 为例） .....	6
2.1.1 源码的获取 .....	错误！未定义书签。
2.1.2 标准 Cmake 工程的结构 .....	错误！未定义书签。
2.1.3 平台适应性编译 .....	错误！未定义书签。
2.2 加入平台 .....	错误！未定义书签。
2.2.1 将编译出来的.so 文件或者现成的库文件放入平台预期的目录下 .....	错误！未定义书签。
2.2.2 添加开关（比如加入 external 目录下） .....	错误！未定义书签。
2.2.3 在对应芯片型号的 Cmakelist.txt 中添加 .....	错误！未定义书签。
2.2.4 具体项目的应用 .....	错误！未定义书签。



## 1、引言

### 1.1 编写目的

本文用于指导用户快速导入自己应用与资源库。

### 1.2 预期读者和阅读建议

本文档可提供给客户、研发人员、技术支持工程师和测试工程师使用。

## 2、创建极简应用

### 2.1 新建应用目录和源码

分如下几个步骤：

1、在 core/apps/ 创建目录，比如：helloapp。

里面创建 mvc 的代码目录结构：Model、Control、view 三个目录。

2、在 view 目录里面添加 pages 目录和 components 目录。

3、现在创建主界面：helloapp\_main.c

4、现在在 helloapp\_main.c 用 lvgl 控件创建一个控件，可以显示 hello world

5、新建编译脚本：CMakeLists.txt

这些可以用 create\_app.py 脚本完成。脚本路径：build/script/create\_app.py

### 2.2、创建项目配置文件

完成源码后，下面添加一个产品，分如下几个目录：

1、在目录 product 创建 helloapp；

2、里面新建 project\_def\_defconfig

参考内容：

```
CONFIG_PRODUCT_HELLOAPP=y  
CONFIG_XOS_PROJECT_TYPE="helloapp"
```

这两条是比较关键的，跟应用启动有关。

其他的 config 定义可以参考《旷明 XOS 平台编译方法》

#### 2.2.1 添加 config 定义

修改 build/kconf 下的 Kconfig.apps 和 Kconfig.product 增加对应 app 和 product 的项。

Kconfig.apps 增加的内容如下：

```
config XOS_USE_APP_HELLOAPP  
    bool "hello_app"  
    default n  
    help  
        select app helloapp
```

Kconfig.product 增加的内容如下：

```
config PRODUCT_HELLO_APP
    bool "hello_app"
    help
        Configuration for helloapp.
```

## 2.2.2 配置项与启动的关系

CONFIG\_XOS\_PROJECT\_TYPE 定义的 helloapp 会被拼凑成应用的启动入口，启动的函数名入口固定为 helloapp\_init

参考例子：

```
void helloapp_init(void)
{
    LV_LOG_USER("helloapp_init: 开始初始化");
    qua_app_ops *ops = malloc(sizeof(qua_app_ops));
    if (ops) {
        memset(ops, 0x0, sizeof(qua_app_ops));
        ops->on_create = xos_helloapp_entry;
        ops->on_destory = app_destory;
        app_manager_register("helloapp", ops);
    }
}
```

## 2.2.3 应用启动

如果按规范实现了 helloapp\_init 函数，里面注册了应用名字，那么系统启动后，就会自动调用

```
app_manager_start("helloapp");  
来启动应用
```

## 2.3、构建应用

### 2.3.1 构建模拟器版本：

```
make distclean;  
make project_helloapp_defconfig; make xos-sim
```

### 2.3.2 构建真机版本

```
make distclean; make project_helloapp_defconfig; make xos-qxosui
```

具体命令的用法可以参考《旷明 XOS 平台编译方法》

### 3、创建拓展功能

#### 3.1 如何添加模拟器专用的代码

如果是模拟器上面编译，编译会添加-BUILD\_SIMULATOR=1  
所以可以添加如下代码：用于仅在模拟器上面生效的代码：

```
#if defined(BUILD_SIMULATOR) && BUILD_SIMULATOR
//code for simulator
#endif
```

同理：如下为非模拟器代码

```
#if !defined(BUILD_SIMULATOR) || BUILD_SIMULATOR == 0
//code for qm10x
#endif
```

更多功能请参考《旷明 XOS 应用开发指南》

#### 3.2 其他应用能力

需要参考：《旷明 XOS 应用开发指南》，里面包含了事件处理、多应用调用、资源存放等；