Osire demo 使用说明

发布清单
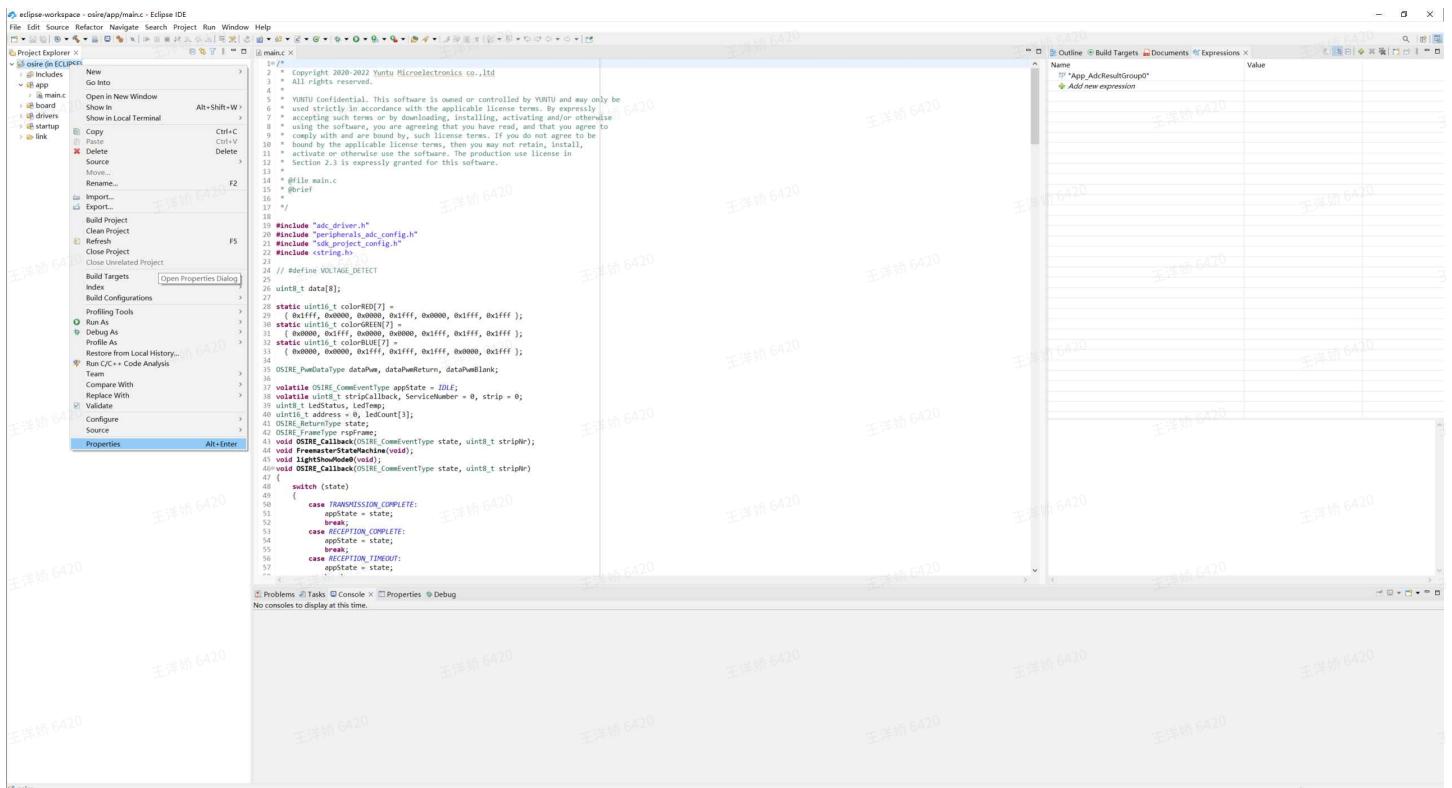
- Osire demo 应用程序 (双击 osire.yct 生成工程)

- lib_osire.a (OSIRE 的驱动包，适用于 GCC 与 YuntuIDE 开发环境)

- mdk_osire.lib (OSIRE 的驱动包，适用于 KEIL 开发环境)
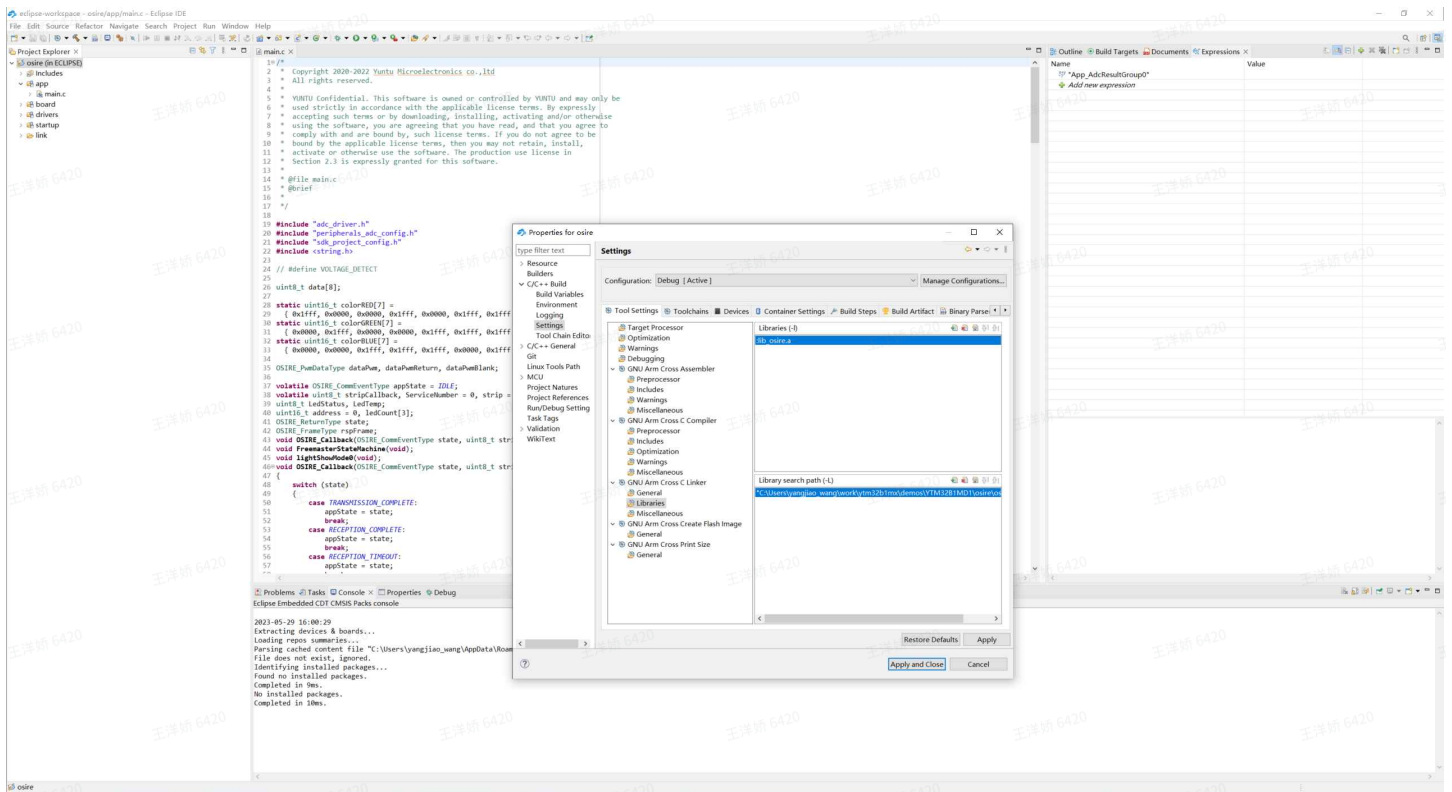
- iar_osire.a (OSIRE 的驱动包，适用于 IAR 开发环境)

常见问题

YuntuIDE添加库

在 YuntuIDE 加载完 OSIRE 的示例后，请注意下面几条提示。

右键工程并点击 **Properties** 来打开工程属性窗口。

然后点击 **C/C++ Build** 来展开并点击 **Setting**, 并选择 **Libraries**。*Libraries(-l)* 里的内容是实际的库文件, 例如, :lib_osire.a, Library search path(-L) 里的内容是库所在的路径。

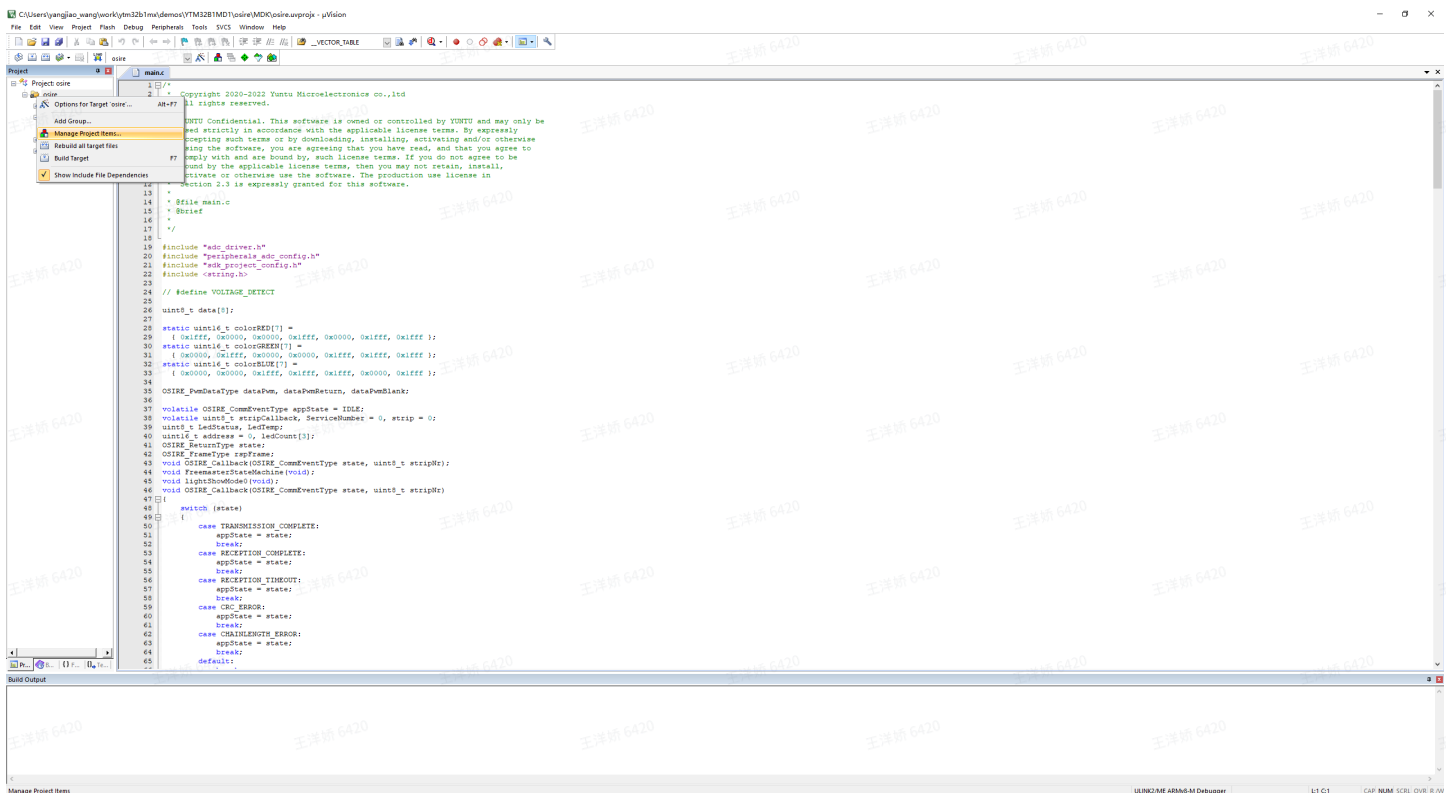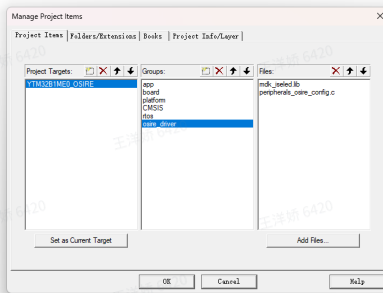随后正常编译即可。

KEIL添加库

在 KEIL 加载完 OSIRE 的示例后，请注意下面几条提示。
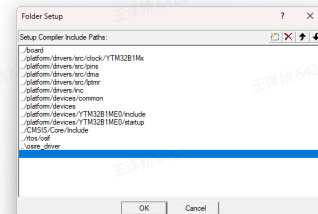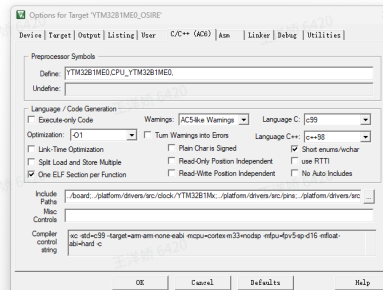
右键工程并点击 **Manage Project Items** 来打开工程文件配置窗口

添加 **osire_driver** 文件夹，并添加 **mdk_osire.lib** 与 **peripherals_osire_config.c**
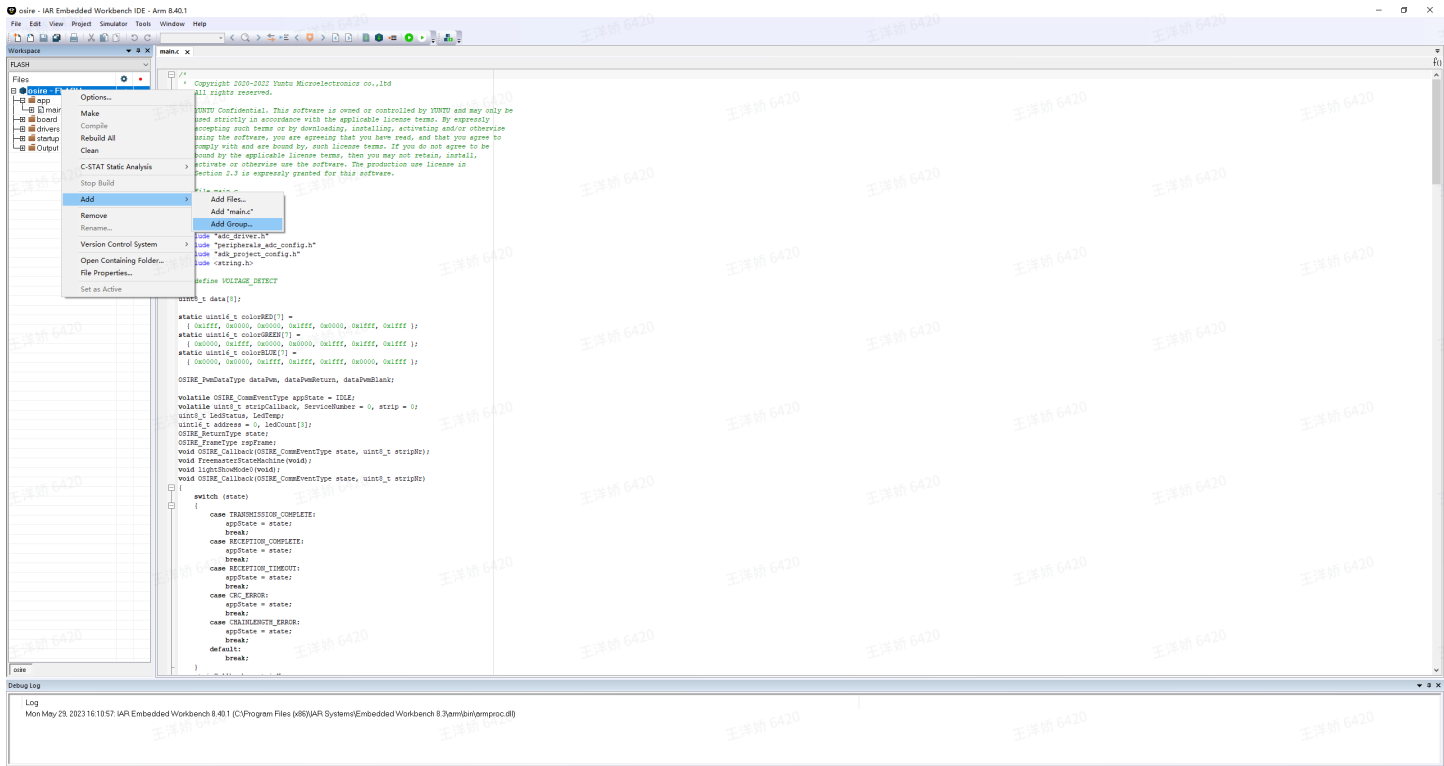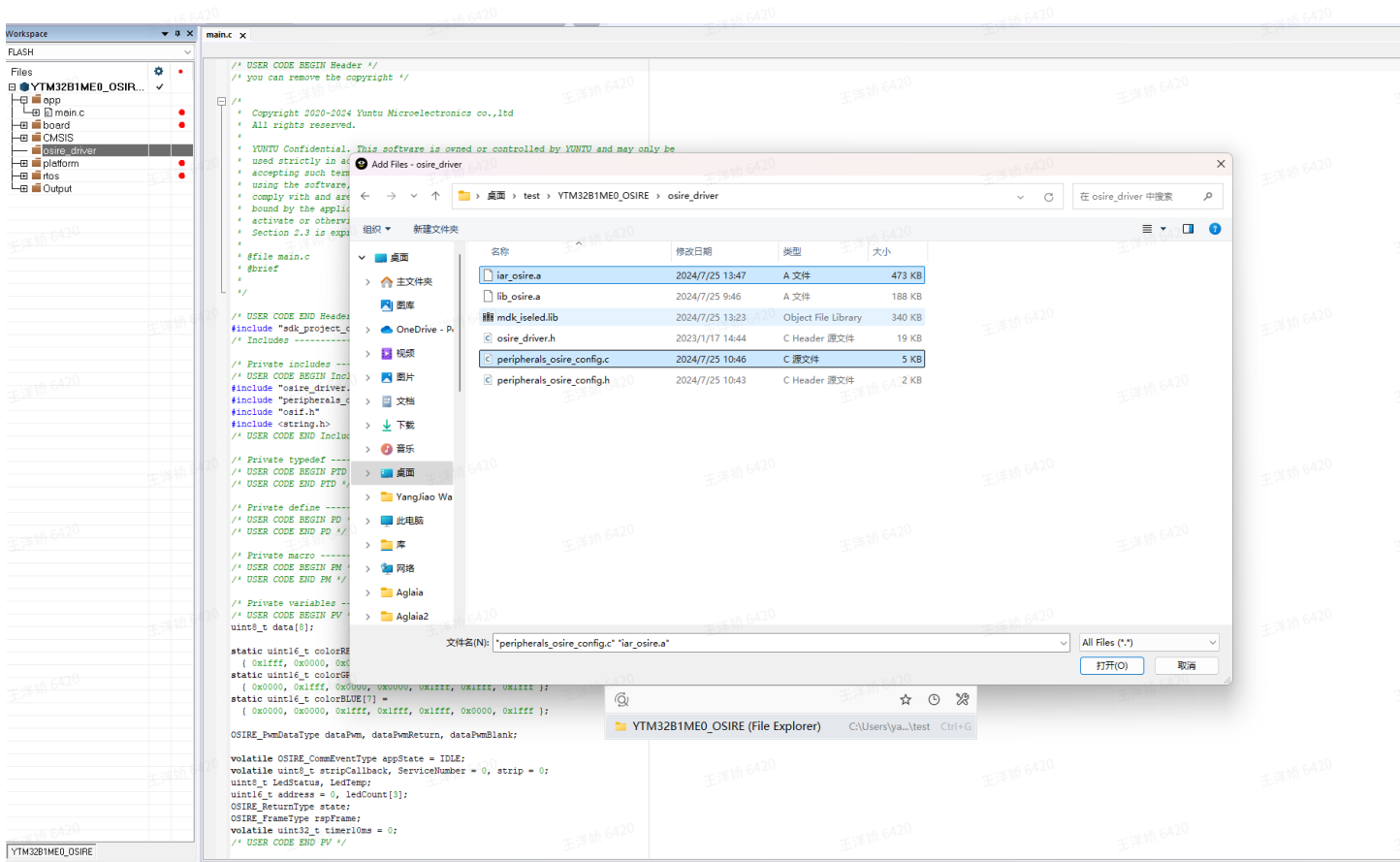
添加编译路径，增加 **osire_driver**

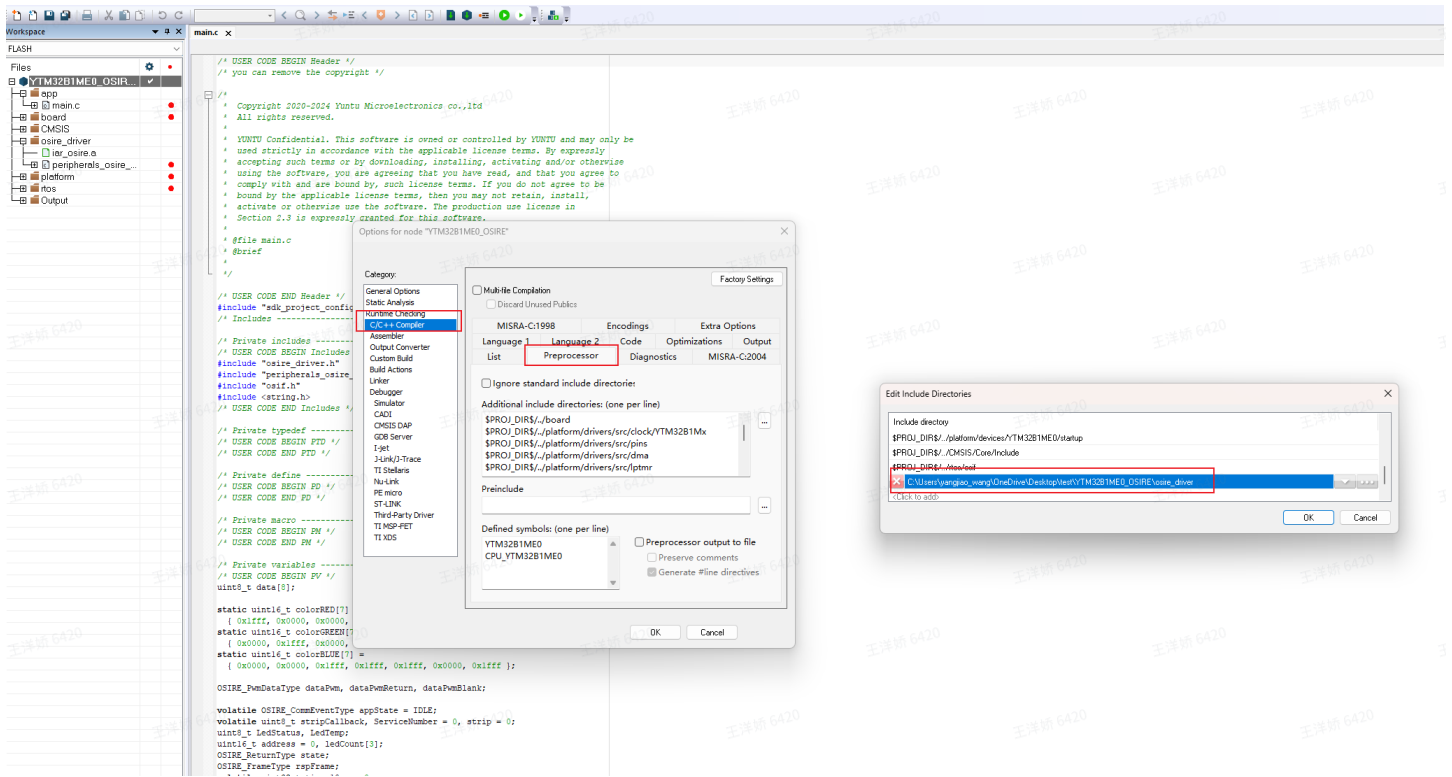

随后正常编译即可。

IAR添加库

在 IAR 中加载完 OSIRE 的示例后，请注意下面几条提示。

右键工程并点击 Add，并选择 Add Group，添加 osire_driver 的目录。

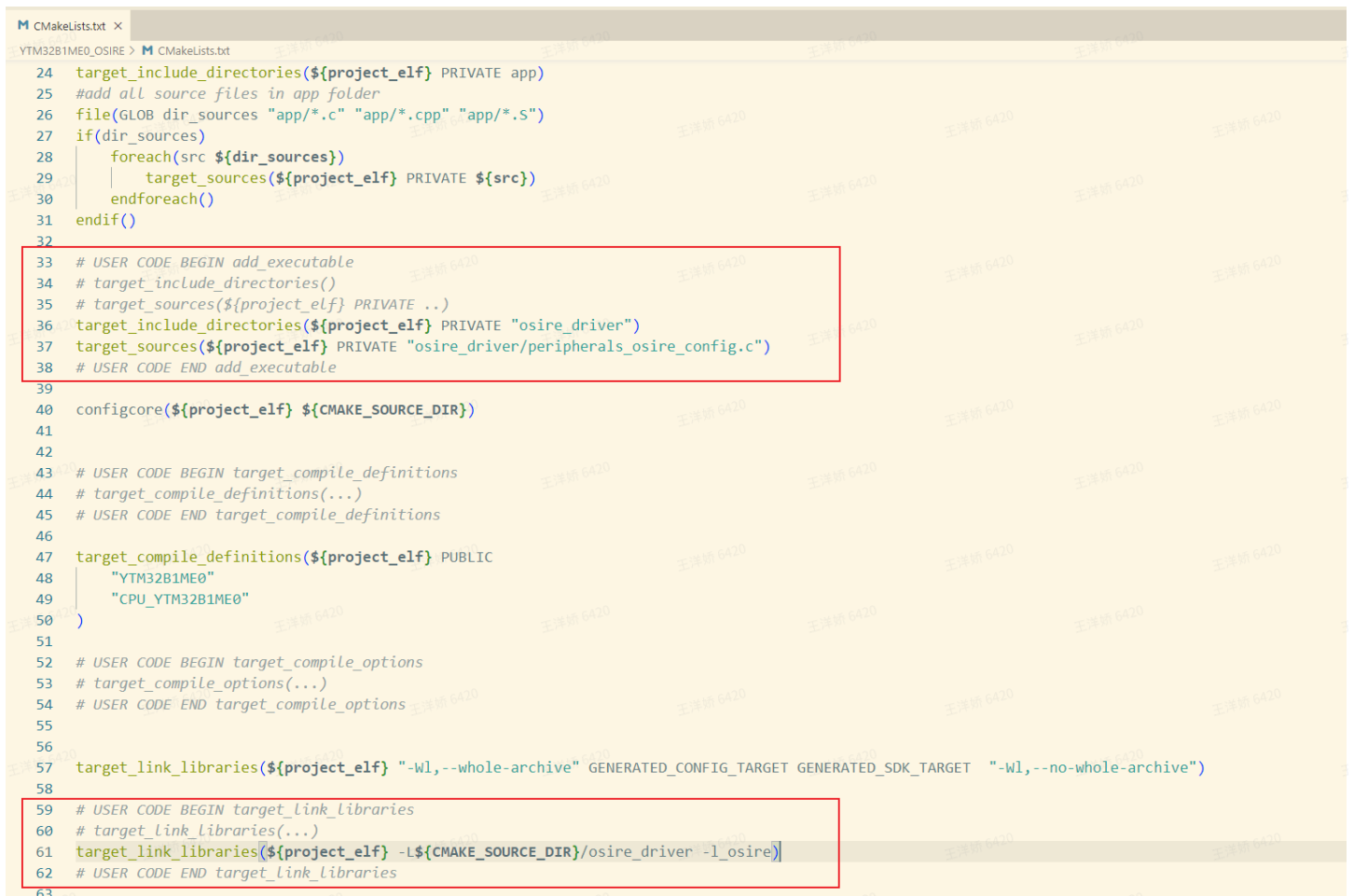右键 **osire_driver** Group，并选择 Add Files 添加 **iar_osire.a** 与 **peripherals_osire_config.c** 。

添加编译路径，增加 **osire_driver**

随后正常编译即可

CMakeGCC

通过使用文件夹内的 CMakeLists.txt ，直接编译。

关于软件调试的注意点

避免在 **FreemasterStateMachine**() 函数中的那部调试。为提高利用率，OSIRE 的通信控制采用异步方式，且设有超时机制，当程序在 FreemasterStateMachine() 函数中单步调试时，可能会导致通信来不及响应，导致 OSIRE 灯珠控制失败。

硬件连接

驱动板上的 CANH 与 灯板上的 CANH 连接，驱动板上的 CANL 与 灯板上的 CANL 连接，驱动板上的 GND 与 灯板上的 GND 连接。

若希望驱动板与灯板统一上电，可将驱动板上的 VDD 与 灯板上的 VDD 连接；否则不要将驱动板上的 VDD 与 灯板上的 VDD 连接。