

AMT630HV160 BSP

用户使用说明

版本: _____ V1.1

日期: _____ 2026-07

深圳开阳电子股份有限公司拥有随时修改本手册的权利，内容如有更改，恕不另行通知。深圳开阳电子股份有限公司对本手册不承担任何形式的保证，包括但不限于对产品特定用途适销性和适用性的隐含保证。深圳开阳电子股份有限公司对本手册中包含的错误或对本手册的使用所带来的偶然或继起损害不承担任何责任。

版本记录：

日期	版本号	更新说明	
2024-11	V1.0	初始版本	
2026-07	V1.1	添加 AMT630HV160S 支持说明	

1 BSP 目录结构

目录名	描述
amt630hv160-freertos-beta	应用工程文件，使用 IAR ARM 8.30编译调试，基于 FreeRTOS + AWTK，支持 JTAG调试
amt630hv160-amtldr	启动工程文件，使用 IAR ARM 8.30编译调试, 裸机程序
amt630hv160-mcu	MCU工程文件，同样使用 IAR ARM 8.30，裸机程序
Doc	相关文档说明
Tools	开发中需要用到的各种工具
升级文件	包含AWTK官方Demo升级文件和公板Demo升级文件

1.1 应用工程目录结构(主要内容)

目录名	描述
app	应用逻辑代码
ArkmicroFiles	板级以及芯片驱动代码
FreeRTOS	FreeRTOS内核
FreeRTOS-Plus	FreeRTOS扩展，包括FAT，POSIX，CLI(命令行)等
lib	第三方库，包括awtk、bt和hx170dec等
demo	USB、MMC、Audio等测试例程
proj	工程文件

2 SD 卡升级说明

2.1 升级方法

首先、将启动模式调整为 SD 启动模式（公板上将跳线帽顺序切换为

BIT2(0):BIT1(1):BIT0(1))。

其次、拷贝升级文件到 TF 卡（包括 HV160LDR.bin、image.bin、amtldr.bin、amt630hvl60.bin、bootanim.bin 和 rom.bin 六个升级文件）。

最后、插入 TF 卡，断电重启升级。升级过程 LCD 会显示升级进度，MCU 串口同步输出打印信息。升级成功打印信息如下图所示：

```
cpu/ahb clk = 240000000 Mhz
SDMMC card successfully
EMMC_SetBusWidth 4
SDMMC card successfully
UPDATE_CheckApp ok
Open fastboot file fail
Burn cpuldr file start.
Burn cpuldr file end.
Burn cpuapp file start.
Burn cpuapp file end.
Start cpu...
Waiting cpu update finished...
Update finished! Please change the bootmode and repower.
```

需要注意的是升级卡需要格式化成 FAT32 文件系统，另外分配单元大小要小于等于 4096 字节，最好使用 4g~16g 容量的卡。



3 U 盘升级说明

3.1 升级方法

首先、将启动模式调整为 USB-HOST 启动模式（公板上将跳线帽顺序切换为 BIT2(0):BIT1(1):BIT0(0)）。

其次、拷贝升级文件到 U 盘（包括 HV160LDR.bin、image.bin、amtldr.bin、amt630hvl60.bin、bootanim.bin 和 rom.bin 六个升级文件）。

最后、插入 U 盘，断电重启升级。升级过程 LCD 会显示升级进度，MCU 串口同步输出打印信息。升级成功打印信息如下图所示：

```
cpu/ahb clk = 240000000 Mhz
SDMMC card successfully
EMMC_SetBusWidth 4
SDMMC card successfully
UPDATE_CheckApp ok
Open fastboot file fail
Burn cpuldr file start.
Burn cpuldr file end.
Burn cpuapp file start.
Burn cpuapp file end.
Start cpu...
Waiting cpu update finished...
Update finished! Please change the bootmode and repower.
```

升级完成后需要手动切换回 EMMC 启动模式，断电重启，LCD 依次显示开机 logo，开机动画和仪表界面。

4 升级文件说明

HV160LDR.bin

MCU 阶段的 U 盘/SD 卡升级引导文件。由 amt630hvl60-mcu 工程编译生成的目标文件 amt630hvl60.bin 重命名得到。编译前需要将宏定义 APP_FOR_BURN 的值改为 BURN_IAMGE（文件路径 amt630hvl60-mcu/src/amt630hvl60_conf.h）。

image.bin

MCU 应用程序。由 amt630hvl60.bin 通过工具 Tools/ImgcrGUI/ImgcrGUI.exe 转换得到（其中 amt630hvl60.bin 文件由 amt630hvl60-mcu 工程编译生成，工具使用请参考“Doc/AMT630HV160 启动流程说明.pdf”）。编译前需要将宏定义 APP_FOR_BURN 屏蔽掉（文件路径 amt630hvl60-mcu/src/amt630hvl60_conf.h）。

amtldr.bin

CPU 引导程序。amt630hvl60-amtldr 工程编译生成。

amt630hvl60.bin

CPU 应用程序。amt630hvl60-freertos-beta 工程编译生成。

bootanim.bin

开机动画文件。具体制作方法参考“Doc/AMT630HV160 开机动画制作说明.pdf”，需将生成的 bootanimation 文件重命名为 bootanim.bin。

rom.bin

UI 资源文件。将工具“Tools/RomMaker/RomMaker.exe”拷贝到资源文件所在的文件夹中，运行 RomMaker.exe，则会在当前目录下生成 rom.bin 文件。**以资源所在目录作为当前目录，所有文件相对于当前目录的相对路径名长度不能大于 64 个字符。**为了加快图片解码速度，芯片支持 QOI 格式图片硬解，因此需要将 UI 资源里的 PNG 格式图片使用 ARKQoiConverter 工具转换成 QOI 格式，具体转换方法参考“Doc/ARKQoiConverter 使用说明.pdf”。

dcicrom.bin

MCU 资源文件。固化资源文件（DCIC 模块资源文件，非必要升级文件）。

5 升级注意事项

在生产或调试阶段可以通过 TF 卡或者 U 盘进行升级。

生产后的机器，后续升级时通常只需要升级 amt630hvl60.bin、rom.bin、bootanim.bin 三个文件。没有特殊需求，可以不用升级 amtldr.bin(如果是 SD 卡或者 USB 升级，则不用拷贝此文件到 SD 卡或者 U 盘里)。

6 JTAG 调试说明

以 EMMC 版本为例，将启动模式切换到 EMMC 模式(BIT2(1):BIT1(1):BIT0(0))，调试前要确保机器可以正常启动（非空片），否则可能导致程序运行异常。系统启动后，即可使用 JTAG 进行调试（MCU 工程要求 Jlink 工具版本在 V6.32f 以上，CPU 工程无此要求）。

7 其他说明

7.1 SDK 内核 rtos 使用的是 FreeRTOS，相关文档可以查询 FreeRTOS 官方网站

<https://www.freertos.org>。

7.2 应用 UI 框架默认使用的是 AWTK。

7.3 amt630hvl60 封装的 DDR3 容量为 128MB。

7.4 芯片的管脚复用在 pinctrl.c 文件里配置，可以根据自己的产品板的管脚使用不同，在 vPinctrlSetup 函数配置正确的管脚使用。

7.5 大部分管脚都可以设置驱动能力，比如要修改 LCD CLK 信号的驱动强度，可以先查看芯片原理图找到 LCD_CLK 和 I055 是复用的，之后在 pinctrl.c 文件 pin_groups 数组里找到 LCD 的管脚配置中对应 I055 的部分，在后面添加想要设置的驱动电流，比如 PAD_DRIVE_4MA，未添加驱动电流设置的管脚使用的是默认值 PAD_DRIVE_2MA。

```
{.groupid = PGRP_LCD_TTL_CH0, .pins_num = 28,
 /* de      clk      vynsc      hsync */
 .pins = {{54, 2}, {55, 2, PAD_DRIVE_4MA}, {56, 2}, {57, 2},
          {40, 2}, {41, 2}, {42, 2}, {43, 2}, {44, 2}, {45, 2}, {46, 2}, {47, 2},
          {101, 2}, {102, 2}, {48, 2}, {49, 2}, {50, 2}, {51, 2}, {52, 2}, {53, 2},
          {103, 2}, {104, 2}, {105, 2}, {106, 2}, {107, 2}, {108, 2}, {109, 2}, {110, 2}},
```

7.6 支持 EMMC，默认存储介质为 EMMC。

7.7 支持 Spi Nor Flash，系统默认配置是 EMMC，可以根据需求切换为 Spi Nor Flash。具体步骤：把 amt630hvl60-mcu 工程 amt630hvl60_conf.h 文件、amt630hvl60-amtldr 和 amt630hvl60-freertos-beta 工程 board.h 文件的宏定义 DEVICE_TYPE_SELECT 配置为 SPI_NOR_FLASH 即可。同时支持 Spi Nor Flash 四线模式，默认是关闭的，如果需要可以将上述文件中注释掉的宏定义 SPI0_QSPI_MODE 打开。

7.8 Demo 应用默认没有打开倒车显示，需要的话修改 main_awtk.c 文件，将注释掉的 carback_init() 调用打开。此时会创建了一个 carback_test_thread 线程来模拟发送进入和退出倒车消息，大约每隔 30s 会进入倒车显示，10s 后退出。Demo 程序配置的 RN6752 支持 AHD 720P 信号，如果板上的 AHD 输入座接入了正确的 AHD 信号，在进入倒车的时候可以看到视频图像，如果没有接入信号的话此时则会显示蓝屏。

7.9 支持 SD 卡，默认是打开的，如果不需要可以修改 board.h 头文件将 SDMMC1_SUPPORT 宏定义注释掉。打开 SDMMC1_SUPPORT 宏定义，宏定义 SDMMC_TEST 为 1 时会创建 sdcard_read_thread 线程来检测插入的卡里面是否有 amtldr.bin、

amt630hvl60.bin、bootanim.bin 和 rom.bin 升级文件，并且会判断这些文件的版本和已经烧录到 flash 里的版本是否一致，不一致的话就会进行 OTA 升级。

7.10 支持 U 盘，默认是打开的，如果不需要可以修改 board.h 头文件将 USB_SUPPORT 宏定义注释掉。打开 USB_SUPPORT 宏定义，且宏定义 USB_TEST 为 1 时，会创建 uab_read_thread 线程来检测插入的 U 盘里面是否有 amtldr.bin、amt630hvl60.bin、bootanim.bin 和 rom.bin 升级文件，并且会判断这些文件的版本和已经烧录到 flash 里的版本是否一致，不一致的话就会进行 OTA 升级。

7.11 默认支持 OTA 升级功能。如果存储设备没有足够的空间保证 bootanim.bin 和 rom.bin 文件做 A/B 备份的话可以修改 board.h 头文件注释 IMAGE_RESOURCES_BACKUP 宏定义，bootanim.bin 和 rom.bin 文件做 OTA 升级时就会覆盖旧文件。

7.12 支持差分升级功能，默认关闭。相关使用说明参考“Doc/AMT630HV160 差分升级使用说明.pdf”文件。

7.13 支持 TTL, LVDS, MIPI 点屏接口，默认设置的是 1920x720 的 LVDS 屏，使用其他屏可以在 board.h 头文件里修改屏参。

7.14 支持 RN6752 视频解码芯片，RN6752 支持 AHD，可以在 board.h 头文件中选择配置。

7.15 支持 ADC 按键，默认是关闭的，如果需要打开可以修改 board.h 头文件将 ADC_KEY 宏定义注释。

7.16 支持屏幕旋转(横屏竖用或者竖屏横用)，修改 board.h 头文件中宏定义 LCD_ROTATE_ANGLE 的值来设置旋转角度。

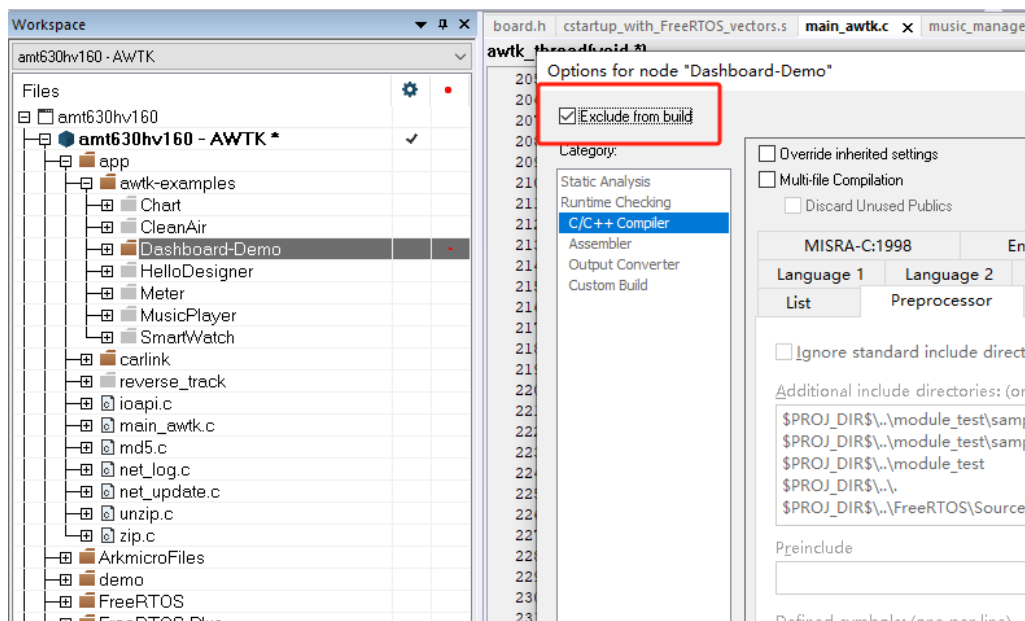
7.17 支持屏幕水平或者垂直镜像显示，修改 board.h 头文件中宏定义 LCD_H_FLIP 和 LCD_V_FLIP 的值打开相应的功能。

7.18 支持 wav 格式音频放音和录音功能，默认是关闭的，如果需要打开可以修改 board.h 头文件，将 AUDIO_REPLAY 和 AUDIO_RECORD 宏定义打开，当宏定义 REPLAY_TEST 和 RECORD_TEST 为 1 时，会创建 wavplay_demo 和 wavrecord_demo 音频相关参考 demo。

7.19 Demo 应用默认打开图像刷新帧率监测，会在屏幕左上角显示刷新帧率。

7.20 如果需要打印 Demo 应用每个线程的状态信息，需要在 main_awtk.c 文件添加 TASK_STATUS_MONITOR 宏定义，该打印会影响界面刷新效率，量产软件需要去掉该打印。

7.21 支持 AWTK UI 引擎，编译 AWTK 应用需要选择 AWTK 工程配置，工程包含 7 个 AWTK 的官方示例程序，默认选择的是 Dashboard-Demo，如果需要运行其他的示例程序需要修改工程配置，把 Dashboard-Demo 代码从编译中排除，之后把其他的 Demo 代码包含到编译中，另外注意不同的 Demo 需要使用对应的资源文件。



7.22 为了优化 AWTK 内存使用，添加了 AWTK 单独的堆大小设置，通过 platform.c 文件中的宏定义 AWTK_HEAP_SIZE 来修改。但是需要注意的是，修改该值后需要同步修改 FreeRTOSConfig.h 文件中的 configTOTAL_HEAP_SIZE 值来调整系统的堆大小，其中 AWTK_HEAP_SIZE 和 configTOTAL_HEAP_SIZE 的和是总的堆大小。

7.23 模块访问权限由 SEMA 模块管理，MCU 和 CPU 若想有效访问硬件模块需通过 SEMA 模块获取互斥锁（上电时 SEMA 所有锁空闲，此时 MCU 未获取 SEMA 锁也能有效访问所有模块）。MCU 允许访问板子所有模块。CPU 允许访问除 CANFD、DCIC、加密相关模块（PKA、SPACC、TRNG）、RDC 外的所有模块。

7.24 MCU 通过配置 SEL_PADCTLx、SEL_DRVCTLx、SEL_PUDCTLx 寄存器决定 pin 脚的复用、驱动能力、上下拉功能控制权归 MCU 还是 CPU 所有。

7.25 MCU 通过配置 SEL_PADCTL15 寄存器控制 PWM 通道信号从 MCU 输入输出还是从 CPU 输入输出。

7.26 MCU 通过配置 SEL_PADCTL13 寄存器控制 I2S 通道信号从 MCU 输入输出还是从 CPU 输入输出以及指定 I2S 通道作输入还是输出。

7.27 通过配置 PADCTL13 (CPU 端)、SYS_PADCTL13 (MCU 端) 寄存器中 i2s oe 相关位使能 pin 脚复用为 i2s sdata 时的输入或输出功能，配置 i2s sadata in sel 相关位确定 pin 脚复用为 i2s sdata 时作数据输出还是输入用。

7.28 支持 USB NCM 模式，实现 NCM 升级、NCM 传输 log 功能，具体使用用法参考：“Doc/ AMT630HV160 NCM 模式使用说明.pdf”。

7.29 支持 DCIC 错误检测，使用 DCIC 检测需要打开 MCU 端中 DCIC_FAULT_DETECT 宏定义与 CPU 中的 ADC_KEY 宏定义和 ERROR_CHECK_TEST 宏定义，并且选择 awtk-examples 文件夹下的 Dashboard-Demo。

按键 1 按下闪烁点冻结

按键 2 按下 3 秒远光灯，近光灯切换显示

按键 3 按下右上角青色方块会变成紫色，再次按下青色，依次循环

按键 4 按下画面显示偏红，再按下恢复

按键 5 按下画面显示偏绿，再按下恢复

按键 6 按下画面显示偏蓝，再按下恢复

按键 7 按下画面显示偏亮，再按下恢复

按键 8 按下画面显示偏暗，再按下恢复

按键 9 按下进入故障检测界面，再按下恢复正常界面显示切换

7.30 支持 WRAP 畸变校正，打开 CPU 端 board.h 头文件中宏定义 WRAP_SUPPORT。把原数据通过 Jlink 写入到 srcwrap_buf 数组，同时也把校正坐标写入到 coordinatewrap_buf 中。

7.31 支持 SPI+EMMC 版本运行 (MCU 程序存储在 spi nor flash 中，CPU 程序存储在 EMMC 中)，使用时需将 MCU app 工程 amt630hv160_conf.h 文件中的 DEVICE_TYPE_SELECT 宏定义修改为 SPI_NOR_FLASH，CPU_DEVICE_TYPE_SELECT 宏定义修改为 EMMC_FLASH。同时需要将 CPU app 工程 board.h 文件中的

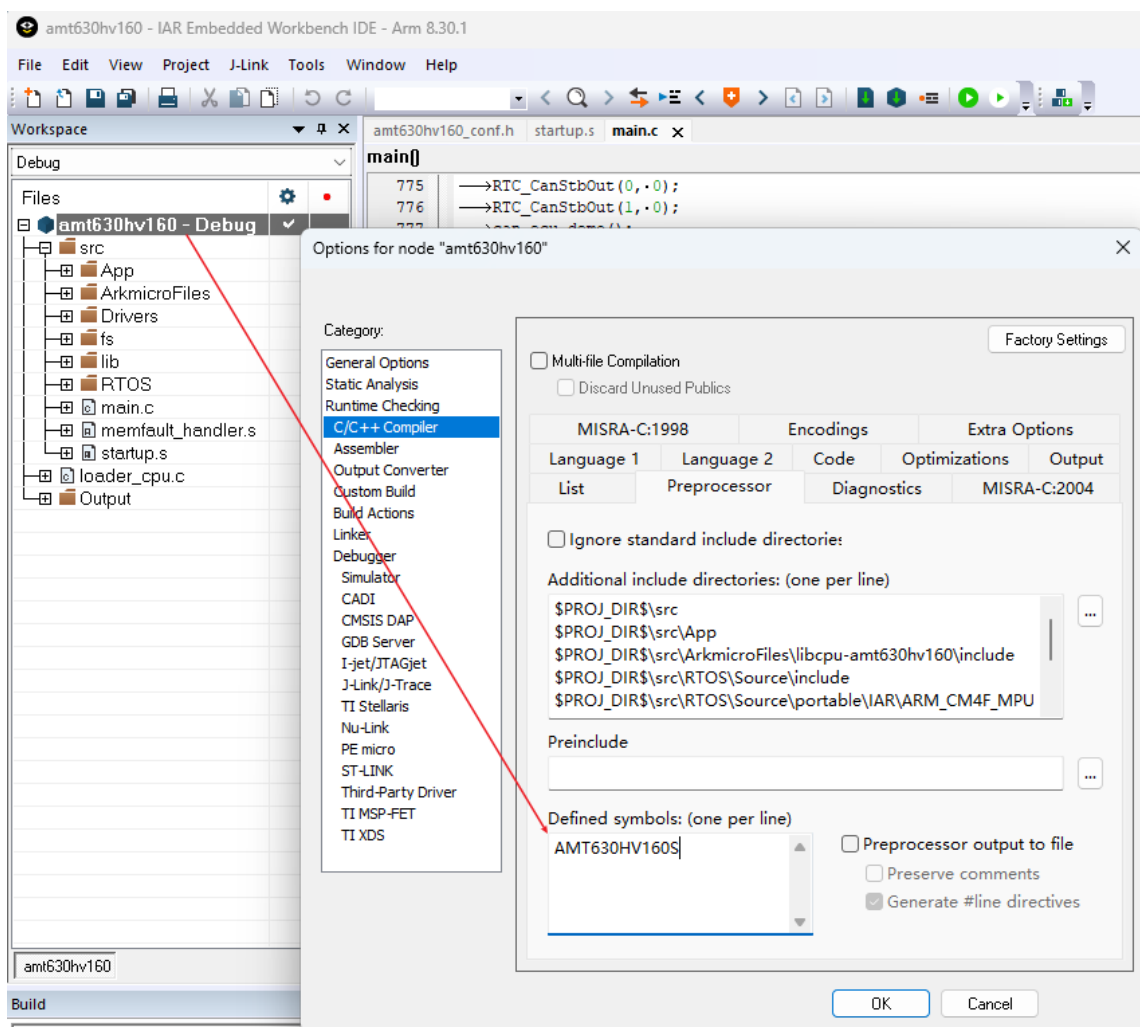
DEVICE_TYPE_SELECT 宏定义修改为 EMMC_FLASH, MCU_DEVICE_TYPE_SELECT 宏定义修改为 SPI_NOR_FLASH。

7.32 添加 AMT630HV160S 芯片支持, 默认 sdk 为 AMT630HV160 的版本, 切换至 AMT630HV160S 需修改以下部分:

1). MCU/CPU 工程需要在 IAR->options->C/C++

Compiler->Preprocessor->Defined symbols:(one per line)中添加

AMT630HV160S 宏定义, 如下: (必改项)



2). AMT630HV160 所使用的为 DDR2-64MB, 故需修改 CPU 工程 icf 文件, 最大访问至 60MB (预留 4MB 作为共享内存), 如下: (必改项)



```
F:\...\Local\amt630hv160-freertos-beta\proj\amt630hv160_awkicf
2026/6/15 18:24:30 2,562 bytes Everything Else UTF-8 UNIX
1 /*##ICF### Section handled by ICF editor, don't touch! *****/
2 /*-Editor annotation file-*/
3 /*-IcfEditorFile="$TOOLKIT_DIR$\config\ide\IcfEditor\av1_0.xml"-*/
4 /*-Specials-*/
5 define symbol _ICFEDIT_intvec_start_ = 0x60000000;
6 /*-Memory Regions-*/
7 define symbol _ICFEDIT_region_ROM_start_ = 0x60000000;
8 define symbol _ICFEDIT_region_ROM_end_ = 0x603FFFFF;
9 define symbol _ICFEDIT_region_RAM_start_ = 0x60400000;
10 define symbol _ICFEDIT_region_RAM_end_ = 0x677FFFFF;
11 /*-Sizes-*/
12 define symbol _ICFEDIT_size_cstack_ = 0x4000;
13 define symbol _ICFEDIT_size_svcstack_ = 0x1000;
14 define symbol _ICFEDIT_size_irqstack_ = 0x1000;
15 define symbol _ICFEDIT_size_fiqstack_ = 0x1000;
16 define symbol _ICFEDIT_size_undstack_ = 0x1000;
17 define symbol _ICFEDIT_size_abtstack_ = 0x4000;
18 define symbol _ICFEDIT_size_heap_ = 0;
19 /**** End of ICF editor section.***ICF###*/

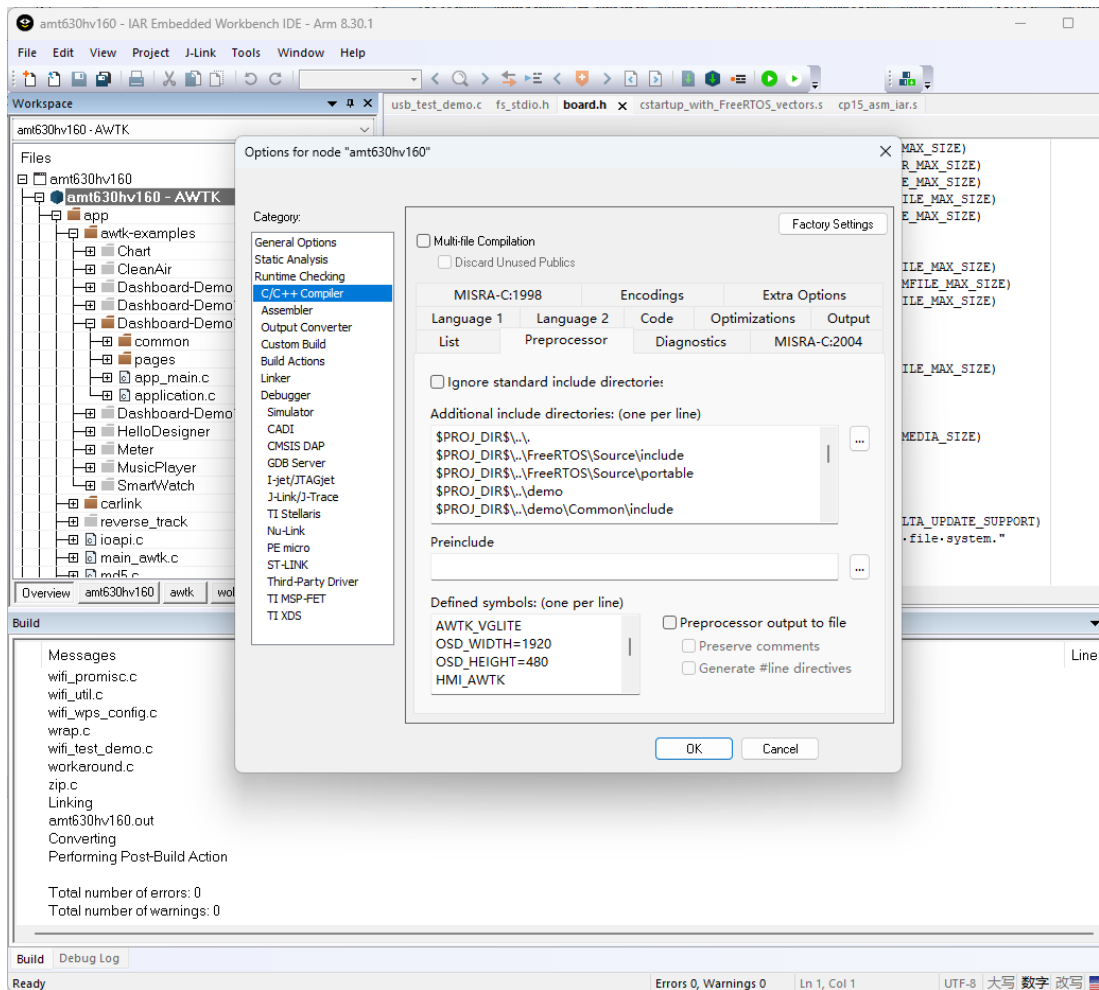
F:\...\Local\amt630hv160-freertos-beta\proj\amt630hv160_awkicf
2026/6/30 17:01:03 2,562 bytes Everything Else UTF-8 UNIX
1 /*##ICF### Section handled by ICF editor, don't touch! *****/
2 /*-Editor annotation file-*/
3 /*-IcfEditorFile="$TOOLKIT_DIR$\config\ide\IcfEditor\av1_0.xml"-*/
4 /*-Specials-*/
5 define symbol _ICFEDIT_intvec_start_ = 0x60000000;
6 /*-Memory Regions-*/
7 define symbol _ICFEDIT_region_ROM_start_ = 0x60000000;
8 define symbol _ICFEDIT_region_ROM_end_ = 0x603FFFFF;
9 define symbol _ICFEDIT_region_RAM_start_ = 0x60400000;
10 define symbol _ICFEDIT_region_RAM_end_ = 0x63BFFFFF;
11 /*-Sizes-*/
12 define symbol _ICFEDIT_size_cstack_ = 0x4000;
13 define symbol _ICFEDIT_size_svcstack_ = 0x1000;
14 define symbol _ICFEDIT_size_irqstack_ = 0x1000;
15 define symbol _ICFEDIT_size_fiqstack_ = 0x1000;
16 define symbol _ICFEDIT_size_undstack_ = 0x1000;
17 define symbol _ICFEDIT_size_abtstack_ = 0x4000;
18 define symbol _ICFEDIT_size_heap_ = 0;
19 /**** End of ICF editor section.***ICF###*/
```

3). 由于 AMT630HV160S 默认 demo 代码为单 SPI_NOR_FLASH，故需将 MCU 工程的 amt630hv160_conf.h 和 CPU 工程的 board.h 中将设备类型切换为 SPI_NOR_FLASH，如下：（可选）

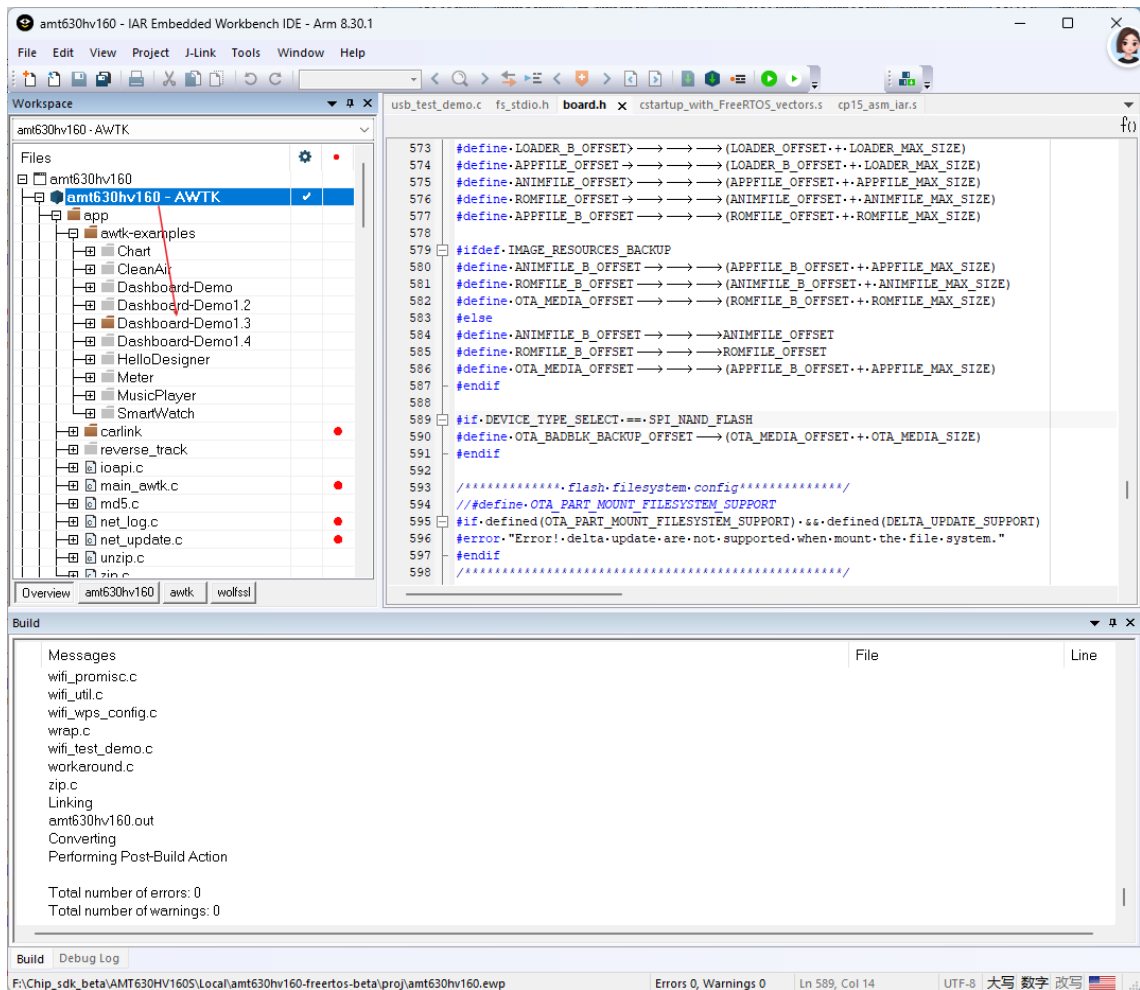
```
F:\...\Local\amt630hv160-mcu\amt630hv160-mcu-iram\src\amt630hv160_conf.h
2026/7/1 16:17:47 21,798 bytes C,C++,C#,ObjC Source UTF-8 UNIX
324 #define LCD_TIMING_HSW >>> 12
325 #define LCD_CLK_FREQ >>> 40000000
326 #elif (LCD_WIDTH == 480 && LCD_HEIGHT == 1280)
327 #define LCD_TIMING_VBP >>> 147/6
328 #define LCD_TIMING_VFP >>> 16/6
329 #define LCD_TIMING_VSW >>> 2/16
330 #define LCD_TIMING_HBP >>> 16
331 #define LCD_TIMING_HFP >>> 26/36
332 #define LCD_TIMING_HSW >>> 26
333 #define LCD_CLK_FREQ >>> 50000000
334 #elif (LCD_WIDTH == 1920 && LCD_HEIGHT == 480)
335 #define LCD_TIMING_VBP >>> 8
336 #define LCD_TIMING_VFP >>> 8
337 #define LCD_TIMING_VSW >>> 3
338 #define LCD_TIMING_HBP >>> 16
339 #define LCD_TIMING_HFP >>> 50
340 #define LCD_TIMING_HSW >>> 12
341 #define LCD_CLK_FREQ >>> 60000000
342 #else
343 #error "no lcd timing configuration."
344 #endif
345
346 #define FB_SIZE >>> (LCD_WIDTH * LCD_HEIGHT * LCD_BPP / 8)
347 /*****/
348
349 /***** flash type configuration *****/
350 #define SPI_NOR_FLASH >>> 0
351 #define SPI_NAND_FLASH >>> 1
352 #define EMMC_FLASH >>> 2
353 //MCU使用EMMC启动存在初始化失败的风险，不建议
354 #define DEVICE_TYPE_SELECT >>> SPI_NOR_FLASH //需要与CPU工程board.h中
355 #define CPU_DEVICE_TYPE_SELECT >>> EMMC_FLASH //需要与CPU工程board.h中

F:\...\Local\amt630hv160-mcu\amt630hv160-mcu-iram\src\amt630hv160_conf.h
2026/7/1 16:13:23 22,413 bytes C,C++,C#,ObjC Source UTF-8 PC
324 #define LCD_TIMING_HSW >>> 12
325 #define LCD_CLK_FREQ >>> 40000000
326 #elif (LCD_WIDTH == 480 && LCD_HEIGHT == 1280)
327 #define LCD_TIMING_VBP >>> 147/6
328 #define LCD_TIMING_VFP >>> 16/6
329 #define LCD_TIMING_VSW >>> 2/16
330 #define LCD_TIMING_HBP >>> 16
331 #define LCD_TIMING_HFP >>> 26/36
332 #define LCD_TIMING_HSW >>> 26
333 #define LCD_CLK_FREQ >>> 50000000
334 #elif (LCD_WIDTH == 1920 && LCD_HEIGHT == 480)
335 #define LCD_TIMING_VBP >>> 8
336 #define LCD_TIMING_VFP >>> 8
337 #define LCD_TIMING_VSW >>> 3
338 #define LCD_TIMING_HBP >>> 16
339 #define LCD_TIMING_HFP >>> 50
340 #define LCD_TIMING_HSW >>> 12
341 #define LCD_CLK_FREQ >>> 60000000
342 #else
343 #error "no lcd timing configuration."
344 #endif
345
346 #define FB_SIZE >>> (LCD_WIDTH * LCD_HEIGHT * LCD_BPP / 8)
347 /*****/
348
349 /***** flash type configuration *****/
350 #define SPI_NOR_FLASH >>> 0
351 #define SPI_NAND_FLASH >>> 1
352 #define EMMC_FLASH >>> 2
353 //MCU使用EMMC启动存在初始化失败的风险，不建议
354 #define DEVICE_TYPE_SELECT >>> SPI_NOR_FLASH //需要与CPU工程board.h
355 #define CPU_DEVICE_TYPE_SELECT >>> SPI_NOR_FLASH //需要与CPU工程boa
```

4). 由于 AMT630HV160S 的默认 demo 屏幕为 1920-480 的单 lvdS 屏幕，故需将 CPU 工程的 IAR->options->C/C++ Compiler->Preprocessor->Defined symbols:(one per line)中的 OSD_WIDTH 和 OSD_HEIGHT 修改为 1920、480，如下：（可选）



5). 由于 AMT630HV160S 的默认 demo 为 Dashboard-Demo1.3/1/4, 故需在 CPU 工程上选择这两个其中一个, 如下: (可选)



备注:

- 文中出现的一些代码文件完整路径如下:

amt630hv160-freertos-beta\ArkmicroFiles\libboard-amt630hv160\include\board.h

amt630hv160-freertos-beta\ArkmicroFiles\libcpu-amt630hv160\source\pinctrl.c

amt630hv160-freertos-beta\app\main_awtk.c

amt630hv160-freertos-beta\FreeRTOSConfig.h

amt630hv160-freertos-beta\lib\awtk\awtk\src\platforms\ark\ platform.c

- 文中提到的修改工程设置，添加或者删除某个符号定义是指下图所示位置添加或者删除。

