

## MPLAB® Harmony 3 之基础篇（06）

### 一 基于 SAME70 LCC 创建图形工程

Microchip Technology Inc.  
MCU32 产品部

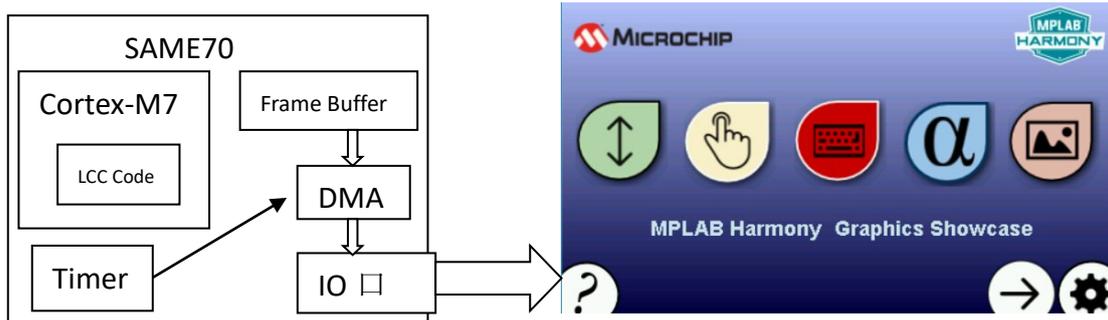
#### 一、 简介

随着时代的发展，电子产品对图形显示的要求越来越高，Microchip 的 Harmony 3 应运而生。图形开发是 Harmony 3 一个非常重要的应用，本文介绍基于 SAME70 的 LCC 方案创建图形工程的完整流程。利用 Harmony 3 的强大配置工具实现图形显示，从 SAME70 的配置，到图形界面的设计以及最终的代码生成和编译下载，整个过程不需要编写一行代码。

#### 二、 关于 Low-Cost Controllerless (LCC) 图形方案

LCC 是 Microchip 提供的一项技术，是一种的低成本无 LCD 驱动器实现 LCD 屏图形显示的解决方案。

LCC 方案利用 DMA 和 Timer 结合软件编码在 IO 口上输出满足 LCD 屏时序要求的控制信号和 RGB 信号（如下图所示）。由于使用 DMA，不会占用大量 CPU 时间，可以直接驱动 LCD，免除了外部图形控制器的成本和复杂性。



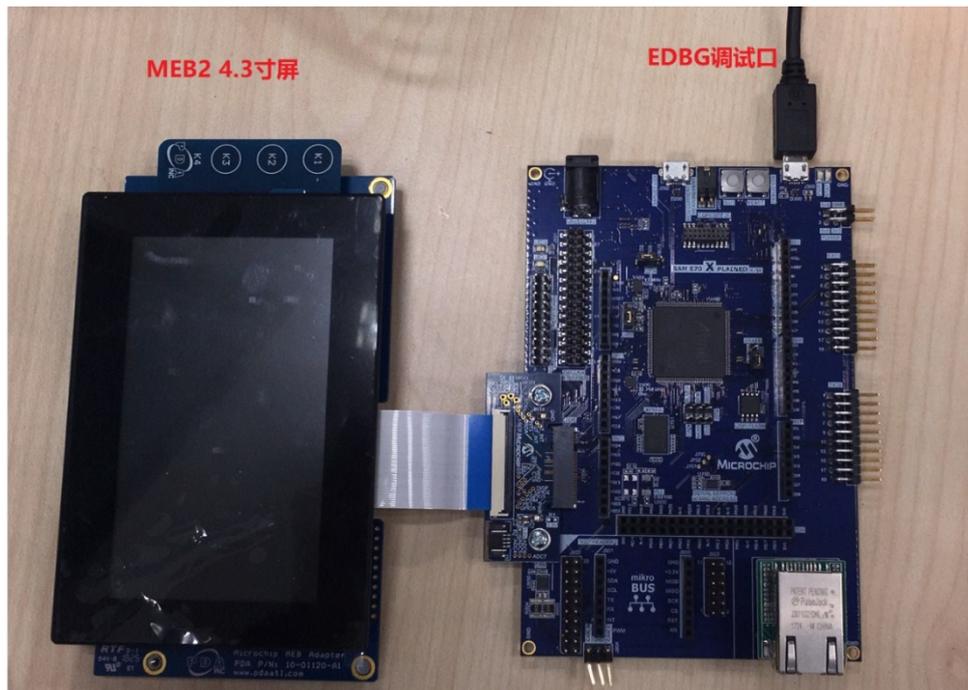
### 三、 硬件工具和软件平台

**硬件:**

SAM E70 Xplained Ultra 板和 MEB2 的 4.3 寸屏

EDBG 调试口通过 USB 连接 PC 的 USB 口

MEB2 的 4.3 寸屏与 Ultra 板的 Graphics card interface 接口连接



**软件:**

MPLAB® X IDE: v5.10 或者更新

XC32: v2.15 或者更新

Harmony 3: v3.10 或者更新

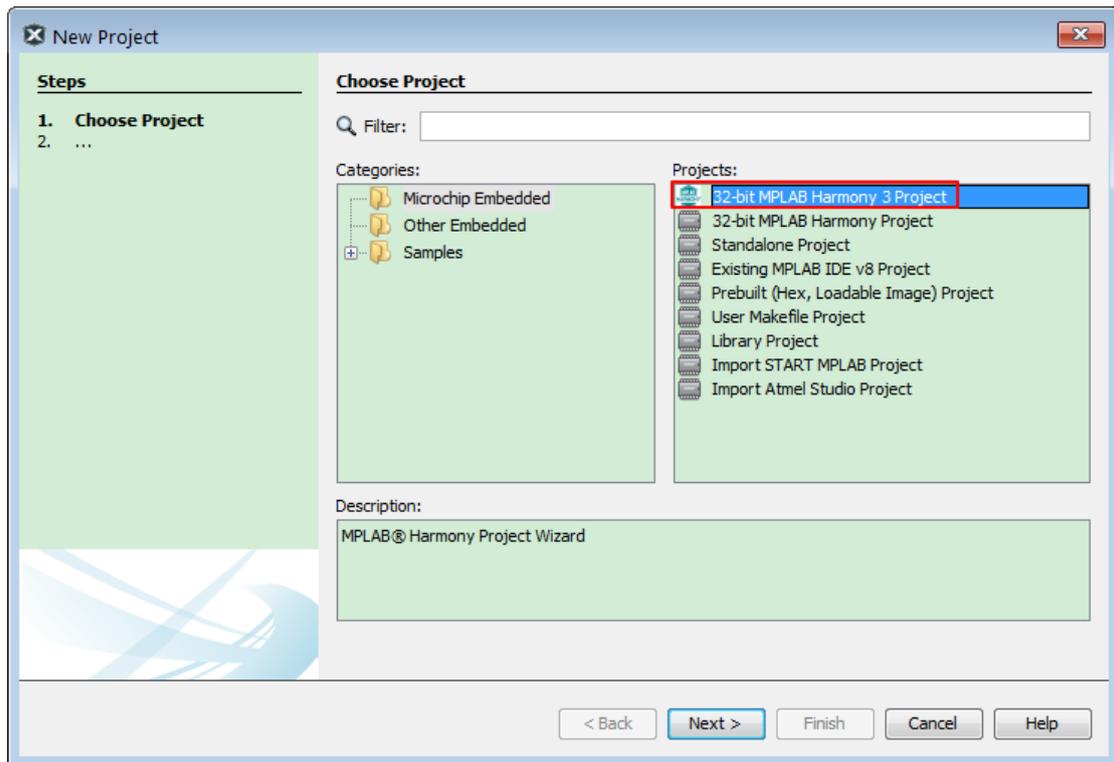
## 四、 详细步骤

下面开始一步一步创建图形工程。

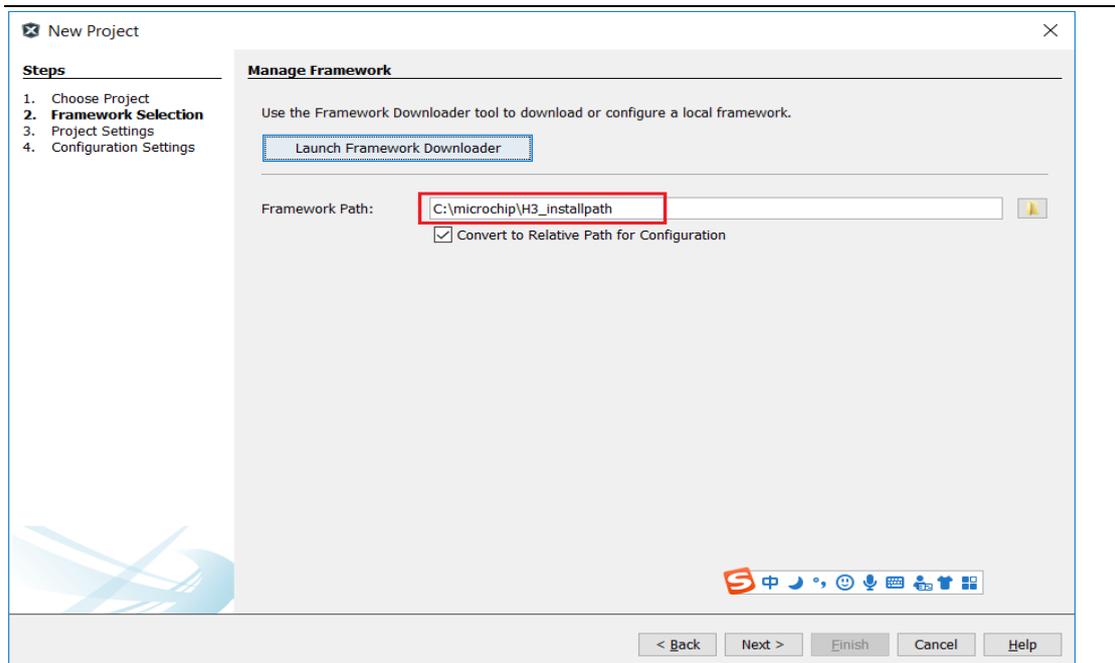
注：以下没有特别标注出来的地方，说明使用的是默认选项。

### (一) 新建 Harmony 3 项目

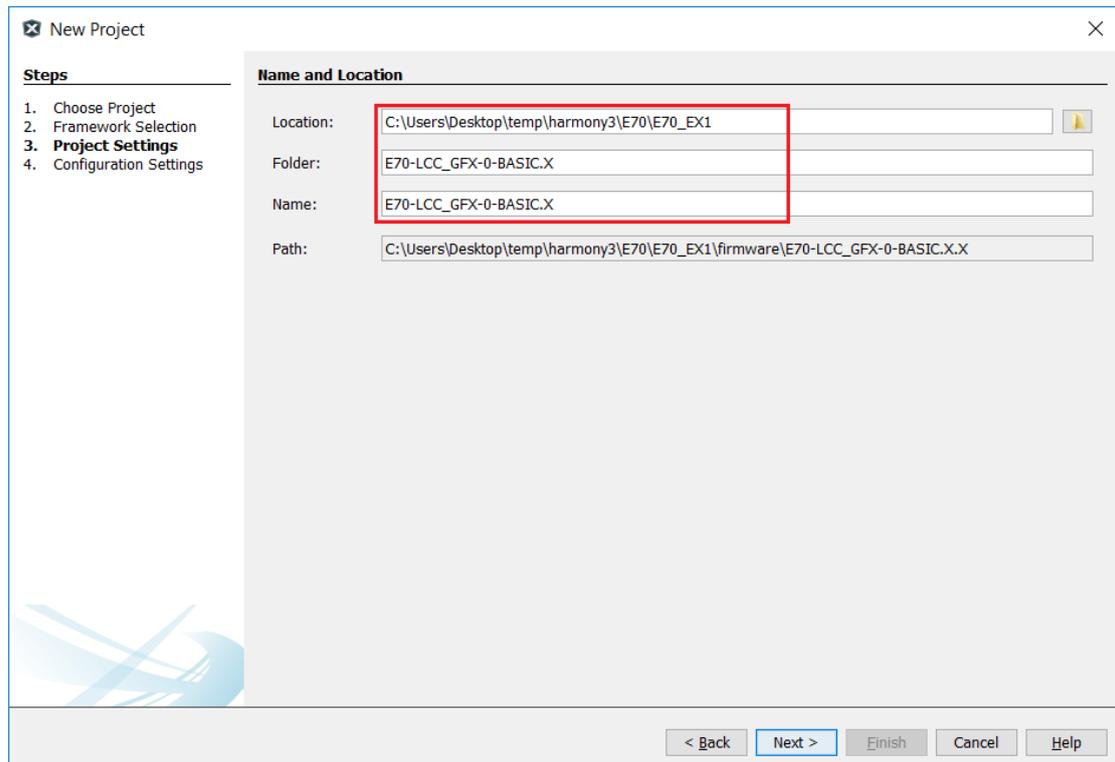
在 MPLAB X IDE 里点击 File -> New Project, 选择“32-bit MPLAB Harmony Project”，然后点击“Next”按钮。



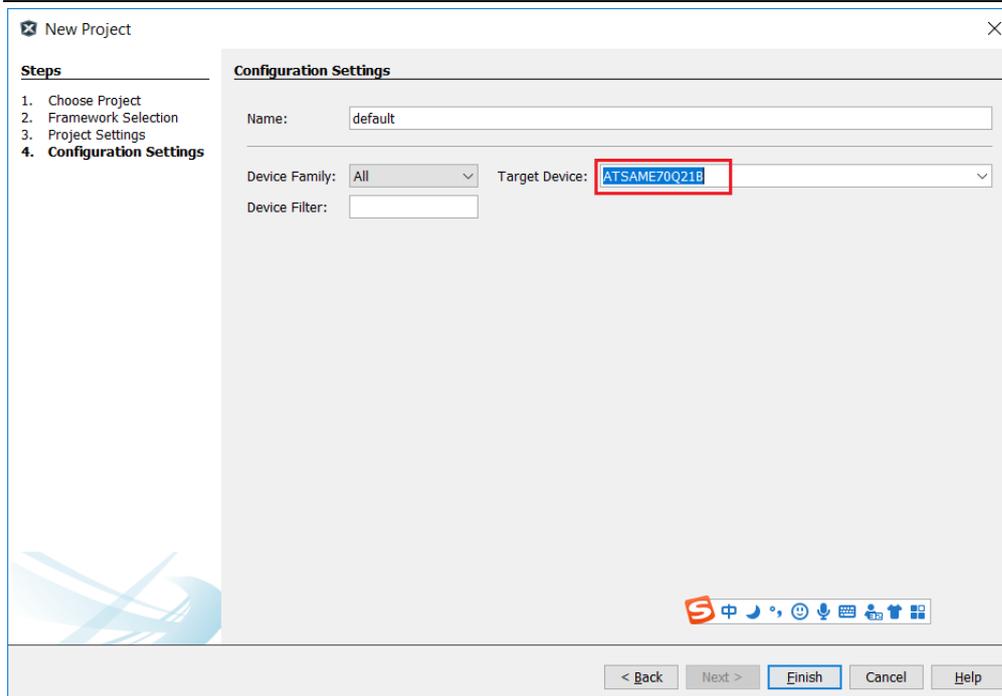
选择“Harmony Framework”的实际路径，然后点击“Next”按钮。



填写项目路径和项目名称，然后点击“Next”按钮。

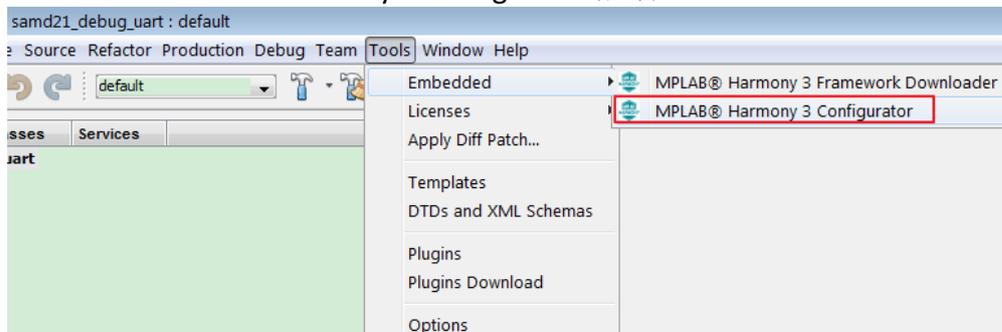


选择芯片类型“ATSAME70Q21B”，最后点击“Finish”按钮。

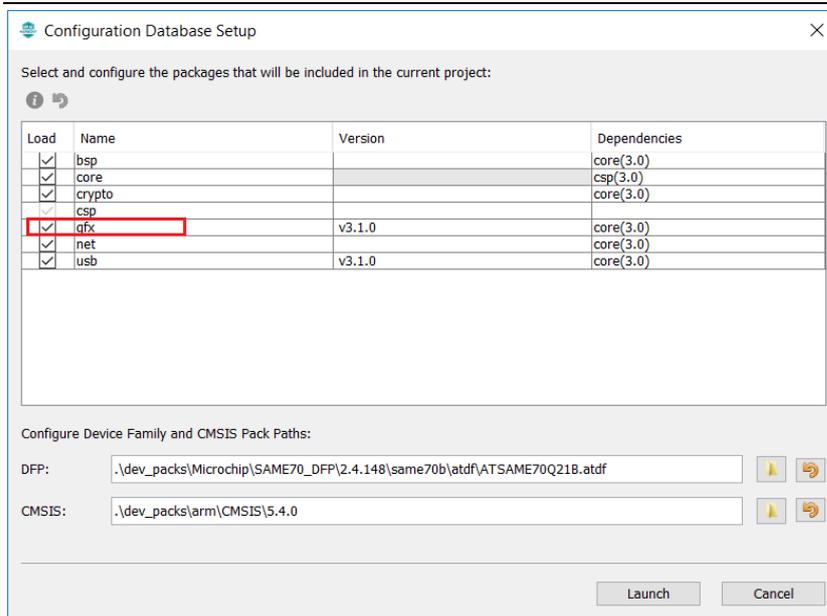


## (二) 使用 MHC 来配置 SAME70 Xplained Ultra 板

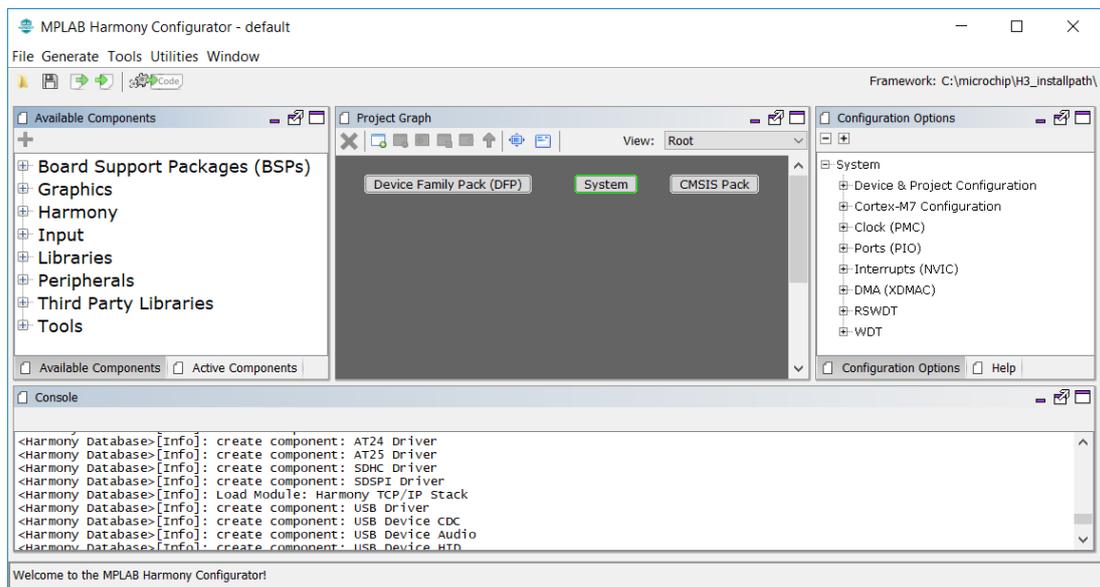
第一次创建项目时，MHC 界面会自动启动。或者手动在 X IDE 里点击 Tools -> Embedded -> MPLAB Harmony 3 Configurator 启动：



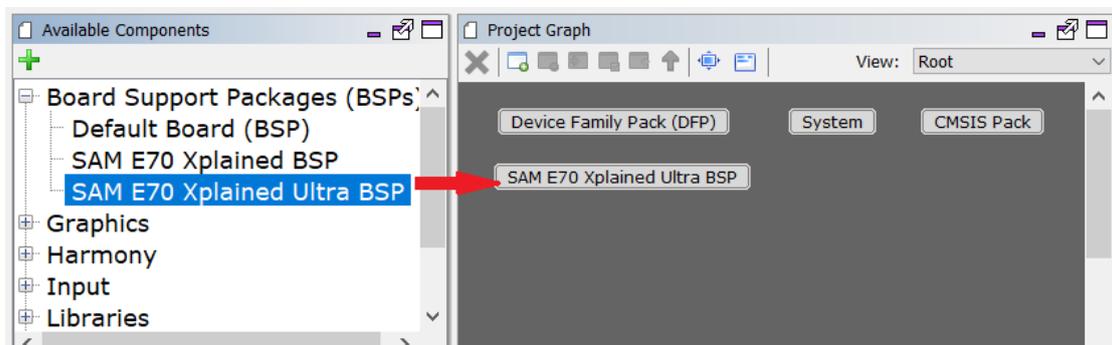
勾选上 gfx，点击“Launch”按钮启动 MHC。



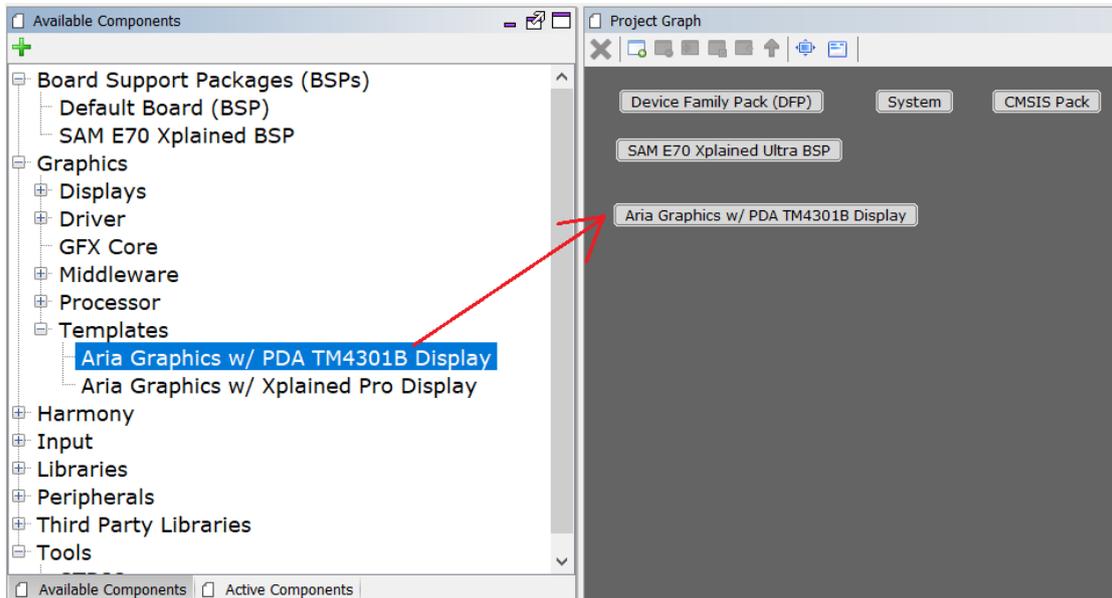
MHC 界面如下图。



添加 BSP，将“SAM E70 Xplained Ultra BSP”从“Available Components”窗口拖放到“project Graph”窗口。

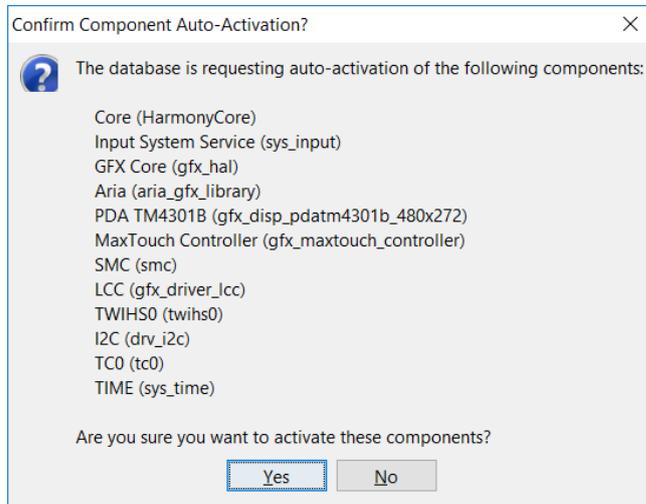


同样方法，添加模板 “Aria Graphics w/ PDA TM4301B Display”。

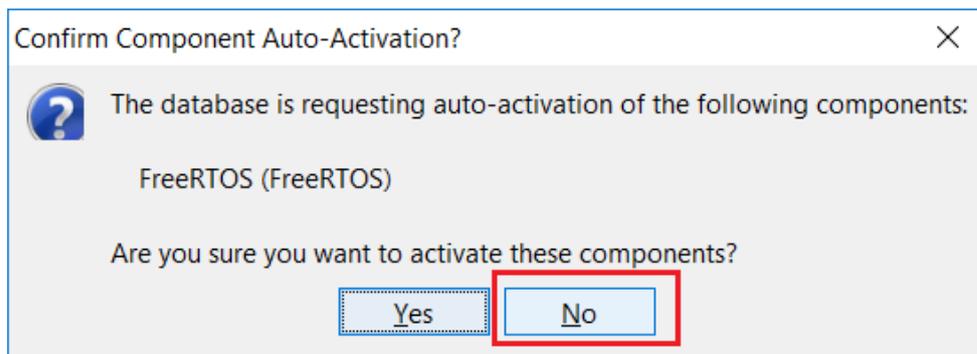


在添加模板 “Aria Graphics w/ PDA TM4301B Display”之后，会自动弹出一系列对话框，让用户选择，具体如下：

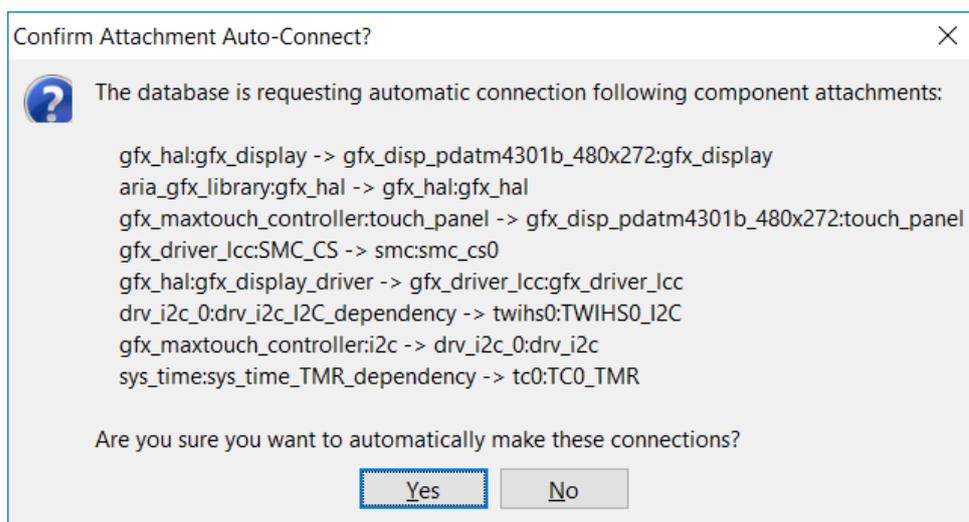
确认是否自动激活该模板相应的组件，选择 yes。



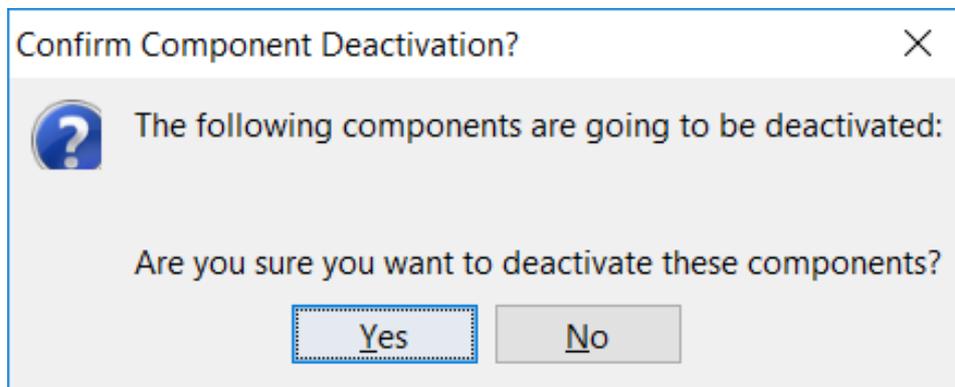
确认是否激活 FreeRTOS, 本例不需要 OS, 选择 No。



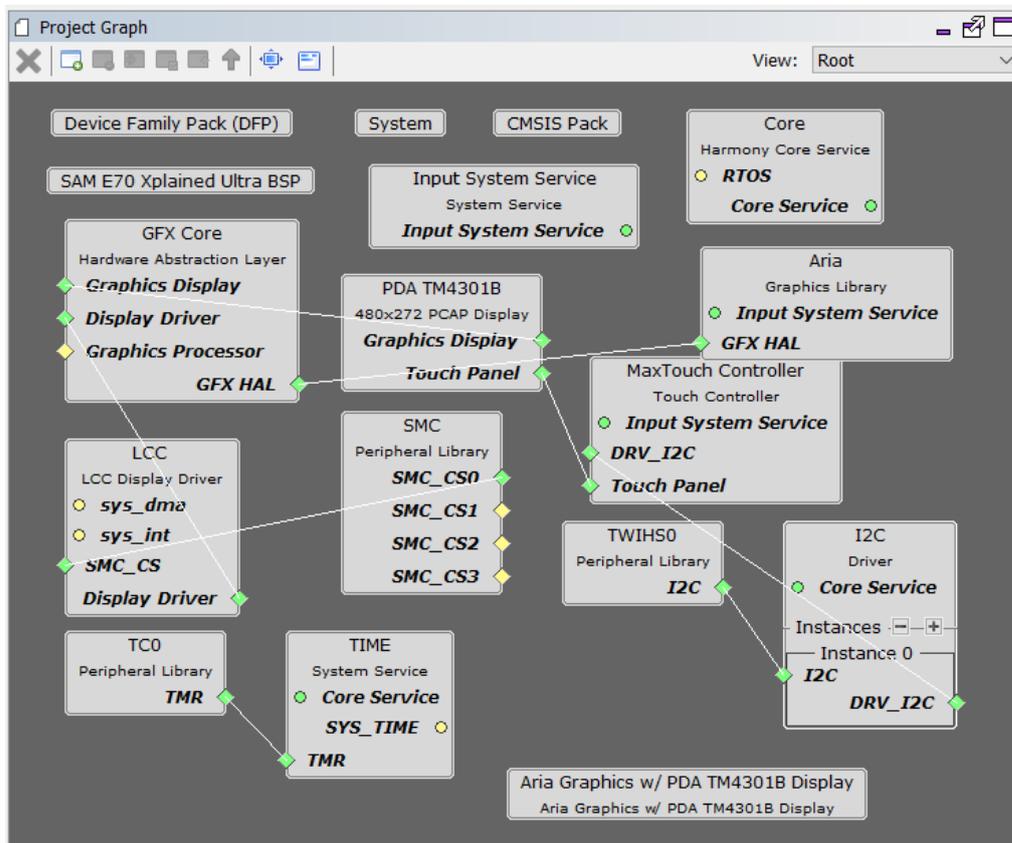
确认是否自动连线, 选择 yes。



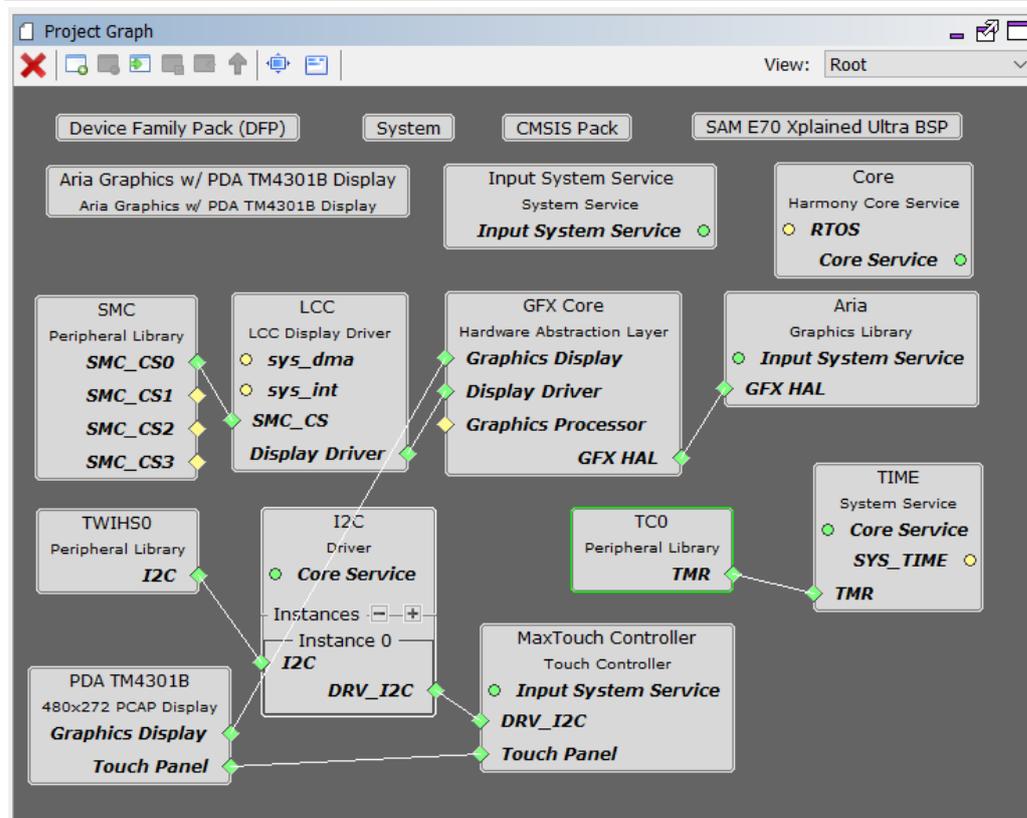
确认是否解除不用的组件, 选择 yes。



组件添加和连线完成，如下图。



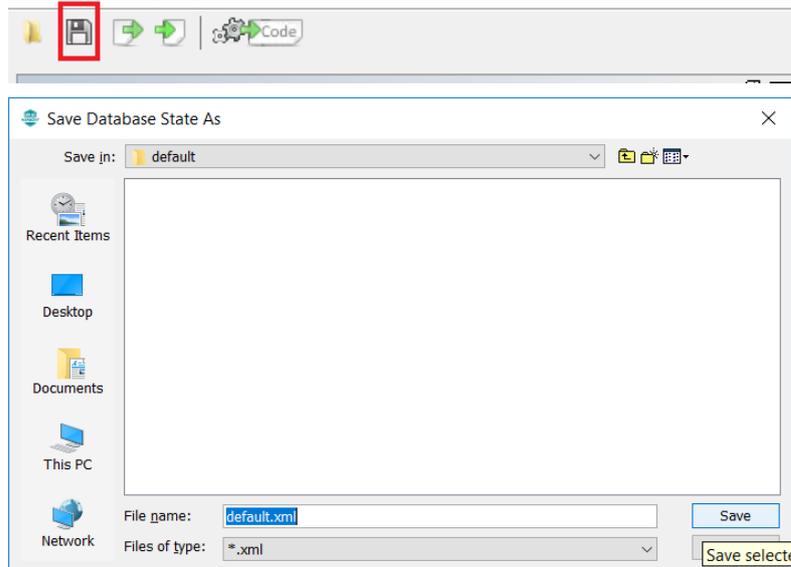
根据自己的喜好，调整一下组件的位置。



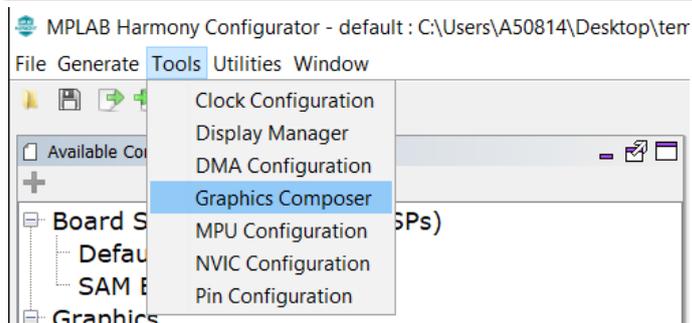
点击保存按钮来保存配置，文件名可以使用默认的即可。

MPLAB Harmony Configurator - default\*

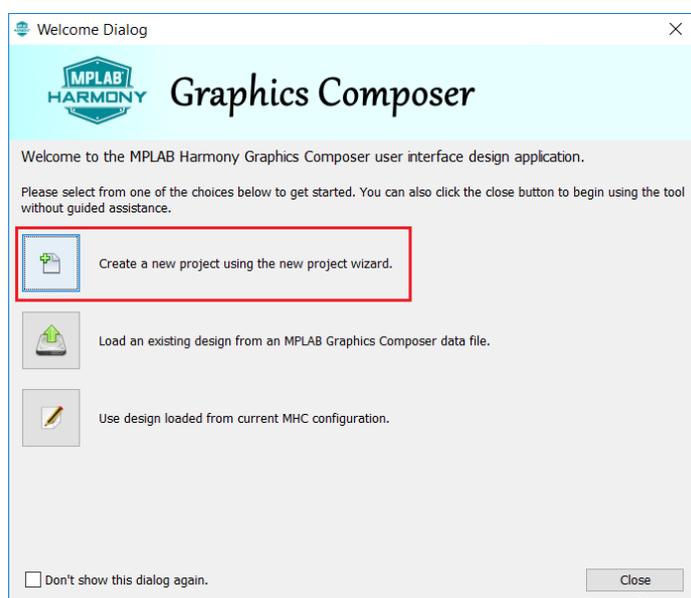
File Generate Tools Utilities Window



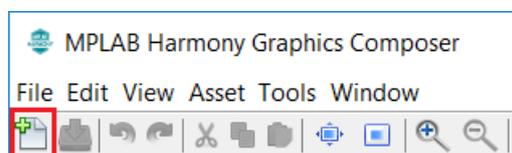
(三) 使用 MHGC(Microchip Harmony Graphics Composer)创建图形设计  
MHC 主菜单 Tools-> Graphics Composer 来启动 MHGC。



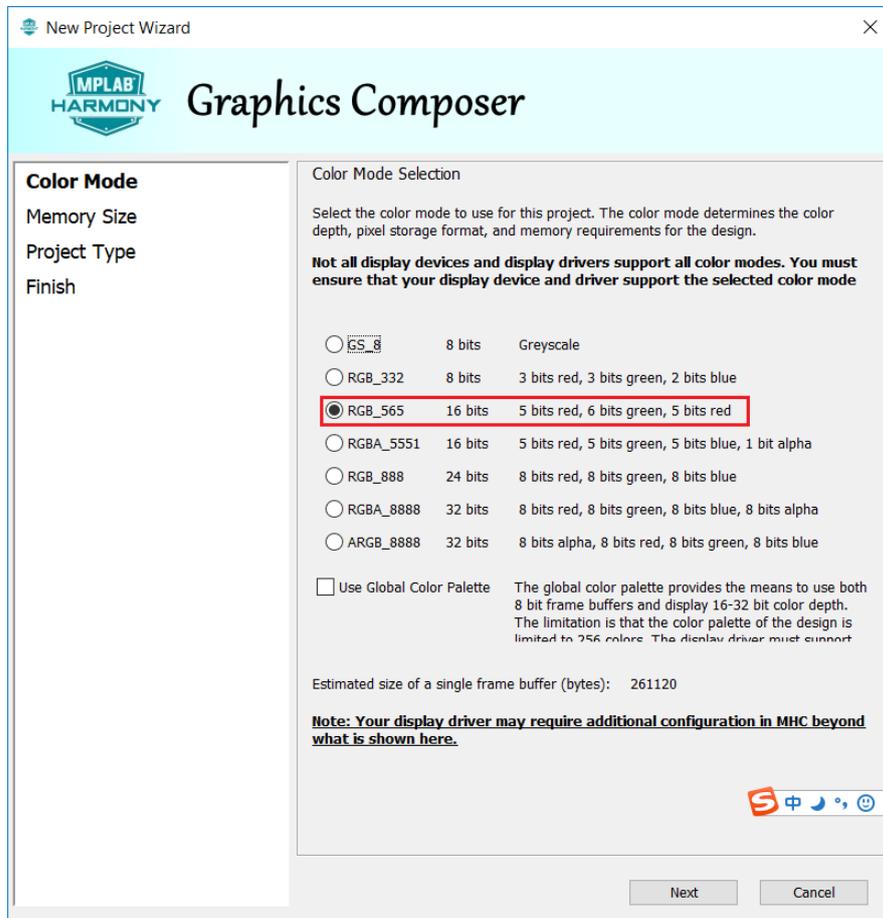
在弹出的“Welcome Dialog”中点击第一个按钮，来打开“new project wizard”对话框。



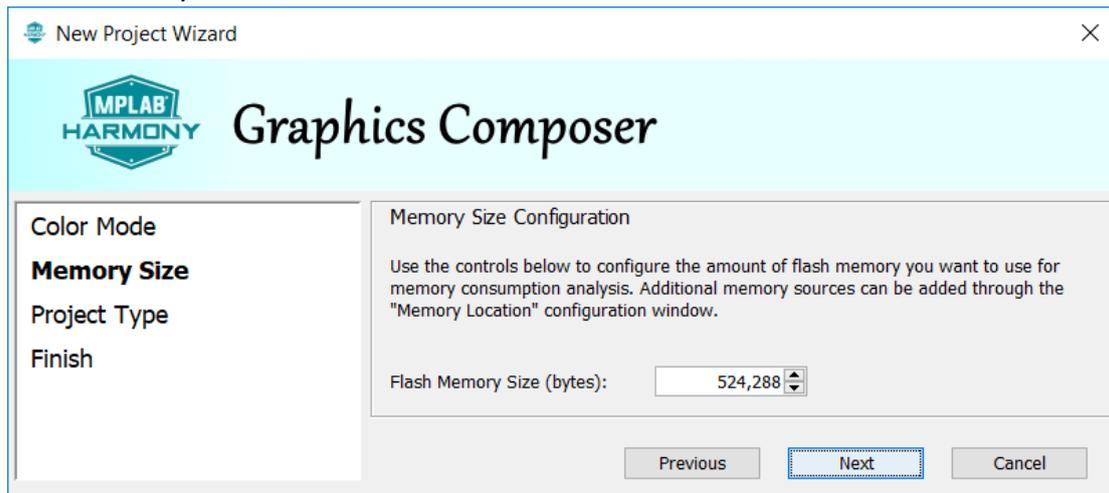
如果“Welcome Dialog”没有自动弹出，则在 MHGC 的工具栏中点击如下按钮来打开“new project wizard”对话框。关于如何配置 Welcome Dialog 自动弹出，请参考附录章节。



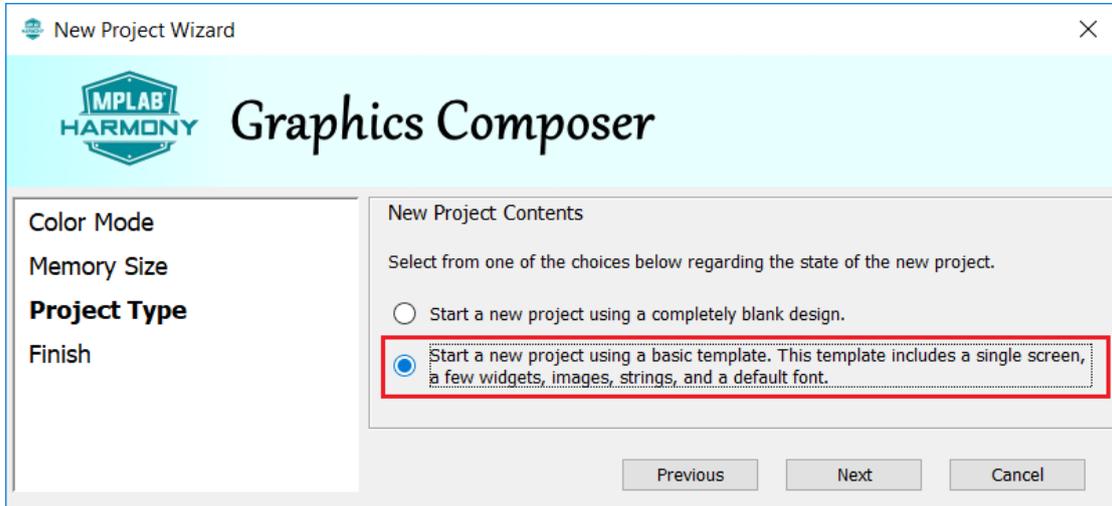
Color mode 选择 RGB565，然后点击 Next 按钮



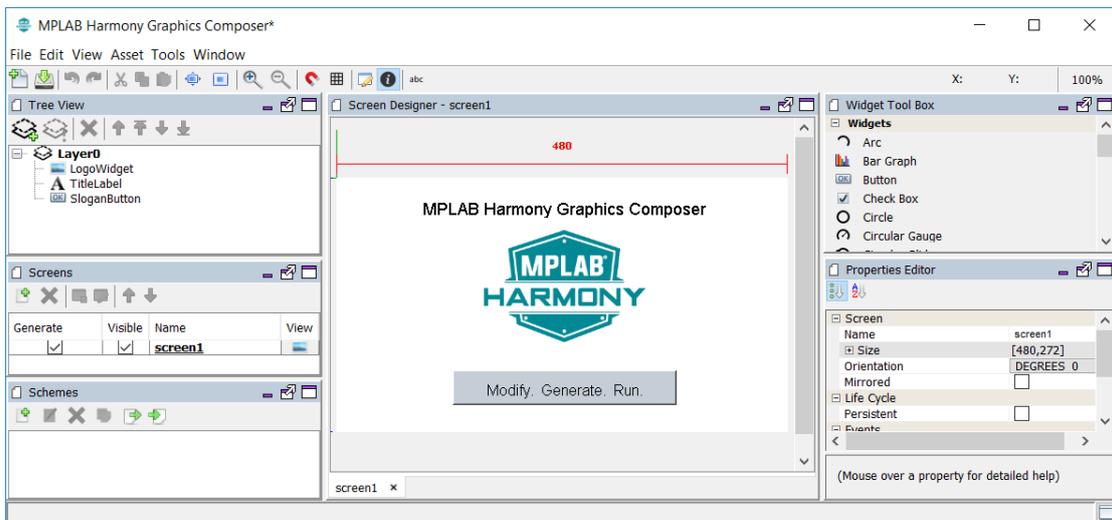
Flash memory size 使用默认值，然后点击 Next 按钮。



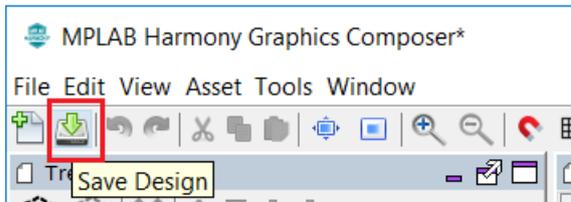
选择“Start a new project using a basic template”，即使用基础模板，然后点击 Next 按钮。



最后点击完成，Screen Designer 窗口出现基础模板的图形。

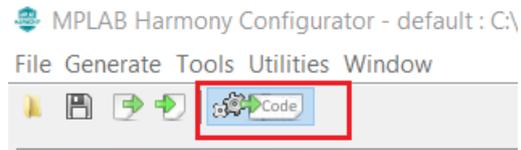


图形工程创建结束，保存工程并关闭 MHGC。

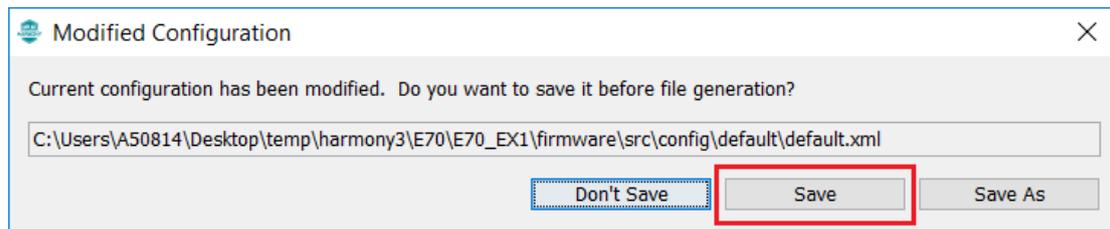


#### (四) 使用 MHC 生成代码

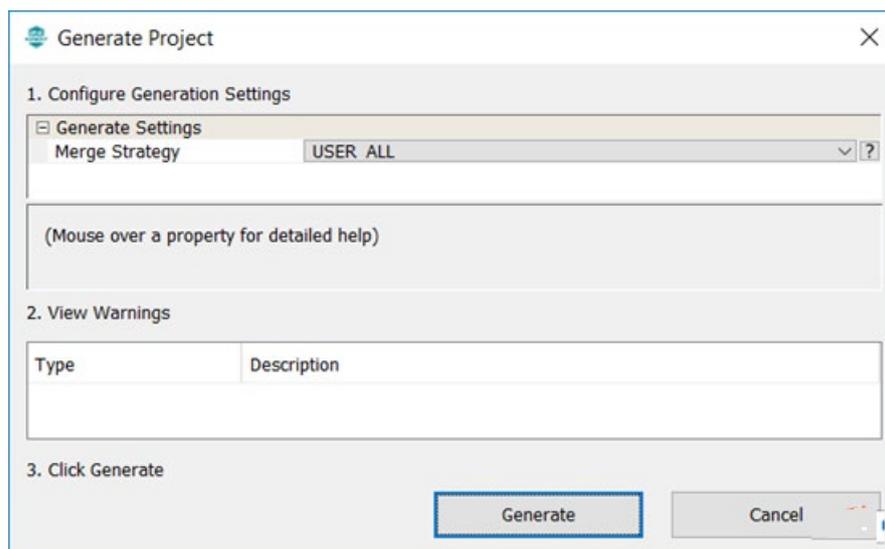
回到 MHC 界面，点击 generate code 按钮来生成代码。



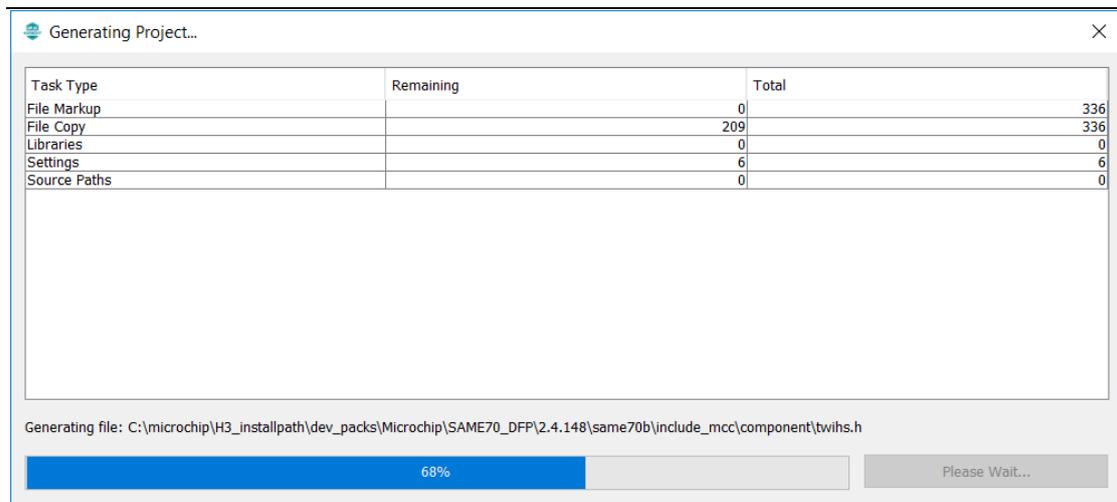
是否保存配置，点击 Save 按钮。



点击 generate 按钮。关于 Merge Strategy，请参考附录章节。



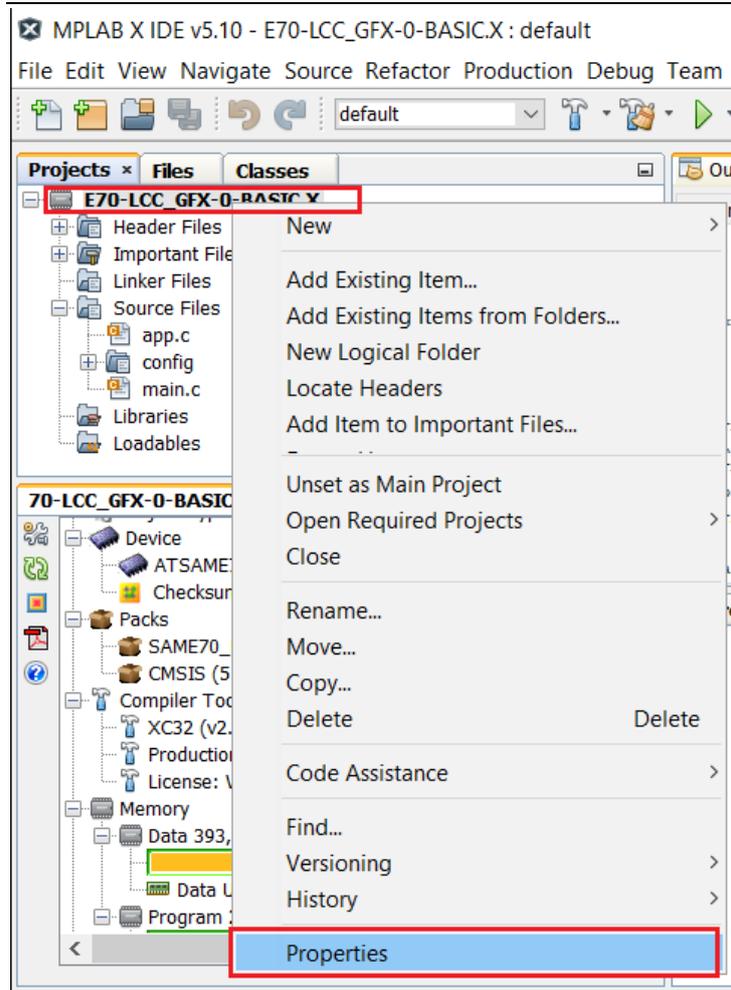
生成代码中。



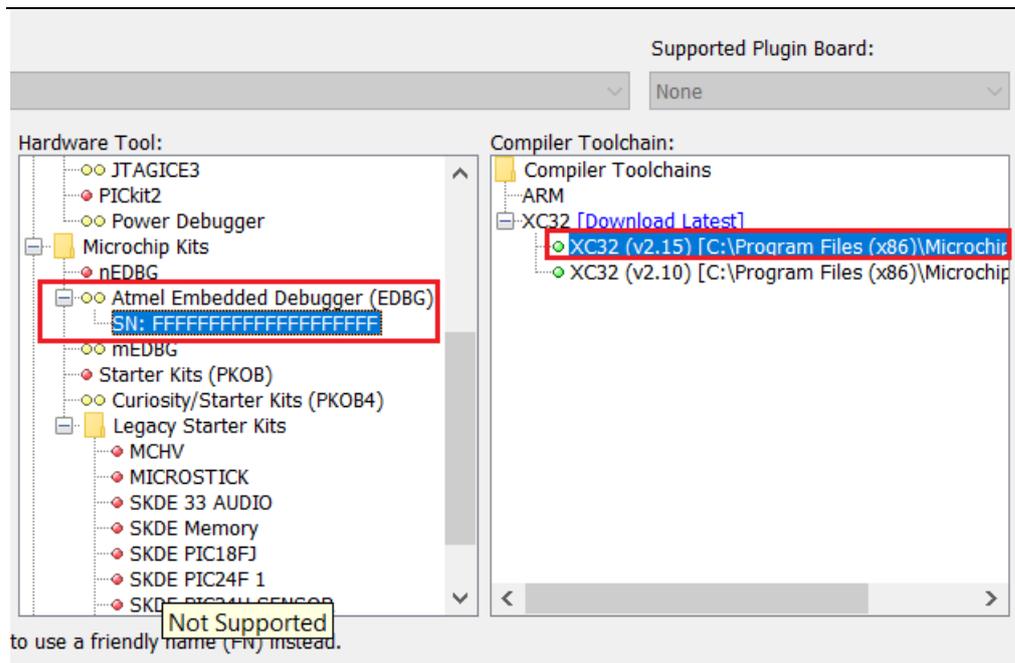
代码生成结束后，可以关闭 MHC。

#### (五) 编译下载测试

在 MPLAB X IDE 的 projects 窗口，右键工程选择 properties 来打开工程属性窗口。



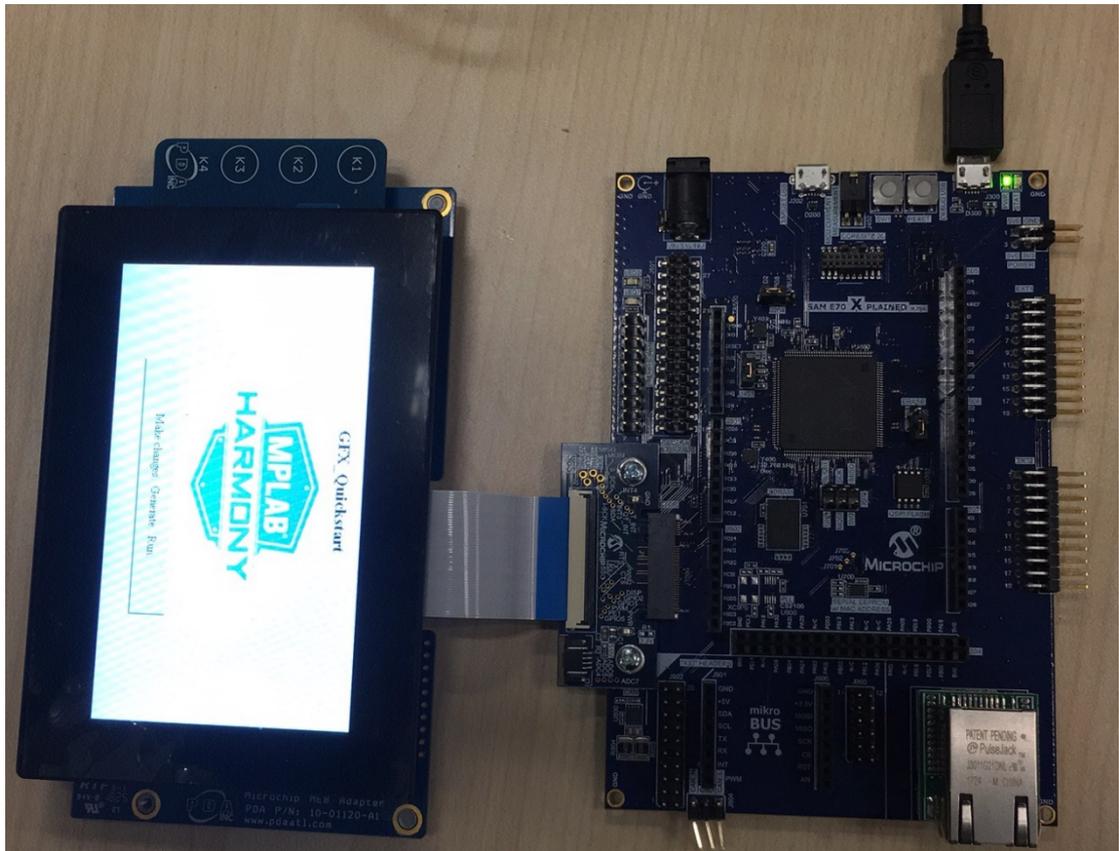
选择 SAME70 Xplained Ultra 板对应的 EDBG 口和 XC32 编译器。



点击工具栏的“Make and Program”来完成编译和烧录。



烧录成功后，运行结果如下图。



## 五、 总结

本文通过 MHC 配置 SAME70 Xplained Ultra 板和利用 MHGC 创建图形工程，实现图形显示功能。

更多图形示例工程和帮助文档请参考：

Harmony3 安装目录\gfx\_apps

Harmony3 安装目录\gfx\_apps\doc\help\_apps\_gfx.chm

关于 LCC，可以参考 App note：

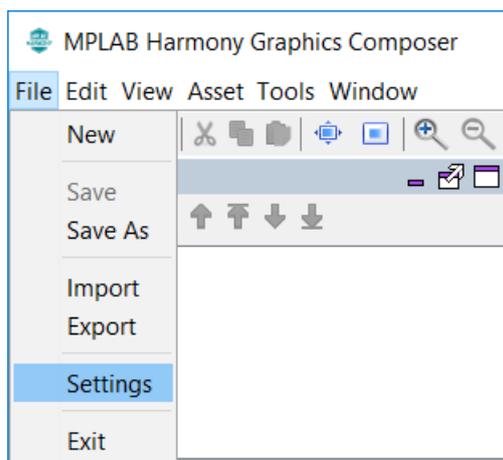
Using PIC32 MCUs to Develop Low-Cost Controllerless (LCC) Graphics Solutions

<http://ww1.microchip.com/downloads/en/AppNotes/00001387C.pdf>

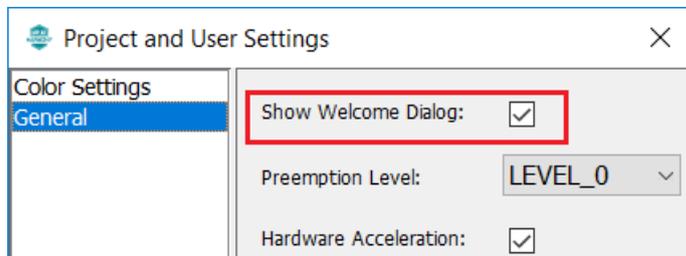
## 六、 附录

### (一) 配置自动弹出 **Welcome Dialog** 对话框

在 MHGC 的主菜单点击 File -> Settings，弹出 Project and User Settings 对话框。

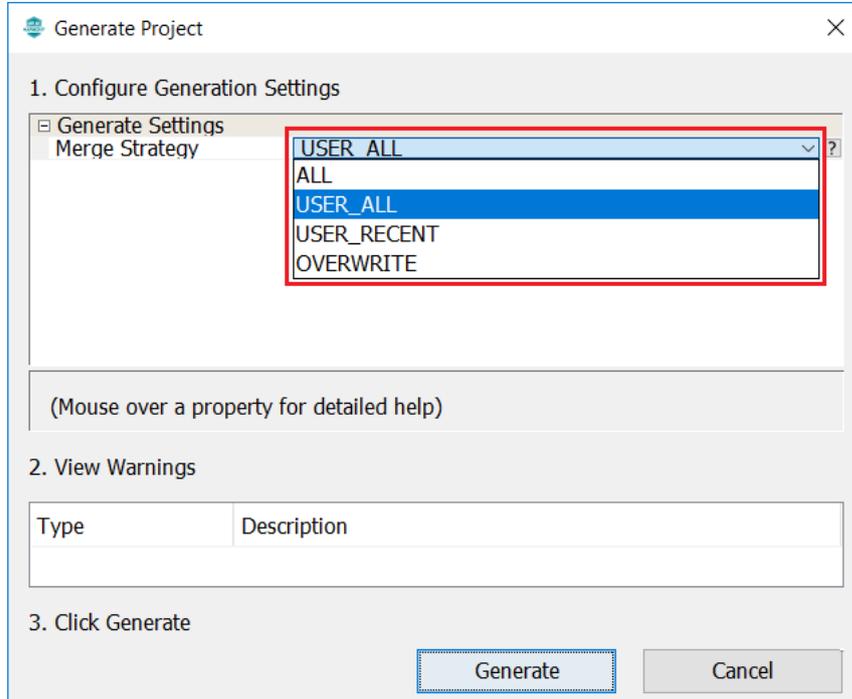


在 Project and User Settings 的 General 页面，勾选 Show Welcome Dialog 复选框。



## (二) 关于 Merge Strategy

在 Generate Project 对话框中有 4 种 Merge Strategy。具体解释如下：



ALL: MHC 生成的所有文件都弹出合并窗口。

USER\_ALL: MHC 生成的并且用户做过改动的文件弹出合并窗口。

USER\_RECENT: MHC 生成的并且用户近期做过改动的文件弹出合并窗口。

OVERWRITE: MHC 生成的文件直接覆盖以前生成的文件，即不会弹出合并窗口。

关于合并窗口，下图左边窗口是新生成文件内容，右边窗口是原始的文件内容，如果需要新生成的内容替换旧文件中对应的内容，点击箭头按钮进行替换可。如果想保留原始文件的内容，不进行任何操作即可。



Merging: libaria\_demo\_mode.c\_411933553158288826.old

Pending Merge Actions: 1

Note: The content of the right pane displays the current state of this merge.

Generated Code	1/1	Current File: libaria_demo_mode.c_411933553158288826.old
<pre> demoModeEvents.demoEvents[demoModeEvents.numEvents].y = y; demoModeEvents.numEvents++; } }  void LibAria_DemoModeRecordInputEvent (DEMO_MODE_INPUT_TYPE ie, int32_t index, int32_t x, int32_t y) { }  static void LibAria_LoadDefaultEvents(void) { // Add default input events here. These will be loaded if no events were // recorded. // ex. LibAria_DemoModeAddInputEvent(delay, index, event, x, y); }  void LibAria_DemoModeTouchDownHandler(const SYS_INP_TouchStateEvent* const evt) { LibAria_DemoModeSendEvent(DEMO_EVENT_INPUT); }  void LibAria_DemoModeTouchUpHandler(const SYS_INP_TouchStateEvent* const evt) { LibAria_DemoModeSendEvent(DEMO_EVENT_INPUT); } </pre>	<pre> 129 int32_t index, 130 int32_t x, 131 int32_t y) 132 { 133 } 134 135 static void LibAria_LoadDefaultEvents(void) 136 { 137 // START OF CUSTOM CODE. DO NOT MODIFY OR REMOVE!!! 138 LibAria_DemoModeAddInputEvent(0x217a, DEMO_MODE_INPUT_PRESS, 0, 0x26, 0x07F); 139 LibAria_DemoModeAddInputEvent(0x82, DEMO_MODE_INPUT_RELEASE, 0, 0x26, 0x07F); 140 141 //Turn the list wheels 142 LibAria_DemoModeAddInputEvent(0x384, DEMO_MODE_INPUT_PRESS, 0, 0x87, 0x09D); 143 LibAria_DemoModeAddInputEvent(0x28, DEMO_MODE_INPUT_MOVE, 0, 0x87, 0x99); 144 LibAria_DemoModeAddInputEvent(0x14, DEMO_MODE_INPUT_MOVE, 0, 0x88, 0x93); 145 LibAria_DemoModeAddInputEvent(0x14, DEMO_MODE_INPUT_MOVE, 0, 0x8A, 0x08C); 146 LibAria_DemoModeAddInputEvent(0xA, DEMO_MODE_INPUT_MOVE, 0, 0x8D, 0x07D); 147 LibAria_DemoModeAddInputEvent(0x14, DEMO_MODE_INPUT_MOVE, 0, 0x91, 0x62); 148 LibAria_DemoModeAddInputEvent(0xA, DEMO_MODE_INPUT_MOVE, 0, 0x96, 0x45); 149 LibAria_DemoModeAddInputEvent(0x001E, DEMO_MODE_INPUT_RELEASE, 0, 0x96, 0x4); 150 LibAria_DemoModeAddInputEvent(0x02C6, DEMO_MODE_INPUT_PRESS, 0, 0xCE, 0x09B); 151 LibAria_DemoModeAddInputEvent(0x28, DEMO_MODE_INPUT_MOVE, 0, 0xCE, 0x98); 152 LibAria_DemoModeAddInputEvent(0x001E, DEMO_MODE_INPUT_MOVE, 0, 0xD0, 0x92); 153 LibAria_DemoModeAddInputEvent(0xA, DEMO_MODE_INPUT_MOVE, 0, 0xD1, 0x08D); 154 LibAria_DemoModeAddInputEvent(0x14, DEMO_MODE_INPUT_MOVE, 0, 0xD2, 0x86); 155 LibAria_DemoModeAddInputEvent(0x14, DEMO_MODE_INPUT_MOVE, 0, 0xD4, 0x78); 156 LibAria_DemoModeAddInputEvent(0xA, DEMO_MODE_INPUT_MOVE, 0, 0xD6, 0x05E); 157 </pre>	