

STM32CubeMX系列教程

www.waveshare.net/study/portal.php

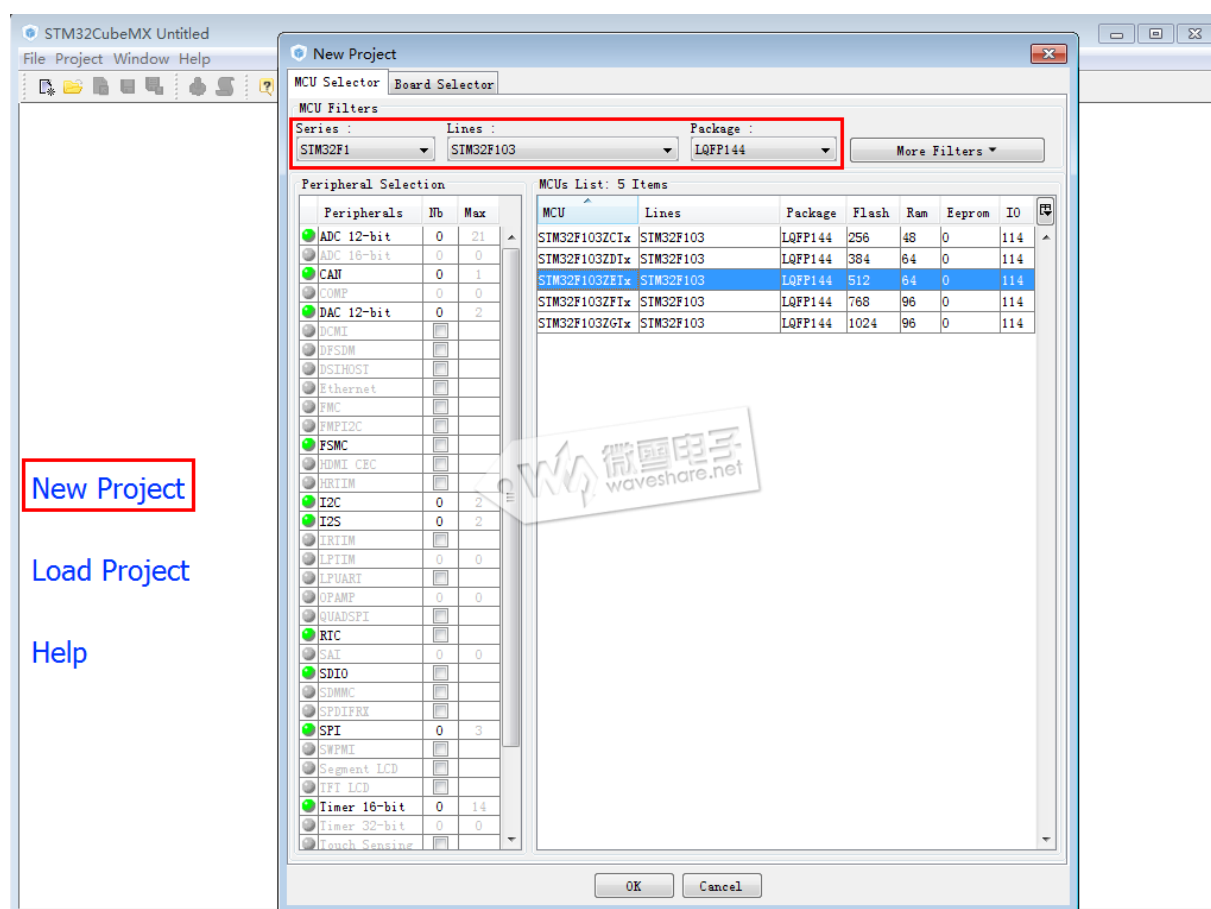
STM32Cube 是一个全面的软件平台，包括了ST产品的每个系列。平台包括了STM32Cube 硬件抽象层(一个STM32抽象层嵌入式软件，确保在STM32系列最大化的便携性)和一套的中间件组件(RTOS, USB, FatFs, TCP/IP, Graphics, 等等)。

- 直观的STM32微控制器的选择和时钟树配置
- 微控制器图形化配置外围设备和中间件的功能模式和初始化参数
- C代码生成项目覆盖STM32微控制器的初始化符合IAR™，Keil的™和GCC编译器。

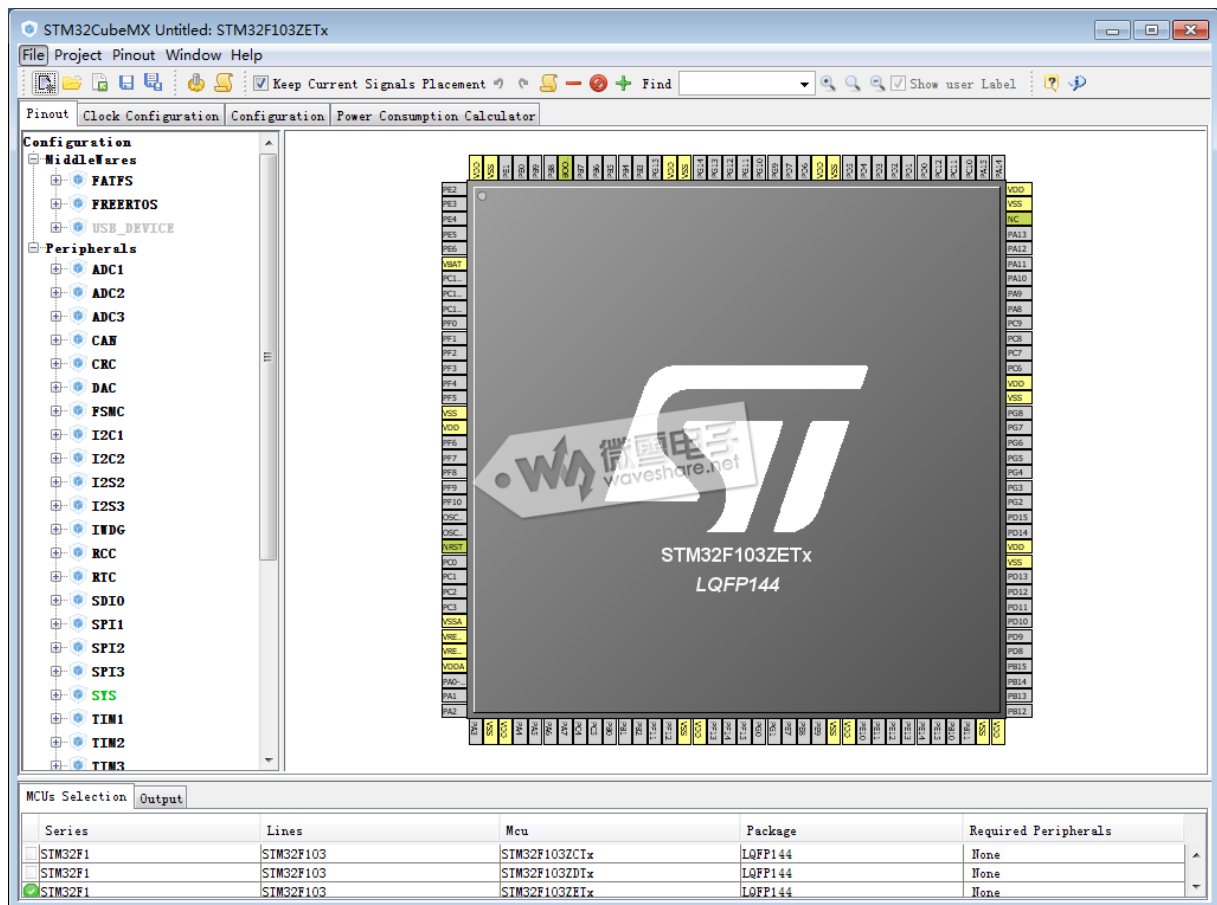
对于新的产品设计，我们强烈推荐使用STM32Cube来加速你的开发过程，并为以后的产品平台移植打下良好的基础。

1.新建工程

打开STM32cubeMX软件，点击New Project。选择对应开板MCU（STM32F103ZET6）。

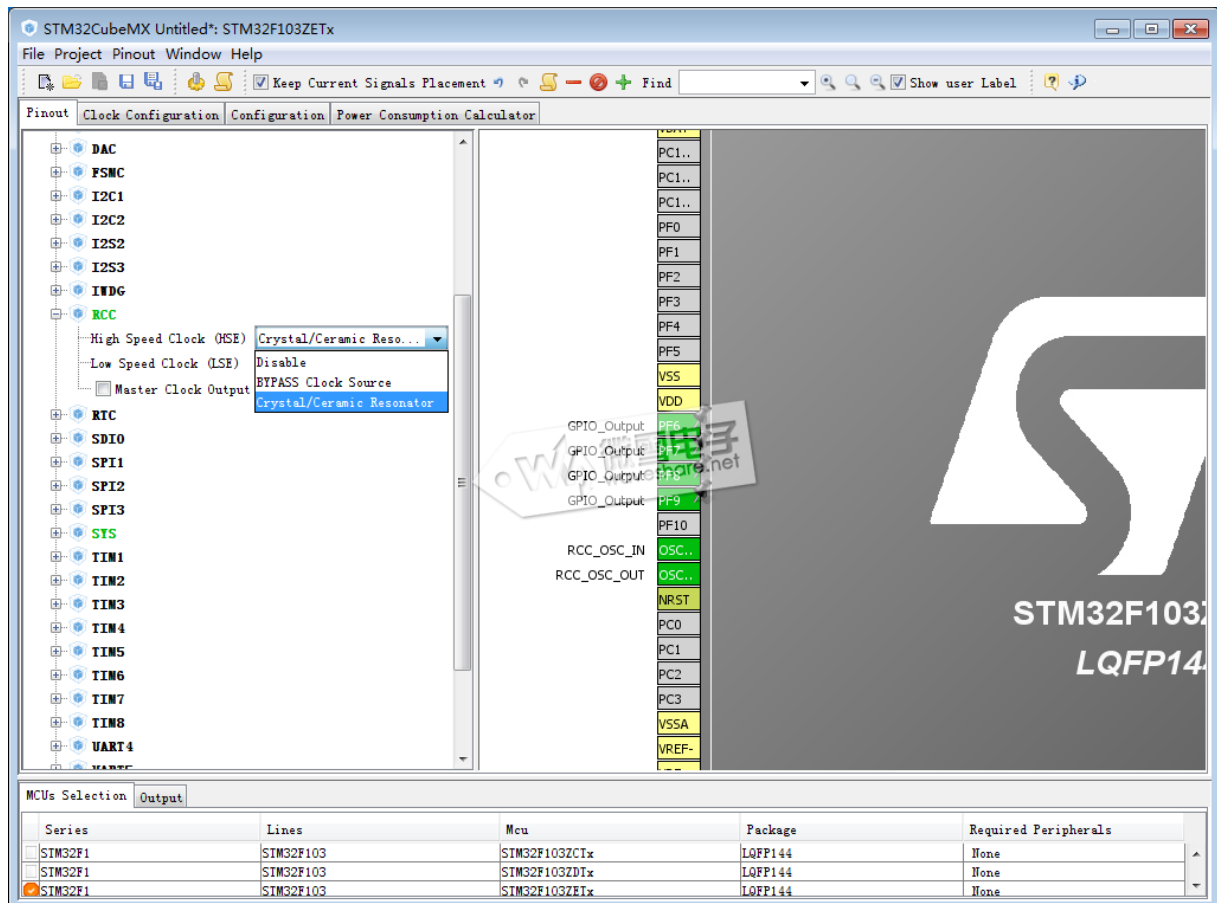


选择工程后进入工程界面，如下图所示。



2. 配置外设。

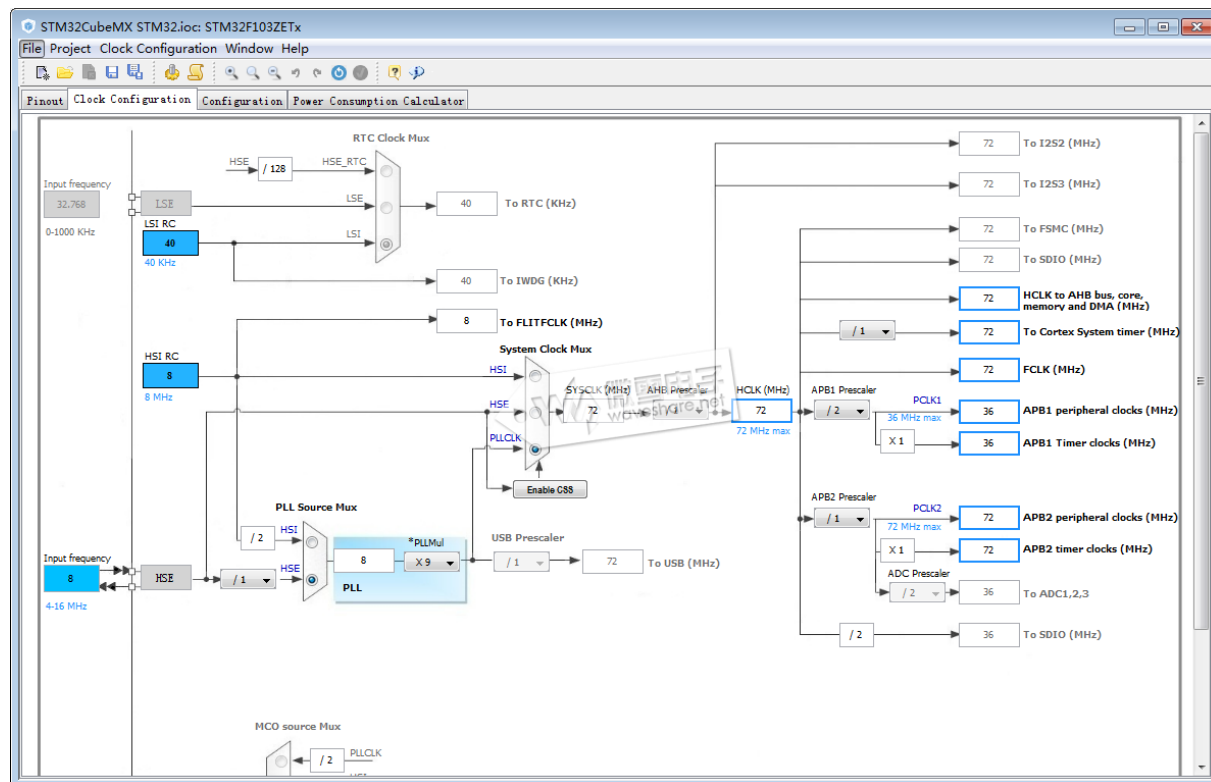
RCC设置，选择HSE(外部高速时钟)为Crystal/Ceramic Resonator(晶振/陶瓷谐振器)



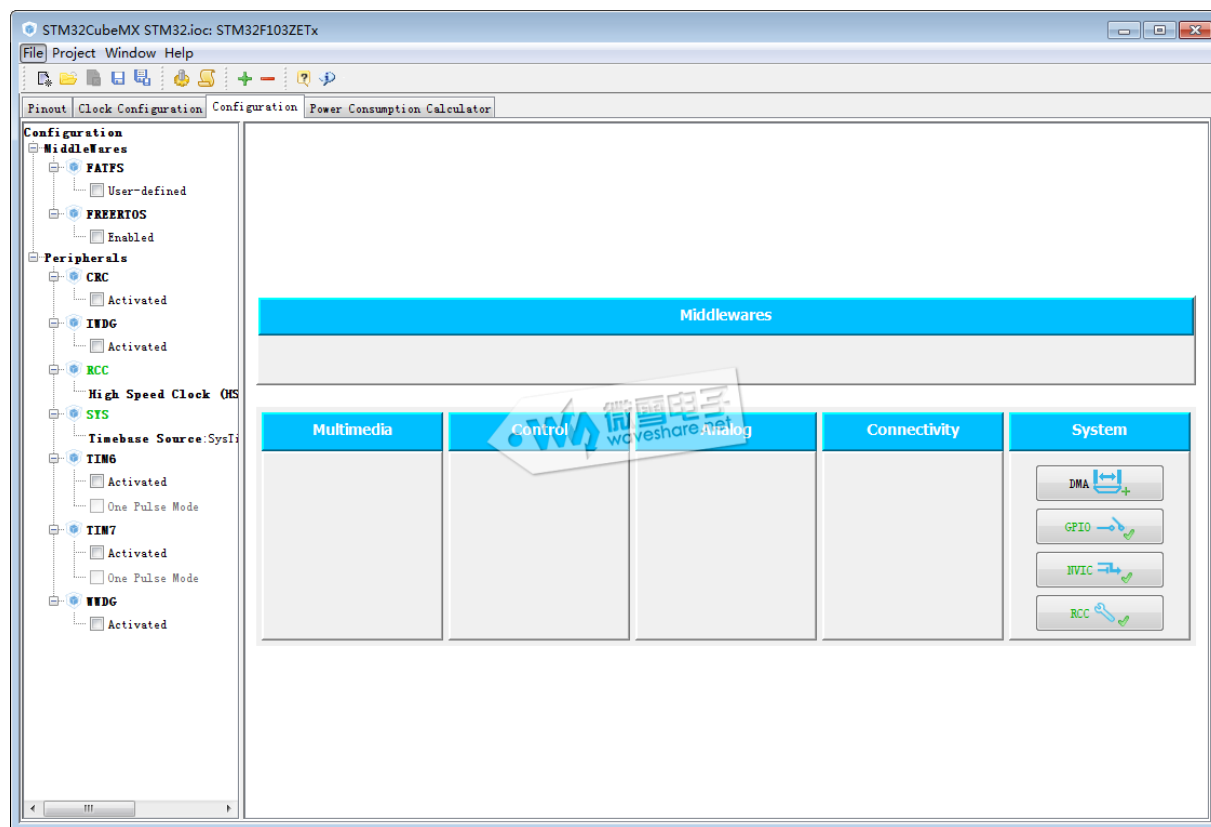
GPIO口功能选择，PF6,PF7,PF8,PF9为LED1-LED4.找到对应管脚设置为GPIO_Output模式。（黄色引脚为该功能的GPIO已被用作其他功能，可以忽略。绿色表示管脚已使用）

3. 时钟配置

时钟配置采用图形配置，直观简单。各个外设时钟一目了然。STM32最高时钟为72M，此处只有在HCLK处输入72，软件即可自动配置。（RCC选择外部高速时钟）。



4. 功能外设配置

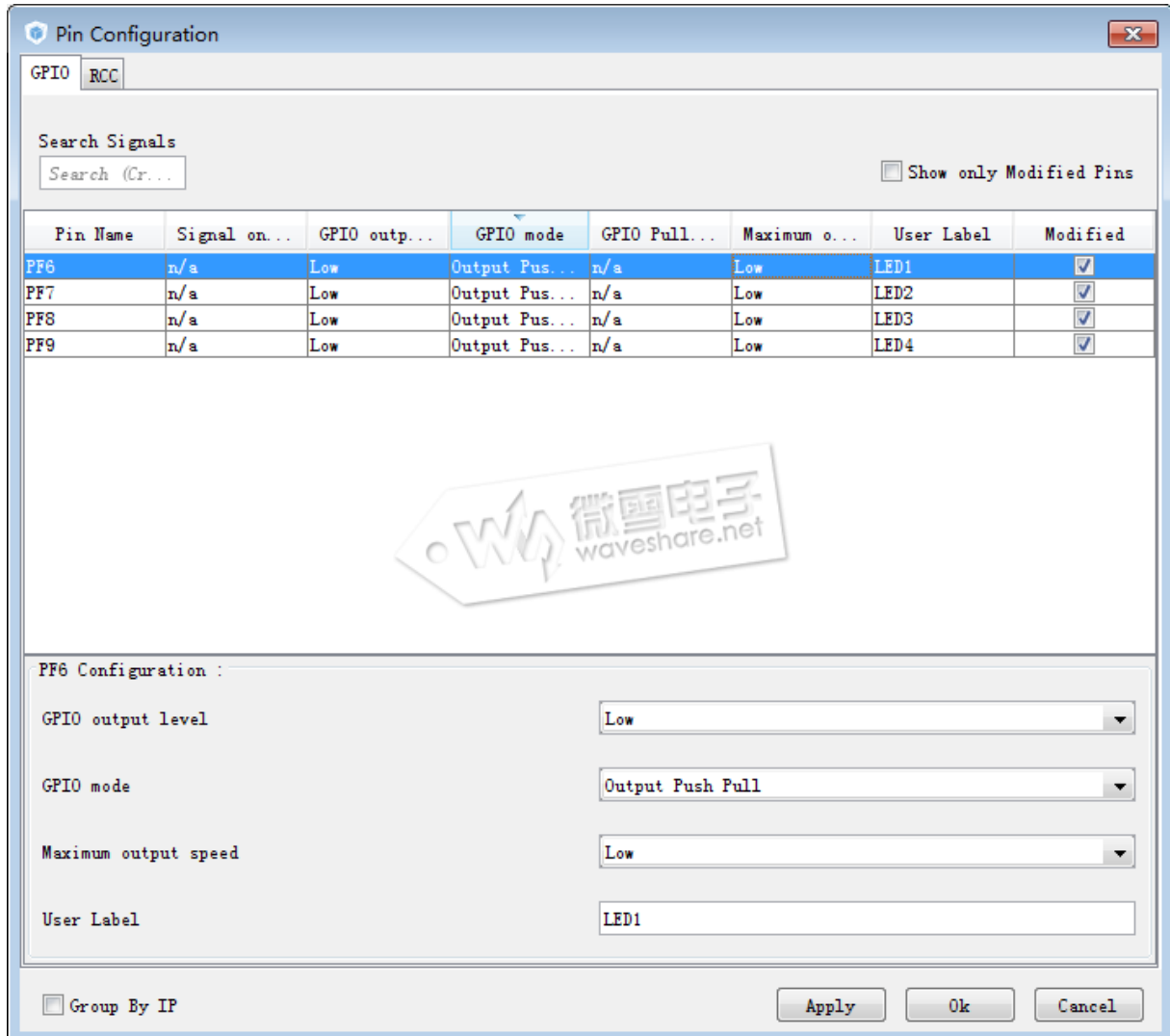


在配置框我们可以看到有几个区域，分别对应的功能设置如下

- Multimedia(多媒体)：音频视频、LCD
- Control(控制)：定时器
- Analog(模拟)：DAC、ADC
- Connectivity(通讯连接)：串口、SPI、I2C、USB、ETH
- SYStem(系统)：DMA(直接存储器存取)、GPIO、NVIC、RCC、看门狗

- middlewares(中间件): FreeRTOS、FATFS、LwIP、USB

此工程中DMA没用的不用配置，NVIC（嵌套中断向量控制器(Nested Vectored Interrupt Controller)）配置中断优先级。RCC不用配置。



Pin Configuration

GPIO **RCC**

Search Signals
Search (Cr...)

☐ Show only Modified Pins

Pin Name	Signal on...	GPIO outp...	GPIO mode	GPIO Pull...	Maximum o...	User Label	Modified
PF6	n/a	Low	Output Pus...	n/a	Low	LED1	<input checked="" type="checkbox"/>
PF7	n/a	Low	Output Pus...	n/a	Low	LED2	<input checked="" type="checkbox"/>
PF8	n/a	Low	Output Pus...	n/a	Low	LED3	<input checked="" type="checkbox"/>
PF9	n/a	Low	Output Pus...	n/a	Low	LED4	<input checked="" type="checkbox"/>

PF6 Configuration :

GPIO output level: Low

GPIO mode: Output Push Pull

Maximum output speed: Low

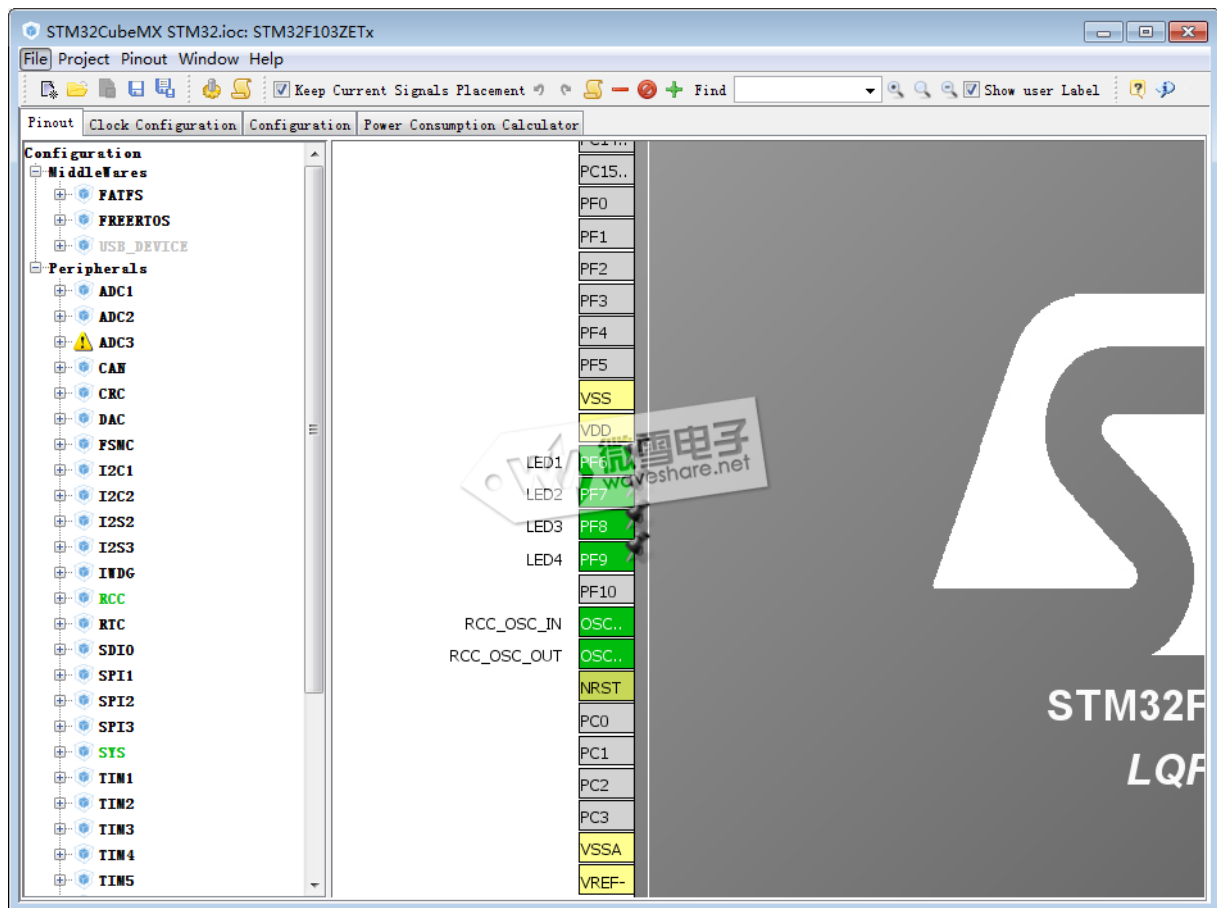
User Label: LED1

☐ Group By IP

Apply Ok Cancel

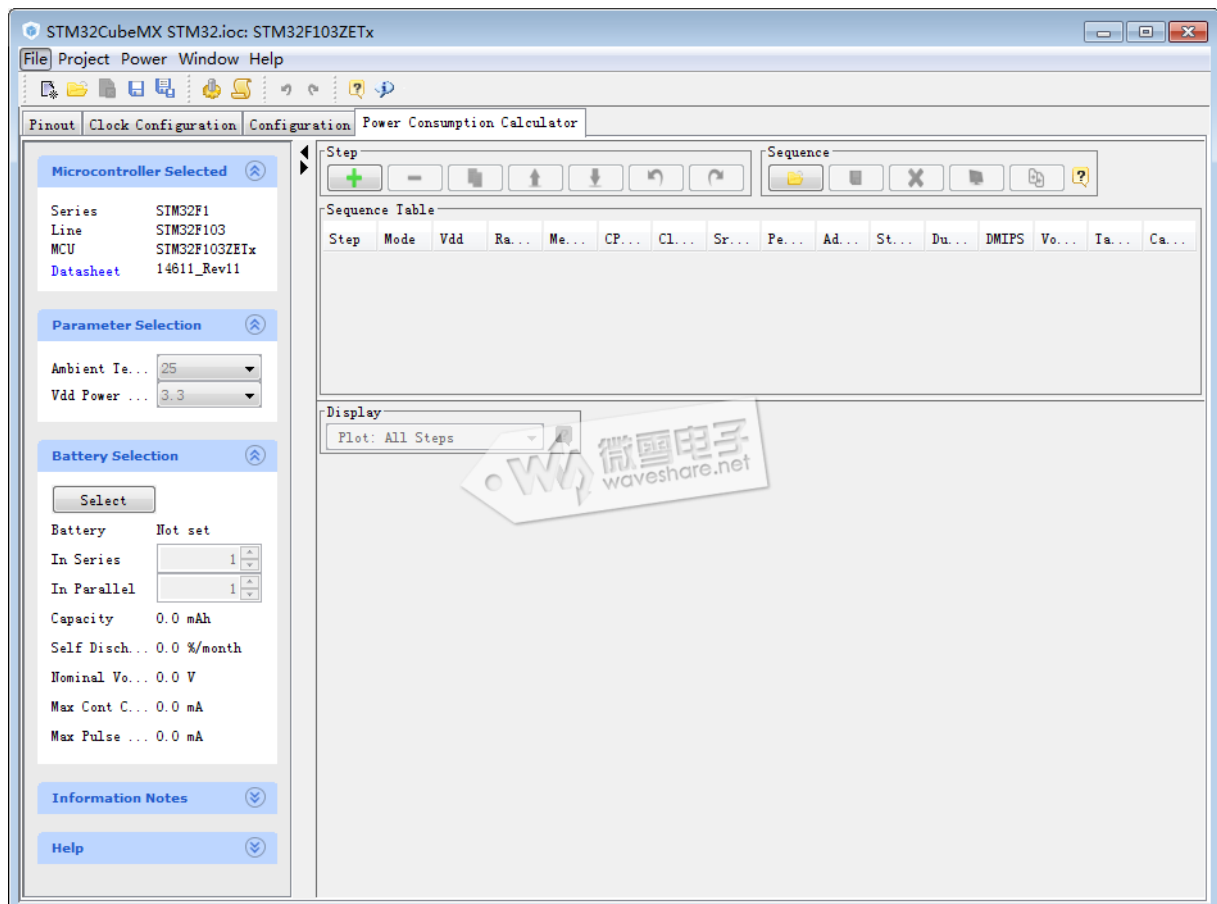
- GPIO Pin Level (管脚状态):低电平
- GPIO mode (管脚模式):推挽输出
- Maximum output speed (最大输出速度):低速
- User Label (用户标签):LED1

更改用户标签，管脚配置图会显示管脚的标签。



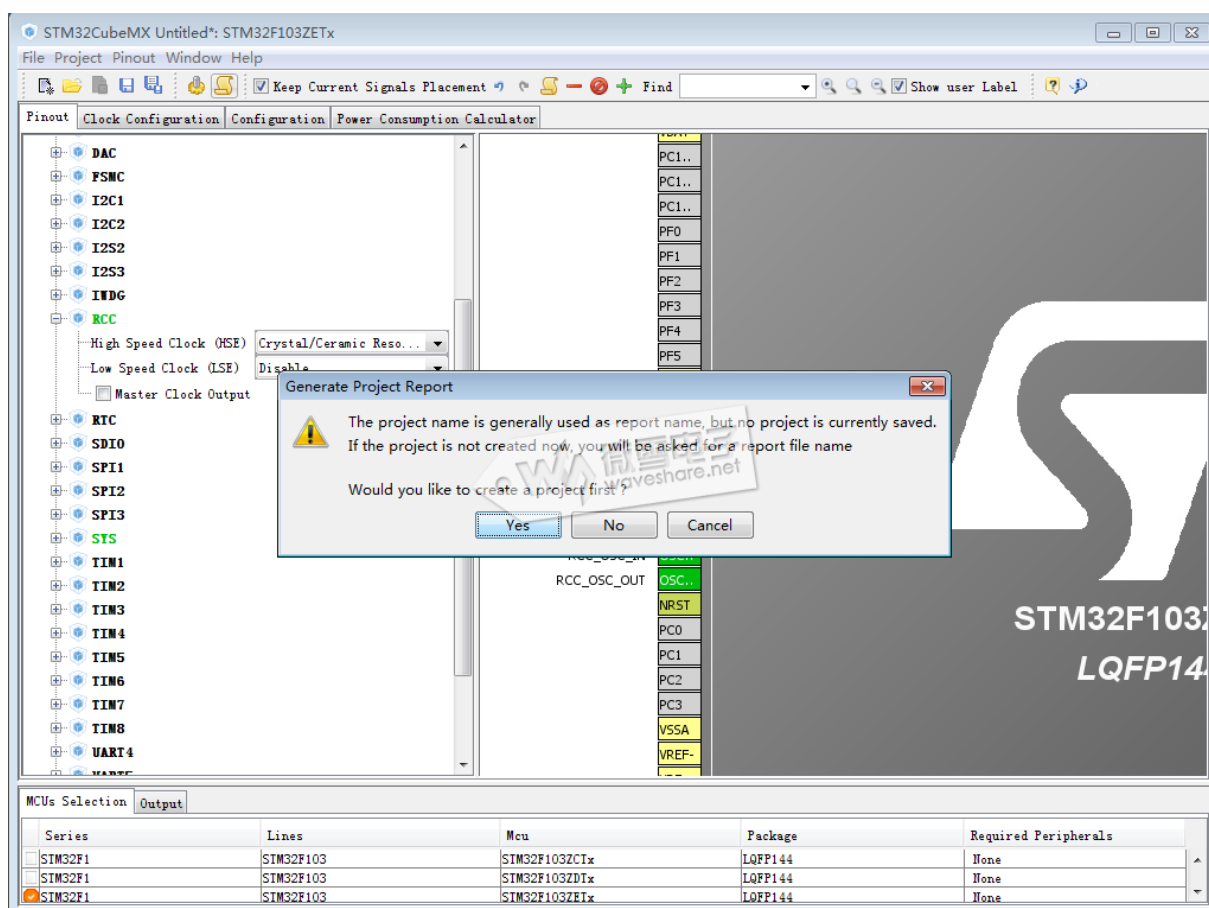
4. 功耗计算

这个根据配置的外设计算功耗，不用理会。

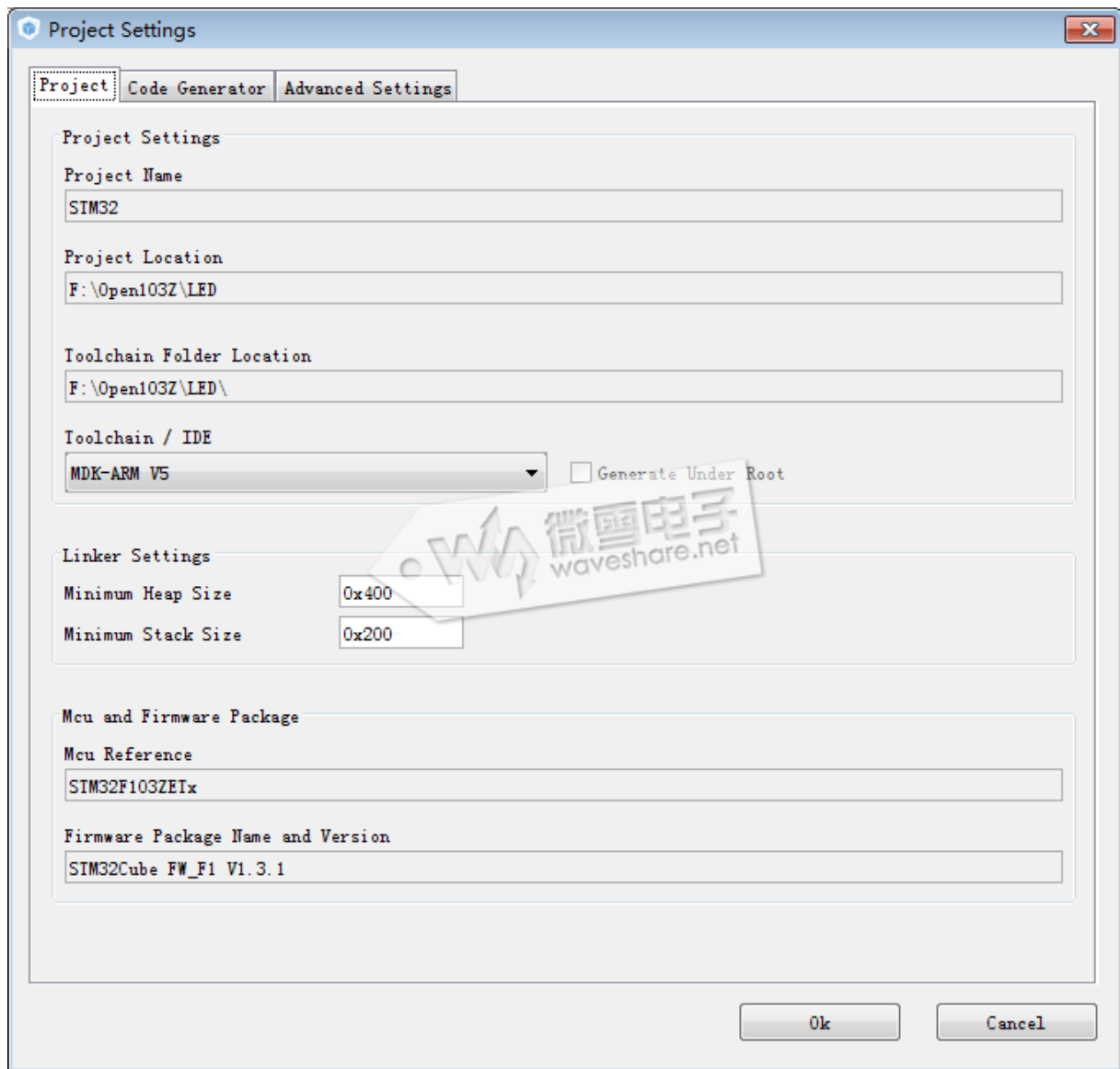


5. 生成工程报告

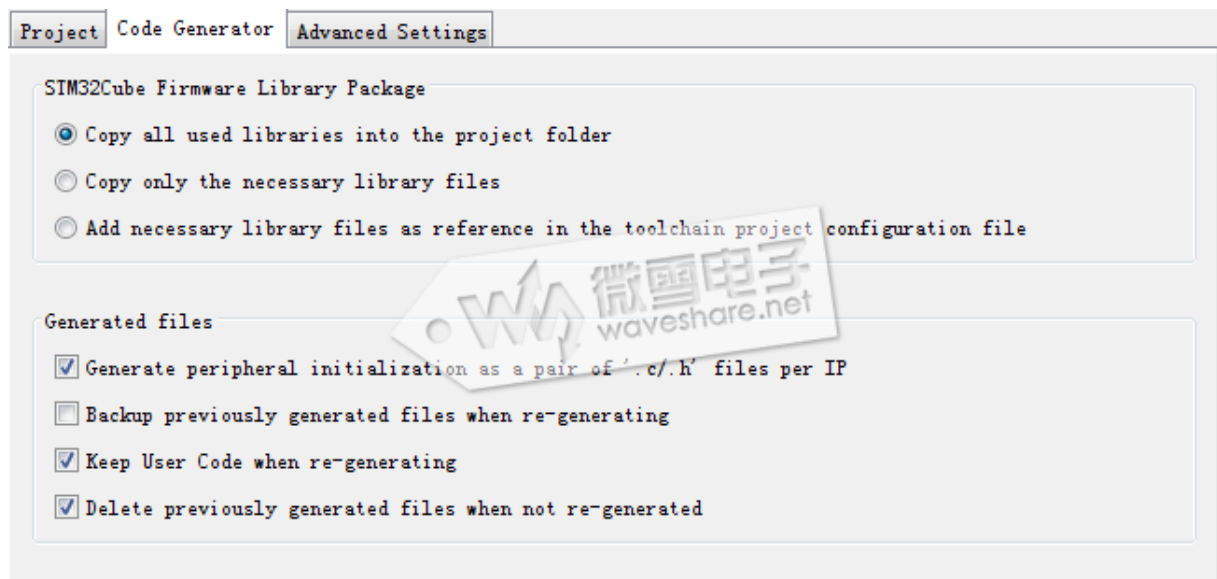
点击Project ->Generate Reports或者点击快捷图标生成报告。系统会提示先创建一个工程项目。点击Yes设置工程。



输入工程名，选择工程路径（注意不要出现中文，否则可能出错）。工具链/IDE选择MDK-ARM V5。最后面可以设置堆栈大小，此处默认不作修改。



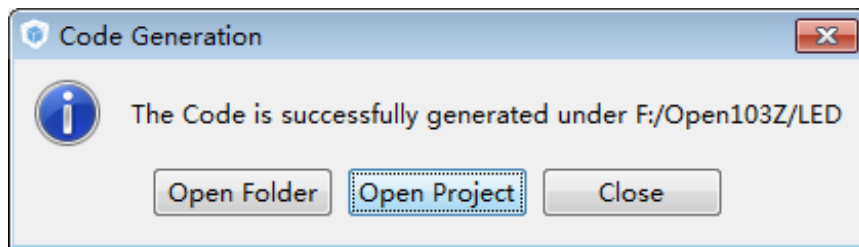
在Code Generator中找到Generated files框，勾选Generated periphena initialization as a pair of '.c/.h' files per IP。外设初始化为独立的C文件和头文件。



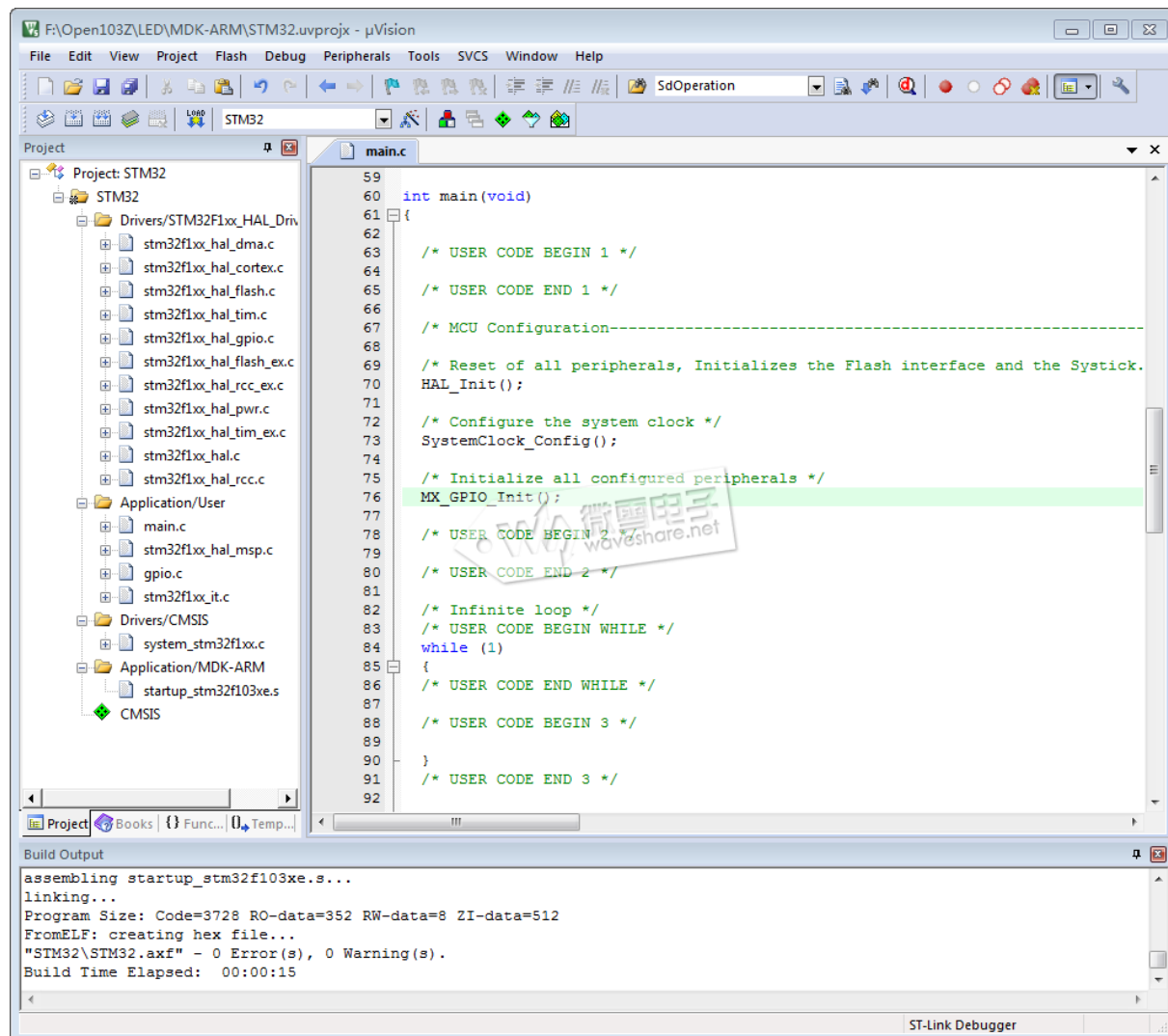
点击生成报告，工程目录下会生成txt文件和pdf文件，里面记录了我们刚才的设置。

6. 生成工程代码

点击Project ->Generate Code或者点击快捷图标生成工程代码。



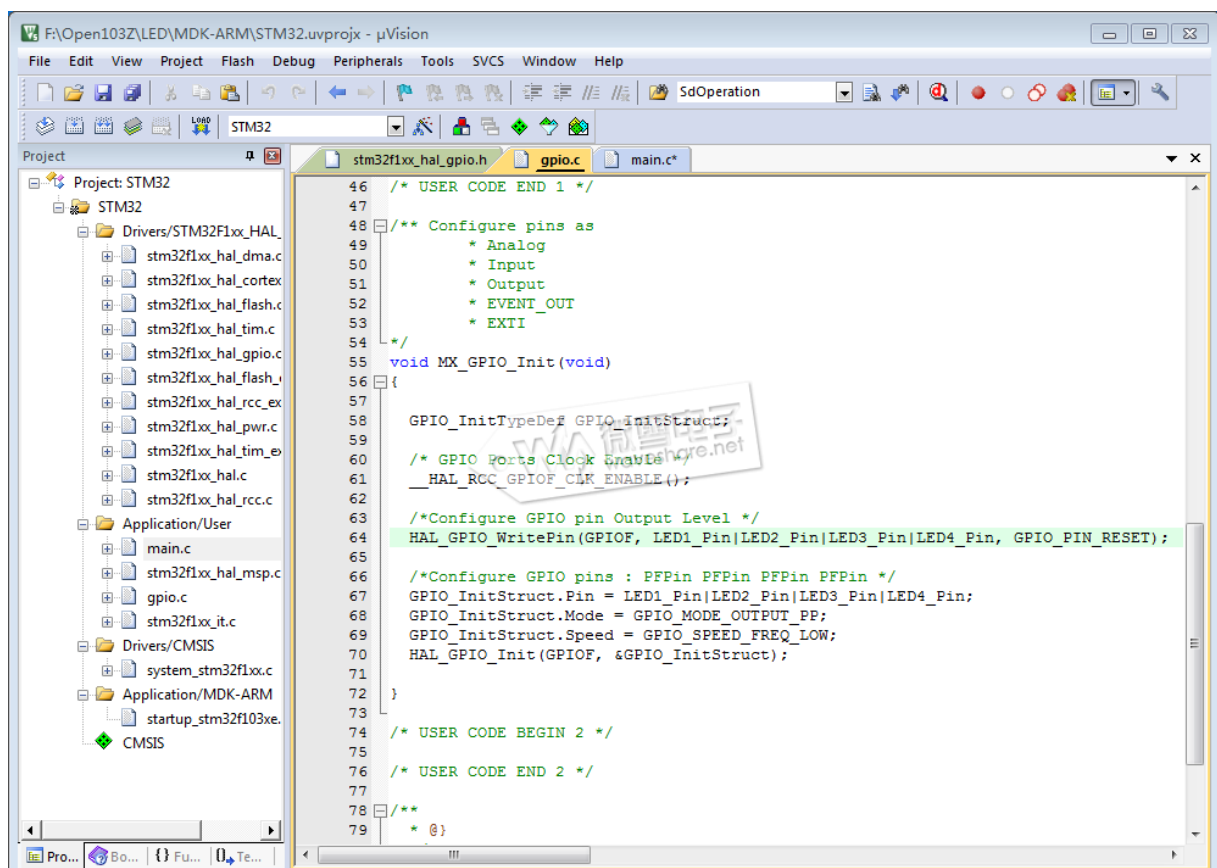
点击Open Project打开工程。到此就配置好工程外设初始化。



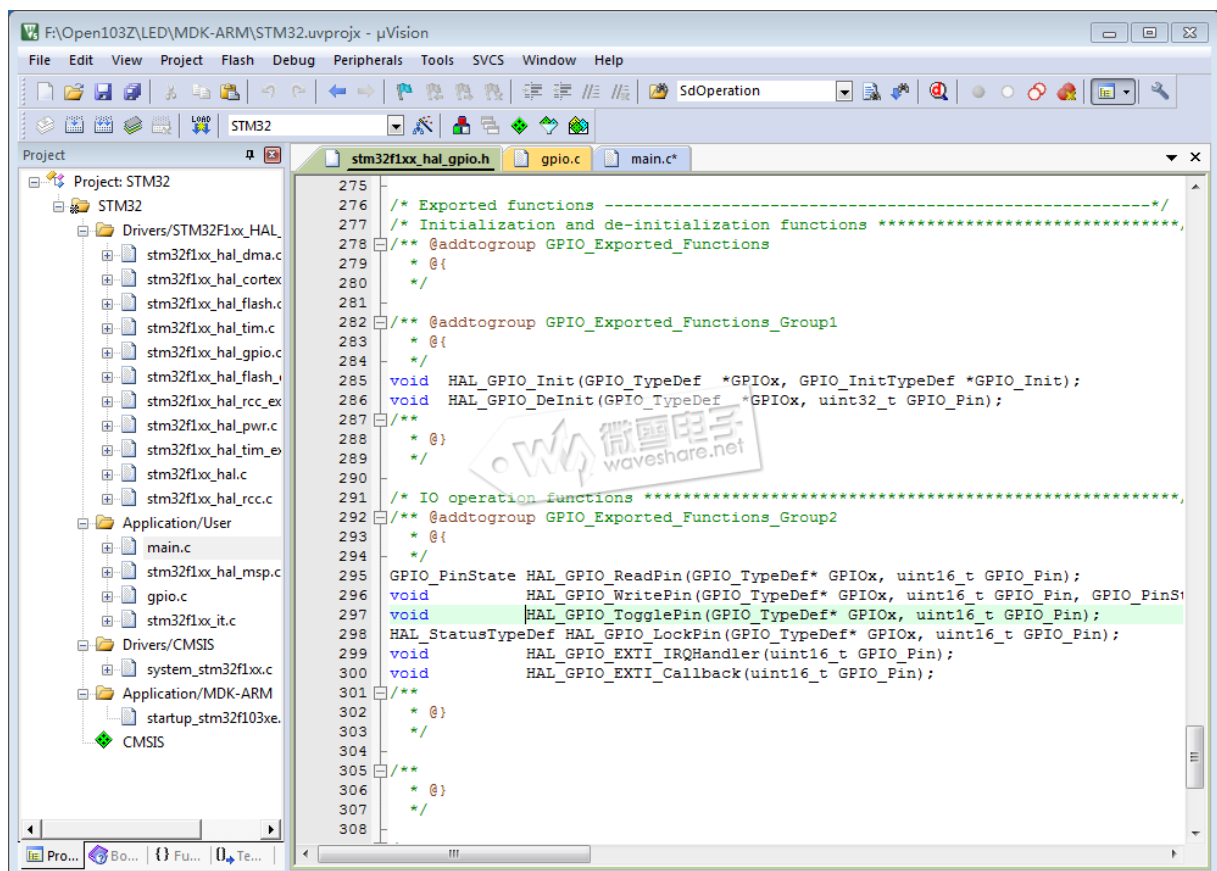
点击Build按钮，Build Optput信息框会输出没有错误没有警告。

6. 添加应用程序

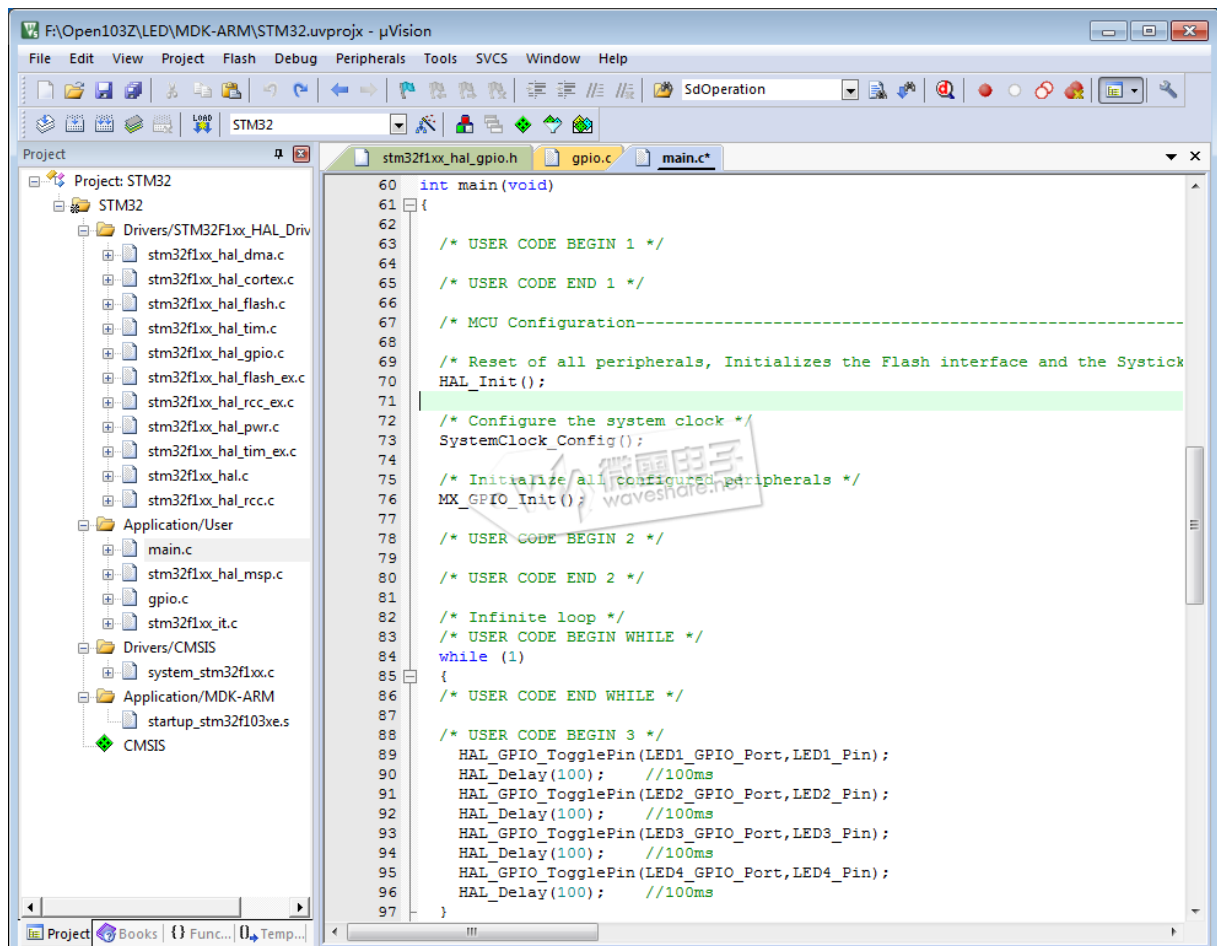
在gpio.c文件中可以看到LED管脚的初始化函数。



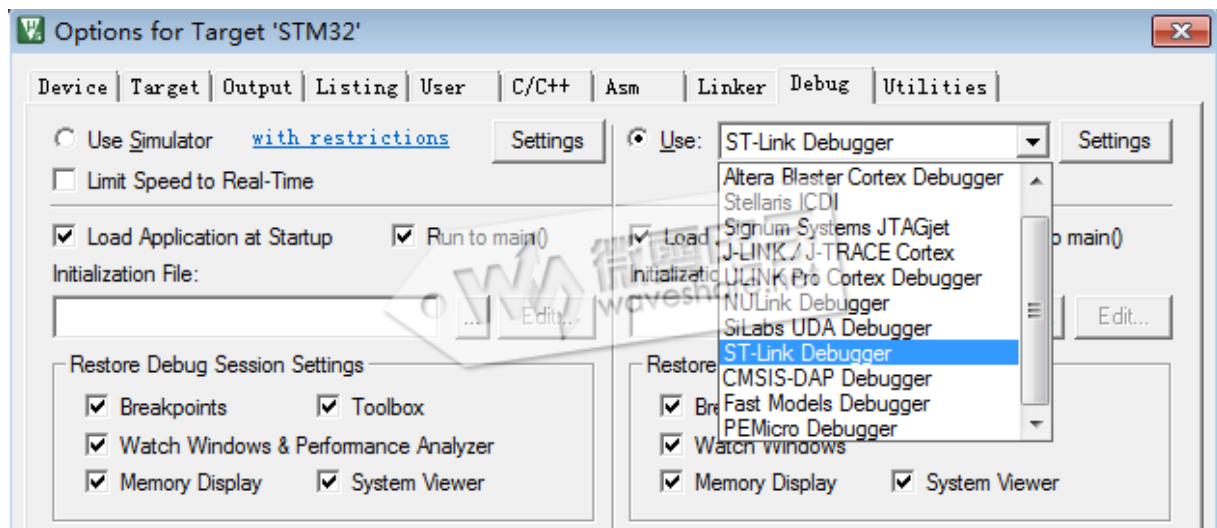
在stm32f1xx_hal_gpio.h头文件中可以看到GPIO的操作函数。



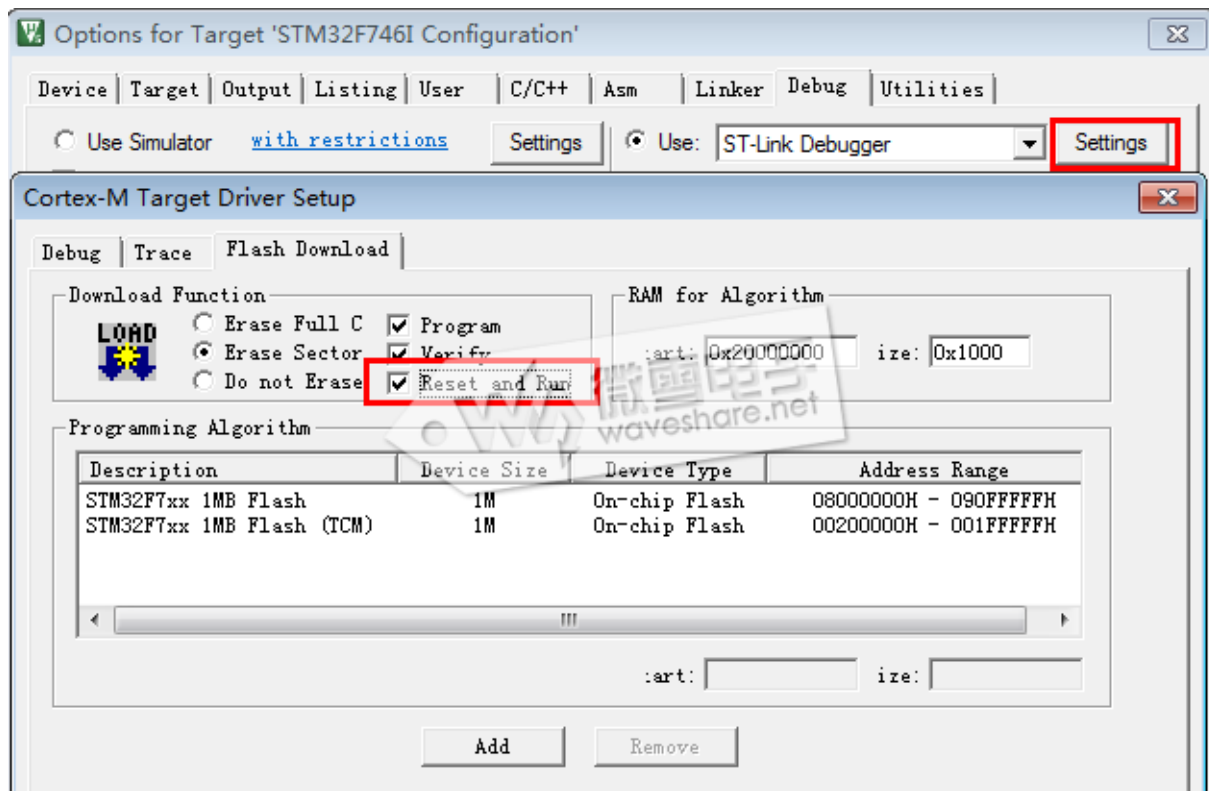
在main函数中的while循环中添加LED流水灯效果的应用程序。



重新编译程序，点击下载到Open103Z-C开发板。如果提示错误，可以点击图标对Option for Target 的 Debug选项进行修改。(图上选的是ST-LINK)



点击Settings->Flash Download勾选 Reset and Run选项。这样程序下载后自动启动运行，不用再按一下复位或者重新上电才能运行。



程序下载到Open103Z-C开发板。可以看到LED1~LED4依次被点亮，实现流水灯的效果。

总结：STM32Cube提供了固件库，用户可直接调用固件库函数来开发，并且可以很好的实现STM32-MCU全系列的代码一致性。同时STM32CubeMX工具提供的可视化引脚、外设、时钟等配置功能，可以帮助快速完成工程的建立、初始化。大大降低了开发者的工作量。

后续我们会结合微雪Open746I-C开发板，通过STM32cubeMX系列教程讲解如何学习STM32F7系列的片上外设。通过STM32F7系列的学习，可以让各位更快的掌握其他系列的使用方法。