

基于 Proteus8.9 和 STM32 的 HAL 库模拟仿真 STM32F103R6 芯片方法

一、几点重要问题说明

1. MCU 仿真型号要统一

Proteus 中 STM32F103 就几个型号，不全。而且 proteus 的 MCU 型号必须和利用 cubemx 生成的工程的 MCU 型号一致，否则不能用。例如 STM32F103R6 和 STM32F103RE 是两个 MCU，不要混了。

2. 无法 C 调试

利用 HAL 库开发，目前在 Proteus 中只有汇编语言，没有办法编译为 C 进行单步调试。

3. 选 elf 格式文件

Proteus 中 MCU 可以加载 hex 和 elf 两个格式文件，建议用 elf 文件，hex 不稳定。

4. MCU 的主频和电源地等可以默认

网上很多例子说必须设置这些参数，目前我利用向导建立的工程不需要。

5. 报错后一定重启 proteus

调试串口的时候总报错死程序，一定要重启 proteus，否则一堆问题。

二、安装 Proteus8.9

1. 下载安装软件

下载地址，链接：<https://pan.baidu.com/s/1D9lk8FTMwH15nTiWKumqeQ>

提取码：vcm4

下载后安装时英文版，里面有汉化包，复制所有汉化文件到安装目录的语言文件夹即可。



图 1 汉化文件夹

2. 建立 STM32F103R6 工程

按下图操作，主要最后一个界面选择对应型号的 MCU

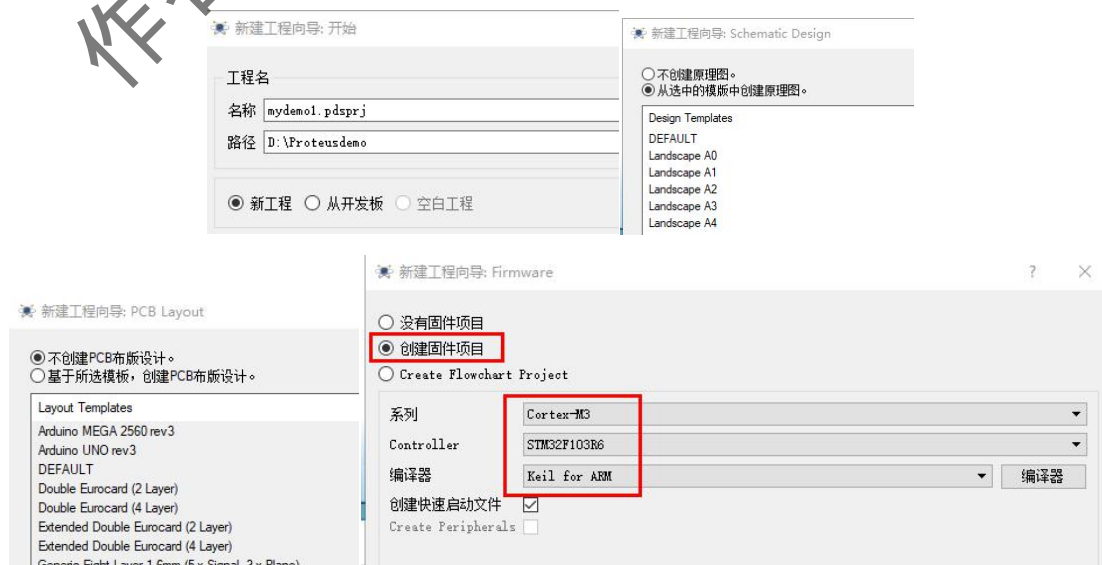


图 2 工程步骤

3. 添加器件

这个软件是英国人开发的，使用时左键选取、移动。右键菜单、中间键移动屏幕，滚轮放大缩小，都有功能，用着不爽。

如下图所示 1 为选取，2 为电子全器件选取、3 为 ground 和 VDD、4 是示波器等、5 和 6 是模拟仿真的开始与结束。

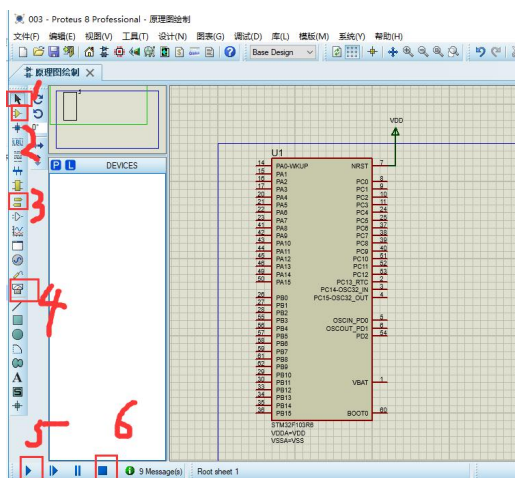


图 3 项目窗体

点击 2，在右侧框中点击 P 字母，弹出选取器件窗口。利用关键词找到器件，就添加到器件串口中，后面就可以直接用了，不用每次都搜索。

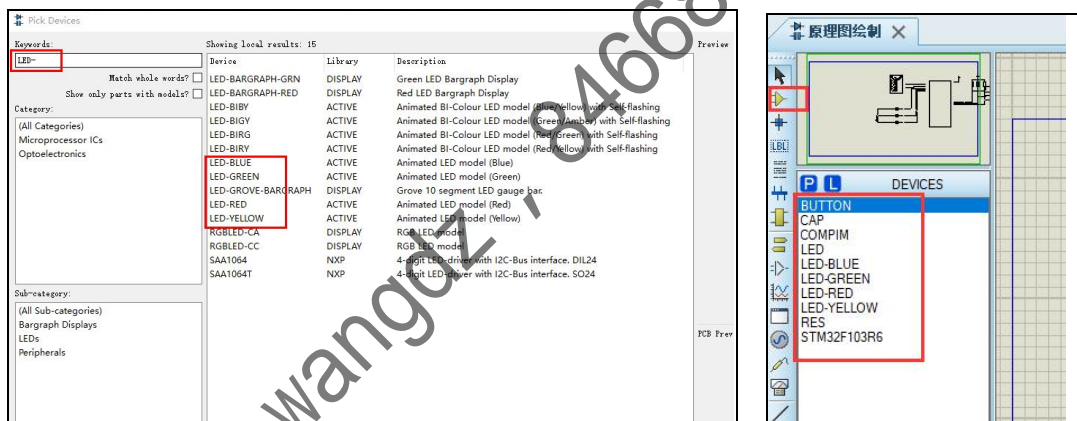


图 4 添加候选器件

4. 添加常用器件

如下图所示，建立三个输入按钮、三个输出 LED 小灯，一个串口。

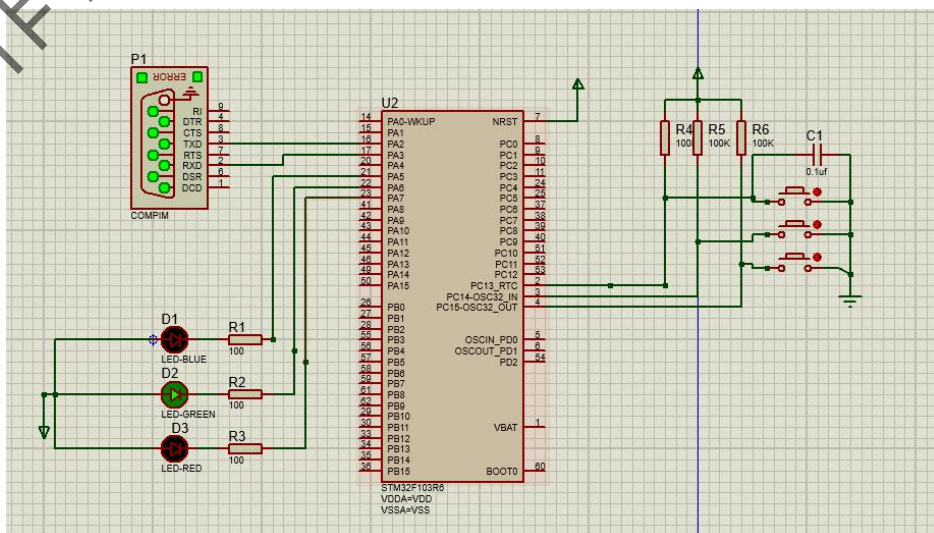


图 5 建立测试项目

三、几点注意事项

1. LED 灯

要连接一个 100 欧姆的电阻，没有会短路。电阻太大了，电流不够，不亮。100K 就不亮。

2. 开关

用的是上拉电阻模式，软件的中断的采样一定用下降沿触发中断。如果用上升沿，按一次会有 2 次中断，不知道为什么，一个坑。

3. 串口

也是坑最大的，网上有人用 proteus 里面的虚拟终端调试，我试了一下，不稳定，有乱码。用 COMPM 器件好一点，但是也是不稳定。波特率不要太高了，用 9600。另外如果导致工程死机，就重新启动 proteus，否则后面怎么模拟都有问题一堆。

四、配置串口参数

这个单独说一下，需要一些第三方的软件。

1. 安装虚拟串口

下载地址: <https://dl.pconline.com.cn/download/825163.html>

2. 串口助手

自己网上找吧。

3. 利用虚拟串口虚拟串口

虚拟两个 COM8 和 COM9，他们内部已经互相联通了。

4. 配置串口器件参数

在 proteus 中双击串口器件，添加如下参数。

另外注意，COMPM 和 MCU 是直连关系。

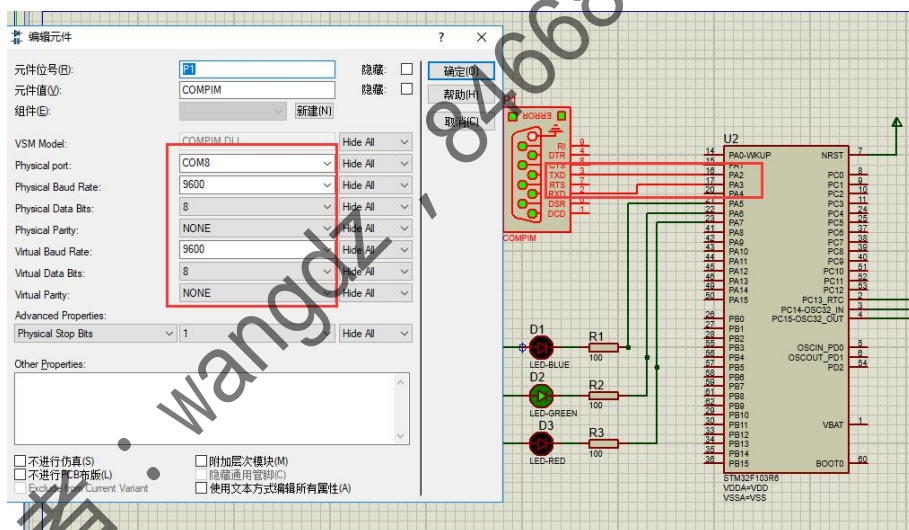


图 6 串口参数

5. 程序运行

仿真开始后，利用串口助手打开 COM9，因为 com9 和 com8 是互通的，这样 com9 就可以跟 mcu 的串口通信了。

五、MCU 加载程序

双击 MCU，在 program file 中选择你的 keil 项目中的 elf 文件，晶振自己设置吧。如果用了定时器，最好设置一下。

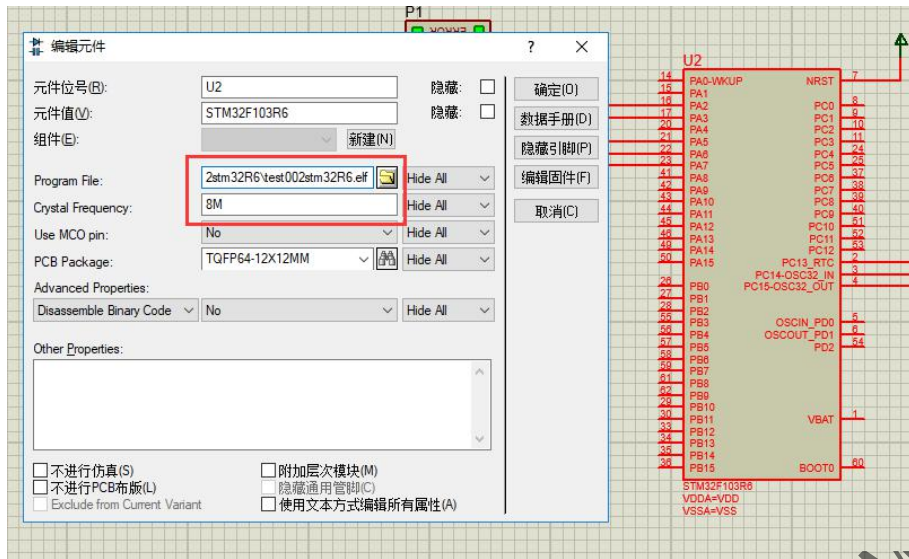


图 7 加载参数

六、运行仿真

点击窗口左下的仿真运行、停止按钮就可以控制仿真了。仿真过程中有问题，会在 message 窗口中有红色文字进行提示，自己理解吧。有问题，一定重启一下 proteus。



图 8 仿真

几点说明：

- (1) 现在仿真对 cubemx 生成的 HAL 库程序支持不好，都是汇编语句，没有办法单步调试，只能看硬件效果了。
- (2) 如果修改器件，添加、删除器件等，一定要停止仿真，才能操作。
- (3) 如果添加了示波器、逻辑分析仪、虚拟终端等，一定要谨慎，容易导致项目死掉。
- (4) 启动仿真报下面这个错，就是你 MCU 的程序文件加载错了，在加载一次。



图 9 MCU 文件加载错误

七、Keil 工程的配置

默认 keil 工程生成的是 hex 文件，如果需要 elf 文件，需要修改默认选项。

1. 调出项目 Option 窗口



图 10 调出 options 窗口

2. 修改 output 子窗口

去掉 hex 选项，在输出文件添加 elf 扩展名。Proteus 的 MCU 芯片就找这个文件加载就可以了。



图 11 参数修改

八、案例

做了一个基于 cubemx5.6、keil MDK5.0 和 proteus8.9 的案例。涵盖 GPIO 输入、输出、外部中断、定时器中断、串口输出、串口中断输入的基本功能案例。需要的话下载地址

链接: <https://pan.baidu.com/s/1HcjC2x7L5MXkPjA3pP2Agw>

提取码: b1rv

九、Proteus 与 keil 仿真的比较

前面写过一个 keil 下 stm32F103 仿真的方法，跟 proteus 比较。Keil 主要 MCU 内部功能的仿真，没有过多的外部接口支持、例如 AD /DA/SPI 等，只能看管脚和寄存器的 0/1 状态，不直观，分析有一定难度。但是优点是使用简单、系统稳定。

而 proteus 可以设计符合实际的模拟各种电路，比较直观。支持的接口也多。看最后的效果很容易分析问题。但是问题是，proteus 的使用需要数字和模拟电路基础，否则不能配置出合理的工程。另外一点也比较头痛，就是稳定性问题，过多的监控和通信器件很容易死机，是不是和谐版的问题，不好说。

总体来说，如果有电路基础，倾向于用 proteus，还是更加接近真实。Keil 仿真可以验证内部的逻辑，尽量少跟外部端口打交道。