



苏州大学

SOOCHOW UNIVERSITY



# 第7讲 程序的基本调试方法

**7.1** 常用调试方法简介

**7.2** 使用AHL-GEC-IDE 打桩调试

**7.3** 使用AHL-GEC-IDE进行Printf调试



苏州大学

SOOCHOW UNIVERSITY



# 7.1 常用调试方法简介





## 7.1

## 常用调试方法简介

### 单步调试

在调试位置设置断点，到达断点位置之后，逐句单步跟踪调试了解与硬件打交道的寄存器值的变化。

单步跟踪调试也用于底层驱动构件设计阶段。不进入子函数内部执行的单步跟踪调试，可用于整体功能跟踪。

注：AHL-GEC-IDE主要用于User程序开发，基本不涉及构件开发过程，较少需要使用单步调试，目前尚未集成单步调试功能。



## 7.1

## 常用调试方法简介

### 打桩调试

在程序中需要停住的位置加上for(;;);无限循环，若运行效果和预期一致说明无限循环之前的语句正确，否则需要在此之前查找问题。

打桩调试常常用于断点单步调试不适用的场景，辅助定位问题语句，如系统长时间运行后出现问题，单步跟踪调试花费大量精力而不方便实现的情况。



## 7.1

## 常用调试方法简介

### Printf调试

在程序中使用printf构件，通过串口格式化输出程序编写者想要知道的运行信息，如变量值、寄存器值、任务或函数执行信息等。

例如，可以在一些关键的代码或子函数调用语句的前后可以加入两条printf语句，标志关键语句的执行情况。





苏州大学

SOOCHOW UNIVERSITY



**7.2**

# 使用AHL-GEC-IDE 打桩调试





## 7.2

### 使用AHL-GEC-IDE打桩调试-基本调试步骤

在程序需要停住的地方，增加for(;;);  
无限循环

导入待调试工程

添加打桩语句

编译新程序

观察开发板的运行状况

烧录程序

若结果符合预期则  
无限循环前的语句  
正确，否则问题语  
句在此之前

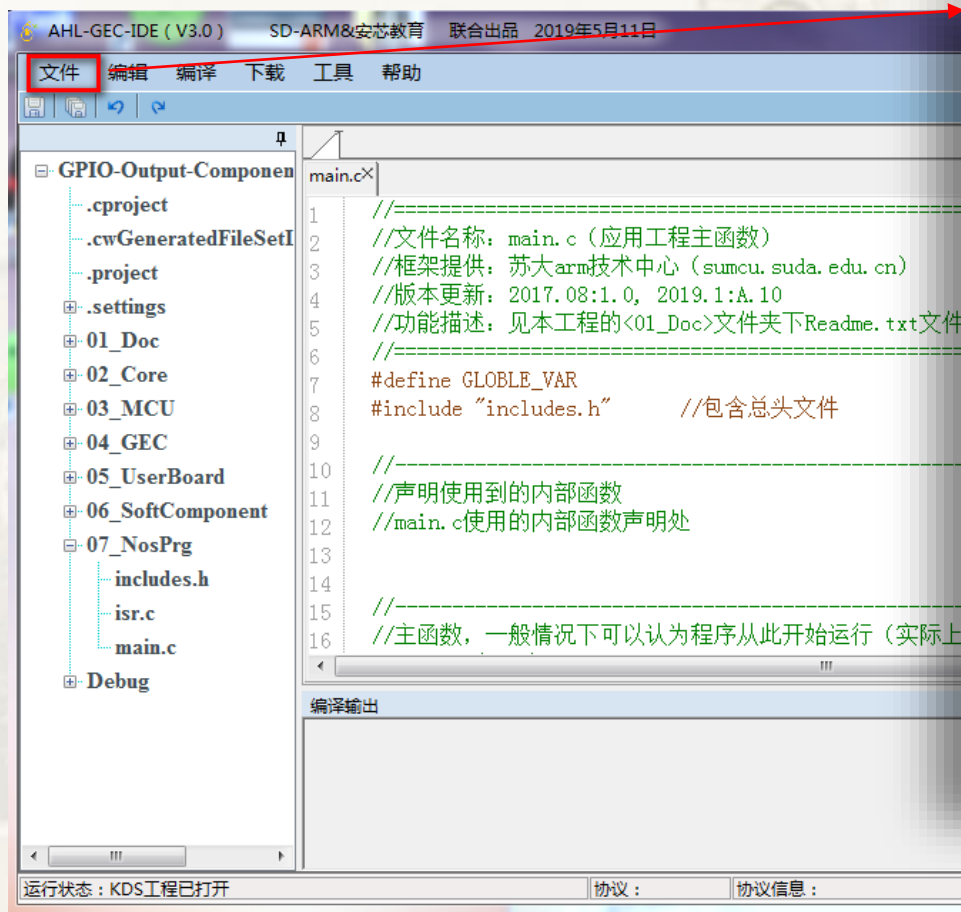


## 7.2

## 使用AHL-GEC-IDE打桩调试

## 01

## 导入工程



样例导入：

AHL-A00Kit01(V1.0)-190507\04-Soft\CH04-GPIO文件夹下的GPIO-Output-Component(KL36)

导入方法：

AHL-GEC-IDE 工具菜单栏“文件” -> “导入工程”，选择样例工程所在文件夹。



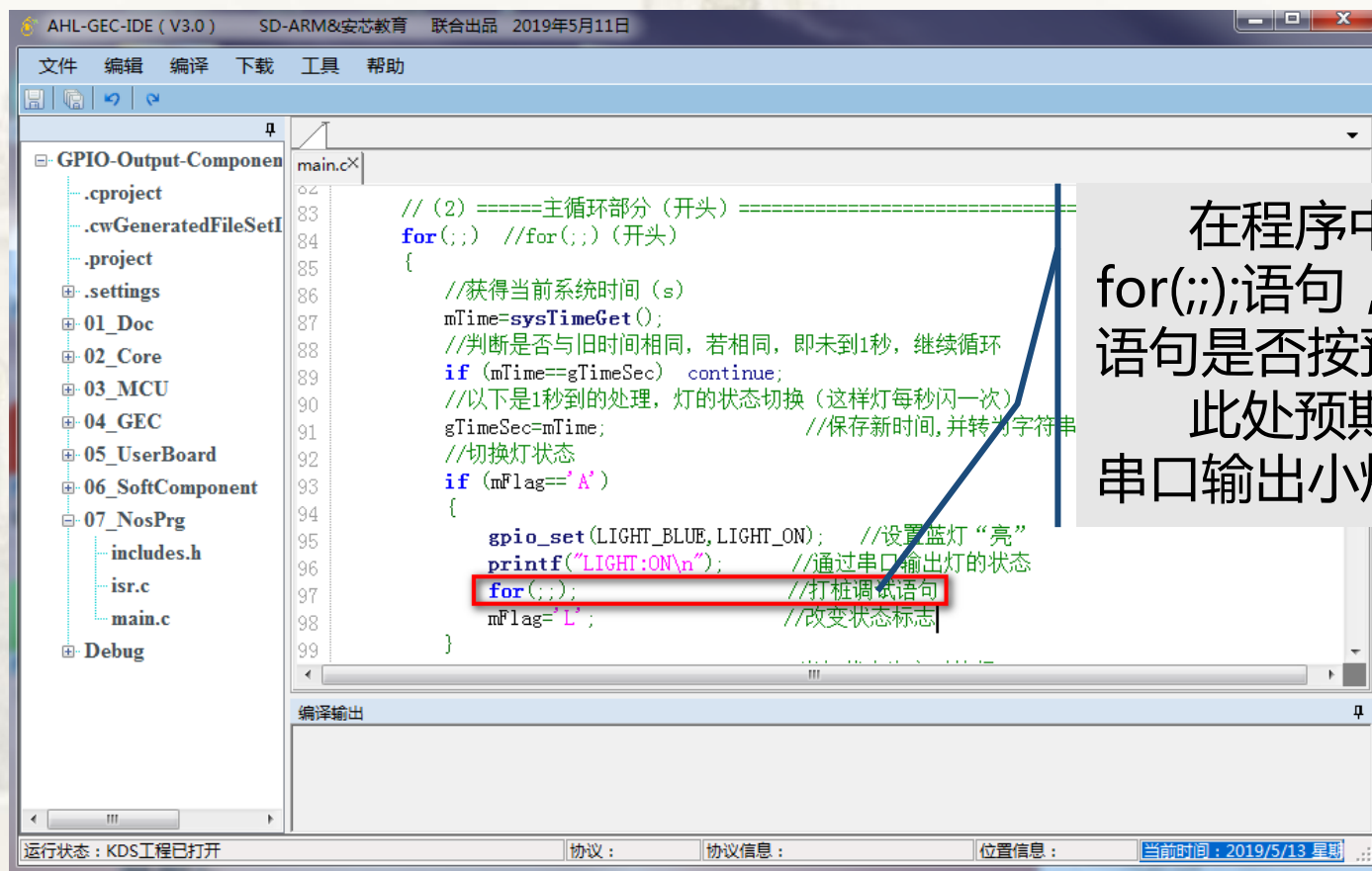


## 7.2

## 使用AHL-GEC-IDE打桩调试

## 02

## 添加for(;;);无限循环语句



在程序中适当位置添加  
for(;;);语句,之后查看其之前  
语句是否按预期执行。

此处预期结果:红灯亮,  
串口输出小灯状态

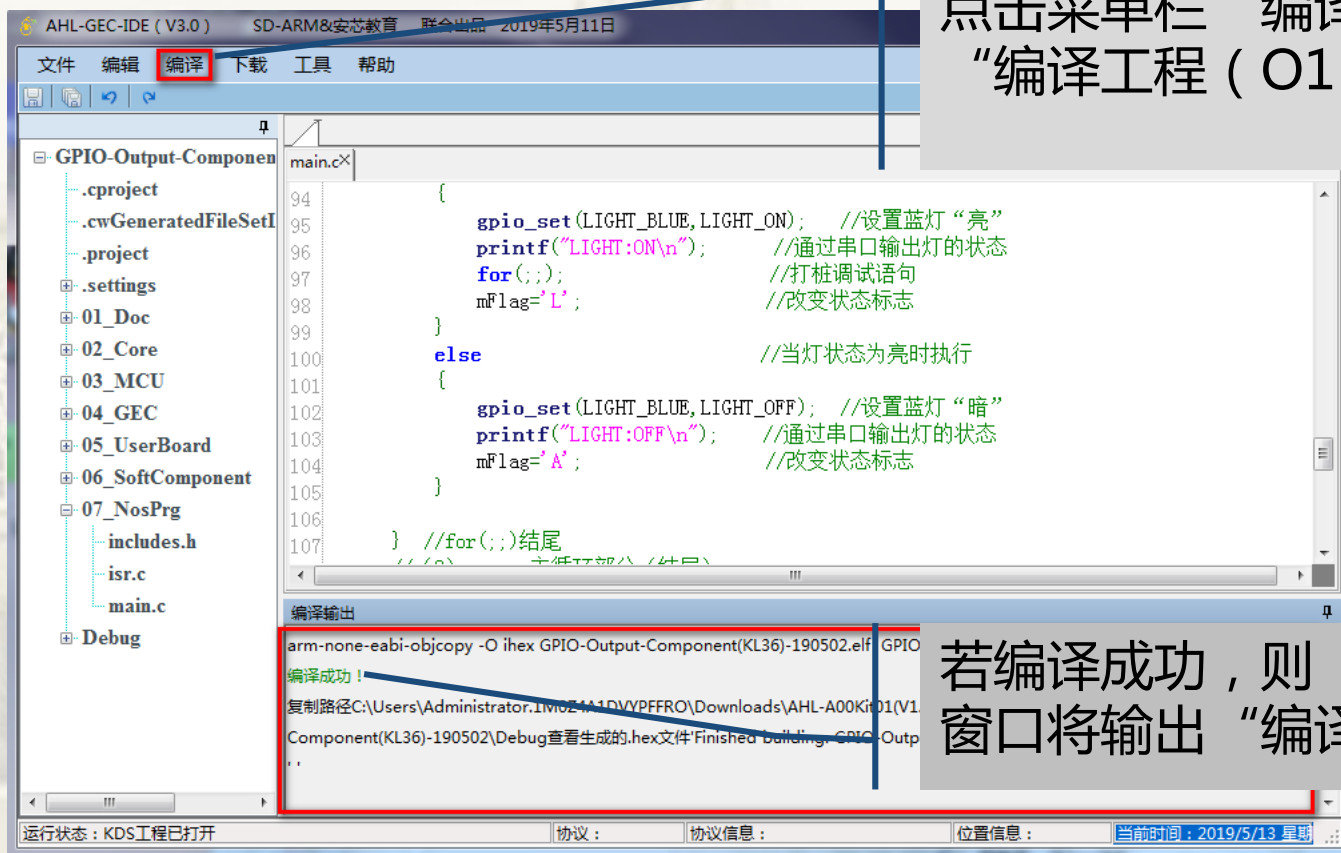


## 7.2

## 使用AHL-GEC-IDE打桩调试

## 03

## 编译新程序



点击菜单栏“编译” ->  
“编译工程 (O1)”

若编译成功，则“编译输出”  
窗口将输出“编译成功”字样

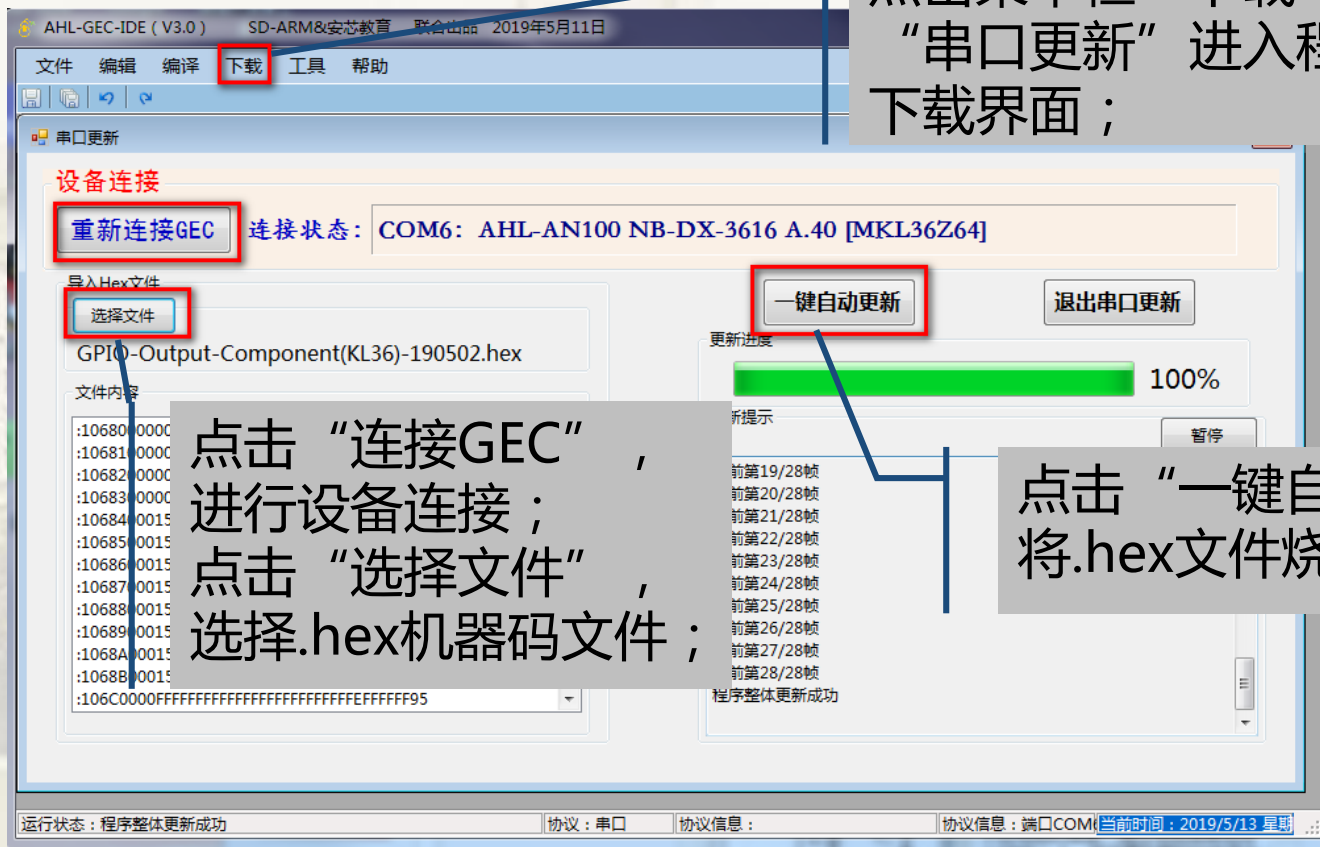


## 7.2

## 使用AHL-GEC-IDE打桩调试

### 04

### 串口烧录程序



点击菜单栏“下载” -> “串口更新” 进入程序下载界面；

设备连接

重新连接GEC

连接状态: COM6: AHL-AN100 NB-DX-3616 A.40 [MKL36Z64]

导入Hex文件

选择文件

GPIO-Output-Component(KL36)-190502.hex

文件内容

:10680:0000  
:10681:0000  
:10682:0000  
:10683:0000  
:10684:0015  
:10685:0015  
:10686:0015  
:10687:0015  
:10688:0015  
:10689:0015  
:1068A:0015  
:1068B:0015  
:106C0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

点击“连接GEC”，  
进行设备连接；  
点击“选择文件”，  
选择.hex机器码文件；

一键自动更新

退出串口更新

更新进度

100%

新提示

暂停

前第19/28帧  
前第20/28帧  
前第21/28帧  
前第22/28帧  
前第23/28帧  
前第24/28帧  
前第25/28帧  
前第26/28帧  
前第27/28帧  
前第28/28帧  
程序整体更新成功

点击“一键自动更新”，  
将.hex文件烧录到芯片

运行状态: 程序整体更新成功

协议: 串口

协议信息:

协议信息: 端口COM6 当前时间: 2019/5/13 星期三



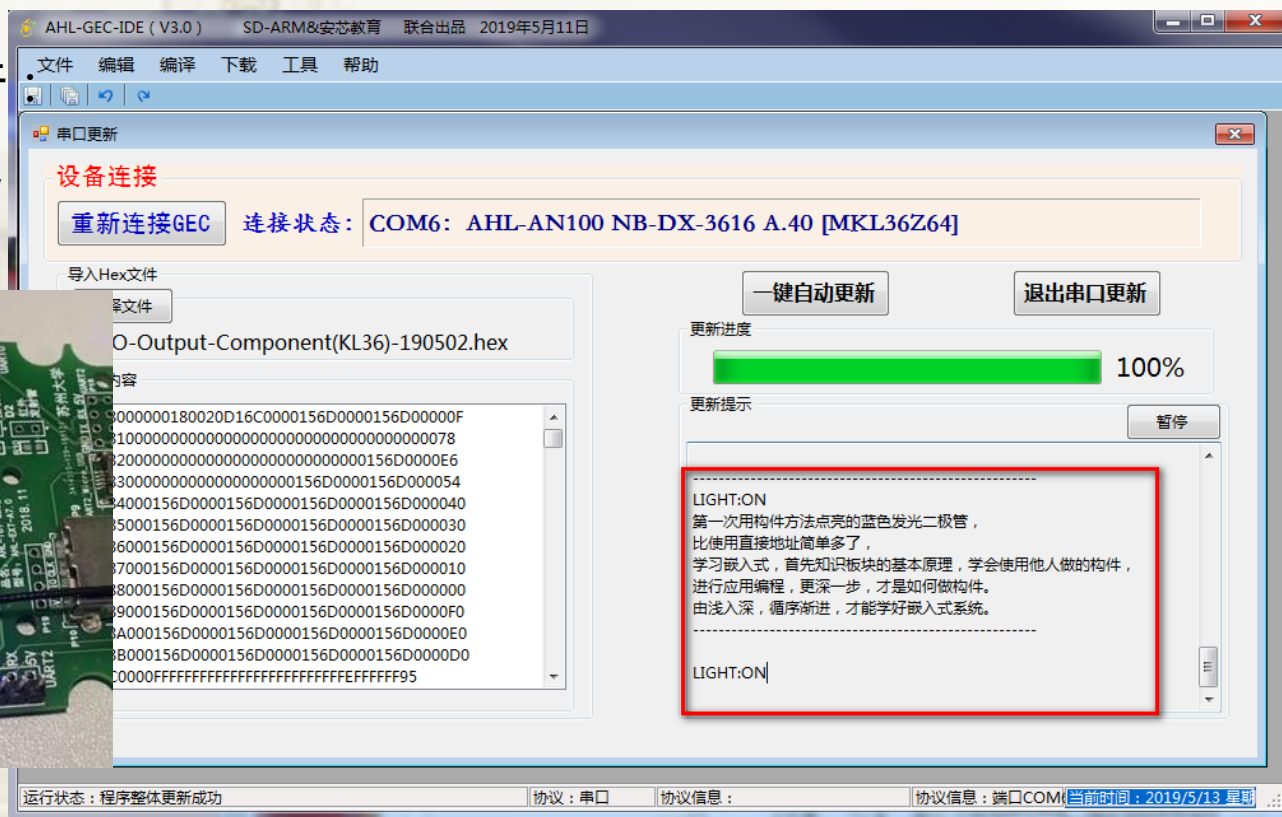
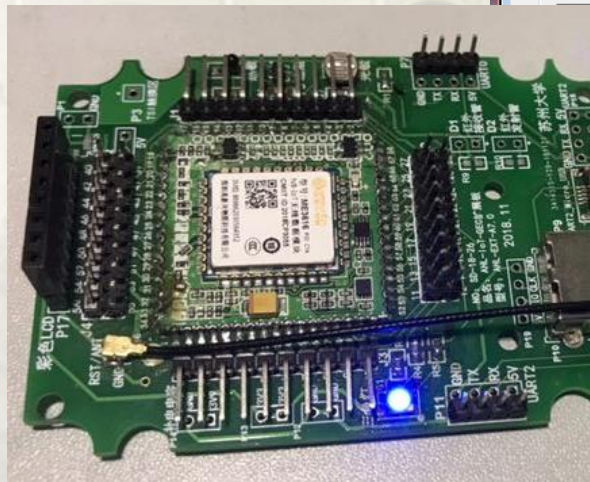
## 7.2

## 使用AHL-GEC-IDE打桩调试

### 05

### 观察开发板的运行状况

程序下载之后，自动运行  
蓝灯亮，  
串口输出 “LIGHT : ON”







苏州大学

SOOCHOW UNIVERSITY



## **7.3** 使用AHL-GEC-IDE进行Printf调试







## 7.2

### 使用AHL-GEC-IDE打桩调试-基本调试步骤

在程序需要查看变量值、  
函数语句执行状态等位  
置，增加printf语句

导入待调试工程

添加printf语句

编译烧录程序

观察串口输出情况

根据串口输出信息，  
判断程序运行状态  
是否符合预期。

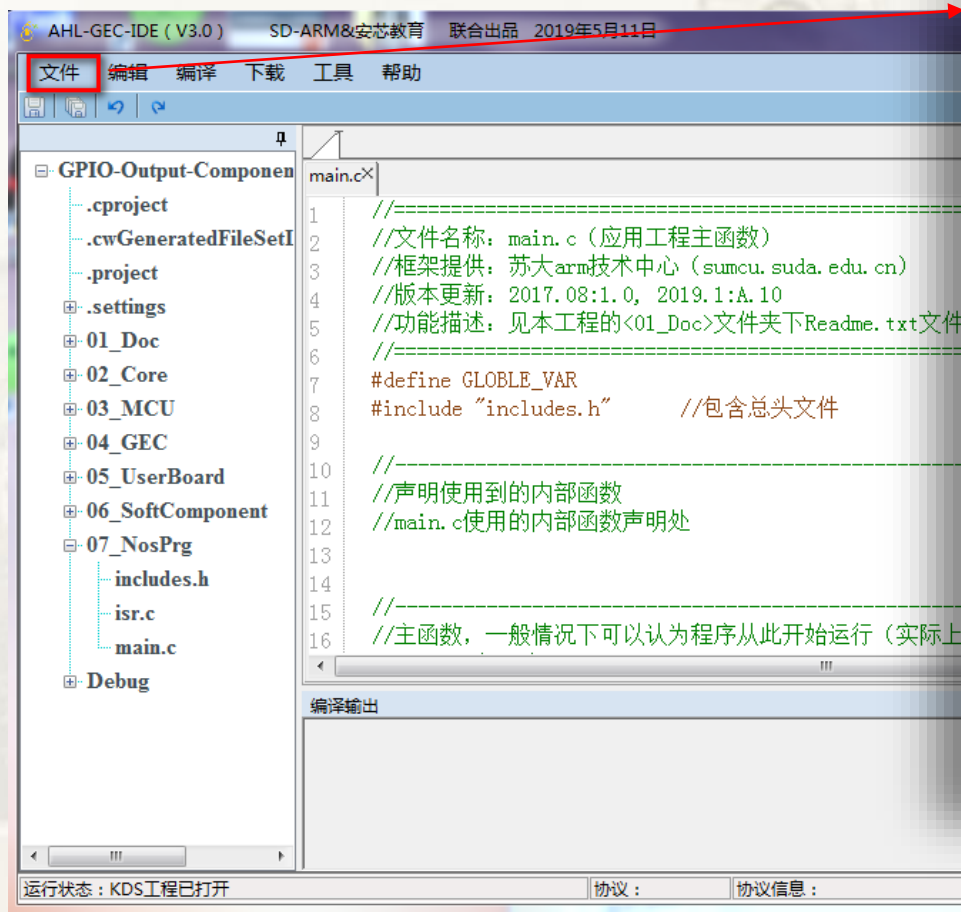


## 7.3

## 使用AHL-GEC-IDE进行Printf调试

### 01

### 导入工程



仍旧使用打桩调试处的样例：  
AHL-A00Kit01(V1.0)-190507\04-  
Soft\CH04-GPIO文件夹下的  
GPIO-Output-Component(KL36)

导入方法：  
AHL-GEC-IDE 工具菜单栏  
“文件” -> “导入工程”，  
选择样例工程所在文件夹。



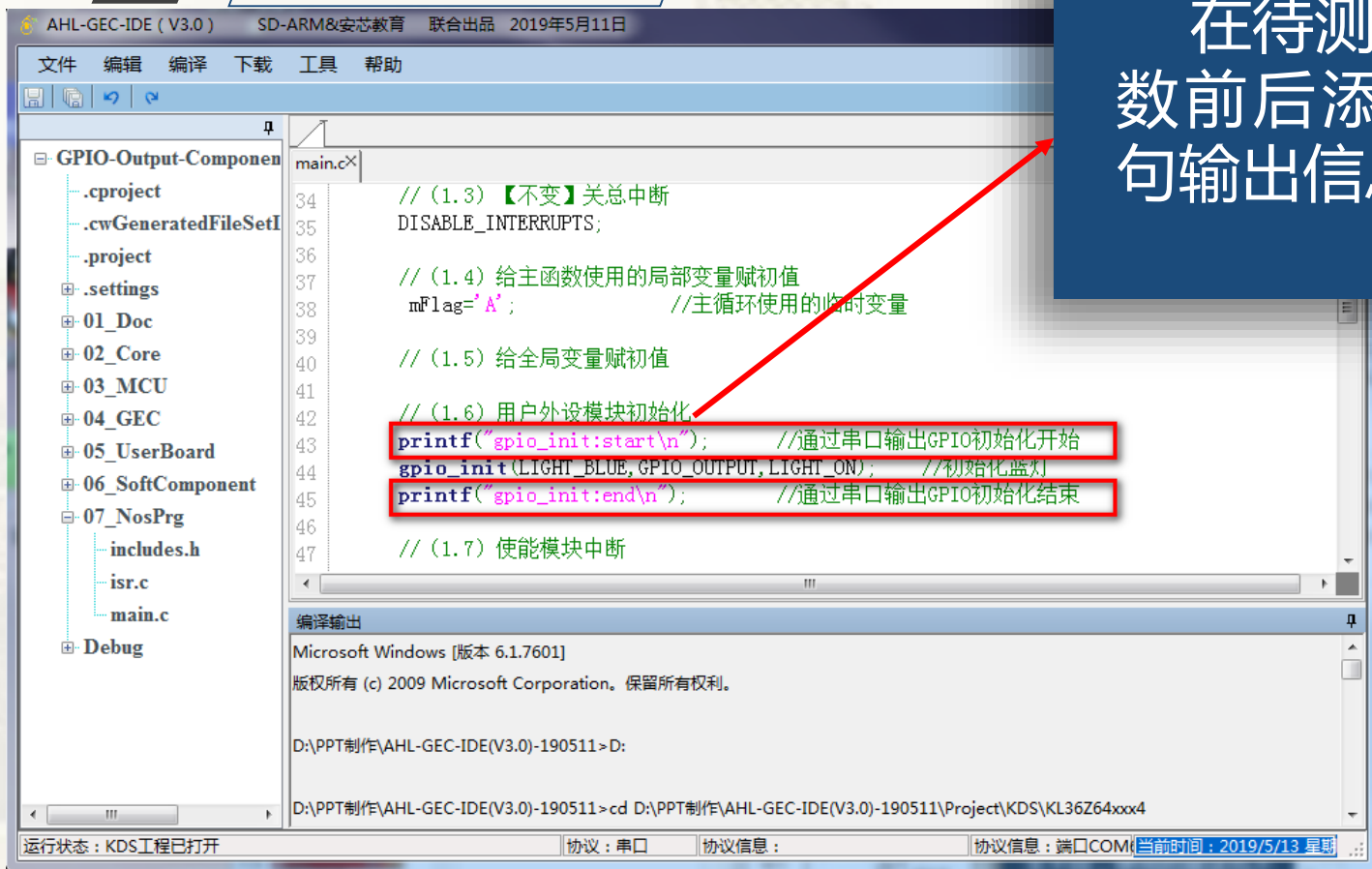
## 7.3

## 使用AHL-GEC-IDE进行Printf调试

## 02

## 程序中添加printf语句

在待测试语句或函数前后添加Printf语句输出信息





## 7.3

# 使用AHL-GEC-IDE进行Printf调试

## 03

## 编译烧录

The screenshot displays the AHL-GEC-IDE (V3.0) interface. On the left, the project tree shows the 'GPIO-Output-Component' selected. The main editor shows the 'main.c' file with C code for GPIO initialization and printing. The '编译输出' (Compilation Output) window at the bottom left shows the compilation command and a red box indicating '编译成功!' (Compilation successful!).

The '串口更新' (Serial Update) window is open, showing the '设备连接' (Device Connection) section with a red box around '重新连接GEC' (Reconnect GEC) and the connection status 'COM6: AHL-AN100 NB-DX-3616 A.40 [MKL36Z64]'. Below this, the '导入Hex文件' (Import Hex File) section shows the file 'GPIO-Output-Component(KL36)-190502.hex' selected. The '文件内容' (File Content) section displays the hex data. The '更新进度' (Update Progress) section shows a progress bar at 21% and a '一键自动更新' (One-click automatic update) button. The '更新提示' (Update Prompt) section shows the update status and a list of frames.

The status bar at the bottom indicates the current state: '运行状态: 编译成功' (Run status: Compilation successful) and '协议: 串口' (Protocol: Serial port).



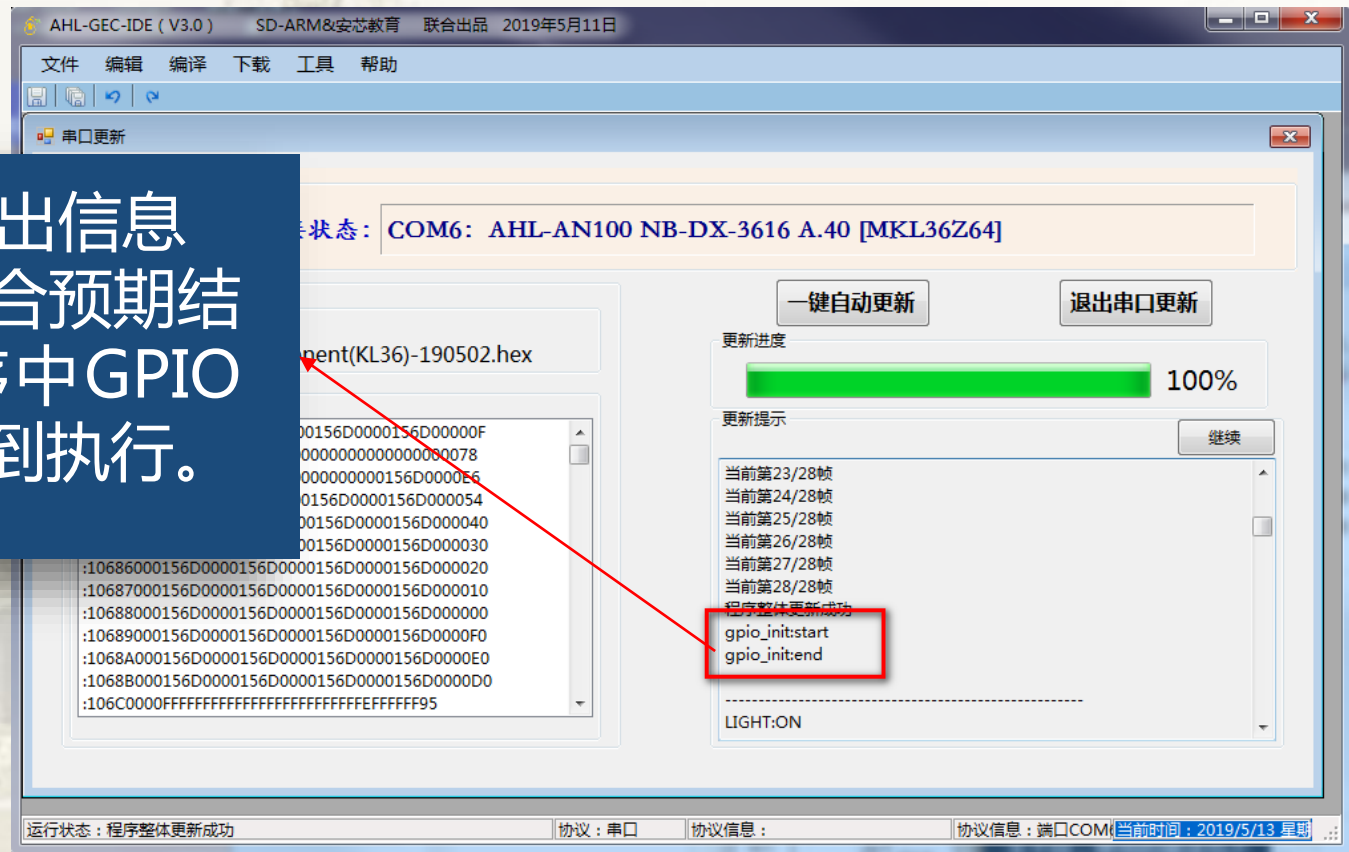
## 7.3

## 使用AHL-GEC-IDE进行Printf调试

### 04

### 观察串口输出情况

Printf语句输出信息  
得到显示，符合预期结  
果，表示程序中GPIO  
初始化语句得到执行。







苏州大学

SOOCHOW UNIVERSITY

arm

谢谢!

