苏州大学实验报告

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算机科学与技术 | | 年级专业 | 计科 | | 姓名 |  | | 学号 | |  |
| 课程名称 | | 微型计算机原理及应用实践 | | | | | | 同组实验者 | | | 无 |
| 指导教师 | | 王宜怀 | 实验日期 | | 2025.3.31 | | 成绩 | | |  | |

|  |  |
| --- | --- |
| 实 验 名 称 | 理解汇编程序框架及运行 |

一、实验目的

本实验通过编程控制LED小灯，并分析GPIO的输出作用，可将其扩展至能够控制蜂鸣器、继电器等；通过编程获取引脚状态，并分析GPIO的输入作用，可将其用于获取开关的状态。实验主要目的如下。

（1）了解集成开发环境的安装与基本使用方法。

（2）掌握GPIO构件基本应用方法，理解第一个汇编程序框架结构。

（3）掌握硬件系统的软件测试方法，初步理解printf输出调试的基本方法。

二、实验准备

（1）硬件部分。PC或笔记本计算机一台、AHL-CH573开发套件一套。

（2）软件部分。根据附录A，下载合适的电子资源。

（3）软件环境。按照附录A，进行有关软件工具的安装。

3. 参考样例

参照电子资源【03-Software\CH04\GPIO-ASM 工程，该样例是通过调用GPIO 驱动构件方式，使得一个发光二极管闪烁。使用构件方式编程干预硬件是今后编程的基本方式，因此要充分掌握构件的使用方法

三、实验过程或要求

（1）1） 验证性实验

第一次实验要做一些必要的一次性基础性工作。

（1） 下载与安装开发环境 AHL-GEC-IDE。按照附录 A，下载并安装由苏州大学嵌入式实验室开发的金葫芦集成开发环境（AHL-GEC-IDE），该集成开发环境针对本书使用的芯片CH573 兼容沁恒的开发环境 MounRiver。

（2）建立自己的工作文件夹。按照“分门别类，各有归处”之原则，建立自己的工作文件夹。并考虑随后内容安排，建立其下级子文件夹。

（3）拷贝模板工程并重命名。所有新工程可通过功能接近的样例工程来建立。例如，【03-SofewarelCH04）GPI0-ASM】工程到自己的工作文件夹，可以改为自己确定的工程名，建议尾端增加日期字样，避免混乱。

（4导入工程。在假设您已经下载了 AHL-GEC-IDE 并安装完成，则可以开始运行该软件。打开 AHL-GEC-IDE 之后，接着单击顶部菜单“文件”-“导入工程”，找到你要导入工

釋的文件夹，软件界面下部的“文件夹”提示框会显示该文件夹，请确认是否为你要打开的工程，若是，则单击软件界面右下角的“选择文件夹”按钮即可。导入工程后，左侧为工程树把目录，右边为文件内容编辑区，初始显示 main.s 文件的内容。

（S）编译工程。在已经打开工程，并显示文件内容的前提下，可编译工程。单击顶部染单“编译”一“编译工程”，则开始编译。

（6）下载并运行。第一步，硬件连接。用 Type-线连接PC 机与GEC：第二步，软件连接。单击顶部莱单“下载”一“串口更新”，将进入更新窗体界面。点击“连接”查找到目标

GEC，则提示“成功连接⋯”。第三步，下载机器码。点击“选择文件”按钮导入编译过程产生的Debug 文件中的hex 文件（者准生成时间，确认是自己现在编译生成的机器码），然后单击“一键自动更新”按钮，等待程序下载完毕，此时程序已经自动运行。若遇到问题可参阅附录A中的“常见问题及解决办法”进行解决。

（7）观察运行结果与程序的对应。为了表明程序已经开始运行了，在每个样例程序进入主循环之前，使用printf语句输出一段话，程序写入后立即执行，就会显示在开发环境下载界面的中的右下角文本框中，提示程序的基本功能。利用 printf语句将程序运行的结果直接输出到PC 机屏幕上，使得嵌入式软件开发的输出调试变得十分便利，与PC 机软件开始流程基本一致，改变了传统交叉调试模式。

2）设计性实验

在验证性实验的基础上，自行编程实现开发板上的红灯、蓝灯和绿灯交替闪烁。LED 三色灯电路原理参见图 4-3，对应三个控制端接 MCU 的三个 GPIO 引脚，具体的引脚对应参见样例工程中的汇编工程用户头文件 user.inc。

二极管是否与图中接法一致。

可以通过程序，测试你使用的开发套件中的发光

3） 进阶实验

对目标板上的三色灯进行编程，通过三色灯的不同组合，实现红、蓝、绿、青、紫、黄和白等灯的亮暗控制。灯颜色提示：青色为绿蓝混合，黄色为红绿混合，紫色为红蓝混合，白色为红蓝绿混合。

5. 实验报告要求

（1） 基本掌握 WORD 文档的排版方法。

（2）用适当文字、图表描述实验过程。

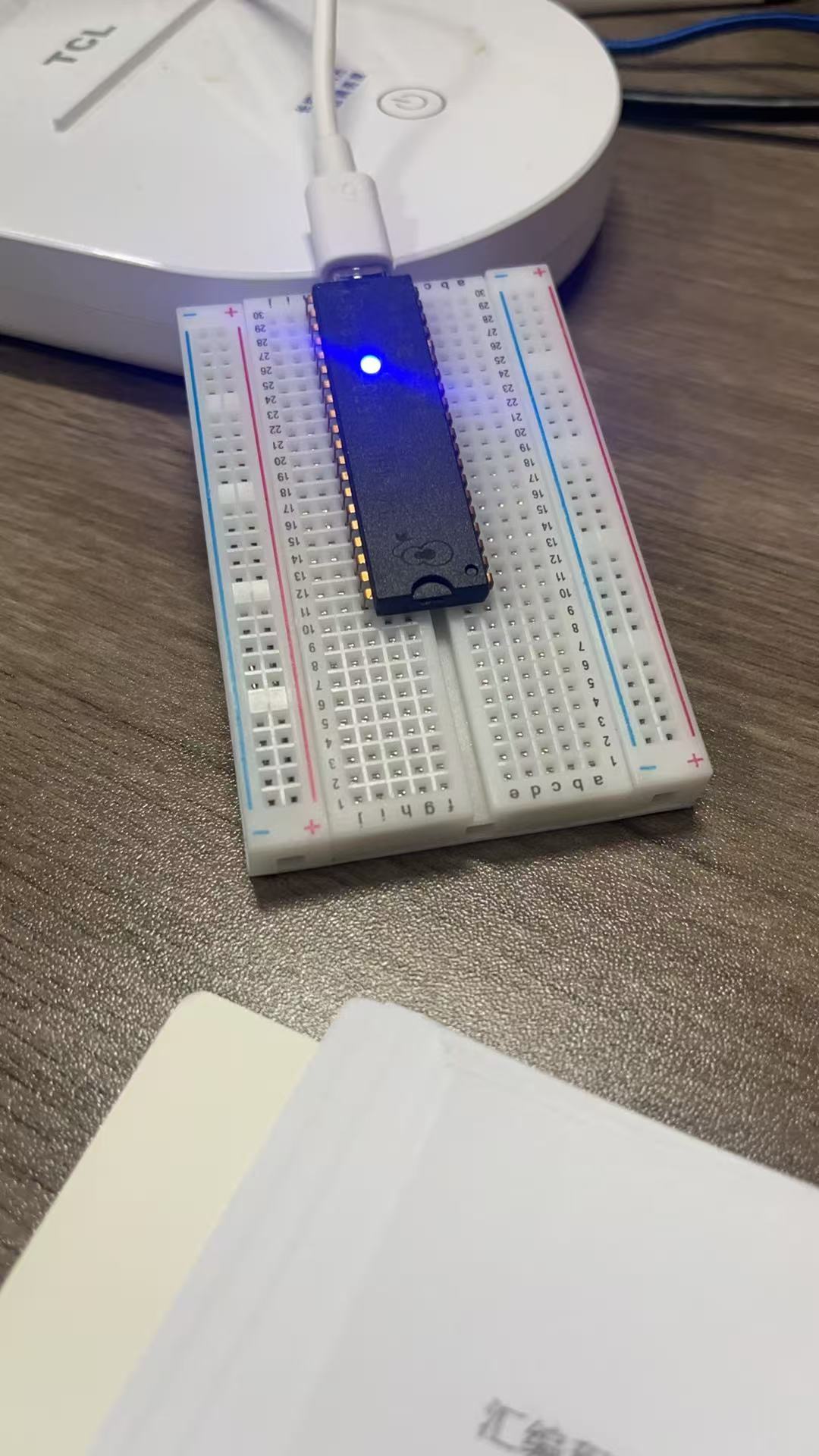
（3）用200~300字写出实验体会。

（4）在实验报告中完成实践性问答题。

四、实验结果

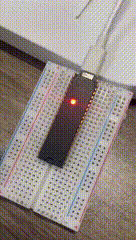
（1）验证性实验





蓝灯效果图

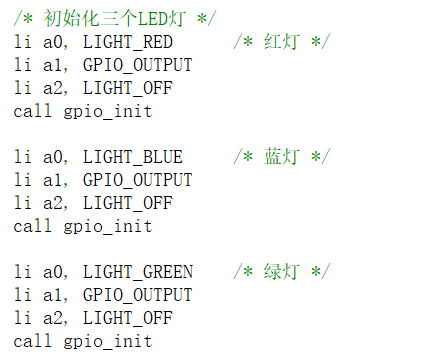
（2）设计性实验



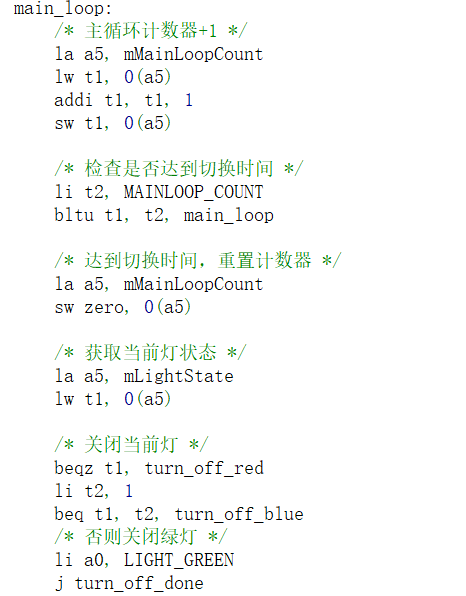
闪烁效果图

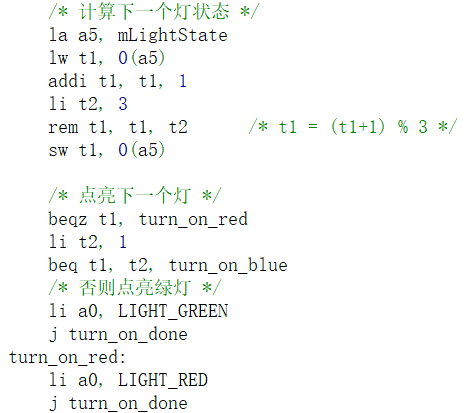
设计性实验关键代码：

* 初始化彩灯



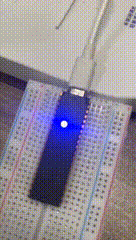
* 灯亮灭控制





（3）进阶实验





效果图

进阶实验代码：

1. .include "includes.inc"  *// 总头文件，包含硬件定义*
2. */\* 数据段定义 \*/*
3. .section .data
4. hello\_information:
5. .ascii "-------------------------------------------------------\n"
6. .ascii " 使用GPIO点亮七种颜色的二极管（RISC-V版本）\n"
7. .ascii "------------------------------------------------------\n\0"
8. light\_off:
9. .asciz "LIGHT:OFF--\n"
10. light\_red\_on:
11. .asciz "LIGHT\_RED:ON--\n"
12. light\_green\_on:
13. .asciz "LIGHT\_GREEN:ON--\n"
14. light\_blue\_on:
15. .asciz "LIGHT\_BLUE:ON--\n"
16. light\_count:
17. .asciz "灯的闪烁次数mLightCount=%d\n"
18. light\_type:
19. .asciz "当前显示颜色类型mLightType=%d\n"
20. .align 4
21. mMainLoopCount:  *// 主循环计数器*
22. .word 0
23. mFlag:          *// 灯状态标志 ('L'=亮, 'A'=灭)*
24. .byte 'L'
25. .align 4
26. mLightCount:    *// 闪烁次数计数器*
27. .word 0
28. .align 4
29. mLightType:     *// 颜色类型 (0-6)*
30. .word 0
31. .equ MainLoopNUM, 2000000  *// 主循环延迟常数*
32. */\* 代码段定义 \*/*
33. .section .text
34. .global main
35. .align 2
36. main:
37. */\* 关中断 \*/*
38. li t0, 0x8
39. csrc mstatus, t0
40. */\* 初始化GPIO \*/*
41. li a0, LIGHT\_RED
42. li a1, GPIO\_OUTPUT
43. li a2, LIGHT\_OFF
44. call gpio\_init
45. li a0, LIGHT\_GREEN
46. li a1, GPIO\_OUTPUT
47. li a2, LIGHT\_OFF
48. call gpio\_init
49. li a0, LIGHT\_BLUE
50. li a1, GPIO\_OUTPUT
51. li a2, LIGHT\_OFF
52. call gpio\_init
53. */\* 开中断 \*/*
54. li t0, 0x8
55. csrs mstatus, t0
56. */\* 显示欢迎信息 \*/*
57. la a0, hello\_information
58. call printf
59. main\_loop:
60. */\* 主循环计数器递增 \*/*
61. la a5, mMainLoopCount
62. lw t1, 0(a5)
63. addi t1, t1, 1
64. sw t1, 0(a5)
65. */\* 检查是否达到切换时间 \*/*
66. li t2, MainLoopNUM
67. bltu t1, t2, main\_loop
68. */\* 重置主循环计数器 \*/*
69. la a5, mMainLoopCount
70. sw zero, 0(a5)
71. */\* 检查灯状态标志 \*/*
72. la a5, mFlag
73. lb t1, 0(a5)
74. li t2, 'L'
75. bne t1, t2, main\_light\_off
76. */\* 灯状态为'L'（点亮阶段） \*/*
77. */\* 闪烁次数递增 \*/*
78. la a5, mLightCount
79. lw t1, 0(a5)
80. addi t1, t1, 1
81. sw t1, 0(a5)
82. */\* 打印闪烁次数 \*/*
83. la a0, light\_count
84. lw a1, 0(a5)
85. call printf
86. */\* 更新灯状态标志 \*/*
87. la a5, mFlag
88. li t1, 'A'
89. sb t1, 0(a5)
90. */\* 根据颜色类型点亮相应LED \*/*
91. la a5, mLightType
92. lw t1, 0(a5)
94. */\* 颜色类型判断 \*/*
95. beqz t1, light\_red        *// 0:红*
96. li t2, 1
97. beq t1, t2, light\_green   *// 1:绿*
98. li t2, 2
99. beq t1, t2, light\_blue    *// 2:蓝*
100. li t2, 3
101. beq t1, t2, light\_yellow  *// 3:黄(红+绿)*
102. li t2, 4
103. beq t1, t2, light\_purple  *// 4:紫(红+蓝)*
104. li t2, 5
105. beq t1, t2, light\_cyan    *// 5:青(绿+蓝)*
106. j light\_white             *// 6:白(红+绿+蓝)*
107. light\_red:
108. li a0, LIGHT\_RED
109. li a1, LIGHT\_ON
110. call gpio\_set
111. la a0, light\_red\_on
112. call printf
113. j end\_light\_type
114. light\_green:
115. li a0, LIGHT\_GREEN
116. li a1, LIGHT\_ON
117. call gpio\_set
118. la a0, light\_green\_on
119. call printf
120. j end\_light\_type
121. light\_blue:
122. li a0, LIGHT\_BLUE
123. li a1, LIGHT\_ON
124. call gpio\_set
125. la a0, light\_blue\_on
126. call printf
127. j end\_light\_type
128. light\_yellow:
129. li a0, LIGHT\_RED
130. li a1, LIGHT\_ON
131. call gpio\_set
132. la a0, light\_red\_on
133. call printf
135. li a0, LIGHT\_GREEN
136. li a1, LIGHT\_ON
137. call gpio\_set
138. la a0, light\_green\_on
139. call printf
140. j end\_light\_type
141. light\_purple:
142. li a0, LIGHT\_RED
143. li a1, LIGHT\_ON
144. call gpio\_set
145. la a0, light\_red\_on
146. call printf
148. li a0, LIGHT\_BLUE
149. li a1, LIGHT\_ON
150. call gpio\_set
151. la a0, light\_blue\_on
152. call printf
153. j end\_light\_type
154. light\_cyan:
155. li a0, LIGHT\_GREEN
156. li a1, LIGHT\_ON
157. call gpio\_set
158. la a0, light\_green\_on
159. call printf
161. li a0, LIGHT\_BLUE
162. li a1, LIGHT\_ON
163. call gpio\_set
164. la a0, light\_blue\_on
165. call printf
166. j end\_light\_type
167. light\_white:
168. li a0, LIGHT\_RED
169. li a1, LIGHT\_ON
170. call gpio\_set
171. la a0, light\_red\_on
172. call printf
174. li a0, LIGHT\_GREEN
175. li a1, LIGHT\_ON
176. call gpio\_set
177. la a0, light\_green\_on
178. call printf
180. li a0, LIGHT\_BLUE
181. li a1, LIGHT\_ON
182. call gpio\_set
183. la a0, light\_blue\_on
184. call printf
185. end\_light\_type:
186. */\* 打印当前颜色类型 \*/*
187. la a0, light\_type
188. la a5, mLightType
189. lw a1, 0(a5)
190. call printf
191. */\* 更新颜色类型 (0-6循环) \*/*
192. la a5, mLightType
193. lw t1, 0(a5)
194. addi t1, t1, 1
195. li t2, 7
196. rem t1, t1, t2      *// t1 = (t1+1)%7*
197. sw t1, 0(a5)
198. j main\_exit
199. main\_light\_off:
200. */\* 灯状态为'A'（熄灭阶段） \*/*
201. */\* 更新灯状态标志 \*/*
202. la a5, mFlag
203. li t1, 'L'
204. sb t1, 0(a5)
205. */\* 熄灭所有LED \*/*
206. li a0, LIGHT\_RED
207. li a1, LIGHT\_OFF
208. call gpio\_set
210. li a0, LIGHT\_GREEN
211. li a1, LIGHT\_OFF
212. call gpio\_set
214. li a0, LIGHT\_BLUE
215. li a1, LIGHT\_OFF
216. call gpio\_set
218. la a0, light\_off
219. call printf
220. main\_exit:
221. j main\_loop

五、实验体会及问答

问答：

1. 针对你所使用的芯片，给出汇编程序中的字变量、半字变量的数值范围。

字变量：-2,147,483,648 到 +2,147,483,647。

半字变量：-32,768 到+32,767。

（2）若声明一个字节变量，但在读写时错用了Iw、sw，会有问题吗？

有问题。用lw读取字节变量会读取包含目标字节的整个32位字导致读取额外3字节的相邻数据，导致数据污染。用sw写入字节变量 会写入整个32位字 覆盖相邻3字节内存，引发数据损坏。

（3）编写程序输出参考样例中mMainLoopCount变量的地址和值。

ldr r1, =mMainLoopCount

ldr r0, =data\_format

bl printf

（4）集成的红绿蓝三色灯最多可以实现几种不同颜色LED灯的显示？通过实验给出组合列表。

共七种：

红：红色；绿：绿色；蓝：蓝色

红+绿：黄色

红+蓝：紫色

绿+蓝：青色

红+绿+蓝：白色

体会：

通过本次实验，我了解了集成开发环境的安装与基本使用方法，掌握了GPIO构件基本应用方法、硬件系统的软件测试方法，初步理解了printf输出调试的基本方法。完成通过编程控制LED小灯，并分析GPIO的输出作用，可将其扩展至能够控制蜂鸣器、继电器等；通过编程获取引脚状态，并分析GPIO的输入作用，可将其用于获取开关的状态。

实验中，我学习了如何通过编程控制GPIO来操作LED灯，并分析了GPIO的输入输出功能。通过理论学习，我了解到GPIO不仅是简单的数字输入输出，更是嵌入式系统中实现各种功能的基础。经过实践，我实现了开发板上的红灯、蓝灯和绿灯交替闪烁。 通过对目标板上的三色灯进行编程，让三色灯进行不同组合，可以实现对红、蓝、绿、青、紫、黄、白等灯的亮暗控制。

通过这次实验，我不仅学习了汇编语言的基础知识，还掌握了硬件编程的基本技能。将理论知识应用到实践中，加深了我对汇编程序框架及运行的理解。