**《嵌入式系统及应用》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2019级计算机科学与技术卓越** | | | **姓名** | **李燕琴 李放** |
| **实验题目** | **UART 串联接口** | | | | | |
| **实验时间** | **2022年4月17日** | | **实验地点** | **DS3 305** | | |
| **实验成绩** |  | | **实验性质** | **■验证性 □设计性 □综合性** | | |
| 教师评价：  ■算法/实验过程正确； ■源程序/实验内容提交 ■程序结构/实验步骤合理；  ■实验结果正确； ■语法、语义正确； ■报告规范；  其他：  评价教师签名： | | | | | | |
| 1. 实验目的   锻炼学生对于嵌入式设备常见IO、器件的设计开发能力，积累实际动手开发能力，将以前所学习的数字逻辑、计算机组成原理、计算机体系结构、软件开发等课程进行综合，努力提高实际动手能力以及将理论转化为能力。 | | | | | | |
| 1. 实验项目内容 2. 学习如何把数字转为字符串； 3. 学习UART串联接口相关操作； 4. 学会如何通过UART接口展示字符信息。   **实验讲授内容：**  1）UART接口介绍；  2）数字如何转为ASCII；  3）相关展示函数  **学生实践内容：**   1. 学习UART接口相关知识。   编写代码验证接口功能 | | | | | | |
| 1. 实验过程或算法（源程序）   1.实验软硬件环境要求：  1）TM4C123G实验开发板一块  2）安装Keil uVision4的windows开发环境   1. 本实验使用实验板的PF1，PF2与PF3作为输出,使用PF0与PF4作为输入。串口通讯使用的是PC4(U1TX)与PC5(U1RX)。 2. 实验原理图   IMG_20151116235752  4. 实验代码   1. #include "PLL.h" 2. #include "SysTick.h" 3. #include "UART.h" 5. #define GPIO\_PORTF\_DATA\_R       (\*((volatile unsigned long \*)0x400253FC)) 6. #define GPIO\_PORTF\_DIR\_R        (\*((volatile unsigned long \*)0x40025400)) 7. #define GPIO\_PORTF\_AFSEL\_R      (\*((volatile unsigned long \*)0x40025420)) 8. #define GPIO\_PORTF\_PUR\_R        (\*((volatile unsigned long \*)0x40025510)) 9. #define GPIO\_PORTF\_DEN\_R        (\*((volatile unsigned long \*)0x4002551C)) 10. #define GPIO\_PORTF\_LOCK\_R       (\*((volatile unsigned long \*)0x40025520)) 11. #define GPIO\_PORTF\_CR\_R         (\*((volatile unsigned long \*)0x40025524)) 12. #define GPIO\_PORTF\_AMSEL\_R      (\*((volatile unsigned long \*)0x40025528)) 13. #define GPIO\_PORTF\_PCTL\_R       (\*((volatile unsigned long \*)0x4002552C)) 14. #define SYSCTL\_RCGC2\_R          (\*((volatile unsigned long \*)0x400FE108)) 15. #define GPIO\_PORTF\_PDR\_R        (\*((volatile unsigned long \*)0x40025514)) 17. **void** PortF\_Init(**void**){ **volatile** unsigned **long** delay; 18. SYSCTL\_RCGC2\_R |= 0x20;         // 1) 激活Port F 19. delay = SYSCTL\_RCGC2\_R;         // allow time for clock to stabilize 20. GPIO\_PORTF\_LOCK\_R = 0x4C4F434B; // 2) unlock Port F lock 21. GPIO\_PORTF\_CR\_R = 0x1F;         //   允许PF4-0改变配置 22. GPIO\_PORTF\_AMSEL\_R = 0x00;      // 3) 禁用PF4-0的模拟信号量 23. GPIO\_PORTF\_PCTL\_R = 0x00000000; // 4) 配置 PF4-0 为 GPIO并行输入输出端口 24. GPIO\_PORTF\_DIR\_R = 0x0E;        // 5) PF4 and PF0 为输入, PF3-1 为输出 01110 25. GPIO\_PORTF\_AFSEL\_R = 0x00;      // 6) disable alt funct on PF4-0 26. GPIO\_PORTF\_DEN\_R = 0x1F;        // 7) 启用PF4-0的I/O信号量 27. // GPIO\_PORTF\_PUR\_R = 0x11;        //    enable pull-up on PF4 and PF0 28. GPIO\_PORTF\_PUR\_R = 0x00;        //    disable pull-up on PF4 and PF0 29. GPIO\_PORTF\_PDR\_R = 0x11;        //    enable pull-down on PF4 and PF0 30. } 32. //  red, yellow, green, light blue, blue, purple,  white,  dark 33. **const** **long** ColorWheel[8] = {0x02,0x0A,0x08,0x0C,0x04,0x06,0x0E,0x00}; 34. **int** main(**void**){ 35. unsigned **long** SW1,SW2; 36. **long** prevSW1 = 0;        // SW1的先值 37. **long** prevSW2 = 0;        // SW2的先值 38. unsigned **char** inColor;   // 其他实验板传入的颜色的值 39. unsigned **char** color = 0; // 当前LED灯的颜色 40. PLL\_Init();              // 设置系统时钟为80 MHz 41. SysTick\_Init();          // initialize SysTick 42. UART\_Init();             // initialize UART 43. PortF\_Init();            // initialize buttons and LEDs on Port F 44. **while**(1){ 45. SW1 = GPIO\_PORTF\_DATA\_R&0x10; // Read SW1 46. **if**((SW1 == 0) && prevSW1){    // falling of SW1? 47. color = (color+1)&0x07;     // 在按下开关SW1时进入ColorWheel的下一种颜色 48. } 49. prevSW1 = SW1; // 当前按下的是SW1 50. SW2 = GPIO\_PORTF\_DATA\_R&0x01; // Read SW2 51. **if**((SW2 == 0) && prevSW2){    // falling of SW2? 52. UART\_OutChar(color+0x30);   // 在按下开关SW2时发送颜色值的ASCALL码，as '0' - '7' 53. } 54. prevSW2 = SW2; // 当前按下的是SW2 55. inColor = UART\_InCharNonBlocking(); 56. **if**(inColor){ // FIFO是否有新到达的数据 57. color = inColor&0x07;     // 更新当前实验板的颜色 58. } 59. GPIO\_PORTF\_DATA\_R = ColorWheel[color];  // 更新LED灯 60. SysTick\_Wait10ms(2);        // 防止开关反跳，即在20ms之内按下两次 61. } 62. } | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程   1、针对自己的LaunchPad，按下sw1时，3个LED灯变换8种状态中的下一个状态。没有按sw1时，LED灯保持状态。如下图所示：      2、针对队友的LaunchPad，按下sw2时，将自己当前3个LED灯状态传送到队友的LaunchPad上，其接收并将LED灯置位为该状态。如下图所示： | | | | | | |
| 1. 实验心得   本实验学习了UART端口将并联信号转为串联信号进行传输，并将串联信号转为并联信号进行接受，实现简单的两个LaunchPad互信的过程；并设计FSM，让3个LED灯在8中状态中进行轮转。 | | | | | | |