实验课程名称： 嵌入式系统设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | ARM IO接口实验 | | | **实验成绩** |  |
| **实 验 者** | 林兆先 | **专业班级** | 测控2203 | **理论课序号** | 81 |
| **同 组 者** |  | | | **实验日期** | 2025年5月30日 |
| 第一部分：实验预习报告   1. 实验目的   （1）掌握ARM处理器的I/O接口基本原理和编程方法。  （2）熟悉ARM的I/O端口配置、数据读写及控制操作。  （3）通过实验了解如何通过ARM开发环境（ADS1.2）进行代码编写、编译、仿真和调试。  （4）培养使用嵌入式系统开发工具进行软硬件协同设计的能力。   1. 实验内容   （1）连接并配置ARM开发板与PC机串口，实现基本通信。  （2）利用ADS1.2开发环境编写和调试I/O控制程序，控制CPU板上LED灯交替闪烁。  （3）观察并分析LED灯的状态变化，理解I/O口配置寄存器（如GPGCON、GPGDAT）的作用。  （4）修改代码实现LED灯同亮同灭的效果。  （5）针对不同硬件端口连接（如PORTG第3、4位），调整代码进行适配。   1. 实验设备 2. ARM CPU开发板 3. PC机（安装有ADS1.2开发环境）   （3）连接线（串口直连线）  （4）电源设备（CPU板电源和试验箱电源）  （5）配套软件工具：ADS1.2集成开发环境，超级终端（配置串口通信）   1. 实验原理   本实验基于 ARM 处理器的 I/O 接口，通过控制 GPIO（通用输入输出）端口实现外部设备的控制。实验通过使用 ARM 2410 的 PORTG 接口对外部 LED 灯进行控制，学习如何在嵌入式系统中配置和操作 I/O 接口。    GPIO 配置：GPIO 引脚可配置为输入或输出，通过设置对应寄存器（如 GPGCON）的值，来选择每个引脚的工作模式（输入/输出）。例如，将 PORTG 的第 3、4 位设置为输出，可以控制连接到这些引脚的外部设备（如 LED）。  控制输出：通过 GPGDAT 寄存器来控制数据输出。当 GPIO 配置为输出时，向该寄存器写入相应的值（高或低电平）即可控制外部设备的状态，如点亮或熄灭 LED。  延时控制：为了实现 LED 闪烁的效果，通过软件延时的方式控制程序运行速度，使得 LED 状态变化能被肉眼观察到。延时通常通过循环实现。  实验环境与调试：使用 ADS1.2 开发环境进行程序编译与调试，确保代码正确加载并能在硬件平台上正确运行。 | | | | | |
| 第二部分：实验过程记录  实验过程记录：   1. 首先，先打开电脑的超级终端，配置串口的属性（如COM1），配置波特率为115200，校验位无，数据位为8，停止位为1，数据控制流为无；      1. 打开CodeWarrior for ARM Developer Suite，打开第一个实验的IO程序，加载的可编辑程序界面，点击其中的Main.c程序，如下图所示：      1. 编辑并保存程序，同时对程序进行Make编译，在没有报错的情况下点击Project选项下的Remove Object Code选项，得到我们程序的可执行文件。          1. 接下来，打开AXD Debugger软件，进行程序的调试，打开后如下所示：          1. 执行后，单片机所对应的LED灯开始进行闪烁，呈现出两个LED灯在间隔一段时间后同时亮灭的效果： | | | | | |
| 第三部分 思考题  1. 下图每个文件完成什么功能？  根据提供的文件结构图，下面是每个文件及其功能的简要说明：  Application 文件夹   * config.h：这个文件通常用于存放与系统配置相关的宏定义和设置，例如时钟配置、端口配置等。 * Main.c：主程序文件，通常包含程序的入口点 main() 函数。在此文件中，会初始化硬件资源、配置系统并控制程序的主逻辑。   Startup2410 文件夹   * 2410addr.h：包含 ARM 2410 系统的地址定义文件，用于标识内存或硬件寄存器的地址。 * 2410lib.h：包含 ARM 2410 系统库的头文件，可能包含一些基础函数的声明。 * 2410slib.h：同样是 ARM 2410 系统相关的库文件，可能包含不同于 2410lib 的其他支持性功能。 * def.h：一般用于定义常量、宏、数据结构等，便于代码的统一管理和修改。 * mmu.h：与内存管理单元（MMU）相关的头文件，负责内存保护、分页等操作。 * option.h：可能用于定义一些编译选项、调试选项或平台特定的参数。   SRC 文件夹   * TARGET.C：包含与目标硬件平台相关的源代码，通常涉及硬件初始化、设备驱动的配置。 * 2410INIT.S：启动汇编文件，用于初始化 ARM 2410 处理器及其硬件组件。 * 2410LIB.C：提供 ARM 2410 相关的底层库函数实现，可能包括一些硬件操作的函数。 * 2410SLIB.S：与 2410 相关的汇编文件，实现特定的硬件操作或底层功能。 * MMU.C：实现内存管理单元（MMU）的功能，例如内存分配、映射等。 * 2410addr.s：汇编文件，定义 ARM 2410 相关的内存地址。   UartDriver 文件夹   * Vart\_driver.c：实现串口驱动程序的源代码，提供串口的初始化、发送和接收功能。 * Vart\_driver.h：包含串口驱动程序的头文件，定义串口操作的函数原型、数据结构等。   2. 如果硬件连接采用了PORTG的第3,4位，代码应该如何修改？void Main(void){  int flag, i;  Target\_Init();  for(;;){  if(flag==0){  for(i=0;i<1000000;i++); // 延时  rGPGCON = rGPGCON & 0xFF0FFFFF | 0x00050000; // 设置GPG3、GPG4为输出  rGPGDAT = rGPGDAT & 0xFF0FFFFF | 0x00080000; // 设置GPG3为高电平，GPG4为低电平  for(i=0;i<1000000;i++); // 延时  flag = 1;  }  else{  for(i=0;i<1000000;i++); // 延时  rGPGCON = rGPGCON & 0xFF0FFFFF | 0x00050000; // 设置GPG3、GPG4为输出  rGPGDAT = rGPGDAT & 0xFF0FFFFF | 0x00000000; // 设置GPG3、GPG4为低电平  for(i=0;i<1000000;i++); // 延时  flag = 0;  }  }  } | | | | | |