**《嵌入式系统原理与实验》实验思考与练习题**

1. 在实验1.3中采用了APB访问方式还是AHB访问方式？请采用另一种访问方式改写程序。
2. 阅读头文件hw\_types.h、hw\_memmap.h、hw\_sysctl.h、hw\_gpio.h，了解它们的作用。函数HWREG( )的作用是什么？
3. 在实验1.3中，如要求配置GPIO引脚的输出为4mA，LED0点亮，LED1熄灭，试修改、调试程序。
4. 在实验1.4中，每按一次五向键，LED灯产生相应的反应，试修改程序，达到如下功能：每连续按两次五向键，LED灯产生相应的反应。

执行GPIOPinRead语句，对应的键被按下，相应的位被置为0

1. 利用宏函数HWREGBITW()、HWREGBITH()或HWREGBITB()实现实验相同的功能。
2. 利用LM3S9B96微控制器片上ROM驱动库，完成实验1.3的功能。
3. 查阅实验板的电路图和相关资料，看看该实验板使用了几片晶振。芯片LM3S9B96的主晶振的频率为多少？
4. 在本实验的主程序中包含如下语句：

#define Second 4000000

#define MilliSecond 4000

#define MicroSecond 4

试分析其含义。如果将主程序中的语句

SysCtlClockSet(SYSCTL\_SYSDIV\_1 |

SYSCTL\_USE\_OSC |

SYSCTL\_OSC\_MAIN |

SYSCTL\_XTAL\_16MHZ);

修改为

SysCtlClockSet(SYSCTL\_SYSDIV\_2 |

SYSCTL\_USE\_OSC |

SYSCTL\_OSC\_MAIN |

SYSCTL\_XTAL\_16MHZ);

试重新改写Second、MilliSecond、MicroSecond定义值。

1. 将时钟设置语句改为：

SysCtlClockSet(SYSCTL\_USE\_PLL |

SYSCTL\_OSC\_MAIN |

SYSCTL\_XTAL\_16MHZ |

SYSCTL\_SYSDIV\_4);

试观察系统时钟频率为多少？

1. 试编程将时钟设置为33.33MHz。验证你的设置。
2. 试改写实验3.1中的程序，使黄色网络灯(LED0)和绿色网络灯(LED1)轮流点亮的时间为1s。
3. 试改写实验3.1中的程序，使黄色网络灯(LED0)和绿色网络灯(LED1)轮流点亮的时间为0.789s。
4. 试改写实验3.2中的程序，使黄色网络灯(LED0)和绿色网络灯(LED1)的闪烁周期时间为0.8s。
5. 设置GPTM0模块为16位周期定时器，完成与实验3.1相同的功能。
6. 使用中断的方式获取五向键的状态，并以此来控制LED灯的亮灭和闪烁：按下五向键中除Press外的另外4个键，黄色网络灯(LED0)和绿色网络灯(LED1)将开始闪烁：

左键：黄灯闪烁周期为0.5s；绿灯1s；

右键：黄灯闪烁周期为1s；绿灯0.5s；

前键：黄灯闪烁周期为1s；绿灯1s；

后键：黄灯闪烁周期为0.5s；绿灯0.5s；

若按下五向键中的Press键则停止闪烁。

1. 修改系统时钟频率为33.33MHz，完成题4.1同样的功能。

TimerPrescaleSet(TIMER0\_BASE,TIMER\_A,255);

1. 修改程序，将看门狗周期设定为4s。
2. 利用微控制器看门狗定时器的中断功能，完成实验系统的任务。
3. 修改系统时钟频率为33.33MHz，完成题5.1同样的功能。

UART传输高位先传过来~

1. 修改例程，完成中断方式的UART数据接收：
2. 在main函数中使能UART0接收中断和接收超时中断，并进入无限循环；
3. 编写UART中断服务程序void UARTIntHandler(void)，获取中断状态并清中断，同时获取接收FIFO中的所有字符，完成状态3的操作，即大小写字母互换后回显；
4. 注册UART中断服务程序。
5. 实验例程采用函数UARTCharGet ()和UARTCharPut()完成数据接受和发送。试参考驱动库函数手册，采用UARTCharGetNonBlocking ()和UARTCharPutNonBlocking ()完成数据接受和发送，并比较两种方式的区别。

while(UARTCharsAvail(UART0\_BASE))

{

tempChar = UARTCharGetNonBlocking(UART0\_BASE);

后续内容写在while函数里面

1. 本例程UART模块的FIFO设置的深度是多少？

默认为1/2

1. 编程实现下述任务：编写开发板上的UART设置程序，使得在电脑主机端使用超级终端程序，能够完成最少两条AT指令如下

1) 主机端输入AT，返回OK

2) 主机端输入AT+IPR?，返回+IPR 19200

3) 除以上两条正确指令外，其余错误输入如AT+IPR，均返回ERROR

1. 阅读实验7.1源程序，对照S700硬件电路图分析语句SystemState |= I2C0PullUpTest(); 的执行结果。
2. 修改实验7.1源程序，使系统定时器的中断周期为100ms，观察实验效果。你认为系统定时器的中断周期取多少为合适？
3. 阅读实验7.2源程序，指出数组NixieTube[6]中各个元素的含义。

前四位为米字管各位，第五位为：，第六位为\0

I2CMasterBusBusy主要用于检查数据传输是否完成，完成后再检查传输是否有错，就完成一次传送

1. 阅读实验八（1）例程，指出程序是如何实现呼吸灯效果的。
2. 修改实验八（1）例程，实现不同的呼吸灯效果。可修改的参数包括PWM周期、PWM输出脉宽的修改周期、PWM输出脉宽的修改步进值等。
3. 阅读实验八（2）例程，指出程序是如何实现不同蜂鸣声效果的。
4. 阅读Stellaris® Graphics LibraryUSER’S GUIDE，理解函数void GrContextInit (tContext \*pContext, const tDisplay \*pDisplay)的含义，了解结构类型tContext、tDisplay的内容。
5. 阅读下述函数：

static void LCDWriteData(unsigned short usData)

static void LCDWriteCommand(unsigned char ucData)

理解LCD写命令字和LCD写数据的区别。

1. 修改例程，使LCD上显示内容的字体全部为g\_sFontCm14，并重新合理布局显示内容在LCD上的位置。
2. 修改例程，实现当LCD上显示“Pause”时米字管显示“00：00”。
3. 修改例程，实现当LCD上显示“Running”、米字管显示时钟时，发光二极管D5～D7、D13～D16按序点亮，且与时钟显示节拍一致。
4. 修改例程，实现当LCD上显示“Pause”、米字管显示“59：59”时，将“59：59”发送到串口终端。
5. 阅读例程，修改ADC0模块的过采样率为8或者32，观察采样结果的稳定性。
6. 修改实验十（1）例程，使UP键和DOWN键的功能对调，即按下UP键时根据范围值确定电子钟减1秒，减10秒，减1分钟，减10分钟的显示；按下DOWN键时根据范围值确定电子钟加1秒，加10秒，加1分钟，加10分钟的显示。
7. 修改实验十（1）例程，利用微控制器的ADC1模块完成实验的功能。
8. 将实验十（2）中函数void ADCInitial(void)中的程序段：

ADCSequenceEnable(ADC0\_BASE,1);

ADCSequenceConfigure(ADC0\_BASE,1,ADC\_TRIGGER\_PROCESSOR,0);

ADCSequenceStepConfigure(ADC0\_BASE,1,0,ADC\_CTL\_TS |

ADC\_CTL\_END |

ADC\_CTL\_IE);

修改为

ADCSequenceEnable(ADC1\_BASE,1);

ADCSequenceConfigure(ADC1\_BASE,1,ADC\_TRIGGER\_PROCESSOR,0);

ADCSequenceStepConfigure(ADC1\_BASE,1,0,ADC\_CTL\_TS |

ADC\_CTL\_END |

ADC\_CTL\_IE);

为保证该实验原来的功能，请完成例程其它部分的修改。

1. 阅读中断服务例程void I2C0\_ISR(void)，理解该程序是如何来管理I2C设备（米字管、发光二极管、加速度计）的。指出程序进入I2C中断服务的条件。
2. 修改流程，将LCD上显示项目“Acceleration Dir-X:”与“Acceleration Dir-Z:”的显示位置对调，相应的显示内容也对调。
3. 阅读例程并参考ADXL345的数据手册，指出程序是如何设置加速度计设置加速度的测量范围及数据输出格式的。
4. 阅读例程并参考ADXL345的数据手册，说明例程是如何读取加速度计中的三维加速度的。