

A

西安电子科技大学

考试时间 120 分钟

试 题

题号	一	二	三	四	五	六	七	八	九	十	总分
分数											

1. 考试形式：闭卷 ☐ 开卷 ☐

2. 考试日期：2021 年 12 月 31 日(答题内容请写在装订线外)

一、选择题（共 20 分，每题 2 分，请将正确选项填写在下面表格中）

题号	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
答案										

1. 从广义角度来说，嵌入式系统是一切非 PC 和大型机的计算机系统。嵌入式系统以 (1) 为中心，以计算机技术为基础。嵌入式处理器是嵌入式系统的核心，常用的嵌入式处理器不包括 (2)。嵌入式操作系统是嵌入式系统的重要组成，下列不属于嵌入式操作系统的是 (3)。

(1) A. 需求

B. 应用

C. 设计

D. 测试

(2) A. ARM

B. PowerPC

C. x86

D. DSP

(3) A. uC/OS-II

B. Android

C. iOS

D. Windows 10

2. ARM 的 Cortex 系列处理器中，面向手机处理器应用的系列是 (4)。STM32F103 作为一款基于 Cortex-M3 内核的微控制器，有多种低功耗运行模式，其中功耗最低的是 (5)。基于 STM32F103 微控制器构建一个最小系统（指使用最少的电子元件使微控制器正常工作的系统），在最小系统中可以不包括 (6)。使用小端格式将字数据 0x234567 存放到地址 0x20002000 上时，则地址 0x20002001 上的字节数据是 (7)。

(4) A. Cortex-A

B. Cortex-M

C. Cortex-R

D. 以上都不是

(5) A. 正常运行模式

B. 睡眠模式

C. 停机模式

D. 待机模式

(6) A. 电源系统

B. 时钟系统

C. 复位系统

D. 外部存储

(7) A. 0x67

B. 0x45

C. 0x23

D. 0x00

3. 中断就是通知 CPU 当前有某个事件发生, STM32 中决定中断响应顺序的因素有中断的①抢占优先级、②子优先级、③中断向量表中的位置, 则确定中断响应顺序时对上述因素判断顺序是_(8)_. DMA 的传输方式有三种, 如果要传输 100 个字节的数据, 能够保证在传输这 100 个字节过程中不被打断的传输方式是_(9)_. ADC 的四阶段模/数转换过程包括采样、保持、量化和编码, STM32F103 的 ADC 是逐次逼近型, 一次转换的总共转换时间_(10)_.

(8) A. ①②③

B. ①③②

C. ②①③

D. ③②①

(9) A. 周期挪用

B. 请求方式

C. 快传输

D. 以上都不行

(10) A. 只包括采样时间

B. 只包括量化编码时间

C. 包括采样和量化编码时间

D. 完全不可预测

二、填空题 (共 20 分, 每空 1 分)

4. ARM Cortex-M3 内核里中断屏蔽寄存器有 3 个, 分别是_____、
_____、_____。

5. ARM Cortex-M3 处理器的运行模式有两种, 分别是_____; ARM Cortex-M3 处理器的特权级别有 2 种, 分别是
_____、_____。

6. STM32F103 上 ADC 的转换模式包括_____, _____、
_____以及间断模式, STM32F103 上 ADC 的通道组有两种, 分别是
_____, _____。

7. 在 uC/OS-II 中任务状态有 5 种, 分别是_____, _____、
_____, _____、_____, _____。

8. 如果把嵌入式 Linux 软件划分为 4 个层次, 那么除了面向用户的应用程序, 还应
包括_____, _____、_____。

三、简答题（共 15 分，每题 5 分）

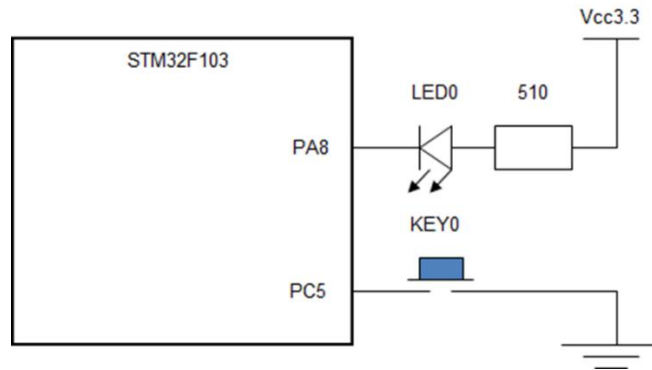
9. 简述 UART 的数据格式组成和传输流程。

10. 简述 SPI 总线的接口组成。

11. 简述 I2C 总线与 UART、SPI 总线相比的优缺点。

四、程序题（共 45 分，第 1 题 10 分，第 2 题 14 分，第 3 题 21 分）

1. 根据下图所示的硬件原理图以及要实现的功能，把空缺的代码补充完整。



要实现的功能如下：

采用查询方式，KEY0 按下时 LED0 灯灭，KEY0 抬起时 LED0 灯亮。

【问题】代码实现如下，把空缺的代码补充完整，每处 2 分，答案写在横线上。

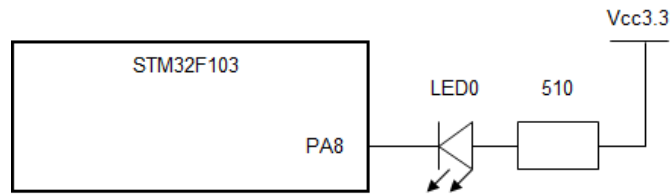
```
/* 使能 GPIOA 和 GPIOC 的时钟 */
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC, ENABLE);

/* 配置 GPIOA.8 */
GPIO_InitStructure.GPIO_Mode = _____;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8;    //PA.8
GPIO_Init(GPIOA, &GPIO_InitStructure);

/* 配置 GPIOC.5 */
GPIO_InitStructure.GPIO_Mode = _____;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;    //PC.5
GPIO_Init(GPIOC, &GPIO_InitStructure);

/* 主循环 */
while (1)
{
    if (_____)
        _____;
    else
        _____;
}
```

2. 根据下图所示的硬件原理图以及要实现的功能，把空缺的代码补充完整。



要实现的功能如下：

采用 PWM 方式控制 LED0 的闪烁，使其周期性地点亮 1 秒、熄灭 1 秒。假设 TIMxCLK 为 8MHz，根据如下定时周期计算方法，配置 TIM1 及其 PWM1 的输出，TIM1 采用向上计数模式，使得 LED0 周期性闪烁。

$$T = \frac{(\text{TIM_Period} + 1) \times (\text{TIM_Prescaler} + 1)}{\text{TIMxCLK}}$$

【问题】代码实现如下，把空缺的代码补充完整，每处 2 分，答案写在横线上。

```
/* 配置 GPIO */
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 ;
GPIO_InitStructure.GPIO_Mode = _____;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure);

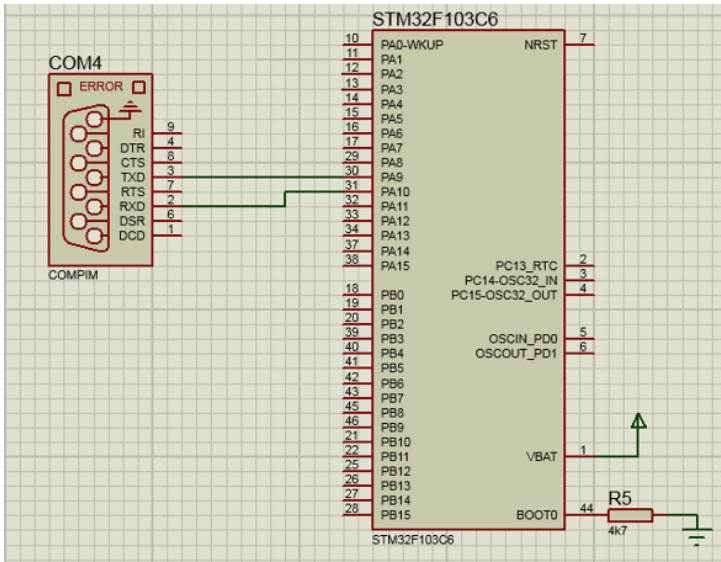
/* 配置 TIM1，定时周期为 2s */
TIM_TimeBaseStructure.TIM_Prescaler = _____; //0-65535
TIM_TimeBaseStructure.TIM_Period = _____; //0-65535
TIM_TimeBaseStructure.TIM_ClockDivision = 0;
TIM_TimeBaseStructure.TIM_CounterMode = _____;
TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);

/* 配置 TIM1_CH1 为 PWM 模式 */
TimOCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
TimOCInitStructure.TIM_OCPolarity = TIM_OCPolarity_Low;
TimOCInitStructure.TIM_Pulse = _____; //占空比为 50%
TimOCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OC1Init(TIM1, &TimOCInitStructure);

/* 使能 TIM1 计数器 */
_____

/* 使能 TIM1 的 PWM 输出 */
_____
```

3. 根据下图所示的硬件原理图以及要实现的功能，把空缺的代码补充完整。



要实现的功能如下：

根据上述原理图配置 STM32F103C6 的 USART1，使用 DMA 实现 USART1 TX 数据的按字节发送，并在中断响应函数中清除 DMA 的状态；设置 USART1 为发送模式，波特率 115200，8 位数据位，1 位停止位，不奇偶校验。

【问题】代码实现如下，把空缺的代码补充完整，每处 1.5 分，答案写在横线上。

```
#define DMA_TX_SIZE          300
uint8_t DMA_TX_BUFF[DMA_TX_SIZE];
int main(void)
{
    ...

    /* 配置 DMA1 Channel4 用于 USART1_TX */
    DMA_DeInit(DMA1_Channel4);
    DMA_InitStructure.DMA_PeripheralBaseAddr = (u32)&USART1->DR;
    DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)DMA_TX_BUFF;
    DMA_InitStructure.DMA_DIR = _____;
    DMA_InitStructure.DMA_BufferSize = 0;
    DMA_InitStructure.DMA_PeripheralInc = _____;
    DMA_InitStructure.DMA_MemoryInc = _____;
    DMA_InitStructure.DMA_PeripheralDataSize = _____;
    DMA_InitStructure.DMA_MemoryDataSize = _____;
    DMA_InitStructure.DMA_Mode = DMA_Mode_Normal;
    DMA_InitStructure.DMA_Priority = DMA_Priority_High;
    DMA_InitStructure.DMA_M2M = _____;
```

```

DMA_Init(DMA1_Channel4, &DMA_InitStructure);

/* 配置 USART1 的中断 */
NVIC_InitStructure.NVIC_IRQChannel = _____;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority=1;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);

/* 配置 USART1 的参数 */
USART_InitStructure.USART_BaudRate = _____;
USART_InitStructure.USART_WordLength = USART_WordLength_8b;
USART_InitStructure.USART_StopBits = _____;
USART_InitStructure.USART_Parity = _____;
USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART_InitStructure.USART_Mode = _____;
USART_Init(USART1, &USART_InitStructure);
USART_ITConfig(USART1, USART_IT_TC, ENABLE);
_____;
USART_Cmd(USART1, ENABLE);

while (1);
}

void USART1_IRQHandler(void)
{
    //TX 发送完成中断
    if(USART_GetITStatus(USART1,USART_IT_TC)!= RESET)
    {
        //关闭 DMA
        _____;
        //清除数据长度
        DMA1_Channel4->CNDTR=0;
        //清除中断标志
        _____;
    }
}
}

```