

思考题__课程提供

第1章绪论思考题解答

CPU和MCU的区别

答：CPU中央处理器，计算机的计算和控制核心；MCU是微控制器，=CPU+外部设备（存储、IO..

什么是嵌入式系统（二选一）

答：1) 具有特定功能的专用计算机系统，对功耗、可靠性、成本等有严格要求。

2) 在一个大型/复杂的机械电子系统中，嵌入其中负责某个特定任务的计算机系统

你接触过哪些嵌入式系统电子产品（3种以上）

答：智能手环/手表，mp3，智能门锁，电子血压计，数字示波器。。

说出3个以上的知名MCU厂商

答：NXP（恩智浦），ST（意法），Microchip（微芯），Renesas（瑞萨），Samsung（三星）

你对ARM的理解

答：1) ARM（AdvancedRISCMachines）英国芯片设计公司；2) ARM-Cortex内核架构（分A/R/M系列）

为什么选择学习ARM-Cortex架构的32位MCU

答：首先，32位是MCU主流；其次，ARM-Cortex是内核架构主流，占市场95%以上。

MCU选型时你会考虑哪些方面

答：主频，引脚，开发资源，成本，功耗

第2章MCU硬件基础

u CPU内部，PC指针和SP指针的作用是什么？

Key：PC-程序计数器，指示CPU下一条要执行指令的地址。SP-堆栈指针，永远指向栈顶，入栈栈顶上浮，出栈栈顶下降。

u CPU内部，PSR中四个主要标志位含义：Z、N、O、C。

Key：Z-Zero 零标志，置位表示运算结果为0。N-Negative 符号标志，置位表示结果为负。O-Overflow溢出标志，置位表示结果超出存储的寄存器数据宽度。C-Carry 进位标志，置位表示运算产生进位(加法)或借位(减法)

u 对堆栈做一个简单描述，比如它的结构、存取方式、作用

Key：堆栈是一段存储空间，入口和出口相同且只有一个，存取方式为后入先出，作用包括存储函数调用、中断时的返回地址，以及保护寄存器上下文。

u 简述ARM Cortex-M架构体系中的寄存器组，比如有哪些类型、作用

Key：有通用寄存器，如R0~R12，用于存储操作数和运算结果；特殊寄存器，如PC-程序计数器、SP-堆栈指针、PSR-程序状态寄存器，作用见第1,2题。

u 能否解释什么是中断、中断服务子程ISR、中断向量

Key：中断是CPU处理突发事件的一种机制。ISR是CPU具体处理中断事件的程序。中断向量是ISR程序的入口地址，CPU凭此地址找到ISR的存放位置并执行。

u MCU复位后首先执行main()函数吗？为什么？

Key：不是。因为先要执行一段汇编启动代码，包括初始化堆栈以及PC指针，而后引导至main()函数执行。

u Flash存储器和SRAM存储器的区别，如易失性、用途

Key：Flash-非易失存储器，用于存储程序和固定数据，如表单。SRAM-易失存储器，用于存储临时数据和运行中的程序。

u 什么是总线？哈佛总线和冯·诺依曼总线的区别及优点

Key：总线是MCU内部一组公共的信号线，包括控制总线CB、数据总线DB和地址总线AB。哈佛总线的数据总线和指令总线是独立的，取数据和取指令可以同步进行，执行效率高。冯·诺依曼总线数据、指令的传输都要通过DB总线，因此取指和取数据要分时进行。

u 为何对MCU外设作地址映射？外设映射的起始地址是多少？

Key：外设地址映射就是把外设寄存器的地址映射到存储空间上，统一编址，这样就可以像访问存储器一样，通过地址来访问到外部设备。起始地址是0x4000 0000。

u MCU编程模型中两大重要方面是什么？

Key：一个是寄存器，一个是指令集。在STM32开发中，特指外设的寄存器和C语言的指令

第3章 STM32开发初步

u STM32每组端口共多少个引脚？GPIO引脚的命名规则，如PA0三个字符各代表什么？

答：每组16个引脚，Px0~Px15。命名规则：字母P代表Port，x-代表哪一组 如A组，数字代表哪一个引脚。

u IO引脚的控制逻辑中通常有哪三类寄存器？大致功能如何

答：控制寄存器CR，选择复用功能还是普通IO功能，普通IO是输出还是输入模式。输出数据寄存器ODR，写入1/0后可在引脚上产生高/低电平。输入数据寄存器IDR，读取该寄存器就存取会引发读取引脚状态的操作。

u 解释什么是推挽输出，解释为何采用IO输出低电平点亮LED

答：推挽输出就是IO引脚端采用PMOS和NMOS上下对称连接，两只管轮流导通从而在引脚上输出高或低电平，具有响应速度快、驱动能力强的优点。IO输出低电平时，电流流入引脚，称为灌电流，反之为拉电流，通常灌电流能力强于拉电流，故采用输出低电平点亮LED。（注：LED要外接限流电阻至Vcc）

u GPIOA被映射到存储空间的哪个地址段？

答：0x4002 0000 ~ 0x4002 03FF（课件P31页）

u 什么是CMSIS？它的两个核心层是什么？

答：CMSIS是Cortex架构微控制软件接口标准之意，向下定义了内核和外设寄存器，向上为用户提供调用接口。两个核心层分别是内核访问层和外设访问层。

u 什么是STM32的标准函数库？其文件命名规则，如STM32Fxxx_ppp.c，xxx和ppp代表什么？

答：STM32标准函数库指片上外设的驱动库，由函数、数据结构和宏定义构成，涉及了外设的所有功能。命名规则：xxx表示芯片系列，如F10x、F4xx，ppp表示具体外设的缩写，如GPIO、TIM等。

u 解释寄存器操作语句“GPIOC->MODER = ...”的机理

答：首先通过宏定义，GPIOC定义为C端口在存储空间的起始地址(0x4002 8000)，其次，MODER是GPIOx端口结构体中的第一个成员变量，其占据的正好是x端口MODER寄存器的地址位置。

u int16_t和uint16_t分别代表什么数据类型？表示的数据范围各是多少？

答：int16_t表示16位有符号数，范围-32768~+32767；uint16_t表示16位无符号数，范围0~65535。

u 头文件和源文件的用途

答：头文件里主要包括宏定义(如端口和寄存器)、数据结构(如结构体、枚举)和函数声明。源文件里主要是实现函数的程序代码。

u 结构体和枚举类型的区别？在GPIO初始化中有什么应用（可举例）？

答：结构体是将归属于同一对象的属性进行封装，如在GPIO初始化中，将工作模式、速度、输出类型等参数进行了结构体封装。枚举则是数据取值多选一，在GPIO初始化中，对于工作模式，有IN、OUT、AF等模式，只能是多中选一，故封装到一个枚举类型中。

u 库函数编程和寄存器编程的优劣比较

答：寄存器编程简洁、效率高，但是抽象，且寄存器数量极其庞大。库函数清晰、直观反映其功能，但存在冗余，有时一条寄存器操作语句需多条库函数语句来对应。

u 什么情况需要用户再建立设备库文件（标准库之上）？

答：一是用户加入的外部设备，如按键、显示屏、传感器；二是高于标准库之上的一些操作，如中断配置、定时器配置。

第4章 中断

中断请求和事件请求有何区别

答：中断触发后必须有中断处理子程序，事件是没有中断处理子程序的，事件发生后，会直接发出一个脉冲信号给与之联动的操作。

解释什么是NVIC和EXTI

答：NVIC是嵌套中断向量控制器，管理所有中断源。EXTI是外部中断控制器，主要负责管理GPIO引脚

上发生的中断请求

·外中断请求线和中断请求通道的区别及映射关系（见PPT32页图）

答：外中断请求线是GPIO引脚向EXTI发出中断走的线路，中断请求通道是EXTI向NVIC发出中断走的通道。映射关系为：外中断线0~4#与中断通道的0~4#是一对一映射，外中断线5~9#统一映射为一条中断通道，10~15#统一映射为另一条中断通道。

中断嵌套的规则是怎样的

答：高级别中断可以中断低级别，反之不行，同级别中断发生时按先后顺序响应。

描述一下通用定时器的时基单元构成及各部分功能

答：预分频单元，对CKPSC进行分频得到计数脉冲CKCNT：计数器单元，对CK_CNT计数，达到预

设上限值时产生中断，并重新计数；自动重装寄存器ARR，存放预设的计数上限值。

若APB1总线频率20MHz，TIM3的预分频为4000，计数上限为5000，列式计算定时时长

答： $1 / (40 \times 10^6 \div 4000) \times 5000 = 0.5$ 秒

定时器中断编程分哪三步走

答：初始化NVIC，初始化定时器TIMx，编写中断服务子程ISR

中断服务子程（ISR）最后一条语句为什么要清中断标志

答：如果不清除，就会出现执行完ISR后马上再次发生中断的情况。

第5章通信I_UART串口

u 解释UART通信方式为什么是“串行，异步，点对点，全双工，对等”

答：串行，收发双方通过一根线传输(Tx→Rx)；异步，收发双方没有共同时钟，用各自独立时钟控制；点对点，收和发各只有固定的一方，不需要地址确认；全双工，收发双方各有一对Tx和Rx信号端（两条信道），因此收发可同时进行；对等网络，收发双方任何一方都可以发起通信，在网络中是平等的。

u 解释UART通信的数据帧格式(Frame format)

答：包括1bit的起始位（低电平），8bit的数据位，LSB在前、MSB在后，1bit的校验位，1bit的停止位（高电平）。

u UART术语：115200，8N1 是何意？每秒可完成多少字节数据量的传输？

答：波特率115200bps，8bit数据位，N-无校验位，1bit停止位。

u 解释UART过采样(Oversampling)，包括原因及方法。

答：为了避免电磁干扰产生的信号毛刺和线缆寄生电容造成的信号边沿平缓，导致接收端数据采集错误，接收端以16倍波特率进行采样，这称为过采样。取中间三个采样值，并以“投票”的方式最终确定采样的信号值。

u 何为偶校验？数据“10010111”做偶校验，校验位的值是多少？

答：保证数据中（含校验位）1的个数是偶数，这称为偶校验。“10010111”中有奇数个1，所以校验位需要是1，以保证1的总个数为偶数。

u UART简化的三线制连接是哪三条线？各自用途是什么？

答：Rx—接收端，Tx—发送端，GND—接地端。

u 如何使MCU的UART和你笔记本电脑的USB接口进行通信？

答：需要一个转换器，内置USB转UART的芯片（如CH340），并在电脑端安装驱动程序，从而将USB接口虚拟为一个UART接口。

u STM32F407IG的UART初始化时，为什么要加一条清除TC标志的指令？(TC – Transmission complete)

答：UART发送一个数据帧时，要检测TC标志，TC=1表示发送完成，才可进行下一帧的发送。但TC在系统复位时即被置1，这显然是不合理的，所以在UART初始化时，要将其清0，以免影响UART的发送功能。

第5章通信II_SPI接口 jXG

▮ 解释SPI通信方式为什么是“同步，串行，全双工，主从”模式

答：同步，通信有一个时钟信号来协同收发双方；串行，数据是一个比特一个比特传输的；主从，收发双方地位不对等，一个主机、一个从机；全双工，主机通过数据输出端向从机发数据，同时用数据输入端从从机收数据。

▮ 解释SPI四根信号线的用途

答：MOSI，作为主机时是输出端，作为从机时是输入端；MISO，作为主机时是输入端，作为从机时是输出端；SCK，时钟信号，同步主从双方的数据传输；NSS，主从选择，接高电平时设备身份为主机，接低电平时设备为从机。

▮ SPI通信两种时钟相位(CPHA)的区别，两种时钟极性(CPOL)的区别

答：时钟相位决定通信开始的方式，CPHA=0时，通信从拉低片选信号开始，然后在时钟的奇数沿采样输入端的数据，偶数沿发送端发一下个bit；CPHA=1时，通信完全由时钟控制发起，在时钟的奇数沿输出端发出数据，偶数沿采样输入端的数据。时钟极性决定了时钟起始电平，若CPOL=0，则起始电平为低电平，此时奇数沿是上升沿；若CPOL=1，则起始电平为高电平，此时奇数沿是下降沿。

▮ STM32F407IG的SPI2接口最大和最小传输波特率是多少？计算原理。

答：SPI2挂在APB1总线，总线频率最高为42MHz，SPI的波特率分频值为 2^n （ $n=1\sim 8$ ），由此，最大波特率 $=42M/2=21Mbps$ ，最小波特率为 $42M/256\approx 164Kbps$ 。

□ 如果你买的一块新开发板上有一个SPI器件时，想把它用起来，需要怎么做？（比如怎么知道器件的引脚连接、器件通信所采用的时钟相位和极性、以及该器件允许的最大传输波特率）

答：首先，查看商家提供的开发板的电路原理图，查看芯片引脚连接情况，包括SPI通信脚和其他脚如复位；其次，查看芯片的数据手册(DataSheet)，找到芯片的通信时序图，观察对比确定SPI的时钟相位和极性，以及字节序等；再找到电气特性表，确定通信时钟的最小周期/最大频率指标；最后，编程初始化上述SPI通信参数，并初始化该芯片(按手册或例程)。

□ 描述一下规格 16×16 的“燕”字，在OLED/LCD屏上的显示过程（图在课件39页）

答：“燕”的编码共占32个字节，分两个Page写入，每个Page刷16列，Page0显示“燕”的上半部，写入前16个字节，Page1显示“燕”的下半部，写入后16个字节。

□ 若想竖排显示“燕山大学”，该如何修改下面这条横排显示的语句？（图在课件40页）
`for(i = 0, i < 4, i++) { OledShowFont16X16(16 * i, 0, Fontysu+32 * i); }`

答：`for(i = 0, i < 4, i++) { OledShowFont16X16(0, 16 * i, Fontysu+32 * i); }`

第5章通信III_I2C总线

u I2C通信方式为什么是“半双工”

答：因为通信只有一条数据线SDA，同一时间只能单向传输数据，从主机到从机或者从从机到主机，因此是半双工。

u I2C两根信号线的名称及用途

答：SCL，时钟线，用于同步主机和从机的通信；SDA，数据线，传输串行数据。

u 推挽输出和OD输出结构上的区别？OD门线与连接原理(为何通过一根导线就可实现与逻辑)

答：推挽结构有两个MOS管，上管导通输出高电平，下管导通输出低电平；OD结构只有一个MOS管，导通时输出低电平，截止时必须通过外部上拉电阻至高电压，才能输出逻辑1。线与连接指用一根总线将多个OD门输出端并联起来；所有OD门全输出1时，总线上才为1，只要有一个OD门输出0，就会将总线电压拉低，输出逻辑0，由于有上拉(限流)电阻的作用，不会出现短路情况。

u I2C的信号电平规范（三点）

答：1) SCL高电平时，SDA保持稳定；2) SCL低电平时，SDA发生变化；3) 通信开始或结束时，SDA在SCL为高时变化。

u 描述I2C主机**写**从机寄存器的数据帧格式（课件20页）

答：主机，发起起始位、从机地址+写控位；从机，应答A；主机，写寄存器地址；从机，应答A；主机，写寄存器数据；从机，应答或非应答；主机，结束通信。

u 描述I2C主机**读**从机寄存器的数据帧格式（课件20页）

答：主机，发起起始位、从机地址+写控位；从机，应答A；主机，写寄存器地址；从机，应答A；主机，发重复起始位、从机地址+读控位；从机，应答、发出寄存器中数据；主机，非应答、结束通信。

u MPU6050的为什么有两个7bit地址？分别是什么？

答：查阅数据手册，MPU6050地址为b110100X，末位X由引脚AD0决定；AD0接地时，X=0，地址为b1101000，AD0接Vcc时，X=1，地址为b1101001。

u 若MPU6050初始化配置为：不自检，量程±4g，高通滤波5Hz，则语句“MPU6050_WriteReg(0x1C, 0x01);”该如何修改？（课件44页）

答：改为 MPU6050_WriteReg(0x1C, 0x09)

u 如果想读取MPU6050三个轴的角速度数据，则语句“for (i = 0; i < 6; i++) {

readbuff[i] = MPU6050_ReadReg (0x3B + i);} ”该如何修改？（课件45页结合MPU6050寄存器手册）

答：查阅寄存器手册，知陀螺仪数据寄存器地址为0x43~0x48，所以改为 for (i = 0; i < 6; i++) { readbuff[i] = MPU6050_ReadReg (0x43 + i); }。

第六章嵌入式系统

u 带嵌入式操作系统的编程有什么优势？

答：一是可以显著降低开发难度；二是让每个任务都认为自己独占CPU，方便代码编写；三是增加代码移植性。

u 什么是线程？

答：线程就是完成特定功能的函数。不用于裸机系统中的函数，OS中的线程需要进行初始化配置，就绪后加入线程队列等待CPU调度运行。

u 线程的状态有哪些？如何相互转换？

答：包括初始化、就绪、运行、挂起、关闭共5种状态。转换需要调用相关函数，调用 `rt_thread_init()` 来初始化一个线程，调用 `rt_thread_startup()` 来就绪一个线程，就绪到运行状态不需要调用函数，就绪后自动排入线程队列等待CPU调度运行，调用 `rt_thread_delay()` 进入挂起状态，运行中调用 `rt_thread_exit()` 来关闭一个线程。

u 简要解释线程通信方式中的信号量、事件和消息

答：信号量等同于打开临界区的一把钥匙，只有拿到钥匙的线程才能访问临界区(公共数据交换区)。事件机制等同于标志位，一个线程执行时设置标志，另一个线程查看标志并依据置位情况运行。消息就是直接发生通信的数据，发送线程将数据放入消息池中，接收线程从消息池取出数据。