

嵌入式系统设计工程技术人员认证考试

(初级) 考试大纲

考试目的

面向电子信息类、计算机类、电气类、自动化类、物联网、人工智能、机器人及其他相关专业的本科或高职在校、毕业学生，以及希望从事相关行业的技术人员进行能力认证，测查应试人员对微控制器基本结构和工作原理的理解，以及嵌入式软件和硬件开发能力。

考核方式

嵌入式系统设计工程技术人员认证考试（初级）将从理论知识和实践能力两方面对专业技术水平进行考核，理论知识考核包括客观题（100分）及主观题（50分）两部分，共计150分，客观题和主观题分数合计达到90分（含90分）以上者为合格；实践能力考核为150分，实践题分数达到90分（含90分）以上者为合格。考核合格者将获得《嵌入式系统应用与开发》初级技术等级证书。

理论知识客观题（判断、单选、多选）考核采用机考方式进行，主要考查嵌入式系统设计工程技术人员从事本职业应掌握的基本知识和专业知识；理论知识主观题（填空、简答、计算）考核采用笔试或机考方式考核运用嵌入式系统的综合能力；实践能力考核采用线下实操考核方式进行，主要考查嵌入式系统设计工程技术人员的嵌入式编程及系统调试等从事本职业应具备的实际工作能力。

考试内容与要求

理论知识部分

第一部分 微控制器基本结构和工作原理

1. 掌握微处理器的组成和微处理器系统的结构；掌握系统总线和存储系统的构成及操作过程；理解输入输出系统的基本构成；掌握中断与异常处理的基本原理；了解系统运行的常规操作。

2. 掌握 Cortex M4 处理器内核中的寄存器组作用、流水线工作原理、工作模式切换和堆栈操作。

3. 了解存储器分类与作用；理解存储系统的数据组织；掌握数据访问的具体操作过程。

4. 了解时钟系统基本结构、系统启动及复位机制；理解时钟系统的配置和 RCC 的基本概念和功能。

5. 掌握 DMA 基本概念和原理。理解 DMA 访问机制与中断访问机制各自特点。合理使用 DMA 部件完成数据传输工作。

6. 理解 NVIC 的作用；了解可屏蔽、非可屏蔽中断概念；理解外部中断系统工作原理和中断优先级的概念。掌握中断响应过程及其中断嵌套原则，能够根据项目需求合理配置中断响应顺序。掌握外部中断触发机制和应用编程。

第二部分 嵌入式系统硬件开发技能

1. 了解 GPIO 电路基本结构；掌握 GPIO 设备寄存器的操作基本概念。能够根据应用需求配置 GPIO 相关寄存器；熟练使用 GPIO 驱动外部器件工作。

2. 了解管脚复用基本概念。

3. 了解定时器的基本结构和功能；掌握计数器的的工作模式、输入捕获及 PWM 输出工作原理。能够熟练配置定时器的定时模式、计数模式、捕获模式和比较模式，完成应用程序设计。

4. 了解同步通信与异步通信基本概念；理解 UART 的结构与工作原理；熟悉 UART 通信数据格式、波特率概念及其计算方法。掌握 UART 的通信参数和操作流程；掌握轮询、中断和 DMA 方式进行 UART 通信的操作流程。

5. 了解 SPI 同步通信的基本概念和操作原理；理解 STM32 的 SPI 结构；掌握 SPI 总线扩展外部器件的接口方式，掌握 SPI 总线的配置流程及应用方法。

6. 了解 ADC 概念和 STM32 的 ADC 结构；理解 ADC 的操作参数和流程；掌握 ADC 的应用。

7. 了解 I2C 通信的基本概念及操作原理。

8. 了解看门狗的基本概念、作用和应用。

9. 掌握基本原器件的作用及简单应用。初步掌握常用阻抗元件主要工作参数含义和选型。能够阅读常用器件数据手册，并根据相关参数正确设计电路。熟悉常用器件封装，能够根据需要选择合适的封装设计印刷电路板（PCB）。

10. 掌握最小系统的基本概念和常用的外围电路设计。能够根据最小系统要求，选择元器件，完成电路设计。

11. 能够阅读嵌入式系统电路图，能够分析常见电路模块工作原理和参数计算。熟悉常见电路符合、标识和电路连接方法。

12. 初步掌握 PCB 设计的基本原则，包括元器件布局策略、线宽、线距及过孔、焊盘尺寸选择；熟悉 PCB 设计流程；能够使用现成的 PCB 工程查阅线路连接及其位置。

13. 了解 EMC 的基本概念和措施。初步了解电路设计中的 EMC 常用措施和器件。

14. 了解如何利用数据手册查阅器件参数；通过参考手册和应用笔记了解器件工作原理，完成程序设计。

15. 了解芯片选型的基本方法。

第三部分 嵌入式系统软件开发技能

1. 掌握数据基本类型、运算符、数组、指针和结构体的应用；熟练使用数组、指针和结构体等完成数据处理；熟练应用程序基本结构和函数进行程序设计；了解标识符、关键字、预处理及常用的 C 语言基本知识。初步掌握程序模块化设计和可移植性设计方法。

2. 掌握数据传输、变量定义和跳转等语句的操作原理；熟悉堆栈数据结构和操作原理。掌握 Cortex M4 中断控制原理及操作语句；掌握 C 语言处理中断方式，初步了解混合编程相关概念。

3. 熟悉并掌握 HAL/LL 库中 C 语言相关知识点的运用，能够读懂相关代码。

第四部分 嵌入式实时操作系统

1. 了解 RTOS 的背景、发展、适用场景、与裸机的区别；了解 RTOS 分类；了解 RT-Thread 的特点、版本。

2. 开发环境：了解开发环境，例如 Keil、IAR、GCC、Studio；掌握源码下载方法；掌握使用 simulator 学习内核部分，无需开发板，进行日志打印。

3. 基础知识：掌握实时内核的实现；掌握系统的入口函数和执行的顺序；了解通过宏定义的方式进行申明后在系统启动过程中被执行；理解采用内核对象管理系统来访问 / 管理所有内核对象；掌握如何裁剪内核/裁剪组件。

4. 线程：掌握基于多任务设计的方法；掌握线程控制块的成员组成，线程控制块是系统内核用来控制线程的一个数据结构，存放线程信息，如优先级、线程名称、限制状态等，也包含了线程间连接所使用的链表结构；掌握线程的属性，如栈和状态、优先级和时间片的作用。

5. 线程调度：掌握线程的工作机制，决定任务顺序 和 进行任务切换；掌握调度规则：优先级抢占、时间片轮转；了解调度接口 API。

6. 线程 API：掌握静态线程的 API: init 和 detach；掌握动态线程的 API: create 和 delete；掌握线程启动过程和线程状态切换过程，线程使用时必须让出 CPU 使用权。

7. 线程应用示例：掌握线程的初始化和创建；掌握多线程间的优先级抢占和时间片轮转；线程其它的应用：让出、抢占、恢复和挂起；了解空闲线程必要性和优先级及空闲线程的钩子函数。

8. 系统节拍：理解操作系统中最小的时间单位是时钟节拍和系统时钟的定义与实现；理解硬件定时器 OS tick 的实现；理解如何获取时钟节拍和使用时钟节拍。

9. 定时器管理：了解定时器的定义和分类；掌握定时器工作原理：rt_tick 和定时器链表；掌握定时器控制块概念，定时器控制块继承关系，定时器控制块定义及结构体实现；掌握定时器的基本操作：创建、启动、运行和删除；掌握如何动态创建定时器、静态创建定时器、定时器控制接口和合理使用定时器，以及一些注意事项。

10. 线程同步：了解临界区的概念；了解如何使用中断锁和临界区的方式对临界区进行保护；理解如何进行多线程协调访问内存。

11. 信号量：了解信号量的概念、分类和选择；掌握二值信号量的值与状态及计数信号量值与状态；掌握管理信号量数据结构的成员含义；掌握如何创建、删除、初始化和脱离信号量；掌握信号量在创建成功后进行获取与释放；了解利用信号量进行生产和消费的方法；了解信号量用于锁、同步、资源计数。

12. 互斥量：了解相互排斥的信号量；掌握二值信号量的开锁或闭锁；掌握互斥量的数据结构成员和含义；掌握互斥量的创建与删除、初始化与脱离；掌握互斥量获取与释放；掌握互斥锁、理解优先级反转、优先级继承。

13. 事件集：了解事件集概念，完成一对多，多对多的线程间同步；掌握事件采用与/或触发方式，从而触发事件发生；掌握事件集控制块数据结构的成员含义；掌握事件集的创建、发送、接收、删除；掌握事件集的发送和接收 API；掌握事件使用方法：线程 1 发送事件、线程 2 接收事件，协调运行；了解事件集的使用场合。

14. 邮箱：理解邮箱是线程间通信方法；理解邮箱时用于线程间通信，可交换消息；理解邮箱控制块数据结构的成员含义；掌握邮箱的创建与删除，初始化与脱离；掌握邮箱发送和接收；理解邮箱的使用方法：线程 1 邮箱中发送邮件，线程 2 从邮箱中收取邮件。

15. 消息队列：理解消息队列概念多条消息排成的队列形式和消息队列特性；在线程与线程间，中断与线程间传递信息；理解消息队列的数据结构，包含消息队列名称，内存缓冲区，消息大小及队列长度等消息队列的属性；理解消息队列的创建与删除，初始化与脱离；理解消息队列发送接收 API；理解消息队列的使用：线程 1 会从消息队列中收取消息；线程 2 定时给消息队列发送普通消息和紧急消息；理解发送消息和同步消息的不同。

16. 信号：理解信号的概念，是软中断，用作异常通知、应急处理；理解信号的工作机制，用于异步通信；理解信号的接口的使用；掌握如何在系统中安装信号；掌握如何发送和等待信号；掌握安装信号并解除阻塞，然后发送信号给线程。

17. 内存管理：了解软件运行时对计算机内存资源的分配和使用的技术；掌握内部存储空间和外部存储空间的分配；掌握静态内存池动态内存堆管理；掌握内存堆管理方法，用于管理一段连续的内存空间；掌握分配大量大小相同的小内存块加快内存分配与释放的速度；理解 memheap 堆初始化；分配内存设置钩子；内存堆管理 API；动态申请内存并释放；理解内存池操作 API；了解创建内存池；获取和释放内存块。

18. 理解代码规范：1) 名称（目录名、文件名、函数名）规范；2) 注释（头文件、代码）规范；3) 定义（头文件、结构体、宏）规范；4) 函数（函数名、声明、初始

化)规范;5)格式(缩进、括号、空格)规范;6).开发安全、可靠稳定代码:声明和初始化、表达式、整数、数组、字符和字符串、内存管理、其它。

第五部分 嵌入式通信

1. 了解蜂窝通信的发展趋势;了解 CAT1 的基本概念;理解 CAT1 与其他制式网络的技术特点区别;了解 CAT1 的发展前景。

2. 物联网架构:了解物联网总体架构,即端-管-云-应用终端;理解蜂窝通信模块在嵌入式通信系统中的架构位置;了解常见应用场景,即精准定位、联动呼叫、智能测亩、自动计产等。

3. 模块硬件框架:了解蜂窝通信模块硬件各部分的作用;了解通信模块常见的封装:LCC、LGA、LCC+LGA 封装、MiniPCIe 封装。

4. 控制接口:了解蜂窝通信模块的 5 类常见控制接口作用。

5. 通讯接口:了解蜂窝通信模块的通讯接口 USB 和 UART 概念;了解 USB 协议演进;了解 USB 和 UART 接口物理形态;理解 USB 和 UART 电路设计要点和布线设计要点。

6. SIM 卡接口:了解 SIM 卡概念及主要功能;了解 SIM 卡接口的标准引脚定义;理解 SIM 卡接口电路设计要点和布线设计要点;了解 SIM 卡类别:Mini SIM、Micro、Nano、E-SIM;了解 SIM 卡座管脚定义。

7. 射频接口:了解蜂窝通信模块的射频概念;了解蜂窝通信模块的天线分类;了解蜂窝通信模块的频段;了解射频的主要器件及结构;理解射频主要指标(频谱特性之线性度、误差向量幅度 EVM、最大吞吐量、射频接口的接收灵敏度);理解射频指标的测量方法;理解射频接口的电路设计要点和布线设计要点。

8. 通讯协议:了解蜂窝通信模块与微处理器之间的通讯协议;了解通信协议的概念及分类;了解特定通信数据;掌握 AT 命令的概念、分类、语法、格式。

9. AT 命令:理解 AT 命令的注意事项(同步、异步处理);掌握 AT 命令应用。

10. 内置协议栈:了解内置协议栈的概念、特点;理解内置协议栈的常见功能;了解内置协议栈的适用场景;理解内置协议栈的应用实例。

11. TCP:了解 TCP 协议功能和 TCP 头格式;理解 TCP 可靠传输技术,包含建立连接、传输确认、流控、错误重传、断开连接;理解 TCP 的适用场景;理解如何获取 TCP 的 Wireshark log。;

12. MQTT:了解 MQTT 概念、特性;理解 MQTT 的协议远离;了解 MQTT 协议数据包格式;掌握标准 MQTT 和专用 MQTT 的 AT 指令实现。

13. Windows 驱动集成:了解 Windows 驱动概念;了解 Windows 驱动安装与卸载;了解 Windows 驱动主要文件。

14. Windows 数据拨号:了解 PPP 拨号概念和协议原理;了解 RNDIS 拨号概念;理解 RNDIS 拨号应用 AT 命令;了解 PPP 拨号和 RNDIS 拨号的差别。

实践能力部分

第一部分 微控制器基本结构和工作原理

1. 能够利用 CubeMX 软件完成时钟的配置。
2. DMA：掌握 DMA 的相关原理与配置方法；熟练编写、调试 DMA 的 C 语言程序；掌握 DMA 与其他资源综合应用的原理与方法；能够利用 CubeMX 软件完成 DMA 的配置。
3. 中断：掌握中断的相关原理与配置方法；熟练编写、调试中断的 C 语言程序；掌握中断与其他资源综合应用的原理与方法；能够利用 CubeMX 软件完成中断的配置行政法基础知识。

第二部分 嵌入式系统硬件开发技能

4. GPIO：掌握端口的相关原理与配置方法；熟练编写、调试端口的 C 语言程序；能够利用 CubeMX 软件完成 GPIO 的相关配置。
5. 定时器：掌握定时器的相关原理；掌握定时器的工作模式、参数计算等；熟练编写、调试定时器的 C 语言程序；掌握定时器与其他资源综合应用的原理与方法；能够利用定时器实现基本定时、外部脉冲计数、PWM 信号输出以及外部信号捕获等应用；能够利用 CubeMX 软件完成定时器相关功能的配置。
6. UART：掌握串口的相关原理；掌握串口的工作模式、参数计算等；熟练编写、调试串口的 C 语言程序；掌握串口与其他资源综合应用的原理与方法；掌握相关串口外部驱动电路的原理并编写基于串口的外设驱动；能够使用轮询、中断和 DMA 三种方式完成串口通信；能够设计简单的通信协议，并基于该协议实现串口通信；能够利用 CubeMX 软件完成串口的配置。
7. SPI：掌握 SPI 的相关原理；掌握 SPI 的工作模式、参数计算等；熟练编写、调试 SPI 的 C 语言程序；掌握 SPI 与其他资源综合应用的原理与方法；掌握相关 SPI 外部驱动电路的原理并编写基于 SPI 接口的外设驱动；能够利用 CubeMX 软件完成 SPI 的配置。
8. ADC：掌握 ADC 的相关原理；掌握 ADC 的工作模式、参数计算等；熟练编写、调试 ADC 的 C 语言程序；掌握 ADC 与其他资源综合应用的原理与方法；掌握相关 ADC 外部驱动电路的原理与驱动程序设计；能够利用 CubeMX 软件完成 A/D 的配置。
9. I2C：能掌握 I2C 的相关原理；能掌握 I2C 的工作模式、参数计算等；能熟练编写、调试 I2C 的 C 语言程序；能掌握 I2C 与其他资源综合应用的原理与方法；能掌握相关 I2C 外部驱动电路的原理并编写基于 I2C 接口的外设驱动；能够利用 CubeMX 软件完成 I2C 的配置。
10. 熟练掌握 STM32CubeMX 的使用
11. 掌握常用电子元器件的原理与电路设计；能够编写、调试、测试常用外设电路的驱动程序。
12. 掌握最小系统的原理与电路设计；掌握最小系统的 PCB 设计与注意事项；能够编写、调试、测试最小系统的程序。

13. 掌握嵌入式系统的电路原理图的绘制与设计原理。

第三部分 嵌入式系统硬件开发技能

1. 对 C 语言的语法、语义有较好的理解；能熟练的阅读 C 源程序，并具有分析程序的能力；能熟练编写、调试 C 语言程序。

2. 掌握 C 语言程序设计的方法和技巧，能用 C 语言编写良好的面向过程的程序；掌握程序的调试与测试方法；能够按照统一的命名规则对变量及函数命名，做到“见名知义”；关键代码需要相关的注释，函数及变量定义添加必要的说明。

3. 掌握基于 HAL/ LL 库的程序设计方法；熟悉基于 HAL/ LL 库的程序的调试与测试方法；基于 HAL/ LL 库设计良好的 C 语言程序。

第四部分 嵌入式实时操作系统

1. 掌握线程的初始化、创建、启动，实现外设操作。
2. 创建定时器，启动定时器，编写定时器超时函数。
3. 掌握信号量在线程间同步的方法，编写代码。
4. 掌握互斥量在线程间同步的方法，编写代码。
5. 掌握事件集在线程间同步的方法，编写代码。
6. 掌握邮箱在线程间通信的方法，编写代码。
7. 掌握消息队列在线程间通信的方法，编写代码。

第五部分 嵌入式通信

1. 掌握使用 UART 实现 MCU 与蜂窝通信模块的通信。
2. 掌握 AT 指令熟练运用。
3. 掌握 TCP 协议。
4. 掌握公有云基于 MQTT 连接。