

嵌入式技术栈列表 V2.0

作者：郝翹活

目录

一、基础知识类：	3
二、编程语言类：	4
三、嵌入式设备类：	6
四、常用通信协议类：	7
五、嵌入式开发环境类：	10
六、操作系统类：	11
七、 组件库类（MCU）：	13
八、编程设计模式和函数接口设计：	15
九、常用调试技巧及软件类：	15
十、常用工具软件类：	16
十一、常用电子模块使用类：	16
十二、硬件类：	16
十三、 嵌入式自动化测试：	19
十四、 上位机开发：	19
十五、 学习目标：	20

一、基础知识类：

1.1 数据结构与算法

1. 数据结构

队列，循环队列，链表，堆和栈，原子表，二叉树
C 语言实现方式及常用操作

2. 算法

- (1) 排序算法：快速排序、归并排序、计数排序；
- (2) 搜索算法：回溯、递归、剪枝；
- (3) 图论：最短路径、最小生成树、网络流建模；
- (4) 动态规划：背包问题、最长子序列、计数问题；
- (5) 基础技巧：分治、倍增、二分法、贪心算法。

1.2 计算机原理

1. 数字电路设计基础

包括数字技术的相关概念、数制与编码、逻辑代数基础及逻辑门电路等内容；组合逻辑电路部分，讲述了逻辑电路的描述方法、组合逻辑电路的分析与设计方法和常用的 MSI 组合逻辑器件的原理及其应用；时序逻辑电路部分，讲述了时序逻辑电路中基本的双稳态元件、时序逻辑电路的分析与设计方法和常用的 MSI 时序逻辑器件的原理及其应用；可编程逻辑器件部分，讲述了可编程逻辑的基本概念、简单可编程逻辑、复杂可编程逻辑和硬件描述语言 VHDL 的基础知识内容；数字系统的描述工具、小型控制器的设计、简易计算机的设计、A/D 和 D/A 转换以及 PCM 编码的原理。

2. 计算机组成与原理

了解并掌握计算机的基本组成、计算机的指令系统、计算机的运算器、计算机的控制器、存储器的层次结构、输入输出系统、性能评价、操作系统的基本原理、计算机网络的基本原理等内容。

3. 微机原理与接口技术

了解并掌握计算机基础知识、8086 微处理器、8086 汇编语言程序设计、存储器、微型计算机的输入 / 输出、微机中断系统、可编程接口芯片、A/D 和 D/A 转换器、常用外围设备及接口。

4. ARM 汇编及调试技术

了解并掌握 GNU 汇编语法和常见汇编指令

1.3 芯片数据手册阅读

能理解信号的时序关系，根据数据手册能够编写、调试、测试常用外设电路的驱动程序。

1.4 电子元器件辨识

了解并掌握半导体分立元器件和半导体集成元器件的常见类型、特点及常见参数和区别。

1.5 常用设备使用

了解并掌握直流稳压电源，数字示波器，逻辑分析仪，电烙铁，热风枪，恒温加热台，信号发生器、电参数测量仪，万用表，功率计等常见电子测量设备的使用方法。

二、编程语言类：

2.1 C 语言

1. C99 标准

了解并掌握 C 语言中数据类型，变量结构，数据结构，不定常数组，变长数组 VLA，restrict 关键字，inline 关键字，内联函数，_Bool 类型，可移植类型，stdint.h 和 inttypes.h 等概念和使用方法。

2. C11 标准

了解并掌握 C11 中泛型编程和原子操作方法。

3. 要求

能够做到：

（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）对于 MCU 开发：

①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-M 系列中断控制原理及操作语句；掌握 C 语言处理中断方式，初步了解汇编和 C 混合编程相关概念；

②了解并使用 ARMGCC 编译器的一些特定指令，如__attribute_/GUN C 机制；

③了解并可以使用 GCC 完成对 C/C++程序的编译、链接和调试；

（3）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。

2.2 MicroPython/Python

了解并掌握 Pyhon 语言中变量及类型；标识符与关键字；常用格式化输出符号；运算符（算数运算符、赋值运算符、比较(即关系)运算符、逻辑运算符）；数据类型转换；判断与循环语句；字符串、列表、元祖、字典初始化和常见操作；函数；文件操作；面向对象（概念，命名、分类、属性与方法，私有权限，魔法方法，继承，类属性与实例属性，类方法和静态方法，单例模式）；异常与模块。

2.3 LUA 语言

了解并掌握 Lua 语言中注释用法；变量类型；字符串操作；运算符；条件分支语句；循环语句；Function 函数；多脚本执行；Table 表的遍历、公共方法、特殊用法（字典，类）；元表概念和方法；OOP（封装、继承、多态）；特殊用法；自带库函数；垃圾回收。

2.4 Bat 文件

了解并掌握 Bat 中批处理基本命令操作(基本运算、重定向运算、多命令运算、管道操作运算、批处理文件参数传递定义、时间日期使用、启动命令、调用其它.bat 文件、任务列表查看命令、任务终止命令、文件夹结构查看命令、关机命令、计划任务命令)；文件夹或文件相关命令（目录文件的浏览、新建与删除、切换、重命名、拷贝、剪切）；网络相关命令（用户操作命令、用户组操作命令、主机连通性检测命令、网络路由信息命令）。

2.5 bash shell

了解并掌握 shell 概念；脚本的简单介绍；变量和引用；算术运算；交互式编程；选择判断；条件测试；循环；函数。

2.6 arm 和 thumb-2 汇编

了解并掌握指令集基本指令:跳转指令、数据处理指令、单寄存器加载和存储指令、批量寄存器加载和存储指令、系统指令、条件执行指令、处理器控制指令、协处理器数据处理与寄存器传输指令。

2.7 matlab 语言

1. 编程基础：

（1）数据类型：数值类型;字符与字符串;结构;单元数组;函数句柄;数据类型识别与转换；

（2）运算符与运算：算术运算符;关系运算符;逻辑运算符；

（3）编程基础：变量；语句；程序控制；M 文件与脚本；

（4）函数：M 文件函数；匿名函数；子函数；程序调试。

2. 常用工具箱：控制系统工具箱、通讯工具箱、图象处理工具箱、线性矩阵不等式工具箱、神经网络工具箱、信号处理工具箱、统计工具箱、滤波器设计工具箱。

三、嵌入式设备类：

3.1 ARM-MCU

1. Cortex-m 内核、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) 了解 AMBA 总线模型、中断机制和中断向量控制器、内核总线矩阵中数据传输、mpu 内存保护单元、寄存器操作和汇编操作。

(3) 工作原理：

①掌握 Cortex-M 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；了解存储器分类与作用；

②理解存储系统的数据组织；掌握数据访问的具体操作过程；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；

③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；

④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。

2. 外设的概念和基本使用

了解并掌握 GPIO 使用（工作模式，通信速率，外部中断），ADC 模数转换，定时器，dma 数据传输 通信协议（IIC,IIC,SPI,串行总线（rs485，rs232，rs422，全双工，半双工，单工，智能卡，can 总线、IIS），单总线通信，usb（usb2.0，usb3.0）、ETH 以太网）。

3. Flash 编程和内核编程

(1) Flash 编程：

①了解主存储器、信息块和闪存存储器接口寄存器的工作结构；

②了解并会使用 FLASH 闪存的读写和擦除(页擦除、全片擦除)操作。

(2) 内核编程：

对于 CM 内核中寄存器组、操作模式、存储器映射和内核中中断、Systick 等做到基本了解即可。

3.2 RISC-V MCU

1.内核模型、基本结构和工作原理

对 RISC-V 架构中的指令集体系结构、特权架构、通用寄存器、CSR 寄存器等做到基本了解即可。

2.外设

GPIO 使用（工作模式，通信速率，外部中断），ADC 模数转换，定时器，dma 数据传输 通信协议（IIC,IIC,SPI,串行总线（rs485, rs232, rs422, 全双工，半双工，单工，智能卡，can 总线、IIS），单总线通信，usb（usb2.0, usb3.0）、ETH 以太网）。

3.flash 编程和内核编程

3.3 ARM-MPU

能够做到 uboot 与 linux 内核的裁剪和移植、Linux 基本使用、相关库的开发、与内部 CM 核芯片进行通信和控制。

3.4 DSP 数字信号处理器

DSP 芯片基本结构、CCS 开发环境使用、寄存器操作和外设使用 (GPIO\ADC\SRAM\FLASH\外部中断\SPI\ECAN\DMA\XINTF\FFT)、simulink 软件仿真。

四、常用通信协议类：

4.1 通用网络协议

1.OSI 模型

了解 OSI 模型基本内容、各层功能及数据传输过程。

2.tcp/ip 协议簇：Tcp、udp

- （1）了解 TCP/IP 协议网络分层模型和数据传输过程；
- （2）了解 IP 协议主要功能及其规范，对 IP 寻址、路由（最终节点为止的转发）以及 IP 分包与组包三个部分有基本了解，对路由控制、Hop、路由控制表、数据链路的抽象化、IP 多播等基本概念和 workflows 做到基本理解；
- （3）对传输层的 TCP 与 UDP 协议区别和关系、各自的工作流程、特点和应用场景做到基本了解；
- （4）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。

3.http 和 https 超文本传输协议

- （1）对 HTTP 基本概念、消息结构、常见请求方法、响应头信息、状态码、content-type、MIME 类型做到基本了解；
- （2）对 SSL/TLS 协议做到基本了解，对 HTTPS 传输数据的流程和 HTTPS、HTTP 区别和特点做到基本了解。

4.ftp/tftp 文本传输协议

- （1）对 FTP 的模块架构、数据连接建立方式（主动模式、被动模式）、命令和应答、数据传输模式（二级制和 ASCII）做到基本了解；
- （2）对 TFTP 协议特点、五种报文、数据包格式、数据传输流程、流量控制、差错控制等做到基本了解。

5.ssh 协议

对 SSH 协议基本概念、登录原理、基本用法、远程登录、端口转发、远程操作、本地转发、远程转发、动态转发做到基本了解并能够应用。

4.2 无线通信协议

0.AT 指令

理解 AT 命令的注意事项(同步、异步处理)；掌握 AT 命令应用。

1.mqtt 协议-应用层

(1) 了解 MQTT 中发布/订阅消息模式、QoS 服务质量、设计规范、网络传输与应用消息、协议实现方式、订阅、主题、会话等概念和 MQTT 特性；了解 MQTT 协议数据包格式；

(2) 掌握标准 MQTT 和专用 MQTT 的 AT 指令实现；能够做到在 Windows 平台上搭建一个简单的 MQTT 应用；

(3) 了解并掌握 MQTT 协议 3.1.1。

2.zigbee 通信协议-物理层

(1) 对 zigbee 协议基本架构、应用场景、特点、网络拓扑结构、节点实现、通信全过程及信息传递流程做到基本了解；

(2) 了解 XBee、CC2530、ETRX2、JN516x、NRF52840 等常见 zigbee 通信模块的物理接口、板间通信协议、SDK 开发方式；

(3) 掌握 TI Z-Stack3.0 协议栈以及 CC2530 芯片的应用方法，OSAL 中任务调度、内存管理等机制做到基本了解。

3.CoAP 协议-应用层

(1) CoAP 协议的产生、定义、位置、特点、报文结构和特点，以及与 HTTP 协议的比较和优点做到基本了解；

(2) 能够搭建简易 CoAP 服务器，做到应用 NB 模组连接私有 CoAP 服务器。

4.lora 协议-物理层

(1) 对 Lora 的基本概念、主要特点、系统架构、网络连接方式、工作模式和协议帧做到基本了解；

(2) 可以做到搭建 LoRa 终端、实现 LoRa 终端的软件功能和硬件搭建，能够做到使用 SX1278 芯片和 AT 指令实现 lora 透传、工作模式切换、参数配置（工作频率、信号带宽、无线发射功率、编码率）、数据收发等功能。

5.nb/iot 协议-物理层

(1) 了解 nb/iot 协议特点、网络架构、主要应用场景、工作模式、数据交换流程、工作参数和常用模组与 MCU 对接方案；

(2) 能够使用通用模组完成 MCU SDK 移植、数据接收发送和平台域的设备创建、设备接入和数据接收与下发。

6.蓝牙-物理层

了解蓝牙架构、常用协议帧，能够做到利用蓝牙模组完成开源协议栈移植、数据传输、参数配置等功能，可以利用 Wireshark/Frontline/Ellisys 等分析工具进行调试。

7.蜂窝网络

(1) 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；

(2) 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；

(3) 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；

(4) 可以利用 gsm/gprs 模块完成发送短消息、通话、数据传输等功能。

8.OTA 升级

(1) 对 MCU 中的 BootLoader 的作用、流程做到基本了解；能够做到完成 BootLoader 程序的开发、跳转、固件新版本的检测、固件的升级更新等功能；

(2) 了解 OTA 升级的基本概念、基本流程、工作原理，能够做到使用蓝牙、TCP 协议或 HTTP 协议等完成应用程序的升级。

4.3 有线通信协议

1. Modbus

(1) 对 Modbus 基本概念、通信机制、寄存器、协议类型、通信帧格式做到基本了解；

(2) 可以移植并使用 FreeModbus/libmodbus 开源代码。

2.USB 2.0

(1) 对 USB 体系、协议层规范、数据传输流程、设备架构、工作流程、工作模式等基本概念做到了解；

(2) 能够在 MCU 上应用 USB 接口完成 USB 设备功能（通信、虚拟串口、加电检测等）和主机功能（通信、主机驱动、设备识别、扩展主机模式、OTG 主机模式等）。

3.ETH 以太网

(1) 了解以太网通信层次、IEEE 802.3 标准、TCP/IP 协议栈的基本概念，对 MCU 中的以太网外设 ETH 中寄存器、工作原理、接口做到基本了解，了解 MAC 数据包发送和接收的基本流程；

(2) 可以应用 LwIP 等基本网络协议栈完成网络通信功能。

4.串口通信

对串口通讯的基本概念、数据格式、物理标准、硬件接口、工作参数等做到了解和应用。

五、嵌入式开发环境类：

5.1 构建工具

1. Cmake/CMakeList

(1) 掌握并会 CMake 语法：注释、变量、读取变量的值、控制结构、循环语句、函数和宏、常用命令；

(2) 能够编写基本 CMakeList 脚本，包括设置 CMake 最低版本要求、设置 C++ 标准、定义用户可配置的选项、自定义宏、链接静态库和动态库、修改链接标志，使用延迟加载机制、根据目标架构定制编译选项和链接选项、添加子项目、生成配置文件、指定安装规则；

(3) 可以使用 cmake 完成项目编译、安装、打包和测试等功能。

2. Make/makefile

(1) 掌握并会 makefile 语法：包括语法规则-依赖关系和依赖方法、变量、变量赋值、预定义变量、函数、伪目标 .PHONY、嵌套执行、路径指定等；

(2) 可以使用 make 完成项目编译、安装、打包和测试等

3. Buildroot

了解并会使用使用 buildroot 构建嵌入式 Linux 系统的根文件系统，包括配置 buildroot 源码、工具链、系统配置、文件系统格式和编译 Linux 内核和 uboot。

5.2 编译器

1. GCC

可以使用 GCC 编译工具链完成项目编译或者交叉编译。

2. ARMCC5/ARMCC6

了解并可以使用 armcc 编译.c/.cpp 源文件文件，生成目标文件，同时了解并可以配置通过各种编译选项 command-line 来支持各种特性。

5.3 集成开发环境

1. mdk5.35

主要学习 mdk5.35，常用软件包安装、MDK 软件包管理及导入、SVCS-Git 版本管理、编译及调试配置、调试（程序编译与一键下载、单步执行、逐行执行、跳出函数、断点、执行到光标行、外设寄存器状态查看、变量监测、内存查看、逻辑分析、Debug Viewer、系统分析、Event Recorder 调试、代码时间统计、命令窗口、工具箱的使用）。

2. Vs2019

能够使用 vs 完成项目创建、项目编译和调试。

3. Vscode-mdk

能够使用 vscode+Keil Assistant 插件+openocd 配合 mdk 完成项目代码的编写、编译和调试。

4. Clang

能够使用 Clang 完成项目创建、项目编译和调试。

5. stm32cubeIDE/stm32cubeMX

(1) 能够使用 STM32CubeMX 完成项目创建、MCU 选型、时钟和外设配置、工程配置、功耗计算、组件库添加和项目生成等功能；

(2) 能够使用 STM32CubeIDE 完成代码编写、编译、下载和调试等功能。

6. 串口屏开发软件 USART HMI

能够使用 USART HMI 完成 tjc 串口屏项目创建、界面设置、项目编译下载和调试。

7. Maixpyide

能够使用 Maixpyide 完成 K210 开发板的项目创建、项目编译下载和调试。

8. OpenMV IDE

能够使用 OpenMV IDE 完成 OpenMV 开发板的项目创建、项目编译下载和调试。

9. Thoony

能够使用 Thoony 完成 Mpy 开发板的项目创建、固件烧录、项目编译下载和调试。

10.pycharm

可以完成在 PyCharm 中编写代码、运行代码、代码 debug 和测试、编辑已有项目、搜索和导航、使用版本控制、使用插件和外部工具、使用 PyCharm Professional 功能，如 Django 支持和科学模式。

六、操作系统类：

6.1 linux 基本使用

1. 远程连接

可以使用 SSH 协议、Telnet 协议、RDP 协议、RFB 协议进行 Linux 远程管理与配置。

2. 常用命令行操作

了解并可以使用 Linux 命令完成文件传输、备份压缩、文件管理等操作。

3. 任务调度

了解并掌握 Linux 操作系统任务调度中：调度单位、任务类型、调度策略、优先级、调度类、调度实体、调度算法、调度执行、时钟中断处理等概念。

4. 内存管理

了解并掌握 Linux 操作系统内存管理中：虚拟地址、物理地址、用户空间、进程与内存、内核空间、直接映射区、内存数据结构、内核空间动态分配内存数据结构、物理内存管理、外部和内部碎片、页面管理算法、虚拟内存分配等概念。

5. 同步互斥机制

(1) 对 Linux 中常用的同步机制包括互斥锁、读写锁、自旋锁、信号量、条件变量等做到理解和掌握；

(2) 对 Linux 互斥机制中的中断屏蔽、原子操作、自旋锁及其各种衍生锁和信号量做到理解和掌握。

6. 文件系统操作

掌握并了解文件系统的基本组成、虚拟文件系统、文件的使用、文件的存储、连续与非连续空间存放方式、Unix 文件的实现方式、空闲空间管理、文件系统的结构、目录的存储、软链接和硬链接、文件 I/O、缓冲与非缓冲 I/O、直接与非直接 I/O、阻塞与非阻塞 I/O 与同步与异步 I/O。

7. 进程通信

了解并掌握管道（包括无名管道和命名管道）、消息队列、信号量、共享内存、Socket（套接字）等进程间通信方式。

6.2 嵌入式实时操作系统

1.Uclinux/ucos/RT-Thread/ThreadX

同下类似

2.Freertos

包括任务创建，任务调度，消息队列，信号量，互斥组，任务同步，内存管理，Tracealyzer 调试追踪。

(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 CMSIS RTOS 标准，了解不同 RTOS 间区别和特性；

(2) 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) 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) 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) 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) 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) 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) 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 OSTick 的实现；理解如何获取时钟节拍和使用时钟节拍；

(9) 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) 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) 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) 了解事件组概念，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；掌握事件组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；掌握事件组的发送和接收 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；了解事件组的使用场合；

(13) 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；

(14) 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；

(15) 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；

(16) 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。

七、组件库类（MCU）：

1. ARM-DSP

(1) 学习数字信号处理基本原理，能够应该 C 语言实现相关功能；

(2) 可以使用 DSP 库完成信号的统计计算、复数计算、矩阵运算等功能，使用 DSP 库完成 FFT 和信号滤波功能，使用 Matlab 设计滤波器等相关算法。

2. CMSIS-NN/X-CUBE-AI

使用 TinyML 框架完成：

- (1) 从预训练神经网络和经典机器学习模型生成 STM32 优化库；
- (2) 调用内置 scikit-learn 模型，如孤立森林、支持向量机(SVM)、K-means；

3. CANopen

了解并掌握 CAN 应用层协议、CANopen 网络模型、常见规范、对象字典、通讯标识符等概念，CANopen 协议中管理报文、紧急报文、SDO（服务数据对象）通讯、PDO（过程数据对象）通讯、同步报文。

4. FreeModbus

- (1) 了解 MODBUS 协议介绍标准、Modbus 主站/从站协议原理和两种传输模式等概念；
- (2) 能够利用 FreeModbus 完成主从站通信功能。

5. EmWIN/TouchGFX/STemWin

能够运用常见 GUI 框架完成人机交互界面的设计。

6. RL-USB/ST USB

- (1) 了解 USB 相关基础知识，包括 USB 架构、硬件设计、通信相关概念（管道与传输、事务、描述符和类）；
- (2) 能够移植开源 USB 协议栈，完成 USB 通信；

7. LwLP/RT-TCPnet

- (1) 了解以太网基础知识、STM32 自带 MAC 基础知识、外置以太网 PHY 基础知识，熟悉 CMSIS-Driver 概念、工作流程、驱动实现；
- (2) 可以进行小型入式网络协议栈移植，利用网络调试助手使用 TCP、UDP 协议进行通信。

8. FatFS/RL-FlashFS

- (1) 了解 SD 卡硬件接口设计、SD 卡概念、各自存储卡特点等基础知识；
- (2) 可以移植小型嵌入式文件系统，利用相关驱动函数完成文件浏览、文件创建、读写、删除等操作。

9. letter-shell/cmd-parser

应用 letter-shell 完成命令自动补全、快捷键功能定义、命令权限管理、用户管理、变量支持、代理函数和参数代理解析。

10. MultiButton

事件驱动型按键驱动模块。

11. Easylogger

使用 elog 完成日志输出与过滤功能、缓冲输出模式、异步输出模式。

12. SFUD

串行 Flash 通用驱动库。

13. FlashDB/EasyFlash/sqlite

轻量级的嵌入式数据库，应用其完成产品参数存储、用户配置信息存储、小文件管理、记录运行日志、存储动态产生的结构化数据等功能。

14. MultiTimer/SmartTimer

软件定时器扩展模块/定时器调度器。

15. cJSON/jsmn

基于 C 的 JSON 解析库。

16. CmBacktrace

错误代码自动追踪、定位，错误原因自动分析的开源库。

17. Ringbuff/lwrb

循环队列，应用于数据高吞吐量场景。

八、编程设计模式和函数接口设计：

1. 编程设计模式

了解并掌握宏协程（cothread、Protothreads）、事件驱动架构、消息驱动框架、数据驱动框架（表驱动法-静态结构体数组式构建、链表式构建、链接式构建）、嵌入式实时操作系统、osal 操作系统抽象层等编程设计模式的应用原理和设计方法。

2. 函数接口设计

（1）了解并掌握面向对象编程和模块化思想，了解接口的概念和实现方法以及抽象数据类型的定义与实现；

（2）了解并掌握 C 语言的原子定义、实现方法和操作方法；

（3）了解并掌握 C 语言中的异常与断言中接口与实现；

（4）了解并掌握 C 语言中的链表、关联表、集合、动态数组、序列、循环数组等数据结构的实现方法。

九、常用调试技巧及软件类：

9.1 调试技巧：

断点调试、单步执行、逐行执行、外设寄存器状态查看、变量监测、内存查看、逻辑分析、系统分析、时序分析、汇编调试、linux 命令行调试。

9.2 软件：

Event Recorder+MDK、逻辑分析 Kingst、mcuISP 串口助手、ST-LINK Utility、NetAssist 网络调试、STM32CubeMonitor 内存监测、SecureCRT、MobaXterm、whbots PID 调试工具、usb2s 调试工具、字库制作软件、putty 远程连接工具、balenaEtcher 镜像烧录工具、CVSP Driver 虚拟串口工具、UltraISO 镜像烧录工具、understand 代码分析工具、bouml 代码分析工具。

十、常用工具软件类：

10.1 代码管理工具

1.Git+gitgui

2.svcs+git

3.git+github desktop

- (1) 了解并掌握 Git 的基本概念、工作流程、工作区、暂存区和版本库；
- (2) 可以使用 Git 完成创建仓库、克隆仓库、添加文件到暂存区、将暂存区内容添加到仓库中、回退版本、分支切换、查看历史提交记录、从远程获取代码库、下载远程代码并合并、上传远程代码并合并操作；
- (3) 了解并掌握 Git 分支管理功能，包括：列出分支、删除分支、分支合并、合并冲突等功能；
- (4) 可以在 Github 上添加远程库、查看当前的远程库、提取远程仓库、推送到远程仓库、删除远程仓库等操作；
- (5) 可以将集成 IDE 与 Git 配合使用。

10.2 文本编辑类

1.Notepad++/UltraNote

文本编辑

2.Typora

- (1) 学习 MD 语法：包括标题、段落格式、列表、区块、代码区块、链接、图片插入、表格插入等；
- (2) 内容编辑、字数统计、文件导出、编辑配置等功能。

3.beyond compare

使用 Beyond Compare 完成文件和文件夹比较、文件和文件夹同步、数据备份等操作。

十一、常用电子模块使用类：

OLED 屏幕，DTH11 温湿度传感器，LM75 温度传感器，MPU6050，LCD1206，L298N，步进电机，超声波测距，SD 卡，OV7670 摄像头，串口屏，舵机，继电器，HC05 蓝牙，NRF24L01 单片射频收发模块，霍尔传感器，无刷直流电机，AD 模数转换模块（SPI），遥感模块。

十二、硬件类：

12.1 常用电路设计与仿真软件

了解并掌握立创 EDA，AD，NI Multisim 14.0，Proteus 8 Professional，LTspice 等软件的基本操作。

12.2 模拟电子电路设计

0. 基础

了解常见电路分析方法：基尔霍夫定律、叠加定理、戴维宁定理原则，了解常见电子元器件作用与特性。

1. 三极管放大电路

了解三极管的基本功能、输入回路、输出回路、输入特性曲线、输出特性曲线、静态工作点、放大电路分析、负反馈放大电路、常见三极管型号、低频小信号模型等内容。

2. 三极管开关电路

了解三极管的工作状态分析、开关特性、开关电路的逻辑关系和快速判断方法，以及三极管的过饱和和过流的原因和影响。

3. 三极管组合放大电路

CE-CB 放大电路、CC-CE 放大电路中静态分析和动态分析。

4. mos 管开关电路

了解 MOS 管的基本知识（种类和结构、导通特性、开关管损耗）、开关特性和开关电路（电源 IC 直接驱动、电源 IC 结合图腾柱电路、带关断瞬间驱动的开关电路）实例，以及注意事项和选择原则。

5. Mos 管驱动放大电路

MOSFET 的特性及各电极间所加电压极性的规律、主要参数、MOSFET 构成基本放大电路及其分析方法、MOSFET 放大电路三种组态。

6. 差分放大电路

了解并掌握 BJT 差分放大电路设计和分析方法。

7. 集成运算放大电路

（1）运放基础：了解运算放大器的概念、分类、原理和应用，以及常见的放大电路和信号放大器的原理和结构及其分析方法；

（2）线性应用：了解引入负反馈，线性工作区；介绍运放的基本知识，传输特性曲线，同相比例、反向比例运算电路，电压跟随器，差分比例运算电路、加法电路、减法电路、微分积分电路等；

（3）非线性应用：了解并掌握运放非线性区特性、基本电压比较器、限幅电压比较器、滞回电压比较器等；

（4）调理电路：了解并掌握有源滤波电路、低通滤波器、一阶低通无源滤波器、一阶低通有源滤波器、二阶低通有源滤波器、低通滤波器的高频馈通、高通滤波器、三阶高通、高 Q 值带通、带通滤波、带阻滤波有源滤波器中无源器件和运放的选择原则。

8. 反馈电路

了解并掌握电路反馈的基本概念、分类、判断、作用和分析方法，以及反馈的优缺点和应用场景。

9. 功率放大电路

了解并掌握功率放大器的特点、功率管的结构和特性、功率放大器的设计思路和分析方法，以及功率放大器的应用场景和注意事项。

10. 振荡电路/非正弦波振荡电路

(1) 了解并掌握 RC 振荡器的原理、选频特性和选择方法，以及 RC、LC 桥式、文氏、超前、滞后等不同类型的 RC 振荡电路的结构和性能；

(2) 了解并掌握矩形波、三角波、锯齿波、占空比可调的矩形波发生电路。

11. 直流稳压电路

了解并掌握了整流、滤波、稳压电路的基本概念、结构和性能指标，以及不同类型的滤波电路的原理和应用。

12. 频率特性

了解并掌握放大电路频率响应、幅频特性、相频特性、下限频率、上限频率、通频带、频率失真、波特图、高通电路、低通电路、共射截止频率、特征频率等概念和测量方法。

12.3 数字电子电路设计

1. 组合逻辑电路及分析方法

2. 半导体存储电路

3. 时序逻辑电路

4. 脉冲波形产生和整形电路

5. 数模和模数转换电路

12.4 PCB 设计

0. 基础

(1) 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；

(2) 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；

(3) 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。

1. PCB 设计一般性原则

2. 电磁兼容性设计原则

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。

12.5 电子电路仿真

1. 模拟电子电路仿真

了解电路仿真软件基本概念、原理图录入与探针、虚拟测试仪器、分析与仿真（直流工作点仿真、直流参数扫描、瞬态分析、正弦稳态仿真与单一频率交流仿真、功率计、Bode 图绘制仪与交流分析、温度扫描仿真）、Spice 仿真模型导入等方法。

2. 数字电子电路仿真

十三、嵌入式自动化测试：

了解并掌握软件测试基础知识、单元测试、静态测试、测试用例设计、TDD 相关机理、测试框架环境搭建的基本概念和使用方法，了解常见 C 语言测试框架，如 zUnit、Unity、Embedded Unit、CuTest；

（1）可以使用嵌入式测试框架，完成编写测试用例、断言、信息输出等功能，同时完成代码覆盖率分析；

（2）能够完成编写配置文件、编写测试脚本、生成数据文件、测试报告分析等工作。

十四、上位机开发：

1. Python 面向对象

了解并掌握 Python 面向对象编程的基本概念，如方法、属性、类（类的成员、成员修饰符、类的特殊成员，包括字段、方法、属性和类的特性）、对象等，以及如何用 Python 编程语言实现面向对象编程的逻辑和优缺点。

2. Python 并行计算

（1）了解 Python 并行编程、内存架构、内存管理、并行编程模型的相关概念，掌握并行程序设计、评估并行程序的性能的基本方法，了解 Python 中多进程、多线程、协程和异步编程的基本概念；

（2）了解多线程的特性、使用方法、线程同步、线程通信、多线程应用的性能评估方法；了解多进程的特性、使用方法（创建杀死操作）、进程通信、进程同步、进程状态管理方法、进程池的使用方法、进程间点对点通讯方法、集体通信方法；了解并掌握协程的概念、优势、使用方法和示例，以及如何使用 yield、gevent 和 greenlet 模块实现协程函数；

（3）了解并掌握 Python 异步编程基本概念、能够使用 Asyncio 完成管理事件循环、管理协程、控制任务等操作。

3. Python 相关数据处理库

使用 Numpy、SciPy、Matplotlib、Pandas、ScikitLearn、Keras、Gensim 能库完成数据分析、数值计算、数据挖掘等操作。

4. Python 相关通信库

（1）可以使用 socket 库实现网络通信，包括创建 TCP/P 或 UDP 套接字、发送和接收数据等；

（2）可以 pySerial、serial 等库实现与串口设备进行通信；

十五、学习目标：

能够基于科学原理并采用科学方法，对电子信息领域智能电子产品及系统软硬件开发、智能系统集成与设计等工程问题，开展有针对性的建模、仿真与优化研究并设计实验方案，构建实验系统进行相关的数据采集、分析与处理，并通过信息综合得到合理有效的结论。

修订历史

v1.0 版本

时间：2023 年 01 月 04 日

修改人：工训 A413-郝翹活

修改内容：初始化文件

v2.0 版本

时间：2024 年 01 月 07 日

修改人：工训 A413/Freak 嵌入式-郝翹活

修改内容：

1. 增加 MCU 组件库使用、编程设计模式和函数接口设计、嵌入式自动化测试、上位机开发、学习目标等大纲；
2. 增加具体学习知识点内容。

更多精彩内容可以关注：

公众号：工训 A 座 413/Freak 嵌入式

Bilibili：工训 A413 实验室/Freak 嵌入式

抖音：大学竞赛帮

联系方式：a413liqingshui@gmail.com/<https://github.com/leezisheng>