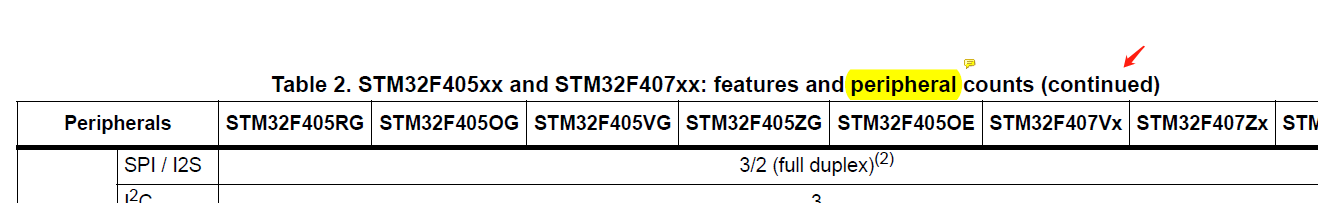
**5.6**

1. 设计不能太简单，需要考虑未来需要添加的功能
2. 在设计的时候需要留好测试口，为了方便在机器故障的时候去检测出问题的地方
3. 设计完电路以后，需要仿真电路怎么弄
4. 在设计电路的时候电压一般在芯片手册中都有告诉，**电流需要考虑到什么程度呢？是不用去管它还是说有的芯片手册上明确标明就去考虑没有标明就不去考虑**？
5. 1 mm **height profile** 这是什么意思 在ADP2504芯片手册中
6. **Compact** PCB Footprint
7. quiescent current 静态电流
8. **typical** quiescent current
9. Low-Dropout Regulator 低压差稳压器
10. APOWER中的DAC\_LV和HPOWER中的DAC\_HV是做什么的——先去看LMR62014XMF中的电路去理解
11. 查找CPU芯片中的ADC端口
12. 什么是看门狗？
13. 芯片手册上ADC123，ADC12和ADC3为什么要分为这三个名字呢
14. Oscillator mode 震荡模式
15. FSMC是什么——总线控制器
16. 晶振的频率——2^15=32768Hz
17. NRST是异步复位引脚
18. Camera interface 视频接口 在这里interface是接口的意思
19.  图中标题是什么意思，箭头指的地方是接上一个表格的意思，“继续”
20. 单精度和双精度（single precision and double precision）的差别：表明单精度和双精度精确的范围不一样，单精度，也即float，一般在计算机中存储占用4字节，也32位，有效位数为7位；双精度（double）在计算机中存储占用8字节，64位，有效位数为16位。

原因：不管float还是double 在计算机上的存储都遵循IEEE规范，使用二进制科学计数法，都包含三个部分：符号位，指数位和尾数部分。其中float的符号位，指数位，尾数部分分别为1, 8, 23. 双精度分别为1, 11, 52。精度主要取决于尾数部分的位数，float为23位，除去全部为0的情况以外，最小为2的-23次方，约等于1.19乘以10的-7次方，所以float小数部分只能精确到后面6位，加上小数点前的一位，即有效数字为7位。 类似，double 尾数部分52位，最小为2的-52次方，约为2.22乘以10的-16次方，所以精确到小数点后15位，有效位数为16位。

1. MSPS 和 KSPS转换速率的单位 每秒采样百万次，后者是每秒采样千次
2. USB协议：<https://blog.csdn.net/k331922164/article/details/53349360>
3. LSB 最低有效位 MSB 最高有效位 <https://zhidao.baidu.com/question/155072477.html>
4. 20个基础电路<https://www.bilibili.com/video/BV1T4411G7Kj?p=16>
5. 自制单片机最小系统 <https://www.bilibili.com/video/BV1Ab411K7yp?p=8>
6. 焊接<https://www.zhihu.com/question/22556498>
7. 单片机最小系统详解

<https://blog.csdn.net/alala120/article/details/81266148?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1>

1. 32编程通常是使用库的，而51是需要自己写函数的，所以要是以51的方式去写32的程序的话就会复杂很多
2. 实时操作系统——RTOS
3. 一种按键扫描方案<https://mp.weixin.qq.com/s/vhMHsnWEx9fZGCQt4cvgQw>

5.7

1. Ipv5是奠定了IP语音技术，Ipv4是旧一代的互联网协议（其地址已耗尽），Ipv6是新一代的互联网协议，128位，能够提供大量的IP地址
2. **消除电磁干扰EMA**的八个方法：<https://mp.weixin.qq.com/s/NuVoZz5k6PEgrQ8mAWalOw>

传导干扰其实很容易解决，只要增加电源输入电路中EMC滤波器的节数，并适当调整每节滤波器的参数，基本上都能满足要求

* 1. 减少每个回路的有效面积
  2. 屏蔽、减小各电流回路面积及带电导体的面积和长度
  3. 对变压器进行磁屏蔽、尽量减少每个电流回路的有效面积
  4. 用铜箔对变压器进行屏蔽
  5. 采用双线传输和阻抗匹配——当导线的长度等于或大于四分之一波长时，传输信号的线路一定要考虑阻抗匹配，不匹配的 传输线会产生驻波，并对周围电路产生很强的辐射干扰
  6. 减小电流回路的面积
  7. 不要采用多个回路串联供电——几个电流回路，互相串联在一起进行供电，很容易产生电流共模干扰，特别是在高频放大电路中，会产生高频噪音。电流共模干扰的原因是： ∆I2 = ∆I3+ ∆I4+ ∆I5
  8. 避免干扰信号在电路中产生谐振

1. **PCB层叠设计**遵守的要求：<https://mp.weixin.qq.com/s/km5-J3xwRi3VsTyDANqELA>

① 元件面、焊接面为完整的地平面（屏蔽）。

② 尽可能无相邻平行布线层。

③ 所有信号层尽可能与地平面相邻。

④ 关键信号与地层相邻，不跨分割区

1. **进程**：进程是系统进行资源分配和调度的一个独立单位——一个进程可以包括多个线程

**线程**：线程是进程的一个实体，是CPU调度和分派的基本单位，线程自己基本上不拥有系统资源，只拥有一点在系统运行中必不可少的资源，但是它可与同属一个进程的其他线程共享进程所拥有的全部资源。

<https://mp.weixin.qq.com/s/62oUz3N5pCDXeM8cLt00zw>

1. 进程（process）和线程（thread）是操作系统的基本概念，但是它们比较抽象，不容易掌握。
2. **电容的诸多作用**：<https://mp.weixin.qq.com/s/_ORqNaeBCf-7sQJ34HnaGQ>
3. 32开发先去使用标准外设库**SPL**，在熟悉数据结构以后（指针，结构体，回调函数这一类C语言知识）再进行学习**STM32CubeMX**

<http://m.elecfans.com/article/885008.html>

**cubeMX是软件**。相当于可视化的IDE，而**HAL库**就像标准外设库一样。把库安装到cubeMX里就可以配置MCU管脚初始化,生成程序框架,不用再自己搭建工程了

1. **IGBT**与**MOS**管的区别：IGBT是MOS管和晶体三极管组合而成的复合型半导体器件
2. **堆栈**
3. **短路的判定方法**
4. 分布电容、有极性电容、无极性电容、钽电容、安规电容
5. **电容的分类**：按结构、介质、极性、用途

按结构可分为：固定，可变，微调

按介质可分为：空气，液体，无机固体，有机固体

按极性分为：有极性和无极性

<https://blog.csdn.net/lz_kwok/article/details/52020350>

1. 电容的作用：级间耦合、滤波、去耦、旁路及信号调谐。——**黄工的公众号里也有讲**

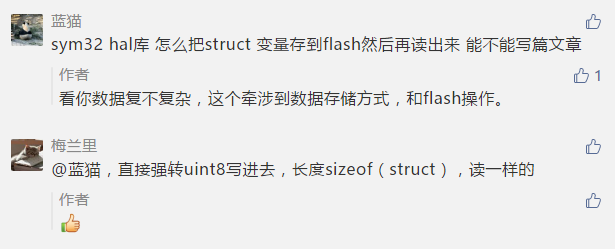
<https://mp.weixin.qq.com/s/_ORqNaeBCf-7sQJ34HnaGQ>

1. **有极性电解电容器**通常在电源电路或中频、低频电路中起电源滤波、退耦、信号耦合及时间常数设定、隔直流等作用。一般不能用于交流电源电路，在直流电源电路中作滤波电容使用时，其阳极（正极）应与电源电压的正极端相连接，阴极（负极）与电源电压的负极端相连接，不能接反，否则会损坏电容器。
2. 看完《开关电源理论及设计》---周洁敏著，明白了电解电容和无极性电容的区别：

一般极性电容只允许工作在直流电路，纹波交流幅值相对于直流偏置很小；**无极性电容既允许在直流电路工作，也允许在交流电路工作**。但由于交流损耗限制，交流允许电压一般比直流电压低，并随着工作频率增加而降低。

1. 电子电路抗干扰实用技术——一些技术问题还是需要去看论文，然后整理出来思路，自己去验证。网上的东西太多，好多人都是谈了一下自己的经验之谈。没有理论的支撑，不知道知识是真是假。如果说能找到一些权威的书籍去用去参考就方便的多。电子书虽说看起来方便，但是如果自己的分类做不好的话就是一团乱麻。所以需要自己去整理，去做读书笔记。
2. 只是输出电压，需不需要接上USB，PC端上可以去读取SD卡上的数据。什么时候输出电压呢？还是说测量时主功能，预留和控制器功能合并的位置。手动输出电压么，还是说以蓝牙为载体去控制测量检测器，去输出电压。这样的话还需要加上通讯功能。现在仅仅是能够测量然后并显示出来电流和电压就可以了。其他功能考虑实时系统去控制。使用32做一个显示器。
3. Hardware、STM32、programming
4. 电池管理芯片：**CP4061QN32-A1** <https://item.szlcsc.com/238278.html>
5. **74HC595芯片**，这款芯片是一个**串转并**的芯片，将串行数据并行输出，它最大的好处是，能剩下大量的IO，理论上，无数个595首尾相连，你能够用三个单片机引脚控制无数个管脚
6. **超级轻黏土**，可以作为任何东西的外壳
7. PCB封装、CAE封装、元件类型
8. CAE封装绘制
   1. 点击元器件编辑，然后点击确认后出来的四个框框都是做什么用的，后面Label1和Label2这两个难道是仅仅是为了补充元件说明么
   2. 管脚封装类型
      1. PIN
      2. PINB
      3. PCLK
      4. PCLKB
      5. PINIEB
      6. PINORB
      7. PINSHORT
      8. PINVRTS
   3. 管脚电器类型
      1. Source——信号源引脚，也就是output，输出信号。
      2. Bidirectional——双向引脚，也就是GPIO，具有输入输出功能
      3. Open collector——开集电极引脚，三极管集电极没有上拉，参考模拟电路三极管部分，基础知识
      4. Or-tieable table——或可连接的源引脚，可以或方式连接在一起的输出信号源，参考数字电路基础知识。
      5. Tristate——三态信号引脚，参考数字电路基础知识
      6. Load——负载引脚，也就是input
      7. Terminator——信号终端引脚，传输线信号端接，参考传输线理论基础知识，微波电路里面也有这个介绍。
      8. Power
      9. Ground
      10. Undefined
   4. 元件封装制作规范<https://wenku.baidu.com/view/04de01750066f5335a812165.html?fr=search>
   5. 低电平的话需要在端点名称上添加 \N\ 这样两个斜杠

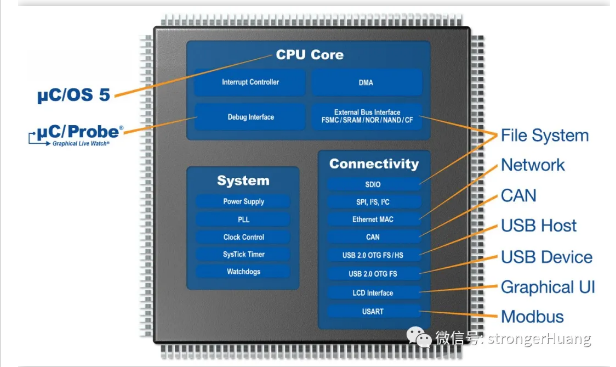
5.8

1. **数据结构**：分类大概包括三种数据结构，线性表、树结构、图存储结构
   1. 线性表：可细分为顺序表、链表、栈和队列……
   2. 树结构：可分为普通树、二叉树、线索二叉树……
   3. 图存储结构
2. 数据结构是研究数据存储方式的一门学科，数组只是各种编程语言中的基本数据类型，并不属于数据结构的范畴。
3. 链表的存储方式并不是存储地址相邻式的存储，而是随机存储，它用到了指针，每一个数据块的指针都指向了下一个存储块，最后一个数据块的指针指向了NULL，毫无排列顺序的数据通过指针就建立了依次排列的链表。
4. 栈是先进后出，封死了一端。队列是先进先出，并未封死一端
5. 树是一对多的数据结构。
6. 图是多对多的数据结构。
7. 
8. 实时内核和实时操作系统的差别：

实时内核是管理微处理器（MPU）、微控制器（MCU）或数字信号处理器（DSP）的时间和资源的软件，比如任务调度、信号量、队列等资源，它为应用程序提供了不可或缺的服务。

实时操作系统（RTOS）通常包含实时内核和其他高级服务，如文件管理，协议栈，图形用户界面（GUI）和其他组件等，大多数其他服务都围绕I / O设备。

拿µC/OS来说：Micriμm提供了一套完整的RTOS组件，包括嵌入式文件系统，TCP / IP堆栈，图形用户界面，USB设备和主机堆栈等，你只需要选择应用程序所需的组件即可。



1. 小型文件系统：FatFS和LittleFS

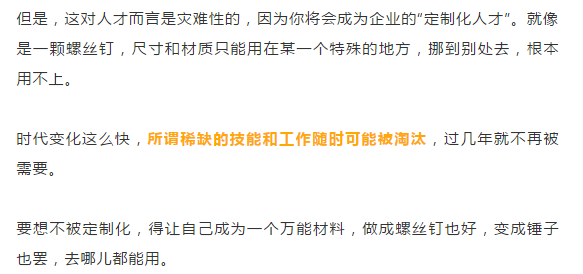
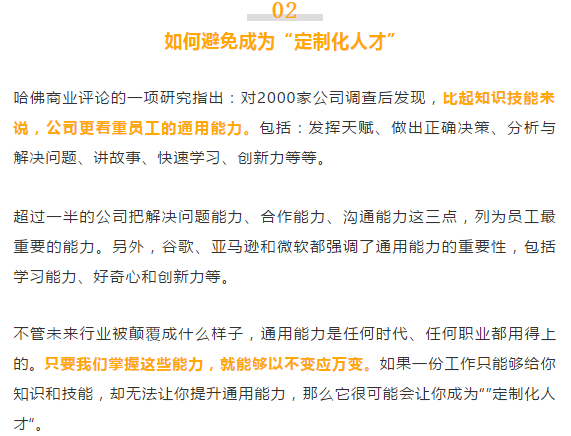
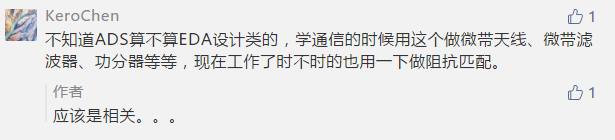
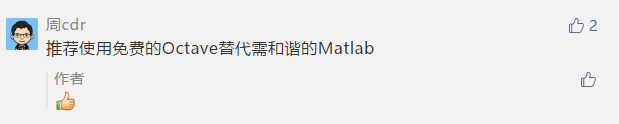
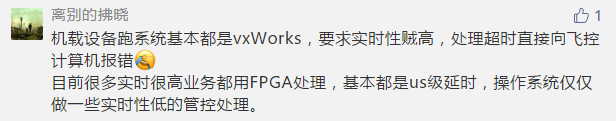
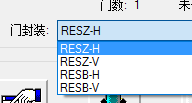
FatFS是一个通用的文件系统(FAT/exFAT)模块，用于在小型嵌入式系统中实现FAT文件系统。网址：

<http://elm-chan.org/fsw/ff/00index_e.html>

Littlefs是Mbed OS的一部分，Mbed OS是arm公司针对Cortex M系列处理器开发的一套免费开源的嵌入式操作系统

<https://github.com/armmbed/mbed-littlefs>

<https://os.mbed.com/blog/entry/littlefs-high-integrity-embedded-fs/>

1. 
2. **Linux系统到底好在哪里，为什么能够赢得嵌入式工程师的青睐呢。**ubuntu
3. The internet of things是物联网的意思，缩写为loT 包含云端、联网模块、终端APP三大部分
4. GPRS英文简称为 General packet radio service，中文名称为通用无线分组业务，是一种基于GSM系统的无线分组交换技术，提供端到端的、广域的无线IP连接。相对原来GSM的拨号方式的电路交换数据传送方式，GPRS是分组交换技术
5. PCB的间距要求：<https://mp.weixin.qq.com/s/81e4hu-UbypQ7PRqJiU2Hw>
6. 
7. 航空软件并不神秘，从计算机架构上来说，同普通的计算机系统并无太大区别，都是由处理器，总线，I/O设备，存储设备，网络设备，通讯设备，操作系统和应用软件所构成的
8. 
9. CAE封装建立——先建立门（CAE封装），保存，然后建立元件类型，匹配好PCB封装 属性是什么，必须设置么
10. 原理图纸尺寸设置：工具-选项-设计
11. BOM输出——报告-材料清单、或者可以利用脚本输出到excel
12. 在制作PCB封装的时候，焊盘要是规格书上管脚的两倍，不然焊接不上，这是贴片型封装设计方法，如果是BGM封装就不一样了
13. PCB封装的层是怎么分的，在设置的时候有好多好多层，是不是有些层是不重要的，有些层是重要的
14. PCB封装制作的时候可以直接使用封装向导来制作，方便快捷，省去了自己去排布
15. 设备类型中通孔是指的是管脚插件，留一个孔以便插过管脚。而SMD是指的是贴片
16. 无模命令 S 元件编号 就可以快速定位到该元件
17. PCB规则设置是什么规则的，比如过孔的直径，钻孔尺寸。安全间距 PCB设计规范
18. 仿真，怎么仿真
19. 吴川斌的博客
20. 封装0805、0603、0402、SOP8 根据封装去学习，了解相应的元器件
21. 过孔大小 0402 0806 0604 指的是直径为0.8 钻孔为0.6
22. 无模命令dro——不按规则走线
23. 无模命令 <https://www.cnblogs.com/asus119/archive/2012/02/06/2339732.html>
24. mil : mm = 1 : 39.37
25. **先做一个最小系统板练一下手**
26. RESZ-H RESZ-V RESB-H RESB-V这些都是啥：

4.9

1. C11标准文档——<https://ishare.iask.sina.com.cn/f/Rh1IWMBuXS.html>
2. **软件分层**——分为设备层和应用层，这个界限是什么。**软件工程**的教材
3. 面向对象分析与设计——原作者是外国人，国内是译文版

<https://book.douban.com/subject/11509672/>

<https://www.jianshu.com/p/c0d4c23da7c7>

1. 页间连接符——电源特殊符号的名称对电路有影响没
2. PCB为什么要分为电气层和常规层
3. 同样是0402的封装，电阻的封装和电容的封装有区别么
4. 分模块，分层次，单模块测试

先做一次全流程，从原理图到PCB封装——最小系统板

再做一个单芯片测试，仿真？AD采集功能，从废旧的板子上面拆下来元器件，然后功能测试，在洞洞板上

之后就是可以把全部功能加进去，电池测量，显示什么的

1. 元器件的名称在CAE封装的时候，或者是在PCB封装的时候就摆好位置，不要离得太远，以免在排布PCB的时候重新去调整位置，麻烦的很