



所思所想

2019 年 9 月 1 日

柳 钧腾

目录

学习与发展方向2

具体经验汇总3

 Altium Designer3

 软件驱动4

 CubeMX.....4

实验室制度管理.....4

学习与发展方向

实验室待了一年，算是一个年度总结。

小黄车比赛后来到实验室，要选方向，软件 or 硬件，本来准备敲码去，看着同届实验室的人自己写的俄罗斯方块我沉默了，就我这个水平，也许都不够打杂的，硬件大家都没接触过，于是果断选硬件，事实上现在看起来当初选择是对的，硬件打开了新天地。

智能车比赛中，画了三代主板，因为是小白四轮所以几乎没有难度，调过一些模块，做过代码移植，但感觉所有的东西都是涉猎，并不精通，这是我一年最大的问题。

接下来是对学习的一些看法：

1. C 语言可以不懂文件链表，但是基础的比如循环，结构体，枚举体，联合体要熟练，对我而言我不是专门做软件的，C 语言有这个水平足矣。
2. 我们最常用的通讯协议，SPI，IIC，UART 要弄明白，可以自己不会写但是要看得懂例程，UART 只要会配模块就行了，其他两个要自己做软件式通讯练习一下。
3. 关于 PCB，大部分观点是在实现相同功能的情况下尽量减少面积，我的观点一直是要在面积相同的情况下尽量增加功能。好像看起来一样，其实不一样，决定面积的是你要放的位置，而不是你能画多小。板子上功能的多少应该取决于你能做多少，而不是软件队友要多少。
4. 关于创新，比如有一款实验室没有人用过的芯片要不要用？这是一个稳定性与学习知识的冲突，要使用可能会带来稳定性问题，要稳定又学不到新的知识。对此问题仁者见仁智者见智，我倾向于稳定性。

具体经验汇总

首先必须承认我这一年没有深入学习到知识，大部分只是涉猎甚至没有入门，所以此篇我会写一些很简单的知识。

Altium Designer

写程序是不可能写程序的，这辈子都是不可能的，搞上位机又不会，只有学学 AD 画画板子，勉强维持的了生活这样子。AD 作为实验室一直在用的画 PCB 的软件，上手容易，操作简单，功能强大。教程可以去 mooc 找，我们一起学硬件的当初是两天学会的 AD，也就是只要两天就可以学会使用 AD，但要学的深还要多画。

a. 画板子步骤：原理图，布局，布线，铺铜，修线，打丝印。

Trace Carrying Capacity per mil std 275									
Temp Rise	10 C			20 C			30 C		
Copper	1/2 oz.	1 oz.	2 oz.	1/2 oz.	1 oz.	2 oz.	1/2 oz.	1 oz.	2 oz.
Trace Width	Maximum Current Amps								
.010	.5	1.0	1.4	0.6	1.2	1.6	.7	1.5	2.2
.015	.7	1.2	1.6	0.8	1.3	2.4	1.0	1.6	3.0
.020	.7	1.3	2.1	1.0	1.7	3.0	1.2	2.4	3.6
.025	.9	1.7	2.5	1.2	2.2	3.3	1.5	2.8	4.0
.030	1.1	1.9	3.0	1.4	2.5	4.0	1.7	3.2	5.0
.050	1.5	2.6	4.0	2.0	3.6	6.0	2.6	4.4	7.3
.075	2.0	3.5	5.7	2.8	4.5	7.8	3.5	6.0	10.0
.100	2.6	4.2	6.9	3.5	6.0	9.9	4.3	7.5	12.5
.200	4.2	7.0	11.5	6.0	10.0	11.0	7.5	13.0	20.5
.250	5.0	8.3	12.3	7.2	12.3	20.0	9.0	15.0	24.5

b. 时间分配。原理图完成后，布局占 60%，布线 30%，铺铜修线 9%，丝印 1%。为什么布局占这么多？布局上花时间可以让你布线很轻松，万一你布线到一半发现布局需要修改，怎么办？删了线重画？

c. 大电流尽量不走过孔。这不意味着不可以走，如果需要尽量用过孔阵列，其中过孔和电流的关系很复杂，对于我们智能车这种要求不高的 PCB，就那个默认的过孔尺寸 28/50 的有 1 安培就打一个，差不多就够了，郑重声明这个方法只是大致的方法，电流不会均匀分布过孔的，有兴趣自行百度。过孔阵列操作方法：先复制（记住 AD 里复制完要点一下让十字消失），然后快捷键 E，A，选择阵列粘贴，就能弄出一行过孔。

d. 不要排斥过孔。你尽量不打过孔会发现最后地回路还是打了一堆。

e. 关于走线。我一贯都是能走多粗走多粗，这里有张表，一般要有容限，比如要过 1A，就要设计至少

1.5A。要多利用芯片封装中间，电容电阻封装中间用来走线。

- f. 丝印一定要好好标。可以不标，你会看见把 5V 口直接怼 GND 的，然后板子就烧了，实验室老规矩，硬件背锅。

软件驱动

别看上面，写程序真香。

- a. 调模块一般流程：找模块，改代码，测试。
- b. 第一次一般调不通，这时从源头找问题，你的数据发过去了吗？模块回应你了吗？多用逻辑分析仪看看时序哪里不符合预期。
- c. 一般情况下给模块写驱动很多时候都是一个地方没改导致整个不能用，模块不能用绝大部分都是代码问题，多从代码找问题。比如寄存器配置对吗？上拉配置了吗？
- d. 当初我调 NRF24L01 的时候怎么都调不出来，后来发现是电压纹波比较大，加电容解决，射频模块电源要做好。

CubeMX

最后这段时间我在调 STM32F427 的时候发现了一个很好用的软件 STM32CubeMX，这个软件可以调用 hal 库直接生成代码完成初始化寄存器配置等等，支持 STM32 系列所有单片机，强烈建议学习一下。

- a. 这个软件完全和中文没关系，安装路径，文件路径，工程路径等等和它扯到的地方不要出现中文。
- b. 用这个软件生成的代码会有一定的格式，会有用户写码的区域，所以不要写在别的地方，会被覆盖的。
- c. 安装方法等等自行百度，安装完成后记得下载芯片包，在 ST 官网上设计与资源板块，然后解压到指定位置。ps：我当初因为芯片包不对吃了不少苦头。
- d. 代码要整齐，一定要牢记评判代码的指标中最重要的是可读性！

实验室制度管理

在实验室一年很大感受是实验室的自由，但这带来了一定程度的混乱，比如工具，我们很多的时间都花来找工具，这有必要改变。

下面对实验室管理提出个人见解，纯属个人意见，不针对任何人任何事。

1. 我在实验室没有找到白纸黑字的条例，行政与法制应该分离，其中制度应该有纸质文件，并且更改难度极高，比如匿名表决通过率达到 90%。

2. 实验室的开源平台，可以借助 FTP, GitHub 等做开源，而不是通过 QQ 群文件之类方法做共享。FTP 可以将 IAR, Keil, Altium 等软件安装包进行整理。个人文件应当有一定的存放格式。
3. 实验室仪器，材料之类的很多很乱，有些东西屯着没啥卵用。
4. 例会应当作出改革，例会应该有进度检查和交流分享的作用。
5. 前几届学长的智能车会被拆零件，拆轮胎，此种方法不可取，实验室应当设立展示区，用于展示历年的智能车等。