
基于嵌入式硬件构件的软硬件设计规范

2.1 电路原理图设计与绘制基本要求

1. 设计时需要考虑的基本问题

1) MCU的选择

选择 MCU 时要考虑 MCU 所能够完成的功能、MCU 的价格、功耗、供电电压、I/O 口电平、管脚数目以及 MCU 的封装等因素。MCU 的功耗可以从其电气性能参数中查到。供电电压有 5V、3.3V 以及目前正在开发中的 1.8V 超低电压供电模式。为了能合理分配 MCU 的 I/O 资源，在 MCU 选型时一定要做一张引脚分配表。

2) 电源部分

(1) 要考虑系统对电源的需求，例如系统需要几种电源，如 24V、12V、5V 或者 3.3V 等，估计各需要多少功率或最大电流 (mA)。在计算电源总功率时要考虑一定的余量，可取 2 倍，即：电源总功率 = 2 × 器件总功率。(2) 考虑芯片与器件对电源波动性的需求。一般要求在正负 5% 以内。对于 A/D 转换芯片的参考电压一般要求正负 1% 以内。(3) 考虑使用电源模块还是外接电源的方式提供工作电源。

3) 普通 I/O 口

(1) 上拉、下拉问题：考虑用内部或者外部上/下拉电阻，内部上/下拉阻值一般在 700 Ω 欧姆左右，低功耗模式不宜使用。外部上/下拉根据需要可选 10K Ω ~ 1M Ω 之间。(2) 开关量输入：一定要保证高低电压分明，理想情况下高电平就是电源电压，低电平就是地的电平。如果外部电路无法正确区分高低电平，但高低仍有较大压差，可考虑用 A/D 采集的方式设计处理。对分压方式中的采样点，要考虑分压电阻的选择，使该点通过采样端口的电流不小于采样最小输入电流，否则无法进行采样。(3) 开关量输出：基本原则是保证输出高电平接近电源电压，低电平接近地电平。I/O 口的吸纳电流一般大于放出电流，对小功率元器件控制是最好采用低电平控制的方式。一般情况下，负载要求小于 10mA 可用芯片引脚直接控制；电流在 10~100mA 时可用三极管控制，100mA~1A 时用 IC 控制；更大的电流则适合用继电器控制，同时需要使用光电隔离芯片。

4) A/D 电路与 D/A 电路

A/D 电路：要清楚前端采样基本原理，对电阻型、电流型和电压型传感器采用不同的采集电路。如果采集的信号微弱，要考虑怎样进行信号放大。D/A 电路：考虑 MCU 的引脚通过何种输出电路控制实际对象。

5) 其它

通信接口：如 USB、SCI、SPI、以太网接口等。设计接口时一定要分模块设计，各模块之间要完全独立。增加或删除某一接口时按模块取舍。键盘、LCD、LED 等注意选型。

2. 各模块的设计原则

(1) 每个模块取一个合适的名字。(2) 在文档中，对每个模块分别设计，画出原理图或框图，简明扼要写出基本原理。(3) 任何模块可看成是一个基本的输入/输出系统。(4) 使用复接点复接线方式，利用冗余提高稳定。

3. 绘制原理图的基本要求

(1) 用 A4 纸。(2) 不使用网格，底色为白色。(3) 每个模块独立画出，用虚线框框

好，分清输入输出，左侧输入右侧输出。(4) 每个模块要标注模块名。(5) MCU 最小系统不能做任何更改，各系统中的相同 MCU 的最小系统保持一致。(6) 以 MCU 为中心扩展其它模块，形成分级结构。网标命名原则是下一级硬件对象迁就上一级硬件对象，例如某一 MCU 的 PTA1 引脚控制一盏小灯 LED，应该在 LED 的引脚上标 PTA1，而不是在 PTA1 引脚上标 LED。(7) 对外接口的设计，尽可能采用防止反插的接口。(8) 设计时，每个电路板都应该设计有电源指示灯、运行指示灯、故障指示灯等来表达电路的运行状态。实际使用时，若确实不需要这些指示灯，可以不焊接。(9) 文字标注可使用五号或小五号字体，电路原理图中的说明使用汉字。(10) 电路绘制完成后，不要忘记要填写版权框中的有关信息。

2.2 电路板布板基本要求

1. 布局

在正式走线之前要对 PCB 的大体格局进行规划，布局规划基本原则：

- (1) 在 PCB 布板之前首先要打印出相应的原理图，然后根据原理图确定整个 PCB 板的大体布局即各个模块的位置安排；
- (2) PCB 板的形状如无其他要求，一般为矩形，长宽比为 4:3 或 3:2；
- (3) 考虑面板上元件的放置要求；
- (4) 考虑边缘接口。

2. 元件放置

- (1) 芯片尽可能正放；
- (2) 元件放置要求整齐，各模块内同类元件尽可能排放在一起；
- (3) 电容的位置要特别注意，其中电源模块的滤波电容要求靠近电源，而 IC 的滤波电容要靠近 IC 的引脚；
- (4) 考虑元件间的距离，防止元件之间出现重叠。

3. 有关设定

在正式布线之前，需要对 PROTEL 的一些属性进行设置：

- (1) 线宽。导线尽可能宽，这样既可以减小阻抗，又可以防止由于制造工艺的原因导致导线断路。电源、地线的宽度要大于普通信号线。三者关系为地线>电源线>信号线。
- (2) 间距。导线间距离以及导线与元件间距离要尽可能的大，这样可以有效解决焊接时短路的问题。
- (3) 过孔大小。过孔大小设定要适中。

4. 测量点

考虑到硬件测试的方便，在 PCB 布板时要留下一些测量点，以便调试之用。测量点要根据原理图确定。以下几处需要留测量点：

- (1) 原理图中模块的输入输出引脚；
- (2) 最小系统模块中 MCU 的引脚；
- (3) 各硬件模块单元的输入、输出处；

画测量点的要求：“引出、打孔、标字”方式。孔的大小以万用表头好测量为原则。不要在线上直接打孔。记住一定要标字。

5. 模块标示

由于在整体布局的时候基本将各个模块的元件放在一起，因而可以在 PCB 板上将各个模块区分开，并用汉字标出模块名（与原理图一致）。注意字体字号。

6. 空余位置的利用

PCB 板的空余位置可作以下之用：

- (1) 电源、地。空白地多留几排电源和地。

-
- (2) 双排孔。留出几排两孔相连的排孔，以用来扩展或试验时焊接其他元件。
 - (3) 在 PCB 上画固定板的固定孔，一般在板的四个角落。

7. 铺地

铺地可以减小干扰，提高 PCB 板的稳定性，因此在布板的最后都要铺地。以下几点为铺地注意点：

- (1) 在铺地前，要设定地与导线、地与引脚之间的距离，并要求该距离尽可能大；
- (2) 铺地本应该双面铺，作为实验用板，为了方便检查，可只铺反面地；
- (3) 如果电路板中有数字地和模拟地，应隔离开，两者间使用磁珠相连。

8. 标注相关信息

在完成 PCB 板的铺地之后，要在板的正面适当位置标出以下几点：

- (1) 单位；
- (2) 日期；
- (3) 责任人；
- (4) PCB 板的名称；
- (5) 编号。如 2008-XX。

9. 检查

布板完成后一定要进行自动与人工检查。

10. 其他补充

- (1) 过孔数尽可能少。
- (2) 过孔避免放在焊盘上，以免漏锡导致虚焊。
- (3) 最小系统中未使用的 I/O 口，可通过电阻接地。
- (4) 走线注意点。元件的排列方位尽可能保持与原理图相一致，布线方向最好与电路图走线方向相一致。要尽量避免长距离平行布线，电路中电气互连点间布线力求最短。设计布线图时，走线尽量少拐弯，印刷弧上的线宽不要突变，导线拐角应 $\geq 90^\circ$ ，力求线条简单明了。信号线的拐角应设计成钝角走向，或成圆形、圆弧形，切忌画成 90° 或更小角度形状。印刷电路中不允许有交叉电路，对于可能交叉的线条，可以用“钻”、“绕”两种办法解决。即让某引线从别的电阻、电容、三极管脚下的空隙处“钻”过去，或从可能交叉的某条引线的一端“绕”过去。晶振电路要注意外界的干扰，晶振下方尽量不走线。

2.3 基于嵌入式硬件构件模型的硬件构件设计要求

1. 硬件构件定义及分类

定义：嵌入式硬件构件是将一个或多个硬件功能模块、支撑电路及其功能描述封装成一个可重用的硬件实体，并提供一系列规范的输入/输出接口。

分类：从“服务”的角度，硬件构件分为核心构件、中间构件和终端构件三种类型。

2. 硬件构件设计通用要求

- (1) **图纸设置**：大小：A4，底色：白色，字体：五号，网格捕获：10
- (2) **元器件命名格式**：构件名-元器件名，对同种类型的元件命名时冠以相同前缀。
- (3) **连线要求**：从元件端点处连线，一线连通，禁止直接进行端点对接。
- (4) **构件描述包括**：中文名称、英文名称、功能描述、注意事项等。
- (5) **封装方法**：用矩形虚线框将构件实体和构件描述封装起来。
- (6) **接口注释命名格式**：构件名-注释名，位于矩形虚线框内，为斜体字。
- (7) **接口网标命名格式**：构件名-网标名，位于矩形虚线框外。
- (8) 电源符号 V_{CC} 向上，地线符号 \perp 向下。

3. 核心构件设计要求

核心构件是 MCU 的最小系统，设计时要求：

- (1) 将所有可能使用到的引脚都标注上接口网标，位于矩形虚线框外，名称为引脚名。
- (2) 对于复用引脚，接口网标依据第一功能选项命名。
- (3) 实体内需包括编程接口电路。

4. 中间构件设计要求

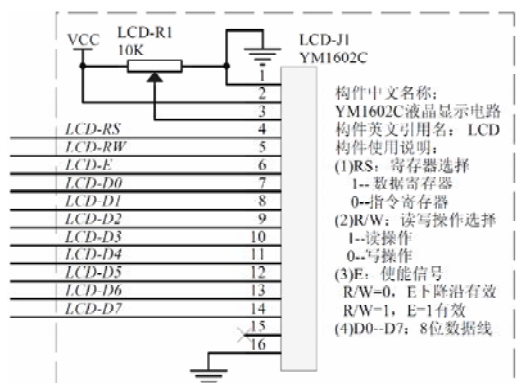
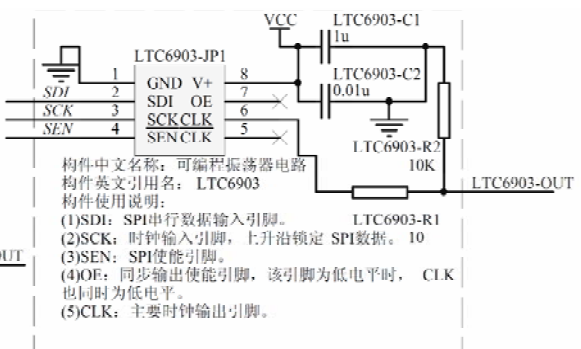
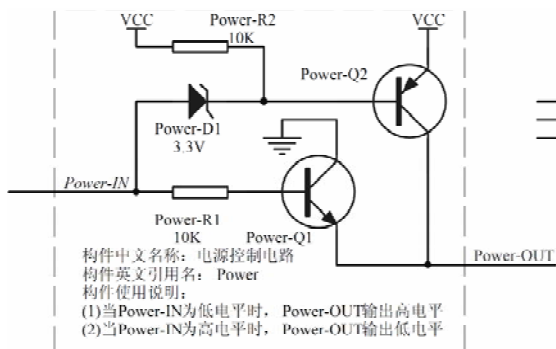
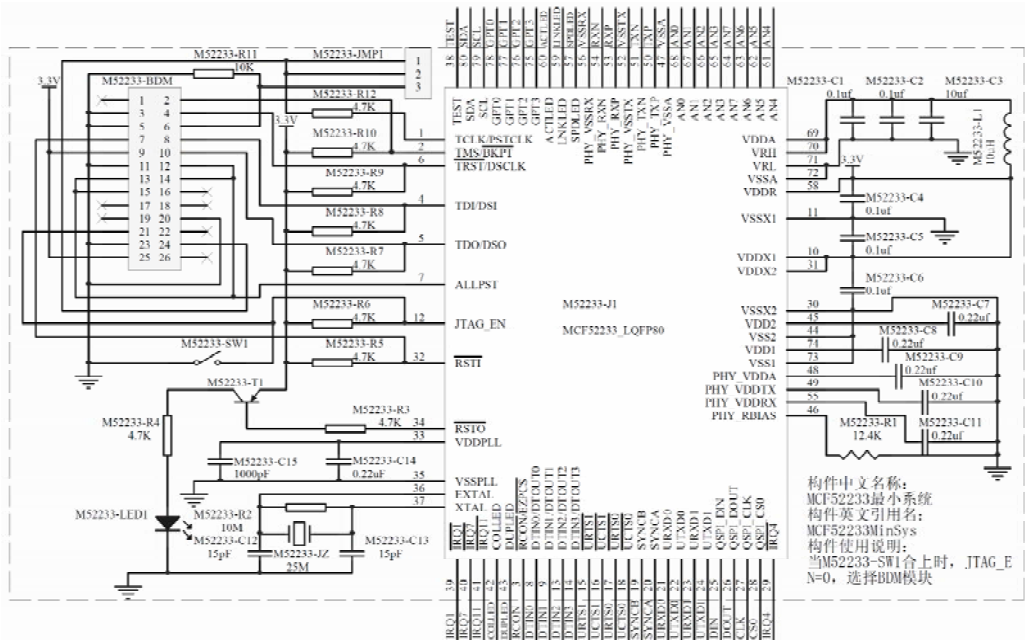
中间构件是既有需求接口（接受服务）又有提供接口（提供服务）的硬件构件，设计时要求：

- (1) 需求接口采用接口注释，位于矩形虚线框内侧、左边。
- (2) 提供接口采用接口网标，位于矩形虚线框外侧、右边。

5. 终端构件设计要求

终端构件是仅有需求接口的硬件构件，设计时要求：需求接口采用接口注释，位于矩形虚线框内侧、左边。

6. 示例



2.4 基于嵌入式硬件构件模型的软件构件设计要求

1. 底层构件设计要求

底层构件有两个基本元素：描述和实体。其中，描述可看作是底层构件的使用说明书，详细介绍构件的功能、移植时需修改的关键参数以及注意事项等。由于具有与硬件密切相关的特性，底层构件的实体可看作是属性和操作的集合。属性用来描述底层构件的特性，如端口属性、硬件接口属性等。操作是构件可以执行的动作，如硬件的初始化、发送数据、接收数据、中断处理等。属性描述可采用宏定义形式，而操作通过内部函数和外部函数体现。

从文件角度来看，构件包含两个文件，即头文件（.h）和源文件（.c），它们的主文件名相同。头文件中包含下层构件头文件（一级底层构件除外）、属性宏定义以及外部函数说明。构件的源文件中主要包含自身构件头文件、内部函数定义以及外部函数定义。

具体要求如下：

- (1) 头文件和程序文件的主文件名一致，且为构件名。
- (2) 对外接口函数命名格式：构件名_功能名，如 LCD_Init
- (3) 宏名命名格式：以“构件名_”开头
- (4) 重定义：若构件使用到 MCU 头文件中定义的模块寄存器名，如：

```
#define SCDR *(volatile unsigned char *)0x18 //SCI 数据寄存器
```

则在构件头文件中重定义宏名，如：

```
#define SCI_ReSendDataR SCDR //重定义 SCI 数据寄存器
```

(5) 头文件中内容次序：①文件描述②包含文件③宏定义④对外接口函数原型说明。为防止重复定义，使用条件编译语句，形如：

```
#ifndef 构件名_H  
#define 构件名_H  
.....  
#endif
```

(6) 程序文件中内容次序：①文件描述②内部函数定义③外部函数定义

(7) 函数描述：在头文件中进行对外接口函数原型说明时，或者在程序文件中进行内/外部函数定义时，都要给出函数描述，内容包括函数名、功能、参数、返回值和说明共 5 项。

(8) 严格禁止通过全局变量来传递参数。

2. 底层构件封装要完整、封装之间不能交叉

在对底层构件封装时，首先通过分析并抽取出底层构件的属性和操作，尽量做到当该构件被应用到具体一个系统中时，用户需修改的条目要尽可能的少。一个构件的操作是与上层构件进行交互的接口，上层构件不能直接对该构件的属性进行读取或设置，必须借助于该构件提供的接口操作函数来实现。对于上层构件而言，只需知道操作函数的功能和调用方法，而不必关心该操作函数具体实现的过程。为避免重复，属性和操作的命名统一以“构件名_”开头。接着编写构件描述，构件描述包含构件对外接口函数的使用方法、注意事项以及移植时所修改的项目等等。

构件内部变量，外部不能调用。同一层次的构件之间不允许相互调用。不同层次的构件之间虽然允许调用，但只能是高层构件调用低层构件，并且能够越层调用。上层构件在使用底层构件时，严格禁止通过全局变量来传递参数，所有的数据传递都直接通过函数的形式参

数来接收。

3. 严格区分内部函数和外部函数

内部函数仅提供给同一文件中的其它内部函数或外部函数调用，作用域仅限于定义该函数的文件。内部函数主要是与对 MCU 操作，应放入构件的源文件中，而不放入构件的头文件中。头文件中需说明有关宏定义和构件被调用时要修改的内容。外部函数是对外接口函数，其原型说明需列在构件的头文件中，以便供其他构件的函数调用。当构件 B 需要使用构件 A 的宏定义或调用构件 A 的外部函数时，只要在构件 B 的头文件中包含构件 A 的头文件即可。在定义外部函数时，应对函数名、函数功能、入口参数、函数返回值、使用说明、函数适用范围以及修改说明等进行详细描述，以增强程序的可读性。

4. 头文件的设计内容

在头文件中要声明变量类型，对于类型为整型的变量要给出取值范围。每个函数的参数类型及其含义要说清楚。头文件中也要对函数调用时需修改的内容做详细说明。

5. 底层构件的移植

在设计底层构件时，构件描述应详细说明在特定的条件下使用该构件时是否需要修改相关参数，若要修改，具体又有哪些。这样，用户在使用该构件时，只要根据构件描述的内容对头文件中的相关参数进行配置即可，构件的源文件不必作任何改动。

在对底层构件进行移植时，大多数情况下只需拷贝头文件和源文件即可，无需修改，只有当实施不同芯片之间的移植或硬件连接发生改变时，才需要修改头文件中与硬件相关的宏定义。

6. 注释

注释不是简单的说明语句，而是要解释该语句的作用。整行注释需要标明序号。

7. 功能性函数

功能性函数不作为构件使用，该函数在主程序中写出。

8. 通用函数

通用函数与硬件无关，该函数也包含头文件和源文件。

9. 测试

测试是当模块编写完成后对其功能进行的测试，先进行白盒测试，然后进行黑盒测试，测试要测试模块所有可能的输入，对非法的输入进行检测，一定测试全面。底层软件构件做完后，编写测试程序。测试通过后，写明构件测试的过程和测试结果。

2.5 硬件驱动模块的测试要求

当通过面向硬件对象的封装技术进行了底层驱动模块设计后,需要进入的下一步工作即是对各个驱动模块的测试,本节主要内容为硬件驱动模块的测试技巧。

1. 一定要测试,而且要保留测试过程

对于已经设计好的驱动模块,必须要经过严格的测试后才能使用;同时,测试过程要以目录的形式保留下来,用于备案。

2. 对于有可视执行结果的模块要先测试

可视执行结果是指能通过小灯的亮灭,LED 或串口等设备去显示的执行结果。对于有可视结果的模块先行测试,主要是用于具有不可视执行结果模块的测试。

当评估板上对应没有的可视化执行设备(如大电机)时,可以用小灯来等效处理。因为本质上都是开关量的输出,只要输出的开关量正常即可证明模块是正确的。

3. 测试用例要尽可能覆盖

测试只能证明错误的存在,而不能证明错误的不存在。

测试用例至少要包括边界值和中间值,合法值和非法值,要保证尽可能多的情况在时间和空间上被覆盖。

例如:对三盏小灯驱动模块的测试,必须有每一盏的亮和灭,三盏全亮,全灭,三盏逐渐全亮,三盏逐渐全灭等测试用例。

4. 对于无可视执行结果的模块的测试

对于无可视化执行结果的模块,特别是把数据从外设硬件传到内存的输入模块,由于无法观测模块的执行结果,必须要借助于具有可视执行结果的模块来显示。这也是可视执行结果的模块要先行测试的原因。

注意,由于借助了具有可视执行结果的模块,就有可能发现该模块有问题,此时必须要返回到有问题模块的测试目录,找出问题,重新生成新的驱动模块,然后再拿过来使用。而不能简单的直接在无可视执行结果的模块的测试中直接修改,因为这将违反了面向硬件对象的驱动封装原则。

5. 测试工程中的说明

用于测试驱动模块的工程中,必须要有一个用于记录测试信息的文本文件。记录的内容包括测试人,测试时间,测试结果及意见等相关信息。如果测试信息的内容比较多,也可以在测试目录中专门建一个 WORD 文档,用于进一步的说明测试信息。

6. 编写测试工程的注意事项

- (1) 在测试工程中调用测试模块时,最好把测试用例以常量形式传入测试模块。
- (2) 由于主循环或定时中断能重复多次执行,所以在有很多测试用例的情况下,可以通过把测试用例放到静态变量中,然后不断修改静态变量,实现了自动对测试用例的覆盖测试。
- (3) 使用辅助测试工具:如果 USB、串口或其他通讯模块已经被正确的测试了,可以把测试结果发送到 PC 机进行显示。

