

第 3 章



硬件基础知识学习

通过第 2 章的学习,我们成功地点亮了一个 LED 小灯,但是有些知识读者还没彻底搞明白。单片机是根据硬件电路图的设计编写代码的,所以不仅要学习编程知识,还要学习基本的硬件知识,本章就来穿插介绍电路硬件知识。

3.1 电磁干扰

首先介绍下去耦电容的应用背景,这个背景就是电磁干扰(Electromagnetic Interference, EMI)。下面是一些电磁干扰的现象。

(1) 冬天的时候,尤其是在空气比较干燥的内陆城市,很多朋友都有这样的经历,手触碰到计算机外壳、铁柜子等物品的时候会被电击,这就是“静电放电”现象,也称为 ESD。

(2) 不知道读者有没有这样的经历,早期使用电钻这种电机设备,并且同时在听收音机或者看电视的时候,收音机或者电视会出现杂音,这就是“快速瞬间群脉冲”的效果,也称为 EFT(Electrical Fast Transient)。

(3) 以前的老计算机,有的性能不是很好,带电热插拔优盘、移动硬盘等外围设备的时候,内部会产生一个百万分之一秒的电源切换,直接导致计算机出现蓝屏或者重启现象,就是热插拔的“浪涌”效果,称为 Surge。

电磁干扰的内容有很多,这里不一一列举,但是有些内容非常重要,后边要一点点地了解。这些问题读者不要认为是小问题,比如一个简单的静电放电,用手能感觉到的静电,可能已经达到 3kV 以上了,如果用眼睛能看得到的,至少是 5kV 了,只是因为这个电压虽然很高,能量却非常小,持续的时间非常短,因此不会对人体造成伤害。但是应用这些半导体元器件就不一样了,一旦瞬间电压过高,就有可能造成器件的损坏,即使不损坏,在上面介绍的(2)、(3)两种现象,也已经严重干扰到设备的正常使用了。

基于以上的这些问题,就诞生了电磁兼容(Electromagnetic Compatibility, EMC)这个名词。本章仅仅讲下去耦电容的应用,电磁兼容的处理在今后设计电路,对 PCB 布局中应用尤为重要。

3.2 去耦电容的应用

首先来看图 3-1,前面已经见过 USB 接口和供电电路。

在图 3-1(a)中,过了保险丝以后,接了一个 470 μ F 的电容 C16,在图 3-1(b)中,经过开



教学视频

关后,接了一个 $100\mu\text{F}$ 的电容 C19,并且并联了一个 $0.1\mu\text{F}$ 的电容 C10。其中 C16 和 C19 起到的作用是一样的,C10 的作用和它们两个不一样,先来介绍这两个大一点的电容。

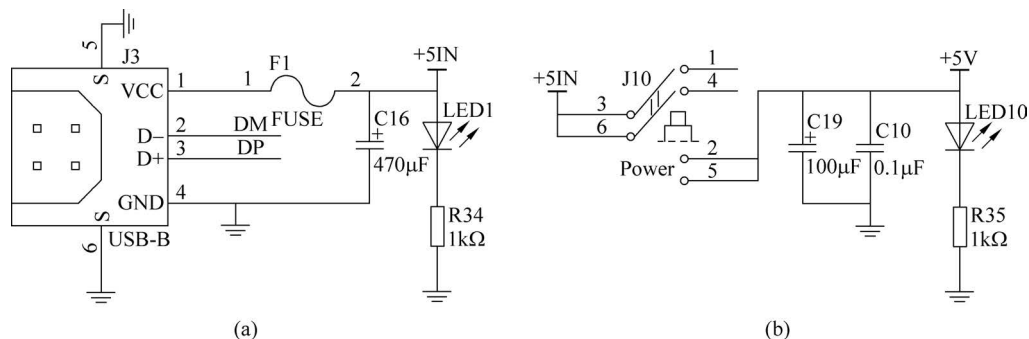


图 3-1 USB 接口和供电电路

容值比较大的电容,理论上可以理解成水缸或者水池子,同时,读者可以直接把电流理解成水流,其实大自然万物的原理都是类似的。

(1) 缓冲作用。当上电的瞬间,电流从电源处流下来的时候,不稳定,容易冲击电子器件,加个电容可以起到缓冲作用。就如同直接用水龙头的水浇地,容易冲坏花花草草。只需要在水龙头处加个水池,让水经过水池后再缓慢流进草地,就不会冲坏花草,起到有效的保护作用。

(2) 稳定作用。电路中后级电子器件的功率大小都不一样,而器件正常工作的时候,所需电流的大小也不是一成不变的。比如后级有个器件还没有工作的时候,电流消耗是 100mA ,突然它参与工作了,电流猛地增大到了 150mA ,这个时候如果没有一个水缸,电路中的电压(水位)就会直接突然下降,比如 5V 电压突然降低到 3V 了。而系统中有些电子元件,必须高于一定的电压才能正常工作,电压太低就直接不工作了,这个时候水缸就必不可少。电容会在这个时候把存储在里边的电量释放一下,稳定电压,当然,随后前级的电流会及时把水缸充满的。

有了这个电容,可以说电压和电流就会很稳定了,不会产生大的波动。这种常用的电容如图 3-2~图 3-4 所示。



图 3-2 铝电解电容

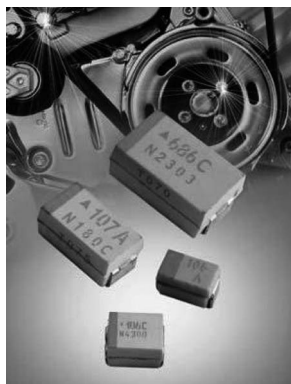


图 3-3 钽电容

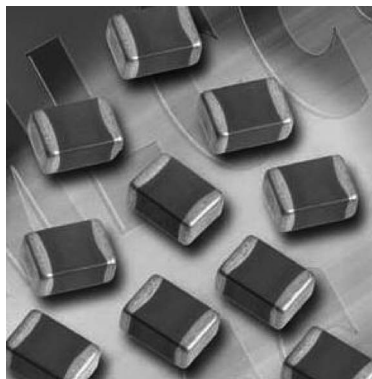


图 3-4 陶瓷电容

这三种电容是最常用的。其中第一种个头大,占空间大,单位容量价格最便宜,第二种和第三种个头小,占空间小,性能一般略好于第一种,但是价格贵不少。当然,除了价格,还有一些特殊参数,在通信要求高的场合也要考虑很多,这里暂且不说。开发板上现在用的是第一种,在同样的符合条件的耐压值和容值下,第一种 $470\mu\text{F}$ 的电容价格不到一角钱,而第二种和第三种可能要一元钱左右了。

电容的选取,第一个参数是耐压值的考虑。我们用的是 5V 系统,电容的耐压值要高于 5V ,一般推荐 $1.5\sim 2$ 倍即可,有些场合再稍微高点也可以。我们开发板上用的是 10V 耐压的。第二个参数是电容容值,这个就需要根据经验选取了,选取的时候,要看这个电容起作用的整套系统的功率消耗情况,如果系统耗电较大,波动可能比较大,那么容值就要选大一些,反之可以小一些。

刚开始设计电路也是要模仿的,其他人用多大的电容,自己也用多大,慢慢积累。比如上边讲电容稳定作用的时候,电流从 100mA 突然增大到 150mA 的时候,其实即使加上这个电容,电压也会轻微波动,比如从 5V 波动到 4.9V ,但是只要开发板上的器件在电压 4.9V 以上也可以正常工作,这点波动是被容许的,如果不加或者加得很小,电压波动比较大,有些器件的工作就会不正常了。如果加得太大,占空间并且价格也高,所以这个地方电容的选取多参考经验。

再来看图 3-1 中的另一种电容 C_{10} ,它的容值较小,是 $0.1\mu\text{F}$,也就是 100nF ,是用来滤除高频信号干扰的,比如 ESD、EFT 等。初中学过电容的特性——可以通交流隔直流,但是电容的参数对不同频率段的干扰的作用是不一样的。这个 100nF 的电容,是前辈们根据干扰的频率段、开发板的参数、电容本身的参数所总结出来的一个值。也就是说,以后大家在设计数字电路的时候,在电源处去耦高频电容,直接用这个 $0.1\mu\text{F}$ 就可以了,不需要再去计算和考量太多。

还有一点,读者可以仔细观察 KST-51 开发板,在电路中需要较大电流供给的器件附近,会加一个大电容,比如在 1602 液晶显示器左上角的 C_{18} ,靠近单片机的 VCC 以及 1602 液晶显示器背光的 VCC,起到稳定电压的作用,而图 3-1 中的 C_{19} 的实际位置也是放在了左上角电机和蜂鸣器附近,因为它们所需的电流都比较大,而且工作时电流的波动也很大。还有,在所有的 IC 器件的 VCC 和 GND 之间,都会放一个 $0.1\mu\text{F}$ 的高频去耦电容,特别在布板的时候, $0.1\mu\text{F}$ 电容要尽可能地靠近 IC,尽量很顺利地 IC 的 VCC 和 GND 连到一起,这个读者先了解,细节以后再讨论。

3.3 三极管在数字电路中的应用

三极管在数字电路和模拟电路中都有大量的应用,在 KST-51 开发板上也用了多个三极管。在我们学习的开发板上的 LED 小灯部分,就有三极管的应用,图 3-5 的 LED 电路中的 Q16 就是一个 PNP 型的三极管。

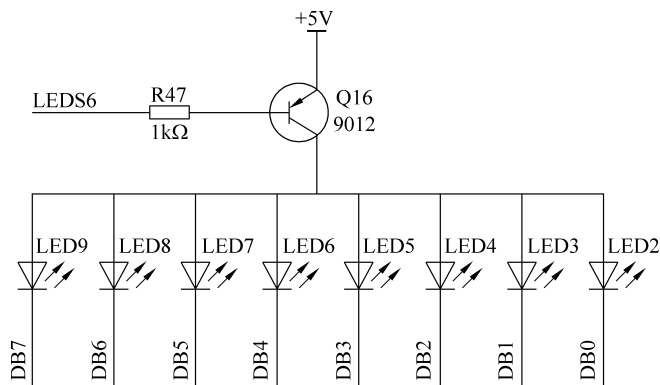


图 3-5 LED 电路

3.3.1 三极管的初步认识

三极管是一种很常用的控制和驱动器件,常用的三极管根据材料分有硅管和锗管两种,它们的原理相同,压降略有不同,硅管用得较普遍,而锗管应用较少。本书就用硅管的参数进行讲解。三极管有两种类型,分别是 PNP 型和 NPN 型,如图 3-6 所示。

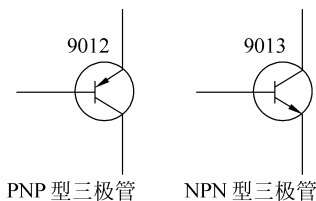


图 3-6 三极管示意图

三极管一共有 3 个极,从图 3-6 来看,横向左侧的引脚叫作基极(base,简称 b 极),中间有一个箭头,一头连接基极,另一头连接的是发射极(emitter,简称 e 极),那剩下的一个引脚就是集电极(collector,简称 c 极)了。这是必须记住的内容,死记硬背即可,后边慢慢用得多了,死记硬背多次以后就会深入脑海了。

3.3.2 三极管的原理

三极管有截止、放大、饱和三种工作状态。放大状态主要应用于模拟电路中,且用法和计算方法也比较复杂,暂时用不到。而数字电路主要使用的是三极管的开关特性,只用到了截止与饱和两种状态,所以只讲解这两种用法。三极管的类型和用法作者总结了一句口诀,读者要把这句口诀记牢了:箭头朝内 PNP,导通电压顺着箭头过,电压导通,电流控制。

下面来一句一句解析口诀。读者可以看图 3-6,三极管有两种类型,箭头朝内就是 PNP,那箭头朝外的自然就是 NPN 了,在实际应用中,要根据实际电路的需求选择到底用哪种类型,读者多用几次也就懂了,很简单。

三极管的用法特点,关键点在于 b 极(基极)和 e 极(发射极)之间的电压情况,对于 PNP 而言,e 极电压只要高于 b 极 0.7V 以上,这个三极管 e 极和 c 极之间就可以顺利导通。也就是说,控制端在 b 极和 e 极之间,被控制端是 e 极和 c 极之间。同理,NPN 型三极管的导通电压是 b 极比 e 极高 0.7V,总之是箭头的始端比末端高 0.7V 就可以导通三极管的 e 极和 c 极。这就是关于“导通电压顺着箭头过,电压导通”的解释。

以图 3-7 为例,三极管基极通过一个 10kΩ 的电阻接到了单片机的一个 I/O 口上,假定是 P1.0,发射极直接接到 5V 的电源上,集电极接了一个 LED 小灯,并且串联了一个 1kΩ 的限流电阻,最终接到了电源负极 GND 上。

如果 P1.0 由程序给一个高电平 1, 那么基极 b 和发射极 e 都是 5V, 也就是说 e 极到 b 极不会产生一个 0.7V 的压降, 这个时候, 发射极和集电极也就不会导通, 那么竖着看这个电路在三极管处是断开的, 没有电流通过, LED2 小灯也就不会亮。如果程序给 P1.0 一个低电平 0, 这时 e 极还是 5V, 于是 e 极和 b 极之间产生了压差, 三极管 e 极和 b 极之间也就导通了, 三极管 e 极和 b 极之间大概有 0.7V 的压降, 那还有 $(5 - 0.7)V$ 的电压会在电阻 R47 上。这个时候, e 极和 c 极之间也会导通了, 那么 LED 小灯本身有 2V 的压降, 三极管本身 e 极和 c 极之间大概有 0.2V 的压降, 忽略不计。那么在 R41 上就会有大概 3V 的压降, 可以计算出来, 这条支路的电流大概是 3mA, 可以成功点亮 LED。

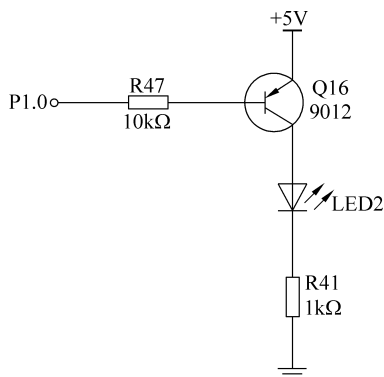


图 3-7 三极管的用法

最后一个概念, 电流控制。前边讲过, 三极管有截止、放大、饱和三个状态, 截止就不用说了, 只要 e 极和 b 极之间不导通即可。要让这个三极管处于饱和状态, 就是所谓的开关特性, 必须满足一个条件。三极管都有一个放大倍数 β , 要想处于饱和状态, b 极电流就必须大于 e 极和 c 极之间电流值除以 β 。这个 β 对于常用的三极管大概可以认为是 100。那么上边的 R47 的阻值必须要计算一下了。

刚才算过了, e 极和 c 极之间的电流是 3mA, 那么 b 极电流最小就是 3mA 除以 100 等于 $30\mu A$, 大概有 4.3V 电压会落在基极电阻上, 那么基极电阻最大值就是 $4.3V / 30\mu A = 143k\Omega$ 。电阻值只要比这个值小就可以, 当然也不能太小, 太小会导致单片机的 I/O 口电流过大烧坏三极管或者单片机, STC89C52 的 I/O 口输入电流最大理论值是 25mA, 推荐不要超过 6mA, 用电压和电流算一下就可以算出最小电阻值, 图 3-7 取的是经验值。

3.3.3 三极管的应用

三极管在数字电路里的开关特性最常见的应用有两个: 一个是控制应用, 另一个是驱动应用。所谓控制就是如图 3-7 所示, 可以通过单片机控制三极管的基极间接控制后边小灯的亮灭, 用法读者基本熟悉了。还有一个控制就是进行不同电压之间的转换控制, 比如单片机是 5V 系统, 它现在要跟一个 12V 的系统对接, 如果 I/O 直接接 12V 电压就会烧坏单片机, 所以加一个三极管, 三极管的工作电压高于单片机的 I/O 口电压, 用 5V 的 I/O 口来控制 12V 的电路, 如图 3-8 所示。

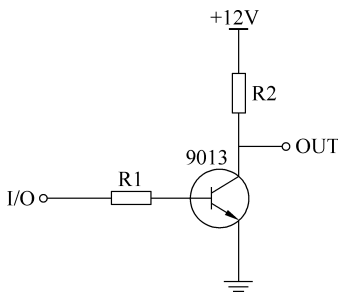


图 3-8 三极管实现电压转换

图 3-8 中, 当 I/O 口输出高电平 5V 时, 三极管导通, OUT 输出低电平 0V, 当 I/O 口输出低电平时, 三极管截止, OUT 则由于上拉电阻 R2 的作用而输出 12V 的高电平, 这样就实现了低电压控制高电压的工作原理。

所谓的驱动主要是指电流输出能力。再来看如图 3-9 所示的两个电路之间的对比。

图 3-9 中上边的 LED 灯, 和前面讲过的 LED 灯是一样的, 当 I/O 口是高电平时, 小灯熄灭, 当 I/O 口是低电平时,

小灯点亮。那么下边的电路呢,按照这种推理,I/O口是高电平的时候,应该有电流流过并且点亮小灯,但实际上却并非这么简单。

单片机主要是个控制器件,具有四两拨千斤的特点。就如同杠杆必须有一个支点一样,想要撑起整个地球必须有力量承受的支点。单片机的 I/O 口可以输出一个高电平,但是它的输出电流却很有限,普通 I/O 口输出高电平的时候,大概只有几十到几百 μA 的电流,达不到 1mA ,也就点不亮这个 LED 小灯或者是亮度很低,这时如果用高电平点亮 LED,就可以用三极管来处理了,开发板上的这种三极管型号,可以通过 500mA 的电流,有的三极管通过的电流还更大一些,如图 3-10 所示。

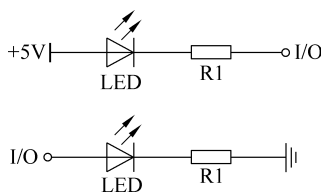


图 3-9 LED 小灯控制方式对比

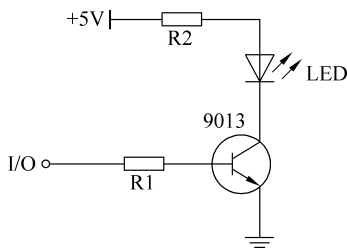


图 3-10 三极管驱动 LED 小灯

图 3-10 中,当 I/O 口是高电平,三极管导通,因为三极管的电流放大作用,c 极电流就可以达到 mA 以上了,就可以成功点亮 LED 小灯。

虽然用 I/O 口的低电平可以直接点亮 LED,但是单片机的 I/O 口作为低电平,输入电流就可以很大吗?这个读者都能猜出来,当然不可以。单片机的 I/O 口电流承受能力,不同型号单片机不完全一样,就 STC89C52 来说,官方手册的 81 页有对电气特性的介绍,整个单片机的工作电流,不要超过 50mA ,单个 I/O 口总电流不要超过 6mA 。即使一些增强型 51 的 I/O 口承受电流大一点,可以到 25mA ,但是还要受到总电流 50mA 的限制。下面来看 8 个 LED 小灯这部分电路,如图 3-11 所示。

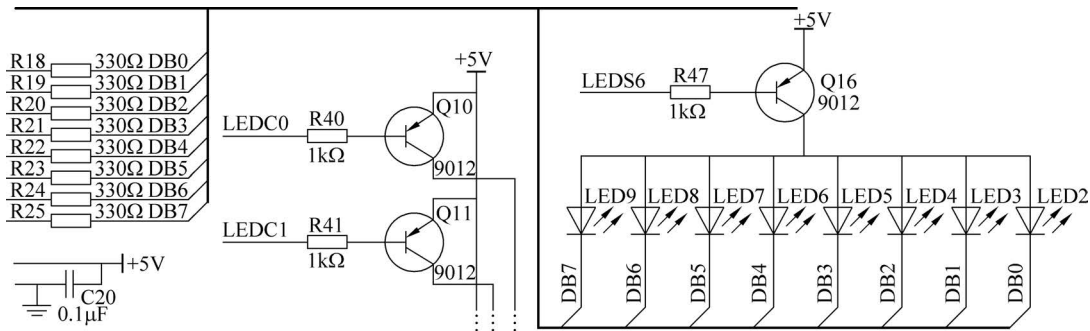


图 3-11 LED 电路图(一)

这里要学会看电路图的一个知识点,电路图右侧所有的 LED 下侧的线最终都连到一根黑色的粗线上去了,大家注意,这个地方不是实际的完全连到一起,而是一种总线的画法,画了这种线以后,表示这是个总线结构。而所有的名字一样的节点是一一对应地连接到一起,其他名字不一样的,是不连在一起的。比如左侧的 DB0 和右侧的最右边的 LED2 小灯下边的 DB0 是连在一起的,而和 DB1 等其他线不是连在一起的。

把图 3-11 中现在需要讲解的这部分电路单独摘出来,如图 3-12 所示。

现在通过 3-12 的电路图计算,5V 的电压减去 LED 本身的压降,减掉三极管 e 极和 c 极之间的压降,限流电阻用的是 330Ω ,那么每条支路的电流大概是 8mA,那么 8 路 LED 如果全部同时点亮的话,电流总和就是 64mA。这样如果直接接到单片机的 I/O 口,那单片机肯定是承受不了的,即使短时间可以承受,长时间工作就会不稳定,甚至导致单片机烧毁。

有的用户会提出来可以加大限流电阻的方式降低电流。比如限流电阻改到 $1k\Omega$,那么电流不到 3mA,8 路总的电流就是 20mA 左右。首先,降低电流会导致 LED 小灯亮度变暗,小灯的亮度可能关系还不大,但因为同样的电路接了数码管,后边数码管还要动态显示,如果数码管亮度不够,那视觉效果就会很差,所以降低电流的方法并不可取。其次,对于单片机来说,它主要起到控制作用,电流输入和输出的能力相对较弱,P0 的 8 个口总电流也

有一定限制,所以如果接一两个 LED 小灯观察,可以勉强直接用单片机的 I/O 口来接,但是接多个小灯,从实际工程的角度考虑,就不推荐直接接 I/O 口了。如果要用单片机控制多个 LED 小灯该怎么办呢?

除了三极管之外,其实还有一些驱动 IC,这些驱动 IC 可以作为单片机的缓冲器,仅仅是电流驱动缓冲,不起到任何逻辑控制的效果,比如开发板上用的 74HC245 芯片,这个芯片在逻辑上起不到什么别的作用,就是当作电流缓冲器,通过查看其数据手册,74HC245 稳定工作在 70mA 电流是没有问题的,比单片机的 8 个 I/O 口大多了,所以可以把它接在小灯和 I/O 口之间做缓冲,如图 3-13 所示。

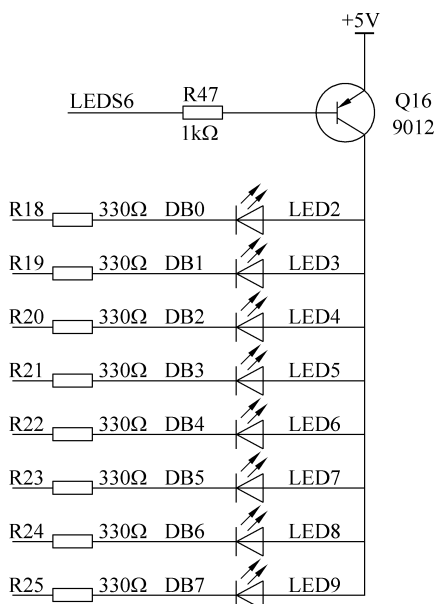


图 3-12 LED 电路图(二)

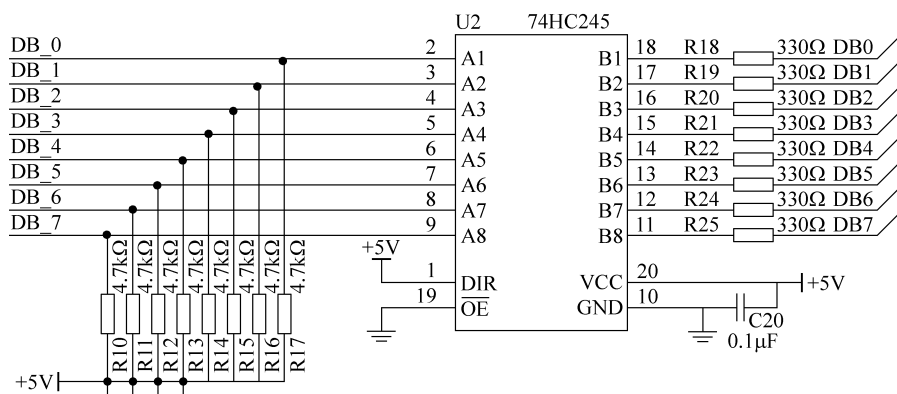


图 3-13 74HC245 功能图

从图 3-13 分析,其中 VCC 和 GND 就不用多说了,细心的读者会发现这里有个 $0.1\mu\text{F}$ 的去耦电容。

74HC245 是个双向缓冲器,1 引脚 DIR 是方向引脚,当这个引脚接高电平的时候,右侧

所有的 B 编号的电压都等于左侧 A 编号对应的电压。比如 A1 是高电平,那么 B1 就是高电平,A2 是低电平,B2 就是低电平等。如果 DIR 引脚接低电平,得到的效果是左侧 A 编号的电压都会等于右侧 B 编号对应的电压。因为这个地方控制端是左侧接的是 P0 口,要求 B 等于 A 的状态,所以 1 脚直接接的 5V 电源,即高电平。图 3-13 中还有一排电阻 R10~R17 是上拉电阻,这个电阻的用法将在后边介绍。

还有最后一个使能引脚,即 19 引脚 $\overline{\text{OE}}$,叫作输出使能,这个引脚上边有一横,表明是低电平有效,当接了低电平后,74HC245 就会按照刚才上边说的起到双向缓冲器的作用,如果 $\overline{\text{OE}}$ 接了高电平,那么无论 DIR 怎么接,A 和 B 的引脚是没有关系的,也就是 74HC245 功能不能实现出来。

从图 3-14 可以看出来,单片机的 P0 口和 74HC245 的 A 端是直接接起来的。有读者可能有疑问,就是明明在电源 VCC 处加了一个三极管驱动了,为何还要再加 74HC245 驱动芯片呢?这里读者要理解一些道理:首先电路从正极经过器件到地,首先必须有电流才能正常工作,电路中任何一个位置断开,都不会有电流,器件也就不会参与工作了;其次,电流和水流一样,从电源正极到负极的电流管的粗细都要满足要求,任何一个位置的管子过细,都会出现瓶颈效应,电流在整个通路中的细管处会受到限制而降低,所以在电路通路的每个位置上,都要保证通道足够畅通,这个 74HC245 的作用就是消除单片机 I/O 这一环节的瓶颈。

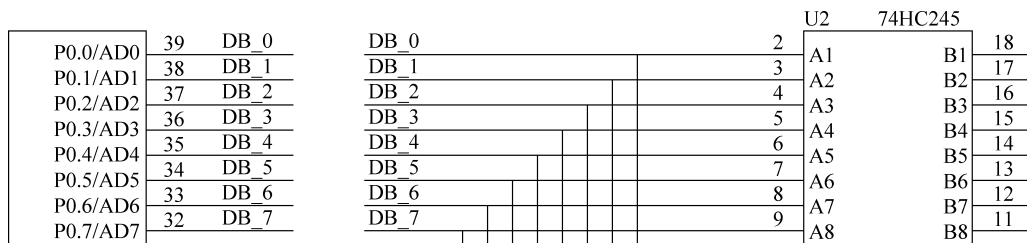


图 3-14 单片机与 74HC245 的连接

3.4 74HC138 三八译码器的应用

在设计单片机电路的时候,单片机的 I/O 口数量是有限的,有时满足不了设计需求,比如 STC89C52 一共有 32 个 I/O 口,但是为了控制更多的器件,就要使用一些外围的数字芯片,比如 74HC138 三八译码器,这种数字芯片由简单的输入逻辑控制输出逻辑,图 3-15 是 74HC138 应用原理图。

从名字分析,三八译码器就是把 3 种输入状态翻译成 8 种输出状态。从图 3-15 能看出来,74HC138 有 1~6 共 6 个输入引脚,但是其中 4、5、6 这 3 个引脚是使能引脚。使能引脚和前边讲的 74HC245 $\overline{\text{OE}}$ 引脚是一样的,这 3 个引脚如果不符合规定的输入要求,不管输入的 1、2、3 引脚是什么电平状态,Y0~Y7 总是高电平。所以要想让这个 74HC138 正常工作,ENLED 必须输入低电平,ADDR3 必须输入高电平,这两个位置都是使能控制端口。不知道读者是否记得在前面的程序有这么两句:

```
ENLED = 0; ADDR3 = 1;
```

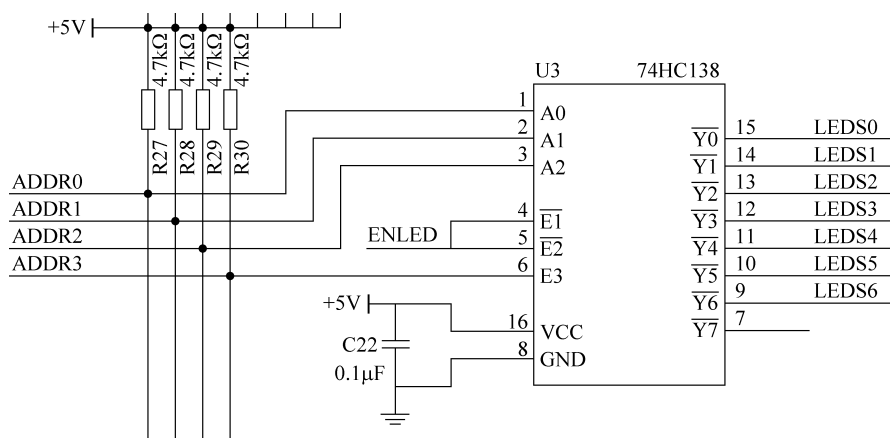



图 3-15 74HC138 应用原理图

就是控制使 74HC138 使能的。

这类逻辑芯片,大多是有使能引脚的,使能引脚符合要求了,那下面就要研究控制逻辑了。对于数字器件的引脚,一个引脚输入的时候,有 0 和 1 两种状态;两个引脚输入的时候,就会有 00、01、10、11 这 4 种状态;3 个引脚输入的时候,就会出现 8 种状态。可以看图 3-16 所示的 74HC138 真值表,其中输入是 A2、A1、A0 的顺序,输出是从 Y0, Y1, ..., Y7 的顺序。

从图 3-16 可以看出,任一输入状态下,只有一个输出引脚是低电平,其他的引脚都是高电平。在前面的电路中已经看到,8 个 LED 小灯的总开关三极管 Q16 基极的控制端是 LEDS6,也就是 Y6 输出一个低电平的时候,可以开通三极管 Q16,从右侧的希望输出的结果可以推导出 A2、A1、A0 的输入状态应该是 110,如图 3-17 所示。

那么再整体回顾一遍点亮 LED 小灯的过程,首先看 74HC138,要让 LEDS6 为低电平才能导通三极管 Q16,所以

```
ENLED = 0; ADDR3 = 1;
```

保证 74HC138 使能。然后

```
ADDR2 = 1; ADDR1 = 1; ADDR0 = 0;
```

这样保证了三极管 Q16 这个开关开通,5V 电源加到 LED 上。

而 74HC245 左侧是通过 P0 口控制,让 P0.0 引脚等于 0,就是 DB_0 等于 0,而右侧 DB0 等于 DB_0 的状态,也是 0,那么这样在这一排共 8 个 LED 小灯中,只有最右侧的小灯和 5V 电压之间有压差,有压差就会有电流通过,有电流通过 LED2 就会发光了。

从图 3-17 中可以看出,74HC245 左侧是直接接到 P0 口上的,而 74HC138 的 ADDR0~ADDR3 接在何处呢? 来看图 3-18。

A2, A1, A0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0 0 0	0	1	1	1	1	1	1	1
0 0 1	1	0	1	1	1	1	1	1
0 1 0	1	1	0	1	1	1	1	1
0 1 1	1	1	1	0	1	1	1	1
1 0 0	1	1	1	1	0	1	1	1
1 0 1	1	1	1	1	1	0	1	1
1 1 0	1	1	1	1	1	1	0	1
1 1 1	1	1	1	1	1	1	1	0

图 3-16 74HC138 真值表

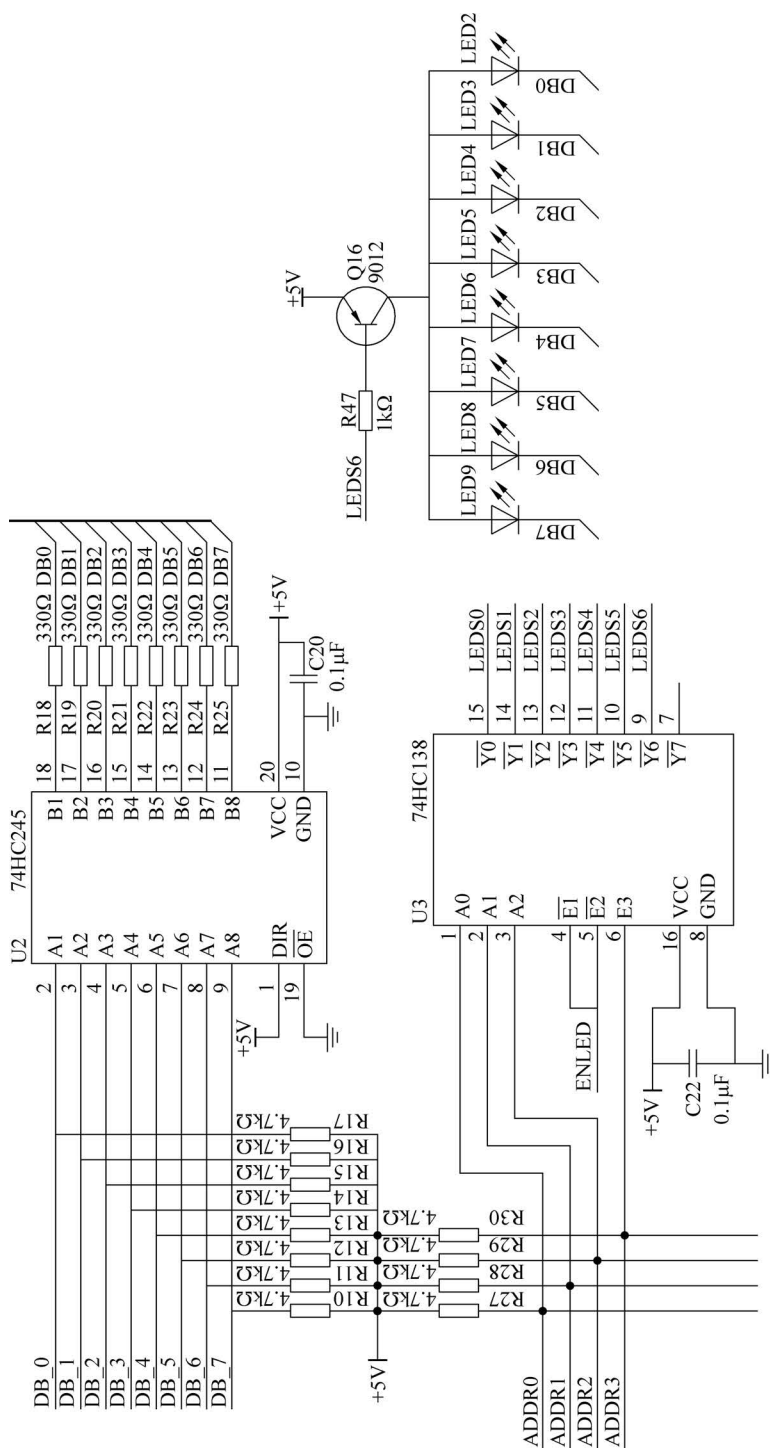


图 3-17 LED 小灯整体电路图

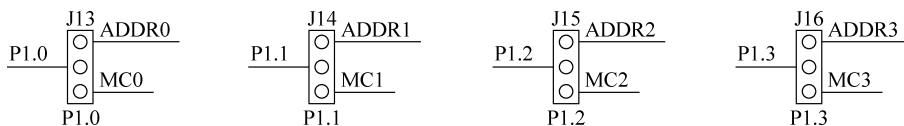


图 3-18 显示译码与步进电机的选择跳线

跳线是读者以后经常接触到的一个器件，它就是 2 根或者 3 根靠在一起的排针，然后用跳线帽连接其相邻的 2 根针。它起到导线的作用，可以通过跳线帽实现连接线的切换，如图 3-19 所示。

从图中可以看出，跳线帽本身可以占两根针的位置，现在把右侧和中间的针连到了一起，这样实现的就是图 3-18 中的 P1.0 和 ADDR0 连接到一起，P1.1 和 ADDR1 接一起，P1.2 和 ADDR2 接一起，P1.3 和 ADDR3 接一起，这样读者就可以透彻理解前面的程序了。

可以认真再回顾一下前面的程序，领悟这几个数字器件的用法。

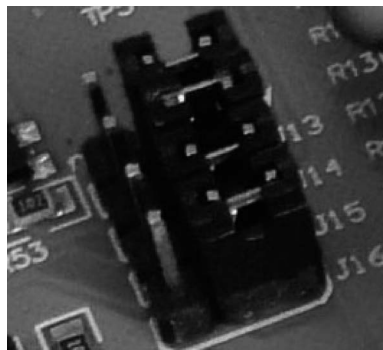


图 3-19 跳线实物图

3.5 LED 闪烁程序

点亮 LED 小灯的程序就是让 $LED=0$ 。熄灭小灯的程序也很简单，就是让 $LED=1$ 。点亮和熄灭都会了，那么如果在亮和灭中间加个延时，反复不停地点亮和熄灭小灯，就成了闪烁了。

首先复习一下 Keil 编写程序的过程：建立工程→保存工程→建立文件→添加文件到工程→编写程序→编译→下载程序。

LED 闪烁程序对于有 C 语言基础的读者来说很简单。下面先把程序写出来，读者可以先看下。没有 C 语言基础的读者可以跟着抄一遍，之后会补充讲解部分 C 语言基础知识。先抄一遍，再跟着看 C 语言基础知识，这样就比较容易理解透彻了。

```
#include <reg52.h>

sbit LED = P0^0;
sbit ADDR0 = P1^0;
sbit ADDR1 = P1^1;
sbit ADDR2 = P1^2;
sbit ADDR3 = P1^3;
sbit ENLED = P1^4;
void main()                                     //void 即函数类型
{
    //以下为声明语句部分
    unsigned int i = 0;                         //定义一个无符号整型变量 i,并赋初值 0

    //以下为执行语句部分
```

```
ENLED = 0;           //U3、U4 两片 74HC138 总使能
ADDR3 = 1;           //使能 U3 使之正常输出
ADDR2 = 1;           //经 U3 的 Y6 输出开启三极管 Q16
ADDR1 = 1;
ADDR0 = 0;
while (1)
{
    LED = 0;          //点亮 LED 小灯
    for (i = 0; i < 30000; i++); //延时一段时间
    LED = 1;          //熄灭小灯
    for (i = 0; i < 30000; i++); //延时一段时间
}
```

把这个程序编译,然后下载到单片机中,就会发现 LED2 这个小灯会闪烁了。

3.6 习题

1. 深刻理解电容的意义,并且在今后学习电路的过程中,要多多注意并且参考其他人设计的电路中所用到的去耦电路,积累经验。
2. 完全记住三极管的导通原理,并且熟练掌握 NPN 和 PNP 三极管在开关特性下的应用方法。
3. 学习并且掌握 74HC245 和 74HC138 的应用原理,能够在设计电路时正确应用。
4. 能够独立实现点亮开发板上的每一个小灯,并且实现小灯点亮和关闭以及闪烁功能。