

第九章 IO 口不够 扩展器来凑

在我们进行单片机开发的时候，经常会发现 IO 口不够用。一方面是因为我们产品中往往都包括很多的功能，又有显示，又有存储等等；另一方面是我们所使用单片机芯片确实 IO 资源不多。有人说，你可以选 IO 口比较多的芯片，比如说把 STM32F103C8T6 换成同型号大封装的，或者直接换 F407。没错，如果在成本允许的情况下，确实是 IO 不够一换了之。但成本往往是需要考虑的首要因素。越是上量的产品，就越是对成本敏感。尤其是 2020 年前后这几年芯片严重短缺，新冠病毒的国际大流行和美国芯片法案是最重要因素。这导致很多存量芯片的价格飞涨，那些不常用的、高端的芯片价格更是离谱，而且不好买。在这种特殊的背景下，我们会尽量用中低端的、常用的、市场存量大、交货周期短的芯片。这个时候，我们去自己通过廉价的方案扩展出一些必要的硬件资源，就显得更加迎合时宜了。

振南为了写一章作足了功课，花了大量精力收集了市面上主要的一些 IO 扩展方案，同时不乏创新性。

1、基于 74 系列芯片的廉价方案

用惯了各种高大上的芯片，似乎我们已经忘记了那些基础简单而又极为重要的芯片，比如 74 系列芯片。我认为芯片没有低高端之分，只有是否用得好之别。

1.1 并行输出端口扩展

回想我们使用 8 位单片机的时候，比如 51、AVR、PIC 等，它们的 IO 端口是 8 位的。但是很多器件是 16 位接口的，比如一些液晶屏，或是存储器。此时我们一般会把两个 IO 端口拼起来，形成 16 位。但前提是我们的 IO 比较富裕。如果只剩下一个 IO 端口了，那怎么办？我们可以使用 74HC573、74HC373 这类的锁存芯片，请看图 9.1。

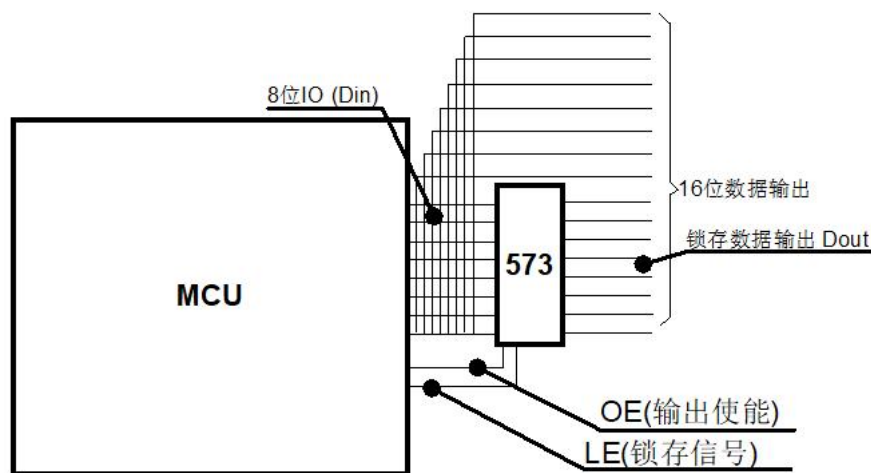


图 9.1 使用 74HC573 将 8 位端口扩展出 16 位

对于 74HC573 想必大家都比较熟悉了，振南就不赘述了。这里简单介绍一下上图的操作逻辑。单片机首先通过 8 位 IO 端口输出 16 位的高字节 ByteH，同时将 OE 置低，LE 由低置高，这样 ByteH 就被锁存输出到了 Dout（锁存后的 Dout 不会再随 Din 变化而变化）。然后单片机再将 8 位端口输出 16 位的低字节 ByteL，[ByteH ByteL]就拼成了 16 位输出。

这算是 IO 扩展的最基础、最常用的方法。大家可以回想一下 51 单片机是如何扩展 SRAM 的，比如 62256 之类的芯片。对，它就是使用了一片 8 位锁存芯片，由 P0 来产生访存地址的低 8 位（P2 产生高 8 位）。请看图 9.2。

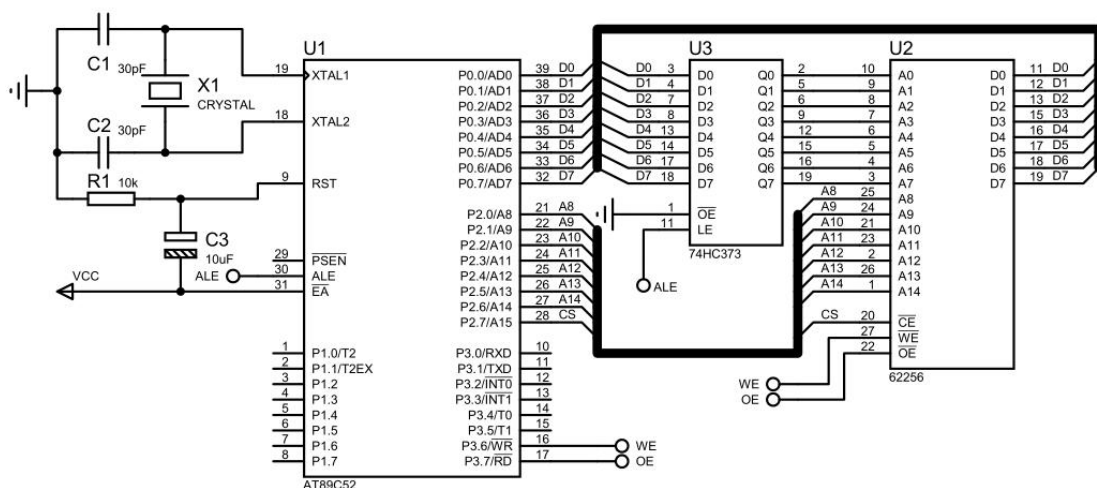


图 9.2 51 单片机外扩 62256 的原理图

简单描述：51 单片机为了节省引脚，在外部访存中对 P0 进行了复用。P0 首先输出 16 位地址的低 8 位 A0~A7，其通过锁存器由 8 位锁存器输出，其与 P2 输出的高 8 位 A8~A15 共同形成了 16 位访存地址。随后 P0 将要写入到外部存储器的 8 位数据从 P0 输出，在#WE 信号的作用下，数据被写入到相应地址单元中。这是一个 xdata uint8 数据赋值操作单片机在背后所产生的一系列动作，当然这些时序都是它自动产生的，属于 CPU 的访存原子操作。

是不是现在很多初学者已经对 51 单片机不太熟悉了？而是直接使用 STM32 来入门，那上面的这个例子，对这些人来说可能就不好理解了。

有人问“那我想扩展出 32 个 IO 怎么作呢？”如法炮制，请看图 9.3。

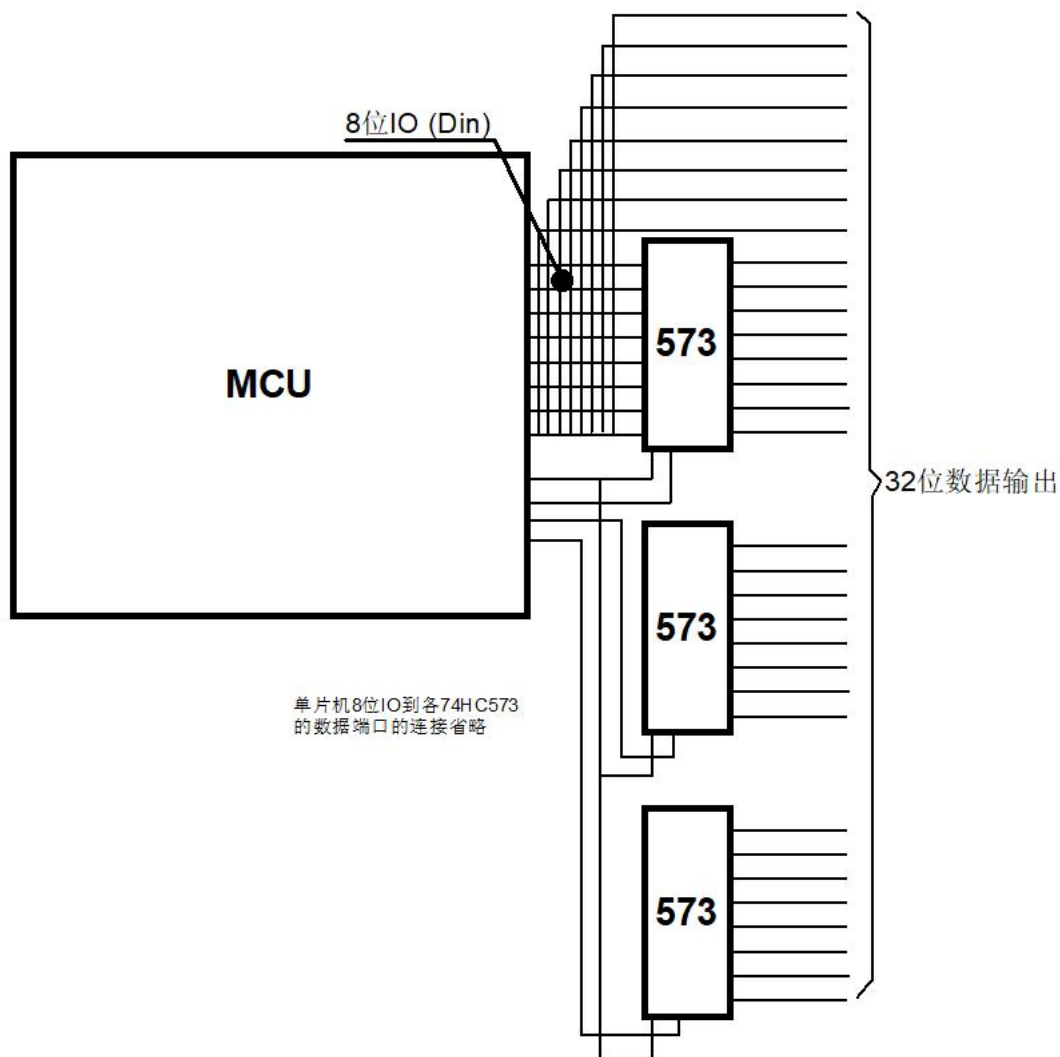


图 9.3 使用 3 片 74HC573 将 8 位端口扩展出 32 位

那是不是我们可以通过这种堆砌 74HC573 的方法，扩展出无限的 IO？理论上是的，但实际上需要考虑器件的负载能力，一般来说 CMOS 电路可同时接 8 个负载。

除了 8 位锁存器，还有 16 位锁存器，比如 74ALVCH162373，芯片示意如图 9.4。

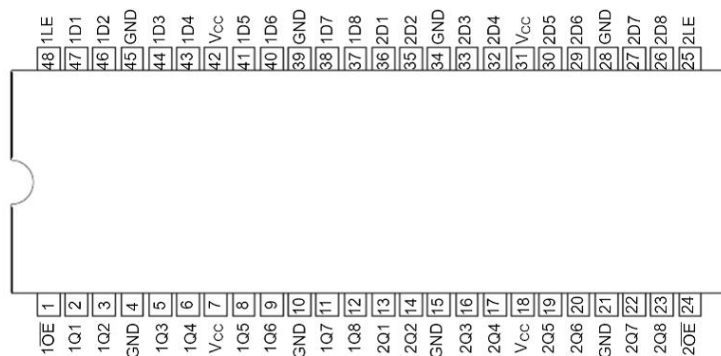


图 9.4 16 位锁存器芯片 74ALVCH162373 引脚分布

乍一看，便条蜈蚣一样，一共有 48 个引脚。可以看到，它有两个 8 位输入端口 1D1~1D8 2D1~2D8，两个 8 位输出端口 1Q1~1Q8 2Q1~2Q8，同时还有两个#OE 和两个 LE。通过它我们

可以把 16 位 IO 端口，扩展为 32 位。其实这一芯片比较小众，需要 32 位端口的应用场景并不多。什么？驱动大量 LED 或数码管？OK，这算一个。但实际上这种应用场景下，使用这种方案来扩展 IO 也并不可取。因为它需要单片机能够先提供 16 个 IO 出来，而这在 IO 资源本就很拮据的前提下，显然是困难的。

通常会使用 8 位锁存器+译码器的方式来实现。原理很简单，用译码器的输出作为 8 位锁存器的锁存信号。如图 9.5 所示。

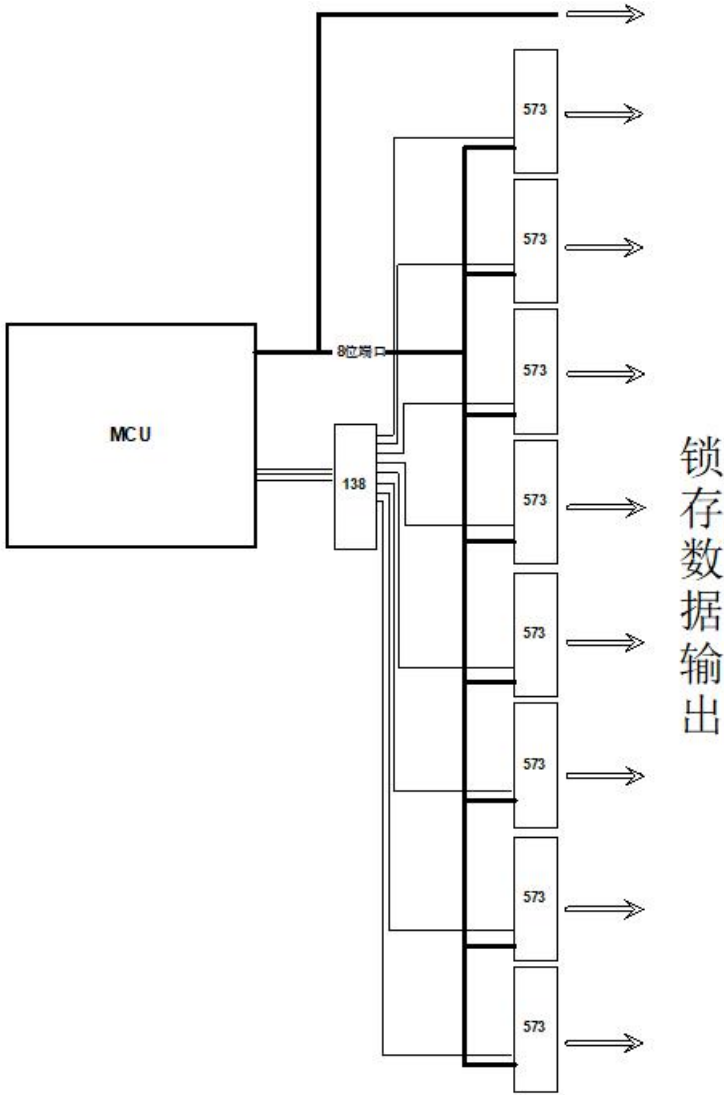


图 9.5 74HC138+74HC573 实现更多 IO 的扩展

关于原理振南就不再赘述了，大家应该都明白。另外关于译码器，我们常用的是 74HC138，3-8 译码器，上图中就是。其实还有更多路的译码器，比如 CD54HC154，它是 4-16 译码器。使用译码器，或者称数据选择器，可以适当较少在并行 IO 扩展过程中所需要额外分配的 IO 数量（如果不用译码器，每增加一片 74HC573 就需要多拿出一个 IO 来控制其 LE）。

1.2 串行输出端口扩展

上面介绍了一些并行输出端口扩展的方法，可能有一些读者有这样的疑问：“你这有点押注的感觉，要想得到，先得搭进去一些，如果我单片机连 8 个 IO 都不剩了，那怎么办？”可以用串行扩展方式。如图 9.6 所示。

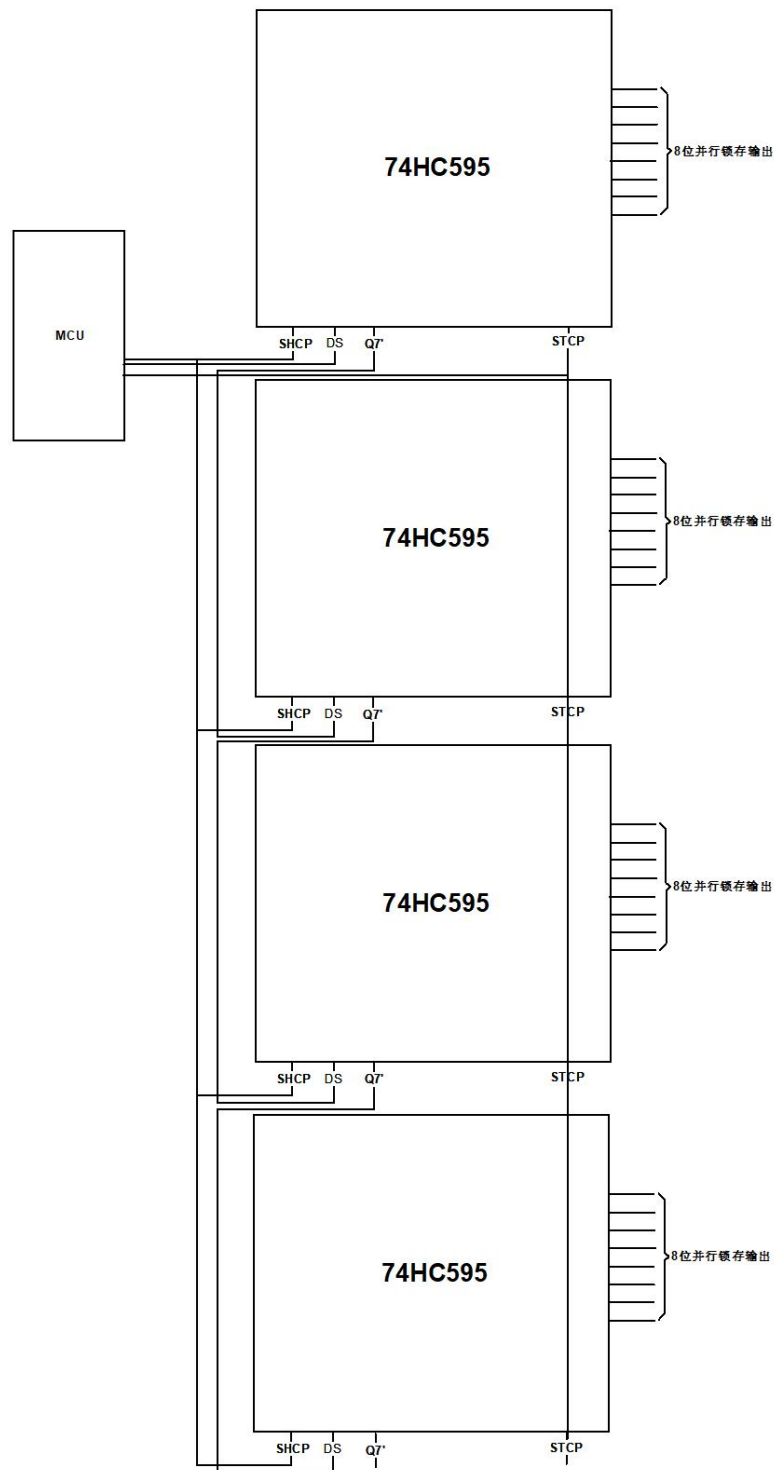


图 9.5 采用多片 74HC595 级联实现串行输出端口扩展

大家对 74HC595 应该很了解了，它是串行转 8 位并行的位移缓存器，也称移动寄存器。它最大的特点就是支持菊花链(daisy chain)。仔细观察上图，可以看到 74HC595 除了 8 位并行输出之外，还有一个 Q7'，它可以连接到下一个 74HC595 的 DS 上，这样串行数据位就可以在时钟 SHCP 的作用下，像流水这样贯穿多片 74HC595。有了这一机制，理论上我们就可以仅用 3 个 IO 扩展出无数的输出端口来。同样的，也有 16 位的串转并芯片，比如 NJU3716，但也属于小众芯片。

我们在作产品研发的时候，在器件选型方面要考虑它的功能性能是否满足我们的设计需要，同时还要考虑器件的可替代性和供应链的情况。一要使用主流的，最好有多个厂商都生产的芯片，比如 74 系列芯片，作为最最基础的门电路，欧森美、TI、NXP 等厂商都在生产；现比如这几年炒得沸沸扬扬的 STM32，它不光有 ST 在生产，相关的兼容型号也有很多其它厂商在生产，比如国内的兆易创新生产的 GD32 系列、灵动微的 MM32 系列、艾派克的 APM32 系列等等，这样就使得用 STM32 设计产品比较安全。因为兼容型号很多，市场存量非常大，这样就算在半导体极端行情的背景下，我们产品的物料元件也不至于断供。

同时，尽量使用最畅销，最常用的型号，比如 STM32F103。因为在极端情况下，半导体厂商会削减产能，把有限的生产能力放到那些出货量最大的型号上。所以，在我看来，那些高端的芯片，比如 STM32L/F4、H7 系列都是有风险的。还有不要使用特异封装，比如 VFQFPN、WLCSP、UFBGA 等。一方面是这些特异封装用量小，特殊时期可能因为厂商减产采购不到，就算能采购到，交货周期也会很长。另一方面，这些封装都需要比较高的加工焊接工艺，能接单的板家比较少，而且周期长，次品率高。不要一味地猎奇，或者过分的在意电路尺寸，给自己制造麻烦。

说了这么多，可能很多人还是认识不深刻，容易被正常时期的供应链充沛状态所蒙蔽。说两个活生生的例子。在很久以前，有一款单片机非常流行，知名度非常高，性能也比同期的其它芯片要高。基于它，国内外的工程开发和发布了很多优秀的硬件开源项目 and 商业产品，甚至有人用它作音视频编解码、图像识别、跑 Linux。对，它就是 AVR，Atmel 的当家花旦，就如同 Microchip 的 PIC，SiliconLab 的 C8051F 一样的历史地位。但是在 2015 年前后，AVR 的供货急转直下，甚至一片难求。究其原因应该是 Atmel 与 Microchip 的并购导致的市场动荡。用 Atmel 自己的话说是：原因在于这一收购案过程的不确定性，导致经销商降低库存水位，特别是在亚洲地区。这样的芯片短缺状态持续了 2 年左右，导致大量的 AVR 用户不得不转向其它芯片。一方面是供货周期过长，另一方面是因为经销商惜售而造成的高价格。又鉴于当时 STM32 以及很多国产 8 位 MCU 已经崛起，从而使得 AVR 惨遭抛弃。昔日那些基于 AVR 的开源项目和社区论坛，比如国内比较有名的 ourAVR 就此没落，从而开始寻求综合化论坛的发展路径，也因此改名为 ourDEV。

另一个例子是发生在我身上的。前面我说过，有好几年我一直在作智能传感器。因为涉及的产品非常多，有加速度、倾角、温度、应变等等。所以我所带领的硬件团队考虑作一个通用的核心板，而把通信和采集作成模块，方便插接集成和复用，希望通过这种模块化的架构来提高整体的研发效率。核心板主芯片选型时，考虑到尽量把 PCB 作小，以便可以直接利用现有的外壳，以及为电池留出更多的空间。我们最终选定了 STM32L452RET6 UFBGA 封装（一种 pitch 非常小的 BGA 封装），如图 9.6 所示。

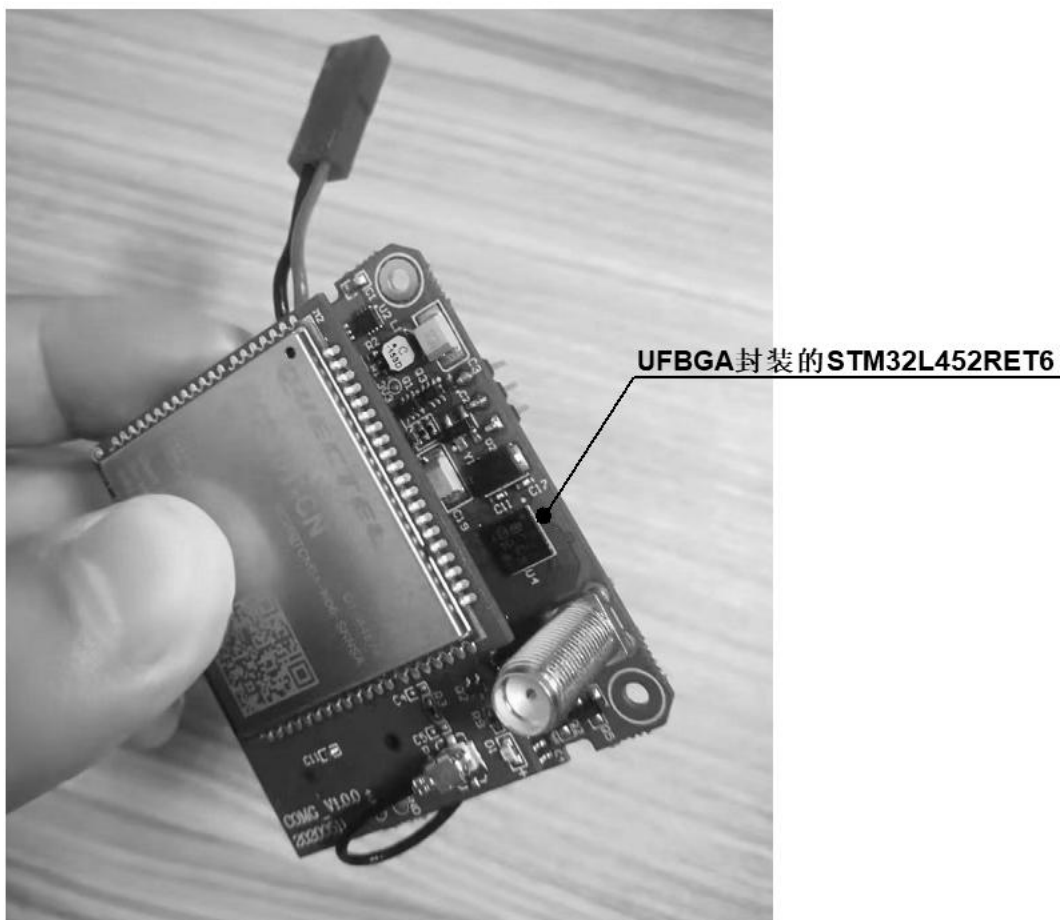


图 9.6 使用 UFBGA 封装的 STM32L452RET6 作为模块化核心板的主芯片

产品不断上量，困境也随之而来。先不说 UFBGA 的高加工费，当然加工费可以用产品批量均摊掉。随着 2019 年年末席卷而来的新冠疫情，因 ST 芯片产能的限制，整个 STM32 的供应链出现了很大的问题。供应链团队是直接向 ST 原厂订货的，周期已经排到了 2 年以后，而且能排上已经是谢天谢地了。而 2019~2020 年正处于公司 B 轮融资的极速扩张期，大量的项目订单纷至沓来，而库存芯片仅有 1000 多片，一个项目就可能倾数耗尽。起初尝试向兆易创新咨询是否有替代型号，但是 L452 这款芯片并不常用，无果而终。最后逼得没有办法，开始替换为 STM32L452REY6，其封装为 WLCSP，它的 pitch 更小，加工焊接需要工艺更高。但是我们确实发现这一封装的市场存量比较大，可以支撑产品渡过困难期。我考虑，就是因为这种封装不易焊接，才作为库存一直压在经销商手上。

经过这些事情，一改我对供应链的观念。以前我认识供应链就是买买买，但实际上并没那么简单。它直接关系到我们的研发成果是否有实际批量意义。因为某个关键器件供货出问题，而不得不整体改版的事情我见得太多了，所以请大家一定要谨慎选型。

请叫我“跑题大王”！OK，继续。

关于 74HC595 可能有人会问：“串转并，那速度不会很慢吗？”并行一定比串行快，这是一个认知误区。其实串行比并行更加稳定，而且速度不慢。这也是现在串行总线大行其道的原因，比如 USB、SATA 等。我们可以用硬件 SPI 向 74HC595 写入数据，串行时钟可以达到 10MHz，这样转为并行，也有 1Mbps 的数据速率了。这对于一般的应用是足够了。如果你非要用级联几百个 74HC595 去驱动 LED 大屏，还要实现显示特效的话，我们就需要性能更高、速度更快的控制器了(大多使用专门的 ASIC 芯片或 FPGA)。74HC595 的时钟频率支持

到 100Mhz 是没有问题的。

1.3 并行输入端口扩展

前面我们介绍的都是输出端口扩展,输入也同样重要。我们可以使用数据选择器来实现。请看图 9.7。

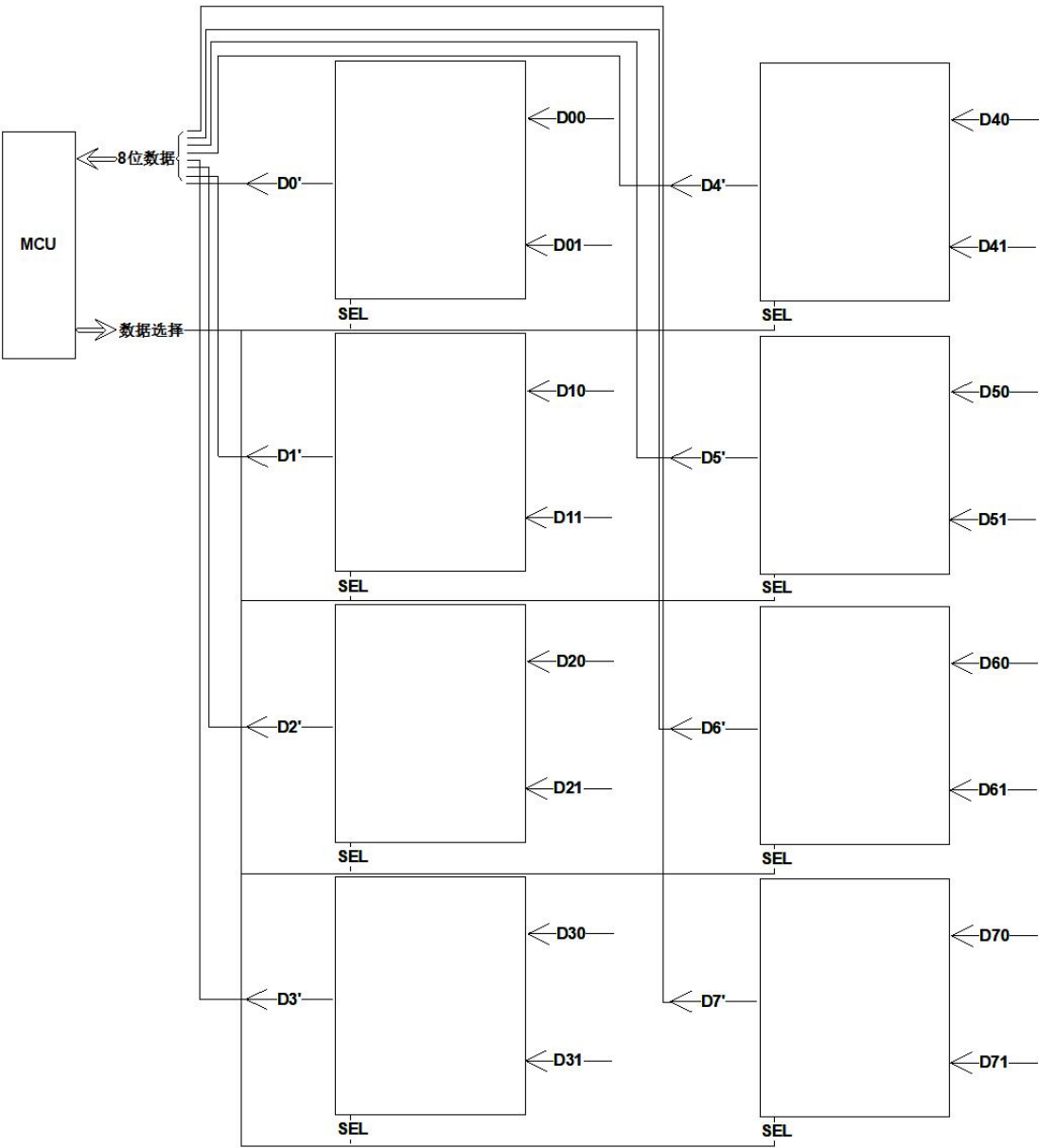


图 9.7 使用 2 选 1 数据选择器实现 8 位并行输入端口扩展

图中 D00~D70 是一个 8 位并行输入端口, D01~D71 是第二个端口。单片机通过控制数据选择引脚 SEL, 来切换两个输入端口。如果要扩展更多的端口, 可以使用更多路的数据选择器, 比如 4 选 1 或 8 选 1。

OK, 同样的套路, 我们来看看如果通过串行方式来扩展并行输入端口。

1.4 串行输入端口扩展

74HC595 是数据串行输入, 并行输出。那是否有芯片可以实现数据并行输入, 串行输出呢? 当然有, 74HC165, 而且它也支持菊花链。来看一下它的脚引分布, 如图 9.8。

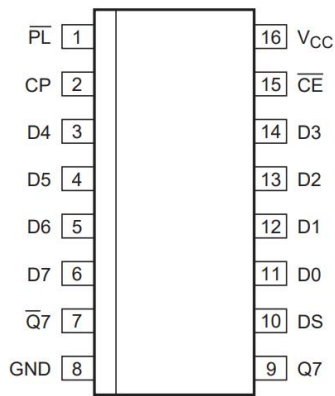


图 9.8 74HC165 的芯片引脚分布图

D0~D7 是 8 位并行输入，DS 与 CP 是串行数据与时钟，#PL 是数据装载，它为低电平时 D0~D7 输入状态将被读入到内部的移位寄存器中。Q7 这个引脚很重要，它是移位寄存器的输出，通过它我们就可以实现菊花链级联。如图 9.9 所示。

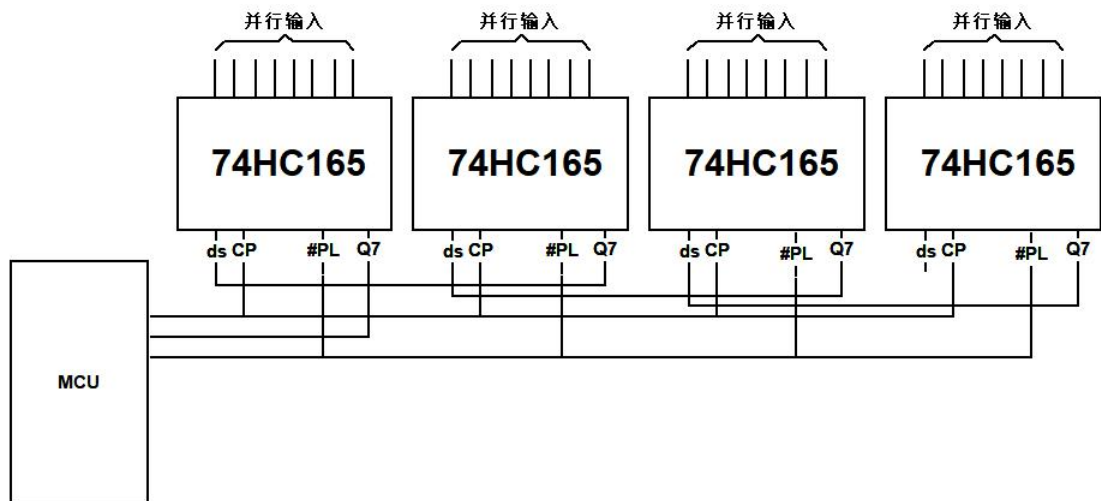


图 9.7 使用多片 74HC165 级联实现大量输入端口扩展

以上振南介绍了输出端口与输入端口的扩展方法，但它们单独拿出来都算不上是 IO 扩展，因为它们只能输出或输入。那如何实现可以输出同时又能输入的真正意义上的 IO 扩展呢？很简单把输出和输入端口拼在一起就行了。但是要注意一点，当它用于输入的时候，输出端口要保持高阻状态。拿 74HC595+74HC165 这样的组合来说，当使用 74HC165 读取并行数据的时候，74HC595 要将#OE 拉高，使其输出呈现高阻态。否则，它会影响 74HC165 的电平读取。

有人还是在顾虑串行速率的问题“我用 51 单片机，没有硬件 SPI，只能用 IO 模拟，怎么让串转并速度快一些？”既然你使用这么古老的 51 芯片(STC51 基本上都支持硬件 SPI)，还奢求速度？好吧。51 单片机有一个被很多人忽视的功能，即 UART 的模式 0，它就是一种硬件同步串行接口，大家可以去了解一下。

2、基于专门 IO 扩展芯片的方案

使用 74 系列芯片固然成本低，而且安全，但是它有一个很大的问题，比较占用 PCB 面积。一个板子，放眼望去有很多片 74 芯片，这无形之中拉低了板子的档次。IO 短缺其实是

行业内一个很常见的问题，因此芯片厂商专门研发了一些 IO 扩展芯片。它们功能丰富，集成度高，同时价格可控，对于一些成本不是极为严苛的项目和产品，不失为是终极解决方案。

以下的讲解振南不会去细述芯片的具体使用方法，只会简单介绍其主要功能，因为复述芯片手册是一件毫无营养的事情。振南主要进行一些芯片的推荐，大家可以依自己的实际情况灵活选型。

1、并行 IO 扩展芯片

这一类的专用芯片不多，我仅以我知道的说一下。

1) 8255

8255 是一个代号，或者说别称，它不是芯片型号，而是指与之兼容的由不同厂家生产的多个型号，如图

| | | | |
|-------|-----------------------------------|--|--|
| 82C55 | OKI electronic componets | | |
| 82C55 | Toshiba Semiconductor and Storage | | |
| 82C55 | Intel | | |
| 82C55 | Intersil Corporation | | |
| 82C55 | Wing Shing Computer Components | | |
| 82C55 | ETC | | |

图 9.8 多个厂商都在生产 8255 兼容型号

8255 是一个功能强大而又古老的芯片，最初由 Intel 开发，基本与 8051 是同时代的产品。它有多种工作模式，我们这里仅介绍其最简单最常用的一种，即模式 0。

先来看一下 8255 的引脚分布，如图 9.9。

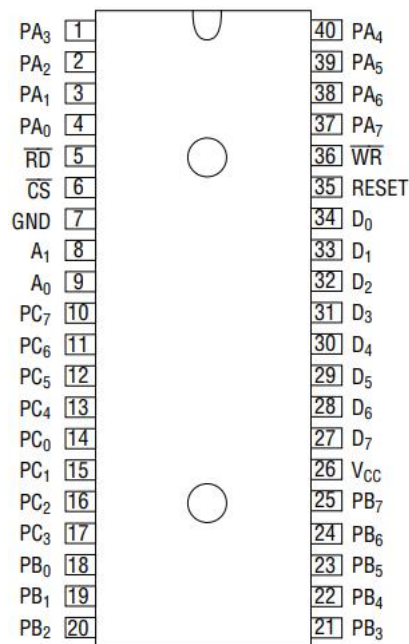


图 9.9 8255 的引脚分布

D0~D7 是 8 位输入输出端口，用于与单片机连接，以实现内部寄存器的读写。PA、PB 和 PC 为 3 个 8 位输入输出端口，即扩展端口。A1-A0 是两位的地址码，用于寄存器寻址，所以 8255 有 4 个寄存器，分别是 PA、PB 和 PC 的数据寄存器，和控制寄存器。

我们先来看一下 8255 的基本操作，如图 9.10 所示。

| A1 | A0 | \overline{RD} | \overline{WR} | \overline{CS} | INPUT OPERATION (READ) |
|----|----|-----------------|-----------------|-----------------|-----------------------------|
| 0 | 0 | 0 | 1 | 0 | Port A → Data Bus |
| 0 | 1 | 0 | 1 | 0 | Port B → Data Bus |
| 1 | 0 | 0 | 1 | 0 | Port C → Data Bus |
| 1 | 1 | 0 | 1 | 0 | Control Word → Data Bus |
| | | | | | OUTPUT OPERATION (WRITE) |
| 0 | 0 | 1 | 0 | 0 | Data Bus → Port A |
| 0 | 1 | 1 | 0 | 0 | Data Bus → Port B |
| 1 | 0 | 1 | 0 | 0 | Data Bus → Port C |
| 1 | 1 | 1 | 0 | 0 | Data Bus → Control |
| | | | | | DISABLE FUNCTION |
| X | X | X | X | 1 | Data Bus → Three-State |
| X | X | 1 | 1 | 0 | Data Bus → Three-State |

图 9.10 8255 的基本操作

控制寄存器是比较重要的寄存器，通过它我们可以设置扩展端口的工作模式、输入输出方向等，如图 9.11。

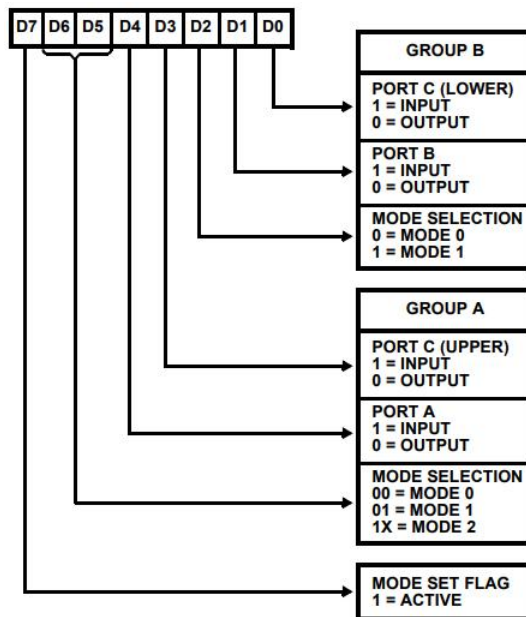


图 9.11 控制寄存器的定义

比起现在芯片动辄几百个寄存器，8255 相对要简单多了。从控制寄存器的定义中可以看到，PC 端口被砍成了两段 UPPER 与 LOWER，分别为 4 位，可以单独设置它们的方向。而且更强大的是，PC 端口是支持位操作的，置位或者清零，类似于 51 单片机 sbit 位寻址或者是 STM32 的 bitband 位带。

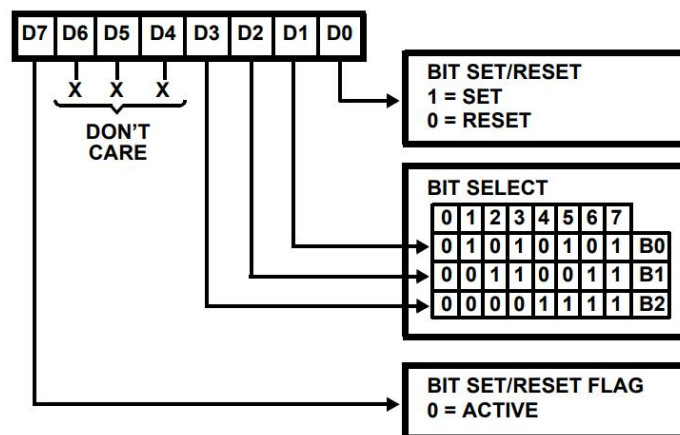


图 9.12 控制寄存器用于 PC 端口位操作时的定义

这样的设计非常实用。我们在使用 IO 端口进行数据读写的时候，通常还要配有相应的控制信号，比如 8080 总线除了数据线还有控制线#RD 和#WR。所以 PC 口就可以充当控制线，位控制可以提供执行效率。

关于 8255 更多的功能，大家可以去它的数据手册。

哪里可以找到它的数据手册？ 1、官网 2、21icsearch 3、alldatasheet 4、百度 2) CH351

前面说过 8255 是比较老的芯片，实际上现在可能不太好买，而且市面上散料比较多。我再推荐一款芯片 CH351，来自于南京沁恒。对于这个公司大家应该比较熟悉，起码有所耳闻。它主要是研发接口扩展、转换、桥接这一类的芯片。最有名的，也是用量

最大的就是他的 CH341（USB 转 TTL 串口桥接芯片）。有必要跟大家说一下国内的几个我们经常用到的芯片和器件供应商及其代表产品，如表 9.1。

表 9.1 国内的一些常见芯片器件供应商及其代表产品

| 芯片厂商 | 代表产品 | 说明 |
|--------|-------------------|---|
| 南京沁恒 | CH375、CH341 | 专注于连接技术和 MCU 内核研究 |
| 武汉天马 | TM1668 | 提供显示解决方案 |
| 江苏宏晶 | STC12C5A60S2 | 专注于国产 8051 内核芯片的研制 |
| 广州金升阳 | ACDC DCDC 芯片模块 | 一站式电源解决方案 |
| 北京兆易创新 | NOR Flash、GD32 系列 | 专注于存储、Arm 核 MCU 和触控芯片 |
| 北京圣邦 | SGM3001、SGM706 | 专注于高性能、高品质模拟集成电路的研发和销售 |
| 福州瑞芯微 | RK3288 | 物联网(IoT)及人工智能物联网(AIoT)处理器芯片 |
| 珠海全志 | A133 A40i-H | 卓越的智能应用处理器 SoC、高性能模拟器件和无线互联芯片设计厂商 |
| 上海复微 | FM25Q64 | 已形成安全与识别、智能电表、非挥发存储器、智能电器四大成熟的产品线和系统解决方案 |
| 台湾华邦 | W25Q256 | 是台湾唯一同时拥有 DRAM 和 Flash 自有开发技术的厂商 |
| 台湾新唐 | Mini51 系列 | 专注于控制/微处理、智能家居及云端安全相关应用 IC、电池监控 IC、影像感测 IC、IoT 应用 IC、半导体组件等 |

可以看到国内半导体产业其实并不弱，不管是研发设计，还是晶圆生产封装测试，整个产业链是很完整。平时大家说的中国芯片落后是指高端和尖端的芯片设计和制造，其实一般性的芯片，比如汽车、功率器件等等自给自足是完全没有问题的。一定要对国产器件有信心，选型多选国产，因为自己的才是最安全的。

CH351 可以扩展出 32 个 IO，请看图 9.13。

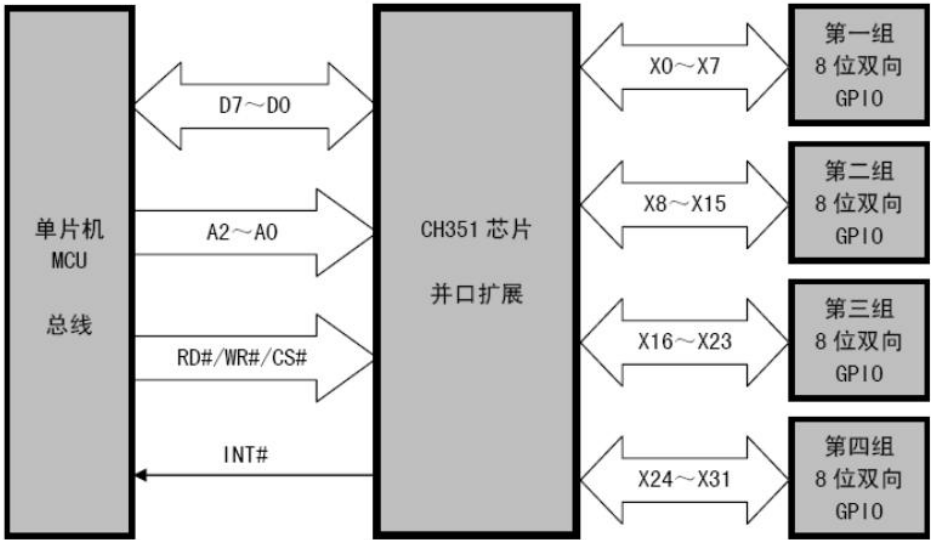


图 9.13 CH351 的芯片应用框图

具体的使用方法振南就不赘述了，大家可以到沁恒官网去下载芯片手册。

2、串行 IO 扩展芯片

串行 IO 扩展芯片是 IO 扩展的主流方案，它最大的优势是使用极少的 IO 即可扩展出大量 IO，而且集成度高，性价比高。按照所使用的串行接口，振南对这些芯片进行罗列，请看表 9.2。

表 9.2 主流常见的串行 IO 扩展芯片

| 芯片厂家 | 芯片型号 | 接口类型 | 说明 |
|-----------|------------|------|---------------------------------------|
| 南京沁恒 | CH423 | I2C | 提供 8 个双向输入输出引脚和 16 个通用输出引脚，支持输入电平变化中断 |
| 上海艾为 | AW9523 | I2C | 可提供 16 个 IO 端口扩展 |
| 无锡力芯 | ET6416 | I2C | 可提供 16 个 IO 端口扩展 |
| TI | TCA6424 | I2C | 可提供 24 个 IO 端口扩展 |
| TI | TCA9555 | I2C | 可提供 16 个 IO 端口扩展 |
| TI | PCF8574 | I2C | 可提供 8 个 IO 端口扩展 |
| TI | PCF8575 | I2C | 可提供 16 个 IO 端口扩展 |
| ISSI | IS31IO7325 | I2C | 可提供 8 个 IO 端口扩展 |
| NXP | PCA9698 | I2C | 提供 40 个并行 IO 端口扩展 |
| Microchip | mcp23017 | I2C | 可提供 16 个 IO 端口扩展 |
| MAXIM | MAX7312 | I2C | 可提供 16 个 IO 端口扩展 |
| 成都国腾 | GM8166 | SPI | 提供 32 个并行 IO 端口扩展 |
| 成都国腾 | GM8164 | SPI | 提供 32 个输入引脚与 40 个输出引脚 |
| Microchip | mcp23s17 | SPI | 可提供 16 个 IO 端口扩展 |
| 日本 JRC | NJU3716 | SPI | 可提供 16 个 IO 端口扩展 |

其中 TCA9555 有非常多的兼容型号,比如 PCA9555、AT9555、XL9555、NCA9555 等等。

上面只是罗列了常见的一些串行 IO 扩展芯片，实际上要比这里看到的要多得多。可见，IO 扩展是业内的一个普遍问题，各大芯片厂商都有自己的方案，但其实它们都是大同小异。具体哪一款适合你，大家可以根据自己的产品功能与成本需求进行选型。

除了这些专用 IC，其实还有人用 CPLD 或 FPGA 器件来实现 IO 扩展，也是一种很好的方案，因为它具有很好的灵活性。而且现在 FPGA 这块也不是老美 Xilinx 和 Altera 独揽的局面，很多国产 FPGA 具有很高的性价比，比如 AGM 系列、安路 FPGA 系列等等。同时，在可编译逻辑器件这方面，有一个比较新的概念 CIMC，即可编程混合信号芯片，感兴趣的读者可以关注一下，据说性价比巨高。

OK，这一章就写到这里，继续。