

嵌入式硬件设计必备基础知识

一、嵌入式计算机体系结构

计算机能用来干什么？它必须完成什么任务？它是如何实现与人和其他系统交互的？这决定了这台计算机的功能，进而决定了其体系结构、存储器和 I/O。

计算机系统大致分为两类。第一类就是您目前手上用的计算机，最常见的台式计算机。再就是，您最近有没有用全自动洗衣机洗衣服，把衣服往里面一扔（别忘了放洗衣粉，我就常常忘），只消按一个键便高枕无忧了，这里面就有一个第二类计算机：嵌入式计算机，这种计算机往往作为控制和/或监控目的而集成到另外的系统中（比如洗衣机）。那这样说来我们身边岂不全是计算机？电视机、DVD、遥控器、手机、甚至玩具？就是他们！

一台专用的台式计算机（不一定是 PC）有很大的主内存来支持操作系统、应用程序和数据，以及一个大容量存储设备（硬盘、DVD/CD-ROM 等）的接口。这种台式计算机带有各种各样的 I/O 设备以使用户输入（键盘、鼠标和话筒）、输出（显示器及耳机）以及互联（网络和外设）。快速的处理器需要一个系统管理器来监视其核心温度、供给电压以及进行系统重启。

大规模嵌入式系统计算机也可能采用上述的形式。例如，它可能作为一个网络路由器或网关从而需要一个或多个网络接口、大容量内存以及快速操作。它们也可能需要某种形式的用户界面来作为嵌入式应用的一部分。或者在许多情况下，它们也可能只是专用于某一特定任务的常规计算机。因此，单就硬件而言，许多高性能嵌入式系统与常规台式机没有什么大的差别。

较小的嵌入式系统使用微控制器作为它们的处理器，这样做的优点在于处理器可以将很多的计算机功能包含在一个芯片上。

微控制器至少有一个 CPU，一个小容量的内部存储器（ROM 和/或 RAM）以及作为子系统模块在微控制器内部实现的某种 I/O。这些子系统为处理器提供了附加功能，许多处理器通常都有这种子系统。你经常可以在微处理器里发现这些子系统，这些子系统最普通的 I/O 是数字 I/O，它们都是端口，可以以引脚为基础，由软件配置成数字输入或输出。作为数字输入，这些端口可以用来读取开关或按钮的状态以及读取另外一个设备的数字状态；作为数字输出，它们可以用来开启或关闭外部设备，也可以向外部设备传达工作状态。

例如，一个数字输出端口可以为一个电机激活控制电路，开/关灯，或者可能的话还可以触发诸如花园浇灌系统的水压阀这样的设备，当然也控制洗衣机。将数字输入/输出端口结合在一起可以用来合成对其他芯片的接口和协议。除了数字 I/O 外，大多数控制器还有其他子系统。而如果不需要其他子系统的功

能，那么微控制器本身也能将其他子系统转换到通用数字 I/O。作为一个系统设计者，这就为您在所设计的应用里如何使用微控制器提供了多种选择。

许多微控制器也有模拟输入，允许为监控或记录目的的传感器进行采样。这样，一台嵌入式计算机就可以用来测量光强、温度、震动或加速度，大气或水的压力，湿度或者磁场之类的项目。或者，这些模拟输入可以用来监测简单的电压，或许用来确保较大规模系统的可靠操作。

有些微控制器有串口，通过串口，嵌入式计算机可以与一台主机、另外的嵌入式系统或者也许是一个简单的网络进行连接。诸如 SPI 和 I2C 这样的专用形式的串行接口，为扩展微控制器的功能提供了一种简单的方法。这些串口允许外部设备与微控制器连接起来，为诸如片外存储器（用于数据或参数存储）、时钟/日历芯片、带数字接口的传感器、外部模拟输入或输出，甚至音频芯片及其他微处理器提供了访问途径。

大多数微控制器都带有计时器和计数器，用来在固定的时间间隔里为多任务处理产生内部中断，为片外系统提供外部触发，或者为电机提供控制脉冲。或者用来为来自其他设备的外部触发（脉冲）计数。

有少数的微控制器也包含诸如 USB、以太网、或 CAN 这类的网络接口。

一些比较大的微处理器还提供一个总线接口，将内部地址、数据和控制总线展现给外部世界。这就使得处理器能够以与常规处理器近乎同样的方式与大量可能的外部设备进行连接。前面所描述的所有可能的设备和接口都可以通过总线接口和适当选择的外部设备来实现。总线接口为微控制器带来了巨大的发展潜力。

各个微控制器具有的 I/O 子系统的组合差别相当大。有些微控制器针对简单的数控应用，可能只含有数字 I/O；其他的一些微控制器可能面向工业应用，因此可能有数字 I/O、模拟输入、电机控制以及网络连接等。微控制器（几千个种类）的选择依赖于你的处理需要和接口需求，请选择一款最合适的一款。

以后将选取三星公司生产的以 ARM7TDMI 为核心的 S3C4510B 为例讲解基于它的电路设计以及程序设计。

名词解释：

系统体系结构：单个处理器是不能完成任何任务的，还需要存储器（用于程序核数据存储）、支撑逻辑以及至少一个 I/O 设备（输入/输出设备），用以实现计算机系统于外界之间的数据传输。计算机系统的存储器包含了处理器将要执行的指令以及将要操作的数据。指令由计算机系统从存储器里读取，而数据则可以读出和写入存储器，上述形式的计算机结构就是冯·诺伊曼体系结构。冯·诺伊曼计算机可以归类为控制流计算机，在这种体系结构中，计算机的执行步骤由程序的顺序控制所管理。另有一种为哈佛体系结构，其指令和数据具有不同的内存空间。

总线：所谓总线，就是具有相关功能的信号线的物理集合。总线允许在计算机系统的不同部件之间传输电子信号，因此可把信息从一个设备传输到另外一个设备上。当今的多数微处理器采用了三总线系统体系结构，分别为地址总线、数据总线和控制总线。数据总线是双向的，具体的传输方向由处理器决定。地址总线传输的是地址信息，这一地址指向微处理器想要访问的内存单元。控制总线携带来自处理器的当前访问状态信息，如写操作还是读操作。控制总线也将当前访问状态信息返回给处理器，例如地址错误等。

二、关于电子学的一些东西

模拟信号

一个模拟信号可以拥有一定范围内任何电压的幅值，而不像数字信号那样只能处于两种定义电压状态（要么低电平，要么高电平）之一。如波形为正弦曲线的正弦信号。

信号的电压可以随时间的推移而变化，或者为常量。如果电压是变化的它会每隔一定时间就重复，这种情况下就说这种信号有周期性。这个周期就是信号模式重复的时间间隔（例如，从一个波峰到另外一个波峰）。信号的频率就是每秒钟信号模式重复的次数。

频率的单位是赫兹（单位符号为 Hz），它与周期有如下关系：

$$f = 1 / T \quad (f : \text{频率} \quad T: \text{周期})$$

这样，周期为 1ms 的信号，其频率就是 1KHz。

单极信号组成的电压，要么全正要么全负。双极信号具有正和负两种电压。

典型的模拟信号既有交流(Alternating Current, AC)分量又有直流(Direct Current, DC)分量，DC 分量是信号的固定电压，而 AC 分量则是一个加在 DC 分量上的变化着的电压。AC 分量通常是指信号的峰—峰振幅，一般以后缀 pp 来表示。例如，一个 5V 的 AC 分量可以写作 5Vpp。

电阻

连接到电源的电阻提供高电平信号，称为“上拉电阻”。

连接到地的电阻提供低电平信号，称为“下拉电阻”。

RC 电路

把电容和电阻结合起来会产生一些有趣和有用的效果。一个电阻—电容的结合就是所谓的一个 RC 电路，它们的结合有三种形式。

在第一种形式中，电阻和电容并联在一起。这种结合能用来做什么呢？施加在这两个元件上的电压（V）将对电容充电（也会有一些电流穿过电阻）。当所施加的电压撤去时，电容便会通过电阻释放电容。电阻将限制电荷释放的速率，因为电阻限制了电流。

第二种形式的 RC 电路为串联 RC 电路，当一个电压加到 RC 电路输入端（靠近电阻端）时，电流就会通过电阻，电容也会开始充电。然而电阻限制了电流，进而限制了电容充电的速率。这种类型的 RC 电路是一个简单的低通滤波器。这种电路能将信号的高频分量滤掉，从而在主信号中削弱它们，而低频分量没有任何衰减。这种类型的电路对于去掉叠加在信号上的高频噪声来说非常有用。

任何处理器和外部设备芯片的每个输入引脚上都会有少量的输入电容。输入电容连同电路连接固有的微小阻抗以及引脚的输入阻抗一起意味着施加在引脚上的数字电压实际上呈指数增长，而不是（数字的）陡然上升的。这种影响很小，但在高速电路中，或者当几个设备连接到同一个信号线上而所有的输入电容不可忽略时，这种影响就很大了。

输入端的感应作用会引起另外的一个特性。当源电压突然变化时，电感应作用导致“阻尼振荡”。

第三种形式的 RC 电路仍是 RC 串联，但输入电压靠近电容端。这种类型的电路是一个简单形式的高通滤波器，因为它只输出高频信号。这种电路里的电容通常被称作隔直流电容。

时钟和振荡器

所有的处理器都需要时钟。所谓时钟就是来自振荡器的输出，这一输出管理着处理器以及其他与时钟相关的所有系统事件。其实时钟就是一个数字脉冲流。

任何处理器都有一个最大和最小时钟频率。这表明了振荡器驱动处理器的频率范围。我们说最小时钟速率为零的处理器具有静态操作（Static Operation）或 DC 操作。这意味着处理器能够使时钟停下来，也能在稍后的时刻重新启动操作而不受影响。如果处理器的最小操作频率大于零值，那么称该处理器具有动态操作。如果振荡器的频率低于上述动态处理器的最小频率值，那么处理器的寄存器的内存有可能会被破坏掉。

产生时钟的方式有好几种，有些处理器期望一个数字（方波）时钟输入。对于以通用频率运行的处理器——大多数处理器都属于这类，最好的选择是采用一个名为振荡器模块的设备。另外许多处理器都包含振荡器电路，并且通常只需要增加一个晶体和旁路电容即可。这里的电容用来去除来自振荡的高次谐波。

数字信号

一个数字设备的输出引脚可能处于下列三种状态之一：高电平（逻辑 1）、低电平（逻辑 0）、或三态（高阻态，也叫漂移）。当引脚上的输出电压高于一

一个给定阈值时就定义为逻辑高电平。当一个设备的引脚输出为高时，这一引脚就对其连接提供电流（sourcing current）。同样地，当输出电压低于给定阈值时就定义为逻辑低电平，我们称设备的引脚吸收电流（sinking current）。典型情况是，元件吸收的电流多于它所提供的电流。

一个三态引脚的输出既不是高电平也不是低电平，而是高阻状态，这样流入或流出该引脚的电流都被忽略。实际上，流入或流出该引脚的电流对于与这一引脚相连的其他元件而言是不可见的。例如，计算机系统中可能有好几个存储设备连接到数据总线，当从一个设备读取数据时，它的数据输出要么为高电平要么为低电平。系统内所有其他的存储设备，由于它们未被访问，因此会把它们的数据总线置为三态，不会参与处理器和被访问存储器之间的读数据处理。

电压阈值

TTL(Transistor-Transistor Logic, 晶体管-晶体管逻辑)系列的电压定义：低电平，输入最高电压为 0.8V，输出最高电压为 0.4V；高电平，输入最低电压为 2.0V，输出最低电压为 2.4V。处理器不同规定就不同，所以器件的电气（和时序）规和列在产品手册里。