C语言：

类似于C++，没有类与对象

嵌入式：

（1）硬件：嵌入式系统的微处理器通常是由32位及以上的RISC（Reduced Instruction Set Computer，精简指令集计算机处理器）组成，例如ARM、[MIPS](https://baike.baidu.com/item/MIPS/2173143" \t "_blank)等。

（2）软件：嵌入式系统的软件系统通常是以嵌入式操作系统为核心，外加用户应用程序。

单片机：

单片机（Single-Chip Microcomputer）是一种[集成电路芯片](https://baike.baidu.com/item/%E9%9B%86%E6%88%90%E7%94%B5%E8%B7%AF%E8%8A%AF%E7%89%87/4872662)，是采用超大规模集成电路技术把具有数据处理能力的[中央处理器](https://baike.baidu.com/item/%E4%B8%AD%E5%A4%AE%E5%A4%84%E7%90%86%E5%99%A8/284033)CPU、随机存储器[RAM](https://baike.baidu.com/item/RAM/144481)、[只读存储器](https://baike.baidu.com/item/%E5%8F%AA%E8%AF%BB%E5%AD%98%E5%82%A8%E5%99%A8/2399075)ROM、多种I/O口和中断系统、定时器/计数器等功能（可能还包括显示驱动电路、脉宽调制电路、模拟多路转换器、[A/D转换器](https://baike.baidu.com/item/A%2FD%E8%BD%AC%E6%8D%A2%E5%99%A8/4883848)等电路）集成到一块硅片上构成的一个小而完善的微型计算机系统

运算器

运算器由运算部件——[算术逻辑单元](https://baike.baidu.com/item/%E7%AE%97%E6%9C%AF%E9%80%BB%E8%BE%91%E5%8D%95%E5%85%83)（ALU）、累加器和寄存器等几部分组成。ALU的作用是把传来的数据进行算术或逻辑运算，输入来源为两个8位数据，分别来自累加器和数据寄存器。ALU能完成对这两个数据进行加、减、与、或、比较大小等操作，最后将结果存入累加器。

运算器有两个功能：

(1) 执行各种算术运算。

(2) 执行各种逻辑运算，并进行逻辑测试，如零值测试或两个值的比较。

运算器所执行全部操作都是由控制器发出的控制信号来指挥的，并且，一个算术操作产生一个运算结果，一个逻辑操作产生一个判决。

控制器

控制器由程序计数器、指令寄存器、指令译码器、时序发生器和操作控制器等组成，是发布命令的“决策机构”，即协调和指挥整个微机系统的操作。其主要功能有：

(1) 从内存中取出一条指令，并指出下一条指令在内存中的位置。

(2) 对指令进行译码和测试，并产生相应的操作控制信号，以便于执行规定的动作。

(3) 指挥并控制[CPU](https://baike.baidu.com/item/CPU" \t "_blank)、内存和输入输出设备之间数据流动的方向。

微处理器内通过内部总线把ALU、计数器、[寄存器](https://baike.baidu.com/item/%E5%AF%84%E5%AD%98%E5%99%A8" \t "_blank)和控制部分互联，并通过外部总线与外部的存储器、输入输出接口电路联接。外部总线又称为系统总线，分为数据总线DB、地址总线AB和控制总线CB。通过输入输出接口电路，实现与各种外围设备连接。

主要寄存器

（1）累加器A

累加器A是微处理器中使用最频繁的寄存器。在算术和逻辑运算时它有双功能：运算前，用于保存一个操作数；运算后，用于保存所得的和、差或逻辑运算结果。

（2）数据寄存器DR

数据寄存器通过数据总线向存储器和输入/输出设备送（写）或取（读）数据的暂存单元。它可以保存一条正在译码的指令，也可以保存正在送往存储器中存储的一个数据字节等等。

（3）指令寄存器IR和指令译码器ID

指令包括操作码和操作数。

指令寄存器是用来保存当前正在执行的一条指令。当执行一条指令时，先把它从内存中取到数据寄存器中，然后再传送到指令寄存器。当系统执行给定的指令时，必须对操作码进行译码，以确定所要求的操作，指令译码器就是负责这项工作的。其中，指令寄存器中操作码字段的输出就是指令译码器的输入。

（4）程序计数器PC

PC用于确定下一条指令的地址，以保证程序能够连续地执行下去，因此通常又被称为指令地址计数器。在程序开始执行前必须将程序的第一条指令的内存单元地址（即程序的首地址）送入PC，使它总是指向下一条要执行指令的地址。

（5）地址寄存器AR

地址寄存器用于保存当前CPU所要访问的内存单元或I/O设备的地址。由于内存与CPU之间存在着速度上的差异，所以必须使用地址寄存器来保持地址信息，直到内存读/写操作完成为止。

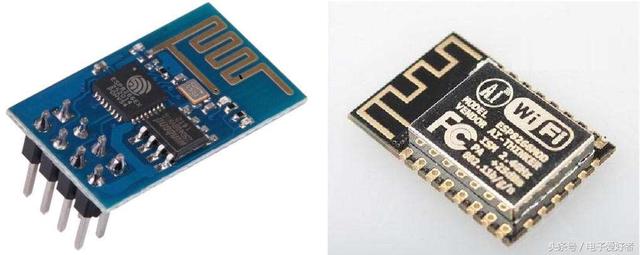
显然，当CPU向存储器存数据、CPU从内存取数据和CPU从内存读出指令时，都要用到地址寄存器和数据寄存器。同样，如果把外围设备的地址作为内存地址单元来看的话，那么当CPU和外围设备交换信息时，也需要用到地址寄存器和数据寄存器。

Arduino与51单片机的区别

比51更先进一些，编程变得简单更人性化和直观，没有单片机基础的人也能很快上手。  
　　Arduino是一款便捷灵活、方便上手的开源电子原型平台，包含硬件(各种型号的Arduino板)和软件(Arduino IDE)。它适用于爱好者、艺术家、设计师和对于"互动"有兴趣的朋友们。  
　　Arduino的核心开发团队成员包括:Massimo Banzi，David Cuartielles，Tom Igoe，Gianluca Martino，David Mellis和Nicholas Zambetti。  
　　Arduino是一个基于开放原始码的软硬件平台，构建于开放原始码simple I/O介面版，并且具有使用类似Java、C语言的Processing/Wiring开发环境。  
　　Arduino包含两个主要的部分:硬件部分是可以用来做电路连接的Arduino电路板;另外一个则是Arduino IDE，你的计算机中的程序开发环境。你只要在IDE中编写程序代码，将程序上传到Arduino电路板后，程序便会告诉Arduino电路板要做些什么了。  
　　Arduino能通过各种各样的传感器来感知环境，通过控制灯光、马达和其他的装置来反馈、影响环境。板子上的微控制器可以通过Arduino的编程语言来编写程序，编译成二进制文件，烧录进微控制器。对Arduino的编程是利用 Arduino编程语言 (基于 Wiring)和Arduino开发环境(基于 Processing)来实现的。基于Arduino的项目，可以只包含Arduino，也可以包含Arduino和其他一些在PC上运行的软件，他们之间进行通信 (比如 Flash, Processing, MaxMSP)来实现。

Arduino编程、指令、外设、外围电路都是模块化。  
51都需要自己弄。  
  
Arduino容易上手适合新手，有成就感。  
51很枯燥。  
  
Arduino价格高做一个小项目，50到100多元。  
51价格便宜5到20元就行。  
  
如果有点基础的爱好者，建议玩STM32.

ESP8266简介：三种编程方式

随着互联网的日益发展，智能家居的观念也逐渐深入人心。想要玩转智能家居，就离不开互联网，今天给大家介绍一款模块——ESP8266。小编将手把手教大家利用8266实现家电的控制。  
ESP8266 可以用来做串口透传，PWM 调控，远程控制开关：控制插座、开关、电器等。该模块有三种工作模式，大家可以根据自己的具体情况来选择：  
  
STA 模式：ESP8266 模块通过路由器连接互联网，手机或电脑通过互联网实现对设备的远程控制。

AP 模式：ESP8266 模块作为热点，手机或电脑直接与模块连接，实现局域网无线控制。

STA+AP 模式：两种模式的共存模式，即可以通过互联网控制可实现无缝切换，方便操作。  
ESP8266有几种不同的使用方式，适用于不同水平的开发工作者。下面给大家介绍：

1、使用AT指令进行操作：这是最常见的方式，也是最简单是一种方式。无需编程，使用PC端的串口助手配合简单的指令就可以实现，也可以配合单片机发送指令使用。  
  
2、LUA语言编程：这是一种单独8266编程的方式，可以不依靠单片机和串口调试软件，直接把程序编写到8266内部。

3、Arduino 开发环境编程：这个接触过Arduino的都会比较熟悉。可以直接在Arduino ide的环境下使用Arduino的开发方式进行开发。个人比较推荐这种方式，因为比较容易接受与理解。arduino相关资料也比较多。

ESP8266 for Arduino

整个ESP8266 的模块与Arduino的逻辑是这样的：ESP8266 是一个单片机， Arduino UNO 是一个单片机，我们需要解决的是ESP8266 与Arduino的通信以及传输问题。 整个步骤如下：

1. 编写ESP8266的代码 并进行烧录

2. 编写Arduino的代码并进行烧录

3. 编写Java 代码 与ESP8266 进行通信

ADC

模拟数字转换器即A/D转换器，或简称ADC，通常是指一个将[模拟信号](https://baike.baidu.com/item/%E6%A8%A1%E6%8B%9F%E4%BF%A1%E5%8F%B7/706796" \t "_blank)转变为数字信号的电子元件。通常的模数转换器是将一个输入电压信号转换为一个输出的数字信号。

WIFI通讯协议

https://blog.csdn.net/zimiao815/article/details/90375243

Ethenet和Wifi采用的协议都属于IEEE 802协议集。其中，Ethenet以802.3协议做为其网络层以下的协议；而Wifi以802.11做为其网络层以下的协议。无论是有线网络，还是无线网络，其网络层以上的部分，基本一样。  
  
        这里主要关注的是Wifi网络中相关的内容。Wifi的802.11协议包含许多子部分。其中按照时间顺序发展，主要有：  
  
        （1）802.11a，1999年9月制定，工作在5gHZ的频率范围（频段宽度325MHZ），最大传输速率54mbps，但当时不是很流行，所以使用的不多。  
  
        （2）802.11b，1999年9月制定，时间比802.11a稍晚，工作在2.4g的频率范围（频段宽度83.5MHZ），最大传输速率11mbps。  
  
        （3）802.11g，2003年6月制定，工作在2.4gHZ频率范围（频段宽度83.5MHZ），最大传输速率54mbps。  
  
        （4）802.11n，2009年才被IEEE批准，在2.4gHZ和5gHZ均可工作，最大的传输速率为600mbps。  
  
        这些协议均为无线网络的通信所需的基本协议，最新发展的，一般要比最初的有所改善。  
  
        另外值得注意的是，802.11n在MAC层上进行了一些重要的改进，所以导致网络性能有了很大的提升例如：  
  
        （\*）因为传输速率在很大的程度上取决于Channel（信道）的ChannelWidth有多宽，而802.11n中采用了一种技术，可以在传输数据的时候将两个信道合并为一个，再进行传输，极大地提高了传输速率（这又称HT-40，high through）。  
  
        （\*）802.11n的MIMO（多输入输出）特性，使得两对天线可以在同时同Channel上传输数据，而两者却能够不相互干扰（采用了OFDM特殊的调制技术）

无线通信技术

无线通信(Wireless communication)是利用电磁波信号可以在自由空间中传播的特性进行信息交换的一种通信方式。

状态机

状态机由状态寄存器和组合逻辑电路构成，能够根据控制信号按照预先设定的状态进行状态转移，是协调相关信号动作,完成特定操作的控制中心。状态机分为摩尔（Moore）型状态机和米莉（Mealy）型状态机。[1]

状态机就是状态转移图。举个最简单的例子，人有三个状态：健康，感冒，康复中。触发的条件有淋雨（t1），吃药（t2），打针（t3），休息（t4）。所以状态机就是健康-（t4）->健康；健康-（t1）->感冒；感冒-（t3）->健康；感冒-（t2）->康复中；康复中-（t4）->健康，等等。就是这样状态在不同的条件下跳转到自己或不同状态的图。

它能处理任何顺序的事件，并能提供有意义的响应——即使这些事件发生的顺序和预计的不同。有限状态机正是为了满足这方面的要求而设计的。

有限状态机是一种概念性机器，它能采取某种操作来响应一个外部事件。具体采取的操作不仅能取决于接收到的事件，还能取决于各个事件的相对发生顺序。之所以能 做到这一点，是因为机器能跟踪一个内部状态，它会在收到事件后进行更新。为一个事件而响应的行动不仅取决于事件本身，还取决于机器的内部状态。另外，采取 的行动还会决定并更新机器的状态。这样一来，任何逻辑都可[建模](https://baike.baidu.com/item/%E5%BB%BA%E6%A8%A1)成一系列事件/状态组合。

**[2]**状态机可归纳为4个要素，即现态、条件、动作、次态。这样的归纳，主要是出于对状态机的内在因果关系的考虑。“现态”和“条件”是因，“动作”和“次态”是果。详解如下：

①现态：是指当前所处的状态。

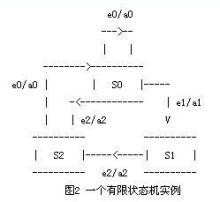
②条件：又称为“事件”，当一个条件被满足，将会触发一个动作，或者执行一次状态的迁移。

③动作：条件满足后执行的动作。动作执行完毕后，可以迁移到新的状态，也可以仍旧保持原状态。动作不是必需的，当条件满足后，也可以不执行任何动作，直接迁移到新状态。

④次态：条件满足后要迁往的新状态。“次态”是相对于“现态”而言的，“次态”一旦被激活，就转变成新的“现态”了。

有限状态机的工作原理如图1所示，发生事件(event)后，根据当前状态(cur\_state) ，决定执行的动作(action)，并设置下一个状态号(nxt\_state)。

[](https://baike.baidu.com/pic/%E7%8A%B6%E6%80%81%E6%9C%BA/6548513/0/f31fbe096b63f624ee0fb8158944ebf81a4ca352?fr=lemma&ct=single)

[](https://baike.baidu.com/pic/%E7%8A%B6%E6%80%81%E6%9C%BA/6548513/0/562c11dfa9ec8a1304d3d7f6f903918fa1ecc0df?fr=lemma&ct=single)

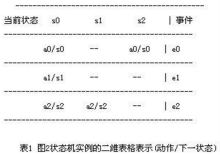
[](https://baike.baidu.com/pic/%E7%8A%B6%E6%80%81%E6%9C%BA/6548513/0/d0c8a786c9177f3e9f577f1b7ecf3bc79f3d5641?fr=lemma&ct=single)

表1 图2状态机实例的二维表格表示(动作/下一状态)

图2为一个状态机实例的状态转移图，它的含义是：

在s0状态，如果发生e0事件，那么就执行a0动作，并保持状态不变；

如果发生e1事件，那么就执行a1动作，并将状态转移到s1态；

如果发生e2事件，那么就执行a2动作，并将状态转移到s2态；

在s1状态，如果发生e2事件，那么就执行a2动作，并将状态转移到s2态；

在s2状态，如果发生e0事件，那么就执行a0动作，并将状态转移到s0态；

有限状态机不仅能够用状态转移图表示，还可以用二维的表格代表。一般将当前状 态号写在横行上，将事件写在纵列上，如表1所示。其中“--”表示空(不执行动作，也 不进行状态转移)，“an/sn”表示执行动作an,同时将下一状态设置为sn。表1和图2表示 的含义是完全相同的。

观察表1可知，状态机可以用两种方法实现：竖着写(在状态中判断事件)和横着写( 在事件中判断状态)。这两种实现在本质上是完全等效的，但在实际操作中，效果却截然 不同。

SPI

SPI是串行外设接口（Serial Peripheral Interface）的缩写，是一种高速的，全双工，同步的通信总线，并且在[芯片](https://baike.baidu.com/item/%E8%8A%AF%E7%89%87/32249)的管脚上只占用四根线，节约了芯片的管脚，同时为PCB的布局上节省空间，提供方便，正是出于这种简单易用的特性，越来越多的芯片集成了这种通信协议，比如AT91RM9200。

UART

UART是一种通用串行[数据总线](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%80%BB%E7%BA%BF)，用于[异步通信](https://baike.baidu.com/item/%E5%BC%82%E6%AD%A5%E9%80%9A%E4%BF%A1)。该总线双向通信，可以实现[全双工传输](https://baike.baidu.com/item/%E5%85%A8%E5%8F%8C%E5%B7%A5%E4%BC%A0%E8%BE%93)和接收。在嵌入式设计中，UART用于主机与辅助设备通信，如汽车音响与外接AP之间的通信，与PC机通信包括与监控调试器和其它器件，如[EEPROM](https://baike.baidu.com/item/EEPROM)通信。

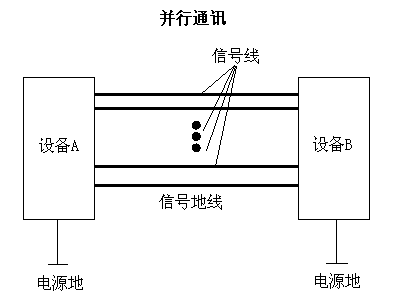
IIC

**IIC**（**Inter-Integrated Circuit**）其实是**IIC**Bus简称，所以中文应该叫**集成电路总线**，它是一种串行通信总线，使用多主从架构

串行通信与并行通信的区别

**串行通讯**

       一条信息的各位数据被逐位按顺序传送的通讯方式称为串行通讯。串行通讯的特点是：数据位传送，传按位顺序进行，最少只需一根传输线即可完成，成本低但送速度慢。串行通讯的距离可以从几米到几千米。 根据信息的传送方向，串行通讯可以进一步分为单工、半双工和全双工三种。信息只能单向传送为单工；信息能双向传送但不能同时双向传送称为半双工；信息能够同时双向传送则称为全双工。 串行通讯又分为异步通讯和同步通讯两种方式。在单片机中，主要使用异步通讯方式。



串行通讯中，两个设备之间通过一对信号线进行通讯，其中一根为信号线，另外一根为信号地线，信号电流通过信号线到达目标设备，再经过信号地线返回，构成一个信号回路。

初级读者会产生疑问：为何不让信号电流从[电源](http://www.8080.net/key_search.asp?key=74)地线返回？答案：公共地线上存在各种杂乱的电流，可以轻而易举地把信号淹没。因此所有的信号线都使用信号地线而不是电源地线，以避免干扰。

这一对信号线每次只传送1bit（比特）的信号，比如1Byte（字节）的信号需要8次才能发完。传输的信号可以是数据、指令或者控制信号，这取决于采用的是何种通讯协议以及传输状态。串行信号本身也可以带有时钟信息，并且可以通过算法校正时钟。因此不需要额外的时钟信号进行控制。

**并行通讯**

        并行通讯中，基本原理与串行通讯没有区别。只不过使用了成倍的信号线路，从而一次可以传送更多bit的信号。

并行通讯通常可以一次传送8bit、16bit、32bit甚至更高的位数，相应地就需要8根、16根、32根信号线，同时需要加入更多的信号地线。比如传统的PATA线路有40根线，其中有16根信号线和7根信号地线，其他为各种控制线，一次可以传送2Byte的数据。并行通讯中，数据信号中无法携带时钟信息，为了保证各对信号线上的信号时序一致，并行设备需要严格同步时钟信号，或者采用额外的时钟信号线。

通过串行通讯与并行通讯的对比，可以看出：串行通讯很简单，但是相对速度低；并行通讯比较复杂，但是相对速度高。更重要的是，串行线路仅使用一对信号线，线路成本低并且抗干扰能力强，因此可以用在长距离通讯上；而并行线路使用多对信号线（还不包括额外的控制线路），线路成本高并且抗干扰能力差，因此对通讯距离有非常严格的限制。