**8042的60H端口和64H端口介绍**

主板的键盘有一块专用的接口芯片，一般是采用一块单片微处理器8042（现在大多已集成在南桥或SIO里）。它控制整个键盘的工作，包括加电自检、键盘扫描码的缓冲以及与主板的通讯。

两个重要的中断:计匠网论坛" L; }+ \_. d1 y9 a" ?\* o  
   INT 09H是H/W中断，对应IRQ1，INT 16H是一个S/W中断。当键盘的一个键被按下时，键盘接口芯片根据被按下的位置，INT 09H负责把键值转换成INT16H认识的值，返回给INT 16H。INT 16H再把该值根据OS所选定的不同语系键盘而转换成相应的二进制字符传给OS或应用程序。当用户敲击键盘速度过快，使主CPU来不及处理时，则先将所键入的内容送往住存储器的键盘缓冲区，等CPU能处理时，便从缓冲区中取出，送入CPU进行分析和执行。一般在PC机的内存中安排了大约20个字符的键盘缓冲区。

8042分输入缓冲和输出缓冲，它的数据传输在I/O口60H和64H进行。基本上，I/O 64H是命令和状态口，I/O 60H是数据口，它们同时可做读写动作，在读和写时有着不同的意义。I/O 64H的bit0、1置位分别代表输出/输入缓冲满。如果发现输入缓冲满（即判断出I/O 64H[1]=1），要从I/O 60H将数据读完。BIOS在自检时如果确定输入/输出缓冲都没有问题，会发“AAH”给I/O 64H，让它自测试。等到输入缓冲空（说明上一个命令已执行完），输出缓冲满（KB控制器对自测试命令有反应），再读I/O 60H是否为“55H”（IBM PC/AT规范）。如果是，则表示KB没有问题，若等不到输出缓冲满，说明有问题。

在写命令之前，必须对I/O 64H口送一个60H的值，并等到输入缓冲空，再操作I/O 60H。同样，在读状态之前，也必须对I/O 64H口送一个20H的值，并等到输出缓冲满（表示有状态输出），再操作I/O 60H。这时，我们可以把64H看作索引口，而60H看作数据口。

A20地址线的切换：

键盘接口芯片除了接受来自键盘的信息外，还要负责A20地址线的切换，因为当CPU从实模式切换到保护模式时便是通过A20地址线的切换完成的。平常A20为“0”时，CPU工作于DOS的实模式；当A20切换为“1”时，便可进入保护模式。但由于键盘接口芯片切换A20地址线的速度不够快，目前多由主板上的芯片组以模拟方式取代，这样也就省去了一块键盘接口芯片。 & Y+ W& S& T" g! ~  
8042 端口的操作：  , O0 I. K3 n2

4 i# ds a& z通过8042芯片，可以：

1.向8042芯片发布命令（通过64h），并通过60h读取命令的返回结果（如果有的话），或通过60h端口写入命令所需的数据（如果需要的话）。

2.读取Status Register的内容（通过64h）；

 3.向8048发布命令（通过60h）

 4.读取来自于Keyboard的数据（通过60h）。这些数据包括Scan Code（由按键和释放键引起的），对8048发送的命令的确认字节（ACK）及回复数据。

 再次强调一遍，Command(命令)分为发送给8042芯片的命令和发送给8048的命令。它们是不相同的，并且使用的端口也是不相同的（分别为64h和60h）。

64h端口（读操作）8 }1 X" L; E8 y. I1 J  
      对64h端口进行读操作，会读取Status Register的内容。

  in al, 0x64 计匠网论坛" z( @" P/ A8 }  A4 V) O! o+ o  
      执行这个指令之后，AL寄存器中存放的就是Status Register的内容。

64h端口（写操作）; ?\* {) h8 r& \_9 N1 K

 向64h端口写入的字节，被认为是对8042芯片发布的命令（Command）： 写入的字节将会被存放在Input Register中； 同时会引起Status Register的Bit-3自动被设置为1，表示现在放在Input Register中的数据是一个Command，而不是一个Data； 在向64h端口写某些命令之前必须确保键盘是被禁止的，因为这些被写入的命令的返回结果将会放到Output Register中，而键盘如果不被禁止，则也会将数据放入到Output Register中，会引起相互之间的数据覆盖；9 J; |9在向64h端口写数据之前必须确保Input Register是空的（通过判断Status Register的Bit-1是否为0）

60h端口（读操作）

 对60h端口进行读操作，将会读取Output Register的内容。Output Register的内容可能是： . 来自于8048的数据。这些数据包括Scan Code，对8048发送的命令的确认字节（ACK)及回复数据。 . 通过64h端口对8042发布的命令的返回结果。 在向60h端口读取数据之前必须确保Output Register中有数据（通过判断Status Register的Bit-0是否为1）。

60h端口（写操作）% b9 N5 \# B' I# s2 I7 L( s) ^- e  
   向60h端口写入的字节，有两种可能： 1．如果之前通过64h端口向8042芯片发布的命令需要进一步的数据，则此时写入的字节就被认为是数据； 2．否则，此字节被认为是发送给8048的命令。 在向60h端口写数据之前，必须确保Input Register是空的（通过判断Status Register的Bit-1是否为0）。

**8042内容补充：**

i8042 键盘控制器

    键盘驱动直接读写 i8042 芯片，通过 i8042 间接的向键盘中的 i8048 发命令。所以对于驱动来说，直接发生联系的只有 i8042 ，因此我们只介绍 i8042 ，不介绍 i8048。

    象 i8042，i8048 这样的芯片，本身就是一个小的处理器，它的内部有自己的处理器，有自己的 Ram，有自己的寄存器，等等。

    i8042 有 4 个 8 bits 的寄存器，他们是 Status Register（状态寄存器），Output Buffer（输出缓冲器），Input Buffer（输入缓冲器），Control Register（控制寄存器）。使用两个 IO 端口，60h 和 64h。

Status Register（状态寄存器）

状态寄存器是一个8位只读寄存器，任何时刻均可被cpu读取。其各位定义如下

Bit7: PARITY-EVEN(P\_E): 从键盘获得的数据奇偶校验错误  
Bit6: RCV-TMOUT(R\_T): 接收超时，置1  
Bit5: TRANS\_TMOUT(T\_T): 发送超时，置1  
Bit4: KYBD\_INH(K\_I): 为1，键盘没有被禁止。为0，键盘被禁止。  
Bit3: CMD\_DATA(C\_D): 为1，输入缓冲器中的内容为命令，为0，输入缓冲器中的内容为数据。  
Bit2: SYS\_FLAG(S\_F): 系统标志，加电启动置0，自检通过后置1  
Bit1: INPUT\_BUF\_FULL(I\_B\_F): 输入缓冲器满置1，i8042 取走后置0  
Bit0: OUT\_BUF\_FULL(O\_B\_F): 输出缓冲器满置1，CPU读取后置0

Output Buffer（输出缓冲器）

输出缓冲器是一个8位只读寄存器。驱动从这个寄存器中读取数据。这些数据包括，扫描码，发往 i8042 命令的响应，间接的发往 i8048 命令的响应。

Input Buffer（输入缓冲器）

输入缓冲器是一个8位只写寄存器。缓冲驱动发来的内容。这些内容包括，发往 i8042 的命令，通过 i8042 间接发往 i8048 的命令，以及作为命令参数的数据。

Control Register（控制寄存器）

也被称作 Controller Command Byte （控制器命令字节）。其各位定义如下

Bit7: 保留，应该为0  
Bit6: 将第二套扫描码翻译为第一套  
Bit5: 置1，禁止鼠标  
Bit4: 置1，禁止键盘  
Bit3: 置1，忽略状态寄存器中的 Bit4  
Bit2: 设置状态寄存器中的 Bit2  
Bit1: 置1，enable 鼠标中断  
Bit0: 置1，enable 键盘中断

2个端口 0x60,0x64

驱动中把 0x60 叫数据端口  
驱动中把 0x64 叫命令端口

 命令

驱动可以直接给 i8042 发命令，可以通过 i8042 间接给 i8048 发命令。

发给i8042的命令

驱动对键盘控制器发送命令是通过写端口64h实现的，共有12条命令，分别为

20h   
准备读取8042芯片的Command Byte；其行为是将当前8042 Command Byte的内容放置于Output Register中，下一个从60H端口的读操作将会将其读取出来。

60h   
准备写入8042芯片的Command Byte；下一个通过60h写入的字节将会被放入Command Byte。

A4h   
测试一下键盘密码是否被设置；测试结果放置在Output Register，然后可以通过60h读取出来。测试结果可以有两种值：FAh=密码被设置；F1h=没有密码。

A5h   
设置键盘密码。其结果被按照顺序通过60h端口一个一个被放置在Input Register中。密码的最后是一个空字节（内容为0）。

A6h  
让密码生效。在发布这个命令之前，必须首先使用A5h命令设置密码。

AAh  
自检。诊断结果放置在Output Register中，可以通过60h读取。55h=OK。

ADh  
禁止键盘接口。Command Byte的bit-4被设置。当此命令被发布后，Keyboard将被禁止发送数据到Output Register。

AEh  
打开键盘接口。Command Byte的bit-4被清除。当此命令被发布后，Keyboard将被允许发送数据到Output Register。

C0h  
准备读取Input Port。Input Port的内容被放置于Output Register中，随后可以通过60h端口读取。

D0h  
准备读取Outport端口。结果被放在Output Register中，随后通过60h端口读取出来。

D1h  
准备写Output端口。随后通过60h端口写入的字节，会被放置在Output Port中。

D2h  
准备写数据到Output Register中。随后通过60h写入到Input Register的字节会被放入到Output Register中，此功能被用来模拟来自于Keyboard发送的数据。如果中断被允许，则会触发一个中断。