**RongReu**

Parallel Port - Kiến trúc & Hoạt động

Chào mừng các bạn đón đọc đầu sách từ dự án sách cho thiết bị di động  
  
*Nguồn:* [*http://vnthuquan.net/*](http://vnthuquan.net/)  
Tạo ebook: Nguyễn Kim Vỹ.

**MỤC LỤC**

[Parallel Port - Kiến trúc & Hoạt động](" \l "bm2)

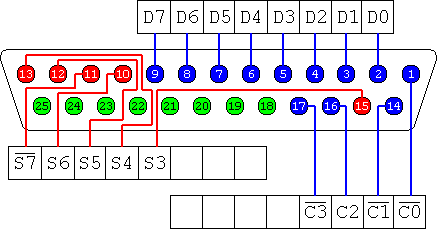
**RongReu**

Parallel Port - Kiến trúc & Hoạt động

**Giới thiệu**   
Ða số các máy vi tính đều trao đổi thông tin thông qua các ngã sau đây: Parallel port, Serial port, USB và Network card. Parallel port là một phần không thể thiếu trong việc sử dụng computer để giao tiếp với các thiết bị điện tử khác. Bài nầy chủ yếu dùng cho sinh viên hardware nhưng việc hiểu biết của nó cũng không thừa đối với các bạn học software.   
**Cấu trúc của Parallel port nhìn trên phương diện hardware**   
Parallel port bao gồm 25 pins (chân) được bố trí theo sơ đồ dưới đây, đa số giao diện đầu cắm của Parallel port đều ở dạng female:

* 8 pins dùng để gởi và nhận data (từ pin số 2 đến số 9) gọi là **DATA Port** (dân software cũng gọi như thế). Dữ liệu trao đổi qua 8 pin này được gói gọn trong 1 byte.
* 5 pins dùng để hiển thị tình trạng hoạt động của parallel port: đang bận, đang gởi/nhận thông tin...(các pin số 10-13 và pin số 15) gọi là **STATUS Port**. Dữ liệu trao đổi qua 8 pin này dùng 5 bit cao của byte.
* 4 pins dùng để điều khiển gọi là **CONTROL Port**, là các pin số 1, 14, 16 và 17 . Dữ trao đổi qua pnin này dùng 4 bit thấp của byte.
* 8 pins còn lại được dùng tùy theo ý người sử dụng. Nếu không được sử dụng thì chúng sẽ được grounded *(nối đất-thuật ngữ ngành điện?)*.

Ðây là cấu hình được thống nhất trong công nghệ vi tính và được công nhận bởi IEEE (vốn là một tổ chức lớn nhất về qui định hardware quốc tế). Bạn có thể kiểm tra lại các số pin và đánh dấu bằng cách nhìn rõ hơn vào các dây parallel port cũng như parallel port phía sau máy vi tính của bạn.   
**Vài ví dụ cho hoạt động của parallel port**   
DATA port là nơi thông tin sẽ được trao đổi từ computer đến các thiết bị khác (hai chiều). Khi lập trình ắt hẳn cũng có khi bạn nghe nói đến chuyện viết 1 program/driver cho các hardware (nếu bạn làm cho một số hãng máy in, viễn thông...).  Ở đây driver cho parallel port chính là chương trình quản lý và điều khiển quá trình trao đổi thông tin này.  
DATA port có 8 pins tức là 1 bytes. Bạn có lẽ từng nghe kỹ thuật tải thông tin qua ngã parallel port là nhanh nhất (trong quá khứ) nhưng kỳ thực nó cũng chỉ dùng có 1byte = 8 bit = 8 cái pins nầy mà thôi! Tôi sẽ trỡ lại sau trong việc bàn thảo thế nào là một sợi dây parallel tốt.   
STATUS port là nơi hiển thị các quá trình vận hành của parallel port. Một ví dụ đơn giản là giả sử bạn muốn in một bài viết ra printer (dĩ nhiên là qua ngã parallel port) nhưng khi nhấn nút "print" thì lại thấy máy vi tính hiển thị một thông báo hết giấy! Trên thực tế phía sau những hàng động nầy là một chuổi phối hợp giửa software và hardware. Khi bạn click "print" tức là bạn kích hoạt một trong những pins của CONTROL port bằng software để bảo cái printer in bài ra. Nhưng trước khi thực hiện việc in printer cũng tự biết nó hết giấy và tự kích hoạt một trong số những pins của STATUS port để báo cho computer biết là hết giấy. Kết quả là software điều khiển quá trình in kiểm tra (trước khi in) thấy được cho nên nó hiện thông báo hết giấy cho bạn. Nhiều hoạt động tương tự như printer chưa on, printer hết mực, printer bị kẹt giấy...cũng do phối hợp giửa những cái pins nầy mà ra. Tôi sẽ bàn kỹ hơn ở mức độ cấu trúc điện sau này.   
**Cấu trúc của parallel port nhìn trên phương diện software**   
Thực ra thì với dân software, họ cũng không cần biết phía parallel port sau lưng máy tính có bao nhiên pin và mỗi pin cần bao nhiêu điện, cấu trúc như thế nào...Mấy cái nầy hơi thừa cho dân software! Tất cả những gì mà một người lập trình cần biết là **address** của các pin trên parallel port là đủ! 25 pins kia sẽ được chia làm 3 phần với tên gọi là DATA port (hay là DATA register), STATUS port (hay là STATUS register), và CONTROL port (hay là CONTROL register). Mỗi port là 8 bits với address hẵn hòi. Như mô tả từ đầu, DATA port sẽ là 8 bits, STATUS port có 5 pins cho nên sẽ cộng thêm 3 bit trống để tạo một byte, tương tự như thế cho CONTROL port. Riêng phần địa chỉ cho các port nầy cũng khá là phức tạp, vì nó liên quan đến BIOS.  Nếu bạn đã hiểu cách phân bố memory của máy tính thì đơn giản hơn, còn không thì hy vọng là lối giải thích của tôi sẽ làm bạn hiểu phần nào. Ðại khái là khi máy tính bật lên (turn on) thì BIOS sẽ làm việc trước, nó sẽ tìm kiếm và định địa chỉ cho cái port trong máy của bạn. Vì BIOS không cái nào giống cái nào cho nên lối qui định địa chỉ của nó cũng khác, tuy nhiên dưới đây là một ví dụ điển hình (bạn thường thấy) trong các máy vi tính ngày nay. Những địa chỉ nầy bạn có thể thấy khi khởi động máy trong các thông số BIOS hiện ra.



**Port Address**

**Ghi chú**

3BCh - 3BFh

dùng cho prallel port vốn dính vào Video Card (cách cũ)

378h - 37Fh

khu vực memory thường dùng cho LPT 1

278h - 27Fh

khu vực memory thường dùng cho LPT 2 ....   
(nên nhớ là mỗi khoản là 8 bits, tính theo hệ hexadecimal)  
Một điều tôi muốn nhắc các bạn là những thông tin đưa ra trên đây **thường** là thay đổi tùy theo từng máy tính, một máy có thể có nhiều LPT, thông thường thì BIOS sẽ dò xem trong máy có bao nhiêu port và sẽ qui định địa chỉ cho từng port. Theo tôi thường thấy thì nếu máy bạn có hai cái parallel port (nếu bạn mua motherboard có hai parallel port) thì LPT1 sẽ được gán vào điạ chỉ 378h-37Fh (8 bits).  Nếu có LPT2 thì sẽ được gán vào địa chỉ 278h-27Fh. Riêng phần 3BCh-3BFh trong quá khứ thường được dùng khi parallel port cài sẵn trong video card. Những loại nầy đã củ rồi, cho nên nhiều BIOS sẽ gán vào LPT1 cũng không chừng. Ðiều tốt nhất là bạn vào BIOS kiểm tra là biết ngay (life was not meant to be easy!).   
Cũng lưu ý các bạn các điạ chỉ trên là port address qui định trên BIOS, khi BIOS qui định những địa chỉ trên (tắt máy vẫn còn) nó sẽ qui định kèm theo điạ chỉ lưu thông tin (tắt máy sẽ mất) cho từng port. Những địa chỉ dưới đây sẽ được dùng đa số bởi các bạn lập trình để kiểm tra xự hiện diện của parallel port trên máy bạn.

**Start Address**

**Function**

**Software Identify**

0000:0408h

LPT1 s Base Address   
Base

0000:040Ah

LPT2 s Base Address   
Base + 1

0000:040Ch

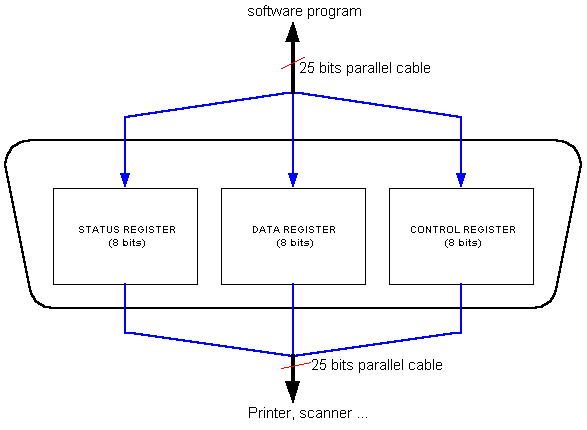
LPT3 s Base Address   
Base + 2

0000:040Eh

LPT4 s Base Address (note 1)   
Base + 3

Hãy lấy một ví dụ đơn giản về một chương trình kiểm tra vị trí parallel port trên máy bạn để minh hoạ cho lập trình điều khiển parallel port dưới đây:   
  
  
  
#include <stdio.h>   
#include <dos.h>   
void main(void)  
{  
   /\* Pointer to location of Port Addresses \*/  
   unsignedintfar\*ptraddr;  
   /\* Address of Port \*/  
   unsignedint address;          
   int a;  
   ptraddr =(unsignedintfar\*)0x00000408;  
   **for**(a =0; a <3; a++)  
   {  
      address =\*ptraddr;  
      **if**(address ==0)  
         printf("No port found for LPT%d \n", a+1);  
      **else**  
         printf("Address assigned to LPT%d is %Xh\n",  
                a+1, address);  
      \*ptraddr++;  
   }  
}   
Trương trình C nầy đơn giản là tìm đến điạ chỉ 0000:0408h và dò tìm xem có port nào tồn tại hay không.  Nó sẽ dò từ 0000:0408h đến 0000:040Dh để kiểm tra LPT1, LPT2, và LPT3.  Bạn nên nhớ mổi base address sẽ có 2 bytes (sizeof(unsigned int) = 2 bytes!).  
Như tôi mô tả phía trên, Base+1, Base+2...chẳng qua là lối gọi phổ thông trong cách lập trình. Chẳng hạn bạn kiểm tra nhà số 10, 11, 12.  Bạn có thể kiểm tra nhà số 10 sau đó nhà số 10 +1 = 11 vân vân...   
  
**Ứng dụng của việc điều khiển parallel port**   
Việc hiểu hoạt động và biết điều khiển parallel port là tối cần thiết cho các bạn đi chuyện sâu trong các kỹ nghệ hardware. Hầu hết các dụng cụ tân tiến thời nay điều liên quan đến việc dùng software để vận hành hardware, ví dụ như bạn có thể gỡi một lệnh từ máy vi tính làm cho tên lửa phóng đi, shutdown computer...điều thuộc dạng software điều khiển hardware. Và với trách nhiệm một người điều khiển nó, bạn phải thấu hiểu tất cả.  Một ví dụ đơn giản khác trong điều khiển học như điều khiển robot, nếu bạn dùng software từ máy vi tính kích hoạt một pin nào đó của cổng parallel và gỡi tới robot như mệnh lệnh, chẳng hạn đi tới phía trước, quay qua bên trái....   
**Cấu trúc đào sâu bên trong của Parallel port**   
Dưới đây là liệt kê 25 chân của parallel port với tên gọi (hardware và software) và thứ tự của từng chân.   
  
  
  
**Chân**   
  
**tên signal**   
*(dùng cho hardware)*   
**Direction/type**   
*(nhìn từ PC)*   
  
  
**Tên signal và thứ tự  
của bit**   
*(dùng cho software)*   
**Normal signal line function**   
  
1   
-STROBE   
OC/Pullup   
Control register bit 0   
kích hoạt thông báo gỡi hoặc nhận data, 0 là đọc, 1 là viết   
  
2   
D0   
hai chiều   
Data register bit 0   
bit 0 chứa data   
  
3   
D1   
hai chiều   
Data register bit 1   
bit 1 chứa data   
  
4   
D2   
hai chiều   
Data register bit 2   
bit 2 chứa data   
  
5   
D3   
hai chiều   
Data register bit 3   
bit 3 chứa data   
  
6   
D4   
hai chiều   
Data register bit 4   
bit 4 chứa data   
  
7   
D5   
hai chiều   
Data register bit 5   
bit 5 chứa data   
  
8   
D6   
hai chiều   
Data register bit 6   
bit 6 chứa data   
  
9   
D7   
hai chiều   
Data register bit 7   
bit 7 chứa data   
  
10   
-ACK   
Input   
Status register bit 6   
Pulsed low by printer to acknowledge data byte  
Rising (usually) edge causes IRQ if enabled   
  
11   
BUSY   
Input   
Status register bit 7   
kích hoạt khi printer đang bận (busy)   
  
12   
NOPAPER   
Input   
Status register bit 5   
kích hoạt khi printer hết giấy   
  
13   
SELECTED   
Input   
Status register bit 4   
kích hoạt khi printer đang hoạt động   
  
14   
-AUTOFEED   
OC/Pullup   
Control register bit 1   
kích hoạt thông báo data đã sẵn sàng để đọc hoặc viết   
  
15   
-ERROR   
Input   
Status register bit 3   
kích hoạt khi printer bị lổi (vì nhiều lý do)   
  
16   
-INITIALIZE   
OC/Pullup   
Control register bit 2   
kích hoạt để printer reset lại vị trí ban đầu   
  
17   
-SELECT   
OC/Pullup   
Control register bit 3   
kích hoạt để đánh dấu printer nhận được valid address   
  
18   
Ground   
  
  
  
  
...   
Ground   
  
  
chân (18-25) bỏ trống, dùng tùy ý   
  
25   
Ground   
Thêm một hình minh hoạ tổng thể hoạt động của parallel port.

**Làm sao để truyền dữ liệu tới parallel port bằng software đây?**   
Chắc các bạn học lập trình không xa lạ gì với lệnh **outport()** và **inport()** của C. Nếu bạn viết **outportb(0x378,0xff);** trong một chương trình C thì nó sẽ gởi giá trị 0xff (hệ hexa) = 128 (hệ decimal) = 11111111 (hệ binary) ra data register của parallel port (LTP1).  Theo giá trị vừa tính thì "11111111" nghĩa là 8 chân của data register trên parallel port sẽ có dòng điện 5volts hiện hữu. Nếu bạn là dân hardware, bạn có thể đo dòng điện tại các chân của parallel port (nên nhớ là với high frequency). Riêng với các bạn software, một khi đã hiểu sự hoạt động của cổng parallel port, bạn có thể hiểu rõ hơn cách hoạt động của các driver software.   
RongReu [(thuongem5phut@hotmail.com](mailto:thuongem5phut@hotmail.com?subject=[DDTH] - Tute: Parallel Port))  
Viết cho Diễn đàn tin học



Lời cuối: Cám ơn bạn đã theo dõi hết cuốn truyện.  
Nguồn: http://vnthuquan.net  
Phát hành: Nguyễn Kim Vỹ.  
  
Nguồn: diendantinhoc.net  
Được bạn: mickey đưa lên  
vào ngày: 21 tháng 8 năm 2004