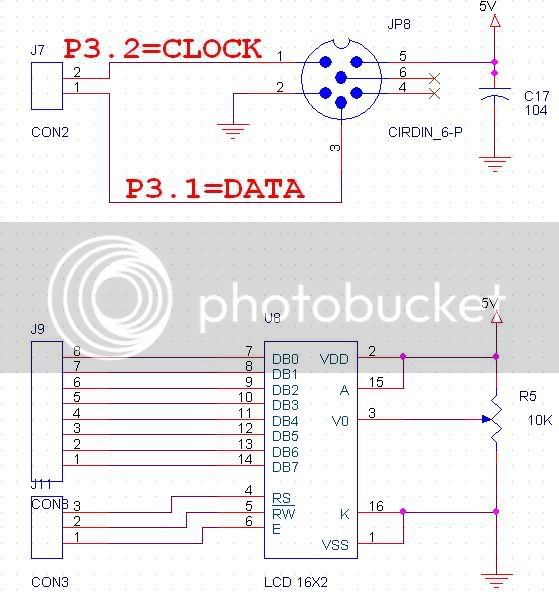
Giao tiếp keyboard pc với 8051

20-10-2008, 10:52

*   
  Tụi mình đang làm project về Giao tiếp Bàn phím máy tính với VĐK để nhập nội dung hiển thị cho LCD,đã lấy được mã ASCII của các ký tự vào VĐK dùng ngắt ngoài P3.2, nhưng mà không thoát khỏi ngắt để về chương trình main được.  
    
  Có ai biết về chủ đề này xin chỉ bảo dùm tụi mình với nhé!  
    
  Đây là mạch nguyên lý của tụi mình (mạch board thí nghiệm 89 chỉ thêm phần LCD và cổng sp2 giao tiếp bàn phím máy tính):  
  +Chân P3.1 nối chân DATA của SP2 bàn phím máy tính  
  +Chân P3.2 nối chân CLOCK của sp2 bàn phím (dùng làm ngắt để bắt đầu lấy mã phím ở chân DATA)  
    
    
  **Nhắc lại**: Tín hiệu gửi từ bàn phím lên VDK là tín hiệu nối tiếp theo thứ tự gửi: bit Start, 8bit data (từ LSB đến MSB), bit Parity, cuối cùng là bit Stop. Sau đó còn có 1byte có giá trị F0H được gửi từ bàn phím lên vdk để báo là phím đã được nhả (sau khi bấm).  
    
  Mỗi xung Clock (Chu kỳ khoảng 35 đến 40 micro giây) có tác dụng gửi 1 bit, trong đó một nửa chu kỳ là để chuẩn bị bit từ chân DATA sp2, còn một nửa chu kỳ là để chờ cho VĐK lấy bit đó từ chân DATA. Như vậy ta viết hàm delay400 (400 micro giây) để chờ cho bàn phím gửi xong byte 0f0h(bỏ qua 8bit này vì ta không dùng đến nó) rồi mới tiếp tục cho phép ngắt trở lại)  
    
  Tài liệu tham khảo: [***MODULE SP2\_VDK\_LCD***](http://www.mediafire.com/?l4q4zugiyjy)  
    
  Còn đây là code ASM của bọn mình (dùng mã phím A là 1Ch để điều khiển led trên P0), nhưng kô đổi trạng thái p0 được mà VDK lại bị treo (mình nghĩ là nó chưa thoát khỏi ngắt)  
    
  org 000h  
  jmp main  
  org 003h  
  jmp ngat  
    
  main:  
  mov r4,#3  
  mov tmod,#02h  
  mov th0,#-200 ;khoi tao timer cho ham delay400  
    
  mov ie, #81h ;Cho phep ngat ngoai  
  setb it0 ;Cho phep ngat cạnh  
  cjne R3,#1ch,mainx1  
  lcall choptatp0  
  jmp main  
  mainx1:  
  lcall choptatp02  
  jmp main  
    
  ngat:  
  clr ex0 ; ngat ngoai thu 0  
  mov r2, #8  
    
  jb p3.2, $ ;cho bit start => bo qua  
  jnb p3.2, $  
    
  nhan2: jb p3.2, nhan2 ;cho canh xuong cua xung ck  
  jnb p3.2, $  
  mov c, p3.1  
  rrc a ;doc bit lsb vao a7 truoc  
  djnz r2, nhan2 ;doc cho đến hết 8 bit data  
  mov r3, a ;dua ma cua phim nhan ra port 2  
    
  jb p3.2, $  
  jnb p3.2, $ ;chờ bit parity => bo qua  
    
  jb p3.2, $  
  jnb p3.2, $ ;chờ bit stop => bo qua  
    
  lcall delay400 ;chờ bàn phím gửi byte 0f0h => bo qua  
  setb p3.2  
  setb ex0 ;cho phép ngắt trở lại  
    
    
  RETI  
    
  delay400:  
  setb tr0  
  loop: jnb tf0,loop  
  djnz r4,loop  
  clr tr0  
  ret  
    
  choptatp0:  
  mov p0,#0ffh  
  lcall delay  
  mov p0,#0  
  lcall delay400  
  ret  
    
  choptatp02:  
  setb p0.2  
  lcall delay  
  clr p0.2  
  lcall delay  
  ret  
    
  delay:  
  mov r2, #255  
  nhan8: mov r6, #255  
  nhan9: djnz r6, nhan9  
  djnz r2, nhan8  
  ret  
    
  end

Mình nghĩ vấn đề của bạn ở hàm delay400:  
Trong hàm này bạn sử dụng r4 với giá trị setting là 3, tiếc rằng, cứ mỗi vòng lặp main nó lại reload giá trị này một lần.  
Khi bạn gọi hàm delay400, lệnh **djnz r4,loop**sẽ giảm r4, tuy nhiên (theo mình nghĩ) r4 không bao giờ về 0 do vòng lặp main luôn reload lại giá trị 3. Kết quả là bạn bị mắc kẹt ở vòng loop.  
Để tránh trường hợp này bạn phải đẩy lệnh mov r4,#3 ra khỏi vòng lặp main. Tuy nhiên trước khi gọi hàm delay bạn phải reload lại r4.  
Mà này, delay có 400ms mà cũng phải gọi đến timer à. Mình delay 1 giây còn không cần timer nữa mà. VD:  
  
DELAY1S: MOV R5,#20  
LOOPSEC: DJNZ R5,STARTMS  
SJMP ENDDELAY  
STARTMS: MOV R6,#200  
LOOPMS: DJNZ R6, STARTMMS  
SJMP LOOPSEC  
STARTMMS: MOV R7,#250  
LOOPMMS: DJNZ R7, LOOPMMS  
SJMP LOOPMS  
ENDDELAY: RET  
  
Ý tưởng của bạn cũng khá hay, nhưng bàn phím PS/2 đã dần bị thay thế bởi USB rồi, bạn nên tìm các giao tiếp với USB thì hay hơn  
Chúc vui.

delay400:  
setb tr0  
loop: jnb tf0,loop  
djnz r4,loop  
clr tr0  
ret  
  
Có thể chương trình delay chạy không đúng vì bạn chưa "CLR TF0" mà đã "JNB TF0,LOOP". Nếu đúng như vậy thì VDK sẽ ngắt tiếp tục do những xung clock của Keyboard gửi kèm theo byte số 2 và byte số 3. Kết quả là ngắt hoài ko chịu nghỉ hehe...  
Ko dùng ngắt cũng được mà

ngat không thoát được là do bạn sử dụng lệnh jnb p3.2, $  
việc bạn tính toán là sẽ có 11 bit la không có gì chắc chắn  
vì co thể là 12 bít thi sao  
lúc đó se không thoát được ngắt  
theo kinh nghiệm của mình bạn nên dung một hàn đếm thời gian trên mức của chân p3.2  
nếu không hợp lý thì thoát khỏi chương trinh ngắt  
  
  
khi nhấn một phím thì PC sẽ phát một mã nhấn phím, nhấn giữ lâu thì mã đó phát liên tục  
khi nhả phím sẽ phát mã nhả phím đổng thời phát lại mã của phím đó  
  
đoạn code này sừ dụng ngắt ngoài đẻ nhận data phím  
sau đó dung hàm Converkeyboard(unsigned char nDataPS2) để chuyển về mã ký tự

Code:

#define phim\_rong 0xff

#define p\_DataPS2 PIND.4

unsigned char m\_CurselPS2;

unsigned char m\_DataPS2;

bit m\_break;

bit m\_thuong\_hoa;

unsigned char unshifted[48][2]= {

0x1c,'a',

0x32,'b',

0x21,'c',

0x23,'d',

0x24,'e',

0x2b,'f',

0x34,'g',

0x33,'h',

0x43,'i',

0x3b,'j',

0x42,'k',

0x4b,'l',

0x3a,'m',

0x31,'n',

0x44,'o',

0x4d,'p',

0x15,'q',

0x2d,'r',

0x1b,'s',

0x2c,'t',

0x3c,'u',

0x2a,'v',

0x1d,'w',

0x22,'x',

0x35,'y',

0x1a,'z',

0x45,'0',

0x16,'1',

0x1e,'2',

0x26,'3',

0x25,'4',

0x2e,'5',

0x36,'6',

0x3d,'7',

0x3e,'8',

0x46,'9',

0x0e,'`',

0x4e,'-',

0x55,'=',

0x5d,'\\',

0x54,'[',

0x5b,']',

0x4c,';',

0x52,'\'',

0x41,',',

0x49,'.',

0x4a,'/',

0,0

};

// Shifted characters

unsigned char shifted[48][2]= {

0x1c,'A',

0x32,'B',

0x21,'C',

0x23,'D',

0x24,'E',

0x2b,'F',

0x34,'G',

0x33,'H',

0x43,'I',

0x3b,'J',

0x42,'K',

0x4b,'L',

0x3a,'M',

0x31,'N',

0x44,'O',

0x4d,'P',

0x15,'Q',

0x2d,'R',

0x1b,'S',

0x2c,'T',

0x3c,'U',

0x2a,'V',

0x1d,'W',

0x22,'X',

0x35,'Y',

0x1a,'Z',

0x45,')',

0x16,'!',

0x1e,'@',

0x26,'#',

0x25,'$',

0x2e,'%',

0x36,'^',

0x3d,'&',

0x3e,'\*',

0x46,'(',

0x0e,'~',

0x4e,'\_',

0x55,'+',

0x5d,'|',

0x54,'{',

0x5b,'}',

0x4c,':',

0x52,'"',

0x41,'<',

0x49,'>',

0x4a,'?',

0,0

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//---------------------------------------------------------------------

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

unsigned char Converkeyboard(unsigned char nDataPS2)

{

unsigned char i;

if(nDataPS2 == 0xf0) //ma bao nha phim

{

m\_break = 1;

return phim\_rong;

}

if(m\_break) //xu ly ma nha phim

{

if(nDataPS2 == 0x12 || nDataPS2 == 0x59)//neu la ma Shift

{

m\_thuong\_hoa = 0;

}

m\_break = 0;

return phim\_rong;

}

else

{

if(nDataPS2 == 0x12 || nDataPS2 == 0x59)//nhan phim Shift

{

m\_thuong\_hoa = 1;

return phim\_rong;

}

if(nDataPS2 == 0x29) return ' ';

if(!m\_thuong\_hoa)

{

for(i = 0; unshifted[i][0]!=nDataPS2 && unshifted[i][0]; i++);

if (unshifted[i][0] == nDataPS2) return unshifted[i][1];

}

else

{

for(i = 0; unshifted[i][0]!=nDataPS2 && shifted[i][0]; i++);

if (unshifted[i][0] == nDataPS2) return shifted[i][1];

}

return phim\_rong;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//---------------------------------------------------------------------

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

interrupt [EXT\_INT0]//su kien ngat ngoai 0

void CK\_PS2(void)//ham xu ly ngat ngoai 0

{

m\_CurselPS2++;

if(m\_CurselPS2==1) //bit START

{

if(p\_DataPS2)m\_CurselPS2=0; //text START

}

else

if(m\_CurselPS2>1&&m\_CurselPS2<10)

{

m\_DataPS2 = (m\_DataPS2 >> 1); //quay trai m\_DataPS2 (m\_DataPS2.7 = 0)

if(p\_DataPS2) m\_DataPS2 = m\_DataPS2 | 0x80; // Store a ’1’

}

else

if(m\_CurselPS2>10) //bit STOP

{

if(p\_DataPS2)

{

// makytu = Converkeyboard(m\_DataPS2);

// tuy ban xu ly ma ky tu

}

m\_CurselPS2=0;

} //bo qua Parity

}

bác nào giúp em phát , em mò mấy ngày hôm nay k xong,  
em dùng ngắt ngoài 0 chân P3\_2 , nối vào chân xung clock của bàn phím  
chân DATA nối vào chân P2\_7;  
theo em hiểu thì keyboard gửi data đến khi ta ấn phím và bít START luôn =0;  
không hiểu em sai ở đâu mà nó không thể đọc mã scan của keyboard đựoc , các bác xem hộ em cái  
  
void Clock (void )interrupt 0 //su dung ngat ngoai 0, ham xua ly xung clock , data =P2\_7  
{  
dem++; // biến đếm số bit  
switch (dem)  
{  
case 1 :  
{  
if (P2\_7)// neu bit stat =1 ,chan P2\_7 nối vào chân DATA của keyboard  
{  
dem=0; // xóa biến đếm thứ tự bít để bắt đầu lại  
}  
else { dem++;} // néu bit START khác 0  
  
};break;  
case 2: //bit thu 1  
{  
nhap = P2\_7; // nhap và nháp là biến trung gian, scancode là biến lưu trữ mã scan của keyboard  
nhap1=nhap;  
scancode=scancode +nhap1;  
nhap=0; //xoa bien nhap  
nhap1=0;  
};break;  
case 3: //bit thứ 2  
{  
nhap = P2\_7;  
nhap1= nhap << 1 ; // bít thứ 2 thì dịch sang trái 1 bít  
scancode=scancode +nhap1;  
nhap=0; //xoa bien nhap  
nhap1=0;  
};break;  
case 4: // bit thu 3  
{  
nhap = P2\_7;  
nhap1= nhap << 2; // bit thứ 3 thì dịch sang trái 2 bit  
scancode=scancode +nhap1;  
nhap=0; //xoa bien nhap  
nhap1=0;  
  
};break;  
case 5:  
{  
nhap = P2\_7;  
nhap1= nhap << 3;  
scancode=scancode +nhap1;  
nhap=0; //xoa bien nhap  
nhap1=0;  
  
};break;  
case 6:  
{  
nhap = P2\_7;  
nhap1= nhap << 4;  
scancode=scancode +nhap1;  
nhap=0; //xoa bien nhap  
nhap1=0;  
  
};break;  
case 7:  
{  
nhap = P2\_7;  
nhap1= nhap << 5;  
scancode=scancode +nhap1;  
nhap=0; //xoa bien nhap  
nhap1=0;  
  
};break;  
case 8:  
{  
nhap = P2\_7;  
nhap1= nhap << 6;  
scancode=scancode +nhap1;  
nhap=0; //xoa bien nhap  
nhap1=0;  
  
};break;  
case 9:  
{  
nhap = P2\_7;  
nhap1= nhap << 7;  
scancode=scancode +nhap1;  
nhap=0; //xoa bien nhap  
nhap1=0;  
};break;  
case 10: //bit chan le  
{  
; // bit chẵn lẻ thì bỏ qua  
};break;  
case 11:  
{  
dao\_nguoc(scancode); chú thích \*  
hien\_thi\_bit(scancode) ; chú thích \*\*  
delay(1000);  
dem=0;  
}  
} }  
  
\* vì khi keyboard gửi đa ta luôn gửi bit nhỏ (LSB) nhất dầu tiên mà ta lại dịch bít sang trái  
nên byte sẽ bị đảo ngựoc như B0 B1 B2 B3 B4 B5 B6 B7;  
DÙNG HÀM dao\_ngưoc để sắp xếp lại cho đúng B7 B6 B5 B4 B3 B2 B1 B0  
\*\* HÀM hien\_thi hiển thị giá trị nhị phân của scancode , hic nhưng em toàn nhận đựoc giá trị 00000000, em đã kiểm tra hàm này hoạt đong tốt, chỉ có hàm ngắt là không biết sai đâu  
bác nào rồi chạy rồi cho em xin ít kinh nghiệm  thank bác