Sau đây là chút bí quyết về Quang báo thần chưởng của tai hạ. Viết quang báo không cần phải dùng tới ram ngoài làm chi. Quan trọng là thủ thuật thi triển nội công của mình. Cần noi gương Trương tam Phong, lấy nhu thắng cưong. Viết quang báo không nen xài tới Keil C, Bascom chân kinh vì sẽ tiêu hao bộ nhớ hơi nhiều, xài Hợp ngữ cho chắc. Chúc các bạn mới học quang báo thành công. Trong lúc nhớ bạn gái, viết bài thơ chơi. Nếu có sai xin Nguyễn Bính bỏ qua cho. Bài viết chắc có nhiều sơ sót, mong các cao nhân dị sỹ, các minh chủ, đường chủ diễn đàn bỏ qua cho.  
  
  
dong equ 34h ;Khai báo số ảnh hiển thị / 1 câu  
timeout equ 35h ;Tạo trì hoãn giữa các lần hiển thị  
org 0  
main:  
mov r3,#00h  
mov r5,#00h  
mov p2,#0ffh  
  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
vong:  
mov timeout,#20h  
mov dong,#134  
mov dptr,#Topic  
mov r3,#0  
lcall loopmain ;Cho chạy câu mở đàu  
;\*\*\*\*\*\*\*\*\*\*\*  
mov timeout,#10h  
mov dong,#113  
mov dptr,#cau1  
mov r3,#0  
lcall loopmain ;Cho chạy câu 1  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#95  
mov dptr,#cau2  
mov r3,#0  
lcall loopmain ;Cho chạy câu 2  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#105  
mov dptr,#cau3  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#87  
mov dptr,#cau4  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#131  
mov dptr,#cau5  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#95  
mov dptr,#cau6  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#65  
mov dptr,#cau7  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#65  
mov dptr,#cau8  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#111  
mov dptr,#cau9  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#93  
mov dptr,#cau10  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#117  
mov dptr,#cau11  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#137  
mov dptr,#cau12  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#123  
mov dptr,#cau13  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#71  
mov dptr,#cau14  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#119  
mov dptr,#cau15  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#73  
mov dptr,#cau16  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#123  
mov dptr,#cau17  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#55  
mov dptr,#cau18  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#79  
mov dptr,#cau19  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#83  
mov dptr,#cau20  
mov r3,#0  
lcall loopmain  
;\*\*\*\*\*\*\*\*\*\*\*\*  
mov dong,#216  
mov dptr,#cau21  
mov r3,#0  
lcall loopmain  
  
jmp vong  
  
loopmain:  
mov r6,#0ffh  
mov r7,#0ffh  
mov 10h,#0ffh  
mov r5,#0  
inc r3  
mov a,r3  
inc dptr  
cjne a,dong,loop  
ret  
loop:  
  
mov a,r5  
movc a,@a+dptr  
dec r6  
djnz r6,tiep  
mov r6,#0ffh  
dec r7  
tiep:  
  
mov p0,r5  
mov p2,a  
lcall delay  
mov p2,#00h  
  
lcall dich  
  
djnz r7,loop  
jmp loopmain  
  
delay: mov r1,#03h  
lai1: mov r2,timeout  
lai2: djnz r2,lai2  
djnz r1,lai1  
ret  
  
dich:  
inc r5  
cjne r5,#32,exitdich  
mov r5,#0  
exitdich:  
ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;\*\*\*\*\*\*\*\*\*\*MA MA TRAN\*\*\*\*\*\*\*\*\*\*\*\*\*  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Topic:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 10h,10h,38h,0feh,38h,10h,10h,00h ;Sao  
db 00h  
db 10h,10h,38h,0feh,38h,10h,10h,00h ;Sao  
db 00h,00h,00h,00h  
db 04h,7fh,84h,40h,00h ;t  
db 0ffh,08h,08h,70h,80h,40h,00h ;h  
db 70h,88h,88h,74h,0ch,00h ;ô  
db 00h,00h  
db 70h,88h,88h,00h ;c  
db 70h,88h,88h,70h,00h ;o  
db 08h,0f8h,08h,08h,0f0h,00h ;n  
db 00h,00h  
db 70h,88h,88h,00h ;c  
db 70h,88h,8ch,72h,00h ;où  
db 70h,88h,88h,00h ;c  
db 00h,00h,00h,00h  
db 10h,10h,38h,0feh,38h,10h,10h,00h ;Sao  
db 00h  
db 10h,10h,38h,0feh,38h,10h,10h,00h ;Sao  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
  
cau1:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h ;G  
DB 00H,00H,00H  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 0ffh,00h ;I  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 00h,00h  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 01h,02h,0fch,02h,01h ;Y  
cau2:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h ;G  
DB 00H,00H,00H  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 00h,00h  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 00h,00h  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 0ffh,89h,89h,89h,89h,00h ;E  
cau3:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 46h,89h,89h,89h,72h,00h ;S  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
db 0ffh,11h,31h,51h,8eh,00h ;R  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
db 00h,00h  
db 3fh,40h,80h,40h,3fh,00h ;V  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
DB 00H,00H  
cau4:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00h ;T  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00h ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
DB 00H,00H  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
cau5:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 01h,01h,0ffh,01h,01,00h ;T  
db 00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 00h,00h  
db 0ffh,11h,11h,11h,0eh,00h ;P  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
cau6:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 01h,01h,0ffh,01h,01,00h ;T  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
db 00h,00h  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 00h,00h  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 00h,00h  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 0ffh,89h,89h,89h,89h,00h ;E  
cau7:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,89h,89h,89h,76h,00h ;B  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 00h,00h  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00h ;T  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
cau8:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,89h,89h,89h,76h,00h ;B  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 00h,00h  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00h ;T  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
cau9:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 01h,01h,0ffh,01h,01,00h ;T  
db 00h,00h  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
db 00h,00h  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 00h,00h  
db 0e3h,14h,08h,14h,0e3h,00h ;X  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 00h,00h  
db 3fh,40h,80h,40h,3fh,00h ;V  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
db 00h,00h  
cau10:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 00h,00h  
db 0e3h,14h,08h,14h,0e3h,00h ;X  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
db 00h,00h  
db 08h,0ffh,89h,81h,7eh,00h ;-D  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 01h,01h,0ffh,01h,01,00h ;T  
cau11:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 08h,0ffh,89h,81h,7eh,00h ;-D  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 00h,00h  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 7eh,81h,81h,81h,42h,00H ;C  
db 00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
db 0ffh,11h,31h,51h,8eh,00h ;R  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,00h ;I  
db 00h,00h  
db 08h,0ffh,89h,81h,7eh,00h ;-D  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
cau12:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 01h,01h,0ffh,01h,01,00H ;T  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
db 0ffh,00h ;I  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00h ;T  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G

cau13:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 01h,01h,0ffh,01h,01,00H ;T  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
db 00h,00h  
db 3fh,40h,80h,40h,3fh,00h ;V  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 00h,00h  
db 0ffh,18h,24h,42h,81h,00h ;K  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 7fh,80h,80h,80h,7fh,00H ;U  
cau14:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 7eh,81h,91h,91h,72h,00h ;G  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
db 00h,00h  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 00h,00h  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 0ffh,89h,89h,89h,89h,00h ;E  
cau15:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 7eh,81h,91h,91h,72h,00h ;G  
db 0ffh,00h ;I  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
db 0ffh,11h,31h,51h,8eh,00h ;R  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 01h,01h,0ffh,01h,01,00H ;T  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
db 0ffh,00h ;I  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 00h,00h  
db 01h,02h,0fch,02h,01h,00h ;Y  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 7fh,80h,80h,80h,7fh,00H ;U  
cau16:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,18h,24h,42h,81h,00h ;K  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 0ffh,00h ;I  
db 00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
DB 0ffh,08h,08h,08h,0ffh,00H ;H  
db 0ffh,00h ;I  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 00h,00h  
db 0e3h,14h,08h,14h,0e3h,00h ;X  
db 0fch,12h,11h,12h,0fch,00H ;A  
cau17:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 3fh,40h,80h,40h,3fh,00h ;V  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 00h,00h  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
DB 0ffh,08h,08h,08h,0ffh,00h ;H  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 00h,00h  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 01h,02h,0fch,02h,01h,00h ;Y  
db 00h,00h  
db 7eh,81h,91h,91h,72h,00h ;G  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,11h,11h,11h,0eh,00h ;P  
db 00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,00h ;I  
cau18:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 7eh,81h,81h,81h,42h,00H ;C  
db 00h,00h  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 01h,02h,0fch,02h,01h,00h ;Y  
cau19:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 00h,00h  
db 7eh,81h,91h,91h,72h,00h ;G  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,00h ;I  
cau20:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
  
db 08h,0ffh,89h,81h,7eh,00h ;-D  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 01h,01h,0ffh,01h,01,00H ;T  
db 00h,00h  
db 01h,01h,0ffh,01h,01,00H ;T  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,00h ;I  
db 00h,00h  
db 0ffh,89h,89h,89h,76h,00h ;B  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
cau21:  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
db 7eh,81h,81h,81h,7eh,00h ;O  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 00h,00h  
db 0ffh,80h,80h,80h,80h,00h ;L  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 00h,00h  
db 7eh,81h,81h,81h,42h,00H ;C  
db 7fh,80h,80h,80h,7fh,00H ;U  
db 0fch,12h,11h,12h,0fch,00H ;A  
db 00h,00h  
db 0ffh,11h,31h,51h,8eh,00h ;R  
db 0ffh,00h ;I  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 0ffh,04h,18h,20h,0ffh,00H ;N  
db 7eh,81h,91h,91h,72h,00h ;G  
db 00h,00h  
db 0ffh,89h,89h,89h,89h,00h ;E  
db 0ffh,02h,04h,02h,0ffh,00H ;M  
db 00h,00h  
db 02h,01h,0b1h,09h,06h,00h ;?  
db 0bfh,00h ;!  
db 0bfh,00h ;!  
db 0bfh,00h ;!  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
db 00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00 h,00h,00h,00h  
end

Ai bảo chăn trâu là khổ ...!?

Ha ha , vậy là LINH đã thành công rồi, và anh chàng Diode cũng đam mê cái này lắm đó Linhnc308 ah!  
Lâu nay buôn tẩu giang hồ, tha phương làm ăn quên chuyện QB mất rồi. Bữa nào mình Post mấy cái Demo QB lên anh em xem!

Rất cảm ơn anh Phi! Em sẽ đợi mấy bản của anh để tham khảo.  
Khi làm quang báo tôi cũng đã tìm hiểu rất nhiều giải pháp để giải quyết vấn đề hiển thị và cập nhật thông tin cho bảng đèn.  
- Về các quét thì như mọi người đều biết, ta có thể dùng cách quét hàng hay cột. Tôi thường dùng quét cột, bởi trong pham vi nghiên cứu với các bảng LED 8x8 ghép nối tiếp với nhau(của tôi là 8x32). Nếu bảng có số cột nhiều hơn thì ta nên chuyển sang quét hàng để đảm bảo thời gian quét và các LED không bị nháy.  
- Khi ghép nối các bảng lại với nhau, ta nên chia thành các module nhỏ và ghép lại. Ta có thể dùng chuẩn giao tiếp nối tiếp SPI để ghép lại  
Với cái bảng 8x32 tôi có làm theo 3 cách:  
1. Dùng hai con 74154 để quét cột. Các hàng tôi dùng 8 con A1013 (dòng 1A) để đẹm dòng, các LED sáng đều. (thời gian trễ tôi để 2ms). Nhưng tôi nhận thấy việc dùng 74154 cũng hay nhưng ko kinh tế(12K/1con74154, giá này tôi mua ở chợ trời), và việc phát triển ra bảng to thì ta gặp phải vấn đề cần thêm chân để chọn chip do đó mạch sẽ có thêm một số mạch phụ dùng để chọn chip(như 74138 chảng hạn, hay dùng chính 74154 thì ta ghép được 16 chip). tôi đã thử dùng với 74138, nói chung kết quả cũng được. Đấy là một vài nhận xét của tôi khi làm với 74154. Khi nộp cho thầy tôi sẽ dung mạch này.  
2. Vẫn quét cột nhưng ta dùng 74595(vào nối tiếp, ra song song và nối tiếp).  
Các hàng tôi vẫn dùng tran để đệm dòng. Tôi thấy cách này hay hơn nhiều vì ta tiết kiệm được tối đa chân IC (Cái ý tưởng này tui phải cảm ơn anh BinhAnh nhiều nhiều, nhờ có nhưng lời chỉ dẫn của anh!). Ta dùng kiểu giao tiếp SPI. Mỗi một bảng 8x8 ta cần 1 con 595, và ta ghép nối tiếp các con này lại với nhau. SHcp và STcp thì đều nối chung với nhau thành một đường CLK đưa đến VDK, DATAin của con thứ nhất nối với VDK, các con sau thì dataout con này nối với datain của con kia, cứ như vậy và OK mạch chạy ngon lành mà lại đơn giản. Tính ra ta chỉ cần dùng AT2051 hay PIC16F(84,88) là đủ, nếu thiều bộ nhớ thì thêm một con EEPROM ngoài.  
3. Ta dùng các IC chuyên dụng cho các ma trận LED như con max6953 tôi sample từ maxim, hỗ trợ quét 4 bảng 5x7, giao tiếp I2C và có cả loại SPI. Hay có thể dùng ICM7219/7221. Thực ra con này ta dùng để quét 8 đèn LED 7 thanh, nhưng cũng có thể dùng để quét cho một bảng 8x8 như là con 595, giao tiếp SPI và I2C. Tôi có một cái đề tài quang báo dùng con 7221, tôi down được từ trang [www.woe.de.vu](http://www.woe.de.vu/) (ledfont). Đọc và tìm hiểu code họ viết cho 8051 ta thu được khá nhiều điều thú vị, như việc để cho chữ chạy họ dùng RAM làm đệm và dùng phương pháp dịch RAM (là đoạn mã có nhã shiftram) và đưa ra bảng LED. Dùng cả ROM và EEPROM để lưu ký tự, có giao tiếp RS232 để cập nhật EEPROM. Tôi sẽ đưa đoạn code đó lên. Anh em cung phân tích.  
Trên đây là 3 phương án tôi đã tìm hiểu. Tất nhiên chỉ cho việc nghiên cứu và viết code thôi, còn thực tế nều làm theo kiểu công nghiêp thì còn nhiêug vấn đề cần nghiên cứu(như anh BA đang làm cái bảng cao 16 ý, mình cũng sẽ làm một cái...ka ka ka)  
Chúc thành công!

Thank mafd đã góp ý. Đúng là bảng QB có rất nhiều thiếu sót. Mình áy náy nhất là việc dùng Decoder 74154 để quét. Khi đó sẽ tốn không gian mạch (Con này khá bự) , và cả tón tiền nữa. Tại vì đây là lần đầu tiên làm QB mà. Ý mình là nên dùng con thanh ghi dịch 16 bit 74576 để dịch. Khi đó , muốn ghép nhiều ma trận lại với nhau chỉ cần ghép A15 con trước với A0 con sau, các chân xóa được nối chung với nhau. KHi đó muốn quét cột chỉ cần 2 chân của PORT chứ không như n chân quét 2 mũ n cột nữa.Với lại con 74245 cũng cấp dòng không đủ cho các led.  
Áy náy nữa là với phần mềm như thế không thể hiển thị được các kí tự ngẫu nhiên, các số trong đồng hồ.  
Bạn hãy trình bày giải thuật của việc hiển thị các kí tự ngẫu nhiên được không.

Chào cả nhà. Không ngờ chủ đề quang báo lại đông vui đến vậy.  
Hôm nay đây tôi đưa lên cái files ASM tôi viết cho con AT89S52, dung 1 con 74154 để quét cột. Chương trình này đọc trực tiếp từ ROM và đưa ra bảng LED. Anh em xem qua va cho nhận xét. Đây là bản đầu tiên, do đó lập trình còn đơn giản. Trong bản nâng cấp của nó tôi dùng RAM làm bộ đệm màn hình và EEPROM để lưu font, do đó ta có thể kết nối máy tính để thay đổi nội dung.  
Đang nghiên cứu fung bàn phím hồng ngoại để thay đổi nội dung mà chưa nghĩ ra. Các cao thủ cho xin ý kiến. thx  
  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*  
; LED matrix display Using 74HC154  
; Microprocessor = AT89S52  
; Designer: linhnc308  
; P1 la cong gui du lieu cua 8 hang, 16 cot duoc noi voi cac dau ra cua 74154  
; Cac chan A,B,C,D cua 74154 ket noi voi chan cua P3  
; P3.0 = A  
; P3.1 = B  
; P3.2 = C  
; P3.3 = D  
; Chuong trinh se cho chay chu I Love U, sau do la bang ky tu ASCII  
; Cac chu duoc ma hoa theo ma tran 5x7  
; Viet bang phan mem Reads51 cua Rigel. [www.rigelcorp.com](http://www.rigelcorp.com/)  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
#include <sfr51.inc>  
ORG 0000H  
LJMP MAIN  
  
ORG 0030H  
MAIN:  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
; Hien thi chu I Love U  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
MOV DPTR,#MYDATA ; Dua con tro Du lieu toi bang du lieu  
LEDDISPLAY:  
ACALL DISPLAY ; Goi chuong trinh con tao hien thi tren bang LED  
INC DPTR ; Tang con tro DPTR de tao hieu ung chu chay  
CLR A ; Xoa con tro A cho viec lay du lieu tiep theo  
MOVC A,@A+DPTR ; Lay du lieu vao thanh ghi A  
CJNE A,#'#',LEDDISPLAY ; Kiem tra xem da den cuoi bang du lieu chua.  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
; Chuong trinh hien thi bang chu cai va cac ky tu ASCII  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
NOP  
MOV DPTR,#MYDATA2 ; Dua con tro den bang chua ky tu ASCII  
ALPHATABLE:  
ACALL DISPLAY  
INC DPTR  
CLR A  
MOVC A,@A+DPTR  
CJNE A,#'#',ALPHATABLE ; Kiem tra ket thuc chua  
LJMP MAIN ; Quay tro lai chuong trinh chinh  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;CHUONG TRINH CON HIEN THI DU LIEU  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
DISPLAY:  
MOV R5,#8 ; Bien quy dinh toc do chu chay tren man hinh  
INIT:  
MOV R2,00H ; Bien dem khi quet ky tu va lay du lieu  
LOOP: ; Dong thoi no cung la gia tri gui den 74154 de quet cot  
MOV A,R2  
MOVC A,@A+DPTR  
MOV P1,A ; Gui du lieu ra cac hang  
MOV P3,R2 ; Gui du lieu toi 74154 de kich hoat cot  
ACALL DELAY ; Tao tre 1mS  
INC R2 ; Tang R2  
CJNE R2,#16,LOOP ; Kiem tra xem da quet du 16 cot chua  
DJNZ R5,INIT ; Lap lai cho den khi R5 =0  
RET  
; Toc do chu chay tren man hinh phu thuoc nhieu vao viec quy dinh gia tri cua R5  
; va thoi gian tre ma ta tao ra trong ham Delay  
DELAY:  
MOV R7,#1  
DL1: MOV R6,#250  
DL2: NOP ; 1MC  
NOP  
DJNZ R6,DL2 ; 2MC TOTAL = 4\*255uS = 1000uS ~ 1mS  
DJNZ R7,DL1  
RET  
  
MYDATA:  
; Đoạn này dài quá tôi cắt bỏ, các bạn xem trong files gửi kèm nhé!  
END

 Về các quét thì như mọi người đều biết, ta có thể dùng cách quét hàng hay cột. Tôi thường dùng quét cột, bởi trong pham vi nghiên cứu với các bảng LED 8x8 ghép nối tiếp với nhau(của tôi là 8x32). Nếu bảng có số cột nhiều hơn thì ta nên chuyển sang quét hàng để đảm bảo thời gian quét và các LED không bị nháy.  
- Khi ghép nối các bảng lại với nhau, ta nên chia thành các module nhỏ và ghép lại. Ta có thể dùng chuẩn giao tiếp nối tiếp SPI để ghép lại  
Với cái bảng 8x32 tôi có làm theo 3 cách:  
1. Dùng hai con 74154 để quét cột. Các hàng tôi dùng 8 con A1013 (dòng 1A) để đẹm dòng, các LED sáng đều. (thời gian trễ tôi để 2ms). Nhưng tôi nhận thấy việc dùng 74154 cũng hay nhưng ko kinh tế(12K/1con74154, giá này tôi mua ở chợ trời), và việc phát triển ra bảng to thì ta gặp phải vấn đề cần thêm chân để chọn chip do đó mạch sẽ có thêm một số mạch phụ dùng để chọn chip(như 74138 chảng hạn, hay dùng chính 74154 thì ta ghép được 16 chip). tôi đã thử dùng với 74138, nói chung kết quả cũng được. Đấy là một vài nhận xét của tôi khi làm với 74154.  
Do không sử dụng được trích dẫn nên em chép đoạn trên lại  
Theo em cách nầy thuận tiện hơn,nếu dùng 74138 để ĐK 1 led 8X8 và dùng 74154 để ĐK 74138 hoạt động,nếu ta quét led theo tọa độ theo cột cần hiển thị thì bảng quang báo có thể thực hiện được với các thông báo phức tạp(chữ ở đầu dòng và chử ở cuối dòng,thậm chí có thể hiển thị hình ảnh nếu có phương pháp xác định tọa độ các điểm ảnh ).Em đã làm thử,mạch sáng đều không có hiện tượng led sáng led mờ.

ox nay ngung hoat dong roi sao rat mong duoc bac linh va anh em giup do giai thich dum em voi cach bac truy xuat de lay du lieu tu bang ky tu la the nao vay  
; Chuogn trinh quang bao hien thi ban tin len bang LED  
; MCU = AT89S52  
; ICs = 74154 (decoder/demux 4-of-16)  
; Software: READ51s  
; Desinger: [linhnc308@yahoo.com](mailto:linhnc308@yahoo.com)  
; BK design - DT8K47. DHBKHN  
#include <sfr52.inc>  
  
; Flag bit area in RAM 20h  
flag\_bit equ 1  
  
index\_l equ 21h  
index\_h equ 22h ;con tro chi so cua ham doc ky tu  
offset\_l equ 23h ;con tro cua ham tra bang ma ASCII  
offset\_h equ 24h  
number\_char\_l equ 25h  
number\_char\_h equ 26h  
;read\_ok equ 26h  
  
org 0  
ljmp main  
  
org 0030h  
main:  
; dinh nghia cac bien  
mov r0,#20h  
mov a,0ffh  
clr\_ram\_loop:  
mov @r0,a ; clear memory 40..127(internal Ram)  
inc r0  
cjne r0,#128,clr\_ram\_loop  
mov r4,#0  
  
acall so\_ky\_tu  
main\_pro:  
clr a  
mov r0,a  
mov index\_l,#0  
mov index\_h,#0  
clr flag\_bit  
get\_char:  
acall doc\_ky\_tu  
end\_get\_char:  
check\_ascii:  
clr c  
subb a,#32  
jnc check\_ok  
clr a  
check\_ok:  
mov b,#5  
mul ab  
mov offset\_l,a  
mov offset\_h,b  
  
copy\_2\_ram:  
mov r1,#127  
mov r0,#126  
shift\_ram:  
mov a,@r0  
mov @r1,a  
dec r1  
dec r0  
cjne r1,#111,shift\_ram  
;acall display  
cjne r4,#5,copy\_code\_ascii  
  
next\_char:  
mov 111,#0ffh ; tao diem ngan cac giau cac chu  
mov r4,#0  
setb flag\_bit ; Bat co Flag =1 cho biet co mot tu da duoc copy  
inc index\_l  
mov a,#00h  
cjne a,index\_l,no\_inc\_counter\_h  
inc index\_h  
no\_inc\_counter\_h:  
acall doc\_ky\_tu  
  
check\_ascii2:  
clr c  
subb a,#32  
jnc check\_ok  
clr a  
check\_ok2:  
mov b,#5  
mul ab  
mov offset\_l,a  
mov offset\_h,b  
ljmp disp  
; Start copy text to ram  
copy\_code\_ascii:  
mov dptr,#ascii\_code  
mov a,offset\_l  
add a,dpl  
jnc no\_inc\_dph  
inc dph  
no\_inc\_dph:  
mov dpl,a  
mov a,offset\_h  
add a,dph  
mov dph,a  
mov a,r4 ; Doc ma LMD thu i(1..5) cua ky tu  
movc a,@a+dptr  
mov 111,a ; LED-column on the right side  
inc r4  
disp:  
acall display  
jnb flag\_bit,jump\_point  
clr flag\_bit  
mov a,number\_char\_l  
cjne a,index\_l,jump\_point  
mov a,number\_char\_h  
cjne a,index\_h,jump\_point  
ljmp main  
jump\_point:  
ljmp copy\_2\_ram  
  
  
  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*Cac chuong trinh con\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
display:  
mov r1,#127  
mov r5,#2 ; Bien quy dinh toc do chu chay, ket hop cung voi  
ini\_disp: ; thoi gian cua ham Delay  
mov a,r1  
mov r0,a  
mov r3,#0  
loop:  
mov a,@r0  
mov p1,a  
mov p3,r3  
acall delay  
dec r0  
inc r3  
cjne r3,#16,loop  
djnz r5,ini\_disp  
ret  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
so\_ky\_tu:  
mov dptr,#number\_char  
clr a  
movc a,@a+dptr  
mov number\_char\_h,a  
mov a,#1  
movc a,@a+dptr  
mov number\_char\_l,a  
ret  
  
doc\_ky\_tu:  
mov dptr,#my\_text  
mov a,dpl  
add a,index\_l  
jnc ko\_tang\_dph  
inc dph  
ko\_tang\_dph:  
mov dpl,a  
mov a,dph  
add a,index\_h  
mov dph,a  
movc a,@a+dptr  
ret  
  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;HAM TAO THOI GIAN TRE KHI QUET COT  
DELAY: MOV R7,#2  
DL3: MOV R6,#250  
DL2: NOP ; 1 MC  
NOP ; 1 MC  
DJNZ R6,DL2 ; 2 MC TOTAL=(1+1+2)\*250uS = 1000uS = 1mS  
DJNZ R7,DL3 ; 2 MC TOTAL=2\*1mS + 2MC ~ 2mS  
RET  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
DELAY2: MOV R7,#4  
DJNZ R7,$  
RET  
  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
number\_char: ; Dinh nghia so chu toi da. so 16bit tuc = 65000 tu  
db 01h,40h  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
my\_text: ; toi da la 65000 tu nhung neu thay doi code thi co the dung het ROM thi thoi  
DB " Dien tu8-K47 Khoa Dien Tu Vien Thong Truong Dai Hoc Bach Khoa" ;64 ky tu dau tien  
DB " Ha Noi. Bo mon Thiet ke mach Logic.Giang Vien Nguyen Nam Quan" ;64 ky tu tiep theo  
db " De tai: Mach Quang bao ung dung Vi dieu khien ho 8051.(AT89S5" ;64  
db "2) Sinh vien Thiet ke: Nguyen Chi Linh. Email:linhnc308@yahoo." ;64  
db "com Time is 5:05 Ngay 21 thang 10 nam 2005. 1\_2,3:4'5>6<7 8 9 0" ;64 ky tu  
; la 35 chu cai trong 1 lan hien thi(giua cac hu co mot khoang trang)  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
my\_text2:  
DB " "  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;BANG MA HOA KY TU  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
ascii\_code:  
DB FFH,FFH,FFH,FFH,FFH;SPACE 0  
DB FFH,FFH,A0H,FFH,FFH;! 1  
DB FFH,FFH,F8H,F4H,FFH;' 2  
DB EBH,80H,EBH,80H,EBH;# 3  
DB DBh,D5H,80H,D5H,EDH;$ 4

|

*Chính xác! font chữ lưu trên máy tính nhưng bạn làm thế nào để ghép được những chữ đó đúng theo font chữ mình gõ  
VD: với kiểu gõ telex khi bạn gõ aa->â vậy ta phải dùng thuật toán thế nào để lấy ra được cái font chứa chữ "â" đó (ta luôn phải so sánh 2 kí tự liền nhau ???). Mặt khác khi thêm các dấu thì ghép font thế nào?*

Khi đó trên PC chúng ta sẽ xử lý trước các kí tự 2byte truoc khi truyền.

*Như vậy ta phải tạo 1 bảng font không phải la 26 chữ cái nữa mà phải nhiều hơn rất nhiều đúng không vậy. Còn với lập trình VB có thể giải quyết được các vấn đề đó không(tôi chưa dùng VB bao giờ) Mong các cao thủ chỉ giúp*

Bộ font có tất cả 255-32 kí tự có thể được hiển thị, VB không cần phải xử lý gì thêm.

*Thank bác Binhanh! Ý kiến của bác rất đúng, Sơ đồ đó tôi vẽ thử thôi thực chất khi làm mạch tôi điều khiển cả chân 4 của 74138 khi đó giải quyết tốt hiện tượng bóng mờ.  
Còn về bảng chữ 16 mang được sự trả lời của các bác!*

co cach nay tuy chua toi uu ( ton nhieu bye) nhung voi ban chac cung huu ich.  
Luu bo font chu 16 hang trong bo nho, sau do doc ra de hien thi. cach duoi day se cho luu duoc luon 3 mau sac tron voi nhau - duoc 16 kieu mau:  
1.Dinh dia chi dau font trong bo nho  
2. 1 font duoc luu trong bo nho gom: 1 byte len + data  
1 byte len: chieu dai chu ( so cot cua chu).  
data chua cac diem font cua ky tu, so byte cua ky tu be voi ky tu to phai bang nhau, vi du: chi I va chu M dieu co so byte nhu nhau--the nen moi bao cach nay hoi ton bo nho.  
Nhu vay khoang cach font cua cac ky tu la deu nhau = 1+ data  
  
3. tu PC truyen xuong VXL de luu noi dung chu, moi ky tu mat 2 byte: 1 byte ma font ( .vntime or vni-time) va mau sac, 1 byte ma ky tu. byte ma font va mau sac: 4 bit cho mau sac, 4 bit cho ma font ( luu duoc 16 bo font).  
Bang chua ma pha mau sac ta de trong bo nho chuong trinh cua vxl  
  
Khi ta doc noi dung, den ky tu nao ta deu doc 2 byte de tach font va mau sac, byte ma ky tu dem nhan (\*) voi so byte data + dia chi font se chi den dung vung font cua ky tu can tra.  
dua vao chieu dai ta de o dau moi font ky tu ta doc duoc ky tu, dem tron voi cac byte mau sac da tinh duoc, sau do dem out ra bang LED chu, the la OK  
  
can 1 cach khac rat Ok nhung toi giu lam bi kip

 Nếu trên PC xu ly het toan bo thi duoc cai loi la phia lap trinh cho VDk nho gon, dung luong Code bé.  
- Con de lap trinh duoi vi dieu khien thi lap trinh tren PC de, nhung duoi vi dieu khien hoi bi cong kềnh va dung luong code co the len den chu kbye.  
- Hien tai Phi vẫn chọn cách cuối. Ưu điểm nỗi bat la de, nhung doi lai lap trinh cho vi dieu khien hoi bi nhieu.  
Va phuong phap van lam theo hoahoc89x

*Đừng chơi kiểu hiển thị đến đâu truyền đến đấy, nếu vậy rs232 làm sao chịu được tốc độ.  
hãy làm theo cách sau: trong bộ nhớ con IC chủ lưu nhiều nội dung, mỗi nội dung có nhiều câu, mỗi câu có nhiều font chữ và mỗi chữ có nhiều mầu sắc-> trước khi hiển thị 1 nội dung, IC chủ giải mã ra nội dung để hiển thị ra LED, nó truyền cả gói nầy đồng thời cho các IC tớ-> cả họ nhà nó đồng thời bảo nhau ( đồng bộ) hiển thị phần của mình-> thế là được 1 bảng to theo ý thích-> đã làm ,chạy rẩt OK*

đệ vẫn chưa hiểu ý huynh lắm  
giả sử đệ có 2 module 16\*32 mỗi module được điều khiển bởi một con vdk riêng biệt . hai con này sẽ được nối với con trung tâm bằng cổng nối tiếp  
dữ liệu chính được lưu tại con trung tâm . các con vdk trên các module chỉ để hiển thị phần dữ liệu mà con chủ đẩy xuống thôi.  
cứ mỗi module cần có (16\*32)/8=64 byte dữ liệu  
vậy hai module sẽ cần có 128 byte để hiển thị chữ font 16  
đệ truyền nối tiếp từ con master xuống với tốc độ 19200bit/s tương ứng với khoảng 1920 byte/s vậy để truyền hết 128 byte thì mất một khoảng thời gian là 128/1920=0.066s=66,66 ms. trong thời gian này các vdk slave bị ngắt hàm quét led hiển thị để thu nhận dữ liệu với 66,66ms bị ngắt chắc chắn sẽ bị giựt.  
đệ chỉ hiểu như vậy , biết chắc là sai nhưng vẫn nói ra những gì mình nghĩ mong các huynh chỉ cho đệ cái hướng đúng đắn hơn.

Tính như trên thì đúng nhưng không phải nhất thiết phải mất thời gian lớn như vậy. Chúng ta chỉ cần xét 1 byte địa chỉ trong 128 byte cần gởi là được như vậy thời gian lớn lắm cũng không chỉ 100us mà thôi. Những byte còn lại (N-1) nếu không đúng thì không cần nhận, do đó không phải mất ~66ms như bạn đã nói. Còn làm thế nào bạn hãy tự làm lấy.

*mấy huynh xem hộ đệ cái thiết kế này với :  
có vấn đề gì không nhỉ*

\* EEFROM dùng để lưu nội dung thông tin, RAM6264 dùng để xử lý kiểu hiển thị, DS1307 hiển thị ngày giời.  
Theo mình thì mạch trên chạy được, tuy nhiên cần thiết kế lại phần Driver để có thể sau này mở lớn, có thể đến 512 cột/32 dòng. Con A1013 còn hơi nhỏ, nên thay bằng TIP127- dòng khá lớn, ưu điểm nữa là kích bằng áp. Và một góp ý nữa là nên dùng DS12887 - con này chạy chính xác hơn nhiều( theo kinh nghiệm của mình thôi).  
Thay vì dùng 74595 & 2803 thì hiện nay đã có chip thay thế con này rồi TPIC595, nó như con 74595 bình thường nhưng ngõ ra có thể đạt 350 đến 500mA.

Mạch quảng báo 89c51+74595+74154. 8x128  
  
chuong trình này tôi viết thử cho chạy mô phỏng trên Proteus  
có File Code va mophong kèm mong mọi người góp ý  
$mod51  
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx  
; chuong trinh xuat du lieu len bang ma tran  
; ngang gom co 16 Led matrix 8x8 = 16\*8 =128 cot  
; cao gom co2Led matrix 8x8 =2\*8 =16 hang  
; Xuat tin hieu ra cot dung IC 74HC595 gom N con mat noi tiep  
;Quet hang dung 1 con IC 74HC154  
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxx  
; A bit p1.0  
; B bit p1.1  
; C bit p1.2  
; D bit p1.3  
E bit p1.4  
Clock bit p2.0  
Din bit p2.1  
Lacth bit p2.2  
Colume equ 30h  
bodem equ 31h  
contro equ 32h  
Row equ 33h  
ABCD equ 34h  
Length equ 35h  
contro1 equ 36h  
org 0000h  
  
KhoiTao:  
mov Length, #196  
mov Colume, #128  
mov bodem, #0  
mov contro, #0  
mov row, #0  
mov ABCD, #00h  
mov dptr, #DuLieu  
mov contro1, #0  
mov r1, #5  
  
  
;''''''''''''''''''''''''''''''''''''''''''''''''' '''''''''''''''''''''''''''''''''''''''''''''''''' '''''''''  
Main:  
call Xoamanhinh  
Call Xuat\_Frame  
call delay  
inc Row  
inc ABCD  
mov a, Row  
cjne a, #8, main  
mov ABCD, #00h  
mov Row, #0  
djnz r1, main  
mov r1,#5  
inc contro  
mov a, contro  
cjne a, Length,main  
jmp Khoitao  
;''''''''''''''''''''''''''''''''''''''''''''''''' '''''''''''''''''''''''''''''''''''''''''''''''''' ''''''''''''  
DATA\_Cot:  
  
mov r2, colume  
mov r3, Row  
clr Lacth  
dichcot: mov a, contro  
add a, bodem  
movc a, @a+dptr  
clr Clock  
call LuaDong  
mov Din, c  
setb Clock  
inc bodem  
djnz r2, dichcot  
mov bodem, #0  
mov r2, colume  
setb Lacth  
  
ret  
  
;-----------------------------------------------------------  
Xuat\_Frame:  
  
call DATA\_cot  
mov p1, ABCD  
  
  
ret  
;----------------------------------------------------------  
XoaManHinh:  
  
mov r2, colume  
clr Lacth  
xoa: clr clock  
clr Din  
setb clock  
djnz r2, xoa  
setb Lacth  
  
  
ret  
;----------------------------------------------------------  
LUADONG:  
cjne r3, #0, dong1  
mov c, acc.0  
mov din, c  
call nhay12  
sjmp het  
dong1: cjne r3, #1, dong2  
mov c, acc.1  
mov din, c  
call nhay10  
sjmp het  
dong2: cjne r3, #2, dong3  
mov c, acc.2  
mov din, c  
call nhay8  
sjmp het  
dong3: cjne r3, #3, dong4  
mov c, acc.3  
mov din, c  
call nhay6  
sjmp het  
dong4: cjne r3, #4, dong5  
mov c, acc.4  
mov din, c  
call nhay4  
sjmp het  
dong5: cjne r3, #5, dong6  
mov c, acc.5  
mov din, c  
nop  
nop  
sjmp het  
dong6: cjne r3, #6, dong7  
mov c, acc.6  
mov din, c  
sjmp het  
dong7: cjne r3, #7, het  
mov c, acc.7  
mov din, c  
het: ret  
;-------------------------------------------------  
NHAY:  
nhay14: nop  
nop  
nhay12: nop  
nop  
nhay10: nop  
nop  
nhay8: nop  
nop  
nhay6: nop  
nop  
nhay4: ret  
  
  
  
  
;---------------------------------------------------  
DELAY:  
mov r7,#80  
laplai: mov r6, #10  
djnz r6, $  
djnz r7, laplai  
ret  
Delay1:  
mov r5, #2  
mov r4, #2  
tt: djnz r4, $  
djnz r5, tt  
ret  
;--------------------------------------------------------  
DULIEU:  
db 7fh,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
  
db 00h,41h,7fh,41h,00h ;I  
db 00h,00h  
db 00h,41h,7fh,40h,00h ;l  
  
db 38h,44h,44h,44h,38h ;o  
db 00h  
db 1ch,20h,40h,20h,1ch ;v  
db 00h  
db 38h,54h,54h,54h,18h ;e  
db 00h,00h,00h  
db 03h,04h,78h,04h,03h ;Y  
db 00h  
db 20h,54h,54h,54h,38h ;a  
db 00h  
db 7ch,04h,78h,04h,78h ;m  
db 00h  
db 20h,54h,54h,54h,38h ;a  
db 00h  
db 7fh,08h,04h,04h,78h ;h  
db 00h  
db 20h,54h,54h,54h,38h ;a  
  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
  
db 01h,01h,7fh,01h,01h ;T  
db 00h  
db 34h,4ah,4ah,4ah,34h ;ô  
db 00h  
db 00h,44h,7dh,40h,00h ;i  
db 00h  
db 00h,04h,3fh,44h,44h ;t  
db 00h  
db 7fh,08h,04h,04h,78h ;h  
db 00h  
db 00h,44h,7dh,42h,01h ;í  
  
db 00h,38h,44h,44h,44h ;c  
db 00h  
db 7fh,08h,04h,04h,78h ;h  
db 00h,00h,00h  
db 03h,04h,78h,04h,03h ;Y  
db 00h  
db 20h,54h,54h,54h,38h ;a  
db 00h  
db 7ch,04h,78h,04h,78h ;m  
db 00h  
db 20h,54h,54h,54h,38h ;a  
db 00h  
db 7fh,08h,04h,04h,78h ;h  
db 00h  
db 20h,54h,54h,54h,38h ;a  
  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
db 00h,00h,00h,00h,00h ;space  
end

Attached Files

* [74595.zip.zip (39.8 KB, 1791 lượt xem)](http://www.dientuvietnam.net/forums/filedata/fetch?id=1325796)