

```

;
; IRcmd v0.1 -----
;
; Comandament a distància infrarroig programable.
; Llegeix comandes d'un teclat i modula un LED IR en conseqüència.
;
; Nota: el programa actual està fet per a controlar una TV Toshiba, però amb
; les especificacions és fàcil canviar-lo per a controlar qualsevol altre
; aparell IR que funcioni amb una modul.lació d'aproximadament 40KHz.
;
; Guillem Cantallops Ramis, DNI 78211957C.
; UIB, 2003.09.29.
;
; -----
;

processor 16f84a          ; Microchip PIC16F84A
radix hex                ; hexadecimal per defecte
#include <p16f84a.inc>    ; declaracions utils de registres, etc.
__config 0x3ffa         ; no cp, no pwrte, no wd, oscil.lador hs
__idlocs 0xffff         ; no id

; constants -----

ram      equ 0x0c        ; inici GPRs (SRAM)

; variables als GPRs (SRAM) -----

tecla    equ ram        ; de 0x00 a 0x0F, tecla entrada pel teclat
codi     equ ram+1      ; codi de sortida per al port RA (1+4 LEDs)
cnt1     equ ram+2      ; comptadors per als retards
cnt2     equ ram+3
cnt3     equ ram+4
cnt      equ ram+5      ; comptador per als períodes
bit      equ ram+6      ; comptador per a fer recorreguts de bits
byte     equ ram+7      ; byte a enviar com a part d'una comanda

; punt d'entrada -----

org 0x00

start    call    setup    ; inici programa -----

loop     call    kbscan    ; bucle principal -----
         btfss   STATUS, C ; carry? (i.e. tecla pulsada?)
         goto    loop     ; no -> tornam a començar
         movwf  tecla     ; si -> guardam tecla actual
         call   kbscan    ; tornam a mirar el teclat
         btfss   STATUS, C ; carry? (i.e. tecla pulsada?)
         goto    loop     ; no -> tornam a començar
         subwf  tecla, W  ; si -> comparam tecla actual amb anterior

```

```

btfs STATUS, Z ; zero? (i.e. tecles iguals?)
goto loop ; no -> tornam a començar
call sendcmd ; si -> enviam comanda
goto loop ; hem acabat, tornam a començar

```

```

setup ; inicialitzacions -----
bsf STATUS, RP0
clrf TRISA ; tot RA<4:0> son sortides (LEDs: 1 IR, 4 vermells)
bcf STATUS, RP0
clrf PORTA ; netejam tot RA (sortides a zero -> LEDs on)

bsf STATUS, RP0
movlw B'00001111' ; RB<7:4> out (files), RB<3:0> in (columnes)
movwf TRISB
bcf STATUS, RP0
clrf PORTB ; netejam tot RB (no importa gaire com quedi aquí)

return

```

```

kbscan ; scan del teclat -----
bsf STATUS, C ; inicialment suposam que trobarem tecla

movlw B'11101111' ; primera fila
movwf PORTB
nop
nop
btfs PORTB, 3
goto k1
btfs PORTB, 2
goto k2
btfs PORTB, 1
goto k3
btfs PORTB, 0
goto kA

movlw B'11011111' ; segona fila
movwf PORTB
nop
nop
btfs PORTB, 3
goto k4
btfs PORTB, 2
goto k5
btfs PORTB, 1
goto k6
btfs PORTB, 0
goto kB

movlw B'10111111' ; tercera fila
movwf PORTB
nop
nop
btfs PORTB, 3
goto k7
btfs PORTB, 2
goto k8
btfs PORTB, 1
goto k9

```

```
    btfs    PORTB, 0
    goto    kC

    movlw   B'01111111' ; quarta fila
    movwf   PORTB
    nop
    nop
    btfs    PORTB, 3
    goto    kF
    btfs    PORTB, 2
    goto    k0
    btfs    PORTB, 1
    goto    kE
    btfs    PORTB, 0
    goto    kD

    bcf    STATUS, C ; si arribam aqui, no hem trobat tecla
    return

k0
    movlw 0x02 ; transposicions degudes al cablejat!
    return

k1
    movlw 0x0f
    return

k2
    movlw 0x00
    return

k3
    movlw 0x0e
    return

k4
    movlw 0x07
    return

k5
    movlw 0x08
    return

k6
    movlw 0x09
    return

k7
    movlw 0x04
    return

k8
    movlw 0x05
    return

k9
    movlw 0x06
    return

kA
    movlw 0x0d
    return

kB
    movlw 0x0c
    return

kC
    movlw 0x0b
    return

kD
    movlw 0x0a
```

```
    return
kE    movlw 0x03
    return
kF    movlw 0x01
    return

sendcmd    ; enviam comanda (realment, paquet on/off) -----

    call genera_codi    ; calculam l'estat dels LEDs
    call leadin        ; enviam capçalera

    movlw 0x00          ; tecla 0
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_0
    movlw 0x01          ; tecla 1
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_1
    movlw 0x02          ; tecla 2
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_2
    movlw 0x03          ; tecla 3
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_3
    movlw 0x04          ; tecla 4
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_4
    movlw 0x05          ; tecla 5
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_5
    movlw 0x06          ; tecla 6
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_6
    movlw 0x07          ; tecla 7
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_7
    movlw 0x08          ; tecla 8
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_8
    movlw 0x09          ; tecla 9
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_9
    movlw 0x0A          ; tecla A
    subwf tecla, W
    btfsc STATUS, Z
    goto sendcmd_volup
    movlw 0x0B          ; tecla B
    subwf tecla, W
```

```
    btfst STATUS, Z
    goto sendcmd_voldown
    movlw 0x0C          ; tecla C
    subwf tecla, W
    btfst STATUS, Z
    goto sendcmd_chup
    movlw 0x0D          ; tecla D
    subwf tecla, W
    btfst STATUS, Z
    goto sendcmd_chdown
    movlw 0x0E          ; tecla E
    subwf tecla, W
    btfst STATUS, Z
    goto sendcmd_chrtn
    movlw 0x0F          ; tecla F
    subwf tecla, W
    btfst STATUS, Z
    goto sendcmd_pwr
sendcmd_pwr
    call cmd_pwr
    goto cmdcontinue
sendcmd_volup
    call cmd_volup
    goto cmdcontinue
sendcmd_voldown
    call cmd_voldown
    goto cmdcontinue
sendcmd_chup
    call cmd_chup
    goto cmdcontinue
sendcmd_chdown
    call cmd_chdown
    goto cmdcontinue
sendcmd_chrtn
    call cmd_chrtn
    goto cmdcontinue
sendcmd_0
    call cmd_0
    goto cmdcontinue
sendcmd_1
    call cmd_1
    goto cmdcontinue
sendcmd_2
    call cmd_2
    goto cmdcontinue
sendcmd_3
    call cmd_3
    goto cmdcontinue
sendcmd_4
    call cmd_4
    goto cmdcontinue
sendcmd_5
    call cmd_5
    goto cmdcontinue
sendcmd_6
    call cmd_6
    goto cmdcontinue
sendcmd_7
    call cmd_7
    goto cmdcontinue
```

```
sendcmd_8
    call cmd_8
    goto cmdcontinue
sendcmd_9
    call cmd_9
    goto cmdcontinue
cmdcontinue
    call leadout           ; enviam cua
    movlw 0x02           ; descans d'uns 100ms aproximadament
    call delay3
    return

leadin
    movlw 0xad           ; capçalera: 346T ON, 162T OFF -----
    movwf cnt

liloop1
    call leds_on
    call periode
    call leds_off
    call periode
    decfsz cnt, F
    goto liloop1
    movlw 0xa2
    movwf cnt

liloop2
    call periode
    decfsz cnt, F
    goto liloop2
    return

one
    movlw 0x0b           ; digit 1: 22T ON, 62T OFF -----
    movwf cnt

oloop1
    call leds_on
    call periode
    call leds_off
    call periode
    decfsz cnt, F
    goto oloop1
    movlw 0x1f
    movwf cnt

oloop2
    call periode
    decfsz cnt, F
    goto oloop2
    return

zero
    movlw 0x0b           ; digit 0: 22T ON, 20T OFF -----
    movwf cnt

zloop1
    call leds_on
    call periode
    call leds_off
    call periode
    decfsz cnt, F
```

```

    goto zloop1
    movlw 0x0a
    movwf cnt
zloop2
    call periode
    decfsz cnt, F
    goto zloop2
    return

leadout
    movlw 0x0b
    movwf cnt
loloop1
    call leds_on
    call periode
    call leds_off
    call periode
    decfsz cnt, F
    goto loloop1
    movlw 0xea
    call delay2
    return
; cua: 22T ON, 1730T OFF -----

sendbyte
    movwf byte
    movlw 0x08
    movwf bit
    bcf STATUS, C
; enviam un byte -----
; guardam el byte que hem d'enviar
; recorrerem els 8 bits del byte
; netejam el carry

sbloop
    rlf byte, f
    btfss STATUS, C
    goto sbzero
; rotam 1 posició passant pel carry
; carry? (i.e. el bit era un 1?)

sbone
    call one
    goto sbcontinue
; si -> enviam un u

sbzero
    call zero
; no -> enviam un zero

sbcontinue
    decfsz bit, F
    goto sbloop
    return

cmd_pwr
    movlw 0x02
    call sendbyte
    movlw 0xfd
    call sendbyte
    movlw 0x48
    call sendbyte
    movlw 0xb7
    call sendbyte
    return
; comanda PWR -----

cmd_volup
    movlw 0x02
    call sendbyte
; comanda Vol+ -----

```

```
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0x58          ; enviam byte 3 de 4
    call sendbyte
    movlw 0xa7          ; enviam byte 4 de 4
    call sendbyte
    return
```

```
cmd_voldown          ; comanda Vol- -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0x78          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x87          ; enviam byte 4 de 4
    call sendbyte
    return
```

```
cmd_chup             ; comanda Ch+ -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0xd8          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x27          ; enviam byte 4 de 4
    call sendbyte
    return
```

```
cmd_chdown           ; comanda Ch- -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0xf8          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x07          ; enviam byte 4 de 4
    call sendbyte
    return
```

```
cmd_chrtn            ; comanda ChRtn -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0xe8          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x17          ; enviam byte 4 de 4
    call sendbyte
    return
```

```
cmd_0                ; comanda 0 -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
```

```
movlw 0xfd          ; enviam byte 2 de 4
call sendbyte
movlw 0x00          ; enviam byte 3 de 4
call sendbyte
movlw 0xff          ; enviam byte 4 de 4
call sendbyte
return
```

```
cmd_1              ; comanda 1 -----
movlw 0x02          ; enviam byte 1 de 4
call sendbyte
movlw 0xfd          ; enviam byte 2 de 4
call sendbyte
movlw 0x80          ; enviam byte 3 de 4
call sendbyte
movlw 0x7f          ; enviam byte 4 de 4
call sendbyte
return
```

```
cmd_2              ; comanda 2 -----
movlw 0x02          ; enviam byte 1 de 4
call sendbyte
movlw 0xfd          ; enviam byte 2 de 4
call sendbyte
movlw 0x40          ; enviam byte 3 de 4
call sendbyte
movlw 0xbf          ; enviam byte 4 de 4
call sendbyte
return
```

```
cmd_3              ; comanda 3 -----
movlw 0x02          ; enviam byte 1 de 4
call sendbyte
movlw 0xfd          ; enviam byte 2 de 4
call sendbyte
movlw 0xc0          ; enviam byte 3 de 4
call sendbyte
movlw 0x3f          ; enviam byte 4 de 4
call sendbyte
return
```

```
cmd_4              ; comanda 4 -----
movlw 0x02          ; enviam byte 1 de 4
call sendbyte
movlw 0xfd          ; enviam byte 2 de 4
call sendbyte
movlw 0x20          ; enviam byte 3 de 4
call sendbyte
movlw 0xdf          ; enviam byte 4 de 4
call sendbyte
return
```

```
cmd_5              ; comanda 5 -----
movlw 0x02          ; enviam byte 1 de 4
call sendbyte
```

```
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0xa0          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x5f          ; enviam byte 4 de 4
    call sendbyte
    return

cmd_6                      ; comanda 6 -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0x60          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x9f          ; enviam byte 4 de 4
    call sendbyte
    return

cmd_7                      ; comanda 7 -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0xe0          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x1f          ; enviam byte 4 de 4
    call sendbyte
    return

cmd_8                      ; comanda 8 -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0x10          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x7f          ; enviam byte 4 de 4
    call sendbyte
    return

cmd_9                      ; comanda 9 -----
    movlw 0x02          ; enviam byte 1 de 4
    call sendbyte
    movlw 0xfd          ; enviam byte 2 de 4
    call sendbyte
    movlw 0x90          ; enviam byte 3 de 4
    call sendbyte
    movlw 0x6f          ; enviam byte 4 de 4
    call sendbyte
    return

genera_codi                ; calculam el codi que han de mostrar els LEDs ----
    comf tecla, W        ; invertim tecla (1+4 LEDs actius per nivell baix)
    movwf codi           ; la inversió va dins "codi", conservam "tecla"
```

```

bcf codi, 4           ; sempre posam a zero el bit del LED IR
return

leds_on                ; encesa dels 1+4 LEDs -----
movfw codi             ; això "imprimeix" el codi als 1+4 LEDs
movwf PORTA          ; escrivint el codi al port A
return

leds_off               ; apagada dels 1+4 LEDs -----
movlw 0xff           ; això apaga els 1+4 LEDs
movwf PORTA          ; escrivint uns al port A
return

periode                ; retard d'un periode T = 26us -----
movlw 0x21           ; 0x21 * 3 + 6 cicles ~= 26us
call delay1
return

delay1                 ; retard de W * 3 + 6 cicles (max1 = 192.75us) ----
movwf cnt1
dloop1
decfsz cnt1, F
goto dloop1
return

delay2                 ; retard de W * max1 segons (max2 ~= 49.15ms) -----
movwf cnt2
dloop2
movlw 0xff
call delay1
decfsz cnt2, F
goto dloop2
return

delay3                 ; retard de W * max2 segons (max3 ~= 12.533s) -----
movwf cnt3
dloop3
movlw 0xff
call delay2
decfsz cnt3, F
goto dloop3
return

end ; final -----

; EOF

```