

Software para el modulo 5i-5or

Existen muchas aplicaciones del tipo secuencial que solamente requieren de dos botones: uno para arrancar la secuencia y otro para pararla. Las salidas de este tipo de controles se conectan a solenoides, motores u otros dispositivos que se van activando en función del tiempo.

Como el modulo 5i-5or-1, Clave: 704 cuenta con 5 entradas y 5 salidas "Listas para ser utilizadas", es un hardware adecuado para este tipo de aplicaciones. Ud. no tiene que preocuparse por el mismo, PicMicro Estudio ya lo tiene listo para ser utilizado en su aplicación. En este artículo aprenderá como hacerlo.

En la Fig. 1 podemos observar como conectar los botones. El botón de arranque Start se conecta a la terminal E5 que corresponde al Pin RA4, BIT 4 del Puerto A. El botón Stop, que se usará para parar la secuencia en cualquier momento, esta conectado al Pin Restablecer RST del microcontrolador (Reset)

En la Fig. 2 esta el diagrama esquemático completo de esta tarjeta. Observe como la terminal E5 tiene conectada una resistencia de 1K a positivo, por lo tanto el Pin RA4 se mantiene también en positivo. Cuando se oprime el botón de Start entonces el Pin RA4 pasa a cero Volts arrancando la secuencia. En este diagrama también podemos ver que el Pin de RESET se mantiene en positivo hasta que se oprime el botón de Stop que al aplicar una tierra al micro controlador suspende la ejecución del programa y detiene la secuencia.

El software fuente, en Ensamblador, para implementar esta aplicación se ilustra en las siguientes páginas. Veamos algunas partes importantes del mismo.

```
Checa_in    btfsc    porta,in    ;checa que in a 0
            goto    Checa_in
```

Con estas instrucciones se inicia la secuencia.

Porta, in

Se refiere al botón de Start. Cuando el Pin RA4 pasa a cero se inicia la secuencia.

Las secuencias se resuelven implementando tres Macros.

Timer

Output

Repite

El uso de estas Macros es de lo más fácil. Veamos como hacerlo.

Timer

Timer m,s,dm

Esta macro es una “pausa” que se puede programar: m son los minutos, s son los segundos y dm son décimas de segundo.

Timer 0,1,0

Quiere decir una pausa de 0 minutos, 1 segundo, 0 décimas de Segundo.

Output

Output S1, S2, S3, S4, S5

Esta Macro se utiliza para activar los relevadores de salida: S1 representa el relevador S1, si este valor se pone en 0, el relevador S1 se desactiva y si S1 se pone en 1 el relevador S1 SE ACTIVA. De igual manera los demás relevadores.

Output 0,0,0,0,1

Quiere decir que el relevador S5 se activa mientras que los relevadores restantes permanecen desactivados.

Veamos una pequeña secuencia

Timer	0,1,0
Output	0,0,0,0,0
Timer	0,1,0
Output	0,0,0,0,1
Timer	0,1,0
Output	0,0,0,1,1
Timer	0,1,0
Output	0,0,1,1,1
Timer	0,1,0
Output	0,1,1,1,1
Timer	0,1,0
Output	1,1,1,1,1

Se inicia con todos los relevadores apagados, después, cada segundo se enciende un relevado, empezando con S5, hasta que todos están energizados. ¡Ud. puede crear la secuencia que quiera!

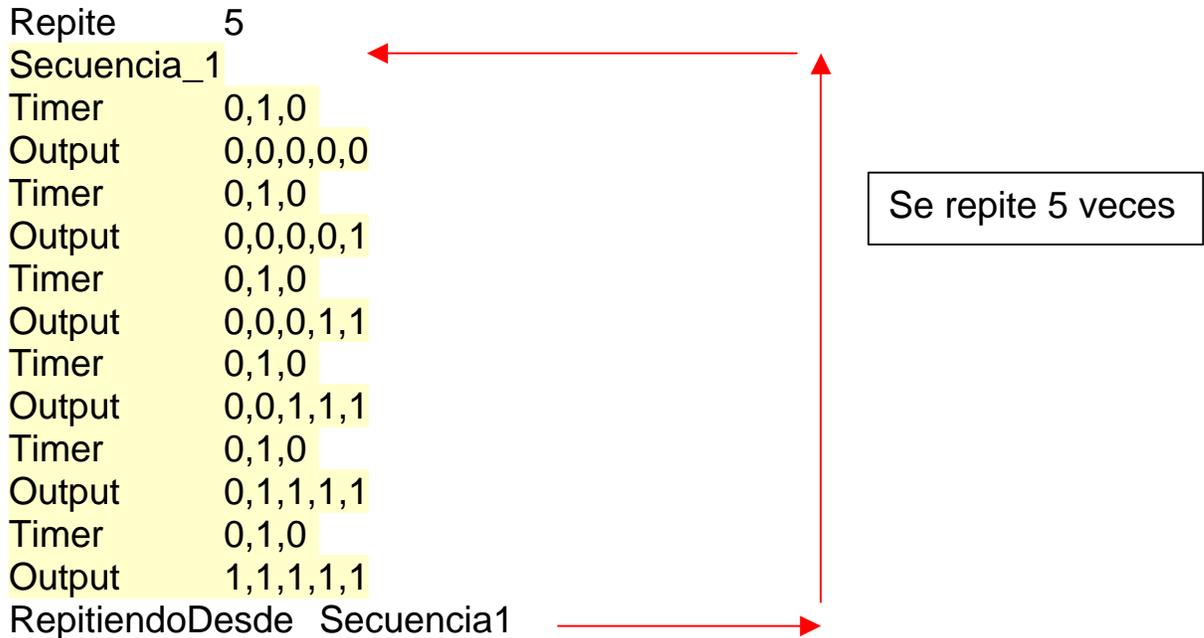
Ahora solamente nos falta una manera de repetir esta secuencia un numero determinado de veces. Veamos las siguientes Macros.

Repite y RepitiendoDesde

Repite Numero_de_veces
Quiere decir que se quiere repetir una secuencia un
Numero_de_veces

RepitiendoDesde Etiqueta
Quiere decir que se repite la secuencia desde la Etiqueta.

Veamos un ejemplo para aclarar las cosas:



Para terminar el programa analicemos las siguientes líneas

```
Ciclo
    ciclo=1
    if ciclo==0
Fin    goto    Inicio_de_nuevo
      else
      goto    Prog2
    endif
```

Se puede seleccionar ciclo en 1 o en 0

Cuando “ciclo” es 1 el programa se dirige a ejecutar las instrucciones que se encuentren a partir de la etiqueta Prog2 que se encuentra al principio del Listado. Esto quiere decir que todo el programa se repite indefinidamente. Para detenerlo se puede oprimir el botón de Stop, como vimos anteriormente. Cuando “ciclo” lo ponemos en “0” entonces el programa va a verificar el estado del botón Start, esperando que se oprima de nuevo. Esto quiere decir que la secuencia de todo el programa se ejecuta una sola vez.

Una vez que el programa ha sido terminado se debe ensamblar con MPASAM y el PIC se puede grabar con el Programador de Pics Clave 501.

En el CD Clave: 500 se pueden encontrar mas información y muchas otras aplicaciones de los módulos PicMicro Estudio.

Se puede concluir que para aplicaciones secuenciales de hasta 5 entradas con 5 salidas la tarjeta prefabricada Clave: 704 puede ser de mucha utilidad ya que esta "Lista para usarse", Ud. no tiene que preocuparse por la fabricación de una tarjeta específica para su aplicación, que implicaría el desarrollo de una tarjeta de circuito impreso, PicMicro Estudio ya la tiene para que Ud la programe de inmediato.

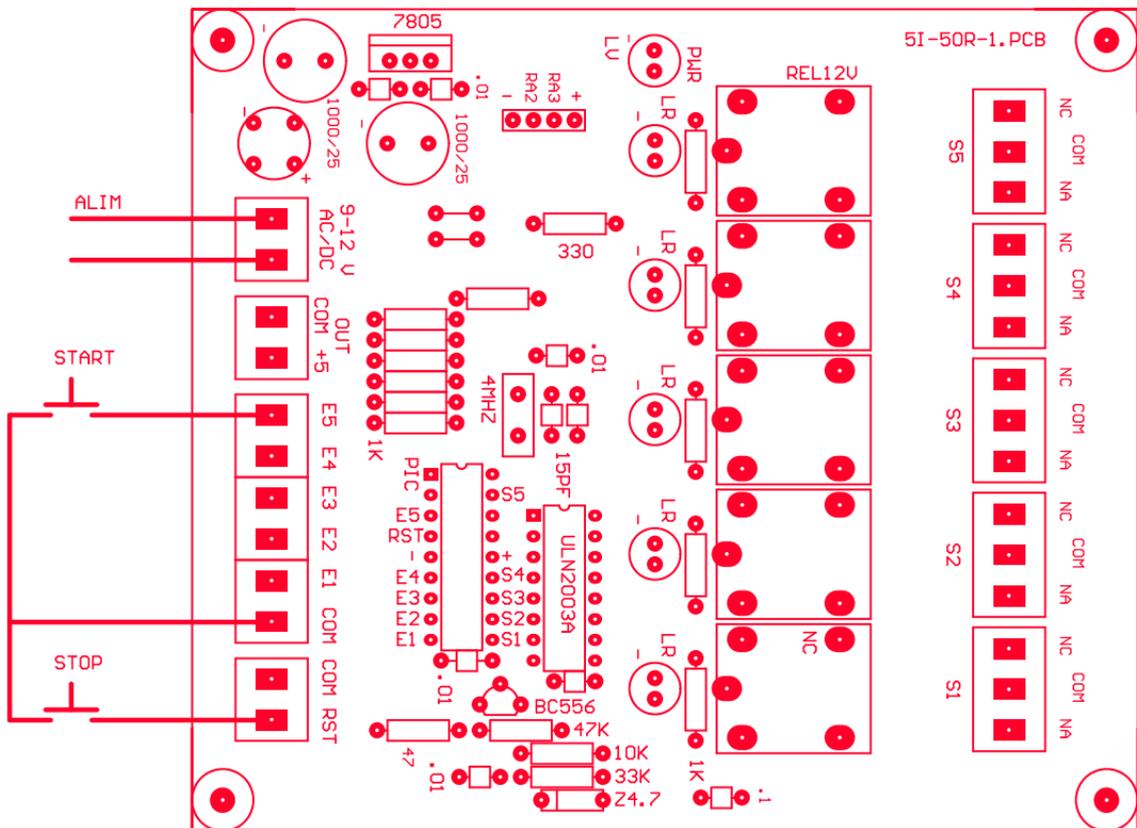
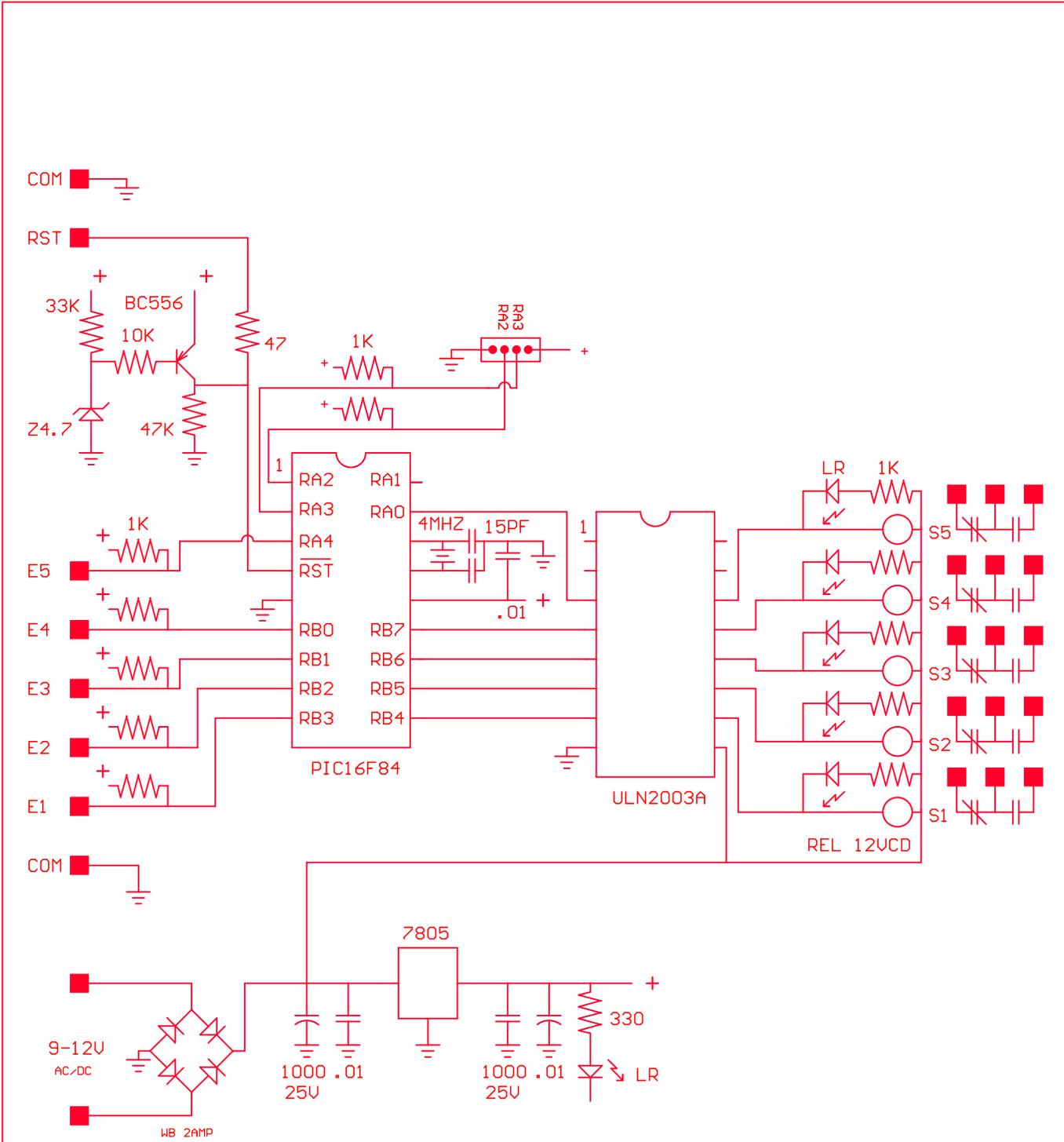


Fig. 1



PIC micro Estudio

MODULO DE
5 ENTRADAS
5 SALIDAS
CON RELE

5i-5or-1

Fig. 2

=====prfacil.asm=====27-Enero del 2003=====

;PARA SER USADO EN LA TARJETA 5i-5or

;
; list p=16c84
; radix hex

; destination designator equates

w equ 0
f equ 1
in equ 4
out1 equ 4
out2 equ 5
out3 equ 6
out4 equ 7
out5 equ 0

led_R equ 1
led_V equ 0

; cpu equates (memory map)

porta equ 0x05
portb equ 0x06

ncount equ 0x0c ;registro interno de paus_100ms
mcount equ 0x0d ;registro externo de paus_100ms
pcount equ 0x0e ;registro de npause_100ms

rcount equ 0x0f ;registro mas interno de paus_1s
scount equ 0x10 ;registro medio de paus_1s
tcount equ 0x11 ;registro externo de paus_1s
ucount equ 0x12 ;registro de npaus_1s

count1 equ 0x13 ;registro mas interno de paus_1m
count2 equ 0x14 ;registro medio de paus_1m
count3 equ 0x15 ;registro externo de paus_1m
count4 equ 0x16 ;registro mas externo de paus_1m
count5 equ 0x17 ;registro de npaus_1m

veces equ 0x18 ;registro que repite n veces un programa

; equus de configuracion
;-----

_CP_ON EQU H'000F'
_CP_OFF EQU H'3FFF'
_PWRTE_ON EQU H'3FF7'
_PWRTE_OFF EQU H'3FFF'
_WDT_ON EQU H'3FFF'
_WDT_OFF EQU H'3FFB'
_LP_OSC EQU H'3FFC'
_XT_OSC EQU H'3FFD'
_HS_OSC EQU H'3FFE'
_RC_OSC EQU H'3FFF'

list p=16F84A

__CONFIG _XT_OSC & _PWRTE_ON & _CP_OFF & _WDT_OFF

```

;-----
;MACROS

Minutos      macro                min
              movlw                min
              movwf                count5
              call                 npaus_1m
              endm

Segundos     macro                seg
              movlw                seg
              movwf                ucount
              call                 npaus_1s
              endm

Miliseg      macro                milliseg
              movlw                milliseg
              movwf                pcount
              call                 npaus_100ms
              endm

Timer        macro                min,seg,milliseg
              if min>0
Minutos      min
              endif

              if seg>0
Segundos     seg
              endif

              if milliseg>0
Miliseg      milliseg
              endif
              endm

Output       macro                o1,o2,o3,o4,o5
              if o1==0
              bcf                  portb,out1
              else
              bsf                  portb,out1
              endif
              if o2==0
              bcf                  portb,out2
              else
              bsf                  portb,out2
              endif
              if o3==0
              bcf                  portb,out3
              else
              bsf                  portb,out3
              endif
              if o4==0
              bcf                  portb,out4
              else
              bsf                  portb,out4
              endif
              if o5==0
              bcf                  porta,out5

```

```
else
bsf          porta,out5
endif
endm
```

```
Repite      macro          Repeticiones          ;Carga numero de repeti
movlw      Repeticiones          ;Repeticiones 1 a 255, n
movwf
endm
```

```
RepitiendoDesde macro          Etiqueta
decfsz          veces,f
goto           Etiqueta
endm
```

```
-----
;Principio
```

```
org          0x000
```

```
start      movlw          b'11110000'
tris       porta
movlw      b'00001111'
tris       portb
```

```
Inicio_de_nuevo
```

```
clrf       portb          ;apaga puerto B
clrf       porta          ;apaga puerto A
```

```
Checa_in   btfsc          porta,in          ;checa que in a 0
goto       Checa_in
```

```
-----
;Programa
;-----
```

```
Prog1
Prog1Step1 Timer          0,1,0          ;min,seg,miliseg
```

```
FinProg1
```

```
Prog2
```

```
Repite     d'3'          ;Repite 1 a 255, no se v
```

```
Prog2Step1
```

```
Output    0,0,0,0,1
Timer     0,0,5
```

```
Prog2Step2
```

```
Output    0,0,0,1,0
Timer     0,0,5
```

```
Prog2Step3
```

```
Output    0,0,1,0,0
Timer     0,0,5
```

```
Prog2Step4
```

```
Output 0,1,0,0,0
Timer 0,0,5
```

Prog2Step5

```
Output 1,0,0,0,0
Timer 0,0,5
```

```
RepitiendoDesde Prog2Step1 ;Repite este programa de
```

FinProg2

;-----

Prog3

```
Repite d'3' ;Repite 1 a 255, no se v
```

Prog3Step1

```
Output 1,1,1,1,1
Timer 0,0,7
```

Prog3Step2

```
Output 0,0,0,0,0
Timer 0,0,7
```

```
RepitiendoDesde Prog3Step1 ;Repite este programa de
```

FinProg3

;-----

Ciclo

```
ciclo=1
if ciclo==0
Fin goto Inicio_de_nuevo
else
goto Prog2
endif
```

;-----
;SUBRRITINAS
;-----

;paus_100ms es una pausa de 100 mili segundos = a .1 seg

```
paus_100ms movlw 0x82 ;M
movwf mcount ;to M counter
loadn movlw 0xff ;N
movwf ncount ;to N counter
decn decfsz ncount,f ;decrement N
goto decn ;again
decfsz mcount,f ;decrement M
goto loadn ;again
return ;done
```

;npause_100ms repite 100ms las veces que contenga el registro pcount

;antes de entrar cargar el registro pcount con el numero decaedo

```
npaus_100ms call paus_100ms
decfsz pcount,f
goto npaus_100ms
return
```

;paus_1s es una pausa de 1 segundo

```
paus_1s movlw 0x0a ;carga
movwf tcount ;tcount
```

```

loads      movlw    0x82          ;carga
           movwf   scount      ;scount
loadr      movlw    0xff        ;carga
           movwf   rcount      ;rcount
decr       decfsz  rcount,f     ;decrement r
           goto   decr         ;again
           decfsz  scount,f     ;decrement s
           goto   loadr        ;again
           decfsz  tcount      ;decrement t
           goto   loads        ;again
           return              ;done

```

```

;-----
;npause_1s repite 1s las veces que contenga el registro ucount
;antes de entrar cargar el registro pcount con el numero decaido
npaus_1s   call    paus_1s
           decfsz  ucount,f
           goto   npaus_1s
           return

```

```

;-----
;paus_1m es una pausa de 1 minuto

```

```

paus_1m    movlw    0x3c        ;carga 60 decimal
           movwf   count4      ;count4
load3      movlw    0x0a        ;carga
           movwf   count3      ;count3

load2      movlw    0x82        ;carga 82 se ADELANTA .060
           movwf   count2      ;count2
load1      movlw    0x00        ;carga
           movwf   count1      ;count1
decl       decfsz  count1,f     ;decrement 1
           goto   decl         ;again
           decfsz  count2,f     ;decrement 2
           goto   load1        ;again
           decfsz  count3      ;decrement 3
           goto   load2        ;again
           decfsz  count4      ;decrement 3
           goto   load3        ;again
           return              ;done

```

```

;-----
;npause_1m repite 1m las veces que contenga el registro count5
;antes de entrar cargar el registro count5 con el numero decaido
npaus_1m   call    paus_1m
           decfsz  count5,f
           goto   npaus_1m
           return

end

```