

Control Para Máquinas

Traga monedas.

Existe una gran variedad de máquinas traga monedas para la diversión de los niños: Video juegos, simuladores o simplemente “carritos” o “animalitos” en las que los niños disfrutan de un “paseo” por unos cuantos minutos. El funcionamiento es sencillo, se deposita una o varias monedas en una ranura, enseguida la máquina inicia su funcionamiento. Después de un tiempo la diversión se termina. Fig. 1

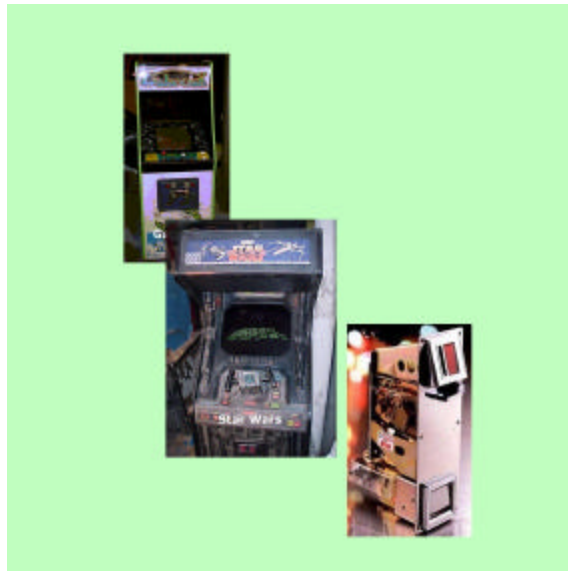


Fig. 1

Un interruptor de limite que se ubica dentro del depósito de monedas se activa cada vez que una moneda se deposita. Fig. 2ª y Fig. 2b

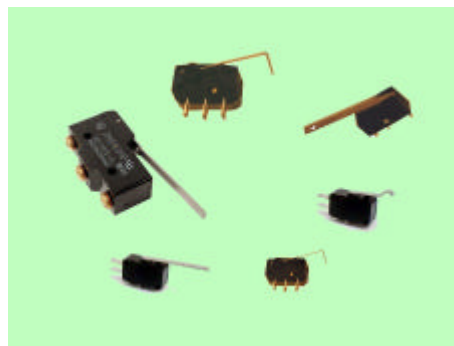


Fig.2a
Interruptores de limite

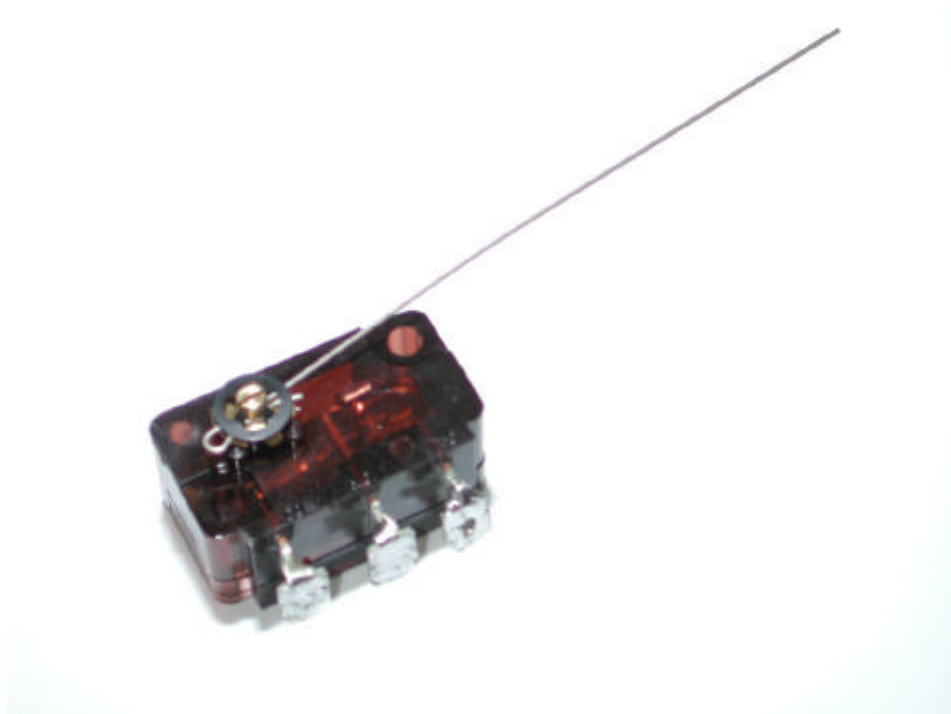


Fig. 2b
Interruptor de monedero

En la tarjeta de control se podrá seleccionar el numero de monedas que el usuario debe depositar y también la duración del servicio. Por ejemplo una, dos, o tres monedas. Uno, dos o tres minutos. En la Fig. 3 se muestra un diagrama de bloques de estos requerimientos. El interruptor de limite es una entrada, los interruptores selectores son también entradas que deberán estar a bordo de la tarjeta de control y por ultimo con un contacto de salida se podrá energizar un motor o algún otro dispositivo como un video juego, etc.

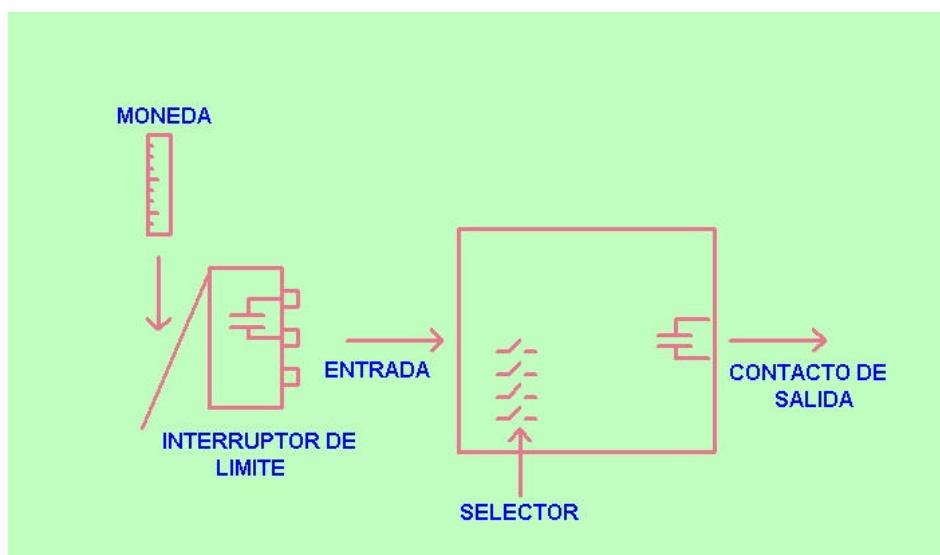


Fig. 3

El módulo de 5 entradas y 3 salidas con relevador (Clave 703) es una buena opción para resolver este ejercicio. Fig. 4

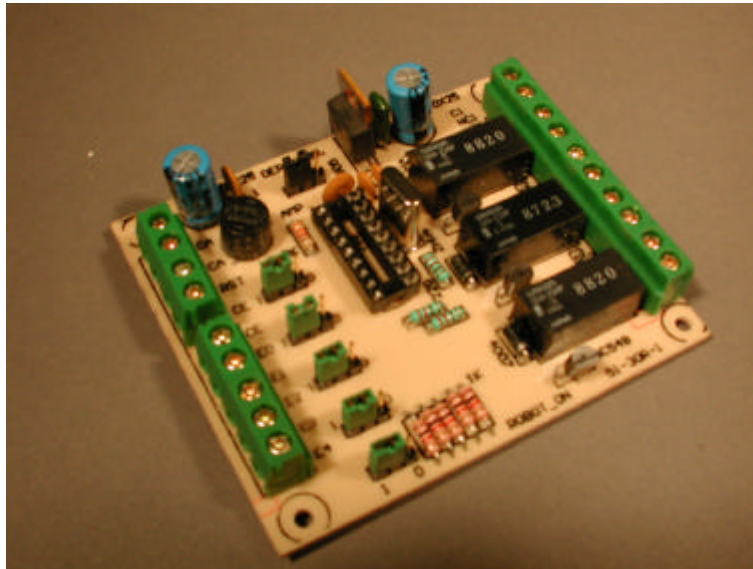


Fig. 4

Como se puede observar en el diagrama esquemático de la Fig. 5 esta tarjeta es muy flexible y puede ser utilizada para esta aplicación y para muchas otras del mismo tipo.

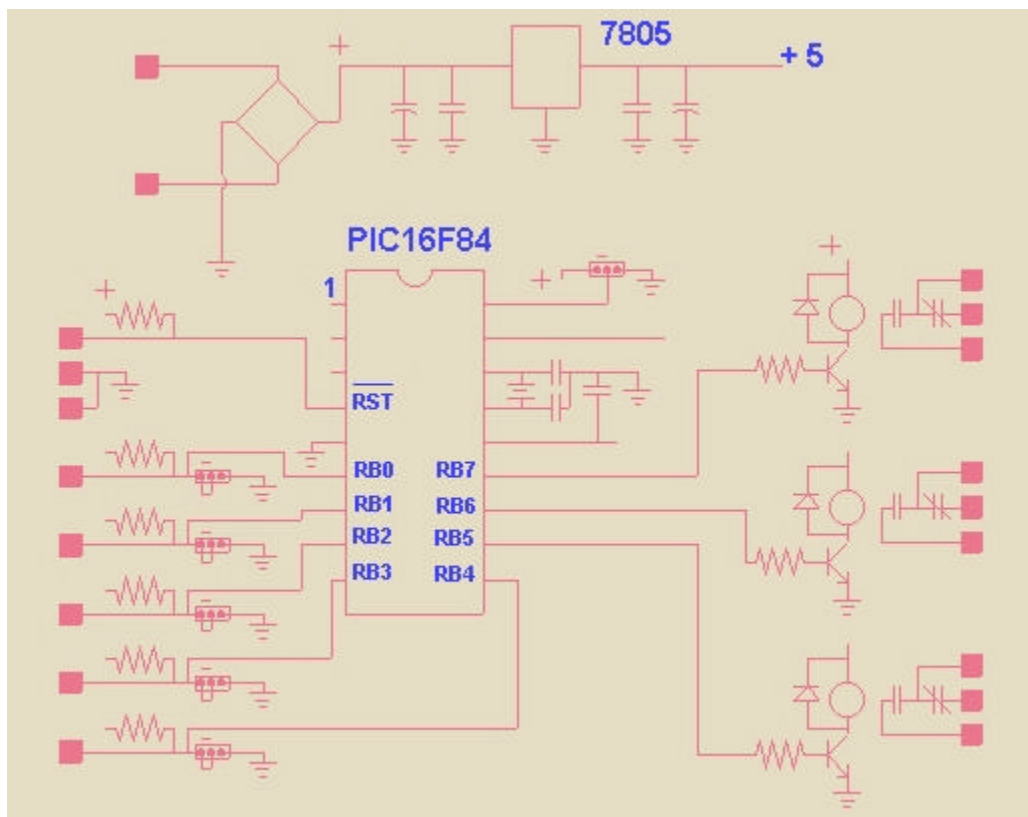


Fig. 5

Los pines RB0 a RB4 están configurados como entradas. Con el puente conectado como se indica en la figura la resistencia mantiene un “uno” en el pin del microcontrolador. Si el borne atornillable se conecta a tierra se envía un “cero”.

Los pines RB5 a RB7 están configurados como salidas. Mediante unos transistores se energizan los relevadores. En las terminales atornillables se proporcionan los contactos UPDT de los relees.

La fuente de alimentación permite que el usuario pueda energizar la tarjeta con AC/DC. Se puede conectar por ejemplo un eliminador o simplemente un transformador de 9 volts.

En este caso el interruptor de limite, que es el elemento que va a contar las monedas se conecta a RB0, Los puentes RB1, RB2, RB3 y RB4 se van a utilizar como selectores del funcionamiento de la tarjeta. Los puentes en la posición hacia la izquierda mandan un “uno” a los pines del micro y en la posición de la derecha mandan un “cero”. Finalmente RB7 se dedica a la salida. Mediante los contactos del relevador de salida se podrá controlar un motor un Video Juego, etc.

Para contar las monedas los selectores RB1, RB2 actuarán de la siguiente manera:

RB2	RB1	MONEDAS
1	1	UNA
1	0	DOS
0	1	TRES
0	0	CUATRO

Para ajustar el tiempo los selectores RB3, RB4 tendrán las posiciones siguientes:

RB4	RB3	MINUTOS
1	1	UNO
1	0	DOS
0	1	TRES
0	0	CUATRO

Recuerde que el puente (Interruptor) hacia el lado izquierdo es un “uno” y hacia el lado derecho es un “cero” Fig. 6

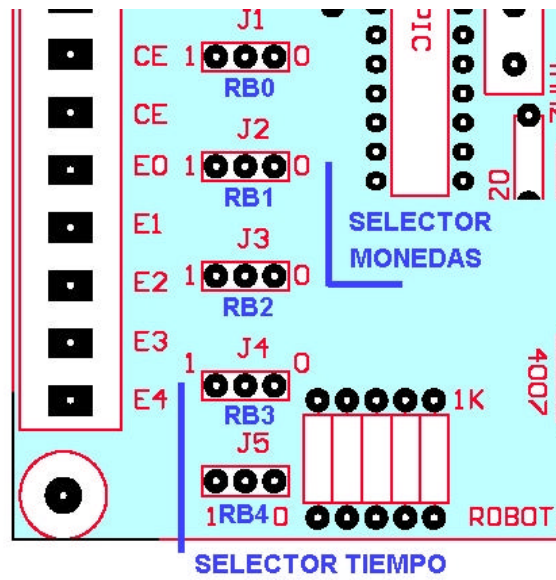


Fig. 6

En la Fig. 7 se muestra el diagrama de alambrado.

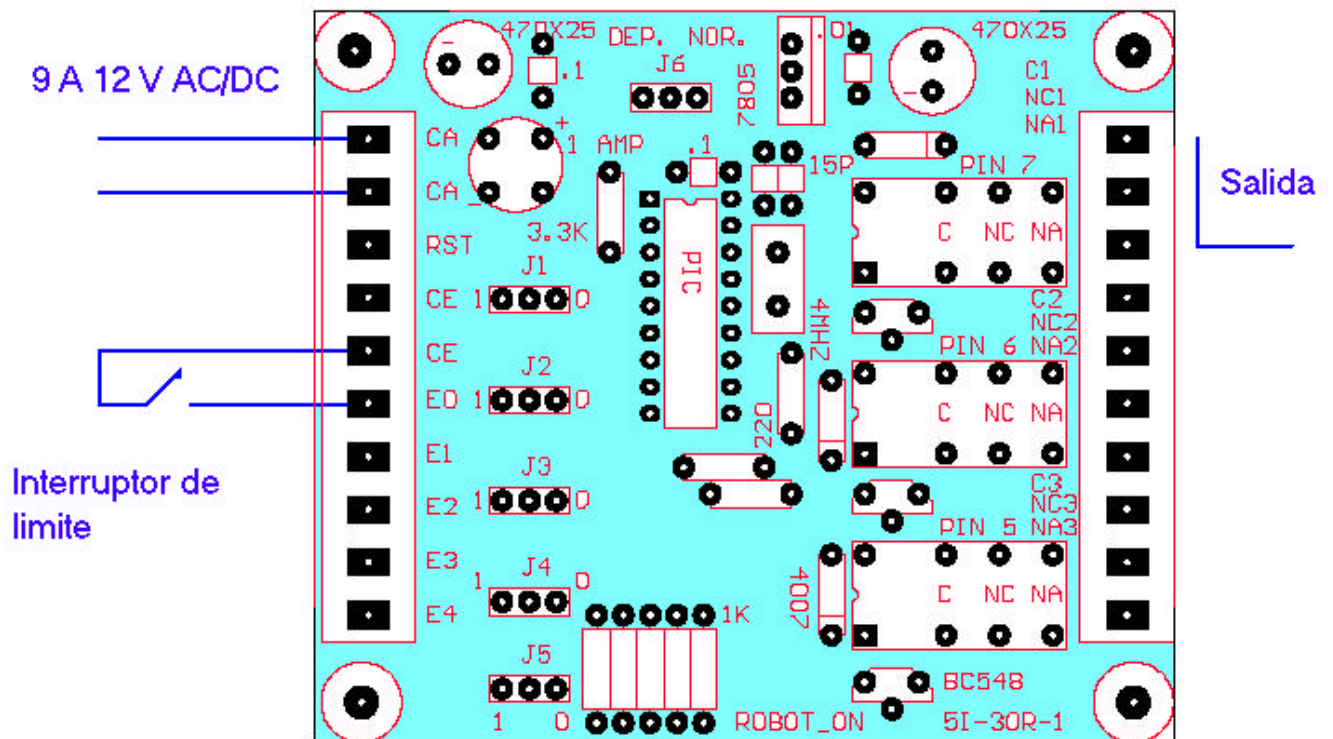


Fig. 7

A continuación se transcribe el programa para esta aplicación.

```
;=====monedas.asm=====13 de Julio del 2001=====
;PARA SER USADO EN LA TARJETA Modulo de 5 entradas y 3 salidas con relevador
;5i-30r Clave 703
;-----
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

;-----
;      equs de este programa
;-----

NumeroMonedas  equ    0x18    ;registro que almacena el numero de monedas
NumeroMinutos  equ    0x19    ;registro para almacenar los minutos

IntMonedero    equ    0x0
RB1            equ    0x1
RB2            equ    0x2
RB3            equ    0x3
RB4            equ    0x4

;-----
;      equs 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

#define ActivaRele    bsf    portb,7
#define DesactivaRele bcf    portb,7

;-----
;MACROS
;-----
Miliseg          macro    miliseg          ; d'1' < miliseg < d'255'
                    movlw    miliseg
                    movwf    pcount
                    call    npaus_100ms
                    endm
;-----

org              0x000

                    movlw    b'00011111'
                    tris     portb          ;Define como salida al PuertoB

                    clrf     portb          ;Apaga el puerto B

                    goto     Programa

;-----
;SUBROUTINAS
;-----
;paus_100ms es una pausa de 100 mili segundos = a .1 seg
paus_100ms        movlw    0x82
                    movwf    mcount
loadn              movlw    0xff
                    movwf    ncount
decn              decfsz    ncount,f
                    goto     decn
                    decfsz    mcount,f
                    goto     loadn
                    return

;-----
;npause_100ms repite 100ms las veces que contenga el registro pcount
;antes de entrar cargar el registro pcount con el numero deseado
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    ;decrementa r
            goto    decr      ;again
            decfsz scount,f    ;decrementa s
            goto    tcount     ;decrementa t
            goto    loads
            return

;-----
;npause_1s repite 1s las veces que contenga el registro ucount
;antes de entrar cargar el registro pcount con el numero deseado
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
dec1         decfsz   count1,f   ;decrement 1
            goto    dec1        ;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 deseado
npaus_1m     call    paus_1m
            decfsz   count5,f
            goto    npaus_1m
            return

;-----
;NumeroDeMonedas revisa el estado de los pines RB1, RB2 que actuan como
;selectores para definir el numero de monedas que se van a contar.
;
;      RB2      RB1      Numero de monedas
;      1        1        Una
;      1        0        Dos
;      0        1        Tres
;      0        0        Cuatro
NumeroDeMonedas

            movlw    d'1'
            movwf    NumeroMonedas        ;carga Numero de Monedas

            btfss    portb,RB1

```



```

        goto        Dos_Cuatro        ;RB1=0

        btfss       portb,RB2
        goto        Tres              ;RB2=0, RB1=1
        goto        FinNumeroDeMonedas ;RB2=1, RB1=1

```

```

Dos_Cuatro    btfss       portb,RB2        ;RB1=0
              goto        Cuatro          ;RB2=0, RB1=0

```

```

Dos           movlw     d'2'              ;RB2=1, RB1=0
              movwf     NumeroMonedas
              goto      FinNumeroDeMonedas

```

```

Tres          movlw     d'3'
              movwf     NumeroMonedas
              goto      FinNumeroDeMonedas

```

```

Cuatro        movlw     d'4'
              movwf     NumeroMonedas

```

```

FinNumeroDeMonedas
Return

```

```

;-----
;NumeroDeMinutos revisa el estado de los pines RB3, RB4 que actuan como
;selectores para definir el numero de minutos que va ha durar el evento
;
;      RB4      RB3      Numero de minutos
;      1        1        Una
;      1        0        Dos
;      0        1        Tres
;      0        0        Cuatro
NumeroDeMinutos

```

```

        movlw     d'1'
        movwf     NumeroMinutos        ;carga Numero de minutos

```

```

        btfss     portb,RB3
        goto      Dos_CuatroM          ;RB3=0

```

```

        btfss     portb,RB4
        goto      TresM                ;RB4=0, RB3=1
        goto      FinNumeroDeMinutos   ;RB3=1, RB3=1

```

```

Dos_CuatroM   btfss     portb,RB4        ;RB3=0
              goto      CuatroM         ;RB4=0, RB3=0

```

```

DosM          movlw     d'2'              ;RB4=1, RB3=0
              movwf     NumeroMinutos
              goto      FinNumeroDeMinutos

```

```

TresM         movlw     d'3'
              movwf     NumeroMinutos
              goto      FinNumeroDeMinutos

```

```

CuatroM       movlw     d'4'

```

```

movwf    NumeroMinutos

FinNumeroDeMinutos
Return
;-----
;-----
Programa

    call NumeroDeMonedas    ;Define numero de monedas

ChecaIntMonedero
    btfsc    portb,IntMonedero
    goto    ChecaIntMonedero    ;IntMonedero no se ha activado
    Miliseg    d'1'    ;pausa de 100 miliseg
    btfsc    portb,IntMonedero
    goto    ChecaIntMonedero    ;Despues de la pausa se desactivo
    ;Activado mas de 100 miliseg

    Miliseg    d'1'

ChecaDesactive
    btfss    portb,IntMonedero
    goto    ChecaDesactive    ;InMonedero no se ha desactivado
    Miliseg    d'1'    ;pausa de 100 miliseg
    btfss    portb,IntMonedero
    goto    ChecaDesactive    ;Despues de la pausa se activo
    ;Desactivado mas de 100 miliseg

    decfsz    NumeroMonedas,f
    goto    ChecaIntMonedero    ;espera mas monedas
    ;Termino de contar monedas

    call NumeroDeMinutos    ;Define numero de minutos

ActivaRele

    movf    NumeroMinutos,w    ;Minutos al acumulador
    movwf    count5    ;Minutos al contador que define los
    ;minutos
    call    npaus_1m    ;pausas de un minuto

DesactivaRele

goto Programa

end

```

Las instrucciones nuevas en este articulo son:

list p=<type>

Define el tipo de microcontrolador que se va a programar.
Por ejemplo

```
list    p=16F84A
```

```
--_config <expr>
```

Define los bits de configuración al valor dado en **<expr>**

Diríjase a la referencia: PICmicro Microcontroller Data Book para una descripción de los bits de configuración de cada microcontrolador.

En la Fig, 8 se presenta una fotografía del proyecto armado en nuestro laboratorio.

Para editar y compilar este programa es conveniente usar MPLAB y MPASAM respectivamente. Si se desea también se puede bajar el file **monedas.zip** de la dirección de Internet :

www.prodigyweb.net.mx/wgb/articulos

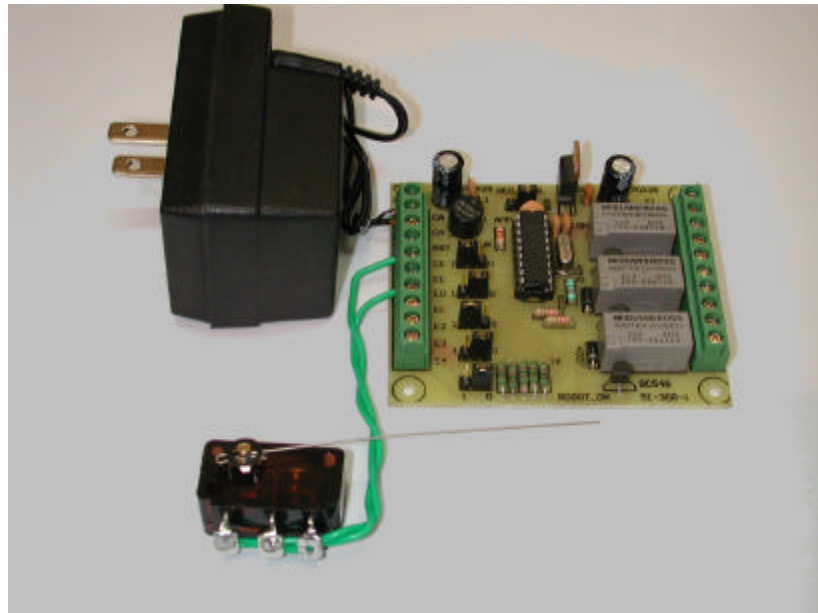


Fig. 8

