

SISTEMAS ELECTRÓNICOS Y AUTOMÁTICOS PRACTICAS DE MICROCONTROLADORES PIC

PRÁCTICA 6: *Gobierno del display 7 segmentos*

-
- ***Display 7 segmentos***
-

1. Objetivos:

Gobierno del display 7 segmentos en la tarjeta EasyPIC4.

2. Display 7 segmentos

Una de las aplicaciones mas populares de los LED's es la de señalización. Quizás la más utilizada sea la de 7 LED's colocadas en forma de ocho tal y como se indica en la figura 1. Aunque externamente su forma difiere considerablemente de un diodo LED típico, internamente están constituidos por una serie de diodos LED con unas determinadas conexiones internas. En la figura 1 se indica el esquema eléctrico de las conexiones del interior de un indicador luminoso de 7 segmentos.

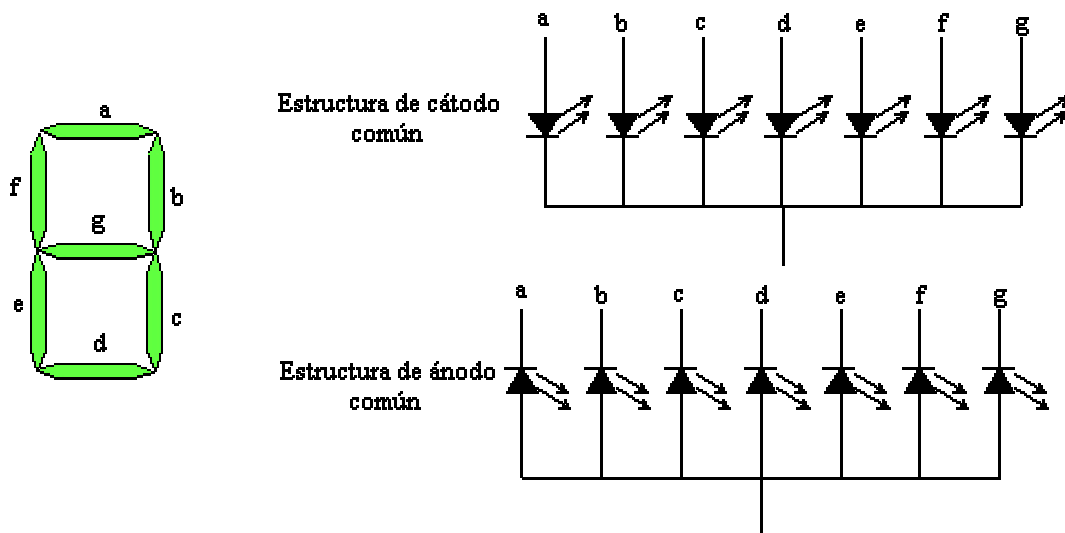


Figura 1. Display 7 segmentos. A la izquierda aparecen las dos posibles formas de construir el circuito

La figura 1 muestra un indicador de siete segmentos. Contiene siete LED rectangulares (a - g), en el que cada uno recibe el nombre de segmento porque forma parte del símbolo que esta mostrando. Con un indicador de siete segmentos se pueden formar los dígitos del 0 al 9, también las letras a, c, e y f y las letras minúsculas b y d. Los entrenadores de microcontroladores (como la EasyPIC4) usan a menudo displays de 7 segmentos para mostrar todos los dígitos del 0 al 9 más A, B, C, D y F .

Por ejemplo, si queremos representar el número de 1 en el display deberemos poner a ON los leds b y c, y los otros leds deben estar apagados. Esto lo podemos escribir así 0110000(0). El primer dígito representa al led a, el segundo al b, el tercero al c, y así sucesivamente. Un cero representa que no polarizamos el led, es decir no le aplicamos tensión. Un uno representa que el led esta polarizado, y por lo tanto, emite luz.

Muchas veces aparece un octavo segmento, entre paréntesis en el ejemplo anterior, que funciona como punto decimal (figura 2).

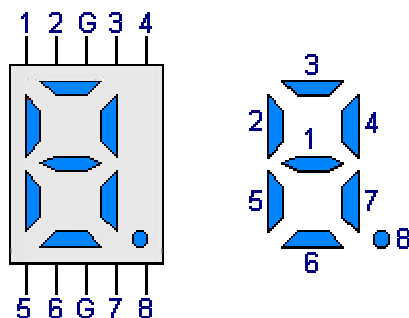


Figura 2. Octavo segmento

2.1. Displays 7 segmentos en la tarjeta EasyPIC

En la práctica anterior vimos que la tarjeta EasyPIC tiene 4 displays 7 segmentos en modo multiplexado.

Las líneas de datos que conectan al microcontrolador con los displays son los bits de **PORTD** y cada display se habilita a través de los cuatro bit menos significativos de **PORTA**, es decir, RA0 = 1, activa el display DIS0, RA1=1 activa DIS1, RA2=1 activa DIS2 y RA3=1 activa DIS3, mientras que el dato a visualizar en cada display se envía a través de las líneas de PORTD.

7-segment displays

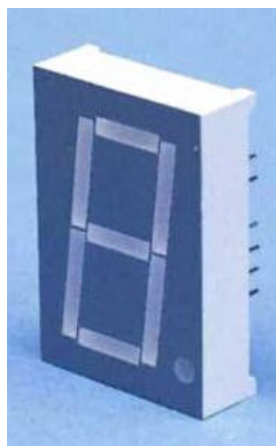
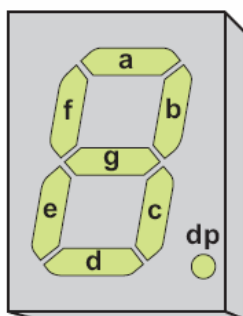


Figura 3. Display 7 segmentos.

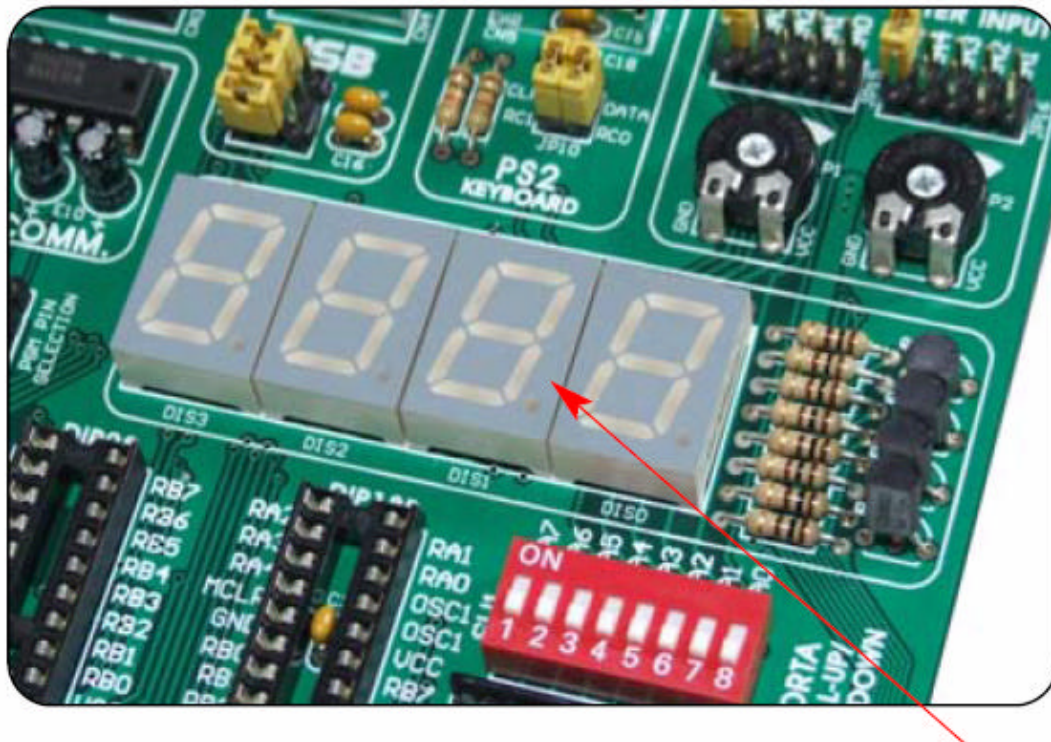


Figura 4. EasyPIC tiene 4 displays 7 segmentos.

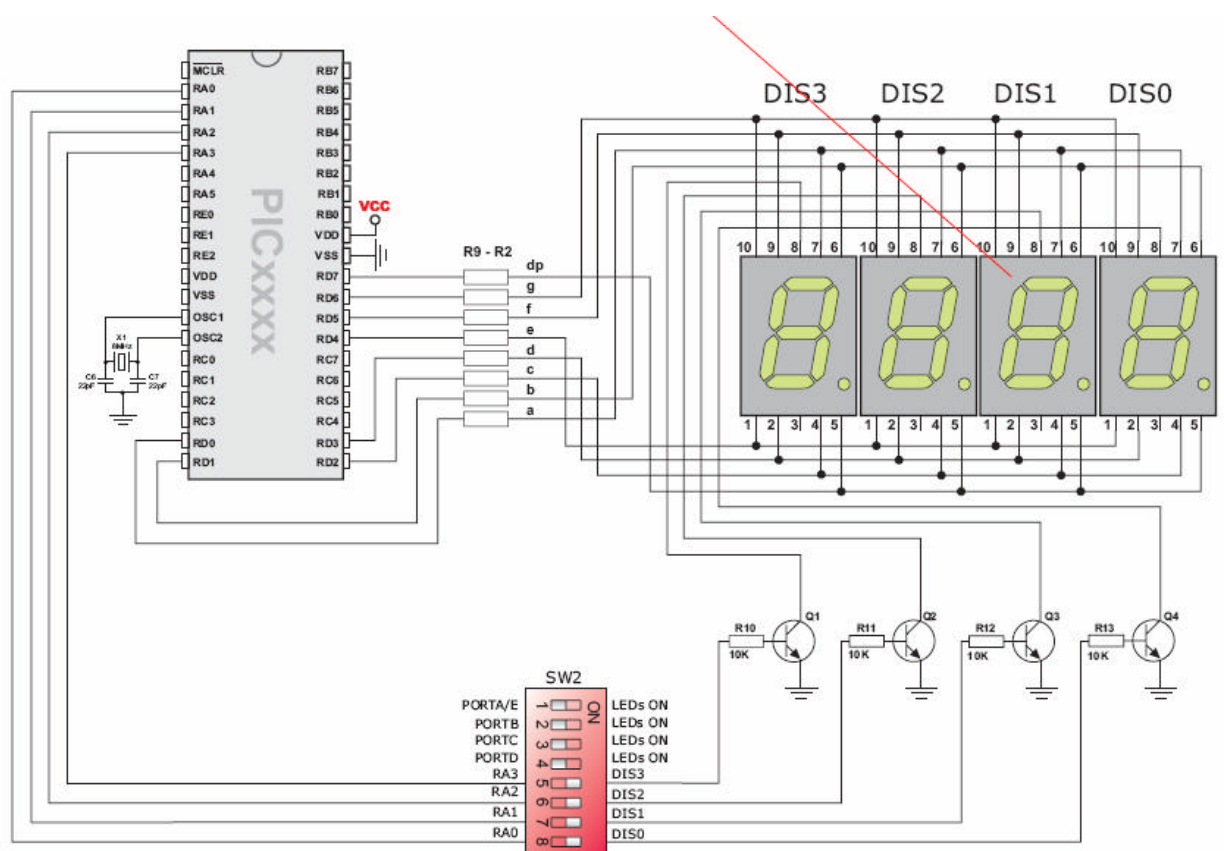


Figura 5. Conexionado de los 4 displays 7 segmentos en la EasyPIC.

3. EJERCICIOS

EJERCICIO 1:

- a) El programa **Display_01.asm** produce que en un display 7 segmentos conectado al Puerto B se visualice la cantidad leída por el Puerto A.

Modifica el código del programa para que éste funcione sobre la tarjeta EasyPIC4 con el PIC16F877A, para ello:

- Utiliza el Puerto C en lugar del Puerto A para las entradas.
- Habilita el display 7 segmentos DIS0, poniendo a "1" el bit RA0.
- Utiliza el Puerto D en lugar del Puerto B para enviar el dato al display 7 segmentos.
- Modifica el modelo del microcontrolador

En la página siguiente viene el código del programa con las líneas de código que debéis modificar resaltadas. Además tendréis que añadir más instrucciones para habilitar DIS0, poniendo a "1" el bit RA0.

- b) Comprueba el funcionamiento del nuevo programa en la tarjeta EasyPIC4 con el PIC16F877A.

```

;***** Display_01.asm *****
;
;
; En un display de 7 segmentos conectado al Puerto B se visualiza la cantidad leída por
; el Puerto A. Así por ejemplo si por la entrada lee "---0101" en el display visualiza
; "5".
;
; ZONA DE DATOS *****

    _CONFIG    _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
    LIST      P=16F84A
    INCLUDE   <P16F84A.INC>

; ZONA DE CÓDIGOS *****

Inicio
    ORG      0                ; El programa comienza en la dirección 0.
    bsf     STATUS,RP0       ; Acceso al Banco 1.
    clrf    PORTB            ; Las líneas del Puerto B se configuran como salida.
    movlw   b'00011111'      ; Las 5 líneas del Puerto A se configuran como entrada.
    movwf   PORTA
    bcf     STATUS,RP0       ; Acceso al Banco 0.

Principal
    movf    PORTA,W          ; Lee la entrada
    andlw   b'00001111'      ; Máscara para quedarse con el valor de las
                                ; entradas correspondientes al nibble bajo.
    call    Binario_a_7Segmentos ; Convierte código binario a 7seg del display.
    movwf   PORTB            ; Resultado se visualiza por el puerto de salida.
    goto   Principal

; Subrutina "Binario_7Segmentos" -----
;
;
Binario_a_7Segmentos    ; Tabla para display de 7 segmentos.
    addwf   PCL,F
Tabla
    retlw   3Fh          ; El código 7 segmentos para el "0".
    retlw   06h          ; El código 7 segmentos para el "1".
    retlw   5Bh          ; El código 7 segmentos para el "2".
    retlw   4Fh          ; El código 7 segmentos para el "3".
    retlw   66h          ; El código 7 segmentos para el "4".
    retlw   6Dh          ; El código 7 segmentos para el "5".
    retlw   7Dh          ; El código 7 segmentos para el "6".
    retlw   07h          ; El código 7 segmentos para el "7".
    retlw   7Fh          ; El código 7 segmentos para el "8".
    retlw   67h          ; El código 7 segmentos para el "9".
    retlw   77h          ; El código 7 segmentos para el "A".
    retlw   7Ch          ; El código 7 segmentos para el "B".
    retlw   39h          ; El código 7 segmentos para el "C".
    retlw   5Eh          ; El código 7 segmentos para el "D".
    retlw   79h          ; El código 7 segmentos para el "E".
    retlw   71h          ; El código 7 segmentos para el "F".

END                ; Fin del programa.

```

EJERCICIO 2:

- a) El programa **Display_03.asm** realiza la siguiente tarea:
 Por el display de 7 segmentos conectado al Puerto B se visualiza una de las 26 letras del alfabeto internacional: de la "A" a la "Z". La letra a visualizar lo determina el orden leído por el Puerto A. Así por ejemplo:
- Si por el Puerto A se lee "---0000" (cero) la letra visualizada será la "A" que es la que está en el orden cero.
 - Si por el Puerto A se lee "--11001" (veinticinco) la letra visualizada será la "Z" que es la que está en el orden veinticinco.

Modifica el código del programa para que éste funcione sobre la tarjeta EasyPIC4 con el PIC16F877A, para ello:

- a. Utiliza el Puerto C en lugar del Puerto A para las entradas.
- b. Habilita el display 7 segmentos DIS0, poniendo a "1" el bit RA0.
- c. Utiliza el Puerto D en lugar del Puerto B para enviar el dato al display 7 segmentos.
- d. Modifica el modelo del microcontrolador

En la página siguiente viene el código del programa con las líneas de código que debéis modificar resaltadas. Además tendréis que añadir más instrucciones para habilitar DIS0, poniendo a "1" el bit RA0.

- b) Comprueba el funcionamiento del nuevo programa en la tarjeta EasyPIC4 con el PIC16F877A.

NOTA: Directiva DT

Para simplificar el uso de las instrucciones **retlw** el ensamblador MPASM facilita la directiva **DT (Define Tabla)** que sustituye el empleo repetitivo de muchas instrucciones **retlw**. Su sintaxis es:

DT <expr>[,<expr>,<expr>,...,<expr>]

Esta directiva genera durante la fase de ensamblado instrucciones **retlw**, una instrucción por cada <expr>. Cada carácter de una cadena es almacenado en su propia instrucción **retlw**.

Ejemplo:

```
DT 0x10, 0x11, 0x12
```

Esta directiva genera las instrucciones:

```
retlw 0x10
retlw 0x11
retlw 0x12
```

```

;***** Display_03.asm*****
;
;
;   =====
;   Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
;   E. Palacios, F. Remiro y L. López.
;   Editorial Ra-Ma. www.ra-ma.es
;   =====
;
; Por el display de 7 segmentos conectado al Puerto B se visualiza una de las 26 letras
; del alfabeto internacional: de la "A" a la "Z". La letra a visualizar lo determina el
; orden leído por el Puerto A. Así por ejemplo:
; - Si por el Puerto A se lee "---0000" (cero) la letra visualizada será la "A"
;   que es la que está en el orden cero.
; - Si por el Puerto A se lee "--11001" (veinticinco) la letra visualizada será la "Z"
;   que es la que está en el orden veinticinco.
;
; Por ahora no se contempla la posibilidad que el número de entrada sea mayor de 25.
;
; ZONA DE DATOS *****

        _CONFIG    _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
        LIST       P=16F84A
        INCLUDE    <P16F84A.INC>

; ZONA DE CÓDIGOS *****

ORG     0                ; El programa comienza en la dirección 0.
Inicio
        bsf       STATUS,RP0          ; Acceso al Banco 1.
        clrf     TRISE                ; Las líneas del Puerto B se configuran como salida.
        movlw    b'00011111'         ; Las 5 líneas del Puerto A se configuran como entrada.
        movwf    TRISA
        bcf      STATUS,RP0          ; Acceso al Banco 0.
Principal
        movf     PORTA,W              ; Lee la entrada.
        call    Letra_a_7Segmentos   ; Convierte a 7 Segmentos.
        movwf    PORTE                ; Resultado se visualiza por el puerto de salida.
        goto    Principal

; Subrutina "Letra_a_7Segmentos" -----
;
Letra_a_7Segmentos
        addwf    PCL,F
InicioTabla
        DT      77h, 7Ch, 39h, 5Eh, 79h, 71h, 6Fh, 76h, 19h, 1Eh, 7Ah, 38h, 37h
        DT      54h, 3Fh, 73h, 67h, 50h, 6Dh, 78h, 1Ch, 3Eh, 1Dh, 70h, 6Eh, 49h
FinTabla
        END

;
;   =====
;   Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
;   E. Palacios, F. Remiro y L. López.
;   Editorial Ra-Ma. www.ra-ma.es
;   =====
;

```


EJERCICIO 3:

- a) Confecciona un programa en ensamblador para el PIC16F877A, suponiendo que se ejecutará sobre la tarjeta EasyPIC4 ($F_{OSC}=8\text{Mhz}$), que haga una cuenta atrás de 9 a 0 y se visualice en el display 7 segmentos **DIS0** con una temporización de 1 segundo entre cada dígito. Utiliza el TIMER1 para la subrutina de retardo de 1 segundo.
- b) Comprueba el funcionamiento de tu programa en la tarjeta EasyPIC. Utiliza el PIC16F877A.

AYUDA: Ejemplo de subrutina de retardo de 1 segundo con el TIMER 1

```

.....
.....

movlw  b'00110001' ; se configura el Timer1 con un prescaler de 8
movwf  T1CON

.....
.....

; Subrutina "espera_1s" -----
; realiza 4 veces la temporizacion 250ms, es decir,4*250ms=1s
; utiliza un prescaler de 8

Carga equ .3036

espera_1s
    clrf Contador2
Loop
    movlw  low Carga
    movwf  TMR1L
    movlw  high Carga
    movwf  TMR1H
    bcf    PIR1,TMR1IF          ; Resetea el flag de desbordamiento del TMR1.
Rebosamiento
    btfss  PIR1,TMR1IF        ; ¿Se ha producido desbordamiento?
    goto   Rebosamiento      ; Todavía no. Repite.
    incf   Contador2,f
    movf   Contador2,W
    xorlw  .4                 ;Comprobación si ha llegado a 4 cuentas, es decir,4*250ms=1s
    btfss  STATUS,Z
    goto   Loop
    return

```

EJERCICIO 4:

- a) Confecciona un programa en ensamblador para el PIC16F877A, suponiendo que se ejecutará sobre la tarjeta EasyPIC4 ($F_{OSC}=8\text{Mhz}$), que produzca la **visualización de la palabra "HOLA"** en los displays 7 segmentos de la tarjeta EasyPIC. Para confeccionar el programa puedes utilizar el esquema que se muestra en la figura 6.
- b) Comprueba el funcionamiento de tu programa en la tarjeta EasyPIC. Utiliza el PIC16F877A.

AYUDA:

Dígito H → 0x76

Dígito O → 0x3F

Dígito L → 0x38

Dígito A → 0x77

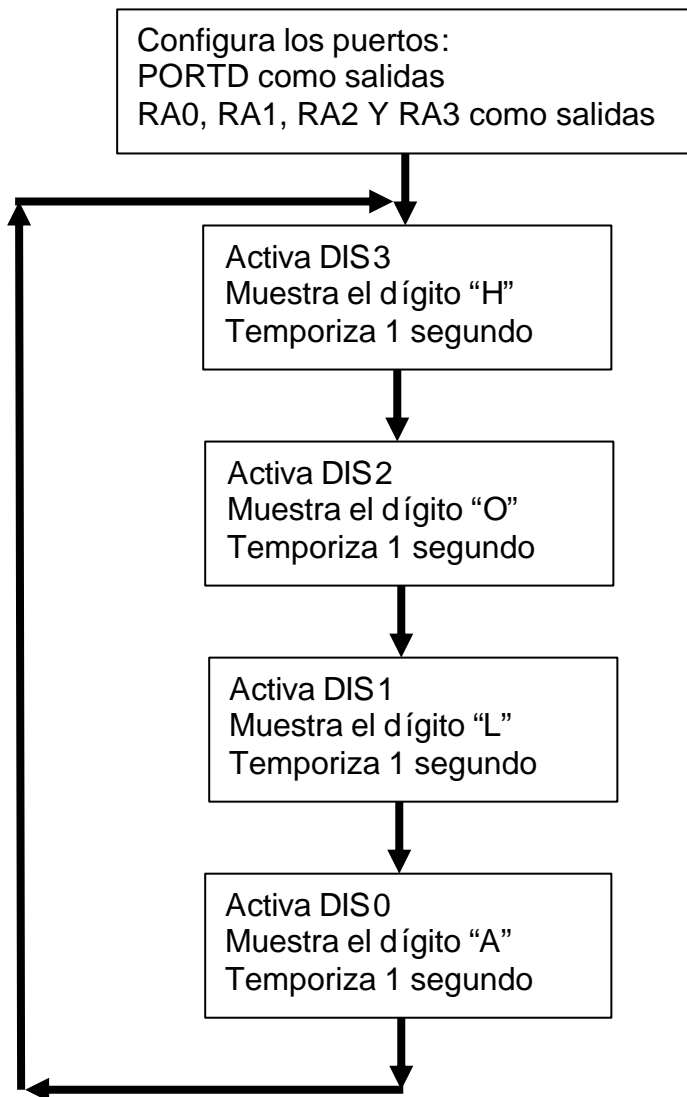


Figura 6. Ejercicio 4: visualización de la palabra "HOLA" en los displays 7 segmentos de la tarjeta EasyPIC.

EJERCICIO 5:

- a) Confecciona un programa en ensamblador para el PIC16F877A, suponiendo que se ejecutará sobre la tarjeta EasyPIC4 ($F_{OSC}=8\text{Mhz}$), que produzca la visualización de tu **nombre** (o bien de las siglas o bien de una abreviatura del mismo) en los displays 7 segmentos de la tarjeta EasyPIC. De manera que las letras aparezcan desde el display situado más a la derecha (DIS0) y se desplacen de dra a izqda hasta completar el nombre. La figura 7 muestra un esquema de la visualización que se pretende obtener.
- b) Comprueba el funcionamiento de tu programa en la tarjeta EasyPIC. Utiliza el PIC16F877A.

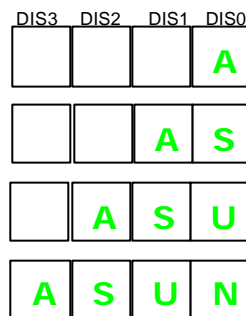


Figura 7. Ejercicio 5: visualización de tu nombre.