



Biblioteca De Códigos

Indice Del Documento:

1. [Activar-Desactivar-Puertos](#)
2. [Activar-Modo-Texto-LCD](#)
3. [Activar-Por-Bucle](#)
4. [Activar-Por -Etiqueta](#)
5. [Activar-Puertos-Condicional](#)
6. [Alias-Periféricos](#)
7. [Bucle-Acción](#)
8. [Capturar-Valor-Decimal](#)
9. [Configurar-Comunicación-Serial](#)
10. [Configurar-Entrada-Analógica](#)
11. [Configurar-Fuses-18F4550](#)
12. [Configurar-LCD](#)
13. [Configurar-LCD-Gráfico](#)
14. [Configurar-Memoria](#)
15. [Configurar-Pines-Digitales](#)
16. [Configurar-Puerto-I/O](#)
17. [Configurar-Teclado](#)
18. [Configurar-Variables](#)
19. [Contar-Pulsos](#)
20. [Organizar-Instrucciones](#)
21. [Posicionar-Cursor-LCD](#)
22. [Quiebre-De-Bucle](#)
23. [Retardos-Temporización](#)
24. [Configurar-Micro](#)
25. [Velocidad](#)

Abrir Archivos (Extensión Bas)

1. [Activar-Desactivar-Puertos](#)
2. [Activar-Modo-Texto-LCD](#)
3. [Activar-Por-Bucle](#)
4. [Activar-Por -Etiqueta](#)
5. [Activar-Puertos-Condicional](#)
6. [Alias-Periféricos](#)
7. [Bucle-Acción](#)
8. [Capturar-Valor-Decimal](#)
9. [Configurar-Comunicación-Serial](#)
10. [Configurar-Entrada-Analógica](#)
11. [Configurar-Fuses-18F4550](#)
12. [Configurar-LCD](#)
13. [Configurar-LCD-Gráfico](#)
14. [Configurar-Memoria](#)
15. [Configurar-Pines-Digitales](#)
16. [Configurar-Puerto-I/O](#)
17. [Configurar-Teclado](#)
18. [Configurar-Variables](#)
19. [Contar-Pulsos](#)
20. [Organizar-Instrucciones](#)
21. [Posicionar-Cursor-LCD](#)
22. [Quiebre-De-Bucle](#)
23. [Retardos-Temporización](#)
24. [Configurar-Micro](#)
25. [Velocidad](#)



```
*****
' * Name      : Activar-Desactivar_Puertos.BAS      *
' * Author    : [Adrián Manuel López Cosenza]      *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]       *
' *           : All Rights Reserved                 *
' * Date      : 23/05/2015                           *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO      *
' * Notes     : Leer datasheet antes de configurar  *
' *                                           *
*****
```

' Sintaxis:

```
'High Port      ' High es una palabra reservada por PROTON para activar
'High port.pin  ' La sintaxis correcta es indicar la acción con la palabra reservada
                  seguida de la dirección del puerto
'Low Port       ' Low es una palabra reservada por PROTON para desactivar (APAGAR)
'Low port.pin   ' La sintaxis correcta es indicar la acción con la palabra reservada
                  seguida de la dirección del puerto
```

' Ejemplo:

```
High PORTB      ' Esto activa todas las salidas del puerto B (Debe haberse declarado el
                  puerto B como salida)
Low PORTB        ' Esto desactiva todas las salidas del puerto B

High PORTB      ' También expresa que el puerto B cambiara al estado ALTO (ACTIVO)
Low PORTB       ' También expresa que el puerto B cambiara al estado BAJO (DESACTIVADO)
```

' De igual manera podemos utilizar las siguientes sintaxis:

```
PORTB=0          ' El puerto B a nivel BAJO (APAGADO)
PORTB.0=0        ' El puerto B.0 solo el pin 0 con nivel BAJO (APAGADO)
PORTB=%00000000  ' Todo el puerto B con nivel bajo (APAGADO)
PORTB=1          ' Puerto B solo el pin 0 con nivel ALTO (ACTIVADO)
PORTB.0=1        ' Puerto B solo el pin 0 con nivel ALTO (ACTIVADO)
PORTB=7          ' El puerto B reproduce el dato 00000111 Decimal
PORTB=3          ' El puerto B reproduce el dato 00000011 Decimal
PORTB=$F         ' El puerto B reproduce el dato 00001111 Hexadecimal
PORTB=$3F        ' El puerto B reproduce el dato 00111111 Hexadecimal
PORTB=%11111111  ' Todo el puerto con nivel ALTO (ACTIVADO)
```





```
*****
' * Name      : Activar_Modo_Texto_LCD.BAS      *
' * Author    : [Adrián Manuel López Cosenza]   *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]    *
' *           : All Rights Reserved             *
' * Date      : 03/06/2015                      *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO   *
' * Notes     :                                  *
' *           :                                  *
*****
```

```
' Se recomienda completar con espacios en blanco los caracteres no utilizados,
Máximo 16
' La instrucción imprime en pantalla el texto escrito entre "comillas dobles"
```

```
Print At 1,1, "Mostrar Mensaje"      ' Escribir "Mostrar Mensaje" en la linea 1
posición 1
Print At 2,1, "Segundo Mensaje"      ' Escribir "Segundo Mensaje" en la linea 2
posición 1
DelayMS 5000 ' Retardo del estado del micro durante 5 segundos, se puede ajustar a
valores menores o mayores.
```

```
' Nota:
' Antes de continuar con la ejecución debe haber una pausa.
```





```
*****
' * Name      : Activar_Por_Bucle.BAS                      *
' * Author    : [Adrián Manuel López Cosenza]              *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]               *
' *           : All Rights Reserved                         *
' * Date      : 03/06/2015                                  *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO              *
' * Notes     :                                              *
' *           :                                              *
*****
Device 18F4550      ' Micro-controlador seleccionado como ejemplo
*****
' Ejemplo 1

Inicio:              ' Etiqueta principal, ejecuta una rutina
PORTD.2=1            ' Activar (ALTA) PORTD PIN 2
DelayMS 200          ' Esperar 2 mili-segundos
PORTD.2=0            ' Desactivar(CERO) PORTD PIN 2
DelayMS 200          ' Esperar 2 mili-segundos
GoTo Inicio          ' Termina la rutina y regresa a la etiqueta

*****
' Ejemplo 2

Loop:                ' Etiqueta de secuencia, ejecuta una rutina
LED = 1              ' Activar LED
DelayMS 300          ' Esperar 3 mili-segundos
For scan = 1 To 7    ' Comienzo del ciclo FOR Escanea hasta completar 7
LED = LED << 1       ' Mientras LED sea menor a 7 debe continuar
DelayMS 300          ' Esperar 3 mili-segundos
Next                 ' Siguiente
GoTo Loop            ' Regresa a loop

*****
' Ejemplo 3

Otra:
For n=1 To 50 Step 2  ' Etiqueta de secuencia, ejecuta una rutina
PulsOut PORTD.0,2500  ' Mantén el pulso en PORTD PIN 0 por 2 segundos y medio
DelayMS 10            ' Esperar 0.1 mili-segundos
Next                  ' Siguiente
End                   ' FIN de la rutina
```





```
! *****
! *   Name       : Activar_Por _Etiqueta.BAS                      *
! *   Author    : [Adrián Manuel López Cosenza]                  *
! *   Notice    : Copyright (c) 2015 [GNU-GPL3]                   *
! *              : All Rights Reserved                             *
! *   Date      : 23/05/2015                                       *
! *   Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO                  *
! *   Notes     :                                                  *
! *                                                      *
! *****
```

' Útil para realizar acciones que no requieren lazos de control, o como parte de un lazo que ejecuta acciones inmutables.

GoSub Gira ' Activar una rutina llamada Gira

' La rutina necesita esta sintaxis:

Gira:

' Lo que queremos que haga el micro-controlador

Return ' Retornar al punto de inicio





```
! *****
! *   Name      : Activar_Puertos_Condicional.BAS      *
! *   Author    : [Adrián Manuel López Cosenza]       *
! *   Notice    : Copyright (c) 2015 [GNU-GPL3]        *
! *             : All Rights Reserved                 *
! *   Date      : 23/05/2015                          *
! *   Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO      *
! *   Notes     :                                       *
! *                                                    *
! *****

' Sintaxis:   If (EXPRESSION) Then (INSTRUCTIONS)

If S1=0 Then PROG ' El micro-controlador espera que S1 sea 0 para activar una
                  rutina .
If S1=0 Then PROG2 ' El micro-controlador espera que S1 sea 1 para activar una
                  rutina .

If S1=0 And S2=0 And S3=0 Then PROG3 ' En este caso para ejecutar la rutina S1, S2
                                     y S3 deben ser =0

EndIf ' así debe terminar una condicional que no posee bucle (retorno)
```





```
*****
' * Name      : Alias_Perifericos.BAS                      *
' * Author    : [Adrián Manuel López Cosenza]              *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]               *
' *           : All Rights Reserved                         *
' * Date      : 23/05/2015                                  *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO              *
' * Notes     : Leer datasheet antes de configurar          *
' *                                                    *
*****
```

' Un alias o un modificador es una palabra que expresa un dispositivo conectado a
a un puerto:

Symbol MOTOR = PORTA.1

' Establece que existe un motor conectado en el puerto A pin 1

Symbol SW = PORTA.2

' Establece que existe un interruptor conectado en el puerto A pin 2

Symbol S1 = PORTA.3

' Establece que existe un interruptor conectado en el puerto A pin 3





```
! *****
! *   Name       : Bucle_Accion.BAS                               *
! *   Author    : [Adrián Manuel López Cosenza]                   *
! *   Notice    : Copyright (c) 2015 [GNU-GPL3]                     *
! *              : All Rights Reserved                             *
! *   Date      : 03/06/2015                                         *
! *   Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO                   *
! *   Notes     :                                                  *
! *              :                                                  *
! *****
```

' Esto ejecuta un bloque de instrucciones hasta que una condición sea verdadera

```
var1 = 1      ' Variable a incrementar
Repeat       ' Repetir hasta ser mayor del valor declarado
Print At 1, 1, Dec var1
```

' También podemos utilizar

```
Repeat High led: Until PORTA.0 = 1
DelayMS 200      ' Evitar rebote
Inc var1         ' Incrementar la variable
Until var1 > 10  ' Con esto indicamos que al llegar a 10 rompa el bucle
```





```
! *****
! * Name      : Capturar_Valor Decimal.BAS          *
! * Author    : [Adrián Manuel López Cosenza]       *
! * Notice    : Copyright (c) 2015 [GNU-GPL3]        *
! *           : All Rights Reserved                 *
! * Date      : 03/06/2015                          *
! * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO      *
! * Notes     :                                       *
! *           :                                       *
! *****
```

' Antes de proceder a capturar el valor se debe haber establecido la variable,e
invocar el nemonico respectivo

```
Dim cc As Word
```

```
Print At 1,1, "f -", Dec cc ' Imprimir el dato en el LCD desde la variable cc
```





```
*****
' * Name      : Configurar_Comunicación_Serial.BAS      *
' * Author    : [Adrián Manuel López Cosenza]          *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]           *
' *           : All Rights Reserved                     *
' * Date      : 23/05/2015                              *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO          *
' * Notes     : Varía según el micro-controlador        *
' *           : Leer la datasheet correspondiente antes de *
' *           : seleccionar el puerto                    *
*****
Device 16F877

Serial_Baud = 9600
Rsout_Pin = PORTC.6
Rsout_Mode = TRUE
Rsout_Pace = 1
Rsin_Pin = PORTC.7
Rsin_Mode = TRUE

Hserial_Baud = 9600      ' 1# Speed in baud = 9600
Hserial_RCSTA = %10010000 ' 2# Enable serial port and continuous receive
Hserial_TXSTA = %00100100 ' 3# Enable transmit and asynchronous mode
Hserial_Clear = On       ' 4# Enable error clearing on received characters

' Notas:
' 1# Velocidad en baudios = 9600
' 2# Activar puerto serial y recepción continua
' 3# Activar modo de transmisión asíncrona
' 4# Activar auto limpieza al recibir caracteres
```





```
! *****
! * Name      : Configurar_Entrada_Analogica.BAS      *
! * Author    : [Adrián Manuel López Cosenza]        *
! * Notice    : Copyright (c) 2015 [GNU-GPL3]         *
! *           : All Rights Reserved                   *
! * Date      : 03/06/2015                             *
! * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO        *
! * Notes     :                                         *
! *           :                                         *
! *****

Declare Adin_Res    10      ' Resolución del convertidor analógico digital
Declare Adin_Tad    2       ' CLK options are 0,1,2,3 (are based internal cycles)
Declare Adin_Stime  100     ' Tiempo de muestreo

adcon1 = %10000000         ' Setup del registro ADCON1, AN1 / RA1 es Analógico

' Nota para mayor información refiera al datasheet del micro-controlador
```





```
*****
! * Name      : Configurar_Fuses_18F4550.BAS                               *
! * Author    : [Adrián Manuel López Cosenza]                             *
! * Notice    : Copyright (c) 2015 [GNU-GPL3]                               *
! *           : All Rights Reserved                                         *
! * Date      : 23/05/2015                                                  *
! * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO                             *
! * Notes     :                                                             *
! *           :                                                             *
*****
```

Device 18F4550

```
*****
```

Config_Start

```
PLLDIV = 5           ' Divide por 5 (20 MHz la entrada del oscilador)
CPUDIV = OSC1_PLL2   ' [OSC1/OSC2 Src: /1][96 MHz PLL Src: /2]
USBDIV = 2           ' La fuente del reloj USB viene del lazo de enganche de
                      ' fase a 96 MHz/2
FOSC = HSPLL_HS      ' El oscilador HS, lazo de enganche de fase habilitado,
                      ' HS usado por USB
FCMEN = OFF          ' El monitor contra fallos del Reloj - Deshabilitado
IESO = On            ' El modo del oscilador Switchover - Habilitado
PWRT = On            ' PWRT - Habilitado
BOR = OFF            ' Brown-out - Reset/Habilitado en hardware y software
BORV = 0             ' Configuración Máxima
VREGEN = On          ' Regulador de voltaje USB / Habilitado
WDT = OFF            ' HW Deshabilitado /SW Controlado
WDTPS = 1            ' 1:1
MCLR = On            ' MCLR Habilitado ; RE3 entrada Deshabilitada
LPT1OSC = OFF        ' Timer1 Configurado Para la operación n máxima de poder
PBADEN = OFF         ' PORTB <4:0> Pines configurados como digital I/O en
                      ' Reset
CCP2MX = On          ' CCP2 Entrada y Salida multiplexada con RC1
STVREN = OFF         ' La pila en el desbordamiento negativo completo no
                      ' causar Reanudación n
LVP = OFF            ' Single-Supply ICSP Deshabilitado
XINST = OFF          ' La instrucción Coloca la extensión n, y el modo de
                      ' Direccinamiento
                      ' Indexado deshabilitadas (modo Legado)
Debug = OFF          ' El depurador de fondo deshabilitado, RB6 Y RB7
                      ' configurados
                      ' Como de propósito general Pines I/O
CP0 = OFF            ' Block 0 (000800-001FFFh) Sin protección de código
CP1 = OFF            ' Block 1 (002000-003FFFh) Sin protección de código
CP2 = OFF            ' Block 2 (004000-005FFFh) Sin protección de código
CPB = OFF            ' Boot block (000000-0007FFFh) Sin protección de código
CPD = OFF            ' Data EEPROM Sin protección de código
WRT0 = OFF           ' Block 0 (000800-001FFFh) Sin protección de código
WRT1 = OFF           ' Block 1 (002000-003FFFh) Sin protección de código
WRT2 = OFF           ' Block 2 (004000-005FFFh) Sin protección de código
WRTB = OFF           ' Boot block (000000-0007FFFh) Sin protección de código
WRTC = OFF           ' Configuration registers (300000-3000FFFh) Sin protección
                      ' de código
WRTD = OFF           ' Data EEPROM Sin protección de código
EBTR0 = OFF          ' Block 0 (000800-001FFFh) No protegido de lecturas de
                      ' las tablas ejecutadas en otros bloques
EBTR1 = OFF          ' Block 1 (002000-003FFFh) No protegido de lecturas de
                      ' las tablas ejecutadas en otros bloques
EBTR2 = OFF          ' Block 2 (004000-005FFFh) No protegido de lecturas de
                      ' las tablas ejecutadas en otros bloques
EBTRB = OFF          ' Boot block (000000-0007FFFh) No protegido de lecturas de
                      ' las tablas ejecutadas en otros bloques
Config_End
```



```
*****
*****
' * Name      : Configurar_LCD.BAS                      *
' * Author    : [Adrián Manuel López Cosenza]          *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]           *
' *           : All Rights Reserved                     *
' * Date      : 23/05/2015                              *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO          *
' * Notes     : Valido para 16X2 16X4                   *
' *           : Antes de configurar revisar la datasheet *
' *           : del dispositivo.                         *
*****
'Seleccionar dispositivo
*****
Device 18F4550
*****
' Declarar Puertos
*****
Declare LCD_Type = 0
Declare LCD_DTPin = PORTD.0
Declare LCD_ENPin = PORTD.5
Declare LCD_RSPin = PORTD.4
Declare LCD_Interface = 4
Declare LCD_Lines = 2      ' Si se utiliza una 16X4 LCD_Lines debe ser = 4
```





```
*****
' * Name      : Configurar_LCD_Grafico.BAS      *
' * Author    : [Adrián Manuel López Cosenza]   *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]    *
' *           : All Rights Reserved             *
' * Date      : 23/05/2015                      *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO   *
' * Notes     : Valido para LCD convencional     *
' *           : Antes de configurar revisar la datasheet *
*****
'Seleccionar dispositivo
*****
Device 18F4550
*****
' Declarar Puertos
*****

LCD_DTPort = PORTD
LCD_RSPin  = PORTC.1
LCD_ENPin  = PORTE.0
LCD_RWPin  = PORTC.0
LCD_CS1Pin = PORTE.1
LCD_CS2Pin = PORTE.2
LCD_Type   = GRAPHIC      ' Escribir en mayúsculas
Internal_Font = OFF       ' Apagar las fuentes internas
Font_Addr  = 0
```





```
! *****
! *   Name      : Configurar_Memoria.BAS                               *
! *   Author    : [Adrián Manuel López Cosenza]                       *
! *   Notice    : Copyright (c) 2015 [GNU-GPL3]                         *
! *             : All Rights Reserved                                  *
! *   Date      : 23/05/2015                                           *
! *   Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO                       *
! *   Notes     :                                                       *
! *             :                                                       *
! *****
! Este ejemplo no aplica para todos lo micro-controladores.
! En todo caso remítase al datasheet del dispositivo.

! BUS I2C
SCL_Pin= PORTA.7
SDA_Pin= PORTA.6
```





```
! *****
! * Name      : Configurar_Pines_Digitales.BAS      *
! * Author    : [Adrián Manuel López Cosenza]      *
! * Notice    : Copyright (c) 2015 [GNU-GPL3]       *
! *           : All Rights Reserved                 *
! * Date      : 23/05/2015                           *
! * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO      *
! * Notes     :                                       *
! *           :                                       *
! *****
```

```
' CONFIGURAR los pines como digital
' Ejemplo:
```

```
All_Digital= true ' Puertos A y B
```





```
*****
' * Name      : Configurar_Puerto_I_O.BAS      *
' * Author    : [Adrián Manuel López Cosenza]  *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]   *
' *           : All Rights Reserved            *
' * Date      : 23/05/2015                     *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO  *
' * Notes     : Leer datasheet antes de configurar *
' *                                           *
*****
```

```
' Esto es poco claro en principio, sin embargo vamos a explicarlo
' Declaramos un 18F4550
```

```
*****
Device 18F4550
*****
' Para declarar la totalidad del puerto como entrada
*****
```

```
Input PORTC      ' Todo el puerto es de entrada
Input PORTC.0     ' Solo el puerto C.0 es entrada, el resto es salida
```

```
' También podemos utilizar esta forma:
```

```
TRISC=1          ' Todo el puerto es de entrada
TRISC.0-1        ' Solo el puerto C.0 es entrada, el resto es salida
```

```
' Si se prefiere podemos utilizar el método binario:
```

```
TRISC=%11111111 ' Todo el puerto es entrada
TRISC=%10000000 ' Solo el puerto C.0 es entrada, el resto es salida
```

```
' En hexadecimal seria así:
```

```
TRISC=$FF        ' Todo el puerto es entrada
```

```
' En decimal seria así:
```

```
TRISC=255        ' Todo el puerto es entrada
```

```
*****
' Una manera de entender esto seria así:
*****
```

```
' Cada pin del dispositivo representa un puerto, si asignamos un uno a cada puerto,
' este se convierte en entrada, si lo colocamos en cero es una salida.
```

```
PORTC.0 PORTC.1 PORTC.2 PORTC.3 PORTC.4 PORTC.5 PORTC.6 PORTC.7
% 1      1      1      1      1      1      1      1
' Todo el puerto es entrada
```

```
% 1      0      0      0      0      0      0      0
' Puerto C.0 es una entrada, el resto es salida
```

```
% 0      1      1      0      0      0      1      0
' Puerto C.0, C.3, C.4, C.5 y C.7 son entradas - Puerto C.1, C.2 y C.6 son salidas
```

```
' para expresar un valor con cualquier método solo necesitamos transformarlo
```

```
' En binario / En Hexadecimal / En Decimal
11111111    FF          255
11111110    FE          254
11111100    FC          252
11111000    F8          248
```



```
*****
' Para declarar la totalidad del puerto como salida
*****

Output PORTC      ' Todo el puerto es una salida
Output PORTC.0     ' Solo el puerto C.0 es salida, el resto es entrada

' También podemos utilizar esta forma:

TRISC=0           ' Todo el puerto es salida
TRISC.0-0         ' Solo el puerto C.0 es salida, el resto es entrada

' Si se prefiere podemos utilizar el método binario:

TRISC=%00000000   ' Todo el puerto es salida
TRISC=%01111111   ' Solo el puerto C.0 es salida, el resto es entrada

' En hexadecimal seria así:
TRISC=$0          ' Todo el puerto es salida

' En decimal seria así:
TRISC=0           ' Todo el puerto es una salida
*****

' Configuración Combinada

TRISC=%00001111   ' Declaramos que el puerto C Será entrada para los pines 0,1,2,3 y
salida para los pines 4,5,6 y 7
TRISC=$F          ' Declaramos que el puerto C Será entrada para los pines 0,1,2,3 y
salida para los pines 4,5,6 y 7
TRISC=15          ' Declaramos que el puerto C Será entrada para los pines 0,1,2,3 y
salida para los pines 4,5,6 y 7

' Hay que estar muy pendiente cuando utilizamos combinaciones, al momento de utilizar
el dispositivo

' se pueden ocasionar daños irreversibles al mismo. Es conveniente simular el
funcionamiento.
```





```
! *****
! *   Name      : Configurar_Teclado.BAS                               *
! *   Author    : [Adrián Manuel López Cosenza]                       *
! *   Notice    : Copyright (c) 2015 [GNU-GPL3]                         *
! *             : All Rights Reserved                                   *
! *   Date      : 23/05/2015                                           *
! *   Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO                       *
! *   Notes     : Varía según el micro-controlador                     *
! *             :                                                       *
! *****
```

Keypad_Port = PORTB

' Leer la datasheet correspondiente antes de seleccionar el puerto





```
*****
' * Name      : Configurar_Variables.BAS                      *
' * Author    : [Adrián Manuel López Cosenza]                 *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]                 *
' *           : All Rights Reserved                             *
' * Date      : 23/05/2015                                      *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO                 *
' * Notes     :                                                *
' *           :                                                *
*****
' El nombre de la variable lo asignamos a nuestra entera conveniencia.

'Ejemplos: (Utilizar nemonicos fáciles de recordar)

' Declaración   Variable      Palabra
Dim             Tecla_1       As Byte
Dim             Salida_0     As Word
Dim             Contador-a    As Dword

' No es correcto expresarlo de esta forma, puede provocar errores.

Tecla_1        var Byte
Salida_0       var Word
Contador-a     var Dword

' No utilizar nemonicos de esta forma

' Declaración   Variable      Palabra
Dim             pos1         As Byte
Dim             7este        As Byte
Dim             motorderecho As Byte
Dim             sensor       As Word

' Trate de asociar sus nemonicos a la acción que representa.
```





```
*****
!* Name      : Contar_Pulsos.BAS                      *
!* Author    : [Adrián Manuel López Cosenza]          *
!* Notice    : Copyright (c) 2015 [GNU-GPL3]           *
!*           : All Rights Reserved                     *
!* Date      : 03/06/2015                              *
!* Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO          *
!* Notes     :                                          *
!*           :                                          *
*****
```

Dim WRD As Word ' Declarar la variable

WRD = Counter Pin , 1000 ' La variable WRD contiene el valor en 1 segundo

' Se puede utilizar cualquier nemonico

' Para contar eventos y utilizarlos como condición

' Recuerda que debe existir una rutina a donde ir y su respectivo retorno al programa principal

If WRD=1 Then (Aquí va la acción)

DelayMS 500

cont1 = cont1 + 1 ' Esto ejecuta una suma en la memoria

EndIf ' Final de la condicional





```
! *****
! *   Name       : Organizar_Instrucciones.BAS           *
! *   Author    : [Adrián Manuel López Cosenza]         *
! *   Notice    : Copyright (c) 2015 [GNU-GPL3]          *
! *              : All Rights Reserved                   *
! *   Date      : 03/06/2015                             *
! *   Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO        *
! *   Notes     :                                         *
! *              :                                         *
! *****
```

' Una buena organización ayuda a evitar conflictos, o instrucciones que prelan a otras

```
If var1 = 0 Then GoTo LAB_0  Branch VAR1, (LAB_0 LAB_1, LAB_2)
If var1 = 1 Then GoTo LAB_1  On var1 GoTo LABEL_0, LABEL_1, LABEL_2
If var1 = 2 Then GoTo LAB_2  On var1 GoSub LABEL_0, LABEL_1, LABEL_2
```





```
*****
' * Name      : Posicionar_Cursor_LCD.BAS          *
' * Author    : [Adrián Manuel López Cosenza]      *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]       *
' *           : All Rights Reserved                 *
' * Date      : 03/06/2015                          *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO      *
' * Notes     :                                     *
' *           :                                     *
*****
```

' Útil para evitar errores al imprimir el mensaje en la LCD

```
Dim Line As Byte
Dim Xpos As Byte
Line = 2
Xpos = 1
Cls
Print "Hola" ' Display LCD
Cursor Line, Xpos ' Linea 2, posición 1
```





```
! *****
! *   Name       : Quiebre_De_Bucle.BAS                               *
! *   Author    : [Adrián Manuel López Cosenza]                       *
! *   Notice    : Copyright (c) 2015 [GNU-GPL3]                       *
! *              : All Rights Reserved                                *
! *   Date      : 03/06/2015                                           *
! *   Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO                      *
! *   Notes     :                                                       *
! *              :                                                       *
! *****
```

```
If var1 = 10 Then Break      ' Breack rompe el bucle cuando llega a 10
DelayMS 200                  ' No olvidar una pausa para evitar efecto rebote
```





```
! *****
! * Name      : Retardos_Temporizacion.BAS          *
! * Author    : [Adrián Manuel López Cosenza]      *
! * Notice    : Copyright (c) 2015 [GNU-GPL3]       *
! *           : All Rights Reserved                 *
! * Date      : 03/06/2015                           *
! * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO      *
! * Notes     :                                     *
! *           :                                     *
! *****
```

' Un retardo es una pausa en la constante tiempo de ejecución, en algunas ocasiones es necesario

' establecer una pausa de al menos 1 mili-segundo para evitar el efecto rebote en el micro-controlador

```
DelayMS 1000 ' Retardo del estado del micro durante 1 segundo
DelayMS 500  ' Retardo del estado del micro durante 1/2 segundo
DelayUS 1000 ' Retardo del estado del micro durante 1 mili-segundo
pause 1000   ' No se recomienda este tipo de instrucción
```





```
*****
' * Name      : Configurar_Micro.BAS                      *
' * Author    : [Adrián Manuel López Cosenza]             *
' * Notice    : Copyright (c) 2015 [GNU-GPL3]              *
' *           : All Rights Reserved                        *
' * Date      : 23/05/2015                                  *
' * Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO             *
' * Notes     :                                             *
' *           :                                             *
*****
```

```
' CONFIGURAR EL MODELO DE MICRO-CONTROLADOR
' Ejemplo:
```

Device 16F870

```
'device 16F873
'device 16F873A
'device 18F4520
'Device 18F4550
```

```
' Eliminar el (') antes de la palabra Device descomentar la linea.
' Nota: No puedes seleccionar mas de 1 dispositivo a la vez.
```





```
! *****
! *   Name       : Velocidad.BAS                               *
! *   Author    : [Adrián Manuel López Cosenza]               *
! *   Notice    : Copyright (c) 2015 [GNU-GPL3]                *
! *              : All Rights Reserved                         *
! *   Date      : 23/05/2015                                    *
! *   Versión   : 1.0 / NO COMPILAR ESTE EJEMPLO              *
! *   Notes     : Unidad Mhz                                    *
! *              :                                              *
! *****
```

```
' CONFIGURAR EL OSCILADOR DEL MICRO-CONTROLADOR
' Ejemplo:
```

Xtal 4

```
'Xtal 10
```

```
'xtal 20
```

```
'xtal 40
```

```
' Eliminar el (') antes de la palabra Xtal descomentar la linea.
```

```
' Nota: No puedes seleccionar mas de 1 Xtal a la vez.
```

