

CONCEPTO DE SUBRUTINA

En informática, una subrutina o subprograma (también llamada procedimiento, función, rutina o método), como idea general, se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica.

Se le llama subrutina a un segmento de código separado del bloque principal, el cual puede ser invocado en cualquier momento desde ésta u otra subrutina.

Una subrutina, al ser llamada dentro de un programa, hace que el código principal se detenga y se dirija a ejecutar el código de la subrutina.

Una subrutina es una porción de código que forma parte de un programa más grande. Esa porción de código realiza una tarea específica, relativamente independiente del resto del código. La mayoría de los lenguajes de programación soportan la creación de subrutinas y métodos para llamarlas (invocarlas) y retornarlas.

Existen varias ventajas de "romper" un programa en subrutinas:

- * Reducción de código duplicado.
- * Permite la reutilización de código en múltiples programas.
- * Descomposición de problemas complejos en simples piezas (lo que aumenta el mantenimiento y posibles mejoras).
- * Aumenta la legibilidad del código de un programa.

EJEMPLO DE PROGRAMA QUE UTILIZA SUBRUTINAS

Este programa rastrea el puerto a en las entradas RA0 - RA1- RA2-RA3 en forma secuencial buscando un cero, en la entrada que lo encuentra ejecuta la tarea asociada correspondiente, tarea0 - tarea1 - tarea2 - tarea3. No se chequean las entradas nuevamente hasta que se ejecuto la tarea iniciada.

```
list p=16f84
#include p16f84.inc
portb equ 06h ;el puerto b esta en la posicion 06h de la ram
porta equ 05h ;el puerto a esta en la posicion 05h de la ram
trisb equ 86h ;regis de config de puerto b esta en la direcc 86h de la ram
trisa equ 85h
status equ 03h ;registro de estado esta en la direcc 03h de la ram

;se definen5 registros personales.
tempo1 equ 0ch
tempo2 equ 0dh ;registro tempo2 esta en la direcc 10h de la ram
tempo3 equ 0eh ;registro tempo3 esta en la direcc 11h de la ram
;-----
;definicion de la funcion de los puertos

bsf status,5 ;se posiciona en banco1
clrf trisb ;define el portb como salida
movlw b'00001111'
movwf trisa ;define RA0 RA1 RA2 RA3 RA4 como entradas
bcf status,5; se posiciona en banco 0
;-----
;---parte principal del programa que rastrea el estado de 4 entradas del porta---

rastreo
    movlw b'11111111'
    movwf portb ; apaga todo el display de anodo comun
    btfss porta,0 ; prueba si es 1 el bit 0 del porta. Si lo es salta
    call tarea0
    btfss porta,1 ; prueba si es 1 el bit 1 del porta. Si lo es salta
    call tarea1
    btfss porta,2 ; prueba si es 1 el bit 2 del porta. Si lo es salta
    call tarea2
    btfss porta,3 ; prueba si es 1 el bit 3 del porta. Si lo es salta
    call tarea3
    goto rastreo
;----fin parte principal del programa que rastrea el estado de 4 entradas del porta--
```

```
-----  
;  
;Comienzan las tareas de encender display anodo comun con el nro de tarea  
;Son 4 subrutinas
```

```
tarea0  
    movlw b'01000000'; decimal 0  
    movwf portb  
    call retardo1  
    return
```

```
tarea1  
    movlw b'01111001'; decimal 1  
    movwf portb  
    call retardo1  
    return
```

```
tarea2  
    movlw b'00100100'; decimal 2  
    movwf portb  
    call retardo1  
    return
```

```
tarea3  
    movlw b'00110000'; decimal 3  
    movwf portb  
    call retardo1  
    return
```

```
-----  
;  
;Subrutina retardo
```

```
retardo1  
    incf tempo1,1  
retardo2  
    incf tempo2,1  
retardo3  
    incf tempo3,1  
    btfss tempo3,6  
    goto retardo3  
    clrf tempo3  
    btfss tempo2,7  
    goto retardo2  
    clrf tempo2  
    btfss tempo1,6  
    goto retardo1  
    clrf tempo1  
    return
```

```
-----  
;  
end
```