

V.- El manejo de las Interrupciones

Conceptos Básicos

Modos de Manejo

Ejemplos de aplicación

Carlos E. Canto Quintal

El manejo de las Interrupciones en el 8051

Las interrupciones son fáciles de entender con una analogía humana.

Supongamos que estás trabajando en tu escritorio y alguien te llama por teléfono. Esto es una Interrupción.

La persona que te llamó por teléfono podría darte diferentes mensajes, por ejemplo:

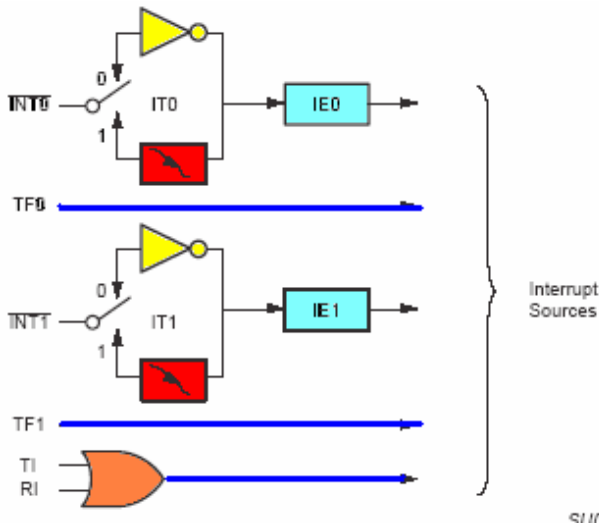
- *"Se está quemando tu casa!"*
- *"Llena por favor estos documentos."*
- *" ¡Levántate!"*

Tu respuesta en cada caso será bastante diferente. Lo mismo es con las interrupciones en los Microcontroladores.

Las interrupciones son mensajes del mundo exterior. Se puede escribir software para reaccionar a los mensajes si lo deseas o bien simplemente se pueden ignorar los mensajes.

Carlos E. Canto Quintal

El manejo de las Interrupciones en el 8051



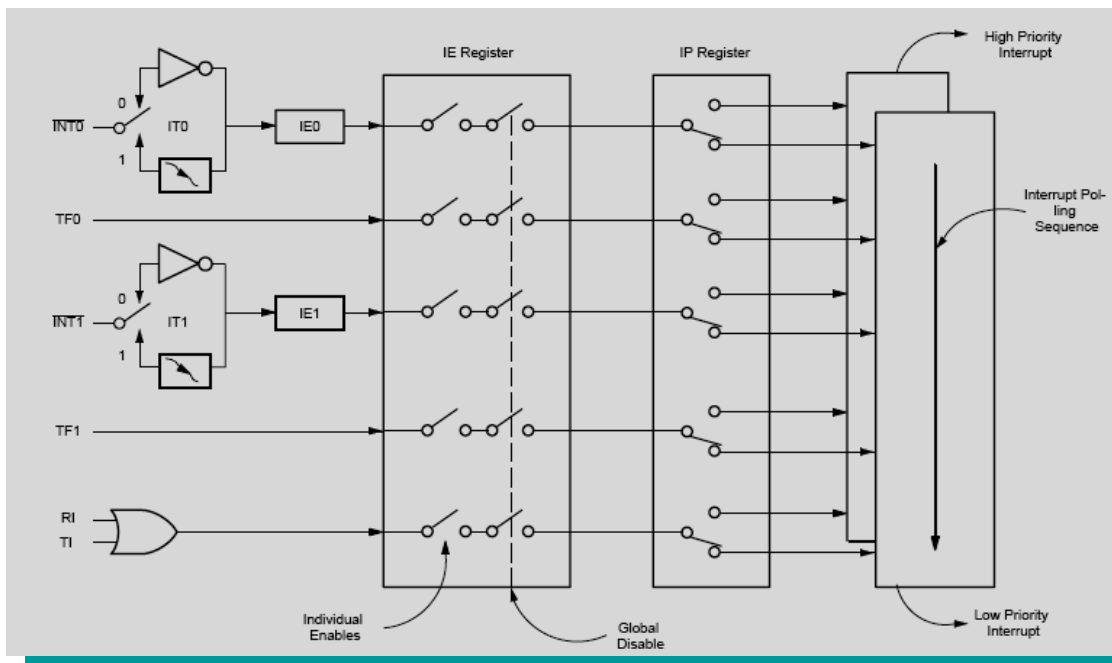
The 80C51 provides 5 interrupt sources. The External Interrupts INT0 and INT1 can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON.

The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated.

If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

Carlos E. Canto Quintal

Sistema de control de las interrupciones



TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

Individual
Enables

Global
Disable

Low Priority
Interrupt

Microcontroladores

Las Interrupciones

El manejo de las Interrupciones en el 8051

Habilitando y deshabilitando Interrupciones
Las interrupciones pueden ser habilitadas y deshabilitas activando algunas banderas de cierto registro.

El manejo de las Interrupciones en el 8051

Registro habilitador de las Interrupciones (IE: Interrupts Enable Register)



BIT	SYMBOL	FUNCTION
IE.7	EA	Disables all interrupts. If EA=0, no interrupt will be acknowledged. If EA=1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
IE.6	—	Reserved.
IE.5	—	Reserved.
IE.4	ES	Enables or disables the Serial Port interrupt. If ES=0, the Serial Port interrupt is disabled.
IE.3	ET1	Enables or disables the Timer 1 Overflow interrupt. If ET1=0, the Timer 1 interrupt is disabled.
IE.2	EX1	Enables or disables External Interrupt 1. If EX1=0, External interrupt 1 is disabled.
IE.1	ET0	Enables or disables the Timer 0 Overflow interrupt. If ET0=0, the Timer 0 interrupt is disabled.
IE.0	EX0	Enables or disables External interrupt 0. If EX0=0, External interrupt 0 is disabled.

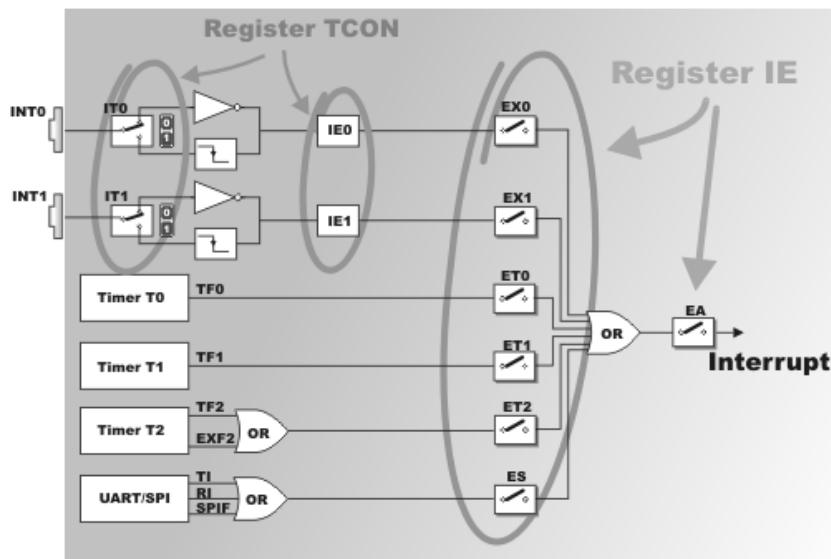
Carlos E. Canto Quintal

Registro de Control de Timers/Contadores (TCON: Timer/Counter Control Register)



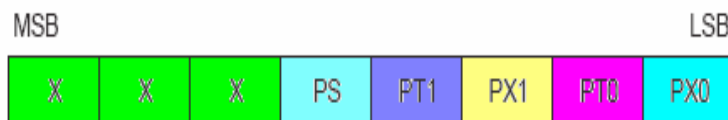
TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

Carlos E. Canto Quintal



El manejo de las Interrupciones en el 8051

Registro de Prioridades de las Interrupciones (IP: Interrupts Priority Register)



BIT	SYMBOL	FUNCTION
IP.7	—	Reserved.
IP.6	—	Reserved.
IP.5	—	Reserved.
IP.4	PS	Defines the Serial Port interrupt priority level. PS=1 programs it to the higher priority level.
IP.3	PT1	Defines the Timer 1 interrupt priority level. PT1=1 programs it to the higher priority level.
IP.2	PX1	Defines the External Interrupt 1 priority level. PX1=1 programs it to the higher priority level.
IP.1	PT0	Enables or disables the Timer 0 interrupt priority level. PT0=1 programs it to the higher priority level.
IP.0	PX0	Defines the External Interrupt 0 priority level. PX0=1 programs it to the higher priority level.

El manejo de las Interrupciones en el 8051

Si dos peticiones de diferente nivel de prioridad son recibidas simultáneamente, la solicitud de más alta prioridad será atendida.

Si solicitudes de la misma prioridad son recibidas simultáneamente una secuencia de sondeo interno determinará que solicitud será atendida.

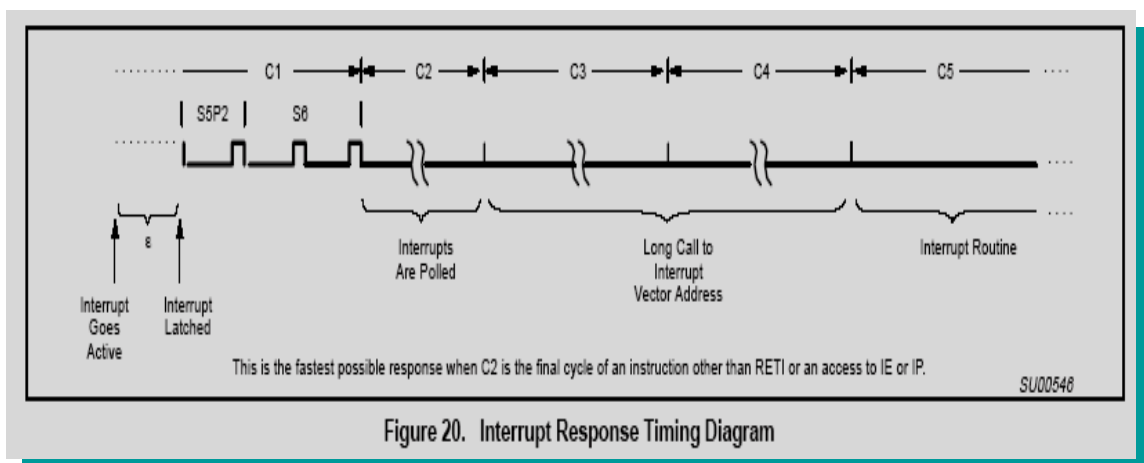
Por lo tanto, dentro de cada nivel de prioridad hay una segunda estructura de prioridades determinado por una la secuencia de sondeo como sigue: :

Fuente	Nivel de prioridad
1. IE0	(más alta)
2. TF0	
3. IE1	
4. TF1	
5. RI+TI	(más baja)

Carlos E. Canto Quintal

El manejo de las Interrupciones en el 8051

Como se manipulan las interrupciones



The interrupt flags are sampled at S5P2 of every machine cycle . The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine,

Carlos E. Canto Quintal

hardware-generated LCALL if is not blocked by any of the following conditions:

1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine.

Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine.

Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least one more instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note that if an interrupt flag is active but not being responded to for one of the above conditions, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered.

Every polling cycle is new.

Carlos E. Canto Quintal

¿Qué ocurre cuando se produce una interrupción?

Todas las interrupciones en el 8051 son enmascarables. Esto quiere decir que cuando alguna de las fuentes de interrupción realiza una petición el proceso que se desencadena es el siguiente:

1. Se comprueba si la fuente de interrupción está habilitada y las interrupciones del microcontrolador permitidas.
2. Se finaliza la ejecución de la instrucción en curso.
3. Se almacena en la pila el contador de programa actual y se modifica el registro de habilitación de las interrupciones.
4. Se salta a una dirección fija de memoria que depende de la fuente de interrupción.
5. Al finalizar la rutina de interrupción se ejecuta la instrucción de retorno de interrupción (RETI) que recupera el contador de programa y el valor del registro de habilitación de interrupciones.

Carlos E. Canto Quintal

¿Qué ocurre cuando se produce una interrupción?

<i>Fuente de interrupción</i>	<i>Dirección de salto</i>
Interrupción externa 0	0003H
Timer 0 flag	000BH
Interrupción externa 1	0013H
Timer 1 flag	001BH
Puerto serie	0023H

Carlos E. Canto Quintal

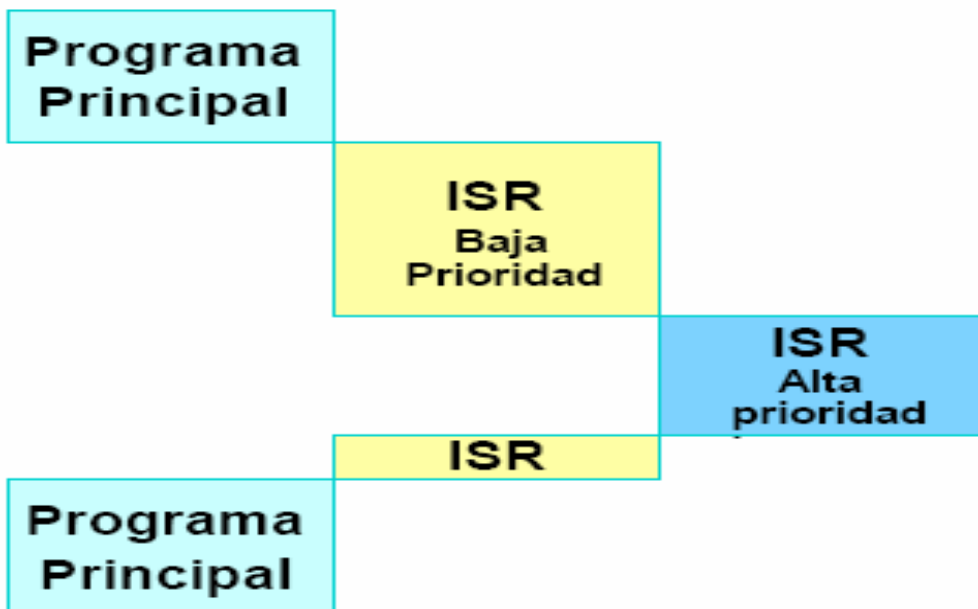
¿Qué ocurre si se produce una interrupción mientras se está atendiendo otra?

Lo que indican las prioridades es que si cuando se está atendiendo una interrupción menos prioritaria se produce una más prioritaria, se deja de atender la menos prioritaria, se atiende la otra y al finalizar ésta se retorna a la menos prioritaria. O dicho de otro modo, las interrupciones de la línea más prioritaria se anidan dentro de las de menos prioridad.

Gráficamente: La pregunta que surge es ¿ocurre lo mismo con la prioridad de las interrupciones dentro de una misma línea de prioridad? o dicho de otro modo, si dos interrupciones están en una misma línea de prioridad ¿puede una interrumpir a la otra si se encuentra antes en la cadena de Daisy Chain?. La respuesta es que dentro de una misma línea de prioridad no pueden anidarse interrupciones aunque estén antes en la cadena. Inmediatamente hay que plantearse ¿entonces para qué vale esa cadena?. La prioridad dentro de la cadena sirve sólo en el caso en el que haya dos interrupciones de una misma prioridad (alta o baja) pendientes de ser procesadas en un momento dado. En este caso es cuando se tienen en cuenta las prioridades dentro de la cadena. Por tanto, y a modo de resumen, podemos decir que este microcontrolador permite el anidamiento de interrupciones entre líneas de prioridad pero no entre interrupciones de una misma prioridad.

Esta forma de gestionar las prioridades condiciona el modo en el que hay que programar las rutinas de atención a las interrupciones. Si se generan rutinas muy largas puede ocurrir que mientras se estén ejecutando se produzcan interrupciones más prioritarias que se queden esperando a que la otra finalice.

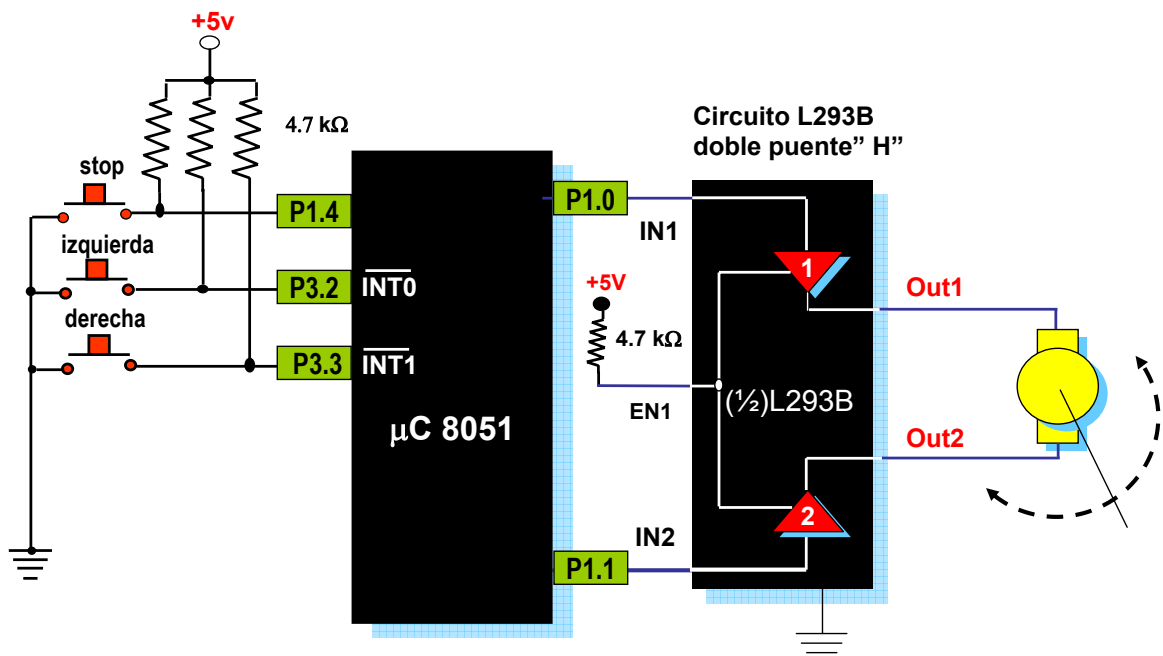
Carlos E. Canto Quintal

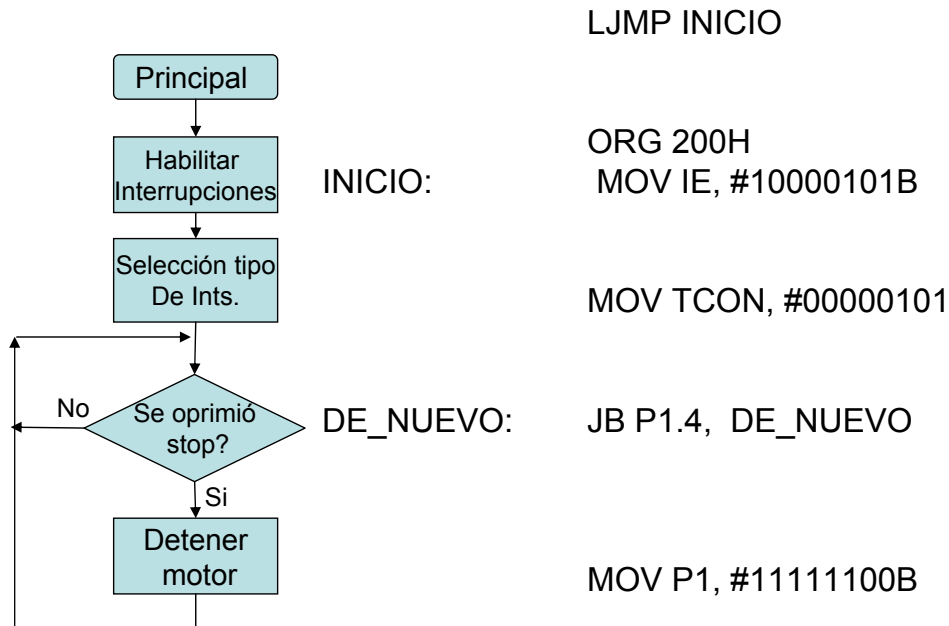
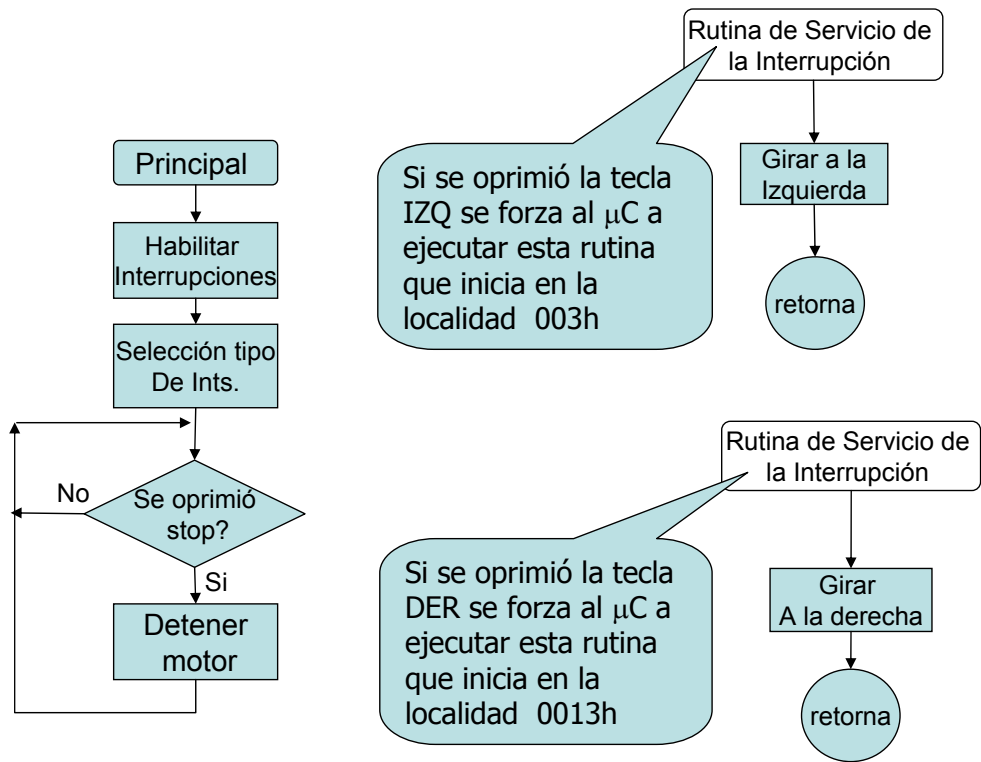


Carlos E. Canto Quintal

Ejemplo: control de sentido de giro de un motor de CD

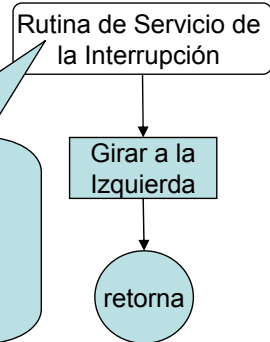
Implemente un programa para hacer que un motor de CD gire en el sentido de la tecla correspondiente oprimida y se de tenga al oprimir el botón stop (usando Interrupciones)





```
ORG 003  
MOV P1,#11111101B  
RETI
```

Si se oprimió la tecla
IZQ se forza al μ C a
ejecutar esta rutina
que inicia en la
localidad 003h



```
ORG 013H  
MOV P1,#11111110B  
RETI
```

Si se oprimió la tecla
DER se forza al μ C a
ejecutar esta rutina
que inicia en la
localidad 0013h

