

Trabajo de investigación

INTERRUPCIONES

Juan Esteba Lopez Dominguez

26 de junio de 2020

1. Desarrollo

“Los controladores surgieron junto con la primera computadora personal, ya que esta necesitaba interactuar con los periféricos, pero no de forma mecánica si no mediante un software provisto de instrucciones en un procesador

La primera técnica que se empleó fue que el propio procesador se encargaba de sondear el dispositivo cada cierto tiempo, para saber si tenía pendiente alguna comunicación para él.”¹.

“Una interrupción es un mecanismo que permite ejecutar un bloque de instrucciones interrumpiendo la ejecución de un programa, y luego reestablecer la ejecución del mismo sin afectarlo directamente. De este modo un programa puede ser interrumpido temporalmente para atender alguna necesidad urgente del computador y luego continuar su ejecución como si nada hubiera pasado ”²

Esto se puede asociar con la vida real, como por ejemplo cuando un empresario está en su oficina revisando datos estadísticos de rentabilidad y suena el teléfono de su escritorio, él atiende la llamada y una vez finaliza continúa con la revisión de datos, en este caso la revisión de datos se podría asociar con la ejecución de un programa y la llamada telefónica con una interrupción.

Las interrupciones generalmente se aplican para realizar tareas elementales y que son asíncronas en el proceso, en el caso de computación, responder al teclado, leer y escribir archivos, etc, y en el caso de los microcontroladores atender llamados de timer, atender interrupciones de usuario, etc.

Por ejemplo, al presionar las teclas Ctrl+Alt+Supr se genera una interrupción en el sistema operativo Windows que ofrece al usuario opciones de administrador de tareas y cambios de usuario.

Atendiendo a la fuente que las produce, las interrupciones pueden clasificarse de la siguiente manera:

- ”.Interrupciones de hardware. Estas son asíncronas a la ejecución del procesador, es decir, se pueden producir en cualquier momento independientemente de lo que esté haciendo el CPU en ese momento. Las causas que las produce son externas al procesador y a menudo suelen estar ligadas con los distintos dispositivos de entrada y de salida.

¹<https://issuu.com/kevinaguilar13/docs/interrupciones.docx>

²<http://logica-reptilia.blogspot.com/2009/03/interrupciones.html>

- Excepciones. Son aquellas que se producen de forma asíncrona a la ejecución del procesador y por lo tanto podrían producirse si se analiza con detenimiento la traza del programa que en ese momento estaba siendo ejecutada en la CPU. Normalmente son causadas al realizare operaciones no permitidas tales como la división entre 0 el desbordamiento, el acceso a una posición de memoria no permitida, etc,
- Interrupciones por software. Las interrupciones por software son generadas por un programa en ejecución. Para generarlas, existen distintas instrucciones en el código de maquina que permiten al programador producir una interrupción.”

³.

”

-Una lista de las interrupciones generadas por hardware son las siguientes:

IRQ PRIORIDAD FUNCIÓN

- 0	1	Timer
- 1	2	Teclado Hardware
- 2		Reservada
- 3	11	COM 2
- 4	12	COM 1
- 5	13	Tarjeta de sonido
- 6	14	Controlador floppy
- 7	15	Puerto paralelo
- 8	3	Reloj
- 9	4	Tarjeta de red, sonido
- 10	5	Idem
- 11	6	Idem
- 12	7	PS-mouse
- 13	8	Co-procesador matemático
- 14	9	Canal IDE primario
- 15	10	Canal IDE secundario ” ⁴

2. ¿Por qué considera que es importante saber manejarlas?

Las interrupciones son el medio por el cual la maquina se comunica con el usuario a través de diferentes periféricos, es importante saber manejarlos, porque con estas se logra desarrollar una mejor experiencia de usuario y una correcta optimización de los recursos de la máquina.

Por ejemplo, en el caso de un microcontrolador se puede optimizar energía poniendo el microcontrolador a dormir y gracias a una interrupción que lo despierte, el microcontrolador pasara a modo de operación cuando sea requerido.

³`bluepower.law.fit{igraph}`

⁴<http://logica-reptilia.blogspot.com/2009/03/interrupciones.html>

3. Ejemplo de interrupción usando la plataforma Arduino

Este programa en Arduino maneja dos interrupciones de HW haciendo uso de los pines de interrupción del Arduino UNO (pines digitales 2 y 3) y una de SW haciendo uso del Timer1

Circuito:

- Se debe conectar dos pulsadores entre GND y los pines digitales 2 y 3.
- Se debe conectar un potenciómetro como divisor de voltaje, es decir, los pines extremos entre VCC y GND y el pin central se conecta a la entrada analógica A0

FUNCIONAMIENTO:

- El programa está diseñado para controlar la frecuencia de encendido y apagado del LED asociado al PIN 13
- El microcontrolador, inicia con un Periodo de 1 Segundo por defecto
- El usuario podrá cambiar ese periodo a un periodo fijo de 0.2 segundos, haciendo uso de la interrupción asociada al PIN2
- El usuario podrá cambiar el periodo a un periodo variable que toma valores dependiendo de la posición del potenciómetro, haciendo uso de la interrupción asociada al PIN3
- El usuario podrá ver en todo momento a través del monitor serial el Periodo actual de funcionamiento en segundos

4. EXPLICACIÓN:

El funcionamiento general del programa es totalmente por fuera del main (función loop), esto se logra gracias a las interrupciones, con lo cual se logra una altísima precisión al momento de trabajar con valores de tiempo, ya que es una interrupción de timer la que se encarga de actualizar los valores de tiempo.

También, las actualizaciones del valor del periodo son prácticamente instantáneas ya que se hacen por medio de interrupciones.

La importancia de este tipo de interrupciones es mas notoria cuando hay un código muy robusto dentro de la función principal (en este caso es un código muy simple, solo es para mostrar un valor a través del monitor serial), pero supongamos que por ejemplo en el caso de una maquina que tenga un sensor de final de carrera (una señal que se puede conectar a los pines 2 y 3), en este caso, como se trata de un final de carrera, es una señal que indica al microcontrolador que se debe apagar inmediatamente el motor, o cambiar el sentido de giro, si se hace por interrupciones se logra el objetivo porque el cambio es prácticamente instantáneo, pero si se hace a través de código, se pierde tiempo valioso en la ejecución de las líneas anteriores a esta condición dentro del programa.