

# Las interrupciones en los microprocesadores

Camilo Alvarez Muñoz  
camilo.alvarezm@udea.edu.co

24 de junio de 2020

Conforme ha avanzado la tecnología, la humanidad ha sido capaz de resolver tareas que antes requerían de mucho esfuerzo y tiempo, pero ahora, podemos hacer estas tareas de una forma muy sencilla con la ayuda de los avances tecnológicos. Uno de estos avances son los microprocesadores, pequeños dispositivos que nos permiten realizar una infinidad de operaciones lógicas.

Con el tiempo poco a poco se han ido mejorando los microprocesadores y los microcontroladores, y entonces se vio la necesidad de introducir un concepto importante, las interrupciones, este concepto es necesario para que el microprocesador o microcontrolador pueda interactuar correctamente con el exterior sin perder su eficiencia y aprovechando su máximo potencial “Una interrupción consiste en un mecanismo por el cual un evento interno o externo puede interrumpir la ejecución de un programa en cualquier momento. A partir de entonces se produce automáticamente un salto a una subrutina de atención a la interrupción, ésta atiende inmediatamente el evento y retoma luego la ejecución del programa exactamente donde estaba en el momento de ser interrumpido, continuando su tarea justo donde la dejó”[Apaza Condori, 2017]. Este concepto, en el caso de los microprocesadores, significa que cuando los periféricos (mouse, teclado, pantalla, entre otros) requieren mandar algún tipo de información o realizar una operación en el microprocesador, esta interrupción se puede atender en cualquier momento de una forma eficiente y que al momento de terminar el evento, se retome las instrucciones que antes se estaban ejecutando; también pueden haber algunos casos específicos en el que las instrucciones vengan del interior del procesador, estos serán atendidos de la misma forma.

El concepto de interrupciones es muy importante y puede ser ilustrado por muchos ejemplos de nuestra vida cotidiana, como cuando somos distraídos de lo que estamos haciendo por una llamada telefónica o mensaje y debemos responder para poder retomar lo que estábamos haciendo, o cuando necesitamos recibir algún paquete o abrirle la puerta a alguna persona, los casos son muchos, el punto es que tanto nosotros como los microprocesadores o los microcontroladores estamos expuestos a situaciones exteriores a nuestra forma de operar básica que debemos atender rápidamente.

Se puede hablar de que hubo una evolución “histórica” en las interrupciones, en el año 1971 cuando se crearon los primeros microprocesadores y microcontroladores fue introducido un método llamado “polling”, este método consistía en un mecanismo que tenía la instrucción de estar periódicamente monitoreando las entradas/salidas (Periféricos) del microcontrolador verificando si había algún evento que debía ser atendido, este método tuvo que ser reemplazado, ya que no era eficiente, se consumían mucho tiempo y recursos para que el microcontrolador estuviera siempre revisando si ocurría un

evento y no permitía al controlador poder enfocarse totalmente en otras tareas y mucha información se podía llegar a perder debido a los periodos de tiempo de monitoreo [Reyes Cortés and Cid Monjaraz, 2015].

Debido a la poca eficiencia del “Polling” se vio la necesidad de introducir un nuevo método más directo y eficiente, las interrupciones, el microcontrolador/microprocesador cuenta con un pequeño dispositivo llamado Unidad de Interrupciones, esta unidad es la que recibirá las peticiones de interrupción (IRQ) que indican que tiene que ocurrir una interrupción en el procesador, la unidad de interrupciones establece prioridad entre cual interrupción atender primero, cuando se recibe una petición de interrupción el procesador pausa las instrucciones que se estaban ejecutando, luego ejecuta una rutina de instrucciones previamente establecidas para la interrupción y luego retoma su proceso normal en donde lo había dejado previamente [Reyes Cortés and Cid Monjaraz, 2015].

Se puede hablar de dos tipos principales de interrupciones: las interrupciones por hardware y las interrupciones por software. Las que ocurren por hardware pueden ocurrir en cualquier momento, por esto también se les llama asincrónicas, están casi siempre relacionadas con los periféricos (mouse, teclado, etc), estos envían un cambio de voltaje o una señal (flanco de subida o de caída) a los pines de entrada/salida, cuando esta señal es detectada se habilitará una bandera de interrupción, la unidad de ejecución del procesador termina las instrucciones que se estaban realizando y las guarda, luego, el microprocesador maneja un vector de interrupciones en el que se guardan direcciones de memoria de las instrucciones de cada interrupción, al activarse la bandera de interrupción se buscará en el vector la subrutina adecuada y se ejecutará, luego se restaurará al estado en el que el microprocesador estaba antes de la interrupción y se restauran las banderas para así poder recibir otras interrupciones. Dependiendo del modelo del microprocesador o microcontrolador se manejan pines de entrada/salida distintos, posiciones de memorias distintas, instrucciones distintas, prioridades entre interrupciones distintas y diferentes límites de interrupciones admitidas.

Las interrupciones por software son aquellas que se generan dentro del sistema, pueden ser generadas por el propio procesador en casos específicos (excepciones), pueden ser generadas por los programas (por ejemplo se requiere que se lea alguna archivo o se ingrese un dato a una consola), y pueden ser generadas por el propio usuario; las interrupciones por software se atienden de forma muy similar que a las de hardware.

Las excepciones por lo general son algunos errores que impiden el desarrollo de una instrucción, algunos ejemplos son las divisiones entre cero, logaritmos de cero, raíces de números negativos, instrucciones o códigos incorrectos, desbordamiento de contadores y muchos otros.

El propio usuario puede generar interrupciones por software por ejemplo al detener un proceso en el administrador de tareas, o también, puede programar sus propias interrupciones usando diferente tipo de hardware o software, por ejemplo, en un Arduino UNO, el usuario puede generar sus propias subrutina de una interrupción (ISR) con la función `attachInterrupt(num-int, ISR, mode)` y con distintos parámetros que le pueden indicar al

dispositivo que la interrupción solo sucede si hay un cambio de voltaje o un flanco de bajada de una señal [Reyes Cortés and Cid Monjaraz, 2015].

El software y hardware usado para las interrupciones afecta en muchos factores, como las diferentes tipos de interrupciones administradas en el sistema, las subrutinas, la velocidad de procesamiento, la prioridades entre interrupciones y la cantidad de interrupciones admitidas.

## Referencias

- [Apaza Condori, 2017] Apaza Condori, D. (2017). Microcontroladores pic fundamentos y aplicaciones un enfoque didáctico.
- [Reyes Cortés and Cid Monjaraz, 2015] Reyes Cortés, F. and Cid Monjaraz, J. (2015). Arduino: aplicaciones en robótica, mecatrónica e ingenierías. *Mexico, Alfaomega, Barcelona, Marcombo*.