Martes 6 Octubre

Tema: Aplicaciones orientadas a Eventos

El concepto de **Interrupciones** en el sistema de Arduino

🡪 Fuentes de interrupción posibles

--- Internas: Temporizadores; WDT; BrownOut; Problemas con el Oscilador; Comunicaciones (IIC, SPI, SCL); Otras fuentes)

--- Externas: Cambios de nivel en líneas de entrada; Otras fuentes

🡪 ¿Por qué es importante el manejo de las interrupciones? Permite realizar acciones en consecuencia de la ocurrencia de eventos (externos o internos)

Algunos elementos externos que pueden emplearse para ocasionar interrupciones que *se pueden atender*.

Las condiciones que puede haber para que se originen las interrupciones están:

--- Cuando el pin está en LOW

--- Cuando el pin cambia o conmuta de estado, CHANGE

--- FALLING, en un flanco de bajada, cambio de “1” a “0”

--- RISING, en un flanco de subida, cambio de “0” a “1”

A la función que se habilita y “dispara” para atender a un interrupción, frecuentemente, se le denomina “*call back*”

Las señales (o condiciones) que generan interrupciones ocurren constantemente, entonces, ¿*cómo se habilita la atención a estas interrupciones*? Se utiliza la siguiente función:

**attachInterrupt(interrupt, ISR, mode)**

**Interrupt** es el número asociado con la Interrupción

**ISR** es la función “call back” que atiende la ocurrencia de la interrupción

**mode** es alguna de las anteriores condiciones

Por ejemplo:

**attachInterrupt(0, ISR\_Tecla, RISING);**

Se supone que se conectó un interruptor mecánico a la terminal 2 de la tarjeta. Cuando suba el nivel eléctrico en esa terminal de “0” a “1” se originará una interrupción que será atendida en la función **ISR\_Tecla.**

| MODELO ARDUINO | INT 0 | INT 1 | INT 2 | INT 3 | INT 4 | INT 5 |
| --- | --- | --- | --- | --- | --- | --- |
| UNO | Pin 2 | Pin 3 |  |  |  |  |
| MEGA | 2 | 3 | 21 | 20 | 19 | 18 |
| DUE | Todos | | | | | |
| Leonardo | 3 | 2 | 0 | 1 | 7 |  |

Cosas importantes para considerar en las rutinas que atienden interrupciones (“call backs”)

. Deben ser lo más cortas y limpias posibles, no deben requerir funciones o rutinas externas.

. Deben consumir la menor cantidad de tiempo posible.

. No deben incluir funciones de retardo (milis(), delay(), por ejemplo).

. No deben depender de otras interrupciones.

. Nunca incluir funciones de comunicación, menos serial().

. Una ISR o función CallBack no puede devolver parámetros, tampoco recibirlos.

Tratamiento especial para variables que existirán en funciones ISR, de las cuales se desea preservar su valor a lo largo de la existencia del programa, pero que la ocurrencia de las interrupciones pueda variar.

🡪 Empleo del calificador ***volatile*** en las variables.

En la sección de Referencia del Lenguaje (<https://www.arduino.cc/reference/en/>) se encuentran dos secciones importantes **attachInterrupt()** (<https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>) y **detachInterrupt()** (<https://www.arduino.cc/reference/en/language/functions/external-interrupts/detachinterrupt/>)

------------------------------------------------------------------------------------------------------------------------------------------------------------------

Hacer una aplicación para Arduino, simulada en Proteus, empleando interrupciones, que realice lo siguiente:

1. Genere un letrero en la terminal del canal serial asíncrono (Tx, Rx, a 57.600 bauds) que especifique cuando ocurre la interrupción.

2. Utilice tres LEDs, uno rojo, uno verde, uno amarillo, asociado con distintas fuentes de interrupción, en donde cada uno de ellos se enciende en consecuencia de que se pulse una señal eléctrica que cambia de “1” a “0” en alguna terminal de algún puerto, cada entrada estará asociada con un LED en particular.

3. Construir una aplicación que atienda la llegada de comunicaciones por el canal serial en función de las interrupciones que ocasione el arribo de datos por el canal serial. Cada vez que llegue información por el canal serial, que se invoque una función que procese lo que llegó.

4. Implementar un *timer* de tiempo real (minutos y segundos) empleando el concepto de interrupciones (Revisar ejercicios anteriores)

5.

------------------------------------------------------------------------------------------------------------------------------------------------------------------

Revisión de estado de las cuentas GitHub de cada alumna(o) ¿Ya está subido el material?

- Dudas

- Comentarios

- Uso de la versión de escrito de GitHub. Instalación. Copia de un sitio externo. Edición del sitio local. Actualización del sitio remoto,

------------------------------------------------------------------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------------------------------