"MANEJO DE INTERRUPCIONES EXTERNAS EN ARDUINO"

Informe N°5
Laboratorio de Sistemas Embebidos

Melanny Dávila
Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
melanny.davila@epn.edu.ec

2nd Jonathan Álvarez
Ingeniería en Telecomunicaciones
Facultad de Elétrica y Eléctronica
Quito, Ecuador
jonathan.alvarez@epn.edu.ec

***Abstract*—Las interrupciones son importantes debido a que permiten ejecutar segmentos de codigo al recibir la señal correspondiente sin esperar a que se acabe un bloque de ejecución. En el siguiente informe se hablará de algunas aplicaciones en donde este comportamiento es necesario, y para finalizar se presentarán dos ejercicios en los que se pone en practica las interrupciones estudiadas en el laboratorio.**

***Index Terms*—Arduino, interrupción, subrutina, función, estructura, bloque de código.**

I. Introducción

En muchas aplicaciones prácticas durante su ejecución es importante que estos respondan a una señal que indique a la placa de un evento que tiene que ser atendido de forma inmediata y por lo tanto los bloques de codigo que encontraban ejecutandose deberán esperar a otros que corresponden a la función de la interrupción. El uso de las interrupciones se relaciona a los pines que permiten implementarla, y dependen del modelo.

II. Objetivos

- Familiarizar al estudiante a través de un ejercicio de ejemplo con el uso y manejo de interrupciones externas la plataforma Arduino.
- Conocer las fuentes de interrupciones en la plataforma Arduino, hardware asociado, vectores de interrupción y aplicaciones [1].

III. Cuestionario

A. Resaltar la importancia del uso de las interrupciones externas. (100 palabras min)

El uso de las interrupciones externas en Arduino resulta de gran importancia ya que estas permiten ante un evento externo como la presión (cabe mencionar que existen diferentes modos para considerar en el manejo de interrupciones externas) de un botón como ejemplos más práctico, ejecutar una subrutina denominada ISR, esta debe de ser corta, interrumpiendo el flujo normal de ejecución del programa. Esta interrupción permite que eventos con prioridad sean atendidos por el microcontrolador de la placa Arduino, reflejándose en un sistema eficiente y dinámico que permita interactuar con el usuario gracias a las interrupciones externas, ya que no se presentarían retrasos en la atención del evento generado por el usuario.

B. Describir detalladamente 5 aplicaciones del uso de Interrupciones Externas e Internas.

1) **Apertura de una barrera usando interrupciones externas**
   Haciendo uso de un servo motor, que sea modificado para este gire de 0 a 90 grados, se puede realizar un sistema que presionando un botón accione una ISR que permita abrir dicha compuerta durante un periodo de tiempo determinado.
   Para visualización del usuario se puede usar leds de diferentes colores para identificar si se está realizando la apertura de la barrera o no. Para que se visualice el tiempo que permanece abierta la barrera se puede hacer uso de displays. Este sistema puede ser aplicado para abrir bombas de agua, apertura de las puertas de un garaje, etc.[1]
2) **Cambio del nivel de velocidad en un sistema de luces.**
   Haciendo uso de interrupciones, se puede cambiar la velocidad en que un flujo de luces es encendido, es decir si las luces se encienden en un patrón definido a velocidad muy lenta, haciendo uso de botones que suban la velocidad en que estas luces son encendidas, o caso contrario haciendo uso de otro botón se puede aumentar la velocidad en que las luces son encendidas. Para esta aplicación se hará uso de distintos leds y una placa Arduino.[2]
3) **Sistema de seguridad**
   Sistema de alarma basado en Arduino. Con dos sensores uno de movimiento y otro de luz se detecta la alarma. Después de detectar alarma, se entra en un estado de pre-alarma que si no se introduce la clave correcta en 10 segundos, paso a un estado de alarma.
   Para entrar en estado de alarma hay dos condiciones:
   - Detectar un movimiento (activar el sensor tilt)

- Detectar más de 5 segundos una iluminación superior a 900.

Para salir del estado de pre-alarma se debe introducir la clave correcta por teclado en los 10 segundos siguientes a la detección de alarma. Para salir del estado de alarma no debe haber ninguna de las dos condiciones de alarma y se debe pulsar el botón. En estado normal led y zumbador apagados.

4) **Botón de pánico**

La idea es generar un dispositivo capaz de emitir una señal silenciosa, que alerta a la central de seguridad de la empresa contratada sobre situaciones irregulares que estén ocurriendo en el hogar. Para realizer este Sistema se necesitará conectar los materiales de la siguiente forma a la placa del arduino UNO, use una protoboard más pequeña para ahorrar espacio.

Los cables (rojo y azul) colocados en el botón deben ser lo suficientemente largos, por lo menos 30 o 40 cm. Las fuentes de poder externas deben estar conectadas a la entrada VIN del Arduino. Se debe usar Jumpers (hembra-macho) para conectar el módulo de bluetooth a la placa. Las entradas TXD y RXD del módulo de bluetooth deben estar conectadas al Arduino en sus contrarios, es decir, TXD (Bluetooth) con RXD(Arduino) , RXD(Bluetooth) con TXD(Arduino); de esta manera el Arduino tendrá comunicación con el modulo. Basicamente se estará sensando hasta el boton sea aplastado, en ese momento se disparara la interrupción, se conectará al modulo bluetoth y con un dispositivo externo se contactará a la policia.[4]

5) **Paso peatonal**

En la mayoría de pasos peatonales existen botones que permiten a las personas cambiar el estado para poder cruzar la calle o acelerar el cambio, por su conveniencia y simplicidad es de gran implementación. Puesto que el botón en si causa una interrupción al flujo normal que experimenta el semáforo dependiente del tiempo, el pulsador también causará que se reinicien los tiempos en el semáforo debido a la interrupción.

C. Simular en Tinkercad dos aplicaciones de Interrupciones externas, de las consultadas en el punto 6.2. Las aplicaciones deben ser de complejidad media.

Cambio del nivel de velocidad en un sistema de luces

```
// Practica 5
//Codigo par acontrolar la rapidez con la que
//cambian de color una serie de LEDs
int vel = 300; // Velocidad al inicio del
    programa
int i;
void setup(){
  for(i = 4; i<11; i++){
    pinMode(i,OUTPUT);
  }
  //Interrupciones
  attachInterrupt(digitalPinToInterrupt(2),
    aumentar, RISING);
  attachInterrupt(digitalPinToInterrupt(3),
    disminuir, RISING);
}

// Funciones de las interrupciones
// Las funciones aumentan o disminuyen
    dependiendo del boton
void aumentar(){
  vel += 100;
}
void disminuir(){
  vel -= 100;
  //Asegura que la velocidad no sea cero o menor
  if(vel < 140){
    vel = 100;
  }
}
void loop()
{
  //Secuencia de encendido de leds
  digitalWrite(4, HIGH);
  digitalWrite(10, LOW);
  delay(vel);
  digitalWrite(5, HIGH);
  digitalWrite(4, LOW);
  delay(vel);
  digitalWrite(6, HIGH);
  digitalWrite(5, LOW);
  delay(vel);
  digitalWrite(7, HIGH);
  digitalWrite(6, LOW);
  delay(vel);
  digitalWrite(8, HIGH);
  digitalWrite(7, LOW);
  delay(vel);
  digitalWrite(9, HIGH);
  digitalWrite(8, LOW);
  delay(vel);
  digitalWrite(10, HIGH);
  digitalWrite(9, LOW);
  delay(vel);
}
```

Script 1. Código control de luces con interrupciones

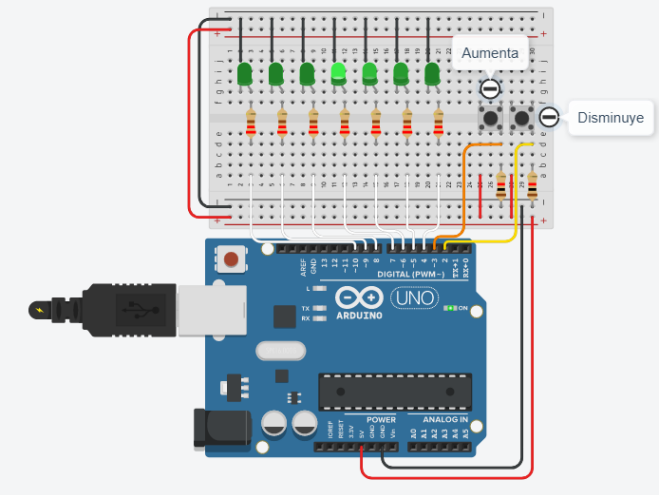

Fig. 1. Esquemático del circuito de cambio de velocidad leds con interrupciones.

Sistema de paso peatonal

```
// Practica 5
// Aplicacion semaforo peatonal con interrupcion
```

```
// Se definen los pines de salida y las
    variables
// para el control
int verde = 4;
int rojo = 5;
int t;
int estado = 1;
volatile bool semaf = false;
bool encend = false;
void setup()
{
  pinMode(verde, OUTPUT);
  pinMode(rojo, OUTPUT);
  // Condiciones iniciales
  digitalWrite(verde, LOW);
  digitalWrite(rojo, LOW);
  t = millis();
  attachInterrupt(digitalPinToInterrupt(3), paso
    , RISING);
}
// Funcion de control de la interrupcion
void paso(){
  semaf = true;
}

void loop()
{
  // Estados logicos para el semaforo
  if(estado==1){
    if((millis()- t)<2000){
      //Simula semaforo en rojo
      digitalWrite(rojo, HIGH);
      digitalWrite(verde, LOW);
    }
    else{
      estado=2;
      t = millis();
    }
  }

  if(estado==2){
    if((millis()- t)<2000){
      //Simula semaforo en verde
      digitalWrite(rojo, LOW);
      digitalWrite(verde, HIGH);
    }
    else{
      estado=1;
      t = millis();
      if(encend)
      encend= false;
    }
  }
  if(semaf && !encend){
    //Reestablece la variable de interrupcion
    semaf = false;
    if(estado==2){
      encend=true;
      //Aumenta el tiempo en verde
      t -= 1000;
    }
  }
}
```

Script 2. Código de paso peatonal con interrupciones

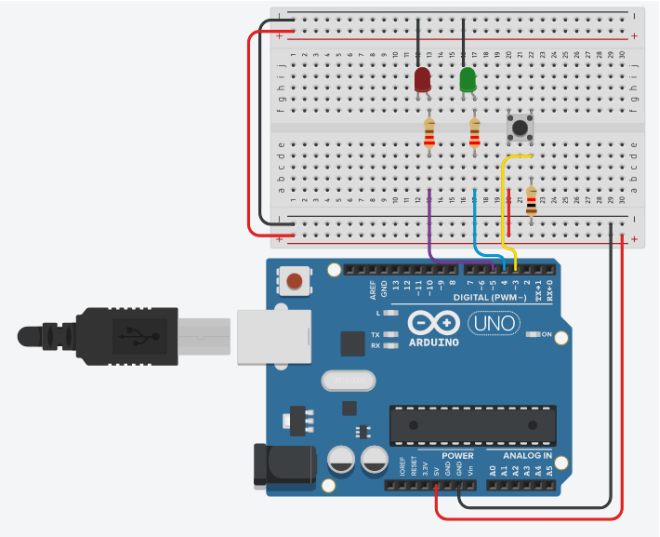

Fig. 2. Sistema de paso peatonal con boton de interrupción

D. *Conclusiones:*

Jonathan Álvarez

- La interrupción reset nos permite reiniciar nuestro programa ante situaciones en el que no podría arreglarse por si solo.
- Las interrupciones vuelven a los programas más dinámicos y proporcionan proveen de la función necesaria a señales que sean de gran importancia.
- Existen más interrupciones que las provistas por las señales en los pines correspondientes, algunas se pueden activar mediante software y pueden ser útiles para controlar problemas lógicos en nuestros programas.

Melanny Dávila

- Las interrupciones externas permiten controlar problemas en ”tiempo real” ya que permiten ejecutar código en cuanto se produce determinada acción
- Se concluye que las estructuras de control de condicionales permiten dotar al programa (sketch) de capacidad de ejecutar diferentes códigos distinguiendo diferentes casos presentes durante la ejecución del programa.
- Mediante esta sesión de laboratorio, se comprendió la importancia del uso de interrupciones debido a que gracias a ellas se puede ejecutar diferentes partes de código y de esta manera ahorrar recursos de procesamiento y energía de Arduino.

E. *Recomendaciones:*

Jonathan Álvarez

- Verificar que la subrutina que acompaña a la interrupción no sea de gran longitud y su implementación sea rapida.
- Definir el mejor tipo de estado que pasará el pin de la interrupción dependiendo del programa. Aunque una puede ser conveniente utilizar RISING.

Melanny Dávila

- Analizar el sistema a diseñar para poder definir si este requiere de interrupciones para un correcto funcionamiento.

- Es importante conocer los pines que serán utilizados para la implementación de las interrupciones.

REFERENCES

[1] E. Tatayo, “MANEJO DE INTERRUPCIONES EXTERNAS EN ARDUINO”. C.P. SISTEMAS EMBEBIDOS, Accedido: dic. 23, 2020. [En línea].

[2] ”Circuit design W4: barrier, 2 lights and CountDown activated by an interrupt”, 2020. [Online]. Disponible en: https://www.tinkercad.com/things/9qLZlKQRdJw-w4-barrier-2-lights-and-countdown-activated-by-an-interrupt-. [Accedido: 12- Enero- 2021].

[3] ”encendido de leds aleatorios y apagado por boton”, Forum.arduino.cc, 2020. [Online]. Disponible en: https://forum.arduino.cc/index.php?topic=405833.0. [Accedido: 12- Enero- 2021].

[4] ”Proyecto – Sistema de Alarma”, Aprendiendo Arduino, 2020. [Online]. Disponible en: https://aprendiendoarduino.wordpress.com/2017/06/29/proyecto-sistema-de-alarma/. [Accessed: 12- Enero- 2021].

[5] ”PaniK Button Prototype (español)”, Instructables, 2020. [Online]. Disponible en: https://www.instructables.com/id/PaniK-Button-Prototype-Espa%C3%B1ol/. [Accedido: 12- Enero- 2021].

[6] ”Timer with alarm using keypad : arduino”, Reddit.com, 2020. [Online]. Disponible en: https://www.reddit.com/r/arduino/comments/31sby9/timer_with_alarm_using_keypad/. [Accedido: 12- Enero- 2021].