



Universidad Veracruzana

# UNIVERSIDAD VERACRUZANA

FACULTAD DE INSTRUMENTACIÓN  
ELECTRÓNICA



FACULTAD  
DE  
INSTRUMENTACIÓN  
ELECTRÓNICA

1-11-2020

## TÓPICOS AVANZADOS DE INSTRUMENTACIÓN ELECTRÓNICA I (SISTEMAS EMBEBIDOS)

### **Tarea 5. – Interrupciones**

Alumnos:

Rangel Pulido Julio

Martínez González Sergio David

Ramírez Mendoza Jesús Emiliano

Catedrático:

**Hernández Reyes Sergio Machuca**

# ÍNDICE

1.	ACTIVIDAD .....	3
2.	DESCRIPCIÓN .....	3
3.	LISTA DE COMPONENTES .....	6
4.	DIAGRAMA.....	7
5.	CONCLUSIONES .....	8
6.	BIBLIOGRAFÍA .....	9

## 1. ACTIVIDAD

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

## 2. DESCRIPCIÓN

### Temporizador con Arduino:

Para esta tarea se requiere realizar un timer con Arduino, en el que sean empleadas las interrupciones.

Para esto, se realizará un temporizador configurable, el cual consistirá con 4 interrupciones que serán activadas por un push button por cada interrupción. Estos mismos crearán las ordenes siguientes:

- Botón 1 (Minutos): Aumenta los minutos al que se quiera configurar el temporizador.
- Botón 2 (Segundos): Aumenta los segundos al que se quiera configurar el temporizador.
- Botón 3 (Inicio/Pausa): Inicia el conteo regresivo del temporizador, al igual que lo pausará.
- Botón 4 (Reiniciar): Realizará el reinicio del conteo del temporizador, por lo cual, la cuenta regresa a cero.

Gracias a que primero se visualiza cómo se quiere realizar el programa es más sencillo plasmarlo en el código, este código se presentará más adelante.

### Líneas del código.

Temporizador

```
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x20, 16, 2);
4
5 int s=0; //Se definen variables
6 int m=0;
7 int inicio = 0;
8 volatile int st=0;
9 int Ast = 0;
10
11 void setup()
12 {
13   lcd.begin(16,2);
14   lcd.backlight ();
15
16   attachInterrupt(digitalPinToInterrupt(2),ISR_Min,RISING);// Definimos los pines de interrupcion
17   attachInterrupt(digitalPinToInterrupt(3),ISR_S,RISING);
18   attachInterrupt(digitalPinToInterrupt(18),INICIA,RISING);
19   attachInterrupt(digitalPinToInterrupt(19),REINICIA,RISING);
20   pinMode(1,OUTPUT);
21   lcd.setCursor(0,0); //Se configura el cartel de inicio
22   lcd.print("  Temporizador");
23   lcd.setCursor(0,1);
24   lcd.print("    Equipo 6");
25   delay(1000); //Despues de un tiempo predeterminado, el cartel cambia
```

```

26
27 lcd.clear();
28 lcd.setCursor(0,0);
29 lcd.print(" Elegir tiempo ");
30 lcd.setCursor(5,1);
31 lcd.print("00:00");
32
33 }
34
35 void loop()
36 {
37
38 while((inicio%2) == 1) //Se inicia el ciclo, con esto, hacemos que
39 {                               //la cuenta regresiva inicie hasta que se presionado el botón de Inicio
40     if(st>0){
41         delay(500);
42         st--;
43         m = (st / 60) % 60; //Convertimos los segundos totales en minutos
44         s = st % 60;        //Convertimos los segundos totales en periodos de 60 segundos
45
46         lcd.setCursor(0,0);
47         lcd.print("   Corriendo   ");
48
49         lcd.setCursor(5,1);
50
51         if(m<10)lcd.print("0");
52
53         lcd.print(m);
54         lcd.print(":");
55
56         if(s<10) lcd.print("0");
57         lcd.print(s);
58
59     }
60     if(st==0) { //Cuando acabe el tiempo se cambia el cartel y se enciende el LED
61         digitalWrite(1,HIGH);
62         lcd.setCursor(0,0);
63         lcd.print("               ");
64         lcd.setCursor(0,1); lcd.print("   Finalizado");
65         delay (3000);
66         lcd.clear();
67         inicio = 0;
68     }
69 }
70
71 while ((inicio%2) == 0){ //Se realiza la configuración para que pause o reinicie la cuenta.
72
73     if (Ast > 0){
74         st=0;
75         Ast=0;
76     }
77
78     digitalWrite(1,LOW);
79     delay(500);
80

```

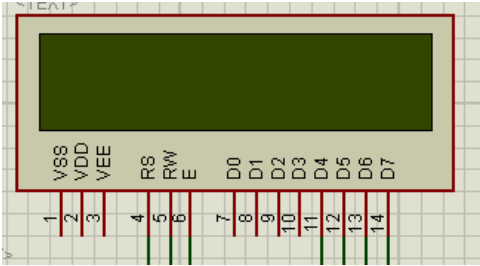
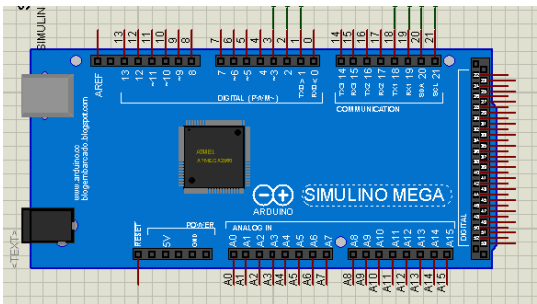
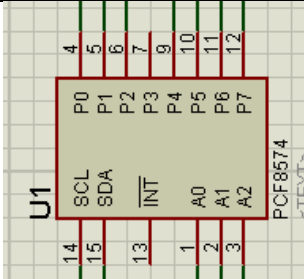
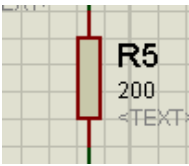
```



81     m = (st / 60) % 60; //Convertimos los segundos totales en minutos
82     s = st % 60;        //Convertimos los segundos totales en periodos de 60 segundos
83
84     lcd.setCursor(0,0);
85     lcd.print(" Elegir tiempo ");
86
87     lcd.setCursor(5,1);
88
89     //lcd.setCursor(0,1);
90
91     if(m<10)lcd.print("0");
92
93     lcd.print(m);
94     lcd.print(":");
95
96     if(s<10) lcd.print("0");
97     lcd.print(s);
98 }
99 }
100 void ISR_Min()
101 {
102     st=st+60;
103     delay(200);
104 }
105
106 void ISR_S()
107 {
108     st=st+1;
109
110     delay(200);
111 }
112 void INICIA()
113 {
114     inicio = inicio + 1;
115 }
116 void REINICIA()
117 {
118     Ast++;
119     inicio = 0;
120 }

```

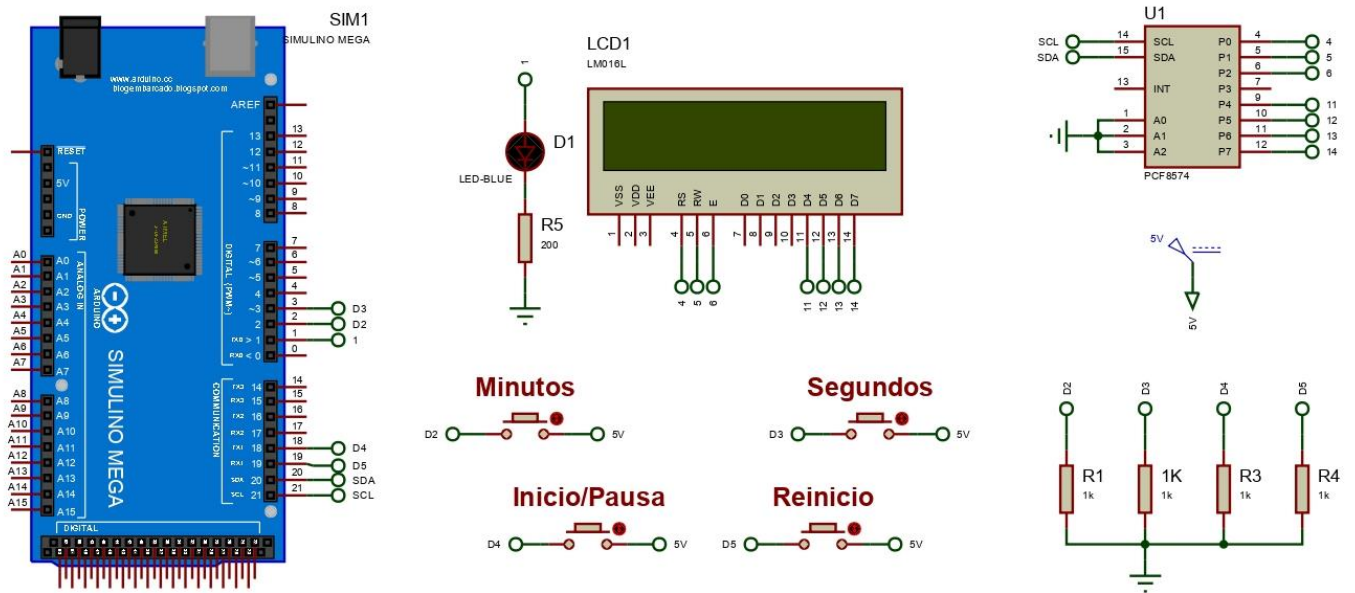
---

### 3. LISTA DE COMPONENTES

COMPONENTE	IMAGEN	DESCRIPCIÓN
LM016L (LCD 16x2)		El LCD (Liquid Crystal Display) o pantalla de cristal líquido, un componente empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres y símbolos.
Arduino Mega		Placa de 8 bits con 54 pines digitales, 16 entradas analógicas y 4 puertos seriales.  Utilizamos esta placa especialmente debido a que es capaz de realizar mas de 2 interrupciones (en los pines 2,3,21,20,19,18)
PCF8574		Expansor de entradas y salidas digitales por bus I2C
Resistencias		Componente electrónico diseñado para introducir una resistencia eléctrica determinada entre dos puntos de un circuito eléctrico.

LEDs		Fuente de luz constituida por un material semiconductor dotado de dos terminales.
Push Button		Dispositivo utilizado para realizar cierta función. Los botones son de diversas formas y tamaños y se encuentran en todo tipo de dispositivos, aunque principalmente en aparatos eléctricos y electrónicos.

#### 4. DIAGRAMA



## 5. CONCLUSIONES

El realizar esta práctica me hizo reflexionar de la forma sencilla en que se puede realizar un programa. Ya que al principio puede que se nos parezca difícil, mas por los detalles que se piden, pero una vez que vas desarrollando el programa, todo comienza a tomar sentido y es cuando lo vas realizando de forma fluida.

Las interrupciones son una gran herramienta en el uso de las programaciones, ya que son una forma sencilla de indicar alguna indicación específica que se quiera realizar.

Las interrupciones son un mecanismo muy potente y valioso en procesadores y autómatas. Arduino, por supuesto, no es una excepción.

Para entender la utilidad y necesidad de las interrupciones, fue necesario la consulta de varias fuentes que fueron de gran apoyo para implementarlas



## 6. BIBLIOGRAFÍA

- [1] L. Llamas. (2016, Abr 28). QUÉ SON Y CÓMO USAR INTERRUPCIONES EN ARDUINO [Online]. Disponible en: <https://www.luisllamas.es/que-son-y-como-usar-interrupciones-en-arduino/>
- [2] R. Rivas. (2017, May 3). Interrupciones [Online]. Disponible en: <https://aprendiendoarduino.wordpress.com/2016/11/13/interrupciones/>
- [3] 5 Hertz Electrónica (2018, Sep 10). Introducción a las LCD [Online] Disponible en: [https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial\\_id=9](https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=9)
- [4]