



Facultad de Instrumentación Electrónica y Ciencias Atmosféricas



UNIVERSIDAD VERACRUZANA

Experiencia educativa:

Tópicos avanzados de instrumentación electrónica I (Sistemas embebidos)

Académico:

Hernández Machuca Sergio Francisco

Alumnos:

Hernández Reyes Roberto Saul

(Reporte y código)

Sánchez López Luis Uriel

(Simulación y Compilación)

Velásquez Reyes Román Gabriel

(Reporte y código)

Contenido

4.1 ACTIVIDAD	3
4.2 DESCRIPCIÓN	3
4.3 LISTA DE COMPONENTES	5
4.4 DIAGRAMA	6
4.5 CONCLUSIONES //	6
4.6 BIBLIOGRAFÍA	7

4.1 ACTIVIDAD

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

4.2 DESCRIPCIÓN

Timer con Arduino:

Este código está pensado para trabajarse en plataformas Arduino Mega.

Como primera etapa se debe hacer notar que únicamente se utilizan 3 variables, minutos (m), Segundos (s) y una variable acumuladora de segundos (st). Cuando se declaran las variables ahora podemos notar que únicamente necesitamos dos interrupciones.

Las interrupciones son ISR_Min (para minutos) e ISR_Seg (para segundos), estas interrupciones se encargan de incrementar la variable “st”, si se llama a la primera $st=st+60$, mientras que si se llama a la segunda interrupción $st=st+2$, así aumentando el contador.

Líneas del código.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x20, 16, 2);

int s=0;
int m=0;

volatile int st=0;

void setup()
{
  lcd.begin(16,2);
  lcd.backlight ();

  attachInterrupt(digitalPinToInterrupt(2),ISR_Min,FALLING);// Definimos los pines de interrupcion
  attachInterrupt(digitalPinToInterrupt(3),ISR_S,FALLING);// We define the interrupt pins
  pinMode(1,OUTPUT);
  lcd.setCursor(0,0);
  lcd.print("Bienbenido");
  delay(1000);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Restan: 00:00");
}
```

```

void loop()
{

    while(st>0)
    {
        delay(800);
        st--;
        m = (st / 60) % 60; //Convertimos los segundos totales en minutos
        s = st % 60;        //Convertimos los segundos totales en periodos de 60 segundos

        lcd.setCursor(0,0); lcd.print("Restan: ");

        //lcd.setCursor(0,1);

        if(m<10)lcd.print("0");

        lcd.print(m);
        lcd.print(":");

        if(s<10) lcd.print("0");
        lcd.print(s);
        if(st==0) {
            digitalWrite(1,HIGH);
            lcd.setCursor(0,1); lcd.print("Fin del conteo");
        }
        }

        delay(5000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Restan: 00:00");
        digitalWrite(1,LOW);

    }
}

void ISR_Min()
{
    st=st+60;
    delay(200);
}

void ISR_S()
{
    st=st+2;
    delay(200);
}

```

4.3 LISTA DE COMPONENTES

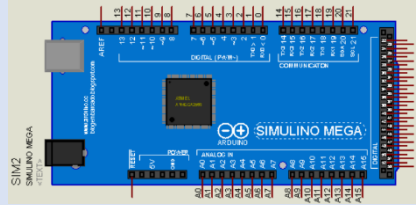

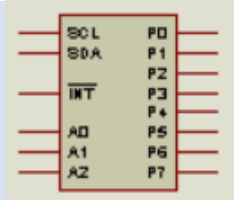
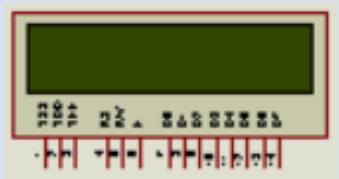
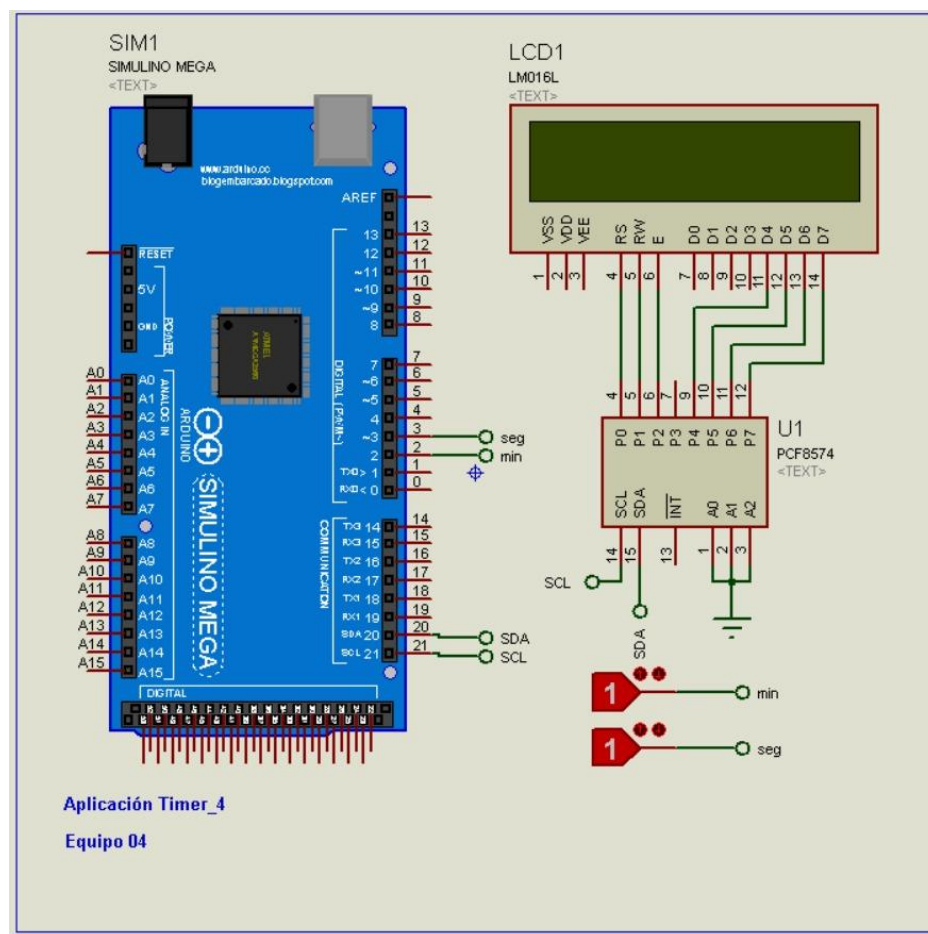
COMPONENTES	DESCRIPCIÓN	IMAGEN (Proteus)
Arduino Mega 2560	<p>Placa de 8 bits con 54 pines digitales, 16 entradas analógicas y 4 puertos seriales.</p> <p>Utilizamos esta placa especialmente debido a que es capaz de realizar mas de 2 interrupciones (en los pines 2,3,21,20,19,18)</p>	
Logicstate	Componente de visualización, para "inyectar" niveles lógicos en los pines, muestran mediante un recuadro azul o rojo si la señal digital en tal pin es "0" o "1" lógico respectivamente.	
PCF8574	Expansor de entradas y salidas digitales por bus I2C	
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.	

Tabla 4.1: LISTA DE COMPONENTES

4.4 DIAGRAMA



4.5 CONCLUSIONES

El uso de interrupciones permite utilizarlo este código sin la necesidad de un botón de “start” o comienzo, que se suele ver mucho en códigos de internet, ya que permiten realizar acciones en consecuencia de eventos, ya sean externos o internos. En este caso las interrupciones se encargan de incrementar el tiempo, independientemente del valor en el que se encuentra el micro en ese momento.

Esta activación además depende de las condiciones que se necesiten atender, para nuestro caso utilizamos “FALLING”, es decir la interrupción se activa al cambiar de 1 a 0, un dato a tener en cuenta es que los pines de interrupción varían entre placas de la familia Arduino.

Como nota, la idea de implementar una función “delay” en las interrupciones nace de la complicación de la simulación de almacenar los datos de manera inmediata a la interrupción.

4.6 BIBLIOGRAFÍA

"Más pines digitales con Arduino y expansor E/S I2C PCF8574", Luis Llamas, 2020. [Online]. Available: <https://www.luisllamas.es/mas-pines-digitales-con-arduino-y-el-expansor-es-i2c-pcf8574/#:~:text=El%20PCF8574%20es%20un%20expansor,dispositivos%20empleando%20con%20menos%20pines.&text=La%20comunicaci%C3%B3n%20se%20realiza%20a,sencillo%20obtener%20los%20datos%20medidos>. [Accessed: 28- Oct- 2020].

"Arduino Mega 2560 Rev3 | Arduino Official Store", Store.arduino.cc, 2020. [Online]. Available: <https://store.arduino.cc/usa/mega-2560-r3>. [Accessed: 28- Oct- 2020].

"Componentes de audio", Arrow, 2020. [Online]. Available: <https://www.arrow.com/es-mx/categories/audio-components>. [Accessed: 30- Sep- 2020].

[7]V. perfil, "Lcd 16x2", Todoelectrodo.blogspot.com, 2020. [Online]. Available: <http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>. [Accessed: 28- Oct- 2020].