INFORME DE LABORATORIO: TEMPORIZADORES.

MARÍA JOSÉ GIL HERRERA

VALENTINA RODRÍGUEZ AGUIRRE

MELISSA TAMAYO SEPULVEDA

Informe de laboratorio

Profesor del módulo:

AUGUSTO CARMONA VALENCIA

UNIVERSIDAD EAFIT

ESCUELA DE CIENCIAS

MEDELLÍN (ANTIOQUIA)

2020

Contenido

[**Introducción** 2](#_Toc40565084)

[**1.** **Objetivos** 2](#_Toc40565085)

[**2.** **Procedimiento** 3](#_Toc40565086)

[**Conclusiones** 6](#_Toc40565087)

[**Referencias** 6](#_Toc40565088)

[Ilustración 1: Diagrama de estados. 3](#_Toc40565080)

[Ilustración 2: Montaje de la simulación completa. 4](#_Toc40565081)

[Ilustración 3: Montaje pantalla, teclado y LED. 5](#_Toc40565082)

[Ilustración 4: Montaje motor y pantalla. 5](#_Toc40565083)

# **Introducción**

Los temporizadores, o Timers, se usan para controlar ciertos periodos de tiempo y para llevar la cuenta de algunos acontecimientos que ocurran en el exterior. Empleando estos, se tiene un control total sobre el sistema embebido, ya que no es necesario usar elementos exteriores o interrupciones innecesarias de este sistema. Para la medida de tiempos se carga un registro con el valor adecuado, luego este valor se va incrementando o decrementando al ritmo de los impulsos de reloj, ciclos de instrucción o algún múltiplo hasta que se desborde y llegue a 0, en este momento se produce un aviso, que generalmente es en forma de interrupción [1].

# **Objetivos**

* Escribir una MEF indicando el control de un dispositivo con ingreso de información.
* Programar adecuadamente el uso de Timers en un sistema embebido tipo microcontrolador.
* Programar y simular el funcionamiento del dispositivo de manera virtual usando Proteus como sistema de prueba.
* Desarrollar un temporizador usando un sistema embebido tipo microcontrolador, controlando el tiempo mediante los temporizadores del dispositivo.
* Documentar el desarrollo y guardar la información en el *git* correspondiente.

# **Procedimiento**

En esta parte, se mostrará la MEF que se realizó para el control del dispositivo, las conexiones en el programa Proteus para su simulación, y se explicara el código que se implementó para llevar a cabo este laboratorio, usando Timers.

Primeramente, se realizó un diagrama de estados finitos, para ayudar en la programación y mostrando el control que tendrá el dispositivo. El dispositivo que se va a realizar es un temporizador, que junto a un motor DC y un LED, harán o simularán el papel de un horno microondas. A continuación, se mostrará el diagrama de estados finitos que se realizó para el artefacto.

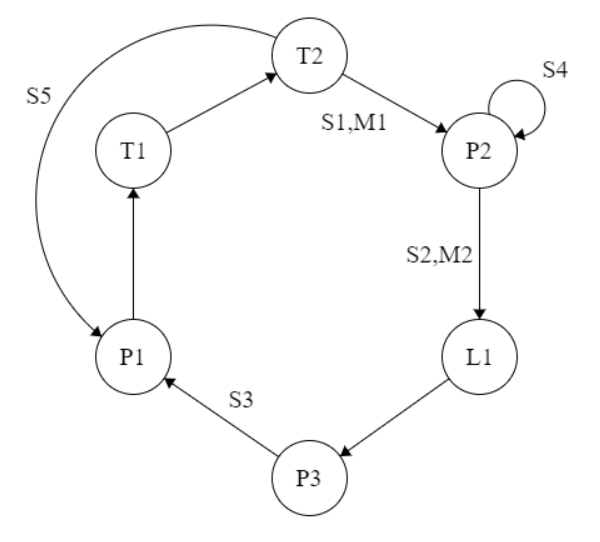


Ilustración 1: Diagrama de estados.

Se puede ver, que se emplea ciertas convenciones que se explican a continuación:

***Pantalla:***

* *P1:* En la pantalla se observa “Ingresar tiempo”.
* *P2:* En la pantalla se verá el conteo regresivo que ingresó el usuario.
* *P3:* En la pantalla se observa “Fin”.

***Teclado:***

* *T1:* El usuario ingresa el tiempo que desee.
* *T2:*El usuario puede oprimir \* para darle inicio al conteo, o # para cancelar y regresar al inicio.

***Motor:***

* *M1:* El motor empieza a girar, simulando el movimiento del plato del horno.
* *M2:* El motor se detiene al momento de terminar el temporizador.

***Variables adicionales:***

* *S1:* Se empieza el conteo cuando el usuario presiona \*.
* *S2:* Se termina el conteo y el motor se detiene.
* *S3:* Se reinicia el sistema para que el usuario pueda volver a ingresar otro valor.
* *S4:* El temporizador esta contando y se va mostrando en la pantalla.
* *S5:* Se regresa al inicio cuando el usuario presiona #.

Para la parte de la simulación en Proteus, se realizó un montaje usando: un Arduino MEGA, una pantalla LCD 16X2, una Keypad 4X3, un LED verde para avisar cuando termine el proceso, un potenciómetro, un motor DC y un driver para controlar mejor el motor. Pero al momento de probar el código en esta simulación se presentaron problemas, que es posible que sea por el programa, ya que el código se probo de forma separa y logro funcionar.

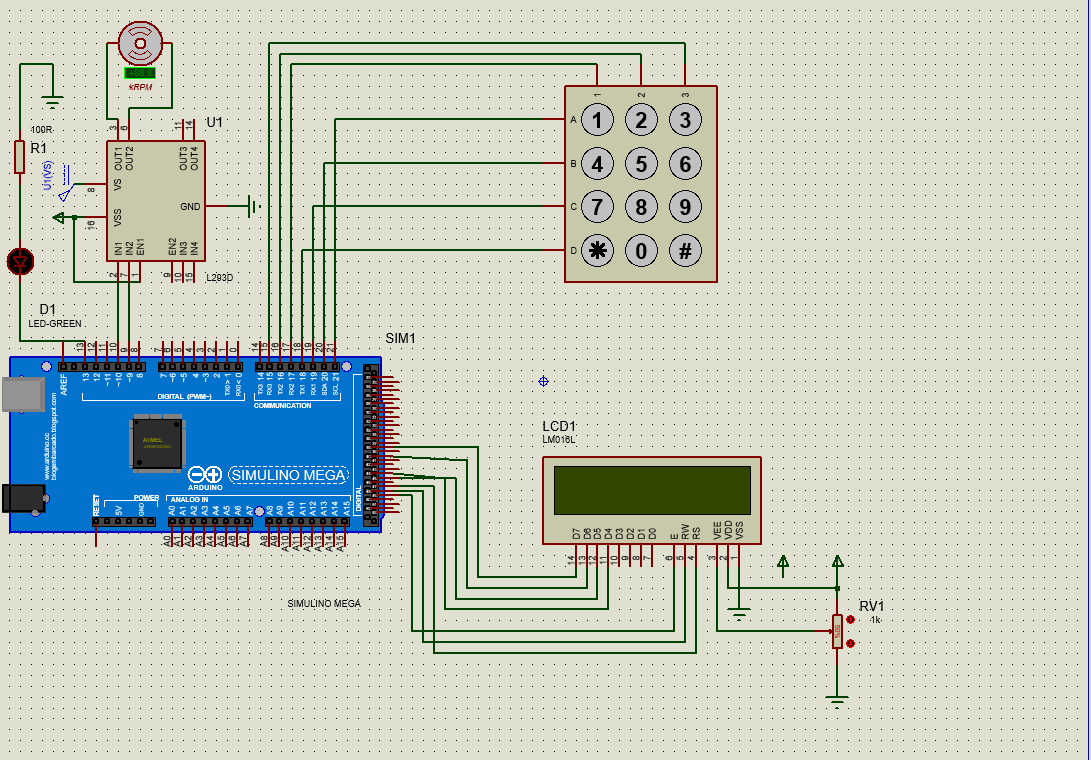


Ilustración 2: Montaje de la simulación completa.

Como se mencionó anteriormente, la simulación se dividió en dos usando un Arduino UNO en ambos casos, por una parte, se simulo el comportamiento de la pantalla con el teclado y un LED que avisara cuando el temporizador terminara. Por otra parte, se simulo el comportamiento de la pantalla con el motor, es decir, ver cómo se puede el motor mientras esta el temporizador. A continuación, se muestra el montaje de ambas simulaciones:

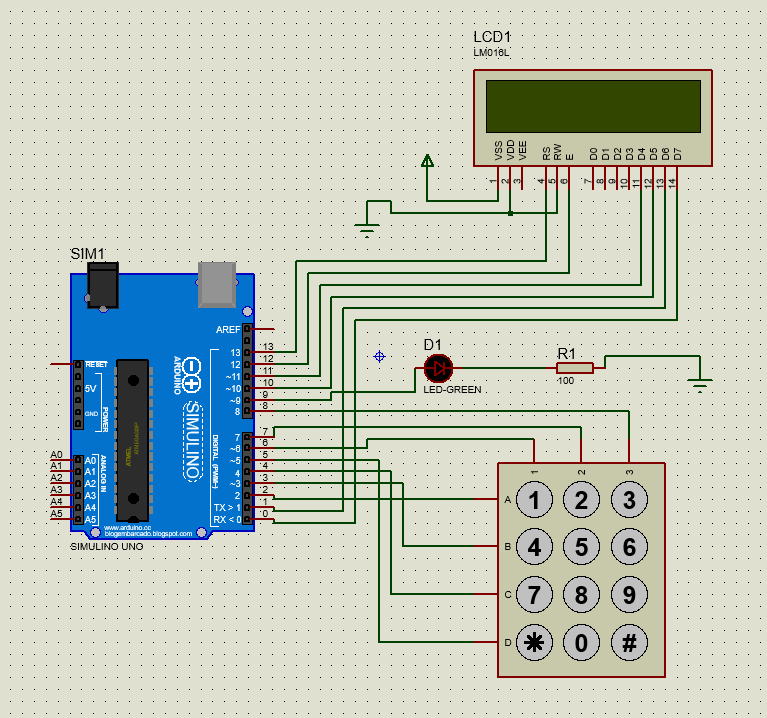


Ilustración 3: Montaje pantalla, teclado y LED.

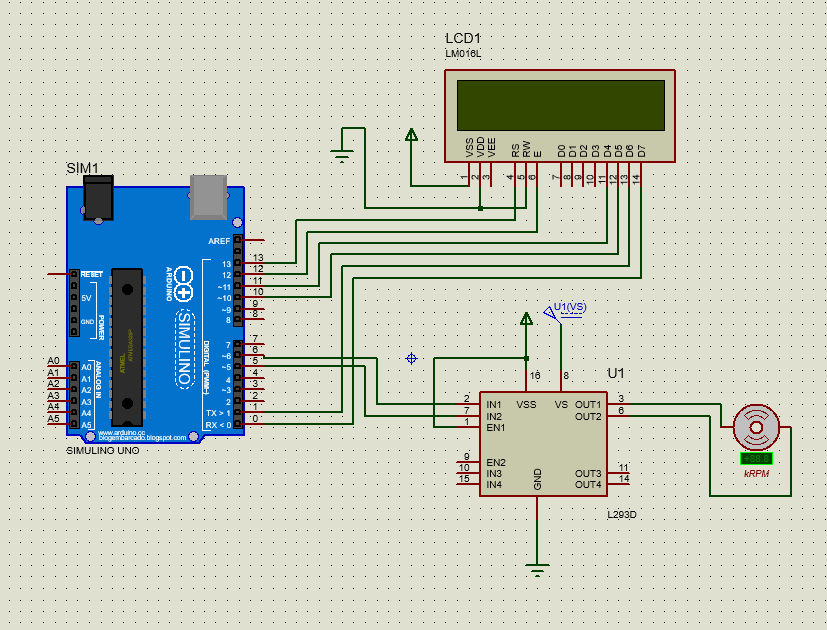


Ilustración 4: Montaje motor y pantalla.

El código implementado, se lleva a cabo de la siguiente manera, primeramente, se lleva a cabo la defunción de parámetros del teclado y la pantalla utilizados, además de otras variables que interceden en el proceso. Luego, se establece una relación de cada dígito, con el equivalente, para minutos, horas, segundos etc. Posteriormente, se establece un *void* donde define el funcionamiento del led.

En este código se inicia el parámetro del timer seleccionado, y se alternan parámetros anteriormente definidos, se reinicia la función millis, y seguimos al *Void loop* donde guardamos valores de los caracteres de tiempo definidos y se definen los casos.

# **Conclusiones**

El comportamiento de los Timers es muy importante para tener un mejor control sobre el microcontrolador, sin embargo, se tuvo varios problemas al momento de entender correctamente su funcionamiento. Además, se tuvieron problemas con el Arduino MEGA en el programa Proteus, y no se encontró otra solución que no fuera separar los componentes e intentar hacer códigos que se pudieran hacer en ambos casos, claramente teniendo algunas diferencias por el montaje.

Se tiene claro que con la ayuda del MEF la programación y el cómo se controla el sistema es mucho más fácil, por lo que, lo primero que se realizó fue eso para ayudarnos a entender el sistema completo.

# **Referencias**

[1] <https://www.academia.edu/11149589/Apuntes_a_los_sisitemas_embebidos_tema_2>

[2] <https://creatividadcodificada.com/arduino/timer-con-arduino-o-interrupciones-internas/>

[3] <https://hetpro-store.com/TUTORIALES/arduino-timer/>

[4] <https://stackoverflow.com/questions/54364494/conflicting-declaration-in-c/54364688>

[5] <https://www.youtube.com/watch?v=z-X0eKF_I64>

[6] <https://controlautomaticoeducacion.com/arduino/timer-arduino/#Timer_3_4_5_Arduino_Mega>

[7] <https://www.prometec.net/timers/>

[8] <https://aprendiendoarduino.wordpress.com/tag/timer/>

[9] <http://www.playbyte.es/electronica/arduino/lcd-keypad-shield/>

[10] <https://forum.arduino.cc/index.php?topic=240822.0>