PRÁCTICA N°3: MANEJO DEL TIEMPO

Presentado por:

MARÍA PAULA REY BARRERA

ESTEFANÍA ZULUAGA PASTOR

PROYECTO INSTRUMENTAL II

Presentado a:

Augusto Carmona Valencia

INGENIERÍA FÍSICA

DEPARTAMENTO DE CIENCIAS FÍSICAS

ESCUELA DE CIENCIAS

UNIVERSIDAD EAFIT

2020

**TABLA DE CONTENIDO**

Pág.

[**1.** **OBJETIVOS** 3](#_Toc40704248)

[**2.** **INTRODUCCIÓN** 3](#_Toc40704249)

[**3.** **SIMULACIONES COMPUTACIONALES DE LOS CIRCUITOS IMPLEMENTADOS Y PROCEDIMIENTO** 3](#_Toc40704250)

[***3.1.*** ***Montaje*** 3](#_Toc40704251)

[**3.2.** **MEF** 4](#_Toc40704252)

[***3.3.*** ***Código para el sistema embebido*** 5](#_Toc40704253)

[**4.** **RESULTADOS** 5](#_Toc40704254)

[**5.** **DISCUSIÓN Y CONCLUSIONES** 6](#_Toc40704255)

[**BIBLIOGRAFÍA** 6](#_Toc40704256)

# **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.

# **INTRODUCCIÓN**

Uno de los elementos más robustos en los sistemas embebidos son los conocidos Temporizadores o Timers. El uso de estos elementos es indispensable para el correcto funcionamiento en el control de hardware y software, dentro de sus grandes ventajes es que éstos no pasan por el contador del programa (PC) entonces siempre serán asíncronos al programa principal.

El uso de estos elementos potencia enormemente las tareas disponibles para el microcontrolador, liberando el control del tiempo y generando precisión en los controles finales del dispositivo. Por otra parte, el uso de los registros del microcontrolador genera un control absoluto sobre el desempeño del microcontrolador, sin pasar por intérpretes (softwares) innecesarios.

# **SIMULACIONES COMPUTACIONALES DE LOS CIRCUITOS IMPLEMENTADOS Y PROCEDIMIENTO**

## ***Montaje***

Se realizó el montaje mostrado en la Figura 1. Este consiste de un microcontrolador Arduino ATMEGA2560 para controlar todos los procesos, una pantalla LCD para visualizar la cuenta regresiva del microondas, un teclado o keypad que permite ingresar el tiempo deseado, un motor DC (conectado mediante un circuito integrado L293D), y un LED como indicativo que el horno está en funcionamiento y como indicador de que el proceso ha terminado.

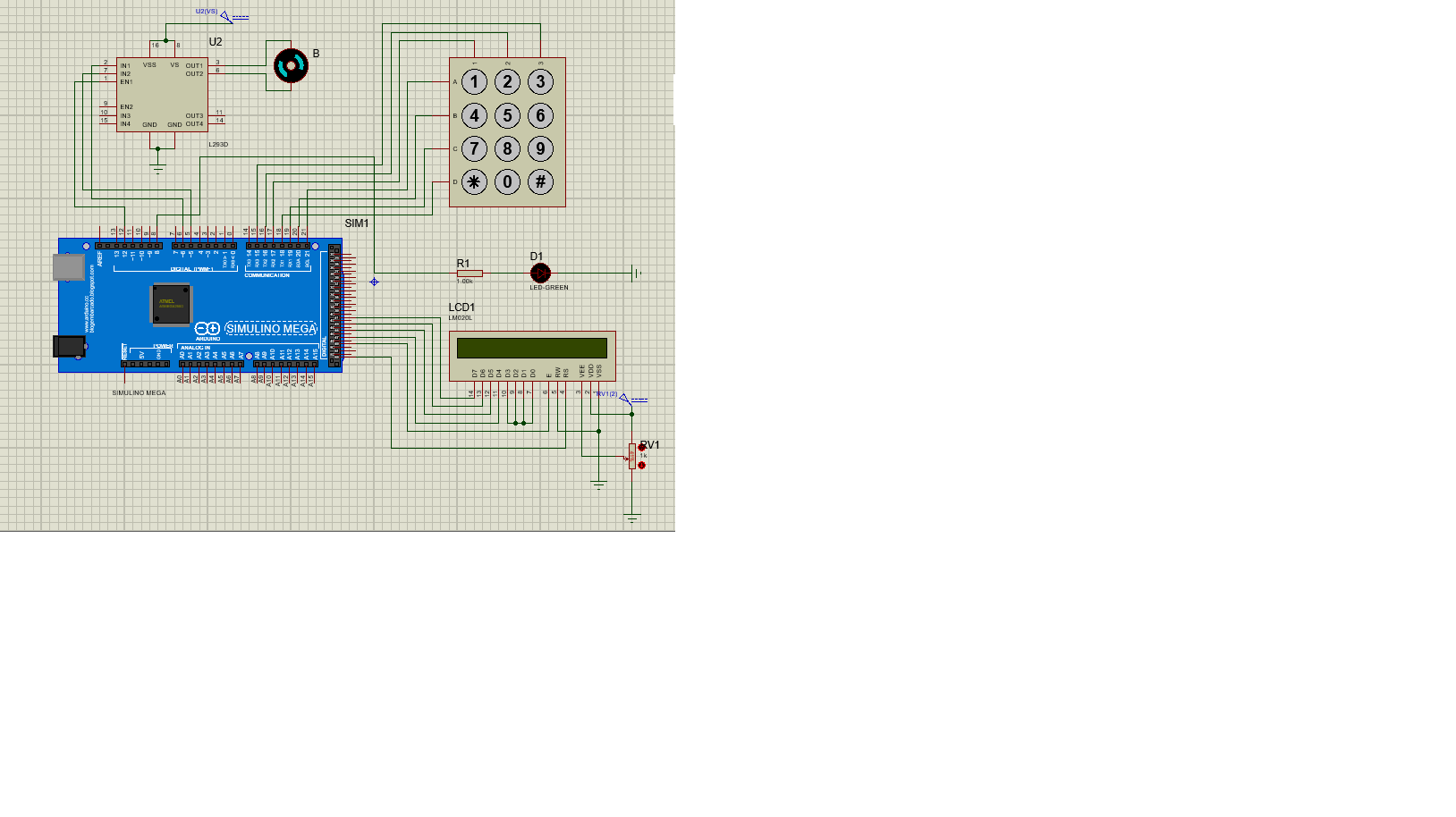


Ilustración 1. Montaje en Proteus.

En el presente laboratorio, se realizó el temporizador de un horno microondas usando un AVR como sistema embebido.

Este sistema emplea el *timer* 3 como elemento de conteo. Se diseñó de manera que el tiempo deseado se ingresa vía teclado, mostrando dicho valor sobre el *display* LCD, donde también se muestra también la cuenta regresiva. En el teclado existen tres funciones adicionales a ingresar el tiempo deseado:

* Aceptar: la tecla “\*” sirve la función de aceptar el proceso después de haber ingresado el tiempo deseado.
* Pausar: presionar la tecla “#” una vez para pausar la cuenta regresiva.
* Cancelar: presionar la tecla “#” dos veces para cancelar la cuenta regresiva.
* Reiniciar: Después de cancelar el proceso, se puede presionar la tecla “\*” para reiniciar el horno.

Con el paso del temporizador, el motor está encendido, simulando el proceso de cocción. En el momento de terminar, el LED titila para indicar que ya se finalizó el proceso.

## **MEF**

El código se realizó haciendo uso de la función switch – case. En este, cada caso representa un posible estado del sistema, para un total de 3 posibles estados que se ilustran a continuación:

Imagen que contiene objeto, interior, hombre, colgando

Descripción generada automáticamente

Tiempo establecido por el usuario

Transcurre tiempo establecido

Se oprime “#”

Se oprime “\*”

Correr

Terminar

Se enciende el motor

Inicio

Estado 3

Estado 3

Estado 1

Ilustración 2. Diagrama de MEF

## ***Código para el sistema embebido***

La unidad que aporta capacidad de cómputo al sistema es un microcontrolador Arduino ATMEGA2560, el subsistema de presentación es una pantalla LCD, y se tienen como actuadores el motor DC y el LED. Para controlar el subsistema de presentación, se emplearon las librerías *LiquidCrystal.h* y *Keypad.h.* Dentro del código adjunto a este documento se encuentra el desarrollo de este sistema con sus respectivos comentarios y/o explicaciones.

Para realizar la cuenta regresiva, es importante definir que al transcurrir un minuto, se debe pasar a 59 segundos, y no a 99 segundos. Para hacer esta especificación, mediante la función *countdown()* se determinó que si el número ingresado termina en ceros, debe pasar a un valor de 59s:

int countdown(){

if (timeroven == 0){ //Termina proceso

return 2;

}

if (timeroven%100 == 0){ // Si termina en 00, pasar a 59 s

timeroven= (((timeroven/100) - 1)\*100)+59;

}

else {

timeroven -= 1; //Restar segundos

}

return 1;

}

Otra función importante es la posición de los números ingresados, la cual se diseñó de acuerdo a cómo funciona el ingreso del tiempo en un horno microondas regular. Se definió que cada vez que llegan nuevos números, se ubican en la posición de segundos (posición 3), y se corren los otros dígitos una posición hacia la izquierda:

timeove[0] = timeove[1];

timeove[1] = timeove[2];

timeove[2] = timeove[3];

timeove[3] = key;

Para controlar el LED, se determinó que debe permanecer encendido durante el proceso y al finalizar debe titilar, lo cual se logra con la función *blink()*, donde el estado del LED se niega constantemente:

volatile byte state = LOW; //Variable para cambiar el estado del pin

void blink() { //Función para cambiar de estado el LED

state = !state;

}

# **RESULTADOS**

Se encontró que empleando los valores del *datasheet* [1] para los registros *OCR3A y TCCR3B,* el sistema no funciona de manera correcta; es decir, los segundos transcurren excesivamente rápido. Según el *datasheet*, en el registro de control del *timer* (*TCCR3B*) se tiene que para el modo CTC (Clear Time Counter), se debe poner 1 en la posición 3, y para un preescalador de 1024 bits, hay que poner 1 en la posición 2, 0 en la posición 1 y 1 en la posición 0, según la ilustración 2. Esto equivale a un valor de 1011 (en binario) para este registro, lo que equivale a darle un valor de 11 (en decimal) en el código: al hacer esto, los segundos transcurren con excesiva rapidez. De esta forma, se encontró que al asignarle un valor de 1111 (en binario), es decir, un valor de 15 (en decimal), parece funcionar mejor. Observando de nuevo la ilustración 2, se observa que este valor de 1111 significa que está usando un reloj externo.

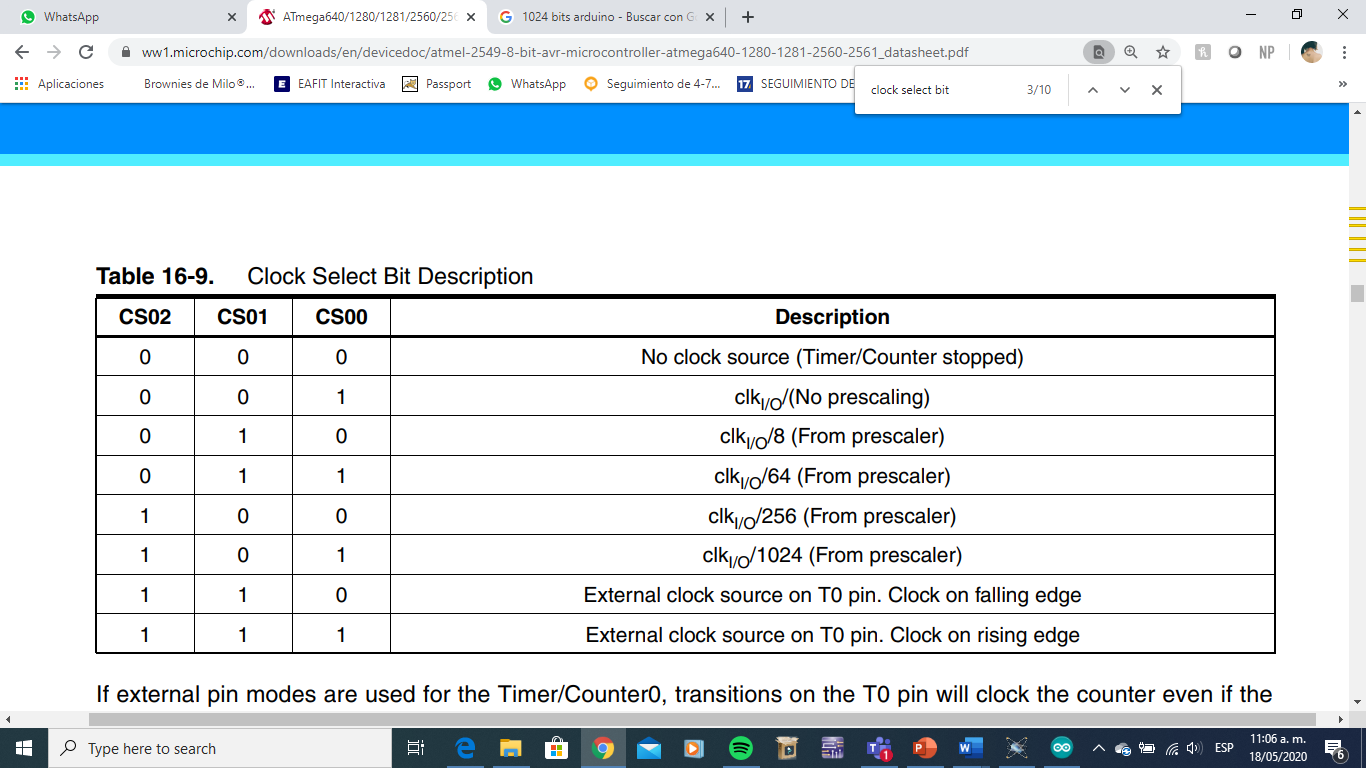


Ilustración 3. Descripción de bit de selección de reloj [1].

Para definir a cuántos ciclos equivale un segundo (con un incremento de 1Hz), se divide la frecuencia del reloj interno del microcontrolador ATMEGA2560 (16MHz) entre 1Hz multiplicado por 1024 bits menos uno, es decir:

# **DISCUSIÓN Y CONCLUSIONES**

* Se programó y se simuló el funcionamiento de un horno microondas de manera virtual usando Proteus como sistema de prueba. Para el desarrollo de esto se usó un ARDUINO ATMEGA2560 como microcontrolador. En el proceso se aprendió sobre funcionamiento de una pantalla LCD, un teclado y un motor, y principalmente, en cuanto a la programación, se trabajó con MEFs y se usó la función de timer, la cual representa muchas más ventajas comparada con la función que se suele usar para el mismo propósito, delay.

# **BIBLIOGRAFÍA**

|  |  |
| --- | --- |
| [1] | Atmel Corporation, «Atmel,» 2014. [En línea]. Available: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561\_datasheet.pdf. [Último acceso: Mayo 2020]. |