

# INSTITUTO POLITÉCNICO NACIONAL - UPIIZ



## Proyecto Final

Bernardo Martínez Medina

Adrián Torres Aranda

**Fecha de Entrega:** 14/06/2022

**Programa Académico:** Implementación de sistemas digitales

**Docente:** Ramon Jaramillo Martínez

8to Semestre

ING MECATRÓNICA

## • Resumen

Se programo y se diseñó el circuito para una alarma especialmente para las alarmas de las máquinas de comida en las empresas ya que así nos dice el tiempo en que ya está terminado el proceso o reiniciar el tiempo

## • Introducción

Este proyecto tiene bastantes usos en las empresas como el cual nos avisa una alarma para las cámaras frigoríficas

Las cámaras frigoríficas son espacios con una atmósfera modificada a temperaturas bajo cero. Solo por estas circunstancias ya es necesario contar con un sistema de alarma que evite riesgos innecesarios. Las posibilidades de que se dé algún percance o incluso la muerte son reales [1].

Para todos los procesos de la comida se puede implementar la alarma para con un indicador nos muestre cuando ya este o la comida ya se puede procesar.

Las alarmas cuentan ascendente y descendente son soluciones de seguridad alimentaria esenciales para cronometrar la cocción, la conservación del calor y el frío y el enfriamiento de los alimentos en restaurantes, cocinas comerciales y todas las áreas de servicio de alimentos [2].

## • Desarrollo

Para lograr realizar el proyecto como se tenia previsto primero se hizo una consideración sobre el cómo se plantearía el código al igual que hardware que se utilizaría para la creación de el circuito y como se comunicaría el usuario con él.

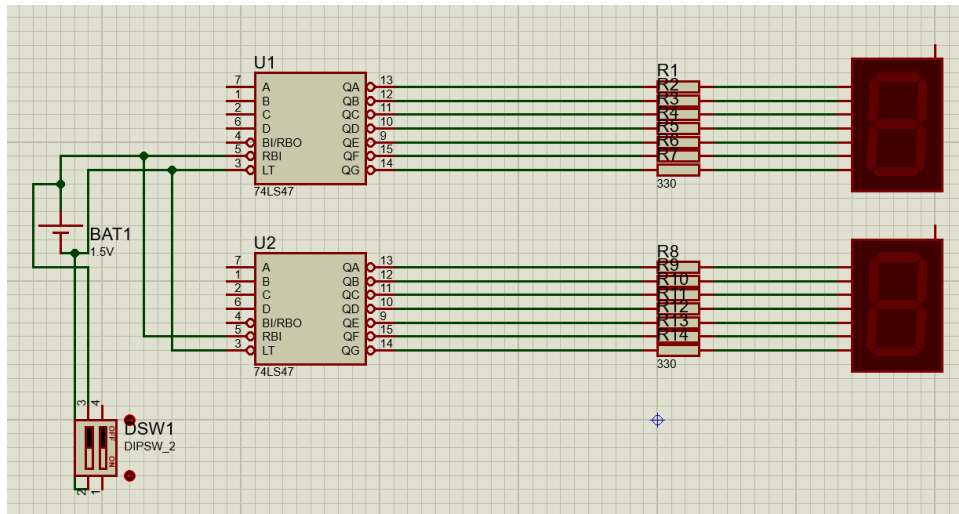
Para describir el comportamiento del circuito se realizo una secuencia en binario en formato descendente desde el 59 hasta el 00, donde los puertos usados para lograr esta programación fueron los siguientes.

```
GPIO_setAsOutputPin(GPIO_PORT_P6, GPIO_PIN0);           // P6.0 -> A1
GPIO_setAsOutputPin(GPIO_PORT_P3, GPIO_PIN2);           // P3.2 -> B1
GPIO_setAsOutputPin(GPIO_PORT_P3, GPIO_PIN3);           // P3.3 -> C1
GPIO_setAsOutputPin(GPIO_PORT_P4, GPIO_PIN1);           // P4.1 -> D1

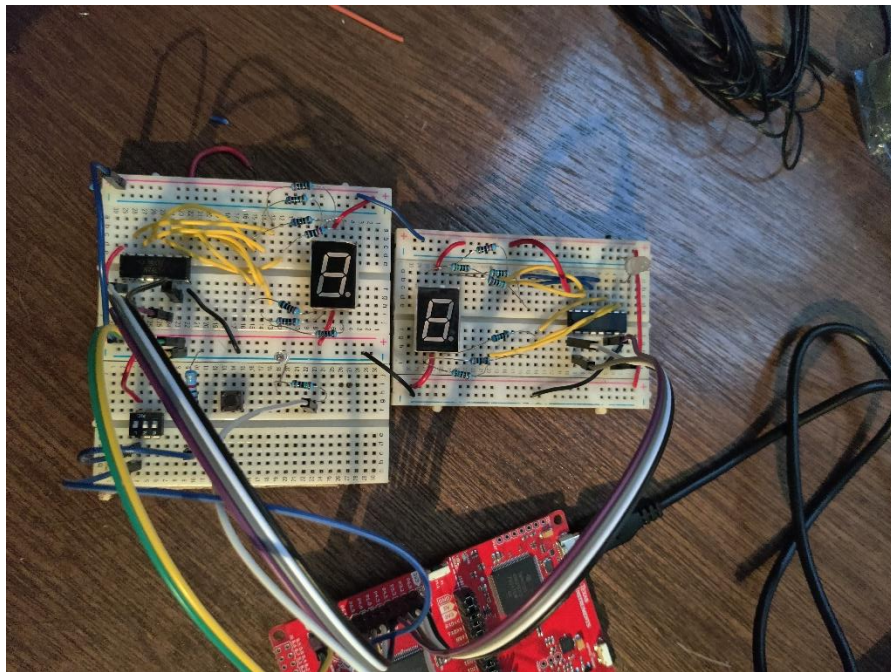
GPIO_setAsOutputPin(GPIO_PORT_P4, GPIO_PIN3);           // P4.3 -> A2
GPIO_setAsOutputPin(GPIO_PORT_P1, GPIO_PIN5);           // P1.5 -> B2
GPIO_setAsOutputPin(GPIO_PORT_P4, GPIO_PIN6);           // P4.6 -> C2
GPIO_setAsOutputPin(GPIO_PORT_P6, GPIO_PIN5);           // P6.5 -> D2
```

Donde se pueden observar dos paquetes de salidas, diferenciadas por su número. Esto debido a que el diseño del circuito contendría 2 decodificadores del tipo 74LS47. Por lo tanto, se diseñó el siguiente circuito:

Donde las entradas de ambos decodificadores al igual que la salida del dipswitch son puertos provenientes de la tarjeta MPS432.



Con este diseño se procedió a la construcción del circuito donde se obtuvo el siguiente resultado.



Por otro lado, para la descripción del circuito se hizo el siguiente desarrollo.  
Para la configuración del timer encargado de tener una secuencia exacta de 1 segundo se asignaron los siguientes valores al Timer A1

```

P2->SEL1 &= ~2;          /* C
P2->SEL0 &= ~2;
P2->DIR |= 2;            /* F

TIMER_A1->CTL = 0x01D1;
TIMER_A1->CCR[0] = 512 - 1;
TIMER_A1->EX0 = 7;

```

Además de agregarse a una secuencia del tipo while donde se togglearia un led para indicarnos cada que pasara 1 segundo. Cada que este led prendiera se descontaría 1 valor al primer auxiliar usado, en este caso el “cont1”, auxiliar encargado de los segundos.

```

while ((TIMER_A1->CCTL[0] == 0)); /*
TIMER_A1->CCTL[0] &= ~1;
P2->OUT ^= 2;
cont1 = cont1 - 1;

```

Entonces con el primer auxiliar establecido para nuestra secuencia se hizo uso del siguiente código para lograr un conteo de 59 a 00, el cual a llegar a este ultimo valor se mantendrá en este hasta que el usuario ingrese un nuevo tiempo y de pulse el botón de inicio.

```

if (cont1 == -1 )
{
    cont2 = cont2 - 1;
    cont1 = 9;
}
if(cont2 == -1)
{
    cont2 = 5;
}

```

Para la asignación de valores se programo en una secuencia binaria ya que como se menciono antes al hacer uso del decodificador solo se tendrían que programar 4 puertos.

```

if (cont1 == 0)
{
    GPIO_setOutputLowOnPin(GPIO_PORT_P6, GPIO_PIN0);          // P6.0 -> A1
    GPIO_setOutputLowOnPin(GPIO_PORT_P3, GPIO_PIN2);          // P3.2 -> B1
    GPIO_setOutputLowOnPin(GPIO_PORT_P3, GPIO_PIN3);          // P3.3 -> C1
    GPIO_setOutputLowOnPin(GPIO_PORT_P4, GPIO_PIN1);
}
if (cont1 == 1)
{
    GPIO_setOutputHighOnPin(GPIO_PORT_P6, GPIO_PIN0);          // P6.0 -> A1
    GPIO_setOutputLowOnPin(GPIO_PORT_P3, GPIO_PIN2);          // P3.2 -> B1
    GPIO_setOutputLowOnPin(GPIO_PORT_P3, GPIO_PIN3);          // P3.3 -> C1
    GPIO_setOutputLowOnPin(GPIO_PORT_P4, GPIO_PIN1);
}

```

Pero para que el usuario lograra establecer un tiempo se hace uso de un nuevo puerto el cual cada vez que se detectara un alto, se sumaria un valor al contador secundario encargado de los decimas y el contador 1 obtendría un valor de 0.

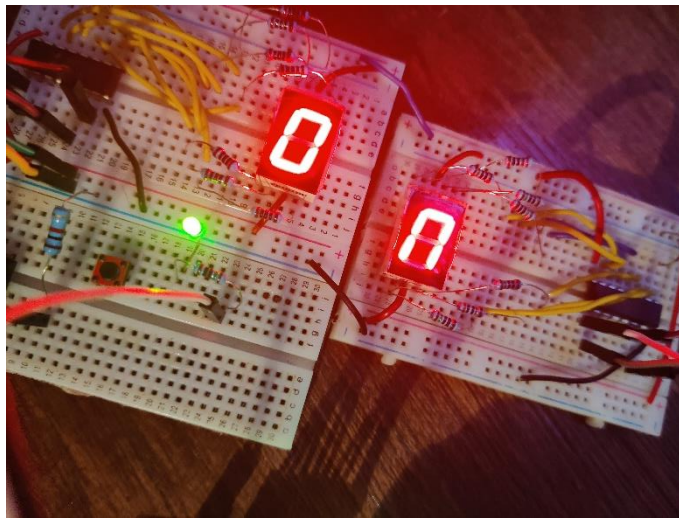
```

if (GPIO_getInputPinValue(GPIO_PORT_P1, GPIO_PIN1) == GPIO_INPUT_PIN_LOW)
{
    for(i = 0;i<45000;i++){
    }
    cont2 = cont2 + 1;
}

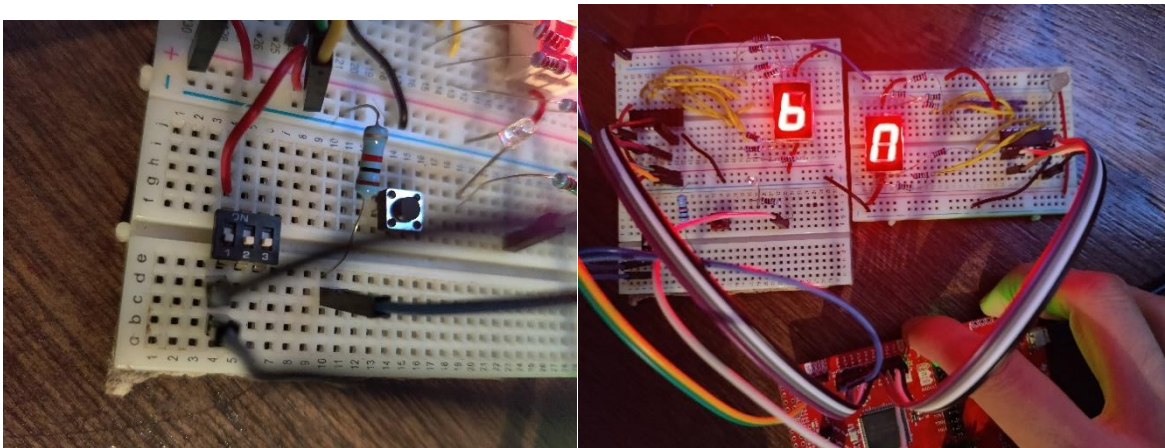
```

- **Análisis de resultados**

Por medio del presente proyecto se tienen una aplicación real donde podemos observar que al accionar este dispositivo y la cuenta llegar a 0 se prende un led, principal indicador para saber que el tiempo a llegado a su fin.

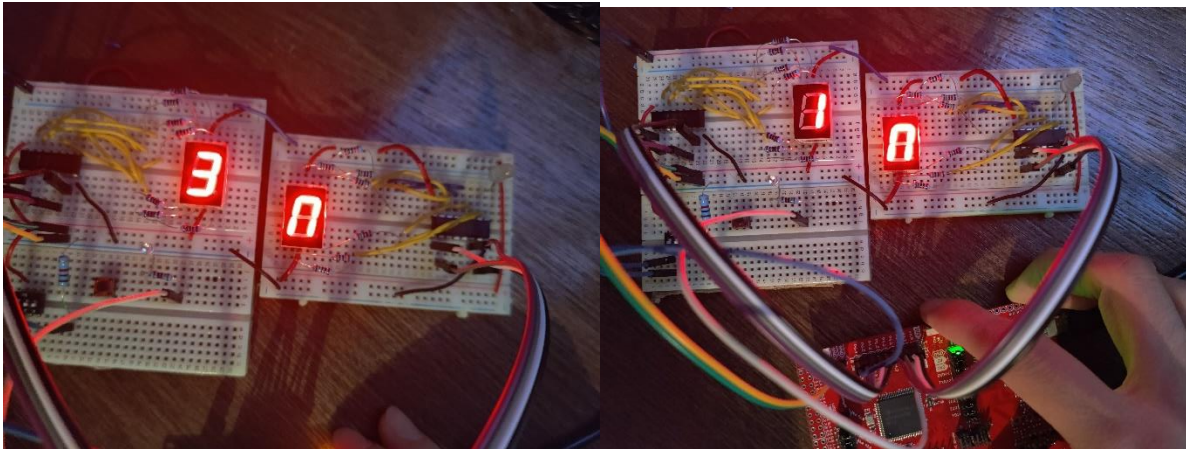


Aunque abarcando mas el diseño del prototipo, este se acciona o por medio de un dipswitch el cual en caso de querer aplicarse fuera del entorno de las placas de prueba o protoboards podria ser sustituido por un boton o un swithc convecional.





Para lograr los conteos o cambios de tiempo según requiera la aplicación se tiene un botón el cual al llegar a un tiempo máximo establecido de 60 segundos este se reinicia a un valor de 10 segundos.



Por lo cual después ver el comportamiento del dispositivo podemos ver como el sistema tiene un funcionamiento deseado, aunque con muchas mejoras por realizar destacando el sistema de elección de tiempo. Una forma en la cual se pensó que podría hacerse esta selección de tiempo es por medio de un capacitor y la lectura de puerto UART- Esto no solo haría más rápida la selección de tiempo si no que la facilitaría al replicar un comportamiento con un reloj analógico.

## • Conclusiones

Adrián Torres Aranda

Para la realización del proyecto lo más complicado fue las elecciones de tiempo y como acomodar esa parte en la programación además de implementar un indicador para saber cuándo acaba el tiempo y un botón para poder reiniciar el tiempo y al final se implementó el circuito para ver su realización mediante los displays para mostrar el tiempo, un led para que sea el indicador y un botón para reiniciar el tiempo.

Bernardo Martínez Medina

Con el desarrollo del proyecto final obtenemos lo que es una conclusión de todo lo aprendido para ser aplicado a un problema real haciendo uso de una gran cantidad de diseño de sistemas digitales vistos anteriormente destacando los puertos TIMER, UART y los GPIO, mismo que fueron indispensables para una comunicación y desarrollo del código en base a nuestra necesidad. Como tarea en cuanto a esto es que por medio de otros tópicos que abarcan los sistemas digitales es que se puede implementar una memoria flash con el uso de comunicación encriptada. Uno de los problemas que se tuvo fue el reinicio de todo el sistema ya que la estructura del código en ciertas secuencias ignoraba esta instrucción por lo cual se tuvo que reestructurar este para tener un funcionamiento como el deseado.

## • Referencias

- [1] «Bernad Refrigeracion,» [En línea]. Available: <https://www.josebernad.com/algunos-tipos-de-alarmas-para-camaras-frigorificas/>. [Último acceso: 18 junio 2022].
- [2] «Deltatrak,» [En línea]. Available: <https://www.deltatrakmexico.com/kitchen-timer/50048-flashcheck-count-up-count-down-timer>. [Último acceso: 18 junio 2022].