

Proyecto de Curso

---

Sistemas Electrónicos

(Introducción al manejo del  
Microcontrolador MSP430G2553)



Alejandro Roche Aniento  
Alfredo Zarazaga Montalbán

# Índice:

1. Introducción .....	3
2. Recursos usados de la placa .....	4
3. Descripción del código .....	6
1.Máquina: Modificador de pizzas.....	7
2. Máquinas: Cocina .....	9
3. Funciones: dibuja .....	10
4. Ejemplo de uso y galería de imágenes .....	11
5. Uso de memoria .....	14
6. Conclusión .....	15

## 1.- Introducción.

Como ingenieros especializados en el campo de la electrónica, uno de nuestros rasgos más característicos debe ser el correcto y frecuente uso de la lógica, pues nuestros objetivos sólo podrán llevarse a cabo si conseguimos, con nuestra destreza, poder definir cómo y cuándo ocurran las acciones.

Para ello, haremos uso de todo lo que esté a nuestro alcance (lógica booleana, lenguajes de programación, hardwares y softwares adaptados a la situación, etc), y así poder ofrecer la mejor solución al problema que se nos presenta.

Para que comencemos a desarrollar nuestras capacidades, además de los ejercicios propuestos en la asignatura, se nos presenta a los estudiantes un proyecto de curso, en el que tendremos que aplicar todo lo aprendido durante las clases teórico-prácticas.

Al ser de libre elección, estos alumnos han escogido desarrollar un menú de selección tomando como inspiración los que se encuentran presentes en las cadenas de comida rápida como son McDonald's o Burger King, enfocado a una pizzería autoservicio en el que tú mismo te preparas la pizza.

Se presenta una interfaz en la que se podrá personalizar una pizza con los ingredientes que se le proporcionan al cliente, además de otorgarle un turno y un tiempo de espera de preparación, en relación al número de ingredientes, además de su respectivo precio.

Se ha añadido también una simulación de cocina, la cual cuenta con dos puestos de trabajo y un horno, de la cual van saliendo las pizzas y avisando de que los pedidos están listos. En todo momento, el cliente sabrá la cantidad de ingredientes que quedan, avisando a cocina de que recarguen alguno si se acaba, el precio de añadirlo y si ha llegado a la cantidad máxima de dicho ingrediente.

Así podemos conseguir demostrar tanto el manejo del microcontrolador proporcionado durante la asignatura, como la comunicación con el mismo mediante las entradas y salidas, así como de sus periféricos, pues se ha usado además el complemento del Boosterpack, también proporcionado por el departamento.

## 2.- Recursos usados de la placa.

Para llevar a cabo el proyecto, se han usado distintas prestaciones que nos ofrece el microcontrolador MSP430G2553.

En primer lugar, en el programa se incluyen varias librerías importantes que proporcionan las funcionalidades necesarias para manejar el hardware y la lógica gráfica:

- `<msp430.h>`: librería del microcontrolador.
- `"uart_STDIO.h"`: librería que gestiona la comunicación serie (UART).
- `"grlib.h"`: librería gráfica.
- `"Crystalfontz128x128_ST7735.h"`: librería para la pantalla LCD.
- `"HAL_MSP430G2_Crystalfontz128x128_ST7735.h"`: librería diseñada para vincular el microcontrolador con la pantalla.
- `<stdio.h>`: librería de entrada y salida en C.

Por un lado, se configura un reloj a 16 MHz y se habilita el puerto serie (UART). Se define el temporizador de forma que trabaje con el SMCLK como fuente de reloj con el modo UP/DOWN habilitado por el MC\_3 y se configura la UART con una velocidad de 115200 baudios. En cuanto al temporizador, este se configura de modo que haya una interrupción cada 50ms. Esta interrupción provoca el incremento de varias variables; que se usan para medir tiempos; además de permitir que la placa salga del modo reposo a partir de LPM0\_EXIT.

También se hace uso de las interrupciones, pues con cinco veces el desborde el contador se sale del modo de bajo consumo LPM0, además de que cada veinte veces que lo hace, se consigue contar un segundo, útil posteriormente.

Además, también se ejecuta la interrupción cuando se detecta movimiento en el joystick, volviendo a sacar el micro del modo de bajo consumo LPM0 para leer el valor del joystick. Esto es gracias a las funciones usadas durante las sesiones prácticas **lee\_ch** e **inicia\_ADC**. Al igual que del BoosterPack se usa el joystick, también se usa la pantalla del mismo, en el cuál se presenta el personalizador de pizzas.

Cada vez que se realiza un pedido, su precio se guarda en un vector, el cual se va guardando en flash mediante la función **guarda\_flash**, usando como entrada el vector de precios y la dirección en la que se guardarán los valores, usando en nuestro caso la correspondiente al segmento D.

Como entradas y salidas al sistema se tienen:

1. P2.5: referente al botón del joystick.
2. P1.1 y P1.2: para la comunicación por puerto serie entre el usuario y el microcontrolador.

Cabe recalcar que todos los valores referidos a precios y tiempos se han escogido enteros y pequeños para tener una simulación acorde al tiempo de muestra y con el objetivo de no abusar de la memoria del microcontrolador.

### 3.-Descripción del código.

Ahora, se procede a comentar el programa realizado.

Cabe destacar que se han definido los valores de las cantidades máximas permitidas para cada ingrediente, así como los valores base del precio (5 euros) y tiempo de espera, y los valores máximos y mínimos de la variable 'x', la cual se usa para el movimiento del selector. Estas variables se definen dado a su facilidad de modificación (son altamente utilizadas para cálculo de precios, tiempo de espera...)y ya que permite un menor consumo de la memoria(es importante siempre que se pueda ahorrar espacio en memoria para un mejor funcionamiento de la placa)

Además de las configuraciones del timer, de la UART y de los pines de entrada y salida, se definen las funciones anteriormente mencionadas, a las que se les añaden las funciones de **dibuja\_pizza\_vacia** y **dibuja\_cantidades**, con las cuales se dibuja la pizza y las cantidades que aparecen en la pantalla del BoosterPack. También se declaran todas las variables globalmente, de las cuáles vamos a definir las siguientes:

1. Tms y seg: variables que sirven para salir del modo de bajo consumo LPM0 y contar un segundo, respectivamente.
2. Cadena, carácter y cant: variables usadas para manejarlas por puerto serie.
3. Cantidad\_x: variable referida a cuanto queda de cada ingrediente x en la cocina.
4. C\_x: variable referida a la cantidad de ingrediente x que se encuentre en la pizza que se está personalizando.
5. Precios: vector que guarda los precios de los pedidos.
6. Pm: guarda el precio medio de los pedidos realizados.
7. Hueco: indica si el horno está ocupado.
8. Puestos: número de puestos de trabajo disponibles.
9. Pedido: variable que indica a la cocina de que ha llegado un nuevo pedido.
10. Tiempos de espera: variables que indican los tiempos de espera en cada lugar.

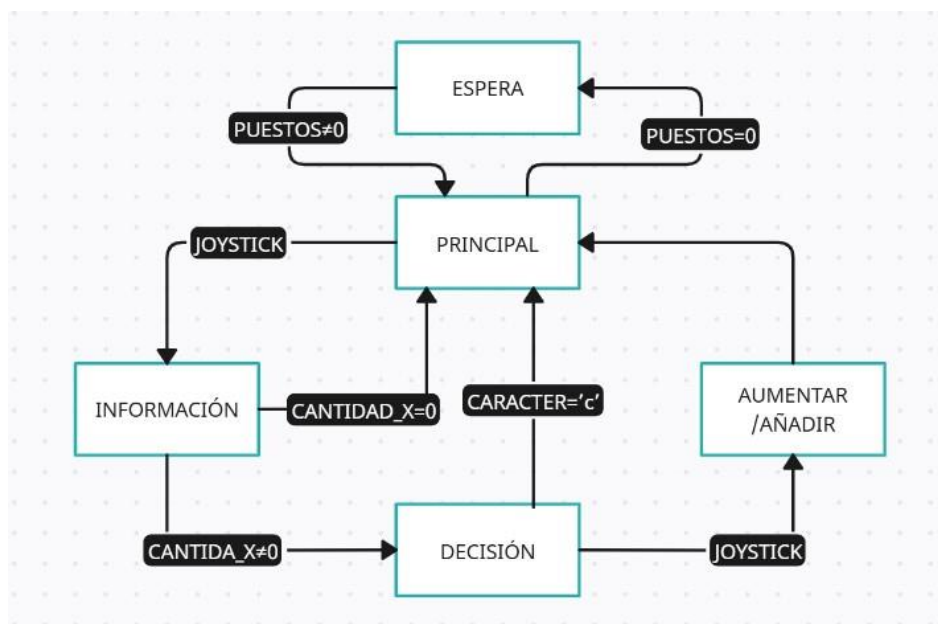
11. x: indica la posición deseada del selector.
12. Pos: indica la posición actual del selector.
13. Eje x: usada para leer el joystick.

Tras las configuraciones del timer, la UART, los pines usados, se dibuja la primera pizza vacía, se da la bienvenida y se entra en el bucle while(1), en el cual se están ejecutando las dos máquinas de estado de nuestro proyecto, además de poder solicitar recargar los ingredientes tras que se hayan agotado, enviando 'r' por puerto serie, y poder consultar el precio medio de los pedidos, enviando 'q'.

Cabe destacar el uso de un temporizador por pantalla el cual marca el tiempo de espera de la última pizza. Dicho temporizador se configura en base al timer implementado, el cual cuenta con una variable que calcula los segundos transcurridos una vez salido del reposo de la placa. Por tanto, el funcionamiento es básico, ya que una vez que se calcule el tiempo de espera, se le va restando esta variable que aumenta cada segundo, y una vez terminado el temporizador (es decir, llega a 0) se pone a 0 dicha variable para dejar de restar por pantalla.

En las siguientes líneas, se procederá a describir dichas máquinas.

### 3.1- Máquina: Modificador de pizza.

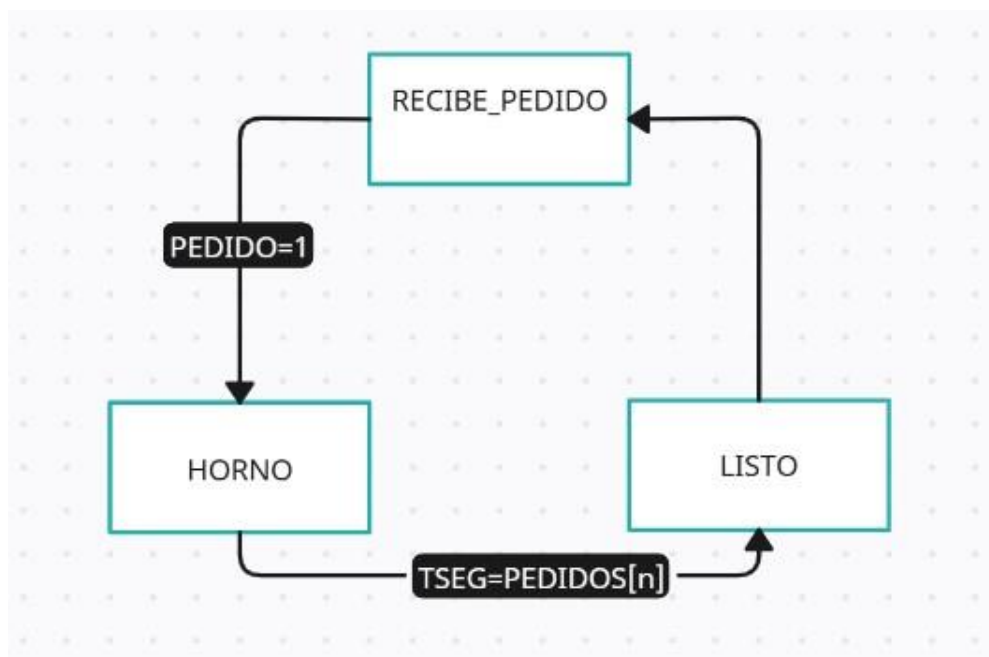


1. Principal: Es el estado donde movemos el selector por la pantalla según el ingrediente que queramos añadir a nuestra pizza. Al pulsar el botón del Joystick (P2.5), se va al estado de Información. En caso de que no hubieran puestos de trabajo, se avanza al estado de espera. Si se envía por puerto serie el carácter 'd', se realiza el pedido (PEDIDO=1) que queremos. En este punto, se calcula el tiempo de espera en función de los ingredientes añadidos y de los pedidos anteriores además de mostrarte por pantalla el precio de tu pedido. Posteriormente se mira la disponibilidad del horno y de los puestos de trabajo, además de reiniciar las variables para un nuevo pedido.
2. Espera: Estado auxiliar para el caso de que no queden más puestos de trabajo. En cuanto se libere un puesto, se volverá al estado principal para poder editar otra pizza. Se tiene que tener en cuenta que en la pizzería solo trabaja un cocinero y que por tanto los puestos de trabajo son limitados. Esto se traduce a que, se dispone de 2 puestos de trabajo donde se pueden dejar las pizzas ya editadas, además de un horno donde se cocinarán las pizzas posteriormente. Por tanto, una vez editado una pizza esta se mete directamente al horno, y se da la opción de poder crear 2 más a la espera de que salga la pizza que ocupa el horno. Para no sobrecargar al cocinero, se crea este estado en el que no se deja modificar ninguna pizza cuando tanto el horno como los puestos de trabajo están ocupados.
3. Información: Estado que nos ofrece información sobre el producto. En caso de que se haya alcanzado el límite de dicho producto, se nos envía al estado Principal para seleccionar otro. Lo mismo ocurre si no quedan unidades de ese producto, enviándonos al Principal para seleccionar otro producto o esperar que se recargue.  
En otro caso, nos comunica cuantas unidades quedan y los que nos costaría añadirlo a nuestro pedido. En este último caso, se va al estado de Decisión.



4. Decisión: Aquí decidimos si queremos añadir este producto al pedido. Si hemos recapitado y no lo queremos, enviamos el carácter 'c' por puerto serie, y volvemos así al estado Principal. En caso de que lo queramos añadir, volvemos a accionar el pulsador del Joystick, enviándonos al estado de Aumentar/ Añadir.
5. Aumentar/ Añadir: Aumenta la variable asociada a dicho ingrediente (C\_x), y disminuye la cantidad disponible del mismo (Cantidad\_x). Además, se le comunica al cliente el nuevo precio de su pedido (se añade 1 euro por el Queso, 2 por el Jamón York, 1 por el Pimiento y 3 por el Pepperonni), para volver al estado Principal por si se quiere añadir más ingredientes o realizar ya el pedido.

### 3.2- Máquina: Cocina.



1. Recibe Pedido: cuando se recibe un pedido (PEDIDO=1), se vuelve a poner a 0 la variable, se resta el hueco del horno y se libera un puesto de trabajo, para ir al estado de horno.

2. Horno: La pizza estará dentro del horno el tiempo que marque el pedido. En cuanto esto ocurra, se liberará el hueco del horno, se mirará el siguiente pedido y se irá al estado Listo.
3. Listo: Se le indica al cliente que su pedido ya está listo, y se vuelve al estado de Recibe Pedido.

Cabe destacar que esta máquina de estados funciona independientemente de la máquina de estados de la edición de la pizza. Esto permite que se puedan editar varias pizzas mientras una se haga en el horno, consiguiendo un funcionamiento más real y productivo. Además, al liberar todos los puestos de trabajo y horno, es importante comentar que a la hora de editar una pizza si el horno esta libre esta pasa directamente al horno para optimizar el tiempo

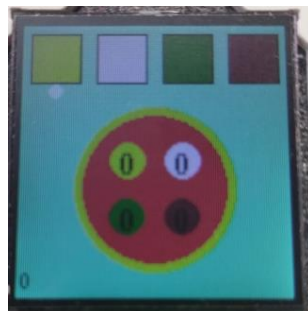
Mas allá de la idea general, el funcionamiento obviamente es ideal, es decir, si tenemos 2 pizzas esperando y una en el horno, una vez se termine la pizza del horno, automáticamente se entrega al cliente y se recarga el horno con otra pizza, y así sucesivamente.

### 3.3- Funciones: **dibuja\_pizza\_vacia** y **dibuja\_cantidades**.

La función **dibuja\_pizza\_vacia** limpia la pantalla y dibuja una nueva pizza vacía, ya que una vez añadida una pizza a un puesto de trabajo se da la opción al usuario de poder editar una siguiente pizza en caso de que se haya libre algún puesto de trabajo. La función **dibuja\_cantidades** escribe la cantidad de cada ingrediente en su respectivo lugar.

#### 4-Ejemplo de uso y Galería de imágenes.

Bienvenido a PizzaBuffet Arazar, escoja sus ingredientes



Se mueve el selector con el Joystick para seleccionar el ingrediente, en este caso se selecciona el Queso dando click al botón del Joystick (El recuadro amarillo corresponde

con el queso, el rosa con el jamón York, el verde con los pimientos y por ultimo el rojo con el pepperoni).

Al seleccionar el queso, por inicialización la cantidad en cocina es 0 , se informa que no quedan unidades de ese ingrediente. Por tanto, se da la opción de o bien elegir otro ingrediente o recargarlo:

No quedan unidades de este producto

En este caso se muestra por pantalla la recarga del ingrediente pulsando la tecla 'r' por puerto serie:

Se ha recargado el queso

Se vuelve a seleccionar el queso con el botón del Joystick, y se indica lo que queda y lo que cuesta.

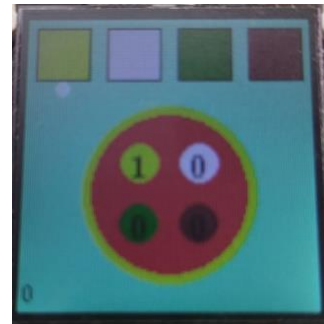
Quedan 2 unidades de queso  
Precio de la unidad 1 euros

En este punto el usuario puede decidir cancelarlo, pulsando 'c' por puerto serie, o añadir el ingrediente volviendo a pulsar el Joystick. En este caso se decide cancelar:

Operacion cancelada

Ahora se decide añadir el queso. Aumenta la cantidad por pantalla, envía el precio total de la pizza y al volver a seleccionarlo disminuye la cantidad en cocina:

```
El precio de su pedido es de 6 euros.  
Quedan 1 unidades de queso  
Precio de la unidad 1 euros
```



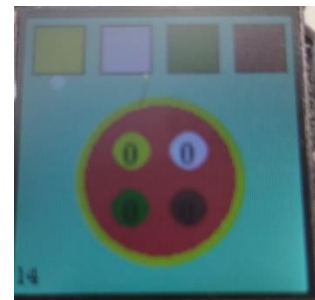
Si se alcanza la cantidad máxima, se avisa por puerto serie y se vuelve al estado de selección teniendo bloqueado añadir mas cantidad de ese producto:

```
Has alcanzado el limite de este producto
```



Una vez el usuario ha decidido el pedido a su gusto, se puede pulsar por puerto serie 'd' y mandar la pizza a un puesto de trabajo, en caso de que el horno este libre, la pizza pasa directamente al horno. Además, se informa el turno del cliente y el tiempo que va a tardar en realizarse su pedido.

```
Su turno es el 0 y la pizza va a tardar 21 segundos
```



(El tiempo por pantalla LCD es un cronometro que marca cuanto tiempo queda para la última pizza mientras que el que se muestra por puerto serie es el tiempo exacto una vez mandado el pedido a cocina, por tanto, las fotos no coinciden por que al hacer la foto se tarda unos segundos).

Tras pasar el tiempo, se comunica que la pizza está lista con el número del turno otorgado al usuario.

La pizza 0 esta lista

Al observar la memoria flash, se ve que este primer precio se ha guardado (5 de base más 3 de queso).

0x1000 8 0 0 0 0 0 0 0 0 0

## 5-Uso de memoria

```
MSP430: Flash/FRAM usage is 15574 bytes. RAM usage is 268 bytes.  
MSP430: Warning: On-board LDO is now used to supply the target with power
```

Se observa que con nuestro proyecto nos acercamos bastante al límite de memoria que nos ofrece el MSP430G2553. Sin embargo, esto no impide el correcto funcionamiento del proyecto.

Mejoras a destacar; se podría añadir externamente un periférico que ahorre espacio en memoria, lo cual puede permitir una interfaz más detallada y visual para la comprensión del usuario.

Por ejemplo:

- Añadir un logotipo para cada ingrediente para poder diferenciarlos correctamente.
- Que al añadir el ingrediente este mismo aparezca en la pizza repetidas veces simulando lo que sería una pizza real.
- etc.

## 6-Conclusión

El desarrollo de este proyecto ha permitido aplicar de manera práctica los conocimientos adquiridos durante la asignatura, desde la programación de microcontroladores hasta el uso de periféricos y la gestión eficiente de memoria.

A través de la simulación de una pizzería, se ha podido replicar de manera efectiva un sistema interactivo en el que el usuario puede personalizar un pedido, visualizar precios y tiempos de espera en pantalla, y recibir notificaciones sobre la disponibilidad de ingredientes y el estado de su orden. El manejo de varias máquinas de estado facilita el funcionamiento del sistema permitiendo haciéndolo más sencillo y ordenado.

Como posibles mejoras futuras, se podría optimizar aún más el uso de memoria, implementar un sistema de almacenamiento más dinámico para los pedidos, o incluso expandir la funcionalidad del proyecto con nuevas opciones en la interfaz de usuario.