

Tablero Indicador de Turnos para Oficinas Públicas

Maggiolo Gustavo – Caballero Raul – Reula Germán – Pfarher Iván

Email: {gustavomaggiolo, raul_caballero, germanreula}@frp.utn.edu.ar ivan_pfarher@hotmail.com

Facultad Regional Paraná

Universidad Tecnológica Nacional

Almafuerte 1033 – 3100. Entre Ríos. Argentina

Eje temático: aplicaciones académicas

Palabras claves: tablero, turnos, circuito electrónico.

RESUMEN

El presente trabajo se encuadra en las actividades prácticas de la cátedra Técnicas Digitales II dentro de la carrera de Ingeniería en Electrónica en la UTN-FRP y tiene como objeto iniciar a los alumnos en la programación de los microcontroladores.

Este caso consiste en el desarrollo de un tablero indicador de turnos, similar al que usan en oficinas públicas y otras organizaciones, con dos dígitos indicando el turno, un dígito para el número de caja o escritorio y una bocina que indica el cambio del turno.

El sistema se conforma por dos cajas, o escritorios, que tienen dos pulsadores uno de ellos, y el otro tiene tres pulsadores, que ya indicaremos para que; un tablero con los dígitos que indica el turno y el número de caja. Cada pulsador está conectado directamente al módulo que visualiza los números; a su vez, identifica que número de pulsador se presionó para indicar el número de caja que está libre.

Uno de los módulos pulsadores cuenta con la opción de resetear el indicador de turnos, lo que permite restablecer el sistema para iniciar una nueva secuencia de atención.

Cada vez que se incrementa el contador de turnos, se hace sonar una bocina para indicar el evento de cambio de turno.

Introducción

No es de hoy en día, que llegado a una oficina, de servicios públicos o privados, las personas tengan que esperar para ser atendidos.

En su comienzo se realizaba la selección de quien era atendido haciendo una cola, uno tras otro, y cuando llegaba el turno la persona se acercaba a la

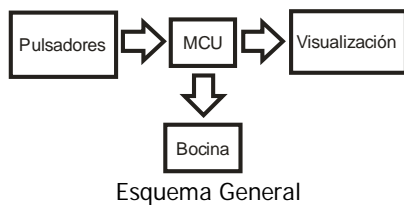
caja, o ventanilla, para realizar el trámite. Luego se fueron incorporando otras alternativas como ser, la persona que llegara sacaba de un aparato un número y se dedicaba a esperar, cuando le llegaba su turno, que podía ser indicado por el cajero, nombrando en voz alta, el número que seguía.

Hoy en día, este esquema de atención al público se puede mejorar en varios

aspectos, como son: la incorporación de un dispositivo electrónico que visualice el número del turno que espera ser atendido, de forma clara y en un lugar accesible a la vista por todos, el aviso por medio de un sonido que el turno acaba de cambiar, alertando a las personas que esperan ser atendidas, la incorporación del número de caja o ventanilla que esta libre lo cual agiliza la búsqueda de la caja desocupada.

Esquema general del sistema

En la imagen siguiente se observa un esquema general del sistema.



Está formado por el bloque de los pulsadores, que están conectados directamente al microcontrolador, la bocina, el microcontrolador en si y el bloque de visualización, el cual incorpora los display de siete segmentos que indican, en dos dígitos, el turno, y también, en un dígito, el de la caja o ventanilla.

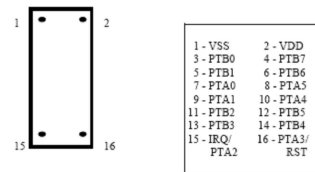
Kits utilizado

Para desarrollar el sistema los alumnos utilizan los kits de entrenamiento EVAL08QTY[1] que posee el laboratorio. De este modo los alumnos no necesitan adquirir el microcontrolador ya que el mismo kit ya lo incorpora. En la próxima figura se observa una imagen del kits.



Kits EVAL08QTY

Este kit incorpora los circuitos necesarios para programar el microcontrolador y la herramienta de debugger al mismo tiempo, lo cual es muy versátil a la hora de hacerlos prácticos, además de que incorpora un conector tipo "header" el cual conecta el kit con el protoboard, o placa de armado, del alumno para desarrollar los trabajos prácticos. La siguiente imagen muestra la distribución de pines del conector.



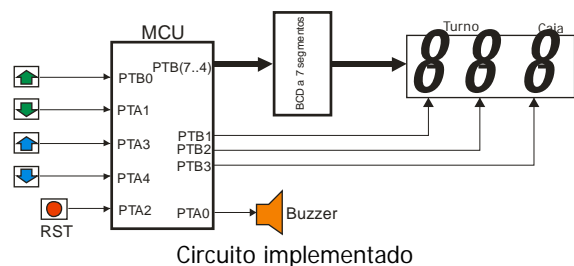
Conector del Kits

Es de destacar que el kit EVAL08QTY sirve para dos tipos de microcontroladores, y de ahí su nombre también, para los MC68HC908QTx y los MC68HC908QYx, mediante zócalos dispuestos en la placa se puede trabajar con uno u otro.

El trabajo práctico lo desarrollan con el microcontrolador MC68HC908QY4 [2] que es el que trae el kit EVAL08QTY.

Circuito Implementado

En la siguiente figura se observa un esquema con más detalle, del circuito implementado por los alumnos, para desarrollar el trabajo práctico.

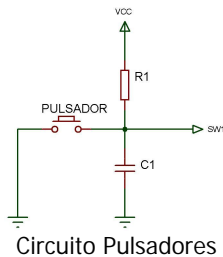


Circuito implementado

A continuación se describirán los circuitos para los pulsadores, el buzzer y los display.

Los pulsadores

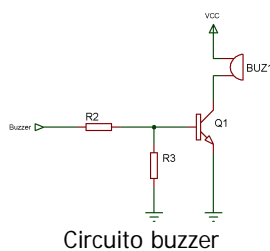
El circuito de los pulsadores se muestra en la próxima figura. Es de notar que el mismo circuito incorpora un capacitor, C1, para reducir el rebote del pulsador al ser presionado.



Esta configuración del pulsador la incorporan todos los pulsadores del sistema, ya sea, los de subir y bajar el turno como el de reset.

El buzzer

En la siguiente figura vemos el circuito correspondiente a la buzzer.

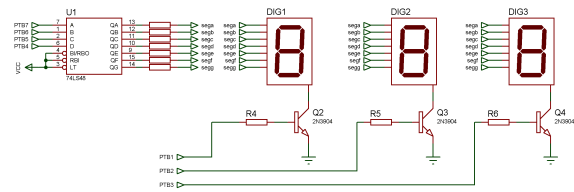


En el circuito se observa un divisor resistivo entre R2 y R3, el cual tiene como objetivo reducir la tensión aplicada a la base del transistor Q1. La alimentación del buzzer, VCC, dependerá del tipo de dispositivo que se utilice, habiendo disponibles en diversas tensiones.

Al utilizar este puerto para el buzzer, el PTA0, no se lo podrá hacer sonar; al estar conectado con el debug, debido a que éste utiliza este pin para comunicarse con el MCU.

Los display

En la siguiente figura se observa el circuito de los display.



Circuito de los display

Como se puede apreciar, la salida del dato se ingresa a un decodificador de BCD a 7 segmentos (que puede ser el integrado 74LS48), la salida ingresa a los display de siete segmentos cátodo común. Mediante otras tres salidas del microcontrolador, que son B1, B2 y B3, se activa el dígito correspondiente al dato que se quiere visualizar. Esta técnica (de multiplexado de los dígitos) se utiliza dado los pocos pines que dispone el microcontrolador.

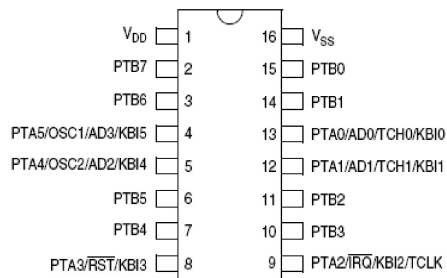
Para activar el dígito correspondiente se debe poner en alto (o uno lógico) el pin del puerto B, ya sea B1, B2 o B3. Esto es así dado que los display son de cátodo común y lo que hace el transistor es alimentarlos conectarlos a la referencia de cero volts o no. También hay que notar el tipo de decodificador utilizado. En este caso se planteó el 74LS48 cuyas salidas son activas en alto; lo cual se utilizan activando un display cátodo común.

El Microcontrolador

El microcontrolador utilizado por los alumnos en el trabajo práctico, como ya se mencionó es el MC68HC908QY4 de Freescale. Dicho dispositivo tiene las siguientes características, en cuanto a capacidad de memoria se refiere.

Dispositivo	Tamaño de la FLASH	Convertor A/D	Pines
MC68HC908QT1	1536 bytes	-	8 pines
MC68HC908QT2	1536 bytes	4ch, 8 bit	8 pines
MC68HC908QT4	4096 bytes	4ch, 8 bit	8 pines
MC68HC908QY1	1536 bytes	-	16 pines
MC68HC908QY2	1536 bytes	4ch, 8 bit	16 pines
MC68HC908QY4	4096 bytes	4ch, 8 bit	16 pines

Y la distribución de pines se muestra a continuación:



Distribución de pines del MCU

Las características más importantes se pueden resumir en las siguientes:

- Voltajes de operación de 5V y 3 V (VDD).
- 8 MHz bus interno a 5V, 4 MHz a 3V.
- Capacidad de auto-despertarse de STOP.
- Programación de FLASH en sistema.
- 128 Bytes de RAM.
- Hasta 13 líneas de I/O.
- 6 bits para interrupciones de teclado (KBI).
- Interrupción externa (IRQ).
- Mapeo en memoria de los registros de I/O.
- Modos de ahorro de energía STOP y WAIT.

El software

El software para desarrollar el programa del microcontrolador se desarrolla en CodeWarrior, provisto por la misma empresa Freescale fabricante de los dispositivos.

Éste es un entorno integrado de desarrollo (IDE) que permite programar en assembler, o en los lenguajes de C y C++. Es de destacar que para programar en lenguaje ensamblador no tiene limitaciones. La interfaz de programación se puede observar en la siguiente imagen:

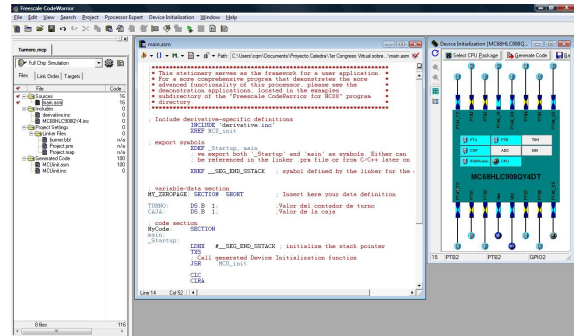
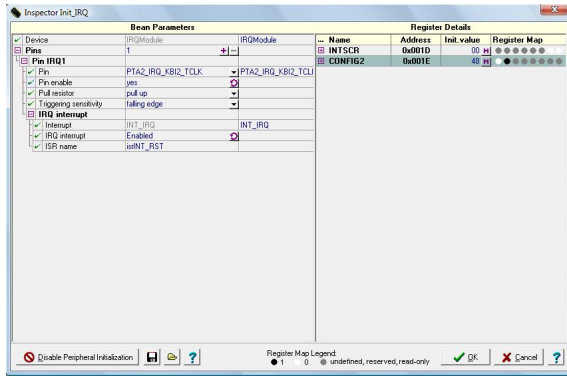


Imagen del proyecto en CodeWarrior

En el programa se tienen diferentes bloques. Como son:

- Inicialización del microcontrolador (función MCU_init).
- Lectura de teclado (función KBD_Read).
- Incrementar o disminuir el número del turno (funciones ADD_TURNO y SUB_TURNO).
- Borrado del número del turno (función isrINT_RST de la interrupción de IRQ).
- Refresco de los números en los display (función Refresh_DISP).
- Activación de la bocina (función Buzzer_ON)

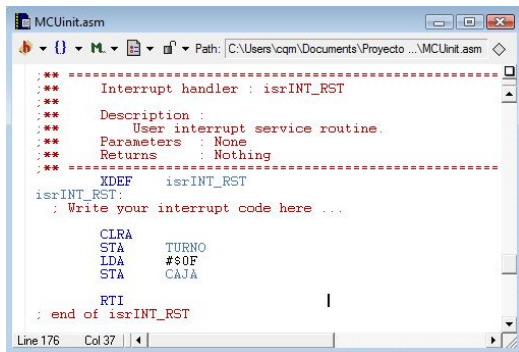
La función para inicializar el MCU, que como ya se dijo es la función denominada MCU_init, es generada automáticamente por el propio programa CodeWarrior. Lo hace a través de una interfaz gráfica, muy intuitiva, donde el alumno configura las distintas opciones del microcontrolador, como son: oscilador (tipo, frecuencia, etc.), pines de entradas y de salida, configuración de las interrupciones, etc., que como es de esperar, dependerá del tipo de dispositivo, se tendrán unas u otras opciones de configuración. A continuación se muestra la pantalla de configuración de las interrupciones:



Configuración de la interrupción IRQ

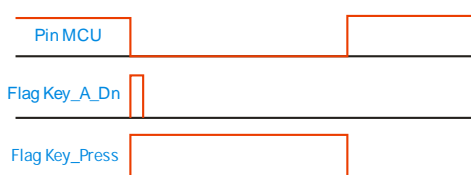
En este caso, la interrupción es la del pin IRQ (PTA2), la cual genera la función isrINT_RST que es para establecer en cero el indicador de turno y borra el indicador de caja.

A continuación se puede observar el código de la rutina de interrupción:



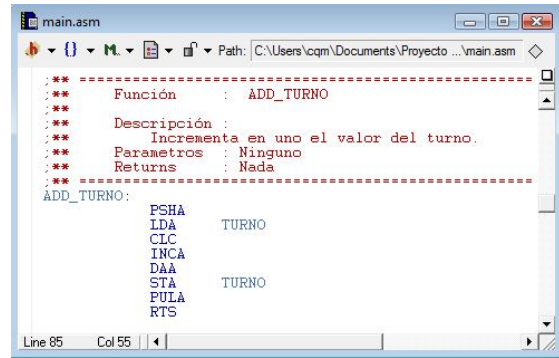
Código perteneciente a la interrupción

La función de lectura de teclado es un poco más extensa, por lo que no se escribirá el código completo, solo se dará un detalle de su funcionamiento. La detección de la tecla presionada se realiza con la ayuda de bits que son utilizados como banderas, indicando la ocurrencia de cierto evento. En el gráfico siguiente se muestra un diagrama de los valores que va adquiriendo el pin propio del MCU y los valores de las banderas.



Banderas para detectar el accionamiento de los pulsadores

La función para incrementar o disminuir el contador es muy similar, por eso que en la siguiente imagen se observa el código correspondiente al incremento del indicador de turno.



Función que incrementa el número de turno

Se observa que luego de incrementar el turno se hace el ajuste decimal del número.

Las funciones que siguen utilizan la interrupción del timer 1 como base de tiempo. Estas funciones son la del refresco de los display y la de activar el buzzer. La interrupción del timer se produce cada 10 milisegundos.

Para el primer caso, en cada interrupción se va colocando los datos de las decenas, unidades y del número de caja y se va activando el display que correspondiente.

En el caso del buzzer se lo activa y se lanza un timer de 1 segundo, al terminar el tiempo del timer se lo detiene y se apaga el buzzer.

Consideraciones finales

Con este proyecto se pretende que el alumno se inicie en la programación de este tipo de microcontroladores, es de destacar que en el plan de la cátedra ya los alumnos anteriormente a esta práctica han desarrollado trabajos prácticos con otras marcas de microcontroladores, por eso este trabajo tiene ciertas

características en cuanto al grado de dificultad del mismo.

Por otro lado, el indicador de turno, a la hora de montarlo en la realidad, se tendrían que hacer ciertas modificaciones, como son: el display, que este trabajo se considero uno de siete segmentos, debería ser uno realizado con LED, para tener un tamaño considerable y poder ser visualizado desde cierta distancia lejana del tablero. Otro elemento, y no menos importante, es que en este trabajo se conectaron los pulsadores directamente al microcontrolador. En la práctica esta pensado reemplazar los pulsadores con módulos de RF XBee para poder tener una mayor cantidad

de pulsadores, y por ende, una mayor cantidad de cajas para atender a los usuarios. En el caso de la utilización de módulos inalámbricos los pulsadores se pueden conectar directamente al módulo RF, y desde el MCU se recibiría el dato cada vez que se presione un pulsador.

Referencias

- [1] Kits didáctico EVAL08QTY
<http://www.electrocomponentes.com>
- [2] MC68HC908QY4 Data Sheet, Freescale, Documento: MC68HC908QY4.pdf