



SISTEMAS BASADOS EN MICROPROCESADOR
TRABAJO FINAL DE PRÁCTICAS
Grado en Ingeniería Electrónica Industrial (3º curso)
Curso 2023/2024

1 INTRODUCCIÓN

El presente documento recoge el trabajo final de prácticas de la asignatura de Sistemas Basados en Microprocesador de los estudios de Grado en Ingeniería Electrónica Industrial. El objetivo de este trabajo es reunir los aspectos tratados durante las prácticas de laboratorio y evaluar los avances de los alumnos. Este trabajo formará parte de la nota conforme se recoge en el apartado 3 y en la normativa de la asignatura.

2 ESTUDIO PRÁCTICO

2.1 Objetivos

El objetivo de este trabajo es el de desarrollar un programa mixto ensamblador y C que realice las siguientes tareas simultáneamente.

- Reproducción de sonidos mediante un zumbador piezoeléctrico.
- Edición mediante teclado matricial y por interrupciones de las teclas de un piano electrónico.
- Gestión de un display LCD.
- Gestión de teclas y leds del *Launchpad*.

2.2 Requisitos funcionales

El programa tendrá que correr en el *Launchpad MSP430FR6989* y utilizará los siguientes dispositivos de E/S:

- Teclado del *BOOST-IR* de Texas Instruments o compatible.
- Zumbador piezoeléctrico conectado conforme a lo recogido en el boletín de la práctica 7.

La aplicación gestionará un teclado de piano electrónico de tres octavas. Tendrá dos modos de funcionamiento: *piano* y *reproducción*.

En el primero las teclas de notas emularán una octava de un piano según la siguiente disposición:

| | | | |
|---|----|----|----|
| ⏪ | C# | D# | + |
| C | D | E | - |
| ⏩ | F# | G# | A# |
| F | G | A | B |

Serigrafía del teclado

| | | | |
|------------|---|---|---|
| <i>Bel</i> | C | D | + |
| c | d | e | - |
| <i>Int</i> | F | G | A |
| f | g | a | b |

Códigos ASCII de cada tecla

Mientras la tecla esté pulsada, sonará la nota correspondiente de la octava actual y se visualizará su nombre en la pantalla usando el mismo formato que en la práctica 7. Será posible cambiar la octava actual con las teclas '+' y '-'. La tecla *Bel* (código ASCII 7) apaga/enciende el sonido y la tecla *Intro* (código ASCII 13) permite el paso al modo de reproducción.

En el modo de reproducción sonará una melodía pregrabada y se mostrará en pantalla la octava y la nota que está sonando. Las únicas teclas disponibles durante la reproducción serán la `Bel` y la `Intro`. La primera apaga/enciende el sonido y la segunda pasa al modo de piano, en cuyo caso, se para la reproducción.

El programa empezará en modo piano, en la octava 5 y sonido activado.

Las teclas disponibles serán:

- `Bel` (código ASCII 7): Apaga/enciende el sonido. El icono ‘!’ del display LCD se encenderá cuando el sonido esté apagado.
- `Intro` (código ASCII 13): Permite el cambio de modo. Si está en modo piano, pasa al modo de reproducción y viceversa.
- ‘+’ y ‘-’: sólo disponibles en modo piano, incrementan o disminuyen respectivamente la octava actual. Sólo se permitirán octavas de la 4 a la 6. El valor resultante se visualizará en la pantalla con un mensaje “Ox ”, donde x es el nuevo número de octava.
- `Notas` (ver códigos ASCII en la figura de arriba): sólo disponible en modo piano, cuando se pulsa, reproduce un tono de esa nota para la octava actual y se muestra su nombre, junto con la octava, en la pantalla “Ox Not”, donde x es el número de octava y Not es el nombre de la nota con tres caracteres (rellene con espacios si es necesario). Cuando se suelta la tecla, cesa el sonido y se borra el nombre de la nota, permaneciendo el número de octava actual.

2.3 Requisitos estructurales

Se deberá entregar un proyecto mixto C/Ensamblador que gestione la temporización y la lectura de teclas por interrupciones, conforme a las directrices dadas en las prácticas de laboratorio. La generación de sonido se hará por *hardware*.

2.3.1 Sección ensamblador

Regirán las indicaciones de los boletines de prácticas y no deberían necesitarse cambios respecto del código de la práctica 7.

2.3.2 Sección C

Realizará la gestión descrita en el apartado 2.2. El código se recogerá en tres funciones: `Piano`, `Play` y `AnimaBat`.

El programa principal se encargará de las labores de inicialización y del superbucle. Las tareas cooperativas serán:

- `Piano`. Básicamente la misma de la práctica 7. Recoge la funcionalidad del piano, pero añadiendo las nuevas funcionalidades de este trabajo (paso a modo reproducción).
- `Play`. Reproduce una melodía pregrabada.
- `AnimaBat`. Realiza una de las posibles animaciones de los segmentos de batería. Inicialmente se hará la animación 1 a 8hz. Cada vez que se pulse el `switch1` del *Launchpad* se cambiará de animación. Cada vez que se pulse el `switch2`, se cambiará circularmente de frecuencia según la secuencia 4, 8, 16, 32 (hz). Las animaciones pueden ser:
 - ◆ Animación 1. Los cuadros de animación serán los siguientes. Una vez que se llegue al final, se repetirá desde el principio:
 - ◆ “I I”.
 - ◆ “ I I ”.

siendo T el tiempo base de la *redonda* y que estableceremos en 2 segundos (120 bpm).

La letra puede ser:

| Letra | Función | Dígito que le sigue |
|-------|------------------------------------------------------------------------------------|-----------------------------------------------------|
| 'o' | Establece octava actual. Al iniciar la reproducción, la octava por defecto es la 4 | Número de nueva octava. Valores posibles: '4' a '6' |
| 's' | Hace un silencio en la reproducción | Duración |
| 'c' | Reproduce la nota <i>Do</i> de la octava actual | Duración |
| 'C' | Reproduce la nota <i>Do#</i> de la octava actual | Duración |
| 'd' | Reproduce la nota <i>Re</i> de la octava actual | Duración |
| 'D' | Reproduce la nota <i>Re#</i> de la octava actual | Duración |
| 'e' | Reproduce la nota <i>Mi</i> de la octava actual | Duración |
| 'f' | Reproduce la nota <i>Fa</i> de la octava actual | Duración |
| 'F' | Reproduce la nota <i>Fa#</i> de la octava actual | Duración |
| 'g' | Reproduce la nota <i>Sol</i> de la octava actual | Duración |
| 'G' | Reproduce la nota <i>Sol#</i> de la octava actual | Duración |
| 'a' | Reproduce la nota <i>La</i> de la octava actual | Duración |
| 'A' | Reproduce la nota <i>La#</i> de la octava actual | Duración |
| 'b' | Reproduce la nota <i>Si</i> de la octava actual | Duración |

Por ejemplo, la cadena “c5d5e5f5g5a5b5o5c5d5e5f5g5a5b5o6c5d5e5f5g5a5b5” haría una escala de corcheas (duración T/8) por las tres octavas. Por tanto, debería durar $7*3*2/8=6$ segundos).

Play debe seguir la arquitectura de las tareas cooperativas para gestionar el modo de reproducción. La variable `ProxEjec` se ajustará para que sea igual al final de la nota actual (y principio de la siguiente). De esta manera, en cada instante de ejecución de `Play` se gestionará un par de letra-dígito. La variable `ProxEjec` será global para que se pueda iniciar la reproducción desde el momento actual o para salir de este modo al pulsar la tecla `Intro` en modo `Play`. Si se lee un par letra-dígito erróneos o si se alcanza el final de la cadena `partitura`, se apagará el sonido, se desactivará `Play`, se borrará la pantalla y se pasará al modo piano automáticamente. `Play` no hace gestión del teclado para salir del modo de reproducción o para activar/desactivar el sonido. Dicha labor se hace en `Piano`. Sin embargo, sí que visualiza en la pantalla en todo momento la octava y el nombre de la nota que se está reproduciendo. Las únicas teclas activas en el modo `Play` serán `Intro` y `Mute`.

Dado que se comienza en el modo piano, inicialmente se hará `ProxEjec=0xFFFFFFFF` para que no suene la melodía. Para entrar en modo reproducción bastará con cargar el instante actual en `ProxEjec` y la melodía en la variable `partitura`. `Play` gestionará la variable `partitura` para que apunte siempre al siguiente par letra-dígito. El cambio de octava en el modo `Play` no debe afectar a la octava activa en el modo piano. Es decir, se deberá respetar la octava activa en el modo `Piano` al volver del modo `Play`.

El contenido de una partitura de ejemplo se puede encontrar en el fichero `partitura.txt` disponible en la web de la asignatura.

2.4 Requisitos formales

Se deberá entregar un proyecto mixto C/Ensamblador que gestione los requisitos previamente indicados empleando las técnicas de multitarea cooperativa, máquina de estados, configuración de puertos y temporización y lectura de teclas por interrupciones tal y como se ha desarrollado tanto en las clases de teoría como en las prácticas.

En cualquier caso, ninguna tarea debe detener la ejecución de las demás salvo en lo referente al arranque/paro de `Play` desde `Piano`.

2.4.1 Sección ensamblador

Respecto de la parte ensamblador: se incluirán, al menos, los módulos desarrollados en prácticas correspondientes a los *drivers* de los dispositivos de E/S.

Podrán añadirse funciones extra de control de dispositivos en la parte correspondiente si se considera necesario.

Los alumnos con las prácticas convalidadas podrán usar las librerías de años anteriores cambiando/añadiendo aquellas partes que sean necesarias para cumplir con los requisitos de este curso. Por ejemplo, deberán desarrollar el módulo `snd.asm` de la práctica 7 de este curso.

2.4.2 Sección C

Se escribirá en C el código restante.

2.4.3 Documentación del trabajo

Se adjuntará al proyecto un documento de texto conteniendo la siguiente documentación:

- Relación alfabética de todas las variables usadas (en C y ensamblador) indicando propósito (indicando la interpretación de los valores), módulo en el que se define, la función que la escribe y la función o funciones que la leen.
- Relación alfabética de funciones utilizadas en el programa (en C y ensamblador) indicando propósito de la misma, el módulo en el que se codifica, parámetros de entrada (tipo e interpretación de los valores), parámetros de salida (tipo e interpretación de los valores), qué funciones que la llama y funciones que llama (si es el caso).

3 CRITERIOS DE PRESENTACIÓN Y VALORACIÓN

El trabajo final de prácticas se valorará entre 0 y 10 puntos. Para tener derecho a ser evaluado, es necesario haber obtenido una calificación de APTO en prácticas o en el examen de laboratorio y haber obtenido al menos un 4 en el examen teórico. La nota del trabajo, siempre que sea no inferior a 4 puntos, contribuirá al 40% de la nota de la asignatura. El 60% restante se obtendrá de la nota del examen teórico siempre que ésta sea mayor o igual a 4. La asignatura se considerará aprobada para notas no inferiores a 5. A no ser que se indique lo contrario y con carácter general, será de aplicación lo siguiente:

- Los trabajos se realizarán de manera individual. La violación de esta regla supondrá automáticamente la evaluación negativa (suspense 0) de los alumnos implicados.
- El proyecto CCS ha de tener un nombre diferente para cada alumno. Concretamente, el formato a seguir será: AABB_UVUS, donde “AABB” será sustituido por el curso académico, y “UVUS” será sustituido por el UVUS de cada alumno. Por ejemplo, para un alumno con UVUS (Usuario Virtual de la Universidad de Sevilla y el comienzo del correo universitario, sin @alum.us.es) nomapaapb, matriculado en el curso 23/24, deberá usar como nombre de proyecto 2324_nomapaapb. El **incumplimiento** de este requisito da lugar a la **calificación directa de suspense con 2’5 puntos**.
- "El fichero de documentación tendrá como nombre el nombre del proyecto, indicado en el punto anterior seguido de "_documentacion.pdf". Así, en el ejemplo dado se llamaría: "2324_nomapaapb_documentacion.pdf".
- Se valorará la presentación en general, el estilo de programación, el uso de comentarios, así como el grado de interés y atención mostrados por el alumno durante el desarrollo de las prácticas.
- Todos los ficheros fuentes deberán incluir una cabecera, a modo de comentario, como en el ejemplo siguiente (para fuentes en C, los comentarios empezarán con “//”):

```

;-----
; Apellido1 Apellido2, Nombre
; Grupo: Lx
;-----

```

- Los programas deberán estar libres de errores de compilación/ensamblado. El **incumplimiento** de este punto da lugar a la **calificación directa de suspense con 2.5 puntos**.
- Es obligatorio seguir la estructura de ficheros, nombres, etc. de las prácticas. La falta de alguna subrutina, el cambio de nombre, o la implementación de una diferente funcionalidad da lugar a la calificación de 0 en la contribución de la subrutina a la nota global del trabajo.
- Los alumnos podrán ser llamados a defender su trabajo si el profesor lo estima necesario.

3.1 Fecha y lugar de entrega

En la página de la plataforma de enseñanza virtual de la asignatura, sección "Contenido", hay una actividad titulada "Entrega trabajo práctico asignatura". En dicha actividad se deberá subir un fichero que contenga lo indicado en el apartado siguiente. Se permitirán entregar tantas versiones como se desee, corrigiéndose exclusivamente la última entregada. Fecha tope entrega: con carácter general, no después del día del examen teórico final de la asignatura. Cada día hábil de retraso en la entrega penalizará la nota con un punto sobre la calificación del trabajo. Pasados 5 días de la fecha tope de entrega, el trabajo no será evaluado y se considerará entregado para la siguiente convocatoria.

3.2 Contenido de la entrega

Un fichero comprimido (ZIP o RAR) en el que se encuentre la carpeta etiquetada con el convenio de nombres especificado en el apartado 3. Dentro de esa carpeta se incluirán todos los ficheros del proyecto y el de documentación del trabajo.

Control de versiones de este documento:

- v1.0: Versión original.
- v1.1: Se ha especificado mejor el punto 2.3.3 y se ha añadido la sección 2.4.