

Ficheros que contiene la práctica 4

E/S por interrupciones vectorizadas

ec

Ficheros que contiene la práctica 4

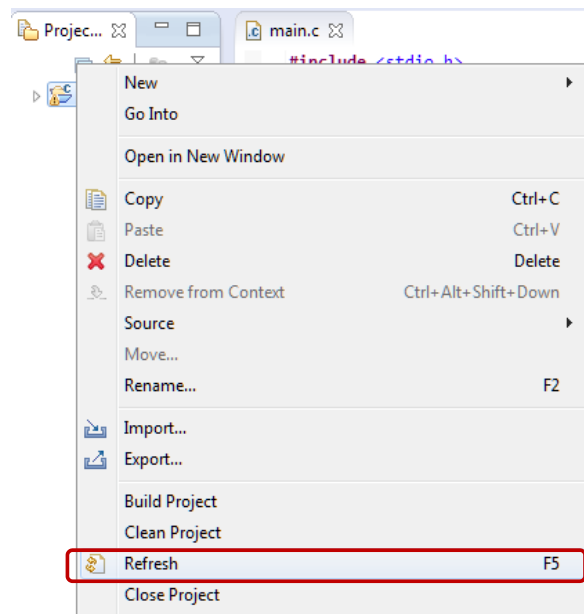
- Tenéis que **añadir** los siguientes **ficheros** que habéis hecho en la **práctica 3**
 - **main.c**
 - **init.S**
 - **gpio.h y gpio.c**: interfaz e implementación de funciones para el **manejo de los puertos B y G del controlador GPIO**
 - **leds.c y leds.h**: ficheros para implementar las funciones para el **manejo de los leds**
 - **D8Led.h y D8Led.c**: ficheros para implementar las funciones para el **manejo del display 8 segmentos**
 - **intcontroller.h e intcontroller.c**: ficheros para implementar las funciones que **manejan el controlador de interrupciones**
 - **timer.h y timer.c**: ficheros para implementar las funciones del módulo que **maneja los temporizadores**

ec

Para añadir los ficheros de la práctica 3



- 1- **Añadir** los ficheros al workspace **desde el explorador**
- 2- Abrir el proyecto con el Eclipse →ratón encima del proyecto → botón derecho → **refresh**

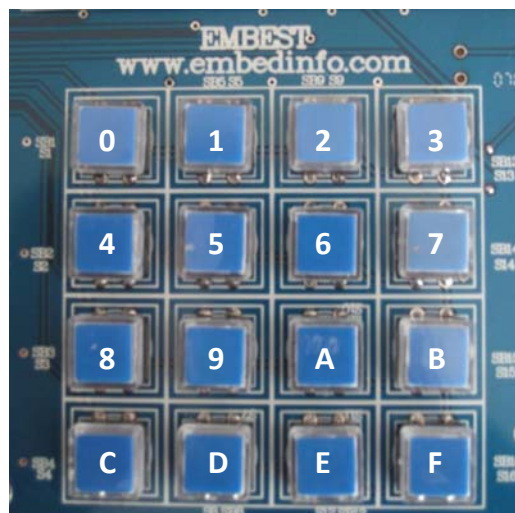


ec

Ficheros que contiene la práctica 4



- Keyboard.c**: fichero con el código para escanear la tecla del teclado matricial que ha sido pulsada
 - El alumno deberá **completar la función** `int kb_scan(void)`
 - Esta función identifica la tecla pulsada usando la técnica scanning y devuelve su valor



ec

Programa principal de la práctica 4



- **main.c**: fichero que habéis hecho en la práctica 3. Hay que hacer lo siguiente:
 - **Modificar** las funciones **timer_ISR** y **button_ISR** para que sean RTIs
 - **Codificar** la función **keyboard_ISR**
 - Añadir en el **setup()** la configuración del teclado

- **void timer_ISR (void)**
 - Función para tratar la interrupción del temporizador
- **void button_ISR (void)**
 - Función para tratar la interrupción de los pulsadores
- **void keyboard_ISR (void)**
 - Función para tratar la interrupción del teclado

Las 3 son RTIs, ya que las interrupciones se configuran como vectorizadas

IMPORTANTE

Al ser interrupciones vectorizadas **no existe** la subrutina **irq_IRS**
Eliminarla del fichero init.S

Programa principal de la práctica 4



- **void keyboard_ISR (void)**
 - Función para tratar la interrupción del teclado, sigue el siguiente esquema:
 - Esperar 20ms para eliminar los rebotes de presión
 - Escanear el teclado utilizando la función **int kb_scan (void)**
 - Poner en el display la tecla
 - Cambiar la configuración del timer 0 para que la generación de interrupciones periódicas tenga un periodo distinto en función de la tecla pulsada:
 - Tecla 0: periodo de 2s, valor de cuenta 62500 y divisor 1/8
 - Tecla 1: periodo de 1s, valor de cuenta 31250 y divisor 1/8
 - Tecla 2: periodo de 0.5s, valor de cuenta 15625 y divisor 1/8
 - Tecla 3: periodo de 0.25s, valor de cuenta 15625 y divisor ¼
 - Resto de teclas: no cambiaremos la configuración del timer
 - Esperar a que se deje de presionar la tecla leyendo el bit 1 del registro de datos del puerto G
 - Esperar 20ms para eliminar rebotes de depresión.
 - Borrar el flag de interrupción por la línea EINT1

Programa principal de la práctica 4



■ `int setup (void):`

- **Inicializar los leds** y configurarlos para que sean pines de salida
 - Utilizar las funciones definidas en `leds.c`
- **Inicializar el Display** 8 segmentos
- **Configurar los pulsadores:** Utilizar las funciones definidas en `gpio.c` para:
 - Activar interrupciones (EINT6 y EINT7)
 - Que la interrupción se detecte en flanco de bajada
 - Activar la resistencia pull-up
- **Configurar el teclado:** Utilizar las funciones definidas en `gpio.c` para:
 - Activar interrupciones (EINT1)
 - Que la interrupción se detecte en flanco de bajada
 - Activar la resistencia pull-up
- **Configurar el TIMER0** para que genere interrupciones periódicas (2s)
 - Utilizar las funciones definidas en `timer.c`
 - Factor de división: 1/ 8
 - Pre-escalado: 255
 - Inicializar la cuenta con 62500.
 - Valor de comparación puede ser cualquier valor mayor que 0 y menor que cuenta
- **Configurar el controlador de interrupciones** para que:
 - Active la línea IRQ en modo vectorizado
 - Deshabilite la línea FIQ
 - Configure las líneas TIMER0, EINT1 y EINT4567 por la línea IRQ del procesador y las deje habilitadas
- **Inicializar la tabla de direcciones de las subrutinas de tratamiento de interrupción**