

Bienvenido: [Ingresar](#)

location: [WebHome](#) / [TrabajosPracticos](#) / [PracticoASM6](#)



Trabajo Práctico Nro.6 Programación en la placa

Para los ejercicios en esta etapa deberemos utilizar lo siguiente

Archivos de Entrada

- **head.s** STARTUP, configurará el hardware de microcontrolador, asignará los valores iniciales de las variables globales que así lo requieran y le pasará el mando al main del C
- **main.c** es el archivo de arranque del C, contendrá el main() {} de nuestro programa.
- ***.c** según el práctico podrá haber algún archivo de funciones extra, los cuales serán llamado desde el C.

Archivo de configuración y estándar


- **lpc2114.h** Archivo de cabecera ( [lpc2114.h](#)), en esta etapa los proyectos generados deberán correr en un microcontrolador real, se agrega entonces el archivo de cabecera utilizado para describir a los registros del micro con nombres identificatorios de su función y no con simples números que indican posiciones de memoria
- **Makefile** el caso del makefile, se agrega el comando *grabar* para grabar el micro, posee también la variable SOURCE para asignar los fuentes del proyecto y TARGET para el nombre del archivo binario de salida.
- **lpc2114_flash.ld (Linker Script)** ( [lpc2114_flash.ld](#)) idem al práctico anterior.

Archivos de salida generados por make Para el caso ejemplo de que en Makefile se asigne *TARGET = prueba.hex*, serán los mismo que en el anterior ejercicio, sumándose los .o de cada archivo .c

- **prueba.elf** archivo binario.
- **prueba.hex** archivo binario formato Intel, usado para grabar el microcontrolador.
- **prueba.lst** archivo con el código fuente y el binario desensamblado.
- ***.o**


Ejercicio 1

Realizar un programa que prenda los led en alguna secuencia

Proyecto ( [led.zip](#))


Ejercicio 2

Realizar un programa que envíe por RS232 una cadena de caracteres.

Proyecto ( [uart.zip](#))

Ejercicio 3

Realizar un programa que envíe por RSR232 un contador, el mismo se incrementará cada 10ms por medio de Timer 0 y la FIQ.

Proyecto ( [fiq.zip](#))

Ejercicio 4

Realizar un programa que tome la entrada de un pulsador, cuente las cantidad de veces que se presiona la tecla y muestre el resultado a través de 2 display de 7 segmentos.

Nota:

- Toda salida del microcontrolador debe pasar por un buffer compatible con TTL.
- Se debe implementar un antirrebote en la tecla totalmente por software.
- El display mostrará un contador bcd de dos dígitos, el cual al llegar a 99 comenzará de nuevo desde 0. Proyecto (display.rar)

Ejercicio 5

Realizar un programa que lea 20 veces por segundo una entrada analógica realice un promedio de los últimos 10 y muestre el resultado en el display de 7 segmentos utilizado en el ejercicio 2, el fondo de escala del display (99), corresponderá al fondo de escala del ADC

Proyecto (adc.rar)

Ejercicio 6

Modificar el proyecto anterior, para que además de realizar la conversión, promedio y envío al display, se envíe por RS232 el promedio original (10 bits) con el número de canal donde fue tomado a una PC, la PC podrá también enviar al ARM el caracter '1','2','3' o '4', para modificar dicho canal.

Proyecto (uart_adc.rar)

Ejercicio 7

Rehacer el Práctico Nro 1, pero ahora implementar los retardo de los led mediante interrupciones y el timer 0, se propone un retardo de 500 ms

Proyecto ( led_irq.zip)

UntitledWiki: WebHome/TrabajosPracticos/PracticoASM6 (última edición 2012-09-27 13:34:00 efectuada por GuillermoSteiner)