

## Guia de Ejercitación

### Programación de la EDU-CIAA en lenguaje C (sin RTOS)

#### Objetivos

- Estudiar el hardware de la EDU-CIAA-NXP.
- Analizar características de la arquitectura ARM Cortex M4 sobre el microcontrolador LPC4337.
- Desarrollar experiencia en la instalación y uso del CIAA-IDE.

## 1. Introducción

La presente guía de ejercitación está desarrollada para trabajar sobre el hardware de EDU-CIAA NXP. Esta placa es una versión de bajo costo de la CIAA-NXP pensada para la enseñanza universitaria, terciaria y secundaria.

La documentación relacionada al diseño de la EDU-CIAA NXP se encuentra disponible en el sitio del proyecto accesible a través del siguiente link.

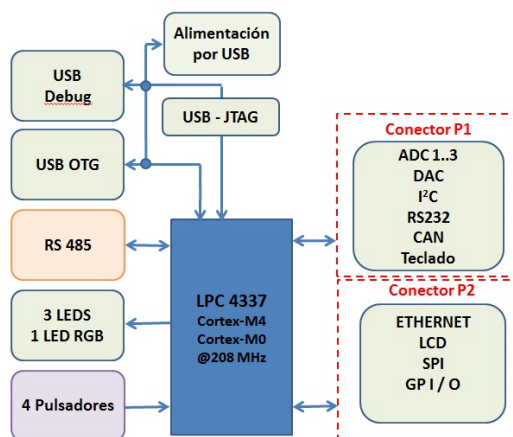


Figura 1: Diagrama en Bloques de la EDU-CIAA NXP. Fuente: [www.proyecto-ciaa.com.ar](http://www.proyecto-ciaa.com.ar)

El hardware se basa en el microcontrolador LPC4337 de la firma NXP. La distribución de pines se encuentra detallada en el documento asignación de pines.

## 2. Manejo de Puertos I/O

Dentro de los dispositivos disponibles en el hardware de la placa, se encuentran tres leds, un led RGB y cuatro teclas. Es recomendable realizar un driver de dispositivo que permita configurar y acceder los puertos de entrada/salida relacionados con los mismos.

Para la resolución de las consignas que se presentan a continuación se pide que diseñe e implemente sus propios drivers, basado en las funciones GPIO de la biblioteca LPCOpen, para el manejo de los dispositivos mencionados. Los documentos *Manejo de Puertos de Entrada y Manejo de Puertos de Salida* provee un resumen de las funciones necesarias.

Deberá realizar entonces el código y las definiciones necesarias en los siguientes archivos:

- Driver leds: `led.c`, `led.h`
- Driver teclas: `teclas.c`, `teclas.h`

El driver del leds deberá contar con funciones para:

- Inicializar los leds. Ejemplo: `void InicializarLeds(void);`
- Encender un led recibiendo como parámetro el LED al que se refiere<sup>1</sup>.
- Apagar un led recibiendo como parámetro el LED al que se refiere.
- Cambiar el estado de un led recibiendo como parámetro el LED al que se refiere.

El driver de teclas deberá contener funciones para:

- Inicializar los leds. Ejemplo: `void InicializarTeclas(void);`
- Detectar tecla presionada recibiendo como parámetro la tecla a consultar, si se usa antirrebote y liberación de tecla.
- Escanear teclado. Ejemplo: `uint8_t kbhit(void);`

## 2.1. Consignas

- a) Diseñe e implemente un firmware sobre la EDU-CIAA que haga parpadear el led amarillo con un periodo que permita visualizar el proceso. Emplear retardo por software.
- b) Diseñe e implemente un firmware sobre la EDU-CIAA que haga parpadear un led con un periodo que permita visualizar el proceso. Mediante las cuatro teclas disponibles se debe poder seleccionar el led activo. Emplear retardo por software.

## 3. Temporizador e Interrupciones

Tanto para el empleo de los temporizadores como para el manejo de interrupciones, se emplearán los drivers provistos por la biblioteca LPCOpen.

El documento *Manejo de Timer e Interrupciones* provee un resumen de las funciones necesarias.

### 3.1. Consignas

- a) Diseñe e implemente un firmware sobre la EDU-CIAA que encienda de a un led por vez y de manera secuencial. El tiempo de encendido de cada led será 250ms. Se deberá temporizar mediante interrupciones sin usar funciones de retardo por software.
- b) Diseñe e implemente un firmware sobre la EDU-CIAA que haga parpadear un led con un periodo de 250 ms. El sistema debe permitir seleccionar uno de entre 4 de los leds disponibles empleando una tecla para cada led.
  - Tec 1: Selecciona LED RGB (uno de los tres colores)
  - Tec 2: Selecciona LED 1.

---

<sup>1</sup> **Nota:** Tener en cuenta que al controlar el LED RGB, se deberá especificar además el color al cual se refiere.

- Tec 3: Selecciona LED 2.
- Tec 4: Selecciona LED 3.

c) **Consigna 2.3:**

Incorpore al ejercicio anterior la funcionalidad de variar el periodo de parpadeo del led activo.

- Tec 1: Selecciona el LED a la izquierda del actual.
- Tec 2: Selecciona LED a la derecha del actual.
- Tec 3: Disminuye el periodo de parpadeo.
- Tec 4: Aumenta el periodo de parpadeo.

## 4. Generación de Señales Analógicas (D/A)

Para el manejo de módulo AD-DA se empleará el drivers provisto por la biblioteca LPCOpen. El documento *Manejo de ADC* provee un resumen de las funciones necesarias.

- **Consigna 3.1:** Diseñe e implemente un firmware sobre la EDU-CIAA que genera una señal tipo diente de sierra de periodo 100 ms y excursión de 0 a 3V.
- **Cosigna 3.2:** Incorpore al ejercicio anterior la funcionalidad de variar el periodo y la amplitud de la señal.
  - Tec 1: Aumenta la amplitud de la señal.
  - Tec 2: Disminuye la amplitud de la señal.
  - Tec 3: Aumenta el periodo de la señal.
  - Tec 4: disminuye el periodo de la señal.

## 5. Adquisición de datos

### 5.1. Consignas

- a) Diseñe e implemente un Firmware sobre la EDU-CIAA que permita adquirir una señal analógica de excursión entre 0 y 3.3V, presente en el pin CH1 del conector P1. El sistema debe encender el led rojo si la señal toma su valor máximo (¿1020 d) y led verde si la señal toma su valor mínimo (¿5 d). Puede resolverlo por pooling.
- b) Incorpore al ejercicio anterior la funcionalidad de variar los umbrales máximo y mínimo, y ajuste la frecuencia de conversión en 100Hz empleando timer e interrupciones. La lectura del conversor, también debe hacerse por interrupciones.
- Tec 1: Aumenta el valor del umbral.
  - Tec 2: Disminuye el valor del umbral.

## 6. Transmisión de datos adquiridos a través del Puerto Serie

### 6.1. Consignas

- a) Diseñe e implemente un firmware sobre la EDU-CIAA que envíe por el puerto serie la cadena *Hola Mundo* a tasa de transferencia de 9600 baudios. La cadena debe enviarse cada vez que el usuario presiona la tecla 1.
- b) Diseñe e implemente un firmware sobre la EDU-CIAA que lea el puerto serie a la espera del caracter ascii 'a'. En respuesta el sistema debe enviar por el mismo puerto la cadena "Hola Mundo" cambiar el estado del LED 2 de la placa.

- c) Diseñe e implemente un firmware sobre la EDU-CIAA que lea el puerto serie a la espera del caracteres ascii. El sistema solo responderá a los caracteres 'a', 'r' y 'v' según el siguiente detalle:
- Caracter 'a': Se debe cambiar el estado del LED 1 y enviar por el puerto serie la cadena: *Hola Mundo*
  - Caracter 'r': Se debe cambiar el estado del LED 2 y enviar por el puerto serie la cadena: *Hola Mundo*
  - Caracter 'v': Se debe cambiar el estado del LED 3 y enviar por el puerto serie la cadena: *Hola Mundo*
- d) Diseñe e implemente un firmware sobre la EDU-CIAA que envíe por el puerto serie, cada vez que se presiona la tecla 1, el valor de un contador de 256 cuentas. El valor del contador debe modificarse mediante las teclas 4 y 5.
- e) Diseñe e implemente un firmware sobre la EDU-CIAA que permita adquirir una señal analógica de excursión entre 0 y 3.3V, presente en el CH1. El sistema debe enviar por el puerto serie una cadena de caracteres con el valor en decimal del dato convertido.

## 7. Manejo de periféricos con POSIX

Repita las consignas 2.1.a y 2.1.b usando las funciones POSIX definidas en el Firmware de la CIAA.

## 8. Ejercicios Integradores

Realice un firmware sobre la EDU-CIAA, que permita adquirir una señal analógica acondicionada entre 0 y 3,3 V disponible en el pin CH1 de P1 a una frecuencia de 100 hz.

La señal se debe reproducir por el DA, luego de algún procesamiento, por ejemplo ganancia, filtrado, recorte, etc. El procesamiento debe controlarse mediante comandos provenientes del puerto serie que permiten encender, apagar y modificar cada una de las rutinas de procesamiento disponibles.

Nota: Link Calculo de Filtros.