Instituto Tecnológico de Costa Rica **Área Académica de Ingeniería en Computadores** (Computer Engineering Academic Area) Programa de Licenciatura en Ingeniería en Computadores (Licentiate Degree Program in Computer Engineering) **Curso: CE-5303 Introducción a los Sistemas Embebidos** (Course: CE-5303 Introduction to Embedded Systems) Profesor: M.Sc. Ing. Jeferson González Gómez. (Professor) Semestre:II, 2019 (Semester)	**Examen Final** (Final Exam) Fecha: 8 de noviembre de 2019 (Date) Grupo: 1 Valor: 50 pts. (Value: 50 pts.) **Puntos obtenidos**:_____ (Score) **Nota**:_____ (Percentage)

Nombre de estudiante: Fabián Astorga Cerdas.
(Student's full name)
Carné: 2014040808.
(Student's ID)
Nombre de estudiante: Ernesto Ulate Ramírez.
(Student's full name)
Carné: 2014092260.
(Student's ID)

INSTRUCCIONES GENERALES.

- Trabaje individual o en parejas.
- La modalidad de entrega de este examen es a través del TecDigital.
- Cada estudiante debe subir una solución a este examen, aunque el trabajo se realice en parejas.
- Adjunte este enunciado como parte de su solución.

1. I Parte. Respuesta breve. 10 puntos

A continuación, se le presentan 10 preguntas de respuesta corta. Responda cada una de ellas de manera correcta. (1 punto c/u)

1. Mencione 2 características que deben tener los sistemas operativos para sistemas embebidos.
2. Lucía trata de portear el paquete alsa-utils para raspberry pi zero, utilizando Yocto. Al realizar el comando bitbake rpi-basic-image, tiene el siguiente error: "Nothing provides alsa-utils". ¿Cómo podría solucionarle el error a Lucía? (ya Lucía se aseguró que el nombre del paquete está bien escrito)
3. ¿A qué comando o acción, en espacio de usuario, corresponde el método module_open(struct inode *inode, struct file *filp)?
4. Escriba 2 características que deben tener los Modelos de Computación para describir el comportamiento de sistemas embebidos.
5. En el comando $(CC) -o this isright.cpp $-I../include - L../lib - lsomething$ ¿Cuál es el nombre de archivo completo de la biblioteca dinamica enlazada?
6. Escriba un comando (bash) que escriba en un archivo *dir.log* una lista de archivos y subdirectorios de un directorio actual.
7. Mencione dos características que tienen que tener los modelos de computación (MoC) para modelar Sistemas Embebidos?
8. ¿Qué es un toolchain?

9. Al tratar de compilar un código fuente, Juanito tiene el siguiente error "undefined reference to 'sqrt' ". Juanito dice que "ya incluyó math.h", pero el error no se corrige. ¿Cómo solucionaría el problema que tiene Juanito?

10. ¿Cuál es la diferencia principal un driver en espacio de kernel y uno en espacio de usuario?

2. II parte. Desarrollo (40 puntos)

A continuación, se le presentan 2 problemas de desarrollo, los cuales se deben contestar con el mayor nivel de detalle posible, mostrando todos los pasos requeridos para llegar a la solución. Se deberá enviar un archivo con las respuestas de los ejercicios, así como el archivo de programación correspondiente. Puede trabajar individual o en parejas si así lo desea. De ser en parejas, debe asegurarse que el documento lleve el nombre y carné de ambos integrantes.

1. El código exam.c presenta una descripción del comportamiento de un sistema embebido a diseñar, para la adquisición, filtrado y procesamiento de una señal analógica de sonido. A partir de dicha descripción:

   a) Realice un modelo (gráfico) de alto nivel del código utilizando máquinas de estado-proceso (PSM). No es necesario que muestre la descripción totalmente funcional de cada proceso, pero sí debe describir qué hace en alto nivel. Represente adecuadamente elementos de jerarquía y concurrencia (5 puntos)

   b) Muestre detalladamente los pasos de una síntesis manual, para obtener un modelo estructural de un sistema que pueda implementar dicha aplicación. Asuma que seguirá una metodología de diseño basado en plataforma, para un dispositivo Altera Cyclone® V SE 5CSEMA4U23C6N (ver Altera De0-Nano-SoC, como referencia). Puede utilizar componentes básicos de sistema, para dicha descripción (memorias, buses, procesadores, sistemas operativos, bloques de IP, HW específico, etc), siempre y cuando estén disponibles de una forma u otra en el dispositivo. Muestre el diagrama del modelo estructural del sistema obtenido luego de la síntesis. Justifique detalladamente toda decisión de de diseño. (20 puntos)

2. Parte del proceso de encodificación JPEG para frames de imágenes en sistemas embebidos, corresponde al algoritmo de run-zero encoding (RZE). En este tipo de encodificación, ante un arreglo de elementos discretos, proveniente de un FIFO, se deberá generar un arreglo salida que contenga, para cada número natural x (no incluye al 0), la cantidad de ceros que haya hasta x, seguido del número entero x correspondiente.

   Por ejemplo,

   $in = [0, 0, 1, 0, 3] \rightarrow out = [2, 1, 1, 3]$

   $in = [1, 2, 3] \rightarrow out = [0, 1, 0, 2, 0, 3]$

   $in = [0, 0, 0, -1, 255, 3] \rightarrow out = [3, -1, 0, 255, 0, 3]$

   Realice un modelado utilizando redes de procesos de Kahnd el algoritmo de **encodificación** RZE (mostrado arriba), es decir que ante una entrada del algoritmo mostrado arriba (in), produzca en arreglo de salida correspondiente (out). Incluya en su modelo una representación gráfica (5 puntos), una descripción de cada proceso (5 puntos) y una representación ejecutable descrita en Python (adjuntar archivo .py)completamente funcional del algoritmo.(5 puntos)

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores

(Licentiate Degree Program in Computer Engineering)

Curso: CE-5303 Introducción a los Sistemas Embebidos

(Course: CE-5303 Introduction to Embedded Systems)

Examen

(Exam)

Realizado por:

Made by:

Fabián Astorga Cerdas - 2014040808

Ernesto Ulate Ramírez - 2014092260

Profesor:

(Professor)

Jeferson González Gómez

Fecha: Cartago, 13 de Noviembre, 2019

(Date: Cartago, November 13th, 2019)

1.

   a. Los sistemas operativos para sistemas embebidos se caracterizan por contener únicamente los ejecutables y dependencias necesarios para hacer funcionar el dispositivo. Los desarrolladores se encargan de configurar las herramientas que compilan sistemas operativos, como yocto, para que solo instalen los programas y bibliotecas mínimos junto al kernel.
   b. También estos sistemas operativos, deben tener la capacidad de proveer seguridad y estabilidad en la ejecución sus procesos. Para un sistema embebido, se desea que este pueda ejecutar su acción objetivo todo el tiempo que sea necesario sin perder mucho tiempo en interrupciones por mantenimiento.

2. Lucía necesita crear la receta de alsa-utils con las dependencias correspondientes. Escribió bien el nombre en el archivo local.conf, pero no se aseguró de configurar la capa en el archivo bblayers.conf que contiene la receta de dicho paquete.

3. El comando *fopen* destinado a espacio de usuario ejecuta la acción contenida en *module_open*.

4.

   a. Para poder describir el comportamiento de un sistema embebido, es necesario contemplar la cantidad de elementos de procesamiento (PE) que ejecutarán. Así mismo debe incluir consideraciones con respecto a la concurrencia y segmentación de dichos PE.
   b. También es necesario determinar la forma de interacción entre hardware y software. En este modelo debería especificarse los métodos de sincronización y canales respectivos para poder obtener un comportamiento predecible del sistema.

5. El nombre completo de la biblioteca estática es: *libsomething.so*.

6. El comando en bash sería:

```
ls > dir.log
```

7.
    a. Típicamente un modelo de computación está representado de una manera formal, utilizando por ejemplo funciones matemáticas, notaciones e incluso teoremas.
    b. Un modelo de computación debe estar conformado por componentes como elementos y funciones, y los cuales tienen que comunicarse entre sí. El modelo se encarga de describirlos y definir los métodos para la comunicación.

8. Toolchain:
    a. Conjunto de programas informáticos y herramientas utilizados para crear o desarrollar otro producto, normalmente otro programa. Es una cadena, ya que la salida de un programa es la entrada del siguiente programa en la secuencia.

9. A Juanito se le olvidó enlazar la biblioteca al compilador agregando la bandera -lm, para que en tiempo de compilación el compilador pueda encontrar las definiciones en math.h, como por ejemplo la definición de *sqrt.*

10. La diferencia principal entre drivers en espacio de usuario y espacio de kernel, es que para los driver en espacio de kernel es necesario realizar unas tareas preliminares para poder ejecutarlo. Entre ellas se encuentran la reservación de espacios de memoria, definición de puertos de entrada y salida o incluso interrupciones. Estas tareas deben ser desarrolladas por el programador explícitamente y asociarlos a los métodos que establezca el kernel (module_init y module_exit en linux).

Problema 1

A. Máquina de estados-proceso (PSM)

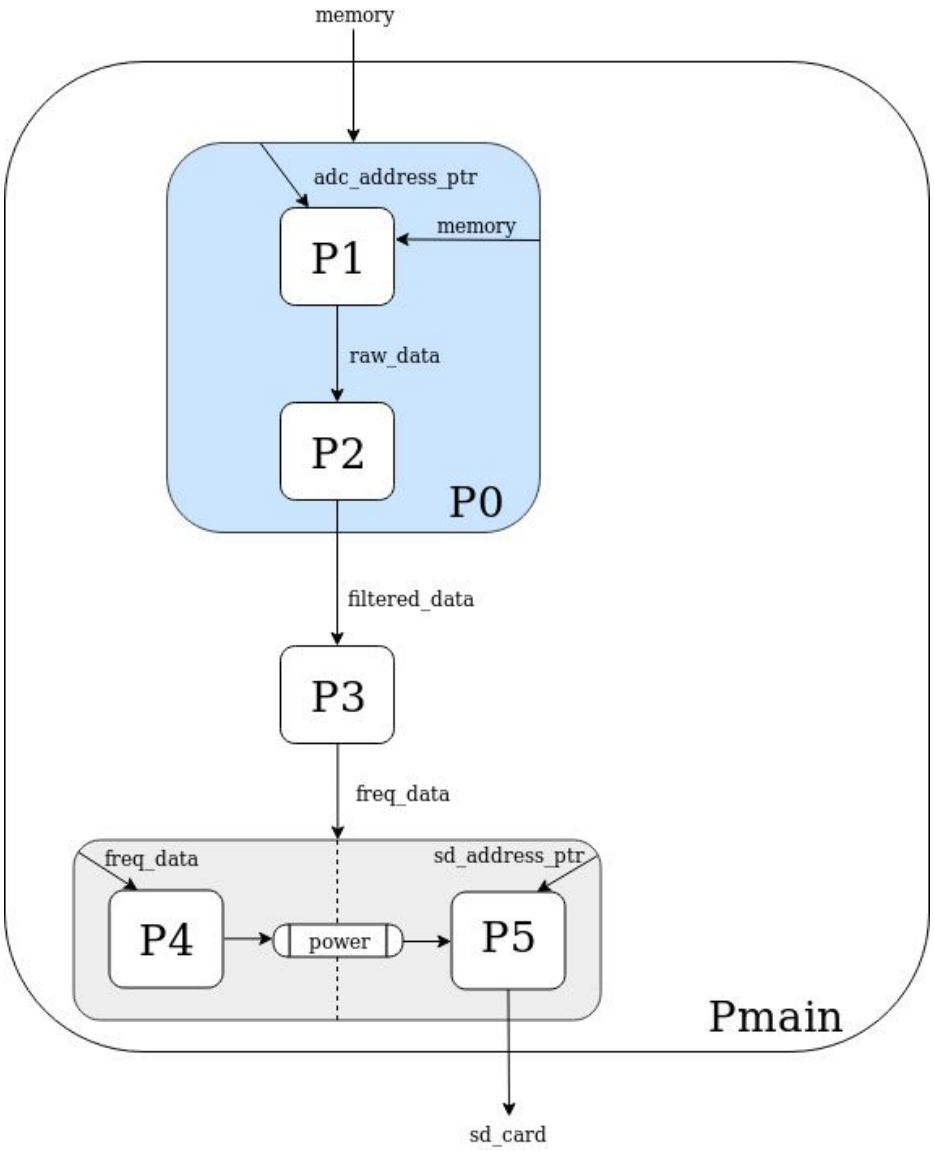

Figura 1. Diagrama PSM para la adquisición, filtrado y procesamiento de una señal análógica de sonido

B. Realización de la síntesis manual.

- Comportamiento de programa
  - P0:
    - Crea los hilos para los 2 procesos necesarios para la lectura de datos.
  - P1:
    - Declaración de sentencia de iteración for por 1024 ciclos.
      - Se realizan 1024 asignaciones de memoria para raw_data.
      - Se leen 1024 datos del módulo ADC.
      - Se realizan 1024 comparaciones y sleeps para la sincronización.
  - P2:
    - Declaración de sentencia de iteración for por 1019 ciclos.
      - Declaración de sentencia de iteración *for* por 5 ciclos.
        - Lee 5 datos de memoria (raw_data).
        - Escribe 5 datos en memoria (filtered_data).
  - P3:
    - Asignación de memoria para W.
    - Se escribe en W 2 datos.
    - Declaración de sentencia for por 512 ciclos.
      - Escribe en W por 1024 veces.
      - Se asignan 2 variables en memoria. (n, a)
      - Declaración de sentencia for por 10 ciclos.
        - Declaración de sentencia for por 1024 ciclos.
          - Se asignan 2 variables en memoria. (temp, Temp)
          - Se escribe en freq_data 1024 veces.
  - P4:
    - Sentencia *for* para 1024 ciclos.
      - Realiza 2048 operaciones lógicas
      - Realiza 4196 operaciones matemáticas.
  - P5:
    - Sentencia de iteración *for* por 1024 ciclos.
      - 2048 comparaciones entre variables y para finalización de bucle.
      - Lee 1024 veces en arreglo power.

- Escribe 1024 veces en memoria SD.

- Perfilado y estimación.
  - Perfilado:
    - Frecuencia de operaciones:
      - P1:
        - 1024 sumas y comparaciones.
      - P2:
        - 5095 sumas y comparaciones.
        - 1019 divisiones.
      - P3:
        - 83889159 operaciones matemáticas y comparaciones.
      - P4:
        - 3075 operaciones lógicas y matemáticas.
      - P5:
        - 2048 operaciones lógicas.
        - 1024 operaciones matemáticas.
    - Transferencias de buses:
      - Según las arquitectura donde se desea trabajar el dispositivo, toda transferencia sería por buses de 32 bits.
    - Llamadas a funciones:
      - P1:
        - 1024 llamadas al módulo ADC.
      - P3:
        - 1 llamada a malloc.
        - 1 llamada a sizeof.
        - 1 llamada a polar.
        - 512 llamadas a pow.
      - P4:
        - 1024 llamadas a sqrt.
        - 2048 llamadas a pow.

        - Accesos a memoria:
            - Cada vez que se realizan las acciones de leer o escribir datos en memoria, se realizan accesos de manera secuencial.
    - Estimación:
        - Desempeño: El diseño del sistema cuenta con alto desempeño respecto a sus recursos. Además, se espera que el consumo de potencia esté optimizado.
        - Costo:
            - El sistema, por sus características, se esperaría que tenga un costo reducido en comparación a utilizar un computador de propósito más general.
            - Esto se debe a la optimización en la utilización de los recursos y en no desperdiciar en elementos sobrantes.
        - Consumo de potencia: Se debe tener conocimiento del consumo de potencia de cada componente del sistema, con el fin de verificar con precisión si existe desperdicio de energía.
- Selección de componentes:
    - Memoria:
        - Memoria de instrucciones:
            - 83902444 instrucciones de operaciones lógicas o matemáticas + 23662 instrucciones de control. Para lo que se requiere una memoria de aproximadamente 20.5kB, por ende se elige una memoria de 32kB.
        - Memoria de datos:
            - Se necesitan alrededor de 20kB + 4kB para poder almacenar las variables del programa. Por razones de estandarización, se establece como requisito una memoria de 32kB.
    - Buses de datos:
        - Buses de 32 bits.
    - Procesador:
        - Nios II.
    - Sistema operativo:

        - Por motivos de calendarización de los procesos, se podría utilizar un sistema operativo para que se encargue de manejar los problemas de concurrencia.
    - ➢ Bloques de IP:
        - Módulo de ADC, SD Card Socket.
    - ➢ Hardware específico:
        - Conexión electrónica con el módulo ADC.

- ❖ Mapeo de memoria

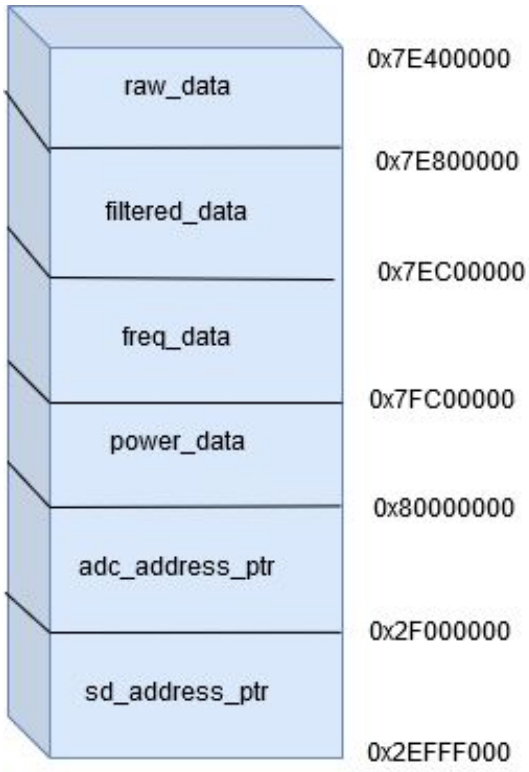

Figura 2. Diagrama de mapeo de memoria

- ❖ Asignación de memoria
    - ➢ P1: Lectura de memoria del ADC. El módulo guarda en un sector determinado de la memoria gracias al hardware específico.
    - ➢ P2: Filtrado.
    - ➢ P3: Aplicar la transformada de fourier a los datos filtrados.
    - ➢ P4: Estimación del espectro de potencia con los datos transformados por P3.

- ➢ P5: Log de información generada por P3 en memoria SD externa conectada al sistema.

- ❖ Asignación de canales o buses:
  - ➢ SPI: Comunicación serial entre componentes.
  - ➢ I2C: Comunicación entre ADC y controlador de memoria.
- ❖ Calendarización de procesos:
  - ➢ La calendarización de los procesos la determina el sistema operativo encargado de administrar el sistema.
- ❖ Inserción de interfaces:
  - ➢ Módulos de interfaz:
    - ■ Módulo ADC capaz de escribir datos en un espacio de memoria especificado mediante un controlador específico.
  - ➢ Driver de dispositivos:
    - ■ Especificación necesaria para la correcta comunicación con el módulo ADC. Este driver se debería encargar de obtener los datos y distribuirlos digitalmente al proceso determinado.
  - ➢ Controladores de memoria:
    - ■ Controladores provistos por la Altera Cyclone V.
- ❖ Refinamiento:
  - ➢ Este consta de un proceso de simulación y perfilado del sistema completo.
  - ➢ Optimizar recursos para evitar desperdicios de estos.
- ❖ Diagrama:

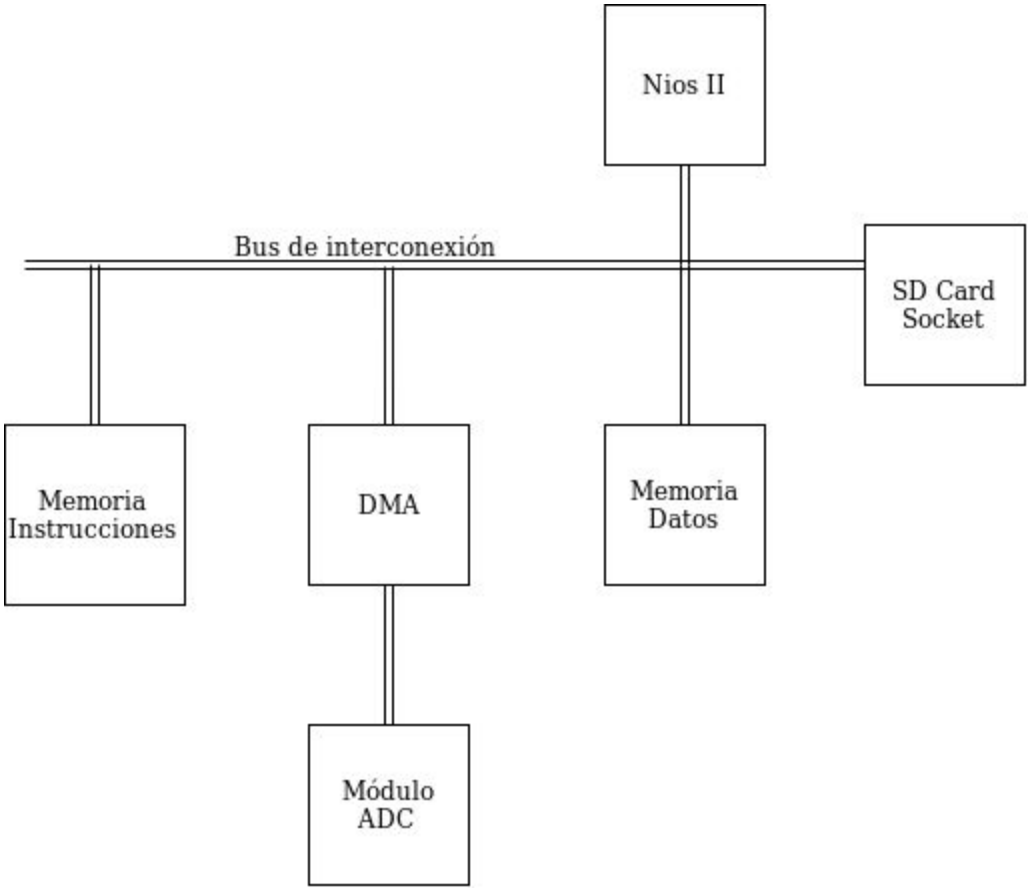

Figura 3. Diagrama de bloques.

Problema 2

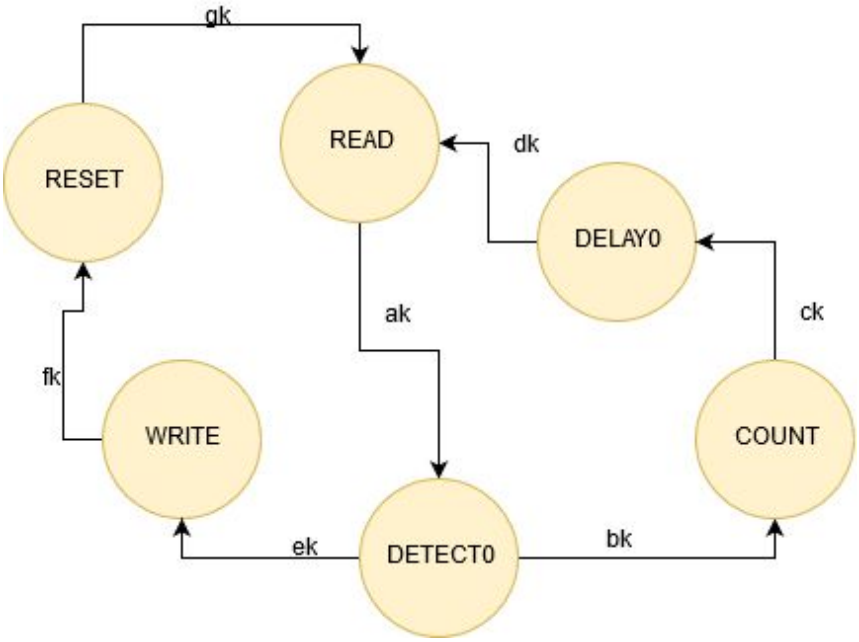

Figura 4. Diagrama de modelado de encodificación RZE

Tabla 1. Representación de procesos del modelo de encodificación RZE

Proceso	Descripción
READ	Lee un dato del arreglo de entrada.
DETECT0	Verifica si el dato leído es igual a cero.
COUNT	Aumenta en uno el contador de ceros.
DELAY0	Espera un ciclo para mantener el modelo síncrono.
WRITE	Escribe un dato en el arreglo de salida.
RESET	Vuelve a iniciar el contador de ceros, es decir, se iguala a cero.