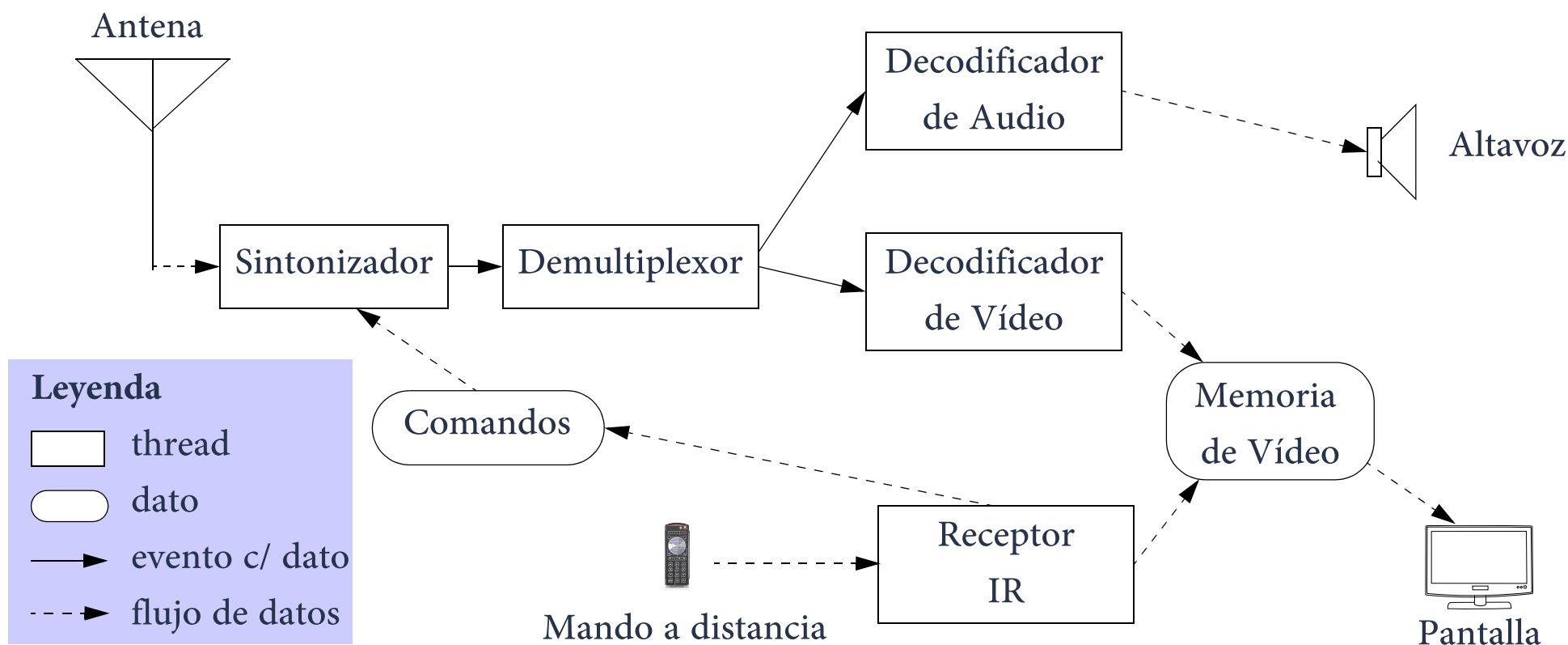


Desarrollo de software para sistemas empotrados

Examen Febrero 2022

Introducción

El objetivo de este ejercicio es realizar el análisis y diseño arquitectónico de una parte de un televisor digital, que se corresponde con el siguiente diagrama de bloques:



Descripción general

El sistema realizará las siguientes acciones en software:

- Sintonizador, demultiplexor, decodificador de audio y decodificador de vídeo
 - Procesan la señal de audio y vídeo que proviene de la antena. Se ejecutan en secuencia, tal como se muestra en el diagrama.
 - Se comunican mediante eventos con datos.
 - La primera actividad de la secuencia, el sintonizador, es periódica y funciona a 50 Hz.
 - Las siguientes son aperiódicas, aunque como se activan desde las anteriores al final ejecutarán también a 50 Hz.
- Receptor IR (de infrarrojos)
 - Funciona periódicamente a 125 Hz, comprobando si se reciben señales del mando a distancia, poniendo imágenes en la memoria de vídeo y escribiendo comandos para el sintonizador.

Objetos de datos

La memoria de vídeo

- No requiere exclusión mutua
- Escriben en ella el decodificador de vídeo y el receptor IR
- La pantalla accede a ella directamente por medios electrónicos, sin software

Los comandos

- Se representan mediante un dato de tipo entero que al ser de lectura/escritura atómica no requiere exclusión mutua

No hay funciones específicas para acceder a ninguno de estos datos

Software funcional

La funcionalidad del software de las actividades principales está ya desarrollada en forma de 5 funciones:

- `sintonizador()`: lee la señal de la antena y los comandos, procesa la señal y envía un frame de imagen mediante un evento con datos
- `demultiplexor()`: recibe un frame de imagen y lo divide en la parte de audio y la de vídeo; envía un frame de imágenes y un búffer de sonido, respectivamente a los decodificadores de vídeo y audio mediante eventos con datos
- `decodificador_audio()`: recibe un puntero a un búffer de audio, decodifica y procesa el sonido y lo transfiere al altavoz por medios electrónicos
- `decodificador_video()`: recibe un puntero a un frame de imagen, decodifica y procesa la imagen y la escribe en la memoria de vídeo

Software funcional (cont.)

- `receptor_ir()`: Lee señales del mando a distancia por medios electrónicos, y en caso necesario escribe imágenes o partes de imágenes en la memoria de vídeo y comandos en el objeto compartido de comandos.

Tanto los frames de imagen como los búfferes de audio se representan mediante punteros a la información. Por sencillez, se pueden representar estos punteros mediante números enteros.

Requisitos funcionales

1. El sistema debe invocar a `sintonizador()` periódicamente, con una frecuencia de 50 Hz. Esta función, al acabar envía un evento con datos
2. La función `demultiplexor()` se ejecuta mediante el evento con datos que proviene del sintonizador. A su vez, al acabar envía sendos eventos con datos a los decodificadores
3. La función `decodificador_audio()` se ejecuta al recibir el evento procedente del demultiplexor. Durante su ejecución envía la señal de sonido al altavoz
4. La función `decodificador_video()` se ejecuta al recibir el evento procedente del demultiplexor. Durante su ejecución escribe en la memoria de vídeo
5. El sistema debe invocar a la función `receptor_ir()` periódicamente, con una frecuencia de 125 Hz. Esta función escribe en la memoria de vídeo durante su ejecución y también escribe comandos.

Requisitos no funcionales

6. Las actividades periódicas tienen plazos iguales a los periodos.
7. Existe un plazo de principio a fin de 40ms desde el inicio del periodo de la función `sintonizador()` hasta que el decodificador de vídeo finaliza.
8. Igualmente, existe un plazo de principio a fin de 40ms desde el inicio del periodo de la función `sintonizador()` hasta que el decodificador de audio finaliza.
9. Los requisitos temporales deben validarse con un análisis de planificabilidad.
10. Los modelos *temporal* y *arquitectónico* solo tendrán en cuenta los elementos software. En cambio, el modelo de requisitos tendrá en cuenta también los componentes físicos, tales como la antena, el mando a distancia, el altavoz o la pantalla.

Requisitos no funcionales (cont.)

11. Los tiempos de ejecución de peor y mejor caso (C y C^b) medidos para las funciones software ya disponibles son los siguientes:

función	C (ms)	C^b (ms)
sintonizador()	1.0	0.5
demultiplexor()	2.0	1.0
decodificador_audio()	2.0	1.5
decodificador_video()	4.0	1.1
receptor_ir()	0.25	0.2

Otros requisitos no funcionales

Los desarrollos y el software básico estarán basados en una plataforma que dispone de un procesador con un sistema operativo gobernado por eventos, con prioridades fijas y las siguientes características

Propiedad	Valores
Rango de prioridades	1...32
Rango de prioridades de interrupción^a	32...32
Tiempo de cambio de contexto (ms)	0.002...0.003
Overhead de las interrupciones	0.002...0.002
Tipo de Temporizador	Alarm Clock
Overhead del temporizador (ms)	0.001...0.002

a. En todo caso, observar que en este sistema no hay interrupciones

Ejercicios

1. Dentro del proceso de análisis de requisitos, generar uno o varios diagramas UCM para los requisitos del sistema
2. Modelar con AADL una arquitectura con 2 flujos de eventos:
 - procesado de la señal de audio/vídeo
 - procesado de la señal del mando a distancia

Ejercicios

3. Modelar con MAST la arquitectura del sistema^{1 2} para realizar un análisis de planificabilidad inicial y responder a estas preguntas:
- ¿Cuáles son las prioridades óptimas?
 - ¿Es planificable el sistema con estas prioridades?
 - ¿Cuánto podríamos aumentar los tiempos de ejecución y seguir manteniendo el sistema planificable? ¿O cuánto habría que disminuirlos en su caso?
 - ¿Cómo afectaría a la planificabilidad hacer que las dos tareas de decodificación de audio y vídeo tengan la misma prioridad? ¿Ambas iguales a la menor de las dos? ¿O iguales a la mayor?
 - justifica la respuesta

1. Observar que en MAST se puede usar un manejador de eventos de múltiples salidas de tipo “Multicast” (bifurcación) a la salida del demultiplexor para conectarlo con los decodificadores de audio y vídeo

2. Para usar elementos de tipo “Multicast” debe usarse la versión de MAST 1.5.1 o posterior

Entregar 4 ficheros

Un informe en pdf con:

- diagrama(s) UCM
- diagrama AADL de nivel de sistema, con el máximo nivel de detalle
- una captura de pantalla de los resultados de MAST
- las respuestas a las preguntas planteadas

Workspace de UCMNav comprimido

Workspace de OSATE comprimido

Ficheros del modelo MAST en un archivo comprimido