

EQUIPO DOCENTE PARA CONTROL DE PROCESOS EN TIEMPO REAL

Abel Mora, Juan M. Cerezo, Aurelio Vega, Jorge Monagas y Alfonso M. Escuela
Departamento de Ingeniería Electrónica y Automática.
Univ. de Las Palmas de Gran Canaria. 35017 – Las Palmas de Gran Canaria

El uso de los sistemas de control en tiempo real va en continuo aumento, se pueden ver tanto en la industria como en cualquier tipo de aplicaciones y tienen un pujante desarrollo en la Ingeniería. A partir de las especificaciones de un cliente se entrega un producto terminado y autónomo, que estrictamente no depende de un operador para su funcionamiento.

El presente trabajo describe el diseño e implementación de un equipo de control de procesos en tiempo real aplicado a la docencia. El diseño consta de dos bloques fundamentales: el hardware (CPU, memorias y dispositivos de E/S), que nos proporciona los recursos básicos de computación y, por otra parte, el diseño del software residente en el equipo, es decir, el sistema operativo en tiempo real que nos proporciona un entorno multitarea donde se ejecutan múltiples procesos concurrentemente.

1. Introducción.

Este trabajo describe el diseño e implementación de un equipo para el control de procesos en tiempo real, tales sistemas se utilizan en entornos en donde deban ser aceptados y procesados un gran número de sucesos, la mayoría externos al mismo, en breve tiempo o dentro de ciertos plazos. Entre otras se pueden destacar las siguientes aplicaciones: control industrial, equipos de conmutación telefónica, control de vuelo, simulaciones en tiempo real, control médico, etc.

Dentro de estos sistemas son fundamentales las interfaces físicas entre el procesador, el sistema y el entorno. Se deben establecer los mecanismos de comunicación y temporización, tanto desde el punto de vista hardware como software. El sistema debe asegurar simultáneamente la observación de magnitudes para seguir la evolución del entorno, la consideración de sucesos que sobrevienen aleatoriamente, la evaluación de decisiones con base en los sucesos y observaciones y, la generación de acciones junto al entorno para asegurar la coherencia del conjunto de la aplicación. Los requerimientos de un sistema en tiempo real en base a los datos y tiempos de respuesta dependen de la aplicación específica en tiempo real a la que vaya a ser destinado.

2.- Descripción del Hardware del equipo de prácticas.

Debido a la necesidad de diseñar un equipo de control de propósito general, es útil disponer de una placa lo más versátil posible para cualquier tipo de aplicaciones, teniendo en cuenta las limitaciones de la misma.

Puesto que se trata de un sistema de control, el equipo debe ser autónomo, es decir, debe poseer algún tipo de procesador que interprete y ejecute las tareas de aplicación programadas. Dichas tareas, junto al sistema operativo en tiempo real, debe estar contenido en memoria no volátil, que en este caso se trata de una memoria EPROM, además, se necesita también una memoria RAM para manipulación de datos temporales, variables globales, etc. Ambas deben ser lo suficientemente amplias para contener el RTOS y los programas de aplicación. Hay que tener en cuenta que el sistema operativo ocupa memoria RAM, donde se localizan variables globales, bloques de control de procesos, pilas de los procesos y demás, aparte de memoria inalterable de sólo lectura donde se aloja el código del mismo.

Las características de la tarjeta son las que muestra la siguiente tabla.

- Placa microcontroladora basada en el MC68HC11 de Motorola.
- Diferentes rangos de memoria RAM/EPROM seleccionables por el usuario. Hasta 32Kbytes de RAM y hasta 56Kbytes de EPROM.
- Convertidor analógico/digital de 8 entradas con 8 bits de resolución
- Convertidor digital/analógico de salida con una resolución de 8 bits
- Puerto serie de comunicaciones asíncrono (SCI)
- 3 líneas sólo de entrada (PA0..PA2), 3 líneas sólo de salida (PA4..PA6) y 14 líneas bidireccionales (PA3, PA7, PC0..PC7, PD2..PD5)

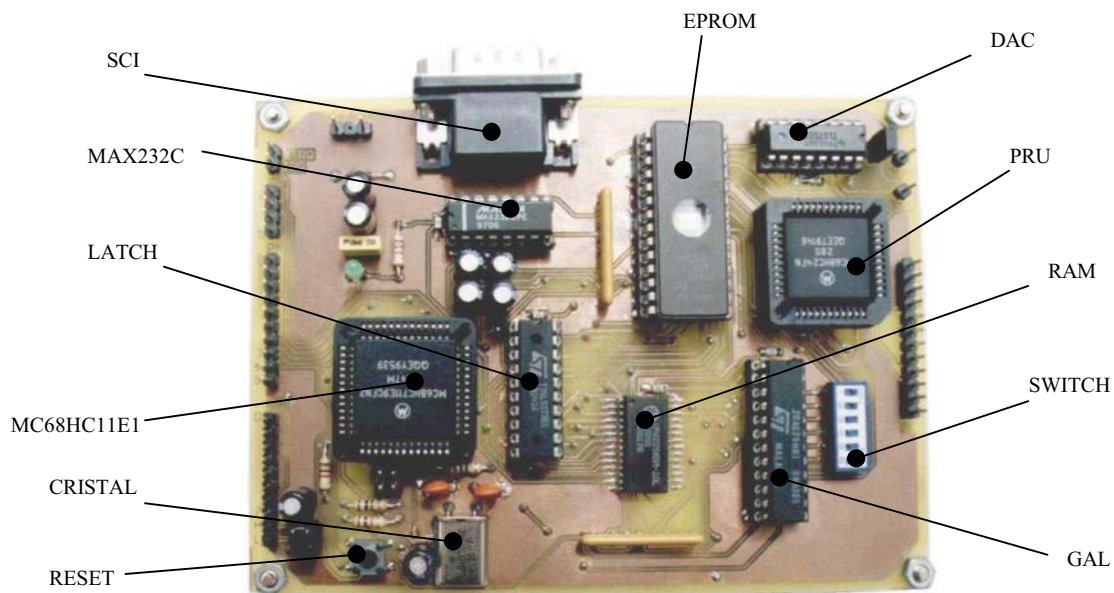


Figura 1 – Imagen de la tarjeta de control

3.- Descripción del Software del equipo de prácticas.

El software del equipo está constituido por un Sistema Operativo en Tiempo Real (RTOS), que permita soportar la ejecución concurrente de varias tareas y proporcionar rápidos tiempos de respuesta a sucesos tanto internos como externos al mismo. Este tipo de sistemas

operativos juegan un papel imprescindible en los sistemas de control en tiempo real, donde tanto el hardware como el RTOS y los programas de aplicación están íntimamente relacionados.

El diseño e implementación del sistema operativo consta de varias etapas: estructura del núcleo, gestión de procesos, comunicación y sincronización entre procesos y gestión de interrupciones, además del desarrollo de una interfaz (Nivel 5) entre el usuario y el sistema (Shell).

Las características más importantes que ofrece el sistema implementado son las siguientes:

- Entorno multitarea en tiempo real
- Servicios de temporización
- Servicios de comunicación y sincronización
- Interfaz entre el usuario y el sistema operativo a través de un terminal
- Carga, ejecución, suspensión y reanudación de procesos desde el terminal, etc.

El núcleo del sistema operativo implementado gestiona las siguientes funciones:

- Estructura y elementos de un proceso.
- Estados de los procesos.
- Lista de preparados.
- Planificador de procesos.
- Proceso Nulo.
- Temporizador del sistema.
- Inicialización del sistema.
- Comienzo de la multitarea

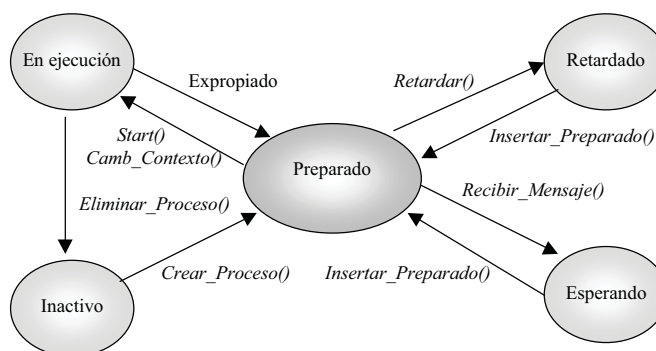


Figura 2 – Diagrama de transiciones de estado

La **Figura 2** muestra el diagrama de transiciones de estado de los procesos, indicando los servicios del RTOS que los hace conmutar de un estado a otro.

El sistema operativo proporciona un mecanismo indirecto y con almacenamiento para intercambio de mensajes a través de buzones. Los mensajes son de longitud variable y sólo se pasa el puntero desde el emisor al receptor. El RTOS gestiona las interrupciones hardware a través de mensajes, es decir, informa a los procesos de usuario sobre la llegada de una interrupción a través de mensajes y buzones. Esto permite proteger al sistema de una manipulación indebida en los ficheros privados del mismo que pudiera poner en peligro su integridad.

El *Shell* o intérprete de comandos constituye el Nivel 5 dentro de una jerarquía de niveles de un sistema operativo y proporciona una interfaz entre el usuario interactivo y el sistema operativo.

Puesto que se trata de un sistema empotrado y autónomo de control donde los programas de aplicación están en memoria EPROM y el elemento de procesamiento se encuentra en el mismo sistema, no se necesitaría implementar este nivel en el RTOS. Sin embargo, se ha realizado esta interfaz principalmente para facilitar al usuario o programador la labor del diseño y depuración de sus programas de aplicación. Además, dota al sistema de control de una mayor flexibilidad a la hora de situarlo en un entorno de trabajo, pues podría estar tanto controlando procesos en una fábrica de productos, como en un laboratorio donde los alumnos realizan prácticas en él.

A través de esta interfaz, el usuario interactivo puede realizar los siguientes servicios.

- Cargar un nuevo proceso por el puerto serie
- Ejecutar el proceso cargado
- Ver lista de todos los procesos del sistema (nombre, prioridad y estado)
- Suspender la ejecución de un proceso
- Reanudar la ejecución de un proceso
- Escribir y leer datos en una posición de un vector global
- Cambiar la frecuencia del temporizador del sistema

4.- Ejemplos de Aplicación.

Se han desarrollado ejemplos de aplicación que muestran las prestaciones del equipo diseñado. Estas son:

- Controlador PID que implementa este tipo de controlador sobre las variables asignadas
- Ejemplo de comunicación entre procesos que sincroniza distintos procesos mediante IRQ.

5.- Conclusiones.

Con el trabajo se ha conseguido diseñar una electrónica muy compacta para su uso en el laboratorio en aplicaciones de control de sistemas. Se le ha dado máxima importancia al capítulo de software, donde se ha logrado implementar una versión experimental de un mini RTOS muy atractiva desde el punto de vista docente. De esta forma, se permite al alumno llegar a todos los niveles del sistema y comprobar su eficacia.

Por otro lado, se ha desarrollado un interfaz muy cómodo para la carga de nuevos procesos que pueden interactuar en el sistema. EL mecanismo de comunicación entre proceso permite una gran flexibilidad en el desarrollo de aplicaciones para este equipo de prácticas.

6.- Referencias.

- [1] Jean J. Labrosse, "MicroC/OS-II", The Real Time Kernel, R&D Books, 1999.
- [2] Karl J. Åström, Björn Wittenmark, "Compuer-Controlled Systems, Theory and Design", Third Edition, Prentice Hall.