

# ELEMENTOS DE COMPUTACIÓN EN TIEMPO REAL APLICADOS A LA SOLUCIÓN DE PROBLEMAS DE CONTROL INDUSTRIAL CON PLC'S

Hinostroza Zubia Hector Raul

Instituto Tecnológico de Cd. Juárez  
Blvd. Tecnológico # 1340  
Cd. Juárez Chihuahua  
Tel 656 6 88 25 00  
[hhinostroza@terra.com.mx](mailto:hhinostroza@terra.com.mx)  
[www.hhpowersof2.com](http://www.hhpowersof2.com)

## Resumen

*Los procesos industriales funcionan bajo un esquema de tiempo real, es decir el proceso real desarrolla sus actividades en un medio ambiente dinámico y asíncrono en naturaleza, esta característica se refleja en el controlador que lo gobierna y que permite que opere adecuadamente, algunos de estos procesos son manejados y controlados por los llamados controladores lógicos programables (PLC'S), los cuales tienen una estructura muy similar a una computadora digital, basados en un microprocesador, memoria, dispositivos de entrada salida, etcétera. Con la diferencia, que en un PLC, tenemos un sistema operativo en tiempo real (RTOS) el cual es llamado monitor, de tal forma que el programa de control que se ejecuta en el PLC determina la operación misma del proceso que esta controlando y esta función se transmite hacia el sistema. Los conceptos de programación en tiempo real como sincronización de tareas, semáforos, mutua exclusión, etcétera. son una herramienta adecuada para la solución de problemas de control secuencial, las actividades de los procesos industriales requieren de controladores que trabajen en las condiciones en las cuales el proceso esta funcionando, un proceso en tiempo real.*

## 1 - Introducción

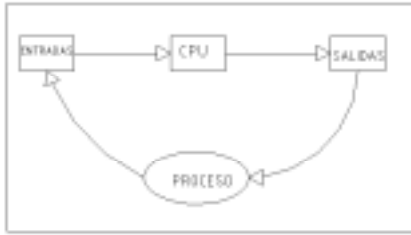
Los sistemas en tiempo real controlan los eventos que ocurren en el tiempo común y

corriente y para ello deben actuar con un tiempo de respuesta restringido, la exactitud de los resultados es tan importante como el tiempo en que se realizan, y un proceso industrial es en naturaleza un sistema en tiempo real, el sistema de control que gobierna el proceso industrial tiene básicamente una serie de tareas comunes como las que se plantean a continuación:

- Secuencia
- Control de movimiento
- Conteo y Temporizado
- Control de Trafico
- Control de Procesos (PID)
- Toma de decisiones
- Comunicación con otros dispositivos

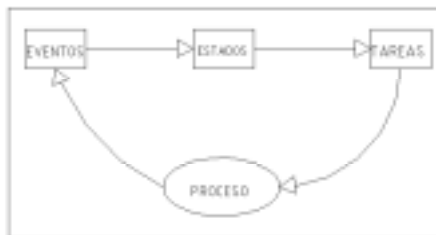
Estas tareas son llevadas a cabo por medio de la activación de salidas (digital y/o análogas) específicas de un controlador en base a eventos sucedidos dentro del mismo proceso, y los eventos son identificados como entradas del controlador (digitales y/o análogas), en el programa de control se lleva a cabo la función de ejecutar las tareas en el momento que se requieran y con el valor adecuado que se requiere, lo que se llama en suma el proceso de control.

La figura # 1 representa la configuración más simple de un PLC el cual esta formado por entradas- salidas, CPU y el proceso a controlar, podemos hacer una analogía de este, con la configuración que queremos utilizar como se observa en la figura # 2.



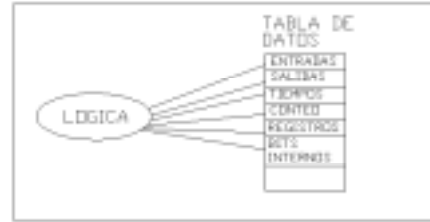
**Fig. # 1 Configuración de un PLC**

En la cual tenemos los eventos identificados con los dispositivos externos (sensores) que son las entradas que nos proporcionan información del estado del proceso a controlar, las tareas son las acciones que requiere el proceso para operar adecuadamente y estas son llevadas a cabo a través de otros dispositivos externos (actuadores) identificados con las salidas. En consecuencia los estados son condiciones que se plantean en el programa de control dentro de la CPU del PLC y es propiamente el programa de control. Para generar estos estados y sincronizarlos adecuadamente es necesario generar la lógica precisa requerida y en su caso utilizar conceptos como mutua exclusión, semáforos, sincronización de tareas, etcétera. Para llevar a cabo esta función en la forma más adecuada.



**Fig. # 2 Aproximación en tiempo Real**

Un controlador requiere conocer el estado presente del proceso y basado en ello toma las decisiones adecuadas, para lograr esto requerimos una tabla de estados, como la que se indica en la figura # 3 que en un PLC es conocida como tabla de datos donde esta el valor digital de cada una de las entradas salidas, valor de las bases de tiempo, de conteo, e información requerida para elaborar el procedimiento de sincronización, la evaluación de los datos para toma de decisiones y el funcionamiento normal de operación del sistema. En esta tabla tenemos información digital de cada uno de los elementos que comprenden el sistema.



**Fig. # 3 Tabla de Estados**

## 2 - Conceptos de Programación en Tiempo Real

Antes de entrar en una discusión de este tipo de programación, sería conveniente el dar una definición más general de los siguientes conceptos.

Sistema en Tiempo Real<sup>~</sup> es un sistema que siempre responde en forma determinista y garantiza que esta respuesta este dada en un determinado periodo de tiempo [1]. La utilización de la programación en tiempo real en un proceso industrial, como en otros sistemas es necesaria por razones puramente técnicas, como puede ser la eficiencia, funcionalidad y operación del sistema, así mismo podemos ver la introducción del tiempo como un factor importante dentro de la programación. En programas convencionales, las acciones se realizan para cambiar los valores de variables en pasos discretos. La ejecución de estas acciones por cualquier procesador toma una cantidad finita de tiempo por tal motivo el tiempo es un factor importante que se debe de tomar en cuenta, muy especialmente en este tipo de aplicaciones.

Es posible diseñar un sistema de control en el cual todas o algunas de las tareas estén en estados definidos, de tal manera que su actividad se refleje como si estas estuvieran activas y ejecutándose a la vez, a continuación tenemos una serie de elementos de programación en tiempo real que se pueden utilizar para crear los programas de control de un sistema industrial.

### 2.1 - Semáforos.

Al analizar mas detalladamente la coordinación de procesos en tiempo real, el problema básico es la sincronización parcial en la ejecución asíncrona de los procesos, los eventos suceden en forma asíncrona en todos los

dispositivos que componen el proceso. Para ello se hace necesaria una forma de coordinar un conjunto de eventos concurrentes y asíncronos, los cuales son sensibles en su ejecución en el tiempo, la solución más simple a este problema es tener un mecanismo que diga a una tarea o varias tareas cuando un evento que esperaban ha sucedido, este mecanismo utiliza una variable lógica llamada Semáforo. Este es un mecanismo por el cual tareas y eventos son sincronizados. Los semáforos pueden ser expresados en términos de señales o variables ordinarias, lo que se llama semáforos binarios y semáforos generales.

Los semáforos binarios solo son usados para resolver el problema de mutua exclusión y son llamados binarios por el hecho de que su valor es de "1" o "0" dependiendo de las condiciones de su variable asociada.

## 2.2 - Semáforos Generales

Los semáforos generales son variables cuyo contenido representa un contador, este es inicializado en cero y se le suma o se le resta en uno, según sea la operación realizada. Las operaciones válidas en semáforos son "E" (regularmente llamada espera) y "S" (regularmente llamada señal), las dos operaciones de semáforos permiten a un proceso bloquearse automáticamente y esperar por un cierto evento, entonces será despertado por otro proceso cuando el evento ocurra, "E" y "S" tienen el siguiente sentido.

E(N): Espera hasta que "N" sea mayor que cero y entonces resta 1 de "N"

S(N): Suma un 1 a "N" genera una señal.

## 2.3 - Mutua Exclusión.

Para compartir recursos en procesos concurrentes, estos tienen que ser sincronizados. Por tal motivo se tiene que hacer reglas de sincronización una de ellas es la llamada Mutua Exclusión. Un ejemplo es cuando un recurso que solo puede ser accesado por un proceso a la vez, para lo cual el proceso requiere "Exclusividad" de este recurso en un momento dado y ninguna otra tarea podrá accederlo.

## 2.4 - Relojes de Tiempo Real

La sincronización en tiempo real requiere en ocasiones utilizar un tiempo de intervalo, el cual es manejado y controlado por un reloj interno, el cual tiene un valor definido por programa, o esta en base a parámetros del mismo proceso, este valor es almacenado en la tabla de datos, su función es manejar los valores de tiempo asignados a determinados eventos e inicia o detiene las tareas que esperan por estos eventos. Todas las tareas internas pueden generar bases de tiempo, cuando este tiempo estipulado se consume, el reloj retorna una respuesta a la tarea que hizo el requerimiento.

## 2.5 - Deadlock

Cuando varias tareas compiten por uno o varios recursos se puede presentar una situación en la cual las tareas queden trabadas entre sí por la siguiente situación, supongamos que la tarea X requiere el recurso W para continuar su proceso, el cual está asignado a la tarea Y de igual forma la tarea Y requiere el recurso Z que tiene asignado la tarea X, de esta manera las tareas X y Y requieren los recursos que están asignados a su contraparte y esta es una situación que no se destrabara de ninguna forma, así que las tareas quedaran suspendidas indefinidamente, este es el problema del estancamiento (deadlock).

## 2.6 – Sincronización de Tareas

Uno de los elementos de un sistema operativo, es el control de procesos o manejador de procesos. Una de las principales funciones de este manejador son las siguientes:

CREAR PROCESOS

REANUDA PROCESOS

ELIMINA PROCESOS

Algunas de las funciones complejas son las de reanudar y detener procesos. ya que un proceso es una unidad bien definida que compite por los recursos de un sistema. Este es básicamente un programa o una tarea en ejecución, y un proceso está asociado a un usuario. El problema básico en esta configuración, es asignar a un proceso el recurso que requiere de una manera eficiente, ya que el manejador de procesos tiene que controlar todos los procesos que existen dentro de la configuración; esto quiere decir que el sistema

puede crear nuevos procesos y eliminar algunos otros, cuando ellos han terminado su actividad.

La principal función en la sincronización de tareas es guardar el estado corriente de todos los procesos que existen y cambiar su estado cuando sea necesario.

#### 2.7 - Mecanismo de Sincronización.

Consideremos dos procesos los cuales son llamados el "PRODUCTOR" y el "CONSUMIDOR" respectivamente. El productor es un proceso cíclico y cada vez que se realiza un ciclo produce una porción de información que es procesada por el consumidor. El consumidor también es un proceso cíclico y cada vez que realiza un ciclo procesa a porción de información creada por el productor, así que la relación productor-consumidor requiere un canal de comunicación entre los dos procesos. Este semáforo tiene que proveer de las acciones de bloquear, reanudar y mutua exclusión.

### 2- Estados del Sistema

Una de las formas de representar los estados del sistema es utilizando las entradas salidas internas, estas son una serie de datos internos (bits o registros) que no están relacionados directamente con los dispositivos externos como los sensores y actuadores, y representan las condiciones o pasos intermedios que reflejan todas las situaciones que se presentan en el sistema, por lo tanto pueden representar los estados del mismo y el programa de control debe estar formado por estos estados, los cuales posteriormente son relacionados con las salidas externas (tareas), y la generación de los estados regularmente son creados por la combinación lógica de las entradas externas (eventos), en combinación con las mismas entradas salidas internas (estados), cerrando el ciclo del proceso. Es la actividad intermedia que se lleva a cabo en el programa de control, el cual toma los datos de entrada (eventos) y genera las salidas (tareas) correspondientes, La idea es segmentar el problema a resolver creando una relación de todas las condiciones o situaciones posibles en el sistema y tratando cada uno de ellos por separado, así el programa se simplifica, configuramos todas las condiciones que se presenten, y cada una la tratamos como estado del sistema obviamente interrelacionado con los otros estados generando un conjunto de datos

que representan al mismo, y cada tarea será relacionada con los estados que la requieran. Toda esta información esta en la tabla de datos.

### 3 - Aplicaciones Practicas

Una de las aplicaciones realizadas consiste en la aplicacion del concepto de mutua exclusion y semaforos, el problema consiste en compartir un recurso en este caso es una banda transportadora, por un lado tenemos un robot que coloca una pieza (monobloque)



**Fig. # 4 Aplicación 1**

en una banda transportadora, esta banda es compartida por el robot y un transportador que toma este monobloque a otra parte del proceso, cuando el robot lo deposita en la banda esta se mueve hacia arriba y se activa para que el monobloque se deposite sobre un gancho colgado en el transportador aéreo, una vez dentro del gancho, el transportador se mueve, en este caso la mutua exclusión debe de aplicarse en el programa de control, puesto que tiene un recurso compartido (la banda) por dos usuarios el transportador aéreo y el robot.

Otra aplicación es la relacionada a un proceso de producción de televisiones de proyección (PTV) este proceso es una línea de producción, que tiene en una parte de la misma, un área un tanto complicada, donde tenemos en un punto de alto trafico, donde tenemos 5 entradas (FIGURA # 6) de producto y tres salidas del mismo, para poder hacer la sincronización de los PTV es una tarea que



**Fig. # 5 - Aplicación 2**

requiere de la aplicación de mecanismos de sincronización de tareas, para poder realizar esta actividad adecuadamente.



**Fig. # 6 Flujo de Linea**

Como se puede ver en la figura # 6 tenemos graficado el flujo dentro de este proceso, el movimiento de los productos está indicado con flechas, y estos se mueven en transportadores, en una dirección o en otra, existen puntos críticos, donde tenemos tráfico intenso, y requerimos sincronizar esta actividad, utilizamos conceptos como semáforos, mutua exclusión, sincronización de tareas, para llevar a cabo esta sincronización, tratando de generar un sistema de control eficiente y funcional.

#### 4 - Conclusiones.

Las tareas en un proceso real se tienen que sincronizar de manera tal que cumplan la asignación para la cual se crearon, dicho de otra forma una tarea en un proceso industrial que funciona en tiempo real tendrá forzosamente que generar sus resultados en este mismo esquema y por ello se requiere utilizar por un lado un sistema operativo que soporte este tipo de funcionamiento y por otro lado los esquemas de sincronización requeridos que proporcionen

resultados correctos. Uno de los efectos que esto genera es la utilización de mecanismos de sincronización que se representan en el programa de control por medio de primitivos de computación en tiempo real, que resuelven este problema, el sistema de control genera las condiciones y los resultados en función de las condiciones imperantes en un momento dado. Creando las tareas tomando como base los eventos que están sucediendo en este momento. Se han realizado bastantes proyectos de aplicación práctica utilizando estos elementos como base de los programas de control, generando soluciones con los resultados requeridos en los problemas industriales.

#### Referencias

- [1] Phillip A. Laplante *REAL- TIME SYSTEMS DESIGN AND ANALYSIS An Engineer's Handbook* IEEE PRESS second edition pg 141-187 1997.
- [2] Andrew s. Tanenbaum *MODERN OPERATING SYSTEMS* PRENTICE-HALL pg. 33-69 241-260 1992
- [3] Robert Carrow *SOFT LOGIC A Guide to using a PC as Programmable Logic Controller* MCGRAW HILL pg. 197-206 1998
- [4] G. Warnier "Implementing Real Time Scheduling in a Time Sharing Operating System" Proceedings of the 3rd. Workshop of Algorithms and Architectures for Real Time Control AART'95 Oostend, belgium 1995.
- [5] M. Gergeleit, J Kaiser "DIRECT: Towards a distributed object-oriented Real-Time Control System German National Research Center for Information Technology
- [6] Richard C. Holt "Structure of Computer Programs: A Survey" IEEE Proceedings for Interactive Computer Systems pg. 879-893 1975
- [7] E. W Dijkstra "Cooperating sequential processes" Programming Languages F. Genuys De. New York pg 43-112

**M. C. Héctor Hinostroza** - Ingeniero Industrial en Electricidad, con Maestría en Ing. Eléctrica

esp. En Control, Director del centro de estudios tecnológicos, profesor investigador UACJ, , Estancia Técnica en el Instituto de Investigaciones (IIE), consultor Tecnológico del CONACYT (RCCT\_0364), Consultor Tecnológico Especialista avalado por el CONACYT (RCT-E0010), Evaluador del Premio Nacional de Tecnología Edición 2000 Socio y Gerente de ingeniería de la empresa Diseño e Instalaciones del Norte, Presidente de la firma hpowerof2 [www.hpowerof2.com](http://www.hpowerof2.com) ,Autor del Ebook Controladores Lógicos programables .