

Elección de la herramienta de software correcta para hacer el trabajo

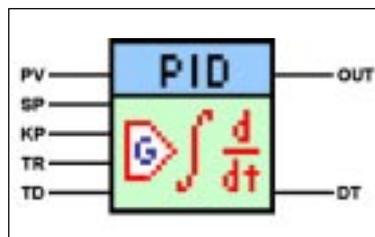
Por Arun Veeramani y Todd Walter

Arun Veeramani es director de producto de National Instruments
 Todd Walter es director de grupo de National Instruments

«Una imagen vale más que mil palabras» es un proverbio familiar que asevera que las historias complejas pueden contarse con una sola imagen o que una imagen puede ser más influyente que una gran cantidad de texto. También caracteriza acertadamente los objetivos del software basado en imágenes para control industrial. Por ejemplo, un diagrama de bloques de funciones (FBD: Function Block Diagram) puede sustituir a miles de líneas de texto de un programa. La programación gráfica es una manera intuitiva de especificar la funcionalidad del sistema ensamblando y conectando los bloques de funciones. Otras opciones es la programación textual y con diagramas en escalera como opciones para los modelos de cálculo. Este artículo ofrece un análisis en profundidad de los diagramas de bloques de funciones (FBD) y sus puntos fuertes y débiles en comparación con otros modelos populares de control y medida.

Figura 1. Un bloque de función típico con entradas y salidas

vative), contadores y temporizadores en las distintas partes de una aplicación o en proyectos diferentes. Un bloque de función es un elemento empaquetado de software que describe el comportamiento de los datos; una estructura de datos y una interfaz externa definida como un conjunto de parámetros de entrada y salida. En muchos sentidos, los bloques de función pueden ser, teóricamente, comparados con los circuitos integrados que se utilizan en los equipos electrónicos. Un bloque de función se presenta como un bloque rectangular, con entradas en la parte izquierda y salidas en la parte derecha.



Los diagramas de bloques de funciones (FBD) fueron introducidos por la norma IEC 61131-3 para superar las deficiencias relacionadas con la programación basada en texto y en diagramas en escalera. Una red FBD consta principalmente de las funciones de interconexión y de los bloques de función para expresar el comportamiento del sistema. En primer lugar, echaremos un vistazo a la definición y propiedades de las funciones y los bloques de funciones antes de sumergirnos en algunas de las características principales de los FBDs y su flujo de ejecución.

Bloques de funciones

Los bloques de funciones se introdujeron para abordar la necesidad de reutilización de las tareas comunes, como el control mediante PID (Proportional-Integral-Derivative),

contadores y temporizadores en las distintas partes de una aplicación o en proyectos diferentes. Un bloque de función es un elemento empaquetado de software que describe el comportamiento de los datos; una estructura de datos y una interfaz externa definida como un conjunto de parámetros de entrada y salida. En muchos sentidos, los bloques de función pueden ser, teóricamente, comparados con los circuitos integrados que se utilizan en los equipos electrónicos. Un bloque de función se presenta como un bloque rectangular, con entradas en la parte izquierda y salidas en la parte derecha.

Funciones

Una función es un elemento del software, que cuando se ejecuta con un determinado conjunto de valores de entrada, produce un resultado primario y no tiene almacenamiento interno. Las funciones son a menudo confundidas con los bloques de función que tienen almacenamiento interno y pueden tener múltiples salidas. Algunos ejemplos de funciones son las funciones trigonométricas como $\sin()$ y $\cos()$, funciones aritméticas como sumar y multiplicar y las funciones de gestión de cadenas de caracteres, mientras que entre los ejemplos de bloques de funciones se incluyen PID, contadores y temporizadores.

Diagrama de bloques de funciones

Un FBD es un programa construido conectando múltiples funciones y bloques de función. El resultado de un bloque se convierte en la entrada para la etapa siguiente. A diferencia de la programación basada en texto, no se necesitan variables para pasar los datos de una subrutina a otra, porque los cables que conectan bloques diferentes encapsulan y transfieren datos automáticamente.

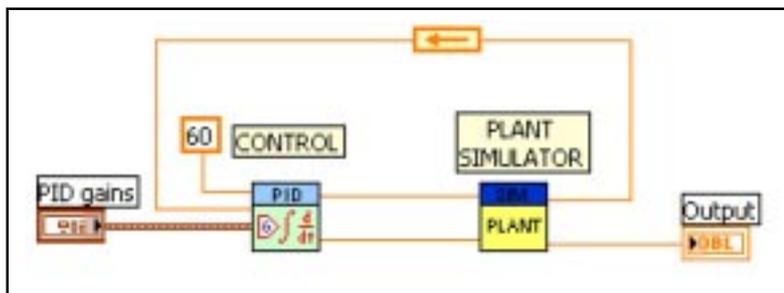
Se puede utilizar un FBD para expresar el comportamiento de los bloques de función, así como de los programas. También se pueden utilizar para describir los pasos, las acciones y las transiciones dentro de un SFC (Sequential Function Charts).

Flujo de la ejecución

Un bloque de función no se evalúa a menos que todas sus entradas que provienen de otros elementos estén disponibles. Cuando un bloque de función es ejecutado se evalúan todas sus variables,

las cuales incluyen no sólo las entradas y las variables internas, sino también las variables de salida. Durante su ejecución, el algoritmo genera nuevos valores para las variables internas y de salida.

Como ya se ha indicado, las funciones y bloques de función son los elementos básicos de un FBD. En los FBDs se considera que las señales fluyen desde las funciones o desde los bloques de función a las entradas de otras funciones o bloques de función. Las salidas de los bloques de función se actualizan como resultado de las evaluaciones del bloque de función. Por consiguiente, los cambios de los estados y de los valores de la señal se propagan naturalmente de izquierda a derecha a través de la red FBD. La señal puede también convertirse en una retroalimentación desde las salidas de los bloques de función a las entradas de los bloques anteriores. La existencia de un camino de retroalimentación implica que un valor dentro de la ruta de retroalimentación se mantiene después de que la red FBD es evaluada y usada como el valor inicial de la red en la siguiente evaluación.



El control de la ejecución de los bloques de función en una red FBD está implícito por la posición del bloque de función dentro del FBD. Por ejemplo, en la figura 2, la función «Simulador de Planta» es evaluada después del bloque de función «Control». Sin embargo, la or-

den de ejecución puede ser controlada mediante la habilitación de un bloque de función para la ejecución y disponiendo de terminales de salida que cambien de estado una vez que la ejecución se haya completado. Se considera que la ejecución de una red FBD está completada cuando todas las salidas de todas las funciones y de los bloques de función están actualizadas.

Puntos Fuertes de un FBD

- **Intuitivo y fácil de programar.**

Debido a que los FBDs son gráficos, es muy fácil para los diseñadores del sistema que no tengan una amplia formación en programación, comprender y programar la lógica de control. Esto beneficia especialmente a los expertos en esta área porque no tienen que ser necesariamente expertos en la escritura de los algoritmos de control específicos en lenguajes basados en texto, pero que entienden la lógica del algoritmo de control. Estos expertos pueden utilizar los bloques de función para construir los programas que ejecutan la adquisición y el proceso de los datos y el control discreto.

- **Amplia la reutilización de código.**

Una de las principales ventajas de los bloques de función es la reutilización del código. Como ya se ha indicado, los diseñadores del sistema pueden utilizar bloques de función ya existentes como PIDs y filtros o encapsular

la lógica personalizada y reutilizar fácilmente este código en todos sus programas. Dado que se hacen copias por separado cada vez que estos bloques de función son llamados, los diseñadores del sistema no corren el riesgo de la sobrescritura accidental de los datos. Además, los bloques de función también puede ser invocados desde los diagramas en escalera e incluso desde lenguajes basados en texto tales como *Structured Text*, lo que los hace altamente transportables entre diferentes modelos de cálculo.

- **Ejecución en paralelo.**

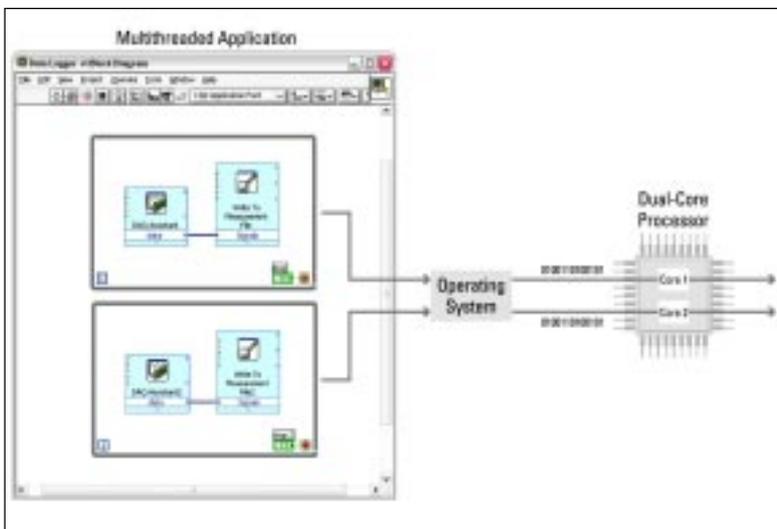
Con la introducción de los sistemas basados en múltiples procesadores, los controladores de automatización programables y los PCs son ahora capaces de ejecutar múltiples funciones literalmente en el mismo tiempo. Los lenguajes de programación gráfica, tales como FBDs, son capaces de representar eficientemente la lógica en paralelo. Mientras que los programadores de lenguajes basados en texto utilizan librerías específicas de hilos y de temporización para aprovechar la capacidad de los lenguajes multi-hilo, los lenguajes gráficos, los FBD y los lenguajes de flujo de datos como LabVIEW de National Instruments pueden ejecutar automáticamente bloques de función en paralelo en hilos diferentes. Esto es útil en aplicaciones que requieren un control avanzado tales como aquellas que hacen funcionar múltiples PIDs en paralelo.

- **Ejecución trazable y de fácil de depuración.**

El flujo gráfico de datos de los FBDs hace que la depuración del sistema sea fácil, ya que los diseñadores del sistema pueden seguir los hilos de conexión entre las funciones y los bloques de

Figura 2. Una red FBD muestra los caminos de realimentación y del control de la ejecución.

Figura 3. Los entornos especializados de desarrollo gráfico tales como NI LabVIEW ofrecen la posibilidad de ejecución en paralelo.



función. Muchos editores de programas FBD como *Siemens-Paso 7* proporcionan también animación para mostrar el flujo de datos y así hacer más fácil la depuración de los fallos.

Debilidades de los FBDs

- **Desarrollo del Algoritmo.**

Las funciones y los algoritmos matemáticos de bajo nivel están tradicionalmente representados con funciones de texto; incluso los algoritmos para los bloques de función se han escrito convencionalmente utilizando la programación basada en texto. Por otro lado, los bloques de función abstraen las complejidades de un algoritmo, lo que hace difícil para los expertos en este dominio tratar de aprender los detalles de las técnicas avanzadas de control y procesamiento de señales.

- **Control limitado de la ejecución.**

La ejecución de una red FBD es de izquierda a derecha y es conveniente para una actividad continua. A pesar de que los diseñadores del sistema pueden controlar la ejecución de una red a través de las sentencias de «salto» y tam-

bién mediante el uso de la dependencia de datos entre dos bloques de función, los FBDs no son la solución ideal para resolver los problemas secuenciales. Por ejemplo, al pasar del estado «llenar el depósito» al estado «agitar el depósito» se requiere la evaluación de todos los estados actuales. Dependiendo de los resultados de la salida, debe llevarse a cabo una acción de transición antes de pasar a la siguiente situación. Aunque esto todavía se podría lograr utilizando la dependencia de los datos de los bloques función, podría requerir mucho tiempo y esfuerzo por parte del diseñador del sistema para lograr esta secuencia.

- **Integración de TI.**

Dado que las empresas buscan cada vez más la forma de conectar sus modernas plantas de fabricación a la empresa, la conectividad a la Web y las bases de datos se han vuelto sumamente importantes. Mientras los programas basados en texto tienen capacidades incorporadas de registro en bases de datos y características de control del código fuente, los FBDs son generalmente incapaces de integrarse nativamente con los sistemas de

TI (Tecnología de Información). Por otra parte, los administradores de TI son a menudo formados solamente en programación basada en texto.

- **Necesidades de formación.**

Aunque es intuitivo, el flujo de datos no es comúnmente enseñado como un modelo de cálculo. En los EE.UU., los ingenieros son formados para el uso de lenguajes basados en texto como C++, Fortran, Visual Basic y los técnicos son formados en la lógica en escalera o en circuitos eléctricos. Los FBDs requieren una formación especial, ya que representan un cambio de paradigma en la escritura de un programa de control.

Los FBDs son una forma de representación gráfica del programa de control de datos y son un modelo de programación mediante flujo de datos. El que sean muy intuitivos, fáciles de usar y con capacidad de reutilización del código, hace que los FBDs sean muy populares entre los ingenieros. Los FBDs son ideales para aplicaciones complejas con ejecución en paralelo y para procesamiento continuo. También cubren con efectividad los espacios en blanco de la lógica de escalera como la encapsulación y la reutilización del código. Para superar algunas de sus debilidades los ingenieros deben emplear modelos mixtos de cálculo.

Los FBDs se utilizan en combinación con la programación basada en texto en el caso de los algoritmos y de la integración de TI.

Las operaciones en lotes y discretas se mejoran añadiendo SFCs (Sequential Function Charts). El modelo SFC de cálculo hace frente a algunos de los desafíos a los que se enfrentan los FBDs. ○