LABVIEW

Fase 3: Establecer comunicación entre el sistema de supervisión y servidor web.

Presentado por

SAMAEL AMADO

Dídier Javier Ramírez Henao

Presentado a

ALEXÁNDER FLÓREZ

Grupo

208055\_1

NOVIEMBRE 02 de 2021

Manizales – Caldas

**INTRODUCCIÓN**

**Con el desarrollo de la fase 3, se va a iniciar con la etapa individual que consiste en implementar y trabajar con la librería datalogging and supervisory control (DSC) de LabVIEW. Esta librería como es complementaria al software que viene instalado de forma predeterminada en LabVIEW y como lo indica en la página de National Instruments “se puede usar este módulo para desarrollar un sistema de monitoreo y control distribuidos que va desde docenas hasta decenas de miles de etiquetas. Incluye herramientas para registrar desde datos a una base de datos histórica en red, rastrear tendencias de datos históricos y en tiempo real, administrar alarmas y eventos, crear redes de dispositivos LabVIEW Real-Time y dispositivos OPC en un sistema completo y añadir seguridad a interfaces de usuarios”. Esta librería se va a utilizar para personalizar los instrumentos virtuales indicados en la guía de actividades.**

**En la etapa colaborativa, se va a continuar con el subproceso 2 de la empresa Cocoa Bens, llamado subproceso de torrefacción; allí, de acuerdo con las especificaciones brindadas, básicamente se debe tener controlada la temperatura del proceso entre el límite superior e inferior, funcionando el extractor de aire o la caldera según se requiera. Para esta actividad se va a integrar el módulo DSC de LabVIEW para visualizar de forma más realista lo que sucede en el proceso de torrefacción y la implementación de la programación se realiza por medio de diagrama de estados, los cuales permiten que la programación sea más sencilla y entendible.**

**CONTENIDO**

**Interfaz gráfica individual**

Figura 1. Pantallazo interfaz gráfica Dídier Javier Ramírez Henao.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 2. Pantallazo interfaz gráfica Samael Amado.

Interfaz de usuario gráfica

Descripción generada automáticamente

**Interfaz gráfica colaborativa**

Figura 3. Interfaz gráfica seleccionada.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Descripción de cada estado del algoritmo**

Figura 4. Pantalla inicial.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**En la figura 4 se observa que por fuera del ciclo “while loop principal” se tienen un registro; el cual es con el fin de que al salir de dicho ciclo se conserve el estado al cual se debe dirigir el programa (inicio, fin, pausa, abortar, paro, enfriar, calentar). También se tiene un “Case Structure”, el cual si se oprime el botón de paro no pasa nada y cuando se deja de oprimir pone en cero la temperatura superior, la temperatura del proceso, la temperatura inferior y el indicador de la temperatura del proceso llamado “Temperat”.**

**Dentro del “while loop principal” se tiene:**

* **El stop, el cual para el programa y cumple la función de “abortar”.**
* **Dos selectores que vamos a denominar selector 1 y selector 2.**
* **El selector 1 depende del stop (abortar). En caso de que el Stop esté presionado, selecciona el estado “Abortar” y si no se presiona Stop, continúa el estado que viene por fuera del “while loop principal” a través del registro. El resultado de la selección va al selector 2.**
* **El selector 2 depende de varias variables booleanas (Marcha 2, inicio, calentar, enfriar, parada 2); todas estas variables van unidas a través de compuertas OR, lo que quiere decir que, si al menos una de estas variables booleanas está activa, se toma la selección como verdadera y en caso de ser verdadera, se envía el estado que fue seleccionado a través del selector 1. Si ninguna de las variables booleanas está activa el selector 2 toma la opción de falso y toma el estado “Fin”. La decisión a través del selector se envía a un “case structure” donde según el estado se cumplen determinados procesos para el correcto funcionamiento de lo que se requiere del programa.**
* **Se encuentra también dentro del “while loop principal” el graficador de señales para la temperatura superior, inferior y de proceso y un tiempo de espera de 500 milisegundos.**

Figura 5. Estado "INICIO"

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Cuando dentro del “case struct” se selecciona el estado “INICIO” por medio del selector 2, se tiene la siguiente programación:**

* **Las variables booleanas Calentamiento, Enfriamiento, Calentar, Enfriar, Ventilador, Fin, se les asigna el valor de “falso”, indicando que estas variables no están activas y al horno se le asigna el valor “True”, indicando también que no está activo. La temperatura del proceso se hace igual al indicador de temperatura llamado “Temperat” y se da un tiempo de espera de 1 segundo para que el indicador booleano de inicio se pueda visualizar.**
* **Se tiene un “Case Struct” que depende de la variable “parada 2”, la cual cuando el botón de parada 2 está activo, hace que la selección sea verdadera y vaya al estado de “paro”. Si “parada 2” es falso, se compara si la temperatura superior es igual a 125 grados y si la temperatura inferior es igual a 115 grados y ambas comparaciones se llevan a una compuerta AND, con el fin de asegurar que el operario haya ingresado los límites de temperatura correctos. El resultado de la operación AND va a otro “Case Struct”. En caso de que al menos una de las temperaturas (superior o inferior) ingresadas por el operario no sean correctas la operación AND es “falsa” enviando el programa al estado “FIN”. Si la operación AND es verdadera quiere decir que las temperaturas ingresadas por el usuario son correctas, con lo que el “Case Struct” es verdadero y, por lo tanto, se activa el indicador booleano de “Inicio” y continúa al estado “CALENTAR”.**

Figura 6. Estado "CALENTAR".

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

**Cuando dentro del “case struct” se selecciona el estado “CALENTAR” por medio del selector 2, se tiene la siguiente programación:**

* **Los indicadores booleanos de inicio, enfriar, ventilador, fin y enfriamiento se ponen en estado lógico “falso”, ya que dichas variables corresponden con la etapa de enfriamiento.**
* **Se tiene un “Flat Secuence Structure” para que las variables horno, calentamiento y calentar alternen entre los valores de verdadero y falso, con el fin de dar animación al proceso de calentamiento; se genera un retardo de 50 milisegundos para que la animación sea perceptible por el usuario.**
* **Se cuenta con tres selectores que para la explicación se van a denominar selector 3, selector 4 y selector 5. El selector 3 compara el valor de 125 con la temperatura superior ingresada por el operario; si dicha temperatura es diferente el selector toma la opción falsa enviando el estado “ABORTAR” al selector 4 por su entrada falsa; en caso de que ambos valores sean iguales se envía al selector 4 por su entrada falsa el estado de “CALENTAR”.**

**El selector 4 compara si la temperatura del proceso es mayor o igual a 125 grados; si la comparación es falsa se toma el resultado que llega del selector 3; si la comparación es verdadera, quiere decir que la temperatura ha sobrepasado el límite superior y se envía el estado “ENFRIAR” a la entrada falsa del selector 5.**

**El selector 5 es para verificar si el botón de paro está activo; si no lo está se toma el resultado que viene del selector 4. En caso de que el paro se encuentre activo el selector envía al estado de “PARO”.**

* **La temperatura del proceso se hace igual al indicador de la temperatura del proceso (Temperat) y se va agregando una unidad a la temperatura del proceso mientras esté en el estado “CALENTAR”.**

Figura 7. Estado "ENFRIAR".

Interfaz de usuario gráfica

Descripción generada automáticamente

**Cuando dentro del “case struct” se selecciona el estado “ENFRIAR” por medio del selector 2, se tiene la siguiente programación:**

* **Los indicadores booleanos de inicio, calentar, calentamiento y fin se ponen en estado lógico “falso”, ya que dichas variables corresponden con la etapa de calentamiento. El horno queda en estado lógico verdadero, ya que en este estado queda apagado.**
* **Se tiene un “Flat Secuence Structure” para que las variables ventilar, enfriar y enfriamiento alternen entre los valores de verdadero y falso, con el fin de dar animación al proceso de enfriamiento; se genera un retardo de 50 milisegundos para que la animación sea perceptible por el usuario.**
* **Se cuenta con tres selectores que para la explicación se van a denominar selector 6, selector 7 y selector 8. El selector 6 compara el valor de 115 con la temperatura inferior ingresada por el operario; si dicha temperatura es diferente el selector toma la opción falsa enviando el estado “ABORTAR” al selector 7 por su entrada falsa; en caso de que ambos valores sean iguales se envía al selector 7 por su entrada falsa el estado de “ENFRIAR”.**

**El selector 7 compara si la temperatura del proceso es menor o igual a 115 grados; si la comparación es falsa se toma el resultado que llega del selector 6; si la comparación es verdadera, quiere decir que la temperatura ha sobrepasado el límite inferior y se envía el estado “CALENTAR” a la entrada falsa del selector 8.**

**El selector 8 es para verificar si el botón de paro está activo; si no lo está se toma el resultado que viene del selector 6. En caso de que el paro se encuentre activo el selector envía al estado de “PARO”.**

* **La temperatura del proceso se hace igual al indicador de la temperatura del proceso (Temperat) y se va restando una unidad a la temperatura del proceso mientras esté en el estado “ENFRIAR”.**

Figura 8. Estado "PARO".

Interfaz de usuario gráfica

Descripción generada automáticamente

**El estado de “PARO” se activa al accionar el botón de parada 2 y tiene la siguiente programación:**

* **Al ingresar en el “Case Struct” más externo se desactivan los indicadores booleanos de calentamiento, enfriamiento y ventilador poniéndolos en estado lógico “falso”; el horno se pone en estado lógico “verdadero”, ya que en este estado se apaga este indicador.**
* **Dentro del “Case Struct” del estado “PARO” se encuentra un “while loop” el cual funciona hasta que se desactive el botón de paro y se encuentran también tres “Case Struct” anidados que funcionan de la siguiente manera:**

**El “Case Struct” más externo depende del indicador booleano “inicio”. Si este indicador está habilitado, quiere decir que se dio marcha al proceso y casi al instante se oprimió el botón de paro; por lo que, se va a enviar al estado de “INICIO” cuando se desactive el botón de paro. Si el indicador booleano de “INICIO” está deshabilitado, entonces se ingresa al “Case Struct” intermedio.**

**El “Case Struct” intermedio consulta el estado lógico de la variable booleana “calentar”. Si esta variable está en verdadero, cuando se desactive el botón de paro se va a continuar en el estado “CALENTAR”; si por el contrario esta variable es falsa, se ingresa al “Case Struct” más interno.**

**El “Case Struct” más interno consulta el estado lógico de la variable booleana Enfriar; si su valor es verdadero, al momento de desactivar el botón de parada, se va a regresar al estado “ENFRIAR”; si su valor es falso, se va a enviar al estado “FIN”.**

Figura 9. Estado "ABORTAR".

Interfaz de usuario gráfica

Descripción generada automáticamente

**En el estado “ABORTAR” las variables Horno, Inicio, Calentar, Enfriar, Calentamiento, Enfriamiento se desactivan. El indicador de la temperatura del proceso se hace igual a la temperatura del proceso. Se tiene un “Flat Secuence Structure” donde se activa por un segundo el indicador de Fin para que se pueda visualizar por el usuario y por último se envía al estado donde se activa por un segundo el indicador de Fin para que se pueda visualizar por el usuario y por último se envía al estado “FIN”.**

Figura 10. Estado "FIN".

Interfaz de usuario gráfica

Descripción generada automáticamente

**En el estado “FIN” se desactivan todos los indicadores booleanos y se envía al estado “INICIO”, continuando el programa de forma cíclica a espera del ingreso de datos y movimientos del operario.**

**Implementación comunicación cliente -servidor:**

<https://drive.google.com/file/d/10k5JcBDctZPPUHeiy2acL2yh1ujUw5pF/view?usp=sharing>

**CONCLUSIONES**

* **La librería datalogging and supervisory control (DSC) de LabVIEW contiene diversos instrumentos virtuales y tienen la ventaja de que se les puede agregar animación, para obtener una interfaz gráfica más real de los procesos que suceden dentro de una planta.**
* **La programación por medio de máquinas de estados hace que el problema se pueda subdividir en partes más pequeñas y que están relacionados de forma lógica. Este tipo de programación tiene en cuenta como su nombre lo indica los estados involucrados en el proceso y lo hace más amigable para el programador.**
* **Es posible realizar la comunicación entre el sistema de supervisión y un servidor web de diversas maneras. Una de estas maneras es mediante la herramienta que tiene integrada LabVIEW llamada “Web Publishing Tool”, la cual permite tener el control de una aplicación a través de un servidor web.**

**REFERENCIAS BIBLIOGRÁFICAS**

Julián, E., y Almidón, A. (2019). Manual de programación LabVIEW 9.0. Páginas 45-48. Recuperado de <https://doi-org.bibliotecavirtual.unad.edu.co/10.5281/zenodo.2557815>

Ehsani, B. (2016). Data Acquisition Using LabVIEW. Packt Publishing. Páginas 19-22. Recuperado de <https://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1440569&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_19>

Lajara Vizcaíno, J. R. (2008). LabVIEW: entorno gráfico de programación. Marcombo. Páginas 205-208. Recuperado de <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/35715?page=216>

Flórez Martínez, A. (26,01,2017). Unidad2\_Sistema de Supervisión y Control. [Archivo de video]. Recuperado de <http://hdl.handle.net/10596/10878>