

Driver de comunicaciones entre un controlador lógico programable llamado SiBAS y un sistema SCADA sobre plataforma de software libre.

Ing. Claudia Castillo, Ing. Wilfrido López
Facultad de ingeniería de sistemas, Universidad Autónoma de Bucaramanga,
Bucaramanga, Santander, Colombia.

cpcs_76@yahoo.es
wlopez1@drummondltd.com

Resumen: El presente trabajo aborda la implementación de un Sistema SCADA en un entorno de software libre aplicado a la extracción de carbón, más precisamente, para el monitoreo en línea de las Palas Eléctricas del fabricante Bucyrus.

Esta implementación tienen como objeto principal, realizar lectura de variables en la Red de PLC, entre los cuales se encuentra el SiBAS, un PLC dedicado al control de movimiento en las Palas Bucyrus, como el de traslación, giro, subida-bajada, ataque y retractar. Para la lectura de los estados de la máquina se requiere un driver que permita la comunicación a través del protocolo HTTP entre el SiBAS y el Sistema SCADA.

I .INTRODUCCIÓN

Los sistemas SCADAS son muy importantes a nivel industrial ya que permiten de forma remota y en tiempo real verificar el funcionamiento de la máquina, brindando una herramienta para prevenir y responder de forma eficaz y eficiente a los requerimientos que se generen dentro de la empresa,

disminuyendo el costo asociado de operación y mantenimiento e incrementando la producción de las mismas. Esto ha impulsado el desarrollo de sistemas SCADAS cada día más robustos y confiables y de interfaces graficas (HMI) más amigables para el operador.

En el presente trabajo se propone utilizar para la implementación del Sistema SCADA y el driver de comunicación una plataforma de trabajo en entornos libres no solo por los costos del proyecto, sino por todas las fortalezas del software libre como lo son el sistema operativo Linux Debian, el gestor de base de datos MySQL, el lenguaje de programación Python y la plataforma de diseño Ignition desarrollada 100% en Java.

La plataforma de Ignition nos ofrece un entorno de diseño, el lenguaje de programación Python integrado, a nivel de adquisición de datos brinda la tecnología OPC-UA que tiene compatibilidad con muchos dispositivos de diferentes fabricantes dentro de una misma aplicación.

Con la implementación de este proyecto se persiguen dos objetivos específicos:

- El sistema SCADA para la supervisión de las 4 palas Bucyrus.
- El driver de comunicación que permita comunicar el PLC SiBAS y el sistema SCADA.

Este sistema además de ser capaz de monitorear las variables del proceso, genera alarmas, almacena la información en una base de datos, genera los histogramas y permite la comunicación a través de diferentes protocolos con muchos dispositivos de campo, todo esto de forma remota, sin importar la ubicación física.

II. PALA ELECTRICA BUCYRUS

Las palas eléctricas son máquinas que permiten de la manera más eficiente la extracción del carbón y del material estéril de las diferentes capas del subsuelo.

El control eléctrico y electrónico de las palas Bucyrus 495HR (ver fig. 1) es realizado por dos grupos de controladores principales, el primero de ellos y el más importante, es llamado SiBAS encargado del control motriz de la máquina. El segundo PLC SIEMENS S7400, encargado de todas las variables utilitarias a requerir que complementan el control motriz[1].

Fig. 1 Pala Bucyrus



SiBAS

Es una de las partes más importantes de la pala, con la cooperación del PLC S7400 se encarga de realizar el control de movimiento del Crowd, Hoist, Swing, Propel y del suministro de energía y control de reactiva mediante los AFE (Active Front End), está compuesto por un procesador para los módulos de entradas y salidas, un procesador central llamado ZR, y cinco procesadores adicionales encargado de las protecciones electrónicas.

III. DISEÑO DEL SISTEMA PROPUESTO

En la figura 2 se presenta un diagrama para mostrar la disposición de los nodos conectados mediante enlaces de radiofrecuencia y fibra óptica en la compañía Drummond LTD.

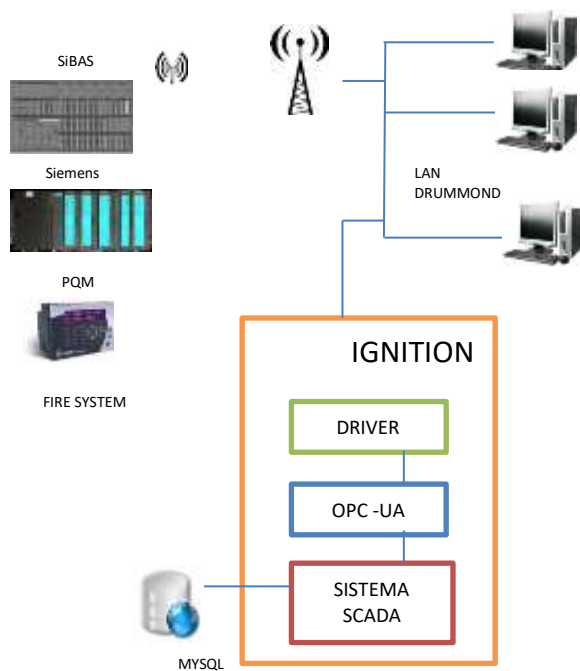
Fig. 2 Diagrama de Despliegue



La figura 3 muestra la arquitectura propuesta para el Sistema SCADA desarrollada con la tecnología de Ignition, los dispositivos están conectados a

la red y se accede a través del servidor OPC-UA permitiendo integrar los datos en el sistema de visualización. El OPC-UA ofrece flexibilidad para adicionar controladores y declarar cualquier dispositivo. Esto resulta conveniente al no disponer de un driver de comunicaciones para el PLC SiBAS en la plataforma Linux.

Fig. 3 Arquitectura Sistema SCADA



con diferentes niveles que muestren el detalle de la máquina.

- El sistema despliega un panel de alarmas activas para indicar una situación anormal.
- El sistema SCADA visualiza interfaces gráfica de supervisión para observar la evolución de las variables del proceso en tiempo real.
- El sistema SCADA debe disponer del conjunto de drivers necesario para recoger, procesar y almacenar la información recibida de los diversos dispositivos de campos.
- El sistema debe almacenar datos históricos para hacer análisis de tendencias, generar reportes y buscar fallas.

El diagrama de casos de uso del sistema SCADA se muestra en la figura 4.

Fig. 4 Casos de Uso

Actualmente la Pala dispone de un sistema local que solo permite intervención cuando la máquina se encuentra detenida en fase de inspección o mantenimiento es así como se hace necesario un sistema remoto para monitorear la actividad de la máquina. A partir de la notación UML se planteó los requisitos de este sistema dándole prioridad a los siguientes:

- La interfaz de Visualización debe contar



IV. IMPLEMENTACIÓN DEL SISTEMA SCADA

La principal característica de un sistema SCADA es la adquisición de datos. Para iniciar cualquier transmisión es necesaria la configuración de los siguientes aspectos en el Servidor de Ignition:

- Configuración del Servidor OPC-UA
- Configuración de dispositivos
- Configuración de Usuarios
- Configuración Base de Datos
- Configuración de Alarmas

Fig. 5 Configuración de Dispositivos

Modbus v2

General	
Device Name	SiemensPLC01 <small>Name of this device</small>
Unit	Default Group <small>An arbitrary number group for this device to be used in. This is primarily used as a tool for organizing the list of installed devices.</small>
Network Timeout	3000 <small>Amount of time (in milliseconds) before a broadcast communication ends.</small>
Read Timeout	3000 <small>Amount of time (in milliseconds) before a read operation times out.</small>
Write Timeout	3000 <small>Amount of time (in milliseconds) before a write operation times out.</small>
Enable Device	<input checked="" type="checkbox"/> Enable this device?

Communication	
Hostname	192.168.1.10 <small>Modbus/TCP address of the Modbus device.</small>
Port	502 <small>Port to connect to.</small>
Communication Protocol	RTU <small>Modbus amount of data to wait for a response.</small>

A. Driver de comunicación del PLC SiBAS

El driver de comunicación en entornos industriales es un programa que permite la comunicación entre los dispositivos de campo y el OPC Server, cada dispositivo a utiliza un protocolo que "traduce" la señal que envía y recibe.

En el PLC SiBAS se definen tres tipos de datos:

BOOL: Tamaño en bit 1. Rango True o False

BYTE: Tamaño en Bit 8. Rango de 0 a 255

INT: Tamaño en Bit 16. Rango - 32.768 a 32.767

Para el intercambio de datos el PLC SiBAS utiliza el Protocolo HTTP y mediante esta conexión se realiza la lectura de 57 registros o Tags de información.

El driver de comunicaciones es un script desarrollado en el lenguaje de programación Python e integrado a la interfaz de la plataforma de Ignition enlazándolo al OPC Server para el acceso a los datos.

La información suministrada por el PLC viene en el siguiente formato:

ADDRESS_DEL_TAG
0xNUMERO_HEXADECIMAL

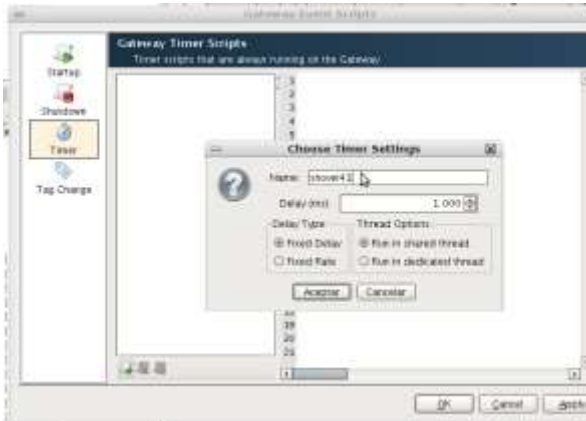
Donde ADDRESS_DEL_TAG es la Id del tag en el que se debe escribir el dato encontrado en la línea y 0Xnumero_HEXADECIMAL es un número hexadecimal cualquiera precedido por 0x. A continuación un ejemplo de cómo luciría un archivo del PLC:

HDSE 0x9
HSDS 0xC
HDSF 0x1
HDSW 0x5

Para mantener los tag actualizados, es necesario realizar un scan de los datos del SiBAS, para ello se activa un temporizador que permite la ejecución de forma cíclica del Script cada 750 milisegundos.

La figura 6 muestra la configuración del tiempo de scan del driver de comunicación.

Fig. 6 Configuración del Scan del Driver.



Para la ejecución del script se definen los siguientes parámetros:

- FilePath: Dirección IP del SIBAS de la Pala
- tagPath: Ubicación de los Tags en la estructura de Ignition.

Fig. 7 Parámetros del Driver.



En un sistema SCADA, la interfaz gráfica es importante porque proporciona una vista del proceso mediante los cuales se representa cada uno de los dispositivos y subsistemas del mismo.

En la elaboración de las pantallas de visualización se tuvieron consideraciones en colores, símbolos y formas

los cuales son importantes para reflejar un diseño parecido a la máquina.

Fig.8 Pantalla principal del Sistema SCADA.



V. CONCLUSIONES

Con la construcción de este driver de comunicaciones, el sistema SCADA implementado supera las prestaciones en costo y herramientas de información del SCADA del fabricante actualmente instalado en Drummond LTD mina, el cual será remplazado por el SCADA de Ignition, disminuyendo costos y ofreciendo mejores servicios.

RECONOCIMIENTOS

Este trabajo ha sido financiado Drummond LTD compañía a la cual los autores expresan sus agradecimientos.

REFERENCIAS

[1] Pala para minería. Manual de mantenimiento & Operación. Revisión de 2010. Wisconsin, USA: Bucyrus International.