

Monitoreo de señales a/d con transmisión de video (estándar IEEE 802.11), mediante software libre

Gómez O.*; Carguachi J.; Herrera D. **; Saquina D. ****

*Sucre Instituto Superior Tecnológico, Carrera de Electrónica Industrial, Quito, Ecuador

e-mail: ogomez@@tecnologicosucre.edu.ec

** Sucre Instituto Superior Tecnológico, Carrera de Electromecánica Industrial, Quito, Ecuador

e-mail: {jcarguachi; dherrera; dsaquinga}@tecnologicosucre.edu.ec

RESUMEN:

Este trabajo se enfoca en el desarrollo e implementación de un sistema, en software libre, para el monitoreo de señales A/D mediante la transmisión de datos y video. Para la transmisión de los datos entre los elementos primarios en el módulo y la interfaz se utiliza un protocolo de comunicación RS-232 (Recommended Standard-232), el video se transmite a través de una cámara IP (Internet Protocol), compatible con el nivel de red del modelo TCP/IP, basado en el estándar IEEE 802.11 (Wi-Fi). Para adquirir los datos y video del módulo, se diseña una interfaz en el entorno de desarrollo NetBeans, Java.

Los datos monitoreados en el proceso fueron: la temperatura y humedad del ambiente, así como velocidad de un motor DC. Los resultados se obtuvieron en base a la observación, análisis de datos y medición; además de métodos de inducción, deducción y síntesis. Finalmente se determinó que los parámetros en transmisión digital y analógica, fueron obtenidos en tiempo real, para constatarlo; se utilizó el software, Wireshark, que permitió comprobar que el tráfico de los datos en la red inalámbrica y serial; presentaron un promedio de tiempo de transmisión de 0.000193 milisegundos, en las dos redes configuradas.

Palabras clave: Comunicación, Java, Monitoreo, Microcontrolador, Transmisión.

ABSTRACT:

The research work in hand focuses on the development and implementation of a free software system for monitoring A / D signals, through the transmission of data and video.

For the transmission of the data between the primary elements in the module and the interface, the RS-232 (Recommended Standard-232) communication protocol was used. The video was played on an IP camera (Internet Protocol), which is compatible with the network level in the TCP / IP model, based on the IEEE 802.11 (Wi-Fi) standard. For fetching the data and video from the module, the interface was designed in the NetBeans, Java development environment.

The following data was monitored during the process: temperature, humidity, and speed of a DC motor. Results were obtained based on observation, data analysis and measurement, as well as induction, deduction and synthesis methods. The parameters in the digital and analog transmission were obtained in real time. Data traffic in the wireless and serial networks was verified through the Wireshark software, resulting in an average transmission time of 0.000193 milliseconds, in both set-up networks.

Keywords: Communication, Java, Monitoring, Microcontroller, Transmission.

Introducción

Para el Monitoreo de señales A/D con transmisión de video basado en el estándar IEEE 802.11, mediante software libre, se ha desarrollado una interfaz para la adquisición de datos en tiempo real.

El presente trabajo aborda dos aspectos importantes a nivel industrial los cuales están revolucionando el sistema productivo, el uso de la nueva tecnología y el correcto manejo de la información obtenida mediante la misma. "A comienzo del 2010 el concepto de Industria 4.0 ha ganado popularidad en Europa para referirse a un tipo de factoría automatizada, digitalizada, modulable y flexible", (**Gutiérrez, 2019**) por lo que a nivel mundial los procesos industriales se vuelven cada vez más competitivos, reduciendo los gastos de producción y mejorando la calidad del producto terminado.

En la actualidad, "los términos de industria 4.0 y el internet de las cosas o IOT van tomando fuerza, lo cual se traduce en una manufactura inteligente, automatizada y programada para cumplir procesos autónomos con el mínimo riesgo de error". (**Arancegui, 2016**). La toma de decisiones basada en información adquirida en tiempo real permite anticipar eventos a nivel de mercado, se cual sea el giro de negocio de la empresa, de esta manera la industria se hace más competitiva la misma permitirá desplazar a los negocios tradicionales generando modelos de negocios más flexibles y ágiles (**Pereson, 2019**).

"Los países con mayor desarrollo en la industria 4.0 son los europeos y Norteamérica; sus primeros indicios toman fuerza desde el año 2011, debido a su mayor desarrollo tecnológico de estas sociedades. Distinto resulta ser el caso de Latinoamérica, la cual en la actualidad no se encuentra preparada para dar avances contundentes". **(Bermúdez, 2019)**. Particularmente en Ecuador "se habla de cinco pilares esenciales que coadyuvarán el objetivo de alcanzar entrar en el mundo de la Revolución Industrial 4.0, como los son: Emprendimiento, Calidad, Innovación, Mercados, Productividad, Inversión". **(Véliz, 2018)**.

Por otro lado, las nociones más incipientes de interconexión entre máquinas-máquina o máquina-hombre (HMI) se remontan al tiempo de Nicolás Tesla y ya por el año 2008 surgen cambios importantes con métodos de conexión fiables. **(Soto Huerta, 2018)**. A nivel nacional en pequeñas y medianas empresas (pymes) hay proyectos de implementación de estos conceptos en muy baja escala, pero con un costo de implementación significativo, se puede citar el trabajo de tesis de maestría, mediante un protocolo de comunicación MQTT y el uso de un controlador lógico programable para el proceso industrial. **(López Flores, 2019)**, lo que da una esperanza a que las pymes se hagan más competitivas.

Las industrias multinacionales que aportan al desarrollo productivo del país tienen implementado estos conceptos con tecnología, equipos y procesos bien diseñados, lo que les da una gran ventaja comercial, la productividad y la calidad mejora notablemente con el uso de tecnología. **(Yagual Ramírez, 2019)** Haciendo una comparación de las dos realidades hay una diferencia muy grande en el aspecto de competitividad.

Falta un largo camino que recorrer en Ecuador para una inserción total de este sistema en los procesos industriales especialmente en las pymes, en la actualidad, aún no se cuenta con la tecnología necesaria, la cual en muchos casos implica una inversión muy alta, debido al costo de los equipos y software utilizados para el propósito. De acuerdo a los informes del INEC (2015), en el Ecuador el 66,7% las empresas invierten en TIC, siendo sus sectores más representativos la manufactura y el comercio quedando una brecha importante a cubrir de más del 30%. **(Bravo Alay, 2019)**.

En respuesta a esta limitante, se ha desarrollado e implementado un sistema que funciona mediante software libre para el monitoreo de señales A/D, a través de la transmisión de datos y video; esto representa una reducción de inversión significativa. Pallares en su trabajo de fin de master hace referencia la importancia del uso del software libre para lograr mayor competitividad en las pymes debido a su bajo costo de implantación. **(Pallarés Martínez, 2020)**.

El sistema consta de varias etapas de implementación y mejora continua, la primera consiste en la transmisión de datos y video, la segunda etapa, en la adquisición de los datos, monitoreo y control de los procesos en tiempo real. Cabe señalar, además, que se aumentará la lista de variables que intervengan en un proceso.

Materiales y Métodos

Para el diseño de la interfaz del sistema se utilizó Oracle Java JDK 8u111 (Java SE Development Kit), el mismo que contiene un conjunto de programas y librerías, que permiten compilar, depurar, monitorear y ejecutar los programas diseñados y además generar documentos útiles para el análisis de procesos.

NetBeans IDE 8.2 (Entorno de Desarrollo Integrado)

Una vez instalado el JDK de Java, se procede a instalar el entorno de trabajo NetBeans IDE la versión 8.2, este programa; es un software libre orientado a objetos, en este software permitió diseñar la interfaz gráfica y la comunicación RS-232 con el microcontrolador ATmega328.

Trasmisión de Video, basado en el Estándar IEEE 802.11

Para transmitir el video desde la planta industrial, se configuro una red TCP/IP, la cual es la base del internet. El mecanismo permite utilizar diferentes sistemas operativos tales como: Android, MacOS, MS Windows, GNU/Linux, estos pueden estar instalados en: PC personales, micro computadoras, Smartphone, Tablets.

En la figura 1 se observa la arquitectura de la comunicación entre los elementos primarios de la planta, y el dispositivo de trasmisión de video inalámbrico.

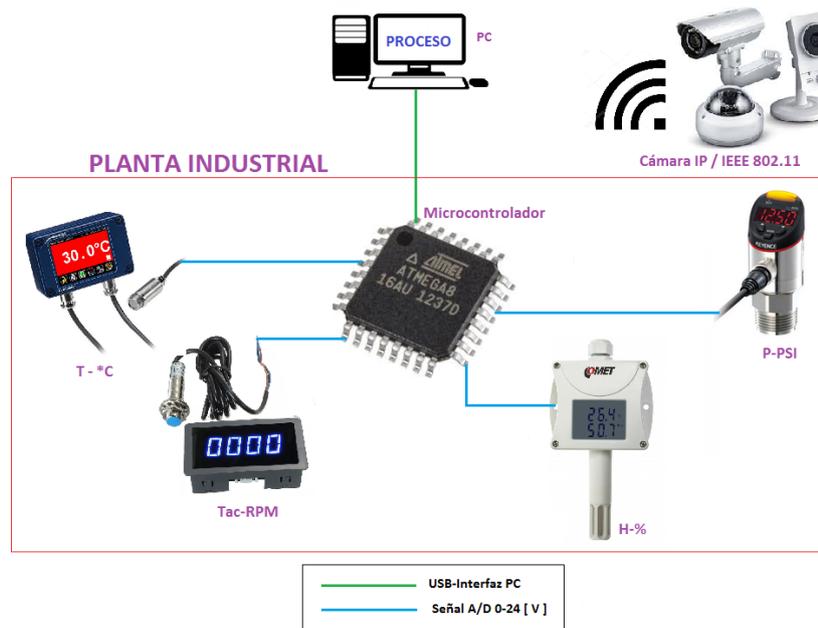


Fig. 1. Arquitectura de la comunicación del sistema de monitoreo

En el entorno de desarrollo de Java, para adquirir la señal que envía la cámara IP o un dispositivo móvil desde la planta, se debe abrir el puerto de internet 8080, concatenando la dirección IP del dispositivo, el cual está transmitiendo la señal de video. Se procede de esta manera para no interferir con el puerto 80, el cual es más utilizado para navegar por las páginas de Internet del dispositivo que está recepcionando los datos.

En la figura 2 se observa la clase realizada en Java, misma que recepta la señal de video del dispositivo Tx, con su respectiva dirección IP y por el puerto de entrada de los datos en este caso el 8080.

```

6 package camera_ip;
7 import com.googlecode.javacv.CanvasFrame;
8 import com.googlecode.javacv.FrameGrabber;
9 import com.googlecode.javacv.OpenCVFrameGrabber;
10 import com.googlecode.javacv.cpp.opencv_core;
11 import java.awt.GridLayout;
12 import java.util.logging.Level;
13 import java.util.logging.Logger;
14 /**
15  * @Oscar Gómez
16  */
17 public class camara_thread2 {
18     Camara_Panel panel=new Camara_Panel();
19     boolean back=false;
20     OpenCVFrameGrabber frameGrabber = new OpenCVFrameGrabber("http://192.168.43.1:8080/video?dummy=param.mjpg");
21     Thread cam= new Thread() {
22         @Override
23         public void run() {
24             try {

```

Fig. 2. Configuración de la clase de JAVA para la recepción de video

Para configurar de manera correcta la clase, y no tener inconvenientes de recepción de la señal de video, se debe colocar la IPv4 adecuadamente y con previa configuración, ya sea de forma manual o con la que entregan los dispositivos móviles automáticamente; como se observa en la figura 3.

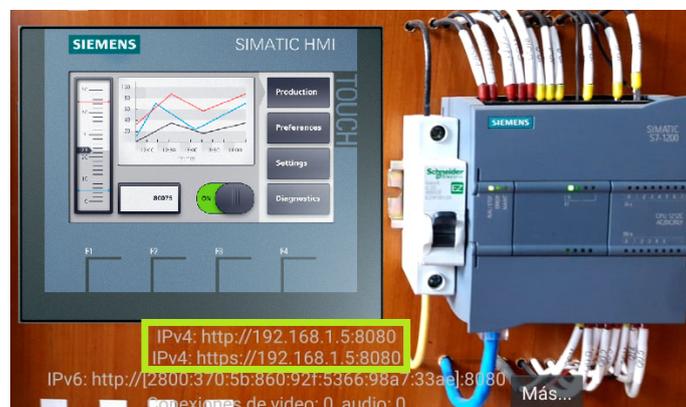


Fig. 3. Dirección IPv4 del dispositivo de trasmisión con su respectivo puerto de comunicación

Comunicación RS-232, con el microcontrolador ATmega328

Para la adquisición de los datos de los elementos primarios de la planta al microcontrolador y de este al sistema de monitoreo, se realizó con el estándar de comunicación RS-232, para la resección de los datos en Java; se realizó una conversión de la comunicación RS-232 a USB, que permite al microcontrolador ser reconocido por la PC como un dispositivo conectado a un puerto COM, aun cuando la conexión física sea mediante USB.

La velocidad de transmisión que se utilizó fue de 9600 baudios (bit por segundo), es importante configurar este valor, para que la parte de transmisión y la de recepción se sincronicen, en la figura 4 se observa la configuración de los baudios en la clase de la comunicación en Java y del microcontrolador.

```
Transmisor Tx
29 void setup() {
30 Serial.begin(9600); //Se inicia la comunicación serial
31 dht.begin(); //Se inicia el sensor
32 Wire.begin();
33 lcd.begin(16,2);
34 lcd.clear();
35 lcd.backlight();
36 // RPM
37 pinMode(encoder_pin, INPUT);
38 attachInterrupt(0, counter, RISING);
39 pulses = 0;
40 rpm = 0;
41 timeold = 0;
}

Receptor Rx
88 };
89 initComponents();
90 try {
91     arduinouno.arduinoRX("COM3", 9600, listener);
92 } catch (ArduinoException ex) {
93     Logger.getLogger(Camara_Panel.class.getName()).log(Level.SEVERE, null, ex);
94 } catch (SerialPortException ex) {
95     Logger.getLogger(Camara_Panel.class.getName()).log(Level.SEVERE, null, ex);
96 }
97
98 Serie.add(0,0);
99 Coleccion.addSeries(Serie);
```

Fig. 4. Configuración de la velocidad de transmisión de bits por segundo

Discusión de Resultados

Para el análisis de la transmisión de datos desde el microcontrolador a Java, se utilizó el software Wireshark, es un analizador de protocolos open-source su principal objetivo es el análisis del tráfico de red debido a que entiende la estructura de los protocolos, se analizó las 2 redes por separado, la red serial y la red inalámbrica para determinar el tiempo que un paquete de datos se demora en ir desde un host a otro host específico, esto se determina mediante el time delta del software.

Análisis del tráfico de datos en la red inalámbrica

En la Tabla 1 se observa los datos obtenidos en el análisis del tráfico de la red inalámbrica de la cámara IP, la cual indica todos los parámetros obtenidos por el software Wireshark como son: el tiempo de envío de los paquetes la fuente y el destino de los mismos, el protocolo de comunicación la longitud del paquete y el puerto de comunicación por donde accede los datos.

Tabla1. Resultados del análisis de tráfico de la red inalámbrica

TIME	SOURCE	DESTINATION	PROTOCOL	LENGTH	INFO
0.000000	192.168.1.8	192.168.1.12	TCP	1514	8080
0.000978	192.168.1.8	192.168.1.12	TCP	1514	8080
0.008878	192.168.1.8	192.168.1.12	TCP	1514	8080
0.008935	192.168.1.8	192.168.1.12	TCP	1514	8080
0.009095	192.168.1.8	192.168.1.12	TCP	1514	8080
0.009131	192.168.1.8	192.168.1.12	TCP	1514	8080
0.009179	192.168.1.8	192.168.1.12	TCP	1514	8080
0.009308	192.168.1.8	192.168.1.12	TCP	1514	8080
0.009363	192.168.1.8	192.168.1.12	TCP	1514	8080
0.009558	192.168.1.8	192.168.1.12	TCP	1514	8080

El tiempo que se demora en llegar los paquetes de datos de la cámara IP, a la interfaz de monitoreo es aproximadamente de 0,000193 milisegundos, con el análisis de los datos adquiridos.

De igual forma se realizó el análisis de los datos de la red serial el tiempo de envío de un paquete de datos es de 0,166009 milisegundos en la Tabla 2 se observa los parámetros medidos y los tiempos de ejecución de los mismos.

Tabla 2. Resultados del tráfico de red serial

TIME	SOURCE	DESTINATION	PROTOCOL	LENGTH	INFO
0.000000	2.1.2	host	USB	36	URB
0.125008	2.1.2	host	USB	36	URB
0.127008	2.1.2	host	USB	36	URB
0.127008	2.1.28	host	USB	36	URB
0.127008	2.1.2	host	USB	36	URB
0.127008	2.1.2	host	USB	36	URB
0.293017	2.1.2	host	USB	36	URB
0.587034	2.1.2	host	USB	36	URB
0.880051	2.1.2	host	USB	36	URB
1.125065	2.1.2	host	USB	36	URB

Interfaz para el monitoreo de los datos y video diseñada en Java

Al ejecutar el programa en Java se abre una ventana de inicio del sistema, donde se encuentra el título del proyecto y un botón, como se muestra en la figura 5. La función del botón es dar inicio a la clase de Java y librerías para la comunicación de la cámara IP y la comunicación serial.

En Java, existe el concepto de librerías. Una librería se puede decir, es un conjunto de clases que tiene una serie de métodos y atributos, la función de las librerías es facilitar muchas operaciones, reutilizando la programación; pues bien, así es como se unifico todo el código en un solo programa principal para realizar el sincronismo de las dos comunicaciones con las librerías.



Fig. 5. Ventana de inicio del sistema de monitoreo

Al instante de sincronizarse las dos redes, se abre una segunda ventana como se observa en la figura 6, en la cual se observa todos los datos adquiridos por los elementos primarios de la planta; en este caso particular la temperatura, humedad del ambiente y la velocidad de un motor DC, adicionalmente se generó en el mismo sistema las gráficas de las dos señales analógicas.

En el apartado izquierdo de la figura 6 se visualiza la señal de video, donde la cámara IP, enfoca a las variables de la planta las mismas que son enviadas por la comunicación serial a la interfaz del sistema.



Fig. 6. Sistema de monitoreo

Conclusiones

- La elaboración del sistema tiene grandes ventajas al momento de implementar en una industria, mediante software libre se puede monitorear procesos y variables en tiempo real.
- NetBeans es un entorno profesional con funciones avanzadas para el desarrollo de interfaces gráficas, este IDE incluye librerías que facilita la comunicación de las dos redes de datos; inalámbrica y la Rs-232, de igual forma NetBeans es fiable para la recepción de paquetes de datos.
- En la etapa de pruebas el sistema no genera ningún tipo de error o conflicto con los dispositivos empleados para el monitoreo de los parámetros establecidos.
- La velocidad de respuesta presentó un promedio de tiempo de transmisión de 0.000193 milisegundos en la transmisión de datos.

Referencias Bibliográficas

- Arancegui, M. N. (2016). Reflexiones sobre la Industria 4.0 desde el caso vasco. *Revista vasca de economía*, 142-173.
- Bermúdez, G. M. (2019). ¿ Nuevo modelo de seguridad social en el contexto de la industria 4.0?. . *Revista Internacional y Comparada de Relaciones Laborales y Derecho del Empleo*.
- Bravo Alay, K. V. (2019). Economía circular y la industria 4.0 como estrategia del Comercio Internacional en el Ecuador. (*Bachelor's thesis*). Universidad de Guayaquil Facultad de Ciencias Administrativas, Guayaquil.
- Gutiérrez, R. T. (2019). La fabricación abierta:¿ un camino alternativo a la industria 4.0?. *Revista iberoamericana de ciencia, tecnología y sociedad*, 263-285.
- López Flores, M. X. (2019). Industria 4.0 para la monitorización de un proceso industrial. (*Master's thesis*). Universidad Técnica de Ambato, Ambato.
- Pallarés Martínez, V. (2020). Implementación de la Industria 4.0 en PYMES del Sector Productivo. (*Master Thesis*). politecnico Di Milano, Milán.
- Pereson, M. (2019). Sistemas embebidos y las opciones de software libre para aplicaciones IoT. *AJEA*, 1.
- Soto Huerta, A. V. (2018). Internet de las cosas empresarial; orígenes, evolución y tendencias., (pág. 59).
- Véliz, M. A. (2018). Perspectivas de la economía digital en Latinoamérica: Caso Ecuador. *investigación y pensamiento crítico*, 28-43.
- Yagual Ramírez, M. J. (2019). Diseño e implementación de un sistema Scada para el proceso de mezcla de pintura utilizando herramientas de la industria 4.0. (*Bachelor's thesis*). Universidad Estatal Península de Santa Elena, Santa Elena.