

Tipo de artículo: Artículo original
Temática: Desarrollo de aplicaciones informáticas
Recibido: 16/06/2023 | Aceptado: 15/09/2023

Sistema de Comprobación en tiempo real de señales de un autopiloto de una aeronave

System for real-time verification of signals from the autopilot of an aircraft

Arlet Montiel Ramos^{1*} <https://orcid.org/0009-0000-5105-5747>

Alian Ernesto Matos Rodríguez ¹ <https://orcid.org/0000-0001-7333-7411>

Bárbaro Rodríguez Cárdenas ¹ <https://orcid.org/0009-0002-9458-1530>

¹ Centro de Investigación y Desarrollo Naval. Calle Estrada Palma No. 13, Casa Blanca, Regla, La Habana, Cuba. {[montiel96](mailto:montiel96@nauta.cu), [cuaematos](mailto:cuaematos@nauta.cu)}@nauta.cu , aid171alt@gmail.com

*Autor para la correspondencia. (montiel96@nauta.cu; montielarlet94@gmail.com)

RESUMEN

Las avanzadas técnicas de visualización permiten que la información relevante se presente en el formato y en el momento adecuado por medio de interfaces visuales para que las operaciones se ejecuten de forma eficiente. Estas técnicas ayudan a prevenir errores y malas interpretaciones que podrían conducir a situaciones de inseguridad o fallos en los equipos. Las mismas constituyen una de las técnicas más relevantes dentro del campo de la informática, en muchos casos empleando hardware especializados como son las Computadoras de Placas Reducidas. Se caracterizan por su bajo costo, alto rendimiento y menor consumo de potencia con respecto a las computadoras convencionales. Lo que puede propiciar un resultado

eficiente y con un menor costo. La principal contribución de este trabajo es un software capaz de comprobar y representar visualmente el estado técnico del autopiloto de una aeronave empleando el computador de placa reducida Lattepanda V1. Para la solución se empleó métodos de programación paralela con Thread en Visual Studio y .Net con el Framework 4.5, lo que contribuyó al uso eficiencia de los recursos del SBC y aumentó la rentabilidad económica en comparación con las computadoras personales.

Palabras clave: Interfaz hombre-máquina; Lattepanda V1; estándar de comunicación RS485; computadora de placa reducida.

ABSTRACT

Advanced visualization techniques allow relevant information to be presented in the right format and at the right time through visual interfaces so that operations can be executed efficiently. These techniques help prevent errors and misinterpretations that could lead to insecurity situations or equipment failures. They constitute one of the most relevant techniques in the field of computing, in many cases using specialized hardware such as Small Plate Computers. They are characterized by their low cost, high performance and lower power consumption compared to conventional computers. This can lead to an efficient result and at a lower cost. The main contribution of this work is a software capable of checking and visually representing the technical status of the autopilot of an aircraft using the Lattepanda V1 single board computer. For the solution, parallel programming methods were used with Thread in Visual Studio and .Net with Framework 4.5, which contributed to the efficient use of SBC resources and increased economic profitability compared to personal computers.

Keywords: Human-machine interface; Lattepanda V1; communication standard RS485; single board computer.

Introducción

La interfaz de usuario es el medio por el cual las personas interaccionan con las máquinas. Estas han evolucionado desde la inicial consola de comandos, donde los programas se empleaban mediante la manipulación del teclado y *mouse*. Con el devenir de los años dichas interfaces fueron progresando hasta llegar a convertirse en las Interfaces Hombre-Máquina (HMI, por sus siglas en inglés) que hoy comúnmente se emplean. Son una interfaz gráfica, muy simple, que trata de diseñar y desarrollar sistemas informáticos para apoyar la necesidad de las personas de usar la máquina para que puedan planificar prolíficamente sus actividades con seguridad, además es la plataforma que se utiliza en los sistemas de control (Mahmud et al., 2020) para mostrar la información de los procesos en tiempo real, mediante diagramas, esquemas y animaciones en pantalla.

La interfaz para la interacción hombre-máquina se ha convertido en un área destacada de investigación debido al rápido crecimiento de la automatización y la robótica en las últimas décadas (Mahmud et al., 2020),(Montero and Trajano, 2019), siendo objeto de investigación en neurociencia, aeroespacial, robótica e inteligencia artificial (Scibilia et al., 2022), además a partir de una exhaustiva investigación se determinó que las HMI son utilizadas en varias esferas, incluidas plantas de fabricación (Pina Beti, 2022), máquinas expendedoras (HARO CUADRADO, 2022),(Chanchicocha and Fabricio, 2022), la industria de alimentos y bebidas (Matilla Buenaposada, 2022), (De La Cruz Intriago et al., 2022), productos farmacéuticos (Concepción et al., 2022) y otros servicios públicos (Cardona Morales and Pérez Roa, 2020) ,(Sainz Padrón et al., 2020),(Rodríguez Esteva, 2020). La HMI permite la conversión de gran cantidad de datos complejos en información accesible para que el operario la utilice como herramienta para controlar procesos, es por ello, que mientras más receptivo y fácil sea la HMI, más eficiente y beneficioso será el trabajo (HARO CUADRADO, 2022). Pueden ser soportadas sobre diferentes hardware en dependencia de las características del lugar donde se emplee, entre estos podemos encontrar las Computadoras de Placas Reducidas (SBC, por sus siglas en inglés), que no es más que una computadora construida en una sola placa de circuito, que se utiliza para la creación de interfaces de usuario (Paunski and Angelov, 2019), donde se están volviendo cada vez más avanzados y fuertes muy próximos a los sistemas basados en x86 de Intel, que por décadas han sido considerados como los mejores (Wazir et al., 2020), además por su relación calidad-precio y por su

eficiencia energética se han visto beneficiado en su rendimiento (Basford et al., 2020). Esta arquitectura no se usa en las computadoras personales sino más bien en entornos industriales o en sistemas embebidos, pues sirven como controladores e interfaces. Por los elevados niveles de integración y reducción de componentes, son pequeños, livianos y confiables, y debido a su arquitectura RISC consumen menos energía y presentan un bajo costo con respecto a las computadoras de múltiples tarjetas (Wazir et al., 2020).

A partir de la tecnología descrita anteriormente y teniendo en cuenta la necesidad de comprobar la información de los procesos, sensores y de recopilar datos en tiempo real del autopiloto de una aeronave, la cual fue modernizada por el déficit de piezas de repuesto, su inestabilidad en el funcionamiento y la obsolescencia de su tecnología, surge la necesidad de desarrollar una solución basada en HMI para el diagnóstico de los autopilotos modernizados que permita su validación y cumpla con el tiempo máximo de actualización de la información de 50 milisegundos. El objetivo principal de la presente investigación es el desarrollo de un software de comprobación que presenta como algoritmos principales la recepción y representación de los datos del autopiloto modernizado, encargado de garantizar las exigencias de tiempo y los parámetros de comprobación apoyada en una computadora de placa reducida LattePanda V1. Dicho resultado tributa a la completa disposición combativa del autopiloto modernizado de las Fuerzas Armadas Revolucionarias.

Métodos o Metodología Computacional

El software de comprobación constituye una herramienta de certificación encargada de determinar el estado técnico del autopiloto. Para un mejor aprovechamiento de los recursos computacionales del SBC, se utilizó HMI para la visualización de los parámetros a controlar, un gestor de base de datos *SQLite*, para el almacenamiento de los datos generados y un fichero en formato pdf, para el reporte de las comprobaciones. Como dispositivo hardware a emplear se optó por la placa reducida LattePanda; la cual es de fácil uso, pequeño tamaño y ofrece las posibilidades avanzadas de los mini ordenadores (Concepción et al.).

Para la búsqueda de información a incluir en la investigación se utilizaron fuentes principales como: *Google Scholar* para identificar otras fuentes de interés científico con énfasis en las bases de datos electrónicas, *Science Direct*, *Mendeley* y *IEEE Xplore*. La literatura estudiada comprende artículos de revistas, eventos, y tesis de maestría y postgrado. Se consideraron principalmente las publicaciones comprendidas entre 2019 y 2023 en idioma inglés y español. Los términos de búsqueda empleados para encontrar estudios relativos al tema fueron: “interfaz hombre máquina”, “computador de placa reducida”, “estándar RS 485” y “comunicación UART”. Se acotaron los términos entre comillas, así mismo, se emplearon los operadores AND y OR para búsquedas avanzadas en consultas “interfaz hombre máquina” AND “computadora de placa reducida” OR “estándar RS 485”. Se seleccionaron los trabajos a partir de las coincidencias, reduciendo el volumen de trabajos considerando la relevancia de los aportes del análisis del contenido.

Diseño e Implementación de la solución

La conformación de la estructura de la transmisión de datos entre el autopiloto y el software de comprobación, constituye la etapa más importante en su diseño, pues en esta se organiza y se definen los elementos a contener en la trama reflejado en la Figura 1, además en esta etapa se define el enlace de comunicación y el soporte sobre el cual se ejecutará dicho sistema.

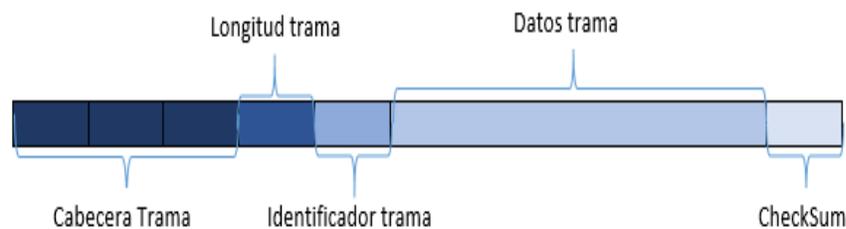


Fig. 1 – Estructura de la trama de la Recepción datos y de Transmisión.

Como se muestra en la Figura 1 la trama de datos está compuesta por cabecera, longitud, datos a procesar y *Checksum*, la misma cuenta con 176 bytes que se transmiten cada 50 milisegundos a 115200 baudios lo que significa que es capaz de recibir un máximo de 115200 bits por segundo.

Durante mucho tiempo las computadoras de placa única han estado estrechamente vinculadas a las placas madres. Realizaban tareas similares y se han esforzado por alcanzar velocidades más rápidas, generar mejores gráficos y ofrecer más memorias. La llegada de los SBC los cuales logran reducir su tamaño y ser capaces de realizar tareas básicas de forma aceptable y una inversión financiera mucho menor, resultando una de las mejores opciones para los problemas que se presentan en la actualidad.

LattePanda constituye uno de los tantos SBC de hoy en día, siendo una computadora súper pequeña con el primer sistema operativo Windows 10 en el mundo, presenta un tamaño mini, pero no se puede dudar de su capacidad, y podría decirse que esta mini PC (Personal Computer) es capaz de llevar a cabo actividades que normalmente se realizan en una PC en general (Nasution et al., 2019), se puede usar como placa controladora porque está diseñada para ser compatible con Arduino, por lo que es una de las opciones para placas controladoras más económicas. LattePanda tiene las siguientes especificaciones: Procesador: *Intel Cherry Trail Z8300* de cuatro núcleos a 1,8 GHz, Sistema operativo: edición completa pre activada preinstalada de Windows 10, RAM: 2GB DDR3L, Capacidad de almacenamiento: 32 GB, USB 3.0*1, USB 2.0*2, Wifi y Bluetooth 4.0, 2 GPIO para chip Intel, 20 GPIO para Arduino, Potencia: 5v/2^a (DFRobot, 2022),(Tjandra et al., 2019). Ver Figura 2.

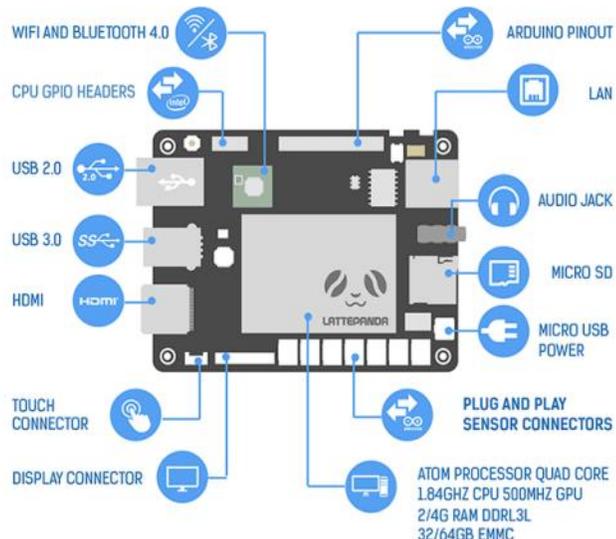


Fig. 2 – Estructura del LattePandas.(Nasution et al., 2019)

La comunicación del SBC con el autopiloto es mediante el protocolo de comunicación Transmisor-Receptor Asíncrono Universal (UART, por sus siglas en inglés) y como estándar el RS 485, el cual es muy popular debido a su capacidad para comunicarse a largas distancias (1200m), su capacidad multipunto y su inmunidad al ruido (Marias, 2008),(Marín Hernández, 2022), además emplea transmisión diferencial de datos, por lo que es ideal para transmisiones a altas velocidades, ambientes ruidosos y con gran interferencia electromagnética (Quinga Loya and Torres Torres, 2022).

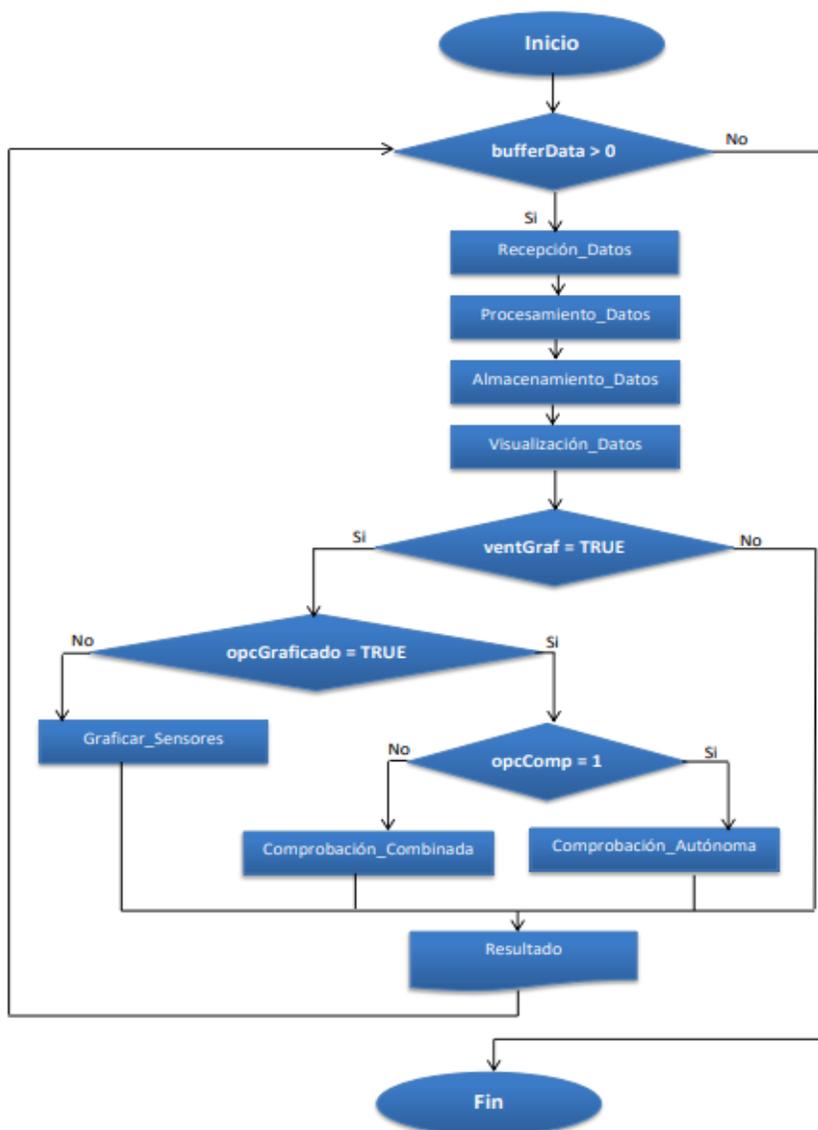


Fig. 3 –Representación del flujo de trabajo del software de comprobación.

El proceso de implementación se realiza en el entorno de desarrollo de Visual Studio por lo que permitió editar, compilar y depurar el código en el lenguaje C# con sistema operativo Windows 10 Home.

Concebida la trama de datos, la implementación se organiza en seis etapas (Recepción, Procesamiento, Almacenamiento, Visualización, Representación Gráfica y Comprobaciones de Datos), como se demuestra en el diagrama de la Figura 3 donde se representa el flujo de los procesos por el cual transitan los datos recibidos. Inicialmente, los valores son recibidos, se analizan si corresponden con la estructura definida empleando varios métodos de la biblioteca de clases *SerialPort* como *ReadByte*, *DiscardInBuffer*, *BytesToRead* entre otros y luego procesar los datos facilitando la identificación de cada sensor del autopiloto, para ser almacenados, representados numéricamente y graficados, además de comprobar sus parámetros y sensores.

Por el elevado procesamiento de datos se hace necesaria la implementación de un método de programación paralela para acelerar la ejecución de la función que más tiempo consume evitando una colisión de datos y logrando una distribución de tareas entre sus núcleos, para un mejor aprovechamiento de los recursos computacionales.

Una de las formas de realizar paralelismo es por medio de ejecutar aplicaciones en diferentes y concurrentes hilos de ejecución. Para ello se trabaja sobre el entorno de desarrollo de Visual Studio y .NET proporcionando un tiempo de ejecución, tipos de bibliotecas de clases y herramientas de diagnóstico (Autores., 2023). Se pudo determinar que el proceso de recepción de datos es el método que más tiempo consume por lo que la función de recepción se incluyó dentro de la biblioteca de clase *Thread* y se logró ejecutar en el mismo período el método de procesamiento de datos reduciendo considerablemente el tiempo de recepción. A su vez fue necesario agregar la clase *Timer* la cual se ejecuta en su propio hilo empleando sus métodos, con el objetivo de ganar en tiempo para realizar los procesos de graficado y comprobaciones de parámetros, teniendo en cuenta que necesitan actualizar y representar más rápido para evitar una colisión de

datos. Es por ello, que las aplicaciones que usan varios hilos responden mejor a la interfaz de usuario, esta permanece activa a diferencia de las tareas que demandan un mayor uso del procesador que se ejecutan en subprocesos separados. Para dar solución a lo antes mencionado fue necesario el empleo de las clases *System.Threading.Tasks.Parallel* y *System.Threading.Tasks.Task*, y los métodos *Thread.Start* y *Thread.Sleep*, *BeginInvoke*, entre otros.

Resultados y discusión

Las aplicaciones se benefician enormemente de una buena asignación de recursos del procesador, pues muchas ejecutan varios procesos simultáneos. Tal es el caso del software de comprobación desarrollado, en el que se realiza varios métodos como: Recepción de Datos, Procesamiento de Datos, Representación Gráfica de Datos, Visualización de Datos, así como la Comprobación del Autopiloto y el Almacenamiento de Datos, todas las tareas mencionadas se realizan en un tiempo inferior a los 50 milisegundos establecido como requisito principal. Con el objetivo de verificar que el software desarrollado cumple con las exigencias de tiempo planteado y determinar la configuración óptima del rendimiento se realizan pruebas de experimentos tomando como factores controlable los siguientes:

1. Afinidad del procesador: donde el sistema operativo vincula un proceso a los procesadores especificados ya sea por defecto o por los que el usuario haya seleccionado.
2. Prioridad del Proceso: como su nombre lo indica es la prioridad asignada por el sistema operativo para ejecutarse, teniendo en cuenta que los procesos de mayor prioridad se ejecutan antes que los procesos con menor prioridad.

Para cada uno de los factores seleccionados en el experimento existen dos niveles, uno de ellos se considera como el valor bajo y otro como el alto. En la Tabla 1 se muestran los factores con los niveles asociados a cada uno de ellos.

Tabla 1- Niveles de los factores controlables seleccionados para el experimento.

Factor	Nivel bajo	Nivel alto
Afinidad del procesador	2	4
Prioridad del proceso	Por debajo de lo normal (DN)	Por encima de los normal (EN)

Se realizó una fase de experimento bajo el principio de repetición para minimizar el error experimental, donde se tomaron 1200 muestras aproximadamente durante un minuto. Las mismas se realizaron en un escenario de experimentación con las siguientes condiciones: SBC LattePanda V1, con Sistema Operativo *Windows 10 Home*, memoria RAM DDR3 de 2 GB, memoria de almacenamiento eMMC de 32 GB, procesador Intel *Cherry Trail Z8350 Quad Core*.

En la Figura 4 se muestra el comportamiento en función del tiempo en milisegundos para el factor controlable prioridad por encima de lo normal para dos procesadores (X01) y cuatro procesadores (X02), donde se refleja que la media obtenida es de 16, 1546 milisegundos y 14.0602 milisegundos respectivamente así como un 100 % de la trama de datos adquirida para ambas condiciones de prueba, concluyendo que en las condiciones realizadas los métodos del sistema cumplen con un tiempo inferior a los 50 milisegundos siendo más óptimo para X02 con una desviación estándar de 1.6388 milisegundos lo que indica que hay un rendimiento más estable o consistente dentro del sistema.

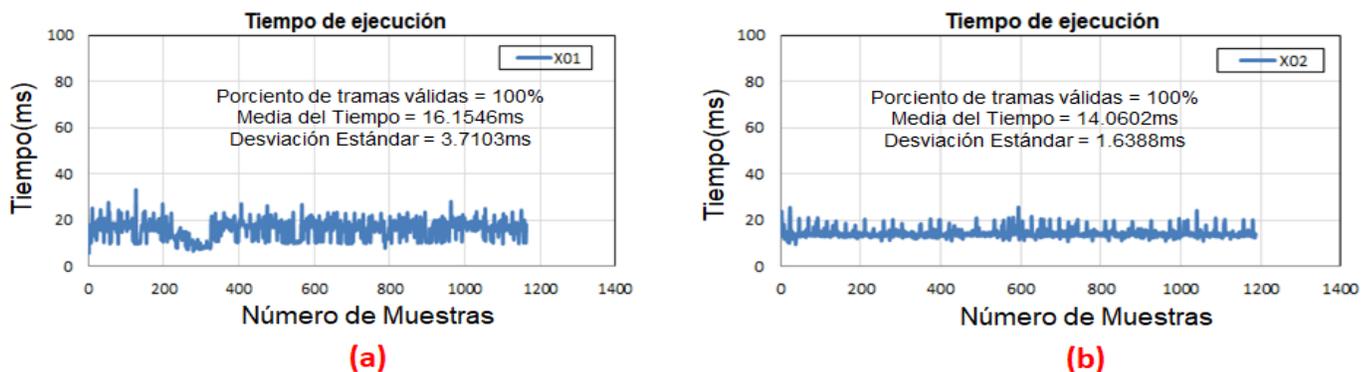


Fig. 4 – Comportamiento del tiempo de ejecución para prioridad por encima de lo normal. (a) 2 procesadores (b) 4 procesadores.

En la Figura 5 se muestra el comportamiento en función del tiempo en milisegundos para el factor controlable prioridad por debajo de lo normal para dos procesadores (X03) y cuatro procesadores (X04), donde se refleja que la media obtenida es de 16, 5822 milisegundos y 16, 5774 milisegundos respectivamente así como un 100 % de la trama de datos adquirida para ambas condiciones de prueba, concluyendo que en las condiciones realizadas los métodos del sistema cumplen con un tiempo inferior a los 50 milisegundos siendo más eficiente para las condiciones de prioridad por debajo de lo normal para cuatro procesadores con una desviación estándar 2.8078 ms inferior al operar con dos procesadores.

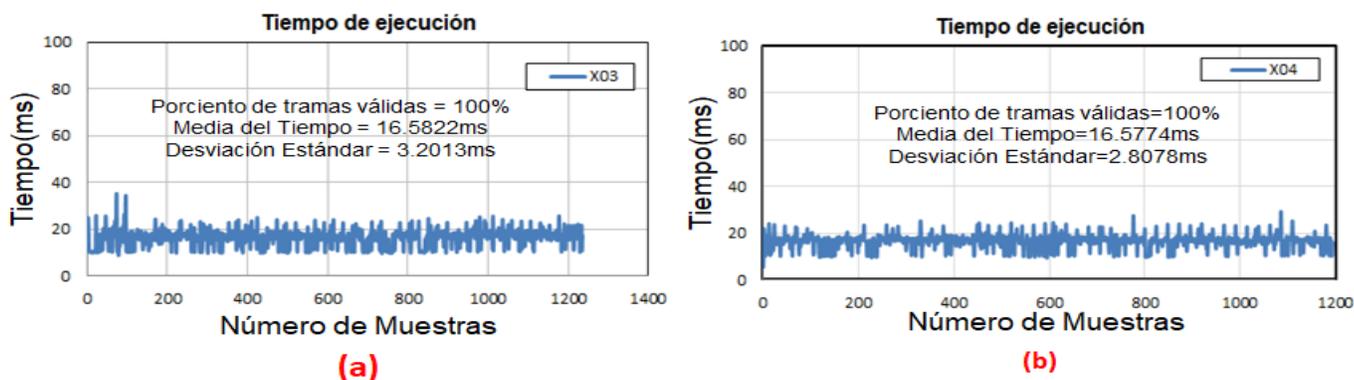


Fig. 5 – Comportamiento del tiempo de ejecución para prioridad por debajo de lo normal. (a) 2 procesadores (b) 4 procesadores.

Como resultado del diseño experimental se pudo determinar que la configuración óptima donde el software obtiene mejores resultados se realiza con prioridad por encima de lo normal y con afinidad de cuatro procesadores llegando a reducir hasta 2.4 milisegundos aproximadamente el tiempo de ejecución de sus procesos como se muestra en la Figura 6, además , se muestra una disminución de la distribución del tiempo de la desviación estándar en la Figura 4 (b), demostrando que en esas condiciones es la configuración ideal para un rendimiento eficaz del funcionamiento del software.

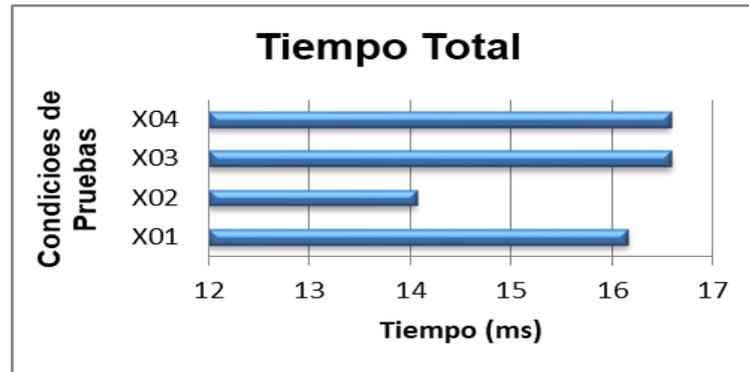


Fig. 6 – Gráfica de la media del tiempo de trabajo en las diferentes condiciones de configuración.

Como factores no controlables identificados son:

1. Procesos ejecutados por el sistema operativo: al ejecutarse el software hay procesos en ejecución en el sistema operativo, que consumen recursos y no pueden ser detenidos.
2. Tráfico en la red: los datos del autopiloto son transmitidos hacia la computadora por protocolo de comunicación UART mediante la interfaz de comunicación RS-485. En el momento de transmisión no se puede controlar el tráfico que existe en la red en ese momento.

Demostrada la configuración óptima es necesario validar el método de recepción de datos siendo el más importante para él, para ello se tomaron 7524 muestras para 50 milisegundos de muestreo lo que indica 376.2 segundos de vuelo y se analizaron en la herramienta *Matlab* permitiendo obtener el número de muestras válidas. En la Figura 7 se representa el comportamiento de las tramas de datos recibidas con respecto al tiempo total de trabajo del autopiloto.

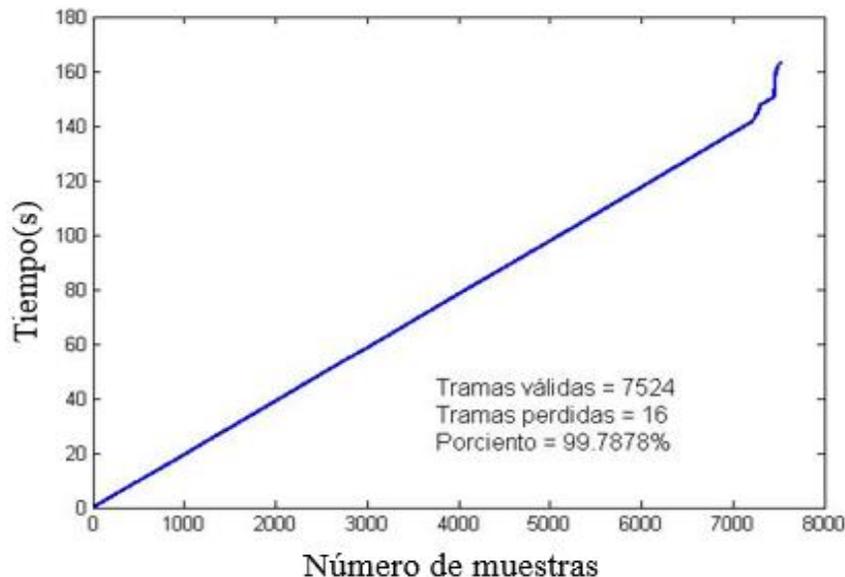


Fig. 7 – Gráfica de la variable de tiempo de trabajo del autopiloto enlazado con el supervisorio de comprobación.

Como resultado del muestreo realizado se obtiene una sincronización satisfactoria de 7524 tramas de datos de las cuales se perdieron 16, lo que representa un 99% de las tramas válidas para una evaluación con profundidad de los resultados alcanzados.

Conclusiones

El software de comprobación propuesto en la presente investigación garantiza la recepción de los datos permitiendo realizar las comprobaciones del autopiloto para garantizar su completa disposición combativa en el tiempo establecido y con la configuración óptima para su mejor rendimiento. Queda validado mediante mediciones reales en la fase experimental determinando la influencia de los factores de programación y dispositivo en los tiempos de respuesta del sistema, al emplear el cómputo de placa reducida LattePanda V1 permitió obtener una solución de hardware la cual requiere un bajo costo de implementación por el empleo de computadoras de placa reducida y software libre lo que propicia un mejor aprovechamiento de los recursos de cómputo de los dispositivos del hardware.

Referencias

- Autores., C. D. 2023. *Programación En Paralelo En .Net: Guía De La Documentación*. [Online]. Available: <https://Learn.Microsoft.Com/Es-Es/Dotnet/Standard/Parallel-Programming/> 2023].
- Basford, P. J., Johnston, S. J., Perkins, C. S., Garnock-Jones, T., Tso, F. P., Pezaros, D., Mullins, R. D., Yoneki, E., Singer, J. & Cox, S. J. 2020. Performance Analysis Of Single Board Computer Clusters. *Future Generation Computer Systems*, 102, 278-291.
- Cardona Morales, A. I. & Pérez Roa, J. J. 2020. Sistema Inteligente Basado En Industria 4.0 Para Control De Alumbrado Externo.
- Concepción, F. S., Aguilar, R. P., Moreno, A. S. P., Cordoví, A. A., Castro, J. C. C. & Llanes-Santiago, O. 2022. Metodología Para La Automatización De Procesos Tecnológicos En La Industria Farmacéutica Cubana. *Ingeniería Industrial*, 43, 1-14.
- Chanchicocha, P. & Fabricio, A. 2022. Construcción De Un Sistema De Refrigeración En Procesos De Corte, Controlado Y Monitoreado A Través De Un Sistema Hmi.
- De La Cruz Intriago, E. D., García Betancourt, J. A. & Yumbla, F. 2022. *Diseño De Una Dosificadora De Líquidos Automática Con Caudalímetros Para El Cálculo De Volumen De Envasado, Controlado Por Un Plc Y Hmi*. Espol. Fimcp.
- Dfrobot. 2022. Selection Guide For Lattepanada Board. *Blog* [Online]. Available From: <https://www.lattepanada.com/blog-308749.html>.
- Haro Cuadrado, D. L. 2022. *Sistema De Control Con Interfaz Hmi Para El Proceso De Enfriamiento De Las Unidades De Generación De La Hidroeléctrica Topo*. Quito, Ecuador: Universidad Tecnológica Israel.
- Mahmud, S., Lin, X. & Kim, J.-H. Interface For Human Machine Interaction For Assistant Devices: A Review. 2020 10th Annual Computing And Communication Workshop And Conference (Cccw), 2020. Ieee, 0768-0773.
- Marias, H. 2008. Rs-485/Rs-422 Circuit Implementation Guide Application Note. *Analog Devices*, 12.
- Marín Hernández, F. 2022. Diseño E Implementación De Un Tablero Electrónico Para La Comunicación De Posición Absoluta.
- Matilla Buenaposada, O. F. 2022. Puesta En Marcha Y Control De Un Horno Industrial Usando Un Automata Programable Y Una Pantalla Hmi.

- Montero, B. & Trajano, D. 2019. Sistema De Manufactura Flexible Orientado A Industria 4.0.
- Nasution, T., Muchtar, M., Seniman, S. & Siregar, I. 2019. Monitoring Temperature And Humidity Of Server Room Using Lattepanada And Thingspeak. *Journal Of Physics: Conference Series*. Iop Publishing, 012068.
- Paunski, Y. K. & Angelov, G. T. 2019. Performance And Power Consumption Analysis Of Low-Cost Single Board Computers In Educational Robotics. *Ifac-Papersonline*, 52, 424-428.
- Pina Beti, V. 2022. Proyecto De Control-Supervisor Mediante Pantallas Hmi Para Una Planta Cementera.
- Quinga Loya, D. I. & Torres Torres, S. F. 2022. *Implementación De Un Sistema De Comunicación Mediante Protocolo Modbus Rtu Para La Obtención De Parámetros Eléctricos De Una Estación Fotovoltaica De 3kw*. Ecuador: Latacunga: Universidad Técnica De Cotopaxi (Utc).
- Rodríguez Esteva, A. E. 2020. Simulación Y Control De Sistema De Saneamiento.
- Sainz Padrón, L., Sainz Padrón, L., Cabrera Rodríguez, N. & González López, Y. 2020. Sistemas De Expertos Desarrollados En El Mundo Para La Detección, El Diagnóstico Y El Tratamiento Del Cáncer. *Revista Cubana De Información En Ciencias De La Salud*, 31.
- Scibilia, A., Pedrocchi, N. & Fortuna, L. 2022. Human Control Model Estimation In Physical Human–Machine Interaction: A Survey. *Sensors*, 22, 1732.
- Tjandra, L., Affandi, A., Arifin, I. & Adityo, M. F. 2019. Designing Interface Of Rhino Robot’s Operator Control Unit Using Development Board Lattepanada. *Iptek Journal Of Proceedings Series*, 48-53.
- Wazir, S., Imran, H. A., Latif, U., Mujahid, U. & Bilal, M. 2020. Single Board Computers (Sbc): The Future Of Next Generation Pedagogies In Pakistan. *Arxiv Preprint Arxiv:2008.06576*.

Conflicto de interés

Los autores autorizan la distribución y uso de su artículo.

Contribuciones de los autores

1. Conceptualización: Arlet Montiel Ramos.
2. Curación de datos: Alian Ernesto Matos Rodríguez.
3. Análisis formal: Arlet Montiel Ramos.

4. Adquisición de fondos: -
5. Investigación: Arlet Montiel Ramos.
6. Metodología: Alian Ernesto Matos Rodríguez.
7. Administración del proyecto: Arlet Montiel Ramos
8. Recursos: Bárbaro Rodríguez Cárdenas
9. Software: Arlet Montiel Ramos
10. Supervisión: Bárbaro Rodríguez Cárdenas.
11. Validación: Alian Ernesto Matos Rodríguez.
12. Visualización: Arlet Montiel Ramos
13. Redacción – borrador original: Arlet Montiel Ramos
14. Redacción – revisión y edición: Alian Ernesto Matos Rodríguez.