

CIATEQ, A. C. Centro de Tecnología Avanzada  
Dirección de Posgrado



*Implementación de protocolo de comunicación para  
pruebas funcionales entre sistema embebido de bajo costo  
y secadora convencional de ropa*

TESIS QUE PRESENTA

**Ing. Hugo Enrique García Rico**  
**Asesor: Dr. Marco Antonio Garduño Ramón**

Para obtener el grado de

**Maestro en**  
**Sistemas Inteligentes Multimedia**

Querétaro, Querétaro  
julio, 2021

## CARTA DE LIBERACIÓN DEL ASESOR

Querétaro, Querétaro, 19 de Julio del 2021.

Mtro. Geovany González Carlos  
Coordinador Académico de Posgrado  
CIATEQ, A.C.

Los abajo firmante, miembro del Comité Tutorial del Ing. Hugo Enrique García Rico, una vez revisado su Proyecto Terminal de tesis/tesina, titulado **“Implementación de Protocolo de Comunicación para Pruebas Funcionales entre Sistema Embebido de Bajo Costo Y Secadora Convencional de Ropa”** autorizo que el citado trabajo sea presentado por el alumno para su revisión, con el fin de alcanzar el grado de **Maestro en ciencias**.

Sin otro particular por el momento, agradezco la atención prestada.

Firma



---

Dr. Marco Antonio Garduño Ramón

Asesor Académico

## CARTA DE LIBERACIÓN DEL REVISOR



Querétaro, Querétaro, 5 de Julio del 2021.

Dra. María Guadalupe Navarro Rojero  
Directora de Posgrado  
CIATEQ, A.C.

Por medio de la presente me dirijo a usted en calidad de Revisor del proyecto terminal del alumno Ing. Hugo Enrique García Rico, cuyo título es:

**"Implementación de Protocolo de Comunicación para Pruebas Funcionales entre Sistema Embebido de Bajo Costo Y Secadora Convencional de Ropa"**

Después de haberlo leído, corregido e intercambiado información con el alumno, y realizado los cambios que le fueron sugeridos, puede ser autorizada su impresión, a fin de que se inicien los trámites correspondientes para su defensa.

Sin otro particular por el momento, y en espera de que mis sugerencias sean tomadas en cuenta en beneficio del estudiante y la Institución, agradezco la atención prestada.

Atentamente,

A handwritten signature in black ink, appearing to be "VMS", written over a horizontal line.

Mtro. Victor Manuel Saldaña

## DEDICATORIA

A mi esposa por su apoyo incondicional, paciencia, comprensión, por sus palabras de aliento y por caminar a mi lado en este proyecto desde su inicio hasta este momento.

A mis padres por su apoyo incondicional, por formar en mí las bases para una vida de éxito y por haberme dado las herramientas que me permitieron llegar a este nivel académico.

A mis profesores el posgrado en Sistemas Inteligentes Multimedia de CIATEQ por compartir sus conocimientos obtenidos a lo largo de su amplia experiencia, muy en especial al Dr. Marco Antonio Garduño Ramón por toda la asesoría brindada para el logro de este proyecto, a la compañía Samsung Electronics por el apoyo brindado y por la confianza para poder realizar las ideas de este proyecto.

Ing. Hugo Enrique García Rico

Querétaro, Querétaro

Julio 2021

## RESUMEN

Esta investigación ha sido desarrollada con el propósito de resolver dos puntos para la empresa Samsung Electronics, uno es con respecto a las operaciones realizadas en el área de pruebas funcionales, donde la realización de las pruebas depende de una serie de pasos hechos por el operador de forma manual y el segundo punto se refiere al juicio de los resultados de la prueba, al ser pruebas manuales el operador es quien determina el resultado de la prueba, dejando una brecha para caer en error humano, más aparte la nula trazabilidad de los defectos.

En base a la información obtenida del documento de especificación de requerimientos que siguió las normas del estándar IEEE 830, se desarrolló un protocolo de comunicación instalado en un sistema embebido de bajo costo para poder mantener libre comunicación con la secadora y poder obtener los datos de los sensores, así como controlar y activar los modos de prueba de manera automática.

Con el análisis de los resultados adquiridos de las pruebas realizadas al sistema, concluye que la integración del protocolo de comunicación como un sistema de pruebas automáticas genero beneficios en la reducción de tiempo, así como la optimización de las operaciones del área de pruebas, fortaleciendo la detección y aplicación de las pruebas.

**Palabras clave:** Pruebas funcionales, Protocolo de comunicación, Sistema embebido.

## ABSTRACT

This research has been developed with purpose of solving two problems for the Samsung Electronics Company, one is with respect to the operations carried out in the functional testing area, where the performance of the tests depends on a series of steps made by the operator manually, and the second point refers to the judgment of the test results, as manual tests are the operator who determines the test result, leaving a gap to fall into human error, apart from the null traceability of defects.

Based on the information obtained from the requirements specification document that followed the standards of the IEEE 830 standard, a communication protocol installed in a low-cost embedded system was developed to be able to maintain free communication with the dryer and obtain the data from the sensors, as well as control and activate test modes automatically.

With the results obtained from the tests carried out on the system, it is concluded that the integration of the communication protocol as an automatic test system generated benefits in the reduction of time as well as the optimization of the operations of the test area, strengthening the detection and application of tests.

**Keywords:** Functional tests, Communication protocol, Embedded system.

## ÍNDICE DE CONTENIDO

RESUMEN .....	iv
ABSTRACT .....	v
ÍNDICE DE CONTENIDO .....	vi
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE TABLAS .....	xi
GLOSARIO .....	xii
1. INTRODUCCIÓN .....	1
1.1. ANTECEDENTES .....	1
1.2. DEFINICIÓN DEL PROBLEMA .....	2
1.3. JUSTIFICACIÓN .....	4
1.4. OBJETIVOS.....	5
1.4.1. Objetivo general.....	5
1.4.2. Objetivos específicos .....	5
1.5. HIPÓTESIS.....	6
2. MARCO TEÓRICO .....	7
2.1. PROTOCOLO DE COMUNICACION .....	8
2.2. COMUNICACIÓN SERIAL .....	9
2.2.1. Formato comunicación serial.....	12
2.2.2. Modelo de transmisión y recepción .....	14
2.2.3. Handshaking .....	15
2.3. Zigbee & bluetooth.....	16
3. PROCEDIMIENTO.....	18
3.1. INTRODUCCIÓN.....	18
3.1.1. Propósito.....	18
3.1.2. Enfoque .....	18
3.2. DESCRIPCION GENERAL.....	19
3.2.1. Sistema embebido .....	19
3.2.2. Secadora de ropa.....	20
3.2.3. Usuario final .....	20
3.2.4. Restricciones del sistema .....	20
3.3 REQUISITOS DEL SISTEMA .....	21

3.3.1. Interfaz gráfica.....	21
3.3.2. Sistema embebido .....	22
3.3.3. PC .....	22
3.4. DISEÑO Y ARQUITECTURA DEL SISTEMA.....	22
3.4.1. Master PC .....	23
3.4.2. Dispositivo receptor .....	24
3.4.3. Protocolo de comunicación .....	24
3.5. DISEÑO DE MENSAJE .....	27
3.5.1. Encendido.....	27
3.5.2. Versión de software.....	28
3.5.3. Prueba de encendido .....	29
3.5.4. Eeprom .....	31
3.5.5. Lectura EEPROM .....	31
3.5.6. Modo de prueba.....	32
3.5.7. Leer temperatura .....	33
3.5.8. Power off .....	34
3.6. ARQUITECTURA DEL SISTEMA .....	35
3.6.1. Hardware del sistema .....	36
3.6.2. Hardware del sistema embebido .....	37
3.7. INTERFAZ GRÁFICA .....	41
3.8. PRUEBAS .....	41
3.8.1. Simulación de señales.....	41
3.8.2. Prueba 1 .....	43
3.8.3. Prueba 2 .....	44
3.8.4. Protocolo final.....	45
3.8.5. Escenario de pruebas .....	46
4. RESULTADOS .....	50
4.1. CONECTIVIDAD .....	50
4.2. PRUEBA PILOTO.....	51
4.3. REDUCCIÓN DE TIEMPO CICLO .....	53
CONCLUSIONES .....	56
RECOMENDACIONES .....	59
APORTACIÓN DE LA TESIS.....	60

APORTACIÓN SOCIAL DE LA TESIS .....	61
REFERENCIAS .....	62
ANEXO .....	64

## ÍNDICE DE FIGURAS

Figura 1. Diagrama de comunicación humana. ....	9
Figura 2. Diagrama Comunicación Serial .....	10
Figura 3. Diagrama Comunicación Paralelo .....	11
Figura 4. Modos de la comunicación serial. ....	12
Figura 5. Formato de paquetes .....	13
Figura 6. Interacción maestro-esclavo. ....	14
Figura 7. Diagrama general del sistema propuesto. ....	23
Figura 8. Header enviado por UART. ....	26
Figura 9. Secadora en modo test. ....	30
Figura 10. Ejemplo de la concatenación de 2 bytes. ....	34
Figura 11. Arquitectura del sistema completo. ....	36
Figura 12. Diagrama de hardware del sistema. ....	37
Figura 13. Esquemático de Receptor .....	39
Figura 14. Ensamble SE transmisor. ....	40
Figura 15. SE Trasmisión 4 estaciones .....	40
Figura 16. Elementos de GUI. ....	41
Figura 17. Puertos COM utilizados. ....	42
Figura 18. Software para simulación In/out. ....	43
Figura 19. GUI inicial .....	44
Figura 20. Prueba performance .....	45
Figura 21. Resultado final de la GUI .....	46
Figura 22. Escenario de pruebas funcionales. ....	47
Figura 23. Facetas de la prueba .....	47
Figura 24. SE receptor conectado a secadora .....	48
Figura 25. Vista previa de Sistema Operando .....	49
Figura 26. Prueba de rango. ....	50
Figura 27. Grafica Niveles RSSI para Estación de Prueba .....	51
Figura 28. Estación de Pruebas Línea de Ensamble Secadoras. ....	52
Figura 29. Caracterización de prueba manual. ....	55

Figura 30. Variación entre operaciones manuales .....	56
Figura 31. Operaciones Automáticas.....	57

## ÍNDICE DE TABLAS

Tabla 1. Estructura de paquetes de Rx & Tx.....	15
Tabla 2. Estructura de protocolo de comunicación propuesto .....	25
Tabla 3. Mensaje Encendido .....	27
Tabla 4. Respuesta ACK Encendido .....	27
Tabla 5. Mensaje software versión. ....	28
Tabla 6. ACK mensaje software versión. ....	28
Tabla 7. Versión software es .....	29
Tabla 8. Versión SW ACK Ejemplo. ....	29
Tabla 9. Mensaje power test.....	30
Tabla 10. ACK power test. ....	30
Tabla 11. Mensaje EEPROM write. ....	31
Tabla 12. ACK de mensaje EEPROM write.....	31
Tabla 13. Mensaje EEPROM read. ....	32
Tabla 14. ACK de mensaje eeprom read. ....	32
Tabla 15. Mensaje Test mode. ....	33
Tabla 16. ACK de mensaje test mode.....	33
Tabla 17. Mensaje temperatura.....	33
Tabla 18. Mensaje ACK temperatura. ....	34
Tabla 19. Mensaje power off. ....	35
Tabla 20. ACK de mensaje power off.....	35
Tabla 21. Resultado de prueba piloto. ....	53
Tabla 22. Comparación de tiempo manual vs automático. ....	54
Tabla 23. Resumen de resultados .....	57

## GLOSARIO

**ACK:** (del inglés acknowledgement, en español recibo), es un mensaje que se transmite entre dispositivos electrónicos, enviado por el destinatario de la comunicación hacia el remitente para confirmar el recibo de un mensaje.

**Arduino:** Es una plataforma de hardware y software libre que lo vuelve fácil de usar, permitiendo la creación de prototipos en tiempos muy cortos, teniendo como ventaja su enorme comunidad de soporte, siendo una herramienta necesaria para el desarrollo de aplicaciones estudiantiles como industriales de bajo costo.

**Automatización:** En la industria se le llama al conjunto de procesos que trabajan entre sí, con la mínima o nula intervención de un operador. Normalmente son implementados para mejorar la productividad, la calidad de los productos o la eficiencia de la inspección de calidad.

**Bits:** En computación se usa la palabra bit (en inglés Binary digit, en español dígito binario) al valor de la numeración binaria. Este sistema solo se compone de dos valores 1 y 0, comúnmente son usados para representar condiciones binarias como encendido y apagado o verdadero y falso etc.

**Byte:** Es la unidad básica normalmente utilizada en la rama de la informática y la electrónica digital, formado por un conjunto de bits (sistema binario) que son ordenados en conjuntos de 8 dígitos binarios, por lo tanto, cada byte equivale a 8 bits ordenados.

**Checksum:** Se trata de una suma utilizada en forma de verificación (suma de verificación), normalmente usada en la rama de las telecomunicaciones, su función principal es ayudar a la detección de cambios en un paquete de datos, haciendo una comprobación al inicio del envío y comparándola con una segunda comprobación al final de la transmisión, de tal manera que el paquete enviado contenga el valor de suma así el receptor puede calcular y comprobar la integridad del paquete recibido, se pueden rechazar los paquetes cuando hay discrepancia y solicitar una retransmisión.

**COM:** Es simplemente una interfaz de E/S que permite conectar un dispositivo serie a un ordenador. Es posible que también escuche que los puertos COM se llaman

puertos serie. La mayoría de los ordenadores modernos no tienen puertos COM, pero hay muchos dispositivos de puerto serie que aún utilizan la interfaz.

**Eeprom:** (En inglés Electrically Erasable Programmable Read Only Memory) Memoria que puede ser de sólo lectura y puede borrar su contenido eléctricamente. Este tipo de memorias pueden almacenar su contenido incluso en su estado desenergizado, puede ser reutilizable al borrar su contenido con una descarga de más de 5v.

**GUI:** Una graphical user interface en inglés, es considerada un conjunto de gráficos que pueden ser utilizados para controlar o interactuar con un ordenador. Los elementos gráficos que utiliza normalmente son dibujos representativos a alguna acción o tarea, esto hace que las aplicaciones sean más amigables y fáciles de usar para ser humano.

**Hardware:** Es la parte que compone el interior de un ordenador, conformado principalmente de elementos electrónicos unidos y trabajando entre sí. Estos elementos electrónicos realizan funciones específicas que garantizan el procesamiento de la información.

**Header:** Cada nuevo paquete comienza con un byte de "encabezado". Este byte se usa para determinar el comienzo del paquete e informa que los bytes que contienen datos lo siguen. El número especificado de bytes de datos va seguido del byte de "suma de comprobación". La parte de envío cuenta la "suma de verificación" de los datos y envía el paquete. La parte receptora lee el paquete y considera su suma de control. Si la suma de comprobación es igual a la especificada por el lado emisor, el paquete llegó sin daños y se puede utilizar.

**IEEE 830:** Es un conjunto de lineamientos que ayudan a la organización y claridad de las especificaciones de los requerimientos o requisitos de software, generando la documentación que contiene los acuerdos, características entre el cliente y el desarrollador para cumplir con cada uno de estos.

**Manufactura:** Es el proceso que da como resultado de convertir materias primas en algo más sofisticado, elaborado por medio de uno o varios procesos industriales que da como resultado un producto listo para su venta al usuario final.

**Master PC:** Es la computadora principal que controla uno o varios procesos, dentro del ámbito de manufactura en serie. Esta computadora normalmente se encuentra conectada con sensores y señales de entrada/salida.

**Microcontrolador:** es un componente electrónico utilizado para tareas de propósito específico, está conformado por una alta integración de elementos necesarios en un ordenador, Son comúnmente usados para controlar procesos o dispositivos electrónicos de propósito específico.

**Operador:** Es la persona que participa directamente en el proceso de producción, manejo de maquinaria y herramientas específicas necesarias para lograr la transformación de producto, interpreta planos, gráficos de diseño y entiende los planes de producción.

**PBA:** Es el proceso de conectar los componentes electrónicos con los cables de las placas de circuito impreso. Las trazas o vías conductoras grabadas en las láminas de cobre laminado de PCB se utilizan dentro de un sustrato no conductor para formar el conjunto. Unir los componentes electrónicos con las placas de circuito impreso es la acción final antes de utilizar el dispositivo electrónico en pleno funcionamiento.

**Protocolo de comunicación:** Conjunto de normas o lineamientos seguidos por los dispositivos electrónicos o software que intervienen en una comunicación, ya sea con ordenadores transmitiendo datos, sin el protocolo resulta difícil entender a ambas partes el mensaje a comunicar.

**Pruebas funcionales:** Son las acciones que se llevan a cabo para comprobar la funcionalidad de cada componente y garantizar la funcionalidad del producto final, suelen utilizarse tanto para productos de software como para productos con ensamble de varios componentes.

**RS232:** (en inglés Recommended Standard 232, en español: "Estándar Recomendado 232"), es una norma que se usa para el intercambio de datos binarios entre componentes electrónicos que pueden ser un ordenador y un dispositivo electrónico como un modem utilizando una interfaz.

**RSSI:** Es una medida que estima la intensidad de una señal, se usa para medir la comunicación que hay entre dos componentes electrónicos para determinar que

tan bien es la transmisión de datos entre cada uno por medio de la intensidad de la señal.

**SE receptor:** Dispositivo electrónico formado por tarjeta electrónica arduino, shield Zigbee y Xbee. Estos componentes juntos son capaces de recibir paquetes de datos de manera inalámbrica.

**SE trasmisor:** Dispositivo electrónico formado por tarjeta electrónica arduino, shield rs232, shield Zigbee y Xbee. Estos componentes facilitan la transmisión de datos desde la PC al sistema receptor de manera inalámbrica.

**Sensores:** Es un dispositivo capaz de convertir un fenómeno físico en una unidad de medición ya sea digital o analógica, Esta medición puede utilizarse para detectar cambios alrededor del dispositivo y poder mostrarlos en un sistema más complejo por ejemplo una pantalla, esto lo hace más entendible para los humanos.

**Shield arduino:** es una placa de circuito impreso, hecha a la medida de un arduino UNO, capaz de empotrarse únicamente mediante sus pines y conectar diferentes componentes de hardware de propósito específico, Esta funcionalidad de las placas shield incrementa las capacidades del arduino uno como son la conexión a un ordenador, extensión de puertos, etc.

**Sistema embebido:** Es un sistema electrónico diseñado para propósitos específicos como resolver tareas o controlar procesos, son considerados minicomputadoras ya que son capaces de procesar datos, pero a un nivel más bajo cumpliendo con el único fin para el cual fueron diseñados.

**Smart factory:** Término utilizado por empresas manufactureras que utilizan procesos automatizados o inteligentes, para mejorar ya sea el proceso de ensamble, el tiempo ciclo o incluso la calidad del producto, Se caracteriza por el análisis de datos y el uso de internet para interactuar con las máquinas y procesar la recolección de datos.

**SRS:** (en español especificación de requisitos de software) es una representación completa del propósito y el entorno previstos para el software en desarrollo. El SRS describe completamente lo que hará el software y cómo se espera que funcione.

**SW:** Son líneas de código que se transforman en código binario, entendible por los ordenadores y le dicen qué hacer. El software abarca el conjunto completo de programas, procedimientos y rutinas asociados con el funcionamiento de un

sistema informático. El término se acuñó para diferenciar estas instrucciones del hardware, es decir, los dispositivos físicos de un sistema informático.

**UART:** (en español Transmisor receptor asíncrono universal) Es un circuito electrónico que forma el puerto serie. Convierte bytes paralelos de la CPU en bits seriales para la transmisión y viceversa. Genera y quita los bits de inicio y parada agregados a cada carácter.

**XOR:** (OR exclusivo) Es una operación lógica booleana que se usa ampliamente en criptografía, así como en la generación de bits de paridad para verificación de errores y tolerancia a fallas. XOR compara dos bits de entrada y genera un bit de salida. La lógica es simple. Si los bits son iguales, el resultado es 0. Si los bits son diferentes, el resultado es 1.

**Zigbee:** Es considerado un protocolo de comunicación inalámbrica digital de bajo consumo, es utilizado ampliamente para comunicar y enviar paquetes de datos entre dispositivos electrónicos de manera inalámbrica, puede ser configurado para enviar paquetes de datos en formato UART través de radiodifusión.

**Xbee:** Son dispositivos electrónicos de tamaño pequeño que son capaces de comunicarse de forma inalámbrica con otros Xbee, se pueden configurar para reemplazar cables de comunicación UART, lo que lo hace una excelente herramienta para implementar comunicación de paquetes entre dispositivos electrónicos.

# 1. INTRODUCCIÓN

En este trabajo se exponen las bases para la implementación de un protocolo de comunicación entre un sistema embebido y una secadora de ropa convencional, esto para facilitar el monitoreo, la puesta en marcha de pruebas al sistema completo y el análisis de datos cuantitativos que arrojan los sensores de la secadora al hacer las pruebas funcionales, esto se hace por medio de un sistema embebido de bajo costo conectado a unos de los puertos de la secadora. Se presentan los beneficios de utilizar un microcontrolador de bajo costo y las funciones de software para evitar fallos en la comunicación. Se analiza, además, el impacto que puede tener dentro de una planta de manufactura de electrodomésticos con las mejoras que podría presentar en la línea de pruebas funcionales.

## 1.1. ANTECEDENTES

En la actualidad los equipos de prueba de tarjetas electrónicas (PBA) para electrodomésticos son muy robustos y costosos, algunas veces se necesitan hacer pruebas rápidas de las principales funciones ya sea para análisis de fallas o para reparación de las mismas tarjetas electrónicas; las pruebas en la mayoría de los casos son manuales, para realizarlas se ingresa al modo "test" del aparato electrodoméstico de manera manual. El uso de datos de máquinas para mejorar el negocio de servicios ofrece nuevos potenciales de diferenciación a las empresas manufactureras (Jens, Inka, Ganz, & Spath, 2018).

Los equipos actuales de pruebas cuentan con una simulación de sensores, actuadores y demás hardware para tratar de formar un sistema funcional lo más cercano al producto final ya sea una secadora, lavadora, refrigerador, etc. No se puede asegurar la calidad del sistema completo sin requerir de pruebas funcionales reales del comportamiento de cada uno de los componentes y su interacción con la tarjeta principal, todo esto antes de enviar el producto al cliente final. En la industria de manufactura de aparatos electrodomésticos cada vez es más común encontrar fábricas inteligentes y máquinas tomando decisiones en base a datos obtenidos del resultado de pruebas. Es por esa razón que se vuelve

parte importante del sistema la conectividad entre el sistema principal de los electrodomésticos y el sistema de análisis de resultados de las pruebas. (Hameed, y otros, 2011) Comentan que en el panorama de la fabricación moderna, las empresas confían cada vez más en los sistemas de servicio de productos, es decir, la combinación de productos y servicios para obtener una ventaja competitiva que vea su impacto en los costos y tiempo de producción en serie.

En un ambiente de manufactura en serie es muy importante el tiempo de producción, se hacen esfuerzos para hacer más eficientes y rápidas las pruebas funcionales antes de enviar el producto al cliente final. Este trabajo busca formar parte de la automatización de pruebas funcionales sustituyendo las operaciones y evaluación de resultados realizados por un operador de producción, utilizando puertos de comunicación serial para la conexión entre el sistema de control y el dispositivo a probar. El puerto serie es una interfaz de comunicación digital de datos, donde la información se transmite de forma secuencial bit a bit por los conductores, de esta forma un puerto serie debe enviar toda la información en un bit detrás de otro. Los puertos serie más usados son USB, I2C, CAN, RS2323 y UART, este último tiende a usar menos cables y enviar la información de manera secuencial (Castillo, 2020). Este proyecto pretende aprovechar las ventajas del protocolo de comunicación UART (en inglés Universal Asynchronous Receiver-Transmitter), es el dispositivo encargado de controlar los puertos y dispositivos serie. Se encuentra integrado en la tarjeta electrónica principal de la mayoría de los sistemas electrónicos. El controlador del UART es el componente electrónico clave del subsistema de comunicaciones series entre sistemas electrónicos. El UART toma bytes de datos y transmite los bits individuales de forma secuencial. En el destino, un segundo UART ensambla los bits en bytes completos nuevamente.

## **1.2. DEFINICIÓN DEL PROBLEMA**

Este proyecto se enfoca en la dificultad presente al momento de analizar las fallas del producto ensamblado en serie por la empresa Samsung Electronics Digital Appliances, enfocándonos en el producto de secadora de ropa convencional.

Estos productos son ensamblados para ser exportados a mercado americano y canadiense.

Las fallas presentadas en la tarjeta electrónica son detectadas en el último proceso de ensamble y tan solo antes de salir de la fábrica para su venta. En la actualidad no se cuenta con un dispositivo para probar las tarjetas electrónicas hasta antes de ser ensambladas por la línea de ensamble final. La única manera de probarlas es conectando todo el sistema de la secadora, esto implica tiempo de conexión, espacio para los diferentes tipos de voltaje manejados y tiempo elevado en la iniciación de las pruebas, haciéndose mediante activación de modo de prueba de la secadora, implicando manipulación de operador.

La distribución de pruebas funcionales dentro de la línea de ensamble en serie de la empresa Samsung Electronics Digital Appliances, se hace separando las pruebas en 4 estaciones de operación las cuales son, consumo de energía, funcionamiento, inspección de seguridad eléctrica, registro y comprobación de datos en la EEPROM, utilizando un total de 3 operadores, ellos realizan la activación de cada prueba y evalúan el resultado de esta. En este proyecto se expone los beneficios de utilizar un sistema embebido de bajo costo para hacer una herramienta de autodiagnóstico y monitoreo de aparatos electrodomésticos en este caso específico de una secadora de ropa convencional, utilizando el puerto serial UART para la conexión y creando un protocolo de comunicación capaz de identificar los llamados o las peticiones de su contraparte, pueda analizar fallos en la transmisión del mensaje y sea fácil de entender para cada una de las partes entre el electrodoméstico y el sistema embebido.

Buscando resolver las dificultades de la aplicación en un ambiente industrial se toman los siguientes puntos como partida para analizar su funcionalidad y proponer soluciones:

**Compatibilidad de hardware:** Se requiere analizar las tarjetas electrónicas de los aparatos electrodomésticos e identificar el hardware necesario para la aplicación,

si es funcional o se requiere agregar circuitos, normalmente las tarjetas electrónicas están diseñadas para optimizar recursos por lo tanto debemos analizar el tipo de arquitectura de trabajo.

**Sistema embebido:** Se requiere un sistema basado en la tarjeta electrónica de preferencia ya diseñado el hardware, debe contar con puertos UART, debe contar con el hardware necesario para para almacenar, mostrar datos y donde podamos cargamos SW con posibles upgrade al mismo, facilitando diseño del dispositivo.

**Software:** se requiere modificar el SW para la aplicación, esto en la tarjeta electrónica de la secadora y además no interfiera con las funciones principales que controlan el sistema principal del producto, Así mismo SW de sistema embebido debe ser capaz de hacer las peticiones de pruebas o recibir los resultados obtenidos de las pruebas.

**Protocolo de comunicación:** deberá ser creado en base a protocolo de transmisión con respuesta. La tarjeta a probar debe identificar el protocolo y responder con lo solicitado por el dispositivo de prueba. Una vez completada la comunicación el dispositivo deberá analizar la respuesta de la tarjeta y los datos obtenidos.

### 1.3. JUSTIFICACIÓN

Este proyecto se enfoca en las necesidades de la producción en serie de secadora de ropa convencional, ensamblada en la fábrica de manufactura de Samsung Electronics Digital Appliances, principalmente a los técnicos y operadores de producción encargados de ejecutar las pruebas funcionales y el análisis de fallas en el sistema de la secadora.

Las ventajas del sistema:

- Reducción en el tiempo de prueba de más de 20%. En la actualidad las pruebas son realizadas de forma manual y no se puede iniciar una prueba sin terminar la otra.

- Reducción de mano de obra 30%. Al sustituir personal operativo dedicado a realizar las pruebas.
- Reducción de error humano al sustituir la evaluación de la prueba de un operador a un sistema embebido, el cual analizará los datos de la prueba y emitirá el resultado.
- Análisis de datos en tiempo real, datos de los sensores pueden ser leídos en el momento exacto de la prueba y analizar el funcionamiento.
- Herramienta de bajo costo.
- Movilidad para poder usar en cualquier lugar.

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo general**

Resolver el problema de ejecución de pruebas en un ambiente de producción en serie, así mismo mejorar la calidad, tiempo de ejecución, robustez de las pruebas y la obtención de datos para análisis futuros, mediante un protocolo de comunicación serie encargado de conectar un sistema embebido con la tarjeta electrónica de una secadora y ambos sistemas puedan entender las peticiones y respuestas entre sí.

### **1.4.2. Objetivos específicos**

A continuación, se enlistan los objetivos específicos más importantes:

- a. Comunicación efectiva entre sistema embebido y tarjeta electrónica de una secadora.
- b. Protocolo de comunicación capaz de entender ambos sistemas.
- c. Software preparado para enviar y recibir paquetes de datos sin pérdida de información.
- d. Modificación de SW para agregar funciones del protocolo de comunicación en tarjeta electrónica de secadora.
- e. Interface fácil de entender para cualquier persona.

## **1.5. HIPÓTESIS**

Es posible mediante el uso de un protocolo de comunicación específicamente diseñado para esta aplicación lograr realizar pruebas automáticas con mínima manipulación de operadores de forma eficiente, robusta e impactando considerablemente en el tiempo de producción en serie de secadoras convencionales.

## 2. MARCO TEÓRICO

Una Smart Factory alberga procesos de fabricación inteligentes, esta es futurista porque puede producir y ofrecer una productividad mucho más allá de nuestras expectativas. Si miramos cómo esto es posible, podemos ver las Smart Factory son una combinación de tecnologías las cuales proporcionan los métodos y técnicas óptimos en la fabricación. Además, podemos ser testigos de las fábricas inteligentes no son solo máquinas y robots inteligentes comunicándose a través de un producto de software avanzado. Sin embargo, es importante darse cuenta de las fábricas inteligentes, como las casas inteligentes, no son una visión futurista; están con nosotros hoy y lo han estado durante al menos una década.

Con el desarrollo de la tecnología informática, las actividades humanas han cambiado mucho. El uso del control por computadora también es una tendencia. El objetivo de utilizar el lenguaje informático para controlar dispositivos es parte de la realización de la automatización. La comunicación de datos en serie a través de la programación se ha vuelto cada vez más común convirtiéndose en una herramienta poderosa para las empresas quienes adoptan Smart Factory (Tao, 2013).

La fabricación de herramientas es una industria con un grado relativamente alto de valor agregado e innovación, tiene un papel importante en el desarrollo y fabricación de diversos productos. Es una industria intensiva en conocimiento que requiere un gasto de material relativamente pequeño. Debido a esto, la fabricación de herramientas también tiene una influencia significativa en elevar el nivel de desarrollo tecnológico de otras actividades económicas vinculadas a la producción de diversos productos. Es por esa razón por la cual esta investigación se enfoca en las herramientas, las cuales se pueden utilizar para hacer más eficiente el proceso de pruebas funcionales inteligentes las cuales puedan ser ejecutadas de manera confiable y evitando los errores humanos.

Un ejemplo es la producción en Amberg, una filial de Siemens Digital que está altamente automatizada, las personas finalmente toman las decisiones. Por ejemplo, un técnico en electrónica supervisa la estación de prueba en busca de placas de circuito impreso defectuosas, aunque él mismo no prueba los componentes ni los circuitos. El técnico usa una computadora para monitorear toda la cadena de producción desde su lugar de trabajo ya que cada placa de circuito tiene su propio código de barras único que le permite comunicarse con las máquinas de producción. Más de 1000 escáneres documentan todos los pasos de fabricación en tiempo real y registran los detalles del producto, como la temperatura de soldadura, la ubicación, datos y resultados de las pruebas. Esta gran cantidad de datos específicos de productos está disponible para cada persona y máquina del proceso de producción en Amberg (Gilchrist, 2016).

## **2.1. PROTOCOLO DE COMUNICACION**

Durante la historia del ser humano la comunicación ha sido causante del éxito de la supervivencia humana, ha sido vital para el traspaso de la información de generación en generación, desde los inicios el hombre creó protocolos o reglas para la comunicación. Un protocolo de comunicación está formado por un conjunto de reglas y formatos de mensajes establecidas a priori con la cual la comunicación entre el emisor y un receptor sea posible (Tolosa, 2002).

Las reglas definen la forma como deben de efectuarse las comunicaciones, incluyendo la temporización, la secuencia, la revisión y la corrección de errores.

Hay 3 elementos clave por considerar:

- Sintaxis (formato de los mensajes: datos + comandos).
- Semántica (significado de los comandos).
- Secuencia y temporalización (adecuado de las acciones que se toman respecto de los comandos).

La especificación de un protocolo consiste de varias partes:

- Un formato preciso para los mensajes válidos.
- Un conjunto de reglas de procedimiento para el intercambio de los datos.

- Un conjunto de mensajes válidos que se pueden intercambiar, junto con su significado.
- El servicio que provee el protocolo.

En la [figura 1](#) se muestra un ejemplo de protocolo de comunicación usado por nosotros los humanos para comunicarnos con otras personas.

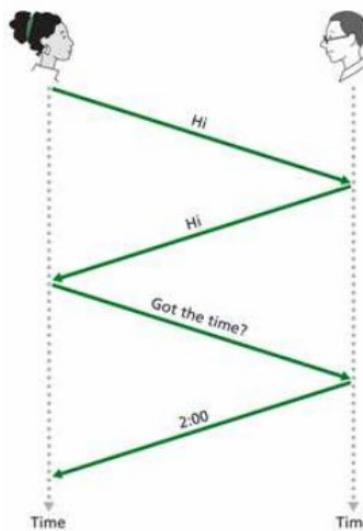


Figura 1. Diagrama de comunicación humana.  
(Tolosa, 2002)

## 2.2. COMUNICACIÓN SERIAL

La transferencia de datos entre dos dispositivos puede realizarse en comunicación paralelo o en serie. En la comunicación en paralelo, se transfieren varios bits en cada evento de reloj, mientras que en la comunicación en serie, se transfiere un solo bit en cada evento de reloj. En la comunicación en paralelo, el número de líneas determina el número de bits que se pueden transferir simultáneamente. Por lo tanto, para una comunicación en paralelo de 8 bits, habrá 8 líneas de datos y pocas líneas más para señales de protocolo de enlace. Pero en la comunicación en serie, los datos se transfieren bit a bit en serie en la misma línea, por lo que el

número de líneas para datos la comunicación será menor en comparación con la comunicación paralela.

La velocidad de operación en la comunicación en serie será menor en comparación con la operación en paralelo. Para la comunicación entre dos placas, se prefiere la comunicación en serie ya que requiere menos líneas de datos para conectarse de una placa a otra. Hay varios protocolos de comunicación en serie como UART (Receptor y transmisor asíncrono universal), SPI (Interfaz periférica en serie) o I2C (Circuito interintegrado) (Gadre & Gupta, 2018).

La comunicación en serie y la comunicación en paralelo son las dos formas con las cuales los componentes de un dispositivo electrónico intercambian datos o hace peticiones el uno al otro. Como sugiere el nombre, la comunicación en serie se usa para transferir un bit a la vez a través de un medio dado, ver [Figura 2](#), mientras que en la comunicación en paralelo, un bloque de datos se transfiere al mismo tiempo y cada uno de los bits requiere un canal separado y adicionalmente una línea de referencia, típicamente tierra.

Debido a que la comunicación en paralelo transfiere una gran cantidad de datos a la vez, este método requiere varias líneas separadas para facilitar la comunicación. Como puede imaginar, esto resultaría en el requisito de más espacio en el tablero de conexiones, lo que a menudo no es lo ideal. En la [Figura 3](#) se muestra un protocolo de comunicación paralelo.



Figura 2. Diagrama Comunicación Serial (Gupta, 2019)

Esa es la razón por la que la comunicación en serie es un método de comunicación más común siempre que se trate de dispositivos integrados: a diferencia de la comunicación en paralelo, solo requiere una línea para facilitar el intercambio de datos (Gupta, 2019).

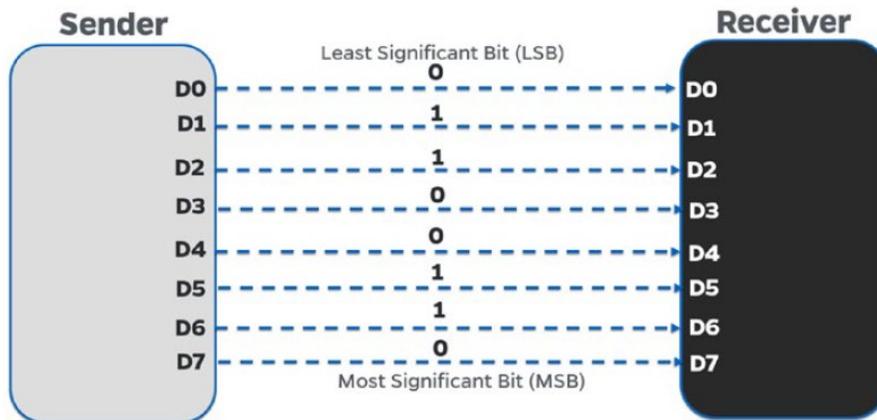


Figura 3. Diagrama Comunicación Paralelo (Gupta, 2019)

El protocolo serial se ha convertido en uno de los más usados gracias a sus canales, son la principal forma de comunicación utilizada en los sistemas electrónicos y digitales actualmente, pudiendo encontrarse diversas formas de formatos y protocolos de comunicación serial en aplicaciones como interconexiones cortas entre chips e intra-chip, hasta la comunicación de largo alcance con naves espaciales distantes.

La comunicación en serie se divide en función de la dirección del flujo de datos en la línea de comunicación.

1. Simplex: en este tipo de comunicación, la dirección del flujo de datos es solo en una dirección, es decir, los datos pueden transmitirse o recibirse. Por ejemplo, como la comunicación entre la impresora y la PC host, donde los datos se transmiten desde la PC host a la impresora, no al revés.

2. Half Dúplex: En este tipo de comunicación, los datos se pueden transmitir o recibir a través de la línea de comunicación. Pero en un instante de tiempo, los datos se transmitirán o recibirán al final de la comunicación. Por ejemplo, walkie-talkies donde los datos se pueden transmitir de un walkie a otro, pero en un momento solo se pueden transmitir y uno puede recibir.
3. Full Duplex: en este tipo de comunicación, los datos se pueden transmitir y recibir simultáneamente a través de la línea de comunicación. Por ejemplo, como teléfono.

La [figura 4](#) muestra los modos de comunicación mencionados anteriormente.

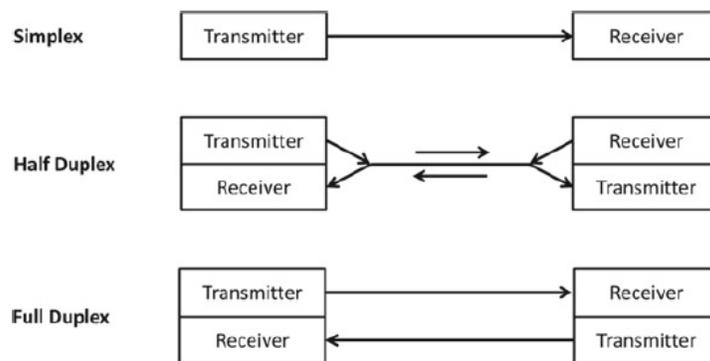


Figura 4. Modos de la comunicación serial.  
(Anónimo)

### 2.2.1. Formato comunicación serial

Los paquetes de datos asincrónicos tienen una estructura simple. Sus encabezados contienen solo un bit llamado bit de inicio, que es siempre un cero lógico o un símbolo de espacio. El cuerpo del paquete contiene un carácter de cinco a ocho bits ordenados desde su bit menos significativo hasta el más significativo. El pie de página contiene un bit de paridad opcional que permite la verificación de errores y uno o más bits de parada para indicar el final del paquete. Un bit de parada es siempre uno lógico. La [Figura 5](#) muestra el formato de un paquete asincrónico.

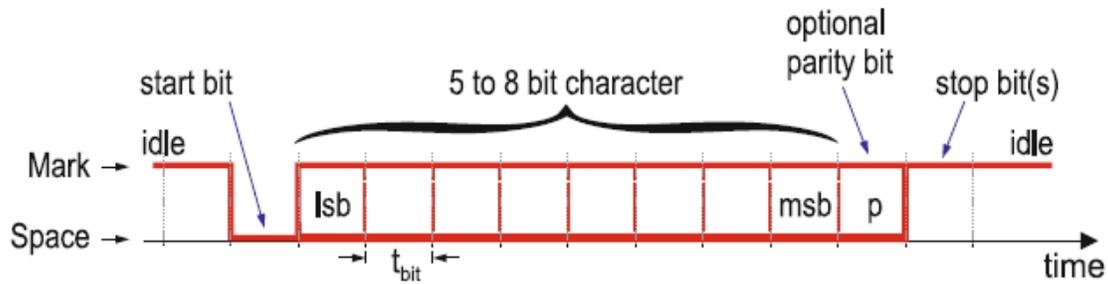


Figura 5. Formato de paquetes  
(Jiménez, Palomera, & Couvertier, 2014)

El estado inactivo de un canal asíncrono está en un nivel de lógica alto. Un bit de inicio indica una transición de canal de inactivo a activo. Un bit de inicio válido tiene una duración de un bit de tiempo completo. Cuando se detecta una transición de uno a cero en estado inactivo, se supone que es un bit de inicio. Para confirmar que se está recibiendo un bit de inicio, se vuelve a muestrear la línea, medio  $t_{bit}$  más tarde, a la mitad del tiempo de bit, y si todavía se encuentra bajo, se acepta un bit de inicio válido. De lo contrario, la condición se marcará como un error de trama.

Después de que se ha validado un bit de inicio, la línea se muestrea a intervalos separados por un tiempo de bit para detectar todos los bits restantes en el paquete. En el lado del transmisor, la longitud de cada bit está determinada por la velocidad de transmisión, obtenida del reloj en el lado del transmisor. En el extremo del receptor, otro reloj establece la velocidad de recepción. Aunque los relojes de transmisión y recepción son independientes, ambas partes deben acordar la misma tasa de bits con la cual la comunicación sea posible. Las interfaces en cada extremo del canal utilizan divisores de frecuencia para obtener la tasa de bits correcta de sus correspondientes relojes locales.

Debido a la deriva que generalmente se desarrolla entre relojes independientes, la longitud de los paquetes asíncronos se mantiene corta y el bit de inicio los vuelve a sincronizar. Además, la frecuencia de muestreo en el extremo del receptor

siempre es mayor a la frecuencia de bits del transmisor en un factor  $k$ , con valores típicos de 16, 32 o 64. (Jiménez, Palomera, & Couvertier, 2014).

### 2.2.2. Modelo de transmisión y recepción

Es probable que se produzca una pérdida de datos debido a factores ambientales externos cuando los datos se intercambian en un entorno de comunicación inalámbrica. Por tanto, debe evitarse una pérdida de datos durante la transmisión de datos colocando la información de las marcas de inicio y fin de los datos en el primer y el final de los datos.

Si se recibe un paquete de inicio, pero no se recibe el final del paquete, se considera como transmisión de datos aún incompleta. Luego, se vacía un búfer y el paquete de respuesta se envía al maestro para aumentar la confiabilidad de la transmisión de datos. El maestro no recibe la respuesta, en la [figura 6](#), puede notar un problema durante la transmisión. El maestro sin recibir la respuesta ejecutará la retransmisión. El maestro produce un mensaje de error si no se recibe respuesta a pesar de la retransmisión.

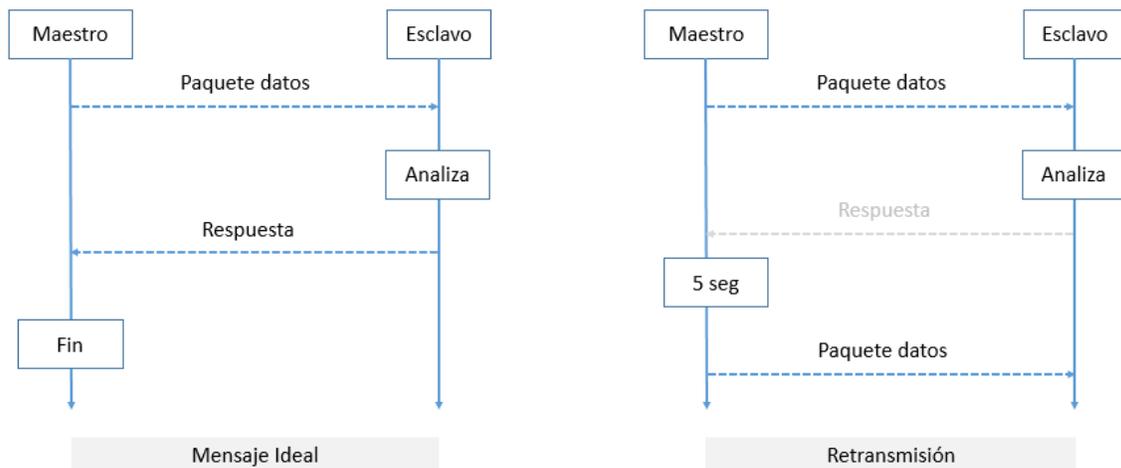


Figura 6. Interacción maestro-esclavo.  
Elaboración propia

En la tabla 1, se muestra un ejemplo, de las identificaciones del remitente y el receptor de Arduino se envían para distinguir el maestro y los esclavos, así como una cadena de comandos se envía junto para distinguir el Arduino se pueden montar múltiples sensores en un solo dispositivo. El contenido de datos más importante que se enviará se divide en los siguientes tipos de datos: int, booleano, float, char y string para permitir la extracción de datos. (Lee, Kim, Lee, Hong, & Kim, 2018).

Tabla 1. Estructura de paquetes de Rx & Tx.

Recepción	Paquete inicial <STX>	Cadena para enviar	ID arduino remitente	ID arduino destinatario	Metodo sobrecarga	Paquete final
Transmisión	Paquete inicial <STX>	Cadena para enviar	ID arduino remitente	ID arduino destinatario	solicitar transmisión de datos	Paquete final

(Lee, Kim, Lee, Hong, & Kim, 2018)

### 2.2.3. Handshaking

Handshaking por Software es una técnica de paridad entre dispositivos (también llamado XON / XOFF) utiliza caracteres de control en el flujo de bytes para señalar la detención y reanudación de la transmisión de datos. Control-S (ASCII 19) le indica al otro dispositivo que deje de enviar datos. Control-Q (ASCII 17) le indica al otro dispositivo que reanude el envío de datos. La desventaja de este enfoque es que el tiempo de respuesta es más lento y dos caracteres en el conjunto de caracteres ASCII deben reservarse para el uso del protocolo de enlace (Qian, den Haring, & Cao, 2009).

Para poder usar la técnica handshaking es necesario la existencia de un acuerdo en cuanto:

- Velocidad de datos.
- Cantidad de información soportada.
- Solución de los problemas por bloqueo de algunas de las partes.

Este protocolo sigue las pautas de enviar información al receptor hasta que el buffer está lleno, para saber si el buffer se encuentra lleno se usan los caracteres mencionados anteriormente ASCII 17, 19. Los cables para la comunicación siguen siendo solo 3 Tx, Rx, GND, ya que los caracteres de control se envían a través de los cables (Morales, 2013).

### **2.3. Zigbee & bluetooth**

El estándar inalámbrico IEEE 802.15.1, que utiliza Bluetooth, se dirigió al segmento de aplicaciones de comunicación para el consumidor. Admite velocidades de datos de medianas a altas para texto, audio, voz y video. Pero, IEEE 802.15.1 no es adecuado para las necesidades únicas de sensores y dispositivos de control.

ZigBee proporciona soportes de red de área personal inalámbrica de bajo alcance. El consumo de energía de un chip ZigBee es muy bajo. De hecho, a este nivel de consumo, una batería AAA puede durar un año.

Los dispositivos ZigBee pasan la mayor parte del tiempo durmiendo y por lo tanto, tienen ciclos de trabajo muy bajos. Utiliza una pila de protocolos simple en comparación con Bluetooth. Su implementación también requiere un código de software más pequeño en comparación con Bluetooth. Por lo tanto, ZigBee no debe tratarse como una tecnología competidora sino complementaria a Bluetooth (Ghosh, 2017).

ZigBee está más orientado a monitorear y controlar aplicaciones que a transferencias de datos reales. Admite la comunicación bidireccional entre varios dispositivos a través de redes simples, con muy poca energía. El bajo consumo de energía es el único punto de venta de ZigBee y lo posiciona idealmente para WPAN, así como para una cantidad de aplicaciones que involucran entornos inteligentes / omnipresentes. Algunas de las posibles aplicaciones de ZigBee podrían ser:

- Control de automatización monitoreado por sensores para aplicaciones industriales y comerciales.
- Sistema de alerta y control monitoreado por sensores para pacientes críticos en hospitales.
- Aplicación de vigilancia de seguridad, iluminación y control de ambiente para hogares inteligentes.
- Juguetes y juegos a control remoto.
- Red de PC y periféricos (mouse, teclado, joystick, etc.) sin cables.
- Varias aplicaciones de electrónica de consumo como control remoto de programas de TV, cine en casa, etc.

## **3. PROCEDIMIENTO**

### **3.1. INTRODUCCIÓN**

En la mayoría de los proyectos de software, hemos visto el crecimiento de la ingeniería de requerimientos de software. Los requerimientos de software nos ayudan a plantear las ideas principales del proyecto mediante la trazabilidad. Esto permitirá resolver de una manera más organizada el problema. Los documentos de requerimientos de software tienen como objetivo registrar las ideas y objetivos claves del sistema que definirán la manera de trabajar entre el sistema y el usuario.

El SRS es una representación completa de la interacción de un sistema que se desarrollará desde la perspectiva del proveedor. El estándar IEEE 830 se considera el estándar más famoso y universal que describe los enfoques recomendados para el SRS. Es un estándar muy flexible porque brinda la posibilidad al usuario de seleccionar el formato más apropiado dependiendo del tipo de proyecto (Chikh & Aldayel, 2014).

#### **3.1.1. Propósito**

El presente documento fue desarrollado con el propósito de describir y resaltar los requerimientos del desarrollo de un protocolo de comunicación entre un sistema embebido y una secadora convencional de ropa para facilitar las pruebas funcionales en la línea de producción de Samsung Electronics Querétaro y este documento se dirigió a los directores de producción e ingeniería quienes formularon y aprobaron los requerimientos basados en el estándar estándar IEEE 830. En anexo A se exponen las especificaciones de los requerimientos necesarios aplicados de acuerdo al estándar IEEE 830.

#### **3.1.2. Enfoque**

Este proyecto se enfoca en la dificultad que se presenta al hacer pruebas funcionales en la línea de ensamble de secadora convencional de ropa presentadas la empresa Samsung Electronics Querétaro. Actualmente las pruebas son realizadas por un operador entrenado y capacitado, sin embargo las pruebas

siguen siendo manuales y el juez quien determina la condición de la prueba es el operador.

El operador realiza una serie de pasos establecidos para poder ingresar al modo de prueba tomando un tiempo en cada prueba aproximado 48s por prueba. El tiempo de ejecución de la prueba depende mucho de la habilidad del operador, de que tan rápido puede accionar la prueba. Hay algunas limitantes o factores de riesgo en este método, los cuales son los siguientes:

- Jornada laboral de 12 horas.
- Error humano
- Perdida del resultado.
- Repetitividad en el proceso
- Tiempo de fabricación

Para resolver el problema en el proceso de pruebas, se pretende diseñar y desarrollar un protocolo de comunicación entre la secadora y un sistema embebido que cumpla con los siguientes requerimientos principales:

- Mejorar el tiempo de ejecución de las pruebas.
- Integrar pruebas más robustas.
- Pruebas automáticas.
- Historial de resultado de cada prueba.

## **3.2. DESCRIPCION GENERAL**

En este apartado se da una descripción general del sistema, explicando las funciones principales que deben ejecutarse, la información utilizada y las restricciones o modos de falla las cuales pueden afectar al desarrollo del proyecto.

### **3.2.1. Sistema embebido**

El sistema embebido que se utilizara para esta aplicación debe ser capaz de comunicarse con la secadora, ser de bajo costo (considerando el mantenimiento preventivo) y fácil de programar. Debe ser capaz de enviar paquetes de bytes

completos sin errores para que la secadora pueda identificar las órdenes o peticiones requeridas por el usuario final.

### **3.2.2. Secadora de ropa**

Debe de conocer la información de las órdenes y peticiones definidas en el protocolo de comunicación (Sistema embebido ↔ Secadora). Dichas ordenes o peticiones serán enviadas por comunicación serial compatible con el hardware de la tarjeta electrónica de la secadora. Creando las interrupciones en el programa principal para la ejecución y la obtención de datos de las pruebas en la secadora.

Funciones del producto:

- Comunicar la secadora con la GUI para mostrar información de la prueba.
- Generar una base de datos de las pruebas realizadas y sus resultados.
- Pruebas secuenciales automáticas.
- Análisis de resultados de las pruebas automáticas y veredicto final.

### **3.2.3. Usuario final**

El proyecto fue diseñado pensando en el operador de producción que utilizara la herramienta como parte de su proceso de inspección.

### **3.2.4. Restricciones del sistema**

- Sistema debe adaptarse al tiempo ciclo de cada operación, realizar las pruebas y funciones dentro del tiempo establecido para cada operación de 15.5 s.
- El microcontrolador de la secadora contiene 4 canales UART, para lograr la conexión con el sistema embebido solo se podrá utilizar el canal A.
- Cualquier cambio en el sistema requiere un proceso de aprobación bajo el estándar propio de Samsung Electronics, el cual consiste en elaborar un

documento relacionado con el cambio y obtener las aprobaciones de los gerentes de la planta Querétaro.

### **3.3 REQUISITOS DEL SISTEMA**

A continuación, se presentan algunos de los requerimientos funcionales que debe cumplir el sistema, en el Anexo A se encuentra el documento completo de la aplicación del estándar IEEE 830, dichos requisitos fueron seleccionados en conjunto con los departamentos de Calidad de producto, Producción y el Operador. El propósito del documento de requisitos es determinar y describir de una manera más clara las funciones que debe cumplir el sistema.

#### **3.3.1. Interfaz gráfica**

El sistema debe poder realizar las siguientes funciones Relacionadas con la visualización de la herramienta, debe ser capaz de visualizar de manera ordenada los siguientes componentes:

- Motor: Estado del motor, Resultado de prueba de funcionamiento.
- Calentador: Estado del calentador, Resultado de prueba de funcionamiento.
- Sensores 1: Resultado de temperatura de salida en la secadora.
- Sensores 2: Resultado de temperatura de entrada en la secadora.
- SW versión: De micro controlador asociado al modelo de secadora que se probara.
- Escritura de EEPROM: Debe ser capaz de mostrar el resultado de la escritura en memoria de la tarjeta de secadora.
- Ver datos de la memoria: Debe ser capaz de acceder los datos almacenados en la memoria EEPROM de la tarjeta de secadora.
- Historial de pruebas: Guardar y generar reporte de las pruebas realizadas.
- Tiempo: Mostrar el tiempo de la prueba.

- Datos de modelo: Debe mostrar los datos relacionados al modelo y número de serie.

### **3.3.2. Sistema embebido**

El sistema debe poder realizar las siguientes funciones:

- Tarjeta electrónica: Debe ser una tarjeta de propósito general, de bajo costo y con mínimo 2 puerto UART.
- Conexión Sistema Embebido→Secadora: Debe ser capaz de conectar con la secadora en el puerto de servicio originalmente de 5 pines.
- Conexión Sistema Embebido→PC: Debe ser capaz de conectar con la PC, con la interface principal y de control.

### **3.3.3. PC**

- Conexión PC→ Sistema Embebido: Debe ser capaz de conectar con 4 dispositivos al mismo tiempo, el máximo de secadoras a probar es de 4.
- PC: debe controlar las señales de inicio, monitoreo y termino de las pruebas.

## **3.4. DISEÑO Y ARQUITECTURA DEL SISTEMA**

El diseño de este sistema embebido comienza después del análisis de requerimientos y necesidades de la aplicación. Se plantea la idea de la arquitectura del sistema en el último nivel. En la [Figura 7](#) se muestra un panorama general de la arquitectura del sistema propuesto.

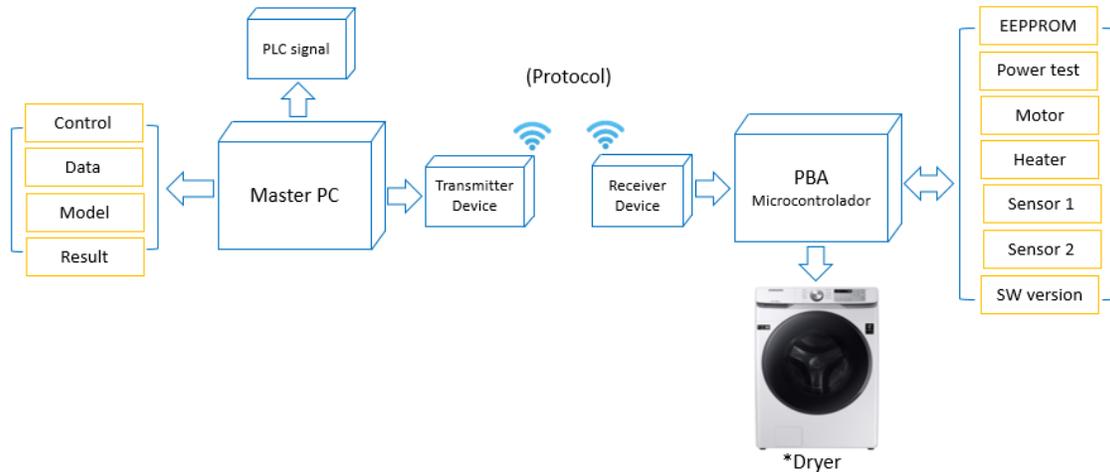


Figura 7. Diagrama general del sistema propuesto.

Elaboración propia

De acuerdo con diagrama propuesto, se observa la conexión entre el Master PC y la Secadora, será por medio inalámbrico, respetando las necesidades de la línea de producción ya que las secadoras a probar están en movimiento durante la etapa de pruebas, El sistema embebido se encarga de comunicarse con la secadora, recibe y envía datos al receptor, que está conectado con la Master PC.

### 3.4.1. Master PC

Es el bloque donde se presentan los datos recibidos de la secadora, se controla y monitorea la posición de las secadoras que se están probando. Provee la información clara y entendible para que el operador de producción pueda utilizar la herramienta sin problemas. Los puntos importantes de este bloque son:

- Control: Monitorea las señales de los sensores involucrados en el área de test de la línea de ensamble, asignando el inicio de la prueba dependiendo de la posición de la secadora.
- Data: Administra el envío y recibo de datos, para mostrarlos en la GUI (Interface Gráfica).
- Model: Recibe el dato del escáner con la información del modelo, número de serie de la secadora a probar y lo muestra en la GUI.

- Result: Muestra los resultados recibidos desde la PBA de la secadora, analiza y determina si los datos se encuentran dentro de la tolerancia permitida para mostrar el resultado final de la prueba.

### **3.4.2. Dispositivo receptor**

Juega un papel muy importante, es el que permanece conectado físicamente a la secadora, a su vez conecta la secadora con la master PC por medio de las órdenes enviadas por el "transmitter device", Este bloque es el que contiene el protocolo de comunicación, el cual fue creado para esta aplicación en específico, dicho protocolo es reconocido por las funciones agregadas al software principal de la secadora.

Las siguientes funciones son agregadas pensando en el cumplimiento de los requerimientos expuestos y aprobados por el personal de Samsung Electronics Querétaro, donde se agregan 7 pruebas activadas por el protocolo de comunicación creado para esta aplicación en específico. Las 7 pruebas son:

- EEPROM: Protocolo creado para escribir y leer en la memoria Eeprom.
- Power Test: Secuencia de activación de hardware de secadora para medir la potencia consumida por cada uno de los componentes activados.
- Motor: Protocolo creado para activar el motor.
- Calentador: Protocolo creado para activar el calentador.
- Sensor 1: Protocolo creado para obtener los valores de temperatura en el flujo de salida que van de 20°C a 40°C.
- Sensor 2: Protocolo creado para obtener los valores de temperatura en el flujo de entrada que van de 30°C a 50°C.
- Sw versión: Protocolo creado para obtener la versión de software cargada en la PBA principal de la secadora.

### **3.4.3. Protocolo de comunicación**

El protocolo de comunicación propuesto fue creado para facilitar el control entre el sistema embebido y la tarjeta principal de la Secadora. EL software de ambos tiene cargadas las instrucciones de este protocolo de comunicación en forma de

paquetes de bytes. El sistema embebido enviara una cadena de bytes en formato hexadecimal a la tarjeta controladora de una secadora para poder probar y controlar las principales funciones. En la tabla 2, se muestra la estructura del paquete de bytes que forma el protocolo de comunicación, definido tanto para TX como para RX.

Tabla 2. Estructura de protocolo de comunicación propuesto

	<b>Header 1</b>	<b>Header 2</b>	<b>Data Size</b>	<b>Identifier</b>	<b>Data</b>	<b>Check sum</b>
Hex	0x5A	0xA5	CMD	CMD	CMD	CKS
Byte	1	1	1	1	1~10	1

Elaboración propia

De acuerdo a la tabla anterior, se observa la estructura de cada palabra debe contener mínimo 5 bytes para que ambos dispositivos puedan entender la trama de bytes, si alguno byte falta o llega incompleto o tiene diferente valor al asignado en la descripción de requerimientos, será considerado un error de transmisión o recepción según sea el caso. Para facilitar la identificación de estos errores cada palabra o paquete cuando es recibido de manera correcta, el dispositivo ya sea la secadora o el sistema embebido debe enviar al origen la confirmación de la recepción mediante un ACK (acknowledgement o acuse de recibo).

El ACK es representado con la misma estructura de la figura, puede ser enviado en su forma mínima de 5 bytes si la respuesta no regresa un valor, es decir si la petición enviada a la secadora no genera un valor de retorno como un resultado o un dato que deba ser presentado en la GUI principal. A continuación, se describen cada uno de los 6 bytes definidos para el protocolo de comunicación entre la secadora y el sistema embebido:

Header 1: Es considerado el byte de inicio o identificador de un paquete valido, Es el primer byte que se analiza para identificar un paquete valido, la [figura 8](#) muestra el primer byte de un mensaje.

Header 2: Su función se considera como una segunda verificación para, Validar que la trama de bytes recibidos sea un paquete valido.

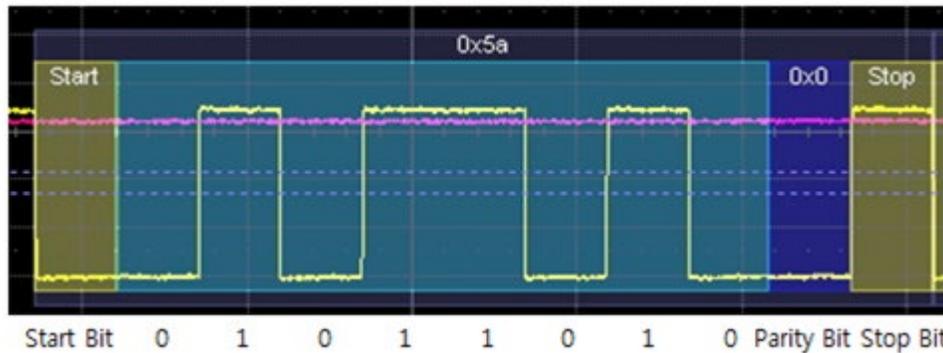


Figura 8.Header enviado por UART.

Elaboración propia

**Data size:** Dado que se conoce la estructura y definición de cada paquete, se agrega un contador de bytes, de modo que al ser validado el byte header se espera un número de bytes conocido por el receptor, esta técnica se implementó para la prevenir la pérdida de bytes en el mensaje.

**Identifizer:** Este byte es el encargado de identificar la prueba a realizar, dentro del software del receptor se encuentran cargadas las funciones del banco de pruebas, con la cual la secadora al recibir el 4 byte analiza y reconoce el protocolo de comunicación transformándolo en una acción.

**Data:** Byte destinado al envió de información, tal como los datos de censado de temperatura, versión de software y estado de la prueba. Enviar y recibir mensajes en formato de texto implica enviar comandos y valores numéricos como letras y palabras legibles por humanos. Los números se envían como la cadena de dígitos representando el valor. Por ejemplo, si el valor de la versión de software es "HL14", es un valor legible para los humanos en código ASCII, el sistema embebido y la PBA de secadora lo interpretan los caracteres H, L, 1 y 4, se envían como caracteres individuales con formato hexadecimal.

**Checksum:** Último Byte destinado para identificar el final de un mensaje y detección de errores en la transmisión, se compone de la suma XOR de cada byte del mensaje enviado o recibido, cualquier cambio en la trama del mensaje

provocaría un cambio en el valor XOR del checksum, interpretándose como un error en la transmisión o pérdida de información.

### **3.5. DISEÑO DE MENSAJE**

En la siguiente sección se expone el protocolo de comunicación creado para esta aplicación en su forma de mensajes con su respectivo ACK.

#### **3.5.1. Encendido**

El primer mensaje enviado destinado a encender la secadora, este mensaje prepara la secadora para iniciar la secuencia de pruebas, en la tabla 3 se observa la cadena de bytes respetando el diseño del protocolo de comunicación.

Tabla 3. Mensaje Encendido

<b>CONFIDENCIAL</b>
---------------------

En la tabla 4 se muestra el mensaje ACK de confirmación que envía la secadora para determinar la recepción del mensaje completo y sin fallas.

Tabla 4. Respuesta ACK Encendido

[

<b>CONFIDENCIAL</b>
---------------------

### 3.5.2. Versión de software

Mensaje destinado a hacer la petición a la secadora para el envío de los datos relacionados con la versión de software, la tarjeta principal de la secadora está conectada con 2 tarjetas las cuales son parte de su propio sistema, Controlan la función de display y la función los botones touch. En el proceso de fabricación existen 37 diferentes variaciones o modelos de secadora, implicando el manejo de 8 tarjetas diferentes combinadas entre sí para formar las especificaciones requeridas del modelo final. Esto hace necesario poder identificar los códigos de tarjetas electrónicas en el área de pruebas, en la siguiente tabla 5 se muestra el protocolo de comunicación para esta tarea con el id 0x41.

Tabla 5. Mensaje software versión.

**CONFIDENCIAL**

En la tabla 6. Se muestra el ACK que devolverá la secadora con los datos obtenidos las tarjetas conectadas a la tarjeta principal PBA.

Tabla 6. ACK mensaje software versión.

**CONFIDENCIAL**

De acuerdo a la tabla anterior podemos observar el bloque 5 destinado para él envió de información proveniente de la secadora contiene 16 bytes, enviar y recibir mensajes en formato de texto que puedan ser interpretados por los humanos implica utilizar código ASCII, los mensajes entre los dispositivos del sistema son en

formato hexadecimal utilizamos 1 byte por cada carácter. Por ejemplo, si el valor de la versión de software es "HL14", es un valor legible para los humanos en código ASCII, se enviaría en paquetes como se muestra en la tabla 7.

Tabla 7. Versión software es

ASCII	HEX	BIN
H	48	0100 1000
L	4C	0100 1100
1	31	0011 0001
4	34	0011 0100

Elaboración propia

Por lo tanto los valores enviados en el mensaje ACK software versión serian como se muestra en la tabla 8.

Tabla 8. Versión SW ACK Ejemplo.

<b>CONFIDENCIAL</b>
---------------------

### 3.5.3. Prueba de encendido

Relaciona con la medición de consumo de potencia, este mensaje está diseñado para entrar en modo test y activar de manera secuencial los componentes eléctricos de carga como son motor y calentador, este mensaje se diseñó cumpliendo los requerimientos para cambiar de acceso manual a automático, haciendo más eficiente la activación de los componentes en un tiempo más corto, en la tabla 9 se puede observar a trama del mensaje power test.

Tabla 9. Mensaje power test.

**CONFIDENCIAL**

La secadora al recibir el mensaje power test muestra en su display "02" como se muestra en la [Figura 9](#), esto indica que secadora entro en modo test (modo diseñado por HQ) donde se mantiene encendido el motor durante 3 segundos, seguido de 1 segundo de encendido para el calentador.

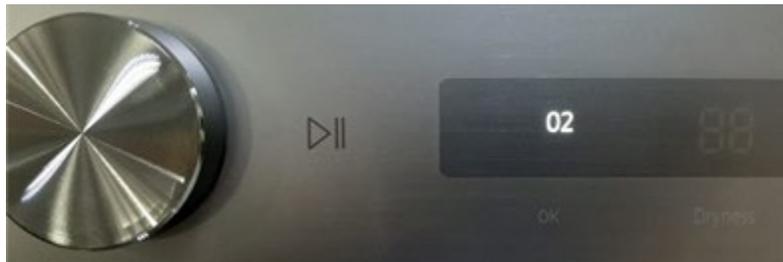


Figura 9. Secadora en modo test.

Elaboración propia

En la tabla 10 podemos observar el mensaje ACK se mantiene con 5 Byte's ya que este mensaje no genera una respuesta con datos a mostrar.

Tabla 10. ACK power test.

**CONFIDENCIAL**

### 3.5.4. Eeprom

El mensaje EEPROM write se compone de 27 byte's, donde los primeros 4 bytes son relacionados al protocolo de comunicación header, número de bytes e id del mensaje, los bytes restantes serán grabados en la eeprom de micro controlador principal de la secadora, a continuación se describen los bytes de la tabla 11.

Data 1: Byte header del paquete que se guardara en la eeprom.

Data 2~5: Almacena la información del model option (identificados de modelos gas, eléctrico y voltajes).

Data 6~20: Bytes que almacenan la información del número de serie de la secadora.

Un Byte por carácter completando 15 byte's del número de serie.

Data 21: Comprende los dos primeros bytes de la versión del SW principal de la secadora.

Tabla 11. Mensaje EEPROM write.



Tabla 12. ACK de mensaje EEPROM write.



### 3.5.5. Lectura EEPROM

Protocolo enviado a la secadora para obtener o leer los datos almacenados en la eeprom, este protocolo de comunicación se diseñó con la finalidad de hacer una

comprobación de los datos grabados y poder determinar fallas en el mensaje guardado.

Tabla 13. Mensaje EEPROM read.



En la tabla 14 se puede observar los 4 primeros Byte's corresponden al protocolo de comunicación, seguido de 21 byte's correspondientes a la información obtenida de la eeprom. Los datos obtenidos de la EEPROM son enviados a la PC master para mostrarlos datos en la GUI.

Tabla 14. ACK de mensaje eeprom read.



### **3.5.6. Modo de prueba**

Protocolo de activación de modo de prueba, es un modo de prueba originalmente cargado para hacer una prueba a nivel usuario final, para poder entrar a este modo de prueba se utilizaba una serie de paso para poder acceder incrementando el tiempo ciclo de la prueba. Es por esa razón que se implementa el siguiente protocolo para poder acceder al modo de prueba por medio del siguiente mensaje (ver tabla 15), donde Data 1 es quien determina si el modelo es de gas o eléctrico.

Tabla 15. Mensaje Test mode.

**CONFIDENCIAL**

En la siguiente tabla 16, se muestra el mensaje ACK que enviara la secadora.

Tabla 16. ACK de mensaje test mode.

**CONFIDENCIAL**

### **3.5.7. Leer temperatura**

El siguiente protocolo de comunicación fue creado para obtener los datos de los 2 sensores de temperatura existentes en la secadora, los sensores son 2 termistores ubicados en la entrada de aire y en la salida del flujo de aire dentro de la secadora. Los datos obtenidos del sensor son convertidos de analógicos a digitales de 10 bits, donde los almacenamos en 2 bytes.

Tabla 17. Mensaje temperatura.

**CONFIDENCIAL**

Tabla 18. Mensaje ACK temperatura.



En la tabla 18 se muestra el mensaje ACK, donde Data1, Data2 representan los valores obtenidos del sensor 1 y Data3, Data4 datos del sensor 2, en la [figura 10](#) se observa un ejemplo de la interpretación que hace el sistema embebido de la temperatura. El valor concatenado de los 2 bytes es enviado a la PC master para ser analizado y mostrado en el resultado final de la GUI.

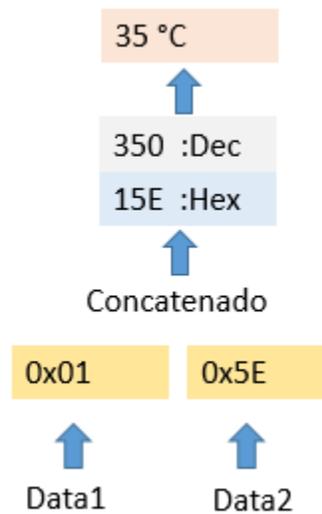


Figura 10. Ejemplo de la concatenación de 2 bytes.

Elaboración propia

### 3.5.8. Power off

El último paso después de completar las pruebas el sistema principal envía el siguiente comando por medio del sistema embebido transmisor. Cuando llega el comando de apagado de la secadora, esta misma envía una señal de respuesta antes de apagarse automáticamente.

Tabla 19. Mensaje power off.



La tabla 20 muestra ACK que envía la secadora una vez que la tarjeta principal reconoció el mensaje, después de enviar el mensaje ACK, la secadora se apaga por medio del software de la tarjeta principal.

Tabla 20. ACK de mensaje power off



### 3.6. ARQUITECTURA DEL SISTEMA

Se muestra en la [figura 11](#) un panorama general de la arquitectura del sistema, Como podemos observar hay una relación con los sistemas actuales, funcionan de manera sincronizada en el área de pruebas.

**El master PC:** Tiene conexión con el PLC, el PLC se encarga de monitorear la posición del set en el área de pruebas, también se encarga de activar los servomotores para que la secadora avance a la siguiente estación.

**Barcode escáner:** envía el número de serie de la secadora para iniciar la prueba.

**Power test:** Equipo encargado de las mediciones de potencia.

**GMES:** Modulo encargado de registrar los resultados de la prueba en la base de datos.

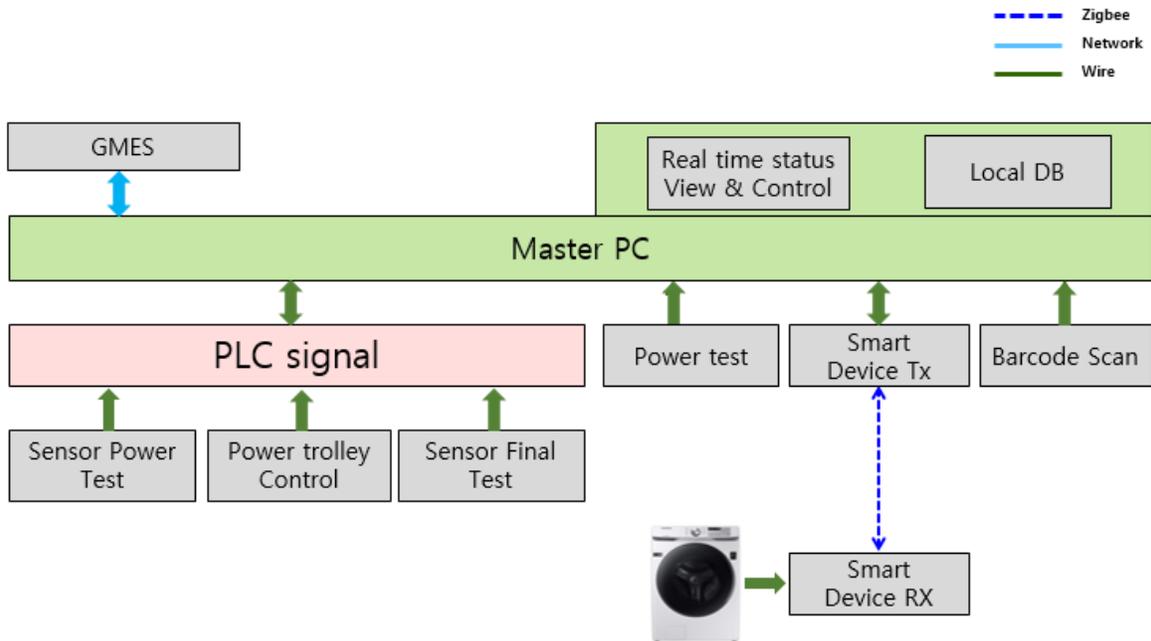


Figura 11. Arquitectura del sistema completo  
Elaboración propia

### 3.6.1. Hardware del sistema

En la [figura 12](#) se muestra el diagrama de hardware del sistema completo, está dividido por 3 bloques input, performance test y output.

**Input:** Representa las entradas del sistema, los componentes involucrados y el AVR que suministra el voltaje para cada secadora. El escáner de código de barras y el sistema embebido en modo transmisor, envían el número de serie de cada secadora.

**Master PC:** En este bloque se encuentran los dispositivos necesarios para el monitoreo y control del proceso de pruebas como son multi port, sensores y device receiver.

**Output:** Bloque donde se encuentran los dispositivos de salida que mostraran la información y estado actual de las pruebas a la secadora.

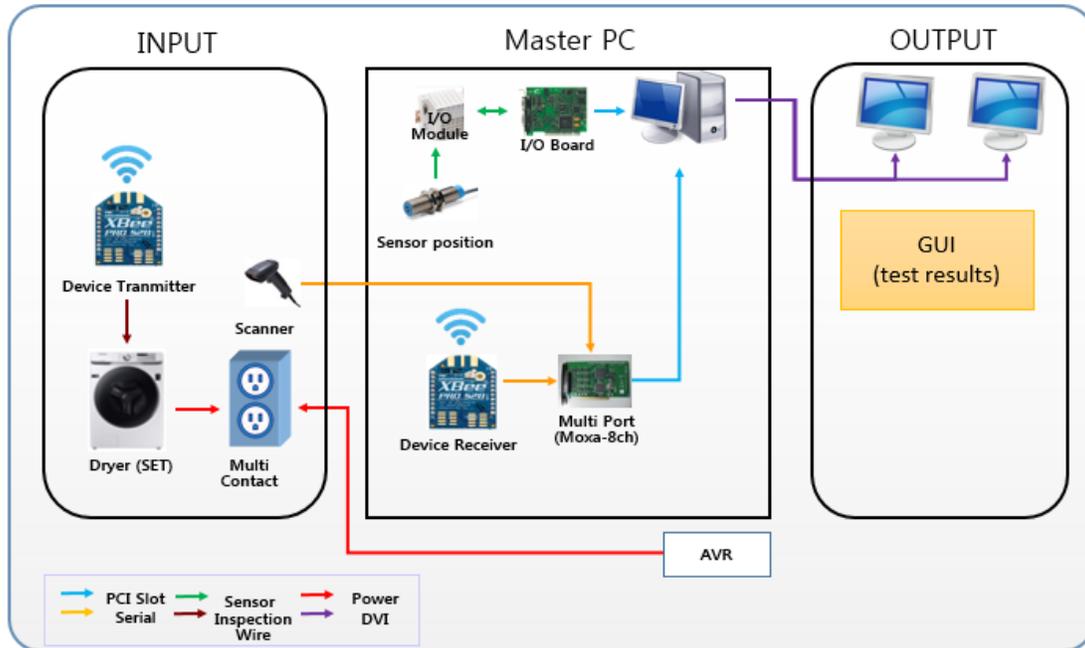


Figura 12. Diagrama de hardware del sistema.

Elaboración propia

### 3.6.2. Hardware del sistema embebido

1.- Módulo transmisor con módulo Zigbee.

- Arduino Uno:
  - Micro controlador: ATmega328.
  - Voltaje Operativo: 5v.
  - Voltaje de Entrada (Recomendado): 7 – 12 v.
  - Pines de Entradas/Salidas Digital: 14 (De las cuales 6 son salidas PWM)
  - Pines de Entradas Análogas: 6.
  - Memoria Flash: 32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader.
  - SRAM: 2 KB (ATmega328)
  
- Módulo Zigbee modelo XBEE S2C, es un módulo de bajo costo que permite realizar conexiones inalámbricas entre dispositivos electrónicos. Trabaja con una frecuencia de 2.4Ghz y te permitirá crear redes de conexión punto a punto, punto a multipunto, broadcast y mesh.

Características:

- Frecuencia de trabajo: 2.4 GHz.
  - Velocidad de datos (máx): Hasta 250.000b/s.
  - Técnica de modulación: DSSS.
  - Rango de señal: Hasta 1200 metros.
  - Sensibilidad: 0 a -80 dBm.
  - Voltaje de la fuente: 2.1 V ~ 3.6 V.
  - Potencia – Recepción: 33 a 45mA.
  - Corriente – Transmisión: 28 a 31 mA.
  - Interfaz de comunicación: SPI, UART.
  - Capacidad de memoria: 32kB Flash, 2kB RAM.
  - Temperatura de operación: -40°C ~ 85°C.
- Shield Arduino RS232, diseñado para el controlador Arduino y puede convertir fácilmente la interfaz UART a RS232.

Características:

- Voltaje: + 5 V
  - 16 puertos IO digitales (incluida una interfaz I2C)
  - 6 puertos de E / S analógicos y alimentación
  - Interruptor de modo de operación y programación
  - LED indicador del transceptor
  - Conectores DB9 (hembra), pines RS232
  - Tamaño del módulo: 55 mm x 53 mm (2,16 "x2,08")
- Shield XBEE, El Xbee Shield simplifica la tarea de conectar un módulo XBee con tu Arduino. Esta tarjeta puede ser utilizada directamente con cualquier tarjeta Arduino o compatible.

Características:

- Fácil interfaz con Arduino que permite la conexión en cascada.
- 3 indicadores LED (ON/SLEEP, RSSI, ASS).
- Provee una corriente máxima de 500mA con 3.3V.

- Conexión configurable con el chip FTDI USB o Arduino con Hardware Serial o Software Serial.
  
- PC Master  
Características:
  - CPU: i5
  - RAM: 8G
  - Moxa 8 puertos RS232
  - LED monitor: 27"
  - OS: Windows 10(64 BIT)

Estos componentes son parte central del proyecto, se comunican entre sí para poder realizar y visualizar los resultados de cada prueba. La PC master concentra toda la información de las 4 estaciones de prueba, es la controladora de la secuencia de las pruebas, enviando señales de inicio de prueba, resultado y fin de la prueba. Las señales se transmiten hacia el receptor esperando la señal para comenzar la comunicación con la secadora.

En la [figura 13](#), se puede ver el esquemático entre el Arduino uno, el módulo Xbee y el puerto de la secadora.

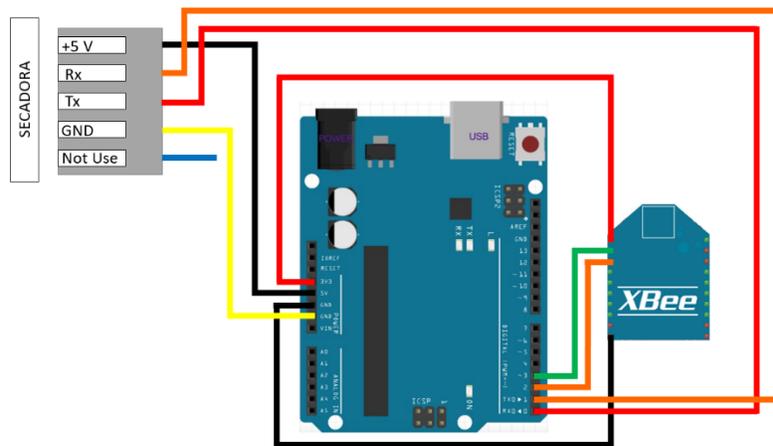


Figura 13. Esquemático de Receptor  
Elaboración propia

Para facilitar las conexiones y el uso de refacciones se optó por utilizar placas de conexiones de propósito específico adaptándose perfectamente al proyecto. Estas placas llamadas Shield Arduino (DFROBOT; 2018), están diseñadas para facilitar la conexión con el arduino, esto las hace de fácil aplicación y presenta beneficios a la hora de hacer cambio físico o mantenimiento preventivo. En la [figura 14](#), se muestra el sistema embebido transmisor de este proyecto.

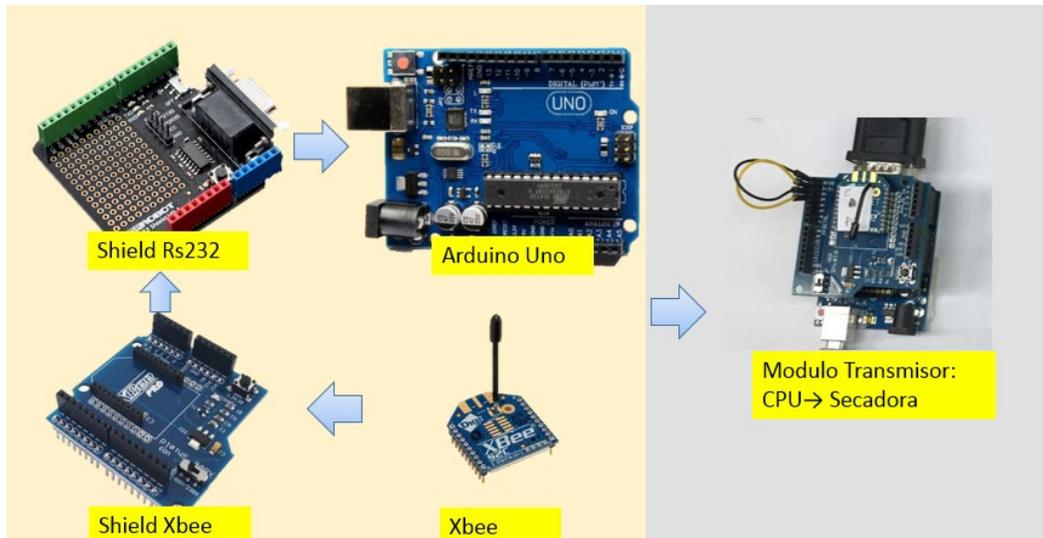


Figura 14. Ensamble SE transmisor.

Elaboración propia

En la [Figura 15](#), se observa el prototipo de los SE transmisor utilizado para cada una de las 4 estaciones de prueba.



Figura 15. SE Trasmisión 4 estaciones

Elaboración propia

### 3.7. INTERFAZ GRÁFICA

La interface gráfica está diseñada para cumplir con los objetivos del proyecto. El diseño presenta los resultados, status de la prueba y menú para el acceso al sistema. La información mostrada tiene la finalidad de visualizar lo referente al proceso de pruebas, así como facilitar el diagnostico en caso de fallas, esto no significa que el operador tenga que determinar si la unidad fallo en la prueba, el sistema analiza de forma automática los datos enviados por la secadora para determinar el resultado de la prueba. En la [Figura 16](#), se muestra la descripción de los elementos de la GUI.

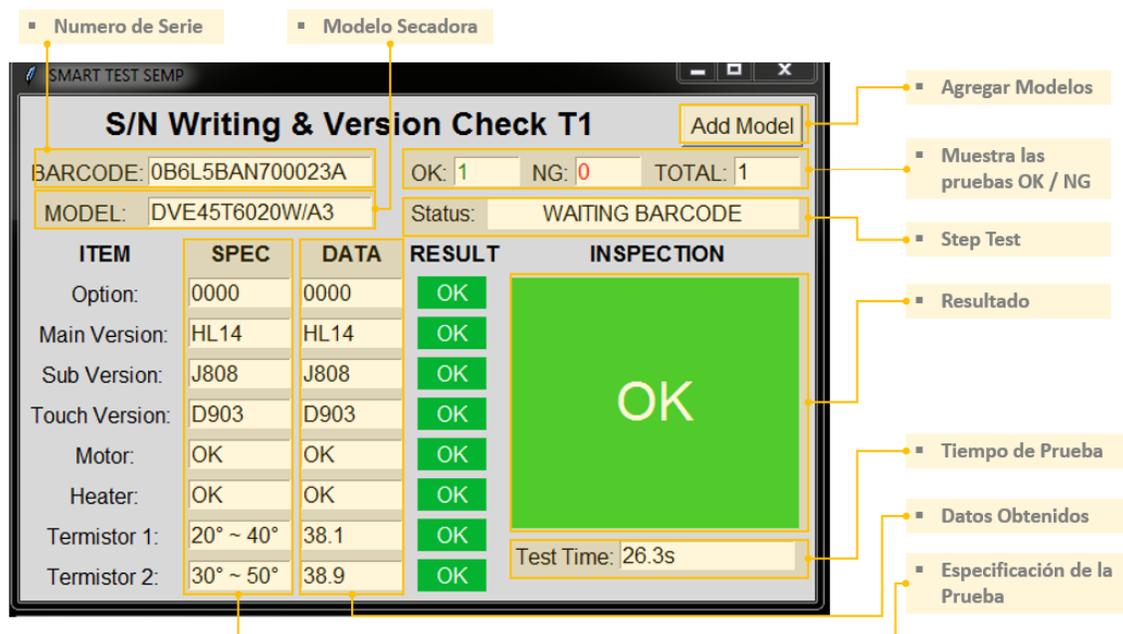


Figura 16. Elementos de GUI.

Elaboración propia

### 3.8. PRUEBAS

#### 3.8.1. Simulación de señales

En la etapa de pruebas se diseñó un pequeño algoritmo para la simulación de las señales presentes en el sistema real. Se prueban todas las entradas del Software principal con las conexiones de los puertos COM simulados con puertos virtuales, ver la [figura 17](#).

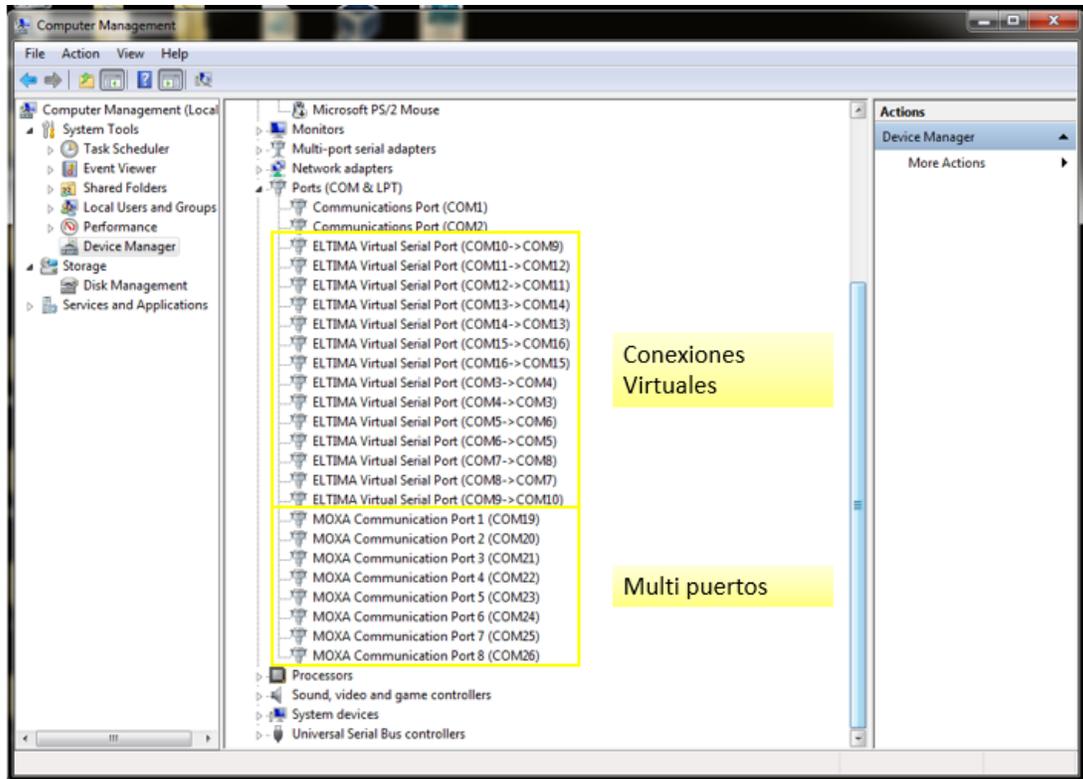


Figura 17. Puertos COM utilizados.

Elaboración propia

Las señales de entrada son simuladas son las siguientes:

- Conexión de scanner para cada caso de prueba.
- La señal de prueba OK.
- La señal de prueba NG.

Estas señales nos sirven para hacer una prueba real entre la secadora y el sistema embebido, tanto receptor como emisor. En la [figura 18](#), se puede observar la interface de software para la simulación de señales.

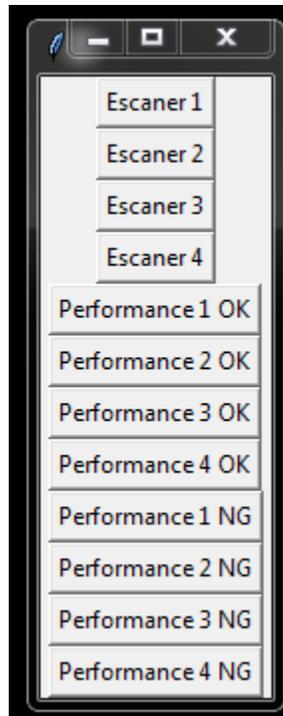


Figura 18. Software para simulación In/out

Elaboración propia

### 3.8.2. Prueba 1

En la prueba que muestra la [figura 19](#), el SW principal espera la señal de inicio, esta señal de entrada esta simulada con 16 dígitos que equivalen a un código serial normal.

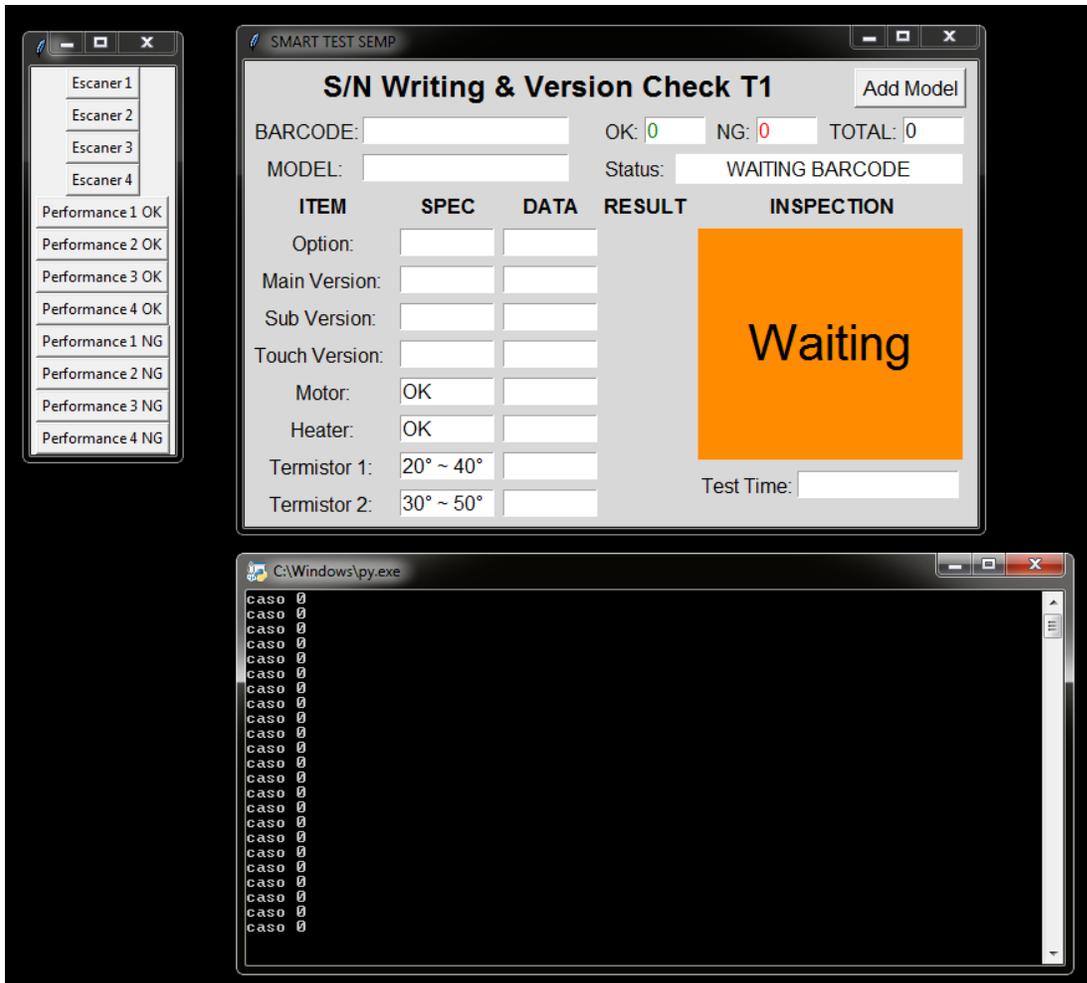


Figura 19. GUI inicial  
Elaboración propia

### 3.8.3. Prueba 2

El software principal al recibir la señal de entrada y los valores de scanner simulado por el puerto COM. Inicia el algoritmo de pruebas enviando los paquetes de datos por medio del emisor conectado a un puerto físico, cada sistema embebido en su modo emisor está conectado a un puerto físico de modo que cada estación de prueba tiene su emisor y su receptor. En la [figura 20](#), se muestra la GUI recibe los datos del scanner y los datos ACK del primer protocolo.



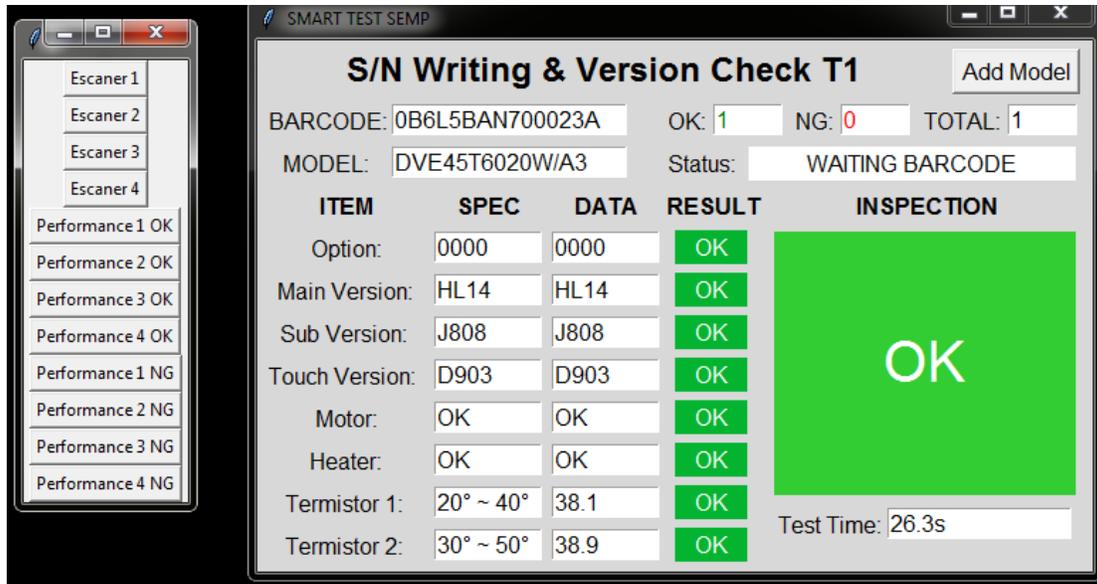


Figura 21. Resultado final de la GUI

Elaboración propia

### 3.8.5. Escenario de pruebas

Con el software de simulación de señales corriendo en el pc master ver [figura 22](#), se realizaron las pruebas de confiabilidad del sistema, el escenario fue el laboratorio de pruebas funcionales, donde las pruebas consistieron en:

- Pruebas funcionales con el sistema para 4 secadoras al mismo tiempo.
- Pruebas de funcionalidad de la aplicación gráfica.
- Pruebas de comunicación entre el sistema embebido receptor y emisor.
- Prueba de funcionamiento y conexión del sistema embebido.



Figura 22. Escenario de pruebas funcionales.

Elaboración propia

En la [figura 23](#), podemos observar el funcionamiento del sistema completo sobre la secadora donde entra a cada modo de prueba al recibir los paquetes de la master PC.



Figura 23. Faces de la prueba

Elaboración propia

Como se observa en la [Figura 24](#). El sistema embebido receptor es el que permanece en comunicación directa con la secadora por medio de cables conectados a su puerto de diagnóstico.

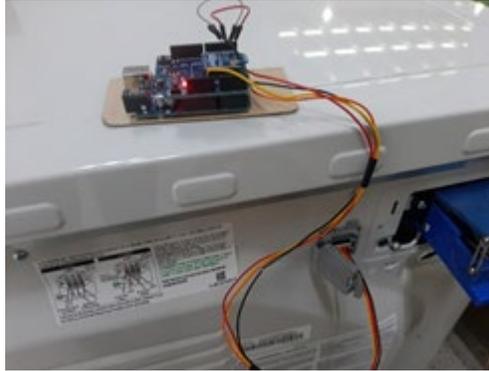


Figura 24. SE receptor conectado a secadora  
Elaboración propia

En la siguiente [Figura 25](#), se muestra el software trabajando de manera simultánea entre las 4 estaciones de prueba. Cada dispositivo puede trabajar de manera independiente, si hay algún fallo en el sistema embebido transmisor o receptor aún puede trabajar el sistema con los otros 3 dispositivos, previniendo así paros de línea por falla en los sistemas embebidos.

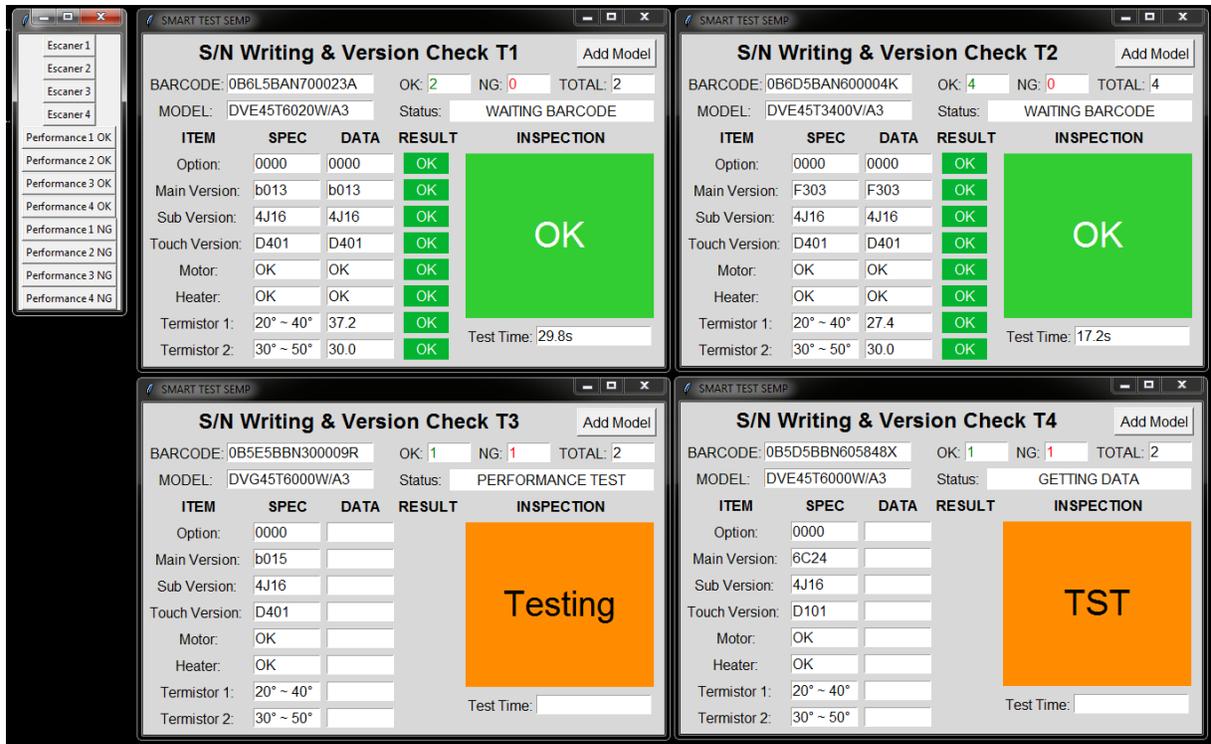


Figura 25. Vista previa de Sistema Operando  
 Elaboración propia

## 4. RESULTADOS

### 4.1. CONECTIVIDAD

Al ser una aplicación estática solo se usará dentro del área de pruebas funcionales, probamos los niveles de RSSI en el ambiente destinado para la aplicación, obteniendo las mediciones de RSSI podemos observar la conexión, así como los niveles de potencia de la señal de comunicación entre el sistema embebido y el sistema permanece conectado a la secadora. Para esta prueba se tomaron mediciones de 50 secadoras por estación, arrojando como resultado el rango de operación dentro de la cabina de pruebas funcionales, ver [figura 26](#).

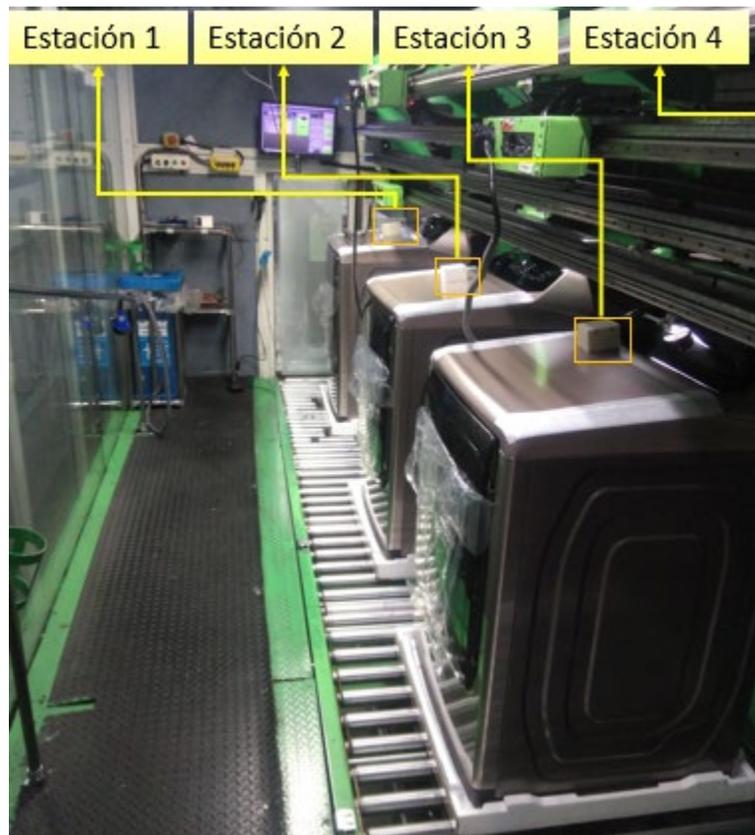


Figura 26. Prueba de rango  
Elaboración propia

En la [Figura 27](#), podemos observar los niveles de RSSI arrojados en cada estación de prueba, los niveles de RSSI obtenidos no sobrepasan el rango de operación del módulo Zigbee S2C, presentando los niveles máximos arriba de los -20 dB en la estación 1,2 y 3, para la estación 4 los niveles no sobrepasan los -20 dB, podemos deducir que cada estación trabaja dentro de un rango valido y al ser una aplicación donde su rango de operación es solo el área de pruebas tiene una distancia de alrededor de 8 metros entre la primer estación y la pc de control donde operan los módulos transmisores, podemos confirmar que la recepción siempre estará dentro del rango de operación.

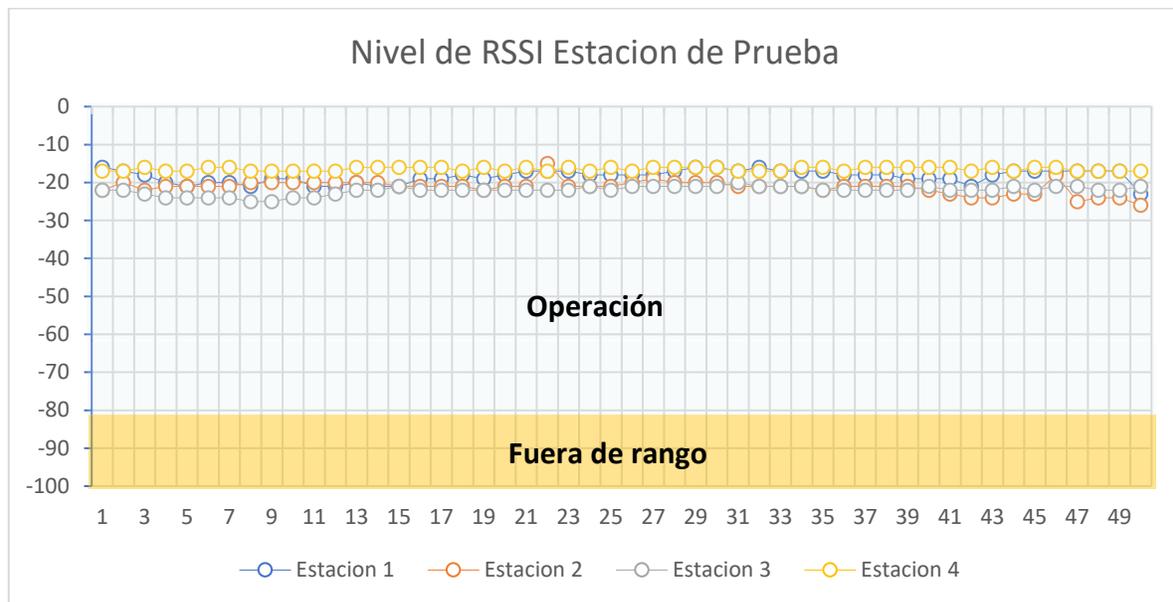


Figura 27. Grafica Niveles RSSI para Estación de Prueba

Elaboración propia

## 4.2. PRUEBA PILOTO

EL proyecto fue instalado en una línea de producción de la planta Samsung Electronics México. Los sistemas embebidos creados para esta aplicación fueron operados por completo en un ambiente de producción normal bajo las siguientes características:

Duración de la prueba: 5 días hábiles.

Tiempo de prueba: 12 horas por día / 60 horas.

Horario de prueba: 8:00a.m ~ 8:00p.m

Lugar: Samsung Electronics Queretaro / Línea B.

Estaciones de prueba: 4.

Producto a probar: Secadora Carga Frontal

Cantidad a probar: 12,574 set's.

En la [figura 28](#), se observa el habiente de prueba piloto.



Figura 28. Estación de Pruebas Línea de Ensamble Secadoras  
Elaboración propia

En la Tabla 21, se observan los resultados obtenidos de una de las cuatro estaciones de prueba con un total de 12,574 secadoras inspeccionadas, arrojando una tasa de re-transmisión de 3% con 345 intentos dentro del intervalo permitido en el algoritmo de detección de errores es de 10 intentos por paquete enviado o recibido.

Esta prueba nos ayuda a reconocer el porcentaje de paquetes perdidos sin poder recuperar fue de 0.16% que equivalen a 20 paquetes que no se pudieron recuperar, en este caso la prueba debió reiniciarse.

Tabla 21. Resultado de prueba piloto.

Día	Línea	Test	Re-transmisión	% <10 intentos	Re-Test	% >10 intentos
1	B_1	1,111	21	2%	2	0.18%
	B_2	1,408	25	2%	1	0.07%
2	B_1	1,179	35	3%	0	0.00%
	B_2	1,355	54	4%	1	0.07%
3	B_1	1,053	58	6%	3	0.28%
	B_2	1,274	40	3%	5	0.39%
4	B_1	1,225	23	2%	2	0.16%
	B_2	1,393	41	3%	1	0.07%
5	B_1	1,255	29	2%	0	0.00%
	B_2	1,321	19	1%	5	0.38%
<b>TOTAL</b>		<b>12,574</b>	<b>345</b>	<b>3%</b>	<b>20</b>	<b>0.16%</b>

Elaboración propia

### 4.3. REDUCCIÓN DE TIEMPO CICLO

En la tabla 22, se han descrito lecturas de tiempos de cada estación donde se instaló el proyecto. Se tomaron 40 mediciones de tiempo de cada una de las estaciones de prueba, para determinar la eficiencia de las pruebas usando protocolo de comunicación versus modo manual.

Tabla 22. Comparación de tiempo manual vs automático.

Prueba	Test Manual	Test con Protocolo	Resultado
	Tiempo Prueba	Tiempo Prueba	% Reducción
1	46.11	34.76	25%
2	46.7	35.14	25%
3	44.11	35.66	19%
4	46.56	35.71	23%
5	48.63	34.84	28%
6	44.6	35.27	21%
7	44.47	34.44	23%
8	46.86	33.86	28%
9	45.68	34.27	25%
10	46.74	35.65	24%
11	45.14	34.46	24%
12	42.18	37.06	12%
13	46.42	36	22%
14	43.78	36	18%
15	46.39	36.46	21%
16	46.47	35.57	23%
17	44.43	34.95	21%
18	45.37	36.11	20%
19	44.28	35.11	21%
20	45.67	34.38	25%
21	47.88	35.6	26%
22	44.9	34.15	24%
23	47.8	34.75	27%
24	44.61	34.9	22%
25	47.97	34.45	28%
26	46.68	36.87	21%
27	44.05	35.76	19%
28	45.63	33.86	26%
29	45.2	35.82	21%
30	46.79	35.87	23%
31	47.18	35.35	25%
32	47.09	36.02	24%
33	44.97	34.95	22%
34	44.48	36.45	18%

<b>35</b>	47.26	36.17	23%
<b>36</b>	44.73	34.88	22%
<b>37</b>	45.78	35.44	23%
<b>38</b>	46.75	34.89	25%
<b>39</b>	44.96	34.13	24%
<b>40</b>	44.87	34.47	23%
<b>Prom</b>	45.75	35.26	23%
<b>Max</b>	48.63	37.06	28%
<b>Min</b>	42.18	33.86	12%
<b>Media</b>	45.675	35.205	23%

Elaboración propia

De las 40 mediciones tomadas en cada una de las cuatro estaciones que componen el área de pruebas funcionales, podemos observar la diferencia entre las operaciones realizadas de forma manual y las realizadas con el protocolo de comunicación a través del sistema embebido, de los cuales obtenemos un promedio de 45.75 s contra 35.36 s de la prueba utilizando el protocolo de comunicación, teniendo una mejora en el tiempo de ejecución de las pruebas de 23% aproximadamente. En la [Figura 29](#), se muestra las operaciones del área de pruebas cuando son realizadas de manera manual.

Stage	#1	#2	#3	#4	TTL
<b>As-Is</b>					48s
Manual	Power (12s)	Power Test 12s	Performance Test 14s	Safety Test 10s	

Figura 29. Caracterización de prueba manual.

Elaboración propia

## CONCLUSIONES

Con los valores obtenidos en el conjunto de pruebas, podemos concluir la existencia de una mayor eficiencia en la realización de pruebas funcionales utilizando el protocolo de comunicación diseñado para esta aplicación. Esto facilita las operaciones de pruebas, la detección de fallas en la secadora y mejora el juicio del resultado de cada prueba, comparado con las operaciones anteriores se realizaban para poder hacer las pruebas, donde el operador debía hacer una serie de pasos para poder entrar al modo de prueba de la secadora utilizando sus dos manos y lo más preocupante era que el operador tenía el juicio total de la prueba es el quien decidía el resultado de la prueba.

Los tiempos de operación en el área de pruebas se estaban trabajando en el límite superior de 14.5 s por estación, es decir había riesgo de generar cuellos de botella, Otro factor importante es la variación de tiempo entre las operaciones, al ser las operaciones de forma manual el operador no siempre las realiza en el mismo tiempo. En la [figura 30](#), se puede observar el tiempo de operación antes de la implementación y en la [figura 31](#) el después de los tiempos de operación del área de pruebas.

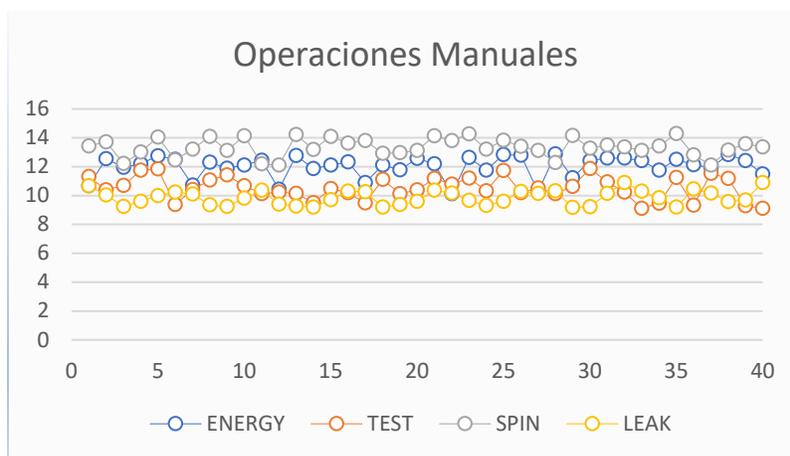


Figura 30. Variación entre operaciones manuales

Elaboración propia

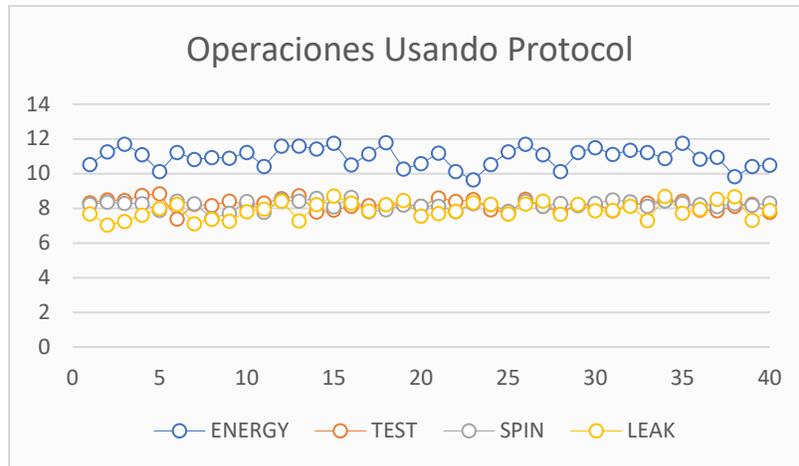


Figura 31. Operaciones Automáticas  
Elaboración propia

En la tabla 23, se observa un resumen de los beneficios y aportaciones del proyecto, es una comparativa entre los resultados obtenidos y los resultados de las operaciones anteriores.

Tabla 23. Resumen de resultados

Actividad	Antes	Después	Resultado
Operaciones	11	16	5 ↑
Tiempo	45.8 s	35.2 s	10.8 s ↓
Operador	3	2	1 ↓
Historial de pruebas	No	Si	Si
Pruebas	5	9	4 ↑
Modo	Manual	Automático	Auto
Comprobación de sw	No	Si	Si
Juicio	Operador	PC	PC

Elaboración propia

De la hipótesis planteada se desprende. Es posible mediante el uso de un protocolo de comunicación específicamente diseñado para esta aplicación lograr realizar pruebas automáticas con mínima manipulación de operadores de forma eficiente,

robusta e impactando considerablemente en el tiempo de producción en serie de secadoras convencionales. Esto se pudo demostrar con los datos de las pruebas tanto simuladas como las que se hicieron físicamente donde obtuvimos un porcentaje de reducción de tiempo de 23%, lográndose un incremento de pruebas de 5 a 9.

Las pruebas consisten en la detección de la versión de software de cada una de las tarjetas electrónicas de la secadora, esta eficiencia en las pruebas se logró gracias al diseño del protocolo. La adición de las funciones de software en la tarjeta electrónica de la secadora, permitió obtener los datos de los periféricos conectados en la secadora para así determinar el resultado del funcionamiento de cada parte logrando la eficiencia en cada prueba además de hacerlo de manera automática y en menor tiempo.

## RECOMENDACIONES

De las conclusiones se desprenden las siguientes recomendaciones:

Con la reducción y estabilidad de los tiempos de operación de las pruebas se recomienda hacer un nuevo balanceo de tiempos y operaciones para ocupar el tiempo ganado por las pruebas automáticas, en operaciones de inspección de apariencia.

Los modelos de secadora desde el 2019 están integrados con hardware de tecnología Smart home, convirtiendo la secadora en dispositivo inteligente capaz de conectarse a red wifi, en un futuro se recomienda utilizar el mismo protocolo de comunicación sin sistema embebido de recepción ya que todos los modelos tendrán conexión a Wifi.

La última recomendación está enfocada en la mejora del sistema embebido, mejorar el diseño para hacerlo más pequeño y manejable, pensando en un diseño propio de shield Xbee y cambiando el microcontrolador, esto para hacer una herramienta diminuta y fácil de manejar.

## **APORTACIÓN DE LA TESIS**

Uno de los aportes más importantes de este trabajo es la documentación del desarrollo de un protocolo de comunicación que puede ser utilizado para comunicar dispositivos electrónicos, La estructura del protocolo documentado puede ser aplicado para la detección de errores en el envío y recibo de paquetes de datos entre dispositivos embebidos, lo que le permite tener una plataforma robusta y flexible basada en una tecnología estándar como lo es el protocolo UART.

## **APORTACIÓN SOCIAL DE LA TESIS**

La disminución del tiempo en la prueba funcional de las secadoras ensambladas en Samsung Electronics México, presenta una aportación importante en la mejora de proceso de inspección, mejorando la ergonomía de las operaciones y así facilitando la jornada de trabajo de los operadores de producción, usando el protocolo de comunicación cambiamos las operaciones manuales por automáticas donde el operador se ha convertido en un inspector dejando atrás las operaciones manuales y repetitivas, las operaciones en el área de pruebas disminuyeron de 8 operaciones a 2 operaciones gracias a la implementación de este proyecto.

## REFERENCIAS

- Castillo, J. A. (2020, marzo 7). *Puerto serie: qué es, para qué sirve y tipos*. Retrieved from Profesional review:  
<https://www.profesionalreview.com/2020/03/07/puerto-serie-que-es-para-que-sirve-y-tipos>
- Chikh, A., & Aldayel, M. (2014). Reengineering requirements specification based on IEEE 830 standard and traceability. *Advances in Intelligent Systems and Computing*. 275, pp. 211-227. Cham: Springer.  
doi:[https://doi.org/10.1007/978-3-319-05951-8\\_21](https://doi.org/10.1007/978-3-319-05951-8_21)
- DFROBOT;. (2018). *RS232 Shield for Arduino*. Retrieved from  
<https://www.dfrobot.com/product-1030.html>
- Gadre, D. V., & Gupta, S. (2018). Universal asynchronous receiver and transmitter (UART). (pp. 151-167). New Delhi: IEEE.
- Ghosh, R. K. (2017). Low power communication protocols: ZigBee, 6LoWPAN and ZigBee IP. *Wireless Networking and Mobile Data Management* (pp. 147-177). Singapore: Springer. doi:[https://doi.org/10.1007/978-981-10-3941-6\\_6](https://doi.org/10.1007/978-981-10-3941-6_6)
- Gilchrist, A. (2016). Industrial Internet use-cases. In A. Gilchris (Ed.), *Industry 4.0: the industrial Internet of thing* (pp. 13-31). Berkeley: Apress.  
doi:[https://doi.org/10.1007/978-1-4842-2047-4\\_2](https://doi.org/10.1007/978-1-4842-2047-4_2)
- Gupta, A. (2019). UART communication. In *The IoT hacker's handbook: a practical guide to hacking the Internet of things* (pp. 59-80). Berkeley: Apress.  
doi:[10.1007/978-1-4842-4300-8](https://doi.org/10.1007/978-1-4842-4300-8)
- Hameed, B., Minguez, J., Wörner, M., Hollstein, P., Zor, S., Silcher, S., . . . Rothermel, K. (2011). The smart real-time factory as a product service system. In J. Hesselbach, & C. Herrmann (Eds.), *Functional thinking for value creation* (pp. 326-331). Berlin: Springer. doi:[https://doi.org/10.1007/978-3-642-19689-8\\_57](https://doi.org/10.1007/978-3-642-19689-8_57)
- Iturriaga de la Fuente, G. (1999). *México Patente n° 2268902*.
- Jens, N., Inka, W., Ganz, W., & Spath, D. (2018). An approach for a quality-based test of industrial smart service concepts. In T. Ahram (Ed.), *Advances in Artificial Intelligence, Software and Systems Engineering. AHFE 2018. Advances in Intelligent Systems and Computing*. 787, pp. 171-182. Cham: Springer. doi:[https://doi.org/10.1007/978-3-319-94229-2\\_17](https://doi.org/10.1007/978-3-319-94229-2_17)
- Jiménez, M., Palomera, R., & Couvertier, I. (2014). *Introduction to embedded systems: using microcontrollers and the MSP430*. New York: Springer.  
doi:[10.1007/978-1-4614-3143-5](https://doi.org/10.1007/978-1-4614-3143-5)

- Lee, Y., Kim, J., Lee, E., Hong, T., & Kim, P. (2018). Implementation of 1:N communication model using serial communication in an RF-based environment. In L. Barolli, F. Xhafa, & J. Conesa (Ed.), *Proceedings of the 12th International Conference on Broadband and Wireless Computing, Communication and Applications*. 12, pp. 376-385. Barcelona: Springer. doi:[https://link.springer.com/chapter/10.1007%2F978-3-319-69811-3\\_34](https://link.springer.com/chapter/10.1007%2F978-3-319-69811-3_34)
- Morales, D. (2013, marzo 3). *Control de flujo en redes de telecomunicaciones*. Retrieved from Slideshare: <https://es.slideshare.net/DanielMoralesMexico/control-de-flujo-en-telecomunicaciones>
- Qian, K., den Haring, D., & Cao, L. (2009). Serial communications. In *Embedded software development with C* (pp. 179-202). Boston: Springer. doi:[https://doi.org/10.1007/978-1-4419-0606-9\\_6](https://doi.org/10.1007/978-1-4419-0606-9_6)
- Song, Y., Rampley, C. P., & Chen, X. (2019). Application of bacterial whole-cell biosensors in health. In Y. Song, C. P. Rampley, & X. Chen, *Handbook of Cell Biosensors* (pp. 1-17). Springer Nature Switzerland AG.
- Tao, Z. (2013). Research and Application of Serial Communication Modules Based on VB. *Advances in Intelligent Systems and Computing* (pp. 1029-1033). Berlin: Springer.
- Tolosa, M. G. (2002, Julio). *Protocolos y modelo OSI*. Retrieved from <http://www.tyr.unlu.edu.ar/pub/02-ProtocolosOSI.pdf>
- Wright, S., Bisson, C., & Duffy, A. (2012). Applying a behavioural and operational diagnostic typology of competitive intelligence practice: empirical evidence from the SME sector in Turkey. *Journal of Strategic Marketing*, 20(1), 19-33. doi:<http://dx.doi.org/10.1080/0965254X.2011.628450>

## **ANEXO**

## ANEXO A

Aplicación del estándar IEEE 830 para la especificación de Requerimientos

### 1. Introducción

#### 1.1 Propósito

El presente documento fue desarrollado con el propósito de describir y resaltar los requerimientos del desarrollo de un protocolo de comunicación entre un sistema embebido y una secadora convencional de ropa para facilitar las pruebas funcionales en la línea de producción de Samsung Electronics Queretaro y este documento se dirigió a los directores de producción e ingeniería quienes formularon y aprobaron.

#### 1.2 Ámbito del Sistema

El sistema embebido que se utilizara para esta aplicación debe ser capaz de comunicarse con la secadora, ser de bajo costo (considerando el mantenimiento preventivo) y fácil de programar. Debe ser capaz de enviar paquetes de bytes completos sin errores para que la secadora pueda identificar las órdenes o peticiones requeridas por el usuario final.

- Comunicar la secadora con la GUI para mostrar información de la prueba.
- Generar una base de datos de las pruebas realizadas y sus resultados.
- Pruebas secuenciales automáticas.
- Análisis de resultados de las pruebas automáticas y veredicto final.

#### 1.3 Definiciones, Acrónimos y Abreviaturas

UART: Universal Asynchronous Receiver-Transmitter  
GUI: comunicarse con la Zigbee: es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica

PBA: Printing Board Assembly

ACK: acknowledgement, en español acuse de recibo o asentimiento

Header: uno o más caracteres especiales que identifican el inicio del mensaje.

Checksum: Una suma de verificación de Byte's

#### 1.4 Referencias

N/A.

#### 1.5 Visión General del Documento

En siguiente documento de especificación de requerimientos se describen más detalles sobre el funcionamiento del sistema embebido y el protocolo de comunicación, partiendo desde una Descripción General la cual describe el contexto y los ambientes sobre los cuales se implementará.

### 2. Descripción General

#### 2.1 Perspectiva del producto

El sistema embebido será de bajo costo y enviara paquetes de datos ordenados de acuerdo al diseño del protocolo de comunicación. Los

datos serán enviados en forma serial por protocolo Zigbee a una PC de control, donde se mostrara el resultado de las pruebas y las demás funciones. Este documento expone los requerimientos del sistema embebido y GUI.

## **2.2 Funciones del producto**

Sistema embebido:

- Debe ser una tarjeta de propósito general, de bajo costo y con mínimo 1 puerto UART.
- Debe ser capaz de conectar con la secadora en el puerto de servicio originalmente de 5 pines.
- Debe ser capaz de conectar con la PC, con la interface principal y de control.

GUI:

- Interface de usuario: El usuario debe ser capaz de visualizar de manera ordenada los siguientes componentes:
  - Motor: Estado del motor, Resultado de prueba de funcionamiento.
  - Calentador: Estado del calentador, Resultado de prueba de funcionamiento.
  - Sensores 1: Resultado de temperatura obtenido durante la prueba.
  - Sensores 2: Resultado de temperatura obtenido durante la prueba.
  - SW versión: De microcontrolador asociado al modelo de secadora que se probara.
  - Escritura de EEPROM: Debe ser capaz de mostrar el resultado de la escritura en memoria de la tarjeta de secadora.
  - Ver datos de la memoria: Debe ser capaz de acceder los datos almacenados en la memoria EEPROM de la tarjeta de secadora.
  - Historial de pruebas: Guardar y generar reporte de las pruebas realizadas.
  - Tiempo: Mostrar el tiempo de la prueba.
  - Datos de modelo: Debe mostrar los datos relacionados al modelo y número de serie.

## **2.3 Características de los usuarios**

El proyecto fue diseñado pensando en el operador de producción que utilizara la herramienta como parte de su proceso de inspección.

## **2.4 Restricciones**

### **2.4.1 Restricciones físicas**

El sistema embebido solo se utilizara dentro de la cabina de inspección, la cabina de inspección no sobrepasa los 8 metros.

Se debe considerar para agregar más funciones al protocolo de comunicación se debe de seguir el proceso de cambios al proceso, que debe ser aprobado por los directivos.

### **2.4.2 Limitaciones del hardware**

El hardware se limitará únicamente a mantener la conexión entre la secadora y la PC. Solo puede ser controlado o recibir órdenes desde la PC por medio del sistema embebido en modo transmisor. Cada transmisor y receptor tendrá una conexión punto a punto, es decir no podrá enviar o recibir datos de otro device.

### 2.4.3 Interfaces con otros sistemas

El sistema embebido recibe los paquetes y las ordenes desde la interface gráfica, Convirtiéndose en esclavo de la PC esperando recibir los paquetes de la comunicación.

### 2.4.4 Protocolos de comunicación

Para la aplicación de este sistema se manejaran 3 protocolos de comunicación UART, Zigbee y protocolo diseñado para esta aplicación donde solo puede ser comprendido por la secadora.

## 2.5 Suposiciones y dependencias

### 2.5.1 Suposiciones:

- Por políticas de la organización solo el personal autorizado puede entrar a la PC del área de pruebas
- El desarrollo del prototipo debe ser de bajo costo.
- Los resultados de las pruebas se cargaran a la red del sistema GMES

### 2.5.2 Dependencias:

- El hardware principal deberá ser un Arduino uno (ATmega328P). Se adapta perfectamente a Shield Zigbee y RS232, con alguno otro modelo se deberá cambiar el shield.
- Modulo Zigbee debe ser XBee S2C, cambio en el modelo implicaría perdida en los datos o una reconfiguración del sw.
- El arnés que conecta con la secadora debe ser de 5 pines(aunque solo se usen 4), por arquitectura de fabricación de las secadoras.

### 2.6 Requisitos futuros

Las siguientes mejoras a Sistema embebido detectadas a partir de este documento son:

- Implementar un sistema de baterías recargables para mejorar la conexión.
- Implementar una gráfica en tiempo real de la temperatura de la secadora, para facilitar el diagnostico de fallas.
- Extender el protocolo de comunicación a los demás productos de fabricación en Samsung electronics.
- Estandarizar las funciones del protocolo.

## 3. Requisitos Específicos

Los requisitos se dividen en 2 secciones requisitos funcionales y no funcionales, así mismo se identificarán de acuerdo con el siguiente sistema:

Requisitos Funcionales:

ID Req	Requerimiento Funcional	Descripción
RF1	Mostrar Datos	Visualizar información de los periféricos del sistema
RF2	Ver datos de la memoria	Accesar a memoria EEPROM de la Tarjeta de secadora.
RF3	Solicitar pruebas	Solicitar pruebas de cada componente o de todos
RF4	Comunicar PBA	Debe ser capaz de establecer comunicación sin errores
RF5	Enviar protocolo	Debe ser capaz de entender los mensajes enviados entre pba
RF6	Configurar puerto	Configura el medio de transmisión
RF7	Controlar Motor	Debe ser capaz de controlar y obtener datos desde la pba
RF8	Controlar calentador	Debe ser capaz de controlar y obtener datos desde la pba

RF9	Monitorear sensores	Debe ser capaz de obtener datos desde la pba
RF10	Solicitar pruebas	Debe ser capaz de hacer pruebas automáticas
RF11	Comparar checksum	Debe comparar checksum del micro controlador de secadora
RF12	Checar datos	Debe enviar datos para mostrar en la GUI
RF13	Setpoint	Debe comparar los datos en tiempo real y comparar con set
RF14	Retransmitir protocolo	Debe transmitir el protocolo sin errores
RF15	Detección de error	Debe ser capaz de detectar la cadena de datos enviada.

Requisitos No Funcionales:

A continuación se enlistan los requisitos necesarios para el sistema pero no son parte del funcionamiento propio del sistema embebido.

ID Req	Requerimiento no Funcional	Descripción
RNF1	GUI	Solo se puede iniciar prueba por medio de la GUI
RNF2	Power test	El medidor de potencia debe estar encendido
RNF3	Zigbee	Deben ser configurados antes de usarse en la aplicacion
RNF4	idioma	Sistema solo puede usar español

### 3.1 Interfaces Externas

Los requisitos los clasificaremos por Requisitos de Entrada, Requisitos de Hardware y Requisitos de Salida y serán identificados de la siguiente manera:

#### Requisitos de Entrada:

RE01: El valor del escáner con número de serie.

RE02: Voltaje de alimentación de la secadora.

#### Requisitos de hardware:

RH01: Arduino uno (ATmega328P).

RH02: Shield RS232 arduino uno.

RH03: Shield Zigbee arduino uno.

RH04: Zigbee modelo XBee S2C

#### Requisitos GUI:

RG01: Mostrar los valores de número de serie.

RG02: Mostrar el modelo.

RG03: Mostrar el resultado de calentador.

RG04: Mostrar el resultado de motor.

RG05: Mostrar el resultado de sensor 1.

RG06: Mostrar el resultado de sensor 2.

RG07: Mostrar el resultado de Sw versión.

RG08: Mostrar el resultado es estado de la prueba.

RG09: Mostrar el conteo de las pruebas.

RG05: Mostrar el tiempo de la prueba.

### 3.2 Funciones

- Sistema embebido está formado por dos partes una el modulo receptor y otra el modulo transmisor, a continuación se describe su funcionamiento:

**Entradas:** Los valores de entrada son proporcionados por la PC master.

**Proceso:** El modulo transmisor recibirá las ordenes provenientes de la pc, procesar las ordenes y enviara los datos al módulo receptor, al ser un enlace punto a punto el receptor analiza los paquetes recibidos y determina que protocolo o paquete se le enviara a la secadora, una vez recibido el paquete por la secadora envía un ACK al módulo receptor para que reenvíe ese valor al módulo emisor y pueda ser mostrado por la GUI.

**Salidas:** Los paquetes recibidos por el modulo emisor deben ser procesados y mostrados en la GUI, al término de la prueba los resultados podrán ser mostrados en las 2 pantallas del área de pruebas.

### 3.3 Requisitos de Rendimiento

El sistema debe poder realizar las siguientes funciones:

- Conexión Sistema Embebido→PC: Debe ser capaz de conectar con la PC, con la interface principal y de control.
- Conexión PC→ Sistema Embebido: Debe ser capaz de conectar con 4 dispositivos al mismo tiempo.
- PC: debe controlar las señales de inicio, monitoreo y termino de las pruebas.
- El sistema embebido no almacenara información de las pruebas, solo enviara los paquetes con la información del resultado
- El modulo transmisor solo enviara los comandos que sean enviados por la GUI.
- El modulo receptor solo podrá comunicarse o enviar paquetes de datos a la secadora, si las ordenes vienen de la GUI.

### 3.4 Restricciones de Diseño

- las placas de los módulos receptor y transmisor son de propósito general y de bajos costo.
- El campo de operación del sistema es únicamente en el área de pruebas de Samsung Electronics Queretaro, específicamente en el área de pruebas funcionales, que es un área de menos de 8 metros desde la pc master hasta cada uno de los dispositivos
- Solo se podrán probar 4 secadoras al mismo tiempo no más.

### 3.5 Atributos del Sistema

La GUI estará diseñada en lenguaje de programación Python y el protocolo de comunicación diseñado para esta aplicación debe seguir las normas establecidas en este documento.

### 3.6 Otros Requisitos

N/A