

INFOTEC CENTRO DE INVESTIGACIÓN E
INNOVACIÓN EN TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIÓN

DIRECCIÓN ADJUNTA DE INNOVACIÓN Y
CONOCIMIENTO
GERENCIA DE CAPITAL HUMANO
POSGRADOS

“Prototipo de dispositivo para monitoreo de estatus operacional de máquinas industriales (sistema Andon)”

PROYECTO DE TITULACIÓN
Que para obtener el grado de MAESTRO EN
SISTEMAS EMBEBIDOS

Presenta:

José Ramón Viramontes Olivar

Asesor:

Dr. Víctor Miguel Hernández Maldonado

Aguascalientes, Octubre, 2021.

AUTORIZACIÓN DE IMPRESIÓN Y NO ADEUDO EN BIBLIOTECA
MAESTRÍA EN SISTEMAS EMBEBIDOS

Ciudad de México, 20 de octubre de 2021
INFOTEC-DAIC-GCH-SE-0339/2021.

La Gerencia de Capital Humano / Gerencia de Investigación hacen constar que el trabajo de titulación intitulado

"Prototipo de dispositivo para monitoreo de estatus operacional de máquinas industriales (sistema Andon)".

Desarrollado por el alumno: José Ramón Viramontes Olivar y bajo la asesoría del Dr. Víctor Miguel Hernández Maldonado; cumple con el formato de Biblioteca. Por lo cual, se expide la presente autorización para impresión del proyecto terminal al que se ha hecho mención.

Asimismo, se hace constar que no debe material de la biblioteca de INFOTEC.

Vo. Bo.



Lic. Juan Ramón Abarca Damián
Coordinador de Biblioteca

Anexar a la presente autorización al inicio de la versión impresa del trabajo referido que ampara la misma.

C.p.p Servicios Escolares

Agradecimientos

Dedicado a mi madre Ramona, mi padre José, mi esposa Natsuki, mi hermano Gibran y especialmente a mi abuelo Ramón, abuelito en donde quiera que estés, esta te la dedico a ti, a todos ellos gracias infinitas.

Tabla de contenido

Resumen	18
Introducción	1
<i>Capítulo 1 Contexto Tecnológico.</i>	4
1.1 Redes Zigbee	4
1.2 Los dispositivos Xbee en aplicaciones industriales	7
1.3 Sistemas Andon.....	8
1.4 Cálculo de OEE	9
1.5 Definición del problema.....	14
1.7 Objetivo general.....	15
1.8 Objetivos específicos	15
1.9 Justificación.....	16
<i>Capítulo 2 Desarrollo del proyecto.</i>	20
2.1 Modelo de desarrollo.....	20
2.2 Cálculo de KPI's principales para obtención de OEE.....	23
2.3 Especificación de requerimientos.....	24
2.4 Elección de tecnologías.....	25
2.5 Arquitectura general de comunicación	28
2.6 Formato de paquetes de datos.....	30
2.7 Diagrama de flujo del proceso de recolección de datos.....	31
2.8 interpretación de datos en el lado de la aplicación web.	32
2.9 Obtención de eficiencia general	34
.....	39
<i>Capítulo 3 Pruebas, resultados y validación.</i>	40
3.1 Pruebas y resultados de la solución tecnológica propuesta.....	40
3.2 validación de la solución tecnológica propuesta	47
3.3 Mejoras y puntos de oportunidad detectados	48
3.4 Estándares, y Registros de patentes	49
3.5 Impacto social	50
3.6 Impacto tecnológico	51
3.7 Impacto económico	51
3.8 Impacto ambiental.....	52
<i>Conclusiones</i>	53

<i>Bibliografía</i>	55
<i>Anexo A - Diagrama principal</i>	58
<i>Anexo B - Diagrama de salidas digitales</i>	59
<i>Anexo C - Diagrama de entradas digitales</i>	60
<i>Anexo D - Diagrama de comunicaciones</i>	61
<i>Anexo E - Firmware de microcontrolador</i>	62

Índice de figuras

<i>Figura 1 Topología estrella[20].....</i>	<i>5</i>
<i>Figura 2 Topología de árbol[20]</i>	<i>5</i>
<i>Figura 3 Topología de malla[20].....</i>	<i>6</i>
<i>Figura 4 Topología de malla[20].....</i>	<i>6</i>
<i>Figura 5 Dispositivo Xbee Pro</i>	<i>7</i>
<i>Figura 6 Interpretación de colores rojo, amarillo, verde en sus primeras fases de implementación.</i>	<i>8</i>
<i>Figura 7 Ejemplo de contenido de una pantalla de Leds tradicional con los KPI's estándar.</i>	<i>12</i>
<i>Figura 8 Tablero con target vs producción actual.</i>	<i>13</i>
<i>Figura 9 Ejemplo de tablero de led tradicional con indicadores de estado Andon[11]......</i>	<i>14</i>
<i>Figura 10 Ciclo de procesos de desarrollo de sistemas embebidos[13]</i>	<i>20</i>
<i>Figura 11 Modelo en espiral de Boehm[12].</i>	<i>22</i>
<i>Figura 12 Diseño de primer prototipo de tarjeta electrónica.....</i>	<i>25</i>
<i>Figura 13 Tarjeta prototipo con componentes ensamblados.</i>	<i>28</i>
<i>Figura 14 Diagrama arquitectónico de comunicación de la solución propuesta.</i>	<i>29</i>
<i>Figura 15 Descripción de contenido de paquete de datos transmitido.....</i>	<i>30</i>
<i>Figura 16 Ciclo de recolección de datos.....</i>	<i>31</i>
<i>Figura 17 Registro de eventos.</i>	<i>32</i>
<i>Figura 18 Ejemplo de interpretación de un pulso digital.....</i>	<i>33</i>
<i>Figura 19 Interpretación de registros en una línea de tiempo.</i>	<i>34</i>
<i>Figura 20 Ejemplo de interpretación de registros de eventos de partes producidas.</i>	<i>35</i>
<i>Figura 21 Primer prueba con botón simple.</i>	<i>40</i>
<i>Figura 22 Ensamble con botonera de 4 selectores, cada selector cuenta con dos estados sumando en total 8 estados posibles.</i>	<i>41</i>
<i>Figura 23 Captura de pantalla de tablero virtual de aplicación web Enlacex visualizado en un navegador Chrome</i>	<i>42</i>
<i>Figura 24 Base receptor Xbee Universal.....</i>	<i>43</i>

Figura 25 Primer prueba real de recolección de datos(Interpretación de datos utilizando software Enlაცex). 44

Figura 26 Línea de tiempo y gráfica de pastel con tiempos acumulados por estado operacional(Interpretación de datos utilizando software Enlაცex). 45

Figura 27 Reporte de resultados obtenidos de la prueba(Interpretación de datos utilizando software Enlაცex). 46

Figura 28 Gráfica de piezas producidas y piezas defectuosas por hora(Interpretación de datos utilizando software Enlაცex)..... 46

Figura 29 Visualización de registro de eventos(Interpretación de datos utilizando software Enlაცex)..... 47

Índice de tablas

Tabla 1 Requerimientos funcionales.	24
Tabla 2 Requerimientos no funcionales.	24
Tabla 3 Validación de requerimientos funcionales	48
Tabla 4 Validación de requerimientos no funcionales	48
Tabla 5 Tabla general de interpretación de colores en un sistema Andon.....	50
Tabla 6 Tabla comparativa de precios de implementación.	51

Siglas y abreviaturas

MTBF: *Mean time Between Failure (Tiempo medio entre fallas).*

MTTR: *Mean time to Repair (Tiempo medio de reparaciones).*

OEE: *Overall Equipment Efficiency (Eficiencia general de Equipo).*

KPI: *Key Performance Indicator (Indicador clave de rendimiento).*

ANDON: *Es una ayuda visual que alarma y resalta dónde se requiere la acción en una planta de manufactura.*

CSV: *comma separated values (valores separados por coma).*

SQL: *Structured Query Language (lenguaje de consulta estructurada).*

IEEE: *Institute of Electrical and Electronics Engineers (Instituto de Ingeniería Eléctrica y Electrónica).*

PLC: *Programmable Logic Controller (Controlador lógico programable).*

TCP/IP: *Transmission Control Protocol / Internet Protocol (Protocolo de Control de Transmisión/ Protocolo de Internet).*

RS232: *Recommended Standard 232 (Estándar Recomendado 232).*

RoHS: *Es una sigla que proviene del Inglés y significa: "Restriction of Hazardous Substances".*

LR-WPAN: *low-rate wireless personal area network, protocolo de comunicación utilizado para el control de acceso a redes inalámbricas a tasas bajas de transmisión de datos .*

Resumen

Este trabajo se enfoca al desarrollo de un prototipo de sistema embebido, que tiene la función de recopilar información de los cambios de estado de una línea de producción a través de la conexión de señales digitales con una torreta industrial. Dicho sistema representa a un estado por medio del encendido de lámparas de diferente color, envía los datos de los cambio de estado vía inalámbrica a un receptor y de esta manera obtiene en tiempo real, si la línea de producción está en modo de trabajo, si está en fallo, si se está siendo intervenida para hacerle mantenimiento, si hace falta material, etc.

La información extraída de la torreta fue utilizada para alertar a personal remoto de los cambios de estado de la línea de producción, por medio de una plataforma web privada y para determinar el porcentaje de eficiencia general al que trabaja y facilita un análisis del registro histórico de los datos que fueron almacenados.

Los resultados de reducción de costos de implementación de ~64.82% (ver página 51) obtenidos de esta investigación abren una prometedora línea de desarrollo de sistemas embebidos que tienen por objeto el monitoreo de líneas de producción, vale la pena decir que de manera general, los objetivos de este trabajo fueron cumplidos satisfactoriamente y serán de utilidad en futuros proyectos similares para la industria 4.0.

Introducción

En este documento se detalla el diseño, selección de componentes y construcción de un sistema embebido prototipo que se utilizará para la recolección y transmisión de datos de estatus operacional de una máquina o línea de producción por medio de su interconexión a una torreta industrial, con la condicionante de que dicha máquina o línea utilice el estándar de torretas de colores Andon (ver Figura 6) para indicar su estatus operacional, para el desarrollo de este proyecto se implementa el ciclo de procesos de desarrollo de sistemas embebidos[13] propiedad de Infotec limitando el alcance hasta el paso cinco (integración y pruebas). Para el muestreo y procesamiento de datos utilizaremos la plataforma web propiedad de la empresa Grupo Nauko la cual proporcionó información pertinente para poder enlazar datos de nuestra tarjeta electrónica prototipo, esta tarjeta tiene una serie optoacopladores para detectar eventos de activación de la torreta y enviar los cambios de estado inalámbricamente al receptor de datos, también tendrá una serie de relevadores de estado sólido para activar o desactivar salidas en casos concretos e interactuar con la torreta. De tener éxito, en etapas posteriores se podría implementar en conjunto con el sistema informático de la empresa el cual actualmente se utiliza en diversas empresas de la industria manufacturera.

Capítulo 1

Contexto Tecnológico

Capítulo 1 Contexto Tecnológico

A pesar de no ser un término muy nombrado, los sistemas embebidos hoy en día se encuentran en casi todas las áreas de nuestra vida como en el hogar con los electrodomésticos, en nuestra vida social con dispositivos móviles, el trabajo con impresoras a color, pantallas LED, laptops, equipos industriales y una amplia variedad áreas y dispositivos los cuales en su totalidad están basados de una u otra manera en un sistema embebido, estos sistemas se componen de dos partes principales hardware y software, en conjunto ambas partes se diseñan con una función en específico y dicha función puede ser parte de una solución única o puede formar parte de una solución más grande.

Actualmente existe una tendencia de utilizar sistemas embebidos para la recolección y procesamiento de datos de forma masiva, dicha tendencia es conocida como Big Data [16][18], debido a la gran cantidad de datos que se procesan también se requiere de métodos más eficientes para la recolección de dichos datos, esto nos lleva al auge en desarrollo de sistemas embebidos más complejos [14] enfocados en la optimización de protocolos de comunicación inalámbricos utilizados para la transferencia de información hacia sistemas de almacenamiento locales [15] para después ser enviados a cualquier sitio en la internet, uno de los protocolos inalámbricos más utilizados para dispositivos IoT es el protocolo Zigbee que trabaja sobre el estándar IEEE 802.15.4 definido en el nivel físico y el control de acceso a redes inalámbricas a tasas bajas de transmisión de datos (low-rate wireless personal area network, LR-WPAN).

1.1 Redes Zigbee

El estándar zigbee se utiliza en una amplia variedad de aplicaciones donde la necesidad de transferencia de datos es mínima, aplicaciones tales como la domótica en sistemas de control de temperatura e iluminación, en sistemas de control automático y automatización.

Una red Zigbee está compuesta por varios dispositivos que pueden ser configurados para cumplir tres funciones diferentes, como dispositivo final para recolectar datos directo de los sensores, como router para recopilar datos de los dispositivos finales o como coordinador, este último se encarga de recopilar datos de los routers que a su vez recopilan datos de los dispositivos finales, entrando un poco más en detalle, el uso selectivo de estos dispositivos en sus diferentes configuraciones nos permite establecer cuatro tipos de topologías de red.

1.1.1 Topología de estrella:

En esta topología se configura un dispositivo como coordinador y los demás como dispositivos finales, esta es la más sencilla y práctica de aplicar.

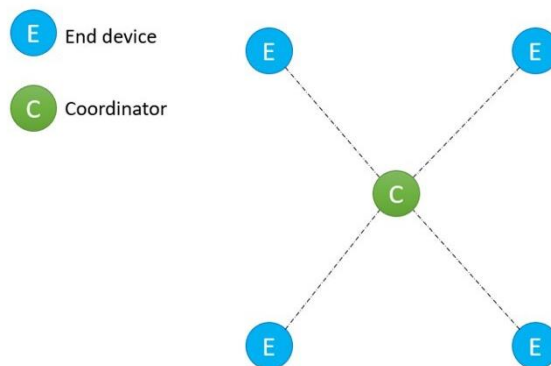


Figura 1 Topología estrella. Fuente: elaboración propia basada en T. Kumar and P. B. Mane, "ZigBee topology: A survey," 2016. [20]

1.1.2 Topología de árbol:

Esta red coordinadores (padres) , routers y dispositivos finales(hijos).

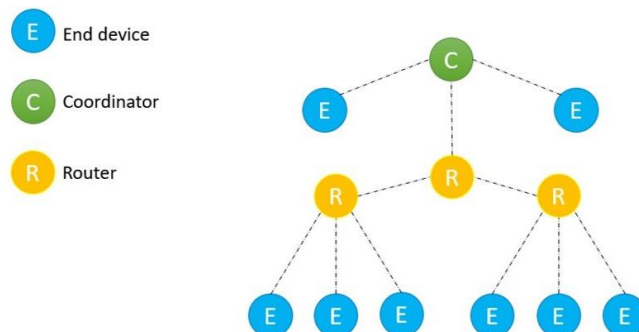


Figura 2 Topología de árbol. Fuente: elaboración propia basada en T. Kumar and P. B. Mane, "ZigBee topology: A survey," 2016. [20].

1.1.3 Topología de clúster de árbol:

Los coordinadores con ramificaciones son llamados clústeres y cada clúster es identificado y conectado a la red con un id.

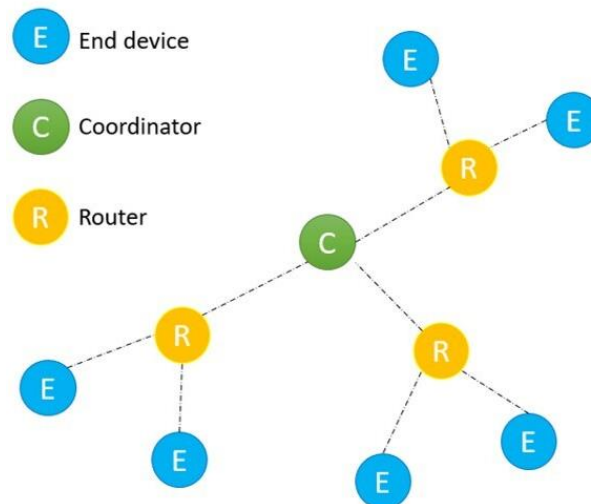


Figura 3 Topología de malla. Fuente: elaboración propia basada en T. Kumar and P. B. Mane, "ZigBee topology: A survey," 2016. [20]

1.1.4 Topología Malla:

Este tipo de red, también llamada peer-to-peer consiste en que cada dispositivo de dicha red puede comunicarse con cualquier otro dispositivo dentro de la misma red.

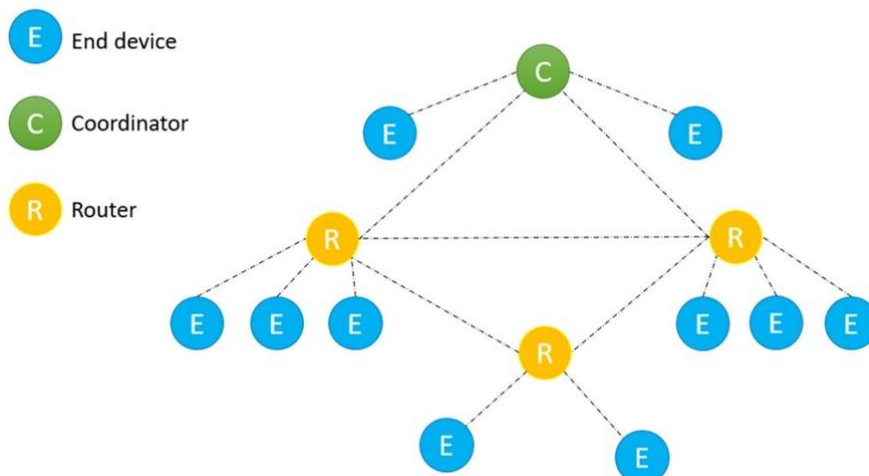


Figura 4 Topología de malla. Fuente: elaboración propia basada en T. Kumar and P. B. Mane, "ZigBee topology: A survey," 2016. [20]

Existe una variedad amplia de fabricantes y marcas de sistemas embebidos que manejan el estándar Zigbee, los cuales al funcionar bajo el mismo estándar de

comunicación IEEE 802.15.4 permite integrar sistemas de recolección de datos con diferentes dispositivos sin complejos problemas de compatibilidad, por esta razón es que suelen ser utilizados en ambientes industriales que requieren y demandan trabajar sobre estándares comunes.

1.2 Los dispositivos Xbee en aplicaciones industriales

Una de las marcas más comunes de sistemas embebidos que trabajan bajo el estándar Zigbee IEEE 802.15.4 son los dispositivos marca Xbee (ver Figura 5), esto por la facilidad que tienen para integrarse con otros sistemas embebidos al utilizar un número mínimo de conexiones: Power (3.3V), GND y TX/RX de la UART, junto con otras conexiones recomendables como reset y sleep para ahorro de energía por lo que suelen ser usados para redes de comunicación de aplicaciones industriales[19] e incluso para sistemas Andon [17] en cualquiera de las configuraciones y topologías mencionadas anteriormente, para este proyecto se utilizará el modelo Xbee Pro S2C el cual trabaja a 2.4Ghz .



Figura 5 Dispositivo Xbee Pro. Tomada de “¿Qué es Xbee?” [fotografía], xbee.cl, 2020, <https://xbee.cl/que-es-xbee/>

1.3 Sistemas Andon

El origen de los sistemas Andon se remonta al sector manufacturero automotriz de Japón en los años setenta, en aquel momento la demanda de autos llegó a un nivel sin precedentes trayendo consigo una serie de problemas en las áreas de producción debido a los incrementos en la demanda de automóviles, obligando a las compañías a incrementar su eficiencia en el mismo tiempo y con la misma calidad. Uno de los departamentos que tuvo que enfrentarse de manera directa con tales problemas y se vio en la necesidad de implementar mejoras para reducir el tiempo de respuesta de atención a fallas fue el departamento de mantenimiento, el cual, con el objetivo de reducir los costos de producción diseñó una torre de bombillas forradas con papel de diferentes colores agrupadas de manera vertical como se muestra en la Figura 6.

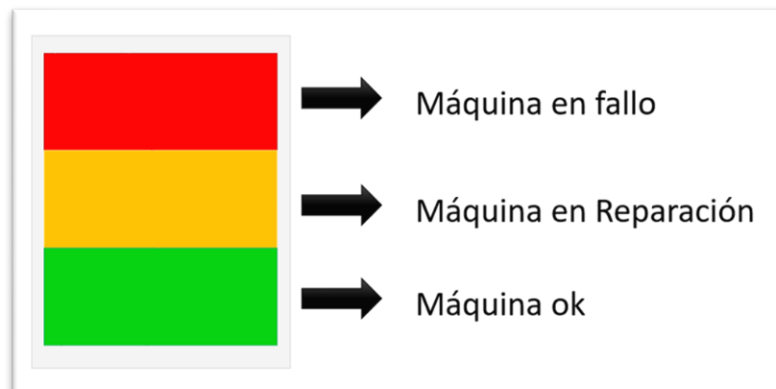


Figura 6 Interpretación de colores rojo, amarillo, verde en sus primeras fases de implementación. Fuente: elaboración propia.

Dichas torres se colocaban en cada máquina y se utilizaban para alertar a diferentes departamentos de que existía una anomalía, dependiendo de qué color se iluminaba era un llamado para un departamento en específico, las torres se colocaban en cada una de las máquinas de alta prioridad, una vez que se presentaba un fallo o era requerida la asistencia de un departamento en específico el operador activaba la bombilla correspondiente con un botón enclavado alertando de manera visual al

personal requerido logrando un tiempo de respuesta más rápido ante cualquier eventualidad.

Posteriormente el proceso fue mejorando hasta realizarse de manera automática en las máquinas cuyo nivel de automatización lo permitía. En poco tiempo su uso se extendió a todo el sector automotriz, y en unos pocos años su uso se ha extendido por todo el mundo hasta llegar a convertirse en un estándar internacional para el monitoreo de estatus operacional de equipos, llegando, al día de hoy, a integrarse con sistemas de recolección de datos de gran robustez utilizando desde bases de datos locales en archivos CSV, SQL, o MYSQL hasta la implementación en conjunto con sistemas de almacenamiento [1] y procesamiento de datos en la nube para la obtención de KPI's (Key Point Indicators) utilizados actualmente entre ellos el OEE (Overall Equipment Efficiency) [2][3] por sus siglas en inglés, el cual es uno de los indicadores más utilizados para medir la eficiencia general de una máquina.

1.4 Cálculo de OEE

En términos generales, el OEE es el indicador clave de rendimiento (KPI) del rendimiento que permite medir la eficiencia general del equipo y es reconocido por las principales industrias de todo el mundo, este índice considera los parámetros fundamentales en la producción industrial tales como *disponibilidad, productividad y calidad*. Con esto, es posible detectar equipos con baja productividad o percatarse de las causas o problemas principales de fallas y poder determinar qué necesita una máquina para alcanzar el 100% de eficiencia y a su vez saber que tanto se ha perdido debido a la falta de disponibilidad de la maquinaria debido a la baja productividad o debido a una disminución en la calidad cuando se producen unidades defectuosas. El indicador OEE es el resultado de la multiplicación de los tres indicadores o KPI's calidad, disponibilidad y productividad, a continuación, se describe en términos generales el concepto de cada uno de los KPI's más utilizados [2][3]:

1.4.1 Calidad

Este indicador representa el porcentaje total productos buenos producidos sobre el total de productos malos en un rango de 0 a 100.

Fórmula de cálculo:

$$\text{Calidad} = 100 - \left(\left(\frac{\# \text{ de piezas rechazadas}}{\# \text{ de piezas totales}} \right) * 100 \right)$$

Ejemplo de aplicación de cálculo:

$$\text{Calidad} = 100 - \left(\left(\frac{30}{223} \right) * 100 \right) = 86.54\%$$

Con esto se obtiene que la calidad de producción del ejemplo anterior fue de 86.54%, la interpretación de si es buena o mala depende de los estándares de calidad de cada empresa, no obstante, el porcentaje utópico es que sea 100%.

1.4.2 Disponibilidad

Indica la relación de tiempo que la máquina estuvo en producción entre el tiempo que se tenía planificado y se representa en el rango 0 -100%.

Fórmula de cálculo:

$$\text{Disponibilidad} = \left(\frac{\text{Tiempo total de producción real}}{\text{tiempo total disponible y/o planificado}} \right) * 100$$

Ejemplo de aplicación de cálculo:

$$\text{Disponibilidad} = \left(\frac{25585}{28800} \right) * 100 = 88.83\%$$

Del porcentaje de disponibilidad obtenido de 88.83% podemos decir que la disponibilidad que tuvo la máquina en ese periodo de tiempo fue alta, el 11.17%

faltante se debe al tiempo acumulado de paros registrados u otras razones desconocidas, igual que en el indicador anterior el porcentaje utópico es de 100%.

1.4.3 Rendimiento

Es el índice es representado por el porcentaje de cumplimiento de los objetivos de producción, se obtiene de la división del número de piezas fabricadas versus el número de piezas planeadas en el rango 0-100.

Fórmula de cálculo:

$$\text{Rendimiento} = \left(\frac{\# \text{ de piezas producidas reales}}{\# \text{ de piezas planeadas}} \right) * 100$$

Ejemplo de aplicación de cálculo:

$$\text{Rendimiento} = \left(\frac{1150}{1200} \right) * 100 = 95.83\%$$

Un porcentaje de 95.83% de rendimiento es un porcentaje relativamente bueno, quiere decir que del plan de producción se estuvo cerca de cumplir el 100%.

1.4.4 OEE (Overall Equipment Efficiency)

Es el índice de Eficiencia general de un equipo, se obtiene de la multiplicación directa de los KPI's de calidad, disponibilidad y rendimiento y está dado en valores de 0 a 100.

Fórmula de cálculo:

$$\text{OEE} = \left(\frac{\text{Calidad} * \text{Disponibilidad} * \text{Rendimiento}}{10000} \right)$$

Ejemplo de aplicación de cálculo:

$$OEE = \left(\frac{86.54 * 88.83 * 95.83}{10000} \right)$$

$$OEE = 73.66\%$$

Esto quiere decir que la eficiencia general del equipo en cuestión es del 73.66%, al conocer esto, se puede interpretar que alguno de los KPI's que lo conforman está afectando para no obtener una eficiencia más alta, con estos cuatro KPI's se puede hacer una interpretación de la situación actual de dicha máquina o interpretarse su evolución en un registro de tiempo.

De implementarse este sistema en cada línea de producción se pueden aplicar medidas correctivas para contrarrestar un porcentaje bajo de calidad, rendimiento o disponibilidad obtenidas y de esta manera incrementar el OEE, un OEE alto representa calidad, disponibilidad y rendimiento altos para una línea de producción.

Actualmente existen muchas empresas que fabrican pantallas leds para ser utilizados como visores de estos KPI's y algunas de ellas utilizan torretas industriales como indicadores visuales en campo como complemento torretas industriales de marcas como Patlite, Werma entre otras más.



% Performance	90
% Quality	100
% Availability	95
OEE	85

Figura 7 Ejemplo de contenido de una pantalla de Leds tradicional con los KPI's estándar. Fuente: elaboración propia.



Figura 8 Tablero con target vs producción actual. Fuente: elaboración propia.

El uso de tableros de led sigue siendo una práctica común en las plantas de producción del sector manufacturero principalmente el automotriz, incluso hoy cuando existen herramientas como el internet o el uso redes locales, es por eso que para este proyecto se utilizarán tableros virtuales de una plataforma web de la empresa Grupo Nauko en lugar de las pantallas de led tradicionales para mostrar los KPI's de OEE, disponibilidad, rendimiento y calidad obtenidos con las entradas digitales del prototipo de tarjeta electrónica incrementando de esta manera el alcance visual de dichos tableros hasta cualquier equipo con navegador web que este dentro de la red en la cual este el sistema instalado, el objetivo es que la empresa que implemente este equipo en conjunto con el sistema web puedan extraer de cada línea de producción dichos KPI's de forma más económica obteniendo los mismos beneficios que si utilizaran otros equipos industriales y de manera adicional las empresas puedan sustituir el uso de tableros de LED's por el uso de pantallas planas en conjunto con cualquier dispositivo embebido que soporte Google Chrome como navegador para acceder al host principal de los tableros.

1.5 Definición del problema

Actualmente el monitoreo de producción y estatus actual de máquinas o estaciones de producción se realiza por medio de la utilización de tableros de Leds con indicadores fijos [8] como se muestra en la Figura 9, indicadores como la cantidad de producción, piezas malas, porcentaje de disponibilidad de máquina y calidad entre otros. No obstante, esta tecnología no ha cambiado mucho hasta la actualidad.

Para este proyecto se plantea el desarrollo de un prototipo de una tarjeta electrónica que se pueda conectar al modelo tradicional de torretas las cuales utilizan los tres colores estándar de Andon para recopilar datos de producción y cambios de estado de las máquinas para enviarlos inalámbricamente por una red zigbee a un receptor de datos conectado a una PC por medio de cable USB para poder registrar y visualizar dichos cambios de estado de manera remota. Para este proyecto se utilizará una plataforma web desarrollada por la empresa Grupo Nauko para mostrar los datos recopilados por nuestra tarjeta prototipo la cual tiene el objetivo de reducir los costos de implementación de un sistema de este tipo sustituyendo el uso computadoras industriales también llamadas PLC cuya implementación suele ser costosa aunado a los costos de integración e infraestructura de red ethernet.



Figura 9 Ejemplo de tablero de led tradicional con indicadores de estado Andon. Tomada de “**ANDON System** [fotografía], Itek, 2020, <http://www.itek.mx/language/andon-system.html> [11].

1.6 Hipótesis

Los avances en las tecnologías de comunicación inalámbricas para la transmisión de información permitirán desarrollar un sistema embebido de bajo costo e implementación versátil para la monitorización de estatus operacional de máquinas industriales por medio de la utilización de dispositivos que trabajan bajo el estándar Zigbee IEEE 802.15.4.

1.7 Objetivo general

Para este proyecto se desarrollará un prototipo de tarjeta electrónica diseñada para facilitar el cálculo de la Eficiencia General de un Equipo OEE por sus siglas en inglés [2][3] en el contexto industrial, esta tarjeta tendrá la capacidad de detectar cambios de estado en los colores de las torretas industriales utilizadas para sistemas Andon para posteriormente enviar notificaciones de dichos cambios a través de una red inalámbrica Zigbee hacia un receptor de datos, para realizar dicho envío de datos el prototipo de tarjeta electrónica también tendrá embebido un dispositivo Xbee el cual utiliza el estándar de comunicación IEEE 802.15.4.

1.8 Objetivos específicos

- 1.- Desarrollar un prototipo de tarjeta electrónica para detectar pulsos de subida y bajada de 4 entradas digitales de 24v que utilice un dispositivo con comunicación inalámbrica con el estándar IEEE 802.15.4 Zigbee y que cumpla con los requerimientos funcionales listados en la Tabla 1 y Tabla 2 para mostrar el estado actual de las entradas en un tablero virtual.
- 2.- Dicho dispositivo deberá enviar mensajes en el formato de paquetes de bytes mostrado en la Figura 15 del capítulo 3 de este documento solicitado por la empresa Grupo Nauko a través de dos dispositivos zigbee en configuración transparente de punto a punto desde el xbee embebido en la tarjeta prototipo

al xbee del receptor de datos conectado a un equipo de cómputo por puerto serial con el sistema Enlacex de la misma empresa, este proceso se deberá realizar cada vez que haya un pulso en alto o un pulso en bajo de cualquiera de las entradas digitales.

- 3.- Utilizar una botonera y torreta industrial para la activación de los colores, para esta etapa se realizará de manera manual sin conectar directamente a un equipo.
- 4.- Mostrar en un tablero virtual del software Enlacex el estado actual de las entradas digitales de la tarjeta electrónica prototipo representadas por los colores rojo, verde y amarillo, así como los KPI's de OEE, Calidad, disponibilidad y rendimiento de los mismos datos que el software calcula con la información enviada desde la tarjeta electrónica prototipo.
- 5.- Mostrar en un tablero virtual del software Enlacex la línea de tiempo con los tres diferentes estados obtenidos de las entradas digitales de la tarjeta electrónica prototipo.
- 6.- El alcance de este proyecto se establece únicamente hasta el paso cinco (integración y pruebas) del ciclo de procesos de desarrollo de sistemas embebidos [13] propiedad de Infotec, ver Figura 10.

1.9 Justificación

Este proyecto contribuye a la reducción de los costos de implementación de un sistema de monitoreo de status operacional y producción de equipos industriales, esto por medio del remplazo de displays de LED por un sistema informático desarrollado por la empresa Nauko y el desarrollo de un prototipo de dispositivo de recolección de datos inalámbrico de bajo costo conectado a una torreta industrial para detectar los cambios de estado de los colores.

Al realizar este proyecto exitosamente se podrían reducir los costos de implementación de estos sistemas en un 83% aproximadamente haciéndolos más accesibles para un rango mayor de empresas y no solo para los grandes corporativos.

Capítulo 2

Desarrollo del proyecto

Capítulo 2 Desarrollo del proyecto

2.1 Modelo de desarrollo

Dado a que el objetivo de este proyecto es la realización de un prototipo de sistema embebido la metodología que se eligió para el desarrollo de este proyecto se basa en el ciclo de procesos de desarrollo de sistemas embebidos propiedad de Infotec [13].

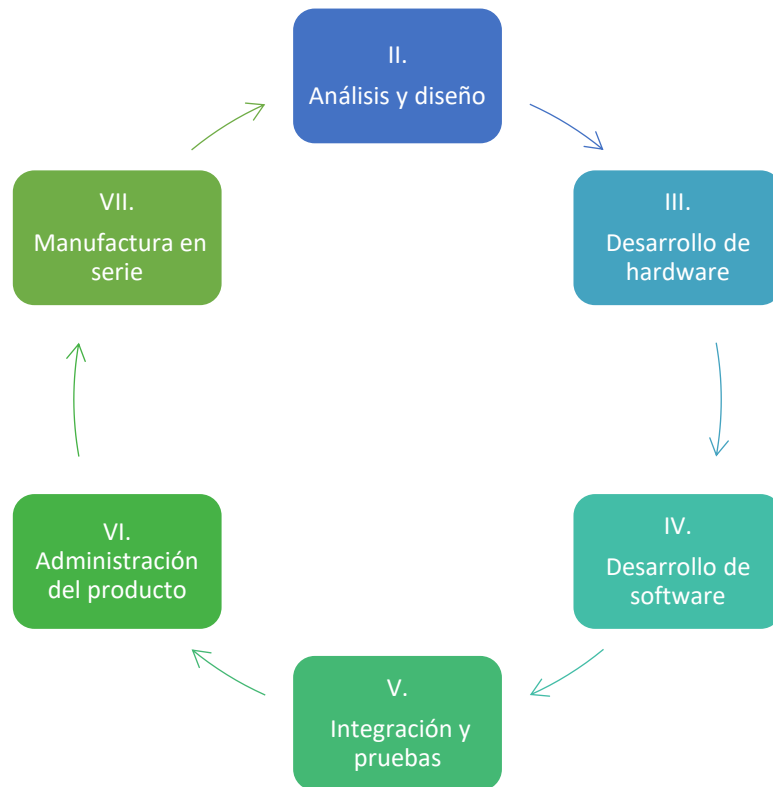


Figura 10 Ciclo de procesos de desarrollo de sistemas embebidos. Fuente: elaboración propia basada en Diagnóstico para la fundamentación de la Maestría en Sistemas Embebidos, México, D.F: INFOTEC, Feb. 2014 [13].

En dicha referencia el autor plantea siete pasos para el ciclo completo los cuales son:

- 1.- **Análisis y diseño:** En este proceso se define la lista de requerimientos funcionales y no funcionales del proyecto, así como la arquitectura y diseño.
- 2.- **Desarrollo de hardware:** Seleccionar componentes adecuados y que cumplan al menos con los estándares mínimos como que todos los componentes cumplan con la RoHS que se refiere a la *Directiva 2011/65/UE de Restricción de ciertas Sustancias Peligrosas en aparatos eléctricos y electrónicos*.
- 3.- **Desarrollo de software:** Establecer los estándares y metodología de desarrollo de software o middleware, para lo cual se utilizó el modelo de espiral de bohem limitando la iteración a un solo ciclo para este prototipo, ver Figura 11.
- 4.- **Integración y pruebas:** Ensamblar y probar el prototipo diseñado y programado para validar su funcionamiento en base los requerimientos funcionales y no funcionales Tabla 1 y Tabla 2

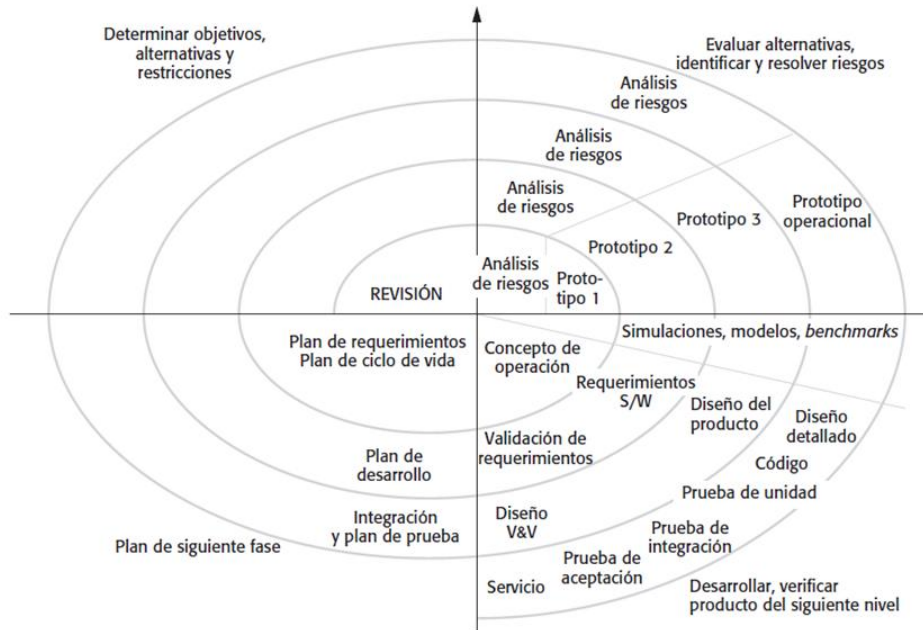


Figura 11 Modelo en espiral de Boehm. Tomada del libro de Sommerville, I. (2011). Ingeniería de software. 9 Edición. [12]

Basándonos en la Figura 11, a continuación, se describen las actividades principales a implementar en cada una de las 4 etapas del primer ciclo del Modelo en espiral de Boehm para el desarrollo del middleware del prototipo de tarjeta de este proyecto como parte del ciclo de procesos de desarrollo de sistemas embebidos [13]:

- 1.- **Establecimiento de objetivos:** en el proyecto se establecerán los objetivos específicos para el primer ciclo en el desarrollo del prototipo en base a los requerimientos funcionales listados en la Tabla 1.
- 2.- **Valoración y reducción del riesgo:** Debido a que este proyecto es un prototipo y no una versión final, se minorizan los riesgos en la primera etapa del ciclo de desarrollo a la que llega este proyecto, no obstante, es muy posible que se requiera un rediseño o cambio de componentes en un segundo ciclo sin embargo estas decisiones se tomarán al final del primer ciclo y se anotarán como conclusiones de este documento.

3.- **Desarrollo y validación:** una vez planteadas las funciones y ciclo del middleware se procederá con la programación del proyecto prototipo siguiendo el Modelo en espiral de Boehm.

4.- **Planeación:** al finalizar el primer ciclo, y por lo tanto el alcance de este proyecto, se planeará si se requiere y si es conveniente iniciar un segundo ciclo y si se opta por continuar se trazarán los planes para la siguiente fase del proyecto.

2.2 Cálculo de KPI's principales para obtención de OEE

En términos generales, el OEE es el indicador clave de rendimiento para medir la eficiencia general de un equipo y es reconocido por las principales industrias de todo el mundo, este indicador considera tres parámetros fundamentales en la producción industrial: disponibilidad, Productividad y calidad. Monitoreando la OEE es posible detectar si alguno de sus indicadores principales está causando problemas a su vez permite determinar que necesita una máquina para alcanzar el 100% de eficiencia. Todo el procesamiento y muestreo de datos lo realiza el sistema EnlaceX de Grupo Nauko y nuestra tarjeta debe ser capaz de comunicarse con su sistema y enviar la información necesaria del estatus actual de la máquina para poder procesar dicha información. Los estados que debe de monitorear son tres, representados cada uno por los colores Rojo, Amarillo y Verde, el color rojo indicará cuando la máquina está en fallo y se utilizará para calcular la disponibilidad de la máquina restando el acumulado total de tiempo en fallo al tiempo total del turno, el color amarillo será representativo del estatus de tiempo en espera de suministro de material y el verde indica el estatus OK de la máquina, la entrada digital para el conteo de piezas se utilizará para obtener el rendimiento de la máquina en referencia a la producción planeada versus la producción real, y la entrada digital de piezas rechazadas se utilizará para calcular la calidad obteniendo este dato con las fórmulas mostradas en el estado del arte para finalmente obtener la eficiencia

general de la máquina hipotética en función, a continuación se muestran la lista de requerimientos funcionales y no funcionales del dispositivo.

2.3 Especificación de requerimientos

Requerimiento	Descripción
RF1	La tarjeta deberá utilizar un módulo XBee embebido para enviar datos a la base receptora Xbee universal en modo transparente y enviar un paquete de datos cada que se detecte cualquier cambio de estado en las entradas digitales de la misma.
RF2	La tarjeta deberá tener mínimo 6 entradas con optoacopladores y barrier blocks, estas deberán tener un común para su activación.
RF3	La tarjeta deberá tener al menos 6 salidas de relevador a 1 Amper de un polo un tiro y barrier blocks para su conexión, adicional dichas salidas deberán ser alimentadas por dos comunes diferentes por lo que la tarjeta deberá tener conectores para cada uno de dichos comunes.
RF4	La tarjeta deberá enviar paquetes de información del estatus actual de cada color conectado a la torreta cada vez que exista un cambio en cualquiera de esos estados hacia el receptor de datos Xbee (base universal Xbee).
RF5	La tarjeta deberá tener los conectores y relevadores necesarios para activar las lámparas de colores de una torreta industrial de 3 o 5 estados estándar, cada relevador deberá soportar una carga máxima de .5 amperes.
RF6	La tarjeta deberá funcionar con una entrada de 5v y tener un conector tipo Jack para recibir dicha alimentación por medio de un eliminador.
RF7	La tarjeta deberá utilizar un microcontrolador Atmega2560 como unidad principal de procesamiento.

Tabla 1 Requerimientos funcionales. Fuente: elaboración propia.

Requerimiento	Descripción
RNF1	El sistema debe tener una disponibilidad de 99.99%.
RNF2	El equipo debe usar el protocolo de comunicación inalámbrica 802.15.4.

Tabla 2 Requerimientos no funcionales. Fuente: elaboración propia.

2.4 Elección de tecnologías

Tomando como base los requerimientos se realiza una búsqueda y selección de tecnologías para el ensamble del prototipo. A continuación, se muestra el diseño realizado en la plataforma gratuita de diseño electrónico Eagle con la lista de componentes seleccionados:

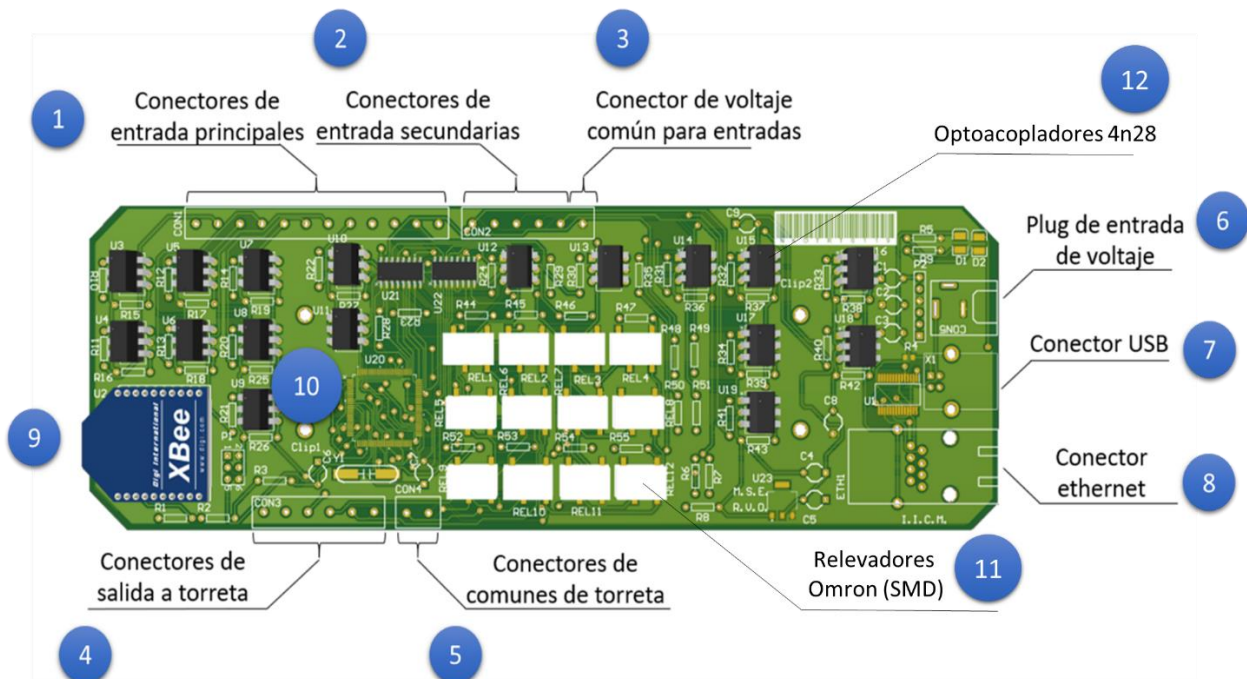


Figura 12 Diseño de primer prototipo de tarjeta electrónica. Fuente: elaboración propia.

A continuación, se describe cada uno de los componentes principales indicando al principio que requerimiento funcional se está cubriendo con cada uno de ellos:

- 1.- **Conectores de entrada principales (RF2):** estos 12 conectores tipo barrier block van directamente al clúster de optoacopladores para recibir entradas digitales de 24v alimentadas por la entrada común del punto 3, al ser 12 la

cantidad cumple con el requerimiento inicial de 6 entradas digitales, para ver el diagrama de conexión ir al anexo C (Diagrama de entradas digitales).

- 2.- **Conectores de entrada secundarios:** estos son auxiliares para posibles actualizaciones o modificaciones posteriores.
- 3.- **Conector de voltaje común para entradas digitales (RF2):** se decidió utilizar común externo para proteger de esta manera el circuito interno junto con los optoacopladores.
- 4.- **Conectores de salida a torreta (RF5 y RF3):** estas salidas son utilizadas para encender los pilotos verde, amarillo o rojo de las torretas industriales dependiendo de qué entrada digital se active o desactive, para ver el diagrama de conexión ir al Anexo B - **Diagrama de salidas digitales.**
- 5.- **Conectores de comunes de torreta (RF3 y RF5):** la tarjeta se diseñó para tomar en cuenta dos comunes diferentes para la activación de los pilotos de las torretas, esto debido a que en el mercado actual existen varias marcas como Patlite y Werma que utilizan dos comunes para la activación de sus torretas, uno lo utilizan para encender en modo estático y otro en modo parpadeante los pilotos.
- 6.- **Plug de entrada de voltaje (RF6):** para alimentar la tarjeta con un eliminador de 5v.
- 7.- **Conector USB:** reservado para aplicaciones futuras.
- 8.- **Modulo ethernet xpico integrado:** reservado para aplicaciones futuras.
- 9.- **Modulo Xbee Pro S2C a 2.4Ghz (RF1 y RF4):** utilizado para enviar los paquetes de datos inalámbricamente a la base universal de Xbee a una

distancia de al menos 200 metros dentro de planta, para ver el diagrama de conexión ir al Anexo D - **Diagrama de comunicaciones**.

- 10.- **Microcontrolador Atmega2560(RF7)**: utilizado para detectar los pulsos en alto y bajo de los 4 colores la señal de conteo y enviar los paquetes de datos por el puerto serial hacia el Xbee.
- 11.- **Relevadores Omron GL6-1F 5VDC**: Se seleccionaron los modelos de montaje superficial para reducir el tamaño de la tarjeta y para cumplir con RF5, dichos relevadores soportan por especificación una carga máxima de 1 ampere lo cual es el doble de la requerida en la Tabla 1 de .5 Ampere.
- 12.- **Optoacopladores 4n28**: Estos optoacopladores se seleccionaron por indicación de la empresa Grupo Nauko basados en la facilidad de acceso y cercanía de proveedores, dichos optoacopladores se utilizaron para aislar las entradas de 24v del circuito interno a 5v para entradas digitales del microcontrolador utilizando una resistencia de 1500 ohms a ½ watt en el circuito de entrada de 24v y una de 10k a ½ watt ohms en el circuito de entrada al microcontrolador en configuración pulldown para evitar los falsos cambios de estado en las entradas digitales al microcontrolador.

En la Figura 13 se muestra la tarjeta prototipo con los componentes ensamblados en base los diagramas Anexo A - Diagrama principal, Anexo B - Diagrama de salidas digitales, Anexo C - Diagrama de entradas digitales y Anexo D - Diagrama de comunicaciones basados en los requerimientos funcionales y no funcionales de la Tabla 1 y Tabla 2:

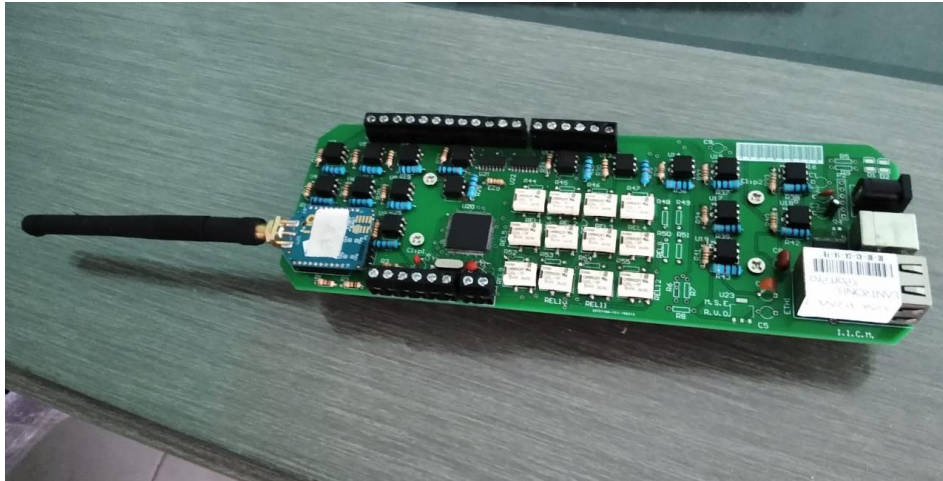


Figura 13 Tarjeta prototipo con componentes ensamblados. Fuente: elaboración propia.

2.5 Arquitectura general de comunicación

Como se menciona en un inicio en este documento, la tarjeta electrónica deberá enviar por medio del protocolo de comunicación inalámbrica IEEE 802.15.4 la información actual del estado de la torreta hacia el receptor universal Xbee ambos en configuración transparente y utilizando la topología estrella, el paquete de datos se envía acorde al formato del buffer especificado por la empresa Grupo Nauko ver Figura 15, a continuación, se muestra una imagen con la arquitectura general de comunicación:

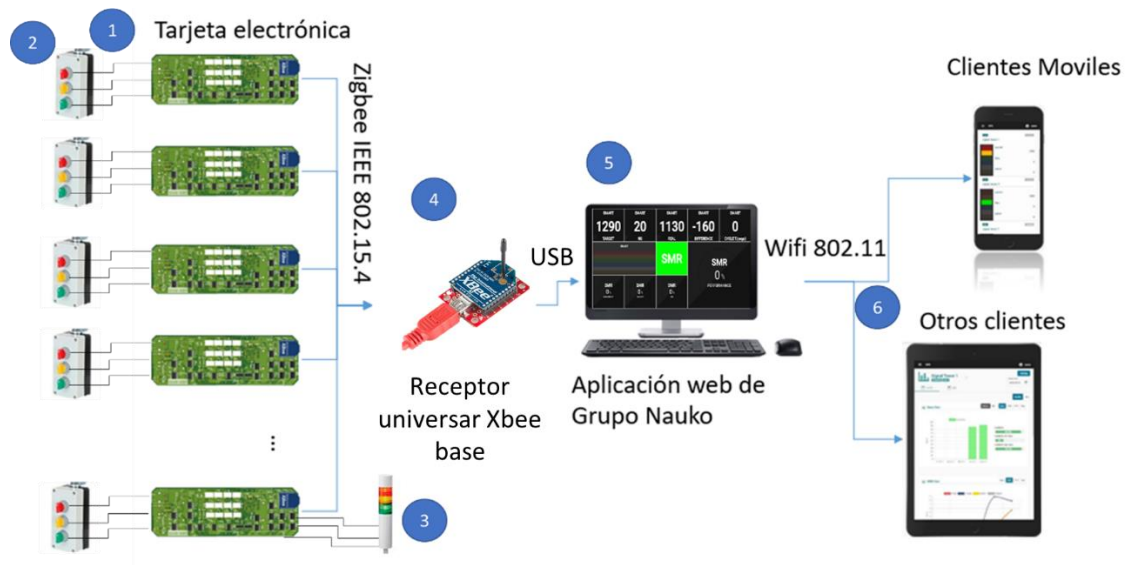


Figura 14 Diagrama arquitectónico de comunicación de la solución propuesta. Fuente: elaboración propia.

En la Figura 14 se muestra el diagrama de comunicación que va de izquierda a derecha desde la recolección de datos de las entradas digitales en la tarjeta hasta su proyección en tableros virtuales de la plataforma web de Grupo Nauko.

- 1.-Prototipo de Tarjeta electrónica desarrollada.
- 2.-Selector de colores conectado a las entradas digitales de la tarjeta prototipo.
- 3.-Torreta con 3 pilotos(colores) que se activa con las salidas a relevador de la tarjeta prototipo.
- 4.-Receptor de datos Xbee utilizado para recibir los paquetes de datos de la tarjeta prototipo recibidos por protocolo IEEE 802.15.4 desde el xbee embebido en la tarjeta prototipo.
- 5.-Plataforma Enlacex de Grupo Nauko utilizada para visualizar los datos recolectados con el prototipo.
- 6.-Dispositivos con acceso a la plataforma

2.6 Formato de paquetes de datos

Cada paquete de datos transmitido tendrá el tamaño fijo de 8 bytes y será transmitido cada que se detecte un pulso en alto o un pulso en bajo en cualquiera de las entradas digitales de la tarjeta, el formato se muestra en la siguiente figura:

Paquete de buffer de datos enviado							
byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8
byte llave de inicio (valor fijo)	ID del dispositivo	status de color rojo	status de color amarillo	status de color verde	status de pin de conteo de piezas	status de pin de conteo de piezas defectuosas	byte llave de fin (valor fijo)
253	1-251	0-1	0-1	0-1	0-1	0-1	252

Figura 15 Descripción de contenido de paquete de datos transmitido. Fuente: elaboración propia.

Para ver el código utilizado en la recolección y transmisión de datos ver Anexo E - Firmware de microcontrolador.

2.7 Diagrama de flujo del proceso de recolección de datos.

Para la recolección de datos de estado de entradas digitales se utilizan funciones de detección de cambio de estado en cada entrada digital en el firmware cargado al microcontrolador (ver Anexo E - Firmware de microcontrolador, función “monitorDeEntrada1”), para posteriormente ser enviados como un array por medio del puerto serial hacia el dispositivo XBee.

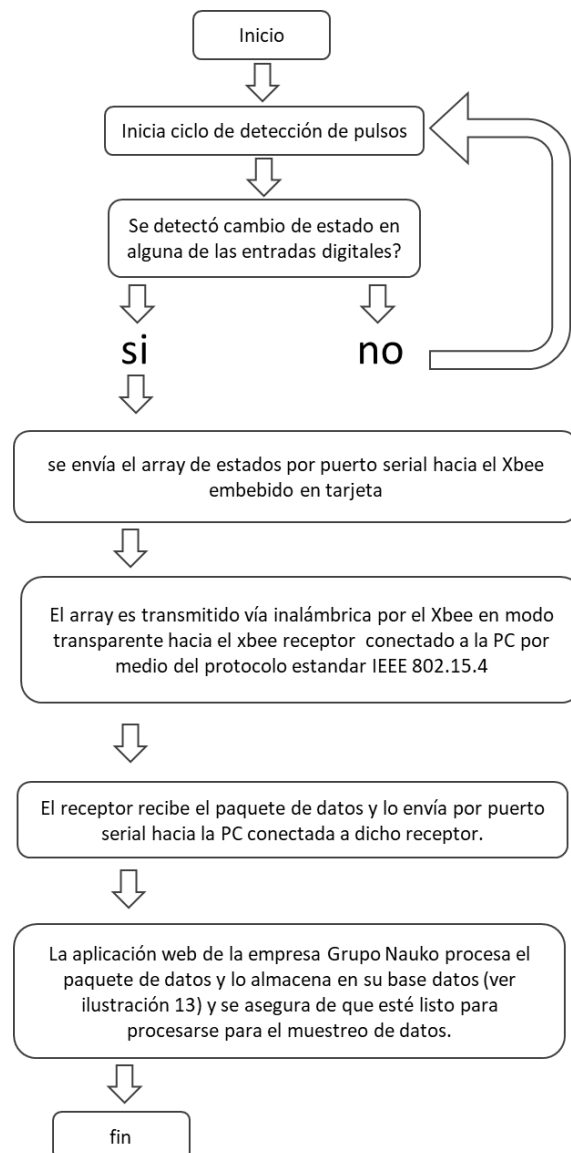
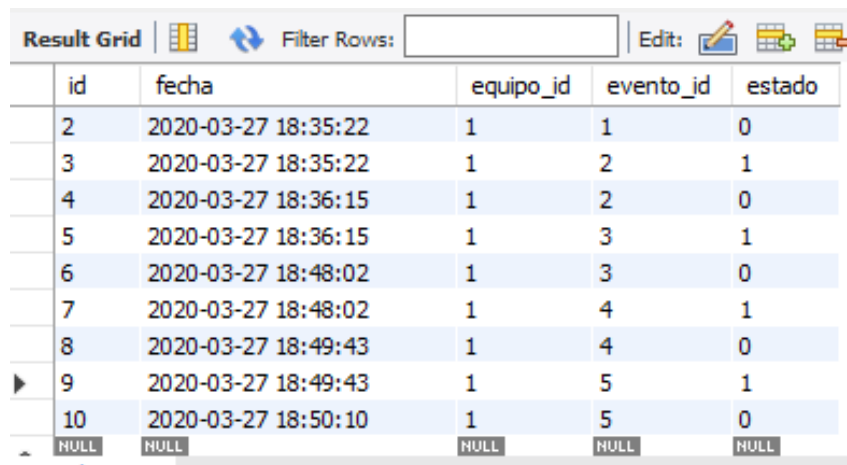


Figura 16 Ciclo de recolección de datos. Fuente: elaboración propia.

Una vez almacenada la información en la PC esta puede ser procesada y mostrada de diferentes maneras por ejemplo en la Figura 26, Figura 27, Figura 28 y Figura 29 de resultados extraídos de plataforma EnlaceX, información como la línea de tiempo del estado operacional en un determinado periodo de tiempo o los indicadores de OEE, disponibilidad, calidad y rendimiento.

2.8 interpretación de datos en el lado de la aplicación web.

2.8.1 Registro de eventos en base de datos de MYSQL



id	fecha	equipo_id	evento_id	estado
2	2020-03-27 18:35:22	1	1	0
3	2020-03-27 18:35:22	1	2	1
4	2020-03-27 18:36:15	1	2	0
5	2020-03-27 18:36:15	1	3	1
6	2020-03-27 18:48:02	1	3	0
7	2020-03-27 18:48:02	1	4	1
8	2020-03-27 18:49:43	1	4	0
9	2020-03-27 18:49:43	1	5	1
10	2020-03-27 18:50:10	1	5	0
NULL	NULL	NULL	NULL	NULL

Figura 17 Registro de eventos. Fuente: elaboración propia.

En la Figura 17 se puede apreciar un registro de los eventos recibidos del prototipo de tarjeta en cuestión, cada entrada de registro viene fechada con la hora y día que se recibió el paquete de datos, el id del equipo(equipo_id) del que corresponde a esa entrada, el id de evento(evento_id) y el estado de la entrada digital de la tarjeta en ese momento, donde el estado 1 corresponde a encendido o un 1 lógico, y el estado 0 corresponde a apagado o a un 0 lógico.

2.8.2 Interpretación de registros

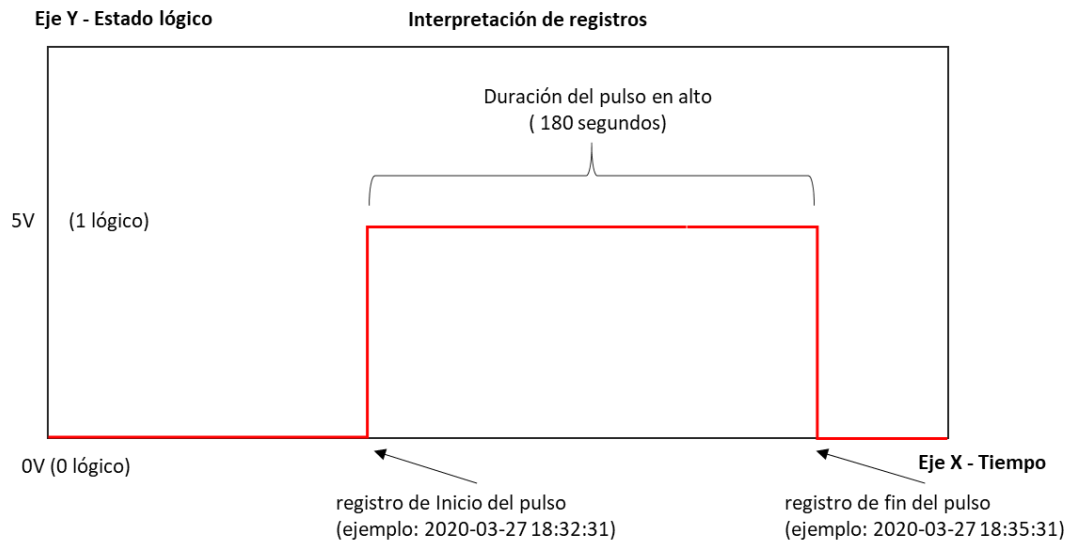


Figura 18 Ejemplo de interpretación de un pulso digital. Fuente: elaboración propia.

En la Figura 18 se muestra cómo se interpretan los registros con el id 1 e id 2 mostrados en la Figura 17, para interpretar un pulso con inicio y fin se requieren dos registros, uno de inicio y otro de fin.

2.8.3 Interpretación de una línea de tiempo en base a los registros

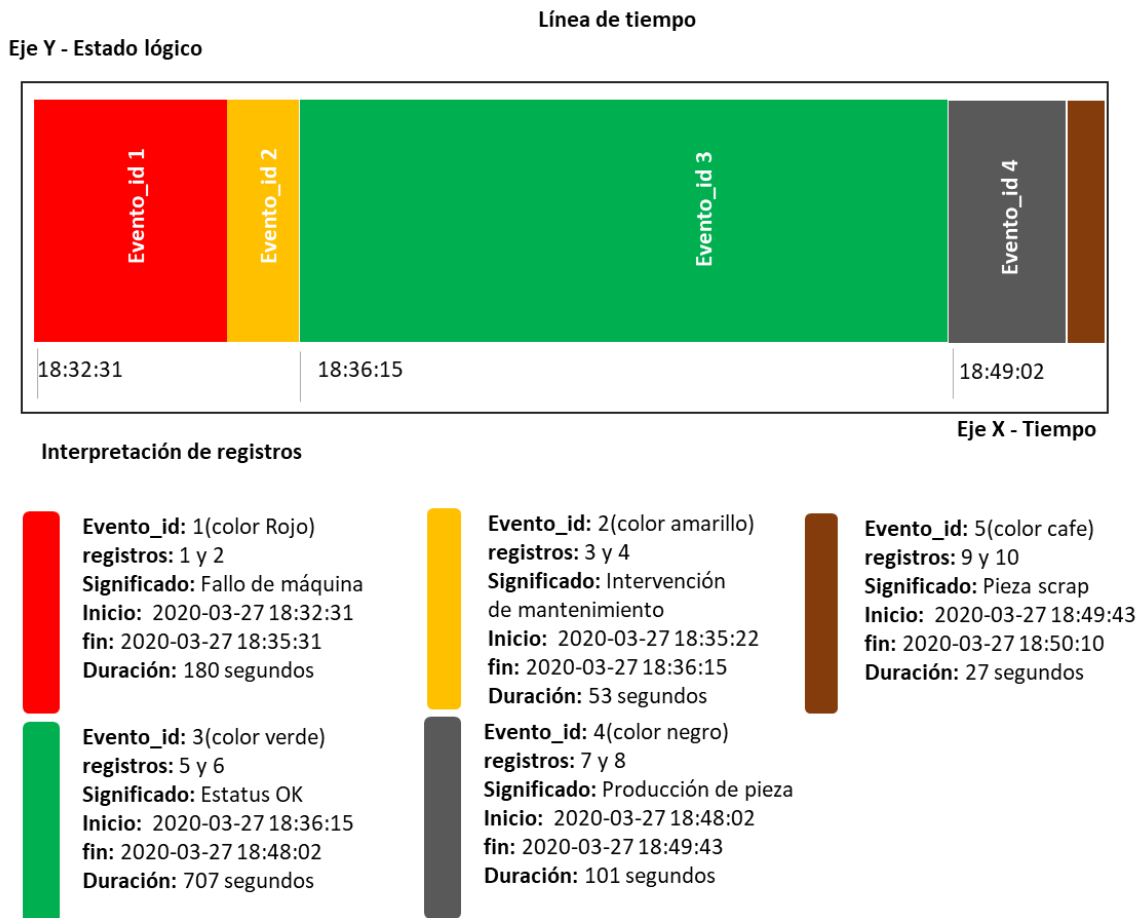


Figura 19 Interpretación de registros en una línea de tiempo. Fuente: elaboración propia.

En la Figura 19 se aprecia un ejemplo de cómo se interpretan los registros de la Figura 17 en una línea de tiempo, tomando dos registros por cada evento.

2.9 Obtención de eficiencia general

Como se explica en el estado del arte, para la obtención de la eficiencia general de una máquina (OEE por sus siglas en inglés) se requiere calcular previamente la disponibilidad, rendimiento y calidad de las piezas producidas, a continuación, se describe como se interpretan los registros de la base de datos en MYSQL obtenidos de los registros de la tarjeta prototipo.

2.9.1 Obtención de Disponibilidad de la máquina

Gráfica de eventos de partes producidas(evento_id 4)

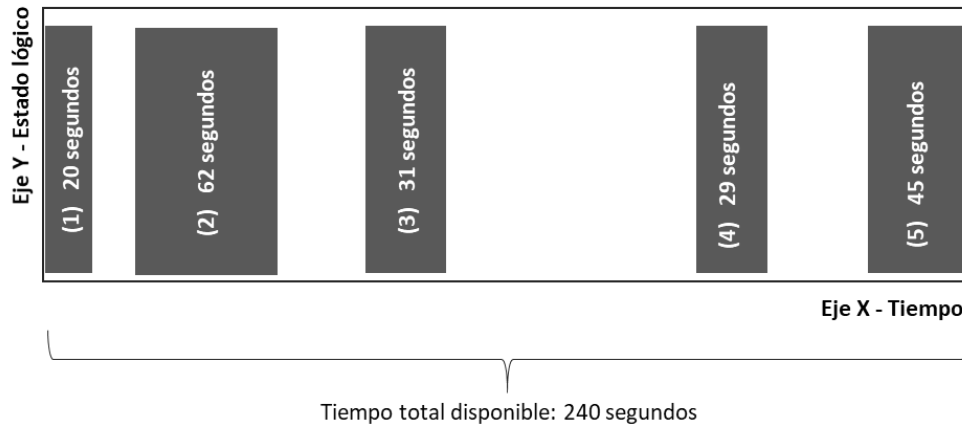


Figura 20 Ejemplo de interpretación de registros de eventos de partes producidas. Fuente: elaboración propia.

En la Figura 20 observamos un ejemplo de 5 eventos de la señal que se toma como inicio y fin de ciclo de producción de partes (evento_id: 4) cada uno con duraciones y tiempos de inicio y fin diferentes, a continuación, se muestra cómo se obtiene el rendimiento con estos 5 registros:

Nota: Debido a que el tiempo disponible es relativo a como se seleccionen los filtros para la toma de muestras de eventos en la base de datos tomaremos un tiempo disponible de 240 segundos para efectos demostrativos.

Fórmula:

$$\text{Disponibilidad} = \left(\frac{\text{Tiempo total de producción real}}{\text{tiempo total disponible y/o planificado}} \right) * 100$$

Cálculo paso 1:

$$\text{Disponibilidad} = \left(\frac{(20 + 62 + 31 + 29 + 45)}{240} \right) * 100$$

Cálculo paso 2:

$$\text{Disponibilidad} = \left(\frac{187}{240}\right) * 100$$

Cálculo paso 3:

$$\text{Disponibilidad} = 77.5\%$$

De esta manera obtenemos el cálculo de disponibilidad con la señal digital 4 la cual se utiliza para medir el inicio y fin de ciclo de producción de una pieza cualquiera que sea su naturaleza.

2.9.2 Obtención de rendimiento de la máquina

Reutilizando el ejemplo de la Figura 20 en la que se muestran 5 eventos de piezas producidas tomaremos la variable de piezas producidas reales como 5, y debido a que la variable de piezas planeadas es introducida en el sistema de Enlacex por el usuario final para este ejemplo la fijaremos en 7:

Fórmula:

$$\text{Rendimiento} = \left(\frac{\# \text{ de piezas producidas reales}}{\# \text{ de piezas planeadas}}\right) * 100$$

Cálculo paso 1:

$$\text{Rendimiento} = \left(\frac{5}{7}\right) * 100$$

Cálculo paso 2:

$$\text{Rendimiento} = 71.42\%$$

2.9.3 Obtención de calidad de las piezas producidas

Tomando como ejemplo para este cálculo los registros de la Figura 17 en donde se obtienen en resumen 5 eventos de piezas producidas con fechas de inicio y fin correspondientes y 1 evento de pieza defectuosa realizamos el siguiente cálculo para obtener el parámetro deseado de calidad:

Fórmula:

$$\text{Calidad} = 100 - \left(\left(\frac{\# \text{ de piezas rechazadas}}{\# \text{ de piezas totales}} \right) * 100 \right)$$

Cálculo paso 1:

$$\text{Calidad} = 100 - \left(\left(\frac{1}{5} \right) * 100 \right)$$

Cálculo paso 2:

$$\text{Calidad} = 80\%$$

2.9.4 Obtención de eficiencia general de la máquina

Una vez obtenidos los tres parámetros necesarios para el cálculo de la eficiencia general aplicamos la fórmula mostrada inicialmente.

Fórmula:

$$\text{OEE} = \left(\frac{\text{Disponibilidad} \times \text{Rendimiento} \times \text{Calidad}}{10000} \right)$$

Cálculo paso 1:

$$\text{OEE} = \left(\frac{77.5 \times 71.42 \times 80}{10000} \right)$$

Cálculo paso 2:

$$\text{OEE} = 44.28\%$$

2.9.5 Análisis de OEE obtenida

Tomando como ejemplo la OEE de 44.28% obtenida del ejemplo anterior podemos deducir que la máquina no está trabajando al 100% de su capacidad, de este resultado podemos interpretar lo siguiente:

- 1.- Una disponibilidad que no está a al máximo no necesariamente es mala, una disponibilidad de 77.5% quiere decir que dicha máquina aún tiene una capacidad de incrementar producción en un 22.5% por lo que para los dueños de esta máquina ficticia es una buena noticia.
- 2.- Un rendimiento de 71.42% sí es un indicador de que algo anda mal ya que no se cumplió el objetivo de producción planeado, para esto es necesario revisar las causas por las cuales no se llegó al objetivo y corregirlas.
- 3.- Una calidad de 80% es igualmente un resultado malo, aunque la muestra fue pequeña (6 piezas), nos dice que 1 de cada 6 piezas ha resultado con algún defecto de calidad.

Cabe mencionar que medir de manera automática una pieza de mala calidad no es muy común ya que automatizar su detección conlleva grandes inversiones de dinero ya sea en cámaras de visión o sensores de alta precisión por lo que la mayoría de las veces es una persona del departamento de calidad la responsable de determinar cuándo una pieza es de mala calidad y se debe desechar o retrabajar.

Para interpretar la OEE es necesario analizar también sus KPI's para entender las causas del estado actual de una máquina y así poder determinar con más certeza las causas de una eficiencia baja.

Capítulo 3

Pruebas, resultados y validación

Capítulo 3 Pruebas, resultados y validación

3.1 Pruebas y resultados de la solución tecnológica propuesta

3.1.1 Primer prueba con botón simple

En la Figura 21 se muestra el gabinete con la tarjeta y una torreta estándar de 24v conectada a las salidas de relevador de la tarjeta y un solo botón para la prueba uno:



Figura 21 Primer prueba con botón simple. Fuente: elaboración propia.

3.1.2 Segunda prueba con botonera completa

En la Figura 22 se muestra el gabinete con la tarjeta y una torreta estándar de 24v conectada a las salidas de relevador de la tarjeta y una botonera con selectores conectados a las entradas digitales por medio de un eliminador de 24v simulando el voltaje de salida de un PLC con salidas a 24v:

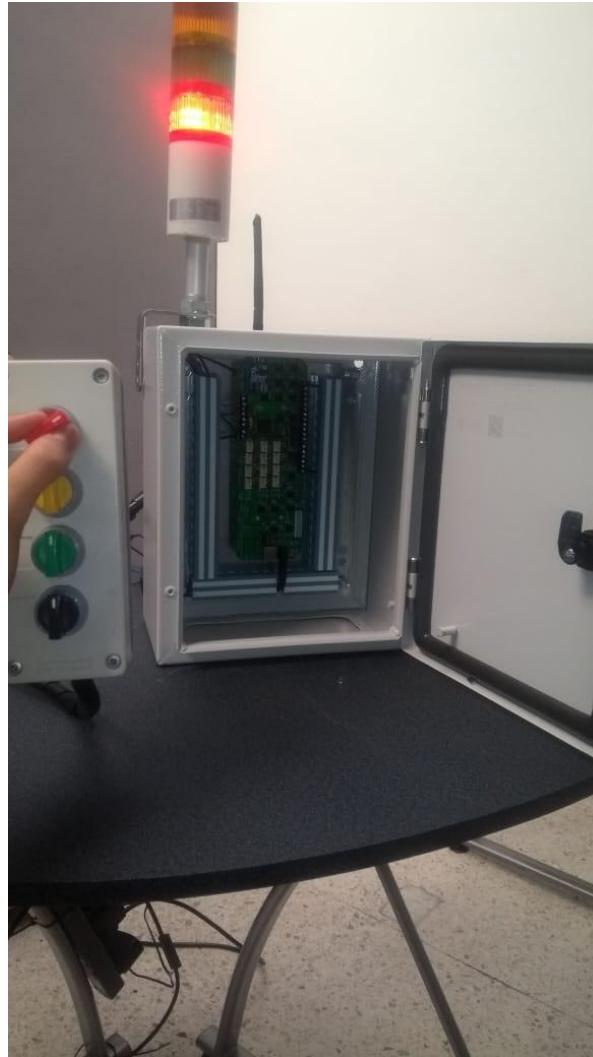


Figura 22 Ensamble con botonera de 4 selectores, cada selector cuenta con dos estados sumando en total 8 estados posibles. Fuente: elaboración propia.

En la Figura 23 se muestra la Captura de pantalla de tablero virtual de aplicación web Enlacex visualizado en un navegador Chrome en donde se muestra el estado actual de la entrada digital de la tarjeta prototipo en color rojo encendido (fallo), en la misma figura se aprecian diferentes tipos de controles con información adicional, como el número de piezas producidas o el tiempo que el equipo lleva en el estado de color rojo (fallo) indicado con un nombre corto “P1”.

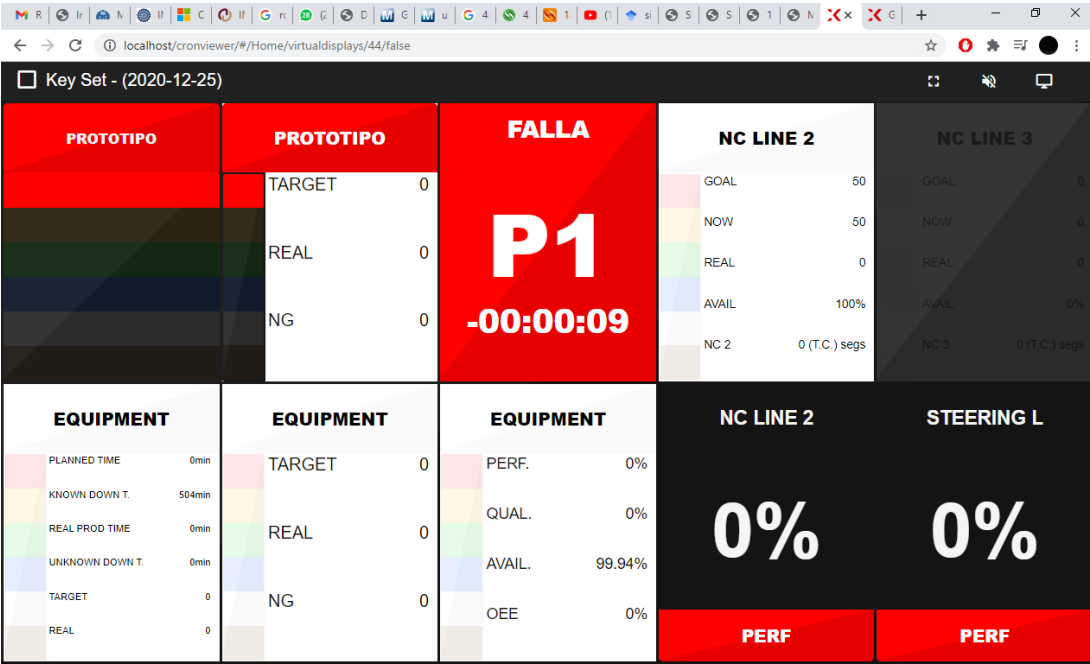


Figura 23 Captura de pantalla de tablero virtual de aplicación web Enlacex visualizado en un navegador Chrome. Fuente: elaboración propia.

De la Figura 23 se desprende una de las motivaciones originales que era utilizar la tarjeta prototipo como medio para visualizar el estado actual de una máquina en un navegador web.

3.1.3 Receptor Xbee (Xbee base)

En la Figura 24 se muestra el receptor de Xbee utilizado, el cual es una base universal para Xbee que convierte de la información recibida a protocolo serial para procesarse por medio de un adaptador virtual de entrada USB.



Figura 24 Base receptor Xbee Universal. Fuente: elaboración propia.

3.1.4 Prueba de recepción de datos

En la Figura 25 se muestra el receptor de Xbee utilizado conectado a la PC en la cual corre el sistema Enlacex, para la muestra de datos se activaron los selectores de la botonera de manera intermitente desde las 6 de la mañana hasta las 4 de la tarde:

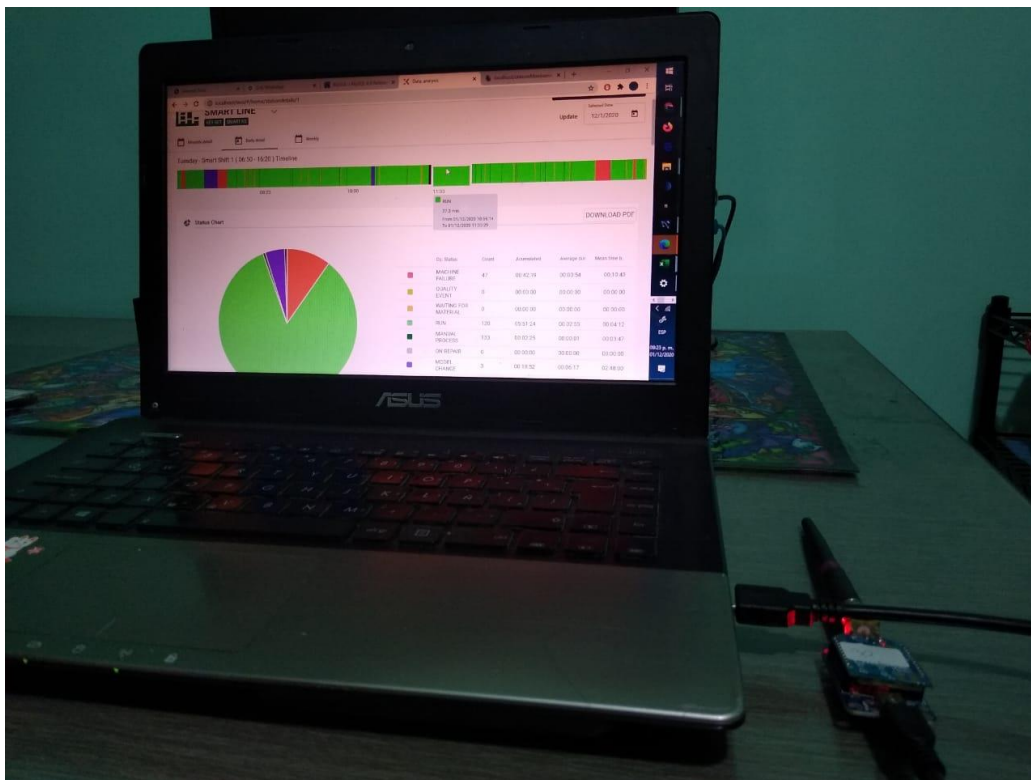


Figura 25 Primer prueba real de recolección de datos (Interpretación de datos utilizando software Enlacex).
Fuente: elaboración propia.

3.1.5 Resultados extraídos de plataforma Enlaxex

En la Figura 26 se muestra una línea de tiempo con 8 señales habilitadas y cada una con una interpretación diferente, tomando la señal del color azul como contador de piezas y el buzzer como contador de piezas defectuosas:

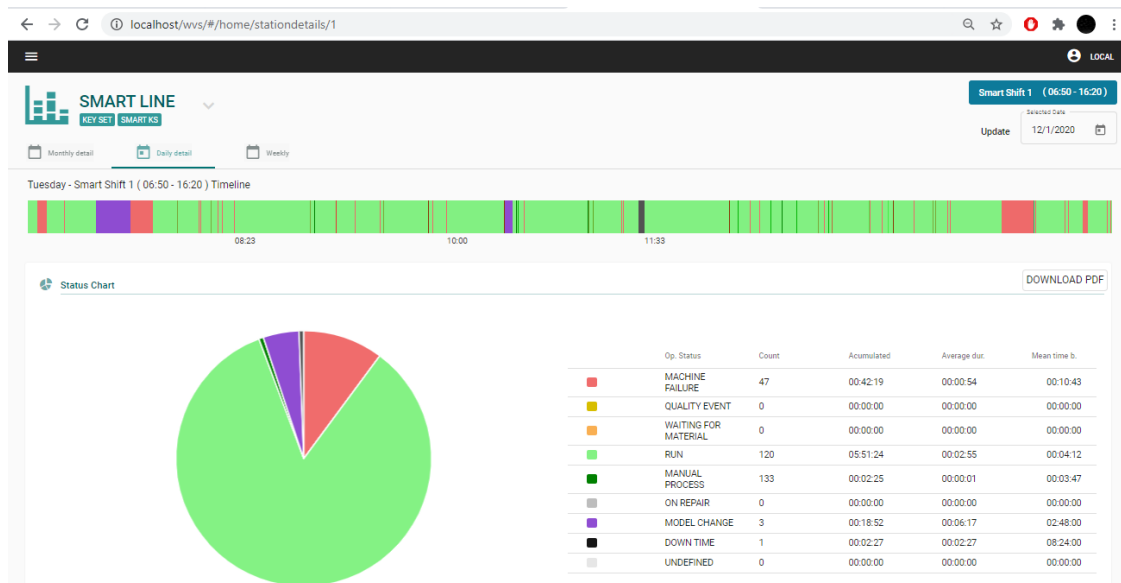


Figura 26 Línea de tiempo y gráfica de pastel con tiempos acumulados por estado operacional (Interpretación de datos utilizando software Enlaxex). Fuente: elaboración propia.

En las figuras siguientes se muestran aún más funciones y datos obtenidos con los mismos registros simples como los de la Figura 17, como la gráfica de pastel de la Figura 26, los KPI's mostrados en la Figura 27, la gráfica de producción por hora de la Figura 28 y la visualización de los registros de la Figura 29; no obstante, cabe recalcar que el objetivo del proyecto no es desarrollar la plataforma dado que parte de los objetivos específicos es conectarse a la plataforma Enlaxex con el prototipo de tarjeta para buscar alternativas a PLC más económicas para la implementación de estos sistemas.

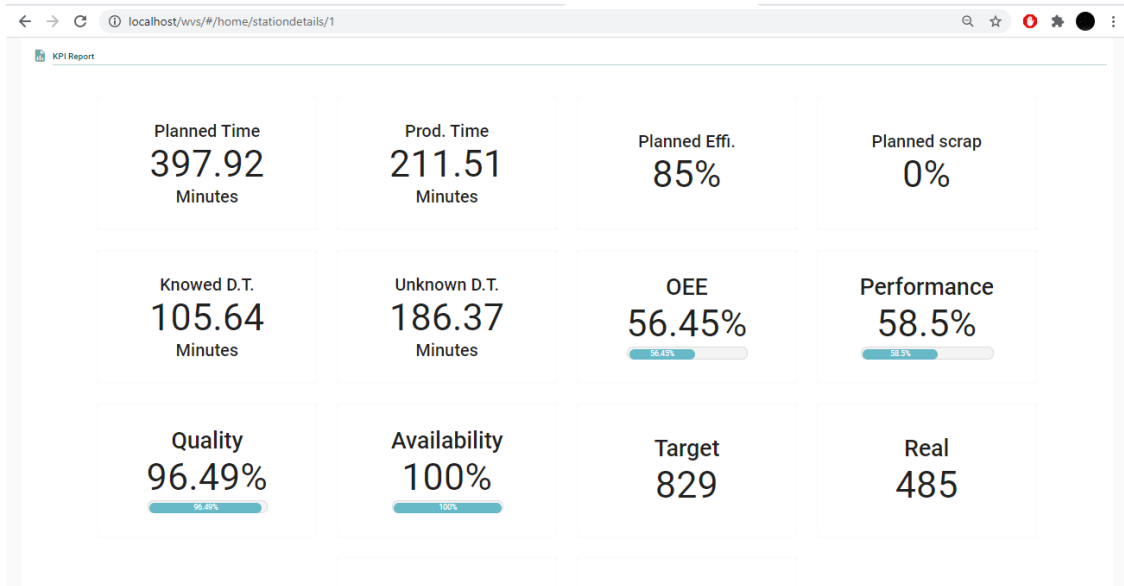


Figura 27 Reporte de resultados obtenidos de la prueba (Interpretación de datos utilizando software Enlaxex). Fuente: elaboración propia.

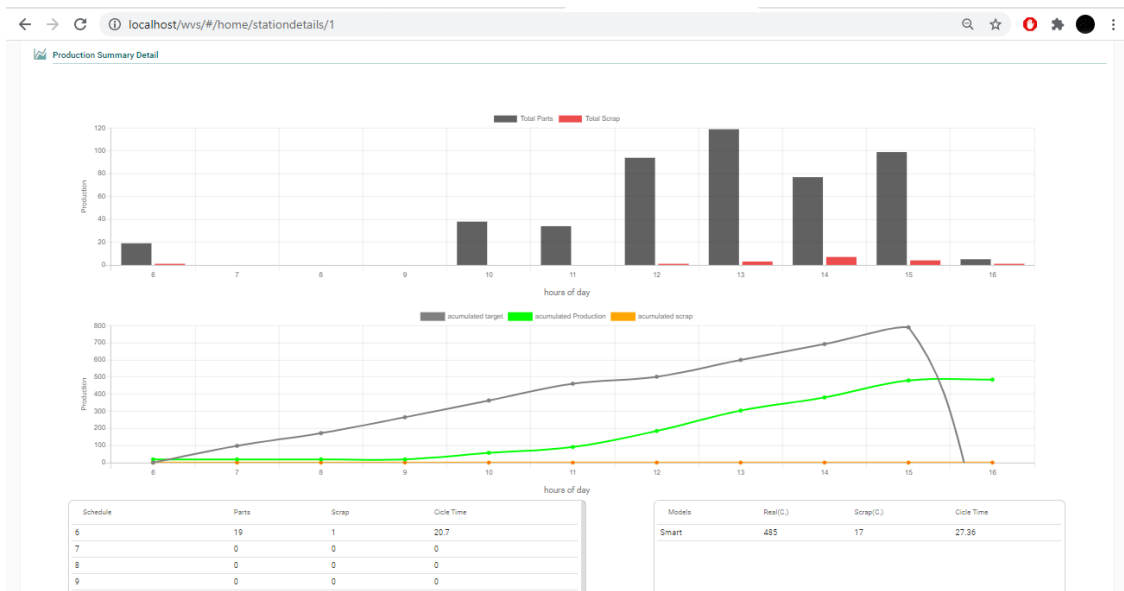


Figura 28 Gráfica de piezas producidas y piezas defectuosas por hora (Interpretación de datos utilizando software Enlaxex). Fuente: elaboración propia.

Event	Category	Start	End	Steps	User	Comments
RUN		18/11/2020 06:50:05	18/11/2020 06:52:28	142		
MANUAL PROCESS		18/11/2020 06:52:28	18/11/2020 06:52:28	0		
MACHINE FAILURE		18/11/2020 06:52:26	18/11/2020 06:52:38	11		
RUN		18/11/2020 06:52:28	18/11/2020 06:54:21	112		
MANUAL PROCESS		18/11/2020 06:54:21	18/11/2020 06:54:21	0		
MACHINE FAILURE		18/11/2020 06:54:20	18/11/2020 06:54:41	20		
RUN		18/11/2020 06:54:21	18/11/2020 07:17:03	1361		
MANUAL PROCESS		18/11/2020 07:17:03	18/11/2020 07:17:04	0		
QUALITY EVENT		18/11/2020 10:19:35	18/11/2020 10:26:29	413		
ON REPAIR		18/11/2020 07:10:47	18/11/2020 10:26:30	11743		
MODEL CHANGE		18/11/2020 10:26:30	18/11/2020 10:31:14	284		
MACHINE FAILURE	Operador no capacitado 弊 害発けない	18/11/2020 07:16:59	18/11/2020 10:31:15	11656	Admin	Rod joint de otro modelo, operador no cambio el jck de la maquina ni el rod joint 5 piezas NG
MANUAL PROCESS		18/11/2020 10:53:11	18/11/2020 10:53:12	0		
RUN		18/11/2020 07:17:04	18/11/2020 10:53:11	12947		
RUN		18/11/2020 10:53:12	18/11/2020 11:34:39	2486		
MANUAL PROCESS		18/11/2020 11:34:39	18/11/2020 11:34:39	0		
RUN		18/11/2020 11:34:39	18/11/2020 11:44:50	610		
MANUAL PROCESS		18/11/2020 11:44:50	18/11/2020 11:44:51	0		
RUN		18/11/2020 11:44:51	18/11/2020 11:53:00	489		
MANUAL PROCESS		18/11/2020 11:53:00	18/11/2020 11:53:01	0		
MANUAL PROCESS		18/11/2020 12:03:46	18/11/2020 12:03:47	0		
RUN		18/11/2020 11:53:01	18/11/2020 12:03:46	645		
MODEL CHANGE		18/11/2020 12:20:00	18/11/2020 12:20:22	22		
ON REPAIR		18/11/2020 12:20:22	18/11/2020 12:22:52	150		

Figura 29 Visualización de registro de eventos (Interpretación de datos utilizando software Enlაცex). Fuente: elaboración propia.

3.2 Validación de la solución tecnológica propuesta

3.2.1 Validación de requerimientos funcionales:

A continuación, se muestra la Tabla 3 de validación de requerimientos funcionales listados en la Tabla 1, en la cual se muestra el cumplimiento de todos los requerimientos funcionales:

Requerimiento	Descripción	Estado
RF1	La tarjeta deberá utilizar un módulo XBee embebido para enviar datos a la base receptora Xbee universal en modo transparente y enviar un paquete de datos cada que se detecte cualquier cambio de estado en las entradas digitales de la misma.	Cumplido
RF2	La tarjeta deberá tener mínimo 6 entradas con optoacopladores y barrier blocks, estas deberán tener un común para su activación.	Cumplido
RF3	La tarjeta deberá tener al menos 6 salidas de relevador a 1 Amper de un polo un tiro y barrier blocks para su conexión, adicional dichas salidas deberán ser alimentadas por dos comunes diferentes por lo que la tarjeta deberá tener conectores para cada uno de dichos comunes.	Cumplido
RF4	La tarjeta deberá enviar paquetes de información del estatus actual de cada color conectado a la torreta cada	Cumplido

	vez que exista un cambio en cualquiera de esos estados hacia el receptor de datos Xbee (base universal Xbee).	
RF5	La tarjeta deberá tener los conectores y relevadores necesarios para activar las lámparas de colores de una torreta industrial de 3 o 5 estados estándar, cada relevador deberá soportar una carga máxima de .5 amperes.	Cumplido
RF6	La tarjeta deberá funcionar con una entrada de 5v y tener un conector tipo Jack para recibir dicha alimentación por medio de un eliminador.	Cumplido
RF7	La tarjeta deberá utilizar un microcontrolador Atmega2560 como unidad principal de procesamiento.	Cumplido

Tabla 3 Validación de requerimientos funcionales. Fuente: elaboración propia.

3.2.2 Validación de requerimientos no funcionales:

A continuación, se muestra la Tabla 4 de validación de requerimientos no funcionales listados en la Tabla 2, en la cual se muestra el cumplimiento de todos los requerimientos no funcionales resultado de la selección de componentes en el diseño y desarrollo del proyecto:

Requerimiento	Descripción	Estado
RNF1	El sistema debe tener una disponibilidad de 99.99%.	Cumplido
RNF2	El equipo debe usar el protocolo de comunicación inalámbrica 802.15.4.	Cumplido

Tabla 4 Validación de requerimientos no funcionales. Fuente: elaboración propia.

3.3 Mejoras y puntos de oportunidad detectados

- 1.- En cuanto a la interfaz de comunicación con la PC: se sugiere que en la segunda etapa se utilice un Gateway industrial o en todo caso se desarrolle un receptor con convertidor serial a ethernet para poder enviar los datos directamente a un modem o switch y de ahí incrementar su alcance a una red local de ethernet o en todo caso a un servidor DNS.
- 2.- En cuanto a el establecimiento de un canal adecuado en la red Zigbee: dado a que Zigbee trabaja al igual que Wifi a 2.4Ghz, es necesario establecer el canal de comunicación de la red Zigbee en canales que no estén en uso por la red wifi o en todo caso que tengan poco tráfico de red para evitar

interferencias, para esto se recomienda realizar en la segunda etapa un análisis de espectro y de la misma manera un realizar dicho análisis en cada instalación que se planea implementar una red Zigbee o como alternativa utilizar módulos Xbee que trabajen a frecuencias de 900Mhz.

- 3.- Para una segunda etapa se recomienda agregar un dip switch de 8 entradas y dos posiciones para ajustar el ID de cada dispositivo sin la necesidad de reprogramar o volver a cargar el firmware.
- 4.- Para una segunda etapa se utilizarán en su mayoría componentes de montaje superficial para reducción de costos y miniaturización del prototipo.
- 5.- Se recomienda cambiar de posición los pines de programación por ISP debido a que están demasiado cerca del dispositivo Xbee.

3.4 Estándares, y Registros de patentes

3.4.1 Estándares

Debido a que en un sistema Andon cada empresa interpreta de manera diferente los colores del sistema de luces no es posible encasillar un color con un significado, sin embargo, se puede definir una tendencia general de interpretación de colores como se ve en Tabla 5.

Color/Luz	Significado
Rojo estático	Máquina en fallo
Rojo parpadeante	Intervención de mantenimiento
Amarillo estático	En espera de abastecimiento de material
Amarillo parpadeante	Tiempo excedido de espera de material
Verde estático	Máquina en operación normal
Verde parpadeante	Máquina en preparación
Azul estático	Llamado a departamento de calidad
Azul parpadeante	Problema de calidad
Blanco estático	Corrida de producción piloto/conteo de piezas
Blanco parpadeante	Corrida de producción piloto/conteo de piezas
Café	Pieza defectuosa

Tabla 5 Tabla general de interpretación de colores en un sistema Andon. Fuente: elaboración propia.

Uno de los objetivos del proyecto era utilizar dispositivos Xbee debido a que funcionan bajo el estándar IEEE 802.15.4 por lo que al integrarlo a la tarjeta prototipo también se está trabajando bajo dicho estándar.

3.4.2 Registros de patente

Dado que el equipo se encuentra aún en etapa de desarrollo no se tienen intenciones de registro de patente, a su vez que por la naturaleza de su funcionamiento no se ha determinado que exista motivo o descubrimiento que justifique el inicio de trámite de patente y en todo caso si la empresa al final lo desea puede intentar registrar un modelo de utilidad.

3.5 Impacto social

Con los resultados de línea de tiempo (ver Figura 26) obtenidos de la tarjeta prototipo se pueden detectar máquinas con tiempos en fallo fuera de lo normal lo cual permite prestar más atención a dichas máquinas y enfocar esfuerzos concentrados en los problemas reales, con las alertas y notificaciones de fallo de máquina que envía la tarjeta se alerta oportunamente de dichos eventos y se reduce gradualmente la cantidad de tiempo en implementación de acciones correctivas

impactando directamente en la reducción de estrés general generando un mejor ambiente laboral en las empresas en las que se implementa [9][10].

3.6 Impacto tecnológico

Con la implementación de este dispositivo se ayuda a digitalizar los procesos de producción en empresas pequeñas, medianas y grandes ayudando en general a acelerar la convergencia de las empresas hacia la industria 4.0.

3.7 Impacto económico

En la Tabla 6 se muestra una comparativa de precios de implementación con el primer prototipo, la tabla toma en cuenta la instalación de un solo dispositivo de cada sistema:

Costo de sistema comercial			Costo de sistema desarrollado y software Enlacex		
Item	Concepto	costo(USD)	Item	Concepto	costo(USD)
A	Costo unitario de torreta Patlite con transmisor inalámbrico con protocolo Zigbee, gabinete y miscelaneos.	\$ 850.00	A	Costo unitario de torreta estandar con tarjeta electrónica, gabinete y miscelaneos	\$ 450.00
B	Costo de receptor Zigbee con salida ethernet	\$ 1,800.00	B	Costo de base receptora Zigbee universal	\$ 11.00
C	Costo de software de partner Japonés Signal Chain	\$ 5,200.00	C	Costo de software Enlacex	\$ 2,500.00
	Costo de hardware(items A y B)	\$ 2,650.00		Costo de hardware(items A y B)	\$ 461.00
	Costo de software(item C)	\$ 5,200.00		Costo de software(item C)	\$ 2,500.00
	Costo total	\$ 7,850.00		Costo total	\$ 2,961.00

Tabla 6 Tabla comparativa de precios de implementación. Fuente: elaboración propia.

En cuanto a costo de hardware por una parte tenemos el costo de \$2,650 USD en el sistema comercial y un costo de \$461 USD utilizando la tarjeta prototipo desarrollada en este proyecto en conjunto con un receptor universal de Xbee lo que

da una diferencia de \$2,200 USD equivalente a una reducción de 83.01% en el costo de hardware.

Hablando de costo total que incluye el costo del software, la diferencia de costo total es de 5,089 USD que equivale a una reducción de 64.82% con respecto a la versión comercial bastante significativa, esto utilizando el software Enlacex de Grupo Nauko.

3.8 Impacto ambiental

Al reducir la cantidad de piezas defectuosas por medio de la retroalimentación continua se evita utilizar materia prima adicional para contrarrestar la fabricación de dichas piezas reduciendo el uso de energía necesaria y el impacto ambiental que representa esta actividad correctiva.

Conclusiones

El módulo de XBee embebido en la tarjeta ha establecido comunicación con el receptor de datos adecuadamente, de igual manera los paquetes de datos fueron enviados al receptor satisfactoriamente al detectarse cambios en las entradas digitales de la tarjeta.

El total de entradas digitales de la tarjeta prototipo se extendió a 16 por lo que se cumplió satisfactoriamente el requerimiento original de seis entradas digitales.

En total de salidas de relevador de un polo un tiro de la tarjeta prototipo fue de doce siendo esta cantidad satisfactoria con el planteamiento original.

El envío de paquetes de datos desde la tarjeta prototipo hacia el receptor de datos se realizó con éxito cumpliendo de manera satisfactoria con el objetivo específico.

La tarjeta electrónica prototipo cuenta con barrier blocks en todas sus salidas y entradas por lo que cumple de manera satisfactoria el requerimiento y a su vez se conectó y a activó exitosamente a las lámparas de la torreta de muestra.

El prototipo cuenta con un Jack para conectar un eliminador de 5v y alimentar el circuito como se planteó en los objetivos.

El procesador Atmega 2560 funcionó correctamente y las pruebas de funcionamiento de entradas y salidas digitales funcionaron correctamente.

Con la tarjeta prototipo se realizaron las recopilaciones de datos satisfactoriamente de las entradas digitales para el conteo de piezas buenas, piezas malas, y la entrada indicadora de fallos por lo que se obtuvo satisfactoriamente el cálculo de la OEE por medio de la obtención de rendimiento, calidad y disponibilidad cumpliendo el objetivo general de este proyecto.

En implementaciones futuras se tiene pensado utilizar el mismo protocolo de comunicación Zigbee 802.15.4 pero esta vez en configuración red mesh y no en modo estrella transparente con el fin de extender el rango de alcance e incrementar la cantidad de equipos a cubrir.

Para las implementaciones futuras se recomienda realizar un análisis de espectro para determinar en qué canal de frecuencia sería más adecuado establecer la red ya que al incrementar el número de dispositivos se incrementan las posibilidades de

interferencia con otros equipos debido a que el rango de frecuencia de la gama de Xbee utilizado es de 2.4Ghz el cual es el mismo rango que utiliza el protocolo Wifi 2.4.

Dado que en el prototipo se requiere cargar el middleware con un ID diferente a cada dispositivo, en la segunda etapa del proyecto se agregará un dip switch de 8 entradas para asignar de manera sencilla el ID a cada dispositivo.

Bibliografía

- [14] G. Berry, "Challenges and potential solutions for complex embedded systems," in 2011 Proceedings of the Ninth ACM International Conference on Embedded Software (EMSOFT), Oct. 2011, p. 1,
DOI: [10.1145/2038642.2038644](https://doi.org/10.1145/2038642.2038644).
- [16] G. Carvalho and J. Bernardino, "The Internet of Things and big data: Future trends," in 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), 2017, pp. 1–4,
DOI: [10.23919/CISTI.2017.7975972](https://doi.org/10.23919/CISTI.2017.7975972).
- [17] G. Lei, G. Lu and Y. Sang, "Design of wireless Andon system based on ZigBee," 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI), Shenyang, 2015, pp. 821-825,
DOI: [10.1109/BMEI.2015.7401616](https://doi.org/10.1109/BMEI.2015.7401616).
- [10] I. Nada et al., "A study of organizational factors in occupational stress problems among workers in a polymer manufacturing factory," 2012 Southeast Asian Network of Ergonomics Societies Conference (SEANES), Langkawi, Kedah, 2012, pp. 1-5.
DOI: [10.1109/SEANES.2012.6299559](https://doi.org/10.1109/SEANES.2012.6299559)
- [11] Itek. (2015). ANDON System. 20/12/2020, de Zeit Sitio web: <http://www.itek.mx/language/andon-system.html>
- [19] I. Rašović and Z. Mijanovic, "Proposal of an industrial communication system based on ZigBee technology," 2020 24th International Conference on Information Technology (IT), Zabljak, Montenegro, 2020, pp. 1-4,
DOI: [10.1109/IT48810.2020.9070659](https://doi.org/10.1109/IT48810.2020.9070659).
- [15] J. Delsing, J. Eliasson, J. van Deventer, H. Derhamy and P. Varga, "Enabling IoT automation using local clouds," 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, 2016, pp. 502-507,
DOI: [10.1109/WF-IoT.2016.7845474](https://doi.org/10.1109/WF-IoT.2016.7845474).

- [2] K. E. Chong, K. C. Ng and G. G. G. Goh, "Improving Overall Equipment Effectiveness (OEE) through integration of Maintenance Failure Mode and Effect Analysis (maintenance-FMEA) in a semiconductor manufacturer: A case study," 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 2015, pp. 1427-1431.
DOI: [10.1109/IEEM.2015.7385883](https://doi.org/10.1109/IEEM.2015.7385883)
- [1] K. Krachangchan and N. Thawesaengskulthai, "Loss time reduction for improve Overall Equipment Effectiveness (OEE)," 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), Singapore, 2018, pp. 396-400.
DOI: [10.1109/IEA.2018.8387132](https://doi.org/10.1109/IEA.2018.8387132)
- [13] P. A. M. Juan Carlos Téllez Mosqueda, Diagnóstico para la fundamentación de la Maestría en Sistemas Embebidos, México, D.F: INFOTEC, Feb. 2014, p. 57, ISBN 978-607-7763-13-0
- [12] Pearson Educación; (1 enero 2011) Sommerville, I. (2011). Ingeniería de software. 9 edición. ISBN-13 : 978-6073206037
- [4] P. Jonsson and M. Lesshammar, "Measuring manufacturing performance: dimensions and OEE," Innovation in Technology Management. The Key to Global Leadership. PICMET '97, Portland, OR, USA, 1997, pp. 799-.
DOI: [10.1109/PICMET.1997.653644](https://doi.org/10.1109/PICMET.1997.653644)
- [6] P. A. Kullstam, "Availability, MTBF and MTTR for Repairable M out of N System," in IEEE Transactions on Reliability, vol. R-30, no. 4, pp. 393-394, Oct. 1981.
DOI: [10.1109/TR.1981.5221134](https://doi.org/10.1109/TR.1981.5221134)
- [20] T. Kumar and P. B. Mane, "ZigBee topology: A survey," 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2016, pp. 164-166,
DOI: [10.1109/ICCICCT.2016.7987937](https://doi.org/10.1109/ICCICCT.2016.7987937).
- [3]T. Sahoo, "Improving Overall Equipment Effectiveness (OEE) of process plant equipments through e-diagnostics," 2010 2nd International Conference on

Reliability, Safety and Hazard - Risk-Based Technologies and Physics-of-Failure Methods (ICRESH), Mumbai, 2010, pp. 107-111.

DOI: [10.1109/ICRESH.2010.5779815](https://doi.org/10.1109/ICRESH.2010.5779815)

- [8] W. Huang, "Designing a display unit to drive the 8x8 LED dot-matrix displays," 2013 IEEE 5th International Nanoelectronics Conference (INEC), Singapore, 2013, pp. 385-388.

DOI: [10.1109/INEC.2013.6466054](https://doi.org/10.1109/INEC.2013.6466054)

- [9] W. Zhou, F. Alexandre-Bailly and S. Piramuthu, "Dynamic Organizational Learning with IoT and Retail Social Network Data," 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, 2016, pp. 3822-3828.

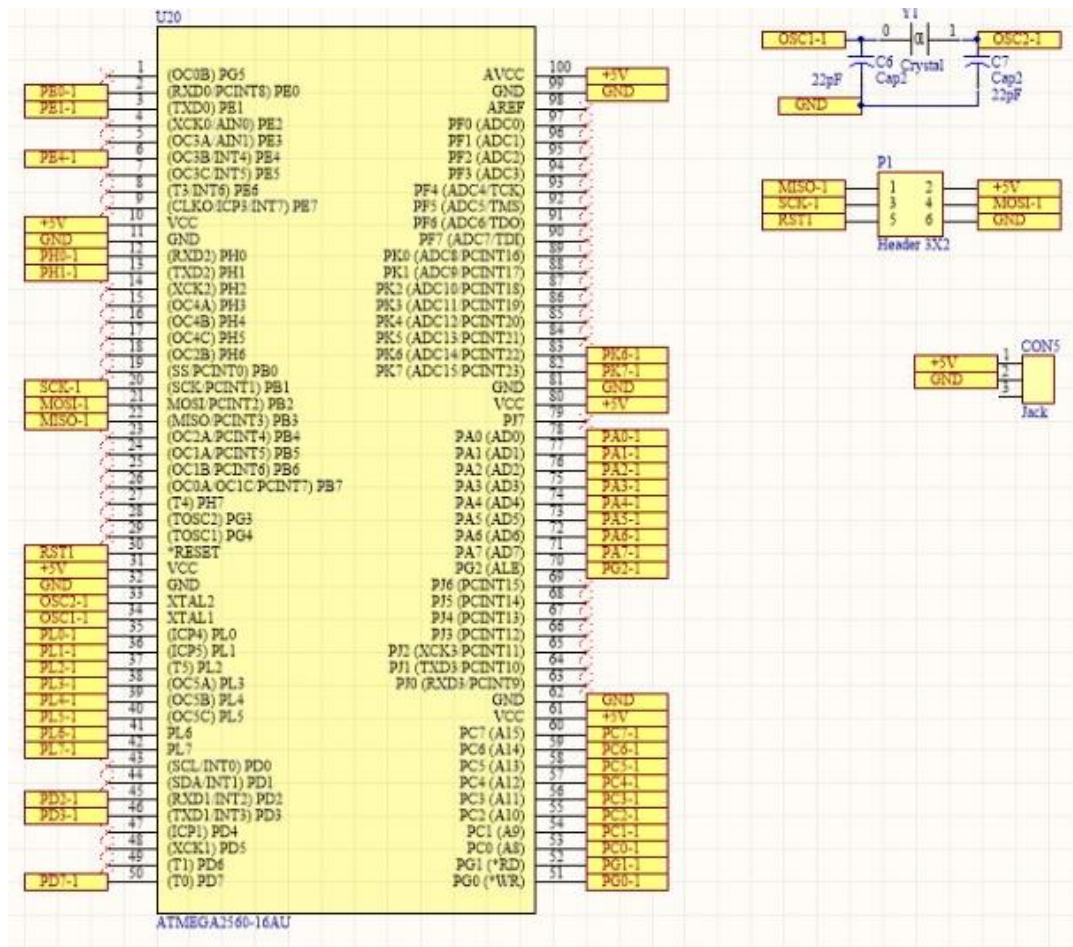
DOI: [10.1109/HICSS.2016.476](https://doi.org/10.1109/HICSS.2016.476)

- [5] X. Zhu, "Analysis and improvement of enterprise's equipment effectiveness based on OEE," 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, 2011, pp. 4167-4171.

DOI: [10.1109/ICECC.2011.6067706](https://doi.org/10.1109/ICECC.2011.6067706)

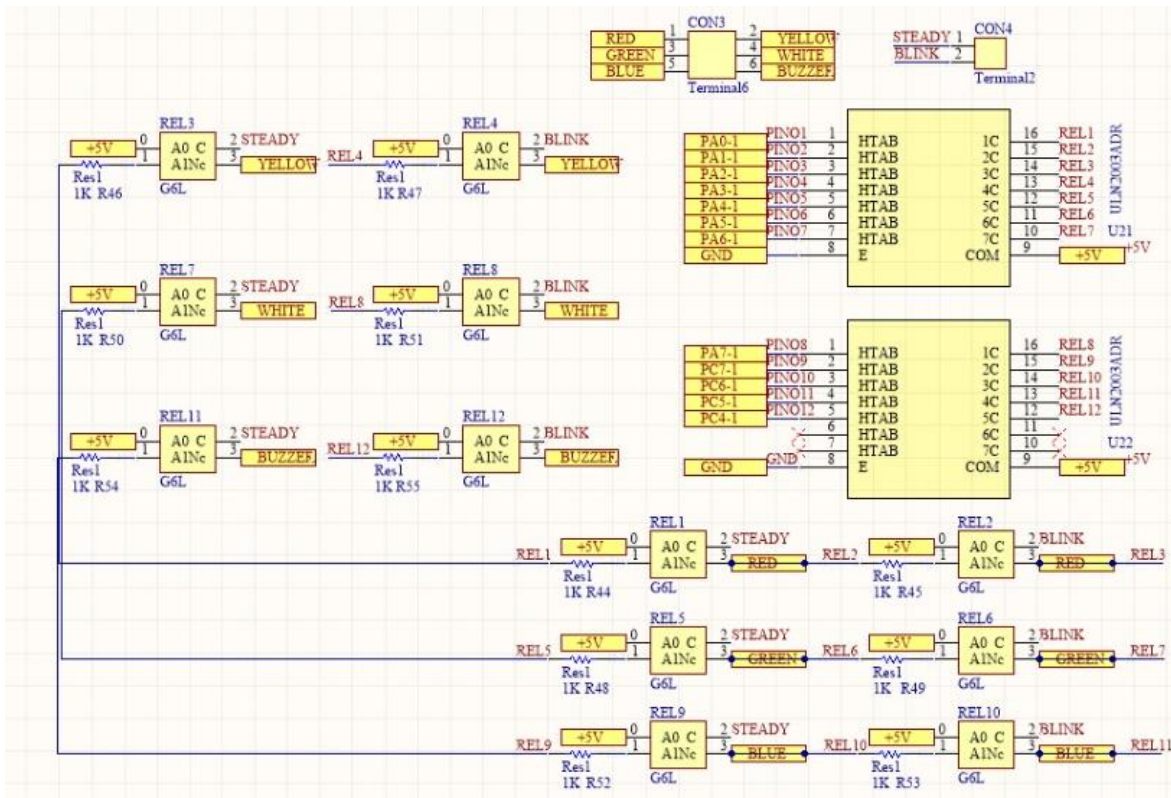
Anexo A - Diagrama principal

En este diagrama se muestra el microcontrolador principal Atmega2560, su arreglo de capacitores de 22pF, el conector Jack para alimentación y el conector ISP para la carga del firmware.



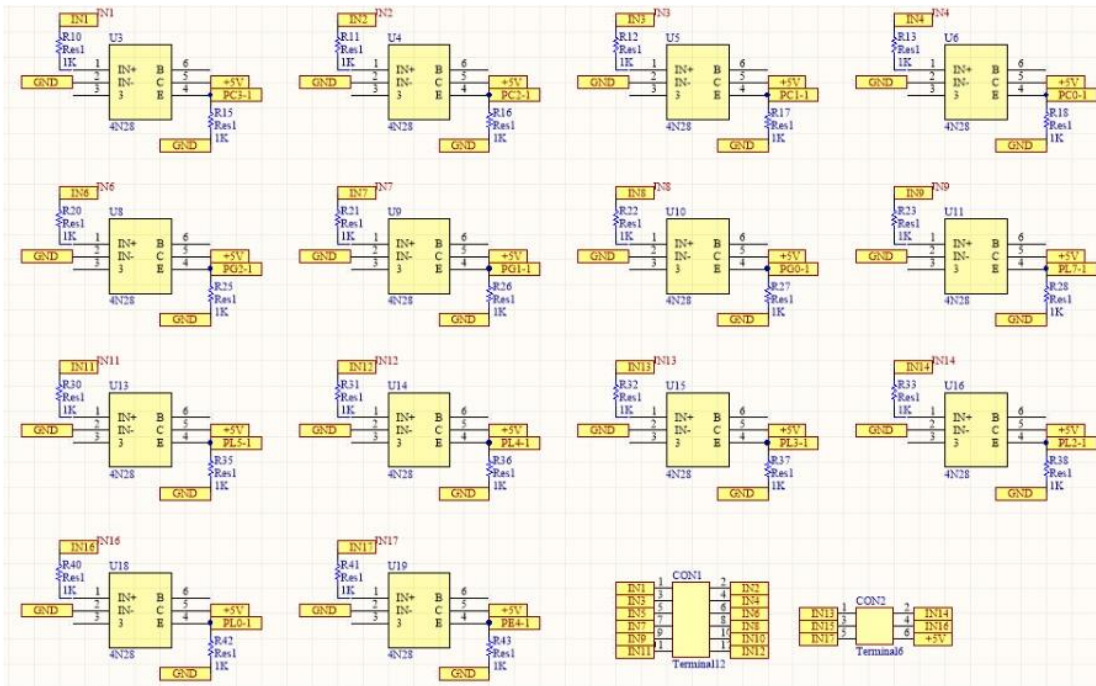
Anexo B - Diagrama de salidas digitales

Este diagrama muestra los relevadores Omron GL6-1F, utilizados para encender los leds de la torreta cuando una persona activa una entrada en específico con el selector de manera manual.



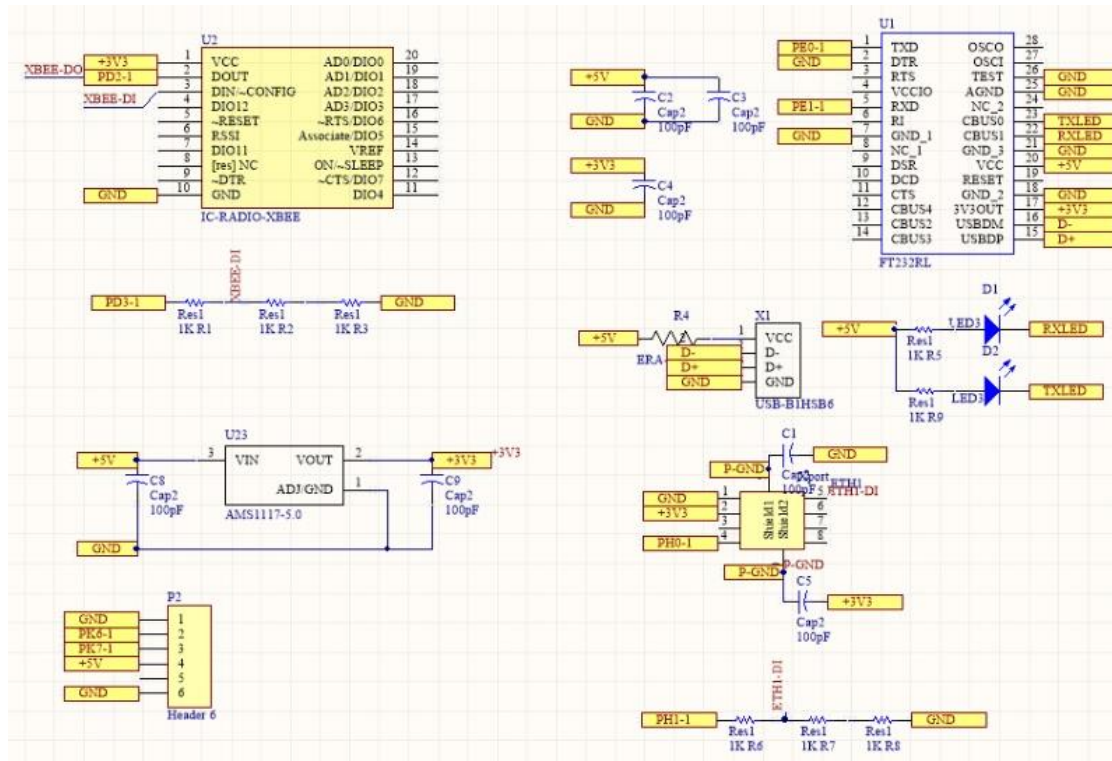
Anexo C - Diagrama de entradas digitales

En este diagrama se incluyen los optoacopladores 4n28 y su arreglo de resistencias para tensiones de entrada de 24v descritos en la sección 2.4 *Selección de tecnologías*.



Anexo D - Diagrama de comunicaciones

En este diagrama se incluyen los componentes de radio comunicación Xbee, el circuito integrado FT232RL para comunicación USB (reservado para uso futuro), el footprint para un conector ethernet (reservado para uso futuro), un regulador de voltaje AMS1117 de 3.3v, arreglo de tres resistencias de 10k ohms para regular voltaje de comunicación de entrada hacia el xbee y un par de leds de indicación de estado de comunicaciones.



Anexo E - Firmware de microcontrolador

A continuación, se presenta el código escrito utilizando la librería *Arduino.h*, para realizar la carga del código al microcontrolador atmega2560 de la placa prototipo se utilizó el conector ISP (In-system programming) localizado en la placa prototipo (Ver anexo A – Diagrama principal).

```
/*
  Firmware 2.1 Para tarjeta prototipo

  Funciones principales:
  -Lectura cíclica de entradas digitales asignadas a estatus específicos
  -
  Escritura en puertos con salida a relevador para encendido de colores de la
  torreta de ejemplo
  -
  Envío de buffer de datos con información de estado actual de cada entrada di
  gital a través de puerto serial 1 a 9600 baudios

  Escrito el 12 Mayo de 2020
  Por José Ramón Viramontes Olivares

*/
#include "arduino.h"

int relay12 = 22; //Salida a luz rojo estatica de la torreta
int relay11 = 23; //Salida a luz rojo parpadeante de la torreta
int relay10 = 24; //Salida a luz amarilla estatica de la torreta
int relay9 = 25; //Salida a luz amarilla parpadeante de la torreta
int relay8 = 26; //Salida a luz verde estatica de la torreta
int relay7 = 27; //Salida a luz verde parpadeante de la torreta
int relay6 = 28; //Salida a luz blanca estatica de la torreta
int relay5 = 29; //Salida a luz blanca parpadeante de la torreta
int relay4 = 30; //Salida a luz azul estatica de la torreta
int relay3 = 31; //Salida a luz azul parpadeante de la torreta
int relay2 = 32; //Salida a buzzer de audio 1 de la torreta
```



```
int relay1 = 33; //Salida a buzzer de audio 2 de la torreta

int input1 = 34; //Entrada para monitoreo de pin digital correspondiente a l
a luz roja estatica de la torreta
int input2 = 35; //Entrada para monitoreo de pin digital correspondiente a l
a luz roja parpadeante de la torreta
int input3 = 36; //Entrada para monitoreo de pin digital correspondiente a la
luz amarilla estatica de la torreta
int input4 = 37; //Entrada para monitoreo de pin digital correspondiente a l
a luz amarilla parpadeante de la torreta
int input5 = 38; //Entrada para monitoreo de pin digital correspondiente a la
luz verde estatica de la torreta
int input6 = 39; //Entrada para monitoreo de pin digital correspondiente a la
luz verde parpadeante de la torreta
int input7 = 40; //Entrada para monitoreo de pin digital correspondiente a la
luz azul estatica de la torreta
int input8 = 41; //Entrada para monitoreo de pin digital correspondiente a la
luz azul parpadeante de la torreta
int input9 = 42; //Entrada para monitoreo de pin digital correspondiente a la
luz blanca estatica de la torreta
int input10 = 43; //Entrada para monitoreo de pin digital correspondiente a l
a luz blanca parpadeante de la torreta
int input11 = 44; //Entrada para monitoreo de pin digital correspondiente a l
a salida a buzzer de audio 1 de la torreta
int input12 = 45; //Entrada para monitoreo de pin digital correspondiente a l
a salida a buzzer de audio 2 de la torreta
int input13 = 46; //extra reservado para funciones futuras
int input14 = 47; //extra reservado para funciones futuras
int input15 = 48; //extra reservado para funciones futuras
int input16 = 49; //extra reservado para funciones futuras
int input17 = 2; //extra reservado para funciones futuras

//función de inicio
```

```

void setup() {
Serial.begin(9600); //Declaramos el puerto serial a utilizar y los ajustes a
9600 baudios

// Declaramos los pines del microcontrolador que se utilizarán como salida
s
pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode (relay3, OUTPUT);
pinMode (relay4, OUTPUT);
pinMode (relay5, OUTPUT);
pinMode (relay6, OUTPUT);
pinMode (relay7, OUTPUT);
pinMode (relay8, OUTPUT);
pinMode (relay9, OUTPUT);
pinMode (relay10, OUTPUT);
pinMode (relay11, OUTPUT);
pinMode (relay12, OUTPUT);

// Declaramos los pines del microcontrolador que se utilizarán como entrad
as
pinMode (input1, INPUT);
pinMode (input2, INPUT);
pinMode (input3, INPUT);
pinMode (input4, INPUT);
pinMode (input5, INPUT);
pinMode (input6, INPUT);
pinMode (input7, INPUT);
pinMode (input8, INPUT);
pinMode (input9, INPUT);
pinMode (input10, INPUT);
pinMode (input12, INPUT);
pinMode (input13, INPUT);
pinMode (input14, INPUT);

```

```

pinMode (input15, INPUT);
pinMode (input16, INPUT);
pinMode (input17, INPUT);

}

/*
Variables para almacenar el estatus actual de cada entrada digital
    -cuando vale 0 es valor no asignado o inicial
    -cuando vale 1 es señal en alto o 5v de entrada
    -cuando vale 2 es señal en bajo o 0v de entrada
*/
int status_input1 = 0;
int status_input2 = 0;
int status_input3 = 0;
int status_input4 = 0;
int status_input5 = 0;
int status_input6 = 0;
int status_input7 = 0;
int status_input8 = 0;
int status_input9 = 0;
int status_input10 = 0;
int status_input11 = 0;
int status_input12 = 0;
int status_input13 = 0;
int status_input14 = 0;
int status_input15 = 0;
int status_input16 = 0;
int status_input17 = 0;

/*

```

```

Variables para almacenar el último estatus de cada entrada digital
    -cuando vale 0 es valor no asignado o inicial
    -cuando vale 1 es señal en alto o 5v de entrada
    -cuando vale 2 es señal en bajo o 0v de entrada

*/
int last1 = 2;
int last2 = 2;
int last3 = 2;
int last4 = 2;
int last5 = 2;
int last6 = 2;
int last7 = 2;
int last8 = 2;
int last9 = 2;
int last10 = 2;
int last11 = 2;
int last12 = 2;
int last13 = 2;
int last14 = 2;
int last15 = 2;
int last16 = 2;
int last17 = 2;

//contadores deciclos de lectura para el filtro de ruido por contacto mecánico en cada entrada digital
int c_ruido1 = 0;
int c_ruido2 = 0;
int c_ruido3 = 0;
int c_ruido4 = 0;
int c_ruido5 = 0;
int c_ruido6 = 0;
int c_ruido7 = 0;
int c_ruido8 = 0;

```

```

int c_ruido9 = 0;
int c_ruido10 = 0;
int c_ruido11 = 0;
int c_ruido12 = 0;
int c_ruido13 = 0;
int c_ruido14 = 0;
int c_ruido15 = 0;
int c_ruido16 = 0;
int c_ruido17 = 0;

bool existioCambioDeEstado=0;//variable para filtro de cambios de estado

int filtroderuido = 5000;//cantidad de ciclos de lectura necesarios para que
se tome un valor lógico como verdadero

//función de monitoreo para entrada digital 1
void monitorDeEntrada1() {

    status_input1 = digitalRead(input1);//leemos el estatus actual de la entra
da digital

    if (status_input1 != last1) {//comparamos si es diferente a la ultima lect
ura

        if (c_ruido1 > filtroderuido) {//comparamos si ya han pasado los ciclos
de lectura necesarios para un filtrado de ruido mecánico

            last1 = status_input1;//asignamos la lectura actual a la variable de ú
ltima lectura para el siguiente ciclo

```

```

    if (status_input1 == 1) {
        //si el estado actual de la entrada es 1 o 5v entonces activamos el
relevador de salida correspondiente para encender la luz correspondiente de
la torreta
        digitalWrite(relay1, HIGH);
    } else {

        //si el estado actual de la entrada es 2 ó 0 ó 5v entonces d
eactivamos el relevador de salida correspondiente para apagar la luz corres
pondiente de la torreta

        digitalWrite (relay1, LOW);
    }

} else {

    c_ruido1++;//si la lectura actual es diferente a la ultima validada po
r los filtros y el contador de ciclos sigue siendo menor al límite estableci
do para ser validado como cambio entonces incrementamos el contador de ciclo
s en +1
    }

} else {

    c_ruido1 = 0;//si no hay ningún cambio detectado reiniciamos el contador
de ciclos a cero
    }

}
}

```

```

//función de monitoreo para entrada digital 2
void monitorDeEntrada2() {

    status_input2 = digitalRead(input2);

    if (status_input2 != last2) {

        if (c_ruido2 > filtroderuido) {

            last2 = status_input2;

            if (status_input2 == 1) {

                digitalWrite(relay2, HIGH);
            } else {
                digitalWrite(relay2, LOW);
            }

        } else {

            c_ruido2++;
        }

    } else {

        c_ruido2 = 0;
    }

}
}

```

```

void monitorDeEntrada3() { //función de monitoreo para entrada digital 3

    status_input3 = digitalRead(input3);

    if (status_input3 != last3) {

        if (c_ruido3 > filtroderuido) {

            last3 = status_input3;

            if (status_input3 == 1) {

                digitalWrite(relay3, HIGH);
            } else {
                digitalWrite(relay3, LOW);
            }

        } else {

            c_ruido3++;
        }

    } else {

        c_ruido3 = 0;
    }

}
}

```



```

//función de monitoreo para entrada digital 4
void monitorDeEntrada4() {

    status_input4 = digitalRead(input4);

    if (status_input4 != last4) {

        if (c_ruido4 > filtroderuido) {

            last4 = status_input4;

            if (status_input4 == 1) {

                digitalWrite(relay4, HIGH);
            } else {
                digitalWrite(relay4, LOW);
            }

        } else {

            c_ruido4++;
        }

    } else {

        c_ruido4 = 0;
    }

}

//función de monitoreo para entrada digital 5
void monitorDeEntrada5() {

```

```

status_input5 = digitalRead(input5);

if (status_input5 != last5) {

    if (c_ruido5 > filtroderuido) {

        last5 = status_input5;

        if (status_input5 == 1) {

            digitalWrite(relay5, HIGH);
        } else {
            digitalWrite(relay5, LOW);
        }

    } else {

        c_ruido5++;
    }

} else {

    c_ruido5 = 0;
}

}

//función de monitoreo para entrada digital 6
void monitorDeEntrada6() {

    status_input6 = digitalRead(input6);

    if (status_input6 != last6) {

        if (c_ruido6 > filtroderuido) {

```

```

    last6 = status_input6;

    if (status_input6 == 1) {

        digitalWrite(relay6, HIGH);
    } else {
        digitalWrite(relay6, LOW);

    } else {

        c_ruido6++;
    }

} else {

    c_ruido6 = 0;
}

}

//función de monitoreo para entrada digital 7
void monitorDeEntrada7() {

    status_input7 = digitalRead(input7);

    if (status_input7 != last7) {

        if (c_ruido7 > filtroderuido) {

            last7 = status_input7;

            if (status_input7 == 1) {

                digitalWrite(relay7, HIGH);

```

```

    } else {
        digitalWrite(relay7, LOW);
    }

    } else {

        c_ruido7++;
    }

} else {

    c_ruido7 = 0;
}

}

//función de monitoreo para entrada digital 8
void monitorDeEntrada8() {

    status_input8 = digitalRead(input8);

    if (status_input8 != last8) {

        if (c_ruido8 > filtroderuido) {

            last8 = status_input8;

            if (status_input8 == 1) {

                digitalWrite(relay8, HIGH);
            } else {
                digitalWrite(relay8, LOW);
            }

        } else {

```

```

        c_ruido8++;
    }

} else {

    c_ruido8 = 0;
}

}

//función de monitoreo para entrada digital 9

void monitorDeEntrada9() {

    status_input9 = digitalRead(input9);

    if (status_input9 != last9) {

        if (c_ruido9 > filtroderuido) {

            last9 = status_input9;

            if (status_input9 == 1) {

                digitalWrite(relay9, HIGH); // turn the LED on (HIGH is the voltage level)
            } else {
                digitalWrite(relay9, LOW); // turn the LED on (HIGH is the voltage level)
            }

        } else {


```

```

        c_ruido9++;
    }

} else {

    c_ruido9 = 0;
}

}

//función de monitoreo para entrada digital 10
void monitorDeEntrada10() {

    status_input10 = digitalRead(input10);

    if (status_input10 != last10) {

        if (c_ruido10 > filtroderuido) {

            last10 = status_input10;

            if (status_input10 == 1) {

                digitalWrite(relay10, HIGH); // turn the LED on (HIGH is the volta
ge level)
            } else {
                digitalWrite(relay10, LOW); // turn the LED on (HIGH is the voltag
e level)
            }

        } else {

            c_ruido10++;
        }
    }
}

```

```

    }

    } else {

        c_ruido10 = 0;
    }
}

//función de monitoreo para entrada digital 11
void monitorDeEntrada11() {

    status_input11 = digitalRead(input11);

    if (status_input11 != last11) {

        if (c_ruido11 > filtroderuido) {

            last11 = status_input11;

            if (status_input11 == 1) {

                digitalWrite(relay11, HIGH); // turn the LED on (HIGH is the volta
ge level)
            } else {
                digitalWrite(relay11, LOW); // turn the LED on (HIGH is the voltag
e level)
            }

        } else {

            c_ruido11++;
        }
    }
}

```

```

} else {

    c_ruido11 = 0;
}

}

//función de monitoreo para entrada digital 12
void monitorDeEntrada12() {

    status_input12 = digitalRead(input12);

    if (status_input12 != last12) {

        if (c_ruido12 > filtroderuido) {

            last12 = status_input12;

            if (status_input12 == 1) {

                digitalWrite(relay12, HIGH); // turn the LED on (HIGH is the volta
ge level)
            } else {
                digitalWrite(relay12, LOW); // turn the LED on (HIGH is the voltag
e level)
            }

        } else {

            c_ruido12++;
        }

    } else {

        c_ruido12 = 0;
    }
}

```



```

}

}

byte bufferDeMensaje[]={0,0,0,0,0,0,0,0}; //buffer para el mensaje de salida
de datos

byte ID=1; //Byte para el ID del dispositivo, en este caso es 1

//método cíclico principal
void loop() {

    existioCambioDeEstado=false; //inicializamos en false la variable que indic
a si hubo un cambio de estado
    //llamamos a las funciones de lectura de entradas digitales 1,2,3 y 4, (la
s demás están reservadas para uso futuro)
    monitorDeEntrada1();
    monitorDeEntrada2();
    monitorDeEntrada3();
    monitorDeEntrada4();

    if(existioCambioDeEstado==true){ //Si dentro de las funciones de lectura de
entradas digitales se detecto un cambio de estado procedemos a enviar el men
saje por el puerto serial

    //llenamos el buffer del mensaje con las llaves de entrada y cierre, ID del
dispositivo y los últimos estados de cada entrada digital
    bufferDeMensaje[0]=253; //llave de inicio de mensaje

    //ID del dispositivo (reassignable)
    bufferDeMensaje[1]=ID;

```

```

//Estatus de entradas digitales 1,2,3, 4 y 5
(color rojo estático, amarillo estático, verde estático, señal de conteo de
piezas y señal de conteo de piezas defectuosas)
bufferDeMensaje[2]=last1;
bufferDeMensaje[3]=last2;
bufferDeMensaje[4]=last3;
bufferDeMensaje[5]=last4;
bufferDeMensaje[6]=last5;

bufferDeMensaje[7]=252;//Llave de fin de mensaje

//nota: las llaves de inicio y final sirven para indicarle al receptor que e
l mensaje enviado es un mensaje correspondiente al sistema.

Serial.write(bufferDeMensaje, 8);//enviamos el buffer de datos por el puerto
serial, este pasará por el XBee en modo transparente hasta el Xbee receptor
.

}

}

```