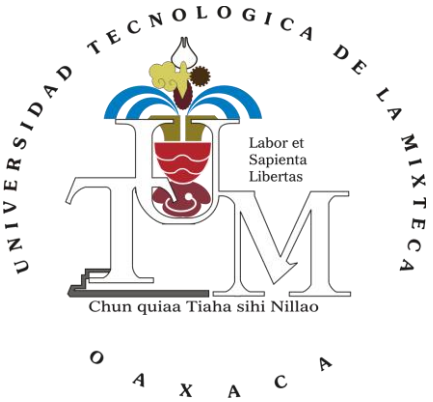


UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

DISEÑO DE UN SISTEMA REDUNDANTE TIPO HOT-STANDBY PARA CONTROLAR UN PROCESO TRANSFER CIRCULAR

TESIS
PARA OBTENER EL TÍTULO DE:
INGENIERO EN MECATRÓNICA

PRESENTA:
JULIO JOSAFAT PÉREZ MEDINA

DIRECTOR DE TESIS:
DR. ANTONIO ORANTES MOLINA

CO-DIRECTOR:
DR. ROSEBET MIRANDA LUNA

HUAJUAPAN DE LEÓN, OAXACA, MÉXICO
JUNIO DE 2024

Tesis presentada en junio de 2024
ante los siguientes sinodales:

Dr. Fermín Hugo Ramírez Leyva
Dra. Esther Lugo González
M.C. Moisés Manzano Herrera

Director de tesis:
Dr. Antonio Orantes Molina

Codirector de tesis:
Dr. Rosebet Miranda Luna

A todos mis seres queridos,
por su apoyo incondicional, paciencia
y amor. Sin ustedes esto jamás
hubiera sido posible

Agradecimientos

A mi padre y a mi madre, Julio y Martha, cuya inquebrantable fe en mí y constante apoyo han sido pilares fundamentales en mi vida. Su amor incondicional y su ejemplo de perseverancia me han inspirado a lo largo de este camino académico. Gracias por estar siempre a mi lado, por creer en mis sueños y por el sacrificio que hacen para brindarme las mejores oportunidades. Esta tesis es un logro tanto mío como de ustedes y les dedico este trabajo con todo mi cariño y gratitud.

A mis hermanos, Eliud y Axel, por ser el motor de mi motivación a ser una mejor persona, espero ser un buen ejemplo en sus vidas. Gracias por todos los buenos y malos momentos que hemos vivido juntos, por estar con mamá y papá en mi ausencia. Sin ustedes esto no hubiera sido posible.

A Andrea, me gustaría agradecer su apoyo incondicional e inquebrantable a lo largo de la realización de este proyecto de tesis. Por siempre estar dispuesta a ayudarme y escucharme en los momentos difíciles. Tu amor ha sido mi guía a lo largo de este camino, gracias por ser mi compañera de vida y por siempre creer en mí y alentarme a seguir adelante. Este proyecto también es gracias a ti, ya que indirectamente influiste en su realización.

A mis mejores amigos, les agradezco profundamente. A Pavel Palma (Pavel), mi mejor amigo, le agradezco todos los consejos que me dio, sobre todo el enseñarme a siempre ser mejor persona y hacer el bien sin esperar nada a cambio, a Josué Santiago (Josué) e Isaías Paz (Chay) por ser los mejores compañeros de equipo que alguien pudiera pedir, por todos los desvelos y momentos que vivimos a lo largo de la carrera, a Fernando Pérez (Nando) por haber sido mi

primer amigo en esta travesía que fue la universidad, a Eduardo Ferra (Lalito) por ser un ejemplo de trabajo y esfuerzo, y a todos los amigos que me acompañaron a lo largo de este camino y a aquellos que estaban conmigo desde antes, a todos ustedes mi más sincera gratitud, gracias por sus consejos y por todos los buenos momentos que pasamos juntos. Siempre atesoraré su amistad.

A mi director de tesis, el Dr. Antonio Orantes Molina, por todo el apoyo que me ha otorgado a lo largo de los últimos años. Desde la realización de mi trabajo de servicio social hasta la culminación de este trabajo de tesis. Mi más sincero agradecimiento por haber confiado en mí, por haberme guiado a lo largo de este trabajo y por haberme transmitido sus conocimientos en el salón de clases. De igual manera, quiero agradecer a mi codirector de tesis, el Dr. Rosebet Miranda Luna, por haber creído en mi y por tomar siempre el tiempo para la revisión y orientación de este trabajo.

A mis sinodales, la Dra. Esther Lugo, el Dr. Hugo Leyva y el M.C. Moisés Manzano, cuyos valiosos comentarios y observaciones han sido fundamentales para el desarrollo de esta tesis. Su generosidad al compartir su tiempo y conocimientos ha enriquecido significativamente este trabajo. A cada uno de ustedes, mi más profundo agradecimiento.

Y, por último, agradecer a la Universidad Tecnológica de la Mixteca, por sus instalaciones y el personal en general que facilitó el equipo para la realización del prototipo físico de este proyecto. Sin el apoyo de la institución y del personal este trabajo no hubiera sido posible.

Gracias.

RESUMEN

En la automatización, el objetivo es reducir al mínimo las fallas en los procesos y detectarlas rápidamente para repararlas y evitar interrupciones. A pesar de que las fallas suelen ser causadas por factores externos, los controladores de los procesos también pueden fallar, lo que desestabiliza el sistema y provoca daños significativos. Por esta razón, la implementación de arquitecturas redundantes tipo ***hot-standby*** en los controladores es clave para garantizar la fiabilidad y disponibilidad de un proceso, permitiendo que un sistema de respaldo se active de inmediato en caso de fallo del sistema principal.

En este trabajo de tesis se aborda el diseño y la construcción de un sistema redundante tipo ***hot-standby*** utilizando dos micro-PLC S7-200 de SIEMENS. Esto se aplica a un prototipo de un proceso tipo transfer circular, el cual consiste de tres estaciones de trabajo que pueden operar de manera simultánea. Este sistema es construido utilizando la metodología de la guía GEMMA, permitiendo crear desde cero la automatización de todo el proceso y llevando un control sobre las estaciones de trabajo para poder añadir más en un futuro. En cuanto al sistema redundante, el proceso es controlado por un PLC principal, mientras un segundo PLC se encuentra monitoreando los estados del primero y obteniendo información del mismo, pues en caso de que el PLC principal sufra una falla o una desconexión el segundo PLC se encuentra listo para tomar el mando del sistema de manera inmediata, logrando evitar pérdida de información y aumentando la disponibilidad del sistema. También se considera un error de congelamiento en las salidas RLY del PLC principal, donde las salidas permanecen encendidas en un estado específico. En este caso el PLC secundario emite una alarma para indicar que se debe realizar la desconexión del PLC principal. Una vez que el PLC principal se encuentre desconectado el PLC secundario toma el mando del sistema.

ABSTRACT

In automation, the goal is to minimize process failures and quickly detect them for repair to avoid interruptions. Although failures are usually caused by external factors, process controllers can also fail, which destabilizes the system and causes significant damage. For this reason, implementing hot-standby redundant architectures in controllers is key to ensuring process reliability and availability, allowing a backup system to activate immediately in case of a main system failure.

This thesis addresses the design and construction of a hot-standby redundant system using two SIEMENS S7-200 micro-PLC. This is applied to a prototype of a circular Transfer type process, which consists of three workstations that can operate simultaneously. This system is built using the GEMMA guide methodology, allowing the entire process automation to be created from scratch and controlling the workstations to add more in the future. Regarding the redundant system, the process is controlled by a main PLC, while a second PLC monitors the states of the first and gathers information from it, so that in case the main PLC fails or disconnects, the second PLC is ready to take control of the system immediately, preventing information loss and increasing system availability. It is also considered an error of freezing in the RLY outputs of the main PLC, where the outputs remain on in a specific state. In this case, the secondary PLC issues an alarm to indicate that the main PLC should be disconnected. Once the main PLC is disconnected, the secondary PLC takes control of the system.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1

Introducción

En el campo de la automatización siempre se ha buscado que los procesos automatizados sufran la menor cantidad posible de fallas y en caso de presentarlas, que sean detectadas en el menor tiempo posible para poder ser reparadas a la brevedad y evitar perder demasiado tiempo de operación. A lo largo de los años se han desarrollado distintas tecnologías y técnicas que permiten reducir la cantidad de fallas, tiempo de espera y costos de mantenimiento, desde medir los tiempos de operación normal de ciertos actuadores para detectar fallas cuando el tiempo es mayor al normal, hasta algoritmos complejos que adquieren datos de distintos sensores que monitorean ciertas variables en busca de anomalías en el sistema [1].

A pesar de que las fallas en un proceso industrial suelen ocurrir por eventos externos como atascamiento de actuadores u obstrucción del flujo del sistema, los controladores del proceso no se encuentran exentos de presentar algún fallo durante su funcionamiento. Esto puede llevar a la inestabilidad del sistema, por lo que un fallo del controlador del proceso se traduce en grandes daños y pérdidas [2]. En este contexto, la implementación de arquitecturas redundantes tipo ***hot-standby*** es una estrategia para asegurar la fiabilidad y disponibilidad del sistema en todo momento [3]. Estas arquitecturas requieren un sistema de respaldo que pueda

activarse de inmediato en caso de fallo del sistema principal. Esto implica una integración cuidadosa de múltiples componentes, desde sensores y actuadores hasta lógica de control, para garantizar una transición suave y sin interrupciones en caso de fallo del sistema principal.

Los aspectos fundamentales en el diseño de sistemas redundantes son la elección adecuada de componentes y la implementación de algoritmos de conmutación eficientes. Esto implica evaluar detenidamente las características de los componentes, su confiabilidad y su capacidad para adaptarse a situaciones de emergencia. Además, se requiere un análisis exhaustivo de los posibles escenarios de falla y un plan de emergencia detallado para garantizar una respuesta rápida y efectiva en caso de que ocurra una falla en el sistema principal.

En este trabajo se presenta la propuesta de un método de construcción de un sistema redundante tipo ***hot-standby*** compuesto por dos PLC S7-200 del fabricante alemán SIEMENS conectados en una red de interfaz punto a punto, de tal forma que siempre se encuentren comunicados entre sí. Se presenta, además, el diseño de un prototipo físico que simula un proceso tipo transfer circular, cuyo objetivo es validar la actuación del sistema redundante, pues un PLC principal debe de controlar este proceso en todo momento, mientras un PLC secundario se mantiene en espera, aunque siempre intercambiando información con el PLC principal, pues en caso de falla de éste, el PLC secundario debe asumir el control total del proceso sin ningún tipo de pérdida de información.

El proceso representado por el prototipo cuenta con la particularidad de ser diseñado utilizando como base la guía de estudio de los modos de marcha y paro (GEMMA, por sus siglas en francés), la cual trata de una serie de técnicas utilizadas para monitorear operaciones industriales, que incluyen tanto la seguridad del sistema como la supervisión activa del operador sobre las diferentes etapas del proceso de producción. Esto abarca desde el inicio o detención del algoritmo de control, hasta la intervención en actividades automáticas, semiautomáticas, modo manual, diagnóstico y manejo de fallos, entre otras acciones [4].

1.1 Marco teórico

A continuación, se proporciona el contexto necesario para comprender el tema en cuestión. En esta sección se exploran teorías, conceptos y estudios previos relevantes que sustentan y enriquecen el trabajo.

1.1.1 Automatización

La automatización es la tecnología en la que un proceso es operado y controlado por sistemas mecánicos, electrónicos y computacionales sin la participación directa de un trabajador humano [5]. Para implementar la automatización en un proceso se pueden construir desde sistemas lógicos de relés cableados hasta sistemas complejos que procesan información a través de técnicas de programación. Estos últimos incluyen tecnologías como microprocesadores, PC industriales, autómatas programables, controles de robots y buses de control. La automatización se encuentra presente en la mayoría de los componentes de los sistemas de producción de las empresas, principalmente en los sistemas de manufactura y computación. Algunos ejemplos pueden ser sistemas de ensamblado automatizado, procesos tipo transfer que realicen una serie de operaciones de maquinado, sistemas de inspección automática para control de calidad, sistemas de almacenamiento automático de material para operaciones de manufactura, estaciones robotizadas, etc.

1.1.2 Controladores Lógicos Programables (PLC)

Los autómatas programables o Controladores Lógicos Programables (PLC) son máquinas electrónicas que pilotan o gobiernan procedimientos lógicos secuenciales en ambientes industriales en tiempo real [6]. Un PLC trabaja con base en la información que recibe de sus entradas y, siguiendo un programa lógico interno, genera una respuesta en sus salidas. Generalmente, las entradas suelen ser conectarse a dispositivos como botones, interruptores o sensores de los que se espera recibir una señal, ya sea de parte de un operador humano o la misma máquina, mientras que las salidas suelen estar conectadas a distintos actuadores que se espera que ejecuten una función en

la máquina, como pistones o motores que realicen algún tipo de movimiento.

1.1.2.1 PLC S7-200

Los micro-PLC de la gama S7-200 del fabricante SIEMENS son dispositivos compactos, con configuración flexible e incluyen un amplio juego de operaciones, desde operaciones de lógica booleana, contadores, temporizadores, operaciones aritméticas complejas y comunicación. En la Figura 1.1 se muestran los componentes externos del PLC.

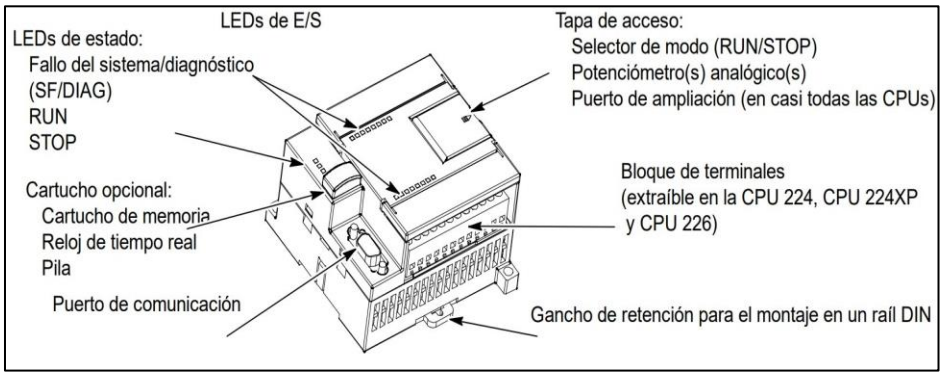


Figura 1.1: Micro PLC S7-200 [7].

1.1.2.1.1 CPU 224

SIEMENS ofrece diferentes modelos de la CPU S7-200. En este caso en particular se redactan las especificaciones del modelo CPU 224.

- Dimensiones de 120.5 x 80 x 62 mm.
- 14 entradas y 10 salidas digitales.
- 12288 bytes de memoria del programa con edición en tiempo de ejecución y 16384 bytes sin edición en tiempo de ejecución.
- 8192 bytes de memoria de datos.
- Compatible con 7 módulos de ampliación.
- 2 potenciómetros analógicos.
- Reloj de tiempo real incorporado.

El PLC puede ser alimentado tanto con corriente directa (24 VDC) o alterna (85 a 265 V AC). Para programar este PLC se hace uso de la herramienta de software STEP 7-MicroWIN®, la cual cuenta con un entorno para la programación en lenguaje Escalera, además de ofrecer fácil acceso a todas las operaciones con las que cuenta el PLC. El S7-200 ejecuta las tareas del programa en un ciclo de manera repetitiva, el cual es llamado ciclo *scan* y consiste en leer las entradas físicas, ejecutar la lógica de control en el programa, procesar las peticiones de comunicación, efectuar el auto diagnóstico de la CPU y finalmente escribir en las salidas físicas [7]. El PLC tiene un tiempo de respuesta de 3 a 10 milisegundos. Es decir, cuando todos los registros de memoria se encuentren en uso durante la ejecución del programa y el PLC se encuentre a su máximo nivel de estrés, el tiempo más largo de respuesta en las salidas será de 10 milisegundos.

1.1.3 Procesos tipo transfer

Los procesos tipo transfer son los sistemas más automatizados y productivos en cuanto a la cantidad de operaciones que pueden efectuar para acomodar geometrías de trabajo complejas, pues son procesos que cuentan con dos o más estaciones de trabajo que cumplen un paso dentro del procesamiento de una pieza o parte de manera secuencial, pero a la vez trabajan de manera simultánea, incrementando así la productividad del sistema. Estos procesos pueden acomodarse de manera lineal, donde las estaciones de trabajo se encuentran de manera lineal y las partes a trabajar se mueven de estación en estación a través de mecanismos como vigas móviles o bandas transportadoras. Otra configuración muy común de los procesos tipo transfer son las máquinas rotatorias, donde una mesa de trabajo circular horizontal sostiene las piezas a ser procesadas, mientras que en la periferia de esta mesa se encuentran las estaciones de trabajo. La mesa es capaz de girar de tal modo que cada pieza pueda pasar por todas las estaciones de trabajo. Estos sistemas rotatorios están limitados a menor carga de trabajo y menos estaciones de trabajo que los sistemas de línea [5].

1.1.4 Grafcet

El Gráfico Funcional de Control mediante Etapas y Transiciones (Grafcet, por sus siglas

en francés) es una herramienta importante para la automatización de procesos, pues es un modelo gráfico que sirve para esquematizar un proceso autómata secuencial. Este consta de etapas, acciones y transiciones. Las etapas definen el estado en el que se encuentra el automatismo (las etapas de inicio se marcan con doble cuadro). Las acciones se encuentran asociadas a cada etapa y define lo que se realiza durante ésta, como accionar un motor, encender un indicador, etc. Las transiciones son las condiciones que hacen que el Grafcet pase de una etapa a la siguiente, como un pulsador, una señal de sensor o la cuenta de un temporizador [8]. En la Figura 1.2 se muestra la representación gráfica de cómo se acomodan estos componentes del Grafcet.

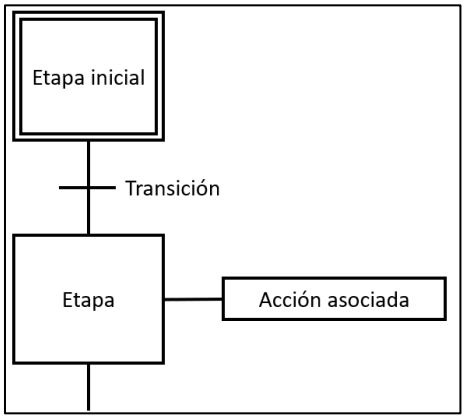


Figura 1.2: Componentes y sintaxis de un Grafcet.

El Grafcet cuenta con distintas reglas para su elaboración, como el hecho de que obligatoriamente después de una transición debe ir una etapa, así como no se pueden conectar dos etapas sin una transición de por medio. La etapa inicial puede tener acciones asociadas que son las condiciones iniciales del sistema.

1.1.4.1 Secuencia en bucle

Una secuencia en bucle se repite constantemente, de tal forma que, al desactivarse la última etapa por medio de una transición, se activará la etapa inicial [9]. Mediante una flecha se indica el retorno ascendente de la línea de enlace, tal como se muestra en la Figura 1.3. El punto dentro de la etapa 0 indica que es la etapa que se encuentra activa.

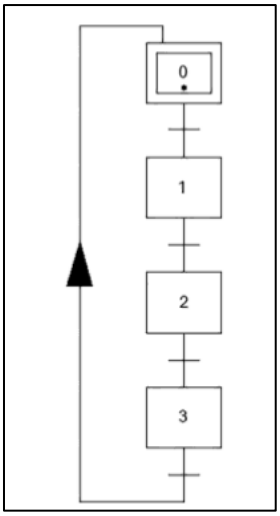


Figura 1.3: Secuencia en bucle de un Grafcet [9].

1.1.4.2 Etapas simultáneas

Las etapas simultáneas son etapas que se activan al mismo tiempo después de que una transición se cumple. Se representan con una línea doble que indica la separación simultánea de las etapas. En la Figura 1.4 se muestra que la condición **c*d** activa las etapas 6,7 y 8 al mismo tiempo.

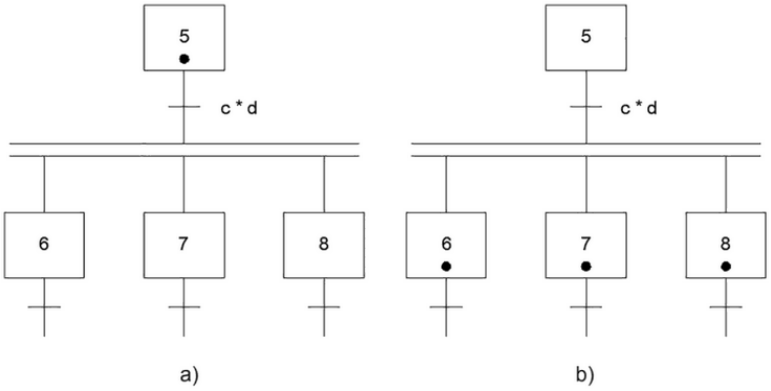


Figura 1.4: Secuencias simultáneas (a) Situación inicial. (b) Situación tras cumplirse la transición [9].

1.1.4.3 Sincronización

La sincronización de etapas hace referencia a una convergencia de las etapas simultáneas, pues para que la transición que se encuentra después de la línea doble se cumpla, todas las etapas simultáneas deben estar activas (Figura 1.5). Normalmente estas últimas etapas no tienen ninguna

acción asociada, pues los procesos simultáneos pueden realizar acciones que no necesariamente se realizan en el mismo tiempo, por lo que estas últimas etapas permanecen vacías y se denominan etapas de espera, pues así se garantiza que todos los procesos simultáneos se encuentren finalizados.

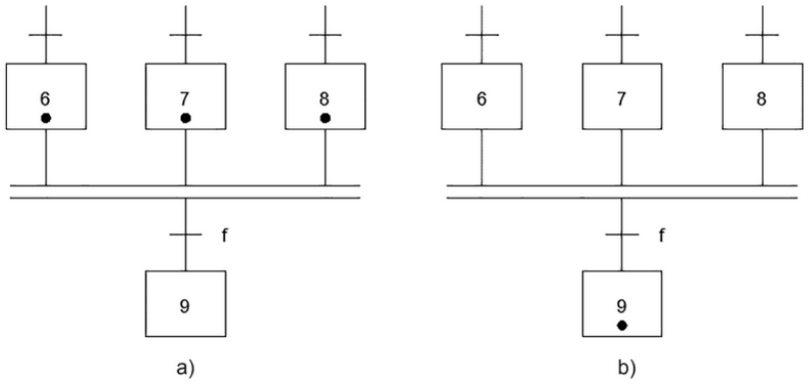


Figura 1.5: Sincronización de etapas (a) Situación inicial. (b) Situación tras cumplirse la transición [9].

1.1.4.4 Programación en lenguaje Escalera

El Grafcet brinda una forma de entender el comportamiento del proceso automatizado de forma gráfica, y es una herramienta bastante útil para poder realizar el programa en lenguaje Escalera (*LADDER*) de un proceso, pues cada etapa se puede traducir en una marca dentro de la programación, mientras las transiciones se traducen a las entradas que recibe el autómata y las acciones son las salidas que este genera.

Normalmente, se hacen uso de las marcas del autómata para representar las etapas en las que se encuentra el Grafcet, por ejemplo, para representar en lenguaje Escalera la secuencia en bucle mostrado en la Figura 1.3, se necesitan de 4 marcas, las cuales pueden ser M0.0, M0.1, M0.2 y M0.3, estas representan cada una uno de los estados del Grafcet. La marca M0.0 comienza siempre estando activa, pues es el estado inicial, y sólo se desactiva en caso de que se cumpla la transición (en este caso, I0.0) que active la siguiente marca, indicando que se ha realizado un cambio de etapa, tal como se muestra en la Figura 1.6.

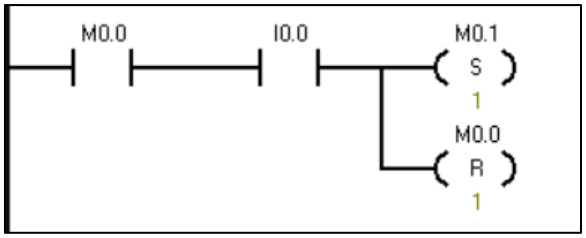


Figura 1.6: Transición de etapas en lenguaje Escalera.

Este mismo esquema de programación se repite realizando los cambios correspondientes de cada marca y transición para cumplir con los datos que presenta el Grafcet.
Las marcas sirven como un enclavamiento una vez que se cumpla una transición, por esto es que se mantienen activas en todo momento y sólo se desactivan al cumplirse una nueva transición. Para añadir salidas a estas marcas simplemente en otro bloque del programa se activan las salidas del PLC que se encuentren adjuntas a cada etapa (Figura 1.7).

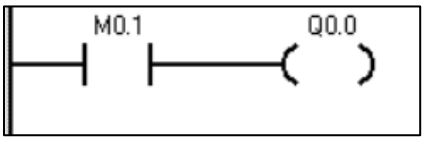


Figura 1.7: Salidas de la etapa.

1.1.5 Guía GEMMA

La Guía para el Estudio de los Modos de Marcha y Paro (GEMMA) presenta un enfoque de diseño estructurado con el fin de modelar, de manera parcial, las tareas de un sistema automatizado, esto para disminuir la complejidad de los factores que intervienen en la automatización de procesos [4]. En el diseño estructurado aparecen tres módulos:

- Módulo de seguridad
- Módulo de modos de marcha
- Módulo de producción

En la Figura 1.8 se muestra la jerarquía de estos módulos, en donde se observa que el módulo de seguridad es prioritario sobre los otros dos, ya que este comprende situaciones de fallos de emergencia, producción defectuosa o fallo en los dispositivos. Estos son tratados mediante el

Grafcet de seguridad que comprende diagnósticos de fallas, paradas de emergencia y solución de problemas.

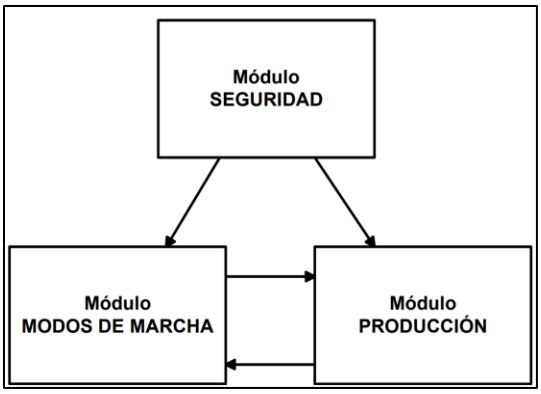


Figura 1.8: Estructura modular del diseño de sistemas [4].

El módulo de producción comprende el control básico en condiciones idóneas del proceso. En este se encuentra el Grafcet de producción, el cual contiene las operaciones básicas que el sistema debe realizar. En el módulo de modos de marcha, el operario vigila el Grafcet de producción mediante el Grafcet de conducción, el cual consta con todos los botones e indicadores de los tipos de marcha y paro que tiene el sistema [4]. La guía GEMMA permite enlazar estos módulos mediante una secuencia de estados y transiciones.

Los estados de funcionamiento del sistema automatizado se agrupan según el modo de funcionamiento que realicen y se integran tres grupos:

- **A** - Procedimientos de paradas
- **F** - Procedimientos en funcionamiento
- **D** - Procedimientos en defecto

En la Figura 1.9 se muestran todos los estados y las distintas transiciones que se pueden realizar entre ellos. En el grupo F se encuentran los estados cuyo funcionamiento está orientado a que el sistema se prepare para producir, verificar el funcionamiento correcto y finalmente, que el sistema trabaje de manera automática. El grupo A comprende todos los modos en el que el sistema se puede detener, desde el estado inicial, hasta paradas a mitad de la producción o a final de cada ciclo. Finalmente, en el grupo D se encuentran los modos en los que el sistema se encuentra en defecto,

ya sea que esté en producción, en paro o en fase de diagnóstico y tratamiento de los defectos [4]. La implementación de todos los estados de la guía GEMMA no siempre es obligatoria para realizar un proceso automatizado, pues cada proceso tiene sus requerimientos específicos que pueden satisfacerse sin la necesidad de implementar todos los estados.

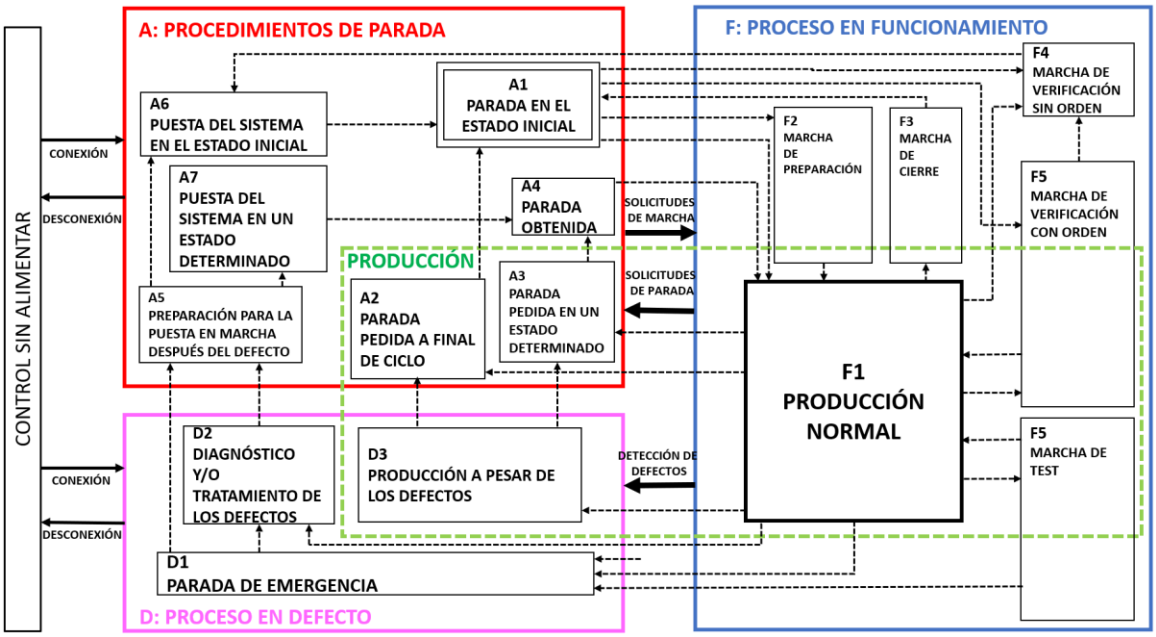


Figura 1.9: Todos los estados previstos en la guía GEMMA [4].

1.1.6 RS-485

El estándar de bus RS-485 es uno de los diseños de bus de capa física de comunicación bastante utilizada en aplicaciones industriales y de instrumentación. Es un método de transmisión de información entre múltiples sistemas, normalmente a largas distancias. Las características principales del RS-485 son [10]:

- Enlaces de larga distancia: hasta 1000 metros.
- Comunicación bidireccional a través de un sólo par de cables trenzados.
- La transmisión diferencial aumenta la inmunidad al ruido y reduce las emisiones de ruido.
- Se pueden conectar múltiples controladores y receptores en un mismo bus.
- Permite velocidades de transmisión de datos de hasta 10 Mbps (megabits por segundo).

En la comunicación de la norma RS-485, el emisor opera el uno lógico a un voltaje de -1.5 a -5 Volts, mientras que el cero lógico a la entrada del receptor opera en el rango de +0.2 a +12 Volts. La máxima tensión aplicada en la línea de salida es de -7 a +12 Volts [11]. El alcance de la transmisión está dado por la relación existente entre el volumen de los datos a transferir y el tiempo de la señal en la portadora determinado por la velocidad de transferencia.

1.1.7 Interfaz Punto a Punto (PPI)

PPI es un protocolo de SIEMENS usado para los PLC CPU S7-200. El protocolo es maestro esclavo, donde el maestro envía peticiones a los esclavos y estos responden. Los esclavos no inician mensajes, sino que esperan a que un maestro envíe una instrucción o solicite una respuesta. En el protocolo no se limita el número de maestros que se pueden comunicar con un mismo esclavo, sin embargo, la red no puede comprender más de 32 maestros [12].

1.1.7.1 Red PPI

Una red PPI se caracteriza por su estructura simple y eficiente, donde dos dispositivos se conectan directamente entre sí para el intercambio de datos, sin necesidad de pasar por un nodo central. Esta red se basa en las redes *token ring* (redes de anillo con testigo circulante), donde el maestro pasa el control de la red haciendo circular un *token* o testigo en el orden de las estaciones que componen la red. El testigo le permite a una estación transmitir datos y solo una estación a la vez puede tener el testigo [12]. Esto se observa en la Figura 1.10, donde cada estación está conectada con otra en serie, sin ningún nodo central de por medio.

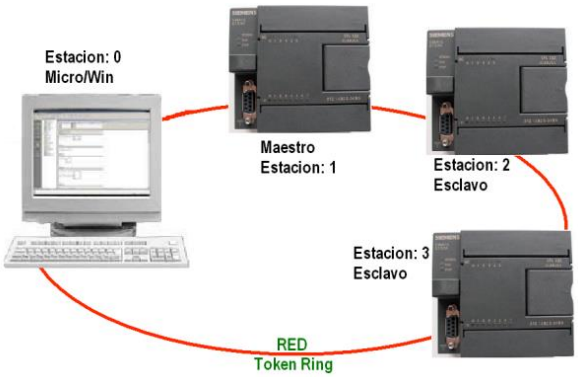


Figura 1.10: Red Token Ring para el protocolo PPI de SIEMENS [12].

1.1.7.2 Conexión de la red

Los puertos de comunicación del PLC S7-200 son compatibles con el estándar RS-485 vía un conector D sub miniatura de nueve pines, conforme al estándar PROFIBUS [13]. SIEMENS ofrece dos tipos de conectores de bus que permiten conectar distintos dispositivos a una red: un conector de bus estándar y un conector que incorpora un puerto de programación, permitiendo que se pueda conectar una PC o una HMI (Interfaz Hombre-Máquina) a la red. Ambos conectores poseen juegos de tornillos para fijar los cables de entrada y salida. Asimismo, disponen de interruptores para polarizar y cerrar la red de forma selectiva. En la Figura 1.11 se muestran ambos tipos de conectores.

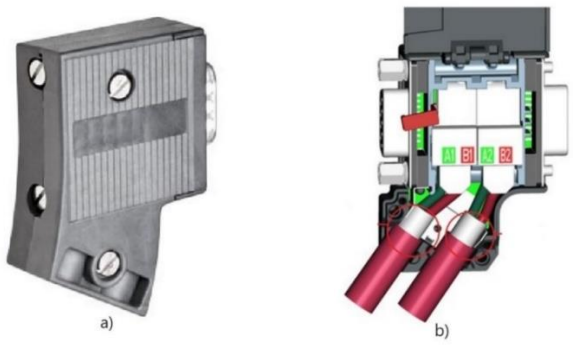


Figura 1.11: (a) Conector de bus estándar [14]. (b) Conector con puerto de programación [13].

1.1.8 Redundancia

En procesos industriales, el concepto de redundancia se refiere a la inclusión de un componente que ya se tiene dentro de un sistema, el cual no es indispensable incluirlo para su funcionamiento adecuado, sin embargo, este segundo componente redundante permite retomar el trabajo del componente original en caso de que éste presente un fallo de tal manera que, el sistema no sufra contratiempos en su funcionamiento pudiendo continuar su operación de manera normal sin que sea necesario parar el proceso mientras se repara o sustituye el componente original que falló.

1.1.8.1 Redundancia *hot-standby*

La redundancia *standby* es una técnica altamente aplicada para lograr una alta confiabilidad y tolerancia al fallo en sistemas industriales, aeroespaciales, de energía, telecomunicaciones, entre otros [17]. Existen tres tipos de técnicas de redundancia *standby*, estas son *hot*, *cold* y *warm*. En un sistema *hot-standby*, existe un elemento operando en línea y al mando del control del proceso, mientras que un segundo elemento opera al mismo ritmo y bajo las mismas condiciones de estrés que el principal, así en caso de que el elemento principal falle, el segundo se encuentra listo para tomar el mando en cualquier momento, reduciendo significativamente los tiempos muertos del proceso. Por otro lado, en un sistema *cold-standby*, existe un elemento en espera que se mantiene apagado y no realiza ninguna operación, es decir, que nunca presentará una falla y presenta un costo casi inexistente, sin embargo, el tiempo de las reparaciones y la configuración es normalmente largo en cuanto este elemento debe reemplazar el elemento operando en línea debido a una falla. Un sistema *warm-standby* presenta un caso intermedio entre los dos tipos anteriores, donde un elemento en espera está expuesto a cierta carga de trabajo y estrés, pero no tanta como el elemento operando en línea. En caso de falla, los tiempos de reparación y configuración son relativamente menores que los de un sistema *cold-standby*.

1.2 Trabajos relacionados

En este apartado se detallan algunos trabajos relacionados con la aplicación de la redundancia y la redundancia ***hot-standby*** en distintos procesos.

1.2.1 Arquitecturas redundantes

La redundancia ha sido utilizada numerosas veces como forma de aumentar la confiabilidad de distintos procesos o sistemas. Por ejemplo, en [15] se describe un método para realizar e implementar la redundancia de unidades en un simulador de tiempo real. Para interactuar con el simulador y las entradas y salidas del sistema se emplea un PLC que garantiza una implementación de configuración más rápida y la automatización de pruebas, además que el sistema cuenta con un panel de interfaz de usuario. En la investigación se implementan distintos casos de prueba que evalúan el rendimiento del accionamiento en diferentes condiciones de funcionamiento. En la

Figura 1.12 se muestra el diagrama propuesto de una unidad completamente redundante (FRD, por sus siglas en inglés) donde se puede observar que existen dos configuraciones exactamente iguales capaces de controlar el motor síncrono. Con esto, se consiguió un método eficiente y rápido de probar y evaluar el desempeño de la unidad.

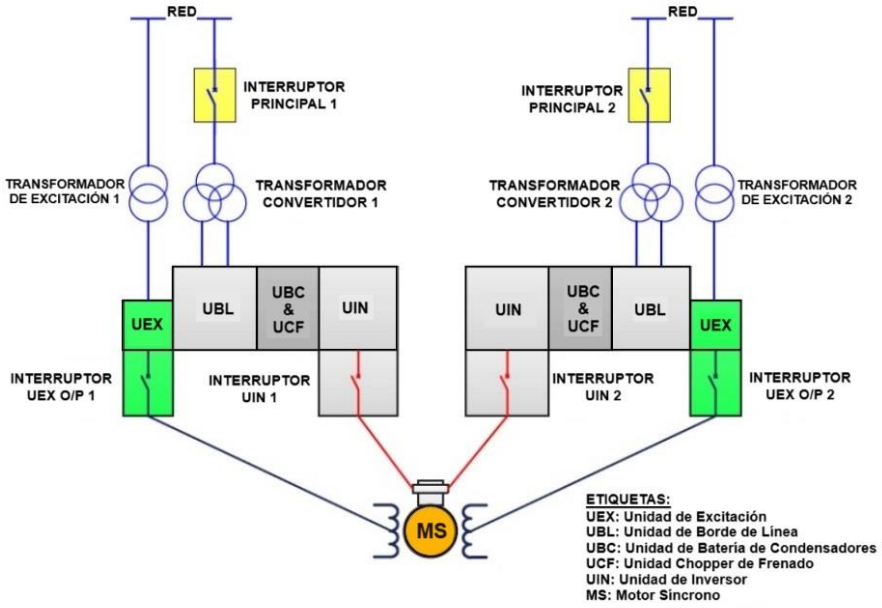


Figura 1.12: Diagrama de la configuración FRD [15].

Siguiendo con las arquitecturas redundantes, en [16] se introduce un nuevo concepto, los PLC de seguridad, una herramienta que considera apropiada para implementación de sistemas de control relacionados con la seguridad o SRCS, por sus siglas en inglés (*Safety-Related Control Systems*). Las partes básicas del PLC de seguridad son los sensores, el módulo F-I (módulo de entrada a prueba de fallos), la F-CPU (unidad central de procesamiento a prueba de fallos), los módulos F-DO (módulo de salida digital a prueba de fallos) y los actuadores (Figura 1.13). Debido a las funciones de seguridad implementadas por el SRCS, se pueden utilizar contactores en lugar de actuadores.

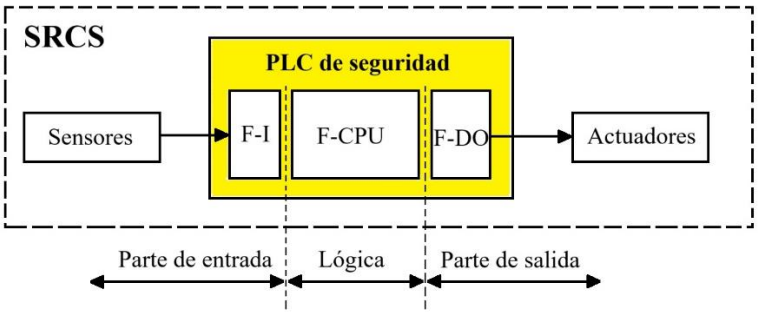


Figura 1.13: Partes básicas del SRCS con PLC de seguridad [16].

Los SRCS con PLC de seguridad se pueden dividir en aquellos que implementan la redundancia y aquellos que no. La redundancia se puede aplicar al nivel de los actuadores, es decir, en las salidas del PLC, pero también existen configuraciones para aplicar la redundancia al nivel de los módulos F-DO e incluso para aplicar una redundancia en ambos niveles. Esto disminuye la probabilidad de entrar a un estado de falla en el PLC.

1.2.2 Redundancia *hot-standby*

En [18] se propone un diseño con dos PLC trabajando en modo redundante como núcleo de control para un sistema inteligente de supervisión y control de refrigeración. Ambos PLC trabajan en modo *hot-standby* para la CPU redundante. Las entradas y salidas son preferentemente controladas por el PLC1, mientras que el PLC2 se encuentra en espera. Si el PLC1 falla en controlar las entradas y salidas (E/S) más de tres veces en el ciclo de *scan*, las E/S aceptarán el control del PLC2. Ahora, si ninguno de los PLC controla las E/S, estas mantendrán el estado del ciclo anterior. En la Figura 1.14 se muestra el diagrama de flujo de la supervisión redundante del PLC doble.

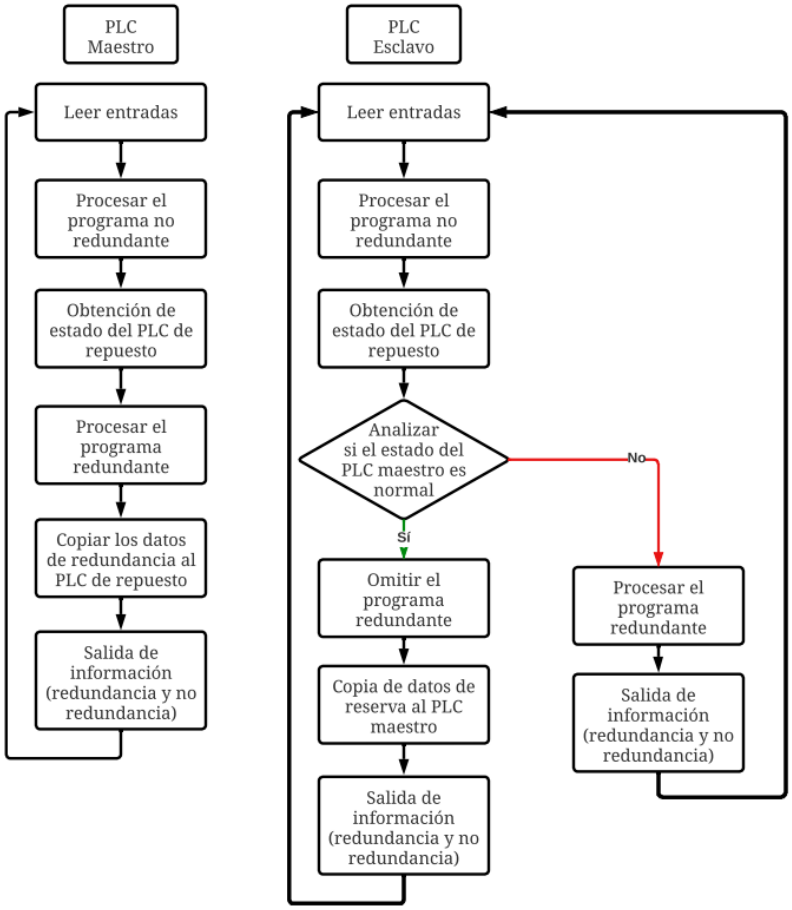


Figura 1.14: El proceso de software de la supervisión redundante del PLC doble [18].

Tomando los PLC como núcleo de control, el sistema inteligente de supervisión y control de los refrigeradores combina la tecnología de los sensores para leer en tiempo real los parámetros de funcionamiento del transformador, realiza el control inteligente y el diagnóstico de fallas en el sistema. Los PLC supervisan en tiempo real el estado de funcionamiento del dispositivo de refrigeración del semiconductor para garantizar que el sistema sea seguro y fiable incluso en las condiciones ambientales más complicadas.

En [19] se presenta una arquitectura de red PROFIBUS doblemente redundante que evita la pérdida de control del proceso en sistemas tolerantes al fallo en modo *hot standby*. Aquí, se utilizan dos redes PROFIBUS independientes para conectar el rack principal del CPU con los racks de E/S principales y redundantes. Esto garantiza que incluso si falla una red, la otra red pueda seguir manteniendo la comunicación entre el PLC y los dispositivos de E/S. Además de esto, se establece redundancia en la adquisición de datos de los esclavos PROFIBUS DP (*Decentralized Peripheral*),

significando que, si se producen fallos en las interfaces de comunicación, la arquitectura redundante pueda adquirir los datos necesarios de estos esclavos. Esta configuración permite que el sistema pueda tolerar múltiples fallos simultáneos en las redes PROFIBUS. En la Figura 1.15 se muestra el modelo de la arquitectura de la red. Esta utiliza las redes PROFIBUS MAESTROS DP 1 y 3 para acceder a las E/S principales, mientras que los racks de E/S principales y redundantes se conectan a los dispositivos de E/S correspondientes. El rack principal de la CPU se comunica con el rack principal de E/S a través del módulo de interfaz 1 (MI 1), mientras que el rack de CPU redundante se comunica con el rack de E/S redundante a través del módulo de interfaz 2 (MI 2). En resumen, el MAESTRO DP 3 es redundante del MAESTRO DP 1. De manera similar, el MAESTRO DP 4 es redundante del MAESTRO DP 2. El rack principal del CPU (RACK-0) se comunica con las principales E/S a través de los respectivos racks de E/S en la red.

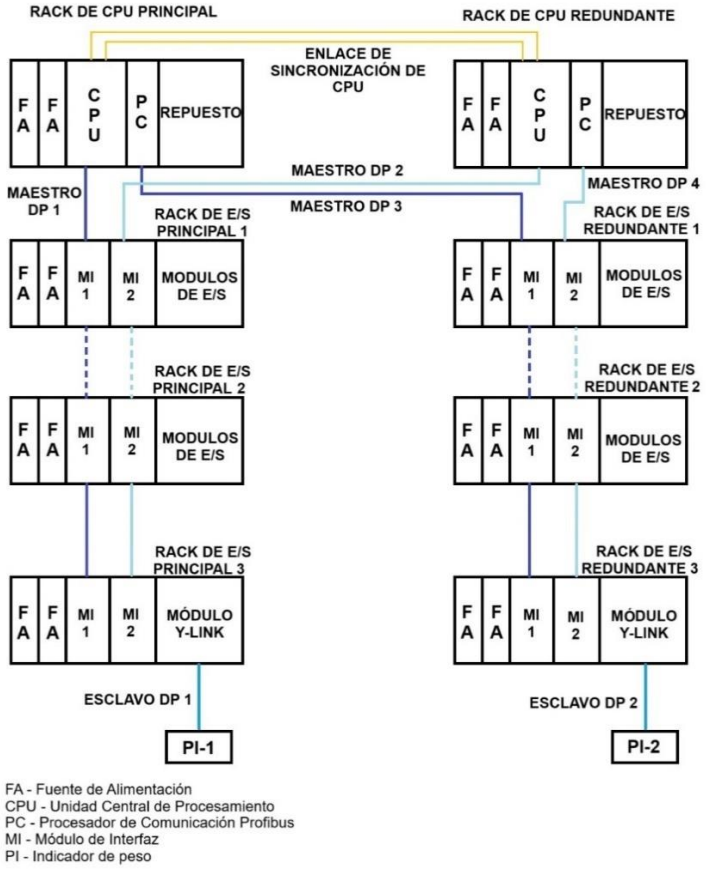


Figura 1.15: Arquitectura doblemente redundante de redes PROFIBUS [19].

Esta arquitectura proporciona vías de comunicación confiables y tolerantes a fallos en la

comunicación PLC-E/S. El sistema puede tolerar tres fallos simultáneos de diferentes componentes en la configuración general, garantizando la disponibilidad del sistema para ser controlado o monitoreado sin problemas.

El diseño de un sistema de control para una central eléctrica de la marina basado en tecnología redundante se presenta en [3]. Este es un sistema que requiere de gran confiabilidad y seguridad. Se utilizan dos PLC de la gama S7-1200, el PLC maestro y el esclavo se configuran como distintas fuentes de alimentación, se usa un mismo módulo CPU y módulos de E/S. Aparte existen tres PLC encargados de adquirir los datos y enviarlos al maestro y al esclavo, los cuales se encargan del procesamiento de la información y la intercambian entre ellos. Normalmente, el PLC maestro es el encargado del control automático respondiendo a las instrucciones de entrada de la pantalla táctil, mientras que el PLC esclavo permanece en modo *hot-standby*, es decir, no puede escribir en sus salidas a pesar de recibir las mismas entradas y básicamente tener el mismo programa en su interior, con el cambio de que este esclavo siempre está monitoreando al PLC maestro. En caso de que el maestro falle, el esclavo obtiene el control sobre el sistema y sobre todas las tareas de monitoreo y salidas, previniendo que el sistema se detenga. La Figura 1.16 muestra una vista del diagrama de control del sistema, mientras que la Figura 1.17 muestra el diagrama de flujo del proceso por cada ciclo del PLC esclavo.

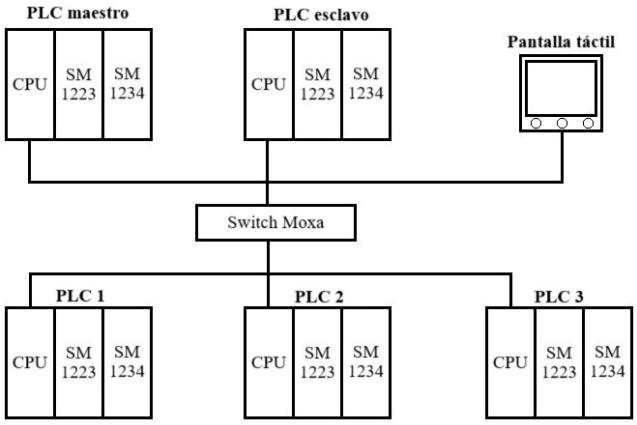


Figura 1.16: Diagrama del sistema de control [3].

Para que el esclavo detecte la falla del maestro hay dos condiciones, la primera es monitoreando la salida Q0.1 del maestro, la cual siempre debe estar en un uno lógico para indicar un funcionamiento

normal, en caso de presentar un cero lógico significa una falla en el PLC maestro. La segunda condición es mediante una señal pulso PWM, pasando de cero a uno cada 30 ms, entonces si el esclavo deja de recibir esta señal por más de tres ciclos lo considera una falla en el PLC maestro y procede a tomar el control del sistema.

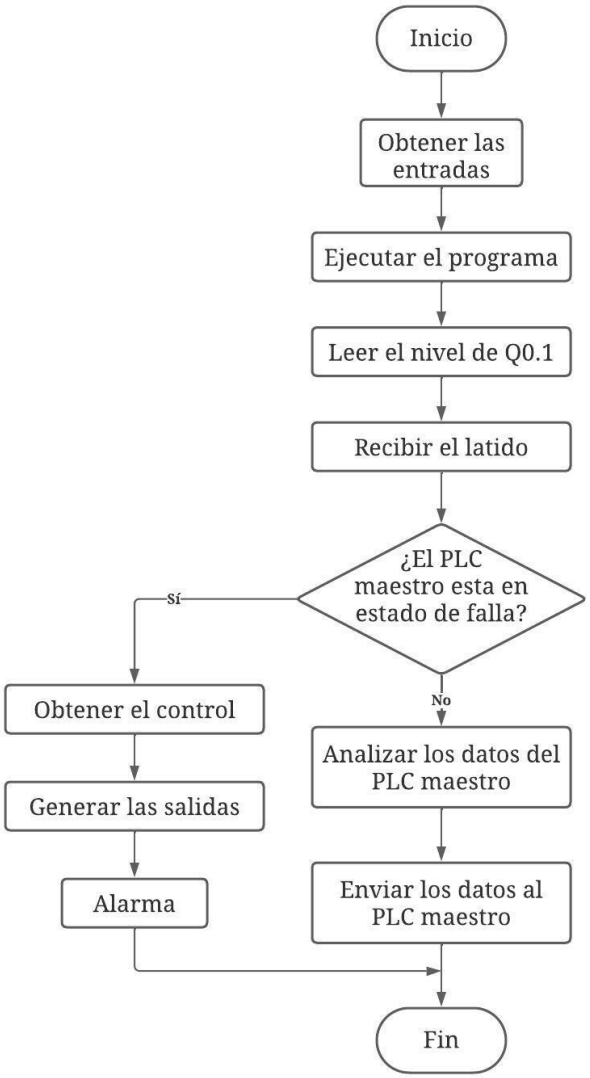


Figura 1.17: Diagrama de flujo del PLC esclavo [3].

En cuanto a la sincronización del proceso, los dos PLC cuentan con el mismo código y realizan simultáneamente las mismas tareas de tal forma que, al final de cada instrucción se evalúa que el PLC maestro siga funcionando, de esta forma se espera que en caso de falla el PLC esclavo tome el control de las salidas justo en la parte del programa donde se quedó el PLC maestro.

1.3 Planteamiento del problema

Los sistemas de control convencionales son diseñados para situaciones en las que el sistema se desempeña libre de fallas, sin considerar que estas puedan llegar a ocurrir [2]. En la realidad, los dispositivos de control, tales como los PLC pueden resultar averiados parcial o totalmente, necesitando mantenimiento o quedando fuera de servicio, ocasionando desde interrupciones en la producción, tiempos largos de reparación hasta situaciones más críticas, como accidentes, dependiendo del tipo de proceso industrial involucrado. Esto representa un problema, sobre todo en sistemas donde la seguridad es crítica, pues incluso una falla menor en un componente puede desencadenar situaciones de gran magnitud. Esto también representa problemas en la parte de la confiabilidad del sistema, la cual es el conjunto de atributos que permite cuantificar la calidad del servicio prestado y el grado de confianza que el usuario puede depositar en el sistema [20]. Por consiguiente, se requieren estándares de fiabilidad, seguridad y capacidad de tolerancia a fallas excepcionalmente elevados para garantizar el funcionamiento seguro y eficiente de estos sistemas.

El PLC S7-200 de SIEMENS puede presentar errores fatales y no fatales durante su tiempo de funcionamiento. Los errores no fatales normalmente tienen que ver con la estructura del programa, que por lo general no detendrá el funcionamiento del PLC, sino que almacena el error en marcas especiales. Por otro lado, existen los errores fatales, aquellos que sí detienen la ejecución del programa, cambiando a modo "*STOP*", encendiendo el *LED* de falla del sistema en color rojo y desactivando las salidas. El PLC permanecerá deshabilitado hasta que se elimine el error fatal [7]. Eliminar este error puede ser tan sencillo como desconectar y conectar de nuevo el autómata o realizar un borrado completo de la memoria del dispositivo en caso de que el error se encuentre en la programación del usuario (como desbordamiento de registros dentro de la memoria del PLC). Sin embargo, también puede ser tan complejo como presentar una falla en el *hardware*, como componentes averiados (relevadores, fusibles, *chips*, transistores), ruido eléctrico proveniente de elementos de potencia, corrientes de fuga inversa o variaciones de voltaje en la alimentación [21], en donde se necesita un tiempo para desconectar el dispositivo, abrirlo y realizar las reparaciones correspondientes.

1.4 Justificación

Un sistema de control redundante permite incrementar la disponibilidad de un proceso industrial [22] evitando que este se detenga en caso de que se deba solucionar un problema manual con el *hardware* del controlador principal. Además de esto, vuelven más confiables los procesos al asegurar que los datos de operación de estos no se perderán en caso de falla del controlador principal, pues muchos procesos requieren exactitud de tiempos y movimientos para evitar paros o interrupciones en la línea de producción [23].

A pesar de que existen dispositivos que implementan la redundancia de una manera interna, estos muchas veces llegan a tener un costo elevado, por lo que se propone una forma de realizar el sistema redundante utilizando dos PLC de la gama S7-200 conectados entre sí mediante una red PPI. Esto debido a la alta disponibilidad y uso de este tipo de PLC en industrias pequeñas, permitiendo garantizar la confiabilidad de los sistemas industriales que manejan.

Se propone un prototipo que sirva como representación de un proceso industrial tipo transfer circular con la finalidad de facilitar el entendimiento al momento de aplicar un control redundante en este tipo de procesos, pues estos muchas veces son complejos y requieren de la simultaneidad del programa de control de cada estación, así como el uso de variables de los contadores y temporizadores. Así, cuando el sistema se encuentre en funcionamiento, se inducirán dos tipos de falla, la primera siendo una desconexión total del PLC principal, para poder observar que el proceso continúa su funcionamiento sin detenerse con el mando del PLC secundario sin perder ningún tipo de dato. La segunda falla que se contempla es un congelamiento en las salidas del PLC principal, esto considerando un error en los relevadores de salida de este mismo, como el fin de su vida útil. Esta falla le dará control de mando al PLC secundario, activando una alarma en caso de que el PLC principal continúe encendido, pues se necesita que este cambie a modo "*STOP*", ya sea de manera automática al ocurrir el error o manual, para que el secundario pueda continuar el proceso por su cuenta.

1.5 Hipótesis

Un sistema redundante *hot-standby* que utiliza dos PLC del fabricante SIEMENS S7-200 CPU 224 conectados en una red PPI (Interfaz Punto a Punto) es capaz de soportar dos tipos de fallas en el PLC principal, de manera que, al desconectarlo o se produzca una avería en sus salidas tipo RLY, entre en acción el PLC secundario para que el proceso continúe operando de manera normal.

1.6 Objetivos

A continuación, se redactan los objetivos, tanto general como específicos, que se tienen planeado alcanzar con el desarrollo de este proyecto.

1.6.1 Objetivo general

Desarrollar un sistema redundante a prueba de fallas de tipo *hot-standby* compuesto por dos micro-PLC S7-200 del fabricante SIEMENS aplicado a un proceso de tipo transfer circular para observar la continuidad de operación de este proceso.

1.6.2 Objetivos específicos

- Aplicar una metodología genérica para el diseño del sistema automatizado.
- Construir el prototipo físico del proceso tipo transfer circular.
- Generar el programa en el *software STEP 7 MicroWIN* ® por medio de diseño estructurado y el método de coordinación de tareas.
- Diseñar el panel de mando del sistema.
- Validar, redactar y presentar resultados.

1.7 Estructura de la tesis

La tesis se encuentra dividida en cinco capítulos, las referencias bibliográficas y los anexos. A continuación, se describe el contenido de los cinco capítulos y un breve resumen de los mismos.

El Capítulo 1 contiene la introducción a este trabajo, así como la teoría básica necesaria para la comprensión del sistema. Se explica en que consiste la redundancia ***hot-standby*** y se muestran ejemplos de arquitecturas redundantes. Finalmente, se redacta el problema, la justificación, hipótesis y objetivos del trabajo.

El Capítulo 2 contiene el marco metodológico utilizado para el desarrollo de este proyecto, pues abarca métodos y técnicas utilizados para el desarrollo de sistemas automatizados, así como algoritmos del comportamiento del sistema.

El Capítulo 3 abarca el desarrollo de la metodología, configurando los algoritmos y adaptándolos a los requisitos del prototipo. Se detalla el funcionamiento del sistema, se muestra el circuito de conexiones y cada una de las estaciones de trabajo del sistema. Dentro de este mismo capítulo se detalla la programación del sistema redundante tipo ***hot-standby***, y la conexión de dos PLC al circuito de control del sistema. También se explica la programación del error de congelamiento en las salidas del PLC principal.

El Capítulo 4 muestra el análisis de los resultados obtenidos del sistema redundante tipo ***hot-standby***. Se muestran las mediciones obtenidas con ayuda de un osciloscopio y se realiza una comparativa de los distintos tiempos de respuesta del sistema.

Finalmente, el Capítulo 5 contiene las conclusiones del trabajo, así como los trabajos futuros que se desprenden de esta investigación.

Capítulo 2

Propuesta Metodológica

En este capítulo se presenta la propuesta metodológica que se desarrolló a lo largo de la tesis. Esto incluye el detallar los métodos y técnicas que se utilizan para abordar los objetivos de investigación previamente planteados. Además, se describen los algoritmos que se implementaron con el propósito de resolver los distintos problemas presentados a lo largo del proyecto, proporcionando una visión clara y estructurada de la estrategia metodológica elegida.

Esta propuesta metodológica establece el camino a seguir durante el desarrollo del proyecto y garantiza que cada etapa del proceso se realice con el suficiente detalle para asegurar la fiabilidad y validez de los resultados obtenidos. De este modo, se busca contribuir al conocimiento existente en esta disciplina proporcionando resultados que sean tanto relevantes como aplicables en contextos reales.

Añadido a esto, se discute la justificación de cada método y técnica seleccionada, destacando su importancia y adecuación al problema de investigación, asegurando que los hallazgos sean lo más robustos y generalizables posible.

2.1 Métodos y técnicas

La metodología a utilizar es el marco metodológico genérico en el que se ubica la guía GEMMA descrito en [4]. En este se describen cinco fases a realizar para llevar a cabo un proyecto de automatización (Figura 2.1). Si la metodología quiere llevarse a la práctica hay que seguir paso a paso el método de forma secuencial. El operario es quien lleva a cabo cada una de las fases y hace la transición entre ellas, para finalmente, hacer una iteración para rehacer el primer ciclo en búsqueda de introducir mejoras [4].

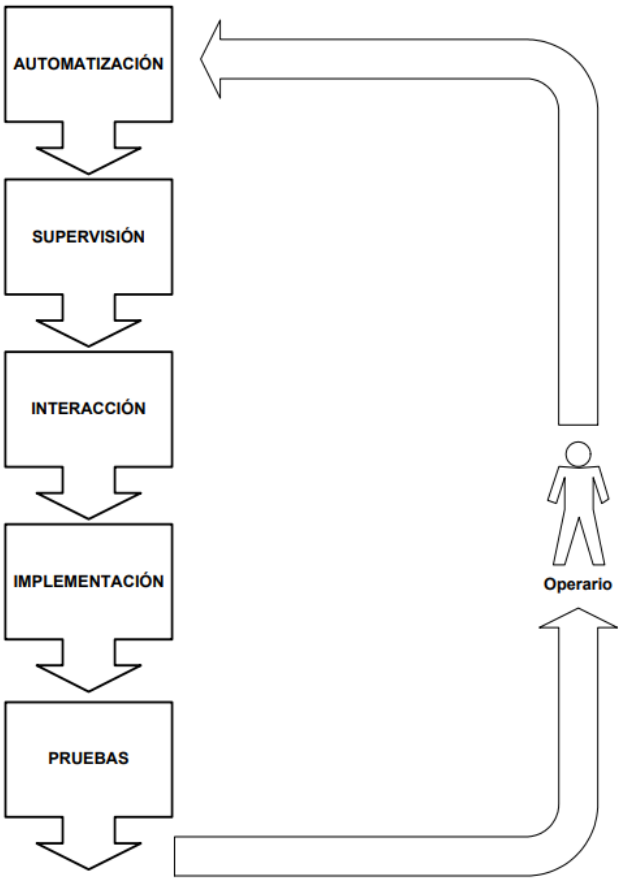


Figura 2.1: Marco metodológico genérico [4].

Las fases del marco metodológico no son conceptos fijos y cada una tiene un grado de profundidad que incluyen distintas técnicas que se pueden aplicar con el fin de recabar información del sistema a automatizar, así como del equipo necesario para llevarlo a cabo. A continuación, se describen

cada una de las fases del marco metodológico.

2.1.1 Automatización

En esta fase se debe de entender al proceso a automatizar y generar el Grafcet de primer nivel, este contiene una descripción global sin mucho detalle del automatismo, lo que permite comprender de manera rápida su función [24]. En la Figura 2.2 se muestra un ejemplo de este tipo de Grafcet, resaltando que no se especifica ningún tipo de actuador, sensor o alguna función del autómata a utilizar.

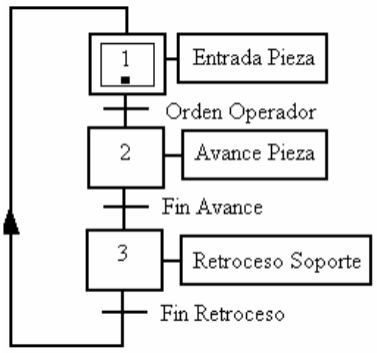


Figura 2.2: Grafcet de primer nivel [24].

Una vez que se tiene un entendimiento general del proceso a automatizar, se selecciona el autómata que se encargará de controlar dicho proceso, así como todas las tecnologías adicionales, como lo pueden ser cilindros, motores, cintas transportadoras, sensores, etc.

Finalmente, se debe realizar el Grafcet de segundo nivel, el cual ya comprende las tecnologías utilizadas para cada función, describiendo las tareas que realiza cada elemento elegido [24]. La Figura 2.3 muestra un ejemplo de cómo se ve el Grafcet anteriormente mostrado en la Figura 2.2 pero ahora en su segundo nivel.

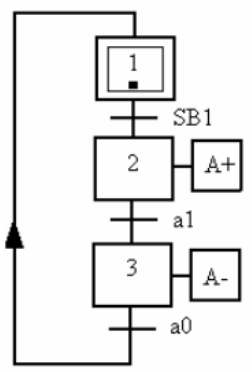


Figura 2.3: Grafcet de segundo nivel [24].

2.1.2 Supervisión

En esta segunda fase de la metodología se despliega a detalle el funcionamiento del proceso, pues se reúnen el máximo de especificaciones sobre los estados posibles en las que se puede encontrar una máquina o proceso [4]. Todo esto basado en la experiencia del operador y en las tecnologías con las que se cuentan, considerando, por ejemplo, el número de entradas y salidas del autómata o las funciones internas con las que se cuentan.

Aquí se definen los módulos a utilizar según la complejidad del problema (módulo de seguridad, modos de marcha y producción) y se representan de manera gráfica mediante los estados y transiciones de la guía GEMMA que describan el funcionamiento específico de este proceso.

Para cada módulo se genera un Grafcet parcial. Para el módulo de producción, sin embargo, ya se han realizado estos Grafcet en la fase de automatización, por lo que realmente hay que establecer en esta fase es la integración de estos en un Grafcet general de producción, generalmente cíclico, donde se puedan tener procesos simultáneos, además de integrar la relación con el resto de módulos [4]. El módulo de modos de marcha promueve la activación y desactivación del módulo de producción y a su vez el módulo de seguridad vigila los dos módulos anteriores ante posibles fallas o paros de emergencia.

Dentro de esta fase se debe considerar el cómo se llevó a cabo la comunicación entre los dos dispositivos para lograr la comunicación constante obtenida.

2.1.3 Interacción

En esta tercera fase se concreta la forma en la que el supervisor humano interactúa con el proceso automatizado. Normalmente es mediante el diseño de un panel de mando o de una HMI en función de las acciones físicas sobre dispositivos y la recepción de señales de información visuales o acústicas [4].

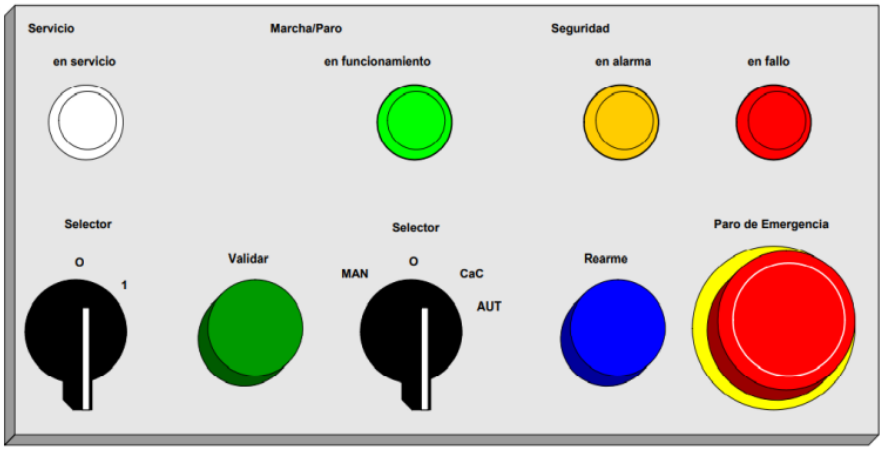


Figura 2.4: Panel de mando [4].

En la Figura 2.4 se observa un panel de mando genérico que está formado por un conjunto de dispositivos visuales indicadores situados en la parte superior del panel, las cuales sirven para indicar el estado de funcionamiento de los distintos módulos que se tienen, mientras que un conjunto de controladores situados en la parte inferior del panel sirve para que el operario interactúe directamente con el sistema.

A continuación, se explica brevemente el funcionamiento de cada uno de los dispositivos que se encuentran en el panel:

Indicadores

- **En servicio:** Indicador luminoso que se enciende para indicar que el proceso se encuentra conectado a su alimentación.
- **En funcionamiento:** Indicador luminoso que se enciende al momento de que el proceso inicia su funcionamiento.
- **En alarma:** Indicador luminoso que se enciende cuando ha ocurrido un defecto dentro del proceso, indicando la necesidad de la intervención del operario.
- **En fallo:** Indicador luminoso que se enciende cuando se ha presionado el paro de emergencia del panel de control.

Controladores

- **Selector 0/1:** Selector que dependiendo de su posición conecta o desconecta el proceso a su alimentación.
- **Validar:** Botón que valida que las condiciones de marcha sean correctas y en caso de ser así, inicia el funcionamiento del proceso.
- **Selector MAN/0/CaC/AUT:** Selector de modos de marcha, puede estar desactivado, indicando una marcha manual (validar cada paso del proceso), Ciclo a Ciclo (validar cada fin de ciclo) y Automática (No se requiere validación después de la primera vez).
- **Rearme:** Botón que coloca todos los elementos del proceso en su posición inicial, generalmente después de alguna falla.
- **Paro de Emergencia:** Botón que detiene por completo todo el proceso. Normalmente es un botón con enclavamiento y normalmente cerrado, con el fin de que en caso de que no funcione, simplemente con cortar el cable de este se pueda activar.

Cabe resaltar que no es obligatorio que todos los procesos cuenten con exactamente la misma configuración en sus paneles de control, pues dependiendo de las limitaciones de entradas y salidas de cada autómata estos se pueden ver reducidos en cantidad de dispositivos o, al contrario, se pueden agregar todo tipo de dispositivos de control o de información.

2.1.4 Implementación

En esta fase, primero se selecciona el lenguaje de programación del automatismo, pues cada uno puede tener una diferente estructura o características especiales. En el caso del PLC S7-200 se utiliza el lenguaje Escalera SIMATIC 27-200.

Una vez elegido el lenguaje de programación, se desarrollan y escriben los códigos en este lenguaje, traduciendo todos los Grafcet correspondientes (Grafcet global). Para realizar esta traducción a lenguaje Escalera se toman en cuenta las reglas previamente detalladas en la sección 1.1.4.4, adaptando todos los estados y transiciones a las entradas, salidas y marcas del autómata.
El programa examina en cada ciclo de reloj las entradas al autómata y en función de las condiciones de activación de cada etapa es que las va activando o desactivando. Las salidas del autómata generalmente están asociadas a las acciones asociadas a cada etapa, las cuales se encienden siempre y cuando su etapa se encuentre activa.

2.1.5 Pruebas

En esta última etapa, una vez que ya se encuentra implementado el algoritmo general sobre el automatismo, el operario puede verificar el funcionamiento de este mismo, ya sea realizando marchas de prueba, o incluso imitando situaciones de emergencia para evaluar el comportamiento del sistema para analizar la respuesta del sistema ante la implantación de la guía GEMMA [4]. En caso de presentar problemas, el operario puede modificar parcialmente los algoritmos o añadir más estados de la guía GEMMA que originalmente se habían ignorado y rehacer el algoritmo en general.

Para afrontar los problemas se recomienda que el programa se encuentre dividido en módulos funcionales básicos y así poder modificar sólo las partes que contienen el error sin afectar el resto del programa [4].

El operario puede volver a cualquiera de las fases anteriores con el fin de corregir errores o añadir elementos que no se consideraron previamente, lo que vuelve este proceso un ciclo y realmente es

cuestión de la experiencia del operario y de las limitaciones a las que se enfrente el cuántos ciclos se realizarán, pues a pesar de que un proceso automatizado se encuentre funcionando perfectamente siempre se deja la puerta abierta a nuevas tecnologías o a expansión de sus funciones sin necesidad de comenzar de nuevo todo el proceso de realización de algoritmos.

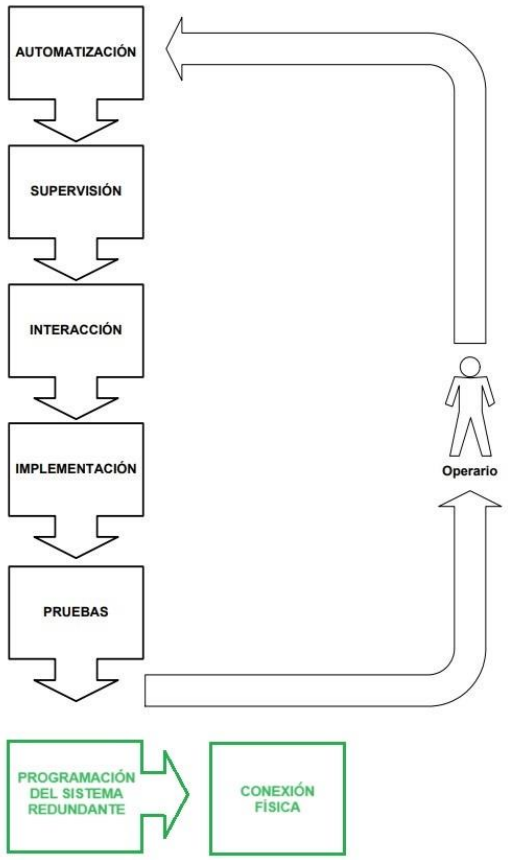


Figura 2.5: Marco metodológico completo [4].

Este marco metodológico es conveniente para la realización completa de un proceso automatizado, sin embargo, para este caso en específico donde se necesita implementar la comunicación entre dos autómatas para realizar un sistema redundante *hot-standby* es preciso añadir dos fases adicionales que pueden encontrarse fuera del ciclo de la metodología principal, pues estas dos fases comprenden técnicas que trabajan sobre el proceso ya realizado dentro del ciclo que plantea la metodología. En la Figura 2.5 se muestra el diagrama del marco metodológico completo, contemplando estas nuevas fases. A continuación, se describen cada una de ellas a detalle.

2.1.6 Programación del sistema redundante

Esta fase se realiza una vez que las pruebas del sistema son satisfactorias y se sabe que el proceso es completamente funcional con un sólo autómata. Es entonces que se debe de agregar al programa la sección de comunicación y transferencia de datos con un segundo autómata.
En el caso del PLC S7-200 se cuenta con dos operaciones principales para la comunicación en una red. La operación Leer de la red (**NETR**) inicia una comunicación que permite leer datos de una estación remota a través de un puerto indicado, mientras que la operación Escribir en la red (**NETW**) permite escribir datos en la estación remota [7].

La operación NETR puede leer hasta 16 bytes de información de una estación remota, mientras que su contraparte NETW puede escribir la misma cantidad en una estación remota. El programa puede contener un número ilimitado de operaciones NETR y NETW, pero sólo ocho pueden estar activas al mismo tiempo, sin importar si son de lectura o escritura [7].

El entorno de programación para el PLC S7-200, *STEP 7-MicroWin*®, cuenta con un asistente de operaciones NETR/NETW que facilita la configuración de estas operaciones (Figura 2.6) y las encapsula en una subrutina que puede ser llamada en cualquier momento dentro del programa, además de permitir la modificación de estas en todo momento, ofreciendo flexibilidad durante el desarrollo del programa y mejorando la eficiencia en la gestión del código.

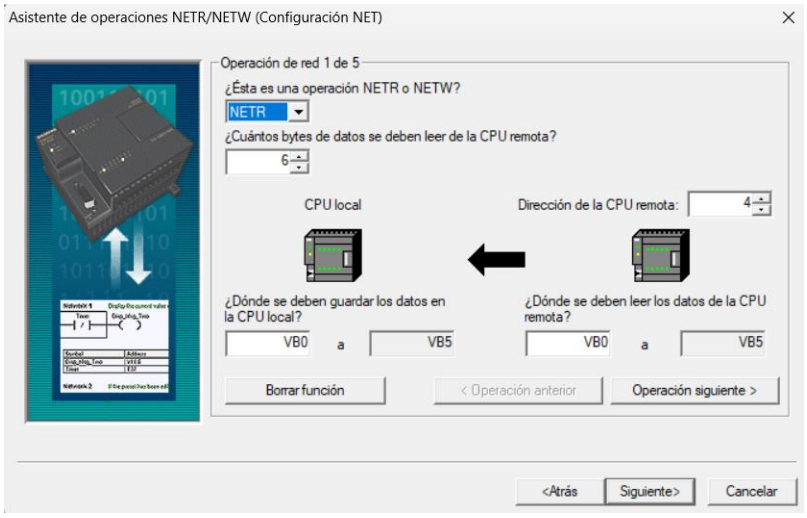


Figura 2.6: Asistente de operaciones NETR/NETW.

Dentro del asistente de operaciones se puede modificar el número de bytes de lectura o escritura, así como la dirección en la que se guardaran dichos datos tanto en la CPU local como la CPU remota. Todas las operaciones se encontrarán encapsuladas dentro de una subrutina con nombre NET_EXE, la cual al ser llamada realizará las operaciones de lectura o escritura que hayan sido configuradas.

Como parte del envío de datos, dentro del programa se hace uso de dos bloques de operación disponibles en lenguaje Escalera del PLC S7-200 (SIMATIC). Estos son las operaciones de conversión y las de transferencia de bytes. La primera operación permite la conversión entre un byte a un entero. Su bloque en lenguaje Escalera se puede ver en la Figura 2.7. Esta operación es especialmente útil para enviar el valor de un temporizador o un contador (valor entero) hacía una dirección en la memoria, como lo pueden ser las direcciones donde se almacenan los datos de la comunicación.

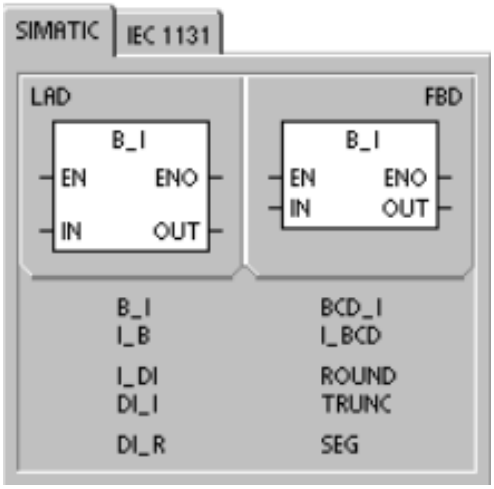


Figura 2.7: Operaciones de conversión [7].

Por otro lado, las operaciones de transferencia de bytes (Figura 2.8) ayudan a transferir directamente el valor de un byte en alguna otra dirección, lo cual puede ser bastante útil para la transferencia de la información del estado activo del programa, pues se puede escribir un número en específico en alguna dirección de byte dependiendo de la etapa que se encuentre activa en el programa.

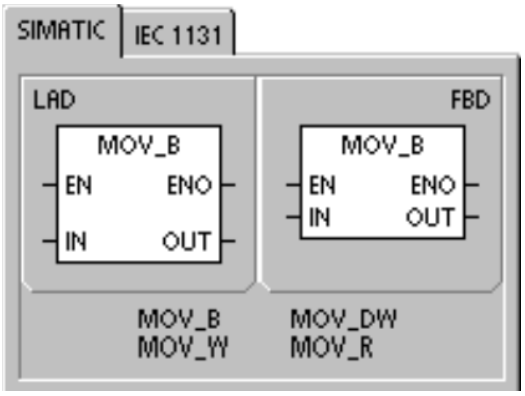


Figura 2.8: Operaciones de transferencia de bytes [7].

2.1.7 Conexión física de los dispositivos

Como última fase, se tiene la conexión física de los dispositivos, donde se debe de buscar la forma de que este segundo autómata se integre al circuito ya existente del proceso. Este proceso debe ser realizado con precisión para garantizar que ambos controladores trabajen de manera coordinada y eficiente.

Es fundamental que los dos autómatas se conecten a las mismas entradas, permitiendo que ambos reciban los mismos datos y señales del entorno del proceso. Además, se debe asegurar que ambos controladores tengan la capacidad de operar y controlar los actuadores del proceso físico.

Una vez conectados los dos dispositivos se puede dar fin al marco metodológico, recalcando que cualquier cambio en el proceso puede ser realizado en cualquier momento regresando a las primeras fases de la metodología. Estas últimas dos fases también pueden ser modificadas, pero únicamente después de que se concreten las pruebas del funcionamiento del sistema con un solo autómata.

2.2 Algoritmos

Los algoritmos que se utilizan en este proyecto comprenden desde diagramas de flujo para explicar el comportamiento de cada ciclo de reloj del PLC hasta la lógica necesaria para realizar la conexión de dos dispositivos que sean capaces de controlar los mismos actuadores.

2.2.1 Comportamiento del sistema

En la Figura 2.9 se muestra el ciclo de reloj del PLC S7-200, indicando que en cada iteración se realiza la lectura de las entradas, la ejecución de todo el programa con las entradas recién leídas, después las operaciones de comunicación, el diagnóstico y finalmente se escriben las salidas del mismo.

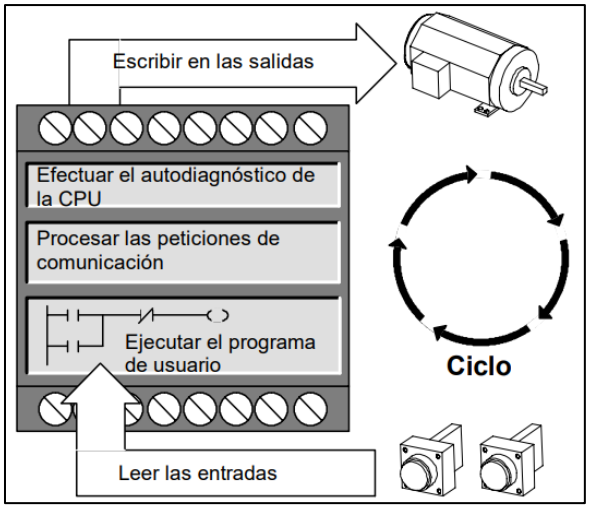


Figura 2.9: Ciclo del PLC S7-200 [7].

Tomando como base este ciclo, se muestra en la Figura 2.10 el ciclo del comportamiento del sistema logrado en este proyecto. Se observan dos diagramas de flujo, una para el PLC 1 o PLC principal, el cual tiene un comportamiento muy parecido al ciclo normal del S7-200, realizando el programa en cada ciclo y recolectando los distintos datos para enviar al PLC 2.

El segundo diagrama de flujo comprende el comportamiento del PLC 2 o PLC secundario, el cual es un poco más complejo, pues además de realizar la lectura de entradas y el proceso del programa, debe de realizar la lectura de datos que el PLC 1 recolectó. También debe de monitorear en todo momento el estado de funcionamiento del PLC1, pues en caso de que este sea desconectado, el PLC 2 debe asumir el control del sistema y activar sus salidas, mientras que si no se ha desconectado el PLC principal se evalúa si ha ocurrido algún congelamiento en sus salidas, pues en caso de ser ese el caso, se activa una alarma y el PLC secundario toma el mando, de lo contrario continúa con sus salidas inactivas y vuelve a comenzar el ciclo.

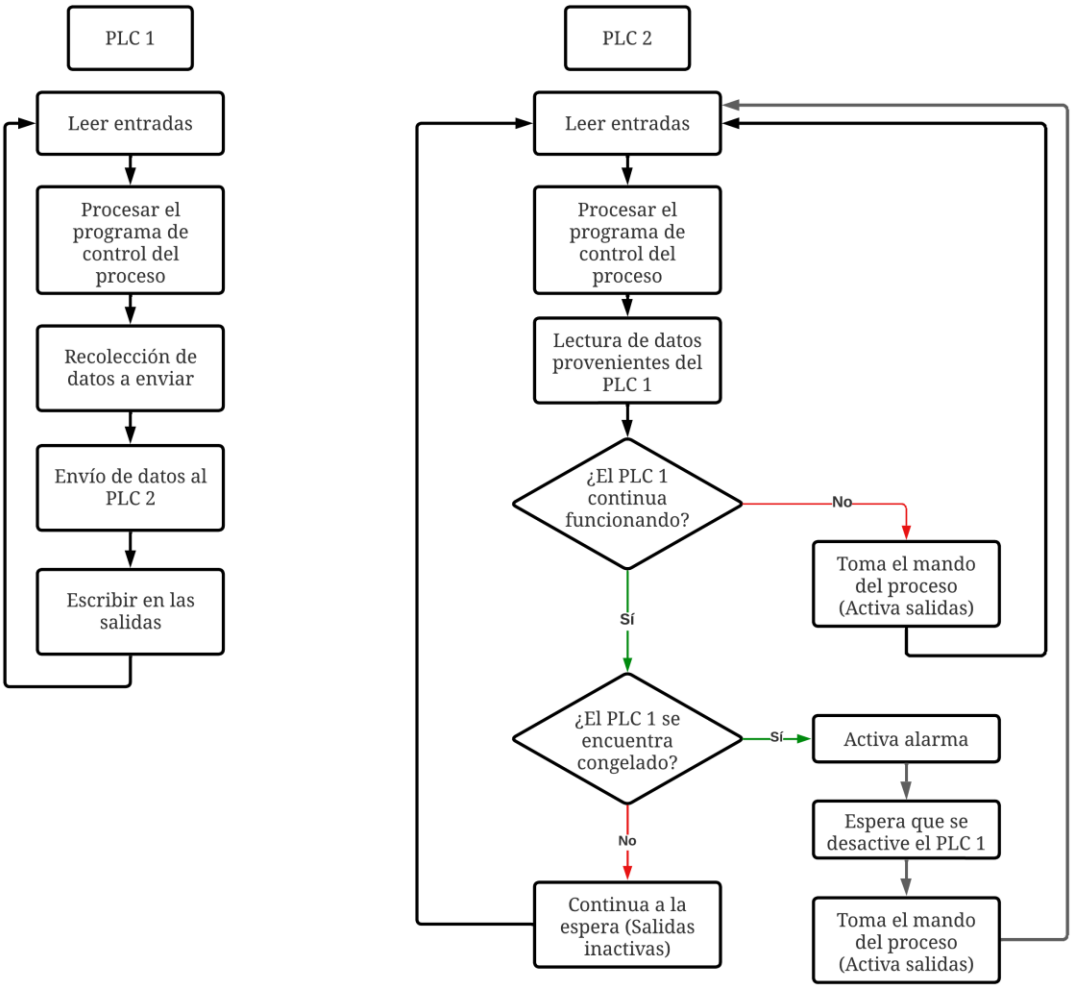


Figura 2.10: Comportamiento del sistema.

Capítulo 3

Diseño del sistema

En este capítulo se presenta el desarrollo del sistema, aplicando la metodología descrita en el capítulo anterior. Se describen los materiales y el cómo fueron utilizados en el desarrollo de un prototipo de un proceso tipo transfer circular, además se presentan las configuraciones de los distintos algoritmos aplicados hacia este caso en particular. La programación el sistema redundante también se presenta al final de este capítulo.

En la Figura 3.1 se puede observar el prototipo de proceso transfer rotatorio al que le fue aplicado el control redundante en este trabajo. Este prototipo se encuentra basado en el puesto de taladro mostrado en [25]. Este cuenta con tres estaciones de trabajo: la estación de carga, la estación de maquinado y la estación de descarga. En la estación de carga, existe una columna a la cual, le son alimentadas piezas cúbicas de dimensiones 5x5x5 cm de colores tanto rojo como verde. La pieza es detectada por un sensor y posteriormente empujada por medio de un cilindro eléctrico hacia un plato rotatorio. Este cilindro es de doble efecto, ya que después de extenderse hasta la distancia donde la pieza se deposita en el plato, este se debe retraer para no obstruir el movimiento del sistema. Al girar el plato rotatorio, la pieza llega a la estación de maquinado en donde se simula el

proceso de realizar una operación en dicha pieza. En este caso, otro cilindro eléctrico se extiende hasta encontrarse con la pieza, para posteriormente volver a retraerse al hacer contacto con la misma. La estación de descarga comprende la inspección y control de algún defecto en la pieza, esto se realiza a través de un sensor de color que detecta el color de la pieza, donde rojo es una pieza defectuosa y verde una pieza correctamente maquinada. Adicionalmente, se incluye un tercer cilindro eléctrico para empujar al centro del plato la pieza mecanizada con éxito. En caso de que la pieza presente una anomalía, esta no será empujada al centro y en la siguiente rotación deberá ser retirada de forma manual. Un servomotor permite la rotación del plato, calibrado en tiempo y ángulo, para posicionar la pieza por debajo de cada herramienta de la estación de trabajo. Este prototipo cuenta con medidas máximas de 733.32 x 364 x 516.66 milímetros.

Este sistema es un prototipo para corroborar el comportamiento correcto del sistema redundante, pues a pesar de contar con actuadores y sensores, realmente no modifica las piezas que entran en él, ya que estas cuentan con un color específico desde que son colocadas con el fin de simular cierta cantidad de piezas defectuosas y correctas. Se descartó el uso de motores para realizar algún tipo de maquinado con el fin de evitar vibraciones en el sistema. La descarga de piezas se debe realizar de manera manual, debido a las limitaciones de entradas y salidas con las que cuenta el PLC S7-200 CPU 224 que controla el proceso.

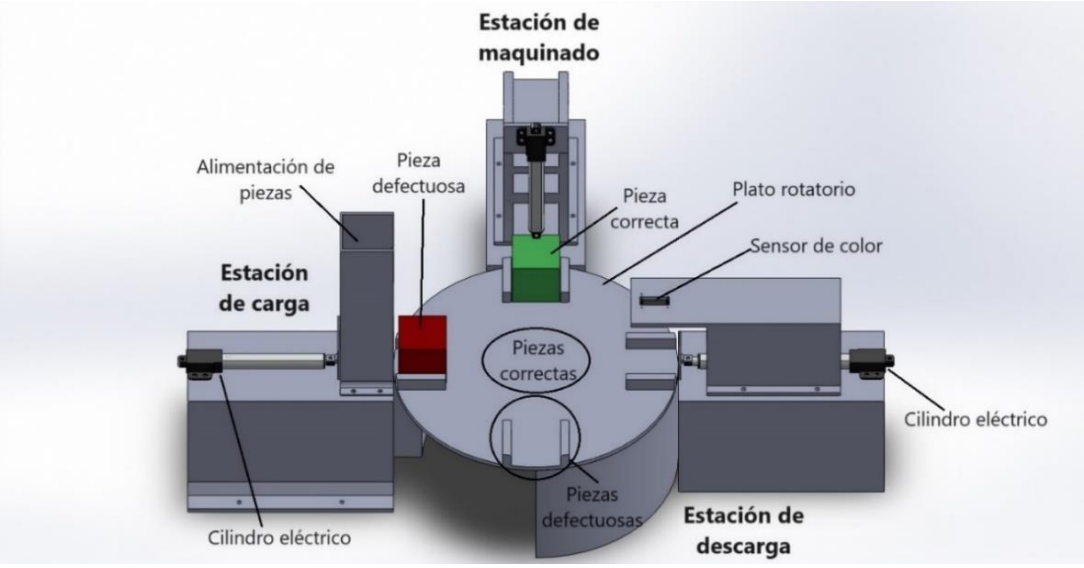


Figura 3.1: Diseño preliminar del prototipo de un proceso tipo transfer circular.

En la Figura 3.2 se puede ver el diagrama de bloques que representa las conexiones necesarias del sistema, donde las entradas de los PLC provienen de botones pulsadores, sensores de fin de carrera, encargados de indicar el fin del movimiento de los cilindros eléctricos, y de una tarjeta Arduino, la cual se encarga de enviar la información del sensor de color. Los dos PLC comparten las mismas entradas, sin embargo, sus señales de salida atraviesan un bloque llamado circuito de control antes de llegar a los puentes H y la tarjeta Arduino. Este circuito permite que dos bloques de salidas distintos puedan controlar los mismos componentes del sistema.

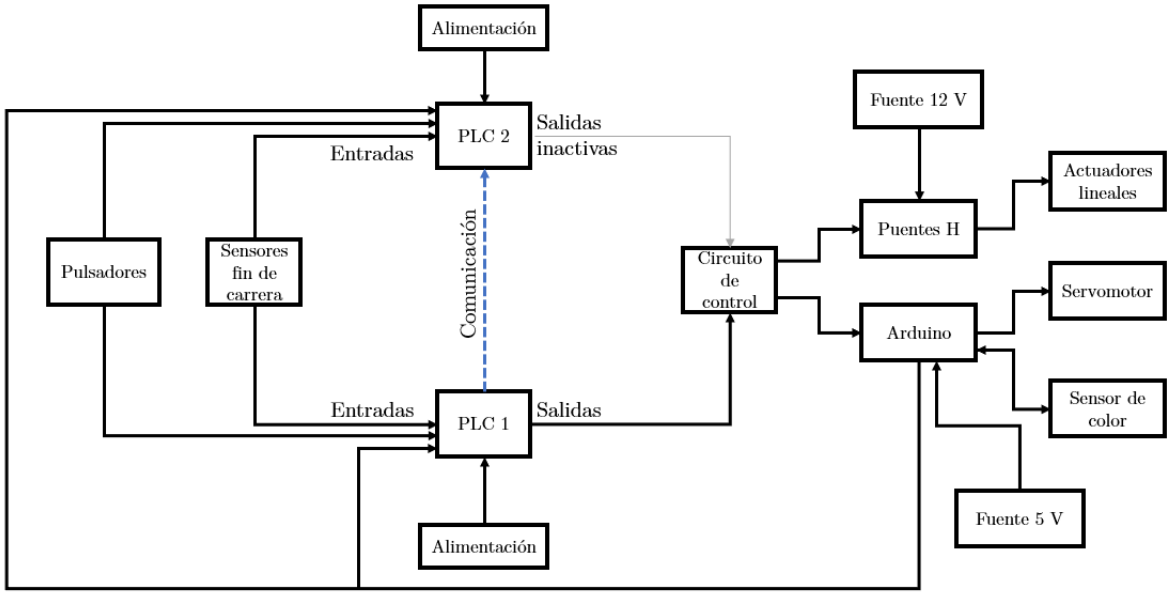


Figura 3.2: Diagrama en bloques del hardware del sistema.

Las salidas de los PLC desembocan en una tarjeta Arduino, encargada de dar giro a un servomotor de 360° el cual mueve el plato rotatorio, así como de realizar el intercambio de información con el sensor de color e identificar si la pieza dentro del sistema es correcta o defectuosa. Las salidas del PLC también se conectan a un módulo de puente H que ayuda a definir la dirección en la que avanzan los cilindros eléctricos, de manera similar a controlar el giro de un motor. En la siguiente sección se detallan cada uno de los componentes utilizados.

3.1 Material y contextos

En esta sección se muestra el equipo utilizado para la construcción del prototipo físico, así como los contextos en los que se utilizan, como lo son el circuito del sistema y el panel de control.

3.1.1 Selección de equipo

A continuación, se describe el equipo que se utilizó para la construcción del prototipo físico de un proceso tipo transfer circular.

3.1.1.1 PLC S7-200 CPU 224

En la Universidad Tecnológica de la Mixteca (UTM) se cuenta con una gama de PLC S7-200 del fabricante alemán SIEMENS, los cuales son de los modelos CPU 214, 222, 224 y 226. Para la realización del presente proyecto de tesis, se optó por el CPU 224 (Mostrado en la Figura 3.3), principalmente por el hecho de que se cuentan con dos unidades de este modelo. Este CPU tiene 10 salidas y 14 entradas digitales, lo que implica tener que reducir entradas y salidas del panel de control en favor del funcionamiento del proceso. Para mayor detalle del equipo consultar la sección 1.1.2.1.1 de este documento.

Figura 3.3: PLC S7-200 de SIEMENS, CPU 224.

3.1.1.2 Micro actuador lineal de motor

En la Figura 3.4 se muestra un actuador lineal eléctrico. Este tipo de actuador se encuentra presente en cada una de las etapas del sistema y cumplen la función de ser el empujador y el expulsor en las estaciones de trabajo. Funcionan con 12V de alimentación y se controlan de la misma forma que un motor de DC, donde al invertir la alimentación se puede invertir el giro; aquí se puede controlar si se extiende o si se retrae el actuador. La velocidad máxima es de 150 mm/seg.

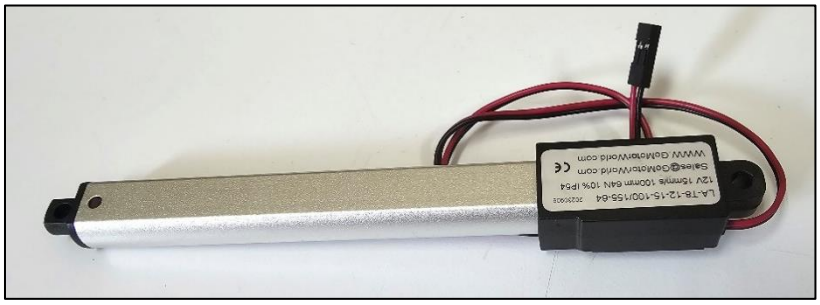


Figura 3.4: Actuador lineal de motor 12V.

3.1.1.3 Sensor de color TCS3200

El sensor TCS2300 (Figura 3.5) es un sensor usado para la detección de colores. Cuenta con un matriz de fotodetectores, cada uno con un filtro rojo, verde o azul o sin filtro. Los filtros de cada color están distribuidos de manera equitativa a través de la matriz para eliminar el sesgo de localización entre los colores. En el interior del dispositivo hay un oscilador que produce una salida de onda cuadrada cuya frecuencia es proporcional a la intensidad del color elegido [26].

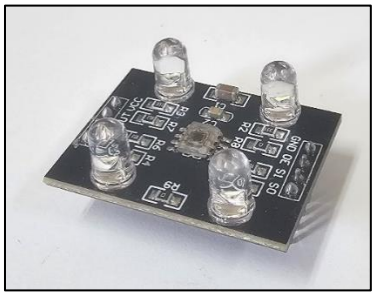


Figura 3.5: Sensor de color TCS3200.

Este sensor es ideal para la evaluación de las piezas, donde el verde significa una pieza correcta y el rojo una pieza defectuosa. Este sensor puede ser controlado mediante la placa de desarrollo Arduino UNO y cuenta con 8 pines los cuales son:

- **GND:** Tierra de la fuente de alimentación.
- **OE:** Habilitación para la frecuencia de salida (activo bajo).
- **OUT:** Frecuencia de salida.
- **S0, S1:** Entradas de la selección de escalado de la frecuencia de salida.
- **S2, S3:** Entradas de selección del tipo de fotodiodo.
- **VCC:** Voltaje de alimentación (2.7 a 5.5 V).

3.1.1.4 Servomotor Mg996r 360°

El servomotor MG996R (Figura 3.6) se alimenta de 4.8 a 7.2 V, con lo que puede generar torques de hasta 11 kg/cm [27]. Cuenta con la característica especial de girar 360° en rotación continua, lo que lo hace adecuado para el control del giro del plato rotatorio. Cuenta con tres pines los cuales son:

- **Café:** GND.
- **Rojo:** Alimentación.
- **Amarillo:** Señal.

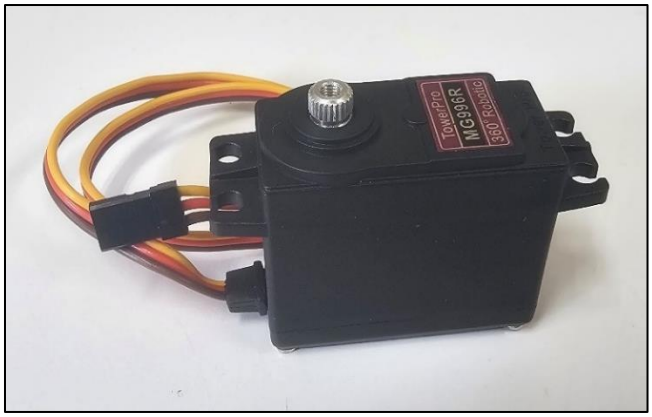


Figura 3.6: Servomotor MG996R con giro de 360°.

3.1.1.5 Módulo de puente H L298N

Este módulo cuenta con el circuito integrado L298N, el cual es un doble puente H. Es capaz de manejar altos voltajes y corrientes, además de soportar cargas inductivas como relés o motores de CD [28]. Esto lo hace un componente útil para el control de movimiento de los actuadores lineales, pues cada módulo puede controlar dos de estos. El módulo se puede observar en la Figura 3.7.

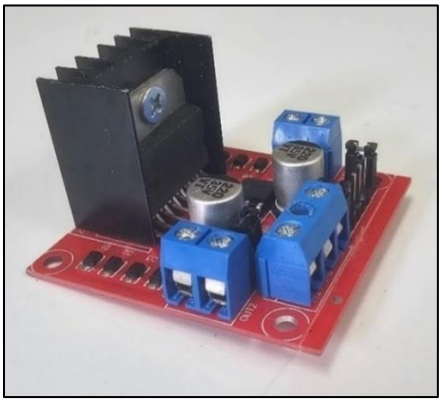


Figura 3.7: Módulo de puente H L298N.

3.1.1.6 Arduino UNO

La placa Arduino UNO, mostrada en la Figura 3.8, es una herramienta muy utilizada en el desarrollo de proyectos electrónicos, tiene aplicaciones tanto educativas como industriales. Para este caso, la placa será de ayuda para el control del servomotor y del sensor de color, lo que significa que el PLC se comunicará directamente con la placa mediante sus entradas y salidas.

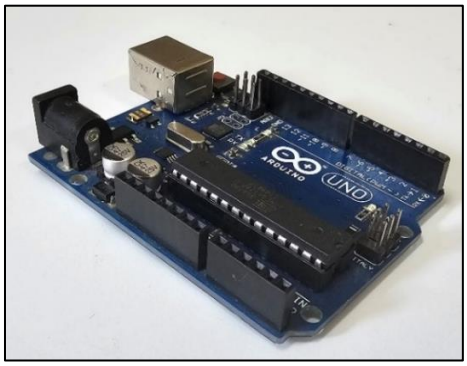


Figura 3.8: Placa Arduino UNO.

3.1.1.7 Interruptor final de carrera

Este tipo de interruptores cuentan con una pequeña palanca con rodillo para activar el interruptor al momento de que un objeto choca con ellos marcando el final de su carrera. Pueden tener una entrada común de voltaje, una salida normalmente abierta y una salida normalmente cerrada. Estos dispositivos se pueden utilizar como sensores de fin de carrera para los límites de los actuadores lineales. En la Figura 3.9 se muestra un ejemplo con sólo la entrada común y una entrada normalmente abierta.

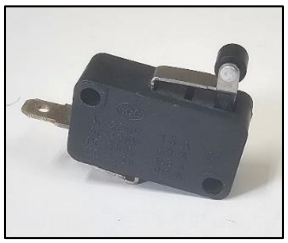


Figura 3.9: Interruptor de fin de carrera.

3.1.1.8 Botones pulsadores

Estos botones sirven como entradas desde el panel de mando. Cuentan con distintos colores para que sea más fácil la identificación de cada uno (Figura 3.10a). Además, también se cuenta con un botón especial que funciona como paro de emergencia, el cual es normalmente cerrado (Figura 3.10b), a diferencia de los otros que son normalmente abiertos y cuenta con enclavamiento, con el objetivo de garantizar que se mantenga pulsado hasta que sea seguro volver a poner el sistema en marcha.

a) b)

Figura 3.10: Botones pulsadores (a) Botón pulsador. (b) Botón paro de emergencia con enclavamiento.

3.1.2 Descripción del prototipo implementado

En esta sección se describe el funcionamiento de cada una de las estaciones del prototipo, detallando desde sensores hasta el equipo que conforman a cada una de estas.

3.1.2.1 Estación de carga

La estación de carga (Figura 3.11) es con la que comienza todo el proceso al proveer al sistema las piezas que se encuentran apiladas en su almacén. La posición inicial de la estación es el cargador retraído y el sensor de fin de carrera activado para indicarlo (sensor 2). El sensor que detecta cuando el cilindro está extendido (sensor 3) está apagado. Cuando el cargador se extiende, este sensor se enciende, sin embargo, está conectado de manera que pareciera estar siempre activo y al extender el cilindro se apaga, es decir, que es un sensor normalmente cerrado.

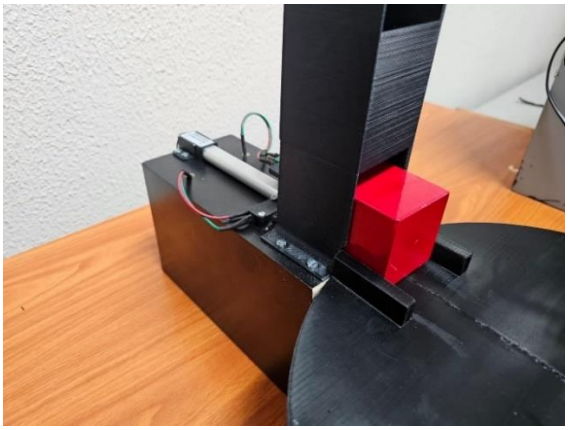


Figura 3.11: Estación de carga del sistema.

El sensor que detecta las piezas (sensor 1) se encuentra en la parte inferior del almacén donde no será presionado accidentalmente por el cargador, solamente por las piezas que caigan por la boca del almacén. En la Figura 3.12 se puede observar el esquemático de la estación, con todas las partes que la conforman.

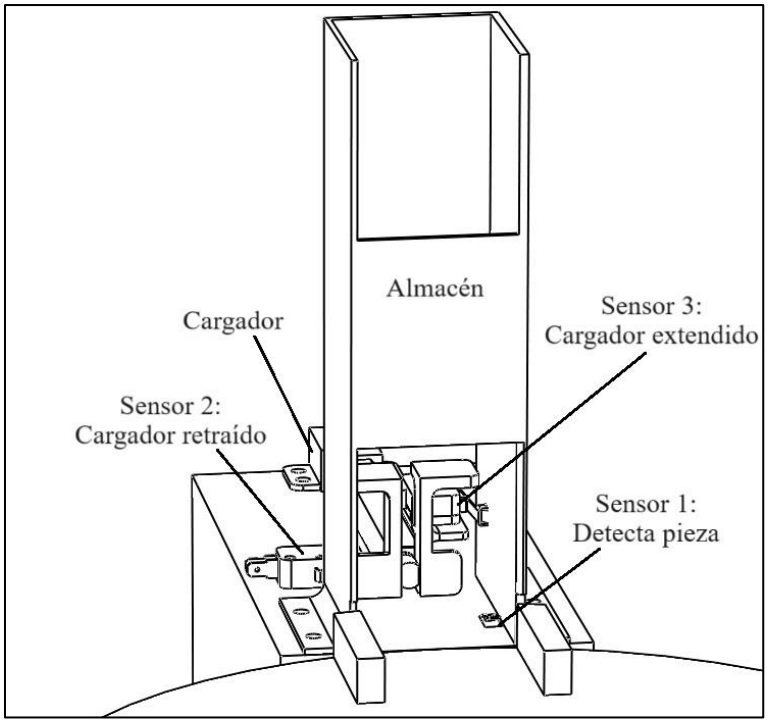


Figura 3.12: Partes que conforman la estación de carga.

Para inicializar esta estación se necesitan validar las condiciones iniciales y posteriormente pulsar el botón de marcha.

Secuencia de trabajo:

1. Entra una pieza al almacén.
2. El sensor 1 detecta la pieza.
3. Se extiende el cargador y se espera a que el sensor 3 se libere indicando que la pieza ha sido colocada en el plato rotatorio.
4. Se retrae el cargador hasta que el sensor 2 indique que se encuentra completamente retraído y fuera del almacén, donde otra pieza puede caer sin ninguna restricción.

En caso de falla o paro de emergencia, se requiere hacer un rearme. En este caso, se debe verificar que tanto el sensor 2 como el sensor 3 se encuentren pulsados. En caso de no estarlo, se retraerá el cargador hasta que llegue a su posición inicial.

3.1.2.2 Estación de maquinado

La estación de maquinado (Figura 3.13) es la representación de un trabajo de maquinado sobre la pieza en el plato rotatorio. El cilindro representa algún mecanismo de maquinado, como podría ser un taladro o una máquina de estampado la cual se extiende para hacer contacto con la pieza y después regresar a su posición inicial, la cual es con el cilindro retraído de forma que el sensor 4 se encuentre activo. La Figura 3.14 muestra todas las partes que conforman esta estación.

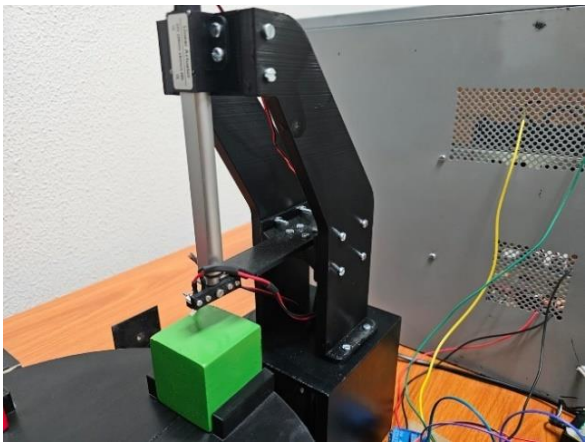


Figura 3.13: Estación de maquinado del sistema.

Esta estación se inicializa de manera automática en el segundo ciclo de trabajo del sistema, después de que la primera etapa se haya activado. La secuencia es la siguiente:

Secuencia de trabajo:

1. Una pieza está presente en la etapa.
2. Extender cilindro hasta que el sensor 5 se active por contacto con la pieza.
3. Retraer el cilindro hasta que el sensor 4 toque el límite de la base, regresando a su posición inicial

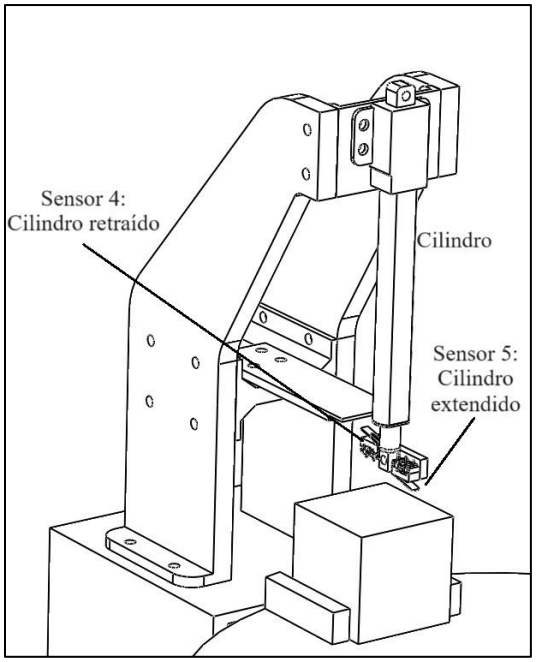


Figura 3.14: Partes que conforman la estación de maquinado.

En caso de rearme, se evalúa que el sensor 4 se encuentre activo, mientras que el sensor 5 se encuentre desactivado. Se retraerá el cilindro con el objetivo de regresar a su posición inicial.

3.1.2.3 Estación de descarga

La tercera y última etapa del proceso (Figura 3.15) consta de la evaluación de las piezas para determinar si estas son correctas o defectuosas, para después poder descargarlas dependiendo de a qué clasificación pertenezcan.

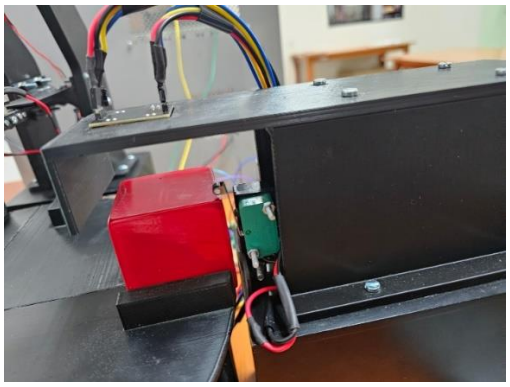


Figura 3.15: Estación de descarga del sistema.

En su posición inicial, el empujador está retraído de forma que esté activando el sensor 6. El sensor 7 debe mantenerse inactivo, pues este sirve para indicar que la pieza ha sido expulsada. En la Figura 3.16 se muestran las partes que conforman esta tercera y última estación.

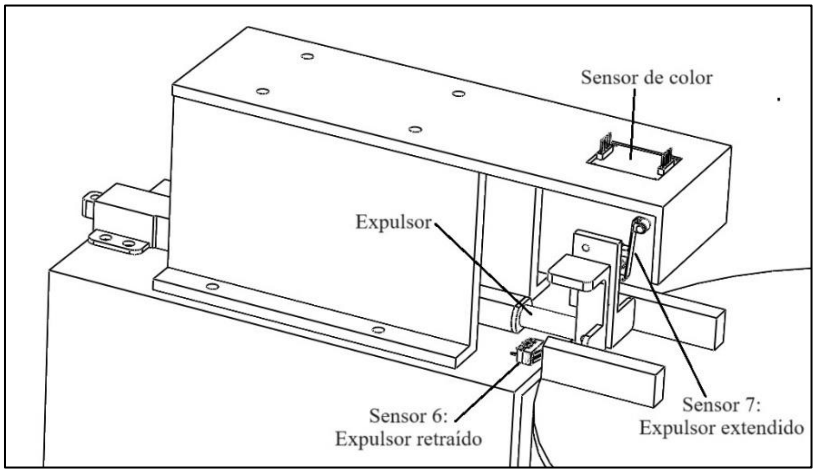


Figura 3.16: Partes que conforman la estación de descarga.

Para inicializar esta etapa es necesario que las dos anteriores hayan sido inicializadas y hayan procesado una pieza.

Secuencia de trabajo:

1. Una pieza está presente en la etapa.
2. Se activa el sensor de color que sirve para detectar si la pieza es correcta (color verde) o defectuosa (color rojo).
3. Si la pieza es color verde, se activa el expulsor.
4. El expulsor se extiende, empujando la pieza, hasta que el sensor 7 se active con el contacto del límite de la pieza.
5. Se retrae el expulsor hasta que el sensor 6 se active, volviendo a la posición inicial.

En caso de que la pieza sea color rojo, el sistema lo interpreta como una pieza defectuosa y, por lo tanto, omite el movimiento del cilindro. En el siguiente ciclo la pieza defectuosa continúa en los soportes del plato rotatorio y se dirige a una zona de descarga manual donde no afecta ninguna

etapa del proceso.

3.1.2.4 Plato rotatorio

El plato rotatorio (Figura 3.17) se encarga de mover las piezas a través de las distintas estaciones de trabajo con la ayuda de un servomotor de 360°. Su posición inicial es cuando alguno de los topes del plato se encuentre activando al sensor fin de giro, pues esto garantiza que una pieza pueda encajar de manera correcta desde la estación de carga.

Figura 3.17: Plato rotatorio

El plato cuenta con 4 topes colocados a 90° entre sí. El servomotor de 360° se encuentra colocado en la base del plato y su eje se encuentra acoplado al centro del plato rotatorio, permitiendo así un giro que se detiene cada 90°, es decir, cada vez que un tope activa el sensor fin de giro.

En la parte superior del plato se encuentran 4 pares de soportes en los que el cargador del sistema acomoda cada pieza que va entrando en este. La Figura 3.18 muestra a todas las partes que conforman el plato rotatorio.

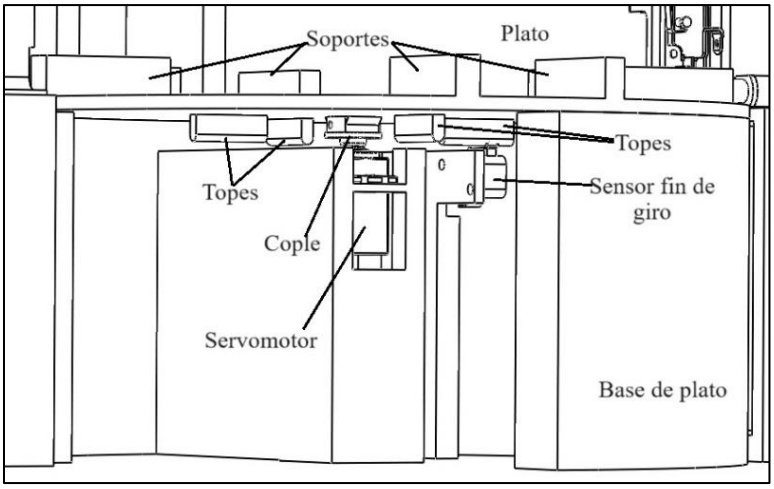


Figura 3.18: Partes que conforman el plato rotatorio.

3.1.3 Panel de mando

En cuanto al panel de control del sistema, se cuenta con un panel genérico existente desarrollado para el control de un prototipo para prácticas con autómatas programables. Este panel se muestra en la Figura 3.19. Cuenta con 4 indicadores, cuatro botones pulsadores, un selector y un botón de paro de emergencia.

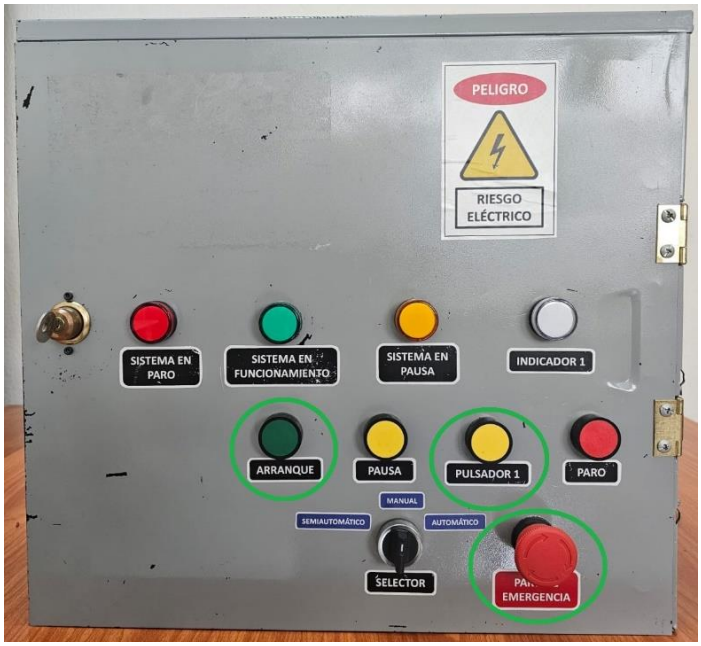


Figura 3.19: Panel de control funcional del sistema.

Idealmente, el selector tiene la función de elegir entre tres modos de marcha (manual, semiautomático y automático), sin embargo, este no es utilizado debido a la cantidad limitada de entradas y salidas del modelo del PLC utilizado. En cuanto a los botones pulsadores, solamente el botón de arranque o marcha y el pulsador 1 o rearme serán utilizados, por el mismo motivo. De igual forma, los indicadores no tienen lugar para funcionar, ya que el PLC no cuenta con salidas suficientes para su manejo. El paro de emergencia sí será utilizado, pues es indispensable para cualquier proceso industrial. Se recuerda que este es un botón normalmente cerrado con enclavamiento, lo que quiere decir que al presionarlo se mantendrá en ese estado hasta girar el botón para liberarlo.

Teniendo en cuenta estas adaptaciones, el panel de mando en la Figura 3.19 muestra seleccionados de color verde los botones que realmente están asociados a entradas en el PLC físico del sistema. El resto de pulsadores e indicadores se mantienen en el panel por cuestión estética.

3.1.4 Circuito del sistema

Una vez conocido el equipo necesario, y siguiendo el diagrama de bloques mostrado en la Figura 3.2 se procede a realizar el circuito del sistema de una manera más detallada, con cada una de las entradas y salidas del PLC asignados hacía un componente en específico.

En las Tabla 3.1 yTabla 3.2 se muestran los nombres de cada entrada y salida de los PLC, así como el equipo que cumple esta función y la dirección absoluta del PLC a la que responden. Tres entradas son utilizadas para los botones de marcha, de rearme y el botón de paro de emergencia, mientras que ocho entradas se conectan a sensores de fin de carrera dentro del prototipo del sistema, para finalmente dejar una última entrada para recibir la señal de si una pieza es color verde, proveniente de la tarjeta Arduino UNO.

En cuanto a las salidas, seis de ellas se conectan a un puente H, encargado de controlar un actuador lineal en específico, mientras que otras dos se conectan al Arduino UNO para enviar la señal de giro del servomotor e indicar que se realice la evaluación del señor de color. Se observa que se dejan sin utilizar dos entradas y dos salidas. Esto es para que con estas se realicen las operaciones

de comunicación entre los dos PLC. Esto se aborda en la sección 3.4.1.

Tabla 3.1: Entradas del sistema

ENTRADAS		
Nombre	**Equipo**	**Dirección PLC**
Marcha (M)	Botón pulsador	I0.0
Paro de emergencia (PE)	Botón paro de emergencia	I0.1
Rearme	Botón pulsador	I0.2
Pieza alimentada (Pieza)	Sensor fin de carrera	I0.3
Cargador extendido (a1)	Sensor fin de carrera	I0.4
Cargador retraído (a0)	Sensor fin de carrera	I0.5
Cilindro extendido (b1)	Sensor fin de carrera	I0.6
Cilindro retraído (b0)	Sensor fin de carrera	I0.7
Fin de rotación del plato (r0)	Sensor fin de carrera	I1.0
Expulsor extendido (d1)	Sensor fin de carrera	I1.1
Expulsor retraído (d0)	Sensor fin de carrera	I1.2
Color verde (cv)	Sensor de color	I1.3
-	-	I1.4
-	-	I1.5

Tabla 3.2: Salidas del sistema

SALIDAS		
Nombre	**Equipo**	**Dirección PLC**
Extiende cargador (A+)	Puente H (Actuador 1)	Q0.0
Retrae cargador (A-)	Puente H (Actuador 1)	Q0.1
Extiende cilindro (B+)	Puente H (Actuador 2)	Q0.2
Retrae cilindro (B-)	Puente H (Actuador 2)	Q0.3
Evaluar Pieza (SC)	Arduino UNO (Sensor de color)	Q0.4
Extiende expulsor (D+)	Puente H (Actuador 3)	Q0.5
Retrae expulsor (D-)	Puente H (Actuador 3)	Q0.6
Gira plato (R+)	Arduino UNO (Servomotor)	Q0.7
-	-	Q1.1
-	-	Q1.2

3.1.5 Circuito de control

Para lograr que los dos PLC puedan controlar el mismo sistema se hace uso de una compuerta lógica **XOR**, la cual permite que entren dos señales y sólo salga la que se encuentre activa, mientras que en caso de que se encuentren ambas activas no habrá una señal de salida. En la Figura 3.20 se muestra un diagrama de conexión donde se observan dos compuertas **XOR**. En la compuerta superior las entradas son la salida que extiende el cargador en ambos PLC, mientras que la salida de la compuerta se conecta directamente al puente H encargado de controlar la señal de extender el cargador. Un caso similar se observa en la compuerta **XOR** inferior pero ahora con la salida que retrae el cargador.

Este diagrama se repite requiriendo un total de 8 compuertas lógicas en total, esto debido a las 8 salidas que tienen los programas de ambos PLC. Haciendo uso del circuito integrado **74LS86N**, el cual contiene 4 compuertas **XOR** se realiza un circuito en placa PCB para poder realizar la conexión física de los PLC en el prototipo.

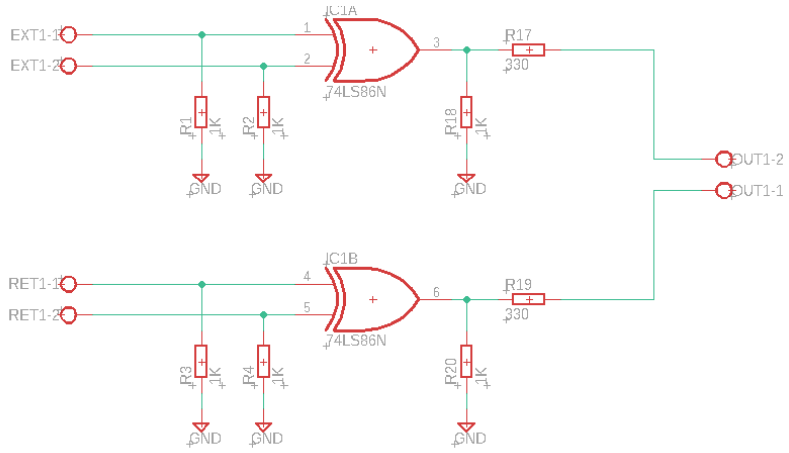


Figura 3.20: Diagrama de conexión de las salidas de los PLC a la compuerta XOR.

Para que esta configuración fuera posible se utilizó el circuito integrado **SN74LS86N** (Figura 3.21), el cual cuenta con 4 compuertas **OR exclusivas** (**XOR**). Su configuración se muestra en la Figura 3.22.

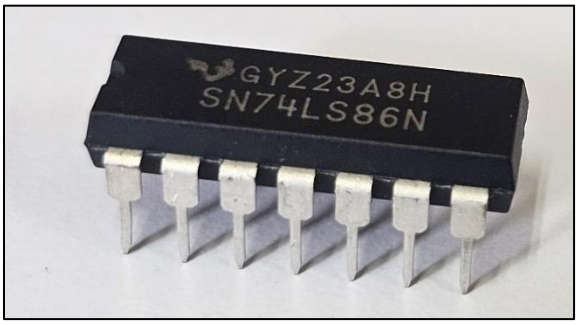


Figura 3.21: Circuito SN74LS86N.

Se utilizaron dos de estos circuitos integrados, se conectaron a una placa PCB que optimiza el espacio que puede utilizar este circuito de control dentro del sistema.

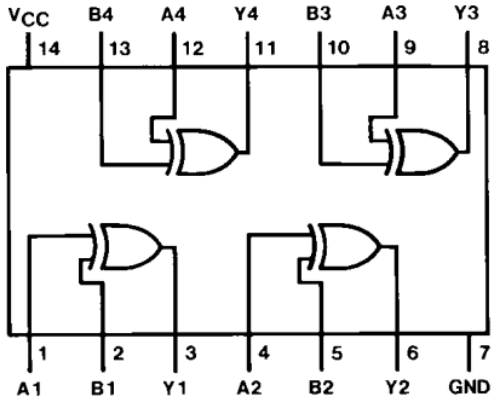


Figura 3.22: Disposición de pines SN74LS86N [29].

En la Figura 3.23 se muestra entonces el circuito final del sistema, considerando los dos PLC y el circuito de compuertas que permite el control del sistema. Se observa que cada una de las entradas y salidas de ambos PLC corresponde al equipo descrito en las Tabla 3.1 yTabla 3.2 respectivamente.

Ambos PLC se conectan a una fuente de corriente alterna mediante su cable de alimentación. Los PLC ofrecen una fuente de 24V para alimentar los sensores de entrada, la cual se utilizó para conectar los interruptores de fin de carrera al sistema. Por otro lado, se tiene una fuente de corriente directa de 5V la cual se conecta a la terminal de salidas de cada PLC, pues este es el voltaje que necesita el puente H para controlar los actuadores y la tarjeta Arduino para recibir un uno lógico en sus entradas digitales.

La tarjeta Arduino alimenta al sensor de color y además contiene el programa para activarlo al momento de recibir 5V en su entrada digital en el pin 2. En caso de que el sensor detecte que la pieza es de color verde se envía una señal en bajo a la salida digital en el pin 12, la cual se encuentra conectada al control de un módulo de relevador. Este módulo sirve para que al activarse se cierre el circuito de 24V conectado a la entrada I1.3 del PLC, permitiendo que el controlador reciba una señal en caso de que la pieza sea correcta. Cabe recalcar que este tipo de módulos activan el relevador con una señal en bajo y lo desactivan con una señal en alto (5V).

El circuito a color se puede encontrar en el Anexo 1.

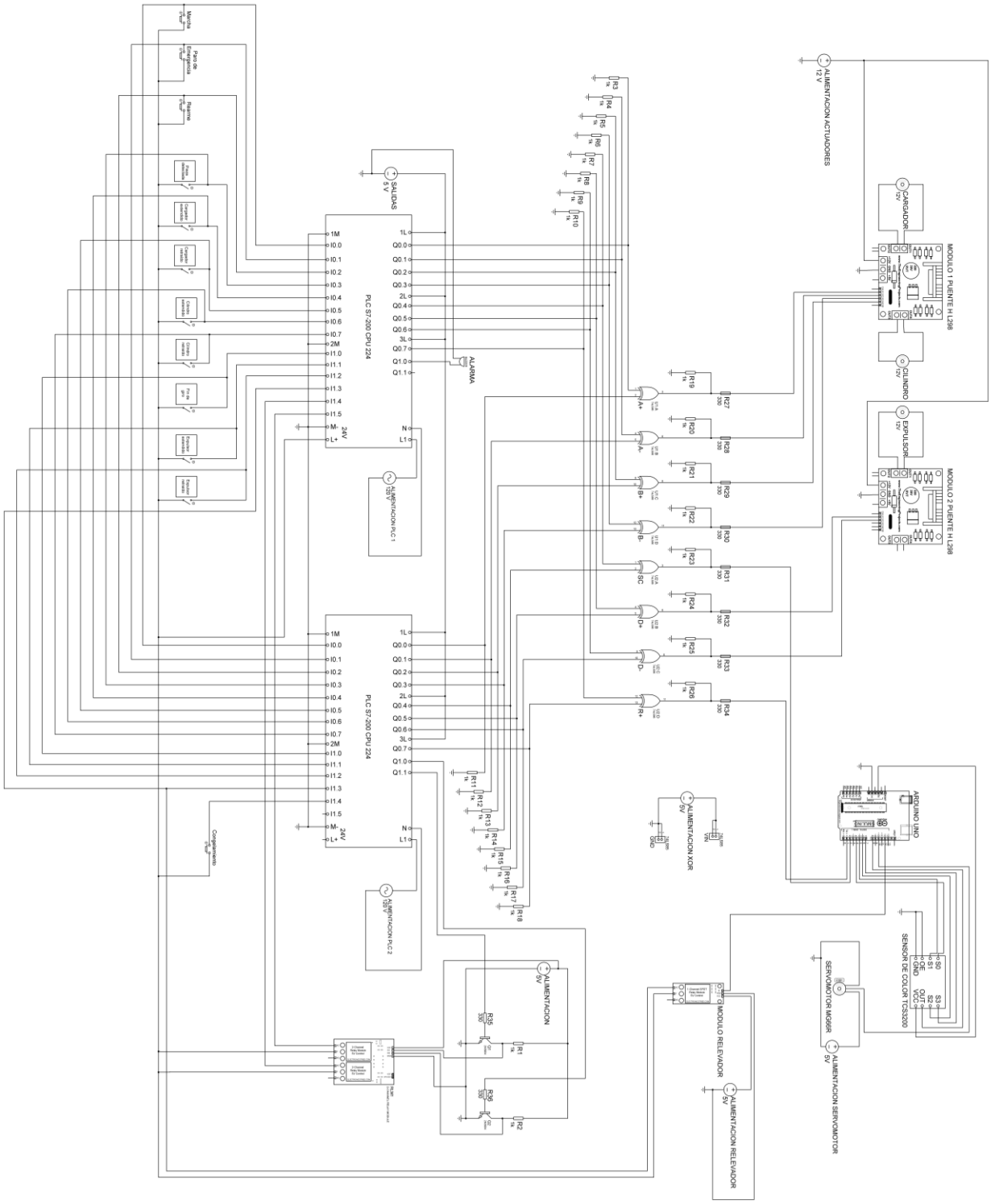


Figura 3.23: Circuito completo del sistema con dos PLC (Anexo 1).

También se observa que el Arduino tiene otra salida, esta vez en su pin 10, la cual sirve para controlar el giro del servomotor, el cual gira en sentido de las manecillas del reloj hasta que el sensor de fin de giro se encienda, provocando la caída de una señal en la salida Q0.7 de los PLC, conectada a la entrada digital del pin 1 del Arduino para indicar que se detenga el giro del servomotor.

Se debe tener en cuenta que los símbolos de tierra o GND que aparecen en el circuito hacen referencia a una sola tierra, pues es indispensable que todos los componentes tengan su tierra conectada entre sí para que el sistema pueda funcionar correctamente.

Finalmente, se puede notar un módulo de relevadores conectados a las dos salidas libres del PLC 1 y a las dos entradas libres del PLC 2. Estos se activan con los 5V de las salidas Q1.1 y Q1.2 del PLC 1 y en sus salidas permiten la conexión de 24V para activar las entradas I1.4 e I1.5 del PLC 2. Se observa que se encuentran conectados un par de transistores, debido a que este módulo se activa con un cero lógico y se desactiva con un uno lógico, por lo que los transistores funcionan como un interruptor controlado por voltaje que permite un cero lógico al momento de que se presenta un voltaje en su base y un uno lógico al momento de retirar dicho voltaje.

3.2 Configuración de algoritmos

En este apartado se muestran los algoritmos, métodos y técnicas descritos en el anterior capítulo aplicados al desarrollo del prototipo del proceso tipo transfer circular.

3.2.1 Grafcet de primer nivel

El prototipo del proceso cuenta con tres estaciones de trabajo, tal como se mostró en la Figura 3.1, y cada una de estas realiza una tarea distinta. La estación de carga se encuentra inactiva hasta que se detecte que una pieza ha sido alimentada al sistema, entonces se extiende el cargador de la estación con el fin de empujar dicha pieza al plato rotatorio. Un sensor detecta que el cargador ha cumplido su función e inmediatamente se procede a retraerlo hasta que haya vuelto a su posición inicial, listo para iniciar el ciclo nuevamente.

En cuanto a la estación de maquinado, esta sigue un funcionamiento similar a la estación anterior, donde se mantiene inactiva hasta que se detecte una pieza en la estación, entonces el cilindro se extenderá hasta activar un sensor de fin de carrera al llegar a tocar la pieza, esto comenzará la retracción del cilindro hasta que se detecte que haya llegado a su posición inicial.

Finalmente, en la estación de descarga, se requiere primero la detección de una pieza para iniciar su funcionamiento, al igual que en las anteriores estaciones, sin embargo, esto lleva a que se ejecute una evaluación de pieza para determinar si esta se encuentra en buen estado o está defectuosa. En caso de ser una pieza correcta se extiende el expulsor hasta colocar la pieza al centro del plato rotatorio. Una vez cumplida esta tarea se retrae el expulsor hasta que vuelva a su posición inicial. Por otro lado, si la pieza es defectuosa, simplemente se omite todo el proceso de descarga y se deja que la pieza siga dentro del plato rotatorio para llegar a la descarga manual. En la Figura 3.24 se muestran los Grafcet parciales de primer nivel que ilustran el comportamiento esperado de cada estación de trabajo.

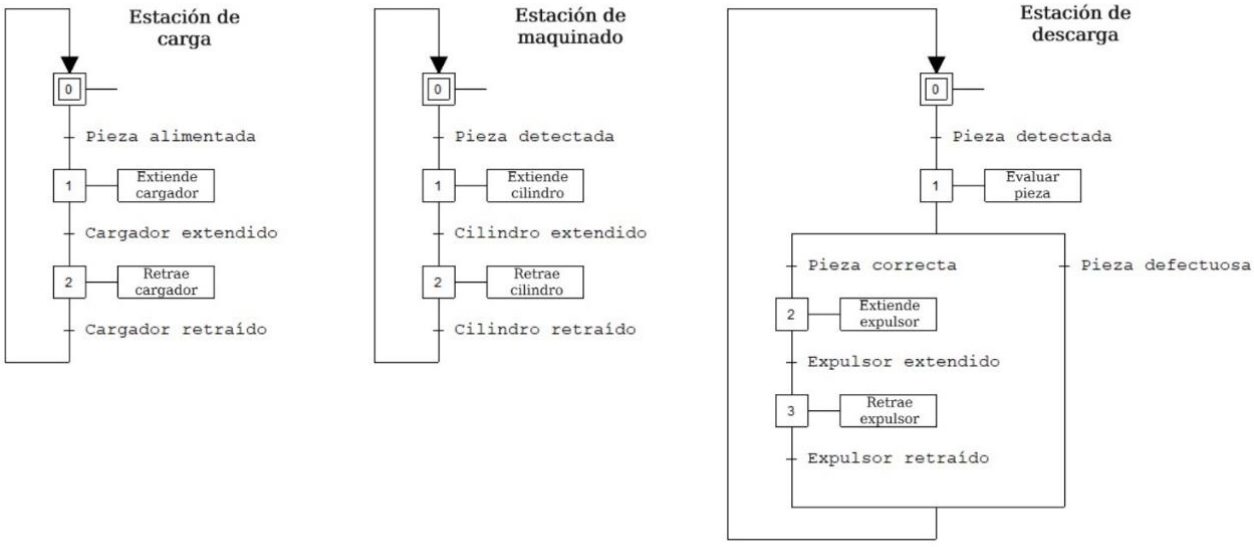


Figura 3.24: Grafcet parciales de primer nivel de las estaciones de trabajo.

3.2.2 Grafcet de segundo nivel

El Grafcet de segundo nivel implica el tener un mejor entendimiento de las entradas y salidas al sistema, ahora considerando los componentes que realizan las acciones descritas en el Grafcet de primer nivel. Para esto es necesario recordar los nombres que se le da a cada pieza en función de la acción que se encuentre realizando. En la Tabla 3.3 se muestran los nombres de las acciones mostradas en el Grafcet de primer nivel, destacando que SC representa la activación del sensor de color para determinar si una pieza es correcta o defectuosa, así como el uso de dos contadores, los cuales cumplen la función de detectar piezas en las estaciones de maquinado y descarga.

Tabla 3.3: Nombres de las acciones de los Grafcet de segundo nivel.

Nombre	**Acción**
A+	Extiende cargador
A-	Retrae cargador
B+	Extiende cilindro
B-	Retrae cilindro
D+	Extiende expulsor
D-	Retrae expulsor
SC	Activar sensor de color
C1	Contador 1
C2	Contador 2

La estación de carga no presenta cambios, pues es un sensor de fin de carrera el que determina si una pieza ha sido alimentada al sistema, mientras esto no sea así, el sistema no realiza ninguna acción. Sin embargo, ahora se puede notar que al final de este Grafcet cuando se llama a la acción de retraer el cargador, también se incrementa en uno la cuenta de **C1**. Este contador se incrementa de cero a uno cada vez que una pieza entra al sistema por medio de la estación de carga y se puede observar al inicio del Grafcet de la estación de maquinado, que mientras tenga este valor se realizan las acciones necesarias de la estación, en caso contrario se omite cualquier acción. Nuevamente en

la última acción de esta estación se incrementa un contador, ahora llamado **C2**, mientras que a **C1** se le resta uno en la cuenta, regresándolo a un valor de cero, esto indicando que la pieza ahora pasa a la siguiente estación. Esta técnica de utilizar contadores para la detección de piezas en cada estación permite el ahorro de entradas físicas al PLC, pues sólo se necesita un sensor al inicio de la primera estación.

En la estación de descarga se observa que nuevamente que un contador indica la presencia de una pieza en la estación. Además de eso se hace uso de un temporizador (T5), el cual tiene una cuenta de cinco segundos. Si al finalizar este tiempo no se ha recibido la señal que indica la detección de una pieza verde (cv), se considera que la pieza es defectuosa. En este caso, el sistema procede a decrementar el contador. Posteriormente, el proceso se detiene temporalmente y espera a que se realice el rearme del sistema antes de continuar con el proceso.

La Tabla 3.4 muestra los nombres de los sensores que funcionan como transiciones de los Grafcet cuando se encuentran activos. De esta manera, los Grafcet de segundo nivel de cada una de las estaciones del sistema se muestran en la Figura 3.25, donde de una manera simplificada se puede entender el funcionamiento de cada estación.

Tabla 3.4: Nombres de sensores de los Grafcet de segundo nivel.

Nombre	**Sensor**
a0	Cargador extendido
a1	Cargador retraído
b0	Cilindro extendido
b1	Cilindro retraído
d0	Expulsor extendido
d1	Expulsor retraído
cv	Color verde detectado
T5	Temporizador de 5 segundos
Pieza	Pieza detectada

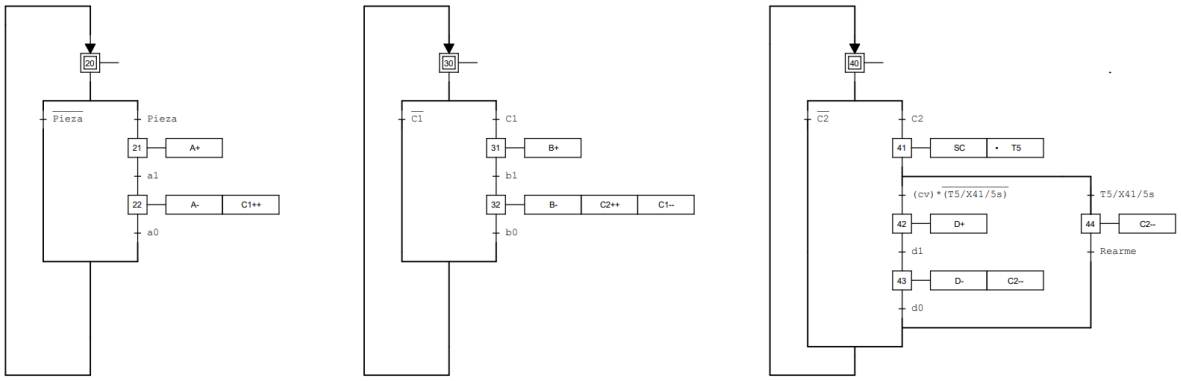


Figura 3.25: Grafcet de segundo nivel de las estaciones de trabajo.

3.2.3 Grafcet de producción del proceso

En la Figura 3.26 se muestra el Grafcet de producción del proceso de primer nivel, este integra los tres Grafcet parciales de las estaciones de trabajo y los coordina para trabajar de manera simultánea, de modo que al inicio de todo el proceso se ponen a cero las cuentas de ambos contadores, así como verificar que los actuadores se encuentren en su posición inicial. El pulsador de marcha es aquel que da inicio a todo el proceso, para pasar a una etapa en blanco y, posteriormente, evaluar las condiciones iniciales del proceso y verificar que una pieza haya sido alimentada antes de entrar a los Grafcet parciales de cada estación de trabajo. Además de esto, se evalúa que los contadores tengan un valor en su cuenta, esto en caso de que ya no se alimenten piezas al sistema, pero siga habiendo algunas dentro de las estaciones de trabajo.

Una vez terminados cada uno de los Grafcet de las estaciones se pasa a una etapa de espera para que ningún proceso se realice a destiempo y se cierre la simultaneidad. Finalmente, se evalúa si los contadores no están en cero, en caso de que no, se procede a hacer el giro del plato rotatorio hasta que el sensor detecte un giro de 90°. En caso de que ambos contadores se encuentren vacíos no se realiza la rotación.

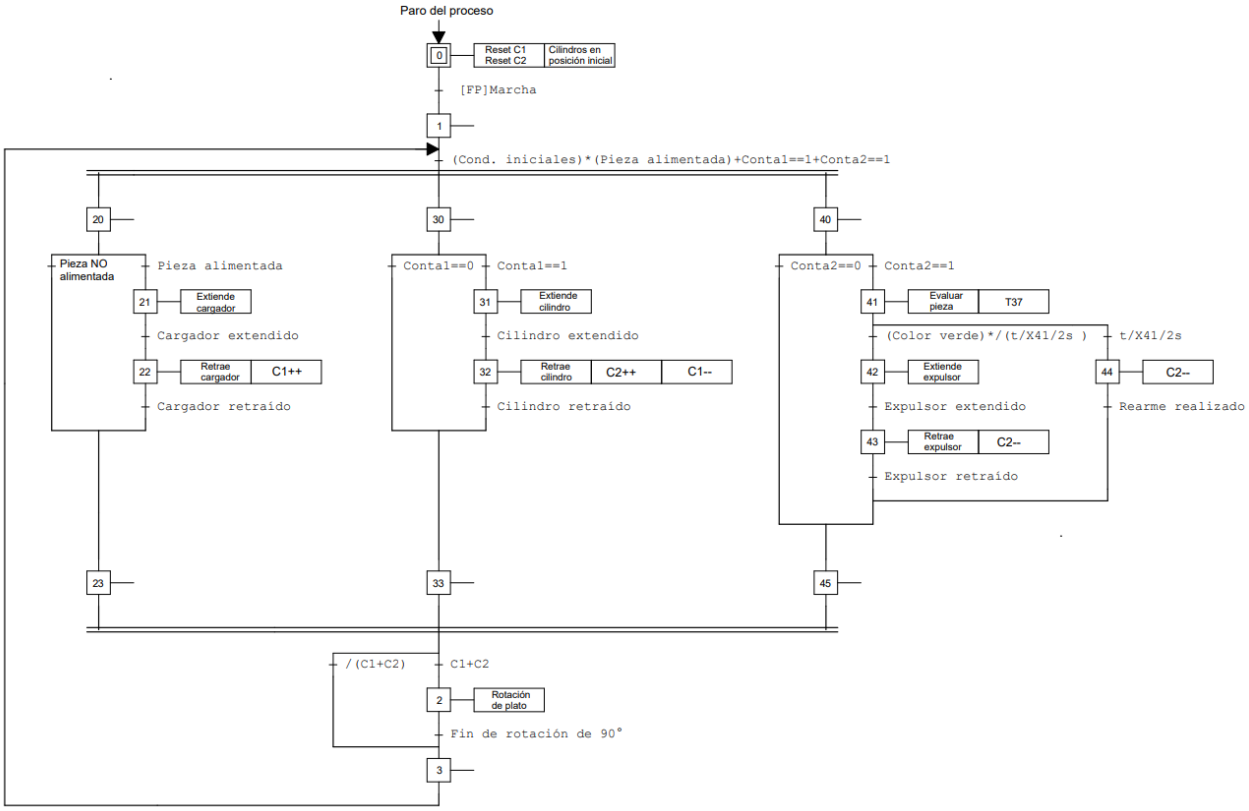


Figura 3.26: Grafcet de producción de primer nivel del proceso.

La Figura 3.27 muestra ahora el Grafcet de producción de segundo nivel, donde se observa que las acciones y transiciones han sido reemplazadas por los nombres previamente establecidos en las Tabla 3.3 yTabla *3.4*.

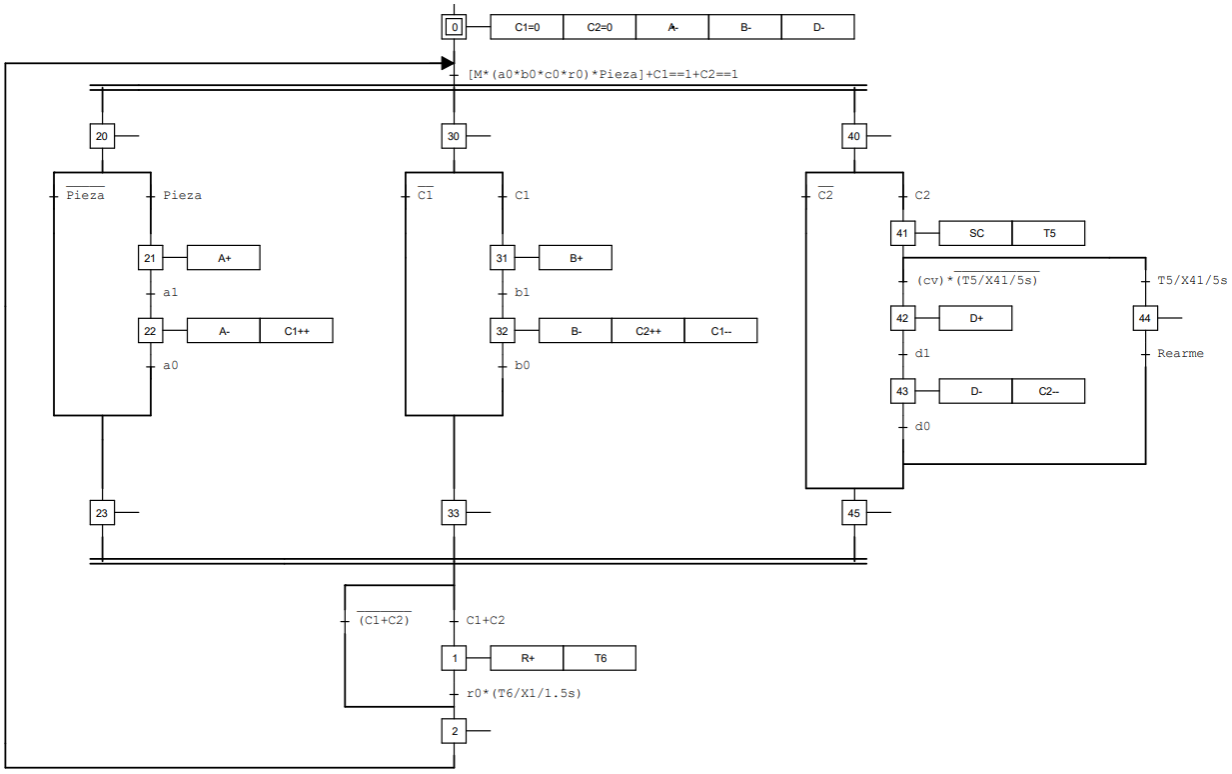


Figura 3.27: Grafcet de producción de segundo nivel del sistema (Anexo 2).

3.2.4 Guía GEMMA del sistema

En la Figura 3.28 se muestra la guía GEMMA que se utiliza para el proyecto, se observa que sólo existe un estado de producción, mientras que se mantienen dos de defecto y tres de parada. Esta reducción se debe a la limitante de las entradas y salidas en los PLC, ya que no se cuentan con las suficientes para abastecer las entradas del panel de control, junto con todas las entradas y salidas que requiere el proceso en sí.

El estado A1 es el inicio del proceso, donde todo se encontrará en estado inicial y se requiere de la activación de un botón de marcha para entrar al estado F1 de producción. En este estado el sistema se encuentra en un bucle de repetición del programa, del que se puede salir presionando el botón de paro de emergencia, el cual es normalmente cerrado y cuenta con enclavamiento. Al presionar el botón de paro se activa el estado D1, manteniendo el sistema inactivo hasta que el botón de paro de emergencia se desenclave, para después pasar al estado A5, es decir, la preparación del sistema

para la puesta en marcha, donde se debe asegurar que no exista ningún objeto que pueda ocasionar un atascamiento en el sistema o que ningún sensor o actuador se encuentre dañado. Una vez asegurado esto, se presiona el botón de rearme, el cual activa el estado A6, donde todos los actuadores deben volver a su posición inicial. Una vez que esto se cumpla, nuevamente se activa el estado A1, donde se puede volver a repetir el ciclo.

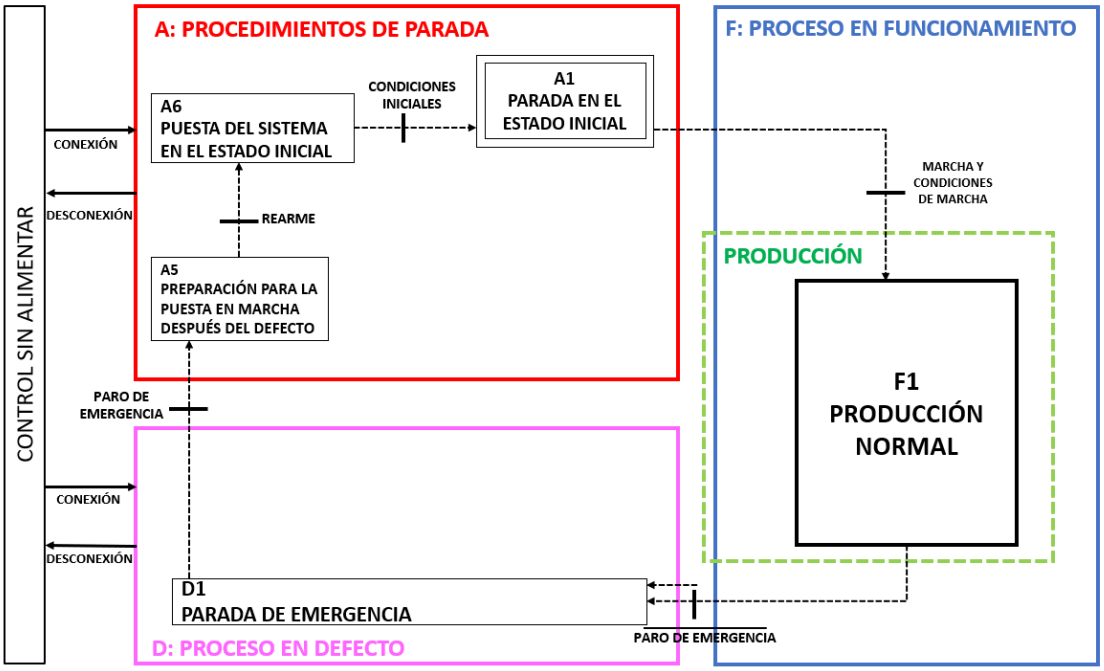


Figura 3.28: Guía GEMMA del sistema.

Es necesario mencionar que el botón de paro de emergencia puede ser activado en cualquier momento, aún si el estado F1 no se encuentra activo.

3.2.5 Grafcet de seguridad

La Figura 3.29 muestra el Grafcet de seguridad del sistema, el cual se encarga de supervisar el correcto funcionamiento del sistema. Además, contiene los estados de la guía GEMMA que corresponden al paro del proceso y a su puesta en marcha.

Este Grafcet tiene como primera transición el que se presione el paro de emergencia (PE), el cual es normalmente cerrado, por lo que en la transición se coloca negado. Esto activará el estado D1,

el cual ocasiona un forzado del Grafcet de producción, obligándolo a activar su etapa inicial o etapa cero, de la cual no podrá salir hasta que el paro de emergencia haya sido liberado. Una vez que esto ocurrió se debe llevar a cabo el estado A5, donde manualmente se debe liberar cualquier obstrucción o desperfecto que pueda ocasionar un error, para después pulsar el botón de rearme y activar A6, llevando todos los actuadores a su posición inicial hasta que las condiciones iniciales se cumplan.

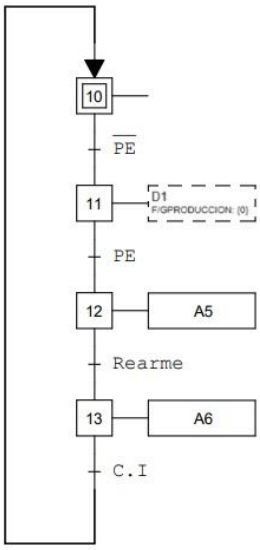


Figura 3.29: Grafcet de seguridad (Anexo 3).

Como se puede notar, este Grafcet es básicamente la traducción de los grupos A y D de la guía GEMMA de este sistema.

3.2.6 Grafcet de rearme

El último Grafcet a revisar es el de rearme (Figura 3.30), el cual se puede activar durante las etapas 12 del Grafcet de seguridad o 44 del Grafcet de producción, lo que representan X12 y X44 en la primera transición, además de presionar el botón de rearme. Una vez activado se comienzan a retraer todos los actuadores en orden hasta su posición inicial y asegurando que el sensor que indica que están extendidos se encuentre desactivado, para evitar algún error.

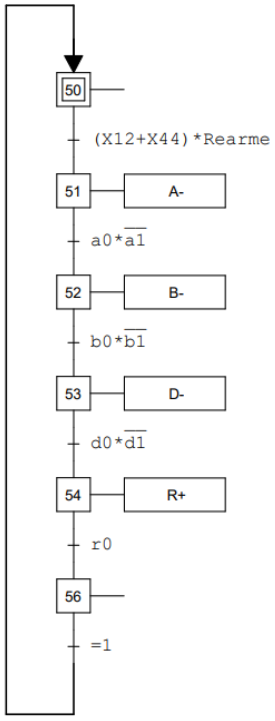


Figura 3.30: Grafcet de rearme del sistema (Anexo 3).

Una vez que el último sensor, en este caso r0, se encuentre activo, se activa una etapa de transición la cual rápidamente entra a una transición que marca el fin del proceso de rearme y regresa a la etapa inicial de este Grafcet. Este Grafcet es básicamente la representación del estado A6 de la guía GEMMA, pues coloca todos los actuadores en su posición inicial listos para la puesta en inicio del sistema.

3.2.7 Programación

Para implementar el programa en lenguaje Escalera con base en los Grafcet de producción, seguridad y rearme, se consideró que cada etapa estaría representada por una marca dentro del programa, tal como se menciona en la sección 1.1.4.4. Cada una de estas marcas activa la salida que está asociada a cada actuador en el PLC. Para cambiar de marca es necesario que se cumpla la transición, la cual es una bobina en el programa que representa las entradas que se tienen que activar o los contadores o temporizadores que deben tener un estado alto para realizar la transición. Cada

marca se pone en alto cuando las condiciones para su activación se cumplan y esta permanecerá en alto hasta que la activación de alguna otra marca también ponga en estado bajo la marca actual. Este programa es la base del funcionamiento del sistema. Este se puede cargar a un solo PLC con el fin de observar el funcionamiento del sistema y corroborar que este sea el correcto.

3.3 Sistema en funcionamiento

En la Figura 3.31 se observa el proceso en funcionamiento, con tres piezas apiladas dentro de la estación de carga. Se observa que la primera pieza, en este caso de color verde, se encuentra siendo empujada hacía el centro del plato rotatorio.

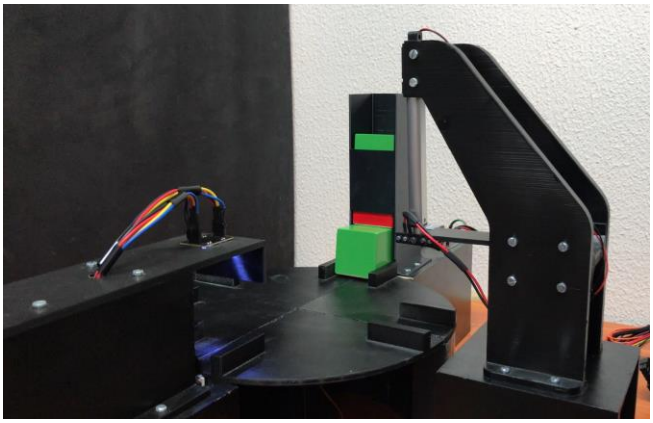


Figura 3.31: Sistema en funcionamiento: Etapa de carga.

La Figura 3.32 muestra la pieza verde ya cargada en el plato rotatorio siendo desplazada mediante este hacía la estación de maquinado, mientras que, en la estación de carga, una pieza roja se encuentra lista para ser empujada hacía el plato rotatorio.

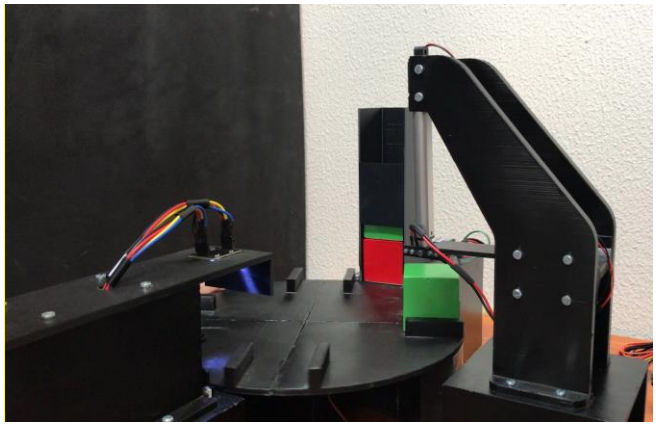


Figura 3.32: Sistema en funcionamiento: Giro del plato.

La Figura 3.33 muestra el funcionamiento simultáneo de la estación de carga y de maquinado. En este caso la pieza verde se encuentra en la estación de maquinado, mientras que la pieza roja se encuentra siendo cargada en el plato rotatorio.

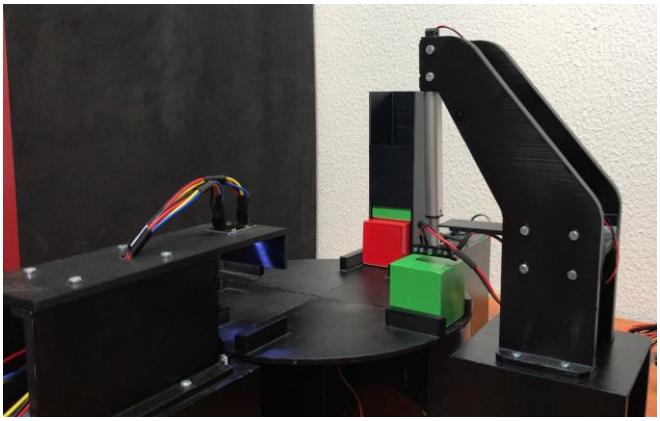


Figura 3.33: Sistema en funcionamiento: Etapa de carga y maquinado.

En la Figura 3.34 se muestran las tres estaciones trabajando de manera simultánea. En este caso, la primera pieza verde ahora se encuentra siendo empujada hacía el centro del plato por la estación de descarga, mientras la pieza roja se encuentra en la estación de maquinado y una nueva pieza verde está siendo empujada hacía el plato rotatorio en la estación de carga.

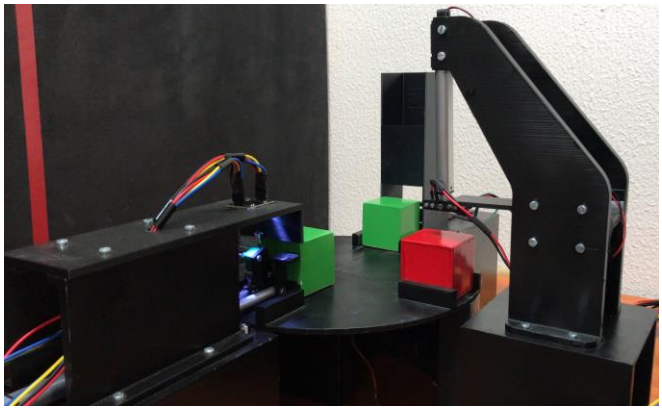


Figura 3.34: Sistema en funcionamiento: Etapa de carga, maquinado y descarga.

Por otro lado, la Figura 3.35 muestra la primera pieza verde ya lista para ser retirada del plato de manera manual, mientras que la pieza roja ahora se encuentra en la estación de descarga, sin ser empujada, debido a que esta es considerada una pieza defectuosa. Al mismo tiempo, se observa la última pieza verde en la estación de maquinado.

Figura 3.35: Sistema en funcionamiento: Etapa de descarga y maquinado.

Finalmente, la Figura 3.36 muestra la conexión física de los dos PLC, junto con todos los cables y componentes que se encuentran dentro del panel de mando.

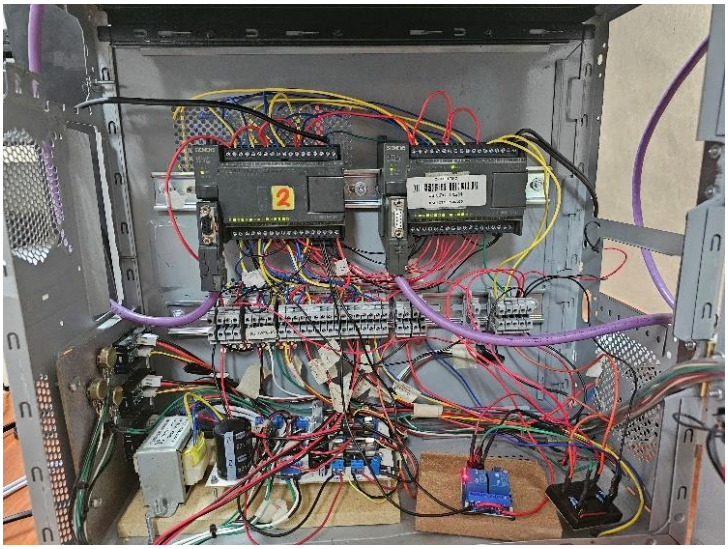


Figura 3.36: Conexión entre dos PLC.

3.4 Caso de estudio

En este apartado se describen los resultados obtenidos en el caso de estudio del prototipo. El caso de estudio es la realización del sistema redundante para el control de este prototipo del proceso tipo Transfer. Se describe la programación del sistema redundante para que el sistema continúe funcionando a pesar de una desconexión en el PLC 1, además se añade la programación del error de congelamiento en las salidas RLY del PLC principal.

3.4.1 Programación del sistema redundante

Dentro del programa de funcionamiento del sistema se tienen cinco variables que son de interés para su constante observación. Estas son, la etapa en la que se encuentra el programa, el temporizador de espera para la detección de color (T5), el temporizador de espera para el giro del plato rotatorio (T6) y los dos contadores que sirven para indicar la presencia de una pieza en las etapas de maquinado y de descarga.

Para la programación del sistema redundante se hizo uso del asistente de operaciones

NETR/NETW que ofrece el software STEP 7-MicroWin® [7]. Se configuraron un total cinco operaciones de red, todas de lectura a las que se les asoció una de las variables anteriormente descritas. La Tabla 3.5 muestra la dirección absoluta de los datos en la CPU de cada una de las operaciones:

Tabla 3.5: Configuración de las operaciones de lectura

Operación	**Tabla de datos locales de la CPU**	**Variable**
1	VB0 - VB5	Etapa del programa
2	VB10 - VB11	Temporizador 5
3	VB20 - VB21	Temporizador 6
4	VB30 - VB31	Contador 1
5	VB40 - VB41	Contador 2

Se puede observar que la primera operación tiene un tamaño de 6 bytes, mientras que el resto solamente de 2 bytes. Esto se debe a que la primera operación debe almacenar los datos de etapas que pueden estar activas de manera simultánea, mientras que el resto de operaciones sólo almacenan el valor de cuenta de un temporizador o contador.

Otras configuraciones a tener en cuenta es que el puerto de comunicación debe ser cero y la dirección del CPU remoto en este caso específico es cuatro. Además, el asistente asigna de manera automática las direcciones donde se almacena el estado de cada una de las operaciones. Al finalizar la configuración se crea una subrutina y la pestaña del asistente NETR/NETW pasa a verse como se muestra la Figura 3.37.

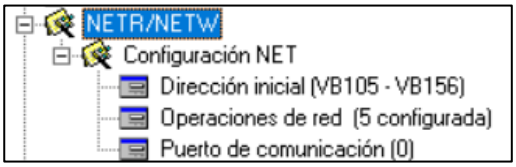


Figura 3.37: Asistente al terminar la configuración.

Para llevar a cabo el proceso de comunicación se modifica el programa original para crear dos

nuevos programas, uno encargado del envío de datos y otro de la recepción de los mismos, los cuales se cargan en el PLC 1 y PLC 2, respectivamente.

El PLC 1 funcionará como autómata principal, el cual tendrá el mando de las salidas al encenderse, cuya dirección remota es 4 y debe escribir sus variables en los bytes de las operaciones NETR para que el PLC 2 pueda leerlas. En la Figura 3.38 se puede observar un segmento del programa de este PLC, el cual justamente se encarga del envío de datos en todo momento, pues se utiliza la marca especial M0.0, la cual se encuentra activa en cada ciclo de reloj del autómata. En este segmento se realiza la escritura de los valores de los temporizadores T5 y T6, así como de los contadores C1 y C2 a los bytes correspondientes, según la Tabla 3.5, ya que estos bytes serán leídos por el programa del PLC 2.

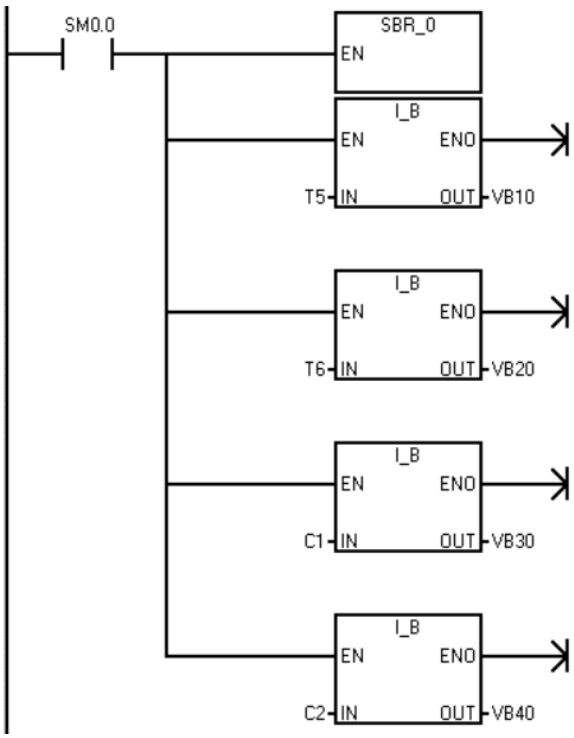


Figura 3.38: Envío de datos.

Además del envío de los contadores y temporizadores se observa una subrutina SBR_0 que se llama al mismo tiempo que se realiza la escritura de los datos. Esta subrutina contiene un programa donde dependiendo de la etapa que se encuentre activa se escribe un valor en el byte

correspondiente al Grafcet de esa etapa que, de manera similar, el PLC 2 lee y activa la etapa correspondiente en su respectivo programa.

Para aplicar este algoritmo, se deben asociar las marcas que indican que se encuentra activa una etapa del Grafcet del proceso con los bytes de la operación de lectura. La Tabla 3.6 muestra la relación que hay entre cada uno de los 6 bytes de la primera operación de lectura con las marcas de etapas que pueden encontrarse activas, estas son las etapas del Grafcet de producción, de seguridad y el de rearme. Cabe recalcar, ya que el Grafcet de producción contiene etapas que requieren de simultaneidad se toman por separado las etapas de cada uno de los Grafcet parciales.

Tabla 3.6: Relación entre etapas activas y bytes de la primera operación NETR.

Byte	**Etapas que se encuentran activas**	**Valores que puede tomar**	**Marcas**
VB0	Grafcet principal de producción	1 - 3	M0.1 – M0.3
VB1	Grafcet de seguridad	0 - 3	M1.0 – M1.3
VB2	Grafcet parcial de etapa de carga	0 - 3	M2.0 – M2.3
VB3	Grafcet parcial de etapa de maquinado	0 - 3	M3.0 – M3.3
VB4	Grafcet parcial de etapa de descarga	0 - 5	M4.0 – M4.5
VB5	Grafcet de rearme	0 - 5	M5.0 – M5.5

En la Figura 3.39 se muestra un fragmento del programa de la subrutina SBR_0 que muestra el envío de las etapas del Grafcet de seguridad, dependiendo qué etapa es la que se encuentre activa es que se escribe un número en específico dentro del byte que corresponde a este Grafcet, en este caso VB1. De esta manera es que el PLC 1 puede comunicar al PLC 2 la etapa activa de cada uno de los Grafcet del proceso.

El segundo programa es aquel encargado de leer los datos de los bytes de las operaciones NETR. Este programa se carga en el PLC secundario, el cual tiene una dirección local 2. El bloque fundamental en este programa se observa en la Figura 3.40, el cual es la llamada a la subrutina NET_EXE, la cual contiene las operaciones de lectura NETR que lee todos los bytes del puerto 0 que escribió previamente el PLC principal.

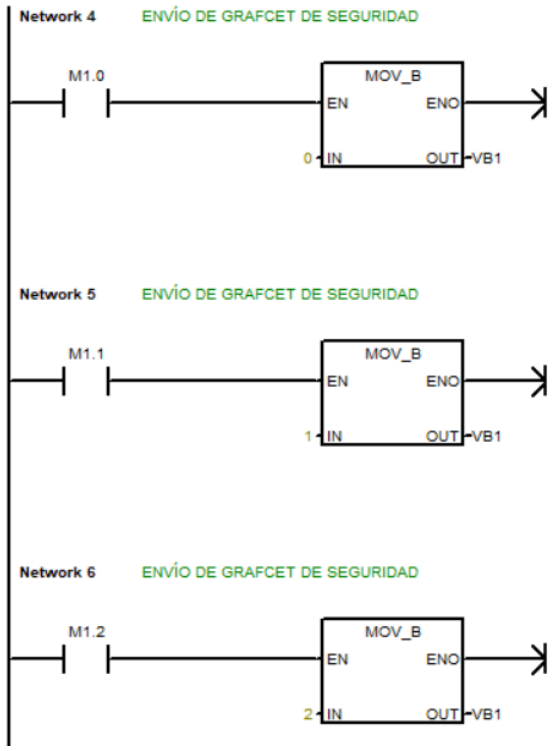


Figura 3.39: Envío de etapa activa.

En la Figura 3.40 se observa que la subrutina se llama en cada ciclo de reloj, debido a la marca especial SM0.0, por lo que los datos de las variables del PLC 1 se encuentran siempre actualizadas, recordando que este PLC envía sus datos de igual manera, en cada ciclo de reloj.

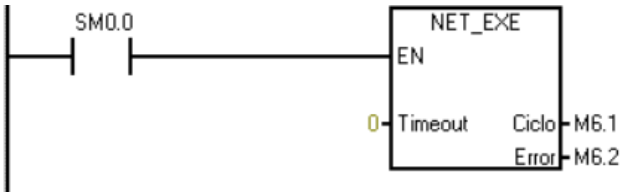


Figura 3.40: Llamada a la subrutina de las operaciones de lectura.

De manera similar al programa del PLC 1, el PLC 2 cuenta con una subrutina, sólo que en esta se compara el valor del byte correspondiente a cada etapa y si el valor coincide con el número de la etapa se activa la marca que la representa. Hay que tener en cuenta que al activarse una etapa de un Grafcet se deben desactivar el resto de sus etapas. La Figura 3.41 muestra un fragmento de esta

subrutina. El programa completo se añade como anexo físico a este documento.

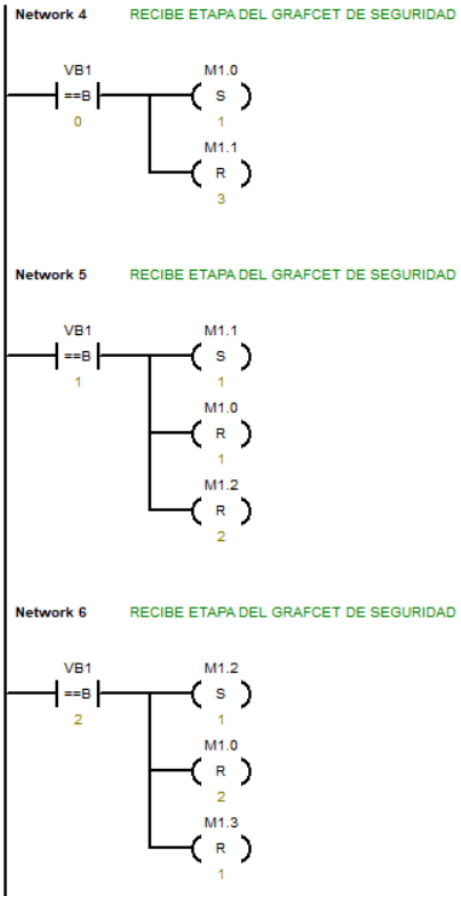


Figura 3.41: Lectura de etapa activa.

3.4.1.1 Falla 1: Desconexión del PLC 1

Una configuración importante que se debe realizar a este programa es que las salidas no pueden mostrarse, hasta saber que ocurrió una falla en el PLC principal, para lo cual se hace uso de las dos entradas y salidas que se dejaron disponibles al momento de asignar los dispositivos a las entradas y salidas del PLC.

En la Tabla 3.7 se muestran las entradas correspondientes al PLC 2, donde las últimas dos entradas constan de una señal constante que se encuentra encendida en todo momento, y una señal intermitente que tiene un ciclo de medio segundo encendido y medio segundo apagado. Estas señales vienen de las salidas que quedaban libres del PLC 1, tal como se observa en la Tabla 3.8.

Tabla 3.7: Entradas del PLC 2

ENTRADAS		
Nombre	**Equipo**	**Dirección PLC**
Marcha (M)	Botón pulsador	I0.0
Paro de emergencia (PE)	Botón paro de emergencia	I0.1
Rearme	Botón pulsador	I0.2
Pieza alimentada (Pieza)	Sensor fin de carrera	I0.3
Cargador extendido (a1)	Sensor fin de carrera	I0.4
Cargador retraído (a0)	Sensor fin de carrera	I0.5
Cilindro extendido (b1)	Sensor fin de carrera	I0.6
Cilindro retraído (b0)	Sensor fin de carrera	I0.7
Fin de rotación del plato (r0)	Sensor fin de carrera	I1.0
Expulsor extendido (d1)	Sensor fin de carrera	I1.1
Expulsor retraído (d0)	Sensor fin de carrera	I1.2
Color verde (cv)	Sensor de color	I1.3
Señal constante	PLC 1	I1.4
Señal intermitente	PLC 1	I1.5

De esta manera, el PLC 1 siempre se encuentra enviando dos señales a las entradas del PLC 2, sin embargo, la que interesa para el caso específico de que ocurra una desconexión total del PLC 1 es la señal constante que se recibe en la entrada I1.4, pues al momento de la desconexión, esta señal se apaga, permitiendo inmediatamente que el PLC 2 active sus salidas para continuar con el proceso de manera normal.

Tabla 3.8: Salidas del PLC 1

SALIDAS		
Nombre	**Equipo**	**Dirección PLC**
Extiende cargador (A+)	Actuador 1	Q0.0
Retrae cargador (A-)	Actuador 1	Q0.1
Extiende cilindro (B+)	Actuador 2	Q0.2
Retrae cilindro (B-)	Actuador 2	Q0.3
Evaluar Pieza (SC)	Arduino UNO	Q0.4
Extiende expulsor (D+)	Actuador 3	Q0.5
Retrae expulsor (D-)	Actuador 3	Q0.6
Gira plato (R+)	Servomotor	Q0.7
Señal constante	PLC 2	Q1.1
Señal intermitente	PLC 2	Q1.2

3.4.1.2 Falla 2: Congelamiento de salidas del PLC 1

El segundo error que se consideró en este trabajo tiene que ver con un problema en las salidas del PLC. Se considera entonces, que los relevadores de las salidas tengan un fallo en su *hardware*, dejando las salidas en un estado fijo. Esto se simula utilizando un botón conectado a la entrada I1.4 del PLC 1, recordando que este aún cuenta con dos entradas disponibles. Al presionar este botón se activa una marca que interrumpe la conexión de los segmentos del programa, dejando el PLC activo en el último estado en que se encontraba antes de presionar el botón, esto incluyendo sus salidas, las cuales permanecen en ese estado.

Para detectar este error se hace uso de la señal intermitente con ciclo de medio segundo encendido y medio segundo apagado, que emite la salida Q1.2 del PLC 1 y es leída por la entrada I1.5 del PLC 2. Dentro del programa de este último PLC se encuentran dos temporizadores que monitorean el tiempo en alto y bajo de esta señal. En la Figura 3.42 se muestra la programación de la lectura de la señal, observando que dependiendo de si la señal ha sufrido un flanco de bajada o de subida

se activa o desactiva la marca M7.0.

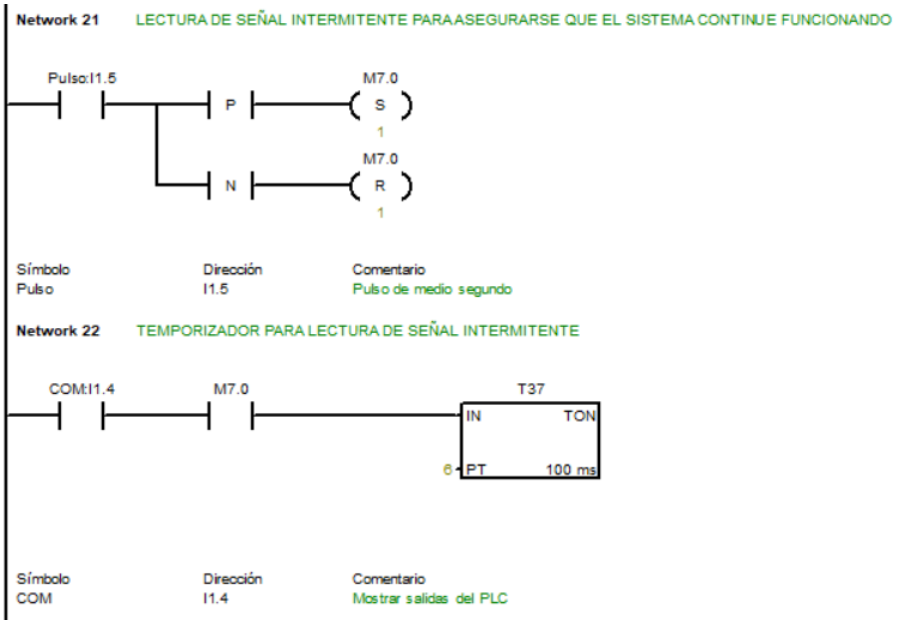


Figura 3.42: Lectura de señal intermitente.

Cuando la marca M7.0 se encuentra activa se enciende un temporizador (T37) de 600 ms. Por otro lado, mientras esta marca se encuentra desactivada se enciende un segundo temporizador (T38) de 600 ms (Figura 3.43). Ambos temporizadores funcionan de manera que no activen su variable correspondiente (T37 y T38) al menos que lleguen al valor de 600 ms, lo cual no debe suceder en caso de que la señal intermitente se encuentre activa.

Al momento de que la señal intermitente sufra un congelamiento y se encuentre ya sea en un estado alto o bajo, se activa la variable del temporizador correspondiente, activando la marca M7.1 y desactivando la marca M7.0, tal como se muestra en la Figura 3.43. Esta nueva marca activa una alarma que se enciende en la salida Q1.1 del PLC 2, la cual le indica al operario que debe de desconectar el PLC 1 debido a que se encuentra con un error de congelamiento en sus salidas.

Cabe recalcar que este tipo de errores pueden ser detectados dentro del programa utilizando la marca especial SM5.0 que se activa en caso de que ocurra algún error en las entradas o salidas del PLC [7], y de esta manera realizar un segmento de programa donde al activarse esta marca especial

el PLC 1 cambie a modo *STOP*, cediendo todo el control al PLC 1. Esto no se incluye en el programa, con la finalidad de observar mejor cómo es que se observa el error de congelamiento, así como apreciar el funcionamiento de la alarma.

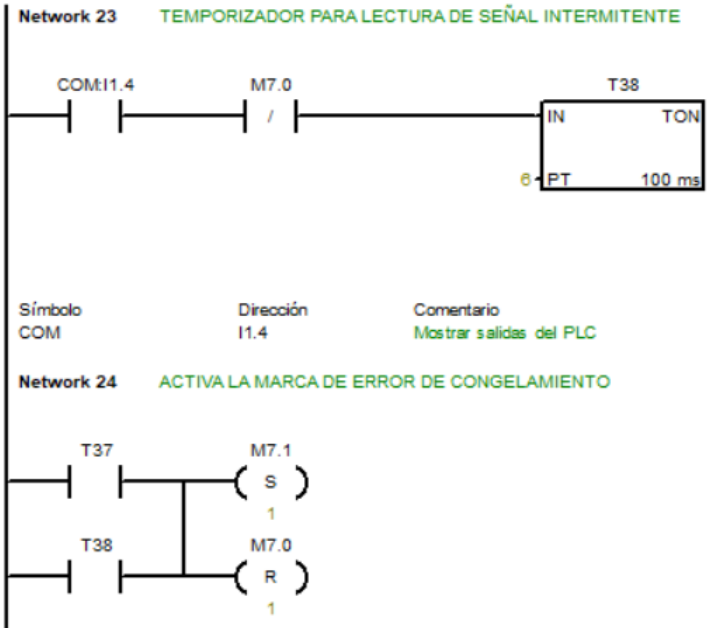


Figura 3.43: Activación de alarma

Por último, es importante recalcar que en la activación de ambos temporizadores de medición de la señal intermitente se incluye la condición de que la entrada I1.4 debe estar activa, pues esta es la señal constante que indica que el PLC 1 aún continúa encendido.

Capítulo 4

Pruebas y resultados

En este capítulo se describen los resultados obtenidos del sistema redundante tipo ***hot-standby*** de las pruebas de desconexión realizadas. Se muestran los tiempos de respuesta para distintas salidas del sistema donde ocurrió una desconexión. Se presentan los resultados medidos con la ayuda de un osciloscopio, el cual permite observar la caída de tensión en una salida durante el cambio de mando de dos PLC. Este se conecta en las salidas del circuito de control, para observar desde un solo canal dicha caída. De igual manera se conectaron los dos canales del osciloscopio hacía las entradas del circuito de control conectadas hacía la misma salida para observar la caída de tensión, pero desde dos canales, donde uno desde el estado en alto presenta una caía y el segundo canal desde el estado en bajo comienza a subir hasta tener el control sobre el componente específico.

De igual manera se muestran las pruebas y resultados obtenidos del error de congelamiento en las salidas del PLC 1, mostrando por medio del osciloscopio el tiempo de respuesta a la caída de la señal intermitente y la activación de la alarma.

4.1 Pruebas de la falla 1: Desconexión del PLC 1

Las pruebas realizadas para comprobar el funcionamiento del sistema redundante consistieron en desconectar el PLC 1 durante distintos momentos en el funcionamiento del sistema para observar su tiempo de respuesta entre la caída de tensión del PLC 1 y la toma de mando del PLC 2. En la primera prueba se desconecta el PLC principal a mitad de un evento temporizado para verificar que no se pierda la cuenta del temporizador. En este caso se considera la señal proveniente de la salida de la compuerta **XOR** que tiene como entrada ambas salidas Q0.4 de cada PLC. Esta señal se activa durante la etapa de descarga por cinco segundos para indicar al Arduino que active el sensor de color. En caso de que el temporizador llegue a su fin se considera que la pieza se encuentra defectuosa y no se extiende el expulsor. En la Figura 4.1 se observa la lectura completa de esta señal medida con la ayuda de un osciloscopio. Se observa que en algún momento el voltaje cae y vuelve a recuperarse, esto debido a la desconexión del PLC 1 que ocurrió durante este tiempo, sin embargo, se observa que a pesar de esto el tiempo que la señal se encuentra en alto es de cinco segundos, es decir exactamente lo que debe durar el temporizador, significando que no ocurrió una pérdida de datos de la cuenta de este mismo.

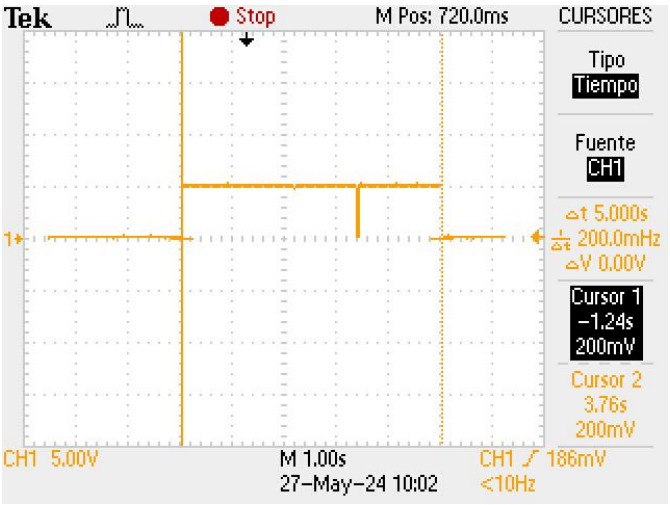


Figura 4.1: Cambio de PLC durante evento temporizado.

La Figura 4.2 presenta un acercamiento a esta caída de voltaje y se puede observar que el tiempo total de la caída es de 20.4 milisegundos. Significando que a pesar de que el voltaje cae un

momento, el PLC 2 se encuentra realizando la operación de temporización desde antes de activar sus salidas.

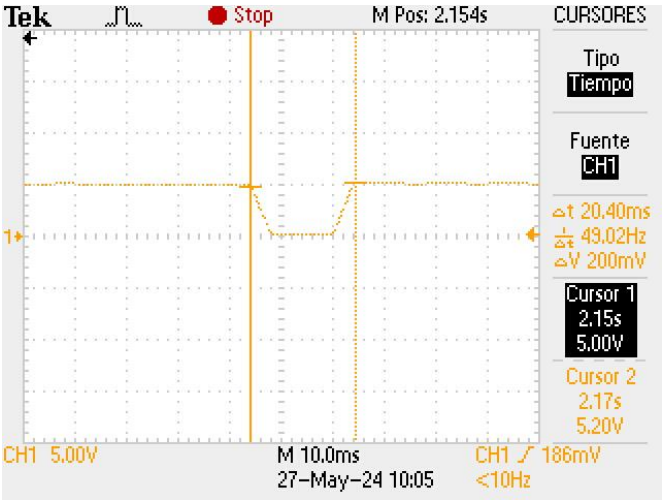


Figura 4.2: Caída de tensión durante evento temporizado.

Para medir el tiempo de respuesta del PLC 2 para tomar el mando en otras partes del sistema se conectó el osciloscopio hacia la salida de la compuerta lógica que se enciende cuando se retrae el cilindro de maquinado. El resultado se observa en la Figura 4.3, donde se observa que la señal tiene una caída de tensión similar a la mostrada en el resultado anterior, esta vez generando un tiempo de respuesta de 30 milisegundos.

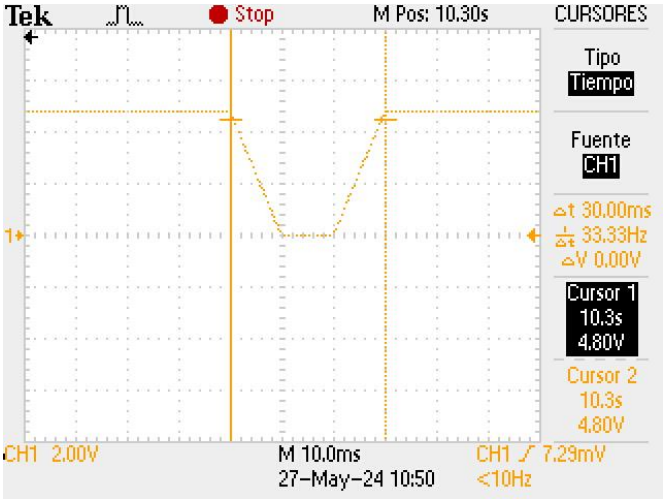


Figura 4.3: Tiempo de respuesta de la extensión del cilindro.

Se realizaron distintas medidas de esta caída de tensión en la salida de cada uno de los actuadores con el fin de obtener una gráfica del promedio del tiempo de respuesta del sistema. En la Figura 4.4 se muestra el tiempo de respuesta de la señal que controla el plato rotatorio.

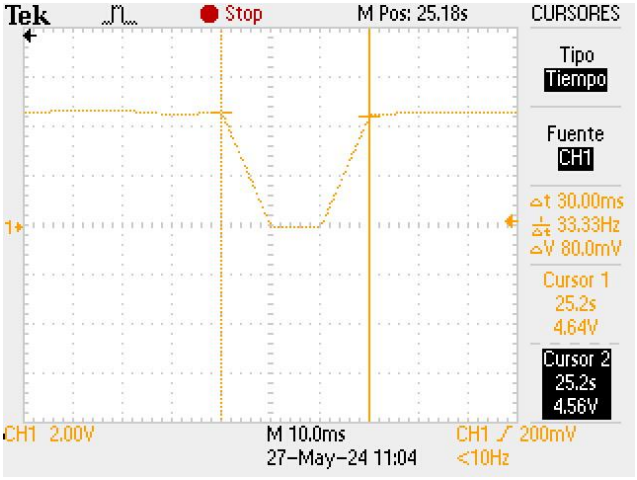


Figura 4.4: Tiempo de respuesta del giro del plato rotatorio.

La Figura 4.5 muestra el cambio de mando de PLC durante el mismo evento temporizado de la Figura 4.1, pero visto utilizando los dos canales del osciloscopio. Esta vez resaltando que el canal 1 es la señal activa del PLC 1 que pasa a desactivarse, al tiempo que la señal del canal 2 (PLC 2) se activa, generando un tiempo de respuesta de 30.4 milisegundos.

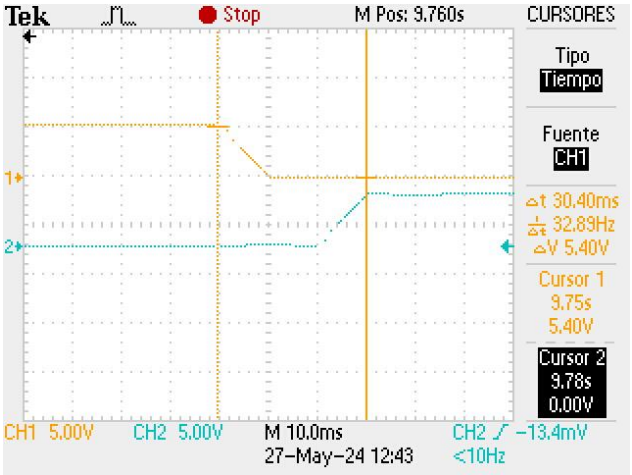


Figura 4.5: Cambio de PLC durante evento temporizado visto desde dos canales.

4.2 Pruebas de la falla 2: Congelamiento de salidas del PLC 1

Para las pruebas del error de congelamiento se utilizan nuevamente los dos canales del osciloscopio. El canal 1 monitorea la salida intermitente del PLC 1, mientras que el canal 2 monitorea la salida Q1.1 del PLC 2, que representa la alarma del sistema, indicando un congelamiento.

La Figura 4.6 muestra el tiempo de un ciclo de la señal intermitente, con un total de un segundo, teniendo medio segundo en alto y medio segundo en bajo. En la Figura 4.7 se observa el tiempo que tarda en activarse la alarma una vez que el sistema se ha quedado congelado en estado bajo. Este tiempo es de 600 milisegundos, justamente el valor del temporizador que monitorea la señal intermitente, pues al cumplir su cuenta sin que haya existido un cambio en esta señal se activa de inmediato la alarma para indicar el congelamiento de las salidas del PLC 1.

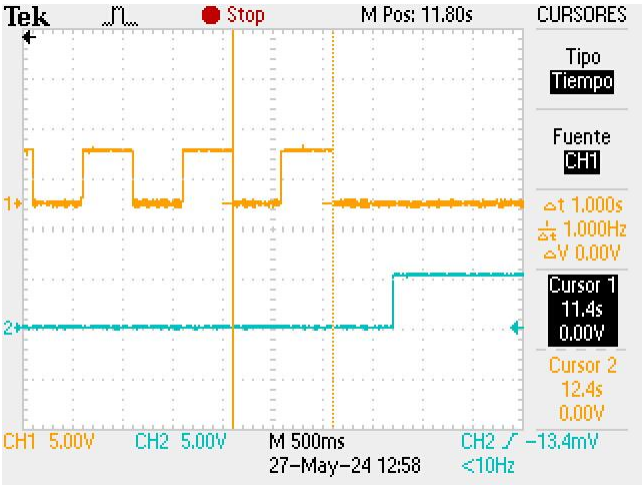


Figura 4.6: Tiempo de ciclo de señal intermitente.

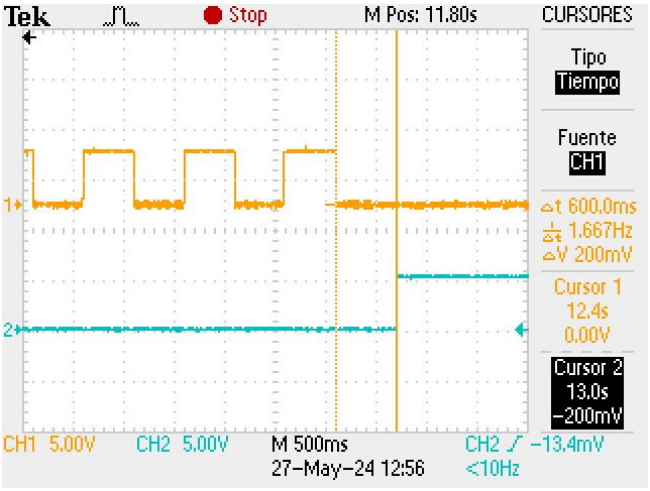


Figura 4.7: Tiempo de respuesta de la alarma

Finalmente, en la Figura 4.8 se observa la reacción al error de congelamiento por parte de los dos PLC montados. Por un lado, el PLC1 (derecha) se encuentra simulando el error de congelamiento, manteniendo sus salidas Q0.0, Q1.1 y Q1.2 activas. Por otro lado, el PLC2 (izquierda) encendió su salida Q1.1, significando que encendió su alarma para alertar el estado de falla del PLC1.

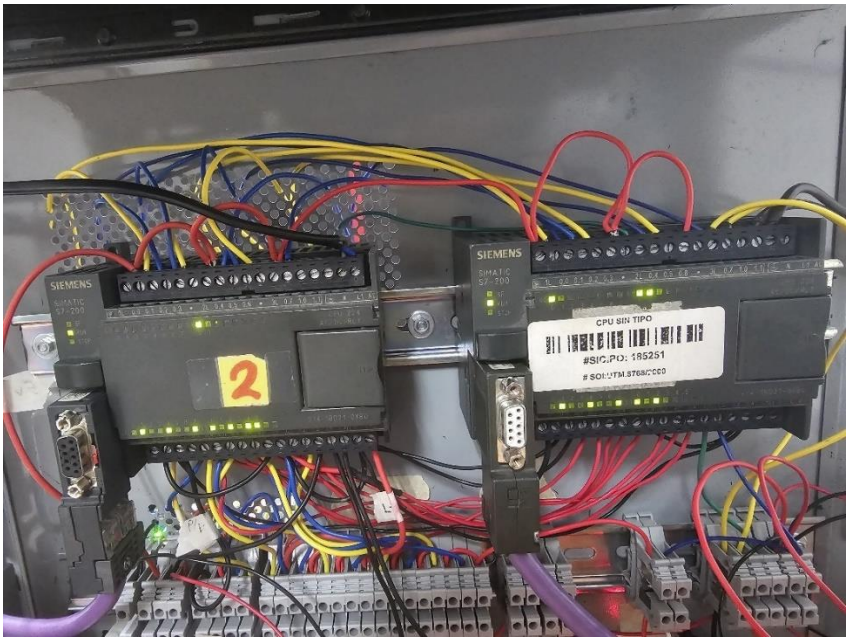


Figura 4.8: Error de congelamiento en el PLC 1.

4.3 Análisis de resultados y discusión

Se realizaron un total de 23 pruebas a lo largo de todas las salidas del circuito de control, con el objetivo de medir distintos tiempos de respuesta en el sistema a la falla de la desconexión del PLC 1. Los resultados se muestran en la gráfica de la Figura 4.9, donde se observa un valor recurrente de 30 milisegundos aproximadamente, con ligeras variaciones a valores de 20 milisegundos. Para observar de manera detallada estos valores, revisar el Anexo 4.

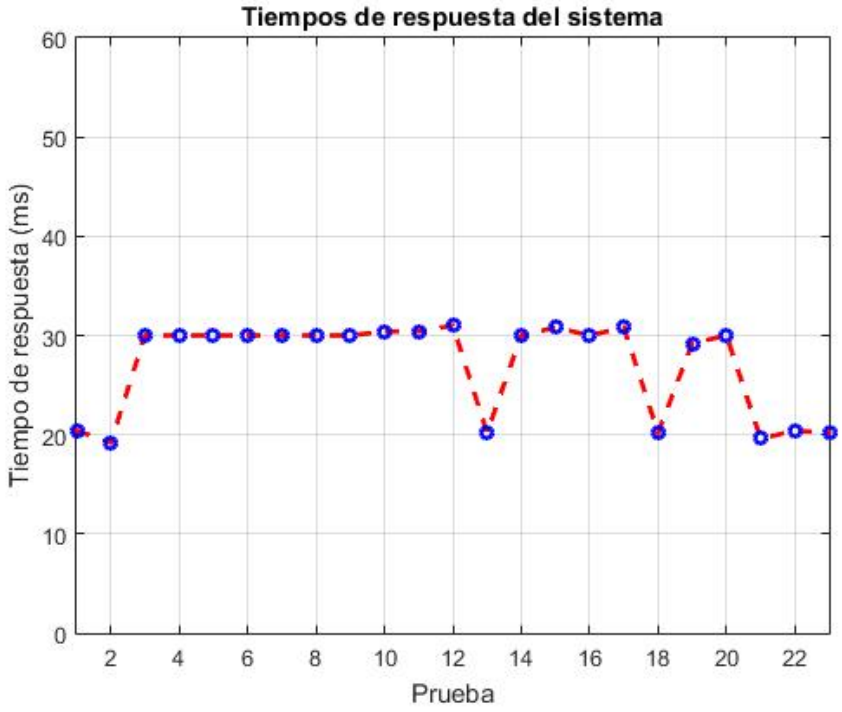


Figura 4.9: Gráfica de tiempos de respuesta.

Desde el *software STEP 7 Micro-Win*® se puede leer el tiempo que tarda el PLC en ejecutar un ciclo del programa, así como el tiempo máximo y mínimo que tarda en ejecutar un ciclo desde que se inició el modo *RUN*. La Figura 4.10 muestra que el tiempo de ciclo es de 2 milisegundos, llegando a ser 5 milisegundos el tiempo de ciclo más largo y 1 milisegundo el más corto.

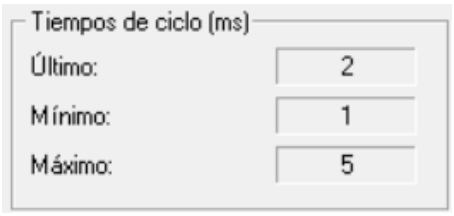


Figura 4.10: Tiempo de ciclo del programa.

Entonces se puede decir que, cuando el tiempo de un ciclo es de 5 milisegundos, ocurren de cuatro a seis ciclos del programa antes de que la salida del sistema pueda estabilizarse después de la desconexión del PLC 1.

Por otro lado, si el tiempo de un ciclo es de 2 milisegundos, ocurrirán de diez a quince ciclos del programa antes de que la salida se estabilice. Los tiempos de funcionamiento dependen de la cantidad de información que se encuentre manejando el programa

Capítulo 5

Conclusiones y trabajos futuros

En este capítulo final, se presentan las conclusiones derivadas del trabajo de investigación realizado. A lo largo de éste, se han abordado diversos conceptos que han servido para contestar las incógnitas planteadas al inicio del documento y en este apartado se resumen los resultados más significativos, destacando la importancia que tuvieron al momento de resolver estas incógnitas.

A lo largo de este capítulo también se analizan las contribuciones fundamentales de la investigación, destacando su relevancia y originalidad.

Por último, se escriben los trabajos de investigación futuros que pueden derivar de este trabajo, sugiriendo posibles direcciones y enfoques que pueden ampliar los conocimientos adquiridos en este estudio. Se incluyen actividades que por falta de tiempo o recursos no se pudieron realizar, sin embargo, se deja en claro que este sistema no es ajeno a recibir modificaciones tanto en su programación como en el prototipo.

A lo largo de este trabajo se llevaron a cabo diversas observaciones que derivaron en distintos resultados y conclusiones significativas. En relación al sistema ***hot-standby***, se logró su construcción utilizando dos micro-PLC CPU S7-200 del fabricante SIEMENS, los cuales fueron conectados entre sí mediante una red PPI. Los resultados obtenidos de este sistema fueron altamente satisfactorios, destacando principalmente los tiempos de respuesta alcanzados, que oscilaron entre 20 y 30 milisegundos, tiempos prácticamente imperceptibles para el ojo humano, asegurando que el sistema continúe operando de manera normal y sin interrupciones notables, incluso ante la desconexión de uno de los PLC. Cabe recalcar que estos tiempos de respuesta no interfieren con las cuentas de los temporizadores, pues estas suceden dentro del programa aún si la salida no ha sido mostrada, asegurando que no se vean afectados tiempos de ejecución en el sistema. Este logro resalta la robustez y eficiencia del sistema ***hot-standby*** implementado, ya que demuestra la capacidad que tiene para mantener la continuidad de la operación y la integridad del proceso.

Por otro lado, el sistema es capaz de soportar un segundo tipo de error, el cual es el congelamiento de las salidas del PLC1, lo cual añadió una nueva capa de complejidad para detectar otro tipo de error sin comprometer el trabajo previamente realizado. Con esto se concluye que la hipótesis planteada al inicio del documento es aceptada, pues el sistema es capaz de soportar dos tipos de falla distintos, siendo estos la desconexión del PLC principal y la avería en sus salidas tipo RLY. En ambos casos de error el sistema es capaz de continuar con la operación del proceso de manera inmediata y sin ninguna pérdida de información.

En cuanto al desarrollo del prototipo físico que simula un proceso tipo transfer circular demostró ser de gran utilidad para la comprensión del funcionamiento de este sistema, pues no sólo permitió visualizar de manera práctica la operación del proceso, sino que además proporcionó un entendimiento más profundo sobre las conexiones físicas necesarias para el funcionamiento del sistema, así como los distintos problemas relacionados al cableado e interconexiones de componentes. Por ejemplo, se logró que dos PLC con salidas distintas puedan controlar un solo sistema, para esto se evaluaron distintas opciones, como el uso de relevadores o una tarjeta de control, para finalmente decidir realizar un circuito con compuertas lógicas para optimizar el espacio de las conexiones del sistema y hacerlo de una manera entendible dentro del diagrama de

conexión.

Otra particularidad con la que cuenta este sistema es que al estar diseñado utilizando la metodología de la guía GEMMA se pueden añadir nuevas estaciones de trabajo con facilidad al sistema, sólo es cuestión de añadir su Grafcet parcial al Grafcet de producción del sistema, además el uso de contadores para indicar la presencia de piezas en cada estación proporciona una manera sencilla de ahorrar en el número de componentes necesario.

En conclusión, las observaciones y pruebas realizadas en este estudio han proporcionado un conocimiento valioso y aplicable en el ámbito industrial. Los PLC S7-200, que continúan siendo ampliamente utilizados en diversas industrias, demostraron su capacidad para poder implementar este tipo de sistemas que optimicen procesos y minimicen los errores, incluidos los que puedan surgir dentro del propio PLC, resaltando así su utilidad en la automatización y control de sistemas industriales.

5.1 Trabajos futuros

En futuros trabajos se tiene considerado el mejorar el prototipo del sistema, pues esta versión es funcional, sin embargo, presenta áreas de oportunidad. Se puede mejorar el plato rotatorio para que admita piezas con un diseño más redondeado para evitar atascamientos de las piezas, así como realizar mejoras en el cableado del sistema, ya sea diseñando nuevas placas PCB que reemplacen ciertos circuitos cableados o expandir el panel de mando para que todo pueda encontrarse dentro y no sea necesario tener partes del circuito en la parte exterior del sistema.

Como se explicó a lo largo de este trabajo, las entradas y salidas del PLC S7-200 CPU 224 presentan una limitación importante, por lo que sería interesante replicar el trabajo utilizando un CPU 226 con mayor número de E/S o utilizar módulos de expansión para poder expandir el sistema o volverlo más complejo, pues esto permitiría hacer uso de todos los componentes del panel de control, incluyendo indicadores, así como el selector de modos de marcha, lo que también representa un aumento en los estados de la guía GEMMA, implicando el crecimiento del programa de control y creando la necesidad de expandir el circuito de control de las salidas, o buscar nuevas

alternativas para lograr que dos dispositivos controlen un sólo sistema.

Finalmente, se evalúa la posibilidad de replicar este mismo sistema utilizando distintos tipos de controladores, como pueden ser los PLC S7-1200 del mismo fabricante SIEMENS, pues sería interesante observar si estos equipos más actualizados proporcionan alguna mejora en los tiempos de respuesta o si obtienen resultados similares a los encontrados en este trabajo.

Bibliografía

[1] S. A. Karim, N. Ranjan, and D. Shah, "A scalable approach to time series anomaly detection & failure analysis for industrial systems," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0678–0683, IEEE, 2020.

[2] M. F. Picó, "Diseño de controladores tolerante a fallas aplicados a procesos de la industria química," 2015.

[3] Z. Wu, N. Li, and Q. Fu, "Design of control system for marine power station based on redundant technology," in *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pp. 174–177, IEEE, 2019.

[4] P. Ponsa Asensio and R. V. i Arbós, *Automatización de procesos mediante la guía GEMMA*, vol. 102. Univ. Politèc. de Catalunya, 2005.

[5] M. P. Groover, *Automation, production systems, and computer-integrated manufacturing*. Pearson Education India, 2016.

[6] A. G. Higuera, *El control automático en la industria*, vol. 48. Universidad de Castilla La Mancha, 2005.

[7] S. AG, "Manual del sistema de automatización s7-200," tech. rep., 6ES7298–8FA24–8DH0, 2008.

[8] J. M. Gea, "Introducción al grafcet," 2006.

[9] J. Guerrero Saiz, *Programación estructurada de autómatas programables con Grafcet*. Ediciones Paraninfo, SA, 2019.

[10] H. Marais, "Rs-485/rs-422 circuit implementation guide," *AN-960 Application Note*. ANALOG DEVICES, 2008.

[11] N. Saboya, "Normas de comunicación en serie: RS-232, RS-422 y RS-485," *Revista Ingenio Libre*, vol. 9, no. 1, pp. 86–94, 2012.

[12] N. A. Ángel Cabarcas and W. J. Cantillo Pájaro, "Laboratorio de redes industriales con PLCs Siemens S7-200," 2008.

[13] SIEMENS, "Conector de bus profibus: Manual del producto," tech. rep., A5E50543132-AB, 2022.

[14] *SIEMENS, "Hoja de datos 6es7972-0ba30-0xa0," tech. rep., 2023.*

[15] R. Barkur and S. Shriram, "An expeditious method for implementing a full redundant drive in hils with the use of plc and user interface panel," in *2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, pp. 1–5, IEEE, 2017.

[16] J. Ždánsky and K. Rástočn`, "Influence of redundancy on safety integrity of srcs with safety plc," in *2014 ELEKTRO*, pp. 508–512, IEEE, 2014.

[17] G. Levitin, L. Xing, and Y. Dai, "Optimal design of hybrid redundant systems with delayed failure-driven standby mode transfer," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1336–1344, 2015.

[18] G. Chen, M. Xu, T. Liu, J. Ni, D. Xie, and Y. Zhang, "Intelligent control system of transformer cooling based on dcs and dual plc," in *2013 Fifth International Conference on Measuring Technology and Mechatronics Automation*, pp. 648–651, IEEE, 2013.

[19] S. Singh, V. M. Chary, and P. A. Rahman, "Dual redundant profibus network architecture

in hot standby fault tolerant control systems," in *2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014)*, pp. 1–5, IEEE, 2014.

[20] S. A. Ruiz, E. R. Beltrán, J. A. P. Olivares, J. L. O. Mora, and J. O. V. Valadez, "Diagnóstico de fallas intermitentes en sistemas automáticos de manufactura usando machine learning (diagnosis of intermittent failures in automatic manufacturing systems using machine learning)," *Pistas Educativas*, vol. 44, no. 144, 2023.

[21] Z. G. D. Armando, "Sistema de diagnóstico remoto de fallas de plc en máquinas automatizadas," 2014.

[22] J. Pelegrí Aldavert, "Implementación de un sistema redundante en un proceso industrial: Enclavamientos para una caldera en entorno seguro," 2016.

[23] P. E. Moreda, "Ingeniería de manufactura," 2020.

[24] R. E. Jaspe Lombana and A. D. Mosquera Amaya, "Grafcet aplicado al diseño de automatismos con PLC S7-200," 2007.

[25] A. Orantes-Molina, J. A. Juárez-Abad, J. Linares-Flores and R. Miranda-Luna, "Automation of a drilling process via a Wireless connection using the EMQX protocol," in *Memorias del 2022 Congreso Nacional de Control Automático ISSN: 2594-2492*, pp. 404-409, AMCA, 2022.

[26] DFRobot, "Tcs2300 color sensor," tech. rep., SKU:SEN0101, 2017.

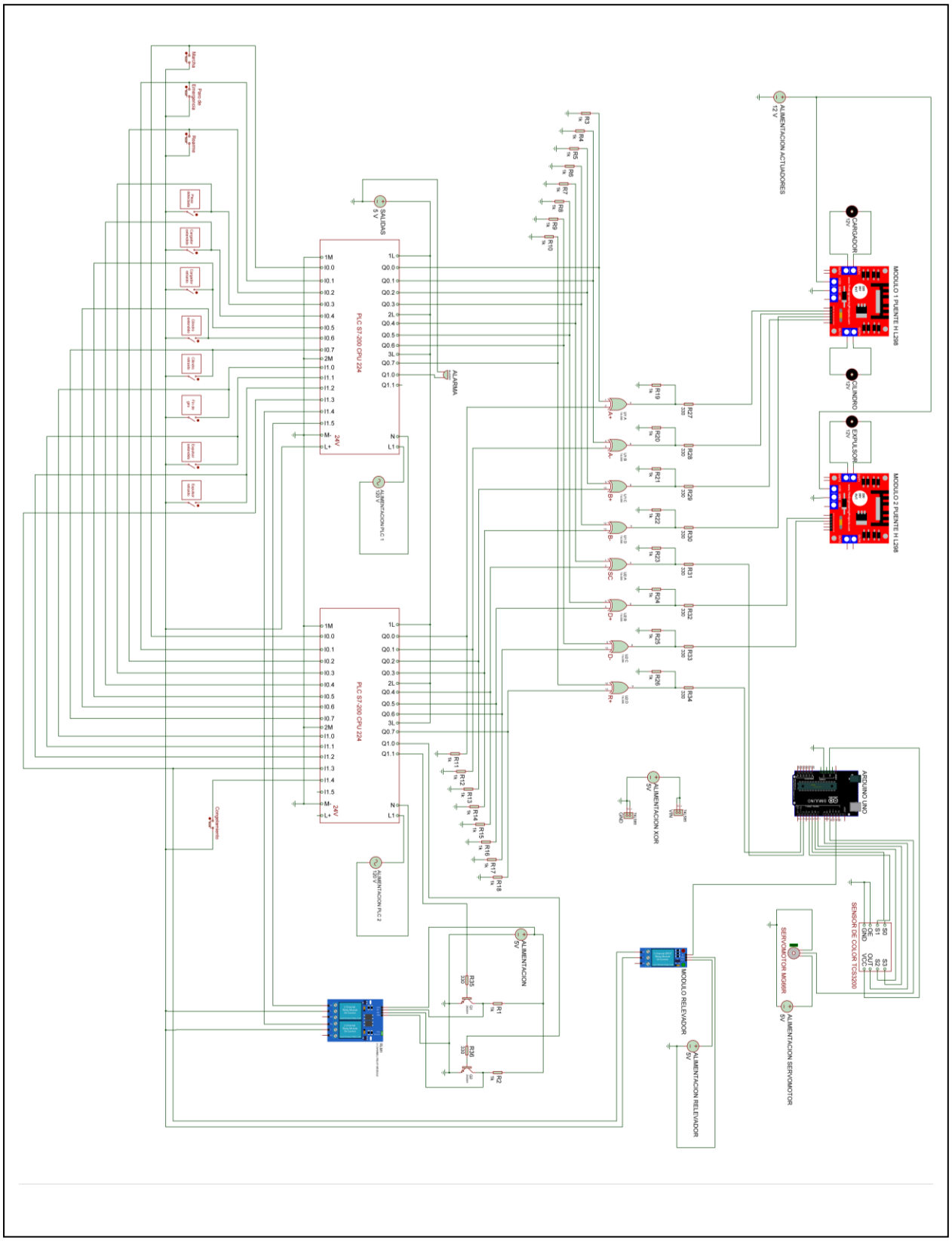
[27] Vistrónica, "Servomotor mg996r de rotación continua."

[28] L. R. . Electronics, "Hoja de datos del controlador de motores l298n (rojo)," tech. rep.

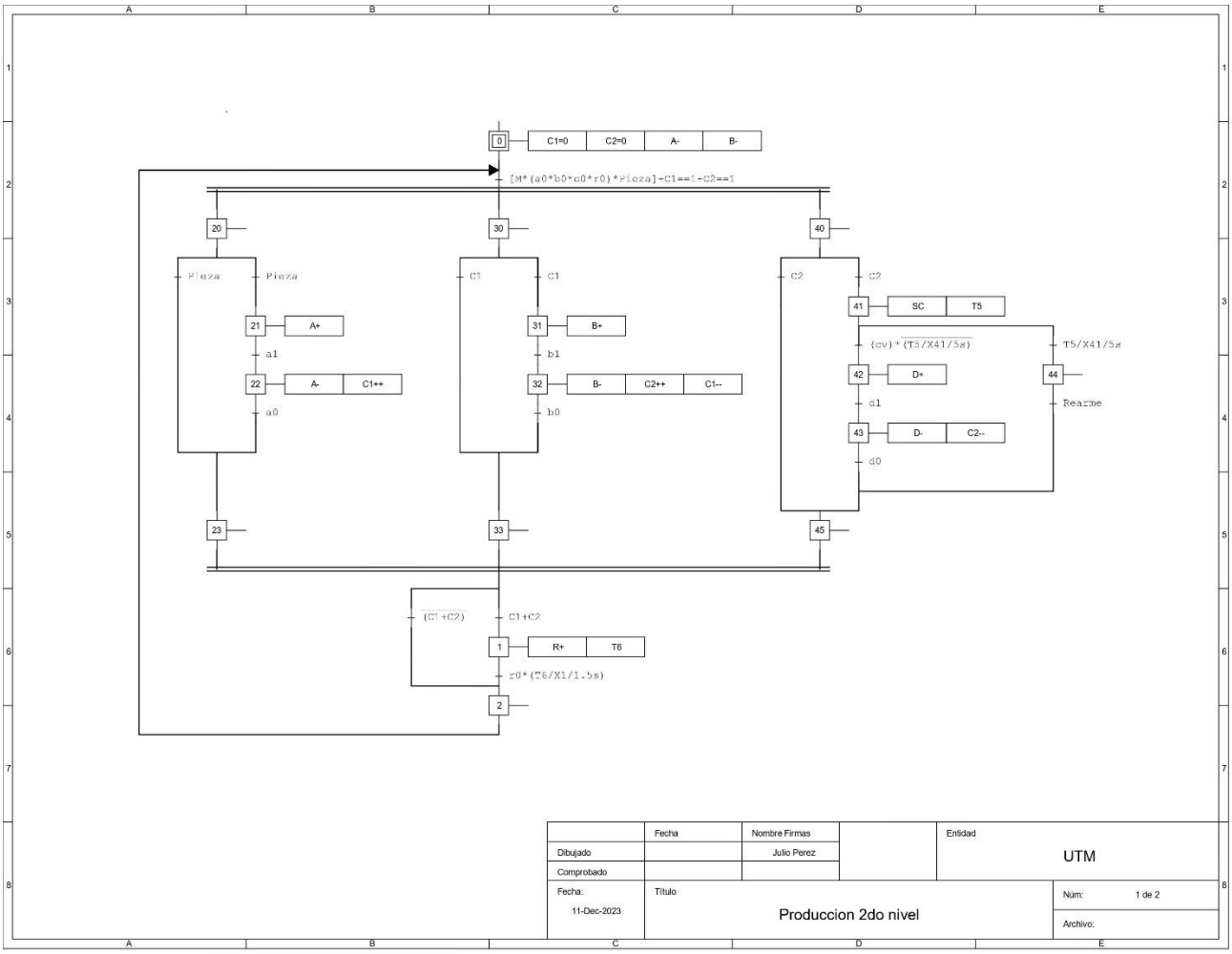
[29] Fairchild Semiconductor, "Dm74ls86 Quad 2-input exclusive-or gate", tech. rep., DS006380, 2000.

ANEXOS

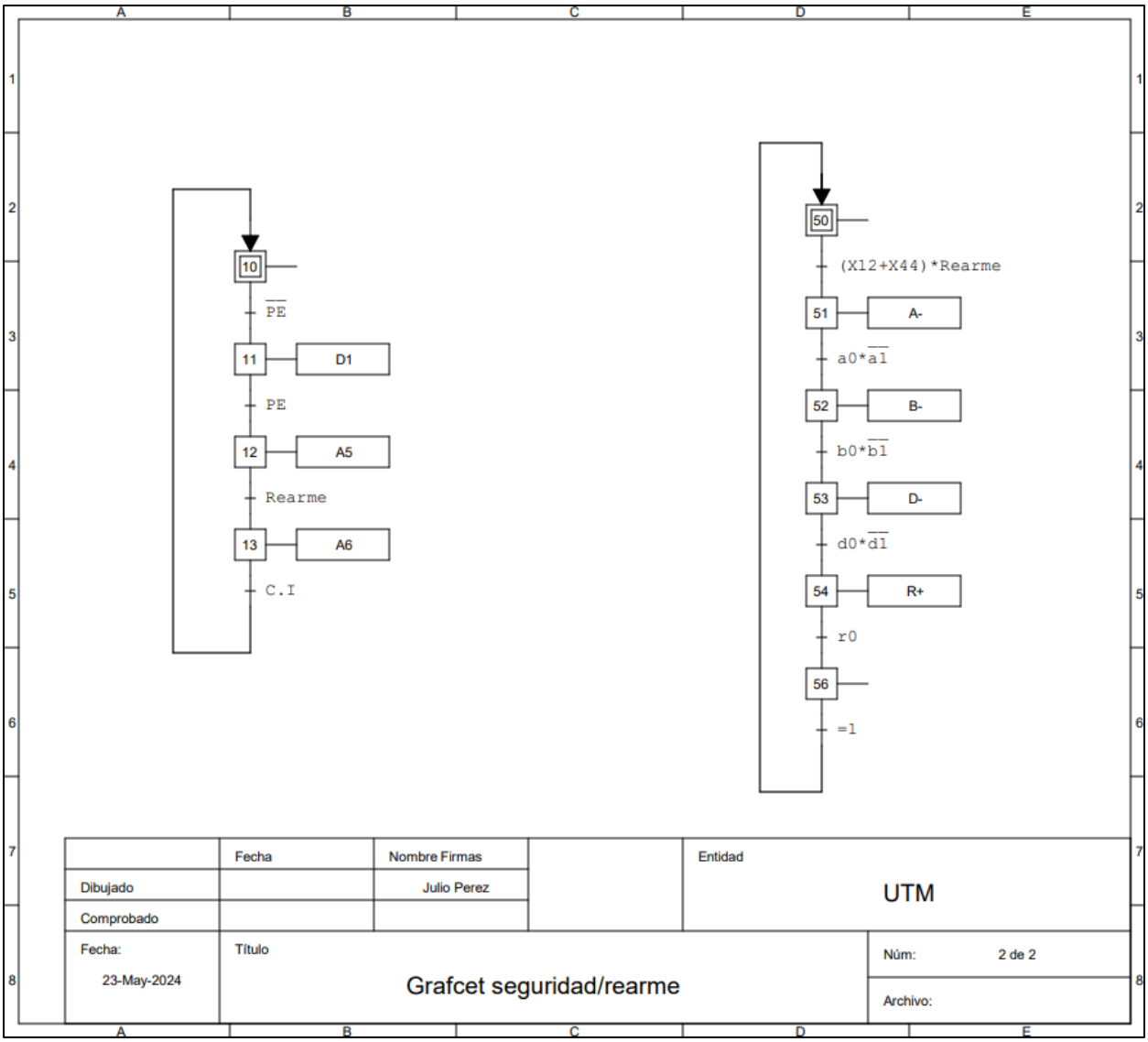
Anexo 1: Circuito del sistema



Anexo 2: Grafcet de producción de segundo nivel



Anexo 3: Grafcet de seguridad y rearme del Sistema



Anexo 4: Tabla de resultados obtenidos

Número de prueba	Tiempo obtenido (ms)
1	20.4
2	19.2
3	30
4	30
5	30
6	30
7	30
8	30
9	30
10	30.4
11	30.4
12	31
13	20.2
14	30
15	30.8
16	30
17	30.8
18	20.2
19	29.2
20	30
21	19.6
22	20.4
23	20.2

Anexo 5: Código de programación de la tarjeta Arduino UNO

```
#include <Servo.h> //Libreria para control de servomotor

#define RELE_ON 0 //Cero logico activa modulo de relevador
#define RELE_OFF 1 //Uno logico desactiva modulo de relevador

const int gira = 1;//Pin de entrada de señal que activa giro de servo
const int color = 2;//Pin de entrada de señal que activa sensor de color

const int s0 = 4;//Pines de
const int s1 = 5;//control
const int s2 = 6;//del sensor
const int s3 = 7;//de color

const int out = 8;//Lectura del sensor de color

const int rele = 12;//Pin de control del modulo de relevador

Servo myservo;// crea el objeto servo

int vel = 0;// velocidad del servo

void setup()
{
  /* Entradas y salidas del sistema */
  pinMode(gira, INPUT);
  pinMode(color, INPUT);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);
  pinMode(rele, OUTPUT);

  myservo.attach(10);// vincula el servo al pin digital 10

  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);
```

```
  pinMode(LED_BUILTIN, OUTPUT);//LED integrado como salida
  digitalWrite(rele, RELE_OFF); //RELE apagado a menos que el color registrado sea verde
  vel = 90; //Servomotor detenido
  myservo.write(vel);
}

void loop()
{
  bool e = digitalRead(color);//Lee estado del pin "color"
  bool i = digitalRead(gira);//Lee estado del pin "gira"
  digitalWrite(rele, RELE_OFF);//Mantiene modulo relevador apagado

  if(e==1){//Se requiere lectura del sensor de color
    int R = getRojo();//Llamada a la función para leer el color rojo
    delay(200);
    int V = getVerde();//Llamada a la función para leer el color verde
    delay(200);
    int A = getAzul();//Llamada a la función para leer el color azul
    delay(200);


    if (R < V && V > A && R >= 28 && V >= 48){ //Parametros que indican que el color es
rojo
      digitalWrite(rele, RELE_OFF);//Mantiene desactivo modulo relevador
      digitalWrite(LED_BUILTIN, LOW);//Mantiene apagado el LED integrado
    }
    else if (R >= V && R > A && R >= 48 && V >=48){//Parametros que indican que el color
es verde
      digitalWrite(rele, RELE_ON);//Activa modulo de relevador
      digitalWrite(LED_BUILTIN, HIGH);//Enciende el LED integrado
    }
    else //Color no registrado
    {
       digitalWrite(rele, RELE_OFF);//Mantiene desactivo modulo relevador
       pinMode(LED_BUILTIN, OUTPUT);//Mantiene apagado LED integrado
    }

  }
  if(i==1){//Se requiere giro del servomotor
     vel = 75; //Servomotor EN MARCHA a velocidad lenta en sentido de las manecillas del reloj
     myservo.write(vel);
     digitalWrite(LED_BUILTIN, HIGH);   //Enciende el LED integrado
  }
```

```
  if(i==0){ //Se requiere detener el servomotor
    vel = 90; //servomotor detenido
    myservo.write(vel);
    digitalWrite(LED_BUILTIN, LOW);   //Apaga el LED integrado
  }

}

int getRojo(){ //Funcion para leer el color rojo
  //leer color rojo
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  int ROJO = pulseIn(out, LOW);
  return ROJO;
}

int getAzul(){ //Funcion para leer el color azul
  //leer color azul
  digitalWrite(s2, LOW);
  digitalWrite(s3, HIGH);
  int AZUL = pulseIn(out, LOW);
  return AZUL;
}

int getVerde(){ //Funcion para leer el color verde
  //leer color verde
  digitalWrite(s2, HIGH);
  digitalWrite(s3, HIGH);
  int VERDE = pulseIn(out, LOW);
  return VERDE;
}
```