



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

“ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
INALÁMBRICO CON PROTOCOLO DE COMUNICACIÓN UDP
ENTRE EL ROBOTINO, BRAZO ROBÓTICO Y LA MESA DE
TRABAJO PARA REALIZAR PROCESOS DE PALETIZACIÓN”

Trabajo de titulación

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, CONTROL Y REDES
INDUSTRIALES

AUTOR: PAUCAR ATI ALEX JAVIER

DIRECTOR: ING. VITERI BARRERA MARCO ANTONIO

Riobamba – Ecuador

2020

©2020, Alex Javier Paucar Ati.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES**

El tribunal del trabajo de titulación certifica que: El Trabajo de Titulación: Proyecto Técnico “ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA INALÁMBRICO CON PROTOCOLO DE COMUNICACIÓN UDP ENTRE EL ROBOTINO, BRAZO ROBÓTICO Y LA MESA DE TRABAJO PARA REALIZAR PROCESOS DE PALETIZACIÓN”, de responsabilidad del señor **ALEX JAVIER PAUCAR ATI**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, quedando autorizada su presentación.

FIRMA

FECHA

(año/mes/día)

Ing. Jorge Hernández

PRESIDENTE DEL TRIBUNAL

Ing. Marco Viteri Barrera.

**DIRECTOR DEL TRABAJO DE
TITULACIÓN**

Ing. Fausto Cabrera Aguayo.

MIEMBRO DEL TRIBUNAL

Yo, Alex Javier Paucar Ati soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación, y el patrimonio intelectual de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

Alex Javier Paucar Ati

Yo Alex Javier Paucar Ati declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba 29 de enero de 2020

Alex Javier Paucar Ati

060493776-3

DEDICATORIA

El presente trabajo investigativo lo dedico a mis padres Carmen y Olmedo, quienes son el pilar fundamental de mi vida, con su esfuerzo dedicación y sacrificio ha logrado forjarme como profesional y persona. A mi hermana y sobrino, quienes son las personas que me motivan a seguir adelante. Gracias por permanecer en todo momento y darle alegría a mi vida.

Alex Javier Paucar Ati

AGRADECIMIENTO

Ante todo, inicio agradeciendo a Dios por darme salud, vida y permitirme compartir este proceso junto a mis seres queridos. Agradezco a mis padres: Carmen y Olmedo por ser siempre mi apoyo incondicional y por el esfuerzo infinito para darme todo lo necesario. A mi hermana y sobrino: Jhoselin y Jhair por ser el motivo a seguir adelante, por alegrar siempre mi vida y por demostrarme como la vida sigue a pesar de las adversidades. También agradezco a mis tíos Luis, Ana, Elsa, Iván y Klever por ser uno de los pilares fundamental este proceso, por su apoyo económico y moral, y más aún por siempre demostrarme el verdadero valor de la familia.

Alex Javier Paucar Ati

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	viii
ÍNDICE DE GRÁFICOS.....	xiv
ÍNDICE DE FIGURAS.....	xv
ÍNDICE DE ANEXOS.....	xviii
RESUMEN.....	xix
ABSTRACT.....	xx
INTRODUCCIÓN.....	1
CAPÍTULO I	
1. MARCO TEÓRICO REFERENCIAL.....	6
1.1 Generalidades de la Robótica.....	6
1.2 Robots móviles.....	6
1.3 Robots manipuladores y Modelo biomecánico del brazo humano.....	5
1.3.1 <i>Robotino® Festo</i>	7
1.3.1.1 <i>Datos Técnicos del Robotino®</i>	8
1.3.1.2 <i>Diseño y función</i>	8
1.3.1.3 <i>Robotino Objetivos de enseñanza</i>	9
1.3.1.4 <i>El módulo de la unidad de accionamiento</i>	9
1.3.1.5 <i>La unidad de control</i>	9
1.3.1.6 <i>Módulo tarjeta de circuito de E/S</i>	10
1.3.1.7 <i>Fuente de alimentación/cargador de batería</i>	10
1.3.1.8 <i>Sensores</i>	11
1.3.1.9 <i>Sensores de medición de distancia por infrarrojos</i>	11
1.3.1.10 <i>Encoder incremental</i>	11
1.3.1.11 <i>El sensor anticolidión</i>	11
1.3.1.12 <i>Sensor de proximidad inductivo analógico</i>	12
1.3.1.13 <i>Sensores de reflexión directa</i>	12
1.3.1.14 <i>Posiciones de los sensores y sus denominaciones en Robotino® View</i>	12

1.3.1.15	<i>Punto de acceso LAN inalámbrico.....</i>	13
1.3.1.16	<i>La tarjeta compact flash.....</i>	13
1.3.1.17	<i>El interfaz E/S.....</i>	14
1.3.1.18	<i>Control e interfaces.....</i>	14
1.3.1.19	<i>Software.....</i>	15
1.3.1.20	<i>Ampliaciones.....</i>	15
1.3.1.21	<i>Robustez y movilidad.....</i>	15
1.3.1.22	<i>Construcción flexible y adaptable.....</i>	16
1.3.1.23	<i>Plug and Play</i>	16
1.3.1.24	<i>Entorno de programación abierto del Robotino®.....</i>	17
1.3.2	<i>Brazo robótico RV2SDB.....</i>	17
1.3.2.1	<i>Descripción general.....</i>	17
1.3.2.2	<i>Movilidad</i>	18
1.3.2.3	<i>Eficiencia.....</i>	18
1.3.2.4	<i>Integración sencilla en sistemas</i>	19
1.3.2.5	<i>Herramientas de Programación</i>	19
1.3.2.6	<i>Rango de movimiento.....</i>	20
1.3.2.7	<i>Datos técnicos.....</i>	21
1.3.2.8	<i>El cómodo terminal de operador para aprendizaje móvil.....</i>	22
1.3.2.9	<i>Operaciones Básicas.....</i>	23
1.4	<i>Controlador lógico programable “PLC”</i>	24
1.4.1	<i>PLC SIEMENS S7 -1200</i>	25
1.4.1.1	<i>Entradas y salidas digitales integradas</i>	26
1.4.1.2	<i>Memoria de usuario integrada memoria de usuario.....</i>	26
1.4.1.3	<i>Rendimiento.....</i>	27
1.4.1.4	<i>Corriente Necesaria.....</i>	28
1.4.1.5	<i>Comunicaciones a todos los niveles.....</i>	30
1.4.1.6	<i>Interfaz PROFINET Integrada</i>	30
1.4.1.7	<i>Comunicación Punto-a-Punto</i>	30

1.5	Protocolos de Comunicación.....	30
1.5.1	<i>Diseño de protocolo.....</i>	30
1.5.1.1	<i>Pasos</i>	31
1.5.1.2	<i>Arquitectura TCP/IP</i>	31
1.5.2	<i>Protocolo UDP</i>	32
1.5.2.1	<i>Características principales.....</i>	32
1.5.2.2	<i>Funciones de los campos de la cabecera de UDP.....</i>	32
1.5.2.3	<i>Aplicaciones de UDP</i>	32
1.5.2.4	<i>La importancia de UDP</i>	32
1.5.2.5	<i>Bus de comunicación.....</i>	32
1.6	Comunicación Wi-Fi	37
1.7	Procesos de Paletización.....	37
CAPÍTULO II		
2.	MARCO METODOLÓGICO	37
2.1	Introducción.....	39
2.2	Diagnóstico del estado actual de área de trabajo y materiales.....	39
2.3	Definición de Requerimientos del sistema	40
2.3.1	<i>Rapidez en intercambio de datos</i>	41
2.3.2	<i>Comunicación inalámbrica:.....</i>	42
2.3.3	<i>Controlador lógico programable con comunicación UDP</i>	42
2.3.4	<i>Robot con protocolo de comunicación UDP.....</i>	42
2.3.5	<i>Robots con características específicas</i>	42
2.3.5.1	<i>Programables</i>	42
2.3.5.2	<i>Suministros de voltaje</i>	44
2.3.5.3	<i>Alta calidad de cámara</i>	44
2.3.5.4	<i>Sensores de distancia</i>	46
2.3.5.5	<i>Suficientes grados de libertad.....</i>	47
2.3.5.6	<i>Alto grado de precisión en posicionamiento y movilidad de Robots</i>	48
2.3.5.7	<i>Módulo de interfaz para comunicación con Robots.....</i>	49

2.4	Detalles de costos.....	54
2.5	Características del Área de Trabajo.....	56
2.6	Condiciones del Área de Trabajo	56
2.7	Adecuación de la Mesa de Trabajo.....	57
2.8	Montaje de Ampliación de Robotino	60
2.9	Montaje del PLC de mando	61
2.9.1	<i>Estructura o arquitectura General del PLC.....</i>	<i>62</i>
2.9.2	<i>Asignación de dirección</i>	<i>63</i>
2.9.3	<i>Lenguaje de programación.....</i>	<i>64</i>
2.10	Adaptación de la Unidad de Interfaz de Robot 534364.....	65
2.10.1	<i>Acople.....</i>	<i>65</i>
2.11	Puesta en marcha de Robots	67
2.11.1	<i>Brazo Robótico Mitsubishi RV2SDB.....</i>	<i>68</i>
2.11.1.1	<i>Colación de baterías y conexión a suministro de voltaje</i>	<i>68</i>
2.11.1.2	<i>Programación en software</i>	<i>68</i>
2.11.2	<i>Puesta en marcha del ROBOTINO PLUS.....</i>	<i>68</i>
2.11.2.1	<i>Colación de baterías y conexión a suministro de voltaje</i>	<i>68</i>
2.11.2.2	<i>Interfaz Entradas y Salidas.....</i>	<i>72</i>
2.11.2.3	<i>Cámara del Robot</i>	<i>72</i>
2.11.2.4	<i>Programación en software</i>	<i>73</i>
2.12	Establecimiento de la comunicación UDP	77
2.12.1	<i>Configuración PLC.....</i>	<i>77</i>
2.12.2	<i>Configuración Robotino.....</i>	<i>85</i>
2.12.3	<i>Codificación Python.....</i>	<i>89</i>
2.12.4	<i>Establecimiento comunicación UDP PLC – Robotino.....</i>	<i>90</i>
2.13	Diseño y Programación de la HMI	92
2.14	Funcionamiento de sistema	93
2.14.1	<i>Diagrama de Funcionamiento Robotino</i>	<i>93</i>
2.14.2	<i>Diagrama de funcionamiento Brazo Robótico.....</i>	<i>94</i>

2.14.3	<i>Secuencia</i>	95
CAPÍTULO III		
3.	MARCO METODOLÓGICO	98
3.1	Introducción	98
3.2	Prueba de comunicación UDP	102
3.3	Prueba de comunicación Caja de Interfaz con Brazo Robótico	103
3.4	Prueba de detección piezas con bit 900	105
3.5	Prueba del Algoritmo de procesamiento de imágenes	106
3.6	Pruebas de tiempos de ejecución de procesos	108
3.7	Pruebas del Algoritmo de Posicionamiento	108
3.8	Prueba de funcionamiento del sistema	108
3.9	Nivel de Automatizado del Sistema	114
3.10	Prueba de Funcionamiento según el nivel de batería del Robotino	115
3.11	Aplicación de Prueba Estadística	115
CONCLUSIONES		116
RECOMENDACIONES		117
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-1: Datos técnicos del Robotino®	8
Tabla 1-2: Comparación de características de Protocolos.....	39
Tabla 2-2: Salidas con sus respectivos pines.....	50
Tabla 3-2: Entradas con sus respectivos pines	51
Tabla 4-2: Datos de Unidad de Interfaz de Robot	51
Tabla 5-2: Detalles de presupuesto.....	52
Tabla 6-2: Conexiones a PLC.....	62
Tabla 7-2: Pasos y Puntos de la ruta del Robot Mitsubishi RV2SDB con Home 1	67
Tabla 8-2: Pasos y Posiciones de la ruta del Robot Mitsubishi RV2SDB con Home 2.....	67
Tabla 1-3: Pruebas sometidas a comunicación UDP	98
Tabla 2-3: Pruebas sometidas a comunicación Caja Interfaz- Brazo Robótico.....	99
Tabla 3-3: Pruebas detección de pieza Bit 900.....	101
Tabla 4-3: Prueba algoritmo procesamiento de imágenes.....	102
Tabla 5-3: Prueba tiempo programa-real	103
Tabla 6-3: Pruebas tiempo de cada proceso	104
Tabla 7-3: Pruebas algoritmo de posicionamiento para X	109
Tabla 8-3: Pruebas algoritmo de posicionamiento para Y	106
Tabla 9-3: Pruebas algoritmo de posicionamiento para phi	107
Tabla 10-3: Número de pallets recogidos a la entrada del sistema	108
Tabla 11-3: Número de pallets colocados en home 4 o entrada del Robot RV2SDB.....	108
Tabla 12-3: Número de pallets colocados en la salida del Robot RV-2SDB	109
Tabla 13-3: Número de pallets colocados en la Salida de Sistema.....	109
Tabla 14-3: Procesos involucrados en el sistema de comunicación.....	110
Tabla 15-3: Prueba Nivel de batería	111
Tabla 16-3: Prueba CHI cuadrada de bondad de ajuste en caso comunicación UDP	113
Tabla 17-3: Prueba CHI cuadrada de bondad de ajuste en caso procesos automáticos y manuales	114

ÍNDICE DE GRÁFICOS

Gráfico 1-1: Rango de trabajo de Robot	21
Gráfico 2-1: Especificaciones y detalles del RV-2SDB.....	22
Gráfico 3-1: Cuadro comparativo de acuerdo al número de E/S	27
Gráfico 1-2: Etapas de Análisis e Implementación de un sistema inalámbrico.....	53
Gráfico 2-2: Diagrama generalizado de un PLC	59
Gráfico 3-2: Proceso de comunicación inalámbrica controlado desde un PLC.....	60
Gráfico 4-2: Diagrama de bloques del funcionamiento de Robotino.....	90
Gráfico 5-2: Diagrama de bloques del funcionamiento Brazo Robótico	91
Gráfico 6-2: Funcionamiento total de sistema.....	96
Gráfico 1-3: Prueba comunicación UDP.....	99
Gráfico 2-3: Prueba comunicación Caja de interfaz - Brazo Robótico	100
Gráfico 3-3: Prueba Detección de piezas Bit 900.....	101
Gráfico 4-3: Prueba algoritmo procesamiento de imágenes	103
Gráfico 5-3: Variación de incertidumbre tiempo real - programa	104
Gráfico 6-3: Variación incertidumbre posicionamiento X.....	106
Gráfico 7-3: Variación incertidumbre posicionamiento Y	107
Gráfico 8-3: Variación incertidumbre posicionamiento phi.....	108
Gráfico 9-3: Prueba pallets salida sistema	109
Gráfico 10-3: Nivel de automatización de los procesos.....	111
Gráfico 11-3: Prueba nivel de batería	112

ÍNDICE DE FIGURAS

Figura 1-1: Robotino Festo Plus.....	7
Figura 2-1: Esquema del módulo de accionamiento.....	8
Figura 3-1: Esquema de módulo del controlador de Robotino.....	11
Figura 4-1: Ubicación de los sensores	12
Figura 5-1: Ubicación de los sensores	13
Figura 6-1: Asignación de bornes del interface E/S	14
Figura 7-1: Movilidad del Robot.....	15
Figura 8-1: Flexibilidad y adaptabilidad del robot	16
Figura 9-1: Plug and Play del robot.....	16
Figura 10-1: Brazo Robótico Mitsubishi RV-2SDB.....	17
Figura 11-1: Alcance del espacio de trabajo del Brazo Robótico.....	18
Figura 12-1: Integración del RV-2SDB en diferentes aplicaciones.....	19
Figura 13-1: Herramientas de programación del RV-2SDB	20
Figura 14-1: Panel de operador del RV-2SDB.....	23
Figura 15-1: Operaciones Del Sistema Articulado.....	23
Figura 16-1: Operaciones en el Sistema Cartesiano de Coordenadas.....	24
Figura 17-1: Conexión de un producto o proceso industrial a un controlador lógico.....	25
Figura 18-1: Conexión de un controlador lógico a un producto o un proceso industrial.....	25
Figura 19-1: Autómata Programable S/-1200 de Siemens.....	26
Figura 20-1: Esquema de Comunicación Punto a Punto.....	29
Figura 21-1: Conjunto de protocolos en la arquitectura de red TCP/IP.....	31
Figura 22-1: Representación de capas o niveles OSI y TCP/IP	32
Figura 23-1: Formato de la cabecera del protocolo UDP.....	33
Figura 1-2: Estado actual del área de trabajo	37
Figura 2-2: Panorama general de software de Programación de Brazo Robótico	41
Figura 3-2: Panorama general de software de Programación de Robotino	41
Figura 4-2: Cámara del Robotino (parte roja)	43
Figura 5-2: Sensores ubicados en Robotino	44
Figura 6-2: Uniones de Brazo Robótico	45
Figura 7-2: Eslabones y juntas del Mitsubishi MELFA RV – 2SDB	45
Figura 8-2: Determinación de grados de libertad	46
Figura 9-2: Vista frontal de la unidad de interfaz.....	47
Figura 10-2: Vista posterior de la unidad de interfaz	47
Figura 11-2: Esquema de funcionalidad de la unidad en base a cada LED	48
Figura 12-2: Asignación de pines del conector centrónico de 24 pines.....	49

Figura 13-2: Modo de conexión del enchufe NA 24V.....	49
Figura 14-2: Mesa de trabajo del sistema	54
Figura 15-2: Estación Brazo Robótico.....	55
Figura 16-2: Tablero de control del sistema.....	55
Figura 17-2: Base de la unidad 534364	56
Figura 18-2: Laboratorio de Automatización.....	56
Figura 19-2: Colocación de canaletas para cableado.....	59
Figura 20-2: Adaptación montacargas del Robotino	58
Figura 21-2: Montaje de PLC en Tablero	59
Figura 22-2: Direccionamiento IP del PLC.....	61
Figura 23-2: Conexiones a borneras de PLC.....	61
Figura 24-2: Colocación de sensores para detectar pallets	62
Figura 25-2: Conexión Syslink.....	63
Figura 26-2: Conexión de entradas y salidas a la caja de Interfaz.....	63
Figura 27-2: Conexión Interfaz-Controlador	64
Figura 28-2: Estado activo de salidas y entradas de caja Interfaz	64
Figura 29-2: Suministro de voltaje a Brazo Robótico	65
Figura 30-2: Código en Software RT Toolbox2.....	65
Figura 31-2: Descripción de Punto posición	66
Figura 32-2: Baterías de Robotino.....	71
Figura 33-2: Conexión a entradas digitales	72
Figura 34-2: Cámara del Robotino (parte roja)	72
Figura 35-2: Secuencia en software Robotino.....	71
Figura 36-2: Subprograma-sensores de distancia y control Montacargas.....	71
Figura 37-2: Subprograma Posicionamiento	72
Figura 38-2: Subprograma seguir línea de color	73
Figura 39-2: Comunicación Ethernet del PLC	74
Figura 40-2: Declaración de variables.....	75
Figura 41-2: Generación del Bloque TCON en programa	75
Figura 42-2: Configuración de Bloque TCON.....	75
Figura 43-2: Parámetros del Bloque TCON.....	76
Figura 44-2: Bloque TDISCON en programa principal.....	76
Figura 45-2: Configuración de ID de bloque TDISCON.....	76
Figura 46-2: Parámetros del Bloque TDISCON.....	80
Figura 47-2: Bloque TURCV en el programa principal.....	81
Figura 48-2: Bloque TUSEND en el programa principal	81
Figura 49-2: Bloque MOVE.....	82

Figura 50-2: Bloque TUSEND en el programa principal	82
Figura 51-2: Parámetros del Bloque TUSEND1	82
Figura 52-2: Condición para funcionamiento bloque TUSEND	83
Figura 53-2: Parámetros del Bloque TUSEND2	83
Figura 54-2: Condición para funcionamiento bloque TUSEND1	84
Figura 55-2: Datos enviados en bloque TUSEND	84
Figura 56-2: Bloque condicionante para comunicación UDP	84
Figura 57-2: Asignación de dirección Robotino.....	81
Figura 58-2: Especificaciones de estructura de datos	86
Figura 59-2: Librería de comunicación UDP en Robotino View	86
Figura 60-2: Asignación de valores a entradas y salidas	87
Figura 61-2: Cuadro de diálogo para configurar comunicación UDP	88
Figura 62-2: Salidas y entradas generadas para comunicación UDP.....	88
Figura 63-2: Código Python.....	89
Figura 64-2: Activación de modo prueba.....	90
Figura 65-2: Conexión online con Robotino.....	87
Figura 66-2: Comprobación de intercambio mensaje Robotino-PLC.....	87
Figura 67-2: Comprobación de intercambio mensaje Robotino PLC.....	91
Figura 68-2: Inicio de programa HMI y conexión online PLC	92
Figura 69-2: Activación envío y recepción de datos desde HMI	93
Figura 70-2: Bodega 1 y Bodega 2	96
Figura 71-2: Robotino Plus	96
Figura 72-2: Brazo Robótico.....	97
Figura 73-2: Home 1 y Home 2.....	97
Figura 74-2: Home 3 y Home 4.....	98
Figura 75-2: Home 5.....	98
Figura 76-2: Tablero del Control.....	99
Figura 77-2: Robotino Plus	99

ÍNDICE DE ANEXOS

Anexo A: Valores de tabla de Chi Cuadrado

Anexo B: Realización de pruebas estadísticas en Excel

Anexo C: Código Brazo Robótico en RT ToolBox2

RESUMEN

El objetivo del presente trabajo de titulación fue analizar e implementar un sistema inalámbrico con el Protocolo de Datagramas de Usuario (UDP) entre el Robotino, Brazo Robótico y la Mesa de Trabajo para realizar procesos de paletización. Para el proceso se utilizó un router que permitió la comunicación inalámbrica, un Controlador Lógico Programable (PLC) para la automatización de la secuencia, sensores para activar señales de detección de objetos, válvulas para mover los pallets y una Caja de Interfaz de Robot para trabajar con el Brazo Robótico. El sistema consta de un tablero de control con un PLC, el cual permite la comunicación con protocolo UDP con el Robotino, de éste modo se realiza el intercambio de mensajes en forma de datagramas. Los mensajes se envían solo si los sensores ubicados sobre distintos puntos de la mesa de trabajo emiten una señal de detección. La secuencia inicia cuando el Robotino se dirige al punto de bodegas, recoge y lleva el pallet al punto de entrada del Brazo Robótico RV2SDB, el cual completa un proceso de paletización y lleva el pallet a su punto de salida (salida RV2SDB), desde donde el Robotino lo lleva al punto de salida del sistema. En cada punto de bodegas, entradas y salida del Brazo Robótico y salida del sistema se tiene sensores que detectan el pallet y emiten una señal de activación para que se realice la secuencia establecida. Adicionalmente una interfaz hombre máquina (HMI) siempre se encuentra monitoreando cada paso del proceso. En conclusión, el protocolo UDP permitió la comunicación de los robots de una manera óptima, de modo que se pudo realizar procesos de manera automática y con comunicación inalámbrica. Se recomienda desarrollar más proyectos con los robots utilizados de tal forma que se puede explorar a profundidad su estructura y aplicaciones.

Palabras clave: <TECNOLOGÍA DEL CONTROL AUTOMÁTICO>, <COMUNICACIÓN INALÁMBRICA>, <PROTOCOLO DE COMUNICACIÓN>, <ROBOT OMNIDIRECCIONAL>, <PROCESOS DE PALETIZACIÓN>, <CONTROLADOR LÓGICO PROGRAMABLE>, <INTERFAZ HOMBRE MÁQUINA>



ABSTRACT

The objective of this titling work was to analyze and implement a Wireless system with the User Datagram Protocol (UDP) between the Robotino, Robotic Arm and the Worktable to carry out palletizing processes. For the process a router was used that allowed wireless communication, a Programmable Logic Controller (PLC) for sequence automation, sensors to activate object detection signals, valves to move pallets and a Robot Interface Box to work with the robotic arm. The system consists of a control panel with s PLC, which allows communication with UDP protocol with the Robotino, thus the exchange of messages in the form of datagram. The messages are sent only if sensors located on different points of the worktable emit a detection signal. The sequence begins when the Robotino goes to the point of warehouses, picks up and take the pallet to the entry point of the RV2SDV Robotic Arm, which completes palletizing process and takes the pallet to its exit point (RV2SDB exit), from where Robotino takes it to the exit point of the system. At each point of warehouses, entrance and exit of the Robotic Arm and exit of the system there are sensors that detect the pallet and emit an activation signal so that the established sequence is carried out. Additionally, a man - machine interface (HMI) is always monitoring every step of the process. In conclusion, the UDP protocol allowed the robots to communicate in an optimal way, so that processes could be carried out automatically and with Wireless communication. It is recommended to develop more projects with the thefts used in such a way that their structure and applications can be explored in depth.

Palabras clave: <AUTOMATIC CONTROL TECHNOLOGY>, <WIRELESS COMUNCATION>, <COMMUNICATION PROTOCOL>, <OMNIDIRECTIONAL ROBOT>, <PALETIATION PROCESSES>, <PROGRAMMABLE LOGIC CONTROLLER>, < MACHINE MAN INTERFACE>



INTRODUCCIÓN

Hoy en día la automatización de proceso y la robótica están estrechamente relacionados, surgiendo así la necesidad de desarrollar sistemas de comunicación que permitan el intercambio de datos entre dispositivos sin la necesidad de estar siempre conectados mediante cables, de éste modo un sistema inalámbrico consiste en una serie de transmisores y un dispositivo receptor para todos los transmisores existentes, el receptor puede ser una placa agregada al panel de alarma, un panel propiamente inalámbrico y para transmisores deben usarse dispositivos que sean compatibles con el receptor (Herrero, 2015). Paralelamente se identifica el concepto de un protocolo de comunicación, el mismo está formado por un conjunto de reglas y formatos de mensajes establecidas a priori para lograr la comunicación entre el emisor y un receptor. Las reglas definen la forma en cómo deben de efectuarse las comunicaciones de las redes, incluyendo la temporización, la secuencia, la revisión y la corrección de errores (Tolosa, 2015)

En el presente proyecto se realizará el análisis y la implementación de un sistema inalámbrico con protocolo de comunicación UDP entre el Robotino, brazo robótico y la mesa de trabajo para realizar procesos de paletización. Para el proceso se va a utilizar un router que permitirá la comunicación inalámbrica, un PLC para la automatización de la secuencia, sensores para activar señales de detección de objetos, válvulas para mover los pallets, una caja de interfaz de robot para trabajar con el brazo robótico y dispositivos eléctricos y electrónicos necesarios para la implementación. El sistema tendrá un tablero de control con un PLC. El sistema realizará la siguiente secuencia: El Robotino se dirigirá al punto de bodegas, recogerá y llevará el pallet al punto de entrada del Brazo Robótico RV2SDB, el cual completará un proceso de paletización y llevará el pallet a su punto de salida RV2SDB, de donde el Robotino lo llevará al punto de salida del sistema. Para monitorear el sistema se tendrá un HMI.

FORMULACIÓN DEL PROBLEMA

¿Cómo se realizará el análisis e implementación de un sistema inalámbrico con protocolo de comunicación UDP entre el Robotino, Brazo Robótico y la Mesa de Trabajo para realizar procesos de paletización?

SISTEMATIZACIÓN DEL PROBLEMA

¿Cuáles son los protocolos de comunicación inalámbrica que permita integrar el Robotino, el Brazo Robótico y la Mesa de Trabajo?

¿Cuáles son los requerimientos que debe cumplir el sistema a implementar?

¿Cuáles son las características que posee el sistema implementado para la comunicación mediante protocolo UDP?

¿El sistema implementado cumplirá con los requerimientos planteados?

JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN

JUSTIFICACIÓN TEÓRICA

La tecnología está cambiando vertiginosamente y principalmente en la robótica, procesos de automatización y sus sistemas de comunicación. Cada día se van desarrollando protocolos de comunicación que permitan trabajar de manera cooperativa a varios sistemas con el fin de realizar una tarea. Los robots realizan trabajos colaborativos con el fin de irse perfeccionando y cada vez asumiendo trabajos más complejos. De hecho, existen infinidad de situaciones en las cuales los robots necesitan compartir tareas y en esos momentos la comunicación entre ellos es fundamental para aumentar la eficiencia y reducir el tiempo de producción.

Un primer trabajo comprende a Diego Rivera y otros autores quienes en 2016 publican su trabajo denominado “Detección y Amortización de Contención UDP Sockets para Multithreading en Máquinas Multinúcleo”. En ésta investigación nos muestra que el uso del paralelismo se ha convertido en la primera opción cuando se busca una mayor eficiencia en la implementación y ejecución de tareas. Sin embargo, hay casos en los que no se obtienen ganancias reales con esta técnica, por ejemplo, el caso de acceso concurrente y acceso de lectura en sockets UDP en Linux. Este documento presenta técnicas de creación de perfiles y pruebas en Linux para mostrar cómo los sockets UDP tienen puntos de contención inherentes a su diseño, recursos que

los hacen incompatibles con los esquemas de paralelismo tradicionales que proporcionan acceso a una estructura compartida, y comprueba cómo finalmente se transfiere el problema La capa de comunicación entre diferentes procesadores con un costo real. Finalmente, se presenta un enfoque simple y novedoso para proporcionar eficiencia en el contexto especificado sin penalizar los canales de comunicación entre procesadores (Blasco et al., 2016).

Complementariamente se tiene el trabajo de Néstor Adrián Balich, quien en Buenos Aires-Argentina en el año 2009 realizó la “Construcción de robots autónomos colaborativos”. En esta tesis se detallan los principales aspectos necesarios para realizar la construcción de un equipo de robots colaborativos. Se realiza una puesta al día de los diferentes tipos de robot, sensores, actuadores y consideraciones a tomar en cuenta en el diseño y programación del software control. Se diseña y desarrolla las placas de control, electromecánica y sistema de comunicación por radio enlace. Se plantea como eje del mismo la factibilidad de construir un conjunto de robots con materiales encontrados en el mercado nacional. Se realiza una comparación de los principales robots en el mercado nacional y posibles alternativas importadas (costos y capacidades). Al final pudo obtener la programación y la definición de algoritmos orientados al manejo de un grupo de robots tanto en lenguaje de bajo nivel (Assembler) y alto nivel (VB.net / Python). Por último, se logró la integración y comunicación entre robots en un equipo de fútbol íntegramente desarrollada en Python que tomo como base el presente trabajo y participo en CAFR2009 organizado por la Universidad de Morón (Balich, 2009).

Un tercer trabajo comprende a Zambrano Pérez Víctor Daniel, quien en Quito-Ecuador en el año 2015 realizó “Implementación de algoritmos de determinación de rutas para el Robotino® de Festo”. La tesis va orientada a analizar el rendimiento de los métodos planificadores bajo diferentes condiciones como el computador utilizado, complejidad del mapa de entorno etc. Y analizar el tiempo de ejecución, ruta obtenida del método planificador. Con estos requerimientos se diseña un programa de interacción simple con el usuario (HMI), en el ingreso de un mapa de entorno, configuración y calibración de parámetros, además que tenga la capacidad de presentar toda la información necesaria acerca del rendimiento de los métodos planificadores de rutas, y el seguimiento de las mismas utilizando un robot Omnidireccional (Z. Pérez & Daniel, 2015).

En medio de todo el proceso de investigación se tomó en consideración el cumplimiento del objetivo 11 del plan Nacional, dice: “Asegurar la soberanía y eficiencia de los sectores estratégicos para la transformación industrial tecnológica. El Ecuador posee una oportunidad histórica para ejercer soberanamente la gestión económica, industrial y científica, de sus sectores estratégicos. Esto permitirá generar riquezas y elevar en forma general el nivel de vida de nuestra población. Para el Gobierno de la Revolución Ciudadana, convertir la gestión de los

sectores estratégicos en la punta de lanza de la transformación tecnológica e industrial del país, constituyendo un elemento central de ruptura con el pasado” (Jomy Duarte, 23:59:54 UTC)

JUSTIFICACIÓN APLICATIVA

Para el trabajo de titulación se va a analizar e implementar un sistema inalámbrico con el Protocolo de Datagramas de Usuario (UDP) entre el Robotino, Brazo Robótico y la Mesa de Trabajo para realizar procesos de paletización. Para el proceso se utilizará un router que permita la comunicación inalámbrica, un Controlador Lógico Programable (PLC) para la automatización de la secuencia, sensores para activar señales de detección de objetos, válvulas para mover los pallets y una Caja de Interfaz de Robot para trabajar con el Brazo Robótico.

El sistema constará de un tablero de control con un PLC, el cual permitirá la comunicación con protocolo UDP con el Robotino, realizando así el intercambio de mensajes en forma de datagramas. La secuencia iniciará una vez que se coloquen pallets sobre los puntos de entrada del sistema y los sensores emitan la señal de detección de los mismos, luego el Robotino se dirigirá al punto de bodegas, recogerá y llevará el pallet al punto de entrada del Brazo Robótico RV2SDB, el cual completará un proceso de paletización y llevará el pallet a su punto de salida (salida RV2SDB), desde donde el Robotino se encargará de llevarlo al punto de salida del sistema.

OBJETIVOS

OBJETIVO GENERAL

Analizar e implementar un sistema inalámbrico con protocolo de comunicación UDP entre el Robotino, Brazo Robótico y la Mesa de Trabajo para realizar procesos de paletización.

OBJETIVOS ESPECIFICOS

Investigar los protocolos de comunicación inalámbrica que permita integrar el Robotino, el Brazo Robótico y la Mesa de Trabajo.

Determinar cuáles son los requerimientos que debe cumplir el sistema a implementar.

Implementar el sistema que cumpla con los requerimientos para la comunicación mediante protocolo UDP.

Evaluar si el sistema implementado cumple con los requerimientos planteados al inicio.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

1.1 Generalidades de la Robótica

Definimos a la robótica como una rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas repetitivas, tareas en las cuales es necesario una alta precisión, tareas peligrosas para el ser humano o tareas irrealizables sin intervención de una máquina. Las ciencias y tecnologías de las que aparece podrían ser: el álgebra, los autómatas programables, las máquinas de estados, la mecánica, la electrónica y la informática (Villca Tapia, 2008).

1.2 Robots móviles

Este tipo de robots brindan la posibilidad de navegar en diferentes terrenos y presentan aplicaciones como: automatización de procesos, vigilancia, reconocimiento de terreno, y también son utilizados como plataformas móviles que incorporan un brazo manipulador (Ortigoza et al., 2007).

Una de las principales características de los robots móviles es su capacidad de desplazamiento autónomo en un entorno desconocido o conocido sólo parcialmente. En todas estas aplicaciones la justificación más importante para la aplicación de la robótica es la dificultad o imposibilidad de intervención humana, bien directa o teleoperada. (Jiménez & Baturone, 1996). Se sabe que uno de los principales problemas de los robots móviles es su navegación en entornos estacionarios, el desafío actual es la navegación autónoma en entornos que contiene obstáculos en movimientos (Rosales et al., 2009).

1.3 Robots manipuladores y Modelo biomecánico del brazo humano

Los robots manipuladores surgieron como la imitación mecánica de los brazos humanos (Rodríguez et al., 2007).

Primeramente, se define un modelo para describir la estructura del brazo y su movilidad asimilándolo a un robot manipulador: el brazo se considera como un mecanismo en cadena abierta de cuatro eslabones que representan el tórax (base), el brazo, el antebrazo y la mano. Estos cuatro eslabones se presentan como cuerpos rígidos conectados por uniones mecánicas holonómicas, los cuales representan las articulaciones de hombro, codo y muñeca (Milanés Hermosilla & Castilla Pérez, 2016).

1.3.1 Robotino® Festo

El robot móvil Festo Robotino se emplea para probar la realización del control de velocidad. El Robotino utiliza la unidad omnidireccional para moverse, consta con un sensor de parachoques alrededor, sensores de distancia infrarrojos, una cámara a color y está controlada por un sistema de PC estándar de la industria (Katona et al., 2016).

El control del Robotino se lo realiza a través de la conexión Wifi, desde el llamado entorno de desarrollo gráfico Robotino View. Robotino View es considerado de programación gráfica interactiva "nativa". En el entorno de desarrollo de bloques funcionales, existen disponibles controladores de procesador de imagen, lógica, vector algebra, etc. y se puede lograr un control complejo con ellos. A través de conexiones TCP / IP y UDP el entorno de desarrollo garantiza el establecimiento de la conexión de datos desde software externo (Katona et al., 2016).



Figura 1-1: Robotino Festo Plus

Fuente: (Robotino® 4.0—Education and Research Robots: Robotino®—Learning Systems—Festo Didactic, 2019)

1.3.1.1 Datos Técnicos del Robotino®

Tabla 1-1: Datos técnicos del Robotino®

Parámetros	Valor
Alimentación de tensión	24 V DC, 4.5 A
Entradas Digitales	8
Salidas Digitales	8
Entradas Analógicas	8(0-10 V)
Salidas por Relé	2

Fuente: (Katona et al., 2016)

1.3.1.2 Diseño y función

Robotino® es un sistema de robot móvil de alta calidad, de gran funcional y con accionamiento omnidireccional. Las tres unidades de accionamiento permiten realizar movimientos en todas direcciones adelante, atrás y lateralmente, y girar sobre un punto. Adicionalmente consta de una webcam y varios tipos de sensores analógicos para medición de distancias, por ejemplo, sensores binarios para protección de colisiones y sensores digitales para detectar la velocidad real (Weber & Bellenberg, 2010).

La marcha del sistema se puede realizar sin la necesidad de conectarlo a un PC. El controlador de Robotino® es un PC embebido con una tarjeta compact flash, en la cual se han instalado varias aplicaciones de demostración y el sistema operativo (Linux). Las aplicaciones de demostración pueden ejecutarse directamente desde el panel de control del Robotino®. (Weber & Bellenberg, 2010). Robotino® puede programarse con el software Robotino® View en un PC a través del LAN inalámbrico. Éste software es capaz de transmitir señales al controlador del motor, visualizar, cambiar y evaluar valores de los sensores. Robotino® puede programarse con Robotino® View incluso durante el funcionamiento real. La webcam permite visualizar y evaluar una imagen de cámara en vivo con ayuda del Robotino® View, con lo que es posible implementar aplicaciones tales como el trazado de rutas y seguimiento de objetos (Weber & Bellenberg, 2010).

Robotino® es autónomo. Está integrado de numerosos sensores, una cámara y un controlador de altas prestaciones aportan al sistema la necesaria “inteligencia”. Puede accederse al controlador directamente a través de la LAN inalámbrica (WLAN). Cuando está correctamente programado, Robotino® realiza de forma autónoma las tareas asignadas. Pueden conectarse actuadores y sensores adicionales a través de un interface de E/S. (FESTO, 2019) Robotino® cuenta con un sistema de robot móvil, creado para el aprendizaje de automatización y actualización de

tecnología de profesionales en estas áreas, además del uso como material didáctico para perfeccionar habilidades en estudiantes incluyendo la educación acerca de medidas de seguridad en la industria y la robótica(Weber & Bellenberg, 2010).

1.3.1.3 Robotino Objetivos de enseñanza

Es indispensable que para la enseñanza los robots deben cumplan ciertas características: primero para que sean completamente didácticos y que sea posible explorar todas las opciones de tecnología familiarizándose fácilmente:

La tecnología del robot está a la vista y disposición de su manipulador o programador, no presenta limitaciones de ejecución de tareas, por lo que es posible que el estudiante conozca mejor la parte física y la pueda explorar, de la misma manera le da la opción de crear y darle innumerables aplicaciones, estudiando a profundidad su funcionamiento, para aplicarlo en la industria (FESTO, 2013).

1.3.1.4 El módulo de la unidad de accionamiento

Robotino® es accionado por 3 unidades de accionamiento omnidireccionales independientes. Se hallan montadas formando un ángulo de 120° entre sí.

Cada unidad de accionamiento posee los siguientes componentes:

- Motor DC
- Reductor con una relación de reducción de 16:1
- Rodillos omnidireccionales
- Correa dentada
- Encoder incremental

Cada uno de los componentes se encuentran fijados a la brida de montaje en la parte posterior. Junto con la brida frontal, la unidad de accionamiento está sujeta al chasis con tornillos. La velocidad real del motor puede compararse con la velocidad deseada por medio del encoder incremental, y puede regularse con un regulador PID a través de la placa de circuito de E/S (Weber & Bellenberg, 2010).

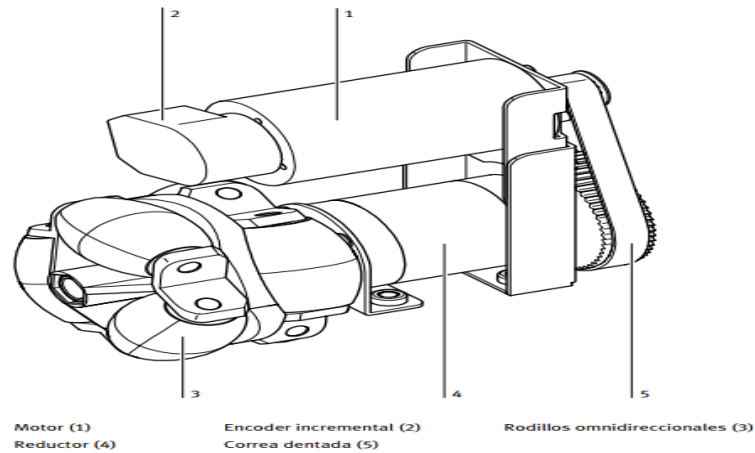


Figura 2-1: Esquema del módulo de accionamiento

Fuente: (Weber & Bellenberg, 2010)

1.3.1.5 La unidad de control

Para configurar la unidad de forma flexible se puede utilizar diversos módulos plug-in.

El controlador del Robotino® consta de 3 componentes:

- Procesador PC 104, compatible con MOPSlcdVE, 300 MHz, y sistema operativo Linux con kernel en tiempo real, SDRAM 128 MB
- Tarjeta Compact flash con API C++ para controlar el Robotino®
- Punto de acceso LAN inalámbrico

La unidad de control está equipada con los siguientes interfaces: Ethernet, 2 ea. USB y VGA, las cuales se utilizan para conectar un teclado, un ratón y una pantalla. Gracias a esto se puede acceder al sistema operativo y a la librería C++ sin un PC (Weber & Bellenberg, 2010).

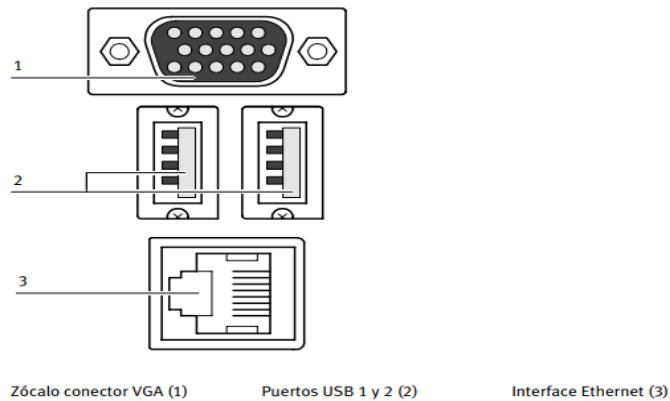


Figura 3-1: Esquema de módulo del controlador de Robotino

Fuente: (Weber & Bellenberg, 2010)

1.3.1.6 Módulo tarjeta de circuito de E/S

Esta tarjeta de E/S permite la comunicación entre la unidad de control y los sensores, la unidad de accionamiento y el interfaz E/S incluidos con el Robot. Los motores de las unidades de accionamiento individuales son controlados por un regulador PID. Por otro lado, las señales del encoder de pasos, de todos los sensores y actuadores instalados permanentemente, que están conectados al interfaz de E/S son transferidas a la unidad de control o a los actuadores adicionales (Weber & Bellenberg, 2010).

1.3.1.7 Fuente de alimentación/cargador de batería

La alimentación eléctrica es suministrada por dos baterías recargables de 12 V con una capacidad de 4 Ah. Ambas baterías recargables están montadas en el chasis. Robotino® se suministra con 2 baterías adicionales y un cargador de baterías. Así, mientras dos baterías se hallan en funcionamiento, las otras dos pueden estar en proceso de recarga (Weber & Bellenberg, 2010).

1.3.1.8 Sensores

En el Robot están integrados sensores para la medición de distancias a objetos y para medir la velocidad del motor. Un sensor anticolidión montado alrededor de una circunferencia en el chasis, indica el contacto con objetos (Weber & Bellenberg, 2010).

1.3.1.9 Sensores de medición de distancia por infrarrojos

Robotino® está integrado con nueve sensores de medición de distancia por infrarrojos, los mismos que están montados en el chasis formando un ángulo de 40° entre sí. Gracias a estos sensores, el robot puede detectar objetos en las zonas circundantes. Cada uno de estos sensores puede ser interrogado individualmente por medio de la placa de circuito de E/S. De este modo pueden evitarse obstáculos, pueden mantenerse distancias y adoptar protecciones frente a un determinado objetivo. Los sensores son capaces de medir distancias con precisión o relativas a objetos, con valores entre 4 y 30 cm (Weber & Bellenberg, 2010).

1.3.1.10 Encoder incremental

La velocidad real de cada motor se mide en RPM por el encoder incremental. Si la velocidad real del motor difiere del punto de consigna, puede ajustarse para que coincida con el valor deseado por medio de un regulador PID, cuyos parámetros están configurados con la ayuda del software Robotino® View (Weber & Bellenberg, 2010).

1.3.1.11 El sensor anticolidión

El sensor anticolidión está compuesto por una banda de detección fijada alrededor de un aro que circunda el chasis. Una cámara de conmutación se halla ubicada dentro de un perfil de plástico. Dos superficies conductoras se hallan dispuestas dentro de la cámara, manteniendo una determinada distancia entre sí. Estas superficies entran en contacto cuando se aplica una mínima presión a la banda. Con ello, una señal perfectamente reconocible es transmitida a la unidad de control (Weber & Bellenberg, 2010).

1.3.1.12 Sensor de proximidad inductivo analógico

Éste sensor se suministra como un componente adicional. Permite detectar objetos metálicos en el suelo y se utiliza para el control filo guiado. Lee señales de diferente intensidad dependiendo de si se halla en el medio o en el borde de una tira metálica, gracias a esto se puede controlar el recorrido de forma perfectamente diferenciada. El sensor de proximidad inductivo debe fijarse al elemento suministrado para este fin y debe conectarse al interface de E/S (Weber & Bellenberg, 2010).

1.3.1.13 Sensores de reflexión directa

El seguimiento de una ruta también se puede implementar con los dos sensores de reflexión directa (de luz difusa) incluidos. Los cables flexibles de fibra óptica se conectan a una unidad óptica que funciona con luz roja visible. Se detecta la luz reflejada. Diferentes superficies y colores producen diferentes grados de reflexión. Sin embargo, no pueden detectarse diferencias graduales en la luz reflejada. Los sensores deben colocarse con los accesorios suministrados para ello y deben ser conectados al interface de E/S (Weber & Bellenberg, 2010).

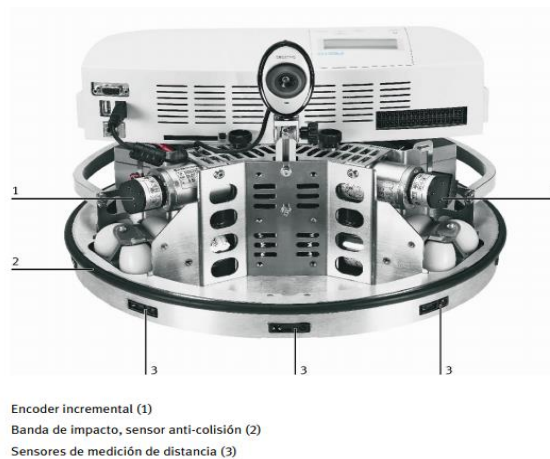


Figura 4-1: Ubicación de los sensores

Fuente: (Weber & Bellenberg, 2010)

1.3.1.14 Posiciones de los sensores y sus denominaciones en Robotino® View

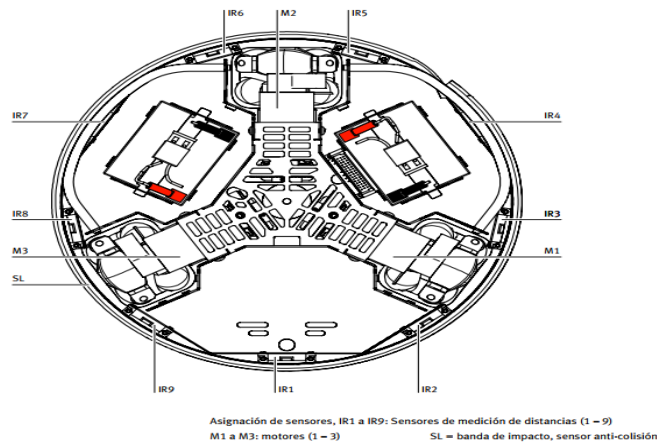


Figura 5-1: Figura: Ubicación de los sensores

Fuente: (Weber & Bellenberg, 2010)

1.3.1.15 Punto de acceso LAN inalámbrico

El punto de acceso LAN inalámbrico es un componente que da paso a la comunicación con el robot a través de una dirección en la red.

- El punto de acceso se caracteriza por su bajo consumo de corriente y puede alimentarse a través del puerto USB.
- El punto de acceso cumple con los siguientes estándares: IEEE 802.11g y 802.11b.
- Sus velocidades de transmisión son de hasta 54 Mb por segundo para 802-11g, y 11 Mb por segundo para 802.11b con un amplio alcance de las transmisiones (hasta 100 m dentro de edificios).
- Se puede establecer una red segura con encriptación WEP y función WPA-PSK.
- Rápida y simple de configurar a través de la utilidad de gestión de la web (Weber & Bellenberg, 2010)

1.3.1.16 La tarjeta compact flash

La unidad de control está equipada con una ranura en la cual se ha insertado una tarjeta de PC. Esta tarjeta de PC contiene el sistema operativo, las librerías de funciones y los programas incluidos. Las actualizaciones pueden instalarse fácilmente con la simple sustitución de la tarjeta PC. La ranura para la tarjeta PC está situada a la derecha de los interfaces de la unidad de control (Weber & Bellenberg, 2010)

1.3.1.17 El interfaz E/S

El interfaz E/S permite conectar sensores y actuadores adicionales. Estos se conectan por medio de un conector incluido.

- 8 entradas analógicas (0 a 10 V) (AIN0 hasta AIN7)
- 8 entradas digitales (DI0 hasta DI7)
- 8 salidas digitales (DO0 hasta DO7)
- 2 relés para actuadores adicionales (REL0 y REL1). Los contactos de los relés pueden utilizarse como contacto normalmente abierto (NA=NO), contacto normalmente cerrado (NC) a (CO= colutorio) conmutados (Weber & Bellenberg, 2010).

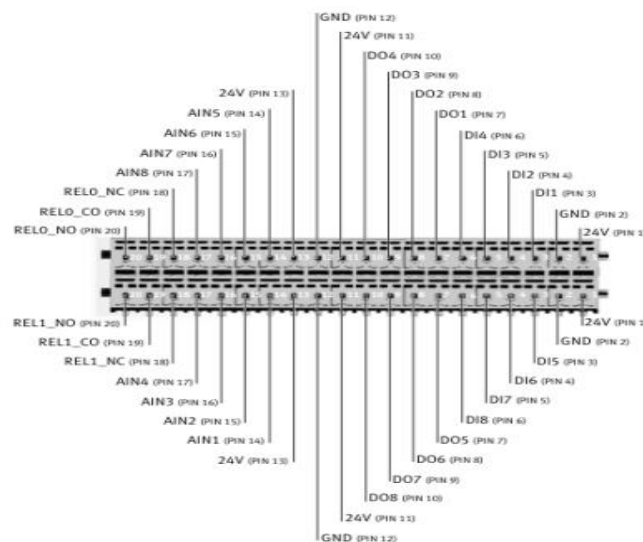


Figura 6-1: Asignación de bornes de la interface E/S

Fuente: (Weber & Bellenberg, 2010)

1.3.1.18 Control e interfaces

- Ordenador personal incrustado conforme a la especificación COM Express
- Premium Edition: Intel i5, 2,4 GHz, Dual-Core, 8 GB RAM, 64 GB SSD
- Basic Edition: Intel Atom, 1,8 GHz, Dual-Core, 4 GB RAM, 32 GB SSD
- WLAN conforme a la especificación 802.11g/802.11b como cliente o Access Point
- Regulación del motor con microcontrolador de 32 bit y conexión libre de motor
- 2 Ethernet, 6 USB 2.0 (HighSpeed), 2 ranuras PCI Express, 1 VGA
- 1 interfaz I/O para la integración de otros componentes eléctricos (Weber & Bellenberg, 2010).

1.3.1.19 Software

- Entorno de programación gráfico para PC externo ejecutable con Windows XP, Vista, Windows 7/8
- API para programación con C/C++, JAVA, .Net, LabVIEW, MATLAB\Simulink, ROS y Microsoft Robotics Developer Studio
(Weber & Bellenberg, 2010).

1.3.1.20 Ampliaciones

Las interfaces estándar como USB y Ethernet permiten conectar otros componentes al control del motor. Adicionalmente, el control pone a disposición entradas/salidas analógicas y digitales para una ampliación posterior, así como salidas de relé para una técnica adicional de los actuadores. Para soportar interfaces no disponibles, como por ejemplo EIA-485 o IEEE 1394, se consta con dos ranuras PCI Express disponibles para tarjetas de interfaces. Tanto en una salida de motor adicional como en una entrada del codificador se pueden conectar, por ejemplo, ejes eléctricos adicionales y pinzas (Weber & Bellenberg, 2010).

1.3.1.21 Robustez y movilidad

El robot posee tres rodillos omnidireccionales de accionamiento independiente que le permiten desplazarse en todas las direcciones. Las construcciones de acero inoxidable del chasis garantizan una gran estabilidad en cada una de las situaciones de desplazamiento (FESTO, 2013).



Figura 7-1: Movilidad del Robot

Fuente: (FESTO, 2013)

1.3.1.22 Construcción flexible y adaptable

Una gran cantidad de posibilidades de montaje, así como la torre de montaje opcional con plataformas de posicionamiento individual, garantizan la utilización del Robot en las más diversas tareas (FESTO, 2013).



Figura 8-1: Flexibilidad y adaptabilidad del robot

Fuente: (FESTO, 2013)

1.3.1.23 Plug and Play

Gracias a Plug and Play, el Robotino® es compatible con diferentes pinzas, manipuladores y sensores. Una vez conectados, el Robotino® reconoce los componentes y se puede iniciar el control (FESTO, 2013).



Figura 9-1: Plug and Play del robot

Fuente: (FESTO, 2013)

1.3.1.24 Entorno de programación abierto del Robotino®

La interfaz de programación (API) del Robotino® permite trabajar con varios lenguajes y sistemas de programación para el desarrollo de un programa de mando. La API soporta los siguientes lenguajes y sistemas:

1. C/C++, JAVA, .Net
 2. LabVIEW und MATLAB\Simulink
 3. Robot Operating System (ROS)
 4. Microsoft Robotics Developer Studio
- (Weber & Bellenberg, 2010)

1.3.2 Brazo robótico RV2SDB

Definimos a un robot serial como una serie de barras rígidas unidas por articulaciones, generalmente de un grado de libertad de tipo rotacional o prismática. Un extremo del robot se fija a la base mientras que el otro extremo se puede mover libremente, de este modo formando una cadena cinemática abierta (Villalobos et al., 2015).



Figura 10-1: Brazo Robótico Mitsubishi RV-2SDB

Fuente: (Mitsubishi Electric, 2010)

1.3.2.1 Descripción general

La alta movilidad del robot está en el diseño constructivo de su brazo, el cual permite un acceso a la zona en torno a la base, haciendo de este modo posible aprovechar al máximo el entorno. Gracias a su ángulo de giro de ± 240 grados no existe en el entorno ninguna pieza a la cual no

pueda acceder el robot. La posibilidad de montaje en techo y la construcción compacta de la muñeca del robot completan esta imagen (Mitsubishi Electric, 2010).

1.3.2.2 Movilidad

El estratégico modo constructivo permite al robot RV-2SDB tener una movilidad extraordinaria incluso en células de trabajo dotadas de un espacio apretadísimo. Debido a esto, el robot puede integrarse también en células de trabajo pequeñas. Para la construcción de nuevos sistemas, esto significa una doble reducción de costes gracias al ahorro de material y al menor espacio necesario (Mitsubishi Electric, 2010).

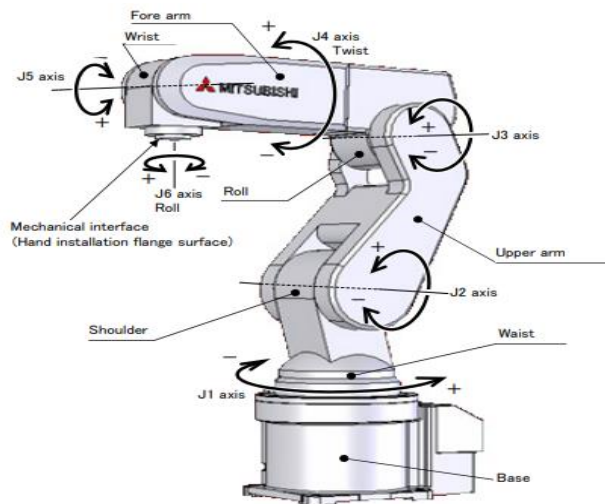


Figura 11-1: Alcance del espacio de trabajo del Brazo Robótico

Fuente: (Mitsubishi Electric, 2010)

1.3.2.3 Eficiencia

Se sabe que las soluciones óptimas requieren un alto grado de eficacia productiva. Precisamente en este aspecto, el robot, con un peso de manipulación de 2 kg (3 kg con la muñeca apuntando hacia abajo) alcanza una velocidad resultante máxima de 4 400 mm/s. Esto permite lograr tiempos de ciclo mínimos y una productividad considerablemente superior. El robot posee una excelente repetibilidad permitiendo así, posicionar componentes y piezas con una precisión de $\pm 0,02$ mm. Así, el RV-2SDB ofrece ambas cualidades: Seguridad en la producción en la instalación y calidad (Mitsubishi Electric, 2010).

1.3.2.4 Integración sencilla en sistemas

El robot RV-2SDB se puede combinar fácilmente con un gran número de componentes de automatización. De este modo, la unidad de control del robot puede comunicarse vía un puerto Ethernet con un terminal de operador de la serie GOT. Así, es posible configurar varios paneles de control mediante un único GOT. La interfaz SSCNET III, disponible de serie en el robot, permite controlar hasta 8 servo ejes MR-J3B mediante un cable de fibra óptica inmune a las interferencias. A ello se añade dos entradas de encoder de la unidad de control que permiten una sincronización sin complicaciones con cintas transportadoras. Por otro lado, la interfaz Ethernet integrada permite conectar una cámara de un sistema de captación de imágenes (Mitsubishi Electric, 2010).

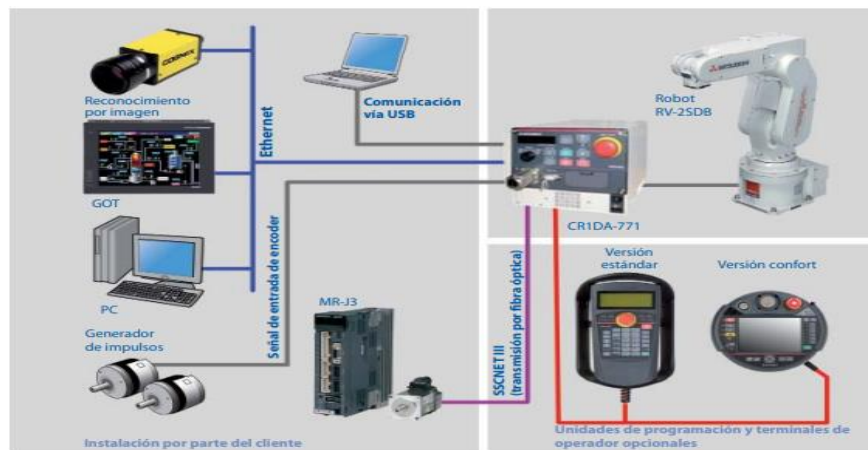


Figura 12-1: Integración del RV-2SDB en diferentes aplicaciones

Fuente: (Mitsubishi Electric, 2010)

1.3.2.5 Herramientas de Programación

Una de las herramientas de programación para desarrollo del sistema y pruebas es el software multilingüe RT Toolbox 2, ésta es la herramienta de programación estándar para el sistema del robot. Con la misma se pueden verificar los programas del robot, simular y optimizar los ciclos de trabajo. Con ayuda del software MELFA-Works es posible representar en 3D el robot y la célula de trabajo en el software CAD SolidWorks, simular las operaciones y verificar la ejecución del programa del robot (Mitsubishi Electric, 2010).

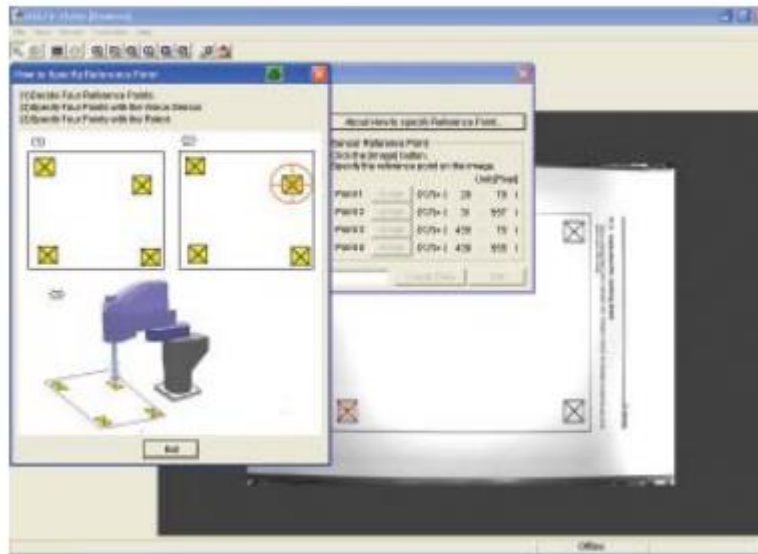


Figura 13-1: Herramientas de programación del RV-2SDB

Fuente: (Mitsubishi Electric, 2010)

1.3.2.6 Rango de movimiento

- Degrees of freedom: 6
- Arm length: 230mm + 270mm
- Maximum reach radius: 504mm
- Operating range:
 - J1: 480° (od -240° do +240°)
 - J2: 240° (od -120° do +120°)
 - J3: 160° (od 0° do +160°)
 - 4: 400° (od -200° do +200°)
 - J5: 240° (od -120° do +120°)
 - J6: 720° (od -360° do +360°)
- Maximum angular velocity:
 - J1: 225°/s
 - J2: 150°/s
 - J3: 275°/s

J4: 412°/s

J5: 450°/s

J6: 720°/s

(Papcun, 2012)

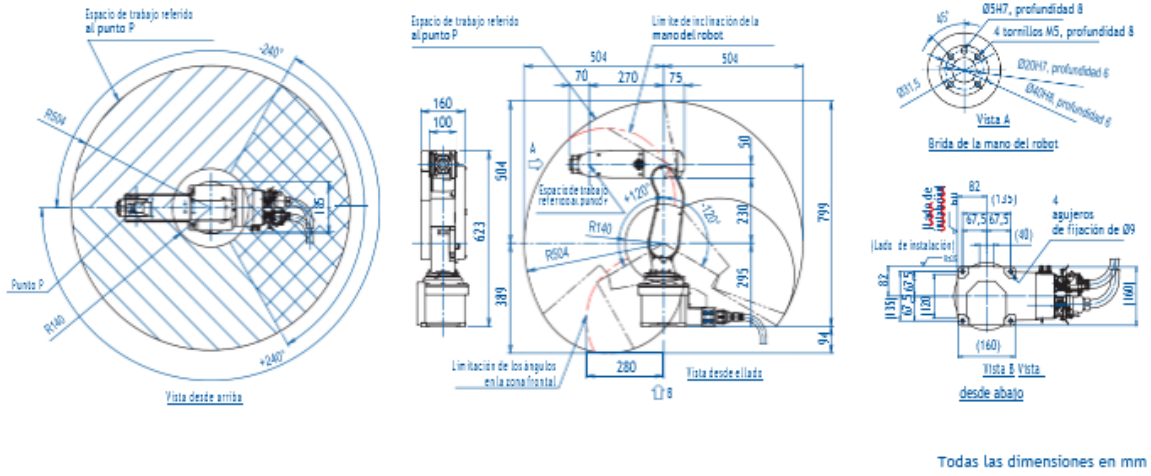


Gráfico 1-1: Rango de trabajo de Robot

Fuente: (Mitsubishi Electric, 2010)

1.3.2.7 Datos técnicos

Control de robot		CR10A-771 (RV-2SDB)	Observación
Lenguaje de programación		MELFA-BASIC V	
Determinación de la posición		Aprendizaje, introducción manual de datos (IMD)	
Entradas/salidas generales	Entradas/salidas generales	0 entradas/0 salidas (máx. 256/256 como equipamiento especial)	
	Entradas/salidas especiales	Definido por usuario	
Entradas/salidas para pinza-mano		4 entradas/0 salidas	Como opción pueden incorporarse 4 salidas adicionales
PARADA DE EMERGENCIA externa		1	con diseño basado en dos circuitos
Entradas/salidas externas	Contacto de cierre de puerta	1	con diseño basado en dos circuitos
	Pulsador de permiso	1	con diseño basado en dos circuitos
	Sincronización para eje adicional	1	con diseño basado en dos circuitos
	Estado operativo	1	con diseño basado en dos circuitos
	Salida de errores	1	con diseño basado en dos circuitos
	RS-232	1	Ampliación para PC, sensor de captación de imagen, etc.
	Ethernet	1	10BASE-T/100BASE-TX
Interfaces	USB	1	sólo función de dispositivo, conector Mini-B
	Eje adicional	1	para un total de hasta 8 servomotores
	Seguimiento de la cinta transportadora	1	para dos encoders
	Conexión en la mano del robot	1	para dos pinzas neumáticas
	Slot de ampliación	1	para interfaces adicionales
	Tensión de entrada		200–230 V AC \pm 10 %, monofásica (180–253 V)
Alimentación eléctrica	Potencia absorbida	kVA	0,5
	Frecuencia	Hz	50/60
	Temperatura ambiente	°C	0–40
Nivel de rendimiento (PL)			d
Clase de sala blanca ISO			7
Dimensiones (AnxAlxPr)		mm	240x200x290
Peso		kg	aprox. 9
Carcasa/Clase de protección			Carcasa cerrada para colocación sobre suelo/IP20

Robot		RV-2SDB
Clase de protección		IP30
Montaje		Puede montarse en suelos, paredes o techos ⁽¹⁾
Tipo constructivo		Brazo articulado vertical
Número de ejes		6
Longitud de brazo		mm
Límites de trabajo máx.		mm
Velocidad máxima resultante		mm/s
Tiempo de ciclo		0,6 s ⁽²⁾
Esfuerzo de elevación	Valor nominal	kg
	Máximo	kg
Repetibilidad		mm
Peso		kg

(1) Montaje en pared con limitación en el eje de la base

(2) Pick and Place (base de ciclo en mm)



Gráfico 2-1: Especificaciones y detalles del RV-2SDB

Fuente: (Mitsubishi Electric, 2010)

1.3.2.8 El cómodo terminal de operador para aprendizaje móvil

El R56TB es un potente panel de operador para la ejecución de todo tipo de tareas directamente en el robot, tales como: el control del robot, la visualización de las cargas, pasando por la visualización de las entradas y las salidas, y la creación de programas completos y la parametrización. Las numerosas funciones que integra aseguran un aprovechamiento óptimo del sistema robotizado, acortando de este modo los tiempos de puesta a punto.

El puerto USB integrado permite intercambiar datos cómodamente, también permite restaurar o crear copias de seguridad completas de los programas y parámetros del controlador mediante una memoria USB (Mitsubishi Electric, 2016).



Figura 16-1: Panel de operador del RV-2SDB

Fuente:(Mitsubishi Electric, 2016)

1.3.2.9 Operaciones Básicas

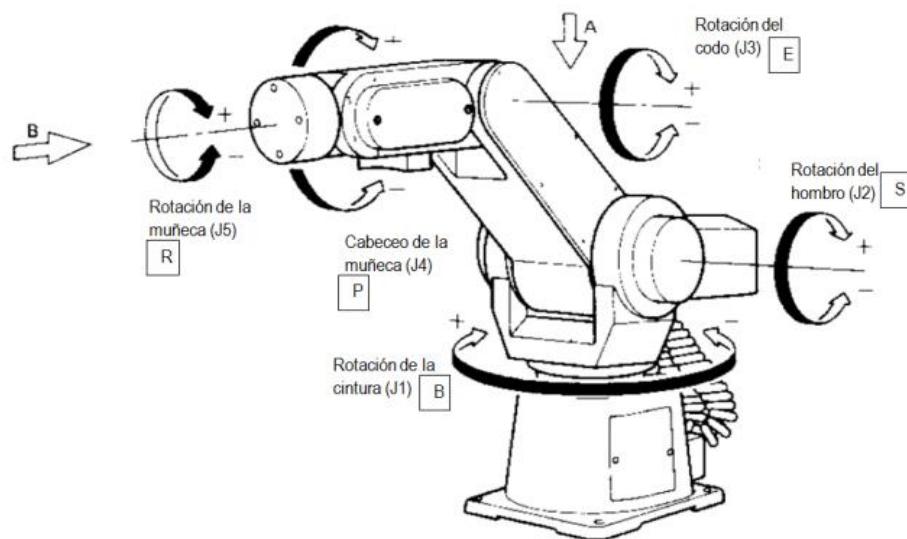


Figura 17-1: Operaciones Del Sistema Articulado

Fuente: (Gutiérrez Salazar, 2010)

Nota 1: El sentido positivo de operación de los ejes J1 y J5 es el de las manecillas del reloj visto desde las flechas A y B, respectivamente.

Nota 2: El sentido positivo de operación de los ejes J2, J3 y J4 es en la dirección hacia arriba del brazo y la muñeca (Gutiérrez Salazar, 2010).

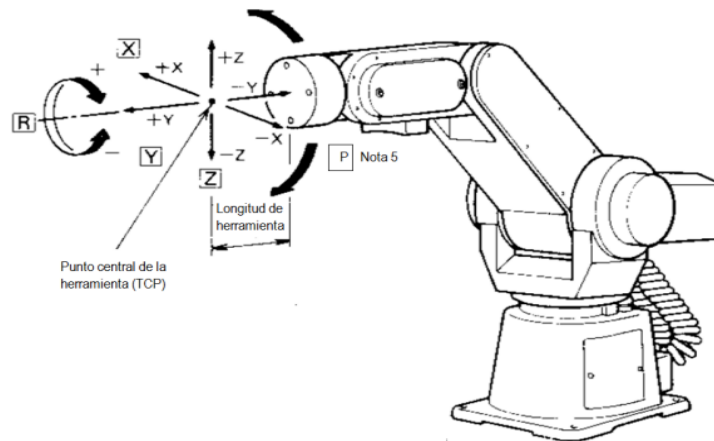


Figura 18-1: Operaciones en el Sistema Cartesiano de Coordenadas

Fuente: (Gutiérrez Salazar, 2010)

Nota 3: El TCP (punto central de la herramienta) se mueve en línea recta en el sistema cartesiano.

Nota 4: La longitud de la herramienta está fijada por un parámetro. (Vea el comando TL “Tool Length”.

Nota 5: P indica la operación de cambio de actitud del robot sin mover el TCP

(Gutiérrez Salazar, 2010).

1.4 Controlador lógico programable “PLC”

En la actualidad el elemento fundamental en la industria es el PLC, el cual tiene la función principal de controlar los sistemas y procesos presentes en la industria. El uso de laboratorios virtuales en fases preliminares al completo montaje de las líneas de producción resulta muy útil para ahorrar dinero y tiempo. Se presenta la ventaja de constar con los controladores y la lógica secuencial ya funcionando en su gran mayoría, antes de contar con el sistema completamente listo. (Cáceres & Amaya, 2016)

Los sistemas automáticos modernos usan en su gran mayoría PLC’s, los cuales se encargan de coordinarlos. Debido a lo expuesto, vemos que la industria actual requiere ingenieros con grandes capacidades y conocimientos en automatización, es aquí donde surgen uno de los mayores retos en la industria y la educación, enseñar la programación de los controladores a los estudiantes (Cáceres & Amaya, 2016).



Figura 19-1: Conexión de un producto o proceso industrial a un controlador lógico

Fuente: (E. M. Pérez et al., 2009)



Figura 20-1: Conexión de un controlador lógico a un producto o un proceso industrial

Fuente: (E. M. Pérez et al., 2009)

1.4.1 PLC SIEMENS S7 -1200

Éste tipo de PLC proporciona la potencia y flexibilidad necesaria para controlar una amplia variedad de dispositivos de automatización. Posee un diseño compacto, configuración flexible y potente conjunto de flexible, además de un conjunto de instrucciones que se combinan para lograr que el S7-1200 sea la solución perfecta para el control de una amplia variedad de aplicaciones (Oñate Lazo & Pinta Yaucan, 2013).

En la industria, numerosos sistemas de manufactura y procesos continuos requieren máquinas y/o plantas productivas típicamente controladas mediante algoritmos generados en controladores lógicos programables (PLC) que puedan ser modificados ante nuevos requerimientos de las máquinas o plantas de manera fácil, rápida y segura mediante el uso de PLC, existen aplicaciones desde el control de motores hasta sistemas de control inalámbrico (José Carlos et al., 2014).



Figura 21-1: Autómata Programable S/-1200 de Siemens

Fuente: (Portero et al., 2018)

1.4.1.1 Entradas y salidas digitales integradas

Entradas Digitales

- Tipo: Sumidero/Fuente
- Tensión nominal: 24 VDC a 4 m

Salidas Digitales

- Tipo: Relé
- Rango de voltaje: 5 a 30 VDC o 5 a 250 VAC
- Corriente (máx.): 2.0 A
- Tipo: Fuente
- Rango de voltaje: 20.4 a 28.8 VDC
- Corriente (máx.): 0.5 A

Entradas analógicas integradas

- 2 entradas analógicas
 - Tipo: Tensión (unipolares)
 - Rango: 0 - 10 V
 - Resolución: 10 bits
- (SIEMENS, 2015)

<i>Integrated I/O</i>	CPU 1211C	CPU 1212C	CPU 1214C
Integrated Digital I/O	6 / 4 salidas	8 entradas / 6 salidas	14 entradas / 10 salidas
Integrated Analog I/O	2 entradas	2 entradas	2 entradas
Max. Local I/O – Digital	14	82	284
Max. Local I/O – Analog	3	19	67
Tamaño de imagen de proceso	1024 Bytes for entradas / 1024 Bytes for salidas		

Gráfico 3-1: Cuadro comparativo de acuerdo al número de E/S

Fuente: (SIEMENS, 2015)

A través de un PLC y tarjetas de adquisición de datos se podrán manipular las variables de un sistema, permitiendo de este modo, la implementación de Interfaces para comunicarse con software Matlab y LabView. Para la programación del PLC se debe tomar en cuenta las condiciones iniciales de operación del proceso y será de acuerdo a los tipos de entradas y salidas (Martínez et al., 2018).

1.4.1.2 Memoria de usuario integrada memoria de usuario

EL S7-1200 CPU dispone de las siguientes áreas de memoria:

- Memoria de carga. - un área de almacenamiento no volátil para el programa de usuario, los datos y la configuración.
- Memoria de trabajo Memoria de trabajo. - un área de un área de almacenamiento volátil para algunos elementos del proyecto de usuario utilizado por la CPU mientras se ejecuta utilizado por la CPU durante la ejecución del programa de usuario.
- Memoria retentiva. - un área de almacenamiento no volátil que se utiliza almacenamiento no volátil que se utiliza para "retener" una cantidad limitada de valores de la memoria de trabajo durante una pérdida de potencia.

(SIEMENS, 2015)

1.4.1.3 Rendimiento

La CPU S7-1200 combina un microprocesador, una fuente de alimentación integrada, entradas, salidas y en una carcasa compacta para crear un controlador muy potente.

La CPU controla cambios de entradas/ salidas y de acuerdo a la lógica del programa de usuario, el mismo puede incluir lógica booleana conteo temporizador lógica booleana, conteo,

temporizador, operaciones matemáticas, y la comunicación con otros dispositivos (SIEMENS, 2015).

1.4.1.4 Corriente Necesaria

La CPU dispone de una fuente de alimentación interna que suministra energía eléctrica a la CPU, los módulos de señales, la Signal Board y los módulos de comunicación, así como otros equipos consumidores de 24 V DC. La CPU provee una alimentación de sensores de 24V DC que puede suministrar 24 V DC a las entradas y bobinas de relé de los módulos de señales, así como a otros equipos consumidores. Si los requisitos de corriente de 24 V DC exceden la capacidad de la alimentación de sensores, es preciso añadir una fuente de alimentación externa de 24 V DC al sistema. Si se requiere una fuente de alimentación externa de 24 V DC, vigile que no se conecte en paralelo con la alimentación de sensores de la CPU. Para aumentar la protección contra interferencias, se recomienda conectar los cables neutros (M) de las distintas fuentes de alimentación (SIEMENS, 2015).

1.4.1.5 Comunicaciones a todos los niveles

La familia S7-1200 ofrece una variedad de opciones de comunicación para satisfacer todas sus necesidades de satisfacer todas sus necesidades de red.

- PROFINET
- PROFIBUS
- Punto a Punto (PtP) Communication
- Universal Serial Interface (USS)
- Modbus RTU
- (Modbus TCP/IP en proceso)
- Comunicación Telecontrol

(SIEMENS, 2015)

1.4.1.6 Interfaz PROFINET Integrada

La CPU S7-1200 dispone de una interfaz PROFINET integrada que soporte TCP / IP basados soporte TCP / IP basados en estándares de comunicación.

Los siguientes protocolos de aplicación son compatibles con el S7- 1200:

- PROFINET – Servicios básicos y controlador I/O Servicios básicos y controlador I/O
- Comunicación abierta entre usuarios – TCP, ISO on TCP y UDP

(SIEMENS, 2015)

1.4.1.7 Comunicación Punto-a-Punto

La comunicación Punto-a-Punto permite una gran variedad de posibilidades:

- La capacidad de enviar información directamente a un dispositivo externo, como una impresora como una impresora
- La capacidad de recibir información de otros dispositivos, tales como lectores de códigos de barras lectores RFID y códigos de barras, lectores RFID, y sistemas de visión

La capacidad de intercambio de información envío y recepción de datos información, envío y recepción de datos, con otros dispositivos tales como dispositivos GPS, módems de radio, y muchos otros tipos de dispositivos (SIEMENS, 2015).

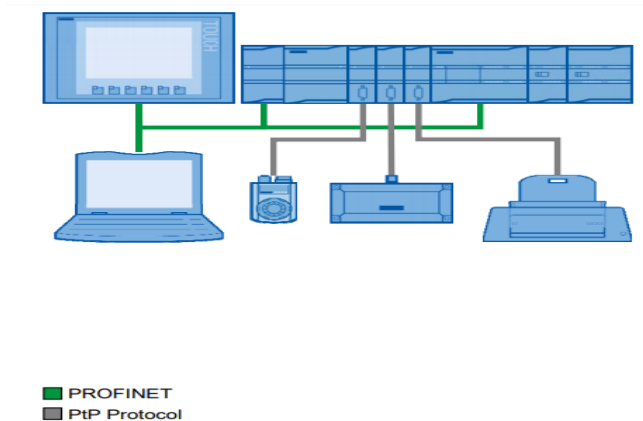


Figura 23-1: Esquema de Comunicación Punto a Punto

Fuente: (SIEMENS, 2015)

1.5 Protocolos de Comunicación

Se define un protocolo como el lenguaje con el que se comunican los dispositivos en la red. Por ejemplo, dos computadores conectados en la misma red, pero con protocolos diferentes no

podrían comunicarse jamás, para ello, es necesario que ambas "hablen" el mismo idioma, por tal sentido, el protocolo TCP/IP fue creado para las comunicaciones en Internet, para que cualquier computador se conecte a Internet, es necesario que tenga instalado este protocolo de comunicación (*Interfaces y Protocolos de Comunicación*, 2012).

Los protocolos de comunicación nos ayudan a establecer la comunicación entre los diferentes dispositivos ya sean para el fin de automatizar, de interactuar, de controlar, etc. Y dependiendo de la necesidad de respuesta o de fabricación se emplean unos, al conocer las diferentes ventajas de estos, los fabricantes pueden seleccionar con cual protocolo desarrollan algún dispositivo, teniendo en cuenta que la rapidez, la compatibilidad con la mayor cantidad de dispositivos que sean posibles, así como los softwares que existe en el mercado. Lo anterior puede dar un punto de decisión al consumidor para poder comprar un producto, ya que en la implementación existen protocolos que son más difíciles de entablar comunicación con otras interfaces, estaciones de nivel, flujo, presión y temperatura en el Laboratorio de Redes Industriales y Control de Procesos de la ESPE (Staff, 2019).

Algunas de las funciones que debe cumplir un protocolo son:

- Detección de la conexión física sobre la que se realiza la conexión (cableada o sin cables)
- Pasos necesarios para comenzar a comunicarse (Handshaking)
- Negociación de las características de la conexión.
- Cómo se inicia y cómo termina un mensaje.
- Formato de los mensajes.
- Qué hacer con los mensajes erróneos o corruptos (corrección de errores)
- Cómo detectar la pérdida inesperada de la conexión, y qué hacer en ese caso.
- Terminación de la sesión de conexión.
- Estrategias para asegurar la seguridad (autenticación, cifrado)

(*Interfaces y Protocolos de Comunicación*, 2012).

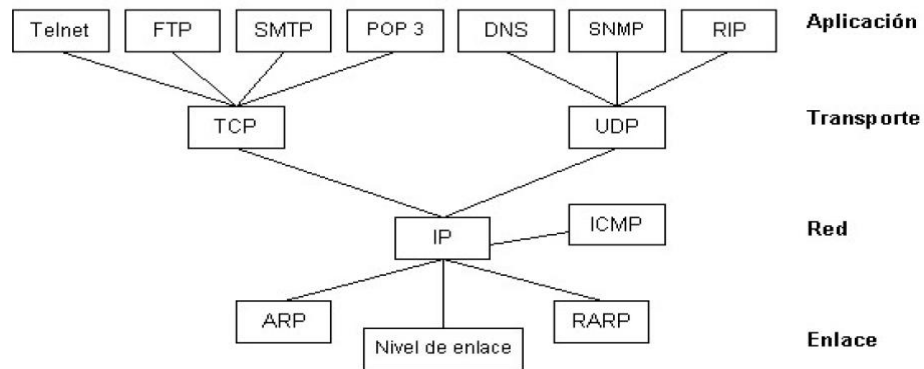


Figura 24-1: Conjunto de protocolos en la arquitectura de red TCP/IP

Fuente: (Herías & Baeza, 2009a)

1.5.1 Diseño de protocolo

1.5.1.1 Pasos

- Definición Del Servicio
- Primitivas De Soporte Y Suposiciones Del Entorno
- Reglas Del Protocolo
- Vocabulario De Mensajes
- Codificación

(marelysmolina21, 2017)

1.5.1.2 Arquitectura TCP/IP

La arquitectura TCP/IP se constituye por una serie de capas o niveles en los que se encuentran los protocolos que implementan las funciones necesarias para la comunicación entre dos dispositivos en red. OSI es el modelo empleado en el estudio de las redes de datos mientras que la arquitectura TCP/IP es un modelo real empleado en las redes actuales (marelysmolina21, 2017).

En la siguiente figura se aprecian los niveles o capas de los modelos OSI y TCP/IP.



Figura 25-1: Representación de capas o niveles OSI y TCP/IP

Fuente: (marelysmolina21, 2017)

1.5.2 *Protocolo UDP*

Éste protocolo recoge el mensaje y envía el paquete por la red sin necesidad de establecer una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Para garantizar la llegada, el protocolo exige a la máquina de destino del paquete que envíe un mensaje. Si enviado un mensaje no llega luego un tiempo establecido, la máquina de destino envía el mensaje de nuevo. Lo que esto puede originar es la duplicación y/o desordenación de los datagramas a su destino. Por ello se clasifica como un protocolo de tipo best-effort (máximo esfuerzo), porque hace lo que puede para transmitir los datagramas hacia la aplicación, pero no puede garantizar que la aplicación los reciba (Still Puebla, 2018).

Por tanto, UDP es un protocolo sencillo no orientado a la conexión al no establecer una conexión segura y fiable. Muchas aplicaciones cliente/servidor que funcionan en base a una solicitud y una respuesta recurren al protocolo UDP cuando no es necesario establecer y luego liberar una conexión. La mayoría de las aplicaciones claves de Internet utilizan el protocolo UDP como el Protocolo de Administración de Red, el Protocolo de Información de Enrutamiento o el Sistema de Nombres de Dominio. En este último, por ejemplo, es preferible utilizar UDP a TCP debido a que las consultas deben ser rápidas y solo necesitan una solicitud (Still Puebla, 2018).

1.5.2.1 *Características principales*

Las características principales del protocolo UDP son:

- Sin conexión. No emplea ninguna sincronización entre origen y destino.
- Trabaja con paquetes o datagramas enteros, no con bytes individuales como TCP. Una aplicación que emplea el protocolo UDP intercambia información en forma de bloques de bytes, de forma que, por cada bloque de bytes enviado de la capa de aplicación a la capa de transporte, se envía un paquete UDP.
- No emplea control del flujo ni ordena los paquetes.
- Su gran ventaja es que provoca poca carga adicional en la red, ya que es sencillo y emplea cabeceras muy simples.
- Un paquete UDP puede ser fragmentado por el protocolo IP para ser enviado fragmentado en varios paquetes IP si resulta necesario.
- Puesto que no hay conexión, un paquete UDP admite utilizar como dirección IP de destino la dirección de broadcast o de multicast de IP. Esto permite enviar un mismo paquete a varios destinos de forma simultánea.

(Herías & Baeza, 2009b)

El formato de un paquete UDP es el representado en la figura a continuación:



Figura 27-1: Formato de la cabecera del protocolo UDP

Fuente: (Herías & Baeza, 2009b)

1.5.2.2 Funciones de los campos de la cabecera de UDP

- **Puerto fuente y puerto destino.** Valores de 16 bits correspondientes a los puertos de nivel de transporte.
- **Longitud.** Número total de bytes en el paquete UDP original (incluye la cabecera y los datos), antes de ser fragmentado en paquetes IP.
- **SVT.** Suma de verificación, aplicada a la cabecera y datos UDP, además de a algún campo de la cabecera IP.

(Herías & Baeza, 2009b)

1.5.2.3 Aplicaciones de UDP

- Transmisión de datos en LANs fiable, como el protocolo TFTP (Trivial File Transfer Protocol), que es una variante del protocolo FTP que emplea como protocolo de transporte UDP.
- Operaciones de sondeo. Transmisión de paquetes de datos pequeños o esporádicos para informar del estado de los equipos de la red, o para intercambiar información de encaminamiento, como es el caso de los protocolos DNS (Domain Name System), RIP (Routing Information protocol) o SNMP (Simple Network Management Protocol).
- Transmisiones multicast de video o audio. UDP es usado por aplicaciones de VoIP (*Voice over IP*), difusión de video y multiconferencia. En la transmisión de señales digitales suele ser más importante una respuesta rápida de los protocolos que un envío completamente fiable. No importa que se pierdan algunos datos: lo importante es que se mantenga un flujo constante de información.
- Otra aplicación es el envío de transacciones rápidas a BB.DD a través de redes LAN fiables. En este caso también premia la rapidez de respuesta, y dado que la red ofrece una alta calidad, no es necesario el complejo control de flujo de TCP.

(Herías & Baeza, 2009b)

1.5.2.4 La importancia de UDP

En el nivel de transporte de la pila de protocolos TCP/IP, TCP proporciona un servicio confiable orientado a la conexión, es decir, los paquetes llegan sin error y en el orden en el que se envían. Por otro lado, UDP es un protocolo que proporciona un servicio orientado a datagramas, no asegurando que los paquetes lleguen a su destino, y si llegaran, no garantizando su orden. UDP es un protocolo más simple que TCP, y mucho menos fiable, aunque más rápido. Es útil para aplicaciones que sean simples, que no necesiten de una transmisión fiable de datos, o incluso que necesiten que sus datos sean transmitidos lo más rápidamente posible (Dávila et al., 2001).

Los protocolos de audio y video están adquiriendo una gran importancia en la actualidad, debido a la gran ayuda que pueden prestar en el ámbito académico y empresarial. De este modo, mediante videoconferencia y audioconferencia, los miembros de diversas sucursales de una

empresa pueden comunicarse, y grupos de investigación de varios países pueden compartir sus opiniones y resultados de trabajos utilizando para ello la red pública Internet (Dávila et al., 2001).

Además, utilizando el mecanismo de streaming, se pueden escuchar o visualizar contenidos previamente grabados o que están siendo filmados en tiempo real. Un ejemplo de esto son las Webcams. Las aplicaciones de audioconferencia y videoconferencia utilizan UDP porque no necesitan de los mecanismos de comunicación fiable que TCP ofrece (Dávila et al., 2001).

1.5.2.5 Bus de comunicación

Definimos bus de comunicación como un medio de conexión entre distintos equipos que permite a todos los dispositivos intercambiar datos en cualquier momento y lugar. Parece que este sistema puede ser más costoso, pero no es así ya que la instalación es un proceso muy rápido y su puesta en marcha es muy sencilla (Business, 2018).

1.6 Comunicación Wi-Fi

Entendemos el medio de transmisión Wi-Fi como un medio compartido en el cual un dispositivo que se encuentre dentro del rango de la señal puede escuchar o interferir en el mensaje de la comunicación. Por otra parte, en lo referente a la tecnología Wi-Fi el coste de los elementos hardware requeridos para poder captar o interferir en las comunicaciones es bajo, y el acceso es libre (Castro, 2005).

1.7 Procesos de Paletización

El proceso de manipulación automatizada de objetos y piezas ha llegado hasta el uso de palets, que son contenedores compartimentados o elementos de fijación, que permiten ordenar un determinado número de piezas iguales, para su transporte y manipulación del conjunto. Esta manera hace que no se pierda el orden, facilitando las tareas de manipulación (Universidad Don Bosco, 2016).

Entonces definimos al proceso de paletizado como la acción y efecto de disponer mercancía sobre una estructura para su almacenaje y transporte; el proceso de paletización ha sido considerado como una de las mejores prácticas dentro de los procesos logísticos, puesto que se logra un mejor desempeño en las actividades de cargue, transporte, descargue y almacenamiento de producción, mejorando así, la utilización de los recursos y la eficiencia de los procesos que se llevan a cabo entre los componentes de una cadena de producción (Granada et al., 2012)

CAPÍTULO II

2. MARCO METODOLÓGICO

2.1 Introducción

En el presente capítulo se especifican cada una de las etapas de análisis e implementación de un sistema de comunicación inalámbrico con protocolo de comunicación UDP. Durante el desarrollo también se va detalla el trabajo del Brazo Robótico RV2SDB y el Robotino en el proceso de paletización. Dichas etapas se detallan en la figura a continuación:

ANÁLISIS

Como primer paso se realizó un diagnóstico del estado actual del área de trabajo y de los materiales con los que se cuenta para el proceso, además se estableció los requerimientos que se deben cumplir para implementar el sistema.

2.2 Diagnóstico del estado actual de área de trabajo y materiales



Figura 1-2: Estado actual del área de trabajo

Realizado por: Paucar, Alex; 2019

En la Figura 1-2 se puede observar las condiciones en las que se encuentra el laboratorio al inicio del trabajo. Se conoció el lugar en donde se iba a realizar la tesis y se evaluó el estado, de modo que me permitiera tomar decisiones a posterior.

En el proceso de análisis y evaluación se abarcan las siguientes áreas:

Infraestructura: El laboratorio cuenta con un amplio espacio que permite ubicar la Mesa de Trabajo, la estación del Brazo Robótico y demás piezas que sean necesarias en el trabajo. De éste modo el laboratorio es apto para realizar el proyecto.

Tecnología, comunicación y servicios: El laboratorio cuenta con puertos de conexión a internet, consta de suficientes tomas para el suministro de voltaje de 100 V, con servicio de tuberías para aire y con un compresor que provee de aire para los procesos aerodinámicos. Para mejorar éste aspecto se requerirá tomas para suministros de voltaje de 220 V para el Brazo Robótico y un router para tener comunicación inalámbrica.

Materiales: Se consta de los 2 robots necesarios para el proceso, el Robotino y el RV-2SDB; y válvulas. Para el proyecto se determina que entre los principales materiales faltantes están los sensores inductivos, el PLC y un módulo de comunicación para robots.

Organización: El laboratorio está apto para trabajar en horarios de 14h00 a 21h00, lo que nos permite trabajar de manera continua y sin interrupciones.

Con los resultados obtenidos en ésta fase se determina que con el estado actual del área de trabajo y de los materiales si se puede realizar la implementación del proyecto. A continuación, se procede a definir los requerimientos del sistema.

2.3 Definición de Requerimientos del sistema

Para la implementación se definieron los requisitos que el sistema requería para ser puesto en funcionamiento:

2.3.1 Rapidez en intercambio de datos

Se presenta una tabla de comparativa entre 2 protocolos:

Tabla 1-2: Comparación de características de Protocolos

UDP vs TCP		
	TCP	UDP
Abreviatura	Transmission Control Protocol	User Datagram Protocol o Universal Datagram Protocol
Conexión	ICP es un protocolo orientado a conexiones	UDP es un protocolo sin conexiones.
Función	Se usa para enviar mensajes por Internet de una computadora a otra. Eso se hace por medio de conexiones virtuales.	UDP se usa para transporte de mensajes y/o transferencias. Pero no está basada en conexiones, que significa que un programa puede enviar una carga de paquetes de data y hasta ahí llega esa relación.
Uso	TCP es útil para aplicaciones que requieren confiabilidad alta y donde el tiempo de transmisión es menos crítico.	UDP es útil para aplicaciones que necesitan transmisión rápida y efectiva.
Velocidad de transferencia	TCP es más lento que UDP	UDO es más rápido porque no hace verificación de errores por paquete de data
Peso	TCP es pesado. Requiere tres paquetes para establecer una conexión antes de transmitir.	UDP es liviano, No hay ordenamiento de mensajes, conexiones de verificación, etc. Es un nivel pequeño de transporte sobre el protocolo de Internet (IP, en inglés).
Control de flujo de data	TCP hace Control de Flujo. Requiere tres paquetes para establecer una conexión antes de transmitir.	UDP no tiene opción para Control de Flujo.

Fuente: (Diffen, 2017)

En la Tabla 1-2 se puede ver de manera detallada cada uno de las características del Protocolo UDP y TCP, de modo que se puede comparar y ver cuál sería el óptimo dentro de mi sistema.

2.3.2 Comunicación inalámbrica:

Mediante un Router se logra éste tipo de comunicación. Un router Wireless o Wifi nos provee acceso a la red local y a internet de forma inalámbrica a cualquier dispositivo, ya sea notebook, tablet, impresoras, discos de almacenamiento o smartphones que esté dentro del alcance de la señal (*¿Qué es un Router?*, 2018).

2.3.3 Controlador lógico programable con comunicación UDP y características específicas:

- Tipo de PLC: S7-1200
- E/S: 14 Entradas/10 Salidas
- Tensión Nominal: 24 VDC
- Protocolo de comunicación: PROFINET UDP

2.3.4 Robot con protocolo de comunicación UDP

Con el dispositivo de intercambio de datos UDP, pueden intercambiarse datos entre Robotino View y aplicaciones externas a través de UDP.

2.3.5 Robots con características específicas

2.3.5.1 Programables

El Robot RV-2SDB2SDB se puede programar con el software á el RT ToolBox2.

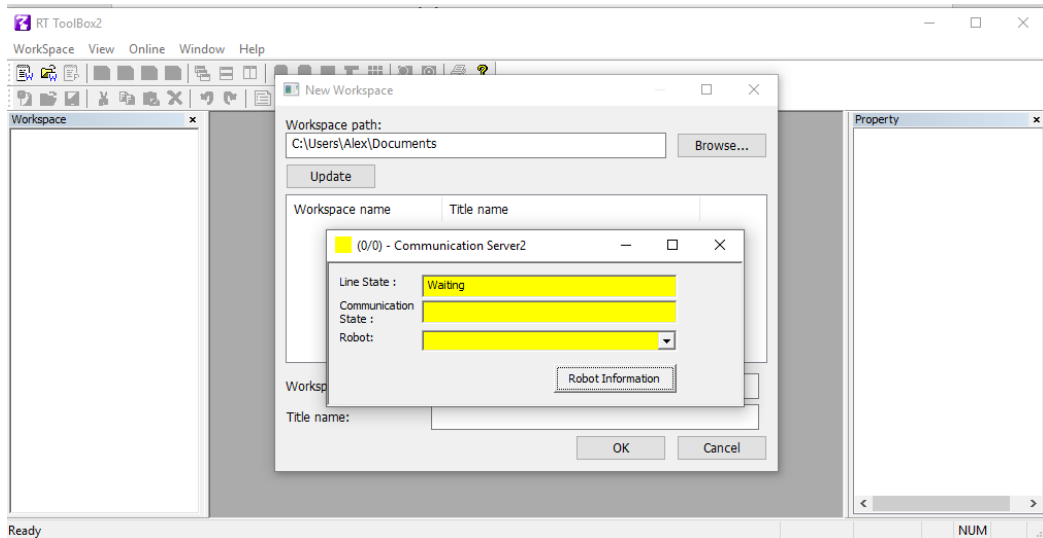


Figura 2-1: Panorama general de software de Programación de Brazo Robótico

Realizado por: Paucar, Alex; 2019

En la En la Figura 2-2se puede observar el software RT ToolBox2. El software para PC admite todo, desde el inicio del sistema hasta la depuración, simulación, mantenimiento y operación., es decir que se pudo programar y editar, verificar operativamente antes de instalar los robots, medir el tiempo de tacto del proceso, depurar durante el arranque del robot, monitorear del funcionamiento del robot después del arranque y solucionar problemas que surgían durante el proceso (*Features RT ToolBox2 Software Industrial Robots-MELFA | MITSUBISHI ELECTRIC FA, 2019*).

Para la programación del Robotino se utiliza el software Robotino® View:

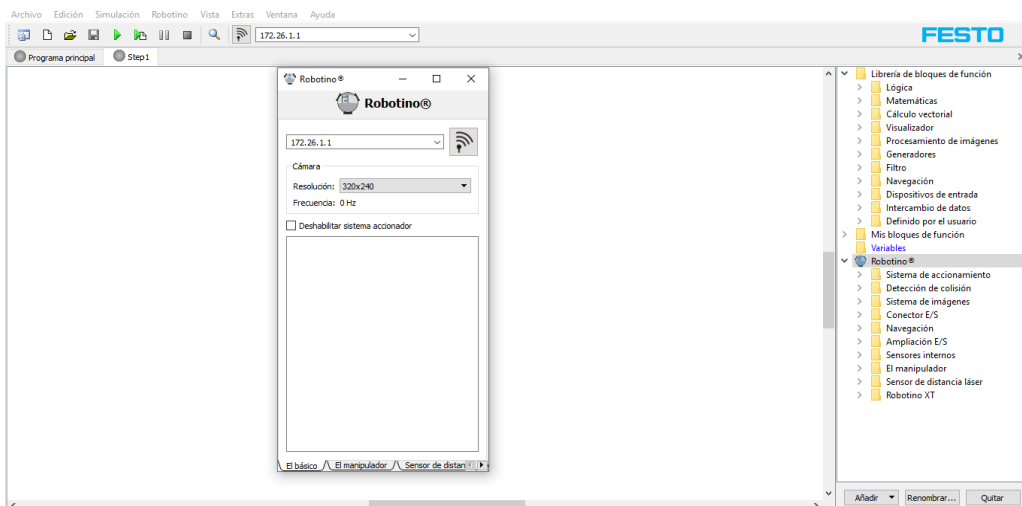


Figura 3-2: Panorama general de software de Programación de Robotino

Realizado por: Paucar, Alex; 2019

En la Figura 3-2 se puede ver el software de nuestro Robotino. El Software Robotino® View es un entorno de programación gráfico interactivo en cual se pueden crear e iniciar programas de mando para Robotino®.

En el entorno se pudo ejecutar las siguientes funciones:

- Los programas secuenciales se muestran en GRAFCET.
- Representación de componentes de hardware como bloques funcionales.
- Motores, E/S, sensores, cámara, odómetro, pinzas, unidades de manipulación, salida funcional, entrada de encoder
- Bloques funcionales para el procesamiento de imágenes
- Bloques funcionales para la navegación:
- Bloques funcionales para el intercambio de datos: UDP

(FESTO, 2013)

2.3.5.2 *Suministros de voltaje*

Brazo Robótico:

- Baterías: De litio ER6V/3.6V
- Voltaje: 220V

Robotino:

- Baterías: Recargables propias del Robot
- Voltaje: 110V

2.3.5.3 *Alta calidad de cámara*

El Robotino consta de una cámara que da transmisión en vivo:



Figura 4-2: Cámara del Robotino (parte roja)

Fuente: (FESTO Didactic, 2014b)

En la Figura 4-2 se puede ver la cámara que usa el Robotino para captar imágenes de su entorno. A continuación detallamos sus aspectos:

Función

La cámara de Robotino® nos da una imagen en vivo y directo, 1080p a 30 fps. La imagen que se obtiene se puede utilizar tanto para navegar, así como también para diferenciar colores y detectar obstáculos y objetos (FESTO Didactic, 2014b).

Configuración

La configuración de la resolución y de otros parámetros de la cámara se hace en Robotino® View, o bien recurriendo a los parámetros de interfaces del lenguaje de programación utilizado (FESTO Didactic, 2014b).

2.3.5.4 Sensores de distancia



Figura 5-2: Sensores ubicados en Robotino

Fuente: (FESTO Didactic, 2014e)

En la Figura 5-2 se puede observar los sensores de distancia que se encuentran adaptados al Robotino. A continuación detallamos sus aspectos:

Función

Los sensores de luz infrarroja permiten determinar la distancia que hay entre Robotino® y objetos que se encuentran en su entorno. Robotino® tiene, en total, nueve sensores de luz infrarroja, dispuestos en su base en ángulos de 40° entre sí. Cada uno de los sensores de distancia emite una tensión de un determinado voltaje. El voltaje depende de la distancia a la que se encuentra el objeto reflectante (FESTO Didactic, 2014e).

Configuración

Los sensores de distancia siempre están en estado activo (FESTO Didactic, 2014e) .

Montaje y conexión

Los sensores de distancia están montados en la parte inferior del chasis y están conectados a la placa de circuitos impresos de la unidad de electrónica funcional. Los sensores están conectados entre sí de tres en tres mediante un cable plano (FESTO Didactic, 2014e).

Tecnología

Un sensor de distancia de luz infrarroja permite efectuar mediciones absolutas o relativas de cualquier objeto que se encuentre a una distancia entre 4 y 30 centímetros. Su peculiaridad consiste en tener una conexión sencilla, compuesta únicamente de la alimentación de tensión y de una señal analógica de salida. La electrónica procesa la señal correspondiente a la distancia y emite una señal analógica de tensión. La siguiente imagen muestra la dependencia existente entre el valor de la tensión obtenido mediante un sensor de distancia y la distancia hasta el objeto reflectante (FESTO Didactic, 2014e).

2.3.5.5 Suficientes grados de libertad

Los grados de libertad del Brazo Robótico son la cantidad de movimientos independientes que determinan la última posición a la que puede llegar el efector final del mecanismo como un gripper. Según el criterio de Kutzbach y Grübler, para el cálculo de los grados de libertad de un mecanismo en 3 dimensiones (X Y Z), se lo hace mediante la ecuación 1:

$$GDL = 6(n - 1) - 5J_1 - 4J_2 - 3J_3 - 2J_4 - J_5 \quad \text{Ecuación 1}$$

Donde:

n = Número de eslabones del mecanismo

J_1 = Uniones de 1 grado de libertad

J_2 = Uniones de 2 grados de libertad

J_3 = Uniones de 3 grados de libertad

J_4 = Uniones de 4 grados de libertad

J_5 = Uniones de 5 grados de libertad

Figura 6-2: Uniones de Brazo Robótico

Fuente: (Borja & Fernanda, 2018)

En la Figura 6-2 se tiene unos de los pases que se necesitan para calcular los grados de libertad del robot. Para calcular los grados de libertad del manipulador se debe primero determinar el número de eslabones y juntas como se muestra en la figura

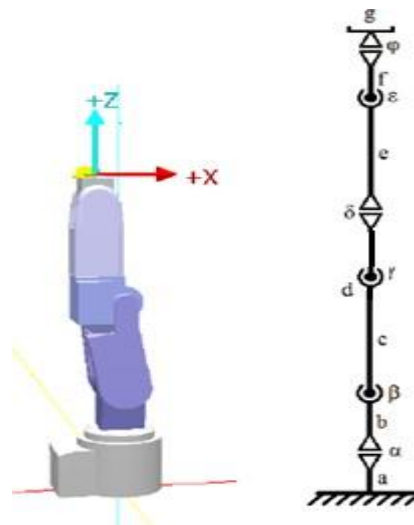


Figura 7-2: Eslabones y juntas del Mitsubishi MELFA RV – 2SDB

Fuente: (Borja & Fernanda, 2018)

Según la ecuación 1., el robot Mitsubishi RV – 2SDB posee: $J_1, J_2, J_3, J_4, J_5, J_6$ pares cinemáticos de revolución de 1 grado de libertad y 6 eslabones. Por tanto, este manipulador robótico posee 6 grados de libertad.

$$GDL = 6 (7 - 1) - 5(6)$$

$$GDL = 6 (6) - 30$$

$$GDL = 36 - 30$$

$$**GDL = 6**$$

Figura 8-22: Determinación de grados de libertad

Fuente: (Borja & Fernanda, 2018)

En la Figura 8-2 se puede apreciar el cálculo final para obtener los grados de libertad de nuestro Brazo Robótico.

2.3.5.6 *Alto grado de precisión en posicionamiento y movilidad de Robots*

Robotino® está equipado con un giroscopio, utilizado para aumentar la precisión de la determinación de posiciones. Para aumentar el grado de precisión, el giroscopio determina los cambios de la orientación de Robotino® (FESTO Didactic, 2014c).

Una vez que el sistema de Robotino® detecta la presencia del giroscopio, se aprovechan sus señales para corregir la determinación de las posiciones recurriendo a los datos obtenidos mediante su sistema de accionamiento (odometría). El usuario no tiene que llevar a cabo programación alguna (FESTO Didactic, 2014c).

Los giroscopios miden velocidades angulares basándose en el mantenimiento del impulso de rotación. Si se intenta hacer girar un objeto rotatorio sobre un eje que no es el eje sobre el que dicho objeto está rotando, el objeto ejercerá un momento de giro contrario al movimiento con el fin de preservar el impulso de rotación total (FESTO Didactic, 2014c).

2.3.5.7 *Módulo de interfaz para comunicación con Robots*

Unidad de Interfaz de Robot 534364

Diseño

A continuación, representamos las vistas de la unidad con sus respectivas estructuras y funcionalidades.

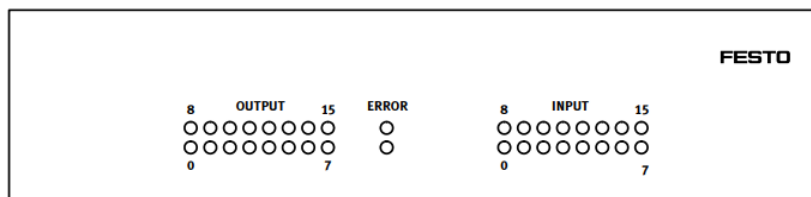


Figura 9-3: Vista frontal de la unidad de interfaz

Fuente: (FESTO, 2003)

La Figura 9-2 enfoca la parte frontal de la unidad de interfaz del robot que está equipada de la siguiente manera:

- 16 LED verdes y 16 LED amarillos, los cuales sirven para la visualización del estado de las entradas / salidas digitales del robot
- 2 LED rojos que sirven para la visualización del estado de error de las salidas.

(FESTO, 2003)

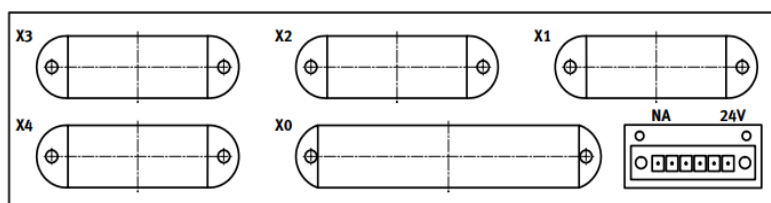


Figura 10-2: Vista posterior de la unidad de interfaz

Fuente: (FESTO, 2003)

La Figura 10-2 enfoca la parte posterior de la unidad de interfaz del robot que está equipada de la siguiente manera:

- Cuatro enchufes centrónicos de 24 pines (X1 - X4) para la conexión de dispositivos periféricos.
- Un enchufe centrónico de 50 pines (X0) para la conexión de la unidad de control del robot (RVAJ).
- Un enchufe de terminal de tornillo de 6 pines (NA 24 V) para la conexión de la tensión de alimentación de 24 V y la realizar la desconexión de la parada de emergencia de las salidas.

(FESTO, 2003)

Función

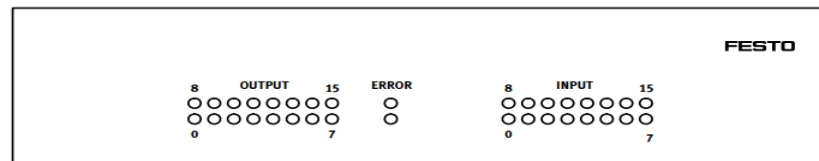


Figura 11-2: Esquema de funcionalidad de la unidad en base a cada LED

Fuente: (FESTO, 2003)

De la Figura 11-2 se puede detallar los siguientes aspectos:

- Los LEDs amarillos “OUTPUT” de 0 a 7 y 8 a 15 indican el estado de salidas de 0 a 7 y de 8 a 15 respectivamente.
- Los LEDs verdes “INPUT” de 0 a 7 u 8 a 15 indican el estado de las entradas de 0 a 7 y de 8 a 15 respectivamente de la interfaz de E / S de la unidad de accionamiento.

El estado de error de los controladores de salida se muestra a través de los LED rojos. El LED superior o inferior se ilumina si una de las salidas 8 a 15 o 0 a 7 respectivamente está en cortocircuito o está sobrecargada térmicamente. Las 16 entradas / salidas digitales de la unidad de accionamiento del robot conectadas al enchufe X0 están distribuidas a los enchufes desde X1 a X4 a través de la interfaz del robot. Las E / S están distribuidas de tal forma que facilitan el control de una estación de montaje MPS completa vía la unidad de accionamiento.

- Enchufe X1
Las entradas del robot de I8 a I15 y las salidas de Q8 a Q15 se encuentran en el enchufe X1, por lo que más que X1 8 Entradas/ Salidas son habilitadas para el control de la Estación de montaje de MPS.
- Enchufe X2
El enchufe X2 proporciona entradas de robot I1, I2, I6 e I7 en los pines 14, 15, 13 y 20. Este enchufe es para la conexión de las entradas de la estación de robot
- Enchufe X3
El enchufe X3 proporciona las entradas I3, I4, I5 en los pines 13, 14, 15, así como salidas Q0, Q1, Q2 y Q3 en los pines 1, 2, 3 y 4. Este enchufe facilita la conexión de un panel de control.
- Enchufe X4
El enchufe X4 proporciona el resto de las salidas d robot Q4, Q5, Q6 y Q7 en los pines 5, 6, 7 y 8.

(FESTO, 2003)

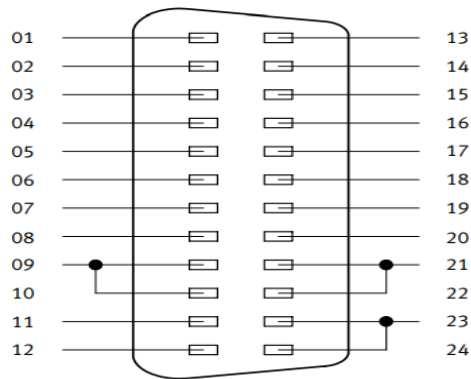


Figura 12-2: Asignación de pines del conector centrónico de 24 pines

Fuente: (FESTO, 2003)

De la Figura 12-2 se describe el enchufe NA 24V. La fuente de alimentación de 24 V a la unidad de interfaz se realiza a través de este enchufe. La desconexión de salidas específicas se puede realizar a través de dos puentes NA1 y NA2. NA1 desconecta las salidas Q0 a Q7. NA2 apaga las salidas Q8 a Q15. Esto facilita un medio simple para integrar un apagado de parada de emergencia existente.

(FESTO, 2003)

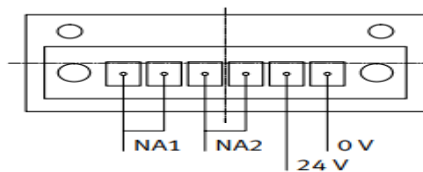


Figura 13-2: Modo de conexión del enchufe NA 24V

Fuente: (FESTO, 2003)

En la Figura 13-2 se puede ver el modo de conexión para la alimentación del módulo con 24 V.

Las siguientes tablas enlistan la distribución de las Entradas/ Salidas de forma resumida:

Tabla 2-1: Salidas con sus respectivos pines

X0		X1		X2		X3		X4	
Salida	Pin	Salida	Pin	Salida	Pin	Salida	Pin	Salida	Pin
Q0	4					1	Q0		
Q1	5					2	Q1		
Q2	6					3	Q2		
Q3	7					4	Q3		
Q4	29							5	Q4
Q5	30							6	Q5
Q6	31							7	Q6
Q7	32							8	Q7
Q8	10	1	Q0						
Q9	11	2	Q1						
Q10	12	3	Q2						
Q11	13	4	Q3						
Q12	35	5	Q4						
Q13	36	6	Q5						
Q14	37	7	Q6						
Q15	38	8	Q7						

Fuente: (FESTO, 2003)

Tabla 3-2: Entradas con sus respectivos pines

X0		X1		X2		X3		X4	
Salida	Pin	Salida	Pin	Salida	Pin	Salida	Pin	Salida	Pin
I0	15								
I1	16			14	I1				
I2	17			15	I2				
I3	18					13	I0		
I4	19					14	I1		
I5	20					15	I2		
I6	21			13	I0				
I7	22			20	I7				
I8	40	13	I0						
I9	41	14	I1						
I10	42	15	I2						
I11	43	16	I3						
I12	44	17	I4						
I13	45	18	I5						
I14	46	19	I6						
I15	47	20	I7						

Fuente: (FESTO, 2003)

Datos técnicos

Tabla 4-2: Datos de Unidad de Interfaz de Robot

Descripción	Valor
Suministro	24 V \pm 20%
Número de salidas	16 digitales
Voltaje de salida	24 V DC
Corriente de salida	0.3 A/ Salida total máx. 4.5 A
Protección contra cortocircuitos	Sí, , pantalla a través de LED
Protección contra sobrecarga térmica	Sí, , pantalla a través de LED
Número de entradas	16 digitales
Nivel Trigger ON	$V_{in} > 8 V$ or $I_{in} > 2 mA$
Nivel Trigger OFF	$V_{in} > 4 V$ or $I_{in} > 1 mA$

Continúa

Continúa

Voltaje de entrada/Corriente de entrada	12 V/ 3 mA, 24 V/7 mA
Clasificación de protección	IP20
Temperatura ambiente admisible	0 - + 55 ^o C
Temperatura de almacenamiento permitida	0 - + 70 ^o C
Compatibilidad electromagnética	CE
Dimensiones	Largo: 215 mm Ancho: 220 mm Altura: 55 mm
Peso	900 g

Fuente: (FESTO, 2003)

2.4 Detalles de costos

Tabla 5-2: Detalles de presupuesto

Cantidad total	Detalle	V. Unitario	V. Total
	HARDWARE		
8	Sensores	15,00	120,00
1	Router	15,00	15,00
1	PLC	400,00	400,00
1	Caja de Interfaz de Robot	1800,00	1800,00
	SOFTWARE	0,00	0,00
	Software de diseño		0,00
	Software de programación del Robotino	0,00	0,00
	Software de programación del Brazo Robótico	0,00	0,00
	Software de programación de PLCs	0,00	0,00
	MATERIALES A UTILIZAR		
	Elementos eléctricos y electrónicos	200,00	200,00
4	Baterías TOSHIBA de litio	20,00	100,00
2	Fuentes de 24 Vdc	30,00	60,00
	OTROS		
	Construcción de estructuras mecánicas	50,00	50,00
	Internet	5,00	5,00
	Copias	10,00	10,00
	COSTO TOTAL		2760,00

Realizado por: Paucar, Alex; 2019

En la Tabla 5-2 se observa una lista detallada de los materiales a utilizar tanto a nivel de software y hardware, una obtenido el costo total se decidió que si se va a implementar el proyecto.

IMPLEMENTACIÓN

Éste proceso consta de las siguientes etapas:

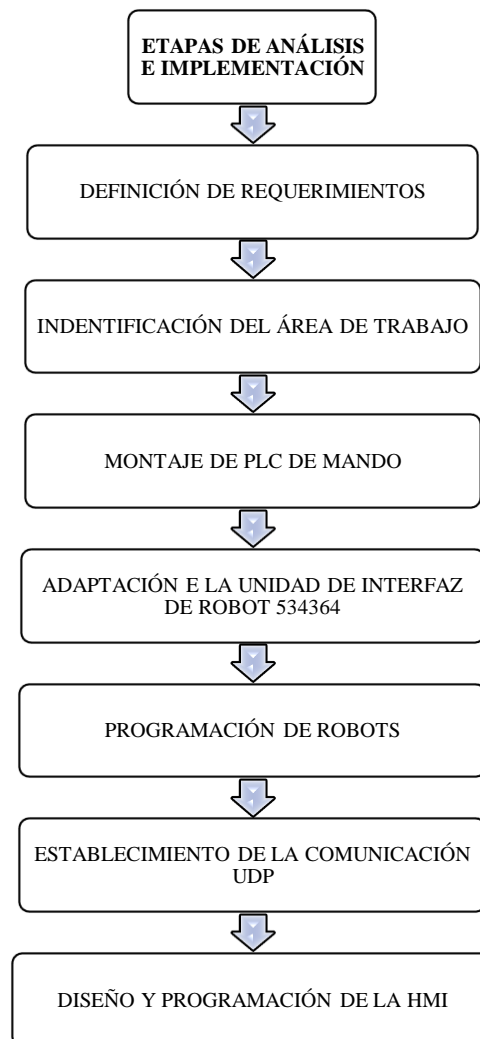


Gráfico 1-2: Etapas de Análisis e Implementación de un sistema inalámbrico

Realizado por: Paucar, Alex; 2019

En el Gráfico 1-2 se detalla cada una de las etapas que se realizan para el desarrollo del proyecto de inicio a fin.

2.5 Características del Área de Trabajo

La implementación de la tesis en su totalidad se va a desarrollar en el laboratorio de “Automatización Industrial” en la Escuela de Ingeniería Electrónica en Control y Redes Industriales de la Escuela Superior Politécnica de Chimborazo (ESPOCH).

La investigación presenta un enfoque científico, enfoque en el cual se va a manipular y controlar variables para posteriormente medirlas y probarlas. No obstante, se determina que se realizará un trabajo de tipo experimental.

A continuación, se detalla las medidas de las estaciones de nuestra área de trabajo:

Mesa de trabajo

- Ancho: 2.5 m
- Largo: 2.5 m
- Altura: 0.8 m

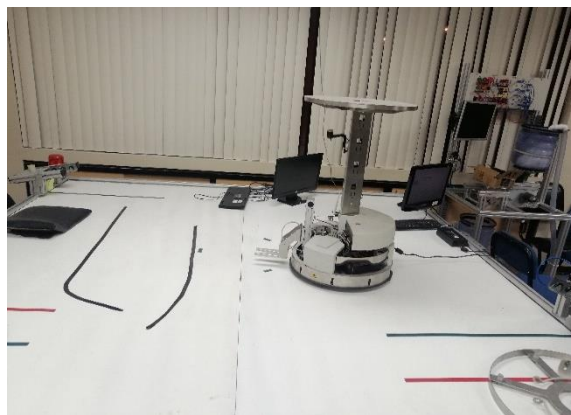


Figura 14-2: Mesa de trabajo del sistema

Realizado por: Paucar, Alex; 2019

De la Figura 14-2 se detalla los siguientes aspectos: La estructura está diseñada con barras de aluminio de 0.04 m de ancho y 0.08 m de grosor. Para sostener la base sobre la cual el Robotino PLUS desarrollará su trabajo se han colocado 4 patas con ruedas, las mismas que además facilitarán el traslado de la mesa de un lugar a otro dependiendo de las necesidades del sistema.

Estación del Brazo Robot RV2SDB

- Ancho: 0.7 m
- Largo: 0.7 m
- Altura: 0.5 m



Figura 15-2: Estación Brazo Robótico

Realizado por: Paucar, Alex; 2019

En la Figura 14-2 se presenta la estructura de la estación del robot. La superficie sirve como base sobre la cual se ubica el Robot. Además, sobre la misma superficie están colocados dos soportes para los pallets.

Tablero de Control

- Ancho: 0.31 m
- Largo: 0.40 m
- Altura: 0.05 m



Figura 16-2: Tablero de control del sistema

Realizado por: Paucar, Alex; 2019

De la Figura 16-2 se detalla lo siguiente: En un extremo se ha ubicado el tablero de control, en dicha superficie está ubicado el PLC y las fuentes de voltaje, por ende, también tenemos ubicados las borneras para las perspectivas salidas y entradas del controlador lógico programable.

Base de la Unidad de Interfaz de Robot 534364

- Ancho: 0.30 m
- Largo: 0.40 m

- Altura: 0.03 m



Figura 17-2: Base de la unidad 534364

Realizado por: Paucar, Alex; 2019

En la Figura 17-2 se tiene una base metálica que sirve de soporte al módulo 534364 y su respectiva fuente de voltaje.

2.6 Condiciones del Área de Trabajo



Figura 18-2: Laboratorio de Automatización

Realizado por: Paucar, Alex; 2019

En la Figura 18-2 se observa el área de trabajo. El proyecto se desarrolla en laboratorio de automatización de la Facultad de Informática y Electrónica, en donde se pudo tomar en cuenta tres aspectos: las condiciones físicas y las condiciones organizativas.

En lo referente a las condiciones físicas se puede mencionar los siguientes aspectos:

- Cuenta con una iluminación adecuada.
- Amplio espacio físico para desenvolverse
- Consta de tomacorrientes de 110 v y 220 V necesario para el Brazo Robótico
- Posee varios puntos de acceso a internet

En relación a las condiciones organizativas se puede mencionar los siguientes aspectos:

- Se logró trabajar en horarios matutinos y vespertinos, de modo que se organizó adecuadamente el desarrollo de cada una de las etapas del proyecto.

2.7 Adecuación de la Mesa de Trabajo



Figura 19-2: Colocación de canaletas para cableado

Realizado por: Paucar, Alex; 2019

En la Figura 19-2 se ilustra las actividades que se llevaron a cabo en el lugar de trabajo:

- Acoplamiento de cilindros y válvulas 4/2 a la entrada y salida de cada pallet de la estación del Brazo Robótico.
- Acoplamiento de sensores inductivos en las denominadas bodegas, entrada y salida de la estación del Brazo Robótico y salida del sistema.
- Ubicación y ajuste de canaletas en la parte inferior de la mesa de trabajo para el cableado.
- Cableado de cada uno sensores inductivos desde el lugar donde se ubicaron hasta el tablero de control.
- Cableado de unidad de interfaz de Robot con el Brazo Robótico
- Conexión de 2 fuentes de 24 VDC tanto para el PLC como para la unidad de Interfaz.
- Acoplamiento de la estación del Brazo Robótico con la mesa de trabajo mediante barras de aluminio
- Colocación de un Router para la dotación de Wi-Fi al sistema
- Colocación de mangueras para dotar de aire a las válvulas
- Limpieza de la mesa de trabajo.

2.8 Montaje de Ampliación de Robotino

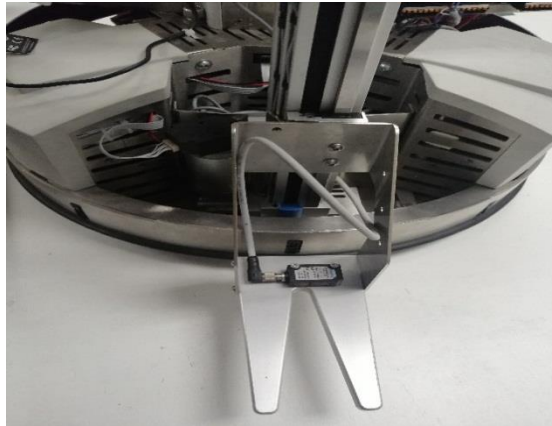


Figura 20-2: Adaptación montacargas del Robotino

Realizado por: Paucar, Alex; 2019

En la Figura 20-2 se observa el acoplamiento que se realizó al Robotino. Para transportar los pallets de un punto a otro el Robotino necesita una ampliación denominada montacargas. El montacargas tiene una estructura a manera de pinzas, lo que le permite levantar y transportar objetos.

Para el correcto funcionamiento de la extensión se debe tener una cinta dentada controlada por un motor, el cual conecta el puente de mando del robot a un puente H controlado por una señal PWM y un bit de dirección.

Además, el montacargas consta de dos sensores magnéticos que se conectan a los conectores E/S, logrando de este modo evitar daños por atascamientos o golpes. Como es lógico se necesita ordenar la conexión de los sensores adicionales en el puente de control E/S.

2.9 Montaje del PLC de mando



Figura 21-2: Montaje de PLC en Tablero

Realizado por: Paucar, Alex; 2019

En la Figura 21-2 se observa el montaje del PLC. Al requerir nuestro sistema de comunicación un control automático del proceso utilizamos un PLC S7 -1200, el cual tiene las características requeridas y cumple con las condiciones necesarias para el proceso en cuestión y sobre todo se puede trabajar con comunicación con protocolo UDP.

El PLC nos permite la comunicación con el Robotino y el Brazo Robótico, los cuales una vez las señales iniciaron el proceso de transporte y paletización respectivamente.

2.9.1 Estructura o arquitectura General del PLC

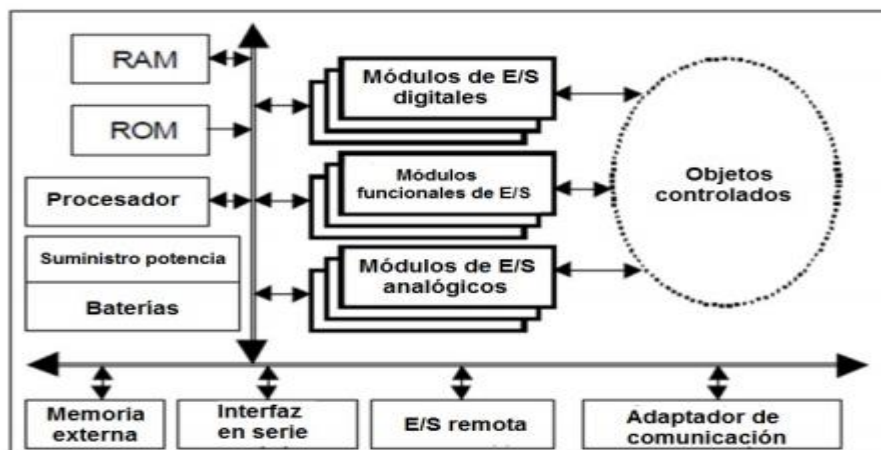


Gráfico 2-1: Diagrama generalizado de un PLC

Fuente: (C. Pérez, 2011)

En el Gráfico 2-2 se puede observar un esquema general de un PLC. Es necesario que exista un suministro de potencia cuyo propósito principal es garantizar los voltajes de operación internos

del controlador y sus bloques, tenemos la parte principal que es la “unidad central de procesamiento” o CPU que contiene la parte de procesamiento del controlador y está basada en un microprocesador que permite utilizar aritmética y operaciones lógicas para realizar diferentes funciones (C. Pérez, 2011).

También se tiene el lugar donde se guardan los datos y las instrucciones es la memoria que se divide en memoria permanente, PM, y memoria operacional, conocida como memoria de acceso aleatorio o RAM. Finalmente, tenemos los módulos de E/S que son aquellos módulos de señal (SM) que coordinan la entrada y salida de las señales, con aquellas internas del PLC. Estas señales pueden ser digitales (DI, DO) y analógicas (AI, AO), y provienen o van a dispositivos como sensores, interruptores, actuadores, etc (C. Pérez, 2011)

En la siguiente figura tenemos el esquema de modo de trabajo de un PLC, el cual actúa o controla sobre un proceso a través de una interfaz de operación constituida por el controlador y las respectivas entradas y salidas.

En esta parte se muestra el control del PLC sobre el sistema de comunicación inalámbrico:

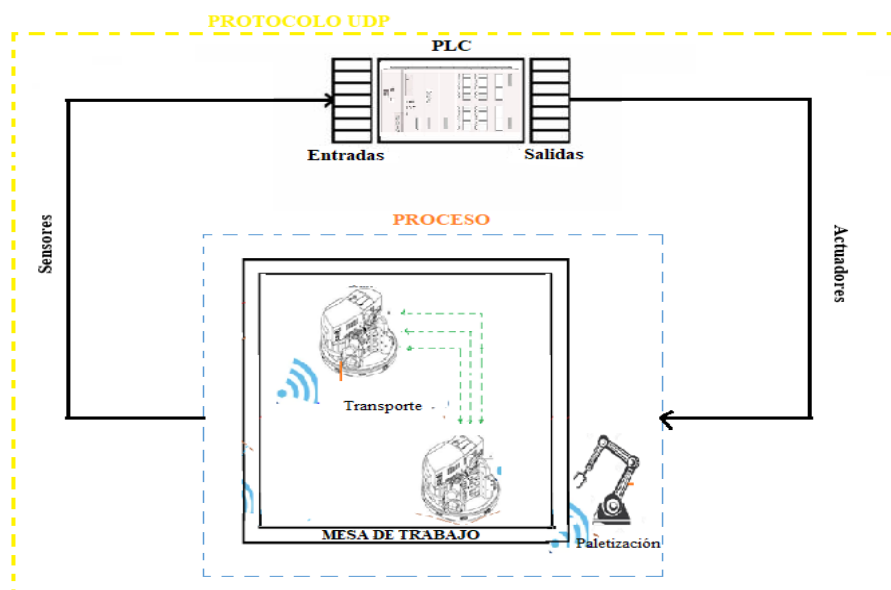


Gráfico 3-2: Proceso de comunicación inalámbrica controlado desde un PLC

Realizado por: Paucar, Alex; 2019

En el Gráfico 3-2 se presenta la esquematización del control que realiza el PLC sobre el proceso de comunicación inalámbrica mediante el protocolo UDP. En mencionado proceso tenemos la intervención del Robotino y Brazo Robótico, los cuales bajo los estándares del protocolo realizan el transporte y la paletización respectivamente.

2.9.2 Asignación de dirección

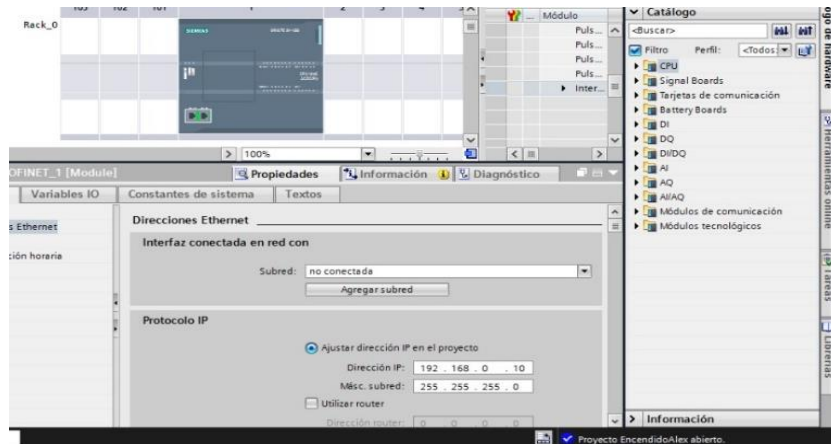


Figura 22-2: Direccionamiento IP del PLC

Realizado por: Paucar, Alex; 2019

En la Figura 22-2 se observa el direccionamiento del controlador. El PLC tiene una dirección IP asignada 192.168.45, la misma que nos permitirá comunicarnos con otros ordenadores.

2.9.3 Conexiones (Suministro de voltaje, salidas y entradas)

Salidas y entradas

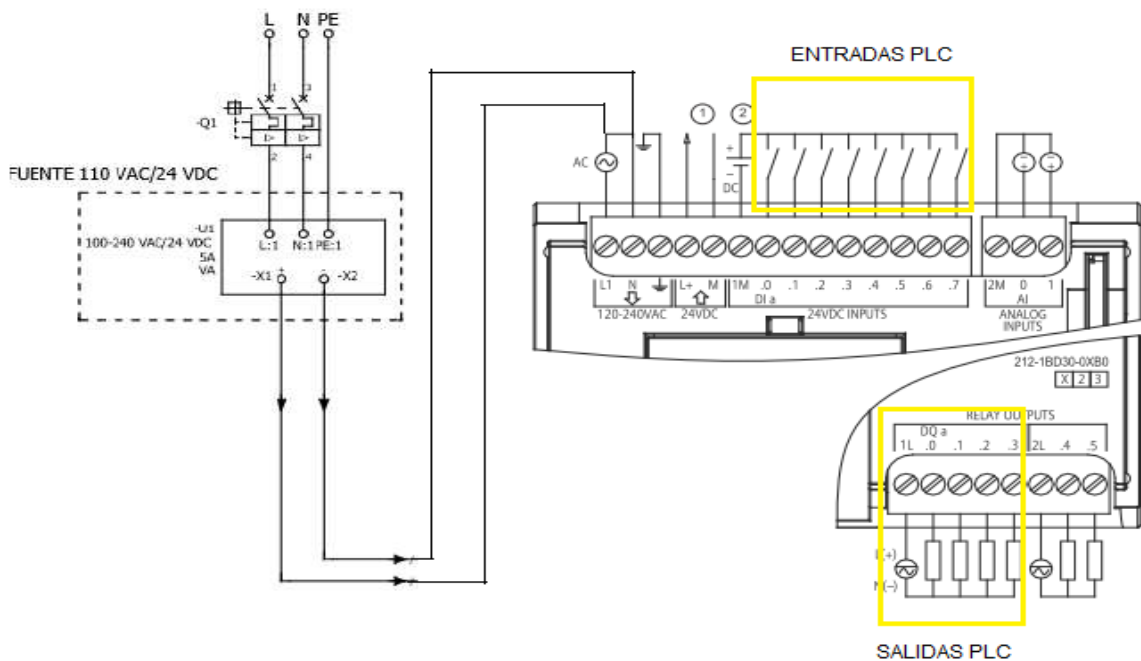


Figura 23-2: Conexiones a borneras de PLC

Realizado por: Paucar, Alex; 2019

En la Figura 23-2 se observa la ubicación de las entradas salidas y conexiones a fuente de voltaje en el PLC.

Tabla 6-2: Conexiones a PLC

Simbología	Descripción
L	Conexión a línea 110 VAC
N	Conexión a neutro 110 VAC
PE	Conexión a tierra 110 VAC
X1	Bornera de conexión a terminal positivo de alimentación 24 VDC
X2	Bornera de conexión a terminal negativo de alimentación 24 VDC
I1	Sensor inductivo de la bodega 12
I2	Sensor inductivo de la bodega 1
I3	Sensor inductivo de la salida del sistema
I4	Sensor inductivo de la entrada del robot RV2SDB
I5	Sensor inductivo de la salida del robot RV2SDB
Q1	Sensor magnético del cilindro 1
Q2	Sensor magnético del cilindro 1

Realizado por: Paucar, Alex; 2019

En la Tabla 6-2 se detallan las conexiones de las entradas a los sensores, las salidas a las válvulas y las conexiones a Línea, Neutros y Tierra del PLC.

2.10 Adaptación de la Unidad de Interfaz de Robot 534364

2.10.1 Acople

Paso 1:



Figura 24-2: Colocación de sensores para detectar pallets

Realizado por: Paucar, Alex; 2019

En la Figura 24-2 se ve la colocación de un sensor inductivo en la base del pallet para su respectiva detección, de éste modo generando una señal de activación que dará inicio a la tarea que realizará el Brazo Robótico.

Paso 2:

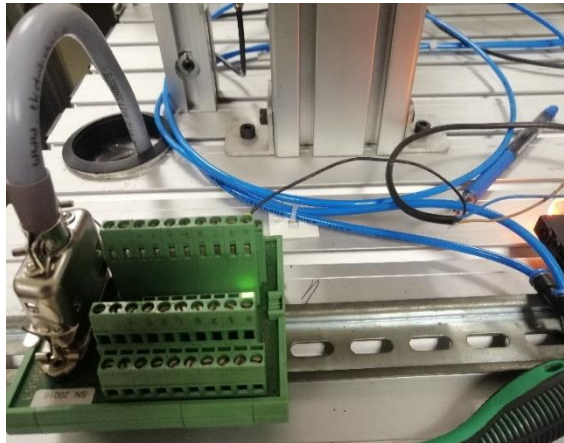


Figura 25-2: Conexión Syslink

Realizado por: Paucar, Alex; 2019

En la Figura 25-2 se ve que se coloca un Terminal E/S SysLink (estructura verde) a donde se conectarán las señales de los sensores detectores de pallets. En la terminal también se debe conectar el cable de datos de E/S con conectores SysLink (color plomo) que irán al Unidad de Interfaz de Robot 534364:

Paso 3:



Figura 26-2: Conexión de entradas y salidas a la caja de Interfaz

Realizado por: Paucar, Alex; 2019

En la Figura 26-2 se tiene el cable de datos conectado en la Terminal SysLink que se debe conectar al socket 1, 2, 3 o 4 dependiendo de las entradas y salidas que se desee manejar.

Paso 4:



Figura 27-2: Conexión Interfaz-Controlador

Realizado por: Paucar, Alex; 2019

La Figura 27-2 se muestra la utilización de un cable adaptador I/O para conectar el Socket 0 de la caja de Interfaz con el interfaz 2D-TZ378 E/A del controlador del Brazo Robótico.

Paso 5:



Figura 28-2: Estado activo de salidas y entradas de caja Interfaz

Realizado por: Paucar, Alex; 2019

La Figura 28-2 hace referencia a la utilización de una fuente de 24 Vdc para alimentar la caja de Interfaz y procedemos a comprobar si las entradas y salidas se activan.

2.11 Puesta en marcha de Robots

Dependiendo de cada robot se realiza las acciones necesarias para ponerlos en funcionamiento, entre las principales acciones está la programación en software. De éste modo se logra que los robots realicen las tareas asignadas.

2.11.1 Brazo Robótico Mitsubishi RV2SDB

2.11.1.1 Colación de baterías y conexión a suministro de voltaje



Figura 29-2: Suministro de voltaje a Brazo Robótico

Realizado por: Paucar, Alex; 2019

La Figura 29-2 se observa la colocación de las baterías en el robot. Para encender el Brazo Robótico se debe conectar desde una toma de 220 V al controlador del mismo. Además, en la parte trasera de la base del robot se necesitan conectar 4 Pilas de Litio Toshiba ER6V de 50 x 14,45 mm, las cuales tienen una capacidad de 2400mAh y suministran un voltaje de 3,6V.

2.11.1.2 Programación en software

1	PROGRAMA PARA TESIS CON UDP'
2	'Llevada de pallet Entrada Robot a Home 2'
3	servo on
4	hopen 1
5	dly 2
6	ovrd 30
7	dly 2
8	ovrd 30
9	mov Phome
10	dly 2
11	ovrd 30
12	mov P41
13	dly 2
14	Ovrd 30

<

Add Edit Delete

VV7 Y V 7 A

Figura 30-2: Código en Software RT Toolbox2

Realizado por: Paucar, Alex; 2019

La Figura 30-2 nos deja ver que se ha programado con RT Toolbox 2. La secuencia del programa inicia en una posición de inicio denominada HOME, el conjunto de movimientos comienza una vez que el sensor ubicado en la bodega de entrada del Robot detecte la presencia del pallet. Desde Paso 1 al Paso 7 son movimientos encargados de transportar el pallet desde el home 4 (bodega entrada de brazo Robótico) vacío hacia la estación el Home 1 o Home 2. Una vez que se ejecutan estas instrucciones vienen el siguiente bloque de instrucciones desde Paso 8 hasta Paso 40 son movimientos encargados de transportar cada una de las piezas desde la estación Home 5 hacia el Home 1 o Home 2, esta subrutina se repite cuatro veces. En la parte final están los movimientos encargados de llevar el pallet lleno desde el Home 1 o Home 2 hacia el Home 3 (bodega de salida del Robot), Paso 41 al Paso 47.

A continuación, explicamos la numeración de los puntos y mostramos la secuencia de los movimientos en tablas:

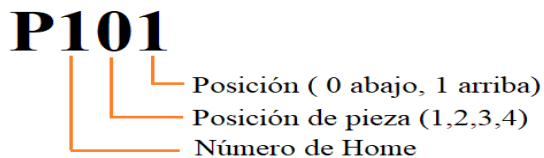


Figura 31-2: Descripción de Punto posición

Realizado por: Paucar, Alex; 2019

La Figura 31-2 detalla la posición de los puntos utilizados en la programación:

P30: Punto abajo home 3

P31: Punto arriba home 3

P40: Punto abajo para recolección de pieza

P41: Punto abajo para recolección de pieza

P50: Punto abajo para recolección de pieza

P51: Punto arriba para recolección de pieza

P52: Punto abajo para detección de pieza con Bit 900

P53: Punto abajo para detección de pieza con Bit 900

Tabla 7-2: Pasos y Puntos de la ruta del Robot Mitsubishi RV2SDB con Home 1

Paso	XYZ	Paso	XYZ	Paso	XYZ
1	Phome	17	P53	33	P53
2	P41	18	P52	34	P52
3	P40	19	P51	35	P51
4	P41	20	P50	36	P50
5	P101	21	P51	37	P51
6	P100	22	P121	38	P141
7	P101	23	P120	39	P140
8	PHome	24	P121	40	P141
9	P53	25	P53	41	Phome
10	P52	26	P52	42	P100
11	P51	27	P51	43	P101
12	P50	28	P50	44	P31
13	P51	29	P51	45	P30
14	P111	30	P131	46	P31
15	P110	31	P130	47	Phome
16	P111	32	P131		

Realizado por: Paucar, Alex; 2019

En la Tabla 7-2 se muestra el número y detalle de cada uno de los pasos que realiza el Robot para completar la parte de secuencia con Home 1.

Tabla 8-2: Pasos y Posiciones de la ruta del Robot Mitsubishi RV2SDB con Home 2

Paso	XYZ	Paso	XYZ	Paso	XYZ
1	Phome	17	P53	33	P53
2	P41	18	P52	34	P52
3	P40	19	P51	35	P51
4	P41	20	P50	36	P50
5	P201	21	P51	37	P51
6	P200	22	P221	38	P241
7	P201	23	P220	39	P240
8	PHome	24	P221	40	P241
9	P53	25	P53	41	Phome
10	P52	26	P52	42	P200
11	P51	27	P51	43	P201
12	P50	28	P50	44	P31
13	P51	29	P51	45	P30
14	P211	30	P231	46	P31
15	P210	31	P230	47	Phome
16	P211	32	P231		

Realizado por: Paucar, Alex; 2019

En la Tabla 8-2 se muestra el número y detalle de cada uno de los pasos que realiza el Robot para completar la parte de secuencia con Home 2.

2.11.2 Puesta en marcha del ROBOTINO PLUS

2.11.2.1 Colación de baterías y conexión a suministro de voltaje



Figura 32-2: Baterías de Robotino

Fuente: (FESTO Didactic, 2014a)

En la Figura 32-2 se observa las baterías que requiere el robot para su funcionamiento. El Robotino necesita de dos acumuladores de plomo/ácido de 12 voltios que alimentan una tensión de 24 voltios de corriente continua a Robotino®.

Montaje y conexión

Los dos acumuladores están montados en el chasis de Robotino® y están conectados a la placa de circuitos impresos de la unidad de electrónica funcional a través de la electrónica de carga. Para recargar los acumuladores, debe conectarse la unidad de alimentación a la conexión que se encuentra en la parte posterior de Robotino® a una toma de 110V.

(FESTO Didactic, 2014a)

2.11.2.2 Interfaz Entradas y Salidas



Figura 33-2: Conexión a entradas digitales

Realizado por: Paucar, Alex; 2019

En la Figura 33-2 se tiene el módulo de entradas y salida del Robotino. Detallamos lo siguiente:

Función

El Robotino tiene un módulo de entradas y salidas al cual se le conectan las señales de los diferentes sensores que forman parte del Robot (FESTO Didactic, 2014d).

Montaje y conexión

En éste caso se conectan señales de 2 sensores que sirven para indicar la posición (abajo y arriba) del Montacargas a las entradas DI5 y DI6 del Robot (FESTO Didactic, 2014d).

2.11.2.3 Cámara del Robot



Figura 34-2: Cámara del Robotino (parte roja)

Fuente: (FESTO Didactic, 2014b)

En la Figura 34-2 se aprecia la cámara del robot para la toma de imágenes.

Función

La cámara de Robotino® nos da una imagen en vivo y directo. La imagen que se obtiene se puede utilizarse tanto para navegar, así como también para diferenciar colores y detectar obstáculos y objetos (FESTO Didactic, 2014b).

Configuración

La configuración de la resolución y de otros parámetros de la cámara se hace en Robotino® View, o bien recurriendo a los parámetros de interfaces del lenguaje de programación utilizado (FESTO Didactic, 2014b).

Montaje y conexión

La cámara está atornillada a la chapa frontal de Robotino® y está unida a la unidad de control a través del puerto USB (FESTO Didactic, 2014b).

2.11.2.4 Programación en software

Se utilizó Robotino View para la programación, en el programa tenemos una pestaña principal en donde se muestra la secuencia global y otras pestañas llamadas subprogramas en donde se muestra de manera detallada la programación de cada bloque. El programa principal que desarrollamos para el Robotino plus, tiene 28 subprogramas para cada ejecución de tarea. Entre los principales subprogramas el control de posicionamiento del robot con odometría, detección de colores, seguimiento de línea, detección de distancias, etc.

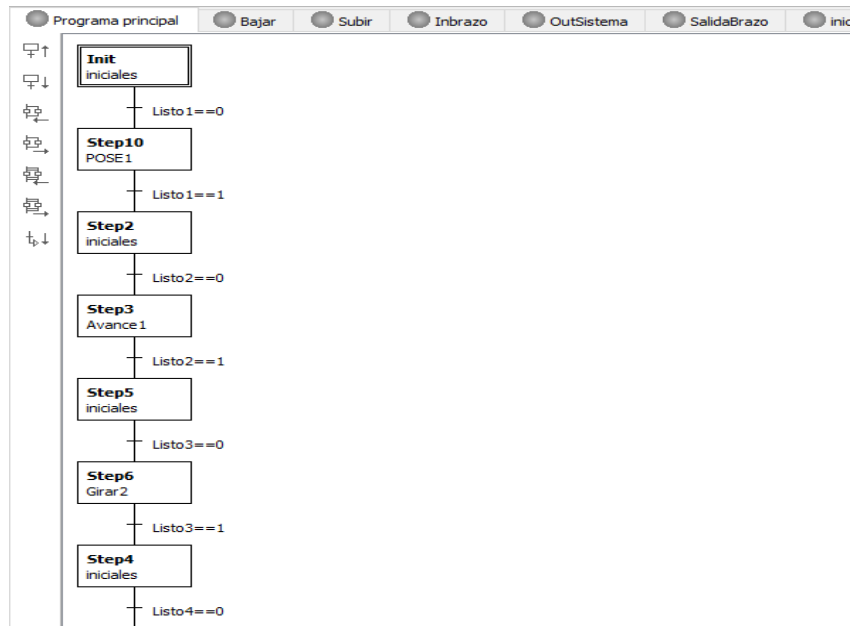


Figura 35-2: Secuencia en software Robotino

Realizado por: Paucar, Alex; 2019

En En la Figura 35-2 se puede apreciar el programa principal en donde están todos los pasos o subprogramas a seguir, para decidir si el proceso pasa de un bloque a otro se declaran variables globales que toman un valor determinado al cabo de cumplir con un proceso.

Detección de distancia y control de Montacargas

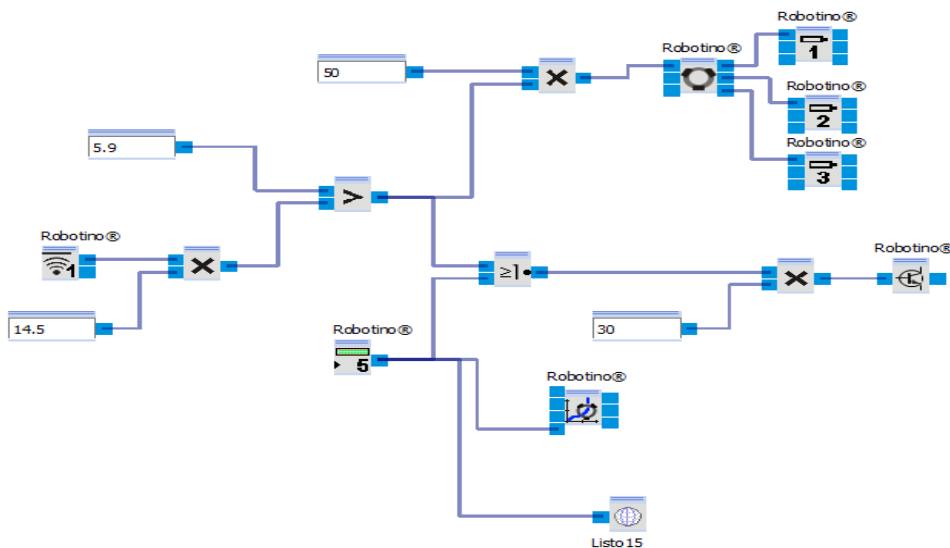


Figura 36-2: Subprograma-sensores de distancia y control Montacargas

Realizado por: Paucar, Alex; 2019

En la Figura 36-2 se observa la programación en Robotino View para el control de Montacargas. Éste subprograma se puede ver el uso del sensor de distancia 1, el cual permitirá

detectar al Robotino un objeto a cierta distancia y detenerse. También en éste subprograma se trabaja con las entradas 5 y 6 que sirven para el control del movimiento del aditamento instalado en el Robotino, con la ayuda de sensores magnéticos como fin de carrera conectados en las entradas digitales 5 y 6 respectivamente del load in bay, los valores constantes se refieren a la velocidad y la dirección del movimiento del montacargas, en nuestro caso 20 para tomar un pallet y -20 para dejar un pallet en una rampa, ya sea del robot RV2SDB o la salida del sistema (Sánchez Basantes & Sandoval Parra, 2018).

Posicionamiento con Odometría

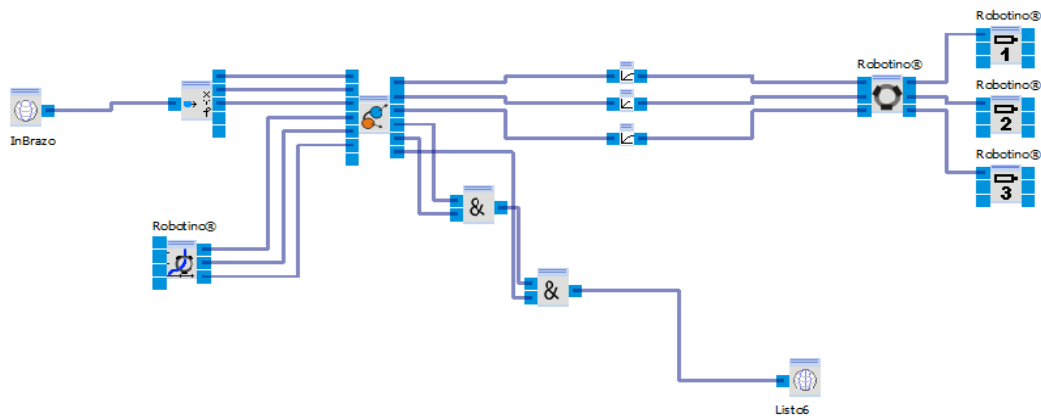


Figura 37-2: Subprograma Posicionamiento

Realizado por: Paucar, Alex; 2019

En la Figura 37-2 se parecía la programación para posicionamiento con Odometría. Se realiza el movimiento del robot de un punto a otro con control de posicionamiento (odometría). El controlador de posición permite establecer una posición cero o de origen y dar una posición final (punto a donde va a llegar). Se ingresa 3 parámetros, valor x, valor en y y ángulo ϕ .

Procesamiento de imágenes: Detección de colores y seguidor de línea

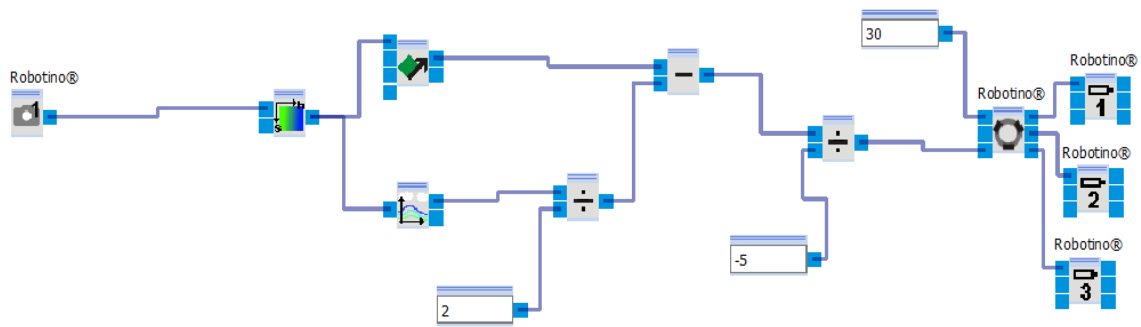


Figura 38-2: Subprograma seguir línea de color

Realizado por: Paucar, Alex; 2019

En la Figura 38-2 se observa la programación para procesamiento de imágenes. El posicionamiento se hace mediante el reconocimiento de las líneas ubicadas con cinta de diferente color frente a cada una de las bodegas, entradas y/o salidas del proceso, para lo cual se recolecta imágenes mediante el bloque cámara y se procesa mediante el bloque detector de línea.

En el área de la mesa se han enmarcado cinco líneas, dos de color verde, dos de color rojo para la ejecución del algoritmo y el posterior reconocimiento de la posición correcta sobre la cual el Robotino debe ejecutar determinada ruta (Sánchez Basantes & Sandoval Parra, 2018).

El algoritmo se encarga de diferenciar las líneas, en caso del color verde el Robotino ubica el punto en donde se enmarcaron las líneas, toma las imágenes mediante el bloque cámara a lo largo de todo el recorrido y de forma simultánea las procesa a través del bloque detector de líneas, cambiándolas a una escala binaria, de esta forma las líneas de gamma RGB cambian a blanco y el resto de la Imagen que también está en gamma RGB cambia a color negro (Sánchez Basantes & Sandoval Parra, 2018).

Una vez identificadas las líneas, el bloque de programación ubica el primer pixel de color blanco haciendo un barrido de la Imagen de izquierda a derecha y posterior a esto se controla movimientos del Robotino necesarios para posicionar este pixel en el centro de la Imagen consiguiendo ajustar la trayectoria del robot (Sánchez Basantes & Sandoval Parra, 2018).

2.12 Establecimiento de la comunicación UDP

2.12.1 Configuración PLC

Para lograr que el PLC se comunique con el Robotino se debe programar los siguientes pasos:

Paso 1: Se establece la comunicación Ethernet.

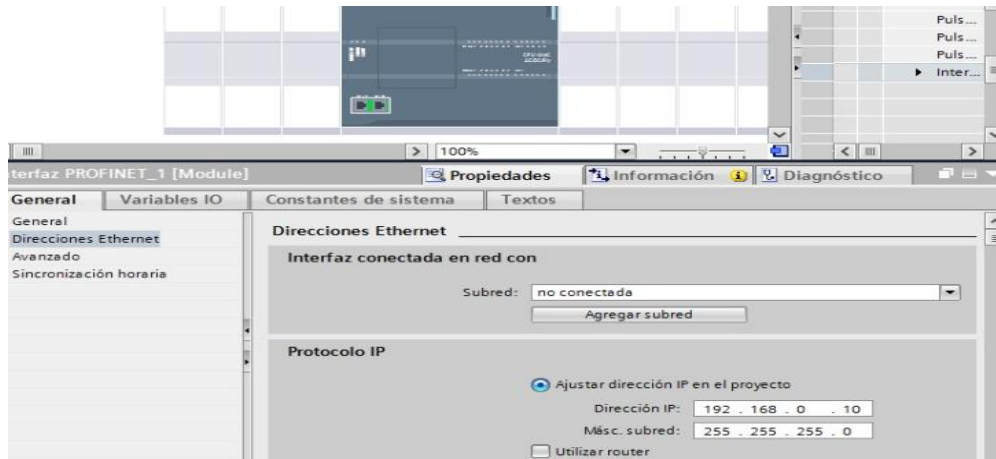


Figura 39-2: Comunicación Ethernet del PLC

Realizado por: Paucar, Alex; 2019

En la Figura 39-2 se puede observar el establecimiento de la comunicación Ethernet. Al PLC se le asigna la dirección IP: 192.168.1.35

Paso 2: Declaración de variables.

	Nombre	Tabla de variables	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	Comentario
15	SI_home4	Tabla de variables e..	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	SE_home4	Tabla de variables e..	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	SI_home3	Tabla de variables e..	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	SI_bodega1	Tabla de variables e..	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	SI_bodega2	Tabla de variables e..	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	SI_salida	Tabla de variables e..	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	LuzPiloto	Tabla de variables e..	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	Valvula1	Tabla de variables e..	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	Valvula2	Tabla de variabl...	Bool	%Q0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	M_home4	Tabla de variables e..	Bool	%M4.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	M_home3	Tabla de variables e..	Bool	%M5.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	M_home5	Tabla de variables e..	Bool	%M5.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	M_home6	Tabla de variables e..	Bool	%M5.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	<Agrega>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 40-2: Declaración de variables

Realizado por: Paucar, Alex; 2019

En la Figura 40-2 se aprecia las variables utilizadas en la programación. Dependiendo de la necesidad en cada segmento de la programación se va generando variables que pueden ser entradas (I), memorias (M) o salidas (Q). Estas variables pueden ser de tipo Bool o tipo Word.

Paso 3: Bloque TCON en el programa principal.

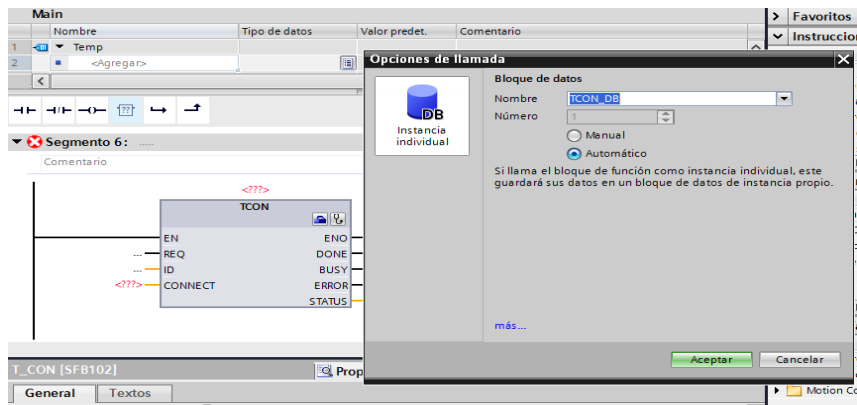


Figura 41-4: Generación de Bloque TCON en programa

Realizado por: Paucar, Alex; 2019

De la Figura 41-2 se dice: En el programa se genera un bloque TCON (bloque de conexión) que permitirá la comunicación con protocolo UDP.

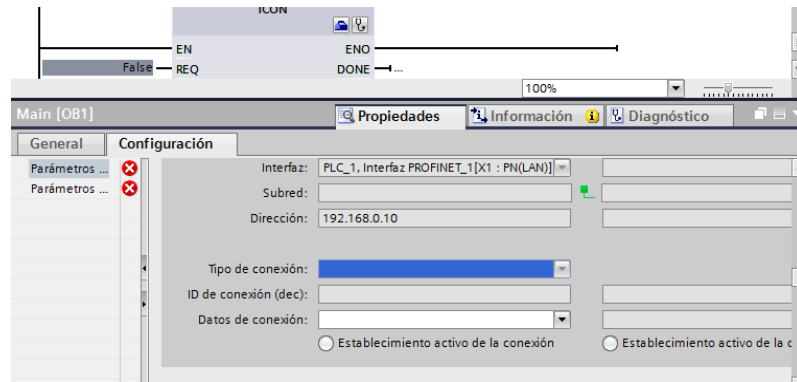


Figura 42-2: Configuración de Bloque TCON

Realizado por: Paucar, Alex; 2019

En la Figura 42-2 se detallan las configuraciones del bloque.

Tipo de conexión: UDP

Datos de Conexión: TCON_Conexión (Dice que es del bloque TCON y de tipo conexión)

Puerto: 2000 (Libre elección)

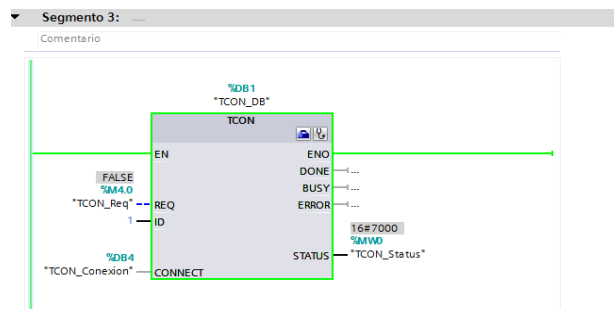


Figura 43-2: Parámetros del Bloque TCON

Realizado por: Paucar, Alex; 2019

En la Figura 43-2 se puede observar que en lado izquierdo tenemos el parámetro REQ, el cual da (TRUE) o no (FALSE) paso al funcionamiento de todo el bloque y damos paso con la memoria denominada “TCON_Req”. Al lado de la salida se tiene el parámetro STATUS el cual indica si se hizo o no la conexión y se asigna con la memoria “TCON_Status”.

Paso 4: Bloque TDISCON en el programa principal.

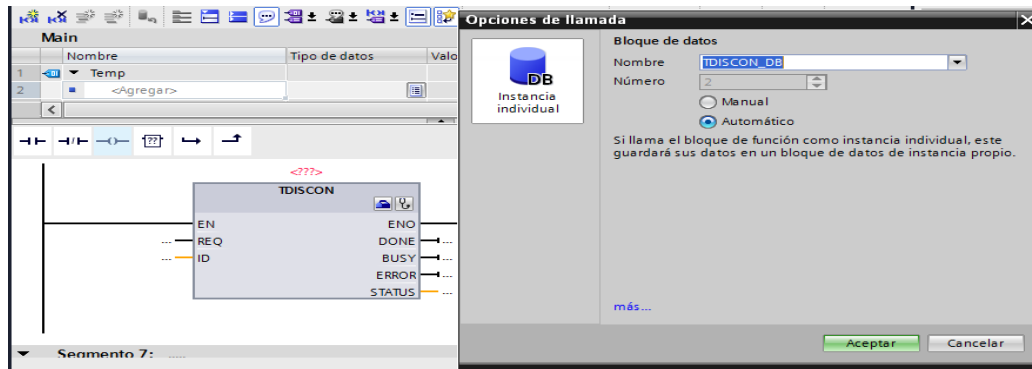


Figura 44-2: Bloque TDISCON en programa principal

Realizado por: Paucar, Alex; 2019

De la Figura 44-2 se dice: En el programa se genera un bloque TDISCON (bloque de desconexión).

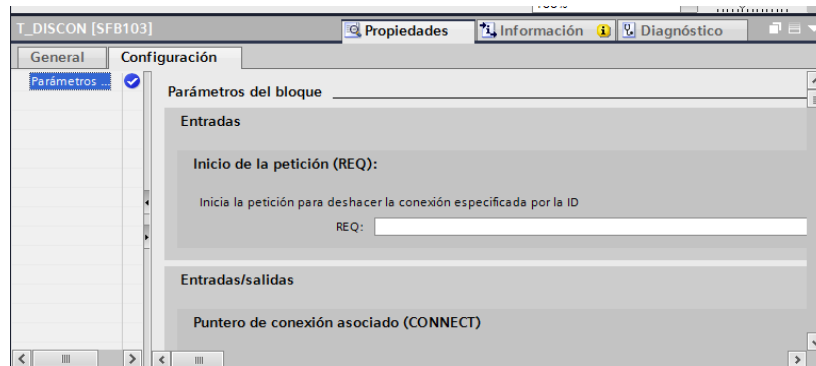


Figura 45-2: Configuración de ID de bloque TDISCON

Realizado por: Paucar, Alex; 2019

En la Figura 45-2 se detalla la configuración del bloque: Lo que es importante en ésta parte es que el ID del bloque TCON y DISCON tengan el mismo ID, en éste caso es 1.

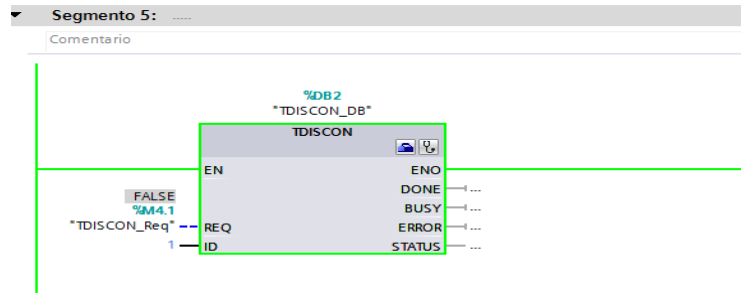


Figura 46-2: Parámetros del Bloque TDISCON

Realizado por: Paucar, Alex; 2019

En la Figura 46-2 se puede observar que en lado izquierdo tenemos el parámetro REQ, el cual da (TRUE) o no (FALSE) paso al funcionamiento de todo el bloque y se da paso con la memoria denominada “TDISCON_req”.

Paso 5: Bloque TURCV en el programa principal.

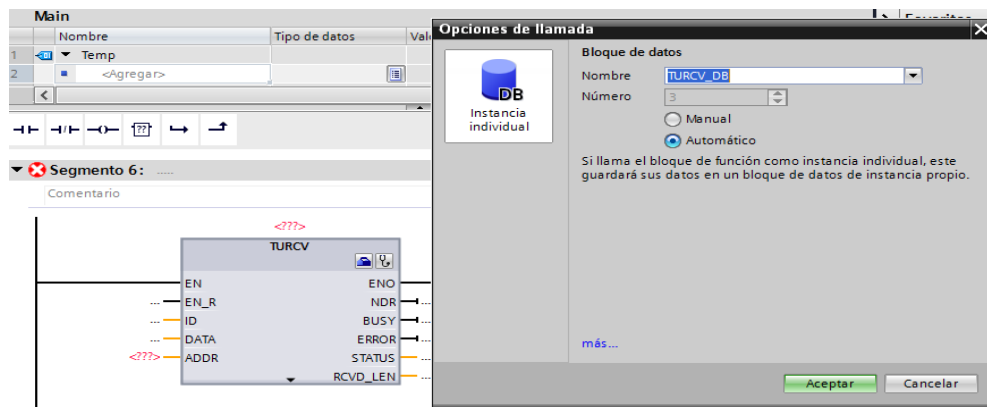


Figura 47-2: Bloque TURCV en el programa principal

Realizado por: Paucar, Alex; 2019

De la Figura 47-2 se dice: El bloque TURCV es el bloque que recibirá la comunicación UDP en caso de que algún elemento lo envíe. No olvidar que se tenga el ID 1.

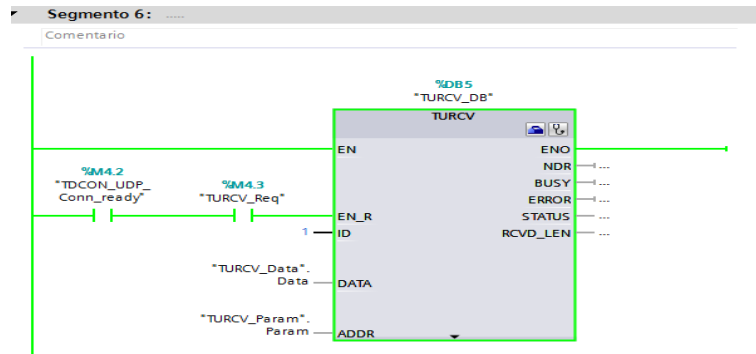


Figura 48-2: Parámetros del Bloque TURCV

Realizado por: Paucar, Alex; 2019

En la Figura 48-2 se puede observar que en lado de las entradas parámetro EN_R, el cual da (TRUE) o no (FALSE) paso al funcionamiento de todo el bloque y se da paso con la memoria “TDISCON_UDP_Conn_ready” y la memoria “TURCV_Req”.

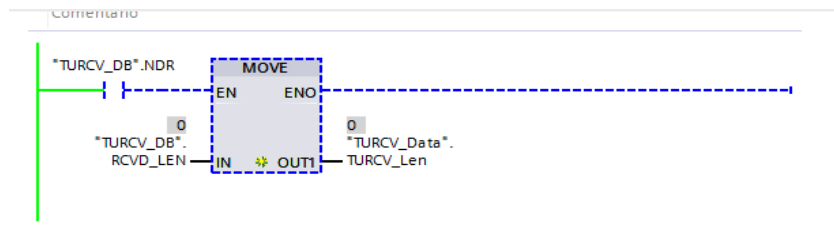


Figura 49-2: Bloque MOVE

Realizado por: Paucar, Alex; 2019

En la Figura 49-2 se tiene un bloque MOVE. El bloque MOVE se activa con el parámetro EN, el cual toma el valor del parámetro “TURCV_DB”. El valor de tipo WORD de cualquier longitud de la entrada IN se copia en la dirección de la salida OUT. El Bloque MOVE puede copiar objetos de datos que solo tengan longitudes de tipo WORD, BYTE o DWORD.

Paso 6: Bloque TUSEND en el programa principal.

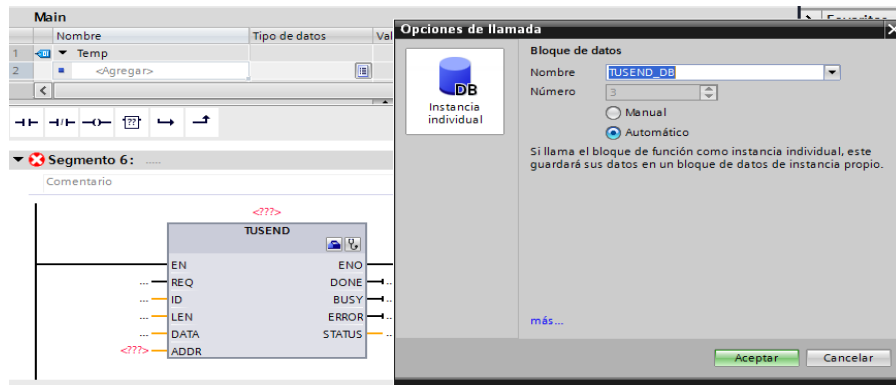


Figura 50-2: Bloque TUSEND en el programa principal

Realizado por: Paucar, Alex; 2019

De la Figura 50-2 se dice: El TUSEND es el bloque que va a enviar los paquetes de datos UDP. Debe tener ID 1.

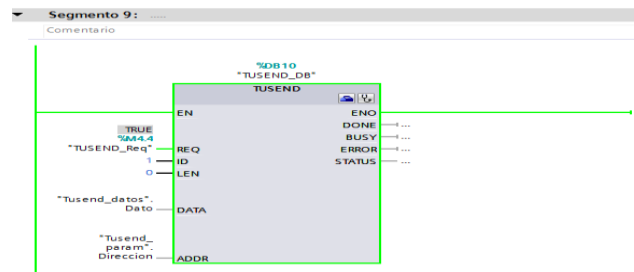


Figura 51-2: Parámetros del Bloque TUSEND1

Realizado por: Paucar, Alex; 2019

En la Figura 51-2 se puede observar que en lado de las entradas tenemos el parámetro REQ, el cual da (TRUE) o no (FALSE) paso al funcionamiento de todo el bloque y se da paso con la memoria denominada “TUSEND_Req”.

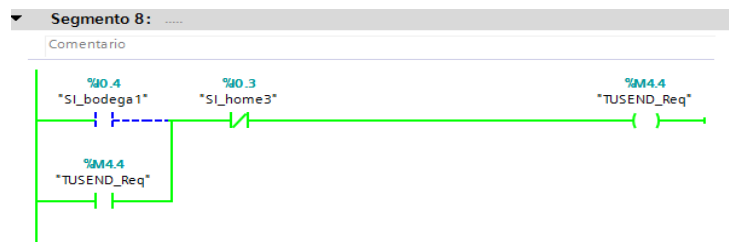


Figura 52-2: Condición para funcionamiento bloque TUSEND

Realizado por: Paucar, Alex; 2019

En la Figura 52-2 se ve cómo el bloque TUSEND se activa con la memoria “TUSEND_Req”, la cual se activa una vez que se detecta el sensor inductivo de la bodega 1 emita una señal de detección, se enclava la memoria y se desactiva solo si el sensor inductivo del Home 3 emita su señal de detección.

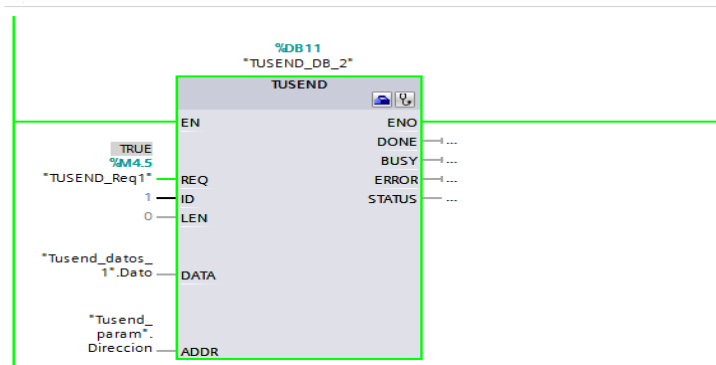


Figura 53-2: Parámetros del Bloque TUSEND2

Realizado por: Paucar, Alex; 2019

En la Figura 53-2 se puede observar que en lado de las entradas tenemos el parámetro REQ, el cual da (TRUE) o no (FALSE) paso al funcionamiento de todo el bloque y se da paso con la memoria denominada “TUSEND_Req1”.

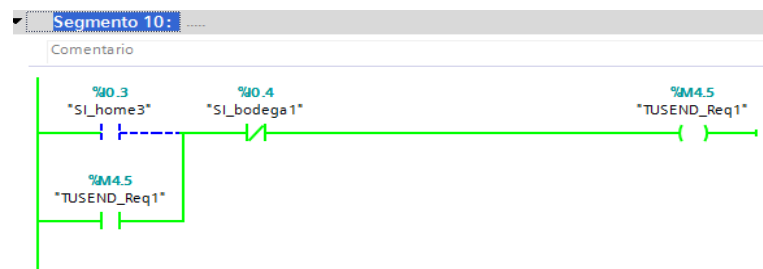


Figura 54-2: Condición para funcionamiento bloque TUSEND1

Realizado por: Paucar, Alex; 2019

En la Figura 54-2 se ve cómo el bloque TUSEND1 se activa con la memoria “TUSEND_Req1”, la cual se activa una vez que se detecta el sensor inductivo de Home 3 emita una señal de detección, se enclava la memoria y se desactiva solo si el sensor inductivo de la bodega 1 emita su señal de detección.

Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ...	Valor de a...
Static							
Dato	Array[1..5] o...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Dato[1]	Char	'D'		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Dato[2]	Char	'e'		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Dato[3]	Char	'j'		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Dato[4]	Char	'a'		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Dato[5]	Char	'r'		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 55-2: Datos enviados en bloque TUSEND

Realizado por: Paucar, Alex; 2019

En la Figura 55-2 se muestra el detalle de los datos enviados. En el bloque TUSEND se envía el dato que se desee, en éste caso se manda un array de 5 caracteres. Dicho dato tipo WORD es indispensable para su uso en Python.

Paso 7: Bloque condicionante en el programa principal.

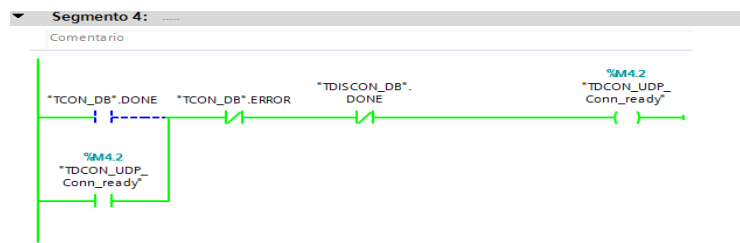


Figura 56-2: Bloque condicionante para comunicación UDP

Realizado por: Paucar, Alex; 2019

En la Figura 56-2 se observa un segmento de condición. Realizamos segmento con una condicionante IF. Si el TCON está finalizado y el TCON no da error y el TDISCON está desconectado entonces se almacena una memoria LISTO, la cual dará paso a la comunicación UDP.

Paso 8: Direccionamiento de dispositivo al cual enviar datos.

Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ...	Valor de a...	Comentario
Static								
Direccion	TADDR_Param			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
REM_IP_ADDR	Array[1..4] of USInt			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote station address
REM_IP_ADDR[1]	USInt	192		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote station address
REM_IP_ADDR[2]	USInt	168		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote station address
REM_IP_ADDR[3]	USInt	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote station address
REM_IP_ADDR[4]	USInt	45		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote station address
REM_PORT_NR	UInt	2000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote port number
RESERVED	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		unused; has to be 0

Figura 57-2: Asignación de dirección Robotino

Realizado por: Paucar, Alex; 2019

De la Figura 57-2 se observa la asignación de dirección de envío de datos UDP. En el Tusend_param se pone la dirección a donde se quiere enviar el dato, en este caso es la dirección del Robotino: 192.168.1.45 (ip de la computadora) por el puerto 2000.

2.12.2 Configuración Robotino

Con el dispositivo de intercambio de datos UDP, pueden intercambiarse datos entre Robotino View y aplicaciones externas a través de UDP (Festo Didactic GmbH & Co. KG, 2010).

Para realizar la comunicación de Robotino con el PLC con protocolo UDP se debe tener en cuenta algunos aspectos y seguir los siguientes pasos:

Protocolo

Especificación de la estructura de los datos

Byte	Función
0	Message ID
1-2	Número de bytes de todo el mensaje N. El tipo es UINT16 ^[172]
3	Checksum (a inicializar con 0 cuando se genera el paquete, véase Suma de prueba ^[174])
N-1	Último byte del mensaje

Figura 58-2: Especificaciones de estructura de datos

Realizado por: Paucar, Alex; 2019

En la Figura 58-2 se observa especificaciones de datos UDP. Es la estructura de los datos que recibe el PLC, cuya estructura debe ser cumplida exactamente ya que caso contrario no hay una lectura de datos UDP.

Paso 1: Librería intercambio de datos UDP

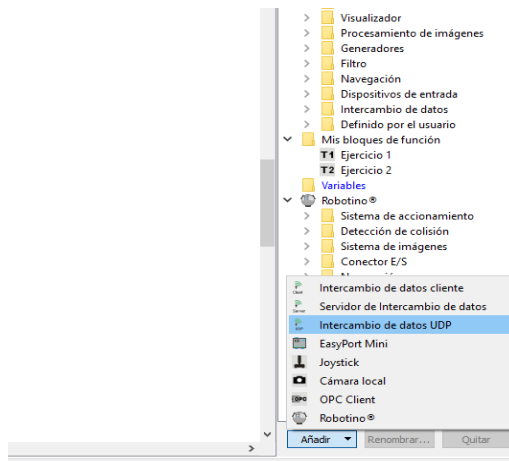


Figura 59-2: Librería de comunicación UDP en Robotino View

Realizado por: Paucar, Alex; 2019

En la Figura 59-2 se observa la librería UDP en el software. La comunicación UDP del Robotino no está presente en las librerías normales, son bloques especiales que se debe poner desde añadir y elegimos intercambio de datos UDP. Mensajes UDP

Los bloques de función en la categoría Mensaje 0 permiten el envío y la recepción de datos (Festo Didactic GmbH & Co. KG, 2010).

Entrada y salidas



Figura 60-5: Asignación de valores a entradas y salidas UDP

Realizado por: Paucar, Alex; 2019

En la Figura 60-2 se observa que mediante la generación de una constante podemos asignar valores a las entradas y consecuentemente a las salidas de la comunicación UDP.

Paso 2: Configuración de datos UDP.

En la ventana de diálogo pueden configurarse los datagramas de envío y recepción UDP:

Con "Puerto del servidor" se configura el número de puerto UDP en el cual escucha datagramas el servidor y desde el cual se envían.

Con "Iniciar servidor" el servidor empieza a escuchar. Una vez se ha puesto en marcha el servidor, los paquetes de datos UDP se reciben, se interpretan y se envían.

El "intervalo" es el intervalo de tiempo que debe transcurrir tras una transmisión hasta que se permita la siguiente transmisión.

Para cada mensaje (mensaje 0 ó mensaje 1) el envío puede activarse (on) o desactivarse (off) individualmente.

Las direcciones IP y puertos de los receptores de datos pueden introducirse en la tabla "Receptores"

(Listeners). Si no se especifica un puerto, se utilizará el 9180 de forma predeterminada.

(Festo Didactic GmbH & Co. KG, 2010)

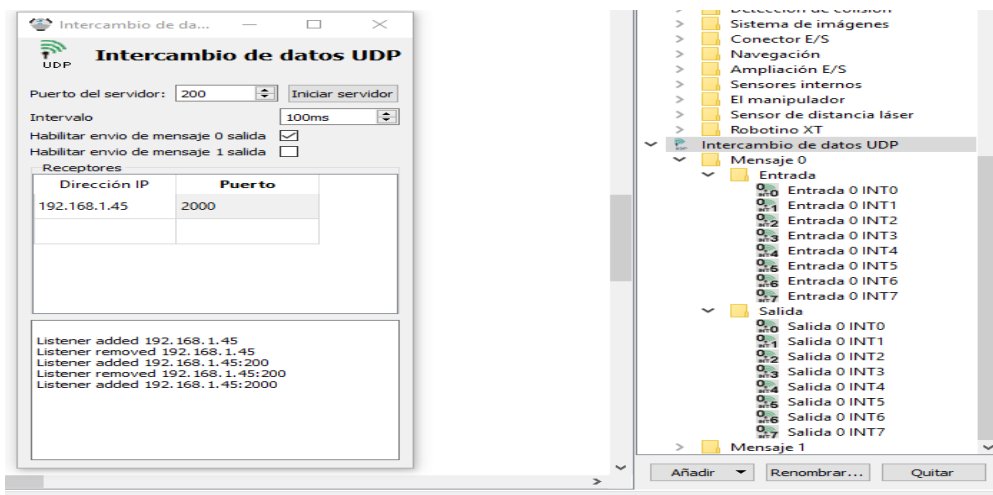


Figura 61-2: Cuadro de diálogo para configurar comunicación UDP

Realizado por: Paucar, Alex; 2019

Como se muestra en la Figura 61-2 se da doble clic sobre "Intercambio de datos UDP" y sale el cuadro de diálogo en donde se coloca el número de puerto 2000. En la parte de dirección IP se pone la dirección del PLC 192.168.1.45 por el Puerto 2000.

Es necesario resaltar que toda comunicación UDP tiene un servidor escuchador y un cliente transmisor.

Paso 3: Generación de salida y entrada

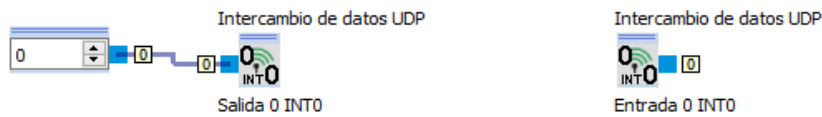


Figura 62-2: Salidas y entradas generadas para comunicación UDP

Realizado por: Paucar, Alex; 2019

De la Figura 62-2 se observa cómo se genera la entrada y la salida, para comprobar si se comunica generamos una constante y se da valores al a salida, si hay comunicación se podrá visualizar el mismo valor tanto en la salida como en la entrada.

2.12.3 Codificación en Python

```

Archivo  Editar  Vista  Ejecutar  Device  Herramientas  Ayuda
ServidorUdp.py
1 import socket
2
3
4 # Datos de Inicio
5 IpLocal      = "192.168.1.45"
6 IpMV        = ('192.168.1.5', 2000)
7
8 PuertoLocal  = 2000
9
10 bufferSize  = 1024
11
12 # Datos para enviar a Robotino
13 RobotinoDejar = b'\x00$\x00\xd6\x05\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
14 RobotinoLlevar = b'\x00$\x00\xcc\x0f\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
15 RobotinoParar = b'\x00$\x00\xdb\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
16 #Variables de prueba
17 Variable1     = b'\x00$\x00s\n\x00\x00\x00\x14\x00\x00\x00\x1e\x00\x00\x00(\x00\x00\x002\x00'
18 Variable2     = b'\x00$\x00\t\n\x00\x00\x00\xc8\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
19
20 # Crear el socket para los datagramas
21 UDPSocketServidor = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
22

```

Figura 63-2: Código Python

Realizado por: Paucar, Alex; 2019

En la Figura 63-2 se observa que se utilizó lenguaje de programación Python para interpretar palabras como números debido a que el PLC solo envía palabras y el Robotino solo acepta números. Así en Python se debe tomar en cuenta los siguientes aspectos:

- En “IPLocal” ponemos la dirección de la computadora

- En “IpMV” ponemos la dirección de hacia donde se quiere enviar el dato y el número de puerto, el cual se utilizó en los bloques generados en el PLC. Ésta dirección es la del Robotino.
- En la parte donde dice datos para enviar a Robotino se debe hacer una comparación de las palabras con un número. Almacenamos la palabra que se envió en el bloque TUSEND desde el PLC en un dato tipo byte “b” y asignamos un valor con su respectivo código.
- Una vez obtenido el dato como número se trabaja a manera de una condición IF, que si es ese dato realice un proceso y que si no lo es realice otro proceso.

2.12.4 Establecimiento comunicación UDP PLC – Robotino

Paso 1: Se compila y carga el código al PLC y se establece la comunicación online, se compila el código y se carga en el PLC.

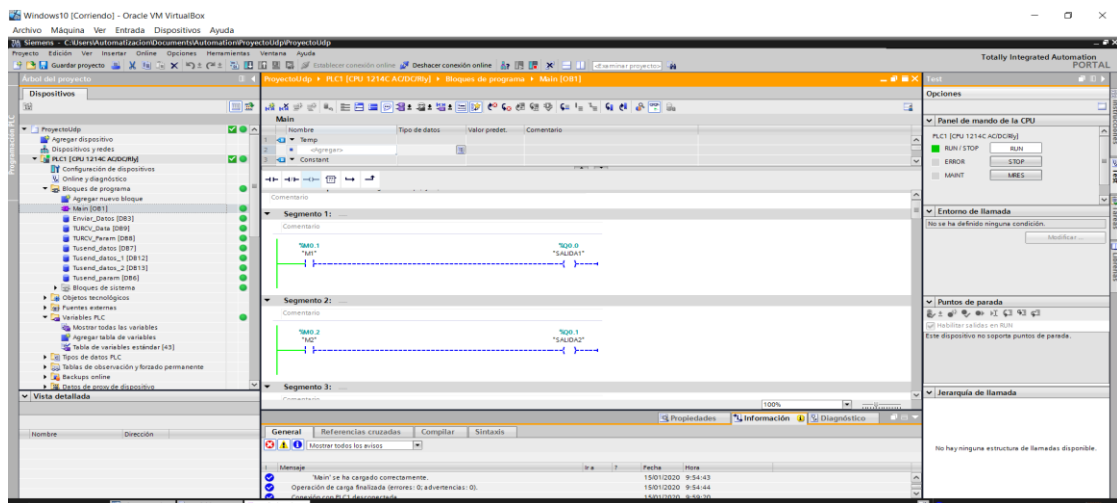


Figura 64-2: Activación de modo prueba

Realizado por: Paucar, Alex; 2019

En la Figura 64-2 se tiene la barra de herramientas en la parte posterior tenemos una figura pequeña a manera de lentes, se da clic sobre ella y el programa se torna de color tomate, lo que permite ver el funcionamiento de cada segmento en tiempo real. Además, en la parte izquierda se puede ver unos indicadores de color verde que nos indican la correcta configuración de cada bloque.

Paso 2: Se conecta vía online el Robotino con PLC.

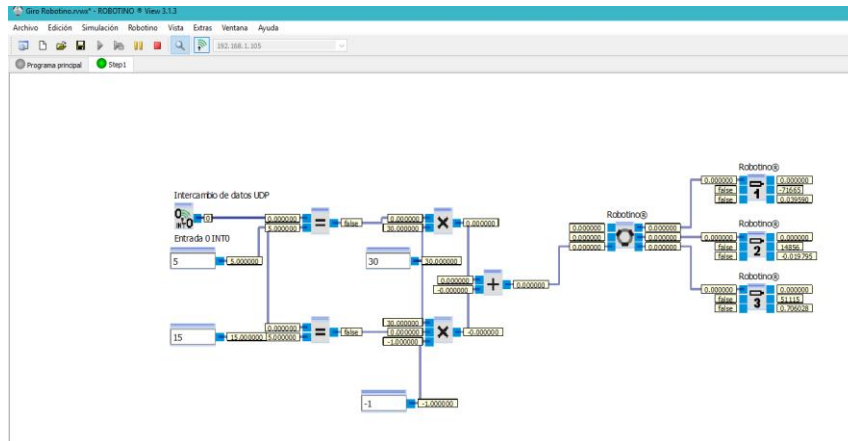


Figura 65-2: Conexión online con Robotino

Realizado por: Paucar, Alex; 2019

En la Figura 65-2 en la parte superior de la ventana se tiene la opción de ingresar la IP con la cual se desea conectar, en nuestro caso se coloca la IP de Robotino 192.168.1.105 y se activa la comunicación wifi.

Paso 3: Se verifica inicio de comunicación.

Utilizando servido Python se verifica si la comunicación inició, al correr el programa sale un mensaje que dice servidor levantado y escuchado.

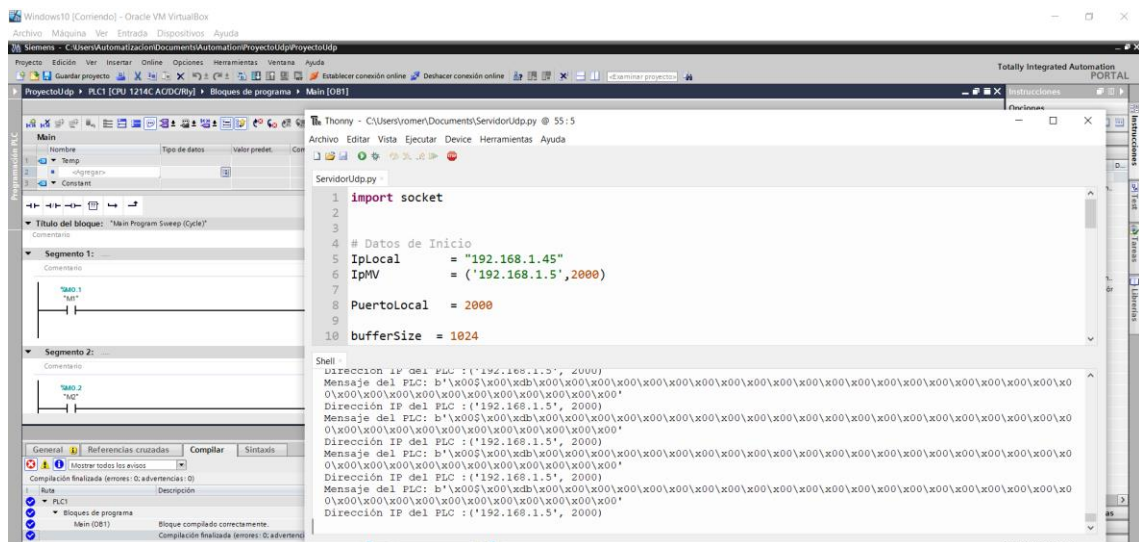


Figura 66-2: Comprobación de intercambio mensaje Robotino-PLC

Realizado por: Paucar, Alex; 2019

Como se observa en la Figura 66-2 en Python aparece el código del mensaje que se envió, lo que permite comprobar la correcta comunicación UDP, dependiendo del dato que se envíe realice una acción determinada.

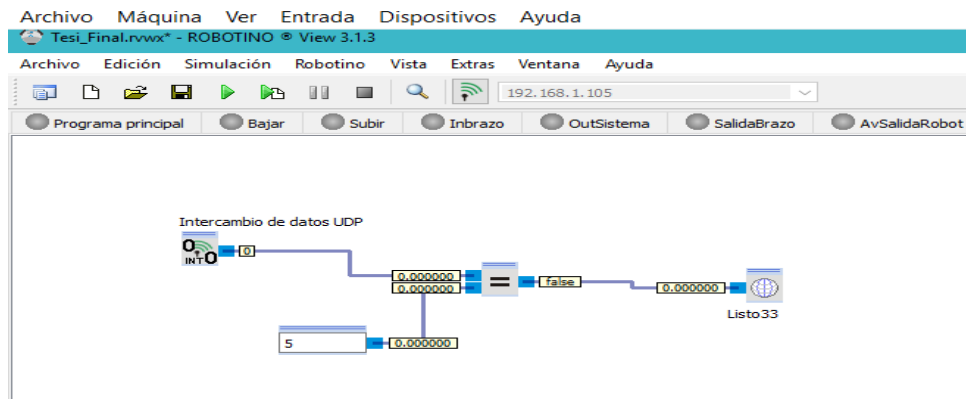


Figura 67-2: Comprobación de intercambio mensaje Robotino-PLC

Realizado por: Paucar, Alex; 2019

Como se observa en la Figura 67-2 en Robotino View se programó la comunicación UDP, recibiendo el dato y comparándolo con un valor de 5. El valor de la comparación se almacena en una variable, si es verdadero es 1 y si es falso es 0. Para comprobar si se está realizando la comunicación UDP se lo puede hacer desde Robotino View, al colocar el sensor sobre bodega 1 o Home 3 el valor de la comparación que se explicó anteriormente debería toar 1, si es verdadero da paso al siguiente bloque de programación y caso contrario se queda en el sub programa actual.

2.13 Diseño y Programación de la HMI

La HMI tiene como objetivo monitorear el proceso de comunicación PLC Robotino, para lo cual se utilizó Labview.

Para la comunicación se utilizó el OPC server, el mismo que permite compartir variables y establecer la comunicación LabVIEW- Tía portal a través de un canal. Se forman bloques: En el primer bloque se tiene 2 botones que nos dan la opción de activar o desactivar la comunicación y su respectivo indicador que se ponen color verde y rojo respectivamente. El segundo bloque corresponde a la recepción UDP, se tiene el botón para activar la recepción y su respectivo indicador. El tercer bloque indica si está activo o desactivo el sensor inductivo de la bodega 1 o el de home 4.

Paso 1: Se verifica la correcta comunicación de variables HMI-Tía Portal.

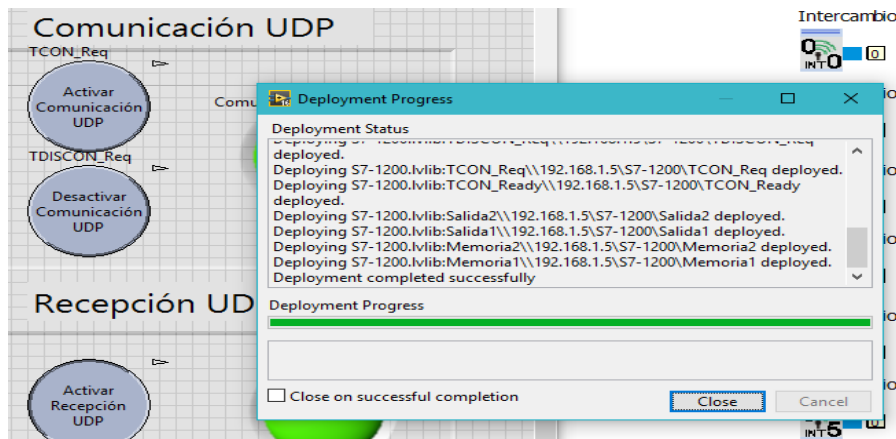


Figura 68-2: Inicio de programa HMI y conexión online PLC

Realizado por: Paucar, Alex; 2019

Como se observa en la Figura 68-2 para verificar si la HMI esta funcionando antes de abrir el programa damos clic derecho sobre HMI y ponemos “Deploy All”, si esta todo correcto sale un mensaje de aceptación y da paso.

Paso 2: Se activa comunicación UDP desde HMI.

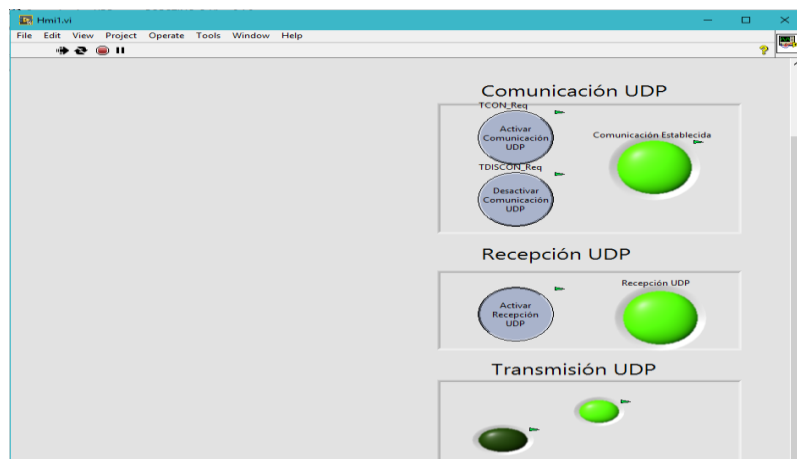


Figura 69-2: Activación envío y recepción de datos desde HMI

Realizado por: Paucar, Alex; 2019

Como se observa en la Figura 69-2 se manda a correr el programa y se da clic sobre activar envío y recepción de datos UDP. Mediante los indicadores se puede ver si está apagada o prendida la comunicación, color verde On y color rojo OFF. Además, se puede observar que sensor es el que está enviando la señal.

2.14 Funcionamiento de sistema

2.14.1 Diagrama de Funcionamiento Robotino

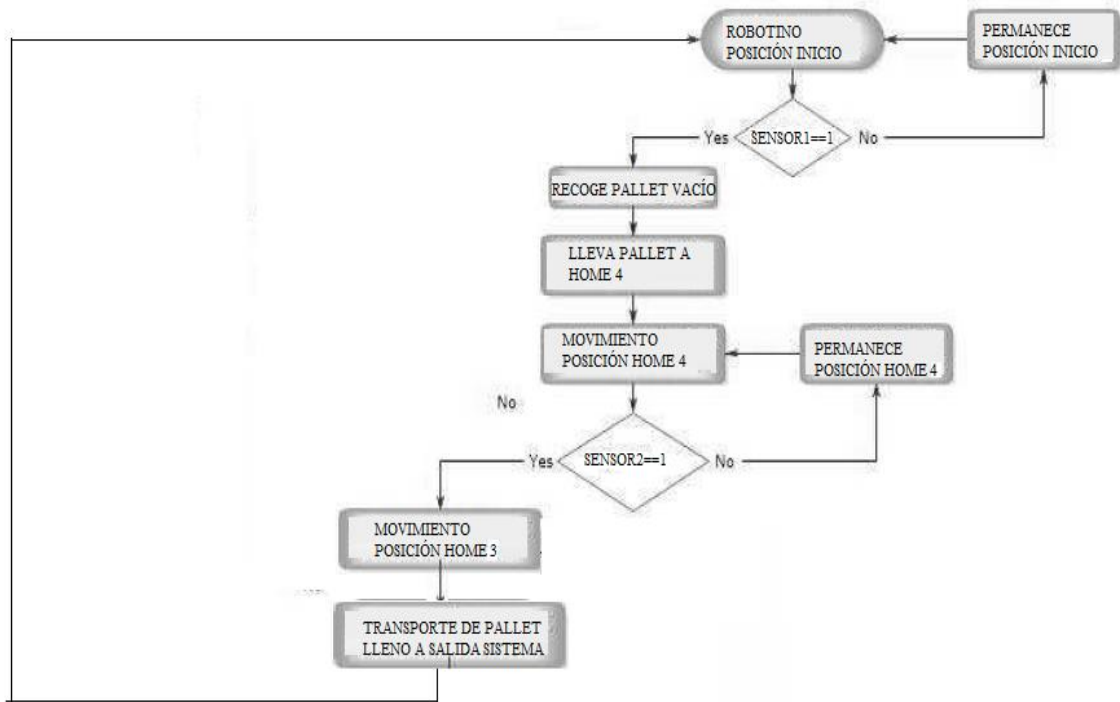


Gráfico 4-2: Diagrama de bloques del funcionamiento de Robotino

Realizado por: Paucar, Alex; 2019

En el Gráfico 4-2 se puede apreciar el modo en que se programó el Robotino. Empieza su secuencia en una posición inicio, si el sensor 1 detecta la presencia del pallet el robot procede a recogerlo en la bodega 1 o 2 y llevarlo a Home 4 en donde nuevamente se deja el pallet para que el Brazo Robótico realice la paletización. En la segunda parte nuevamente se tiene una condición IF, si el sensor 2 detecta el pallet en Home 3 el Robotino se mueve hasta dicha posición, lo recoge y lo lleva a la salida donde finaliza el proceso.

2.14.2 Diagrama de funcionamiento Brazo Robótico

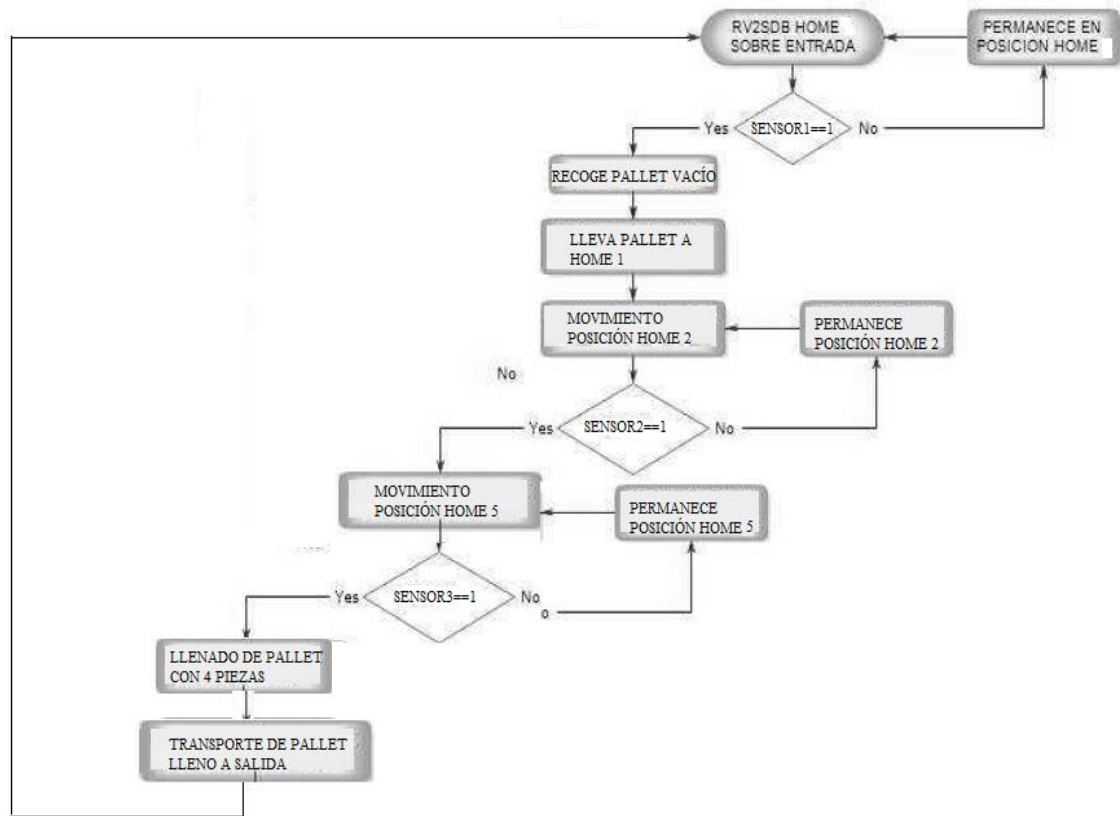


Gráfico 5-2: Diagrama de bloques del funcionamiento Brazo Robótico

Realizado por: Paucar, Alex; 2019

En el Gráfico 5-2 se observa el modo en que se programó el Brazo Robotino. Empieza su secuencia en una posición Home de inicio, si el sensor 1 detecta la presencia del pallet en Home 4 el robot procede a recogerlo en la bodega y llevarlo a Home 1 o 2. Inicia la segunda parte del proceso, se tiene un sensor 2 que da paso a que el robot se mueve a la Posición Home 5. En la tercera parte del proceso se tiene nuevamente una condición IF en la cual un sensor 3 da paso a transporte de pieza de Home 5 a Home 1 o 2, una vez lleno el pallet el robot lo lleva a Home 3 y da por terminada s secuencia.

2.14.3 Secuencia

Para comprensión del proceso general se presentan imágenes de cada una de los elementos y áreas de trabajo que constituyen el sistema:

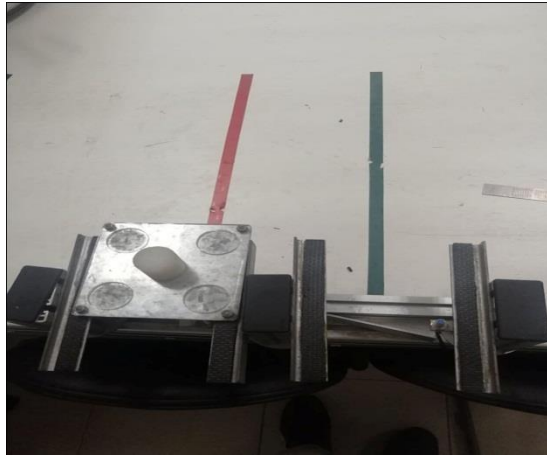


Figura 70-2: Bodega 1 Bodega 2

Realizado por: Paucar, Alex; 2019

En la Figura 70-2 se observa las bodegas 1 y 2 del sistema, donde se colocan los pallets vacíos.

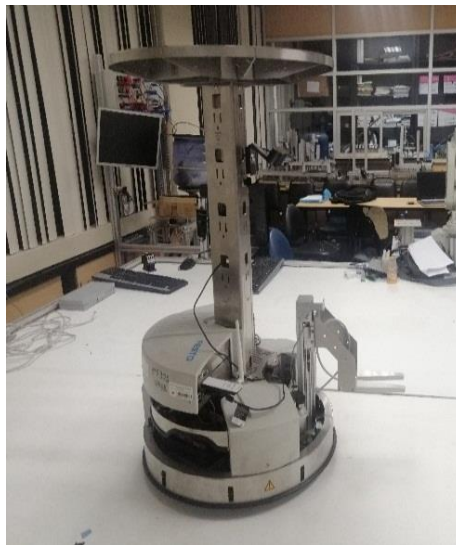


Figura 71-2: Robotino PLus

Realizado por: Paucar, Alex; 2019

En la Figura 71-2 se observa el robot Robotino.



Figura 72-2: Brazo Robótico

Realizado por: Paucar, Alex; 2019

En la Figura 72-2 se observa el Brazo Robótico.

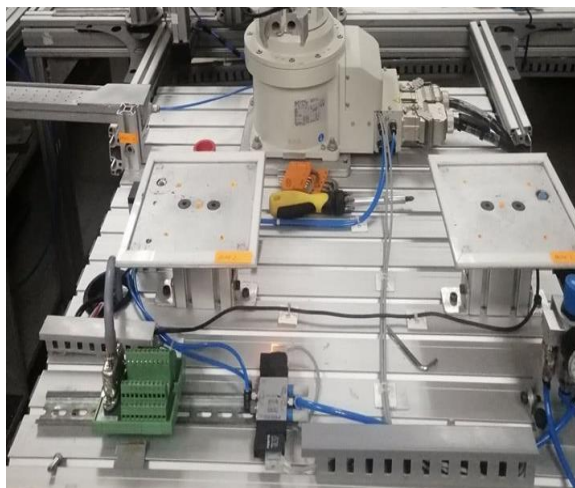


Figura 73-2: Home 1 y Home 2

Realizado por: Paucar, Alex; 2019

En la Figura 73-2 la ubicación de Home 1 y Home 2.



Figura 74-2: Home 3 y Home 4

Realizado por: Paucar, Alex; 2019

En la Figura 74-2 se observa la ubicación de Home 3 y Home 4.

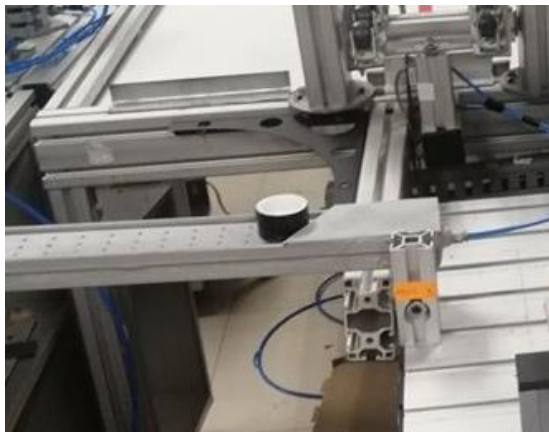


Figura 75-2: Home 5

Realizado por: Paucar, Alex; 2019

En la Figura 75-2 se observa la ubicación de Home 5.



Figura 76-2: Salida de sistema

Realizado por: Paucar, Alex; 2019

En la Figura 76-2 se observa la salida del sistema.



Figura 77-2: Tablero del Control

Realizado por: Paucar, Alex; 2019

En la Figura 77-2 se observa la ubicación del PLC en el Tablero de Control.

A Continuación presentamos un gráfico donde se detalla el funcionamiento del proceso:

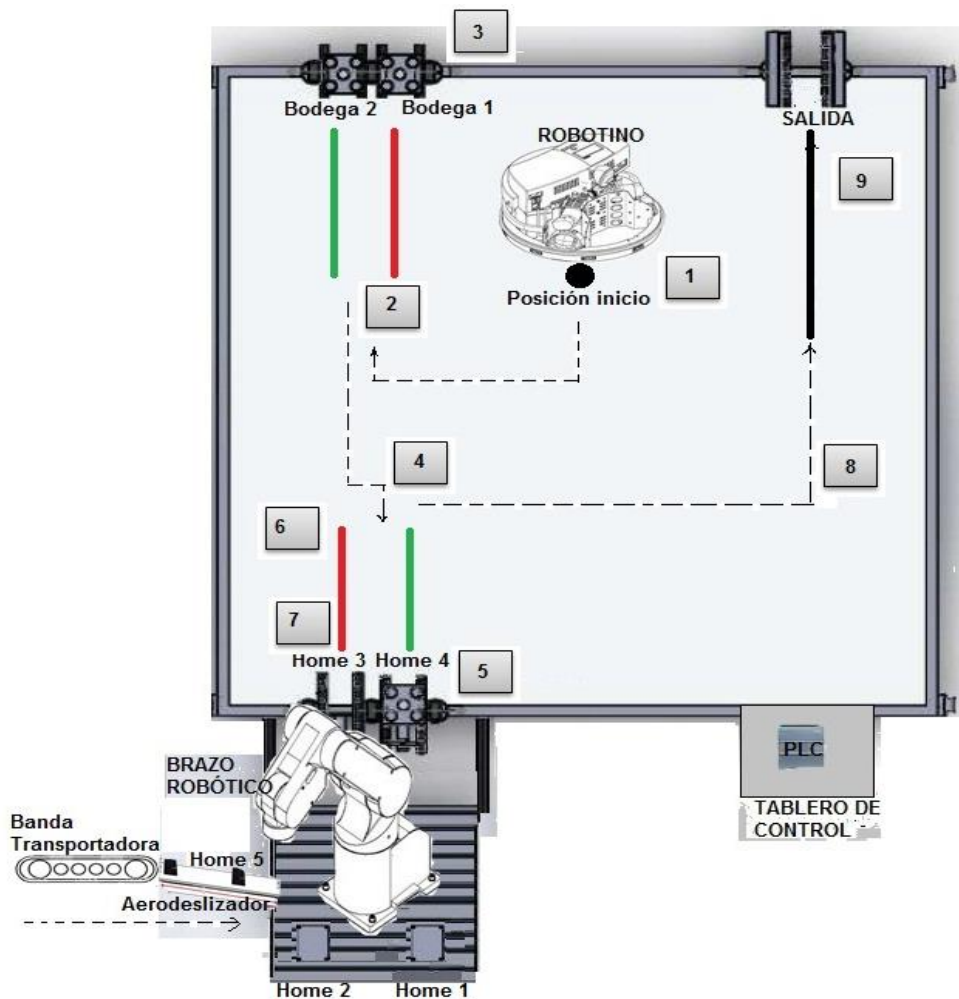


Gráfico 6-2: Funcionamiento total de sistema

Realizado por: Paucar, Alex; 2019

En el Gráfico 6-2 se visualiza el funcionamiento total del sistema. El Robotino inicia el proceso desde una posición de inicio (ubicación 1), posición que debe ser exacta ya el Robotino trabaja con distancias expresadas en milímetros. Una vez iniciado el programa, el Robotino se mueve y gira a una posición establecida (ubicación 2), desde ese punto el Robot identifica el color de línea (roja o negra) y la sigue hasta llegar a la ubicación del pallet (ubicación 3) donde utiliza uno de sus sensores de luz infrarroja para evitar chocarse y activa mueve el montacargas para recoger el pallet. Luego el robot va a otra posición (ubicación 4), una vez ahí nuevamente diferencia color y se dirige a dejar el pallet en Home 4 (ubicación 5) donde utiliza el sensor de distancia para no chocar e igualmente baja el montacargas para bajar el pallet.

Inicia la secuencia del Brazo Robótico, el pallet inmediatamente es detectado por un sensor y da la señal de activación de válvula para mover el pallet y además da la señal de inicio del proceso del robot. Pasa a dejar el pallet en el Home 2 o Home 1 donde se activa un sensor que esta comunicado con el Brazo Robótico mediante la Caja de Interfaz y da pasó a secuencia del

mismo, se colocan las piezas manualmente sobre una banda transportadora cuyo movimiento se debe a un pequeño motor que se encuentra adaptado en un extremo, de éste modo lleva la pieza hasta el inicio del aerodeslizador, el mismo que se encarga de llevar la pieza hasta la posición Home 5, en donde el Brazo Robótico la detecta (Bit 900) y lleva cada una de las 4 piezas y a cada una de las sub posiciones del pallet ubicado sobre el Home 1 o 2 para luego transportar el pallet lleno a la salida del Robot o Home 3.

Inicia la última parte del proceso, al momento de que el pallet se encuentra en Home 3 se activa la señal de un sensor que es enviada al Robotino para continuar con el proceso va de ubicación 6 a ubicación 7, recoge el pallet va a ubicación 8 y finalmente deja el pallet en la salida del sistema (ubicación 9).

CAPÍTULO III

3. MARCO METODOLÓGICO

3.1 Introducción

Las pruebas ejecutadas a nuestro proceso están orientadas a comprobar la comunicación UDP entre los dispositivos y a determinar la eficiencia del sistema en general, así como de cada uno de los procesos involucrados dentro del mismo.

3.2 Prueba de comunicación UDP

Tabla 1-3: Pruebas sometidas a comunicación UDP

Número de prueba	Dato enviado	Confirmación de dato enviado	Indicador
1	Dato 1	-----	Robotino no gira a la derecha
2	Dato 2	“Servidor UDP levantado y escuchado”	Robotino gira a la izquierda
3	Dato 3	“Servidor UDP levantado y escuchado”	Robotino se detiene
4	Dato 1	“Servidor UDP levantado y escuchado”	Robotino gira a la derecha
5	Dato 2	-----	Robotino no gira a la izquierda
6	Dato 3	“Servidor UDP levantado y escuchado”	Robotino se detiene
7	Dato 1	“Servidor UDP levantado y escuchado”	Robotino gira a la derecha
8	Dato 2	“Servidor UDP levantado y escuchado”	Robotino gira a la izquierda
9	Dato 3	“Servidor UDP levantado y escuchado”	Robotino se detiene
10	Dato 1	“Servidor UDP levantado y escuchado”	Robotino gira a la derecha
11	Dato 2	“Servidor UDP levantado y escuchado”	Robotino gira a la izquierda
12	Dato 3	“Servidor UDP levantado y escuchado”	Robotino se detiene
13	Dato 1	“Servidor UDP levantado y escuchado”	Robotino gira a la derecha
14	Dato 2	“Servidor UDP levantado y escuchado”	Robotino gira a la izquierda
15	Dato 3	“Servidor UDP levantado y escuchado”	Robotino se detiene
16	Dato 1	“Servidor UDP levantado y escuchado”	Robotino gira a la derecha
17	Dato 2	-----	Robotino no gira a la izquierda
18	Dato 3	“Servidor UDP levantado y escuchado”	Robotino se detiene
19	Dato 1	“Servidor UDP levantado y escuchado”	Robotino gira a la derecha
20	Dato 2	“Servidor UDP levantado y escuchado”	Robotino gira a la izquierda
21	Dato 3	“Servidor UDP levantado y escuchado”	Robotino se detiene
C. Correctos/Fallidos/TOTAL			18/3/21

Realizado por: Paucar, Alex; 2019

En la Tabla 1-3 se puede ver las pruebas que se realizaron para comprobar si estableció o no la comunicación UDP entre el PLC y el Robotino. Se tiene 3 tipos de datos que dan una orden diferente al Robotino, con el Dato1 el robot gira a la derecha, con el Dato 2 el robot gira a la izquierda y con el Dato 3 el robot se detiene.

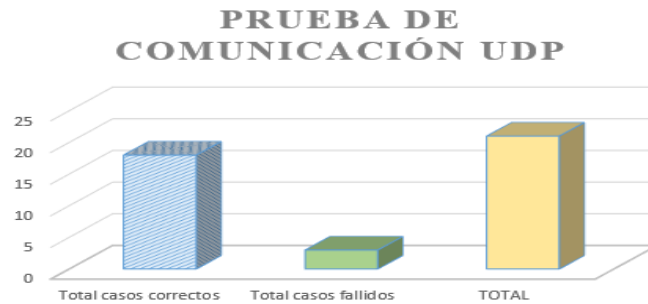


Gráfico 1-3: Prueba comunicación UDP

Realizado por: Paucar, Alex; 2019

En Gráfico 1-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 21 pruebas, se enviaba desde el PLC un dato y si había comunicación el Robotino ejecutaba la tarea, de las 21 pruebas 7 eran para el Dato1, 7 para el Dato 2 y 7 para el Dato 3. Se obtuvo un total de 18 casos correctos y 3 casos fallidos, los cuales se debieron a que el PLC y el robot no estaban configurados correctamente o a veces el Robotino se encontraba sin batería. Expresando en términos de porcentajes tenemos un 14% de fracaso frente a 86% de éxito, éste último valor nos refleja el éxito en la comunicación UDP.

3.3 Prueba de comunicación Caja de Interfaz con Brazo Robótico

Tabla 2-3: Pruebas sometidas a comunicación Caja Interfaz- Brazo Robótico

Número de prueba	Ubicación de sensor	Entrada de terminal Syslink a la cual se conecta la señal del sensor	Indicador
1	Home 1	Entrada 1	Prende led entrada 1 de Caja de Interfaz
2	Home 2	Entrada 2	No prende led entrada 2 de Caja de Interfaz
3	Home 1	Entrada 1	Prende led entrada 1 de Caja de Interfaz
4	Home 2	Entrada 2	Prende led entrada 2 de Caja de Interfaz

Continúa

Continúa

5	Home 1	Entrada 1	Prende led entrada 1 de Caja de Interfaz
6	Home 2	Entrada 2	Prende led entrada 2 de Caja de Interfaz
7	Home 1	Entrada 1	Prende led entrada 1 de Caja de Interfaz
8	Home 2	Entrada 2	Prende led entrada 2 de Caja de Interfaz
9	Home 1	Entrada 1	Prende led entrada 1 de Caja de Interfaz
10	Home 2	Entrada 2	Prende led entrada 2 de Caja de Interfaz
		Total casos correctos	9
		Total casos fallidos	1
		TOTAL	10

Realizado por: Paucar, Alex; 2019

En la Tabla 2-3 se puede ver las pruebas que se realizaron para comprobar si estableció o no la comunicación entre la Caja de Interfaz y el Brazo Robótico. Se tiene dos sensores ubicados en el Home 1 y Home 2, cada sensor se conectó mediante un terminal Syslink a las entradas de la Caja de Interfaz. Se realizaron varias pruebas para verificar la comunicación entre los dispositivos. Si el sensor detecta el pallet se prende un led en la Caja de Interfaz que nos indica si hay comunicación.

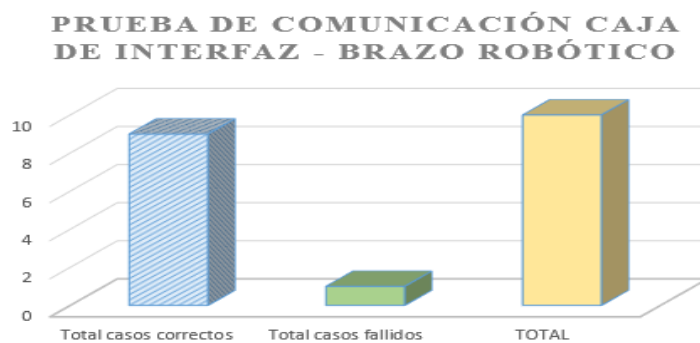


Gráfico 2-3: Prueba comunicación Caja de interfaz - Brazo Robótico

Realizado por: Paucar, Alex; 2019

En Gráfico 2-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 10 pruebas, se conectaba y se enviaba señales desde el sensor; si había comunicación el led de determinada entrada de la Caja de Interfaz se prendía., de las 10 pruebas 9. Se obtuvo un total de 9 casos correctos y 1 casos fallidos, el cual se debió a la mala conexión realizada ya sea del sensor o de la caja. Expresando en términos de porcentajes tenemos un 10% de fracaso frente a 90% de éxito, éste último valor nos refleja el éxito en la comunicación entre los dispositivos.

3.4 Prueba de detección piezas con bit 900

Tabla 3-1: Pruebas detección de pieza Bit 900

Número de prueba	Acción programada	Número de Bit utilizado	Indicador
1	Detectar pieza	900	Realiza proceso no cuando detecta
2	Detectar pieza	900	Realiza proceso cuando detecta
3	Detectar pieza	900	Realiza proceso no cuando detecta
4	Detectar pieza	900	Realiza proceso cuando detecta
5	Detectar pieza	900	Realiza proceso cuando detecta
6	Detectar pieza	900	Realiza proceso cuando detecta
7	Detectar pieza	900	Realiza proceso cuando detecta
8	Detectar pieza	900	Realiza proceso cuando detecta
9	Detectar pieza	900	Realiza proceso cuando detecta
10	Detectar pieza	900	Realiza proceso cuando detecta
11	Detectar pieza	900	Realiza proceso cuando detecta
12	Detectar pieza	900	Realiza proceso cuando detecta
13	Detectar pieza	900	Realiza proceso cuando detecta
14	Detectar pieza	900	Realiza proceso cuando detecta
15	Detectar pieza	900	Realiza proceso cuando detecta
16	Detectar pieza	900	Realiza proceso cuando detecta
17	Detectar pieza	900	Realiza proceso cuando detecta
18	Detectar pieza	900	Realiza proceso cuando detecta
19	Detectar pieza	900	Realiza proceso cuando detecta
20	Detectar pieza	900	Realiza proceso cuando detecta
Total casos correctos			18
Total casos fallidos			2
TOTAL			20

Realizado por: Paucar, Alex; 2019

En la Tabla 3-3 se puede ver las pruebas que se realizaron para comprobar si el Brazo Robótico detectaba piezas con el sensor Bit 900. Se realizó un programa con una condicionante IF, si el sensor detectaba se ejecutaba una determinada secuencia y si no detectaba la pieza se ejecutaba otra secuencia.

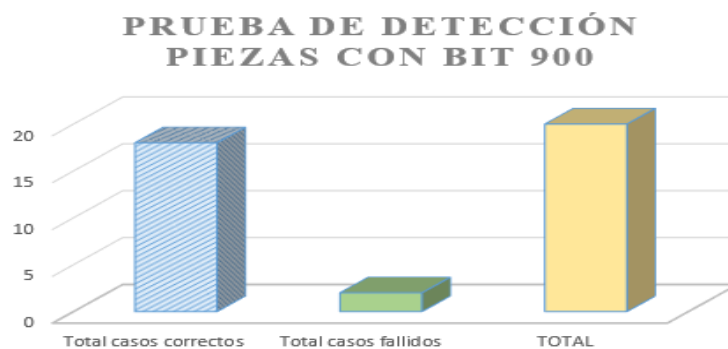


Gráfico 3-1: Prueba Detección de piezas Bit 900

Realizado por: Paucar, Alex; 2019

En Gráfico 3-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 20 pruebas, se enviaba ejecutaba una secuencia si la pieza estaba en el lugar u otra secuencia en el caso de que la pieza no este. De las 20 pruebas, se obtuvo un total de 18 casos correctos y 2 casos fallidos, los cuales se ocurrieron por errores en la programación del Robot. Expresando en términos de porcentajes tenemos un 10% de fracaso frente a 90% de éxito, éste último valor nos refleja el éxito en la detección de piezas de sensor Bit 900.

3.5 Prueba del Algoritmo de procesamiento de imágenes

Tabla 4-3: Prueba algoritmo procesamiento de imágenes

Número de imagen	Color de línea dado	Color de línea a la cual se programa	Indicador
1	Negro	Negro	Robotino no sigue línea requerida
2	Verde	Verde	Robotino sigue línea requerida
3	Rojo	Rojo	Robotino sigue línea requerida
4	Celeste	Celeste	Robotino no sigue línea requerida
5	Negro	Negro	Robotino sigue línea requerida
6	Verde	Verde	Robotino sigue línea requerida
7	Rojo	Rojo	Robotino sigue línea requerida
8	Celeste	Celeste	Robotino sigue línea requerida
9	Negro	Negro	Robotino sigue línea requerida
10	Verde	Verde	Robotino sigue línea requerida
11	Rojo	Rojo	Robotino sigue línea requerida
12	Celeste	Celeste	Robotino sigue línea requerida
13	Negro	Negro	Robotino sigue línea requerida
14	Verde	Verde	Robotino sigue línea requerida
15	Rojo	Rojo	Robotino sigue línea requerida
16	Celeste	Celeste	Robotino sigue línea requerida
17	Negro	Negro	Robotino sigue línea requerida
18	Verde	Verde	Robotino sigue línea requerida
19	Rojo	Rojo	Robotino sigue línea requerida
20	Celeste	Celeste	Robotino sigue línea requerida
	Total casos correctos		18
	Total casos fallidos		2
	TOTAL		20

Realizado por: Paucar, Alex; 2019

En la Tabla 4-3 se puede ver las pruebas que se realizaron para comprobar si el programa para la detección de líneas de acuerdo al color cargado en el Robotino funcionaba. Se dibujaron 4 líneas cada una de diferente color y se puso a prueba, el robot debía seguir una línea negra, verde, roja o celeste de acuerdo a lo programado.

PRUEBA DEL ALGORITMO PROCESAMIENTO DE IMÁGENES

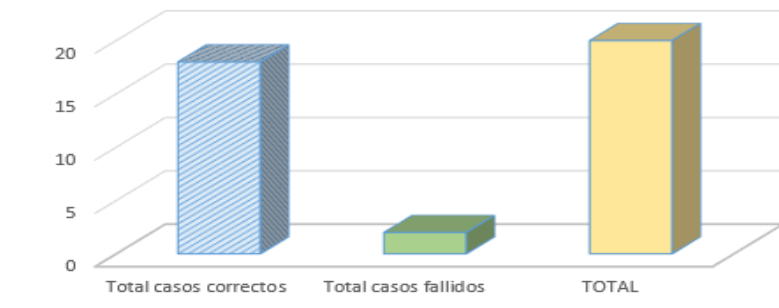


Gráfico 4-3: Prueba algoritmo procesamiento de imagines

Realizado por: Paucar, Alex; 2019

En Gráfico 4-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 20 pruebas, se cargaba el programa en el robot y se ejecutaba; si funcionaba correctamente el robot seguía el color de línea que se deseaba, caso contrario el robot giraba indefinidamente, de las 20 pruebas 4 eran para el color negro, 4 para el color rojo, 4 para el color verde y 4 para el color celeste. Se obtuvo un total de 18 casos correctos y 2 casos fallidos, los cuales se suscitaron porque no se tomaba el extracto correcto del color al momento de la programación. Expresando en términos de porcentajes tenemos un 10% de fracaso frente a 90% de éxito, éste último valor nos refleja el éxito en la programación y ejecución del código seguimiento de líneas por colores.

3.6 Pruebas de tiempos de ejecución de procesos

Tabla 5-3: Prueba tiempo programa-real

Número de programa	Tiempo dado en programa /s	Tiempo real /s	Diferencia/s
1	10	9,5	+0,5
2	10	9,5	+0,5
3	10	10	0
4	10	9,5	+0,5
5	10	10,5	-0,5
6	5	4,5	+0,5
7	5	5	0
8	5	5,5	-0,5
9	5	5	0
10	5	5,5	-0,5

Realizado por: Paucar, Alex; 2019

En la Tabla 5-3 se puede ver las pruebas que se realizaron para medir y comprar entre el tiempo de ejecución de proceso real y el tiempo dado en el programa, para lo cual se utilizó un contador incremental.

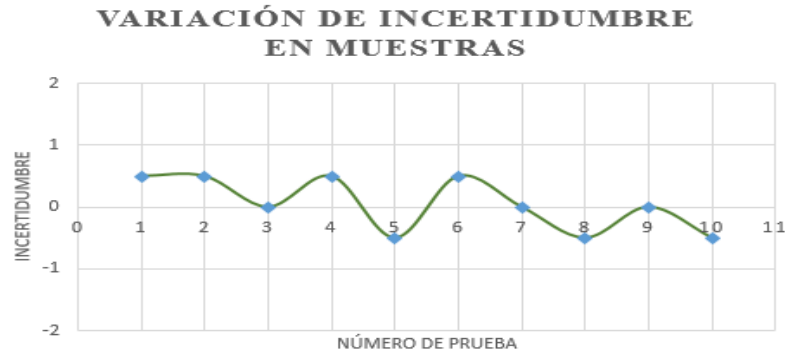


Gráfico 5-3: Variación de incertidumbre tiempo real - programa

Realizado por: Paucar, Alex; 2019

En Gráfico 5-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 10 pruebas, se cargaba el programa en el Robotino desde el PLC, de las 10 pruebas 5 eran con un tiempo de 10 segundos y las 5 eran con un tiempo de 5 segundos. Se obtuvo una incertidumbre de $\pm 0,5$, éste último valor nos refleja la mínima diferencia entre el valor real y el valor del programa.

Tabla 6-3: Pruebas tiempo de cada proceso

Número de prueba	Tiempo de transporte de pallet Posición inicial-Entrada Brazo Robótico /s	Tiempo de proceso de paletización de Brazo Robótico/s	Tiempo de transporte de pallet Salida Brazo Robótico-Salida sistema/s	Tiempo total de ejecución procesos completo/s
1	59	150	30	239
2	58	150	30	238
3	59	150	31	240
4	59	149	31	239
5	58	151	30	239
6	59	150	30	239
7	59	150	30	239
8	59	151	30	240
9	61	150	29	240
10	59	149	30	237

Realizado por: Paucar, Alex; 2019

En la Tabla 6-3 se puede ver las pruebas que se realizaron para medir el tiempo de los procesos principales de la secuencia total. Se obtuvo datos del tiempo de transporte de pallet Posición inicial-Entrada Brazo Robótico, tiempo de proceso de paletización de Brazo Robótico, tiempo de transporte de pallet Salida Brazo Robótico- Salida sistema y tiempo total de ejecución procesos completo.

3.7 Pruebas del Algoritmo de Posicionamiento

Tabla 7-3: Pruebas algoritmo de posicionamiento para X

Número de programa	Coordenada dada/ (x, y, phi)	Coordenada obtenida (x, y, phi)	Diferencia/m m
1	(500, 0, 0)	(510, 0, 0)	+10
2	(500, 0, 0)	(500, 0, 0)	0
3	(500, 0, 0)	(490, 0, 0)	+10
4	(700, 0, 0)	(700, 0, 0)	0
5	(700, 0, 0)	(710, 0, 0)	-10
6	(700, 0, 0)	(690, 0, 0)	+10
7	(1000, 0, 0)	(990, 0, 0)	+10
8	(1000, 0, 0)	(1000, 0, 0)	0
9	(1000, 0, 0)	(1000, 0, 0)	0
10	(1000, 0, 0)	(1010, 0, 0)	+10

Realizado por: Paucar, Alex; 2019

En la Tabla 7-3 se puede ver las pruebas que se realizaron para medir y comparar el algoritmo de posicionamiento del programa con la distancia recorrida real. Para dar las coordenadas de posición se utilizó una variable en donde se da los valores de X, Y y phi. En éste caso se dio valores a X se ejecutó el programa y una vez realizado el subproceso se midió el valor real y se comparó.



Gráfico 6-3: Variación incertidumbre posicionamiento X

Realizado por: Paucar, Alex; 2019

En Gráfico 6-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 10 pruebas, se enviaba la orden desde el software al Robotino y se medía distancias recorridas, de las 10 pruebas 3 se trabajó con una distancia de 500 mm, 3 con una distancia de 700 mm y 4 con una distancia de 1000 mm, de éste modo se obtuvo una incertidumbre de ± 10 , considerando que éste último es en milímetros se puede ver la mínima diferencia entre valores reales y del programa y la exactitud al trabajar con valores de posicionamiento.

Tabla 8-3: Pruebas algoritmo de posicionamiento para Y

Número de programa	Coordenada dada/ (x, y, phi)	Coordenada obtenida (x, y, phi)	Diferencia/mm
1	(0,500, 0)	(0,490, 0)	-10
2	(0,500, 0)	(0,510, 0)	+10
3	(0,500, 0)	(0, 490, 0)	+10
4	(0,700, 0)	(710, 0, 0)	+10
5	(0,700, 0)	(0,710, 0)	-10
6	(0,700, 0)	(0,690, 0)	+10
7	(0,1000, 0)	(0, 990, 0)	+10
8	(0,1000, 0)	(0, 1000, 0)	0
9	(0,1000, 0)	(0, 1010, 0)	+10
10	(0,1000, 0)	(0,1000, 0, 0)	0

Realizado por: Paucar, Alex; 2019

En la Tabla 8-3 se puede ver las pruebas que se realizaron para medir y comparar el algoritmo de posicionamiento del programa con la distancia recorrida real. Para dar las coordenadas de posición se utilizó una variable en donde se da los valores de X, Y y phi. En éste caso se dio valores a Y se ejecutó el programa y una vez realizado el subprocesso se midió el valor real y se comparó.

VARIACIÓN DE INCERTIDUMBRE EN MUESTRAS

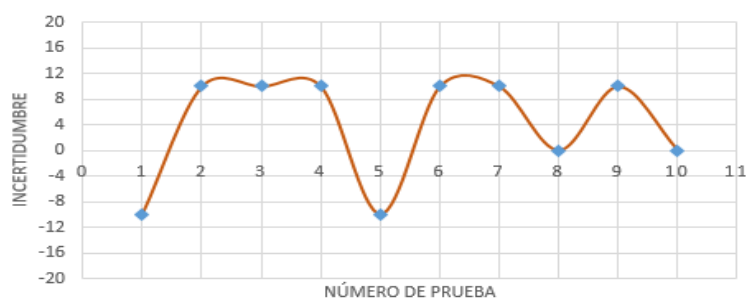


Gráfico 7-3: Variación incertidumbre posicionamiento Y

Realizado por: Paucar, Alex; 2019

En Gráfico 7-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 10 pruebas, se enviaba la orden desde el software al Robotino y se medía distancias recorridas, de las 10 pruebas 3 se trabajó con una distancia de 500 mm, 3 con una distancia de 700 mm y 4 con una distancia de 1000 mm, de éste modo se obtuvo una incertidumbre de ± 10 , considerando que éste último es en milímetros se puede ver la mínima diferencia entre valores reales y del programa y la exactitud al trabajar con valores de posicionamiento.

Tabla 9-3: Pruebas algoritmo de posicionamiento para phi

Número de programa	Coordenada dada/ (x, y, phi)	Coordenada obtenida (x, y, phi)	Diferencia/ grados
1	(0, 0, 0)	(0, 0, 0)	0
2	(0, 0, 0)	(0, 0, 0)	0
3	(0, 0, 0)	(0, 0, 0)	0
4	(0, 0, 90)	(0, 0, 92)	-2
5	(0, 0, 90)	(0, 0, 90)	0
6	(0, 0, 90)	(0, 0, 88)	+2
7	(0, 0, 180)	(0, 0, 182)	-2
8	(0, 0, 180)	(0, 0, 182)	-2
9	(0, 0, 180)	(0, 0, 178)	+2
10	(0, 0, 270)	(0, 0, 270)	0
11	(0, 0, 270)	(0, 0, 272)	-2
12	(0, 0, 270)	(0, 0, 272)	-2
13	(0, 0, 360)	(0, 0, 362)	-2
14	(0, 0, 360)	(0, 0, 360)	0
15	(0, 0, 360)	(0, 0, 360)	0

Realizado por: Paucar, Alex; 2019

En la Tabla 9-3 se puede ver las pruebas que se realizaron para medir y comparar el algoritmo de posicionamiento del programa con la distancia recorrida real. Para dar las coordenadas de posición se utilizó una variable en donde se da los valores de X, Y y phi. En éste caso se dio

valores a phi se ejecutó el programa y una vez realizado el subproceso se midió el valor real y se comparó.



Gráfico 8-3: Variación incertidumbre posicionamiento phi

Realizado por: Paucar, Alex; 2019

En Gráfico 8-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 15 pruebas, se enviaba la orden desde el software al Robotino y se medía ángulos de giro del robot sobre su propio eje, de las 15 pruebas 3 se trabajó con un ángulo de 0° , 3 con un ángulo de 90° , 3 con un ángulo de 180° , 3 con ángulo de 270° y 3 con un ángulo de 360° , de éste modo se obtuvo una incertidumbre de ± 2 , considerando que éste último es en grados puede ver la mínima diferencia entre valores reales y del programa y la exactitud al trabajar con valores de posicionamiento.

3.8 Prueba de funcionamiento del sistema

En esta prueba se ha puesto en funcionamiento la red en un período considerable, y se ha determinado la eficiencia del sistema así como el porcentaje de error que este presenta en el proceso de paletizado. A continuación en tablas presentamos los resultados.

Tabla 10-3: Número de pallets recogidos a la entrada del sistema

ENTRADA SISTEMA	CANTIDAD
BODEGA 1	10
BODEGA 2	10
TOTAL	20

Realizado por: Paucar, Alex; 2019

Tabla 11-3: Número de pallets colocados en home 4 o entrada del Robot RV2SDB

ENTRADA SISTEMA	CANTIDAD
CORRECTOS	18
ERRORES	2
TOTAL	20

Realizado por: Paucar, Alex; 2019

Tabla 12-3: Número de pallets colocados en la salida del Robot RV-2SDB

ENTRADA SISTEMA	CANTIDAD
CORRECTOS	18
ERRORES	0
TOTAL	18

Realizado por: Paucar, Alex 2019

Tabla 13-2: Número de pallets colocados en la Salida de Sistema.

ENTRADA SISTEMA	CANTIDAD
CORRECTOS	17
ERRORES	1
TOTAL	18

Realizado por: Paucar, Alex; 2019

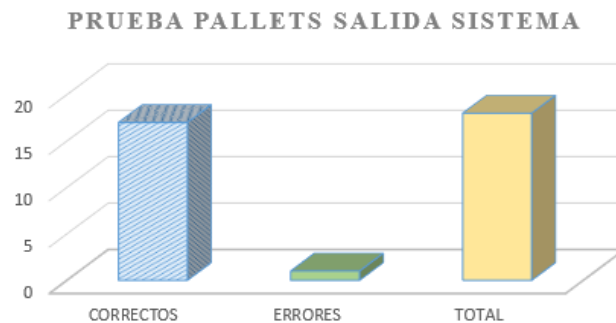


Gráfico 9-3: Prueba pallets salida sistema

Realizado por: Paucar, Alex; 2019

En Gráfico 9-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En ésta parte se puede ver barras que representan a los datos de las tablas, en donde se ver el éxito en la mayoría de pruebas ya que de las 20 que se realizaron solo 2 resultaron fallidas. Expresando en términos de porcentajes tenemos un 10% de fracaso frente a 90% de éxito.

3.9 Nivel de Automatizado del Sistema

Con esta prueba se logró determinar el qué porcentaje de todo el sistema es automático, además se determinará la eficiencia del sistema con una prueba chi cuadrada de ajuste de bondad.

Se ha tabulado un compendio de procesos que nos han permitido determinar el nivel de automatización de todo el sistema. En la siguiente tabla se muestra los procesos que son parte del sistema, a cada uno se le determinó si es automático o manual. (Sánchez Basantes & Sandoval Parra, 2018).

Tabla 14-3: Procesos involucrados en el sistema de comunicación.

N o.	Proceso	Tipo de proceso	
		Automático	Manual
1	Colocación de pallets en las bodegas 1 y 2		✓
2	Toma de pallets de las bodegas 1 y 2	✓	
3	Transporte de pallets desde las bodega hacia la entrada del robot RV-2SDB	✓	
4	Colocación de pallets en la entrada del robot rv2sdb	✓	
5	Transporte de pallets desde la entrada del robot rv2sdb hacia la estación del robot RV2SDB	✓	
6	Ubicación de piezas de trabajo sobre banda transportadora		✓
7	Transporte de piezas hacia el aerodeslizador	✓	
8	Transporte de piezas punto Home 5 (utilizando aerodeslizador)	✓	
9	Llenado de pallets con 4 piezas	✓	
10	Ubicación de pallets con piezas de trabajo en la salida del robot RV2SDB	✓	
11	Toma de pallets de la salida del robot RV2SDB	✓	
12	Transporte de pallets desde la salida del robot RV-2SDB hacia la salida del sistema	✓	
13	Toma de pallets de la salida del sistema	✓	
14	Detección de objetos con sensor Bit 900	✓	
15	Detección de pallet en Home 1 y Home 2	✓	
16	Accionamiento de motor banda transportadora		✓
17	Accionamiento de válvulas	✓	
18	Obtención de imágenes para procesamiento	✓	
19	Definición de pose en el Robotino®	✓	
20	Ejecución del algoritmo de posicionamiento mediante el concepto de seguidor de línea	✓	
21	Inicio de proceso Sistema		✓
22	Re inicio de algoritmo odometría en programa Robotino en la secuencia	✓	
23	Ubicación y espera en posición home del Robotino	✓	
24	Ubicación y espera en posición home del robot RV2SDB	✓	
25	Ubicación de Robotino en posición home		✓
	TOTAL	20	5

Realizado por: Paucar, Alex; 2019

En la Tabla 14-3 anterior se muestra un listado de los procesos involucrados en el funcionamiento del sistema, a cada uno de los procesos se los clasificados en dos grupos: automáticos y manuales, teniendo así un total de 25 procesos, de los cuales 20 son automáticos y los 5 restantes son manuales. Se tiene un 80 % de procesos automatizados.

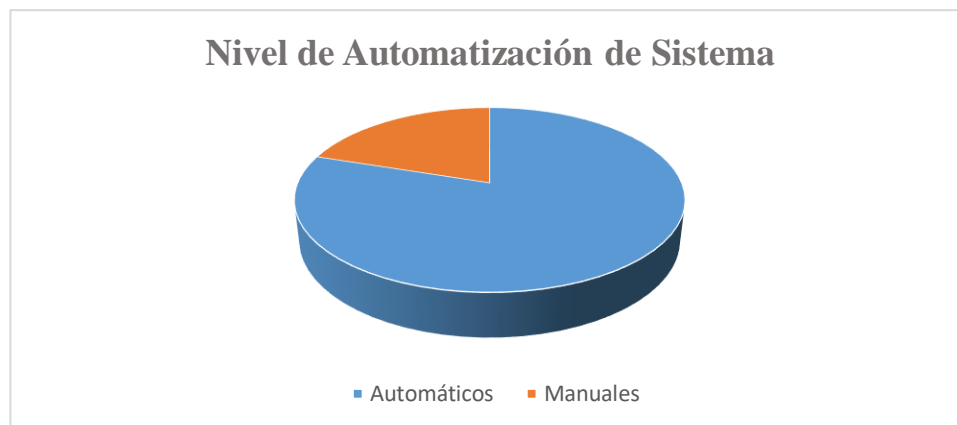


Gráfico 10-3: Nivel de automatización de los procesos

Realizado por: Paucar, Alex; 2019

En Gráfico 10-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se muestran un gráfico que muestra la superioridad en número de los procesos automáticos frente a procesos manuales, lo que representa el gran grado de autonomía del sistema ya que mediante programación en PLC se utilizaban señales de sensores para dar inicio o fin a un proceso sin la necesidad de la intervención del humano.

3.10 Prueba de Funcionamiento según el nivel de batería del Robotino

Tabla 15-3: Prueba Nivel de batería

Número de prueba	Nivel de batería/%	Proceso	Indicador
1	100	Conexión a internet	Ejecuta proceso
2	100	Inicio de un programa	Ejecuta proceso
3	100	Control desde internet	Ejecuta proceso
4	100	Comunicación UDP	Ejecuta proceso
5	50	Conexión a internet	Ejecuta proceso
6	50	Inicio de un programa	Ejecuta proceso
7	50	Control desde internet	Ejecuta proceso
8	50	Comunicación UDP	Ejecuta proceso
9	30	Conexión a internet	Ejecuta proceso
10	30	Inicio de un programa	Ejecuta proceso
11	30	Control desde internet	No ejecuta proceso
12	30	Comunicación UDP	No ejecuta proceso
13	28	Conexión a internet	No ejecuta proceso
14	28	Inicio de un programa	No ejecuta proceso
15	28	Control desde internet	No ejecuta proceso
16	28	Comunicación UDP	No ejecuta proceso
		Total casos correctos	10
		Total casos fallidos	6
		TOTAL	16

Realizado por: Paucar, Alex; 2019

En la Tabla 15-3 se puede ver las pruebas que se realizaron para comprobar el correcto funcionamiento del Robotino según su nivel de batería.

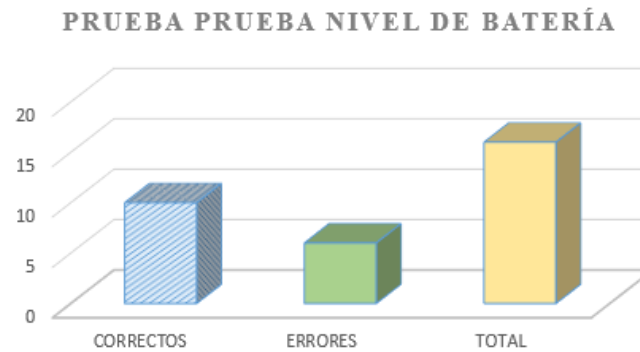


Gráfico 11-3: Prueba nivel de batería

Realizado por: Paucar, Alex; 2019

En Gráfico 11-3 se presenta un diagrama de barras en función a los datos de tabla anterior: En éste caso se realizaron 16 pruebas, de las cuales 10 casos eran correctos y 6 fallidos. Expresando en términos de porcentajes tenemos un 37% de fracaso frente a 63% de éxito. De éste modo se pudo determinar que:

- Al realizar una prueba de duración de batería, se logró identificar que el proceso está apta para un total de 1 hora de trabajo.
- Se pudo determinar que con niveles de batería inferiores al 30% el Robotino empieza a tener fallas en su funcionamiento y por ende en la comunicación con otros dispositivos y en ciertas ocasiones realiza parcialmente el proceso.
- Las baterías necesitan de 30 minutos para cargarse completamente.
- También se determinó que las baterías necesitan ser cambiadas puesto que en condiciones óptimas tienen una duración de 24 horas.

3.11 Aplicación de Prueba Estadística

En ésta punto es necesario entender lo que es una prueba estadística y su utilidad. Una prueba estadística se realiza una vez obtenidos los datos de muestra y se utiliza en una prueba de hipótesis, de éste modo se puede determinar si se puede o no rechazar la hipótesis nula. Se va a comparar datos con lo que se espera bajo la hipótesis nula. Al final se logra medir el grado de concordancia entre una muestra de datos y una hipótesis nula y se decide si se puede rechazar la hipótesis mencionada (Minitab, LLC, 2019).

Para realizar la prueba chi cuadrado de bondad de ajuste. Ésta prueba la utilizamos para discriminar si un conjunto de datos se ajusta a una distribución teórica de una determinada

población, es decir la utilizamos para para determinar si la muestra disponible representa (ajusta) razonablemente los datos que se esperaría encontrar en la población (Probabilidades y Estadística, 2017).

PRUEBA 1

Se establece un nivel de significancia de 0,05 y al existir dos clases los grados de libertad es igual a 1, con lo que se puede obtener un valor de chi cuadrado teórico de tabla y comparar con el valor calculado. Así:

- Grados de libertad = $k-1-m = 2-1-0 = 1$
- Margen de error = 0,05
- $x^2_t = 3,84$

Ahora se plantea la hipótesis nula “ H_o ” y la hipótesis alternativo “ H_i ”:

H_o : El uso de protocolo de comunicación UDP no permitirá la comunicación entre PLC y Robotino de manera satisfactoria.

H_i : El uso de protocolo de comunicación UDP permitirá la comunicación entre PLC y Robotino de manera satisfactoria.

$$x^2 = \frac{(fo - fe)^2}{fe}$$

Tabla 16-3: Prueba CHI cuadrada de bondad de ajuste en caso comunicación UDP

No.		Frecuencias observadas	Frecuencias Esperadas	x^2
1	Casos correctos	18	10,5	5,36
2	Casos fallidos	3	10,5	5,36
	TOTAL	21	21	
			$\sum x^2$	10,72

Realizado por: Paucar, Alex; 2019

$$x^2_c = 9$$

Criterios de comparación

Si: $x^2_c > x^2_t$ rechaza H_o .

Si: $x^2_c < x^2_t$ rechaza H_i .

De la Tabla 16-3 se comenta lo siguiente: Una vez realizada la prueba de Chi cuadro en Excel

se ha obtenido un valor de 10,72, con lo cual se puede descartar la hipótesis aplicando los criterios de comparación. Como el valor calculado es mayor que el valor de tabla (10,36>3,84) se descarta la hipótesis nula y se toma la hipótesis alternativa, por lo que se puede establecer que el uso de protocolo de comunicación UDP permitirá la comunicación entre PLC y Robotino de manera satisfactoria.

PRUEBA 2

Se establece un nivel de significancia de 0,05 y al existir dos clases los grados de libertad es igual a 1, con lo que se puede obtener un valor de chi cuadrado teórico de tabla y comparar con el valor calculado. Así:

- Grados de libertad = $k-1-m = 2-1-0 = 1$
- Margen de error = 0,05
- $\chi^2_t = 3,84$

Ahora se plantea la hipótesis nula “ H_o ” y la hipótesis alternativo “ H_i ”:

H_o : La implementación de un sistema inalámbrico con protocolo de comunicación inalámbrica dentro del sistema no mejorará el proceso de transporte del ROBOTINO PLUS de piezas paletizado del Brazo Robótico RV-2SDB.

H_i : La implementación de un sistema inalámbrico con protocolo de comunicación inalámbrica dentro del sistema mejorará el proceso de transporte del ROBOTINO PLUS de piezas paletizado del Brazo Robótico RV-2SDB.

$$\chi^2 = \frac{(fo - fe)^2}{fe}$$

Tabla 17-3: Prueba CHI cuadrada de bondad de ajuste en caso procesos automáticos y manuales

No.		Frecuencias observadas	Frecuencias Esperadas	χ^2
1	Automáticos	20	12.5	4.5
2	Manuales	5	12.5	4.5
	TOTAL	25	25	
			$\sum \chi^2$	9

Realizado por: Paucar, Alex; 2019

$$\chi^2_c = 9$$

Criterios de comparación

Si: $x^2_c > x^2_t$ rechaza H_0 .

Si: $x^2_c < x^2_t$ rechaza H_i .

De la Tabla 17-3 se comenta lo siguiente: Una vez realizada la prueba de Chi cuadro en Excel se ha obtenido un valor de 9, con lo cual se puede descartar la hipótesis aplicando los criterios de comparación. Como el valor calculado es mayor que el valor de tabla ($9 > 3,84$) se descarta la hipótesis nula y se toma la hipótesis alternativa, por lo que se puede establecer que la implementación de un sistema inalámbrico con protocolo de comunicación inalámbrica dentro del sistema sí mejorará el proceso de transporte del ROBOTINO de piezas paletizado del Brazo Robótico RV-2SDB.

CONCLUSIONES

- Se investigó sobre los protocolos de comunicación inalámbrica que permiten integrar el Robotino, el Brazo Robótico y la Mesa de Trabajo. De resultado de investigación se tiene el protocolo TCP y “UDP”. De TCP se determinó que es un protocolo orientado a conexiones, es pesado, se utiliza en aplicaciones que requieren confiabilidad alta y donde el tiempo de transmisión no es esencial. Por otro lado, de UDP se estableció que es protocolo sin conexiones, es liviano, se utiliza en aplicaciones que necesitan transmisión rápida y efectiva. Luego de haber visto las características de ambos protocolos se eligió el protocolo UDP que nos proporciona mayor rapidez y se adapta de mejor manera a nuestro sistema.
- Se determinaron los requerimientos que debía cumplir el sistema a implementar: Modo de comunicación inalámbrica, rapidez en intercambio de datos por lo que se trabaja con protocolo UDP, un PLC S7-1200 que trabaja con el protocolo seleccionado y un el número de entradas y salidas necesarias, robot comunicación UDP, robot con suficientes grados de libertad para procesos de paletización y una unidad de interfaz de robot para utilizar señales de sensores externos en programación de secuencias.
- Se implementó el sistema que cumplió con los requerimientos para la comunicación mediante protocolo UDP. Se logró comunicación inalámbrica mediante la utilización de un router, se estableció la comunicación con protocolo UDP entre el PLC y Robotino logrando rapidez en intercambio de datos, se colocaron sensores inductivos para detectar pallets y mediante utilización de cables y terminal Syslink se conectaron a la caja de interfaz la cual a su vez se comunicó con el Brazo Robótico. De éste modo, se pudo realizar la secuencia de transportar pallet de un punto a otro y realizar el proceso de paletización.
- Se evaluó el sistema implementado y se determinó que sí cumplía con los requerimientos, se realizaron pruebas para representar de manera estadística los resultados. Con la ayuda del HMI realizado en LabVIEW se realizó 21 pruebas para comprobar la comunicación UDP, de las cuales 18 resultaron exitosas, corroborando también así las características propias del protocolo. Al final con los resultados obtenidos de la prueba estadística de Chi cuadrado de bondad de ajuste se dieron por ciertas dos hipótesis: Que el uso de protocolo de comunicación UDP permitirá la comunicación entre PLC y Robotino de manera satisfactoria y que la implementación de un sistema inalámbrico con protocolo de comunicación inalámbrica dentro del sistema sí mejorará el proceso de transporte del ROBOTINO de piezas paletizado del Brazo Robótico RV-2SDB.

RECOMENDACIONES

- Tener en consideración que la superficie sobre la cual se trabaje con el Robotino sea plana ya que puede afectar al movimiento y posicionamiento del mismo.
- Cambiar las baterías del Robotino de modo que se pueda trabajar con todo su tiempo de vida útil y tener interrupciones o apagues del robot durante la ejecución de programas.
- Medir tiempos de transmisión de datos con protocolo de comunicación UDP utilizando software para analizar tráfico de red, tales como Wireshark, para ampliar el estudio realizado.
- Mejorar programa posicionamiento de Robotino mediante mapeo del área de la Mesa de Trabajo de modo que el Robot puede llegar a un punto deseado desde cualquier coordenada en la que se encuentre.
- Utilizar comandos específicos en la programación del Brazo Robótico para definir trayectoria de punto a punto de manera rápida, sin errores (que sea posible el movimiento del Robot) y segura (para el operador).
- Implementar un sistema inalámbrico de comunicación con protocolo TCP de modo que se puede comparar cada una de sus características con el protocolo UDP de manera real.
- Desarrollar proyectos en los cuales se pueda estudiar y aplicar de manera profunda todas las aplicaciones del Robotino en entornos didácticos y de competencia.

BIBLIOGRAFÍA

BALICH, Néstor Adrián. *Construcción de robots autónomos colaborativos*. [en línea], 2009.

[Consulta: 2019-05-21]. Disponible en:

<http://imgbiblio.vaneduc.edu.ar/fulltext/files/TC104132.pdf>

BLASCO, S., BUSTOS, J., & RIVERA, D. Detection and Containment Amortization UDP Sockets for Multithreading on Multicore Machines. *IEEE Latin America Transactions*, 14(6),

2853-2856. [en línea], 2016. [Consulta: 2019-05-22]. Disponible en:

<https://doi.org/10.1109/TLA.2016.7555264>

BORJA, M., & FERNANDA, M. *Diseño e implementación de una línea de clasificación asistida por un manipulador robótico MELFA RV – 2SBD, mediante el uso de visión artificial para generación de trayectorias reconfigurables y la incorporación de un sistema de seguridad activa*. [en línea], 2018. [Consulta: 2019-05-22]. Disponible en:

<http://repositorio.espe.edu.ec/jspui/handle/21000/15547>

BUSINESS, V. Cursos de ingeniería industrial. *Vester Training Center*. [en línea], 2018, abril 30. [Consulta: 2019-05-22]. Disponible en: <https://vestertraining.com/protocolos-comunicacion-redes-industriales/>

CASTRO, R. *Avanzando en la seguridad de las redes WIFI*. 10. [en línea], 2005. [Consulta: 2019-05-22]. Disponible en:

<https://www.rediris.es/difusion/publicaciones/boletin/73/ENFOQUE1.pdf>

DÁVILA, J., LÓPEZ, J., & ROMÁN, R. *Introducción de Aplicaciones UDP en Redes Privadas Virtuales*. 8. [en línea], 2001. [Consulta: 2019-05-22]. Disponible en:

<https://www.nics.uma.es/pub/papers/JorgeDavila2001.pdf>

DIFFEN. *TCP vs UDP Comparación / Diffen.* [en línea], 2017. [Consulta: 2019-05-23].

Disponible en: <https://es.diffen.com/tecnologia/TCP-vs-UDP>

Features RT ToolBox2 Software Industrial Robots-MELFA / MITSUBISHI ELECTRIC FA. [en

línea], 2019. [Consulta: 2019-05-23]. Disponible en:

<https://www.mitsubishielectric.com/fa/products/rbt/robot/smerit/rt2/index.html>

FESTO. *Robot Interface Unit.* [en línea], 2003. [Consulta: 2019-05-23]. Disponible en:

<https://www.festo-didactic.com/ov3/media/customers/1100/00573337001075223560.pdf>

FESTO. *Robotino® Plataforma de robot móvil para investigación y formación.* [en línea],

2013. [Consulta: 2019-06-02]. Disponible en: [https://www.festo-](https://www.festo-didactic.com/ov3/media/customers/1100/robotino_brochure_es_56941_2013_10_monitor.pdf)

[didactic.com/ov3/media/customers/1100/robotino_brochure_es_56941_2013_10_monitor.pdf](https://www.festo-didactic.com/ov3/media/customers/1100/robotino_brochure_es_56941_2013_10_monitor.pdf)

FESTO Didactic. *Baterías—Suministro eléctrico—Hardware—Robotino®—Servicios—Festo*

Didactic. [en línea], 2014. [Consulta: 2019-06-02]. Disponible en: [https://www.festo-](https://www.festo-didactic.com/int-es/servicios/robotino/hardware/suministro-electrico/baterias/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDY1)

[didactic.com/int-es/servicios/robotino/hardware/suministro-](https://www.festo-didactic.com/int-es/servicios/robotino/hardware/suministro-electrico/baterias/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDY1)

[electrico/baterias/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDY1](https://www.festo-didactic.com/int-es/servicios/robotino/hardware/suministro-electrico/baterias/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDY1)

FESTO Didactic. *Cámara—Sensores—Hardware—Robotino®—Servicios—Festo Didactic.*

[en línea], 2014. [Consulta: 2019-06-03]. Disponible en: [https://www.festo-didactic.com/int-](https://www.festo-didactic.com/int-es/servicios/robotino/hardware/sensores/camara/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDU0)

[es/servicios/robotino/hardware/sensores/camara/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDU](https://www.festo-didactic.com/int-es/servicios/robotino/hardware/sensores/camara/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDU0)

0

FESTO Didactic. *Giroscopio—Sensores—Hardware—Robotino®—Servicios—Festo*

Didactic. [en línea], 2014. [Consulta: 2019-06-05]. Disponible en: [https://www.festo-](https://www.festo-didactic.com/int-es/servicios/robotino/hardware/sensores/giroscopio/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDU0)

[didactic.com/int-](https://www.festo-didactic.com/int-es/servicios/robotino/hardware/sensores/giroscopio/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDU0)

es/servicios/robotino/hardware/sensores/giroscopio/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDUz

FESTO Didactic. *Interfaz E/S - Interfaces—Hardware—Robotino®—Servicios—Festo Didactic.* [en línea], 2014. [Consulta: 2019-06-06]. Disponible en: <https://www.festo-didactic.com/int-es/servicios/robotino/hardware/interfaces/interfaz-es/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDU5>

FESTO Didactic. *Sensores de distancias—Sensores—Hardware—Robotino®—Servicios—Festo Didactic.* [en línea], (2014e). [Consulta: 2019-06-06]. Disponible en: <https://www.festo-didactic.com/int-es/servicios/robotino/hardware/sensores/sensores-de-distancias/?fbid=aW50LmVzLjU1Ny4xNC4zNC4xNDUy>

FESTO Didactic GmbH & Co. KG. *Robotino® View 2.* [en línea], 2010. [Consulta: 2019-06-10]. Disponible en: http://doc.openrobotino.org/download/RobotinoView/RobotinoView2_ES.pdf

GRANADA, I. M., TURIZO, J., PORRAS, J. E., & SANTA, F. M. AUTOMATIZACIÓN DEL PROCESO DE PALETIZADO DE LA CELDA HAS-200. *Redes de Ingeniería*, 3(2), 92-99. [en línea], 2012. [Consulta: 2019-06-11]. Disponible en: <https://doi.org/10.14483/2248762X.6385>

GUTIÉRREZ SALAZAR, U. *ROBOT INDUSTRIAL MITSUBISHI Modelo RV-M1.* 33. [en línea], 2010. [Consulta: 2019-06-11]. Disponible en: <http://www.cnad.edu.mx/pdf/materialDidactico/manualRvm1.pdf>

HERÍAS, F. A. C., & BAEZA, J. P. *Manual de la Práctica 3: Protocolos de Transporte TCP y UDP.* 19. [en línea], 2009. [Consulta: 2019-06-11]. Disponible en: <https://rua.ua.es/dspace/bitstream/10045/11606/1/Pr3-2009-10.pdf>

HERÍAS, F. A. C., & BAEZA, J. P. 2009. *Manual de la Práctica 3: Protocolos de Transporte TCP y UDP*. 19. [en línea], (2009a). [Consulta: 2019-06-11]. Disponible en:
<https://rua.ua.es/dspace/bitstream/10045/11606/1/Pr3-2009-10.pdf>

HERRERO, S. 2015. *Sistemas inalámbricos*. [Consulta: 2019-06-11]. Disponible en:
http://www.rnds.com.ar/articulos/021/RNDS_124W.pdf

Interfaces y Protocolos de Comunicación. [en línea], 2012. [Consulta: 2019-07-02]. Disponible en: <http://tecsup-r5-ac-interfaces-y-protocolos.blogspot.com/>

JIMÉNEZ, J. G., & BATURONE, A. O. Estimación de la posición de un robot móvil. *Informática y automática: revista de la Asociación Española de Informática y Automática*, 29(4), 3-18. [en línea], 1996. [Consulta: 2019-07-03]. Disponible en:
<https://dialnet.unirioja.es/servlet/articulo?codigo=4274810>

JOMY Duarte. *Objetivos del buen vivir - analisis-* [Soziale Medien]. [en línea], 23:59:54 UTC. [Consulta: 2019-07-03]. Disponible en: https://de.slideshare.net/Diana_chaguay/objetivos-del-buen-vivir-analisis

JOSÉ CARLOS, Q.-Q., ERNESTO, F.-G., VÍCTOR, Q.-A., & JORGE, B.-L. Diseño e implementación de un sistema de control y monitoreo basado en HMI-PLC para un pozo de agua potable. *Ingeniería, Investigación y Tecnología*, 15(1), 41-50. [en línea], 2014. [Consulta: 2019-07-11]. Disponible en: [https://doi.org/10.1016/S1405-7743\(15\)30005-6](https://doi.org/10.1016/S1405-7743(15)30005-6)

KATONA, J., UJBANYI, T., SZILADI, G., & KOVARI, A. Speed control of Festo Robotino mobile robot using NeuroSky MindWave EEG headset based brain-computer interface. *2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 000251-

000256. [en línea], 2016. [Consulta: 2019-07-13]. Disponible en:

<https://doi.org/10.1109/CogInfoCom.2016.7804557>

marelysmolina21. Protocolos de comunicación. *Criterios de Selección de un PLC y Protocolos de Comunicación*. [en línea], 2017, abril 4. [Consulta: 2019-07-13]. Disponible en:

<https://protocolosdecomunicacionblog.wordpress.com/2017/04/04/protocolos-de-comunicacion/>

MARTÍNEZ, E. F., GUERRA, P. R., REYES, J. L. J., MELÉNDEZ, J. G., &

ESCOBEDO, Q. C. H. DISEÑO Y CONSTRUCCIÓN DE PROTOTIPO DE

ENTRENAMIENTO PARA PRÁCTICAS EN INSTRUMENTACIÓN Y CONTROL. *Pistas*

Educativas, 39(128). [en línea], 2018. [Consulta: 2019-07-13]. Disponible en:

<http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/view/1118>

MATEOS, F. *Sistema automatizado (PLC´s)*. [en línea], 2001. [Consulta: 2019-07-14].

Disponible en: <http://isa.uniovi.es/docencia/iea/teoria/plc.pdf>

MILANÉS HERMOSILLA, D., & CASTILLA PÉREZ, A. Generación de trayectorias para el brazo robótico (ArmX). *Ingeniería Electrónica, Automática y Comunicaciones*, 37(3), 58-71.

[en línea], 2016. [Consulta: 2019-07-14]. Disponible en:

[http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1815-](http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1815-59282016000300006&lng=es&nrm=iso&tlng=es)

[59282016000300006&lng=es&nrm=iso&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1815-59282016000300006&lng=es&nrm=iso&tlng=es)

MINITAB, LLC. *¿Qué es un estadístico de prueba?* [Mtbconcept]. [en línea], 2019.

[Consulta: 2019-07-14]. Disponible en: <https://support.minitab.com/es-mx/minitab/19/help-and-how-to/statistics/basic-statistics/supporting-topics/basics/what-is-a-test-statistic/>

mitsubishi electric. *RV-2SDB – Robots Industriales*. 4. [en línea], 2010. [Consulta:

2019-07-21]. Disponible en: [https://logismarketes.cdnwm.com/ip/mitsubishi-electric-europe-](https://logismarketes.cdnwm.com/ip/mitsubishi-electric-europe-robots-de-brazo-articulado-robots-de-brazo-articulado-melfa-serie-rv-2sdb-)

[robots-de-brazo-articulado-robots-de-brazo-articulado-melfa-serie-rv-2sdb-](https://logismarketes.cdnwm.com/ip/mitsubishi-electric-europe-robots-de-brazo-articulado-robots-de-brazo-articulado-melfa-serie-rv-2sdb-)

632568.pdf?fbclid=IwAR0Vfc-cs6-

GFYJa6CaddEh9YWVnrwN8aUoaISFdzTJMIvnJbHXXzzWGjw

MITSUBISHI ELECTRIC. *Robots Industriales MELFA*. [en línea], 2016. [Consulta: 2019-07-21]. Disponible en: <https://www.ncsolutions.es/wp-content/uploads/2016/04/Cat-Tecnico-Robots-206648h-ESP.pdf>

OÑATE LAZO, C. A., & PINTA YAUCAN, O. F. “*DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO PARA EL PROCESO DE CLASIFICACIÓN DE PIEZAS CONTROLADO MEDIANTE UN PLC SIEMENS S7-1200*”. [en línea], 2013. [Consulta: 2019-07-21]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/2898/1/108T0061.pdf>

ORTIGOZA, R. S., SÁNCHEZ, J. R. G., SOTELO, V. R. B., MOLINA, M. A., GUZMÁN, V. M. H., & ORTIGOZA, G. S. *UNA PANORÁMICA DE LOS ROBOTS MÓVILES*. 3, 15. [en línea], 2007. [Consulta: 2019-07-21]. Disponible en: <https://www.redalyc.org/pdf/784/78460301.pdf>

PAPCUN, P. *Center of Modern Control Techniques and Industrial Informatics—Mitsubishi robot*. [en línea], 2012. [Consulta: 2019-07-21]. Disponible en: <http://kyb.fei.tuke.sk/laben/modely/mit.php>

PÉREZ, C. *Controladores Lógicos Programables (PLCs)*. 21. [en línea], 2011. [Consulta: 2019-07-22]. Disponible en: http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE6_1_1.pdf

PÉREZ, E. M., ACEVEDO, J. M., & SILVA, C. F. *Automatas programables y sistemas de automatización / PLC and Automation Systems*. Marcombo. [en línea], 2009. [Consulta: 2019-07-22]. Disponible en: <https://books.google.es/books?id=5jp3bforBB8C>

PÉREZ, Z., & DANIEL, V. *Implementación de algoritmos de determinación de rutas para el robotino® de festo*. [en línea], 2015. [Consulta: 2019-07-22]. Disponible en:
<http://bibdigital.epn.edu.ec/handle/15000/11042>

PORTERO, A., LOZANO, J., & MIÑO, S. S. *Control de robot cartesiano mediante PLC S7-1200 de Siemens y Arduino para impresión 3D*. 10. [en línea], 2018. [Consulta: 2019-08-11].
Disponible en: http://dehesa.unex.es/bitstream/handle/10662/8317/978-84-09-04460-3_322.pdf?sequence=1&isAllowed=y

Probabilidades y Estadística. Prueba de Bondad de Ajuste Chi Cuadrado. *Probabilidades y Estadística*. [en línea], 2017, febrero 24. [Consulta: 2019-08-11]. Disponible en:
<https://www.probabilidadesyestadistica.com/prueba-de-bondad-de-ajuste-chi-cuadrado/>

¿Qué es un Router? ¿Y un Router WiFi? Comprar un router. Ampliar el alcance de un router. Tecnología + Informática. [en línea], 2018, marzo 30. [Consulta: 2019-08-11]. Disponible en:
<https://tecnologia-informatica.com/que-es-router-wifi-comprar-ampliar-alcance/>

Robotino® 4.0—Education and Research Robots: Robotino®—Learning Systems—Festo Didactic. [en línea], 2019. [Consulta: 2019-08-12]. Disponible en: <https://www.festo-didactic.com/int-en/learning-systems/education-and-research-robots-robotino/robotino-4.0.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC44NTguMTAzMDc1>

RODRÍGUEZ, C. F., BOTERO, J. C., & QUINTERO, H. Aplicación de la teoría de robots manipuladores a la biomecánica del brazo humano1. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 4(4), 51-57. [en línea], 2007. [Consulta: 2019-08-13]. Disponible en: [https://doi.org/10.1016/S1697-7912\(07\)70244-3](https://doi.org/10.1016/S1697-7912(07)70244-3)

ROSALES, A., SCAGLIA, G., MUT, V., & DI SCIASCIO, F. Navegación de Robots

Móviles en Entornos no Estructurados utilizando Álgebra Lineal. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 6(2), 79-88. [en línea], 2009. [Consulta: 2019-08-13].

Disponible en:

https://www.academia.edu/11720045/Navegaci%C3%B3n_de_Robots_M%C3%B3viles_en_Entornos_no_Estructurados_utilizando_%C3%81gebra_Lineal

SÁNCHEZ BASANTES, S. G., & SANDOVAL PARRA, R. I. *Diseño e implementación de*

una red M2M integrando el robot RV2SDB y el Robotino Plus, que incorpore un sistema de posicionamiento mediante procesamiento de imágenes. [en línea], 2018. [Consulta: 2019-08-

13]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/9215>

SIEMENS. *SIMATIC S7-1200.* [en línea], 2015. [Consulta: 2019-08-14]. Disponible en:

https://w5.siemens.com/spain/web/es/industry/automatizacion/sce_educacion/Documentacion-Didactica/Documents/SIMATIC%20S71200R.pdf

STAFF, L. Protocolos de comunicación industriales. *LOGICBUS.* [en línea], 2019, junio 17.

[Consulta: 2019-08-14]. Disponible en: <https://www.logicbus.com.mx/blog/protocolos-de-comunicacion-industriales/>

STILL PUEBLA, S. A. *Study and Design of a UDP/IP Protocol Infrastructure for Optical*

Underwater Communications. [en línea], 2018. [Consulta: 2019-08-14]. Disponible en:

<https://repositorio.unican.es/xmlui/bitstream/handle/10902/14509/410031.pdf?sequence=1&isAllowed=y>

TOLOSA, G. *Protocolos y Modelo OSI.* [en línea], 2015. [Consulta: 2019-08-15]. Disponible

en: <http://www.tyr.unlu.edu.ar/TYR-publica/02-Protocolos-y-OSI.pdf>

Universidad Don Bosco. *Programación del Robot Mitsubishi*. [en línea], (2016). [Consulta: 2019-08-16]. Disponible en: <http://www.udb.edu.sv/udb/archivo/guia/electronica-ingenieria/fundamentos-de-robotica/2016/ii/guia-7.pdf>

VILLALOBOS, L., PIMENTEL, K., RIVERA, O., & CARDONA, M. *Análisis Cinemático Directo del Robot RV-2AJ de Mitsubishi Utilizando Teoría de Screws*. 6. [en línea], 2015.

Disponible en:

http://rd.udb.edu.sv:8080/jspui/bitstream/11715/968/1/CONESCAPAN_2015_submission_30.pdf

VILLCA TAPIA, W. P. *Robótica*. *Revista de Información, Tecnología y Sociedad*, 132. [en línea], 2008. [Consulta: 2019-08-17]. Disponible en: www.revistasbolivianas.org.bo

WEBER, R.-C., & BELLENBERG, M. *Robotino*. Festo Didactic GmbH & Co. KG • 544305. [en línea], 2010. [Consulta: 2019-08-17]. Disponible en: https://www.festo-didactic.com/ov3/media/customers/1100/544305_robotino_esfr2.pdf

ANEXOS

Anexo A: Valores de tabla de Chi Cuadrado

DISTRIBUCION DE χ^2

Grados de libertad	Probabilidad										
	0,95	0,90	0,80	0,70	0,50	0,30	0,20	0,10	0,05	0,01	0,001
1	0,004	0,02	0,06	0,15	0,46	1,07	1,64	2,71	3,84	6,64	10,83
2	0,10	0,21	0,45	0,71	1,39	2,41	3,22	4,60	5,99	9,21	13,82
3	0,35	0,58	1,01	1,42	2,37	3,66	4,64	6,25	7,82	11,34	16,27
4	0,71	1,06	1,65	2,20	3,36	4,88	5,99	7,78	9,49	13,28	18,47
5	1,14	1,61	2,34	3,00	4,35	6,06	7,29	9,24	11,07	15,09	20,52
6	1,63	2,20	3,07	3,83	5,35	7,23	8,56	10,64	12,59	16,81	22,46
7	2,17	2,83	3,82	4,67	6,35	8,38	9,80	12,02	14,07	18,48	24,32
8	2,73	3,49	4,59	5,53	7,34	9,52	11,03	13,36	15,51	20,09	26,12
9	3,32	4,17	5,38	6,39	8,34	10,66	12,24	14,68	16,92	21,67	27,88
10	3,94	4,86	6,18	7,27	9,34	11,78	13,44	15,99	18,31	23,21	29,59
	No significativo								Significativo		

Anexo B: Realización de pruebas estadísticas en Excel

No.		Frecuencias observadas	Frecuencias esperadas	$\frac{(fo - fe)^2}{fe}$
1	Automáticos	20	12,5	4,5
2	Manuales	5	12,5	4,5
	TOTAL	25	25	
			$\sum \frac{(fo - fe)^2}{fe}$	9
	Grados de Libertad		1	
	Chi cuadrado de tabla		3,84	

Anexo C: Código Brazo Robótico en RT ToolBox2

```

1 | 'PROGRAMA PARA TESIS CON UDP'
2 | 'Llevada de pallet Entrada Robot a Home 2'
3 | servo on
4 | hopen 1
5 | dly 1.1
6 | ovrd 30
7 | dly 1.1
8 | ovrd 30
9 | mov Phome
10 | dly 1.2
11 | ovrd 30
12 | mov P41
13 | dly 1.1
14 | Ovrd 30

```

XYZ	X	Y	Z	A	B	C	L1	L2	FLG1	FLG2
P1	219.460	289.890	502.300	-145.420	3.490	-112.130	X	X	7	0
P2	371.280	63.140	391.430	-163.260	-36.390	-76.140	X	X	7	0
P3	433.880	202.210	267.380	-132.940	3.540	-155.020	X	X	7	0
P30	-42.340	-455.230	361.150	-154.980	40.700	37.840	X	X	7	0
P31	0.080	-416.180	459.050	-168.390	43.930	20.210	X	X	7	0
Joint	J1	J2	J3	J4	J5	J6	J7	J8		